REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)
14-11-2011

2. REPORT TYPE
Final Report

3. DATES COVERED (From - To)
1-Dec-2006 - 30-Jun-2011

4. TITLE AND SUBTITLE
SPATIO-TEMPORAL NONLINEAR FILTERING WITH APPLICATIONS TO INFORMATION ASSURANCE AND COUNTER TERRORISM

5a. CONTRACT NUMBER
W911NF-07-1-0044

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611103

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
A. Bertozzi, A. Galstyan, G. Medioni, C. Papadopoulos, B. Rozovsky,
A. Tartakovsky, V. Veeravalli

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Brown University
Office of Sponsored Projects
Brown University
Providence, RI 02912 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
51431-MA-MUR.99

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This final technical report outlines research and main results obtained during the period from May 1 2006 through October 31 2011 of the MURI project. The objective was to develop a general and systematic foundation and algorithms for spatiotemporal statistical inference and for fusion of heterogeneous information from multi-source, multi-sensor distributed sensor networks. Immediate applications of the proposed work are Network Centric Warfare, where new and emerging systems such as MASINT and FORCENet collect but do not adequately

15. SUBJECT TERMS
Spatiotemporal nonlinear filtering, changepoint detection, information fusion, video tracking, information assurance, rapid intrusion detection, detection and tracking of malicious terrorist activity, hidden Markov models, cybersecurity

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
UU	UU	UU

17. LIMITATION OF ABSTRACT
UU

15. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON
Boris Rozovsky

19b. TELEPHONE NUMBER
401-863-9246

Report Title

SPATIO-TEMPORAL NONLINEAR FILTERING WITH APPLICATIONS TO INFORMATION ASSURANCE AND COUNTER TERRORISM

ABSTRACT

This final technical report outlines research and main results obtained during the period from May 1 2006 through October 31 2011 of the MURI project. The objective was to develop a general and systematic foundation and algorithms for spatiotemporal statistical inference and for fusion of heterogeneous information from multi-source, multi-sensor distributed sensor networks. Immediate applications of the proposed work are Network Centric Warfare, where new and emerging systems such as MASINT and FORCENet collect but do not adequately interpret vast amounts of data; information assurance and network security; and homeland security applications, including video monitoring, and near-field and far-field intelligence analysis. Our research was targeted to solving three central problems: (a) nonstationarity, (b) integrating metric and symbolic information, and (c) very high dimensionality. Current methods for pattern recognition in monitoring and surveillance are designed for stationary patterns, and cannot cope with new patterns in ever-changing environments. We developed new statistical methods for the nonstationary environment, particularly spatiotemporal nonlinear filtering, changepoint detection, and advanced fusion methods. A distinctive feature of our approach is that the spaces in which estimation, classification and tracking is performed are both metric and symbolic.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
2011/11/09 1‹	60	G. O. Mohler. Self-Exciting Point Process Modeling of Crime, Journal of the American Statistical Association, (03 2011): 0. doi: 10.1198/jasa.2011.ap09546
2011/11/09 1‹	59	Rachel A. Hegemann, Laura M. Smith, Alethea B.T. Barbaro, Andrea L. Bertozzi, Shannon E. Reid, George E. Tita. Geographical influences of an emerging network of gang rivalries, Physica A: Statistical Mechanics and its Applications, (10 2011): 0. doi: 10.1016/j.physa.2011.05.040
2011/11/09 1‹	58	A. E. Allahverdyan, G. Ver Steeg, A. Galstyan. Community detection with and without prior information, EPL (Europhysics Letters), (04 2010): 0. doi: 10.1209/0295-5075/90/18002
2011/11/09 1‹	57	Aram Galstyan, Vahe Musoyan, Paul Cohen. Maximizing influence propagation in networks with community structure, Physical Review E, (5 2009): 0. doi: 10.1103/PhysRevE.79.056102
2011/11/09 1‹	56	Aram Galstyan, Paul Cohen. Cascading dynamics in modular networks, Physical Review E, (3 2007): 0. doi: 10.1103/PhysRevE.75.036109
2011/11/09 1‹	55	Yuping Lin, Qian Yu, Gérard Medioni. Efficient detection and tracking of moving objects in geo-coordinates, Machine Vision and Applications, (4 2010): 505. doi: 10.1007/s00138-010-0264-1
2011/11/09 1:	51	M. Short, P. Brantingham, M. D’Orsogna. Cooperation and punishment in an adversarial game: How defectors pave the way to a peaceful society, Physical Review E, (12 2010): 0. doi: 10.1103/PhysRevE.82.066114
2011/11/09 1:	52	Greg Ver Steeg, Aram Galstyan, Armen E Allahverdyan. Statistical mechanics of semi-supervised clustering in sparse graphs, Journal of Statistical mechanics, (08 2011): 0. doi: 10.1088/1742-5468/2011/08/P08009
2011/11/07 1‹	50	PAUL A. JONES, P. JEFFREY BRANTINGHAM, LINCOLN R. CHAYES. STATISTICAL MODELS OF CRIMINAL BEHAVIOR: THE EFFECTS OF LAW ENFORCEMENT ACTIONS, Mathematical Models and Methods in Applied Sciences, (01 2010): 1397. doi: 10.1142/S0218202510004647
2011/11/07 1‹	49	George K. Atia, Venugopal V. Veeravalli, Jason A. Fuemmeler. Sensor Scheduling for Energy-Efficient Target Tracking in Sensor Networks, IEEE Transactions on Signal Processing, (10 2011): 4923. doi: 10.1109/TSP.2011.2160055
2011/11/07 1‹	48	Jason A. Fuemmeler, George K. Atia, Venugopal V. Veeravalli. Sleep Control for Tracking in Sensor Networks, IEEE Transactions on Signal Processing, (09 2011): 4354. doi: 10.1109/TSP.2011.2159496
2011/11/07 1‹	47	Jason A. Fuemmeler, Venugopal V. Veeravalli. Energy Efficient Multi-Object Tracking in Sensor Networks, IEEE Transactions on Signal Processing, (07 2010): 3742. doi: 10.1109/TSP.2010.2046896
2011/11/07 1‹	46	Vasanthan Raghavan, Venugopal V. Veeravalli. Quickest Change Detection of a Markov Process Across a Sensor Array, IEEE Transactions on Information Theory, (04 2010): 1961. doi: 10.1109/TIT.2010.2040869
2011/11/07 1‹	45	J.A. Fuemmeler, V.V. Veeravalli. Smart Sleeping Policies for Energy Efficient Tracking in Sensor Networks, IEEE Transactions on Signal Processing, (05 2008): 2091. doi: 10.1109/TSP.2007.912265
2011/11/07 1‹	44	X HE, C PAPADOPOULOS, J HEIDEMANN, U MITRA, U RIAZ. Remote detection of bottleneck links using spectral and statistical methods, Computer Networks, (02 2009): 279. doi: 10.1016/j.comnet.2008.10.001
2011/11/07 1‹	43	Jelena Mirkovic, Peter Reiher, Christos Papadopoulos, Alefiya Hussain, Marla Shepard, Michael Berg, Robert Jung. Testing a Collaborative DDoS Defense In a Red Team/Blue Team Exercise, IEEE Transactions on Computers, (08 2008): 1098. doi: 10.1109/TC.2008.42
2011/11/07 1‹	42	Min Cai, Kai Hwang, Jianping Pan, Christos Papadopoulos. WormShield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation, , (04 2007): 88. doi: 10.1109/TDSC.2007.1000
2011/11/07 1‹	41	U. Mitra, A. Ortega, J. Heidemann, C. Papadopoulos. Detecting and identifying malware: a new signal processing goal, IEEE Signal Processing Magazine, (09 2006): 107. doi: 10.1109/MSP.2006.1708419
2011/11/07 1‹	6	Matthew S Keegan, Todd Wittman, LauraM Smith, George O Mohler, Andrea L Bertozzi. Improving Density Estimation by Incorporating Spatial Information, EURASIP Journal on Advances in Signal Processing, (01 2010): 0. doi: 10.1155/2010/265631

2011/11/07 1	40	M. B. Short, P. J. Brantingham, A. L. Bertozzi, G. E. Tita. From the Cover: Dissipation and displacement of hotspots in reaction-diffusion models of crime, Proceedings of the National Academy of Sciences, (02 2010): 0. doi: 10.1073/pnas.0910921107
2011/11/07 1	39	A. L. Bertozzi, P. J. Brantingham, M. B. Short. Nonlinear Patterns in Urban Crime: Hotspots, Bifurcations, and Suppression, Journal of Applied dynamical systems, (05 2010): 462. doi: 10.1137/090759069
2011/11/07 1	38	NANCY RODRIGUEZ, ANDREA BERTOZZI. LOCAL EXISTENCE AND UNIQUENESS OF SOLUTIONS TO A PDE MODEL FOR CRIMINAL BEHAVIOR, Mathematical Models and Methods in Applied Sciences, (01 2010): 1425. doi: 10.1142/S0218202510004696
2011/11/07 1	37	Sung Ha Kang, Berta Sandberg, Andy M. Yip. A regularized k-means and multiphase scale segmentation, Inverse Problems and Imaging, (5 2011): 407. doi: 10.3934/ipi.2011.5.407
2011/11/07 1	36	G. Prato, Michael Röckner, B. L. Rozovskii, Feng-yu Wang. Strong Solutions of Stochastic Generalized Porous Media Equations: Existence, Uniqueness, and Ergodicity, Communications in Partial Differential Equations, (01 2006): 277. doi: 10.1080/03605300500357998
2011/11/07 1	5	Erik Lewis, George Mohler, P Jeffrey Brantingham, Andrea L Bertozzi. Self-exciting point process models of civilian deaths in Iraq, Security Journal, (09 2011): 100. doi: 10.1057/sj.2011.21
2011/11/07 1	35	George O. Mohler. Fast TV Regularization for 2D Maximum Penalized Likelihood Estimation, Journal of Computational and Graphical Statistics, (06 2011): 0. doi: 10.1198/jcgs.2010.09048
2011/11/07 1	34	Wuan Luo, Boris Rozovskii, Thomas Y. Hou, Hao-Min Zhou. Wiener Chaos expansions and numerical solutions of randomly forced equations of fluid mechanics, Journal of Computational Physics, (8 2006): 687. doi: 10.1016/j.jcp.2006.01.008
2011/11/07 1	32	CHIA YING LEE, BORIS ROZOVSKII. A STOCHASTIC FINITE ELEMENT METHOD FORSTOCHASTIC PARABOLIC EQUATIONSDRIVEN BY PURELY SPATIAL NOISE, Communications on Stochastic Analysis, (03 2010): 271. doi:
2011/11/07 1	31	S. V. Lototsky, B. L. Rozovskii. Wiener chaos solutions of linear stochastic evolution equations, Annals of Probability, (03 2006): 638. doi: 10.1214/009117905000000738
2011/11/07 1	30	X. Wan, B. Rozovskii, G. E. Karniadakis. A stochastic modeling methodology based on weighted Wiener chaos and Malliavin calculus, Proceedings of the National Academy of Sciences, (08 2009): 14189. doi: 10.1073/pnas.0902348106
2011/11/07 1	29	Sergey V. Lototsky, Boris L. Rozovskii. Stochastic Partial Differential Equations Driven by Purely Spatial Noise, SIAM Journal on Mathematical Analysis, (08 2009): 1295. doi: 10.1137/070698440
2011/11/07 1	28	Sergey V. Lototsky, Boris L. Rozovskii, Xiaoliang Wan. Elliptic equations of higher stochastic order, Mathematical Modeling and Numerical Analysis, (8 2010): 1135. doi: 10.1051/m2an/2010055
2011/11/07 1	27	C. Y. Lee, B. L. Rozovskii, H. M. Zhou. Randomization of Forcing in Large Systems of PDEs for Improvement of Energy Estimates, Multiscale Model Simulation, (01 2010): 1419. doi: 10.1137/090766292
2011/11/07 1	26	A TARTAKOVSKY, B ROZOVSKII, R BLAZEK, H KIM. Detection of intrusions in information systems by sequential change-point methods, Statistical methodology, (07 2006): 252. doi: 10.1016/j.stamet.2005.05.003
2011/11/07 1	25	Alexander G. Tartakovsky, Boris L. Rozovskii, Rudolf B. Blazek, Hongjoong Kim. A Novel Approach to Detection of Intrusions inComputer Networks via Adaptive Sequential andBatch-Sequential Change-Point Detection Methods, IEEE Tran. on Signal Processing, (09 2006): 3372. doi:
2011/11/07 1	4	R. Mikulevicius, B. L. Rozovskii. On unbiased stochastic Navier–Stokes equations, Probability Theory and Related Fields, (8 2011): 0. doi: 10.1007/s00440-011-0384-1
2011/11/07 1	23	Alexander G. Tartakovsky, Venugopal V. Veeravalli. Asymptotically Optimal Quickest Change Detection in Distributed Sensor Systems, Sequential Analysis, (11 2008): 441. doi: 10.1080/07474940802446236
2011/11/07 1	22	M. Baron, A. G. Tartakovsky. Asymptotic Optimality of Change-Point Detection Schemes in General Continuous-Time Models, Sequential Analysis, (09 2006): 257. doi: 10.1080/07474940600609597

2011/11/07 1(	21	A. G. Tartakovsky. Asymptotic Optimality in Bayesian Changepoint Detection Problems under Global False Alarm Probability Constraint, Theory of Probability and its Applications, (01 2009): 443. doi: 10.1137/S0040585X97983754
2011/11/07 1(	20	M. Pollak, A. G. Tartakovsky. Asymptotic Exponentiality of the Distribution of First Exit Times for a Class of Markov Processes with Applications to Quickest Change Detection, Theory of Probability Applications, (01 2009): 430. doi: 10.1137/S0040585X97983742
2011/11/07 1(	19	Alexander G. Tartakovsky. Multidecision Quickest Change-Point Detection: Previous Achievements and Open Problems, Sequential Analysis, (05 2008): 201. doi: 10.1080/07474940801989202
2011/11/07 1(	17	Alenxander G. Tartakovsky. Discussion on "Is Average Run Length to False Alarm Always an Informative Criterion?" by Yajun Mei, Sequential Analysis, (01 2008): 396. doi:
2011/11/07 1(	16	Moshe Pollak, Alexander G. Tartakovsky. Optimality Properties of the Shiryaev-Roberts procedure, Statistica Sinica, (1 2009): 1729. doi:
2011/11/07 1(	15	Alexander Novikov, Alexander G. Tartakovsky, Robert Liptser. Preface: Celebrating Albert Shiryaev's 75th Anniversary, Sequential Analysis, (04 2010): 107. doi: 10.1080/07474941003740963
2011/11/07 1(	14	Robert Liptser, Alexander G. Tartakovsky. From Disorder Detection to Optimal Stopping and Mathematical Finance, Sequential Analysis, (04 2010): 112. doi: 10.1080/07474941003740971
2011/11/04 1!	13	George V. Moustakides, Aleksey S. Polunchenko, Alexander G. Tartakovsky. Numerical Comparison of CUSUM and Shiryaev–Roberts Procedures for Detecting Changes in Distributions, Communications in Statistics. Part A: Theory and Methods, (08 2009): 3225. doi: 10.1080/03610920902947774
2011/11/04 1!	12	Alexander G. Tartakovsky, George V. Moustakides. State-of-the-Art in Bayesian Changepoint Detection, Sequential Analysis, (04 2010): 125. doi: 10.1080/07474941003740997
2011/11/04 1!	11	Alexander G. Tartakovsky, George V. Moustakides. Discussion on “Quickest Detection Problems: Fifty Years Later” by Albert N. Shiryaev, Sequential Analysis, (10 2010): 386. doi: 10.1080/07474946.2010.520581
2011/11/04 1:	10	George V. Moustakides, Alexander G. Tartakovsky. A Note on “The Optimal Stopping Time for Detecting Changes in Discrete Time Markov Processes” by Han and Tsung, Sequential Analysis, (10 2010): 483. doi: 10.1080/07474946.2010.520633
2011/11/04 1:	8	Aleksey S. Polunchenko, Alexander G. Tartakovsky. On optimality of the Shiryaev--Roberts procedure for detecting a change in distribution, Annals of Statistics, (12 2010): 3445. doi:

TOTAL: **50**

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received		Paper
2011/11/07 1'	24	M. Baron, A.G. Tartakovsky. Asymptotic Optimality of Change-Point DetectionSchemes in General Continuous-Time Models, Sequential Analysis, (01 2006): 257. doi:
2011/11/04 1:	9	George V. Moustakides, Alexander G. Tartakovsky, Aleksey S. Polunchenko. A Numerical Approach to Performance Analysis of Quickest Change-Point Detection Procedures, Statistica Sinica, (04 2011): 571. do

TOTAL: **2**

Number of Papers published in non peer-reviewed journals:

(c) Presentations

1. A.G. Tartakovsky, “Spatiotemporal Image Processing with Applications to Remote Sensing,” Department of Statistics and Department of Computer Sciences, University of Chicago, September, 2011 (Invited).
2. A.G. Tartakovsky, “Sequential Changepoint Detection: State of the Art,” Department of Statistics, University of Illinois at Urbana-Champaign, October, 2011 (Invited).
3. A.G. Tartakovsky, “Spatiotemporal Image Processing with Applications to Remote Sensing,” Departmentof Electrical and Computer Engineering and Coordinated Science Lab, University of Illinois at Urbana-Champaign, October, 2011 (Invited).
4. A.S. Polunchenko, A.G. Tartakovsky, and N. Mukhopadhyay, “Nearly Optimal Change-Point Detection with An Application to Cybersecurity,” The 3rd International Workshop in Sequential Methodologies, Stanford University, 14–16 June 2011.
5. A.G. Tartakovsky and A.S. Polunchenko, “Optimality of the Shiryaev–Roberts Procedure for Detecting Changes in Distributions,” The 73rd Annual Meeting of the Institute of Mathematical Statistics, Gothenburg, Sweden, 9–13 August 2010.
6. A.P. Brown and A.G. Tartakovsky, “Spatiotemporal Clutter Rejection and Track-Before-Detect Methods for Tracking Small Dim Objects,” 32nd Review of Atmospheric Transmission Models Meeting, Lexington, Massachusetts, 14-15 June 2010 (Invited).
7. A.G. Tartakovsky, “Spatial-Temporal Image Processing Techniques and Applications to Remote Sensing,” Department of Mathematics, Stanford University, 2009 (Invited).
8. A.G. Tartakovsky, “Quickest Changepoint Detection: Recent Advances and Open Problems,” NSF Sponsored Workshop in Honor of Professor A.V. Balakrishnan, January 30, 2009 (Invited).
9. A.G. Tartakovsky, “Efficient Numerical Methods for Optimization and Performance Evaluation of Changepoint Detection Procedures,” Department of Probability, Moscow State University, Moscow, Russia, March 25, 2009 (Invited).
10. A.G. Tartakovsky, “Adaptive Spatial-Temporal Image Processing Techniques and Applications to Clutter Rejection in Remote Sensing,”Workshop “Spatiotemporal Image Processing and Visual Surveillance”, University of Southern California, 2008 (Invited).
11. A.G. Tartakovsky, “Exact Optimality of the Shiryaev–Roberts Procedure for Detecting Changes in Distributions,” Department of Mathematical Sciences, University of Technology, Sydney, Australia, November 27, 2008 (Invited).
12. A.G. Tartakovsky, “Detection and Classification in Distributed Multisensor Systems with Applications to Network Security,” Workshop “Sensor Networks and Future Internet Security”, University of Southern California, May 23, 2007 (Invited).
13. A.G. Tartakovsky, “Asymptotic Optimality in Sequential Quickest Change-Point Detection: Theory and Applications,” Princeton, September 25, 2007 (Invited).
14. A.G. Tartakovsky, “Quickest Change-Point Detection: Previous Achievements and Open Problems,” First International Workshop on Sequential Methodologies, Auburn, AL, 22-25 July 2007 (Invited).
15. A.G. Tartakovsky, “Asymptotic Optimality in Sequential Hypothesis Testing and Quickest Change-Point Detection for General Continuous-Time Stochastic Processes,” Workshop on Inverse Problems in Stochastic Differential Equations, University of Southern California, Los Angeles, CA, 22-26 May, 2007 (Invited).
16. A.G. Tartakovsky, “An Asymptotically Optimal Change Detection Strategy Under Nontraditional Global False Alarm Probability Constraint,” The 2007 Taipei International Statistical Symposium and ICSA International Conference (Session: Change-Point Analysis and Applications), Taipei, Taiwan, 24-28 June, 2007 (Invited).
17. V.V. Veeravalli and A.G. Tartakovsky, “Quickest Change Detection in Sensor Networks,” First International Workshop on Sequential Methodologies, Auburn, AL, July 2007 (Invited).
18. B.L. Rozovsky, “Generalized Malliavin calculus and Stochastic PDEs,” Columbia University, Minerva Foundation Lectures, December 2010.
19. B.L. Rozovsky, “Stochastic Fluid Dynamics,” NSF Institute for Pure and Applied Mathematics, Invited lecture, January 2011.
20. B.L. Rozovsky, “Stochastic Fluids and Malliavin Calculus,” Conference on Malliavin Calculus and Stochastic Analysis, University of Kansas, Invited talk, March 2011.
21. B.L. Rozovsky, “On Unbiased Stochastic Navier-Stokes Equation,” Workshop on SPDEs, Archimedes Center for Modeling, Analysis, and Computations, Heraklion, Greece, Invited lecture, June 2011.
22. B.L. Rozovsky, “Recent Advances in Nonlinear Filtering,” Imperial College, London. Invited lecture, June 2011.
23. B.L. Rozovsky, “Stochastic Fluid Dynamics and Malliavin Calculus,” Oxford University, Invited lecture, 2011.
24. B.L. Rozovsky, “Uncertainty Quantification and Nonlinear Filtering,” ICIAM 2011, Vancouver, Canada, 2011.
25. B.L. Rozovsky, “On Unbiased Stochastic Navier-Stokes Equation,” ICIAM 2011, Vancouver, Canada, 2011.
26. B.L. Rozovskii, Invited Talk, SIAM conference on Computational Science and Engineering, Maiami, 2009.
27. B.L. Rozovskii, Invited Talk, 7th ISAAC Congress, London, 2009.
28. B.L. Rozovskii, Invited Talk, International Conference on Spectral and High Order Methods, 2009, Trondheim, Norway.
29. Alethea Barbaro, Agent-based Complex Systems Workshop at IPAM: Organized and spoke October 14, 2009 1 hour, Title: ”Agent-based modeling for animal migration and gang behavior”.
30. Alethea Barbaro, American Soc. Criminology Meeting Philadelphia, joint presentation with Shannon Reid, Nov. 6, 2009, ”Agent-based simulations: modeling gang violence in Hollenbeck”.
31. Alethea Barbaro, UCSB Hypatian Seminar, Nov 30, 2010, ”Agent-based modeling of complex systems, and how to claim your mathematical territory after your doctorate”.

32. Alethea Barbaro, IPAM’s Optimal Transport Reunion Workshop at Lake Arrowhead (invited talk) December 10, 2009, 20 minutes, Title: ”On limits of a discrete time interacting particle system”
33. Alethea Barbaro, 2nd Annual Southern California Women in Math Symposium February 20, 2010,
30 minutes, Title: ”Agent-based models of social dynamics”.
34. Alethea Barbaro, 2010 Mathematics Festival at UCLA (2 sessions) February 13, 2010, Two sessions, each 50 minutes Title: ”Modeling the Real World: Using Math to Study Migration, Territoriality, and
Social Networks”.
35. Alethea Barbaro, Invited Talk at USC for theWomen in Math Seminar, March 12, 2010, 1 hour, Title: ”Simulating social dynamics with interacting particle models”.
36. Alethea Barbaro, Invited Seminar Talk at Redlands, March 31, 2010, 1 hour, Title: ”Simulating Social Dynamics with Interacting Particle Models”.
37. Alethea Barbaro, Talk at SIAM’s DSPDEs conference in Barcelona, Spain (Organized Mini-symposium and spoke), Mini-symposium title: Particle and mean field models for flocking and swarming, co-chair
Massimo Fornasier, June 1, 2010, 30 minutes, Talk Title: ”Interacting particle models for Social Dynamics”.
38. Alethea Barbaro, Workshop: Modeling Complex Dynamics in Biological Systems, Universite Paul Sabatier, Toulouse, France (invited talk) June 9, 2010 1 hour Title: ”Interacting particle models for animal social dynamics”.
39. Alethea Barbaro, Workshop: Mathematics of Complex Systems, Universite Paul Sabatier, Toulouse, France (invited talk) June 10, 2010, 45 minutes, Title: ”Agent-based models for gang dynamics”.
40. Alethea Barbaro, Kinetic and Mean-field models in the Socio-Economic Sciences: workshop at ICMS, Edinburgh, Scotland (25 minute invited talk), July 31, 2009, Title: ”Fish migration, interacting
particles, and scaling laws”.
41. Andrea Bertozzi, Women in Mathematics Seminar, Univ. of Wisconsin, Madison, WI, October 7, 2009.
42. Andrea Bertozzi, Invited talk and co-organizer, IPAM workshop on ”Agent Based Complex Systems”, October 14, 2009.
43. Andrea Bertozzi, Invited address, Southern California-Nevada MAA Section Meeting, October 17, 2009.
44. Andrea Bertozzi, Invited talk, Workshop on Self-Organization and Multi-Scale Mathematical Modeling of Active Biological Systems, Statistical and Applied Mathematical Sciences Institute, Durham,
NC, October 27 2009.
45. Andrea Bertozzi, Invited talk, Army Research Office, Durham, NC October 28, 2009.
46. Andrea Bertozzi, Invited talk UBC Vancouver, PIMS mini-symposium in PDE, one hour talk, November 13, 2009.
47. Andrea Bertozzi, Invited talk on “A Variational Approach to Hyperspectral Image Fusion”, Minisymposium on Variational Methods in Image Processing and Interface Problems, Maria Westdickenberg and Sung Ha Kang, Organizers, SIAM Conference on Analysis of Partial Differential Equations, Miami, December 7, 2009.
48. Andrea Bertozzi, Invited talk on ”Mathematical Models for Urban Crime” Minisymposium on “Nonlinear Stochastic PDEs and Applicationss to Complex Systems”, Hakima Bessaih and Bjorn Birnir, Organizers, SIAM Conference on Analaysis of Partial Differential Equations, Miami, December 8, 2009.
49. Andrea Bertozzi, Invited talk in SIAM Minisymposium on New Trends in Mathematical Methods in Imaging Science, Rick Chartrand, Stacey Levine, Jennifer Mueller, and Luminita Vese organizers, Joint Math Meetings, San Francisco, Sat Jan 16, 2010.
50. Andrea Bertozzi, Invited talk, China Lake Distinguished Speakers Colloquium Series, China Lake Naval Air Warfare Center, Ridgecrest, CA, Feb 2, 2010.
51. Andrea Bertozzi, Invited talk Rand Corp. Santa Monica, Feb 11, 2010.
52. Andrea Bertozzi, Invited talk, Session on ”Traffic, Crowds and Society”, AAAS Annual Meeting, San Diego, February 20, 2010.
53. Andrea Bertozzi, Invited talk, Imperial College London, Institute for Mathematical Sciences, invited talk in three part session on Geometric Mechanics, Darryl Holm host, March 8, 2010.
54. Andrea Bertozzi, Fluid Mechanics Seminar, DAMTP, Univ. of Cambridge, UK, March 5, 2010.
55. Andrea Bertozzi, Brown University, Mathematics Department, Distinguished Lecture Series, three one hour lectures, March 11-12, 2010.
56. Andrea Bertozzi, Brown University, Mathematics Department, faculty speaker, Symposium for Undergraduates in the Mathematical Sciences, 45 minute talk, March 13, 2010.
57. Andrea Bertozzi, Invited talk, Minisymposium on Advanced Frameworks for Restructuring High Dimensional Datasets, SIAM Conf. on Imaging Science, Chicago, IL April 13, 2010, Edward H. Bosch
Organizer.
58. Andrea Bertozzi, Invited talk, Plenary talk, Joint SIAM/RSME-SCM-SEMA Meeting on Emerging Topics in Dynamical Systems and Partial Differential Equations DSPDEs’10 June 1, 2010, Barcelona, Spain.
59. Andrea Bertozzi, Invited talk 2010 DTRA/NSF Algorithm workshop, talk on “Undergraduate Research Training in Defense Applications”, June 22, 2010, Chapel Hill, NC.
60. Andrea Bertozzi, Invited talk 2010 DTRA/NSF Algorithm workshop, talk on “Imaging of multispectral and hyperspectral data”, June

23, 2010, Chapel Hill, NC.
61. Andrea Bertozzi, invited talk talk in workshop Fluid Dynamics Analysis and Numerics, a conference in honor of Tom Beale's 60th Birthday, Duke Univ., Durham, NC June 28, 2010.
62. Andrea Bertozzi, invited talk at Park City Mathematics Institute, Program on Imaging Sciences, Park City UT, July 5, 2010.
63. Andrea Bertozzi, Graduate School of Engineering and Applied Sciences, Distinguished Lecture, Naval Postgraduate School, Sept. 2, 2010.
64. Andrea Bertozzi, London Taught Course Centre 8 hour intensive course on Mathematics of Crime, Univ. College London, Sept. 9-10, 2010.
65. Andrea Bertozzi, Department of Applied Mathematics and Statistics Johns Hopkins University, Colloquium Sept. 16, 2010.
66. Andrea Bertozzi, Allman Family Public Lecture, Southern Methodist University, Mathematics in the Real World, Sept. 23, 2010.
67. Andrea Bertozzi, Invited talk, IPAMworkshop on Machine Reasoning: Mission Focused Actions/Reactions Based on System Integration of Information Derived from Complex Real-World Data, Oct 19, 2010.
68. Andrea Bertozzi, Distinguished Lecture, Department of Mathematics, Simon Fraser Univ., Oct. 29, 2010.
69. Andrea Bertozzi, Invited talk, 9th Annual Image Fusion Workshop, Institute for Defense and Government Advancement, Tyson's Corner, VA, November 16, 2010.
70. Andrea Bertozzi, Invited talk, RCIM Symposium Mathematical Aspects of Image Processing and Computer Vision 2010 Sapporo, Japan, November 26, 2010.
71. Andrea Bertozzi, Invited talk, NSF workshop on New Directions in Dynamical Systems Inspired by Biological, Energy, Environmental, and Information Sciences, Atlanta, GA, Jan 4, 2011
72. Andrea Bertozzi, Invited talk, Dynamics Days, Chapel Hill, NC, Jan 5, 2011.
73. Andrea Bertozzi, AMS Invited Address, Joint Mathematics Meetings, New Orleans, LA, Jan 7, 2011.
74. Andrea Bertozzi, Invited Talk (one hour), 2011 annual meeting of the Australian and New Zealand Industrial and Applied Mathematics division of the Australian Mathemtaical Society. ANZIAM 2011
in Glenelg, Australia, Feb. 1, 2011.
75. Andrea Bertozzi, PIMS Applied Mathematics Seminar, University of Saskatchewan, Saskatoon, March 14, 2011.
76. Andrea Bertozzi, Seminar, Ecole Normal Superieur de Cachan, Centre de Mathematiques et de leurs Applications, March 17, 2011.
77. Andrea Bertozzi, Groupe de travail - Mathematiques de la decision, Seminar, Univ. of Toulouse, March 24, 2011.
78. Andrea Bertozzi, Colloquium de L'Institut de Mathematiques de Toulouse, March 25, 2011.
79. Andrea Bertozzi, Mathematics Colloquium Univ. of Warwick, UK, June 3, 2011, "Mathematics of Crime".
80. Andrea Bertozzi, Nonlinear Diffusion: Applications, Analysis and Computation conference to celebrate the 60th Birthday of Charlie Elliot, Univ. Warwick, June 6-8, 2011, invited 45 minute talk.
81. Andrea Bertozzi, 7th East Asian SIAM meeting, Waseda University Kitakyushu Campus, Japan Keynote Talk, June 29, 2011.
82. Andrea Bertozzi, Invited talk, Minisymposium on Modern Methods and Applications of the Calculus of Variations: Image Processing , July 20, 2011, International Congress on Industrial and Applied
Mathematics, Vancouver BC.
83. Andrea Bertozzi, Invited talk, Duke Workshop on Sensing and Analysis of High-Dimensional Data (SAHD), July 26, 2011.
84. Andrea Bertozzi, Plenary talk, AWM 40th Anniversary Conference, ICERM, Brown University, September 18, 2011.
85. Andrea Bertozzi, Applied Mathematics Seminar, Mathematics of Crime, Harvard University, September 19, 2011.
86. Andrea Bertozzi,Widely Applied Mathematics Seminar, Swarming by Nature and by Design, Harvard University, September 20, 2011.
87. P. Jeffrey Brantingham, Repeats and Reprisals: The Dynamics of Burglary and Rival Gang Violence in Los Angeles. Invited lecture at the Workshop on Modeling and Analysis of Security, January 4-7,
2010, University of Chile, 2010.
88. P. Jeffrey Brantingham, "Agent-based and continuum models of crime pattern formation," Invited lecture presented at the Agent-based Complex Systems workshop, Institute of Pure and Applied Mathematics, UCLA, October 12-14, 2009.
89. P. Jeffrey Brantingham, "Why seeking to reduce gang rivalries might increase gang violence," UC Irvine Criminology, Law and Society, April 13, 2011.
90. P. Jeffrey Brantingham, "The Mathematical Ecology of Criminal Street Gangs," UCLA Marschak Colloquium, April 8, 2011.
91. P. Jeffrey Brantingham, Stochastic Models of Crime with Practical Implications for Policing. Workshop on Geospatial Abduction Problems, University of Maryland, March 3-4, 2011.
92. P. Jeffrey Brantingham, "University-Agency Collaboration in Predictive Policing," 10th Anniversary Celebration of the Institute for Canadian Urban Research, Simon Fraser University, February 3, 2011.
93. Maria D'Orsogna, invited talk, Kinetic and mean-field models in the socio-economic sciences, Edinburgh, Scotland, July 2009
94. Erik Lewis, poster presentation at the Agent-Based Complex Systems workshop at IPAM, October 12-14, 2009. "Comparing Gang Rivalries and Civilian Deaths in Iraq Using Self-Exciting Point Processes."
95. George Mohler, "Agent-based, Bayesian Geographic Profiling" Workshop on Analysis and Modeling of Security," Jan 4-7, 2010, Santiago Chile, 40 minute talk.

96. George Mohler, “Crime as a Self-Exciting Point Process: An innovative approach in crime prediction,” American Society of Criminology annual meeting, Nov 4-7, 2009, 20 minute talk.
97. Todd Wittman, Contributed talk. 25 minutes. “Image Processing in the UCLA REU Program.” International Conference on Technology in Collegiate Mathematics. Chicago, IL. March 2010.
98. Todd Wittman, Contributed talk. 20 minutes. ”Problems in Geospatial Image Processing.” Center for Nonlinear Analysis Summer School on Image Processing and PDEs. Pittsburgh, PA. June 2010.
99. Todd Wittman, “The UCLA Math REU Program: Getting Students Involved in Research.” University of Southern California, Department of Mathematics Colloquium. Los Angeles, CA. April 2010.
100. Todd Wittman, Contributed talk. 25 minutes. “Variational Methods in Hyperspectral Image Processing.” SIAM Conference on Analysis of Partial Differential Equations. Miami, FL. December 2009.
101. Y. S. Cho, G. Ver Steeg, and A. Galstyan “Co-Evolving Mixed-Membership Blockmodels”, NIPS Workshop on Networks Across Disciplines, 2010.
102. A. Allahverdyan, A. Galstyan, and G.V. Steeg, “Clustering with Prior Information,” NIPS Workshop: Clustering: Science or Art? Towards Principled Approaches, 2009.
103. A. Galstyan, “Modeling Covert Activities with Hidden Markov Processes,” SIAM CADS Mini-symposium on Terrorism Modeled as a Dynamical System, Snowbird, Utah, 2009 (invited).
104. A. Galstyan and P.R. Cohen, “ Comparing Diffusion Models for Graph-Based Semi-Supervised Learning,” 6th International Workshop on Mining and Learning with Graphs (MLG-08), Helsinki, Finland,
2008.
105. A. Galstyan and P.R. Cohen, “Influence Propagation in Modular Networks,” AAAI Symposium on Social Information Processing (SIP-08), Stanford, CA, 2008.
106. A. Galstyan, S. Mitra, and P.R. Cohen, “Probabilistic Tracking of Plans and Intentions in Intelligence Analysis”, talk presented at the WNAR/IMS Annual Meeting, UC Irvine, June 2007.
107. A. Galstyan, S. Mitra, and P.R. Cohen, “Detecting and Tracking Hostile Plans in the Hats World,” AAAI Workshop on Plan, Activity and Intent Recognition (PAIR–07), Vancouver, Canada, 2007.
108. A. Galstyan, S. Mitra, and P.R. Cohen, “Probabilistic Plan Tracking and Detection for Intelligence Analysis,” poster presented as the Joint Statistical Meetings (JSM), Salt Lake City, July 2007.
109. G. Medioni, Keynote lecture, Workshop on Perceptual Organization, San Francisco, CA, June 13, 2010.
110. G. Medioni, Keynote lecture, International Workshop on Computer Vision, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, July 14, 2010.
111. G. Medioni, Invited lecture, “Recent progress in object tracking (Multi target tracking, tag and track, active tracking, tracking in flow)”, INRIA, Rocquencourt, France, October 2009.
112. G. Medioni, Keynote speaker, Los Angeles/Anaheim, 2009 World Congress on Computer Science and Information Engineering, March 31, 2009.
113. G. Medioni, Keynote speaker, San Diego (Coronado), Automated Imaging, February 4, 2009.
114. G. Medioni, Keynote speaker, ISVC, Las Vegas, November 22, 2008.
115. G. Medioni, “Tensor Voting in 2 to N dimensions: Fundamental Elements,” Distinguished Lecture, Brown University, September 15, 2008.
116. V.V. Veeravalli, “Sensor Control for Information Collection and Fusion.” International Workshop on Information Fusion, Xi’an, China, August 2011 (Plenary Lecture).
117. V.V. Veeravalli and T. Banerjee, “Quickest Change Detection with On-Off Observation Control.” International Workshop in Sequential Methodologies, Palo Alto, CA, June 2011 (Invited).
118. V.V. Veeravalli and J. Fuemmeler, “Energy-Efficient Multi-Target Tracking Using Sensor Networks.” Army Conference on Applied Statistics (ACAS), Lexington, VA, October 2008 (Invited).
119. V.V. Veeravalli, “System-Theoretic Foundations for Sensor Networks.” IEEE Communication Theory Workshop, Sedona, AZ, May 2007 (Keynote Lecture).
120. V.V. Veeravalli, “System-Theoretic Foundations for Sensor Networks.” IWWAN, New York, NY, June 2006 (Keynote Lecture).
121. V.V. Veeravalli, “Smart Sleeping Policies for Wireless Sensor Networks.” NSF Workshop on Future Directions in Networked Sensing, Boston, MA, May 2006 (Invited).

Number of Presentations: 121.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
2011/11/11 1:	98	Steve DiBenedetto, Kaustubh Gadkari, Nicholas Diel, Andrea Steiner, Dan Massey, Christos Papadopoulos. Fingerprinting custom botnet protocol stacks, 2010 6th IEEE Workshop on Secure Network Protocols (NPSec). 2010/10/04 03:00:00, Kyoto, Japan. : ,
2011/11/11 1:	97	Genevieve Bartlett, John Heidemann, Christos Papadopoulos. Low-rate, flow-level periodicity detection, IEEE INFOCOM 2011 - IEEE Conference on Computer Communications Workshops. 2011/04/09 03:00:00, Shanghai, China. : ,
2011/11/11 1:	96	Chris Wilcox, Christos Papadopoulos, John Heidemann. Correlating Spam Activity with IP Address Characteristics, IEEE INFOCOM 2010 - IEEE Conference on Computer Communications Workshops. 2010/03/14 04:00:00, San Diego, CA, USA. : ,
2011/11/11 1:	95	Genevieve Bartlett, John Heidemann, Christos Papadopoulos. Inherent Behaviors for On-line Detection of Peer-to-Peer File Sharing, 2007 IEEE Global Internet Symposium. 2007/05/10 03:00:00, Anchorage, AK, USA. : ,
2011/11/11 1:	94	A. Hussain, J. Heidemann, C. Papadopoulos. Identification of Repeated Denial of Service Attacks, Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications. 2006/04/22 03:00:00, Barcelona, Spain. : ,
2011/11/11 1:	93	V.V. Veeravalli, J.A. Fuemmeler. Efficient Tracking in a Network of Sleepy Sensors, 2006 IEEE International Conference on Acoustics Speed and Signal Processing. 2006/07/24 03:00:00, Toulouse, France. : ,
2011/11/11 1:	92	Jason A. Fuemmeler, Venugopal V. Veeravalli. Sensor scheduling for effective and energy efficient tracking in sensor networks, 2007 46th IEEE Conference on Decision and Control. 2007/12/11 03:00:00, New Orleans, LA, USA. : ,
2011/11/11 1:	91	Vasanthan Raghavan, Venugopal V. Veeravalli. Bayesian quickest change process detection, 2009 IEEE International Symposium on Information Theory - ISIT. 2009/06/27 03:00:00, Seoul, South Korea. : ,
2011/11/11 1:	90	Yuping Lin, Qian Yu, Gerard Medioni. Map-Enhanced UAV Image Sequence Registration, 2007 IEEE Workshop on Applications of Computer Vision (WAC V '07). 2007/02/20 03:00:00, Austin, TX, USA. : ,
2011/11/11 1'	89	Qian Yu, Gerard Medioni. Map-Enhanced Detection and Tracking from a Moving Platform with Local and Global Data Association, 2007 IEEE Workshop on Motion and Video Computing (WMVC'07). 2007/02/22 03:00:00, Austin, TX, USA. : ,
2011/11/11 1'	88	Qian Yu, Gerard Medioni, Isaac Cohen. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association, 2007 IEEE Conference on Computer Vision and Pattern Recognition. 2007/06/16 03:00:00, Minneapolis, MN, USA. : ,
2011/11/11 1'	87	Qian Yu, Gerard Medioni. A GPU-based implementation of motion detection from a moving platform, 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops). 2008/06/22 03:00:00, Anchorage, AK, USA. : ,
2011/11/11 1'	86	Qian Yu, Gerard Medioni. Integrated Detection and Tracking for Multiple Moving Objects using Data-Driven MCMC Data Association, 2008 IEEE Workshop on Motion and video Computing (WMVC). 2008/01/07 03:00:00, Copper Mountain, CO, USA. : ,
2011/11/11 1'	85	Qian Yu, Thang Ba Dinh, Gerard Medioni. Online Tracking and Reacquisition Using Co-trainedGenerative and Discriminative Trackers, 10th European Conference on Computer Vision. 2008/01/01 03:00:00, . : ,
2011/11/11 1'	84	Thang Dinh, Qian Yu, Gerard Medioni. Real time tracking using an active pan-tilt-zoom network camera, 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009). 2009/10/09 03:00:00, St. Louis, MO, USA. : ,

2011/11/11 1	83	Qian Yu, G. Medioni. Motion pattern interpretation and detection for tracking moving vehicles in airborne video, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops). 2009/06/19 03:00:00, Miami, FL. : ,
2011/11/11 1	82	Thang Ba Dinh, Gerard Medioni. Co-training framework of generative and discriminative trackers with partial occlusion handling, 2011 IEEE Workshop on Applications of Computer Vision (WACV). 2011/01/04 03:00:00, Kona, HI, USA. : ,
2011/11/11 1	81	Thang Ba Dinh, Nam Vo, Gerard Medioni. High resolution face sequences from a PTZ network camera, Gesture Recognition (FG 2011). 2011/03/20 03:00:00, Santa Barbara, CA, USA. : ,
2011/11/11 1	80	Thang Ba Dinh, Nam Vo, Gerard Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments, 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2011/06/19 03:00:00, Colorado Springs, CO, USA. : ,
2011/11/11 1	79	Aram Galstyan, Paul Cohen. Relational Classification Through Three¿State Epidemic Dynamics, 2006 9th International Conference on Information Fusion. 2006/07/09 03:00:00, Florence. : ,
2011/11/11 1	78	Armen Allahverdyan, Aram Galstyan. On maximum a posteriori estimation of hidden Markov processes, Twenty-Fifth Conference on Uncertainty in Artificial Intelligence . 2009/01/01 03:00:00, . : ,
2011/11/10 2	77	Andrea L. Bertozzi, Zhipu Jin. Environmental boundary tracking and estimation using multiple autonomous vehicles, 2007 46th IEEE Conference on Decision and Control. 2007/12/11 03:00:00, New Orleans, LA, USA. : ,
2011/11/10 2	76	Wangyi Liu, Martin B. Short, Yasser E. Taima, Andrea L. Bertozzi. Multiscale Collaborative Searching Through Swarming, 7th International Conference on Informatics in Control, Automation, and Robotics. 2010/06/12 03:00:00, . : ,
2011/11/10 2	75	Yao-Li Chuang, Yuan R. Huang, Maria R. D'Orsogna, Andrea L. Bertozzi. Multi-Vehicle Flocking: Scalability of Cooperative Control Algorithms using Pairwise Potentials, 2007 IEEE International Conference on Robotics and Automation. 2007/04/09 03:00:00, Rome, Italy. : ,
2011/11/10 2	74	Kevin K. Leung, Chung H. Hsieh, Yuan R. Huang, Abhijeet Joshi, Vlad Voroninski, Andrea L. Bertozzi. A second generation micro-vehicle testbed for cooperative control and sensing strategies, 2007 American Control Conference. 2007/07/08 03:00:00, New York, NY, USA. : ,
2011/11/10 2	73	Y. Landa, D. Galkowski, Y. R. Huang, A. Joshi, C. Lee, K. K. Leung, G. Malla, J. Treanor, V. Voroninski, A. L. Bertozzi, R. Tsai. Robotic path planning and visibility with limited sensor data, American Control Conference. 2007/07/30 03:00:00, . : ,
2011/11/10 2	72	A. Joshi, T. Ashley, Y. Huang, A.L. Bertozzi. Experimental validation of cooperative environmental boundary tracking with on-board sensors, American Control Conference. 2009/06/01 03:00:00, . : ,
2011/11/10 2	71	J.H. von Brecht, S. Thiruvenadam, T.F. Chan. OCCLUSION TRACKING USING LOGIC MODELS, 9th IASTED Conference on Signal and Image Processing. 2007/01/01 03:00:00, . : ,
2011/11/10 2	70	M. Gonzalez, X. Huang, B. Irvine, D. S. Hermina Martinez, C. H. Hsieh, Y. R. Huang, M. B. Short, A. L. Bertozzi. A Third Generation Micro-vehicle Testbed for Cooperative Control and Sensing Strategies, 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO). 2011/01/01 03:00:00, . : ,
2011/11/10 2	69	Alexander G. Tartakovsky, H. Kim. Performance of Certain Decentralized Distributed Change Detection Procedures, 9th International Conference on Information Fusion. 2006/07/10 03:00:00, . : ,
2011/11/10 2	68	Alexander G. Tartakovsky, Aleksey S. Polunchenko. Decentralized Quickest Change Detection in Distributed Sensor Systems with Applications to Information Assurance and Counter Terrorism, 13th Annual Army Conference on Applied Statistics. 2007/10/18 03:00:00, . : ,
2011/11/10 2	67	Moshe Pollak, Alexander G. Tartakovsky, Aleksey S. Polunchenko. Asymptotic Exponentiality of First Exit Timesfor Recurrent Markov Processes andApplications to Changepoint Detection, 2008 International Workshop on Applied Probability. 2008/07/07 03:00:00, . : ,

2011/11/10 2:	66	Alexander G. Tartakovsky, Aleksey S. Polunchenko. Quickest Changepoint Detection in Distributed Multisensor Systems under Unknown Parameters, 11th International Conference on Information Fusion. 2008/07/01 03:00:00, . : ,
2011/11/10 2:	65	Alexander G. Tartakovsky, James Brown, Andrew Brown. Nonstationary EO/IR Clutter Suppression and Dim Object Tracking, 2010 Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference. 2010/09/20 03:00:00, . : ,
2011/11/10 2:	64	Georger V. Moustakides, Alexander G. Tartakovsky, Aleksey S. Polunchenko. Design and Comparison of Shiryaev-Roberts and CUSUM-TypeChange-Point Detection Procedures, The Second International Workshop on Sequential Methodologies . 2009/06/16 03:00:00, . : ,
2011/11/10 2:	63	Nitis Mukhopadhyay, Alexander Tartakovsky, Aleksey Polunchenko. Nearly Optimal Change-Point Detection withAn Application to Cybersecurity, The Third International Workshopin Sequential Methodologies. 2011/06/15 03:00:00, . : ,
2011/11/10 2:	62	Moshe Pollak, Alexander G. Tartakovsky. Nearly Minimax Changepoint Detection Procedures, IEEE International Symposium on Information Theory. 2011/08/01 03:00:00, . : ,
2011/11/10 2:	61	Alexander G. Tartakovsky, Moshe Pollak. Monotone Properties of the First Exit Time of a MarkovProcess Started at a Quasi-stationary Distribution, Markov and Semi-Markov Processes and Related Fields 2011. 2011/09/20 03:00:00, . : ,
2011/11/09 1	54	Qian Yu, Gerard Medioni, Isaac Cohen. Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association, 2007 IEEE Conference on Computer Vision and Pattern Recognition. 2007/06/16 03:00:00, Minneapolis, MN, USA. : ,

TOTAL: 39

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
2011/11/10 1	7	Moshe Pollak, Alexander G. Tartakovsky. On the first exit time of a nonnegative Markov process started at a quasistationary distribution, Journal of Applied Probability ( )
2011/11/09 1	33	Sergey V. Lototsky, Boris L. Rozovsky. Stochastic Parabolic Equations of Full Second Order, ( )
2011/11/02 1	3	Alexander G. Tartakovsky, Moshe Pollak, Aleksey S. Polunchenko. Third-order Asymptotic Optimality of the Generalized Shiryaev-Roberts Detection Procedures, Theory of Probability Applications ( )
2011/11/02 1	2	Georgios Fellouris, Alexander G. Tartakovsky. Nearly Minimax Mixture Rules for One-sided Sequential Testing, Sequential Analysis ( )

TOTAL: 4

Number of Manuscripts:

Books

Received		Paper
2011/11/09 1	53	Boris Rozovsky, Dan Crisan. the oxford handbook of nonlinear filtering, United Kingdom: Oxford University Place, (01 2011)

TOTAL: 1

Patents Submitted

1. G. Medioni and Q. Yu, USC File No:4048 “Spatio-Temporal Multiple Target Tracking Using Markov Chain Monte Carlo Data Association”.
2. G. Medioni and Q. Yu, USC File No:4109 “Online Tracking Using Co-trained Generative and Discriminative Trackers”.
3. G. Medioni and T. B. Dinh, USC File No:11-671, “Visual Tracking in Video Images in Unconstrained Environments by Exploiting On-The-Fly Context Using Distracters and Supporters”.
4. Provisional patent application filed by UCLA with US Patent Office titled “Data Fusion Mapping Estimation”.

Patents Awarded

Awards

1. T.B. Dinh, Travel Grant to participate in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, Colorado, June 20-25, 2011.
2. T.B. Dinh, Travel Grant from NSF to participate n IEEE Conference on Automatic Face and Gesture Recognition (FG), Santa Barbara, California, March 21-25, 2011.
3. A.S. Polunchenko, Institute of Mathematical Statistics’ (IMS) Laha Travel Award, 2011.
4. Laura Smith, UCLA Dissertation Year Fellowship, 2011.
5. Nancy Rodriguez, National Science Foundation Postdoctoral Fellowship, 2011.
6. Andrea Bertozzi, Elected American Academy of Arts and Sciences, 2010.
7. Andrea Bertozzi, Elected SIAM Fellow, 2010.
8. Tony Chan, Elected SIAM Fellow, 2010.
9. V. Veeravalli was appointed IEEE Signal Processing Society Distinguished Lecturer for 2010-2011.
10. Andrea Bertozzi, Sonia Kovalevsky Prize Lecture, SIAM Annual Meeting, 2009.
11. P. Jeffrey Brantingham, (2009-Present) Los Angeles Police Department, Community Police Advisory Board on Counter-Terrorism, Appointed Board Member.
12. T.B. Dinh, 1st runner-up presentation award in scientific sessions of Annual Vietnam Education Foundation, Albany, NY, Jan 3-5, 2009.
13. Vlad Voroninsky, NSF Graduate Fellowship, 2008.
14. Tony Chan, Elected AAAS Fellow, 2007.
15. Alexander Tartakovsky, Abraham Wald Prize in Sequential Analysis, 2007.

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Greg Sokolov	0.50	
Thang Ba Dinh	0.25	
Jan Prokaj	0.25	
Yoon Sik Cho	0.50	
Ming Ji	0.25	
Alexander Chen	0.25	
Nancy Rodriguez	0.25	
Paul Jones	0.25	
Matthew Keegan	0.25	
Laura Smith	0.25	
Erik Lewis	0.25	
Rachel Hegemann	0.25	
Wangyi Liu	0.25	
Yasser Taima	0.25	
Kevin Shen	0.25	
David Hermina	0.25	
James von Brecht	0.25	
Jason Fuemmeler	0.50	
Taposh Banerjee	0.25	
Steven DiBenedetto	0.25	
Kaustubh Gadkari	0.25	
Mengran Hu	0.25	
Zhang Han	0.25	
Andrea Steiner	0.25	
Andrew Papanicolaou	0.50	
C.-Y. Lee	0.25	
FTE Equivalent:	**7.50**	
Total Number:	**26**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
J. Park	0.25
Greg Ver Steeg	0.50
Aleksey Polunchenko	0.50
Vasanthan Raghavan	0.10
Georgios Fellouris	0.10
Berta Sandberg	0.25
Virginia Pasour	0.25
Zhipu Jin	0.25
FTE Equivalent:	**2.20**
Total Number:	**8**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Boris Rozovsky	0.10	
Aram Galstyan	0.25	
Paul Cohen	0.10	
Gerard Medioni	0.10	
Alexander Tartakovsky	0.30	
Andrea Bertozzi	0.10	
Tony Chan	0.10	
P. Jeffrey Brantingham	0.10	
Venugopal Veeravalli	0.20	
Christos Papadopoulos	0.10	
FTE Equivalent:	**1.45**	
Total Number:	**10**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Marina Masaki	0.25	Mathematics
Kym Louie	0.25	Mathematics
Mark Allenby	0.25	Mathematics
Mike Egesdal	0.25	Mathematics
Chris Fathauer	0.25	Mathematics
Jeremy Neumann	0.25	Mathematics
Benjamin Irvine	0.25	Mathematics
Max Gonzales	0.25	Mathematics
Edwin Huang	0.25	Mathematics
Abhijeet Joshi	0.25	Mathematics
Kevin Leung	0.25	Mathematics
Vlad Voroninsky	0.25	Mathematics
Trevor Ashley	0.25	Mathematics
Andrea Steiner	0.25	Computer and Computational Sciences
FTE Equivalent:	**3.50**	
Total Number:	**14**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	12.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......	12.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......	10.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):......	12.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:......	0.00

Names of Personnel receiving masters degrees

NAME
Kevin Shen
David Hermina-Martinez
C. Y. Lee

Total Number: **3**

Names of personnel receiving PHDs

NAME
Aleksey Polunchenko
Andrew Papanicolaou
Qian Yu
Alex Chen
Nancy Rodriguez
Paul Jones
Jason Fuemmeler
C. Y. Lee
Dinh Ba Thang

Total Number: **9**

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See the attached file

Technology Transfer

SPATIO-TEMPORAL NONLINEAR FILTERING WITH APPLICATIONS TO INFORMATION ASSURANCE AND COUNTER TERRORISM

FINAL TECHNICAL REPORT
GRANT # W911NF-07-1-0044
REPORT # FTR-1-11
DATES COVERED: 05/01/2006–10/31/2011

AUTHORS:
A. Bertozzi, A. Galstyan, G. Medioni,
C. Papadopoulos, B. Rozovsky, A. Tartakovsky, and V. Veeravalli

LEADING PERFORMING ORGANIZATION:

Department of Applied Mathematics
Brown University
Providence, RI 02912

Principal Investigator: Boris Rozovsky

Co-Principal Investigator: Alexander Tartakovsky

SPONSORING/MONITORING AGENCY:

The U.S. Army Research Office
Mathematical Sciences Division
P.O. Box 12211
RTP, NC 27709

POC: Dr. Mou-Hsiung Chang

Contents

List of Tables

List of Figures

Chapter 1

Numerical Data Related to This Grant

1. Scientific Personnel Supported by This Project

Brown University (Brown):

1. Boris Rozovsky, Professor
2. Remis Mikulevicius, Consultant, Professor, USC
3. Sergey Lototsky, Consultant, Professor, USC
4. Andrew Papanicolaou, PhD student in Mathematics
5. C.-Y. Lee, PhD student in Applied Mathematics
6. J. Park, Postdoctoral Fellow
7. S. Kaligotla, Visitor, USC
8. S. Peszat, Consultant, Institute of Mathematics, Poland
9. A. Debussche, Consultant, Ecole Normale Superiore, France
10. M. Tretyakov, Consultant, University of Leicester, UK
11. M. Kanovich, Visitor, University of London, UK
12. D. Selesi, Visitor, University of Novi Sad, Serbia
13. M. Sanz-Sole, Visitor, University of Barcelona, Spain
14. B Maslovski, Visitor, Academy of Science of Czech Republic
15. P. Sundar, Visitor, LSU
16. I. Sonin, Visitor, UNC, Charlotte
17. H. Zhou, Visitor, LSU

University of Southern California (USC):

1. Aram Galstyan, Project Leader, Information Sciences Institute
2. Paul Cohen, Project Leader, Information Sciences Institute
3. Gerard Medioni, Professor, Computer Science
4. Alexander Tartakovsky, Associate Director, Research Professor, Department of Mathematics
5. George Moustakides, Professor, ECE Department, University of Patras, Greece, visitor and consultant
6. Moshe Pollak, Professor, Statistics Department, The Hebrew University of Jerusalem, Israel, visitor and consultant
7. Olympia Hadjiliadis, Assistant Professor, Department of Mathematics, SUNY, short-term visitor
8. Alexander Novikov, Professor, Department of Mathematical Sciences, UTS, Sydney, Australia, short-term visitor
9. Albert Shiryaev, Professor, Department of Probability, Moscow State University, Russia, short-term visitor
10. Sergei Zuyev, Professor, Department of Mathematical Sciences, Chalmers University, Sweden, short-term visitor

11. Yoon Sik Cho, PhD student, Information Sciences Institute
12. Ming Ji, PhD student, Information Sciences Institute
13. Greg Ver Steeg, Post-doctoral Associate, Information Sciences Institute
14. Armen Allahverdyan, Visiting Scholar, Information Sciences Institute
15. Clayton Morrison, Research Scientist, Information Sciences Institute
16. Aleksey Polunchenko, Post-doctoral Associate, Department of Mathematics
17. Vasanthan Raghavan, Post-doctoral Associate, Department of Mathematics
18. Georgios Fellouris, Post-doctoral Associate, Department of Mathematics
19. Greg Sokolov, PhD student, Department of Mathematics
20. Thang Ba Dinh, PhD student, Computer Science
21. Jan Prokaj, PhD student, Computer Science

University of California at Los Angeles (UCLA):

1. Andrea Bertozzi, Professor of Mathematics and Director of Applied Mathematics
2. Tony Chan, Professor of Mathematics
3. P. Jeffrey Brantingham, Associate Professor of Anthropology
4. Maria D’Orsogna (consultant), Asst. Prof. of Mathematics, Cal State Northridge
5. J. M. Morel (visitor), Professor, ENS Cachan, Paris
6. Stanley Osher (collaborator), Professor of Mathematics and Director of Special Projects IPAM
7. Lincoln Chayes (Collaborator), Professor of Mathematics, UCLA
8. George Tita (Collaborator), Professor of Criminology, Law, and Society, UCLA
9. Martin Short, CAM Assistant Professor in Mathematics
10. Tim Lucas (visitor), Assistant Professor, Pepperdine University
11. George Mohler, CAM Assistant Professor of Mathematics
12. Alethea Barbaro, Adjunct Assistant Professor of Mathematics
13. Todd Wittman, PIC Instructor
14. Berta Sandberg, Postdoctoral Scholar, Department of Mathematics
15. Virginia Pasour, Postdoctoral Scholar, Department of Mathematics
16. Zhipu Jin, Postdoctoral Scholar, Department of Mathematics
17. Alexander Chen, PhD student, Department of Mathematics
18. Nancy Rodriguez, PhD student, Department of Mathematics
19. Paul Jones, Ph D student, Department of Mathematics
20. Matthew Keegan, Ph D student, Department of Mathematics
21. Laura Smith, Ph D student, Department of Mathematics
22. Erik Lewis, Ph D student, Department of Mathematics
23. Rachel Hegemann (née Danson), P D student, Department of Mathematics
24. Wangyi Liu, Ph D student, Department of Mathematics
25. Yasser Taima, Ph D student, Department of Electrical Engineering
26. Kevin Shen, Masters student, Department of Electrical Engineering
27. David Hermina, Masters student, Department of Electrical Engineering
28. James von Brecht, Masters student, Department of Mathematics
29. Marina Masaki, Undergraduate REU student, Department of Mathematics UC Irvine
30. Kym Louie, Undergraduate REU student, Department of Mathematics Harvey Mudd College
31. Mark Allenby, Undergraduate REU student, Department of Mathematics, Pepperdine
32. Mike Egesdal, Undergraduate REU student, Departments of Mathematics and Economics, UCLA
33. Chris Fathauer, Undergraduate REU student, Department of Mathematics, Harvey Mudd College
34. Jeremy Neumann Undergraduate REU student, Departments of Mathematics and Physics, UCLA

35. Benjamin Irvine, Undergraduate REU student, Department of Mathematics, UCLA
36. Max Gonzales, Undergraduate REU student, Harvey Mudd College
37. Edwin Huang, Undergraduate REU student, Department of Electrical Engineering, UCLA
38. Abhijeet Joshi, Undergraduate REU student, Department of Electrical Engineering, UCLA
39. Kevin Leung, Undergraduate REU student, Department of Electrical Engineering, UCLA
40. Vlad Voroninsky, Undergraduate REU student, Department of Mathematics, UCLA
41. Trevor Ashley, Undergraduate REU student, Harvey Mudd College

University of Illinois at Urbana-Champaign (Illinois):

1. Venugopal Veeravalli, Professor, Illinois-ECE&CSL
2. George Atia, Postdoctoral Fellow, Illinois-ECE&CSL
3. Vasanthan Raghavan, Postdoctoral Fellow, Illinois-ECE&CSL
4. Jason Fuemmeler, PhD student, Illinois-ECE&CSL
5. Taposh Banerjee, PhD student, Illinois-ECE&CSL

Colorado State University at Fort Collins (Colorado):

1. Christos Papadopoulos, Associate Professor, Computer Science
2. Steven DiBenedetto, Graduate student in Computer Science
3. Kaustubh Gadkari, Graduate student in Computer Science
4. Mengran Hu, Graduate student in Computer Science
5. Zhang Han, Graduate student in Computer Science
6. Andrea Steiner, Undergraduate in Computer Science

2. Required Numerical Data Related to This Grant

(1) List of papers submitted or published under ARO sponsorship during the course of the project

(a) Manuscripts submitted, but not published:

1. G. Fellouris and A.G. Tartakovsky, “Nearly Minimax Mixture Rules for One-sided Sequential Testing,” Submitted to *Sequential Analysis*, 2011.
2. A.S. Polunchenko, A.G. Tartakovsky, and N. Mukhopadhyay, “Nearly Optimal Change-Point Detection with an Application to Cybersecurity,” Submitted to *Sequential Analysis*, 2011.
3. A.S. Polunchenko and A.G. Tartakovsky, “State-of-the-Art in Sequential Change-Point Detection,” *Methodology and Computing in Applied Probability*, 2011 (accepted).
4. S. Lototsky, B. Rozovsky, and D. Selesi, “On Generalized Malliavin Calculus,” Submitted to *J. Stochastic Processes and Applications*.
5. G. Bal, S. Kaligotla, and B. Rozovsky, “On Homogenization of Stochastic PDEs of Elliptic Type,” Submitted to *J. Stochastic PDEs. Analysis and Computations*.
6. C.-Y. Lee, B. Rozovsky, “On Unbiased Perturbations of Stationary Navier-Stokes Equation,” in *Malliavin Calculus and Stochastic Analysis*, Editors, E. Nualart, F. Vienes, Springer (to appear in 2012).
7. Z. Zhang, B. Rozovsky, M. Tretyakov, and G. Karniadakis, “A multi-stage Wiener chaos expansion method for stochastic advection-diffusion-reaction,” Submitted to *SIAM Journal of Scientific Computing*.
8. P.J. Brantingham, G.E. Tita, M.B. Short and S. Reid, The Community Context of Gang Territorial Boundaries, submitted to Criminology.
9. M. Keegan, B. Sandberg, and T. Chan, ”A Logic Framework for the Detection and Tracking of Features fusing Motion and Intensity Channels”, Submitted to Journal of Visual Communication and Image Representation.
10. M. Keegan, B. Sandberg, and T. Chan, ”A Logic Framework for Multiphase Multichannel Image Segmentation”, Preprint.

11. G.O. Mohler and M.B. Short, “Geographic Profiling from Kinetic Models of Criminal Behavior,” Preprint, 2009.
12. T. Banerjee and V.V. Veeravalli, “Data-Efficient Bayesian Quickest Change Detection,” Submitted to *Sequential Analysis*, 2011.

(b) Papers published in peer-reviewed journals:

1. M. Pollak and A.G. Tartakovsky, “On the First Exit Time of a Nonnegative Markov Process Started at a Quasistationary Distribution,” *Journal of Applied Probability*, vol. 48, no. 2, pp. 589–595, 2011.
2. A.G. Tartakovsky, M. Pollak, and A.G. Polunchenko, “Third-Order Asymptotic Optimality of the Generalized Shiryaev–Roberts Changepoint Detection Procedures,” *Theoriya Veroyatnostej i ee Primeneniya*, vol. 56, no. 3, 2011.
3. A.S. Polunchenko and A.G. Tartakovsky, “On Optimality of the Shiryaev–Roberts Procedure for Detecting a Change in Distribution,” *The Annals of Statistics*, vol. 38, no. 6, pp. 3445–3457, 2010.
4. G.V. Moustakides, A.S. Polunchenko, and A.G. Tartakovsky, “A Numerical Approach to Performance Analysis of Quickest Change-Point Detection Procedures,” *Statistica Sinica*, vol. 21, no. 2, pp. 571–596, 2011.
5. A.G. Tartakovsky and G.V. Moustakides, “Discussion on “Quickest Detection Problems: Fifty Years Later” by Albert N. Shiryaev,” *Sequential Analysis*, vol. 29, no. 4, pp. 386–393, 2010.
6. G.V. Moustakides and A.G. Tartakovsky, “A Note on “The Optimal Stopping Time for Detecting Changes in Discrete Time Markov Processes” by Han and Tsung,” *Sequential Analysis*, vol. 29, no. 4, pp. 483–486, 2010.
7. R. Liptser, A. Novikov, and A.G. Tartakovsky, “Celebrating Albert Shiryaev’s 75th Anniversary,” *Sequential Analysis*, vol. 29, no. 2, pp. 107–111, 2010. Special issue: Celebrating the Seventy-Fifth Birthday of Albert N. Shiryaev, Guest Editor A.G. Tartakovsky.
8. R. Lipster and A.G. Tartakovsky, “From Disorder Detection to Optimal Stopping and Mathematical Finance,” *Sequential Analysis*, vol. 29, no. 2, pp. 112–124, 2010. Special issue: Celebrating the Seventy-Fifth Birthday of Albert N. Shiryaev, Guest Editor A.G. Tartakovsky.
9. A.G. Tartakovsky and G.V. Moustakides, “State-of-the-Art in Bayesian Changepoint Detection,” *Sequential Analysis*, vol. 29, no. 2, pp. 125–145, 2010. Special issue: Celebrating the Seventy-Fifth Birthday of Albert N. Shiryaev, Guest Editor A.G. Tartakovsky.
10. G.V. Moustakides, A.S. Polunchenko and A.G. Tartakovsky, “Numerical Comparison of CUSUM and Shiryaev–Roberts Procedures for Detecting Changes in Distributions,” *Communications in Statistics – Theory and Methods*: S. Zacks Festschrift, vol. 38, no. 16 & 17, pp. 3225–3239, 2009.
11. M. Pollak and A.G. Tartakovsky, “Asymptotic Exponentiality of the Distribution of First Exit Times for a Class of Markov Processes with Applications to Quickest Change Detection,” *Theory of Probability and Its Applications*, vol. 53, no. 3, pp. 430–442, 2009, SIAM.
12. M. Pollak and A.G. Tartakovsky, “Optimality Properties of the Shiryaev–Roberts Procedure,” *Statistica Sinica*, vol. 19, no. 4, pp. 1729–1739, 2009.
13. A.G. Tartakovsky, “Asymptotic Optimality in Bayesian Change-Point Detection Problems Under Global False Alarm Probability Constraint,” *Theory of Probability and Its Applications*, vol. 53, no. 3, pp. 443–466, 2009, SIAM.
14. A.G. Tartakovsky, “Discussion on “Optimal Sequential Surveillance for Finance, Public Health, and Other Areas” by Marianne Frisen,” *Sequential Analysis*, vol. 28, no. 3, pp. 365–371, 2009.
15. A.G. Tartakovsky, “Multidecision Quickest Changepoint Detection: Previous Achievements and Open Problems,” *Sequential Analysis*, vol. 27, pp. 201–231, 2008.
16. A.G. Tartakovsky, Discussion on “Is Average Run Length to False Alarm Always an Informative Criterion?” by Yajun Mei, *Sequential Analysis*, vol. 27, no. 4, pp. 396–405, 2008.
17. A.G. Tartakovsky and J. Brown, “Adaptive Spatial-Temporal Filtering Methods for Clutter Removal and Target Tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1522–1537, 2008.

18. A.G. Tartakovsky and V.V. Veeravalli, “Asymptotically Optimal Quickest Change Detection in Distributed Sensor Systems,” *Sequential Analysis*, vol. 27, no. 4, pp. 441–475, 2008.
19. M. Baron and A.G. Tartakovsky, “Asymptotic Bayesian Change-Point Detection Theory for General Continuous-Time Models,” *Sequential Analysis*, vol. 25, pp. 257–296, 2006. **(Special Invited Paper in Memory of Milton Sobel; Awarded by 2007 Abraham Wald Prize in Sequential Analysis)**.
20. A.G. Tartakovsky, B. Rozovskii, R. Blažek, and H. Kim, “Detection of Intrusions in Information Systems by Sequential Change-Point Methods,” *Statistical Methodology*, Vol. 3, Issue 3, pp. 252-340, 2006. **(Invited Discussion Paper)**.
21. A.G. Tartakovsky, B. Rozovskii, R. Blažek, and H. Kim, “A Novel Approach to Detection of Intrusions in Computer Networks Via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods,” *IEEE Transactions on Signal Processing*, vol. 54, pp. 3372–3382, 2006.
22. R. Mikulevicius and B. Rozovskii, “On Unbiased Stochastic Navier-Stokes Equation,” *J. Probab. Theory and Related Fields*, 2011.
23. J. Park, B. Rozovsky, and R. Sowers, “Efficient Nonlinear Filtering of a Singularly Perturbed Stochastic Hybrid System,” *London Math. Society J. of Computation and Mathematics*, 2011.
    DOI 10.1007/s00440-011-0384-1
24. C.-Y. Lee, B. Rozovsky, A stochastic finite element method for stochastic parabolic equations driven by purely spatial noise *Communications on Stochastic Analysis*, 4, (2010), no. 2, 271-297.
25. C.-Y. Lee, B. Rozovsky, and H. M. Zhow. Randomization of forcing in large systems of PDE for improvement of energy estimates. *SIAM J. Multiscale Modeling and Simulation*, 8 (2010), no. 4, 1419-1438.
26. S. Lototsky, B.L. Rozovskii, and X. Wan, “Elliptic equations of higher stochastic order,” *ESAIM: Math. Modeling and Numerical Anal.*, 44 (2010) no. 5, 1135-1153.
27. S. Lototsky and B.L. Rozovskii, “Stochastic Differential Equations Driven by Purely Spatial Noise,” *SIAM Journal on Mathematical Analysis*, Vol. 41, No. 4, pp. 1295-1322, 2009.
28. X. Wan, B. Rozovsky, and G. Karniadakis, “A new stochastic modeling methodology based on weighted Wiener chaos and Malliavin calculus,” *Proc. Natl. Acad. Sc. USA*, vol. 106, no. 34, pp. 14189-14104, 2009.
29. S. Lototsky and B.L. Rozovskii, “A Unified Approach to Stochastic Evolution Equations Using the Skorokhod Integral,” *Probability Theory and Its Applications*, 2008.
30. S. Lototsky and B.L. Rozovskii, “Stochastic Differential Equations Driven by Purely Spatial Noise,” *SIAM Journal on Mathematical Analysis*, Vol. 41, No. 4, 2009.
31. S. Lototsky and B.L. Rozovsky, “Stochastic Parabolic Equations of Full Second Order,” Book chapter in “Topics in Stochastic Analysis and Nonparametric Estimation” (Ed. P.- L. Chow et al.), pp. 199–210, The IMA Volumes in Mathematics and its Applications, Springer, 2007.
32. S. Lototsky and B.L. Rozovsky, “Wiener chaos solutions of linear stochastic evolution equations,” *Annals of Prob.* 34 (2006), no. 2, 638–662.
33. T. Hou and B.L. Rozovsky, H.-M. Luo, “Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics,” *J. Comput. Phys.* 216 (2006), no. 2, 687–706.
34. S. Lototsky and B.L. Rozovsky, “Stochastic differential equations: A Wiener chaos approach,” Book chapter in “From Stochastic Calculus to Mathematical Finance” (Ed. Y. Kabanov et al.). 433–506, Springer, Berlin, 2006
35. G. Da Prato, M. Roeckner, and B.L. Rozovsky, “Strong solutions of stochastic generalized porous media equations: Existence, uniqueness and ergodicity,” *Comm. Partial Dif. Eq.* 31 (2006), no. 1-3, 277–291.
36. B. L. Rozovskii and J. Cvitanic, “A filtering approach to tracking volatility from prices observed at random times,” *Annals of Applied Prob.*, 16 (2006), no. 3, 1633–1652.
37. J. Cvitanic, R. Liptser, and B.L. Rozovskii, “ Numerical estimation of volatility values from discretely observed diffusion data,” *J. Comp. Finance*, 9 (2006), no. 4,1-36

38. J. Cvitanic, B.L. Rozovskii, and I. Zalyapin, “Numerical Filtering of Volatility Values from discretely Observed Diffusion Data,” *J. Comp. Finance*, vol.9, no.4, pp. 1-36, 2006.
39. Sung Ha Kang, Berta Sandberg and Andy Yip, “A Regularized K-means and Multiphase Scale Segmentation”, *Inverse problems and imaging*, Volume 5, Issue 2, Pages 407-429, 2011.
40. A. Chen, T. Wittman, A. Tartakovsky, and A. Bertozzi, “Boundary Tracking Through Efficient Sampling,” *AMRX*, Volume 2011, Issue 2, pages 182-214, 2011.
41. George O. Mohler, Andrea L. Bertozzi, Thomas A. Goldstein, and Stanley J. Osher, “Fast TV Regularization for 2D Maximum Penalized Likelihood Estimation,” *Journal of Computational and Graphical Statistics*, June 1, 2011, 20(2): 479-491.
42. R. A. Hegemann, L. M. Smith, A. Barbaro, A. L. Bertozzi, S. Reid, and G. E. Tita, “Geographical influences of an emerging network of gang rivalries,” *Physica A*, Volume 390, Issues 21-22, 15 October 2011, Pages 3894-3914.
43. E. Lewis, G. Mohler, P. J. Brantingham, and A. L. Bertozzi, “Self-Exciting Point Process Models of Civilian Deaths in Iraq,” *Security Journal*, advanced online publication, doi:10.1057/sj.2011.21, 2011.
44. G.O. Mohler, M.B. Short, P.J. Brantingham, F.P. Schoenberg, and G.E. Tita, “Self-exciting point process modeling of crime,” J. Am. Stat. Assoc., 106 (2011).
45. M.B. Short, P.J. Brantingham, and M.R. D’Orsogna, “Cooperation and punishment in an adversarial game: How defectors pave the way to a peaceful society,” Phys. Rev. E 82 (2010).
46. Paul A. Jones, P. Jeffrey Brantingham, and Lincoln R. Chayes, “Statistical models of criminal behavior: the effects of law enforcement agencies,” Math. Modl. Meth. Appl. Sciences (M3AS), vol. 20, Suppl. 1, pp. 1397-1423.
47. Nancy Rodriguez and Andrea Bertozzi, “Local Existence and Uniqueness of Solutions to a PDE model for Criminal Behavior , M3AS, special issue on Mathematics and Complexity in Human and Life Sciences,” Vol. 20, Issue supp01, pp. 1425-1457, 2010.
48. M.B. Short, A.L. Bertozzi, and P.J. Brantingham, “Nonlinear Patterns in Urban Crime – Hotspots, Bifurcations, and Suppression,” *SIAM J. Appl. Dyn. Systems*, Volume 9, Issue 2, pp. 462-483 (2010)
49. M. B. Short, P. J. Brantingham, A. L. Bertozzi, G. E. Tita, “Dissipation and displacement of hotspots in reaction-diffusion models of crime,” *Proc. Nat. Acad. Sci.*, 107(9), March 2, 2010, pp. 3961-3965.
50. Laura M. Smith, Matthew S. Keegan, Todd Wittman, George O. Mohler and Andrea L. Bertozzi, “Improving Density Estimation by Incorporating Spatial Information,” *EURASIP J. on Advances in Signal Processing*, Article ID 265631, doi:10.1155/2010/265631, Volume 2010, 12 pages, special issue on Advanced Image Processing for Defense and Security Applications, 2010.
51. Mike Egesdal, Chris Fathauer, Kym Louie, and Jeremy Neumann, “Statistical and Stochastic Modeling of Gang Rivalries in Los Angeles,” *SIAM Undergraduate Research Online (SIURO)*, May 26, 2010.
52. M. B. Short, M. R. D’Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. Chayes, “A statistical model of criminal behavior,” *M3AS: Mathematical Models and Methods in Applied Sciences*, special issue on Traffic, Crowds, and Swarms, volume 18, Supp., pages 1249-1267, 2008.
53. G. Ver Steeg, A. Galstyan, and A. Allahverdyan “Statistical mechanics of semi-supervised clustering in sparse graphs,” *Journal of Statistical Mechanics: Theory and Experiment*, (08):P08009, 2011.
54. A. Allahverdyan, G. Ver Steeg, and A. Galstyan, “Community Detection with and without Prior Information,” *Europhysics Letters*, vol. 90, no. 1, 18002, 2010.
55. A. Galstyan, V. Musoyan, and P.R. Cohen, “Maximizing Influence Propagation in Networks with Community Structure,” *Phys. Rev. E* vol. 79, 056102, 2009.
56. A. Galstyan and P.R. Cohen, “Cascading Dynamics in Modular Networks,” *Phys. Rev. E*, vol. 75, 036109, 2007.

57. Y. Lin, Q. Yu, and G. Medioni, "Efficient Detection and Tracking of Moving Objects in Geo-coordinates," *Machine Vision and Applications*, April 2010.
58. Q. Yu and G. Medioni, "Multiple Target Tracking with Global Optimization using Markov Chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 2008.
59. G.K. Atia, V.V. Veeravalli and J.A. Fuemmeler, "Sensor Scheduling for Energy-Efficient Target Tracking in Sensor Networks," *IEEE Transactions on Signal Processing*, 59(10): 4923–4937, October 2011.
60. J.A. Fuemmeler, G.K. Atia and V.V. Veeravalli, "Sleep Control for Tracking in Sensor Networks," *IEEE Transactions on Signal Processing*, 59(9): 4354–4366, September 2011.
61. J. Fuemmeler and V.V. Veeravalli, "Energy Efficient Multi-Object Tracking in Sensor Networks," *IEEE Transactions on Signal Processing*, 58(7): 3742–3750, July 2010.
62. V. Raghavan and V.V. Veeravalli, "Quickest Change Detection of a Markov Process Across a Sensor Array," *IEEE Transactions on Information Theory*, 56(4): 1961–1981, April 2010.
63. J. Fuemmeler and V.V. Veeravalli, "Smart Sleeping Policies for Energy Efficient Tracking in Sensor Networks," *IEEE Transactions on Signal Processing*, 56(5): 2091–2102, May 2008.
64. X. He, C. Papadopoulos, J. Heidemann, U. Mitra, and U. Riaz, "Remote Detection of Bottleneck Links Using Spectral and Statistical Methods," *Computer Networks*, 53(3), pp. 279–298, February, 2009.
65. J. Mirkovic, P. Reiher, C. Papadopoulos, A. Hussain, M. Shepard, M. Berg, and R. Jung, "Testings a Collaborative DDoS Defense in a Red Team/Blue Team Exercise," *IEEE Transactions On Computers*, 57(7), July 2008.
66. K. Hwang, M. Cai, J. Pan, C. Papadopoulos, "WormShield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation," *IEEE Transactions on Dependable and Secure Computing*, 4(2), pp. 88–104, April 2007.
67. U. Mitra, J. Heidemann, A. Ortega, and C. Papadopoulos, "Detecting and Identifying Malware: A New Signal Processing Goal," *IEEE Signal Processing Magazine*, 23 (5), pp. 107-111, September 2006.

(c) Books and book chapters:

1. *Handbook on Nonlinear Filtering*, D. Crisan and B. Rozovsky Editors, Oxford University Press, 2011.
2. V.V. Veeravalli, "Fundamentals of Detection Theory," In *Mathematical Foundations for Signal Processing, Communications and Networking*, T. Chen, D. Rajan, and E. Serpedin (Eds.), Cambridge University Press, 2011.
3. J. Fuemmeler and V.V. Veeravalli, "Smart Sleeping Policies for Energy-Efficient Tracking in Sensor Networks," In *Networked Sensing Information and Control,* V. Saligrama (Ed.), Springer 2008.

(d) Papers published in non-peer-reviewed journals or in conference proceedings:

1. M. Pollak and A.G. Tartakovsky, "Monotone Properties of the First Exit Time of a Markov Process Started at a Quasi-stationary Distribution," *Markov and Semi-Markov Processes and Related Fields 2011*, Porto Carras Grand Resort, Chalkidiki, Greece, 20–23 September, 2011.
2. A.G. Tartakovsky and M. Pollak, "Nearly Minimax Changepoint Detection Procedures," *Proceedings of the IEEE International Symposium on Information Theory*, St. Petersburg, Russia, July 31 – August 5, 2011.
3. A.G. Tartakovsky and A.S. Polunchenko, "Minimax Optimality the Shiryaev–Roberts Procedure," *Proceedings of the 5th International Workshop in Applied Probability*, Universidad Carlos III de Madrid, Colmenarejo Campus, Spain, 5 – 8 July 2010 (Invited).
4. A.G. Tartakovsky, A.S. Polunchenko, and G.V. Moustakides, "Design and Comparison of Shiryaev–Roberts- and CUSUM-Type Change-Point Detection Procedures," *Proceedings of the 2nd International Workshop in Sequential Methodologies*, University of Technology of Troyes, Troyes, France, 15 – 17 June 2009 (Invited).

5. A.G. Tartakovsky, A.P. Brown, and J. Brown, “Enhanced Algorithms for EO/IR Electronic Stabilization, Clutter Suppression, and Track-Before-Detect for Multiple Low Observable Targets,” *The 10th Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, Hawaii, September 2009.
6. A.G. Tartakovsky and A.S. Polunchenko, “Quickest Changepoint Detection in Distributed Multisensor Systems under Unknown Parameters,” *Proceedings of the 11th International Conference on Information Fusion*, Hyatt Regency Hotel, Cologne, Germany, 2008, pp. 878-885 (Invited).
7. A.G. Tartakovsky, M. Pollak, and A.S. Polunchenko, “Asymptotic Exponentiality of First Exit Times for Recurrent Markov Processes and Applications to Changepoint Detection,” *Proceedings of the 2008 International Workshop on Applied Probability*, Compiègne, France, 7-10 July 2008 (Invited).
8. A.G. Tartakovsky and H. Kim, “Performance of Certain Decentralized Distributed Change Detection Procedures,” *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, 2006, CD ISBN 0-9721844-6-5, IEEE Catalog No. 06EX1311C (Invited).
9. A.G. Tartakovsky and A.S. Polunchenko, “Decentralized Quickest Change Detection in Distributed Sensor Systems With Applications to Information Assurance and Counter Terrorism,” *Proceedings of the 13th Annual Army Conference on Applied Statistics*, Rice University, Houston, TX, 17-19 October 2007 (Invited).
10. P.J. Brantingham and M.B. Short, “Crime Emergence,” In *When Crime Appears: The Role of Emergence*, edited by J.M. McGloin, C. Sullivan and L.W. Kennedy. New York: Routledge (in press).
11. M. Gonzalez, X. Huang, B. Irvine, D. S. Hermina Martinez, C. H. Hsieh, Y. R. Huang, M. B. Short, and A. L. Bertozzi, “A Third Generation Micro-vehicle Testbed for Cooperative Control and Sensing Strategies,” *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 14-20, 2011.
12. Wangyi Liu, Martin B. Short, Yasser E. Taima, and Andrea L. Bertozzi, “Multiscale Collaborative Searching Through Swarming,” *Proceedings of the 7th International Conference on Informatics in Control, Automation, and Robotics (ICINCO)*, Portugal, June 2010.
13. J.H. von Brecht, S. Thiruvenkadam and T.F. Chan, “Occlusion Tracking Using Logic Models,” *9th IASTED Conf. on Signal and Image Processing*, 2007.
14. A. Joshi, T. Ashley, Y. Huang, and A. L. Bertozzi, “Experimental validation of cooperative environmental boundary tracking with on-board sensors,” *American Control Conference*, St. Louis, MO, June 2009, pp. 2630-2635.
15. Z. Jin and A. L. Bertozzi, “Environmental Boundary Tracking and Estimation using Multiple Autonomous Vehicles,” *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, 2007, pp. 4918-4923.
16. Y. Landa, D. Galkowski, Y. R. Huang, A. Joshi, C. Lee, K. K. Leung, G. Malla, J. Treanor, V. Voroninski, A. L. Bertozzi, and R. Tsai, “Robotic path planning and visibility with limited sensor data,” *Proceedings of the 2007 American Control Conference*.
17. Kevin K. Leung, Chung H. Hsieh, Yuan R. Huang, Abhijeet Joshi, Vlad Voroninski, and Andrea L. Bertozzi, “A second generation micro-vehicle testbed for cooperative control and sensing strategies,” *Proceedings of the 2007 American Control Conference*, pp. 1900-1907.
18. Y.-L. Chuang, Y. R. Huang, M. R. D’Orsogna, and A. L. Bertozzi, “Multi-vehicle flocking: scalability of cooperative control algorithms using pairwise potentials,” *IEEE International Conference on Robotics and Automation*, 2007, pp. 2292-2299.
19. A. Allahverdyan and A. Galstyan, “Comparative Analysis of Viterbi Training and ML Estimation for HMMs,” In *Neural Information Processing Systems (NIPS)*, 2011.
20. Y.S.Cho, G. Ver Steeg, and A. Galstyan, “Co-evolution of Selection and Influence in Social Networks,” In *Proc. of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11)*, 2011.
21. A. Allahverdyan and A. Galstyan, “On Maximum a Posteriori Estimation of Hidden Markov Processes,” In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*, Montreal, Canada, 2009.

22. A. Galstyan and P.R. Cohen, “Relational Classification Through Three-State Epidemic Dynamics,” In *Proceedings of the 9th International Conference on Information Fusion (FUSION’06)*, special session on Making Histories, Florence, Italy, 2006.
23. A. Galstyan and P.R. Cohen, “Empirical Comparison of “Hard” and “Soft” Label Propagation for Relational Classification,” In *Proceedings of the International Conference on Inductive Logic Programming (ILP–07)*, Corvallis, OR, 2007.
24. T. B. Dinh, N. Vo, and G. Medioni, “Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments,”*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs CO, Jun 20-25 2011.
25. T. B. Dinh, N. Vo, and G. Medioni, “High Resolution Face Sequences from a PTZ Network Camera,”*IEEE Conference on Automatic Face and Gesture Recognition (FG)*, Santa Barbara CA, Mar 21-25 2011.
26. T. B. Dinh and G. Medioni, “Co-training Framework of Generative and Discriminative Trackers with Partial Occlusion Handling,”*IEEE Workshop on Motion and Video Computing (WMVC)*, Kona Hawaii, Jan 5-7 2011.
27. Q. Yu and G. Medioni, “Motion Pattern Interpretation and Detection for Tracking Moving Vehicles in Airborne Videos,” *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), Miami FL, Jun 20-25 2009.
28. T. Dinh, Q. Yu, and G. Medioni, “Real Time Tracking Using an Active Pan-Tilt-Zoom Network Camera,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis MO, Oct 11-15 2009.
29. Q. Yu, T. Dinh, and G. Medioni, “Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers,” *European Conference on Computer Vision (ECCV)*, Marseille, France, Oct 12-18, 2008.
30. Q. Yu, G. Medioni, “Integrated Detection and Tracking for Multiple Moving Objects using Data-Driven MCMC Data Association,”*IEEE Workshop on Motion and Video Computing (WMVC)*, Copper Mountain, CO, Jan 08-09 2008.
31. Q. Yu, G. Medioni. “A GPU-based implementation of Motion Detection from a Moving Platform,”*IEEE Workshop on Computer Vision on GPU (CVGPU)*, Anchorage AK, Jun 23-28 2008.
32. Q. Yu, G. Medioni, and I. Cohen, “Multiple Target Tracking Using Spatio-Temporal Monte Carlo Markov Chain Data Association,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* , Minneapolis MN, Jun 18-23 2007.
33. Q. Yu and G. Medioni, “Map-Enhanced Detection and Tracking from a Moving Platform with Local and Global Data Association,” *IEEE Workshop on Motion and Video Computing (WMVC)*, pp. 3-10, Austin TX, Feb 23-24 2007.
34. Y. Lin, Q. Yu, and G. Medioni, “Map-Enhanced UAV Image Sequence Registration,” *IEEE Workshop on Applications of Computer Vision (WACV)*, pp.15-20, Austin TX, Feb 23-24 2007.
35. T. Banerjee and V.V. Veeravalli, “Bayesian Quickest Change Detection Under Energy Constraints,” In *Proc. ITA workshop, UCSD*, San Diego, CA, February 2011 (Invited).
36. G. Atia and V.V. Veeravalli, “Sensor management for energy-efficient tracking in cluttered environments,” In *Proc. ITA workshop, UCSD*, San Diego, CA, February 2011 (Invited).
37. G.K. Atia, V.V. Veeravalli and J.A. Fuemmeler, “Sensor scheduling for energy-efficient target tracking in sensor networks,” In *Proc. IEEE Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, November 2010.
38. K. Premkumar, A. Kumar and V.V. Veeravalli, “Bayesian Quickest Transient Change Detection,” In *Proc. International Workshop on Applied Probability*, Madrid, Spain, July 2010 (Invited).
39. V. Ragahavan and V.V. Veeravalli, “Bayesian quickest change process detection,” In *Proc. IEEE ISIT*, Seoul, South Korea, August 2009.

40. V. Raghavan and V.V. Veeravalli, “Quickest Detection of a Change Process Across a Sensor Array,” In *Proc. IEEE Fusion*, Cologne, Germany, July 2008 (Invited).
41. J. Fuemmeler and V.V. Veeravalli, “Sensor Scheduling for Effective and Energy Efficient Tracking in Sensor Networks,” In *Proc. IEEE CDC,* New Orleans, LA, December 2007 (Invited).
42. V.V. Veeravalli and J. Fuemmeler, “Joint Optimization of Smart Sleeping and Cooperative Localization Strategies for Energy-Efficient Tracking in Sensor Networks,” In *Proc. 56th Session of the International Statistical Institute (ISI)*, Lisbon, Portugal, August 2007 (Invited).
43. J. Fuemmeler and V.V. Veeravalli, “Smart Sleeping Strategies for Localization and Tracking in Sensor Networks,” In *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, November 2006 (Invited).
44. V.V. Veeravalli and J. Fuemmeler, “Efficient Tracking in a Network of Sleepy Sensors,” In *Proc. IEEE ICASSP,* Toulouse, France, May 2006 (Invited).
45. G. Bartlett, J. Heidemann, and C. Papadopoulos, “Understanding Passive and Active Service Discovery,” In *Proceedings of the ACM Internet Measurement Conference*, San Diego, California, USA, October, 2007.
46. A. Hussain, J. Heidemann, and C. Papadopoulos, “Identification of Repeated Denial of Service Attacks,” In *Proceedings of the IEEE Infocom*, Barcelona, Spain, April, 2006.
47. G. Bartlett, J. Heidemann, and C. Papadopoulos, “Inherent Behaviors for On-line Detection of Peer-to-Peer File Sharing,” In *Proceedings of the 10th IEEE Global Internet*, Anchorage, Alaska, USA, May 2007.
48. J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, “Census and Survey of the Visible Internet,” In *Proceedings of the ACM Internet Measurement Conference*, p. 169-182. Vouliagmeni, Greece, October 2008.
49. C. Wilcox, C. Papadopoulos, and J. Heidemann, “Correlating Spam Activity with IP Address Characteristics”, In *Proceedings of the IEEE Global Internet Symposium*, San Diego, California, USA, March 2010.
50. G. Bartlett, J. Heidemann, and C. Papadopoulos, “Low-Rate, Flow-Level Periodicity Detection,” In *Proceedings of the 14th IEEE Global Internet Symposium*, Shanghai, China, April 2011.
51. S. DiBenedetto, K. Gadkari, N. Diel, A. Steiner, D. Massey, and C. Papadopoulos, “Fingerprinting Custom Botnet Protocol Stacks,” *Proceedings of the 6th Workshop on Secure Network Protocols (NPSec)*, Japan, Kyoto, October 2010.

(e) Papers presented at meetings, but not published in conference proceedings:

1. A.G. Tartakovsky, “Spatiotemporal Image Processing with Applications to Remote Sensing,” Department of Statistics and Department of Computer Sciences, University of Chicago, September, 2011 (Invited).
2. A.G. Tartakovsky, “Sequential Changepoint Detection: State of the Art,” Department of Statistics, University of Illinois at Urbana-Champaign, October, 2011 (Invited).
3. A.G. Tartakovsky, “Spatiotemporal Image Processing with Applications to Remote Sensing,” Department of Electrical and Computer Engineering and Coordinated Science Lab, University of Illinois at Urbana-Champaign, October, 2011 (Invited).
4. A.S. Polunchenko, A.G. Tartakovsky, and N. Mukhopadhyay, “Nearly Optimal Change-Point Detection with An Application to Cybersecurity,” *The 3rd International Workshop in Sequential Methodologies*, Stanford University, 14–16 June 2011.
5. A.G. Tartakovsky and A.S. Polunchenko, “Optimality of the Shiryaev–Roberts Procedure for Detecting Changes in Distributions,” *The 73rd Annual Meeting of the Institute of Mathematical Statistics*, Gothenburg, Sweden, 9–13 August 2010.
6. A.P. Brown and A.G. Tartakovsky, “Spatiotemporal Clutter Rejection and Track-Before-Detect Methods for Tracking Small Dim Objects,” *32nd Review of Atmospheric Transmission Models Meeting*, Lexington, Massachusetts, 14-15 June 2010 (Invited).

7. A.G. Tartakovsky, "Spatial-Temporal Image Processing Techniques and Applications to Remote Sensing," Department of Mathematics, Stanford University, 2009 (Invited).
8. A.G. Tartakovsky, "Quickest Changepoint Detection: Recent Advances and Open Problems," *NSF Sponsored Workshop in Honor of Professor A.V. Balakrishnan*, January 30, 2009 (Invited).
9. A.G. Tartakovsky, "Efficient Numerical Methods for Optimization and Performance Evaluation of Changepoint Detection Procedures," Department of Probability, Moscow State University, Moscow, Russia, March 25, 2009 (Invited).
10. A.G. Tartakovsky, "Adaptive Spatial-Temporal Image Processing Techniques and Applications to Clutter Rejection in Remote Sensing," *Workshop "Spatiotemporal Image Processing and Visual Surveillance"*, University of Southern California, 2008 (Invited).
11. A.G. Tartakovsky, "Exact Optimality of the Shiryaev–Roberts Procedure for Detecting Changes in Distributions," Department of Mathematical Sciences, University of Technology, Sydney, Australia, November 27, 2008 (Invited).
12. A.G. Tartakovsky, "Detection and Classification in Distributed Multisensor Systems with Applications to Network Security," *Workshop "Sensor Networks and Future Internet Security"*, University of Southern California, May 23, 2007 (Invited).
13. A.G. Tartakovsky, "Asymptotic Optimality in Sequential Quickest Change-Point Detection: Theory and Applications," Princeton, September 25, 2007 (Invited).
14. A.G. Tartakovsky, "Quickest Change-Point Detection: Previous Achievements and Open Problems," *First International Workshop on Sequential Methodologies,* Auburn, AL, 22-25 July 2007 (Invited).
15. A.G. Tartakovsky, "Asymptotic Optimality in Sequential Hypothesis Testing and Quickest Change-Point Detection for General Continuous-Time Stochastic Processes," *Workshop on Inverse Problems in Stochastic Differential Equations*, University of Southern California, Los Angeles, CA, 22-26 May, 2007 (Invited).
16. A.G. Tartakovsky, "An Asymptotically Optimal Change Detection Strategy Under Nontraditional Global False Alarm Probability Constraint," *The 2007 Taipei International Statistical Symposium and ICSA International Conference (Session: Change-Point Analysis and Applications)*, Taipei, Taiwan, 24-28 June, 2007 (Invited).
17. V.V. Veeravalli and A.G. Tartakovsky, "Quickest Change Detection in Sensor Networks," *First International Workshop on Sequential Methodologies,* Auburn, AL, July 2007 (Invited).
18. B.L. Rozovsky, "Generalized Malliavin calculus and Stochastic PDEs," Columbia University, Minerva Foundation Lectures, December 2010.
19. B.L. Rozovsky, "Stochastic Fluid Dynamics," NSF Institute for Pure and Applied Mathematics, Invited lecture, January 2011.
20. B.L. Rozovsky, "Stochastic Fluids and Malliavin Calculus," *Conference on Malliavin Calculus and Stochastic Analysis*, University of Kansas, Invited talk, March 2011.
21. B.L. Rozovsky, "On Unbiased Stochastic Navier-Stokes Equation," *Workshop on SPDEs, Archimedes Center for Modeling, Analysis, and Computations*, Heraklion, Greece, Invited lecture, June 2011.
22. B.L. Rozovsky, "Recent Advances in Nonlinear Filtering," Imperial College, London. Invited lecture, June 2011.
23. B.L. Rozovsky, "Stochastic Fluid Dynamics and Malliavin Calculus," Oxford University, Invited lecture, 2011.
24. B.L. Rozovsky, "Uncertainty Quantification and Nonlinear Filtering," ICIAM 2011, Vancouver, Canada, 2011.
25. B.L. Rozovsky, "On Unbiased Stochastic Navier-Stokes Equation," ICIAM 2011, Vancouver, Canada, 2011.
26. B.L. Rozovskii, Invited Talk, SIAM conference on Computational Science and Engineering, Maiami, 2009.
27. B.L. Rozovskii, Invited Talk, 7th ISAAC Congress, London, 2009.

28. B.L. Rozovskii, Invited Talk, International Conference on Spectral and High Order Methods, 2009, Trondheim, Norway.
29. Alethea Barbaro, Agent-based Complex Systems Workshop at IPAM: Organized and spoke October 14, 2009 1 hour, Title: ”Agent-based modeling for animal migration and gang behavior”.
30. Alethea Barbaro, American Soc. Criminology Meeting Philadelphia, joint presentation with Shannon Reid, Nov. 6, 2009, ”Agent-based simulations: modeling gang violence in Hollenbeck”
31. Alethea Barbaro, UCSB Hypatian Seminar, Nov 30, 2010, ”Agent-based modeling of complex systems, and how to claim your mathematical territory after your doctorate”
32. Alethea Barbaro, IPAM’s Optimal Transport Reunion Workshop at Lake Arrowhead (invited talk) December 10, 2009, 20 minutes, Title: ”On limits of a discrete time interacting particle system”
33. Alethea Barbaro, 2nd Annual Southern California Women in Math Symposium February 20, 2010, 30 minutes, Title: ”Agent-based models of social dynamics”.
34. Alethea Barbaro, 2010 Mathematics Festival at UCLA (2 sessions) February 13, 2010, Two sessions, each 50 minutes Title: ”Modeling the Real World: Using Math to Study Migration, Territoriality, and Social Networks”.
35. Alethea Barbaro, Invited Talk at USC for the Women in Math Seminar, March 12, 2010, 1 hour, Title: ”Simulating social dynamics with interacting particle models”.
36. Alethea Barbaro, Invited Seminar Talk at Redlands, March 31, 2010, 1 hour, Title: ”Simulating Social Dynamics with Interacting Particle Models”.
37. Alethea Barbaro, Talk at SIAM’s DSPDEs conference in Barcelona, Spain (Organized Mini-symposium and spoke), Mini-symposium title: Particle and mean field models for flocking and swarming, co-chair Massimo Fornasier, June 1, 2010, 30 minutes, Talk Title: ”Interacting particle models for Social Dynamics”.
38. Alethea Barbaro, Workshop: Modeling Complex Dynamics in Biological Systems, Universite Paul Sabatier, Toulouse, France (invited talk) June 9, 2010 1 hour Title: ”Interacting particle models for animal social dynamics”.
39. Alethea Barbaro, Workshop: Mathematics of Complex Systems, Universite Paul Sabatier, Toulouse, France (invited talk) June 10, 2010, 45 minutes, Title: ”Agent-based models for gang dynamics”.
40. Alethea Barbaro, Kinetic and Mean-field models in the Socio-Economic Sciences: workshop at ICMS, Edinburgh, Scotland (25 minute invited talk), July 31, 2009, Title: ”Fish migration, interacting particles, and scaling laws”.
41. Andrea Bertozzi, Women in Mathematics Seminar, Univ. of Wisconsin, Madison, WI, October 7, 2009.
42. Andrea Bertozzi, Invited talk and co-organizer, IPAM workshop on ”Agent Based Complex Systems”, October 14, 2009.
43. Andrea Bertozzi, Invited address, Southern California-Nevada MAA Section Meeting, October 17, 2009.
44. Andrea Bertozzi, Invited talk, Workshop on Self-Organization and Multi-Scale Mathematical Modeling of Active Biological Systems, Statistical and Applied Mathematical Sciences Institute, Durham, NC, October 27 2009.
45. Andrea Bertozzi, Invited talk, Army Research Office, Durham, NC October 28, 2009.
46. Andrea Bertozzi, Invited talk UBC Vancouver, PIMS mini-symposium in PDE, one hour talk, November 13, 2009.
47. Andrea Bertozzi, Invited talk on “A Variational Approach to Hyperspectral Image Fusion”, Minisymposium on Variational Methods in Image Processing and Interface Problems, Maria Westdickenberg and Sung Ha Kang, Organizers, SIAM Conference on Analysis of Partial Differential Equations, Miami, December 7, 2009.

48. Andrea Bertozzi, Invited talk on ”Mathematical Models for Urban Crime” Minisymposium on “Nonlinear Stochastic PDEs and Applicationss to Complex Systems”, Hakima Bessaih and Bjorn Birnir, Organizers, SIAM Conference on Analaysis of Partial Differential Equations, Miami, December 8, 2009.
49. Andrea Bertozzi, Invited talk in SIAM Minisymposium on New Trends in Mathematical Methods in Imaging Science, Rick Chartrand, Stacey Levine, Jennifer Mueller, and Luminita Vese organizers, Joint Math Meetings, San Francisco, Sat Jan 16, 2010
50. Andrea Bertozzi, Invited talk, China Lake Distinguished Speakers Colloquium Series, China Lake Naval Air Warfare Center, Ridgecrest, CA, Feb 2, 2010
51. Andrea Bertozzi, Invited talk Rand Corp. Santa Monica, Feb 11, 2010
52. Andrea Bertozzi, Invited talk, Session on ”Traffic, Crowds and Society”, AAAS Annual Meeting, San Diego, February 20, 2010
53. Andrea Bertozzi, Invited talk, Imperial College London, Institute for Mathematical Sciences, invited talk in three part session on Geometric Mechanics, Darryl Holm host, March 8, 2010
54. Andrea Bertozz,i Fluid Mechanics Seminar, DAMTP, Univ. of Cambridge, UK, March 5, 2010
55. Andrea Bertozzi, Brown University, Mathematics Department, Distinguished Lecture Series, three one hour lectures, March 11-12, 2010
56. Andrea Bertozzi, Brown University, Mathematics Department, faculty speaker, Symposium for Undergraduates in the Mathematical Sciences, 45 minute talk, March 13, 2010
57. Andrea Bertozzi, Invited talk, Minisymposium on Advanced Frameworks for Restructuring High Dimensional Datasets, SIAM Conf. on Imaging Science, Chicago, IL April 13, 2010, Edward H. Bosch Organizer.
58. Andrea Bertozzi,Invited talk, Plenary talk, Joint SIAM/RSME-SCM-SEMA Meeting on Emerging Topics in Dynamical Systems and Partial Differential Equations DSPDEs’10 June 1, 2010, Barcelona, Spain
59. Andrea Bertozzi, Invited talk 2010 DTRA/NSF Algorithm workshop, talk on “Undergraduate Research Training in Defense Applications”, June 22, 2010, Chapel Hill, NC
60. Andrea Bertozzi, Invited talk 2010 DTRA/NSF Algorithm workshop, talk on “Imaging of multispectral and hyperspectral data”, June 23, 2010, Chapel Hill, NC
61. Andrea Bertozzi, invited talk talk in workshop Fluid Dynamics Analysis and Numerics, a conference in honor of Tom Beale’s 60th Birthday, Duke Univ., Durham, NC June 28, 2010
62. Andrea Bertozzi, invited talk at Park City Mathematics Institute, Program on Imaging Sciences, Park City UT, July 5, 2010
63. Andrea Bertozzi, Graduate School of Engineering and Applied Sciences, Distinguished Lecture, Naval Postgraduate School, Sept. 2, 2010.
64. Andrea Bertozzi, London Taught Course Centre 8 hour intensive course on Mathematics of Crime, Univ. College London, Sept. 9-10, 2010.
65. Andrea Bertozzi, Department of Applied Mathematics and Statistics Johns Hopkins University, Colloquium Sept. 16, 2010.
66. Andrea Bertozzi, Allman Family Public Lecture, Southern Methodist University, Mathematics in the Real World, Sept. 23, 2010.
67. Andrea Bertozzi, Invited talk, IPAM workshop on Machine Reasoning: Mission Focused Actions/Reactions Based on System Integration of Information Derived from Complex Real-World Data, Oct 19, 2010.
68. Andrea Bertozzi, Distinguished Lecture, Department of Mathematics, Simon Fraser Univ., Oct. 29, 2010.
69. Andrea Bertozzi, Invited talk, 9th Annual Image Fusion Workshop, Institute for Defense and Government Advancement, Tyson’s Corner, VA, November 16, 2010.
70. Andrea Bertozzi, Invited talk, RCIM Symposium Mathematical Aspects of Image Processing and Computer Vision 2010 Sapporo, Japan, November 26, 2010.

71. Andrea Bertozzi, Invited talk, NSF workshop on New Directions in Dynamical Systems Inspired by Biological, Energy, Environmental, and Information Sciences, Atlanta, GA, Jan 4, 2011
72. Andrea Bertozzi, Invited talk, Dynamics Days, Chapel Hill, NC, Jan 5, 2011
73. Andrea Bertozzi, AMS Invited Address, Joint Mathematics Meetings, New Orleans, LA, Jan 7, 2011
74. Andrea Bertozzi, Invited Talk (one hour), 2011 annual meeting of the Australian and New Zealand Industrial and Applied Mathematics division of the Australian Mathemtaical Society. ANZIAM 2011 in Glenelg, Australia, Feb. 1, 2011
75. Andrea Bertozzi, PIMS Applied Mathematics Seminar, University of Saskatchewan, Saskatoon, March 14, 2011
76. Andrea Bertozzi, Seminar, Ecole Normal Superieur de Cachan, Centre de Mathematiques et de leurs Applications, March 17, 2011
77. Andrea Bertozzi, Groupe de travail - Mathematiques de la decision, Seminar, Univ. of Toulouse, March 24, 2011
78. Andrea Bertozzi, Colloquium de L'Institut de Mathematiques de Toulouse, March 25, 2011
79. Andrea Bertozzi, Mathematics Colloquium Univ. of Warwick, UK, June 3, 2011, ”Mathematics of Crime”
80. Andrea Bertozzi, Nonlinear Diffusion: Applications, Analysis and Computation conference to celebrate the 60th Birthday of Charlie Elliot, Univ. Warwick, June 6-8, 2011, invited 45 minute talk.
81. Andrea Bertozzi, 7th East Asian SIAM meeting Waseda University Kitakyushu Campus, Japan Keynote Talk, June 29, 2011
82. Andrea Bertozzi, Invited talk, Minisymposium on Modern Methods and Applications of the Calculus of Variations: Image Processing , July 20, 2011, International Congress on Industrial and Applied Mathematics, Vancouver BC
83. Andrea Bertozzi, Invited talk, Duke Workshop on Sensing and Analysis of High-Dimensional Data (SAHD), July 26, 2011
84. Andrea Bertozzi, Plenary talk, AWM 40th Anniversary Conference, ICERM, Brown University, September 18, 2011
85. Andrea Bertozzi, Applied Mathematics Seminar, Mathematics of Crime, Harvard University, September 19, 2011
86. Andrea Bertozzi, Widely Applied Mathematics Seminar, Swarming by Nature and by Design, Harvard University, September 20, 2011
87. P. Jeffrey Brantingham, Repeats and Reprisals: The Dynamics of Burglary and Rival Gang Violence in Los Angeles. Invited lecture at the Workshop on Modeling and Analysis of Security, January 4 7, 2010, University of Chile, 2010.
88. P. Jeffrey Brantingham, “Agent-based and continuum models of crime pattern formation,” Invited lecture presented at the Agent-based Complex Systems workshop, Institute of Pure and Applied Mathematics, UCLA, October 12-14, 2009.
89. P. Jeffrey Brantingham, “Why seeking to reduce gang rivalries might increase gang violence,” UC Irvine Criminology, Law and Society, April 13, 2011.
90. P. Jeffrey Brantingham, “The Mathematical Ecology of Criminal Street Gangs,” UCLA Marschak Colloquium, April 8, 2011.
91. P. Jeffrey Brantingham, Stochastic Models of Crime with Practical Implications for Policing. Workshop on Geospatial Abduction Problems, University of Maryland, March 3-4, 2011.
92. P. Jeffrey Brantingham, “University-Agency Collaboration in Predictive Policing,” 10th Anniversary Celebration of the Institute for Canadian Urban Research, Simon Fraser University, February 3, 2011.
93. Maria D'Orsogna, invited talk, Kinetic and mean-field models in the socio-economic sciences, Edinburgh, Scotland, July 2009

94. Erik Lewis, poster presentation at the Agent-Based Complex Systems workshop at IPAM, October 12-14, 2009. “Comparing Gang Rivalries and Civilian Deaths in Iraq Using Self-Exciting Point Processes.”
95. George Mohler, “Agent-based, Bayesian Geographic Profiling” Workshop on Analysis and Modeling of Security,” Jan 4-7, 2010, Santiago Chile, 40 minute talk.
96. George Mohler, “Crime as a Self-Exciting Point Process: An innovative approach in crime prediction,” American Society of Criminology annual meeting, Nov 4-7, 2009, 20 minute talk.
97. Todd Wittman, Contributed talk. 25 minutes. “Image Processing in the UCLA REU Program.” *International Conference on Technology in Collegiate Mathematics*. Chicago, IL. March 2010.
98. Todd Wittman, Contributed talk. 20 minutes. ”Problems in Geospatial Image Processing.” Center for Nonlinear Analysis Summer School on Image Processing and PDEs. Pittsburgh, PA. June 2010.
99. Todd Wittman, “The UCLA Math REU Program: Getting Students Involved in Research.” University of Southern California, Department of Mathematics Colloquium. Los Angeles, CA. April 2010.
100. Todd Wittman, Contributed talk. 25 minutes. “Variational Methods in Hyperspectral Image Processing.” *SIAM Conference on Analysis of Partial Differential Equations*. Miami, FL. December 2009.
101. Y. S. Cho, G. Ver Steeg, and A. Galstyan “Co-Evolving Mixed-Membership Blockmodels”, *NIPS Workshop on Networks Across Disciplines*, 2010.
102. A. Allahverdyan, A. Galstyan, and G.V. Steeg, “Clustering with Prior Information,” *NIPS Workshop: Clustering: Science or Art? Towards Principled Approaches*, 2009.
103. A. Galstyan, “Modeling Covert Activities with Hidden Markov Processes,” *SIAM CADS Mini-symposium on Terrorism Modeled as a Dynamical System*, Snowbird, Utah, 2009 (*invited*).
104. A. Galstyan and P.R. Cohen, “ Comparing Diffusion Models for Graph-Based Semi-Supervised Learning,” *6th International Workshop on Mining and Learning with Graphs (MLG-08)*, Helsinki, Finland, 2008.
105. A. Galstyan and P.R. Cohen, “Influence Propagation in Modular Networks,” *AAAI Symposium on Social Information Processing (SIP-08)*, Stanford, CA, 2008.
106. A. Galstyan, S. Mitra, and P.R. Cohen, “Probabilistic Tracking of Plans and Intentions in Intelligence Analysis”, talk presented at the *WNAR/IMS Annual Meeting*, UC Irvine, June 2007.
107. A. Galstyan, S. Mitra, and P.R. Cohen, “Detecting and Tracking Hostile Plans in the Hats World,” *AAAI Workshop on Plan, Activity and Intent Recognition (PAIR–07)*, Vancouver, Canada, 2007.
108. A. Galstyan, S. Mitra, and P.R. Cohen, “Probabilistic Plan Tracking and Detection for Intelligence Analysis,” poster presented as the *Joint Statistical Meetings (JSM)*, Salt Lake City, July 2007.
109. G. Medioni, Keynote lecture, Workshop on Perceptual Organization, San Francisco, CA, June 13, 2010.
110. G. Medioni, Keynote lecture, International Workshop on Computer Vision, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, July 14, 2010.
111. G. Medioni, Invited lecture, “Recent progress in object tracking (Multi target tracking, tag and track, active tracking, tracking in flow)”, INRIA, Rocquencourt, France, October 2009.
112. G. Medioni, Keynote speaker, Los Angeles/Anaheim, 2009 World Congress on Computer Science and Information Engineering, March 31, 2009.
113. G. Medioni, Keynote speaker, San Diego (Coronado), Automated Imaging, February 4, 2009.
114. G. Medioni, Keynote speaker, ISVC, Las Vegas, November 22, 2008.
115. G. Medioni, “Tensor Voting in 2 to N dimensions: Fundamental Elements,” Distinguished Lecture, Brown University, September 15, 2008.
116. V.V. Veeravalli, “Sensor Control for Information Collection and Fusion.” *International Workshop on Information Fusion*, Xi’an, China, August 2011 (Plenary Lecture).
117. V.V. Veeravalli and T. Banerjee, “Quickest Change Detection with On-Off Observation Control.” *International Workshop in Sequential Methodologies,* Palo Alto, CA, June 2011 (Invited).

118. V.V. Veeravalli and J. Fuemmeler, "Energy-Efficient Multi-Target Tracking Using Sensor Networks." *Army Conference on Applied Statistics (ACAS)*, Lexington, VA, October 2008 (Invited).
119. V.V. Veeravalli, "System-Theoretic Foundations for Sensor Networks." *IEEE Communication Theory Workshop*, Sedona, AZ, May 2007 (Keynote Lecture).
120. V.V. Veeravalli, "System-Theoretic Foundations for Sensor Networks." *IWWAN*, New York, NY, June 2006 (Keynote Lecture).
121. V.V. Veeravalli, "Smart Sleeping Policies for Wireless Sensor Networks." *NSF Workshop on Future Directions in Networked Sensing*, Boston, MA, May 2006 (Invited).

(2) Demographic Data for this Reporting Period:

(a) Number of Manuscripts submitted during this reporting period: 148
(b) Number of Peer Reviewed Papers published during this reporting period: 136
(c) Number of books: 3
(d) Number of Non-Peer Reviewed Papers submitted during this reporting period: 0
(e) Number of Presented but not Published Papers submitted during this reporting period: 121

(3) Demographic Data for the life of this agreement:

(a) Number of Scientists Supported by this agreement: 75
(b) Number of Inventions resulting from this agreement: 4
(c) Number of PhD(s) awarded as a result of this agreement: 7
(d) Number of Bachelor Degrees awarded as a result of this agreement: 10
(e) Number of Patents Submitted as a result of this agreement: 4
(f) Number of Patents Awarded as a result of this agreement: none
(g) Number of Grad Students supported by this agreement: 28
(h) Number of FTE Grad Students supported by this agreement: 12
(i) Number of Post Doctorates supported by this agreement: 13
(j) Number of FTE Post Doctorates supported by this agreement: 8
(k) Number of Faculty supported by this agreement: 12
(l) Number of Other Staff supported by this agreement: none
(m) Number of Undergrads supported by this agreement: 15
(n) Number of Master Degrees awarded as a result of this agreement: 4

(4) Student Metrics for graduating undergraduates funded by this agreement:

(a) Number of undergraduates funded by your agreement during this reporting period: 15

(b) Number of undergraduate funded by your agreement, who graduated during this period: 12

(c) Number of undergraduates funded by your agreement, who graduated during this period with a degree in a science, mathematics, engineering, or technology field: 12

(d) Number of undergraduates funded by your agreement, who graduated during this period and will continue to pursue a graduate or PhD degree in a science, mathematics, engineering, or technology field: 10

(e) Number of undergraduates funded by your agreement, who graduated during this period and intend to work for the Defense Department: none

(f) Number of undergraduates graduating during this period, who achieved at least a 3.5 GPA based on a scale with a maximum of a 4.0 GPA. (Convert GPAs on any other scale to be an equivalent value on a 4.0 scale.): 12

(g) Number of undergraduates working on your agreement, who graduated during this period and were funded by a DOD funded Center of Excellence for Education, Research or Engineering: none

(h) Number of undergraduates funded by your agreement, who graduated during this period and will receive a scholarship or fellowship for further studies in a science, mathematics, engineering or technology field: none

(5) Report of inventions

Patent applications:

1. G. Medioni and Q. Yu, USC File No:4048 “Spatio-Temporal Multiple Target Tracking Using Markov Chain Monte Carlo Data Association” .
2. G. Medioni and Q. Yu, USC File No:4109 “Online Tracking Using Co-trained Generative and Discriminative Trackers” .
3. G. Medioni and T. B. Dinh, USC File No:11-671, “Visual Tracking in Video Images in Unconstrained Environments by Exploiting On-The-Fly Context Using Distracters and Supporters”.
4. Provisional patent application filed by UCLA with US Patent Office titled “Data Fusion Mapping Estimation”.

(6) Honors and Awards

1. T. B. Dinh, Travel Grant to participate in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, Colorado, Jun 20-25, 2011.
2. T. B. Dinh, Travel Grant from NSF to participate n IEEE Conference on Automatic Face and Gesture Recognition (FG), Santa Barbara, California, Mar 21-25, 2011.
3. A.S. Polunchenko, Institute of Mathematical Statistics’ (IMS) Laha Travel Award, 2011.
4. Laura Smith, UCLA Dissertation Year Fellowship, 2011.
5. Nancy Rodriguez, National Science Foundation Postdoctoral Fellowship, 2011.
6. Andrea Bertozzi, Elected American Academy of Arts and Sciences, 2010.
7. Andrea Bertozzi, Elected SIAM Fellow, 2010.
8. Tony Chan, Elected SIAM Fellow, 2010.
9. V. Veeravalli was appointed IEEE Signal Processing Society Distinguished Lecturer for 2010-2011.
10. Andrea Bertozzi, Sonia Kovalevsky Prize Lecture, SIAM Annual Meeting, 2009.
11. P. Jeffrey Brantingham, (2009-Present) Los Angeles Police Department, Community Police Advisory Board on Counter-Terrorism, Appointed Board Member.
12. T. B. Dinh, 1st runner-up presentation award in scientific sessions of Annual Vietnam Education Foundation, Albany, NY, Jan 3-5, 2009.
13. Vlad Voroninsky, NSF Graduate Fellowship, 2008.
14. Tony Chan, Elected AAAS Fellow, 2007.
15. Alexander Tartakovsky, Abraham Wald Prize in Sequential Analysis, 2007.

(7) Press and News Articles

1. SIAM News Article, Swarming by Nature and by Design, front page article in November 2009 issue about work of coPI Bertozzi including work funded under this grant.
2. “Can Math and Science Help Solve Crimes?” *US News and World Reports*, March 2, 2010.
3. “The Math-ed Avengers: Can Sophisticated Models Fight Crime?” *Popular Mechanics*, February 2010
4. “Finding Coolable Hot Spots for Crime” *Science News*, Feb. 20, 2010
5. “The Mathematics of Clumpy Crime”, *Science*, Feb. 20, 2010
6. “Can Math and Science Help Solve Crimes?” UCLA Newsroom, Feb. 20, 2010
7. “Fighting Crime with Math” National Science Foundation video and press release, Feb. 20 2010
8. ”Gangs and Statistical Mechanics?”, *Science (blog)*, Feb. 20, 2010
9. “Fighting crime with math”, *Scientific American - 60 second podcast*, Feb. 20, 2010
10. “Mit Mathematik Verbrechen bekampfen”, *Zeit* (leading German Newspaper), Feb. 22, 2010
11. “Mit math gegen Kriminalität”, *Deutschlandradio*, Feb. 22, 2010
12. “Crime Hot Spots Can Be Cooled”, *Discovery News*, by Lisa Grossman, Feb. 23, 2010
13. “Los Angeles to Test Simulation Model that Predicts Criminals’ Reactions to Increased Policing”, Elaine Pittman, *Government Technology Magazine*, July 23, 2010
14. “Stopping crime before it starts”, Los Angeles Times, Aug. 21, 2010, by Joel Rubin.

15. “UCLA research project uses math to predict possible crime”, by Sarah Kahn, Daily Bruin, Univ. California, Los Angeles, August 30, 2010.
16. “The aftershocks of crime”, The Economist, Oct. 23, 2010, p. 98.
17. “Predictive Policing”, Fox News Video, Oct. 3, 2010, Interview with LAPD and Jeff Brantingham.
18. The New York Times Magazine 10th Annual Year in Ideas, December 19, 2010, ”Aftercrimes”.
19. Discover Magazine Top 100 Stories of 2010 # 60: Fighting Crime with Mathematics December 16, 2010, by Daniel Lametti.
20. “Sending the Police Before There’s a Crime”, August 15, 2011, The New York Times, by Erica Goode.
21. “Crime Fighting Math”, TV program, August 15, 2011, NBC Bay Area
22. “Police Predict Crimes as in ’Minority Report’ ”, TV program, August 18, 2011, ABC World News with Diane Sawyer
23. “Applied Math: A Tale of Robots, Burglaries, and Rapunzel’s Hair”, UCLA Graduate Quarterly Spring 2009

Chapter 2

Efficient Spatiotemporal Nonlinear Filtering Methods for Recognition and Tracking of Patterns and Trends

The work presented in this chapter has been performed by Dr. Rozovsky's group in Brown University in collaboration with Dr. Lototsky (USC), Dr. Mikulevicius (USC), Dr. Tartakovsky (USC), and Dr. Wan (Princeton).

1. Introduction

Until recently applications of nonlinear filtering (NLF) for hidden Markov models (HMM) have focused mostly on state processes of small to medium complexity insufficient for many applications of interest for DOD and the U.S. Army, in particular. One of the objectives of the research supported by this Grant is developing of spatiotemporal NLF algorithms and their DOD relevant applications.

The main scientific barrier in spatiotemporal filtering is enormous computational complexity. The fundamental challenge of our research is in finding effective models of spatiotemporal systems and the algorithms that would allow real time processing of these models.

Two different types of spatial structure appears in applications of interest for this research: video-streams (in video surveillance) and graphs (in information assurance (IA)). Some progress has already been made in application of NLF to analysis of video streams. However, the IA applications of NLF are quite novel. In video tracking, the spatial components are images characterized by shapes, curves, etc. In IA, the spatial components are graphs depicting relations between IP addresses and port numbers. Other variables, such as source address, source port, protocol, etc. also could be included. Source address and port, however, are typically spoofed in attacks (this corresponds to occlusions and distortions in the image analysis).

One of the main objectives of this project is development of Spatiotemporal NLF algorithms and their DOD relevant applications. The most promising developments in tracking of distributed images were made in [25]. There were interesting developments in quantum MCMC for tracking very large systems (see [140] and references therein); Zakai and Kushner equations for comparatively simple linear spatiotemporal state processes were derived in [3]. Nevertheless, the field is yet in its infancy.

2. Spatiotemporal Nonlinear Filtering for Recognition and Tracking of Patterns and Trends

2.1. A Bayesian Approach to Recognition of Patterns and Trends

In this project, we have farther developed a probabilistic Bayesian framework imbedded into dynamical distributed systems, more specifically, a nonlinear filtering approach for the analysis of heterogeneous spatiotemporal data. A distinctive advantage of NLF formalism is that it generates optimal tracking strategies.

The NLF approach turned out to be very instrumental in tracking point targets with highly nonlinear dynamics and observation structure. The extension of NLF to a spatiotemporal system attempted in the present research provides an algorithmic support for developing of multi-target multi-sensor trackers. Such trackers will be capable of dealing with a large number of extended targets with unknown shapes that are monitored by a network of stationary or moving sensors in complicated scenarios characteristic of urban warfare and counter-terrorism.

The analysis and synthesis of high-volume data and, in particular, spatiotemporal data is addressed by different disciplines and from different perspectives. Our approach is probabilistic in nature: more specifically, it is Bayesian. One important feature of the Bayesian approach is that it interprets the data not as a self-contained information depository but in light of already available knowledge (e.g., human expertise) regarding the events reflected by the data. This feature is an ideal instrument for keeping human operators "in the loop" in the process of automated decision making.

Nonlinear filtering is an extension of the Bayesian framework into dynamical systems. Kalman filter, designed for linear dynamical systems and linearly structured observations, is probably the most famous Bayesian filter. Generalizations of Kalman filter to nonlinear systems and/or observations are usually referred to as nonlinear filtering. NLF is a field on the cutting edge of contemporary stochastic analysis, information theory, and statistical inference. This is an emerging methodology with an enormous breadth of applications.

To be more specific, one could consider the following simple model, with two sequences $(\mathbf{x}_t)_{t\geq 0}$ and $(\mathbf{y}_t)_{t\geq 0}$, called *state* and *observations*, respectively. Often, the state is modeled as a Markov chain with the transition probability kernel $Q_t(x,y)$ and the initial distribution π_0. The observation sequence is related to the state by

$$\mathbf{y}_t = H_t(\mathbf{x}_t) + \mathbf{v}_t,$$

where $\mathbf{v}_t$ is noise (not necessarily Gaussian). The *a priori* information contained in this model consists of π_0, Q_t, and g_t, where $g_t(\mathbf{v})$ is the PDF of noise $\mathbf{v}_t$. This model is often referred to as a hidden Markov model (HMM) because the Markov chain $\mathbf{x}_t$ is hidden from the observer by a possibly nonlinear transformation H_t and noise $\mathbf{v}_t$. The goal of nonlinear filtering is to compute at each time t the posterior distribution $\pi_{t|t}$ of the state $\mathbf{x}_t$ given the realization of the observation $\mathbf{y}_{0|t} = (\mathbf{y}_0, \ldots, \mathbf{y}_t)$.

In the discrete time setting, the algorithm of the nonlinear filtering is very simple. It consists of two steps:

$$\begin{cases} \text{prediction :} & \pi_{t|t-1}(x) = \int Q_t^*(x',x)\,\pi_{t-1|t-1}(x')\,dx' \\ \text{correction :} & \pi_{t|t}(x) = \Psi_t(x)\,\pi_{t|t-1}(x) \,/ \int \Psi_t(x)\,\pi_{t|t-1}(x)\,dx, \end{cases} \tag{2.1}$$

where $\Psi_t(x) = g_t(\mathbf{y}_t - H(x))$ is the likelihood function. The second step is simply the Bayes formula.

The posterior distribution is the centerpiece of the Bayesian estimation. Indeed, the nonlinear filter $\hat{\mathbf{x}}_t = \int x\pi_{t|t}(x)\,dx$ is an optimal (in the mean-square) estimator for the state process $\mathbf{x}_t$. Also, the posterior distribution is extremely valuable for the "visualization" of pattern changes (see Figure 2.1).

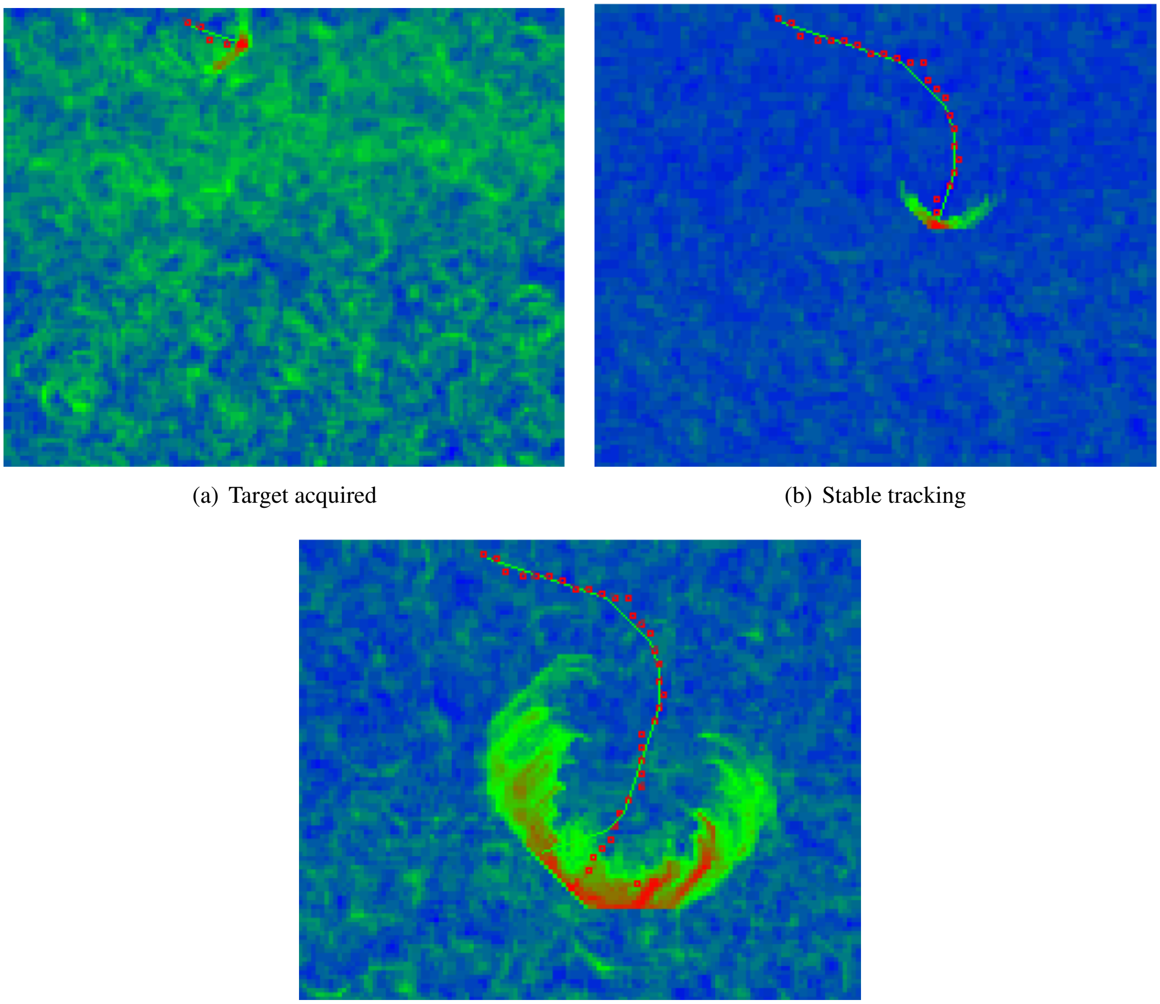
Figure 2.1 shows propagation of the posterior distribution of the position of an acutely maneuvering small and low SNR target (the signal-to-noise ratio (SNR) is -7dB). The spatio-temporal (S-T) input is IR video-stream. The brighter colors correspond to the higher values of the posterior distribution, and the green line is the true trajectory.

In the continuous time case, the prediction and correction steps merge and the posterior distribution $\pi_t(x) = \pi_{t|t}(x)$ is given by $\pi_t(x) = \phi_t(x) \,/ \int \phi_t(x')\,dx'$ and $\phi_t(x)$ is a solution of the stochastic differential equation

$$\dot{\phi}_t(x) = A^*(t,x)\,\phi_t(x) + H_t(x)\dot{\mathbf{v}}_t, \quad \phi_0(x) = \pi_0(x), \tag{2.2}$$

where A is the generator of the transition probability kernel $Q_t(x,y)$ and $\dot{\mathbf{v}}_t$ is white noise. For example, if the state process is given by the noisy kinematic equation $\dot{x}_t = a(t,x_t) + \sigma\dot{\varepsilon}_t$, where $\dot{\varepsilon}_t$ is white noise independent of $\dot{\mathbf{v}}_t$,

$$\dot{\phi}_t(x) = \frac{\sigma^2}{2}\phi_t''(x) - (a(t,x)\,\phi_t(x))' + H_t(x)\phi_t(x)\,\dot{\mathbf{v}}_t. \tag{2.3}$$



(a) Target acquired

(b) Stable tracking

(c) Change of pattern/turn

Figure 2.1: Propagation of the posterior distribution.

Note that equation (2.3) is a stochastic partial differential equation and should be understood in the sense of Itô calculus. Equation (2.2) is usually referred to as the Duncan-Mortensen-Zakai equation and its solution $\phi_t\left(x\right)$ as the unnormalized filtering density (UFD). The posterior density $\pi_t\left(x\right)$ solves a Kushner equation that is similar to the Zakai but slightly more complicated. The theory of nonlinear filtering was initiated in the 1960s (see, e.g., [20, 51, 79, 80, 160, 184, 197], etc.) More contemporary reviews of NLF methodology could be found in [46, 54, 72, 110, 141]). A comprehensive review of the state of the art of NLF is presented in Crisan and Rozovskii [39].

The simplicity of the nonlinear filtering algorithm is deceptive. While being extremely effective and stable, this algorithm is computationally expensive due to its polynomial complexity. The most taxing part of the algorithm is, of course, computing the integrals which has to be repeated on line every time when a new observation arrives. Direct quadrature methods are effective only when the dimension of the state process is not larger than 3. To overcome this complication, a new powerful methodology of Particle Filtering was introduced and developed in the 1980s -1990s. The idea of this approach was to replace quadratures in (2.1) by Monte Carlo averaging. At time t, the averaging is performed with respect to identically distributed random particles

$$\left(\mathbf{x}_{t-1|t-1}^{(1)}, \ldots, \mathbf{x}_{t-1|t-1}^{(N)}\right)$$

with the probability distribution $\pi_{t-1|t-1}^{N}$ computed on the previous step using the recursion (2.1) with quadratures replaced by Monte Carlo averaging. Generation of the particles

$$\left(\mathbf{x}_{s-1|s-1}^{(1)}, \dots, \mathbf{x}_{s-1|s-1}^{(N)}\right)_{s=0,1,2,\dots}$$

is called the resampling procedure. It plays the central role in the success of the method. The nonlinear filter based on the described procedure is often called Markov Chain Monte Carlo (MCMC) filter or particle filter. Various versions of the optimal nonlinear filters based on Monte Carlo resampling were developed in the 1990s, including the iterative particle filter (IPF) [109], sampling/important resampling (SIR), particles filter [46], and a branching particles filter (BPF) [38], etc. For a review, see [46, 110].

2.2. Fundamental Equations of Spatiotemporal NLF and NLF Algorithms for Complex Hidden Markov Models

One of the main objectives of this project is development of spatio-temporal NLF algorithms and their DOD relevant applications. Certain progress in this direction has been made during the current stage of the grant.

To address tracking in distributed images we have derived a preliminary version of Zakai and Kushner equations for spatial-temporal observation process with continuous and discrete observations. In particular, we have derived the analogs of Zakai and Kushner equations of nonlinear filtering in this setting.

Zakai and Kushner equations of nonlinear filtering are theoretically sound and well understood. However, even for comparatively simple systems they are too computationally intensive. To bypass this problem we have developed a new "smart models" of the state. It turned out that stationary and evolution systems driven by space-only noise provide substantial computational advantages. During this period, we have investigated systems of this type described by Stochastic PDEs (see Lototsky and Rozovskii [97, 98, 99]).

To reduce the computational complexity we have also developed state models provided by telescoping Markov processes. The simplest example of this type of processes is referred to as interacting multiple models (IMM). Our recent results will allow to extend this methodology to very complicated systems. Telescopic Markov processes are branching processes of special kind. They could model enormously complicated systems. However, processing of these systems, if done judicially, could be performed on line. The key to this goal is to do estimation (filtering) and hypothesis testing (evaluation of the relevance of each branch) simultaneously. The main benefit of this approach is in early truncation of "low priority" branches.

Partial testing of the obtained algorithm was done on the problem of tracking volatility in financial markets which present a great challenge by their complexity [40, 41].

2.3. Visual Tracking of Dim Extended Targets

Tracking of moving and deforming multiple objects (e.g., battlefield monitoring, visual surveillance in urban environment, etc.) has always been a topic of substantial interest for military and intelligence services. The ability to track extended targets in image sequences is a fundamental problem with many applications of ultimate relevance to the ARMY and DOD. While a substantial progress in this field has been made in the last decade, the development of a robust framework for tracking extended targets subject to obscuration, changes in illumination, pose, etc. is still a challenge. We have developed NLF-based spatial-temporal technology capable of tracking small targets with very low SNR (up to -7dB) in plain images. It is worth mentioning that this technology allows for handling very dim moving targets with evolving appearance in noisy and cluttered environments. Our method is based on combination of nonlinear filtering for interacting multiple models and recently developed Bayesian marginalization technique. Introduction of interacting multiple models allows to account for evolving appearances of the targets due to maneuvering, obscuration, etc. The marginalization technique dramatically reduces the computational complexity of the algorithm. In particular, it makes possible real time multitarget tracking. The precision of the NLF algorithm allows surveillance with low SNR.

Figure 2.2 illustrates the results of stabilization and target tracking with the NLF algorithm for a very difficult UAV scenario that includes severe translations and rotations.

Further results of implementation of this approach to detection and tracking of low-observable targets

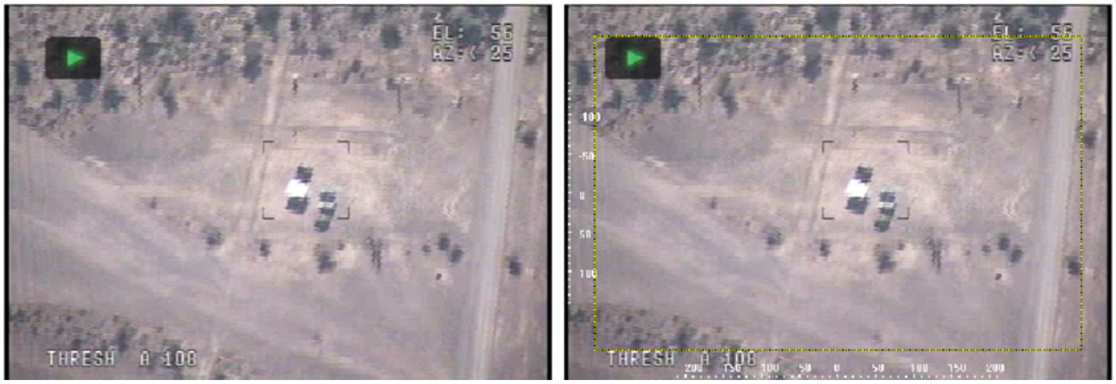


Figure 2.2: Video tracking for UAV data.

in video and IR images and its efficiency compared to industry standard methods will be demonstrated in Chapter 13, Section 6.

3. Nonlinear Filtering for Tracking and Identification of Intentions

One of the central directions of Dr. Rozovsky's research is applications of Nonlinear Filtering to tracking complex distributed systems. In the course of this project this research was extended beyond physical systems. In particular, current methodology is designed to track and identify plans and intensions rather than position, identity, direction of motion, etc. The mathematical foundation of this methodology is based on spatial-temporal processes coupled with "telescopic" Markov processes. This presents a substantial extension of the current paradigm of Hidden Markov Models.

Two approaches for modeling the state process were investigated: direct telescopic Markov chain and a combination of fast mean-reverting diffusion model parameterized by simple telescopic Markov chain. Low computational simplicity is the main advantage of the new approach based on the mean-reverting diffusion. Clearly, low computational complexity allows us to deal with very complex multidimensional systems which is extremely important for the needs of practical video surveillance.

Figure 2.3 illustrates the results of motion detection and tracking in a sequence of images. Despite the fact that the object has very low visibility its motion is tracked very reliably.

4. Nonlinear Filtering Methods for Tracking Hidden Attributes and Inference of Collective Behaviors

4.1. Tracking Hidden Attributes

In 2010 A. Papanicolaou and B. Rozovsky have developed a new parallelization of computations algorithm and applied it to the Pinocchio example (see Figure 2.4) The resulting speed-up credited to the parallel architecture for Pinocchio was quite significant. This example provided us with a sense of what kind of parallel architecture would be appropriate for general nonlinear filtering algorithms for video tracking. It was developed into reasonably universal algorithm for parallelization of algorithms of nonlinear filtering for spatial-temporal data.

Algorithms for tracking complex objects require high-resolution image data as input. This results in a large set of computations to obtain the statistics for an observed image given every possible hypothesis. We have developed a parallel architecture to distribute the tasks associated with calculating image statistics required to temper the posterior distribution. In particular, for the particle filter that has been used in our

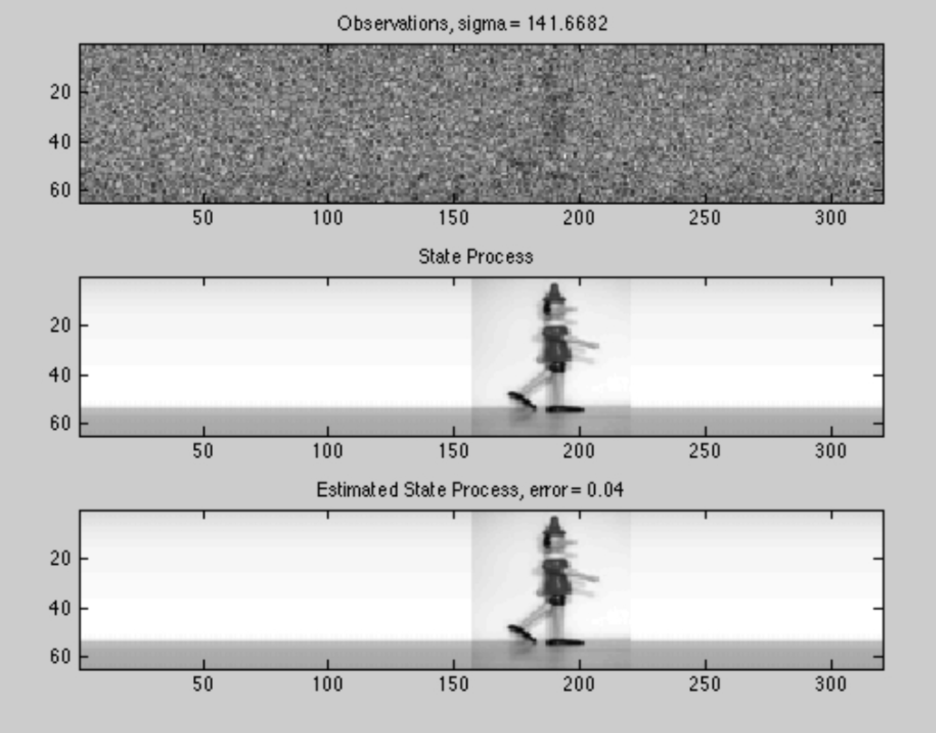


Figure 2.3: Motion detection and tracking in images.

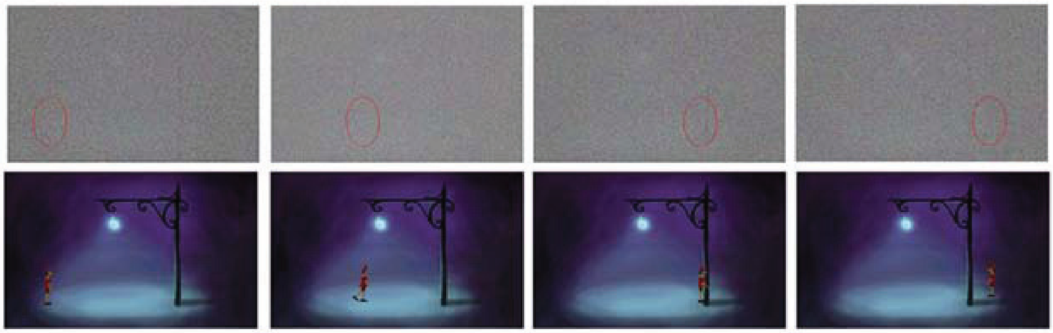


Figure 2.4: Tracking Pinocchio in high noise.

algorithm, computing of the main covariance

$$(y_k - H(x_k^{(i)}))^* R^{-1} (y_k - H(x_k^{(i)}))$$

is hardly possible without extensive parallelization. Standard parallelization algorithms used for similar computations are based on the so called master-slave approach. However, the master-slave architecture is effective only if the sample size N is relatively small. As N grows the workload delegated to master rank will eventually surpass the workload of each slave rank, and such a situation is not a good use of computing resources because the bulk of the processing power will idle as the master rank works. In other words, the workload is not well balanced, or we say there is "workload imbalance." For large N, we have implemented a new parallel architecture that distributes the global workload equally among all ranks, but does not become crippled by the cost of communicating particle weights, particular when importance sampling is invoked. The algorithm also takes advantage of importance sampling and bootstrap sampling. However, a regular bootstrap will defeat the purpose of locally independent particle filters because it creates a complicated global exchange of all local particle data among all of the ranks. To deal with this problem we have implemented a truncated global bootstrap sampling where weights below a certain threshold are truncated to zero when importance sampling is invoked.

Figure 2.4 illustrates the result of tracking.

The related results were partially reported in the paper A. Papanicolaou and B. Rozovsky, "Tracking Hidden Attributes," Submitted to *SIAM J. Imaging Sciences*, 2010.

4.2. Inference of Collective Behaviors: Intent and Target Identification in a Bird Flock

An important direction of our research is tracking intensions of very large groups of agents that exhibit both group and individual behavior. Collective behavior is manifested in a variety of complex systems ranging from bacterial colonies, bird flocks, groups of terrorists, insect swarms, troops on the march, and even to pedestrians. The hallmark of collective behavior is the emergence of self-organization: numerous but simple local interactions among components result in a complex global behavioral pattern. Besides its magnificence, collective behavior has been shown to provide benefits to the members of a group in many facets of their lives such as forage efficiency and reproduction. One of the features that make these possible is a sudden coherent change in direction. This is achieved through rapid transfers of directional information, which are more efficient and faster than direct communications.

Rapid coherent changes are typically induced by the combination of external and internal perturbations. Presence of predators is an example of external perturbation while intrinsic noise is an example of internal one. The prediction and filtering in real-time of these changes are critical to control and monitor collective groups. Quantities related to sudden changes are usually hidden and accumulates slowly over a long period of time. After a certain threshold, however, these quantities take a massive effect on the apparent dynamics of a group and produce rapid collective changes. Our objective in this study is to monitor and predict hidden quantities controlling the sudden coherent changes. Furthermore, we seek to identify the source of these changes. Specifically, we are interested in the collective landing of a bird flock, which usually occurs rather abruptly from horizontal flight over a food source. We model a flock of birds performing foraging flight over a field with sporadic food sources. Each bird flies with others in harmony and has individual landing intent that evolves over time; mainly increases due to hunger. If some birds identify a food source, their landing intent increases rapidly and a part of them starts landing. Via local interaction that is analogous to the exchange of social information, other birds begin landing. We model each bird as a self-propelled particle that assumes a bird to be a lifeless particle and be governed by Newton's laws. In this framework, the social interactions between birds can be modeled as Newtonian forces or interaction potentials. The landing intent and the availability of a food source are incorporated into the model as internal variables. Since these are hidden, an appropriate framework for this inference problem is the Telescopic Markov chain (TMC) where the hidden variables reside higher levels in a system hierarchy and evolve according to their own dynamics conditioned on the higher-level variable. Figure 2.5 shows an evolution of a 30-birds flock. After initial transient stage, a group of birds maneuver a steady flight until a sudden landing phase. Boxed parts in the left figure (a) are magnified in (b) and (c), respectively. The arrows represent velocities of birds.

The inference problem of TMC is similar to the hidden Markov model (HMM) applied to vector-valued Markov chain. However, due to its iterative structure, TMC dramatically reduces the dimension of state space. In this work, the observation only takes the location of each bird in a flock at discrete time steps. Since the lowest level of TMC in our model can be cast into a linear dynamical system, the proper inference model in this level is the Kalman filter. Combining TMC with the Kalman filter, the inference problem can be formulated in the framework of interacting multiple models.

The model and algorithm developed in this work have great potential for other fields of interest such as recognition of emotions from facial expressions, multi-target tracking and element recognition, plan recognition of a multi-agent system, and cooperative decision making. A straightforward application would be brought up in the context of anti-terrorism on the extension of our previous work of Papanicolaou and Rozovsky. While intent inference is performed for one agent from noisy video footage in our previous work, the result here is to be used for intent inference in a multi-agent system as well as target identification.

Several possible improvements are left for future research. First, we need to develop an elaborate model

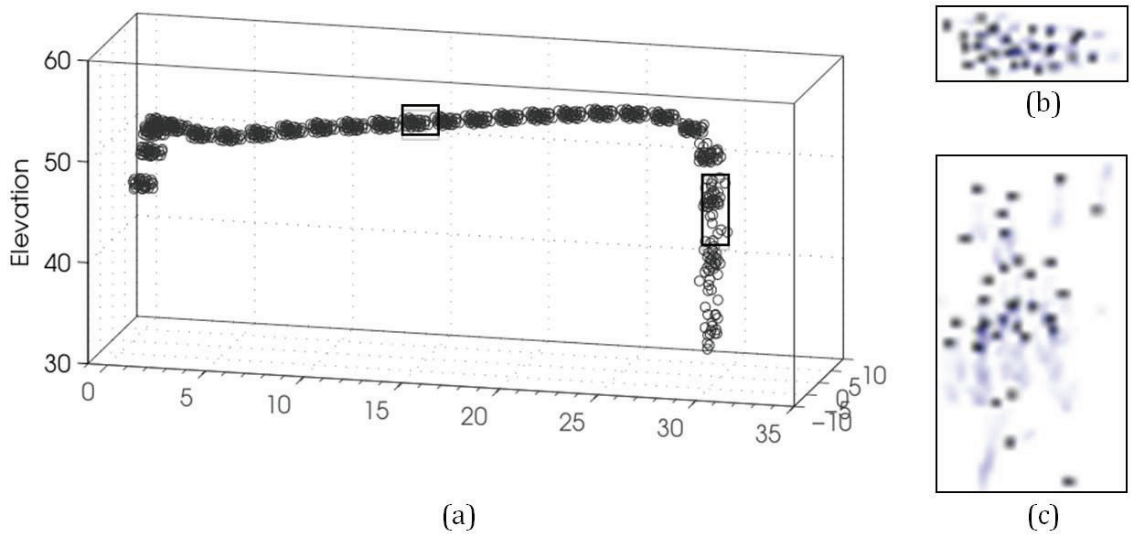


Figure 2.5: Evolution of a bird flock.

for agent interactions, which properly reflects a real-world situation of information exchange. This includes a model for communication mechanism as well as one for agent relationships. Since the relationship between agents could be cooperative or self-seeking, game theoretic approaches would be needed. Second, the role of noise needs to be investigated. In the biological context, noise is found to be beneficial to rapid transitions. However, little has been known for the social context. Last, a simple Markovian assumption may not sufficient to explain complex behaviors and needs to be modified. Collective memory, which considers knowledge of the current state as well as the past, would be needed for complex inference problems. The related results were partially reported in the paper by Park, Rozovsky, and Sowers [120].

5. Uncertainty Quantification in Covert Networks

In collaboration with S. Lototsky (USC) and X. Wan (Princeton) we have developed a novel approach to *uncertainty quantification* (UQ) for large stochastic systems. It applies to various complex random structures, in particular to *covert networks*. *Terrorist networks* is a singularly important application of this methodology.

The main achievement of our program is the UQ method for recovery/estimation of system's structure from the "camouflaged" data available for observations.

Our approach to UQ in large stochastic systems is based on powerful technique originated in Quantum Probability. This technique is often referred to as Polynomial Chaos approach. We use this technique to *filter* the noisy information gathered in the process of system monitoring and to *discover* the true structure of the system (e.g., covert network). The theoretical and numerical foundation of the proposed methodology were described in our recent papers.

The results of this research is published by Proceedings of the National Academy of Science, USA (see the paper "A new stochastic modeling methodology based on weighted Wiener chaos and Malliavin calculus" by Wan, Rozovskii, and Karniadakis [182]). A more general setting was considered in the paper "Elliptic equations of higher stochastic order" by Lototsky, Rozovskii, and Wan [100]. The mathematical foundations for these developments are documented in "A unified approach to stochastic evolution equations using the Skorokhod integral" by Lototsky and Rozovskii [98] and "Stochastic differential equations driven by purely spatial noise" by Lototsky and Rozovskii [99].

6. Conclusion

Our results prove that optimal spatiotemporal NLF methods allow for detection and reliable tracking of small dim objects when standard tracking methods fail (see also Chapter 13).

The developed novel approach to uncertainty quantification can be applied to covert terrorist networks to recover their true structures.

Current technology of precision positioning/navigation of aircraft, tracks, ships and other moving platforms critically depends on availability of satellite guidance (GPS, etc.). Major terror acts or war could disrupt satellite based navigation. Also, some covert Army operation might require electronic silence. If this is the case, the standard satellite based navigation must be ruled out. In contrast, the NLF based algorithms are fully autonomous in that they do not require satellite based inputs. Instead, they utilize recently developed electronic Geographic Information System (GIS) which is available in the on-board version.

Chapter 3

Target Tracking Concepts for Distributed Targets with Unknown Shapes: Multiple Target Tracking and Tag & Track

The material of this chapter is based on the results of the group of Dr. Medioni, USC (Sections 1–9) and Dr. Chan, UCLA (Section 10).

1. Introduction

Extended target tracking is a fundamental problem in video surveillance, as it provides the description of spatio-temporal relationships between observation and targets in the scene. As we discuss above, there are two core subtasks that we aim to address, one is Multiple Target Tracking and the other is Tag-and-Track. The two subtasks both have wide application areas in visual surveillance and content-based video analysis.

Detecting and tracking multiple moving objects from a moving platform, *e.g.* moving objects from an airborne camera, presents the following challenges. As the size of objects is relatively small from an airborne view, appearance based detectors suffer from lack of resolution and blurry images. On the other hand, the motion-based object detection approach relies on the stabilization of the parametric camera motion model (affine or homography). Moving objects are defined as the areas that have not been stabilized. This method works well when the scene can be considered planar, or when the motion of the camera is pan/tilt/zoom. However, 3D depth in the scene produces pixel displacement, which cannot be accounted for by the global parametric model, usually termed as parallax. Other difficult cases affect detection and tracking in airborne videos, such as abrupt illumination changes, registration errors and occlusions. Many approaches have been proposed to improve motion detection and tracking on frame-by-frame and pixel-by-pixel bases, *e.g.* global illumination compensation [190], parallax filtering [195], or detection using contextual information [68, 188]. No much attention has been paid on analyzing the long-term motion pattern of moving objects, which is a distinctive property for moving vehicles in airborne videos. Conceptually similar to "track-before-detect" techniques, we aim to involve temporal information in process as early as possible. Indeed, detection and tracking are coupled: if perfect detection is given, tracking becomes relatively straightforward, on the other hand, if we know the motion and trajectory of an object, detection is easier. However, surveillance videos are often of low quality, and the size of moving objects is relatively small from an airborne view, so the lack of resolution makes it difficult or sometimes even impossible for appearance based detectors to work in complex scenes.

Another important subtask is to track a single object of interest. This subtask of tracking does not require the compensation of camera motion and doest not require any prior knowledge of an object of interest. The object of interest can be any object that is specified by a user during the initialization, *e.g. a person, a vehicle, or an airplane, etc.* The essential problem in this sub-task is to online establish an *appearance model* to describe the appearance of an extended target and update the model on the fly to adapt to appearance changes, which can be caused by varying viewpoints and illumination conditions. Appearance can also change relative to background due to the emergence of clutter and distracters. Also, an object may leave the field of view (or be occluded) and reappear. Another critical challenge is when there are some objects similar to our target appearing in the frame, which makes it hard to model the appearance of the target to distinguish it from the others. While the generative models easily fail to separate them due to the lacks of discriminative power, the discriminative models lead to over-fitting problem when trying to. Moreover, real-time performance in this subtask is an important and challenging requirement.

The main objective of this work is to provide the capability to track a single or multiple extended targets from a moving platform, *e.g. from an airborne camera*. Tracking is a critical component of video surveillance, as it provides the description of spatio-temporal relationships between observations and targets in the scene required by activity recognition modules for surveillance purpose. There are two subtasks that we aim to deal with: 1) Multiple Target Tracking 2) Tag-and-Track.

For multiple target tracking, we proposed a general framework which makes use of spatio-temporal consistency in both motion and appearance and does not require the one-to-one mapping between observations and targets. Inferring the association and targets states according to current observations essentially suffers from the ambiguity existing in data association [32, 194]. Our method works under a deferred logic framework, where the decision is made when enough observations are obtained. In order to deal with the high computational complexity of such an association scheme, a Data-Driven Markov Chain Monte Carlo (DD-MCMC) [177] method is proposed to sample the solution space. Both spatial and temporal association samples are incorporated into the Markov chains transitions. We also proposed a tracklet-based approach and tried to overcome some of the disadvantages of existing methods. It does not require an exhaustive evaluation of data association hypotheses. It does not assume one-one mapping between observations and objects, and provides a confidence measure on each tracklet. The algorithm accomplishes this by formulating the tracking problem as inference in a set of Bayesian networks, and uses consistency of motion and appearance as the driving force. The computed tracklets are then used in a complete multi-object tracking algorithm. Moreover, we proposed to analyze the motion patterns formed by moving object over time, which provides a distinctive property to detect single or multiple moving objects in a spatio-temporal volume. We first provide a straightforward geometric interpretation of a general motion pattern in 4D space (x, y, v_x, v_y), which can describe a large amount of commonly seen 2D motion patterns, *e.g.* traffic at a busy intersection, crowds on a sidewalk. We propose to use the Tensor Voting computational framework to detect and segment such motion patterns in 4D space. Beyond segmenting motion patterns, we apply this technique to facilitate the detection and tracking of each individual object in such a motion pattern.

For the tag-and-track problem, in order to track and reacquire an unknown object with limited labeling data, we propose to learn these appearance variations online and build a model that describes all seen appearance while tracking. To address this semi-supervised learning problem, we propose a co-training based approach to continuously label incoming data and online update a hybrid discriminative generative model. It was then improved in running time by using a co-trained cascade particle filter framework. The cascade manner of organizing the particle filter enables the efficient evaluation of multiple appearance models with different computational costs; thus improve the speed of the tracker. Also, we proposed a method to handle partial occlusion which is a critical factor causing drift in tracking. Moreover, we proposed to use context information to enhance the performance of the tracker and deploy it in a full automatic surveillance system where a pedestrian is detected, and an active camera follows him to acquire the high resolution face sequence.

2. Spatio-Temporal Monte Carlo Markov Chain Data Association

Assume that there are K *unknown* moving targets within the time interval $[1, T]$. Let y_t denote the set of foreground regions at time t, Y is the set of all available foreground regions within $[1, T]$. Here, we define the tracking problem as: given the observation Y, infer an unknown number of K tracks $\omega = \{\tau_0, \tau_1, \cdots, \tau_K\}$, where τ_0 is the set of false alarms, τ_k is the k^{th} track. Each τ_k in ω is defined as a sequence of shapes.

For simplicity, we use rectangles to represent the shapes covering the foreground regions. In the case of a single target with perfect foreground segmentation, the set of MBRs (*Minimum Bounding Rectangles*) of each foreground region at different times forms the best cover of the target. However, when inter-occlusion between multiple targets and noisy foreground segmentation exist, it is not trivial to find the optimal cover. In our framework, the tracking problem is formulated as maximizing a posterior (MAP) of a cover of foreground regions, given the set of observations Y, *i.e.*, $\omega^* = \arg\max(p(\omega|Y))$. By introducing the concept of cover, we overcome the one-to-one assumption at each time instant, one foreground region can be covered by more than one target and one target can cover more than one foreground region as well.

To find a cover with reasonable properties, we first define a prior model which considers the following criteria: we prefer long tracks with few false alarms. In addition, one track should have little overlap with other tracks. We adopt the prior probability of a cover ω as the product of each prior terms. Also, we consider a probabilistic framework for incorporating two parts of likelihoods: motion likelihood L_M, appearance likelihood L_A. We represent the elements (rectangles) in track k as $(\tau_k(t_1), \tau_k(t_2), \ldots, \tau_k(t_{|\tau_k|}))$, where $t_i \in [1, T]$, and $(t_{i+1} - t_i) \geqslant 1$, since missing detection may happen. Given one cover, the motion and appearance likelihood of a target is assumed to be independent of other targets. The joint likelihood of a cover can be factorized in Eq.3.1.

$$\begin{aligned} p(Y|\omega) &= \prod_{k=1}^{K} L(\tau_K) = \prod_{k=1}^{K} \prod_{i=1}^{|\tau_k|-1} L\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) \\ &= \prod_{k=1}^{K} \prod_{i=1}^{|\tau_k|-1} L_M\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) L_A\left(\tau_k(t_{i+1})|\tau_k(t_i)\right) \end{aligned} \tag{3.1}$$

With some manipulations, we combine the prior and the likelihood $p(\omega|Y)$ in Eq. 3.1 to have the whole posterior represented in Eq. 3.2.

$$\begin{aligned} &p(\omega|Y) \propto \exp\{-C_0 S_{len} - C_1 K - C_2 F - C_3 S_{olp} - C_4 S_{app} - S_{mot}\} \\ &S_{len} = -\left(\sum_{k=1}^{K} |\tau_k|\right), S_{olp} = \left(\sum_{T=1}^{T} \Gamma(t)\right), S_{app} = \sum_{k=1}^{K} \sum_{i=1}^{L_k-1} D(\tau_k(t_i), \tau_k(t_{i+1})) \\ &S_{mot} = \sum_{k=1}^{K} \sum_{i=1}^{L_k} \left(\log\left(\det(P_i)^{1/2}\right) + \tfrac{1}{2}\mathbf{e}_i^T P_i^{-1} \mathbf{e}_i\right) \end{aligned} \tag{3.2}$$

where $C_0, \cdots, C_4$ are positive real constants, which are determined automatically by Linear Programming in the training phase [192]. Eq. 3.2 reveals that the MAP estimation is equivalent to finding the minimum of an energy function. The tradeoff between prior and posterior will lead to a MAP solution.

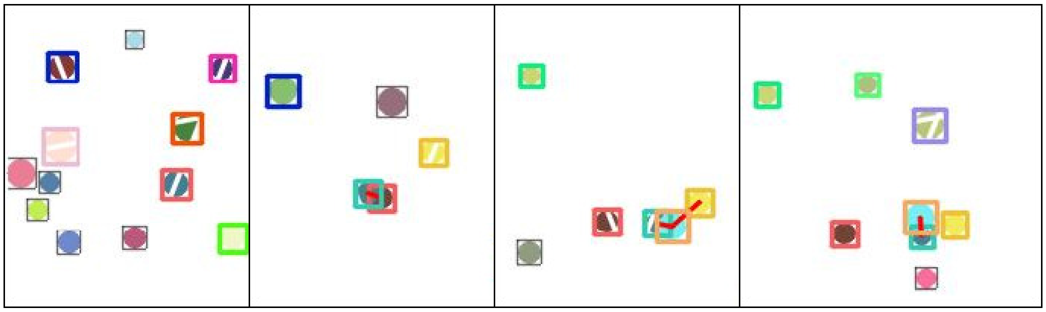
Searching in such a solution space for Eq.3.2 is not trivial. We propose to use a data-driven MCMC to estimate the best spatio-temporal cover of foreground regions. To ensure that detailed balance is satisfied, the Markov chain is designed to be ergodic and aperiodic. It is also important to design samplers that converge quickly.

In our proposal distribution, the sampler contains two types of moves: temporal and spatial moves. Temporal moves correspond to changing the labels of rectangles at different time instants, while spatial moves change covering rectangles at one time instant.

The input to the algorithm is the set of original foreground Y, initial cover ω_0 and the total number of samples n_{mc}. Each move is sampled according to its own prior probability. Note that, instead of keeping all samples, we only keep the cover with the maximum posterior since we don't need the whole distribution but only the MAP estimate. For the same reason, there is no burn-in procedure. The rectangles in the initial cover ω_0 are directly obtained from MBRs of foreground regions. Given the stationary distribution $\pi(\omega) = p(\omega|Y)$, the acceptance ratio $A(\omega, \omega')$ is defined as follows.

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega|\omega')}{\pi(\omega)q(\omega'|\omega)}\right) \tag{3.3}$$

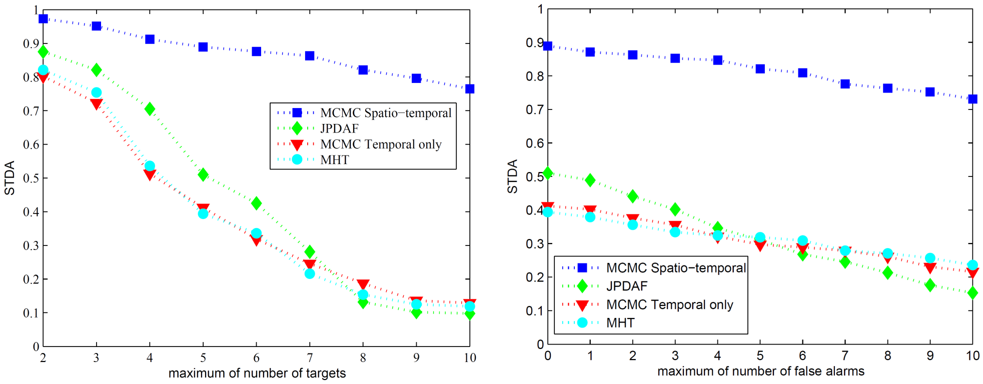
To demonstrate the concept of our approach, we design simulation experiments. In a $L \times L$ square region there are K (*unknown number*) moving discs. Each disc presents an independent color appearance and an independent constant velocity and scale change in the 2D region. False alarms (non-overlapping



(a) 1^{st} frame (b) 30^{th} frame (c) 43^{th} frame (d) 50^{th} frame

Figure 3.1: Simulation result $L = 200$, $N = 7$, $FA = 7$ and $T = 50$. Targets may split or merge when they appear.

with targets) are *u.a.r* (uniform at random) located in the scene and the number of false alarms is an uniform distribution on $[0, FA]$. We compare the tolerance of the target density and false alarms with other methods, including a JPDAF [12] based method from [76], the MHT from [36] and our own algorithm with only temporal moves. For each different setting, we generate 20 sequences and each sequence contains $T = 50$ frames. The MCMC sampler was run for a total of 10K iterations where the first 15% iterations consist solely of temporal moves. The average score from multiple runs of our method is reported. All four methods employ the same motion and appearance likelihood. Figure 3.2(a) compares the performance when the number of targets increases. Figure 3.2(b) shows the tolerance to false alarms for different methods. Because we consider the spatial and temporal association seamlessly, our method is able to handle the case when split or merged observations exist.



(a) $FA = 0$ and $T = 50$ (b) $N = 5$ and $T = 50$

Figure 3.2: (a)STDA as the function of N the maximum number of targets, (b)STDA as the function of FA the number of false alarms

In summary, we proposed a framework to find a global optimal spatio-temporal association which maximizes the consistency of motion and appearance of targets over time. Our method overcomes problems encountered with one-to-one mapping between observations and targets. A data driven MCMC method is used to sample the solution space efficiently and the forward and backward inferences enhance the search performance. Compared to other data association algorithms, the proposed method shows remarkable improvement both temporally (i.e. consistency of labels) and spatially (i.e. accuracy of outlined regions). The work can be extended along the following lines: first, the target motion model can be extended to a more general model. Second, our framework can naturally incorporate object model information in two ways: 1) we can assign a model likelihood for each node to extend our likelihood function. 2) we will also use model information to drive the MCMC proposal.

To track from a moving camera, we need to project targets at different times into a common reference

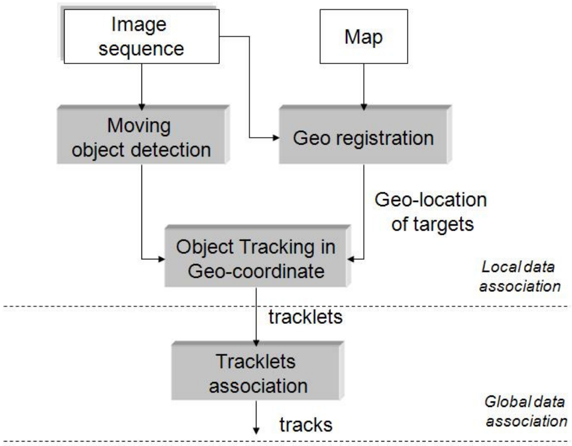


Figure 3.3: Overview of the geo-tracking framework

frame. Accumulated errors are introduced when fixed coordinates are selected and no further alignment is performed. Usually the first frame [76] or the ground plane in the first frame is selected as the reference frame. Moreover, due to scale change, image coordinates of the targets are not meaningful. Here, we propose to use a global map (a satellite image) as the reference frame. By registering UAV (Unmanned Aerial Vehicles) images with the satellite image, we can generate the absolute geo-location of targets. Also, tracking is performed in geo-coordinates, which have clear physical meaning. In surveillance applications, occlusion is common. We introduce a two-step procedure for tracking with occlusion. The first step (called local association) links detected regions within a sliding window and generates tracklets. The second step (called global association) links the tracklets to form longer tracks and maintain tracks ID.

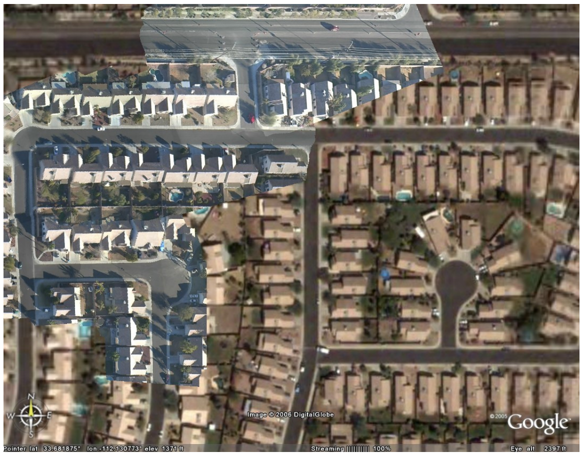
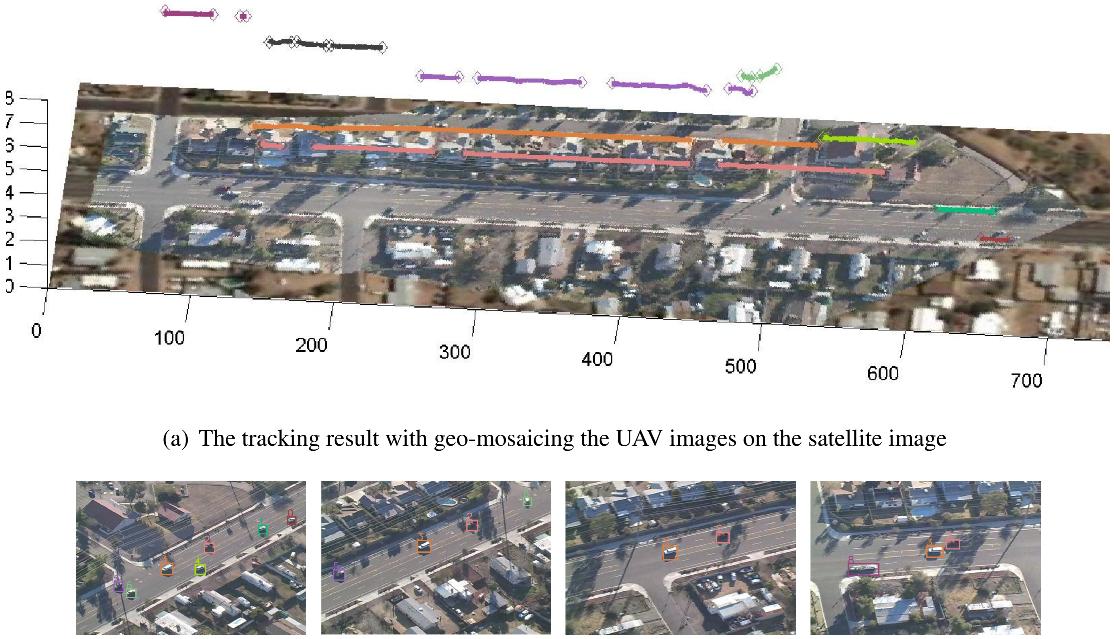


Figure 3.4: Geo-mosaicing 2000 consecutive frames on top of the reference frame.

We have discussed that data association is essential for successful tracking. We propose the above MCMC data association algorithm to deal with local data association since errors in local association are not rectified in the global one. We formulate the local association as multiple targets tracking, in which the purpose is to find the best partition of observation (*i.e.* detected moving regions) graph. In the global association, by assuming the maximum speed and acceleration of targets on the geo-coordinates, we can define the compatibility of tracklets and this reduces ambiguity in tracklet association. In addition, we adopt rotation invariant appearance descriptors [77] to represent both color and shape distribution of targets in each tracklets. The overview of the tracking framework is shown in Figure 3.3.

The geo-registration result is shown in Figure 3.4. Figure 3.5 shows the tracking result on the sequence with multiple moving targets. Again when targets are occluded by shadows, local data association may lose the track identification and thus tracklets are formed. The missing detection caused by occlusion even lasts for longer than the sliding window of local data association (45 frames). However in global data association, the tracklets are associated with correct ID throughout the video. The different tracks are listed in the Z direction in different colors. Figure 3.5(b), 3.5(c), 3.5(d) and 3.5(e) show the beginning frame of the tracklets of the red truck. Although the appearance of the white van and the white SUV in 3.5(b) is quite similar, the temporal and spatial constraint on the global map prevents from associating them together.



(a) The tracking result with geo-mosaicing the UAV images on the satellite image

(b) (c) (d) (e)

Figure 3.5: The tracklets and tracks obtained using the local and global data association framework. The UAV image sequence is overlayed on top of the satellite image.

3. Inferring Tracklets for Multi-Object Tracking

One of the keys to the success of our proposed DD-MCMC approach before is an efficient exploration of the search space. It tries to make more informative moves, and thus are effective enough to be used in practice.However, when being applied into aerial context, where there are a number of object detected, MCMC-based approach is overwhelmed because of the need of an immense number of samples to converge to a good solution. In addition, the low frame-rate causes many dificulties in just initializing the set of tracks. With bad initialization and a large number of detections, the problem becomes intractable for the MCMC-based approach. To address this issue, we proposed a tracklet-based approach and tried to overcome some of the disadvantages of existing methods. It does not require an exhaustive evaluation of data association hypotheses. Also it is a MAP estimate, rather than a heuristic. It does not assume one-one mapping between observations and objects, and provides a confidence measure on each tracklet. The algorithm accomplishes this by formulating the tracking problem as inference in a set of Bayesian networks, and uses consistency of motion and appearance as the driving force. The computed tracklets are then used in a complete multi-object tracking algorithm.

The goal of our algorithm is to infer tracklets, each representing one object, over a (sliding) window of frames. This window is usually 4-8 seconds in length. The input to our algorithm is a set of object detections (blobs) in each frame. These can be as simple as connected components taken directly from background subtraction, or they can be the output of a more complex object detector. Each object detection also has

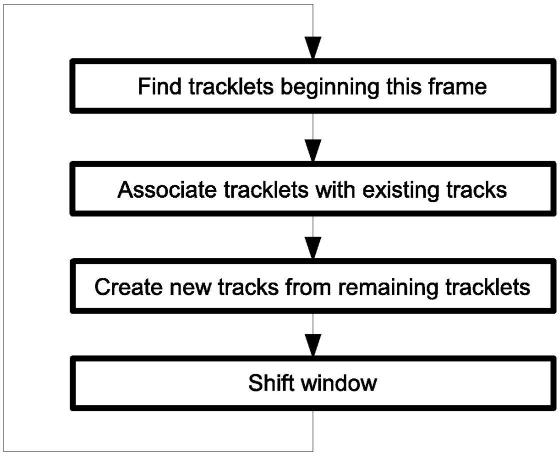

Find tracklets beginning this frame
Associate tracklets with existing tracks
Create new tracks from remaining tracklets
Shift window

Figure 3.6: The role of tracklets in multi-object tracking.

an associated appearance representation, such as the raw image patch, or a histogram. We would like to emphasize that our goal is to find valid tracklets within a window that shifts with each frame. Aggregation of these tracklets into tracks that span several windows is done by the (higher-level) tracking algorithm. Also if the detections of an object become split (or merged) for a period longer than the window size, this algorithm will find several (or one) tracklets in the window. This must be handled at the higher-level as well. Details of a multi-object tracker that takes care of these issues is not given here, but the current implementation is simple. A flow-chart that clarifies multi-object tracking using tracklets is in Figure 3.6.

We do not assume an a priori number of objects in the scene, and the number can vary over time. As a result, we assume that each detection in the first frame of the window is a potential object. Therefore, we find an optimal tracklet, or a set of tracklets, starting at each detection in the first window frame. This is not a problem, because for detections that are false alarms, the model of a valid tracklet (consistency of motion and appearance) is not satisfied, and the tracklet is discarded. Tracklets that start in the second or later frame of the window are found when the sliding window shifts to that frame.

3.1. Problem Formulation

If the initial detection of an object is given to us, we know there must be another detected instance of that object located “nearby” in subsequent frames. We are assuming there are no missed detections (due to occlusion or else) for now, but split detections, or split-merge detections, do not pose a problem, and this statement still holds. Therefore, the optimal tracklet, or a set of tracklets, that we want to find must be composed of a series of “nearby” detections. This can be expressed in a detection tree. For a window size of T frames, this tree would have T levels. A node in level t has links to those nodes in level $t + 1$, which are “nearby.” The root of the tree, $t = 0$, represents the initial detection.

The number of possible tracklets and tracklet combinations arising from this detection tree is huge, and we certainly do not want to evaluate every hypothesis. Instead, we realize that every such hypothesis is making a decision about including or not including each detection. In other words, this is just a binary labeling, or segmentation problem. The valid detections need to be separated from the invalid detections. The invalid detections are detections due to noise, or due to objects other than the one that generated the initial detection. One consequence of this view is that given the valid detections, it is not always known which tracklets generated them. For example, when there are multiple valid detections in several window frames, it could be that a single object generated them (and they correspond to multiple split/merge events), or it could be two (or several) objects being very close to each other. It may seem that nothing was gained by the segmentation, but actually solving this problem is easier than before, because the search space is significantly reduced.

There are several ways that the segmentation problem can be solved. A way that we pursue here (without further detail) is to determine the labeling in a probabilistic framework.

3.2. Tracklets from Detections

One way to solve this problem is to generate possible tracklets (hypotheses), and find zero or more that best explain the detections. Note that the search space is significantly reduced than before the segmentation, as we only need to explain the valid detections. Quite often there will be only one hypothesis.

The possible tracklets are generated by following certain parent pointers up to the root of the tree from each valid detection, and removing any tracklet that is a prefix of another. To determine which (combination) of these best explains the detections we first remove those tracklets that do not satisfy certain criteria. These criteria are:

- the number of detections in the temporal window must be at least half the window size
- the average acceleration of the object must be less than a threshold ($\approx 6\ m/s^2$)
- the object must be undergoing a smooth motion.

The second step is to merge tracklets that are a result of split-merge events. This is done by repeatedly merging tracklets that have similar appearance and motion.

3.3. Occlusion Handling

So far in the discussion we have assumed there is no occlusion or missing data.It turns out that when an object is occluded but the occluder is detected, the algorithm as presented still works. This is because the detection tree does not really change, except that no detections in the frame where the object is occluded will be valid. A tracklet is still found, provided that the object is not occluded for most of the window. In that case, the tracklet would fail the first criterion above.

The real problem that needs to be handled is missing detections. When there is a missed detection, the detection tree will be shorter than T. If it is too short, any tracklets that are found will not have enough detections and fail the first criterion above. This problem is solved by adding “virtual detections” to the detection tree. These are added whenever a detection in frame t has no nearby detections in frame $t+1$. The position of this virtual detection is estimated using the motion model, and the appearance is copied from the (detected) parent. This procedure is recursive, so that when a newly added virtual detection does not have nearby detections in the next frame, the process is repeated.

3.4. Experimental Results

We have evaluated the multi-object tracker on sequences captured from an airborne sensor. The sequences come from the publicly available CLIF 2006 dataset [1]. The video is captured at roughly 2 Hz, and it is in grayscale. As this is a large format video roughly 6600x7500 pixels, we chose 640x480 subregions over an expressway for the purposes of evaluation. The sequences were stabilized prior to tracking. All of the sequences used in evaluation are available for download.

The only moving objects in the video are vehicles, but they are in very low resolution. Each vehicle is only about 7x7 pixels, which makes detection and tracking quite challenging. Since this low resolution gives very limited appearance information, we used a simple sum of squared differences function as our appearance similarity. This is computed after doing a least-squares alignment of the two image patches. The alignment is parameterized by translation and rotation. In this context, this parameterization is satisfactory.

The moving object detection was done using background subtraction. The background is modeled as the mode of a (stabilized) sliding window of frames. We have also tried a mixture of Gaussian model, but we did not see a significant difference. A window size of 16 frames, corresponding to about 8 seconds of video was used.

Ground truth tracks were manually generated for an 80-frame sequence containing 168 vehicles. Tracks shorter than the window size were not used in evaluation (for a window size of 16, this left 123 tracks). Several metrics were used to measure performance: object detection rate (ODR), false alarm rate (FAR),

	ODR	FAR	NTF	IDC
With virtual det.	0.72	0.04	1.01	0.84
No virtual det.	0.61	0.03	1.04	0.90

Table 3.1: Effect of virtual detections on tracking performance.

Window Sz.	ODR	FAR	NTF	IDC
10	0.76	0.04	1.06	0.86
12	0.73	0.04	1.00	0.85
14	0.72	0.04	1.00	0.87
16	0.72	0.04	1.01	0.84
18	0.71	0.05	1.00	0.83
20	0.63	0.04	1.00	0.89

Table 3.2: Effect of sliding window size on tracking performance.

normalized track fragmentation (NTF), and ID consistency (IDC). The definition of the ODR and NTF is the same as in [121]. False alarm rate is the number of false detections divided by the number of total detections (NOT computed per track and averaged). The last metric is introduced to measure the tendency of the tracks to "jump" or switch IDs. To calculate this metric, we first label each detection in a ground truth track with the ID(s) of a test track(s), if any, which contains an overlapping detection. The ID consistency measures the largest fraction of detections in a ground truth track labeled with one label. Just as NTF, it is weighted by the length of the track to avoid favoring short tracks. More formally, let $G = \{g_i\}$ be the set of ground truth tracks, let $g_i = \{g_{ij}\}$ be the set of detections in each track, and $L(g_{ij})$ be the label of the test track associated with the detection g_{ij}. The ID consistency of a track g_i is

$$\text{IDC}_i = \max_l \frac{|\{g_{ij} \text{ s.t. } L(g_{ij}) = l\}|}{|g_i|} \tag{3.4}$$

and the overall ID consistency is

$$\text{IDC} = \frac{1}{\sum_{g_i} |g_i|} \sum_{g_i} |g_i| \text{IDC}_i \;. \tag{3.5}$$

The best ID consistency is 1.

Quantitative results using these metrics are shown in Tables 3.1 and 3.2. The first table shows the effect of using virtual detections in tracking. The second table shows the effect of different sliding window sizes on tracking. Qualitative results are shown in the following figures. Figure 3.7 shows tracking results on sample frames of the sequence used for evaluation.

4. Motion Pattern Analysis and Its Application to Detecting and Tracking Objects on a Moving Platform

We first address the general motion pattern analysis, and then discuss the specific property of the motion pattern created by moving vehicles in airborne videos.

Consider a 2D point P smoothly traversing in a spatio-temporal space. By projecting the motion of the point in a 4D space, (x, y, v_x, v_y), where (x, y) is the location of the point and (v_x, v_y) denotes the time derivatives of motion along x and y axes, we obtain a *fiber* (dimensionality is one) in the 4D space that represents the motion characteristic of that point. If a set of 2D points (*e.g.* on the same object) are moving in a similar way, a bundle of fibers form a *flow*. Many types of object motions can be represented by a flow. A single moving vehicle or a convoy of vehicles observed from an airborne camera are such typical cases.

Generally, we define the motion pattern in the 4D space as a set of motion vectors,

$$\mathcal{F} = \{(x, y, v_x, v_y), (x, y) \in R^2\}. \tag{3.6}$$

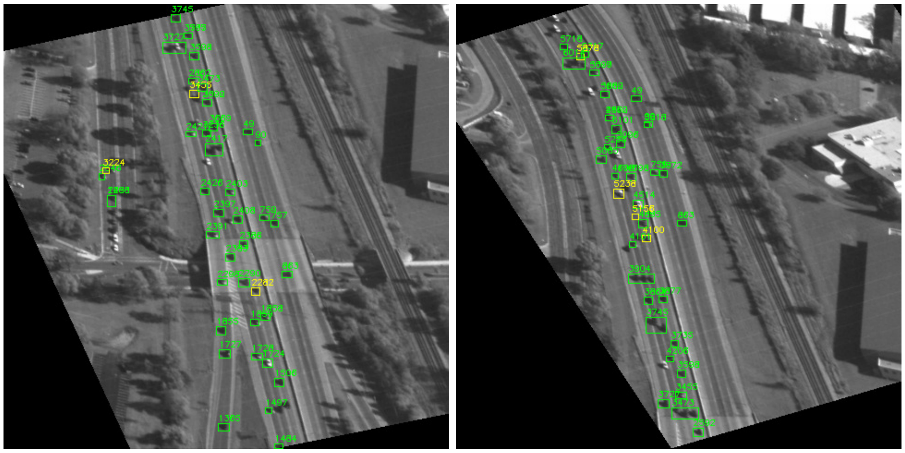
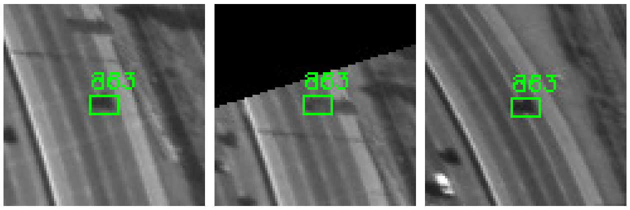


Figure 3.7: Tracking results on sequence 1. A green box denotes a real detection, whereas a yellow box denotes an interpolated detection.



(a) Frame 49 (b) Frame 51 (c) Frame 54

Figure 3.8: Example of occlusion handling.

Without loss of generality, in one motion pattern, one motion vector (v_x, v_y) is assigned at one location (x, y), *i.e.* (v_x, v_y) is a function of (x, y), $(v_x, v_y) = \mathcal{F}(x, y)$. Note that there may exist multiple motion patterns at the same location, (*e.g.* at a road intersection). Objects whose motion complies with the same motion pattern are called objects moving in the same *motion context*. A motion flow can be regarded as one particular type of motion pattern.

The motion pattern defined in Eq.3.6 essentially describes the general motion characteristics of objects over a period of time. In practice, the motion estimation of one object at a time inevitably contains noise. The estimated motion vectors in a motion pattern $\mathcal{F}$ can be written as $\mathcal{F} = \{(x, y, f(x, y) + e)\}$, where e accounts for the noise in motion estimation. We aim to analyze the general motion pattern from multiple noisy motion vectors over time, and then use this information to facilitate detection and tracking of each object in the motion pattern.

The essential property of a motion pattern is that *each smooth motion pattern corresponds to a smooth sheet in the 4D space*, *i.e.* the local dimensionality is 2. It is easy to know that the dimensionality is 2, since

- the projection of a motion pattern in (x, y) space is 2 in non-degenerate cases, thus the dimensionality in the 4D space is no less than two;

- at most one motion vector is assigned at one location, thus the local dimensionality is no more than two.

According to the smooth motion assumption, the normal on the sheet created by one motion pattern changes smoothly and different motion patterns produce discontinuity between them in 4D space. Noise caused by erroneous optical flows do not form a coherent sheet with local smoothness.

Suppose we have a set of noisy input samples in the 4D space, we aim to find smooth 2D sheets in this space. In order to analyzing the normal and tangent space at each point to infer the geometric structure while filtering noise out, we adopt Tensor Voting to achieve this task.

Tensor Voting [161] can be regarded as an unsupervised computational framework to estimate local geometric information. Tensor voting has been proved capable of estimating structures in N-D space with very noisy input data. In the Tensor Voting framework, the local geometric information at one point in N-D space is encoded in a symmetric, nonnegative definite matrix. The local geometry can be derived by examining its eigensystem. Recall that a tensor can be decomposed as

$$T = \sum_{i=1}^{N} \lambda_i e_i e_i^T = \sum_{i=1}^{N-1} (\lambda_i - \lambda_{i+1}) \sum_{k=1}^{i} e_k e_k^T + \lambda_N \sum_{k=1}^{N} e_i e_i^T \tag{3.7}$$

where $\{\lambda_i\}$ are the eigenvalues arranged in descending order, $\{e_i\}$ are the corresponding eigenvectors, and N is the dimensionality of the input space. The decomposition in Eq.3.7 provides a way to interpret the local geometry. The largest gap between two consecutive eigenvalues, $\lambda_i - \lambda_{i+1}$, indicates the dimensionality d,

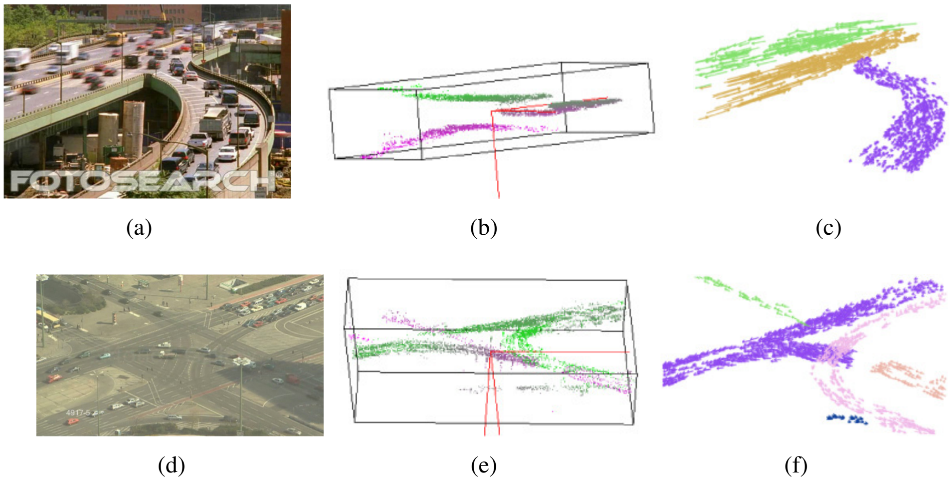
$$d = \arg\max_i (\lambda_i - \lambda_{i+1}) \tag{3.8}$$

The largest difference value $\lambda_d - \lambda_{d+1}$ is the saliency of the dimensionality. The corresponding eigenvectors $\{e_1, ..., e_d\}$ span the normal space of the structure, and $e_{d+1}, ..., e_N$ span the tangent space. In our case, we are interested in the structures whose normal space's dimensionality is 2 in the 4D space. Given the input data, a set of 4D motion vectors, $\{f_i\}$, we encode each sample as a ball tensor, which indicates no orientation as at the beginning we have no knowledge of the local structure at a point. Each f_i receives a vote $T_{j\to i}$ from its neighbors f_j in 4D space. The voting result at one point, which indicates its geometric property, is obtained by adding up all the incoming votes from its neighbors. The vote from a voter f_i to a receiver f_j encodes the tensor at f_i, the orientation and the distance from f_i to f_j. The result of this process can be interpreted as a local, nonparametric estimation of the geometric structure at each sample position. After accumulating all cast tensors, the local geometry can be interpreted according to of Eq.3.8.

To detect and segment motion patterns, we take original optical flows computed across multiple frames as input, without any pruning or clustering. After the voting process, we examine the cast tensor T and keep the structures of dimensionality 2 with saliency larger than a threshold. After this tensor voting process, most of the structures created by noise are filtered out. There may exist multiple motion vectors that belong to the same motion pattern over time, we use the average of the motion vectors to represent the estimated motion pattern. Note that, we only average motion vectors on the same sheet in the 4D space. This averaging is essentially different with prefiltering as it is performed after we have the knowledge of local structures. After detecting the desired structure and filtering out the noise, we use a flood-fill algorithm in 4D space to segment each motion pattern. According to the smooth motion assumption, the sheet formed by one motion context has local smoothness and discontinuity exists between sheets caused by different motion patterns. The neighbor samples in 4D space that have similar normal are assigned the same label. We use principal angles [23] to measure the similarity between two normal spaces. Two examples of motion pattern are shown in Figure 3.9. The video used in the second example is at a road intersection, where exist multiple motion patterns at some location. Directly smoothing in the (x, y) space will destroy such motion patterns.

We have discussed the properties of general motion patterns in 4D space. Now, we analyze motion patterns in airborne videos, where we aim to find those created by moving vehicles. The essential difference between motion patterns created by parallax and by moving vehicles is as follows. After principal motion estimation, the motion pattern of moving vehicles is generated by the intrinsic properties of the objects and static environmental constraints on the ground plane (*e.g.* road, or non-road area), which is independent of camera motion. The motion pattern of parallax is caused by the camera motion, as each motion vector on a 3D structure should be along the epipolar line that is determined by camera's translation. According to the relative affine structure [143], the projection p_t of a 3D point P on I_t can be decomposed as,

$$p_t = \underbrace{A_{t,r} p_r}_{(1)} + \underbrace{k e_t}_{(2)} \tag{3.9}$$



(a) (b) (c)

(d) (e) (f)

Figure 3.9: One example of motion pattern segmentation (a)(d) scenes with traffic flows (b)(e) motion pattern shown in (x, y, v_x) space (c)(f) segmentation of motion patterns

where p_r is the projection of P in the reference frame, k is a scalar, which is independent of the camera pose at time t, and e_t is the epipole at time t. The first term in Eq.3.9 is compensated for by the affine motion. From the second term, we can see the motion of parallax (ke_t) is indeed determined by the camera motion. Interestingly, when the epipole is moving in a non-smooth way, the motion of parallax cannot form smooth patterns, thus non-smooth epipole motion actually helps us to remove parallax. When the camera is moving in a smooth way, however, the parallax can still form a smooth motion pattern. Specifically, in airborne videos, the motion patterns of moving vehicles forms *flows*. Such flow shows a fiber property (dimensionality is 1) on a larger scale. This property is due to the fact that the motion range of a vehicle over time is much larger than its 2D dimensions. In order to examine the geometric property at a larger scale, we can simply enlarge the voting scale. In our experiments, we observe that, when we enlarge the scale of voting, the motion field caused by a small 3D structure becomes a point tensor or it remains a sheet for a large 3D structure. Thus, the procedure of segmenting the motion field created by moving vehicles is: first vote at a small scale and keep only the 2D structures to remove noise, and then vote at a large scale, keep only the 1D fiber structures. In practice, instead of directly enlarging the voting scale, we down-sample the 4D space to achieve an efficient implementation.

The motion flow created by moving vehicles may be fragmented due to occlusion. Thus, we propose a method to stitch them up. After we find each flow, we randomly place several “floats” (square 2D Gaussian kernels) in the flow and apply the meanshift like method used in along both positive and negative directions, the “floats” terminate at the ends of the flow. After we have the ends (both entry and exit) of the flows, we use a vote-casting method inspired from Tensor Voting to calculate the motion consistency between flows.

Given the flow information, detection and tracking becomes much easier. First, most of the residual pixels caused by noise and parallax have been filtered out. Second, the local dynamics in the motion field are known. In residual images along the flows, we adopt the motion history image method to segment independent motion regions. Each segmented region is represented as an oriented rectangle. An association score between regions R_i and R_j from neighboring ($|i-j| \leqslant \delta$) frames encodes both appearance similarity and consistency with the local motion field as:

$$p_{ij} = CS_{ij}e^{-\frac{|R(i)-\bar{R}_j(i)|+|R(j)-\bar{R}_i(j)|}{2}} \tag{3.10}$$

The appearance similarity S_{ij} is simply the normalized cross correlation between two image patches, $\bar{R}_j(i)$ is the predicted location from j to i by using $|i-j|$ steps of mean shift (the direction is $sign(i-j)$) in the motion field. According to this similarity measure, we aggregate these isolated regions from different frames into tracklets. We further filter out isolated regions and very short tracklets that come from noisy motion segmentation. For a pre-filtered tracklet, we use the average image patch of the oriented rectangles as its appearance template. A local translation relaxing is used to find the best matching location for averaging appearance template. The motion of a tracklet is encoded in its start and end points. Then, we apply the Hungarian algorithm to associate tracklets into tracks. Here, we encode the entry and exit information of a flow in the utility matrix used in the Hungarian algorithm. Suppose there are n tracklets in the pool, the utility matrix $\mathcal{A}_{2n\times 2n}$ is a matrix of size $2n \times 2n$. $\mathcal{A}_{(1,\cdots,n)\times(1,\cdots,n)}$, except its diagonal elements, contains the similarity between any pair of tracklets, the diagonal of $\mathcal{A}_{(n+1,\cdots,2n)\times(1,\cdots,n)}$ stores the termination probability of each tracklet, which is computed according to the distance between the end point of a tracklet and the exit of the flow; the diagonal of $\mathcal{A}_{(1,\cdots,n)\times(n+1,\cdots,2n)}$ stores the birth probability of each tracklet, which is computed according to the distance between the start point of a tracklet and the entry of the flow. All the other elements in $\mathcal{A}$ are zero. By expanding the similarity matrix, we impose the environmental information in the tracklet association to avoid the fragmented tracks in the middle of a flow. Note that the tracklet association is performed among flows that have been stitched in the motion pattern analysis phase.

The video shown in Figure 3.10, contains strong parallax and a convoy of vehicles passing through a forest where long term occlusions occur. This video challenges existing motion segmentation and tracking methods. The residual pixels that do not belong to valid motion patterns are shown in red in Figure 3.10. Such regions caused by parallax, which sometimes form larger regions than moving objects, cannot be filtered out by morphological operations or the motion history image method. By flow stitching, the long occlusion is correctly handled in the forest video. In Figure 3.10, we show the estimated the motion field after flow segmentation in both the mosaic space and the image space.

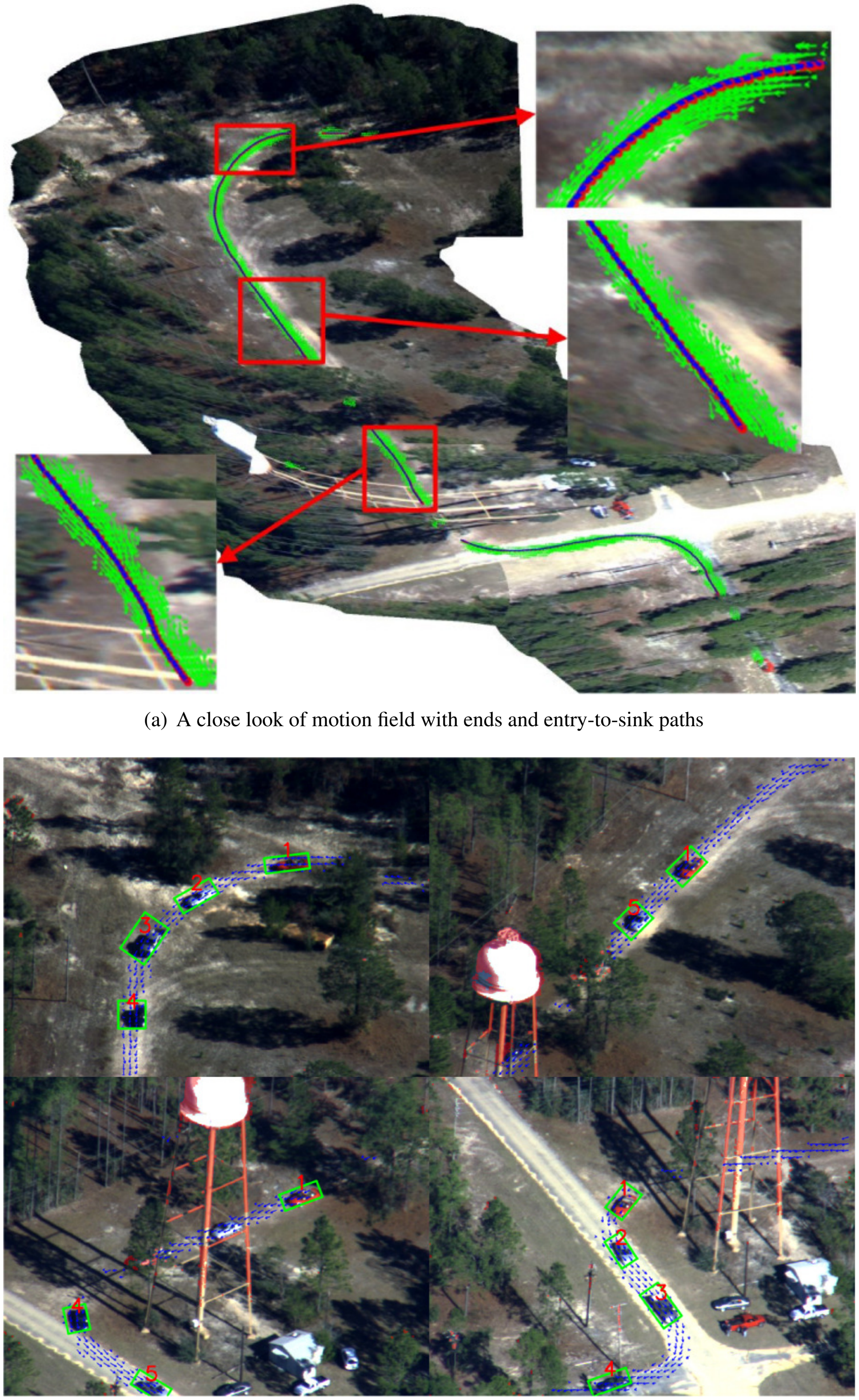
5. Online Appearance Modeling with Co-training of Hybrid Discriminative Generative Trackers

Tag and track problem by modeling the appearance of the target can be formulated in two different ways: generative and discriminative. Generative tracking methods learn a model to represent the appearance of an object. Tracking is then expressed as finding the most similar object appearance to the model. Several examples of generative tracking algorithms are Eigentracking [24], and IVT [91]. To adapt to appearance changes, the object model is often updated online, as in [91]. Due to the fact that the appearance variations are highly non-linear, multiple subspaces [84] and non-linear manifold learning methods [53] have been proposed.

Instead of building a model to describe the appearance of an object, discriminative tracking methods aim to find a decision boundary that can best separate the object from the background. Recently, many discriminative trackers are proposed [10], [33, 115] and demonstrate strong robustness to avoid distracters in the background. In order to update the decision boundary according to new samples and background, discriminative tracking methods with online learning are proposed in [10, 33]. Oza and Russell [116] proposed an online boosting algorithm, which is applied to the visual tracking problem [65, 94]. Due to the large number of features, either an offline feature selection procedure or an offline trained seed classifier is usually required in practice. Thus, for tracking methods based on online boosting, it is difficult to generalize to arbitrary object types.

We propose to use co-training to combine generative and discriminative models. Here, the online learning an appearance model of an arbitrary object with limited labeled data is treated as a semi-supervised problem. The co-training approach proposed by Blum and Mitchell [26] is a principled semi-supervise training method. The basic idea is to train two classifiers on two conditionally independent views of the same data (with a small number of exemplars) and then use the prediction from each classifier to enlarge the training set of the other. It is proved that co-training can find an accurate decision boundary, starting from a small quantity of labeled data as long as the two feature sets are independent [26].

We formulate the visual tracking problem as a state estimate problem. Given a sequence of observed image regions $O_t = (o_1, \dots, o_t)$ over time t, the goal of visual tracking is to estimate the hidden state s_t. In our case, the hidden state refers to an object's 2D position, scale and rotation. Assuming a Markovian state



(a) A close look of motion field with ends and entry-to-sink paths

(b) Snapshots of tracking multiple vehicles in the flows. Red indicates the residual pixels that do not belong to valid motion patterns.

Figure 3.10: Tracking with strong parallax

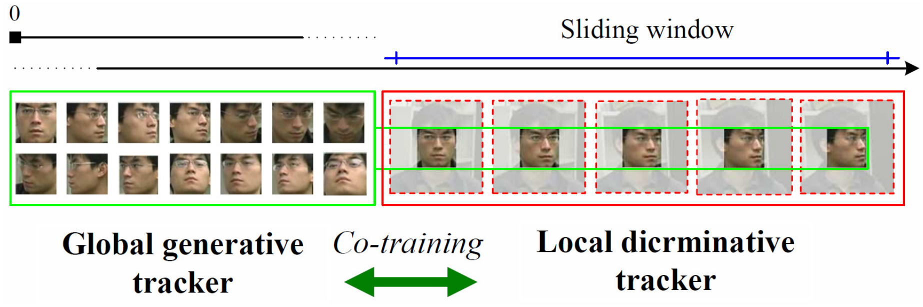


Figure 3.11: Online co-training a generative tracker and a discriminative tracker with different life span (the area bounded by dashed red boxes indicates the background)

transition, the posterior can be formulated as a recursive equation

$$p(s_t|O_t) \propto p(o_t|s_t) \int p(s_t|s_{t-1}) p(s_{t-1}|O_{t-1}) ds_{t-1} \tag{3.11}$$

where $p(o_t|s_t)$ and $p(s_t|s_{t-1})$ are the observation model and state transition model respectively. $p(s_{t-1}|O_{t-1})$, which is represented as a set of particles and weights, is the posterior distribution given all the observations up to time $t-1$. The recursive inference in Eq.3.11 is implemented with resampling and importance sampling processes. In our approach, the transition of the hidden state is assumed to be a Gaussian distribution as, $p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \mathbf{z}_{t-1}, \Psi_t)$, where Ψ_t is a time variant diagonal covariance matrix. In this recursive inference formulation, $p(o_t|s_t)$ is the crucial part for finding the ideal posterior distribution. $p(o_t|s_t)$ measures the likelihood of observing o_t given one state of the object. Besides the 2D position, our state variables encode an object's rotation and scale. This reduces the appearance variations caused by such motion at the price of that more particles are needed to represent the distribution.

Our measurement of one observation comes from two independent models. One is the generative model, which is based on online constructed multi-subspaces. The other is the discriminative model, which is online trained with HOG features. The features used by these two models, namely intensity pattern and local gradient features, are complementary. After limited initialization, these two models are co-trained with sequential unlabeled data. The final decision is made by the combined hybrid model. Due to the independence between the two observers, our observation model $p(o_t|s_t)$ can be expressed as a product of two likelihood functions from the generative $\mathcal{M}$ model and the discriminative model $\mathcal{C}$, $p(o_t|s_t) \propto p_{\mathcal{M}}(o_t|s_t) p_{\mathcal{C}}(o_t|s_t)$.

Let $\mathcal{M} = \{\Omega_1, ..., \Omega_L\}$ represent the appearance manifold of one object and $\Omega_l, l \in [1, ..., L]$ denote the local sub-manifold. An appearance instance $\mathbf{x}$ is a d-dimension image vector. Let $\Omega_l = (\hat{\mathbf{x}}_\mathbf{l}, U_l, \Lambda_l, n_l)$ denote one sub-manifold, where $\hat{\mathbf{x}}_l, U_l, \Lambda_l$ and n_l represent the mean vector, eigenvectors, eigenvalues and the size (number of samples) of the subspace respectively. For simplicity, we omit the subscript when this causes no confusion. Here, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with sorted eigenvalues of the subspace, $\lambda_1 \geqslant \lambda_2 \ldots \geqslant \lambda_n$. A η-truncation is usually used to truncate the subspaces, namely $m = \arg\min_i (\sum_i \lambda_i / \text{tr}(\Lambda) \geqslant \eta)$. From a statistical point of view, a subspace with m eigenbases can be regarded as a m-dimensional Gaussian distribution. Suppose Ω is a subspace with the first m eigenvectors, the projection of $\mathbf{x}$ on Ω is $\mathbf{y} = (y_1, \ldots, y_m)^T = U^T(\mathbf{x} - \hat{\mathbf{x}})$. Then, the likelihood of $\mathbf{x}$ can be expressed [107] as

$$p(\mathbf{x}|\Omega) = \left[\frac{\exp\left(-\frac{1}{2}\sum\limits_{i=1}^{m} \frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{m/2} \prod\limits_{i=1}^{m} \lambda_i^{1/2}} \right] \cdot \left[\frac{\exp\left(-\frac{\varepsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(d-m)/2}} \right] \tag{3.12}$$

where $\varepsilon(\mathbf{x}) = |\mathbf{x} - UU^T\mathbf{x}|$ is the projection error, namely L_2 distance between the sample $\mathbf{x}$ and its projection on the subspace. The parameter $\rho = \frac{1}{d-m}\sum_{i=m+1}^{d} \lambda_i$ [107] or uses the $\frac{1}{2}\lambda_{m+1}$ as a rough approximation.

By using Eq.3.12, we can evaluate the confidence of a sample from one subspace. As our generative model contains multiple subspaces (each subspace can be regarded as a hyper-ellipsoid), we maintain the neighborhood according to L_2 distance between the mean vectors of subspaces. To evaluate the confidence of one sample from such a generative model, we use the maximum confidence of the K-nearest (we use $K = 4$ in experiments) neighboring subspaces.

In order to represent a large number of sequential samples, we use a fixed number subspaces: if the number of subspaces exceeds a predetermined maximum, the most similar two subspaces are merged. In order to maintain the local property of the subspaces, merging only happens between neighboring subspaces. Merging of two subspaces and measuring the similarity between two subspaces are two critical steps in this algorithm.

Suppose there are two subspaces $\Omega_1 = (\mathbf{x}_1, U_1, \Lambda_1, N)$ and $\Omega_2 = (\mathbf{x}_2, U_2, \Lambda_2, M)$, which we are trying to merge to a new subspace $\Omega = (\bar{\mathbf{x}}, U, \Lambda, M+N)$. If the dimension of Ω_1 and Ω_2 are p and q, the dimension r of the merged subspace Ω satisfies: $\max(p,q) \leqslant r \leqslant p+q+1$. The vector connecting the centers of the two subspaces does not necessarily belong to either subspace. This vector causes the additional one in the upper bound of r.

It is easy to verify that the scatter matrix S of the merged subspace Ω satisfies, $S = S_1 + S_2 + \frac{MN}{M+N}(\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)^T$. We aim to find a sufficient orthogonal spanning of S. Let $h_1(\mathbf{x})$ denote the residual vector of a vector $\mathbf{x}$ on Ω_1, $h_1(\mathbf{x}) = \mathbf{x} - U_1 U_1^T \mathbf{x}$. Note that $h_1(\mathbf{x})$ is orthogonal to U_1, *i.e.* $h(\mathbf{x})'U = 0$. Now, $U' = [U_1, \mathbf{v}]$ is a set of orthogonal bases to span the merged space, where $\mathbf{v} = GS\left(h_1\left(U_2, (\mathbf{x}_2 - \mathbf{x}_1)\right)\right)$ and $GS(\cdot)$ denote the Gram-Schmidt process. Given the sufficient orthogonal bases, we can obtain the SVD decomposition of S.

$$U'^T S U' = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} G\Lambda_2 G^T & G\Lambda_2 \Gamma^T \\ \Gamma \Lambda_2 G^T & \Gamma \Lambda_2 \Gamma^T \end{bmatrix} + \frac{MN}{M+N} \begin{bmatrix} gg^T & g\gamma^T \\ \gamma g^T & \gamma\gamma^T \end{bmatrix} = R\Lambda R^T \tag{3.13}$$

where $G = U_1^T(\mathbf{x}_2 - \mathbf{x}_1)$, $\Gamma = \mathbf{v}^T U_2$, $g = U_1^T(\mathbf{x}_2 - \mathbf{x}_1)$ and $\gamma = U'(\mathbf{x}_2 - \mathbf{x}_1)$. Now, the eigenvalue of the merged subspace is Λ in Eq.3.13 and the eigenvector U is simply $U'R$. Note that incrementally updating a subspace with one observation as in [84] is one special case of merging two subspaces using Eq.3.13.

The other critical step is to determine the similarity between two subspaces. We use two factors to measure the similarity between two neighboring subspaces Ω_1, Ω_2, the canonical angles (principal angles) and the data-compactness.

Suppose the dimensions of two subspaces are $p, q, p \geqslant q$, then there are q canonical angles between the two subspaces. A numerical stable algorithm [23] computes the angles between all pairs of orthonormal vectors of the two subspaces as, $\cos\theta_k = \sigma_k(U_1^T U_2), k = 1, \cdots, q$, where $\sigma_k(\cdot)$ is the k^{th} sorted eigenvalue computed by SVD. The consistency of two neighboring subspaces can be represented as follows.

$$Sim_1(\Omega_1, \Omega_2) = \prod_{k=q-d_0+1}^{q} \sigma_k(U_1^T U_2) \;\; k = 1, \cdots, q \tag{3.14}$$

As the dimensionality of subspaces is larger than d_0, the initial dimension, we select the d_0 largest principal angles, which approximately measure the angle between two local subspaces. In a 3D space, the largest canonical angle between two 2D subspaces is equivalent to the angle between the two planes. In this case, we prefer to merge 2D patches with a small plane-to-plane angle. Note that the merge only happens between neighbor subspaces. The neighborhood is defined according to the mean vector L_2 distance. Merging subspaces with a small principal angle can avoid destroying the local structure of the appearance manifold.

The other factor to consider is data-compactness, which measures how much extra dimensionality is incurred by a merge operation. Suppose the dimension of two subspaces Ω_1, Ω_2 is $p, q, p \geqslant q$, the sorted eigenvalues of original merged subspace are $\Lambda_r = (\lambda_1, \ldots, \lambda_r), r = p+q+1$. The similarity based on data-compactness is defined as

$$Sim_2(\Omega_1, \Omega_2) = \sum\nolimits_{i=1}^{p} \lambda_i \Big/ \sum\nolimits_{i=1}^{r} \lambda_i \tag{3.15}$$

If Sim_2 is close to one, this indicates the merge operation does not incur any new dimension; on the contrary, if Sim_2 is small, this indicates the variations in Ω_1 and Ω_2 cannot use common eigenvectors to represent it. Combining the two factors in Eq.3.14 and Eq.3.15, the final similarity between two subspaces is defined in Eq.3.16.

$$Sim(\Omega_1, \Omega_2) = Sim_1(\Omega_1, \Omega_2) + w_d Sim_2(\Omega_1, \Omega_2) \tag{3.16}$$

where w_d is the weight to balance these two factors. We use $w_d = 0.2$ in experiments.

For the discriminative model, we adopt an incremental SVM algorithm, *LASVM* [27], to train a classifier between object and background. SVM [179] is able to form the optimal separating function, which reduces to a linear combination of kernels on the training data, $f(\mathbf{x}) = \sum_j \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}) + b$, with training samples $\mathbf{x_i}$ and corresponding label $y_i = \pm 1$.

In practice, this is achieved by maximizing the dual objective function $\max_\alpha W(\alpha)$ with

$$W(\alpha) = \sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

subject to

$$\sum\nolimits_i \alpha_i = 0, A_i \leqslant \alpha_i \leqslant B_i, \tag{3.17}$$

where $A_i = \min(0, Cy_i), B_i \max(0, Cy_i)$. Here, α is a vector of weights on y_i. A SVM solver can be regarded as updating α along some direction to maximize $W(\alpha)$. Let $g = (g_1, \ldots, g_n)$ denote the gradient of $W(\alpha)$

$$g_k = \frac{\partial W(\alpha)}{\partial \alpha_k} = y_k - \sum_i \alpha_i K(x_i, x_k) = y_k - \hat{y}(x_k) + b \tag{3.18}$$

LASVM suggests that optimization is faster when the search direction mostly contains zero coefficients. *LASVM* uses the search directions whose coefficients are all zero except for a single +1 and a single -1. The two non-zero coefficients, are called *τ-violating pair* (i, j) if $\alpha_i < B_i, \alpha_j > A_j$, and $g_i - g_j > \tau$, where τ is a small positive value. and *LASVM* selects the τ-violating pair (i, j) that maximizes the directional gradient $g_i - g_j$.

We compare our co-trained tracker with two generative methods, including (G1) IVT [91] and our multiple linear subspaces (G2) algorithm and three discriminative methods, including online selection of discriminative color (D1) [33], our online SVM method (D2) and ensemble tracking (E.T) [10]. G1 uses a single 15D linear subspace and updates it incrementally. Note that D1 does not consider tracking with large scale change and rotation. G1, G2, D2 and the co-trained tracker use the same parameters in CONDENSATION algorithm, but G1, G2 and D2 use self-learning to update their models. We compare these methods on challenging data sets, which contain image sequences of various types of objects. Detailed comparison can be found in Table 3.3 [193]. In experiments, we frequently find that the co-trained tracker has better self-awareness of current tracking performance and can safely enlarge the search range (by changing the diffusion dynamics) without being confused by distracters in the background. Also, the co-trained tracker successfully avoids drifting caused by varying viewpoints and illumination changes.

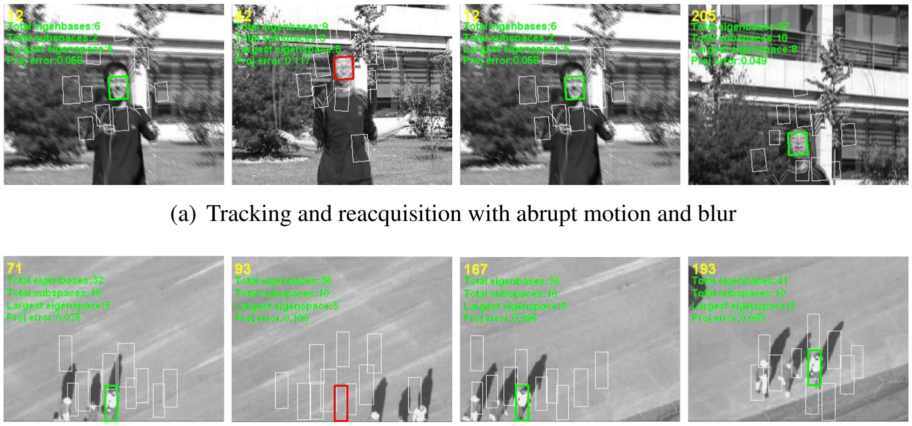
Part of the visual results are shown in Figure 3.12.

6. Co-trained Particle Filter Framework for Robust Object Tracking

Even though our proposed co-trained trackers obtain very good results, the speed of the system is not applicable for using in real-time environment. To improve it, we propose a new co-training framework using a cascade particle filter to label incoming data continuously and online update hybrid models generatively and discriminatively. Each of the layers in the cascade contains one or more either generative or discriminative appearance models. The cascade manner of organizing the particle filter enables the efficient evaluation of multiple appearance models with different computational costs; thus improve the speed of the tracker. The

	Frm	Occluded	G1		G2		D1		D2		Our method	
	No	Frms/ Times	FF	TR/LR	FF	TR/LR	FF	TR/LR	FF	TR/LR	FF	TR/LR
Seq1	761	0/0	17	0/0	261	0/0	n/a	n/a	491	0/3	**749**	**0/1**
Seq2	313	0/0	75	0/2	282	0/3	9	0`/0	214	0/2	**295**	**0/1**
Seq3	184	30/1	50	0/0	50	0/0	25	0/0	50	0/0	**154**	**1/0**
Seq4	338	93/2	33	0/0	70	0/1	33	0/0	72	0/1	**230**	**2/1**
Seq5	140	0/0	11	0/0	15	0/0	6	0/0	89	0/3	**140**	**0/0**
Seq6	945	143/4	163	1/0	506	1/0	382	0/0	54	0/1	**798**	**4/0**

Table 3.3: Comparison of different methods G1:IVT [91], G2: incremental learning multiple subspaces, D1: online selection of discriminative color features [33], D2: online SVM, E.T: ensemble tracking [10]. D1 uses color information, which is not available for Seq1 and Seq6.



(a) Tracking and reacquisition with abrupt motion and blur

(b) Tracking and reacquisition with long leaving out of field of view

Figure 3.12: Tracking various type of objects in outdoor environments

proposed online framework provides temporally local tracking that adapts to appearance changes. Moreover, it provides an object-specific detection ability that allows to reacquire an object after total occlusion.

We formulate the co-training in the last stage of our cascade particle filter. This layer is the most important layer where the final result is given through a co-decision process. This result is then used to update all of the other models in other layers (as shown in Figure 3.13). Due to the flexibility of our framework, there are some obvious advantages.

Fast running time: As mentioned above, when sampling, many of samples have low confidence score (close to zero) because they are not related to the object of interest. To improve the performance, the cascade particle filter is applied in the way that cheaper models will be put in the very first layers while more expensive models are arranged in the latter layers. This configuration efficiently boosts the speed of the whole framework to achieve real-time performance.

General object tracker: Considering tracking as a semi-supervised problem, when we only have very limited labeled data without an explicit offline-trained model like in [90], the idea is to build an appearance model on-the-fly to adapt with all the changes in viewpoints, lighting conditions,... during tracking. To fulfill this task, we adopt co-training as the framework to enhance the powers of different models. In online training, it is hard for one model to train on its own because the early mistakes could *reinforce* themselves while every model will have its own weakness. The best solution is that different models can cooperate with each other to produce a strong one even when many of them fail at one point of time. Co-training is an ideal framework to address this issue.

In cascade particle filter, our co-training is set up in the last stage to give the final decision in the tracking framework, while all of other stages determine if a sample is good to go through the next layer or

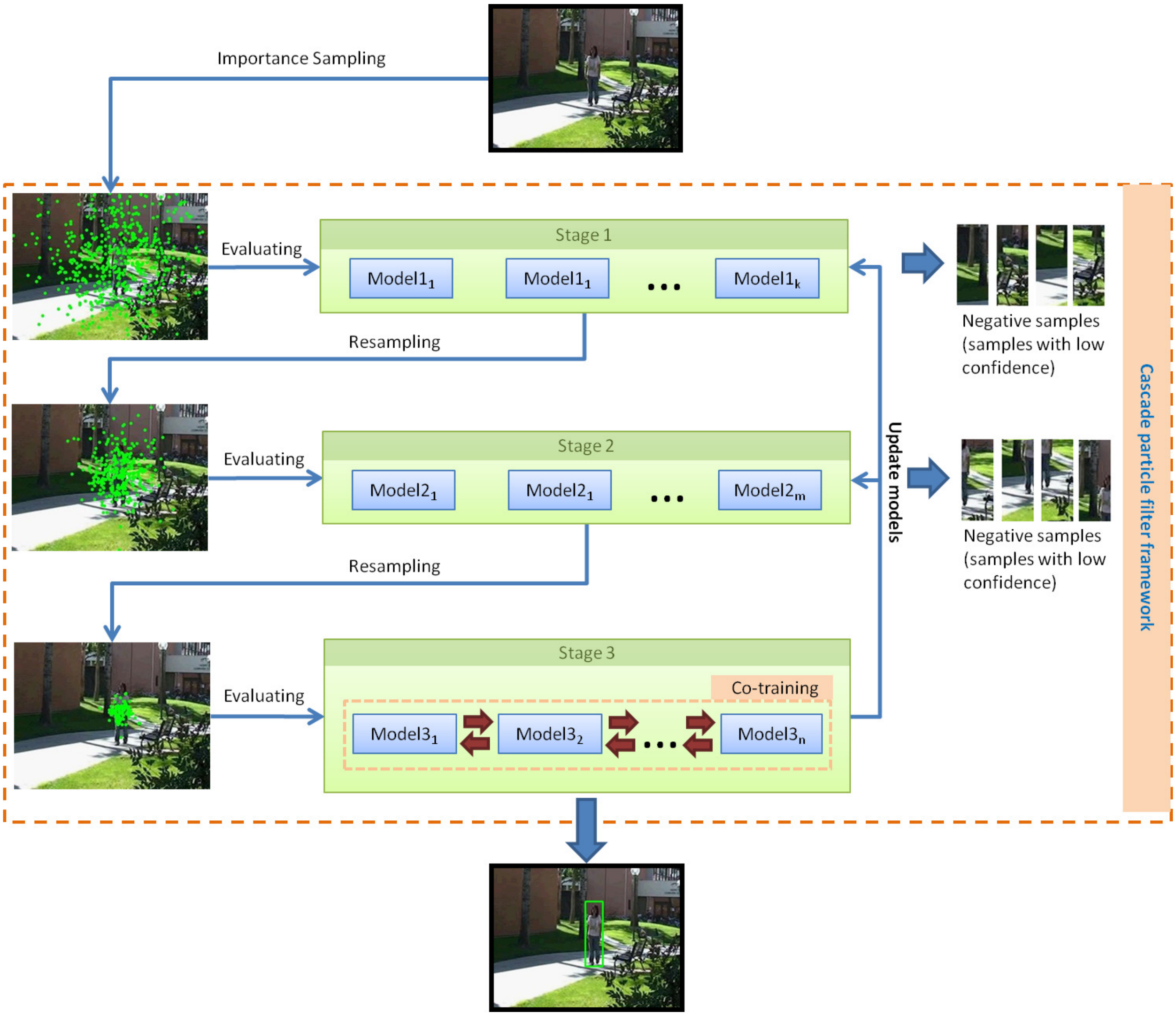


Figure 3.13: Our proposed framework: a co-trained cascade particle filter (CCPF) with three stages.

not. Different from the proposed framework in [90], ours naturally extends to use multiple models in each stage of the cascade to enhance their complementary powers.

Combination of different models There often lies a choice between which direction to go in building a model: generative or discriminative. The goal of the generative model is focusing on describing the appearance of the object while the discriminative model aims to find a way to separate the object and the background. As discussed in Section II, discriminative models usually perform better than generative ones. It is natural because it is always harder to “describe” the appearance of an object X than to “tell” what is different between it and others. However, each of the models has its own advantages and disadvantages. Our framework opens an easy way to incorporate both of them and strengthen their powers logically.

Moreover, in practice, using one type of features may not be enough to produce a robust tracker. For example, to track a pedestrian, not only the curves around the body may be of interest, but also the colors of the clothes and the textures on the whole pedestrian body are very important. To improve performance, we propose to use different types of features: color, local patch, and object template.

Strong reacquisition: It is worth to emphasize that the outcome of our framework is not only the tracking result but a detector for that specific object as well, which is extremely useful for reacquisition. For instance, when the object leaves the field of view and comes back, without any knowledge about exit/entry points, the only way is doing exhaustive search over the frame to find our object. With the learned model, our tracker can find the object easily. Total occlusion is also considered as the same case.

Experiment settings: The framework can run by taking some initial limited data as input. Ideally, the object of interest is tagged from the 5-10 very first frames. Also, a simple template matching method is implemented to help initializing the tracker when the object is provided only in the first frame. We adopt conservative evaluation criteria in decision making for each model. The framework is implemented as follows. Given the labeled data in the first few frames, we initialize all of the models independently. When tracking starts, all particles are sequentially passed through the cascade particle filter which consists of multiple stages. At each stage, the models simultaneously evaluate all the samples and co-decide to filter out a number of negative samples then propagate the rest to the next stage. At the final stage, a co-training process takes place to help each model train each other. In our implementation, we adopt one generative and one discriminative model in this stage.

Comparative analysis *Co-trained tracker vs. others:* We compare our co-trained tracker with two generative methods, which are Incremental Visual Tracking (IVT) [139], fragment-based tracking (FT) [2], and three discriminative methods, including online selection of discriminative color feature (OSDC) [33], ensemble tracking (ET) [10], and multiple instance learning tracking (MIL) [11]. In IVT, we use a single 15D linear subspace and update it incrementally.

Video Sequence	Frames	Occlusion	IVT	OSDC	ET	FT	MIL	**Ours**
Dark	761	0	17	n/a	94	7	230	**759**
Jumping	313	0	75	313	44	263	313	**313**
Handheld	140	0	11	6	22	46	86	**140**
Vehicle	945	143	163	n/a	10	547	202	**802**
UAVperson1	184	30	50	50	53	52	154	**154**
UAVperson2	338	93	33	8	118	92	32	**240**

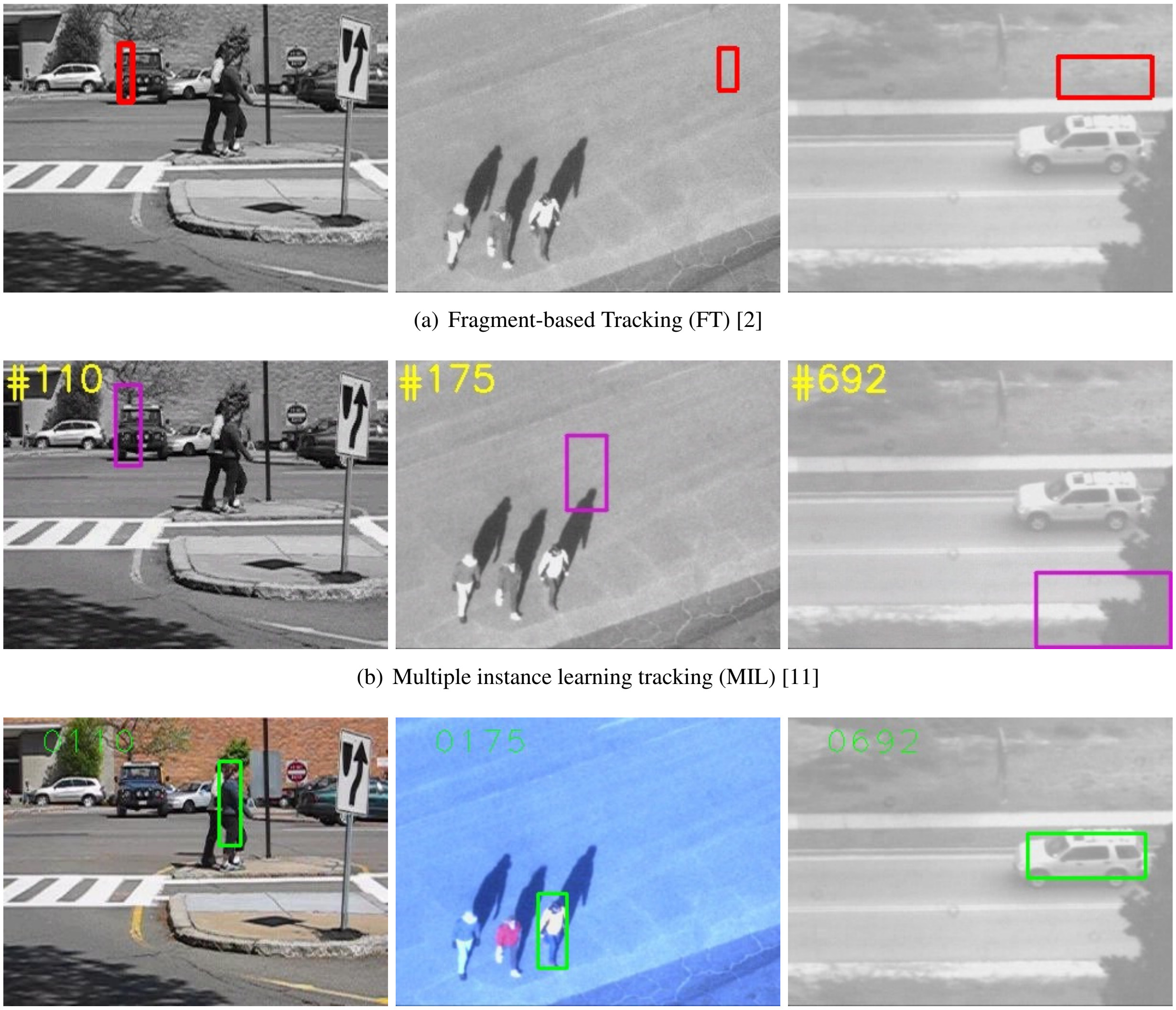
Table 3.4: Comparison of different methods IVT: incremental visual tracking [139], OSDC: online selection of discriminative color features [33], D2: online SVM, ET: ensemble tracking [10], FT: robust fragments-based tracking using the integral histogram [2], MIL: Visual Tracking with Online Multiple Instance Learning [11] and our co-trained tracker. D1 uses color information, which is not available for sequences “Dark” and “Vehicle”.

The comparison demonstrates that the co-trained tracker performs more robustly than other methods. Note that OSDC requires color information, thus it cannot process some sequences, which are indicated as “n/a”. The visual results are shown in Figure 3.14, where the tracked objects are bounded with green boxes. The red box indicates none of the models is updated in this frame. In experiments, we frequently find that the co-trained tracker has better self-awareness of current tracking performance and can safely enlarge the search range (by changing the diffusion dynamics) without being confused by distracters in the background. Also, it can successfully avoid drifting caused by varying viewpoints and illumination changes.

Tracker	Average Center Location Error (pixels)	Running Time (fps)
One-layer	4.38	**4.25**
Two-layer	4.93	**9.34**
Three-layer	5.74	**14.79**

Table 3.5: Comparison between cascade setup and non-cascade setup in terms of precision and running time.

Cascade vs. non-cascade: To demonstrate the efficiency of using cascade particle filter, we set up three trackers which have one layer, two layers, and three layers, respectively. The one-layer tracker indeed uses our proposed co-trained models. The two-layer tracker contains the co-trained tracker and. After the first layer, half of the samples are kept; while after the second layer, 25% of the samples are preserved. Therefore,



(a) Fragment-based Tracking (FT) [2]

(b) Multiple instance learning tracking (MIL) [11]

(c) Our co-trained tracker

Figure 3.14: Some visual comparative results between our co-trained tracker and other trackers in several sequences ‘Handheld”, “UAVperson2”, and ‘Vehicle” (from left to right).

with the sampling of 600 particles, 300 remaining particles are evaluated by the final stage of the two-layer tracker; whereas 75 best particles are evaluated in the last stage of the three-layer tracker.

The performance of these trackers is compared based on the average center location error and the running time, which is shown in Table 3.5. The six sequences used in the previous experiment are adopted. The results clearly show that cascade particle filter improves the running time performance by a large margin, while producing comparable robust results.

Reacquisition performance: One of the main advantages of our tracker is the reacquisition ability which is not well explored by other state-of-the-art algorithms. Our method not only ensures the quality of long-term tracking results but also provides the capability to re-detect an observed object after it leaves the field of view. To evaluate this reacquisition ability, we synthesize the leaving out of the field of view behavior of an object. We create a new sequence after deleting several frames from the original one. We start from the beginning of the sequence (after ignoring some initial frames for the learning phase), for each step of 50 frames, we replace 30 continuous frames by a synthetic background image. Note that each step forms a new sequence. 109 new sequences are thus created. Some examples of the new data sets are shown in Figure 3.15.

To evaluate the reacquisition ability, we count the number of tracked (reacquired) cases, the number of missed cases, and the number of false alarm cases. If the tracker can reacquire the object within at most 10 frames after the object re-appears, and the overlapped region between the tracked and the ground-truth windows is larger than 50% of the ground-truth window, it is counted as tracked case. The details of the results are shown in Table 3.6.

Video Sequence	No of cases	No. of tracked cases	No of missed cases	Number of false alarms
Clutter	29	29	0	0
Handheld	3	3	0	0
Jumping	5	5	0	0
Scale	37	24	9	4
Sylvester	26	20	4	2
UAVperson1	4	4	0	0
UAVperson2	5	5	0	0
Total	**109**	**90**	**13**	**6**

Table 3.6: Reacquisition performance of our tracker.

It can be observed that most of our missed cases come from the two sequences “Scale” and “Sylvester”. This is because they contain extremely difficult cases where the object appearance changes differently in terms of scale, pose, lighting while the synthesized leaving-out-of-field cases happen.

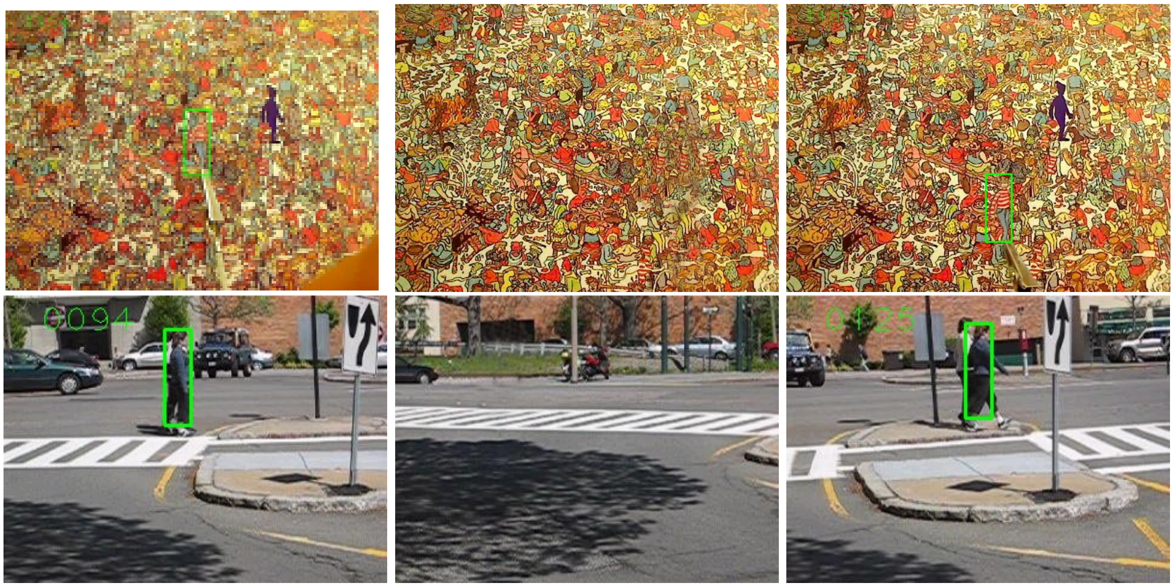
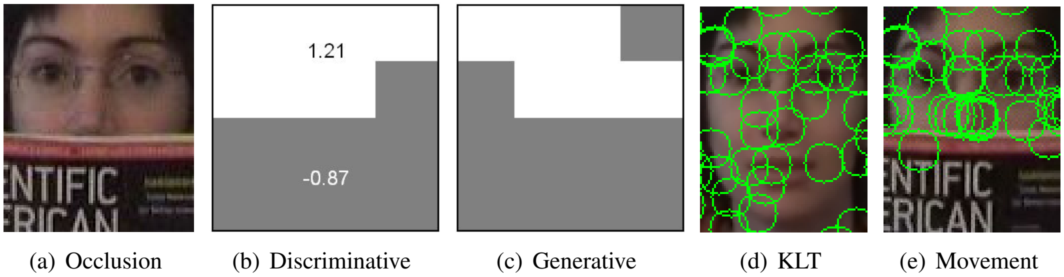


Figure 3.15: Synthesized data sets for testing reacquisition. From left to right: before leaving-field-of-view, synthesized background, reappeared object.

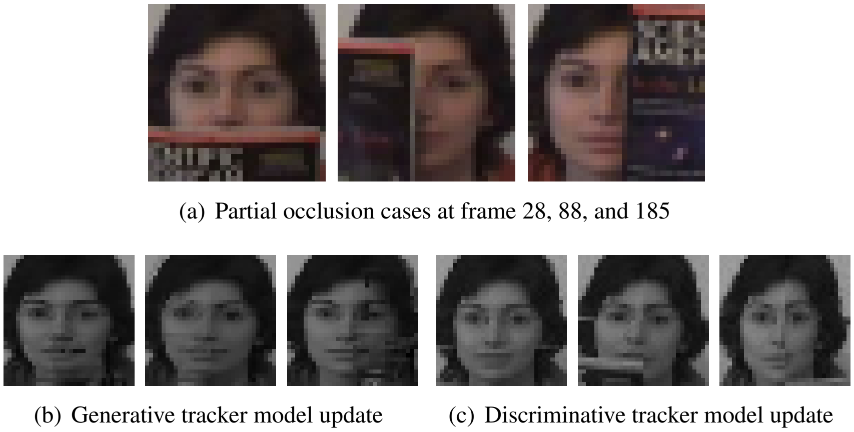
7. Partial Occlusion Handling in Object Tracking

In our previous proposed methods, we did not address the occlusion issue explicitly but use a threshold to control the update process so that the appearance models are only updated when there is no occlusion and vice versa. However, to define such a threshold is not straightforward. There is a trade-off between a “conservative” update or an “easygoing” update in order to adapt to the appearance change. To address this issue, we propose improve our current approach by proposing to use a co-training framework of generative and discriminative trackers to detect occluding region and continuously update both models using the information from the non-occluded part. The generative model encodes all of the appearance variations using a



(a) Occlusion (b) Discriminative (c) Generative (d) KLT (e) Movement

Figure 3.16: Partial occlusion observations on our trackers and KLT feature movement



(a) Partial occlusion cases at frame 28, 88, and 185

(b) Generative tracker model update (c) Discriminative tracker model update

Figure 3.17: Occlusion recovery from our trackers (image is scaled to 32x32 for training)

low dimension subspace, which helps provide a strong reacquisition ability. Meanwhile, the discriminative classifier, an online support vector machine, focuses on separating the object from the background using a Histograms of Oriented Gradients (HOG) feature set. To detect occlusion, a likelihood map is generated by the two trackers through a co-decision process. To handle the cases when there is disagreement between these two trackers, the movement vote of KLT local features [145] is used as a referee. Finally, each tracker recovers the occluded region and updates the models using the new non-occluded information.

To detect the occlusion, each model estimates the occlusion likelihood of each block in a sample and makes the decision together. The KLT features are also generated and tracked in order to determine when occlusion happen through a movement voting process. Some observations obtained from generative model, discriminative model, and KLT tracker are shown in Figure 3.16 when occlusion appears.

7.1. Generative Tracker

We propose to use a single linear subspace to approximate the appearance model of the object. This is close to the incremental visual tracker (IVT) [139], but with partial occlusion handling. To detect partial occlusion, as discussed in Section 2, the projection error is split into blocks as done in the discriminative tracker. We simply compute the occlusion likelihood by using the projection error over each block. The projection error at block i^{th} is:

$$\varepsilon_i(\mathbf{x}_i) = |\mathbf{x}_i - UU^T\mathbf{x}_i| \tag{3.19}$$

These likelihood values are then used to generate a binary occlusion likelihood map. The 0 value corresponds to the block having score lower than 50% of the maximum score block, and 1, otherwise.

7.2. Discriminative Tracker

We also compute the classifier score on each block instead of the whole sample patch to infer where the partial occlusion occurs. The decision function of conventional SVM is

$$f(x) = \beta + \sum_{k=1}^{n_{sv}} \alpha_k K(x, x_k), \tag{3.20}$$

where $x_k : k \in \{1, 2, ..., n_{sv}\}$ are the support vectors. $K(x, x_k)$ is the kernel function; and β is the bias constant. Here, a linear kernel is used, which means $K(x, x_k)$ is the inner scalar product of two vectors in $\Re^n$. Now we have to distribute the bias constant β to each block B_i; so that the contribution score of each block in the final classifier confidence score can be computed after subtracting that local bias β_i from the total feature inner production over that block.

We can have the distribution of bias constant on each block:

$$\beta_i = B.\tilde{w}_i^T . \left(A \sum_{p=1}^{N^+} B_{p;i}^+ + \sum_{q=1}^{N^-} B_{q;i}^- \right) \tag{3.21}$$

The occlusion likelihood map is generated as a binary image based on the block score, which is 0 for negative and 1, otherwise. Each pixel in the likelihood image corresponds to a block in the sample. To recover from the occlusion, the non-occluded part is kept while the occluded area is inferred from a previous frame (as shown in Figure 3.17(b)). In a long-term partial occlusion, we can consider this step as a recursive process where the occluded area of the object in the current frame is projected from that of the object in the previous frame, which may also be drawn from its previous one.

7.3. Local Features Movement Voting Using KLT

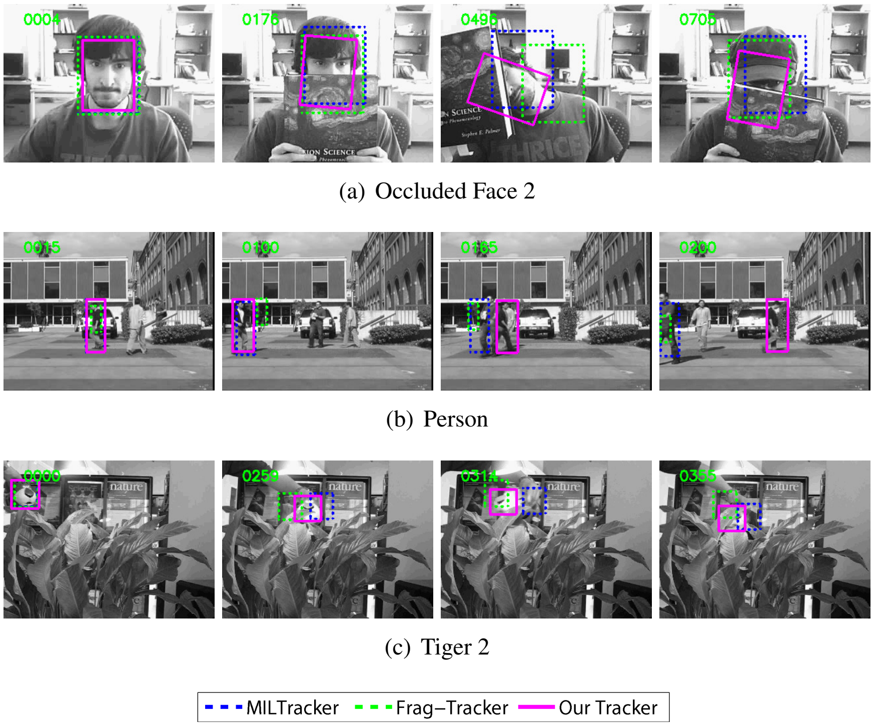
Taking advantage of the simplicity and fast computation of KLT features [145], tracking consistency is checked based on the movement of these features in the object region at every frame. Due to the discontinuity between non-occluded and occluding regions, some KLT features are driven in the same direction and velocity which are different from the remaining part. Taking account this observation, we propose a voting scheme on the movement of these local features to detect occlusion.

After being detected in the first frame, these features are tracked in every frame. After removing all of the outliers, the magnitude displacement of each feature is then normalized to [0,1] and encoded in a 4-bin histogram. The direction of the movement is encoded in a 8-bin histogram, each of which covers a $\frac{\pi}{4}$ span. All displacement vectors, thus, are accumulated into a 4x8 2D histogram.

When there is a majority of KLT features in a region having different movement behavior than the rest, partial occlusion is detected. In practice, we choose $\theta = 0.7$. It is important to note that the KLT features are re-initialized after occlusion and this step is only applied as a referee when there is disagreement on occlusion detection between the two generative and discriminative trackers.

7.4. Experiments

In the first frame, we manually select the object and apply simple template matching for the next 4 frames. These initial labeled data are then transferred to both generative and discriminative trackers for training. Our Bayesian framework generates 600 particles at each frame. The combined tracker is implemented in C++ and runs at 4fps on an Intel QuadCore 3.0GHz system. We tested our algorithm on several challenging published video sequences of different types of objects in indoor and outdoor environments. Several related state-of-the-art trackers included in the comparison are the Co-Tracker [193], which is the most related to our tracker, the Frag-Tracker [2], the Online and Semi-Boosting Tracker (OAB, SB) [62, 64], the P-N Tracker (PNT) [75], the MILTracker [11] and its new variation with no regret MIO Tracker [89]. We use the



(a) Occluded Face 2

(b) Person

(c) Tiger 2

Figure 3.18: Some screen shots from the testing results. Because of clarity issue, we only choose Frag-Tracker[2], MILTracker[11] to show some results comparing with our tracker.

provided results and published source code from the authors[1,2,3] . To prove the precision of our tracker, we used the same measurement, average center location errors (in pixels), used for evaluation in [11, 89]. All of the testing sequences provide long-term and heavy partial and total occlusions, and challenging appearance changes such as illumination changes, abrupt motion, rotation, and cluttered backgrounds.

Video Sequence	Frames	GT	DT	FT	OAB	ST	PNT	MILT	MIO	CoT	**Ours**
Coke Can	292	102	9	67	25	85	**8**	21	22	10	**8**
Occluded Face 1	900	86	17	*7*	44	41	8	27	14	16	**5**
Occluded Face 2	808	14	12	21	21	43	*8*	20	13	12	**7**
Person	200	35	73	44	37	154	44	34	n/a	*33*	**5**
Tiger 1	354	52	6	40	35	46	13	15	24	*5*	**4**
Tiger 2	365	43	*7*	37	34	53	21	17	23	*7*	**5**

Table 3.7: Average center location errors. (GT: Generative Tracker, DT: Discriminative Tracker, FT: Frag-Tracker [2], OAB: Online Boosting Tracker [62], ST: Semi-Boosting Tracker [64], PNT: P-N Tracker [75], MILT: MILTracker [11], MIO: MIL No Regret Tracker [89], CoT: Co-Tracker [193]) in different challenging datasets. The best performance is in bold, the second best is in italic.

8. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments

A major research axis has been focused on building a strong model to encode the variations of object appearance while distinguishing it from the background. By doing this, a fundamental dilemma occurs: the more complex the appearance model, the more expensive it is. At the extreme, the emergence of cluttered back-

[1] MILTracker: `http://vision.ucsd.edu/~babenko/project_miltrack.shtml`

[2] Frag-Tracker: `http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm`

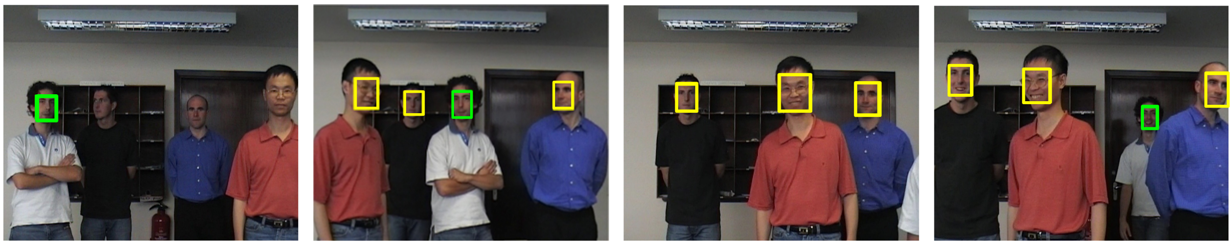
[3] Semi-boosting Tracker: `http://www.vision.ee.ethz.ch/boostingTrackers/index.htm`

ground and the occurrence of regions having similar appearance as the target makes appearance modeling is very challenging.

In fact, there is additional information which can be exploited instead of using only the object region. Context information has been applied actively in object detection [45], object classification [88, 114], object recognition [138]. It has been employed recently in several tracking methods [63, 191]. One reason for context being overlooked is the fast run-time requirement. Also, visual tracking, especially single object tracking, is considered as a semi-supervised problem where the only known data is the object bounding box in the first frame (or in first few frames), which means learning such a context needs to be performed on-the-fly.

Here, we propose to exploit the context information by expressing it in two different terms: 1) *Distracters* are regions that have similar appearance as the target, 2) *Supporters* are local key-points around the object having motion correlation with our target in a short time span. Supporters occur in regions belonging to the same object as the target, but are not included in the initial bounding box. In other words, the goal of our algorithm is to find all possible regions which look similar to the target to prevent drift, and to look for useful information around the target to have strong verification. The target and distracters are detected using shared sequential randomized ferns [117]. They are represented by individual evolving templates. The supporters, on the other hand, are represented as keypoints, and described using descriptors of the local region around them.

8.1. Distracters



(a) Initialization (b) Exploit all other distracters (c) Tracking the distracters even when the object leaves FoV (d) Reacquire the target without confusion with other distracters

Figure 3.19: Automatically exploiting distracters. Target is in green, distracters are in yellow.

Distracters are regions which have appearance similar appearance to the target and consistently co-occur with it. Usually, distracters are other moving objects sharing the same object category as our target (Figure 3.19). To build an appearance model to distinguish objects of the same type is equivalent to develop a recognition system which needs a large amount of supervised samples to train. However, in visual tracking, the tracker has temporal and spatial information help to exploit which region is *"dangerous"* to preclude. To prevent our tracker from drifting to these regions, we propose to detect and initiate a simple tracker for each of them so that we can minimize confusion during tracking.

Due to the efficiency of randomized ferns classifier, which is widely used in recognition [28, 118], and tracking [75], we employ it to detect possible distracters in every frame. Randomized ferns were originally proposed by Özuysal *et al.* [117] to increase the speed of randomized forest [29]. Unlike tree-structure in randomized forest, ferns, having non-hierarchical structures, consist of a number of binary testing functions. In our case, each of them corresponds to a set of Binary Pattern features. Each leaf in a fern records the number of added positive and negative samples during training. For a test sample, its evaluation by calculating the binary pattern features leads it to a leaf in the fern. After that, the posterior probability for that input testing sample in feature vector x_i to be labeled as an object ($y = 1$) by a fern j is computed as $Pr_j(y = 1|x_i) = p/(p + n)$, where p and n are the number of positive and negative samples recorded by that leaf. The posterior probability is set to zero if there is no record in that leaf. The final probability is calculated by averaging the posterior probabilities given by all ferns:

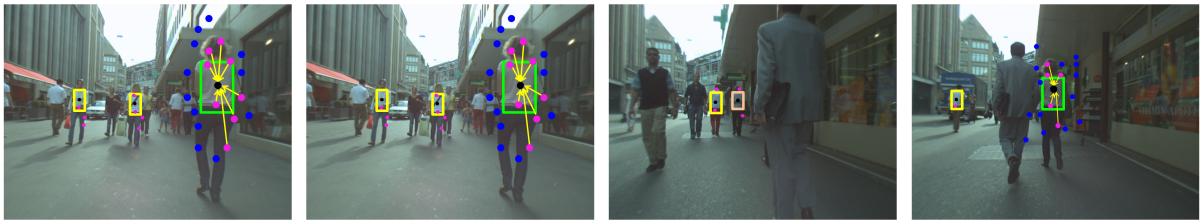
$$Pr(y = 1|x_i) = \sum_{1}^{T} Pr_j(y = 1|x_i) \tag{3.22}$$

where T is the number of ferns. To improve the running time, these randomized ferns are shared between our object detector and distracter detector. Each tracker controls the posterior probability by adding its positive and negative samples to the ferns according to the P-*constraints* and N-*constraints*, respectively as in [75]. The P-*constraints* force all samples close to the validated trajectory to have positive label, while N-*constraints* have all patches far from the validated trajectory labeled as negative. Different from [75], we avoid adding hard negative samples to avoid over-fitting. Also, during tracking, when the appearance of a distracter is different from our target, we discard it. Indeed, it helps to emphasize that our focus is on tracking a single target, not on multiple target tracking. This clearly explains the intuition: when several objects have similar appearance to our object, the target tracker pays attention to them; if these distracters change their appearance and no longer look like our object, they can be ignored.

Therefore, a sample is considered a distracter candidate if it passes the random ferns with a probability $Pr(y = 1|x_i) > 0.5$, and is not the target. How to determine a candidate as our target is discussed in Section 4. We maintain an M frames sliding window and count the frequency fd_k of a candidate k based on its appearance consistency spatial consistency related to the target. Then a candidate is classified as a distracter as follows

$$P_d(y_d = 1|x_i) = \begin{cases} 1 & if \qquad fd_k > 0.5 \\ & and \qquad d(x_i, M) > 0.8 \\ 0 & otherwise \end{cases} \tag{3.23}$$

where $P_d(y_d = 1|x_i)$ is the probability for a candidate i in a feature vector x_i having label y_d, while



(a) Detecting all supporters (b) Learning the active supporters (c) Avoid drifting to other regions while target is under occlusion (d) Reacquire the target with strong supporter model

Figure 3.20: Learning supporters. Active supporters are pink dots, passive supporters are in blue dots, object center is black dot.

$d(x_i, M)$ is the confidence of this candidate evaluated by the template-based model of the target. The first condition allows to detect distracters which repeatedly co-occur with our target, while the second one helps to exploit distracters having very similar appearance to our target.

8.2. Supporters

We aim to build an efficient supporters set which helps to quickly verify the location of the target. Supporters are features which consistently occur around the object as shown in Figure 3.20. They also have a strong correlation in motion with our target. It is worth noting that our goal is tracking in unconstrained environment with several challenges such as frame-cuts, abrupt motion due to hand-held camera recording. It limits us from using some motion model to predict the location of a target based on the motion of the supporters as in [63] or of the auxiliary objects as in [191]. We also would like to emphasize that our candidate responses are obtained based on detection. The supporters are also detected from the local region around each candidate. After that, these supporter detection responses are matched with the ones from previous frames to find the co-occurrence between them and our target. In fact, from these results, the motion correlations are also inferred without using very complex motion models needed in unconstrained environments. Moreover, unlike the supporters proposed in [63] which are expensive to detect and match in the whole frame, our supporters are efficiently detected and matched around the locations of the very few candidates having high probability to be the target in each frame.

To detect supporters, we use the Fast Hessian Detector and employ SURF descriptor as in [18] to describe the region around them. We store all of these supporters in a sliding window of k frames ($k = 5$ in our implementation). There are two types of supporters: *active* and *passive*. The active supporters are the ones co-occurring with our target in high frequency $fs > 0.5$ within the sliding window, while passive ones are the rest. When there are regions having similar appearance to our target but not being tracked by distracter trackers, all of SURF features are detected around these regions. After that, they are matched to our supporter model, which basically are the latest descriptors of the supporters in the sliding window. Finally, the supporting score is computed as follows

$$S_i = \frac{n_{am}}{n_{ta}} \tag{3.24}$$

where n_{am} and n_{ta} are the numbers of active matched supporters and total active supporters in the model. A supporter model is considered strong if $S_i > 0.5$ and $n_{ta} > 5$ (to avoid the unstable information within non-textured regions around our target). Then all of the matched results are used to update the supporter model. Note that the unmatched results are also added to the model.

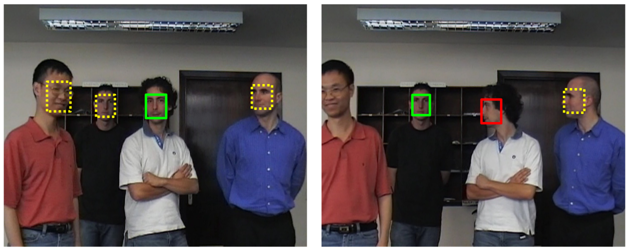
8.3. Context Tracker

We use the P-N Tracker [75] as our basic target tracker with several extensions. First, we extend the randomized ferns to accept multiple objects; in fact, it is not equivalent to a multi-class classifier because each object preserves its own posterior probability while they may share the same object type as our target. Second, we applied our new 6bitBP which helps to boost up the speed of the detector. Our 6bitBP makes use of the constant value of each whole patch during evaluation. Third, instead of using only the first initial patch as the object model, which is quite conservative and vulnerable to appearance changes, we use the online template-based object model as in [74]. However, we improve this model by constructing it in binary search tree using k-means. The model is iteratively split into two subsets to form a binary tree. By doing this, the computational complexity to evaluate a sample is $O(log_n)$ instead of $O(n)$ when using Brute-force. This improvement is important in improving the running time because the online model linearly grows to adapt to appearance changes. It is worth noting that other tracking methods can also be extended using our concepts. However, we choose the PN-Tracker because it uses scanning window to search for all of possible candidates in the whole image which helps to explore the context at the same time. Also, the randomized forest is extendable to reduce the cost of initializing a totally new tracker for a distracter.

As discussed, distracters are regions which have similar appearance as our target. In our tracker, a testing sample confidence score is computed using Normalized Cross-Correlation (NCC) between it and the closest image patch in the object model. The region having the highest confidence is considered as the current target if its score is larger than a threshold $\theta = 80\%$. However, in practice, there are several other regions satisfying this condition. After we choose the best candidate as the tracking result, all of other responses are associated to the distracter trackers using greedy association: the tracker producing higher confidence on a patch is associated with higher priority. The remaining regions trigger new distracter trackers. These trackers are formulated similarly to our basic tracker. However, to avoid the increasing number of unnecessary trackers, they are terminated whenever they lose their target.

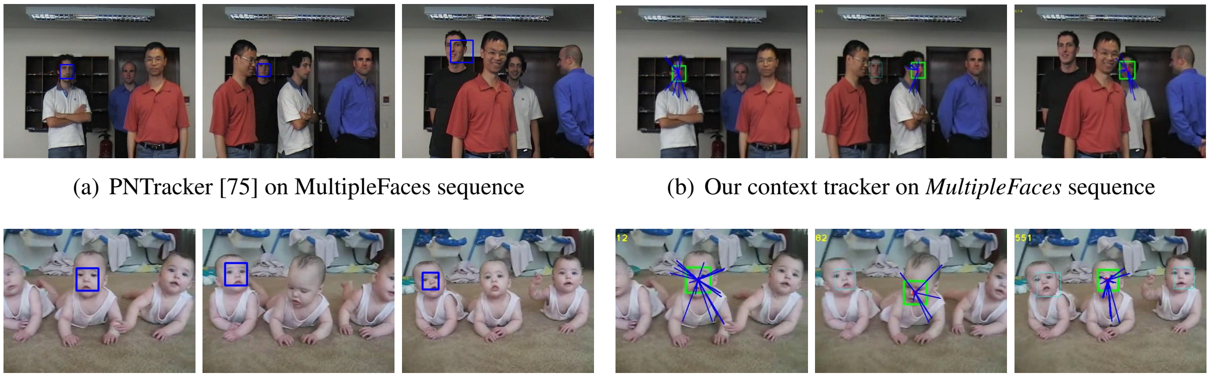
Assuming that we have the valid target at frame t, the supporters are extracted around the location of that target with a radius R. After that, a sliding window of $k = 5$ frames is used to store and match the previous supporters with the current ones. Each match makes the frequency of that supporter increase by 1.

In practice, there are several candidates similar to our target with very high confidence score. In fact, the right candidate may not even obtain the highest score, especially when the appearance is changing (as shown



(a) Start changing appearance (b) Exceeding the threshold, drifting.

Figure 3.21: Drifting to another object with the highest score.



(a) PNTracker [75] on MultipleFaces sequence (b) Our context tracker on *MultipleFaces* sequence

(c) PNTracker [75] on *Babies* sequence (d) Our context tracker on *Babies* sequence

Figure 3.22: Comparison between PNTracker [75] and our context tracker on challenging sequences

in Figure 3.21). Without context, the tracker obviously switches to the one with the highest score. Also, in unconstrained environments, our target may leave the FoV, or be completely occluded by other objects. The tracker will simply switch to another region satisfying the threshold θ. Here, our tracker automatically exploits all the distracters and pays attention to them by tracking them simultaneously. Also, our tracker discovers a set of supporters to robustly identify the target among other similar regions

8.4. Experiments

In our implementation, we use 8 ferns and 4 6bitBP features per fern. All thresholds are fixed as described. The most important threshold is the one which validates the correct target. It is calculated by the *NCC* between a candidate and the online object model. It has been carefully chosen as *80%* according to the experiment demonstrated in [74], which also shown that *LOOP* event outperforms the other growing ones. Our scanning window starts searching the minimum region of 20x20. For a sequence of resolution 320x240, the number of search windows is 100k while in 640x480, it is 600k, approximately.

We compare our tracker with and without context elements. The PNTracker is used as reference. It is worth noting that the implementation of PNTracker[1] is the combination implementation of [74, 75]. To emphasize the contribution of context in terms of distracters and supporters, we choose two very challenging sequences which contains similar objects moving: *Multiplefaces*, and *Babies*. It is important to note that in most of the cases where no strong context exists, our tracker still shows overall better results than PNTracker and outperforms other state-of-the-art methods.

To demonstrate the performance of our context tracker we use several recent state-of-the-art methods for comparison including: FragTracker (FT) [2], MILTracker (MILT) [11], CotrainingTracker (CoTT) [193], PNTracker [75], DNBSTracker (DNBST) [87], VTDTracker [81]. All codes com from the original authors.

[1] PNTracker: `http://info.ee.surrey.ac.uk/Personal/Z.Kalal/`

Video	Frames	FT	MILT	CoTT	DNBS	VTD	PNT	**Ours**
Animal	72	69	9	*8*	19	**6**	37	9
Carchase	5000	lost @ 355	lost @ 355	lost @ 409	lost @ 364	lost @ 357	@1645	**24**
Clutter	1528	lost @ 1081	lost @ 413	9	*6*	*6*	**4**	*6*
ETHPed	874	lost @ 95	lost @ 95	lost @ 95	lost @ 635	lost @ 95	**10**	*16*
Girl	502	lost @ 248	30	**14**	39	69	19	*18*
Liquor	1407	lost @ 47	lost @ 288	30	lost @ 404	lost @ 404	*21*	**10**
Motocross	2665	lost @137	lost @ 485	lost @ 591	lost @ 10	lost @ 10	**10**	*12*
Multifaces	1006	lost @ 64	lost @ 64	lost @ 394	lost @ 64	lost @ 64	@ 97	**26**
Scale	1911	8	11	6	lost @ 269	*3*	6	**2**
Vehicle	946	lost @ 679	lost @ 481	*9*	lost @ 517	lost @ 517	**8**	**8**
Speed		1.6	14	2	7	0.2	12*	10

Table 3.8: Average center location error (pixels). Performance comparison between the trackers (FT: FragTracker [2], MILT: MILTracker [11], CoTT: Co-Tracker [193], DNBS: DNBSTracker [87], VTD: Visual Tracking Decomposition [81], PNT: PNTracker [75], and Ours: Our context tracker) in different challenging video sequences. The best performance is in **bold**, the second best is in *italic*. The number in blue color indicates the frame number when the tracker gets lost. The * indicates that the method was implemented on Matlab using C-Mex.

The chosen data set includes several challenging sequences: *Motocross* and *Carchase* in [75], *Vehicle* in [193], Liquor[3], *ETHPedestrian*[4], *Multifaces*[2], *Clutter* and *Scale*[5], *Animal* used in [81], *and Girl* in [11]. They contain occlusion and object leaving FoV (*Motocross, Carchase, Vehicle, ETHPedestrian, Multifaces, Girl*), very cluttered background (*Carchase, Liquor, ETHPedestrian, Multifaces*) out-of-plane rotation (*Carchase*, *Vehicle*, *Multifaces*, *Girl*), abrupt motion (*Motocross, Clutter, Scale, Animal*), motion blur (*Liquor, Clutter, Animal*). Several of them are recorded in unconstrained environments such as *Motocross, Vehicle, ETHPedestrian, Carchase,* and *Animal*.

Because our chosen data set is very challenging with a number of long-term sequences, most current methods fail somewhere in the middle of a sequence. Therefore, we note the frame number where a tracker starts to lose the object and never reacquires. It means we accept the result of a tracker even when it fails to get the right target in several frames before reacquisition happens. A target is considered "lost" if the overlapping region between its bounding box and the ground-truth is less than 50%.

The quantitative comparisons are shown in Table 3.8. The running time comparison (in the last row) is for a raw reference as different methods have different search range which impacts the speed greatly. It shows that our tracker has overall better performance than PNTracker of [75] with the help of context, and outperforms all other approaches. Although most of them may work well in controlled environments, it is difficult for them to consistently follow the target in long-term sequences and in unconstrained environments. There are some large numbers in our results ("Carchase", "Motocross") is because it reacquires the object in several frames later than the ground truth, which makes the overall score look not good when we calculate the error using its previous position.

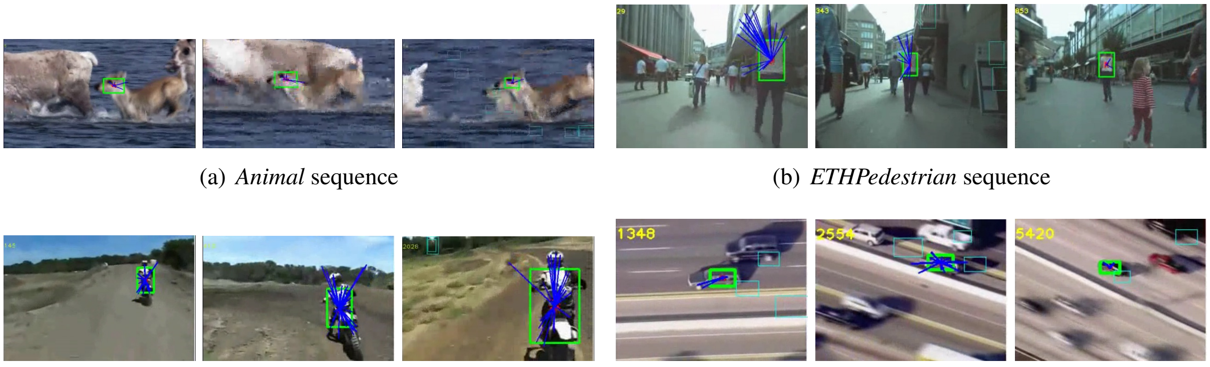
9. Active Vision System Using a Network Pan-Tilt-Zoom Camera

Research on human faces is a very important and interesting topic in computer vision with various applications such as biometric identity. Regular CCTV cameras cannot extract human faces with reasonable resolution when they are far away. PTZ cameras can zoom, pan, and tilt to get a close view. Today, this process only occurs under human control, which is impractical with a large number of cameras. Thus,

[3] PROST dataset: `http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php`

[4] ETH dataset: `http://www.vision.ee.ethz.ch/~aess/dataset/`

[5] `http://www.vision.ee.ethz.ch/boostingTrackers/index.htm`



(a) *Animal* sequence

(b) *ETHPedestrian* sequence

(c) *Motocross* sequence

(d) *Carchase* sequence

Figure 3.23: Some snapshots of our context tracker on several sequences.

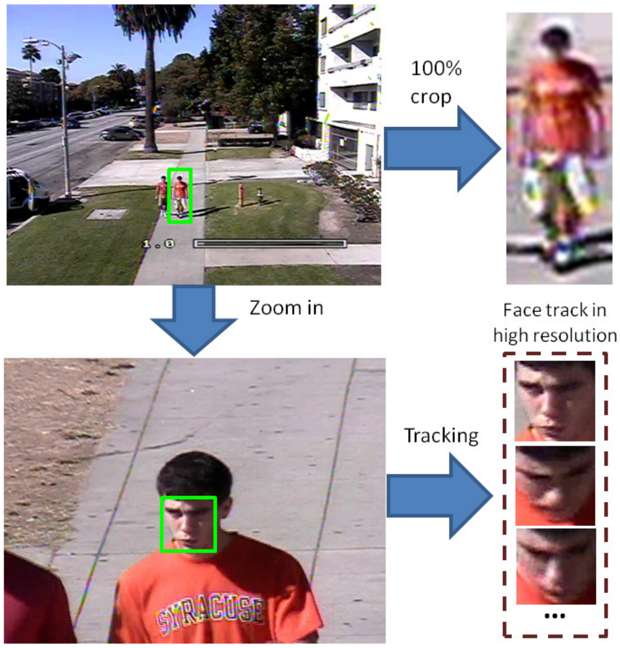


Figure 3.24: Scenario

automatically generating high resolution face sequences from PTZ cameras would be very helpful. Many challenges such as network delays, packet loss, and slow response commands still need to be addressed. Given such a scenario, we need not only a robust real-time tracker but also a flexible and smooth control module. The tracker needs to learn the face appearance changes under different conditions. It also has to be robust against cluttered background, abrupt motion, motion blur, and must reacquire the object after total occlusion or leaving FOV. In addition, when a PTZ camera is in wide mode, it performs as a regular CCTV camera, which means when a person is far from the camera, no face detector can detect the face within only few pixels (as shown in Figure 3.24). A practical system should automatically identify the region of interest containing the face, and zoom to detect, then track this face. To address all of the above issues, we present an autonomous system running in 3 different image modes, and 2 camera control modes. The overview is shown in Figure 3.25, where the operators acting on image and on camera are illustrated in light blue and light orange boxes, respectively. Our system uses a PTZ network camera to capture image sequences. The detector in pedestrian detection mode detects people in the FOV. The camera then switches to ROI focusing mode in order to zoom in the upper part of the detected human body. After that, the system goes into face

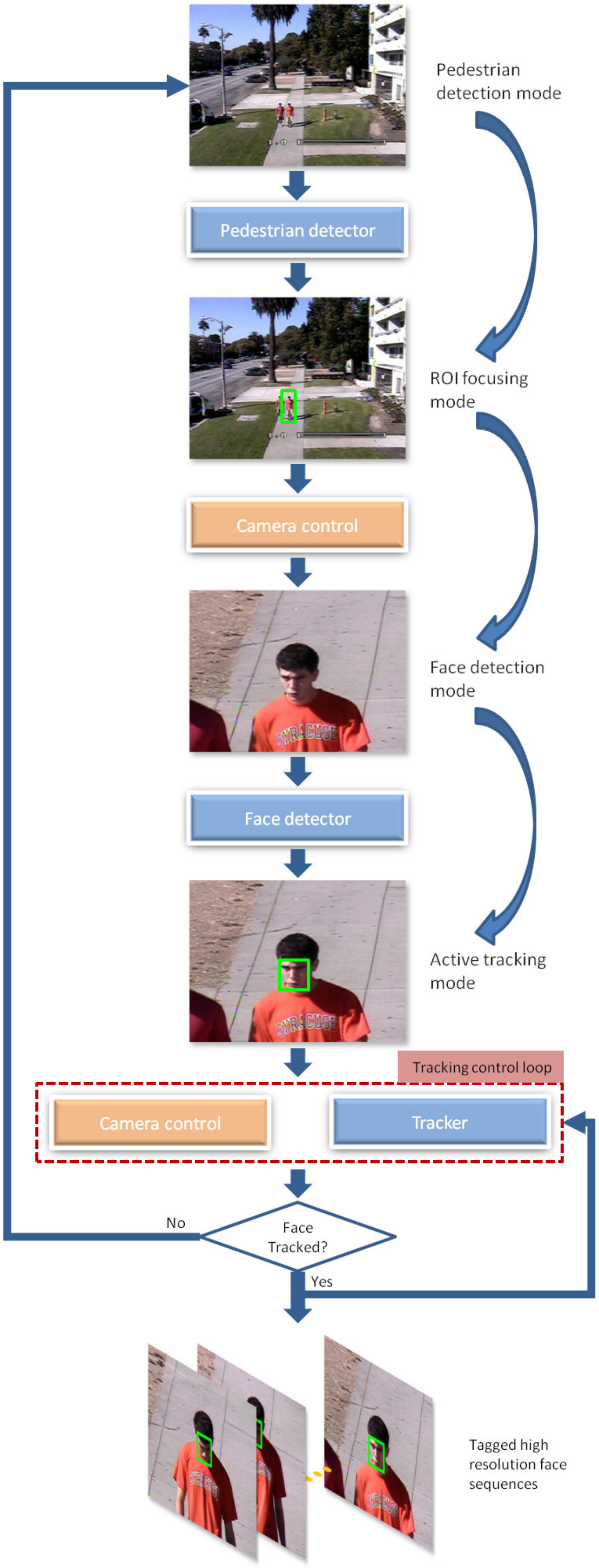


Figure 3.25: Overview of our system

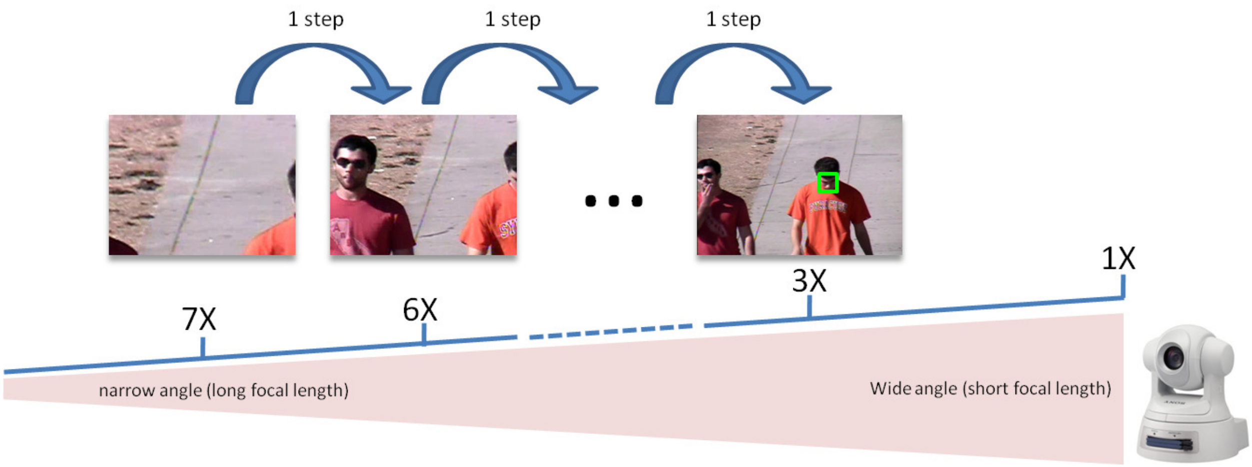

1 step
1 step
1 step
1X
3X
6X
7X
narrow angle (long focal length)
Wide angle (short focal length)

Figure 3.26: One-step-back strategy camera control

detection mode to find the face of interest, which was hard to detect from far away in wide angle focal length (Figure 3.24). An active tracking mode consiting a control loop with two interchangeable modules: camera control and tracking ones is then triggered to follow the target by keeping it around the center of the image at a proper predefined size.

9.1. Pedestrian Detection Module

Many methods model the background then apply a background subtraction algorithm to detect moving objects. However, it is impractical to model the background under all of possible variations of PTZ camera parameters. Moreover, the environment conditions can change all of the time. In practice, there are not only pedestrians moving but also other objects such as vehicles and foliage in wind. Hence, we want to generalize the problem by employing a frame-based state-of-the-art pedestrian detector to find a walking person in the camera's FOV. Several pedestrian detection methods were proposed such as [42, 187]. However, most of them do not have near real-time performance. Recently, Huang *et al.*[70] proposed a high performance object detection using joint ranking of granules (JRoG) features. With a simple ground plane estimation, this detector takes only 70 ms to scan a 640x480 test image at 16 different scales. JRoG is a simplified version of APCF [50] descriptor after excluding gradient granules. For feature selection, two complementary modules which are an SA and an incremental module were successfully proposed to find the optimal solution. After that, the part-based detection approach [187] is adopted to find the optimal combination of multiple partitions of body parts in the complex scene. For more details please refer to [70]. To avoid false alarms, we use the simple frame difference background subtraction technique to filter out most of static region in the background. Also, it is important to note that we do not need to run the pedestrian detector in every frame, and we continuously run it once per second instead.

9.2. ROI focusing module

After receiving the results from the pedestrian detector module, the system automatically chooses the highest confidence response and switches to the ROI focusing mode. Roughly, the head of a person is about $1/5$ the total height of whole body. In this mode, the camera parameters are adjusted so that the head position is close to the center of the image while its height is in the defined range. More clearly, let $C(c_x, c_y)$ denote the center of the image, $P(p_x, p_y)$ the center and h_p the height of the ROI in the current state. The camera pans, tilts, and zooms so that on the image plane:

$$\begin{cases} P \rightarrow C \\ h_p \rightarrow min_h \quad if \quad h_p < min_h \\ h_p \rightarrow max_h \quad if \quad h_p > max_h \end{cases} \tag{3.25}$$

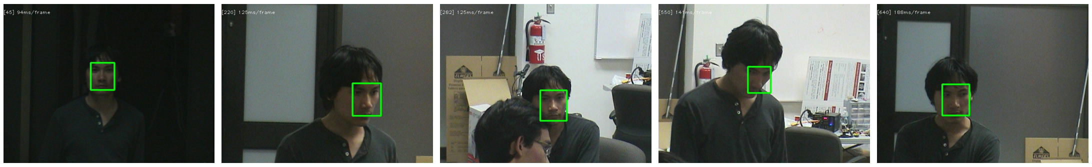


Figure 3.27: Indoor experiment results

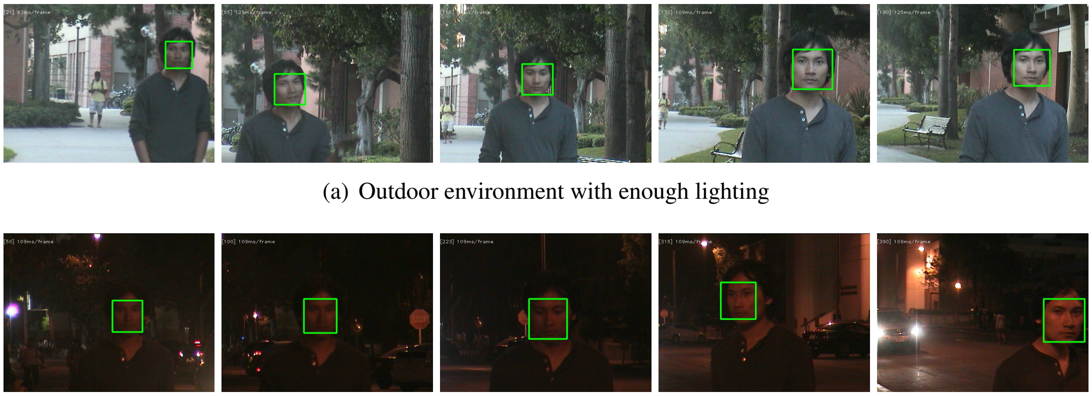
where $min_h = 80$ and $max_h = 120$ are the minimum and maximum heights of the face, respectively. Because this function only needs to be performed once, absolute commands are sent to control the camera. The output of this mode is a new video stream where the face in high resolution is likely in the image. It is important to note there is some possibility that no face exists in the new video stream: a false alarm from pedestrian detector, a person facing back or a fast moving pedestrian.

9.3. Face Detection Module

This mode is in charge of detecting the face in the current image. We employ the real-time face detection proposed by Froba and Ernst [56]. This method introduces a new feature set, illumination invariant Local Structure Features for object detection. The new feature set are computed from a 3x3 pixel neighborhood using a modified version of Census Transform [196] for efficient computation. A cascade of classifiers of four stages is adopted, each of which stands a linear classifier consisting a set of lookup tables of feature weights. The detection is then carried out by scanning all of possible windows with a fixed size of 22x22 pixels for each scaling factor of 1.25 until the size of the image is less than 22x22. Among the set of detected responses, the best one is chosen as the target. This face detector runs at 20fps on 640x480 image sequences and can detect faces with less than $45°$ out-of-plane rotation. For more information please refer to the original work in [56].

Once the face is detected, the active tracking mode is triggered with two modules: camera control and tracking.

9.4. Camera Control Module



(a) Outdoor environment with enough lighting

(b) Outdoor environment with low-light and noisy, out-of-focus images

Figure 3.28: Outdoor experiment results

This mode automatically controls the camera to follow the object. The control needs to perform smoothly and precisely. We use relative control commands with a truncated negative feedback to avoid oscillation. When sending commands to the camera, we adopt a time sharing strategy in which one pan-tilt command and one zoom command are packed as one sending group, and two sending groups with one inquiry command make a control loop. This is because the camera queue is limited and we should not send commands frequently; otherwise, when the queue is full, later commands are dropped. Moreover, the delay in responding to the inquiry command is longer than the others, so a proper time sharing strategy is helpful. Note that most PTZ cameras carry the commands by mechanical processes which accumulate errors by time. This fact prevents us from acquiring the current status camera based on the initial status by computing the difference at every step.

In practice, it is hard to follow the face all of the time, especially when focal length is long, *i.e.* when the camera zooms in on the far away face. The FOV of the camera is narrow, and the face can move out of the image in 2-3 frames. To address this issue, we introduce a one-step-back strategy (as shown in Figure 3.26). We divide the zoom range in nine steps covering from 1X to 18X. We ignore the further zoom because the FOV is too narrow at that time. Whenever the face is no longer tracked after $k = 20$ consecutive frames, the system zooms out one step back with the hope to reacquire the face again. The system iteratively continues this process until it can re-acquire the face (using the tracker, not the detector), otherwise the camera gets back to the widest mode. The system comes back to the home position after missing the target after 10 seconds of no detection. The home position is preset.

9.5. Tracking Module

We use our simplified context tracker described in previous section. We reduce the number of ferns, maximum distracters and supporters allowed.

9.6. Experiments

We use a Sony PTZ network camera SNC-RZ30N. This camera covers a large range of pan angles ($-170° \rightarrow +170°$), tilt angles ($-90° \rightarrow +25°$), and a large zoom ratio ($25X$ optical). The camera image stream sent to the computer is set at 640x480 with medium quality. Our complete system runs at 15fps. The complete system has been tested in challenging indoor and outdoor environments with real-life situations. Our system successfully detects a walking person, zooms in on his face and keeps track for a while. For indoor (see Figure 3.27), it is very challenging due to the face leaving FOV, large viewpoint changes, lighting condition changes. For outdoor environment, we tackle two different situations: one is in the afternoon with enough lighting, one is in the evening with low light. The results show that even when the camera cannot focus on the face with auto-focusing mode, and there is a lot of noise, our system still delivers acceptable results. Some snapshots are shown in Figure 3.28. It is important to note that, to reduce the overhead of recording images, we only save the sequences of images after the face is detected and starts to be tracked. For detailed results, please refer to our supplemental video.

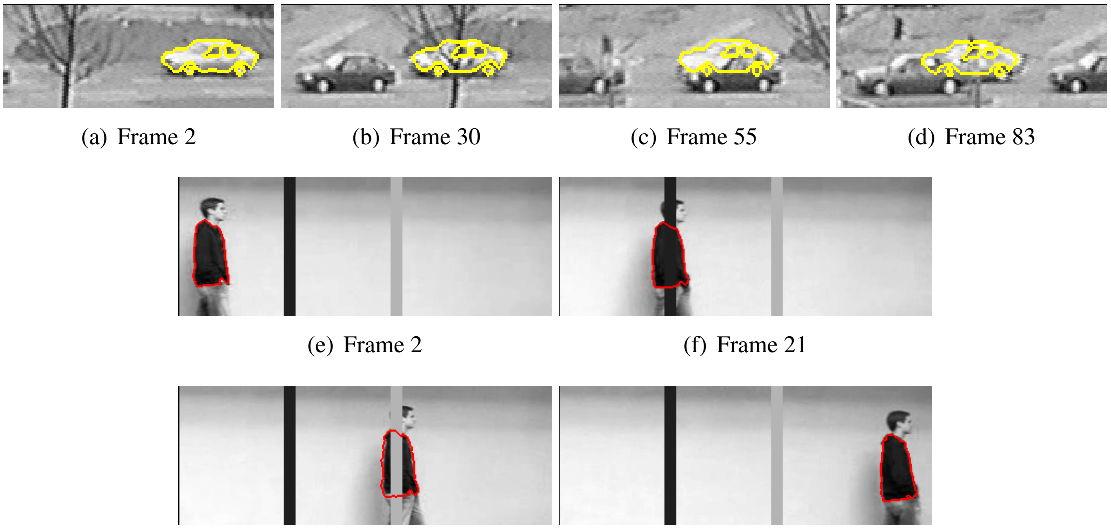
10. Logic Models for Image and Video Tracking

This research was presented at IASTED Conf. on Signal and Image Processing. The work was carried out by UCLA masters student James H. von Brecht UCLA postdoc Sheshadri R. Thiruvenkadam under the direction of Professor Tony Chan. In this work, we present a variational method for tracking objects under occlusion. We utilize prior shape information and the Logic Models of Sandberg and Chan to locate and segment the object of interest in each frame of the video sequence. In particular, we demonstrate how incorporating the shape prior via the Logic Models allows us to avoid segmentation local minima which occur with algorithms that simply additively introduce shape, and how to use the Logic Models within the context of tracking.

We model the object in the current frame as an affine transformation g of a shape prior, which we represent via a level set function ψ. We minimize two separate segmentation energies to find both the appropriate affine registration parameters p and the correct segmentation, represented as the zero level-set of a function ϕ. Each energy corresponds to a different logical interpretation of the object in the frame to be analyzed. Both of the energies take the familiar form

$$E = \int_{\Omega} f_{in} H\left(\phi\right) + \int_{\Omega} f_{out}\left(1 - H\left(\phi\right)\right).$$

The functions $f_{in,out}$ are image terms which drive the segmentation. The shape prior is incorporated as a binary image into each term. Since the shape and image terms are coupled, we avoid local minima problems encountered when shape is introduced additively. The image terms vary depending upon the correct logical interpretation (AND/OR) of the image to be analyzed, hence we minimize each of the two energies separately, then select one of the two as the correct segmentation. Lastly, we introduce a means of automatically determining the correct selection, thus allowing us to utilize the Logic Models within a tracking algorithm. An example of the algorithm is shown in Figure 3.29.



(a) Frame 2

(b) Frame 30

(c) Frame 55

(d) Frame 83

(e) Frame 2

(f) Frame 21

(g) Frame 48

(h) Frame 77

Figure 3.29: Occlusion tracking from two video sequences

Work by postdoc Berta Sandberg and PhD student Matthew Keegan, under the direction of Tony Chan, ha the goal of increasing the speed of detection of an object of interest while decreasing false detection rate. Two sets of algorithms were developed to aid this objective. The first algorithm has been developed to extract the moving object from background. A logic framework model was developed for multiphase segmentation on images that allows the combination of both intensity of the region where movement occurs, and the velocity vector field to identify the best possible view of the moving object. The second algorithm implemented was a clustering algorithm that identifies and sorts the shape of objects using calculated shape features. By extracting the precise shape of the object and identifying its shape features we hope to differentiate between moving objects of interest and noise from trees blowing in the wind or rain/snow.

Chapter 4

Sensor Management for Tracking

This chapter is intended to summarize the contributions of the group of Dr. Veeravalli in the theory of sensor management for tracking made with the support of this grant. Our work focused on two variants of the sensor management problem, namely, sensor scheduling and sensor sleeping. The former refers to the scenario where the sensors can be turned on or off at consecutive time steps and the goal is to select the subset of sensors to activate at each time step. The latter scenario refers to the scenario where an asleep sensor cannot be communicated with or woken up, and hence the sleep duration needs to be determined at the time the sensor goes to sleep based on all the information available to the sensor.

1. Sensor Sleeping

The sensor nodes typically need to operate on limited energy budgets. In order to conserve energy, the sensors may be put into a sleep mode. However, it is clear that having sleeping sensors in the network that cannot be woken up could result in tracking errors, and hence there is a tradeoff between the energy savings and the tracking error that results from the sleeping at the sensors. The sleeping policies at the sensors should be designed to optimize this tradeoff.

A straightforward approach is to have each sensor enter and exit the sleep mode using a fixed or a random duty cycle. A more intelligent, albeit more complicated, approach is to use information about the object trajectory that is available to the sensor from the network to determine the sleeping strategy. In particular, it is easy to see that the location of the object (if known) at the time when the sensor is put to sleep would be useful in determining the sleep duration of the sensor; the closer the object, the shorter the sleep duration should be. We took this latter approach in this work in designing sleeping strategies for the sensors. All information about the object trajectory is stored at some central unit and is used to determine sleep times of sensors that come awake. Using a bottom-up approach, we consider different sensing, motion and cost models with increasing levels of difficulty.

1.1. Simplified Models

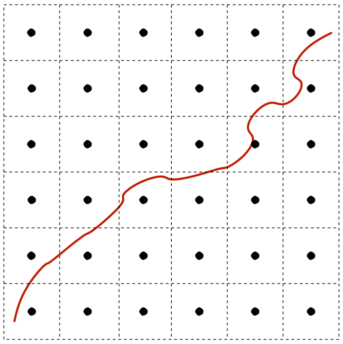


Figure 4.1: Object tracking in a field of sensors (simplified model).

We study the problem of tracking an object that is moving through a network of wireless sensors as shown in Figure 4.1. Each sensor has a limited range for detecting the presence of the object being tracked, and the objective is to track the location of the object to within the accuracy of the range of the sensor. First we considered a simplified model for a sensor network with n sensors [58]. We assume that the sensing ranges of the sensors completely cover the region of interest with no overlap. In other words, the region can be divided into n cells with each cell corresponding to the sensing range of a particular sensor. Each sensor can be in one of two states: awake or asleep. A sensor in the awake state consumes more energy than one in the asleep state. However, object sensing can be performed only in the awake state.

The movement of the object to be tracked is described by a Markov chain whose state is the current location of the object to within the accuracy of a cell. However, we also append a special terminal state, denoted as $\mathcal{T}$, that occurs when the object leaves the network. The statistics for the object movement are described by a $(n+1) \times (n+1)$ probability transition matrix P such that P_{ij} is the probability of the object being in state j at the next time step given that it is currently in state i. Let b_k denote the location of the object at time k. We can describe the evolution of the object location stochastically as $b_{k+1} \sim \boldsymbol{e}_{b_k} P$, where $\boldsymbol{e}_i$ is a row vector with a 1 at the i-th entry and zero elsewhere.

A central controller keeps track of the state of the network and assigns sleep times to sensors that are awake. In particular, each sensor that wakes up remains awake for one time unit during which the following actions are taken: (i) if the object is within its range, the sensor detects the object and sends this information to the central unit, and (ii) the sensor receives a new sleep time (which may equal zero) from the central controller. The sleep time input is used to initialize a timer at the sensor that is decremented by one time unit each time step. When this timer expires, the sensor wakes up. Let $r_{k,\ell}$ denote the value of the sleep timer of sensor ℓ at time k. We call the n-vector $\boldsymbol{r}_k$ the residual sleep times of the sensors at time k. Also, let $u_{k,\ell}$ denote the sleep time input supplied to sensor ℓ at time k. We can describe the evolution of the residual sleep times as

$$r_{k+1,\ell} = (r_{k,\ell} - 1)\mathbb{1} r_{k,\ell} > 0 + u_{k,\ell}\mathbb{1} r_{k,\ell} = 0 \tag{4.1}$$

The first term on the right hand side of this equation expresses that if the sensor is currently asleep (the sleep timer for the sensor is not zero), the sleep timer is decremented by 1. The second term expresses that if the sensor is currently awake (the sleep timer is zero), the sleep timer is reset to the current sleep time input for that sensor.

Hence, our system is described through a discrete-time dynamical model, with control input $\boldsymbol{u}_k$ and exogenous input w_k. The *state* of the system at time k is described by $x_k = (b_k, \boldsymbol{r}_k)$ and the state evolution. Not all of x_k is known to the central unit at time k since b_k is known only if the object is currently being tracked. Thus we have a dynamical system with incomplete (or partially observed) state information. If we denote the observation available to the central unit at time k by z_k, then $z_k = (s_k, \boldsymbol{r}_k)$, with

$$s_k = \begin{cases} b_k & \text{if } b_k \neq \mathcal{T} \text{ and } r_{k,b_k} = 0 \\ \mathcal{E} & \text{if } b_k \neq \mathcal{T} \text{ and } r_{k,b_k} > 0 \\ \mathcal{T} & \text{if } b_k = \mathcal{T} \end{cases} \tag{4.2}$$

where $\mathcal{E}$ denotes an unknown or "erasure" value. The total information available to the control unit at time k is given by

$$I_k = (z_0, \ldots, z_k, \boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}) \tag{4.3}$$

with $I_0 = z_0$ denoting the initial (known) state of the system. The control input for sensor ℓ at time k is allowed to be a function of I_k, i.e.,

$$\boldsymbol{u}_k = \mu_k(I_k) \tag{4.4}$$

The vector-valued function μ_k is the sleeping policy at time k.

We identify the two costs present in our tracking problem. The first is an energy cost of $c \in (0, 1]$ for each sensor that is awake. The second is a tracking cost of 1 for each time unit that the object is not tracked. If the object leaves the network (b_k enters the terminal state), we assume the problem terminates and no further cost is incurred. Thus, the total cost at time k is given by

$$g(x_k) = \mathbb{1} b_k \neq \mathcal{T} \left(\mathbb{1} r_{k,b_k} > 0 + \sum_{\ell=1}^{n} c \mathbb{1} r_{k,\ell} = 0 \right) \tag{4.5}$$

where c is the parameter used to tradeoff energy consumption and tracking errors. The total cost (over a possibly infinite horizon trajectory) for the system is given by

$$J(I_0, \mu_0, \mu_1, \ldots) = \mathsf{E}\left[\sum_{k=1}^{\infty} g(x_k)\middle| I_0\right] \tag{4.6}$$

Since g is bounded by $(cn + 1)$ and the expected time till the object leaves the network is finite, the cost function J well-defined. The goal is to compute the solution to

$$J^*(I_0) = \min_{\mu_0, \mu_1, \ldots} J_0(I_0, \mu_0, \mu_1, \ldots) \tag{4.7}$$

The solution to this optimization problem for each value of c yields an optimal sleeping policy. The optimization problem falls under the framework of partially observable Markov decision process (POMDP).

Partial observability presents a problem since the information for decision-making at time k given in (4.3) is unbounded in memory. To remedy this, we seek a sufficient statistic for optimization that is bounded in memory. It is a standard argument that for such an observation model, a sufficient statistic is given by the probability distribution of the state x_k given I_k. Such a sufficient statistic is referred to as a *belief state*. Since the residual sleep times portion of our state is observable, the sufficient statistic can be written as $v_k = (\boldsymbol{p}_k, \boldsymbol{r}_k)$, where $\boldsymbol{p}_k$ is a row vector of length $n + 1$ that denotes the probability distribution of b_k given I_k. Mathematically, we have

$$p_{k,\ell} = \mathsf{P}(b_k = \ell | I_k) \tag{4.8}$$

We can write the evolution of $\boldsymbol{p}_k$ as

$$\boldsymbol{p}_{k+1} = \boldsymbol{e}_{\mathcal{T}} \mathbb{1} b_{k+1} = \mathcal{T} + \boldsymbol{e}_{b_{k+1}} \mathbb{1} r_{k+1,b_{k+1}} = 0 + [\boldsymbol{p}_k P]_{\{j : r_{k+1,j} > 0\}} \mathbb{1} r_{k+1,b_{k+1}} > 0 \tag{4.9}$$

where $\boldsymbol{r}_{k+1}$ is defined through (4.1) and b_{k+1} (conditioned on $\boldsymbol{p}_k$) is distributed as

$$b_{k+1} \sim \boldsymbol{p}_k P \tag{4.10}$$

To understand (4.9), note that if the object is observed at time $k+1$, $\boldsymbol{p}_{k+1}$ becomes a point-mass distribution with all the probability mass concentrated at b_{k+1}. If the object is not observed, we eliminate all probability mass at sensors that are awake (since the object is known to not be at these locations) and renormalize. Thus, all information from observations is incorporated.

We can then write our policy and cost function in terms of the sufficient statistic. and the optimal cost defined in (4.7) becomes

$$J^*(v_0) = \min_{\mu_0, \mu_1, \ldots} J_0(v_0, \mu_0, \mu_1, \ldots) \tag{4.11}$$

There exists a stationary optimal policy for our problem (i.e. $\mu_0 = \mu_1 = \cdots = \mu^*$). Such a policy and the optimal cost J^* can be found by solving the Bellman equation given as

$$J(v) = \min_{\mu} \mathsf{E}\left[g(x_1) + J(v_1) | v_0 = v, \boldsymbol{u}_0 = \mu(v)\right] \tag{4.12}$$

with μ^* being the minimizing value of μ in this equation. The state space for this problem consists of $\boldsymbol{p}_k$, which is uncountably infinite, and $\boldsymbol{r}_k$, which is countably infinite. Thus, we must either have an analytical solution for the cost function at each iteration (which is not possible given the complexity of the problem) or we must quantize and truncate the state space so that there are a finite number of states. Of course, restricting the infinite state space to a finite state space will lead to some loss of optimality. Even with the restriction to a finite state space, the complexity of value iteration remains intractable except for the most trivial cases. This is because the state space grows exponentially with the number of sensors. For example, even with seven sensors, a maximum sleep time of 10, and a probability mass function quantized to multiples of 0.1, there are about 10^9 possible states v_k. Hence, finding an optimal solution to this problem is not feasible. During the course of this MURI program we identified other approaches which were shown to be near-optimal.

1.1.1. Approximate Sleeping Policies

Much of the complexity of this problem stems from the complicated evolution of $\boldsymbol{p}_k$ given in (4.9). In deriving suboptimal solutions to our problem, we make assumptions about the observations that will be available in the future. These assumptions allow us to simplify the evolution in (4.9) considerably. The result is that the optimization problem easily separates into n simpler problems, one for each sensor. In each of these simpler problems, we are able to eliminate the residual sleep times $\boldsymbol{r}_k$ from the state since the only times of interest will be those when the sensor comes awake. It is then possible to solve each of the n simpler problems to find a cost function and policy.

We now define some additional notation. Under each assumption, we refer to an optimal cost and policy as J^* and μ^*. These functions may written in terms of the state ($J^*(v)$, $\mu^*(v)$) or the component parts of the state ($J^*(\boldsymbol{p},\boldsymbol{r})$, $\mu^*(\boldsymbol{p},\boldsymbol{r})$). The optimal cost and policy for the simpler problem for sensor ℓ will be $J^{*(\ell)}$ and $\mu^{*(\ell)}$ respectively. These functions are written in terms of the $\boldsymbol{p}$ portion of the state ($J^{*(\ell)}(\boldsymbol{p})$, $\mu^{*(\ell)}(\boldsymbol{p})$). Define $J^{*(\ell)}(\boldsymbol{p},r)$ for $r>0$ as

$$J^{*(\ell)}(\boldsymbol{p},r)=\sum_{j=1}^{r-1}\left[\boldsymbol{p}P^j\right]_\ell+\sum_{i=1}^{n}c\left[\boldsymbol{p}P^r\right]_i+\mathsf{E}\left[J^{*(\ell)}(\boldsymbol{p}_r)|\boldsymbol{p}_0=\boldsymbol{p}\right] \tag{4.13}$$

We showed for each assumption that with these definitions, J^* defined through

$$J^*(v)=J^*(\boldsymbol{p},\boldsymbol{r})=\sum_{\ell=1}^{n}J^{*(\ell)}(\boldsymbol{p},r_\ell) \tag{4.14}$$

is a solution to (4.12). It is also shown that if $\mu^{*(\ell)}(\boldsymbol{p})$ is set to the minimizing value of u for all $\boldsymbol{p}$ and for all $\ell\in\{1,\ldots,n\}$, then μ^* defined through

$$\mu^*(v)=\mu^*(\boldsymbol{p},\boldsymbol{r})=[\mu^{*(1)}(\boldsymbol{p}),\mu^{*(2)}(\boldsymbol{p}),\ldots,\mu^{*(n)}(\boldsymbol{p})] \tag{4.15}$$

is the minimizing policy in (4.12). In other words, the sleep times for each sensor can be chosen independently as a function of $\boldsymbol{p}$ alone.

1.1.2. FCR Solution

To generate the first cost reduction (FCR) solution, we assume that we will have no future observations. In other words, we are replacing (4.9) with

$$\boldsymbol{p}_{k+1}=\boldsymbol{p}_kP \tag{4.16}$$

Note that this does not mean that it will be impossible to track the object; we are simply making an assumption about the future state evolution in order to generate a sleeping policy. We can solve the equation

$$J^{(\ell)}(\boldsymbol{p})=\min_u\left(\sum_{j=1}^{u}\left[\boldsymbol{p}P^j\right]_\ell+\sum_{i=1}^{n}c\left[\boldsymbol{p}P^{u+1}\right]_i+J^{(\ell)}(\boldsymbol{p}P^{u+1})\right) \tag{4.17}$$

to find the cost function and policy. It is easy to verify that

$$J^{*(\ell)}(\boldsymbol{p})=\sum_{j=1}^{\infty}\min\left\{\left[\boldsymbol{p}P^j\right]_\ell,\sum_{i=1}^{n}c\left[\boldsymbol{p}P^j\right]_i\right\} \tag{4.18}$$

is indeed a solution to (4.17). In other words, at each time step we incur a cost that is the minimum of the expected tracking cost at sensor ℓ and the expected energy cost at sensor ℓ. Define the set $\mathcal{U}(\boldsymbol{p})$ as

$$\mathcal{U}(\boldsymbol{p})=\left\{u:\left[\boldsymbol{p}P^{u+1}\right]_\ell\geq\sum_{i=1}^{n}c\left[\boldsymbol{p}P^{u+1}\right]_i\right\} \tag{4.19}$$

It is then easily verified that the per-sensor policy $\mu^{*(\ell)}(\boldsymbol{p})$ has the form

$$\mu^{*(\ell)}(\boldsymbol{p}) = \min_{u \in \mathcal{U}(\boldsymbol{p})} u \tag{4.20}$$

More simply, the policy is to come awake at the first time such that the expected tracking cost exceeds the expected energy cost. This is why this solution is called the first cost reduction solution.

1.1.3. Q$_{MDP}$ Solution

A Q$_{\text{MDP}}$ solution is one in which it is assumed that the partially observed state becomes fully known after a control input has been chosen. In our problem, this means assuming that we will have perfect future observations, i.e., the location of the object will be known in the future. In other words, we are replacing (4.9) with

$$\boldsymbol{p}_{k+1} = \boldsymbol{e}_{b_{k+1}} \tag{4.21}$$

Note that this does not mean that it will be impossible to incur tracking errors; we are simply making an assumption about the future state evolution in order to generate a sleeping policy. We can solve the equation

$$J^{(\ell)}(\boldsymbol{p}) = \min_{u} \left(\sum_{j=1}^{u} \left[\boldsymbol{p}P^{j}\right]_{\ell} + \sum_{i=1}^{n} c \left[\boldsymbol{p}P^{u+1}\right]_{i} + \sum_{i=1}^{n} \left[\boldsymbol{p}P^{u+1}\right]_{i} J^{(\ell)}(\boldsymbol{e}_i) \right) \tag{4.22}$$

to find the cost function and policy.

Clearly, if we can solve (4.22) for $\boldsymbol{p} = \boldsymbol{e}_b$ for all $b \in \{1, \ldots, n\}$, then it is straightforward to find the solution for all other values of $\boldsymbol{p}$. We therefore concern ourselves with finding values of $J^{*(\ell)}(\boldsymbol{e}_b)$ and $\mu^{*(\ell)}(\boldsymbol{e}_b)$ that satisfy (4.22) for all $b \in \{1, \ldots, n\}$. This can be achieved through policy iteration. Note that for the Q$_{\text{MDP}}$ solution, we are assuming more information than is actually available. Thus, the cost function obtained under the Q$_{\text{MDP}}$ is a lower bound on optimal performance.

1.1.4. Point Mass Approximations

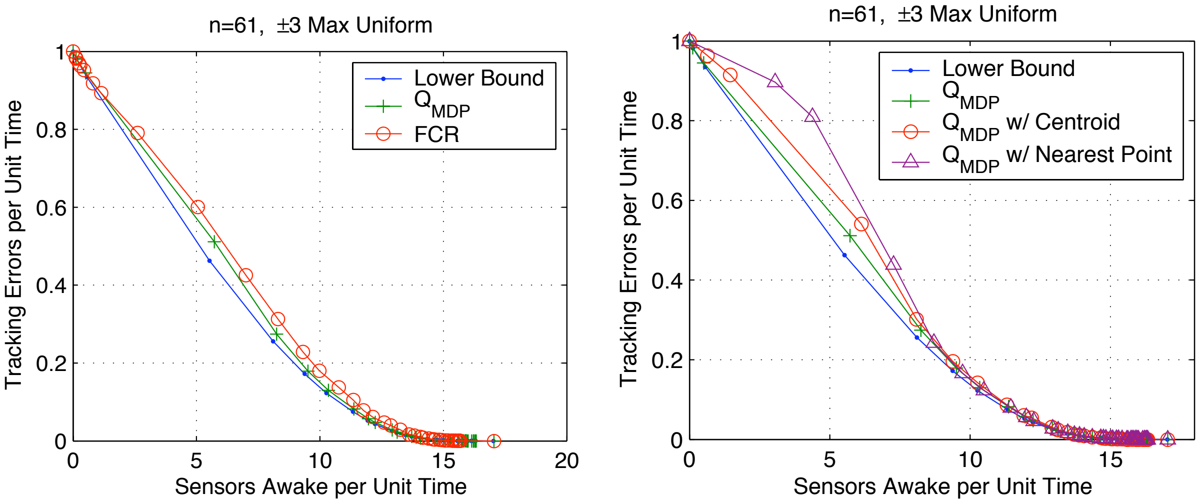
The suboptimal policies derived in the preceding sections are considerably easier to compute than the optimal policy and can be computed on-line after some initial off-line computation has been completed. However, such on-line computation requires sufficient processing power and could introduce delays. It would be convenient if the suboptimal μ^* could be precomputed and stored either at the central controller or distributed across the sensors themselves. The latter option is particularly attractive since it allows for distributed implementation. But the set of possible distributions $\boldsymbol{p}$ is potentially quite large — even if quantization is performed — and could make the storage requirements prohibitive. To make the storage requirements feasible, we considered approximating $\boldsymbol{p}$ with a point mass distribution. The number of sleep times to be stored is then only n per sensor. We considered two options for the placement of the unit point mass when computing the sleep time for sensor ℓ: (i) the centroid of $\boldsymbol{p}$, and (ii) the nearest point to sensor ℓ on the support of $\boldsymbol{p}$. Note that the latter option allows for the implementation of policies without detailed information about the statistics of the random walk – only the support of the random walk is required.

1.1.5. Numerical Results

In this section, we show some results that illustrate the performance of the policies we derived in previous sections. We begin with simulation results for one-dimensional sensor networks. In each simulation run, the object was initially placed at the center of the network and the location of the object was made known to each sensor. A simulation run concluded when the object left the network. The results of many simulation runs were then averaged to compute an average tracking cost and an average energy cost.

We present results for a one-dimensional network with 61 sensors where the object can move anywhere from three positions to the left to three positions to the right at each time step, with the object movement being uniformly distributed on these seven positions.

In Figure 4.2, we show the tradeoff curves between energy cost and tracking cost. From these data,



(a)

(b)

Figure 4.2: Energy-Tracking tradeoff curves.

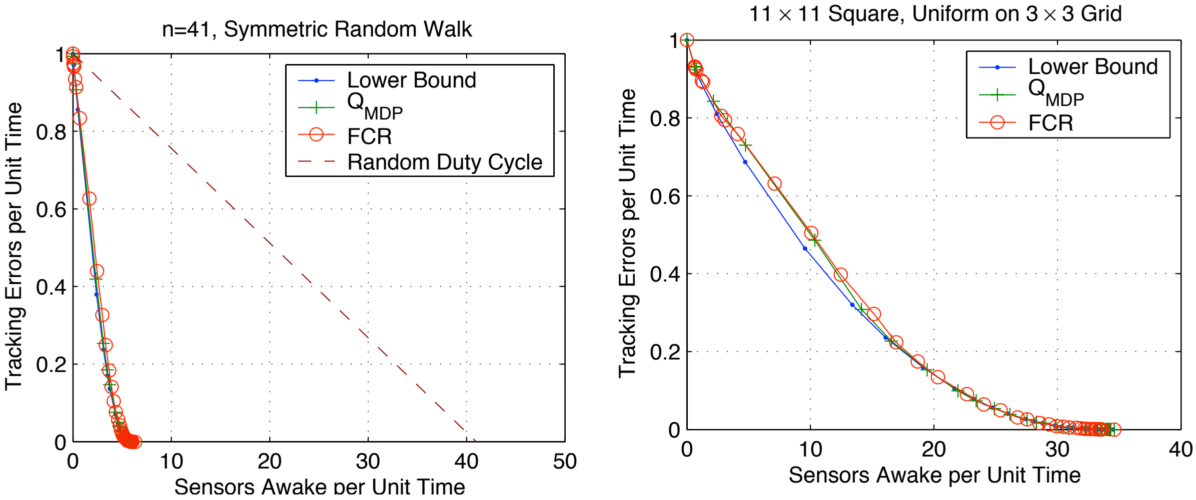
we see that the $\mathrm{Q_{MDP}}$ policy outperforms the FCR policy, although the difference does not appear as large. Note that the difference between the performance of the $\mathrm{Q_{MDP}}$ policy and the lower bound on optimal performance becomes small as the number of tracking errors becomes small. This makes sense since when there are few tracking errors, the $\mathrm{Q_{MDP}}$ assumption (that we will know the object location in the future) becomes realistic. In Figure 4.2(b), we explore the impact of using the point mass approximations on the performance of the $\mathrm{Q_{MDP}}$ policy. Four curves are shown in each figure. The first two are the lower bound and $\mathrm{Q_{MDP}}$ tradeoff curves already seen. The third and fourth curves are the tradeoff curves for the $\mathrm{Q_{MDP}}$ policy using the centroid and nearest point point mass approximations respectively. It can be seen that there is indeed some loss in performance when using point mass approximations, but this loss becomes small as the number of tracking errors becomes small. This makes sense since when tracking errors are infrequent, the object location is usually known exactly and so the distribution is usually already a point mass distribution.

For the moment, consider the traditional duty cycle scheme where each sensor is awake in a fraction π of the time slots. As π is varied, we achieve a tradeoff curve that is a straight line between the points $(0, 1)$ and $(n, 0)$ (where n is the appropriate number of sensors). When compared with this policy, the schemes we have proposed result in significant improvement as seen in Figure 4.3(a). In Figure 4.3(b), we consider a two-dimensional network. The network considered is a 11×11 grid (121 nodes). The movement of the object is best described by stating that at each time step the object starts at the center of a 3×3 grid and moves to any of the nine spaces on that grid with equal probability. It is seen that the results of Figure 4.3 are similar to those already seen in Figure 4.2.

1.2. Generalized Models

In this section we summarize extensions to the work presented in Section 1.1. We extend our analysis to more generalized models for object movement, object sensing, and tracking cost [60]. We allow the number of possible object locations to be different from the number of sensors. The number of possible object locations can even be infinite to model the movement of an object on a continuum. Moreover, the object sensing model allows for an arbitrary distribution for the observations given the current object location, and the tracking cost is modeled via an arbitrary distance measure between the actual and estimated object location.

Not surprisingly, this generalization results in a problem that is much more difficult to analyze. Our approach is to build on the policies developed for the simplified model of Section 1.1. The design of those policies relied on the separation of the problem into a set of simpler subproblems. In [58], we have shown



(a) Comparison to Duty Cycle

(b) Tradeoff curves for a two-dimensional network.

Figure 4.3: Comparison to conventional duty cycle approach and tradeoffs for a 2-D network.

that under an observable-after-control assumption, the design problem lends itself to a natural decomposition into simpler per-sensor subproblems due to the simplified nature of the tracking cost structure. Unfortunately, this does not extend to the generalized cases we consider herein. However, based on the intuition gained from the structure of the solution in the simplified case, we artificially separate our problem into a set of simpler per-sensor subproblems. The parameters of these subproblems are not known *a priori* due to the difficulties in analysis. However, we use Monte Carlo simulation and learning algorithms to compute these parameters. We characterize the performance of the resulting sleeping policies through simulation. For the special case of a discrete state space with continuous Gaussian observations, we derive a lower bound on the optimal energy-tracking tradeoff which is shown to be loose at the high tracking error regime, but is reasonably tight for the low tracking error region.

We extend the definitions presented in Section 1.1 to account for the new models. We denote the set of possible object locations as $\mathcal{B}$ such that $|\mathcal{B}| = m + 1$ where the $(m + 1)$-th state represents the absorbing terminal state that occurs when the object leaves the network. If $\mathcal{B}$ is not a finite set then m is ∞. We define a *kernel* P such that $P(x, \mathcal{Y})$ is the probability that the next object location is in the set $\mathcal{Y} \subset \mathcal{B}$ given that the current object location is x. Suppose p is a probability measure on $\mathcal{B}$ such that $p(\mathcal{X})$ for $\mathcal{X} \in \mathcal{B}$ is the probability that the state is in $\mathcal{X}$ at the current time step. Then the probability that the state will be in $\mathcal{Y}$ after t time steps in the future is given by

$$(pP^t)(\mathcal{Y}) \equiv \int_{\mathcal{B}} p(dx)P^t(x, \mathcal{Y}) \tag{4.23}$$

This defines the measure pP^t which depends on both the prior p and the transition Kernel P. Also, let δ_x denote a probability measure such that $\delta_x(\mathcal{A}) = 1$ if $x \in \mathcal{A}$, and $\delta_x(\mathcal{A}) = 0$ otherwise. Conditioned on the object state b_k, the future state b_{k+1} has a distribution $\delta_{b_k}P$. This defines the evolution of the object location. For a discrete state space this was simply the probability mass function defined by the b_k-th row of a transition matrix P.

To define the tracking cost, we first define the estimated object location at time k to be $\hat{b}_k$. We can think of $\hat{b}_k$ as an additional control input that is a function of I_k, i.e., $\hat{b}_k = \beta_k(I_k)$. Since $\hat{b}_k$ does not affect the state evolution, we do not need past values of this control input in I_k. The tracking cost is a distance measure that is a function of the actual and estimated object locations and is written as $d(b_k, \hat{b}_k)$. We assume that d is a bounded function on $\mathcal{B} \times \mathcal{B}$. Two examples of distance measures we might employ are the Hamming cost (if the space $\mathcal{B}$ is finite), i.e.,

$$d(b_k, \hat{b}_k) = \mathbb{1}\hat{b}_k \neq b_k \tag{4.24}$$

and the squared Euclidean distance (if the space $\mathcal{B}$ is a subset of an appropriate vector space), i.e.,

$$d(b_k, \hat{b}_k) = \|\hat{b}_k - b_k\|_2^2 \tag{4.25}$$

Recall that the input $\hat{b}_k$ does not affect the state evolution; it only affects the cost. Therefore, we can compute the optimal choice of $\hat{b}_k$, given by $\beta_k^*(I_k)$, using an optimization minimizing the tracking error over a single time step. We can thus write

$$\beta_k^*(I_k) = \arg\min_{\hat{b}} \mathsf{E}\left[d(b_k, \hat{b}_k)\middle| I_k\right] \tag{4.26}$$

Remembering that once the terminal state is reached no further cost is incurred, we can write the total cost for time step k as

$$g(b_k, I_k) = \mathbb{1}b_k \neq \mathcal{T}\left(d\left(b_k, \beta_k^*(I_k)\right) + \sum_{\ell=1}^{n} c\mathbb{1}r_{k,\ell} = 0\right) \tag{4.27}$$

The infinite horizon cost for the system is given by

$$J(I_0, \mu_0, \mu_1, \ldots) = \mathsf{E}\left[\sum_{k=1}^{\infty} g(b_k, I_k)\middle| I_0\right] \tag{4.28}$$

The solution to an optimization problem equivalent to the one in 4.12 for each value of c yields an optimal sleeping policy. The task of recursively computing p_k for each k is a problem in nonlinear filtering. In other words, p_{k+1} can be computed using standard Bayesian techniques as the posterior measure resulting from prior measure pP and observations $\boldsymbol{s}_{k+1}$. We can then write our dynamic programming problem in terms of the sufficient statistic and solve for the optimal policy $\mu_k(p_k, \boldsymbol{r}_k)$. Since an optimal solution could not be found for the simpler problem considered in Section 1.1, we immediately turn our attention to finding suboptimal solutions to the generalized version of the problem.

In general, the cooperation among the sensors may be difficult to analyze and understand. Furthermore, unlike the simple model, the tracking cost may not be easily written as a sum across the sensors. To generate suboptimal solutions we artificially write the problem as a set of subproblems that can be solved using similar techniques. The tracking cost expressions (which are a function of the sleeping actions of the sensors) in these subproblems will be left as unknowns. To determine appropriate values for these tracking costs, we either perform Monte Carlo simulations before tracking begins or use data gathered during tracking. The intuition is that if the resultant tracking cost expressions capture the "typical" behavior of the actual tracking cost, then our sleeping policies should perform well.

Our approach has two main ingredients. First, we make assumptions about the observations that will be available to the controller at future time steps. To generate sleeping policies, we assume that the system is either perfectly observable or totally unobservable after control. Hence, we define approximate recursions with special structure as surrogates for the optimal value function. Second, we devise different methodologies to evaluate suitable tracking costs whereby we capture the effect of each sensor on the overall tracking cost. Writing the combined tracking cost as the sum of independent contributions of different sensors (with respect to some baseline) allows us to write the Bellman equation as the sum of per-sensor recursions. Instead of solving the exact Bellman equation, we alternatively solve n simpler Bellman equations to find per-sensor policies and cost functions. The overall policy is then the per-sensor policies applied in parallel.

We denote by $J^{(\ell)}$ the cost function of the ℓ-th sensor approximate subproblem. We define $T^{\Delta}(b, \ell)$ to be the increase in tracking cost due to not waking up sensor ℓ at time k given that $b_{k-1} = b$. This is meant to capture the contribution of the ℓ-th sensor to the total tracking cost. Next we define our approximations.

1.2.1. Q_{MDP}

Based on the structure presented in Section 1.1, we can readily define a *Q_{MDP} per-sensor Bellman equation* analogous to the one in (4.22)

$$J^{(\ell)}(p) = \min_u \left(\sum_{j=0}^{u-1} \int_{\mathcal{B}-\mathcal{T}} T^{\Delta}(b,\ell)\,(pP^j)(db) + \int_{\mathcal{B}-\mathcal{T}} \left(c + J^{(\ell)}(\delta_b)\right)\,(pP^{u+1})(db) \right) \tag{4.29}$$

The first summation in the Right Hand Side (R.H.S.) of (4.29) corresponds to the expected tracking cost incurred by the sleep duration u of sensor ℓ. The second term consists of: (i) the energy cost incurred as the sensor comes awake after its sleep timer expires (after $u+1$ time slots); and (ii) the cost to go under an observable-after-control assumption (hence the belief state is δ_b).

We cannot find an analytical solution for (4.29). However, note that if we can solve (4.29) for $p = \delta_b$ for all b, then it is straightforward to find the solution for all values of p. Thus, given a function T^{Δ}, (4.29) can be solved through standard policy iteration, but only if $\mathcal{B}$ is finite.

1.2.2. First Cost Reduction (FCR)

Similarly, we define a First Cost Reduction *(FCR) Bellman equation* analogous to the one in (4.30) as

$$J^{(\ell)}(p) = \min_u \left(\sum_{j=0}^{u-1} \int_{\mathcal{B}-\mathcal{T}} T^{\Delta}(b,\ell)\,(pP^j)(db) + c\int_{\mathcal{B}-\mathcal{T}} (pP^{u+1})(db) + J^{(\ell)}(pP^{u+1}) \right) \tag{4.30}$$

In this case, it is assumed that we will have no future observations. In other words, we define the belief evolution as $p_{k+1} = p_k P$. Given a function T^{Δ}, it is easy to verify that the solution to (4.30) is

$$J^{(\ell)}(p) = \sum_{j=0}^{\infty} \min\left\{ \int_{\mathcal{B}-\mathcal{T}} T^{\Delta}(b,\ell)\,(pP^j)(db), c\int_{\mathcal{B}-\mathcal{T}} (pP^{j+1})(db) \right\} \tag{4.31}$$

and the associated policy is to choose the first value of u such that

$$c\int_{\mathcal{B}-\mathcal{T}} (pP^{u+1})(db) \geq \int_{\mathcal{B}-\mathcal{T}} T^{\Delta}(b,\ell)\,(pP^u)(db), \tag{4.32}$$

In other words, the policy is to come awake at the first time the expected tracking cost exceeds the expected energy cost where the tracking cost is defined based on T^{Δ} (to be determined).

The solutions to the per-sensor Bellman equations in (4.29) and (4.30) define the $\mathrm{Q_{MDP}}$ and FCR policies for each sensor, respectively. Note that, unlike the simplified case, the solution to the $\mathrm{Q_{MDP}}$ recursion does not necessarily provide a lower bound on the optimal value function since the employed tracking cost is not a lower bound on the actual tracking cost. Later, we present a lower bound on the optimal energy-tracking tradeoff for discrete state spaces with Gaussian Observations. The remaining task was to identify appropriate values of $T^{\Delta}(b,\ell)$ for all $b \neq \mathcal{T}$ and for all ℓ. This is the subject of the next two sections.

1.2.3. Non-learning Approach

For now, suppose that $\mathcal{B}$ is a finite space. Suppose $b_{k-1} = b$. To generate $T^{\Delta}(b,\ell)$ for a particular ℓ, we first assume a "baseline" behavior for the sensors, i.e., we make an assumption about the set of sensors that are awake at time k given that $b_{k-1} = b$. We consider two possibilities: (i) That all sensors are asleep; (ii) That the set of sensors awake is selected through a greedy algorithm. In other words, the sensor that causes the largest decrease in expected tracking cost is added to the awake set until any further reduction due to a single sensor is less than c. Starting with this set of awake sensors, the value of $T^{\Delta}(b,\ell)$ is then computed as the absolute difference in expected tracking cost incurred by changing the state of sensor ℓ. Monte Carlo simulation can be used to evaluate the change in expected tracking cost.

If $\mathcal{B}$ is not finite, then a parameterized version of T^{Δ} can be computed instead. We choose $\tilde{m}$ elements of $\mathcal{B} - \mathcal{T}$ and evaluate T^{Δ} at these points. The value of T^{Δ} at all other values of $b \in \mathcal{B} - \mathcal{T}$ can be computed via an interpolation algorithm. Recall that only an FCR policy is appropriate in the infinite state case, since solving the $\mathrm{Q_{MDP}}$ Bellman equation for an infinite number of point mass distributions is infeasible.

1.2.4. Learning Approach

Next, we describe an alternative learning-based approach. For ease of exposition, suppose that $\mathcal{B}$ is a finite space. Then our probability measure p_k can be characterized by a probability mass function. We refer to this probability mass function as $\boldsymbol{p}_k$ (a row vector). Define $\hat{a}_{k,\ell}$ to be the approximated expected increase in tracking cost due to sensor ℓ sleeping at time k as

$$\hat{a}_{k,\ell} = \sum_{b \neq \mathcal{T}} \boldsymbol{p}_{k-1}(b) T^{\Delta}(b, \ell) \tag{4.33}$$

Ideally, we would like this approximation to be equal to the actual expected increase in tracking cost due to sensor ℓ sleeping. Unfortunately, we do not have access to actual tracking costs at time k since b_k is not known exactly. However, we do have access to $\boldsymbol{p}_k$, $\boldsymbol{r}_k$, and $\boldsymbol{p}_{k-1}$. It is therefore possible to estimate the tracking cost as

$$\int_{\mathcal{B}} d(b, \beta^*(\boldsymbol{p}_k)) \boldsymbol{p}_k(db) \tag{4.34}$$

For example, if Hamming cost is being used, then we can estimate the tracking cost as $1 - \max_b p_k(\{b\})$, and if squared Euclidean distance is being used we can estimate the tracking cost using the variance of the measure p_k. Next we describe how we learn T^{Δ} by solving a least squares problem.

Determining an estimate of the *increase* in the tracking cost due to the sleeping of sensor ℓ at time k, denoted $a_{k,\ell}$, depends on the value of $r_{k,\ell}$. If $r_{k,\ell} = 0$, we ignore the observation from sensor ℓ and generate a new version of $\boldsymbol{p}_k$ called $\boldsymbol{p}'_k$. We can compute $a_{k,\ell}$ as

$$a_{k,\ell} = \sum_{b \neq \mathcal{T}} \boldsymbol{p}'_k(b) d(b, \beta^*(\boldsymbol{p}'_k)) - \sum_{b \neq \mathcal{T}} \boldsymbol{p}_k(b) d(b, \beta^*(\boldsymbol{p}_k)) \tag{4.35}$$

If on the other hand $r_{k,\ell} > 0$, we first generate an object location b'_k according to $\boldsymbol{p}_k$ and then generate an observation according to the probability measure $\sigma_{b'_k}$. This observation is used to generate a new distribution $\boldsymbol{p}'_k$ from $\boldsymbol{p}_k$. Then we compute $a_{k,\ell}$ as

$$a_{k,\ell} = \sum_{b \neq \mathcal{T}} \boldsymbol{p}_k(b) d(b, \beta^*(\boldsymbol{p}_k)) - \sum_{b \neq \mathcal{T}} \boldsymbol{p}'_k(b) d(b, \beta^*(\boldsymbol{p}'_k)) \tag{4.36}$$

We now have an approximation sequence $\hat{a}_{k,\ell}$ and an observation sequence $a_{k,\ell}$. At time $k-1$, our goal is to choose T^{Δ} to minimize

$$\mathsf{E}\left[(\hat{a}_{k,\ell} - a_{k,\ell})^2\right] \tag{4.37}$$

We apply the Robbins-Monro algorithm [135], a form of stochastic gradient descent, to this problem in order to recursively compute a sequence of T^{Δ} that will hopefully solve this minimization problem for large k. The update equation is

$$T^{\Delta}_k(b, \ell) = T^{\Delta}_{k-1}(b, \ell) - 2\alpha_k \mathbb{1} b \neq \mathcal{T} \boldsymbol{p}_{k-1}(b)(\hat{a}_{k,\ell} - a_{k,\ell}) \tag{4.38}$$

where α_k is a step size. Note that $\mathbb{1} b \neq \mathcal{T} \boldsymbol{p}_{k-1}(b)$ is the gradient of $\hat{a}_{k,\ell}$ with respect to $T^{\Delta}(b, \ell)$.

Using a constant step size in our simulations, we could only observe small oscillations in the values of T^{Δ}. If $\mathcal{B}$ is not finite, then we can again parameterize T^{Δ}. The Robbins-Monro algorithm can be applied in this context as well, although the gradient expressions will depend on the type of interpolation used.

1.2.5. A Lower Bound

Deriving a lower bound is generally difficult for the considered problem. However, we derived a lower bound for the special case of a discrete state space with Gaussian observations. The idea is to combine the observable-after-control assumption with a separable lower bound on the tracking cost as we demonstrate in what follows. When awake, the sensors observations are Gaussian, i.e.,

$$s_{k,\ell} \sim \mathcal{N}\left(\frac{V}{(\nu_\ell - b_k)^2 + 1}, 1\right) \tag{4.39}$$

where ν_ℓ is the location of sensor ℓ and V some positive constant.

First, the following Lemma provides a lower bound on the expected tracking cost.

Lemma 1.1. *Given the current belief $\boldsymbol{p}_k$, an action vector $\boldsymbol{u}_k$, the current residual sleep times vector $\boldsymbol{r}_k$, the Gaussian observation model in (4.39), the Hamming cost definition in (4.24), and a mean received signal strength $\boldsymbol{m}_j$ when the target is at state j, the expected tracking cost is lower bounded by*

$$E[d(\hat{b}_{k+1}, b_{k+1})|\boldsymbol{p}_k, \boldsymbol{u}_k] = \sum_{i=1}^{m} \boldsymbol{p}_k(i) \sum_{j=1}^{m} p(b_{k+1} = j | b_k = i) \max_{k \neq j} Q\left(\frac{\boldsymbol{d}_{kj}}{2} + \frac{\ln \frac{\pi_j}{\pi_k}}{\boldsymbol{d}_{kj}}\right) \tag{4.40}$$

where m is the size of the discrete state space, i.e., the number of possible object locations, $\boldsymbol{d}_{kj}^2 = \frac{\boldsymbol{\Delta m}_{kj}^T \boldsymbol{\Delta m}_{kj}}{\sigma^2}$, $\boldsymbol{\Delta m}_{kj} = \boldsymbol{m}_k - \boldsymbol{m}_j$, and $Q(.)$ is the normal distribution Q-function.

Since the mean received signal strength depends on whether the sensors are awake or asleep, the distance $\boldsymbol{d}_{kj}$ is a function of the next step residual sleep vector $\boldsymbol{r}_{k+1}$. To highlight this dependence, we will sometimes use the notation $\boldsymbol{d}_{kj}(\boldsymbol{r})$ when needed.

The next step is to use this result to compute a separable bound on the tracking error, which combined with an observable-after-control assumption would lead to a decomposable lower bound on the optimal value function. The idea is to separate out the contribution of every sensor by assuming that every other sensor is awake and study the tracking error when that sensor is awake or asleep. Our next Lemma establishes a separable lower bound on the expected tracking cost.

Lemma 1.2. *The expected tracking cost is lower bounded by*

$$E[d(\hat{b}_{k+1}, b_{k+1})|\boldsymbol{p}_k, \boldsymbol{u}_k, \boldsymbol{r}_k] \geq \sum_{\ell=1}^{n} \lambda_\ell(\boldsymbol{p}_k) \left\{ \mathbb{1} r_{k+1,\ell} = 0 \sum_{i=1}^{m} \boldsymbol{p}_k(i) T_0(\boldsymbol{p}, i, \ell) + \mathbb{1} r_{k+1,\ell} > 0 \sum_{i=1}^{m} \boldsymbol{p}_k(i) T(\boldsymbol{p}, i, \ell) \right\} \tag{4.41}$$

where

$$T_0(\boldsymbol{p}; i, \ell) \triangleq \sum_{j=1}^{m} p(b_{k+1} = j | b_k = i) \max_{k \neq j} Q\left(\frac{\boldsymbol{d}_{kj}(\boldsymbol{0})}{2} + \frac{\ln \frac{[\boldsymbol{p}P]_j}{[\boldsymbol{p}P]_k}}{\boldsymbol{d}_{kj}(\boldsymbol{0})}\right)$$

$$T(\boldsymbol{p}; i, \ell) \triangleq \sum_{j=1}^{m} p(b_{k+1} = j | b_k = i) \max_{k \neq j} Q\left(\frac{\boldsymbol{d}_{kj}(\boldsymbol{0}_{-\ell})}{2} + \frac{\ln \frac{[\boldsymbol{p}P]_j}{[\boldsymbol{p}P]_k}}{\boldsymbol{d}_{kj}(\boldsymbol{0}_{-\ell})}\right)$$

where $\boldsymbol{0}$ is the all zero vector and $\boldsymbol{0}_{-\ell}$ denotes a vector of length n with all entries equal to zero except for the ℓ-th entry which can be anything greater than 0.

Intuitively, $T_0(\boldsymbol{p}; i, \ell)$ represents the contribution of sensor ℓ to the total expected tracking cost when the underlying state is i, the belief is $\boldsymbol{p}$ and when all sensors are awake. On the other hand $T(\boldsymbol{p}; i, \ell)$ is the ℓ-th sensor contribution when it is asleep and all the other sensors are awake.

We can readily state a lower bound on the optimal value function in the following Theorem.

Theorem 1.1. *A lower bound on the optimal value function at belief state $\boldsymbol{e}_b$ can be obtained as a solution to the following optimization problem*

$$J(\boldsymbol{e}_b) = \max_{\Lambda(c)} \sum_{\ell=1}^{n} \min_{u_\ell^b} \left\{ \sum_{j=0}^{u-1} \sum_{i=1}^{m} [\boldsymbol{e}_b P^j]_i \lambda(i,\ell) T(i,\ell) + \sum_{i=1}^{m} [\boldsymbol{e}_b P^u]_i \lambda(i,\ell) T_0(i,\ell) \right. \\ \left. + c \sum_{i=1}^{m} [\boldsymbol{e}_b P^{u+1}]_i + \sum_{i=1}^{m} [\boldsymbol{e}_b P^{u+1}]_i J^\ell(\boldsymbol{e}_i) \right\} \tag{4.42}$$

$$\textit{subject to} \quad \Lambda \mathbf{1}_n = \mathbf{1}_m$$

where $\mathbf{1}_m$ is a column vector of all ones of length m. The matrix Λ is defined for every value of c, where $\Lambda(c)$ is an $m \times n$ matrix with the (i,ℓ) entry equal to $\lambda(i,\ell)$, i.e., $\Lambda(c) = \{\lambda(i,\ell)\}$. The quantities $T(i,\ell)$ and $\lambda(i,\ell)$ are shorthand for $T(\boldsymbol{e}_i; i, \ell)$ and $\lambda_\ell(\boldsymbol{e}_i)$, respectively. The inner minimization is over the control action u_ℓ^b for sensor ℓ given a belief state $\boldsymbol{e}_b$.

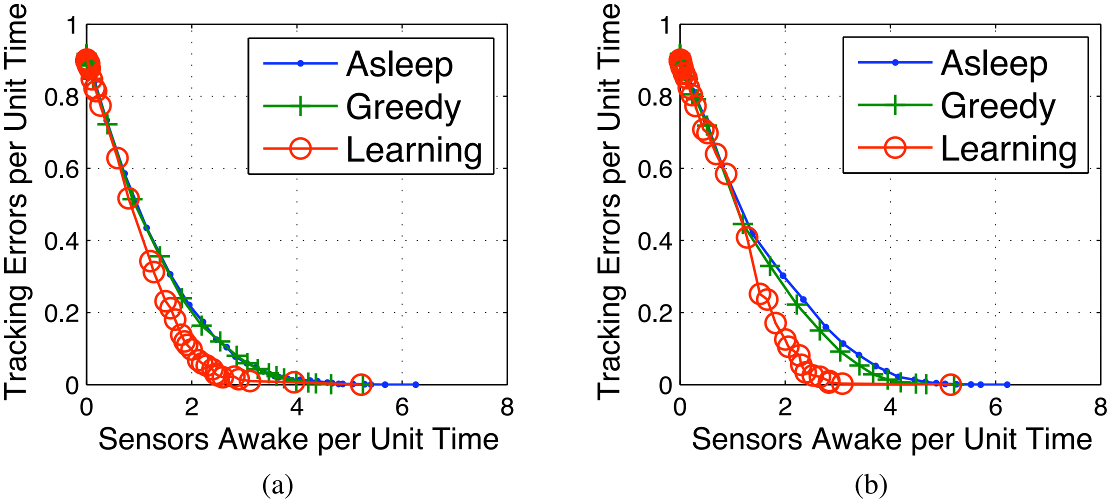
A closed form solution for (4.42) cannot be obtained, and hence, we solve for $J(\boldsymbol{e}_b)$ numerically. First, we fix Λ and use policy iteration to solve for the control of each sensor at each state. Then, we change Λ and repeat the process. The envelope of the generated value functions (corresponding to different instants of Λ) is hence a lower bound on the optimal value function.

1.2.6. Numerical Results

In this section, we show some simulation results illustrating the performance of the described policies. For the non-learning policies, the value of $T^\Delta(b,\ell)$ for each b and ℓ was generated using 200 Monte Carlo simulations. The results of 50 simulation runs were averaged when plotting the curves. For the learning policies, the values for T^Δ were initialized to those obtained from the non-learning approach using greedy sensor selection as a baseline. A constant step size of 0.01 was used in the learning algorithm.

We first consider a simple network that we term Network A. This is a one-dimensional network with 41 possible object locations where the object moves with equal probability either one to the left or one to the right in each time step. There is a sensor at each of the 41 object locations that makes (when awake) a binary observation that determines without error whether the object is at that location. Hamming cost is used for the tracking cost.

For Network A, we illustrate the performance of the Q$_{\text{MDP}}$ versions of our policies in Figure 4.4(a) and the FCR versions of our policies in Figure 4.4(b).



(a) (b)

Figure 4.4: Tradeoff curves for Network A: (a) Q$_{\text{MDP}}$ policies; (b) FCR policies

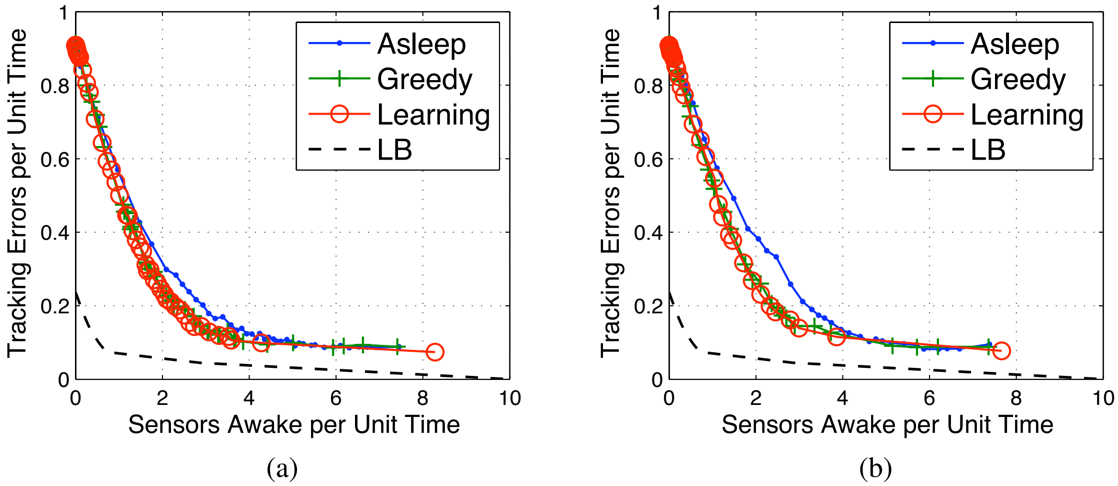


Figure 4.5: Tradeoff curves for Network B and a lower bound: (a) Q_{MDP} policies; (b) FCR policies

The curves labeled “Asleep” are for the nonlearning approach for computing T^{Δ} where we assume that all sensors are asleep as a baseline. The curves labeled “Greedy” are for the nonlearning approach for computing T^{Δ} where we use a greedy algorithm to determine our baseline. The curves labeled “Learning” employ our learning algorithm for computing T^{Δ}. From the tradeoff curves, it is apparent that using the learning algorithm to compute T^{Δ} results in improved performance. A close inspection of Figures 4.4(a) and 4.4(b) will reveal that the Q_{MDP} policies perform somewhat better than their FCR counterparts. This is consistent with what was observed earlier.

Table 4.1: Object movement for Network B.

Change in Position	0	1	2	3
Probability	0.3125	0.2344	0.0938	0.0156

We then consider a new one-dimensional network termed Network B. The possible object locations are located on the integers from 1 to 21. The object moves according to a random walk anywhere from three steps to the left to three steps to the right in each time step. The distribution of these movements is given in table 4.1. The change in position indicate movement by a corresponding number of steps to the right or to the left. There are 10 sensors in this network so that $m \neq n$. The locations of the sensors are given in Table 4.2 and awake sensors make Gaussian observations as in (4.39). Results for the Q_{MDP} and FCR versions of our policies are shown in Figures 4.5(a) and 4.5(b), respectively. The results confirm the same general trends observed for Network A. The figures also show our derived lower bound on the energy-tracking tradeoff. Not surprisingly, the lower bound is particularly loose at the high tracking cost regime, yet the gap is reasonably small for the low tracking error region. This is expected since the lower bound uses an all-awake assumption to lower bound the contribution of each sensor to the tracking error. However, it is worth mentioning that we can exactly compute the saturation point for the optimal scheduling policy, which matches the saturation limit of the shown curves, since every policy has to eventually meet the all-asleep performance curve when the energy cost per sensor is high. At that point, all sensors are put to sleep and

Table 4.2: Sensor locations for Network B.

Sensor	1	2	3	4	5	6	7	8	9	10
Location	1.36	1.61	3.91	8.09	11.96	13.39	13.52	13.66	16.60	18.68

hence the target estimate can only be based on prior information. The small gap at the low tracking error regime combined with the aforementioned saturation effect highlight the good performance of our sleeping policies.

Our results are not restricted to 1-D networks but easily extend to 2-D networks. Namely, Figure 4.6(right) shows the energy-tracking tradeoff of the Q_{MDP} and FCR policies for the 2-D network of Figure 4.6(left) with continuous observations and Hamming cost. To demonstrate that our techniques can be applied to an object that moves on a continuum, we define a new network, Network C. This network is identical to Network B except for two changes. First, the object can take locations anywhere on the interval $[1, 21]$. Second, the object moves according to Brownian motion with the change in position between time steps having a Gaussian distribution with mean zero and variance 1. As mentioned earlier, only FCR policies can be generated for this type of network. Values of T^{Δ} were computed for each integer-valued object location on $[1, 21]$ and linear interpolation used to compute values of T^{Δ} for other object locations. Since continuous distributions cannot be easily stored, particle filtering techniques were employed. The number of particles used was 512 and resampling was performed at each time step. Tradeoff curves for Network C are shown in Figure 4.7. Although the tradeoff curves are less smooth than before, this figure illustrates performance trends similar to those already seen. The reason the curves are not as smooth is that occasionally the particle filter would fail to keep track of the distribution with sufficient accuracy. This would cause the network to lose track of the object and cause abnormally bad tracking for that simulation run.

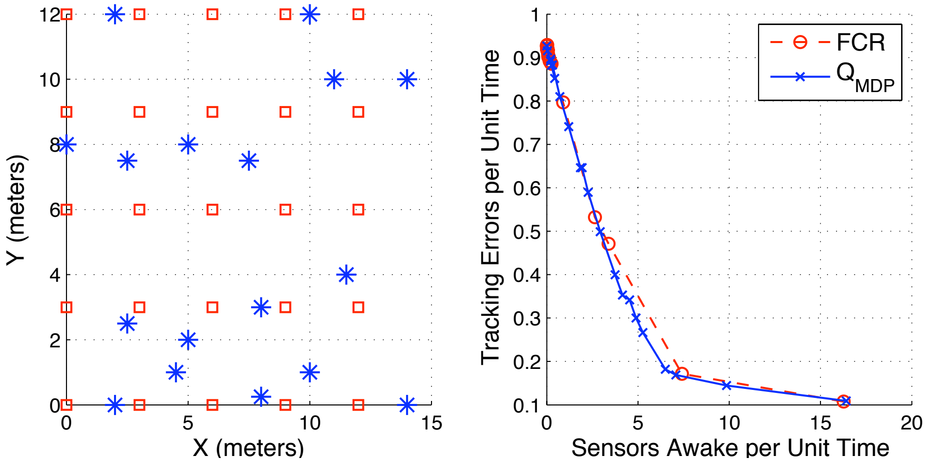


Figure 4.6: (Left) 2-D network with 17 sensors (stars) and 25 possible object locations (squares). (Right) Energy-Tracking tradeoff of the Q_{MDP} and FCR sleeping policies for a 2D network with continuous observations and Hamming cost.

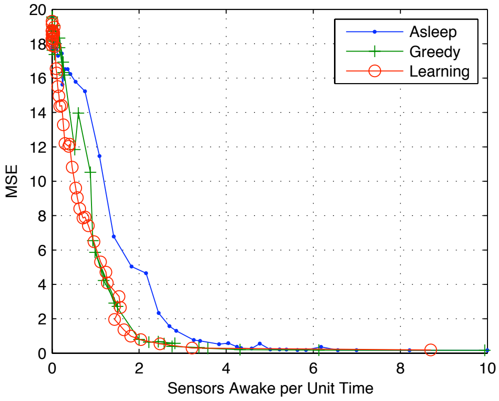


Figure 4.7: Tradeoff curves for FCR policies for Network C.

1.3. Sensor Sleeping for Multi-target Tracking

We extend our analysis to the tracking of multiple objects [59]. A discussion of the tracking of multiple objects, often termed multitarget tracking (MTT), can be found in [92]. Tracking multiple objects is not a simple extension of tracking a single object due to the *data association* problem. This problem arises whenever the identity of the objects cannot be determined from the observations. Thus, even if all locations where objects are located are known exactly, it may not be known which location corresponds to which object. This uncertainty leads to an explosion in the set of possibilities that must be considered and makes optimal solution difficult. Suboptimal tracking algorithms are then needed. We therefore formulate a design problem wherein we keep track of the full joint distribution for the object locations. This approach is optimal but can quickly become intractable for large numbers of objects. Our simulation results will focus on the two-object case to make computation as simple as possible. However, most of our analysis applies to the general q-object case and we indicate how our solutions scale with increasing number of objects. The results of our work are a set of suboptimal sleeping policies. These policies are compared with lower bounds on optimal performance that we derived in the course of our analysis. Our simulation results show how our suboptimal policies compare with optimal performance. Our policies also perform significantly better than naive approaches that do not use information about the locations of the objects. Furthermore, one of our policies uses only information about the marginal distributions of the objects and thus scales well with increasing numbers of objects and can be used in concert with suboptimal tracking algorithms.

We stick to the simplified model of Section 1.1 with non-overlapping sensing regions. An awake sensor can only detect whether one or more objects is within its range and can detect neither the exact number of objects present nor which objects are present.

We are interested in tracking q objects that move independently according to their individual first-order Markov models. We will write the combined state of the q objects as a vector of length q. There are $(n+1)^q$ possible states for this vector. The state $\boldsymbol{\mathcal{T}} \equiv [n+1,\dots,n+1]$ is the terminal state that occurs when all objects have left the network. Once this state is reached, no further cost is incurred.

If we denote the observation available to the central unit at time k by z_k, then we have $z_k = (\boldsymbol{s}_k, \boldsymbol{r}_k)$, where $\boldsymbol{s}_k$ is a $(n+1)$-vector of observations with

$$s_{k,\ell} = \begin{cases} 0 & \text{if } r_{k,\ell} = 0 \text{ and no objects are at location } \ell \\ 1 & \text{if } r_{k,\ell} = 0 \text{ and one or more objects are at location } \ell \\ \mathcal{E} & \text{if } r_{k,\ell} > 0 \end{cases} \tag{4.43}$$

where $\mathcal{E}$ is an erasure that provides no information. We can further decompose tracking error into two components. The first component is *observation error* that occurs when we fail to observe a particular object. The second component is *data association error* that occurs when the objects have been misidentified. To perform the object identification, we define the vector of estimated object locations at time k to be $\hat{\boldsymbol{b}}_k$. We can think of $\hat{\boldsymbol{b}}_k$ as an additional control input that is a function of I_k, i.e.

$$\hat{\boldsymbol{b}}_k = \beta_k(I_k) \tag{4.44}$$

We combine observation and data association errors by defining a tracking error to have occurred when either an observation error or a data association error has occurred. A cost of 1 is incurred for each tracking error. Thus the tracking cost can be written as

$$\sum_{i=1}^{q} \left(1 - \mathbb{1} r_{k,b_{k,i}} = 0 \mathbb{1} \hat{b}_{k,i} = b_{k,i}\right) \tag{4.45}$$

Recall that the $\hat{\boldsymbol{b}}_k$ input does not affect the state evolution; it only affects the cost. We can therefore compute the optimal choice of $\hat{\boldsymbol{b}}_k$, denoted as $\beta_k^*(I_k)$, using an optimization minimizing the tracking error over a single time step. We can thus write

$$\beta^*_k(I_k) = \arg\min_{\hat{\boldsymbol{b}}} \mathsf{E}\left[\sum_{i=1}^{q}\left(1 - \mathbb{1} r_{k,b_{k,i}} = 0\mathbb{1}\hat{b}_i = b_{k,i}\right)\Big| I_k\right] \tag{4.46}$$

we can write the total cost for time step k as

$$g(\boldsymbol{b}_k, I_k) = \mathbb{1}\boldsymbol{b}_k \neq \boldsymbol{\mathcal{T}}\left(\sum_{\ell=1}^{n} c\mathbb{1} r_{k,\ell} = 0 + \sum_{i=1}^{q}\left(1 - \mathbb{1} r_{k,b_{k,i}} = 0\mathbb{1}\beta^*_k(I_k) = b_{k,i}\right)\right) \tag{4.47}$$

Since g is bounded by $(cn + q)$ and the expected time till the object leaves the network is finite, the cost function J is well-defined.

The evolution of p_k is difficult to write mathematically, but it is a standard nonlinear filtering operation. One example of a distribution update is shown in Figure 4.8.

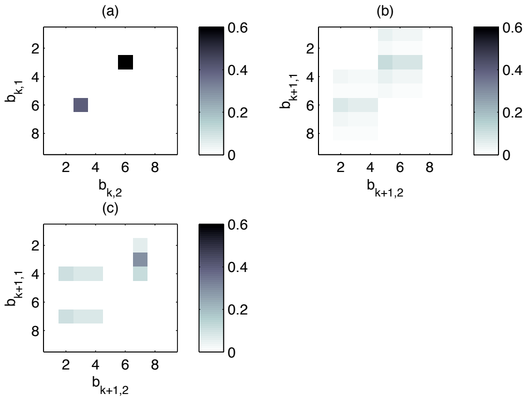


Figure 4.8: An example of a distribution update for $q = 2$ and $n = 9$. In each subfigure, a joint distribution for the objects is shown. In (a), it is known that one object is located at position 3 and one is located at position 6. In (b), the joint distribution at time $k + 1$ before incorporating observations is shown. In generating (c), we suppose that sensors 1, 5, 6, and 8 are awake and have failed to observe the object. The distribution in (c) is the one that results from incorporating these observations.

For notational convenience, we also define

$$p_{k,i}(b) = \sum_{\boldsymbol{b}':b'_i = b} p_k(\boldsymbol{b}') \tag{4.48}$$

In other words, $p_{k,i}$ is the marginal distribution for object i. Each component of the vector valued function β^*_k can be chosen according to

$$\beta^*_{k,i}(p_k, \boldsymbol{r}_k) = \arg\max_{\hat{b}} \mathbb{1} r_{k,\hat{b}} = 0 p_{k,i}(\hat{b}) \tag{4.49}$$

In other words, for each object we select the estimated object location from among the locations where a sensor is awake. From these locations, we select the one with the largest value of the marginal distribution for that object. Note that β^*_k has the same form for every k so we can drop the subscript and refer to β^*_k as β^*. Our approach to generating suboptimal solutions is similar to that used earlier. Namely, we make unrealistic assumptions that greatly simplify the evolution of p_k. These assumptions also allow the tracking and energy costs to be written as a sum of costs, one for each sensor. The result is that the problem then separates into a set of n simpler subproblems, one for each sensor, that can be more easily solved.

The $\mathrm{Q_{MDP}}$ policy is obtained as a solution of the per-sensor Bellman equation for sensor ℓ

$$J^{(\ell)}(p) = \min_{u} \sum_{\boldsymbol{b}\neq\boldsymbol{\mathcal{T}}} \left(\sum_{j=1}^{u} (pP^j)(\boldsymbol{b}) \sum_{i=1}^{q} \mathbb{1} b_i = \ell + c(pP^{u+1})(\boldsymbol{b}) + (pP^{u+1})(\boldsymbol{b}) J^{(\ell)}(\delta_{\boldsymbol{b}}) \right) \tag{4.50}$$

Note that this assumption implies that there will be no future data association errors and thus the only tracking costs present in designing this policy are observation errors.

We also define an FCR policy that is to select the first value of u such that

$$\sum_{i=1}^{q} \sum_{\boldsymbol{b}:b_i=\ell} (pP^{u+1})(\boldsymbol{b}) \geq \sum_{i=1}^{q} \sum_{\boldsymbol{b}:b_i\neq n+1} \frac{c}{q}(pP^{u+1})(\boldsymbol{b}) \tag{4.51}$$

The lower bound that results from the $\mathrm{Q_{MDP}}$ policy is likely to be loose when data association errors dominate the tracking cost. Hence, we designed a $\mathrm{Q_{MDP}}$ -like policy that, instead of assuming the state is known, assumes that

- at the current time step, after selecting sleep times all sleeping sensors will be allowed to make observations (with no energy cost), and
- at future time steps, the distribution for the object location will evolve as if all sensors are awake.

This gives rise to the term All Awake (AA policy). Note that the AA assumption is like assuming we will have "perfect observations." However, this does not imply perfect knowledge of the state due to the presence of data association errors. Note that since we are assuming more information than is actually available, the AA assumption does yield a lower bound on optimal performance. Due to complexity issues, we only designed the AA policy for the $q = 2$ case.

The advantage of the AA assumption is that it allows us to considerably simplify the state space for p_k. Since all the sensors come awake at each time step, the set of at most two locations where an object could be present is known exactly. Suppose for the moment that there are two distinct locations where an object is observed. Let $\tilde{\boldsymbol{b}}_{\boldsymbol{k}} = (\tilde{b}_{k,1}, \tilde{b}_{k,2})$ with $\tilde{b}_{k,1} < \tilde{b}_{k,2}$ being the locations where objects are present at time k. Thus, $\tilde{\boldsymbol{b}}_{\boldsymbol{k}}$ belongs to a subset of $\{1, \ldots, n+1\}^2$. To completely characterize p_k we need only specify the probability that $\boldsymbol{b}_k = \tilde{\boldsymbol{b}}_k$. Denote this probability as d_k. Then with probability $1 - d_k$ we have that $\boldsymbol{b}_k = (\tilde{b}_{k,2}, \tilde{b}_{k,1})$. Note that if there is only one distinct location where an object is observed we can simply let $\tilde{b}_{k,1} = \tilde{b}_{k,2}$ and $d_k = 1$.

The state space for $\tilde{x}_k = (\tilde{\boldsymbol{b}}_k, d_k)$ is not finite due to $d_k \in [0, 1]$. The approach we take is to quantize d_k and construct a kernel $\tilde{P}$ for this quantized version of $\tilde{x}_k$. Note that in doing this we no longer have a true lower bound; however, with fine enough quantization we can well approximate such a lower bound.

We define functions T^W and T^S such that $T^W(\tilde{x}_k, \ell)$ and $T^S(\tilde{x}_k, \ell)$ are the tracking costs incurred by sensor ℓ when it is awake and asleep, respectively. We showed that the per-sensor Bellman equation for sensor ℓ under the above assumptions is given by

$$J^{(\ell)}(\tilde{p}) = \min_{u} \sum_{\tilde{x}\neq\tilde{\mathcal{T}}} \left(\sum_{j=1}^{u} (\tilde{p}\tilde{P}^j)(\tilde{x}) T^S(\tilde{x}, \ell) + (\tilde{p}\tilde{P}^{u+1})(\tilde{x}) T^W(\tilde{x}, \ell) + c(\tilde{p}\tilde{P}^{u+1})(\tilde{x}) + (\tilde{p}\tilde{P}^{u+1})(\tilde{x}) J^{(\ell)}(\delta_{\tilde{x}}) \right) \tag{4.52}$$

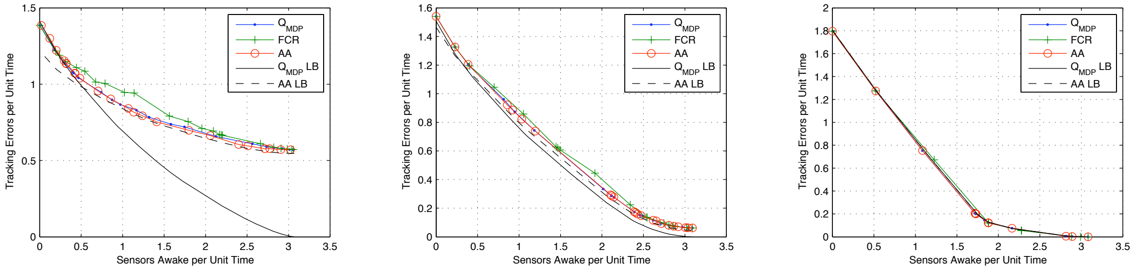
This equation can be solved through the use of policy iteration to yield a policy and a lower bound.

1.3.1. Numerical Results

In this section, we give some simulation results that illustrate the performance of the policies we derived in previous sections.

We first consider a network we term Network A. This network is a one-dimensional network with seven sensors. The small number of sensors was needed because the AA policy must perform policy iteration for a number of states equal to $\frac{n(n+1)}{2}\lambda + n + 1$ where λ is the number quantization levels for d_k. In our simulations, we used $\lambda = 21$ for a total of 596 states. Policy iteration for this number of states required significant computation and the network could not be made much larger. A value of $u_{max} = 50$ was used in computing the AA policy. The object movement in Network A is parametrized by a scalar $a \in [0, 1]$. Object 1 moves one cell to the left with probability a and one cell to the right with probability $1 - a$ in each time step. Object 2 does just the opposite, moving one cell to the left with probability $1 - a$ and one cell to the right with probability a. Note that the closer a is to 0.5, the more difficult it is to distinguish between the objects based on their movements. This means that by varying a we can investigate the performance of our policies for various amounts of data association error.

We illustrate the performance of our policies for Network A for the cases $a = 0.55$, $a = 0.75$, and $a = 0.95$ in Figure 4.9. Curves are shown for the $\mathrm{Q_{MDP}}$ and AA lower bounds as well as the $\mathrm{Q_{MDP}}$, FCR, and AA policies. The curves are tradeoff curves that examine the tradeoff between energy cost and tracking cost as the parameter c is varied. In examining the tradeoff curves, the distance from the right-hand point of each curve to the x-axis is the average number of data association errors when all sensors are awake. No tracking error smaller than this can be achieved. From the figures we can draw the following conclusions:



(a) a = 0.55 (b) a = 0.75 (c) a = 0.95

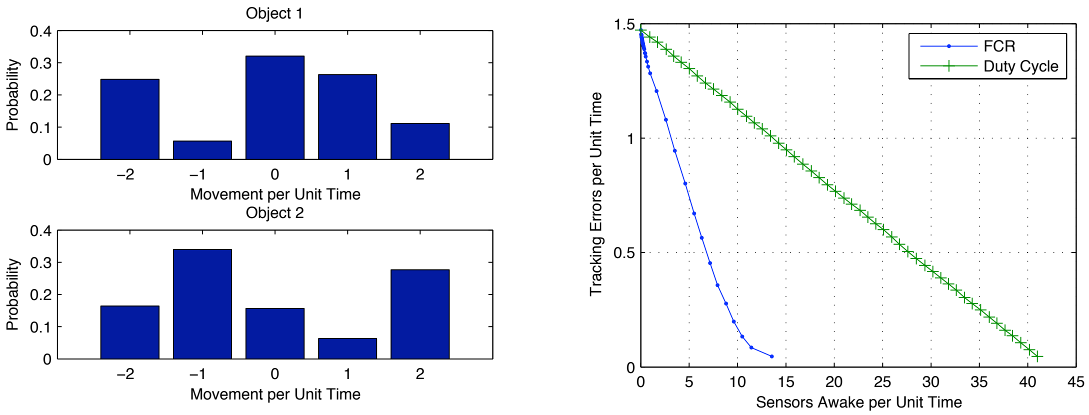
Figure 4.9: Performance of $\mathrm{Q_{MDP}}$, FCR, and AA policies for Network A for various values of a.

- The lower bound due to the $\mathrm{Q_{MDP}}$ assumption is tight when only a few sensors are awake (large c). This is because the $\mathrm{Q_{MDP}}$ assumption incorporates only observation errors and when few sensors are making observations, observation errors dominate.
- The lower bound due to the AA assumption is tight only when many sensors are awake (small c). This is because the tracking cost approximation used in computing the AA policy is loose when neither object is observed.
- The $\mathrm{Q_{MDP}}$ policy performs best when only a few sensors are awake, and the AA policy performs best when many sensors are awake. Not surprisingly, these are the same regions where their bounds are tight.
- The FCR policy is the worst-performing policy. The difference between the FCR policy and the other policies, while never especially large in terms of the tradeoff curves, shrinks as data association errors become small.

We now turn our attention to simulating the FCR policy, which scales better than the other policies, for a larger network, termed Network B. Network B is a one-dimensional network with 41 sensors. Note that policy iteration for the $\mathrm{Q_{MDP}}$ policy would need to be performed over $(n + 1)^2 = 1764$ states and the requirements for the AA policy would be even larger. The distributions for the movement of the objects are given in Table 4.3 and illustrated graphically in Figure 4.10(a). Since no lower bounds are available for Network B, we compare the performance of our FCR policy to a duty cycle policy, where each sensor comes awake with some fixed probability at each time step. Figure 4.10(b) shows tradeoff curves for these

Table 4.3: Object movement distributions for Network B.

Change in Position	-2	-1	0	+1	+2
Probability for Object 1	0.2482	0.0568	0.3205	0.2633	0.1112
Probability for Object 2	0.1641	0.3395	0.1566	0.0633	0.2765



(a) Object movement distributions for Network B

(b) Tradeoff curves for FCR and duty cycle policies for Network B

Figure 4.10: Tradeoff curves and object movement distribution for Network B.

two policies. The tradeoff curve for the duty cycle policy is generated by varying the probability that a sensor is awake between 0 and 1. The FCR tradeoff curve significantly outperforms the duty cycle policy.

2. Sensor Scheduling

Here, we consider a scheduling variant of the problem which can be thought of as a sleeping problem with an external wake-up mechanism, i.e., sensors can be woken up by external means (e.g. a low-power wake-up radio). At time k, the permissible control actions for an n-sensor scheduling problem are n-dimensional binary vectors, i.e., vectors in $\{0,1\}^n$ (corresponding to the set of sensor nodes to activate at each time step), in contrast to vectors in $\mathbb{N}_0^{n_a(k)}$ for the sleeping problem (corresponding to the sleep durations of awake sensors), where $\mathbb{N}_0$ is the set of non-negative integers and $n_a(k)$ the number of awake sensors at time k. The simpler structure of the control space for the scheduling problem does not address the combinatorial nature of the control space, yet it enables efficient approximate solution methodologies [8].

Again, we adopt a bottom-up approach where we consider a range of sensing, motion and cost models with increasing levels of difficulty and devise suboptimal scheduling policies to balance the tradeoff between energy expenditure and tracking performance. In some cases we are also able to derive lower bounds on the optimal energy-tracking tradeoff. In addition to the simple sensing model of Section 1.1, we also consider more generalized models.

2.1. Overlapping Sensors with Discrete Observations Models

In this model, we continue to use a discrete model for the target transition but we redefine a new sensing model and cost structure to account for the fact that sensors could have overlapping visibility regions. Within that model we further consider simple and probabilistic sensing. Simple sensing refers to the case where the target is *perfectly* observed within the visibility region of *any* active sensor. Therefore, a tracking error is incurred if none of the sensors observing the current target location is active. Redefining the cost structure for this model:

$$g(b_k, \boldsymbol{u}_{k-1}) = \mathbb{1}b_k \neq \tau \left(\mathbb{1}u_{k-1,j} = 0, \forall j \in \mathcal{B}_{b_k} + \sum_{\ell=1}^{n} c\mathbb{1}u_{k-1,\ell} = 1 \right) \tag{4.53}$$

By probabilistic sensing we account for observation uncertainty even if the target is within the visibility region of one or more active sensors. In particular, we assume that the observation is uniformly distributed over a location set $\mathcal{R}_k$ (other than the true target location) that belong to the visibility regions of the set of awake sensors monitoring the true location b_k. The number of these locations is function of the control $\boldsymbol{u}_{k-1}$ and the object state b_k. If the true target location does not belong to the visibility region of an awake sensor, we naturally exclude the visibility region of that sensor since no measurement is received from such a sensor. When $\mathcal{R}_k$ is a singleton $\{b_k\}$, we set $q = 1$. A tracking error is incurred if the target is not directly observed and the uncertainty in the target location cannot be resolved.

2.2. Continuous Observation, Continuous State and Arbitrary Cost Models

In this class of models, we allow for an arbitrary distribution of the observations given the current object location. Tracking cost is modeled through an arbitrary distance measure between the actual and the estimated object location. If we denote the set of possible object locations $\mathcal{B}$, we have $\mathcal{B} = m + 1$. In contrast to the simplistic model, m is different from n since object locations are arbitrary and we no longer assume one location corresponds to the sensing range of one particular sensor. The $(m + 1)$-th state again corresponds to a termination state. Furthermore, the target can be moving on a continuous state space in which case m is ∞.

For simplicity of exposition, we focus on discrete state spaces. Also, we omit indexing time whenever the time evolution is well-understood to avoid cumbersome notation. We consider the following observation model for illustration; however, our approach is fairly general:

$$p(\boldsymbol{s}|b, \boldsymbol{u}) = \prod_{i=1}^{n} \left\{ \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(s_i - \frac{V}{(b - p_i)^2 + 1} \right)^2 \right) \mathbb{1}u_i = 1 + \delta(s_i - \varepsilon)\mathbb{1}u_i = 0 \right\} \tag{4.54}$$

where $\boldsymbol{s}$ is an $n \times 1$ continuous observation vector with the i-th entry, s_i, representing the observation of sensor i, $p_i, i = 1, \ldots, n$, is the position of the i-th sensor, b is the target state, V is some positive constant, ε stands for erasure, and $\delta(.)$ is the Dirac Delta function. In (4.54), the observation of an active sensor is Gaussian with a mean received signal strength inversely proportional to the square of the distance between the sensor and the actual target location. The observation of an inactive sensor is just an erasure.

As for the sleeping problem we define the tracking error through an arbitrary bounded distance function $d(b, \hat{b})$ between the actual and the estimated object locations, which can be the Hamming distance or the Euclidean distance for discrete and continuous state spaces, respectively.

2.2.1. Approximate Scheduling Policies

There are a number of algorithms for solving POMDPs exactly [31, 73, 156]. These algorithms rely on the powerful result of Sondik that the optimal value function for any POMDP can be approximated arbitrarily closely using a set of hyper-planes (α-vectors) defined over the belief simplex [156]. The result is a value function parameterized by a number of hyperplanes (or vectors) whereby the belief space is partitioned into a finite number of regions. Each vector minimizes the value function over a certain region of the belief space and has a control action associated with it, which is the optimal control for the beliefs in its region.

To clarify, in value iteration we generally start with some initial estimate for J^* and repeatedly apply the transformation defined by the right hand side of the Bellman equation until the sequence of cost functions converges. Let $\{\boldsymbol{\alpha}_i^{(k)}\}_{i=1}^{|J^{(k)}|}$ denote the set of vectors parameterizing the value function $J^{(k)}$ after k iterations, where $|J^{(k)}|$ is the total number of hyperplanes, and $\boldsymbol{\alpha}_i^{(k)}(b)$, which is a hyperplane in the belief space, represents the value of executing the k-step policy associated with the i-th vector starting from a state b.

Hence, the value of executing the i-th hyperplane policy starting from a belief state $\boldsymbol{p}$ is simply the dot product of $\boldsymbol{\alpha}_i$ and $\boldsymbol{p}$:

$$J_i^{(k)}(\boldsymbol{p}) = \sum_b \boldsymbol{p}(b)\boldsymbol{\alpha}_i^{(k)}(b) = \boldsymbol{p} \cdot \boldsymbol{\alpha}_i^{(k)}.$$

Therefore, the value of the optimal k-step policy starting at $\boldsymbol{p}$ is simply the minimum dot product over all hyperplanes, i.e.,

$$J^{*(k)}(\boldsymbol{p}) = \min_{\{\boldsymbol{\alpha}_i^{(k)}\}} \boldsymbol{p} \cdot \boldsymbol{\alpha}_i^{(k)}.$$

Hence, $J^{*(k)}(\boldsymbol{p})$ is piecewise linear and concave. Some of the vectors (also known as policy trees) may be dominated by others in the sense that they are not optimal at any region in the belief simplex. Thus, many exact algorithms devise pruning mechanisms whereby a parsimonious representation with a minimal set of non-dominated hyperplanes is maintained.

Even though the aforementioned linearity/concavity property makes the policy search a great deal simpler, the exact computation is generally intractable except for relatively small problems. The two major difficulties for exact computation arise from the exponential growth of the vectors with the planning horizon and with the number of observations, and the inefficiencies related to identification of such vectors and subsequently pruning them. Namely, the number of hyperplanes grows double exponentially such that after k steps the number of hyperplanes is $O\left(|\mathcal{U}|^{|\mathcal{S}|^k}\right)$, where $|\mathcal{U}|$ and $|\mathcal{S}|$ denote the cardinality of the control and observation spaces, respectively. Equivalently, the number of hyperplanes per iteration grows as:

$$|J^{(k+1)}| = O\left(|\mathcal{U}||J^{(k)}|^{|\mathcal{S}|}\right).$$

This has led to a number of approximations and suboptimal solutions techniques that trade off solution quality for speed.

Remark 2.1. The intractability of the optimal solution for our problem is primarily due to the following reasons:

(i) The cost function is minimized over the simplex of probability distributions, i.e., the $(m-1)$-dimensional belief simplex for m-state discrete state-space models, and the space of probability density functions for continuous state-space models.

(ii) The exponential explosion of the action space with the number of sensors (2^n actions).

(iii) The exponential growth of the α-vectors with the planning horizon and with the number of observations, especially for continuous observation models.

In addition to the $\mathrm{Q_{MDP}}$ strategy, we also develop sensor scheduling strategies based on point-based approximations. Despite the fact that the generated $\mathrm{Q_{MDP}}$ based policies perform reasonably well, generally the resulting policies would not take actions to gain information (an effect of the observable-after-control assumption), leading to situations wherein the belief state does not get updated appropriately. Furthermore, while decoupling the scheduling problem provides close-to optimal performance for uncoupled or lightly-coupled sensing and tracking models (see Section 3), it might come at the expense of reduction in solution quality for more realistic or heavily-coupled models. While our previous approach reduced complexity via decoupling and learning, the key idea here is to optimize the value function only for a small set of reachable beliefs $\mathcal{P}$ and not over the entire belief simplex. Developing a class of point-based algorithms, which mostly differ in the way the subset of belief points is chosen and the execution order of the backup operations over the selected belief points, has been the focus of recent algorithm-development research targeting large scale POMDPs. These algorithms were designed to deal with large state spaces, yet, two extra difficulties in the scheduling problem arise from the size of the action space (which is 2^n for all models) and the observation space (for the models in Sections 2.2). Regarding the dimensionality of the action

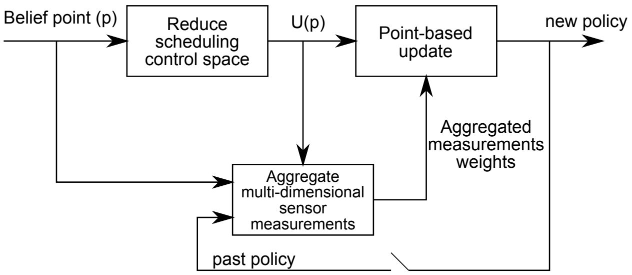

Belief point (p)
Reduce scheduling control space
U(p)
Point-based update
new policy
Aggregated measurements weights
Aggregate multi-dimensional sensor measurements
past policy

Figure 4.11: Structure of the point-based scheduling approximation.

space, we devise a strategy to sample actions based on the support of the beliefs and the sparse structure of the transition models. Intuitively speaking, an object can only move from one side of the network to the other side within time constraints rendering exponentially many scheduling actions irrational at certain times. Hence, instead of performing full updates including 2^n actions, we perform the minimization over a reduced control space $\mathcal{U}(\boldsymbol{p})$ for every $\boldsymbol{p} \in \mathcal{P}$. When dealing with continuous or large observations, we combine that with a methodology that aggregates observations and uses aggregate observations for value iteration updates. At the core of the algorithm we use Perseus [157], a variant of PBVI [122], whereby value iteration updates are not carried out for every sampled belief. Instead, the values for many belief points are improved simultaneously in one update. Figure 4.11 depicts the structure of our point-based approximation, combining control space reduction and observation aggregation with point-based updates. Earlier, we described $\mathrm{Q}_{\mathrm{MDP}}$ based policies, whereby issues (i) and (iii) in Remark 2.1 are resolved since we only needed to solve the underlying Markov Decision Process to describe the full approximate surrogate value function. Decoupling the problem into one-per-sensor subproblems (naturally or artificially) further enabled us to address issue (ii). Yet, the resulting scheduling policies generally do not take control actions to gain information.

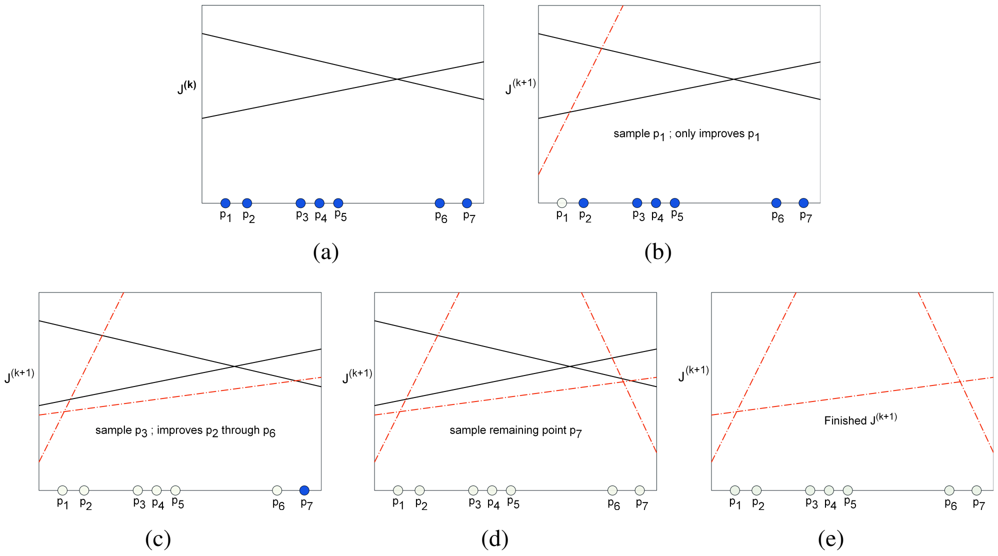
Instead of reducing complexity via artificial decoupling and learning, the key idea of point-based approximate policies is to optimize the value function only for specific reachable sampled beliefs and not over the entire belief simplex (addressing issue (i) in Remark 2.1). Due to the large size of the control space, we also devise strategies to sample actions exploiting the sparsity of the beliefs and the problem structure (to address issue (ii)). Moreover, observation aggregation is used for continuous observation models. Furthermore, since Perseus updates are not carried out for every sampled belief and multiple belief points are improved simultaneously, the number of α vectors grows modestly with the number of iterations addressing issue (iii) in Remark 2.1.

Figure 4.12 illustrates the progress of one iteration of Perseus. The x-axis represents the belief space with circles representing the sampled belief set $\mathcal{P} = \{p_1, \ldots, p_7\}$. The y-axis is the value function at consecutive iterations, i.e. $J^{(k)}$ (solid lines) and $J^{(k+1)}$ (dashed lines). The figure displays the α vectors and different steps illustrating the progress of the algorithm. The algorithm selects a belief point at random and updates the value function for that belief. Then a new update is carried out for a belief point randomly selected from the set of remaining beliefs, i.e., beliefs which did not improve in the previous step. The algorithm repeats till all belief points are updated. Solid lines represent the hyperplanes in the k-th iteration and dashed lines represent the newly added hyperplanes during the $(k+1)$-th iteration.

2.2.2. Sampling Actions Based on the Support of the Belief

Note that the DP update equation involves a minimization over all control actions in $|\mathcal{U}|$. Even though one iteration of the algorithm is linear in the cardinality $|\mathcal{U}|$ of the control space, $|\mathcal{U}|$ itself is exponential in the number of sensors, thus rendering the minimization infeasible for a relatively large sensor network.

The idea here is to exploit the structure of the scheduling/tracking problem. Since the target transition model is naturally sparse, we predict relatively small uncertainty regions for the target state at future time steps. More specifically, for every belief point in $\mathcal{P}$, we use prior information about the target transition



(a) (b)

(c) (d) (e)

Figure 4.12: One iteration of Perseus illustrating the progress of the algorithm. The x-axis represents the belief space with circles representing the sampled belief set $\mathcal{P} = \{p_1, \ldots, p_7\}$. The y-axis is the value function at consecutive iterations, i.e. $J^{(k)}$ and $J^{(k+1)}$. Solid lines represent the hyperplanes in the k-th iteration and dashed lines represent the newly added hyperplanes during the $(k+1)$-th iteration. (a) The initial value function $J^{(k)}$; (b) p_1 is randomly selected and a new α vector is added to $J^{(k+1)}$. This update step only happens to improve p_1. Dark circles represent belief points which did not yet improve; (c) p_3 is sampled and a new hyperplane is added which improves the value for p_2 through p_6; (d) Only p_7 did not improve, thus p_7 is sampled and a new hyperplane is added to $J^{(k+1)}$; (e) All belief points improved, $J^{(k+1)}$ is computed, the iteration ends.

model to project the future state of the target. This is particularly useful when the current belief vector is sparse leading to more restricted uncertainty regions. Subsequently, we restrict our attention to a *significant* subset of sensors, that is, the sensors of relevance to the particulars of the uncertainty region. Hence, we only consider scheduling actions involving scheduling different combinations of a reduced number of sensors which considerably reduces the control space for every belief in $\mathcal{P}$. If the number of significant sensors is still large, we randomly sample actions from the reduced control space. Note that the same intuition extends to more complex motion models wherein information about target speed, maneuver, and acceleration can be factored in to define the future uncertainty regions. Hence, instead of performing full updates including 2^n actions, we perform the minimization over a reduced control space for every $\boldsymbol{p} \in \mathcal{P}$. Specifically, we redefine the point update equation as: $\boldsymbol{\alpha} = \arg\min_{\{\boldsymbol{\alpha}_{\boldsymbol{u}}^{\boldsymbol{p}}\}_{\boldsymbol{u} \in \mathcal{U}(\boldsymbol{p})}} \boldsymbol{p} \cdot \boldsymbol{\alpha}_{\boldsymbol{u}}^{\boldsymbol{p}}$, where $\mathcal{U}(\boldsymbol{p})$ designates the reduced control space for the belief vector $\boldsymbol{p}$. It is worth mentioning that the observation and the cost models need to be computed on the fly for each sampled control action during the algorithm implementation.

2.2.3. Observation Aggregation

The point update equation involves back-projecting all hyperplanes in the current iteration one step from the future and returning the vector that minimizes the value of the belief. Since this involves computing a cross sum by enumerating all possible combinations of alpha vectors for the different observations, a number of vectors which is exponential in the number of the observations is generated at each stage. The recursion has to be redefined to address continuous observation models. It is not hard to see that if different observations map to the same minimizing hyperplane, then they can be aggregated. Hence, if we can partition the observation space into regions that map to the same hyperplane (possibly non contiguous), the continuous model is reduced to a corresponding discrete model. Integration is replaced by a summation over these partitions and the weighing probabilities are obtained by integrating the conditional density over these partitions. This is clarified in the following:

$$\int_s \min_{\boldsymbol{\alpha}_i} \sum_{b'} p(s|u,b') \sum_b p(b'|b)\boldsymbol{p}(b)\boldsymbol{\alpha}_i(b')\ ds = \sum_j \sum_{b'} [\boldsymbol{p}P]_{b'} \mathbb{P}[\mathcal{S}_j|\boldsymbol{u},b']\boldsymbol{\alpha}_j(b'). \tag{4.55}$$

To find the regions $\mathcal{S}_j$ of aggregate observations, we need to solve for the boundaries, i.e., for each pair (i,j) of α vectors we need to solve for $\boldsymbol{s}$:

$$\boldsymbol{\alpha}_i \cdot \phi(\boldsymbol{p},\boldsymbol{u},\boldsymbol{s}) = \boldsymbol{\alpha}_j \cdot \phi(\boldsymbol{p},\boldsymbol{u},\boldsymbol{s}) \tag{4.56}$$

where $\phi(\boldsymbol{p},\boldsymbol{u},\boldsymbol{s}) = \boldsymbol{p}_1(b') \propto \sum_b \boldsymbol{p}(b)p(\boldsymbol{s}|b',\boldsymbol{u})p(b'|b)$

Hence, we need to solve:

$$\sum_{b'}(\boldsymbol{\alpha}_i(b') - \boldsymbol{\alpha}_j(b'))[\boldsymbol{p}P]_{b'} \exp\left\{-\frac{1}{2}\sum_{i:u_i=1}(s_i - \frac{V}{(b'-p_i)^2+1})^2\right\} = 0 \tag{4.57}$$

After solving for the boundaries, we can readily define the regions:

$$\mathcal{S}_{j^*} = \{\boldsymbol{s}|j^* = \arg\max_j \boldsymbol{\alpha}_j \cdot \phi(\boldsymbol{p},\boldsymbol{u},\boldsymbol{s})\} \tag{4.58}$$

Now the update step is simply:

$$J(\boldsymbol{p}) = g(\boldsymbol{p},\boldsymbol{u}^*) + \sum_j \sum_{b'} [\boldsymbol{p}P]_{b'} \mathbb{P}[\mathcal{S}_j|\boldsymbol{u}^*,b']\boldsymbol{\alpha}_j(b') \tag{4.59}$$

where

$$\mathbb{P}[\mathcal{S}_j|\boldsymbol{u}^*,b'] = \int_{\boldsymbol{s}\in\mathcal{S}_j} p(\boldsymbol{s}|\boldsymbol{u}^*,b')d\boldsymbol{s}.$$

Finding a closed form analytical solution for (4.57) is not feasible. Instead, we use Monte-Carlo simulations to solve for the boundaries and get estimates of the weighing probabilities by sampling observations from $p(s|u,b')$ for different combinations of actions and target states.

Akin to the sleeping problem, we are able to derive lower bounds on the energy-tracking tradeoff for the simple as well as the continuous Gaussian observation models. *For the simple model*, the $\mathrm{Q_{MDP}}$ value function is itself a lower bound on the expected total cost since more information is available to the controller at future time steps given the reduced uncertainty assumption. To obtain a lower bound on the optimal energy-tracking tradeoff for such models, we combine the observable-after-control assumption with a decomposable lower bound on the tracking cost as we presented for the sleeping problem.

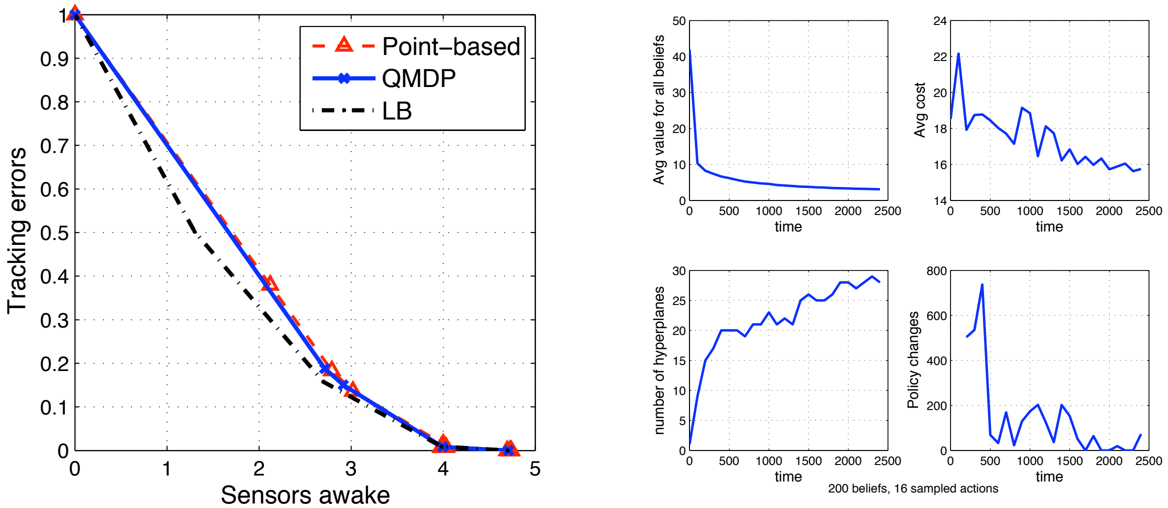
2.2.4. Results

In this section, we show experimental results illustrating the performance of the proposed scheduling policies for the different models considered in this paper.For the planning phase in case of point-based policies, beliefs are sampled by simulating multiple object trajectories through the sensor network. Each trajectory starts from a random state sampled from the initial belief, picking actions at random, until the target leaves the network.

First, we consider the simple model with a linear network of 41 sensors. The object can move anywhere from three steps to the left to three steps to the right in each time step. The distribution for these movements is given in Table 4.4. Figure 4.13(a) shows the tradeoff curve between the number of active sensors per unit time and the tracking error per unit time using the point-based and the $\mathrm{Q_{MDP}}$ policies. The figure also shows a lower bound on the optimal performance. It is clear that both policies lead to tradeoffs that closely approach the lower bound. The $\mathrm{Q_{MDP}}$ policy gets even closer to the lower bound at small tracking errors since the observable-after-control assumption is more meaningful in this regime. In Figure 4.13(b) we show convergence results for the point-based algorithm with reduced control space minimization. The top left

Table 4.4: Object movement for a network of 41 sensors with simple cost and sensing models.

Change in Position	-3	-2	-1	0	1	2	3
Probability	0.23	0.10	0.01	0.33	0.06	0.05	0.22



(a) Energy-Tracking tradeoff for a one-dimensional network of 41 sensors with the simplistic sensing and cost model.

(b) Convergence results for the point-based algorithm for a one-dimensional network of 41 sensors with the simplistic sensing and cost model.

Figure 4.13: Tradeoffs and convergence for a 1-D networks of 41 sensors with the simple sensing and cost model.

subplot displays the convergence of the sum cost of all the belief points in $\mathcal{P}$; the top right shows the expected cost averaged over many trajectories; the bottom left subplot shows the number of hyperplanes constituting the value function as a function of time; the bottom right subplot shows the number of policy changes versus time, i.e., the number of belief points for which the optimal action changed over 2 consecutive iterations of the algorithm. Figure 4.15 displays the average cost and the tradeoff curves for the network in Figure 4.14 with a probabilistic observation model. The cost per unit time is the average ratio of the total energy plus tracking cost and the time the object spends in the network before reaching the termination state. The network is composed of 12 sensors and 20 object locations with the shown connectivity such that the observation range for the different sensors overlap. The object moves according to a random walk anywhere from three steps to the left to three steps to the right in each time step. The distribution of these movements is given in Table 4.1. For the locations close to the boundaries, i.e., when less than three steps are available on the right or left, the remaining probability is absorbed in the transition to the termination state. Since the tracking error for this model is inherently coupled across sensors, the global point-based policy clearly outperforms the learning-based Q_{MDP} policy.

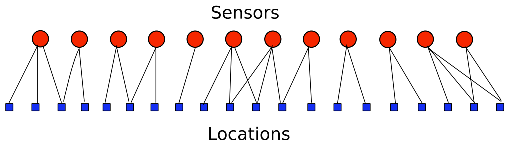


Figure 4.14: A sensor network with overlapping sensing ranges (12 sensors and 20 object locations). An edge connects a sensor to a given location if this location falls within the sensing range of that sensor.

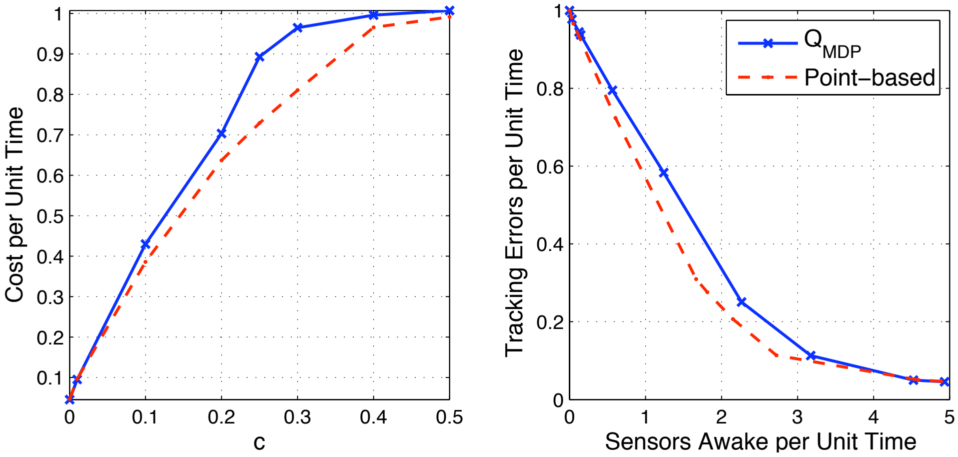


Figure 4.15: Overlap model

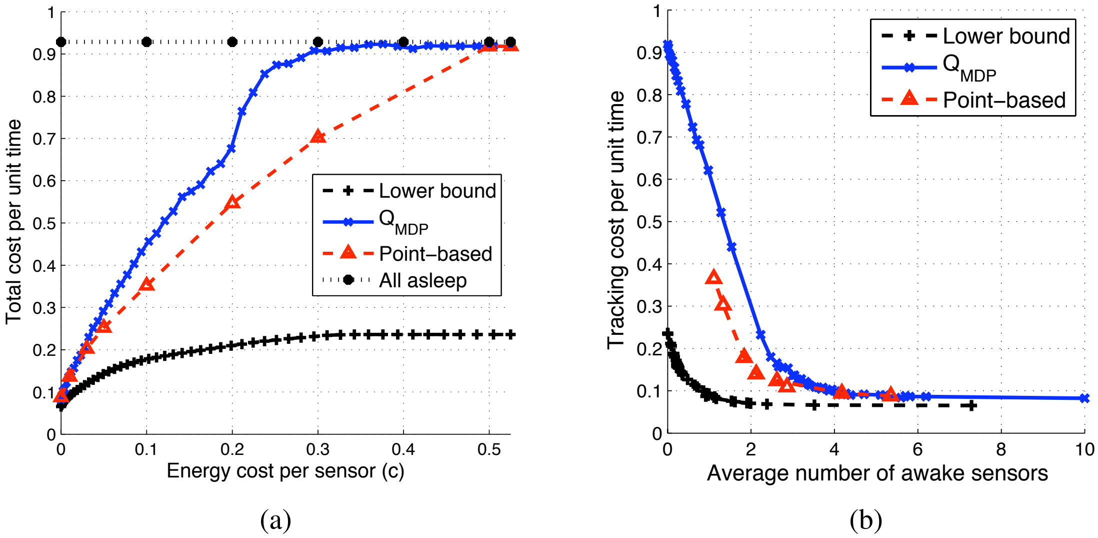


Figure 4.16: Continuous observation model: (a) Total cost versus energy cost per sensor, (b) Energy-tracking tradeoff

Next, we consider a network of 10 sensors where object locations are located on integers from 1 to 21. The observation for each awake sensor is continuous and Gaussian as in (4.54) with $V = 10$. The locations of the sensors are as given earlier in Table 4.2 and the object moves according to the random walk defined in Table 4.1. For every object state and every scheduling action in the reduced control space, we sample 50 observations to construct estimates of the weight probabilities and compute the aggregate observation boundaries. Up to 32 actions are sampled from the reduced control space. In this setup, the belief set consists of 500 sampled belief vectors and we assume a Hamming error cost. Figure 4.16 shows the performance of the different policies for the continuous observation model. It is shown that the point-based scheduling policy outperforms the Q_{MDP} policy. We further show a lower bound on the optimal performance tradeoff. The lower bound is loose especially in the high tracking error regime since the derived bound on per-sensor tracking errors assumes all other sensors are awake. However, we can exactly compute the saturation point for the optimal scheduling policy since every policy has to eventually meet the all-asleep performance curve, shown in Figure 4.16a, when the energy cost per sensor is high. At that point, all sensors are inactive and hence the target estimate can only be based on prior information. Our results are not restricted to 1-D

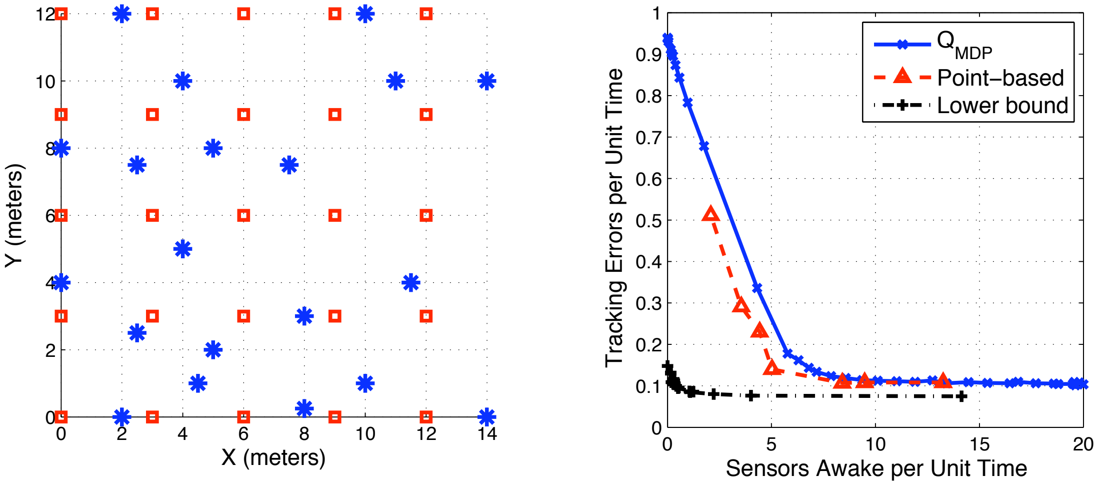


Figure 4.17: (Left) 2-D network with 20 sensors (stars) and 25 possible object locations (squares). (Right) Energy-Tracking tradeoff of the Q_{MDP} and point-based scheduling policies for a 2-D network with continuous observations and Hamming cost.

networks but easily apply to 2-D networks. Namely, Figure 4.17 (right) shows the energy-tracking tradeoff of the Q_{MDP} and point-based policies in addition to a lower bound on optimal performance for the 2-D network of Figure 4.17 (left) with continuous observations and Hamming cost. The entries of the object transition matrix are generated randomly with the restriction that at any state the object can only move to its neighboring locations or remain at its current state. This simulation shows similar trends to the previously observed results. The point-based policies outperform the Q_{MDP} approach at the expense of an increase in the offline computational complexity of the planning phase. Furthermore, the lower bound is reasonably tight in the low tracking error regime.

2.3. Scheduling in Clutter

Here, our goal is to design a central controller to schedule the sensors to track an object of interest in the presence of false alarms (clutter) [7]. Non-linear filtering for tracking in cluttered environments is particularly hard as it requires considering a large number of events due to the so-called data association problem, and is hence computationally intensive. The presence of random interference from nearby objects, false alarms, electromagnetic interference etc. generally leads to ambiguity in the origin of the sensor measurements and hence it is crucial to associate the measurements with their corresponding tracks. One simple and intuitive candidate solution for the association problem is to choose the signal with the highest intensity, among a set of validated measurements, for track update and discard the others. This is known as Strongest Neighbor Filter SNF. The Nearest Neighbor Filter NNF is another solution that uses the measurement closest to the predicted measurement obtained through a prediction step of the track estimation filter. However, these algorithms start to fail when the false alarm rate, or clutter density, increases. Alternatively, probabilistic data association (PDA) for a single target in clutter is another approach which uses all the validated measurements and does not discard any of them [13]. A proper weight, reflecting the association probability, is assigned to each measurement and the weighted average of the validated innovations is used for the update.

While most of the existing literature on target tracking in clutter has focused on the estimation aspect of the tracking problem using one or two sensors, the primary focus of our work is on the design of efficient control policies organizing the activity of a larger network of sensors in the presence of false alarms. We cast the scheduling problem as a Partially Observable Markov Decision Process (POMDP), and devise strategies whereby the sensors are activated to optimize the fundamental tradeoff between energy expenditure and tracking performance in the presence of spurious measurements from clutter.

We focus on the design of scheduling policies rather than the tracking aspect of the problem. Following the same bottom-up approach, we first considered a simplistic model for sensing with non-overlapping sensing ranges. This assumption is then relaxed when we consider sensors with overlapping sensing regions. Thus, if $\boldsymbol{s}_k$ is the measurement vector at time k and $s_{k,\ell}$ the observation of the ℓ-th sensor, then

$$s_{k,b_k} = \begin{cases} 1, & \text{if } u_{k-1,b_k} = 1; \\ 0, & \text{if } u_{k-1,b_k} = 0; \end{cases} \tag{4.60}$$

and

$$s_{k,\ell} = \begin{cases} 1, & \text{w.p. } P_F \text{ if } u_{k-1,\ell} = 1; \\ 0, & \text{if } u_{k-1,\ell} = 0; \\ 0, & \text{w.p. } 1 - P_F \text{ if } u_{k-1,\ell} = 1; \end{cases} \qquad \forall \ell \neq b_k \tag{4.61}$$

The clutter density is captured by the false alarm probability P_F that an active sensor provides a positive measurement. Therefore, clutter leads to uncertainty into the origin of the measurements which could eventually lead to loss in tracking performance. Proper countermeasures should take that into consideration when designing a sensor scheduling policy.

2.3.1. Overlapping Sensing Regions

In this model, we allow the sensing regions to overlap. An example of this model is illustrated in Figure 4.18 depicting a network of $n = 12$ sensors observing $m = 20$ potential object locations according to the shown connectivity. If $\mathcal{B}_{b_k}$ is the set of sensors observing the target at time k, then the observation model of the ℓ-th sensor is given by

$$p(s_{k,\ell} = 1 | b_k, u_{k-1,\ell} = 1) = \begin{cases} 1, & \text{if } \ell \in \mathcal{B}_{b_k}; \\ P_F, & \text{if } \ell \notin \mathcal{B}_{b_k} \end{cases} \tag{4.62}$$

That is, when the target is in the vicinity of an active sensor, the sensor gets a positive observation, however, active sensors which do not belong to the set $\mathcal{B}_{b_k}$ could also falsely declare a target is present with probability P_F. This discrete model is simplistic, yet it captures essential features in real sensing systems, namely, overlapping sensing ranges, limited visibility for each sensor, as well as geographical neighborhood properties. *In the presence of clutter*, the estimation problem becomes more involved. We have to adapt our filter to account for the uncertainty in the origin of sensor measurements. We let $\mathcal{A}(k)$ denote the set of active sensors declaring a target at time k, and $\mathcal{A}_i(k),\ i = 1, \dots, |\mathcal{A}(k)|$, its i-th element. Now define the events

$$\begin{aligned} &\theta_i(k) \triangleq \{s_{k,\mathcal{A}_i(k)} \text{ is target originated}\},\ i = 1, \dots, |\mathcal{A}(k)| \\ &\theta_{|\mathcal{A}(k)|+1}(k) \triangleq \{\text{target reaches termination state } \tau\} \end{aligned} \tag{4.63}$$

and

$$\theta_0(k) \triangleq \bigcap_i \theta_i^c(k) \tag{4.64}$$

with probabilities

$$\beta_i(k) = \mathbb{P}(\theta_i(k) | I_k),\ i = 0, \dots, |\mathcal{A}(k)| + 1 \tag{4.65}$$

where $\theta_i^c(k)$ denotes the complement of the event $\theta_i(k)$. Wherefore, $\theta_0(k)$ refers to the event where none of the measurements at time k is target-originated.

the new belief at time k can be written as Hence,

$$\boldsymbol{p}_k = \beta_0(k) \left[\boldsymbol{p}_{k-1} P\right]_{\{j: u_{k-1,j}=0\}} + \sum_{i=1}^{|\mathcal{A}(k)|} \beta_i(k) e_{\mathcal{A}_i(k)} \qquad + \beta_{|\mathcal{A}(k)|+1}(k) e_\tau \tag{4.66}$$

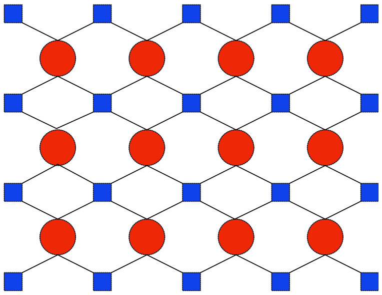


Figure 4.18: Network with overlapping sensor regions. 12 sensors (shown as circles) cover 20 locations (squares). An edge connects a circle and a square when the location falls within the sensing region of that sensor.

Next we compute the association probabilities $\beta_i(k) = P(\theta_i|I_k)$. Using Bayes' rule

$$\beta_i(k) = P_F^{|\mathcal{A}(k)|-1} \prod_{j\in\mathcal{A}(k)} \delta(s_{k,j}-1) \prod_{\ell\notin\mathcal{A}(k)} \delta(s_{k,\ell}) \times \sum_{b_{k-1}} \mathbb{P}(b_k = \mathcal{A}_i(k)|b_{k-1})\boldsymbol{p}_{k-1},\ i = 1,\ldots,|\mathcal{A}(k)| \tag{4.67}$$

and

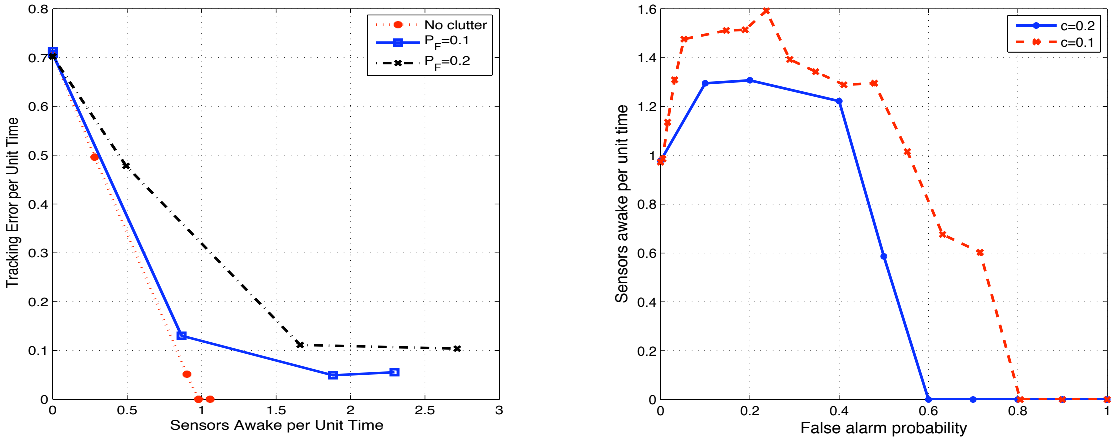
$$\beta_0(k) = P_F^{|\mathcal{A}(k)|} \prod_{j\in\mathcal{A}(k)} \delta(s_{k,j}-1) \prod_{\ell\notin\mathcal{A}(k)} \delta(s_{k,\ell}) \times [p_{k-1}P]_{\{j:u_{k-1,j}=0\}} \tag{4.68}$$

An analogous approach can be used to write the filtering equations for the overlap model but the evolution is generally more difficult to write mathematically in a compact form. Procedurally, it follows the exact same approach described above. All the hypotheses are first enumerated. Then, the evolution is obtained as a weighted combination of the evolution under each individual hypothesis where the weights correspond to the association probabilities. However, the number of hypotheses in this case is significantly larger since under one hypothesis it might very well be the case that multiple measurements from awake sensors are target originated because the sensing regions of different sensors overlap. The number of hypotheses generally scales exponentially with the number of active sensors. To reduce the space of association hypotheses, the controller could limit the *maximum* number of sensors to be activated at every time step to a relatively small number, say n_1 sensors. As we will show in our simulations results, we choose $n_1 = 5$, i.e., at most 5 sensors could be active at a given time instant. we present our proposed point-based scheduler which approximates the optimal solution using a point-based approximation driven by the non-linear filters described previously.

3. Simulation Results

In this section, we show experimental results illustrating the performance of the proposed scheduling policies for the different models.

First, to illustrate some of the basic ideas, consider a simple linear network with 11 sensors where the object moves according to a symmetric random walk either one step to the left or one step to the right. We term this network Net A. Figure 4.19(a) shows the tradeoff curves between the number of active sensors per unit time and the tracking error per unit time using our point-based scheduler for different levels of clutter density for Net A. As expected, for the no clutter case in the low tracking error regime, i.e., at vanishing energy cost per sensor, activating one sensor to the left or to the right of the sensor is enough to perfectly track the target. The reason being that, at each time step the target would be either perfectly observed (by an awake sensor) or its position can be exactly inferred. Hence, the point-based scheduler in this case converges to the optimal scheduling policy. As the clutter density increases, it is clear that the tracking error increases for the same number of active sensors. The figure illustrates the performance of our point-based scheduler for different clutter densities, namely, when 10% and 20% of the time an active sensor measures clutter.



(a)

(b)

Figure 4.19: (a) Energy tracking tradeoff for different clutter densities for Net A; (b) Number of active sensors versus clutter density for different energy costs per sensor for Net A.

Figure 4.19(b) shows the number of active sensors versus the clutter density for fixed energy costs per sensor $c = 0.1$ and $c = 0.2$. Indeed, for the no clutter case, only one sensor is activated. This is optimal as long as $c < 0.5$ since the energy cost is smaller than the expected tracking cost of 0.5. As we increase the clutter density, the scheduler chooses to activate more sensors to compensate for the uncertainty associated with the origin of the measurements until a certain point where the clutter density is just too high that the tracking performance approaches that of uninformed tracking based solely on the available knowledge about the propagation model and the clutter model. In this case, activating sensors is of no avail and the scheduler judiciously chooses to disengage all the sensors to save the energy resources and avoid unnecessary resource expenditure. Not surprisingly, the x-axis intercept i.e., the cutoff clutter density at which all sensors are deactivated, is larger for smaller values of c since turning on more sensors is less costly.

Second, consider the scenario where the object is moving according to a random walk anywhere from three steps to the left to three steps to the right in each time step as in Table 4.1. Figure 4.20 illustrates the energy-tracking tradeoff of the proposed policies for different levels of clutter density. It is clear that the degradation in performance w.r.t. to the no clutter case is graceful at low and moderate clutter densities.

Next we consider a network where the sensing regions of different sensors overlap. The network consists of 20 possible object locations monitored by 12 sensors as shown in Figure 4.18. The total cost per unit time versus the energy cost c is shown in Figure 4.21 (a) for different levels of clutter density. All the curves saturate when the energy cost is too high and the scheduler disengages all the sensors. In this case, the total cost is due to tracking cost based solely on the prior information. Not surprisingly, the saturation point occurs at smaller c for higher values of the clutter density. For moderate values of the clutter density (e.g. 5%), the gap between the saturation points of the cluttered case and the no clutter case is small showing that through judicious use of scheduling actions we are able to compensate for the uncertainty due to clutter. Figure 4.21 (b) shows the tradeoff curves for this overlap network.

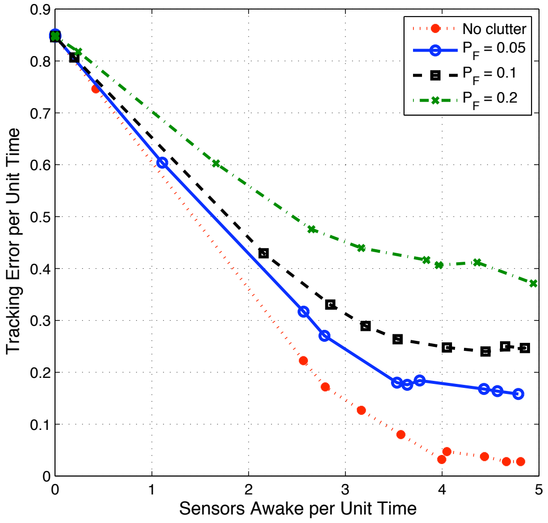
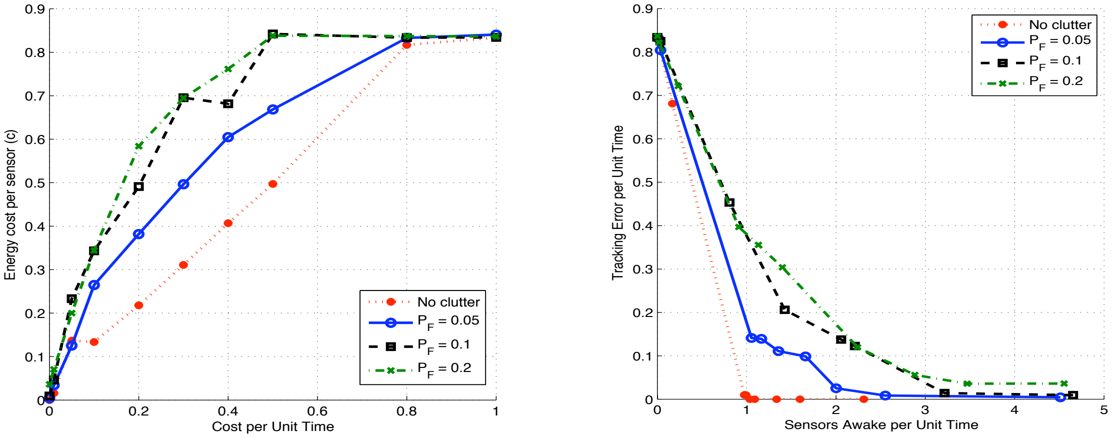


Figure 4.20: Energy tracking tradeoff for different clutter densities for Net B.



(a)

(b)

Figure 4.21: (a) Total cost per unit time versus energy cost per sensor for the overlap network; (b) Energy tracking tradeoff for different clutter densities for network with overlapping sensing ranges

Chapter 5

Agile Sensors and Boundary Tracking

The work presented in this chapter has been done by the group of Dr. Bertozzi (UCLA) in collaboration with Dr. Tartakovsky (USC).

1. Agile Sensors

1.1. Second Generation Testbed

During summer 2006 we constructed the second generation of an economical cooperative control testbed. The original car-based vehicle was improved with on-board range sensing, limited on board computing, and wireless communication, while maintaining economic feasibility and scale. A second, tank based platform, uses a flexible caterpillar-belt drive and the same modular sensing and communication components. We demonstrated practical use of the testbed for algorithm validation by implementing a recently proposed cooperative steering law involving obstacle avoidance. This work was published in the proceedings of the 2007 American Control Conference.

1.2. Boundary Tracking

CoPI Bertozzi and postdoc Zhipu Jin developed a framework for environmental boundary tracking and estimation by considering the boundary as a hidden Markov model (HMM) with separated observations collected from multiple sensing vehicles. For each vehicle, a tracking algorithm is developed based on Pages cumulative sum algorithm (CUSUM), a method for change point detection, so that individual vehicles can autonomously track the boundary in a density field with measurement noise. Based on the data collected from sensing vehicles and prior knowledge of the dynamic model of boundary evolvement, we estimate the boundary by solving an optimization problem, in which prediction and current observation are considered in the cost function. Examples and simulation results were presented to verify the efficiency of this approach. This work was published in the 2007 IEEE Conference on Decision and Control. The algorithm was implemented on the second generation testbed using a convoy of vehicles. Relative positioning between vehicles allows several to maintain a convoy while tracking the boundary. The algorithm performs well in the presence of moderate sensor noise. Some adaptation of the algorithm was necessary to run it on a testbed with limited onboard computing. In particular a modified Kalman filter was implemented in which a constant gain was estimated a priori and used on the vehicle. The implementation work was published in the 2009 IEEE American Control Conference.

1.3. Paper Published in ICINCO 2010

Bobby Liu, Martin Short, Yasser Taima, and Andrea Bertozzi develop a searching algorithm for a group of agents moving in a swarm and sensing potential targets. The objective of the algorithm is to use these groups to efficiently search for and locate targets with a finite sensing radius in some bounded area. We present an algorithm that both controls agent movement and analyzes sensor signals to determine where targets are located. We use computer simulations to determine the effectiveness of this collaborative searching. A scaling analysis compares swarm size to proficiency of searching. This work has been published in Proceedings of the 7th International Conference on Informatics in Control, Automation, and Robotics (ICINCO), Portugal, June 2010. This work is being transitioned to China Lake Naval Air Warfare Center (see Tech Transfer section).

1.4. Paper Published in ICINCO 2011

One time step up funds provided on this contract led to the design and building of new hardware for our microcar testbed at UCLA. The cars were designed and built by Artero's Lab, a small startup company formed by former masters students working in Bertozzi's lab. The initial setup was tested and software controllers were written by EE Masters student David Hermina as part of his final project for the degree. In summer 2010 a team of undergraduates developed algorithms to test the capabilities of the platform. Their work led to a publication in ICINCO 2011 [61].

The micro-cars feature two on board processors; each devoted to different tasks. The upper board is a 350MHz-rated Virtex-4 FPGA. It is dedicated to on-board algorithm processing as well as a user interface. The main motivation for a powerful on-board processor is to increase the autonomy of each vehicle. In previous iterations of the AML Testbed, vehicles relied on a third-party desktop computer to perform all calculations. With this new processor, the cars can perform all required processing on-board. The lower board is a 50MHz-rated ARM Cortex-3 microcontroller. It is dedicated to mo- tion control and sensor management. The lower board directly controls the vehicles motion by generating power modulation signals that feed into the car's motor and steering servo. It also gathers data from the various sensors using a prioritized task controller. The micro-cars have control system made up of two parts. The cars have a rear wheel drive system with a maximum speed of 20 cm/s. The cars are steered by a axel-articulated servo connected to the front wheels. The experimentally verified maximum turning angle is ±18 degrees. This proved to be one of the largest limitations of the testbed as the micro-cars can trace a circle with a minimum diameter of 50cm. Because the increased computing capabilities could allow for more complex programs to be run on board, the memory was also upgraded from the previous generation. The upper board has access to a both a 64MB DDR SDRAM module as well as a 4 MB flash drive. The lower board's microcontroller houses internal 8KB SRAM and a 64KB flash drive. The lower board also contains a 1KB EEPROM module to store system control parameters that are specific to each car, such as the car's identification number, servo gains, and the servo offset.

The sensor systems of the micro-cars is vastly increased from the previous generations. Each car sports a 640x480 digital camera, two high performance uni-directional gyroscopes, an optical encoder used for velocity estimation, and an infrared sensor module. The digital camera is not currently functional but could later be integrated to add advanced image processing capabilities to algorithms. The camera could be applied to detect obstacles when coupled with the IR sensors or to locate destinations points on the testbed. The 140Hz analog gyroscopes provide physical orientation sensing information. They are functional but are not currently used in any experiments due to the completely flat nature of the testbed. Each car is equipped with either a long-range or a short-range forward-facing infrared sensing module. The long range sensors can detect objects in the 10cm-140cm range and the short range sensors can only detect from 10cm-80cm. The IR sensors have been previously characterized in master's student David Hermina's thesis [102]. The IR sensors have been used in several algorithm tests to facilitate an emergency stop protocol and basic barrier avoidance.

The cars have been tested with respect to a steering control algorithm originally proposed by Justh and Krishnaprasad and modified by Morgan and Schwartz for obstacle avoidance [111] this algorithm was featured on the second generation testbed [85]. In our new work we test peer to peer networking in which communication is no longer all-all but rather uses low bandwidth information sharing through subnetworks. One example shown in 5.1 uses a daisy-chain network to perform the steering control algorithm originally written for all-all coupling.

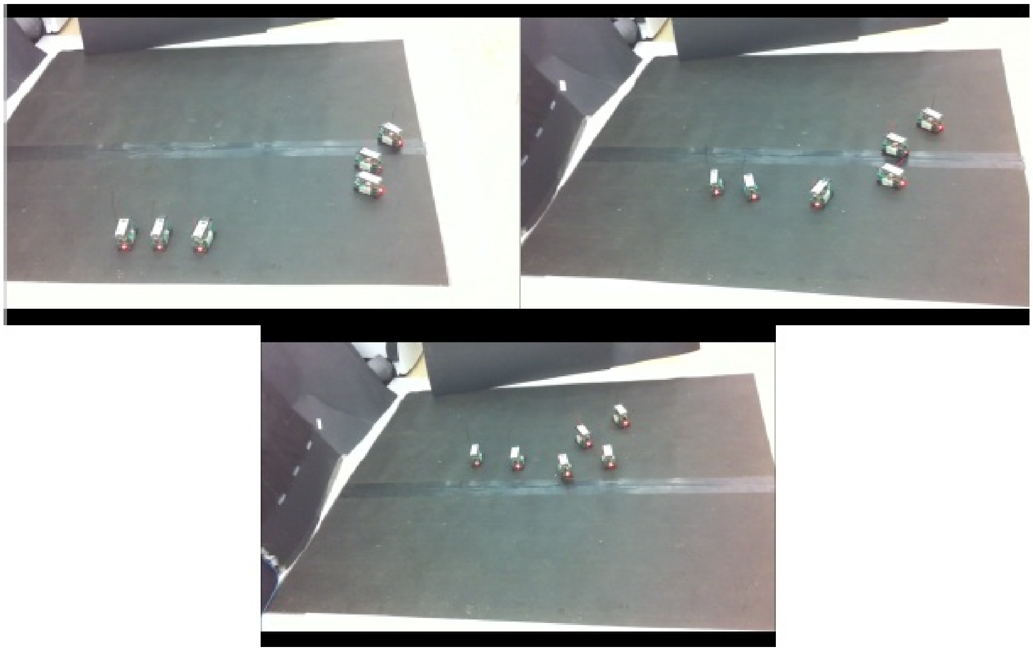


Figure 5.1: Three frames of an experimental run using daisy chain coupling. The cars are originally separated in two groups (top left). During the run, the cars regroup (top right) and a common orientation before exiting the testbed (bottom). Figure from [71].

2. Image Segmentation Through Efficient Boundary Sampling

Segmentation is one of the most important problems in image processing. Partitioning an image into a small number of homogeneous regions highlights important features, allowing a user to analyze the image more easily. Applications include medical imaging, computer vision, and geospatial target detection. Image segmentation methods can be subdivided into region-based vs. edge-based methods. Region-based methods include the Mumford-Shah and related Chan–Vese methods which both involve energy minimization with a least squares fit of the data and a partition, between regions, whose length is minimized. Edge-based methods include the well-known image snakes and Canny edge detector. Other approaches to segmentation have also been effective. Statistical methods such as region competition rely on the fact that images have repetitive features that can be learned and exploited to obtain a segmentation. A more recent fast statistical method called Distance Cut is semi-supervised (the user identifies segments in each region) and is based on weighted distances and kernel density estimation.

All of these methods involve, at some level, sampling all the pixels in an image. For applications involving high dimensional or large data sets, it makes sense to subsample the image. This is especially important for high resolution data where it can be prohibitive to perform calculations on every pixel in the image. Bertozzi, Tartakovsky, Chen, and Wittman develop a segmentation method is designed for this kind of application and is based on prior work for cooperative environmental sampling with robotic vehicles. The algorithm has two levels, namely a global searching method, which locates a boundary point, and a local sampling algorithm, which tracks the boundary using the global method as an initial point. Occasionally, if the tracker strays too far from the boundary, additional uses of the global algorithm are needed. The local algorithm is based on CUSUM statistics (see Figure 5.2). This work has been published in AMRX [30].

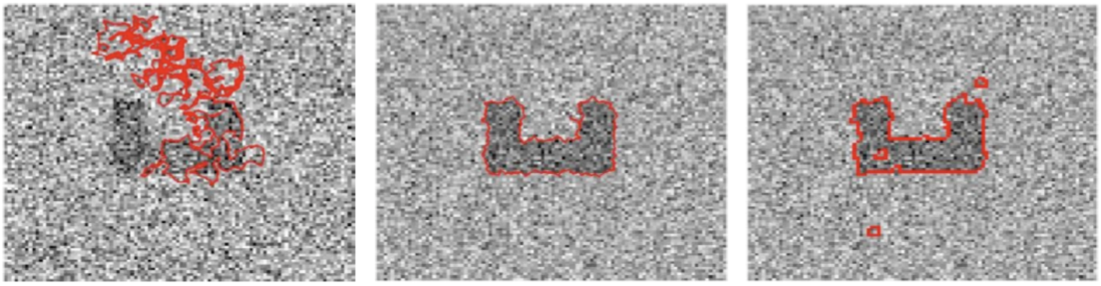


Figure 5.2: A 100 × 100 image was corrupted with additive Gaussian noise, N(0,0.5). Left: Boundary tracking without a change-point detection modification. Middle: Boundary tracking with the CUSUM algorithm. Right: Threshold dynamics - global segmentation method.

Chapter 6

Detection and Tracking of Covert Hostile Activities

The research presented in this chapter has been performed by the group of Dr. Aram Galstyan at ISI in collaboration with Dr. Paul Cohen. The major goal of this part of the project is to develop a scalable probabilistic framework for performing detecting and tracking covert activities of hostile agents. This framework will include an algorithmic toolkit for detecting and tracking hostile activities, methodology for analyzing properties of those algorithms, and theoretical models that will address the general question, such as accuracy and *trackability* and *detectability*. During this project, we have made significant progress in both theoretical and computational aspects of the above problems. Below we summarize these achievements.

1. Modeling Activities via Hidden Markov Models

We have suggested Event–Coupled Factorial HMMs (EC–FHMM) as a generic framework for modeling activities in Hats and other domains. EC–FHMM are different from more conventional HMMs with factorial state representation in the way observations are generated. Specifically, what is observed in EC–FHHMs is the *interactions* between different chains, as shown in Figure 1. In the context of the plan recognition problem, each chain describes an agent, while observations imply that two chains have interacted at a give time step. Thus, in contrast with previously studied factorial HMMs and dynamic Bayesian networks (DBN), where the topology of coupling between the chains is predetermined in advance, in our model the topology is dynamic, which is well suited for capturing dynamically evolving networks. Furthermore, the interaction between the chains is informed by the internal states of the nodes, while the states of those nodes themselves are influenced by those interactions. This provides a feedback mechanism between individual and collective dynamics, which translates into a very reach behavior of the model.

It is easy to see that after a sufficiently long time, approximate inference is infeasible with even moderate N. Thus, one needs develop approximate methods for inference and learning. To this end, we have employed the so called Boyen–Koller factorization, which works by approximating the true belief states by a product of belief states over smaller clusters of variables. This approximation has been shown to produce good results for a number of problems. Importantly, it has been established that the accumulative error caused by this approximation is bounded by the mixing rates of the underlaying Markov dynamics.

Clearly, the accuracy of the approximation depends on the choice of the cluster variables. The perfect accuracy is recovered when all the coupled variables belong to a single cluster. And the most aggressive approximation corresponds to the situation where each agent is represented as a separate cluster. We have previously shown that for simple Hats scenarios even the most aggressive approximation (i.e., each agent is treated as a separate "cluster") produces reasonable results provided that the prior knowledge about the agents is sufficiently accurate.

In this phase, we have extended our work by considering iterative approach for the above collective inference problem. The main premise behind the iterative scheme is the following: Assume we have used

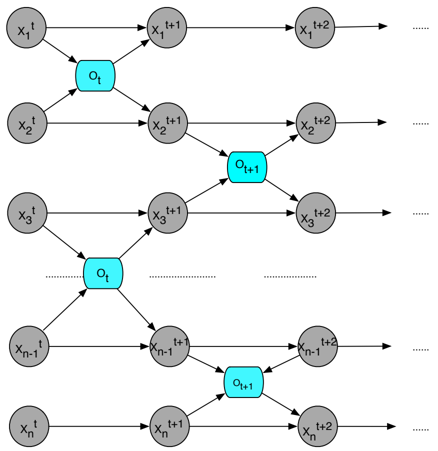


Figure 6.1: Time–rolled diagram of an Event–Coupled Factorial HMM.

the most aggressive factorization (one agent per cluster), and found the hidden state sequences for individual agents. This can be done by the slightly modified Viterbi algorithm. Thus, if we have n agents, then after one iteration we have estimated the hidden state sequences for those agents. Now, some of those estimates will be more *accurate*, or more *certain*, than the others. Here the former means that the given hidden state sequence has a higher likelihood under the assumed model parameters, while the latter means that the entropy of the corresponding posterior distribution is low. Then, we can *freeze* the agent whose hidden state sequence is the most accurate/certain, and repeat the iteration for the remaining $n-1$ agents (in practice, we can freeze not one but the top k agents). Indeed, our preliminary results suggests that this procedure increases the accuracy over original, one–shot approach to the inference.

2. Co-Evolving Stochastic Blockmodel for Dynamic Networks

In EC-FHMM-s, each node is in a certain state, and interaction between different agents depend on those states. Many situations, however, are better described by multi–faceted interactions, where nodes can bear multiple latent roles that influence their relationships to others. MMSB accounts for such "mixed" interactions, by allowing each node to have a probability distribution over roles, and by making the interactions role–dependent [4].

We have developed a model named Co-evolving Mixed Membership Stochastic Blockmodel, or CMMSB, which provides a dynamic generalization of the mixed membership model by explicitly modeling the variation in the node membership vectors. Previously, a dynamic extension of the MMSB (dMMSB) was suggested in [57]. In contrast to dMMSB, where the dynamics was imposed *externally*, our model assumes that the membership evolution is driven by the interactions between the nodes through a parametrized *influence* mechanism. At the same time, the patterns of those interactions themselves change due to the evolution of the node memberships.

Another advantage of our model over dMMSB is that the latter models the aggregate dynamics, e.g., the *mean* of the logistic normal distribution from which the membership vectors are sampled. CMMSB, however, models each node's trajectory separately, thus providing better flexibility for describing system dynamics. Of course, more flexibility comes at a higher computational cost, as CMMSB tracks the trajectories of all nodes individually. This additional cost, however, can be well justified in scenarios when the system as a whole is almost static (e.g., no shift in the mean membership vector), but different subsystems experience dynamic changes. One such scenario that deals with political polarization in the U.S. Senate is presented in our experimental results section.

Consider a set of N nodes, each of which can have K different roles, and let $\vec{\pi}_p^t$ be the mixed membership vector of node p at time t. Let Y_t be the network formed by those nodes at time t: $Y_t(p,q) = 1$ if the nodes

p and q are connected at time t, and $Y_t(p,q) = 0$ otherwise. Further, let $Y_{0:T} = \{Y_0, Y_1, \ldots, Y_T\}$ be a time sequence of such networks. The generative process that induces this sequence is described below.

- For each node p at time $t = 0$, employ a logistic normal distribution over a simplex sample.[1]
  $\pi^0_{p,k} = \exp(\mu^0_{p,k} - C(\vec{\mu}^0_p)), \quad \vec{\mu}^0_p \sim \mathcal{N}(\vec{\alpha}^0, A)$
  where $C(\vec{\mu}) = \log(\sum_k \exp(\mu_k))$ is a normalization constant, and $\vec{\alpha}^0, A$ are prior mean, and covariance matrix.
- For each node p at time $t > 0$, the mean of each normal distribution is updated due to *influence* from the neighbors at its previous step:
  $\vec{\alpha}^t_p = (1 - \beta_p)\vec{\mu}^{t-1}_p + \beta_p \vec{\mu}_{\mathcal{S}(p,t-1)}$
  where $\vec{\mu}_{\mathcal{S}(p,t-1)}$ is average of weighted membership vector $\vec{\mu}$-s of the nodes which node p has met at time $t-1$
  $\vec{\mu}_{\mathcal{S}(p,t-1)} = \sum_q Y(p,q) w^{t-1}_{p \leftarrow q} \vec{\mu}_q$
  β_p describes how easily the node p is influenced by its neighbors. The membership vector at time t is
  $\pi^t_{p,k} = \exp(\mu^t_{p,k} - C(\vec{\mu}^t_p)), \quad \vec{\mu}^t_p \sim \mathcal{N}(\vec{\alpha}^t_p, \Sigma_\mu)$
  where the covariance Σ_μ accounts for noise in the evolution process.
- For each pair of nodes p, q at time t, sample role indicator vectors from multinomial distributions:
  $\vec{z}^t_{p \rightarrow q} \sim Mult(\vec{z}|\vec{\pi}^t_p)$, $\vec{z}^t_{p \leftarrow q} \sim Mult(\vec{z}|\vec{\pi}^t_q)$
  Here $\vec{z}_{p \rightarrow q}$ is a unit indicator vector of dimension K, so that $z_{p \rightarrow q,k} = 1$ means node p undertakes role k while interacting with q.
- Sample a link between p and q as a Bernoulli trial:
  $Y_t(p,q) \sim Bernoulli(y|(1-\rho)\vec{z}^t_{p \rightarrow q} B^t \vec{z}^t_{p \leftarrow q})$
  where B is a $K \times K$ role–compatibility matrix, so that B^t_{rs} describes the likelihood of interaction between two nodes in roles r and s at time t. When B^t is diagonal, the only possible interactions are among the nodes in the same role. Also, ρ is a parameter that accounts for the sparsity of the network.

Thus, the coupling between dynamics of different nodes is introduced by allowing the role vector of a node to be influenced by the role vectors of its neighbors.To benefit from computational simplicity, we updated $\vec{\pi}$ by changing its associated $\vec{\mu}$. This update of $\vec{\mu}$ is a linear combination of $\vec{\mu}$ at its current state, and the values of its neighbors. The influence is measured by a node–specific parameter β_p, and $w^t_{p \leftarrow q}$. β_p describes how easily the node p is influenced by its neighbors: $\beta_p = 0$ means it is not influenced at all, whereas $\beta_p = 1$ means the behavior is solely determined by the neighbors. Conversely, $w^t_{p \leftarrow q}$ reflects the influence that node q exerts on node p, so that larger values correspond to more influence.

2.1. Inference and Learning

Under the Co–Evolving MMSB, the joint probability of the data $Y_{0:T}$ and the latent variables $\{\vec{\mu}^t_{1:N}, \vec{z}^t_{p \rightarrow q} : p,q \in N, \vec{z}^t_{p \leftarrow q} : p,q \in N\}$ can be written in the following factored form. To simplify the notation, we define $\vec{z}^t_{p,q}$ as a pair of $\vec{z}^t_{p \leftarrow q}$, and $\vec{z}^t_{p \rightarrow q}$

$$\begin{aligned} p(Y_{0:T}, \vec{\mu}^{0:T}_{1:N}, \vec{Z}^{0:T}_{\rightarrow}, \vec{Z}^{0:T}_{\leftarrow} | \vec{\alpha}, A, B, \beta_p, w^t_{p \leftarrow q}, \Sigma_\mu) &= \prod_t \prod_{p,q} P(Y_t(p,q)|\vec{z}^t_{p,q}, B^t) P(\vec{z}^t_{p,q}|\vec{\mu}^t_p, \vec{\mu}^t_q) \\ &\times P(\vec{\mu}^{t+1}_p | \vec{\mu}^t_p, \vec{\mu}_{\mathcal{S}(p,t)}, Y_t, \beta_p) \prod_p P(\vec{\mu}^0_p | \vec{\alpha}, A) \end{aligned} \tag{6.1}$$

In Equation 6.1, the term describing the dynamics of the membership vector is defined as follows[2]:

$$P(\vec{\mu}^t_p | \vec{\mu}^{t-1}_p, \vec{\mu}^{t-1}_{\mathcal{S}(p,t)}, \Sigma_\mu, Y_t, \beta_p) = f_G(\vec{\mu}^t_p - f_b(\vec{\mu}^{t-1}_p, \vec{\mu}^{t-1}_{\mathcal{S}(p,t)}), \Sigma_\mu) \tag{6.2}$$

$$f_G(\vec{x}, \Sigma_\mu) = \frac{1}{(2\pi)^{k/2} |\Sigma_\mu|^{1/2}} e^{-\frac{1}{2} x^T {\Sigma_\mu}^{-1} x} \tag{6.3}$$

$$f_b(\vec{\mu}^{t-1}_p, \vec{\mu}^{t-1}_{\mathcal{S}(p,t)}) = (1 - \beta^{t-1}_p) \vec{\mu}^{t-1}_p + \beta^{t-1}_p \vec{\mu}^{t-1}_{\mathcal{S}(p,t)} \tag{6.4}$$

[1] We found that the logistic normal form of the membership vector suggested in [57] led to more tractable equations compared to the Dirichlet distribution.

[2] For simplicity, we will assume Σ_μ is a diagonal matrix.

Algorithm 1 Variational EM

```
**Input:** data $Y_t(p,q)$, size $N, T, K$
Initialize all $\{\vec{\gamma}\}^t$, $\{\sigma\}^t$
Start with an initial guess for the model parameters.
repeat
    repeat
        for $t = 0$ to $T$ do
            repeat
                Initialize $\phi^t_{p\to q}$, $\phi^t_{p\leftarrow q}$ to $\frac{1}{K}$ for all $g, h$
                repeat
                    Update all $\{\phi\}^t$
                until convergence of $\{\phi\}^t$
                Find $\{\sigma\}^t$, $\{\vec{\gamma}\}^t$
                Update all $\{\zeta\}^t$
            until convergence in time $t$
        end for
    until convergence across all time steps
    Update hyper parameters.
until convergence in hyper parameters
```

Performing exact inference and learning under this model is not feasible. Thus, one needs to resort to approximate techniques. Here we use a variational EM [19, 189] approach. The main idea behind variational methods is to posit a simpler distribution $q(X)$ over the latent variables with free (variational) parameters, and then fit those parameters so that the distribution is close to the true posterior in KL divergence.

$$D_{KL}(q||p) = \int_X q(X) \log \frac{q(X)}{p(X,Y)} dX \tag{6.5}$$

Here we introduce the following factorized variational distribution:

$$q(\vec{\mu}^{0:T}_{1:N}, Z^{0:T}_{\rightarrow}, Z^{0:T}_{\leftarrow} | \vec{\gamma}^{0:T}_{1:N}, \Phi, \Phi) = \prod_{p,t} q_1(\vec{\mu}^t_p | \vec{\gamma}^t_p, \Sigma^t_p) \times \prod_{p,q,t} (q_2(\vec{z}^t_{p\to q} | \vec{\phi}^t_{p\to q}) q_2(\vec{z}^t_{p\leftarrow q} | \vec{\phi}^t_{p\leftarrow q})) \tag{6.6}$$

where q_1 is the normal distribution, and q_2 is the multinomial distribution, and $\vec{\gamma}^t_p, \Sigma^t_p, \vec{\phi}^t_{p\to q}, \vec{\phi}^t_{p\leftarrow q}$ are the variational parameters. Intuitively, $\phi^t_{p\to q,g}$ is the probability of node p undertaking the role g in an interaction with node q at time t, and $\phi^t_{p\leftarrow q,h}$ is defined similarly. Note that in the E–step, we need to compute the expected value of $\log[\sum_k \exp(\mu_k)]$ under the variational distribution, which is problematic. Toward this end, we introduce N additional variational parameters ζ, and replace the expectation of the log by its upper bound induced from the first-order Taylor expansion:

$$\log[\sum \exp(\mu_k)] \leq \log \zeta - 1 + \frac{1}{\zeta} \sum \exp(\mu_k) \tag{6.7}$$

The variational EM algorithm works by iterating between the E–step of calculating the expectation value using the variational distribution, and the M–step of updating the model (hyper)parameters so that the data likelihood is locally maximized. The pseudo-code is shown in Algorithm 1, and the details of the calculations are discussed below.

2.2. Variational E-step

In the variational E–step, we minimize the KL distance over the variational parameters. Taking the derivative of KL divergence with respect to each variational parameter and setting it to zero, we obtain a set of equations that can be solved via iterative or other numerical techniques. For instance, the variational parameters $(\vec{\phi}^t_{p\to q}, \vec{\phi}^t_{p\leftarrow q})$, corresponding to a pair of nodes (p,q) at time t, can be found via the following iterative scheme:

$$\phi^t_{p\to q,g} \propto \exp(\gamma^t_{p,g}) \times \prod_h (B(g,h)^{Y_t(p,q)}(1-B(g,h))^{1-Y_t(p,q)})^{\phi^t_{p\leftarrow q,h}} \tag{6.8}$$

$$\phi^t_{p\leftarrow q,h} \propto \exp(\gamma^t_{q,h}) \times \prod_g (B(g,h)^{Y_t(p,q)}(1-B(g,h))^{1-Y_t(p,q)})^{\phi^t_{p\to q,h}} \tag{6.9}$$

In the above equations, $\phi^t_{p\to q,g}$ and $\phi^t_{p\leftarrow q,h}$ are normalized after each update. Note also that Eqs. 6.8 and 6.9 are coupled with each other as well as with the parameters $\gamma^t_{p,g}$, $\gamma^t_{q,h}$.

For the variational parameters Σ^t_p, we have for the diagonal components $(\sigma^t_{p,1}, \sigma^t_{p,2}, ...\sigma^t_{p,k})$:

$$\frac{\eta^2_k}{\sigma^t_{p,k}} = 1 + (1-\beta_p)^2 + \sum_q Y_t(p,q)\beta^2_q {w^t_{q\leftarrow p}}^2 + 2\eta^2_k(N-1)\frac{\sigma^t_{p,k}}{\zeta^t_p}\exp(\gamma^t_{p,k} + \frac{(\sigma^t_{p,k})^2}{2}), \tag{6.10}$$

where η_k is the diagonal component of the covariance matrix Σ_μ. Similarly, we obtain equations for the variational parameters γ-s. Generally, those equations are different for $\gamma^0_{p,g}$, $\gamma^T_{p,g}$, and $\gamma^t_{p,g}$, $0 < t < T$. Since those equations are too cumbersome, here we simply note that their general form is:

$$\vec{\gamma}^t_p = f(\vec{\gamma}^{t-1}_p, \vec{\gamma}^{t+1}_p, \vec{\gamma}^t_q, \vec{\phi}^t_{p\to q}, \vec{\phi}^t_{q\leftarrow p}, \zeta^t_p, \Sigma^t_p), \tag{6.11}$$

Thus, the parameter $\vec{\gamma}^t_p$ depends on its past and future values, $\vec{\gamma}^{t-1}_p$ and $\vec{\gamma}^{t+1}_p$, as well as the parameters of its neighbors. Finally, for the variational parameters ζ we have

$$\zeta^t_p = \sum_i \exp(\gamma^t_{p,i} + \frac{{\sigma^t_{p,i}}^2}{2}) \tag{6.12}$$

Note that the above equations can be solved via simple iterative update as before. To expedite convergence, however, we combine the iterations with Newton–Raphson method, where we solve for individual parameters while keeping the others fixed, and then repeat this process until all the parameters have converged.

2.3. Variational M step

The M-step in the EM algorithm computes the parameters by maximizing the expected log-likelihood found in the E-step. The model parameters in our case are: B^t, the role-compatibility matrix, the covariance matrix Σ_μ, β_p for each node, $w^t_{p\leftarrow q}$ for each pair, $\vec{\alpha}$, and A from the prior.

If we assume that the time variation of the block compatibility matrix is small compared to the evolution of the node attributes, we can neglect the time dependence in B, and use its average across time, which yields:

$$\hat{B}(g,h) = \frac{\sum_{p,q,t} Y_t(p,q)\cdot\phi^t_{p\to q,g}\phi^t_{p\leftarrow q,h}}{\sum_{p,q,t}\phi^t_{p\to q,g}\phi^t_{p\leftarrow q,h}} \tag{6.13}$$

Likewise, for the update of diagonal components of the noise covariance matrix Σ_μ,

$$\hat{\eta}_k = \frac{1}{N(T-1)}E_q[\sum_{p,t}(\mu^t_{p,k} - (1-\beta)\mu^{t-1}_{p,k} - \beta\mu_{\mathcal{S}(p,t-1),k})^2] \tag{6.14}$$

Similar equations are obtained for β_p and $w^t_{p\leftarrow q}$. The update equation of β_p and $w^t_{p\leftarrow q}$ is a function of γ and σ which are related to the transition for specific node p. Since these equations are rather involved, they will be provided elsewhere.

The priors of the model can be expressed in closed form as below:

$$\vec{\alpha}^0 = \frac{1}{N} \sum_p \vec{\gamma}_p^0 \tag{6.15}$$

$$a_k = \sqrt{\frac{1}{N} \sum (\gamma_{p,k}^2 + \sigma_{p,k}^2 - 2\alpha_k^0 \gamma_{p,k} + {\alpha_k^0}^2)} \tag{6.16}$$

2.4. Results for US Senate Co-Sponsorship Network

We have also performed some preliminary experiments for testing our model against real–world data. In particular, we used senate co–sponsorship networks from the 97th to the 104th senate, by considering each senate as a separate time point in the dynamics. There were 43 senators who remained part of the senate during this period. For any pair of senators (p, q) in a given senate, we generated a directed link $p \to q$ if p co-sponsored at least 3 bills that q originally sponsored. The threshold of 3 bills was chosen to avoid having too dense of a network. With this data, we wanted to test (a) to what extent senators tend to follow others who share their political views (i.e., conservative vs. liberal) and (b) whether some senators change their political creed more easily than others.

The number of roles $K = 2$ was chosen to reflect the mostly bi–polar nature of the US Senate. The susceptibility of senator p to influence is measured by the corresponding parameter β_p, which is learned using the EM algorithm. High β means that a senator tends to change his/her role more easily. Likewise, the power of influence of senator q on senator p is measured by the parameter $w^t_{p \leftarrow q}$, where $w^t_{p \leftarrow q_1} > w^t_{p \leftarrow q_2}$ means senator q_1 is more influential on senator p than senator q_2. Here the direction of the arrow reflects the direction of the influence which is opposite to the direction of link. To initialize the EM procedure, we assigned the same β, and w to all the senators, and start with a matrix which is weighted at the diagonal for B.

Another method for validation is to compare the degree of influence. Our model handles, and learns, the degree of influence in the update equation. Sorting out influential senators is an area of active research. Recently, KNOWLEGIS has been ranking US senators based on various criteria, including influence, since 2005. Since our data was extracted from the 97th senate to the 104th senate, direct comparison of the rankings was impossible. Another study [101] ranked the 10 most influential senators in both parties who have been elected since 1955. We compared our top 5 influential senators, and we were able to find 3 senators (Sen. Byrd, Sen. Thurmond, and Sen. Dole) in the list.

2.4.1. Interpreting Results

The role-compatibility matrix learned from the Variational EM has high values on the diagonal confirming our intuition that interaction is indeed more likely between senators that share the same role. Furthermore, the learned values of β showed that senators varied in their "susceptibility". In particular, Sen. Arlen Spector was found to be the *most influenceable* one, while Sen. Dole was found to be one of the most *inert* ones. Note that while there are no direct ways of estimating the "dynamism" of senators, our results seem to agree with our intuition about both senators (e.g., Sen. Spector switched parties in 2009 while Dole became his party's candidate for President in 1996).

To get some independent verification, we compared our results to the yearly ratings that ACU (American Conservative Union), and ADA(Americans for Democratic Action) assign to senators [3]. ACU/ADA rated every senator based on selected votes which they believed to have a clear ideological distinction, so that high scores in ACU mean that they are truly conservative, while lower score in ACU suggests they are liberal, and for ADA vice versa. To compare the rating with our predictions (given by the membership vector) we scaled the former to get scores in the range $[0, 1]$.

[3] Accessible at http://www.conservative.org/,
http://www.adaction.org/

Figure 6.2 shows the relationship between these scores and our mixed membership vector score, confirming our interpretation of the two roles in our model as corresponding to liberal/conservative. Although those values cannot be used for quantitative agreement, we found that at least qualitatively, the inferred trajectories agree reasonably well with the ACU/ADA ratings. This agreement is rather remarkable since the ACU/ADA scores are based on selected votes rather than co–sponsorship network as in our data.

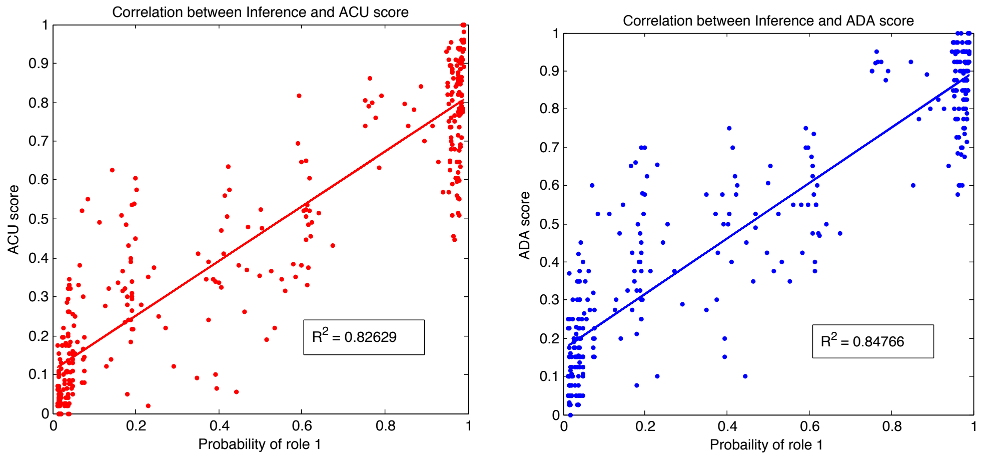


Figure 6.2: Correlation between ACU/ADA scores and inferred probabilities.

Of course, we are most interested in correctly identifying the dynamics for each senator. We compare our inferred trajectory of the most dynamic senator, and the inert senator to the scores of ACU, and ADA. In Figure 6.3 the scores of ADA have been flipped, so that we can compare all of the scores in the same measurement. However, since ACU/ADA scores are rated for every senator each year, the dynamics of inference, and the dynamics of ACU/ADA scores cannot be compared one to one. Not all senators showed high correlation of the trend like senator Specter, and Dole.

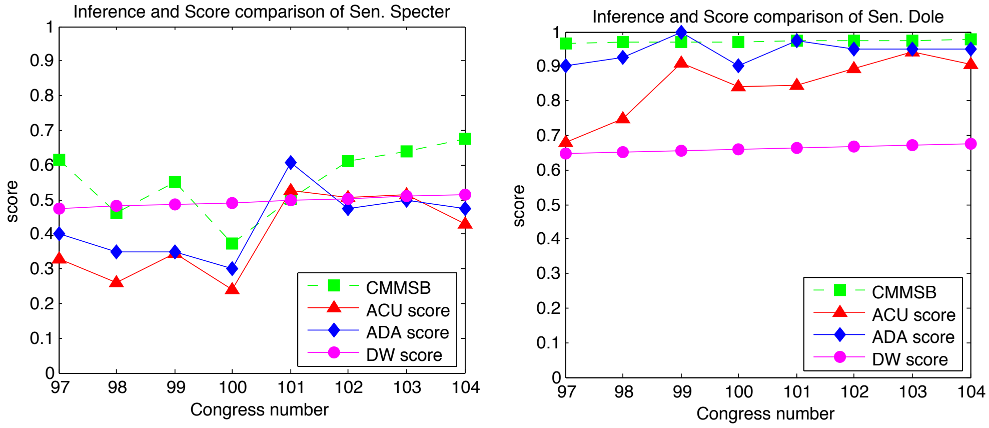


Figure 6.3: Comparison of inference results with ACU and ADA scores: Sen. Specter (top) and Sen. Dole (bottom).

2.4.2. Polarization Dynamics

The yearly ACU/ADA scores give a good comparison of the relative political position of senators scored in each year. However, they are not very appropriate for comparison between years, a point illustrated by the fact that the score is based on voting records for different bills in each year. Therefore, for validation of

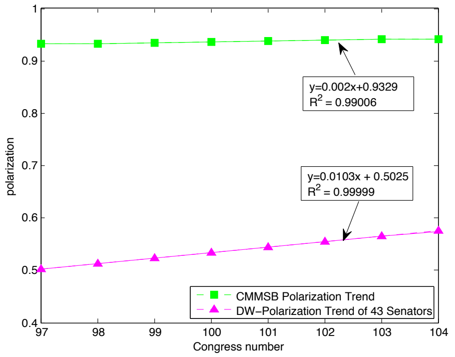


Figure 6.4: Polarization trends during 97th–104th US Congresses.

the dynamics we turn to another scoring system highly regarded by political scientists and used to observe historical trends, the DW-NOMINATE score. For the time period of our study, [103] shows that the political polarization of the senate was increasing. In particular, they show that the gap between the average DW-NOMINATE score of Republicans and Democrats is monotonically increasing, as we show in Figure 6.4. In fact, the polarization for the entire senate was stronger every year. This is due to the unbalanced seats in the entire senate. In other words, our data had 22 Republican, and 21 Democratic, while for the entire senate, majority out numbered minority by around 10 seats. For comparison, for each time step we took the average of our inferred score for the 14 most and least conservative senators. As we show in Figure 6.4, our inferred result agrees qualitatively with the results of [103], showing an increase in polarization for every senate in the studied time-window. Since the DW-NOMINATE scores uses its own metric, and our polarization is measured by the difference between upper average and lower average probability, we should not expect to get quantitative agreement. We would like to highlight, however, that the direction of the trend is correctly predicted for each of the eight terms.

3. Theoretical Analysis of Hidden Markov Models

Hidden Markov Models (HMM) provide one of the simplest examples of structured data observed through a noisy channel. The inference problems of HMM naturally divide into two classes [55, 132]: *i)* recovering the hidden sequence of states given the observed sequence, and *ii)* estimating the model parameters (transition probabilities of the hidden Markov chain and/or conditional probabilities of observations) from the observed sequence. The first class of problems is usually solved via the maximum a posteriori (MAP) method and its computational implementation known as Viterbi algorithm [55, 132]. For the parameter estimation problem, the prevailing method is maximum likelihood (ML) estimation, which finds the parameters by maximizing the likelihood of the observed data.An alternative approach to parameter learning is Viterbi Training (VT), also known in the literature as segmental K-means, Baum–Viterbi algorithm, classification EM, hard EM, etc. Instead of maximizing the likelihood of the observed data, VT seeks to maximize the probability of the most likely hidden state sequence.

During this project we have developed methods based on statistical physics of disordered systems that allowed us to analyze asymptotic properties of inference methods in HMMs. Below we outline the main elements of our approach and summarize our main findings.

3.1. Theoretical Analysis of Trackability

Despite its extensive use of Viterbi algorithm in many applications, its properties, and specifically, the structure of its solution space, have received surprisingly little attention. On the other hand, it is clear that choosing a single state sequence might be insufficient for adequately understanding the structure of the inferred process. To get a more complete picture, one needs to know whether there are other nearly optimal sequences, how many of them, how they compare with the optimal solution, and so on. In our work, we have shown that this can be related to the notion of *trackability*, which can be intuitively defined as one's ability to (accurately) track certain stochastic processes [37, 144]. This problem was addressed by Crespi et. al. [37] for so called *weak models*, where the entries in the HMP transition and emission matrices are either 0 or 1. They reported a sharp transition between the trackable and non–trackable regimes. Here trackable means that the number of hypotheses that can *explain* given observation sequence grows at most *polynomially* with the length of the sequence, whereas non–trackable means that the number of such hypotheses grow exponentially.

For more general stochastic processes, an information-theoretical characterization of trackability was suggested in [144]. Within this approach, the accuracy is characterized by the probability $\Pr[\hat{\mathbf{x}} \neq \mathbf{x}]$ of the estimated sequence $\hat{\mathbf{x}}$ not being equal to the actual one, while the structure of the solution space is described via the number of elements $|\Omega|$ in the (conditional) typical set Ω of $\mathbf{x}$ sequences given an observed sequence $\mathbf{y}$ (complexity).

Note that whereas the accuracy and the complexity measures of [144] deteriorate even for a small (but generic) noise intensity, so that a process is trackable only in the complete absence of noise. During this phase of work, we have suggested an alternative measure for trackability, that is more intuitive in the sense that it allows a finite amount of noise. Namely, we suggested to augment the notion of trackability with the particular inference method being used. Generally, the structure of an inference method can be characterized by the accuracy of the estimation, and the number $\mathcal{N}(\mathbf{y})$ of solutions $\hat{\mathbf{x}}(\mathbf{y})$ that the method can produce in response to a given sequence $\mathbf{y}$. For instance, an HMM process can be said to be trackable under Viterbi (or more generally, Maximum a Posteriori, or MAP) inference if it yields at most polynomial (in the observation length) number of solution with reasonable accuracy.

In our MURI work we have studied the structure of MAP inference for the simplest binary, symmetric HMM, by reducing it to the Ising model in random fields. In this way, the average cost $-\sum_{\mathbf{y}} \Pr(\mathbf{y}) \Pr(\hat{\mathbf{x}}(\mathbf{y})|\mathbf{y})$ of MAP and the logarithm of the number of solutions $\sum_{\mathbf{y}} \Pr(\mathbf{y}) \ln \mathcal{N}(\mathbf{y})$ relate, respectively, to the energy and the entropy of the Ising model at the zero temperature. Consider a binary, discrete-time Markov stochastic process $\mathcal{X} = (X_1, X_2, \ldots, X_N)$. Each random variable X_k has only two realizations $x_k = \pm 1$. The Markov feature implies

$$\mathbf{p}(\mathbf{x}) = \prod\nolimits_{k=2}^{N} p(x_k|x_{k-1}) p(x_1), \tag{6.17}$$

where $p(x_x|x_{k-1})$ is a time-independent transition probability of the Markov process. For the considered binary symmetric situation it is parameterized by a single number $0 < q < 1$, $p(1|1) = p(-1|-1) = 1 - q$, $p(1|-1) = p(-1|1) = q$, and the stationary distribution is $p_{\rm st}(1) = p_{\rm st}(-1) = \frac{1}{2}$. Furthermore, the noise process is assumed to be memory-less, time-independent and unbiased:

$$\mathbf{p}(\mathbf{y}|\mathbf{x}) = \prod\nolimits_{k=1}^{N} \pi(y_k|x_k), \qquad y_k = \pm 1 \tag{6.18}$$

where $\pi(-1|1) = \pi(1|-1) = \epsilon$, $\pi(1|1) = \pi(-1|-1) = 1 - \epsilon$, and ϵ is the probability of error. Here memory-less refers to the factorization in (6.18), time-independence refers to the fact that in (6.18) $\pi(...|...)$ does not depend on k, while unbiased means that the noise acts symmetrically on both realizations of the Markov process: $\pi(1|-1) = \pi(-1|1)$.

It can be shown that by appropriate parameterization, the MAP estimation is identical to minimizing the following Ising Hamiltonian:

$$H(\mathbf{y}, \mathbf{x}) = -J \sum\nolimits_{k=1}^{N} x_k x_{k+1} - h \sum\nolimits_{k=1}^{N} y_k x_k, \tag{6.19}$$

where $2J = \ln[(1-q)/q]$ and $2h = \ln[(1-\epsilon)/\epsilon]$. The (positive) factor J in (6.19) is the spin–spin interaction constant, and h is determined by the observations.

The expected number of ground–state configurations (e.g., sequences $x_1, x_2, ...x_N$) that minimizes the above Hamiltonian can be related to the zero–temperature thermodynamic entropy Θ,

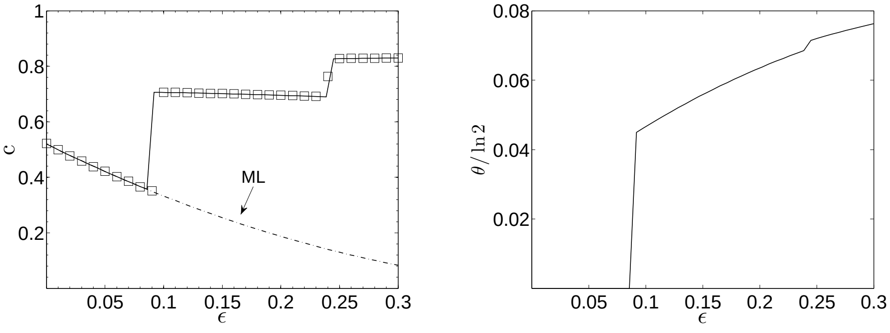
$$\Theta = \sum\nolimits_{\mathbf{y}} \mathbf{p}(\mathbf{y}) \ln \mathcal{N}(\mathbf{y}). \tag{6.20}$$

Thus, a finite $\frac{\Theta}{N}$ means that there are exponentially many outcomes of minimizing $H(\mathbf{y}, \mathbf{x})$ over $\mathbf{x}$. Furthermore, let us introduce free energy (T is the temperature, and $\beta = 1/T$)

$$F(J, h, T) = -T \sum\nolimits_{\mathbf{y}} \mathbf{p}(\mathbf{y}) \ln \sum\nolimits_{\mathbf{x}} e^{-\beta H(\mathbf{y}, \mathbf{x}; J, h)}, \tag{6.21}$$

Then the entropy is given as $\Theta = -\partial_T F|_{T \to 0}$. We can also define the overlap between observed and inferred sequences, which is given by $v = \frac{1}{N} \partial_h F$.

The quantities defined above can be calculated exactly. Our main results can be stated as follows: For small noise intensities, defined by $\epsilon < q/2$, the inferred state sequence coincides with the observation sequence, and hence there is no difference between MAP and Maximum Likelihood (ML) estimations, as the solution is observation–dominated. While it was expected that the two methods agree for a vanishing noise, *the fact of their exact agreement for a finite range of the noise is non-trivial.* In this regime, the entropy is identically zero, which defines a trackable process. Furthermore, upon increasing the noise intensity the MAP solution switches between different operational regimes that are separated by first–order phase transitions. In particular, a first-order phase-transition separates trackable and non–trackable regimes. At this transition point the influence of the prior information becomes comparable to the influence of observations. This is shown in Figure 6.5(a) and 6.5(b)



(a) (b)

Figure 6.5: MAP characteristics versus the noise intensity in the regimes $m = 1, 2, 3$ for $q = 0.24$: (a) Overlap (b) Entropy $\frac{\theta}{\ln 2}$. In (a) the open squares represent simulation results, obtained by running the Viterbi algorithm and calculating the respective quantities directly. We used sequences of size 10^4, and averaged the results over 100 random trials.

There are several directions for further developments. For instance, it will be interesting to generalize the analysis presented here beyond the binary hidden Markov processes considered here. In this case, the MAP optimization problem can be mapped to a Potts model. We would like to note that the behavior observed in the simple binary model can be explained by the emergence of a finite fraction of "frustrated" spins, where the frustration can be attributed to two competing tendencies – accommodating observations on one hand, and the hidden (Markovian) dynamical model on the other. Since this mechanism is rather general, we believe that most features of the MAP scheme uncovered here via an exact analysis of the simplest binary model will survive in more general situations.

3.2. Comparative Analysis of Viterbi Training and Maximum Likelihood Estimation for HMMs

As we have mentioned above, there are two main method for parameter estimation for HMMs.The maximum likelihood (ML) estimation finds the parameters by maximizing the likelihood of the observed data, whereas Viterbi Training (VT) seeks to maximize the probability of the most likely hidden state sequence. Maximizing VT objective function is hard, so in practice it is implemented via an EM-style iterations between calculating the MAP sequence and adjusting the model parameters based on the sequence statistics. It is known that VT lacks some of the desired features of ML estimation such as consistency, and in fact, can produce biased estimates [55]. However, it has been shown to perform well in practice, which explains its widespread use in applications such as speech recognition [17], unsupervised dependency parsing [158], and so on. It is generally assumed that VT is more robust and faster but usually less accurate, although for certain tasks it outperforms conventional EM [158].

The current understanding of when and under what circumstances one method should be preferred over the other is not well–established. For HMMs with continuos observations, Ref. [105] established an upper bound on the difference between the ML and VT objective functions, and showed that both approaches produce asymptotically similar estimates when the dimensionality of the observation space is very large. Note, however, that this asymptotic limit is not very interesting as it makes the structure imposed by the Markovian process irrelevant. A similar attempt to compare both approaches on discrete models (for stochastic context free grammars) was presented in [142]. However, the established bound was very loose.

In this project, one of our goals was to understand, both qualitatively and quantitatively, the difference between the two estimation methods. We develop an analytical approach based on generating functions for examining the asymptotic properties of both approaches. Previously, a similar approach was used for calculating entropy rate of a hidden Markov process [5]. We have provided a non-trivial extension of the methods that allows to perform comparative asymptotic analysis of ML and VT estimation. It was shown that both estimation methods correspond to certain free-energy minimization problem at different *temperatures*. Furthermore, we demonstrated the approach on a particular class of HMM with one unambiguous symbol and obtain a closed–form solution to the estimation problem. This class of HMMs is sufficiently rich so as to include models where not all parameters can be determined from the observations, i.e., the model is not *identifiable* [55].

Our man results are as follows: In contrast to the ML approach that produces continuously degenerate solutions, VT results in finitely degenerate solution that is sparse, i.e., some [non-identifiable] parameters are set to zero, and, furthermore, converges faster. Note that sparsity might be a desired feature in many practical applications. For instance, imposing sparsity on conventional EM-type learning has been shown to produce better results part of speech tagging applications [180]. Whereas [180] had to impose sparsity via an additional penalty term in the objective function, in our case sparsity is a natural outcome of maximizing the likelihood of the best sequence. While our results were obtained on a class of exactly-solvable model, it is plausible that they hold more generally.

The fact that VT provides simpler and more definite solutions—among all choices of the parameters compatible with the observed data—can be viewed as a type of the Occam's razor for the parameter learning. Note finally that statistical mechanics intuition behind these results is that the *a posteriori* likelihood is (negative) zero-temperature free energy of a certain physical system. Minimizing this free energy makes physical sense: this is the premise of the second law of thermodynamics that ensures relaxation towards a more equilibrium state. In that zero-temperature equilibrium state certain types of motion are frozen, which means nullifying the corresponding transition probabilities. In that way the second law relates to the Occam's razor.

4. Semi–Supervised Clustering in Graphs

In recent years there has been a great deal of interest in modeling and understanding relational network–structured data. While traditional learning methods assume data instances are independent and identically distributed, relational learning takes into account non–independencies — in the form of links and relations between different entities — and so extends learning to richly structured data. To represent relational data, a number of different models have been proposed. One of the well–studied models for networked data that has its roots in social network analysis is the so called stochastic block–model [69], where the nodes in the network are assigned to a number of groups (blocks). The main assumption is that the nodes in the same group have similar attributes, and are structurally equivalent in the sense that they have the same pattern of links in the networks. More recent work has suggested "softer" mixed membership models where nodes can be associated with several groups simultaneously [4].

Often, class membership information is not available, and one needs to use statistical inference to recover the latent structures. An important question then is to what extent this can be accomplished. Consider, for instance, a simple block–model with two equal–sized groups , which can be characterized by two numbers p and r – probability of a link within a group and between the groups, respectively. Thus, each node is connected, in average, with pN and rN nodes within the same cluster and across the clusters, respectively, where N is the number of nodes in each group. An important question is how well one can recover the latent group structure through statistical inference in the limit of large N. The answer to this question is straightforward for dense networks, where the network connectivity scales linearly with N. Indeed, it has been shown that the clusters in this *planted partition model* can be recovered with high accuracy in polynomial time if $p - r \geqslant n^{-1/2+\epsilon}$ [34]. Thus, we can say that the detection threshold converges to $p = r$ in the limit $N \to \infty$.

The situation can be significantly different for sparse graphs, where the average connectivity remains finite as $N \to \infty$, $p = \alpha/N$, $r = \gamma/N$, where α and γ are average connectivity with and between the clusters. Indeed, recently it was shown [134] that for sparse block–structured networks with a given within–cluster connectivity α there is a critical between–class connectivity $\gamma_c < \alpha$ so that for any $\gamma > \gamma_c$ clusters cannot be recovered by better than random accuracy in the asymptotic limit. More specifically, it was demonstrated that the model is characterized by a phase transition from *detectable* to *undetectable* regimes as one increases the overlap between the clusters. Clearly, this type of behavior is undesirable as it signals inference instabilities – large fluctuations in accuracy in response to small shifts in the parameters.

In this project, we have shown that this instability can be suppressed if one knows the correct group labels for a finite fraction of nodes. This can be viewed as a semi–supervised version of the problem, as opposed to an unsupervised version where the only available information is the observed graph structure. Generally, graph-based clustering methods can utilize two types of background knowledge – correct cluster group assignment for a subset of nodes, or pair–wise constraints in the form of *must-link* (*cannot-links*), which imply that pair of nodes must be (cannot be) assigned to the same group. Below we describe our studies that examine the impact of such pair-wise constraints on inference.

4.1. Model

Let $\mathbf{A}$ be the observed adjacency matrix of interaction graph of N nodes so that $A_{ij} = 1$ if we have observed a link between nodes i and j, and $A_{ij} = 0$ otherwise. Within the stochastic block–model, the nodes in the network are assigned to a number of groups (blocks), and and the probability of a link between two nodes depends on their group membership. In the simple symmetric bi–cluster scenario considered here, the model is characterized by two numbers – probability of a link within and across the groups, defined as p and r, respectively. The conditional distribution of observation for a given configuration $\mathbf{x}$ reads

$$\mathbf{p}(\mathbf{A}|\mathbf{x}) = p^{c+}[1-p]^{c-}r^{d+}[1-r]^{d-} \tag{6.22}$$

Here c_+, d_+ (c_-, d_-) are the total number of observed (missing) links within and across the groups,

$$c_+ = \sum\nolimits_{i,j} A_{ij}\delta_{x_i,x_j} \ , \ c_- = \sum\nolimits_{i,j} (1 - A_{ij})\delta_{x_i,x_j} \tag{6.23}$$

$$d_+ = \sum\nolimits_{i,j} A_{ij}(1 - \delta_{x_i,x_j}) \ , \ d_- = \sum\nolimits_{i,j} (1 - A_{ij})(1 - \delta_{x_i,x_j}), \tag{6.24}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. Let us define $J_{NL} = \ln[(1-p)/(1-r)]$, $J_L = \ln[p/r] + J_{NL}$. Then the log of the joint distribution over both observed and hidden variables can be written as follows:

$$H(\mathbf{x}, \mathbf{A}) = -\ln[\mathbf{p}(\mathbf{A}|\mathbf{x})\mathbf{p}(\mathbf{x})] = -\sum\nolimits_{i>j} J_L A_{ij}\delta_{x_i,x_j} + \sum\nolimits_{i>j} J_{NL}\delta_{x_i,x_j} + H_\pi(\mathbf{x}) \tag{6.25}$$

Here J_L and J_{NL} stand for the contributions from observed links and non–links, respectively, while the last term $H_\pi(\mathbf{x})$ encodes prior information about the latent structure one might have. In the scenario considered below, we assume that the cluster sizes are known a priori. This constraint can be forced by the appropriate choice of $H_\pi(\mathbf{x})$, so that any clustering arrangement that violates the size constraint will be disallowed. Then it is easy to check that the second term amounts to a constant that can be ignored. Furthermore, since below we are interested in the minimum of $H(\mathbf{x}, \mathbf{A})$, we can set $J_L = 1$ without loss of generality.

For a given parameter values, and a given observed graph, minimizing the above expression is equivalent to maximum a posteriori estimation of the latent structure. The following remark is due: When p and r are chosen irrespective of N (so that average number of links per nodes scales linearly with N), then we expect the group structure to be recovered with high accuracy if N is sufficiently large [34]. However, many real–world networks are *sparse*. To account for this, we introduce average number of neighbors within and across the groups, α and γ, and let $p = \alpha/N$, $r = \gamma/N$, so that the average connectivity remains finite as $N \to \infty$. Below we study the accuracy of the inference depending on α and γ.

To proceed further, we make the bi–component nature of the network explicit by introducing separate variables $x_i = \pm 1$ and $\bar{x}_i = \pm 1$ $(i = 1, \ldots, N)$ for two groups. Then Eq. 6.25 is reduced to the following Ising Hamiltonian (aside from an unessential scaling factor):

$$H = -\sum_{i<j}^{N} J_{ij} x_i x_j - \sum_{i<j}^{N} \bar{J}_{ij} \bar{x}_i \bar{x}_j - \sum_{i,j}^{N} K_{ij} x_i \bar{x}_j + H_\pi(\mathbf{x}) \,. \tag{6.26}$$

Here J_{ij} and $\bar{J}_{ij}$ are the elements on two diagonal blocks of the matrix $\mathbf{A}$ describing the connectivity within each cluster, whereas K_{ij}-s are the elements on the (upper) off–diagonal block of $\mathbf{A}$ that link nodes across the clusters. In the unsupervised block–model, they are random Bernoulli trials with parameters p and r.

To account for background information in the form of pairwise constraints, we use the following form for the prior part of the Hamiltonian:

$$H_\pi(\mathbf{x}) = -w_{ml} \sum_{i<j}^{N} [\theta_{ij} x_i x_j - +\bar{\theta}_{ij} \bar{x}_i \bar{x}_j] + w_{cl} \sum_{i,j}^{N} \phi_{ij} x_i \bar{x}_j \,. \tag{6.27}$$

where $\theta_{ij} = 1$ $(\bar{\theta}_{ij} = 1)$ if the corresponding pair of nodes are connected via a must–link constraint within the first (second) cluster, and $\theta_{ij} = 0$ $(\bar{\theta}_{ij} = 0)$ otherwise. Similarly, $\phi_{ij} = 1$ if there is a cannot link between corresponding nodes in respective clusters, and $\phi_{ij} = 0$ if there is no such link. Here w_{ml} and w_{cl} are the costs of violating a must–link and cannot–link constraints, respectively. For the sake of simplicity, below we will choose $w_{ml} = w_{cl} = M > 1$, where M is an integer.

Below we will assume that the constraints are introduced randomly and independently for each pair of nodes. Namely, $\theta_{ij}, \bar{\theta}_{ij}$-s and ϕ_{ij} are Bernoulli trials with parameter f_+ and f_-, respectively. Then the prior part of the Hamiltonian can be absorbed into 6.26 by the following choice for the distribution of the couplings in 6.26:

$$p(J_{ij}) = [1 - p - f_+]\delta(J_{ij}) + p\delta(J_{ij} - 1) + f_+\delta(J_{ij} - w_{ml}) \tag{6.28}$$

$$p(K_{ij}) = [1 - r - f_-]\delta(K_{ij}) + r\delta(K_{ij} - 1) + f_-\delta(K_{ij} + w_{cl}) \tag{6.29}$$

We are interested in the properties of the above Hamiltonian 6.26 in the limit of large N. Below we study it within the Bethe–Peierls approximation. Let $P(h)$ $(\bar{P}(h))$ denote the probability of an internal (*cavity*) field acting on an x $(\bar{x})$ spin. Then we have according to the zero temperature cavity method [106]:

$$\begin{aligned} P(h) \;&=\; \int \prod_{n=1}^{N} dJ_{0n} p(J_{0n}) \int \prod_{n=1}^{N} dK_{0n} p(K_{0n}) \int \prod_{k=1}^{N} P(h_k) \underset{\cdot}{\mathrm{h}}_k \prod_{l=1}^{N} \bar{P}(g_k) \underset{\cdot}{\mathrm{g}}_k \\ &\times\; \delta\left[h - \sum_{k=1}^{N} \phi[h_k, J_{0k}] - \sum_{k=1}^{N} \phi[g_k, K_{0k}]\right], \end{aligned} \tag{6.30}$$

where $\phi[a, b] \equiv \mathrm{sign}(a) \min[\,|a|, b\,]$, and where g_k (resp. h_k) are the fields acting on the x-spin from $\bar{x}$-spin (respectively from other x-spins).

Once $P(h)$ is found we obtain the first two moments of s_i as

$$m = \int P(h)\,\mathrm{sign}(h), \qquad q = \int P(h)\,\mathrm{sign}^2(h), \ldots, \tag{6.31}$$

Here m is the magnetization averaged over the graph structure (including the constraints) (i.e., averaging over J_{ij}, $\bar{J}_{ij}$ and K_{ij}), and Gibbs distribution, which, at zero temperature case considered here it means averaging over all configurations of x_i and $\bar{x}_i$ that in the thermodynamic limit have— the same (minimal) values of the Hamiltonian H.

In (6.31) m is the magnetization, while q is called EA (Edwards-Anderson) order parameter; q differs from 1 due to possible contribution $\propto \delta(h)$ in $P(h)$. Note that the accuracy of the clustering (i.e., probability that a node has been assigned to the correct cluster) is simply $\frac{1+|m|}{2}$. Thus, $|m| = 1$ corresponds to perfect clustering, whereas $m = 0$ means that discovered clusters have only random overlap with the true cluster assignments.

Equation 6.30 cannot be solved analytically for arbitrary M. Below we study two specific case: $M = 2$, where some analytical insights can be obtained, and $M = \infty$, where we will employ population dynamics to study properties of $P(h)$.

First, we consider the case of soft constraints $M = 2$. The results are shown in Figure 6.6(a) where we plot the magnetization m as a function of α for $\gamma = 1$. For $\rho = 0$, which corresponds to unsupervised scenario, there is a critical value of α below which the magnetization m is zero. Recall that the clustering accuracy (i.e., fraction of correct cluster assignments) is given as $\frac{1+|m|}{2}$. Thus, for any $\alpha < \alpha_c$ the estimation cannot do any better than random guessing. At a certain value of ρ, the detection threshold becomes $\alpha = \gamma$. If ρ is increases even further, the model has a non–zero magnetization even when $\alpha < \gamma$. Note that this shift suggests highly non–linear effect from the added constraints depending on the network parameters. Indeed, when the connectivity is close to its critical value, the constraints can significantly improve the clustering accuracy by moving the system away from the critical regime. And when the system is away from the critical region to start with, then the addition of the constraints might yield no improvement at all compared to the unsupervised scenario.

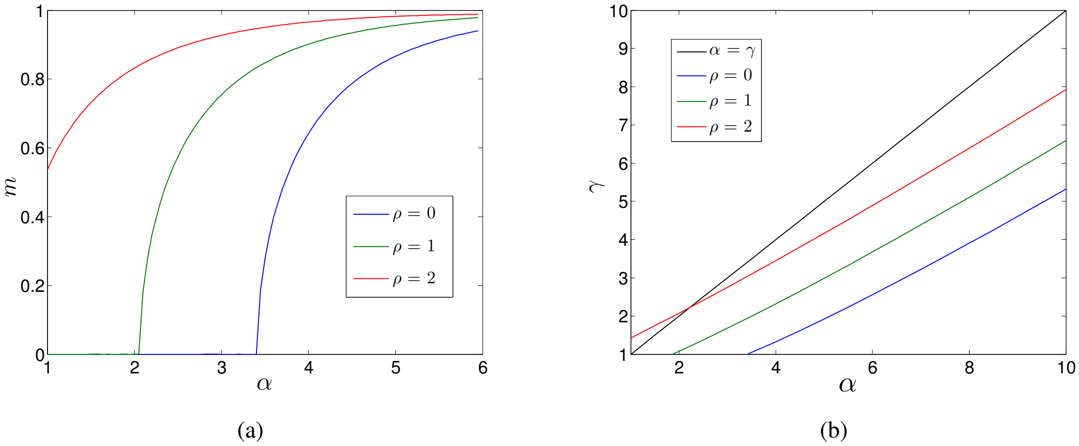


Figure 6.6: (a) Magnetization m vs. α for different ρ. (b) Detectable–non detectable boundary for different ρ.

In Figure 6.6(b) we plot the detection boundaries on (α, γ) plane for different values of ρ. For each value of ρ, the corresponding boundary $\gamma_c(\alpha; \rho)$separates two regimes, so that points below (above) the separator correspond to detectable (undetectable) clusters. The diagonal line $\alpha = \gamma$ is drawn for comparison. In the unsupervised case, the detection boundary starts at $(\alpha, \gamma) = (0, 1)$, and asymptotically behaves as $\alpha - \gamma \propto \sqrt{\alpha + \gamma}$ for large $\alpha + \gamma$. One can see that the presence of supervision keeps the shape of the boundary intact, and simply moves it upwards. Thus, for any fixed α, one can shift the threshold to $\gamma = \alpha$ (and beyond) by labeling appropriate number of edges. Note also that for a fixed number of labeled edges, the impact is stronger for sparser graphs, and diminishes as the link density increases.

We now consider the case of hard constraints, by setting $M = \infty$ [4]. In this case, the cavity equation involves all the order parameters, which makes its analysis more complicated. Instead, we address this case by solving the cavity equation using population dynamics [106]. The results are depicted in Figure 6.7(a),6.7(b). We also compare our results to simulations using synthetic data. After generating random graphs of size $N = 25,000$, we find the ground state of the Hamiltonian 6.26 using simulated annealing.

For a subset of nodes connected by labeled edges, we can determine the relative group membership for any pair in the group due to the transitivity of the constraints. Therefore, finding node assignments that satisfy hard link constraints on a graph amounts to a two-coloring problem and can be done efficiently. As we add random edges to a graph, the size of the connected clusters is well-known. For $\rho < 1$, most clusters are disconnected and the size of the largest cluster is $O(\log(N))$. At $\rho = 1$, we reach the "percolation threshold" where the size of the largest cluster goes like $O(N^{2/3})$. Once $\rho > 1$, $O(N)$ nodes belong to one giant connected component. We will investigate the consequences of these different regimes below.

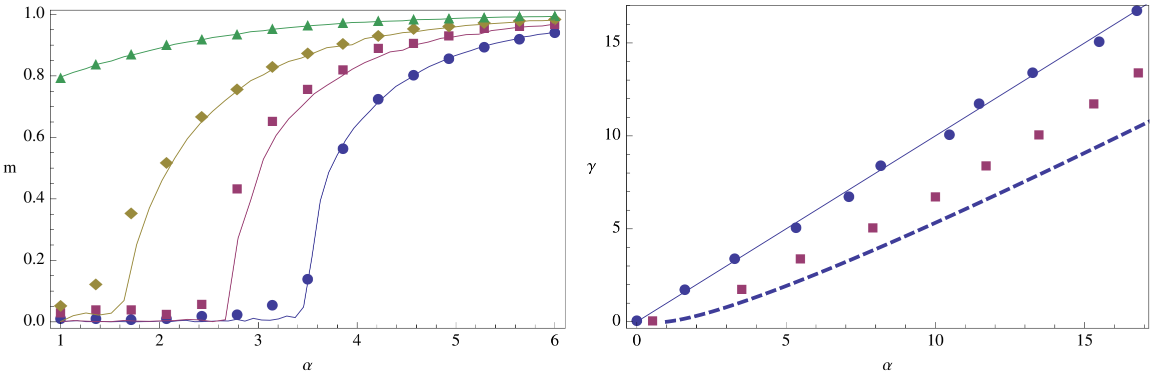
Looking at the results of Figure 6.7(a), we see that without supervision, as we vary the within-cluster connectivity, α, there is a sharp detection threshold (clusters are detectable when $m > 0$). For small amounts of supervision, $\rho < 1$, the impact of the constraints is to shift the detection threshold to smaller values of α. Qualitatively, this is no different than the effect of adding more unlabeled edges within clusters. This behavior is expected, since adding hard constraints is equivalent to studying the same unsupervised clustering problem on a *renormalized* graph (e.g., merging two nodes that are connected via constraints). This is in contrast to results for prior information on nodes in [5], which showed that even small amounts of node supervision shifted the detection threshold to its lowest possible value $\alpha = \gamma$.

As $\rho \to 1$, there is a qualitative change in our ability to detect clusters. A large number of nodes, $O(N^{2/3})$, are connected by labeled edges. If we take the relative labeling of nodes in this largest group as the "correct" one, than we have a situation similar to node supervision, which, as discussed, moves the detection threshold to $\alpha = \gamma$. While this large labeled component suffices to create non-zero magnetization in finite graphs (as seen from the simulated annealing results), as N gets large, the effect of this component diminishes. For $\rho > 1$, we see that the fraction of nodes contained in the largest labeled component suffice to produce non-zero magnetization even at the group-defining threshold $\alpha = \gamma$.

In Figure 6.7(b), we investigate the location of the detection threshold in the (α, γ) plane. For $\rho < 1$ we see that for all values of γ the threshold is simply shifted to a lower value of α, similar to the $M = 2$ case. As $\rho \to 1$, the detection threshold approaches the line that defines cluster structure $\alpha = \gamma$. For $\rho > 1$, a fraction of nodes are fixed by the edge constraints. Therefore, magnetization is nonzero even when $\alpha = \gamma$.

Note that according to our results, addition of constraints does not provide automatic improvement over the unsupervised case. Indeed, when the cost of violating constraints is finite, the only impact of the added pair–wise constraints is to *lift* the detection boundary. Thus, whether adding constraints is beneficial or not depends on the network parameters. More specifically, consider an unsupervised clustering problem with network connectivities α and γ, and let $\Delta(\rho) = \gamma - \gamma_c(\alpha, \rho)$, where $\gamma_c(\alpha)$ is the (unsupervised) detection boundary discussed in Section 1. Then adding ρ constraints per node will be beneficial only if it levitates the detection boundary above the $\alpha = \gamma$ line. Not also that for a fixed ρ, the impact of semi–supervision diminishes for large α, γ limit, and the detection threshold re–emerges.

[4] Note that fixing M to some large but finite number of $O(N)$ will guarantee that all the constraints are satisfied.



(a) (b)

Figure 6.7: (a) Magnetization plotted against α for different ρ. Lines are generated from population dynamics and points are generated from simulated annealing. From bottom to top we have $\rho = 0, 0.5, 1, 2$. (b) Location of the $m = 0$ threshold on the (α, γ) plane. Dashed line corresponds to the analytic result for $\rho = 0$ and the solid line is $\alpha = \gamma$. Squares (circles) calculated using population dynamics at $\rho = 0.5$ $(\rho = 1)$.

Chapter 7

Fighting Crime

The work presented in this chapter has been done by the group of Dr. Bertozzi and Dr. Brantingham.

1. Geographic Profiling from Kinetic Models of Criminal Behavior

UCLA postdocs George O. Mohler and Martin B. Short have considered the problem of estimating the probability density of the "anchor point" (residence, place of work, etc.) of a criminal offender given a set of observed spatial locations of crimes committed by the offender. Starting from kinetic models of criminal behavior, they derive the probability density of anchor points using the Fokker-Planck equation and Bayes' Theorem. Here geographic inhomogeneities such as housing densities and geographic barriers (bodies of water, parks, etc.) are naturally incorporated into the probability density estimate. The resulting equations are elliptic PDEs that can be solved efficiently using Multigrid or other standard computational techniques. They test their methodology against distance to crime data provided by the Los Angeles Police Department. Their results highlight the benefits of incorporating elements of criminal behavior and geographic inhomogeneities into profiling estimates.

2. PDE Models of Crime

2.1. Modeling of Urban Crime Hotspots

Terrorist and insurgent activities have a distinct parallel to urban crime in that they are constrained by the same need to encounter victims and targets in the absence of effective security. Therefore, the fundamental 'physics' of criminal offenses may be classified according to the mobility of offenders and potential targets/victims. Some crime types may arise under a full range of physical conditions (e.g., homicide), while others are more constrained (e.g., burglary). A similar classification can be developed for terrorist and insurgent attacks. The models we have developed to study crime should transfer readily to the study of terrorist and insurgent activity and event patterning. Moreover, we have allied ourselves with LAPD and Long Beach PD in order to develop direct comparisons between our models and real field data from spatially extended urban environments. As in the case of criminal offenders, routine mobility patterns are the proximate cause of how terrorists and insurgents encounter, select and attack targets. We believe that the methods we have developed to study crime pattern formation may serve as a foundation to predicting spatial and temporal patterns in terrorist attack ranging from small-scale events (e.g., sniper, hostage taking) to larger-scale actions including potential WMD attacks.

Criminologists have long known of the existence of crime "hotspots": extended geographic regions which display a higher than average rate of crime, at least temporarily. While empirical measurements of such hotspots have been numerous, there has been little progress in understanding the precise mechanisms underlying the formation and subsequent dynamics of these spots. In this regard, perhaps the most well-developed theories have been the exact- and near-repeat crime hypo theses, which posit that a geographic location and its surrounding areas experience greater rates of criminal events for a period of time following an initial event. These hypotheses have been tested both by other researchers and ourselves using a variety of crime data from around the world, and have been found to hold over a wide range of cities and crime types.

It has been our goal, therefore, to create mathematical models based around these phenomena that may shed light on how and why crime hotspots form, and may even enhance the predictive capability of law enforcement agencies, allowing them to better utilize their limited resources. Toward this end, we have thus far created two possible models of crime hotspot generation: an agent-oriented model, and a target-oriented model. The former seeks to explain the empirical observations from first principles by simulating the known behavior of real criminal agents within a landscape of targets, while the latter takes a more empirical approach and uses the historical crime data in an area to predict future events there. Both of these approaches will be explained below.

The agent-oriented model imagines a number of criminal agents that move around a virtual environment, occasionally committing criminal acts as they encounter targets. Each target i within the environment has an attractiveness value A_i associated with it. This attractiveness serves to bias crime in two way: higher A_i values directly correspond to higher probabilities of an agent committing a crime when located at target i, and agents prefer to move toward targets with higher A_i values when walking on the grid. In order to capture the exact-repeat phenomenon noted above, A_i is temporarily increased after a crime event occurs at target i. To model the near-repeat phenomenon, A_i is allowed to spatially spread to other targets j that are near i. Finally, criminal agents "return home" after offending (they are removed from the grid), and each location gives rise to new agents at a rate Γ.

The full mathematical model imagined in this discrete form possesses a number of parameters (seven), and simulations based on this model can exhibit drastically different behavior depending upon the values these parameters take. In essence, though, only three distinct regimes of behavior are observed: no hotspot formation, transitory hotspot formation, or stationary hotspot formation (see Figure 7.1). To better understand what parameter combinations lead to the various behavioral regimes, we have recast the discrete model in continuum form, deriving two coupled partial differential equations to describe the system:

$$\frac{\partial A}{\partial t} = \eta \nabla^2 A - A + A_0 + \rho A \quad \text{and} \tag{7.1}$$

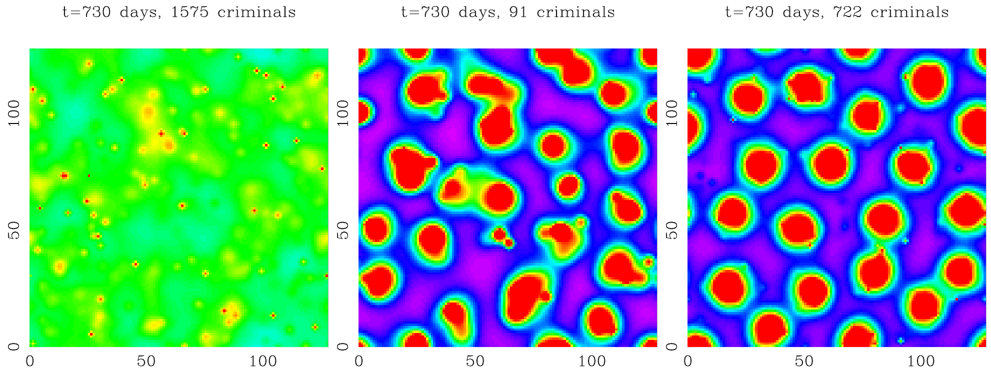
$$\frac{\partial \rho}{\partial t} = \vec{\nabla} \cdot \left[\vec{\nabla} \rho - \frac{2\rho}{A} \vec{\nabla} A \right] - \rho A + \overline{A} - A_0 \,, \tag{7.2}$$

where ρ represents the density of criminal agents, and η, A_0, and $\overline{A}$ are the only three remaining parameters of the system after the continuum limit is taken. Numerical integrations of Eqns. 7.1 and 7.2 result in hotspot maps that are quite similar to those created via the fully discrete model for corresponding parameter choices. This work has been published in the journal *M3AS: Mathematical Models and Methods in the Applied Sciences* in a special issue on traffic, crowds, and swarms [150].

In addition to their numerical integration, we have performed a variety of analytical analyses of Eqns. 7.1 and 7.2. A linear stability analysis has provided us with the understanding of hotspot formation as a classic dynamical instability, and given insights into what parameter regimes should lead to hotspot formation. Specifically, we now know that if the parameters of the system are such that the inequality

$$A^0 < \frac{2}{3}\overline{A} - \frac{1}{3}\eta \overline{A}^2 - \frac{2}{3}\overline{A}\sqrt{\eta \overline{A}} \,. \tag{7.3}$$

holds, the system will exhibit hotspots. The size and spacing of these hotspots can also be determined by the three continuum parameters. A weakly nonlinear analysis of the equations has also been performed, and the results show that hotspots may arise via either supercritical or subcritical pitchfork bifurcations, depending upon parameter values. The possibility of subcritical bifurcations allows for hysteresis effects within the system, indicating that police action may be able to permanently destroy individual hotspots, even if the police presence is eventually removed from the area. Finally, we have observed coarsening behavior within this system, whereby a steady state comprised of n hotspots may spontaneously change to a state with fewer



(a) no hotspots (b) transitory hotspots (c) stationary hotspots

Figure 7.1: Example output from simulations of the agent-oriented model. These figures illustrate the three regimes of behavior observed in this system.

than n spots, even after very long periods of seemingly no change. This could account for the sometimes rapid disappearance or emergence of hotspots observed in real crime data.

PhD student Nancy Rodriguez, under the direction of co-PI Bertozzi, developed a local existence and uniqueness of solutions to the continuum version of this model, a coupled system of partial differential equations, as well a continuation argument. She compared this PDE model with a generalized version of the Keller–Segel model for chemotaxis as a first step to understanding possible conditions for global existence vs. blow-up of the solutions in finite time. Global well-posedness of the model is still an open problem however this work develops the ground work and distinguishes the nonlinearities present in this model from prior mathematical work on related problems of bacterial chemotaxis. The paper by Rodriguez and Bertozzi has been published in M3AS [137].

2.2. Control of Hotspots by Law Enforcement

We extend an agent-based model of crime-pattern formation initiated in [150] by incorporating the effects of law enforcement agents. We investigate the effect that these agents have on the spatial distribution and overall level of criminal activity in a simulated urban setting. Our focus is on a two-dimensional lattice model of residential burglaries, where each site (target) is characterized by a dynamic attractiveness to burglary and where criminal and law enforcement agents are represented by random walkers. The dynamics of the criminal agents and the target-attractiveness field are, with certain modifications, as described in [150]. Here the dynamics of enforcement agents are affected by the attractiveness field via a biasing of the walk, the detailed rules of which define a deployment strategy. We observe that law enforcement agents, if properly deployed, will in fact reduce the total amount of crime, but their relative effectiveness depends on the number of agents deployed, the deployment strategy used, and spatial distribution of criminal activity. For certain policing strategies, continuum PDE models can be derived from the discrete systems. The continuum models are qualitatively similar to the discrete systems at large system sizes. This work was carried out by Paul Jones as part of his PhD thesis under the direction of Lincoln Chayes and Jeff Brantingham.

2.3. Bifurcation Theory for Crime Hotspots

Short, Bertozzi, Brantingham and Tita developed non-linear analyses of a PDE model of crime hostpot formation published during the previous MURI period. This work examines the non-linear stability of crime hotspots form by fundamental behaviors of the diffusion of risk associated with crime in uniform, target rich crime environments. They show that there are at least two types of parameter regimes that produce either super-critical crime hotspots or sub-critical crime hotspots. Super-critical hotspots emerge in linearly unstable regimes from small perturbations in crime and lead to apparent crime displacement when existing hotspots are suppressed by police action. By contrast, sub-critical hotspots form in linearly stable

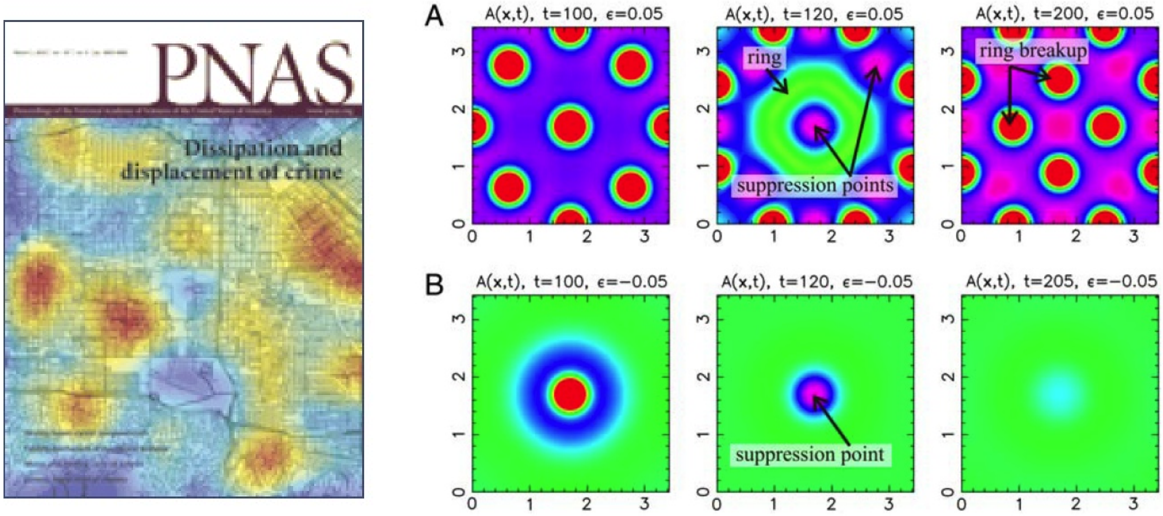


Figure 7.2: (left) Cover of March 2, 2010 issue of PNAS. (right) Crime hotspot suppression figure from the cover article [151]. Suppression results for the PDE system with parameters chosen to generate supercritical or subcritical crime hotspots. (A) Suppression of supercritical crime hotspots. Shown is the configuration of supercritical hotspots at timestep $t = 100$, just prior to the introduction of crime suppression. Crime suppression is then introduced over the area of each visible hotspot, leading to the eradication of the original hotspots but corresponding increases in risk in neighboring regions, seen at $t = 120$. The transient structure at $t = 120$ resembles a hot ring solution surrounding the location of the original central hotspot. By the time of the next suppression at $t = 200$, a new steady state featuring hotspots in positions adjacent to the original ones has been achieved. (B) Suppression of subcritical crime hotspots. Shown is a central subcritical hotspot at $t = 100$, just prior to the introduction of crime suppression. Crime suppression is then introduced over the area of the hotspot, leading to the eradication of the hotspot by $t = 120$. No transient structures appear in this case. Eventually suppression is lifted at $t = 200$ and the system quickly adopts the homogenous steady state. Color scale shows red as higher crime area and blue as lower crime area.

regimes from large perturbations in crime. When sub-critical hotspots are suppressed by police action they do not reemerge (i.e., they dissipate) even when police pressure is removed. This is a hysteresis effect that is shown in the non-linear analysis. The technical analysis is published in *SIAM Journal on Applied Dynamical Systems (SIADS)*. The general features of the model characteristics and its implications are discussed in the March 2010 cover article of the *Proceedings of the National Academy of Sciences (PNAS)*. See Figure 7.2. It also received quite a lot of attention from the Scientific and International Press - see front section for references.

2.4. Extension to Game Theory Models

The evolution of human cooperation has been the subject of much research, especially within the framework of evolutionary public goods games, where several mechanisms have been proposed to account for persistent cooperation. Yet, in addressing this issue, little attention has been given to games of a more adversarial nature, in which defecting players, rather than simply free riding, actively seek to harm others. Short, Brantingham and D'Orsogna [152] have developed a evolutionary game theoretic model that explores how population composed of different criminal and reporting strategies evolve. They use the specific example of criminal activity, recasting the familiar public goods strategies of punishers, cooperators, and defectors in this light. They introduce a strategy "the informant" with no clear analog in public goods games and show that individuals employing this strategy are a key to the emergence of systems where cooperation dominates. They also find that a defection-dominated regime may be transitioned to one that is cooperation-dominated by converting an optimal number of players into informants. They discuss these findings, the role of informants, and possible intervention strategies in extreme adversarial societies, such as those marred by wars and insurgencies. Simulations demonstrate that this idealized society has two stable equilibrium points where the population consists of either (1) offenders/non-reporters and non-offenders/non-reporters ("dystopia"), or (2) non-offenders/reporters and non-offenders/non-reporters ("utopia"). Informants who commit crimes but will also report crimes that they themselves do not commit are not present in either of these equilibria. However, simulation and an equivalent ODE system shows that informants are critical to transitioning between dystopia and utopia. This work led to the formation of a new MURI team led by Milind Tambe at USC starting this year. Current co-PIs Bertozzi and Brantingham are involved with this program as are former postdocs D'Orsogna and Short.

2.5. Gang Rivalry Networks – a Mechanistic Approach

PhD students Rachel Hegemann and Laura Smith use an agent based model to investigate social and physical geographic influences on gang rivalry network formation in the Eastern Los Angeles policing district of Hollenbeck. The model includes basic movement routines based around known territory anchor points ('set spaces') provided by the LAPD. Known physical boundaries such as highways are encoded into the model and are treated as semi-permeable boundaries by the agents. Interactions are recorded between the agents to produce simulated rivalry network. These simulated rivalry network are then compared against the known rivalry network in Hollenbeck as well as network produced by alternative methods such as geographic threshold graphs. This work also involves postdoc Alethea Barbaro and collaborators George Tita and Shannon Reid at UC Irvine. It was recently published in *Physica A* [67].

2.6. Gang Territory Development Based on Graffiti Distributions

Barbaro, D'Orsogna and Chayes study the problem of gang territory formation by simulating an interacting particle system on a lattice. The central hypothesis is that territory formation and defense occurs through territorial marking, which gangs do through graffiti tagging. We show that gang territories can develop in reaction to temporally and spatially evolving distributions of graffiti. We study a two-gang model in which agents deposit distinct graffiti territorial markers, all graffiti decays in time, and agents condition their movement patterns in response to graffiti distribution.. Using methods from statistical mechanics, we prove a phase transition occurs in this system where random distribution of gang members suddenly segregate into distinct territories.

3. Maximum Penalized Likelihood Estimation and Data Fusion

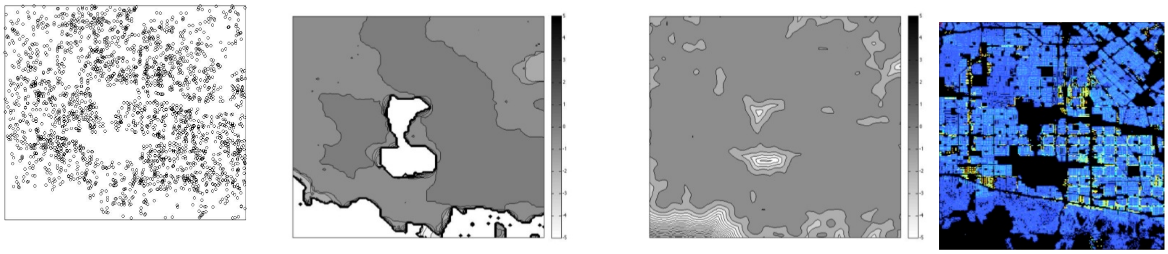


Figure 7.3: Residential burglary in 2004 for an 18x18 km area of the San Fernando Valley, Los Angeles. Point locations of crimes shown on the far left. Middle figures compare density estimation using MPLE and TV regularization (left) with more traditional kernel based methods (right). Actual housing densities are shown on the far right, which could be fused with the point process data of human activity, to create a more accurate crime density map.

3.1. TV Regularized MPLE

Total Variation-based regularization, well established for image processing applications such as denoising, was recently introduced for Maximum Penalized Likelihood Estimation (MPLE) as an effective way to estimate non-smooth probability densities. While the estimates show promise for a variety of applications, the non-linearity of the regularization leads to computational challenges, especially in multi-dimensions. George Mohler, Andrea Bertozzi, Tom Goldstein and Stan Osher present a numerical methodology, based upon the Split Bregman L1 minimization technique, that overcomes these challenges, allowing for the fast and accurate computation of 2D TV-based MPLE (see Figure 7.3). We test the methodology with several examples, including V-fold Cross Validation with large 2D data sets, and highlight the application of TV-based MPLE to point process crime modeling. This work has been accepted in *J. Computational and Graphical Statistics*.

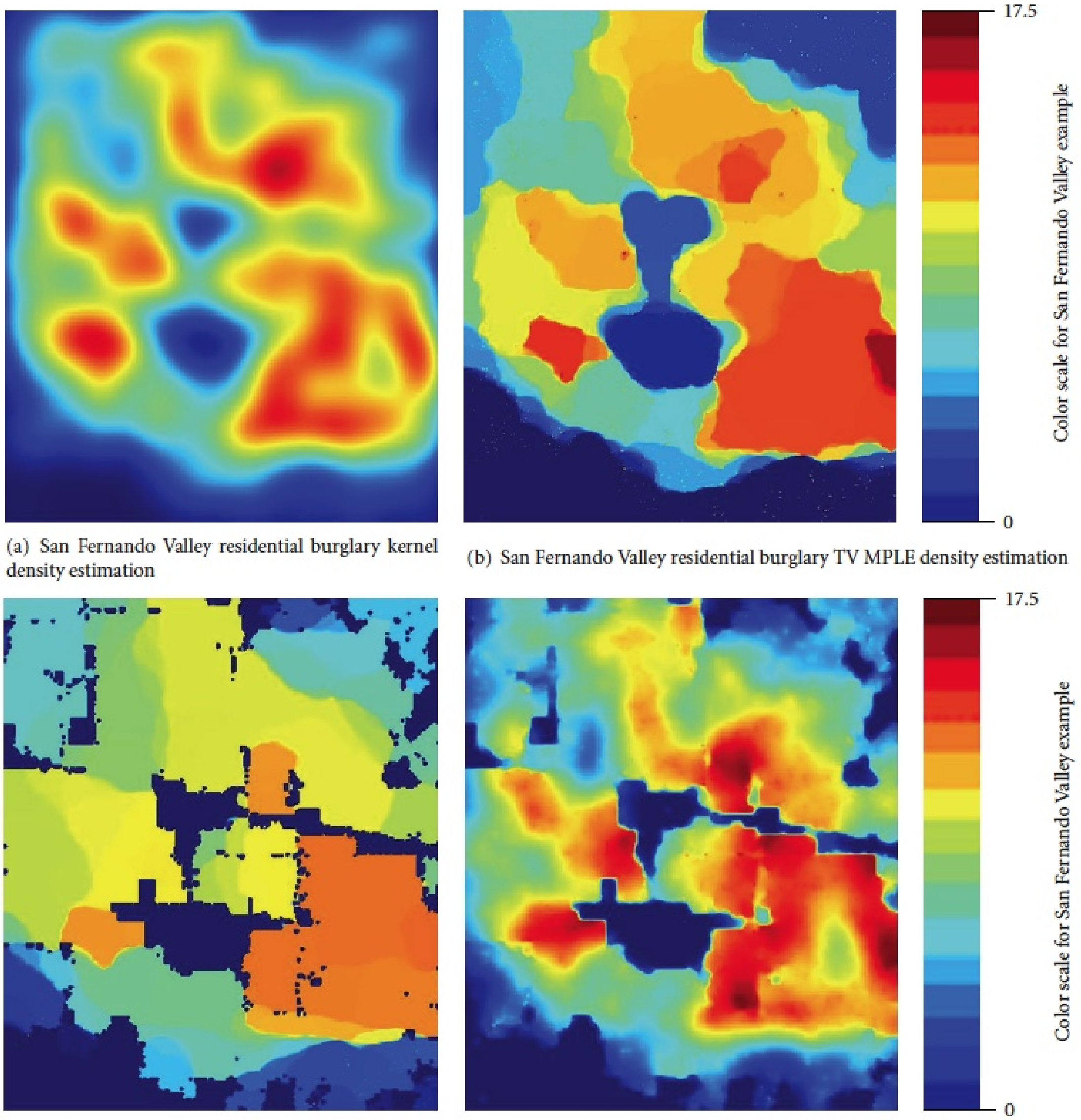
3.2. Improving Density Estimation by Incorporating Spatial Information

The TV regularized MPLE method described above can be improved by incorporating additional spatial information. We propose a set of Maximum Penalized Likelihood Estimation methods based on Total Variation and H1 Sobolev normregularizers in conjunction with a priori high resolution spatial data to obtain more geographically accurate density estimates.We apply this method to a residential burglary data set of the San Fernando Valley using geographic features obtained from satellite images of the region and housing density information. This work was performed by Laura Smith and Matthew Keegan as part of their PhD thesis work with advice from Todd Wittman, George Mohler and coPI Bertozzi. We have published a paper in EURASIP J. on Advances in Signal Processing, special issue on Advanced Image Processing for Defense and Security Applications, 2010. See Figure 7.4.

3.3. Filling in Missing Information in Gang Crime

Dynamic activity involving social networks often has distinctive temporal patterns that can be exploited in situations involving incomplete information. Even when activity is highly stochastic, localized excitations in parts of the network can help identify actors in cases of unknown origin. Pinpointing the source of unknown activity in large social networks is a combinatorially complex problem that can be more easily computed via a non-convex constrained optimization. Gang-related violent crimes pose a major problem for authorities in large cities, where cycles of retaliatory violence can lead to short but intense periods of crime.

The UCLA Institute of Pure and Applied Mathematics Research in Industrial Projects for Students (IPAM-RIPS) project team for 2010 worked on the problem of identifying unknown parties (gangs) in gang-


17.5
0
Color scale for San Fernando Valley example
17.5
0
Color scale for San Fernando Valley example

(a) San Fernando Valley residential burglary kernel density estimation

(b) San Fernando Valley residential burglary TV MPLE density estimation

(c) San Fernando Valley residential burglary modified TV MPLE density estimation

(d) San Fernando Valley residential burglary weighted H_1 MPLE density estimation

Figure 7.4: These images are the density estimates for the San Fernando Valley residential burglary data. (a) and (b) show the results of the current methods Kernel Density Estimation and TV MPLE, respectively. The results from ourModified TV MPLE method and ourWeighted H1 MPLE method are shown in figures (c) and (d), respectively. The color scale represents the number of residential burglaries per year per square kilometer. Figure taken from [155]

related violent incidents. Reports from LAPD indicate that a significant number of gang-related crimes in Hollenbeck have unknown perpetrators with unknown gang affiliation. The RIPS-LAPD project developed an algorithm that LAPD could use to predict the possible perpetrator's gang in cases where the crime is marked as gang-related but the suspect gang is not known. The algorithm uses crimes with complete information to fill in missing data fields for crimes with incomplete information. Spatio-temporal information about the crimes are used as well as information about gang territories and historical rivalries. The students created prototype algorithm that produces an accurate prediction of the gangs involved in individual crimes.

This problem was studied in more detail by Alexey Stomakhin, Martin Short, and Andrea Bertozzi in [159]. The authors considered a model in which the repeat activity between nodes of the network is modeled by a temporal Hawkes process (Figure 1). Here, the nodes of the network represent individual street gangs and the activities are violent crimes between the gangs, some of which are unsolved. The goal is to correctly identify the gang affiliated with the unsolved crimes. The authors construct an energy functional inspired by the true probabilistic likelihood associated with the Hawkes process that depends quadratically on the probability that an unsolved crime was committed by a specific gang. They maximize this functional under an l2 constraint using gradient flow. This problem is well-posed, and generally has a unique optimal solution. The algorithm performs almost identically to a combinatoric approach for small datasets, but runs in a fraction of the time; for large datasets, the combinatoric approach is computationally infeasible. For artificial datasets with properties similar to those of the Los Angeles gang network, the algorithm places the correct gang within the top 4% of likelihood approximately 80% of the time, highlighting the usefulness of this method.

4. Self-exciting Point Process Models of Crime and Insurgent Violence

4.1. Self-exciting Point Process Modeling of Residential Burglaries

Highly clustered event sequences are observed in certain types of crime data, such as burglary and gang violence, due to crime specific patterns of criminal behavior. Similar clustering patterns are observed by seismologists, as earthquakes are well known to increase the risk of subsequent earthquakes, or aftershocks, near the location of an initial event. We have developed a collaboration with Statistician Frederick Schoenberg at UCLA who is an expert on space-time clustering in seismology as modeled by self-exciting point processes. Postdocs Mohler and Short, in collaboration with Brantingham and George Tita (UC Irvine Criminologist) have developed a manuscript illustrating that these methods are well suited for criminological applications. They use residential burglary data, provided by the Los Angeles Police Department, to illustrate the implementation of self-exciting point process models in the context of urban crime. For this purpose they use a fully non-parametric estimation methodology to gain insight into the form of the space-time triggering function and temporal trends in the background rate of burglary. This work has been published in the *J. of the Am. Statistical Assoc.* [108].

4.2. Self-exciting Point Process Models for Gang Activity

Gang violence has plagued the Los Angeles policing district of Hollenbeck for over half a century. With sophisticated models, police may better understand and predict the region's frequent gang crimes. During summer 2009 we organized a summer REU (research experience for undergraduates) project to address whether self-excitation could be quantified in Hollenbeck's gang rivalries. A self-exciting point process called a Hawkes process was used to model rivalries over time. Figure 7.5 shows computed arrival rate functions for the Locke-Lowell rivalry determined from police data and resulting numerical simulations that can be performed in a model for this rivalry. While this is shown to

fit the data well, an agent based model is presented which is able to accurately simulate gang rivalry crimes not only temporally but also spatially. Finally, the students compared random graphs generated by the agent model to existing models developed to incorporate geography into random graphs. This work was published in SIAM Undergraduate Research Online [52] by the team of students.

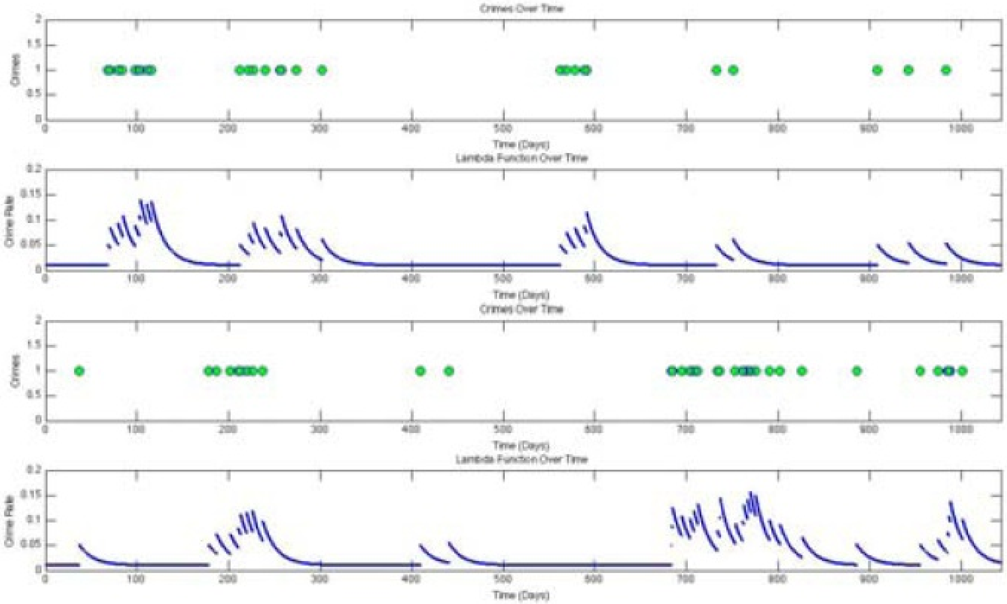


Figure 7.5: On top, a plot of the Locke-Lowell rivalry's crimes over time with the respective arrival rate function $\lambda(t)$. On bottom, simulated crimes from a Hawkes process with the Locke-Lowell rivalry's parameters and the corresponding rate function, $\lambda(t)$. Figure from [52].

In summer 2010 Kym Louie, Mark Allenby and Marina Masaki comprised an undergraduate team mentored by Tim Lucas (Pepperdine) onsite at UCLA expanding on the work described above to include both spatial dependence and directionality of the rivalry behavior. The group presented a model to simulate directed crimes in the 33 gang system of Hollenbeck [6].

4.3. Self-exciting Point Process Models for Insurgent Activity

We recently purchased a copy of the Iraq Body Count Data, compiled by an organization dedicated to accurately recording all civilian deaths in Iraq [35]. The number of fatalities linked to any event is not an estimate by the organization, but a count corroborated by at least two reliable news sources. In the data we consider, from March 20, 2003 to December 31, 2007, there are 15,977 events. Each entry in the data contains a start date, end date, minimum number of deaths, maximum number of deaths, town and possibly a district of where the event occurs. Our goal in this paper is to analyze temporal patterns of civilian death reports. For this purpose we employ a branching point process model similar to those used in earthquake analysis. Here the rate of events is partitioned into the sum of a Poisson background rate and a self-exciting component in which events trigger an increase in the rate of the process. More specifically, each event generated by the process in turn generates a sequence of offspring events according to a Poisson distribution. Whereas the background rate is typically assumed to be stationary for seismic activity, such an assumption is not valid in the context of civilian deaths in Iraq. We propose three simple adjustments to account for background rate variation and compare the effectiveness of each model using Iraq Body Count data from 2003 to 2007. Our results indicate that branching point processes are well suited for modeling the temporal dynamics of violence in Iraq. This work was performed by PhD student Erik Lewis with help from George Mohler and Andrea Bertozzi. Jeff Brantingham obtained the data and helped with interpretation of the data by the mathematics. The work has just been published in *Security Journal* [86].

Chapter 8

Theoretical Results in Quickest Changepoint Detection

This chapter is intended to summarize our contributions in the theory of changepoint detection made with the support of this grant. The work has been performed by the group of Dr. Tartakovsky.

1. The General Problem and Preliminaries

Throughout this chapter we will focus on the basic iid setting of the changepoint detection problem. The setting assumes that one is able to sequentially gather a series of independent random observations, $\{X_n\}_{n\geqslant 1}$. The observations are such that $X_1, X_2, \dots, X_\nu$ are each distributed according to a *known* density f, and $X_{\nu+1}, X_{\nu+2}, \dots$ each adhere to a density $g \not\equiv f$, also *known*. The time instant $0 \leqslant \nu < \infty$ is referred to as the changepoint, and is assumed *unknown*; henceforth, $\nu = \infty$ will mean that all X_n's have density f, and $\nu = 0$ – that all X_n's have density g. The objective is to detect as quickly as possible that the change is in effect, subject to a constraint on the risk of sounding a false alarm. A sequential detection procedure is defined as a stopping time T (defined with respect to the observed data), so that after observing $X_1, X_2, \dots, X_T$ it is declared that a change may be in effect.

We will be interested in the following detection procedures. First, Page's [119] CUmulative SUM (CUSUM) chart. It is based on maximizing the likelihood ratio (LR), and can be defined as the stopping time $\mathcal{C}_A = \inf\{n \geqslant 1\colon V_n \geqslant A\}$, where $V_n = \max\{1, V_{n-1}\}\,\Lambda_n$, $n \geqslant 1$, with $V_0 = 1$, $\Lambda_n = g(X_n)/f(X_n)$ is the LR for the n-th data point, and $A > 0$ is a detection threshold, which determines the procedure's operating characteristics; hereafter in every definition of a detection procedure it will be assumed that $\inf\{\varnothing\} = \infty$.

Next, the Shiryaev–Roberts (SR) procedure. This procedure is due to the independent work of Shiryaev [146, 148], who considered the problem of detecting a change in the drift of Brownian motion, and Roberts [136], who studied detecting a change in the mean of an iid Gaussian sequence. The SR procedure is defined by the stopping time

$$\mathcal{S}_A = \inf\{n \geqslant 1\colon R_n \geqslant A\}, \tag{8.1}$$

where $A > 0$ and

$$R_n = (1 + R_{n-1})\,\Lambda_n, \quad n \geqslant 1 \;\text{ with }\; R_0 = 0. \tag{8.2}$$

Pollak [123] proposed to tweak the SR procedure by starting it off a random point, R_0^Q, sampled from $\{R_n\}_{n\geqslant 0}$'s quasi-stationary distribution. The cdf of this distribution, $Q_A(x)$, is defined as $Q_A(x) = \lim_{n\to\infty} \mathbb{P}_\infty(R_n \leqslant x | \mathcal{S}_A > n)$.

Pollak's [123] tweaked version of the SR procedure, known as the Shiryaev–Roberts–Pollak (SRP) procedure, is defined by the stopping time

$$\mathcal{S}_A^Q = \inf\{n \geqslant 1\colon R_n^Q \geqslant A\}, \tag{8.3}$$

where

$$R_{n+1}^Q = (1 + R_n^Q)\,\Lambda_{n+1}, \quad n \geqslant 0, \quad \text{with } R_0^Q \propto Q_A(x). \tag{8.4}$$

The performance of a detection procedure is judged based on the desired optimality criteria. We will be interested in two described below.

Henceforth, let $\mathbb{P}_k(\cdot)$ and $\mathbb{P}_\infty(\cdot)$ be the probability measures generated by the data $\{X_n\}_{n\geqslant 1}$, respectively, when the changepoint is $\nu = k$, $0 \leqslant k < \infty$, and $\nu = \infty$. Let $\mathbb{E}_k[\cdot]$ and $\mathbb{E}_\infty[\cdot]$ denote the corresponding expectations.

For a generic detection rule, T, the risk of sounding a false alarm is measured via the Average Run Length (ARL) to false alarm defined as $\mathsf{ARL}(T) = \mathbb{E}_\infty[T]$; see Lorden [95]. Let

$$\Delta(\gamma) = \{T\colon \mathsf{ARL}(T) \geqslant \gamma\} \tag{8.5}$$

be the class of procedures for which the ARL to false alarm is no less than the desired *a priori* set level $\gamma > 1$.

The first optimality criterion we will be interested in is due to Pollak [123], and it seeks to find $T_{opt} \in \Delta(\gamma)$ such that $\mathsf{SADD}(T_{opt}) = \inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T)$, for every $\gamma > 1$, where hereafter

$$\mathsf{SADD}(T) = \sup_{0\leqslant\nu<\infty} \mathsf{ADD}_\nu(T), \quad \text{and} \quad \mathsf{ADD}_\nu(T) = \mathbb{E}_\nu[T-\nu|T>\nu]. \tag{8.6}$$

To date, no solution to this problem has been found. Alternatively, three types of asymptotic optimality are distinguished.

Definition 1.1. A procedure $T^*_{opt} \in \Delta(\gamma)$ is *order-1 asymptotically optimal* if

$$\lim_{\gamma\to\infty} \frac{\mathsf{SADD}(T^*_{opt})}{\inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T)} = 1, \quad \text{i.e.,} \quad \inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T) = \mathsf{SADD}(T^*_{opt})[1+o(1)], \quad \text{as } \gamma\to\infty,$$

where hereafter $o(1) \to 0$, as $\gamma\to\infty$.

A procedure $T^*_{opt} \in \Delta(\gamma)$ is *order-2 asymptotically optimal* if $\mathsf{SADD}(T^*_{opt}) - \inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T) = O(1)$, as $\gamma\to\infty$, where hereafter $O(1)$ is bounded, as $\gamma\to\infty$.

A procedure $T^*_{opt} \in \Delta(\gamma)$ is *order-3 asymptotically optimal* if $\mathsf{SADD}(T^*_{opt}) - \inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T) = o(1)$, as $\gamma\to\infty$.

We now describe the second optimality criterion we will consider. The criterion is known as "multi-cyclic disorder detection in a stationary regime". Let $T_1, T_2, \dots$ denote sequential independent applications of the same stopping time T, and let $\mathcal{T}_{(j)} = T_{(1)} + T_{(2)} + \cdots + T_{(j)}$ be the time of the j-th alarm, $j \geqslant 1$. Let $I_\nu = \min\{j \geqslant 1\colon \mathcal{T}_{(j)} > \nu\}$ so that $\mathcal{T}_{(I_\nu)}$ is the point of detection of the true change, which occurs at time instant ν after $I_\nu - 1$ false alarms have been raised. Consider $\mathsf{STADD}(T) = \lim_{\nu\to\infty} \mathbb{E}_\nu[\mathcal{T}_{(I_\nu)} - \nu]$, i.e., the limiting value of the ADD that we will refer to as the stationary ADD (STADD). The multi-cyclic optimality criterion consists in finding $T_{\text{opt}} \in \Delta(\gamma)$ such that $\mathsf{STADD}(T_{opt}) = \inf_{T\in\Delta(\gamma)} \mathsf{STADD}(T)$ for every $\gamma > 1$. For the basic iid version of the changepoint detection problem, Pollak and Tartakovsky [125], showed the optimum to be the SR procedure.

The rest of the chapter is devoted to providing a summary of our accomplishments made over the course of this project for the aforestated problem.

2. Efficient Performance Evaluation for a Class of Detection Procedures

This section is a summary of the work of Moustakides, Polunchenko, and Tartakovsky [113], namely, the part concerned with the problem of efficient performance evaluation for a class of detection schemes. The class is all stopping rules whose detection statistic is a Markov process; in particular, the SR rule, the SRP procedure and the CUSUM chart all belong to this class. We proposed a numerical framework whereby one can evaluate the performance of virtually any detection procedure and with respect to any performance index. Additionally, the framework supplies a concise numerical method for computing the quasi-stationary distribution, thus making the SRP scheme applicable in practice. This framework can be of interest to many scientists in various disciplines where there is need for an on-line detection of changes (or anomalies) in observed processes.

Note that though we have confined ourselves to the case of deterministic unknown changepoint ν, there is also another, Bayesian point of view, which assumes that ν is a random variable with a certain prior distribution. The methodology of Moustakides, Polunchenko, and Tartakovsky [113] can be (and was) extended to the Bayesian context as well; see Tartakovsky and Moustakides [168], Tartakovsky, Pollak, and Polunchenko [175], and Polunchenko and Tartakovsky [129].

Consider a generic detection procedure described by the stopping time $T_A^v = \inf\{n \geqslant 1\colon V_n^v \geqslant A\}$, where $A > 0$ is the detection threshold, and $\{V_n^v\}_{n\geqslant 0}$ is a generic detection statistic generated recursively as $V_n^v = \xi(V_{n-1}^v)\,\Lambda_n$, $n \geqslant 1$ with $V_0^v = v \geqslant 0$. Here $\xi(x)$ is a known sufficiently smooth function such that $\xi(x) > 0$ for any $x \in [0, A)$, and v is a fixed parameter, referred to as the starting point, or the head start.

Note that T_A^v can be turned into the CUSUM chart by setting $\xi(x) = \max\{1, x\}$, and similarly, the choice $\xi(x) = 1 + x$ will "do the trick" for any SR-type rule. Hence, one can evaluate any Operating Characteristic (OC) of any procedure that is a special case of T_A^v simply by choosing the right $\xi(x)$.

Let $A > 0$ and $v \in [0, A)$ be fixed, and define $\phi_d(v) = \mathbb{E}_d[T_A^v]$ and $P_d^{\Lambda}(t) = \mathbb{P}_d(\Lambda_1 \leqslant t)$, where $d = \{0, \infty\}$; clearly, $\phi_\infty(r) = \mathsf{ARL}(T_A^v)$ and $\phi_0(v) = \mathsf{ADD}_0(T_A^v)$. Let

$$\mathcal{K}_d(x, y) = \frac{\partial}{\partial y}\mathbb{P}_d(V_{n+1}^v \leqslant y | V_n^v = x) = \frac{\partial}{\partial y} P_d^{\Lambda}\left(\frac{y}{\xi(x)}\right),\; d = \{0, \infty\}$$

denote the transition probability density kernel for the homogeneous Markov process $\{V_n^v\}_{n\geqslant 1}$; note that both $\mathcal{K}_d(x, y)$, $d = \{0, \infty\}$ depend on $\xi(x)$.

We now proceed to stating the equations. First, it can be shown that

$$\phi_d(v) = 1 + \int_0^A \mathcal{K}_d(v, y)\,\phi_d(y)\,dy; \tag{8.7}$$

cf. Moustakides, Polunchenko, and Tartakovsky [113].

Next, consider $\mathsf{ADD}_\nu(T_A^v) = \mathbb{E}_\nu[T_A^v - \nu | T_A^v > \nu]$ for an arbitrary fixed $\nu \geqslant 1$; note that $\mathsf{ADD}_0(T_A^v) \equiv \phi_0(v)$ for all v. To evaluate $\mathsf{ADD}_\nu(T_A^v)$, Moustakides, Polunchenko, and Tartakovsky [113] first argue that, since at time instance $\nu \geqslant 1$ no change is yet in effect and each observation, X_n, $1 \leqslant n \leqslant \nu$, is still f-distributed, it must be that $\mathbb{P}_\nu(T_A^v > \nu) = \mathbb{P}_\infty(T_A^v > \nu)$ for all x. Consequently, $\mathsf{ADD}_\nu(T_A^v) = \mathbb{E}_\nu[(T_A^v - \nu)^+]\,/\,\mathbb{P}_\infty(T_A^v > \nu)$, and therefore we are to turn attention to $\delta_\nu(v) = \mathbb{E}_\nu[(T_A^v - \nu)^+]$ and $\rho_\nu(v) = \mathbb{P}_\infty(T_A^v > \nu)$. For either, it is direct to see that

$$\delta_\nu(v) = \int_0^A \mathcal{K}_\infty(v, y)\,\delta_{\nu-1}(y)\,dy \;\text{ and }\; \rho_\nu(v) = \int_0^A \mathcal{K}_\infty(v, y)\,\rho_{\nu-1}(y)\,dy,$$

where $\nu \geqslant 1$, $\delta_0(v) = \phi_0(v)$ is as in (8.7), and $\rho_0(v) \equiv 1$ for all v, since $\mathbb{P}_\infty(T_A^v > 0) \equiv 1$ for all x; cf. Moustakides, Polunchenko, and Tartakovsky [113]. As soon as $\delta_\nu(v)$ and $\rho_\nu(v)$ are found, by the above argument $\mathsf{ADD}_\nu(T_A^v)$ can be evaluated as the ratio $\delta_\nu(x)/\rho_\nu(x)$. Furthermore, using $\mathsf{ADD}_\nu(T_A^v)$'s computed for sufficiently many successive ν's beginning from $\nu = 0$ and higher, one can also evaluate $\mathsf{SADD}(T_A^v) = \sup_{0\leqslant\nu<\infty} \mathsf{ADD}_\nu(T_A^v)$, since $\mathsf{ADD}_\infty(T_A^v) = \lim_{\nu\to\infty} \mathsf{ADD}_\nu(T_A^v)$ is independent of the starting point, $V_0^v = v$.

Yet another performance measure of much interest is the local (conditional) false alarm probability $\mathbb{P}_\infty(T \leqslant k+m|T>k)$, $k \geqslant 0$ inside a fixed "window" of size $m \geqslant 1$ ($m=1$ represents the probability of an instantaneous false alarm). In particular, $\sup_{k\geqslant 0}\mathbb{P}_\infty(T \leqslant k+m|T>k)$ can serve as an alternative to the ARL to false alarm. See Tartakovsky [165]. One can readily conclude that $\mathbb{P}_\infty(T_A^v < k+m|T_A^v > k) = 1-\rho_{k+m}(v)/\rho_k(v)$.

We now proceed to $\mathsf{STADD}(T_A^v)$, i.e, to

$$\mathsf{STADD}(T_A^v) = \left(\sum_{\nu=0}^{\infty}\mathbb{E}_\nu[(T_A^v-\nu)^+]\right) / \, \mathbb{E}_\infty[T_A^v],$$

and if we let $\psi(v) = \sum_{\nu=0}^{\infty}\mathbb{E}_\nu[(T_A^v-\nu)^+] = \sum_{\nu=0}^{\infty}\delta_\nu(x)$, then $\mathsf{STADD}(T_A^v) = \psi(v)/\ell(v)$. It can be shown that

$$\psi(x) = \delta_0(x) + \int_0^A \mathcal{K}_\infty(x,y)\,\psi(y)\,dy; \tag{8.8}$$

cf. Moustakides, Polunchenko, and Tartakovsky [113]. We also note that $\psi(v)$ cannot be computed prior to $\delta_0(v) \equiv \phi_0(v)$ as the former depends on the latter.

Consider now randomizing the starting point, $V_0^v = v$, in a fashion analogous that behind the SRP stopping time. Let $Q_A(x) = \lim_{n\to\infty}\mathbb{P}_\infty(V_n^v \leqslant x|T_A^v > n)$ be the cdf of the corresponding quasi-stationary distribution; note that this distribution exists, as guaranteed by Harris [66, Theorem III.10.1].

Let $T_A^Q = \inf\{n \geqslant 1 \colon V_n^Q \geqslant A\}$, where $A>0$ and $V_n^Q = \xi(V_{n-1}^Q)\,\Lambda_n$, $n \geqslant 1$ with $V_0^Q \propto Q_A$, and $\xi(x)$ and $Q_A(x)$ are as defined above; note that T_A^Q can be turned into the SRP procedure by setting $\xi(x) = 1+x$.

For T_A^Q, any OC is dependent upon the quasi-stationary distribution. We therefore first state the equation that determines $q_A(x) = dQ_A(x)/dx$, the quasi-stationary pdf; the equation can be seen to be

$$\lambda_A\, q_A(y) = \int_0^A q_A(x)\,\mathcal{K}_\infty(x,y)\,dx, \;\; \text{subject to} \;\; \int_0^A q_A(x)\,dx = 1; \tag{8.9}$$

cf. Moustakides, Polunchenko, and Tartakovsky [113] and Pollak [123]. We note that $q_A(x)$ and λ_A are both unique. Once $q_A(x)$ and λ_A are found, one can compute $\bar{\phi}_\infty = \mathsf{ARL}(T_A^Q)$, and $\bar{\delta} = \mathsf{ADD}_0(T_A^Q) = \mathsf{ADD}_\nu(T_A^Q)$, $\nu \geqslant 1$, i.e., for T_A^Q, the detection delay is independent from the changepoint. We have

$$\bar{\phi}_\infty = 1/(1-\lambda_A) \;\; \text{and} \;\; \bar{\delta} = \int_0^A \delta_0(x)\,q_A(x)\,dx;$$

cf. Moustakides, Polunchenko, and Tartakovsky [113].

The second equality in the above formula for $\bar{\phi}_\infty$ is due to the fact that, by design, the $\mathbb{P}_\infty$-distribution of the discrete random variable T_A^Q is exactly geometric with parameter $1-\lambda_A$; note that $0<\lambda_A<1$. Put otherwise, $\mathbb{P}_\infty(T_A^Q > \nu) = \lambda_A^\nu$, where $\nu \geqslant 0$; in general, $\lim_{A\to\infty}\lambda_A = 1$. Also, Pollak and Tartakovsky [127] provide sufficient conditions for λ_A to be an increasing function of A; in particular, they show that if the cdf of $\log\Lambda_1$ under measure $\mathbb{P}_\infty$ is concave, λ_A is increasing in A.

To conclude, we have now obtained a set of (exact) integral equations and relations governing all commonly used performance measures (OC-s) for a broad spectrum of detection procedures; equations (8.7), (8.8), and (8.9) comprise the cadre of the set. These equations are Fredholm (linear) integral equation of the second kind, which are known to rarely permit for an analytical solution. Hence, to deal with the corresponding integral equations a numerical solver may be in order. One offered by Moustakides, Polunchenko, and Tartakovsky [113] is a piecewise-constant (zero-order polynomial) collocation method with the interval of integration $[0, A]$ partitioned into $N \geqslant 1$ equally long subintervals. The collocation nodes are the subintervals' middle points. As the simplest case of the piecewise collocation method (see, e.g., Atkinson and Han [9]), the question of accuracy is a well-understood one, and tight error bounds can be easily obtained from, e.g., Atkinson and Han [9]. Specifically, it can be shown that the uniform L_∞ norm of the difference between the exact solution and the approximate one is $O(1/N)$, provided N is sufficiently large.

3. The Shiryaev–Roberts–r Procedure

Since its inception in 1985, the SRP procedure has been (not unfoundedly) believed to possibly be exactly optimal with respect to $\mathsf{SADD}(T)$ in the class $\Delta(\gamma)$. Although many have tried to prove this hypothesis to be true, none have succeeded. We show this hypothesis to be false. This conclusion is archived by proposing an extension of the SR rule competitive to the SRP procedure, and performing direct performance comparison of the twain. The idea of the new procedure is to let the SR detection statistic R_n to start from a fixed deterministic point $R_0 = r \geqslant 0$. We examined the performance of the resulting SR–r scheme in relation to the starting point, and proposed an initializing value for which the SRP rule is *uniformly worse*. This was demonstrated both numerically and analytically, i.e., by virtue of a counterexample. We also suggest a starting point for which the SR–r scheme exhibits a faster initial response to early changes. For details see Moustakides, Polunchenko, and Tartakovsky [113], Polunchenko and Tartakovsky [128], and Tartakovsky and Polunchenko [169].

We first introduce the SR–r procedure. It is defined as the stopping time $\mathcal{S}_A^r = \inf\{n \geqslant 1 \colon R_n^r \geqslant A\}$, where $A > 0$ and

$$R_n^r = (1 + R_{n-1}^r)\,\Lambda_n, \quad n \geqslant 1 \;\text{ with }\; R_0^r = r \geqslant 0. \tag{8.10}$$

The extra "r" in the name is to emphasize the importance of the starting point (head start). The question is now: Can one design the head start so as to obtain a procedure, capable of competing with Pollak's SRP procedure? The answer is "yes", which we will explain in the remainder of this section.

First, recall that the direct way to assess the quality of a detection procedure is to compare it against the exact optimum. However, no exactly $\mathsf{SADD}(T)$-optimal procedure has yet been proposed. Hence, an alternative approach is in order. As a point of reference one could use a *lower bound* on the (unknown) optimum. The following theorem shows that finding such a bound is a much easier task than actually designing the (exactly) optimal test.

Theorem 3.1. *Consider $\mathcal{S}_A^r$, and let $A = A_\gamma$ be selected so that $\mathsf{ARL}(\mathcal{S}_{A_\gamma}^r) = \gamma$. Then $\inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T) \geqslant \underline{\mathsf{SADD}}(\mathcal{S}_A^r)$ for every $r \geqslant 0$, where*

$$\underline{\mathsf{SADD}}(\mathcal{S}_A^r) = \frac{r\mathsf{ADD}_0(\mathcal{S}_A^r) + \sum_{k=0}^{\infty} \mathbb{E}_k[(\mathcal{S}_A^r - k)^+]}{r + \mathsf{ARL}(\mathcal{S}_A^r)}. \tag{8.11}$$

Let us now fix threshold $A > 0$, and propose the following starting point

$$r_A = \underset{0\leqslant r<A}{\arg\inf}\Big\{\mathsf{SADD}(\mathcal{S}_A^r) - \underline{\mathsf{SADD}}(\mathcal{S}_A^r)\Big\} \tag{8.12}$$

as a possible candidate for initialization of the SR–r scheme. In other words, we select the value that brings the two bounds (upper and lower) as close to each other as possible. It can be seen that the resulting stopping time, $\mathcal{S}_A^{r_A}$, is a function of $A > 0$ only, which is set so that the false alarm constraint is satisfied with equality. In the next subsection we offer a numerical study of the performance of the SR–r procedure for various values of r using the numerical framework of Moustakides, Polunchenko, and Tartakovsky [113].

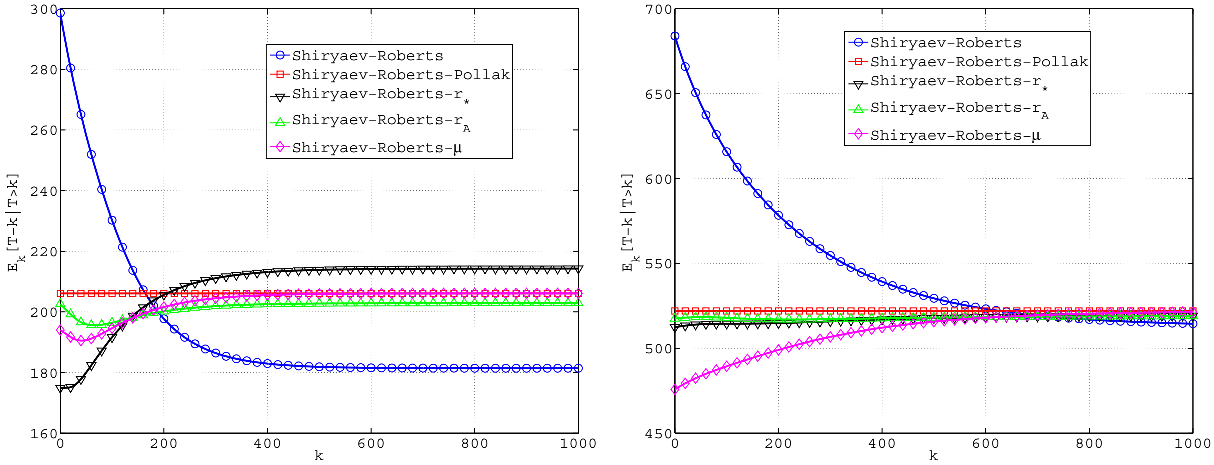
3.1. An Example: Gaussian Scenario

Let $\{X_n\}_{n\geqslant 1}$ be independent unit-variance Gaussian. Specifically, assume $X_1, X_2, \dots, X_\nu$ are each $\mathcal{N}(0,1)$ and $X_{\nu+1}, X_{\nu+2}, \dots$ are each $\mathcal{N}(\theta, 1)$, where $\theta \neq 0$, a *known* constant.

Apart from the initialization strategies introduced above, namely the classical SR test (with $r = 0$), the SRP test, the SR–r_A and the SR–$r_\star$ procedures, where $r_\star$ is to the smallest r for which $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$ an increasing function of ν, we will also examine the case of the SR–μ, where $r = \mu$, the mean of the quasi-stationary distribution.

Let $\theta = 0.1$ (which corresponds to a relatively faint, not easily detectable change), and consider two cases: $\gamma = 10^3$ and $\gamma = 10^4$, i.e., moderate and low risk of sounding a false alarm. This translates into the detection threshold, A, being in the range $10^3 \pm 1\%$ and $10^4 \pm 1\%$, respectively. To solve the corresponding integral equations and obtain the desired OC-s, we partition the interval $[0, A)$ into $N = 10^4$ (for $\gamma = 10^3$) and $N = 10^5$ (for $\gamma = 10^4$) equidistant nodes. This is sufficient to provide the accuracy of 0.5% (confirmed by Monte Carlo simulations with 10^6 repetitions).

Before proceeding with the presentation of our computational results, we would like to mention that in order to evaluate the ARL to false alarm of the SR–r and SRP procedures, it is important to have a fairly accurate initial guess, i.e., to obtain a pilot estimate of $\mathsf{ARL}(\mathcal{S}_A^r)$ to search for appropriate threshold values in a relatively narrow interval. To this end, the following approximation $\mathsf{ARL}(\mathcal{S}_A^r) = (A - r)/\zeta$ is used, where the constant $\upsilon \in (0, 1)$ (related to the "overshoot") is the subject of renewal theory and can be computed numerically. This approximation can be obtained by noticing that $R_n^r - n - r$ is a $\mathbb{P}_\infty$-martingale with zero expectation, so that by the optional sampling theorem we have $\mathbb{E}_\infty[R_{\mathcal{S}_A^r}^r - \mathcal{S}_A^r - r] = 0$. Hence $\mathsf{ARL}(\mathcal{S}_A^r) = \mathbb{E}_\infty[R_{\mathcal{S}_A^r}^r] - r$, and, since $R_{\mathcal{S}_A^r}^r$ is the first excess over A, renewal theory can be applied to the "overshoot" $\log(R_{\mathcal{S}_A^r}^r) - \log A$. This approximation was first derived for $r = 0$ in Pollak [124], and its generalization for any $r \in [0, A)$ is straightforward. For the SRP procedure the value of r should be replaced by $\mu = \mathbb{E}_\infty[R_0^Q]$, the mean of the quasi-stationary distribution.



(a) $\gamma = 10^3$

(b) $\gamma = 10^4$

Figure 8.1: $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$ for different procedures as a function of the changepoint ν for $\theta = 0.1$.

Shown in Figure 8.1(a) is the family of curves $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$ versus ν for all initialization procedures in question when $\theta = 0.1$ and ARL to false alarm $\gamma = 10^3$. Figure 8.1(b) shows the same for $\gamma = 10^4$. Table 8.1 reports the numerical values obtained by our computational method for characteristic values of the change time and for the case of ARL to false alarm equal to 10^3.

Table 8.1: $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$ versus ν for $\gamma = 10^3$ and $\theta = 0.1$.

Procedure $\backslash\nu$	0	50	100	200	400	600	800	1000
SR	298.5	258.3	230.2	197.7	182.9	181.5	181.4	181.4
SR–r_A	202.8	195.9	196.4	200.1	202.5	202.8	202.8	202.8
SR–$r_\star$	174.9	179.9	191.6	205.6	213.1	214.1	214.2	214.3
SR–μ	194.0	190.7	194.6	201.6	205.6	206.0	206.1	206.1
SRP	206.1							

As expected, the SRP procedure is an equalizer. The SR–$r_\star$ test has the fastest initial response (for immediate and early changes), but the worst minimax behavior. The SR–r_A procedure is uniformly better than all competing strategies including the SRP test. In the latter comparison, even though the difference is not dramatic, it is visible. It is interesting to note that the SR–μ rule has an intermediate performance between SR–r_A and SR–$r_\star$, namely sufficiently fast initial response and a minimax performance attained at the steady state which is the same as the SRP test.

Regarding the conventional SR test (with $r = 0$) note that it outperforms all its competing schemes, including SRP for sufficiently large change-time $\nu = k$. This is expected since, as we can see from Figure 8.2(b), when all tests have the same threshold the SR test has the largest ARL to false alarm and the same steady state value for the expected detection delay. For the other tests, in order to attain the same as the SR test ARL to false alarm, the thresholds should be increased. This will result in an increase in the expected detection delay and in particular the corresponding steady state value. Consequently, the expected delay of SR, due to its monotone behavior, will attain smaller values than the other tests for sufficiently large change-time $\nu = k$.

To sum up, the best (in the minimax sense with Pollak's measure of detection delay) performance is delivered by the SR–r procedure. By design, performance-wise this rule is very close to the lower bound $\underline{\mathsf{SADD}}(T)$ given by (8.11). This suggests that the unknown exactly $\mathsf{SADD}(T)$-optimal procedure can offer only a practically insignificant improvement over the SR–r rule. Furthermore, the example considered in the next section indicates that the SR–r procedure may, in fact, be the sought optimum.

3.2. Exact Optimality of the SR–r Procedure

This section constructs an analytical counterexample, which supplies a decisive negative answer to the question of possible minimax optimality of the SRP procedure.

Theorem 3.2. *Let $f(x) = e^{-x}\mathbb{1}_{\{x\geqslant 0\}}$ and $g(x) = 2e^{-2x}\mathbb{1}_{\{x\geqslant 0\}}$. Assume the SR–$r$ starts off $r_A = \sqrt{1+A}-1$, where A solves the transcendental equation*

$$A + (\gamma-1)\sqrt{1+A}\log(1+A) - 2(\gamma-1)\sqrt{1+A} = 0.$$

Then, for every $1 < \gamma < \gamma_0 = (1-0.5\log 3)^{-1} \approx 2.2188$, $\mathsf{ARL}(\mathcal{S}_A^{r_A}) = \gamma$ and the SR–r procedure is minimax, i.e., $\mathsf{SADD}(\mathcal{S}_A^{r_A}) = \inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T)$ for every $\gamma \in (1,\gamma_0)$.

Let the threshold in the SRP procedure be chosen as $A^* = \exp\{2(\gamma-1)/\gamma\} - 1$. Then $\mathsf{ARL}(\mathcal{S}_{A^*}^Q) = \gamma$ and $\mathsf{SADD}(\mathcal{S}_{A^*}^Q) > \mathsf{SADD}(\mathcal{S}_A^{r_A})$ for all $1 < \gamma < \gamma_0$. Therefore, the SRP procedure is suboptimal.

For another analogous result see Tartakovsky and Polunchenko [169].

4. Asymptotic Optimality Properties of the Generalized Shiryaev–Roberts Procedures

This section is a logical continuation of the earlier work on the SR–r procedure. For an extended version of the material presented in this section see Tartakovsky, Pollak, and Polunchenko [175]. Specifically, our intent is to gain a theoretical insight into how the SR rule, the SR–r procedure, and the SRP scheme compare against one another performance-wise. Specifically, we ask and answer the following questions:

1. Is the stationary expected delay of the repeated SR procedure similar to $\lim_{\nu\to\infty}\mathsf{ADD}_\nu(\mathcal{S}_A)$? (Yes, see Theorem 4.2, Theorem 4.3 and Corollary 4.1.)
2. What can be said about the maximal expected detection delays of the SR, SR–r, and SRP procedures? (The SRP procedure and the SR–r procedure with a specially designed r are third-order asymptotically minimax, i.e., to within a negligible term $o(1) \to 0$. See Theorem 4.4. This answer justifies the conjecture of Moustakides, Polunchenko, and Tartakovsky [113].)
3. What can be said about $\lim_{\nu\to\infty}\mathsf{ADD}_\nu(\mathcal{S}_A)$, $\lim_{\nu\to\infty}\mathsf{ADD}_\nu(\mathcal{S}_A^r)$, and $\lim_{\nu\to\infty}\mathsf{ADD}_\nu(\mathcal{S}_A^Q)$ when all have the same ARL to false alarm $\gamma \geqslant 1$? (The ADD at infinity is the smallest for the original SR procedure, $\mathcal{S}_A$, but the difference between them is negligible as $\gamma\to\infty$. See Theorems 4.5 and 4.4.)

We will focus on $\mathsf{SADD}(T) = \sup_{0\leqslant\nu<\infty}\mathsf{ADD}_\nu(T)$ and $\mathsf{ADD}_\infty(T) = \lim_{\nu\to\infty}\mathsf{ADD}_\nu(T)$. We recall that $\mathsf{ADD}_\nu(T) = \mathbb{E}_\nu[T-\nu|T>\nu]$, $\nu \geqslant 0$.

It follows from Pollak [123] that the SRP procedure (8.3) is order-3 asymptotically optimal whenever $\mathbb{E}_0[|\log \Lambda_1|] < \infty$. We offer a proof of the order-3 asymptotic optimality property under the *stronger* second moment condition $\mathbb{E}_0[|\log \Lambda_1|^2] < \infty$ and using different techniques. The second moment condition allows one to obtain higher-order asymptotic approximations for $\mathsf{SADD}(\mathcal{S}_A^Q)$ and $\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T)$ (up to a vanishing term).

More importantly, using the ideas of Moustakides, Polunchenko, and Tartakovsky [113] one is able to design the initialization point $r = r(\gamma)$ in the SR–r procedure so that it is *also* order-3 asymptotically optimal. In this respect, $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$ plays a critical role. To understand why, let us look at Figure 8.2 which shows $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$ versus ν for several initialization values r. This figure was obtained using integral equations and numerical techniques developed by Moustakides, Polunchenko, and Tartakovsky [113]. If $r = 0$, this is the classical SR procedure whose average detection delay is monotonically decreasing to its minimum that is attained at infinity (a steady state value). It is seen that there exists a value $r = r_A^*$ that depends on the threshold A for which the worst point ν is at infinity, i.e., $\mathsf{SADD}(\mathcal{S}_A^{r_A^*}) = \mathsf{ADD}_\infty(\mathcal{S}_A^{r_A^*})$. The value of r_A^* is the minimal value for which this happens and it is also the value that delivers the minimum to the difference between $\mathsf{SADD}(\mathcal{S}_A^{r_A^*})$ and the lower bound for $\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T)$ derived by Moustakides, Polunchenko, and Tartakovsky [113] and Polunchenko and Tartakovsky [128]. This is a very important observation, since it allows us to build a proof of asymptotic optimality based on an estimate of $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$.

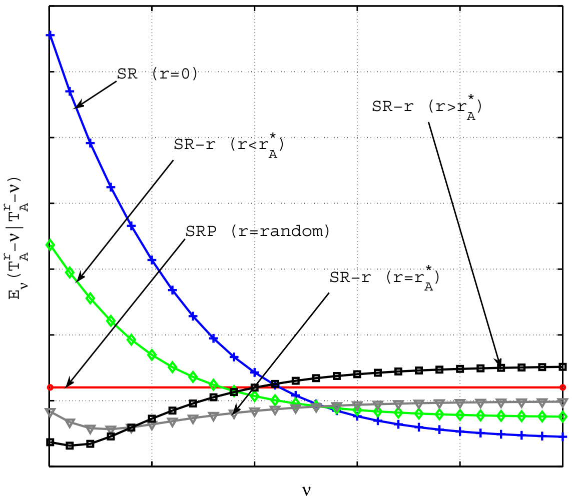


Figure 8.2: Typical behavior of $\mathsf{ADD}_\nu(T)$ as a function of changepoint ν for various initialization strategies.

The monotonicity of the curve for the ADD of the SR procedure allows us also to conclude (intuitively only since this is only a numerical observation and there is no theoretical justification of monotonicity) that the asymptotic lower bound for $\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T)$ can be evaluated based on the value of $\mathsf{ADD}_\infty(\mathcal{S}_A)$. Asymptotically, $\mathsf{ADD}_0(\mathcal{S}_A^Q)$, $\mathsf{ADD}_\infty(\mathcal{S}_A^{r_A^*})$, and $\mathsf{ADD}_\infty(\mathcal{S}_A)$ are the same, since the mean of the quasi-stationary distribution is of order $O(\log A)$ and $r_A^* \to r^*$ as $A \to \infty$, where r^* is a fixed positive number.

4.1. Two Useful Lemmas

From now on, let R_∞ be a random variable whose distribution is given by the cdf

$$\mathbb{P}(R_\infty \leqslant x) = \lim_{n \to \infty} \mathbb{P}_\infty(R_n^r \leqslant x) := Q_{\mathrm{ST}}(x),$$

where $Q_{\mathrm{ST}}(x)$ is the stationary distribution of R_n^r. Also, hereafter assume that $Q_A(x)$ (the quasi-stationary distribution) and $Q_{\mathrm{ST}}(x)$ both exist; note that this is the case when Λ_1 is continuous.

Lemma 4.1. *For any* $r \geqslant 0$, $\lim_{n,A \to \infty} \mathbb{P}_\infty(R_n^r \leqslant x | \mathcal{S}_A^r > n) = Q_{\mathrm{ST}}(x)$ *at all continuity points of* $Q_{\mathrm{ST}}(x)$.

Lemma 4.2. *The mean of the quasi-stationary distribution,*

$$\mu_A = \int_0^A x \, dQ_A(x),$$

is upper-bounded by $O(\log A)$*, i.e.,* $\mu_A \leqslant O(\log A)$*, as* $A \to \infty$*, where* $O(\log A)/\log A$ *is bounded, as* $A \to \infty$.

4.2. Average Run Length to False Alarm

We now present asymptotic approximations for the ARL to false alarm of the SR–r procedure, $\mathcal{S}_A^r$, and for that of the SRP procedure, $\mathcal{S}_A^Q$. Hereafter, let $Z_i = \log \Lambda_i$ denote the log-LR for the i-th observation and define $S_n = Z_1 + \cdots + Z_n$. Introduce a one-sided stopping time $\tau_a = \inf\{n \geqslant 1\colon S_n \geqslant a\}$, $a > 0$. Let $\kappa_a = S_{\tau_a} - a$ be the overshoot (excess over the level a at stopping), and let

$$\zeta = \lim_{a\to\infty} \mathbb{E}_0[e^{-\kappa_a}] \text{ and } \varkappa = \lim_{a\to\infty} \mathbb{E}_0[\kappa_a], \tag{8.13}$$

be the limiting exponential overshoot and the limiting overshoot, respectively. Either is a constant determined by the model, and can be computed numerically; in general, $0 < \zeta < 1$ and $\varkappa > 0$.

Theorem 4.1. *Provided $r_A = o(A)$, and assuming $A \to \infty$, it is true that $\mathsf{ARL}(\mathcal{S}_A^r) = (A/\zeta)[1 + o(1)]$ uniformly in $0 \leqslant r \leqslant r_A$ and $\mathsf{ARL}(\mathcal{S}_A^Q) = (A/\zeta)[1 + o(1)]$.*

For practical purposes we recommend to use

$$\mathsf{ARL}(\mathcal{S}_A^r) \approx A/\zeta - r \text{ and } \mathsf{ARL}(\mathcal{S}_A^Q) \approx A/\zeta - \mu_A. \tag{8.14}$$

4.3. Average Delay to Detection and Asymptotic Optimality

We now proceed to obtaining asymptotic approximations for $\mathsf{ADD}_\nu(\mathcal{S}_A^r)$, $\nu \geqslant 0$, including $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$. To judge whether the SR–r procedure with a certain head start, $r = r_A$, is asymptotically order-3 optimal, we will also derive an asymptotic lower bound for $\inf_{T\in\Delta(\gamma)} \mathsf{SADD}(T)$.

From now on, let $V_\infty = \sum_{j=1}^\infty e^{-S_j}$ and let $I = \mathbb{E}_0[Z_1]$ denote the Kullback-Leibler information number. Also, let $S_n^j = \sum_{i=j}^n Z_i$.

Lemma 4.3. *Let $\mathbb{E}_0[|Z_1|^2] < \infty$ and assume that Z_1 is non-arithmetic. Let $0 < N_A < A$ be such that $N_A/(A^{1-\delta} \log A) \to \infty$ and $N_A = o(A/\log A)$ as $A \to \infty$ for some $\delta \in (0, 1)$. Let $r \geqslant 0$. Then, as $A \to \infty$,*

$$\begin{aligned}\mathbb{E}_\nu[\mathcal{S}_A^r - \nu | \mathcal{S}_A^r > \nu, R_\nu^r] = \frac{1}{I}\Bigg\{ & \log A + \varkappa - \log(1 + R_\nu^r) \\ & - \mathbb{E}_\nu\left[\log\left(1 + \frac{V_\infty}{1 + R_\nu^r}\right) \middle| \mathcal{S}_A^r > \nu, R_\nu^r\right]\Bigg\} + o(1),\end{aligned} \tag{8.15}$$

where $o(1) \to 0$ as $A \to \infty$ uniformly on $\{N_A \leqslant \nu < \infty, R_\nu^r < A/N_A, 0 \leqslant r < \infty\}$.

Let

$$C_\infty = \mathbb{E}[\log(1 + R_\infty + V_\infty)] = \int_0^\infty \int_0^\infty \log(1 + x + y)\, dQ_{\mathrm{ST}}(x)\, d\tilde{Q}(y), \tag{8.16}$$

where $\tilde{Q}(y) = \mathbb{P}_0(V_\infty \leqslant y)$.

The following theorem provides asymptotic approximations for $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$ and for $\mathsf{ADD}_0(\mathcal{S}_A^Q)$.

Theorem 4.2. *If $\mathbb{E}_0[|Z_1|^2] < \infty$ and Z_1 is non-arithmetic, then for any $r \geqslant 0$ and as $A \to \infty$,*

$$\mathsf{ADD}_\infty(\mathcal{S}_A^r) = \frac{1}{I}(\log A + \varkappa - C_\infty) + o(1) \text{ and } \mathsf{ADD}_0(\mathcal{S}_A^Q) = \frac{1}{I}(\log A + \varkappa - C_\infty) + o(1).$$

Let

$$\mathcal{J}(T) = \frac{\sum_{\nu=0}^\infty \mathsf{ADD}_\nu(T)\mathbb{P}_\infty(T > \nu)}{\mathsf{ARL}(T)}. \tag{8.17}$$

The following lemma proposes a lower bound for $\mathsf{SADD}(T)$ in the class $\Delta(\gamma)$, $\gamma > 1$. This bound will be used to obtain an asymptotic lower bound in Theorem 4.3 and to prove order-3 asymptotic optimality of the detection procedures in Theorem 4.4.

Lemma 4.4. *Consider $\mathcal{S}_A$, and let $A = A_\gamma$ be chosen so that $\mathsf{ARL}(\mathcal{S}_A) = \gamma$ is true. Then the following lower bound holds:*

$$\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T) \geqslant \mathcal{J}(\mathcal{S}_A). \tag{8.18}$$

The following theorem provides the asymptotic approximation for the lower bound $\mathcal{J}(\mathcal{T}_A)$.

Theorem 4.3. *Let $\mathcal{J}(T)$ be defined as in (8.17), and C_∞ be as in (8.16). If $\mathbb{E}_0[|Z_1|^2] < \infty$ and Z_1 is non-arithmetic, then*

$$\mathcal{J}(\mathcal{S}_A) = \frac{1}{I}(\log A + \varkappa - C_\infty) + o(1), \quad \textit{as } A \to \infty.$$

The following corollary is a direct consequence of Theorems 4.2 and 4.3.

Corollary 4.1. *If Z_1 is non-arithmetic and $\mathbb{E}_0[|Z_1|^2] < \infty$, then $\mathsf{ADD}_\infty(\mathcal{S}_A) = \mathsf{STADD}(\mathcal{S}_A) + o(1)$, as $A \to \infty$, and*

$$\mathsf{STADD}(\mathcal{S}_A) = \frac{1}{I}(\log A + \varkappa - C_\infty) + o(1), \quad \textit{as } A \to \infty.$$

The following theorem establishes asymptotic optimality of the SRP and SR–r detection procedures under moderate conditions. Its proof is immediate from the above results.

Theorem 4.4. *Let $\mathbb{E}_0[|Z_1|^2] < \infty$ and let Z_1 be non-arithmetic.*

(i) Then

$$\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T) \geqslant \frac{1}{I}[\log(\gamma\zeta) + \varkappa - C_\infty] + o(1), \quad \textit{as } \gamma \to \infty. \tag{8.19}$$

(ii) If in the SRP procedure $A = A_\gamma = \gamma\zeta$, then $\mathsf{ARL}(\mathcal{S}_A^Q) = \gamma[1 + o(1)]$ and

$$\mathsf{SADD}(\mathcal{S}_A^Q) = \frac{1}{I}[\log(\gamma\zeta) + \varkappa - C_\infty] + o(1), \quad \textit{as } \gamma \to \infty. \tag{8.20}$$

Therefore, the SRP procedure is asymptotically order-3 optimal in the class $\Delta(\gamma)$.

(iii) If in the SR–r procedure $A = A_\gamma = \gamma\zeta$, and the initialization point $r = o(\gamma)$ is selected so that $\mathsf{SADD}(\mathcal{S}_A^r) = \mathsf{ADD}_\infty(\mathcal{S}_A^r)$, then $\mathsf{ARL}(\mathcal{S}_A^r) = \gamma[1 + o(1)]$ and

$$\mathsf{SADD}(\mathcal{S}_A^r) = \frac{1}{I}[\log(\gamma\zeta) + \varkappa - C_\infty] + o(1), \quad \textit{as } \gamma \to \infty. \tag{8.21}$$

Therefore, the SR–r procedure is asymptotically order-3 optimal.

Feasibility of selecting r_γ so that $\mathsf{SADD}(\mathcal{S}_A^r) = \mathsf{ADD}_\infty(\mathcal{S}_A^r)$ follows from numerical experiments performed by Moustakides, Polunchenko, and Tartakovsky [113] as well as from the example below.

Remark 4.1. The argument similar to the proof of Theorem 4.2 can be used in order to show that for the SR procedure

$$\mathsf{SADD}(\mathcal{S}_A) = \mathsf{ADD}_0(\mathcal{S}_A) = \frac{1}{I}(\log A + \varkappa - C_0) + o(1), \quad \text{as } A \to \infty,$$

where $C_0 = \mathbb{E}_0[\log(1 + V_\infty)]$. Since $A = \zeta\gamma$ implies $\mathsf{ARL}(\mathcal{S}_A) = \gamma[1 + o(1)]$, it follows that with this choice of threshold

$$\mathsf{SADD}(\mathcal{S}_A) = \mathsf{ADD}_0(\mathcal{S}_A) = \frac{1}{I}[\log(\zeta\gamma) + \varkappa - C_0] + o(1), \quad \text{as } \gamma \to \infty. \tag{8.22}$$

Comparing (8.22) with the lower bound (8.19) shows that $\mathsf{SADD}(\mathcal{S}_A) - \inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T) = O(1)$, as $\gamma \to \infty$. Thus, the SR procedure is only second-order asymptotically optimal and the difference is approximately equal to $(C_\infty - C_0)/I$. This difference can be quite large when detecting small changes (i.e., when I is small).

An interesting question is how $\mathsf{ADD}_\infty(\mathcal{S}_A)$, $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$, and $\mathsf{ADD}_\infty(\mathcal{S}_A^Q)$ are related when all three procedures have the same ARL to false alarm γ? The answer is that $\mathsf{ADD}_\infty(\mathcal{S}_A)$ is the smallest, as established by Theorem 4.5 below. Note also that by Theorems 4.2 and 4.4 the difference between $\mathsf{ADD}_\infty(\mathcal{S}_A)$, $\mathsf{ADD}_\infty(\mathcal{S}_A^r)$ and $\mathsf{ADD}_\infty(\mathcal{S}_A^Q)$ vanishes as $\gamma \to \infty$.

This result can be proved thus: 1) To show that $\mathsf{ARL}(\mathcal{S}_A^Q)$ is increasing in A (the fact that the ARL to false alarm of the SR$-r$ procedure is increasing in A for a fixed r is obvious); and 2) To show that $\mathsf{ADD}_\infty(\mathcal{S}_A^Q)$ is increasing in A (obviously, the ADD's at infinity are the same for all three procedures, assuming the same threshold for all three). Since the SR rule, $\mathcal{S}_A$, requires the lowest threshold to attain the same ARL to false alarm, $\mathsf{ADD}_\infty(\mathcal{S}_A)$ is the lowest. Step 1 can be performed for the most general model, i.e., we can prove that $\mathsf{ARL}(\mathcal{S}_A^Q)$ is increasing in A in the general case. However, while we believe that $\mathsf{ADD}_\infty(\mathcal{S}_A^Q)$ is also increasing in A in the general case, we are able to prove this fact only for the exponential family with a certain log-concavity property, which guarantees monotonicity properties of the Markov detection statistics.

For $\eta > 0$, regard the sequence defined by the recursion

$$R_{n+1}^{(\eta)} = \left(\eta + R_n^{(\eta)}\right)\Lambda_{n+1}, \quad n \geqslant 0, \quad R_0^{(\eta)} = r. \tag{8.23}$$

To prove the required result we need the following lemma.

Lemma 4.5. *Let $f_\theta(x) = \exp\{\theta x - \psi(\theta)\}$ be a density (with respect to some sigma-finite measure) where without loss of generality $\psi(0) = \psi'(0) = 0$, and suppose that the corresponding distribution function $F_{\theta=0}(x)$ is log-concave (i.e., $\log F_0(x)$ is a concave function). Suppose that $g(x) = f_\theta(x)$ for some $\theta > 0$ and that $f(x) = f_{\theta=0}(x)$, so that $\Lambda_i = e^{\theta X_i - \psi(\theta)}$. Then the process $(M_n)_{n\geqslant 0}$ that has transition probabilities*

$$\mathbb{P}(M_{n+1} \leqslant x | M_n = t) = \mathbb{P}_\infty\left(R_{n+1}^{(\eta)} \leqslant x | R_n^{(\eta)} = t, R_{n+1}^{(\eta)} < A\right)$$

is a stochastically monotone Markov process, i.e., $\mathbb{P}(M_{n+1} > x | M_n = t)$ is non-decreasing and right-continuous in t for all x.

Remark 4.2. Note that the Gaussian distribution is log-concave. Note also that the main issue is log-concavity, but not that g, f belong to an exponential family, since "most" pairs g, f can be embedded into an exponential family via

$$f_\theta(x) = \frac{(f(x))^{1-\theta}(g(x))^\theta}{\int (f(t))^{1-\theta}(g(t))^\theta\, dt} \overset{\text{def}}{=} e^{\theta\,\Lambda(x) - \psi(\theta)} f(x),$$

and that without loss of generality one can assume that the observations themselves have been transformed into Λ_i (the likelihood ratios of X_i and of Λ_i are the same, and by translation one can obtain $\psi(0) = \psi'(0) = 0$).

We can now proceed to stating the desired result.

Theorem 4.5. *Assume the exponential family and log-concavity conditions of Lemma 4.5. Let $0 < \gamma < \infty$ be fixed, and let A_γ^r be such that the ARL to false alarm of the SR–r procedure $T_{A_\gamma^r}^r = \inf\{n \geqslant 1 \colon R_n^r \geqslant A_\gamma^r\}$ is γ. Then $\mathsf{ADD}_\infty(T_{A_\gamma^r}^r)$ is an increasing function of r and*

$$\min_{0 \leqslant r < A_\gamma^r} \mathsf{ADD}_\infty(\mathcal{S}_{A_\gamma^r}^r) = \mathsf{ADD}_\infty(\mathcal{S}_{A_\gamma^0}^0) < \mathsf{ADD}_\infty(\mathcal{S}_{A_Q}^Q),$$

where A_Q is such that $\mathsf{ARL}(\mathcal{S}_{A_Q}^Q) = \gamma$.

4.4. Computing Constants C_0 and C_∞

In order to implement the asymptotic approximations we have to be able to compute the constants C_0 and C_∞. To compute C_0 and C_∞ we need to find $q_{\mathrm{ST}}(x) = dQ_{\mathrm{ST}}(x)/dx$ and $q_0(x) = dQ_0(x)/dx$. These pdf-s can be found from the equations

$$q_{\mathrm{ST}}(x) = \int_0^\infty q_{\mathrm{ST}}(y)\left[\frac{\partial}{\partial x} P_\infty^\Lambda\left(\frac{x}{1+y}\right)\right] dy;\;\; q_0(x) = -\int_0^\infty q_0(y)\left[\frac{\partial}{\partial x} P_0^\Lambda\left(\frac{1+y}{x}\right)\right] dy,$$

where $P_d^\Lambda(t) = \mathbb{P}_\infty(\Lambda_1 \leqslant t)$, $d = \{0, \infty\}$. Both C_0 and C_∞ can then be found, e.g., numerically. The next subsection offers a comparative performance analysis for an example where C_0 and C_∞ are computable analytically.

4.5. Accuracy of Asymptotic Approximations: An Example

To test the accuracy of the proposed asymptotic approximations, we carried out an extensive performance evaluation of the procedures discussed in the earlier sections for the following example. Suppose $\{X_n\}_{n\geqslant 1}$ is a series of independent observations such that $X_1, X_2, \dots, X_\nu$ are $\mathsf{beta}(2,1)$ each, and $X_{\nu+1}, X_{\nu+2}, \dots$ are $\mathsf{beta}(1,2)$ each. Put another way, the series undergoes a sudden and abrupt shift in the expected value from $2/3$ pre-change to $1/3$ post-change, while retaining the variance. The pre- and post-change probability densities for this scenario are $f(x) = 2x\mathbb{1}_{\{0\leqslant x\leqslant 1\}}$ and $g(x) = 2(1-x)\mathbb{1}_{\{0\leqslant x\leqslant 1\}}$

To be specific, our goal is to verify the conditions and the accuracy of the asymptotic approximations stated in Theorem 4.4, i.e.,

$$\mathsf{SADD}(\mathcal{S}_A^r) \approx \mathsf{SADD}(\mathcal{S}_A^Q) \approx \frac{1}{I}(\log A + \varkappa - C_\infty) \;\text{ and }\; \mathsf{SADD}(\mathcal{S}_A) \approx \frac{1}{I}(\log A + \varkappa - C_0), \qquad (8.24)$$

and also the approximations (8.14).

To undertake this task, it is necessary to know C_0, C_∞, ζ, $\varkappa$, r_A and μ_A. It is rare that C_0 and C_∞ can be found analytically, yet the $\mathsf{beta}(2,1)$-to-$\mathsf{beta}(1,2)$ model at hand they can: $C_0 = 1$ and $C_\infty = \pi^2/6 \approx 1.6449$. Also, note that $I = 1$. Thus, $\mathsf{SADD}(\mathcal{S}_A^r) \approx \mathsf{SADD}(\mathcal{S}_A^Q) \approx \log A + \varkappa - 1.6449$ and $\mathsf{SADD}(\mathcal{S}_A) \approx \log A + \varkappa - 1$.

Unfortunately, neither $\varkappa$ nor ζ are computable exactly. Monte Carlo simulations with 10^6 trials have been performed to estimate the two as $\varkappa \approx 1.255$ and $\zeta \approx 0.426$ with the standard error less than 10^{-3}. Specifically, these estimates were obtained from the formulas

$$\varkappa = \frac{\mathbb{E}_0[S_1^2]}{2\,\mathbb{E}_0[S_1]} - \sum_{k=1}^\infty \frac{1}{k}\,\mathbb{E}_0[S_k^-] \;\text{ and }\; \zeta = \frac{1}{I}\exp\left\{-2\sum_{k=1}^\infty \frac{1}{k}\Big(\mathbb{P}_0(S_k \leqslant 0) + \mathbb{P}_\infty(S_k > 0)\Big)\right\},$$

where $x^- = \min(0, x)$; see, e.g., Woodroofe [185].

Though we can find C_0 and C_∞, neither the quasi-stationary distribution, required for the SRP procedure, nor $\mathsf{ADD}_\nu(T)$ for $\nu \geqslant 0$ and the ARL to false alarm seem feasible to get analytically. To overcome this difficulty, these quantities were computed numerically, using the approach of Moustakides, Polunchenko, and Tartakovsky [113] with the number of breakpoints set at 3×10^4, high enough to ensure the relative error in the order of a fraction of a percent.

At this point the only unresolved question is that of how to choose r. Several options have been proposed in Moustakides, Polunchenko, and Tartakovsky [113], one of which is to set $r = \mu_A$. Recall that Theorem 4.4 requires (a) $r = o(A)$ as $A \to \infty$ and (b) $\mathsf{SADD}(\mathcal{S}_A^r) = \mathsf{ADD}_\infty(\mathcal{S}_A^r)$. With this choice, the condition (a) is satisfied since according to Lemma 4.2 $\mu_A \leqslant O(\log A)$.

Condition (b) is also satisfied even for small values of the ARL to false alarm. This can be seen from Figure 8.3, which shows how $\mathsf{ADD}_\nu(T)$ evolves as ν runs from 0 to 20 for the SRP test and for the SR-r procedure with $r = \mu_A$. The ARL to false alarm is about 50 for both procedures. Observe that the SR–r rule attains supremum at $\nu \to \infty$. Also, the stationary regime kicks in as early as at $\nu = 6$, and this is for

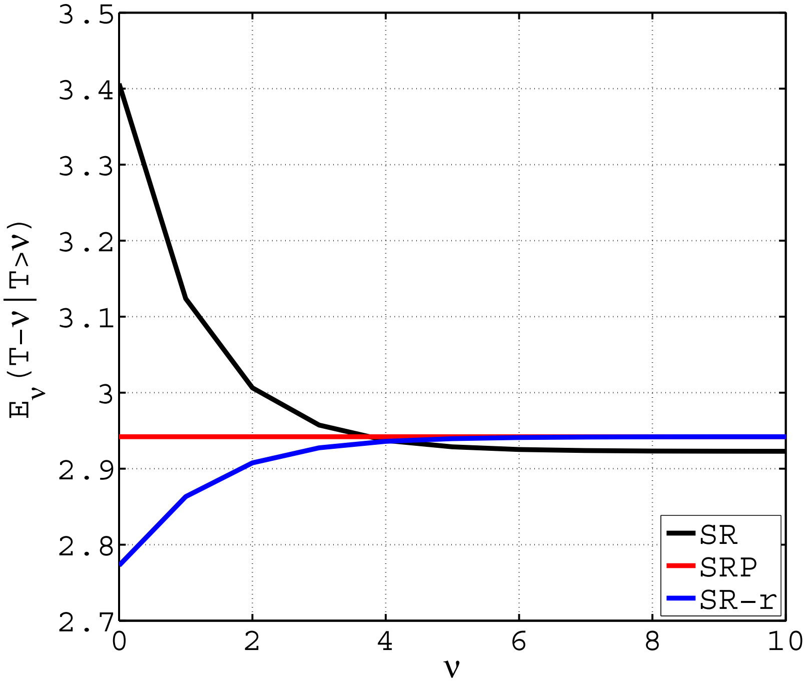


Figure 8.3: Results of numerical evaluation of the conditional average detection delay vs. changepoint ν of the SR, SRP and SR–r ($r = \mu_A$) procedures for the $\mathsf{beta}(2, 1)$-to-$\mathsf{beta}(1, 2)$ model.

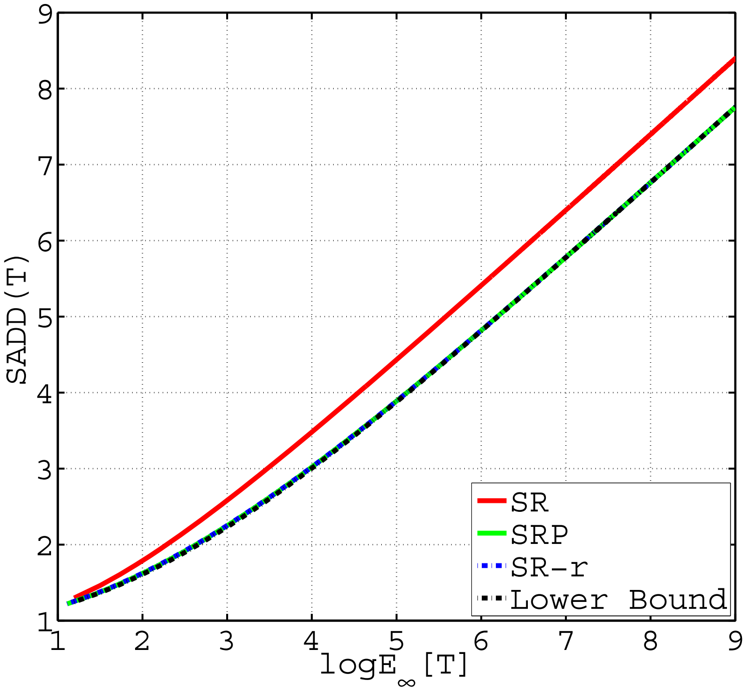


Figure 8.4: Results of numerical evaluation of operating characteristics of the SR, SRP and SR–r ($r = \mu_A$) procedures for the $\mathsf{beta}(2, 1)$-to-$\mathsf{beta}(1, 2)$ model.

$\mathsf{ARL}(T) \approx 50$. Figure 8.3 provides an illustration of Theorem 4.5 – $\mathsf{ADD}_\infty(T)$ is indeed the smallest for the SR procedure, while the difference is not substantial.

Shown in Figure 8.4 are the operating characteristics of the procedures of interest expressed via $\mathsf{SADD}(T)$ versus $\mathsf{ARL}(T)$, and the lower bound $\mathcal{J}(\mathcal{S}_A)$ – all plotted against $\log[\mathsf{ARL}(T)]$. The range of values of $\mathsf{ARL}(T)$ is from very low (as in under 10) to as high as 10^4. The log scale is particularly convenient in this case because the Kullback-Leibler information number is 1, and from the asymptotic expansions it follows that $\mathsf{SADD}(T)$ with respect to $\log[\mathsf{ARL}(T)]$ should be straight diagonal lines with unit slope. Such an expected form of dependence is indeed confirmed for $\mathsf{ARL}(T)$ above roughly 100, i.e., the point at which the asymptotics kick in. When $\mathsf{ARL}(T) < 100$, a slight deviation from the liner curve is observed. It is also seen that the performance of the SRP rule and that of the SR-r procedure with $r = \mu_A$ hardly exhibit any difference.

To better illustrate the performance difference, Table 4.5 provides a summary of selected values $\mathsf{SADD}(T)$ and $\mathcal{J}(\mathcal{S}_A)$. Also presented in parentheses are the corresponding theoretical predictions made based on the asymptotic approximations (8.24) and (9.4).

Table 8.2: Operating characteristics of the SR, SRP and SR–r procedures for the $\mathsf{beta}(2,1)$-to-$\mathsf{beta}(1,2)$ model. Numbers in parentheses are the corresponding theoretical values computed using the asymptotic approximations.

Test	γ	50	100	500	1000	10000
SR	A	21.0	42.0	212.0	424.5	4256.0
	$\mathsf{ARL}(T)$	50.412 (49.342)	99.832 (98.684)	499.866 (498.12)	999.797 (997.415)	9999.675 (10000.0)
	$\mathsf{SADD}(T)$	3.407 (3.312)	4.051 (4.005)	5.622 (5.615)	6.309 (6.308)	8.607 (8.611)
SRP	A	21.5	43.0	213.5	426.5	4259.0
	$\mathsf{ARL}(T)$	49.635 (48.48)	99.664 (98.431)	499.424 (497.595)	999.87 (997.404)	9999.81 (10000.066)
	$\mathsf{SADD}(T)$	2.942 (2.668)	3.534 (3.361)	5.021 (4.97)	5.692 (5.663)	7.965 (7.966)
SR-r	A	21.5	43.0	213.5	426.5	4259.0
	$r = \mu_A$	2.037	2.603	4.052	4.711	6.982
	$\mathsf{ARL}(T)$	49.554 (48.48)	99.582 (98.431)	500.52 (497.595)	999.792 (997.404)	9999.735 (10000.066)
	$\mathsf{SADD}(T)$	2.942 (2.668)	3.534 (3.361)	5.023 (4.97)	5.692 (5.663)	7.965 (7.966)
	Lower Bound	2.939 (2.668)	3.523 (3.361)	5.017 (4.97)	5.688 (5.663)	7.965 (7.966)

Chapter 9

Theoretical Results in Distributed Quickest Changepoint Detection

The work presented in this chapter is the result of collaboration of Dr. Tartakovsky (USC) and Dr. Veeravalli (Illinois).

1. The General Problem and Preliminaries

While the quickest changepoint detection problem has been studied for over fifty years, there has been little prior work on theoretical extensions to the distributed sensor setting and general stochastic models for observations. We developed novel procedures for change detection and isolation, to investigate their properties for general multipopulation and distributed stochastic models, as well as to provide an analytical framework to predict their performance in terms of the tradeoff between detection delay and frequency of false alarms.

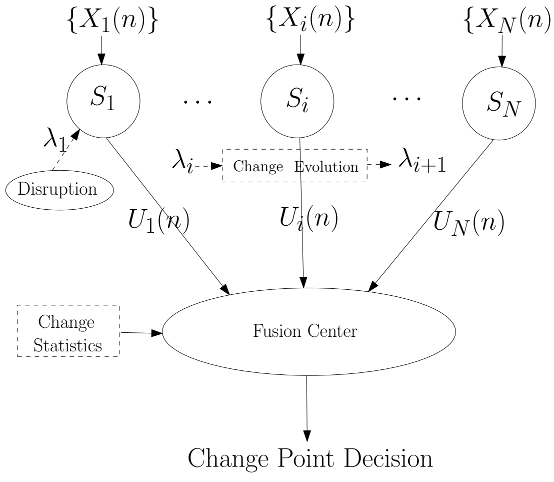


Figure 9.1: Change process detection in sensor network.

To address this goal, we have performed analysis of several generalizations of the change detection problem that arise in the applications to distributed sensor systems. Specifically, we consider the distributed multisensor system with N sensors, $S_1, \dots, S_N$, communicating with a fusion center, as shown in Figure 9.1. At time n, an observation $X_i(n)$ is made at sensor S_i. The changes in the statistical properties of the sequences $\{X_i(n)\}$ are governed by an event. We investigate a variety of models for the change process: only one (or a subset) of the sensors changes, they all change at the same time, or they change at different times. We also include various scenarios for communication with the fusion center, from the centralized one where the sensors send sufficient statistics, to the decentralized one where they send quantized observations or local decisions. We study the role of feedback from the fusion center, and investigate schemes for conserving energy at the sensors such as switching the sensors between on/off modes and censoring their observations. This concert of possibilities leads to a very interesting set of open problems that are discussed in the following sections. In order to address the wide range of potential applications of our theory, we accommodate general statistical models for the observations and allow for different degrees of model uncertainty.

2. A Distributed Scenario with no Feedback and Local or Full Memory

In the rest of this section, we will be interested in a particular distributed and decentralized multisensor scenario where the statistical properties of the sensors' observations change at the same unknown point in time λ, and no communication between sensors and no feedback between the fusion center and sensors are allowed, as shown in Figure 9.2. The goal is to detect this change as soon as possible, subject to false alarm constraints. Therefore, there is a distributed N-sensor system in which at time n one observes an N-component vector stochastic process $(X_1(n), \dots, X_N(n))$. The i-th component $X_i(n)$, $n = 1, 2, \dots$ corresponds to observations obtained from the sensor S_i. We will consider two approaches to the decentralized fusion problem. In the first case, the sensors quantize their observations and these quantized observations are sent to the fusion center. In this scenario there is only local memory, since quantization is performed based on current observations (i.e., no past observations participate in quantization). On the contrary, in the second scenario the sensors make local decisions based on all past observations (full memory). These decisions are then sent to the fusion center for making a final decision.

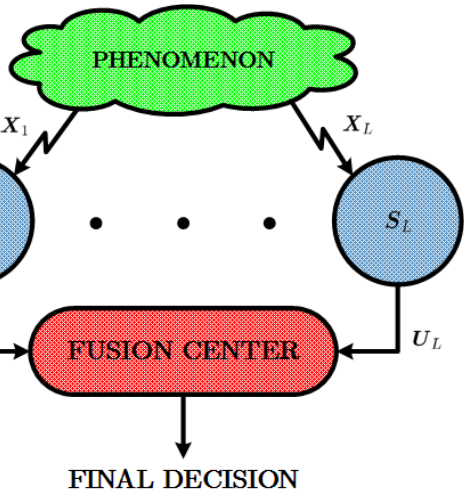


Figure 9.2: Change detection using distributed sensors.

At an unknown point in time ν ($\nu = 1, 2 \dots$) something happens and all of the components change their distribution. Conditioned on the change point, the observation sequences $\{X_1(n)\}$, $\{X_2(n)\}$, …, $\{X_N(n)\}$ are assumed to be mutually independent. Moreover, we assume that, in a particular sensor, the observations are iid before and after the change (with different distributions). If the change occurs at $\nu = k$, then in sensor S_i the data $X_i(1), \dots, X_i(\nu)$ follow density $f_i(x)$, while the data $X_i(\nu+1), X_i(\nu+2), \dots$ have the common distribution with density $g_i(x)$.

From now on, let

$$Z_i(n) = \log \frac{g_i(X_i(n))}{f_i(X_i(n))} \tag{9.1}$$

be the log-LR (LLR) between the "change" and "no-change" hypotheses for the n-th observation from the i-th sensor, and let $I_i = \mathbb{E}_1[Z_i(1)]$ be the Kullback-Leibler (K-L) information number between the densities $g_i(x)$ and $f_i(x)$.

The asymptotic performance of an optimal centralized detection procedure that has access to all data $\bar{b}X^n$ is given by

$$\inf_{T \in \Delta(\gamma)} \mathsf{SADD}(T) = \frac{\log \gamma}{I_{\text{tot}}}[1 + o(1)], \quad \gamma \to \infty, \tag{9.2}$$

where $I_{\text{tot}} = \sum_{i=1}^N I_i$. See, e.g., Basseville and Nikiforov [16], Siegmund [154], Tartakovsky [162]. This performance is attained for the centralized CUSUM and SR tests that use all available data.

3. Centralized CUSUM and Shiryaev–Roberts Detection Procedures for Known Parameter Values and Their Asymptotic Minimaxity Properties

Under the notation introduced above, the centralized CUSUM (detection) statistic is defined recursively as

$$W^c(n) = \max\left\{0, W^c(n-1) + \sum_{i=1}^N Z_i(n)\right\} \quad \text{for} \quad n \geqslant 1, \quad W^c(0) = 0, \tag{9.3}$$

and the centralized CUSUM (C-CUSUM) test is identified with the stopping time $T^c_{\text{CS}}(h) = \inf\{n \geqslant 1 \colon W^c(n) > h\}$, where $h > 0$ is a detection threshold which controls the FAR.

It follows from Basseville and Nikiforov [16], Lorden [95], Siegmund [154], Tartakovsky [162] that $\mathsf{ARL}(T^c_{\text{CS}}(h)) \geqslant e^h$ and, hence, $h = \log \gamma$ guarantees $\mathsf{ARL}(T^c_{\text{CS}}(h)) \geqslant \gamma$. Even though this choice of the threshold is usually conservative, it is useful as a preliminary estimate. Substantial improvements can be obtained using corrected Brownian motion approximations Siegmund [154] and the renewal argument Tartakovsky [165]. In particular, it follows from Tartakovsky [165] that, as $h \to \infty$,

$$\mathsf{ARL}(T^c_{\text{CS}}(h)) = \frac{e^h}{I^c_{tot}\zeta^2}[1 + o(1)], \tag{9.4}$$

where $I^c_{tot} = \sum_{l=1}^N I^c_l$ and ζ is a constant (depending on the model) subject to the renewal theory. Although this approximation is not especially accurate for high FAR, it is satisfactory already for moderate FAR; cf. Pollak and Tartakovsky [126].

If the threshold is selected using (9.4), i.e., $h_\gamma = \log(\gamma I^c_{tot}\zeta^2)$, then $\mathsf{ARL}(T^c_{\text{CS}}(h)) \sim \gamma$ and, as $\gamma \to \infty$,

$$\begin{aligned} \inf_{T \in \gamma} \mathsf{SADD}(T) &\geqslant \frac{\log \gamma}{I^c_{tot}}[1 + o(1)], \\ \mathsf{SADD}(T^c_{\text{CS}}(h_\gamma)) &= \mathbb{E}_1(T^c_{\text{CS}}(h_\gamma) - 1) = \frac{h_\gamma}{I^c_{tot}} + O(1), \end{aligned} \tag{9.5}$$

which means that the C-CUSUM test is asymptotically globally minimax-optimal in the distributed setting.

The centralized SR test (C-SR) is given by the stopping time

$$T^c_{\text{SR}}(A) = \min\left\{n \geqslant 1 \colon R^c(n) > A\right\}, \tag{9.6}$$

where the SR detection statistic obeys the recursion

$$R^c(0) = 0, \quad R^c(n) = (1 + R^c(n-1)) \exp\left\{\sum_{i=1}^N Z_i(n)\right\} \quad \text{for} \quad n \geqslant 1. \tag{9.7}$$

If $A = \gamma\zeta$, then $\mathsf{ARL}(T^c_{\text{SR}}(A)) \approx \gamma$ and, as $\gamma \to \infty$, the relations (9.5) hold true for the C-SR procedure, which means that this test is also asymptotically optimal in the minimax sense.

4. Analytical Techniques for Non-iid Cases

Much of the analysis of change point detection procedures has been restricted to the iid case. The optimality properties of the CUSUM and SR procedures depend crucially on the iid assumption. However recent work, including research by the PIs, has shown that more general models for the distributions can be handled in the asymptotic setting where $\gamma \to \infty$ Lai [83], Tartakovsky [163], Tartakovsky and Veeravalli [172].

In particular, it can be shown that the CUSUM and SR procedures are (first order) asymptotically optimal. Furthermore, these asymptotic optimality results extend to arbitrary moments of the delay Tartakovsky [163], Tartakovsky and Veeravalli [172].

Let

$$\ell^k_{n+k-1} = \sum_{j=k}^{n+k-1} \sum_{i=1}^N \log\left(\frac{g_i(X_i(j)|X_i(1), \dots, X_i(j-1))}{f_i(X_i(j)|X_i(1), \dots, X_i(j-1))}\right),$$

where in a general non-iid case the densities $g_i(X_i)$ and $f_i(X_i)$ are replaced with the corresponding conditional densities.

The asymptotic performance results of (9.5) in the i.i.d. case rely on the almost sure convergence of the normalized LLRs $n^{-1} \sum_{i=1}^N \sum_{j=k}^{n+k-1} Z_i(j)$ to the K-L information number I^c_{tot} as $n \to \infty$. This latter convergence is guaranteed by the Strong Law of Large Numbers as long as I^c_{tot} is finite. To generalize these results to the non-iid case, we need to *assume* the convergence of LLRs ℓ^k_{n+k-1} with some positive and finite number q:

$$\frac{\ell^k_{n+k-1}}{n} \xrightarrow[n\to\infty]{\mathbb{P}_k-\text{a.s.}} q \quad \text{for every } k < \infty. \tag{9.8}$$

Furthermore, we need the following condition on the rate of convergence:

$$\sum_{n=1}^{\infty} \mathbb{P}_k \left\{ | \, \ell^k_{n+k-1} - nq| > n\varepsilon \right\} < \infty \quad \text{for every } \varepsilon > 0. \tag{9.9}$$

The convergence implied by (9.8) and (9.9) is also called *complete convergence*, and can be written compactly as

$$\frac{\ell^k_{n+k-1}}{n} \xrightarrow[n\to\infty]{\mathbb{P}_k-\text{completely}} q \quad \text{for every } k < \infty. \tag{9.10}$$

Note that the quantity q plays the role of the total K-L distance between the "change" and "no change" hypotheses in the non-iid case. In particular, (9.5) can be extended to the non-iid case with I^c_{tot} replaced with q.

It can be shown that the complete convergence assumption is not restrictive and usually holds in practice, especially under Markov and hidden Markov models. Complete convergence allows us to extend most of the first order asymptotic approximations to non-iid cases, and is being used as a powerful tool in this project to obtain results for realistic models that arise in the applications to distributed sensor systems.

5. Multichart Centralized CUSUM and SR Procedures for Unknown Parameter Values

Consider a parametric model with pre-change and post-change densities $f_l^{\mu_l}(x)$ and $g_l^{\theta_l}(x)$, respectively. In many applications the pre-change parameter values μ_l can be estimated quite accurately in advance and, therefore, can be assumed to be known. However, the post-change parameters are seldom known in advance, and the putative θ_l is merely a representation of a meaningful change.

If the true post-change parameter value is not equal to the putative value θ_l, then the C-CUSUM and C-SR detection procedures that are tuned to θ_l are not optimal anymore. For the sake of simplicity consider a symmetric case where $\theta_l = \theta$ for all $l = 1, \dots, N$. In asymmetric case, the argument is essentially the same but the notation becomes cumbersome. Write

$$W_n(\theta) = \max_{1 \leqslant k \leqslant n} \sum_{j=k}^{n} \sum_{l=1}^{N} \log \frac{g_l^{\theta}(X_l(j))}{f_l^{\mu_l}(X_l(j))}, \quad R_n(\theta) = \sum_{k=1}^{n} \sum_{j=k}^{n} \sum_{l=1}^{N} \frac{g_l^{\theta}(X_l(j))}{f_l^{\mu_l}(X_l(j))}$$

for the CUSUM and SR statistics tuned to the value θ. There are several approaches for composite post-change hypotheses: (a) *A generalized* LR approach based on the generalized CUSUM statistic $\sup_\theta W_n(\theta)$ Dragalin [48], Lorden [95]; (b) *A mixture-based* CUSUM (or SR) statistic $\int W_n(\theta)\, d\Pi(\theta)$ averaged over a prior distribution $\Pi(\theta)$ Pollak [124]; and (c) *Adaptive* CUSUM and SR procedures where the parameter θ is replaced with one-stage delayed estimators Dragalin [47], Lorden and Pollak [96], Tartakovsky [163].

All the above methods have pros and cons. The generalized likelihood ratio approach is second-order optimal Dragalin [48] but computationally not feasible. The mixture-based approach is also second-order optimal Pollak [124], but may be difficult to implement since it is not always possible to find a conjugate prior to avoid computational problems. The approach of Lorden and Pollak [96] gives almost optimal performance, but also computationally demanding. The adaptive approaches of Dragalin [47] and Tartakovsky [163] are very simple in implementation (recursive) but are not second-order optimal – the performance degrades dramatically for detecting small changes.

For this reason, we propose to attack this problem from a different standpoint – using a multichart centralized detection test which will be referred to as M-C-CUSUM. Namely, in most applications it is usually possible to define an interval $[\underline{\theta}, \overline{\theta}]$ for the post-change parameter (either using some prior information or designing two values $\underline{\theta}$ and $\overline{\theta}$). For example, the point $\overline{\theta}$ can be selected in such a way that the average detection delay for the values $\theta \geqslant \overline{\theta}$ is small, so that further optimization is unnecessary. The value of $\underline{\theta}$, in turn, can be selected so that for smaller values it would be rather difficult to detect the change with a reasonable detection delay (indifference zone). Once the interval is set, $M \geqslant 2$ "reference" points are selected from that interval to run M C-CUSUM tests in parallel, each tuned to the respective point.

To be specific, let $\theta_m \in [\underline{\theta}, \overline{\theta}]$, $m = 1, \dots, M$ be a set of reference points, such that $\theta_m < \theta_{m+1}$. The M-C-CUSUM statistic is then defined as follows. Let $\ell_{l,m}(n) = \log[g_l^{\theta_m}(X_l(n))/f_l^{\mu_l}(X_l(n))]$ be the LLR tuned to θ_m. First, we define the C-CUSUM statistics for each of the reference points

$$W_m^c(n) = \max\left\{0, W_m^c(n) + \sum_{l=1}^{N} \ell_{l,m}(n)\right\}, \quad W_m^c(0) = 0$$

and the corresponding stopping times for each of the latter statistics $\tau_m^c(h_m) = \min\{n \geqslant 1 \colon W_m^c(n) > h_m\}$. The M-C-CUSUM stopping time is $\tau_{\mathrm{MC}}(\mathbf{h}) = \min_{1 \leqslant m \leqslant M} \tau_m^c(h_m)$, where $\mathbf{h} = (h_1, \dots, h_M)$, $h_m > 0$.

Clearly, the solution is not unique, since in general there are M different threshold values and only one constraint $\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h})) = \gamma$. Therefore, additional constraints are needed. We will use two approaches. In the first one, we use a common threshold by setting $h_m = h$ for all $m = 1, \dots, M$. Obviously, in this case the values of $\mathsf{ARL}(\tau_m^c(h))$ are different for different m. In the second one, we balance the values of $\mathsf{ARL}(\tau_m^c(h_m))$, in which case the thresholds are found from the equations

$$\mathsf{ARL}(\tau_m^c(h_m)) = M\gamma, \quad m = 1, \dots, M. \tag{9.11}$$

In fact, it is possible to show that h_m can be selected so that equations (9.11) hold for sufficiently large γ, and this approximation is asymptotically accurate as $\gamma \to \infty$. In this balanced case, the following approximate equality for $\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h}))$ holds

$$\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h})) \approx \left(\sum_{m=1}^{M} e^{-h_m} I_m \zeta_m^2\right)^{-1},$$

where $I_m = \sum_{l=1}^{N} \mathbb{E}_1^{\theta_m} \ell_{l,m}(1)$ is the corresponding K-L divergence and $0 < \zeta_m < 1$ is a computable constant related to the limiting "exponential overshoot" in the one-sided test, which is subject of a renewal-theoretic argument.

The following theorem establishes asymptotic performance of the M-C-CUSUM detection procedure in these two scenarios. We will need the following additional notation: $\mathcal{J}_m(\theta) = \sum_{l=1}^{N} \mathbb{E}_1^{\theta} \ell_{l,m}(1)$, $\mathsf{SADD}_\theta(\tau_{\mathrm{C}}(h)) = \sup_k \mathbb{E}_k^\theta[\tau - k | \tau \geqslant k]$, where $\mathbb{E}_k^\theta$ is expectation when the post-change parameter value is θ.

Theorem 5.1. *Assume that $\mathcal{J}_m(\theta)$ is monotonically nondecreasing in θ for $\theta \geqslant \theta_m$ and $\mathcal{J}_m(\theta) < \infty$ for all θ.*

(i) For every $h_m > 0$,

$$\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h})) \geqslant \frac{1}{\sum_{m=1}^{M} e^{-h_m}}.$$

If, in addition, the LLR is non-arithmetic, then, as $\min_m h_m \to \infty$,

$$\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h})) = \frac{1 + o(1)}{\sum_{m=1}^{M} [e^{-h_m} I_m \zeta_m^2]}.$$

(ii) Let $h_m = h_m(\nu)$ be such that $\lim_{\nu \to \infty}(h_m/\nu) = 1$, $m = 1, \dots, M$ and let $\theta^* \in (-\infty, \underline{\theta})$ be such that $J_1(\theta^*) = 0$. Then for any $\theta > \theta^*$ as $\nu \to \infty$,

$$\mathsf{SADD}_\theta(\tau_{\mathrm{MC}}(\mathbf{h})) = \frac{\nu}{\max\limits_{1 \leqslant m \leqslant M} \mathcal{J}_m(\theta)}(1 + o(1)), \tag{9.12}$$

where $\mathcal{J}_m(\theta) = I_m$ for $\theta = \theta_m$.

(iii) Let $h_1 = \cdots = h_M = h$. If $h = h_\gamma = \log(\gamma \sum_{m=1}^{M} I_m v_m^2)$, then, as $\gamma \to \infty$,

$$\mathsf{ARL}(\tau_{\mathrm{MC}}(h_\gamma) = \gamma[1 + o(1)],$$

and, for all $m = 1, \dots, M$,

$$\inf_{\tau \in \mathbb{C}_\gamma} \mathsf{SADD}_{\theta_m}(\tau) \sim \mathsf{SADD}_{\theta_m}(\tau_{\mathrm{MC}}(h_\gamma)) \sim \frac{\log \gamma}{I_m}. \tag{9.13}$$

(iv) If $h_m = h_m(\gamma) = \log(\gamma M I_m \zeta_m^2)$, then, as $\gamma \to \infty$,

$$\mathsf{ARL}(\tau_m^c(h_m(\gamma))) = M\gamma[1 + o(1)], \; m = 1, \dots, M;$$
$$\mathsf{ARL}(\tau_{\mathrm{MC}}(\mathbf{h}(\gamma))) = \gamma[1 + o(1)],$$

and, for all $m = 1, \dots, M$, asymptotic relations (9.13) *hold.*

Note that Theorem 5.1 (iii) and (iv) imply that the M-C-CUSUM is asymptotically first-order optimal at the points $\theta_1, \dots, \theta_M$ in both considered scenarios. Conditions of the theorem hold for the exponential family of distributions.

Note also that similar results hold for the M-C-SR detection procedure (where the CUSUM statistics $W_m^c(n)$ are replaced with the SR statistics $R_m^c(n)$), i.e., it is also asymptotically minimax at the points $\theta_1, \dots, \theta_M$.

In Section 7, we consider a decentralized detection procedure that uses compressed data $(U_1(n), \dots, U_N(n))$ by quantizing the data at sensors, so that the required bandwidth for communication with the fusion center is minimal. In this context, the use of the multichart detection procedures is especially important, since it allows us to perform quantization in a (small) number of isolated points. It is interesting to investigate how the loss in information (caused by quantization) affects the efficiency of detection procedures. This problem will be addressed below in detail. The advantage of these tests is that they do not require any processing power at the sensors.

6. Decentralized Detection Based on Local Decisions at Sensors for Known Models

We now consider three decentralized detection schemes with full memory that perform local detection in the sensors and then transmit these local binary decisions to the fusion center for optimal combining and final decision-making. Obviously, these schemes require minimum bandwidth for communication with the fusion center. The abbreviation LD-CUSUM will be used for procedures that perform CUSUM tests in sensors and use local decisions.

6.1. Asymptotically Optimal Decentralized LD-CUSUM Test

Let

$$W_i(n) = \max\{0, W_i(n-1) + Z_i(n)\}, \quad W_i(0) = 0$$

be the CUSUM statistic in the i-th sensor, where $Z_i(n) = \log[f_1^{(i)}(X_i(n))/f_0^{(i)}(X_i(n))]$ is the LLR, and let

$$U_i(n) = \begin{cases} 1 & \text{if } W_i(n) \geqslant \omega_i h \\ 0 & \text{otherwise,} \end{cases}$$

where $\omega_i = I_i / I_{\mathrm{tot}} = I_i / \sum_{i=1}^{N} I_i$ $(I_i = \mathbb{E}_0[Z_i(1)])$ and h is a positive threshold.

The stopping time is defined as

$$T_{\mathrm{ld}}(h) = \min\{n \geqslant 1 \colon \min_{1 \leqslant i \leqslant N} [W_i(n)/\omega_i] \geqslant h\}. \tag{9.14}$$

In other words, binary local decisions (1 or 0) are transmitted to the fusion center, and the change is declared at the first time when $U_i(n) = 1$ for all sensors $i = 1, \dots, N$.

It follows from Mei [104] that if $\mathbb{E}_0[|Z_i(1)|^3] < \infty$, then $\mathsf{ARL}(T_{\rm ld}(h)) \geqslant e^h$ for every $h > 0$. Under an additional Cramér-type condition, it follows from Dragalin, Tartakovsky, and Veeravalli [49] that

$$\mathsf{SADD}(T_{\rm ld}(h)) = \frac{h}{I_{\rm tot}} + C_N\sqrt{\frac{h}{I_{\rm tot}}} + c + o(1) \quad \text{as } h \to \infty, \tag{9.15}$$

where c is a computable constant that depends on the model and

$$C_N = \mathbb{E} \max_{1 \leqslant i \leqslant N} \left\{ \frac{\hat{\sigma}_i}{I_i} Y_i \right\}, \tag{9.16}$$

$Y_1, \dots, Y_N$ are independent standard Gaussian random variables; $\hat{\sigma}_i = \sqrt{\mathrm{Var}_i(Z_1(i))}$; Var_i is the operator of variance under $f_1^{(i)}$.

Therefore, if $h = \log \gamma$, then

$$\inf_{\tau \in \mathbb{C}_\gamma} \mathsf{SADD}(\tau) \sim \mathsf{SADD}(T_{\rm ld}(h)) \sim \frac{\log \gamma}{I_{\rm tot}}, \quad \text{as } \gamma \to \infty, \tag{9.17}$$

and the detection test $T_{\rm ld}(h)$ is globally first-order asymptotically optimal (AO). Correspondingly, we will use the abbreviation AO-LD-CUSUM for this test in the rest of the report.

However, since the second term in the asymptotic approximation (9.15) is on the order of the square root of the threshold, it is expected that the convergence to the optimum is slow. Furthermore, the performance degradation compared to the optimal centralized test is expected to be more and more severe with growth of the number of sensors, since the constant C_N given by (9.16) increases with N. Note that for the optimal centralized CUSUM and SR tests and for the decentralized CUSUM and SR tests with binary quantization introduced below in Subsection 7 residual terms are constants. We therefore expect that for moderate false alarm rates typical for practical applications the procedures with quantization may perform better. In Subsection 9.1, this conjecture is verified for the Poisson model.

It is worth mentioning that the results similar to (9.15) and (9.17) are not available for LD-SR detection test (where local voting is done based on the SR statistics $R_i(n)$ in place of the CUSUM statistics $W_i(n)$) in the class $\mathbb{C}_\gamma$. It turns out that the renewal property of the CUSUM statistics $W_i(n)$ plays a crucial role under the ARL to false alarm constraint (as well as under the local PFA constraint). However, it follows from the work of Tartakovsky and Veeravalli [173] that the LD-SR detection test can be effectively constructed in a Bayesian setting.

6.2. Decentralized Minimal and Maximal LD-CUSUM and LD-SR Tests

Let $T_{\rm CS}^i(h) = \min\{n\colon W_i(n) \geqslant h\}$ denote the (local) stopping time of the CUSUM test in the i-th sensor. Introduce the stopping times

$$T_{\min}(h) = \min(T_{\rm CS}^1, \dots, T_{\rm CS}^N) \quad \text{and} \quad T_{\max}(h) = \max(T_{\rm CS}^1, \dots, T_{\rm CS}^N)$$

that will be referred to as minimal LD-CUSUM (Min-LD-CUSUM) and maximal LD-CUSUM (Max-LD-CUSUM) tests, respectively. Similarly, we may define Min-LD-SR and Max-LD-SR tests based on the local SR stopping times in sensors $T_{\rm SR}^i(h) = \min\{n\colon \log R_i(n) \geqslant h\}$. Below we focus on the CUSUM-based tests keeping in mind that the results hold for the SR-based tests as well.

Consider first the false alarm rate for these detection tests. Clearly, $\mathsf{ARL}(T_{\max}) \geqslant \mathsf{ARL}(T_{\rm CS}^i)$ for every $i = 1, \dots, N$. Since $\mathsf{ARL}(T_{\rm CS}^i) \geqslant e^h$, it follows that $\mathsf{ARL}(T_{\max}) \geqslant e^h$ for every $h > 0$. We now show that $\mathsf{ARL}(T_{\min}) \geqslant N^{-1}e^h$ for every $h > 0$. Indeed,

$$T_{\min} = \min\{n : \max_i W_i(n) \geqslant h\} \geqslant \min\{n : \max_i R_i(n) \geqslant e^h\} \geqslant \min\{n : G_N(n) \geqslant e^h/N\} = \eta,$$

where $G_N(n) = N^{-1}\sum_{i=1}^N R_i(n)$. Since $G_N(n) - n$ is a zero-mean $\mathbb{P}_\infty$-martingale and since $G_N(\eta) \geqslant e^h/N$, it follows from the optional sampling theorem that $\mathsf{ARL}(T_{\min}) \geqslant \mathsf{ARL}(\eta) = G_N(\eta) \geqslant e^h/N$.

However, these inequalities are usually very conservative. For large threshold values, asymptotically sharp approximations can be derived as follows. It follows from Pollak and Tartakovsky [126], Tartakovsky [165] that, as $h \to \infty$, under the no-change hypothesis, the stopping times T_{CS}^i, $i = 1, \dots, N$ are exponentially distributed with mean values $e^h/(\zeta_i^2 I_i)$, where ζ_i are constants that are defined by (9.24) replacing $S^q(\eta_h)$ by $\sum_{k=1}^{\eta_h} Z_i(k)$. These constants can be computed numerically for any particular model using renewal arguments. Therefore, for a large threshold, $T_{\min}(h)$ is approximately exponentially distributed with mean $\mathsf{ARL}(T_{\min}) \sim e^h/c_N$, where $c_N = \sum_{i=1}^N v_i^2 I_i$, while the mean of the stopping time $T_{\max}$ is

$$\mathsf{ARL}(T_{\max}) \sim e^h/c'_N \quad \text{as } h \to \infty,$$

where $c'_N < c_N$ can be easily computed for any N. In particular, for $N = 5$ and in the symmetric case, $c'_5 = (60/137)\zeta^2 I \approx 0.44\zeta^2 I$ and $c_5 = 5\zeta^2 I$.

In order to derive an asymptotic approximation for $\mathsf{SADD}(T_{\min})$, note that $\mathbb{E}_1\, T_{\min}(h) \leqslant \mathbb{E}_1\, T_{\text{CS}}^i(h)$ for all $i = 1, \dots, N$ and, hence,

$$\mathbb{E}_1\, T_{\min} \leqslant \frac{h}{\min_{1\leqslant i\leqslant N} I_i}(1 + o(1)), \quad \text{as } h \to \infty,$$

since $\mathbb{E}_1[T_{\text{CS}}^i(h)] \sim h/I_i$.

To derive an approximation for $\mathsf{SADD}(T_{\max})$, introduce the stopping time $\eta(h) = \min\{n \geqslant 1\colon\ \min_{1\leqslant i\leqslant N} W_i(n) \geqslant h\}$ and note that $\eta(h) \geqslant T_{\max}(h)$. Since $W_i(n) = \sum_{k=1}^n Z_i(k) - \min_{1\leqslant k\leqslant n} Z_i(k)$ and the second term is a slowly changing sequence, applying Theorem 2.3 of Tartakovsky [164] yields

$$\mathbb{E}_1\, \eta(h) \sim \frac{h}{\min_{1\leqslant i\leqslant N} I_i}, \quad \text{as } h \to \infty,$$

which implies that

$$\mathbb{E}_1[T_{\max}] \leqslant \frac{h}{\min_{1\leqslant i\leqslant N} I_i}(1 + o(1)), \quad \text{as } h \to \infty.$$

Therefore, taking thresholds $h = \log(\gamma c_N)$ in the Min-LD-CUSUM and $h = \log(\gamma c'_N)$ in the Max-LD-CUSUM, we obtain the bounds for tradeoff curves that relate the SADD and the ARL, as $\gamma \to \infty$:

$$\mathsf{SADD}(T_{\min}) \leqslant \frac{\log\gamma}{\max_{1\leqslant i\leqslant N} I_i}(1 + o(1)), \quad \mathsf{SADD}(T_{\max}) \leqslant \frac{\log\gamma}{\min_{1\leqslant i\leqslant N} I_i}(1 + o(1)).$$

Thus, in the symmetric case where $I_i = I$, the asymptotic relative efficiency of these detection tests compared to the optimal centralized test (defined as the ratio of the limiting values of SADDs, see (9.34) below) is $\text{ARE}(T_{\min}; \tau_c) \geqslant \text{ARE}(T_{\max}; \tau_c) \geqslant N$.

Note that while based on the first-order asymptotics it may be expected that in the symmetric case the Max-LD-CUSUM test may perform as well as the Min-LD-CUSUM test, Monte Carlo simulations in Section 9.1 show that the Min-LD-CUSUM test performs better even in the symmetric case. The same conclusion has been reached by Moustakides [112] based on the analysis of a 2-sensor continuous-time Brownian motion model.

7. Decentralized Detection Based on Quantization at Sensors for Known Models

Consider the scenario where based on the information available at sensor S_i at time n a message $U_i(n)$ belonging to a finite alphabet of size M_i (e.g., binary) is formed and sent to the fusion center (see Figure 9.2). Write $\mathbf{U}(n) = (U_1(n), \dots, U_N(n))$ for the vector of N messages at time n. Based on the sequence of sensor messages, a decision about the change is made at the fusion center. This test is identified with a stopping time on $\{\mathbf{U}(n)\}_{n\geqslant 1}$ at which it is declared that a change has occurred. The goal is to find tests based on $\{\mathbf{U}(n)\}_{n\geqslant 1}$ that optimize the tradeoff between detection delay and false alarm rate.

Various information structures are possible for the decentralized configuration depending on how feedback and local information is used at the sensors. Here we consider the simplest information structure where the message $U_i(n)$ formed by sensor S_i at time n is a function of only its current observation $X_i(n)$, i.e., $U_i(n) = \psi_{i,n}(X_i(n))$. Moreover, since for a particular sensor S_i, the sequence $\{X_i(n)\}_{n\geqslant 1}$ is assumed to be iid, it is natural to confine ourselves by *stationary* quantizers for which the quantizing functions $\psi_{i,n}$ do not depend on n, i.e., $\psi_{i,n} = \psi_i$ for all $n \geqslant 1$. The quantizing functions $\bar{b}\psi = \{\psi_i; i = 1, \dots, N\}$, together with the fusion center stopping time τ, form a policy $\phi = (\bar{b}\psi, \tau)$.

Let H_ν be the hypothesis that the change occurs at time $\nu \in \{1, 2, \dots\}$, and let H_∞ be the hypothesis that there is no change. Also, let $g_i^{(0)}$ and $g_i^{(1)}$ denote the probability mass function (pmf) induced on U_i when the observation $X_i(n)$ is distributed as f_i and g_i, respectively. Then, for *fixed* sensor quantizers, the LLR between the hypotheses H_k and H_∞ at the fusion center is given by

$$Z^q(k,n) = \sum_{j=k}^{n}\sum_{i=1}^{N} \log \frac{g_i^{(1)}(U_i(j))}{g_i^{(0)}(U_i(j))}. \tag{9.18}$$

Hereafter the superscript index q stands for quantized versions of the corresponding variables to distinguish from the centralized case where we used the superscript c. For fixed sensor quantizers, the fusion center faces a standard change detection problem based on the vector observation sequence $\{\mathbf{U}(n)\}$.

Here the goal is to choose the policy ϕ that minimizes $\mathsf{SADD}(\phi)$ defined by

$$\mathsf{SADD}(\phi) = \sup_{1\leqslant\nu<\infty} \mathbb{E}_\nu(\tau - \nu \mid \tau \geqslant \nu) \tag{9.19}$$

while maintaining the ARL to false alarm at a level not less than $\gamma > 1$.

We can define the CUSUM and SR statistics by $W^q(n) = \max_{0\leqslant\nu\leqslant n} Z^q(\nu, n)$ and $R^q(n) = \sum_{\nu=1}^{n} e^{Z^q(\nu,n)}$, respectively, which obey the recursions:

$$\begin{aligned} W^q(n) &= \max\{0, W^q(n-1) + Z^q(n,n)\}, \quad W^q(0) = 0; \\ R^q(n) &= [1 + R^q(n-1)]\exp\{Z^q(n,n)\}, \quad R^q(0) = 0. \end{aligned} \tag{9.20}$$

Then the CUSUM and SR detection procedures at the fusion center $T^q_{\text{CS}}(h)$ and $T^q_{\text{SR}}(a)$ are, respectively, given by

$$T^q_{\text{CS}}(h) = \min\{n \geqslant 1\colon W^q(n) \geqslant h\}, \quad T^q_{\text{SR}}(a) = \min\{n \geqslant 1\colon \log R^q(n) \geqslant a\}, \tag{9.21}$$

where h and a are positive thresholds which are selected so that $\mathsf{ARL}(T^q_{\text{CS}}) \geqslant \gamma$ and $\mathsf{ARL}(T^q_{\text{SR}}) \geqslant \gamma$.

Let $I_i^q = \mathbb{E}_1[g_i^{(1)}(U_i(1))/g_i^{(0)}(U_i(1))]$ denote the K-L information number for quantized data in the i-th sensor (i.e., divergence between $g_i^{(1)}$ and $g_i^{(0)}$), and let $I^q_{\text{tot}} = \sum_{i=1}^{N} I_i^q$ be the total K-L information accumulated from all sensors.

Similar to (9.2) we obtain that detection procedures $T^q_{\text{CS}}(h)$ and $T^q_{\text{SR}}(h)$ given in (9.21), with $h = a = \log\gamma$, are asymptotically minimax optimal as $\gamma \to \infty$ among all procedures with ARL to false alarm greater than γ (for fixed quantizers ψ_i). To be specific,

$$\inf_{\tau\in\mathbb{C}_\gamma} \mathsf{SADD}(\tau) \sim \mathsf{SADD}(T^q_{\text{CS}}) \sim \mathsf{SADD}(T^q_{\text{SR}}) \sim \frac{\log\gamma}{I^q_{\text{tot}}} \quad \text{as } \gamma \to \infty.$$

This result immediately reveals how to choose the sensor quantizers: It is asymptotically optimum (as $\gamma \to \infty$) for sensor S_i at time n to select ψ_i to maximize I_i^q, the K-L information number. By Tsitsiklis [176], an optimal ψ_i that maximizes I_i^q is a monotone likelihood ratio quantizer (MLRQ), i.e., there exist thresholds $a_1, a_2, \dots, a_{M_i-1}$ satisfying $-\infty < a_1 \leqslant a_2 \leqslant \cdots \leqslant a_{M_i-1}$ such that

$$\psi_{i,\text{opt}}(X_i) = b_i \;\text{ only if }\; a_{b_i} < Z_i(X_i) \leqslant a_{b_i+1}, \tag{9.22}$$

where $Z_i(X_i) = \log[g_i(X_i)/f_i(X_i)]$ is the LLR at the observation X_i (at sensor S_i). Note that function $\psi_{i,\text{opt}}$ is independent of n.

Thus, the asymptotically optimum policy ϕ_{opt} for a decentralized change detection problem consists of a stationary (in time) set of MLRQs at the sensors followed by CUSUM or SR procedures based on $\{\mathbf{U}(n)\}$ at the fusion center (as described in (9.21)).

For each i, we denote the corresponding pmfs induced on $U_i(n)$ by $g_{i,\text{opt}}^{(1)}$ and $g_{i,\text{opt}}^{(0)}$ (i.e., at the output of the stationary MLRQ $\psi_{i,\text{opt}}$ that maximizes I_i^q). Then the effective total K-L information number between the change and no-change hypotheses at the fusion center is given by

$$I_{\text{tot,opt}}^q = \sum_{i=1}^N I(g_{i,\text{opt}}^{(1)}, g_{i,\text{opt}}^{(0)}). \tag{9.23}$$

Further, we denote by T_{CS}^q and T_{SR}^q, respectively, the CUSUM and SR stopping rules at the fusion center for the case where the sensor quantizers are chosen to be $\bar{b}\psi_{\text{opt}} = \{\psi_{i,\text{opt}}\}$. Finally, we denote by $\phi_{\text{opt}} = (\bar{b}\psi_{\text{opt}}, T_{\text{CS}}^q)$ and $\hat{\phi}_{\text{opt}} = (\bar{b}\psi_{\text{opt}}, T_{\text{SR}}^q)$ the corresponding CUSUM and SR policies, respectively, with optimal quantization.

We also need the following additional notation:

$$S^q(n) = \sum_{k=1}^n \sum_{i=1}^N \log \frac{g_i^{(1)}(U_i(k))}{g_i^{(0)}(U_i(k))}, \quad S^q(0) = 0; \quad \eta_h = \min\{n \geqslant 1 \colon S^q(n) \geqslant h\};$$
$$\zeta_q = \lim_{h\to\infty} \mathbb{E}_1 \exp\{-(S^q(\eta_h) - h)\}, \tag{9.24}$$

where ζ_q can be computed using renewal-theoretic arguments.

7.1. Optimality Properties of CUSUM and SR Procedures

The asymptotic performance of the asymptotically optimum solutions to the decentralized change detection problem described above is given in the following theorem.

Theorem 7.1. *Suppose $I_{\text{tot,opt}}$ is positive and finite.*

(i) Then $h = a = \log\gamma$ implies that $\mathsf{ARL}(T_{\text{CS}}^q) \geqslant \mathsf{ARL}(T_{\text{SR}}^q) \geqslant \gamma$.

(ii) If, in addition, $Z^q(1,1)$ is non-arithmetic, then

$$\mathsf{ARL}(T_{\text{CS}}^q(h)) \sim \frac{e^h}{\zeta_q^2 I_{\text{tot,opt}}^q}, \quad \mathsf{ARL}(T_{\text{SR}}^q(a)) \sim e^a/\zeta_q \quad \text{as } h, a \to \infty; \tag{9.25}$$

(iii) If $a = h = \log\gamma$, then

$$\inf_{\phi\in\mathbb{C}_\gamma} \mathsf{SADD}(\phi) \sim \mathsf{SADD}(\phi_{\text{opt}}) \sim \mathsf{SADD}(\hat{\phi}_{\text{opt}}) \sim \frac{\log\gamma}{I_{\text{tot,opt}}^q} \quad \text{as } \gamma\to\infty. \tag{9.26}$$

If $h = h_\gamma = \log[\zeta_q^2 I_{\text{tot,opt}}^q \gamma]$ and $a = a_\gamma = \log(\zeta_q\gamma)$, then $\mathsf{ARL}(T_{\text{CS}}^q(h_\gamma)) \sim \gamma$ and $\mathsf{ARL}(T_{\text{SR}}^q(a_\gamma)) \sim \gamma$ as $\gamma\to\infty$ and asymptotic relations (9.26) *hold.*

7.2. Binary Quantization

We now continue by considering the simplest case where $U_i(n) = \psi_i(X_i(n))$ are the outputs of binary quantizers and specify previous results for this case. Also, in the rest of this section we will consider only the CUSUM detection procedure with understanding that analogous results hold for the SR procedure. It follows from Theorem 7.1 that the optimal binary quantizer is the MLRQ that is given by

$$U_i = \psi_i(X) = \begin{cases} 1 & \text{if } Z_i(X) = \log[g_i(X)/f_i(X)] \geqslant t_i, \\ 0 & \text{otherwise,} \end{cases} \tag{9.27}$$

where t_i is a threshold that maximizes the K-L information in the resulting Bernoulli sequence.

To be precise, for $l = 0, 1$, let $g_l^{(i)}$ denote the probability induced on $U_i(n)$ when the observation $X_i(n)$ is in the pre- and post-change modes. Let $\beta_{0,i} = g_0^{(i)}(U_i(j) = 1)$ and $\beta_i = g_1^{(i)}(U_i(j) = 1)$ denote the corresponding probabilities under the normal and the anomalous conditions, respectively. The resulting binary (Bernoulli) sequences $\{U_i(j), i = 1, \dots, N\}$, $j \geqslant 1$ are then used to form the binary CUSUM statistic similar to (9.20) as

$$W^b(n) = \max\{0, W^b(n-1) + \sum_{i=1}^{N} Z_i^b(n))\}, \quad W^b(0) = 0, \tag{9.28}$$

where $Z_i^b(n) = \log \frac{g_1^{(i)}(U_i(n)))}{g_0^{(i)}(U_i(n))}$ is the partial LLR between the "change" and "no-change" hypotheses for the binary sequence, which is given by

$$Z_i^b(n) = a_i U_i(n) + a_{0,i}. \tag{9.29}$$

Here

$$a_i = \log \frac{\beta_i(1-\beta_{0,i})}{\beta_{0,i}(1-\beta_i)}, \quad a_{0,i} = \log \frac{1-\beta_i}{1-\beta_{0,i}}.$$

Then the CUSUM detection procedure at the fusion center is given by the stopping time

$$T_{\mathrm{CS}}^b(h) = \min\left\{n \geqslant 1 \colon W^b(n) \geqslant h\right\}, \tag{9.30}$$

where h is a positive threshold which is selected so that $\mathsf{ARL}(T_{\mathrm{CS}}^b(h)) \geqslant \gamma$. In what follows this detection procedure will be referred to as the binary quantized CUSUM test (BQ-CUSUM).

The BQ-CUSUM procedure with $h = \log \gamma$ is asymptotically optimal as $\gamma \to \infty$ in the class of tests with binary quantization in the sense of minimizing the SADD in the class $\Delta(\gamma)$. More specifically, the tradeoff curve for the optimal binary test is

$$\mathsf{SADD}(T_{\mathrm{CS}}^b) \sim \frac{\log \gamma}{I_{\mathrm{tot}}^b}, \quad \gamma \to \infty, \tag{9.31}$$

where $I_{\mathrm{tot}}^b = \sum_{i=1}^N \max_{t_i} I_i^b(t_i)$ is the total maximal K-L distance (optimized over the quantization thresholds t_i); $I_i^b(t_i) = [\beta_i(t_i) a_i(t_i) + a_{0,i}(t_i)]$ is the K-L distance for the binary sequence in the i-th sensor for the quantization threshold t_i.

To optimize the performance, one should choose thresholds $t_1, \dots, t_N$ so that the K-L divergence is maximized, i.e.,

$$t_i^0 = \arg\max_{t_i} I_i^b(t_i), \ l = 1, \dots, N, \tag{9.32}$$

in which case the supremum average detection delay for the optimal BQ-CUSUM test is

$$\mathsf{SADD}(T_{\mathrm{CS}}^b) = \frac{h_\gamma}{I_{tot}^b(\mathbf{t}^0)} + O(1) \text{ as } \gamma \to \infty, \tag{9.33}$$

where $I_{tot}^b(\mathbf{t}^0) = \sum_{i=1}^N \max_{t_i} I_i^b(t_i) = \sum_{l=1}^N I_i^b(t_i^0)$, and $\mathbf{t}^0 = (t_1^0, \dots, t_N^0)$.

7.3. Relative Efficiency

The asymptotic relative efficiency (ARE) of a detection procedure τ_γ with respect to a detection procedure η_γ, both of which meet the same lower bound γ for the ARL, will be defined as

$$\mathrm{ARE}(\tau_\gamma; \eta_\gamma) = \lim_{\gamma \to \infty} \frac{\mathsf{SADD}(\tau_\gamma)}{\mathsf{SADD}(\eta_\gamma)}. \tag{9.34}$$

Using (9.2) and (9.31), we obtain that the ARE of the globally asymptotically optimal test ν with respect to the BQ-CUSUM test $T^b_{\rm CS}$ is

$$\mathrm{ARE}(\nu; T^b_{\rm CS}) = \lim_{\gamma\to\infty} \frac{\inf_{\tau\in\mathbb{C}_\gamma} \mathsf{SADD}(\tau)}{\mathsf{SADD}(T^b_{\rm CS}(h_\gamma))} = \frac{I^b_{\rm tot}}{I_{\rm tot}}. \tag{9.35}$$

Since $I_{\rm tot}$ is always larger than $I^b_{\rm tot}$, the value of $\mathrm{ARE} < 1$. However, our study presented below shows that: (a) certain decentralized asymptotically globally optimal tests may perform worse in practically interesting prelimit situations when the false alarm rate is moderately low but not very low, and (b) the centralized CUSUM only 20-30% better (in terms of the ADD). We therefore conclude that the BQ-CUSUM test is a good solution to the decentralized change detection problem whenever the post-change distribution is completely specified.

It can be shown that for the three particular models, namely Gaussian $\mathcal{N}(0,1) \to \mathcal{N}(\theta,1)$, Poisson $\mathcal{P}(1) \to \mathcal{P}(\theta)$, and Exponential $\mathrm{Exp}(1) \to \mathrm{Exp}(\theta)$, the ARE is a monotone function of θ in the interval $[2/\pi, 1]$, and $\lim_{\theta\to 0} \mathrm{ARE}(\theta) = 2/\pi$ for the Gaussian model and $\lim_{\theta\to 1} \mathrm{ARE}(\theta) = 2/\pi$ for the other two models. Also, $\lim_{\theta\to\infty} \mathrm{ARE}(\theta) = 1$ for all three models. Therefore, we expect that in the worst case scenario (for close hypotheses) the loss due to binary quantization is about 36%, and it is small for far hypotheses. This is confirmed by simulations.

For the sake of simplicity, consider a symmetric Gaussian case where the K-L information numbers are identical in all sensors, $\mathcal{N}(0,1)$ pre-change and $\mathcal{N}(\theta,1)$ post-change. Table 9.1 and Figure 9.3 illustrate how the ARE evolves with respect to the parameter θ.

It is seen that in the vicinity of $\theta \approx 0$ the ARE is close to $2/\pi$. While the ARE does not approach 1 too fast, the real relative efficiency RE defined as the ratio of "real" ADDs reaches 1 very fast: for $\theta \geqslant 5$ the real relative efficiency is already 1, so no further improvement is possible/necessary. Specifically, the RE is defined as $\mathrm{RE}(\gamma) = \mathsf{SADD}^c_\theta/\mathsf{SADD}^b_\theta$, where the estimates of

$$\mathsf{SADD}_\theta = \max\left\{0, \frac{\log\gamma}{I} - 1\right\}$$

take into account that they cannot be smaller than 0. More accurate results obtained by Monte Carlo yield similar conclusions. See Section 9.

8. A Decentralized Approach for Composite Hypotheses: Unknown Parameter Values

8.1. Impractical Approach – Worst-case Optimization

We begin with considering the approach that intuitively seems appealing but turns out to be almost completely impractical. Indeed, the first idea which deserves attention is to try optimizing in the Worst Case Scenario (i.e., to optimize in the most unfavorable conditions with respect to the parameter value):

$$\inf_{\theta>0} \mathrm{ARE}_\theta(t_0(\theta)) = \inf_{\theta>0}\max_{t>0} \mathrm{ARE}_\theta(t) = \lim_{\theta\to 0} \mathrm{ARE}_\theta(t_0(\theta)) = 2/\pi \approx 0.637.$$

However, this approach is impractical, which is immediately confirmed by the following computations. Possible but impractical solution: choose small $\theta = \theta'$ such that the detection with a reasonable delay is not possible for smaller values; find $t_0(\theta') = t^*$ that maximizes K-L; and use it in the distributed system. In Table 9.2 we use $\theta' = 0.1$. It can be seen that the ARE decreases very fast when θ increases.

8.2. Practical Approach – Decentralized M-BQ-CUSUM Test

For this reason we now propose a different approach that is based on a multichart CUSUM that uses multiple reference parameter values. The quantization thresholds are optimized for these reference points, where component, partial BQ-CUSUM tests are optimal. As a result, this approach provides a quite accurate approximation to the entire ADD envelop.

Table 9.1: Numerical Results for the Gaussian Scenario (Two Sensors, $N = 2$)

θ	0.01	0.1	1	2	4	5
$t^0(\theta)$	0.00792	0.0792	0.7941	1.60083	3.28628	4.16375
$I_c(\theta)$	0.00005	0.005	0.5	2	8	12.5
$I^b(t^0(\theta)\|\theta)$	0.000032	0.003183	0.318566	1.27879	5.234373	8.332713
ARE_θ	0.636619	0.636624	0.637133	0.639395	0.654297	0.666617
$\text{RE}_\theta(\gamma = 10^3)$	0.636619	0.636624	0.637133	0.639395	0.757753	1
$\text{RE}_\theta(\gamma = 10^4)$	0.636619	0.636624	0.637133	0.639395	0.654297	0.904713

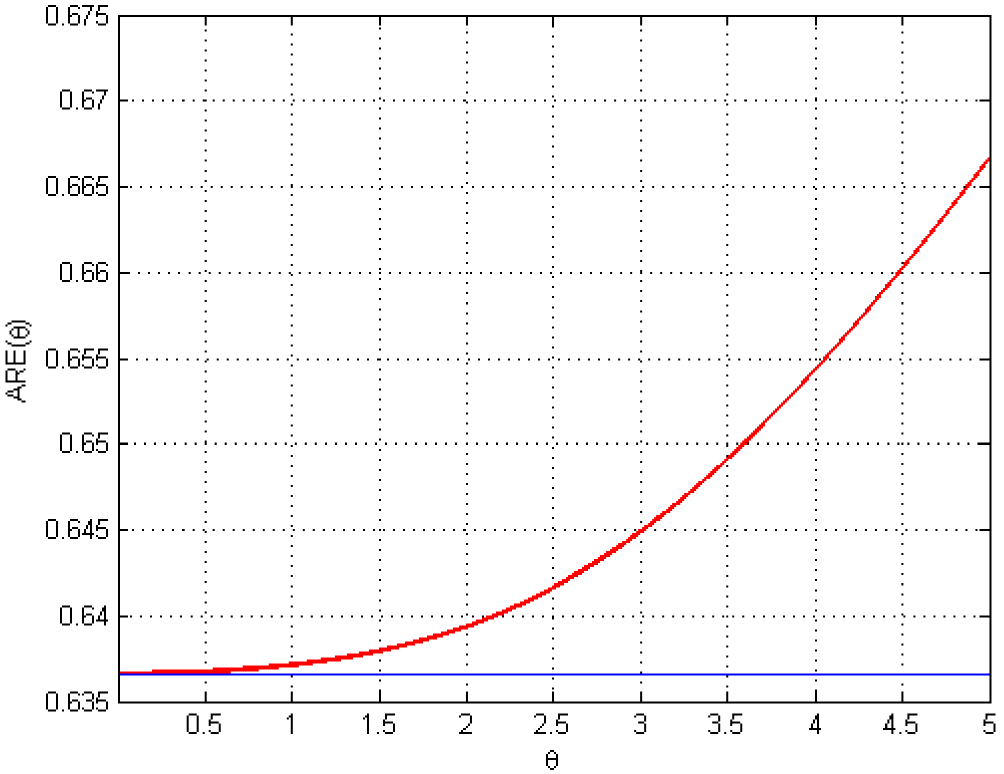


Figure 9.3: Asymptotic relative efficiency.

Table 9.2: Numerical Results for the Gaussian Scenario and with Worst Case Optimization

θ	0.1	0.5	1.0	1.5	2.0	3.0	5.0
$t_0(\theta)$	0.0792	0.3963	0.7941	1.1951	1.6008	2.4306	4.1639
$\text{ARE}_\theta(t^*)$	0.637	0.614	0.533	0.423	0.315	0.166	0.061
$\text{ARE}_\theta(t_0(\theta))$	0.637	0.637	0.637	0.638	0.639	0.645	0.667

As in Section 5, we define an interval $[\underline{\theta}, \overline{\theta}]$ for the post-change parameter and $M \geqslant 2$ reference points in this interval. The M-BQ-CUSUM procedure consists of implementing M BQ-CUSUM tests in parallel, each tuned to and optimized with respect to the corresponding reference point.

To be specific, let $\theta_m \in [\underline{\theta}, \overline{\theta}]$, $m = 1, 2, \dots, M$ be a set of reference points. At each of these points, sensors perform quantization of the observations using LLR-quantizers (9.27), i.e., for the m-th reference point the outputs of the quantizers are $U_{l,m}(n) = \mathbb{1}_{\{\ell_{m,l}(n) \geqslant t_{m,l}\}}$, where $\ell_{m,l}(n) = \log[g_l^{\theta_m}(X_l(n))/f_l^{\mu_l}(X_l(n))]$ is the LLR tuned to θ_m. The quantization threshold $t_{l,m}$ is chosen so that the K-L divergence is maximized for the corresponding point, i.e., for $l = 1, \dots, N$ and $m = 1, \dots, M$,

$$I_{l,m}^{\mathrm{BQ}}(t_{l,m}^0) = \underset{t_{l,m}>0}{\arg\max}\, I_{l,m}^{\mathrm{BQ}}(t_{l,m}), \tag{9.36}$$

where $I_{l,m}^{\mathrm{BQ}} = \mathbb{E}_1^{\theta_m} \ell_{l,m}^{\mathrm{BQ}}(1) = \beta_g(l,m) a_g(l,m) + a_f(l,m)$ is the K-L divergence for the binary sequence at the m-th reference point in the l-th sensor, and all the notation is defined in Section 7.2. In particular, the LLR $\ell_{l,m}^{\mathrm{BQ}}(n)$ for the binary sequence at the m-th reference point is given by (9.29) with obvious inclusion of the parameter θ_m.

The M-BQ-CUSUM stopping time is then defined as follows. First, we define the BQ-CUSUM statistics for each of the reference points

$$W_m^{\mathrm{BQ}}(n) = \max\left\{0, W_m^{\mathrm{BQ}}(n) + \sum_{l=1}^{N} \ell_{l,m}^{\mathrm{BQ}}(n)\right\},$$

with $W_m^{\mathrm{BQ}}(0) = 0$, and the corresponding stopping times for the latter statistics

$$\tau_m^{\mathrm{BQ}}(h_m) = \min\left\{n \geqslant 1 \colon W_m^{\mathrm{BQ}}(n) \geqslant h_m\right\}.$$

The M-BQ-CUSUM stopping time is the minimum of these stopping times:

$$\tau_{\mathrm{MBQ}}(\mathbf{h}) = \min_{1 \leqslant m \leqslant M} \tau_m^{\mathrm{BQ}}(h_m), \tag{9.37}$$

where $\mathbf{h} = (h_1, h_2, \dots, h_M)$, $h_m > 0$.

Now, as it has been outlined in Section 7.2, if the true value of the parameter $\theta = \theta_i$, then the asymptotically minimax-optimal solution to the changepoint problem in the class of binary quantizers is given by the LLR-quantizer with the threshold $t_{l,i}^0$ (in the l-th sensor), as specified in (9.36), followed by the BQ-CUSUM stopping rule $\tau_i^{\mathrm{BQ}}(h_i)$ at the fusion center (see (9.30)). Combining this with Theorem 5.1 allows us to conclude that the proposed M-BQ-CUSUM procedure is asymptotically optimal in the class of procedures with binary quantization when the post-change parameter is equal to reference points $\theta_1, \theta_2, \dots, \theta_M$.

Exact results are given in the following theorem which follows from the above argument and Theorem 5.1. We use the same notation for constants v_m ($0 < v_m < 1$) as above keeping in mind that in this section they are computed for the Bernoulli sequences.

Theorem 8.1. *Assume that LLR $\ell_{l,m}^{\mathrm{BQ}}(1)$ is non-arithmetic.*

(i) Let $h_1 = \cdots = h_M = h$ and let

$$h = h(\gamma) = \log\left(\gamma \sum_{m=1}^{M} \zeta_m^2 \sum_{l=1}^{N} I_{l,m}^{\mathrm{BQ}}(t_{l,m}^0)\right).$$

Then, as $\gamma \to \infty$,

$$\mathsf{ARL}(\tau_{\mathrm{MBQ}}(h(\gamma))) = \gamma(1 + o(1)),$$

and, for all $m = 1, \dots, M$,

$$\inf_{\tau \in \bar{b}\Delta_{\mathrm{BQ}}(\gamma)} \mathsf{SADD}_{\theta_m}(\tau) \sim \mathsf{SADD}_{\theta_m}(\tau_{\mathrm{MBQ}}(h(\gamma))) \sim \frac{\log \gamma}{\sum_{l=1}^{N} I_{l,m}^{\mathrm{BQ}}(t_{l,m}^0)}. \tag{9.38}$$

(ii) Let, for $m = 1, \dots, M$,

$$h_m = h_m(\gamma) = \log\left(\gamma M \zeta_m^2 \sum_{l=1}^{N} I_{l,m}^{\mathrm{BQ}}(t_{l,m}^0)\right).$$

Then, as $\gamma \to \infty$,

$$\begin{aligned} \mathsf{ARL}(\tau_{\mathrm{BQ},m}(h_m(\gamma))) &= M\gamma[1 + o(1)], \; m = 1, \dots, M; \\ \mathsf{ARL}(\tau_{\mathrm{MBQ}}(\mathbf{h}(\gamma))) &= \gamma[1 + o(1)], \end{aligned}$$

and, for all $m = 1, \dots, M$, asymptotic relations (9.38) *hold.*

Therefore, M-BQ-CUSUM is asymptotically optimal in the class $\bar{b}\Delta_{\mathrm{BQ}}(\gamma)$ in the sense of minimizing SADD_{θ_m}, $m = 1, \dots, M$ at the reference points in both scenarios (with and without balancing). Note that the results analogous to Theorem 5.1 (i)–(ii) also hold in the binary case considered.

Note that all the above formulas are valid for the exponential family. The quantization thresholds, however, depend on the model. In the symmetric Gaussian example considered in Section 9.2, the optimal thresholds for each sensors are the same. For the m-th CUSUM tuned to $\theta = \theta_m$, the threshold t_m^0 is

$$t_m^0(\theta_m) = \arg\max_t \left\{ \Phi(t - \theta_m) \log\left[\frac{\Phi(t - \theta_m)}{\Phi(t)}\right] + \Big(1 - \Phi(t - \theta_m)\Big) \log\left[\frac{1 - \Phi(t - \theta_m)}{1 - \Phi(t)}\right] \right\},$$

and the corresponding numbers $\mathcal{J}_m(\theta)$ are

$$\mathcal{J}_m(\theta) = \Phi(t_m^0 - \theta) \log\left[\frac{\Phi(t_m^0 - \theta)}{\Phi(t_m^0)}\right] + \Big(1 - \Phi(t_m^0 - \theta)\Big) \log\left[\frac{1 - \Phi(t_m^0 - \theta)}{1 - \Phi(t_m^0)}\right],$$

where $\Phi(x)$ is the standard Gaussian distribution function.

9. Monte Carlo Experiments

9.1. Monte Carlo Experiments for Simple Hypotheses

In this section, we present the results of MC experiments for the Poisson example where observations in the i-th sensor $X_i(n)$, $n \geqslant 1$ follow the common Poisson distribution $\mathcal{P}(\mu_i)$ in the pre-change mode and the common Poisson distribution $\mathcal{P}(\theta_i)$ after the change occurs, i.e., for $m = 0, 1, 2, \dots$ and $\nu = k$,

$$\mathbb{P}_k(X_i(n) = m) = \begin{cases} \frac{(\mu_i)^m}{m!} e^{-\mu_i} & \text{for } k > n, \\ \frac{(\theta_i)^m}{m!} e^{-\theta_i} & \text{for } k \leqslant n, \end{cases}$$

where without loss of generality we assume that $\theta_i > \mu_i$.

Write $Q_i = \theta_i/\mu_i$. It is easily seen that the LLR statistic in the i-th sensor has the form

$$Z_n(i) = X_i(n) \log(Q_i) - \mu_i(Q_i - 1), \tag{9.39}$$

and the K-L information numbers

$$I_i = \theta_i \log Q_i - \mu_i(Q_i - 1), \quad i = 1, \dots, N. \tag{9.40}$$

It follows from (9.2), (9.40) and the above discussion that the centralized CUSUM and AO-LD-CUSUM tests with the thresholds $h = \log \gamma$ are first-order globally asymptotically optimal and

$$\inf_{T \in \mathbb{C}_\gamma} \mathsf{SADD}(T) \sim \mathsf{SADD}(\tau_c) \sim \mathsf{SADD}(\tau_{ld}) \sim \frac{\log \gamma}{\sum_{i=1}^N [\theta_i \log Q_i - \mu_i (Q_i - 1)]}. \tag{9.41}$$

This means that the ARE of these detection tests with respect to the globally optimal test is equal to 1.

In order to evaluate the ARE of an optimal test ν (e.g., the centralized CUSUM test τ_c) with respect to the BQ-CUSUM test (9.30) we use (9.35), which yields

$$\mathrm{ARE}(\nu; \tau_b) = \frac{\sum_{i=1}^N \max_{t_i} [\beta_i(t_i) a_i(t) + a_{0,i}(t_i)]}{\sum_{i=1}^N [\theta_i \log Q_i - \mu_i (Q_i - 1)]}, \tag{9.42}$$

where the probabilities $\beta_{0,i}(t)$ and $\beta_i(t)$ are given by:

$$\beta_{0,i}(t_i) = \sum_{k=\lceil t_i \rceil}^{\infty} \frac{\mu_i^k \, e^{-\mu_i}}{k!}, \; \beta_i(t_i) = \sum_{k=\lceil t_i \rceil}^{\infty} \frac{\theta_i^k \, e^{-\theta_i}}{k!}.$$

The optimal values of t_i^0 that maximize the K-L numbers are easily found based on these formulas. Consider a symmetric case where $\mu_i = 10$ and $\theta_i = 12$ for all $i = 1, \dots, N$. Then $I_i = I = 0.1879$, the optimum threshold is $t_i^0 = 12$, and the corresponding maximum K-L distance for the binary sequence $I_i^b(t_i^0) = I^b = 0.119$. Therefore, the loss in efficiency of the BQ-test compared to the globally asymptotically optimal detection procedure is $\mathrm{ARE}(\nu; \tau_b) = 0.119/0.1879 = 0.63$, i.e., for the large ARL we expect about 37% increase in the average detection delay compared to the centralized CUSUM (C-CUSUM). The following MC simulations show that for the practically interesting values of the ARL (up to $13,360$) the gain of the optimal C-CUSUM test is even smaller, while the AO-LD-CUSUM test performs worse than the BQ-CUSUM test due to the reasons discussed in Section 6.1.

MC simulations have been performed for the above symmetric situation (i.e., $\mu_i = \mu = 10$ and $\theta_i = \theta = 12$) with $N = 5$ sensors. We used 10^5 MC replications in the experiment. The operating characteristics of the five detection tests (SADD vs $\log(\mathsf{ARL})$) are shown in Figure 9.4. It is seen that the BQ-CUSUM test substantially outperforms the AO-LD-CUSUM test for all false alarm rate range used in simulations. This result confirms our conjecture. It is also seen that both Min-LD-CUSUM and Max-LD-CUSUM perform worse than both BQ-CUSUM and AO-LD-CUSUM tests.

Table 9.3 shows the relative efficiency of the BQ-CUSUM procedure with respect to four other detection procedures, which is defined as the ratio of average detection delays for the same ARL: $\mathsf{SADD}(\tau_b)/\mathsf{SADD}(\nu)$, where ν is a corresponding detection test, i.e., $\nu = \tau_c$, T_{ld}, etc. It follows from the table that for the BQ-CUSUM the increase in the SADD compared to the globally optimal centralized CUSUM is 34% for high false alarm rate, 35% for moderate and low false alarm rate, and 37% for very low false alarm rate. Note that the last column presents the ARE. On the other hand, the BQ-CUSUM outperforms the AO-LD-CUSUM for all range of tested ARL values, from 33 to 13,360. The gain is 30% for high false alarm rate and slowly reduces to 18% for low false alarm rate.

9.2. Monte Carlo Experiments for Composite Hypotheses

By means of Monte Carlo simulations, in this section we demonstrate the capabilities of the multi-chart detection techniques proposed in Section 8.2. We are particularly interested in the relative efficiency (RE) of the detection procedures as a function of the parameter θ. For two procedures, τ and η, the relative efficiency of η with respect to τ at the point θ is defined as $\mathrm{RE}_\theta(\tau, \eta) = \mathsf{SADD}_\theta(\tau)/\mathsf{SADD}_\theta(\eta)$, where it is assumed that both procedures satisfy $\mathsf{ARL} \approx \gamma$. Note that $\lim_{\gamma \to \infty} \mathrm{RE}_\theta(\tau, \eta) = \mathrm{ARE}_\theta(\tau, \eta)$.

We consider a *symmetric* three-sensor scenario ($N = 3$), where for each sensor both the pre- and post-change observations are iid unit-variance Gaussian random variables having expected values zero and $\theta > 0$, respectively. The observations are also assumed independent across the sensors.

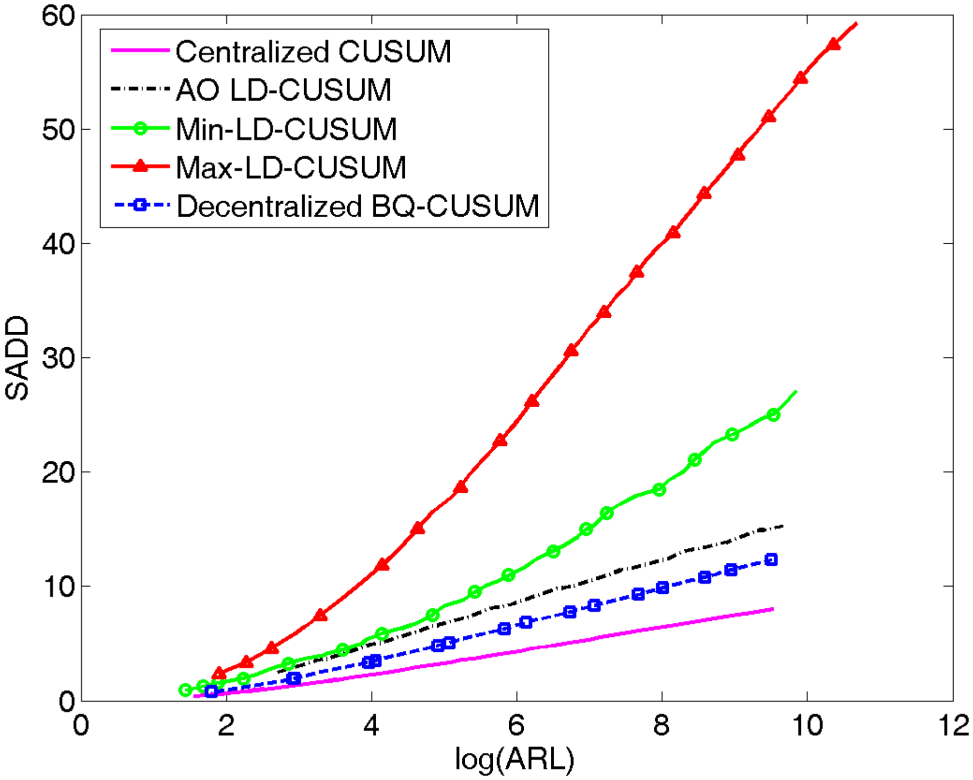


Figure 9.4: Operating characteristics of detection procedures.

Table 9.3: Relative Efficiency of the Decentralized BQ-CUSUM Test

$\log(\mathrm{ARL})$	3.5	4.5	5.5	6.5	7.5	8.5	9.5	∞
ARL	33	90	245	665	1808	4915	13360	∞

Test	Relative Efficiency of the Decentralized BQ-CUSUM Test							
C-CUSUM	1.51	1.51	1.51	1.53	1.53	1.53	1.54	1.59
AO-LD-CUSUM	0.71	0.73	0.75	0.76	0.78	0.80	0.82	1.59
Min-LD-CUSUM	0.62	0.58	0.55	0.54	0.51	0.51	0.51	0.316
Max-LD-CUSUM	0.33	0.30	0.27	0.26	0.25	0.24	0.24	0.316

All simulations have been performed for $\gamma = 10^4$ (relatively low FAR) and for θ between 0.05 and 1, which ensures high enough $\mathsf{SADD}_\theta(\tau)$ to be able to notice the difference in performance (e.g., $\mathsf{SADD}_\theta(\tau) \approx 10$ for $\theta = 1$).

Most importantly, note that in order to minimize the communication load between the fusion center and the sensors, it is desirable to use as few reference points as possible. Moreover, when the number of reference points M increases, the ARL also increases. This requires higher threshold values, which in turn leads to an increase of the detection delay. With this in mind, we considered the case of $M = 2$ and 3.

We begin with comparing centralized CUSUM procedures, C-CUSUM and M-C-CUSUM. The red curve in Figure 9.5 represents the relative efficiency of the unbalanced 2-C-CUSUM (with a single threshold $h_m = h$) with reference points 0.1 and 0.9 with respect to C-CUSUM. It is seen that around the right reference point $\theta_2 = 0.9$ RE is close to 1, while in the vicinity of the left reference point $\theta_1 = 0.1$ RE is around 0.6. This is because the threshold is the same for both stopping times $\tau_{\mathrm{C},1}$ and $\tau_{\mathrm{C},2}$, in which case $\mathsf{ARL}(\tau_{\mathrm{MC}}) \approx \mathsf{ARL}(\tau_{\mathrm{C},2}(h))$. As a result, for most of the values of the parameter the behavior of 2-C-CUSUM is similar to that of C-CUSUM tuned to $\theta = 0.9$. In order to have more efficient detection for small changes, one has to use either more reference points or a symmetric 2-C-CUSUM (with different thresholds) balancing mean times as in (9.11).

To improve the performance for small changes, we added an extra reference point 0.2. The blue curve in Figure 9.5 shows the RE of 3-C-CUSUM with respect to C-CUSUM with reference points 0.1, 0.2 and 0.9. Observe that the relative efficiency never drops below the level of approximately 80% for values of $\theta \geqslant 0.2$, and it is equal to 70% for $\theta = 0.1$. For smaller values of θ, there is a drop in efficiency to approximately 50%. This is not surprising since this procedure is not designed to work with parameter values smaller than 0.1. We may conclude that adding the extra point does help. 3-C-CUSUM has a much better performance for small values of the parameter compared to 2-C-CUSUM. However, there is still a disbalance between the RE for small and large shifts.

Figure 9.6 (red curve) illustrates the performance of the 2-C-CUSUM in the balanced case (9.11) where every reference point contributes equally to the performance of the whole scheme. Comparing with the red curve in Figure 9.5, we observe that for the left reference point (0.1) the performance became much better (88% vs. 65%), while for the right one (0.9) slightly worse (90% vs. 99%). For θ around 0.4, there is a dip caused primarily by the fact that the reference points are too distant from each other.

To eliminate the drop in the middle we used an extra reference point 0.4. The result is shown in Figure 9.6, the blue curve. In this case the performance remains almost constant (uniform) for the entire range of θ (between 85% and 90% for most parameter values, and over 80% for the entire range).

It is clear that if one is interested in relatively high efficiency for all parameter values (small, moderate and large changes), then the proposed balanced approach can be recommended for implementation. However, if one is interested in rapid detection of only moderate and large changes, then the first unbalanced approach with constant thresholds can be used.

We now proceed with the results of the experimental study of the binary quantized procedures. We present the results only for unbalanced 3-BQ-CUSUM (with equal thresholds $h_1 = h_2 = h_3 = h$). Figure 9.7 shows the relative efficiency of 3-BQ-CUSUM with respect to the optimal BQ-CUSUM (which knows θ). The behavior is similar to that in the centralized case. As before, the relative efficiency stays above the level of approximately 80% for values of $\theta \geqslant 0.2$, while for small changes there is a drop in efficiency to approximately 50%.

Figure 9.8 shows the relative efficiency of the binary 3-BQ-CUSUM with respect to the centralized 3-C-CUSUM. RE remains at the level of approximately 70% for all parameter values. Therefore, increase in the average detection delay of the procedure with binary quantization in three points (with rather low requirements to communication bandwidth) is only 30% compared to the centralized scheme that requires transmission of the original uncompressed data.

Finally, Table 9.4 summarizes the results of Monte Carlo simulations for asymmetric M-C-CUSUM (with equal thresholds).

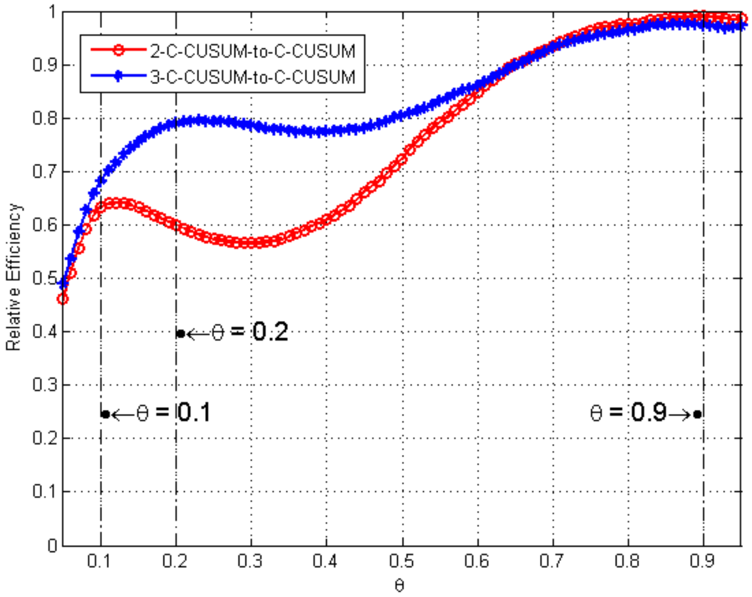


Figure 9.5: Unbalanced M-C-CUSUM-to-C-CUSUM relative efficiency: $\gamma = 10^4$; reference points 0.1, 0.2 and 0.9.

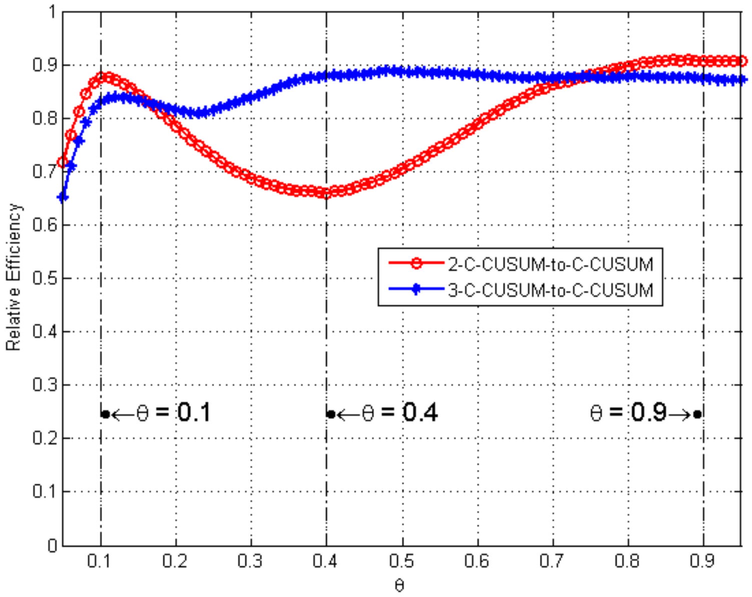


Figure 9.6: Balanced M-C-CUSUM-to-C-CUSUM relative efficiency: $\gamma = 10^4$; reference points 0.1, 0.4 and 0.9.

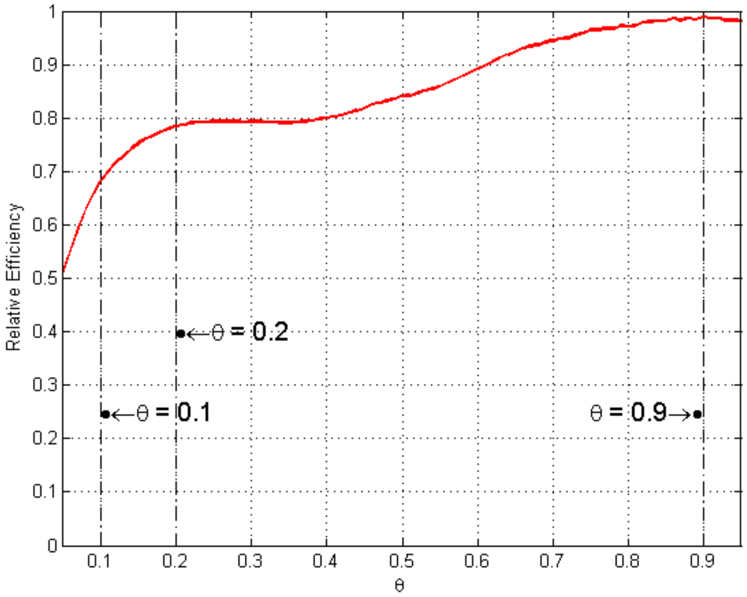


Figure 9.7: Unbalanced 3-BQ-CUSUM-to-BQ-CUSUM relative efficiency: $\gamma = 10^4$; reference points 0.1, 0.2 and 0.9.

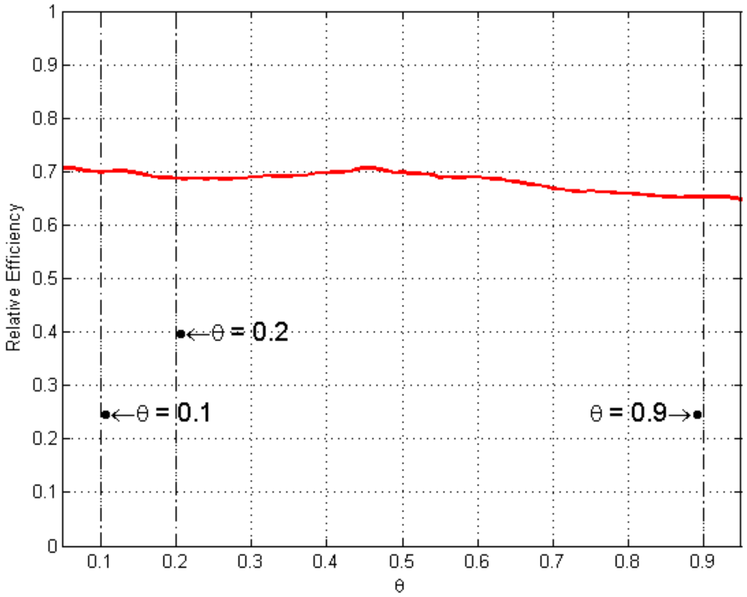


Figure 9.8: Unbalanced 3-BQ-CUSUM-to-3-C-CUSUM relative efficiency: $\gamma = 10^4$; reference points 0.1, 0.2 and 0.9.

Table 9.4: Summary of Numerical Results for the Gaussian Scenario ($N = 3$, $M = 3$, $\gamma = 10^4$)

θ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\mathsf{SADD}(\tau_{\mathrm{C}})$	269.31	89.22	45.06	27.32	18.32	13.13	9.83	7.58	6.0
$\mathsf{SADD}(\tau_{\mathrm{BQ}})$	380.64	128.2	66.02	40.38	27.32	19.69	14.85	11.66	9.32
$\mathsf{SADD}(\tau_{\mathrm{MC}})$	394.72	112.6	57.75	35.55	22.6	15.18	10.59	7.92	6.12
$\mathsf{SADD}(\tau_{\mathrm{MBQ}})$	551.75	163.35	83.18	50.62	32.42	22.1	15.65	12.06	9.41
$\mathrm{RE}(\tau_{\mathrm{MBQ}}; \tau_{\mathrm{BQ}})$	0.69	0.78	0.79	0.8	0.84	0.89	0.95	0.97	0.99
$\mathrm{RE}(\tau_{\mathrm{MBQ}}; \tau_{\mathrm{MC}})$	0.69	0.69	0.69	0.7	0.7	0.69	0.68	0.66	0.65

Chapter 10

Some Variants of the Quickest Change Detection Problem and Their Solutions

This chapter is intended to summarize contributions of the group of Dr. Veeravalli to two variants of the quickest change detection problem and their solutions.

1. Quickest Change Detection of a Markov Process Across a Sensor Array

In the standard formulation of the change detection problem, there is a sequence of observations whose density changes at some unknown point in time and the goal is to detect the changepoint as soon as possible. However, in many scenarios such as detecting pollutants and biological warfare agents, the change process is governed by the movement of the agent through the medium. Thus, it is more suitable to consider the case where the statistics of each sensor's observations may change at different points in time.

In this work, we consider a Bayesian version of this problem and assume that the point of disruption (that needs to be detected) is a random variable with a geometric distribution. More general disruption models can be considered, but the case of a geometric prior has an intuitive and appealing interpretation due to the *memorylessness* property of the geometric random variable. In addition, the practically relevant rare disruption regime can be obtained by letting the geometric parameter go to zero. We assume that the L sensors are placed in an array or a line and they observe the change as it propagates through them. The progression of change in only one strictly determined direction can be thought as a first approximation to more realistic situations. The inter-sensor delay is modeled with a Markov model and in particular, the focus is on the case where the inter-sensor delay is also geometric. This model can be viewed as a first order approximation to more general propagation models, with the zero-th order model being the case where the statistical properties of the sensors' observations change at the same time.

We study the centralized case, where the fusion center has complete information about the observations at all the L sensors, the change process statistics, and the pre- and the post-change densities. This is applicable in scenarios where: i) the fusion center is geographically collocated with the sensors so that ample bandwidth is available for reliable communication between the sensors and the fusion center; and ii) the impact of the disruption-causing agent on the statistical dynamics of the change process and the statistical nature of the change so induced can be modeled accurately. Note that under the centralized model, the special case where the change happens at the same time at all sensors corresponds to the standard (single sensor) quickest change detection problem Shiryaev [149] with an L-vector observation.

1.1. Problem Formulation

Consider a distributed system with an array of L sensors, as in Figure 10.1, that observes an L-dimensional discrete-time stochastic process $\mathbf{Z}_k = [Z_{k,1}, \cdots, Z_{k,L}]$, where $Z_{k,\ell}$ is the observation at the ℓ-th sensor at the k-th time instant. A disruption in the sensing environment occurs at the random time instant Γ_1, and

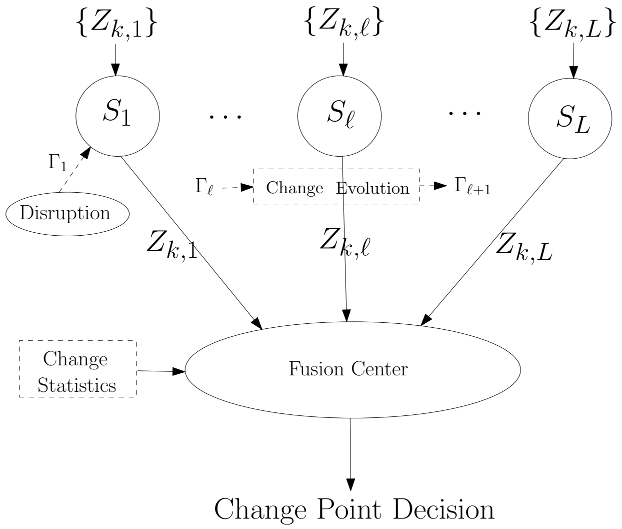


Figure 10.1: Changepoint detection across a linear array of sensors.

hence, the density[1] of the observations at each sensor undergoes a change from the null density f_0 to the alternate density f_1.

Change Process Model: We consider a *change process* where the change-point evolves across the sensor array. In particular, the change-point as seen by the ℓ-th sensor is denoted as Γ_ℓ. We assume that the evolution of the change process is Markovian across the sensors. That is,

$$\begin{aligned} & P\big(\{\Gamma_{\ell_1+\ell_2+\ell_3} = m_1+m_2+m_3\}\big|\{\Gamma_{\ell_1+\ell_2} = m_1+m_2\}, \{\Gamma_{\ell_1} = m_1\}\big) \\ = \; & P\big(\{\Gamma_{\ell_1+\ell_2+\ell_3} = m_1+m_2+m_3\}\big|\{\Gamma_{\ell_1+\ell_2} = m_1+m_2\}\big) \end{aligned}$$

for all ℓ_i and $m_i \geq 0, \; i = 1,2,3$. Further simplification of the analysis is possible under a *joint-geometric* model on $\{\Gamma_\ell\}$. Under this model, the change-point (Γ_1) evolves as a geometric random variable with parameter ρ, and inter-sensor change propagation is modeled as a geometric random variable with parameter $\{\rho_{\ell-1,\ell}, \; \ell = 2, \cdots, L\}$. That is,

$$\begin{aligned} & P\big(\{\Gamma_1 = m\}\big) = \rho\,(1-\rho)^m, \; m \geq 0 \text{ and} \\ & P\big(\{\Gamma_\ell = m_1+m_2\}\big|\{\Gamma_{\ell-1} = m_2\}\big) = \rho_{\ell-1,\ell}\,(1-\rho_{\ell-1,\ell})^{m_1}, \; m_1 \geq 0 \end{aligned}$$

independent of $m_2 \geq 0$ for all ℓ such that $2 \leq \ell \leq L$. We will find it convenient[2] to set $\rho_{0,1} = \rho$ and $\rho_{L,L+1} = 0$ so that $\rho_{\ell-1,\ell}$ is defined for all $\ell = 1, \cdots, L+1$.

While a joint-geometric model is consistent with the Markovian assumption as only the inter-sensory (one-step) propagation parameters are modeled, the change-points at the individual sensors themselves are *not* geometric. The joint-geometric model can be viewed as a first order approximation of more realistic propagation scenarios. In particular, note that $\rho \to 1$ corresponds to the case where instantaneous disruption (that is, the event $\{\Gamma_1 = 0\}$) has a high probability of occurrence. On the other hand, $\rho \to 0$ uniformizes the change-point in the sense that the disruption is equally likely to happen at any point in time. This case where the disruption is "rare" is of significant interest in practical systems Basseville and Nikiforov [15], Poor and Hadjiliadis [130], Tartakovsky and Veeravalli [170, 171, 174], Veeravalli [181]. This is also the case where we will be able to make insightful statements about the structure of the optimal stopping rule. Similarly, we can also distinguish between two extreme scenarios at sensor ℓ depending on whether $\rho_{\ell-1,\ell} \to 0$ or $\rho_{\ell-1,\ell} \to 1$. The case where $\rho_{\ell-1,\ell} \to 1$ corresponds to instantaneous change propagation at sensor ℓ and $\{\Gamma_\ell = \Gamma_{\ell-1}\}$ with high probability. The case where $\rho_{\ell-1,\ell} \to 0$ corresponds to uniformly

[1] We assume that the pre-change (f_0) and the post-change (f_1) densities exist.

[2] This is also consistent with an equivalent $(L+2)$-sensor system where sensor indices run through $\{\ell = 0, \cdots, L+1\}$.

likely propagation delay. The widely-used assumption of instantaneous change propagation across sensors is equivalent to assuming $\rho_{\ell-1,\ell} = 1$ for all $\ell = 2, \cdots, L$.

Observation Model: To simplify the study, we assume that the observations (at every sensor) are independent, conditioned[3] on the change hypothesis corresponding to that sensor, and are identically distributed pre- and post-change, respectively. That is,

$$Z_{k,\ell} \;\; \sim \;\; \begin{cases} \text{i.i.d. } f_0 & \text{if} \;\; k < \Gamma_\ell, \\ \text{i.i.d. } f_1 & \text{if} \;\; k \geq \Gamma_\ell. \end{cases}$$

We will describe the above assumption as that corresponding to an "i.i.d. observation process." Let $D(f_1, f_0)$ denote the Kullback-Leibler divergence between f_1 and f_0. That is,

$$D(f_1, f_0) = \int \log\left(\frac{f_1(x)}{f_0(x)}\right) f_1(x) dx.$$

We also assume that the measure described by f_0 is *absolutely continuous* with respect to that described by f_1. That is, if $f_1(x) = 0$ for some x, then $f_0(x) = 0$. This condition ensures that $E_{\bullet|f_1}\left[\frac{f_0(\bullet)}{f_1(\bullet)}\right] = 1$.

Performance Metrics: We consider a *centralized, Bayesian* setup where a fusion center has complete knowledge of the observations from all the sensors, $I_k \triangleq \{\mathbf{Z}_1, \cdots, \mathbf{Z}_k\}$, in addition to knowledge of statistics of the change process (equivalently, $\{\rho_{\ell-1,\ell}\}$) and statistics[4] of the observation process (equivalently, f_0 and f_1). The fusion center decides whether a change has happened or not based on the information, I_k, available to it at time instant k (equivalently, it provides a stopping rule or stopping time τ).

The two conflicting performance measures for quickest change detection are the probability of false alarm, $\mathsf{PFA} \triangleq P\big(\{\tau < \Gamma_1\}\big)$, and the average detection delay, $\mathsf{ADD} \triangleq E\left[(\tau - \Gamma_1)^+\right]$, where $x^+ = \max(x, 0)$. This conflict is captured by the Bayes risk, defined as,

$$\begin{aligned} R(c) &\triangleq \mathsf{PFA} + c\mathsf{ADD} \\ &= E\left[\mathbb{1}\big(\{\tau < \Gamma_1\}\big) + c\,(\tau - \Gamma_1)^+\right] \end{aligned}$$

for an appropriate choice of per-unit delay cost c, where $\mathbb{1}\big(\{\cdot\}\big)$ is the indicator function of the event $\{\cdot\}$. We will be particularly interested in the regime where $c \to 0$. That is, a regime where minimizing PFA is more important than minimizing ADD, or equivalently, the asymptotics where $\mathsf{PFA} \to 0$.

The goal of the fusion center is to determine

$$\tau_{\mathsf{opt}} = \arg \inf_{\tau \,\in\, \mathbf{\Delta}_\alpha} \mathsf{ADD}(\tau)$$

from the class of change-point detection procedures $\mathbf{\Delta}_\alpha = \big\{\tau : \mathsf{PFA}(\tau) \leq \alpha\big\}$ for which the probability of false alarm does not exceed α. In other words, the fusion center needs to come up with a strategy (a stopping rule τ) to minimize the Bayes risk.

1.2. Dynamic Programming Framework

It is straightforward to check that Shiryaev [149, pp. 151-152] the Bayes risk can be written as

$$R(c) \;\; = \;\; P\big(\{\Gamma_1 > \tau\}\big) + cE\left[\sum_{k=0}^{\tau-1} P\big(\{\Gamma_1 \leq k\}\big)\right].$$

Towards solving for the optimal stopping time, we restrict attention to a finite-horizon, say the interval $[0, T]$, and proceed via a dynamic programming (DP) argument.

[3] More general observation (correlation) models are important in practical settings. This will be the subject of future work.

[4] We assume that the fusion center has knowledge of f_0 and f_1 so that it can use this information to declare that a change has happened. Relaxing this assumption is important in the context of practical applications and is the subject of current work.

The state of the system at time k is the vector $\boldsymbol{S}_k = [S_{k,1}, \ldots, S_{k,L}]$ with $S_{k,\ell}$ denoting the state at sensor ℓ. The state $S_{k,\ell}$ can take the value 1 (post-change), 0 (pre-change), or t (terminal). The system goes to the terminal state t, once a change-point decision τ has been declared. The state evolves as follows:

$$S_{k,\ell} = f(S_{k-1,\ell}, \Gamma_\ell, 1_{\{\tau \le k\}}),$$

where the transition function is given as

$$f(s,\gamma,a) = \begin{cases} 0 & \text{if} \quad \gamma > k, s \neq t, a = 0, \\ 1 & \text{if} \quad \gamma \le k, s \neq t, a = 0, \\ t & \text{if} \quad s = t \ \text{ or } \ a = 1 \end{cases}$$

with $\boldsymbol{S}_0 = \mathbf{0}$. Since $\boldsymbol{S}_{k-1}$ captures the information contained in $\{\Gamma_\ell \le j\}$ for $0 \le j \le k-1$ and all ℓ, given $\boldsymbol{S}_{k-1}$, $\{\Gamma_\ell \le k\}$ is independent of $\{\Gamma_\ell \le j, \ j \le k-1\}$ for all ℓ. Thus, the state evolution satisfies the Markov condition needed for dynamic programming.

The state is not observable directly, but only through the observations. The observation equation can be written as

$$Z_{k,\ell} = V_{k,\ell}^{(S_{k,\ell})} \mathbb{1}\big(\{S_{k,\ell} \neq t\}\big) + \xi \mathbb{1}\big(\{S_{k,\ell} = t\}\big), \ \ell \ge 1$$

where $V_{k,\ell}^{(0)}$ and $V_{k,\ell}^{(1)}$ are the k-th samples from independently generated infinite arrays of i.i.d. data according to f_0 and f_1, respectively. When the system is in the terminal state, the observations do not matter (since a change decision has already been made) and are hence denoted by a dummy random variable, ξ. It is clear that the observation uncertainty $(V_{k,\ell}^{(0)}, V_{k,\ell}^{(1)})$ satisfies the necessary Markov conditions for dynamic programming since they are i.i.d. in time.

Finally, the expected cost (Bayes risk) can be expressed as the expectation of an additive cost over time by defining

$$g_k(\boldsymbol{S}_k) = c\mathbb{1}\big(\{S_{k,1} = 1\}\big)$$

and a terminal cost $\mathbb{1}\big(\{S_{k,1} = 0\}\big)$. Thus the problem fits the standard dynamic programming framework with termination Bertsekas [22], with the sufficient statistic (belief state) being given by

$$P\big(\{\boldsymbol{S}_k = \boldsymbol{s}_k\} | I_k\big),$$

where $I_k = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_k\}$ for k such that $\boldsymbol{S}_k \neq t$, i.e., $S_{k,\ell} \in \{0,1\}$ for each ℓ. Note that this sufficient statistic is described by 2^L conditional probabilities, corresponding to the 2^L values that $\boldsymbol{s}_k$ can take. We will next see that this sufficient statistic can be further reduced[5] to only L independent probability parameters in the general case.

The fusion center determines τ, and hence, the minimum expected cost-to-go at time k for the above DP problem can be seen to be a function of I_k. For a finite horizon T, the cost-to-go function is denoted as $\widetilde{J}_k^T(I_k)$ and is of the form (see Bertsekas [22, p. 133], Veeravalli [181], for examples of similar nature):

$$\begin{aligned} \widetilde{J}_T^T(I_T) &= P\big(\{\Gamma_1 > T\} \big| I_T\big) \\ \widetilde{J}_k^T(I_k) &= \min \Big\{ P\big(\{\Gamma_1 > k\} \big| I_k\big), \ cP\big(\{\Gamma_1 \le k\} \big| I_k\big) \\ &\qquad + E\left[\widetilde{J}_{k+1}^T(I_{k+1}) \big| I_k \right] \Big\}, \ 0 \le k < T \end{aligned}$$

where I_0 is the empty set. The first term in the above minimization corresponds to the cost associated with stopping at time k, while the second term corresponds to the cost associated with proceeding to time $k+1$ without stopping. The minimum expected cost for the finite-horizon optimization problem is $\widetilde{J}_0^T(I_0)$.

[5] This should not be entirely surprising as our assumption of a line (or array) geometry imposes a "natural" ordering on the sensors' change-points. They can be arranged in non-decreasing order: $\Gamma_\ell \ge \Gamma_{\ell-1}$ for all ℓ.

Recursion for the Sufficient Statistics: We define an $(L+1)$-tuple of conditional probabilities, $\{p_{k,\ell},\ \ell = 1, \cdots, L+1\}$:

$$p_{k,\ell} \triangleq P\Big(\big\{\Gamma_{\ell-1} \leq k, \Gamma_\ell > k\big\} \big| I_k\Big).$$

We now show that $\boldsymbol{p}_k \triangleq [p_{k,1}, \cdots, p_{k,L+1}]$ can be obtained from $\boldsymbol{p}_{k-1}$ via a recursive approach. For this, we note that the underlying probability space Ω in the setup can be partitioned as

$$\begin{aligned} \Omega &= \bigcup_{\ell=1}^{L+1} T_{k,\ell} \quad \text{where} \\ T_{k,\ell} &\triangleq \big\{\Gamma_{\ell-1} \leq k, \Gamma_\ell > k\big\}. \end{aligned}$$

The event where no sensor has observed the change is denoted as $T_{k,1}$. On the other hand, $T_{k,\ell}$ (for $\ell \geq 2$) corresponds to the event where the maximal index of the sensor that has observed the change before time instant k is $\ell - 1$.

Observe that $p_{k,\ell}$ is the probability of $T_{k,\ell}$ conditioned on I_k. To show that $p_{k,\ell}$ can be written in terms of $\boldsymbol{p}_{k-1}$, the observations $\mathbf{Z}_k$ and the prior probabilities, we partition $T_{k,\ell}$ further as

$$\begin{aligned} T_{k,\ell} &= \bigcup_{j=1}^{\ell} U_{k,\ell,j} \\ U_{k,\ell,j} &\triangleq \big\{\Gamma_{j-1} \leq k-1, \Gamma_j = k, \cdots, \Gamma_{\ell-1} = k, \\ &\qquad \Gamma_\ell \geq k+1\big\},\ 1 \leq j \leq \ell. \end{aligned}$$

Note that $U_{k,\ell,j} \cap T_{k-1,j} = U_{k,\ell,j}$. Using the new partition $\{U_{k,\ell,j},\ j = 1, \cdots, \ell\}$ and applying Bayes' rule repeatedly, it can be checked that $p_{k,\ell}$ can be written as

$$\begin{aligned} p_{k,\ell} &= \frac{\sum_{m=1}^{\ell} f\left(\mathbf{Z}_k | I_{k-1}, U_{k,\ell,m}\right) P(U_{k,\ell,m} | I_{k-1})}{\sum_{j=1}^{L+1} \sum_{m=1}^{j} f(\mathbf{Z}_k | I_{k-1}, U_{k,j,m}) P(U_{k,j,m} | I_{k-1})} \\ &\triangleq \frac{\mathcal{N}_\ell}{\sum_{j=1}^{L+1} \mathcal{N}_j} \end{aligned} \tag{10.1}$$

where $f(\cdot|\cdot)$ denotes the conditional probability density function of $\mathbf{Z}_k$ and $\mathcal{N}_\ell$ denotes the numerator term.

From the i.i.d. assumption on the statistics of the observations, the first term within the summation for $\mathcal{N}_\ell$ can be written as:

$$\begin{aligned} f\left(\mathbf{Z}_k | I_{k-1}, U_{k,\ell,m}\right) &= \prod_{j=1}^{\ell-1} f_1(Z_{k,j}) \prod_{j=\ell}^{L} f_0(Z_{k,j}) \\ &= \prod_{j=1}^{\ell-1} L_{k,j} \prod_{j=1}^{L} f_0(Z_{k,j}) \end{aligned}$$

where $L_{k,j} \triangleq \frac{f_1(Z_{k,j})}{f_0(Z_{k,j})}$ is the likelihood ratio of the two hypotheses given that $Z_{k,j}$ is observed at the j-th sensor at the k-th instant. For the second term, observe from the definitions that

$$P(U_{k,\ell,m} | I_{k-1}) = P(T_{k-1,m} | I_{k-1}) \frac{P(U_{k,\ell,m})}{P(T_{k-1,m})}.$$

Thus, we have

$$\begin{aligned}
\mathcal{N}_\ell &= \left(\sum_{m=1}^{\ell} \frac{P(U_{k,\ell,m})}{P(T_{k-1,m})} p_{k-1,m} \right) \prod_{m=1}^{\ell-1} L_{k,m} \prod_{m=1}^{L} f_0(Z_{k,m}) \\
&\triangleq \left(\sum_{m=1}^{\ell} w_{k,\ell,m}\, p_{k-1,m} \right) \mathbf{\Phi}_{\mathsf{obs}}(k,\ell) \qquad (10.2)
\end{aligned}$$

where the first part is a weighted sum of $p_{k-1,m}$ with weights decided by the prior probabilities, and the second part of the evolution equation, $\mathbf{\Phi}_{\mathsf{obs}}(k,\ell)$, can be viewed as that part that depends only on the observation $\mathbf{Z}_k$.

Many observations are in order at this stage:

- The above expansion for $\mathcal{N}_\ell$ can be explained intuitively: If the maximal sensor index observing the change by time k is $\ell-1$, then the maximal sensor index observing the change by time $k-1$ should be from the set $\{0,\cdots,\ell-1\}$.

- Using the joint-geometric model for $\{\Gamma_\ell\}$, it can be shown that $w_{k,\ell,m}$ is of the form:

$$\begin{aligned}
w_{k,\ell,m} &= \frac{P(U_{k,\ell,m})}{P(T_{k-1,m})} \\
&= (1-\rho_{\ell-1,\ell}) \cdot \prod_{j=m-1}^{\ell-2} \rho_{j,j+1} \\
&\triangleq (1-\rho_{\ell-1,\ell}) \cdot w_m^\ell \\
\mathcal{N}_\ell &= \prod_{m=1}^{\ell-1} L_{k,m} \prod_{m=1}^{L} f_0(Z_{k,m}) \cdot (1-\rho_{\ell-1,\ell}) \left(\sum_{m=1}^{\ell} p_{k-1,m} \cdot w_m^\ell \right) \qquad (10.3)
\end{aligned}$$

  with the understanding that the product term in the definition of w_m^ℓ is vacuous (and is to be replaced by 1) if $m = \ell$. It is important to note that the joint-geometric assumption renders the weights ($w_{k,\ell,m}$) associated with $p_{k-1,m}$ independent of k. This will be useful later in establishing convergence properties for the DP.

- It is important to note that given a fixed value of ℓ, $p_{k,\ell}$ is dependent on the entire vector $\boldsymbol{p}_{k-1}$ and not on $p_{k-1,\ell}$ alone. Thus, the recursion for $\mathcal{N}_\ell$ implies that $\boldsymbol{p}_k$ forms the sufficient statistic and the function $\widetilde{J}_k^T(I_k)$ can be written as a function of only $\boldsymbol{p}_k$, say $J_k^T(\boldsymbol{p}_k)$. The finite-horizon DP equations can then be rewritten as

$$\begin{aligned}
J_T^T(\boldsymbol{p}_T) &= p_{T,1} \\
J_k^T(\boldsymbol{p}_k) &= \min \Big\{ p_{k,1},\ c(1-p_{k,1}) + A_k^T(\boldsymbol{p}_k) \Big\}
\end{aligned}$$

  with

$$\begin{aligned}
A_k^T(\boldsymbol{p}_k) &\triangleq E[J_{k+1}^T(\boldsymbol{p}_{k+1})|I_k] \\
&= \int \left[J_{k+1}^T\left(\boldsymbol{p}_{k+1}\right) f\left(\mathbf{Z}_{k+1}|I_k\right)\right] \Big|_{\mathbf{Z}_{k+1}=\boldsymbol{z}} d\boldsymbol{z}.
\end{aligned}$$

  Note that the previously established recursion for $\boldsymbol{p}_{k+1}$ implies that $\boldsymbol{p}_{k+1} = g(\boldsymbol{p}_k, \mathbf{Z}_{k+1})$ for an appropriate choice of $g(\cdot,\cdot)$ (the precise form of $g(\cdot,\cdot)$ is clear from equations (10.1) and (10.2)) which ensures that the right-hand side is indeed a function of $\boldsymbol{p}_k$.

- It is easy to check that the general framework reduces to the special case when all the change-points coincide with Γ_1. In this case, as in Veeravalli [181], we define $p_k \triangleq P\big(\{\Gamma_1 \leq k\} \big| I_k\big)$. Note that only $T_{k,1}$ and $T_{k,L+1}$ are non-empty sets with

$$T_{k,1} = \{\Gamma_1 \geq k+1\}, \quad T_{k,L+1} = \{\Gamma_1 \leq k\},$$
$$p_{k,L+1} = p_k, \, p_{k,1} = 1 - p_k, \, p_{k,\ell} = 0, \, \ell = 2, \cdots, L.$$

Furthermore, the recursion for p_k reduces to

$$\begin{aligned} p_k &= \frac{\mathcal{N}}{\prod_{j=1}^{L} f_0(Z_{k,j}) \, (1 - p_{k-1}) \, (1-\rho) + \mathcal{N}} \\ \mathcal{N} &= \prod_{j=1}^{L} f_1(Z_{k,j}) \, ((1 - p_{k-1})\rho + p_{k-1}) \end{aligned}$$

which coincides with Veeravalli [181, eqn. (13)-(15)]. This case can also be obtained from the formula in (10.3) by setting $\rho_{\ell-1,\ell} = 1$ for all ℓ with $2 \leq \ell \leq L$.

1.3. Structure of the Optimal Stopping Rule (τ_{opt})

The goal of this section is to study the structure of the optimal stopping rule, τ_{opt}. For this, we follow the same outline as in [22] and study the infinite-horizon version of the DP problem by letting $T \rightarrow \infty$.

Theorem 1.1. *Let $\boldsymbol{p} = [p_1, \cdots, p_{L+1}]$ be an element of the standard L-dimensional simplex $\mathcal{P}$, defined as, $\mathcal{P} \triangleq \big\{\boldsymbol{p} \, : \, \sum_{j=1}^{L+1} p_j = 1\big\}$. The infinite-horizon cost-to-go for the DP is of the form*

$$J(\boldsymbol{p}) = \min \Big\{ p_1, \; c(1-p_1) + A_J(\boldsymbol{p}) \Big\},$$

where the function $A_J(\boldsymbol{p})$: i) is concave in $\boldsymbol{p}$ over $\mathcal{P}$; ii) is bounded as $0 \leq A_J(\boldsymbol{p}) \leq 1$; and iii) satisfies $A_J(\boldsymbol{p}) = 0$ over the hyperplane $\big\{\boldsymbol{p} : p_1 = 0\big\}$.

Proof. See Raghavan and Veeravalli [133]. □

At this stage, it is a straightforward consequence that the optimal stopping rule is of the form

$$\tau_{\mathsf{opt}} = \inf_k \Big\{ p_{k,1}(1+c) - c < A_J(\boldsymbol{p}_k) \Big\}.$$

That is, a change is declared when the hyperplane on the left side is exceeded by $A_J(\boldsymbol{p}_k)$ and no change is declared, otherwise. We will next see that this test characterization reduces to a degenerate one as $\rho \rightarrow 0$.

To establish this degeneracy, we define the following one-to-one and invertible transformation:

$$q_{k,\ell} = \frac{p_{k,\ell}}{\rho \, p_{k,1}} \Longleftrightarrow p_{k,\ell} = \frac{q_{k,\ell}}{\sum_{j=1}^{L+1} q_{k,j}}, \; \ell = 1, \cdots, L+1$$

which is equivalent to

$$\begin{aligned} p_{k,1} &= \frac{1}{1 + \rho \sum_{j=2}^{L+1} q_{k,j}} \quad \text{and} \\ p_{k,\ell} &= \frac{\rho \, q_{k,\ell}}{1 + \rho \sum_{j=2}^{L+1} q_{k,j}}, \; \ell = 2, \cdots, L+1. \end{aligned}$$

We can write $q_{0,\ell}$ in terms of the priors as

$$\begin{aligned} q_{0,1} &= \frac{p_{0,1}}{\rho p_{0,1}} = \frac{1}{\rho}, \\ q_{0,\ell} &= \frac{p_{0,\ell}}{\rho p_{0,1}} = \frac{P\big(\{\Gamma_1 = \cdots = \Gamma_{\ell-1} = 0, \Gamma_\ell > 0\}\big)}{\rho P\big(\{\Gamma_1 > 0\}\big)} \\ &= \frac{\prod_{j=0}^{\ell-2} \rho_{j,j+1}\,(1-\rho_{\ell-1,\ell})}{\rho(1-\rho)}, \;\; \ell = 2, \cdots, L+1. \end{aligned}$$

Note that while $p_{k,\ell}$ are conditional probabilities of certain events, and hence, lie in the interval $[0,1]$, the range of $q_{k,\ell}$ is in general $[0,\infty)$.

It can be checked that the evolution equation can be rewritten in terms of $q_{k,\ell}$ as

$$q_{k,\ell} = \frac{1-\rho_{\ell-1,\ell}}{1-\rho} \cdot \prod_{j=1}^{\ell-1} L_{k,j} \cdot \left(\sum_{j=1}^{\ell} q_{k-1,j} w_j^{\ell} \right). \tag{10.4}$$

It is interesting to note from (10.4) that the update for $q_{k,\ell}$ is a weighted sum of $q_{k-1,j}, j = 1, \cdots, \ell$ with progressively decreasing weight as j increases. Similarly, we can define $J_k^T(\cdot)$ and $A_k^T(\cdot)$ in terms of $\bar{b}q_k$. Using the transformation $\{q_{k,\ell}\}$, τ_{opt} is seen to have the form:

$$\boxed{\tau_{\mathsf{opt}} = \inf_k \left\{ \sum_{\ell=2}^{L+1} q_{k,\ell} > \frac{1 - A_J(\bar{b}q_k)}{\rho\left(c + A_J(\bar{b}q_k)\right)} \right\}.}$$

When all Γ_ℓ coincide, we have

$$\begin{aligned} q_{k,L+1} &= \frac{p_k}{\rho(1-p_k)} \triangleq q_k, \\ q_{k,1} &= \frac{1}{\rho}, \;\; q_{k,\ell} = 0, \;\; \ell = 2, \cdots, L. \end{aligned}$$

Further, it is straightforward to check that the evolution in (10.4) reduces to

$$q_{k,L+1} = \frac{\prod_{j=1}^{L} L_{k,j}}{1-\rho} \cdot (1 + q_{k-1,L+1}), \tag{10.5}$$

Thus, the space of sufficient statistics and the optimal test reduce to a one-dimensional variable ($p_k = P\big(\{\Gamma_1 \leq k\}\big|I_k\big)$ or equivalently, q_k) and a threshold test on p_k (or equivalently, on q_k), respectively. In the general case, unless something more is known about the structure of $A_J(\cdot)$ (which is possible if there is some structure on $\{\rho_{\ell-1,\ell}\}$), we cannot say more about τ_{opt}. Nevertheless, the following theorem establishes its structure in the practical setting of a rare disruption regime ($\rho \to 0$). The limiting test thresholds (from below) the *a posteriori* probability that no-change has happened, and is denoted as ν_A.

Theorem 1.2. *The test structure corresponding to τ_{opt} converges in probability to a simple threshold operation in the asymptotic limit as $\rho \to 0$. This limiting test is of the form:*

$$\boxed{\nu_A = \begin{cases} \text{Stop} & \text{if } \log\left(\sum_{\ell=2}^{L+1} q_{k,\ell}\right) \geq A \\ \text{Continue} & \text{if } \log\left(\sum_{\ell=2}^{L+1} q_{k,\ell}\right) < A \end{cases}}$$

for an appropriate choice of threshold A.

Proof. See Raghavan and Veeravalli [133]. □

The test ν_A is of low-complexity because of the following properties: i) a simple recursion formula (10.4) for the sufficient statistics; ii) a threshold operation for stopping; and iii) the threshold value that can be pre-computed given the PFA constraint (see Prop. 1.3).

The fact that $\tau_{\mathsf{opt}} \overset{\rho \downarrow 0}{\to} \nu_A$ for an appropriate choice of A *does not* imply that ν_A is asymptotically (as $\mathsf{PFA} \to 0$ or as $\rho \to 0$) optimal. However, the low-complexity of this test, in addition to Theorem 1.2, and the fact that the structure of $A_J(\bar{b}q_k)$ (and hence, τ_{opt}) are not known suggest that it is a good candidate test for change detection across a sensor array. In fact, we will see this to be the case when we establish sufficient conditions under which ν_A is asymptotically optimal.

1.4. Main Results on ν_A

Towards this end, our main interest is in understanding the performance (ADD vs. PFA) of ν_A for any general choice of threshold A.

Special Cases of Change Parameters: To build intuition, we start by considering some special scenarios of change propagation modeling. The first scenario corresponds to the case where one (or more) of the $\rho_{\ell-1,\ell}$ is 1. The following proposition addresses this setting.

Proposition 1.1. *Consider an L-sensor system described in Sec. 1.1, parameterized by $\{\rho_{\ell-1,\ell}\}$, where $\rho_{\ell',\ell'+1} = 1$ for some ℓ' and $\max\limits_{j \neq \ell'} \rho_{j,j+1} < 1$. This system is equivalent to an $(L-1)$-sensor system, parameterized by $\{\delta_{\ell,\ell+1}\}$, where*

$$\begin{aligned} \delta_{j,j+1} &= \rho_{j,j+1}, \ j \leq \ell' - 1 \\ \delta_{j,j+1} &= \rho_{j+1,j+2}, \ j \geq \ell' \end{aligned}$$

with the $(\ell'+1)$-th sensor observing (a combination of) $Z_{k,\ell'+1}$ and $Z_{k,\ell'+2}$ with a geometric delay parameter of $\delta_{\ell',\ell'+1} = \rho_{\ell'+1,\ell'+2}$.

Proof. The proof is straightforward by studying the evolution of $\{q_{k,\ell}\}$ for the original L-sensor system. From (10.4), it can be seen that $q_{k,\ell'+1} = 0$ (identically) for all k and the reduced $(L-1)$-dimensional system discards this redundant information, while the observation corresponding to the $(\ell'+1)$-th sensor is carried over to the $(\ell'+2)$-th original sensor. □

The second scenario corresponds to the case where one (or more) of the $\rho_{\ell-1,\ell}$ is 0.

Proposition 1.2. *Consider an L-sensor system, parameterized by $\{\rho_{\ell-1,\ell}\}$, with ℓ' indicating the smallest index such that $\rho_{\ell',\ell'+1} = 0$. This system is equivalent to an ℓ'-sensor system with the same parameters as that of the original system. It is as if sensors $(\ell'+1)$ and beyond do not exist (or contribute) in the context of change detection.*

Proof. The proof is again straightforward by considering the evolution of $\{q_{k,\ell}\}$ in (10.4) and noting that $q_{k,j}, \ j \geq \ell' + 2$ are identically 0 for all k. □

It is useful to interpret Props. 1.1 and 1.2 via an "information flow" paradigm. If change propagation is instantaneous across a sensor (corresponding to the first case), it is as if the fusion center is *oblivious* to the presence of that sensor conditioned upon the previous sensors' observations. In this setting, the detection delay corresponding to that sensor is zero, as would be expected from the fact that the geometric parameter is 1. In the second case, information flow to the fusion center (concerning change) is *cut-off or blocked* past the first sensor with a geometric parameter of 0. That is, the observations made by sensors $\{\ell'+1, \cdots, L\}$ (if any) do not contribute information to the fusion center in helping it decide whether the disruption has happened or not. Apart from these extreme cases of oblivious/blocking sensors, we can assume without loss in generality that

$$0 < \min_{\ell} \rho_{\ell-1,\ell} \leq \max_{\ell} \rho_{\ell-1,\ell} < 1.$$

Continuity arguments suggest that if some $\rho_{\ell-1,\ell}$ is small (but non-zero), it should be natural to expect that the ℓ-th sensor and beyond *may not* "effectively" contribute any information to the fusion center. We will interpret this observation after establishing performance bounds for ν_A.
Probability of False Alarm: We first show that letting $A \to \infty$ in ν_A corresponds to considering the regime where $\mathsf{PFA} \to 0$.

Proposition 1.3. *The probability of false alarm with ν_A can be upper bounded as*

$$\mathsf{PFA} \leq \frac{1}{1 + \rho \cdot \exp(A)}.$$

That is, if $\alpha \leq 1$ and the threshold A is set as $A = \log\left(\frac{1}{\rho\alpha}\right)$, then $\mathsf{PFA} \leq \alpha$.

Proof. See Raghavan and Veeravalli [133]. □

Universal Lower Bound on ADD***:*** We now establish a lower bound on ADD for the class of stopping times $\boldsymbol{\Delta}_\alpha$. That is, any stopping time τ should have an ADD larger than the lower bound if PFA is to be smaller than α.

Proposition 1.4. *Consider the class of stopping times $\boldsymbol{\Delta}_\alpha = \{\tau : \mathsf{PFA}(\tau) \leq \alpha\}$. Under the assumption that $\min\limits_{\ell=2,\cdots,L} \rho_{\ell-1,\ell} > 0$, we have*

$$\inf_{\tau \in \boldsymbol{\Delta}_\alpha} \mathsf{ADD}(\tau) \geq \frac{|\log(\alpha)| \cdot (1 + \mathrm{o}(1))}{LD(f_1, f_0) + |\log(1-\rho)|} \text{ as } \alpha \to 0$$

where the $\mathrm{o}(1)$ term converges to zero as $\alpha \to 0$.

Proof. See Raghavan and Veeravalli [133]. □

Upper Bound on ADD ***of*** ν_A***:*** We will now establish an upper bound on ADD of ν_A.

Theorem 1.3. *Let $\{\rho_{\ell-1,\ell}\}$ be such that $0 < \min\limits_{\ell} \rho_{\ell-1,\ell} \leq \max\limits_{\ell} \rho_{\ell-1,\ell} < 1$. Further, assume that $D(f_1, f_0)$ be such that there exists some j satisfying $\ell \leq j \leq L$ and*

$$D(f_1, f_0) > \frac{1}{j - \ell + 1} \log\left(\frac{\sum_{p=0}^{\ell-1}(1 - \rho_{p,p+1})}{1 - \rho_{j,j+1}}\right), \tag{10.6}$$

for all $2 \leq \ell \leq L$. Then, the performance of ν_A with $A = \log\left(\frac{1}{\rho\alpha}\right)$ is given by

$$\boxed{E\left[\nu_A\right] \leq \frac{|\log(\rho\alpha)| \cdot (1 + \mathrm{o}(1))}{LD(f_1, f_0) + |\log\left(1 - \rho\right)|} \text{ as } \alpha \to 0.}$$

Corollary 1.1. *Combining Prop. 1.4 and Theorem 1.3, it can be seen that ν_A is asymptotically optimal (as $\alpha \to 0$) for any fixed $\rho > 0$. In other words,*

$$\inf_{\tau \in \boldsymbol{\Delta}_\alpha} \mathsf{ADD}(\tau) \sim E\left[\nu_A\right]$$

where we have used the notation $X_\alpha \sim Y_\alpha$ as $\alpha \to 0$ to mean $\lim\limits_{\alpha \to 0} \frac{X_\alpha}{Y_\alpha} = 1$.

The proof of Theorem 1.3 in the general case of an arbitrary number (L) of sensors with an arbitrary choice of $\{\rho_{\ell-1,\ell}\}$ results in cumbersome analysis. Hence, it is worthwhile to consider the special case of two sensors that can be captured by just two change parameters: ρ and $\rho_{1,2}$. The main idea that is necessary in tackling the general case is easily exposed in the $L = 2$ setting in the following result. The general case is carefully studied in Raghavan and Veeravalli [133]..

Proposition 1.5. ($L = 2$) *The stopping time ν_A is such that $\nu_A \to \infty$ as $A \to \infty$. Further, if $D(f_1, f_0)$ satisfies*

$$D(f_1, f_0) > \log\left(2 - \rho - \rho_{1,2}\right),$$

we also have

$$\lim_{A\to\infty} \frac{E[\nu_A]}{A} \leq \frac{1}{2D(f_1, f_0) + |\log\left(1-\rho\right)|}.$$

1.5. Discussion and Numerical Results

Discussion: A loose sufficient condition for all the L sensors to contribute to the slope of ADD of ν_A is that

$$D(f_1, f_0) > \max_{\ell=1,\cdots,L-1} \ \min_{j\geq \ell+1} \frac{1}{j-\ell} \cdot \log\left(\frac{\sum_{p=0}^{\ell}(1-\rho_{p,p+1})}{1-\rho_{j,j+1}}\right) \triangleq \quad \gamma_u.$$

Another sufficient condition is that

$$D(f_1, f_0) > \max_{\ell=1,\cdots,L-1} \frac{1}{L-\ell} \cdot \log\left(1 - \rho + \sum_{j=1}^{\ell}(1-\rho_{j,j+1})\right).$$

That is, if ρ is such that

$$\rho \geq \sum_{\ell=2}^{L}(1-\rho_{\ell-1,\ell}),$$

then $\gamma_u \leq 0$ and the condition of Theorem 1.3 reduces to a mild one that the K-L divergence between f_1 and f_0 be positive. A special setting where the above condition is true (irrespective of the rarity of the disruption-point) is the regime where change propagates across the sensor array "quickly." The case instantaneous propagation is an extreme example of this regime and Theorem 1.3 recaptures this extreme case.

In more general regimes where change propagates across the sensor array "slowly", either the disruption-point should become less rare (independent of the choice of f_1 and f_0) or that the densities f_1 and f_0 be sufficiently discernible (independent of the rarity of the disruption-point) so that all the L sensors can contribute to the asymptotic slope. When these conditions fail to hold, it is not clear whether the theorems are applicable, or even if all the L sensors contribute to the slope of $E[\nu_A]$. Nevertheless, it is reasonable to conjecture that as long as $\min_{\ell} \rho_{\ell-1,\ell} > 0$, then all the L sensors contribute to the asymptotic slope.

However, the difference between the asymptotic and the non-asymptotic regimes needs a careful revisit. Following the initial remark (Prop. 1.2) on the extreme case of blocking sensors (where some $\rho_{\ell-1,\ell} = 0$), in the more realistic case where some $\rho_{\ell-1,\ell}$ may be small (but non-zero), it is possible that if $D(f_1, f_0)$ is smaller than some threshold value (determined by the change propagation parameters), not all of the L sensors may "effectively" contribute to the slope of ADD, at least for reasonably small, but non-asymptotic values of PFA. For example, see the ensuing discussion where numerical results illustrate this behavior at PFA values of 10^{-4} to 10^{-5} for some choice of change propagation parameters, *even* when the condition in Theorem 1.3 is met. When the condition in Theorem 1.3 is not met, such a behavior is expected to be more typical.

Numerical Study I – Performance Improvement with ν_A: Given that the structure of τ_{opt} is not known in closed-form, we now present numerical studies to show that ν_A results in substantial improvement in performance over both a single sensor test (which uses the observations only from the first sensor and ignores the other sensor observations) and a test that uses the observations from all the sensors but under a mismatched model (where the change-point for all the sensors is assumed to be the same), even under realistic modeling assumptions.

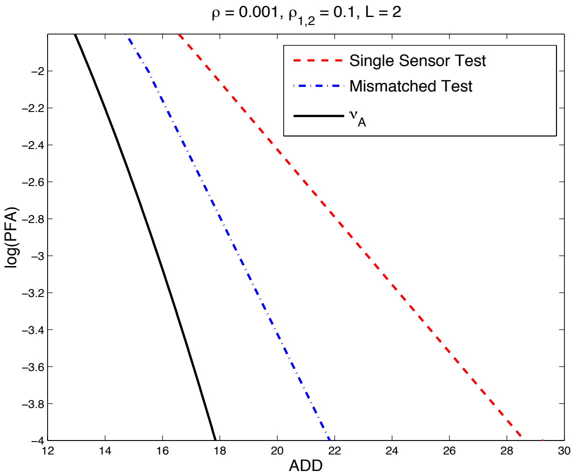


Figure 10.2: Probability of false alarm vs. Average detection delay for a $L = 2$ setting with $\rho = 0.001$ and $\rho_{1,2} = 0.1$.

The first example corresponds to a two sensor system where the occurrence of change is modeled as a geometric random variable with parameter $\rho = 0.001$. Change propagates from the first sensor to the second with the geometric parameter $\rho_{1,2} = 0.1$. The pre- and post-change densities are $\mathcal{CN}(0, 1)$ and $\mathcal{CN}(1, 1)$, respectively so that $D(f_1, f_0) = 0.50$. While the threshold for ν_A is set as in Prop. 1.3, the thresholds for the single sensor and mismatched tests are set as in [171]. The recursion for the sufficient statistic of the mismatched test follows the description in Veeravalli [181]. Figure 10.2 depicts the performance of the three tests obtained via Monte Carlo methods and shows that ν_A can result in an improvement of at least 4 units of delay at even marginally large PFA values on the order of 10^{-3}.

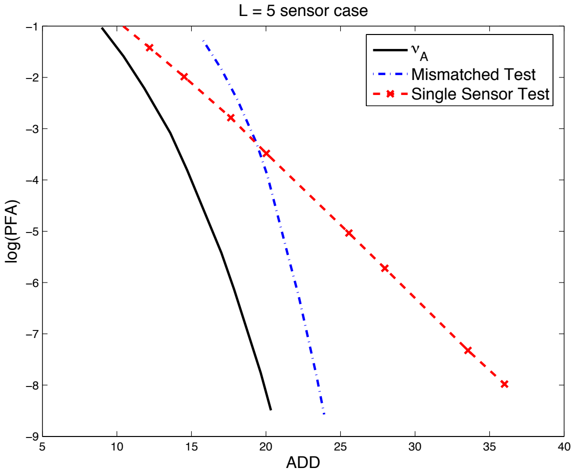
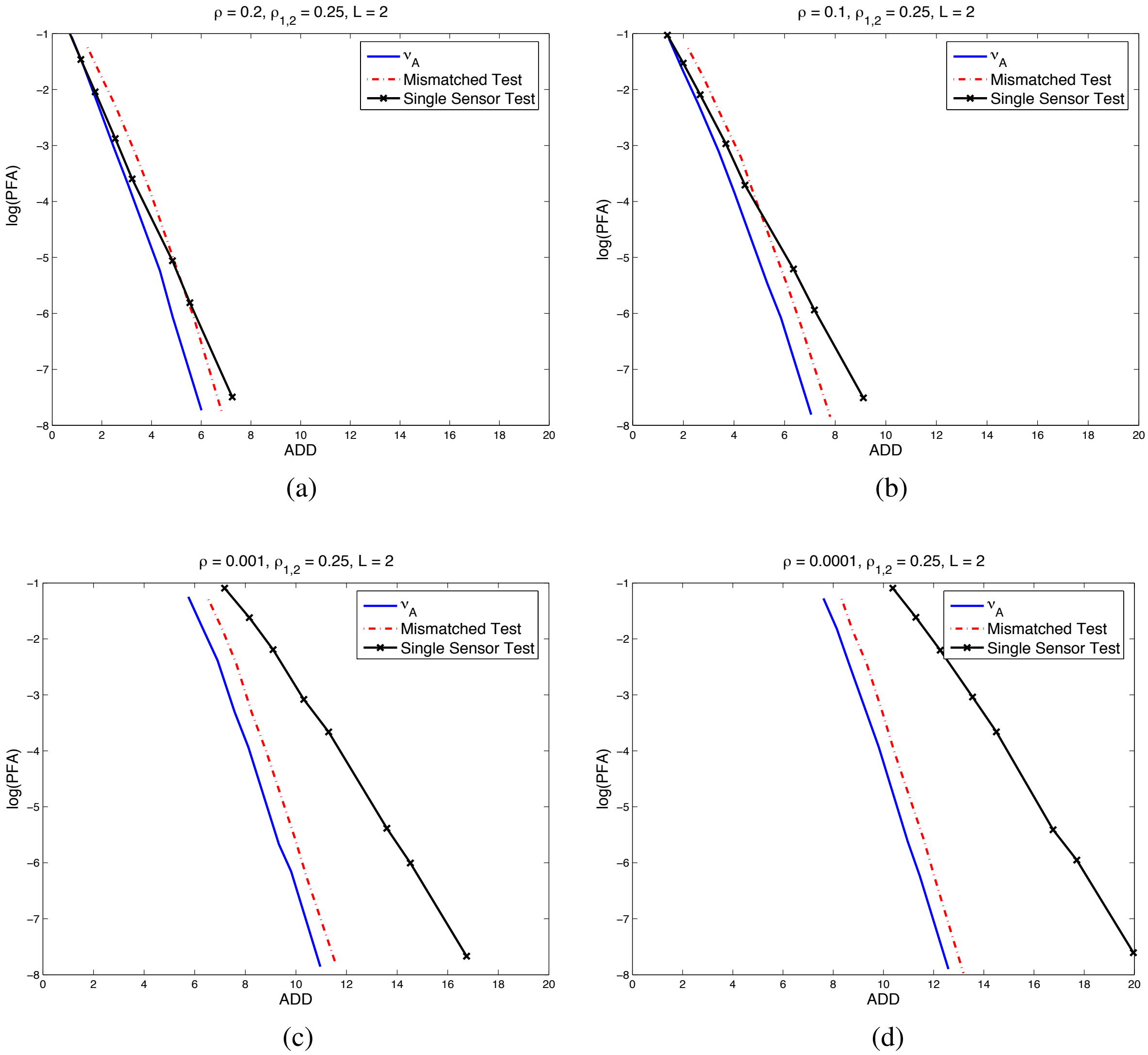


Figure 10.3: Probability of false alarm vs. Average detection delay for a typical $L = 5$ setting.

The second example corresponds to a five sensor system where $\rho = 0.005$. Change propagates across the array according to the following model: $\rho_{1,2} = 0.1, \rho_{2,3} = 0.2, \rho_{3,4} = 0.5$ and $\rho_{4,5} = 0.7$. The pre-

and the post-change densities are $\mathcal{CN}(0,1)$ and $\mathcal{CN}(0.75,1)$ so that $D(f_1, f_0) \approx 0.2813$. With $D(f_1, f_0)$ and the change parameters as above, Theorem 1.3 assures us that at least $L = 2$ sensors contribute to the ADD vs. PFA slope asymptotically. On the other hand, Figure 10.3 shows that more than two sensors indeed contribute to the slope. Thus, it can be seen that Theorem 1.3 provides only a sufficient condition on performance bounds. It is also worth noting the transition in slope (unlike the case in Veeravalli [181]) for both the mismatched test and ν_A as PFA decreases from moderately large values to zero, whereas the slope of the single sensor test (as expected) remains constant.

Numerical Study II – Performance Gap Between the Tests: We now present a second case-study with the main goal being the understanding of the relative performance of ν_A with respect to the single sensor and the mismatched tests. We again consider a $L = 2$ sensor system and we vary the change process parameters, ρ and $\rho_{1,2}$, in this study. The pre- and the post-change densities are $\mathcal{CN}(0,1)$ and $\mathcal{CN}(1.2,1)$ so that $D(f_1, f_0) = 0.72$.



(a)

(b)

(c)

(d)

Figure 10.4: Probability of false alarm vs. Average detection delay for a $L = 2$ setting with different model parameters.

Figure 10.4 and Figure 10.5(b) show the performance of the three tests with varying ρ parameters for a fixed choice of $\rho_{1,2}$. We observe that the gap in performance between the single sensor test and ν_A increases as ρ decreases, whereas the gap between ν_A and the mismatched test stays fairly constant. Similarly, Figure 10.5 shows the performance of the three tests with varying $\rho_{1,2}$ parameters for a fixed choice of ρ.

We observe from these plots that the gap between the mismatched test and ν_A increases as $\rho_{1,2}$ decreases, whereas the gap between the single sensor test and ν_A increases as $\rho_{1,2}$ increases.

The choice of $D(f_1, f_0) = 0.72$ is such that the sufficient condition in Theorem 1.3 are satisfied, independent of the change parameters. Hence, we expect the slope of the ADD vs. PFA plot to be of the form $\frac{1}{2D(f_1,f_0)+|\log(1-\rho)|}$ *asymptotically* as $\mathsf{PFA} \to 0$. Nevertheless, Figure 10.5(c) and (d) show that, when both ρ and $\rho_{1,2}$ are small, the slope of ν_A is only as good as (or slightly better than) the single sensor test, which is known to have a slope of the form $\frac{1}{D(f_1,f_0)+|\log(1-\rho)|}$. Thus, we see that even though our theory guarantees that both the sensors' observations contribute in the eventual performance of ν_A *asymptotically*, we may not see this behavior for reasonable choices of PFA such as 10^{-4}. The case of observation models not meeting the conditions of Theorem 1.3 is expected to show this trend for even lower PFA values.

To summarize these observations, if ADD_{ν_A}, $\mathsf{ADD}_{\mathsf{MM}}$ and $\mathsf{ADD}_{\mathsf{SS}}$ denote the average detection delays for ν_A, mismatched and single sensor tests (respectively) for some fixed choice of PFA, then

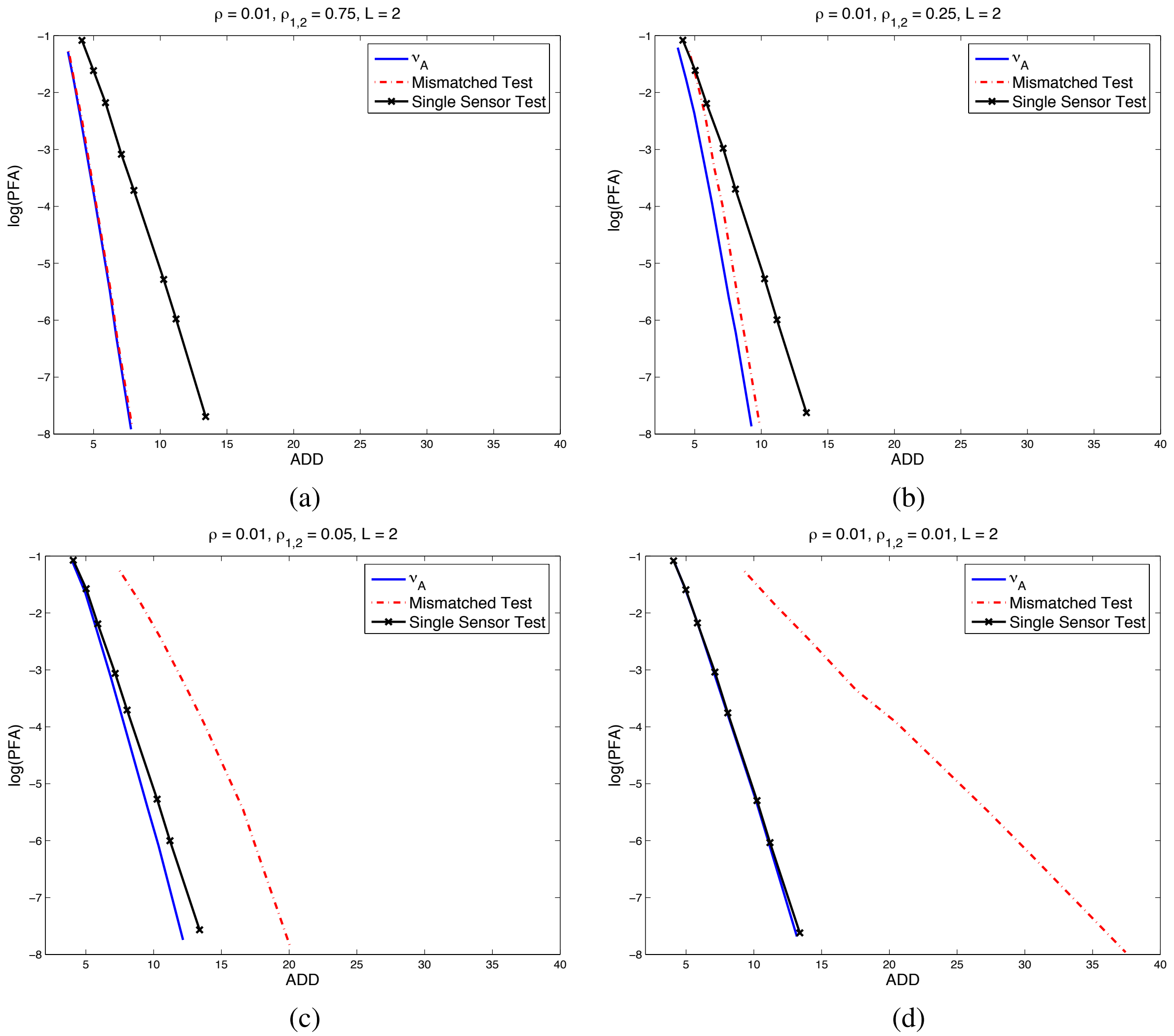
$$\begin{aligned} \mathsf{ADD}_{\mathsf{MM}} - \mathsf{ADD}_{\nu_A} &\propto \frac{1}{\rho_{1,2}} \text{ and independent of } \rho \\ \mathsf{ADD}_{\mathsf{SS}} - \mathsf{ADD}_{\nu_A} &\propto \frac{\rho_{1,2}}{\rho}. \end{aligned}$$

It is interesting to note from the above equations that $\rho_{1,2}$ impacts the gap between the two tests in a contrasting way. The test ν_A is expected to result in significant performance improvement in the regime where ρ is small, but $\rho_{1,2}$ is neither too small nor too large. In fact, this regime where ν_A is expected to result in significant performance improvement is the precise regime that is of importance in practical contexts. This is so because we can expect the occurrence of disruption (e.g., cracks in bridges, intrusions in networks, onset of epidemics, etc.) to be a rare phenomenon. Once the disruption occurs, we expect change to propagate across the sensor array fairly quickly due to the geographical (network proximity in the case of computer networks) proximity of the other sensors, but not so quick that the extreme case of instantaneous propagation is applicable. Classifying the regime of $\{\rho_{\ell-1,\ell}\}$ and $D(f_1, f_0)$ where significant performance improvement is possible with ν_A is ongoing work. It is also of interest to come up with better test structures in the regime where ν_A does not lead to a significant performance improvement.

1.6. Concluding Remarks

We considered the centralized, Bayesian version of the change process detection problem in this work and posed it in the classical dynamic programming framework. This formulation of the change detection problem allows us to establish the sufficient statistics for the DP under study and a recursion for the sufficient statistics. While we obtain the broad structure of the optimal stopping rule (τ_{opt}), any further insights into it are rendered infeasible by the complicated nature of the infinite-horizon cost-to-go function. Nevertheless, τ_{opt} reduces to a threshold rule (denoted in this work as ν_A) in the rare disruption regime.

The test ν_A possesses the following properties and thus serves as an attractive test for practical applications that can be modeled with a change process: i) it is of low-complexity; ii) under certain mild sufficient conditions (more specifically, if the K-L divergence $D(f_1, f_0)$ is more than a number determined by the parameters of the change process), it is asymptotically optimal in the small PFA regime; and iii) numerical studies suggest that it can lead to substantially improved performance over naive tests. Nevertheless, the asymptotic expansion of ADD in terms of $\log(\mathsf{PFA})$ is not enough to determine how small the false alarm probability should be in order for this expansion and asymptotic optimality of ν_A to hold. Studies indicate that PFA should be chosen significantly smaller than those needed for good approximations in the simpler quickest detection problems solved earlier by the same approach.



(a)

(b)

(c)

(d)

Figure 10.5: Probability of false alarm vs. Average detection delay for a $L = 2$ setting with different model parameters.

2. Data-Efficient Quickest Change Detection with On-Off Observation Control

2.1. Introduction

In the Bayesian quickest change detection problem proposed by Shiryaev Shiryaev [147], there is a sequence of random variables, $\{X_n\}$, whose distribution changes at a random time Γ. It is assumed that before Γ, $\{X_n\}$ are independent and identically distributed (i.i.d.) with density f_0, and after Γ they are i.i.d. with density f_1. The distribution of Γ is assumed to be known and modeled as a geometric random variable with parameter ρ. The objective is to find a stopping time τ, at which time the change is declared, such that the average detection delay is minimized subject to a constraint on the probability of false alarm.

In this paper we extend Shiryaev's formulation by explicitly accounting for the cost of the observations used in the detection process. We capture the observation penalty (cost) through the average number of observations used before the change point Γ, and allow for a dynamic control policy that determines whether or not a given observation is taken. The objective is to choose the observation control policy along with the stopping time τ, so that the average detection delay is minimized subject to constraints on the probability of false alarm and the observation cost.

2.2. Problem Formulation and the Two-threshold Algorithm

As in the model for the classical Bayesian quickest change detection problem described in Section 2.1, we have a sequence of random variables $\{X_n\}$, which are i.i.d. with density f_0 before the random change point Γ, and i.i.d. with density f_1 after Γ. The change point Γ is modeled as geometric with parameter ρ, i.e., for $0 < \rho < 1,\ 0 \le \pi_0 < 1$,

$$\pi_k = \mathsf{P}\{\Gamma = k\} = \pi_0 \ 1\!\!1_{\{k=0\}} + (1-\pi_0)\rho(1-\rho)^{k-1} \ 1\!\!1_{\{k\ge 1\}},$$

where $1\!\!1$ is the indicator function, and π_0 represents the probability of the change having happened before the observations are taken. Typically π_0 is set to 0.

In order to minimize the average number of observations used before Γ, at each time instant, a decision is made on whether to use the observation in the next time step, based on all the available information. Let $S_k \in \{0,1\}$, with $S_k = 1$ if it is been decided to take the observation at time k, i.e. X_k is available for decision making, and $S_k = 0$ otherwise. Thus, S_k is an on-off (binary) control input based on the information available up to time $k-1$, i.e.,

$$S_k = \mu_{k-1}(I_{k-1}), \quad k = 1, 2, \ldots$$

with μ denoting the control law and I defined as:

$$I_k = \left[S_1, \ldots, S_k, X_1^{(S_1)}, \ldots, X_k^{(S_k)}\right].$$

Here, $X_i^{(S_i)}$ represents X_i if $S_i = 1$, otherwise X_i is absent from the information vector I_k. The choice of S_1 is based on the prior π_0.

As in the classical change detection problem, the end goal is to choose a stopping time on the observation sequence at which time the change is declared. Denoting the stopping time by τ, we can define the average detection delay (ADD) as

$$\text{ADD} = \mathrm{E}\left[(\tau - \Gamma)^+\right].$$

Further, we can define the probability of false alarm (PFA) as

$$\text{PFA} = \mathrm{P}(\tau < \Gamma).$$

The new performance metric for our problem is the average number of observations (ANO) used before Γ in detecting the change:

$$\text{ANO} = \mathrm{E}\left[\sum_{k=1}^{\min\{\tau,(\Gamma-1)\}} S_k\right].$$

Let $\gamma = \{\tau, \mu_0, \ldots, \mu_{\tau-1}\}$ represent a policy for cost-efficient quickest change detection. We wish to solve the following optimization problem:

$$\begin{aligned} &\underset{\gamma}{\text{minimize}} && \text{ADD}(\gamma), \\ &\text{subject to} && \text{PFA}(\gamma) \le \alpha, \text{ and } \text{ANO}(\gamma) \le \beta, \end{aligned} \tag{10.7}$$

where α and β are given constraints. Towards solving (10.7), we consider a Lagrangian relaxation of this problem which can be approached using dynamic programming.

The Lagrangian relaxation of the optimization problem in (10.7) is,

$$R(\gamma) = \min_{\gamma} \text{ADD}(\gamma) + \lambda_f \ \text{PFA}(\gamma) + \lambda_e \ \text{ANO}(\gamma), \tag{10.8}$$

where λ_f and λ_e are Lagrange multipliers. It is easy to see that if λ_f and λ_e can be found such that the solution to (10.8) achieves the PFA and ANO constraints with equality, then the solution to (10.8) is also the solution to (10.7).

The problem in (10.8) can be converted to an appropriate Markov control problem using steps similar to those followed in [131].

Let Θ_k denote the state of the system at time k. After the stopping time τ it is assumed that the system enters a terminal state $\mathcal{T}$ and stays there. For $k < \tau$, we have $\Theta_k = 0$ for $k < \Gamma$, and $\Theta_k = 1$ otherwise. Then we can write

$$\text{ADD} = \mathsf{E}\left[\sum_{k=0}^{\tau-1} \mathbb{1}_{\{\Theta_k=1\}}\right]$$

and PFA $= \mathsf{E}[\mathbb{1}_{\{\Theta_\tau=0\}}]$.

Furthermore, let D_k denote the stopping decision variable at time k, i.e., $D_k = 0$ if $k < \tau$ and $D_k = 1$ otherwise. Then the optimization problem in (10.8) can be written as a minimization of an additive cost over time:

$$R(\gamma) = \min_{\gamma} \mathsf{E}\left[\sum_{k=0}^{\tau} g_k(\Theta_k, D_k, S_k)\right]$$

with

$$g_k(\theta, d, s) = \mathbb{1}_{\{\theta\neq\mathcal{T}\}}\ \left[\mathbb{1}_{\{\theta=1\}}\mathbb{1}_{\{d=0\}} + \lambda_f\ \mathbb{1}_{\{\theta=0\}}\mathbb{1}_{\{d=1\}}\ + \lambda_e\ \mathbb{1}_{\{\theta=0\}}\mathbb{1}_{\{s=1\}}\mathbb{1}_{\{d=0\}}\right].$$

Using standard arguments [21] it can be seen that this optimization problem can be solved using infinite horizon dynamic programming with sufficient statistic (belief state) given by:

$$p_k = \mathsf{P}\{\Theta_k = 1 \mid I_k\} = \mathsf{P}\{\Gamma \leq k \mid I_k\}.$$

Using Bayes' rule, p_k can be shown to satisfy the recursion

$$p_{k+1} = \begin{cases} \Phi^{(0)}(p_k) & \text{if } S_{k+1} = 0 \\ \Phi^{(1)}(X_{k+1}, p_k) & \text{if } S_{k+1} = 1 \end{cases}$$

where

$$\Phi^{(0)}(p_k) = p_k + (1 - p_k)\rho \tag{10.9}$$

and

$$\Phi^{(1)}(X_{k+1}, p_k) = \frac{\Phi^{(0)}(p_k)L(X_{k+1})}{\Phi^{(0)}(p_k)L(X_{k+1}) + (1 - \Phi^{(0)}(p_k))} \tag{10.10}$$

with $L(X_{k+1}) = f_1(X_{k+1})/f_0(X_{k+1})$ being the likelihood ratio, and $p_0 = \pi_0$. Note that the structure of recursion for p_k is independent of time k.

The optimal policy for the problem given in (10.8) can be obtained from the solution to the Bellman equation:

$$J(p_k) = \min_{d_k, s_{k+1}} \lambda_f\ (1-p_k)\mathbb{1}_{\{d_k=1\}}+\mathbb{1}_{\{d_k=0\}}\left[p_k + B_0(p_k)\mathbb{1}_{\{s_{k+1}=0\}} + (\lambda_e(1 - p_k) + B_1(p_k))\mathbb{1}_{\{s_{k+1}=1\}}\right] \tag{10.11}$$

with

$$B_0(p_k) = J(\Phi^{(0)}(p_k))$$

and

$$B_1(p_k) = \mathsf{E}[J(\Phi^{(1)}(X_{k+1}, p_k))].$$

It can be shown by an induction argument (see, e.g., [131]) that J, B_0 and B_1 are all non-negative concave functions on the interval $[0, 1]$, and that $J(1) = B_0(1) = B_1(1) = 0$. Also, by Jensen's inequality

$$B_1(p) \leq J(\mathsf{E}[\Phi^{(1)}(X, p)]) = B_0(p), \quad p \in [0, 1].$$

Let

$$d(p_k) = B_0(p_k) - B_1(p_k).$$

Then, from the above properties of J, B_0 and B_1, it is easy to show that the optimal policy $\gamma^* = (\tau^*, \mu_0^*, \mu_1^*, \ldots, \mu_{\tau-1}^*)$ for the problem given in (10.8) has the following structure:

$$S_{k+1}^* = \mu_k^*(p_k) = \begin{cases} 0 & \text{if } d(p_k) < \lambda_e(1-p_k) \\ 1 & \text{if } d(p_k) \geq \lambda_e(1-p_k) \end{cases} \tag{10.12}$$
$$\tau^* = \inf\left\{k \geq 1 : p_k > A^*\right\}.$$

Remark 2.1. Since, $d(p_k) \geq 0 \;\; \forall p_k$, the algorithm in (10.12) reduces to the classical Shiryaev test when $\lambda_e = 0$ [147].

The optimal stopping rule τ^* is similar to the one of the Shiryaev problem. But, the observation control is not explicit and one has to evaluate the differential cost function $d(p_k)$ at p_k at each time step to decide on the choice of S_{k+1}. However, numerical studies of the Bellman equation in (10.11) shows that, for most choices of ρ, f_0 and f_1, that we have tried, the optimal algorithm in (10.12) has the following two-threshold structure.

Start with $p_0 = 0$ and use the following control, with $B < A$, for $k \geq 0$:

$$S_{k+1} = \mu_k(p_k) = \begin{cases} 0 & \text{if } p_k < B \\ 1 & \text{if } p_k \geq B \end{cases} \tag{10.13}$$
$$\tau = \inf\left\{k \geq 1 : p_k > A\right\}.$$

The probability p_k is updated using (10.9) and (10.10).

From a practical point of view, even if a two-threshold policy (10.13) is not optimal, one would like to use the algorithm for the following reasons. First, the choice of thresholds uniquely determine the probability of false alarm and the average number of observations used before change in $\gamma(A, B)$. Second, apart from being simple, the two-threshold policy (10.13) is asymptotically optimal. In Section 2.3, we provide an asymptotic analysis of $\gamma(A, B)$, and use the analytical results to show in Section 2.9.1 that $\gamma(A, B)$ is asymptotically optimal in the following sense. If

$$\Delta(\alpha, \beta) = \{\gamma : \text{PFA}(\gamma) \leq \alpha; \;\; \text{ANO}(\gamma) \leq \beta\},$$

then for a fixed β and ρ,

$$\text{ADD}(\gamma(A(\alpha,\beta), B(\alpha,\beta))) = \left[\inf_{\gamma \in \Delta(\alpha,\beta)} \text{ADD}(\gamma)\right](1 + o(1)) \text{ as } \alpha \to 0.$$

Here, $g(x) = o(1)$ as $x \to x_0$ is used to denote that $g(x) \to 0$ in the specified limit. Also for each β, $B(\alpha, \beta)$ is the smallest B such that $\text{ANO}(\gamma(A, B(\alpha, \beta))) \leq \beta$ as $A \to 1$, and $A(\alpha, \beta)$ is such that, for a fixed β, $\text{PFA}(\gamma(A(\alpha, \beta), B(\alpha, \beta))) \leq \alpha$ as $\alpha \to 0$. The reason the result is true is because the best possible asymptotic delay for the class of algorithms $\Delta(\alpha, \beta)$ is the delay of the Shiryaev test and we show in Section 2.6 that the asymptotic delay of $\gamma(A, B)$, for a fixed β and ρ, converges to the Shiryaev delay as $\alpha \to 0$.

While the asymptotic optimality ensures good performance of $\gamma(A, B)$ for low values of PFA, it is important to know for moderate PFA values, how well $\gamma(A, B)$ performs as compared to the optimal solution of (10.7). In Section 2.9.2, we obtain ANO-ADD trade-off curves for $\gamma(A, B)$, and show that it is possible to achieve ANO values as low as 30% of $\mathsf{E}[\Gamma]$ by incurring a delay penalty of less than 10% (Figure 10.7), and without affecting the PFA values. Thus, the only case where any other control policy can significantly outperform $\gamma(A, B)$ is when the PFA constraint is moderate and the ANO constraint is small.

Finally, since $\gamma(A, B)$ uses the state of the system for observation control, we will show in Section 2.9.3 that this results in a significant amount of reduction in the observation cost as compared to the scheme where the observations are skipped randomly.

In Section 2.8, we will provide a set of approximations using which the asymptotic expressions for the two-threshold algorithm $\gamma(A, B)$ can be computed. This can be used to choose the thresholds of the algorithms that satisfy a given set of constraints α and β. We summarize the results and comment on future work in Section 2.10.

2.3. Asymptotic Analysis of $\gamma(A, B)$

In this section we derive asymptotic approximations for ADD, PFA and ANO for the two-threshold policy $\gamma(A, B)$. To that end, we first convert the recursion for p_k (see (10.9) and (10.10)) to a form that is amenable to asymptotic analysis.

Define, $Z_k = \log \frac{p_k}{1-p_k}$ for $k \geq 0$. This new variable Z_k has a one-to-one mapping with p_k. By defining

$$a = \log \frac{A}{1-A}, \quad b = \log \frac{B}{1-B},$$

we can write the recursions (10.9) and (10.10) in terms of Z_k.
For $k \geq 1$,

$$Z_{k+1} = Z_k + \log L(X_{k+1}) + |\log(1-\rho)| + \log\left(1 + \rho\, e^{-Z_k}\right), \text{ if } Z_k \in [b, a) \tag{10.14}$$

and

$$Z_{k+1} = Z_k + |\log(1-\rho)| + \log\left(1 + \rho\, e^{-Z_k}\right), \text{ if } Z_k \notin [b, a) \tag{10.15}$$

with

$$Z_1 = \log\left(e^{Z_0} + \rho\right) + |\log(1-\rho)| + \log\left(L(X_1)\right) \mathbb{1}_{\{Z_0 \in [b,a)\}}.$$

Here we have used the fact that $S_{k+1} = 1$ if $p_k \in [B, A)$, and $S_{k+1} = 0$ otherwise (see (10.12)). The crossing of thresholds A and B by p_k is equivalent to the crossing of thresholds a and b by Z_k. Thus the stopping time for $\gamma(A, B)$ (equivalently $\gamma(a, b)$ with some abuse of notation) is

$$\tau = \inf\left\{k \geq 1 : Z_k > a\right\}.$$

In this section we study the asymptotic behavior of $\gamma(a, b)$ in terms of Z_k, under various limits of a, b and ρ. Specifically, we provide two asymptotic expressions for ADD, one for fixed thresholds a, b, as $\rho \to 0$, and another for fixed b and ρ, as $a \to \infty$. We also provide, as $a \to \infty$ and $\rho \to 0$, an asymptotic expression for PFA for fixed b. Finally, we also provide asymptotic estimates of the average number of observations used before (ANO) and after the change point Γ. Note that the limit of $a \to \infty$ corresponds to PFA going to zero.

Figure 10.6 shows a typical evolution of $\gamma(a, b)$, i.e., of Z_k using (10.14) and (10.15), starting at time 0. Note that for $Z_k \in [b, a)$, recursion (10.14) is employed, while outside that interval, recursion (10.15), which only uses the prior ρ, is employed. As a result Z_k increases monotonically outside $[b, a)$.

From Figure 10.6 again, each time Z_k crosses b from below, it can either increase to a (point τ), or it can go below b and approach b monotonically from below, at which time it faces a similar set of alternatives. Thus the passage to threshold a possibly involves multiple cycles of the evolution of Z_k below b. We will show in Section 2.6 that after the change point Γ, following a finite number of cycles below b, Z_k grows up to cross a, and the time spent on the cycles below b is insignificant as compared to $\tau - \Gamma$, as $a \to \infty$. In fact we show that, asymptotically, the time to reach a is equal to the time taken by the classical Shiryaev algorithm to reach a. (Note that for the classical Shiryaev algorithm the evolution of Z_k would be based on (10.14)).

When Z_k crosses a from below, it does so with an overshoot. Overshoots play a significant role in the performance of many sequential algorithms (see [153], [171]) and they are central to the performance of $\gamma(a, b)$ as well. In Section 2.5, we show that PFA depends on the threshold a and the overshoot $(Z_\tau - a)$ as $a \to \infty$, but is *not* a function of the threshold b.

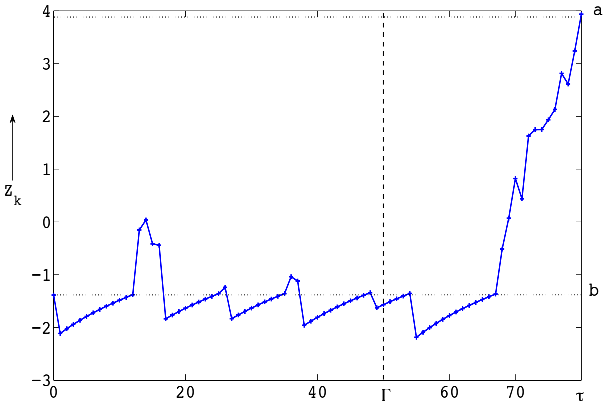


Figure 10.6: Evolution of Z_k for $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.5,1)$, and $\rho = 0.01$, with thresholds $a = 3.89$, and $b = -1.38$, corresponding to the p_k thresholds $A = 0.98$ and $B = 0.2$, respectively. Also $Z_0 = b$.

The number of observations taken during the detection process is the total time spent by Z_k between b and a. As $a \to \infty$, Z_k crosses a only after change point Γ, with high probability. The total number of observations taken can thus be divided in to two parts: the part taken before Γ (ANO), which is the fraction of time Z_k is above b (and hence depends only on b), and the part consumed after Γ. In Section 2.7 we show that, asymptotically, the average number of observations used after Γ is approximately equal to the delay itself.

In Section 2.8 we provide approximations using which the asymptotic expressions can be computed and provide numerical results to demonstrate that under various scenarios, for limiting as well as moderate values of a, b, and ρ, our asymptotic expressions for ADD, PFA and ANO provide good approximations. In Section 2.9 we use the asymptotic expressions for ADD, PFA and ANO to show some optimality properties of $\gamma(a,b)$.

We begin our analysis by first obtaining the asymptotic overshoot distribution for $(Z_\tau - a)$ using non-linear renewal theory [153, 186]. As mentioned above, this will be critical to the PFA analysis.

In what follows, we use E_ℓ and P_ℓ to denote, respectively, the expectation and probability measure when change happens at time ℓ. We use E_∞ and P_∞ to denote, respectively, the expectation and probability measure when the entire sequence $\{X_n\}$ is i.i.d. with density f_0. Also recall that, $g(x) = o(1)$ as $x \to x_0$ is used to denote that $g(x) \to 0$ in the specified limit.

2.4. Asymptotic Overshoot

In this section we characterize the overshoot distribution of Z_k as it crosses a as $a \to \infty$. In analyzing the trajectory of Z_k, it useful to allow for arbitrary starting point Z_0 (shifting the time axis). We first combine the recursions in (10.14) and (10.15) to get:

$$Z_{k+1} = Z_k + \mathbb{1}_{\{Z_k \geq b\}} \log L(X_{k+1}) + |\log(1-\rho)| + \log\left(1 + e^{-Z_k}\rho\right).$$

By defining $Y_k = \log L(X_k) + |\log(1-\rho)|$ and expanding the above recursion, we can write an expression for Z_n:

$$\begin{aligned} Z_n &= \sum_{k=1}^{n} Y_k + \log\left(e^{Z_0} + \rho\right) + \sum_{k=1}^{n-1} \log\left(1 + e^{-Z_k}\rho\right) - \sum_{k=1}^{n} \mathbb{1}_{\{Z_k < b\}} \log L(X_k) \\ &= \sum_{k=1}^{n} Y_k + \eta_n. \end{aligned} \tag{10.16}$$

Here η_n is used to represent all terms other than the first in the equation above:

$$\eta_n = \log\left(e^{Z_0} + \rho\right) + \sum_{k=1}^{n-1} \log\left(1 + e^{-Z_k}\rho\right) - \sum_{k=1}^{n} 1\!\!1_{\{Z_k<b\}} \log L(X_k). \tag{10.17}$$

As defined in [153], η_n is a *slowly changing* sequence if

$$n^{-1} \max\{|\eta_1|, \ldots, |\eta_n|\} \xrightarrow[i.p.]{n\to\infty} 0, \tag{10.18}$$

and for every $\epsilon > 0$, there exists n^* and $\delta > 0$ such that for all $n \geq n^*$

$$\mathrm{P}\{\max_{1\leq k\leq n\delta} |\eta_{n+k} - \eta_n| > \epsilon\} < \epsilon. \tag{10.19}$$

If indeed $\{\eta_n\}$ is a slowly changing sequence, then the distribution of $Z_\tau - a$, as $a \to \infty$, is equal to the asymptotic distribution of the overshoot when the random walk $\sum_{k=1}^n Y_k$ crosses a large positive boundary. We have the following result.

Theorem 2.1. *Let $R(x)$ be the asymptotic distribution of the overshoot when the random walk $\sum_{k=1}^n Y_k$ crosses a large positive boundary under P_1. Then for fixed ρ and b, under P_1, we have the following:*

1. *$\{\eta_n\}$ is a slowly changing sequence.*
2. *$R(x)$ is the distribution of $Z_\tau - a$ as $a \to \infty$, i.e.,*

$$\lim_{a\to\infty} \mathrm{P}_1\left[Z_\tau - a \leq x | \tau \geq \Gamma\right] = R(x). \tag{10.20}$$

2.5. PFA Analysis

We first obtain an expression for PFA as a function of the overshoot when Z_k crosses a.

Lemma 2.1. *For fixed ρ and b,*

$$\text{PFA} \quad = \quad \mathrm{E}[1-p_\tau] = e^{-a}\mathrm{E}[e^{-(Z_\tau - a)}|\tau \geq \Gamma](1+o(1)) \qquad \textit{as } a \to \infty.$$

From Lemma 2.1, it is evident that PFA depends on the overshoot when Z_k crosses a as $a \to \infty$. Since the overshoot has an asymptotic distribution (Theorem 2.1) that depends only on densities f_0, f_1 and prior ρ, and is independent of b, it is natural to expect that as $a \to \infty$, PFA is completely characterized by the asymptotic distribution $R(x)$ and is not a function of the threshold b. This is indeed true and is established in the following theorem.

Theorem 2.2. *For a fixed b and ρ,*

$$\text{PFA}(\gamma(a,b)) = \left(e^{-a}\int_0^\infty e^{-x} dR(x)\right)(1+o(1)) \textit{ as } a \to \infty. \tag{10.21}$$

2.6. Delay Analysis

The PFA for $\gamma(a,b)$ have the following bound:

$$\text{PFA} = \mathrm{E}[1-p_\tau] \leq 1 - A = \frac{1}{1+e^a} \leq e^{-a}. \tag{10.22}$$

Using this upper bound we can show that the ADD of $\gamma(a,b)$ is given by:

$$\begin{aligned} \text{ADD} \quad &= \quad \mathrm{E}\left[(\tau-\Gamma)^+\right] \\ &= \quad \mathrm{E}[\tau - \Gamma|\tau\geq\Gamma](1+o(1)) \;\text{ as } a\to\infty. \end{aligned} \tag{10.23}$$

In the following we provide two different expressions for $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$. The first one is obtained by keeping b fixed and taking $\rho \to 0$. This expression will be used to get accurate delay estimates for $\gamma(a, b)$ in Section 2.8

Next, we will provide another asymptotic expression for $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$ for a fixed b, ρ and as $a \to \infty$. We show that in this limit, $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$ converges to the Shiryaev delay. This fact will be used to prove the asymptotic optimality of $\gamma(a, b)$ in Section 2.9.

It was discussed in reference to Figure 10.6 that each time Z_k crosses b from below, it faces two alternatives, to cross a without ever coming back to b or to go below b and cross it again from below. It was mentioned that the passage to the threshold a is through multiple such cycles. Motivated by this we define the following stopping times λ and Λ:

$$\lambda \stackrel{\Delta}{=} \inf\{k \geq 1 : Z_k \notin [b, a), Z_0 = b\}, \tag{10.24}$$

and

$$\Lambda \stackrel{\Delta}{=} \inf\{k \geq 1 : Z_k > a \ \text{ or } \exists \ k \ s.t. \ Z_{k-1} < b \text{ and } Z_k \geq b \ , Z_0 = b\}. \tag{10.25}$$

Let $t(x, y)$ be the constant time taken by Z_k to move from $Z_0 = x$ to y using the recursion (10.15), i.e.

$$t(x, y) \stackrel{\Delta}{=} \inf\{k \geq 0 : Z_k > y, Z_0 = x, \ x, y \notin [b, a)\}. \tag{10.26}$$

Then, we can write Λ as a function of λ using (10.26):

$$\Lambda = (\lambda + t(Z_\lambda, b))1\!\!1_{\{Z_\lambda < b\}} + \lambda \, 1\!\!1_{\{Z_\lambda > a\}} = \lambda + t(Z_\lambda, b) 1\!\!1_{\{Z_\lambda < b\}}.$$

The significance of these stopping times is as follows. If we start the process at $Z_0 = b$ and *reset Z_k to b each time it crosses b from below*, then the time taken by Z_k to move from b to a is the sum of a finite but random number of random variables with distribution of Λ, say $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$. For $i = 1, \ldots, N-1$, $Z_{\Lambda_i} < b$, and $Z_{\Lambda_N} > a$. Thus the time to reach a in this case is $\mathsf{E}_1 \left[\sum_{k=1}^{N} \Lambda_k\right]$. Let

$$\mathrm{ADD}^s \stackrel{\Delta}{=} \mathsf{E}_1 \left[\sum_{k=1}^{N} \Lambda_k\right].$$

The behavior of the delay path depends on Z_Γ, the value of Z_k at the change point Γ, and how Z_k evolves after that point. We use $\{Z_k \nearrow b\}$ to indicate that Z_k approaches b from below for some $k > \Gamma$, i.e. $\exists k > \Gamma, s.t., Z_{k-1} < b, Z_k \geq b.$ and use $\{Z_k \nearrow a\}$ to represent the event that Z_k crossed a without ever coming back to b, i.e., $Z_k \geq b, \forall k > \Gamma$. We define the following three disjoint events:

$$\begin{aligned} \mathcal{A} &= \{Z_\Gamma < b\}, \\ \mathcal{B} &= \{Z_\Gamma \geq b; Z_k \nearrow b\}, \\ \mathcal{C} &= \{Z_\Gamma \geq b; Z_k \nearrow a\}. \end{aligned}$$

Thus, under the event $\mathcal{A}$, the process Z_k starts below b at Γ, and reaches a after multiple up-crossings of the threshold b. Under the event $\mathcal{B}$, the process Z_k starts above b at Γ, and crosses b before a. It then has multiple up-crossings of b, similar to the case of event $\mathcal{A}$. Under event $\mathcal{C}$, the process Z_k starts above b at Γ, and reaches a without ever coming below b.

Also define,

$$\lambda(x) = \inf\{k \geq 1 : Z_k \notin [b, a), Z_0 = x, b \leq x < a\}, \tag{10.27}$$

and let $\Lambda(x)$ be defined with $Z_0 = x$ similar to (10.25). Thus, λ and $\lambda(b)$ have the same distribution. Similarly, Λ and $\Lambda(b)$ are identically distributed.

The following theorem gives an asymptotic expression for the conditional delay.

Theorem 2.3. *For a fixed values of the thresholds a, b, the conditional delay is given by*

$$\begin{aligned}\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] \;=\; & \Big[\mathrm{ADD}^s \;\; \mathrm{P}(\mathcal{A} \cup \mathcal{B} | \tau \geq \Gamma) \\ & + \;\; \mathrm{E}[\Lambda(Z_\Gamma) | \mathcal{C}, \tau \geq \Gamma] \;\; \mathrm{P}(\mathcal{C} | \tau \geq \Gamma) \\ & + \;\; \mathrm{E}[t(Z_\Gamma, b) | \mathcal{A}, \tau \geq \Gamma] \;\; \mathrm{P}(\mathcal{A} | \tau \geq \Gamma) \\ & + \;\; \mathrm{E}[\Lambda(Z_\Gamma) | \mathcal{B}, \tau \geq \Gamma] \;\; \mathrm{P}(\mathcal{B} | \tau \geq \Gamma)\Big] \big(1 + o(1)\big) \text{ as } \rho \to 0. \end{aligned} \tag{10.28}$$

In Section 2.8 we will provide approximations for various terms in (10.28) to get an accurate estimate of ADD. In Lemma 2.2 we provide expressions for ADD^s.

Let Ψ represent the Shiryaev recursion, i.e., updating Z_k using only (10.14). Define

$$\nu(x, y) = \inf \left\{k \geq 1 : \Psi(Z_{k-1}) > y, \;\; Z_0 = x\right\}. \tag{10.29}$$

Thus, $\nu(x, y)$ is the time for the Shiryaev test to reach y starting at x. Also, define the stopping times:

$$\nu_b = \nu(b, a), \tag{10.30}$$

and

$$\nu_0 = \nu(-\infty, a). \tag{10.31}$$

Note that, ν_0 is the stopping time for the classical Shiryaev test [147] and ν_b is its modified form which starts at b. We have the following asymptotic expression.

Lemma 2.2. *For a fixed b and ρ, ADD^s, the average time for Z_k to cross a starting at b, under P_1, with Z_k reset to b each time it crosses b from below, is given by*

$$\mathrm{ADD}^s = \frac{\mathrm{E}_1[\lambda] + \mathrm{E}_1[t(Z_\lambda, b) | \{Z_\lambda < b\}] \mathrm{P}_1(Z_\lambda < b)}{\mathrm{P}_1(Z_\lambda > a)}, \tag{10.32}$$

and is asymptotically equal to the time taken by the Shiryaev algorithm to move from b to a, i.e.,

$$\mathrm{ADD}^s = \mathrm{E}_1[\nu_b](1 + o(1)) \text{ as } a \to \infty. \tag{10.33}$$

Note that Theorem 2.3 takes $\rho \to 0$. We now provide another expression for $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$, for a fixed b and ρ as $a \to \infty$, which will be used to prove the asymptotic optimality of $\gamma(a, b)$ in Section 2.9.

Theorem 2.4. *For a fixed b and ρ, we have as $a \to \infty$*

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] \leq \mathrm{ADD}^s \left(1 + o(1)\right), \tag{10.34}$$

and hence, we have

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] = \left[\frac{a}{D(f_1, f_0) + |\log(1 - \rho)|}\right] \big(1 + o(1)\big) \text{ as } a \to \infty, \tag{10.35}$$

where, $D(f_1, f_0)$ is the K-L divergence between f_0 and f_1.

2.7. Computation of ANO

First note that,

$$\begin{aligned}
\mathrm{ANO} &= \mathrm{E}\left[\sum_{k=1}^{\min\{\tau,\Gamma-1\}} S_k\right] \\
&= \mathrm{E}\left[\sum_{k=1}^{\Gamma-1} S_k \middle| \tau \geq \Gamma\right] \mathrm{P}(\tau \geq \Gamma) + \mathrm{E}\left[\sum_{k=1}^{\tau} S_k \middle| \tau < \Gamma\right] \mathrm{P}(\tau < \Gamma) \\
&= \mathrm{E}\left[\sum_{k=1}^{\tau} S_k \middle| \tau \geq \Gamma\right] (1 + o(1)) \quad \text{as} \quad a \to \infty.
\end{aligned}$$

The last equality follows because $\sum_{k=1}^{\Gamma-1} S_k \leq \Gamma$ on $\{\tau < \Gamma\}$, and $\mathrm{P}(\tau < \Gamma) < e^{-a} \to 0$ as $a \to \infty$.

Following (10.24), we define

$$\hat{\lambda} = \inf\{k \geq 1 : Z_k < b, Z_0 = b, a = \infty\}. \tag{10.36}$$

The theorem below an gives asymptotic expression for ANO.

Theorem 2.5. *For fixed b, we have as $a \to \infty$, and as $\rho \to 0$,*

$$\mathrm{ANO} = \frac{\mathrm{E}_\infty[\hat{\lambda}]}{\mathrm{P}_\infty[\Gamma \leq \hat{\lambda} + t(Z_{\hat{\lambda}}, b)]} \frac{1}{1 + e^b}(1 + o(1)),$$

where, $\hat{\lambda}$ is as defined in (10.36)*.*

Proof. Let $t(b)$ be the first time Z_k crossed b from below, i.e., $t(b) = t(z_0, b)$. Using the fact that observations are used only after $t(b)$, we can write the following:

$$\begin{aligned}
\mathrm{ANO} &= \mathrm{E}\left[\sum_{k=1}^{\Gamma-1} S_k \middle| \tau \geq \Gamma\right] \\
&= \mathrm{E}\left[\sum_{k=t(b)}^{\Gamma-1} S_k \middle| \Gamma > t(b), \tau \geq \Gamma\right] \mathrm{P}(\Gamma > t(b) | \tau \geq \Gamma).
\end{aligned} \tag{10.37}$$

We now compute each of the two terms in (10.37). For the first term in (10.37), we have the following lemma.

Lemma 2.3. *For a fixed b, as $a \to \infty$, $\rho \to 0$,*

$$\mathrm{E}\left[\sum_{k=t(b)}^{\Gamma-1} S_k \middle| \Gamma > t(b), \tau \geq \Gamma\right] = \frac{\mathrm{E}_\infty[\hat{\lambda}]}{\mathrm{P}_\infty[\Gamma \leq \hat{\lambda} + t(Z_{\hat{\lambda}}, b)]}(1 + o(1)).$$

For the second term in (10.37), we show that $\mathrm{P}(\Gamma > t(b) | \tau \geq \Gamma)$ is equal to $\frac{1}{1+e^b}$ in the limit and is independent of z_0.

Lemma 2.4.

$$\mathrm{P}(\Gamma > t(b) | \tau \geq \Gamma) = \frac{1}{1 + e^b} + o(1) \quad \textit{as} \quad a \to \infty, \rho \to 0.$$

The Lemmas 2.3 and 2.4 taken together completes the proof of Theorem 2.5. □

Define,

$$\mathrm{ANO}_1 = \mathrm{E}\left[\sum_{k=\Gamma}^{\tau} S_k \middle| \tau \geq \Gamma\right].$$

Thus, ANO_1 is the average number of observations used after the change point Γ. In some applications it might be of interest to have an estimate of ANO_1 as well. The following theorem shows that ANO_1 is approximately equal to the delay itself.

Theorem 2.6. *For fixed b and ρ, we have*

$$\mathrm{ANO}_1 = \mathrm{E}_1[\nu_b](1 + o(1)), \quad \textit{as } a \to \infty.$$

2.8. Approximations and Numerical Results

In Sections 2.5-2.7, we have obtained asymptotic expressions for ADD, PFA, and ANO as a function of the system parameters: the thresholds a, b, the densities f_0 and f_1, and the prior ρ. In this section we provide approximations for various analytical expressions obtained in these sections. The observations used are Gaussian with $f_0 \sim \mathcal{N}(0,1)$, and $f_1 \sim \mathcal{N}(\theta,1)$, $\theta > 0$, for the simulations and analysis. In the simulations, the PFA values are computed using the expression $\mathsf{E}[1-p_\tau]$. This guarantees a faster convergence for small values of PFA.

2.8.1. Numerical results for PFA

By Theorem 2.2, we have the following approximation for PFA:

$$\mathrm{PFA} \approx e^{-a} \int_0^\infty e^{-x} dR(x).$$

We note that $\int_0^\infty e^{-x} dR(x)$ and $\bar{r}$ can be computed numerically, at least for Gaussian observations [153]. In this section we provide numerical results to show the accuracy of the above expression for PFA.

In Table 10.1 we compare the analytical approximation with the PFA obtained using simulations of $\gamma(a,b)$ for various choices of ρ, thresholds a, b, and post change mean θ. From the table we see that the analytical approximation is quite good.

Table 10.1: PFA: for $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(\theta,1)$

θ	ρ	a	b	PFA Simulations	PFA Analysis
0.4	0.01	3.0	0	3.78×10^{-2}	3.94×10^{-2}
0.4	0.01	6.0	2.0	1.955×10^{-3}	1.96×10^{-3}
0.75	0.01	9.0	-2.0	7.968×10^{-5}	7.964×10^{-5}
2.0	0.01	5.0	-4.0	2.15×10^{-3}	2.155×10^{-3}
0.75	0.005	7.6	3.0	3.231×10^{-4}	3.235×10^{-4}
0.75	0.1	4.0	-3.0	1.143×10^{-2}	1.157×10^{-2}

In Table 10.2, we show that PFA is not a function of b for large values of a. We fix $a = 4.6$, and increase b from -2.2 to 0.85. We notice that PFA is unchanged in simulations when b is changed this way. This is also captured by the analysis and it is quite accurate.

2.8.2. Approximations and Numerical Results for ANO *and* ANO_1

We recall the expressions for ANO from Theorem 2.5 and for ANO_1 from Theorem 2.6:

$$\begin{aligned}\mathrm{ANO} &\approx \frac{\mathrm{E}_\infty[\hat{\lambda}]}{\mathrm{P}_\infty[\Gamma \leq \hat{\lambda} + t(Z_{\hat{\lambda}}, b)]} \frac{1}{1+e^b} \\ \mathrm{ANO}_1 &= \mathrm{E}_1[\nu_b].\end{aligned}$$

Table 10.2: PFA for $\rho = 0.01$, $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.75,1)$

a	b	Simulations	Analysis
4.6	-2.2	6.44×10^{-3}	6.48×10^{-3}
4.6	-1.5	6.44×10^{-3}	6.48×10^{-3}
4.6	-0.85	6.44×10^{-3}	6.48×10^{-3}
4.6	0	6.44×10^{-3}	6.48×10^{-3}
4.6	0.85	6.44×10^{-3}	6.48×10^{-3}

We first simplify the expression for ANO. Note that

$$\begin{aligned} \mathrm{P}_\infty[\Gamma \leq \hat{\lambda} + t(Z_{\hat{\lambda}}, b)] &= 1 - \mathrm{P}_\infty[\Gamma > \hat{\lambda} + t(Z_{\hat{\lambda}}, b)] \\ &= 1 - \mathrm{E}_\infty[(1-\rho)^{\hat{\lambda}+t(Z_{\hat{\lambda}}, b)}]. \end{aligned}$$

Thus, using Binomial approximation we get

$$\mathrm{P}_\infty[\Gamma \leq \hat{\lambda} + t(Z_{\hat{\lambda}}, b)] \approx \rho \left(\mathrm{E}_\infty[\hat{\lambda}] + \mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)] \right).$$

Thus, we have

$$\mathrm{ANO} \approx \frac{\rho^{-1}\,\mathrm{E}_\infty[\hat{\lambda}]}{\mathrm{E}_\infty[\hat{\lambda}] + \mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]} \frac{1}{1+e^b}. \tag{10.38}$$

We now provide approximation to compute $\mathrm{E}_\infty[\hat{\lambda}]$ and $\mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]$ in (10.38). Invoking Wald's lemma [153], we write $\mathrm{E}_\infty[\hat{\lambda}]$ as,

$$\mathrm{E}_\infty[\hat{\lambda}] = \frac{\mathrm{E}_\infty[Z_{\hat{\lambda}}] - \mathrm{E}_\infty[\eta_{\hat{\lambda}}]}{-D(f_1, f_0) + |\log(1-\rho)|}.$$

We have developed the following approximation for $\mathsf{E}_\infty[\hat{\lambda}]$:

$$\mathsf{E}_\infty[\hat{\lambda}] \approx \frac{\bar{r} + \log(1+\rho e^{-b})}{D(f_1, f_0) - |\log(1-\rho)|}. \tag{10.39}$$

Here, $\log(1+\rho e^{-b})$ is an approximation to $\mathrm{E}_\infty[\eta_{\hat{\lambda}}]$ by ignoring all the random terms after b is factored out of it. This extra b will cancel with the b in $\mathrm{E}_\infty[Z_{\hat{\lambda}}] = b + \mathrm{E}_\infty[Z_{\hat{\lambda}} - b]$. We approximate $\mathrm{E}_\infty[b - Z_{\hat{\lambda}}]$ by $\bar{r}$, the mean overshoot of the random walk $\sum_{i=1}^k Y_k$, with mean $D(f_1, f_0) - |\log(1-\rho)|$, when it crosses a large boundary (see (10.16)). For the term $\mathsf{E}_\infty[t(Z_{\hat{\lambda}}, b)]$, we have obtained the following approximation:

$$\mathsf{E}_\infty[t(Z_{\hat{\lambda}}, b)] \approx \int_0^\infty \frac{\log(1+e^b) - \log(1+e^{b-x})}{|\log(1-\rho)|} dR(x). \tag{10.40}$$

Thus, we approximate the distribution of $(b - Z_{\hat{\lambda}})$ by $R(x)$.

Based on the second order approximation for $\mathrm{E}_1[\nu_0]$ developed in [171], we have obtained the following approximation for $\mathrm{E}_1[\nu_b]$:

$$\mathrm{E}_1[\nu_b] = \frac{a - \mathrm{E}[\eta(b)] + \bar{r}}{D(f_1, f_0) + |\log(1-\rho)|} + o(1) \text{ as } a \to \infty, \tag{10.41}$$

where, $\eta(b)$ is the a.s. limit of the slowly changing sequence η_n with $Z_0 = b$ under f_1, (see (10.17))and

$$\bar{r} = \int_0^\infty x dR(x), \tag{10.42}$$

with $R(x)$ as in Theorem 2.1.

In Table 10.3 we demonstrate the accuracy of approximations for ANO and ANO_1, for various values of ρ, thresholds a, b, and post change mean θ. The table shows that the approximations are quite accurate for the parameters chosen.

Table 10.3: $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(\theta,1)$

				ANO		ANO_1	
θ	ρ	a	b	Simulations	Analysis	Simulations	Analysis
0.4	0.01	8.5	-2.2	66.3	62.88	102.9	111.7
0.75	0.01	6.467	-2.2	34.92	34.24	27.86	29.46
2.0	0.01	7.5	-4.0	42.94	46.4	6.08	6.23
0.75	0.005	8.7	-3.0	77.18	75.09	38.73	40.38
0.75	0.1	8.5	0.0	2.64	3.2	21.17	22.18

2.8.3. Approximations and Numerical Results for ADD

Theorem 2.4 gave a first order approximation for $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$:

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] \approx \left[\frac{a}{D(f_1, f_0) + |\log(1-\rho)|} \right].$$

Note that, from [171], this is also the first order approximation for ADD of the Shiryaev algorithm, and gives a good estimate of the delay when PFA is small. For the Shiryaev delay, a second order approximation was developed in [171] (also see (10.41)):

$$\mathrm{E}_1[\nu_0] = \left[\frac{a - \mathsf{E}[\eta(-\infty)] + \bar{r}}{D(f_1, f_0) + |\log(1-\rho)|} \right] + o(1) \text{ as } a \to \infty.$$

So, instead of using $\frac{a}{D(f_1,f_0)+|\log(1-\rho)|}$, we propose to use the following:

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] \approx \left[\frac{a - \mathsf{E}[\eta(-\infty)] + \bar{r}}{D(f_1, f_0) + |\log(1-\rho)|} \right]. \tag{10.43}$$

For the Shiryaev algorithm, (10.43) provides a very good estimate of the delay even for moderate values of PFA. In case of $\gamma(a,b)$, the accuracy of (10.43) depends on the choice of b and hence on the constraint β, as having $b > -\infty$ increases the delay. Before we demonstrate this by numerical and simulation results we introduce the following concept:

$$\text{ANO\%} = \text{ ANO } \textit{expressed as a percentage of } \mathsf{E}[\Gamma]. \tag{10.44}$$

For example, if $\rho = 0.05$, and for some choice of system parameters $\text{ANO} = 15$, then $\text{ANO\%} = 15 * 0.05 = 75\%$. Thus, the concept of ANO% captures the reduction in the average number of observations used before change by employing $\gamma(a,b)$.

In Table 10.4 we provide various numerical examples where (10.43) is a good approximation for $\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma]$. Since, (10.43) is a good approximation for the Shiryaev delay as well, it follows that, for these parameter values, the delay of $\gamma(a,b)$ is approximately equal to the Shiryaev delay. It might be intuitive that if we are aiming for large ANO% values of say 90%, then the delay will be close to the Shiryaev delay. But values in Table 10.4 shows that it is possible to achieve considerably smaller values of ANO% without significantly affecting the delay.

However, if the ANO% value is small, then this means that the value of b is large, and this implies that the delay is large. In this case, it might happen that (10.43) is a good approximation only for values of PFA which are very small. This is demonstrated in Table 10.5. It is clear from the table that, for the parameter values considered, estimating the delay with less than 10% error is only possible at PFA values of the order of PFA $\approx 10^{-22}$.

This motivates the need for a more accurate estimate of the delay. This is provided below.

Table 10.4: $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(\theta,1)$

				ADD		PFA		ANO%
θ	ρ	a	b	Simulations $\mathsf{E}[\tau - \Gamma \vert \tau \geq \Gamma]$	Analysis (10.43)	Simulations	Analysis	
0.4	0.01	8.5	-2.2	104.9	111.7	1.608×10^{-4}	1.608×10^{-4}	66%
0.75	0.01	6.467	-2.2	32.3	29.5	1.002×10^{-3}	1.004×10^{-3}	35%
2.0	0.01	7.5	-4.0	6.1	6.23	1.77×10^{-4}	1.768×10^{-4}	43%
0.75	0.005	8.7	-3.0	42.6	40.4	1.076×10^{-4}	1.076×10^{-4}	77%
0.75	0.1	8.5	0.0	23.9	22.18	1.286×10^{-4}	1.285×10^{-4}	26%

Table 10.5: $\rho = 0.05$, $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.75,1)$

a	b	Simulations $\mathsf{E}[\tau - \Gamma \vert \tau \geq \Gamma]$	Analysis (10.43)	ANO%	PFA
5.0	1.0	30	13	7.5%	4.3×10^{-3}
9.0	1.0	42	25	7.5%	7.9×10^{-5}
13.0	1.0	54	37	7.5%	1.4×10^{-6}
18.0	1.0	69	52	7.5%	9.7×10^{-9}
50.0	1.0	165	149	7.5%	1.23×10^{-22}

From Theorem 2.3, recall that we had the following three events:

$$\begin{aligned}
\mathcal{A} &= \{Z_\Gamma < b\}, \\
\mathcal{B} &= \{Z_\Gamma \geq b; Z_k \nearrow b\}, \\
\mathcal{C} &= \{Z_\Gamma \geq b; Z_k \nearrow a\}.
\end{aligned}$$

As a first step towards the approximations, we ignore the event $\mathcal{B}$: $\mathsf{P}(\mathcal{B}) \approx 0$. That is, we assume that if $Z_\Gamma > b$, then Z_k climbs to a. Define,

$$P_b = \mathrm{P}(Z_\Gamma \geq b | \tau \geq \Gamma).$$

Then (10.28),

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] \approx P_b \, \mathrm{E}[\lambda(Z_\Gamma) | \mathcal{C}, \tau \geq \Gamma] + (1 - P_b)(\mathrm{E}[t(Z_\Gamma, b) | \mathcal{A}, \tau \geq \Gamma] + \mathrm{ADD}^s). \qquad (10.45)$$

From Lemma 2.2, it is easy to show the following:

$$\mathrm{ADD}^s = \mathrm{E}_1[\lambda | \{Z_\lambda > a\}] + (\mathrm{E}_1[\lambda | \{Z_\lambda < b\}] + \mathrm{E}_1[t(Z_\lambda, b) | \{Z_\lambda < b\}]) \, \frac{\mathrm{P}_1(Z_\lambda < b)}{1 - \mathrm{P}_1(Z_\lambda < b)}.$$

We now use the following approximations:

$$\begin{aligned}
\mathrm{E}_1[\lambda | \{Z_\lambda > a\}] &\approx \mathrm{E}[\lambda(Z_\Gamma) | \mathcal{C}, \tau \geq \Gamma] \approx \frac{a - \mathsf{E}[\eta(-\infty)] + \bar{r}}{D(f_1, f_0) + |\log(1-\rho)|}, \\
\mathrm{E}_1[\lambda | \{Z_\lambda < b\}] &\approx \frac{\bar{r} + \log(1 + \rho e^{-b})}{D(f_1, f_0) - |\log(1-\rho)|}, \\
\mathrm{E}_1[t(Z_\lambda, b) | \{Z_\lambda < b\}] &\approx t(b - \bar{r}, b) \approx \frac{\log(1 + e^b) - \log(1 + e^{b - \bar{r}})}{|\log(1-\rho)|}.
\end{aligned}$$

To compute (10.45), we also need approximations for $\mathrm{P}_1(Z_\lambda < b)$, P_b and $\mathrm{E}[t(Z_\Gamma, b) | \mathcal{A}]$. Those are provided below. Setting $a = \infty$ we have, by Wald's likelihood identity, Proposition 2.24, Pg 13, [153],

$$\mathrm{P}_1(Z_\lambda < b) = \mathrm{E}_\infty \left[\frac{f_1(X_1) \ldots f_1(X_\lambda)}{f_0(X_1) \ldots f_0(X_\lambda)} \right].$$

Under P_∞, λ a.s. ends in b and with high probability it takes very small values. Hence, this expressions can be computed using Monte Carlo simulations. Further,

$$\begin{aligned} P_b &= \mathrm{P}(\Gamma > t(-\infty, b))\mathrm{P}(Z_\Gamma > b | \Gamma > t(-\infty, b), \tau \geq \Gamma) \\ &\approx \frac{1}{1+e^b} \frac{\mathsf{E}_\infty[\hat{\lambda}]}{\mathrm{E}_\infty[\hat{\lambda}] + \mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]}. \end{aligned}$$

We already have the approximations for $\mathsf{E}_\infty[\hat{\lambda}]$ and $\mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]$ from Section 2.8.2. The approximation for $\mathrm{E}[t(Z_\Gamma, b)|\mathcal{A}]$ can be obtained as follows (all expectations conditioned on $\{\tau \geq \Gamma\}$):

$$\begin{aligned} (1-P_b)\mathrm{E}[t(Z_\Gamma, b)|\mathcal{A}] &= (1-P_b)\mathrm{E}[t(Z_\Gamma, b)|\{Z_\Gamma < b\}] \\ &= \mathrm{E}[t(Z_\Gamma, b)|\{Z_\Gamma < b\} \cap \{\Gamma > t(-\infty, b)\}]\mathrm{P}(\{\Gamma > t(-\infty, b)\} \cap \{Z_\Gamma < b\}) \\ &\quad + \mathrm{E}[t(Z_\Gamma, b)|\{Z_\Gamma < b\} \cap \{\Gamma \leq t(-\infty, b)\}]\mathrm{P}(\{\Gamma \leq t(-\infty, b)\} \cap \{Z_\Gamma < b\}). \end{aligned}$$

This can be computed using

$$\mathrm{P}(\{\Gamma > t(-\infty, b)\} \cap \{Z_\Gamma < b\}) \approx \frac{1}{1+e^b} \frac{\mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]}{\mathrm{E}_\infty[\hat{\lambda}] + \mathrm{E}_\infty[t(Z_{\hat{\lambda}}, b)]},$$

and

$$\mathrm{P}(\{\Gamma \leq t(-\infty, b)\} \cap \{Z_\Gamma < b\}) = \mathrm{P}(\{\Gamma \leq t(-\infty, b)\}) \approx \frac{e^b}{1+e^b}.$$

To compute conditional expectation of $t(Z_\Gamma, b)$, we need to subtract from $t(x, b)$, the mean of Γ conditioned on $\{\Gamma \leq t(x, b)\}$. Specifically,

$$\mathrm{E}[t(Z_\Gamma, b)|\{Z_\Gamma < b\} \cap \{\Gamma > t(-\infty, b)\}] = t(b - \bar{r}, b) - \frac{1}{\mathrm{P}(\Gamma \leq t(b - \bar{r}, b))} \sum_{k=1}^{t(b-\bar{r}, b)} k(1-\rho)^{k-1}\rho,$$

and,

$$\mathrm{E}[t(Z_\Gamma, b)|\{Z_\Gamma < b\} \cap \{\Gamma \leq t(-\infty, b)\}] = t(-\infty, b) - \frac{1}{\mathrm{P}(\Gamma \leq t(-\infty, b))} \sum_{k=1}^{t(-\infty, b)} k(1-\rho)^{k-1}\rho.$$

Thus we have obtained approximations for all the terms for the new approximation for $\mathsf{E}[\tau - \Gamma | \tau \geq \Gamma]$ in (10.45).

In Table 10.6, we now reproduce Table 10.5 with a new column containing delay estimates computed using the new ADD (for $\mathsf{E}[\tau - \Gamma | \tau \geq \Gamma]$) approximation (10.45). The values shows that all estimates are nearly within 10% of the actual value.

In Table 10.7, we show the accuracy of the new ADD approximation (10.45), for various values of the system parameters, by comparing it with simulations and also with (10.43). We also set PFA around 1×10^{-3}. The table clearly demonstrates that the new ADD approximation predicts ADD with less than 10% error.

2.9. Asymptotic Optimality and Performance of $\gamma(a, b)$

2.9.1. Asymptotic Optimality of $\gamma(a, b)$

In Theorem 2.4 we saw that for a fixed b and ρ,

$$\mathrm{E}[\tau - \Gamma | \tau \geq \Gamma] = \left[\frac{a}{D(f_1, f_0) + |\log(1-\rho)|}\right] \big(1 + o(1)\big) \text{ as } a \to \infty.$$

We recall that from Tartakovsky and Veeravalli [171], this is also the asymptotic delay of the Shiryaev algorithm.

Table 10.6: $\rho = 0.05$, $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.75,1)$

a	b	Simulations $\mathsf{E}[\tau - \Gamma \mid \tau \geq \Gamma]$	Analysis (10.43)	New Analysis ADD from (10.45)	ANO%	PFA
5.0	1.0	30	13	34	7.5%	4.3×10^{-3}
9.0	1.0	42	25	46	7.5%	7.9×10^{-5}
13.0	1.0	54	37	58	7.5%	1.4×10^{-6}
18.0	1.0	69	52	73	7.5%	9.7×10^{-9}
50.0	1.0	165	149	169	7.5%	1.23×10^{-22}

Table 10.7: $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.75,1)$, PFA $\approx 10^{-3}$, ANO=10% of Shiryaev ANO

			ADD			
ρ	a	b	Simulations	Analysis	Analysis	
				New (10.45)	(10.43)	ANO%
0.01	6.4	2.7	250	260	14.42	0.33%
0.005	6.45	0.6	181	190	22.09	1.5%
0.001	6.47	-2.7	75	80	33.68	7.6%
0.0005	6.47	-3.49	74	79	36.49	8.4%
0.0001	6.47	-5.2	76	80	42.56	9.6%

Moreover, from Theorem 2.2, the PFA for $\gamma(a,b)$ is

$$\text{PFA} = \left(e^{-a} \int_0^\infty e^{-x} dR(x) \right) (1 + o(1)) \text{ as } a \to \infty.$$

Again from Tartakovsky and Veeravalli [171], this is the PFA for the Shiryaev algorithm. We thus have the following asymptotic optimality result for $\gamma(a,b)$.

Theorem 2.7. *With $\gamma = \{\tau, S_1, \ldots, S_\tau\}$ define*

$$\Delta(\alpha, \beta) = \{\gamma : \text{PFA}(\gamma) \leq \alpha; \;\; \text{ANO}(\gamma) \leq \beta\},$$

then for a fixed β and ρ,

$$\text{ADD}(\gamma(a(\alpha,\beta), b(\alpha,\beta))) = \left[\inf_{\gamma \in \Delta(\alpha,\beta)} \text{ADD}(\gamma) \right] (1 + o(1)) \textit{ as } \alpha \to 0. \tag{10.46}$$

Here, for each α, β, $b(\alpha,\beta)$ is the smallest b such that $\text{ANO}(\gamma(a(\alpha,\beta), b(\alpha,\beta))) \leq \beta$ as $a \to \infty$.

Proof. Fix b such that $\text{ANO}(\gamma(a,b)) \leq \beta$ as $a \to \infty$. It may happen that the constraint β is not met with equality. Thus choose the smallest b which satisfies the constraint β as $a \to \infty$. This choice of threshold b is unique for a given β because ANO is not a function of threshold a as $a \to \infty$.

As $a \to \infty$, the PFA and ADD both approach the Shiryaev PFA (10.21) and Shiryaev delay (10.35), respectively. Thus, as $a \to \infty$, $\gamma(a,b)$ is optimal over the class of all control policies $\Delta(\alpha,\beta)$ that satisfy the constraints α and β. □

Remark 2.2. If we select b_1 such that ANO% $< 1\%$ as $a \to \infty$, and then select a_1 such that $\text{ADD}(\gamma(b_1, a_1))$ is within 1% of $\frac{a}{D(f_1,f_0)+|\log(1-\rho)|}$. Then for all thresholds $d \geq a_1$, $\gamma(d, b_1)$ has ANO% $< 1\%$ and delay within 1% of the Shiryaev delay. Thus, for small values of PFA, as long as we are aiming for ANO% of 1-100, no other control policy can outperform the two-threshold policy $\gamma(a,b)$.

2.9.2. Trade-off Curves: Performance of $\gamma(a,b)$ for a Fixed and Moderate α

Theorem 2.7 shows that for small values of PFA, $\gamma(a,b)$ is approximately optimal, i.e., it is not possible to outperform $\gamma(a,b)$ by a huge margin. But for moderate values of PFA, it is not clear if their exists algorithms which can significantly outperform $\gamma(a,b)$. Our aim is to partially address this issue in this section.

In Figure 10.7 we plot the ANO-ADD trade-off for the two-threshold algorithm. Specifically, we compare the two-threshold algorithm with the classical Shiryaev test and study how much ANO can be reduced without significantly loosing in terms of ADD. For Figure 10.7 we pick four values of $\rho : 0.05, 0.01, 0.005, 0.001$. For a fixed ρ, we fix $b = -\infty$ and select threshold a such that the $\text{PFA}(\gamma(a,b)) = 10^{-4}$. We then increase the threshold b to have ANO% values of $75\%, 50\%, 30\%, 15\%$. We note that it was possible to reduce the ANO to 15% of $\mathsf{E}[\Gamma]$ by increasing the threshold b this way, without affecting the probability of false alarm. Figure 10.7 shows that we can reduce ANO by up to 25% while getting approximately the same ADD performance as that of the Shiryaev test. Moreover, if we allow for a 10% increase in ADD compared to that of the Shiryaev test, then we can reduce ANO by up to 70% (see plot for ANO% =30%).

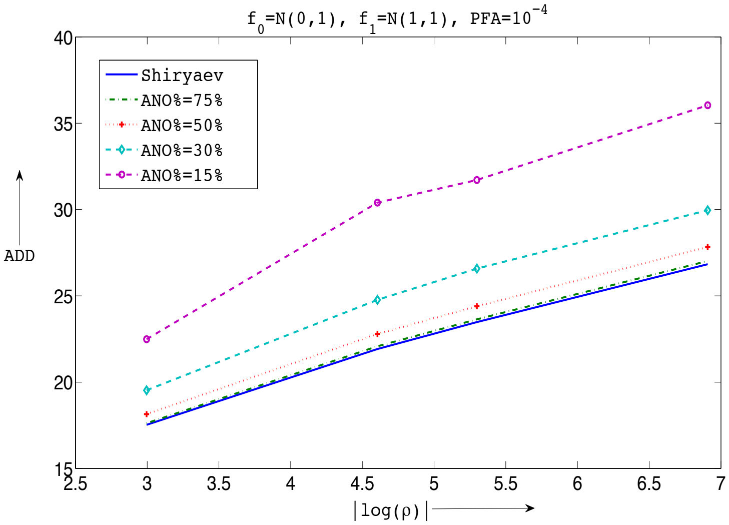


Figure 10.7: Trade-off curves comparing performance of two-threshold algorithm with the Shiryaev test for ANO% of 75, 50, 30 and 15%. $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(1,1)$, and $\text{PFA} = 10^{-4}$.

Such a behavior was also observed in Table 10.4, where we saw that the delay for $\gamma(a,b)$ is approximately equal to the Shiryaev delay for moderate to large ANO% values. Thus, for moderate PFA values, when the ANO% is moderate to large, $\gamma(a,b)$ is approximately optimal.

2.9.3. Comparison with Fractional Sampling

In this section we compare the performance of $\gamma(a,b)$ with the naive approach of fractional sampling, in which an ANO% of ϵ is achieved by employing Shiryaev algorithm and using a sample with probability ϵ. Figure 10.8 compares the two schemes for ANO% of 25%. We also plot the performance of the Shiryaev algorithm for the same values of PFA and ρ. The figures clearly show that $\gamma(a,b)$ helps in reducing the observation cost by a significant margin as compared to the fractional sampling scheme.

2.10. Conclusions

We posed a data-efficient version of the classical Bayesian quickest change detection problem, where we control the number of observations taken before the change occurs. We obtained a two-threshold Bayesian test that is asymptotically optimal, has good trade-off curves and is easy to design. We supported our claim via analytical and simulation results. We derived analytical approximations for the ADD, PFA and ANO performance of the two-threshold policy using which we can design the test by choosing the thresholds. Further, there is a unique pair of thresholds that meets a given set of constraints of probability of false alarm

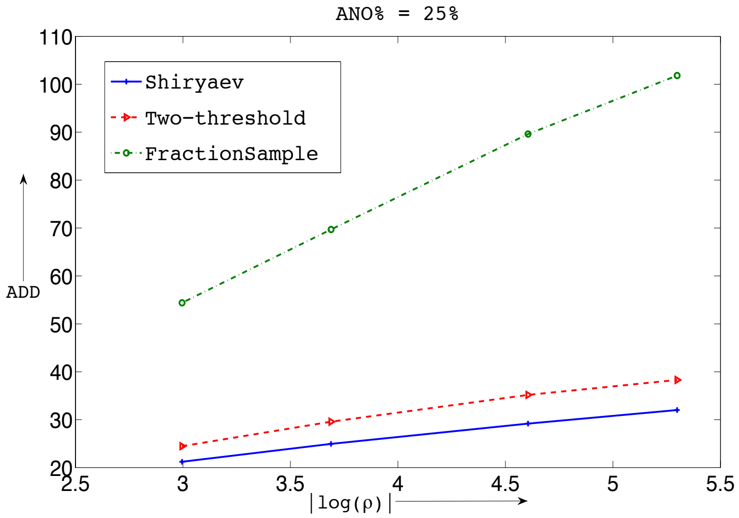


Figure 10.8: Trade-off curves comparing performance of the two-threshold algorithm with the Fractional Sampling Scheme for ANO% 25%. $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(0.75,1)$, and PFA $= 10^{-3}$.

and observation cost. This result has implications in many engineering applications where an abrupt change has to be detected in a process under observation, but there is a cost associated with acquiring the data needed to make accurate decisions.

In the absence of knowledge of the prior on Γ, an important problem for future research is to see if two-threshold policies are optimal in non-Bayesian (e.g., minimax) settings. More importantly, it is of interest to understand how to update the test metric in a non-Bayesian setting when we skip an observation. From an application point of view, one can design a two-threshold test based on the Shiryaev-Roberts or CUSUM approaches Tartakovsky and Moustakides [167], and use the undershoot of the metric when it goes below the threshold 'b', to design the off times. Furthermore, if we are able to find useful lower bounds on delay for given false alarm and ANO constraints, we may be able to use these to prove asymptotic optimality of such heuristic tests, as is done for the standard quickest change detection problem Tartakovsky and Veeravalli [171], Lai [82]. Also, such lower bounds can possibly help in obtaining insights for cases where the observations are not i.i.d. Tartakovsky and Veeravalli [171], Lai [82]. Other interesting problems in this area include the design of data-efficient optimal algorithms for robust change detection or nonparametric change detection.

Chapter 11

Spectral and Measurement Approaches in Information Assurance

This chapter summarizes the work done by the group of Christos Papadopoulos in the area of Spectral Analysis Applications in Network Security. The work includes three fronts. First, applications for a low rate detection algorithm [14]. Second, a study to correlate address characteristics of spammers and non-spammers to determine if spammers have different characteristics and are therefore easier to detect [183]. We present highlights of this work below. Finally, work to detect bots with custom TCP/IP stacks, based on the existence of multiple fingerprints [44]. Due to space restrictions we do not present this work here.

1. Using Low-Rate Flow Periodicities in Anomaly Detection

As desktops and servers become more complicated, they employ an increasing amount of automatic, non-user initiated communication. Such communication can be good (OS updates, RSS feed readers, and mail polling), bad (keyloggers, spyware, and botnet command-and-control), or ugly (adware or unauthorized peer-to-peer applications). Communication in these applications is often periodic but infrequent, perhaps every few minutes to few hours. This infrequent communication and the complexity of today's systems makes these applications difficult for users to detect and diagnose. We show that there are several classes of applications that show low-rate periodicity and demonstrate that they are widely deployed on public networks. In this paper we present a new approach to identify changes in low-rate periodic network traffic. We employ signal-processing techniques, using discrete wavelets implemented as a fully decomposed, iterated filter bank. This approach allows us to cover a large range of low-rate periodicities, from seconds to hours, and to identify approximate times when traffic changed. Network administrators and users can use our techniques for network- or self-surveillance. To measure the effectiveness of our approach, we show that it can detect changes in periodic behavior caused by events such as installation of keyloggers, an interruption in OS update checks, or the P2P application BitTorrent. We quantify the sensitivity of our approach, showing that we can find periodic traffic when it is at least 5–10% of overall traffic.

1.1. Methodology

We use wavelets implemented as an iterated filter bank to identify periods of time when a periodic series of connections is present. In this section we discuss how we go from network events to identifying a change in periodic communication.

Although wavelets provide a well developed mathematical theory, and there has been some work applying wavelets to network traffic before, discovering *infrequent* periodic traffic is particularly demanding because of the long-timescales and sparse signals involved. Here we describe the four main parts of our approach (roughly following the outline of applying signal processing to networking [178]): extracting a timeseries of events from network traffic, decomposing the timeseries using an iterated filter bank, visualizing the resulting multi-resolution representation, and detecting the presence of a periodic signal. Our focus on long-timescales influences each of these steps.

For a complete description of the work please see the paper. here we summarize the results in the following tables.

Table 11.1: Variety of applications that show periodic behavior

Category	Examples	Seen?	Period
User services	WeatherEye	*yes*	30-120
RSS News Feeds	NewzCrawler	*yes*	15-120
Web Counters	Google Analytics	*yes*	5-30
P2P Protocols	Gnutella	*yes*	2-120
Adware	Gator, ISTbar	*yes*	15-60
Spyware/Keyloggers	SpyBuddy	*no*	N/A
Botnet cmd&ctl	(non-commercial)	*no*	N/A

Table 11.1 outlines seven categories of applications which we have researched and identified as participating in periodic communication. We have found numerous examples of applications in five of these seven categories in our four-day trace from USC. However, example applications don't characterize how widespread hosts exhibiting applications with periodic behavior are.

Table 11.2: Prevalence of malware with periodic behavior on our network.

	Blacklisted		**Unique IPs**	
Group	**Destinations**		**(users)**	
active to anywhere	–	–	**128,614**	**[100%]**
active to blacklisted	**181**	**(100%)**	–	–
Non-periodic	120	(66%)	n/a	n/a
Periodic	61	(44%)	n/a	n/a
User Services	5	(3%)	22	[0%]
Web Counters	15	(8%)	16,405	[13%]
Ad Servers	36	(20%)	31,277	[24%]
Other	5	(3%)	6	[0%]

Table 11.2 shows the results of our analysis. We found traffic to 181 of the blacklisted destinations from our campus. About 45,000 IP addresses at USC had traffic to some of these sites, nearly one-third of all active campus addresses. (The presence of dynamic addresses means that this count may not correspond exactly to 45,000 users, since one user may occupy multiple addresses, and vice versa.)

For the 61 blacklisted hosts that had periodic traffic, we manually examined the site and classified it in one of four categories (user services, web counters, ad servers, and other)

1.2. Summary

In this work we have shown that low-rate periodicity is common to several broad classes, both good (OS updates), bad (keyloggers and malware), and ugly (adware), and that these applications are widely deployed on public networks. We have explored a wavelet-based approach to identify such periodic behavior, and begun to explore the sensitivity and robustness of this approach. A promising application of such analysis is self-surveillance, as a user watches his or her own traffic to detect unexpected changes.

1.3. Applications of Low Rate Detection

In the previous section we investigated the underlying fundamentals of detection of low-rate periodic behavior and evidence that periodic behavior occurs. In this section we look at two applications, self-surveillance and network surveillance, and then demonstrate that a variety of applications show low-rate periodicity and those applications occur in real networks.

1.3.1. Self-surveillance: Identifying Changes in Periodic Behavior of a Host

Earlier we demonstrated that malware shows periodic behavior that can be identified, even in the face of noise. Now we show how to identify dynamic *changes* in periodic behavior, and the time *when* these changes occur.

We wish to identify changes in the periodic behavior of a given host to help users better understand activities on their computer. All operating systems and an increasing number of applications automatically poll for updates periodically. In addition, spyware and adware often report back to or request new information from the external master. In fact, application updates sometimes do not reveal the presence of automatic polling, or how much information they disclose. Moreover, malware may terminate automatic updates after infecting a host. Thus, users will want to know when automatic checks stop, or the addition of an automatic reporting service to their machines.

We consider two examples where the *change* in periodic behavior is of interest, namely a change in OS update checks and a change in communication patterns created by the installation of a keylogger.

Detecting operating system updates. Security policies of all operating systems and many applications include automatic polling for updates with typical periods ranging from 30 minutes to a week. Just as network administrators wish to detect the presence of bad behavior, the *absence* of good behavior may also be of great interest. In addition, since automatic updates are often disclosed only in the fine print of an end-user license agreement, users may also wish to know when a newly installed application performs regular update checks.

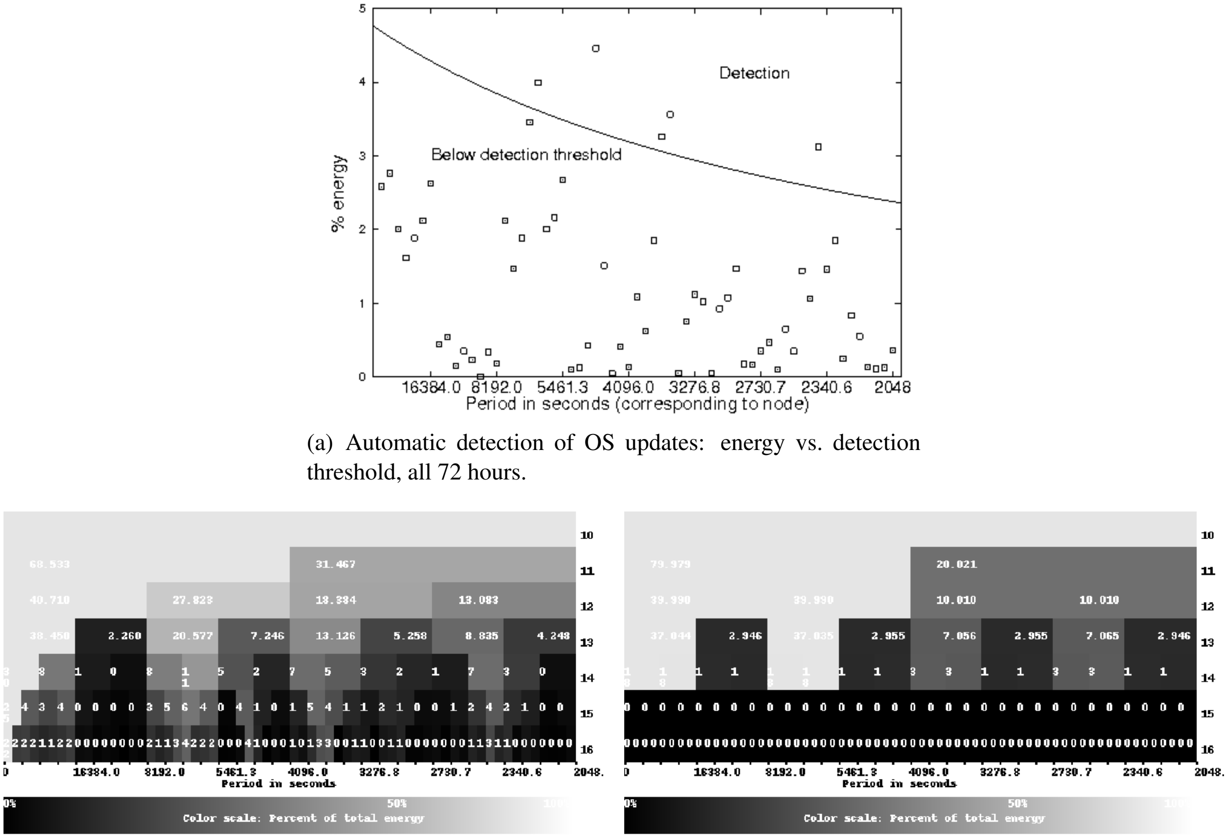
To confirm we can see a change in update checks we monitored a lab machine running the Fedora 10 distribution of Linux for three days. By default, Fedora polls update servers every hour using `yum-updatesd`. During the second day of the experiment, we disabled update checks at 2pm. The machine was lightly used for web browsing and e-mail over the three day period.

Our system correctly identifies periodic behavior near 3600s in this test traffic. Our algorithm to place events in time finds a change in this periodic behavior between noon and 9pm, consistent with our known time of 2pm. We have not tuned our algorithm for temporal placement; a more sophisticated approach would most likely narrow this window, although precision is ultimately limited by the 1-hour period.

To understand how our system can automatically identify absence of OS checks, Figure 11.1 shows traffic periodicity with and without OS update checks. At the 16th level of decomposition of Figure 11.1(b) we see OS update polling appear as energy at the base period of one hour (3600s, two adjacent 3% bins), and as harmonics at half, three fourths and one and a half times the frequency (6553s, 4% energy, 4800s, 4% and 2400s, 3% energy). Disabling updates, by contrast, shows no energy below the 14th level of decomposition (Figure 11.1(c)).

Our algorithms detect periodicity automatically with an adaptive threshold. Figure 11.1(a) shows the numeric comparison corresponding to the whole 72 hour observation. As we can see, each of the periodicities that are visible in Figure 11.1(b) are above the detection threshold in Figure 11.1(a). More importantly, because detection is numeric, it can be automated and more sensitive and consistent than human interpretation.

This example demonstrates that our method successfully identifies a periodic behavior, and can also identify when that behavior starts and stops. While in some cases system administrators may be able to directly monitor OS update polling if they have administrative access to the machines in question, we suggest our approach could be useful when only network access is possible or desirable, for example, due to privacy reasons. In addition, monitoring periodic checks is robust to a potentially changing set of servers hosting OS updates.



(a) Automatic detection of OS updates: energy vs. detection threshold, all 72 hours.

(b) Traffic with automatic polling for OS updates.

(c) Traffic without automatic update polling.

Figure 11.1: Visualization illustrating periodic behavior before and after removal of OS update checks.

1.3.2. Detecting a Keylogging Application

OS updates are an example of desirable periodic behavior. We next look at an example of a periodic behavior which is undesirable, namely a keylogger. Many keyloggers report on user activity at specified intervals, to inform their masters what they have learned and that they are still operational—we confirmed supervisor-configured reporting intervals in both SpyBuddy and Keyboard Guardian.

Experiment: install laptop w. Keyboard Guardian. To investigate if we can detect keylogger reporting we installed Keyboard Guardian on a dedicated Windows computer. We monitored all TCP flows from the test machine for a three day period while using the test machine for occasional e-mail and web browsing.

On the second day of the experiment, we installed Keyboard Guardian at 4pm, and configured Keyboard Guardian to email reports every three hours. Our computer use compared to keylogger reporting resulted to an SNR of 0.1. Figures showing these results are omitted here due to space, but are available in our technical report [14].

We ran our system on trace files collected from the test machine, which correctly identified not only the periodic behavior but also the frequency of the reporting period (10,800s, 92μHz). Additionally, the system identified the presence of the signal between 12pm and 9pm on the second day of our experiment, correctly bracketing the 4pm installation time.

This experiment shows we can detect low rate but regular traffic as well as *changes* in periodic communication associated with a known spyware tool. We anticipate that this approach could be used by a network administrator to monitor a large number of user machines, searching for malicious activity. Although centralized companies could do such monitoring more easily by modifying software individual machines, some companies (for example, Google) and most ISPs do not have this ability. While such network monitoring is possible today with centrally maintained blacklists, our approach detects behavioral changes that would apply to malware before the control site is blacklisted. After detection, network administrators could take action to further investigate, perhaps notifying the machine's owner or subjecting that host to more invasive monitoring or quarantine.

2. Correlating Spam Activity with IP Address Characteristics

It is well known that spam bots mostly utilize compromised machines with certain address characteristics, such as dynamically allocated addresses, machines in specific geographic areas and IP ranges from AS' with more tolerant spam policies. Such machines tend to be less diligently administered and may exhibit less stability, more volatility, and shorter uptimes. However, few studies have attempted to quantify how such spam bot address characteristics compare with non-spamming hosts. Quantifying these characteristics may help provide important information for comprehensive spam mitigation.

In this work, we use two large datasets, namely a commercial blacklist and an Internet-wide address visibility study to quantify address characteristics of spam and non-spam networks. We find that spam networks exhibit significantly less availability and uptime, and higher volatility than non-spam networks. In addition, we conduct a collateral damage study of a common practice where an ISP blocks the entire /24 prefix if spammers are detected in that range. We find that such a policy blacklists a significant portion of legitimate mail servers belonging to the same prefix.

For brevity, we present only the results of the last part of our work in this report. Our full results can be found in the paper that appeared in Global Internet 2010.

2.1. Collateral Damage

Our prior work has shown that both address and hostname characteristics confirm that spam originates from dynamic addresses. We use these results to consider a new question: Is blacklisting an entire /24 prefix based on the presence of one or more spamming hosts an effective policy? While many blacklists enumerate individual IP addresses, blocking entire /24 prefixes are also common. We are concerned about reducing spamming, but also about the blocking of legitimate outgoing email. We define collateral damage as the number of legitimate mail servers which would be incorrectly filtered when an entire /24 prefix is blacklisted. First we identify all survey prefixes which have spamming hosts. If these prefixes also contain non-spamming hosts, then they are subject to collateral damage. Figure 11.2 compares the number of spammers versus non-spammers in the set of intersected prefixes.

Except for outliers, the graph shows that many of the prefixes seem to cluster along the left-axis (grey) or the top diagonal (black). The diagonal is present because the sum of spammers and non-spammers is never more than the size of a prefix (255). The left-axis cluster shows prefixes with a reasonably uniform number of non-spammers and a small number of spammers. The diagonal cluster shows a large number of spammers residing in highly populated prefixes. These clusters may reflect two different situations. The diagonal cluster shows heavily compromised prefixes, which we believe may have negligent administration or a collaborating provider. The other cluster represents a limited number of compromised hosts in an otherwise normal prefix, we believe these may be caused by bots. The latter are prone to collateral damage, since they contain a high number of non-spamming hosts and a low number of spamming hosts.

Anti-spammers typically assume there is no collateral damage in blacklisting a /24 prefix, because many ISPs forward legitimate mail through the ISPs mail server, rather than allowing hosts to send mail directly. We are only able to quantify whether a blacklisted prefix contains mail sources, by studying their hostnames and DNS mail forwarding records. For this study we extract the reverse hostname and MX record for each address in the prefix, using the Linux `host` and `dig` commands. Finally we intersect the mail server IP addresses against the survey dataset to see if any reside in the blacklisted prefixes. Table 11.3 shows the progression of our data analysis.

We start with 646,040 addresses that reside in the 4,126 spamming prefixes in our intersection set. We

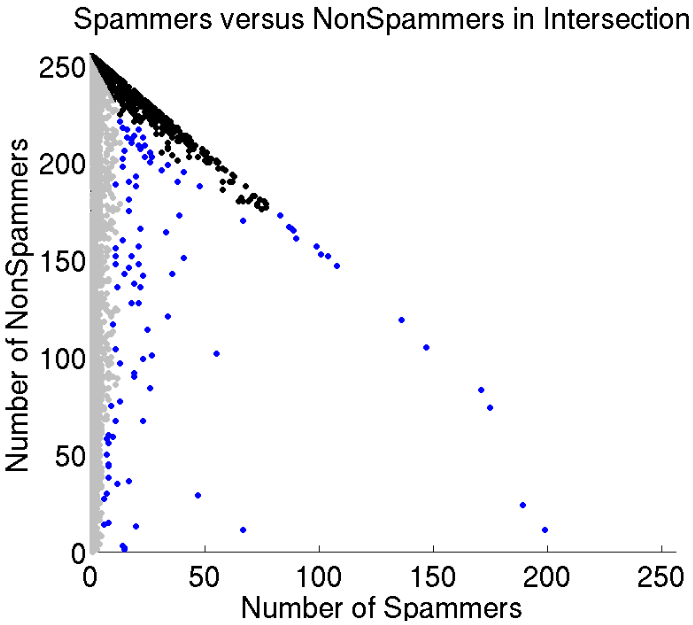


Figure 11.2: Number of Spammers versus Non-Spammers

Table 11.3: Collateral Damage Study

DESCRIPTION	DOMAINS	HOSTS	PREFIXES
INTERSECTED PREFIXES		646,040	4,126
DOMAIN QUERY TIMEOUT		12,899	
DOMAIN QUERY INVALID		175,535	
DOMAIN QUERY VALID		457,606	
UNIQUE DOMAIN NAMES	4,044		
NUMBER MAIL SERVERS		6,718	
UNIQUE MAIL SERVERS		3,872	2,154
COLLATERAL DAMAGE		1,377	365

subtract addresses that timeout or fail to return a valid domain name. From the remainder, our programs identify a set of unique domain names, and the addresses of the corresponding mail servers. Intersecting these addresses with our spamming prefixes, we find collateral damage of 1,377 mail servers and 365 prefixes, which is ∼8.8% of all spamming prefixes. We conclude that prefix blocking incurs a high rate of collateral damage, suggesting the need for finer-grain filtering.

Chapter 12

Application of Quickest Changepoint Detection to Information Assurance and Cybersecurity

This chapter is intended to assess the progress made in *applied* changepoint detection. Specifically, we considered two major areas of application: information assurance and cybersecurity. The work in both was a joint effort of Dr. Tartakovsky (University of Southern California, Department of Mathematics), Dr. Papadopoulos (Colorado State University, Department of Computer Science) and Dr. Heidemann (University of Southern California, Department of Computer Science and Information Sciences Institute).

1. Introduction

One of the important applications that stimulated the research in this project related to development of efficient distributed changepoint detection methods is *intrusion detection in distributed high-speed computer networks*. A significant number of serious cyberattacks on a variety of governmental agencies, universities, and corporations have been identified [78]. These attacks, including a variety of buffer overflows, worm-based, denial-of-service (DoS) and man-in-the-middle (MiM) attacks, are designed to gain access to additional hosts, steal sensitive data, and disrupt network services. As a result, *rapid detection* of a wide spectrum of network intrusions and *robust separation* of legitimate and malicious traffic are vital for the continuation of normal operation of military, federal, industrial, and enterprise networks.

There is a wide variety of intrusion detection methods proposed in the literature [43]. There are two broad IDS categories: (a) signature based and (b) anomaly based [43, 78]. The main problem with signature IDSs is that the signatures must be defined *a priori*; thus, this technique is ineffective against new attacks. In anomaly-based detection, the IDS is first trained to recognize "normal" traffic patterns and then classifies deviations as attacks. The problems with this approach are a high rate of false positives, the cost of training and re-training, and susceptibility to carefully crafted attacks that "train" themselves into normal traffic.

Typically network intrusions occur at unknown points in time and lead to changes in the statistical properties of certain observables. For example, distributed DoS (DDoS) attacks lead to changes in the mean value of the number of packets of a particular type (TCP, ICMP, or UDP) and size. It is therefore intuitively appealing to formulate the problem of detecting attacks as a quickest changepoint detection problem: to detect changes in statistical models as rapidly as possible (i.e., with minimal average delays) while maintaining the false alarm rate at a given low level.

In this project, we developed not only an efficient anomaly IDS based on changepoint methods, but also a hybrid anomaly-signature IDS which allows for filtering of false positives and confirmation of real attacks. Thus, this novel hybrid approach allows us to overcome common drawbacks of both anomaly and signature methods when applied separately.

2. The Hybrid Anomaly–Signature Intrusion Detection System

2.1. The Idea and Structure of the System

In this project, we proposed a novel hybrid approach to network intrusion detection which is particularly efficient for capturing DDOS attacks. The hybrid anomaly–signature Intrusion Detection System (IDS) implements the change detection algorithm (anomaly IDS) and the spectral-based signature IDS in parallel. In other words, the methodology is based on using the changepoint detection method for preliminary detection of attacks, and discrete Fourier transform to reveal periodic patterns in network traffic which can be used to confirm the presence of attacks and reject false detections triggered by the anomaly IDS. It is worth mentioning that in network security applications it is of utmost importance to detect very rapidly attacks that may occur in a distant future (using a repeated application of the same anomaly-based detection algorithm), in which case the true detection of a real change may be preceded by a long interval with frequent false alarms that should be filtered (rejected) by a separate algorithm, which may be built based on signatures (e.g., spectral signatures). At the second stage, we propose to exploit a spectral-based IDS.

The architecture of an automated two-stage (cascade) hybrid anomaly–signature IDS is shown schematically in Figure 12.1. The IDS utilizes changepoint detection for preliminary intrusion detection, and discrete Fourier transform to confirm true and reject false intrusions. Such a *hybrid* approach simultaneously speeds up detection and lowers the frequency of false alarms.

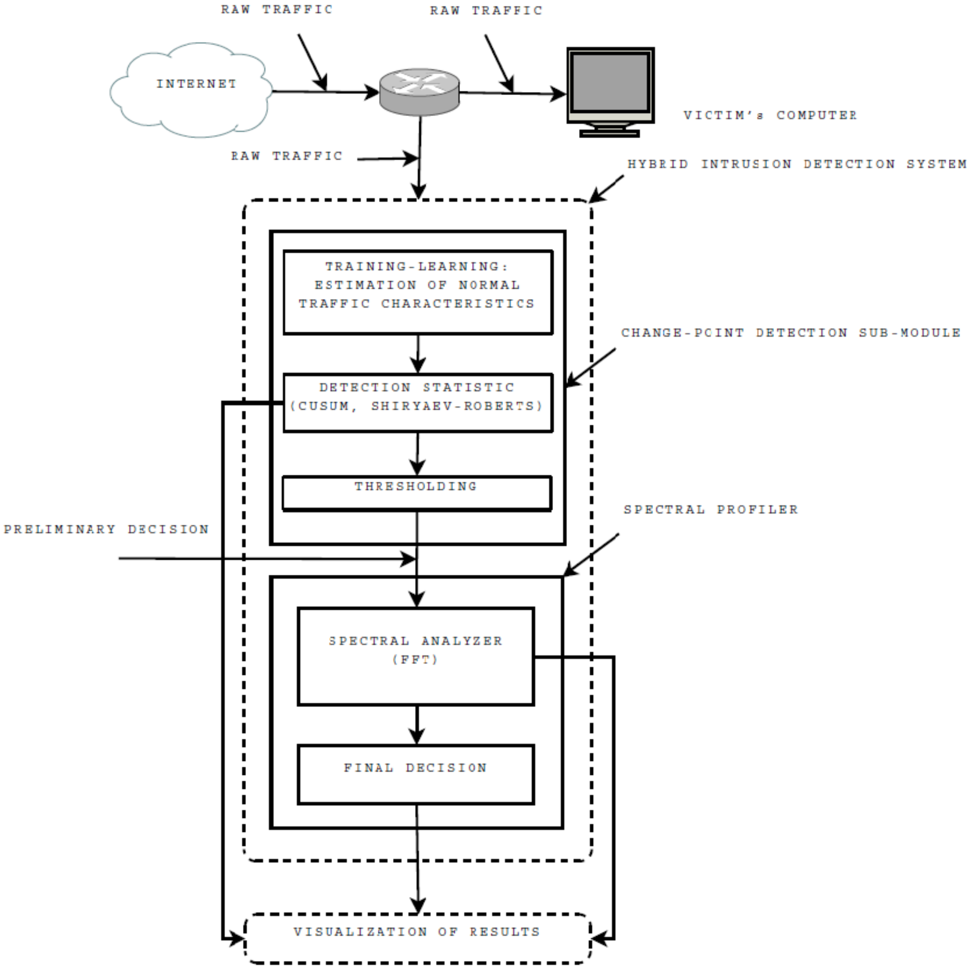
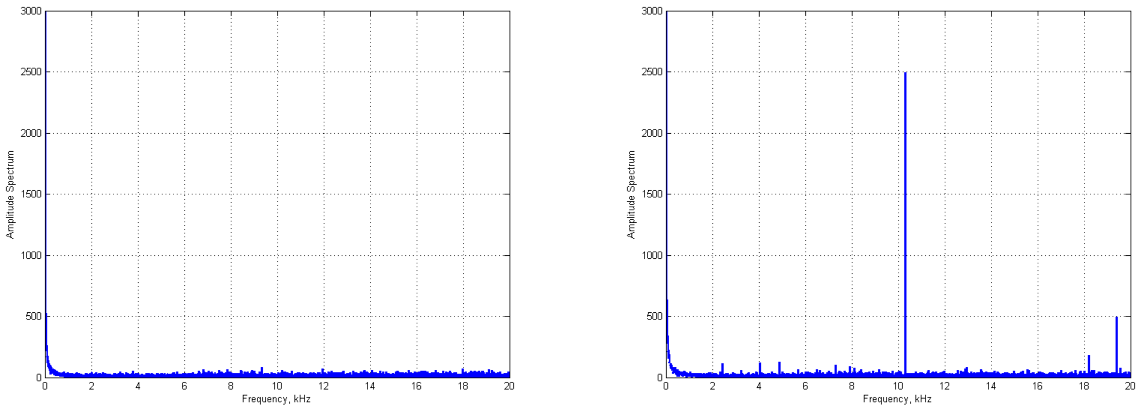


Figure 12.1: The hybrid intrusion detection system.

To illustrate how the hybrid IDS works, consider detecting a DoS attack. A DoS attack is a malicious attempt to disrupt (ideally – completely knock off) an online service. This can be achieved, e.g., by sending a large number of packets to the target (victim) to congest its link. Consequently, once the victim's link is overloaded, it starts to clock attack packets in regular intervals. For example, trying to push through more than 10Mbps of traffic out of a 10Mbps link will clock out packets at approximately 800 packets per second, if the packets are 1500 bytes in size each. This periodicity, though mixed with other non-attack traffic towards the target's network, will result in an easily detectable spike in the spectrum at the corresponding frequency. At the same time, since DoS attacks lead to changes in the statistical properties of traffic date, changepoint detection can be effectively used to detect these changes. The idea of the hybrid IDS is to use changepoint detection as an "early warning" system, and once it sounds a alarm, turn on the spectral analyzer for a more thorough traffic analysis.



(a) No attack (pre-change)

(b) Under attack (post-change)

Figure 12.2: Power spectral density.

Figure 12.2 gives an example of implementation of FFT (fast Fourier transform) for a real data set containing an attack. Note that in the no-attack mode there are no periodic patterns in the traffic distribution, and hence, there is no peak in the spectrum, while under the attack there is a contrast peak suggesting that this might indeed be an attack. Namely this phenomenon is used to filter false positives and confirm true attacks. Separately anomaly- and signature-based IDS's have pros and cons. Combining them in one unit allows us to obtain the best possible performance.

In summary, the hybrid anomaly–signature IDS is based on the following principles and has the following features:

- **Anomaly IDS – Quick Detection with High FAR:** In order to detect attacks quickly, detection threshold in the changepoint detection module are lowered, which leads to frequent false alarms that should be filtered by a separate algorithm.
- **Signature IDS – False Alarm Filtering:** To reject false detections a signature-based approach is used based on a spectral analysis module.
- **Changepoint Detection Module for:** (a) Quick detection with relatively high FAR, and (b) Triggering of spectral analysis algorithms.
- **Spectral Analysis Module for:** (a) False alarm filtering/rejection; and (b) True attack confirmation.

Figure 12.3 illustrates the hybrid anomaly-spectral IDS in action. The first plot shows raw data (packet rate). The second plot shows the behavior of the CUSUM statistic, which is being restarted from scratch (repeated) when a threshold exceedance occurs. The third plot shows PSD (power spectral density) at the output of the spectral analyzer: the peak appears only when the attack starts (which confirms the attack), while previous threshold exceedances (false alarms) are rejected by the spectral analyzer.

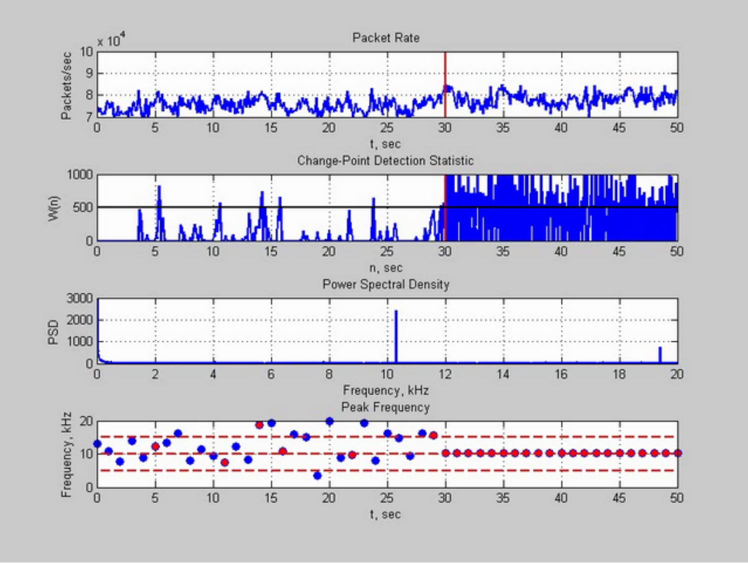


Figure 12.3: Hybrid IDS in action.

2.2. Detection of DoS Attacks

In this subsection, we illustrate efficiency of the hybrid IDS for detecting UDP Packet Storm Attacks, one of the common DoS attacks. Specifically, we report and discuss the results obtained from the empirical performance analysis of the hybrid IDS using a real-life distributed DoS attack, namely, a packet storm attack on User Datagram Protocol (UDP) port 22. This trace was captured off one of the ServePath[1] networks. The attack starts about 60 seconds into the trace, and consists of very short packets (about 15 bytes in size each) sent to the victim's UDP port 22. The rate is about 180Kpps with that of the background traffic being about 53Kpps. Although intensity-wise it is a rather contrast attack, it is quite short in duration – only about 10 seconds long. This poses a challenge for the hybrid system.

Figure 12.4 shows the cumulative packet rate. It is seen that there is a considerable jump in the packet rate at the time moment the attack begins.

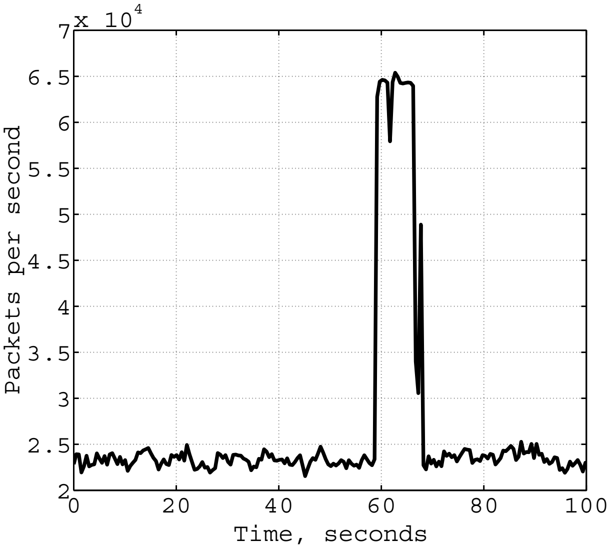


Figure 12.4: Storm attack on UDP Port 22 (cumulative packet rate).

[1] See www.servepath.com for more information.

Figure 12.5 shows the corresponding FFT output. The contrast peak in the middle of the plot suggests that this might indeed be an attack. It is this phenomenon that we use to filter false positives in the hybrid system.

We now turn into discussion of how the proposed system can be used to also isolate shorts attacks. Figure 12.6 illustrates the difference between not using the spectral analyzer and using it to confirm the attack. From Figure 12.6(a) we see that the first attack is detected while the second one is not. This is because we did not rely on the spectral analyzer and had to use high detection threshold. Consequently, we had almost no false alarms but we also failed to detect the second attack. At the same time Figure 12.6(b) shows the case where we lowered the detection threshold, which increased the level of false alarms, but the false alarms were successfully filtered by the spectral analyzer. As it can be seen, in this case we detected and isolated both attacks.

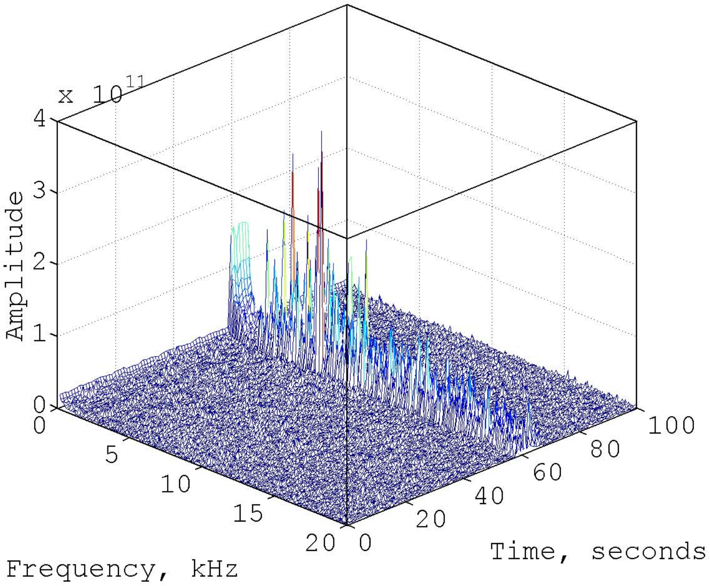
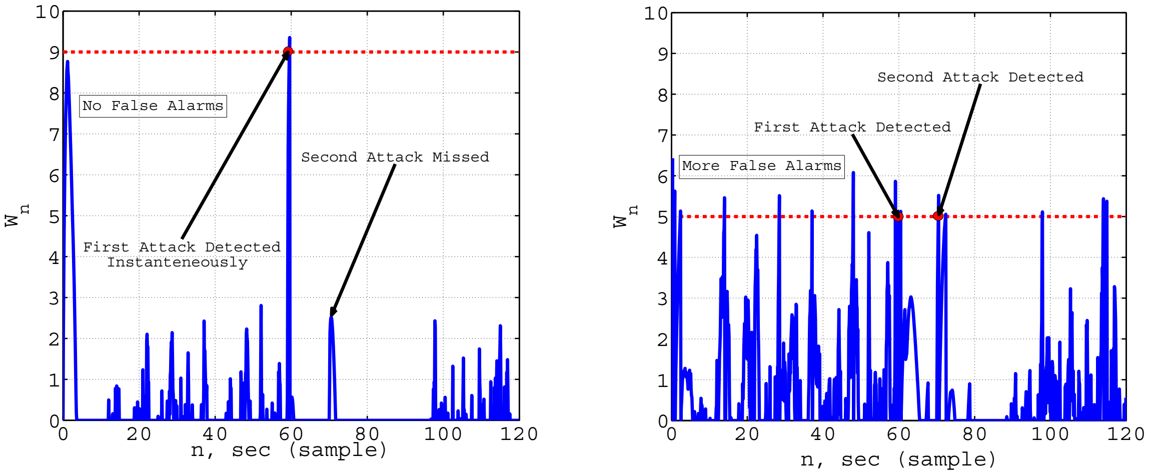


Figure 12.5: Spectral density for the UDP attack.



(a) No detection

(b) Successful detection

Figure 12.6: UDP attack double peak detection.

3. Application to Spam Detection: Fighting Spam at the Network Level

Spam is a well-known problem, and the need for an efficient way to shield against spam is evident. As an anti-spam solution, most organizations run some version of spam filters at their local networks (e.g., Bayesian filters or some sort of block lists). These techniques work quite well, but they are typically expensive both in initial and operational costs. Block lists rely on information that was gathered ahead of time, and thus might be stale. Bayesian approaches, while generally good, are not infallible and require examination of all message content.

The idea of our approach is to monitor traffic at the network level. This has several advantages: it requires no message content examination and thus guards privacy; spammers can be detected almost instantly based on their network behavior; collateral damage is reduced because dynamic addresses released by spammers can be removed from block lists quickly; and IP addresses can be blocked before connections are accepted, saving resources at the mail server.

What features are useful for detecting spammers? Prior work has shown that the Autonomous System the IP address belongs to, the message size, the number of blocked connections and the message length are important features. All of them can be determined from network traffic. We investigated such features and used changepoint detection to detect when traffic patterns from a particular host match known spammer patterns. Examples include the following:

1. ***Message size.*** Most spam campaigns attempt to deliver either identically the same or similar (content-wise) message to many recipients. As a result, with the exception of the receiver's IP address, the size of the message does not vary significantly. Thus, detecting hosts that send email blocks of a similar size is one feature to look for. False positives (e.g., from mailing lists) can be eliminated by looking at other features such as past history and the presence of MX records associated with that IP address.
2. ***Dropped connections.*** If a mail server uses a block list to refuse connections from suspected spammers, these will be detected in network traces. Keeping track of such events can help detect spammers for everyone. Changepoint detection can be used to detect a change in the number of dropped connections from a particular IP address.
3. ***Connection patterns.*** So as to lower the probability of being detected, spammers typically send very few emails to a particular domain. With network monitoring, however, which can monitor many domains at once, this particular pattern can be detected. Changepoint detection can be used to detect a spammer touching many different domains.

We analyzed a number of network traces containing spam-related activity in order to better understand the signature message size pattern (data mining) to subsequently use in a changepoint based detector to defeat spam. Figure 12.7 summarizes our findings.

Specifically, Figure 12.7 shows the evolution of the email size in time (for real-world data). The main observation is that the email size, though has occasional spikes, most of the time is flat. A host having such a pattern is a potential spammer. We equipped a changepoint based detector with the ability to check for such patterns to confirm that a certain host is a spammer. We tested the whole system using an example with a real-world spammer. The whole process is shown in Figure 12.8. Specifically, under surveillance is the email (SMTP) traffic generated by a certain host. Ordinarily, the SMTP traffic produced by a user sending legitimate messages is characterized by a relatively steady intensity, i.e., the number of messages sent per unit time remains more or less constant, with no major bursts or drops. However, the behavior changes completely once a spam attack is initiated: the number of messages sent off explodes, possibly for a very short period of time. The topmost plot in Figure 12.8 illustrates just this. The spike in the traffic intensity that appears in the far right of the plot can be detected by methods of statistical changepoint detection. We considered two most popular detection procedures – the CUSUM and Shiryaev–Roberts procedures. The middle and bottom plots in Figure 12.8 show the detection process for each of the two procedures. Either procedure momentarily rises an alarm as soon as the traffic intensity blunder caused by the spam attack is encountered.

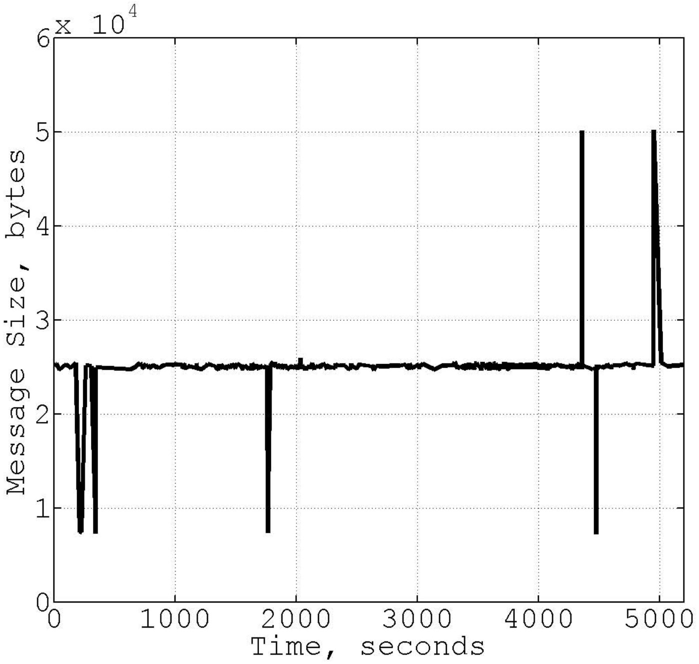


Figure 12.7: The message size pattern of a typical spamper.

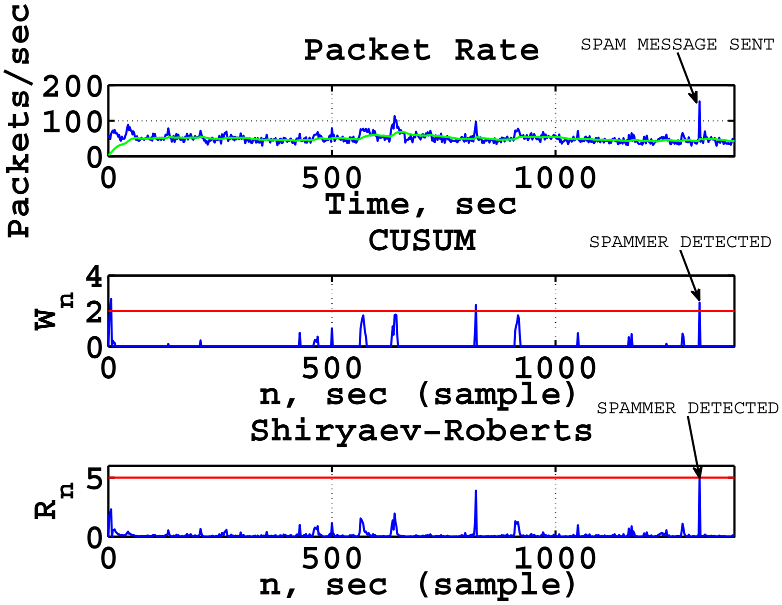


Figure 12.8: Detection of a spammer.

At the point the alarm is sounded, the system checks the message size pattern, and based on the fact that the message size is fairly stable (standard deviation is very small) concludes that this is a legitimate alarm.

4. Application to Repulse Unauthorized Break-Ins

Yet another major type of computer security risk is when a system (whether an individual computer, or an entire network) is broken into by an unauthorized party. An event of this kind, i.e., an unauthorized break-in to a system, is associated with gaining access to a machine without permission to subsequently temper with it. Some of the *short-term* consequences of a successful unauthorized break-in include:

1. Stealing sensitive data;
2. Turning the machine into a relay to send junk email (spam);
3. Bringing down the entire network just for the fun of it.

Long-term, the consequences may snowball to a matter of national security. The need to device a defensive mechanism is thus apparent. We proposed an efficient solution against such threats.

To illustrate our approach, consider a hacker seeking to break into a machine. This process may be described as consisting of two phases. During *Phase 1*, the hacker launches a dictionary attack (see below) attempting to guess a valid username-password combination (ideally, the root one), to subsequently use it to get through to the machine's shell; in real life, the hacker may obtain a valid username-password combination via other means, such as through phishing, social engineering, etc. Next, assuming the hacker was successful in gaining access to the machine's shell, the break-in process rolls on to *Phase 2.* During the latter, successfully logged-in, the hacker proceeds to "playing" with the machine. The specific activities very from case to case, but common to most is downloading malware onto the machine and opening a backdoor (e.g., for a possible accessory as well as for later returns). Thus, an unauthorized break-in is a two-phase process, where each phase is characterized by its own unique features in terms of the traffic patterns it generates. Therefore, an individual approach to detection of each is required. Both can be designed using a changepoint detection based anomaly IDS, or using the hybrid anomaly–signature IDS. However, the problem with simply employing two independent detectors is high level of false positives: it is the level of the first detector plus that of the second one. To overcome this, observe that the two phases are co-related: the traffic pattern generated by Phase 1 activities *is followed* by the pattern generated by Phase 2 activities. Factoring this correlation in helps reducing the overall frequency of false positives.

We performed a simulation of an unauthorized break-in to evaluate the performance of our approach. The test-bed was half real traffic, and half simulated. Specifically, we modeled Phase 1 using real-life traces. However, since a trace of a real attack where a hacker has actually broken into a machine is hard to obtain, we simulated Phase 2. We now describe each phase separately, beginning with the first one.

4.1. Phase 1: Dictionary Attack

Recall that with a dictionary attack the hacker attempts to guess a correct username-password combination to break into a server, typically through SSH. While we illustrate the attack with SSH, the attack applies to any username-password access control method, including web authentication and other similar methods. To achieve this goal the attacker initiates what is essentially a brute force attack: a rapid sequence of SSH authorization requests, where each request contains a username-password combination either guessed based on prior partial information about the real username-password, or trying out common user names and passwords. The word "dictionary" in this context is used figuratively to illustrate that the attacker has a list (dictionary) of "suspected" username-password combinations. In a dictionary attack the hacker successively tries all of them. Figure 12.9 illustrates this kind of an attack schematically.

Consider an example of a real-world SSH dictionary attack. This dataset was provided by a regional ISP. Figure 12.10(a) shows the intensity of the number of packets passing through the victim server's link. Notice that the server remains idle most of the time, occasionally exhibiting interaction with other computers in the network. Eventually, the server starts to receive suspiciously many SSH requests. Figure 12.10(b) shows the behavior of the CUSUM detection statistic. The behavior of the latter is very similar to that of the packet rate. As soon as the server is under attack, the statistic jumps through the detection threshold (red flat line), and an alarm is raised resulting in successful detection of the attack.

The instantaneous and successful detection described above is not a miracle because the attack is very contrast. Therefore, it is not a good illustration of the potential of changepoint detection. To make things more challenging, we intensionally diminished the intensity of the attack. The new dataset is excellent to demonstrate not only the potential of changepoint detection, but also that of the hybrid system. The spectral approach is expected to work because, as mentioned earlier, dictionary attacks introduce periodicity in the

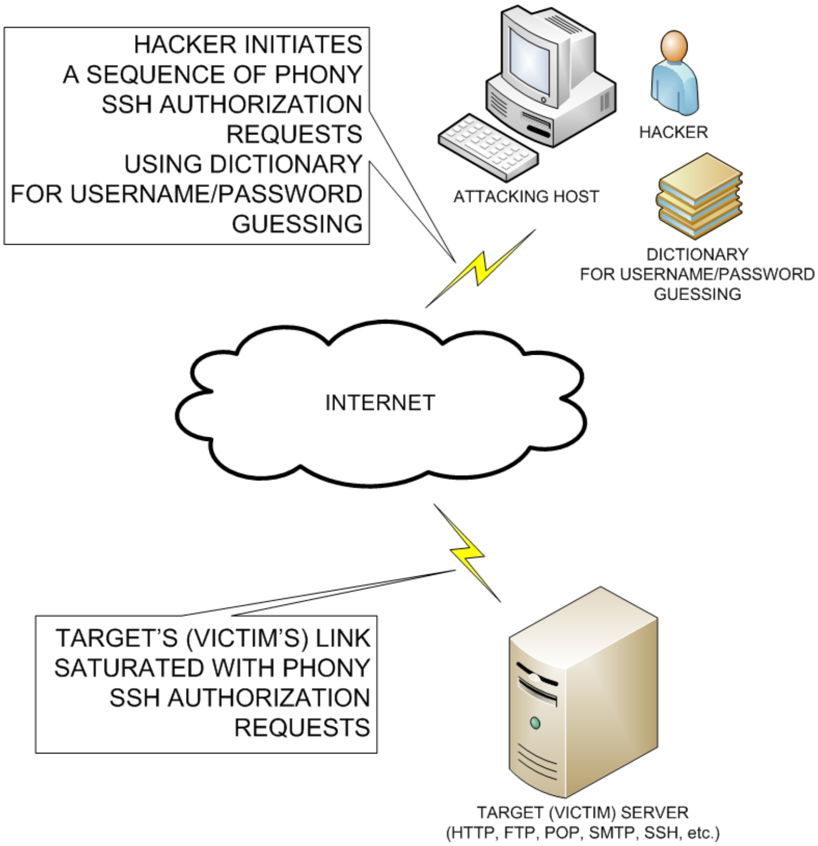
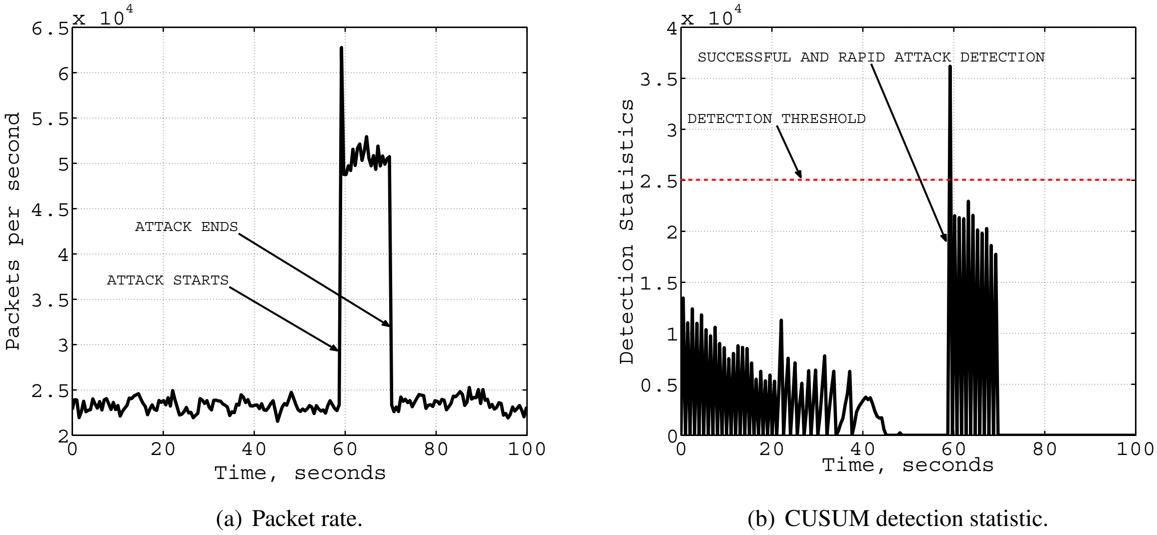

HACKER INITIATES A SEQUENCE OF PHONY SSH AUTHORIZATION REQUESTS USING DICTIONARY FOR USERNAME/PASSWORD GUESSING
ATTACKING HOST
HACKER
DICTIONARY FOR USERNAME/PASSWORD GUESSING
INTERNET
TARGET'S (VICTIM'S) LINK SATURATED WITH PHONY SSH AUTHORIZATION REQUESTS
TARGET (VICTIM) SERVER (HTTP, FTP, POP, SMTP, SSH, etc.)

Figure 12.9: Generic dictionary attack scenario.



(a) Packet rate.

(b) CUSUM detection statistic.

Figure 12.10: SSH dictionary attack traffic pattern and its detection.

traffic flow. Figure 12.11 illustrates this for this attack. Specifically, the figure is a magnified version of the patterns in the attack traffic.

Looking at this figure we see a highly periodic sequence of contrast spikes. If one employs a spectral analyzer, the spectrum power density will have a high spike. Recall that this is exactly the idea behind the hybrid system. We now report our results for the new dataset. Figure 12.12 illustrates the detection

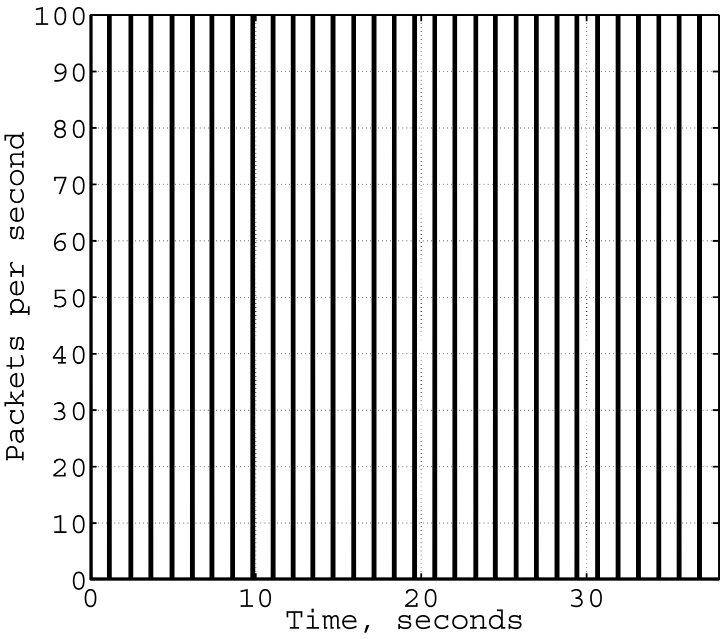
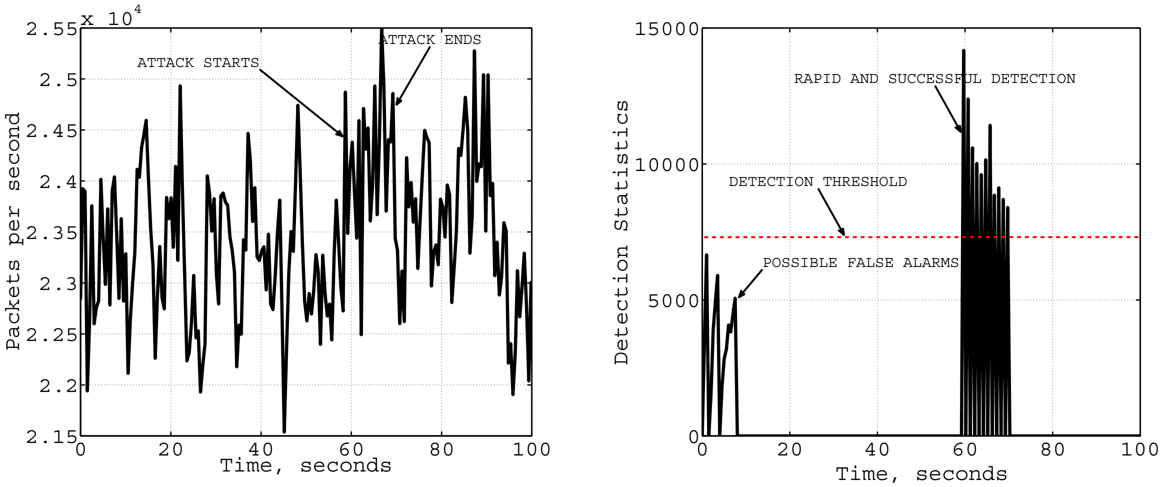


Figure 12.11: SSH dictionary attack signature.

process. The main conclusion is that despite the fact that the intensity of the attack now is far less than before, changepoint detection reveals it instantaneously anyway.



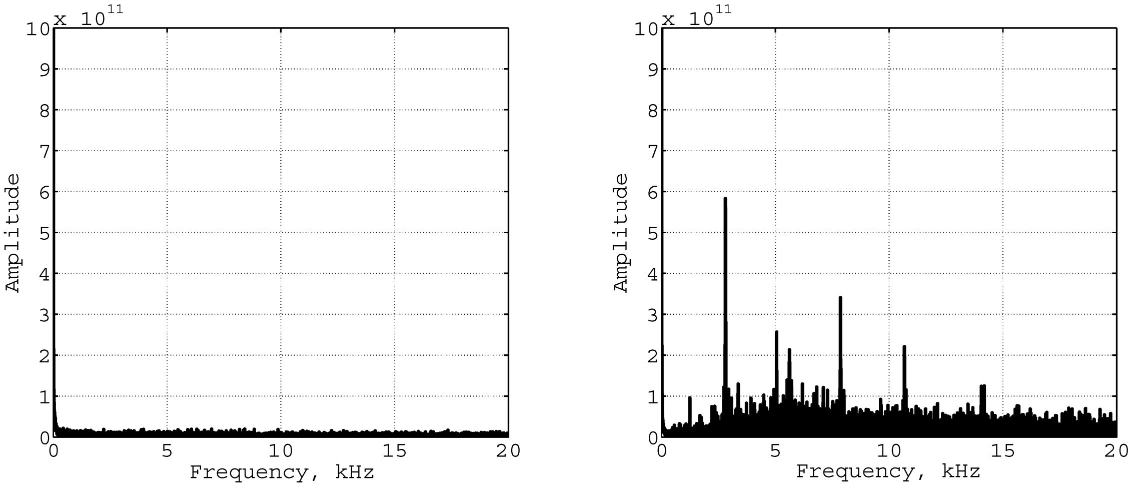
(a) Packet rate.

(b) CUSUM detection statistic.

Figure 12.12: Diminished SSH dictionary attack traffic pattern and its detection.

The problem though is that if the detection threshold is lowered so as to have an even more quick detection, it will inevitably result into numerous false alarms. To filter these false alarms, one can employ FFT to uncover hidden periodicities in the traffic flow caused by the presence of attacks. Figures 12.13(a) and 12.13(b) show the power spectral density for this dataset before and during the attack. Note the spike in the PSD under attack. This is exactly because of the aforementioned periodicity.

To conclude we remark that using the anomaly–spectral IDS, one can achieve unprecedented speeds of detection.



(a) Before attack.

(b) Under attack.

Figure 12.13: Spectral characteristics of the traffic before and during the attack.

4.2. Phase 2: Post Unauthorized Break-In Activity

When trying to detect a compromised machine by looking at network traffic only, challenges are high. Due to privacy concerns we assume we cannot look inside packets to determine what traffic is malicious. Moreover, even if we had access to the packet payload, many machines are compromised through applications such as SSH, which means that the payload is encrypted. We simulate an attack scenario where a hacker breaks into a machine using a compromised SSH password. Note that this covers successful dictionary attacks (such as those we see in Phase 1), and stealing username-password information. We assume that the hacker performs several suspicious activities that normal users are not likely to perform. These include downloading a malicious binary file and running it, fetching a few more binaries from an external server, creating a backdoor for future connections into the compromised machine, and uploading a few files to an external server. We take into account that some of these activities such as access to the backdoor, may happen after the hacker has logged out.

Although this scenario is generic, it is fully customizable: the hacker activities can be altered, timing between commands can be changed, and as a result, any scenario we deem plausible can be simulated. Currently, the simulation consists of the following steps:

1. Create an attack script. This is a set of shell commands representing what the attacker will do once logged into the machine.
2. Create traffic to the machine. This shell script is run after compromise, and accesses the back door the original attack script has created.
3. Run a network capture tool on the machine to capture all attack traffic (we use `tcpdump`).
4. Run the attack scripts.

The above scenario was implemented, resulting in a trace about 38 seconds with about 35K packets. The target machine's IP address is `129.82.138.26`. The machine is assumed to be running such standard network services as HTTP, FTP, POP, SMTP, SSH, etc. These services typically run on ports whose number is less than 1024. Backdoors, however, typically run on higher port numbers. A big challenge is to detect the *correlation* between the SSH communications and deviations (which may be very slight) in the number of traffic packets in and out of the machine. In our scenario, an SSH login is followed by an increase in the number of incoming or outgoing packets, which is a strong indication of suspect activity by the user who just logged in. Thus, not only do we want to detect that the behavior of the traffic flow (both incoming and outgoing) generated by the machine has changed, but also correlate the change and a particular login through SSH.

The surveillance begins from the very first SSH packet sent by the hacker in an attempt to log-in to the machine. Our approach is to treat every connection as suspicious, and create state to monitor all successful connections. In this simple scenario it takes the hacker about 5 seconds to start "fooling around" with the machine: 5 seconds after the beginning of the surveillance, the hacker executes a command that causes the machine to increase its traffic. This increase is readily and successfully detected by the changepoint detection based anomaly IDS, as seen in Figure 12.14. Note that the scenario involves both in- and out-coming traffic, which can be used to reduce false positives.

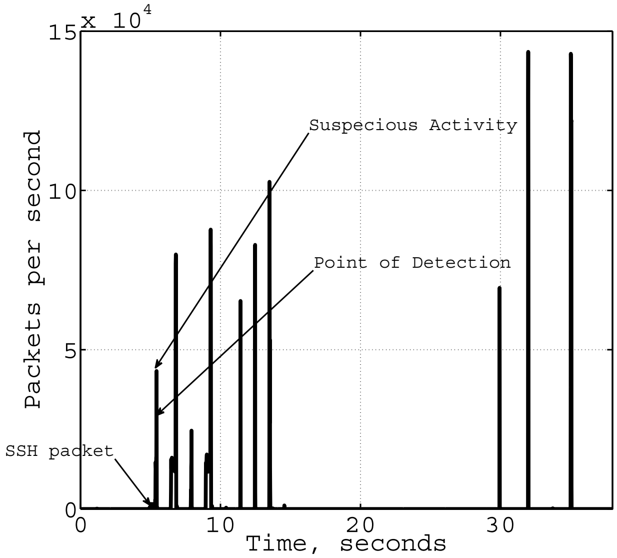


Figure 12.14: Detection of unauthorized SSH break-in attempt.

5. Conclusion

Rapid intrusion detection in high-speed computer networks with low false alarm rate is a challenge for military, government and industrial networks. This problem has been addressed by various agencies, universities, and companies for many years. Still this problem is not solved.

We applied quickest changepoint detection methods for the development of efficient anomaly-based IDS-s that are capable for detecting attacks in computer networks with small detection delays for a given (low) false alarm rate. The effectiveness of the proposed changepoint detection based IDS has been verified by implementation in real-word scenarios for detecting ARP MiM insider attacks and DDoS TCP, ICMP, and UDP external attacks as well as for detecting spam and unauthorized break-ins. These results prove that major drawbacks and technological barriers of existing intrusion detection systems can be overcome through the use of a completely new approach to intrusion detection that relies on advanced changepoint detection methods. In particular, replacing current *ad-hoc deterministic* decision rules with advanced changepoint detection algorithms allows for controlling false alarm rate and detecting unexpected intrusions.

In addition, combining an anomaly IDS and spectral-based signature algorithms with false alarm filtering capability allows one to lower thresholds in the anomaly IDS, which reduces detection delays to a minimum. Therefore, the proposed approach to rapid intrusion detection integrates best possible anomaly-based (statistical) solutions with a spectral-based signature IDS in distributed systems. The feasibility of the proposed hybrid IDS has been proven for detecting DDoS attacks by using LANDER data sets collected by ISI. We believe that the proposed hybrid anomaly-spectral IDS solves both aspects of the problem – it simultaneously provides breakthrough in terms of achieving unprecedented speeds of detection (i.e., small detection delays of true attacks) and a very low false alarm rate.

Chapter 13

Application of Adaptive Spatiotemporal Image Processing and Nonlinear Filtering Methods in Remote Sensing

The work reported in this chapter has been performed mainly by Alexander Tartakovsky. The results related to NLF-based track-before-detect are joint work with Boris Rozovsky.

1. Introduction

The problem of efficient clutter rejection is a challenge for Space-Based Infrared (SBIRS) and Space Tracking and Surveillance System (STSS) sensors with chaotically vibrating lines-of-sight (LOS) that have to provide early detection and tracking of targets (e.g., missile launches) in the presence of highly-structured cloud backgrounds. In such systems, the intensities of cluttered backgrounds due to solar scattering by clouds, aerosols and earth surface (ground, sea, etc.) or by IR airglow emissions are typically dozens and even hundreds of times greater than either sensor noise or the intensities of the targets that are to be detected and then tracked. As a result, reliable target detection and subsequent tracking is impossible without clutter rejection to, or even below, the level of sensor noise.

Most existing clutter rejection technologies for unstabilized or mechanically stabilized platforms rely on spatial-only or simple image differencing methods. However, the results of our study presented below show that even the best spatial-only filters are not efficient enough, especially in unfavorable conditions that are typical for applications of interest. Moreover, these filters cannot be combined with temporal processing in cases where clutter is non-stationary in time due to platform vibrations (jitter), which is always the case. A similar conclusion holds for an industry standard differencing method.

In this project, we argue that the level of clutter suppression required for reliable detection and tracking can be achieved only by implementing novel *spatiotemporal* image processing methods rather than spatial filtering alone. Note that in order to make temporal processing efficient, clutter rejection algorithms must be supplemented by very accurate *jitter estimation and scene stabilization techniques* that compensate for platform vibrations and eliminate residual frame misalignment. Our image registration/stabilization techniques are entirely different from those previously used. Stabilization is performed iteratively in the course of clutter rejection, and the corresponding stabilization algorithm is a part of the clutter rejection architecture. We show that this novel approach is extremely efficient allowing for very accurate interpolation and image reconstruction in a wide variety of conditions.

However, for a variety of moving platforms, one may expect difficult scenarios where even after clutter rejection the effective SNR is low, so that the in-frame detection with a given acceptable FAR is not possible. The only way to overcome this difficulty is to develop efficient track-before-detect architectures based on spatial-temporal NLF methods. These algorithms should process a series of frames simultaneously without detecting targets in each frame accumulating information along hypothetical tracks.

2. The Developed System

In this research, we focus on developing algorithms and software for adaptive clutter suppression, target detection and multiple target tracking for a variety of observation conditions; tuning and optimization of these algorithms for particular scenarios; and testing and validation through synthetic simulations and processing of real data. The primary goal is to develop a viable prototype of the multiple target tracking system that includes a reconfigurable, adaptive clutter rejection (CLUR) system that can be tested using a built-in simulator which mimics real environments. The developed system and corresponding software tools have the following functionalities and capabilities:

1. Built-in generator of image sequences with moving point illumination sources (targets), background clutter due to cloud cover, jitter due to platform vibrations, and sensor noise.
2. A bank of image processing algorithms (CLUR filters) with a reconfigurable architecture and ability to compensate for strong signals from bright targets and outliers.
3. Auto-tuning and auto-selection algorithms that allow for automatic selection of the optimal filter from the bank for current conditions.
4. In-frame detection algorithms with constant false alarm rate (stabilization of false alarms).
5. ONF-based multitarget tracking algorithms, in particular:
   (a) Changepoint detection-based track initiation algorithms
   (b) Identification of detections as belonging to existing tracks
   (c) Forming new tracks and deletion of false tracks
   (d) Changepoint detection-based track termination
6. Graphical user interface (GUI) for visualization of the results of processing and for input data and parameters.

A general block-diagram of the system with the corresponding data/signal processing flow is shown in Figure 13.1.

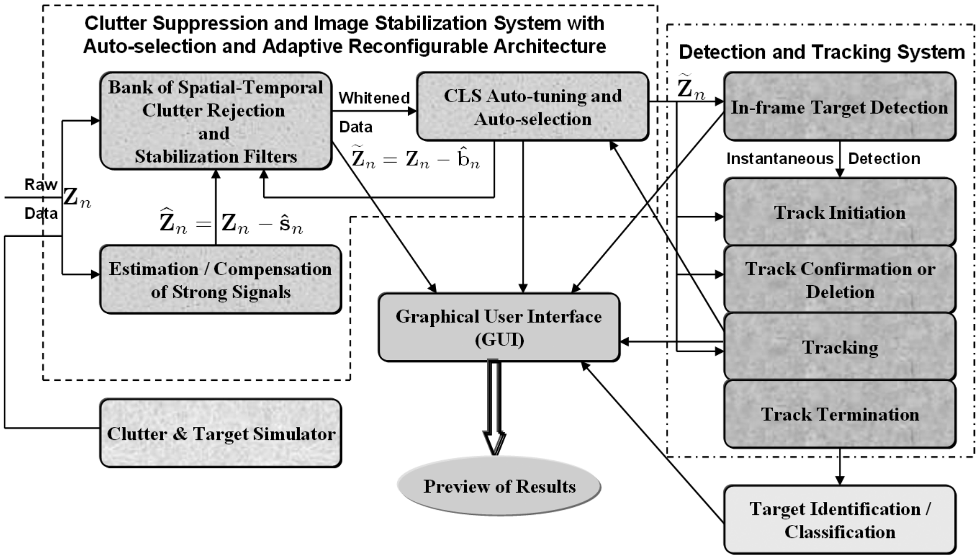

Clutter Suppression and Image Stabilization System with Auto-selection and Adaptive Reconfigurable Architecture
Bank of Spatial-Temporal Clutter Rejection and Stabilization Filters
Whitened Data
CLS Auto-tuning and Auto-selection
$\widetilde{Z}_n = Z_n - \hat{b}_n$
Raw Data
Z_n
$\widehat{Z}_n = Z_n - \hat{s}_n$
Estimation / Compensation of Strong Signals
Graphical User Interface (GUI)
Clutter & Target Simulator
Preview of Results
Detection and Tracking System
$\widetilde{Z}_n$
In-frame Target Detection
Instantaneous Detection
Track Initiation
Track Confirmation or Deletion
Tracking
Track Termination
Target Identification / Classification

Figure 13.1: Block-diagram of the CLUR-detection-tracking system.

This system exploits advanced spatiotemporal image processing methods for clutter rejection with strong signal compensation, changepoint detection based track initiation and termination algorithms, NLF-based target tracking algorithms, and global data association (optimal for association of all detections but not locally optimal for a particular detection), among others.

3. Spatiotemporal Image Processing Algorithms for Clutter Rejection

The developed baseline CLUR technique is based on a multi-parametric approximation of clutter (regression-type modeling) that leads to an adaptive spatial-temporal filtering (image processing) algorithm. The "coefficients" of the filter are calculated adaptively according to the minimum distance algorithm. The adaptive spatial-temporal filter allows one to suppress any background, regardless of its spatial variation. It not only whitens the data but also corrects all translational and rotational distortions. The results of an experimental study presented in Section 5 show that the proposed algorithm gives a substantial gain compared to the best existing spatial techniques as well as to the industry standard temporal differencing method.

We start with a description of the basic idea and a generic CLUR algorithm for the class of parametric problems that involve parametric approximations of the function $b_n(r)$ – background clutter. This approach was first proposed by Tartakovsky and Blažek [166]. Note that we do not use any assumptions on the statistical properties of clutter. All we assume is that clutter is an arbitrary function of spatial coordinates (may be a quite sharp function) and a slowly varying function of time in a certain time interval m.

The basic iterative CLUR algorithm that includes jitter compensation has the following form:

1. Initialization. This step can be performed in various ways. Typically this step requires about m observations and the result of the initialization stage are the pilot estimates $\hat{\theta}_k(m)$, $(\hat{\delta}_1, \dots, \hat{\delta}_m)$, and background $\hat{b}_m(r_{ij}) = \sum_{k=1}^{M} \hat{\theta}_k(m) f_k(r_{ij} - \hat{\delta}_m)$, where $\{f_k\}$ is the orthogonal basis (Fourier, wavelets, splines), δ is unknown shifts due to vibrations (jitter), and θ are parameters in the space-time splitting decomposition (approximation) of clutter $b_n(r)$. Initialization schemes include autonomous algorithms of estimation of shifts between two frames based on simple spline-approximations.

2. Typical Step $\bar{b}n, n > m$.

(a) ***Jitter Estimation.*** The estimate $\hat{b}_{n-1}(r_{ij})$ obtained from the previous step is compared with the n-th frame (starting with $n = m$), and the ML/MD estimate of jitter $\hat{\delta}_n$ is computed as the solution of the nonlinear optimization problem with

$$\hat{b}_{n-1}(r_{ij} - \delta) = \sum_{k=1}^{M} \hat{\theta}_k(n-1) f_k(r_{ij} - \delta).$$

(b) ***Estimation of Parameters***. Having the estimates $\hat{\delta}_{n-m+1}, \dots, \hat{\delta}_n$, the estimates $\hat{\theta}_k(n)$ are computed for the n-th frame from the least squares minimization problem by comparing the observed frame (image) $Z_n(r_{ij})$ with the model. This recomputing in the corrected coordinate system is equivalent to frame alignment.

(c) ***Clutter estimation***. Using the estimates obtained from previous steps, compute the estimate of $b_n(r_{ij})$ for all i and j in the corrected coordinate system,

$$\hat{b}_n(r_{ij}) = \sum_{k=1}^{M} \hat{\theta}_k(n) f_k(r_{ij} - \hat{\delta}_n). \tag{13.1}$$

(d) ***Clutter Rejection.*** Using the estimate (13.1), compute the residuals (filtered background)

$$\widetilde{Z}_n(r_{ij}, \hat{\delta}_n) = Z_n(r_{ij}) - \sum_{k=1}^{M} \hat{\theta}_k(n) f_k(r_{ij} - \hat{\delta}_n).$$

Our study of various algorithms showed that the following parametric models and corresponding spatial-temporal filters are feasible for implementation in the bank of CLUR filters:

1. Two-dimensional Fourier Series with Double Nyquist Rate —"Fourier."
2. Two-dimensional Wavelet Series —"Wavelet."
3. Local Polynomial Approximation —"Pol."
4. Spline-based Interpolation Methods with Double Resolution — "Spl-DR."
5. Spatial-temporal Autoregression — "STAR."

4. Target Detection and Tracking

In this section we address target detection and tracking methods within the conventional detect-then-track approach.

4.1. In-Frame Detection Algorithms

Each target detection method specified below solves the problem of in-frame target detection and estimation of targets' positions and intensities, i.e., forms the so-called "blips." Each blip is characterized by the estimates of both "total" and "effective" signal intensities as well as the target position. It is assumed that a frame is first whitened by one of the CLUR filtering algorithms. The following algorithms are available for selection in the "in-frame target detection module":

- **"Simplest"** is based upon direct data thresholding. The center of a hot pixel (which intensity is greater then a detection threshold) is regarded as the target position estimate and the corresponding signal amplitude as the estimate of the target intensity.
- **"Optimum"** is based on the precise calculation of the optimum decision-making statistic for each node (a square composed from four pixels) with compensation of the signals from already detected targets. It uses information about the form of the PSF. It allows for resolving two targets within one node.
- **"Sub_opt"** is the version of the "optimum" algorithm where the optimal decision-making statistic is substituted by a sub-optimal one calculated approximately at the reference points located in a grid having a period of 0.5 pixel size (two targets within one node cannot be resolved).
- **"Adaptive"** is a version of the "sub_opt" algorithm with the automatic adjustment of the detection threshold with the use of the background estimates. As a result, a density of falsely detected blips is maintained constant in different fragments of a frame.

4.2. The Multitarget Tracker

The tracker performs the following operations:

- Identification/association of new detections (blips) with existing tracks.
- Initiation of new tracks based on blips that are not identified as belonging to existing tracks.
- Confirmation of newly initialized tracks.
- Deletion of unconfirmed tracks.
- Termination of tracks.

Track initiation and track termination algorithms are subdivided into the two classes: (1) signal-based procedures that operate with likelihood ratios built on models of intensity distributions around local detections, and (2) binary algorithms that are based on blips/detections (i.e., binary quantized data).

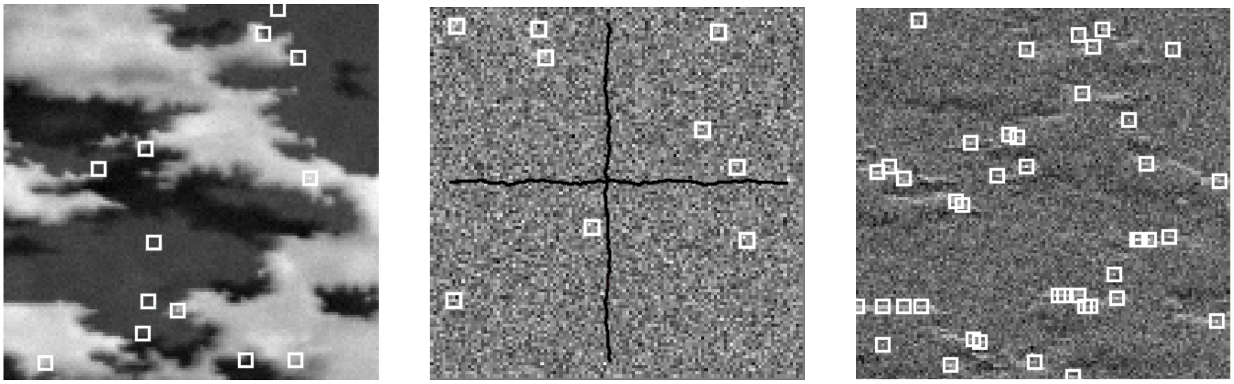
We propose two signal-based track initiation algorithms that are based on the changepoint detection methods that are applied to signals (intensities) in the vicinity of in-frame detections. The first algorithm is the unlimited CUSUM-type algorithm; and the second one is the window-limited CUSUM test (WL-CUSUM). In the following, these algorithms will be referred to as "signal-based."

The same two types of change detection tests are used for binary quantized data (i.e., for a sequence of detections/blips). These algorithms turn out to be more robust compared to the former signal-based, especially when the conjecture on the Gaussian model is not true. At the same time, the loss of efficiency is small (less than 30%) compared to the signal-based approach when the true model is indeed Gaussian.

At the stage of tracking, the track is predicted to the next step and the corresponding gate is calculated. When target disappears, the distribution of the data in gates abruptly changes. Therefore, termination of tracks can be effectively performed based on the quickest detection of the moment of target disappearance similar to track initiation. Note also that a similar approach is used for termination of false tracks.

5. Testing for Geostationary Platforms

We performed a detailed comparative study of the "industry standard" differencing method with our clutter rejection techniques. The differencing clutter rejection method simply subtracts two consecutive frames. It is therefore equivalent to our temporal window-limited clutter rejection filter (implemented in the bank of CLUR filters) with the window size of 1 frame.



(a) Original frame (b) Spatial-temporal *Wavelet* filter (c) *Differencing* CLUR algorithm

Figure 13.2: The results of clutter suppression and target tracking using spatial-temporal Wavelet and differencing CLUR filters.

We used an image sequence with moderately intense clutter and sensor noise STD $\hat{\sigma}_N = 3$. Two weak targets were inserted in the sequence. We first used the *Wavelet* spatial-temporal filter with window of 20 frames. The results were very successful — the standard deviation of the residual clutter plus noise was about 3 and both targets were tracked, as can be seen in Figure 13.2(b). By contrast, the differencing method was not able to track targets, as seen from Figure 13.2(c). In the images, squares with no tracks attached represent instantaneous detections part of which are false, while solid lines correspond to confirmed target tracks.

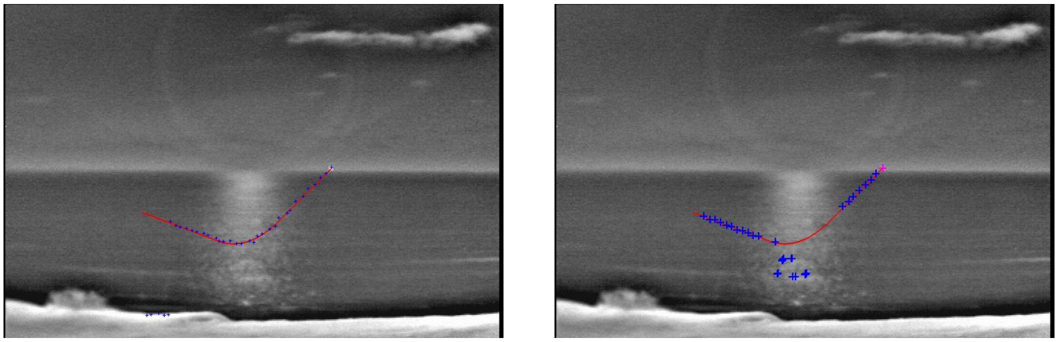
6. Detection and Tracking by Fusing Multiple Frames Through Optimal Nonlinear Filtering

For moving platforms, such as low earth and high elliptic orbits, STSS, shipboard IRST and other sensors, one may expect difficult situations when even after clutter rejection the effective SNR is low, so that the in-frame detection with a given acceptable FAR is not possible. Choosing lower detection thresholds is not an answer, since threshold lowering results in an intense flow of false detections and initiation of multiple false tracks, which in turn leads to tracker failures. To overcome this difficulty we propose track-before-detect (TBD) algorithms. These algorithms process a series of frames simultaneously without detecting targets in each and every frame, i.e., they accumulate information along hypothetical tracks.

The proposed TBD architecture is based on optimal nonlinear filtering (ONF) for switching multiple models. We show that the optimal TBD algorithm can be represented in the form of a bank of interacting nonlinear Bayesian matched filters. The output of this bank is the unnormalized posterior density for the target position given previous data. The TBD algorithm is recursive and does not require pruning of hypotheses. An important feature of the algorithm is that it does not require linear dependence between observations and the state process. This allows one to build a TBD algorithm that handles highly nonlinear δ-like target signatures characteristic for small targets in IR images. This algorithm is a particular case of a more general approach for spatiotemporal nonlinear filtering outlined in Chapter 2, Section 2.

Therefore, in certain important scenarios tracking with the conventional detect-then-track scheme as well as with the bank of 3D matched filters (3DMF) is not possible, in which case the NLF-based TBD methods are needed. We now present a comparative analysis with the industry standard bank of 3D matched filters (3DMF) for a challenging data sets that contain strong sea glint.

As possible target dynamics we chose 16 different directions of motion (throughout a 360 degree angle) and a wide range of velocities (from 1 to 5 pixels per frame). Only transitions to neighboring (direction-wise and speed-wise) states were allowed by the Markov model with probabilities of switch equal to 0.1 for every direction of change. Probabilities of changes in direction and speed were considered separately and the probabilities of complex direction/speed transitions were obtained by multiplying the two separate probabilities. The target signal was simulated by a square of 3×3 pixels with an average per-pixel SNR of approximately 0.2 (-7 dB) for the sea glint area, about 1-2 ($0 - 3$ dB) for the rest of the sea territory and about 0.1 (-10 dB) for some of the artifacts inside the coastal area at the bottom of the scenery. In order to illustrate the power of the developed ONF-based TBD algorithm, no background removal was performed on the images before the tracking, which complicates the problem tremendously. The target enters the sea-glint area at frame 16. First, it moves along a straight-line trajectory. Upon entering the glint area the target starts a slight maneuver, decreasing the vertical component of its velocity by 1 pixel per frame. At frame 26 the maneuver is terminated and the target proceeds to move along a straight-line trajectory.



(a) Target tracking by ONF-TBD

(b) Target tracking by 3DMF

Figure 13.3: The results of target tracking in video.

The screen-shots with the tracking results for ONF-TBD and 3DMF are shown in Figure 13.3. This data set represents a video stream that was obtained with a Raytheon VOx long-wave 320×240 imager pointed at sun-glint over the Pacific Ocean. The solid red line corresponds to the true target trajectory, while the blue "+" line corresponds to the estimated trajectory. Once the ONF-TBD finds the target it tracks it well, even in sea glint and when it performs a maneuver. On the other hand, the 3DMF bank tracks the target well only until the target enters the sea glint area. In sea glint, it immediately loses the target.

Based on the experimental results we can conclude that (1) the popular 3DMF bank has poor performance when detecting dim maneuvering targets, and (2) the proposed ONF-based TBD algorithm is very robust with respect to both target maneuvering and very low SNR.

Chapter 14

Testing and Validation: A Disaster Scenario

1. Creation of a Disaster Scenario Combining all Parts Together

We have created a scenario illustrating how all parts of the MURI project could fit together in a realistic scenario.To this end, we ran an undergraduate REU project based at UCLA called "Disaster LA" in which the students developed the details of the disaster plot. In order to construct a reasonable scenario and also describe a realistic response from the local authorities,the students have been in contact with the Los Angeles Police Department's Real-time Analysis and Critical Response (RACR) division,which is tasked with analyzing incoming crime and emergency data and developing appropriate responses. The officers working in RACR have been instrumental in aiding in development of the scenario's timeline, and in determining how the various technologies may be useful to such a government agency.

The basic element of the disaster scenario involves the terrorist detonation of multiple high explosives in various areas of Los Angeles, and the subsequent aftermath.The attack actually begins, however, with a Denial of Service (DOS) attack on the internet communication capabilities of the LAPD and other emergency responders. The student's then imagine that conventional, chemical (in the form of a noxious gas), and nuclear (a "dirty bomb") explosions have been set off in areas such as Los Angeles International Airport (LAX), downtown LA, and Westwood (where UCLA is located). Following these detonations, the authorities must use the relevant technologies to safely secure the dangerous areas, help people evacuate to safe locations, stop any potential civil unrest, and identify and apprehend the terrorists involved in the plot.

To accomplish these goals, the authorities are first imagined to be using state of the art DOS detection algorithms to block the internet attack and keep their information channels open and working. Though the bombs are successfully detonated by the terrorists, the police are able to respond quickly to the areas, and use swarms of autonomous robots to detect the boundaries of the dangerous areas (those with high radiation or containing poisonous gases) so that civilians can be relocated outside of these danger zones. To deal with possible panic in the general population, the police use real-time crime mapping and prediction algorithms to send their already stressed forces to only those locations automatically designated as most dangerous. In order to identify the suspects involved in the attack, the authorities use video footage captured from a number of networks of autonomous cameras located throughout the city, each running video-tracking algorithms to identify suspicious behavior (an individual leaving behind a brief case in LAX, for example) and follow the individuals involved. These suspects are then fed into a program that keeps track of known terrorists and their associates to identify who among them are likely perpetrators for this type of attack.

Though the scenario envisions technologies that are, in some cases, not yet developed for field work, each of the projects has been simulated on a computer to provide proof-of-concept for the scenario. In doing so, the students have created helpful software packages and, in some cases, improvements in the algorithms or the analysis of data that underlies them. In this way, the students have not only created an interesting presentation that showcases the technologies and their potential, but also aided in the progress needed to achieve the level of capabilities they have imagined.

Chapter 15

Technology Transfer

1. Transition to China Lake Naval Weapons Center

Unmanned air systems, ground- and underwater-vehicles now provide a range of important functions in naval operations. Fielded systems, however, require extensive teams of highly trained operators and support personnel. DOD requirements and science and technology priorities address the need for increased autonomy with the warfighter monitoring and supervising multiple systems to effectively leverage assets while reducing individual workload and numbers of personnel. Co-PI Bertozzi served as a consultant on an ONR funded project developed cooperative control algorithms for unmanned vehicles and suggested that that project use the algorithm developed in [93] funded by this MURI award. The searching algorithm has now been ported to the Multiple Unified Simulation Environment (MUSE) in use at the Naval Air Warfare Center (NAWCWD), China Lake. MUSE integrates ground station control with respective platform six degree-of-freedom simulation and sensor models and employs MetaVR scene generation to support a number of UAVs. This enables high fidelity simulations of larger systems like the RQ/MQ-1 Predadator, and MQ-9 Reaper, as well as smaller systems like the RQ-7 Shadow. Often used in training applications, MUSE was adapted to incorporate new functions and capabilities specific to integrated systems test and evaluation.

Source code for random search based on Levy processes was provided to NAWCWD at China Lake. The researchers at China Lake implemented the code on two different simulation platforms. The first simulation environment was a freeware flight simulator that provided a faster than real time simulation conditions. The second simulation was the MUSE code package that provided realistic simulation for multiple UAV platforms. The capability of a group of random searchers to find a stationary and moving target using the algorithm provided was simulated in both the flight simulator and the MUSE. It can be shown that optimal search performance for stationary targets with no a-priori information is a deterministic search, and it was shown that the random search achieved near optimal search performance. Moreover, the random search found moving targets in a shorter time span than the deterministic code. These results have lead to continued work at NAWCWD-China Lake in the area of autonomous search.

Points of contact at China Lake: Arjuna Flenner and Katia Estabridis.

2. Transition to the Los Angeles Police Department (LAPD)

2.1. Geographic Profiling

A prototype tool for geographic profiling of serial criminal offenders was transferred to LAPD in September 2009. This tool is based on the algorithm for Geographic Profiling developed by the 2009 LAPD-RIPS team and has a user interface based around Google Earth. Geographic profiles score the likelihood that a serial offender has a key anchor point in a given location. This tool is innovating in that it incorporates information about local demographic and spatial features of the environment into the profile. Subsequent interactions with LAPD indicate that the tool is used with the Real-time Analysis and Critical Response Division (RACR) at LAPD. A similar prototype tool for infilling information on suspected gangs involved in gang shootings was provided to LAPD by the 2010 LAPD-RIPS team. This tool examines a given crime and, based on rivalry histories, location and timing of the crime, probabilistically score which of many alternative gangs is a likely suspect gang.

2.2. Crime Mapping

The paper [155] was the impetus for a summer REU project at IPAM as part of their RIPS program. Lead author Laura Smith (finishing PhD student) was the mentor for this project. The students wrote a suite of crime mapping software for LAPD based on their current crime briefing strategies and data collects. The students implemented methods from the above paper along with more standard kernel estimation methods. For the specific police data needs, as discussed below, the kernel methods worked as well as the new method, however there are many crime mapping applications where the standard method is much less effective and these are illustrated in the manuscript. The RIPS code development was funded by an NSF grant to IPAM. The project was specifically chosen based on the work in the above paper and the plan was to implement these and other algorithms for a real police tool application. Code has been given to LAPD for use in an upcoming predictive policing experiment.

UCLA coPI Brantingham is directly involved in this planned predictive policing experiment. The intention is to have a control group and a target area so that we can test the effectiveness of the crime algorithms for crime reduction against current LAPD practices. The plan would be to work the tool into LAPD's daily war room process. Laura Smith and the team built the interface to mirror our current process of looking at a 1, 3 and 7 day retrospective map at the daily war room crime briefing. When LAPD launches the test, they will use the 1, 3 and 7 day forecast density maps to plan deployment of both regular patrol resources and of specialized units or discretionary resources. They hope to have results by the end of the year.

Point of contact at LAPD: Captain Sean Malinowski, sean.malinowski@lapd.lacity.org.

3. Transition to DOE

The developed changepoint detection methods and algorithms, particularly decentralized methods, can be effectively used for the design of working prototypes of detection and tracking systems for detecting and tracking terrorist activities and for detection and isolation of computer network intrusions. The advanced changepoint detection algorithms developed at the University of Southern California (Prof. Tartakovsky) for anomaly-based Intrusion Detection System have been transferred to the Oak Ridge National Laboratory for tuning and testing over UltraScience Net (USN) network research testbed for operation at 1-10 Gbps rates (PI Dr. Rao of the DOE Oak Ridge National Lab). Preliminary testing shows that this statistical anomaly-based IDS provides reliable and rapid detection of certain classes of attacks with very low false alarm rate; these attacks are not easily amenable to detection by existing signature-based methods.

4. Transition to MDA and Air Force

The advanced target detection and tracking system (developed by Prof. Tartakovsky) that includes a bank of clutter rejection filters based on spatial-temporal image processing algorithms has been transferred to MDA and the Hanscom Air Force Base for evaluation and testing (POC James Brown). The developed algorithms guarantee very high detection/tracking performance for small ballistic targets (at boost and mid coarse stages) allowing almost complete suppression of solar cloud clutter in space-based IR sensors (SBIRS HIGH). Such performance cannot be achieved with existing industry standard methods.

5. Transition to Commercial Companies

Prof. Papadopoulos continues discussions and collaboration with ESoft. ESoft is very interested in our approach to characterize the IP addresses of spammers, and wants us to investigate approaches where they can classify entire blocks of addresses as suspicious. ESoft produces appliances that sit in a customer's network, which provide information and receive constant updates from the central office about new vulnerabilities and also send back reports with alerts. They are looking at new algorithms to go into the appliance, and part of them may be our approach to characterize addresses sending spam.

Prof. Medioni has been in discussion with a company named InnoVision Optics, Co about transferring our TNT techniques in a multi-camera system. InnoVision Optics provides optical capturing and camera support for many industries, such as movie production, etc. Their current sophisticated multi-camera capturing system involves intensive human interaction to track a tagged object of interest, such as the basketball in a game. They plan to use an automatic (or minimal human interaction) tracking solution to remove human operators from the control loop and achieve more efficient multi-camera collaboration.

Video tracking algorithms developed at USC (Prof. Medioni) have been transferred to Torrey Pines Logic, Inc., San Diego, CA for embedding into a shipboard IRST testbed and testing against real IR data sets (collected by the Navy) that contain multiple extended targets barely visible in sea-glint.

Bibliography

[1] Clif 2006 dataset. https://www.sdms.afrl.af.mil/datasets/clif2006/.

[2] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[3] N. U. Ahmeda, M. Fuhrmanb, and J. Zabczykc. On filtering equations in infinite dimensions. *J. Functional Analysis*, 143(1):180–204, January 1997. doi: doi:10.1006/jfan.1996.2970.

[4] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, June 2008. ISSN 1532-4435.

[5] A. Allahverdyan. Entropy of hidden markov processes via cycle expansion. *Journal of Statistical Physics*, 133:535–564, 2008. ISSN 0022-4715. 10.1007/s10955-008-9613-0.

[6] M. Allenby, K. Louie, and M. Masaki. A point process model for simulating gang-on-gang violence. Technical report, UCLA, 2010. Summer REU program, Tim Lucas mentor.

[7] G. Atia and V. Veeravalli. Sensor scheduling for energy-efficient tracking in cluttered environments. In *Information Theory and Applications Workshop (ITA), 2011*, pages 1 –9, Feb. 2011.

[8] G. Atia, V. Veeravalli, and J. Fuemmeler. Sensor scheduling for energy-efficient target tracking in sensor networks. *IEEE Transactions on Signal Processing*, 59(10):4923 –4937, Oct. 2011.

[9] K. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*, volume 39 of *Texts in Applied Mathematics*. Springer, third edition, 2009. ISBN 978-1-4419-0457-7. doi: 10.1007/978-1-4419-0458-4.

[10] S. Avidan. Ensemble tracking. In *CVPR*, 2005.

[11] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[12] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Proc. Conf. on Information Sciences and Systems*, 1980.

[13] Y. Bar-Shalom, T. Kirubarajan, and X. Lin. Probabilistic data association techniques for target tracking with applications to sonar, radar and eo sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 20(8):37 –56, Aug. 2005.

[14] G. Bartlett, J. Heidemann, and C. Papadopoulos. Using low-rate flow periodicities for anomaly detection: Extended. Technical Report ISI-TR-2009-661, USC/Information Sciences Institute, August 2009.

[15] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[16] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, Englewood Cliffs, 1993. ISBN 0-13-126780-9.

[17] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):pp. 1554–1563, 1966. ISSN 00034851.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *CVIU*, volume 110, pages 346–359, 2008.

[19] M. Beal and Z. Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7:453–464, 2003.

[20] V. E. Benes. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5 (1–2):65–92, January 1981. doi: 10.1080/17442508108833174.

[21] D. Bertsekas. *Dynamic Programming and Optimal Control, vol. I and II.* Athena Scientific, Belmont, Massachusetts, 1995.

[22] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice Hall, NJ, 1987.

[23] A. Bjoerck and G. H. Golub. Numerical methods for computing angles between linear subspaces. In *Mathematics of computation*, pages 579–594, 1973.

[24] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *IJCV*, volume 26, pages 63–84, 1998.

[25] A. Blake and M. Isard. *Active Countours*. Springer-Verlag, New York, NY, 1998.

[26] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[27] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. In *JMLR*, pages 1579–1619, 2005.

[28] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *ICCV*, pages 1–8, 2007.

[29] L. Breiman. Random forests. In *ML*, volume 45, pages 5–32, 2001.

[30] A. Chen, T. Wittman, A. Tartakovsky, and A. Bertozzi. Boundary tracking through efficient sampling. *Applied Mathematics Research eXpress*, 2011(2):182–214, 2011.

[31] H. T. Cheng. *Algorithms for partially observable Markov decision processes*. PhD thesis, University of British Columbia, 1988.

[32] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR*, pages 234–240, 2003.

[33] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. In *PAMI*, volume 27, pages 1631–1643, 2005.

[34] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18, March 2001.

[35] I. B. Count, 2008. URL `http://www.iraqbodycount.net/`.

[36] I. Cox and S. Hingorani. An efficient implementation of reids multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In *ICPR*, pages 437–443, 1994.

[37] V. Crespi, G. Cybenko, and G. Jiang. The theory of trackability with applications to sensor networks. *ACM Trans. Sen. Netw.*, 4:16:1–16:42, June 2008. ISSN 1550-4859. doi: http://doi.acm.org/10.1145/1362542.1362547.

[38] D. Crisan and T. Lyons. Nonlinear filtering and measure-valued processes. *Probab. Theory Relat. Fields*, 109(2):217–244, 1997. doi: 10.1007/s004400050131.

[39] D. Crisan and B. Rozovskii. *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, April 2011.

[40] J. Cvitanić', R. Liptser, and B. Rozovskii. A filtering approach to tracking volatility from prices observed at random times. *Ann. Appl. Probab.*, 16(3):1633–1652, 2006. doi: 10.1214/105051606000000222.

[41] J. Cvitanić', B. Rozovskii, and I. Zaliapin. Numerical estimation of volatility values from discretely observed diffusion data. *J. Computational Finance*, 9(4), 2006.

[42] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[43] H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(9):805–822, April 1999.

[44] S. DiBenedetto, K. Gadkari, N. Diel, A. Steiner, D. Massey, and C. Papadopoulos. Fingerprinting custom botnet protocol stacks. In *Proceedings of the 6th IEEE Workshop on Secure Network Protocols (NPSec)*, pages 61–66, Kyoto, Japan, October 2010. doi: 10.1109/NPSEC.2010.5634448.

[45] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, pages 1271–1278, 2009.

[46] A. Doucet, N. de Freitas, N. Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice*. Springer, June, 2001.

[47] V. Dragalin. Adaptive procedures for detecting a change in distribution. In *4th Umea-Würzburg Conf. Statist.*, pages 87–103, 1996.

[48] V. P. Dragalin. Optimality of a generalized CUSUM procedure in quickest detection problem. In *Proceedings of the Steklov Institute of Mathematics: Statistics and Control of Random Processes*, volume 202, pages 107–120, Providence, RI, 1994. American Mathematical Society.

[49] V. P. Dragalin, A. G. Tartakovsky, and V. V. Veeravalli. Multihypothesis sequential probability ratio tests, part 2: Accurate asymptotic expansions for the expected sample size. *IEEE Transactions on Information Theory*, 46(4):1366–1383, 2000.

[50] G. Duan, C. Huang, H. Ai, and S. Lao. Boosting associated pairing comparison features for pedestrian detection. In *The Ninth IEEE International Workshop on Visual Surveillance*, pages 1097–1104, 2009.

[51] T. Duncan. Tech. report 7001–4. Technical report, Stanford University, Center for Systems Research, May 1967.

[52] M. Egesdal, C. Fathauer, K. Louie, and J. Neuman. Statistical and stochastic modeling of gang rivalries in los angeles, March 2010.

[53] A. Elgammal. Learning to track: Conceptual manifold map for closed-form tracking. In *CVPR*, 2005.

[54] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov Models: Estimation and Control*. Springer, New York, NY, October 1994.

[55] Y. Ephraim and N. Merhav. Hidden markov processes. *Information Theory, IEEE Transactions on*, 48(6):1518 –1569, jun 2002. ISSN 0018-9448. doi: 10.1109/TIT.2002.1003838.

[56] B. Froba and A. Ernst. Face detection with the modified census transform. In *AFGR*, pages 91–96, 2004.

[57] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. of the 26th International Conference on Machine Learning*, 2009.

[58] J. Fuemmeler and V. Veeravalli. Smart sleeping policies for energy efficient tracking in sensor networks. *IEEE Transactions on Signal Processing*, 56(5):2091–2101, May 2008.

[59] J. Fuemmeler and V. Veeravalli. Energy efficient multi-object tracking in sensor networks. *Signal Processing, IEEE Transactions on*, 58(7):3742 –3750, July 2010.

[60] J. Fuemmeler, G. Atia, and V. Veeravalli. Sleep control for tracking in sensor networks. *IEEE Transactions on Signal Processing*, 59(9):4354 –4366, Sept. 2011.

[61] M. Gonzalez, X. Huang, B. Irvine, D. S. H. Martinez, C. H. Hsieh, Y. R. Huang, M. B. Short, and A. L. Bertozzi. A third generation micro-vehicle testbed for cooperative control and sensing strategies. In *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 14–20, 2011.

[62] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, pages 47–56, 2006.

[63] H. Grabner, J. Matas, L. V. Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR*, pages 1285–1292, 2010.

[64] H. Grabner, C. Leistner, and H. Bishof. Semi-supervised on-line boosting for robust tracking. In *2008*, pages 234–247, ECCV.

[65] M. Grabner, H. Grabner, and H. Bischof. Learning features for tracking. In *CVPR*, 2007.

[66] T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, Berlin, 1963.

[67] R. A. Hegemann, L. M. Smith, A. Barbaro, A. L. Bertozzi, S. Reid, and G. E. Tita. Geographical influences of an emerging network of gang rivalries. *Physica A*, 390(21–22):3894–3914, October 2011.

[68] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[69] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983.

[70] C. Huang and R. Nevatia. High performance object detection by collaborative learning of joint ranking of granules features. In *CVPR*, pages 41–48, 2010.

[71] J. Irvine, M. Gonzalez, E. Huang, , and M. Short. Integration of third generation micro-cars into the ucla applied mathematics laboratory testbed. Technical report, UCLA, 2010. summer REU 2010 robotics team.

[72] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, NY, March 1970.

[73] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, May 1998.

[74] Z. Kalal, J. Matas, and K. Mikolajczyk. Online learning of robust object detectors during unstable tracking. In *OLCV*, pages 1417–1424, 2009.

[75] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56, 2010.

[76] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *CVPR*, pages 267–272, 2003.

[77] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using invariant appearance model. In *ICPR*, pages 759–762, 2004.

[78] S. Kent. On the trial of intrusions into information systems. *IEEE Spectrum*, 37(12):52–56, December 2000. doi: 10.1109/6.887597.

[79] H. J. Kushner. On the dynamical equations of conditional probability density functions, with applications to optimal stochastic control theory. *J. Mathematical Analysis and Applications*, 8(2):332–344, April 1964.

[80] H. J. Kushner. Approximations to optimal nonlinear filters. *IEEE Transactions on Automatic Control*, 12(5):546–556, October 1967. doi: 10.1109/TAC.1967.1098671.

[81] J. S. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[82] T. L. Lai. Information bounds and quick detection of parameter changes in stochastic systems. *Information Theory, IEEE Transactions on*, 44(7):2917 –2929, Nov. 1998.

[83] T. L. Lai. Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44:2917–2929, November 1998. doi: 10.1109/18.737522.

[84] K.-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *CVPR*, 2005.

[85] K. Leung, C. Hsieh, Y. Huang, A. Joshi, V. Voroninski, and A. L. Bertozzi. A second generation micro-vehicle testbed for cooperative control and sensing strategies. In *American Control Conference, ACC'07*, pages 1900–1907, 2007.

[86] E. Lewis, G. Mohler, P. J. Brantingham, and A. L. Bertozzi. Self-exciting point process models of civilian deaths in iraq. *Security Journal*, 2011. doi: 10.1057/sj.2011.21.

[87] A. Li, F. Tang, Y. W. Guo, and H. Tao. Discriminative nonorthogonal binary subspace tracking. In *ECCV*, pages 258–271, 2010.

[88] L. J. Li, R. Socher, and L. F. Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, pages 2036–2043, 2009.

[89] M. Li, J. T. Kwok, and B. L. Lu. Online multiple instance learning with no regret. In *CVPR*, 2010.

[90] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *CVPR*, pages 1–8, 2007.

[91] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In *NIPS*, pages 793–800, 2004.

[92] J. Liu, M. Chu, and J. Reich. Multitarget tracking in distributed sensor networks. *IEEE Signal Processing Magazine*, 24(3):36 –46, May 2007.

[93] W. Liu, M. B. Short, Y. E. Taima, and A. L. Bertozzi. Multiscale collaborative searching through swarming. In *Proceedings of the 7th International Conference on Informatics in Control, Automation, and Robotics (ICINCO)*, Portugal, June 2010.

[94] X. Liu and T. Yu. Gradient feature selection for online boosting. In *ICCV*, 2007.

[95] G. Lorden. Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6):1897–1908, 1971.

[96] G. Lorden and M. Pollak. Nonanticipating estimation applied to sequential analysis and changepoint detection. *Ann. Statist.*, 33(3):1422–1454, 2005.

[97] S. V. Lototsky and B. L. Rozovskii. Stochastic parabolic equations of full second order. *Topics in stochastic analysis and nonparametric estimation*, pages 199–210, 2007.

[98] S. V. Lototsky and B. L. Rozovskii. Stochastic partial differential equations driven by purely spatial noise. *SIAM J. Math. Anal.*, 41(4):1295–1322, 2009. doi: 10.1137/070698440.

[99] S. V. Lototsky and B. L. Rozovskii. A unified approach to stochastic evolution equations using the skorokhod integral. *Theory Probab. Appl.*, 54(2):189–202, 2010. doi: 10.1137/S0040585X97984152.

[100] S. V. Lototsky, B. L. Rozovskii, and X. Wan. Elliptic equations of higher stochastic order. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44:1135–1153, September 2010. doi: 10.1051/m2an/2010055.

[101] L. S. Maisel. Rating united states senators the strength of maines delegation since 1955. *The New England Journal of Political Science*, 4(1), 2010.

[102] D. H. Martinez. Integration of 3rd generation vehicles to the applied mathematics laboratory autonomous micro-vehicle testbed. Master's thesis, UCLA, Electrical Engineering, 2010.

[103] N. McCarty, K. T. Poole, and H. Rosenthal. *Polarized America: The Dance of Ideology and Unequal Riches*. MIT Press, 2006.

[104] Y. Mei. Information bounds and quickest change detection in decentralized decision systems. *IEEE Trans. Inform. Theory*, 51(7):2669–2681, 2005.

[105] N. Merhav and Y. Ephraim. Maximum likelihood hidden markov modeling using a dominant sequence of states. *Signal Processing, IEEE Transactions on*, 39(9):2111 –2115, sep 1991. ISSN 1053-587X.

[106] M. Mezard and G. Parisi. The bethe lattice spin glass revisited. *The European Physical Journal B*, 20:217, 2001.

[107] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. In *PAMI*, volume 19, pages 696–710, 1997.

[108] G. O. Mohler, M. B. Short, P. Brantingham, F. Schoenberg, and G. Tita. Self-exciting point process modeling of crime. *J. Am. Stat. Assoc.*, 106, 2011.

[109] P. D. Moral. Nonlinear filtering: interacting particles solution. *Markov process and related fields*, 2 (4):555–580, 1996.

[110] P. D. Moral. *Feynman-Kac Formulae*. Springer, March 2004.

[111] D. Morgan and I. Schwartz. Dynamic coordinated control laws in multiple agent models. *Physics Letters A*, 340(1–4):121–131, 2005.

[112] G. Moustakides. Decentralized CUSUM change detection. In *Proc. 9th Intern. Conf. on Inform. Fusion*, Florence, Italy, 2006.

[113] G. V. Moustakides, A. S. Polunchenko, and A. G. Tartakovsky. A numerical approach to performance analysis of quickest change-point detection procedures. *Statistica Sinica*, 21(2):571–596, April 2011.

[114] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov network. In *CVPR*, pages 975–982, 2009.

[115] H. T. Nguyen and A. W. Smeulders. Robust tracking using foreground-background texture discrimination. In *IJCV*, 2006.

[116] N. C. Oza and S. Russell. Online bagging and boosting. In *Artificial Intelligence and Statistics*, pages 105–112, 2001.

[117] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, pages 1–8, 2007.

[118] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. In *PAMI*, volume 32, pages 448–461, 2010.

[119] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1):100–115, 1954.

[120] J. Park, B. Rozovsky, and R. Sowers. Efficient nonlinear filtering of a singularly perturbed stochastic hybrid system. *London Math. Society J. of Computation and Mathematics*, 2010. submitted.

[121] A. Perera, C. Srinivas, A. Hoogs, G. Brooksb, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, volume 1, pages 666–673, 2006.

[122] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032, 2003.

[123] M. Pollak. Optimal detection of a change in distribution. *The Annals of Statistics*, 13(1):206–227, 1985.

[124] M. Pollak. Average run lengths of an optimal method of detecting a change in distribution. *The Annals of Statistics*, 15(2):749–779, 1987.

[125] M. Pollak and A. G. Tartakovsky. Optimality properties of the Shiryaev-Roberts procedure. *Statistica Sinica*, 19:1729–1739, 2009.

[126] M. Pollak and A. G. Tartakovsky. Asymptotic exponentiality of the distribution of first exit times for a class of Markov processes with applications to quickest change detection. *Theory of Probability and Its Applications*, 53(3):430–442, 2009.

[127] M. Pollak and A. G. Tartakovsky. On the first exit time of a nonnegative Markov process started at a quasistationary distribution. *Journal of Applied Probability*, 46(2):589–595, 2011.

[128] A. S. Polunchenko and A. G. Tartakovsky. On optimality of the Shiryaev-Roberts procedure for detecting a change in distribution. *The Annals of Statistics*, 36(6):3445–3457, 2010. doi: 10.1214/09-AOS775.

[129] A. S. Polunchenko and A. G. Tartakovsky. State-of-the-art in sequential change-point detection. *Methodology and Computing in Applied Probability*, 2011. In press.

[130] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2008.

[131] K. Premkumar and A. Kumar. Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1400 –1408, Apr. 2008.

[132] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 –286, feb 1989. ISSN 0018-9219. doi: 10.1109/5.18626.

[133] V. Raghavan and V. Veeravalli. Quickest change detection of a markov process across a sensor array. *Information Theory, IEEE Transactions on*, 56(4):1961 –1981, Apr. 2010.

[134] J. Reichardt and M. Leone. (un)detectable cluster structure in sparse networks. *Phys. Rev. Lett.*, 101: 078701, Aug 2008. doi: 10.1103/PhysRevLett.101.078701.

[135] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, Sept. 1951.

[136] S. Roberts. A comparison of some control chart procedures. *Technometrics*, 8(3):411–430, August 1966.

[137] N. Rodriguez and A. Bertozzi. Local existence and uniqueness of solutions to a pde model for criminal behavior. In *M3AS, special issue on Mathematics and Complexity in Human and Life Sciences*, 2010.

[138] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where and why? semantic relatedness for knowledge transfer. In *CVPR*, pages 910–917, 2010.

[139] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. In *IJCV*, volume 77, pages 125–141, 2008.

[140] M. Rousset. On the control of an interacting particle estimation of schrdinger ground states. *SIAM J. Math. Anal.*, 38(3):824–844, 2006.

[141] B. L. Rozovskii. *Stochastic Evolution Systems: Linear Theory and Applications to Non-Linear Filtering*. Springer, October 1990.

[142] J.-A. Sanchez, J.-M. Benedi, and F. Casacuberta. Comparison between the inside-outside algorithm and the viterbi algorithm for stochastic context-free grammars. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, volume 1121 of *Lecture Notes in Computer Science*, pages 50–59. Springer Berlin / Heidelberg, 1996. ISBN 978-3-540-61577-4.

[143] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3d reconstruction from perspective views. In *CVPR*, pages 483–489, 2004.

[144] Y. Sheng, G. Cybenko, V. Crespi, and G. Jiang. Trackability analysis of multiple processes using multi-distributed agents. In *Integration of Knowledge Intensive Multi-Agent Systems, 2005. International Conference on*, pages 111 – 116, 18-21, 2005. doi: 10.1109/KIMAS.2005.1427063.

[145] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994.

[146] A. N. Shiryaev. The problem of the most rapid detection of a disturbance in a stationary process. *Soviet Math. Dokl.*, 2:795–799, 1961.

[147] A. N. Shiryaev. On optimal methods in quickest detection problems. *Theory Probab. Appl.*, 8:22–46, 1963.

[148] A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Probability and Its Applications*, 8(1):22–46, January 1963. doi: 10.1137/1108002.

[149] A. N. Shiryaev. *Optimal Stopping Rules*. Springer-Verlag, NY, 1978.

[150] M. B. Short, M. R. D'Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. Chayes. A statistical model of criminal behavior. In *M3AS: Mathematical Models and Methods in Applied Sciences, special issue on Traffic, Crowds, and Swarms*, volume 18, pages 1249–1267, 2008.

[151] M. B. Short, P. Brantingham, A. L. Bertozzi, and G. Tita. Dissipation and displacement of hotspots in reaction-diffusion models of crime. In *PNAS 107*, March 2010.

[152] M. B. Short, P. Brantingham, and M. D'Orsogna. Cooperation and punishment in an adversarial game: How defectors pave the way to a peaceful society. *Phys. Rev. E*, 82, 2010.

[153] D. Siegmund. *Sequential analysis: tests and confidence intervals*. Springer series in statistics. Springer-Verlag, 1985.

[154] D. Siegmund. *Sequential Analysis: Tests and Confidence Intervals*. Springer Series in Statistics. Springer-Verlag, New York, 1985. ISBN 0387961348.

[155] L. M. Smith, M. S. Keegan, T. Wittman, G. O. Mohler, and A. L. Bertozzi. Improving density estimation by incorporating spatial information. *EURASIP J. on Advances in Signal Processing*, 2010, 2010. doi: 10.1155/2010/265631. Special issue on Advanced Image Processing for Defense and Security Applications.

[156] E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University, 1971.

[157] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 24:195–220, 2005.

[158] V. I. Spitkovsky, H. Alshawi, D. Jurafsky, and C. D. Manning. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 9–17, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-83-1.

[159] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 2011.

[160] R. Stratonovich. A new representation for stochastic integrals and equations. *SIAM J. Control*, 4(2): 362–371, 1966. doi: 10.1137/0304028.

[161] C. Tang, G. Medioni, and M. Lee. N-dimensional tensor voting, application to epipolar geometry estimation. In *PAMI*, pages 829–844, 2001.

[162] A. G. Tartakovsky. *Sequential Methods in the Theory of Information Systems*. Radio & Communications, Moscow, Russia, 1991.

[163] A. G. Tartakovsky. Extended asymptotic optimality of certain change-point detection procedures. In *Preprint, University of Southern California*, 1998.

[164] A. G. Tartakovsky. Asymptotic optimality of certain changepoint detection procedures: non-iid case. In *Proceedings of the 5th World Congress of the Bernoulli Society for Mathematical Statistics and Probability and 63rd Annual IMS Meeting*, Guanajuato, Mexico, May 2000.

[165] A. G. Tartakovsky. Asymptotic performance of a multichart CUSUM test under false alarm probability constraint. In *Proceedings of the 2005 IEEE Conference on Decision and Control*, volume 44, pages 320–325, 2005.

[166] A. G. Tartakovsky and R. Blažek. Effective adaptive spatial-temporal technique for clutter rejection in IRST. In O. E. Drummond, editor, *SPIE Proceedings: Signal and Data Processing of Small Targets*, volume 4048, Orlando, FL, 2000.

[167] A. G. Tartakovsky and G. V. Moustakides. State-of-the-art in bayesian changepoint detection. *Sequential Analysis*, 29(2):125–145, 2010.

[168] A. G. Tartakovsky and G. V. Moustakides. State-of-the-art in Bayesian changepoint detection. *Sequential Analysis*, 29(2):125–145, April 2010. doi: 10.1080/07474941003740997.

[169] A. G. Tartakovsky and A. S. Polunchenko. Minimax optimality of the Shiryaev-Roberts procedure. In *Proceedings of the 5th International Workshop on Applied Probability*, Universidad Carlos III of Madrid, Spain, July 2010.

[170] A. G. Tartakovsky and V. V. Veeravalli. An efficient sequential procedure for detecting changes in multi-channel and distributed systems. *Proc. 5th IEEE Intern. Conf. on Inform. Fusion, Annapolis, MD*, 1:41–48, July 2002.

[171] A. G. Tartakovsky and V. V. Veeravalli. General asymptotic Bayesian theory of quickest change detection. *Theory of Probability and its Applications*, 49(3):458–497, 2004.

[172] A. G. Tartakovsky and V. V. Veeravalli. General asymptotic Bayesian theory of quickest change detection. *Theory of Probability and Its Applications*, 49(3):458–497, 2005. doi: 10.1137/S0040585X97981202.

[173] A. G. Tartakovsky and V. V. Veeravalli. Asymptotically optimum quickest change detection in distributed sensor systems. *Sequential Analysis*, 27(4):441–475, 2008. doi: 10.1080/07474940802446236.

[174] A. G. Tartakovsky and V. V. Veeravalli. Asymptotically optimal quickest change detection in distributed sensor systems. *Sequential Analysis*, 27(4):441–475, 2008.

[175] A. G. Tartakovsky, M. Pollak, and A. S. Polunchenko. Third-order asymptotic optimality of the generalized Shiryaev-Roberts changepoint detection procedures. *Theoriya Veroyatnostej i ee Promeneniya*, 56(3), 2011.

[176] J. Tsitsiklis. Extremal properties of likelihood-ratio quantizers. *IEEE Trans. Commun.*, 41(4):550–558, 1993.

[177] Z. Tu and S. Zhu. Image segmentation by data driven markov chain. In *PAMI*, volume 24(5), pages 657–674, 2002.

[178] J. H. Urbashi Mitra, Antonio Ortega and C. Papadopoulos. Detecting and identifying malware: A new signal processing goal. *IEEE Signal Processing Magazine*, 23(5):107–111, September 2006.

[179] V. N. Vapnik. Statistical learning theory. In *John Wiley Sons*, 1998.

[180] A. Vaswani, A. Pauls, and D. Chiang. Efficient optimization of an mdl-inspired objective function for unsupervised part-of-speech tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 209–214, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[181] V. Veeravalli. Decentralized quickest change detection. *Information Theory, IEEE Transactions on*, 47(4):1657 –1665, May 2001.

[182] X. Wan, B. Rozovskii, and G. E. Karniadakis. A new stochastic modeling methodology based on weighted wiener chaos and malliavin calculus. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 106, pages 14189–14194. National Academy of Sciences, August 2009. doi: 10.1073/pnas.0902348106.

[183] C. Wilcox, C. Papadopoulos, and J. Heidemann. Correlating spam activity with ip address characteristics. pages 1–6, March 2010. doi: 10.1109/INFCOMW.2010.5466660.

[184] E. Wong. Explicit solutions to a class of nonlinear filtering problems. *Stochastics*, 5(4):311–321, 1981. doi: 10.1080/17442508108833185.

[185] M. Woodroofe. *Nonlinear Renewal Theory in Sequential Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1982. ISBN 0-89871-180-0.

[186] M. Woodroofe. *Nonlinear renewal theory in sequential analysis*. Society for Industrial and Applied Mathematics, 1982.

[187] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. In *IJCV*, volume 75, pages 247–266, 2007.

[188] J. Xiao, H. Cheng, F. Han, and H. Sawhney. Geo-spatial aerial video processing for scene understanding. In *CVPR*, 2008.

[189] E. Xing, M. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

[190] H. Yalcin, R. Collins, and M. Hebert. Background estimation under rapid gain change in thermal imagery. In *Object Tracking and Classification in and Beyond the Visible Spectrum*, 2005.

[191] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. In *PAMI*, volume 31, pages 1195–1209, 2009.

[192] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal monte carlo markov chain data association. In *CVPR*, 2007.

[193] Q. Yu, T. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers and discriminative trackers. In *ECCV*, 2008.

[194] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *CVPR*, pages 834–841, 2004.

[195] C. Yuan, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in presence of strong parallax from a moving camera by multi-view geometric constraints. In *PAMI*, volume 29, pages 1627–1641, 2007.

[196] R. Zabih and J. Woodfill. A non-parametric approach to visual correspondence. In *PAMI*, 1996.

[197] M. Zakai. On the optimal filtering of diffusion processes. *Z. Wahrscheinlichkeitstheorie verw. Geb*, 11(3):230–243, 1969.