Target Recognition in Cluttered Infrared Scenes via Pattern Theoretic Representations and Jump-Diffusion Processes

Prof. Aaron D. Lanterman

School of Electrical & Computer Engineering
Georgia Institute of Technology
404-385-2548
lanterma@ece.gatech.edu
**Target Recognition in Cluttered Infrared Scenes via Pattern Theoretic Representations and Jump-Diffusion Processes**

**Georgia Institute of Technology, School of Electrical & Computer Engineering, 777 Atlantic Drive NW, Atlanta, GA, 30332-0250**

**Presented at the AFOSR Program Review, Princeton University, June 2003. U.S. Government or Federal Rights License**
Variability in Complex Scenes

– Geometric variability
  • Position
  • Orientation
  • Articulation
  • Fingerprint (in the Bob Hummel sense)

– Environmental variability
  • Thermodynamic variability in infrared
  • Illumination variability in visual

– Complexity variability
  • Number of objects not known
Pattern Theory: The Grenander Program

• Representation:
  – Incorporate variability in the parameter space
    • Possibly many nuisance parameters
  – Model mapping from parameters to data

• Inference:
  – Build inference algorithms using the representation
  – Ex.: Markov chain Monte Carlo (jump-diffusion)

• General notion: avoid “preprocessing” or “feature extraction” as much as possible to avoid loss of information
  – Recall Biao Chen’s mention of the Information Processing Inequality

• Apply tools of Bayesian inference to weird things
Legacy Work

• Sponsored by
  – U.S. Army Center for Imaging Science
    (ARO - David Skatrud/Bill Sander)
  – ONR (Bill Miceli)

• Collaborators
  – Michael Miller (now at Johns Hopkins)
  – Donald Snyder (Washington Univ.)
  – Anuj Srivastava (now with Dept. of Stat., Florida State)
    • Airborne targets – radar
  – Matt Cooper (now with Xerox)
    • Thermodynamic variability of targets
Parameter Space for ATR

- Parameter space for a single target:
  \[ \mathcal{X}(1) = \mathbb{R}^2 \times [0, 2\pi) \times A \]
  \[ A = \{M2, M60, T62\ldots\} \]

- Parameter space for an \( n \)-target scene:
  \[ \mathcal{X}(n) = [\mathbb{R}^2 \times [0, 2\pi) \times A]^n \]

- Number of targets not known in advance:
  \[ \mathcal{X} = \bigcup_{n=0}^{\infty} [\mathbb{R}^2 \times [0, 2\pi) \times A]^n \]
Ingrid’s Third Approach

• Data $y$, parameters $x$
• Likelihood $p(y \mid x)$
  – Render infrared scene onto detector plane
  – Model sensor noise effects
• Prior $p(x)$
• Bayesian posterior

$$\pi(x) \equiv p(x \mid y) \propto p(y \mid x)p(x)$$
  – Analytically forboding!
• Sample via jump-diffusion processes
  – Jump from subspace to subspace
  – Diffuse to refine estimates within a subspace
Take Home Message

Go read Ulf Grenander’s papers and books. They are very cool.

Aaron D. Lanterman  404-385-2548  lanterma@ece.gatech.edu
Perspective Projection
Sensor Effects

Optical PSF

Poisson Photocounting Noise

Dead and Saturated Pixels
FLIR Loglikelihood

- CCD loglikelihood of Snyder, Hammoud, and White

\[ L_{CCD}(y | \lambda) = -\sum_i \mu(i) + \sum_i y(i) \ln \mu(i) \]

where \( \mu(j) = \sum_j \text{psf}(i | j) \lambda(j) \)

- Cascade with \( \text{render} : x \rightarrow \lambda \)

\[ L(y | x) = L_{CCD}(y | \text{render}(x)) \]

- Sensor fusion natural; just add loglikelihoods
• Write posterior in Gibbs form:  \( \pi(x) = \exp\{H(x)\} / Z \)

• Consider a fixed number of \( N \) targets and target classes

• Simulate Langevin diffusion:

\[
dX_N(\tau) = \nabla_{X_N} \{H(X_n(\tau))\} + dW_N(\tau)
\]

• Distribution of  \( X_N(\tau) \xrightarrow{\tau \to \infty} \pi_N(x_n) \)

• Computed desired statistics from the samples

• Generalizes to non-Euclidean groups like rotations

• Gradient computation
  – Numeric approximations
  – Easy and fast on modern 3-D graphics hardware
Diffusion Example on AMCOM Data
Jump Moves

Birth

Death

Type-change

Aaron D. Lanterman 404-385-2548 lanterma@ece.gatech.edu
Helpful Properties of Jump Processes

- Jump at exponentially distributed times
  \[ T^1(x) = \{ \text{states reachable from } x \} \]
  \[ T^{-1}(x) = \{ \text{states from which } x \text{ can be reached} \} \]
- **Move reversibility:** \[ T^1(x) = T^{-1}(x) \]
- **Connectedness:** can go from any point to any other in a finite number of jumps
- Detailed balance (in discrete form)
  \[ \pi(x) \Pr(x \rightarrow z) = \pi(z) \Pr(z \rightarrow x) \]
  (continuous form slightly more complicated)
Jumping Strategies

• Gibbs
  – Sample from a restricted part of the posterior

• Metropolis-Hastings style
  – Draw a “proposal” from a “proposal density”
  – Accept (or reject) the proposal with a certain probability
Example Jump-Diffusion Process
How to Model Clutter?

• Problem: Algorithm only knows about tanks (which Bob doesn’t like anyway), and will try to explain everything with tanks!
  – Cows, swimming pools, school buses

• Solution (?): Let the algorithm use flexible representation in addition to rigid objects
  – Blobs: Simple connected shapes on the lattice to represent structured clutter
  – Could use active curves, level set methods (Clem Karl), active polygons (Hamid Krim)
  – Clutter might be interesting in its own right!
Random Sampling for Blobs

- Set of jump moves
  - Add a pixel along the boundary
  - Remove a pixel along the boundary
  - Keep the blob a blob

- Pick move based on posterior probability
Blob Estimation Examples

M60 spreading

M60 decaying

Ship decaying
NVESD M60 Example
Saccadic Detection

- Current implementation “births” specific target types
- May be better to birth simple shapes, and later change them to more specific target types (clutter or target)
- Example:
  - Birth squares
  - Deform into rectangles
  - Then jump to more detailed targets
Initial Detection

Low-dimensional refinement

High-dimensional refinement

Equilibrium
AMCOM Data Ex.: Finding Tank 2

Initial Detection

Low-dimensional refinement

Equilibrium
AMCOM Data Ex.: Finding ??????

Initial Detection

Low-dimensional refinement
Factory Example
Unified Algorithm

• Extended jump moves
  – Saccadic ↔ blob
  – Saccadic ↔ rigid
  – Blob ↔ rigid
  – Break/combine blobs
  – Change rigid target types

• Difficulty: Make parameters make sense between representation types