AD-A274 388

0

AFIT/GE/ENG/93D-04

DTIC
S ELECTE D
DEC 27 1993
E

THREE DIMENSIONAL OBJECT RECOGNITION USING A COMPLEX AUTOREGRESSIVE MODEL

THESIS

David Eugene Chelen
Captain, USAF

AFIT/GE/ENG/93D-04

112pp

93-31065

Approved for public release; distribution unlimited

93 12 22 178

AFIT/GE/ENG/93D-04

THREE DIMENSIONAL OBJECT RECOGNITION USING A COMPLEX AUTOREGRESSIVE MODEL

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

David Eugene Chelen, B.S.
Captain, USAF

December, 1993

Accesion For
NTIS CRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

DTIC QUALITY INSPECTED 3

Approved for public release; distribution unlimited

Acknowledgements

My Ruckstrations at AFIT have finally come to an end! Before signing off, though, I would like to thank all of the folks who pulled me (kicking and complaining) through these hallowed halls of higher education. Thanks to my advisor, Captain Dennis (Doc) Ruck, I was able to conduct my autoregressive Pattern Ruck research with more unrestricted freedom than that of most students. For keeping me in the right church (just not always the right pew), I thank committee member, Dr Steven Rogers. I thank Dr Mark Oxley, committee member #2, for his attempt to translate my mathematical hieroglyphics into comprehendable expressions. I also give special thanks to my adopted pseudo-committee member and all around good guy, Captain Ken Fielding, for his patience and much appreciated efforts at trying to make that darn HMM work! Oh, well - at least we had some really cool imagery!

On the flip side, I need to thank Capt Kurt Knurr, Lt Tim Pennington, and Lt Curtis Martin, for showing me that computers really can be reasonable (if you push *all* the right buttons). To the Night-Shift crew, we're stupid, but we're gragiatin'! And to my *use or lose* family, a special thanks for your unconditional love and support.

Finally, I thank my Lord and Savior, Jesus Christ, who has held my hand (if not carried me) through more challenges than anyone.

David Eugene Chelen

Table of Contents

Page

Acknowledgements ii

List of Figures vi

List of Tables ix

Abstract x

I. Introduction 1-1
  1.1 Problem 1-1
  1.2 Background 1-2
  1.3 Assumptions 1-2
  1.4 Scope 1-3
  1.5 Approach 1-3
  1.6 Overview of Thesis 1-4

II. Literature Review 2-1
  2.1 Introduction 2-1
  2.2 The Autoregressive Model 2-1
  2.3 Recent Studies 2-2
    2.3.1 Bivariate Autoregressive Model 2-2
    2.3.2 Complex Autoregressive Model 2-4
    2.3.3 Autoregressive Model Classification by Hidden Markov Model 2-5
  2.4 Chapter Summary 2-6

III. Methodology 3-1
  3.1 Introduction 3-1
  3.2 CPARCOR Algorithm Development 3-1

Page

3.3 CPARCOR Algorithm Software Validation 3-3
  3.3.1 Generation of Test Images 3-4
  3.3.2 Load and Enhance Image Subroutine 3-5
  3.3.3 Trace Subroutine 3-6
  3.3.4 Sampling Subroutine 3-8
  3.3.5 Coefficient Stability Test 3-8
  3.3.6 Algorithm Verification Test 3-11
  3.3.7 Section Summary 3-12
3.4 Classifier Application 3-12
  3.4.1 Modifications 3-14
  3.4.2 Template Matching Algorithm 3-16
  3.4.3 K-Nearest-Neighbor (KNN) Techniques 3-25
  3.4.4 Hidden Markov Model (HMM) Testing 3-25
3.5 Chapter Summary 3-29

IV. Results and Discussion 4-1
3.1 Template Test Results 4-1
  4.1.1 Matrix Reduction and Unstored View Testing 4-2
  4.1.2 Occlusion Testing 4-6
  4.1.3 Discussion of Template Test Results 4-12
4.2 K-Nearest-Neighbor (KNN) Test Results 4-13
  4.2.1 Single-Look 1-NN Testing 4-14
  4.2.2 Multiple-Look 1-NN Testing 4-15
  4.2.3 Discussion of KNN Test Results 4-15
4.3 Hidden Markov Model (HMM) Test Results 4-16
4.4 Chapter Summary 4-21

Page

V. Conclusions and Recommendations 5-1

5.1 Conclusions 5-1

5.1.1 Template Matching Algorithm Results 5-2

5.1.2 K-Nearest-Neighbor (KNN) Technique Results 5-2

5.1.3 Hidden Markov Model (HMM) Test Results 5-3

5.2 Recommendations 5-3

5.3 Future Research 5-3

Appendix A. Supplemental Test Results A-1

Appendix B. White Noise Test Source Code B-1

B.1 Introduction B-1

B.2 MATLAB Test Routines and Data Sets B-1

Appendix C. CPARCOR Algorithm Source Code C-1

C.1 Introduction C-1

C.2 CPARCOR Algorithm C-1

Appendix D. Template Matching Algorithm D-1

D.1 Introduction D-1

D.2 Matrix Source Code D-2

D.3 Template Matching Algorithm Source Code D-7

Bibliography BIB-1

Vita VITA-1

List of Figures

Figure Page

2.1. Example of complex coordinates expression. 2-2

3.1. Initial test images: (a) Geometric shapes (b) Typical machine parts. 3-3

3.2. Primary test images. 3-4

3.3. Typical MATLAB image (M60 Tank). 3-5

3.4. Effect of *Mask Subroutine* on typical image. 3-6

3.5. Effect of *Trace Subroutine* on typical image. 3-7

3.6. Effect of *Sampling Subroutine* on typical image. 3-8

3.7. Result of complex white noise test for CPARCOR algorithm. 3-11

3.8. CPARCOR coefficient PSRI comparisons. (a) Feature vectors for scaled and rotated square and machine part (b) Mean value of square and machine feature vectors. 3-13

3.9. Characteristic Views in 360^0 Template (a) Basic template format (b) Representative test images for M60 Tank. 3-18

3.10. Characteristic Views for M60 Tank. 3-20

3.11. Verification of previously reported occluded view recognition rates. Test set consisted of occluded M60 Tank views only. Recognition rates for (a) Dissimilar 3 class (M60, BTR, M35) (b) Similar 3 class (M60, M2, T62). 3-23

3.12. CPARCOR coefficient occlusion comparisons. (a) Feature vectors for M60 characteristic views occluded 0, 5, 10, and 20% (b) Mean value of occluded feature vectors. 3-24

3.13. Example of M60 occluded views for (a) 5% (b) 10% (c) 20%. 3-26

3.14. Example of Mean Distortion Curves (MDC) for cluster centers (Fourier features shown). 3-27

3.15. Example of sphere mapping (Fourier features shown). 3-28

3.16. Example of sphere trajectories. 3-29

Figure Page

4.1. Comparison of 90^0 and 360^0 template for unstored view recognition results with 3 classes. 4-4

4.2. Comparison of 90^0 and 360^0 template for unstored view recognition results with 5 classes. 4-5

4.3. Unstored view recognition results for 3 classes and various 90^0 view templates. . 4-7

4.4. Unstored view recognition results for 5 classes and various 90^0 view templates. . 4-8

4.5. Three class recognition rates for occluded M60 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix. . . 4-10

4.6. Five class recognition rates for occluded M60 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix. . . 4-11

4.7. Comparison of Mean Distortion Curves (MDC) for cluster centers (a) CPARCOR features (b) Fourier features. 4-17

4.8. Comparison of 5 class sphere mapping (a) CPARCOR features (b) Fourier features. 4-19

4.9. Comparison of 5 class (a) Mean and (b) Standard Deviation of CPARCOR feature vectors (unnormalized data). 4-20

A.1. Single-order CPARCOR coefficient recognition analysis: (a) Recognition results for all five classes (b) Mean value of recognition results (all five classes). A-2

A.2. Unstored view comparison using $360^0(3 \times 3)$ for (a) 3 Classes and (b) 5 Classes. A-3

A.3. Comparison of recognition results for 3 and 5 Class, $360^0(3 \times 3)$ template. . . . A-4

A.4. Comparison of $180^0(2 \times 3)$ and $360^0(3 \times 3)$ template for unstored view recognition results with 3 classes. A-5

A.5. Comparison using 180^0 template for unstored view recognition. Results shown for (a) M60 Tank (b) BTR APC (c) M35 Truck A-6

A.6. Comparison using 90^0 $(2 \times 2, 5 \times 5, 10 \times 10)$ and 180^0 $(2 \times 3, 5 \times 9, 10 \times 19)$ templates for unstored view recognition of the M60. . . . . A-7

A.7. Comparison using 90^0 $(2 \times 2, 5 \times 5, 10 \times 10)$ and 180^0 $(2 \times 3, 5 \times 9, 10 \times 19)$ templates for unstored view recognition of the BTR. A-8

A.8. Comparison using 90^0 $(2 \times 2, 5 \times 5, 10 \times 10)$ and 180^0 $(2 \times 3, 5 \times 9, 10 \times 19)$ templates for unstored view recognition of the M35. . . . . A-9

Figure Page

A.9. Comparison of Occlusion results for 15 occluded M60 characteristic views using *dissimilar* 3 class set (M60, BTR, M35). A-10

A.10. Occlusion results for 15 occluded M60 characteristic views using *similar* 3 class set (M60, M2, T62). A-11

A.11. Comparison of 90^0 and 360^0 templates for recognition of occluded M60 characteristic views (a) 3 Classes (b) 5 Classes. A-12

A.12. 3 class recognition rates for 10 occluded M2 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix. A-13

A.13. 3 class recognition rates for 10 occluded M2 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix. A-14

A.14. Mean value of occlusion recognition rates for various size templates. M60 and M35 results shown for 3 and 5 Classes. A-15

List of Tables

Table Page

3.1. White noise test coefficient comparison. 3-10

3.2. Unstored view test set (Class_Azimuth_Elevation). 3-20

3.3. Views occluded 5, 10, and 20%. 3-21

3.4. M60 and M2 views occluded 5, 10, and 20%. 3-22

4.1. Single-Look test results for CPARCOR features. 4-14

4.2. Single-Look test results for Fourier features. 4-14

4.3. Multiple-Look test results for CPARCOR features. 4-14

4.4. Multiple-Look test results for Fourier features. 4-15

AFIT/GE/ENG/93D-04

Abstract

Based on an autoregressive model, Complex Partial Correlation (CPARCOR) features are known to provide exceptional Position, Scale, and Rotation Invariant (PSRI) properties for planar 2-Dimensional (2-D) object recognition. Although autoregressive models have been successfully applied to numerous spatio-temporal recognition tasks, the effects of *out-of-plane* image rotations were never considered. This study investigates application of the CPARCOR model to a five class problem of *nonplanar* 2-D views of 3-D objects. Recognition based on CPARCOR features is evaluated using a Template Matching algorithm, two K-Nearest-Neighbor (KNN) classifiers, and a Hidden Markov Model (HMM). Direct comparisons to recognition based on Fourier features are made. Results indicate that the CPARCOR model parameters provide useful shape-features for recognition of *out-of-plane* rotations. Displaying exceptional PSRI properties, the features are shown capable of classification by simple nonadaptive recognition schemes. Relatively successful results are obtained for a variety of tests. The advantage of classification by a *multiple-look* technique over the traditional *single-look* method is clearly demonstrated. Feature space *crowding* is noted as the cause of unusual recognition rates for occluded-view tests. Although general trends are noted, optimal model order and selection of CPARCOR versus Fourier features are considered application dependent.

THREE DIMENSIONAL OBJECT RECOGNITION
USING A COMPLEX AUTOREGRESSIVE MODEL

I. Introduction

Automatic target recognition capabilities are obviously very important to the Air Force mission. The majority of automatic target recognition systems currently undergoing Air Force research are based on performing a given processing technique on a single frame of sensor (visual or infrared) imagery. Determination of desired information (such as target classification) is made, and then a new frame of imagery is analyzed and the process repeated [6]. Current methods do not account for information contained in the spatio-temporal changes an object undergoes as it moves relative to an observer (or vice versa). This information may be a useful aid in the interpretation of a target's motion and recognition [14].

Based on a complex autoregressive model, Complex Partial Correlation (CPARCOR) features are known to provide exceptional Position, Scale, and Rotation Invariant (PSRI) properties for application to planar 2-Dimensional (2-D) object recognition[17]. Although previous research has successfully applied autoregressive models to spatio-temporal recognition tasks[2][3][9][17], the effects of *out-of-plane* image rotations were never considered.

1.1 Problem

The goal of this thesis is to demonstrate application of the CPARCOR algorithm to spatio-temporal recognition of 3-D objects. Classification based on spatio-temporal information is expected to yield superior results compared to single-frame techniques [7]. For this research, the CPARCOR method of feature extraction will be applied to *nonplanar* 2-D views of 3-D objects. Recognition based on CPARCOR features will be evaluated using the following classification techniques:

-

-

- *Template Matching Algorithm*: Using a Euclidean distance metric, single-frame recognition performance will be evaluated using test sets of ***unstored*** and occluded ***characteristic views***. A characteristic view is basically an object orientation particular to a given class, while unstored views are simply characteristic views that are not included (or stored) in the respective template.
- *K-Nearest-Neighbor (KNN) Techniques*: Recognition of both single and multiple frames of imagery will be performed using the ***hold-one-out*** technique and Single/Multiple-Look 1-NN classifiers. A direct comparison with recognition by Fourier features will be made.
- *Hidden Markov Model (HMM)*: Temporal sequence recognition by CPARCOR features will be attempted for a direct (***apples*** to ***apples***) comparison with recognition by Fourier features.

1.2 Background

Although a variety of shape descriptors have been developed, few have been able to match the PSRI properties of the Complex PARCOR features. Based on an extension of the real PARCOR coefficients (used for speech signal processing), the CPARCOR features are calculated from sampled boundaries of complicated, non-convex 2-D objects. Basic recognition can then be accomplished with a Euclidean metric to measure distance between the coefficients. Exceptional recognition of planar 2-D objects corrupted by both noise and partial occlusion has been reported [17].

1.3 Assumptions

Difficulties encountered with object recognition systems are best summarized in the following statement, "Segmenting a 3-D object cleanly from a complex scene is a very difficult problem in general because of interference from noise, occluding objects, background, illumination, and spatial sampling effects[16:108]." To avoid these problems, the following assumptions for image generation will be made at the outset of this thesis:

- Complete segmentation of the object from a complex scene
- Recognition of only a single object per frame/sequence
- No undesired image corruption due to noise interference

1.4 Scope

The scope of this thesis will focus on an evaluation of CPARCOR features through application of several known classification techniques. Using five classes, imagery will consists of ***nonplanar*** 2-D representations of complicated 3-D images. Particular emphasis shall be devoted to the areas of unstored/occluded (Template Matching), single/multiple-look (KNN), and temporal (HMM) image recognition.

1.5 Approach

Object recognition based on CPARCOR features will be accomplished in the following five steps:

1. The CPARCOR Algorithm will be created using the MATLAB programming environment. Stability of the CPARCOR coefficients will be verified through application of a complex (gaussian) white noise sequence to the model's transfer function. Previously reported classification results (based on CPARCOR features) will then be verified by using simple shapes (squares, circles, triangles, etc.) and a 1-Nearest Neighbor (1-NN) recognition routine.
2. For primary recognition testing, highly detailed test images will be generated from the BRL-CAD software [1]. Test images include the Army's M60 Tank, M35 Truck, BTR60 Armored Personnel Carrier(APC), T62 Tank, and M2 Infantry Fighting Vehicle (IFV). All test images will be 128×128 ASCII arrays, consisting of views taken every five degrees from 0^0 to 355^0 in azimuth and 0^0 to 90^0 in elevation. Recognition performance based on CPARCOR features will be evaluated using several classification techniques (Items 3,4,5).

3. Classification by a Template Matching Algorithm (Euclidean distance metric) will examine the impact of unstored and partially occluded views on single-frame recognition performance. Using unnormalized CPARCOR features, various template sets will be made for object views of 360^0, 180^0, and 90^0 in azimuth. Individual templates will also provide a viewing range from 0^0 to 90^0 in elevation. Using both three and five class problems, the matrix size of each template will vary in order to examine the effective recognition rates.
4. Two versions of a 1-NN classifier will be used to compare the effects of single-look versus multiple-look recognition on unnormalized CPARCOR features. Using a five class problem, recognition of both single-frame and multiple-frame sequences of *uncorrupted* (no noise/occlusion) imagery will be attempted. A direct comparison with recognition by Fourier features will be made.
5. A Hidden Markov Model (HMM) will be applied to determine if CPARCOR features can accurately represent temporally encoded sequences of images. The HMM classifier is known to display exceptional recognition rates (over 97%) for temporal sequences represented by a low frequency, Fourier magnitude feature set [7]. Thus, a direct (*apples* to *apples*) five class performance comparison will be attempted by simply replacing the Fourier features with a set of CPARCOR features.

1.6 Overview of Thesis

This thesis consists of the following chapters:

Chapter II provides a literature review of the most recent studies using autoregressive features for image recognition. In particular, the CPARCOR algorithm is noted for its exceptionally high accuracy when applied to recognition of planar 2-D objects corrupted by both noise and partial occlusion.

Chapter III covers the development and testing of the CPARCOR algorithm. Along with the fundamental principles of the algorithm, the methodology used to verify both the main program and supporting subroutines is described. The chapter concludes with a description of several recognition tests conducted using CPARCOR features and various classifiers.

Chapter IV provides a discussion of the results for tests described in Chapter III. Topics are presented in the same order as they appear in Chapter III. For clarity, additional test results were included in Appendix A.

Chapter V provides conclusions and several recommendations for future research based on the results discussed in Chapter IV.

II. Literature Review

2.1 Introduction

A subset of the most recent studies using autoregressive features for *image recognition* is presented. For this research, image recognition will consist of the classification of objects (such as tanks, trucks, etc.) represented by Complex Partial Correlation (CPARCOR) features.

2.2 The Autoregressive Model

The ability to accurately classify objects in a scene is of great importance with both military and civilian application[2]. Related to the salient features used by humans for scene analysis, classification by features based on the boundaries (or edges) of shapes has shown promising results[5]. Although a variety of shape descriptors exist, three general categories include: scalar transform, space domain, and curve fitting techniques[2]. Considered a method of scalar transform, the *AutoRegressive (AR) model* is noted to perform exceptionally well when applied to the following planar 2-D object recognition tests:

- Classification of scaled, rotated, and translated 2-D planar images[2].
- Classification of complicated, partially occluded images[3].
- Temporal sequence classification of simple, rotated, and occluded 2-D planar images by Hidden Markov Model (HMM)[9].

Receiving widespread application in speech recognition, an *autoregressive model* of order m is defined as a parametric technique that expresses data samples (from an ordered set) as a linear combination of the preceding m samples from the set plus an error term (white noise). For object recognition, model parameters are estimated from a set of the object's boundary samples. Functions of the model parameters can be made invariant to an object's position, scale, and rotation by applying appropriate boundary sampling techniques[3][17]. Although a variety of methods (equal

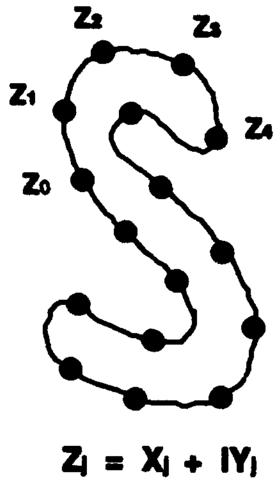


Figure 2.1 Example of complex coordinates expression.

angle, equal curve length, etc.) exists, Figure 2.1[17] shows the boundary representation used by the CAR/CPARCOR model (see Section 2.3.2). In this case, each boundary point is represented by a complex number of form $z_j = x_j + iy_j$. A sequence of complex numbers, z_j, is obtained. From this sequence, an autoregressive model of order m is defined as shown below:

$$z_j = \sum_{k=1}^{m} a_k z_{j-k} + \epsilon_j \qquad\qquad j \in \mathcal{Z} \tag{2.1}$$

where $\mathcal{Z}$ is the set of integers, $\{a_k\}_{k=1}^{m}$ are the CAR coefficients, and ϵ_j represents an error term.

2.3 Recent Studies

The following three sections provide highlights of the most recent studies using autoregressive features for image recognition. Focusing on the methodology applied to each test, this literature review will provide a foundation for further research involving the autoregressive model, CPARCOR, for 3-D object recognition.

2.3.1 Bivariate Autoregressive Model. Monohar Das and others present a bivariate autoregressive model for classification of closed planar shapes[2]. Basically an extension of 1-D Complex

AutoRegressive (CAR) models, a scalar transform method is explored and shown to have certain advantages over the 1-D methods. Shown to be invariant to rotation, translation, and scaling, the bivariate model can be applied to both convex and nonconvex objects without loss of phase information. Depending on the data set, the model allows for flexibility in choice of sampling method, which can be used to bias classification results. Finally, the use of an x, y coordinate system results in lower residual errors than those achieved with 1-D models.

Three data sets (nonoverlapping simple planar shapes) were used to evaluate model sensitivity to variations in boundary shape, larger number of classes, and partial boundary occlusions, respectively. Several shapes were taken from a set of images previously applied to 1-D CAR testing[3]. Using 35 images per class, the shapes from *set one* were arbitrarily rotated, shifted, and scaled within the image plane. For *set two*, 50 images per class were selected in a similar fashion. *Set three*, a subset of *set two* images, consisted of 10 different partial occlusions for each of four shapes (40 total images). Occlusions were made by removing up to 60% of the image. Thresholded to produce binary images, shapes were traced by a boundary follower algorithm. A polygonal approximation was calculated for each shape in order to reduce feature vector variance. Using an equal-curve-length technique, 64 boundary samples were taken for each shape. Classification was based on the feature weighting (FW) and rotated coordinate system (RCS) methods. Model order was limited to four due to 100% classification accuracy achieved with a second order model (for nonoccluded shapes).

Results indicated that the bivariate CAR modeling technique provided greater overall classification accuracy than the 1-D models. Although occlusions were noted to degrade classification performance, the bivariate technique was shown to be less computationally intense than other methods. The bivariate model was also considered ideal for application to automated inspection where occlusions were not likely.

2.3.2 Complex Autoregressive Model. Few shape descriptors have been able to match the Position, Scale, and Rotation Invariant (PSRI) properties of the Complex PARCOR coefficients. Based on an extension of the real PARCOR coefficients (used for speech signal processing), the CPARCOR coefficients are calculated from sampled boundaries of complicated, non-convex 2-D objects. Basic recognition can then be accomplished using a Euclidean metric to measure distance between the coefficients. Exceptional recognition of planar, 2-D objects corrupted by both noise and partial occlusion has been reported [17].

Sekita and others present a *fast algorithm* which generates both 2-D CAR and CPARCOR coefficients of order m. Comparing recognition performance of the CAR/CPARCOR model to various other methods (1-D AR, moment invariants, Fourier descriptors), CPARCOR coefficients are shown to display superior performance. Of the two data sets used, *set one* consisted of a four class set of typical machine parts, while *set two* contained a five class set of 80 different types of plant leaves. Test images were subject to position, scale, and rotational modifications, as well as distortions from partial occlusions and noise. Discriminant analysis was applied to absorb intraclass variations, and the Bayesian decision was used.

PSRI testing used 96 images from *set one* (4 classes × 3 sizes × 8 rotations) and the Rotated Coordinate System (RCS) method. A direct comparison of model orders 1 through 10 revealed that CAR and CPARCOR features were good for recognition of nonconvex, complicated, planar shapes. Additional testing, however, indicated that CAR coefficients were negatively influenced by the number of representative boundary points, while the CPARCOR features remained consistent. Testing of *set two* revealed similar results. Researchers concluded[17] that models of low order contain enough information about the shapes, while higher order models were more sensitive to boundary disturbances.

Occlusion testing, performed on both shape sets, involved images randomly occluded from 5 to 10%. Although CAR and CPARCOR coefficients performed equally well on *set one* (machine parts), both CPARCOR and Fourier descriptors outperformed the CAR coefficients on *set two* data (plant leaves). Also, the negative impact of the number of representative boundary points on the CAR coefficients was again noted. Thus, the CPARCOR coefficients were generally found to be more stable in higher order models than the CAR features.

2.3.3 Autoregressive Model Classification by Hidden Markov Model. He and Kundu[9] present a method that combines the Hidden Markov Model (HMM) with autoregressive parameters for recognition of 2-D planar shapes. Closed (unbroken boundary) shapes are segmented in order to examine the characteristic relations between consecutive segments for purposes of classification at a finer level. Tolerant of moderate amounts of contour perturbation and occlusion, the proposed classifier is also insensitive to orientation of the planar shapes.

The authors'[9] cite the autoregressive model's primary disadvantage as being the use of only *one* set of predictive parameters to model an entire shape. The HMM, on the other hand, is noted to explore the relationship between consecutive segments of a shape's boundary. With the HMM, overall better results were achieved compared to methods that represent the shape with only one set of features. The entire process begins by segmenting the 1-D representation of a closed shape into several pieces. AR modeling is used to characterize each piece, resulting in a vector sequence for each shape. The HMM is then applied to classify the final vector sequence.

Two sets of data were used to test the algorithm. *Set one* consisted of an eight class problem with 30 images per class. Of the images, 24 were individually drawn with moderate boundary perturbations, while 6 were partially occluded. Testing on this set revealed that most initial cluster centers gave very similar results. Recognition rates of 100% were achieved, though, and the authors' note that the correct recognition rate has a tendency to increase as the number of HMM states was increased. Recognition also increased with higher AR model order, where models of orders 4 and 5 gave the best results.

Set two consisted of four country maps (Britain, China, Italy, USA). 30 images were created for each class, six of which were partially occluded. Subject to test procedures identical to those of *set one*, the same overall conclusions were obtained for *set two*. In general, the HMM was found capable of handling 2-D shapes of higher complexity by increasing the number of model states.

2.4 Chapter Summary

This literature review focused on a subset of the most recent studies using autoregressive features for image recognition. Expanding previous test methodologies to include ***nonplanar*** views, this research will be based on results of the articles presented. Displaying exceptional performance and PSRI properties, the CPARCOR model was chosen for application to the task of 3-D object recognition using nonplanar 2-D object-views.

III. Methodology

3.1 Introduction

Citing the work of several research efforts, Chapter II covered a subset of the most recent studies using autoregressive features for image recognition. In particular, the Complex Partial Correlation (CPARCOR) method was noted for exceptionally high accuracy when its features were applied to recognition of planar 2-D objects corrupted by both noise and partial occlusion.

This chapter covers the development (Section 3.2), validation (Section 3.3), and application (Section 3.4) of the CPARCOR algorithm used by this thesis. Along with the fundamental principles of the algorithm, the methodology used to verify both the main program and supporting subroutines is described. The chapter concludes with a description of several recognition tests conducted using the CPARCOR algorithm and various classifiers.

3.2 CPARCOR Algorithm Development

This section highlights development of the CPARCOR algorithm as used by this thesis. Derivation of the CPARCOR algorithm is also explained in the article by Sekita and others[17]; however, only the most important equations required for implementation are presented here.

Although a variety of autoregressive shape descriptors exist[2][3][9][17], few are able to match the Position, Scale, and Rotation Invariant (PSRI) properties of the Complex PARCOR features. Basically an extension of the real PARCOR coefficients (for speech signal processing), the CPARCOR features are calculated from sampled boundaries of complicated, non-convex 2-D objects. Recognition can then be accomplished using a Euclidean metric to measure distance between the coefficients.

Generation of the CPARCOR coefficients begins by representing each point of an object's boundary as a complex number of form $z_j = x_j + iy_j$. In this manner, a vector sequence of complex numbers, z_j, is obtained. From this sequence, a Complex AutoRegressive (CAR) model of order m is defined by a linear combination of the m preceding boundary points [17], or as shown below:

$$z_j = \sum_{k=1}^{m} a_k z_{j-k} + \epsilon_j \qquad j \in \mathcal{Z} \tag{3.1}$$

where $\mathcal{Z}$ is the set of integers, $\{a_k\}_{k=1}^{m}$ are the CAR coefficients, and ϵ_j represents an error term. Based on this model, both CAR and CPARCOR coefficients for a model of order m are generated by recursively applying the following three equations:

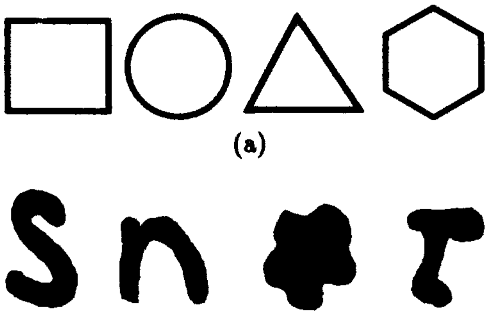
$$\mathbf{a}(1) = r_1/r_0 \tag{3.2}$$

$$\mathbf{a}(m) = \begin{bmatrix} \mathbf{a}(m-1) - \bar{\mathbf{a}}(m-1)^{\updownarrow} p_m \\ p_m \end{bmatrix}. \tag{3.3}$$

$$p_m = \frac{(r_m - \mathbf{r}(m-1)^{\mid}\mathbf{a}(m-1))}{(r_0 - \mathbf{r}(m-1)^T\bar{\mathbf{a}}(m-1))} \tag{3.4}$$

The individual terms for Equations 3.2 to 3.4 are defined as follows:

- $\mathbf{a}(m) = [a_1, a_2, a_3, ..., a_m]^T$ is a vector of CAR coefficients.
- p_m represents the mth order CPARCOR coefficient.
- $r_k = (1/N)\sum_{j=0}^{N-1} z_j \bar{z}_{j-k}$ is the complex autocorrelation of z_j.
- $\mathbf{r}(m) = [r_1, r_2, r_3, ..., r_m]^T$ is a vector of complex autocorrelation coefficients.
- $\bar{\mathbf{a}}$ represents the complex conjugate of $\mathbf{a}$, while $\mathbf{a}^T$ is the vector transpose of $\mathbf{a}$.
- $\mathbf{a}(m-1)^{\updownarrow}$ represents the vector $[a_{m-1}, a_{m-2}, ..., a_1]^T$ with reverse order elements.
- $\mathbf{r}(m-1)^{\mid}$ is the transpose of $\mathbf{r}(m-1)^{\updownarrow}$



(a)

(b)

Figure 3.1 Initial test images: (a) Geometric shapes (b) Typical machine parts.

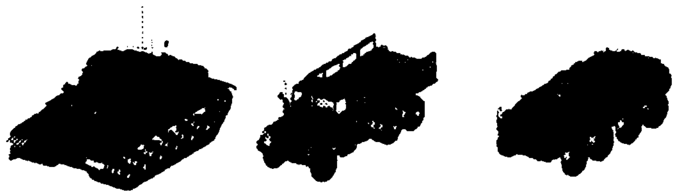
Based on results of algorithm-validation tests (Section 3.3), it should be noted that Equation 3.3 was published incorrectly in the original article by Sekita and others [17]. The *correct* version is as shown. Although the CPARCOR algorithm generates both, the CAR coefficients tend to display considerably lower recognition rates when compared to the CPARCOR coefficients (for identical tests) [17]. Thus, *only* the CPARCOR coefficients will be considered further.

3.3 CPARCOR Algorithm Software Validation

The following subsections provide an extensive examination of the CPARCOR algorithm used by this thesis. Along with a detailed explanation of each subroutine in Sections 3.3.1 to 3.3.4, the methods used to validate the algorithm are also described in Sections 3.3.5 to 3.3.6.

In addition to the primary CPARCOR routine defined by Equations 3.2 to 3.4, the following preliminary tasks/subroutines were required in order to generate software compatible test images:

1. *Generation of Test Images*: Creates initial and primary test images.
2. *Load and Enhance Subroutine*: Creates a smooth, unbroken image boundary.
3. *Trace Subroutine*: Traces image and calculates the boundary's centroid.
4. *Sampling Subroutine*: Provides a sampled version of the image's traced boundary.



(a) M60 Tank (b) M35 Truck (c) BTR APC

(d) T62 Tank (e) M2 IFV

Figure 3.2 Primary test images.

3.3.1 Generation of Test Images. Test image generation involved a two step process. First, eight different classes of preliminary test data were generated in order to validate both the CPARCOR algorithm and previous research results [17]. Examples of the initial test images are shown in Figure 3.1. Images in Set(b) are representative of the actual data set used by others [3][17]. Scaled (50%) and rotated (30^0) versions of the eight classes shown were also created to verify the PSRI properties of the CPARCOR coefficients. Initial testing was conducted using four images from each class (32 total images).

Second, in order to attempt three dimensional (3-D) image recognition, 2-D representations of 3-D images had to be created. This was accomplished with the aid of several software routines created at AFIT [6] and the Army's constructive solid geometry based computer aided design package, BRL-CAD[1]. Images (128 x 128 arrays) were generated for five different classes which included the Army's M60 Tank, M35 Truck, BTR60 Armored Personnel Carrier(APC), T62 Tank, and M2 Infantry Fighting Vehicle(IFV). A representative view from each of the five classes is shown in Figure 3.2.

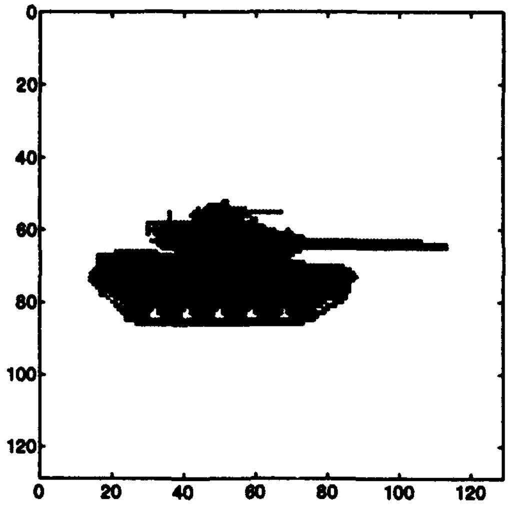


Figure 3.3 Typical MATLAB image (M60 Tank).

Image views were generated every five degrees in both azimuth (from 0^0 to 355^0) and elevation (from 0^0 to 90^0), while various subsets of the total image set were used for actual test applications. For future discussion, all views will be referenced by the following format:

CLASS_Azimuth_Elevation

For example, the front view of an M60 Tank would be labeled as M60_0_0. Similarly, the right side of an M35 Truck would be labeled as M35_90_0.

3.3.2 Load and Enhance Image Subroutine. A typical image, as stored in a MATLAB array, is shown in Figure 3.3. For compatibility with the *Trace Subroutine* (Section 3.3.3), the image pixels were thresholded for two grey scale levels to represent either a background pixel (zero) or a target pixel (one).

As previously mentioned, the image's boundary must be traced and sampled before the CPARCOR algorithm can be applied. Obviously, it would be trivial for a human to trace the perimeter of the image shown in Figure 3.3. However, notice that certain *gaps* exist along the image's boundary due to the use of a 128 x 128 format. For the *Trace Subroutine* to accurately follow the image's

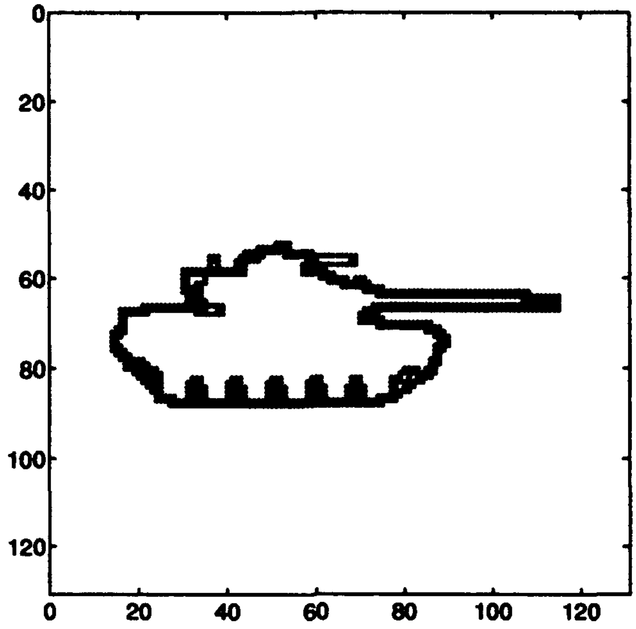


Figure 3.4 Effect of *Mask Subroutine* on typical image.

boundary without getting stuck, these gaps need to be filled in. Thus, to create the smooth, unbroken boundary required, an edge detection mask is first applied to enhance the image prior to boundary tracing. Enhancing both horizontal and vertical lines [13], the effects of the edge detection mask are shown in Figure 3.4.

3.3.3 Trace Subroutine. Calculation of CPARCOR coefficients is based on a sampled version of the original image's boundary. For the CPARCOR algorithm, sampling is performed on on the object's boundary using a simple boundary detect/trace routine. Based loosely on the *Turtle* algorithm [3][4], the *Trace Subroutine* first detects and then follows the image's boundary in a clockwise direction. The basic algorithm is outlined as follows:

1. Scan the image until a target pixel (one) is found.
    - If the *turtle* is in a target pixel, turn left and move one pixel forward.
    - If the *turtle* is in a background pixel, turn right and move one pixel forward.
2. Continue tracing the boundary until the *turtle* has found the starting point

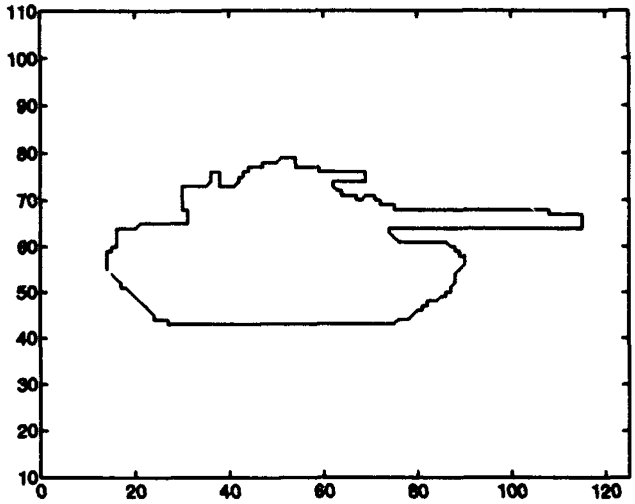


Figure 3.5 Effect of *Trace Subroutine* on typical image.

As discussed in the previous section, two conditions must be satisfied in order for the *Trace Subroutine* to function properly. First, the image (to be traced) must be clearly defined in terms of grey scale values. This is accomplished by simply thresholding the image for two grey scale levels, either background pixel (zero) or target pixel (one). Second, the boundary must not have any spurious gaps that will cause the *turtle* to get stuck inside the image. Again, to create the unbroken boundary required, an edge detection mask is applied prior to boundary tracing.

The *Trace Subroutine* returns the locations of all boundary pixels. A thinning routine is then applied to eliminate any repeated points which sometimes occur when the *turtle* goes around a corner. The image's centroid is finally calculated from the *thinned* sequence of boundary pixels. To insure coefficient invariance to boundary translations, the centroid is used to represent the origin of the model's complex-coordinate system. Basically, by making the coefficients invariant to both translation and rotation about the origin, they are automatically invariant to any rotation [2][3][17]. Figure 3.5 shows the resulting image after application of the *Trace Subroutine*.

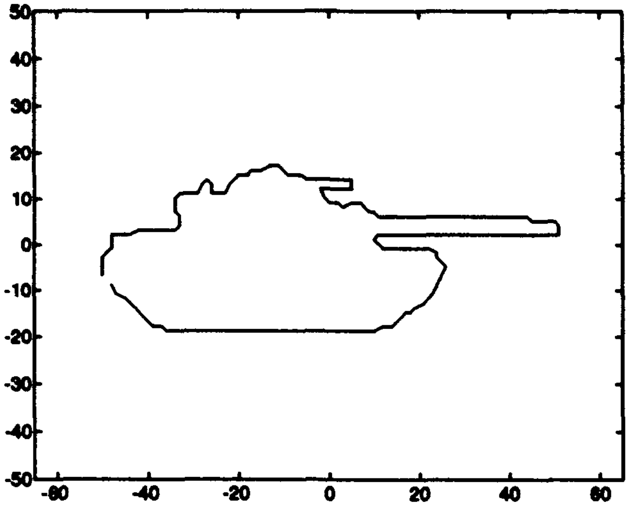


Figure 3.6 Effect of *Sampling Subroutine* on typical image.

3.3.4 Sampling Subroutine. After the centroid has been calculated, a slightly modified version of the *Trace Subroutine* is applied. This time, the image's coordinate system is referenced from the centroid (for invariance). The modified *Trace Subroutine* returns a sequence containing the boundary's coordinates (complex numbers), which are then used by the *Sampling Subroutine*. Although the sampling rate can be user specified, it is recommended that the number of samples be fixed to a constant (for each class of images) in order to provide better PSRI properties [17]. In fact, no significant change in the recognition rate according to the number of representative boundary points was noted for previously reported results [17]. For this thesis, sample rates were chosen to provide an accurate, though not exact, representation of the actual image. For example, 30 samples were used for tests on objects shown in Figure 3.1, while 160 samples were used for all objects shown in Figure 3.2. An example of an M60's right-side view (M60_90_0) recreated from 160 samples is shown in Figure 3.6.

3.3.5 Coefficient Stability Test. As outlined at the start of this chapter, the CPARCOR coefficients of higher order models can be obtained by recursive application of Equations 3.2 to

3.4. To verify stability of the coefficients produced by these equations, a complex white noise test was conducted on the algorithm's transfer function. The next two sections will discuss the transfer function's derivation and the procedures followed for the white noise test, respectively.

3.3.5.1 Transfer Function Derivation. CPARCOR coefficients, an extension of the real PARCOR coefficients, are a direct result of the Complex AutoRegressive (CAR) algorithm. Based on the conventional real autoregressive model, a CAR model of order m is defined by

$$z_j = \sum_{k=1}^{m} a_k z_{j-k} + \epsilon_j^f \qquad j \in \mathcal{Z} \tag{3.5}$$

where $\mathcal{Z}$ is the set of integers, $a_k{}_{k=1}^{m}$ are the CAR coefficients, and ϵ_j^f represents an error term. In general, z_j is defined as a linear combination of the preceding m boundary points.

Assuming inputs to the sequence are unknown, an estimate of z_j can be defined by

$$\hat{z}_j = \sum_{k=1}^{m} a_k z_{j-k} \tag{3.6}$$

The error between the estimate and the actual is given as

$$e(n) = \hat{z}(n) - z(n) = \epsilon_j^f \tag{3.7}$$

Rearranging terms and taking the Z-transform[12][15] of Equation 3.5 yields:

$$Y(z)[1 - \sum_{k=1}^{m} a_k z^{-k}] = X(z) \tag{3.8}$$

The resulting transfer function is defined as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \sum_{k=1}^{m} a_k z^{-k}} \tag{3.9}$$

3.3.5.2 White Noise Test Procedures. To verify stability of the CPARCOR coefficients, a complex white noise test was conducted on the algorithm's transfer function (Equation 3.9) in the following manner:

1. Generate a random sequence of complex white gaussian noise (10,000 samples).
2. Generate stable (within the unit circle) test coefficients for a model of order six (arbitrary).
3. Filter the complex white noise sequence using the filter described by the test coefficients and autoregressive transfer function.
4. Apply the filter's output vector (impulse response) to the CPARCOR algorithm.
5. Create a pole-zero plot of the calculated CPARCOR coefficients. A stable sequence will result in points that lie within the unit circle.

Table 3.1 White noise test coefficient comparison.

Order \ Vector	Original	CPARCOR Algorithm
1	-0.1807	-0.1845 + 0.0017i
2	-0.1023	-0.0993 - 0.0007i
3	-1.2578	-1.2522 + 0.0048i
4	0.0226	0.0231 - 0.0018i
5	0.0939	0.0922 - 0.0025i
6	0.7165	0.7138 + 0.0006i

Table 3.1 provides a direct comparison of the original test coefficients to those generated by the CPARCOR algorithm. The coefficients are shown to be a reasonable representation of the original set. As shown in Figure 3.7, all poles (coefficients) also lie within the unit circle. Thus, the algorithm does in fact produce stable CPARCOR coefficients. MATLAB software routines and relevant data sets used for this test are provided in Appendix B, while the source code for the CPARCOR Algorithm is provided in Appendix C.

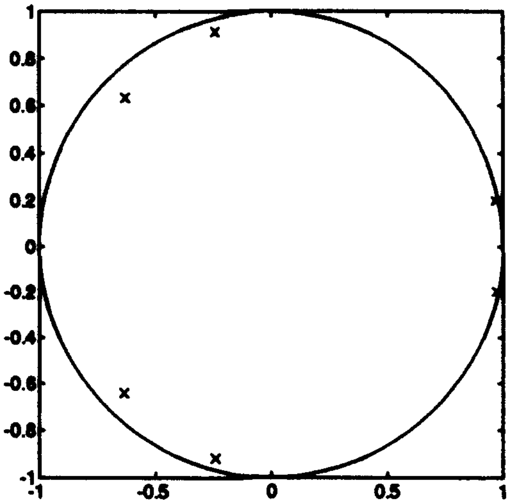


Figure 3.7 Result of complex white noise test for CPARCOR algorithm.

3.3.6 Algorithm Verification Test. This section details the initial test used to verify operation of the entire CPARCOR algorithm. Also included is a look at the nature of the CPARCOR coefficients when subject to changes in scale and rotation.

As highlighted in Chapter II, the CPARCOR algorithm presents an exceptional invariant-feature extraction model for recognition of arbitrary planar objects. Previous experimental results indicated that complicated and partially occluded shapes could be recognized with high accuracy, even with low-order models [17]. Based on these results, several simple test images were generated in order to verify the proper operation of the CPARCOR algorithm, the stated PSRI properties, and the outcomes of previous research efforts [17]. Examples of the initial eight class test set were shown in Figure 3.1 on Page 3-3. Four images (including scaled and rotated versions) were created per class, for a total of 32 images. A tenth order CPARCOR model was created for each image (30 boundary-samples each).

Classification of the initial test set was done using the *hold-one-out* method and a 1-Nearest-Neighbor (1NN) classifier[10]. A twenty dimensional feature vector was created (from a tenth order model) for each class by separating the real and imaginary parts of the complex coefficients [17]. Consistent with previous research, one hundred percent classification results were achieved.

The next logical step was to more fully explore the nature of the CPARCOR coefficients when subject to changes in scale and rotation. To this end, a plot of the scaled and rotated CPARCOR feature vectors was generated for the first image in each set of Figure 3.1. Figure 3.8 (a) is a plot of the original, the scaled(50%), and the unscaled/rotated(30^0) feature vectors for each image. Although some intra-class variation exists, the two classes are clearly separable. To further illustrate class separability, the mean of the three feature vectors for each class was plotted in Figure 3.8 (b). This figure indicates that separability continued to exists, even at higher order models, when the image was subject to scale and rotational changes.

3.3.7 Section Summary. Section 3.3 provided a detailed explanation of the CPARCOR algorithm as well as the methodology used for its validation. The algorithm was shown capable of producing stable CPARCOR coefficients, and the previously reported PSRI properties and high recognition rates of CPARCOR features [17] were verified. Next, Section 3.4 will describe the three different classifiers used to examine recognition performance of the CPARCOR features on nonplanar 2-D images (of 3-D objects). A discussion of the modifications made to the methodology traditionally applied (when classifying by autoregressive models) is also included.

3.4 Classifier Application

Although the *AutoRegressive (AR)* model was previously noted to perform exceptionally well when applied to various planar 2-D object recognition tests, the goal of this thesis was to demonstrate application of the CPARCOR algorithm to spatio-temporal recognition of 3-D objects. As such, the CPARCOR method of feature extraction was applied to ***nonplanar*** 2-D views of 3-D objects. Recognition based on the CPARCOR features was then evaluated using the several different classification techniques.

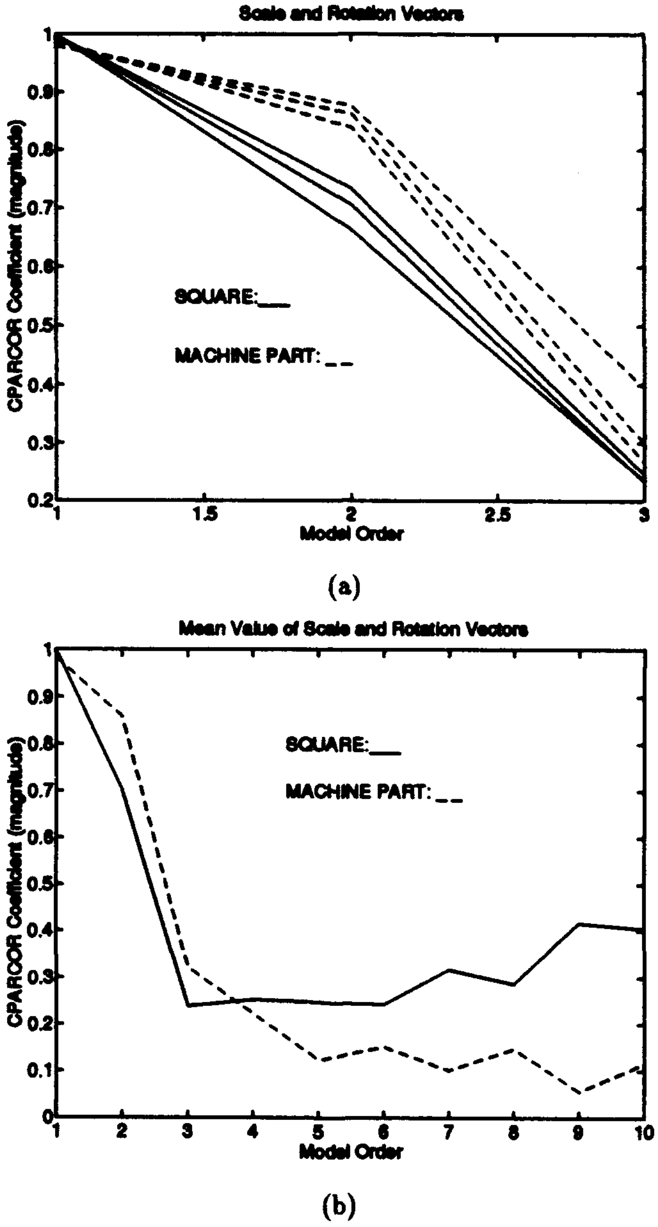


Figure 3.8 CPARCOR coefficient PSRI comparisons. (a) Feature vectors for scaled and rotated square and machine part (b) Mean value of square and machine feature vectors.

Providing an explanation of each classifier applied for recognition testing, this section is broken down as follows:

- Section 3.4.1 contains a discussion of several modifications made to the methodology traditionally applied when classifying by autoregressive models. Also highlighted is a brief summary of the classifiers applied for recognition testing.
- Section 3.4.2 provides an extensive explanation of the *Template Matching* algorithm that was developed specifically for this thesis. Included in Section 3.4.2.1 are the baseline hypotheses used to evaluate the template algorithm's experimental results. Additionally, Sections 3.4.2.2 and 3.4.2.3 provide details of specific tests conducted.
- Section 3.4.3 provides a discussion of the *Single-Look* and *Multiple-Look* 1-NN classifiers used to test their respective recognition techniques on unnormalized CPARCOR feature vectors.
- Section 3.4.4 details the procedures of the Hidden Markov Model recognition test. For clarity, examples based on Fourier features are included in the discussion, while results for CPARCOR features are not presented until Chapter IV.

A discussion of the results for all tests is included Chapter IV. For clarity, additional results for some of the *Template Matching* algorithm tests are included in Appendix A.

3.4.1 Modifications. To more fully examine the robust nature of CPARCOR coefficients, several modifications were made to the methodology traditionally applied when classifying by autoregressive models [2][3][9][17].

First, the complexity and number of stored images (per class) was significantly increased to determine the impact of *crowding* the feature space. The images of Figure 3.1 on Page 3-4 are representative of those used for previous research efforts [2][3][9][17]. Clearly, they pale in comparison to the complexity of the test images shown in Figure 3.2 on Page 3-4. Also, previous efforts typically used well under 100 images per class. This thesis, on the other hand, used anywhere from four to several hundred images per class (depending on the application).

Second, the relationship between the number of ***characteristic views*** and the corresponding recognition rate was explored through application of a Template Classifier. A characteristic view is an object orientation, particular to a given class, in which surrounding views yield a similar set of observed features. For this thesis, the minimal number of characteristic views were defined as the ***front, back, top, left,*** and ***right side*** views of each class. Basically, various dimensional templates were used to classify views ***not*** stored in the templates. This effort was essentially an extension of previous work by Seibert and Waxman [16].

Third, the impact of partial occlusion on the corresponding recognition rate was examined by using ***known*** levels of occlusion. For example, image occlusions were separated into three distinct categories: 5%, 10%, and 20% levels of partial occlusion. Although previous researchers included levels of up to 60% partial occlusion [9], no distinction was ever made (in terms of recognition) between the effect a 5% versus a 60% level of occlusion.

Finally, all classification attempts were based on ***nonplanar*** views. Previous research efforts simply rotated the same image ***in-plane*** for classification testing [2][3][9][17]. For this research, however, views for each of the five classes (reference Figure 3.2) were generated in five degree increments from 0^0 to 355^0 in azimuth and 0^0 to 90^0 in elevation. Images were generated in this manner to determine if a spatio-temporal relationship existed among the characteristic views for each class.

To evaluate the CPARCOR coefficients ability to provide robust PSRI features for recognition of nonplanar 2-D images (of 3-D objects), the following methods of classification were applied:

- *Template Matching Algorithm*: Using a Euclidean distance metric, the impact of unstored and partially occluded views on single-frame recognition performance was examined. Recall that unstored views were defined as characteristic views that are not included (or stored) in the respective template.

- *K-Nearest-Neighbor (KNN) Techniques*: Recognition of both single and multiple frames of imagery was performed using the ***hold-one-out*** method and Single/Multiple-Look 1-NN classifiers. A direct comparison with recognition by Fourier features was made.
- *Hidden Markov Model (HMM)*: Temporal sequence recognition by CPARCOR features was attempted for comparison with recognition by Fourier features.

3.4.2 Template Matching Algorithm.

The relationship between the number of characteristic views and the corresponding recognition rate was explored through application of a template classifier. Providing a list of baseline hypotheses for performance evaluation, this section will examine the basic function of the classifier algorithm. Also included is an explanation of the tests conducted on data sets of ***unstored*** and ***occluded*** characteristic views.

Template matching is a simple, but effective method of pattern recognition. Basically, an input of ***unknown-class*** is compared to a set of ***known-class*** prototypes[18]. Recognition (class identification) is determined by establishing the template's class containing the ***closest match*** to the unknown input. Of the various means available to establish the ***matching*** criteria, this thesis will use a Euclidean distance metric defined as follows:

$$D(x) = \min_i \| \mathbf{x} - \mathbf{z}_i \| \tag{3.10}$$

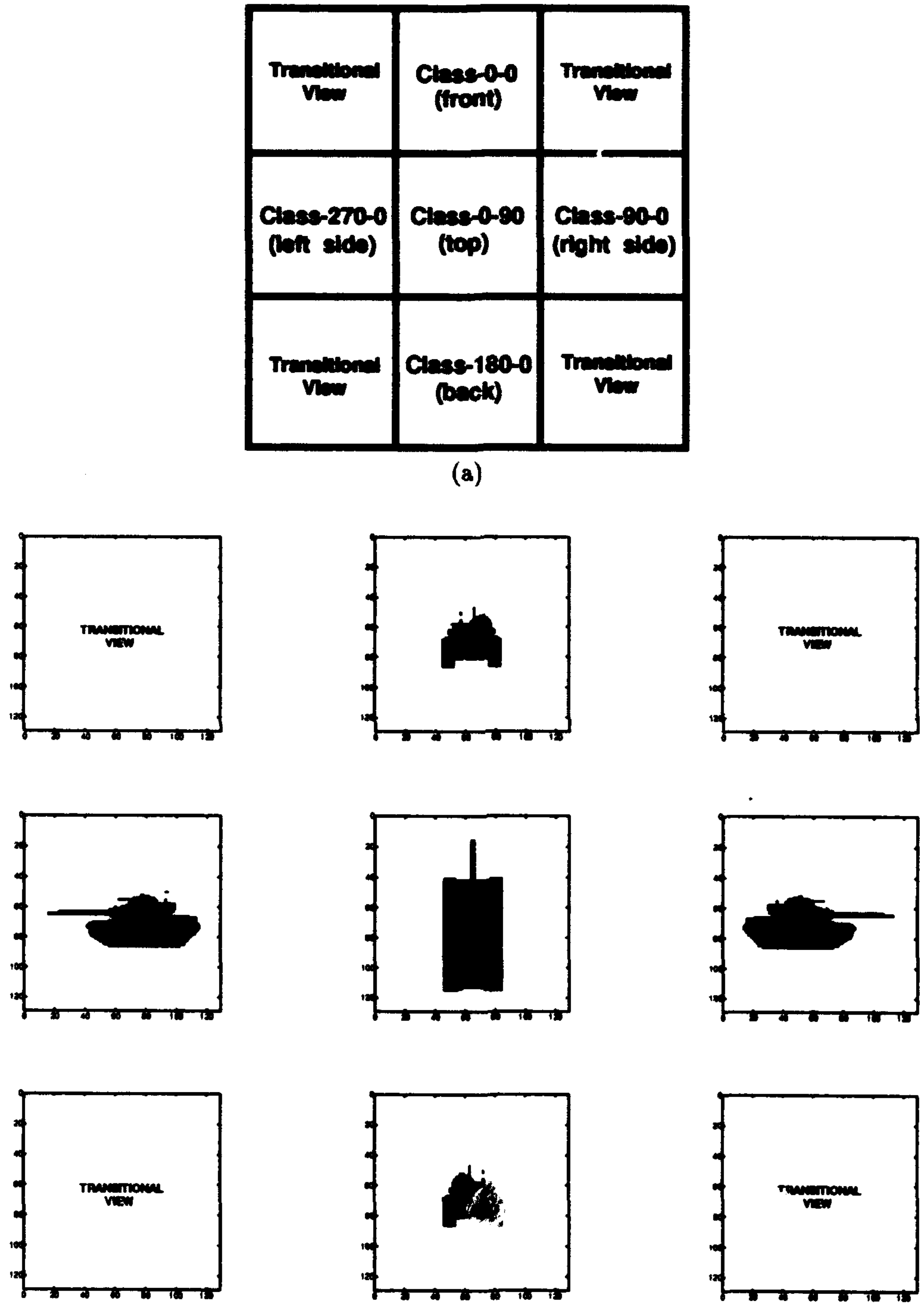
where $\| \cdot \|$ defines the Euclidean norm for the expression, and $D(x)$ is the smallest of all distances between the unknown input-view ($\mathbf{x}$) and the ith stored template-view ($\mathbf{z}_i$)[18]. In other words, classification is based upon the minimum Euclidean distance between a ***known*** template-view and the ***unknown*** input-view. Using this metric, several template sets were developed in order to examine the relationships between ***unstored/occluded*** views and the number of ***stored*** (in the template) characteristic views. Templates were based on **m** × **n** matrices containing unnormalized CPARCOR feature vectors that represent the characteristic views for various allowable object orientations. The following templates were used for testing:

- **360^0 View Template (3×3 Matrix)**: Contains the minimum number of characteristic views (CPARCOR feature vectors) required for total-range 360^0 azimuth and 90^0 elevation image recognition.
- **180^0 View Template ($2 \times 3, 5 \times 9, 10 \times 19$ Matrix)**: Representing the same 180^0 view, the three matrices contain different numbers of characteristic views. For example, the 2 × 3 matrix contains six characteristic views, while the 10 × 19 contains 190.
- **90^0 View Template ($2 \times 2, 5 \times 5, 10 \times 10, 19 \times 19$ Matrix)**: Although representative of the same 90^0 view, the 2 × 2 matrix uses only four characteristic views, while the 19 × 19 uses 361. Note that the 19 × 19 contains the maximum number of characteristic views possible for the given range. Thus, the 19 × 19 matrix was not used for *unstored* view recognition tests.

An illustrative example of the 360^0 View Template (3 × 3 Matrix) is shown in Figure 3.9 on Page 3-18. Note that the CPARCOR feature vectors, *not* the images, are actually stored in the template. Although other possibilities exist, the transitional views (in each corner) for this template were represented by the following:

- Upper left corner: Class_315_45
- Upper right corner: Class_45_45
- Lower right corner: Class_135_45
- Lower left corner: Class_225_45

The *Template Matching* algorithm returns the matrix location and corresponding class label of the best *match* for a given unknown-input view. The source code for this algorithm and the various templates is included in Appendix D.



(b)

Figure 3.9 Characteristic Views in 360^0 Template (a) Basic template format (b) Representative test images for M60 Tank.

3.4.2.1 Baseline Hypotheses. The *Template Matching* algorithm was used to perform a variety of experiments on test sets of both unstored and partially occluded characteristic views of the five classes shown in Figure 3.2 (Page 3-4). Experimental results were evaluated against the following baseline hypotheses:

- Recognition performance should *increase* with model order until a point of optimal performance is achieved. After this point, no addition benefit will be realized by using higher order models.
- For a given model order, recognition performance should *decrease* as the number of available classes is *increased.*
- Object symmetry should allow for a *reduction* in overall template and matrix size. For example, the 360^0 View Template should provide almost identical recognition rates as the 180^0 View Template.
- As the matrix size of a given template is *increased* (to include more characteristic views), recognition performance for unstored views should also *increase.* The exception to this rule would be recognition of partially occluded views (see next item).
- Recognition of partially occluded views should *decrease* as the matrix size of a given template is *increased.* Generally, as more characteristic views are added, the occluded view has a higher probability of being misclassified due to the existence of many similar choices.
- Recognition performance should *decrease* as the percentage of partial occlusion is *increased* from 5% to 20%.

3.4.2.2 Matrix Reductions and Unstored View Testing. This section presents the motivation behind an attempt to reduce the overall template size required for reliable recognition. Initially presented is an examination of symmetrical characteristic views. This is followed by an explanation of how *unstored* views were used to test the template/matrix reduction hypothesis.

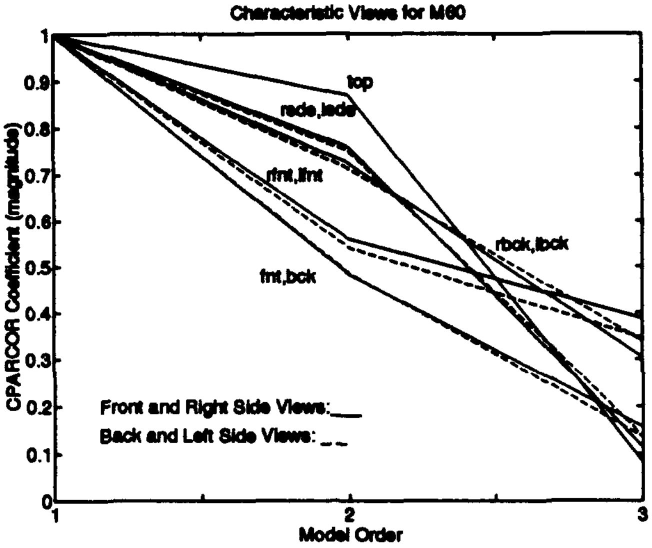


Figure 3.10 Characteristic Views for M60 Tank.

Based on the assumptions of Section 3.4.2.1, preliminary template testing was focused on ways to reduce the overall template size. A straight forward approach to this problem would seem to involve using an object's symmetry to reduce the number of *stored* characteristic views. Thus, a plot of the M60 Tank's characteristic views, as stored in the 360^0 view template, was made in order to examine the CPARCOR coefficients. As shown in Figure 3.10, similar feature vectors are indeed generated for views separated by 180^0 in azimuth.

Table 3.2 Unstored view test set (Class_Azimuth_Elevation).

Class_0_5	Class_5_5	Class_5_35	Class_15_15	Class_15_55
Class_25_5	Class_25_45	Class_35_25	Class_35_55	Class_45_15
Class_45_45	Class_45_75	Class_55_25	Class_55_55	Class_60_65
Class_65_25	Class_65_45	Class_65_55	Class_75_15	Class_75_55
Class_85_5	Class_85_35	Class_85_45	Class_90_15	Class_90_85

Taking advantage of this information, an attempt to reduce the overall template size was conducted by directly comparing the recognition accuracy of templates for 360^0, 180^0, and 90^0 views. Sets of various sized matrices were used to test the recognition rate of each template on a given test set of unstored views. A baseline for comparison was established by restricting the unstored views to the set shown in Table 3.2. Representative results of the matrix reduction and the unstored view tests are fully discussed in Chapter IV. For clarity, additional results were included in Appendix A.

3.4.2.3 Occlusion Testing. One of the more difficult problems associated with image recognition is the classification of partially occluded object views[3]. This section will explain the logic behind partial occlusion tests conducted with the template algorithm. Initially presented are results of a test performed to verify previously reported experimental results [3][17]. Next, an examination of actual CPARCOR feature vectors subject to various levels (0, 5, 10, and 20%) of occlusion is shown. Finally, the methodology used specifically for this thesis is explained.

Table 3.3 Views occluded 5, 10, and 20%.

M60_0_0	M60_0_0(new)	M60_0_90	M60_0_90(new)	M60_90_0

Before examining the effects of *known* levels of partial occlusion, test results reported by others [3][17] were verified using the five M60 Tank views shown in Table 3.3. Note that *only* the M60 class was used for this test. M60 views were occluded 5, 10, and 20% for a total test set of 15 occluded views. Two occluded versions of the M60's front and right side views (indicated by 'new') were included in order to test the effect moving an occlusion to a *new* location on the same view. Using a template set of **2** × **2** matrices (one per class), the test set of occluded M60 views (Table 3.3) was applied to both a three (M60, BTR, M35) and a five class problem. Shown in Figure 3.11 (a), results of the three class comparison readily verify the high recognition rates reported by previous efforts. Note that the recognition rate of *zero* for order *one* was valid due to the use of only one class for testing. The significant drop in recognition for the five class case was thought to be due to confusion caused by the addition of two more *tracked* vehicles (M2, T62). To verify this, Figure 3.11 (b) compares the results of the five class problem to results of a more *similar* three class case (M60, M2, T62). Given that the same line is defined for both cases, the drop in recognition for Figure 3.11 (a) was clearly caused by confusion related to the two additional classes of similar *tracked* vehicles. Again, the *zero* recognition rate (order *one*) was valid due to the use of a one-class test set. Previous efforts [3][17] reported higher classification rates by performing tests on several *less similar* classes.

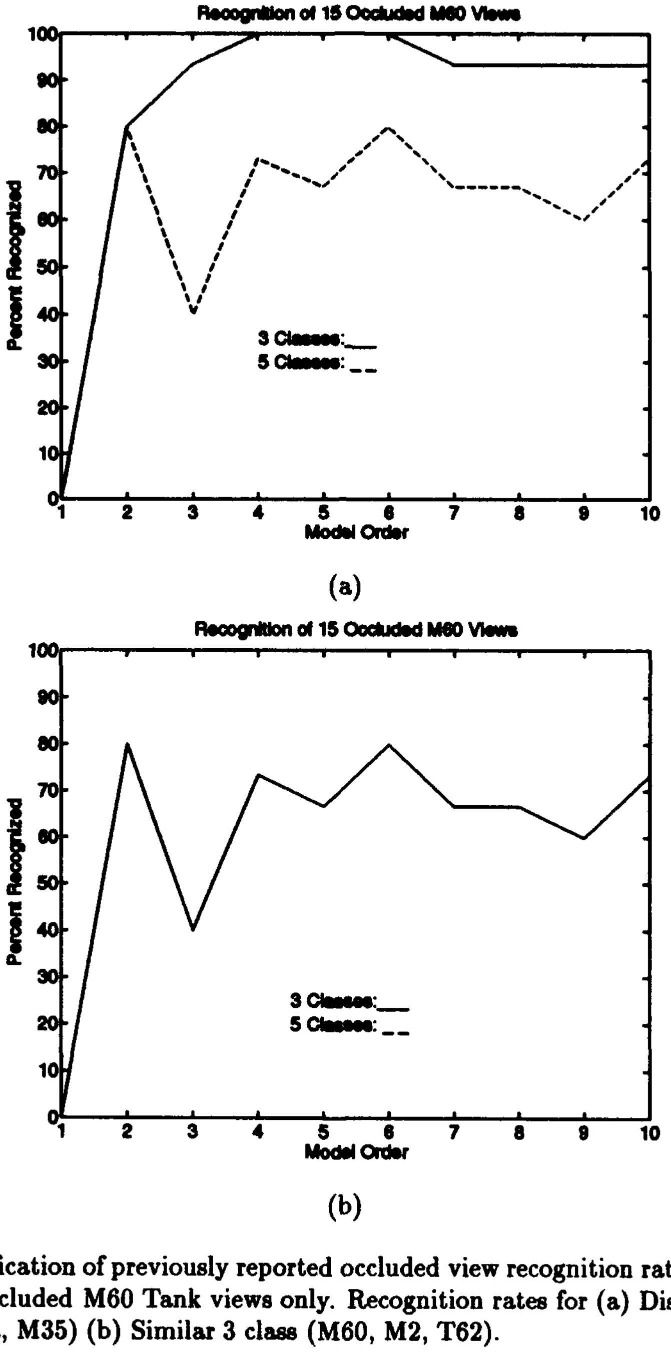
A plot of two feature vectors for occluded M60 characteristic views was made in order to examine CPARCOR coefficients subject to various levels (0, 5, 10, and 20%) of occlusion. As shown in Figure 3.12 (a) on Page 3-24, similar feature vectors are indeed generated for occlusions of the same view. To further illustrate this point, a plot of the overall mean for each set of vectors was made. As shown in Figure 3.12 (b), the two characteristic views are clearly separable.

Table 3.4 M60 and M2 views occluded 5, 10, and 20%.

M60_0_0	M60_15_15	M60_30_30	M60_0_45	M60_45_0
M60_45_45	M60_65_65	M60_75_25	M60_0_90	M60_90_0
M2_0_0	M2_15_15	M2_30_30	M2_0_45	M2_45_0
M2_45_45	M2_65_65	M2_75_25	M2_0_90	M2_90_0

Based on this information, the impact of partial occlusion on the corresponding recognition rate was examined by using *known* levels of occlusion and various template sizes. Separated into three distinct categories (5%, 10%, and 20%), 60 occluded images (total) were made for the M60 Tank and the M2 IFV from the set listed in Table 3.4. The two classes of *tracked* vehicles were chosen in order to test the robustness of recognition on occluded, but similar objects. Also, to make recognition more difficult, all *three-class* testing was performed using the M60, M2, and T62 (tracked vehicles). Occlusions were 'hand-made' by removing a number of image pixels (from the 128 x 128 array) corresponding to the desired level of occlusion. A subset of occluded M60 views is shown in Figure 3.13 on Page 3-26. Chapter IV contains representative results and a discussion of these tests. For clarity, additional results were included in Appendix A.

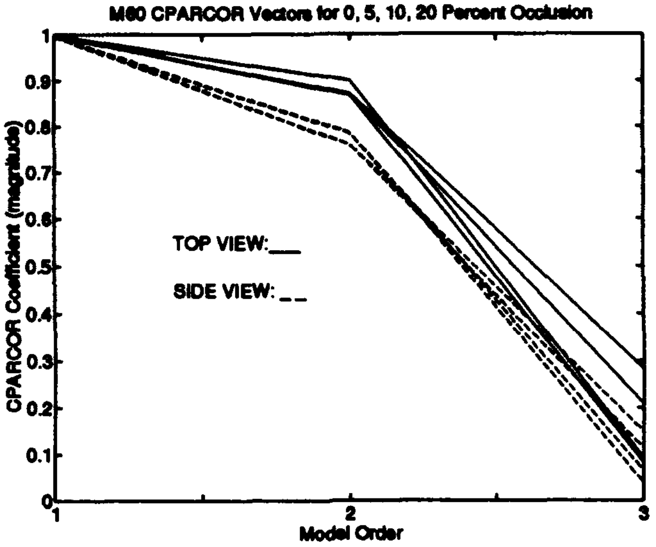
The following sections present the methodology applied to two additional classifiers used by this thesis. Section 3.4.3 provides a discussion of the *Single-Look* and *Multiple-Look* 1-NN classifiers



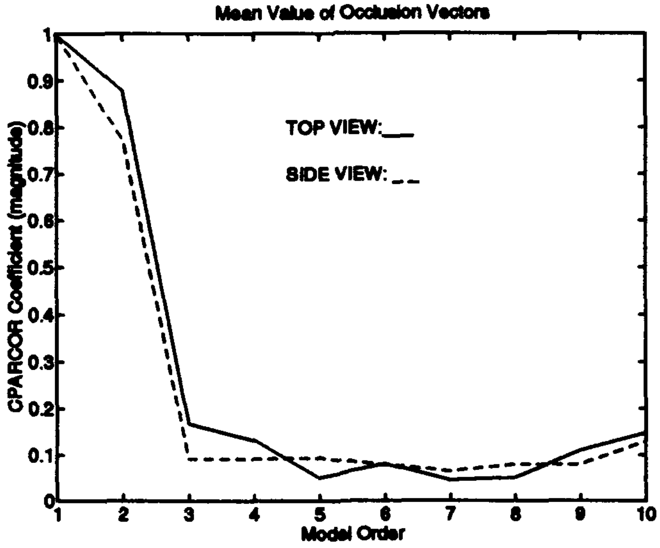
(a)

(b)

Figure 3.11 Verification of previously reported occluded view recognition rates. Test set consisted of occluded M60 Tank views only. Recognition rates for (a) Dissimilar 3 class (M60, BTR, M35) (b) Similar 3 class (M60, M2, T62).



(a)



(b)

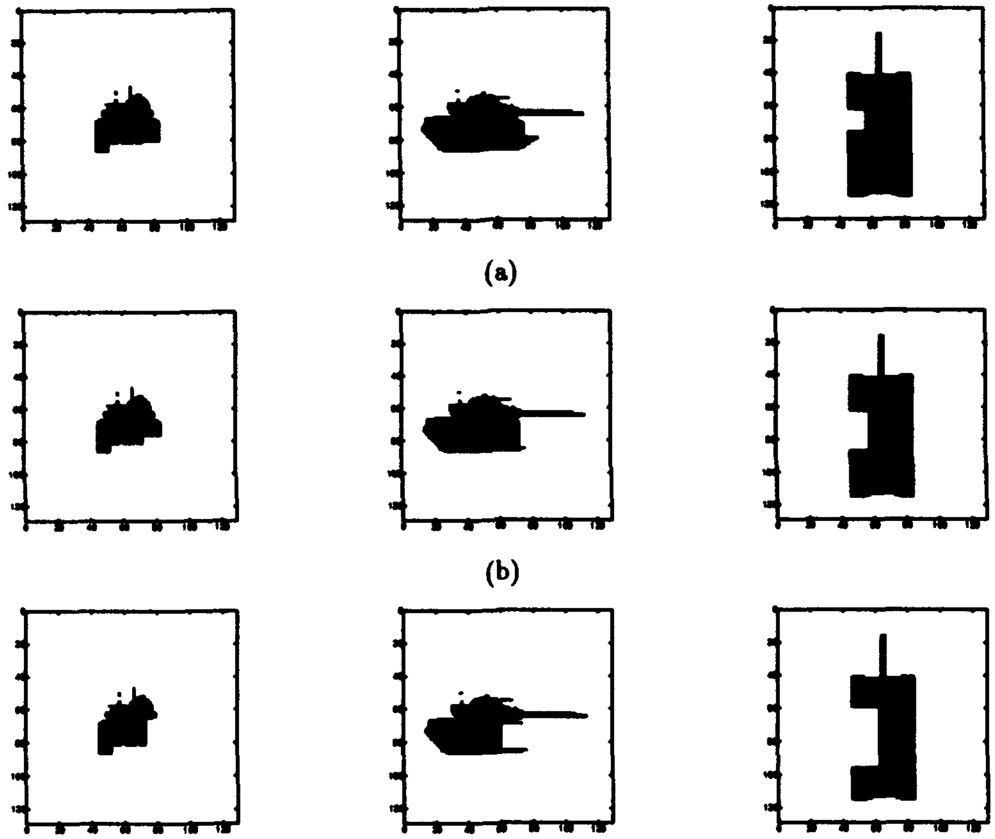
Figure 3.12 CPARCOR coefficient occlusion comparisons. (a) Feature vectors for M60 characteristic views occluded 0, 5, 10, and 20% (b) Mean value of occluded feature vectors.

used to test their respective recognition techniques on unnormalized CPARCOR feature vectors. Similarly, Section 3.4.4 details the procedures of the Hidden Markov Model recognition test.

3.4.3 K-Nearest-Neighbor (KNN) Techniques. The nearest-neighbor rule is a sub-optimal technique that will typically lead to an error greater than the Bayes rate, the minimum possible [4]. For this thesis, two versions of a 1-NN classifier were used for recognition of both single-frame and multiple-frame sequences of ***uncorrupted*** (no noise/occlusion) imagery. Tests were conducted to compare the effects of ***Single-Look*** versus ***Multiple-Look*** recognition on unnormalized CPARCOR feature vectors. A five class problem was attempted, using images from the set of Army ground vehicles shown in Figure 3.2 on Page 3-4. For each class, 50 randomly generated image sequences of lengths 14, 16, 18, and 20 frames each (200 total sequences) were created. For a view area of 0^0 to 180^0 azimuth and 0^0 to 75^0 elevation, sequential frames were allowed to differ (or remain the same) by 5^0 in azimuth or elevation. In this manner, 3400 individual image frames were created for each class (17,000 total frames). A tenth order CPARCOR model, chosen for compatibility with Fourier results, was then created for each image.

For the *Single-Look 1-NN* test, the ***hold-one-out*** method was used to sequentially hold out a single frame of imagery for comparison with the remaining 16,999. The ***hold-one-out*** method provides an upper bound to the Bayes error rate, yielding a ***worst-case*** estimate of the error when generalizing on unseen data[8]. Classification was then based on the class containing the ***single*** best matched frame to the one withheld. Although following the same initial procedures, classification for the *Multiple-Look 1-NN* was based on the class containing the ***majority*** of best matched frames per sequence, ***not*** per frame. The results of these tests, along with a direct comparison with recognition by Fourier features, are discussed in Chapter IV.

3.4.4 Hidden Markov Model (HMM) Testing. To determine if CPARCOR features could accurately represent temporally encoded sequences of images, a ***Hidden Markov Model (HMM)*** classifier was applied. The HMM classifier had previously displayed exceptional recognition rates



(a)

(b)

(c)

Figure 3.13 Example of M60 occluded views for (a) 5% (b) 10% (c) 20%.

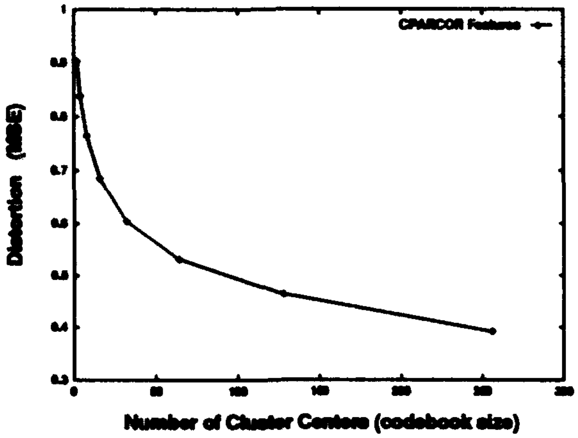


Figure 3.14 Example of Mean Distortion Curves (MDC) for cluster centers (Fourier features shown).

(over 97%) for temporal sequences represented by a low frequency, Fourier magnitude feature set [7]. Thus, a direct 5 class performance comparison was attempted by simply replacing the Fourier coefficients with a set of CPARCOR coefficients. The data set consisted of the five Army ground vehicles previously shown in Figure 3.2.

Based on a viewer centered approach, the HMM used the Linde, Buzo, and Grey (LBG)[11] clustering algorithm to create a vector quantizer whose clusters corresponded to areas (on a viewing sphere) of similar characteristic view[7]. For test purposes, the region of interest was restricted to a 0^0 to 180^0 azimuth, 0^0 to 75^0 elevation portion of the viewing sphere. 592 images per class were generated for this view region. Both the CPARCOR and Fourier coefficients were statistically normalized across all feature vectors to produce zero mean, unit variance features.

A codeword vector quantizer was produced by processing the feature vectors with the LBG algorithm. The result was a *cluster distortion* versus *codebook size* plot as shown (for Fourier feature set) in Figure 3.14. Based on this data, a 64 codeword vector quantizer was chosen for use with the Fourier data. The same size quantizer was applied to the CPARCOR data for consistency in technique comparisons.

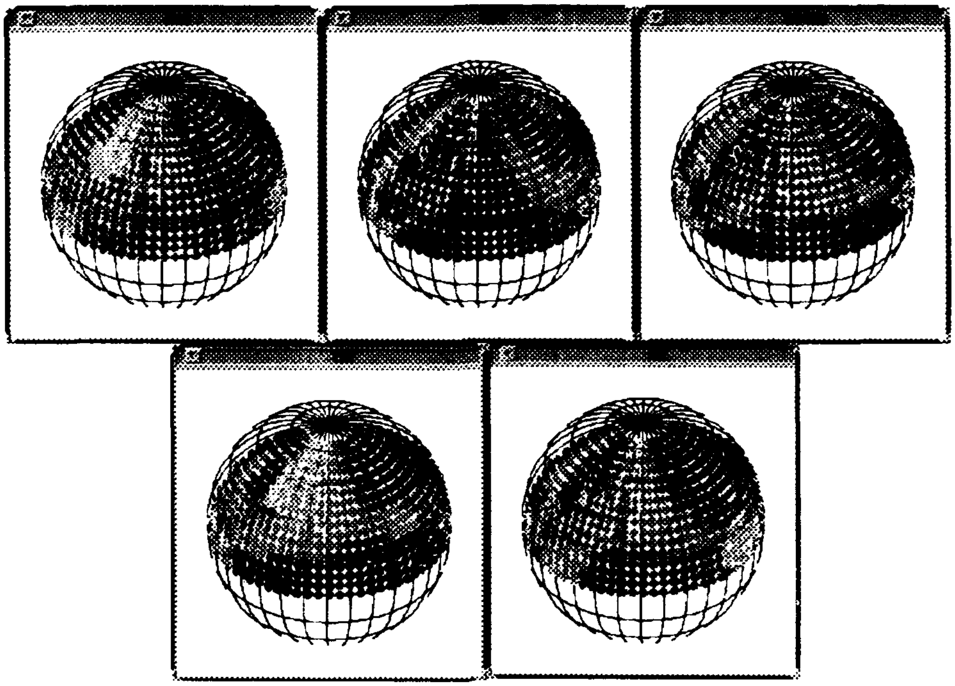


Figure 3.15 Example of sphere mapping (Fourier features shown).

Sphere plots were used to associate the mapping of each viewing position to a particular codebook entry. Basically, each viewing position was associated with a characteristic view. An example of the viewing spheres for the Fourier data (CPARCOR results are shown in Chapter IV) is shown in Figure 3.15. Areas of common shading represent a shared characteristic view or aspect. Recognition ambiguity is seen to exist when more than one object class occupies the same cluster.

Respective class HMMs were trained on the same 200, randomly generated sequences used for the *Single and Multiple-Look K-Nearest-Neighbor (KNN)* tests (Section 3.4.3). Although not expected to have any great impact on the HMM test, it should be noted that 128 x 128 pixel arrays were used for the CPARCOR images, while 256 x 256 arrays were used for the Fourier images. A sphere plot example of five of the randomly generated training sequences is shown in Figure 3.16 on Page 3-29. Results from this direct comparison of CPARCOR and Fourier features are fully discussed in Chapter IV.

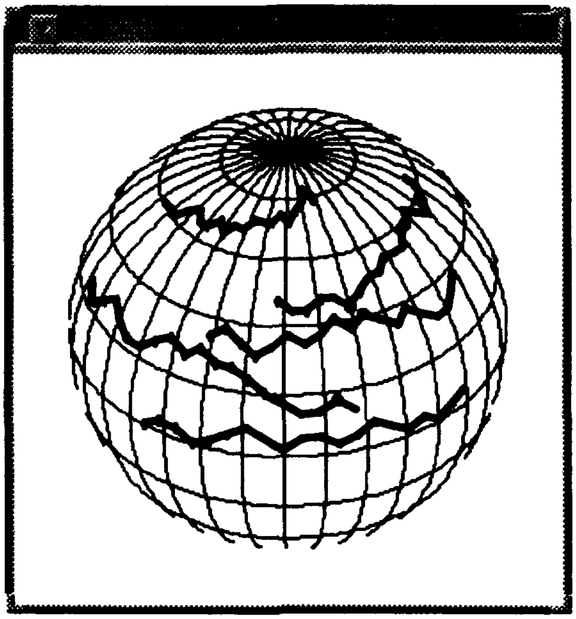


Figure 3.16 Example of sphere trajectories.

3.5 Chapter Summary

The development (Section 3.2), validation (Section 3.3), and application (Section 3.4) of the CPARCOR algorithm has been presented. Along with fundamental principles of the algorithm, the methodology used to verify both the main program and supporting subroutines was described. The chapter concluded with a description of several recognition tests conducted using the CPARCOR algorithm and various classifiers. Chapter IV discusses the results of these tests.

IV. Results and Discussion

Chapter III covered the development, validation, and application of the Complex Partial Correlation (CPARCOR) algorithm. Along with fundamental principles of the algorithm, the methodology used to verify both the main program and supporting subroutines was described. The chapter concluded with a description of several recognition tests conducted using the CPARCOR algorithm and various classifiers.

This chapter provides a discussion of the results for tests described in Chapter III. Presented in the same manner as in Chapter III, the following topics are considered:

- *Template Test Results.* This section provides a discussion of the results for tests conducted with the *Template Matching* algorithm. Although numerous tests were conducted, only representative results are shown (for clarity) in this chapter. Appendix A contains all remaining test results.
- *KNN Test Results.* Single and Multiple-Look 1-NN classifier results are discussed in this section. A direct comparison to recognition by Fourier features is also made.
- *HMM Test Results.* Following a brief review of the methodology attempted for HMM testing, a look at why the CPARCOR model failed to train is presented.

4.1 Template Test Results

After a brief review of the baseline hypotheses used for performance evaluation, this section will discuss the results of tests conducted with the *Template Matching* algorithm. For the sake of clarity, this chapter will only show representative results for the matrix-reduction/unstored-view tests (Section 4.1.1), as well as the occluded view tests (Section 4.1.2). Appendix A contains all remaining results for these tests.

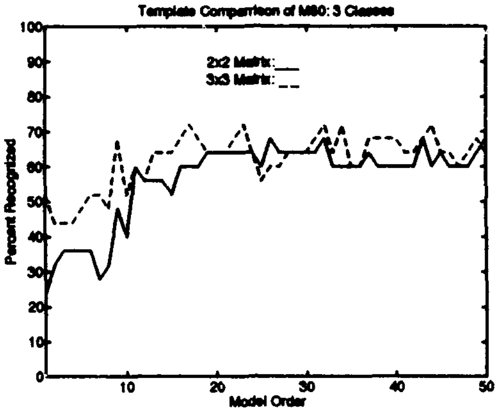
Using a Euclidean distance metric, the relationship between the number of characteristic views and the corresponding recognition rate was explored through application of a template classifier. Template sets were based on **m** x **n** matrices containing unnormalized CPARCOR coefficient vectors. The following baseline hypotheses, originally presented in Chapter III, were used to evaluate template recognition performance:

- Recognition performance should *increase* with model order until a point of optimal performance is achieved. After this point, no addition benefit will be realized by using higher order models.
- For a given model order, recognition performance should *decrease* as the number of available classes is *increased.*
- Object symmetry should allow for a *reduction* in overall template and matrix size.
- As the matrix size of a given template is *increased* (to include more characteristic views), recognition performance for unstored views should also *increase*. The exception to this rule would be recognition of partially occluded views (see next item).
- Recognition of partially occluded views should *decrease* as the matrix size of a given template is *increased.*
- Recognition performance should *decrease* as the percentage of partial occlusion is *increased* from 5% to 20%.

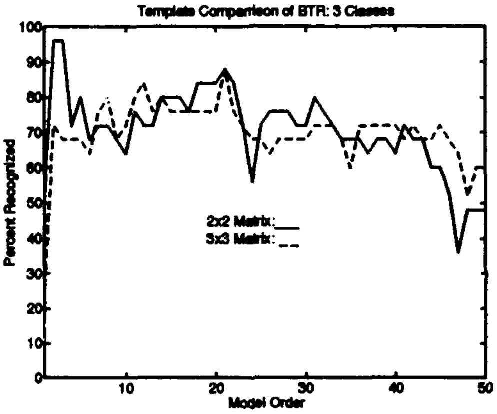
4.1.1 Matrix Reduction and Unstored View Testing. This section examines the results of the matrix reduction and unstored view tests. Motivated by the results of Figure 3.10 (Section 3.4.2.2), various tests were conducted to demonstrate that object symmetry should allow for a *reduction* in overall template and matrix size. For example, the 360^0 View Template was expected to provide almost identical recognition rates as the 90^0 View Template for views in the 90^0 azimuth/elevation region. Applying the unstored view test set (Table 3.2) to the 360^0, 180^0, and 90^0 View Templates, this hypothesis was verified by directly comparing recognition rates.

To illustrate the feasibility of symmetry-related template reductions, unstored view recognition rates for the 360^0 and the 90^0 view templates were directly compared for a three class problem as shown in Figure 4.1 on Page 4-4. Except for some obvious variation in (c), overall recognition rates were practically identical. Extending this idea to a five class case, Figure 4.2 on Page 4-5 indicates that similar results were again obtained. Notice, though, that rates are significantly lower for the five class case due to the addition of two classes of *tracked* vehicles. Although not shown here (see Appendix A), recognition rates were typically higher when larger matrix sizes were used. Similar results were also found for comparisons between the 360^0 and 180^0 templates as well as between the 180^0 and 90^0 templates. Plots for these results can be found in Appendix A. Thus, assuming object symmetries do exists, this method of comparison clearly validated template reduction as a viable option. Obvious advantages of template reduction include a decrease in the number of characteristic views and the processing time required for recognition.

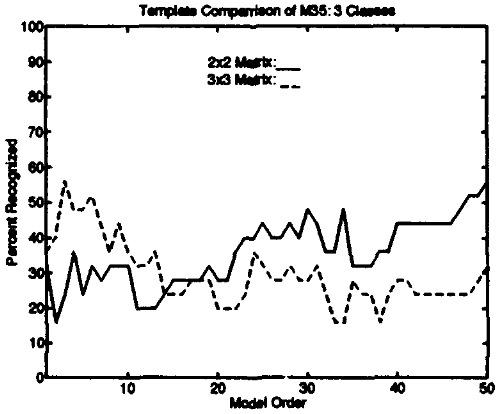
A comparison of 90^0 template recognition rates (using various matrix sizes) for three and five classes is shown in Figures 4.3 and 4.4 (Pages 4-7 and 4-8). Although recognition performance was generally found (for all tests) to *decrease* as the number of available classes was *increased*, recognition performance was not seen to *increase* with model order in a consistent manner. Instead, rates tended to fluctuate randomly with model order. Also, unstored view recognition rates did not necessarily *increase* with an *increase* in matrix size for a given template. In Figure 4.4 (a), for example, consistent recognition (relative to matrix size) at lower orders is seen to suddenly contradict itself after model order 12. At this point, the 2 × 2 matrix displays higher recognition rates than the 5 × 5 or the 10 × 10. These two discrepancies were evident in other tests as well. Thus, the two relevant hypotheses were considered *false*.



(a) M60 Tank

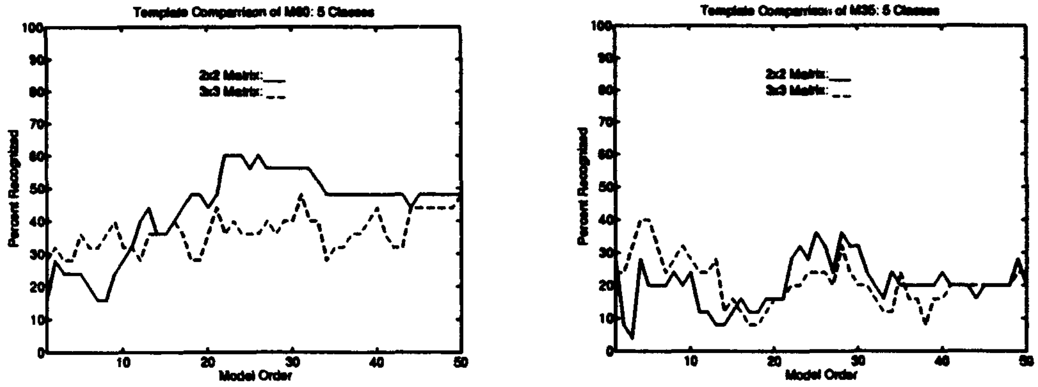


(b) BTR APC

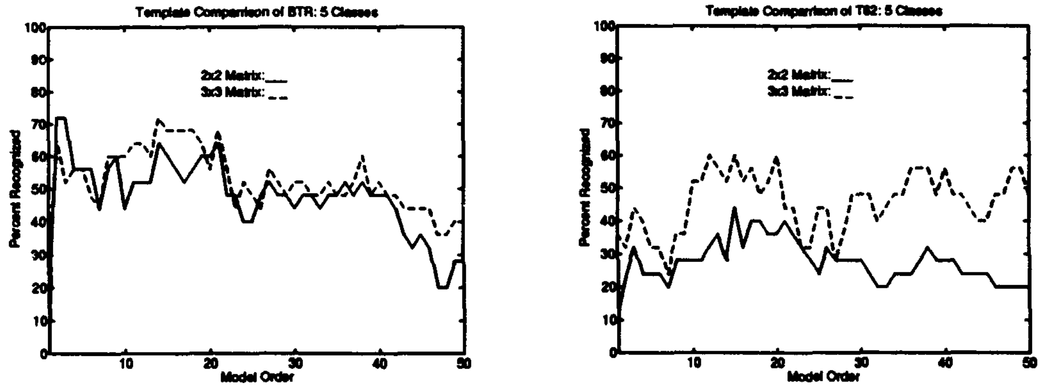


(c) M35 Truck

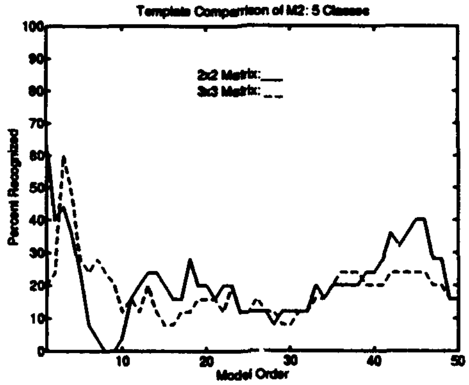
Figure 4.1 Comparison of 90^0 and 360^0 template for unstored view recognition results with 3 classes.



(a) M60 Tank (b) M35 Truck



(c) BTR APC (d) T62 Tank



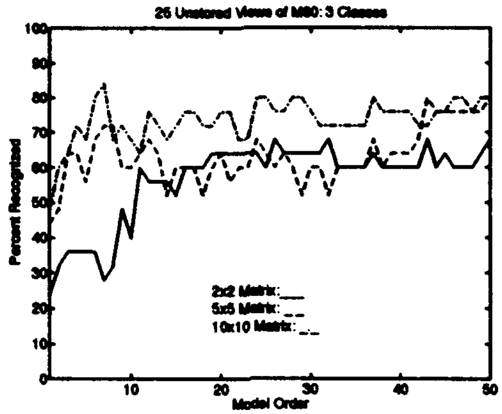
(e) M2 IFV

Figure 4.2 Comparison of 90^0 and 360^0 template for unstored view recognition results with 5 classes.

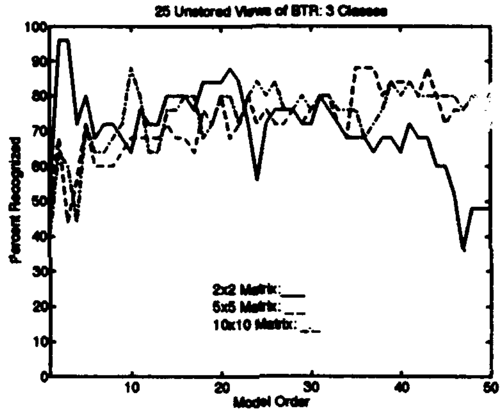
Recall that the data set applied to these tests consisted of ***unstored*** views, which were defined as characteristic views ***not stored*** in the respective template. This means that, depending on the template used, test views could have easily varied by as much as 30^0 (or more/less) in both azimuth and elevation relative to the *stored* views in the template. Significant in a statistical sense, assuming equally likely occurrences, was that the overall recognition trends using these views were typically much higher than that of pure chance. For example, using the 10 x 10 matrix, peak recognition values of 70% and 80% are seen to occur (5 class case), while no rate lower than 24% is noted. As expected, rates were shown even higher for the three class cases where less similarity between classes occurred. Along these lines, higher recognition rates would have also been expected had the original test set been modified to include several *stored*-views. In general, though, the recognition rates from these tests verified that reasonably accurate classification based on a limited number of stored characteristic views was possible.

4.1.2 Occlusion Testing. Following a brief review of the test procedures and baseline hypotheses, this section presents recognition results from tests conducted using partially occluded object-views. For clarity, only representative test results for the M60 Tank shall be presented here. Additional results, including tests conducted on occluded M2 IFV views, are included in Appendix A.

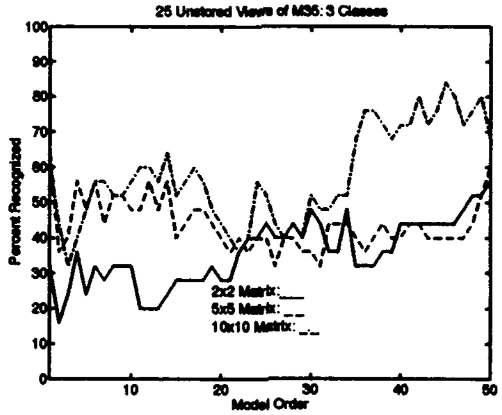
Motivated by the successful results of Figure 3.11 (Section 3.4.2.3), various tests were conducted using *known* levels of occlusion to examine the impact of partial occlusion on the corresponding recognition rate. Separated into three distinct categories (5%, 10%, and 20%), occluded images were made for both the M60 Tank and the M2 IFV. The two classes of *tracked* vehicles were chosen in order to test the robustness of recognition on occluded, but similar objects. A subset of 'hand-made' occluded M60 views was previously shown in Figure 3.13 (Page 3-26). For more *realistic* recognition testing, all *three class* problems consisted of the following *tracked* vehicles: M60 Tank, the M2 IFV, and the T62 Tank. Based on preliminary test results using this three class set



(a) M60 Tank

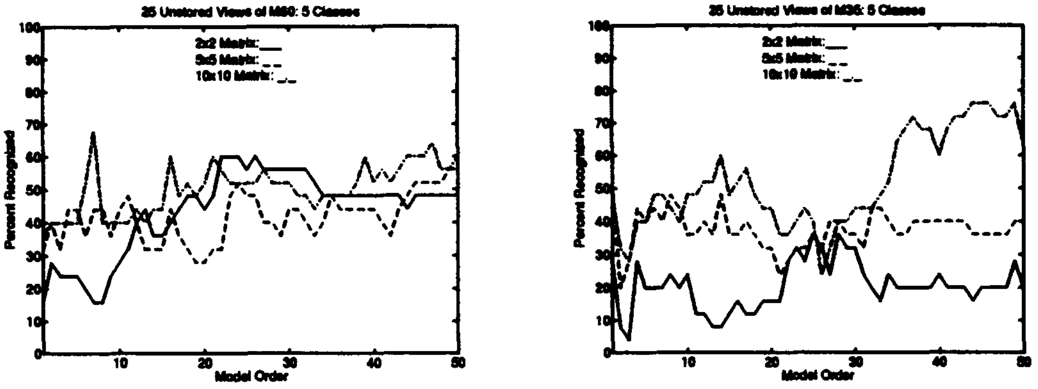


(b) BTR APC



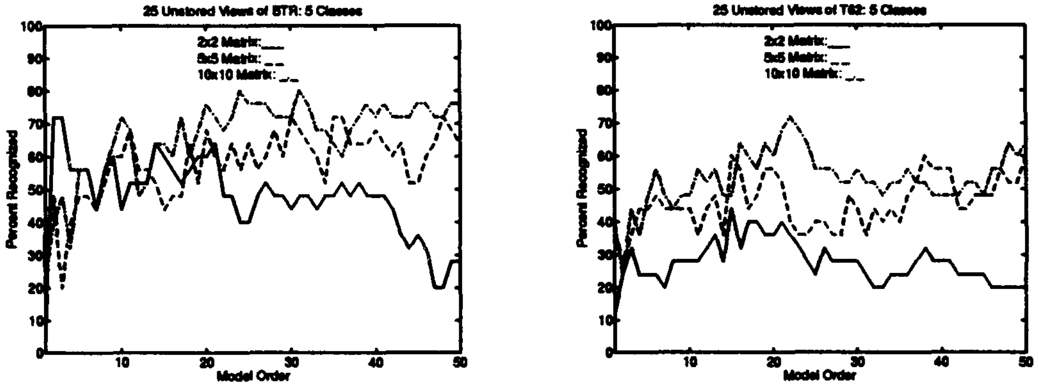
(c) M35 Truck

Figure 4.3 Unstored view recognition results for 3 classes and various 90^0 view templates.



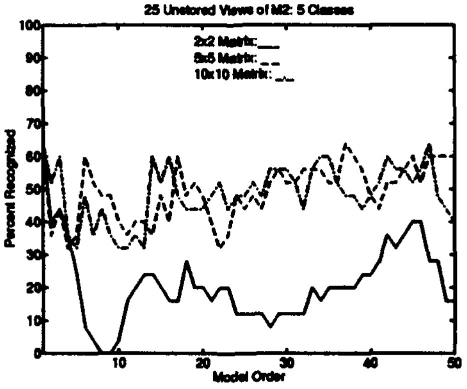
(a) M60 Tank

(b) M35 Truck



(c) BTR APC

(d) T62 Tank



(e) M2 IFV

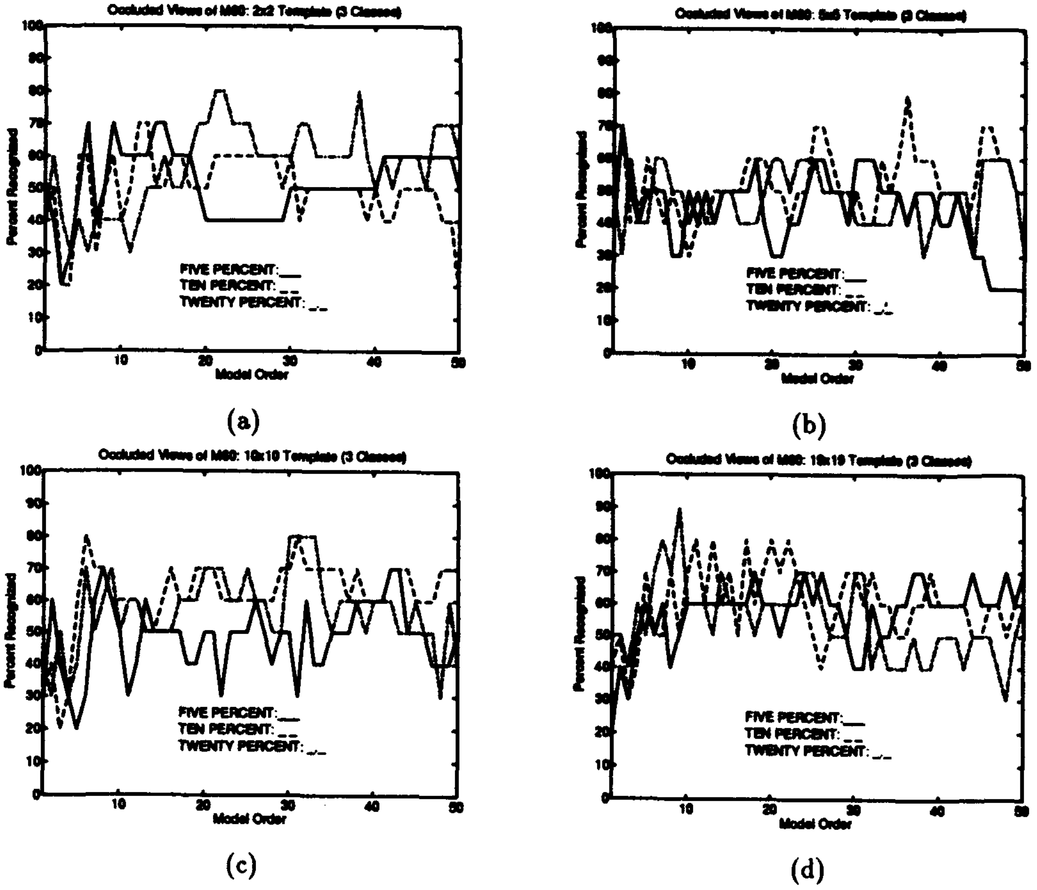
Figure 4.4 Unstored view recognition results for 5 classes and various 90^0 view templates.

(see Appendix A), little variation was expected to exist between three and five class comparisons. For test purposes, 60 occluded images (total) were made for the M60 Tank and M2 IFV from the data set previously shown in Table 3.4 (Page 3-22). Recognition rates for the occluded views were evaluated against the following baseline hypotheses:

- Recognition of partially occluded views should *decrease* as the matrix size of a given template is *increased.*
- Recognition performance should *decrease* as the percentage of partial occlusion is *increased* from 5% to 20%.

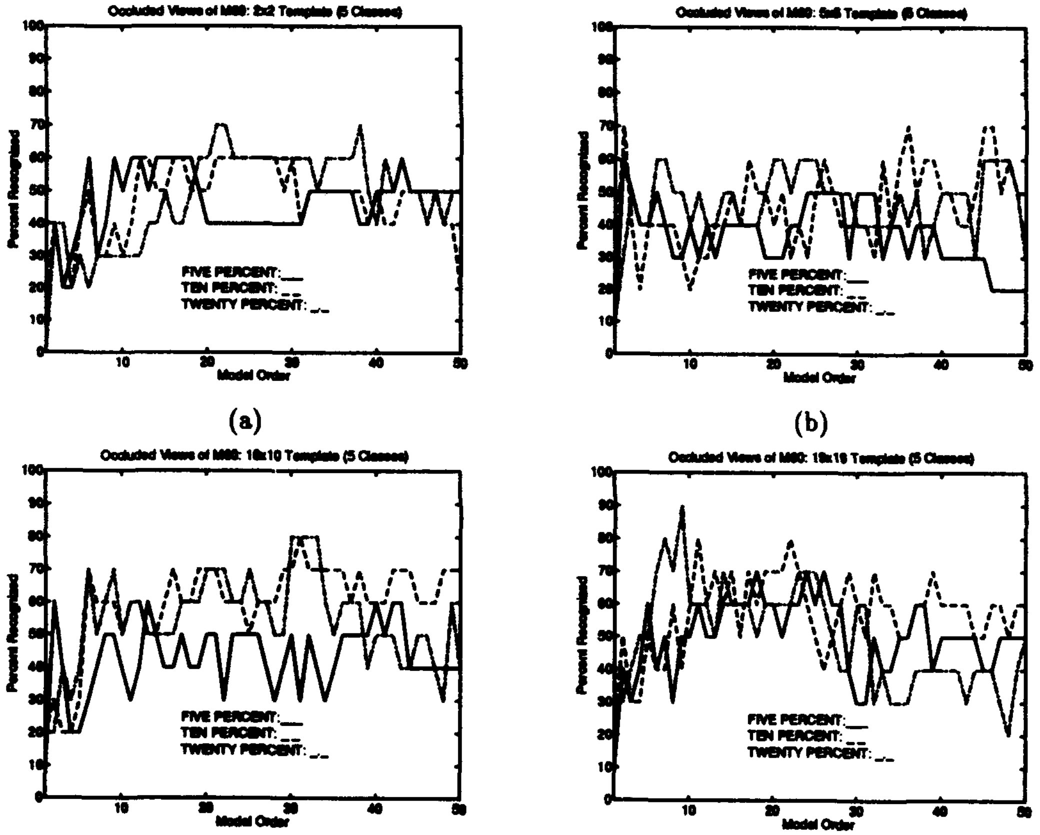
Results for both three and five class tests using the occluded M60 view data set are shown in Figure 4.5 (Page 4-10) and Figure 4.6 (Page 4-11), respectively. The similarity between recognition rates for the two figures was due to use of the stated *tracked* vehicles (three class problem). Although similar results are also shown in Appendix A (Figure A.10), neither hypotheses was shown to be valid. For example, instead of displaying a linear relationship, recognition rates were rather constant no matter what level of occlusion was applied. Also, even though recognition of partially occluded views should have *decreased* as the matrix size was *increased*, peak recognition levels were noted to occur (in all cases) for the 19 × 19 matrix. Thus, both of the original hypotheses were shown to be *false*. In spite of this, however, the consistently high recognition rates noted for all tests demonstrated that CPARCOR features were capable of reasonably accurate classification.

Although not shown until Section 4.3 (Figure 4.9), such unexpected results were found to be caused by an unusual mapping of the coefficients in the feature space. An *overcrowding* had occurred in which feature vectors were basically right on top of one another. Keep in mind that classification was being performed using a Euclidean distance metric. As such, recognition was based on the closest *overall* match within the template, *not* necessarily the closest matching view of similar azimuth and elevation. Thus, it was quite possible that an M60 view occluded 20% could be a better match for the M60 class than the same view occluded only 5%.



(a) (b) (c) (d)

Figure 4.5 Three class recognition rates for occluded M60 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix.



(a) (b)

(c) (d)

Figure 4.6 Five class recognition rates for occluded M60 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix.

4.1.3 Discussion of Template Test Results. This section provides a summary of the results for tests conducted with the *Template Matching* algorithm. Following a discussion of the template reduction and unstored view tests, results from the occlusion tests are considered.

Results from the template reduction and unstored view tests clearly verified that object symmetry allowed for a *reduction* in overall template and matrix size. This is significant in that a decreased in the number of characteristic views and the processing time required for recognition can be realized. Also validated was the hypothesis that, for a given model order, recognition performance should *decrease* as the number of available classes was *increased*. This generally accepted result is based on the idea that classifiers have a greater chance of confusing objects as the number of classes is increased. Unexpected were the recognition rates for increased model order and matrix size. As a result, the CPARCOR coefficients were shown to contradict two of the initial hypotheses. Although relatively high recognition rates were noted (given *unstored view* test set) an optimal CPARCOR model order and corresponding template size could not be determined strictly based on these test results. In a statistical sense, though, overall recognition trends were shown to be much higher than that of pure chance. Peak recognition values of 70% and 80% were noted even for the five class case. Considering that the test set consisted of *unstored* views, the recognition rates clearly verified that reasonably accurate classification based on a limited number of stored characteristic views was possible.

Occluded view testing revealed some rather unexpected results. In fact, both of the original hypotheses used to evaluate the tests were shown to be *false*. For example, recognition rates were rather constant no matter what level of occlusion was applied, and recognition of partially occluded views did not *decrease* in a consistent manner as the matrix size was *increased*. In addition to this, an unusual feature-space mapping was noted in which feature vectors were basically right on top of one another. In spite of the seemingly negative results for this series of tests, overall recognition rates were relatively high. Although an optimal CPARCOR model order and corresponding template size could not be determined, recognition rates well over 50% were displayed for *all* tests considered. Noting that the test sets consisted of three rather similar *tracked* vehicles, the CPARCOR features were considered to exhibit reasonably accurate classification capabilities. Finally, it is interesting to note that previous research *always* combined all levels of occlusion together for test purposes. This test was one of the first to display the effect of individual *known* levels of occlusion.

The following sections discuss results from the two additional classifier tests. Section 4.2 presents the results of the K-Nearest-Neighbor tests, while Section 4.3 provides the results for Hidden Markov Model testing.

4.2 K-Nearest-Neighbor (KNN) Test Results

Following a brief review of the methodology used for K-Nearest-Neighbor testing, results for both the Single-Look and Multiple-Look 1-NN classifiers are discussed. A direct comparison to recognition by Fourier features is also made.

Two versions of a 1-NN technique were used for recognition of both single and sequential frames of *uncorrupted* (no noise/occlusion) imagery. Tests were conducted to compare the effects of *Single-Look* versus *Multiple-Look* recognition on unnormalized CPARCOR features. A five class problem was attempted, using images from the set of Army ground vehicles previously shown in Figure 3.2 (Page 3-4). For each class, 50 randomly generated image sequences of lengths 14, 16, 18, and 20 frames each (200 total sequences) were created. For the view area of 0^0 to 180^0 azimuth and 0^0 to 75^0 elevation, sequential frames were allowed to differ (or remain the same) by only 5^0 in azimuth or elevation. In this manner, a total of 3400 individual image frames were created for each class (17,0[illegible]tal frames). For compatibility, an unnormalized, tenth order CPARCOR model was transformed (*real* and *imaginary* parts separated) into a twenty dimensional feature vector to represent each image.

Table 4.1 Single-Look test results for CPARCOR features.

Input \ Classify	M60	M35	BTR	T62	M2	Percent Correct
M60 Tank	2149	165	44	921	121	63
M35 Truck	54	2332	626	65	323	69
BTR APC	25	669	2440	33	233	72
T62 Tank	790	107	58	2305	140	68
M2 Tank	60	467	260	84	2529	74

Table 4.2 Single-Look test results for Fourier features.

Input \ Classify	M60	M35	BTR	T62	M2	Percent Correct
M60 Tank	3398	0	0	1	1	99.9
M35 Truck	0	3391	9	0	0	99.7
BTR APC	0	9	3391	0	0	99.7
T62 Tank	2	0	1	3396	1	99.9
M2 Tank	0	0	0	5	3395	99.8

4.2.1 Single-Look 1-NN Testing. Single-Look 1-NN testing used the ***hold-one-out*** method to sequentially hold out a single frame of imagery for comparison with the remaining 16,999. Classification was then based on the class containing the single *best* matched frame to the one withheld. Results of this test are shown in Table 4.1. Although achieving an overall accuracy of 69%, recognition was generally considered poor since no image corruption was performed. For comparison, results based on the use of Fourier features are provided in Table 4.2. Results are notably higher, with an average recognition rate of 99.8%.

Table 4.3 Multiple-Look test results for CPARCOR features.

Input \ Classify	M60	M35	BTR	T62	M2	Percent Correct
M60 Tank	181	0	0	12	7	90
M35 Truck	0	198	0	0	2	99
BTR APC	0	9	191	0	0	95
T62 Tank	7	0	0	188	5	94
M2 Tank	0	0	0	0	200	100

Table 4.4 Multiple-Look test results for Fourier features.

Input \ *Classify*	M60	M35	BTR	T62	M2	Percent Correct
M60 Tank	200	0	0	0	0	100
M35 Truck	0	200	0	0	0	100
BTR APC	0	0	200	0	0	100
T62 Tank	0	0	0	200	0	100
M2 Tank	0	0	0	0	200	100

4.2.2 Multiple-Look 1-NN Testing. Following the same initial procedures, classification for the Multiple-Look 1-NN was based on the class containing the *majority* of best matched frames per sequence, *not* per frame. Results of this test using CPARCOR coefficients, shown in Table 4.3, are significantly higher than for the respective 1-NN test. Although not as clearly demonstrated by the Fourier features (Table 4.4), the dramatic improvement is easily noted for the CPARCOR features. An overall accuracy of 96% was achieved, indicating a decisive advantage of sequential classification over the previous single-frame techniques of Template Matching and Single-Look 1-NN. Statistically speaking, however, the results are not all that surprising. Assuming independence of class views, recognition rates are generally expected to increase as more frames are considered for classification. Also note that for this particular classifier, sequence *order* was irrelevant. In other words, classification results would have been the same no matter how the individual frames were arranged in the sequences.

4.2.3 Discussion of KNN Test Results. The advantage of classification by a ***multiple-look*** technique over the traditional ***single-look*** method was clearly demonstrated by the KNN tests. Based on a direct comparison between CPARCOR and Fourier features, the multiple-look technique was noted to provide higher classification rates for each case. Although recognition rates by Fourier features were notably higher than those of the CPARCOR features, the actual application should be considered before deciding on which method to use. For example, the PSRI properties of the CPARCOR features are inherent, while additional signal processing is required in order to achieve similar results with Fourier features. Also, the CPARCOR features may be more readily capable of dealing with partial occlusions than Fourier features. For either case, though, it has been shown that classification should be performed based on multiple-look techniques.

The next section provides a discussion of the Hidden Markov Model test. A direct comparison of recognition by CPARCOR and Fourier features was attempted.

4.3 Hidden Markov Model (HMM) Test Results

Although successful results were obtained for the previous classifier tests, the CPARCOR features failed to provide geographically oriented sphere clustering properties. As such, a direct ***apples*** to ***apples*** comparison of recognition by CPARCOR and Fourier features could ***not*** be performed. Thus, following a brief review of the methodology attempted for HMM testing, a look at why the CPARCOR model failed to train is presented.

A Hidden Markov Model (HMM) classifier was applied to determine if CPARCOR features could accurately represent temporally encoded sequences of images. The HMM classifier had previously displayed exceptional recognition rates (over 97%) for temporally encoded sequences represented by a low frequency, Fourier magnitude feature set [7]. Thus, a direct five class performance comparison was attempted by simply replacing the Fourier coefficients with a set of CPARCOR features. The data set consisted of the five Army ground vehicles previously shown in Figure 3.2 on Page 3-4.

Based on a viewer centered approach, the HMM used the Linde, Buzo, and Grey (LBG)[11] clustering algorithm to create a vector quantizer whose clusters corresponded to areas (on a viewing sphere) of similar characteristic view[7]. For test purposes, the region of interest was restricted to a 0^0 to 180^0 azimuth, 0^0 to 75^0 elevation portion of the viewing sphere. A total of 592 images per class were generated for this view region. Both the CPARCOR and Fourier coefficients were statistically normalized across all feature vectors to produce zero mean, unit variance features.

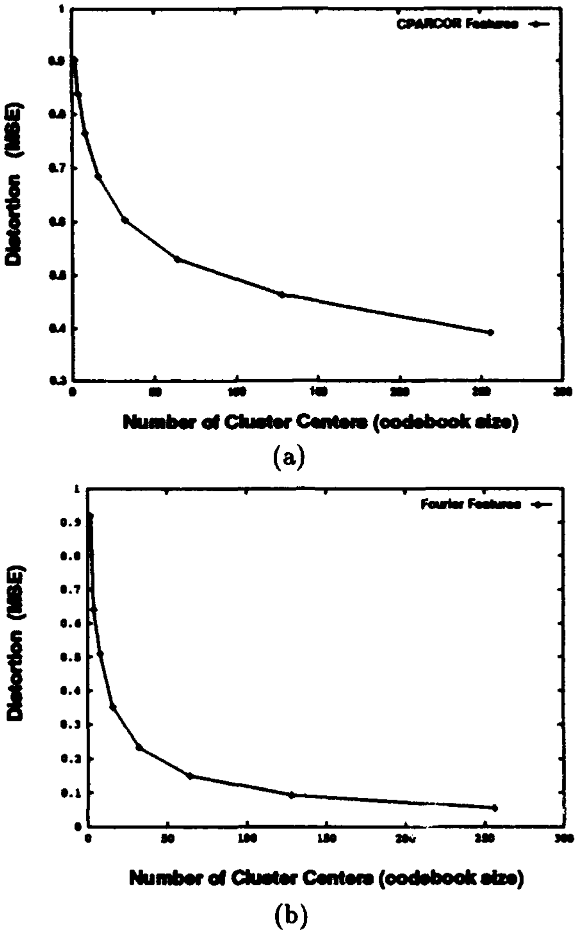


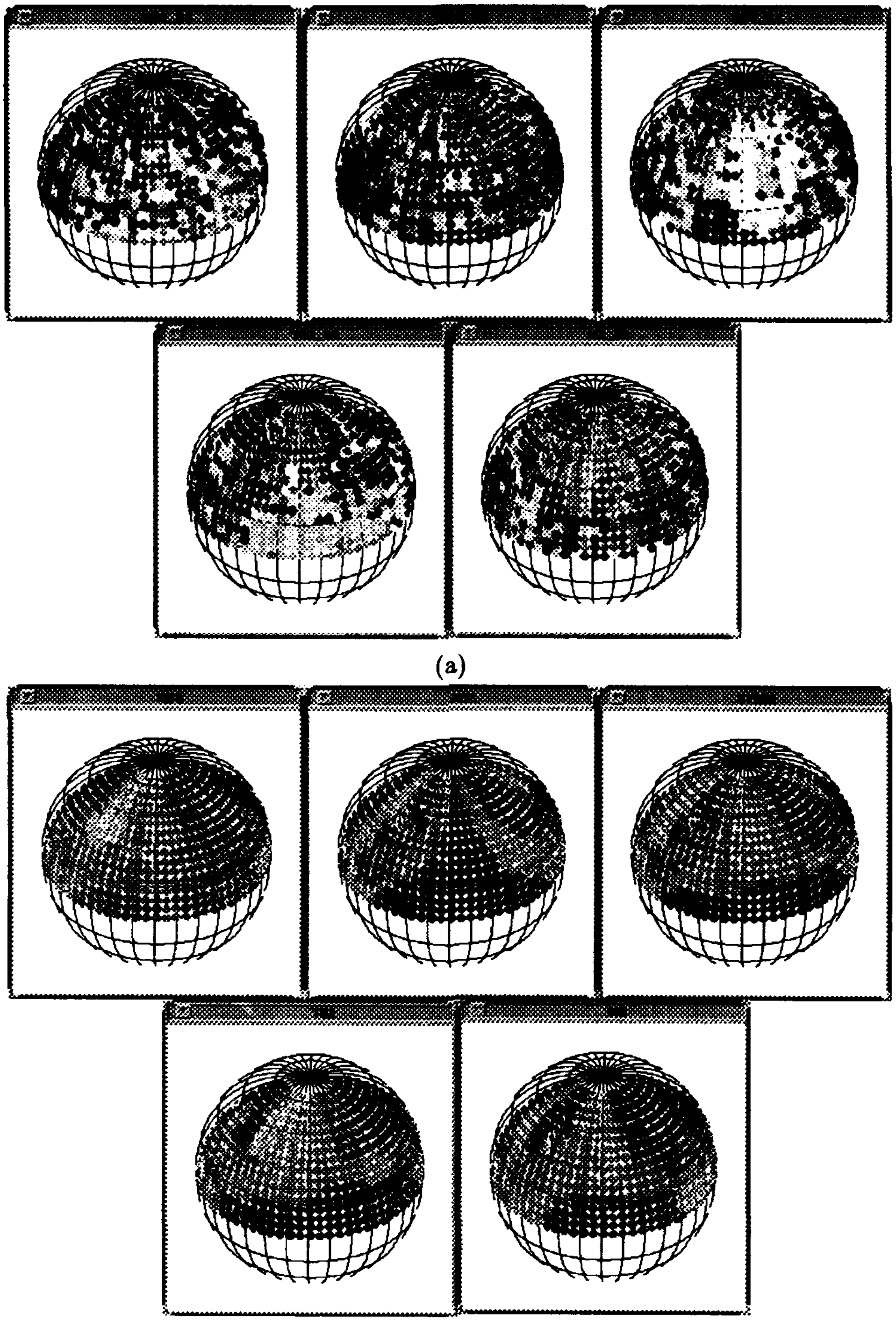
Figure 4.7 Comparison of Mean Distortion Curves (MDC) for cluster centers (a) CPARCOR features (b) Fourier features.

A codeword vector quantizer was produced by processing the feature vectors with the LBG algorithm. The result was a *cluster distortion* versus *codebook size* plot as shown in Figure 4.7 for both the CPARCOR and the Fourier feature sets. Based on this data and a desired *apples* to *apples* performance comparison, a 64 codeword vector quantizer was chosen.

Sphere plots were used to associate the mapping of each viewing position to a particular codebook entry. Basically, each viewing position becomes associated with a characteristic view. Viewing spheres for both the CPARCOR and the Fourier data are shown in Figure 4.8 on Page 4-19. Areas of common shading represent a shared characteristic view or *aspect*. Clearly the spheres in (b) display more highly defined clusters than those in (a). Based on this data and the idea that recognition ambiguity occurs when more than one object class occupies the same cluster, classification results were forecast to be much lower for the CPARCOR features.

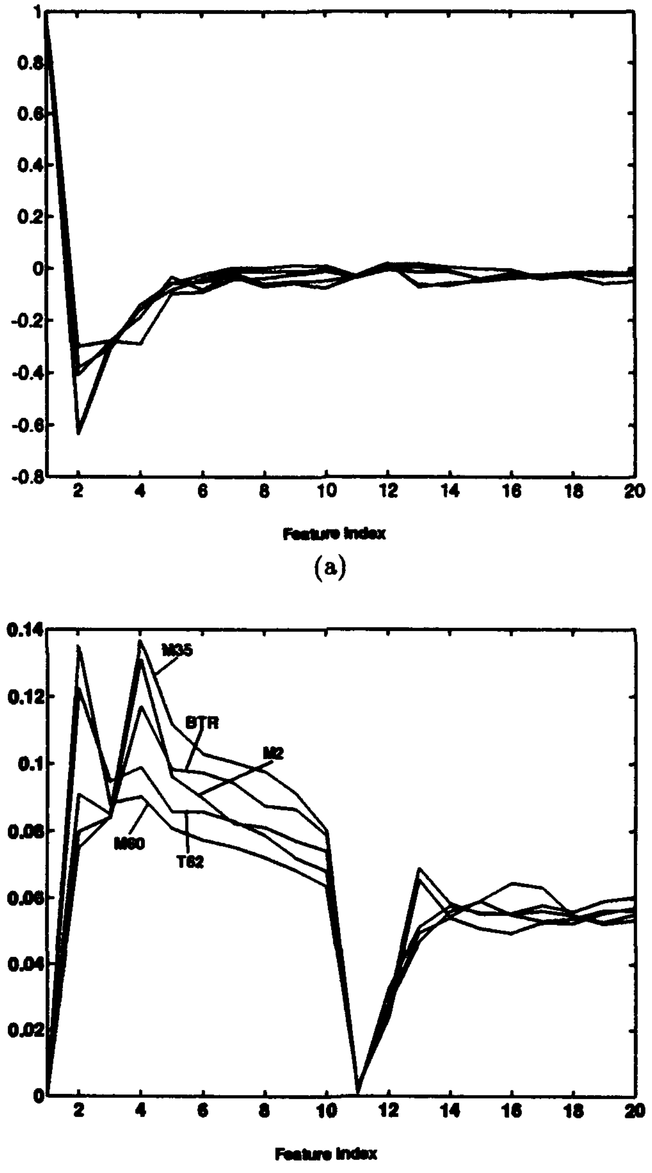
Training of respective class HMMs was attempted on the same 200, randomly generated sequences of 128 x 128 images previously used for the Single and Multiple-Look K-Nearest-Neighbor (KNN) tests (Section 4.2). However, due to the inconsistent sphere clustering (noted earlier), three of the five classes refused to train, and the two that did were of questionable worth. To see if the problem could be corrected, training attempts were made with the use of both more and less codebook vectors, as well as unnormalized data. Although the HMMs eventually trained on a set of unnormalized, 16 codebook vector data, the test was dismissed as being too ambiguous.

In an attempt to determine why the CPARCOR features failed to provide geographically oriented sphere clustering properties, the mean and standard deviation of the feature vectors for each class (unnormalized data) was examined. As shown in Figure 4.9 (a) on Page 4-20, little spatial separation was seen to exist between each of the five classes. Also, Figure 4.9 (b) illustrates that only a slight amount of variance existed between each class (note axis scale). Extremely close to unity for all five classes, first order coefficients displayed very little variation. Recall that the feature vector of complex coefficients was separated into both *real* and *imaginary* parts for this plot. Hence, the variance was shown close to *zero* at indices of one (real part) and eleven (imaginary part). Due to this *crowding* of the CPARCOR features, clearly defined clustering was not possible. Thus, for the model order considered, classification by HMM was not possible.


(a)

(b)

Figure 4.8 Comparison of 5 class sphere mapping (a) CPARCOR features (b) Fourier features.



(a)

(b)

Figure 4.9 Comparison of 5 class (a) Mean and (b) Standard Deviation of CPARCOR feature vectors (unnormalized data).

4.4 *Chapter Summary*

The CPARCOR features were evaluated using several methods of classification. Template Classifiers were used to examine the relative recognition rates for unstored and partially occluded views. Results indicated that template reductions based on object symmetry were possible. Specifically, the 360^0 view template was reduced to a 90^0 view template without any significant impact on recognition rate. Feature space *crowding* was noted as the cause of unusual recognition rates for occluded views. Although general trends were noted, an optimal matrix size and coefficient order could not be determined.

CPARCOR features were also classified by both Single and Multiple-Look KNN classifiers. Using a 1-NN prototype and a twenty dimensional model, recognition rates of 69% and 96% were achieved for respective classifiers. Based on a direct comparison to Fourier feature results, the advantage of classification by the multiple-look technique over the traditional single-look method was clearly shown. Use of the either the CPARCOR or Fourier features was noted to be application dependent.

The inability of the CPARCOR features to exhibit consistent clustering properties resulted in a failed attempt to classify by Hidden Markov Model. A plot of the overall mean and standard deviation for the CPARCOR features reveled that little spatial separation existed between the five classes considered. Having failed at attempts to establish a temporal relationship among the CPARCOR features, the Fourier features were proven more optimal for HMM classification.

V. Conclusions and Recommendations

Expanding previous test methodologies to include ***nonplanar*** views, this thesis effort sought to encode and evaluate the Complex Partial Correlation (CPARCOR) algorithm on three types of classifiers. Initially highlighting conclusions from the classifier tests, this chapter also provides several recommendations for future research.

5.1 Conclusions

The CPARCOR method of feature extraction was applied to nonplanar 2-D views of 3-D objects. Recognition performance was evaluated based on the following object classification techniques:

- *Template Matching Algorithm*: Using a Euclidean distance metric, the impact of unstored and partially occluded views on single-frame recognition performance was examined.
- *K-Nearest-Neighbor (KNN) Technique*: Recognition of both single and multiple frames of imagery was performed using the ***hold-one-out*** method and Single/Multiple-Look 1-NN classifiers. A direct comparison with recognition by Fourier features was made.
- *Hidden Markov Model (HMM)*: Temporal sequence recognition by CPARCOR features was attempted for comparison with recognition by Fourier features.

Results indicate that the CPARCOR model parameters provide useful shape-features for recognition of ***out-of-plane*** rotations. Displaying exceptional PSRI properties, the features were shown capable of classification by simple nonadaptive recognition schemes. Relatively successful results were obtained for a variety of tests. The advantage of classification by a ***multiple-look*** technique over the traditional ***single-look*** method was clearly demonstrated. Although general trends were noted, optimal model order and selection of CPARCOR versus Fourier features were considered application dependent.

5.1.1 Template Matching Algorithm Results. Results from the template reduction and unstored view tests clearly verified that object symmetry allowed for a reduction in overall template and matrix size. Obvious benefits of this result are a *decrease* in required storage capacity and an *increase* in recognition processing time. Also validated was the hypothesis that, for a given model order, recognition performance should *decrease* as the number of available classes was *increased*. Although previously reported high test results for *planar* views were verified, an optimal CPARCOR model order and corresponding template size could not be determined based on results for the *nonplanar* views. Unstored and occluded view test results revealed a *crowding* of the feature space that resulted in several classification inconsistencies. In general, though, recognition rates clearly verified that reasonably accurate classification based on a limited number of stored characteristic views was possible. Results also indicated that reasonably high recognition rates could be achieved by using less similar classes, as well as fewer classes.

5.1.2 K-Nearest-Neighbor (KNN) Technique Results. The advantage of classification by a *multiple-look* technique over the traditional *single-look* method was clearly demonstrated by the KNN tests. As discussed in Section 4.2, a dramatic increase in the recognition rate (69% single versus 96% multiple-look) was noted for the CPARCOR coefficients. Based on a direct comparison between CPARCOR and Fourier features, the multiple-look technique was noted to provide higher classification rates for each case. Although recognition rates by Fourier features were notably higher than those of the CPARCOR features, the actual application should be considered before deciding on which method to use. For example, the PSRI properties of the CPARCOR features are inherent, while additional signal processing is required in order to achieve similar results with Fourier features. Also, the CPARCOR features may be more readily capable of dealing with partial occlusions than Fourier features. For either case, though, it has been shown that classification should be performed based on multiple-look techniques.

5.1.3 Hidden Markov Model (HMM) Test Results. Although the CPARCOR features display significantly better PSRI properties than Fourier features, the CPARCOR coefficients failed to train on the HMM classifier. Although only a tenth order CPARCOR model was applied (for Fourier comparison), similar poor results were expected for other model orders as well due to feature space *crowding*. Based on failed attempts to establish a temporal relationship among the CPARCOR features, the Fourier features were proven more optimal for HMM classification.

5.2 Recommendations

The rather promising results of multiple-look 1-NN tests indicate that a simple, *pseudo-temporal*, template matching classifier could easily be developed. Based on the moderately high recognition rates from the *Template Matching* tests, an experimentally determined optimal model order and matrix size could be combined with a multiple-look classifier. The classifier would be *pseudo-temporal* in the sense that *sequence order* would not be important. Obvious advantages to this approach are high recognition rates and inherent design simplicity.

5.3 Future Research

Although extensive autoregressive model research could be conducted in a variety of areas, of particular interest are the following:

- Image reconstruction from the CPARCOR coefficients.
- Impact on recognition rates from the application of a boundary smoothing technique.
- Characteristic View selection by clustering algorithm.

Appendix A. Supplemental Test Results

To supplement results contained in Chapters III and IV, this appendix contains additional results obtained from the *Template Matching* algorithm tests. Intended only as additional reference material, detailed explanations are not provided. The additional figures are presented in the following order:

1. *Individual Order Test.* Conducted to determine if certain CPARCOR coefficients provide more recognition information than others. Using only *single-orders* of CPARCOR coefficients, instead of the usual combined effect, recognition was attempted using the data set of *unstored* views from Table 3.2. Figure A.1 (a) (Page A-2) clearly illustrates that no *single* order provides consistently higher recognition rates than any other. For clarity, the *mean* recognition rate at each order is shown in (b). Although some fluctuation is evident, no *one* order is noted to have a more optimal recognition performance than the others. Thus, the combined effect of the CPARCOR coefficients was applied to all higher-order tests.
2. The 360^0 View Template consisted of a 3 × 3 matrix (9 characteristic views) for each class. Results of both a three and a five class test are shown in Figure A.2 (a) and (b) (Page A-3), respectively.
3. A comparison of recognition results for both 3 and 5 Classes using the $360^0(3 \times 3)$ template is shown in Figure A.3 (Page A-4).
4. Results for a 3 class comparison of the 180^0 and 360^0 view templates are shown in Figure A.4 (Page A-5). In general, no significant difference in recognition rates were noted.
5. Results of tests performed on the 180^0 view template to compare recognition rates for various matrix sizes are shown in Figure A.5 (Page A-6).
6. Direct comparisons of the 90^0 and 180^0 templates for recognition of a three class problem (M60, BTR, M35) are shown in Figures A.6 to A.8 (Pages A-7 to A-9).
7. Figure A.9 (Page A-10) shows the results of a three *dissimilar* class (M60, BTR, M35) and five class occlusion test using various matrix sizes.
8. Figure A.10 (Page A-11) shows the results of a three *similar* class (M60, T62, M2) and five class occlusion test using various matrix sizes.
9. A comparison of both 3 and 5 Class recognition of occluded M60 characteristic views using 90^0 and 360^0 templates is shown in Figure A.11 on Page A-12. In general, no significant difference in recognition rates were noted.
10. Figure A.12 (Page A-13) and Figure A.13 (Page A-14) show results for occluded M2 view tests.
11. A comparison of the *mean* recognition rates (based on matrix size) for both the M60 and the M2 is shown in Figure A.14 (Page A-15).

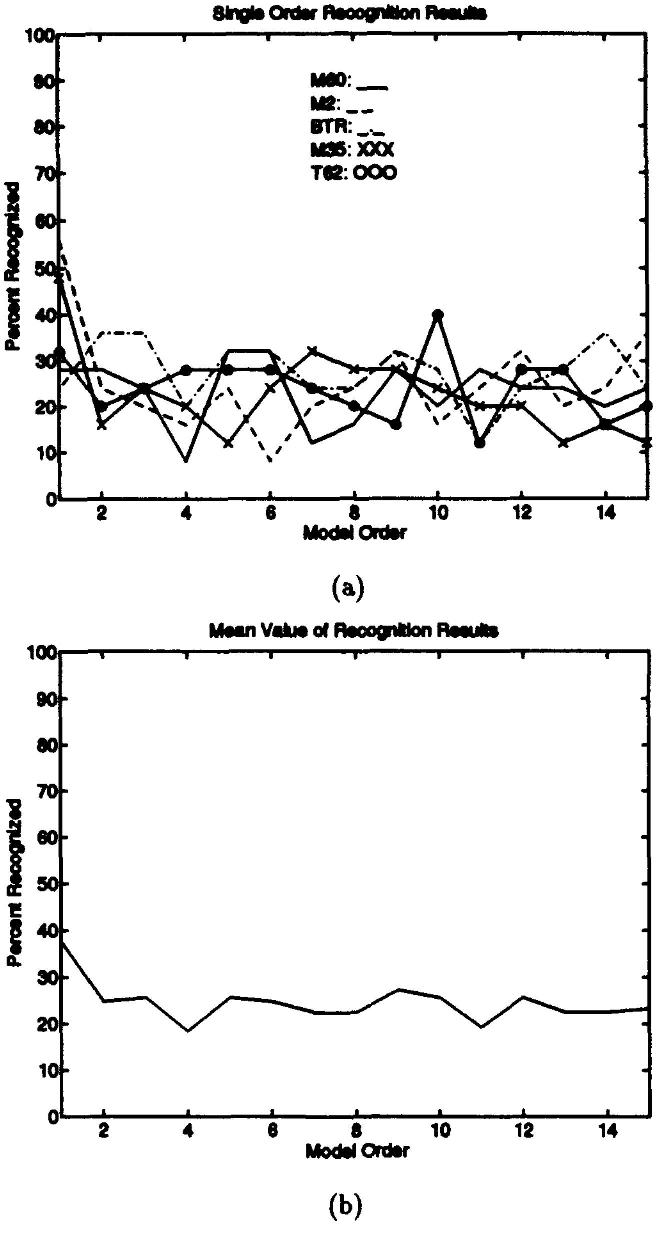
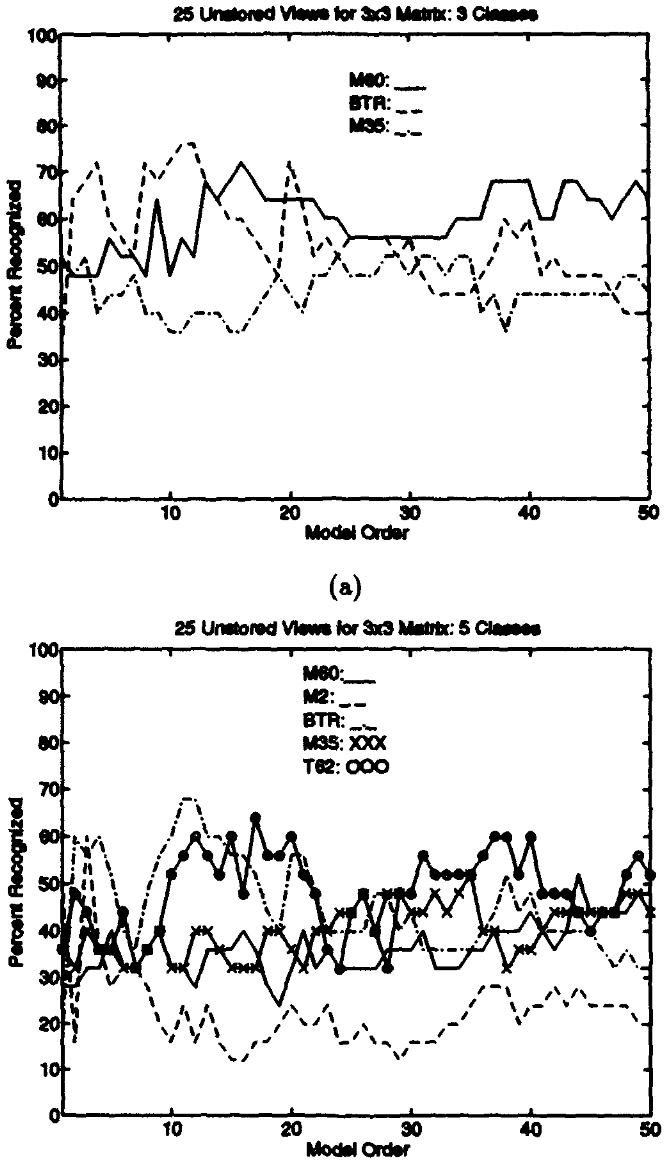


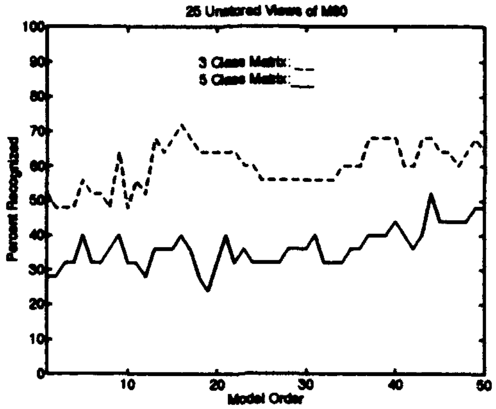
Figure A.1 Single-order CPARCOR coefficient recognition analysis: (a) Recognition results for all five classes (b) Mean value of recognition results (all five classes).



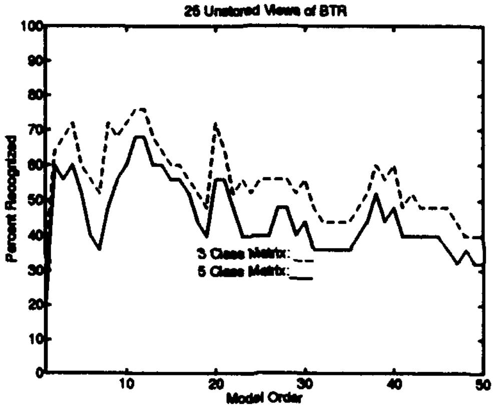
(a)

(b)

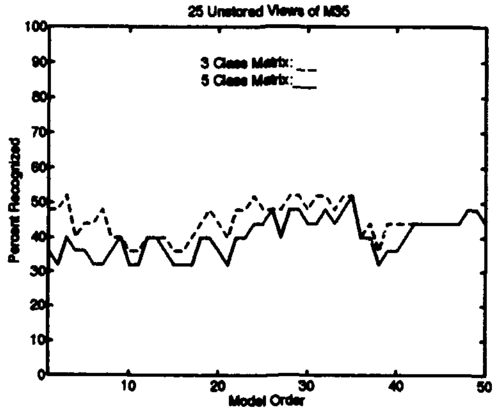
Figure A.2 Unstored view comparison using $360^0(3 \times 3)$ for (a) 3 Classes and (b) 5 Classes.



(a) M60 Tank

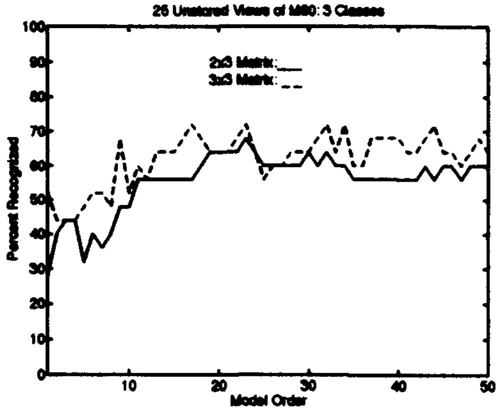


(b) BTR APC

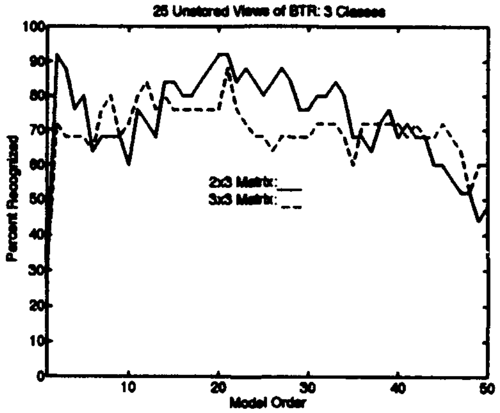


(c) M35 Truck

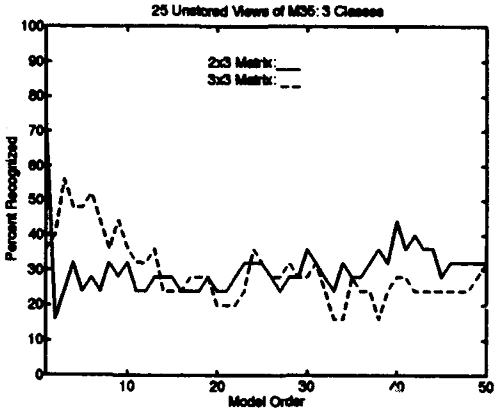
Figure A.3 Comparison of recognition results for 3 and 5 Class, $360^0(3 \times 3)$ template.



(a) M60 Tank



(b) BTR APC



(c) M35 Truck

Figure A.4 Comparison of $180^0(2 \times 3)$ and $360^0(3 \times 3)$ template for unstored view recognition results with 3 classes.

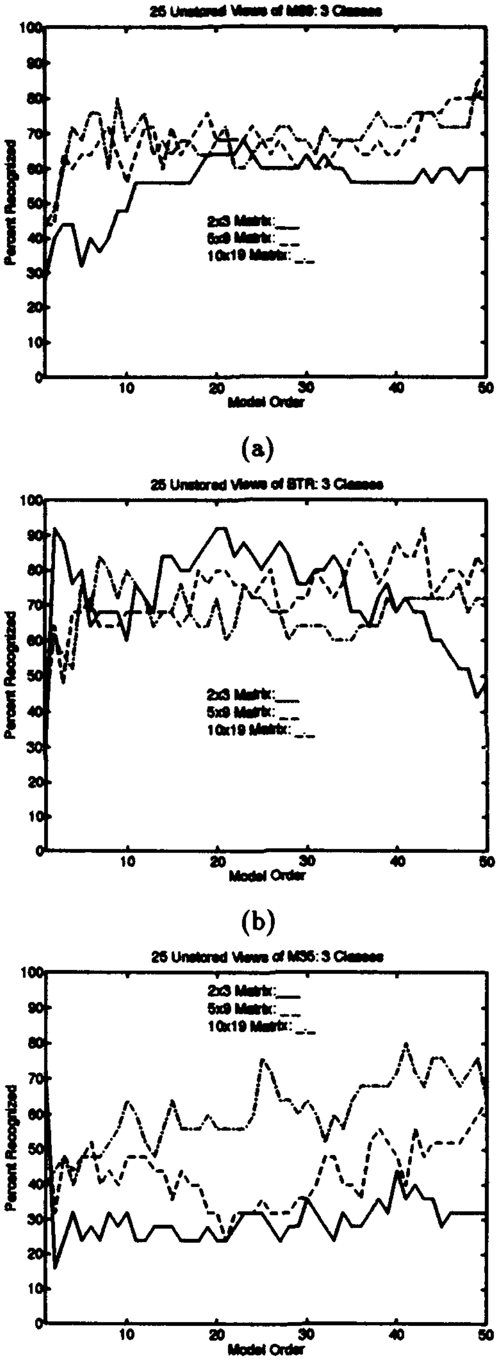
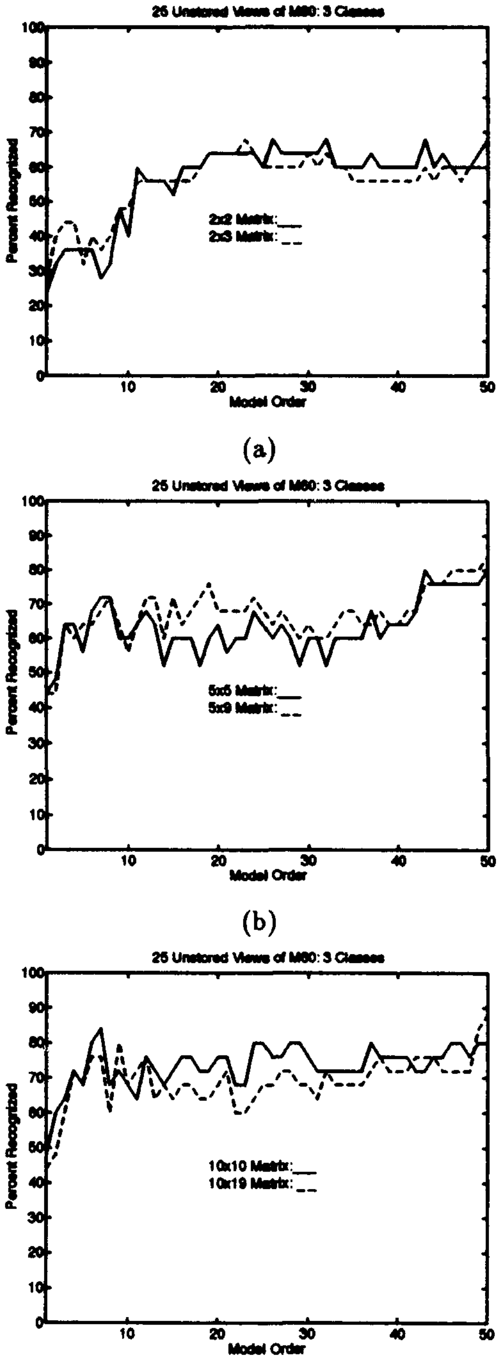


Figure A.5 Comparison using 180^0 template for unstored view recognition. Results shown for (a) M60 Tank (b) BTR APC (c) M35 Truck

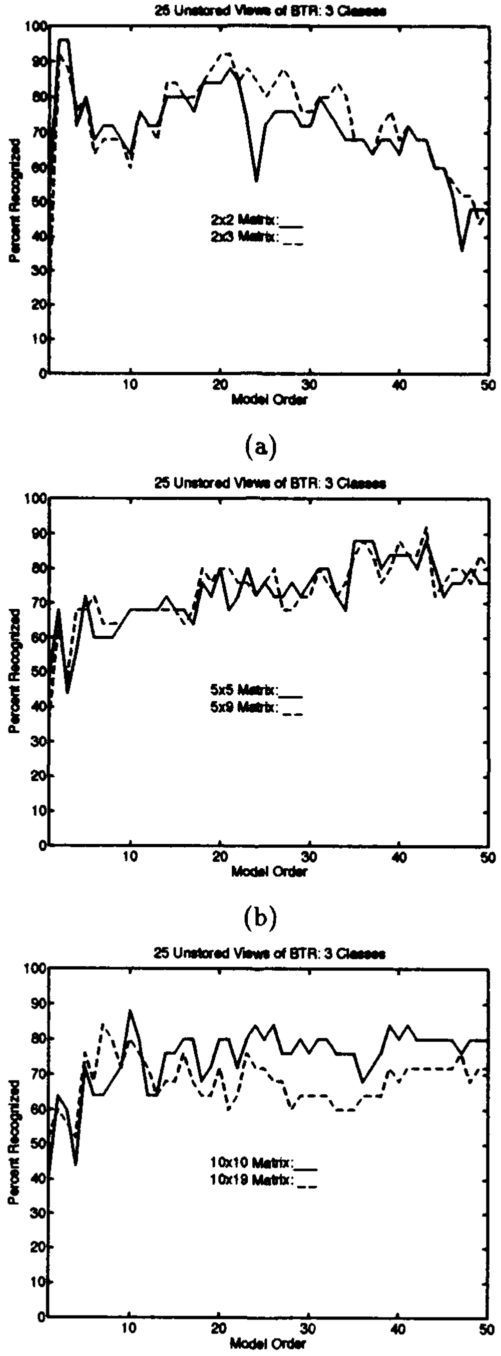


(a)

(b)

(c)

Figure A.6 Comparison using 90^0 ($2 \times 2, 5 \times 5, 10 \times 10$) and 180^0 ($2 \times 3, 5 \times 9, 10 \times 19$) templates for unstored view recognition of the M60.

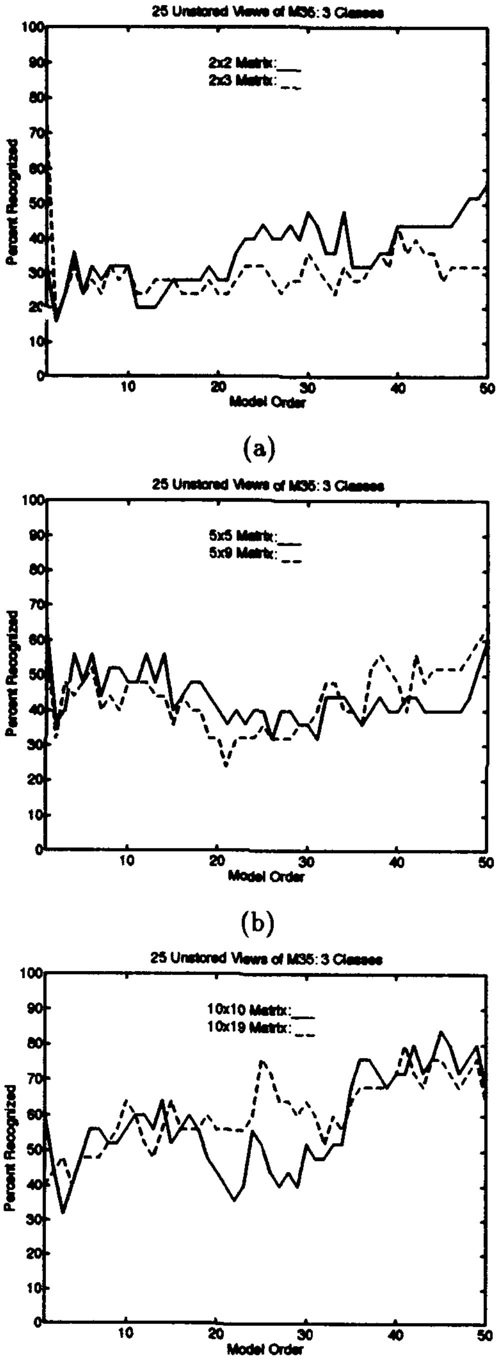


(a)

(b)

(c)

Figure A.7 Comparison using 90^0 ($2 \times 2, 5 \times 5, 10 \times 10$) and 180^0 ($2 \times 3, 5 \times 9, 10 \times 19$) templates for unstored view recognition of the BTR.

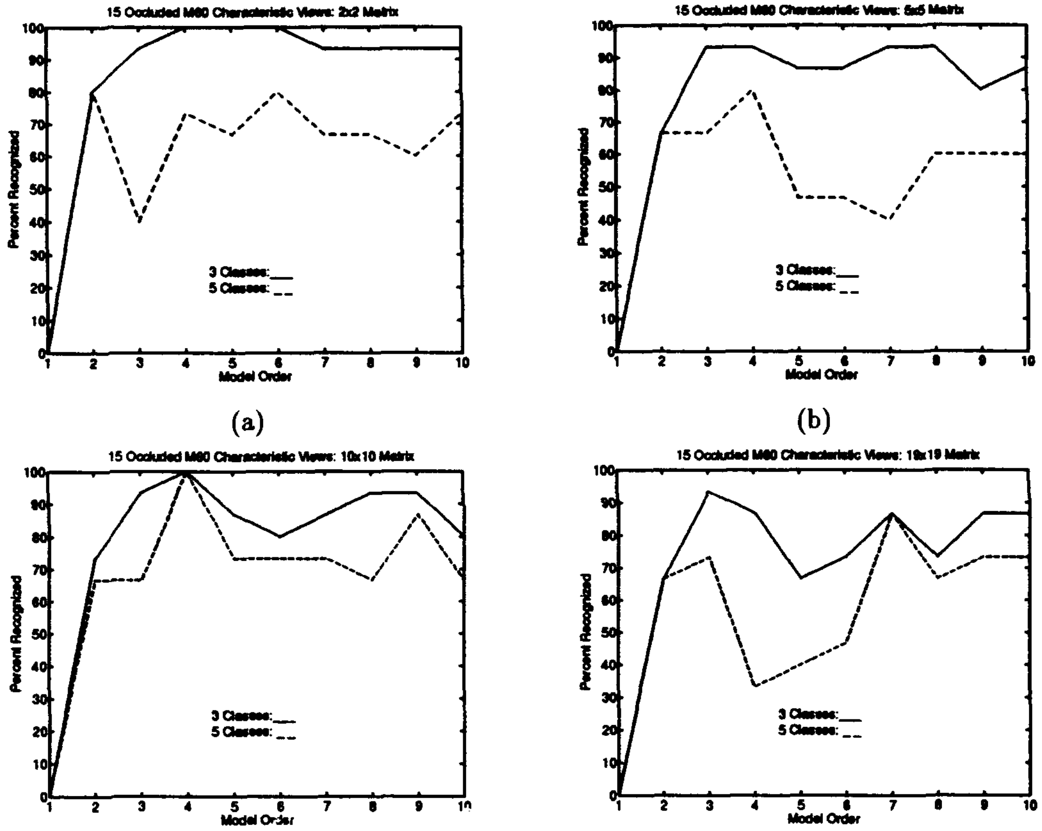


(a)

(b)

(c)

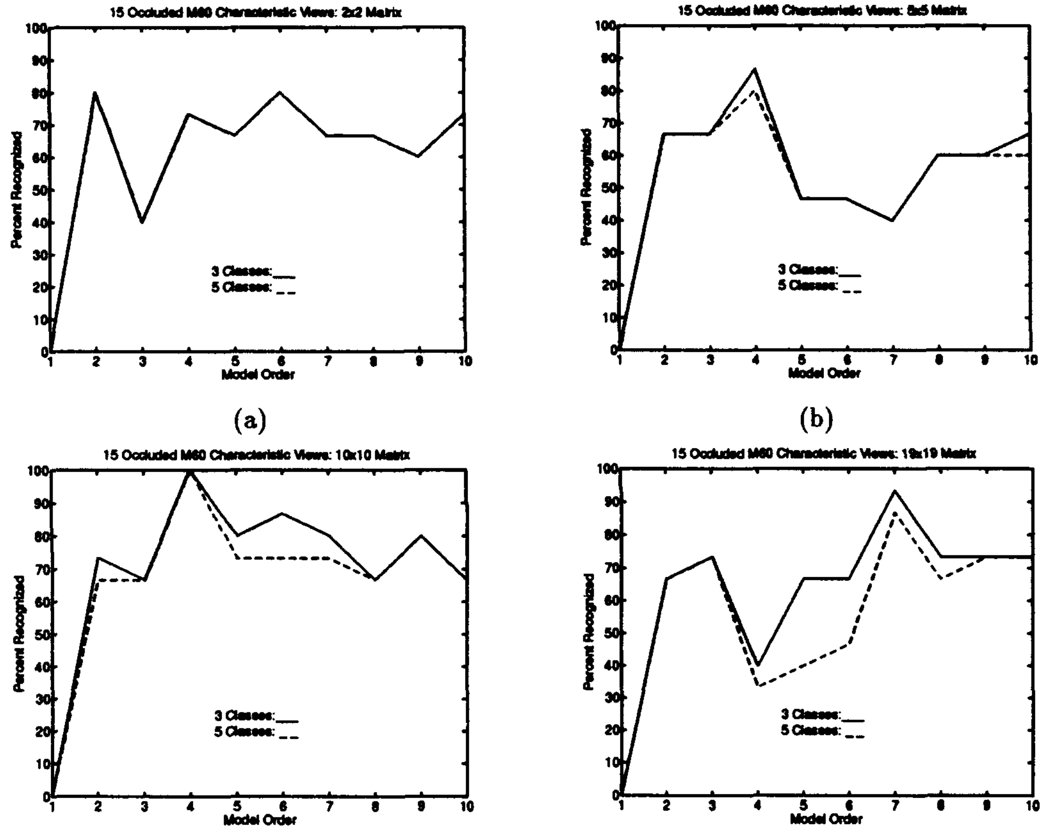
Figure A.8 Comparison using 90^0 ($2 \times 2, 5 \times 5, 10 \times 10$) and 180^0 ($2 \times 3, 5 \times 9, 10 \times 19$) templates for unstored view recognition of the M35.



(a) (b)

(c) (d)

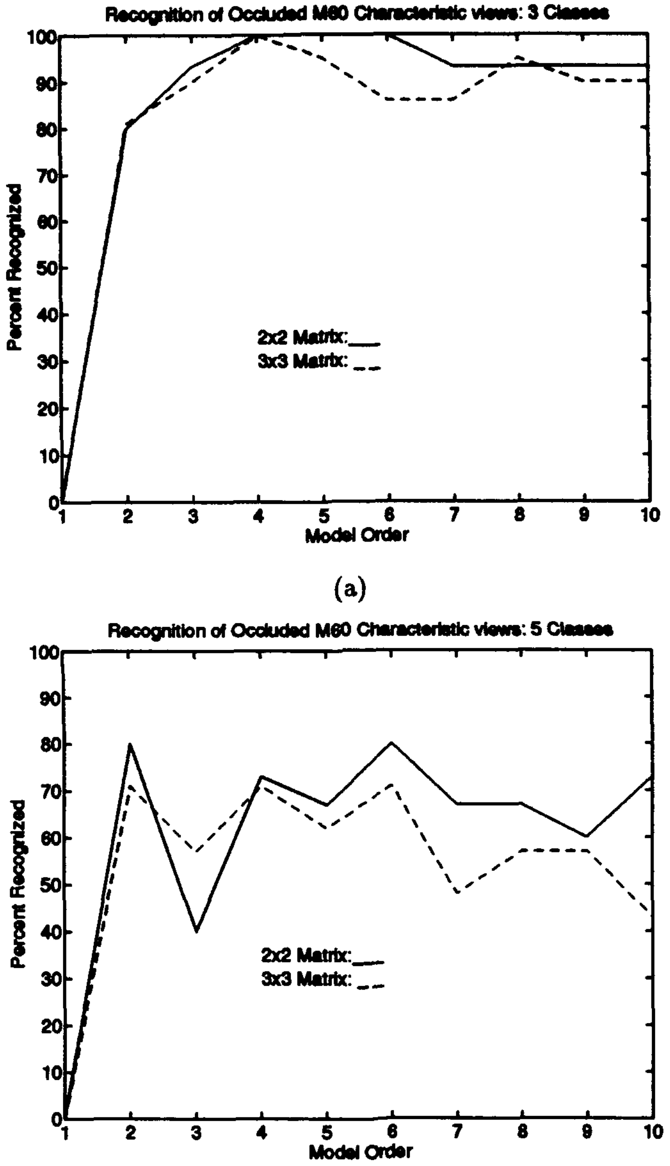
Figure A.9 Comparison of Occlusion results for 15 occluded M60 characteristic views using *dissimilar* 3 class set (M60, BTR, M35).



(a) (b)

(c) (d)

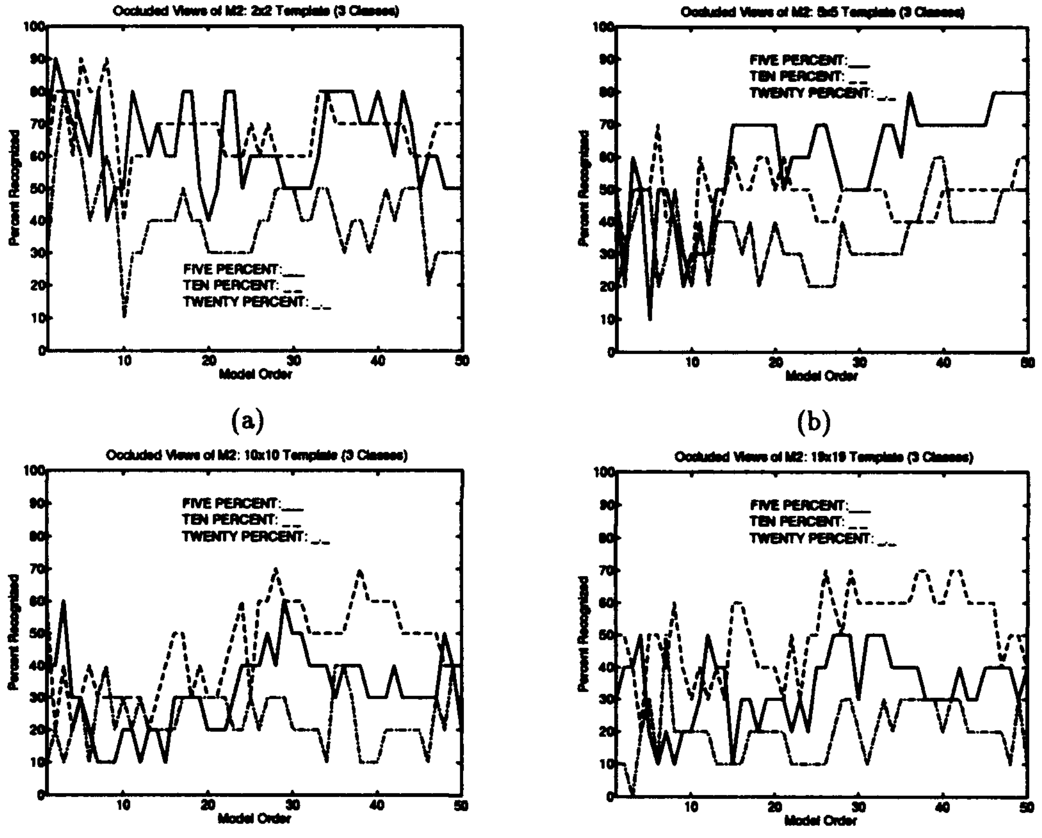
Figure A.10 Occlusion results for 15 occluded M60 characteristic views using *similar* 3 class set (M60, M2, T62).



(a)

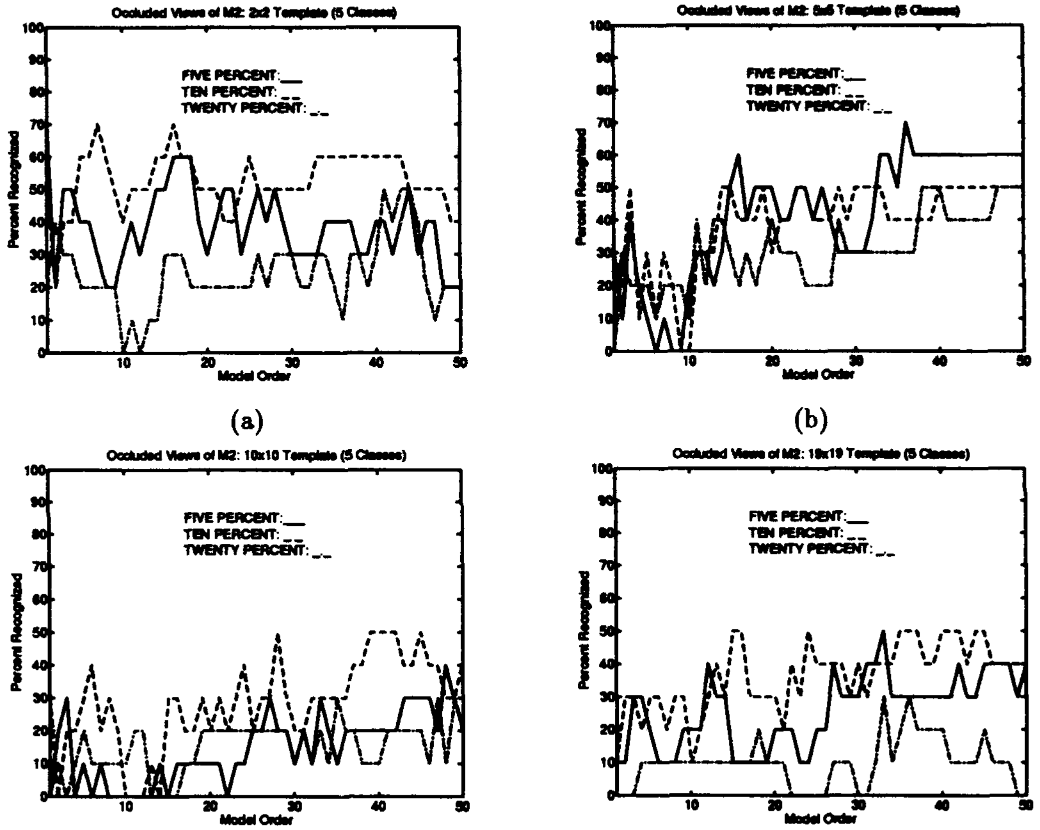
(b)

Figure A.11 Comparison of 90^0 and 360^0 templates for recognition of occluded M60 characteristic views (a) 3 Classes (b) 5 Classes.



(a) (b) (c) (d)

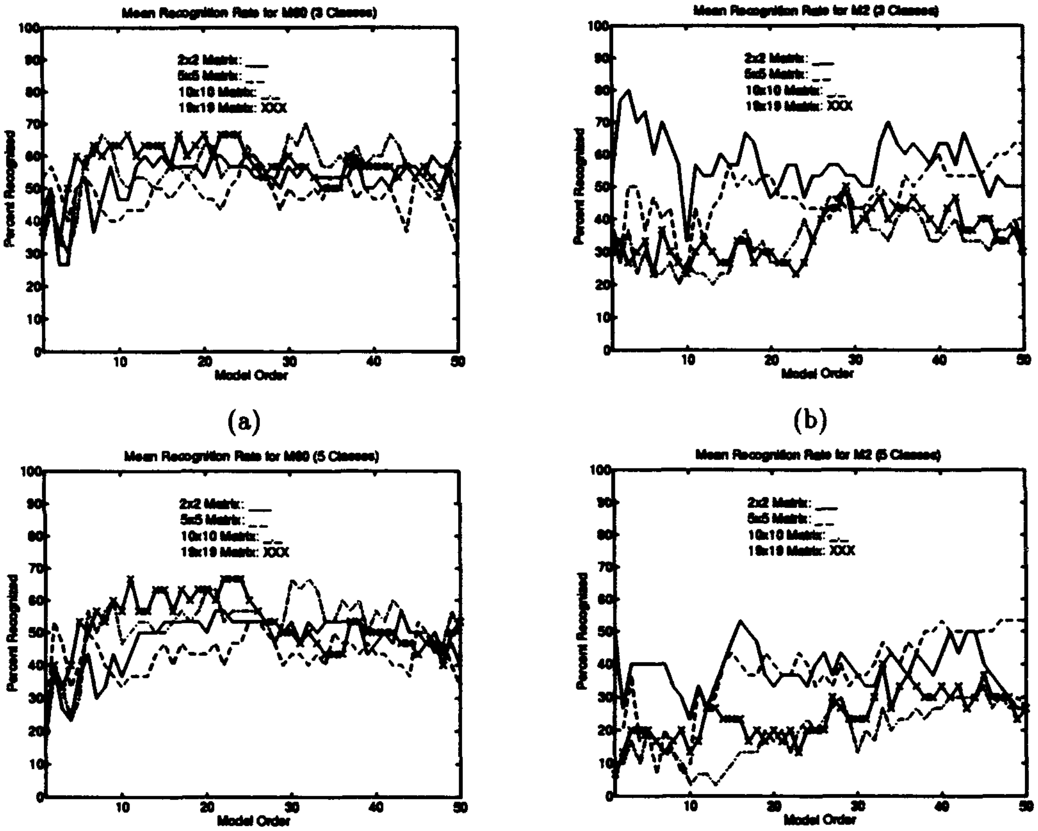
Figure A.12 3 class recognition rates for 10 occluded M2 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix.



(a) (b)

(c) (d)

Figure A.13 3 class recognition rates for 10 occluded M2 views. Various matrix sizes used are (a) 2 × 2 Matrix (b) 5 × 5 Matrix (c) 10 × 10 Matrix (d) 19 × 19 Matrix.



(c) (d)

Figure A.14 Mean value of occlusion recognition rates for various size templates. M60 and M35 results shown for 3 and 5 Classes.

Appendix B. White Noise Test Source Code

B.1 Introduction

This appendix contains MATLAB software routines and relevant data sets used for the CPARCOR coefficient white noise test (see Chapter III). Material is presented in the same order as it was used for testing.

B.2 MATLAB Test Routines and Data Sets

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 This routine will generate a set of stable test coefficients for use
 with the white noise sequence routine.  For stability, all points
 must lie within the unit circle.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% To insure stability, the following parameters were used to
% define the actual test data:

r=[0.99 0.95 0.90];
freq=[255 2330 3000];
fs=8000;
fp=125;
np=100;

% Compute polar representation
  rads=(freq/(fs/2))*pi;
% Compute cartesian coordinates
  x=r.*cos(rads); y=r.*sin(rads);
  p0=x+y*j;
  p1=x-y*j;
% Compute the polynomial representation
  pp=[p0 p1];
  ppoly=real(poly(pp));
  a=ppoly

% Application of the above code results in the following test vector:
%
%  a=[1.0000  -0.1807  -0.1023  -1.2578  0.0226  0.0939  0.7165];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 This routine will generate the complex white noise sequence based on
 a set of parameters defined by both the user and the model's
 transfer function.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The following parameter assignments were used:

N=10000; % Desired white noise sequence length
s=1;     % Variance
b=1;     % Defined based on the model's transfer function (Y(z) = 1).


x=sqrt(s/2)*(randn(1,N)+j*randn(1,N));  % Generates the CWGN sequence

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 This routine will filter the data in vector 'x' with the filter
 described by vectors 'a' and 'b' to produce the filtered data 'y'
 In this case, 'y' is  the filter's infinite impulse response.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


  y=filter(b,a,x);

  z=y;  % Arbitrary label used to define the
        % vector input for the CPARCOR algorithm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 Following are the actual parameters applied to the CPARCOR algorithm
 for generation of the complex coefficients.  Note that only the
 results of the CPARCOR algorithm are included here, not the code.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k = 0;        % Must be initialized to zero for proper indexing
sum = 0;      % Must be initialized to zero for proper indexing
N = 10000;    % Defined based on the number of sample points
order = 6;    % User specified model order, from 1 to N


% Application of the CPARCOR algorithm results in the following coefficients:

% coeff = [-0.1845+0.0017i  -0.0993-0.0007i  -1.2522+0.0048i  0.0231-0.0018i
           0.0922-0.0025i   0.7138+0.0006i]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 This routine will generate a pole-zero plot of the data defined by
 the coefficients shown above.  The resulting plot was included in
```

```
Chapter III.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


  th0=poly2th([1 coeff],[1]);
  [zepo,k]=th2zp(th0);
  zpplot(zepo)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 This routine will transform the pole-zero plot back into a vector form.
 A comparison of the original, CPARCOR algorithm, and pole-zero plot
 coefficients can then be made.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


   th = ar(z,6,'ls','ppw');
   ppoly = th2poly(th);


% The result is the following vector:

% ppoly = [1.0000  -0.1844-0.0015i  -0.0993+0.0008i  -1.2523-0.0048i
          0.0230+0.0017i 0.0922+0.0023i 0.7139-0.0006i]
```

Appendix C. CPARCOR Algorithm Source Code

C.1 Introduction

This appendix contains a listing of the CPARCOR algorithm as written in the MATLAB programming language. The following preliminary subroutines were required to generate software compatible test images:

1. *Load and Enhance Subroutine*: Creates a smooth, unbroken image boundary.
2. *Trace Subroutine*: Traces the image's boundary and calculates its centroid.
3. *Sampling Subroutine*: Provides a sampled version of the image's traced boundary.

C.2 CPARCOR Algorithm

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CPARCOR ALGORITHM:  Using a series of subroutines, the CPARCOR
coefficients of a 2-D image are calculated for a model of order 'm'

written by: David E. Chelen
date: JULY 93

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Load and Enhance Subroutine:  This subroutine will first read a file
into MATLAB and then run an edge detecting mask over the
surface to enhance the image's boundary.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load FILENAME.ext               % Load in image file
function coeff = code(flip)     % Declare the algorithm a function
                                % that will return the variable 'view'

x = view(1:128,1:128);          % Assign image to 128x128 array

a = [30 50 30; 0 0 0; -30 -50 -30];  % Edge enhancing mask
exh = conv2(x,a);                    % Enhance horizontal edges
exv = conv2(x,a');                   % Enhance vertical edges
```

```
ex = sqrt(exh.^2 + exv.^2);          % Combine the two enhanced edges

x=(ex>0);          % Threshold for image=1, background=0


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Trace Subroutine:  This subroutine is used to trace the image's
boundary.  Based on the boundary points returned, the image's
centroid is also determined.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = 1;                  % Index variable assignment
foundit = 0;            % FLAG to indicate start of image pixels
maxm = 128;             % Maximum ‘m’ for the mxn matrix
maxn = 128;             % Maximum ‘n’ for the mxn matrix
while m ~= foundit      % Scan  matrix until an image pixel is found
   m = 1+m;             % Increment matrix row (m)
    for n = 1:maxn      % Continue until last column (n) is reached
if x(m,n) == 0
    oldm = m;
    sm = m;
            oldn = n;
    sn = n;
        else foundit = m;           % Set FLAG ‘hi’ when
                                      image pixel found
     start = [sm sn];
lastzero = [sm sn]; % Keep location of last
                                      background pixel
     stop = [m n];
firstone = [m n];   % Keep location of first
                                      image pixel
break
end
     end
end

t=1;                  % Index variable assignment
flag = 1;             % FLAG set for WHILE loop
while flag
  if stop == start    % Keep going until return to start point
   break
   flag = 0
   else if oldn < n     % Decide which way to turn based on
        % the current pixel value
       oldn = n;
if x(m,n) == 1
   tmp(t) = [m+(n*i)];
   t=t+1;
   m = m-1;
```

```
else m = m+1;
end
   elseif oldn > n
  oldn = n;
if x(m,n) == 1
   tmp(t) = [m+(n*i)];
   t=t+1;
   m = m+1;
        else m = m-1;
        end
   elseif oldm < m
  oldm = m;
if x(m,n) == 1
   tmp(t) = [m+(n*i)];
   t=t+1;
     n = n+1;
else n = n-1;
end
   elseif oldm > m
  oldm = m;
if x(m,n) == 1
   tmp(t) = [m+(n*i)];
   t=t+1;
           n = n-1;
else n = n+1;
end
   end
      stop = [m n];
end
end

tmpsize = length(tmp);  % Find the length of the traced boundary
ztmp(1) = tmp(1);       % Store first point as reference
b=1;                    % Index variable assignment
for old=1:(tmpsize-1)          % If previous and current pixel
    if ztmp(b)~=tmp(old+1);    % locations are the same, get
       ztmp(b+1) = tmp(old+1); % rid of the repeated point
       b=b+1;
    else
       b=b;
    end
end

clear total            % Re-initialize variables
clear sum
total = sum(ztmp);     % Calculate the CENTROID based on the
size = length(ztmp);   % boundary points returned by the trace
center = total/size;
tmp = 0;               % Re-initialize variables
ztmp = 0;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Sampling Subroutine:  This subroutine uses a modified version of the
original Trace Subroutine to return boundary points that are now
complex numbers referenced from the centroid.  The final sampling rate
and model order are user specified.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


start = lastzero;    % Assign the starting point to the same relative
stop = firstone;     % point on the object as was originally used
mref = real(center); % Create the modified x+iy axis
nref = imag(center);
t=1;
flag = 1;            % Set FLAG for while loop
while flag
  if stop == start
   break
   flag = 0;
   else if oldn < n          % Decide which way to turn based on
       oldn = n;         % current pixel type
if x(m,n) == 1
   tmp(t) = [(m-mref)+(n-nref)*i];
   t=t+1;
   m = m-1;
else m = m+1;
end
   elseif oldn > n
  oldn = n;
if x(m,n) == 1
   tmp(t) = [(m-mref)+(n-nref)*i];
   t=t+1;
   m = m+1;
        else m = m-1;
        end
   elseif oldm < m
  oldm = m;
if x(m,n) == 1
   tmp(t) =  [(m-mref)+(n-nref)*i];
   t=t+1;
     n = n+1;
else n = n-1;
end
   elseif oldm > m
  oldm = m;
if x(m,n) == 1
   tmp(t) = [(m-mref)+(n-nref)*i];
   t=t+1;
           n = n-1;
else n = n+1;
```

```
end
   end
      stop = [m n];
end
end

tmpsize = length(tmp);     % Go through and eliminate repeated points
ztmp(1) = tmp(1);
b=1;
for old=1:(tmpsize-1)
    if ztmp(b)~=tmp(old+1);
       ztmp(b+1) = tmp(old+1);
       b=b+1;
    else
       b=b;
    end
end

N = 160;                 % Define the desired number of samples
order=50;                % Set the order of model to be generated
num = length(ztmp);      % Size of temp z vector
interval = (num/N);      % Set sample interval
  if interval == 0
     interval = 1;
  end
z = [ztmp(1:interval:num)]; % Put sampled values into test vector
  if length(z)<N
    N=length(z);
  end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CPARCOR Routine: This routine creates the CPARCOR coefficients, for the
user specified model order, based on the sampled image vector, Z.
Although only three lines long in original form, the MATLAB version of
this algorithm is significantly longer.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


k = 0;      % Initialize to zero for proper indexing
sum = 0;    % Initialize to zero for proper indexing

%************This will find Ro****************

 for j=0:(N-1)
     L =  z(j+1)*conj(z((j+1)-k));
     sum = sum + L;
 end
 Ro = (1/N)*sum;
```

```
%************This gets all the other R's************

R = 0;
for k = 1:order
   sum = 0;
      for j=0:(N-1)
         if ((j+1)-k) <=0
            L =  z(j+1)*conj(z(N+((j+1)-k)));
         else
            L =  z(j+1)*conj(z((j+1)-k));
         end
            sum = sum + L;
      end
            R(k) = (1/N)*sum;
   end

%*****This gets the CAR and CPARCOR Coefficients***********

a = 0;
a(1) = R(1)/Ro;
Cpar(1)=a(1);    % First Pm is a(1)
   for m = 2:order
       Rflp = fliplr(R(1:m-1));
       Atr = a(1:m-1).';
       Pm = (R(m) - (Rflp*Atr))/(Ro-(R(1:m-1)*conj(Atr)));  %CPARCOR coeff
       Cpar(m) = Pm;
       a(1:m) = [(a(1:m-1) - conj((fliplr(a(1:m-1))))*Pm)  Pm];  %CAR coeff
       car(1:m)=a(1:m);
   end

coeff=Cpar; % CPARCOR coefficients returned by the function call
```

Appendix D. Template Matching Algorithm

D.1 Introduction

This appendix contains a listing of the *Template Matching* algorithm as written in the MATLAB programming language. Classification is based on a Euclidean distance metric defined as follows:

$$D(x) = \min_{i} \| \mathbf{x} - \mathbf{z}_i \| \tag{D.1}$$

where $\| \cdot \|$ defines the Euclidean norm for the expression, and $D(x)$ is the smallest of all distances between the unknown input-view ($\mathbf{x}$) and the ith stored template-view ($\mathbf{z}_i$)[18]. In other words, classification is based upon the minimum Euclidean distance between a known template-view and the unknown input-view. Templates were based on $\mathbf{m} \times \mathbf{n}$ matrices that contained unnormalized CPARCOR coefficient vectors representing the characteristic views for various allowable object orientations. Individual views are labeled with the following format:

class(azimuth)_(elevation)(1:model_order)

where individual classes are assigned to class designators **a** thru **e**. For example, the M60 Tank was assigned to the designator **a**, while the BTR APC was labeled **c**. A brief description of each template follows:

- **360^0 View Template (3×3 Matrix)**: Contains the minimum number of characteristic views (CPARCOR coefficients) required for total-range 360^0 azimuth and 90^0 elevation image recognition.
- **180^0 View Template ($2 \times 3, 5 \times 9, 10 \times 19$ Matrix)**: Representing the same 180^0 view, the three matrices contain different numbers of characteristic views. For example, the 2×3 matrix contains six characteristic views, while the 10×19 contains 190.
- **90^0 View Template ($2 \times 2, 5 \times 5, 10 \times 10, 19 \times 19$ Matrix)**: Although representative of the same 90^0 view, the 2×2 matrix does it with only four characteristic views, while the 19×19 uses 361. Note that the 19×19 contains the maximum number of characteristic views possible for the given range. Thus, the 19×19 matrix was not used for *unstored* view recognition tests.

D.2 Matrix Source Code

The following source code creates the template for application to the *Template Matching* algorithm. Examples of the actual test matricies (described above) are also included.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MATRIX GENERATION: Different Matricies for Different Templates
written by: David E. Chelen
date: AUG 93

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Enter all initial test parameters in this section

order = XX          % ENTER order of test CPARCOR model
nsize = XX;         % ENTER 'n' of the mxn matrix
msize = XX;         % ENTER the 'm' of the mxn matrix
numclasses = XX;    % ENTER the number of classes being tested

frac  = order     % Variables assigned to represent model order
f=frac

mstop = (msize*numclasses)+1;  % Identify the final matrix row
elements=order*nsize;          % Size of matrix rows
f1=(-9)*ones(1,order);         % Set matrix boarder elements to a known value
botb=[(-9)*ones(1,elements)];  % Boarder to 'pad' end of matrix

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EXAMPLE of 360 degree template, 3x3 matrix per class (5 classes)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r1  = [a315_0(1:f)  a0_0(1:f)   a45_0(1:f)];   % M60 views
r2  = [a270_0(1:f)  a0_90(1:f)  a90_0(1:f)];
r3  = [a225_0(1:f)  a180_0(1:f) a135_0(1:f)];

r4  = [b315_0(1:f)  b0_0(1:f)   b45_0(1:f)];   % M35 views
r5  = [b270_0(1:f)  b0_90(1:f)  b90_0(1:f)];
r6  = [b225_0(1:f)  b180_0(1:f) b135_0(1:f)];

r7  = [c315_0(1:f)  c0_0(1:f)   c45_0(1:f)];   % BTR views
r8  = [c270_0(1:f)  c0_90(1:f)  c90_0(1:f)];
r9  = [c225_0(1:f)  c180_0(1:f) c135_0(1:f)];

r10 = [d315_0(1:f)  d0_0(1:f)   d45_0(1:f)];   % T62 views
```

```
r11 = [d270_0(1:f)  d0_90(1:f)  d90_0(1:f)];
r12 = [d225_0(1:f)  d180_0(1:f) d135_0(1:f)];

r13 = [e315_0(1:f)  e0_0(1:f)   e45_0(1:f)];   % M2 views
r14 = [e270_0(1:f)  e0_90(1:f)  e90_0(1:f)];
r15 = [e225_0(1:f)  e180_0(1:f) e135_0(1:f)];

template = [r1;r2;r3;r4;r5;r6;r7;r8;r9;r10;r11;r12;r13;r14;r15;botb];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EXAMPLE of 180 degree template, 2x3 matrix per class (3 classes)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r1 = [a0_0(1:f)   a45_0(1:f)];
r2 = [a0_90(1:f)  a90_0(1:f)];
r3 = [a180_0(1:f) a135_0(1:f)];

r4 = [b0_0(1:f)   b45_0(1:f)];
r5 = [b0_90(1:f)  b90_0(1:f)];
r6 = [b180_0(1:f) b135_0(1:f)];

r7 = [c0_0(1:f)   c45_0(1:f)];
r8 = [c0_90(1:f)  c90_0(1:f)];
r9 = [c180_0(1:f) c135_0(1:f)];

template = [r1;r2;r3;r4;r5;r6;r7;r8;r9;botb];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EXAMPLE of 90 degree template, 5x5 matrix per class (3 classes)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r1  = [a0_0(1:f)  a15_0(1:f)  a25_0(1:f)  a35_0(1:f)  a45_0(1:f)];
r2  = [a0_25(1:f) a15_25(1:f) a30_25(1:f) a45_25(1:f) a55_0(1:f)];
r3  = [a0_50(1:f) a25_50(1:f) a45_50(1:f) a60_25(1:f) a65_0(1:f)];
r4  = [a0_75(1:f) a45_70(1:f) a65_50(1:f) a75_25(1:f) a75_0(1:f)];
r5  = [a0_90(1:f) a90_75(1:f) a90_50(1:f) a90_25(1:f) a90_0(1:f)];

r6  = [b0_0(1:f)  b15_0(1:f)  b25_0(1:f)  b35_0(1:f)  b45_0(1:f)];
r7  = [b0_25(1:f) b15_25(1:f) b30_25(1:f) b45_25(1:f) b55_0(1:f)];
r8  = [b0_50(1:f) b25_50(1:f) b45_50(1:f) b60_25(1:f) b65_0(1:f)];
r9  = [b0_75(1:f) b45_70(1:f) b65_50(1:f) b75_25(1:f) b75_0(1:f)];
r10 = [b0_90(1:f) b90_75(1:f) b90_50(1:f) b90_25(1:f) b90_0(1:f)];

r11 = [c0_0(1:f)  c15_0(1:f)  c25_0(1:f)  c35_0(1:f)  c45_0(1:f)];
r12 = [c0_25(1:f) c15_25(1:f) c30_25(1:f) c45_25(1:f) c55_0(1:f)];
r13 = [c0_50(1:f) c25_50(1:f) c45_50(1:f) c60_25(1:f) c65_0(1:f)];
```

```
r14 = [c0_75(1:f) c45_70(1:f) c65_50(1:f) c75_25(1:f) c75_0(1:f)];
r15 = [c0_90(1:f) c90_75(1:f) c90_50(1:f) c90_25(1:f) c90_0(1:f)];

template = [r1;r2;r3;r4;r5;r6;r7;r8;r9;r10;r11;r12;r13;r14;r15;botb];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EXAMPLE of 90 degree template, 10x10 matrix per class (1 class)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


r1  = [a0_0(1:f)    a5_0(1:f)    a10_0(1:f)   a15_0(1:f)   a20_0(1:f)
       a25_0(1:f)   a30_0(1:f)   a35_0(1:f)   a40_0(1:f)   a45_0(1:f)];
r2  = [a0_10(1:f)   a5_10(1:f)   a10_10(1:f)  a15_10(1:f)  a20_10(1:f)
       a30_10(1:f)  a35_10(1:f)  a40_10(1:f)  a45_10(1:f)  a50_0(1:f)];
r3  = [a0_20(1:f)   a5_20(1:f)   a10_20(1:f)  a20_20(1:f)  a30_20(1:f)
       a35_20(1:f)  a40_20(1:f)  a45_20(1:f)  a50_10(1:f)  a55_0(1:f)];
r4  = [a0_30(1:f)   a10_30(1:f)  a20_30(1:f)  a30_30(1:f)  a35_30(1:f)
       a40_30(1:f)  a45_30(1:f)  a50_20(1:f)  a55_10(1:f)  a60_0(1:f)];
r5  = [a0_40(1:f)   a10_40(1:f)  a20_40(1:f)  a30_40(1:f)  a40_40(1:f)
       a45_40(1:f)  a50_30(1:f)  a55_20(1:f)  a60_10(1:f)  a65_0(1:f)];
r6  = [a0_50(1:f)   a15_50(1:f)  a25_50(1:f)  a35_50(1:f)  a45_50(1:f)
       a50_40(1:f)  a55_30(1:f)  a60_20(1:f)  a70_10(1:f)  a70_0(1:f)];
r7  = [a0_60(1:f)   a15_60(1:f)  a30_60(1:f)  a45_60(1:f)  a55_50(1:f)
       a60_40(1:f)  a60_30(1:f)  a70_20(1:f)  a75_10(1:f)  a75_0(1:f)];
r8  = [a0_70(1:f)   a25_70(1:f)  a45_70(1:f)  a60_60(1:f)  a65_50(1:f)
       a70_40(1:f)  a70_30(1:f)  a80_20(1:f)  a80_10(1:f)  a80_0(1:f)];
r9  = [a0_80(1:f)   a45_80(1:f)  a65_70(1:f)  a75_60(1:f)  a75_50(1:f)
       a80_40(1:f)  a80_30(1:f)  a85_20(1:f)  a85_10(1:f)  a85_0(1:f)];
r10 = [a0_90(1:f)   a90_80(1:f)  a90_70(1:f)  a90_60(1:f)  a90_50(1:f)
       a90_40(1:f)  a90_30(1:f)  a90_20(1:f)  a90_10(1:f)  a90_0(1:f)];


template = [r1;r2;r3;r4;r5;r6;r7;r8;r9;r10;botb];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

EXAMPLE of 90 degree template, 19x19 matrix per class (1 class)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a1 = [a0_0(1:f) a5_0(1:f) a10_0(1:f) a15_0(1:f) a20_0(1:f) a25_0(1:f)
      a30_0(1:f) a35_0(1:f) a40_0(1:f) a45_0(1:f) a50_0(1:f) a55_0(1:f)
      a60_0(1:f) a65_0(1:f) a70_0(1:f) a75_0(1:f) a80_0(1:f) a85_0(1:f)
      a90_0(1:f)];
a2 = [a0_5(1:f) a5_5(1:f) a10_5(1:f) a15_5(1:f) a20_5(1:f) a25_5(1:f)
      a30_5(1:f) a35_5(1:f) a40_5(1:f) a45_5(1:f) a50_5(1:f)  a55_5(1:f)
      a60_5(1:f) a65_5(1:f) a70_5(1:f) a75_5(1:f) a80_5(1:f) a85_5(1:f)
      a90_5(1:f)];
a3 = [a0_10(1:f) a5_10(1:f) a10_10(1:f) a15_10(1:f) a20_10(1:f) a25_10(1:f)
```

```
      a30_10(1:f) a35_10(1:f) a40_10(1:f) a45_10(1:f) a50_10(1:f) a55_10(1:f)
      a60_10(1:f) a65_10(1:f) a70_10(1:f) a75_10(1:f) a80_10(1:f) a85_10(1:f)
      a90_10(1:f)];
a4 = [a0_15(1:f) a5_15(1:f) a10_15(1:f) a15_15(1:f) a20_15(1:f) a25_15(1:f)
      a30_15(1:f) a35_15(1:f) a40_15(1:f) a45_15(1:f) a50_15(1:f) a55_15(1:f)
      a60_15(1:f) a65_15(1:f) a70_15(1:f) a75_15(1:f) a80_15(1:f) a85_15(1:f)
      a90_15(1:f)];
a5 = [a0_20(1:f) a5_20(1:f) a10_20(1:f) a15_20(1:f) a20_20(1:f) a25_20(1:f)
      a30_20(1:f) a35_20(1:f) a40_20(1:f) a45_20(1:f) a50_20(1:f) a55_20(1:f)
      a60_20(1:f) a65_20(1:f) a70_20(1:f) a75_20(1:f) a80_20(1:f) a85_20(1:f)
      a90_20(1:f)];
a6 = [a0_25(1:f) a5_25(1:f) a10_25(1:f) a15_25(1:f) a20_25(1:f) a25_25(1:f)
     a30_25(1:f) a35_25(1:f) a40_25(1:f) a45_25(1:f) a50_25(1:f) a55_25(1:f)
     a0_25(1:f) a65_25(1:f) a70_25(1:f) a75_25(1:f) a80_25(1:f) a85_25(1:f)
     a90_25(1:f)];
a7 = [a0_30(1:f) a5_30(1:f) a10_30(1:f) a15_30(1:f) a20_30(1:f) a25_30(1:f)
      a30_30(1:f) a35_30(1:f) a40_30(1:f) a45_30(1:f) a50_30(1:f) a55_30(1:f)
      a60_30(1:f) a65_30(1:f) a70_30(1:f) a75_30(1:f) a80_30(1:f) a85_30(1:f)
      a90_30(1:f)];
a8 = [a0_35(1:f) a5_35(1:f) a10_35(1:f) a15_35(1:f) a20_35(1:f) a25_35(1:f)
     a30_35(1:f)  a35_35(1:f) a40_35(1:f) a45_35(1:f) a50_35(1:f) a55_35(1:f)
     a60_35(1:f) a65_35(1:f) a70_35(1:f) a75_35(1:f) a80_35(1:f) a85_35(1:f)
     a90_35(1:f)];
a9 = [a0_40(1:f) a5_40(1:f) a10_40(1:f) a15_40(1:f) a20_40(1:f) a25_40(1:f)
      a30_40(1:f) a35_40(1:f) a40_40(1:f) a90_40(1:f) a45_40(1:f) a50_40(1:f)
      a55_40(1:f) a60_40(1:f) a65_40(1:f) a70_40(1:f) a75_40(1:f) a80_40(1:f)
      a85_40(1:f)];
a10 = [a0_45(1:f) a5_45(1:f) a10_45(1:f) a15_45(1:f) a20_45(1:f) a25_45(1:f)
       a30_45(1:f) a35_45(1:f) a40_45(1:f) a45_45(1:f) a50_45(1:f) a55_45(1:f)
       a60_45(1:f) a65_45(1:f) a70_45(1:f) a75_45(1:f) a80_45(1:f) a85_45(1:f)
       a90_45(1:f)];
a11 = [a0_50(1:f)  a5_50(1:f) a10_50(1:f) a15_50(1:f) a20_50(1:f) a25_50(1:f)
       a30_50(1:f) a35_50(1:f) a40_50(1:f) a45_50(1:f) a50_50(1:f) a55_50(1:f)
       a60_50(1:f) a65_50(1:f) a70_50(1:f) a75_50(1:f) a80_50(1:f) a85_50(1:f)
       a90_50(1:f)];
a12 = [a0_55(1:f) a5_55(1:f) a10_55(1:f) a15_55(1:f) a20_55(1:f) a25_55(1:f)
       a30_55(1:f) a35_55(1:f) a40_55(1:f) a45_55(1:f) a50_55(1:f) a55_55(1:f)
       a60_55(1:f) a65_55(1:f) a70_55(1:f) a75_55(1:f) a80_55(1:f) a85_55(1:f)
       a90_55(1:f)];
a13 = [a0_60(1:f) a5_60(1:f) a10_60(1:f) a15_60(1:f) a20_60(1:f) a25_60(1:f)
       a30_60(1:f) a35_60(1:f) a40_60(1:f) a45_60(1:f) a50_60(1:f) a55_60(1:f)
       a60_60(1:f) a65_60(1:f) a70_60(1:f) a75_60(1:f) a80_60(1:f) a85_60(1:f)
       a90_60(1:f)];
a14 = [a0_65(1:f) a5_65(1:f) a10_65(1:f) a15_65(1:f) a20_65(1:f) a25_65(1:f)
       a30_65(1:f) a35_65(1:f) a40_65(1:f) a45_65(1:f) a50_65(1:f) a55_65(1:f)
       a60_65(1:f) a65_65(1:f) a70_65(1:f) a75_65(1:f) a80_65(1:f) a85_65(1:f)
       a90_65(1:f)];
a15 = [a0_70(1:f) a5_70(1:f) a10_70(1:f) a15_70(1:f) a20_70(1:f) a25_70(1:f)
       a30_70(1:f) a35_70(1:f) a40_70(1:f) a45_70(1:f) a50_70(1:f) a55_70(1:f)
       a60_70(1:f) a65_70(1:f) a70_70(1:f) a75_70(1:f) a80_70(1:f) a85_70(1:f)
       a90_70(1:f)];
```

```
a16 = [a0_75(1:f) a5_75(1:f) a10_75(1:f) a15_75(1:f) a20_75(1:f) a25_75(1:f)
       a30_75(1:f) a35_75(1:f) a40_75(1:f) a45_75(1:f) a50_75(1:f) a55_75(1:f)
       a60_75(1:f) a65_75(1:f) a70_75(1:f) a75_75(1:f) a80_75(1:f) a85_75(1:f)
       a90_75(1:f)];
a17 = [a0_80(1:f) a5_80(1:f) a10_80(1:f) a15_80(1:f) a20_80(1:f) a25_80(1:f)
       a30_80(1:f) a35_80(1:f) a40_80(1:f) a45_80(1:f) a50_80(1:f) a55_80(1:f)
       a60_80(1:f) a65_80(1:f) a70_80(1:f) a75_80(1:f) a80_80(1:f) a85_80(1:f)
       a90_80(1:f)];
a18 = [a0_85(1:f) a5_85(1:f) a10_85(1:f) a15_85(1:f) a20_85(1:f) a25_85(1:f)
       a30_85(1:f) a35_85(1:f) a40_85(1:f) a45_85(1:f) a50_85(1:f) a55_85(1:f)
       a60_85(1:f) a65_85(1:f) a70_85(1:f) a75_85(1:f) a80_85(1:f) a85_85(1:f)
       a90_85(1:f)];
a19 = [a0_90(1:f) a5_90(1:f) a10_90(1:f) a15_90(1:f) a20_90(1:f) a25_90(1:f)
       a30_90(1:f) a35_90(1:f) a40_90(1:f) a45_90(1:f) a50_90(1:f) a55_90(1:f)
       a60_90(1:f) a65_90(1:f) a70_90(1:f) a75_90(1:f) a80_90(1:f) a85_90(1:f)
       a90_90(1:f)];


plateA=[a1;a2;a3;a4;a5;a6;a7;a8;a9;a10;a11;a12;a13;a14;a15;a16;a17;a18;a19;botb];
```

D.3 Template Matching Algorithm Source Code

The *Template Matching* algorithm returns the matrix location and corresponding class label of the *best match* for a given unknown input view. The source code for this algorithm follows:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

TEMPLATE MATCHING ALGORITHM: Clasification based on Euclidean distance
written by: David E. Chelen
date: AUG 93

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


i=0;                    % Initialize index
f=0;                    % Initialize index
store=0;                % Initialize variable
temp=0;
count=0;
flag=0;                 % FLAGs for event indication
firstflag=0;

perfect='I found a perfect match!!'  % Message for perfect match

knownvec = inputview(1:frac);   % This is where the unknown input view
                                % is brought in (CPARCOR vector) and set
                                % to the desired model order

% Extract stored CPARCOR vectors from template for comparison to input view

for mstart=1:mstop      % Index for matrix rows
 if flag==1            % FLAG to indicate a 'prefect' match was found
    mstart=mstart-1;    % Matrix 'm' address of matching view
    break
 else
      nstart=1;          % Matrix 'n' address
      nstop=order;       % Stop based on vector length

     for ngo=1:nsize      % Repeat based on matrix size
       if flag==1
         break
       else
         tempvec = 0;
         w=1;

        for n = nstart:nstop
          tempvec(w) = template(mstart,n);  % Read in template vectors to compare
      % with known input view
       w=w+1;
      end

      if firstflag==0          % If a 'perfect match was not found
         store=tempvec;        % set up an initial distance reference
         oldgap=sum(abs(knownvec-tempvec(1:frac))); % Euclidean distance is
% defined here as the sum of the magnitude of
% the difference between the input view and
% the stored view
         firstflag=1;
      end

      testvec=tempvec(1:frac); % Assign variable name to stored template view
                               % for comparison with input view


 if knownvec==testvec  % If 'perfect' match is found, BREAK out of routine

    flag=1;
    store=testvec;
    count=0;
    mold=mstart;
    perfect
    break

 else                  % If views do not match, find the next best thing

     newgap=sum(abs(knownvec-testvec)); % Euclidean distance between input
% and stored template views

         if newgap < oldgap  % Initial tollerance comparrison of views

          oldgap = newgap; % Establish new Euclidean distance reference
  store = 0;       % Reset variables to zero for next run
          i=0;
          f=0;
          count=0;
          addm=0;
          addn=0;

 end

 end

     if newgap <= oldgap   % STORE the best matched view

       f=f+1;              % Increment index
       addm(f)=mstart;     % Matrix 'm', 'n' address markers
       addn(f)=nstart;
```

```
            for m=1:order

      i=i+1;
              store(i)=tempvec(m);

            end

            count=count+1; %Number of stored matches that were within tollerance
       end

         nstart = nstop+1;     % Adjust starting point for next run
         nstop = nstop+order;  % Adjust stopping point for next run

end
end
temp=store;     % Assign STOREd match(es) to a TEMP vector
end
end
end

  %After this routine, mstart and nstart values are in addm(f)
  % and addn(f).  Here, 'f' should equal 'count'

% CLASS Assignments, based on matrix location, are made here.
% 'XX' is the first row of the next class (or the last row of each
% class plus one.  Think about it-- you'll get it.  I'm outta here.

if addm< XX
  class=1;
elseif addm< XX
  class=2;
elseif addm< XX
  class=3;
elseif addm< XX
  class=4;
elseif addm< XX
  class=5;
end
```

Bibliography

1. Army, U.S. *BRL-CAD* (4.0 Edition). Aberdeen Proving Ground, Maryland 21005-5066: U.S. Army Ballistic Research Laboratory, December 1991.
2. Das, Manohar, et al. "A Bivariate Autoregressive Modeling Technique for Analysis and Classification of Planar Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(1):97–103 (1990).
3. Dubois, Susan R. and Filsin H. Glanz. "An Autoregressive Model Approach to Two-Dimensional Shape Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(1):55–66 (1986).
4. Duda, R. O. and P. E. Hart. *Pattern Classification and Scene Analysis.* New York: Wiley, 1974.
5. Dudani, Sahibsingh A., et al. "Aircraft Identification by Moment Invariants," *IEEE Transactions on Computers*, *C-26*(1):39–45 (1977).
6. Fielding, Kenneth H. *Spatial-Temporal Pattern Recognition Using Hidden Markov Models.* PhD dissertation, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1993.
7. Fielding, Kenneth H., et al. "Spatio-Temporal Pattern Recognition Using Hidden Markov Models," *Proceedings of the SPIE*, *2032* (July 1993).
8. Fukunaga, Keinosuke and Donald M. Hummels. "Bayes error estimation using Parzen and k-NN procedures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(5):634–643 (September 1987).
9. He, Yang and Amlan Kundu. "2-D Shape Classification Using Hidden Markov Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(11):1172–1184 (1991).
10. Kukolich. "Introducing LNKnet." Unpublished manual, 1993.
11. Linde, Yoseph, et al. "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, *COM-28*(1):84–94 (1986).
12. Little, J. and L. Shure. *MATLAB User's Guide*. Massachusetts: The MathWorks, Inc., 1992.
13. Little, J. and L. Shure. *Signal Processing Toolbox: For Use with MATLAB*. Massachusetts: The MathWorks, Inc., 1992.
14. Nagel, Hans-Hellmut. "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intelligence*, *33*:299–324 (1987).
15. Oppenheim, Alan V. and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 1989.
16. Seibert, Michael and Allen M. Waxman. "Adaptive 3-D object recognition from multiple views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2):107–124 (1992).
17. Sekita, Iwao, et al. "Complex autoregressive model for shape recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(4):489–495 (1992).
18. Tou, J. T. and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley Publishing Company, 1974.

Vita

David E. Chelen was born on 10 March 1967 in Pittsburgh, Pennsylvania. As the class Salutatorian, he graduated in 1985 from Greensburg Salem High School, Greensburg, Pennsylvania. Upon graduating from of the United States Air Force Academy in 1989, he received a Bachelor of Science degree in Electrical Engineering. Assigned to the 6510 Test Wing at Edwards AFB, CA, he served as Lead Flight Test Instrumentation Engineer with the Technical Support Instrumentation Systems (TSIS) division from 1989-91. In 1991, he accepted the position of Assistant Deputy of Engineering with the 6512 Test Squadron. Having logged over 200 hours in 7 aircraft types, Captain Chelen was selected as an alternate candidate for the AF Test Pilot School Class-94B.

Captain Chelen married his beautiful wife, Kimberly Ann, on 12 May 1990. The Lord blessed them with a wonderful daughter, Micah Nicole, on 24 August 1992. Upon graduation from AFIT, the Chelen family will reside in Colorado Springs, CO, for an assignment at Peterson AFB.

Permanent address: RD2 Box 273
New Alexandria, Pennsylvania 15670

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704 0188

[illegible]

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Three Dimensional Object Recognition
Using A Complex Autoregressive Model

5. FUNDING NUMBERS

6. AUTHOR(S)
David E. Chelen, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/93D-04

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr John Sjogren
AFOSR/NM 110 Duncan Ave. Room B115
Bolling AFB, D.C. 20332-0001

10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Abstract

Based on an autoregressive model, Complex Partial Correlation (CPARCOR) features are known to provide exceptional Position, Scale, and Rotation Invariant (PSRI) properties for planar 2-Dimensional (2-D) object recognition. Although autoregressive models have been successfully applied to numerous spatio-temporal recognition tasks, the effects of *out-of-plane* image rotations were never considered. This study investigates application of the CPARCOR model to a five class problem of *nonplanar* 2-D views of 3-D objects. Recognition based on CPARCOR features is evaluated using a Template Matching algorithm, two K-Nearest-Neighbor (KNN) classifiers, and a Hidden Markov Model (HMM). Direct comparisons to recognition based on Fourier features are made. Results indicate that the CPARCOR model parameters provide useful shape-features for recognition of *out-of-plane* rotations. Displaying exceptional PSRI properties, the features are shown capable of classification by simple non-adaptive recognition schemes. Relatively successful results are obtained for a variety of tests. The advantage of classification by a *multiple-look* technique over the traditional *single-look* method is clearly demonstrated. Feature space *crowding* is noted as the cause of unusual recognition rates for occluded-view tests. Although general trends are noted, optimal model order and selection of CPARCOR versus Fourier features are considered application dependent.

14. SUBJECT TERMS
Autoregressive Model, Multiple-look Classification, Occluded-view Recognition, KNN Classifiers

15. NUMBER OF PAGES
112

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL