Shape Classification Models

Roger Allan Herbert

MSc by Research in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

December 2000

ASTON UNIVERSITY

Shape Classification Models

Roger Allan Herbert

MSc by Research in Pattern Analysis and Neural Networks, 2000

Thesis Summary

The aim of this project is to compare and contrast the performance of a representative set of boundary based shape classification models using a large and common data set. A range of noisy environments are considered to measure their performances in realistic experimental conditions. The effects on performance, such as the sampling algorithm, model order and classifier are also considered.

Curvature and angle measurement based sampling methods have been shown to perform poorly in adverse conditions. The low order complex autoregressive (CAR) and complex partial correlation (CPARCOR) models have been shown to be robust in all noise conditions on even the most complex data sets considered in this project. The more complex model of a spatially-varying AR process has been shown to be more sensitive to noise than the more simplistic linear AR models. A high order spectral AR model was also tested and showed inferior performance to the low order linear AR models. The simple Fourier descriptors (FD) showed the most robust performances of all but it is believed that they may perform less well when the data sets contain many similar shapes. Finally a wavelet-based model is presented and improvements in the model are suggested.

Keywords: Time Series, AR models, Fourier Descriptors, Moment invariants, Wavelets

Acknowledgements

I would like to thank Dr. Richard Glendinning of the Pattern and Information Processing Group (PIP) at DERA Malvern, firstly for agreeing to supervise the MSc project and secondly for his guidance during the year. I would also like to thank all those within PIP who offered their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Many industrial applications make use of two dimensional shape classifiers, including aircraft identification, classification of chromosomes, industrial part identification and hand-written character recognition. For real-time applications it is desirable to use only the object boundary information because of the reduction in data compared with the original 2D object image. It is for these reasons that the Pattern and Information Processing Group at DERA are interested in exploring the use of such techniques and their feasibility for real applications. This project was carried out as part of Technology Group 10 of the Ministry of Defence (MoD) Corporate Research Programme at DERA Malvern.

The shape classification processes found in the scientific literature consist of four stages: edge extraction, boundary sampling, feature extraction and classification. A wide variety of sampling methods, features and classifiers have been found. The shape information is encoded in the features and common techniques include moment invariants, Fourier descriptors and time series models. All of the boundary based features and classifiers can be computed relatively quickly but with varying degrees of accuracy and robustness for classification i.e. some features will be more separable than others for a particular dataset, but with another dataset this may not hold.

The performance of the aforementioned techniques can be significantly degraded by changes in the statistical characteristics of the imaging noise in cluttered environments. These changes may be caused by varying atmospheric conditions or even the relative effect of discretisation errors. This suggests the need for a range of data sets and noise conditions to test the use of the features for classification.

1.2 Aims of the project

Most of the articles in the literature present models evaluated in ideal conditions (i.e. shapes with noise-free, or very low noise, boundaries) and demonstrate each technique on small sets of well separated data (in feature space). As the classification performance of any model is dependent on the data used, a common thread of data has been accepted by journal referees to benchmark each one. Usually only one or two well separated data sets are used to demonstrate each particular technique. This project aims to compare each technique *as presented* using a wider range of data, including the well separated benchmark data, in similar noise-free conditions and also consider boundary noise experiments to assess the performance and robustness of each one. The boundary noise experiments represent more realistic and challenging tests to the robustness of the techniques. The work will be used to scope further work on this Technology Group 10 project at DERA.

1.3 Overview

Chapter 2 describes how the shape boundaries were extracted from an image, chapter 3 describes the boundary sampling algorithms tested in the experiments and chapter 4 describes the feature extraction techniques employed. Chapter 5 describes the experiments conducted and discusses the results. Finally chapter 6 presents the conclusions drawn from the comparative experiments and describes some on-going and future work.

Chapter 2

Extracting Shape Boundaries from Images

2.1 Introduction

The first task in shape classification, using grey level imagery, is to segment objects from the background whilst maintaining a good representation of their physical shape. By segmenting objects from background noise we create a binary image where the objects are white and the background is black respectively. For high quality imagery where there is a high contrast between object and background, it is a simple task. But when parameters such as brightness, colour and texture are non-uniform, the task becomes more complicated. For the scope of this project, the shapes can be pre-processed so that they are noise free and converted into a binary image. Image segmentation is an active research area and many different approaches exist and are briefly described below.

- *Edge detection.* Edge detectors produce a binary image containing the object boundaries. They operate by indicating significant changes in gradient in the grey levels with a connected unit width line. The main problem with edge detectors is that they tend to produce edge segments which aren't necessarily connected, forming open boundaries. Further processing is required to close the boundaries, and the method chosen for doing this will depend on the particular data set.

- *Segmentation by thresholding.* Many thresholding schemes exist and some standard examples are fixed, multiple and variable thresholds. Perhaps the best known method of automatic fixed threshold setting is the histogram method. The threshold level is chosen to lie in a trough between peaks in a brightness histogram of the image. Problems with thresholding schemes occur in more complex images, where there are large variations in the grey levels assocaiated

with objects and background. For example, an object's intensity could fade due to illumination effects. A global threshold is not usually successful, and so local thresholding schemes are often implemented.

- *Segmentation by pixel classification.* Region growing is an example of this method, where an initial classification of pixels is made and then the resulting groups of pixels are reassessed to see if the classification could be improved by merging groups. This process continues iteratively, gradually refining the membership of each pixel to the different groups. It is quite a complex process and computationally intensive, as each pixel is examined many times.

2.2 Boundary Extraction

For this shape classification study, the shapes were extracted and preprocessed so that they were in a controlled environment. Many of the shapes used were scanned from various articles using a HP scanjet 6100C scanner. The images were then filtered and converted to binary images using the Paint Shop Pro^{TM} software package. The boundary extraction process is not central to the aims of this project so standard methods were used. More sophisticated boundary extraction techniques should be employed to implement this process in a real-world environment.

The edges of the binary images were detected using a Laplacian of a Gaussian [1] edge detector. The edge detector works by convolving the image with a linear Laplacian of a Gaussian filter, producing a closed unit width line around object boundaries. The boundaries are then located and traced around in an anti-clockwise direction, converting the boundary coordinates into chain code format. Chain code is an efficient way of storing boundary information and consists of a stream of numbers describing a shape's boundary. The first two numbers are the cartesian coordinates of the first pixel found when a shape is encountered while performing a raster scan of the whole image. The succeeding numbers consist of a sequence of code numbers, each one representing a boundary step in one of the eight directions relative to the current pixel. The boundary is traced for one complete circuit until the original point is returned to. Figure 2.1 shows an example of some chain code. The columns represent the first four elements of a series of chain codes.

41	41	126	130	142	151
58	154	191	126	207	94
1	2	3	4	3	4
0	0	0	0	0	0

Figure 2.1: First four elements from each chain code

The experiments described in chapter 5 considered many realisations of each shape at different orientations. The binary shapes were rotated by using the MATLAB image processing toolbox function

⟨*imrotate.m*⟩. The boundaries and chain codes were calculated at each orientation and saved so that each boundary could be recovered in an efficient manner. As the experiments were performed many times using the same data sets, the chain code representation greatly increased the processing speed.

Chapter 3

Sampling Points from Template Boundaries

3.1 Introduction

Sampling M points from a boundary consisting of N points in total is widely used in the boundary classification literature. The reason for its use is the reduction in computation required to describe a shape, compared with sampling all of the boundary points. It has been shown [2] that by careful selection of the number of sample points, sampling causes little, if any, reduction in classification performance compared with using all of the points. Sixty four sample points were commonly used in the literature and so sixty four sample points were chosen in our experiments (after confirming the sensitivity of classification performance to the sampling frequency in preliminary tests). In addition to reduced computation, sampling can also serve to smooth the inevitable digitisation noise.

Various sampling techniques exist including equal angle sampling [3], equi-spaced radial distance sampling [4], equi-spaced complex coordinate sampling [5], equi-spaced curvature sampling [2] and equi-spaced polar coordinate sampling [6]. Each sampling scheme has been implemented and used in our comparative experiments. Below is a detailed description of the schemes. Each scheme assumes that the boundaries are closed and are traced around by a boundary following algorithm [3]. In order to sample the boundaries of the binary images used in our experiments, they are traced around using a boundary following algorithm [3].

3.1.1 Equal angle sampling

The radial distance, measured from the centroid of a binary shape to the boundary, is sampled each time the boundary follower crosses any of the N angularly equi-spaced radial vectors. The radial vectors are projected from the centroid of the shape. Figure 3.1 shows the series of radial distances

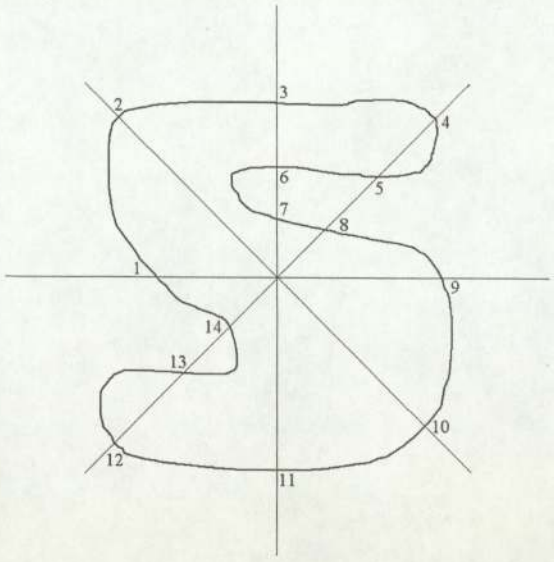

Figure 3.1: Illustration of the equal angle sampling sequence

sampled sequentially from points 1 through to 14, with no reference to the angle at which they were calculated. Note that although there are eight radial vectors in figure 3.1, there are fourteen sample points. As sixty four sample points were specified for our experiments, sixty four radial vectors were used. The precise number of sample points depends on the nature of the boundary (e.g. whether the boundary is convex) using equal angle sampling.

It could be envisaged that sections of a boundary will consist of straight lines and certain rotations of the boundary may cause consecutive points of a line segment to intersect a single radius vector. It is desirable to have just a single sample per radius crossing, as a further rotation of the boundary will cause the radial vector to no longer intersect consecutive points. This would produce samples which are not invariant to the boundary orientation, as different sequences could arise from sampling the same shape at different orientations. To overcome the problem of colinear points, Dubois and Glanz [3] proposed that only one radius vector/boundary intersection be allowed between consecutive boundary points and a particular radius vector. The start point of a sampling sequence is not considered to be important for developing features but the sample values and ordering are, so if the boundaries are sampled in a consistent manner the samples will be invariant to the shape orientation. In our experiments the image was raster scanned until a boundary pixel was encountered.

Equal angle sampling is an extension of an earlier model where the boundary is blindly sampled at all radial intersections i.e. radial vectors are consecutively examined for boundary crossings and sampled at each one; some vectors may have more than one crossing if the boundary is non-convex. This restricted the application to convex shapes only so that each sample is single valued, as multi-valued samples would not be compatible with some feature extraction methods. The equal angle

method allows for non-convex boundaries, so that the sampled sequence is invariant to shifts (as the radial vectors are measured with respect to the shape centroid) and rotations (produces a shifted sample sequence from the original non-rotated sample sequence). However, after sampling by this method the original shape can no longer be reconstructed due to the lack of phase information. This means that the samples could be sampled from different shape boundaries.

3.1.2 Angle of variation sampling: radial distances

This method consists of sampling the shape boundary at N equi-spaced distances, resulting in a polygonal approximation of the original shape. At the i'th vertex of the polygon the distance of the

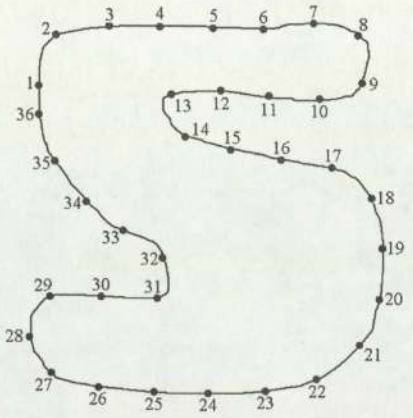

Figure 3.2: Polygonal approximation of boundary by equi-spaced sampling

boundary point i, (x_i, y_i), from the centroid of the object, (x_c, y_c), is computed by:

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad i = 1, \cdots, N \tag{3.1}$$

Figure 3.2 shows 36 equi-spaced samples. Angle of variation sampling is also invariant to the orientation of the shape and is able to sample non-convex boundaries. Again the original boundary cannot be reconstructed from the samples due to a lack of phase information.

3.1.3 Angle of variation sampling: complex coordinates

This method sequentially samples the boundary at equi-spaced distances, again forming a polygonal approximation of the original shape. At each vertex the coordinates of the boundary pixels are sampled with the origin positioned at the centroid of the binary shape. The coordinates are represented as

complex numbers given by equation 3.2.

$$z_i = (x_i - x_c) + j(y_i - y_c) \tag{3.2}$$

This approach circumvents the problems discussed with the previous approaches as they are caused by representing a 2-D time series (a boundary) as a 1-D one i.e. the extra phase information contained in these samples allows the polygonal approximation of the original boundary to be reconstructed.

3.1.4 Angle of variation sampling: curvature measure

Again the boundary is sequentially sampled at equi-spaced distances forming a polygonal approximation. The angular change of a boundary tangent is calculated at each sample point producing a sequence of curvature measures. The curvature of the boundary can be represented by the difference between two consecutive tangents calculated in a window of length w, given by equation 3.3.

$$c_i = tan^{-1}\left(\frac{y_i - y_{i-w}}{x_i - x_{i-w}}\right) - tan^{-1}\left(\frac{y_{i-1} - y_{i-1-w}}{x_{i-1} - x_{i-1-w}}\right) \tag{3.3}$$

This is illustrated in figure 3.3. Again the original shape cannot be reconstructed from the samples

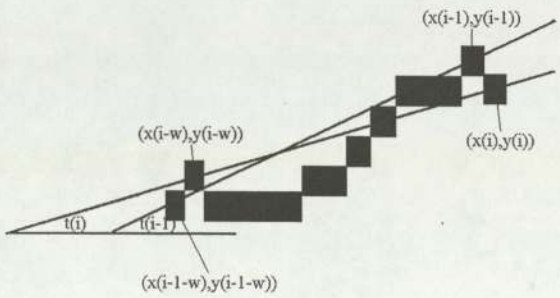

Figure 3.3: Two consecutive tangents of a boundary

due to the lack of phase information. It would be thought that this sampling technique would produce samples that are greatly affected by boundary noise. Curvature sampling has been successfully demonstrated in the literature [2], but was not tested in noisy boundary conditions. It is for this reason that curvature sampling has been included in our experiments.

3.1.5 Angle of variation sampling: polar coordinates

Eom [6] used a sampling method by again sequentially sampling at equi-spaced intervals along the boundary, but computed the polar coordinates of the samples, as given by equation 3.4.

$$\theta_i = tan^{-1}\left(\frac{y_i - y_c}{x_i - x_c}\right),\ r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{3.4}$$

Again, it would be expected that angular measurements would be affected by boundary noise to a much greater degree than radial measurements. As no results have been published using this technique in noisy conditions it is included in our experiments. As these samples contain phase information, the polygonal approximation of the original boundary can be reconstructed.

Chapter 4

Feature Extraction Techniques

4.1 Introduction

A large amount of study has been devoted to two dimensional classification methods based on autoregressive (AR) models, Fourier-based descriptors (FD) and moment invariants. Several approaches based on AR and FD methods are described in [2], along with a wavelet moment-invariant technique. The AR and FD techniques use only information from the sampled boundary, whereas moment invariant techniques use the whole silhouette for feature extraction. The aim of having many types of boundary representations and models from which to derive features is to objectively evaluate the performances of the different approaches.

4.2 Fourier-Based Descriptors

The boundary sampling methods described in chapter 3 represent a boundary in the spatial domain. By simply taking the discrete Fourier transform (DFT) of the samples, a boundary can be represented in the frequency domain by a complex set of numbers, the Fourier descriptors (FD). The lower frequency descriptors describe the general outline of the shape, whereas the higher frequency descriptors describe the smaller details of the shape. The DFT is defined by,

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}} \tag{4.1}$$

where $F(k)$ is the DFT of a periodic complex sequence $x(n)$, $(n = 0, \cdots, N-1)$. The first sample, $F(0)$, is called the DC component, equal to the average of the input series.

The descriptors are translation invariant due to the form of the boundary samples (measurements taken with respect to the shape's centroid), orientation invariant as the absolute values of the FD's are used and scale invariant as the absolute values are normalised.

4.2.1 Curvature Fourier method

The angle of variation method is used to sample the boundary sequence, at each point measuring the curvature. Because the samples are real, the discrete Fourier transform (DFT) of the sample sequence produces Fourier descriptors which are the same in both the positive and negative axes. The curvature Fourier descriptors are then obtained by dividing the absolute values of the positive frequency components by the absolute values of the DC component,

$$x = \left[\frac{|F_1|}{|F_0|}, \cdots, \frac{|F_{N/2}|}{|F_0|}\right]^T \tag{4.2}$$

4.2.2 Radius Fourier method

The radius Fourier method is similar to the curvature Fourier method except that at each sample point, the radial distance measured from the centroid of the shape is computed. As for the curvature Fourier method, the DFT of the sample sequence produces FD's of the form of equation 4.2

4.2.3 Contour Fourier method

The contour Fourier method uses the complex coordinate angle of variation sampling method to samples the complex coordinates at each sample point. The complex coordinates of the shape boundary are directly transformed into the frequency domain by the DFT. As the input is complex, the negative frequency axis is needed, but the DC component depends only on the position of the shape, so it is not needed. The first non-negative frequency component is used to scale the values of the rest of the descriptors, as shown below:

$$x = \left[\frac{|F_{-(\frac{N}{2}-1)}|}{|F_1|}, \cdots, \frac{|F_{-1}|}{|F_1|}, \frac{|F_2|}{|F_1|}, \cdots, \frac{|F_{n/2}|}{|F_1|}\right]^T \tag{4.3}$$

4.2.4 A-Invariant method

In the same manner as for the contour Fourier method, the complex coordinates of the shape boundary are sampled using the angle of variation scheme and are directly transformed into the frequency domain by the DFT. The Fourier coefficients are then normalised (to remove the effects of affine transformations), producing a feature vector of the form:

$$x = \left[|A_{(\frac{N}{2}-1)}|, \cdots, |A_{-2}|, |A_1|, \cdots, |A_{\frac{N}{2}}|\right]^T \tag{4.4}$$

where A_i is given by,

$$A_i = \frac{U_i V_p^* - V_i U_p^*}{U_p V_p^* - V_p U_p^*}, \qquad i = -\left(\frac{N}{2} - 1\right), \cdots, \frac{N}{2}.$$

where U is the DFT of the x components of the boundary coordinates, V is the DFT of the y components of the boundary coordinates and p is a constant (e.g. $p = 1$).

4.3 AR-Based Descriptors

4.3.1 Introduction

Time series models have recently been used for the purpose of shape classification. An autoregressive model expresses each sample in a time series as a linear combination of previous samples plus an error term. Boundary samples can be thought of as an ordered set of data, or time series, and these data can be used in an AR model to produce model coefficients which can be used as features for shape classification.

4.3.2 Radius AR method (1)

Dubois et al [3] continued the work of Kashyap et al [7] using the equal angle sampling method to extract boundary information from shapes. The boundary samples were treated as a time series and an AR model was fitted.

Because boundaries are sampled circularly, the time series are periodic i.e. $r(t) = r(N + t)$ for $t = 1, ..., N$ where, $r(t)$ is a boundary sample at time t and N is the total number of samples describing a closed shape boundary. The general form of an AR-model is given by,

$$r_t = \alpha + \sum_{j=1}^{m} \theta_j r_{t-j} + \sqrt{\beta}\epsilon_t \tag{4.5}$$

where r_t is the current sample at time t from the time series, r_{t-j} is the sample at lag j behind the current sample, θ_j are the model coefficients to be estimated from the observed time series, m is the model order, α is the constant to be estimated, $\sqrt{\beta}\epsilon_t$ is the current error or residual and ϵ_t is a random sequence of independent samples with zero-mean and unit variance.

The variance of ϵ_t is equal to one, therefore $\sqrt{\beta}\epsilon_t$ has a variance of β. So β can be estimated as

$$\beta = \frac{1}{N} \sum_{t=1}^{N} \left(r_t - \alpha - \sum_{j=1}^{m} \theta_j r_{t-j} \right)^2 \tag{4.6}$$

and the unknown parameter, α, can be estimated as

$$\alpha = \bar{r} \left[1 - \sum_{j=1}^{m} \theta_j \right] \tag{4.7}$$

The parameter α is therefore related to the mean radial distance length, $\bar{r}$, and so is indicative of the size of the shape. As $\sqrt{\beta}$ is related to the noise of the radial vector sequence, $\alpha/\sqrt{\beta}$ gives a signal to noise ratio. The coefficients $\{\theta_1, \cdots, \theta_m\}$ model the correlated shape variations and so they describe the shape of the boundary, irrespective of the scale of the shape. The feature vector therefore takes the form of $x = \left[\theta_1, \theta_2, \cdots, \theta_m, \alpha/\sqrt{\beta}\right]^T$.

The model parameters are estimated by fitting the model to the observed time series. The method of least squares is used to find the model parameters that minimise the expected value of the square error [3], as given by equation 4.8.

$$\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \\ \alpha \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{N} r_{t-1}^2 & \cdots & \sum_{t=1}^{N} r_{t-1} r_{t-m} & \sum_{t=1}^{N} r_{t-1} \\ \vdots & \ddots & \vdots & \vdots \\ & & & \sum_{t=1}^{N} r_{t-m} \\ \sum_{t=1}^{N} r_{t-1} & \cdots & \sum_{t=1}^{N} r_{t-m} & N \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{t=1}^{N} r_{t-1} r_t \\ \vdots \\ \sum_{t=1}^{N} r_{t-m} r_t \\ \sum_{t=1}^{N} r_t \end{bmatrix} \tag{4.8}$$

Equation 4.8 is of the form $\mathbf{x} = \mathbf{A^{-1}b}$. In order to compute the inverse of $\mathbf{A}$ for these experiments, Matlab's $\langle inv.m \rangle$ function was used. This uses the method of Gaussian elimination to compute $\mathbf{A}$. If $\mathbf{A}$ is near singular, the algorithm can be susceptible to round-off error. If this occurs a warning message is printed. As no such messages were encountered, as $\mathbf{A}$ was non-singular, $\langle inv.m \rangle$ was used for these experiments.

In practical applications an alternative would be to use $\langle mldivide.m \rangle$, which will solve $\mathbf{x} = \mathbf{A^{-1}b}$ by Cholesky decomposition [8] [9] (assumes $\mathbf{A}$ is symmetric, has positive diagonal elements and is positive definite) which is faster and more numerically stable than using Gaussian elimination.

4.3.3 Curvature AR method

The curvature AR method uses the angle of variation sampling method, computing the curvature function at each sample point. The samples collected from one circuit of a shape boundary gives the time series, which is again assumed to be circular. The AR model coefficients are computed by using the least squares method as shown in equation 4.8. The feature vector is defined as,

$$x = \left[\theta_1, \theta_2, \cdots, \theta_m, \alpha/\sqrt{\beta}\right]^T \tag{4.9}$$

4.3.4 Radius AR method (2)

The radius AR method uses the angle of variation sampling method, computing the radial distance from the centroid of the shape at each sample point. The samples collected from one circuit of the boundary are assumed to be circular. The AR model parameters are estimated by the method of least squares and the feature vector is defined by equation 4.9.

4.3.5 Complex AR method

The complex AR (CAR) method models the complex coordinate sequence, evaluated at each sample point using the angle of variation sampling method. The sequence is treated as a circular one so that,

$x_j = x_{j+N}$, $y_j = y_{j+N}$ and $z_j = x_j + iy_j = z_{j+N}$. The complex AR model is an extension of the real AR model,

$$z_j = \sum_{k=1}^{m} a_k z_{j-k} + \epsilon_j^f \tag{4.10}$$

where ϵ_j^f is a random sequence of complex samples with zero mean and unit variance. The complex AR coefficients are estimated by least squares error [5] so that the mean squares error $\epsilon^2(m, \boldsymbol{a})$ is minimised,

$$\epsilon^2(m, \boldsymbol{a}) = E_j \|\epsilon_j^f\|^2 = \overline{\boldsymbol{a}}^T R(m)\boldsymbol{a} - \overline{\boldsymbol{r}}^T \boldsymbol{a} - \overline{\boldsymbol{a}}^T + r_0 \tag{4.11}$$

where $\boldsymbol{a} = [a_1, \cdots, a_m]^T$, $\boldsymbol{r} = [r_1, \cdots, r_m]^T$, $r_k = E_j(z_j \overline{z}_{j-k})$ and

$$R(m) = \begin{bmatrix} r_0 & \overline{r}_1, & \overline{r}_2, & \cdots, & \overline{r}_{m-1} \\ r_1, & r_0, & \overline{r}_1, & \cdots, & \overline{r}_{m-2}, \\ r_2, & r_1, & r_0, & \cdots, & \overline{r}_{m-3} \\ \cdots, & \cdots, & \cdots, & \cdots, & \cdots \\ r_{m-1}, & r_{m-2}, & r_{m-3}, & \cdots, & r_0 \end{bmatrix} \tag{4.12}$$

The CAR coefficients $\{a_k\}_{k=1}^m$ minimising the mean square error are obtained by,

$$\boldsymbol{a} = R^{-1}\boldsymbol{r} \tag{4.13}$$

The real and imaginary parts of the complex AR coefficients are used as separate features, as shown in equation 4.14 below:

$$x = [re(\theta_1), im(\theta_1), \cdots, re(\theta_m), im(\theta_m)]^T \tag{4.14}$$

If the shape is rotated by an angle θ, then the complex autocorrelation coefficients become $r_k^{Rotated} = E_j\{e^{i\theta} z_j . e^{-i\theta} z_{j-k}\} = r_k$, so $R(m)$ is rotation invariant. Therefore from equation 4.13, it can be seen that the CAR model coefficients are also invariant to rotations of the shape. By the definition of complex autocorrelation coefficients, they are also independent of the initial sampling point on the boundary. The CAR coefficients are also invariant to shifts as the boundary samples are measured from the centroid of the shape.

To make the CAR coefficients invariant to the scale of a shape the number of boundary sample points should be kept constant. If the size of the shape changes by a factor ρ, each of the boundary samples becomes ρz_j. Therefore r_k, R and $\boldsymbol{r}$ become $\rho^2 r_k$, $\rho^2 R$ and $\rho^2 \boldsymbol{r}$ respectively. The CAR coefficients are then obtained from $(\rho^2 R(m))^{-1}(\rho^2 \boldsymbol{r}) = R^{-1}(m)\boldsymbol{r}$.

4.3.6 CPARCOR method

Real partial correlation (PARCOR) coefficients are often seen to be more useful in speech signal processing than real AR coefficients [5]. A real PARCOR coefficient of order m is defined as a correlation coefficient between forward and backward prediction errors in the real AR model of order

$(m-1)$.

Extending this theory, complex partial correlation (CPARCOR) coefficients, p_m, of order m can be viewed as a complex correlation coefficient between forward prediction errors $\{\epsilon_t^f(m-1)\}$ (complex form of prediction error in equation 4.5) and backward ones $\{\epsilon_t^b\}$ in the CAR model of order $(m-1)$:

$$p_m := \frac{E_t\left(\epsilon_t^f(m-1)\epsilon_{t-m}^{-b}(m-1)\right)}{\{E_t(\epsilon_t^f(m-1)\epsilon_t^{-f}(m-1))E_t(\epsilon_{t-m}^b(m-1)\epsilon_{j-m}^{-b}(m-1))\}^{\frac{1}{2}}} \tag{4.15}$$

where,

$$\epsilon_t^b = z_t - \sum_{k=1}^{m-1} b_k z_{j+k}$$

The CPARCOR coefficient p_m is the same as the CAR coefficient a_m of the CAR model of order m. CPARCOR coefficients $\{p_k\}_{k=1}^{m-1}$ of the CAR model of order $(m-1)$ are the same as those of order m because only one CPARCOR coefficient p_m is obtained from the model of order m. A recursive algorithm for computing CAR and CPARCOR coefficients is presented in [5] that is faster than the least squares solution. To denote order explicitly, notations such as $\boldsymbol{a}(m)$ and $\boldsymbol{r}(m)$ are used to denote $\boldsymbol{a}$ and $\boldsymbol{r}$.

The CAR coefficient a_1 of the CAR model of order 1 is given by,

$$\boldsymbol{a}(1) = r_1/r_0 \tag{4.16}$$

For model orders greater than two, $R(m)$ is given by,

$$R(m) = \begin{bmatrix} R(m-1), & \overline{\boldsymbol{r}}(m-1)^\dagger \\ \boldsymbol{r}(m-1)^\sharp, & r_0 \end{bmatrix} \tag{4.17}$$

where $\boldsymbol{r}(m-1)^\dagger$ and $\boldsymbol{r}(m-1)^\sharp$ represent the reversed order elements of $\boldsymbol{r}(m-1)$ and the transpose of $\boldsymbol{r}(m-1)^\dagger$, respectively. The CAR coefficients of order m is given by,

$$\boldsymbol{a}(m) = \begin{bmatrix} \boldsymbol{a}(m-1) - \boldsymbol{a}(m-1)^\dagger p_m \\ p_m \end{bmatrix} \tag{4.18}$$

and the CPARCOR coefficient of order m is shown to be,

$$p_m = (r_m - \boldsymbol{r}(m-1)^\sharp \boldsymbol{a}(m-1))/(r_0 - \boldsymbol{r}(m-1)^T \overline{\boldsymbol{a}}(m-1)) \tag{4.19}$$

By recursively computing equations 4.16 to 4.19 the CAR and CPARCOR coefficients can be determined using the fast algorithm. Similarly for the CAR coefficients, the CPARCOR coefficients are complex valued and the real and imaginary parts are used as features as in equation 4.14.

4.3.7 A Spectral AR Model

The spectral AR method of Eom [6] models the polar coordinate sequence evaluated at each sample point using the angle of variation sampling method. The series of samples collected from one complete

circuit of a shape's boundary is again assumed to be a circular one. The angle sequence is treated as a separate sequence to that of the radial distance sequence and linear AR models are fitted to both sequences by the method of least squares, as shown in equation 4.8. Applying the z-transform to both sides of equation 4.5 gives:

$$A(z)R(z) = \alpha + \sqrt{\beta}\epsilon(z), \tag{4.20}$$

where $R(z)$ and $\epsilon(z)$ are z-transforms of $r(t)$ and ϵ respectively. $A(z)$ is given by,

$$A(z) = 1 - \theta_1 z^{-1} - \cdots - \theta_m z^{-m}. \tag{4.21}$$

The general order power spectral density $S_r(\omega)$ of the process $r(t)$ is estimated by [10]:

$$S_r(\omega) = \frac{\hat{\beta}}{A(e^{j\omega})A(e^{-j\omega})} \tag{4.22}$$

Spectral analysis has been used to estimate the spectrum of stationary stochastic processes [11]. Many stationary stochastic processes can be approximated by a sufficiently high order AR process. Eom used an AR(20) model to demonstrate this model. This approach tends to give a smooth spectrum and it can also pick out narrow peaks in the spectrum [11], provided that the AR process is of a high order. Eom [6] states that the roots of $A(z)$ are related to spectral peaks and can therefore be used as features for classification. As the roots of $A(z)$ are real, the roots will be complex and occur in conjugate pairs. The roots of the AR polynomials $A(z)$ for the set of radial distances, $r(t)$, and the set of angles, $\theta(t)$, are estimated independently to form the feature vectors, $r_1, \cdots, r_p$ and $s_1, \cdots, s_p$ respectively. The DC components of the power spectral densities, given by the real roots of the AR polynomials, are a function of the average length and angle of the series respectively. Each complex root has a corresponding conjugate, and only one root out of the conjugate pair is needed in the feature vector, so those roots with positive imaginary components are retained and rearranged by angular frequency. The feature vector is then formed by combining the two sets of ordered complex roots. Therefore we have p complex features from $2p$ complex roots. By treating the real and imaginary components as independent real numbers, the feature vector will consist of $2p$ real numbers, $x_1, \cdots, x_{2p}$.

Finding all of the roots of a polynomial is a classical problem in numerical analysis and has been widely studied [9]. The root finding algorithm $\langle roots.m \rangle$ was used to compute the roots of the polynomials. The algorithm calculates the eigenvalues of a companion matrix $\mathbf{A}$, where the eigenvalues are the roots of the characteristic polynomial $P(x) = det[\mathbf{A} - \mathbf{xI}]$. The characteristic polynomial of

the special $m \times m$ companion matrix,

$$\mathbf{A} = \begin{pmatrix} -\frac{a_m}{a_{m+1}} & -\frac{a_{m-1}}{a_{m+1}} & \cdots & \frac{a_2}{a_{m+1}} & -\frac{a_1}{a_{m+1}} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \tag{4.23}$$

is equivalent to the general polynomial,

$$P(x) = \sum_{i=1}^{m+1} a_i x^{i-1}, \tag{4.24}$$

see [9, ch. 9.5] for a more detailed description.

4.3.8 Non-stationary AR Modelling

Paulik et al [4] observed that there is no reason to assume that shape boundaries are produced by a stationary AR model. Cases where complicated boundaries are very similar and need to be classified separately, consideration of the possible non-stationary character might provide more accurate boundary modelling, leading to improved classification.

This method [4] represents the radial distance boundary samples, sampled with the radial angle of variation method, as a non-stationary random process which is the output of a spatially varying circular autoregressive (SVCAR) linear system whose model coefficients can be expressed as a truncated function expansion. The features derived from this scheme are invariant to scaling, translation and rotation. A SVCAR model can be expressed as:

$$r(n) = \sum_{i=1}^{M} a_i(t) r(t-i) + \sqrt{\beta} w(t) \tag{4.25}$$

$$s(t) = r(t) + b(t) \tag{4.26}$$

where M is the model order, $\sqrt{\beta}w(t)$ is a random noise sequence with zero mean and variance β, $s(t)$ is the original radial vector sequence, $b(t)$ is a non-stationary mean and $r(t)$ is a zero mean sequence formed by subtracting $b(t)$ from $s(t)$.

The spatially varying parameters, $a_i(t)$ and $b(t)$ are expressed as a finite sum of known time functions. Due to the circular nature of the boundary samples the discrete Fourier series (DFS) basis, given by equations 4.27 and 4.28, is shown to be optimal by Paulik et al [4]

$$\hat{a}_i(n) = \sum_{k=0}^{q} a_{ik} e^{(j2\pi kn/N)} \tag{4.27}$$

$$\hat{b}(n) = \sum_{k=0}^{q} b_k e^{(j2\pi kn/N)} \tag{4.28}$$

The point at which the sampling of a given shape is started affects the coefficients b_k and a_{ik}. This can occur if the boundary sampling is started at the uppermost point in an image frame but the shape has been rotated from it's original angle. Given an unshifted boundary sequence, $s(n)$, and a shifted boundary sequence, $s'(n)$ shifted by Δ (i.e. $s'(n) = s(n + \Delta)$), then $s'(n) = r'(n) + b'(n)$. It has been shown [4] that $b'_k = b_k e^{(j2\pi k\Delta)}$ and $a'_{ik} = a_{ik}e^{(j2\pi kn\Delta/N)}$, where $k = 0, \cdots, q$. Using the DFS as the basis set, the harmonic amplitudes for the sets of coefficients, (b_k, a_{ik}) and (b'_k, a'_{ik}), are identical and the phases only differ by $2\pi k\Delta/N$. Out of all the possible basis sets possible, only the DFS allows a simple relationship between the model coefficients estimated from shifted versions of the same sequence.

Using DFS basis functions, the parameters are estimated as a prediction error minimisation problem. To estimate the non-stationary mean, $b(n)$, the mean approximation coefficients must be computed first. As the DFS basis is an orthogonal family, we have

$$\hat{b}_n = \sum_{k=0}^{q} b_k e^{(j2\pi kn)/N} \tag{4.29}$$

$$b_k = \frac{1}{N} \sum_{n=0}^{N-1} s(n) e^{(-j2\pi kn)/N} \tag{4.30}$$

A real regeneration formula for $\hat{b}_n$ may be derived by using the properties of DFS coefficients for real sequences, given by

$$\hat{b}(n) = b_0 + 2\sum_{k=1}^{q} \left(c_k cos\left(\frac{2\pi kn}{N}\right) - d_k sin\left(\frac{2\pi kn}{N}\right)\right) \tag{4.31}$$

where c_k and d_k are the real and imaginary components of 4.30. The stationary mean, $r(t)$, can then be calculated using equation 4.26 and the coefficients, a_{ik}, may be estimated. The estimator of $r(n)$ is given by

$$\hat{r}(n) = \sum_{i=1}^{M} \left(\sum_{k=0}^{q} a_{ik} e^{(j2\pi kn/N)} \right) r(n-i) \tag{4.32}$$

If ω=2π/N, then

$$Y_i(n) = [r(n-i), e^{j\omega n} r(n-i), e^{j\omega 2n} r(n-i), \cdots, e^{j\omega qn} r(n-i)]^T, \tag{4.33}$$

$$\phi(n) = [Y_1(n)^T, Y_2(n)^T, \cdots, Y_M(n)^T]^T, \tag{4.34}$$

$$\theta = [a_{10}, \cdots, a_{1q}, a_{20}, \cdots, a_{2q}, \cdots, a_{M0}, \cdots, a_{Mq}]^T, \tag{4.35}$$

$$S = [\phi(1), \phi(2), \cdots, \phi(N)]^T, \tag{4.36}$$

and

$$-\alpha = [r(1), r(2), \cdots, r(N)]^T. \tag{4.37}$$

The prediction error is then given by

$$\epsilon = S\hat{\theta} + \alpha \tag{4.38}$$

where $\hat{\theta}$ is an array of estimates of SVCAR time-invariant modelling coefficients. Minimising the squared prediction error, $\epsilon^T\epsilon$, gives

$$S^T S\hat{\theta} = -S^T\alpha \tag{4.39}$$

The least squares estimate of θ then gives

$$\hat{\theta} = (S^T S)^{-1}(-S^T\alpha) \tag{4.40}$$

Finally the residual variance is given by

$$\hat{\beta} = \frac{1}{N}\sum_{n=1}^{N}(r(n) - \phi(n)^T\hat{\theta})^2. \tag{4.41}$$

If the boundary sequence is described by the SVCAR model, then the following features are invariant to translation, rotation (starting from same sample point) and scaling: the model coefficients, a_{ik}, and the spatially variant mean coefficients, b_k, scaled by the residual standard deviation, $\sqrt{\beta}$, giving $b_k/\sqrt{\beta}$, where $0 \leq k \leq q$.

To make the features invariant to rotation starting from any sample point, as it is desirable for classification, an optimal shape matching algorithm which determines the degree of shift (Δ) between two shape boundary sequences can be used [4]. A performance index which compares the SVCAR parameters for the original and shifted starting point sequences is given by:

$$J(\hat{\Delta}) = \sum_{k=1}^{q}\left\{\left|b_k - e^{-j\omega k\hat{\Delta}}b'_k\right|^2 + \sum_{i=1}^{M}\left|a_{ik} - e^{-j\omega k\hat{\Delta}}a'_{ik}\right|^2\right\} \tag{4.42}$$

where $\hat{\Delta}$ is the estimated sequence shift. The primed coefficients relating to the shifted sequence are multiplied by a phase shift term, so when the estimated shift corresponds to the actual shift the primed terms cancel the corresponding unprimed terms. To find the estimated sequence shift, $J(\hat{\Delta})$, equation 4.42 is differentiated with respect to Δ, giving

$$\frac{\delta J}{\delta\Delta} = \sum_{k=1}^{q} 2\omega k\left\{sin(\omega k\hat{\Delta})\left[B_k cos\psi_k + \sum_{i=1}^{M}A_{ik}cos\lambda_{ik}\right] - cos(\omega k\hat{\Delta})\left[B_k sin\psi_k + \sum_{i=1}^{M}A_{ik}sin\lambda_{ik}\right]\right\} = 0 \tag{4.43}$$

A modified version of 4.43 is given by 4.44

$$f(\hat{\Delta}) = \sum_{K=1}^{q}(g_k sin(\omega k\hat{\Delta}) - h_k cos(\omega k\hat{\Delta})) \tag{4.44}$$

where,

$$g_k = \left[k\,Re\left((b_k b_k^{*'}) + \sum_{i=1}^{M}(a_{ik}^* a'_{ik})\right)\right] \tag{4.45}$$

$$h_k = \left[k\,Im\left((b_k b_k^{*'}) + \sum_{i=1}^{M}(a_{ik}^* a'_{ik})\right)\right] \tag{4.46}$$

Equation 4.44 is a simple sum of sinusoids, with q harmonics and therefore a finite number of zeros. The maximum frequency in 4.42 is q/N, so 4.44 will have only one zero crossing in any half-period,

N/(2q). Once all of the zeros have been found the objective function $J(\hat{\Delta})$ can be evaluated and the minimum point chosen as $\hat{\Delta}$. The feature vector can then be multiplied by $exp(-j2\pi k\hat{\Delta}/N)$ prior to classification. This feature vector is then fully rotation invariant.

4.4 A Wavelet Shape Descriptor

4.4.1 Introduction

Hu [12] first presented a paper using a moment based model to recognise shape silhouettes in 2D images. Since then many adaptations have been published in the literature. Prokop and Reeves [13] presented a survey of many of the pertinent techniques. Geometrical moments of an image are determined by integrating the image function over space, so that the image can be uniquely determined by its geometrical moments of all orders. Low order moments can be used to classify significantly different shapes as the moments are designed to capture global information about an image. However if the shapes are similar or are corrupted by noise, the techniques are thought to give poor classification [14]. Shen and Ip [15] have used wavelet moment invariants to capture global and local information from the shape of interest in an image and showed the performance improvement over other moment invariant shape classification techniques.

4.4.2 General moment invariants

If $f(x,y)$ represents a binary image using cartesian coordinates, then $f(r,\theta)$ represents an image in polar coordinates, where $x = r\,cos\theta$ and $y = r\,sin\theta$. A regular moment can be defined by equation 4.47

$$m_{pq} = \int\int x^p y^q f(x,y)dxdy \tag{4.47}$$

Translation invariance is achieved by computing the centroid of the shape and shifting it to the centre of the image. Scale invariance is achieved by the use of a scale factor, $\alpha = \sqrt{m_{00}/E[m_{oo}]}$. The coordinates are then scaled by $(x/\alpha, y/\alpha)$. Representing equation 4.47 in the polar domain,

$$F_{pq} = \int\int f(r,\theta)g_p(r)e^{jq\theta}r\,dr\,d\theta \tag{4.48}$$

where F_{pq} is the pq-order moment, $g_p(r)$ is some function of radial distance r and p and q are integers. If the image $f(r,\theta)$ is rotated by an angle β, then the moment becomes $F_{pq}^{Rotated} = F_{pq}e^{jq\beta}$. As $\|F_{pq}^{Rotated}\| = \sqrt{F_{pq}^{Rotated}(F_{pq}^{Rotated})^*} = \|F_{pq}\|$, the moments are rotation invariant.

In order to reduce the feature extraction process from a 2D to a 1D problem, equation 4.48 is rewritten as

$$F_{pq} = \int S_q(r)g_p(r)r\,dr, \tag{4.49}$$

where $S_q(r) = \int f(r,\theta)e^{jq\theta}$, so $S_q(r)$ is now a 1D sequence dependent on r. The form of $g_p(r)$ determines the type of moment invariant model. If $g_p(r)$ is defined across the whole range of r, as for

most geometric moment invariants, then F_{pq} is a global feature, whereas if $g_p(r)$ is defined locally then F_{pq} is a local feature. This drove the development of the wavelet moment invariant, as the wavelet transform is able to provide both position and scale localisation.

4.4.3 Wavelet moment invariants

If $g_p(r)$ is replaced with $\psi_{mn}(r)$, where $\psi_{mn}(r)$ is a wavelet basis function defined along a radial axis at any orientation and $m = 0, 1, 2, 3$, $n = 0, 1, \cdots, 2^{m+1}$ and $q = 0, 1, 2, 3$ then equation 4.49 becomes

$$\|F_{mnq}^{wavelet}\| = \| \int S_q(r) \cdot \psi_{mn}(r) r \, dr \|. \tag{4.50}$$

The 'well-known' wavelet basis function family can be represented as,

$$\psi^{a,b}(r) = \frac{1}{\sqrt{a}} \psi\left(\frac{r-b}{a}\right), \tag{4.51}$$

where a is a scale parameter and b is a shifting parameter. We used a biorthogonal B-spline mother wavelet as it is close to the forms of other polynomial moments and its use was demonstrated by Shen and Ip [15]. See figure 4.1 for the shape of the mother and father wavelets. The scale parameter is

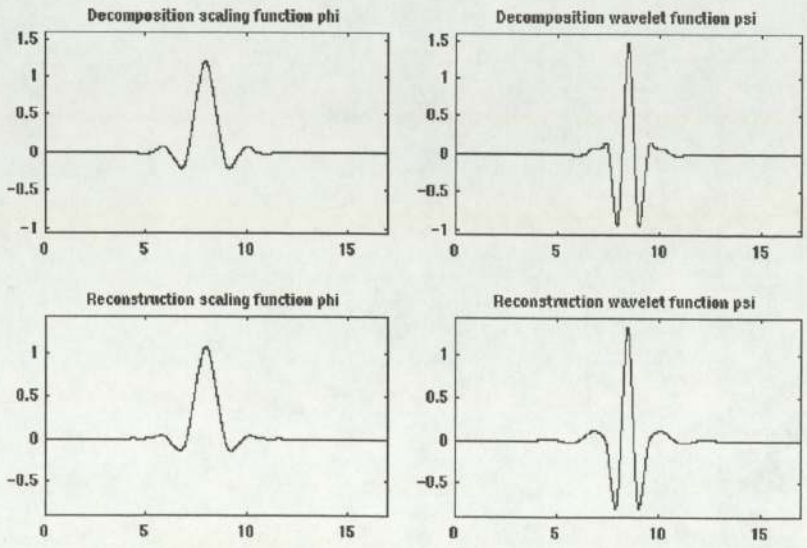

Figure 4.1: Mother and father biorthogonal wavelets

chosen so that $a = a_0^m$, where m is an integer and the shift parameter is chosen so that $\psi(r - b/a)$

traverses the whole range of r, given by $b = nb_0a_0^m$. The parameter r ranges between 0 and 1, so a_0 and b_0 are set to equal 0.5 and m and n are constrained as shown in equation 4.52,

$$\begin{aligned} a &= a_0^m, \qquad m = 0, 1, 2, 3 \\ b &= nb_0a_0^m, \quad n = 0, 1, \cdots, 2^{m+1} \end{aligned} \tag{4.52}$$

So the wavelet defined along a radial axis is given by,

$$\psi_{m,n}(r) = 2^{m/2}\psi(2^m r - 0.5n) \tag{4.53}$$

As there are many more feature coefficients for this model than for the previous models, Shen and Ip [15] suggested the use of a feature selection algorithm to identify the most discriminative ones. It is desirable for discriminative features to have a small intraclass variance and a large interclass separation. Shen and Ip demonstrated the following univariate method which is easily implemented and was sufficient for their experiments.

If the mean of each feature $\|F_{m,n,q}^{wavelet}\|$ for shape S_i is $m(S_i, \|F_{m,n,q}^{wavelet}\|)$ and the standard deviation is $\sigma(S_i, \|F_{m,n,q}^{wavelet}\|)$, estimated from NTS training samples, then the inverse of the between-to within-class variance ratio is given by,

$$Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j) = \frac{\eta(\sigma(S_i, \|F_{m,n,q}^{wavelet}\|) + \sigma(S_j, \|F_{m,n,q}^{wavelet}\|))}{m(S_i, \|F_{m,n,q}^{wavelet}\|) - m(S_j, \|F_{m,n,q}^{wavelet}\|)}, \quad \eta = 3.0 \tag{4.54}$$

$Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j)$ indicates the effectiveness of feature $\|F_{m,n,q}^{wavelet}\|$ to discriminate between shapes S_i and S_j. If the features are assumed to be normally distributed then the probability of a class conditional variable occurring between $[m-3*\sigma, m+3*\sigma]$ is 99.8%. So the smaller $Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j)$ is, the greater the ability of feature $\|F_{m,n,q}^{wavelet}\|$ to discriminate between shapes S_i and S_j. If $Q < 1$, then $\|F_{m,n,q}^{wavelet}\|$ is almost certain to be able to discriminate between S_i and S_j. The feature selection algorithm computes $Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j)$ for each feature over all N_{class} classes and selects those features with $Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j) < 1$. A discrimination number, $NDD(\|F_{m,n,q}^{wavelet}\|)$, reflects the feature's ability to discriminate between pairs of classes,

$$NDD(\|F_{m,n,q}^{wavelet}) = \sum_{i=1}^{N_{class}} \sum_{j=1, j\neq i}^{N_{class}} w(Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j)) \tag{4.55}$$

where $w(x) = 1$ if $x < 1$, else $w(x) = 0$. For each feature the worst overall discriminative measure is calculated by,

$$Q^{worst}(\|F_{m,n,q}^{wavelet}\|) = max_{1\leq i,j\leq N_{class}, i\neq j}\{Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j) \cdot w(Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j))\} \tag{4.56}$$

from the set $Q(\|F_{m,n,q}^{wavelet}\|, S_i, S_j) < 1, 1 \leq, j \leq N_{class}$ and $i \neq j$. Figure 4.2 illustrates the selection process.

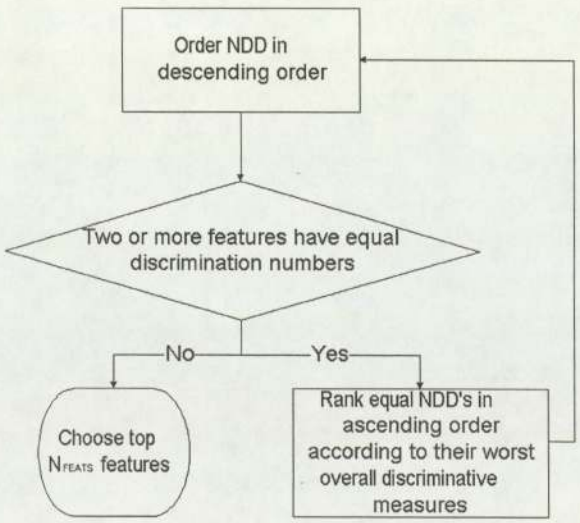

Figure 4.2: Feature selection process

Chapter 5

Comparative Experiments

5.1 Introduction

An experimental comparison of the two dimensional shape classification techniques described in chapter 4 is carried out. The requirement for such a comparison is driven by the small and selective data sets found in the literature describing the techniques. Table 5.1 summarises some of the previous work in the area and the experiments conducted on each type.

Author	Model	Sampling method	Classifier	Data
Dubois et al [3]	AR	Equal angle	Gaussian kNN	letters machine parts
Kartikeyan et al [16]	Non-linear quadratic Volterra model	Angle of variation: radius	Gaussian	aeroplanes
Paulik et al [4]	Non-stationary AR	Angle of variation: radius	kNN	industrial parts lakes aeroplanes
Sekita et al [5]	Complex AR CPARCOR	Angle of variation: radius	Gaussian	machine parts leaves
Kauppinen et al [2]	FD's CAR CPARCOR	Angle of variation: - radius - contour - curvature	kNN	letters machine parts aeroplanes
Antoine et al [17]	Wavelet coefficients	Angle of variation: complex coords	No classification fractal analysis	corkscrew Koch curve
Eom et al [6]	Spectral AR	Angle of variation: polar coords	MLP	aeroplanes numbers machine parts
Shen et al [15]	Wavelet moment invariant	uses whole silhouette	Nearest neighbour	letters synthetic shapes machine parts

Table 5.1: Selective summary of previous work

The experiments attempt to address several issues that will have an effect on the classification performance:

- *Boundary sampling algorithm.* Various boundary sampling algorithms have been used in the shape classification literature and a representative selection have been implemented for comparison in this project.

- *Model Type.* A range of models, that produce feature vectors for use in classification, are evaluated and include AR models, Fourier descriptor models and a wavelet moment-invariant based model.

- *Model order.* To investigate the effect of order on linear AR models, a range of orders from one to ten were computed for each experiment. Standard time series theory would suggest the use of a model order selection algorithm to determine the best order, such as Akaike's information criterion (AIC), but Paulik [4] made the observation that the determination of the best order is an imprecise problem. The main reason being that each shape class constitutes a different set of processes and will therefore have different optimal orders. Additionally, in order to compare features they should be of the same order.

- *Classifier.* A Euclidian distance based k-nearest neighbour (kNN) classifier, where k=3, is compared with the performance of a simple statistical Gaussian classifer (with a full covariance matrix) and a multilayer perceptron (MLP) classifier (where appropriate).

- *Effect of noise.* To evaluate the usefulness of each technique to any real world application, two noise types and levels are added, as boundary noise is inevitable. The noise distributions used were spherical about each boundary sample, namely Gaussian and t-distributions. The experiments included training the models on noisy data and testing on noisy data, and also training on non-noisy data and testing on noisy data.

- *Data type.* A range of data was drawn from a variety of papers to fairly evaluate each technique. Figures 5.1 - 5.4 show the data sets used. The AR, CAR, CPARCOR and FD's were tested over the complete range of data, whereas the spectral and wavelet moment-invariant techniques were tested on data sets B and D, with the non-stationary technique tested on sets B and E. Data set C was chosen as it is a simple, well separated, set of shapes commonly found in the journal papers and so any further work should use them as benchmark tests. Data set B was chosen because the shapes are slightly more complex as some have non-convex boundaries. Both sets B and C consist of relatively simple shapes and are included in the experiments primarily because they have been used in many journal papers and so any further tests should use them as benchmark tests. Set D consists of an array of industrial parts, representing shapes which might be encountered in a real application. Set BCD consists of all of the shapes from sets B, C and D, and was used to investigate the effect of the number of classes on the classification performance of each technique. Set E consists of fifteen shapes, with shapes E13, E14 and E15

all being very similar. The other twelve shapes in set E are a mixture of shapes from sets B and D. It was thought that set E should be used to investigate the performance of each technique to discriminate between similar classes.

During the experiments the scale of the silhouettes was kept constant as scale invariance of the features has already been demonstrated both theoretically and empirically. It was also decided not to consider silhouette occlusion experiments due to time limitations but are none the less important and should be considered for further work.

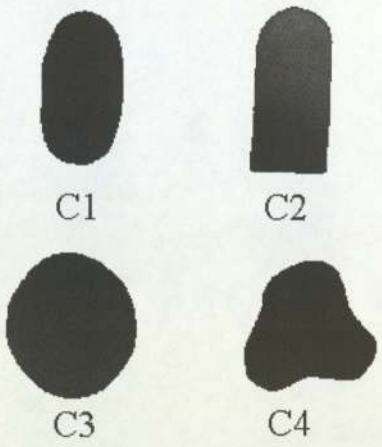

Figure 5.1: Data Set C

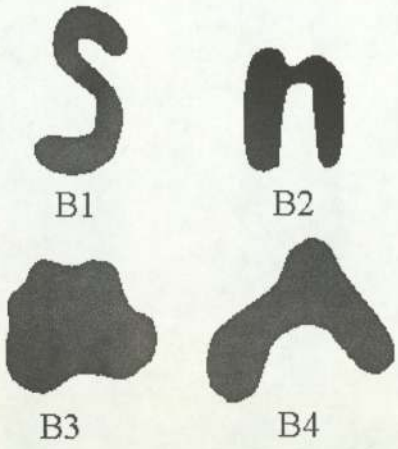

Figure 5.2: Data Set B

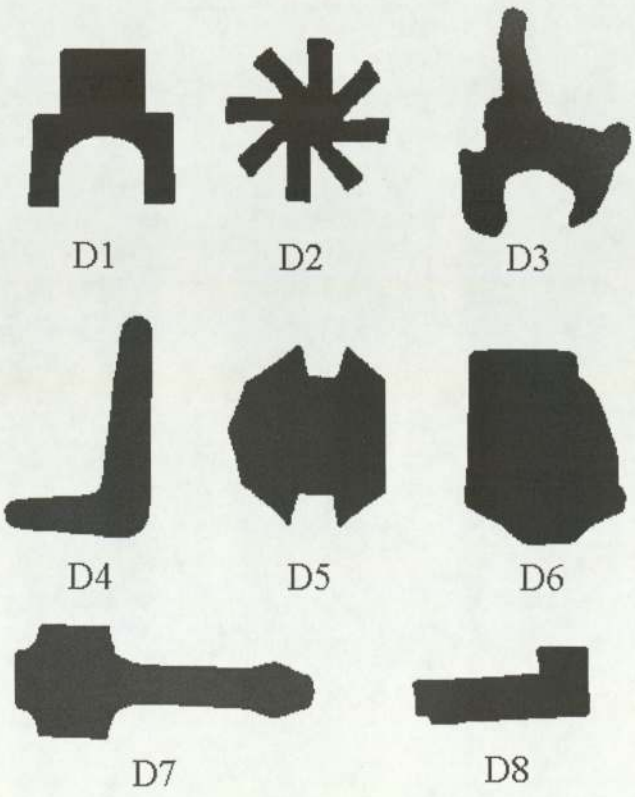

Figure 5.3: Data Set D

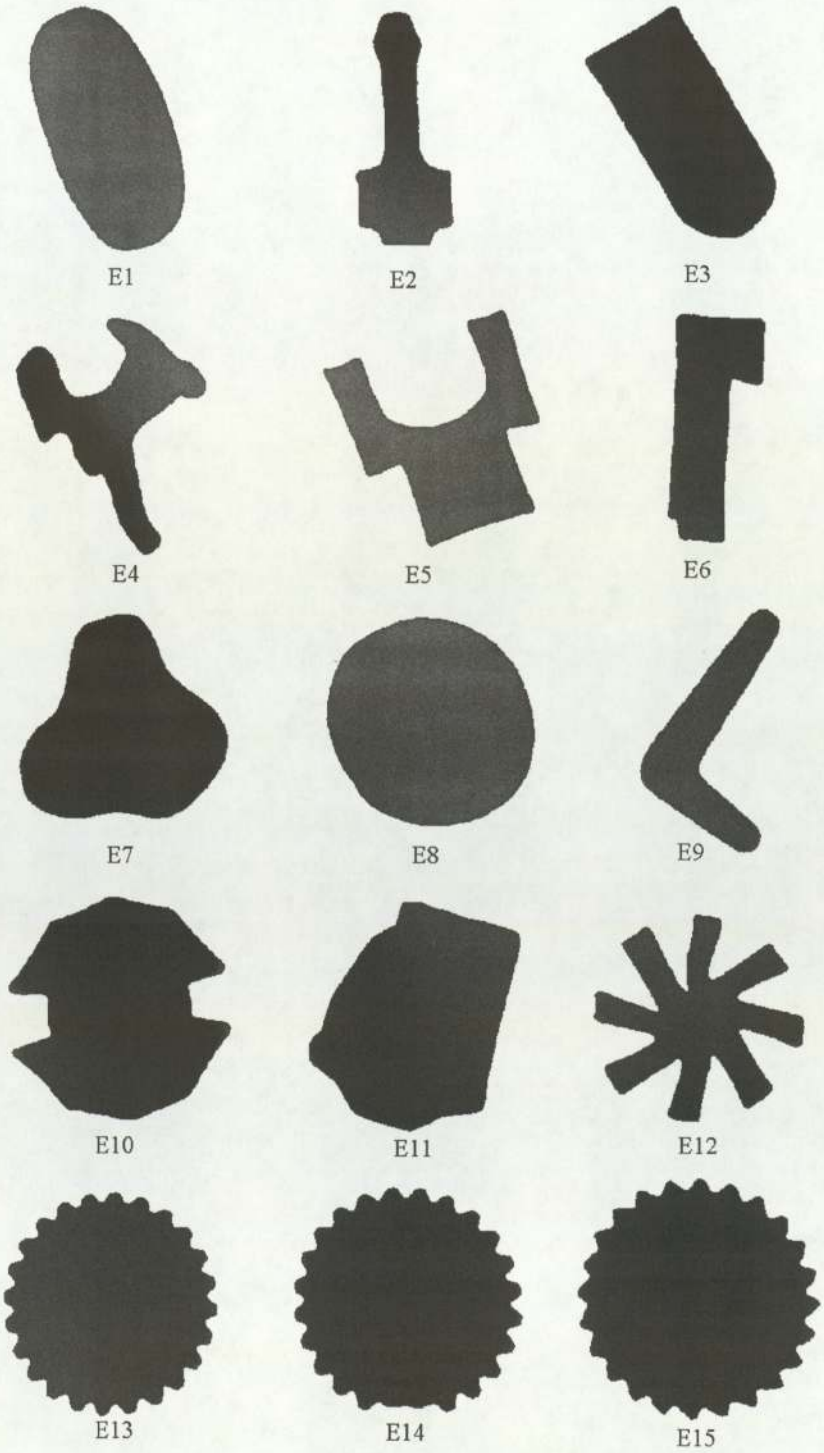

Figure 5.4: Data Set E

5.2 Methodology

The same process was used to conduct all of the experiments and is described below and illustrated in figure 5.5. Firstly a silhouette of each shape was scanned-in and the silhouette boundaries traversed to extract the chain code representation. Chain codes store boundary information in an efficient manner (see chapter 2). The scale of each shape was kept constant as scale invariance has been demonstrated both theoretically and empirically by each of the authors of the techniques being considered. Translation invariance is not an issue as all of the features are measured with respect to the centroid of each shape. The chain code extraction process consisted of keeping a shape in a central position within an image and rotating it using the MATLAB function $\langle imrotate.m \rangle$, at each rotation computing the chain code. After computing the chain code of each shape, the boundaries were consecutively redrawn and any desired synthetic noise was added prior to sampling. The boundary samples were then used to compute a feature vector which was assigned a class label and saved. The next realisation of the shape was then sampled and the process repeated for all of its orientations being considered. The process was repeated for all of the shapes resulting in a full set of features. Hold-out tests were performed throughout the experiments, where the feature vector was divided into separate training and testing sets.

An exception to the algorithm in figure 5.5 is the wavelet moment invariant technique, as it filled-in the closed boundary and extracted its features directly from the resulting silhouette, so no boundary sampling was necessary. After the feature extraction stage, the classifiers can be used to classify the labelled data.

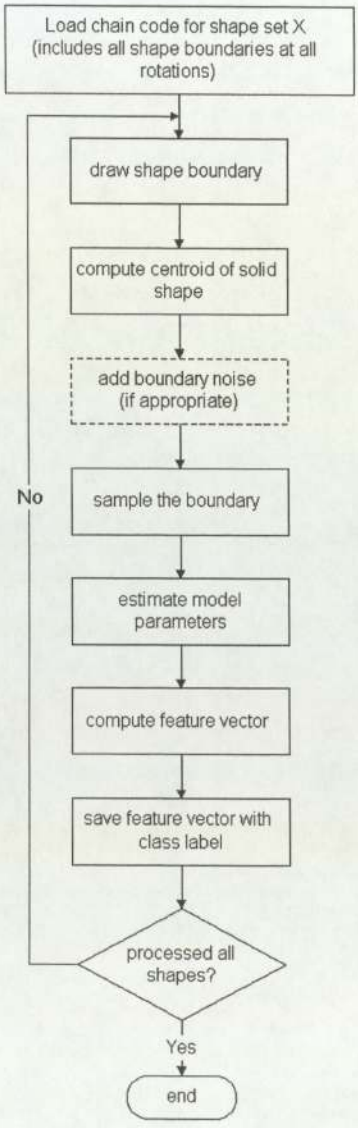

Figure 5.5: Extracting features from boundaries

Some of the experiments considered the effect of random noise on the shape boundaries. The noise was generated by adding random noise to each sample point, (x_j, y_j), on the boundary,

$$\tilde{x}_j = x_j + dr\ cos(\theta), \quad \tilde{y}_j = y_j + dr\ sin(\theta) \tag{5.1}$$

where d is the mean distance between each sampling point, r is a sample from either a Gaussian or a t-distribution and θ is a sample from a uniform distribution, producing spherically symmetric noise. Examples of noisy boundaries are shown in figures 5.6 and 5.7 with noise levels $N(0, 0.5^2)$ and $t_{40}(0, 1)$.

This noise model does not attempt to precisely mimic real-world noise which is not always possible to model without prior knowledge, but is used to model the broad characteristics of a wide range of real world noise generating mechanisms. The model has been used and demonstrated on small sets of data in two journal papers [5] [2].

A random seed was used in these experiments as we are interested in ensemble comparisons. If interest centers on the performance of particular noise realisations then of course the seed should be kept constant.

Figure 5.6: Gaussian noise added to shape B1

Figure 5.7: t-distributed noise added to shape B1

5.3 Experiments and Results

The experiments were conducted in order to complete the comparisons of average correct classification (ACC) performances between the models described in chapter 4. It was intended to closely reproduce techniques found in journal publications so that a fair and more thorough comparison of techniques could be conducted. It was then attempted to identify potential weaknesses and suggest further improvements. All of the experiments were conducted using MATLAB on a Digital alpha station 500 computer. The full set of results are presented in figures 5.8 - 5.198. The AR model graphs plot ACC against model order. The Fourier descriptor graphs plot ACC against the data set used. The x-axis indices index the data sets displayed in the title in order.

The following subsections are structured by firstly presenting a brief description of each technique together with a table describing the relevant experimental conditions. The columns showing the size of the train and test sets represent the number of samples for training and testing per class. The noise column shows the amount and type of noise, if any, added to the training and test sets. The dashes indicate that no noise was added to the boundaries. A general student-t distribution is denoted by, $t_\upsilon(\mu, \sigma)$, where υ is the number of degrees of freedom, μ is the mean and σ is the variance of the distribution. Each experiment is cross-referenced to a corresponding figure presenting the results, which are positioned at the end of each subsection. Following the table is the analysis, which is structured so that the performance of each classifier is discussed separately. The classifiers include a kNN classifier, where k=3 and uses a Euclidian distance metric, a Gaussian classifier with a full covariance matrix (see appendix A) and a multi-layer perceptron (MLP) which is used for the spectral AR experiments only.

The experiments involved testing each classifier on features where:

- the models are fitted to noise free boundaries
- the test and train features were both fitted to boundaries containing noise of the same type
- the training features were fitted to noise free boundaries and the test features were fitted to boundaries containing noise.

Boundary noise can occur in real life for a number of reasons, for example, changing environmental conditions effecting the imaging of shapes and also partial occlusion. The relative noise levels could be unpredictable and not easily modelled, without complete prior knowledge. For this reason, the tests where the models are trained on noise free boundaries and tested on noisy boundaries measures the robustness of each model when the amount, and type, of noise is unknown.

5.3.1 Radius AR method (1)

Introduction

The radius AR method 1 technique of subsection 4.3.2 uses the equal angle sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The equal angle sampling technique measures the radial from the centroid of the shape to the boundary when the boundary follower crosses any of N angularly equi-spaced radial vectors. As a result the number of samples from a shape at different orientations can vary. The boundary samples are treated as circular time series and AR models of orders 1 to 10 are fitted to each series and labelled according to class. All of the linear AR experiments solved the autoregression by the method of least squares. The inverse of the matrix of autocovariance values in equation 4.8 was computed using $\langle inv.m \rangle$, which calculates the inverse by Gaussian elimination. The determinants of several of these matrices were computed at random to check that the they were non-singular. An alternative to $\langle inv.m \rangle$ would be to use the left divide function which will be more stable if the matrix is near singular, as it calculates the inverse in a different manner.

In these experiments we used 64 equi-spaced radial vectors, as increasing the number of samples further had little or no effect on the classification performance. The feature vectors from each class were split into testing and training data sets and a Gaussian and kNN classifier are compared. Table 5.2 describes the different tests performed using the AR method 1 technique.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Radius AR method 1	Set C	10	15	kNN	-/-	5.8
Radius AR method 1	Set C	40	20	kNN	-/-	5.9
Radius AR method 1	Set B	10	15	kNN	-/-	5.10
Radius AR method 1	Set B	40	20	kNN	-/-	5.11
Radius AR method 1	Set D	25	25	kNN	-/-	5.12
Radius AR method 1	Set D	50	30	kNN	-/-	5.13
Radius AR method 1	Set BCD	25	25	kNN	-/-	5.14
Radius AR method 1	Set BCD	50	30	kNN	-/-	5.15
Radius AR method 1	Set E	20	20	kNN	-/-	5.16
Radius AR method 1	Set E	50	30	kNN	-/-	5.17
Radius AR method 1	Set C	10	15	RCS	-/-	5.18
Radius AR method 1	Set C	40	20	RCS	-/-	5.19
Radius AR method 1	Set B	10	15	RCS	-/-	5.20
Radius AR method 1	Set B	40	20	RCS	-/-	5.21
Radius AR method 1	Set D	25	25	RCS	-/-	5.22
Radius AR method 1	Set D	50	30	RCS	-/-	5.23
Radius AR method 1	Set BCD	25	25	RCS	-/-	5.24
Radius AR method 1	Set BCD	50	30	RCS	-/-	5.25
Radius AR method 1	Set E	20	20	RCS	-/-	5.26
Radius AR method 1	Set E	50	30	RCS	-/-	5.27

Table 5.2: AR method 1 experiments

kNN classifier

Figures 5.8 - 5.17 show the performance of the radius AR method using a kNN classifier.

- No noise on boundaries

  The tests on set C show there are no marked improvements in the results with extra training data. The tests on set B show a fall in the average correct classification performance from those on set C, averaging 80% (figure 5.11). It is thought that this is due to the more complex nature of set B, as shapes B1 and B2 have non-convex boundaries.

  Shape set D contains eight classes, but the shapes are visually quite different from one another. With twenty five samples in the test and training sets (see figure 5.12), the performance is lower than for the equivalent test on set C, as there are twice as many classes in set D. With fifty samples in the training set, set D is classified $\simeq$ 100% correctly for all model orders. Data set BCD has sixteen classes and as figures 5.14 and 5.15 show, with enough training data the shapes are correctly classified in excess of 99% for all model orders.

  Shape set E was thought to provide a difficult test to the classifiers, as Paulik et al [4] reported that the linear AR method performed poorly on data sets of a non-stationary nature, such as data set E. Shapes E13 - E15 are visually very similar, so the difficulty lies in the model being able to distinguish between these similar shapes, whilst also being able to distinguish between the more dissimilar shapes. Figure 5.16 shows this to be the case with a low number of training samples. Paulik et al quoted the best performance of this technique to give an average correct classification of 80.33%, using twenty training and 20 testing samples, with a model of order two. Figure 5.16 shows that if the amount of training data is increased the ACC performance is close to 100% for all model orders.

Gaussian classifier

Figures 5.18 - 5.27 show the performance of the radius AR method using a Gaussian classifier.

- No noise on boundaries

  Applying the Gaussian classifier to the same experiments, the average correct classification performances are generally better than the equivalent tests using a kNN classifier for low amounts of training data, but worse at higher model orders ($m = 7 - 10$), where the covariance matrix in the classifier can become singular. An illustration of the effect of increasing the training set size on the potential performance of the linear AR method 1 is best shown in figures 5.26 and 5.27. Figure 5.20 duplicates the results of [3] and also [5]. The Gaussian classifier has shown the ability to classify all of the data sets 100% correctly, which is surprising given the justification of the use of more complex models due to the lack of performance of linear models such as this. Data sets

B, C, and D are particularly well classified at most model orders considered. These data sets are commonly found in journal publications and so are important for benchmarking each technique. As the results are not particularly interesting the results of the succeeding experiments on these data sets are presented in appendix B and are referenced in the experimental description tables in this section.

Summary

The AR method 1 technique has the ability to classify all of the shape sets completely correctly. From reading the literature [4], this is a surprising result as the more complex AR techniques have been demonstrated on sets such as set E and show that linear models are not as good. Another surprising result is that none of the experiments in table 5.2 used many training samples and if more are used the results improve significantly.

It is has been shown that the Gaussian classifier provides the better performance as long as enough training data is used so the covariance matrix can be estimated reasonably accurately. If the training set is too small then the covariance matrix can become close to singular, producing spurious results, as shown in figure 5.26. It is also the most consistent classifier as the kNN could not classify set B more than 85% correctly, whereas the Gaussian classifier classified set B 100% correctly for most model orders.

As expected, shape set E was the most difficult set to classify with low amounts of training data. Set BCD has more classes but the shapes are visually quite different, whereas set E contains a mix of both visually similar and dissimilar shapes.

No noise tests were conducted using the equal angle sampling technique as the AR method (2), shown in the next subsection, showed far better ACC levels in noise-free conditions and so was used to represent the linear AR model in the presence of noise using radial distance sampling.

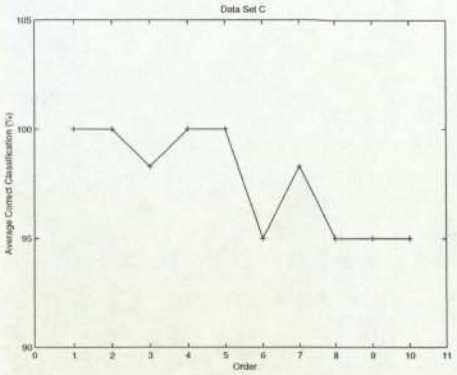

Figure 5.8: Radius AR:1 (kNN), Set C

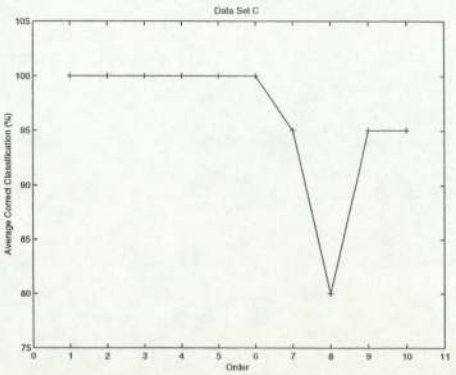

Figure 5.9: Radius AR:1 (kNN), Set C

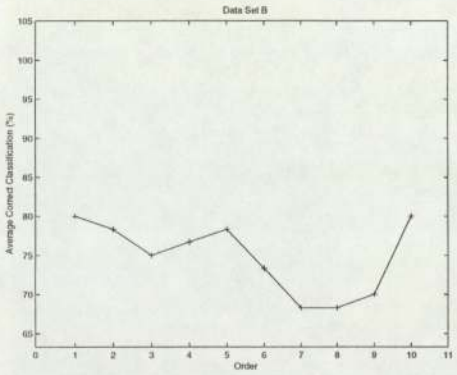

Figure 5.10: Radius AR:1 (kNN), Set B

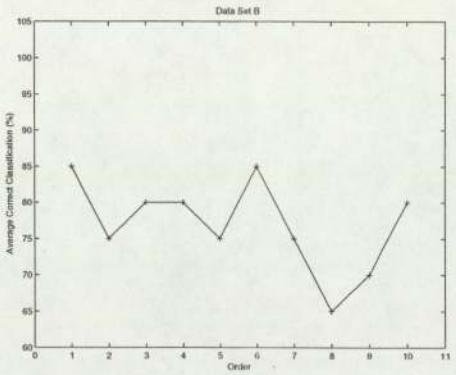

Figure 5.11: Radius AR:1 (kNN), Set B

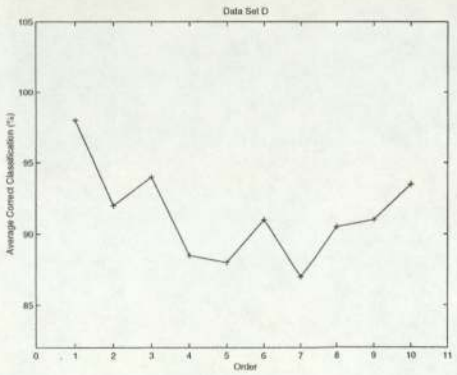

Figure 5.12: Radius AR:1(kNN), Set D

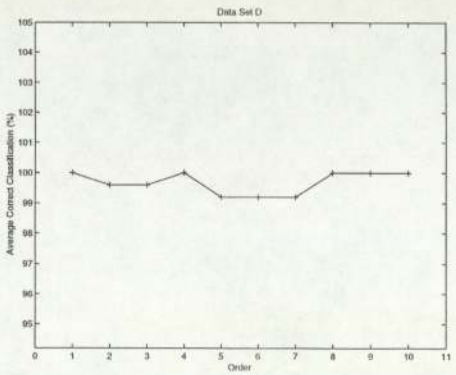

Figure 5.13: Radius AR:1 (kNN), Set D

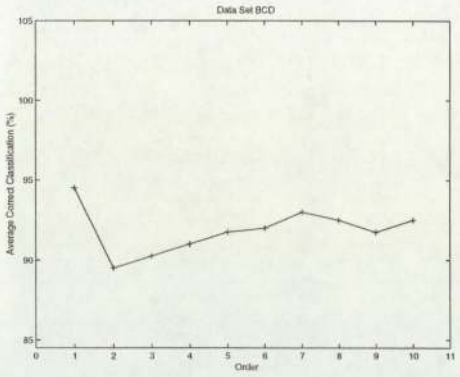

Figure 5.14: Radius AR:1 (kNN), Set BCD

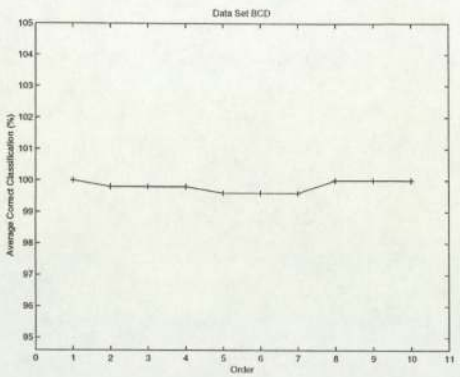

Figure 5.15: Radius AR:1 (kNN), Set BCD

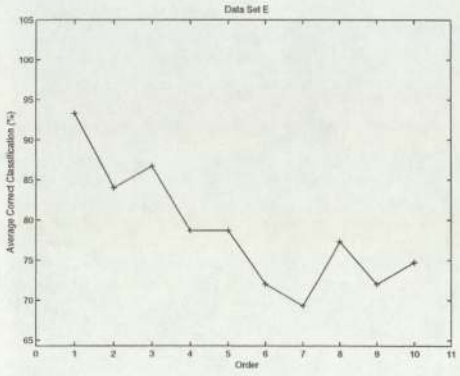

Figure 5.16: Radius AR:1 (kNN), Set E

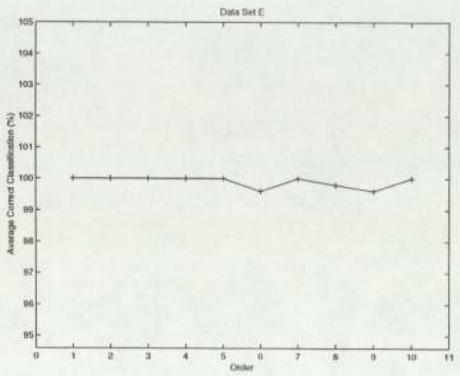

Figure 5.17: Radius AR:1 (kNN), Set E

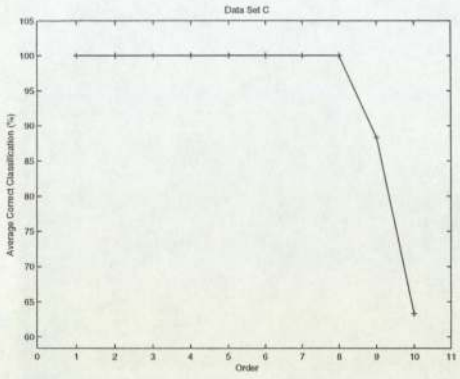

Figure 5.18: Radius AR:1 (RCS), Set C

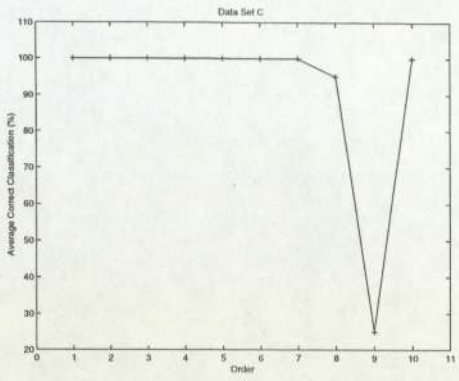

Figure 5.19: Radius AR:1 (RCS), Set C

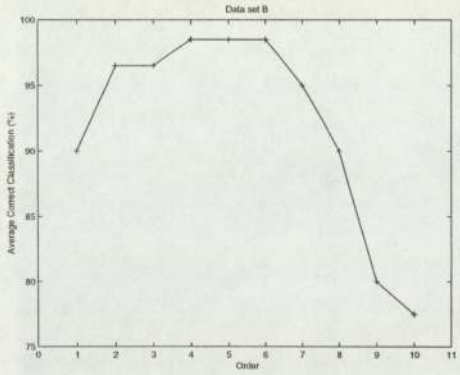

Figure 5.20: Radius AR:1 (RCS), Set B

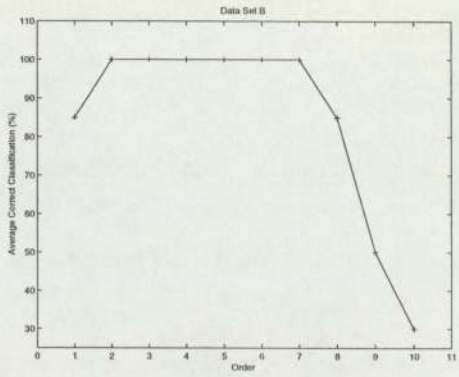

Figure 5.21: Radius AR:1 (RCS), Set B

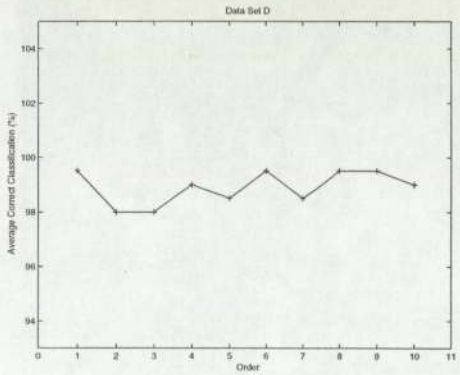

Figure 5.22: Radius AR:1(RCS), Set D

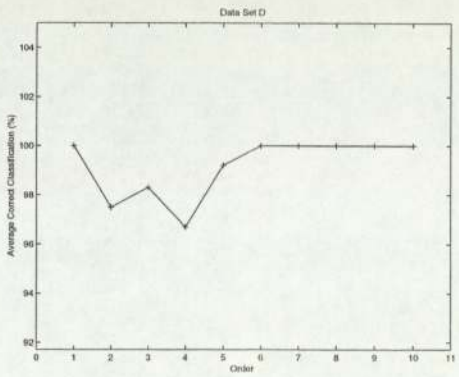

Figure 5.23: Radius AR:1 (RCS), Set D

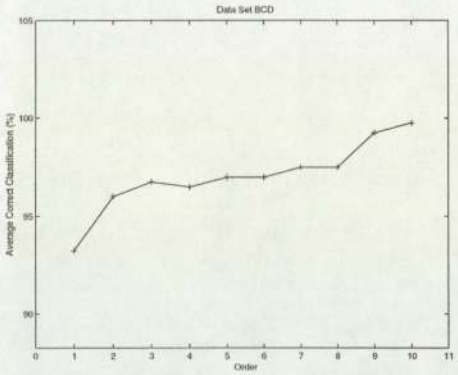

Figure 5.24: Radius AR:1 (RCS), Set BCD

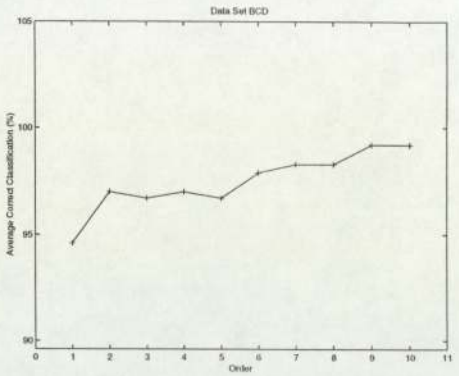

Figure 5.25: Radius AR:1 (RCS), Set BCD

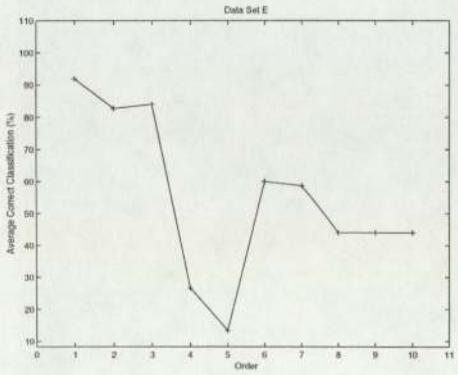

Figure 5.26: Radius AR:1 (RCS), Set E

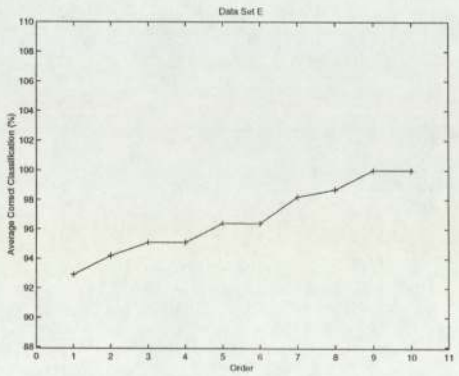

Figure 5.27: Radius AR:1 (RCS), Set E

5.3.2 Radius AR method (2)

Introduction

The radius AR method 2 of subsection 4.3.4 uses the radial angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. Recall that the radial angle of variation technique measures the radial distance from the shape centroid to the boundary at N equi-spaced positions around the boundary. This provides uniform coverage of the boundary, irrespective of the shape's orientation, resulting in a constant number of samples. The boundary samples are treated as circular time series and AR models of orders 1 to 10 are fitted to each series and labelled according to class.

In these experiments we again used 64 spatially equi-distant samples as any more samples had little or no effect on the average classification performance, which is also consistent with [2]. This sampling method samples the radial distances from the centroid to the boundary at equi-spaced intervals, providing spatially uniform coverage of the boundary. As the sampling is spatially uniform, a constant number and an equal distribution of samples are sampled from the boundary, independent of the shape's orientation. The features from each class were split into test and train data sets and a Gaussian and kNN classifier are compared. Table 5.3 shows the tests performed.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Radius AR method 2	Set B	10	15	kNN	-/-	B.1
Radius AR method 2	Set B	40	20	kNN	-/-	B.2
Radius AR method 2	Set D	25	25	kNN	-/-	B.3
Radius AR method 2	Set D	50	30	kNN	-/-	B.4
Radius AR method 2	Set BCD	25	25	kNN	-/-	5.31
Radius AR method 2	Set BCD	50	30	kNN	-/-	5.32
Radius AR method 2	Set E	20	20	kNN	-/-	5.33
Radius AR method 2	Set E	50	30	kNN	-/-	5.34
Radius AR method 2	Set B	10	15	RCS	-/-	B.5
Radius AR method 2	Set B	40	20	RCS	-/-	B.6
Radius AR method 2	Set D	25	25	RCS	-/-	B.7
Radius AR method 2	Set D	50	30	RCS	-/-	B.8
Radius AR method 2	Set BCD	25	25	RCS	-/-	5.35
Radius AR method 2	Set BCD	50	30	RCS	-/-	5.36
Radius AR method 2	Set E	20	20	RCS	-/-	5.37
Radius AR method 2	Set E	50	30	RCS	-/-	5.38
Radius AR method 2	Set B	40	20	kNN	$N(0,1)/N(0,1)$	5.39
Radius AR method 2	Set B	40	20	kNN	$N(0,1)/N(0,1)$	5.40
Radius AR method 2	Set B	40	20	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.41
Radius AR method 2	Set B	40	20	kNN	$t_4(0,1)/t_4(0,1)$	5.42
Radius AR method 2	Set D	50	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.43
Radius AR method 2	Set D	50	30	kNN	$N(0,1)/N(0,1)$	5.44
Radius AR method 2	Set D	50	30	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.45
Radius AR method 2	Set D	50	30	kNN	$t_4(0,1)/t_4(0,1)$	5.46
Radius AR method 2	Set B	40	20	RCS	$N(0,0.52)/N(0,0.52)$	5.47
Radius AR method 2	Set B	40	20	RCS	$N(0,1)/N(0,1)$	5.48
Radius AR method 2	Set B	40	20	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.49
Radius AR method 2	Set B	40	20	RCS	$t_4(0,1)/t_4(0,1)$	5.50
Radius AR method 2	Set D	50	30	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.51
Radius AR method 2	Set D	50	30	RCS	$N(0,1)/N(0,1)$	5.52
Radius AR method 2	Set D	50	30	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.53
Radius AR method 2	Set D	50	30	RCS	$t_4(0,1)/t_4(0,1)$	5.54
Radius AR method 2	Set B	40	20	RCS	$-/N(0,0.5^2)$	5.55
Radius AR method 2	Set B	40	20	RCS	$-/N(0,1)$	5.56
Radius AR method 2	Set B	40	20	RCS	$-/t_{40}(0,1)$	5.57
Radius AR method 2	Set B	40	20	RCS	$-/t_4(0,1)$	5.58
Radius AR method 2	Set D	50	30	RCS	$-/N(0,0.5^2)$	5.59
Radius AR method 2	Set D	50	30	RCS	$-/N(0,1)$	5.60
Radius AR method 2	Set D	50	30	RCS	$-/t_{40}(0,1)$	5.61
Radius AR method 2	Set D	50	30	RCS	$-/t_4(0,1)$	5.62

Table 5.3: AR method 2 experiments

kNN classifier

- No noise on boundaries (all data sets)

  Figures B.1 - 5.34 show the performance of the technique using the kNN classifier. A general observation is the high ($\simeq$ 100%) average correct classification at model order one for all of the data sets, except E. A surprising trend can be seen across all data sets: as the model order increases the performance falls slightly. This is surprising as the performance would intuitively be expected to improve with a higher order model, as it is a more complex model. Increasing the amount of training data does not have an effect on the performances using sets B and D, but increases the performances on the more complex sets BCD and E by 5-10%.

- Noise on boundaries of test and train sets (sets B and D)

  The average correct classification levels decreased from those in the equaivalent noise free experiments, see figures 5.39 - 5.46. The effect of the noise on the classification of the different order models appears to be fairly random. The effect of the noise on the ACC levels on comparable experiments using sets B and D affected the experiments using set D more. This would be expected as we have more classes in set D, possibly resulting in a denser feature space. Intuitively the noise would be expected to increase the variance of the AR parameter coefficient estimates. As set B contains only four classes and set D contains eight classes it might be expected that there would be increased overlap of the coefficients from each class in the feature space of set D. Figure 5.28 shows the first three principal components of the AR(4) coefficients fitted to data set B. Each of the classes are represented by the different plotting symbols. Figure 5.29 shows the equivalent plot when the noise level is drawn from a $N(0, 0.5^2)$ distribution. Note that the features are less tightly clustered in figure 5.29 than in figure 5.28, but are still separable (full separation can only be seen by rotating the figure).

  The equivalent plot for data set D, figure 5.30, shows eight separable clusters (again the separability is clearer if the figure is rotated), although more tightly packed in feature space than in figure 5.28 (as there are more classes).

Gaussian classifier

- No noise on boundaries (all data sets)

  The performance of the Gaussian classifier in equivalent test conditions is shown in figures B.5 - 5.38 and reveals a high level of classification ($\simeq$ 100%) across all model orders and data sets. This is not surprising as the clusters in figures 5.28-5.30 look quite Gaussian. Even with a model order of 1 the average correct classification performance is $\simeq$ 99%, as is illustrated in figure 5.37.

- Noise on boundaries of test and train sets (sets B and D)

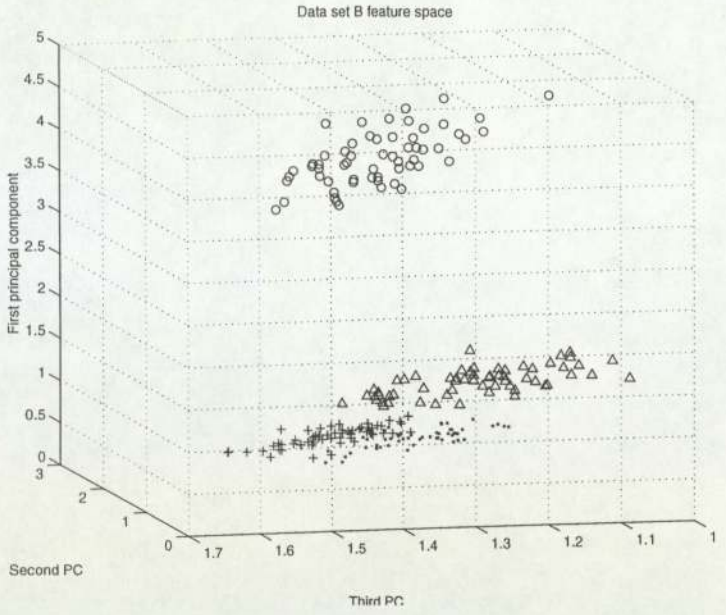

Figure 5.28: Scatter plot of the first 3 principal components of the AR(4) feature vector fitted to set B (no boundary noise). The dots represent shape B1, the crosses represent shape B2, the circles represent B3 and the triangles represent shape B4.

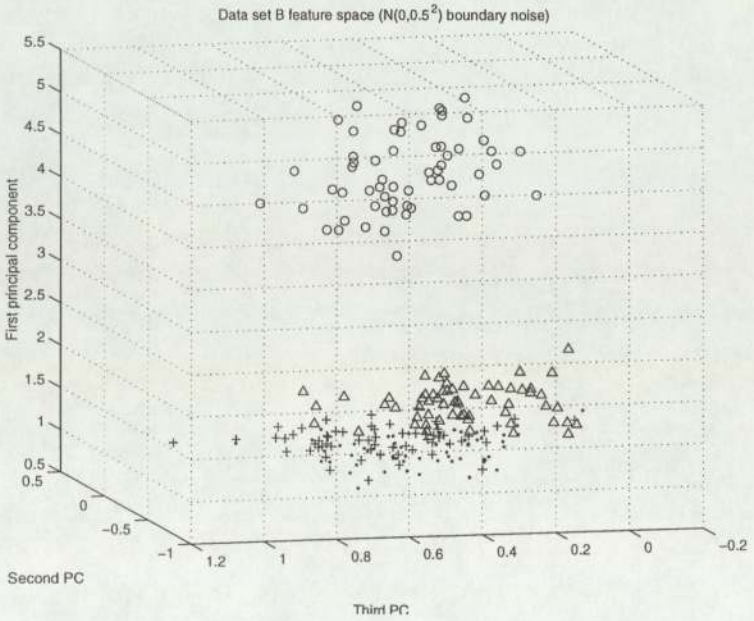

Figure 5.29: Scatter plot of the first 3 principal components of the AR(4) feature vector fitted to set B ($N(0, 0.5^2)$ boundary noise). The dots represent shape B1, the crosses represent shape B2, the circles represent shape B3 and the triangles represent shape B4.

When the Gaussian classifier was applied to sets B and D, see figures 5.47 - 5.54, the results were less affected by the addition of boundary noise than the equivalent kNN classifier tests, presumably because the disperse features can be approximated by a Gaussian distribution. The results also improve with higher model orders, which would be expected.

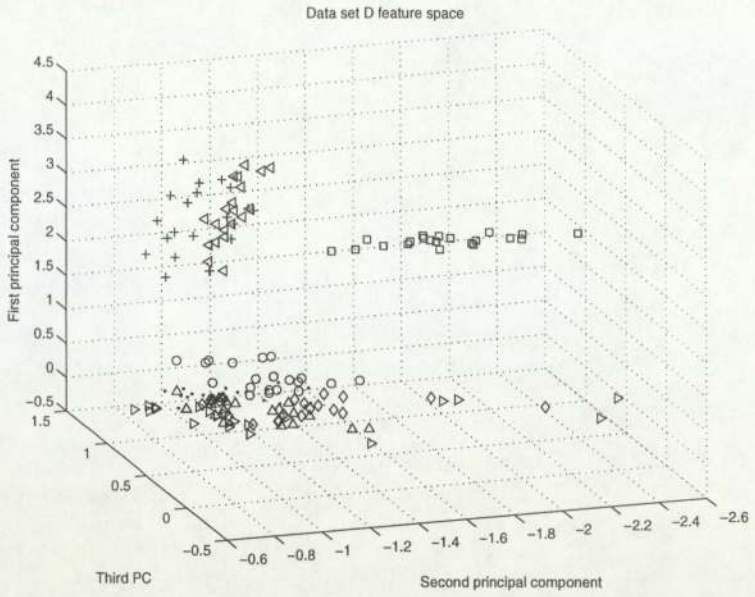

Figure 5.30: Scatter plot of the first 3 principal components of the AR(4) feature vector fitted to set D (no boundary noise). The dots represent shape D1, the crosses represent shape D2, the circles represent shape D3, the triangles (up) represent shape D4, the triangles (left) represent shape D5, the triangles (right) represent shape D6, the squares represent shape D7 and the diamonds represent shape D8.

The Gaussian classifier appears to be the more robust classifier for these experiments.

- Noise on boundaries of test set only (sets B and D)

  Further noise experiments included training on 'non-noisy' data and testing on noisy data using a Gaussian classifier, see figures 5.55 - 5.62. The results are generally lower than the training and testing on noisy data experiments. This is because the class means and covariances of the features will be slightly different in the noisy feature space from those in the 'non-noisy' feature space, depending on the amount and type of boundary noise.

Summary

The tests on noise free boundary data sets show that the Gaussian classifier achieves the better ACC performance on every test, achieving 100% ACC on all data sets with enough training data to allow a reasonable estimation of the covariance matrix. This is a surprising result as Paulik et al presented much lower results for the linear AR method 1 using data set E, as he did not use enough training data. This also shows that the full potential of the radial AR method 1 has not been demonstrated as the radial angle of variation technique has been shown to be superior to the equal angle sampling method.

When noise was added to the boundary of the test and train sets the Gaussian classifier was shown to be more robust than the kNN classifier for all of the data sets (see figures 5.47 - 5.54). The ACC

improved with increasing model order. Very high levels of ACC ($> 95\%$) were achieved on sets B and D at every noise level attempted.

The comparable experiments where the training set is noise free and the testing set has noise added to the boundaries, show that the performance drops considerably. This is because, if for example, the test and train sets have a similar level and type of noise added to the boundaries then their estimated class means and covariances will be similar in both sets. If the test set has noise added to the boundary and the train set contains no boundary noise, then the class means and covariances will be slightly different.

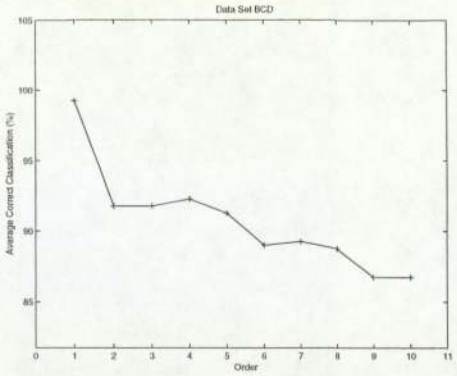

Figure 5.31: Radius AR:2 (kNN), Set BCD

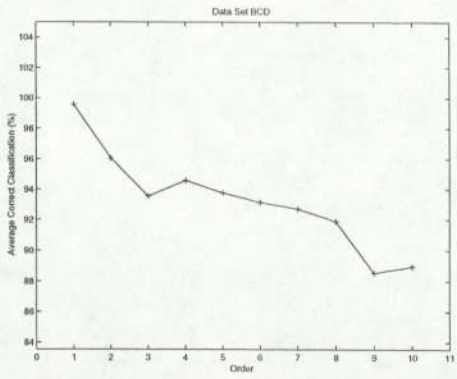

Figure 5.32: Radius AR:2 (kNN), Set BCD

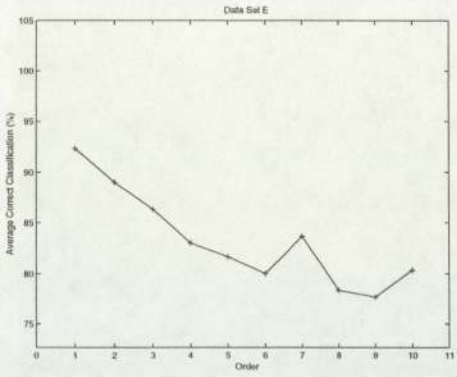

Figure 5.33: Radius AR:2 (kNN), Set E

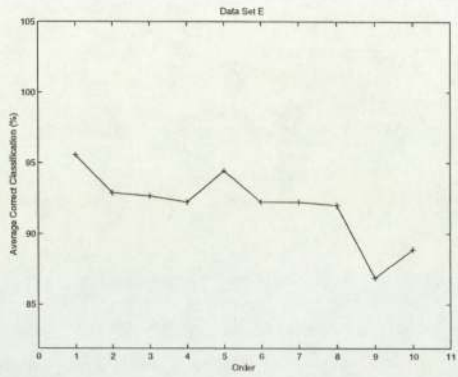

Figure 5.34: Radius AR:2 (kNN), Set E

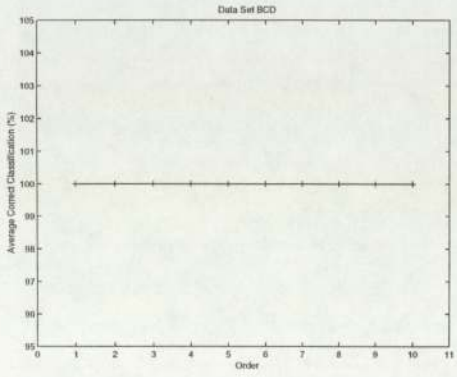

Figure 5.35: Radius AR:2 (RCS), Set BCD

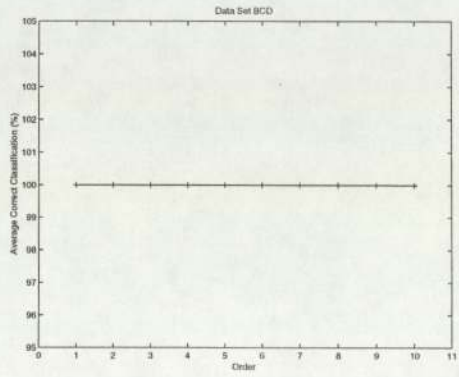

Figure 5.36: Radius AR:2 (RCS), Set BCD

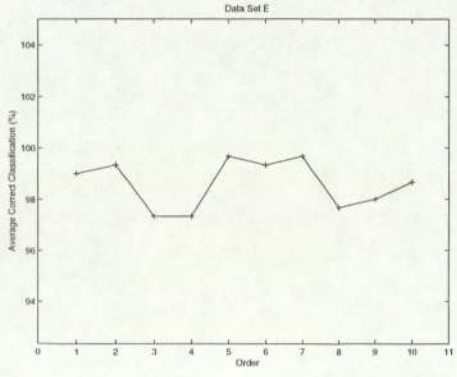

Figure 5.37: Radius AR:2 (RCS), Set E

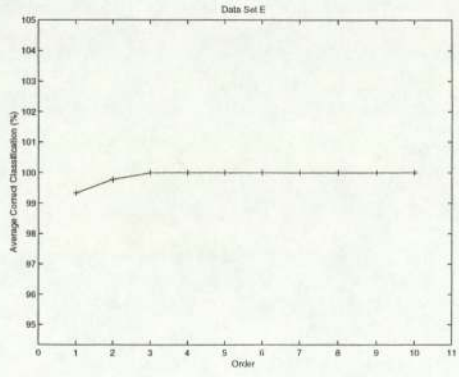

Figure 5.38: Radius AR:2 (RCS), Set E

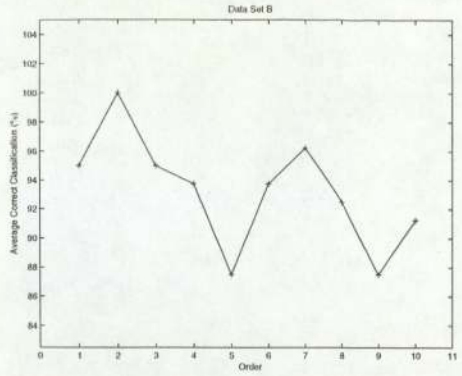

Figure 5.39: Radius AR:2 (kNN), Set B

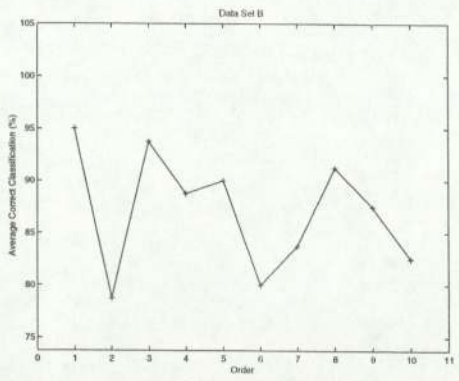

Figure 5.40: Radius AR:2 (kNN), Set B

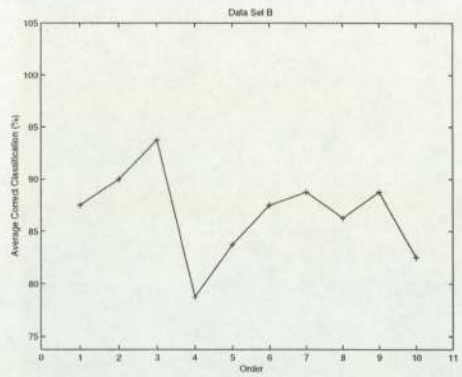

Figure 5.41: Radius AR:2 (kNN), Set B

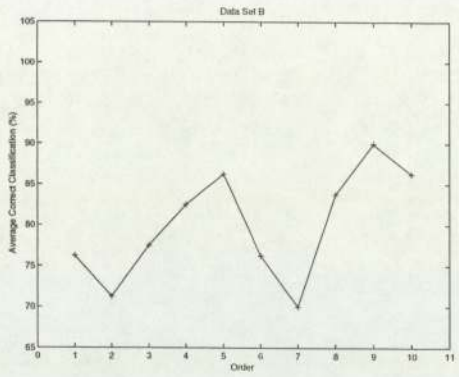

Figure 5.42: Radius AR:2 (kNN), Set B

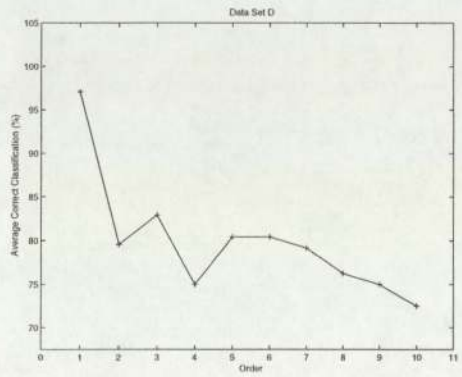

Figure 5.43: Radius AR:2 (kNN), Set D

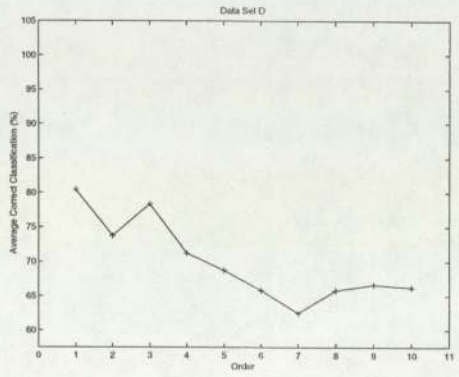

Figure 5.44: Radius AR:2 (kNN), Set D

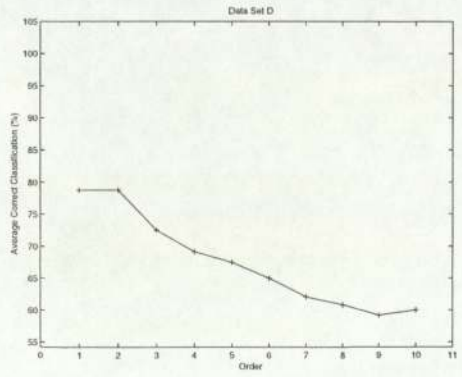

Figure 5.45: Radius AR:2 (kNN), Set D

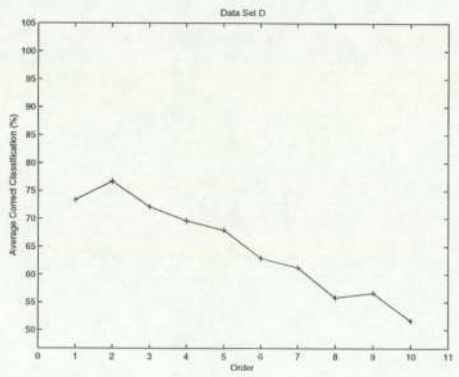

Figure 5.46: Radius AR:2 (kNN), Set D

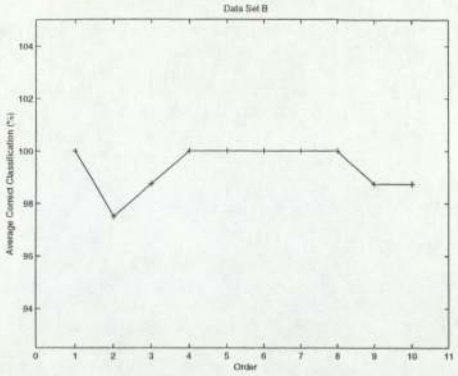

Figure 5.47: Radius AR:2 (RCS), Set B

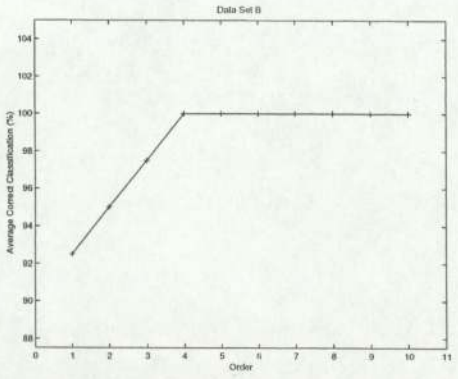

Figure 5.48: Radius AR:2 (RCS), Set B

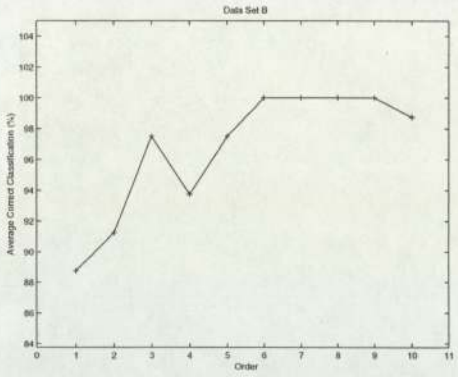

Figure 5.49: Radius AR:2 (RCS), Set B

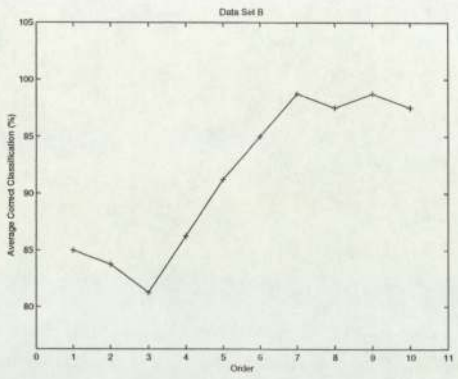

Figure 5.50: Radius AR:2 (RCS), Set B

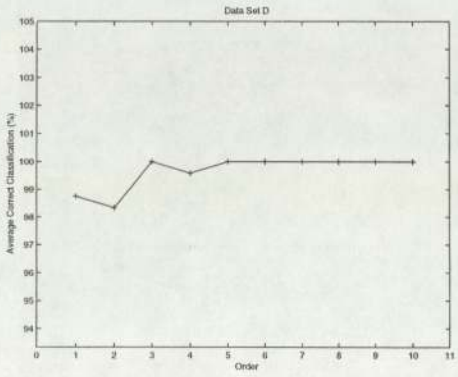

Figure 5.51: Radius AR:2 (RCS), Set D

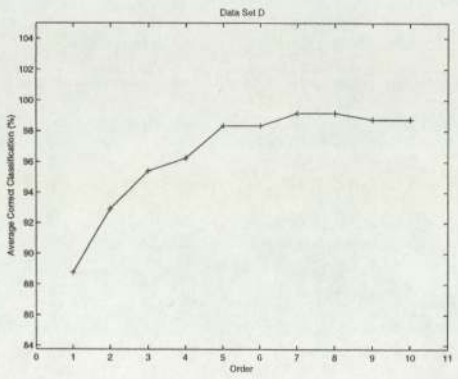

Figure 5.52: Radius AR:2 (RCS), Set D

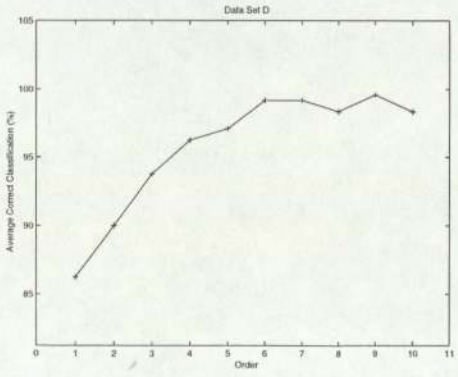

Figure 5.53: Radius AR:2 (RCS), Set D

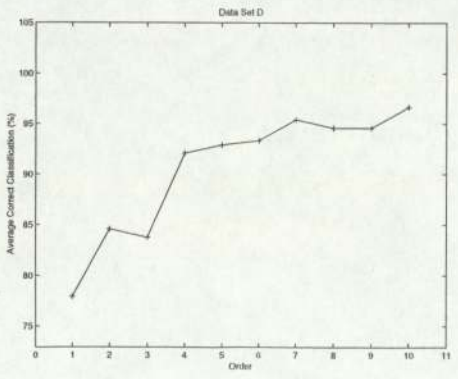

Figure 5.54: Radius AR:2 (RCS), Set D

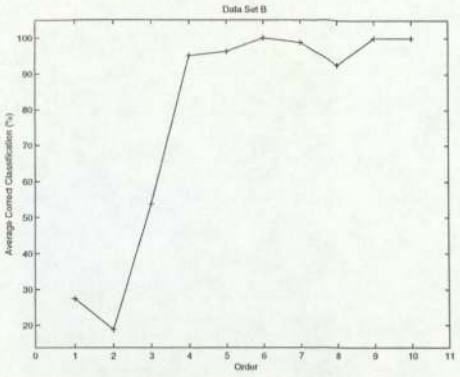

Figure 5.55: Radius AR:2 (RCS), Set B

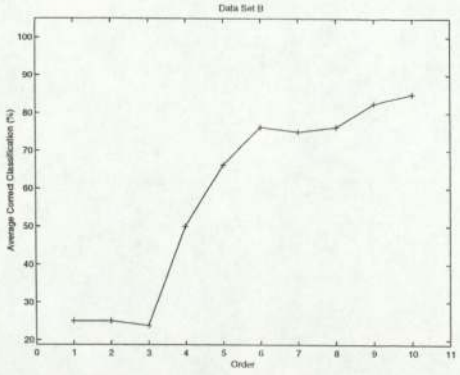

Figure 5.56: Radius AR:2 (RCS), Set B

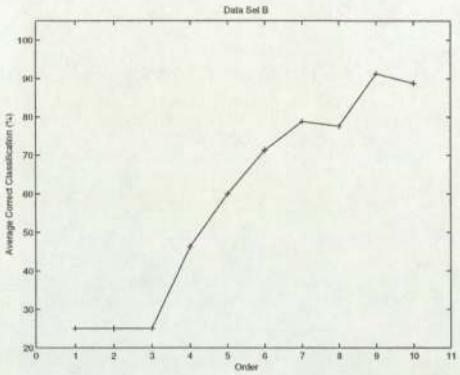

Figure 5.57: Radius AR:2 (RCS), Set B

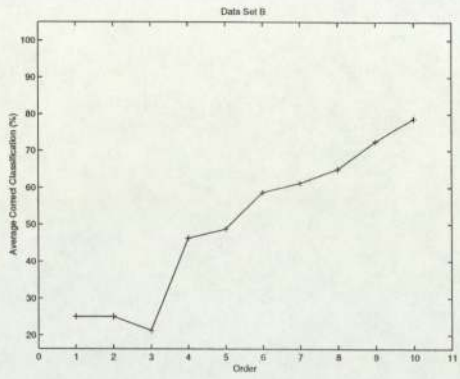

Figure 5.58: Radius AR:2 (RCS), Set B

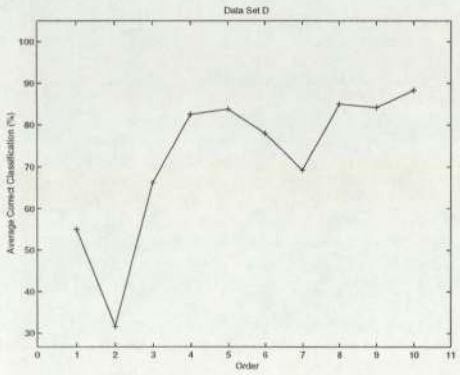

Figure 5.59: Radius AR:2 (RCS), Set D

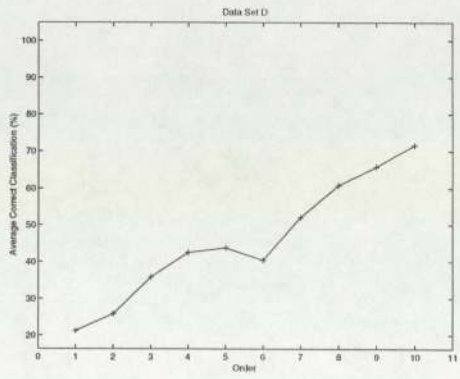

Figure 5.60: Radius AR:2 (RCS), Set D

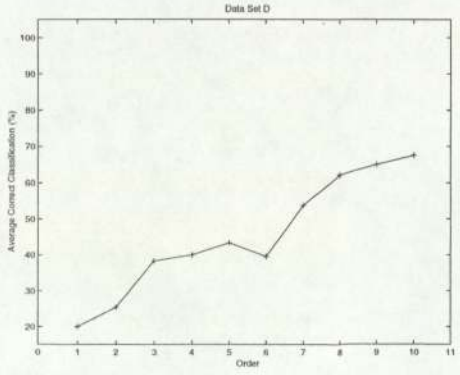

Figure 5.61: Radius AR:2 (RCS), Set D

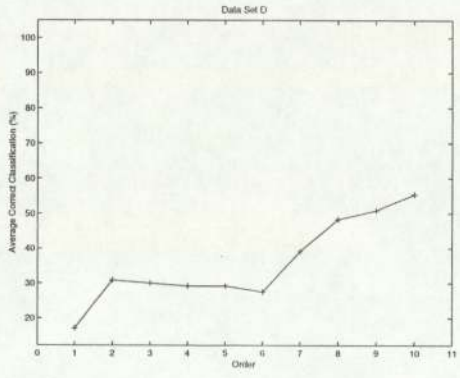

Figure 5.62: Radius AR:2 (RCS), Set D

5.3.3 Curvature AR method

Introduction

The curvature AR method of subsection 4.3.3 uses the curvature angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The curvature function computes the angular difference between two consecutive boundary tangents at each sampling point. The samples are measured at N equi-spaced positions around the boundary, providing uniform spatial coverage. It is thought, but has not been demonstrated in the literature, that curvature based methods would be greatly affected by noisy boundary conditions. The length of the tangents are defined by a window parameter which defines the distance along the boundary between the start and end points of each tangent. For these experiments a window of length twenty five pixels was used to calculate the curvature function. This was empirically set so that the window was long enough to overcome local pixelisation effecting the curvature measure, but not too long so that it was able to extract local curvature information. The boundary samples are treated as a circular time series and AR models of orders 1-10 are fitted to each each series and labelled according to class. The coefficients are used as feature vectors and classified using a kNN and Gaussian classifier.

The results are shown to be comparatively poor, even for the simple benchmark data set B, so only representative results are presented.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Curvature AR	Set B	10	15	kNN	-/-	5.63
Curvature AR	Set B	40	20	kNN	-/-	5.64
Curvature AR	Set D	25	25	kNN	-/-	5.65
Curvature AR	Set D	50	30	kNN	-/-	5.66
Curvature AR	Set BCD	25	25	kNN	-/-	5.67
Curvature AR	Set BCD	50	30	kNN	-/-	5.68
Curvature AR	Set E	20	20	kNN	-/-	5.69
Curvature AR	Set E	50	30	kNN	-/-	5.70
Curvature AR	Set B	10	15	RCS	-/-	5.71
Curvature AR	Set B	40	20	RCS	-/-	5.72
Curvature AR	Set D	25	25	RCS	-/-	5.73
Curvature AR	Set D	50	30	RCS	-/-	5.74
Curvature AR	Set BCD	25	25	RCS	-/-	5.75
Curvature AR	Set BCD	50	30	RCS	-/-	5.76
Curvature AR	Set E	20	20	RCS	-/-	5.77
Curvature AR	Set E	50	30	RCS	-/-	5.78
Curvature AR	Set B	40	20	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.79
Curvature AR	Set B	40	20	RCS	$N(0,1)/N(0,1)$	5.80
Curvature AR	Set B	40	20	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.81
Curvature AR	Set B	40	20	RCS	$t_4(0,1)/t_4(0,1)$	5.82
Curvature AR	Set D	50	30	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.83
Curvature AR	Set D	50	30	RCS	$N(0,1)/N(0,1)$	5.84
Curvature AR	Set D	50	30	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.85
Curvature AR	Set D	50	30	RCS	$t_4(0,1)/t_4(0,1)$	5.86

Table 5.4: Curvature AR experiments

kNN classifier

- No noise on boundaries (all data sets)

  The general ACC trend over all model orders and data sets shows an increase with increasing model order (see figures 5.63 - 5.70).

  For data set B, the maximum classification level achieved was 80% at model order 4. When applied to data set D, containing eight classes, the maximum ACC level achieved was 72%.

  However the experiments using sets BCD and E, containing 15 and 16 classes respectively, show a similar performance with a maximum level of $\simeq 60\%$.

Gaussian classifier

- No noise on boundaries (all data sets)

  Again the general trend over all model orders and data sets shows an increase with increasing model order. For data set B, the maximum ACC achieved was 85% at model order 7. This fell to 75% when considering data set D, which contains twice as many classes.

- Noise on boundaries of test and train sets (sets B and D)

  Figures 5.79 - 5.86 show that the curvature AR technique is not robust to the presence of noise on the boundary as the results are similar to random classification scores.

Summary

The overall correct classification performances were much lower than for the radial linear AR techniques and can be attributed to the curvature sampling algorithm as it is only difference.

The Gaussian classifier shows a superior average correct classification performance over all data sets to that of the kNN classifier, indicating that the Gaussian classifier is more suited to classify AR features.

The curvature AR technique has been shown to fail completely when even low levels of noise are present on the boundary. This is intuitive since any small change to the boundary coordinates can cause a large change in a gradient or angle, whereas a radial distance would only change by a small fraction of its total length.

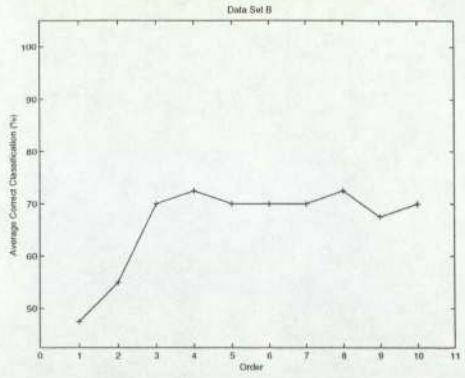

Figure 5.63: Curvature AR (kNN), Set B

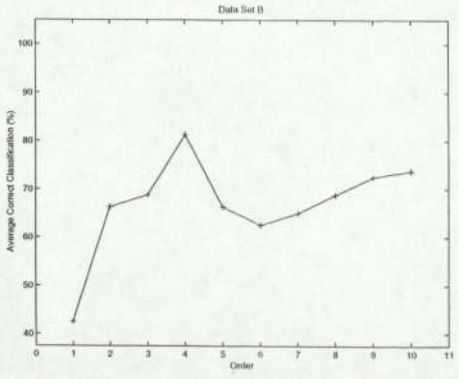

Figure 5.64: Curvature AR (kNN), Set B

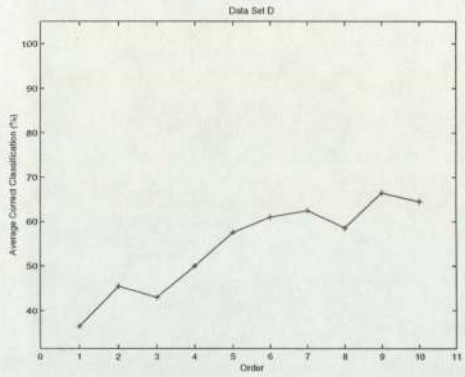

Figure 5.65: Curvature AR (kNN), Set D

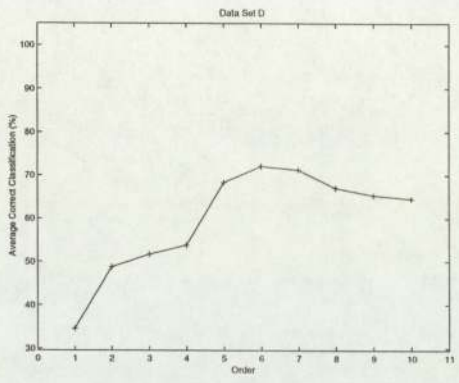

Figure 5.66: Curvature AR (kNN), Set D

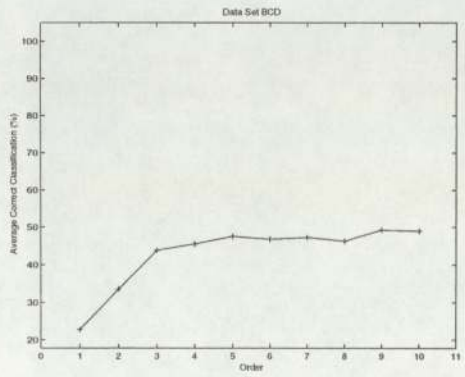

Figure 5.67: Curvature AR (kNN), Set BCD

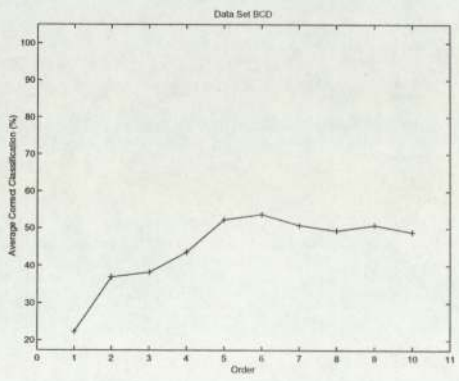

Figure 5.68: Curvature AR (kNN), Set BCD

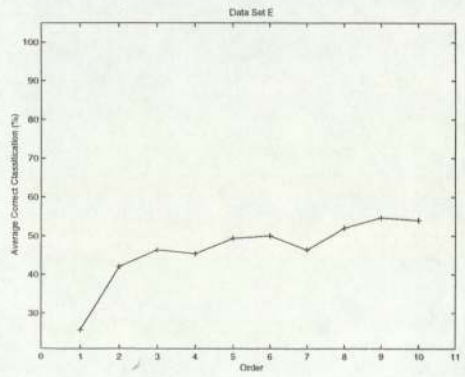

Figure 5.69: Curvature AR (kNN), Set E

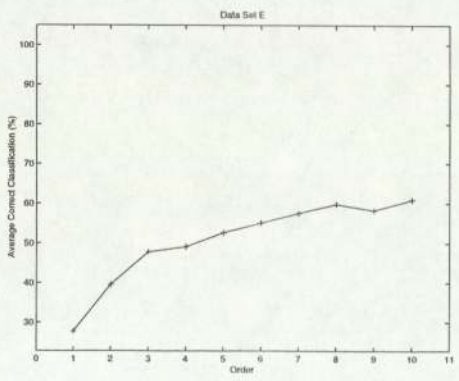

Figure 5.70: Curvature AR (kNN), Set E

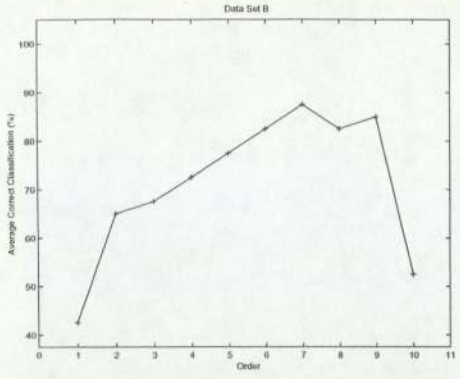

Figure 5.71: Curvature AR (RCS), Set B

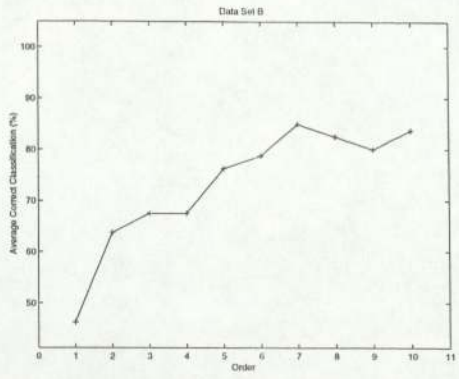

Figure 5.72: Curvature AR (RCS), Set B

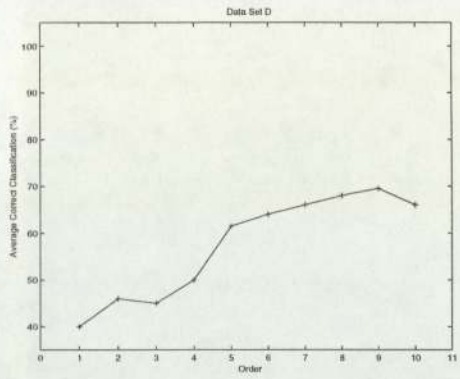

Figure 5.73: Curvature AR (RCS), Set D

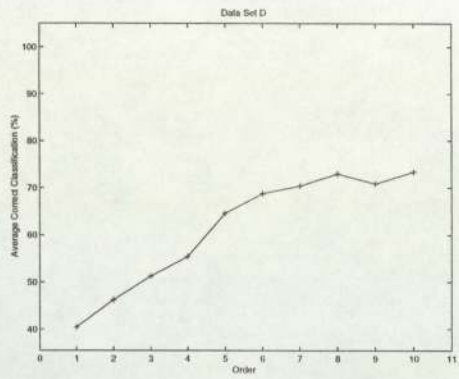

Figure 5.74: Curvature AR (RCS), Set D

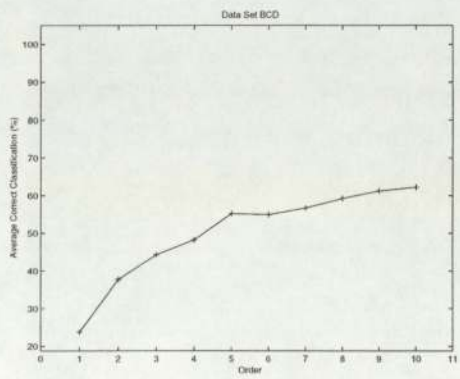

Figure 5.75: Curvature AR (RCS), Set BCD

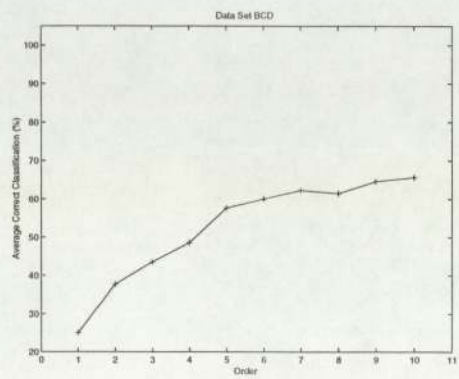

Figure 5.76: Curvature AR (RCS), Set BCD

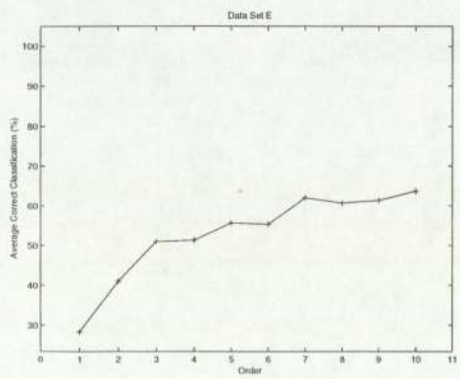

Figure 5.77: Curvature AR (RCS), Set E

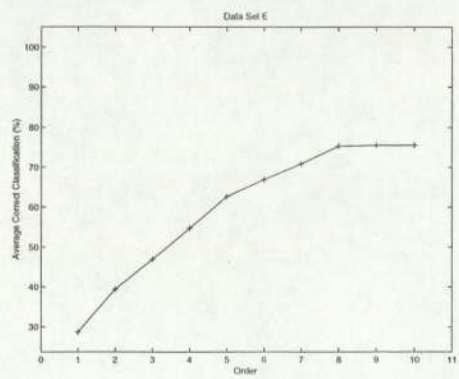

Figure 5.78: Curvature AR (RCS), Set E

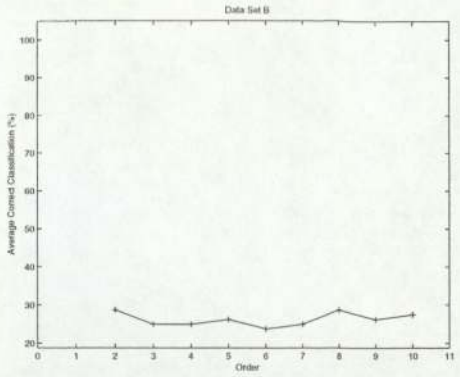

Figure 5.79: Curvature AR (RCS), Set B

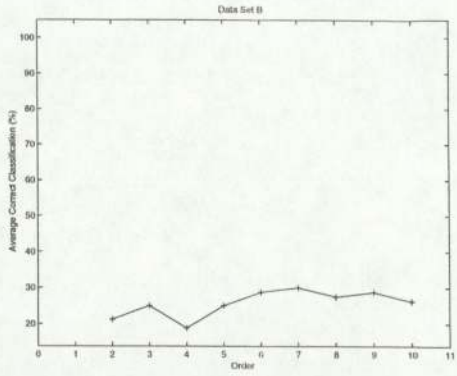

Figure 5.80: Curvature AR (RCS), Set B

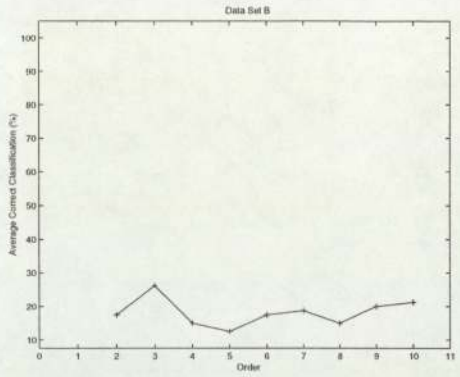

Figure 5.81: Curvature AR (RCS), Set B

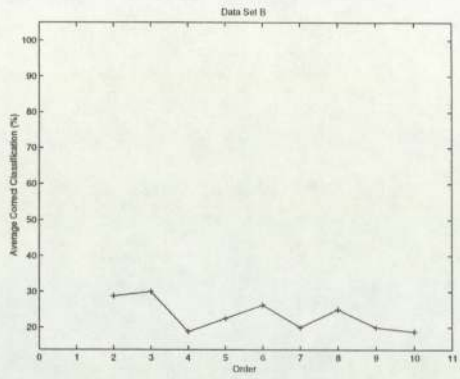

Figure 5.82: Curvature AR (RCS), Set B

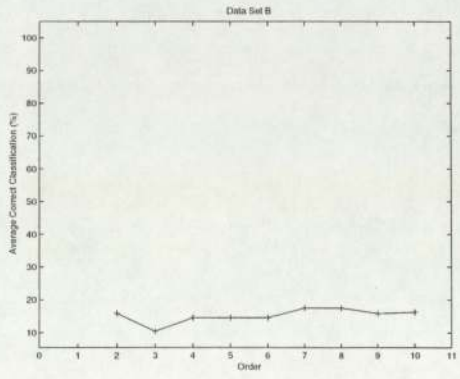

Figure 5.83: Curvature AR (RCS), Set D

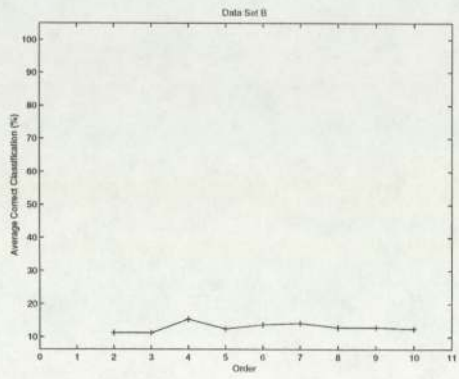

Figure 5.84: Curvature AR (RCS), Set D

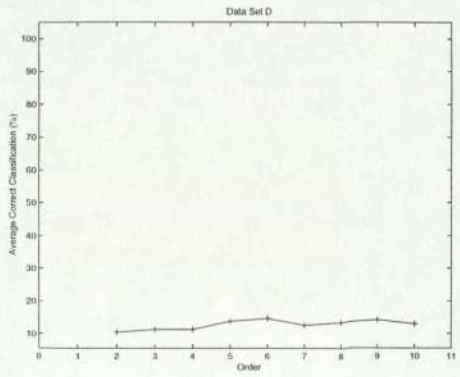

Figure 5.85: Curvature AR (RCS), Set D

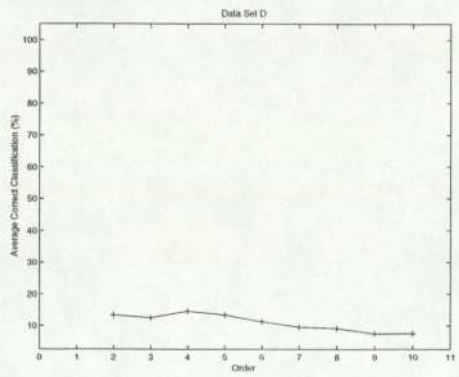

Figure 5.86: Curvature AR (RCS), Set D

5.3.4 Complex AR method (CAR)

Introduction

The CAR method of subsection 4.3.5 [5] uses the complex coordinate angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The samples are measured at N equi-spaced positions along the boundary, providing uniform spatial coverage. The complex form of the samples provides extra phase information so that the polygonal approximation of the original boundary can be reconstructed. The boundary samples are treated as a circular complex time series and CAR models of order 1-10 are fitted to each series and assigned class labels for use as features for shape classification. We used 64 spatially equi-distant samples. Increasing the number of samples had little or no effect on the classification performance.

The feature vectors from each class were split into testing and training sets and a Gaussian and kNN classifier are compared. Table 5.5 describes the experiments performed.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
CAR	Set B	10	15	kNN	-/-	B.9
CAR	Set B	40	20	kNN	-/-	B.10
CAR	Set D	25	25	kNN	-/-	B.11
CAR	Set D	50	30	kNN	-/-	B.12
CAR	Set BCD	25	25	kNN	-/-	5.87
CAR	Set BCD	50	30	kNN	-/-	5.88
CAR	Set E	20	20	kNN	-/-	5.89
CAR	Set E	50	30	kNN	-/-	5.90
CAR	Set B	10	15	RCS	-/-	B.13
CAR	Set B	40	20	RCS	-/-	B.14
CAR	Set D	25	25	RCS	-/-	B.15
CAR	Set D	50	30	RCS	-/-	B.16
CAR	Set BCD	25	25	RCS	-/-	5.91
CAR	Set BCD	50	30	RCS	-/-	5.92
CAR	Set E	20	20	RCS	-/-	5.93
CAR	Set E	50	30	RCS	-/-	5.94
CAR	Set B	40	20	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.95
CAR	Set B	40	20	kNN	$N(0,1)/N(0,1)$	5.96
CAR	Set B	40	20	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.97
CAR	Set B	40	20	kNN	$t_4(0,1)/t_4(0,1)$	5.98
CAR	Set D	50	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.99
CAR	Set D	50	30	kNN	$N(0,1)/N(0,1)$	5.100
CAR	Set D	50	30	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.101
CAR	Set D	50	30	kNN	$t_4(0,1)/t_4(0,1)$	5.102
CAR	Set B	40	20	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.103
CAR	Set B	40	20	RCS	$N(0,1)/N(0,1)$	5.104
CAR	Set B	40	20	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.105
CAR	Set B	40	20	RCS	$t_4(0,1)/t_4(0,1)$	5.106
CAR	Set D	50	30	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.107
CAR	Set D	50	30	RCS	$N(0,1)/N(0,1)$	5.108
CAR	Set D	50	30	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.109
CAR	Set D	50	30	RCS	$t_4(0,1)/t_4(0,1)$	5.110
CAR	Set B	40	20	RCS	$-/N(0,0.5^2)$	5.111
CAR	Set B	40	20	RCS	$-/N(0,1)$	5.112
CAR	Set B	40	20	RCS	$-/t_{40}(0,1)$	5.113
CAR	Set B	40	20	RCS	$-/t_4(0,1)$	5.114
CAR	Set D	50	30	RCS	$-/N(0,0.5^2)$	5.115
CAR	Set D	50	30	RCS	$-/N(0,1)$	5.116
CAR	Set D	50	30	RCS	$-/t_{40}(0,1)$	5.117
CAR	Set D	50	30	RCS	$-/t_4(0,1)$	5.118

Table 5.5: CAR experiments

kNN classifier

- No noise on boundaries (all data sets)

  Figures B.9 - 5.90 show a general trend of decreasing ACC levels with increasing model order for all of the data sets tested. This is thought to be caused by having a small training set with a large dimension feature vector, resulting in a very sparse training set. The decrease in performance is greater than for the equivalent radius AR method 2 experiments as we have a feature vector almost twice as long.

- Noise on boundaries of test and train sets (sets B and D)

  For the tests on data set B, when the model order equals one, the ACC levels were greater then 90% for all noise levels. For model orders 2-10, the ACC performances fell to between 50-65%. Only a small decrease in performance was observed with increasing noise levels.

  For the tests on data set D, again a high level of ACC performance was observed at model order 1. The effect of increasing noise levels was apparent as the ACC performances fell from 98% for for noise sampled from a $N(0, 0.5^2)$ distribution to 82% for noise drawn from a $N(0, 1)$ distribution. The tests where the noise was drawn from a $t_{40}(0, 1)$ distribution reulted in similar ACC levels to the noise drawn from the $N(0, 1)$ distribution, falling to 74% when the noise was drawn from a $t_4(0, 1)$ distribution. The ACC performances of the kNN classifier for model orders 2-10 were similar to those described for set B, typically ACC values ranging between 50-60%.

Gaussian classifier

- No noise on boundaries (all data sets)

  Figures B.13 - 5.94 show the improvement of the Gaussian classifier over the kNN classifier applied to these data, where all model order tests are classified 100% correctly for sets B and D, and tests when the model orders are between 3-10 for data sets BCD and E. The Gaussian classifier performs as well as the equivalent tests using the radial AR method 2 technique on all of the data sets (compare figures B.5 - 5.38 with B.13 - 5.94), although at high orders, typically 8-10, spurious results can occur due to the covariance matrix of the Gaussian classifier becoming almost singular. In these cases we are trying to estimate ten complex parameters (20 real parameters in the feature vector) from only twenty training samples for set E. Note this effect also occurs in figure B.13, but when the number of training samples is increased the covariance can be estimated more accurately, see figures B.14 and 5.94, and the spurious results at higher orders disappear. Generally, with no noise added to the boundary, the Gaussian classifier displays superior classification results to those of the kNN classifier.

- Noise on boundaries of test and train sets (sets B and D)

Tests on set B (figures 5.95 - 5.102) show 100% ACC levels for noise drawn from $N(0, 0.5^2)$, $N(0, 1)$ and $t_{40}(0, 1)$ for model orders 1-10. For the tests using the $t_4(0, 1)$ noise distribution at model order 1, the ACC fell to 75% but increased to 100% for model orders 3-10.

Tests on set D show a general degradation in ACC performances when the noise distributions are, in order, $N(0, 0.5^2)$, $N(0, 1)$, $t_{40}(0, 1)$ and $t_{40}(0, 1)$. The graphs also show that the higher the model order, the higher the ACC performance.

- Noise on boundaries of test set only (sets B and D)

  With no noise present in the training set but with noise added to the boundaries in the test set, figures 5.111 - 5.118 show that high classification levels are achieved when compared with the equivalent results for the radius AR method 2. This is thought to be due to the sampling method as the complex coordinate samples contain phase information about the boundary, whereas the radial distance samples are sequentially ordered according to the tracing of the boundary but different radius shapes could give the same ordered lengths. Additionally we also have a more complex model than the AR method 2 technique as we are estimating complex AR coefficients, resulting in a feature vector twice as long if we consider the complex number components as real numbers.

Summary

The Gaussian classifier was shown to exhibit superior performance to that of the kNN classifier. The results using the Gaussian classifier improved on those achieved using the AR method 2, especially in high boundary noise conditions with a noise free training set and a noisy test set. This could be for two reasons: firstly the complex coordinate samples contain more phase information about the boundary than the radial distance samples and secondly, the CAR model has a higher dimension, more complex feature vector.

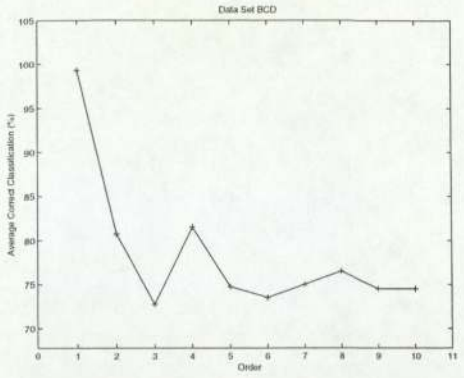

Figure 5.87: CAR (kNN), Set BCD

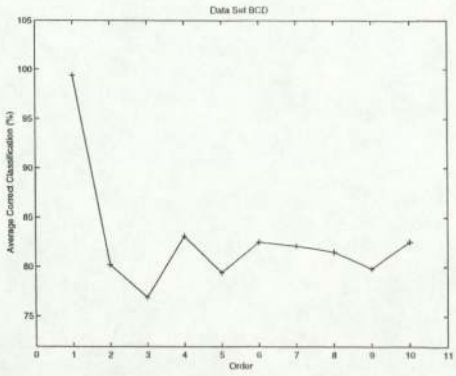

Figure 5.88: CAR (kNN), Set BCD

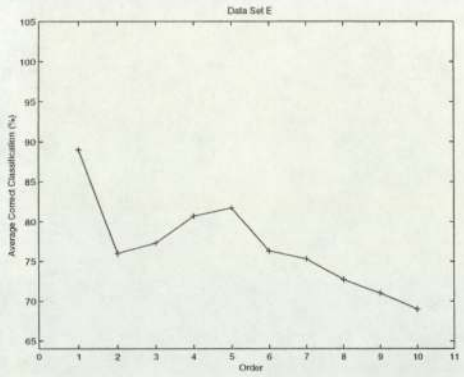

Figure 5.89: CAR (kNN), Set E

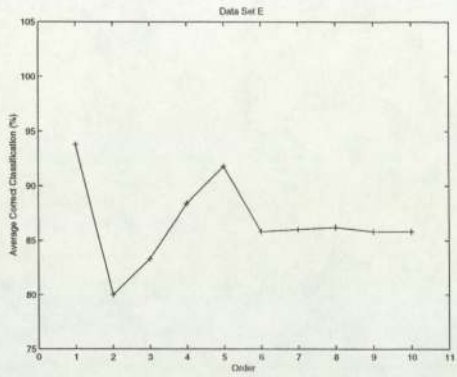

Figure 5.90: CAR (kNN), Set E

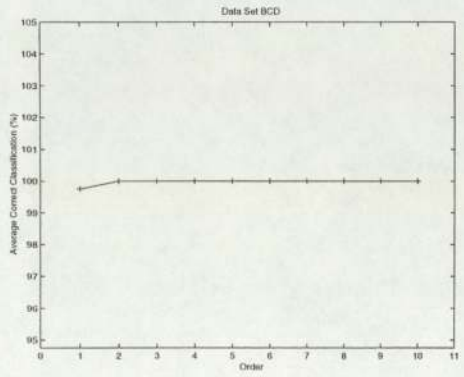

Figure 5.91: CAR (RCS), Set BCD

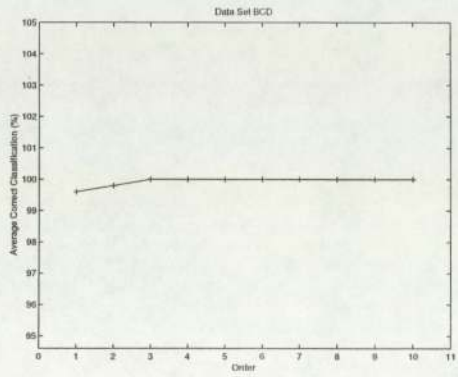

Figure 5.92: CAR (RCS), Set BCD

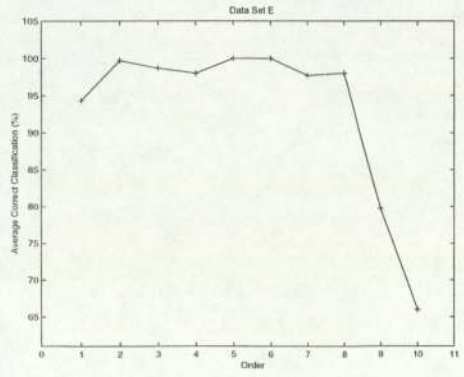

Figure 5.93: CAR (RCS), Set E

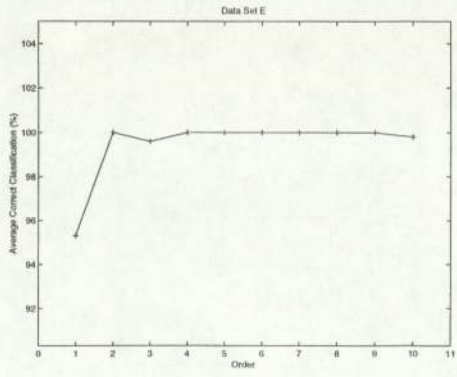

Figure 5.94: CAR (RCS), Set E

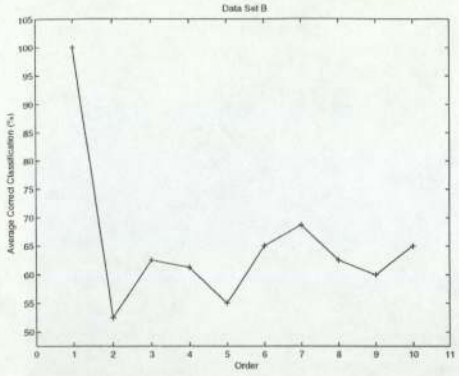

Figure 5.95: CAR (kNN), Set B

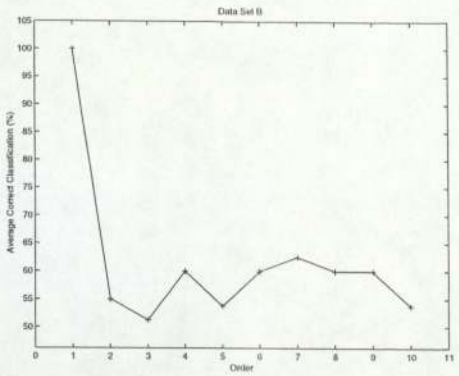

Figure 5.96: CAR (kNN), Set B

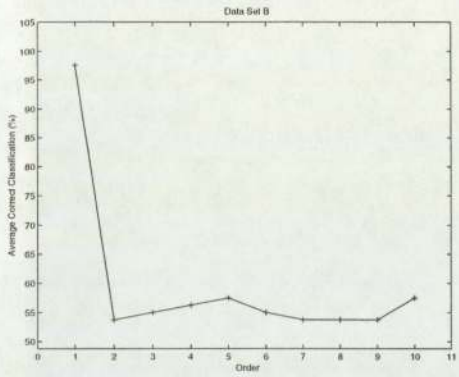

Figure 5.97: CAR (kNN), Set B

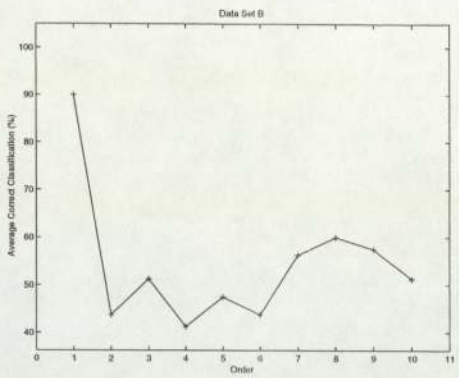

Figure 5.98: CAR (kNN), Set B

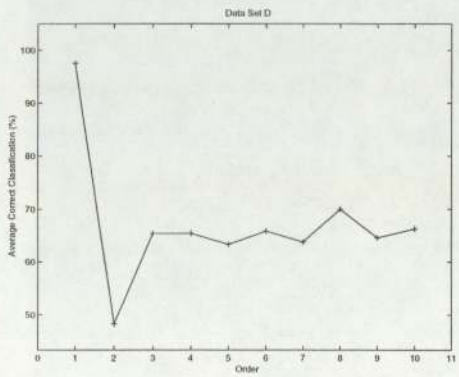

Figure 5.99: CAR (kNN), Set D

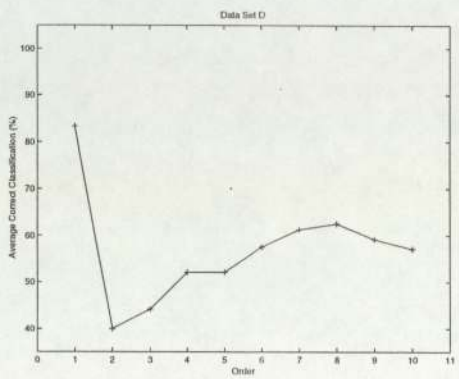

Figure 5.100: CAR (kNN), Set D

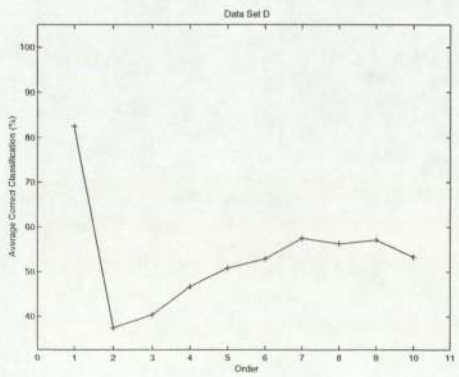

Figure 5.101: CAR (kNN), Set D

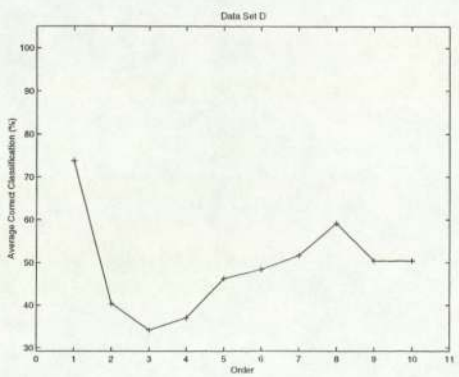

Figure 5.102: CAR (kNN), Set D

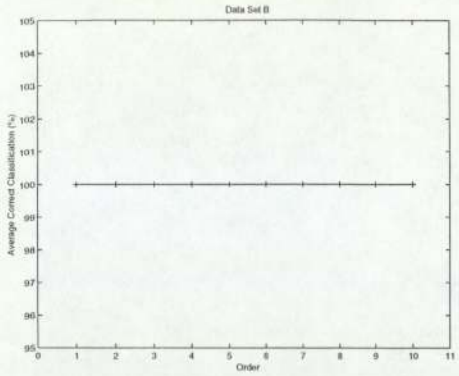

Figure 5.103: CAR (RCS), Set B

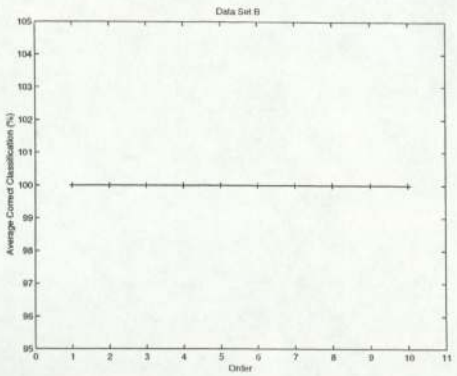

Figure 5.104: CAR (RCS), Set B

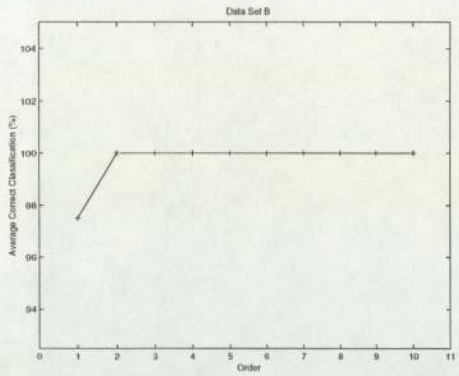

Figure 5.105: CAR (RCS), Set B

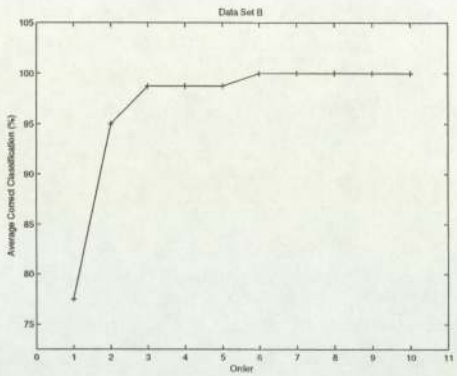

Figure 5.106: CAR (RCS), Set B

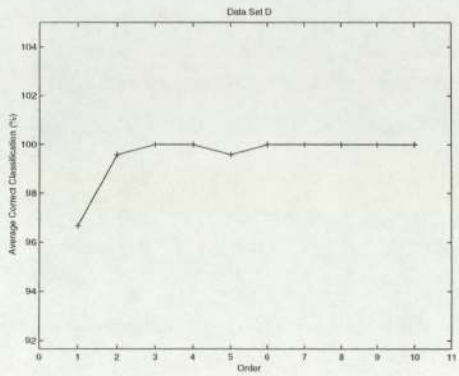

Figure 5.107: CAR (RCS), Set D

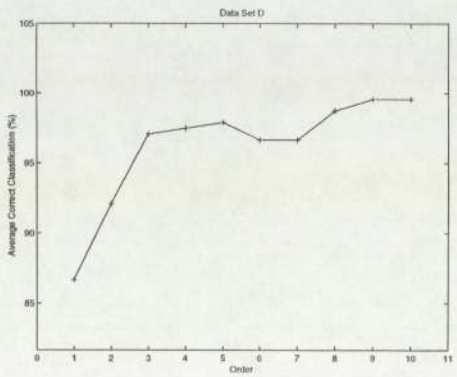

Figure 5.108: CAR (RCS), Set D

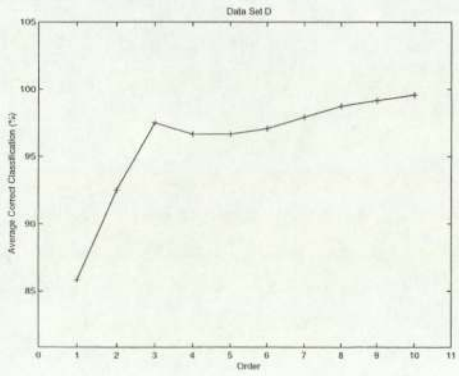

Figure 5.109: CAR (RCS), Set D

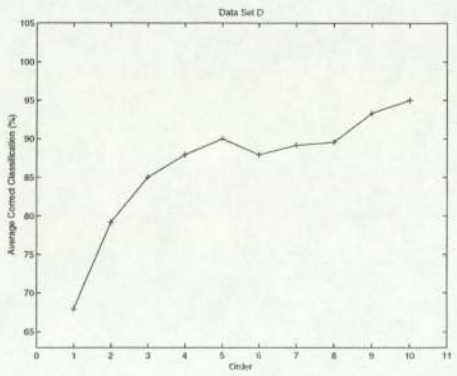

Figure 5.110: CAR (RCS), Set D

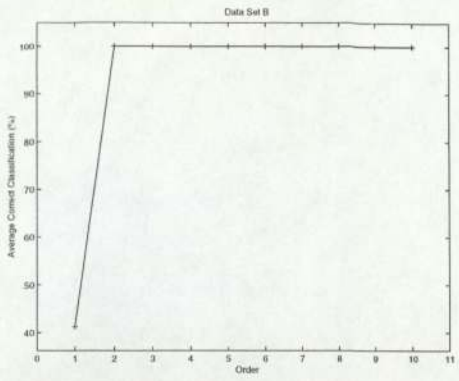

Figure 5.111: CAR (RCS), Set B

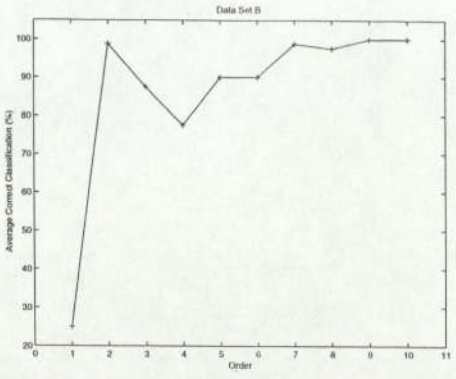

Figure 5.112: CAR (RCS), Set B

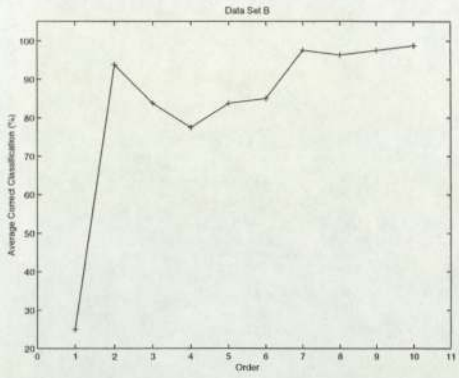

Figure 5.113: CAR (RCS), Set B

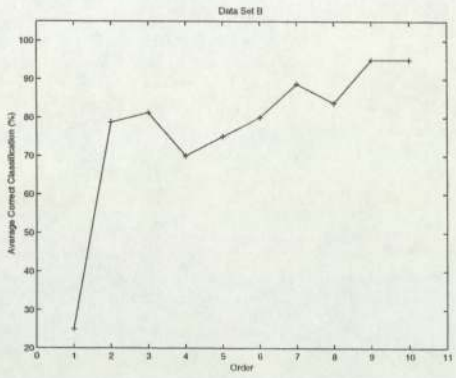

Figure 5.114: CAR (RCS), Set B

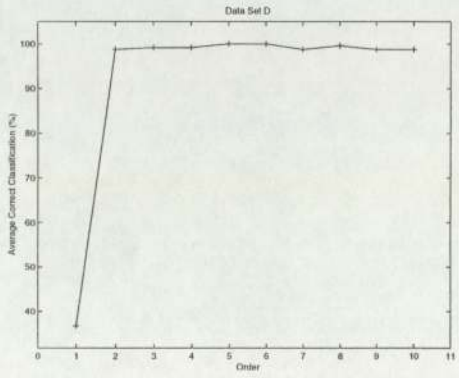

Figure 5.115: CAR (RCS), Set D

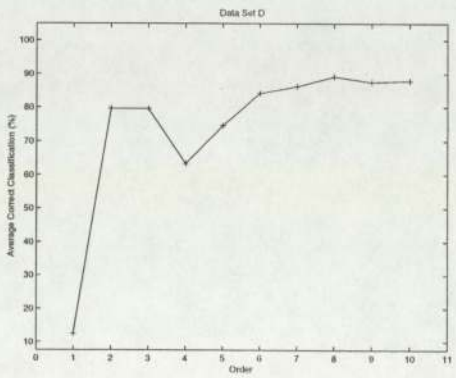

Figure 5.116: CAR (RCS), Set D

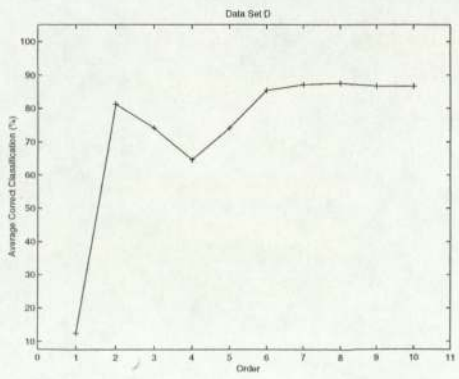

Figure 5.117: CAR (RCS), Set D

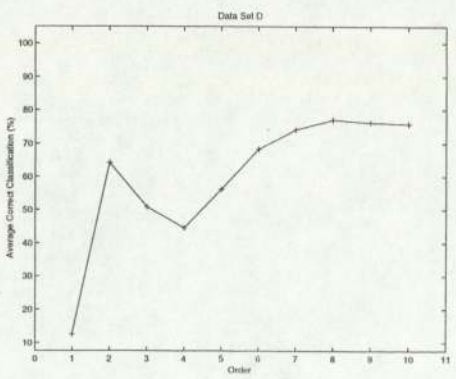

Figure 5.118: CAR (RCS), Set D

5.3.5 CPARCOR method

Introduction

The CPARCOR method [5] of subsection 4.3.6 uses the complex coordinate angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The boundary samples are treated as a circular complex time series and CPARCOR models of order 1-10 are fitted to each series and assigned class labels. The CPARCOR coefficients are first calculated by the fast recursive algorithm and then compared with the equivalent direct method. The CPARCOR features can be computed directly from the CAR coefficients as the CPARCOR coefficient p_m is the same as the CAR coefficient a_m of the CAR model of order m. To be consistent with previous experiments we again used 64 spatially equi-distant samples.

The feature vectors from each class were split into test and train sets and a Gaussian and kNN classifier are compared. Table 5.6 describes the experiments performed for the CPARCOR coefficients computed by the recursive algorithm, and table 5.7 describes the experiments performed using the CPARCOR coefficients computed directly from the CAR coefficients. The direct method was used only as a consequence of the poor performance shown by the recursive method. The results would be expected to be identical but the possible reasons for this are discussed [5].

CPARCOR coefficient estimation by the recursive algorithm

- kNN classifier
  - No noise on boundaries (all data sets)

    Figures B.17 - 5.122 a steady but rapid decline in ACC levels with increasing model order, for all of the data sets.
  - Noise on boundaries of test and train sets (sets B and D)

    The ACC levels again decreased with increasing model order. The ACC levels decreased with increasing noise level, such as samples drawn from $N(0, 0.5^2)$, $N(0, 1)$, $t_{40}(0, 1)$, and $t_4(0, 1)$ distributions, for both sets B and D.
- Gaussian classifier
  - No noise on boundaries (all data sets)

    The results are very similar to those of the CAR model tests, where 100% is achieved for all data sets. Again spurious results occur for high order models with small sets of training data, due to a singular covariance matrix in the classifier.
  - Noise on boundaries of test and train sets (sets B and D)

Figures 5.135 - 5.142 show that the noise degrades the ACC levels significantly. It was expected that the performance would be similar to those of the CAR model.

- Noise on boundary of test set only (sets B and D)

  Figures 5.143 - 5.150 show that ACC levels are similar to random classification scores.

CPARCOR coefficient estimation by direct computation from CAR coefficients

- Gaussian classifier

  - No noise on boundaries (all data sets)

    As the Gaussian classifer was shown to be superior for the recursively computed CPARCOR features, it was again used for the purposes of this comparison. Figures 5.151 - 5.153 show the ACC performances for all data sets and equals 100%.

  - Noise on boundaries of test and train sets (set D)

    Four levels of noise were added to the boundaries of set D. The ACC results are far superior to those calculated by the indirect recursive algorithm. Comparing figures 5.154 - 5.157 with figures 5.107 - 5.110, the results are almost identical to those of the CAR model.

Summary

With no noise added to the boundaries of either test or training sets the Gaussian classifier outperforms the kNN classifier. Similar levels of ACC performance levels to those of the CAR model are shown on all data sets when the CPARCOR coefficients are estimated by both direct and indirect methods. However if noise is added to the boundaries, the ACC levels of the CPARCOR models calculated by the indirect method dropped dramatically, for all data sets and noise types, especially at high orders. This could be because the boundary noise changes the statistics of the boundary samples, so that the assumptions made by the recursive estimation algorithm no longer hold. The recursive method was described by Sekita [5] as an *extension* of a complex series of Levinson-Durbin's algorithm for calculating real PARCOR coefficients. This was the only reference found relating to this extended technique but it is thought to be an approximation because the CAR coefficients computed using both recursive and direct techniques produced very similar but not equal values, when tested on a synthetically produced complex time series. Another possible reason for the poor performance of the recursive algorithm could be due to the numerical sensitivity of the algorithm. The direct method is a straight forward and foolproof implementation and so verifies the true performance of CPARCOR coefficients.

It has been shown that the ACC experiments where the CPARCOR coefficients are calculated directly from the CAR coefficients produce results very similar to those produced by the CAR model and are more robust to boundary noise than the linear AR models. This is due to the fact that the

number of parameters are greater and also the complex samples contain important phase information. If phase information is present then a polygonal approximation of the original shape is possible, whereas radial vector samples (no phase information) could result from any of a number of shapes.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
CPARCOR	Set B	10	15	kNN	-/-	B.17
CPARCOR	Set B	40	20	kNN	-/-	B.18
CPARCOR	Set D	25	25	kNN	-/-	B.19
CPARCOR	Set D	50	30	kNN	-/-	B.20
CPARCOR	Set BCD	25	25	kNN	-/-	5.119
CPARCOR	Set BCD	50	30	kNN	-/-	5.120
CPARCOR	Set E	20	20	kNN	-/-	5.121
CPARCOR	Set E	50	30	kNN	-/-	5.122
CPARCOR	Set B	10	15	RCS	-/-	B.21
CPARCOR	Set B	40	20	RCS	-/-	B.22
CPARCOR	Set D	25	25	RCS	-/-	B.23
CPARCOR	Set D	50	30	RCS	-/-	B.24
CPARCOR	Set BCD	25	25	RCS	-/-	5.123
CPARCOR	Set BCD	50	30	RCS	-/-	5.124
CPARCOR	Set E	20	20	RCS	-/-	5.125
CPARCOR	Set E	50	30	RCS	-/-	5.126
CPARCOR	Set B	40	20	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.127
CPARCOR	Set B	40	20	kNN	$N(0,1)/N(0,1)$	5.128
CPARCOR	Set B	40	20	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.129
CPARCOR	Set B	40	20	kNN	$t_4(0,1)/t_4(0,1)$	5.130
CPARCOR	Set D	50	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.131
CPARCOR	Set D	50	30	kNN	$N(0,1)/N(0,1)$	5.132
CPARCOR	Set D	50	30	kNN	$t_{40}(0,1)/t_{40}(0,1)$	5.133
CPARCOR	Set D	50	30	kNN	$t_4(0,1)/t_4(0,1)$	5.134
CPARCOR	Set B	40	20	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.135
CPARCOR	Set B	40	20	RCS	$N(0,1)/N(0,1)$	5.136
CPARCOR	Set B	40	20	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.137
CPARCOR	Set B	40	20	RCS	$t_4(0,1)/t_4(0,1)$	5.138
CPARCOR	Set D	50	30	RCS	$N(0,0.5^2)/N(0,0.5^2)$	5.139
CPARCOR	Set D	50	30	RCS	$N(0,1)/N(0,1)$	5.140
CPARCOR	Set D	50	30	RCS	$t_{40}(0,1)/t_{40}(0,1)$	5.141
CPARCOR	Set D	50	30	RCS	$t_4(0,1)/t_4(0,1)$	5.142
CPARCOR	Set B	40	20	RCS	$-/N(0,0.5^2)$	5.143
CPARCOR	Set B	40	20	RCS	$-/N(0,1)$	5.144
CPARCOR	Set B	40	20	RCS	$-/t_{40}(0,1)$	5.145
CPARCOR	Set B	40	20	RCS	$-/t_4(0,1)$	5.146
CPARCOR	Set D	50	30	RCS	$-/N(0,0.5^2)$	5.147
CPARCOR	Set D	50	30	RCS	$-/N(0,1)$	5.148
CPARCOR	Set D	50	30	RCS	$-/t_{40}(0,1)$	5.149
CPARCOR	Set D	50	30	RCS	$-/t_4(0,1)$	5.150

Table 5.6: CPARCOR experiments (coefficients calculated by recursive algorithm)

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
CPARCOR	Set B	40	20	kNN	-/-	5.151
CPARCOR	Set D	50	30	kNN	-/-	5.152
CPARCOR	Set E	50	30	kNN	-/-	5.153
CPARCOR	Set D	50	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.154
CPARCOR	Set D	50	30	kNN	$N(0,1)/N(0,1)$	5.155
CPARCOR	Set D	50	30	kNN	$t_4 0(0,1)/t_4 0(0,1)$	5.156
CPARCOR	Set D	50	30	kNN	$t_4(0,1)/t_4(0,1)$	5.157

Table 5.7: CPARCOR experiments (coefficients calculated by direct method)

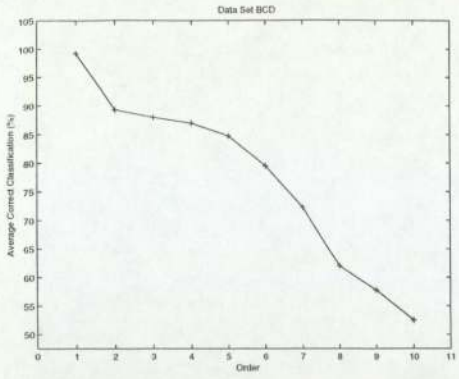

Figure 5.119: CPARCOR (kNN), Set BCD

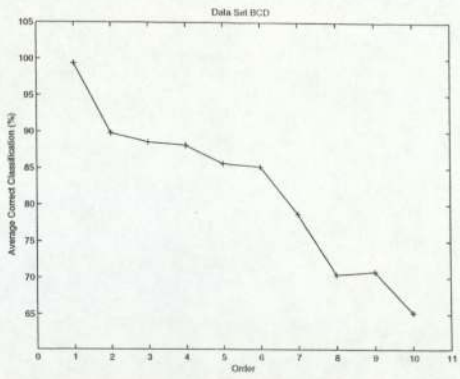

Figure 5.120: CPARCOR (kNN), Set BCD

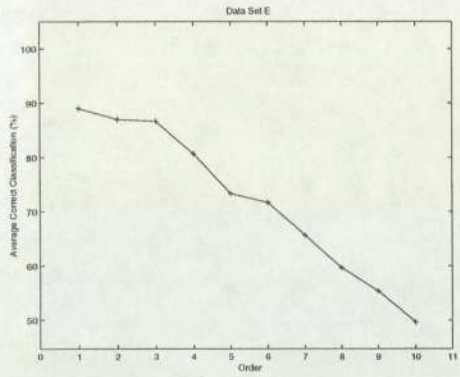

Figure 5.121: CPARCOR (kNN), Set E

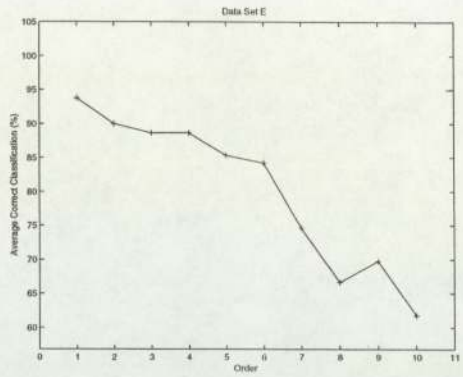

Figure 5.122: CPARCOR (kNN), Set E

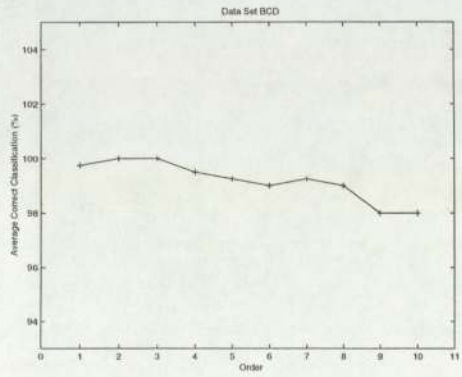

Figure 5.123: CPARCOR (RCS), Set BCD

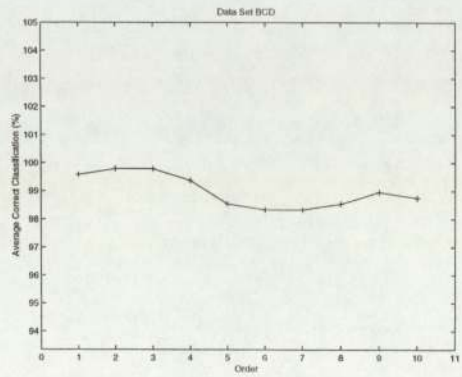

Figure 5.124: CPARCOR (RCS), Set BCD

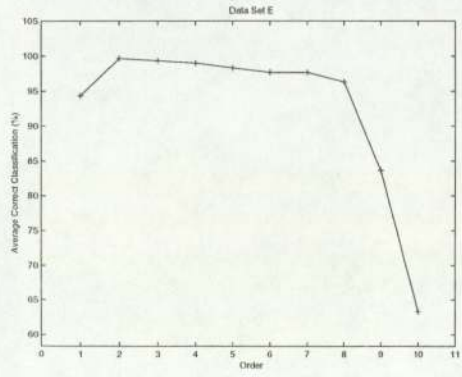

Figure 5.125: CPARCOR (RCS), Set E

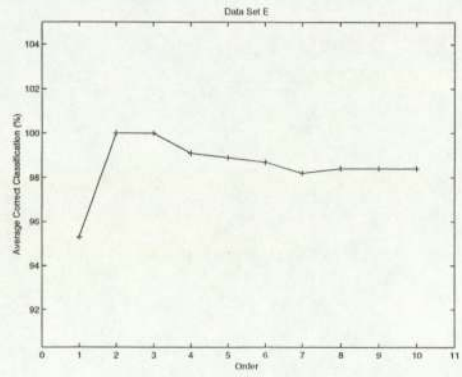

Figure 5.126: CPARCOR (RCS), Set E

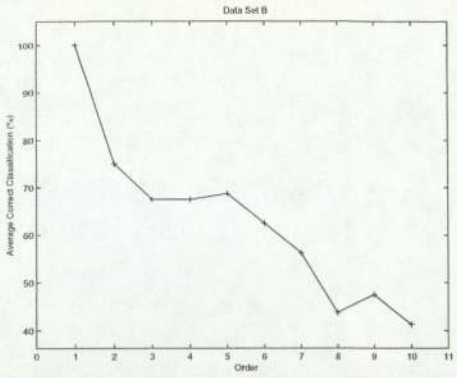

Figure 5.127: CPARCOR (kNN), Set B

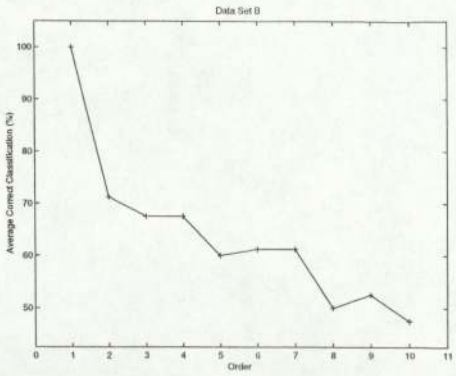

Figure 5.128: CPARCOR (kNN), Set B

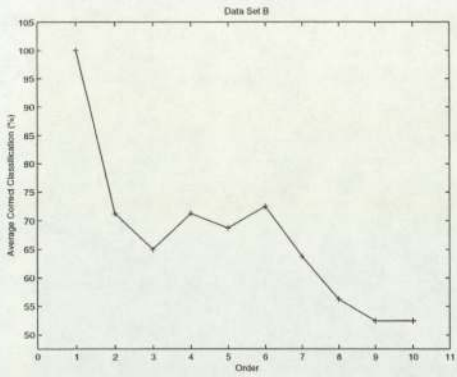

Figure 5.129: CPARCOR (kNN), Set B

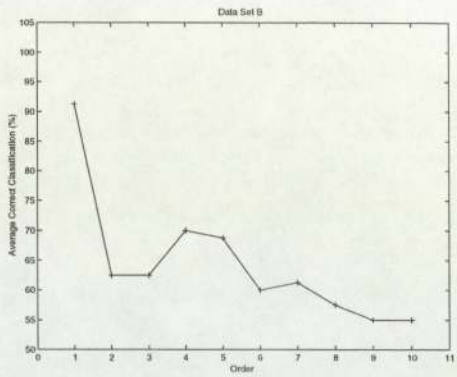

Figure 5.130: CPARCOR (kNN), Set B

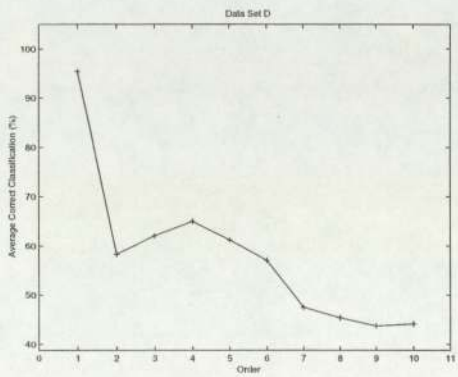

Figure 5.131: CPARCOR (kNN), Set D

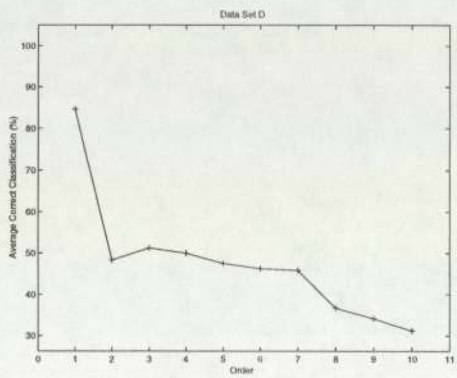

Figure 5.132: CPARCOR (kNN), Set D

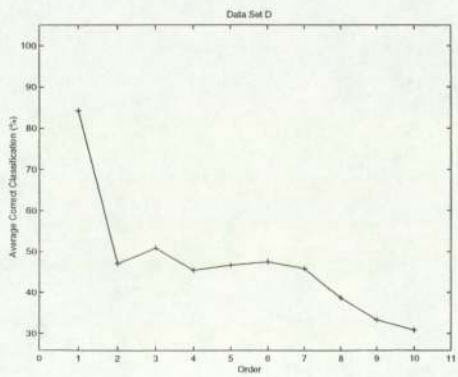

Figure 5.133: CPARCOR (kNN), Set D

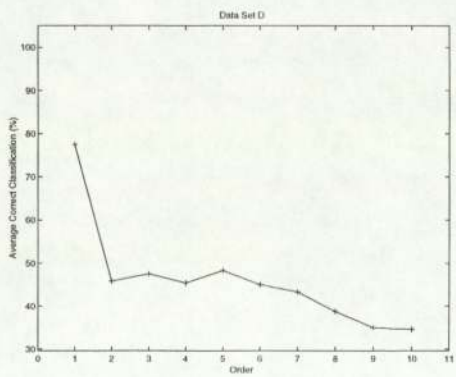

Figure 5.134: CPARCOR (kNN), Set D

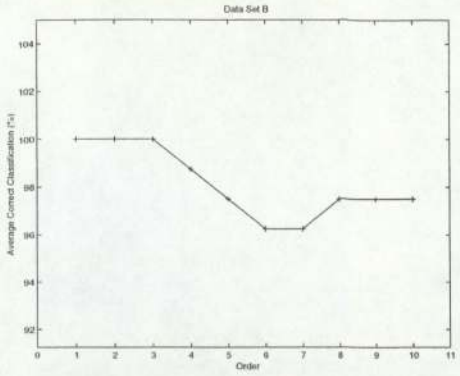

Figure 5.135: CPARCOR (RCS), Set B

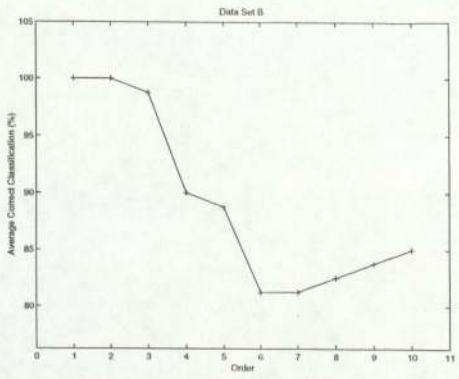

Figure 5.136: CPARCOR (RCS), Set B

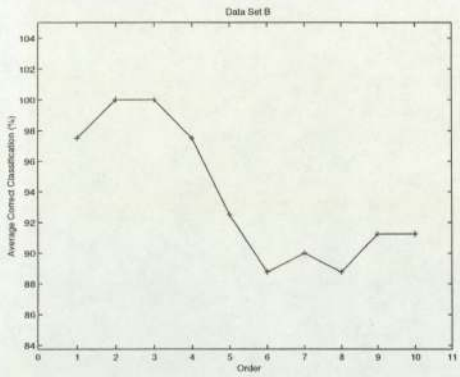

Figure 5.137: CPARCOR (RCS), Set B

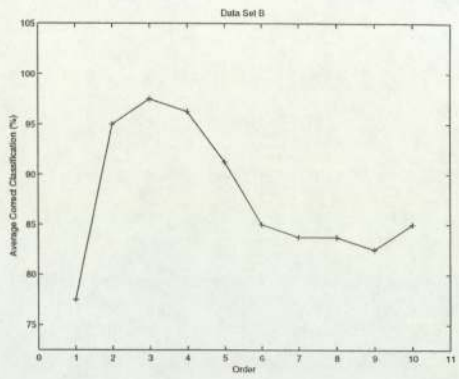

Figure 5.138: CPARCOR (RCS), Set B

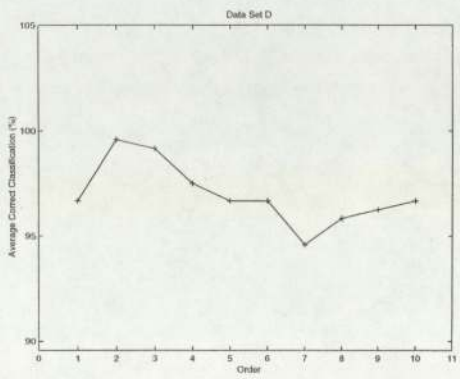

Figure 5.139: CPARCOR (RCS), Set D

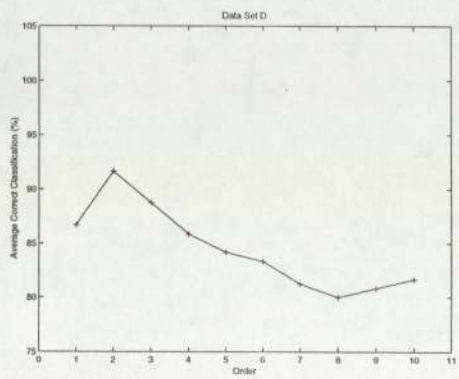

Figure 5.140: CPARCOR (RCS), Set D

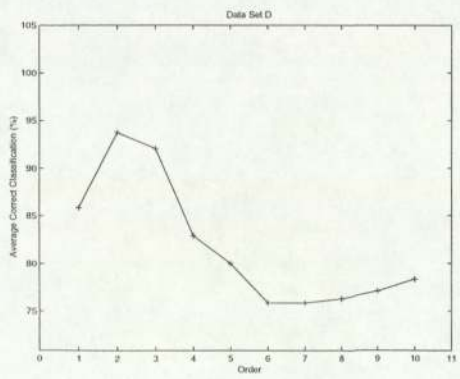

Figure 5.141: CPARCOR (RCS), Set D

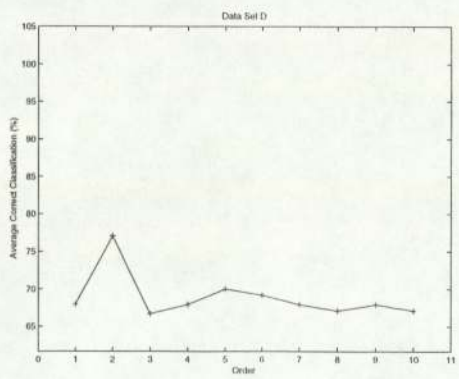

Figure 5.142: CPARCOR (RCS), Set D

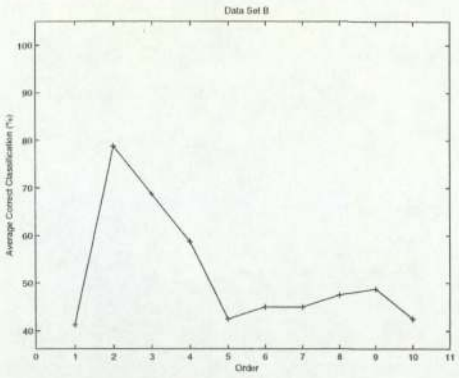

Figure 5.143: CPARCOR (RCS), Set B

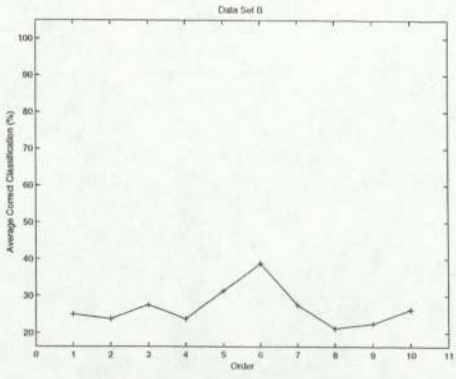

Figure 5.144: CPARCOR (RCS), Set B

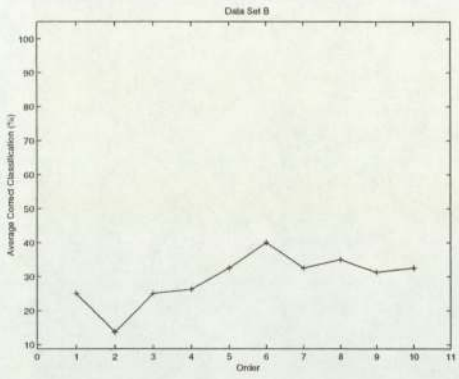

Figure 5.145: CPARCOR (RCS), Set B

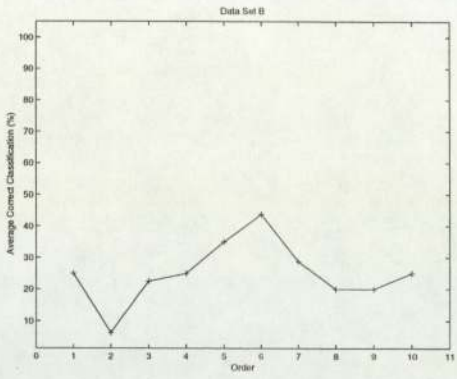

Figure 5.146: CPARCOR (RCS), Set B

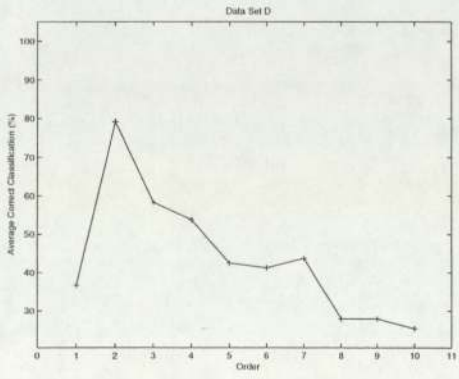

Figure 5.147: CPARCOR (RCS), Set D

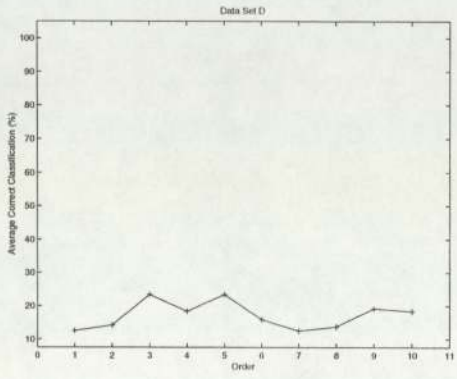

Figure 5.148: CPARCOR (RCS), Set D

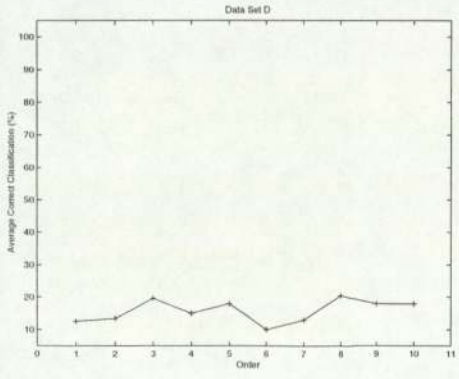

Figure 5.149: CPARCOR (RCS), Set D

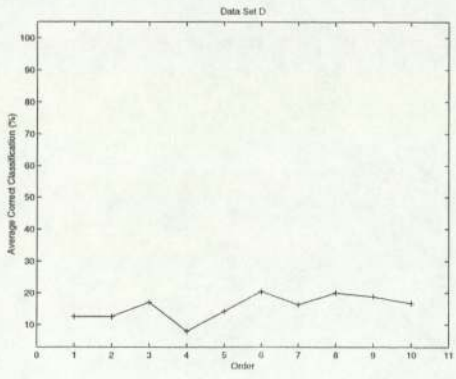

Figure 5.150: CPARCOR (RCS), Set D

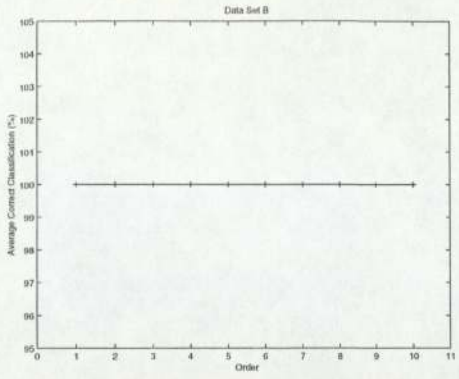

Figure 5.151: Direct CPARCOR (RCS), Set B

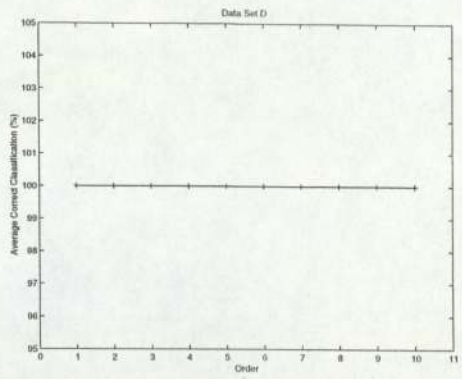

Figure 5.152: Direct CPARCOR (RCS), Set D

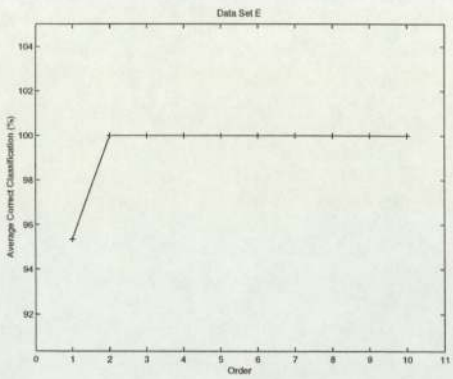

Figure 5.153: Direct CPARCOR (RCS), Set E

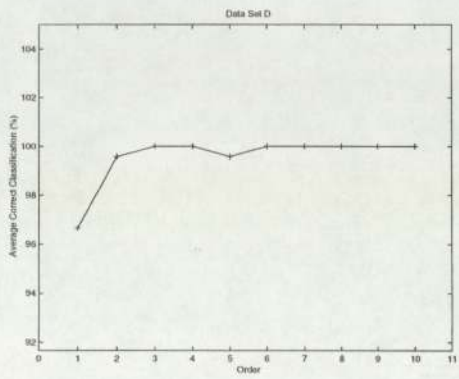

Figure 5.154: Direct CPARCOR (RCS), Set D

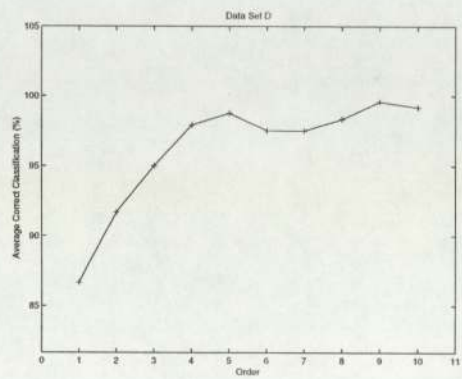

Figure 5.155: Direct CPARCOR (RCS), Set D

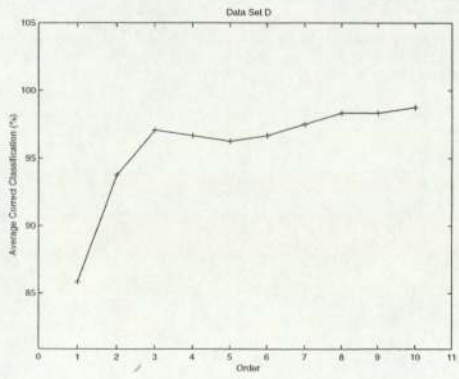

Figure 5.156: Direct CPARCOR (RCS), Set D

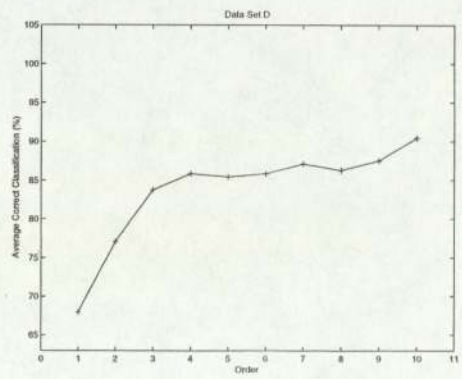

Figure 5.157: Direct CPARCOR (RCS), Set D

5.3.6 Fourier Descriptor (FD) methods

Introduction

The FD models described in subsections 4.2.1 - 4.2.4 include, radial, contour, curvature and affine-invariant FDs. The boundary sampling methods are computed by the radial, complex coordinate, curvature and complex coordinate angle of variation algorithms respectively. The boundary samples are treated as circular time series and the FD models are computed for each series and assigned class labels for use as feature vectors. In these experiments we used 64 spatially equi-distant samples.

Tables 5.8 and 5.9 show the range of Fourier descriptor experiments conducted. The numbers in the noise column refer to the indices on the x-axes of the corresponding figures. The graphs show that all of the Fourier descriptor techniques, with the exception of the curvature FD, yield remarkably high classification performances.

For each type of FD model, each figure shows the ACC performances of the FD models applied to each of the data sets individually, with two different amounts of training data on each plot. The x-axis labels are indices for the data sets listed in the titles. The suffix '2' attached to a data set label indicates that the larger amount of training data was used. Tables 5.8 and 5.9 refer to the precise size of each training set.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Radius FD	All	see AR set sizes		RCS	-/-	5.160
Contour FD	All	see AR set sizes		RCS	-/-	5.161
A-Invariant FD	All	see AR set sizes		RCS	-/-	5.162
Radius FD	All	see AR set sizes		kNN	-/-	5.163
Contour FD	All	see AR set sizes		kNN	-/-	5.164
A-Invariant FD	All	see AR set sizes		kNN	-/-	5.165
Curvature FD	All	see AR set sizes		kNN	-/-	5.166
Radius FD	Set B	40	20	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.167
Radius FD	Set D	50	30	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.168
Contour FD	Set B	40	20	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.169
Contour FD	Set D	50	30	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.170
A-Invariant FD	Set B	40	20	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.171
A-Invariant FD	Set D	50	30	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.172
Curvature FD	Set B	40	20	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.173
Curvature FD	Set D	50	30	kNN	1: $N(0,0.5^2)/N(0,0.5^2)$ 2: $N(0,1)/N(0,1)$ 3: $t_4(0,1)/t_4(0,1)$ 4: $t_{40}(0,1)/t_{40}(0,1)$	5.174

Table 5.8: Fourier descriptor experiments

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Radius FD	Set B	40	20	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.175
Radius FD	Set D	50	30	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.176
Contour FD	Set B	40	20	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.177
Contour FD	Set D	50	30	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.178
A-Invariant FD	Set B	40	20	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.179
A-Invariant FD	Set D	50	30	kNN	1: -/$N(0, 0.5^2)$ 2: -/$N(0, 1)$ 3: -/$t_4(0, 1)$ 4: -/$t_{40}(0, 1)$	5.180

Table 5.9: Fourier descriptor experiments

kNN classifier

- No noise on boundaries (all data sets)

  Figures 5.163-5.166 show the results of each of the tests, on all data sets, using each of the four FD techniques. The radial, contour and affine invariant FD methods show ACC performances of 100% for all data sets, except for set E where the higher dimensional contour FD feature vector required more training data. Figure 5.158 shows the first three principal components of the radial FD feature vectors extracted from shape set B. The classes are very well separated compared to the radius AR method 2 example in figure 5.28.

  The curvature FD graph shows lower ACC levels $\simeq 80\%$ even with high levels of training data.

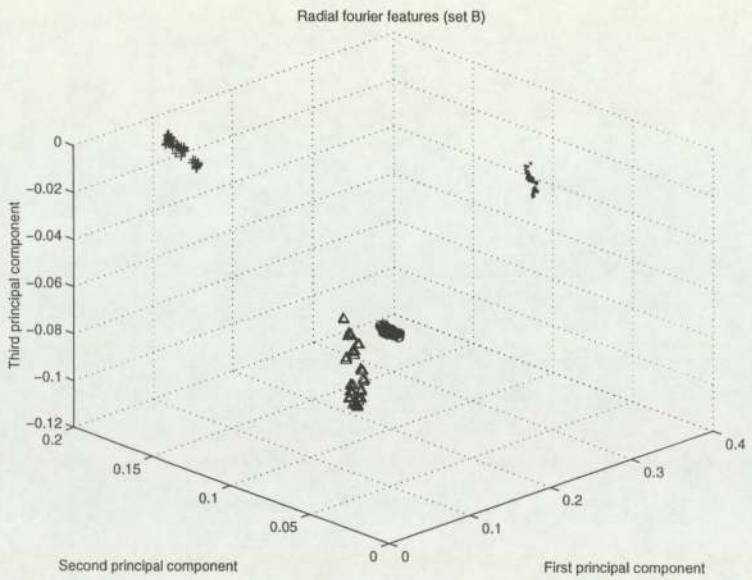

Figure 5.158: Scatter plot of the first 3 principal components of the radial FD feature vector fitted to set B (no boundary noise). The dots represent shape B1, the crosses represent shape B2, the circles represent shape B3 and the triangles represent shape B4.

- Noise on boundaries of test and train sets (set B and D)

  Figures 5.167 - 5.174 plot ACC performance against noise level for each of the data sets, B and D. The noise level index is described in table 5.8.

  The radius FD ACC levels were 100% for all levels. Even with the noise sampled from a $t_4(0,1)$ distribution the ACC was $\simeq 98\%$ for both sets B and D. Figure 5.159 shows the first three principal components of the radial FD feature vectors extracted from shape set B, with noise ($N(0, 0.5^2)$) added to the boundaries. The clusters are slightly less tightly bound owing to the addition of the boundary noise but they are still highly separated.

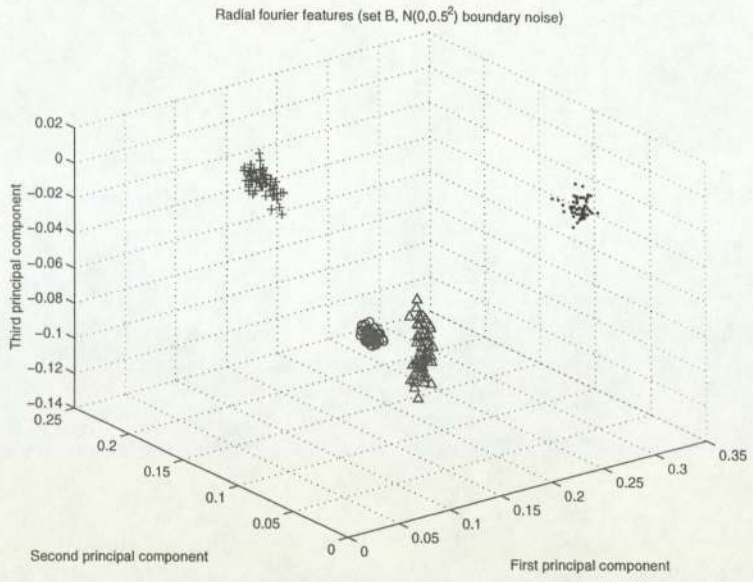

Figure 5.159: Scatter plot of the first 3 principal components of the radial FD feature vector fitted to set B ($N(0, 0.5^2)$ boundary noise). The dots represent shape B1, the crosses represent shape B2, the circles represent shape B3 and the triangles represent shape B4.

The affine-invariant and contour FDs classified set B, in all noise levels, completely correctly. The results for set D were slightly lower by $1 - 2\%$. This could be because the contour and affine-invariant feature vectors have almost twice as many dimensions as the radial FD feature vector, therefore requiring more training data to characterise each class.

The curvature FD model was completely corrupted by the addition of all noise levels for both data sets, as the ACC performances are no better than random.

- Noise on boundaries of test set only (sets B and D)

  Figures 5.175-5.180 show the results for the radius, contour and affine-invariant FD tests on sets B and D. The noise level index is described in table 5.9. Again all models classify set B 100% correctly for all noise levels.

  Set D however, shows some differences between the techniques. The radius FD classifies set D correctly for all noise levels apart from $t_4(0, 1)$ distributed noise, where the the ACC falls to 98%. The affine-invariant FD experiments show better ACC results than the contour FD experiments for all of the noise levels considered. It is thought that these two FDs again require more training data then the radial FD to fully evaluate their potential.

Gaussian classifier

- No noise on boundaries (all data sets)

  The Gaussian classifier gave very poor classification results when applied to all of the FD tech-

niques (figures 5.160 - 5.162), with noise free boundaries. The results show very poor performances as often the covariance matrix was near to, or completely singular. This could be due to the high number of parameters (62 for the contour FD feature vector and 32 for the radial FD feature vector) compared with the number of training samples, resulting in a poor covariance estimate. The low performance could also be due to the higher frequency Fourier coefficients having very small magnitudes in comparison with the low frequency coefficients and only varying by tiny amounts. This can also cause the covariance matrix to become singular.

Summary

The FD techniques, apart from the curvature FD technique, gave excellent classification levels in excess of 90% for all noise conditions on data sets B and D. The kNN classifier showed superior ACC results than the Gaussian classifier due to the very small magnitudes of the high frequency Fourier coefficients, causing the covariance matrix to become singular. This shows that there are too many variables for the Gaussian classifier to estimate accurately with small training sets.

The radius FD gave the most robust performance using these data, the next best was the affine-invariant FD, closely followed by the contour FD technique. But the affine invariant and contour FD's have higher dimensional feature vectors and so require larger training sets to fully investigate their potential. These complex FD's contain extra phase information from the boundary sampling algorithm and so would be expected to perform better than the radial FD. This wasn't seen in these results because the radial FD classified everything correctly, so more 'complex' shapes may be needed to demonstrate the differences between techniques. Identifying the appropriate data is difficult and time-consuming, so further experiments are recommended.

The curvature FD technique gave results no better than random in high levels of noise, see figure 5.174, again because of the weaknesses of the curvature sampling method. These results provide further evidence that curvature sampling methods are very susceptible to boundary noise.

Fourier descriptors are global approximators and shapes in set B are visually very different, hence the well separated feature clusters in figures 5.158 and 5.159. These graphs can be compared with figures 5.28 and 5.29 where the AR features are not as well separated. This is because AR coefficients are local operators and so reflect the differences in local detail. This does not mean that AR features are not as useful for classification as more similar shapes will have very similar global detail and local detail will become more important for feature discrimination. This is difficult to demonstrate as the results for data set E show that both techniques classify it 100% correctly, although the Fourier descriptors contain many more feature coefficients.

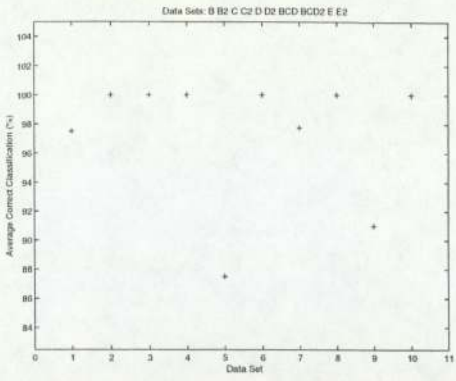

Figure 5.160: Radial FD (RCS)

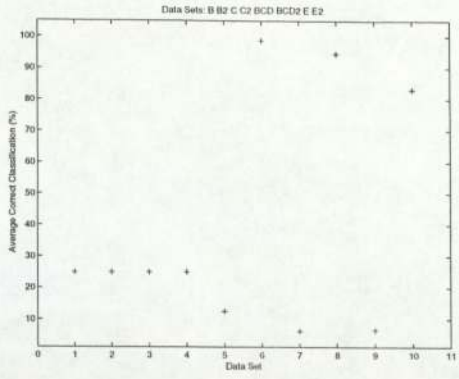

Figure 5.161: Contour FD (RCS)

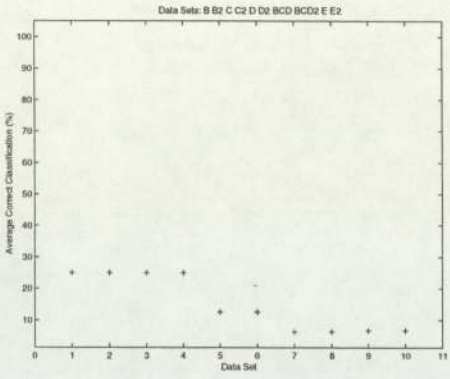

Figure 5.162: Affine Invariant FD (RCS)

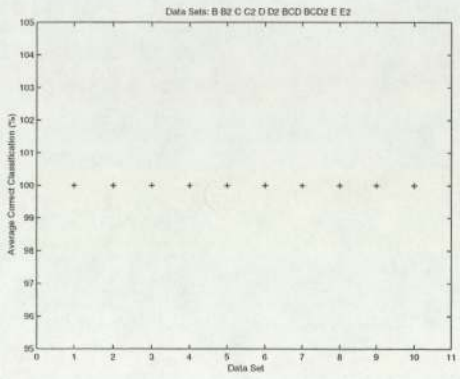

Figure 5.163: Radial FD (kNN)

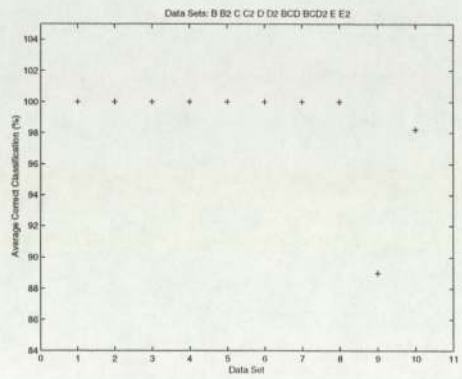

Figure 5.164: Contour FD (kNN)

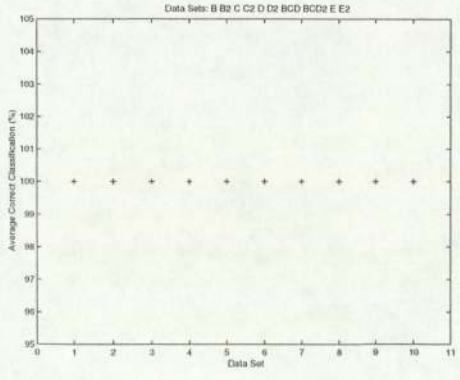

Figure 5.165: Affine Invariant FD (kNN)

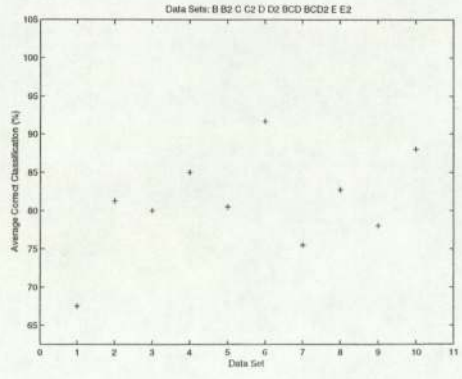

Figure 5.166: Curvature FD (kNN)

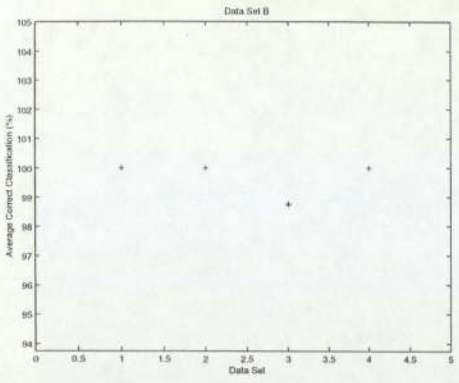

Figure 5.167: Radius FD (kNN)

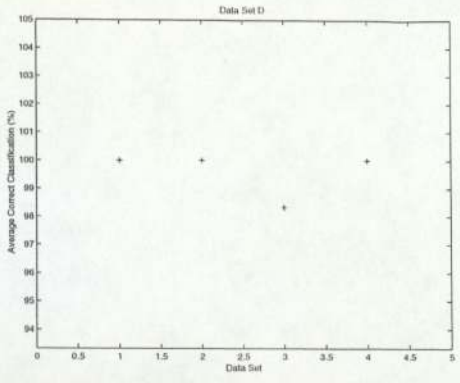

Figure 5.168: Radius FD (kNN)

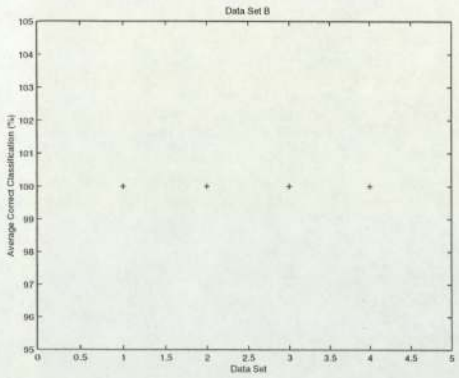

Figure 5.169: Contour FD (kNN)

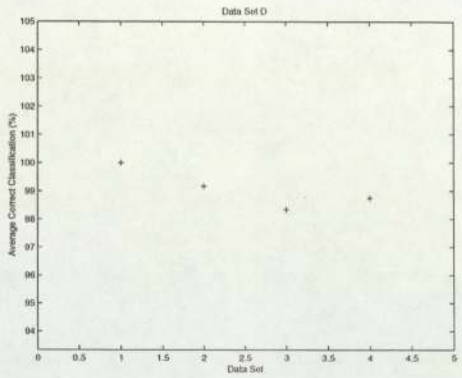

Figure 5.170: Contour FD (kNN)

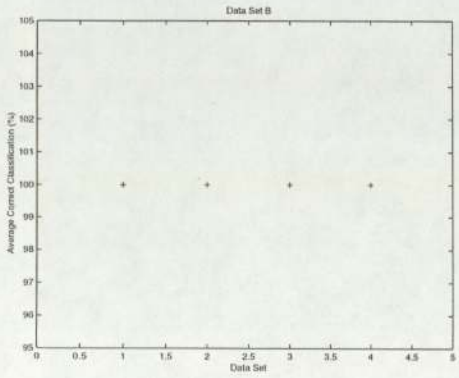

Figure 5.171: Affine Invariant FD (kNN)

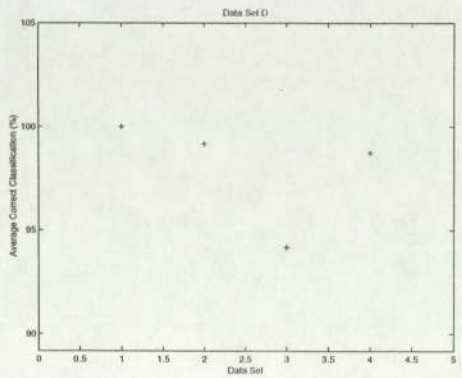

Figure 5.172: Affine Invariant FD (kNN)

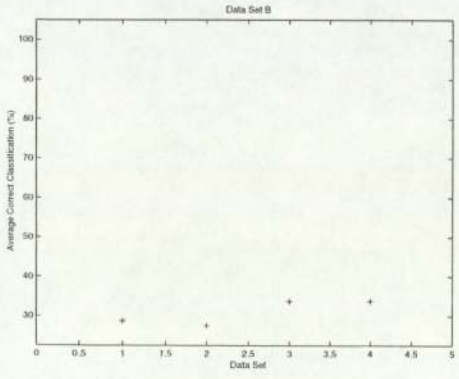

Figure 5.173: Curvature FD (kNN)

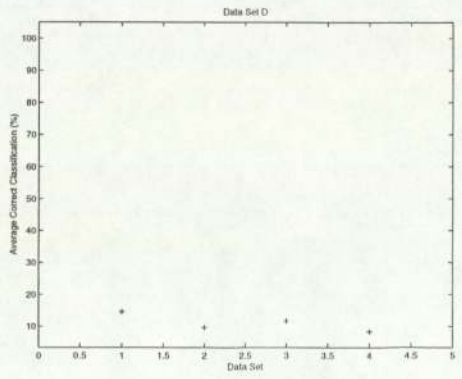

Figure 5.174: Curvature FD (kNN)

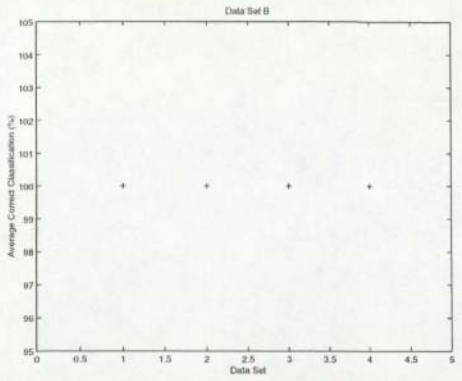

Figure 5.175: Radius FD (kNN)

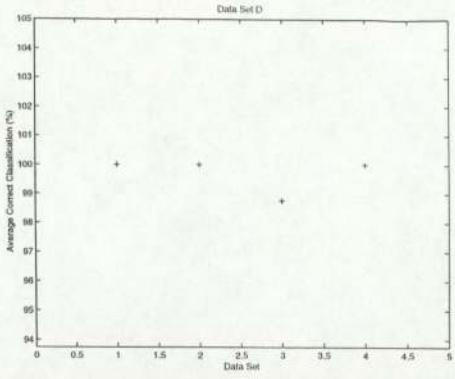

Figure 5.176: Radius FD (kNN)

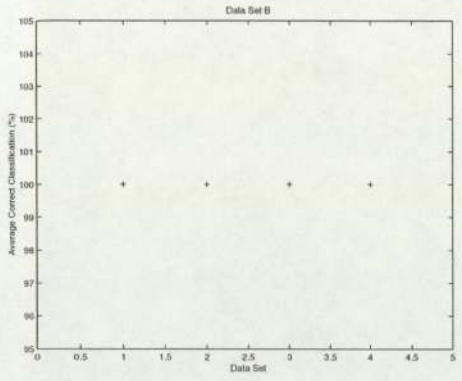

Figure 5.177: Contour FD (kNN)

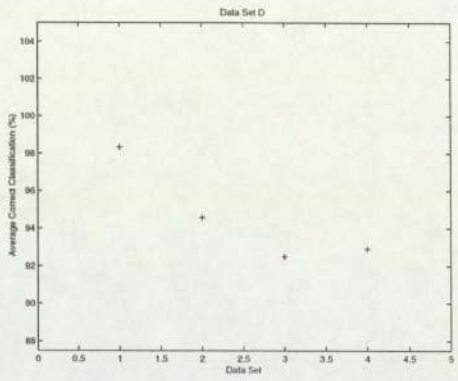

Figure 5.178: Contour FD (kNN)

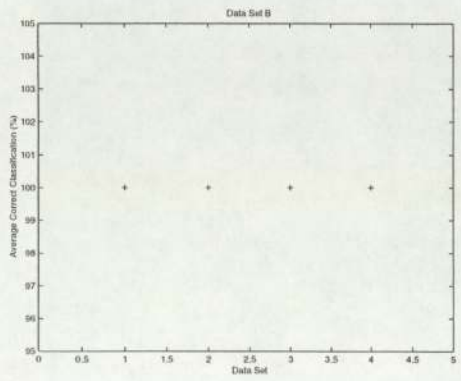

Figure 5.179: Affine Invariant FD (kNN)

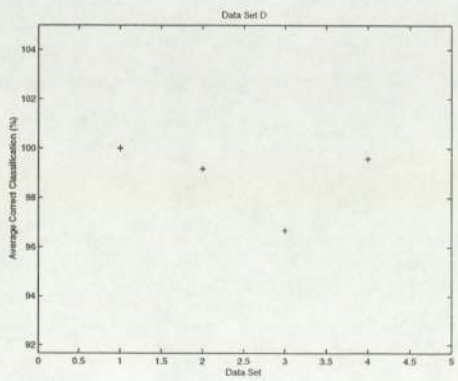

Figure 5.180: Affine Invariant FD (kNN)

5.3.7 Spectral AR method

Introduction

Spectral analysis is concerned with estimating the spectrum of a stationary stochastic process. Many stationary stochastic can be approximated by an AR process of sufficiently high order. Eom [6] used an AR(20) model to demonstrate this technique. The spectral AR method of subsection 4.3.7 uses the polar angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The polar coordinates are measured at N equi-spaced positions around the boundary providing uniform spatial coverage. Similarly to the complex coordinates, the polar coordinates allow the polygonal approximation of the original boundary sequence to be reconstructed. However, it is thought that the angular data would be affected more in noisy boundary conditions, but no empirical evidence exists in the literature. The boundary samples are treated as two circular time series, one containing radial distance samples, the other angle samples. AR models are fitted to both series and the roots of the corresponding AR polynomials, $A(z)$, are computed. The MATLAB function $\langle roots.m \rangle$ was used to compute the roots. The roots are related to the spectral peaks which were suggested by Eom to be useful features for shape classification. The spectral peaks of an AR model are known to be very narrow so a high order model, with many roots, is required to detect the peaks. Increasing the order of the model to 25 and then 30 gave no further improvement in classification so AR models of order 20 were used, as suggested by Eom. In these experiments we used 32 spatially equi-distant samples as increasing the samples further had little or no effect on the classification performance.

The feature vectors were split into testing and training sets and Gaussian, kNN and multilayer perceptron (MLP) classifiers are compared. The MLP is used as Eom [6] demonstrated his model using this classifier. We used the scaled conjugate gradients (SCG) algorithm [18] to optimise the weights and early stopping was used to terminate the training process. Sigmoidal functions were used in the hidden and output layers. In hindsight, perhaps a more sensible activation function to use in the output layer for this multi-class problem would have been a softmax function, which normalises the outputs. Eom used 4p+1 nodes in the hidden layer, assuming 2p inputs, so the same number were used in these experiments. As there are forty inputs (see section 4.3.7), eighty one nodes were used in the hidden layer. This is thought to a very large number of nodes but no mention of any regularisation techniques, such as weight decay for example, was mentioned by Eom. Regularisation controls the complexity of a neural network during the training stage, making the generalisation performance less sensitive to the initial model complexity. To try to follow the technique as closely as possible the same number of hidden nodes were used and early stopping was implemented as the termination criteria for the training of the MLP.

For these experiments we again have a large number of parameters to estimate in the feature vectors

and a relatively small training set. Eom used just forty training samples and forty test samples to estimate the weights in the network, claiming an average correct classification rate of 96.3% on set D in a boundary noise free experiment (compare with figure 5.182). Eom does not however mention a validation set for the MLP and also omits to specify the number of samples that were taken from the boundaries.

Figures 5.181 - 5.183 present ACC results for a range of noise environments on data sets B and D. Table 5.3.7 describes the experiments performed.

Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
B	50	25	RCS/ kNN/ MLP	1: -/- 2: $N(0,0.5^2)/N(0,0.5^2)$ 3: $N(0,1)/N(0,1)$	5.181
D	50	25	RCS/ kNN/ MLP	1: -/- 2: $N(0,0.5^2)/N(0,0.5^2)$ 3: $N(0,1)/N(0,1)$	5.182
D	150	75	RCS/ kNN/ MLP	1: -/- 2: $N(0,0.5^2)/N(0,0.5^2)$ 3: $N(0,1)/N(0,1)$	5.183

Table 5.10: Spectral AR experiments

Analysis

Figures 5.181 and 5.182 show the ACC performances of each of the classifiers in a range of noise environments.

The Gaussian classifier showed the highest and most robust performance levels, having a 90% ACC level with no boundary noise for set B and 78% for set D. This level rapidly declined to less than 60% for set B and 35% for set D when only a low level of noise ($N(0,0.5^2)$) was added to the boundaries of the test and train sets.

The MLP was the next best classifier, giving an ACC performance 80% and 68% for sets B and D, with no boundary noise present. The low performance is most probably due the combination of the lack of training data and the high complexity of the MLP.

The kNN classifier showed the worst ACC levels, being on average 20% lower than the Gaussian classifier in all noise conditions.

The poor performances of the spectral AR technique displayed in figures 5.181 and 5.182 is thought to be due to inadequate amounts of training data. Figure 5.183 shows this to be the case, as the train and test sets were increased from 50 and 25 samples to 150 and 75 samples respectively. The MLP's validation set was also increased from 25 to 75 samples. The performances of all of the classifiers improved, but the MLP ACC was much closer to the ACC levels of the Gaussian classifier. This is still a relatively small training set, but the increase in performance is significant and suggests that

a much larger data set than quoted in [6] is required to achieve a similar level of classification using an MLP classifier. In order to verify the implementation of Eom's technique a data set of eight aircraft taken from the article was used, with eight hundred training samples per class, four hundred test samples per class and four hundred samples per class in the validation set. The average correct classification performance was 93%, compared with 97.8% achieved by Eom.

Shape B1 was sampled eleven times, at different orientations, and the roots of the AR coefficients were plotted. Figures 5.184 and 5.185 show plots of typical root values computed from the AR coefficients that were estimated from the radial and angular sequences using noise-free boundaries, respectively. The roots of the AR coefficients estimated from radial distances are more tightly clustered than the equivalent roots estimated from the angular data. This indicates that the angular information is not as useful a feature as the radial distance information and was verified in a separate test where only the roots estimated from radial distance sequences were used as a feature vector. The test classified set B without any addition of boundary noise using an MLP and was only 5% lower than the results shown in figure 5.181. Figure 5.185 shows that the polynomial computed using the angular AR coefficients is ill-conditioned.

If noise is added to the boundaries then an ill-conditioned polynomial will have widely differing roots. This is shown in both figures 5.186 and 5.187 where $N(0, 0.5^2)$ noise is added to the boundaries. This accounts for the low performance of the spectral AR method in noisy conditions.

Summary

With the low levels of training data compared with the number of estimated parameters, as specified by Eom, the gaussian classifier gave the best ACC performances in all noise conditions.

When the amount of training data was increased, the ACC of both data sets increased significantly. The MLP gave an ACC level of 81% for set D, compared with 85% for the Gaussian classifier. However the MLP was least affected by noisy boundary conditions, although none of the classifiers performed very well. The roots taken from the AR polynomial fitted to the angular sequences were shown to be of little use for shape classification, with most of the discriminitive information contained in the roots of the AR polynomials fitted to the radius sequences. However, both radial and angular AR polynomials were shown to be extremely ill-conditioned in the presence of low amounts of boundary noise.

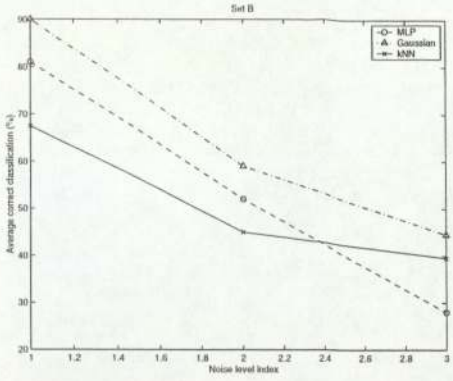

Figure 5.181: Spectral AR method, Set B

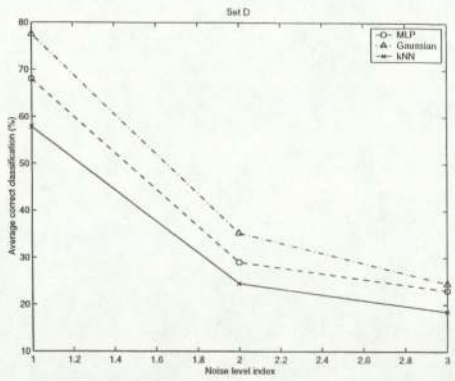

Figure 5.182: Spectral AR method, Set D

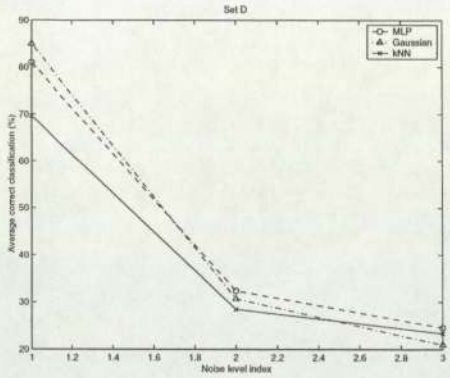

Figure 5.183: Spectral AR method, Set D

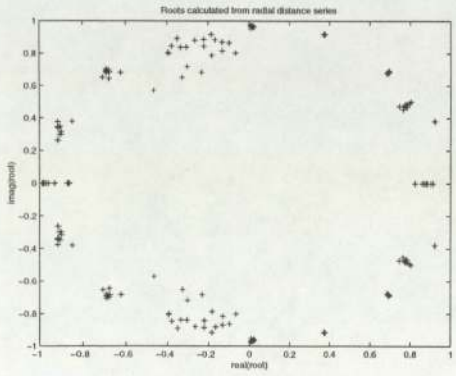

Figure 5.184: Roots estimated from $r(i)$, shape B1

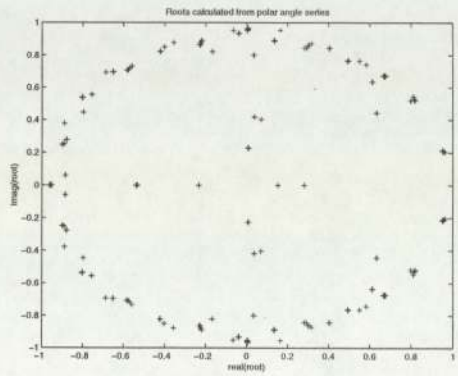

Figure 5.185: Roots estimated from $\theta(i)$, shape B1

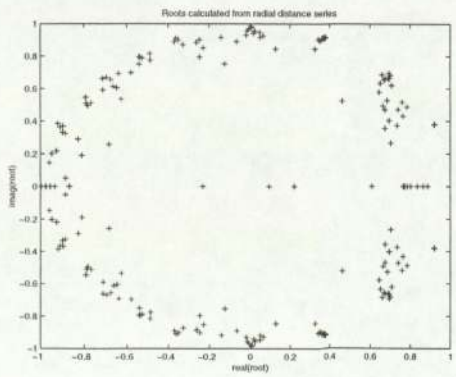

Figure 5.186: Roots estimated from $r(i)$, shape B1 with $N(0, 0.5^2)$ boundary noise

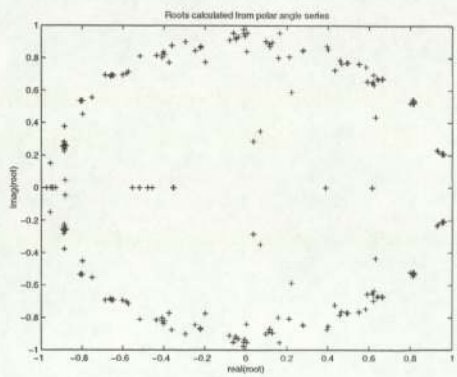

Figure 5.187: Roots estimated from $\theta(i)$, shape B1 with $N(0, 0.5^2)$ boundary noise

5.3.8 Non-stationary method

Introduction

The non-stationary AR method described in subsection 4.3.8 uses the radius angle of variation sampling technique to extract boundary information from a sequence of shapes from each class at different orientations. The boundary samples are treated as a circular time series and time varying AR models are fitted to each series, and after phase correcting for the position of the first sample point on the boundary (a phase difference exists between a reference shape, one from each class, and a corresponding shape of the same class but at a different orientation), the coefficients are assigned class labels and stored as feature vectors. In these experiments it was only necessary to use a model of order 1 and q (an empirically set parameter used to estimate the non-stationary mean sequence) was set to 1.

Sixty four spatially equi-distant samples were again found to represent the boundary, as increasing the sampling frequency had little or no effect on the classification. To demonstrate this technique, Paulik et al [4] used data set E, so sets B and E were tested in a range of noise environments, see figures 5.188 and 5.189. The indices on the x-axes label the experimental conditions shown in table 5.3.8.

Index	Size train set	Size test set	Noise (train/test)
1	50	50	-/-
2	50	50	$N(0,0.5^2)/N(0,0.5^2)$
3	50	50	$N(0,1)/N(0,1)$
4	50	50	$t_{40}(0,1)/t_{40}(0,1)$
5	50	50	$t_4(0,1)/t_4(0,1)$
6	50	50	$-/N(0,0.5^2)$

Table 5.11: Experimental conditions of figures 5.188 and 5.189

kNN classifier

- No noise on boundaries (sets B and E)

  Using the lowest number of parameters possible for this technique (m=1, q=1), the ACC levels for sets B and E were equal to 100%.

- Noise on boundaries of test and train sets (sets B and E)

  With the addition of noise to the boundaries the ACC levels fell dramatically to 70% for both sets B and E, for the lowest level of noise tested ($N(0,0.5^2)$). For higher levels of noise, the ACC levels continued to fall, although they decreased more rapidly for increased noise levels tested on set D. The non-stationary AR technique was not robust to the addition of noise to the boundaries in these conditions as the local mean estimates will be more variable, necessitating

a higher value of q (increasing the number of Fourier coefficients to estimate the non-stationary mean), and the boundary samples are more complex sequences requiring higher order models to minimise the residual error. Having observed the high levels of ACC achieved by the linear AR methods on both sets B and D, the drop in performance is greater than expected.

After fitting the non-stationary AR model to a boundary sequence the coefficients are phase corrected so that the feature vectors are invariant to the starting sample point on the boundary. Noise effectively perturbs all of the boundary points, so any starting point on a reference boundary will not be found on any of the test boundaries. The performance of the phase correction algorithm will therefore begin to degrade with increasing levels of boundary noise.

Summary

The Gaussian classifier produced very poor classification performances using these features. For example, the noise free test on set E gave an ACC of $\approx 30\%$. This was again due to the covariance matrix being near singular. The noise free boundary experiments show the high ACC performance achievable using a kNN classifier with this model, using only 5 parameters. However the ACC falls dramatically with the addition of noise to the boundary, for sets B and E, as the phase matching algorithm performance will degrade in these conditions.

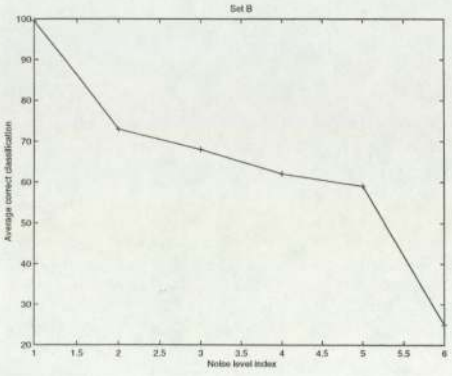

Figure 5.188: Non-stationary AR method, Set B

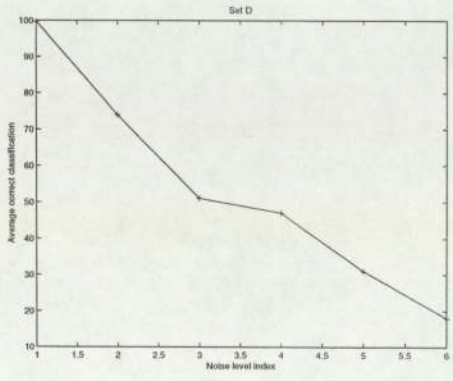

Figure 5.189: Non-stationary AR method, Set E

5.3.9 Wavelet moment-invariant method

Introduction

The wavelet moment invariant technique of subsection 4.4.3 samples the whole shape silhouette, so the hollow shape boundaries are filled-in using the MATLAB image processing function $\langle bwfill.m \rangle$. The image is transformed from a two dimensional cartesian system to a polar system. r and θ were incremented by 0.5 pixels from 0 to r_{max} and 0.5 degrees from 0 to 2π respectively, producing a 2-d array indexed by r and θ. The $S_q(r)$ function is computed by approximating the integral over θ (0-2π) using the MATLAB trapezoidal numerical integration function $\langle trapz.m \rangle$ to integrate over θ. The wavelet transform of $S_q(r)$ is computed for moments 0, 1,$\cdots$, 3 and the wavelet transform is computed for scales 1, 2, 4 and 8 (i.e. levels 0, 1,$\cdots$, 3) using the MATLAB wavelet toolbox function $\langle cwt.m \rangle$ (a discrete approximation of the continuous wavelet transform).

The feature vectors from each class were split into test and train sets, then the features were selected as shown by the flow chart in figure 4.2, and a Gaussian and kNN classifier are compared. Table 5.12 describes the experiments. Figures 5.192 - 5.195 show ACC levels plotted against the number of features chosen by the feature selection algorithm.

Model	Data	Size train set	Size test set	Classifier	Noise (train/test)	Figure
Wavelet	Set B	30	30	kNN	-/-	5.192
Wavelet	Set D	30	30	kNN	-/-	5.193
Wavelet	Set B	30	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.194
Wavelet	Set B	30	30	kNN	$N(0,0.5^2)/N(0,0.5^2)$	5.195
Wavelet	Set B	30	30	kNN	$-/N(0,0.5^2)$	5.196
Wavelet	Set B	30	30	RCS	-/-	5.197
Wavelet	Set D	30	30	RCS	-/-	5.198

Table 5.12: Wavelet moment-invariant experiments

kNN classifier

- No noise on boundaries (sets B and D)

  Figure 5.192 shows that an ACC level for set B of 100% was achieved with 9 features. Figure 5.193 shows the performance on set D. The ACC level increases monotonically with the number of features. The maximum number of 15 features were selected by the algorithm, giving an ACC of 98%.

- Noise on boundaries of test and train sets (set B only)

  Figure 5.194 shows the performance of the kNN classifier on set B when the boundary noise is characterised by a $N(0, 0.5^2)$ distribution. The ACC fell to 83% and a maximum of 7 features were determined by the feature selection algorithm.

  If the same train and test sets are used, but the selected features are chosen from the noise free boundary experiment using set B, figure 5.195 shows the increase in ACC levels. This indicates a weakness in the feature selection algorithm. Shen et al [15] suggest the use of the Branch and Bound algorithm [19] to guarantee an optimum feature set but will be computationally more demanding.

- Noise on boundaries of test set only (set B only)

  Figure 5.196 shows the ACC when the training set does not contain any noise. The features are the same as those in figures 5.192 and 5.195 as they were selected from the noise free training set.

Gaussian classifier

- No noise on boundaries (sets B and D)

  Figure 5.197 shows the ACC of the Gaussian classifier applied to set B. The level of classification is generally lower than the equivalent test using the kNN classifier. For seven features the covariance matrix was singular, causing the classifier to fail. This is because the features are constant for some of the shapes.

  Figure 5.198 shows the ACC of set D, which is generally lower than the equivalent tests using the kNN classifier in figure 5.193.

General analysis

An improvement could be made to in the approximation of $S_q(r)$ by using finer increments in the conversion from cartesian to polar coordinates, but would be computationally more expensive and the advantages of doing so would be limited by the resolution of the image. Figures 5.190 - 5.191 show the

difference between $S_0(r)$ values generated from, (a) a shape with no boundary noise and (b), a shape with $N(0, 0.5^2)$ noise added to the boundary. Both $S_0(r)$ sequences are similar and quite smooth (due to the coarse coordinate system conversion).

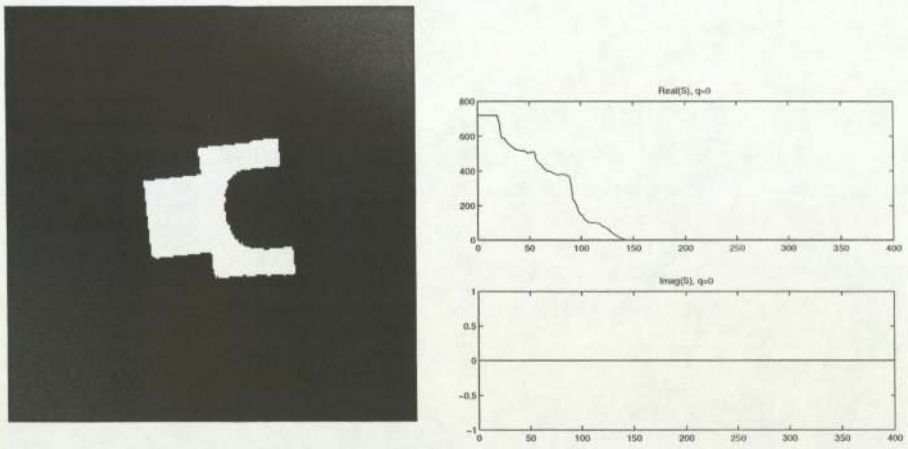

Figure 5.190: $S_0(r)$ plotted against r, estimated from a non-noisy shape

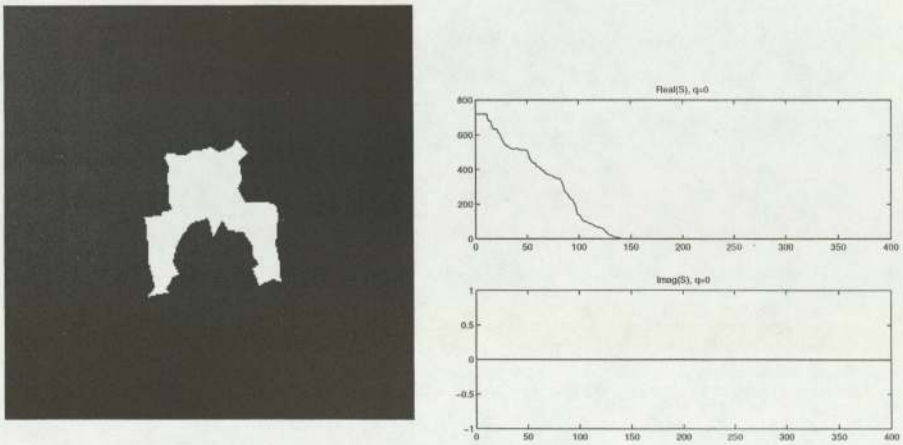

Figure 5.191: $S_0(r)$ plotted against r, estimated from a noisy shape

If the classifier is to be able to differentiate between similar shapes belonging to different classes it would be desirable to maximise the resolution of $S_q(r)$. As wavelet transforms provide good time resolution and poor frequency resolution at high frequencies (low scales) and good frequency resolution and poor time resolution at low frequencies (high scales), another improvement in the algorithm would be to consider higher scales so that lower frequency components of $S_q(r)$ could be exploited. This would provide enhanced capabilities to recognise noisy and noise-free shapes belonging to the same class, allowing the identification of the global shape of $S_q(r)$ rather than the local detail.

Summary

The kNN classifier was shown to provide better ACC levels than the Gaussian classifier on these data. The wavelet moment invariant method has been shown to have potential for shape classification, but requires a more sophisticated feature selection algorithm than the one implemented in this project, such as the Branch and Bound algorithm for example, or even a multivariate algorithm (e.g. canonical variates or PCA). Further improvements were identified such as increasing the accuracy of the coordinate system conversion and also increasing the number of scales, so that more global information can be used to recognise noisy shapes belonging to the same class.

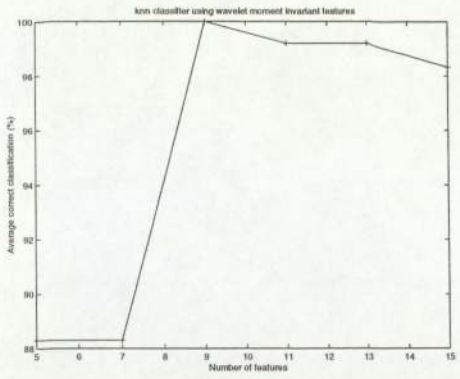

Figure 5.192: Wavelet M. Invariant (kNN), Set B

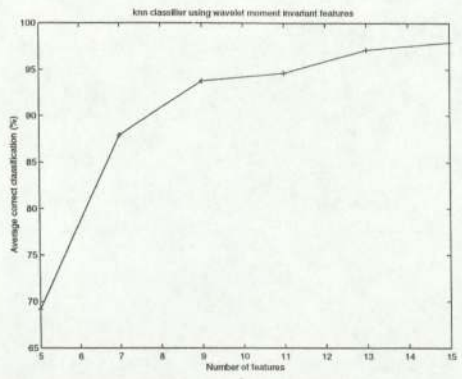

Figure 5.193: Wavelet M. Invariant (kNN), Set D

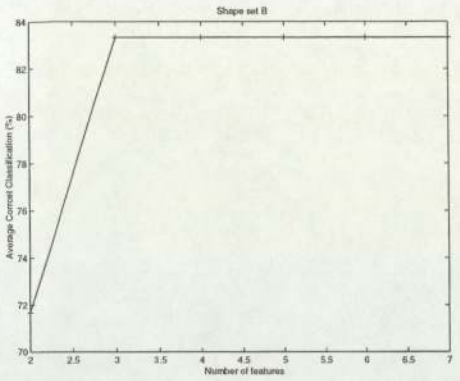

Figure 5.194: Wavelet M. Inariant (kNN), Set B

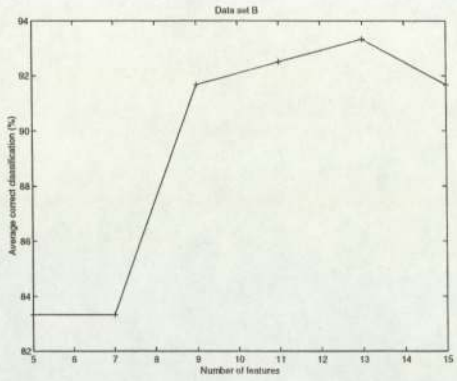

Figure 5.195: Wavelet M. Invariant (kNN), Set B

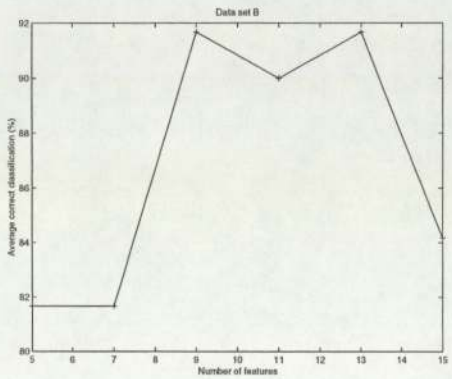

Figure 5.196: Wavelet M. Invariant (kNN), Set B

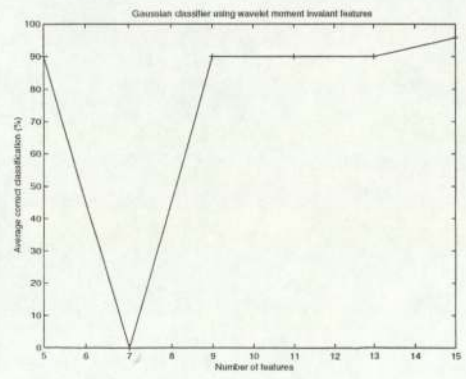

Figure 5.197: Wavelet M. Invariant (RCS), Set B

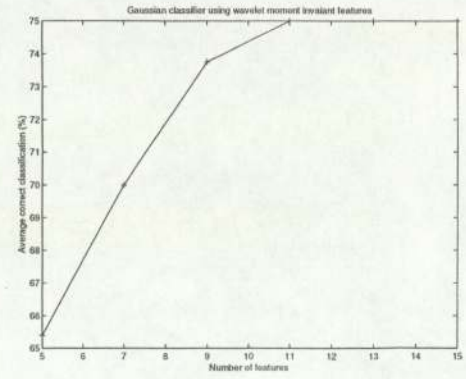

Figure 5.198: Wavelet M. Invariant (RCS), Set D

Chapter 6

Conclusions and future work

In this concluding chapter, the aims of this project are examined, the findings of this project reviewed, conclusions are drawn and future work is discussed.

This project is part of an ongoing study at DERA carried out in collaboration with Dr. R. H. Glendinning and aims to address the following weaknesses of earlier work on shape based classification techniques for near real-time applications:

1. comparisons between technqiues are not comprehensive;
2. they use relatively small sets of self selected shapes;
3. little attention is paid to describing realistic noise;
4. varying environmental conditions are often ignored.

The result of this is that little advice can be given to prospective shape classification engineers or researchers on the relative importance of the various components and techniques of the boundary based shape classification process. Specifically, this project aimed to address:

1. the applicability of different models to describe boundary characteristics;
2. the relative effect of model complexity;
3. the value of different sampling algorithms;
4. the value of different classifiers;
5. computational complexity of the different models.

Attention is paid to the interaction between these issues, the particular shapes used in our experiments and the different noise models. The comparison of techniques was performed on an equal

basis and the metrics used include the average correct classification (ACC) performance and the number of floating point operations (flops), thereby considering both the performance potential and the computational cost.

6.1 Summary of results

To make the findings of this project clearer, we first present a selective summary of the results in tables 6.1 and 6.2. Table 6.1 provides a summary of the results in noise free boundary conditions, showing the maximum ACC performances of each technique when applied to the most challenging data set E (see page 40). Data set E is thought to be challenging because it contains many classes of both visually very different and also very similar shapes. The spectral AR and wavelet moment invariant techniques are tested on set D, which is a simpler subset of set E. The 'flops' column of table 6.1 shows the number of floating point operations (flops) to compute one feature vector at model order 4 for all of the linear AR-based models, order 20 for the spectral AR model and order 1 for the non-stationary AR model. The number of flops for the linear AR tests are represented by the Radius AR 2 value, as they are all approximately equal.

Table 6.2 summarises the best performances of each technique in noisy boundary conditions, where the first set of results considers noise-free training sets and noisy testing sets and the second set of results considers noisy testing and training sets. A noise free training set and a noise testing set is intended to demonstrate the performance of each model when no prior knowledge is available about any potential noise level or type. A noisy training and testing set is intended to demonstrate the performance of each model when the amount and type of noise is known. Note the high amounts of noise from a $t_4(0, 1)$ distribution added to the boundaries during these experiments. The ACC results for the non-stationary AR and wavelet moment invariant techniques are presented for the simpler shape set B for illustrative reasons.

6.2 Conclusions

The issues that this project aimed to address include the effects on shape classification of:

- the type of classifier

  The Gaussian classifier performed well with the AR, CAR and CPARCOR models because the coefficients were easily modelled by multivariate Gaussian distributions, see figures 5.28 and 5.158 for an example of the distributions of the first three principal components of AR features compared with FD features. The Gaussian classifier performed less well when the feature vectors were of a high dimension, as for FD's, as the covariance matrices were almost singular. The kNN

Feature	Classifier	Data	Order (no noise)	ACC %	Flops
Radius AR (1)	RCS	E	9	96	-
Radius AR (2)	RCS	E	4	100	42495
Curvature AR	RCS	E	8	78	-
Spectral AR	RCS	D	20	85	670447
Non-stationary AR	kNN	E	1	100	86603
CAR	RCS	E	4	100	27785
CPARCOR (recursive)	RCS	E	3	100	26781
CPARCOR (direct)	RCS	E	3	100	102053
Radius Fourier	kNN	E	-	100	158780
Curvature Fourier	kNN	E	-	89	158824
Contour Fourier	kNN	E	-	98	158917
A-invariant Fourier	kNN	E	-	100	294799
Wavelet moment invariant	kNN	D	-	97	7489613

Table 6.1: Summary of results: noise free tests

Feature	Classifier	Data $-/N(0,0.5^2)$	Order	ACC %	Data $(t_4(0,1)/t_4(0,1))$	Order	ACC %
Radius AR (2)	RCS	D	10	88	D	10	96
Curvature AR	RCS	-	-	-	D	4	20
Spectral AR	RCS	-	-	-	D $(N(0,1)/N(0,1))$	20	25
Non-stationary AR	kNN	B	1	25	B	1	60
CAR	RCS	D	5	100	D	10	95
CPARCOR (recursive)	RCS	D	2	80	D	2	77
CPARCOR (direct)	RCS	D	6	100	D	10	91
Radius Fourier	kNN	D	-	100	D	-	98
Curvature Fourier	kNN	D	-	-	D	-	12
Contour Fourier	kNN	D	-	100	D	-	98
A-invariant Fourier	kNN	D	-	100	D	-	94
Wavelet moment invariant	kNN	B	-	92	B $(N(0,0.5^2)/$ $N(0,0.5^2))$	-	93

Table 6.2: Summary of results: noise tests

classifier, with k=3 and a Euclidian distance metric, performed well on the well separated high dimensional features of the FD's and struggled to correctly differentiate between the closer and more cloudy AR feature centres. The multi-layer perceptron (MLP) was only tested on spectral AR features where the radial AR roots were classified equally as well by a Gaussian classifier. Figure 5.184 shows the Gaussian-like centres of the radial AR roots.

- the model type and order

Noise free boundary conditions: Low order linear AR models (1-10) have been shown to be able to classify set E $\sim$ 100% correctly using the radial angle of variation method. This was a surprising result as performances shown in the literature [4] were much lower using the same data, typically 70%, due to the lack of training data used. The feature vector was also very fast

to compute, see table 6.1.

The spectral AR model used the polar coordinate angle of variation sampling method and estimated the roots of the AR polynomials fitted to the radius and angle sequences to detect the peaks in the AR spectra. The roots estimated from the angle series varied wildly with each realisation of a shape, resulting in a feature vector that was of little use for classification. The model did however achieve a ACC of 85% for set D, where most of the discriminatory information came from the AR polynomial of the radius sequence. The feature vector typically required ~ 14 times more flops to compute than the linear AR model.

The non-stationary AR model classified set E 100% correctly in its least complex form (m=1, q=1), although it required twice as many flops to evaluate as the linear AR model.

The CAR and CPARCOR (estimated both directly from the CAR coefficients and also recursively) models classified all of the data sets, including set E, 100% correctly. The CAR and CPARCOR features estimated recursively typically required 33% less flops than the linear AR models and require the least number of flops to compute of all.

The radial, contour and affine invariant FD models, using the radial and complex angle of variation sampling schemes classified all of the data, including set E, 100% correctly. The radius and contour FD feature vectors required 4 times more flops than the linear AR feature vectors and the affine invariant feature vectors typically required 8 times that amount.

The wavelet moment invariant technique is a silhouette based method and is accordingly much less efficient to compute. Compared with the CAR model it requires ~ 270 times more flops to evaluate a feature vector due to the hundreds of wavelet coefficients that are computed. The ACC of set D was $\sim 97\%$. The model was not tested on set E, which has 15 classes, due to the large amount of computing time required to do so. The complete set of features for set D, which has just 8 classes, took approximately 12 hours to compute on a DEC Alphastation 500 using MATLAB. The univariate feature selection algorithm was required to reduce the size of the feature vector by choosing the most discriminatory coefficients. However the univariate feature selection algorithm was shown to be sub-optimal and multi-variate alternatives, such as PCA and canonical variates, have been suggested.

Noisy boundary conditions: The tests where the training set was noise free and the testing set contained noise generally showed the same ACC performance trends as tests where both the training and testing sets contained noise, but at a lower level.

The curvature based AR and FD models performed poorly due to the sensitivity of the sampling algorithm.

The spectral AR model used the polar coordinate angle of variation sampling method and

estimated the roots of the AR polynomials fitted to the radius and angle sequences to detect the peaks in the AR spectra. During the boundary noise tests both the radius and angle polynomials became more 'ill-conditioned' and the ACC dropped to 60% even for the simple set B. The spectral AR model is therefore inappropriate for any application in adverse noise conditions.

The non-stationary model produced very low ACC performances even for the small and visually simple set B. This was because the phase matching algorithm adjusts the phase of the feature vector so that the features are invariant to the location of the starting point on the boundary, making the feature vector completely rotation invariant. If boundary noise is introduced it will corrupt the model resulting in a much lower ACC performance. The non-stationary AR model is therefore inappropriate for any application in adverse noise conditions.

The wavelet moment invariant ACC performance for set B was robust in noisy boundary experiments but the best performance was difficult to quantify due to the poor univariate feature selection algorithm.

The radial linear AR model proved to be robust to noise when the same amount and type was included in the training and testing sets, but degraded when the training set was noise free and the testing set contained noise.

The CAR and CPARCOR (estimated directly from the CAR coefficients) models provided much more robust ACC performances in noisy boundary conditions, which was attributed to the phase information contained in the complex coordinate samples. The CPARCOR coefficients estimated recursively proved to be much less robust because the recursive algorithm is thought to be an approximation to the direct method.

All of the FD models, with the exception of the curvature FD, classified all of the data sets $\sim 100\%$ correctly, even in high boundary noise conditions. The FD's were slightly more robust to boundary noise than the CAR model, particularly in high noise when the testing set was noise free and the training set contained boundary noise, where the ACC performances were $\sim 15\%$ better.

- the boundary sampling algorithm

The curvature and polar coordinate sampling methods both estimated angular measures around the boundary and the ACC's of the AR and FD models were typically $80 - 90\%$ in noise free conditions (see table 6.1). Table 6.2 shows that the spectral AR, curvature AR and FD models performed poorly in boundary noise tests. These results are not unexpected due to the noise sensitive curvature measures but have not been demonstrated in the literature. The polar coordinate angle of variation method did not prove to be successful due to the ill-conditioning

of the AR polynomials computed from the angular data. The radial data AR polynomial gave ACC results equal to those using both radial and angular data, indicating that the angular data provided no extra discriminatory information. However, in the presence of noise, both radial and angular AR polynomials became 'ill-conditioned'. Therefore curvature and polar coordinate sampling algorithms are not suitable for application in adverse noisy boundary conditions.

The next most successful sampling algorithms were equal angle and radial angle of variation sampling. The equal angle sampling method was shown to perform less well than the radial angle of variation method with linear AR features on noise free boundary tests. This is because the equal angle sampling method can produce varying numbers of samples at different boundary orientations, especially if the boundary is non-convex. The angle of variation method however provides uniform spatial coverage resulting in a constant number of sample points, which is a more consistent method when the shapes are non-convex.

The complex coordinate angle of variation method provided the best ACC results for linear AR and FD models, especially in adverse boundary noise conditions. Refer to table 6.2, features CAR, CPARCOR and A-invariant FD. The complex coordinate feature vector contains phase information, allowing the original polygonal boundary approximation to be reconstructed whereas the radial distance samples could have resulted from more than one boundary shape.

- the range of data

  Data set E, having fifteen classes with both similar and dissimilar shapes proved the hardest set to correctly classify, especially in noisy boundary conditions. Data set BCD contained sixteen classes but was generally more easily correctly classified as the shapes are all visually dissimilar.

To summarise, the main points to note from these experiments using these particular data sets of shapes are:

- the suitability of the kNN Euclidian distance classfier for use with high dimensional FD's;
- the suitability of the Gaussian classifier for use with low order (m<10) AR features, including AR, CAR and CPARCOR coefficients;
- linear AR models are more powerful for shape classification than suggested in the literature and more complex models are not necessarily an improvement, especially in noisy conditions. Specifically, the spectral AR model involves estimating the roots of AR polynomials which become ill-conditioned in the presence of boundary noise. The non-stationary AR model also performs poorly in boundary noise conditions due to the failure of the phase matching procedure;
- confirmation of [2] of the similarity of the CAR and CPARCOR (estimated by the directly from the CAR coefficients) models average correct classifications in noise free environments;

- the similarity of the CAR and CPARCOR models giving high ACC levels in high noise environments, higher than the real AR models;
- the excellent performance of the FD models in noise-free and noisy boundary conditions using all of the Fourier coefficients, although they are much less efficient than the CAR model to compute;
- the benefit of phase information in sampling, provided by the complex coordinate angle of variation sampling technique;
- empirical evidence of the poor performance of curvature sampling as shown by both AR and FD features.

6.3 Future work

The ultimate aim of this on-going project at DERA is to identify the strengths and weaknesses of many leading and new boundary based shape classification models and to select the more robust methods for use in real applications, of which there are many, including target tracking and condition monitoring. Current and future work will expand on the experiments described here to:

- measure the effects of occlusion on each technique and also investigate techniques designed to cope with occlusion [21], [22]
- consider the robustness of non-linear AR models [16], [23]
- investigate boundary based wavelet techniques [24], [25], as they could provide more robust models than FD's as they combine both local and global information about a signal.

together with

- the development of feature selection algorithms and further classifiers
- investigating the potential of using subset selection to model each class with an AR model, instead of fitting a fixed model to each of the classes [26]. AR models are universal approximators, as are FD's and wavelets. An investigation into subset selection using different model types to model each class could extend this work further.
- the improvement of the selection algorithm of the wavelet moment invariant model
- endeavouring to find more challenging data sets to demonstrate the characteristics of each technique more clearly

- investigating the effects of testing and training with boundaries containing different levels of noise
- exploring the robustness of the FD's. It is believed that the FD's could perform poorly with many similar shapes having small differences, as the FD's are global operators (i.e. they do not provide any local information), whereas AR models are local operators.

The output of this work will hopefully result in a journal publication [27] and a boundary based shape classification toolbox, written in MATLAB.

Bibliography

[1] J Ramesh R Kasturi B Schunck. *Machine Vision.* McGraw-Hill, 1995.

[2] H Kauppinen T Seppanen and M Pietikainen. An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):201–207, February 1995.

[3] S R Dubois and F H Glanz. An autoregressive model approach to two-dimensional shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):55–66, January 1986.

[4] M J Paulik M Das and N K Loh. Nonstationary autoregressive modelling of object contours. *IEEE Transactions on Signal Processing*, 40(3):660–675, March 1992.

[5] I Sekita T Kurita and N Otsu. Complex autoregressive model for shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):489–496, April 1992.

[6] K B Eom. Shape recognition using spectral features. *Pattern Recognition Letters*, 19:189–195, 1998.

[7] R L Kashyap R Chellappa. Stochastic models for closed boundary analysis: representation and reconstruction. *IEEE Transactions on Information Theory*, IT-27(5):627–637, September 1981.

[8] J Stoer and R Bulirsch. *Introduction to Numerical Analysis.* New York: Springer-Verlag, 2nd edition, 1980.

[9] W H Press S A Teukolsky W T Vetterling and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[10] M B Priestley. *Spectral Analysis and Time Series.* Academic Press, 1994.

[11] C Chatfield. *The Analysis of Time Series, An Introduction.* Chapman and Hall, 1996.

[12] M-K Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, pages 179–188, February 1962.

[13] R J Prokop and A P Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *Graphical models and image processing*, 54(5):438–460, September 1992.

[14] T Pavlidis. A review of algorithms for shape analysis. *Comput. Graphics and Image Processing*, 7:243–258, April 1978.

[15] D Shen and H H S Ip. Discriminative wavelet shape descriptors for recognition of 2-d patterns. *Pattern Recognition*, 32:151–165, 1999.

[16] B Kartikeyan and A Sarkar. Shape description by time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):977–984, September 1989.

[17] J-P Antoine D Barache R M Cesar Jr L da Fontoura Costa. Shape characterization with the wavelet transform. *Signal Processing*, 62:265–290, 1997.

[18] C M Bishop. *Neural Networks for Pattern Recognition.* Oxford Univedrsity Press, 1998.

[19] P A Devijver J Kittler. *Pattern Recognition: a Statistical Approach.* Prentice-Hall Englewood Cliffs, 1982.

[20] S Haykin. *Nonlinear Methods of Spectral Analysis.* Berlin: Springer-Verlag, 1979.

[21] Y He A Kundu. 2d shape classification using hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, November 1991.

[22] L Gupta K Malakapalli. Robust partial shape classification using invariant breakpoints and dynamic alignment. *Pattern Recognition*, 23(10):1103–1111, 1990.

[23] X Shenshu Z Zhaoying Z Limin C Tianhong. Complex nonlinear exponential autoregressive model for shape recognition using neural networks. *IEEE Instrumentation and Measurement Technology Conference*, pages 289–292, May 1998.

[24] M J Paulik Y D Wang. A multiwavelet model for 2d object analysis and classification.

[25] Q M Tieng W W Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):910–916, August 1997.

[26] R H Glendinning. Robust shape classification. *Signal Processing*, 7(2):121–138, 1999.

[27] R A Herbert R H Glendinning. Shape classification techniques: a comparison. *In preparation.*

[28] G S Sebestyen. *Decision-Making Processes in Pattern Recognition.* New York: Macmillan, 1962.

Appendix A

A Gaussian classifer variant: RCS method

A.1 The Rotated Coordinate System (RCS) Method

The RCS method rotates the axes of the feature space coordinate system as well as scaling them [28]. The projections of the features onto the new axes result in a reduced mean-square intraclass distance. Scaling the feature space has the effect of emphasising the common features and deemphasising the uncommon features. It has been shown [28] that the optimum coordinate system for a particular pattern class, giving the smallest mean-square intraclass distance, are in the directions of the orthogonal eigenvectors of the sample covariance matrix. The eigenvalues are the variances of the data in the directions of the new axes. The distance between a test vector, $\mathbf{p}$, and the class Q training set, $\mathbf{q_k}$, in the rotated and weighted class Q pattern space is given by equation A.1,

$$S(\mathbf{p}, \{\mathbf{q_k}\}) = \left(\prod_{j=1}^{m} \sqrt{\lambda_j} \right)^{2/m} \cdot \left[\left\{ \sum_{i=1}^{m} \frac{[\mathbf{p} - \overline{\mathbf{q}} \cdot \mathbf{c_i}]^2}{\lambda_i} \right\} + m \right] \tag{A.1}$$

where $\overline{\mathbf{q}}$ is the sample mean vector in the original coordinate system, $\mathbf{c_i}$ and λi are the i'th eigenvector and eigenvalue of the sample covariance matrix respectively and m is the number of dimensions of the feature vector.

As the smallest eigenvalue corresponds to the eigenvector which is in the direction of the minimum sample variance, the projection of the sample points onto this eigenvector result in a feature which is the most invariant of a particular class. The feature weights are inversely proportional to the square root of the corresponding variances, so an unlabelled sample will be assigned to the class having the closest mean in the direction of the eigenvector corresponding to the smallest eigenvalue.

Appendix B

Further results

B.1 Benchmark data

The results presented in this appendix are of each technique applied to well separated benchmark data. The tests were necessary to compare with the techniques in journal papers, but do not provide interesting results as the shapes are well separated in feature space and are easily classified.

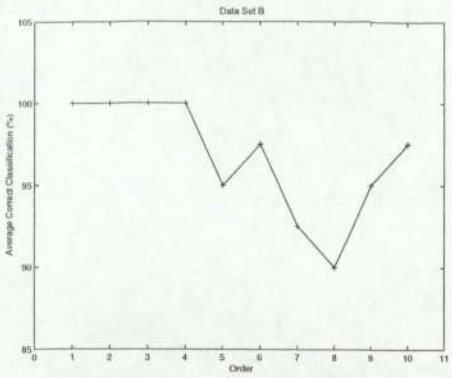

Figure B.1: Radius AR:1 (kNN), Set B

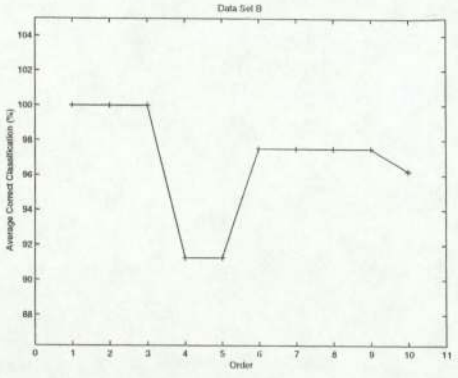

Figure B.2: Radius AR:1 (kNN), Set B

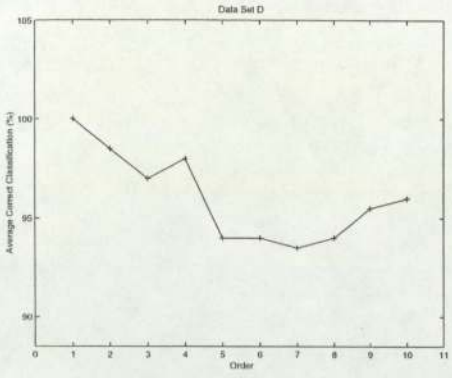

Figure B.3: Radius AR:1(kNN), Set D

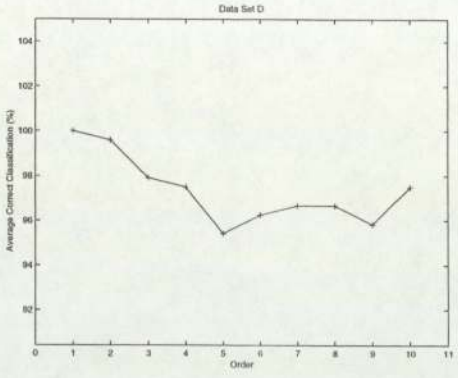

Figure B.4: Radius AR:1 (kNN), Set D

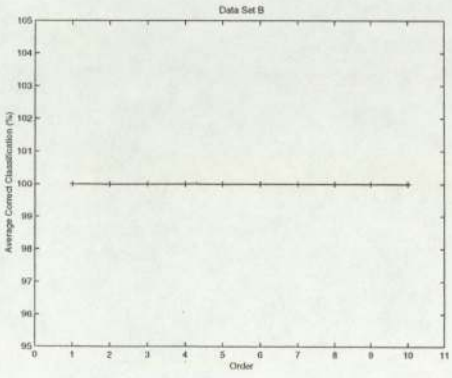

Figure B.5: Radius AR:1 (RCS), Set B

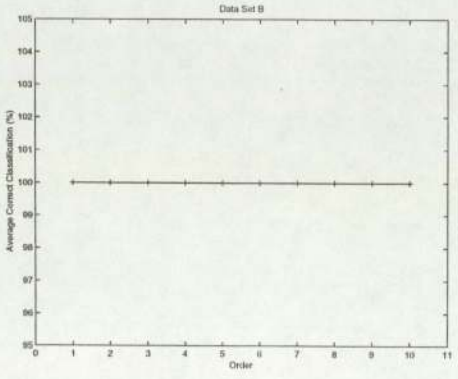

Figure B.6: Radius AR:1 (RCS), Set B

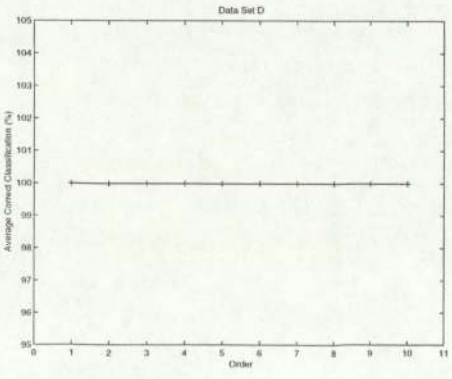

Figure B.7: Radius AR:1(RCS), Set D

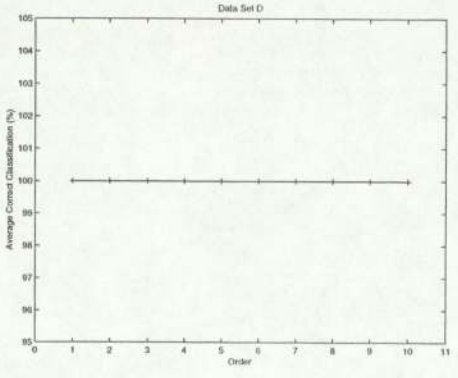

Figure B.8: Radius AR:1 (RCS), Set D