On the use of time series techniques in shape recognition applications

R KRISHNAN and KIRON K RAO

Advanced Data Processing Research Institute, Department of Space, Government of India, Jeedimetla, Hyderabad, 500 855, India

Abstract. In this paper, a method for classifying objects based on the use of autoregressive model parameters which are obtained from a time series representation of shape boundaries in digital images of objects is presented. This technique is insensitive to size and is rotation invariant. The objects chosen are four types of aircraft from a digital photograph. Recognition accuracies of more than 90% were obtained for all the pattern classes. All pattern recognition problems involve two random variables, the pattern vector and the class to which it belongs. The interdependence of the two variables is given by the conditional probability density function. The degree of dependence between the pattern vector and the particular class is measured by the "distance". A simple Bhattacharya distance classifier was used for the purpose.

Keywords. Time series techniques; shape recognition; autoregressive model parameters.

1. Introduction

Recognition of objects is one of the fundamental problems in computer vision, where the goal is to have a computer, capable of performing human-like activities. This has many potential applications in a great variety of topics like industrial parts inspection, quality control tasks, automatic image interpretation of remotely sensed data, and in medicine, and is needed because a computer can perform these routine tasks in a much faster and more efficient way than a human being. In this paper, we present a technique for classification of objects by the use of autoregressive model parameters. These parameters represent the shapes of boundaries, detected in digital images of the objects. Recognition is independent of size, rotation invariant and applicable to objects which are planar in nature or either concave or convex. This paper addresses these problems in two stages, viz., recognition and classification, separately. There are many techniques currently available for shape analysis reported in the literature as a whole, viz. the method of moments (Hu 1962), use of Fourier descriptors (Zahn & Roskies 1972), stochastic models (Kashyap & Chellappa 1981), autoregressive models (Doubois & Glanz 1986) and Volterra-type functions (Kartikeyan & Sarkar 1989).

The Fourier descriptors technique is one of the most widely used techniques for shape analysis. This has been applied successfully to sample objects. The Fourier

coefficients are insensitive to rotation, scaling and variability at the starting point in the tracing of the boundary but do not always regenerate the closed boundary (Pavlidis 1978) which implies that these Fourier descriptors are insufficient to completely represent the object boundary. There are also many more difficulties like broken contours (Gonzalez & Wintz 1987).

The technique of using moments for shape recognition and object recognition/ classification performs well when the objects are simple in nature. The first few moments convey all the information regarding the object boundary. The number of moments to be chosen is purely *ad hoc* and may not perform efficiently under all cases.

Recently, a new method of moments based on Zernike moments has been reported in literature (Teh & Chin 1988; Khotanzad & Hong 1990). These moments are only rotation invariant. The Zernike moment on rotation causes only a change in phase magnitude while the amplitude remains constant. The technique based on stochastic models was first proposed for shape recognition applications by Kashyap & Chellappa (1981). They used their proposed technique specifically as a problem of representation rather than recognition. The stochastic model representation of closed boundaries is invariant to transformations like scaling, translation, choice of starting point and rotation over angles that are multiples of $2\pi/N$, where N is the number of observations. Kashyap & Chellappa (1981) used the stochastic model-based representation for reconstruction using fast Fourier transforms (FFT).

The approach based on autoregressive models (AR) for shape classification was proposed by Dubois & Glanz (1986). They developed a boundary approximation scheme to determine the length N of equiangularly spaced radius vectors projected between the boundary centroid and the boundary which is applicable to both concave and convex objects. The AR model parameters are approximately invariant to shape, size, and translational and rotational positions. Their samples were mainly simulated industrial shapes and several letters from the English alphabet. They have reported a classification accuracy of about 100% in most of the cases.

The method based on Volterra-type functions have been recently reported in literature by Kartikeyan & Sarkar (1989). They use time series modelling techniques to represent closed boundaries of two-dimensional objects. They have developed a non-causal quadratic Volterra model for shape recognition. Their approach contains a concrete guiding principle for the selection of features from objects. They use a D^2 type of statistics which is the Mahalonobis distance measure, in arriving at the conclusions regarding the classification performance. The remaining part of this paper is organized as follows. Section 2 describes the boundary model used, §3 describes the implementations of the algorithm for determining the AR coefficient using the Levinson (1947) and Durbin (1960) algorithm. Section 4 describes the implementation of the problems and the classification approach using the Bhattacharya distance measure. Section 5 presents the results of the experiments performed, while §7 gives the summary and conclusion of this work.

2. Boundary model

The shape-recognition approach described in this paper is based on the autoregressive model. An autoregressive model expresses each sample in an ordered set of data samples as a linear combination of a specified number of previous samples from the data set along with an error term. These concepts are very well known in the disciplines

of speech processing, geophysical data processing and recently in image modelling (Jain 1981). With appropriate boundary sampling, the autoregressive model parameters are invariant to rotation, translation and scaling of the boundary. Hence, these can be used as shape descriptors. These AR model parameters have been used by Kashyap & Chellappa (1981) for boundary representation and reconstruction, while Dubois & Glanz (1986) have used it for shape classification. Dubois & Glanz (1986) have obtained the model parameters using the method of least squares.

2.1 *Boundary representation – The time series*

Before any analysis is performed on the given shape it is necessary to generate the data set from the contour boundary. The boundary is approximated by an ordered sequence of N angularly equispaced radius vectors projected between the object centroid and the boundaries as shown in figure 1 (Dubois & Glanz 1986). It is assumed that the contour boundary is not disconnected. The boundary representation can be improved by increasing the number of data samples i.e. by increasing the sampling interval. The sampling interval equals the perimeter length divided by the series length. Consequently, if the sampling interval becomes so small as to be comparable to the pixel dimensions, the measurements of evenly spaced sampling points on the contour becomes highly prone to error due to the limited spatial resolution. Hence the sampling interval should be greater than two and preferably three units. Hence the series length should preferably be half the length of the perimeter or one third the perimeter length (Kartikeyan & Sarkar 1989).

Two types of objects are normally encountered in these shape recognition problems. When a radius vector is drawn from the boundary centroid of the shape to the contour boundary it may intersect the boundary at one or at more than one point. If the radius vector intersects the contour boundary at only one point for all the radii vectors drawn then this type of shape is termed a wide sense convex object. If the radius vector becomes a multivalued function, then this type of object is called a wide sense concave object. The closure criterion must be satisfied by all objects which introduce a constraint of the form $x(t) = x(t + T)$, where T is the period.
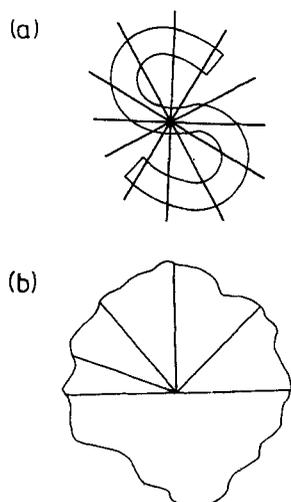
(a)

(b)

Figure 1. The two types of objects normally encountered in any typical shape recognition problem. Wide sense (a) convex, and (b) concave types of objects.

There are many methods available for converting a two-dimensional shape contour into a time series representation. These are discussed in detail by Kashyap & Chellappa (1981) and briefly included here for completeness.

— For a two-dimensional shape boundary, let the contour centroid be at its origin. Let $x(1)y(1)$, $x(2)y(2)$, $x(3)y(3), \ldots, x(n)y(n)$ be the coordinates of the contour with respect to the origin. Then $\{x(i), y(i), i = 1, 2, \ldots, n\}$ is the required time series representation.

— If the shape contour is a wide sense convex type of object then let $r(1), r(2), r(3), \ldots,$ $r(n)$ be the radial distances obtained from the centroid at the origin by the contour boundary at equally spaced angles say (θ). Then this series $\{r(1), r(2), r(3), \ldots, r(n)\}$ is the required time series representation.

3. Autoregressive model

In this section, we describe the theory involved in the formation of the autoregressive model, its order estimation and the estimation of the model parameters.

Let a time series be characterised by $Y = \{y_1, y_2, \ldots, y_n\}$ and let y be the mean of this series. Let there be an independent white noise generator α_i which is normally distributed and having zero mean and unity variance. Then we have a linear stochastic filter model as shown, by Yule (1927) to be of the form
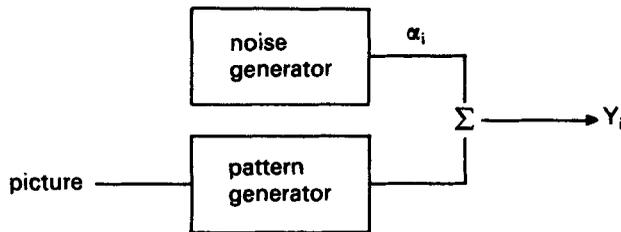
$$y_i = \beta_1 y_{i-1} + \beta_2 y_{i-2} + \beta_3 y_{i-3} + \ldots + \omega_i \qquad (1)$$

$$= \Sigma \, \beta j y_{i-j} + \omega_i. \qquad (2)$$

The description of this process will not be very useful in practice if it contains an infinite number of parameters. Often the stochastic process can be represented as of finite order M given by

$$y_i = \Sigma \, \beta_j y_{i-j} + \omega_i. \qquad (3)$$

This has been obtained based on the assumption that the linear process is reversible and $B(z)$ will converge within a unit circle.

3.1 *Yule–Walker equations*

Consider an autoregressive process of finite order defined in the form given in (2).

$$y_i = \Sigma \, \beta_i y_{i-j} + \omega_i, \qquad (4)$$

$$y_i = \beta_1 y_{i-1} + \beta_2 y_{i-2} + \cdots + \beta_p y_{i-p} + \omega_i. \qquad (5)$$

Multiplying both sides of (4) by y_{i+k} and taking the expectation value and making use of the condition

$$E(y_{i+k} \times \omega_i) = 0, \text{ for } k > 0,$$

which means that the expectation of the product of the signal with noise is zero for k greater than zero. We have

$$E(y_i y_{i+k}) = E(\beta_1 y_{i-1} y_{i+k} + \beta_2 y_{i-2} y_{i+k} + \cdots + \beta_p y_{i-p} y_{i+k}, \tag{6}$$

but we know from the basics that an autocorrection is defined as

$$\rho(k) = (1/N) \Sigma y_i y_{i+k}. \tag{7}$$

This is the nothing but simple autocorrelation. Making use of this relationship in (5) we get

$$\rho(k) = \beta_1 \rho(k-1) + \beta_2 \rho(k-2) + \cdots + \beta_p \rho(k-p), \quad k > 0, \tag{8}$$

Carrying this out for all k ranging from 1 to p we have a family of equations which are commonly referred to as the Yule–Walker equations (Walker 1931; Kay & Marple 1981).

$$
\begin{aligned}
\rho(1) - \beta_1 \rho(0) - \beta_2 \rho(1) \; + \cdots + \; \beta_p \rho(p-1) &= 0 \\
\rho(2) - \beta_1 \rho(1) - \beta_2 \rho(0) \; + \cdots + \; \beta_p \rho(p-2) &= 0 \\
\rho(3) - \beta_1 \rho(2) - \beta_2 \rho(1) \; + \cdots + \; \beta_p \rho(p-3) &= 0 \\
\vdots \qquad\qquad \cdots \qquad\quad \vdots \\
\rho(p+1) - \beta_1 \rho(m) - \beta_2 \rho(m-1) + \cdots + \beta_p \rho(1) &= 0.
\end{aligned}
\tag{9}
$$

3.2 *Calculating the AR coefficients*

Yule–Walker equations have to be solved in order to obtain the estimates for β_i's which in turn can be used as shape features of the object. Estimate the auto-covariance $\rho(k)$ from the autocorrelation estimates $\rho(k)$ as

$$\rho(k) = (1/N) \sum_{i=1}^{N-|k|} (y_{i-k} - m)(y_i - m), \tag{10}$$

where

$$m = \Sigma y_i.$$

The Yule–Walker equations can be rewritten as

$$
\begin{array}{ccccc}
\rho(0) & \rho(1) & \cdots & \rho(m-1) & \beta_{m1} & \rho(1) \\
\rho(1) & \rho(0) & \cdots & \rho(m-2) & \beta_{m2} & \rho(2) \\
\rho(3) & \rho(1) & \cdots & \rho(m-3) & \beta_{m3} & \rho(3) \\
\vdots & & \cdots & & & \vdots \\
\rho(m-1) & \rho(m-2) & \cdots & \rho(0) & \beta_{mm} & \rho(m)
\end{array}
\quad , \tag{12}
$$

where β_{mi} is the ith coefficient of the mth order AR process. Levinson (1947) & Durbin (1960) have proposed a recursive technique to solve for the autoregressive coefficients

which are also termed as the reflection coefficients. They have used a property, $|\beta_{mi}| \leqslant 1$, for m ranging from 1 to m. This is a sufficient condition for the autocorrelation matrix to be semi-positive definite. Note that from (5) we have the estimates,

$$E(y_{t+k} \times \omega_i) = 0, \text{ for } k > 0. \tag{13}$$

But for $k = 0$, we have

$$E(y_t \times \omega_i) = E(\omega_i) = \sigma^2 \omega, \tag{14}$$

$$\rho(0) = \beta_1 \rho(1) + \beta_2 \rho(2) + \cdots + \beta_p \rho(p) + \sigma_{\omega^2} k = 0. \tag{15}$$

Incorporating this condition in the Yule–Walker equations we have

$$
\begin{bmatrix}
\rho(0) & \rho(1) & \cdots & \rho(m) & 1 \\
\rho(1) & \rho(0) & \cdots & \rho(m-1) & -\beta_1 \\
\cdot & & & & \cdot \\
\cdot & & & & \cdot \\
\rho(m) & \rho(m-1) & \cdots & \rho(0) & -\beta_m
\end{bmatrix}
\begin{bmatrix}
\sigma_\omega^2 \\
0 \\
0 \\
\cdot \\
0
\end{bmatrix}
\begin{bmatrix}
B_{m+1} \\
0 \\
0 \\
\cdot \\
0
\end{bmatrix}. \tag{16}
$$

B_{m+1} is the prediction error resulting from the convolution of y_t with $m+1$ point prediction error filter y_i. The solutions of (16) yield

$$\beta_{11} = -\rho(1)/\rho(0), \text{ and} \tag{17}$$

$$\sigma_1^2 = [1 - |\beta_{11}|^2]\rho(0), \tag{18}$$

with the recursion for $k = 2, 3, 4, \ldots, m$ given by

$$\beta_{kk} = -\left\{ \left[\rho_{yy}(k) + \sum_{l=1}^{k-1} \rho_{k-1,l}\rho(k-l) \right] \right\} \bigg/ (\sigma_{k-l}^2) \tag{19}$$

$$\sigma_k^2 = [1 - |\beta_{kk}|^2]\sigma_{k-1}^2.$$

The above equations provide $\{\beta_{k1}, \beta_{k2}, \beta_{k3}, \ldots, \beta_{kk}\}$. Thus the Levinson and Durbin algorithm also provide the autoregressive parameters for all the lower order AR model fits to the data (Kay & Marple 1981).

3.3 *Estimation of the order of the AR process*

The estimation of the order of the autoregressive model is a vital stage in any AR model analysis. There are many techniques available for this estimation. But of all the techniques available, a method called the forward prediction error (FPE) (Akaike 1969) has been reported to perform much better. This is defined as the mean square prediction error. For a given AR process if we define x_t as the estimated prediction of x_i then

$$\text{FPE} = E[(x_i - \hat{x}_i)^2]. \tag{21}$$

From (21) we have the kth order AR coefficient as β_{kk} for a Kth order realization of the AR process.

$$y_i = \sum_{j=1}^{m} \beta_{yy} y_{i-j}, \tag{22}$$

then the

$$\text{FPE} = E[(Y_i - \hat{Y}_i)^2], \tag{23}$$

$$= E\left[\left(y_i - \sum_{m=1}^{m} \beta_{y,m} y_{i-m}\right)^2\right]. \tag{24}$$

Akaike (1969) has shown that this FPE of y_t is composed of the sum of two contributions. The first one corresponds to the minimum residual sum of squares of the mth order AR fit of y_i and the second denotes the statistical deviation of the $b_{i,m}$ from $\beta_{y,m}$, where $\beta_{x,m}$ is the actual and $b_{y,m}$ is the estimated value. The first term decreases as the assumed value of M is increased, whereas the second term increases, hence, that value of m is optimum for which it minimizes the forward prediction error. It has been shown by Akaike (1970) that this optimum value of m in terms of the FPE is given by an equation of the form

$$\text{FPE} = [(N + m + 1)/(N - m - 1)]/\sigma_m^2, \tag{25}$$

where σ_m^2 is the minimum residual sum of squares also called the updated variance. In this present work Akaike's FPE is used to estimate the order of the autoregressive model.

4. Implementation

In this section, a specific computer realization of the autoregressive model is described for the shape recognition problem. The various stages involved in achieving the goal of obtaining shape features include –

(i) digitization of the shape under consideration from the gray scale image;
(ii) performing some preprocessing in order to extract the edge map;
(iii) converting this image into a one-pixel wide contour. This stage is performed if the edge map produced is not single-pixel wide;
(iv) boundary detection and boundary following technique to obtain the time series representation from the shape contour;
(v) estimation of the order and parameters of the AR process;
(vi) classification using a distance classifier. Four types of aircraft were selected from the Janes (1986) book and digitized using a drum scanner at 100×100 microns. Each of these images was of size 256×256 with gray values ranging from 0 to 265.

1	1	0
1	X	1
0	0	0

0	0	0
1	X	0
0	0	0

1	0	1
0	X	0
1	1	1

This was thresholded at 90 (chosen, based on the histogram) in order to produce a binary image. All pixels having gray value below 90 were mapped to 0 while those having gray value greater than 90 were mapped to 1. After the binarization of the original shape it was found that edges were more than the required one-pixel width

contour. The next stage then was to perform a thinning of this boundary. The basic approach is to delete from the object contour border points that have more than one neighbour and whose deletion does not locally disconnect the object. The template used is as follows (Jain 1989). The pixel X having a value 1 cannot be deleted if any one of the above masks is valid. The thinning is performed in an iterative fashion and stopped when no further deletions are performed. At the end of this thinning process a contour of the shape is obtained having a width of only one pixel. This contour is used as input for further processing. As a preprocessing step, a curve-following algorithm was run on the extracted contour in order to make sure that the contour was continous. If any break was observed, then making use of its immediate neighbours, a nearest neighbour interpolation was performed. This contour was used to obtain the time series representation and using the one-pixel wide contour, a contour based centroid was determined. The centroid of an object in 2-D is obtained by projecting the x-coordinates of all the points on to the x-axis and summing up. This sum is then divided by the total number of data points present and the procedure repeated for the y-coordinates also. Mathematically,

$$X_c = (1/N) \sum_{i=1}^{N} x_i \text{ and } Y_c = (1/N) \sum_{i=1}^{N} y_i. \tag{26}$$

Therefore, the coordinates of the centroid are given by $c = (X_c, Y_c)$. After having obtained the centroid of the contour, the origin of the contour is shifted to the centroid. Using this, the time series representation of the contour is obtained. The starting point for obtaining the series is arbitrarily chosen as the intersection of the x-axis with the contour on the left hand side. The time series is generated by moving along the contour until the starting point is reached. At all the locations on the
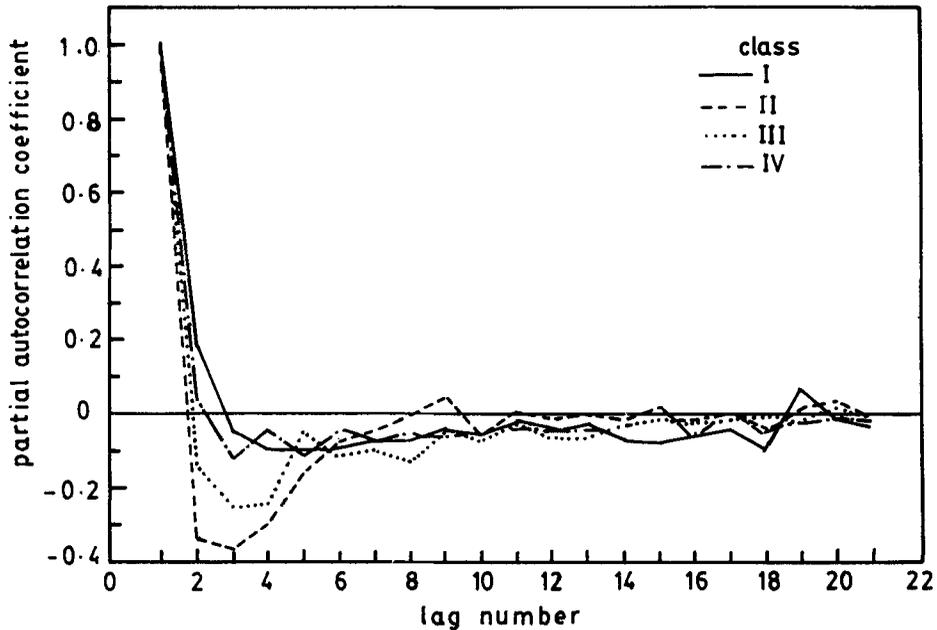

Figure 2. A plot of the partial autocorrelation function against the number of lags for the four different aircraft shapes used, labelled as Class I–IV, respectively.
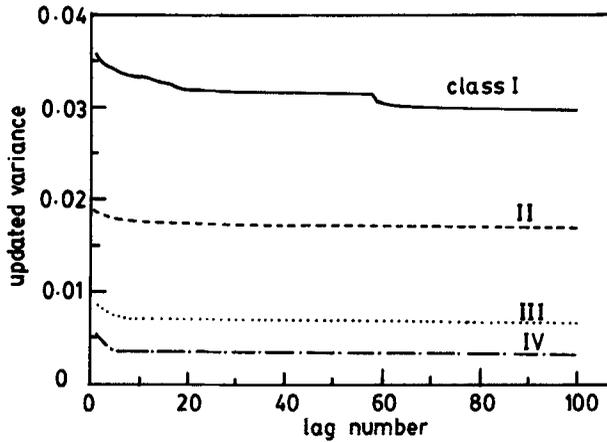
Figure 3. A plot showing the comparison between the updated variance as a function of the number of lags for the four different aircraft shapes. We observe that the variance attains a minimum for that value of lag which corresponds to the order of the model. Beyond this value it remains more or less constant.

contour the x-coordinate and the y-coordinate are recorded resulting in two series. This series is used as input for further processing.

From the series, the mean is computed and subtracted from each element of the series. Using the generated series, the autocorrelation coefficient is computed. This is then fed as input to the Levinson and Durbin algorithm to compute the AR parameters. The partial autocorrelation plot for the four types of aircraft is shown in figure 2. From the figure, we observe that the coefficients attain a minimum and remain constant, on an average, beyond a certain point. Only half the series length was used as the number of lags for computing the autocorrelation coefficient. The updated variance was computed at each stage and this is shown in figure 3. The figure shows that the variance attains a minimum value for the lag which coincides with the order of the autoregressive process. It remains constant, on an average, beyond this point. The forward prediction error is computed at each stage, which in turn is used to compute the order of the autoregressive process. The correct identification of the order of the AR process is a vital stage in the computation. This does not mean that they are more important than the parameters themselves. The order of the model was computed separately for the two series parameters obtained from the two time series in this work. These parameters were clubbed into a single
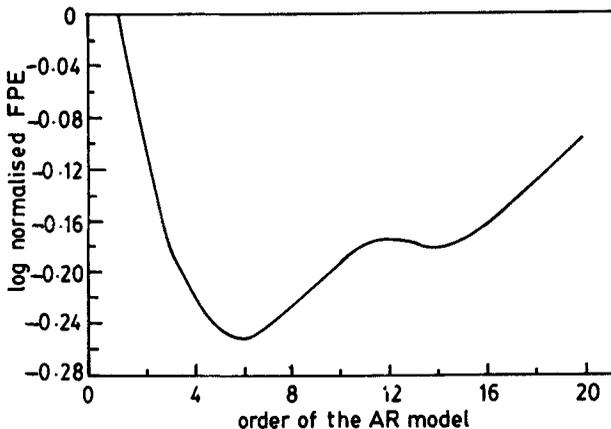

Figure 4. A plot of the log of the normalised forward prediction error as a function of the model order. The model order for which it is minimum gives the order of the autoregressive process.

feature vector for classification purposes. The order of the autoregressive process also becomes a characteristic property of aircraft shape.

5. Results and discussions

Figure 5 shows a plot of features extracted from the four types of aircraft used as inputs in this work. They have been rotated through a fixed angle. The order of the AR process and the coefficients obtained are also tabulated. By just viewing the AR coefficients it may not be always possible to predict whether the coefficients obtained are useful for discriminating between different shapes. If any mathematical technique is applied to the extracted features it is possible to use them in automatic identification of shapes. This has applications in many disciplines like the medical, industrial, aerial and satellite images. The Bayesian decision is known to produce better classification provided the real probability distribution is known. But in most of the practical cases it is difficult to obtain this distribution. Most of the real world situations follow a normal distribution. All pattern recognition problems involve two random variables: the pattern vector and the class. The dependence of the two variables is given by the conditional density function. The degree of dependence between the pattern vector and the particular class is measured by the "distance" between the class conditional density and the mixture density. The probabilistic distance measure can be used for evaluating the probabilistic dependence between the pattern vector and the class. The Bhattacharya distance measure (Kailath 1967) is one such measure, and was used for performing the classification in the present problem.

$$J_B = (1/8)(M_1 - M_2)^T[(\Sigma_1 + \Sigma_2)^{-1}/2](M_1 - M_2)$$
$$+ (1/2)\ln[|(\Sigma_1 + \Sigma_2)/2|/(\Sigma_1)^{1/2}(\Sigma_2)^{1/2}], \tag{27}$$

where M_i and S_i are the mean and covariance matrices of class i respectively. The results obtained indicate the applicability of the autoregressive parameters as features for classification of shape boundaries. In order to test for the boundary approximation case, a portion of the object was clipped off a linear interpolation and the features were extracted. It has been observed that the boundary approximated case can be
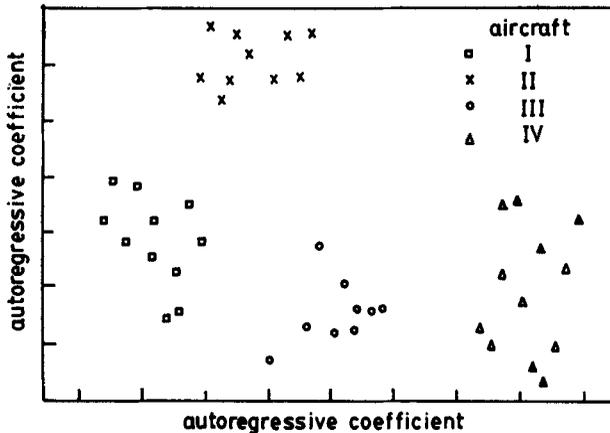

Figure 5. The autoregressive coefficients obtained for the *x*-series and the *y*-series are plotted as the two features. Each aircraft shape was rotated at many angles and the autoregressive coefficients were obtained.

discriminated. In another experiment random Gaussian noise was added to the boundary contour (having $\sigma = 0.5$ and $\sigma = 1.0$). Even in these cases it was possible to correctly classify the aircraft boundaries. In both the above cases, all the data points from the boundary contour were used in order to derive the AR parameters. One of the applications in the shape recognition problem is in cases where the objects of interest are partially occluded. Direct application of these techniques to partially occluded objects is, in fact, a difficult problem because depending on the amount of occlusion the centroid of the boundary will shift, resulting in change in the radius vector lengths. If the occlusion is not very prominent then the time series will retain most of the characteristics of the original shape. All these algorithms were developed and implemented on a VAX 11/780 using Fortran programming language. In order to calculate the AR coefficients of order 1 to p, we require $O(N_p)$ computations, which is less than that for calculating the Fourier descriptors which are of the order $O(N \ln N)$, but the computation is comparable to that required for obtaining moments. The time taken for performing these tasks is not very serious unless the number of objects is few. When shape recognition has to be performed in near-time, the algorithms can be implemented on parallel machines. Initially, a database of the different types of objects has to be prepared. This is something similar to training the system. During the recognition stage, the various shape features are computed and compared with the ones stored in the database. If the object does exist in the database, it is classified accordingly. But, if the object does not exist, the database is updated for future use. Once the number of objects in the database increases, then for every sample object to be recognised, the number of comparisons involved become significantly large.

6. Conclusions

In the present paper, we have reported a technique for shape analysis and classification using the autoregressive modelling approach. A simple classifier, in the form of the Bhattacharya distance measure which determines the between-class scatter and within-class scatter, is sufficient. There may be many more types of classifiers or measures. But it may not be always necessary to use them to increase the computation time. The aircraft shapes from gray level images were first converted into binary images and further processing was performed in order to obtain a single-pixel wide contour of the shape. The objects were also rotated at arbitrary angles and it was observed that the recognition accuracy was not disturbed significantly. The coefficients of the model are invariant to rotation and choice of the starting point of the boundary.

Our results indicate that the autoregressive modelling approach can be used as a regular technique for automatic object recognition in many disciplines like medical images, industrial images, and remotely sensed images, both from aircraft and satellite platforms. But before this technique is applied to any new object the features must be present in a database. In the present implementation, if the object does not fit into any of the classes already present in the database, then the database is updated automatically. Since the autoregressive parameters have got many basic advantages like a small set of features, high recognition rate, small storage requirements and fast recognition time, this approach becomes a novel candidate for use in conjunction with other already proven techniques in near real-time systems.

KKR is grateful to Mr P M Sundaram for going through the preliminary version of the manuscript and suggesting improvements in the basic presentation of the paper. The authors also thank Mr L S Murthy for his help with the paper and the reviewers for suggesting improvements in its overall appearance.

References

Akaike H 1969a Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.* 21: 243–247
Akaike H 1969b Power spectral estimation through autoregressive fitting. *Ann. Inst. Stat. Math.* 21: 407–419
Akaike H 1970 Statistical predictor identification. *Ann. Inst. Stat. Math.* 22: 203–217
Doubois S R, Glanz F H 1986 An autoregressive model approach to two dimensional shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8: 55–66
Durbin J 1960 Fitting the time series models. *Rev. Inst. Int. Stat.* 28: 233–244
Gonzalez R C, Wintz P 1987 *Digital image processing* 2nd edn (Reading, MA: Addison Wesley)
Hu M K 1962 Visual pattern recognition by moment invariants. *IEEE Trans. Inf. Theory* IT-8: 179–187
Jain A K 1981 Advances in mathematical models for image processing. *Proc. IEEE* 69: 502–528
Jain A K 1989 *Fundamentals of digital image processing* (Englewood Cliffs, NJ: Prentice-Hall)
Janes 1986 *All the world aircraft (1985–86)* (New York: Janes)
Kailath T 1967 The divergence and Bhattacharya distance measure in signal selection. *IEEE Trans. Commun. Tech.* 15: 15–60
Kartikeyan B, Sarkar A 1989 Shape descriptors by time series. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-11: 977–984
Kashyap R L, Chellappa R 1981 Stochastic model for closed boundary analysis representation and reconstruction. *IEEE Trans. Inf. Theor.* IT-27: 627–637
Kay S M, Marple S L 1981 Spectrum analysis – A modern perspective. *Proc. IEEE* 69: 1380–1419
Khotanzad A, Hong Y H 1990 Invariant moment recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-12: 489–497
Levinson N 1947 The Wiener (RMS) error criterion in filter design and prediction. *J. Math. Phys.* 25: 261–278
Pavlidis T 1978 A review of algorithms for shape analysis. *Comput. Vision Graph. Image Process.* 7: 243–258
Teh C H, Chin R T 1988 On image analysis by the method of moments. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-10: 496–513
Walker G 1931 On periodicities in series of related terms. *Proc. R. Soc. London* A131: 518–532
Yule G U 1927 On a method of investigating periodicities in distributed series with special reference to Wolfer's sunspot numbers. *Philos. Trans. R. Soc. London* A226: 267–298
Zahn C T, Roskies R Z 1972 Fourier descriptors for plane closed curves. *IEEE Trans. Comput.* C-21: 269–281