AN AUTOREGRESSIVE MODEL APPROACH FOR CLASSIFICATION OF OBJECTS

By

MYSORE N. MANJUNATH

Bachelor of Engineering in Electrical Engineering

Osmania University

Hyderabad, India

1985

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
July, 1988

AN AUTOREGRESSIVE MODEL APPROACH
FOR CLASSIFICATION OF OBJECTS

Thesis Approved:

Keith A. Teague
Thesis Adviser

C. M. Bacon

Richard L. Cummins

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I take this opportunity to thank the people who have helped me all the way through this project. The first person whom I wish to thank is Dr. Keith Teague who heartedly consented to be my adviser and helped in making this academic endeavor profitable from the learning point of view. I also wish to thank Dr. Charles Bacon and Dr. Richard L. Cummins who gladly accepted to be on my committee.

I wish to especially thank Dr. Glenn Kranzier, Professor, Agricultural Engineering Department, who allowed me to use the IRI machine vision system in the Machine Vision Laboratory in their department. And finally, I would like to appreciate the help given to me by Mr. Mark Appleman, Software Specialist, Ag. Engg. Lab., time and again on the IRI machine vision system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Computer vision systems have become an integral part of many automated manufacturing processes. Visual sensors are used in many applications for robot arm positioning and parts placement, while inspection systems using object classification techniques are used in quality control to detect flaws in objects to ensure high quality products in many areas (1). Most of the vision systems used in quality control have not only proved to be faster but also more reliable than human inspectors. Besides their use for quality control, robots are used in the assembly line to assemble parts by making use of their ability to recognize and distinguish a part from a number of other parts and also their mechanical ability to assemble these parts.

Restrictions do apply when it comes to using vision systems in real-time applications. One restriction is that some systems require the part or the object to be recognized be isolated from the others and also be completely visible (2), though there are systems which have overcome this problem (3) with reduced accuracy. The present paper deals with a method of classification of objects using a mathematical model whose parameters represent the shape of the boundary detected in digitized images of the objects. This research is based on a reference method by Susan R. Dubois and Filson H. Glanz (4). Improvements are suggested and investigated in this research. The mathematical model used is an autoregressive (AR) model whose parameters represent the shape of the boundaries of objects invariant to the size and orientation of the object. The recognition techniques used in the research require the object to be isolated and completely visible. An investigation is performed on the ability to accurately classify a number of industrial as well as non-industrial objects.

Overview

Chapter II is an overview of recognition methods. It is divided into two parts. The first part describes several techniques to represent a closed boundary. The second part describes models and methods to extract invariant features of the image of the object from its boundary representation. Chapter III describes the approach used in the present research. It describes the reference method for estimation of the AR model parameters and the modifications on it to investigate improvement in the accuracy of classification. The results of the investigation and the comparision of these results with the prevous work by Dubois and Glanz (4) are presented in Chapter IV. Finally, in Chapter V, a summary of the work done by the author, his conclusions on the methods and future work which could be done furthering this research are described.

CHAPTER II

OVERVIEW OF RECOGNITION METHODS

This chapter is divided into two parts. The first part describes several techniques to represent a closed boundary. The second part describes methods to extract features of the image from its boundary representation. These methods are invariant to scaling, translation, and rotation of the object. These feature extraction techniques could be used on any of the forms of boundary representation described here. Though there are other methods for object recognition, not using boundary representation, this chapter is restricted to only those methods which use boundary representation for classifying objects, this being the point of the present research.

Representation of Closed Boundaries

Most of the vision systems used in the present day are based on recognition techniques which use one of several ways to represent the detected boundary of an object. This section describes some of the different ways to represent a closed boundary. These techniques are restricted to two-dimensional shapes whose boundary does not cross itself. The basic rule in the described methods is that the boundary is represented by a sequence of real numbers in a form much like a time series. The given boundary is approximated to be a polygon of N sides where N is chosen depending on the needed accuracy of the representation. Some of the techniques follow.

Polylines

The method of polylines (5) is one of the simplest ways to represent a boundary. The boundary is seen here as a concatenation of line segments and is represented as a list of points $x_1, x_2, x_3, ..., x_N$. If the first and the last points are the same, then the representation is that of a closed boundary. Polylines can approximate the the boundary to any desired accuracy. The accuracy depends on the number of break points, N. An algorithm (5) is given below for closed boundary approximation which would approximate a boundary given the number of break points.

ALGORITHM for Polylines

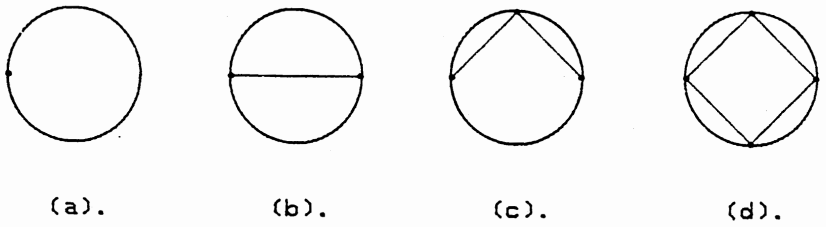
1) Select a starting point.
2) For every point on the boundary, compute its perpendicular distance to the approximating polyline. For the starting point, just compute the distances of all the boundry points. If all the distances are within tolerance, exit.
3) If the distances are above tolerance, pick the point farthest from the approximating polyline and make this the new break point and replace the relevant segment of the polyline with two new line segments.
4) Apply the algorithm recursively to the new segments. Figure (1) shows the different stages of this implementation.

Chain Codes

Chain codes were developed by Freeman (5) and are often called Freeman chain codes. They consist of line segments that must lie on a fixed grid with a fixed set of possible orientations. The starting point is represented by its location, and the other points on the boundary are represented by successive displacements from grid point to grid point along the boundary. A large number of grid points would mean a more accurate representation of the boundary. Figure (2) shows a closed boundary and its corresponding chain code. The grid is usually considered to be four- or eight-connected. The direction can be represented in two or three bits but the starting point (as an example) takes sixteen bits to be represented in a 256 x 256 image. But chain codes may be made position independent by ignoring the starting point. Chain codes can be normalized by choosing a starting point so that the resulting sequence of direction codes forms an integer of minimum magnitude. This could be achieved by choosing a starting point that has the maximum number of '0's that follow the starting point. Periodic correlation provides a measure of the chain code similarity. The derivative of the chain code is useful because it is invariant under boundary rotation. The derivative is just another sequence of numbers indicating the relative direction of the chain code segments. Chain codes are also helpful in the calculation of parameters, such as the area enclosed by a closed boundary, by using the chain code information to determine the slope at the break points. Figure (2a) shows the direction numbers for a chain code (four-connected). Figure (2b) shows a boundary and the corresponding chain code.

ψ -s Curve

The ψ-s curve (5) is similar to the chain code representation. is the angle made between a fixed line and a tangent to the boundary of the shape. It is plotted against s, the arc length traversed. The Ψ-s function is periodic with a discontinuous break from 2π to 0 as the tangent retains the angle of the starting point after traversing the boundary. Horizontal lines correspond to straight lines on the boundary whereas vertical lines correspond to a rapid change in direction

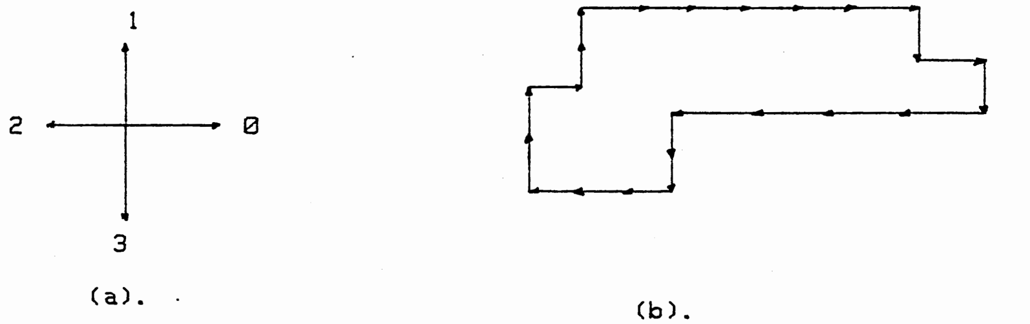

(a).

(b).

(c).

(d).

Figure 1. Stages in the algorithm for polylines.



(a).

(b).

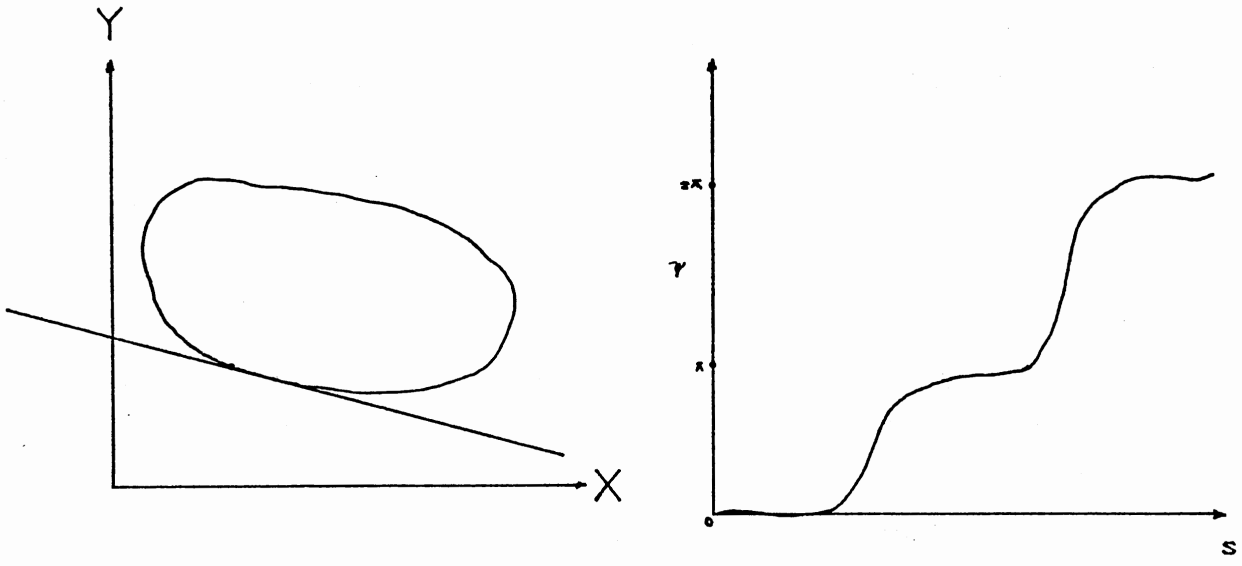
Chain code: 110110000003032222233222

Figure 2. Chain code representation.

along the curve. Figure (3) shows the ψ-s curve for the given boundary. This method is important in the sense that it forms the basis for several measures of shape. There is a disadvantage with this method in that the representation is very sensitive to the noise inherent in the fuzzy boundary. Angle time-series: Angle time-series representation (6) is a one-dimensional boundary representation. A time series describing the boundary is formed by a series of angles $\theta_1, \theta_2, \theta_3, \theta_4$, ..., N which describes a polygon A_1, A_2, ..., A_N, having equal sides. The vertices of the polygon are points on the boundary of the object. The larger the number of sides of the polygon the better is the representation. θ_i is the angle of the sector described by side A with respect to the centroid. The representation would be (θ_i, i=1,..., N). Radius time-series: The radius time-series method (6) is the one used in the present research to represent a closed boundary. A time series is formed by a series of radius vectors r_1, r_2, ..., r_N, which are the distances from the centroid of the closed boundary to N points on the boundary displaced by equal angles. The number of boundary points may be more than N if the shape is wide-sense convex. Figure (4) shows a boundary and the plot of the radius vectors as a function of "time". The representation would be (r_i, i=1, ..., N).

B-Splines

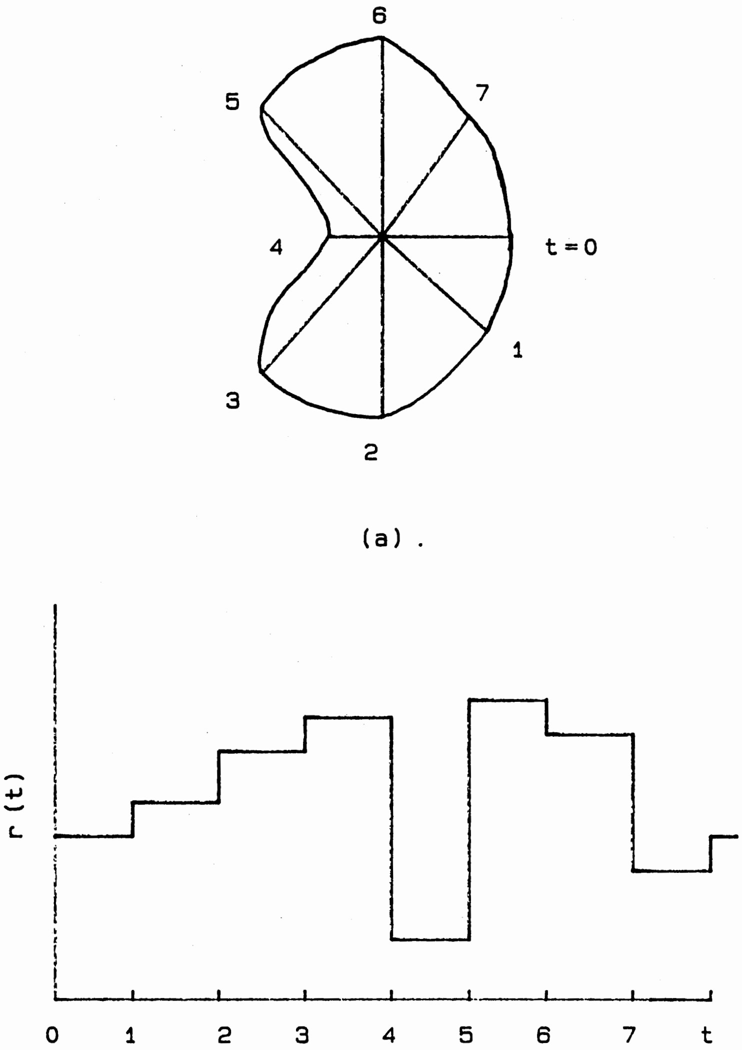
The B-splines technique (5) is an interpolative technique of a piecewise polynomial interpolant. B-splines are piecewise polynomial curves used in approximate representation of a boundary. Cubic polynomials could be used for splines as they are the lowest order in which a curvature can change sign. Spline approximations are accurate and the curve is guaranteed to lie between groups of n+1 consecutive points, where n is the degree of the polynomial. Another advantage is that the interpolation is done locally and the boundary is approximated in a piecewise fashion.



(a).

(b).

Figure 3. ψ-s representation.



(a) .

(b) .

Figure 4. Radius time-series representation.

The above described methods make use of curve fitting or boundary following algorithms to describe the boundary. Some of the other methods which also fall in the above category are the strip tree techniques (5) and conic representation (5) of the boundary. A strip tree is a binary tree representation having the property that it allows efficient computations on its code albeit taking more memory space. A conic representation is a polynomial representation of degree two, having six parameters. Conic representations are terse and serve as good models for physical curves such as edges of industrial objects.

Recognition Methods

The previous section described several ways to represent a closed boundary. In this section the methods to extract invariant properties from these boundary representations are described. The two analytic methods which are described here are Fourier descriptors and the Autoregressive model.

Fourier Descriptor Technique

The Fourier descriptor technique (5, 7) represents the boundary of an object as a periodic function which can be expanded in a Fourier series. The discrete Fourier transform for a series is given by

$$x(s) = \Sigma x_k \, e^{jkw_o s} \qquad w_o = 2\pi/p \tag{2.1}$$

and the coefficients X are given by

$$x_k = \frac{1}{P} \int_o^P x(s) \, e^{-jkw_o s} \, ds \tag{2.2}$$

The discrete Fourier transform performed on a sampled boundary representation such as the polylines, or chain code, or the ψ-s curve could lead to a set of features of the image related to the shape of its boundary. These features, which are given by the Fourier coefficients, could be used for classification purposes. Depending on the number of coefficients included, these descriptors give an accurate characterization of the shape. In most applications these extracted features need to be invariant to orientation and translation of the object. The properties of Fourier transforms make the coefficients invariant to these variations, so this technique provides a way of determining invariant features. The reconstruction of the boundary is possible, but if the number of the coefficients is finite, the resulting reconstruction may not be a closed boundary.

The magnitudes of the Fourier coefficients contain the information about the shape of the object, and are invariant. The phase also contains information about the shape, but is affected by the orientation. The phase could be used as an invariant feature only after performing "phase normalization" which does not affect the magnitudes of the Fourier coefficients. As an example, Figure (5) shows a boundary, its sampled boundary list, and the real and imaginary parts of the discrete Fourier transform.

Autoregressive Model

An autoregressive model (4, 6) is the model which is used in the present research for description of the shape of a closed boundary. An AR model is a parametric equation which expresses each sample of an ordered set of data samples as a linear combination of a specified number of previous samples plus an error term. This model is described by Kashyap and Chellappa (6) who use it for shape storage and reconstruction. This model is given by

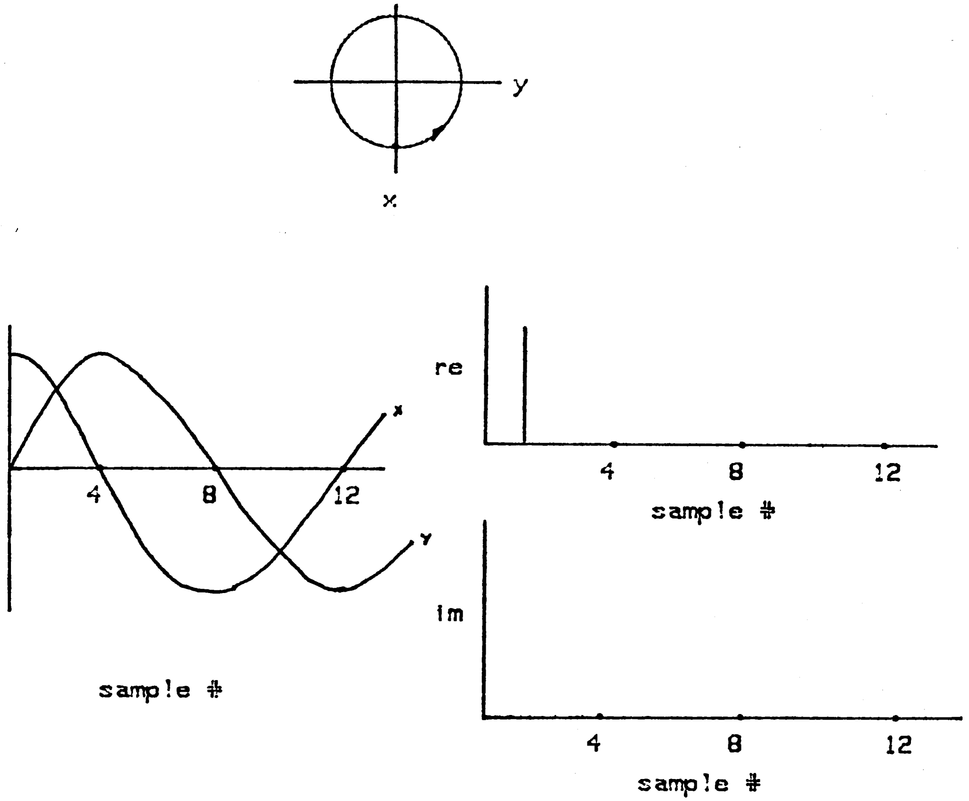


Figure 5. Fourier descriptors representation

$$r_t = \alpha + \sum_{j=1}^{m} \theta_j r_{t-j} + \sqrt{\beta}\, w(t) \qquad t = 1, \ldots, N \qquad (2.3)$$

where

$$w(N+t) = w(t)$$

$$E(w(i)) = 0$$

$$E(w(i)\, w(j)) = \beta\, \delta_{ij} \qquad 1 \le i, j \le N$$

$$\delta_{ij} = 1, \quad \text{if} \quad i = j$$

$$= 0, \text{ otherwise.}$$

The AR model is a simple linear model whose coefficients describe the shape of the boundary of an object. The coefficients are obtained by fitting a linear, autoregressive polynomial to a numerical boundary representation of the object. The boundary representation gives us a way of representing the shape of an object as a vector of radius elements. It is a representative vector for the object, the length of this vector being N. Since only the coefficients of this model are used to describe the shape of an object, the length of the representative vector reduces from N, the number of radius vectors in the boundary representation, to m, the order of the model. m is always much less than N. Any of the sampled boundary representations described earlier could be used as the basis for this model. The radius time series method for boundary representation is used in the present research. The parameters of the model corresponding to a given boundary are invariant to transformations on the boundary, such as translation, rotation and scaling. Hence this model is suitable for classification purposes. Maximum Likelihood (ML) and Least Squares (L2) techniques are usually used to estimate the model parameters.

In the previous work considered by Dubois and Glanz (4), a traditional least square (L2) technique was used to estimate the model parameters. The boundary representation data was assumed to be non-periodic. It is easily seen that the boundary representation is periodic. This is because the representation is that of a closed boundary and so it is periodic with period N, the number of radius vectors in the boundary representation. With the data considered to be periodic, the estimates of the model coefficients are expected to be more accurate and also have lower variance when coefficients are estimated for a number of samples of an object. This is being investigated in the present research. The distribution of the data samples in the boundary representation was not mentioned in the earlier work (4). If the radius data samples are Gaussian distributed, a least square estimate would give the best estimate of the model parameters. However, if the distribution of the radius data samples is Laplacian, a least absolute (L1) estimate of the model parameters is the best estimate. This has also been investigated in the present research.

In brief, the present research investigates the effect on the performance of the recognition system by choosing a least absolute (L1) technique to estimate the model parameters considering the boundary representation data to be periodic and non-periodic over the boundary. The change in performance is also investigated when the data is considered periodic for a least square (L2) estimate.

CHAPTER III

THE APPROACH

This chapter describes in detail the approach taken in the present research for classifying objects based on the parameters of an AR model which represent the shape of the boundary detected from the digitized image of the object. The AR model is described in the first section giving details about the parameters of the model, its invariant properties, and choice of the parameters which are useful for classification. The second section describes the boundary representation technique in detail including the algorithm describing its implementation. In the third section the estimation of the model parameters using the least square and the least absolute error methods is described. The solution to the model is described in detail. The fourth and the final section describes the recognition technique used in classifying the objects. The technique is a pattern recognition technique called the feature weighting method (10). Feature weighting is a method which emphasizes the common features of a set of samples of one class, and deemphasizes the uncommon features.

The Autoregressive Model

As mentioned before, the shape description technique is based on an AR model. An autoregressive model is a mathematical parametric equation which approximates the boundary of a two dimensional image of an object as a linear combination of sequential boundary samples plus an error term (4). It is the invariant properties of this model that make it useful for classification purposes. The data samples taken here form a series of radius vectors, each radius vector giving the distance between the centroid of the object and a boundary point. N such radius vectors represent the boundary of a particular object where each of the radius vectors is equally spaced with an angular difference of 2π/N radians. The model as given by Kashyap and Chellappa (6) is

$$r_t = \alpha + \sum_{j=1}^{m} \theta_j r_{t-j} + \sqrt{\beta}\, w_t \qquad t = 1, \ldots, N \tag{3.1}$$

where

r_t = current radius length

r_{t-j} $(j=1,\ldots,m)$ = previous m radius lengths

θ_j $(j=1,\ldots,m)$ = model coefficients

m = model order

β = residual variance

α = shape descriptor (proportional to shape size)

$\{w_t\}$ = random sequence of independent, zero mean samples

The variance of w_t is unity and so the factor $\sqrt{\beta}$ transforms w_t to a random variable with variance β (4). β is then the residual variance which is estimated as

$$\beta = \frac{1}{N} \sum_{t=1}^{N} (r_t - \alpha - \sum_{j=1}^{m} \theta_j r_{t-j})^2 \tag{3.2}$$

The parameter α (4), which is a descriptor of the shape size is given by

$$\alpha = \bar{r}[1 - \sum_{j=1}^{m} \theta_j], \qquad (3.3)$$

where $\bar{r}$ is the mean radius vector length.

β is defined as the residual variance. It gives a measure of the error in prediction of the t^{th} sample from m previous samples. Hence it can be taken as the noise term. α is proportional to the mean of all the radius values and so is a descriptor of the size of the shape. It gives us a measure of the average signal. $\alpha/\sqrt{\beta}$ can therefore be interpreted as the shape signal-to-noise ratio (SNR). The coefficients of the model (θ_1, θ_2, θ_3, ..., θ_m) are correlated with the shape of the boundary. These coefficients determine the overall shape of the boundary.

The boundary representation which is used here is insensitive to translation, rotation, and changes in the starting point. When there is a variation in the starting point by an integral multiple of $2\pi/N$, then the resulting series of radius vectors is circularly shifted by an amount of this change from the orignal. If the variation in starting point is not an integral multiple of $2\pi/N$, then there would be a variation in the series of radius vectors. If the value of N is large, the variations in the starting point not being an integral multiple of $2\pi/N$ does not matter much because the error in the representation would be small. Hence, the parameters of the AR model, which are derived from this time series, are insensitive to translation, rotation, and variations in the starting point. When the object is scaled, the time series of the radius vectors would be a similar, but scaled version, of the unscaled shape. Hence, the parameters are also insensitive to the size of the object α, which is proportional to the mean radius vector is dependent on the size of the object and so is β, the residual variance. However, the function $\alpha/\sqrt{\beta}$, the SNR, is independent of the shape size because of the similar variations in α and β. Therefore, the vector (   ) which is independent to scaling, translation, rotation and changes in the starting point of the boundary could be used as a feature vector for object classification purposes.

Boundary Representation

The basis for the AR model is one of the boundary representation techniques. Several techniques were discussed in the previous chapter. The technique used here is the radius time-series technique. The boundary is approximated by a series of lengths of N angularly equispaced radius vectors from the centroid to the boundary of the object. Figure (4.a) shows an approximation of the boundary shown, and Figure (4.b) shows the plot of the series of radius vectors versus time. The larger the value of N, the better is the approximation. The radius vector lengths are a function of the angle of projection

$$t * 2\pi/N \tag{3.4}$$

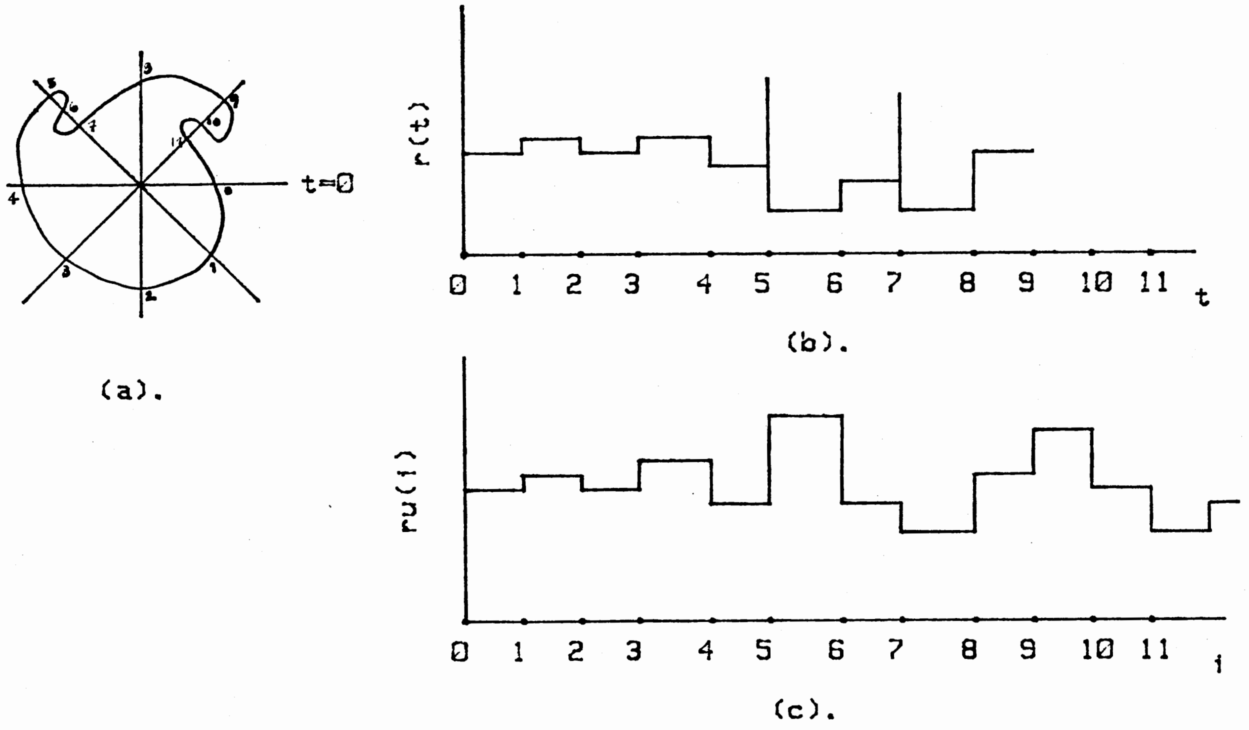
where t = 1, 2, ..., N and r(t) forms a one-dimensional approximation of the boundary. For convenience, the boundary is represented as a time series, t describing the time or the position of the radius vector in increments of 2 /N radians from the starting point.

$$\begin{aligned} r(1) &= r(1 * 2\pi/N) \\ r(2) &= r(2 * 2\pi/N) \\ &\vdots \\ r(N) &= r(N * 2\pi/N) \end{aligned} \tag{3.5}$$

Note that

$$r(t) = r_t \tag{3.6}$$

Since the boundary is closed, the time series is periodic with one rotation from starting point to starting point of the boundary as the period. The restriction here seems to be that the radius function be single valued (i.e., the radius vectors must each intersect the boundary at only one point). If the radius function r(t) is multivalued, the boundary cannot be represented by an AR model. Hence, to avoid this restriction on boundries which are convex or wide-sense convex, an extended series of the radius vectors is created. The extended series of the radius vectors is called the unwrapped or the stretched series of the radius vectors. The function r(t) is still a function of the equispaced angles (Eqn. 3.4). Instead of having N lengths to approximate the boundary, the boundary is now approximated by a larger number of lengths. Figure (6.a) shows a wide-sense convex boundary. Figure (6.b) shows the multivalued radius time-series, and Figure (6.c) shows its unwrapped version. The unwrapped version is a single valued one-dimensional approximation of the boundary which can also be evaluated by the AR model. The new unwrapped version, represented by ru(i), is also obtained by measuring the distance between the centroid and the boundary, but the order of these lengths is not necessarily in the increasing angle of projection. The boundary is searched sequentially until a radius vector crossing is detected and the distance between the centroid and the boundary is measured at this crossing point and stored. The boundary is sequentially searched again for the next crossing vector and again the length is calculated and stored. This is repeated until the starting point is reached. Therefore, the radius vector lengths are stored in the order of detection 'i' by the boundary follower. The period of the new extended series is N_u, which is longer than N of r(t). The observations for the radius vector lengths are still taken at equiangular spaced points, so variations in the starting point would still produce a circularly shifted version of the original boundary series. Thus the new boundary approximation would still have model parameters, $(\bar{\theta}, \alpha/\sqrt{\beta})$, with the invariant properties described earlier. Therefore, the AR model



(a).

(b).

(c).

Figure 6. Wrapped and unwrapped radius vector series.

could be used to represent the boundaries of an unrestricted class of two-dimensional closed boundaries.

Though the representation of the boundary seems simple as described above, some approximations and assumptions are made in its practical implementation. The unwrapped time-series always has a larger number of points than the wrapped boundary when the boundary is convex or wide-sense convex. The estimation of the model parameters would vary if the difference between N and N_u is large.

If the boundary has segments which are straight lines then there is a chance that this line may be coincident with one of the radius vectors. If this were to be the case, the estimation of the model parameters would be biased since all the points on the boundary on this radius vector would have to be considered. A slight rotation of the object might result in all the points on this line being undetected, and the parameters would then be estimated from a lesser number of data points. To avoid this situation, the algorithm is modified so that only one radius vector-boundary intersection between consecutive points is chosen. This is shown in Figure (7). After point P_1 is detected, further points on the line segment are ignored. Though radius vector r_3 crosses the boundary twice, only point P_2 is detected because two consecutive points on the same radius vector are not allowed. The accuracy of the boundary representation could be improved by choosing a large value for the number of radius vectors to be projected from the centroid. One condition which is established when implementing this is that the boundary be traced in the same direction (clockwise or anticlockwise) for all the samples of the object. If the boundary tracing direction is not the same for all the samples, it could result in different representations of the same boundary.

There is a loss of phase information by the method used here for unwrapping a wide-sense convex shape. This is because the change of sign of the angular change is not preserved. Though it is possible that the same extended series of

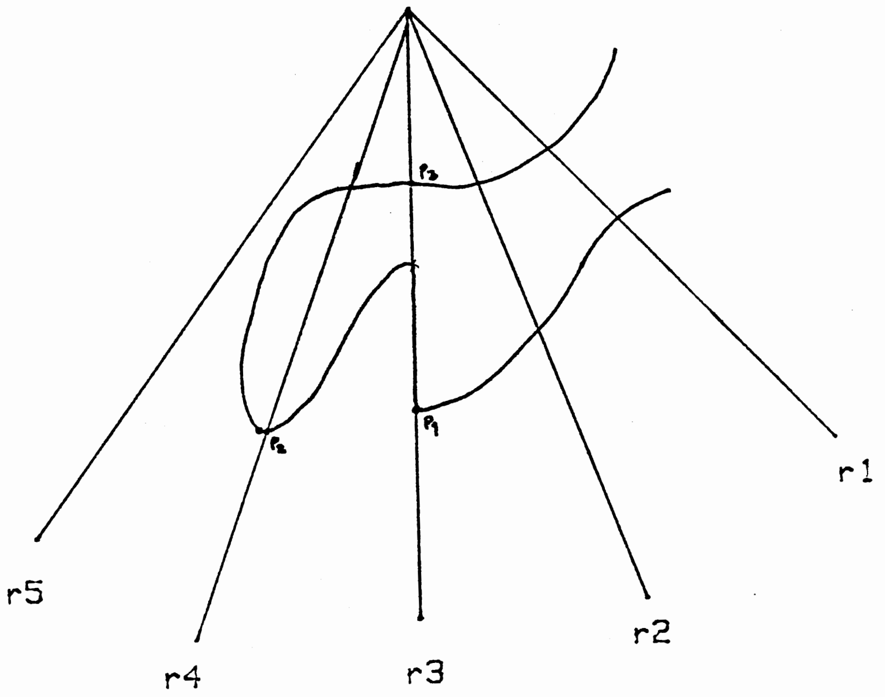


Figure 7. Boundary having a line segment on a radius vector.

radius vectors could come from a different number of objects, it is unlikely that the object would have its centroid at the radius vector origin.

Implementation

In this subsection, the implementation of the algorithm for the boundary representation is discussed. The transformation is from the digitized video image to the extended series of radius vectors.

The first step in this transformation is to obtain the digitized video image. A solid state cctv camera (Hitachi) was used with a macro lens (75 mm) to get an analog image. The continuous image picked up by the monochrome camera has various levels of gray ranging from black to white. The digitized image is displayed on a Panasonic monitor. The image frame has the dimension 240 x 256 pixels. The gray level range is 0-255. The image is processed on an International Robomation and Intelligence, Inc. (IRI) machine vision system.

Once the image is stored in an image frame buffer, the next step is to process the image to make it more useful for the present purpose. The image is segmented at a threshold for the gray level (which could be varied) so as to get a binary image. All pixels below this threshold gray level belong to the object and all the pixels above this gray level form the background. The image is now inverted so that all pixels belonging to the object have the gray level 255 (white) and all the pixels of the background have a gray level 0 (black). Runlength coding is then performed on this binary image to find a starting point on the boundary of the image. Runlength coding is a pixel scan routine which is used for object location and feature extraction of a binary image. The result of this is a data structure containing the coordinate pairs (x,y) of the transitions from black to white and vice versa. Once a starting point on the boundary is located, the next step is to follow the boundary and find all the bounday points of the image. The boundary follower/detector is called a "turtle" (5), and the algorithm for boundary following is called the turtle algorithm. This algorithm can only be used on binary pictures without any gaps in the object boundary. The turtle algotrithm is as follows.

"TURTLE" ALGORITHM (5)

1) Find a pixel on the boundary. Make this the starting pixel.
2) If the current pixel is inside the object (white), then step left relative to the previous step direction. If the current pixel is outside the object (black), then step right relative to the previous step direction. Store the coordinates of the pixel if it is inside the object.
3) Stop, if the boundary point is same as the starting one. Else, go to step 2).

The centroid of the boundary is then calculated. In the present research it was done using the momemt calculating routines present on the IRI system. This could also be done in another simple way. The x and y coordinates of the centroid are the averages of the x and y coordinates respectively of all the pixels which form the boundary of the object.

Once the boundary and the centroid of the object are found, the final step in the transformation, from the video image to the series of radius vectors, is to find the series of radius vectors itself. An algorithm is given below which describes the steps in determining the series ru(i) (4).

ALGORITHM for Finding Radius Vector Series

1) Select the number of radius vectors N to project from the centroid.
2) Find the magnitude of the radius vector slopes for the first quadrant. The boundary is followed sequentially, so the quadrant in which a radius

vector lies need not be determined. The calculations are performed as if all the boundary pixels lie in the first quadrant. Also, the magnitude of the slopes of the radius vectors is the same in all the quadrants.

3) Begin with the starting boundary pixel (X_1, Y_1).

4) If $Y_i - Y_{cn}$ (centroid coordinates: X_{cn}, Y_{cn}) equals zero (i.e., the pixel lies on the same vertical as the centroid), the current pixel lies on the vertical radius vector. The radius length for this point is given by $X_i - X_{cn}$. While the condition that $(Y_i - Y_{cn} = 0)$ is true for the next pixels, stay in this step without evaluating the radius length, as one on this radius vector is already found. Move to the next step when the condition fails.

5) Determine the first quadrant sector in which the current pixel is located. First find the slope of the line joining the current pixel, P_1 in Figure (8), and the centroid

$$\text{Slope} = \left| \frac{X_1 - X_{cn}}{Y_1 - Y_{cn}} \right| \tag{3.7}$$

From the array of slopes in the first quadrant, find the two slopes between which the calculated slope lies. If the pixel is between the slopes of the last slope in the array and the vertical, then the two slopes bettween which the current pixel lies are the last slope of the array and zero. The two slopes, between which the calculated slope for the current pixel lies, define the sector. In Figure (8), 0 and 1 form the sector for the starting pixel P_1.

6) Calculate the x-coordinates, relative to the centroid, of the possible radius vector intersections, XTRY1 and XTRY2, where

$$\mathrm{XTRY1} = |Y_i - Y_{cn}| * \mathrm{SLOPE}(0) \tag{3.8}$$

$$\mathrm{XTRY2} = |Y_i - Y_{cn}| * \mathrm{SLOPE}(1) \tag{3.9}$$

Calculate the differences XDP1 and XDP2 between the y coordinates of the possible radius vector intersections. These are references to the sector.

$$\mathrm{XDP1} = |X_i - X_{cn}| - \mathrm{XTRY1} \tag{3.10}$$

$$\mathrm{XDP2} = |X_i - X_{cn}| - \mathrm{XTRY2} \tag{3.11}$$

7) Take the next boundary pixel. Calculate XTRY1 and XTRY2 for the current pixel (P in Figure (8)). Calculate XD1 and XD2

$$\mathrm{XD1} = |X_i - X_{cn}| - \mathrm{XTRY1} \tag{3.12}$$

$$\mathrm{XD2} = |X_i - X_{cn}| - \mathrm{XTRY2} \tag{3.13}$$

Compare XDP1 with XD1 and XDP2 with XD2. If XD1 and XDP1 are of opposite signs then P is the approximate radius vector-0 boundary intersection. If XD2 and XDP2 are of opposite signs, then P_k is the approximate radius vector-1 boundary intersection (as in Figure (8)). Check to see if the previous radius vector slope for this boundary intersection is not the same as the previous one. If they are not the same then find the radius vector length by the formula

$$\mathrm{Length} = \sqrt{(X_i - X_{cn})^2 + (Y_i - Y_{cn})^2} \tag{3.14}$$

Store the length as a function of i, the order in which it was detected. If the slopes of the present and the previous radius vectors are the same, then do not

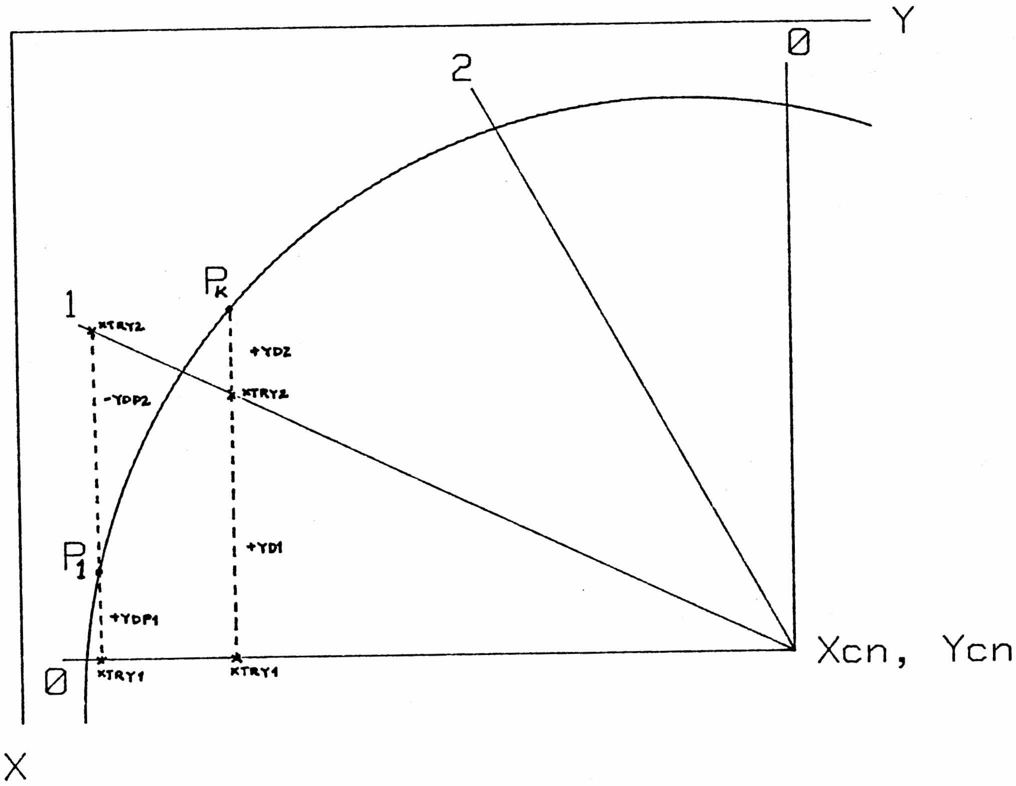


Figure 8. Radius vector-boundary intersection.

find the length. If in the first place, there is no sign change in XD1, XDP1 and XD2, XDP2 then do not process this pixel but proceed to the next pixel and repeat this step. Proceed to the next step if a radius vector is stored or the pixel coordinates are the same as the starting pixel.

8) Exit if all the pixels have been dealt with. Else, go to step (4).

At the end of the implementation of this algorithm, the extended series of the radius vectors is obtained. One of the assumptions made in the implementation of this algorithm is that the value of N is an integral multiple of four. This is just a convenience so that there are equal points on the boundary in all four quadrants. An approximation is that the radius vector-boundary intersection is an approximate value because pixel coordinates are always integers. The programs for the algorithms in this section were written in C on the IRI machine vision system.

Estimation of Model Parameters

The model parameters are estimated by fitting the AR model to the observed time series (r_1, r_2, r_3, ..., r_N). The AR model approximates each data sample by a linear combination of past data samples. By minimizing the sum of the squared differences or the sum of the absolute differences between the actual value and the linearly predicted samples, a set of coefficients for the model can be determined. This set of coefficients is unique if the estimate minimizes the least square error. If the estimates minimize the least absolute error, then the solution is not necessarily unique. In the analysis of speech signals this model is called the Linear Predictive Coding (LPC) model and the determined coefficients are called the predictor coefficients (8). An AR model with coefficients is defined as a system with the output

$$r'_t = \sum_{j=1}^{m} \theta_j \, r_{t-j} \tag{3.15}$$

where r'_t is the predicted value. If r_t is the actual value of the data sample at time t, then the error in prediction is given by

$$e_t = r_t - r'_t = r_t - \sum_{j=1}^{m} \theta_j \, r_{t-j} \tag{3.16}$$

and the average squared prediction error is given by

$$E_t = \sum_n e_t^2(n) \tag{3.17}$$

$$= \sum_n \{r_t(n) - \sum_{j=1}^{m} \theta_j \, r_t(n-j)\}^2 \tag{3.18}$$

This is in the case of least square estimation. If the model parameters were to be estimated using the least absolute value technique, then the average prediction error is given by

$$E_t = \sum_n |e_t(n)| = \sum_n |r_t(n) - \sum_{j=1}^{m} \theta_j \, r_t(n-j)| \tag{3.19}$$

where $r_t(n)$ is a segment of the samples that has been selected in the vicinity of samples at time t.

$$r_t(n) = r_{t+n} \tag{3.20}$$

<u>Least Square Error Technique (8)</u>

The range of the summation, n, is usually a finite interval. To find the values of the model coefficients, the prediction error in the least square error technique has to be minimized. This is done by differentiating the error expression with respect to the model cofficients and equating it to zero.

$$\frac{\partial E_t}{\partial \theta_i} = 0; \; i = 1,\, 2,\, \ldots,\, m \qquad (3.21)$$

By doing so, the following equation is obtained

$$\sum_n r_t(n-i)\; r_t(n) = \sum_{j=1}^{m} \theta_j \; \Sigma \; r_t(n-i)\; r_t(n-j) \quad 1 \le i \le m \qquad (3.22)$$

The js obtained here are the values which minimize E_t.

Defining

$$\theta_t(i,j) = \sum_n r_t(n-i)\; r_t(n-j) \qquad (3.23)$$

Equation (3.24) is obtained.

$$\sum_{j=1}^{m} \theta_j \; \phi_t(i,j) = \phi_t(i,o) \qquad i = 1,\, 2,\, \ldots,\, m \qquad (3.24)$$

Now, m equations in m unknowns are obtained which have to be solved for minimizing the average square or the average absolute error for the segment e (n). Using the Equation (3.22), Equation (3.18) can also be written as

$$E_t = \phi_t(o,o) - \sum_{j=1}^{m} \theta_j \; \phi_t(o,j) \qquad (3.25)$$

As can be seen from the above equation, the total minimum error consists of a fixed component and a component which depends on the coefficients of the model. Now, to solve for the optimum coefficients the quantities $\phi_t(i,j)$ for 1 < = i, j < = m must be obtained. Then solving Equation (3.23) the values of θ_j are obtained.

In defining the limits of summation n, it was defined earlier that it is a finite interval. Two assumptions could be made here while solving for θ_{js}. One is that the radius vector data samples beyond N are zero. This could be expressed as

$$r_t(n) = r(t+n)\, w(n) \qquad (26)$$

where w(n) is a window. A hamming window is chosen if the data samples are chosen non-periodic. The other assumption which could be made is that the radius vector of samples is periodic with period N_u, the length of the expanded series. In fact, in the present case, the boundary representation itself is periodic with period N, because the representation is that of a closed boundary. The window function could be a regular rectangular window.

The advantage of considering data to be periodic is that we have more data which could be used in the estimation of the model parameters. This consideration is investigated in the present research. In the reference method (4), the assumption was that the data samples were non-periodic. The prediction error in the first few samples in the case of non-periodic data is large beacuse the first few samples are predicted from samples which are arbitrarily set to zero. Likewise, the prediction error would be large even at the end because zero is trying to be predicted from samples that are non-zero. A tapering window is thus preferred for non-periodic data samples. If the data samples are non-periodic, then

$$\phi_t(i,j) = \sum_{n=o}^{N-m-1} r_t(n-i)\ r_t(n-j) \tag{3.27}$$

which can also be written as

$$\theta_t(i,j) = R_t(i-k) \tag{3.28}$$

In this case $\theta_t(i,j)$ is the short time autocorrelation function $R_t(i-j)$ given by

$$\phi_t(i,j) = R_t(i-k) \tag{3.29}$$

Since the autocorrelation function is an even function the Equation (3.29) can be expressed as

$$\phi_t(i,j) = R_t(|i-k| \tag{3.30}$$

and the minimum squared prediction error takes the form

$$E_t = R_t(0) - \sum_{j=1}^{m} \theta_j R_t(j) \qquad (3.31)$$

Expressing Equation (3.31) in a matrix form we have

$$\begin{bmatrix} R_t(0) & R_t(1) & \cdots & R_t(m-1) \\ R_t(1) & \cdots & \cdots & R_t(m-2) \\ \vdots & \vdots & & \vdots \\ R_t(m-1) & \cdots & & R_t(0) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix} = \begin{bmatrix} R_t(1) \\ R_t(2) \\ \vdots \\ R_t(m) \end{bmatrix} \qquad (3.32)$$

This pxp matrix of autocorrelation values is a symmetric Toeplitz matrix and a number of algorithms are available for solving it. Solving the matrix equation, the model coefficients are obtained. One method for solving the system of equations given in matrix form by Equation (3.32) is Levinson's method (9) which was used in the present research. This method is used for solving a single channel of normal equations. One equation is present for every coefficient of the model to be solved. The algorithm takes advantage of the symmetric Toeplitz form of the autocorrelation matrix. All the terms along each diagonal are the same in this matrix. Thus, given the entries in the top row of the first column, the whole of the matrix is specified.

The Toeplitz recursion involves determining the model with one parameter, using this to determine the model with two parameters and so on until the model is determined with the desired number of parameters. A benefit of using this scheme is that the mean square error could be computed at each step of the process. A subroutine called 'EUREKA' which solves the least square normal equations for the coefficients of the AR model given in Reference (9) was used to obtain the coefficients of the model. The subroutine uses Levinson's method to determine the model coefficients. The function 'EUREKA' was written in C language on the IRI machine vision system.

Least Absolute Error Technique

The least absolute error criterion is as given in Equation (3.19).

$$E_t = \sum_n |e_t(n)| = \sum_n \left| r_t(n) - \sum_{j=1}^{m} \theta_j \; r_t(n-j) \right| \qquad (3.33)$$

One way the least absolute (L1) solution could be found is by using linear programming (10). In this method the variables of an underdetermined system of equations is allowed to take on only positive values. With this condition, a simple procedure called the simplex method (11) is used to obtain a minimum of a linear objective function. One problem which is faced when using the linear programming approach is that it requires an underdetermined system of equations. Hence, if this approach is used, the formulations of the linear prediction equations will have to be modified.

Another general solution to the linear prediction equations is the residual steepest descent (RSD) algorithm (12). The basic problem here is to minimize E, where the linear prerdiction equations are given in the matrix form as seen below.

$$\begin{bmatrix} r_1 & 0 & \cdots & 0 \\ r_2 & r_1 & \cdots & 0 \\ \vdots & r_2 & r_1 \cdot & 0 \\ & \cdot & \cdots & r_1 \\ 0 & 0 & 0 & 0 \\ r_N & 0 & \cdots & \\ 0 & r_N & \cdots & 0 \\ \vdots & \cdot & \cdots & \\ 0 & 0 & \cdots & r_N \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_m \end{bmatrix} = \begin{bmatrix} r'_1 \\ r'_2 \\ , \\ \cdot \\ \cdot \\ r'_Q \end{bmatrix} \tag{3.34}$$

r_t' is the predicted value of the radius vector r_t at time t. The θ's are the m model coefficients, N is the number of radius vectors in the boundary representation and

$$Q = N + M - 1 \tag{3.35}$$

Symbolically, Equation(3.34) can be written

$$R\bar{\Theta} = R' \tag{3.36}$$

This equation is solved for θ's using the following RSD algorithm.

ALGORITHM (RSD) (13)

1) Calculate the initial value of θ. A least square solution could be used as an initial estimate.

2) Let k=0 (k is the iteration counter). ITERATE OVER k.

3) Calculate $\bar{e}(k) = (R\,\bar{\theta}(k)) - R'$. $\bar{e}(k)$ is the residual vector.

4) Let $\tau_1(k) = \mathrm{SGN}(e_i(k))$ $i=1,..,q$). This means to perform the signum operation on every element of $\bar{e}$. If an element is zero, +1 or -1 is assigned arbitrarily.

5) Minimize E(k) w.r.t. Δ_k, where

$$E(k) = \| \bar{e}(k) - \Delta_k R (R^t R)^{-1} \tau(k) \|$$

This can be solved in a least absolute normed sense or using the iteratively reweighted least squares (IRLS) (12 technique.

6) Let $\theta'(k+1) = \theta(k) - \Delta_k * (R^t R)^{-1} * R^t \tau(k)$.

7) If Δ_k, the tolerance is sufficiently small, then the solution has converged; otherwise go to step 3).

This algorithm does not converge for a true least absolute error solution in all cases. However, it usually converges to a solution within acceptable limits.

Recognition Technique (14)

Pattern recognition consists of two major tasks. One is to characterize the category or the class to which a set of events belongs and the second is to decide the category or the class to which a new described event belongs. In terms of characterization, it involves the construction of regions in the N-dimensional space in which the samples of a class are contained. The second task involves classifying the region or the class to which a new input belongs.

In most of the object recognition schemes, each of the objects to be recognized forms a single class, and a number of samples (f_m, m=1, ..., M) of the features for the object (class) are determined. The similarity of a new event v to a class is measured by the closeness of v to every one of the samples taken in a particular class(f_m). The similarity is represented by S and is taken as the mean square 'distance' between v and the class of events (f_m). f_m is the mth sample in class F. The similarity S(v,(f)) of a new event v and a set (f_m) is mathematically given as

$$S(v, \{f_m\}) = \frac{1}{M} \sum_{m=1}^{M} d^2(v, f_m) \qquad (3.37)$$

The distance measurement is left unspecified. The conditions which d() must satisfy are

$$d(a,b) = d(b,a) \qquad (3.38)$$

$$d(a,c) \le d(a,b) + d(b,c) \qquad (3.39)$$

$$d(a,b) \geq o \qquad (3.40)$$

$$d(a,b) = 0 \quad \text{iff.} \quad a = b \qquad (3.41)$$

Feature Weighting Technique

Consider the distance measurement as given by Equation (3.42)

$$d(a,b) = \sqrt{\sum_{n=1}^{N} W_n^2 \; (a_n - b_n)^2} \; , \qquad (3.42)$$

where N is the dimension of the space. If the vectors are expressed in terms of an orthogonal coordinate system (θ_n), then it might be possible that the events represented by different coordinate directions need not be equally important. This is the concept of feature weighting. W_n represents the weights of the features in each of the N dimensions. It is reasonable, in comparing two points feature by feature, that features with decreasing significance be weighted with decreasing weights W_n. Equation (3.42) expressed in an alternative form would be

$$d(a,b) = \sqrt{\sum_{n=1}^{N} [W_n (a_n - b_n)]^2} \qquad (3.43)$$

This definition of the distance measurement does satisfy the Equations (3.38 - 3.41). The weighting factor W_n is similar to a linear transformation of the signal space which involves only scale factor changes of the coordinates. This is shown in Equation (3.44) where a' and b' are vectors obtained from a and b by a linear transformation denoted by matrix W.

$$a = \sum_{n=1}^{N} a_n \theta_n \qquad b_n = \sum_{n=1}^{N} b_n \theta_n$$

$$(a' - b') = (a-b)W \qquad (3.44)$$

$$= [(a_1-b_1) \; (a_2-b_2) \cdots (a_n-b_n)] \begin{bmatrix} W_{11} \cdots W_{1N} \\ \vdots \qquad \vdots \\ W_{N1} \qquad W_{NN} \end{bmatrix}$$

The euclidian distance between a' and b' is given by

$$d_E(a',b') = \sqrt{\sum_{n=1}^{N} [\sum_{s=1}^{n} W_{ns} (a_s - b_s)]^2} \qquad (3.45)$$

If the linear transformation involves only scale factor changes of the coordinates, only the elements on the main diagonal of the matrix W are non-zero. In this case

$$d_E(a',b') = \sqrt{\sum_{n=1}^{N} W_{nn}^2 (a_n - b_n)^2} \qquad (3.46)$$

This condition is used to minimize the mean square distance between the set of points. The formulation of the minimization is given by

$$D^2 = \frac{1}{M(M-1)} \sum_{p=1}^{m} \sum_{m=1}^{m} \sum_{n=1}^{N} W_{nn}^2 (f_{mn} - f_{pn})^2 = \text{minimum} \qquad (3.47)$$

This can be evaluated under two constraints. These two constraints are

$$\sum_{n=1}^{N} W_{nn} = 1 \quad \text{and} \qquad (3.48)$$

$$\prod_{n=1}^{N} W_{nn} = 1 \qquad (3.49)$$

The constraint in Equation (3.48) is so considered that each weight w is a fractional value of the feature n which it weighs. It denotes a fractional value assigned in the total measure of the distance of the vector. The disadvantage of this constraint is that it does not guarantee that a shrinkage is disallowed in the size of the signal space. This shrinkage would not change the orientation of the points to each other which is what is required. The constraint in Equation (3.49) states that the volume of the space is constant. The minimization when worked out with both the constraints would give similar results. With the constraint given in Equation (3.48) the values of the feature weights which would minimize the distances would be

$$W_{nn} = \frac{\rho}{\sigma_n^2} = \frac{1}{\sigma_n^2 \sum_{p=1}^{N} \frac{1}{\sigma_p^2}} \qquad (3.50)$$

where is the sample variance of the coefficients in the n direction. With the constraint given in Equation (3.49) the resulting feature weights are given by

$$W_{nn} = \left(\prod_{p=1}^{N}\right)^{1/N} \frac{1}{\sigma_N} \qquad (3.51)$$

If the variance of a coordinate of the ensemble is large, then the corresponding wnn is small which means that a small weight is to be given to the measure of the distance with a large variation. If the variance of the magnitude of a coordinate is small, then this would mean that feature is accurately obtained and so its weighting factor must be large. This might mean that if the variance is zero then the corresponding weight for that coordinate would be set to one and the rest to zero. This would surely create problems. To avoid overweighting a coordinate, a higher weighting is given to a feature which is alike in its class and differing from those of the other categories. The feature weighting coefficients can be thought of as descriptors of the category whose feature they weight. The similarity function between a new vector p and a category is then given by

$$S(p,f_m) = \left(\prod_{p=1}^{N} \sigma_n\right)^{2/N} \left[\sum_{n=1}^{N} \left(\frac{p_n - \overline{f_n}}{\sigma_n}\right)^2 + N\right] \tag{3.52}$$

When a new vector is to be classified, the similarity function for each of the classes is evaluated. The class closest in distance to the new vector is then assigned as the class to which the new vector belongs.

CHAPTER IV

TESTS AND RESULTS

This chapter describes the tests that were conducted in order to investigate the effectiveness of the autoregressive model coefficients for classification. The primary reason for using these model coefficients for classification is that they are invariant to scaling, translation, and rotation of the object. The boundary representation obtained from the 2-D image of an object is invariant to the above mentioned variations, and since the boundary representation forms the basis for the AR model, the coefficients obtained for this model too are invariant. The AR model coefficients essentially help in reducing the size of the descriptive vector for a shape. The tests conducted are basically to

i) check the accuracy in classifying a test pattern vector to one of the classes in a set of classes, and

ii) analyze the results by comparing the recognition performance using the four different methods to estimate the model coefficients. The following sections contain the test procedures and an analysis of the obtained results.

Test Procedures

The algorithm used here classifies an object based on the description of the shape of its boundary. Objects having similar shapes as well as objects having different shapes were chosen for the present study. The shapes of seven objects consisting of industrial and non-industrial objects were cut out of black construction paper and used as a pattern set of shapes. Two non-industrial shapes used were the hand-written English alphabetic characters 's' and 'n'. They were chosen to study the performance and evaluate the potential of the recognition scheme being used for character recognition. Figure (9) shows the shapes of the objects which formed the pattern set for the tests. The inner boundary information was not used as, the "turtle" algorithm described in Chapter III follows only the outer boundary of the binary image of an object.

To obtain samples for each object, the coefficients of the AR model were determined for different orientations of the object. Each object was oriented in six different positions (i.e., objects were rotated with a reference point on the boundary at 0^o, 90^o, 180^o, 270^o and two other arbitrary angles with respect to a referance line in the plane). The size of each image of the object was kept constant and was approximately 1.5 times the actual size of the object. The boundaries of the object at each of these orientations were approximated by 64 angularly equispaced radius vectors drawn from the centroid of the object. This number was chosen arbitrarily and it seemed to be a reasonable number for performing computations in the algorithm.

One of the objectives of the study was to compare the the present research with the results obtained in earlier work (4). The data is therefore collected in a manner similar to that described in (4). For each of the six different orientations, the coefficients of the model were determined by running the AR model on the boundary representation of each of the samples. The first step before going into the routine to get the boundary representation was to obtain a binary image from the digitized gray level image of the object. The binary image was obtained by thresholding the initial image at a set gray level (128). The image was thresholded at the same gray level for all the samples of the object while determining the

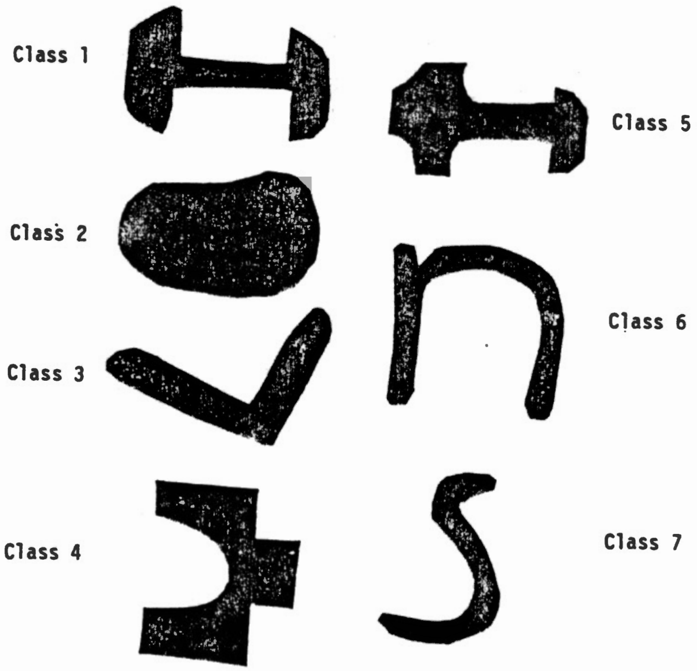


Figure 9. Shapes of objects used in the present research.

model coefficients. This was done to avoid any bias in calculating the model coefficients.

Depending on the orientation and the shape of the object, the number of radius vectors representing the boundary shape for each stayed within +-5 of the number of radius vectors for any of the representations in that class. For one of the objects which had an elliptical shape the number of radius vectors was the same at each of the orientations and was equal to the number of radius vectors projected from the centroid. For objects having two or more boundary - radius vector intersections and whose centroid is enclosed within the boundary of the closed object, the number of radius vectors which represent the boundary is larger than the number of radius vectors projected from the centroid. For objects having centroids outside the closed boundary and with two or more radius vector - boundary intersections, there is a chance that the number of radius vectors representing the boundary is less than the number of radius vectors projected from the centroid. Figures (10a and b) show examples of two such shapes whose boundary is represented by less (a) and more (b) radius vectors than those projected from the centroid.

The coefficients of the AR model are found using the four following approaches.

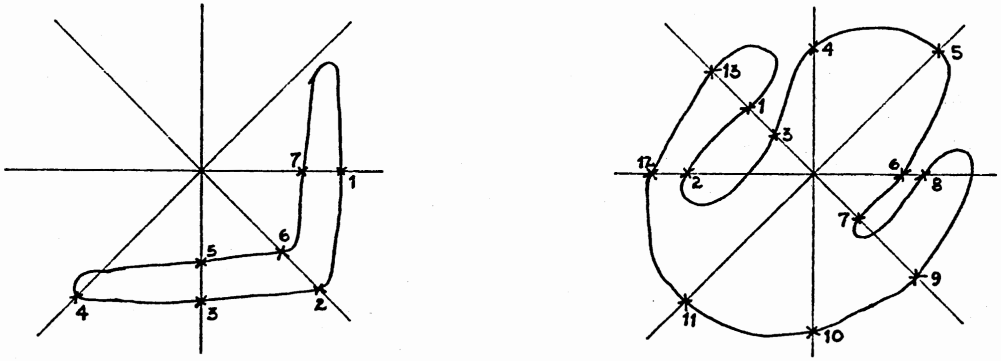
i) Least square error technique considering data to be periodic.

ii) Least square error technique considering the data to be non-periodic. This is the referance method.

iii) Least absolute error technique considering the data to be periodic.

iv) Least absolute error technique considering data to be non-periodic.

The coefficients of the model were obtained for all the objects at each of the six orientations using the above four approaches. The coefficients are expected to be similar for an object at different orientations but using the same



(a).

(b).

radius vectors projected from centroid: 8.

radius vectors for representing boundary (a).: 7.

radius vectors for representing boundary (b).: 13.

Figure 10. Examples of boundaries having lesser (a) and more (b) radius vectors in the representation than those projected from centroid.

approach to estimate the model parameters. The coefficients form clusters, in a weighted m+1 dimensional space, which are at different points in space for the different objects. A test pattern vector is classified by finding the euclidian distance between the test pattern vectors and the cluster of coefficients of the sample pattern vectors of a class. The test vector is labeled to the class to which the euclidian distance is the least. The test pattern vectors for conducting the tests are generated using the same shapes. Two sets of test pattern vectors were generated. One set of the test vectors was obtained by adding noise to the boundary representation. This was done by varying the threshold at which the initial image was transformed to a binary image. The threshold was selected so that a rough boundary is seen in the binary image. No effort was made to characterize this additive noise. The other set of test vectors was generated in the normal way by thresholding the image at the same level as the sample pattern vectors but at different orientations. Two samples of test vectors were generated for the first set of test vectors per class and six samples of test vectors were generated for the second set of sample vectors for each class.

The sample pattern vectors and the test pattern vectors were generated for all the objects using the four different approaches. The performance and the analysis of the results are discussed below.

Results

Table I shows the performance of the recognition system for the model coefficients determined by the least absolute error technique. Table II shows the performance of the recognition system for the coefficients determined by the least square error technique. The results, as seen in Tables I and II, do indicate that the AR model parameters are useful descriptors of the shape of the objects, and the boundary representation which was considered in the present study does

TABLE I

RECOGNITION PERFORMANCE (LEAST ABSOLUTE ERROR TECHNIQUE)

ORDER	DATA	CLASS	CLASSIFIED AS *			
			norm. data 6 sam./cls.	% class.	noisy data 2 sam./cls.	% class.
3	periodic	1		38/42 =90%		12/14 =86%
		2				
		3	4, 4, 4		4	
		4				
		5	1		7	
		6				
		7				
3	non-periodic	1		36/42 =81%		12/14 =86%
		2				
		3	5, 4, 4		4, 4	
		4	7			
		5	2, 1			
		6	3, 2			
		7				
4	periodic	1		37/42 =88%		13/14 =93%
		2				
		3	4, 4		4	
		4				
		5	1, 1			
		6	2			
		7				
4	non-periodic	1		36/42 =85%		11/14 =79%
		2				
		3	4, 4, 4		4, 4	
		4				
		5	1, 1			
		6	2		2	
		7				

* misclassifications are only given.

TABLE II

RECOGNITION PERFORMANCE (LEAST SQUARE ERROR TECHNIQUE)

ORDER	DATA	CLASS	CLASSIFIED AS *			
			norm. data 6 sam./cls.	% class.	noisy data 2 sam./cls.	% class.
3	periodic	1		41/42 =98%		13/14 =93%
		2				
		3			4	
		4				
		5	1			
		6				
		7				
3	non-periodic	1		38/42 =91%		12/14 =86%
		2				
		3	4, 7		4, 4	
		4				
		5	1, 1			
		6				
		7				
4	periodic	1		41/42 =98%		13/14 =93%
		2				
		3			4	
		4				
		5	1			
		6				
		7				
4	non-periodic	1		36/42 =86%		12/14 =86%
		2				
		3	4		4, 1	
		4	3, 3			
		5	1, 1, 1			
		6				
		7				

* misclassifications are only given.

indeed retain enough information for the estimation of these model parameters. The results do depend on the order of the model but not to a very large extent. The performance is expected to improve as the model order is increased in the range 3 to 6. For model orders lower than 3 the error in prediction, of the data in the boundary representation, would would be large because of lesser coefficients to predict from. Thus, the coefficients will not be an accurate representation of the object, so the performance of the recognition system would be bad. For model orders higher than 6 some of the last model coefficients take on very low, noisy, values. The predicted values of the radius vector would thus contain the noise generated by the low values of the model coefficients. Hence, the performance of the recognition system would likely worsen. In the present research only third and fourth order AR models were considered. Since the representative vector for an object contained the coefficients of the AR model and also the extra signal-to-noise ratio term, the order of the pattern vectors was one higher than the AR model orders used. The correct classification percentages were in the range 80% to more than 98%. Table III shows the variance of a coefficient for each of the coefficients of the sample pattern vectors, for all the classes, for the two orders of the model considered, and for periodic and non-periodic data using the least absolute error technique. Table IV shows the same results but considering the least square error technique.

Observing Tables I and II, the following conclusions were reached.

Least Square Vs. Least Absolute Technique

On the whole, using the least square error technique to estimate the model parameters yields a better recognition abilitiy compared to using the least absolute error technique. This conclusion seems to be independent of the order of the model and also the consideration of data to be periodic or non-periodic while

TABLE III

AVERAGE VARIANCE OF COEFFICIENTS (USING LEAST ABSOLUTE ERROR TECHNIQUE)

ORDER	DATA	CLASS	AVERAGE VARIANCE
3	periodic	1	0.007266
		2	0.033540
		3	0.012803
		4	0.003828
		5	0.016625
		6	0.006567
		7	0.004543
3	non-periodic	1	0.009934
		2	0.039232
		3	0.029229
		4	0.005691
		5	0.021926
		6	0.008389
		7	0.009009
4	periodic	1	0.005371
		2	0.046345
		3	0.009486
		4	0.004406
		5	0.021308
		6	0.005144
		7	0.005563
4	non-periodic	1	0.006647
		2	0.056000
		3	0.022077
		4	0.008964
		5	0.027277
		6	0.007387
		7	0.006941

TABLE IV

AVERAGE VARIANCE OF COEFFICIENTS (USING LEAST SQUARE ERROR TECHNIQUE)

ORDER	DATA	CLASS	AVERAGE VARIANCE
3	periodic	1	0.001461
		2	0.017038
		3	0.005277
		4	0.003863
		5	0.016558
		6	0.005321
		7	0.010353
3	non-periodic	1	0.002637
		2	0.020380
		3	0.024858
		4	0.010113
		5	0.014963
		6	0.011629
		7	0.011557
4	periodic	1	0.003456
		2	0.031139
		3	0.004136
		4	0.004077
		5	0.025615
		6	0.004558
		7	0.014382
4	non-periodic	1	0.004346
		2	0.067145
		3	0.019588
		4	0.017221
		5	0.028607
		6	0.002976
		7	0.023866

estimating the model coefficients. The average difference in percentage recognition is about 8%. But, looking at the tables it is seen that object 3 was misclassified as object 4 about 50% of the times while using the least absolute error technique. Neglecting the classification of this object, it is seen that the recognition performance using the least absolute technique is comparable with the performance using least square error technique, though the least square technique still performs better than the least absolute technique. The average difference in percentage recognition is about 5%. The results here would seem to indicate that the probability distribution function of the data is probably Gaussian. This is because the best solution for Gaussian distributed data is obtained from the least square estimate. This is just an hypothesis and no further tests were conducted to verify or oppose it. Looking at the classification of the noisy test vectors, the performance using both methods is comparable except for the misclassification of object 3 as 4. Though only two test vectors per class for the "noisy" case were classified, it can be seen that the order of the model did not seem to matter. The performance was above 85% for both order models.

Periodic Vs. Non-Periodic Assumption

The second analysis which could be made is a comparision of the recognition performance by estimating the model parameters, considering data to be periodic and non-periodic, independent of the order of the model and the coefficient estimating technique. It is seen that estimating coefficients considering data to be periodic yields better recognition ability compared to the non-periodic case. The average difference in percentage recognition is about 8%. This is true for the classification of the "noisy" test vectors as well. The average difference in percentage recognition is about 7%. The results seen here are as expected. This is because periodic data will have more information about the shape of the object from which the model parameters are estimated.

Third Order Vs. Fourth Order

The third and final analysis which could be made is a comparision of the recognition performance by the order of the model independent of periodicity and the estimating technique. It should be remembered that the order of the pattern vectors is one higher than the order of the AR model. There is not much difference in the performance. The average difference in percentage recognition is about 3%. Hence, in a practical implementation of the recognition system, a third order AR model should suffice. All the results described above could be verified by examining Tables III and IV. These tables show the variance of the coefficients for all classes, for the different approaches to estimate the model parameters. Some of the observations are discussed below.

1) The average variance of the coefficients is less for the least square error estimates compared to the least absolute error estimates. This holds true for 19 out of the 28 possible comparisions (7 classes, 2 model orders, and periodic and non-periodic data). A lesser variance in the estimate would mean the estimates are better clustered and probably more accurate. The lesser variance could also result from the numerical procedure used to calculate the least absolute error model coefficients. It is iterative, non-exact, and the solution is also not necessarily unique. This explains the results in the first comparision above.

2) The average variance of the coefficients is lesser for the estimates considering periodic data compared to non-periodic data. This holds true for 26 of the 28 possible comparisions (7 classes, 2 model orders, and 2 techniques to estimate the coefficients). This also explains the

better performance when using periodic data. The average variance cannot be compared for different model orders because of the unequal number of coefficients from which the variance is determined.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

The paper compares the performance of a recognition scheme, using AR model parameters as shape descriptors, described in the present research with a reference method in (4). The model differs from the reference method in the technique of estimating the model parameters. It is shown in the present research that estimating the model parameters considering the data to be periodic would result in better recognition abilities of the recognition system. It is also shown that, neglecting the misclassification of object 3 as 4, the performance of the system using least absolute error techique to estimate model parameters is comparable with the system using least square error technique. In all the methods, the recognition accuracy was in the range 80% to more than 98%. This can be categorized as good performance considering the following advantages offered by this system.

i) The number of parameters required to describe and recognize an object is very small. It is just a function of the order of the AR model.

ii) The storage space required for describing an object is small. This is the memory space required to store the model coefficients.

iii) The time needed to classify a test object is small.

iv) Efficient methods exist to calculate the model parameters.

v) Reconstruction of the shape of the object is possible (6).

Future work which could be done to improve the present research includes

i) Improving the boundary representation technique. Inner boundary information (if any) is not represented in the present research.

ii) The number of radius vectors projected from the centroid is presently fixed at 64. A rule could be formed to find the optimum number of radius vectors for the best boundary representation.

iii) Techniques could be considered which would help in clustering the coefficients of the model more closely, resulting in a better performing recognition system.

iv) Further work could also be done in studying the effects of adding different types of "noise" to the boundary representation, and from it determining an estimation technique for the model coefficients which would produce even better recognition performance.

v) Investigate the usefulness of the AR model coefficients in the reconstruction of the shape of the boundary of an object.

BIBLIOGRAPHY

1. Jeffrey A. Jalkio, Richard C. Kim, and Steven K. Case, "Three Dimensional Inspection Using Multistripe Structured Light", Optical Engineering, November, 1985, pp. 966-974.

2. M. Yachida, and S. Tsuji, "A Versatile Machine Vision System for Complex Industrial Parts", IEEE Trans. Comput., Vol. C-26, 1977, pp. 882-894.

3. W. A. Perkins, "A Model-Based Vision System for Industrial Parts", IEEE Trans. Comput., Vol. C-27, 1978, pp. 126-143.

4. Susan R. Dubois, and Filson H. Glanz, "An Autoregressive Model Approach to Two-Dimensional Shape Classification", IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAM1-8, No. 1, January, 1986, pp. 55-66.

5. David H. Ballard, and Christopher M. Brown, Computer Vision, New Jersey: Prentice-Hall, 1982.

6. R. L. Kashyap, and R. Chellappa, "Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction", IEEE Trans. Inform Theory, Vol. IT-27, 1981, pp. 627-637.

7. Pattern Identification Package manual, International Robomotion Intelligence, Inc., Carlsbad, California, April, 1986.

8. L. R. Rabiner, and D. W. Schafer, Digital Processing of Speech Signals, New Jersey: Prentice-Hall, 1978.

9. Manual T. Silvia, and Enders A. Robinson, Deconvolution of Geophysical Time Series in Exploration of Oil and Natural Gas, New Jersey: Elsevier Scientific Publishing Co., 1979.

10. Trung T. Pham, and Rui J. P. deFigueiredo, "Maximum Likelihood Estimation of a Class of Non-Gaussian Densities With Applications to Deconvolution", Proc. ICASSP 1987, pp. 49-52.

11. Paul R. Thie, An Introduction to Linear Programming and Game Theory, Wiley and Sons: New York, 1979.

12. Yariagadda Rao, J. Bee Bednar, and Terry L. Watt, "Fast Algorithms for L_p Deconvolution", IEEE Trans. ASSP-33, Feb., 1985, pp. 174-182.

13. James L. Lansford, "L_p Models in Speech Coding and Markov Chains in Speech Recognition", Ph.D. Thesis, (Oklahoma State Univ.), May, 1988.

14. G. S. Sebestyen, Decision Making Process in Pattern Recognition, New York: Wiley, 1973.

VITA

Manjunath Mysore N.

Candidate for the Degree of

Master of Science

Thesis: AN AUTOREGRESSIVE MODEL APPROACH FOR CLASSIFICATION OF OBJECTS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Mysore, Karnataka, India, April 16, 1963, the son of M. S. Narasimha Murthy and M. S. Nagarathna Murthy.

Education: Graduated from Kendriya Vidyalaya Golconda, Hyderabad, Andhra Pradesh, India, in May, 1980; received the Bachelor of Engineering degree in Electrical Engineering from Osmania University, Hyderabad, India, in June, 1985; completed requirements for the Master of Science degree at Oklahoma State University in July, 1988.

Work Experience: Research Assistant, Department of Agricultural Engineering, February, 1987, to February, 1988.

Awards/Affiliations: Member, Eta Kappa Nu.