

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Journal Articles

Computer Science and Engineering, Department
of

6-24-1993

Natural shape detection based on principal component analysis

Ashok Samal

University of Nebraska-Lincoln, asamal1@unl.edu

Prasana A. Iyengar

University of Nebraska-Lincoln

Follow this and additional works at: <https://digitalcommons.unl.edu/csearticles>

Samal, Ashok and Iyengar, Prasana A., "Natural shape detection based on principal component analysis" (1993). *CSE Journal Articles*. 222.

<https://digitalcommons.unl.edu/csearticles/222>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Natural shape detection based on principal component analysis

Ashok Samal

Prasana A. Iyengar

University of Nebraska-Lincoln

Department of Computer Science and Engineering

115 Ferguson Hall

Lincoln, Nebraska 68588-0001

e-mail: samal@cse.unl.edu

Abstract. *The classical Hough transform, the generalized Hough transforms, and their extensions are quite robust for detection of a large class of objects that can be categorized as industrial parts. These objects are rigid and have fixed shapes, i.e., different instances of the same object are more or less identical. These techniques, and indeed most current techniques, however, do not adequately handle shapes that are more flexible. These shapes are widely found in nature and are characterized by the fact that different instances of the same shape are similar, but not identical, e.g., leaves and flowers. We present a new technique to recognize natural shapes, based on principal component analysis. A set of basis shapes are obtained using principal component analysis. A Hough-like technique is used to detect the basis shapes. The results are then combined to locate the shape in the image. Experimental results show that the approach is robust, accurate, and fast.*

1 Introduction

The Hough transform (HT) was proposed to fit straight lines to a set of planer points.¹ A direct application is to find straight lines in images,² a common operation in computer vision. The HT uses an elegant technique to transform the problem of finding collinear points to that of finding the intersection of concurrent lines. The latter is computationally more efficient and, hence, the Hough transform has found widespread use in computer vision.

The technique has been extended to detect curves³ and arbitrary 2-D shapes.⁴ The generalized Hough transform (GHT)⁴ is robust for rigid shapes with fixed scale and orientation. Brute-force techniques can be used to handle shapes at different scales and orientations. Several approaches have been proposed to accomplish this more efficiently.^{5–8} Along with these extensions the HT is quite robust for the recognition of a large class of objects that can be categorized as industrial parts. These objects are rigid and have fixed shapes, i.e., different instances of the same object are more or less identical. The GHT with all these extensions, and indeed most current techniques, still does not adequately handle shapes that are more flexible. These shapes are widely found in nature and are characterized by the fact that different instances of the same shape are similar, but not identical, e.g., leaves and flowers.

We present a technique to recognize natural shapes in images. The approach is based on multiple template matching using the GHT. The template set is generated from a general set of shapes by the principal component analysis on a general population. The advantage of this approach is that the basis set of shapes used for detection is small and embodies most of the variation in shapes. Experimental results indicate that shapes can be detected accurately with this approach. The detection time is reasonable and has room for improvement. A similar approach has also been used to detect human face silhouettes in images.^{9,10} This work extends the approach to detect general shapes.

The remainder of the paper is organized as follows: A brief discussion of the Hough-based approaches along with

Paper 93-003 received Jan. 15, 1993; revised manuscript received June 11, 1993; accepted for publication June 24, 1993.
1017-9909/93/\$6.00. © 1993 SPIE and IS&T.

their drawbacks is given in Sec. 2. In Sec. 3 the motivation and the approach used in this work for shape recognition is described. The steps used in the principal component analysis and the recognition process are presented in this section. An estimation of accuracy of the approach is also given. Section 4 describes the implementation, the testing process, and the results obtained. A detailed discussion of the results is also presented. Finally, a summary and several avenues for future research are given in Sec. 5.

2 Previous Work

The HT was designed to locate analytic curves in images.¹ The original formulation was given for straight lines, but it can be generalized for any analytic curve. The central idea in the HT is to recast the problem from the image space to the parameter space. The former is defined by the $X-Y$ coordinate system of the image and the latter is the space spanned by the parameters. The HT transforms the problem of finding collinear points in the image space to finding the intersection of concurrent lines in the parameter space. The latter is computationally more efficient and, hence, the Hough transform has found widespread use in computer vision.

The main drawback of the classical HT is its time and space requirements, particularly for complex shapes. The algorithm is exponential in the number of parameters in the curve. This poses a serious implementational problem for curves with more than a very small number of parameters. Due to this difficulty, the classical HT is seldom used when the number of parameters is more than two or three. The generalized HT attempts to overcome these difficulties.

Merlin and Farber³ and Ballard⁴ extended the HT to make it more general and computationally more efficient. Because many shapes cannot be described by simple analytic curves, the HT cannot be directly used. The GHT allows the shape to be in any form. To make the approach computationally efficient, which is important in many computer vision applications, the GHT makes use of the orientation information available in both the curve and the edge points in images. The advantage of the GHT for general shapes is in its space requirement. The accumulator array has the same size as the image and, hence, is quite manageable. The time and space complexities are superior to the classical Hough technique, in which both are exponential.

Principal component analysis has been used to recognize human faces.^{11,12} Similar approaches have also been used to detect human face silhouettes in images.^{9,10}

2.1 Problems with the Hough-Based Approaches

Traditional Hough techniques have many good properties. First, they are computationally efficient and very effective in recognizing a wide variety of rigid shapes, which can be classified as industrial objects. These objects must conform to strict standards and, hence, different instances of the same object are more or less identical. Second, the techniques are effective even when parts of the object are missing or are deformed. If the objects are uniformly scaled or rotated, Hough-based techniques are also successful. Figure 1 shows an industrial object and various instances of it that Hough-based techniques can adequately handle.

However, if an object's shape has many variations, the Hough-based techniques have major problems. To illustrate

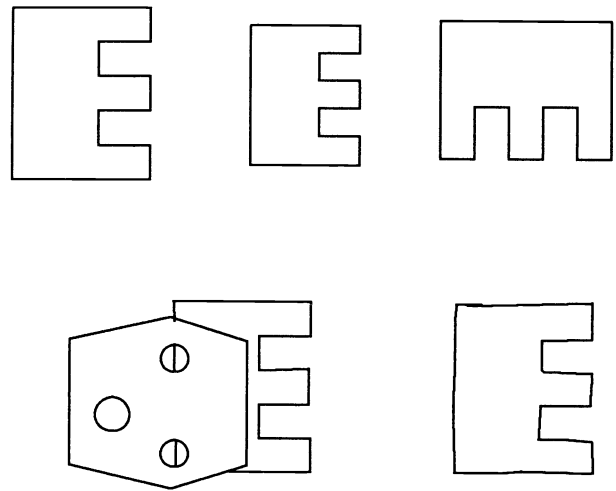


Fig. 1 Different instances of an industrial object: the model, scaled, rotated, partially occluded, and partially deformed.

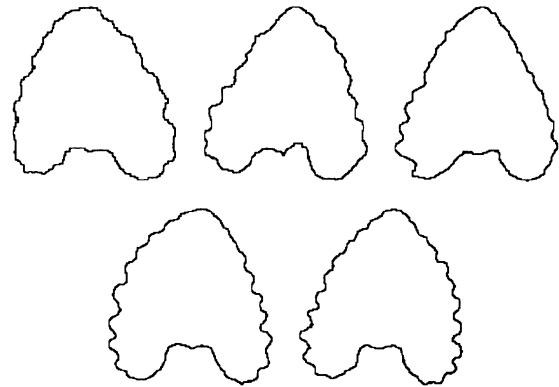


Fig. 2 A family of curves representing a single object.

the problem, consider the shapes shown in Fig. 2. The shapes shown are instances of the same object—they are all ivy leaves. Yet, Hough-like transforms typically do not recognize them easily, no matter which shape is used as the model. The primary reason is that for the Hough techniques to succeed there must be enough localities in the image that match exactly with the model shape. In the example shown in Fig. 2, this condition does not hold and, hence, the Hough techniques are not effective.

This problem is encountered in many classes of objects. They include most natural objects, e.g., leaves, flowers, tree silhouettes, and human face silhouettes. For the GHT approach to work, there must be enough subparts in the unknown curve that match exactly with the corresponding parts in the known curve. Most industrial objects usually adhere to this property. They have rigid shapes, and if an object does not match the model (almost) exactly, it is rejected.

On the other hand, most natural shapes do not have this property. Shapes perceptually grouped into one class are seldom exact matches of each other. The shapes in Fig. 2 do not match exactly in most places to a model shape, yet we usually accept them to be the same object because the salient features of the shape are the same in all objects.

Our goal was to be able to model the shape with a small set of generic templates that could model all the variations of the shapes, yet retain their salient features. This led to the principal component analysis. A set of principal components is derived from the analysis that can effectively model the shapes. Principal components form the basis template set and constitute the working set for the shape matcher. These basis templates, called *eigenshapes* are used on a training set to determine best threshold values for shape/nonshape discrimination. The tradeoffs involved in selecting a threshold can also be determined.

3 Principal Component Analysis for Natural Shapes

Initial experiments in template matching and the straightforward application of GHT indicated that the shapes may have wide variations, yet they are all similar in many respects. To be successful these changes must be modeled. The main reason for the relative failure of the standard GHT is its rigidity. A generic template set must be built that accounts for the most possible occurrences of shapes while preserving the uniqueness of the shape. Principal component analysis provides a mechanism to achieve this.

Principal component analysis, also called Karhunen-Loeve(K-L) expansion,¹¹⁻¹⁴ is a technique used extensively in signal processing.^{15,16} It has also been used to model human face silhouettes.¹⁰ A set of shapes are represented by a set of eigenshapes in this approach. The eigenshapes are the eigenvectors of the correlation matrix of the shapes and can be used to reconstruct the shapes. For some shapes (e.g., human faces), it is generally believed that the dimension of the shape space is relatively low.^{11,12} Thus the eigenshapes provide a way to reduce the complexity of the problem by projecting shapes into the space spanned by the eigenshapes.

Consider a set of m shapes (represented by the boundary) s_1, s_2, \dots, s_m ; algebraically, principal components are linear combinations of the m shapes; geometrically, these represent a new coordinate system obtained by rotating the original system with s_1, s_2, \dots, s_m as the coordinate axes. The new axes represent the directions with maximum variation of the shapes.

Often, there is a smaller set of size m' out of the total m components that accounts for most of the variations. The m' principal components can be used instead of the initial m principal components. Thus the original data set of m shapes is reduced to m' principal components. This analysis results in a new coordinate system into which a shape can be mapped by a small set of mutually independent coefficients.

The goal of this research is to detect shapes of natural objects. The principal component analysis performed on a set of shapes yields the eigenshapes that are used for recognition. This training set of shapes must be heterogeneous to accommodate diversity in the shapes.

3.1 Analysis

Let us assume that a shape s_i is represented by an $n \times n$ array of bits. (It is typically not a square image, but the analysis is similar.) Thus the shapes span a large dimensional (n^2) space in which each shape is represented by a single point. The shapes form a cluster in a small neighborhood. This cluster has a dimension much smaller than n^2 , and principal

component analysis provides a mechanism to represent the shapes in this lower dimensional space.

Let the training set of shapes be $\{s_1, s_2, \dots, s_m\}$ where each s_i , the i 'th shape, is represented as an $n^2 \times 1$ vector. This set of very large vectors can then be used for principal component analysis, to derive a set of m orthonormal vectors, $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$. This set best describes the variations in the shapes. The i 'th vector is chosen such that

$$\lambda_i = \frac{1}{m} \sum_{j=1}^m (\mathbf{u}_i^t s_j)^2 \quad (1)$$

is a maximum, subject to the following orthogonality constraints:

$$\mathbf{u}_k^t \mathbf{u}_l = \begin{cases} 1 & \text{if } l=k \\ 0 & \text{if } l \neq k \end{cases} ,$$

where \mathbf{u}_k^t is the transpose of \mathbf{u}_k . (Note: this notation is used to denote the transpose of matrices for the rest of this section.)

The vectors \mathbf{u}_i and scalars λ_i are the eigenvectors and eigenvalues, respectively, of the correlation matrix:

$$\mathbf{Y} = \mathbf{X}\mathbf{X}' = \frac{1}{m} \sum_{i=1}^m s_i s_i^t , \quad (2)$$

where $\mathbf{X} = [s_1, s_2, \dots, s_m]$ is an $n^2 \times m$ matrix. Thus, $\mathbf{X}\mathbf{X}'$ satisfies the following property:

$$\mathbf{X}\mathbf{X}' \mathbf{u}_i = \lambda_i \mathbf{u}_i , \quad \forall_i 1 \leq i \leq m . \quad (3)$$

However, the matrix \mathbf{Y} is $n^2 \times n^2$, and computing the eigen system for such a large matrix is computationally impractical for typical shapes, in which n is of the order of 100s. Because the number of data points in the shape space is small (i.e., $m \ll n^2$), there is only a small set of (m , rather than n^2) meaningful eigenvectors. The m most significant eigenvectors can be obtained using the following analysis.

Consider another product \mathbf{Z} defined as

$$\mathbf{Z} = \mathbf{X}'\mathbf{X} , \quad (4)$$

where \mathbf{Z} is an $m \times m$ matrix. Its eigenvectors \mathbf{v}_i 's and eigenvalues μ_i 's satisfy

$$\mathbf{Z}\mathbf{v}_i = \mathbf{X}'\mathbf{X}\mathbf{v}_i = \mu_i \mathbf{v}_i . \quad (5)$$

Premultiplying Eq. (5) by \mathbf{X} , we have

$$\mathbf{X}\mathbf{X}'\mathbf{X}\mathbf{v}_i = \mathbf{X}\mu_i \mathbf{v}_i , \quad (6)$$

$$\mathbf{X}\mathbf{X}'(\mathbf{X}\mathbf{v}_i) = \mu_i (\mathbf{X}\mathbf{v}_i) . \quad (7)$$

Equations (3) and (7) show that $\mathbf{X}\mathbf{v}_i$'s are the eigenvectors of the matrix $\mathbf{Y} = \mathbf{X}\mathbf{X}'$. Thus if \mathbf{v}_i is an eigenvector of $\mathbf{Z} = \mathbf{X}'\mathbf{X}$ then $\mathbf{X}\mathbf{v}_i$ is an eigenvector of $\mathbf{Y} = \mathbf{X}\mathbf{X}'$. Since \mathbf{Z} is an $m \times m$ matrix, finding \mathbf{v}_i 's is more tractable. Then the eigenvectors of \mathbf{Y} can be obtained with one matrix multiplication operation.

With this analysis, the matrix \mathbf{Z} [See Eq. (4)] is first constructed and its m eigenvectors \mathbf{v}_i 's are then computed. These vectors are then used to derive the eigenvectors \mathbf{u}_i 's of \mathbf{Y} :

$$\mathbf{u}_i = \mathbf{X}\mathbf{v}_i \quad (8)$$

This analysis shows that the computation is now reduced from the order of the size of the shape to the order of the size of the training set. In practice, the training set is relatively small ($m \ll n^2$). The eigenvalues obtained rank the contribution of the individual eigenshapes to their significance in the training set. This enables the selection of $m' (< m)$ eigenshapes that represent the most significant eigenvectors. Each additional eigenshape means additional time spent in matching. Because extreme accuracy in reconstruction is usually not necessary, the smaller set is adequate for most applications.

3.2 Regeneration of the Shapes

The eigenshapes represent the space spanned by all the shapes. Hence, in principle, it should be possible to describe any shape using the eigenshapes. In particular, it should be possible to reconstruct the training set. Indeed, this can be achieved using the following steps:

1. Project any given shape s onto the eigenshape space to obtain the coefficients. These coefficients, called *weights*, (ω_i 's) are derived as follows:

$$\omega_i = \mathbf{u}_i^t s \quad , \quad i = 1, 2, \dots, m \quad (9)$$

The weights form a vector $[\omega_1, \omega_2, \dots, \omega_m]$.

2. The shape s can then be obtained by a linear combination of the eigenshapes with the weights as the coefficients:

$$s = \sum_{i=1}^m \omega_i \mathbf{u}_i \quad (10)$$

3.3 Recognition

The motivation for using principal component analysis is the generation of a generic template set to be used in recognition. These eigenshapes are then used in the matching process. A GHT-based approach is used to detect each eigenshape in a given image. The results of matching for each eigenshape are then combined to infer the presence or absence of the given shape in the image. This process is divided into two stages: (a) training phase and (b) recognition phase.

In the training phase, the eigenshapes are derived by means of a set consisting of a variety of samples of a given shape. These are then used to detect the shape in a set of images in which the presence or absence of a shape is known. Because a GHT-based approach is used, it is important to select good threshold values for matching. It should be chosen to minimize errors in recognition. Some automatic techniques to threshold determination are also described.

During recognition, a GHT-like approach is used to determine the response to each eigenshape. Classification of a pattern as either a shape or a nonshape is based on the responses and the thresholds selected during training.

3.3.1 Training

The goal of training is twofold: (a) select the most significant eigenshapes and (b) determine the thresholds associated with them. It should be noted at the outset that the computation

is a one-time expense. The steps in training are shown in Fig. 3.

The shapes in the training set should be representative of the whole class. Step 1, i.e., eigenshapes generation, was explained in detail earlier (see Sec. 3.1). The last two steps are explained next.

Threshold determination. The selection of thresholds for shape/nonshape discrimination is critical to any GHT-based approach. First, the GHT is used to generate the votes for shapes and nonshapes in the training set. This is performed for each eigenshape. For each eigenshape the distribution of votes for the shape set and that for the nonshape set must be compared to determine if there is sufficient difference between the two. Statistical tests like the t test¹⁷ may be used for this purpose. Figure 4 shows some sample distributions. They show the numbered objects receiving a given number of votes. The plots for the shapes are shown using dotted lines and those for nonshapes by solid lines.

Assuming that the two distributions are significantly different, the next step is to select a threshold for shape/nonshape discrimination for each eigenshape. It can be done empirically by taking the mean of the means of the two distributions as the threshold. However, if it is known that the two distributions are normal, better estimates can be found.

A distribution can be tested for normality using the standard χ^2 test.¹⁷ If a distribution is normal, its $Q-Q$ plot should approximately be a straight line. A systematic way for selecting the threshold and estimating the associated errors is given in detail in Sec. 3.3.2.

Selection of significant eigenshapes. The principal component analysis yields m eigenshapes, but in practice only $m' (< m)$ of them are sufficient to approximately represent all the points in the shape space. If the two distributions have very little overlap and are separate [Fig. 4(a)], the eigenshape is a good discriminator. If, however, the distributions have large overlapping areas, [Fig. 4(b)] the corresponding eigenshape is not useful. The number of eigenshapes, determined empirically, should be small, yet capable of modeling the variations in the shapes.

3.3.2 Threshold selection

Figure 4 shows that typical distributions of votes for shape images and nonshape images for an eigenshape overlap. This makes the determination of the threshold value more difficult. A lower threshold increases the chance of a nonshape being classified as a shape. Conversely, a higher threshold increases the probability of a shape being misclassified as a nonshape. Hence, the threshold for each eigenshape must be carefully selected.

For the analysis in this section, we consider the response to a single eigenshape only. Assume that the observed mean and variance of the votes for n_f shapes, $\{f_1, f_2, \dots, f_{n_f}\}$, are m_f and s_f^2 , respectively. Similarly, let the observed mean and variance of the votes for $n_{f'}$ nonshapes, $\{f'_1, f'_2, \dots, f'_{n_{f'}}\}$, be $m_{f'}$ and $s_{f'}^2$.

Our goal is to choose a threshold t such that if an unknown pattern receives more votes, it will be classified as a shape and vice versa. We select a value that is a weighted average of the means of the two distributions:

$$t = w m_f + (1 - w) m_{f'} \quad , \quad 0 \leq w \leq 1 \quad (11)$$

```

Input:      s = {s1, s2, ..., sm} /* Shapes */
              ns = {ns1, ns2, ..., nsk} /* Non-Shapes */
Output:   su = {u1, u2, ..., um'} /* Eigenshapes */
              t = {t1, t2, ..., tm'} /* Thresholds */

BEGIN

1. u = Principal_Component_Analysis(s);
2. t = Compute_Threshold(s,ns,u);
3. su = Significant_Eigenshapes(u);

END
    
```

Fig. 3 The training algorithm.

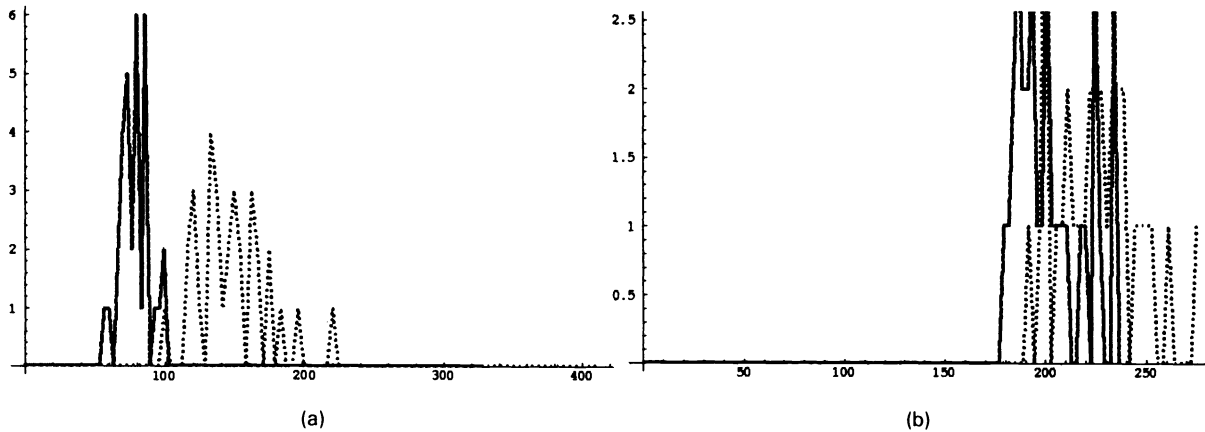


Fig. 4 Voting distributions for two eigenshapes: (a) little overlap and are separate and (b) large overlapping areas.

where w is the weight factor.

Two types of errors can occur in the classification of a pattern as a shape or a nonshape.

- **False shape:** A nonshape is classified as a shape, i.e., the number of votes obtained by a nonshape f' is greater than the threshold t . Let the probability of such occurrence be denoted by $p_{f':f}$. This is shown by the region in the darker shade in Fig. 5.
- **False nonshape:** A shape is classified as a nonshape if the number of votes obtained by a shape f is less than the threshold t . Let the probability of such an occurrence be denoted by $p_{f:f'}$. This is shown by the lightly shaded region in Fig. 5.

Normal distribution. If the votes corresponding to each eigenshape follow a normal distribution, better estimates for the threshold can be found. In addition, it is possible to derive approximate values for the probabilities for false classification (see Ref. 9 for details). Note that in our experiments statistical tests show that the votes do follow normal distributions. The voting patterns corresponding to a single eigenshape are shown in Fig. 5. Figure 5 shows the number of objects receiving a given number of votes for that eigenshape.

The choice of w [see Eq. (11)] determines the two probabilities $p_{f':f}$ and $p_{f:f'}$. Clearly, both probabilities cannot be

simultaneously minimized, and one must trade off one for the other. As can be seen in Fig. 5, the total misclassification error is the lowest if the threshold corresponds to the point of intersection of the two curves. The point of intersection of the normal distribution curves can be determined by solving the following equality:

$$\frac{1}{\sqrt{2\pi}s_f} \exp\left[-\frac{1}{2}\left(\frac{t-m_f}{s_f}\right)^2\right] = \frac{1}{\sqrt{2\pi}s_{f'}} \exp\left[-\frac{1}{2}\left(\frac{t-m_{f'}}{s_{f'}}\right)^2\right], \quad (12)$$

which reduces to

$$2 \ln \frac{s_{f'}}{s_f} = \left[\left(\frac{t-m_f}{s_f}\right)^2 - \left(\frac{t-m_{f'}}{s_{f'}}\right)^2 \right]. \quad (13)$$

Given the values for m_f , $m_{f'}$, s_f , and $s_{f'}$, the value for the optimal threshold t can be determined by using Eq. (13). This threshold minimizes the total errors in the system. However, if one requires that most of the shapes be correctly recognized and is prepared to accept that a higher probability of non-shapes are classified incorrectly, the value of w in Eq. (11) can be chosen appropriately.

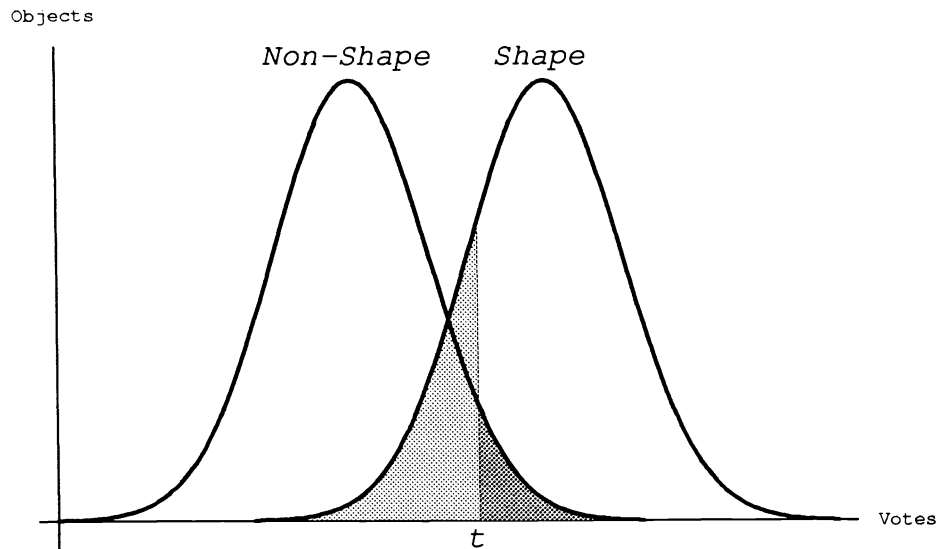


Fig. 5 Illustration of misclassification with two normal distributions.

```

Input:      Image[r × c]          /* Scene Image */
Output:    {(xi, yi)}          /* Locations of shapes in the image */

BEGIN
1. Initialize_Accumulator(Accumulator);

   /* Collect votes at each point for each eigenshape */
2. for each eigenshape, uk ∈ {u1, u2, ..., um'}
   Collect_Votes(Image, Accumulator, k, rtk)

3. Locate_Shapes(Accumulator, t)

END

```

Fig. 6 The recognition algorithm.

3.3.3 Shape recognition

With the principal component analysis, a given shape is modeled as a set of m' eigenshapes. Each eigenshape in turn is represented as a reference table that is used for the GHT. The shape recognition algorithm uses a GHT-like procedure for each eigenshape. Thus, each point in the image receives m' votes, one for each eigenshape. Depending on the votes collected by the point in the image, the point is either marked as a shape or a nonshape. The recognition algorithm is sketched in Fig. 6. Because a point receives m' votes, the accumulator A has a size of $m' \times r \times c$, in which the image size is $r \times c$. The eigenshape set is $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{m'}\}$ and the reference table for shape \mathbf{u}_i is denoted by rt_i . Let the set of threshold values for shape/nonshape discrimination (see Sec. 3.3.2) for the eigenshapes be $\mathbf{t} = \{t_1, t_2, \dots, t_{m'}\}$.

Step 2 is similar to the traditional Hough vote collection for a single shape in a given image. The k 'th plane of the accumulator stores the votes for the k 'th eigenshape. The "Locate_Shapes" procedure (step 3) determines which

points in the image are to be labeled as matches for the given shape. It compares the votes collected at each point for each eigenshape with the corresponding threshold value. In the ideal case, a point where the shape is located receives high votes for each eigenshape and vice versa. In practice, however, it is rarely the case. Thus the decision process must be more sophisticated. Two simple approaches are described below: (a) majority vote and (b) average vote.

In the majority voting scheme, if a point receives votes greater than the corresponding threshold for a majority of eigenshapes, the point is classified as a shape point. The average voting scheme measures the goodness of match between the subimage centered at a given point and the eigenshape. If a point receives votes v_i for the eigenshape \mathbf{u}_i , the average vote for the point is $(\sum_{i=1}^{m'} v_i)/m'$. If the average vote is greater than a suitable threshold, the point is classified as a shape point. While the majority vote strategy is more space and time efficient, both approaches are quite efficient in terms of computation time.

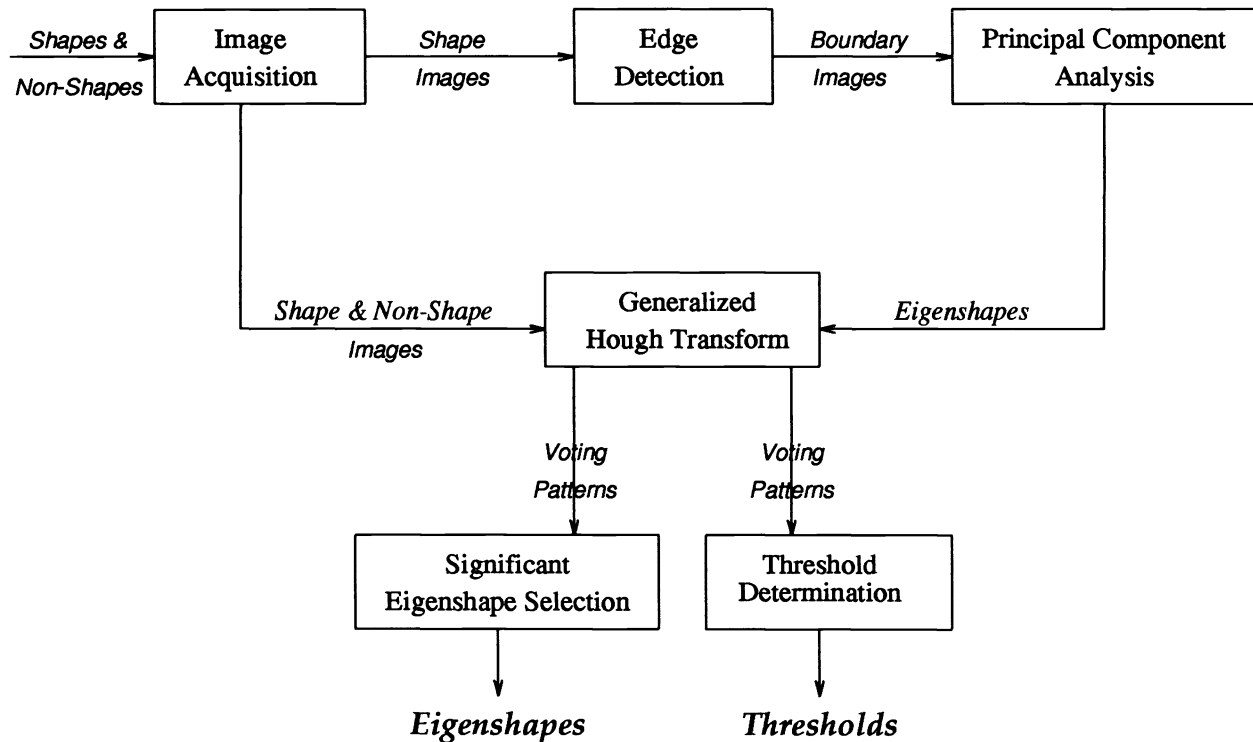


Fig. 7 Detailed layout of the training module.

4 Implementation and Results

This section presents the experimental results of the shape recognition approach using principal components. The goal of the recognition system is to distinguish a pattern that belongs to a class, i.e., a shape pattern, from a pattern that does not, i.e., a nonshape pattern.

The recognition system is composed of the training module and the recognition module. As explained before, during training a set of shapes for a given class is obtained. Then the eigenshapes and their corresponding thresholds for shape/nonshape discrimination are computed. The recognition subsystem uses this knowledge to detect shapes in a given image with the approach explained in Sec. 3.3.3.

The system is implemented on a SUN Sparcstation with a SUN VideoPix 8-bit frame grabber and a Sony video camera. The images are 640×480 in size and are in SUN Raster file format. Several software packages are used by different parts of the system. They are IKS (image kernel system) from the University of Utah; KBVision by Amerinex Artificial Intelligence, Inc.; and Mathematica by Wolfram Research Inc. The system is written in the C programming language, using some shell scripts.

4.1 Training

The training module performs four major functions: (a) it obtains the set of shape and nonshape images to be used for training, (b) it computes the outlines for the shapes, (c) it computes the eigenshapes using the principal component analysis, and (d) it selects the most significant eigenshapes and determines their corresponding thresholds. The steps in the training module are sketched in Fig. 7.

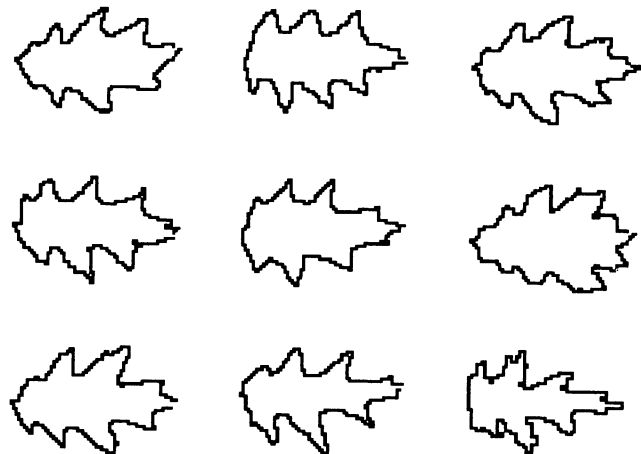


Fig. 8 Some shapes used for training.

Image acquisition. For experimentation, a set of 32 leaves of a particular type are selected. The shapes are approximately centered in a well-lighted scene and the images are acquired under these conditions. Except for these constraints, the acquisition of the training set is very general. The nonshape images are selected from an existing database of miscellaneous images.

Edge detection. The images for the shapes are converted to outline images using edge detection. All the shapes are scaled to their average size, 44×66 . Some leaf shapes are shown in Fig. 8.

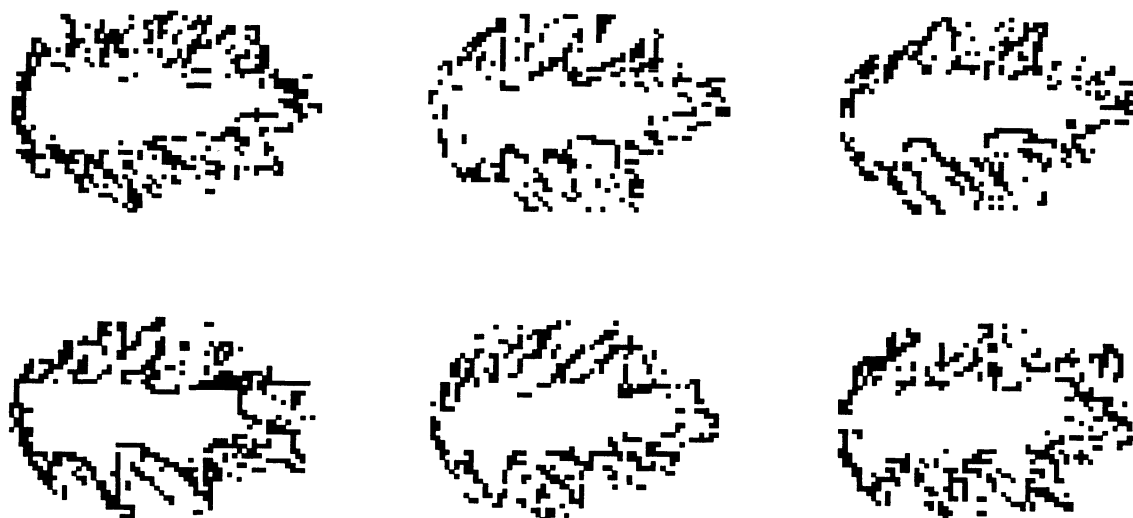


Fig. 9 Some eigenshapes generated by the principal component analysis.

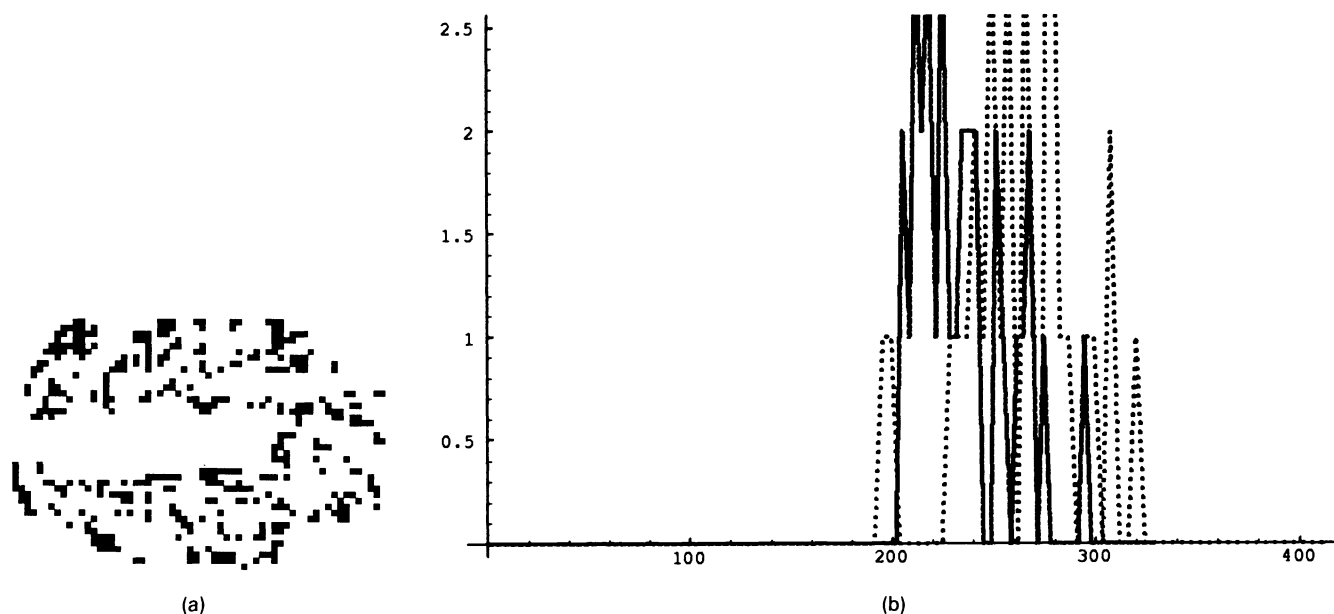


Fig. 10 (a) A rejected eigenshape and (b) its response to the training set.

Principal component analysis. The images are represented as linear vectors. Mathematica software was used to generate the eigenvectors and eigenvalues (see Sec. 3). The eigenvectors correspond to the eigenshapes.

One of the problems with the eigenshapes obtained by the principal component analysis is that they are gray-scale images. To use these with the GHT, they are thresholded to derive the most significant boundary points. Some thresholded eigenshapes are shown in Fig. 9. The eigenshapes resemble the training set shapes fairly closely.

Significant eigenshape selection. The eigenshapes are used as models for the shape, and they are used to obtain the voting distributions for each known shape and nonshape image in the training set. The number of eigenshapes is reduced

from 32 to 18 based on the overlap of shape and nonshape vote distributions. The criteria for rejection are discussed in detail in Sec. 3.3.1. Figures 10 and 11 show two eigenshapes and the corresponding voting distributions.

Figure 10 shows that after the eigenshape is thresholded the significant points leave many holes in the boundary, which would no longer define the outline of a typical shape. This is the reason for their failure to match shapes. Therefore, the eigenshape in Fig. 10 is rejected. As a general rule, in the case of the rejected eigenshapes, the probability of misclassification is high regardless of the choice of threshold.

In Fig. 11, there is a region of overlap between the shape and nonshape distributions. However, a threshold that can correctly classify most of the images does exist. This eigenshape on thresholding has the distinctive shape outline pre-

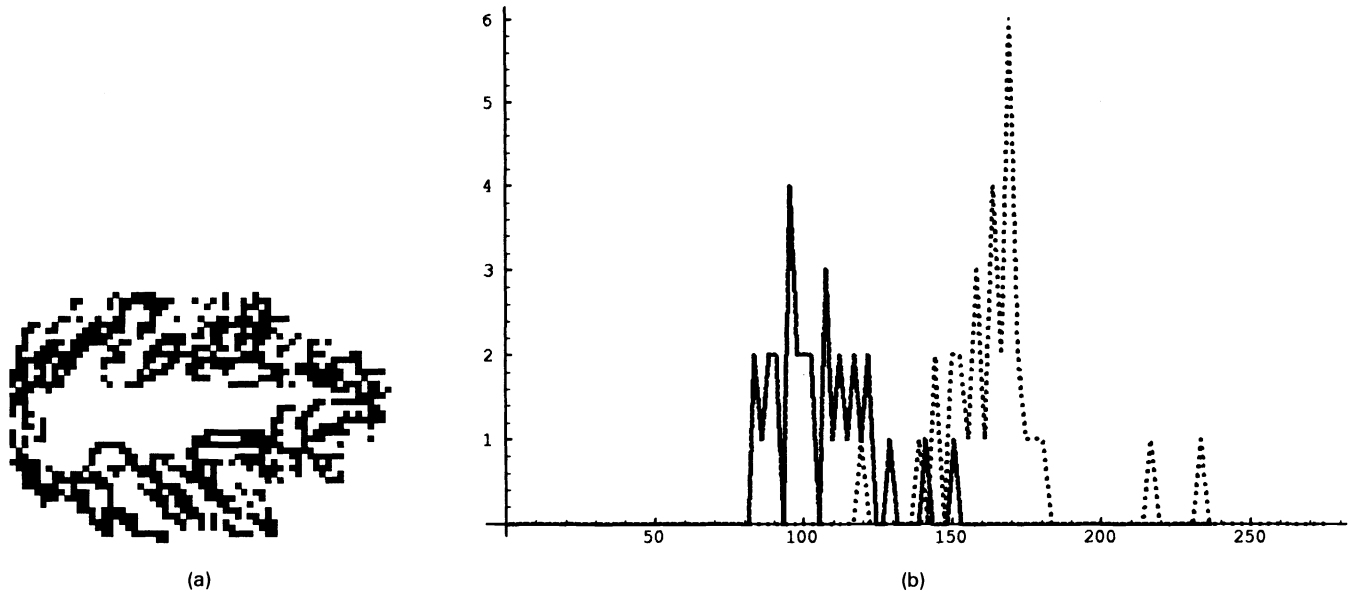


Fig. 11 (a) An accepted eigenshape and (b) its response to the training set.

Table 1 Voting pattern for some test images.

		EIGENSHAPES									
	ID	1 (164)	2 (157)	3 (139)	4 (159)	5 (155)	6 (148)	7 (214)	8 (139)	9 (185)	10 (119)
S	1	183	192	172	173	189	170	336	178	234	165
	2	198	163	178	196	243	201	257	173	218	145
H	3	191	185	199	208	216	173	259	193	198	152
	4	217	200	154	180	199	194	254	160	237	135
A	5	188	169	166	182	175	180	269	198	198	161
	6	186	191	158	186	184	161	237	193	194	144
P	7	209	215	154	214	208	208	242	172	192	157
	8	179	183	162	181	166	160	261	161	187	164
E	9	193	214	205	180	184	180	267	190	207	152
	10	182	202	188	179	169	151	273	185	207	155
N	1	133	134	136	142	143	131	241	129	152	82
O	2	135	106	100	120	137	125	180	98	152	75
N	3	144	116	99	123	114	109	175	118	191	76
-	4	131	103	89	120	108	107	145	126	182	72
S	5	152	117	108	131	147	132	187	112	141	88
H	6	136	136	125	146	166	126	183	133	170	92
A	7	129	124	97	128	140	132	173	113	147	74
P	8	142	163	103	131	135	121	197	111	138	77
E	9	119	126	91	123	115	126	178	114	128	72
	10	107	128	101	116	125	109	166	109	142	64

served, even though it is not continuous. Hence, the eigenshape in Fig. 11 is significant and, hence, is accepted.

Threshold selection for each of the 18 significant eigenshapes is performed by the analysis given in Sec. 3.3.1. The value of w was chosen to be 0.5, i.e., the threshold was the mean of the two means (shape and nonshape). Thresholds for some of the eigenshapes are shown in Table 1 (in parentheses below the eigenshape index).

The class of shapes given in the training set is now represented by the set of significant eigenshapes. The reference tables for each significant eigenshape is computed and is stored. They are used with the GHT during recognition.

4.2 Recognition

A new set of shapes and nonshapes were used for testing the effectiveness of the recognition system. A detailed layout of

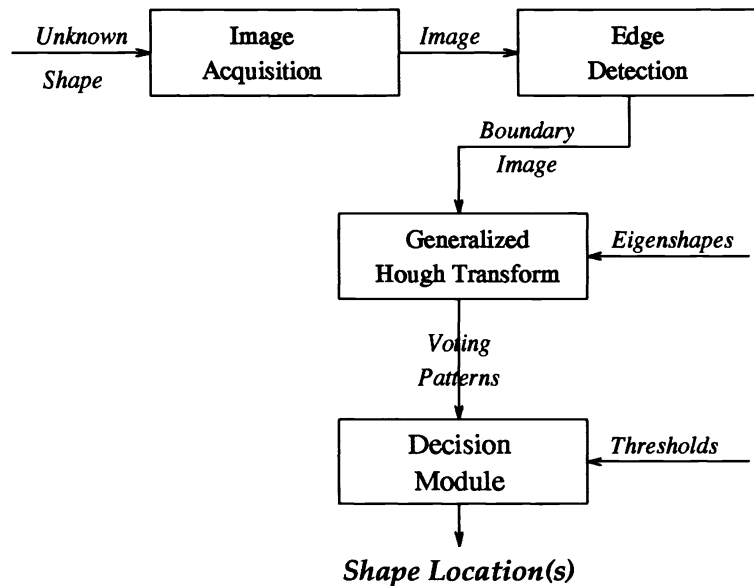


Fig. 12 Detailed layout of the recognition module.

the recognition subsystem is given in Fig. 12. The eigenshapes and the corresponding thresholds for the eigenshapes were generated in the training stage.

A total of 92 shape images were used. All the shapes were obtained under conditions similar to those in training. A set of 56 general images (nonshape) was also included in the test set. The set consisted of images with different types of leaves in addition to images that had no leaves. Some test shapes and nonshapes are shown in Figures 13 and 14, respectively.

Vote collection for each eigenshape was performed by a GHT-like counting algorithm. Table 1 shows a sample voting table for 10 shapes and 10 nonshapes.

Both the majority vote scheme and the average vote scheme were used for the final classification. The system performed well with both the strategies. The success rate for nonfaces for both schemes was 98%, and each misclassified only one nonshape. The majority scheme correctly detected 96% of the test shapes, while the average vote scheme was 94% accurate. The overall success rates for the two schemes were 97 and 95%, respectively.

The recognition time was relatively small. The time for edge detection was negligible and recognition time was approximately 20 s. The recognition time can be improved by optimizing the modules and also by reducing the number of significant eigenshapes.

5 Summary and Future Research

Most current techniques to recognize natural shapes are ineffective and inconsistent. The main reason for this is that the natural shapes are not rigid. A new approach to shape recognition is presented in this paper. The approach is based on multiple template matching using a Hough-like transform. The variations in the shape are represented in a template set that is generated from a general set of shapes using the principal component analysis. Experimental results show that shapes can be detected accurately using this approach.

The major deficiency in the current system is that it does not incorporate scaling and rotation. Existing approaches can be dovetailed into the system without much difficulty, e.g., the scheme to incorporate scaling in GHT given in Ref. 7.

Future directions for research include testing the approach with larger test sets as well as with other natural classes of shapes, e.g., flowers, trees, etc. More thorough testing with a larger test set will lead to more accurate values for the thresholds and will reduce the margin for errors. We also intend to explore systematic and automatic methods to reduce the set of significant eigenshapes.

Acknowledgment

This work is supported in part by NSF grants CDA-9022445 and USE-9152764.

References

1. P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent No. 3 069 654 (Dec. 18, 1962).
2. Richard O. Duda and Peter E. Hart, "Use of Hough transformation to detect lines and curve in pictures," *Commun. ACM* **15**(1), 11–15 (Jan. 1972).
3. Philip M. Merlin and David J. Farber, "A parallel mechanism for detecting curves in pictures," *IEEE Trans. Comput.* **C-24**(1), 96–98 (Jan. 1975).
4. D. H. Ballard "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recog.* **13**(2), 111–122 (1981).
5. V. Cantoni and L. Carrioli, "Structural shape recognition in a multi-resolution environment," *Sig. Process.* **12**, 267–276 (1987).
6. L. S. Davis, "Hierarchical generalized Hough transforms and line-segment-based generalized Hough transforms," *Pattern Recog.* **15**, 277–285 (1982).
7. S.-C. Jeng and W.-H. Tsai, "Scale- and orientation-invariant generalized Hough transform—a new approach," *Pattern Recog.* **24**(11), 1037–1051 (1991).
8. R. Krichnapuram and D. Casasent, "Hough space transformations for discrimination and distortion estimation," *Comput. Vis. Graph. Image Process.* **38**, 299–316 (1987).
9. P. Iyengar, "Human face detection using silhouettes," Master's Thesis, University of Nebraska-Lincoln (Dec. 1992).
10. A. Samal and P. Iyengar, "Human face detection using silhouettes," submitted to the *Intl. J. Pattern Recog. Artif. Intell.* (Jan. 1993).
11. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *OSA* **4**(3), 519–524 (March 1987).
12. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neuro-Sci.* **3**(1), 71–86 (1991).



Fig. 13 Some shapes used for testing.

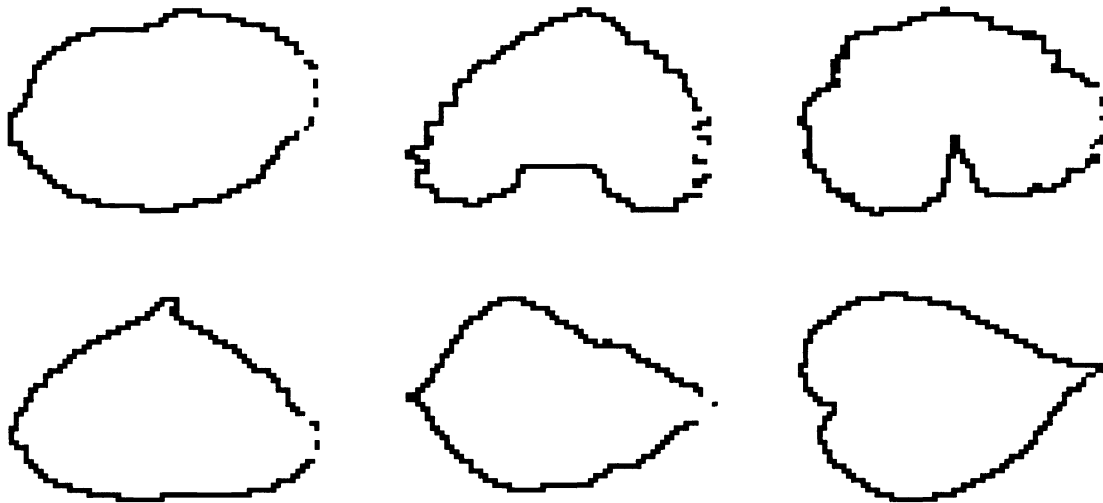


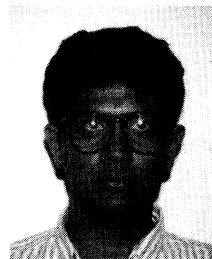
Fig. 14 Some nonshapes used for testing.

13. J. Kittler and P. C. Young, "A new approach to feature selection based on the Karhunen-Loeve expansion," *Pattern Recog.* 5, 335-352 (1973).
14. B. V. K. V. Kumar, D. Casasent, and H. Murakami, "Principal-component imagery for statistical pattern recognition correlators," *Opt. Eng.* 21(1), 043-047 (Jan./Feb. 1982).
15. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey (1989).
16. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., New York (1984).
17. R. A. Johnson and D. A. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall, Englewood Cliffs, New Jersey (1988).



Ashok Samal received his Bachelor of Technology degree in computer science from the Indian Institute of Technology at Kanpur, India, in 1983. He received his PhD in computer science from the University of Utah in 1988. Since Fall 1988 he has been with the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. His primary research interests are computer vision, image processing, parallel and distributed

systems for computer vision, genetic algorithms, and neural networks. He has published articles in these areas in journals and conference proceedings. He is also interested in artificial intelligence, computer graphics, operating systems, and functional languages.



Sunnyvale, California. His research interests include computer vision, networking, and software engineering.

Prasana A. Iyengar received his BE in computer engineering from Bombay University, India, in July 1989. He was a trainee software engineer for a year at Godrej & Boyce Manufacturing Company Ltd., Bombay, India. In December 1992 he graduated with an MS from the Computer Science and Engineering Department, University of Nebraska-Lincoln. Since graduation he has been employed as a software engineer by Pretzel Logic Software Inc.,