Partial Shape Classification Using Contour Matching in Distance Transformation

Hong-Chih Liu and Mandyam D. Srinath, *Senior Member, IEEE*

Abstract— **In this paper, an algorithm is presented to recognize and locate partially distorted two-dimensional shapes without regard to their orientation, location, and size. The proposed algorithm first calculates the curvature function from the digitized image of an object. The points of local maxima and minima extracted from the smooth curvature function are used as control points to segment the boundary and to guide the boundary matching procedure. The boundary matching procedure considers two shapes at a time, one shape from the template data bank, and the other being the object under classification. The procedure tries to match the control points in the unknown shape to those of a shape from the template data bank, and estimates the translation, rotation, and scaling factors to be used to normalize the boundary of the unknown shape. The chamfer 3/4 distance transformation and a partial distance measurement scheme are used as the final step to measure the similarity between these two shapes. The unknown shape is assigned to the class corresponding to the minimum distance. The algorithm developed in this paper has been successfully tested on partial shapes using two sets of data, one with sharp corners, and the other with curve segments. This algorithm not only is computationally simple, but also works reasonably well in the presence of a moderate amount of noise.**

Index Terms— **Distance transformations, edge matching, image recognition, local features, shape identification.**

I. Introduction

RECOGNITION of shapes which are only partially correct is important in many image analysis applications. In this paper, we present an algorithm to recognize and locate two-dimensional shapes which are partially distorted. The algorithm is designed with the following objectives.

1) The algorithm should be able to handle images which contain sharp corners, e.g., industrial parts or aircraft, as well as those images which contain only curve segments, e.g., the shape of a lake.
2) The algorithm must be computationally simple.
3) The algorithm must be responsive to arbitrary changes in orientation, position, and scale of the objects under classification.

The proposed algorithm tries to estimate the orientational, scaling, and translational data between a test shape and model shape by using a small number of control points extracted from both shapes. The goodness of this estimation can be represented by a numerical value.

The algorithm first calculates the curvature functions from the digitized image of an object. The points of local maxima and minima are extracted from the smoothed curvature function, and are used as control points to segment the boundary and to guide the boundary matching procedure. The boundary matching procedure considers two shapes at a time, one shape from the template data bank, and the other being the object under classification. The procedure tries to match the control points in the test shape to those of a shape from the template data bank, and estimates the translation, rotation, and scaling factors to be used to normalize the boundary of the test shape. The chamfer 3/4 distance transformation [3] and a partial distance measurement scheme are proposed as the final step to measure the similarity between these two shapes. The test shape is assigned to the class corresponding to the minimum distance.

This algorithm differs from many of the previous methods for partial shape classification. The Bolles and Cain technique [2] requires knowledge about the scale of the object under classification. Their algorithm also requires that objects contain several special local features such as holes and corners. Objects that are characterized by large and continuous arcs cannot be located. Gorman *et al.*'s and Kuhl's algorithm [8] can recognize objects with different orientation and location. Precise knowledge about the scale of the object is not required. But an automatic method for selecting a threshold value to perform the polygon approximation required in the algorithm is not presented. Appropriate threshold values must be found experimentally. Mitchell and Gogan's [13], Singer and Chellappa's [16], Lin and Chellappa's [12], and Tejwani and Jones's [17] algorithms are very computationally expensive. Knoll and Jain's [11] and Perkins' [14] algorithms require precise knowledge of the scale. It can be seen that most of the previous works with partial shape classification either concentrate on shapes with known scale or are very computationally expensive. The algorithm developed in this paper has been successfully tested on partial shapes using two sets of data (one with sharp corners and the other with curve segments) without knowledge about the scale, orientation, and location of the object. The algorithm not only is computationally simple, but also works reasonably well in the presence of a moderate amount of noise.

The paper is structured as follows. In Section II, boundary segmentation based on control point extraction from the smoothed curvature function is discussed. The boundary matching procedure is presented in Section III. In Section IV, a chamfer 3/4 distance transformation and partial distance measurement scheme are presented to calculate the goodness of the match between two shapes. Experimental procedures and results are presented in Section V. Section VI provides conclusions.

Manuscript received January 15, 1990; revised March 12, 1990. This work was supported in part by the Defense Advanced Research Projects Agency under Grant MDA-903-86-C-0182.

H.-C. Liu is with the Institute of Nuclear Energy Research, P.O. Box 3-11, Lung-Tan, Taiwan, Republic of China.

M. D. Srinath is with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275.

IEEE Log Number 9037573.

II. Boundary Segmentation

The present algorithm uses the local maximum and minimum points of the smoothed curvature function to segment the boundary of the object. If the boundary function, defined parametrically by $s(t) = (x(t), y(t))$, is twice differentiable, the *slope function*

0162-8828/90/1100-1072$01.00

(the angle between the tangent line and the positive x axis) of the boundary can be computed by

$$\frac{dy}{dx} = \frac{\dot{y}}{\dot{x}}.$$

The *curvature function* $k(t)$ (the derivative of the slope with respect to the arc length) is defined as

$$k(t) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{[\dot{x}^2 + \dot{y}^2]^{3/2}}$$

where $\dot{x}$, $\dot{y}$, $\ddot{x}$, and $\ddot{y}$ are the first and second derivatives of $x(t)$ and $y(t)$ with respect to t.

In practice, only the digitized boundary functions are available. Thus, the discrete nature of the image causes the curvature to be quantized. Also, because the curvature function is the output of the second derivative of the boundary function, directly using the angle between successive line segments joining the sample points of the digitized boundary function gives rise to unreliable measures of curvature. Techniques such as filtering the curvature function must be introduced to reduce the effect of quantization and to smooth the curvature function.

Fig. 1 shows the process of extracting the curvature function from the object silhouette image. An encoding algorithm is first applied to the input image to obtain the chain code representation of the boundary. The edge gradient direction of each boundary pixel is computed by the gradient direction at each boundary pixel of the binary image. At pixel P whose neighbors have gray levels

$$\begin{matrix} A & B & C \\ D & P & E \\ F & G & H \end{matrix}$$

the Sobel difference is computed by

$$\Delta_x \equiv \frac{1}{4}[(C + 2E + H) - (A + 2D + F)]$$
$$\Delta_y \equiv \frac{1}{4}[(A + 2B + C) - (F + 2G + H)].$$

The edge gradient direction of pixel P is defined as

$$\theta \equiv \tan^{-1}\frac{\Delta_y}{\Delta_x}.$$

A normalization step is performed here to eliminate the artificial discontinuities from wraparound that often occur in the edge gradient direction. More precisely, the edge gradient direction is calculated from the arctangent of the Sobel difference. Since the arctangent function only returns an angle in the range of $(-\pi, \pi)$, any angle direction outside this range is wrapped around, thus resulting in an artificial discontinuity in the edge gradient direction. The normalization step traces the function and searches for local discontinuities greater than π or less than $-\pi$. An offset of $\pm 2\pi$ is added to the function at each following point to correct the wraparound. This will result in a continuous function along the entire contour, with the exception of the initial and final points of the contour. For a closed contour, these two points will always have an artificial discontinuity of 2π.

After normalization, the curvature function is calculated by convolving the edge direction function of the boundary point $\theta(s)$ with the first derivative of a Gaussian function $G'(s)$ [6] where

$$G(s) = \exp\left(-\frac{s^2}{2\sigma^2}\right).$$

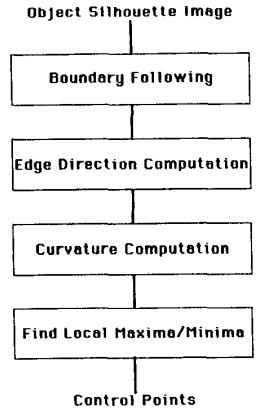


Fig. 1. Image processing, curvature function calculation, and control points extraction.

Varying the value for σ in the filter controls the degree of smoothness of the resulting curve.

The points of local maxima and minima are determined from the smoothed curvature function and are assigned as the control points to break the boundary of the shape into segments. Each shape is represented by an ordered sequence of line segments with control points as vertex points. The value of σ serves to govern the number of vertices that will be found. Proper selection of the value of σ is important to the success of this algorithm. Each line segment must be long enough to have distinguishing characteristics, but also short enough to ensure that an adequate number of segments is used to describe the shape.

Fig. 2 show the curvature functions of several scaled aerial views of Lake Huron. These curvature functions have been smoothed by the first derivative of the Gaussian function, with the value of σ being chosen proportional to the number of boundary points of the shape. From these results, it is clear that when the number of boundary points of a complete shape is known, the curvature functions can be properly smoothed such that objects of the same class have similar curvature functions. The use of control points extracted from these curvature functions will therefore segment these shapes in the same manner, without regard to size, orientation, and location. When the size of the shape is unknown, an automatic method to select the value of σ to make the local maxima and minima of the smoothed curvature function meet the above criteria must be developed. Experimental studies show that choosing a value for σ which breaks the boundary of the shape into approximately 40 segments can meet the above requirements: line segments long enough to have distinguishing characteristics, yet short enough to ensure that an adequate number of segments are used to describe the shape, and breaking up the boundary of a shape in a similar manner irrespective of its size. This choice of 40 segments for the shape may be shape dependent in that it is possible that other segment numbers may perform better on a different set of shapes.

Fig. 3 shows the original shapes and shapes created by linking all the control points extracted from the local maxima and minima of the smoothed curvature function. The number of boundary points for these models ranges from 439 to 713 pixels, and the smoothing factor σ ranges from 4.117 to 6.891. Although this kind of boundary segmentation does not necessarily correspond to the human visual mechanism, it is computationally simple and relatively invariant to size changes.

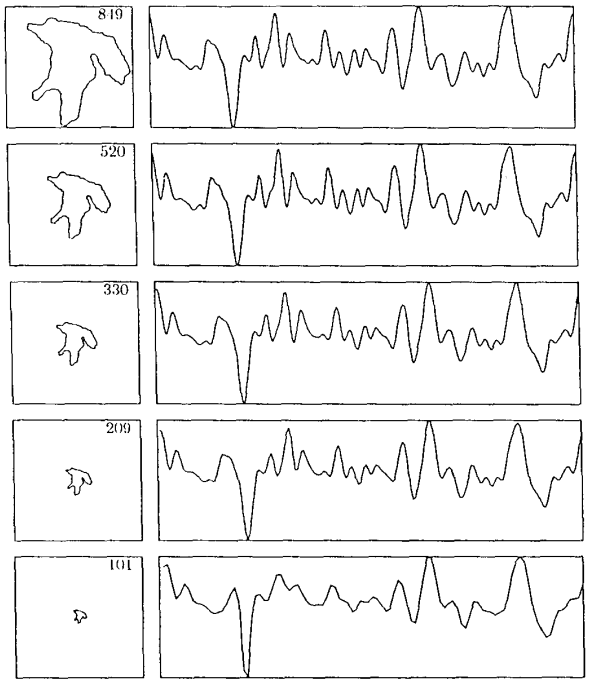


Fig. 2. Shape and curvature function of Lake Huron at different scales. The number in the upper right corner of each shape represents the length of the contour. The curvature functions are normalized along both the θ and s axes to make the comparison easier.

III. Boundary Matching

In the previous section, a process is presented to divide a given shape into segments. A boundary matching process which calculates the rotational, scaling, and translational data between two shapes is presented in this section. The matching process considers two shapes at a time, one shape from the model data bank, and the other being the object under classification. The matching is done from local to global, that is, from matching one segment pair to matching groups of segment pairs. Similar to the method used in [15], there are two passes in the matching process. Pass 1 compares segments between two shapes. Only those segment pairs which are compatible will be used in pass 2. Pass 2 matches groups of segment pairs. The matching process developed in [15] was designed to handle shapes with arbitrary changes in orientation and position. Possible matches are determined by comparing the segment length and by comparing the angle between the current segments and their respective successors. If they do match, then the orientation difference will be stored in an array called the *disparity array*. Since the matching algorithm considered here also has to deal with shapes of different scale, the comparison must be extended to include the length ratio and angle between the current segments and their respective successors. The transformation which maps two segments is thus specified by a quad of parameters (θ, Δx, Δy, S) where θ is the angle difference between two matched segments, S is the length ratio of these two segments, and Δx and Δy are the necessary translations in order to match these two segments after rotation and scaling. The details of the two passes are described in the following sections.

Pass 1—Segment Matching

The segment match between two shapes can be briefly described by the following steps.

Step 1: The boundary of each shape is represented by an ordered sequence of line segments with control points as vertex points.

Step 2: For each shape, number each segment in order around the boundary; thus, consecutive segment numbers correspond to adjacent segments in the boundary.

Step 3: Calculate the length ratio between the current segments and their respective successors.

Step 4: Calculate the angle between the current segments and their respective successors.

Step 5: A segments match is recorded if the data from Steps 3 and 4 of one shape match those data from the second shape within a threshold (chosen by the user).

The algorithm compares every segment of the shape from the template data bank to every segment from the shape under classification. To see how the length ratio and angle between two segments are defined, refer to Fig. 4. Assume that the model shape is represented by vertices $(u_1, v_1), (u_2, v_2), (u_3, v_3)$ and that the unknown object is represented by vertices $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$. Let the segment pairs to be matched be bounded by $[(u_2, v_2), (u_3, v_3)]$ and $[(x_1, y_1), (x_2, y_2)]$. The conditions for these two segments to match will be

$$|\theta_3^m - \theta_2^o| \leq T_\theta$$

and

$$T_1 \leq \frac{L_{23}^m / L_{31}^m}{L_{12}^o / L_{23}^o} \leq \frac{1}{T_1}$$

where T_θ and T_1 are the thresholds chosen by the user.

If the segment pairs are compatible in terms of length ratio and the angle between the two segments, the transformation information between these two segments is entered into a segment match record. The record consists of a list of all segments from the two shapes for which the angles computed in Step 3 do not differ by more than 20° and the length ratios in Step 4 do not differ by more than 30%. Fig. 5(a) shows an example of a segment match record.

Each segment will have many possible matches if only these two criteria are used, but there should be fewer cases where several consecutive segments in one shape match consecutive segments in the other shape. Thus, a group match procedure will be used in pass 2.

Pass 2—Group Match

The pass 2 procedure removes those segments for which less than three consecutive segments in one shape match consecutive segments in the other shape. Since these segments are not members of any consecutive segment sequence, they usually represent the extraneous data and should be deleted. Using the segment match record shown in Fig. 5(a) as an example, consecutive compatible segment pairs between shape 1 and shape 2 are segments 25–39 of shape 1, which match segments 13–27 of shape 2. Pass 2 will only retain transformation data corresponding to this match because they are the most likely correct matching pairs [see Fig. 5(b)]. If matched pairs of less than two consecutive segments are located, the algorithm considers that these two shapes belong to different classes, and no further distance calculation will be performed. If more than two sets of

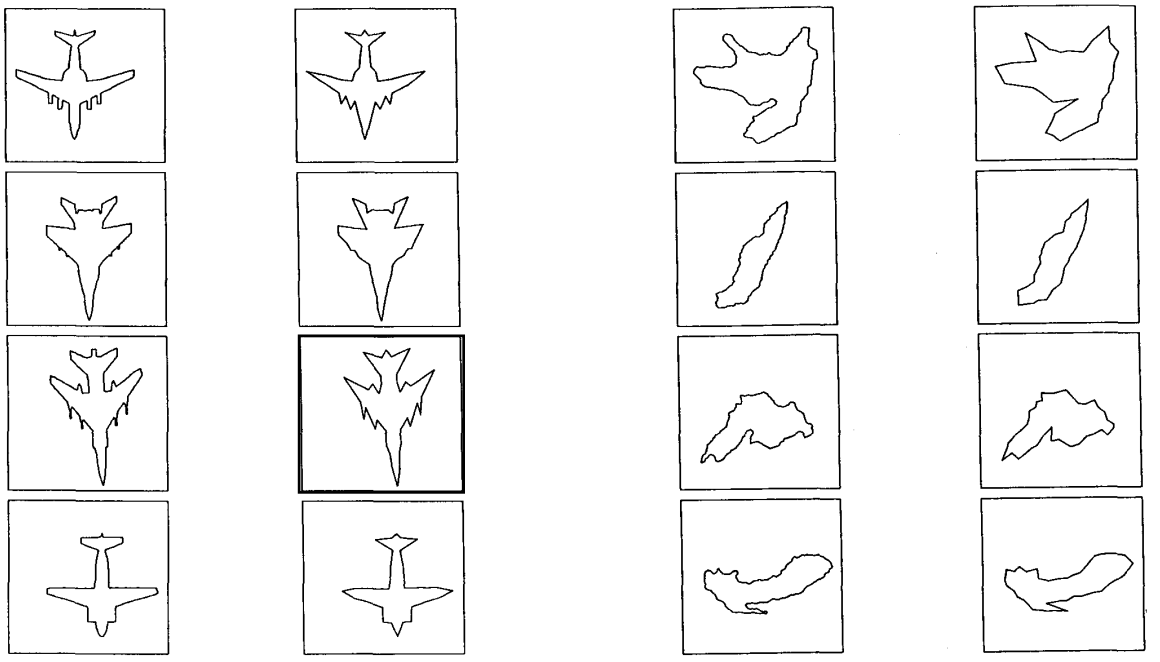


Fig. 3. Template shapes and shapes created by linking the control points.

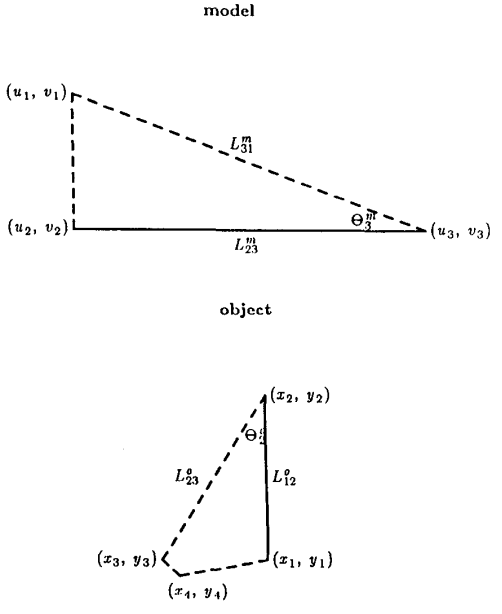


Fig. 4. Example contours for the explanation of segment pair matching.

matched consecutive segments are located between two shapes, the algorithm only records the one with the largest number of matched pairs.

The final task to be accomplished at this pass is to compute the rotational, scaling, and translational average of those segments that match. The distance calculation will use the average of these quad parameters (θ, Δx, Δy, S) to normalize the unknown shape. Examples of partial shapes and their successful boundary matches are shown in Fig. 6.

IV. Distance Calculation

The final step of the algorithm uses the distance transformation and distance measurement techniques to measure the similarity between two shapes. As explained later, the distance transformation converts the boundary pixels of a shape into a gray-level image where all pixels have a value corresponding to the distance to the nearest boundary pixel. Then each boundary point of the second shape is normalized based on the (θ, Δx, Δy, S) calculated from pass 2 of the boundary matching procedure and superimposed on this image. The goodness of the match between the two shapes can be computed from the rms average of the normalized boundary pixels of the second shape on the distance image created from the first shape.

Distance Transformation

Because of the discrete nature of the digital image and the influence of noise on the boundary points, it is an unnecessary waste of effort to compute exact Euclidean distances [7] from the inexact boundary pixels. In most digital image processing applications, it is preferable to use integers to represent distance. Good integer approximations of Euclidean distance can be computed by the process known as chamfer 3/4 distance [1], [3]–[5].

According to [5], the chamfer 3/4 distance can be calculated sequentially by a two-pass algorithm. First, a distance image is created such that each boundary pixel is set to zero and each nonboundary pixel is set to infinity. The forward pass modifies

Shape 1 Segment No.	Shape 2 Segment No.	θ	Δx	Δy	S
3	33	70.765	48.355	-208.494	0.983
4	34	69.139	50.524	-206.060	0.991
4	36	61.044	64.914	-104.339	0.822
5	35	65.171	56.160	-199.759	1.016
6	34	78.111	35.742	-247.470	1.133
6	36	70.017	54.081	-202.739	0.940
7	37	63.435	65.548	-187.958	0.958
8	38	67.663	52.860	-205.003	1.017
9	39	67.834	68.668	-195.132	0.857
10	30	9.074	162.878	-14.652	1.218
10	38	-110.728	-330.344	57.679	0.815
23	30	100.812	86.044	-335.733	1.171
23	38	-18.991	-3.564	154.051	0.784
25	13	70.530	58.724	-204.433	0.946
26	14	69.775	54.276	-207.572	0.988
26	16	78.503	70.323	-233.574	0.830
26	26	-59.170	-169.822	18.177	1.104
27	15	73.960	45.735	-224.620	1.015
28	14	56.101	31.672	-160.797	1.128
28	16	64.832	61.937	-186.782	0.949
28	24	-102.829	-252.661	-86.731	0.840
28	26	-73.142	-233.920	-51.013	1.262
29	17	70.419	57.134	-206.418	0.964
30	8	69.524	-11.487	-147.273	1.021
30	18	67.291	51.030	-208.617	1.024
31	19	67.871	42.386	-223.769	1.085
32	20	62.650	70.414	-156.562	0.837
33	21	55.491	48.310	-170.972	0.996
34	22	72.917	52.602	-218.078	0.994
35	23	71.876	54.704	-210.183	0.968
36	16	-123.832	-207.024	-201.086	1.158
36	24	68.507	49.818	-211.835	1.025
37	25	71.565	55.656	-206.813	0.949
38	26	63.104	63.177	-182.941	0.944
39	27	62.103	54.899	-194.330	1.040

(a)

Shape 1 Segment No.	Shape 2 Segment No.	θ	Δx	Δy	S
25	13	70.530	58.724	-204.433	0.946
26	14	69.775	54.276	-207.572	0.988
27	15	73.960	45.735	-224.620	1.015
28	16	64.832	61.937	-186.782	0.949
29	17	70.419	57.134	-206.418	0.964
30	18	67.291	51.030	-208.617	1.024
31	19	67.871	42.386	-223.769	1.085
32	20	62.650	70.414	-156.562	0.837
33	21	55.491	48.310	-170.972	0.996
34	22	72.917	52.602	-218.078	0.994
35	23	71.876	54.704	-210.183	0.968
36	24	68.507	49.818	-211.835	1.025
37	25	71.565	55.656	-206.813	0.949
38	26	63.104	63.177	-182.941	0.944
39	27	62.103	54.899	-194.330	1.040

(b)

Fig. 5. (a) Example of segment matching record. (b) Example of group matching record.

the distance image as follows:

for $i = 2, \cdots,$ rows do

for $j = 2, \cdots,$ columns do

$$v_{i,j} = \min(v_{i-1,j-1} + 4, v_{i-1,j} + 3, v_{i-1,j+1} + 4, v_{i,j-1} + 3, v_{i,j}).$$

Similarly, the backward pass operates as follows:

for $i =$ rows $- 1, \cdots, 1$ do

for $j =$ columns $- 1, \cdots, 1,$ do

$$v_{i,j} = \min(v_{i,j}, v_{i,j+1} + 3, v_{i+1,j-1} + 4, v_{i+1,j} + 3, v_{i+1,j+1} + 4)$$

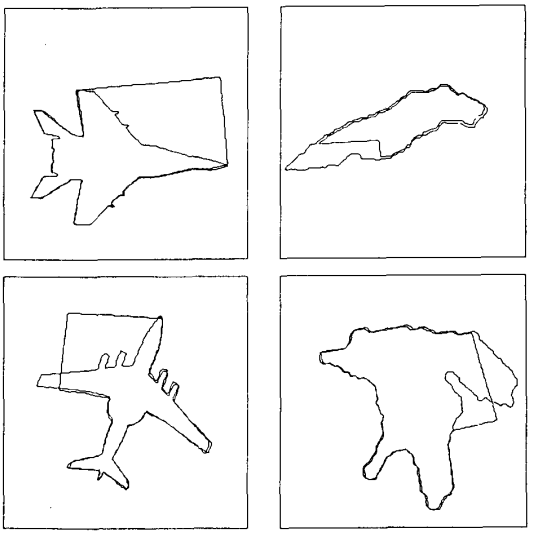


Fig. 6. Examples of successful boundary matching between unknown shape and template shape.

where the $v_{i,j}$ is the value of the pixel in position (i,j). Some sample distance images are shown in Fig. 7.

Distance Measurement

As stated earlier, in order to measure the similarity between two shapes, one shape is converted into a distance image by the chamfer 3/4 distance transformation. The boundary points of the second shape are normalized based on (θ, Δx, Δy, S) calculated from the group matching procedure. These normalized boundary points are superimposed on the distance image created from the first shape as shown in Fig. 8. The rms average [4] of the normalized boundary pixels on the distance image is used to measure the goodness of the match. A perfect match between the two shapes will result in zero distance, as each normalized boundary point from the second shape will then be a zero distance point in the distance image. If the matching is inexact, the average of these values gives a measure of the goodness of the match. For example, a distance of 2 means that each boundary pixel of the second shape is, on the average, located two pixels away from the first shape after being normalized by the transformation data (θ, Δx, Δy, S).

Using every pixel of the boundary to calculate the distance works very well when the object under classification is undistorted or only a small part of it is distorted. When a large part of the object is distorted, the distance measurement scheme cannot serve the purpose of distance calculation. An example of a distorted shape is shown in Fig. 9(a). The segment matching records against two template shapes are listed in Fig. 9(b) and (c), respectively. From these transformation data, the proposed algorithm assumes that both template shapes have the possibility of matching the unknown shape. Fig. 9(d) and (e) show the result of normalizing the boundary of the template shapes and superimposing these points on the unknown shape. Fig. 9(d) shows a good match for boundary points which are not distorted. However, using every pixel of the boundary to measure the similarity yields a distance of 15.153 between these two shapes. Fig. 9(e) shows an incorrect matching between two shapes with

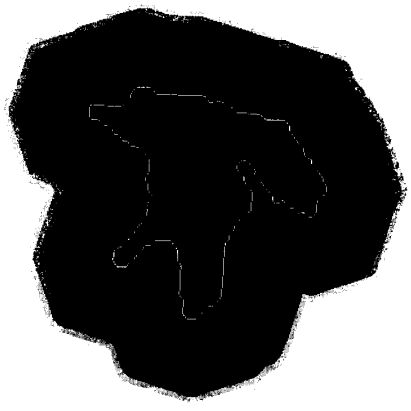
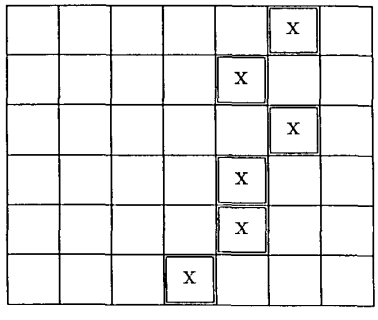


Fig. 7. Examples of chamfer 3/4 distance image. The distances are gray-level coded: the larger the distance, the lighter the tone.



Boundary pixels

13	12	12	11	8	7	4
10	9	9	8	7	4	3
7	6	6	6	4	3	0
4	3	3	3	3	0	3
3	0	0	0	0	3	4
3	0	3	3	3	4	7

Boundary pixels on distance image

Fig. 8. Computation of the distance. The normalized boundary pixel is placed over the distance images. The rms average of the normalized boundary pixels on the distance image is used to measure the goodness of the fit.

a distance of 8.674. Obviously, the unknown object will be classified into a wrong class.

When a human recognizes a partially distorted shape, his judgment relies mainly on the undistorted part. Thus, a distance measure scheme which is based on parts of a shape is proposed to overcome the problem. Because there is no knowledge about how severely the object is distorted or which part or parts of the object is distorted, an approximate procedure is used to find that part of the boundary which corresponds to a match between one half of the boundary points. That is, the boundary of the template

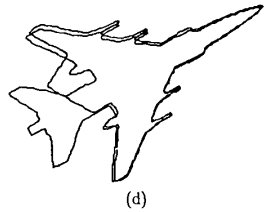
(a)

Shape 1 Segment Number	Shape 2 Segment Number	θ	Δx	Δy	S
3	13	122.618	-77.956	-275.218	0.832
4	14	118.684	-77.931	-282.645	0.944

(b)

Shape 1 Segment Number	Shape 2 Segment Number	θ	Δx	Δy	S
1	13	120.846	-91.561	-269.599	0.959
2	14	118.072	-87.824	-274.294	1.000
3	15	121.369	-100.828	-278.071	1.027
4	16	118.961	-88.785	-274.614	0.999

(c)



(d)

(e)

Fig. 9. Example of misclassified shape. (a) Unknown partial shape. (b),(c) The segment matching records between template shape and unknown shape. (d),(e) The matched boundary between two template shapes and unknown shape.

shape is broken into N equal length segments with an arbitrary starting point. The *local distance* between each segment and the unknown object is defined similarly to the edge distance of [4]:

$$\frac{1}{3}\left[\frac{1}{n}\sum_{i=1}^{n} v_i^2\right]^{1/2}$$

where n is the number of boundary points in the segment and v_i are their superimposed distance values. The average is divided by 3 to compensate for the unit distance 3 in the chamfer 3/4 distance transformation. Then the minimum distance between $N/2$ consecutive segments is considered to be the *partial distance* between two shapes.

The results displayed in Fig. 10 show that partial distance measurements can correctly classify the unknown shape shown in

Segment Number	Local Distance
1	1.06
2	2.95
3	6.36
4	13.30
5	10.90
6	12.62
7	11.58
8	8.49
9	5.20
10	2.57

(a)

Segment Used	Distance
1,2,3,4,5	8.457
2,3,4,5,6	10.126
3,4,5,6,7	11.281
4,5,6,7,8	11.557
5,6,7,8,9	10.192
6,7,8,9,10	8.944
7,8,9,10,1	6.920
8,9,10,1,2	4.763
9,10,1,2,3	3.885
10,1,2,3,4	6.403

(b)

Segment Number	Local Distance
1	2.13
2	2.16
3	2.12
4	5.58
5	17.93
6	34.67
7	28.67
8	2.45
9	1.06
10	1.34

(c)

Segment Used	Distance
1,2,3,4,5	8.857
2,3,4,5,6	17.910
3,4,5,6,7	21.860
4,5,6,7,8	21.867
5,6,7,8,9	21.719
6,7,8,9,10	20.144
7,8,9,10,1	12.912
8,9,10,1,2	1.908
9,10,1,2,3	1.825
10,1,2,3,4	3.051

(d)

Fig. 10. Local distance and partial distance of distance of shapes in Fig. 9.

Fig. 9(a). Both template shapes are broken into ten equal length segments. After using the transformation data from Fig. 9(b) and (c) to normalize the segments, the local distances are listed in Fig. 10(a) and (c). It can be seen that some of these segments match very well with the boundary of the unknown object. Fig. 10(d) shows that the best match for five boundary segments is obtained by using segments [9, 10, 1, 2, 3] with a partial distance of 1.825, which is better than the partial distance of 3.885 from Fig. 10(b) for the misclassified shape.

V. Experimental Results

The algorithm described above was implemented on a sequent S81 computer system using images obtained from a TRAPIX 55/256c image processor system with a VAX 11/750 host computer. Both template data and partial shape data were determined from images of object silhouettes cut out of paper. Images are collected with size 256 × 256 and 8 b gray levels, and then thresholded at 128 to create the object silhouette shapes. For each class of shape, one template image (with arbitrary orientation and location) is collected and stored in the template database. Partial shapes were generated by a method similar to [9]: chopping 0, 10, 20, and 30% of the contour, and replacing the chopped out portion either by a straight line or by a straight line–90° turn–straight line to close the contour. For each class, ten partial shapes, each with different orientation location and size, are collected for each percentage of distorted pixels on the boundary. Different sizes of the partial shapes are obtained by moving the camera closer to and farther from the object. The size of each partial shape varies from 0.8 to 1.2 times the size of the shape in the template database.

Two sets of data were used in the classification experiments to test the effectiveness of the proposed algorithm. The first set consists of four kinds of aircraft. These shapes were selected to study the effects of sharp corners and straight segments on the proposed algorithm. The second data set consists of four classes of shapes which are the aerial views of lakes Erie, Huron, Michigan, and Superior. Each of these lake shapes contains only long curve segments. Fig. 3 shows the images in the template database. Some sample partial shapes are shown in Fig. 11.

It should be noted that the distance measure is asymmetric. That is, the distance calculated by converting one shape into a distance image and superimposing the other shape onto this image is not equal to the distance calculated by interchanging the two shapes. In this investigation, the input unknown shape is converted into a distance image, and the template shape is to be normalized and superimposed on the distance image. This assignment has the advantage that in partial distance measurements, the superimposed shape will be the complete shape. It is easier to get 50% boundary points out of a complete shape, and a better partial distance measurement can be achieved from this assignment.

In the first experiment, the partial distance measurement scheme was used to test the performance of this algorithm. The algorithm has been successfully used to recognize partial shapes of lakes and aircrafts without any misclassification.

The performance of this algorithm on noisy images was also studied in the second experiment. The noisy shapes were generated by a method similar to that in [10]. To create the noisy shapes, for every image in the partial shape database, use a chain code to trace the boundary points. Then for each boundary point, randomly select one of its eight-neighbors, and reverse its value from 0 to 1 or vice versa to create a new noisy image. The algorithm performs very well without any misclassification for lake shapes, and only one shape misclassified for aircraft with 20% distortion.

VI. Conclusions

This paper has presented a new method to recognize a partial shape without knowing its orientation, location, and size. The method uses the control points extracted from the smoothed curvature function to segment the shape. A two-pass boundary matching process is introduced to estimate the transformation data between two shapes. A partial distance measurement scheme is implemented to measure the goodness of the boundary match. Experimental results on shapes with sharp corners and circular arcs show that the method yields results which are satisfactory.

References

[1] H.G. Barrow, J.M. Tenenbaum, and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *Proc. 5th Int. Joint Conf. Artificial Intell.*, 1977, pp. 659–663.

[2] R.C. Bolles and R.A. Cain, "Recognizing and locating partial visible objects: The local-feature-focus method," in *Robot Vision*, A. Pugh, Ed., 1984.

[3] G. Borgefors, "Distance transformations in digital images," *Comput. Vision, Graphics, Image Processing*, vol. 34, pp. 344–371.

[4] ——, "An improved version of the chamfer matching algorithm," in *Proc. Int. Conf. Pattern Recognition*, 1984.

[5] ——, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 849–865, Nov. 1988.

[6] J.F. Canny, "Finding lines and edges in images," M.S. thesis, Massachusetts Inst. Technol., Cambridge, 1983.

[7] P.E. Daniellson, "Euclidean distance mapping," *Comput. Vision, Graphics, Image Processing*, vol. 14, pp. 227–248, 1980.

[8] J.W. Gorman, O.R. Mitchell, and F.P. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 257–266, Mar. 1988.

[9] T.A. Gorgan, "Shape recognition and description: A comparative study," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1983.

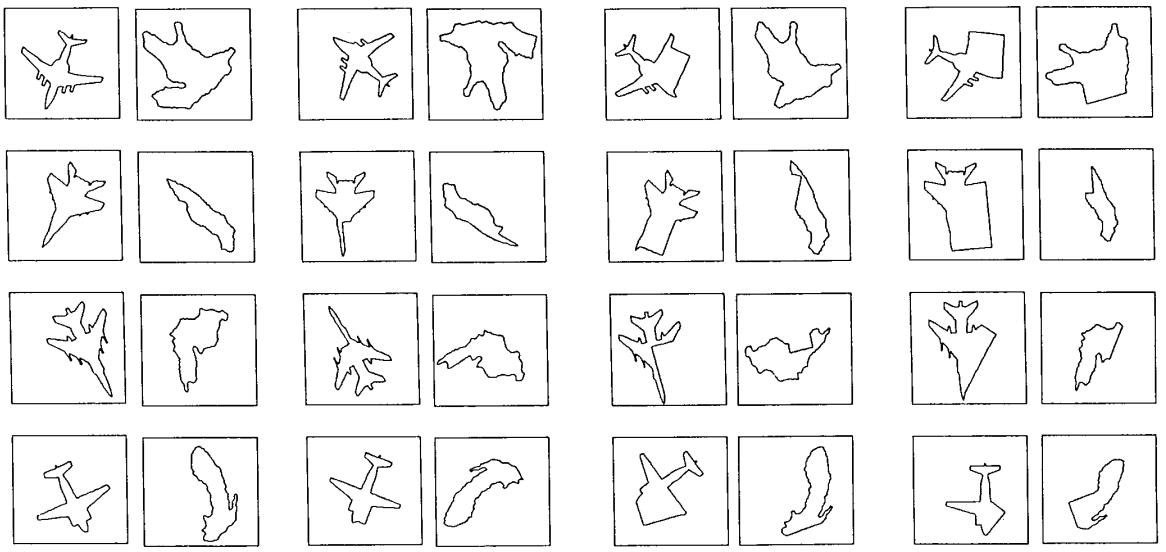
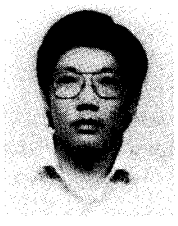


Fig. 11. Typical partial shapes with (a) 0%, (b) 10%, (c) 20%, and (d) 30% of the boundary distorted.

[10] L. Gupta, "Contour transformations for shape classification," Ph.D. dissertation, Southern Methodist Univ., Dallas, TX, 1986.
[11] T. F. Knoll and R. C. Jain, "Recognizing partially visible objects using feature indexed hypotheses," *IEEE J. Robot. Automation,* vol. RA-2, pp. 3–13, Mar. 1986.
[12] C. C. Lin and R. Chellappa, "Classification of partial 2-D shapes using Fourier descriptors," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-9, pp. 686–689, Sept. 1987.
[13] O. R. Mitchell and T. A. Gorgan, "Global and partial shapes discrimination for computer vision," *Opt. Eng.,* vol. 23, no. 5, pp. 484–491, 1984.
[14] W. A. Perkins, "A model-based vision system for industrial parts," *IEEE Trans. Comput.,* vol. C-27, pp. 126–143, Feb. 1978.
[15] K. E. Price, "Matching closed contours," in *Proc. IEEE 1984 Workshop Comput. Vision,* 1984, pp. 130–134.
[16] P. F. Singer and R. Chellappa, "Machine perception of partially specified planar shapes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition,* 1985.
[17] Y. J. Tejwani and R. A. Jones, "Machine recognition of partial shapes using feature vectors," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-12, pp. 504–516, July/Aug. 1985.



Hong-Chih Liu received the B.S. degree in electronic engineering from Chung-Yuan University, Chung-Li, Taiwan, Republic of China, in 1973, the M.S. degree in electrical engineering from Washington University, St. Louis, MO, in 1981, and the Ph.D. degree in electrical engineering from Southern Methodist University, Dallas, TX, in 1989.

He is currently a Research Assistant at the Institute of Nuclear Energy Research, Lung-Tan, Taiwan, Republic of China.

Mandyam D. Srinath (M'59–SM'74) received the B.Sc. degree from the University of Mysore, India, in 1954, the diploma in electrical technology from the Indian Institute of Science, Bangalore, in 1957, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1959 and 1962, respectively.

He has been a faculty member of the Department of Electrical Engineering, University of Kansas, Lawrence, and the Indian Institute of Science, Bangalore. He is currently a Professor of Electrical Engineering at Southern Methodist University, Dallas, TX, where he has been since 1967. He has published numerous papers in signal processing, control and estimation theory, and image processing, and is the coauthor of the books, *Introduction to Statistical Signal Processing with Applications* (Wiley-Interscience, 1979) and *Continuous and Discrete Time Signals and Systems* (Prentice-Hall, 1990).