Journal of Imaging Science and Technology® 55(6): 060501-1–060501-13, 2011.

Robust Partial Shape Recognition Using Curvature and an Iterative Closest Point Algorithm

Samuel H. Chang, Duk-Sun Shim, and Cheol-Kwan Yang
School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 156-756, Korea
E-mail: dshim@cau.ac.kr

Abstract. *This article presents a robust translation, rotation, and scaling invariant pattern recognition algorithm using partially distorted two-dimensional (2D) contours. Both global and local matching accuracies are guaranteed: the former is achieved by using global control point (GCP) and curvature matching and the latter by using the iterative closest point (ICP) algorithm. In the proposed algorithm, an input contour is considered two possible types: first the contour contains GCP and the second contains no GCP. For the contours with no detected GCPs, the algorithm considers the contour as the second type of contour and the curvature matching algorithm is directly applied. For the contour with GCPs, a global alignment is performed using the GCPs and then, the curvature matching algorithm is applied. The validity of the algorithm is illustrated by the presentation of experimental results. The experiments show that the algorithm works well for partial object recognition with and without GCPs. © 2011 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2011.55.6.060501]

INTRODUCTION

The purpose of partial shape recognition is to identify an object represented by a set of partial shapes. Partial shape recognition deals with the representations and transformations of the shapes, and the measurement of the similarity between two shapes. Measurement of similarity between two shapes can be done by shape matching that has been approached in many different ways, including the Hough transform,[1] pose clustering,[2] tree pruning,[3] geometric hashing,[4] statistics,[5] alignment methods,[6] deformable templates,[7] relaxation labeling,[8] Fourier descriptors,[9] wavelet transforms,[10] neural networks,[11] shape contexts,[12] and curvature scale space (CSS).[13,14] In shape representation, while other representations such as the Hough transform,[15–17] shape factors and quantitative measurements,[18] submatrix matching,[19] chain encoding,[20,21] polygonal approximations,[22] and split-and-merge[23] have been used, curvature representation is one of the most important parameters[24] be used to represent shapes.[25–32] In shape matching using curvature presentation, some higher curvature points are defined as control points. It is very common to extract the boundary features from the digitized image of an object and then determine control points and use them to match the object boundary with the model boundary patterns.[25] Most of the computations of the control points are done by extracting the local maximal and minima from the smoothed curvature functions using different techniques. However, high curvature points along a boundary can be very noisy. Smoothing processes such as the Gaussian smoothing techniques have been used to reduce high frequency noises constitutes two major obstacles to obtain a good shape matching results: first it may erase a control point or shift the locations of the control points[33] and second it may shift the matching results. For object shapes containing features such as line segments and angles (e.g., some human made objects such as industrial parts), some high curvature points can be erased during the smoothing process.

One of the possible solutions to this problem of location and matching shifting is to select the control points along a boundary before the smoothing process is applied. In our algorithm, we select the control points before extracting the contours from the images and then use the control points to decompose a boundary into small curve segments. These small curve segments are then smoothed by a path-based Gaussian filter. Additionally, control points are usually used to guide the boundary matching procedure and estimate the translation, rotation and scaling factors.[25,34] Most recognition schemes based on local features[35–37] use only the dominant points or ratio of the lengths and angle between the lines joining the dominant points as features. These local features do not give a unique representation and may fail in global matching.

In our algorithm, the recognition schemes are based on both local and global features. The translation, rotation, and scaling factors are estimated in the curvature matching domain and verified by final boundary matching. The algorithm first detects the corner features in the input image. These corners located on the boundary of an object with some global conditions are first detected. Some of them are then selected as the global control points (GCPs) to decompose the boundary into small curve segments if certain conditions are satisfied. A path-based Gaussian filter is then used to evolve the smaller curve segments. The improved angular-based curvature estimation algorithm is then used to calculate the curvature function from the boundary.

Received Dec. 6, 2010; accepted for publication Sep. 6, 2011; published online Dec. 19, 2011
1062-3701/2011/55(6)/060501/13/$20.00.

Finally, the matching between the partial shapes and models is completed in two steps: representation matching and boundary matching. In representation matching, the model curvature patterns are matched with the object curvature representations. Two transformations are used to complete the matching process: (1) a shifting transformation is used to transform the model curvature pattern into a new initial condition, so that the model can match with the object having a different orientations, and (2) a scaling transformation is used to transform the model curvature pattern to a new curvature function so that the new model can be rescaled. The new model is then matched with the objects having different sizes. Two levels of matching are performed in curvature space: (1) the lower level local matching that searches for an object containing a single partial shape within a contour and (2) the higher level global shape matching that recognizes an object containing multiple partial shapes from multiple contours. The matching process produces a set of transformation parameters and a numerical matching error. While the matching error is used to evaluate the similarity of the curvature representations, the parameters are used to rescale the new model for boundary matching. In boundary matching, distance measurement is used to measure the similarity between the shape of the object and that of the rescaled new model. The 2D iterative closest point (ICP) algorithm is used to verify if the matching is valid and to improve the boundary matching by optimizing the transformation between the two boundaries.

GLOBAL CONTROL POINT DETECTION

A number of factors in the curvature estimation process affect the matching results in different ways. These include the noise from contour tracking, the position changes of the high curvature points and the accuracy of the curvature estimation. While subpixel edge detection produces more accurate locations of the edge points and smoother edges, edge detection is a pixel base operator providing discrete and noisy boundary data from digitized images. However, curvature estimation algorithms are designed based on continuous functions with second derivatives of the contour function. While continuous shape functions may not be available for most shapes, an approximation of the continuous function can be formulated from the discrete boundary data obtained from the edge detector by some kind of smoothing process. Contour smoothing processes such as the Gaussian kernel[38] or other methods are necessary and have been widely used to smooth a contour before any curvature estimation methods are applied.

However, when a shape contains features such as line segments and if an angle between two intersect line segment. The angle performs important roles in curvature matching. However, smoothing process may change the angle between two line segments. In this case, changing the angle results in the change of the curvature value and may cause mismatch. In an example of partial circle and partial triangle recognition from the Kanizsa image in this article, a triangle shape contains three straight sides and three angles with a sum of 2π. During the matching process, the sum of the curvature of a triangle model is 2π. A pattern of three GCPs from an object shape for which the sum of the curvature values is 2π is searched for during the matching process. However, the matching process may fail, because the curvature value of a corner is largely reduced during the smoothing process. Therefore, the GCPs should be detected and recorded before the Gaussian smoothing function is applied. In our algorithm, as contrary to conventional approach, GCP detection is performed before the Gaussian smoothing is applied. Therefore, GCP detection may be very noisy without Gaussian smoothing. In this case, some "globosity" condition is added to define a GCP. A number of algorithms have been developed to detect high curvature points. These include contour based methods, such as multiscale detection,[39] Laplacian-of-Gaussian zero crossing contours,[40] and constrained regularization.[41] However, most of these methods are based on the application of the curvature estimation after the smoothing process. In this case, the high curvature points are found from the positions with local maximum and minimum of the smoothed curvature function.

A corner can also be found from a triangle with a specified opening angle and size over the curve.[42] In our algorithm, a high curvature point is defined as a corner with high curvature value with additional global conditions of its neighborhood that holds a set of angles that converge to a constant in its neighborhood. The angles are defined by a concentric circle representation.[43] The radius of the concentric circle is similar to a Gaussian kernel and the selection of the angle based on the radius is similar to a smoothing process. A threshold angular value is selected based on the experiment requirement of ensuring a higher curvature. A GCP is found if the angles over its concentric circles converge to a constant and this constant is smaller than the threshold angular value. The convergent of the angle within the concentric circle is then selected as a GCP. The procedure used for finding a GCP is illustrated in Figure 1. Around corner p_0 on the curve trace, a 6-neighborhood chain that encodes the trace of the curves is selected. A set of angles $\Psi = \{\psi_i : i = 1, 6\}$ corresponds to the angles between the vectors centered at p_0. By fixing one edge point R on the curve trace from one side, a vector initiated from p_0 to R is defined. By drawing circles centered at p_0 with incrementally increasing radius, these circles intersect with the curve traced from the other side of p_0. The intersections $q_i (i = 1, 6)$ form a set of vectors initiated from p_0.

Given an input image with an object, the procedure used to determine a GCP in our algorithm consists of the following steps:

(1) A set of corners H_i, $i = 1, m$ is found using the Harris corner detector.
(2) A set of contour points P_i, $i = 1, n$ is extracted from Canny[44] edge points.
(3) A corner point p_0 from H is selected if the distance between p_0 and any point of P is less than or equal

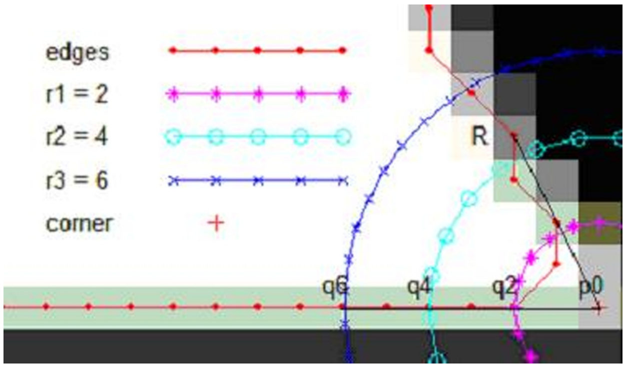

Figure 1. Concentric circle representation showing that the corner p_0 represents a global control point with one side of its curve trace being a straight edge. The straight edge is determined by a set of angles, where the example $\angle RP_0q_i$ $(i = 2, 4, 6)$ in the figure converges to a constant.

to a threshold value. The threshold value here indicates the distance between the GCP and the contour it relates to. It can be selected larger than the maximal of estimate noise from the contour.

(4) Find a set of angles $\Psi = \{\psi_i : i = 1, k\}$ by setting an edge point R on one side of the corner and select a set of intersections $q_i; (i = 1, k)$, as described in Fig. 1, where k is an experimental number depending on the size of the object.

(5) If Ψ converges to a constant, p_0 is recorded. Otherwise, repeat (4) by setting any point of $q_i (i = 1, k)$ to R.

Curvature Pattern Matching and Model Boundary Rescaling

In this section, we explain how curvature functions between a model and an object are matched, and how a model boundary is rescaled using the matching results. One of the objectives of shape recognition is to find a 2D coordinate transformation, consisting of translation, rotation, and scaling, that maps the shape of the model to that of the object with the minimum distance error. Usually, a score based on the least squares error of the mapping is used to measure the overall quality of the matching. In curvature space, the scaling is a function of the length of the arc in the discrete curvature estimation, and the rotation is a function of the shift of the curvature. To search for a pair of matched curvature segments between the object and the model with different orientations, shifting the model curvatures and matching them with the object curvatures produce different curvature matching errors. The minimum error reflects the best match. Its correspondence can be used to transform the whole model curvatures. The same procedure can be applied to match an object with models of different sizes. For the matching between an object and models with different orientations and sizes, the model curvatures will be shifted and scaled in order to find the best match. These shifting and scaling operations are explained by the following two examples.

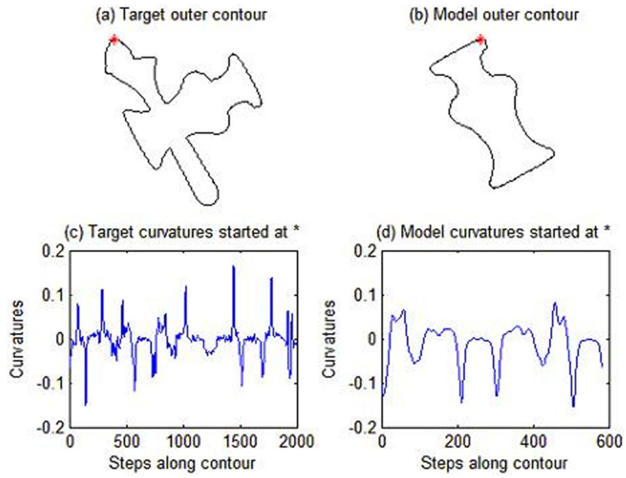

Figure 2. Curvatures estimated from target and model contours.

Figure 2 shows the curvatures of two contours from Kimia's data set. The first is the outer contour of a vase model, as shown in Fig. 2(b), and the second is the outer contours of the partial shapes of the vase and a monkey wrench shown in Fig. 2(a). The orientation of the model's outer contour was rotated in order to show the matching between a partial object and a model with a different orientation. Their corresponding curvatures are shown in Figs. 2(c) and 2(d), respectively.

In the curvature matching process, the model curvature function in Fig. 2(d) is compared with the object curvature function in Fig. 2(c). An error vector with the same length as the model curvature can be obtained each time. By shifting the model curvatures one step and repeating the comparison with the object curvatures, a set of error vectors can be calculated to form an error matrix. The error matrix contains all possible curvature matching results. Therefore, the best matching can be selected from the error matrix.

Let $C_T(i), (i = 1, m)$ be the curvatures of the target contour and $C_M(i), (i = 1, n)$ be those of the model contour, where $m > n$. Let vector $E_{(i)}$ be the matching error at time i. Then, $E_{(i)} = [c_T(i) - C_M(i)]^2$. An error matrix $E_{(i,j)}$ can be formed for all possible curvature matching

$$E_{(i,j)} = [C_{ET}(i + j) - C_M(i)]^2, \tag{1}$$

where $i = 1, n$, $j = 1, m - 1$. Curvature function C_{ET} is an extension of C_T with a size of $m + n - 1$ to fulfill the shift matching. The first part of C_{ET} is C_T

$$C_{ET}(1 : m) = C_T. \tag{2}$$

The second part of C_{ET} is

$$C_{ET}(m + 1 : m + n - 1) = C_T(1 : n - 1). \tag{3}$$

Each column of the error matrix contains a single error between two curvatures and each row contains the errors

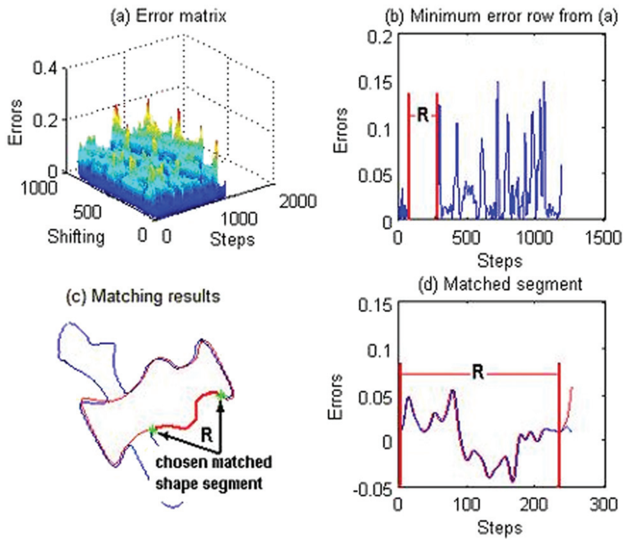

Figure 3. The curvature matching error matrix (a) and the rescaled model compared with the target contour (c) show reasonable curvature matching results with some boundary distance error.

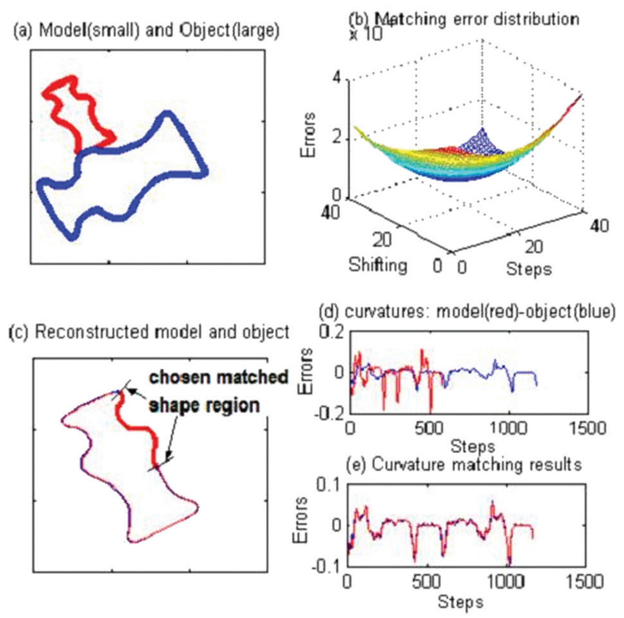

Figure 4. Curvature matching results between an object with a different orientation and scales from its model (c). The contours of the object and model (a) show the size of the object is much larger than that of its model.

from a shift matching. The plot of the error matrix generated during the curvature matching in this example is shown in Figure 3(a).

The objective of curvature matching is to choose the best matched pair of curvature segments between the model and the object. The best matched pair of curvature segments has the minimum error and maximum length of the segment. By setting a minimum curvature matching threshold value, the length of a matched pair of curvature segments can be calculated. The matching threshold value can be estimated from the curvature estimate noise. In this example, a value of 0.002 was calculated from curvature

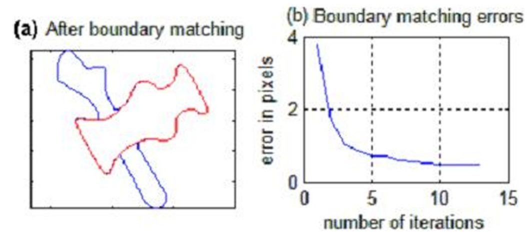

Figure 5. The boundary matching results show that the improvement of the boundary position reduces the matching error by 87% as compared to the curvature matching results.

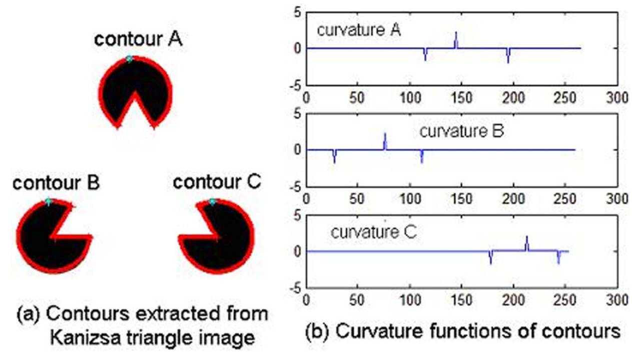

Figure 6. Contours extracted from Kanizsa triangle and their corresponding curvature functions.

estimation. The best matched pair of curvature segments of “R” shown in Fig. 3(b) reached a maximum length of 204 points in the error matrix, as shown in Fig. 3(a). Its corresponding matched pair of contour segments is shown as R in Fig. 3(d). Two curvature segments are overlapped in the region marked by R. The transformation of the match, including the shifting position of 766 in the object curvatures and the starting position of the contour segment in the model curvatures, 39, is used as the best transformation parameters. By taking positions 766 and 767 of the object contour point, with a starting curvature of 39 from the model curvature function, the rescaled model contour is plotted with a red color over the object contour in Fig. 3(c). The chosen matched shape segment is plotted with a red color marked by R. It can be seen that the rescaled model contour overlaps the partial shape of the vase contour with some boundary errors. These boundary errors will be corrected later in the ICP matching algorithm, as discussed in the section on Boundary Matching. By setting a boundary matching threshold error, it can be seen that the partial shape object contour is identified by the vase model.

For a target shape with a different scale from the model shapes, a scaling matching procedure is added to search for a suitable scale. First, a scale factor is selected. For each scale factor, a matching error matrix is formed and the local best matched pair of curvature segments corresponding to the minimum error is calculated. Equations (1) then become

$$E_{(i,j)} = [C_{ET}(ki + j) - C_{M(i)}(i)]^2, \qquad (4)$$

where k is the scale factor.

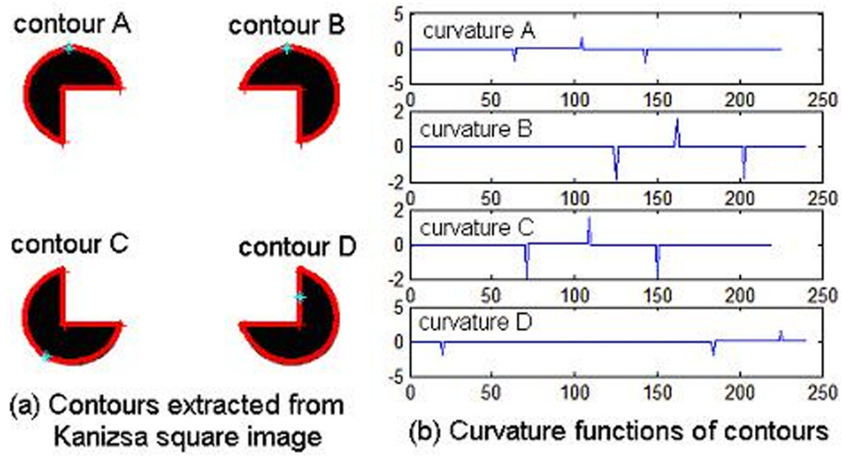

Figure 7. Contours extracted from Kanizsa rectangle and their corresponding curvature functions.

A set of best matched pairs of curvature segments is found corresponding to the different scale values. The global best matched pair of curvature segments is then found from the local best matched pairs of curvature segments. Finally, the best matched pair of segments from the neighborhood is searched for to ensure the optimal scale value and shifting position.

Figure 4 shows the matching results of an object contour obtained from its smaller model contour with different orientations. In this example, the contour of the vase model in Fig. 2 is used as the model contour and its enlarged, and rotated contour is used as the object contour. The local search is shown in Fig. 2(b). The corresponding pair of curvature segments is plotted, as shown in Fig. 2(e), and the rescaled vase model is plotted over the object, as shown in Fig. 2(c).

After the determination of the curvatures of a model that matches with an object contour in the scene, the model is rescaled using the parameters obtained from curvature matching, so that it has a similar size and orientation to the object contour. The location information is recorded as the initial position of a curvature function.

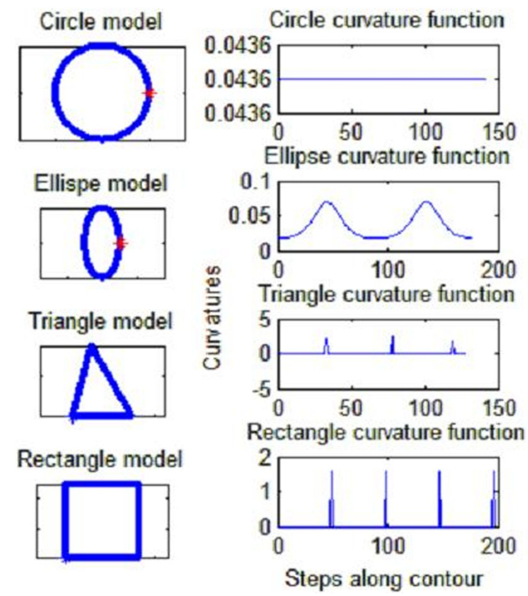

Figure 8. Primitive models and their curvature functions used in partial shape recognition from Kanizsa images.

BOUNDARY MATCHING

Verification and Improvement of Curvature Matching

Boundary similarity is used in most shape recognition applications. In our algorithm, it is used first to verify the curvature matching results and second to reduce the matching errors. The curvature representation is invariant to location and orientation and unique. However, compared with boundary similarity, curvature similarity is far too sensitive and unstable. A curvature function provides a simple mechanism for contour matching with different orientations and different scales. Matching errors in the curvature domain may lead to inaccurate shape recognition results. Furthermore, the process of curvature estimation and model rescaling causes considerable geometrical errors. This may cause large errors in the distance measurement. Therefore, in our algorithm, after curvature function matching, the model is rescaled using the initial positions, and curvature shifting distance and scale factors are provided by the curvature matching results. The rescaled boundary has a similar size and orientation to the object contour and allows the performance of the similarity measurement to be improved.

The similarity between two boundaries can be measured by different distance measurement techniques. The distance transformation converts the boundary points of a shape into the nearest boundary points of the other shape.[25] In fact, the problem can be solved based on the minimization of the sum of squares of distances between corresponding points of the two boundaries. This problem has already been solved using the ICP algorithm. The ICP algorithm was first proposed by Besl and McKay[45] and Chen and Medioni[46] and improved later by Zhang.[47] It has been shown to be a robust method of registering multiple alignments.

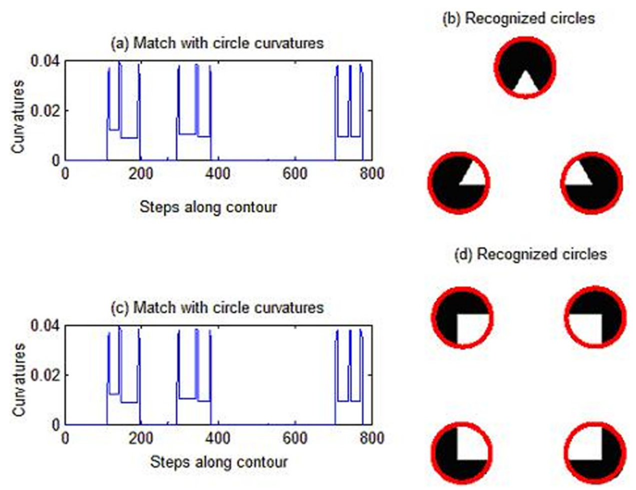

Figure 9. The partial circle curvature functions were matched with the Kanizsa triangle contours (a) and rectangle contours (c). The circle models were then rescaled using the matching results and plotted over the input Kanizsa triangle (b) and rectangle (d).

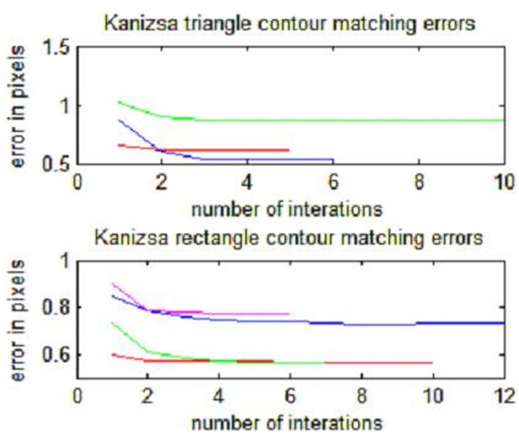

Figure 10. Circle boundary matching errors from Kanizsa triangle and rectangle.

In our algorithm, the boundary matching is done in two steps:

(1) A set of control points from a model is selected in two ways: (a) the control points should cover the whole contours; and (b) those points corresponding to a higher curvature value should be selected.
(2) The unmatched points of the object boundary are removed and, therefore, the ICP algorithm can achieve better performance and a higher processing speed.[47]

The implementation of the ICP matching algorithm is explained by the following example. The control points were selected by two steps: first, all points with a curvature value large than the total average curvature are selected. Second, for the rest of the unselected points, one from every ten points along the contour was selected. The curvature matching result in Fig. 3(c) is used in this example. Figure 5(b) shows that the initial average of the (squared distance errors?) between the corresponding points is 3.7987 pixels. This error is reduced to 0.8746 pixels at iteration 13.

There are two problems in using the ICP algorithm. The first is the initial estimate of the alignment and the second is the slow convergence due to the shallow error landscapes and the tendency to fall into local minima. In this application, the curvature matching solves the problem of the initial alignment. In order to solve the second problem, the unwanted points are removed before the ICP algorithm is applied. This is done in two steps. First, only local points corresponding to matched curvature points are selected for the ICP matching. Second, only those target points close to the model points are selected for the second matching.

EXPERIMENTS

In this section, we compare the proposed approach with the existing techniques in order to demonstrate its significant improvement over previous works and examine the performance of the proposed algorithm with two types of recognition problems:

(1) The recognition of the object shape with the GCPs. The capability of the proposed algorithm for contour extraction, curvature estimation, and curvature matching is tested using the GCPs.
(2) The recognition problem for a model with a different orientation, scale, and contours. The capability of the proposed algorithm in curvature matching with no identical shapes is tested.

Experiments with Partial Shape with Critical Global Point Features

In this experiment, the images containing a Kanizsa triangle and a Kanizsa rectangle are used. The Kanizsa triangle is a typical class of illusory contours and has been discussed in the field of contour recognition and completion in computer vision and human vision perception. There are three partial circles in the Kanizsa triangle image and four in the Kanizsa rectangle image. Each partial circle contains two partial shapes. One is a local partial shape and the other is a global one.

This experiment was implemented using MATLAB codes with a 1.73 GHz Pentium® M processor. There are 804 points extracted along the shapes of four models and 972 points from the shapes of two images. The total execution time for the recognition of both the Kanizsa triangle and Kanizsa rectangle images is 5.6381 s. While the edge extraction and curve smoothing process take 4.4063 s, or 78% of the total execution time, the curvature estimation, matching, and boundary adjustment take only 1.2318 s, or 22% of the total executive time.

In the experiment, the procedure described in the Introduction was used to identify the GCPs. There are 13 candidate GCPs found from the input Kanizsa triangle

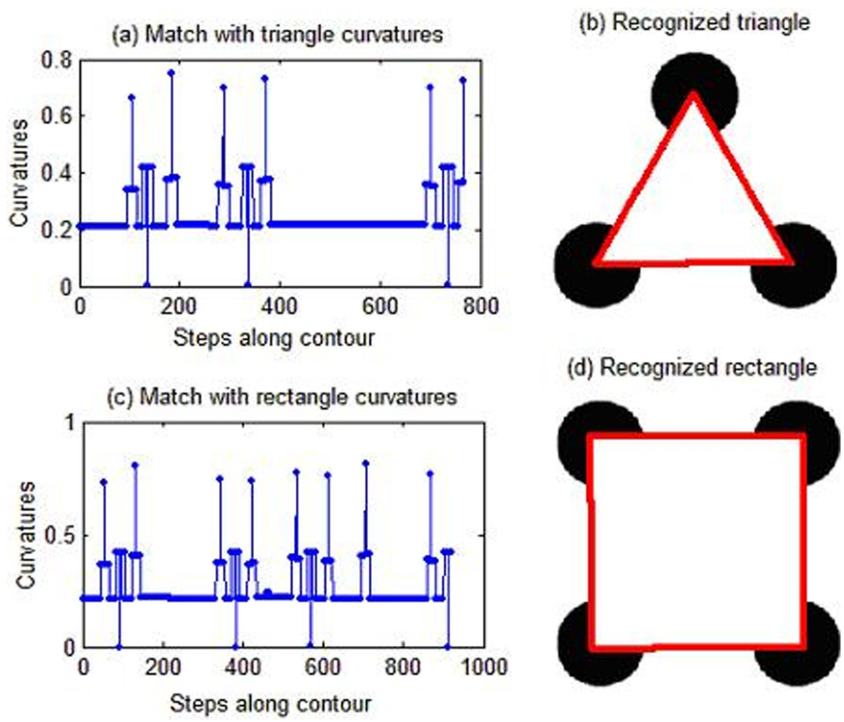

Figure 11. The rescaled triangle and rectangle models are matched using the ICP algorithm and plotted over the Kanizsa triangle image (b) and Kanizsa rectangle image (d). The related curvature matching results are shown in (a) and (c).

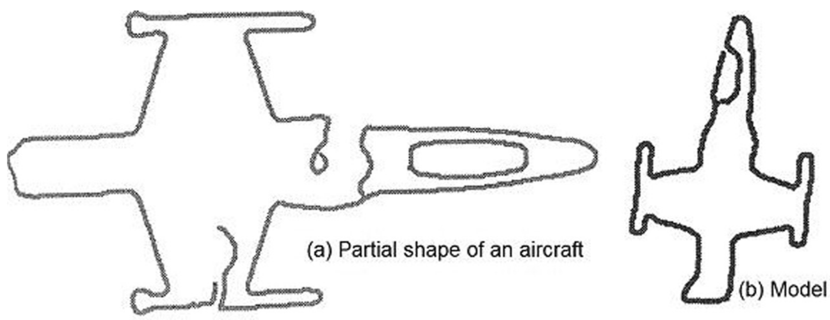

Figure 12. Partial shape of contours of an aircraft target from image and contours of an aircraft model.[29]

image using the Harris corner detector and three edge contours extracted from the images using the Canny edge detector. By checking the closest distance between the 13 candidate GCPs and three contours, and checking the straight edge condition using a concentric circle representation, nine GCPs are selected. There are three GCPs belonging to each contour. Each contour is then segmented into three smaller segments. The Gaussian smoothing method is then applied to each segment. The smoothed contours are plotted over the Kanizsa triangle image, as shown in Figure 6(a).

The curvature estimation method described in the Introduction was used to estimate the contour curvature functions shown in Fig. 6(b). The same procedure was applied to the Kanizsa rectangle image, and the smoothed contours are shown in Figure 7(a). Four curvature functions are shown in Fig. 7(b). From the curvature functions shown in Fig. 6(b) and Figure 8, one can see that the curvature values reflect the positions of the GCPs.

We assume that the Kanizsa images can contain any of the four primitive models: a circle, an ellipse, a triangle, and a rectangle. By using some initial conditions, a set of four models was constructed. Their curvature functions were estimated. Both the plots of the models and their curvature functions are shown in Fig. 8.

The curvature matching method described in the section on Curvature Pattern Matching was used to match the model curvature functions obtained from the curvature functions of the Kanizsa triangle and rectangle contours. Because the curvatures of a circle form a constant, as shown in Fig. 8, the curvature matching is simply done with a scaling matching procedure. The matching results obtained

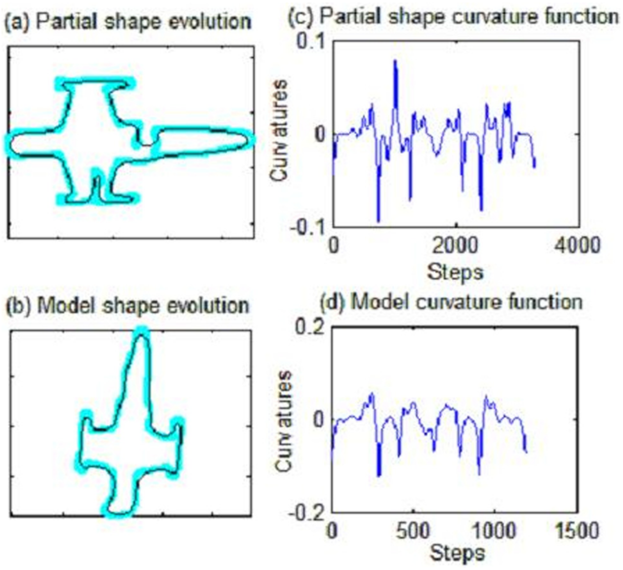

Figure 13. Contour evolution of partial shape of the aircraft target (a) and model (b) from Fig. 12 and their corresponding curvature functions (c) and (d).

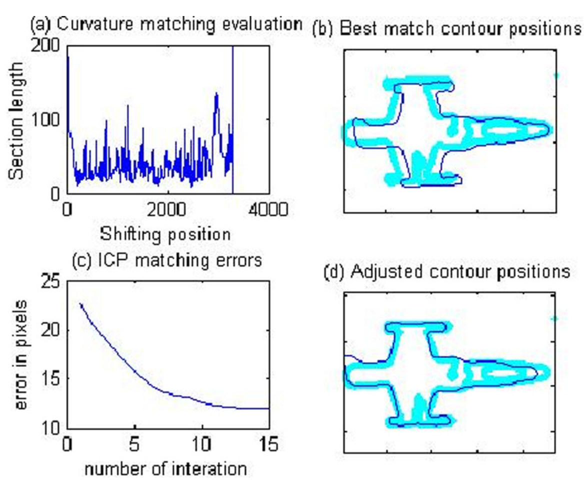

Figure 14. The curvature matching (a) results in reasonable model boundary reconstruction (b), and the ICP matching results (c) show a better model boundary compared with the target contour (d).

with the curvature function of the circle model are shown in Figures 9(a) and 9(c).

During the matching between the curvatures of the circle mode and the curvatures of the object in the Kanizsa triangle image, the matching errors of the three pairs of curvature segments are close to zero, as shown in Fig. 9(a). The three best matches indicate that three circle curvature patterns are found from the object. The scaling parameters and the corresponding initial positions are determined and then used to rescale the circle model. The three rescaled circles obtained from the circle model match the partial shape of the circle objects well with minor location errors. These errors were later corrected using the 2D ICP algorithm. The four circles in the Kanizsa rectangle image were rescaled and corrected using a similar process. The corrected circles are plotted over the Kanizsa images, as shown in Figs. 9(b) and 9(d).

The 2D ICP algorithm was used to improve the positions of the rescaled circles. The matching errors with respect to the three partial circle shapes are plotted in Figure 10. Both corrections show that the boundary distance errors are smaller than one pixel.

The matching between the curvature functions of the Kanizsa contours and ellipse model, triangle model, and rectangle model resulted in no best matched pair of curvature segments. However, the matching with the triangle model and rectangle model provided the three best matched pairs of point curvatures shown in Fig. 11(a) and the four best matched pairs of point curvatures shown in Fig. 11(c). Because of the zero curvatures of the sides of the triangle and the rectangle models, the three best matched pairs of curvature points form a triangle model if the sum of the three points is close to 2π. By selecting the corresponding positions from the positions of the best matched curvature points, a rescaled triangle and rectangle were reconstructed. The ICP algorithm is then used to correct the contours that are plotted over the images of the Kanizsa triangle and rectangle, as shown in Figs. 11(b) and 11(d), respectively.

Experiments with Partial Shape with Differences Contour from Model

In this experiment, the partial shape of an aircraft target from an image and the contours of an aircraft model,[29] as shown in Figure 12, are used. The partial shapes of the target are slightly different from the contours of the model, as can be seen in the figure. The size of the target is larger than that of the model and their orientations are different.

MATLAB codes and the same processor were used to run this experiment. There are 1190 points extracted along the model shape and 2839 points extracted from the target shape. The total execution time for the recognition of the aircraft images is 14.1403 s. While the curve smoothing takes 12.1574 s, or 86% of the total execution time, the curvature estimation, matching, and boundary adjustment take only 1.9829 s, or 14% of the total execution time. For the matching process, the curvature matching takes 0.5408 s and the ICP matching takes 1.1316 s.

The contours of the model and object can be obtained from the figure. It is obvious that there are no line segment intersects at a point forming an angle; therefore, there is no global control point with the definition in this article over the contours by inspection. The global control point detection algorithm discussed in the Introduction, therefore, not used in this experiment. Instead, a Gaussian smoothing function with a larger width is directly applied to the contours. The smoothed contours are plotted over the input contours in Figures 13(a) and 13(b). Their corresponding curvature functions are estimated and plotted in Figs. 13(c) and 13(d), respectively.

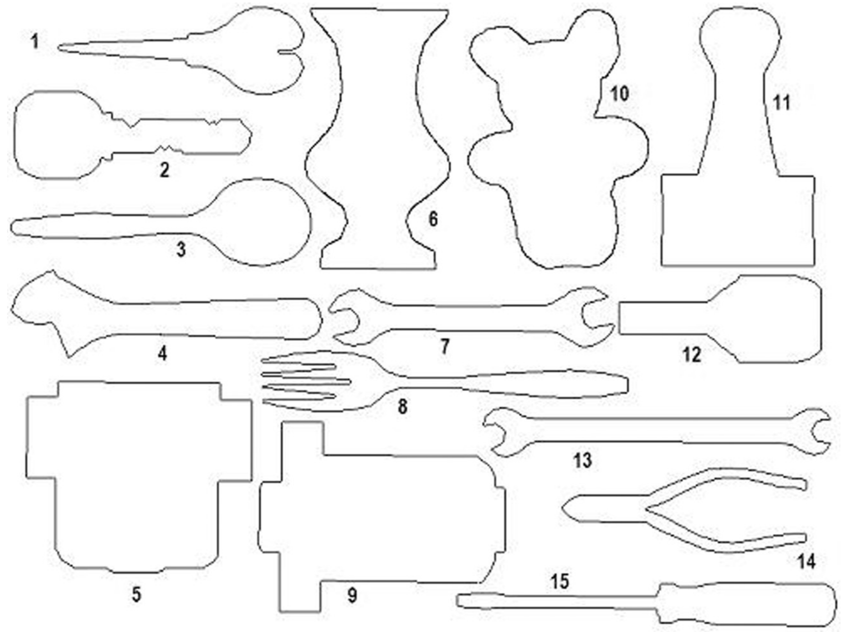

Figure 15. Model objects[29] used to test the algorithm: 1: scissors; 2: key; 3: spoon; 4: monkey wrench; 5: connector case; 6: vase; 7: wrench 1; 8: fork; 9: connector case 2; 10: panda; 11: clip; 12: bottle; 13: wrench 2; 14: cutter; 15: screwdriver.

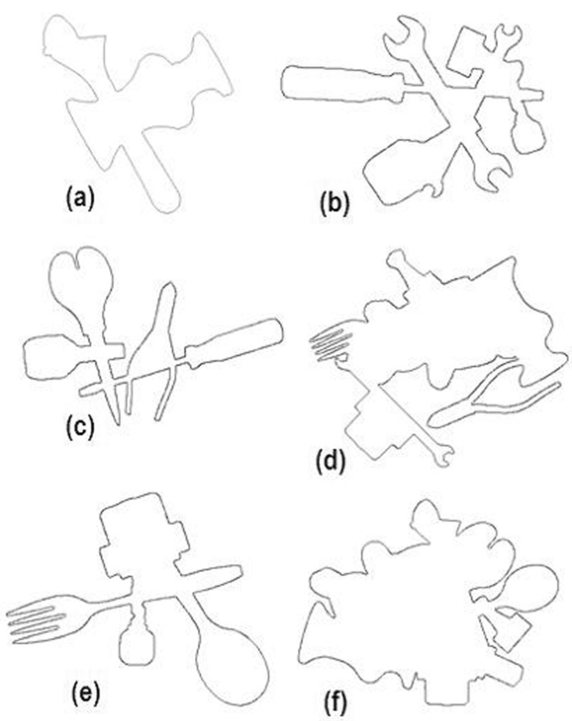

Figure 16. Outermost contours of object sets from images.[29]

The matching method discussed in the section on Curvature Pattern Matching is used to match the curvature functions of the model and object. Both the scaling matching and shifting matching procedures were performed. Figure 14(a) shows the matched region selected from the shifting matching results following the scaling matching. The best match is a segment with 200 points. The rescaled

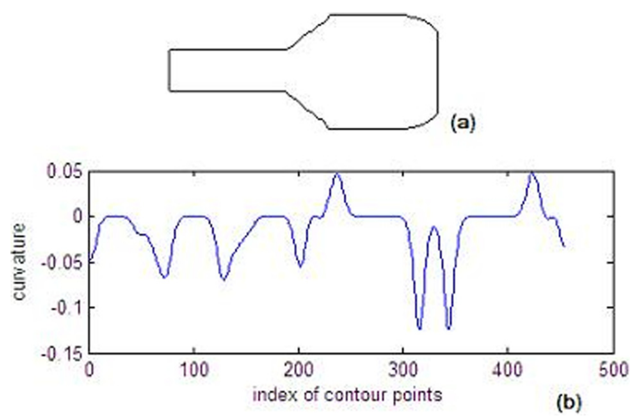

Figure 17. (a) The contour of the bottle model in Fig. 6 and (b) its curvatures.

model of the matching is plotted over the target contour, as shown in Fig. 14(b).

The ICP algorithm is then performed between the rescaled model and the object contour in Fig. 14(b). The matching procedure converged at iteration 15. The transformation reduces the origin boundary matching distance error from 22.5876 pixels to 11.9563 pixels, as shown in Fig. 14(c). The transformed rescaled model is plotted over the target contour, as shown in Fig. 14(d).

Compared with the final stage of registration in the book of Mokhtarian and Bober,[13] the rescaled contour positions obtained from the best curvature matching, shown in Fig. 14(b), show similar results. This is because both algorithms search for a pair of matched segments with the minimum measurement between the model and the object contours and then use the corresponding best

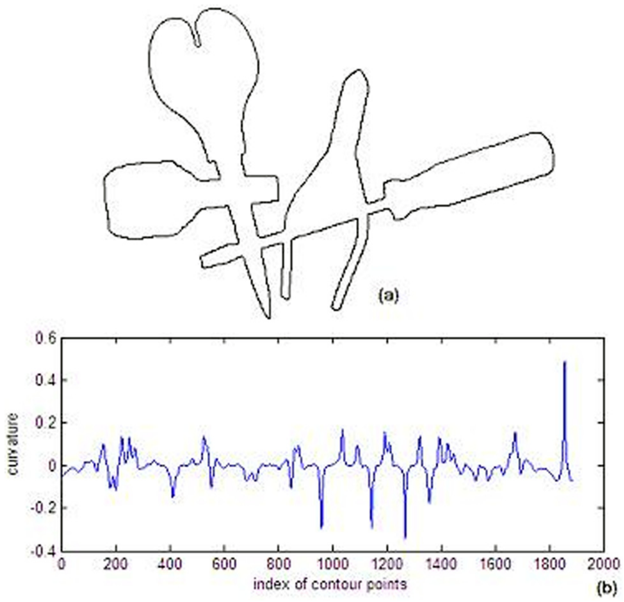

Figure 18. (a) The outermost contour of the object in Fig. 7(c) and (b) its curvature.

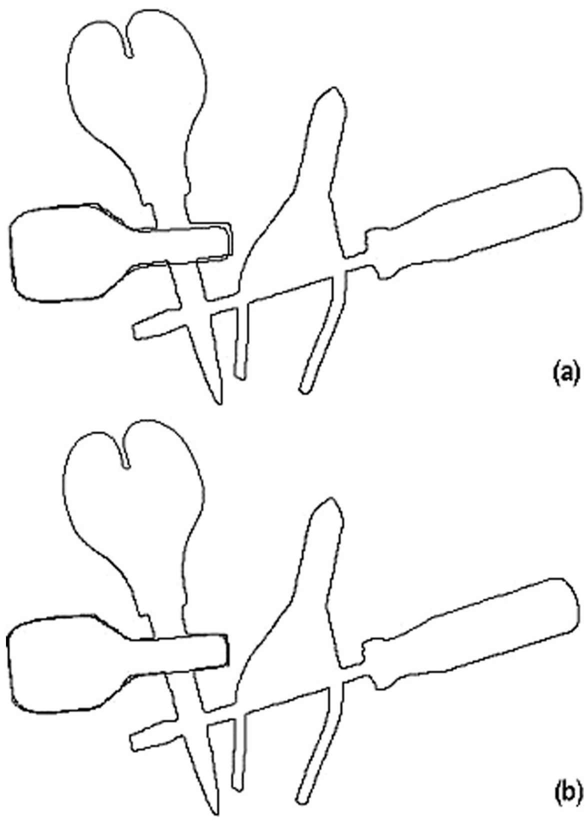

Figure 19. (a) The results of global alignment using curvature matching and (b) the results of local matching using ICP for a single model.

parameters to map the whole model segments to the object contour. In our algorithm, the curvature measurement is used, while the distance measurement was used in Mokhtarian's book. However, the distance matching results using the ICP algorithm show a large improvement, as can be seen in Figs. 14(c) and 14(d).

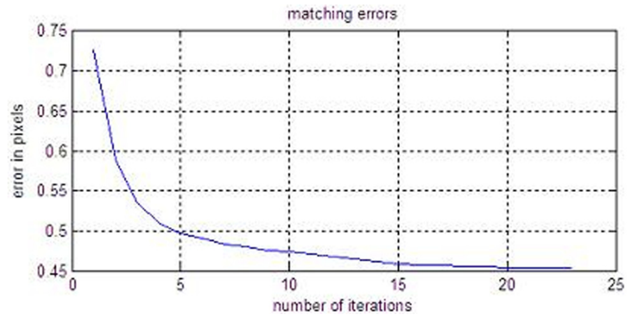

Figure 20. The average distance errors of the model and object boundaries in different iterations during the local alignment process using ICP matching.

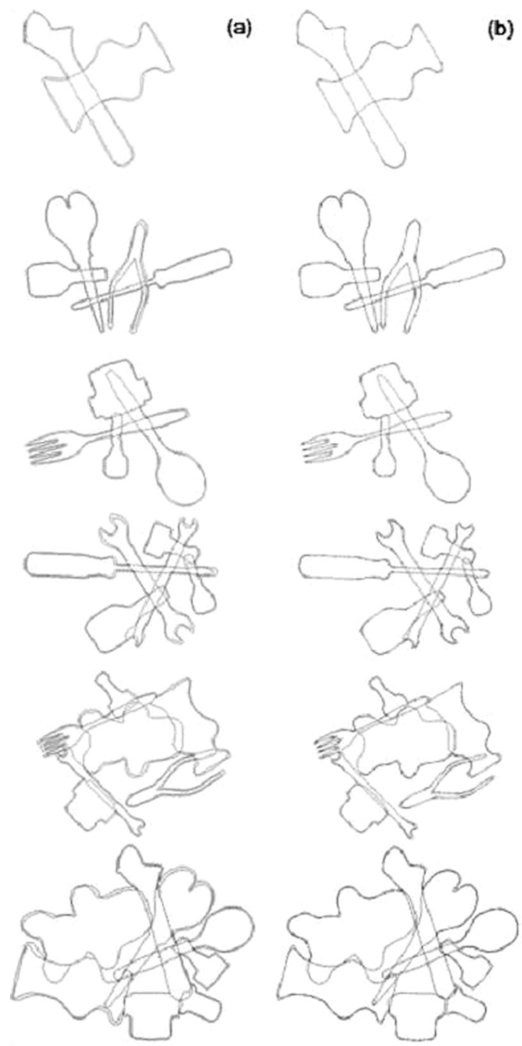

Figure 21. (a) Matching results of SSC compared with (b) those of curvature and ICP matching.

Table I. Matching error comparison (in pixels).

Object set	a	b	c	d	e	f
SSC	2.7848	1.9069	1.9009	2.1305	2.1032	2.1294
Ours	0.8355	0.6221	0.5702	0.6731	0.6550	0.6388

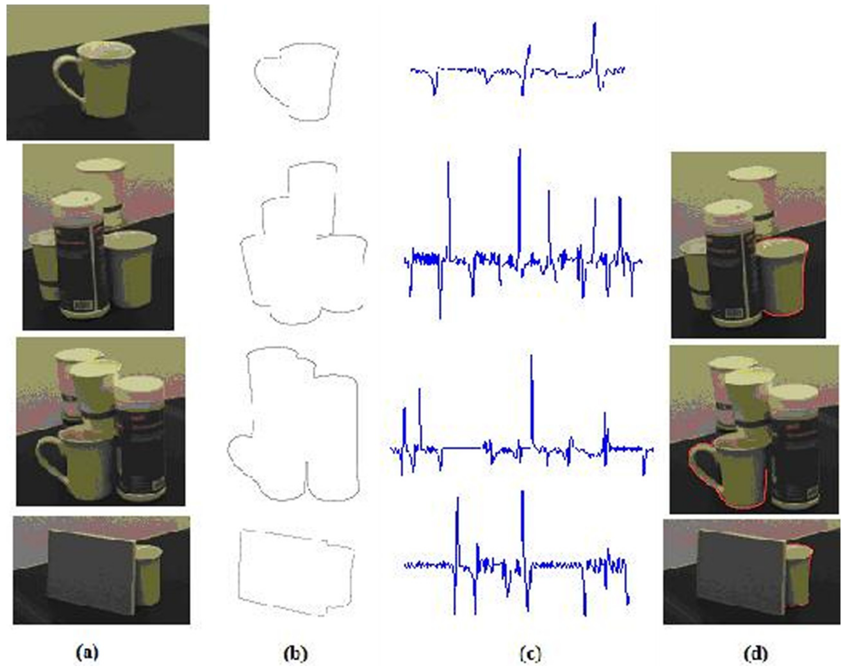

Figure 22. Experiment using real data for a home robot service system to recognize a mass: (a) the object (top) and partial view of the object. (b) Outlines of the view and object; (c) curvatures of the outlines from (b); (d) matching results using curvature representations.

This example is used to show the difference in the matching result compared with the previous work in the case where there is no identical boundary between the model and the object contour. As a special case in this example, no GCP is detected from either the model contour or object boundary. This, in fact, does not affect the result of the matching.

Comparison of the Approach with the Curvature Scale Space Matching Algorithm

In this experiment, the outcomes of our algorithm are compared with those of the work of Mokhtarian and Bober[13] in Silhouette-based isolated object recognition using with the CSS matching algorithm. The comparison results show that our algorithm led to a significant improvement of the boundary matching results.

In the experiment, the contours of all 15 model objects and 6 input images used to evaluate the Silhouette-based object recognition system are used to test our approach. Figure 15 shows the contours of the model objects, and Figure 16 shows the outermost contours of the object sets from the images.[13]

Both the model contours and outermost contours of the object sets are first copied from Ref. 13 and then scanned. The contours detected from the scanned files are extracted. Each contour is then represented as a point sequence. A Gaussian path smoothing function of Eq. (3) with a Gaussian kernel of nine is used to smooth the contours. Then, the curvature estimator of Eq. (2) is used to calculate the curvatures of each smoothed contour.

The curvatures of each model are then used to match the curvatures of the object sets. The best match with a scale factor and the best matched segment are then selected. Then, the model contours are rescaled, as discussed in the section on Boundary Matching.

An example of such a matching is shown in Figures 17–19. Fig. 17(a) shows the contour of the bottle model. Fig. 15(b) shows the curvatures of the smoothed contour of the bottle model. Fig. 18(a) shows the outermost contour of the object from Fig. 16(c), and Fig. 16(b) shows the curvatures of the smoothed outermost contour of the object set.

A scale factor of 0.7861 and the best matching position of 1476 with a best matching region of 190 contour points are obtained from the curvature matching algorithm. The rescaled model is transformed and plotted over the outermost contour of the object, as shown in Fig. 19(a). The new location of the model contour after local alignment is plotted in Fig. 19(b).

The average distance errors between the model boundary and the boundary of the outermost object set from each ICP iteration are plotted in Figure 20. It can be seen in the figure that the local alignment considerably improved the distance error after the first five iterations and resulted in convergence to a distance error of 0.45 pixels after 20 iterations.

An overall comparison of the matching results of the six object sets between SSC without segmentation and our algorithm is plotted in Figure 21. The related distance error between the models and the outermost contours of the six object sets is shown in Table I.

The matching results of our algorithm show a large improvement in the boundary distances. In this comparison, local alignment plays a key role in the improvement of the matching process. Theoretically, local alignment using ICP should be able to achieve the best matching results with a zero distance error, if the model contours and the contours of the outermost contours of the object sets are on the same scale. The matching errors from our algorithm in this example are mostly from the rescaling process.

Experiment with Real Application Data

In this experiment, a set of real application data is used to test our algorithm. The purpose of the application is to detect an object behind others if the whole view of the object is known. The images showing the view of the object and others are shown in Figure 22(a). The shapes of the objects are shown in Fig. 22(b), and corresponding curvatures and matching results of the partial shapes are shown in Figs. 22(c) and 22(d).

CONCLUSIONS

We proposed a simple but robust partial shape recognition algorithm using the curvature and ICP matching methods. Our algorithm differs from many of the previous methods of partial shape classification. First, the proposed algorithm recognizes illusory objects with multiple partial shapes globally. Second, it guarantees global and local matching accuracies. The global matching accuracy is afforded by the GCPs and the local optimal matching by the ICP algorithm. A GCP is independent of any contour evolution process that shifts the positions of the control points. In the selection of the GCPs, the corner detection algorithm is first used to create the local feature of a GCP and the concentric circles representation is used to ensure its globosity. In boundary matching, the 2D ICP algorithm is employed to reach the optimum local matching. It improves and verifies the rescaled boundary obtained from the curvature matching.

The algorithm functions well with the various examples used in the experiment. In the case where the partial shapes contain GCPs, but the object shapes are distorted by moving the different segments around, the local curvature matches become less important, because the global curvature matching only looks for GCP patterns, as shown in the first experiment. However, in the case of a poor image quality or resolution, it is quite possible that no GCPs are detected, but no identical shapes are found. This may be considered matching shapes of the same class rather than the identical shapes shown in the second experiment. In this case, the selections of the threshold values of the curvature matching and boundary matching are critical. First, there may not be even a small identical partial of shapes between the model and target. Second, there may be very large measurement errors in certain parts of the boundary matching. To solve the first problem, a larger smoothing kernel can be selected in order to reduce the curvature matching errors. A higher threshold value can be chosen to enable similar matching. To solve the second problem, certain key segments from the model shapes should be defined and the acceptable ranges of matching errors over those segments need to be defined.

Based on the comparison between the matching results of our algorithm and those of the work of Mokhtarian and Bober,[13] we can conclude that our algorithm provides a large improvement in the distance error between the model and object set boundaries in the case of different contours between the model and object, as shown in the subsection of Experiments with Partial shape with Differences Contour from Model, and of outermost contour matching as shown in subsection of Comparison of the Approach with the Curvature Scale Space Matching Algorithm. However, there is limitation to our algorithm. In curvature matching, while the transformation from the curvature representation is rotation and translation invariant, the rescaling process may change the outermost contours slightly when the size of the model and that of the outermost contour of the object set are not the same. This is because the rescaling process is based on the curvatures that are estimated based on smoothed contours. When a nonzero scaling factor is applied, the new contour is, in fact, the result of a zoom-in or zoom-out operation applied to the smoothed contour rather than the original one.

Our future work will be aimed at solving the problem of no GCPs being detected in the case where the object shape is distorted by moving the different segments around the object boundary.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0014697).

REFERENCES

1. D. H. Ballard, "Generalized Hough transform to detect arbitrary patterns", IEEE Trans. Pattern Anal. Mach. Intell. **12**, 111–122 (1981).
2. G. Stockman, "Object recognition and localization via pose clustering", Comput. Vis. Graph. Image Process. **40**, 361–387 (1987).
3. S. Umeyama, "Parameterized point pattern matching and its application to recognition of object families", IEEE Trans. Pattern Anal. Mach. Intell. **15**, 136–144 (1993).
4. H. Wolfson and I. Rigoutsos, "Geometric hashing: An overview", IEEE Comput. Sci. Eng. **4**, 10–21 (1997).
5. R. Osada, T. Funkhourser, B. Chazelle, and D. Dobkin, *Matching 3D Models with Shape Distributions* (Princeton University, Princeton, NJ, 2001).
6. D. Huttenlicher and S. Ullman, "Object recognition using alignment", *Proc. 1st Int. Conf. Computer Vision* (London, UK, 1987), pp. 102–111.
7. S. Sclaroff and A. P. Pentland, "Model matching for correspondence and recognition", IEEE Trans. Pattern Anal. Mach. Intell. **17**, 545–561 (1995).
8. S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation", Pattern Recogn. **12**, 269–275 (1980).
9. E. Person and K. Fu, "Shape discrimination using Fourier descriptors", IEEE Trans. Syst. Man Cybern. **7**, 170–179 (1977).
10. E. Zahzah, "Wavelet transform for partial shape recognition using submatrix matching", *Proc. 3rd Int. Conf. Computer Vision Theory and Applications* (Funchal-Madeira, Portugal, 2008), pp. 513–517.
11. S. Gold, "Matching and learning structural and spatial representations with neural networks", Ph.D. thesis (Yale University, New Haven, CT, 1995).
12. S. Belongie, J. Malik, and H. Puzicha, "Shape context: A new descriptor for shape matching and object recognition", Adv. Neural Inf. Process. Syst. **13**, 831–837 (2001).

[13] F. Mokhtarian and M. Bober, *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization* (Kluwer Academic, Norwell, MA, 2003).
[14] S. Kopf, T. Haenselmann, and W. Effelsberg, "Enhancing curvature scale space features for robust shape classification", *IEEE Int. Conf. Multimedia and Expo* (IEEE, Piscataway, NJ, 2005), pp. 4–7.
[15] P. V. C. Hough, "Method and means for recognizing complex patterns", US Patent 3,069,654 (1962).
[16] R. O. Duda and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures", Commun. ACM **15**, 11–15 (1992).
[17] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recogn. **13**, 111–122 (1981).
[18] P. E. Danielsson, "A new shape factor", Computer Graphics and Image Processing **7**, 292–299 (1978).
[19] E. Saber, Y. Xu, and A. M. Tekalp, "Partial shape recognition by submatrix matching for partial matching guided image labeling", Pattern Recogn. **38**, 1560–1573 (2005).
[20] H. Freeman, "Computer processing of line-drawing images", ACM Comput. Surv. **6**, 57–97 (1974).
[21] J. W. McKee and J. K. Aggarwal, "Computer recognition of partial views of curved objects", IEEE Trans. Comput. **C-26**, 790–800 (1977).
[22] T. Pavlids, "Polygon approximations by Newton's method", IEEE Trans. Comput. **C-26**, 800–807 (1977).
[23] P. C. Yuen, Y. Y. Wong, and C. S. Tong, "Contour detection using enhanced snakes algorithm", Electron. Lett. **32**, 202–204 (1996).
[24] A. Asada and M. Brady, "The curvature primal sketch", IEEE Trans. Pattern Anal. Mach. Intell. **8**, 2–3 (1986).
[25] H.-C. Liu and M. D. Srinath, "Partial shape classification using contour matching in distance transformation", IEEE Trans. Pattern Anal. Mach. Intell. **12**, 1072–1079 (1990).
[26] K. Sohn, J. H. Kim, and W. E. Alexander, "A mean field annealing approach to robust corner detection", IEEE Trans. Syst., Man, Cybern., Part B: Cybern. **28**, 82–90 (1998).
[27] X. Xu, D. J. Lee, S. Antani, and L. R. Long, "Partial shape matching of spine X-ray shapes using dynamic programming", *Proc. 17th IEEE Symposium on Computer-based Medical Systems* (IEEE, Piscataway, NJ, 2004), pp. 97–102.
[28] N. Ansari and E. J. Delp, "Partial shape recognition: A landmark-based approach", IEEE Trans. Pattern Anal. Mach. Intell. **12**, 470–483 (1990).
[29] C. M. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoff in dynamic time warping algorithms for isolated word recognition", IEEE Trans. Acoust., Speech, Signal Process. **28**, 623–635 (1980).
[30] T. F. Chan and L. A. Vese, "Active contours without edges", IEEE Trans. Image Process. **10**, 266–277 (2001).
[31] F. Mokhtarian and L. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes", IEEE Trans. Pattern Anal. Mach. Intell. **8**, 34–43 (1986).
[32] F. Mokhtarian and S. Abbasi, "Matching shapes with self-intersections: Application to leaf classification", IEEE Trans. Image Process. **13**, 653–661 (2004).
[33] J. O'Rourke and R. Washington, "Curve similarity via signatures", in *Computational Geometry* (Elsevier Science North-Holland, Amsterdam, 1985), pp. 295–317.
[34] F. Y. Shih and P. Yeh, "Model-based partial shape recognition using contour curvature and affine transformation", *Proc. Int. Conf. Systems Integration* (Morristown, NJ, 1990), pp. 294–301.
[35] W. A. Perkins, "A model-based vision system for industrial parts", IEEE Trans. Comput. **C-27**, 126–143 (1978).
[36] T. P. Wallace and P. A. Wintz, "An efficient three-dimensional aircraft recognition algorithm using Fourier descriptors", Comput. Graph. Image Process. **13**, 99–126 (1980).
[37] S. A. Dudani, "Aircraft identification by moment invariants", IEEE Trans. Comput. **C-26**, 39–46 (1977).
[38] D. Marr and E. C. Hildreth, "Theory of edge detection", Proc. R. Soc. London **207**, 187–217 (1980).
[39] A. Bandera, C. Urdiates, F. Arrebola, and F. Sandoval, "Corner detection by means of adaptively estimated curvature function", Electron. Lett. **36**, 124–126 (2000).
[40] D. M. Wuescher and K. L. Boyer, "Robust contour decomposition using a constant curvature criterion", IEEE Trans. Pattern Anal. Mach. Intell. **13**, 41–51 (1991).
[41] K. Sohn, W. E. Alexander, J. H. Kim, and W. E. Snyder, "A constrained regularization approach to robust corner detection", IEEE Trans. Syst. Man Cybern. **24**, 820–828 (1994).
[42] D. Chetverikov and Z. Szabó, "Detection of high curvature points in planar curves", *Proc. Computer Analysis of Images and Patterns* (Groningen, The Netherlands, 2003), pp. 746–753.
[43] M. Marji and P. Siy, "A new algorithm for dominant point detection and polygonization of digital curves", Pattern Recogn. **36**, 2239–2251 (2003).
[44] J. Canny, "A Computational approach to edge detection", IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–697 (1986).
[45] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes", IEEE Trans. Pattern Anal. Mach. Intell. **14**, 239–256 (1992).
[46] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images", *Proc. IEEE Conference Robotics and Automation* (IEEE, Piscataway, NJ, 1991), pp. 2724–2729.
[47] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces", Int. J. Comput. Vis. **13**, 119–152 (1994).