Asian Journal of Applied Sciences (ISSN: 2321 – 0893)
Volume 10 – Issue 6, December 2022

Shape Recognition Using Segmenting and String Matching

Wen-Yen Wu

Department of Industrial Management, I-Shou University
Kaohsiung,Taiwan

Email: wywu [AT] isu.edu.tw

ABSTRACT—*This paper presents an efficient way to represent objects. The image of the object is converted into an edge image. Important points of the curve are identified by the dominant point detection method. A line segment of every two consecutive important points is a categorical line segment or a non-linear line segment. Nonlinear segments are fitted as circular arcs. In addition, the compactness of approximate polygons is used as a feature in the shape recognition process. Experimental results show that using this new global feature has better recognition performance than traditional features such as relative distance, length and angle. Overall the new method is efficient and effective in representing and recognizing shapes.*

Keywords—Shape recognition, string matching, curve, feature

1. INTRODUCTION

Shape recognition is an important problem in many applications. Two broad categories of shape recognition methods are statistical methods and syntactic methods. The first method cannot exploit the structural information of the shape, and the second method is sensitive to noise. In recent years, some combined methods of statistical methods and syntactic methods have been proposed [4, 9, 10, 11, 16].

Tsai and Fu [9] combined statistical and syntactic approaches by using attribute grammars. They use three types of editing operations to transform one string into another [8]. Furthermore, Tsai and Yu [10] use attributed string matching and merging involving more powerful editing operations. By combining operations, the recognition rate can be improved. Tsay and Tsai [11] proposed another new split-editing operation for attributed string matching. They use approximate polygon lengths and angles as features in recognition.

The recognition rate can be improved by introducing a new powerful editing operation, but there are three problems in the above method: the editing sequence is complicated, the editing cost of these two new editing operations is not easy to define, and the reference line needs to be set in terms of calculating the angle, so they are Linear string matching technique. Maes [7] proposed a loop string matching technique for polygon shape recognition to solve the above problems. He uses only the three normal editing operations required for loop string matching.

Chang et al. [2] proposed a simplified loop string matching technique. The relative distances of feature points to centroids are used as features in their paper. If the number of features of the two matching shapes is the same, the recognition rate is better. But this assumption is not practical for most applications. Kaygin and Bulut [5] use vertices instead of line segments as primitives. Delete and insert operations are used to achieve the advantages of split and merge operations. Their method works, but the calculations seem to be complicated. Bunke and Bühler [1] used the curvature of edge pixels as a feature in string matching problems. They use fixed-length line segments to avoid merge operations. Chen et al. [3] proposed fuzzy splitting and merging to find initial line segments. Wu and Wang [15] proposed a two-stage string matching to reduce the effect of uneven segmentation. A feature is the inverse of the compactness of a triangle formed by the centroid and two adjacent vertices. Matched vertices are used to evaluate local dissimilarity.

It is agreed that representation and matching are the two main problems involved in pattern recognition. In many industrial applications, round objects are often found everywhere. It is important to identify them during the manufacturing process. Over the past few years, many methods for detecting circles have been proposed.

Detecting dominant points is an important step in many machine vision applications. In image processing and pattern recognition, the main points are used to preserve the shape of planar curves and to reduce the amount of data. Many algorithms have been developed to detect dominant points. Most of them can be divided into two categories: (1) polygon approximation methods and (2) corner detection methods.

This paper combines principal point detection with circle detection to find a way to represent objects. We propose a new recognition function. Vertices are used as primitives. The characteristic is the compactness of the polygon formed by the centroid and three adjacent vertices. Loop string matching is applied to overcome the problem of initial vertex determination. Experimental results show that the method is efficient and effective in rendering objects.

2. FEATURE EXTRACTION

In general, features for shape recognition should be translation-invariant, rotation-invariant, and scale-invariant (TRS-invariant). Knowing the principal points on the boundary of the object is sufficient to represent the shape of the object. Therefore, we can use information about dominant points to find original features [14].

In this paper, the closeness of a polygon composed of three adjacent principal points and the centroid of a shape are used as features for shape recognition. Also, features should be normalized so that they are not affected by changes in position, orientation, and scaling.

The proposed feature extraction method consists of four stages: dominant point detection, line segment and arc determination, circular arc fitting, and shape representation.

2.1 Dominant point detection

Only breakpoints are considered as candidates for dominant points. It will reduce computation time for determining support regions and curvature estimation. The principal point is the point with the local maximum curvature. For continuous curves, the curvature at a point is defined as the rate of change of the slope as a function of arc length. However, the above definition of curvature does not apply to digital curves. Therefore, dominant point detection algorithms use information that can be extracted from neighbors to estimate curvature. In this paper, the k cosine of two vectors is used as the estimated curvature, which is defined as (see Figure 1):

$$\cos_{ik} = \frac{\vec{a}_{ik} \bullet \vec{b}_{ik}}{|\vec{a}_{ik}||\vec{b}_{ik}|} ,$$

(1)

where $\vec{a}_{ik} = (x_{i-k} - x_i, y_{i-k} - y_i)$ is the backward vector, $\vec{b}_{ik} = (x_{i+k} - x_i, y_{i+k} - y_i)$ is the forward vector, $\cdot$ is the inner product operator, and $|*|$ is the length of the vector.

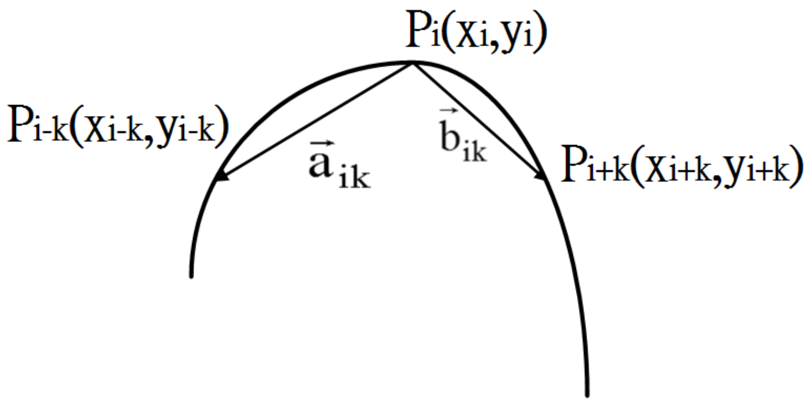

Figure 1: Estimate curvatures by the k-cosine by three points

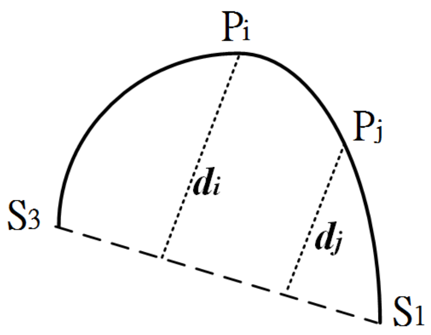

Figure 2: The computation of average distance of a segment.

2.2 Linear or nonlinear segment determination

The above-mentioned dominant point detection method can divide the curve into several segments. The next step is to classify them into linear and nonlinear parts. In this paper, the average distance from each point to the endpoint is used as a criterion to evaluate the distortion caused by the approximate line segment (see Fig. 2). it is defined as

$$D = \sum_{i=1}^{n} d_i / n, \tag{2}$$

where n is the number of points, d_i is the distance from P_i to the approximated line segment.

Since the average distance is smaller, the linearity is better. A line segment with a small average distance is considered a linear line segment. Otherwise, it will be treated as a non-linear segment. For example, the curve shown in Figure 1 has three dominant points S_1 to S_3. The segments between S_1 and S_2 and the segments between S_2 and S_3 will be considered as linear segments. The segment between S1 and S3 will be considered a non-linear segment. Non-linear line segments will be candidates for circles.

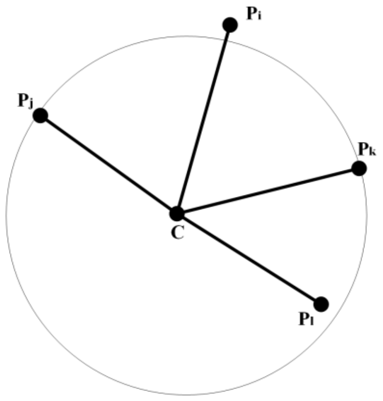

Figure 3: Circle fitting to find a set of center and radius.

2.3 Circular arc fitting

For each nonlinear segment, use the proposed least squares fit to find its estimated center and radius. Figure 3 shows the circular results. Suppose that $(\hat{x}_c, \hat{y}_c, \hat{r})$ is the estimate of (x_c, y_c, r). Estimated center and radius for a set of n points (x_i, y_i), for i=1, 2, …, n, are given as :

$$\hat{x}_c = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1}, \tag{3}$$

$$\hat{y}_c = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1}, \tag{4}$$

$$\hat{r} = \left(\frac{\sum_{i=1}^{n} x_i^2 - 2x_c \sum_{i=1}^{n} x_i + n x_c^2 + \sum_{i=1}^{n} y_i^2 - 2y_c \sum_{i=1}^{n} y_i + n y_c^2}{n}\right)^{1/2}, \tag{5}$$

where $a_1 = 2((\sum_{i=1}^{n} x_i)^2 - n\sum_{i=1}^{n} x_i^2)$, $b_1 = 2(\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i - n\sum_{i=1}^{n} x_i y_i) = a_2$, $b_2 = 2((\sum_{i=1}^{n} y_i)^2 - n\sum_{i=1}^{n} y_i^2)$,

$$c_1 = \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} x_i - n\sum_{i=1}^{n} x_i^3 + \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i^2 - n\sum_{i=1}^{n} x_i y_i^2,\ c_2 = \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - n\sum_{i=1}^{n} y_i^3 + \sum_{i=1}^{n} y_i \sum_{i=1}^{n} y_i^2 - n\sum_{i=1}^{n} x_i^2 y_i$$

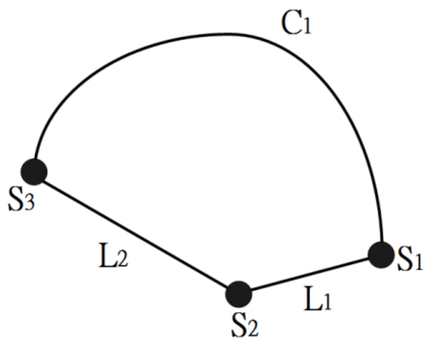

Figure 4: Object representation: $L_1L_2C_1$.

2.4 Shape representation

In the final stage, objects are encoded by detected line segments and arcs. For the objects in Figure 4, there are three main pints. The object consists of two line segments and an arc. Each line segment is represented by two endpoints, and the arc is represented by the center and r.

Let P_i, for i=1,2,...,N, be the ith point whose coordinates are of shape (x_i, y_i). Important points on shape boundaries can be detected by the principal point detection method [14]. Suppose the detected dominant point is Vi, for i=1, 2, ..., M. We can find the centroid of the shape C = (x_c, y_c) by the following equation:

$$x_c = \frac{1}{N} \Sigma x_i \quad \text{and} \quad y_c = \frac{1}{N} \Sigma y_i. \tag{6}$$

(a) Distance. The distance between the vertex and the object's centroid. Let d_i be the i th distance and $d_i = \left|\overline{V_iC}\right|$ (see Figure 5).

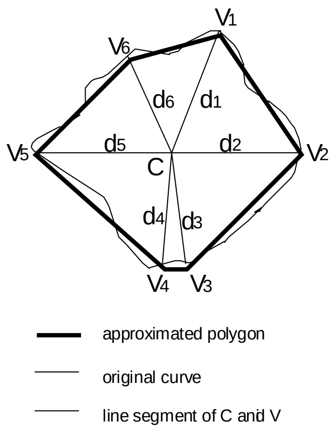

Figure 5: The relative distances: $d_i=\left|\overline{V_iC}\right|$, for i=1,2,...,M.

(b) Length and angle. Approximate polygon lengths and angles. Let l_i and θ_i be the length of $\overline{V_iV_{i+1}}$ and the angle of $\angle V_{i-1}V_iV_{i+1}$, respectively (see Figure 6).

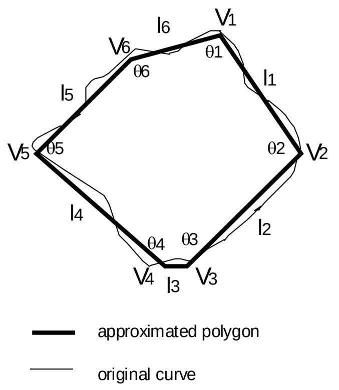

Figure 6: The length $l_i=\left|\overline{V_iV_{i+1}}\right|$ and the angle $\Theta_i=\angle V_{i-1}V_iV_{i+1}$, for i=1,2,...,M, where V_i is the i th vertex.

(c) Compactness. In order to avoid two adjacent principal points and the center of mass of the object being on the same straight line, the area of the triangle formed by these three points will be zero. We use three adjacent three dominant points instead of two adjacent points. Assuming ci is the ith compactness, it is defined as [15]

$$c_i=\frac{p_i^2}{a_i}, \tag{7}$$

where $p_i=\left|\overline{V_iV_{i-1}}\right|+\left|\overline{V_iV_{i+1}}\right|+\left|\overline{V_{i-1}C}\right|+\left|\overline{V_{i+1}C}\right|$ is the perimeter and a_i is the area of the polygon (see Figure 7).

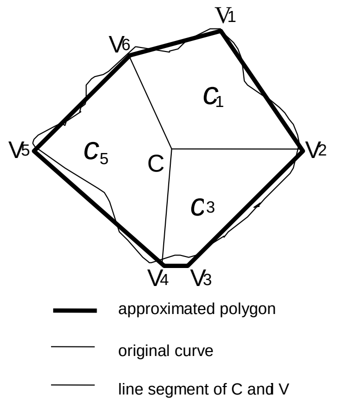

Figure 7: The compactness of the polygon: $c_i= p^2/a_i$, for i = 1, 2,..., M.

3. SHAPE RECOGNITION BY CYCLIC STRING MATCHING

The cyclic string matching technique was proposed by Maes [6]. Suppose X is a set of symbols. We can define the concatenation of symbols $s_1,s_2,\ldots,s_n$ to be the string **s**. The number of symbols in the string s is called the length of s, which can be expressed as |**s**|. Also, a string of length zero is called the empty string and can be denoted as λ.Suppose **s** and **t** are two strings and let |**s**|=n and |**t**|=m. For a given editing cost function ε, three types of arcs can be defined:

Insertion. (v(i,j), v(i,j+1)) with weight $w_{0,j+1}=\varepsilon(\lambda,t_{j+1})$, for i=0, 1,..., n, and j=0,1,...,m-1.

Deletion. (v(i, j),v(i+1, j)) with weight $w_{i+1,0}=\varepsilon(s_{i+1},\lambda)$, for i=0, 1, ..., n-1, and j=0,1,..., m.

Change. (v(i, j), v(i+1, j+1)) with weight $w_{i+1,j+1}=\varepsilon(s_{i+1},t_{j+1})$, for i=0,1,...,n-1, and j=0, 1,..., m-1.

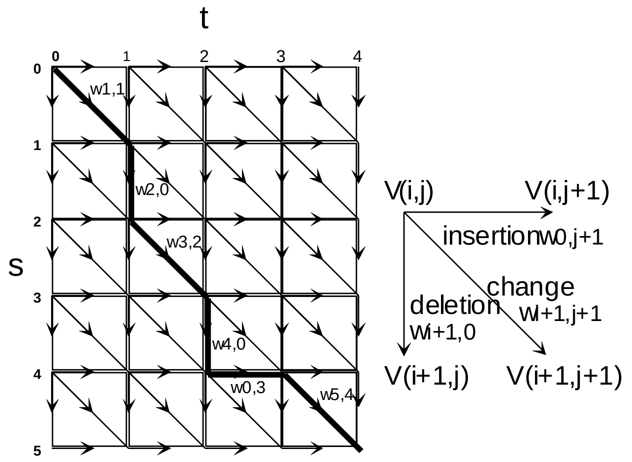

Figure 8: The edit graph **G** for |**s**| = 5 and |**t**| = 4, and the shortest path.

Now we can define the edit graph associated with **s** and **t** as a weighted graph **G** (see Figure 8) with vertices v(i, j), for i=0,1,...,n, and j=0,1,...,m. Wagner and Fischer [12] showed that the problem of finding the sequence of minimum-cost edits that bring s into t is now reduced to finding the shortest path in G from v(0, 0) to v(n, m). Wong and Chandra [13] showed that the above algorithm will have optimal O(mn) computation time. Examples of shortest paths and their corresponding editing sequences are shown as bold lines in Figure 8.

For the cyclic string matching problem, the problem now is to find the edit distance δ([s],[t]) and its corresponding edit sequence, where [s] and [t] are the cyclic strings of s and t, respectively. It's known as the circular string-to-string correction problem. In general, it is assumed that m≤n. Also, assume that the strings s and t are cyclic strings. The edit distance is defined in Maes [7]:

δ([**s**], [**t**])=min{δ(**s**,σ^j(**t**)): j=0, 1, …,m-1}, (8)

where σ^j(**t**) is the string obtained from **t** after j cyclic shifts.

To find the edit distance, suppose that **tt**=$t_1t_2…t_mt_1t_2…t_m$ is the string which concatenates **t** with itself. The edit graph **H** associated with **s** and **tt** can be shown as in Figure 9. We can find the shortest path from v(0, j) to v(n, m+j), for j=0,1,...,m-1 in Figure 9. And the edit distance δ(**s**,σ^j(**t**)) can be determined. Further, the minimum edit distance can be found and its corresponding edit sequence can be identified. Maes [6] showed that the above algorithm can be done in O(mn log m) computation time.

To find the edit distance, suppose **tt**=$t_1t_2…t_mt_1t_2…t_m$…tm is the string concatenating t with itself. The edit graph H associated with s and tt can be shown in Figure 9. We can find the shortest path from v(0, j) to v(n, m+j) for j=0,1,... ,m-1 in Figure 9. The edit distance δ(s,σj(t)) can be determined. Further, the minimum edit distance can be found and its corresponding edit sequence can be identified. Maes [6] showed that the above algorithm can be completed in O(mn log m) computation time.

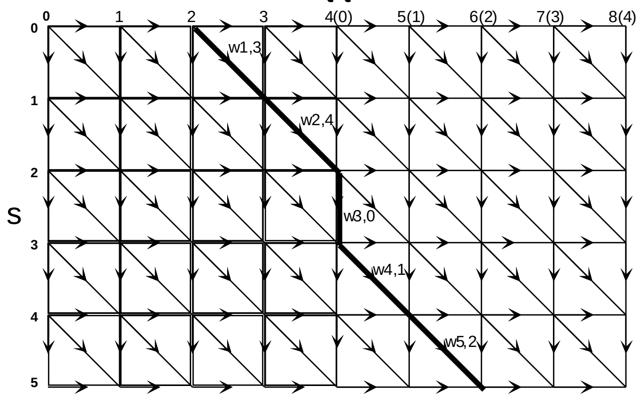

Figure 9: The edit graph **H** associated **s** and **tt,** the edit sequence the best matched pair, and the shortest path .

After principal point detection is performed on a shape, it can be approximated by a polygon of vertices [14]. The relative distances, lengths, and angles of polygons, as well as the compactness of polygons can be computed by the definitions in Section II. Additionally, we can obtain normalized values of these features.

Assume I and R are the input shape and the reference shape, respectively. Approximate polygons of shapes are also denoted $\mathbf{I_A}$ and $\mathbf{R_A}$. Suppose n and m are the number of vertices of $\mathbf{I_A}$ and $\mathbf{R_A}$ respectively. We know that features can be represented as symbols, so we can use these strings to represent shapes. That is, we can use the strings $\mathbf{s}=s_1s_2\ldots s_n$ and $\mathbf{t}=t_1t_2\ldots t_m$, where s_i and t_j, for i=1,2,...,n, j=1,2,...,m , are the eigenvalues representing the input shape and the reference shape, respectively. Let X be the set whose elements are features. Since each feature of a polygon is cyclic, the problem of matching two shapes is the same as the problem of cyclic string matching of strings s and t.

Given an editing cost function ε, we can construct an editing graph H associated with s and tt as shown in Figure 9. Further, the shortest path can be determined, and the minimum edit distance in cyclic string matching is called the matching cost.

Maes [7] gives the cost function by using length and angle as features: for all x, y $\in$X,

$$\varepsilon(x\rightarrow y)=\begin{cases}|x-y| & \text{if x and y are angles,}\\ w|x-y| & \text{if x and y are lengths,}\\ \infty & \text{otherwinses,}\end{cases}$$

(9)

$$\varepsilon(x\rightarrow\lambda)=\varepsilon(\lambda\rightarrow x)=\begin{cases}|x| & \text{if x is an angle,}\\ w|x| & \text{if x is a length,}\end{cases}$$

where w ∈□ is a weighting factor.

The above formula is very simple, but the main problem is that an appropriate weight factor w must be selected during calculation. Choosing an appropriate set of weights is not easy when using length and angle as raw features. In this paper, we do not have the problem of choosing an appropriate weighting factor because only one feature (compactness) is used in string matching. Let X be a set whose elements are compact (including 0), and let the value of λ be zero. Then, for all x,y $\in\mathbf{X}$, the cost function can be simplified as:

$$\varepsilon(x\rightarrow y)=|x-y|.$$
(10)

4. RESULTS AND DISCUSSIONS

To evaluate the proposed method, an experiment is designed to test the performance of the new features. For comparison, two other general features, namely relative distance and length and angle, were also evaluated. Therefore, this experiment had to evaluate three distinct features in string matching. The edit cost defined in (10) is only for 1D features. Not applicable to two-dimensional features such as length and angle. So, here, the cost function for length and angle features is defined as:

$$\varepsilon(x\rightarrow y)=\frac{1}{2}(|x_1-y_1|+|x_2-y_2|),$$
(11)

where x=(x_1, x_2)=(length, angle) and the setting of y is similar to x.

Also, features should be normalized to be independent of scaling. Suppose f_i is the ith feature and f_i is its normalized value. The three normalization methods are defined as follows:

(a) Divided by the maximum value:

$$f_i□=\frac{f_i}{\max_f}, \quad \text{where} \quad \mathbf{max}_f=\mathbf{max}\{f_i\}.$$
(12)

(b) Divided by the sum:

$$f_i□=\frac{f_i}{sum_f}, \quad \text{where} \quad \mathbf{sum}_f=\Sigma f_i.$$
(13)

(c) Normalized to [0, 1]:

(d) f_i□= $\frac{f_i - min_f}{max_f - \min_f}$, where **max**$_f$=**max**{f_i} and **min**$_f$ =min {f_i}. (14)

Therefore, we have 3×3 = 9 different feature combinations. The 10 test images are shown in Figure 10. In the experiments, identification is first performed on each test set. Since a good shape recognition method should be able to correctly classify shapes in different orientations and zoom ratios. Here, for each shape, 100 test images of different orientations and scales are grabbed for recognition. Therefore, a total of 12 × 100 = 1200 test images are used in the experiment. A cyclic string matching algorithm is applied to each test image. An error is logged when it is misclassified. The recognition rate can then be calculated.

The experimental results for the test shapes in Fig. 10 are shown in Table 1. The data listed in Table 1 are the recognition rates of nine combinations of feature settings and normalization methods. It is easy to see that the recognition rate using compactness as the recognition feature is very high (above 92%).

In addition, the recognition rate was found to be low when relative distance or length and angle were used as features. The way to improve the recognition rate of these two features is to set some parameters in the recognition process. However, parameter setting is not an easy task and may be very tedious. In addition, the recognition rates using relative distance, length, and angle as features vary greatly under different normalization methods. For example, using length and angle as features, the recognition rate is the largest at 81% when normalized to [0, 1], but the recognition rate is poor (75%) when normalized with the **sum**$_f$ method. Using compactness as a feature in a loop string matching algorithm has very high and consistent recognition rates. This is a very useful feature for recognizing the shapes in Figure 10.

Table 1: Comparison of recognition rates (%) for testing shapes in Figure 6.

Normalization Method	FEACTURES		
	d_i	l_i&Θ_i	c_i
max$_f$	83	79	96
sum$_f$	86	75	95
[0, 1]$_f$	89	81	92

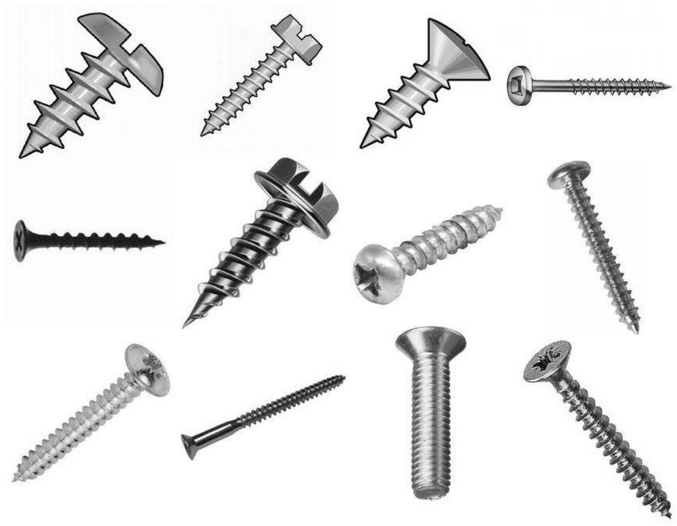

Figure 10: The testing images of the 12 screws.

In order to evaluate the proposed method, another set of images were tested in the experimens. They are the gear images consisting of line segments and circular arcs. Figure 11(a) is the original images. Figure 11(b) shows the edgy images. The experimental results show that the method is efficient and effective in presenting objects.

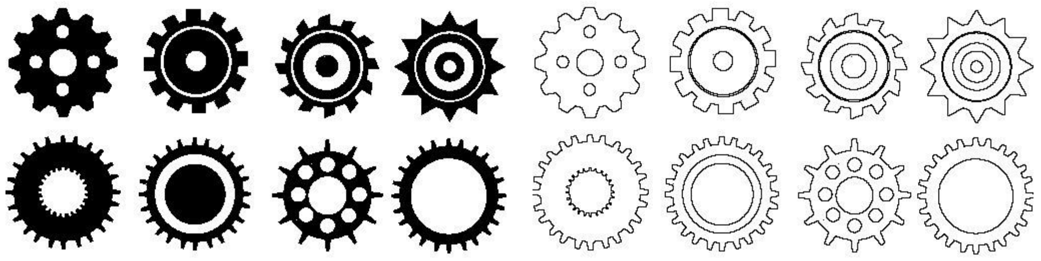
(a) (b)

Figure 11: Gear images: (a) original images, (b) edgy images.

5. CONCLUSIONS

This paper proposes a simple shape recognition method. We propose a method for representing objects. The proposed method consists of four stages: dominant point detection, line segment and arc determination, circular arc fitting, and shape representation. Experimental results show that the method is efficient and effective in rendering objects. In addition, the compactness of approximate polygons is used as a feature in the shape recognition process. Experimental results show that using this new global feature has better recognition performance than traditional features such as relative distance, length and angle. Furthermore, the new features do not require any parameters to be used in shape recognition. Therefore, the proposed method is more robust to changes in size or rotation.

6. REFERENCES

[1] H. Bunke and U. Bühler, "Applications of approximate string matching to 2D shape recognition," Pattern Recognition, vol. 26, no. 12, pp. 1797-1812, 1993.
[2] C. C. Chang, S. M. Hwang, and D. J. Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid." Pattern Recognition, vol. 24, pp. 1053-1063, 1991.
[3] S. W. Chen, S. T. Tung, C. Y. Fang, S. Cherng, A. K. Jain, "Extended attributed string matching for shape recognition," Computer Vision and Image Understanding, vol. 70, no. 1, pp. 36-50, 1998.
[4] K. S. Fu, "A step towards unification of syntactic and statistical pattern recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 5, 200-205, 1983.
[5] [S. Kaygin and M. M. Bulut, "Shape recognition using attributed string matching with polygon vertices as the primitives," Pattern Recognition Letters, vol. 23, pp. 287-294, 2002.
[6] [M. Maes, "On a cyclic string-to-string correction problem Information," Processing Letters, vol. 35, pp. 73-78, 1990.
[7] [M. Maes, "Polygonal shape recognition using string-matching techniques," Pattern Recognition, vol. 24, pp. 433-440, 1991.
[8] [D. Sankoff and J. B. Kruskal (ed.), Time Warps, String Edits and Micromolecules: The Theory and Practice of Sequence Comparison, Addison Wesley, Reading, MA, 1983.
[9] [W. H. Tsai and K. S. Fu, "Attributed grammar- a tool for combining syntactic and statistical approaches to pattern recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 10, pp. 873-885, 1980.
[10] [W. H. Tsai and S. S. Yu, "Attributed string matching with merging for shape recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 7, pp. 453-462, 1985.
[11] Y. T. Tsay and W. H. Tsai, "Model-guided attributed string matching by split-and-merge for shape recognition," International Journal of Pattern Recognition and Artificial Intelligence, vol. 3, pp. 159-179, 1989.
[12] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," J. ACM, vol. 21, pp. 168-173, 1974.
[13] C. K. Wong and A. K. Chandra, "Bounds for the string-editing problem," J. ACM, vol. 23, pp.13-16, 1976.
[14] W. Y. Wu, "A simple method for dominant point detection," Imaging Science Journal, vol. 49, pp. 125-133, 2001.
[15] W. Y. Wu and M. J. J. Wang, "Two-dimensional object recognition through two-stage string matching," IEEE Trans. Image Processing, vol. 8, no. 7, pp. 978-981, 1999.
[16] K. C. You and K. S. Fu, "A syntactic approach to shape recognition using attributed grammars," IEEE Trans. Systems, Man, and Cybernetics, vol. 9, pp. 334-345, 1979.