Shape recognition using fuzzy string-matching technique

W-Y Wu* and **C-S Lai**
Department of Industrial Engineering and Management, I-Shou University, Kaohsiung, Taiwan 84008

Abstract: Object recognition is a very important task in industrial applications. Attributed string matching is a well-known technique for pattern matching. The present paper proposes a fuzzy string-matching approach for two-dimensional object recognition. The fuzzy numbers are used to represent the edit costs. Therefore, the edit distances are also presented as fuzzy numbers. The attributed string-matching problem is then equivalent to a fuzzy shortest path problem. The edit distance between two shapes is presented as a fuzzy number. By ranking the fuzzy edit distances, the input shape is classified as the reference shape that has the minimum fuzzy edit distance. The experimental results show that the proposed method can effectively recognize two-dimensional objects.

Keywords: object recognition, string matching, shortest path, fuzzy numbers, feature extraction

1 INTRODUCTION

Flexibility in designing and manufacturing products is very important for modern and future industry. The use of computer vision can contribute much more in this area. The primary goal for a manufacturing eye is to build systems that can globally understand the scene being detected. In general, there are three valuable applications of computer vision in industrial environments. They are (1) identification and location determination, (2) visual inspection and (3) control of machines and processes.[1]

A vision system may be required to identify objects in an automated warehousing environment or be needed in some other situations. The present paper focuses on the determination of identification and location, specifically two-dimensional (2D) object recognition. Recognizing 2D objects is very important and practical, because the thickness of many mechanical parts is considered to be insignificant compared with the other two dimensions. It has been agreed that representation and matching are the two major problems involved in pattern recognition. To solve these problems, various methods have been proposed.[2]

The 2D object recognition techniques can be classified into two major categories: the statistical method and the syntactic method. Generally, the former fails to make use of the structural information of the shape, and the latter is sensitive to noise. Hence, a combined approach has been proposed,[3–10] which will have the merits of both methods and eliminate the corresponding disadvantages.

Fu[3] pointed out that the statistical and syntactic approaches should be unified to solve the pattern recognition problems. Tsai and Fu[6] used the attributed grammar to combine statistical and syntactic methods. Three types of edit operations, namely, changes, insertions and deletions of symbols are defined for transforming one string into another. Tsai and Yu[7] proposed attributed string matching with merging, which involves more powerful edit operations. With the merging operation, the recognition rates can be increased. Further, Tsay and Tsai[8] introduced another new edit operation, i.e. split, to the attributed string matching. The above methods

The MS was accepted for publication on 13 April 2006.
* *Corresponding author: W-Y Wu, Department of Industrial Engineering and Management, I-Shou University, Kaohsiung, Taiwan 84008; e-mail: wywu@isu.edu.tw*

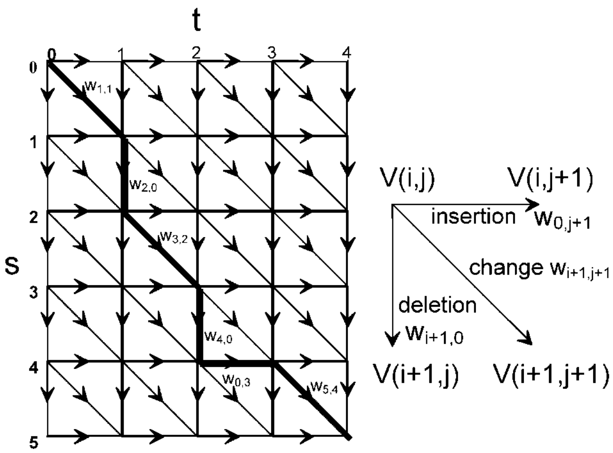
used the lengths and angles of the approximated polygons as features.

Although the recognition rates can be increased by introducing new powerful edit operations, the above methods have three main drawbacks: (1) the edit sequence is too complex; (2) it is not easy to define the edit costs of these two new edit operations; and (3) they need to define a reference line to compute the angles. To solve the above three disadvantages, Maes[4,5] proposed a cyclic string-matching technique for polygonal shape recognition. Only the three conventional edit operations were needed for cyclic string matching. However, the recognition results are not reported in the paper. Wu[9] used the reciprocal of the compactness as the features and developed a cyclic string-matching method for recognizing objects. Wu and Wang[10] proposed a two-stage cyclic string-matching approach to solve the recognition problem. They used both local and global information in determining the similarities of two objects.

Fuzzy set theory has recently become an attractive approach to the pattern recognition problem.[11–13] Owing to the uncertainty principle, fuzzy sets describe the similarity by membership functions instead of crisp concept. Hottori and Takahashi[12] used the nearest-neighbour rule to classify pattern. Keller *et al.*[13] proposed a fuzzy k-nearest neighbour approach to the problem of pattern classification. Zahid *et al.*[14] proposed an unsupervised fuzzy clustering approach to solve the clustering problems. Klein proposed a model based on fuzzy shortest paths.[15] He used a dynamic programming approach to solve the fuzzy shortest path problem. Okada and Soper developed an algorithm based on multiple labelling methods for a multi-criteria shortest path problem.[16]

The present paper proposes a fuzzy string-matching method. After the relevant features are extracted from objects, the edit costs are then presented as fuzzy numbers instead of crisp numbers. First, cyclic fuzzy string matching is performed to find the edit distances between the input and reference objects. The fuzzy edit distances are then ranked by a fuzzy number ranking method.[17] The input shape is classified as the reference shape that has the minimum fuzzy edit distance among all the reference shapes.

Section 2 presents the proposed fuzzy string-matching method and the operations involved in the cyclic string-matching problem. The experimental results are shown in section 3. The concluding remarks are presented in section 4.



1 Edit graph **G** for $|\mathbf{s}|=5$ and $|\mathbf{t}|=4$, shortest path, and its corresponding edit sequence [e_1: ($s_1 \rightarrow t_1$), e_2: ($s_2 \rightarrow \lambda$), e_3: ($s_3 \rightarrow t_2$), e_4: ($s_4 \rightarrow \lambda$), e_5: ($\lambda \rightarrow t_3$), e_6: ($s_5 \rightarrow t_4$)]

2 FUZZY STRING-MATCHING ALGORITHM

2.1 Overview of cyclic string-matching technique

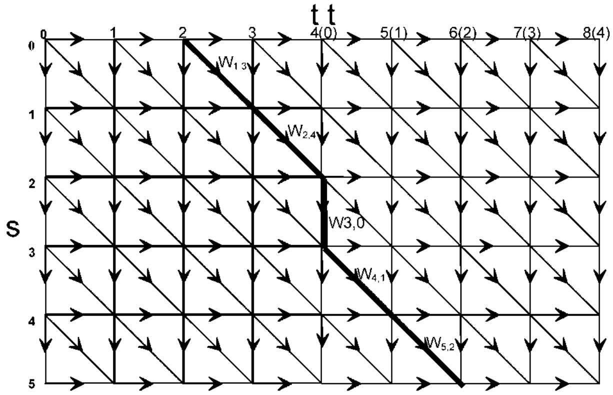
In the following, the notations in cyclic string matching are introduced.[18] The sequence of symbols $s_1 s_2 \ldots s_n$ is called the string **s**. Let $|\mathbf{s}|$ be the length of string **s**. The string with zero length is called the null string, and it is denoted by λ. For convenience, λ also represents the null symbol. Suppose that **s** and **t** are two strings, and let $|\mathbf{s}|=n$ and $|\mathbf{t}|=m$.

The edit graph associated with s and t can then be determined by the weighted graph **G** (see Fig. 1) with vertices $v(i, j)$, for $i=0, 1, \ldots, n$, $j=0, 1, \ldots, m$. The weights of the three types of arcs of **G** can be found by an edit cost function ε.

1. *Insertion*. $[v(i, j), v(i, j+1)]$ with weight $w_{0,j+1}=\varepsilon(\lambda, t_{j+1})$, for $i=0, \ldots, n$, $j=0, \ldots, m-1$.
2. *Deletion*. $[v(i, j), v(i+1, j)]$ with weight $w_{i+1,0}=\varepsilon(s_{i+1},\lambda)$, for $i=0, \ldots, n-1$, $j=0, \ldots, m$.
3. *Change*. $[v(i,j), v(i+1,j+1)]$ with weight $w_{i+1,j+1}=\varepsilon(s_{i+1}, t_{j+1})$, for $i=0, \ldots, n-1$, $j=0, \ldots, m-1$.

The problem of finding a minimum cost edit sequence from **s** to **t** is now equivalent to find a shortest path in **G** from $v(0, 0)$ to $v(n, m)$.[19] This algorithm takes $O(mn)$ times to be an optimum.[20] The thick lines in Fig. 1 show the shortest path and its corresponding edit sequence.

The above method solves the linear string-to-string correction problem by finding the edit distance and its corresponding edit sequence. Hence, the problem is to find the edit distance $\delta([\mathbf{s}], [\mathbf{t}])$ and its



2 Edit graph **H** associated **s** and **tt**, shortest path and edit sequence [e_1: ($s_1 \rightarrow t_3$), e_2: ($s_2 \rightarrow t_4$), e_3: ($s_3 \rightarrow \lambda$), e_4: ($s_4 \rightarrow t_1$), e_5: ($s_5 \rightarrow t_2$)]

corresponding edit sequence, where [**s**] and [**t**] are the cyclic strings of **s** and **t**, respectively. The problem becomes a cyclic string-to-string correction problem. One can assume that $m \leqq n$ without losing the generality, and the strings **s** and **t** are the cyclic strings. Then the edit distance proposed by Maes[5] is

$$\delta([s],[t]) = \min\{\delta(s, \sigma^j(t)) : j = 0, 1, \cdots, m-1\} \qquad (1)$$

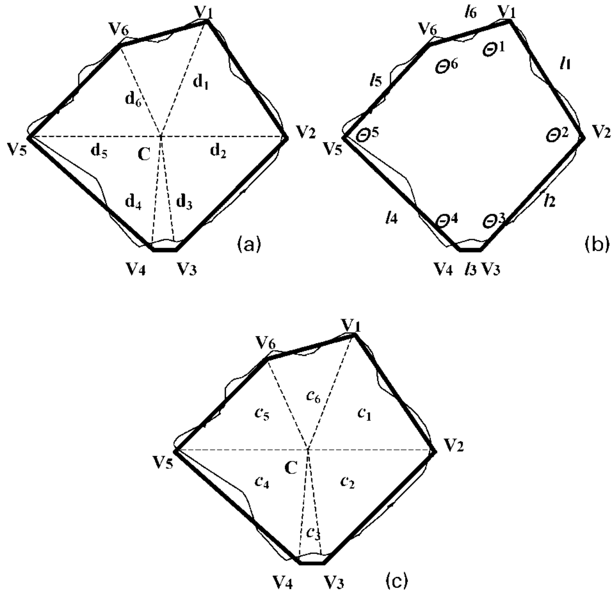
where $\sigma^j(\mathbf{t})$ is the string obtained from **t** after j cyclic shifts.

To calculate the edit distances, let **tt**$= t_1 t_2 \ldots t_m t_1 t_2 \ldots t_m$ be the string which concatenates **t** with itself. One can first construct the edit graph **H** associated with **s** and **tt** (see Fig. 2). Then one can determine the edit distance $\delta[\mathbf{s}, \sigma^j(\mathbf{t})]$ by finding the shortest path from $v(0, j)$ to $v(n, m+j)$, for $j=0, 1, \ldots, m-1$. The minimum edit distance can be found, and its corresponding edit sequence can be identified. This algorithm can be done in $O(mn \log m)$ times.[4]

2.2 Primitive feature extraction

For the robustness of the recognition algorithm, it is desirable that the features should be translation-invariant, rotation-invariant and scale-invariant (TRS-invariant). Since the dominant points on the object boundary are sufficient to represent the shape of an object, the primitive features are derived using the information on the dominant points.

Suppose P_i, $i=1, 2, \ldots, N$, is the ith point with coordinates (x_i, y_i) on the boundary of the object. The dominant points on the boundary of the object can be detected by a dynamic method.[21] The detected dominant points are denoted as V_is, for $i=1, 2, \ldots, M$. Suppose the centroid of the object is $C(x_c, y_c)$.



3 Four different features: (a) relative distances $d_i = |\overline{V_i C}|$; (b) length $l_i = \overline{V_i V_{i+1}}$ and angle $\Theta_i = \angle V_{i-1} V_i V_{i+1}$; (c) modified compactness $c_i = p_i^2/(a_i + e)$

Figure 3 shows the features used for the fuzzy string matching in the experiments.

1. *Relative distance*: The first feature is the relative distance between the vertices and the centroid of the object.[22] This is, $d_i = |\overline{V_i C}|$, where d_i is the ith relative distance.
2. *Length and angle*: The second feature is the length and angle of the approximated polygons. Let l_i and Θ_i be the length of $\overline{V_i V_{i+1}}$ and the angle of $\angle V_{i-1} V_i V_{i+1}$, respectively.
3. *Modified compactness*: The area of the triangle formed by the two adjacent dominant points and the centroid will be zero when these three points are on the same line. Therefore, the modified compactness, c_i, can be defined as[9]

$$c_i = p_i^2/(a_i + e) \qquad (2)$$

where $p_i = |\overline{V_i V_{i+1}}| + |\overline{V_i C}| + |\overline{V_{i+1} C}|$ is the perimeter, a_i is the area of the triangle, and e is a small positive real number.

In order to make the features independent of position, orientation and scaling, these features should be normalized. The features are the relative distances, lengths and angles. These quantities are divided by their maximum value as in the following formula:

$$f_i' = \frac{f_i}{\max_f} \qquad (3)$$

where f_i, f_i' and $\max_f$ are the feature, the normalized feature and the maximum value of features, respectively.

2.3 Fuzzy string matching

Let **I** and **R** be the input shape and the reference shape, respectively. For convenience, the approximated polygons of the shapes are also denoted by **I** and **R**. Furthermore, let n and m be the numbers of vertices for **I** and **R**, respectively. Now, the features are denoted as the symbols, and one can use the strings $\mathbf{s}=s_1s_2\ldots s_n$ and $\mathbf{t}=t_1t_2\ldots t_m$ to represent the input shape and the reference shape, respectively.

The features and the vertices of polygons are both cyclic. The problem of matching two shapes is therefore identical to the cyclic string matching between strings **s** and **t**. Given an edit cost function ε, one can construct the edit graph **H** associated with **s** and **tt** to find the shortest path. The paired symbols of **I** and **R** can then be found by tracing the minimum cost edit sequence. Thus, the matching relation between the vertices of **I** and **R** can be determined.

Suppose that the two ordered sequences $\mathbf{B}_I=(\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_k)$ and $\mathbf{B}_R=(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_k)$ are used to represent the matching result. This means that the $\mathbf{I}_j$th vertex of **I** matches with the $\mathbf{R}_j$th vertex of **R**. In addition, V_i' and W_i' indicate the vertices of the ith matched pair for the two shapes. These two ordered sequences are called the best-matched pair of **I** and **R**, and the minimum edit distance in the string matching stage is called matching cost. Therefore, the dissimilarity of two shapes can then be defined as the matching cost between them.

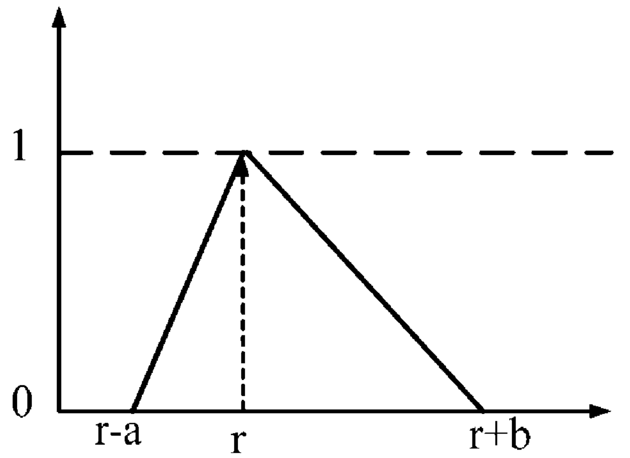
It is important to define a cost function in string matching. Maes[5] gave a cost function by using the lengths and angles as the features.

$$\varepsilon(s_i \to t_j)\begin{cases} |s_i - t_j|, & \text{if } s_i \text{ and } t_j \text{ are angles,} \\ w|s_i - t_j|, & \text{if } s_i \text{ and } t_j \text{ are lengths,} \\ \infty, & \text{otherwise,} \end{cases} \tag{4}$$

$$\varepsilon(s_i \to \lambda) = \varepsilon(\lambda \to s_i) = \begin{cases} |s_i| & \text{if } s_i \text{ is an angle,} \\ w|s_i| & \text{if } s_i \text{ is a length,} \end{cases} \tag{5}$$

where w is a weighting factor.

The above formula is quite straightforward, but the major disadvantage is that one must choose a proper weighting factor w. It is not an easy task to set the weight and it may be very tedious. In order to the weight setting problem, one can define the edit cost function that is suitable for either the



4 Triangular fuzzy number $A=(r, a, b)$

one-dimensional or higher dimensional features.

$$\varepsilon(s_i \to t_j) = \|s_i - t_j\| \tag{6}$$

where ||*|| is the norm of the vector *.

In the conventional string-matching approach, the input shape can then be classified as the reference shape with the minimum matching cost. However, owing to the uncertainty principle, the edit costs can be defined as the triangular fuzzy numbers, as seen in Fig. 4. The triangular fuzzy number $A=(r, a, b)$ is a fuzzy number with membership function f_A defined as

$$f_A(x) = \begin{cases} (x-r+a)/a, & r-a \leqslant x \leqslant r, \\ -(x-r-b)/b, & r \leqslant x \leqslant r+b, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Instead of the crisp definition in equation (6), the edit cost is represented as a fuzzy number in this paper. In fact, the fuzzy edit cost is defined as

$$\varepsilon(s_i \to \underline{t}_j) = (r,a,b) \tag{8}$$

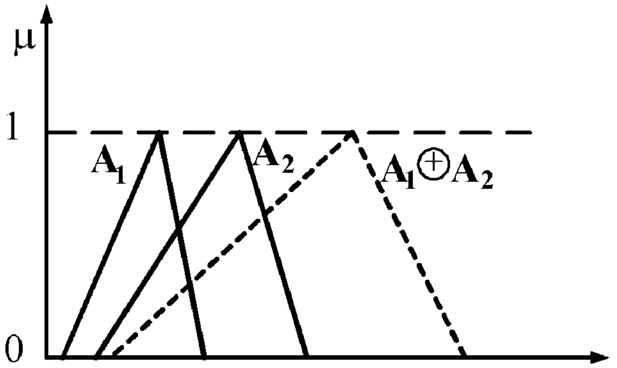
where $r=\|s_i - t_j\|$, and a and b are constants.

Adding two triangular fuzzy numbers (see Fig. 5) can be defined as follows:[16]

$$(r_1,a_1,b_1) \oplus (r_2,a_2,b_2) = (r_1 + r_2,\ a_1 + a_2, b_1 + b_2) \tag{9}$$

The edit costs in equation (9) are defined as fuzzy numbers. They represent the weights of the arcs on the network in Fig. 2. That is, the edit graph is a fuzzy edit graph. Therefore, the edit distances between two shapes are also the fuzzy numbers. In each stage of finding the shortest path in the fuzzy edit graph, the fuzzy edit distances should be ranked to find the minimum fuzzy number. Thus, the fuzzy shortest paths can be found by ranking fuzzy numbers.

The method for ranking fuzzy numbers can be done by a simple method proposed by Liou and



5 Adding two triangular fuzzy numbers

Wang.[17] They used the integral values of the inverse of membership functions to rank fuzzy numbers. For a fuzzy number $A=(r, a, b)$, the total integral value can be constructed from the left integral value, and the right integral value is two values. The total integral value $I_T(A)$ with index of optimism α is then defined as

$$I_T(A)=\alpha I_L(\mathrm{A})+(1-\alpha)I_R(\mathrm{A}) \tag{10}$$

where α is a constant.

The left integral value $I_L(\mathrm{A})$ and the right integral value $I_R(\mathrm{A})$ of a triangular fuzzy number $A=(r, a, b)$ can be found as (see Fig. 6)

$$I_L(A)=r-\frac{a}{2} \tag{11}$$

and

$$I_R(A)=r+\frac{b}{2} \tag{12}$$

From equations (10)–(12), the total integral value of a triangular fuzzy number is defined as

$$I_T(A)=r+(b-(a+b)\alpha)/2 \tag{13}$$

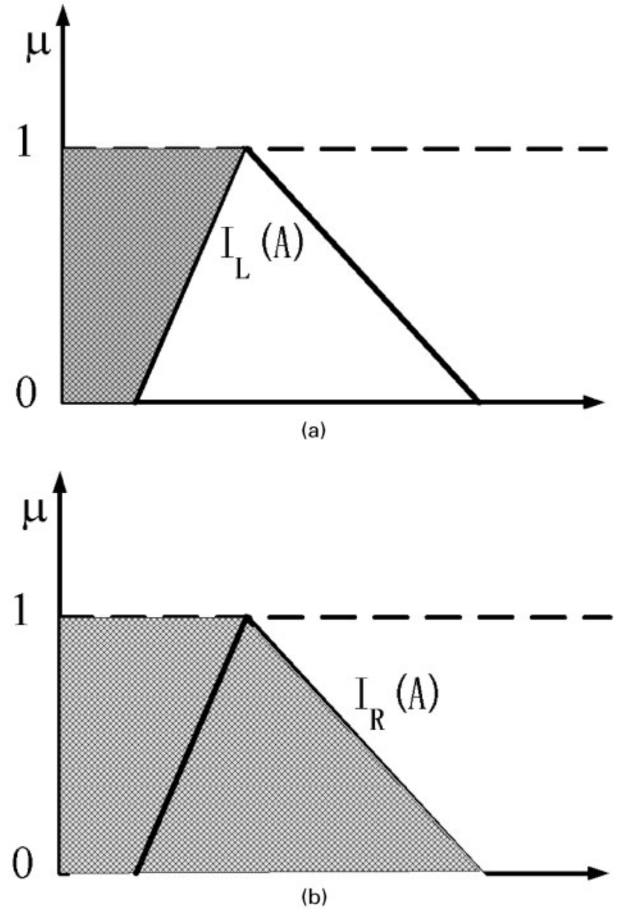
For two triangular fuzzy numbers, the total integral values can be found by equation (13). By comparing the total integral values, one can rank the corresponding fuzzy numbers. For triangular fuzzy numbers, it is very effective to find the inverse of the membership functions. Therefore, the integral values can be found effectively: for example, for two fuzzy numbers $A_1=(12, 11, 17)$ and $A_2=(18, 10, 11)$ as shown in Fig. 7. From the formula in equation (13) one can find that

$$I_T(A_1)=20{\cdot}5-14\alpha \tag{14}$$

and

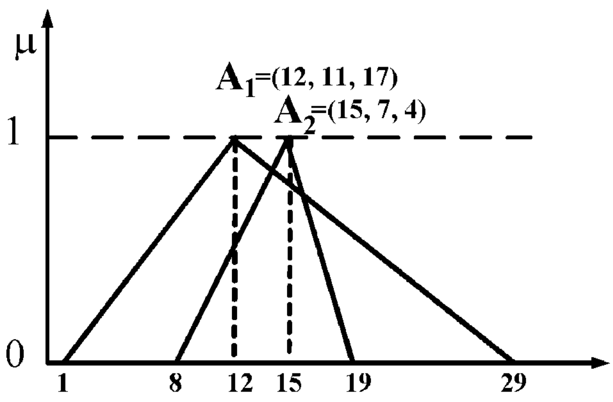
$$I_T(A_2)=17-5{\cdot}5\alpha \tag{15}$$

If $\alpha=1$, then $I_T(A_1)=6.5$ and $I_T(A_2)=11.5$. In this situation, one has a smaller A_1. However, if $\alpha=0$, then $I_T(A_1)=20.5$ and $I_T(A_2)=17$. In this situation, one has a smaller A_2. And A_1 will be equal to A_2 when $\alpha=7/17$.

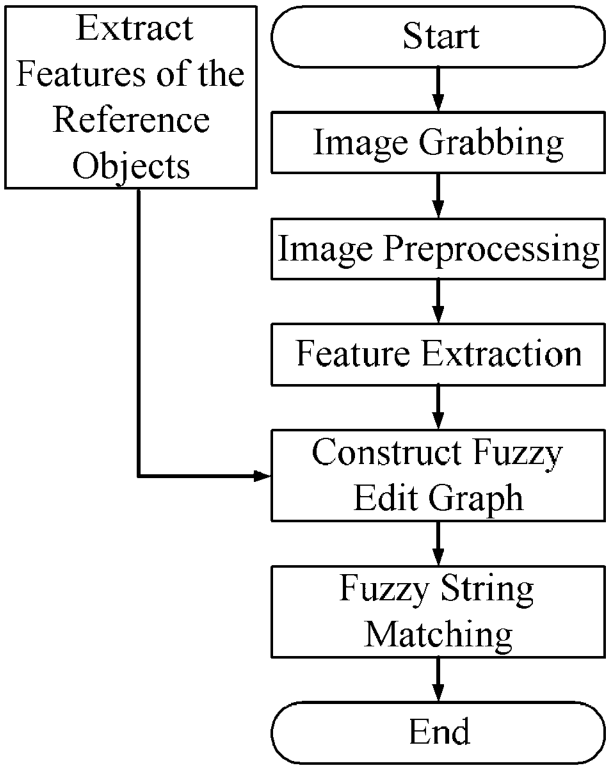


6 Evaluation of fuzzy number using integral value: (a) left integral value; (b) right integral value

Figure 8 shows the flow chart of the fuzzy string-matching approach. The proposed algorithm can be summarized as follows:



7 Example of ranking two triangular fuzzy numbers: $I_T(A_1)=20.5–14\alpha$ and $I_T(A_2)=17–5.5\alpha$



8 Flowchart of proposed fuzzy string-matching algorithm

Step 0. Perform dominant point detection by the curvature-based polygonal approximation method for each reference shape to obtain the vertices of the approximated polygon. Find the features for each of the reference shapes.

Step 1. Perform dominant point detection on the input shape and extract significant features of the input shape.

Step 2. Construct the fuzzy edit graph by using the fuzzy edit costs, which define the three edit operations.

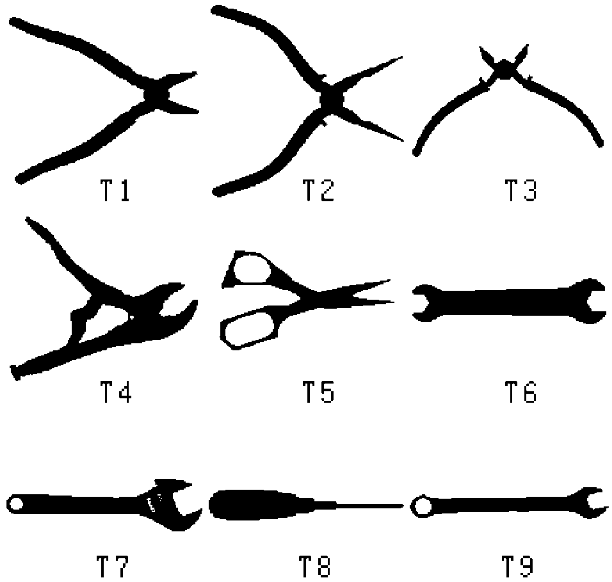
Step 3. Perform fuzzy shortest path algorithm to find the fuzzy edit distances of the input shape and each of the reference shapes.

Step 4. Classify the input shape as the reference shape with the minimum fuzzy edit distance.

Step 5. Repeat steps 1–4 until all input shapes have been recognized.

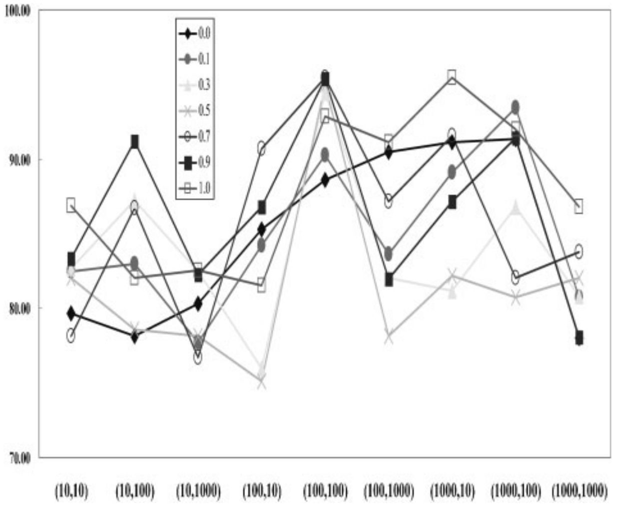
3 EXPERIMENTAL RESULTS

An experiment was designed to test the performance of the proposed method. The relative distances, the lengths and angles, and the modified compactness were used as the features in the experiment. Further, nine different hand tools were tested for evaluating the proposed method (see Fig. 9). As a good object recognition method should be robust for different orientations and scales, for each tool image, there are 32 different orientations and eight different scales conducted in the experiment. The 32 different orientations were arbitrarily chosen by rotating the tools, and the positions of the tools were changed at the same time. For each orientation, seven additional images were generated by reducing the image to 90%, 80%, 70%, 60%, 50%, 40% and 30% of their original density in both x and y dimensions. Thus, 256 ($=32\times 8$) testing images for each tool were used for recognition, and a total of 2304 ($=9\times 256$) testing images were used in the experiment. In addition, the opening levels of tools 1–5 were fixed.



9 Testing images of nine hand tools

The e value in computing the modified compactness is set to 1×10^{-5}. This is used to prevent the case when the area is equal to zero. All of the four features are normalized by their maximum values as in equation (3). The string-matching algorithm (SMA) and the fuzzy string-matching algorithm (FSMA) were applied to each testing image for recognition. Nine settings of the values of (a, b) are used to evaluate the proposed method: (r/10, r/10), (r/10, r/100), (r/10, r/1000), (r/100, r/10), (r/100, r/1000), (r/100, r/1000), (r/1000, r/10), (r/1000, r/100) and (r/1000, r/1000). For convenience, (10, 10) was used instead of (r/10, r/10) to present the values of a and b,



10 Comparison of recognition rates (%) under different settings of (a, b) and α, for hand tools in Fig. 9

and similarly for the others. In addition, seven levels of the values of α were set to be 0.0, 0.1, 0.3, 0.5, 0.7, 0.9 and 1.0, respectively. If a wrong classification is made, an error will be recorded, and the recognition rate can be computed. From the experiment, the recognition rate for the SMA is about 80.51%. For the FSMA, Fig. 10 shows the recognition rates with different values of α and different settings of (a, b). In some cases, the SMA has better performance than the FSMA. However, from Fig. 10, it is seen that the FSMA has the better recognition rates in most of the combinations of α and (a, b). Furthermore, it is seen that it has the best recognition rate when $a=b=r/100$.

The data shown in Table 1 are the recognition rates for the nine hand tools using the SMA and the FSMA with $a=b=r/100$ and different values of α. For tool 7 in Table 1, it can be seen that the recognition rates (only about 45.3%) are relatively low for the SMA. This is because tool 7 is very similar to tool 9. The use of fuzzy edit cost can improve the recognition rates significantly. It was seen that all the recognition rates using the FSMA are larger than 80%. For the SMA, the recognition rates for tools 3, 8 and 9 are less than 80%. However, the recognition rates for tools 3, 8 and 9 are larger than 80%, 86% and 89%, respectively, when using the FSMA. This indicates that the proposed FSMA can significantly improve the recognition rates of tools 3, 8 and 9 when compared with the SMA. Moreover, using the proposed FSMA has an increased recognition rate for all hand tools. This result demonstrates the advantage of the FSMA. Furthermore, to examine the recognition rates displayed in Table 1, it can be seen that the average recognition rates of the SMA, FSMA with α=0.0, 0.1, 0.3, 0.5, 0.7, 0.9 and 1.0 are 80.51%, 88.63%, 90.28%, 94.44%, 95.01%, 95.49%, 95.40% and 92.84%, respectively. It can be seen that that the proposed FSMA with $\alpha=0.7$ has the best recognition rate among the different values of α on average. Overall, it is clear that the use of fuzzy edit costs tends to give a better performance in recognition rate than does crisp edit costs.

In order to validate further the effectiveness of the proposed method, testing of another set of images was conducted. In this experiment, 10 geometric shapes were used for evaluation (see Fig. 11). For each shape, 256 images were also taken in different orientations and scales (32 different orientations and eight different scales) as for the hand tools. The 256 testing images were created in the same way as those in experiment 1. Again, the positions and orientations of the input shapes were arbitrarily chosen. There were 2560 ($=10\times256$) testing images used in the recognition experiment as well. The e value in computing the modified compactness is again set to 1×10^{-5}. For convenience, the values of a and b are

Table 1 Comparison of recognition rates (%) with $a=b=r/100$ for nine hand tools in Fig 9

Method Tool	SMA	FSMA α=0.0	α=0.1	α=0.3	α=0.5	α=0.7	α=0.9	α=1.0
T1	89.5	92.2	95.7	96.1	97.3	98.4	97.7	91.4
T2	95.3	95.3	95.7	98.4	96.5	97.7	98.0	95.3
T3	75.0	80.5	84.4	95.3	94.1	94.5	93.0	91.4
T4	92.6	93.0	94.1	96.1	98.4	96.5	97.7	93.4
T5	90.6	94.1	93.4	96.1	97.3	98.0	97.7	93.4
T6	84.8	86.3	88.3	87.5	95.7	98.0	98.4	96.5
T7	45.3	80.5	81.3	90.6	89.5	93.8	90.2	90.2
T8	73.4	86.3	89.5	94.5	92.6	90.6	92.6	89.5
T9	78.1	89.5	90.2	95.3	93.8	91.8	93.4	94.5
Average	80.51	88.63	90.28	94.44	95.01	95.49	95.40	92.84

Table 2 Comparison of recognition rates (%) with $a=b=r/100$ for 10 geometric shapes in Fig 10

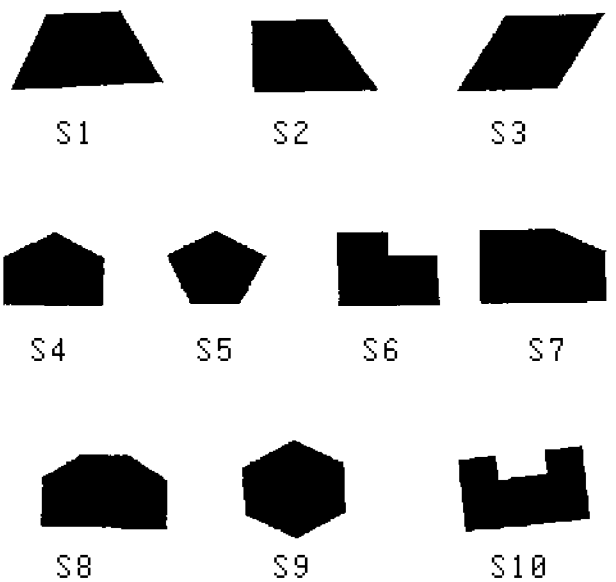
	FSMA						
SMA	α=0.0	α=0.1	α=0.3	α=0.5	α=0.7	α=0.9	α=1.0
91.25	93.44	92.34	96.88	98.63	97.89	97.66	94.38

set to $a=b=r/100$. The comparison results in recognition rates for the SMA and FSMA with $a=b=r/100$ under different values of α are shown in Table 2.

For the corresponding algorithms, the recognition rates listed in Table 2 are larger than those presented in Table 1. All the algorithms perform better on recognition rates for the testing images in Fig. 11 than those in Fig. 9. This is due to the fact that the geometric shapes are simpler than the hand tools. Furthermore, the recognition rates for using the FSMA are better than those of the conventional SMA. However, it is clear that the effect of using fuzzy costs is not very significant for simple shapes such as the testing images in Fig. 11. That is, the proposed FSMA can improve the recognition rates most in recognizing complex objects when compared with conventional SMA.

4 CONCLUSIONS

Attributed string matching is a useful tool for 2D object recognition, but it tends to be affected by uneven segmentation problems. The present paper evaluated the cyclic string-matching technique for 2D object recognition. The fuzzy string-matching approach was used to improve the recognition rates. The edit cost is formulated as a fuzzy number instead of a crisp number. Therefore, the edit distance is also a fuzzy number, and the string-matching problem was equivalent to a fuzzy shortest path problem. The memberships for the input shape with the reference shapes are then determined as fuzzy edit distances. By ranking the fuzzy edit distances, the input shape is classified as the reference shape that has the minimum fuzzy edit distance. Two sets of testing images were evaluated under different levels of orientation and scale in the experiments. The results indicate that using the fuzzy string-matching method gives superior recognition performance to using the conventional method.



11 Testing images of 10 geometric shapes

ACKNOWLEDGEMENT

This paper was partially supported by I-Shou University under Grant No. ISU 95-03-04.

REFERENCES

1 Niemann, H., Hünig, B., Salzbrunn, R. and Schröder, S. A knowledge-based vision system for industrial applications. *Mach. Vision Applic.*, 1990, **3**, 201–229.
2 Marshall, S. Review of shape coding techniques. *Image Vision Comput.*, 1989, **7**, 281–294.
3 Fu, K. S. A step towards unification of syntactic and statistical pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1983, **5**, 200–205.
4 Maes, M. On a cyclic string-to-string correction problem. *Informat. Process. Lett.*, 1990, **35**, 73–78.
5 Maes, M. Polygonal shape recognition using string-matching techniques. *Pattern Recognition*, 1991, **24**, 433–440.
6 Tsai, W. H. and Fu, K. S. Attributed grammar — a tool for combining syntactic and statistical approaches to pattern recognition. *IEEE Trans. Syst. Man Cybernet.*, 1980, **10**, 873–885.
7 Tsai, W. H. and Yu, S. S. Attributed string matching with merging for shape recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1985, **7**, 453–462.

8 Tsay, Y. T. and Tsai, W. H. Model-guided attributed string matching by split-and-merge for shape recognition. *Int. J. Pattern Recognition Artificial Intell.*, 1989, **3**, 159–179.
9 Wu, W. Y. Two-dimensional object recognition through string matching. *Imaging Sci. J.*, 2001, **49**, 213–221.
10 Wu, W. Y. and Wang, M. J. Two-dimensional object recognition through two-stage string matching. *IEEE Trans. Image Process.*, 1999, **8**, 978–981.
11 Bezdek, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*, 1981 (Plenum Press., New York).
12 Hottori, K. and Takahashi, M. A new nearest-neighbor rule in the pattern classification problem, *Pattern Recognition*, 1999, **32**, 425–432.
13 Keller, J. M., Gray, M. R. and Ir, J. G. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybernet.*, 1985, **4**, 580–585.
14 Zahid, N., Abouelala, O. and Essaid, M. A. Unsupervised fuzzy clustering. *Pattern Recognition*, 1999, **20**, 123–129.
15 Klein, C. M. Fuzzy shortest paths. *Fuzzy Sets Syst.*, 1991, **39**, 27–41.
16 Okada, S. and Soper, T. A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Sets Syst.*, 2000, **109**, 129–140.
17 Liou, T. S. and Wang, M. J. Ranking fuzzy numbers with integral value. *Fuzzy Sets Syst.*, 1992, **50**, 247–255.
18 Sankoff, D. and Kruskal, J. B. (eds) *Time Warps, String Edits and Micromolecules: The Theory and Practice of Sequence Comparison*, 1983 (Addison Wesley, Reading, MA).
19 Wagner, R. A. and Fischer, M. J. The string-to-string correction problem. *J. ACM*, 1974, **21**, 168–173.
20 Wong, C. K. and Chandra, A. K. Bounds for the string-editing problem, *J. ACM*, 1976, **23**, 13–16.
21 Wu, W. Y. A dynamic method for dominant point detection. *Graph. Models*, 2003, **64**(5), 304–315.
22 Chang, C. C., Hwang, S. M. and Buehrer, D. J. A shape recognition scheme based on relative distances of feature points from the centroid. *Pattern Recognition*, 1991, **24**, 1053–1063.