Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 6 Issue 4, April - 2020

On Fuzzy String Matching for Detecting Shapes

Fuzzy String Matching

Wen-Yen Wu
Department of Industrial Management
I-Shou University
Kaohsiung, Taiwan
wywu@isu.edu.tw

***Abstract*—In this paper, we propose a fuzzy string matching approach to solve the pattern recognition problems. The edit cost is presented as a fuzzy number and the string matching problem with fuzzy edit cost was then equivalent to a shortest path problem with fuzzy weights. By ranking the fuzzy numbers, the shape is classified as the reference shape that has the minimum fuzzy distance.**

Keywords—pattern recognition, string matching, shortest path, fuzzy numbers, ranking.

I. INTRODUCTION

It has been agreed that representation and matching are the two major problems involved in pattern recognition. Many methods have been proposed, for example, Fourier descriptor, Gaussian decomposition, polar transformation, invariant moments, chord length distribution, structuring elements, auto-regressive model, energy-function, and primitives are the well-known shape representation methods [5, 12]. Fu [1] proposed a combined approach that has the merits of both the statistical approaches and the syntactic approaches. Further, some edit operations have been used to improve the attributed string matching approaches [7-10]. However, the above methods need a reference line to compute angles. To solve the problem, Maes [4] proposed a cyclic string matching technique for polygonal shape recognition.

Fuzzy sets theory becomes an attractive approach in the pattern recognition problem recently. Due to the uncertainty principle, fuzzy sets describe the similarity by the membership functions instead of crisper function. Klein [2] proposed a model based on fuzzy shortest paths. He used a dynamic programming approach to solve the fuzzy shortest path problem. Okada and Soper [6] developed an algorithm based on the multiple labeling methods for a multi-criteria shortest path problem.

In this paper, the fuzzy edit distances for the input shape with the reference shapes are first determined. The input shape is classified as the reference shape that has the minimum fuzzy edit distance among all the reference shapes.

II. FUZZY STRING MATCHING

A. String Matching Technique

The sequence of symbols $s_1s_2...s_n$ is called the *string s* and |*s*| is the length of the string *s*. The string with zero length is called the *null string*, and it is denoted asλ. Suppose that *s* and *t* are two strings and let |*s*|=n and |*t*|=m. The *edit network associated with s and t* can be constructed by defining three operations that are insertion, deletion, and change.

Let *S* and *T* represent two shapes and they can be expressed as the strings *s* and *t*, respectively. The paired symbols of *S* and *T* can then be found by tracing the minimum cost edit sequence. Thus, the matching relation between the vertices of *S* and *T* can be determined. For instance, we can use the two ordered sequences $B_s = (S_1, S_2,\dots, S_k)$ and $B_t=(T_1, T_2,\dots, T_k)$ to represent the matching result. The two ordered sequences mean that the S_j th vertex of *S* matches with the T_j th vertex of *T*. These two ordered sequences are called the *best-matched pair* of *S* and *T*.

The above matching process is shown by an example as in Fig. 1. Two objects shown in Figs. 1(a) and 1(b) have 5 and 4 vertices, respectively. Therefore, they can be expressed as $s= s_1s_2...s_5$ and $t=t_1t_2...t_4$. We can find that the shortest path and its corresponding edit sequence (e_1: ($s_1\to t_1$), e_2: ($s_2\to\lambda$), e_3: ($s_3\to t_2$), e_4: ($s_4\to\lambda$), e_5: ($\lambda\to t_3$), e_6: ($s_5\to t_4$)).

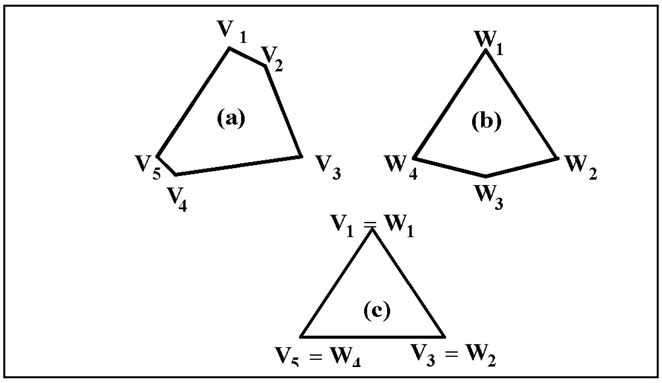

Fig. 1. *Matching two objects: (a) object with 5 vertices, (b) object with 4 vertices, and (c) the best-matched result between (a) and (b).*

The edit sequence e_1: ($s_1\to t_1$) means that a change from vertex 1 of *S* to vertex 1 of *T*. That is, vertex 1 of *S* should be matched to vertex 1 of *T*. The edit sequence e_2: ($s_2\to\lambda$) presents that a deletion from vertex 2 of *S*. That is, vertex 2 of *S* is matched to a null symbol. The edit sequence e_3: ($s_3\to t_2$) indicates that vertex 3 of *S* should be matched to vertex 2 of *T*.

The edit sequence e_4: ($s_4 \rightarrow \lambda$) matches vertex 4 of *S* to a null symbol. The edit sequence e_5: ($\lambda \rightarrow t_3$) presents that an insertion for vertex 3 of *T*. That is, a null symbol is matched to vertex 2 of *S*. The edit sequence e_6: ($s_5 \rightarrow t_4$) matches vertex 3 of *S* to vertex 2 of *T*. Therefore, we can obtain the best matched pair is B_s=(1, 3, 5) and B_t=(1, 2, 4). That is, vertices 1, 3, and 5 in Fig. 1(a) should be matched to vertices 1, 2, and 4 in Fig. 1(b), respectively. The best-matched result is shown in Fig. 1(c).

However, the above method fails to find the best match between two shapes while their orientations change. In order to solve the problem of orientation changing, we can define the *cyclic string [s]* as the set of strings shifted from the string *s*. Let $\sigma^j(s)$ be the string obtained from *s* after j cyclic shifts. That is, $\sigma^j(s) = s_{j+1}s_{j+2}\ldots s_n s_1 \ldots s_j$. The cyclic string [*s*] is the set $\{\sigma^j(s) |\ j=0, 1, 2, \ldots, n-1\}$.

We can assume that $m \leqq n$ without losing the generality. The edit distance between [*s*] and [*t*] is then determined by finding the minimum among the m edit distances $\delta(s,\sigma^j(t))$, j=0, 1, …, m-1. That is, the edit distance proposed by Maes [4] is

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

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

To simplify the computation of cyclic string matching, we can first construct the edit network *H* associated with *s* and *tt* (see Fig. 2), where $tt = t_1t_2\ldots t_m t_1 t_2 \ldots t_m$ is the string which concatenates *t* with itself. Therefore, the edit distance between [*s*] and [*t*] can then determined by finding the minimum edit distance with complexity O(mn log m) time [4].

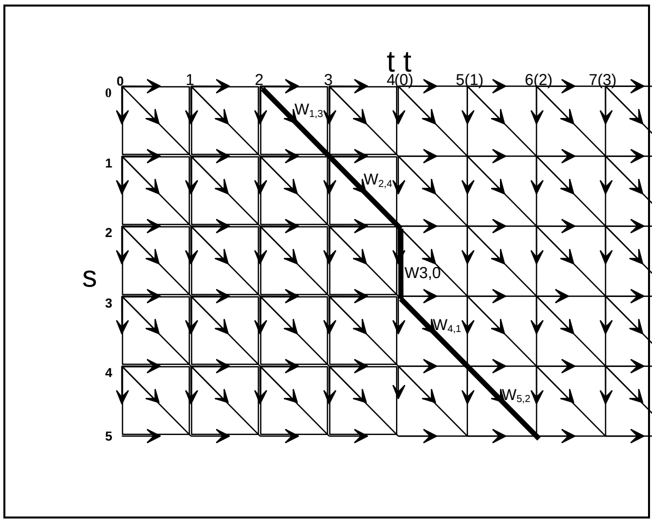

Fig. 2. The edit network H associated s and tt, and the shortest path.

B. Fuzzy Edit Distances

Once the nonlinear segments have been determined, it needs to find the locations of the circular objects. We use the geometric properties of a circle to derive a method for finding center and radius. Suppose that line L is the perpendicular bisector of a chord on a circle as seen in Fig. 5, line L will pass through the center C.

It is important to define a cost function in string matching. Suppose that the value of λis zero. We can define the edit cost function that is both suitable for the one-dimensional or higher dimensional features.

$$\varepsilon(s_i \rightarrow t_j) = \left\| s_i - t_j \right\|, \quad (2)$$

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

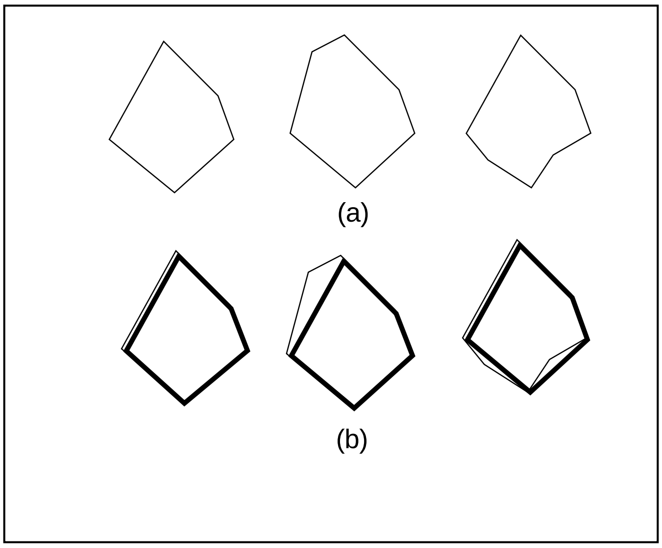

Fig. 3. *Example of segmentation inconsistency problem: (a) three objects, and (b) the matched vertices..*

The edit costs are 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

$$f_A(x) = \begin{cases} (x-r+a)/a, & r-a \le x \le r \\ -(x-r-b)/b, & r \le x \le r+b \\ 0, & otherwise \end{cases} \quad (3)$$

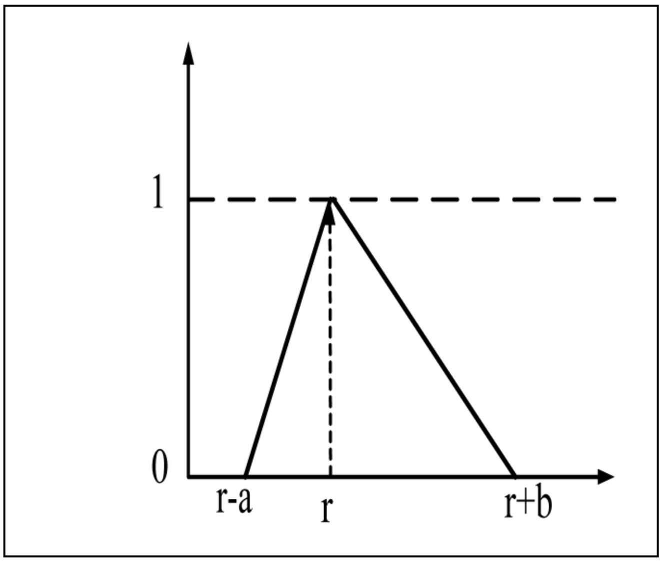

Fig. 4. *A triangular fuzzy number A=(r, a, b).*

Instead of the crisper definition in Eq. 2, the edit cost is represented as a fuzzy number in this paper. Therefore, weights for three types of arcs: insertion, deletion, and change operations, are also represented as fuzzy numbers. In fact, we can define the fuzzy edit cost as a triangular fuzzy number

$$\varepsilon(s_i \rightarrow t_j)=(r, a, b) \quad (4)$$

where $r=\left\|s_i - t_j\right\|$ and a and b are constants; for $s_i=\lambda$, it indicates an insertion operation; $t_j=\lambda$ is a deletion operation; otherwise, it is a change operation.

In addition, the summation of two triangular fuzzy numbers is again a fuzzy number as defined in [6]. Since the weights are fuzzy numbers and the edit distances are the summation of the fuzzy weights, the edit distances are also the fuzzy numbers. Further, the fuzzy shortest paths of an edit network can be found by ranking fuzzy numbers. The method for ranking fuzzy numbers can be done by a simple method proposed by Liou and Wang [3]. 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 are two values. The total integral value $I_T(A)$ with index of optimism α is then defined as

$$I_T(A)= \alpha\, I_L(A) +(1-\alpha)\, I_R(A) \quad (5)$$

where $I_L(A)$ and $I_L(A)$ are the left integral value and the right integral value, respectively.

The total integral value is defined as

$$I_T(A)= r+(b-(a+b)\alpha)/2 \quad (6)$$

For two triangular fuzzy numbers, they can be ranked by finding their integral values for the inverses of the membership functions. For triangular fuzzy numbers, it is very effective to find the inverse of the membership functions. Therefore, the integral values can be found effectively.

III. EXPERIMENTAL RESULTS

Nine different tools were used for evaluation in the experiment (see Fig. 5). A good shape matching approach should be robust under different orientations and scales. For each tool image, there were 16 different orientations and 4 different scales conducted. The 16 different orientations were arbitrary chosen by rotating the tools and the positions of the tools were changed at the same time. For each orientation, three additional images were generated by reducing the image to 90%, 70%, and 50% of original in both the horizontal and vertical directions. Thus, 64 (=16×4) testing images for each tool were used for matching, and there were a total of 576(=9×64) testing images.

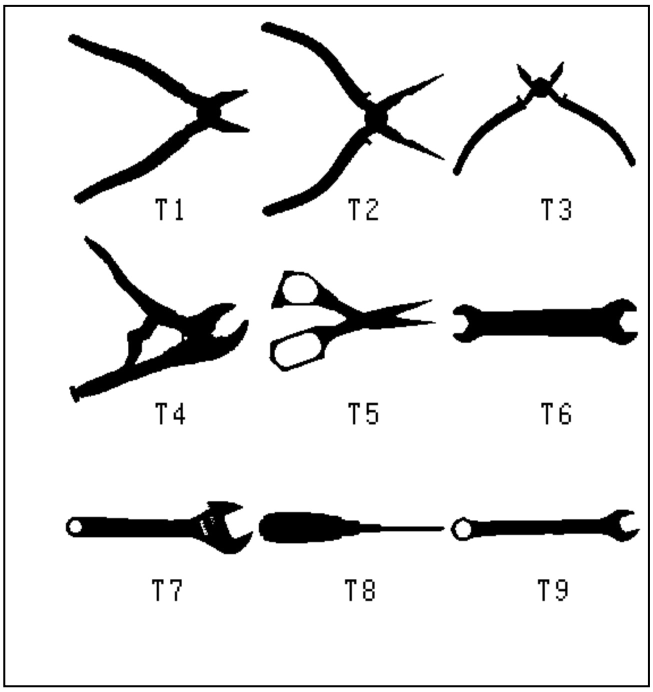

Fig. 5. *The nine hand tools.*

From dominant points [11], the modified compactness can be computed and is defined as

$$c_i= p_i^2/(a_i+e), \quad (7)$$

where p_i 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 to be independent of position, orientation, and scaling, these features should be normalized. In this paper, all the features are divided by the maximum value. The e value in computing the modified compactness is set to 0.00001. The matching algorithm was applied to each testing image for matching. If a wrong classification was made, an error was recorded. And the recognition rate can be computed. For the proposed fuzzy string matching algorithm, both the values of a and b are set to r/40 in Eq. (4). Three values of the index α were

0.3, 0.5, and 0.7. The data in Table 1 are the recognition rates for string matching algorithm (**SMA**) and fuzzy string matching algorithm. (**FSMA**).

It is seen that the fuzzy string matching algorithm has better recognition rates than that of the string matching algorithm. Especially, for tool 7 in Table 1, it can be seen that the recognition rates are very low.

TABLE I. THE RESULTS OF RECOGNITION RATES FOR SMA AND FSMA (%).

Tool	**Method**			
	SMA	***FSMA* (*α*=0.3)**	***FSMA* (*α*=0.5)**	***FSMA* (*α*=0.7)**
1	94	99	98	99
2	97	100	99	97
3	79	95	95	94
4	95	100	99	98
5	93	100	98	97
6	88	96	97	98
7	49	90	89	93
8	79	94	92	90
9	80	95	93	92
Average	83.8	97.6	95.6	95.3
Processing Time	0.13 sec.	0.21 sec.		

IV. CONCLUSIONS

String matching is a useful tool for two-dimensional object recognition, but it tends to be affected by uneven segmentation problem. In this paper, we propose a fuzzy string matching method to solve the shape matching problems. The edit cost is formulated as a fuzzy number instead of a real number. Therefore, the edit distance is also a fuzzy number. The string matching problem with fuzzy edit costs was then equivalent to a shortest path problem with fuzzy weights. 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. The experimental results indicate that the use of the fuzzy string matching method has superior recognition performance to the use of the conventional one.

REFERENCES

[1] K. S. Fu, "A step towards unification of syntactic and statistical pattern recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 5, pp. 200-205, 1983.

[2] C. M. Klein, "Fuzzy shortest paths," Fuzzy Sets and Systems, vol. 39, pp. 27-41, 1991.

[3] T. S. Liou and M. J. Wang,"Ranking fuzzy numbers with integral value," Fuzzy Sets and Systems, vol. 50, pp.247-255, 1992.

[4] M. Maes, "Polygonal shape recognition using string-matching techniques," Pattern Recognition, vol. 24, pp. 433-440, 1991.

[5] S. Marshall, "Review of shape coding techniques," Image and Vision Computing, vol. 7, pp. 281-294, 1989.

[6] S. Okada and T. Soper, "A shortest path problem on a network with fuzzy arc lengths," Fuzzy Sets and Systems, vol. 109, pp. 129-140, 2000.

[7] W. H. Tsai and K. S. Fu, "Attributed grammar- a tool for combining syntactic and statistical approaches to pattern recognition," IEEE Trans. System, Man, and Cybernetics, vol. 10, pp. 873-885, 1980.

[8] 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.

[9] 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.

[10] E. Ukkonen, "Algorithms for approximate string matching," Information and Control, vol. 64, pp. 100-118, 1985.

[11] W. Y. Wu, "An adaptive method for detecting dominant points," Pattern Recognition, vol. 36, no. 10, pp. 2231-2237, 2003.

[12] W. Y. Wu and M. J. Wang, "Two-dimensional object recognition through two-stage string matching," IEEE Trans. Image Processing, vol. 8, pp. 978-981, 1999.