SHAPE RECOGNITION USING GENETIC ALGORITHMS

Ender Ozcan and Chilukuri K. Mohan*
School of Computer and Information Science
Syracuse University, Syracuse, NY 13244-4100, U.S.A.

July 8, 1996

Abstract

Shape recognition is a challenging task when shapes overlap, forming noisy, occluded, partial shapes. This paper uses a genetic algorithm for matching input shapes with model shapes described in terms of features such as line segments and angles (extracted using traditional algorithms). The quality of matching is gauged using a measure derived from attributed shape grammars [12, 13]. Preliminary results, using shapes with about 30 features each, are extremely encouraging.

1 Introduction

Shape recognition is an important problem studied by numerous researchers. Given a set of *model* shapes and an input shape, the problem consists of identifying which of the model shapes are present in the input shape. For example, most robotics applications for part inspection and VLSI design involve locating and identifying objects. For shapes of objects that are occluded, or touch or overlap with other objects, a flexible shape recognition algorithm is needed to identify overlapping shapes, making use of incomplete information.

The Generalized Hough Transform (GHT) [3] is one technique widely used in shape matching algorithms. Although this approach can be used for partial shape matching, the time-consuming nature of GHT makes it impossible to use it for real-time applications with large data. Turney *et al.* [14] suggested improvements to the GHT approach. The performance of such algorithms depend on how shapes are represented, and the extraction of combinations of global and local features of the model shapes. Model shape features are compared with features extracted from the input shape [1, 4, 8, 10]. Neural networks also provide powerful tools for solving the shape recognition problem [10, 11, 16]. Yang *et al.* [15] proposed an algorithm which uses the attributed string representation for shapes and a two-layer network of Hopfield-like modules.

In this paper, a new approach is proposed for partial shape matching which utilizes genetic algorithms (GAs) and attributed string representations [12, 13]. The GA gave robust matching results for the test shapes, providing rotation and size independence. Di Ianni [5] has also applied GAs for matching shapes but the results they obtained were not encouraging, possibly because of using raw pixel arrays rather than shape features. Bala and Wechsler [2] use GAs to develop morphological operators that can discriminate among classes containing different shapes, not directly for shape matching.

Section 2 is an overview of attributed string representation. The new algorithm is introduced in Section 3. Section 4 describes experimental results, and the conclusions are presented in Section 5.

2 Attributed Strings

Attributed strings are used for the representation of shapes. Each shape is a polygon, made up of line segments. A string of features

$$(x_1, x_2, ..., x_i, ..., x_n)$$

is used to represent each shape. Each feature $x_i = (l_i, \theta_i)$ is formed of two attributes: the length l_i of the corresponding line segment, and the relative angle θ_i it forms with the preceding line segment x_{i-1}.

*Author for correspondence: C. K. Mohan, 2-120 Center for Science and Technology, School of CIS, Syracuse University, Syracuse, NY 13244-4100; 315-443-2322/(fax) 1122; eozcan/mohan@top.cis.syr.edu

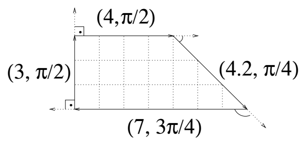

Figure 1: Attributed string representation of a shape.

Example: The quadrilateral in Figure 1 has the following attributed string representation:

$$\Big((4, \frac{\pi}{2}),\ (4.2, \frac{\pi}{4}),\ (7, \frac{3\pi}{4}),\ (3, \frac{\pi}{2})\Big)$$

To provide size invariance, we propose a representation in which each feature's length is normalized by the preceding feature's length attribute:

$$l'_i = \frac{l_i}{l_{i-1}}$$

Lengths are not normalized using the total length of a shape, since overlapped shapes may have a total length different from the sum of the lengths of the constituent shapes. This would cause the length information to be unreliable during the matching process.

Example: After normalization, the quadrilateral in Figure 1 is represented as:

$$\Big((1.33, \frac{\pi}{2}),\ (1.05, \frac{\pi}{4}),\ (1.67, \frac{3\pi}{4}),\ (0.43, \frac{\pi}{2})\Big)$$

3 Genetic Algorithm for Shape Matching

Genetic Algorithms (GAs) were introduced by J. Holland [7], and have been used to solve many difficult problems [6]. We assume familiarity with GA notation.

Given an input shape with n features and S model shapes with a total number of M features, we can cast partial shape matching as the problem of finding n matching features from M features. The search space is immense, of size ranging up to M^n, since the input shape may contain multiple partial instances of the same model shape.

3.1 Representation

Each individual maps each input shape feature to one of the model shape features. Each individual is represented as a list of lists in which each entry has two slots, one showing the matching model shape and the other showing its corresponding feature. We use the following notation:

- For each feature, the normalized lengths and angles are obtained by the functions l and θ.
- Input shape $I = (I_1, I_2, ..., I_n)$, where I_p is the pth feature, consisting of attributes $(l(I_p), \theta(I_p))$. Size of input shape $|I| = n$, the number of features in I.
- The model shapes are $M_1, M_2, ..., M_S$, where

  $$M_j = (M_{j,1}, M_{j,2}, ..., M_{j,m_j}),$$

  where $M_{j,r}$ is the rth feature of the jth model shape, consisting of attributes $(l(M_{j,r}), \theta(M_{j,r}))$. Size of the jth model shape $|M_j| = m_j$.
- Each individual $P = (P_1, P_2, ..., P_k, ..., P_n)$ corresponds to a mapping μ from input shape features to model shape features such that $P_k = \mu(I_k) = m_{j,i}$, where $1 \leq k \leq n$, $1 \leq j \leq S$, and $1 \leq i \leq m_j$.

Example: The following is an example individual:

$$[(2,3), (2,4), (2,5), (3,4) \ldots]$$

The first feature of the input shape is mapped by this individual to (2,3), the third feature of the second model shape.

3.2 Initialization

The initial population is a set of randomly chosen individuals. Each individual is obtained as follows. First, an input shape feature I_k is randomly chosen, and mapped to a randomly chosen feature $M_{j,r}$ in a randomly chosen model shape M_j. Features $I_{k+1}, I_{k+2}, \ldots$ are then mapped to $M_{j,r+1}, M_{j,r+2}$, etc., with the features of M_j being traversed in circular manner. After assigning all possible features of M_j, if any input shape features remain unassigned, then the next unassigned input shape feature I_{k+m_j} is mapped to a new randomly chosen model shape feature, $M_{j',r'}$, followed by mapping as many as possible of the remaining input shape features $I_{k+m_j+1}, I_{k+m_j+2}, \ldots$ to $M_{j',r'+1}, M_{j',r'+2}, \ldots$. This process is repeated until all input shape features have been assigned, tracing through input shape features in circular sequence, i.e., I_1 is the feature presumed to follow I_n.

The size of the population is kept *twice* the size of the number of features of the input shape.

3.3 Fitness

The fitness of an individual describes how well each feature of the input shape matches with the model shape feature to which it is matched. Fitness also depends on the degree of consistency between model features to which neighboring shape features are mapped.

Fitness is calculated by testing the compatibility of the input shape features and the corresponding model features to which an individual maps them. The difference (dissimilarity) between input shape feature I_k and model feature $f_k = \mu(I_k)$ is measured by means of a distance function $d(I_k, \mu(I_k))$, defined below.

$$d(I_k, \mu(I_k)) = d_\theta(I_k, \mu(I_k)) + d_l(I_k, \mu(I_k))$$

$$d_\theta(I_k, \mu(I_k)) = c_\theta \text{ abs}(\theta(I_k) - \theta(\mu(I_k)))$$

The constant c_θ is chosen in our experiments so that differences upto $\pi/18$ are considered negligible. If two successive features are not mapped by an individual to the same model shape, we use $d(I_k, \mu(I_k)) = \infty$. Otherwise,

$$d_l(I_k, \mu(I_k)) = \frac{\text{abs}(l(I_k) - l(\mu(I_k)))}{\max(l(I_k), l(\mu(I_k)))}$$

The distance between two features is compared with a threshold value. If the distance is small, then the corresponding primitive is marked as matched. Otherwise, the features are not considered to have matched:

$$Matched(I_k, \mu(I_k)) = \begin{cases} 1 & \text{if } d(I_k, \mu(I_k)) < \epsilon \\ 0 & otherwise \end{cases}$$

The fitness function penalizes the number of models to which input shape features are mapped by an individual. Fitness is calculated using the following formula:

$$\begin{aligned} Fitness \;=\; - \;\; & (\text{Number of partial shapes } + \\ & \text{Number of unmatched features} \\ & \text{of input shape}) \end{aligned}$$

3.4 Selection Mechanism

A linear ranking strategy was used during reproduction, with the best individual being allocated roughly five times more offspring than the worst individual. An elitist survival selection mechanism was also used: the best two third (67%) of all individuals in a generation are allowed to survive into the next generation. These choices were made arbitrarily.

3.5 Operators

Crossover and mutation generate new individuals, and hill climbing is used to improve the solutions obtained.

- Traditional one-point crossover (1PTX) is applied to the individuals, producing two children.

- Mutation randomly chooses points on an individual, and maps input shape features to the features of randomly chosen model shapes. All the individuals are mutated with 30% mutation rate. This rate yielded the best performance in an experiment although the algorithm was successful even when lower mutation rates were used.

- Hill climbing is applied to improve the mappings obtained at the borders between feature sequences mapped to different model shapes. Each hill climbing step attempts to improve the fitness of an individual by shifting the "intersection point" (between feature sequences mapped to different model shapes) in either direction, replacing the relevant component by the most appropriate feature from the model to which neighboring shape features are mapped. For instance, if $\mu(I_k) = M_{j,i}$ and $\mu(I_{k+1}) = M_{j',i'}$, hill climbing changes either $\mu(I_k)$ to $M_{j',i'-1}$ or $\mu(I_{k+1})$ to $M_{j,i+1}$, in case such a change improves fitness.

4 Experimental Results

In our experiments, all shapes are polygons described as a sequence of adjacent features. Three sets of model shapes were used. The first set is shown in Figure 2, the second set is shown in Figure 3 and the third set is shown in Figure 4. The total number of features of first, second and third sets of model shapes were 77, 131 and 436 respectively. All of the input shapes used in the experiments can be found in Figure 5. In the figures, all the input and model shapes are uniquely numbered. We use the prefixes "i" and "m" to refer to input and model shapes, respectively, e.g., "i2" is the second input shape.

Input shapes i0, i1, i2, i3 and i4 constitute the first set of input shapes in which the algorithm attempts to search for model shapes of the first set; i5 and i6 constitute the second set; and i7, i8 and i9 constitute the third set. Except i6, all of the input shapes are obtained by overlapping two or three model shapes; in i6, the model shapes m8 and m9 are touching each other. The shape i2 is formed by overlapping two occurrences of the same model shape m5. Also, i5 is formed of m7 and m3, with m3 scaled to a size about 3 times larger than its original size; a similar case occurs in i7. The number of features of input shapes and the corresponding overlapping model shape labels are shown in Table 1.

Several experiments were first performed to determine appropriate parameter values of the GA. For instance, experiments were conducted with input shape i3, varying mutation rate in the presence and absence of crossover and hill climbing. The experimental results, indicated in Table 2, show that each operator does not perform satisfactorily by itself. The algorithm works best if all of the three operators (crossover, mutation and hill climbing) are used. In such a case, the algorithm converged in tens of generations to the correct matching result (Figure 6).

Experimental results are summarized in Table 3. In the first set of experiments, the input shape contains more features than each of the overlapping model shapes. In the second set of experiments, the input shape contains fewer features than one of the overlapping model shapes used. In the third set of experiments, a larger database is used.

Since our algorithm performs size invariant matching, input shapes such as i8 can also be matched successfully. In all cases, the algorithm succeeded in finding the model shapes, even if the input shape contained overlapped instances of the same model shape, as in i2.

Each test was repeated 100 times for i0-i6 and 50 times for i7-i9. Except in the case of i4, the correct result was found in every experiment with every input shape. For i4, 1 out of 100 runs converged to a wrong matching result.

As shown in Table 3, correct results are reached in 8-21 generations, on average. The maximum number of generations, in each run, was fixed to be twice as large as the total number of model shape features. The number of hill climbing moves is larger, depending on the size of the input shape and the number of model shapes. Average system run times were less than a minute in all experiments, on a Sun workstation.

If a model shape is symmetric, and the input shape was obtained by overlapping such a symmetric shape with another model shape, there might be more than one possible solution, of equally good quality. This phenomenon was observed in the case of i2 and i5.

5 Conclusion

A new approach for shape recognition is developed, which combines genetic algorithms and attributed string representations. Outline features of shapes are represented using attributed strings. Each line segment is associated with a feature of two attributes: length and angle. Relative lengths and angles are used for size invariance. Experimental results show that the GA has been successful in shape-matching experiments attempted so far.

This new approach fro shape matching has several advantages. Firstly, it is much more computationally efficient than exhaustive search algorithms. Secondly, it is space-efficient compared to neural network models, most of which have large memory requirements. The algorithm is fast, and explores a relatively small number of elements of the search space. The use of many-individual populations and operations such as crossover overcomes the primary problem faced by a greedy algorithm which tends to get stuck in locally optimal solutions.

So far, only polygons without holes have been used as model shapes and input shapes. Current work involves applying this promising algorithm to larger databases with many more model shapes, and analyzing input shapes containing a larger number of overlapping model shapes.

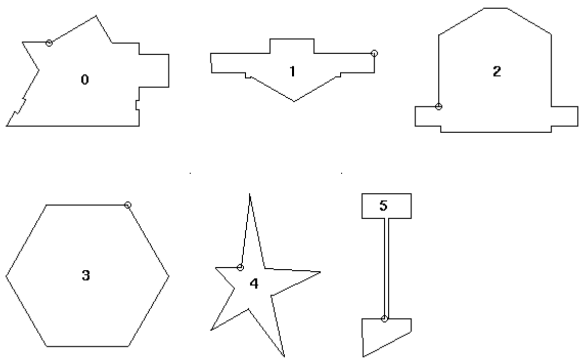

Figure 2: Normalized model shapes used for the first set of experiments.

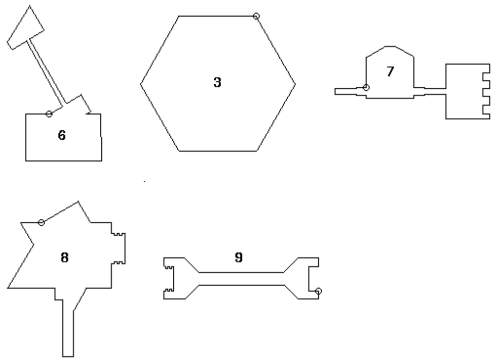

Figure 3: Normalized model shapes used for the second set of experiments.

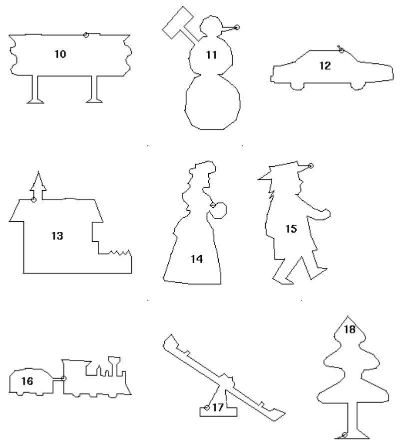

Figure 4: Normalized model shapes used for the third set of experiments.

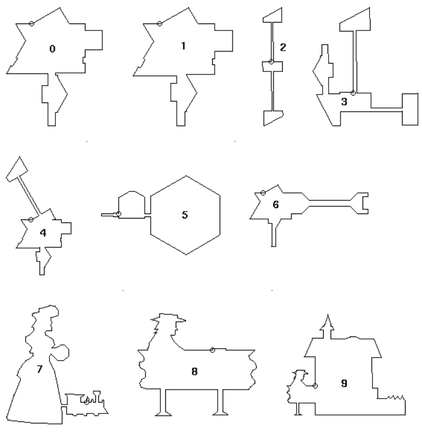

Figure 5: Normalized input shapes used for the experiments.

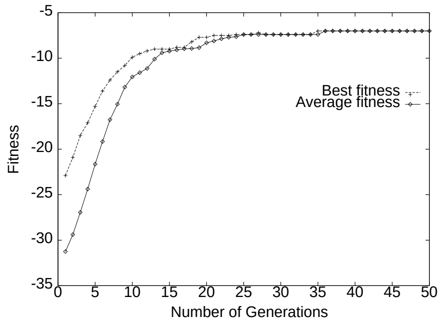

Figure 6: Average fitness and best fitness, plotted against number of generations, for input shape i3, averaged over 10 runs.

Table 1: Composition of input shapes: number of features and model shapes used to obtain each input shape.

Shape label	No. of features	Model shapes overlapped
i0	35	m0, m1
i1	32	m0, m1
i2	20	m5, m5
i3	37	m1, m5, m5
i4	41	m0, m1, m5
i5	24	m3, m7
i6	29	m8, m9
i7	113	m14, m16
i8	71	m10, m15
i9	86	m10, m15, m13

Table 2: Matching frequency (rate of discovery of correct solution) and average number of generations, in 25 runs for i3, varying the GA operators (Maximum number of generations is 154).

GA Operators	Match. Freq.	Avr. Gen.	Std. Dev.
Mut(0.3), HC, 1PTX	1.00	9	4
Mut(0.3), HC	0.00	154	-
Mut(0.3), 1PTX	0.32	154	-
HC, 1PTX	0.28	107	68
HC	0.00	154	-
Mut(0.3)	0.00	154	-
1PTX	0.00	154	-

Table 3: Average number of generations, fitness values and hill-climbing steps, in over 50 runs.

Shp. Lab.	Avr. Gen.	Std. Dev.	Avr. Fit.	Std. Dev.	Avr. HC	Std. Dev.
i0	9	5	-22.51	3.23	1260	640
i1	9	5	-20.67	3.09	1149	557
i2	8	7	-13.86	2.23	552	412
i3	12	7	-23.39	2.85	1601	823
i4	16	10	-24.69	3.79	2262	1379
i5	13	10	-17.09	1.83	833	478
i6	23	35	-17.49	3.96	2328	3761
i7	17	13	-72.49	11.11	15449	12181
i8	13	7	-52.36	6.19	5934	3925
i9	21	22	-55.49	13.92	15194	20702

References

[1] N. Ansari and E. J. Delp, *Partial shape recognition: A landmark-based approach* **IEEE Trans. on PAMI**, vol 12., pp. 489-497, 1990.

[2] J. Bala, and H. Wechsler, *Shape Analysis Using Morphological Processing and Genetic Algorithms*, **Int. Conf. on Tools for AI**, pp.130-137, 1991.

[3] D. H. Ballard, *Generalizing the Hough transform to detect arbitrary shapes*, **Pattern Recognition**, vol. 13, pp. 111-122, 1981.

[4] 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, no. 11, pp. 1053-1063, 1991.

[5] M. Di Ianni, *Simulated Annealing and Genetic Algorithms for Shape Detection*, **Polish J. Control and Cybernetics**, to appear.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading (MA), 1989.

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Univ. Mich. Press, 1975.

[8] M. W. Koch and R. L. Kashyap, *Using polygons to recognize and locate partially occluded objects*, **IEEE Trans. on PAMI**, vol. 9, pp. 483-494, July 1987.

[9] J.-G Leu and I.-N. Huang, *Partial shape matching based on binary tree shape representation*, **Pattern Recognition**, vol. 21, no. 6, pp. 607-622, 1988.

[10] N. M. Nasrabadi, and W. Li, *Object recognition by a Hopfield Neural Network* **IEEE Trans. on Sys., Man, and Cybernetics**, vol. 21, no.6, pp.1523-1535, 1991.

[11] M. Seibert, and A. M. Waxman, *Learning and Recognizing 3D Objects from Multiple views in a Neural System*, **Neural Networks for Perception**, vol. 1, pp. 426-444, 1992.

[12] W. -H. Tsai and K. S. Fu, *Attributed grammer - a tool for combining syntactic and statistical approaches to pattern recognition*, **IEEE Tran. on SMC**, vol. 10, Dec 1980.

[13] W. -H. Tsai and S.-S. Yu, *Attributed string matching with merging for shape recognition*, **IEEE Tran. on PAMI**, vol. 7, pp 453-462, July 1985.

[14] J. L. Turney, T. N. Mudge and R. A. Voltz, *Recognizing partially occluded parts*, **IEEE Trans. on PAMI**, pp. 410-421, July 1985.

[15] M. C. Yang, K. Mehrotra, C. K. Mohan, S. Ranka, *A New Algorithm for Shape Matching*, **Proc. Artificial Neural Networks in Engg. Conf.**, pp.523-528, Nov. 1993.

[16] S. S. Young, P. D. Scott and N. Nasrabadi, *Object Recognition Using Multi-layer Hopfield Neural Network*, **Conference on Computer Vision and Pattern Recognition**, pp. 417-422, June 21-23, 1994.