PARTIAL SHAPE MATCHING USING GENETIC ALGORITHMS

Ender Ozcan and Chilukuri K. Mohan

eozcan/mohan@top.cis.syr.edu

2-120 Center for Science and Technology,

Department of Electrical Engineering and Computer Science,

Syracuse University, Syracuse, NY 13244-4100, U.S.A.

315-443-2322/(fax) 1122

Abstract

Shape recognition is a challenging task when images contain overlapping, noisy, occluded, partial shapes. This paper addresses the task of matching input shapes with model shapes described in terms of features such as line segments and angles. The quality of matching is gauged using a measure derived from attributed shape grammars. We apply genetic algorithms to the partial shape-matching task. Preliminary results, using model shapes with 6 to 70 features each, are extremely encouraging.

Key words : Partial Shape Matching, Genetic Algorithms, Attributed Strings, Pattern Recognition.

PARTIAL SHAPE MATCHING USING GENETIC ALGORITHMS

1 Introduction

Shape recognition techniques attempt to identify which of a fixed set of model shapes are present in the input shape. For example, most robotics applications for part inspection and VLSI design involve locating and identifying objects, requiring good shape recognition algorithms. Existing shape recognition techniques [Turney *et al.* (1985), Ansari and Delp (1990), Nasrabadi and Li (1991)] do not perform well for shapes of objects that are occluded, or touch or overlap with other objects. Such problems necessitate a flexible shape recognition algorithm that makes use of incomplete matching information.

This paper proposes a new approach for partial shape matching, applying a Genetic Algorithm (GA) to an attributed string representation [Tsai and Yu (1985)] that provides rotation and size independence. DiIanni *et al.* (1996) have also applied GAs and simulated annealing for matching shapes but the results reported were not encouraging, possibly because of using raw pixel arrays rather than shape features. In contrast to our work, the shape analysis method of Bala and Wechsler (1991) does not use GAs directly for shape matching; instead, GAs are used in their work to develop morphological operators that can discriminate among classes containing different shapes.

The new approach is introduced in Section 2. Section 3 contains experimental results, and conclusions are presented in Section 4.

2 Genetic Algorithm for Shape Matching

Genetic Algorithms (GAs), introduced by Holland (1975), conduct search using a fixed size *population* of *individuals* (candidate solutions). New solutions are generated using *operators*, and a *selection mechanism* is used to obtain new *generations* containing better individuals.

Given an input shape with n features and S model shapes with a total number of M features, partial shape matching involves associating input features with model features. The search space is immense, of size ranging up to M^n, since the input image may contain multiple partial instances of the same model shape.

2.1 Representation

Attributed strings [Tsai and Yu (1985)] are used for the representation of polygonal shapes, consisting 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}. We assume the use of existing feature extraction algorithms. We perform normalization using the immediately preceding feature's length: $l'_i = l_i/l_{i-1}$; an example is shown Figure 1. These features are invariant under translation, scale and rotation transformations. The shape recognition problem now reduces to multiple substring matching. We use the following notation:

- Input shape $I = (I_1, I_2, ..., I_n)$ with $|I| = n$ features, where I_p is the pth feature, consisting of attributes $(l(I_p), \theta(I_p))$, where l and θ yield normalized lengths and angles.
- The model shapes are $M_1, M_2, ..., M_S$, where $M_j = (M_{j,1}, M_{j,2}, ..., M_{j,m_j})$, and where $M_{j,r}$ is the rth feature of the jth model shape, consisting of attributes $(l(M_{j,r}), \theta(M_{j,r}))$.
- Each individual $P = (P_1, P_2, ..., P_k, ..., P_n)$ corresponds to a mapping μ_P from input shape features to model shape features such that $P_k = \mu_P(I_k) = M_{j,i}$, where $1 \le k \le n$, $1 \le j \le S$, and $1 \le i \le m_j$.

Example: With $P = [(2,3), (2,4), (2,5), (3,4) \ldots]$, for instance, the first feature I_1 of the input shape is mapped by this individual to $P_1 = \mu_P(I_1) = (2,3)$, the third feature of the second model shape.

2.2 Fitness

$$Fitness(P) = -(||\{I_k|\ \mu_P(I_k) = M_{i,j},\ \mu_P(I_{k-1}) = M_{l,m},\ i \ne l \text{ or } m \ne j-1\}|| + ||\{I_k|\ d(I_k, \mu_P(I_k)) > \epsilon\}||)$$

The first term in the above expression penalizes the number of model shape fragments assigned to feature sequences by P. The second term represents the number of features in the input image that are not matched by P. The matching threshold ϵ is a nonlinear function of $t = \max(l(I_k), l(\mu_P(I_k)))$, allowing less error for high values, e.g., 0.2 for $t > 0.5$, and higher error for lower values, e.g., 0.9 for $t < 0.005$.

$d(I_k, \mu_P(I_k))$ measures the dissimilarity between input shape feature I_k and model feature $\mu_P(I_k)$:

$$d(I_k, \mu_P(I_k)) = \begin{cases} d_\theta(I_k, \mu_P(I_k)) + d_l(I_k, \mu_P(I_k)) & \text{if } \mu_P(I_{k-2}) = M_{i,j-2},\ \mu_P(I_{k-1}) = M_{i,j-1} \text{ and} \\ & \mu_P(I_k) = M_{i,j} \text{ for some } i, j, k \\ d_\theta(I_k, \mu_P(I_k)) & \text{if } \mu_P(I_{k-1}) = M_{i,j-1} \text{ and } \mu_P(I_k) = M_{i,j} \text{ for some } i, j, k \\ \infty & otherwise. \end{cases}$$

where $d_\theta(I_k, \mu_P(I_k)) = c_\theta\ |\theta(I_k) - \theta(\mu_P(I_k))|$, and $d_l(I_k, \mu_P(I_k)) = |(l(I_k) - l(\mu_P(I_k))|/\max(l(I_k), l(\mu_P(I_k)))$. The constant c_θ is chosen in our experiments so that differences of $\pi/18$ are considered negligible.

2.3 Selection Mechanism

A linear ranking strategy is used during reproduction, with the best individual being allocated roughly five times more offspring than the worst individual. An elitist survival selection mechanism is also used: the best two thirds (66%) of all individuals in a generation are allowed to survive into the next generation.

2.4 Operators

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

- Traditional one-point crossover (1PTX) is applied to the individuals, producing two children. This operator randomly chooses two individuals P^i and P^j as parents from the population. P^i and P^j are then cleaved at a randomly chosen crossover point, c, where $1 \leq c < n$, and the sub-sequences from different parents are recombined to generate two offspring.
- Individuals are mutated with 30% mutation rate. Mutation randomly chooses points on an individual and replaces the existing model features (to which input shape features are mapped) by the features of a randomly chosen model shape.
- 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 one direction, then in the opposite direction, replacing the relevant component by the most appropriate feature from the model to which neighboring shape features are mapped. For instance, if $\mu_P(I_k) = M_{j,i}$ and $\mu_P(I_{k+1}) = M_{j',i'}$, hill climbing first attempts to change $\mu_P(I_k)$ to $M_{j',i'-1}$. If this attempt does not improve the fitness, hill climbing attempts to change $\mu_P(I_{k+1})$ to $M_{j,i+1}$.

3 Experimental Results

In previous experiments, reported in Ozcan (1996), best results were obtained using 1PTX, mutation rate of 0.3, and hill climbing. A large library of 40 polygonal model shapes with a total of 1505 features was used for the experiments. Some of these shapes are shown in Figure 3. Input shapes contain overlapped model shapes, possibly scaled differently (Figure 2). Prefixes "i" and "m" refer to input and model shapes, respectively, e.g., "$i2$" is the second input shape. We used a population size *twice* the number of features of each input shape. Each test was repeated 100 times for all input shapes. A Sun workstation was used for all experiments.

Table 1 shows the results obtained by applying the GA to noisy versions of $i4$. Locations of 5%-25% of the input image vertices were randomly perturbed. All runs resulted with a fitness which approximately equals the best possible fitness. In some experiments, the random perturbations assist the GA in moving out of local optima.

Experimental results, summarized in Table 2, are very promising. The GA found the best possible correct match in all runs. For $i0$, in which $m22$ is overlapped twice, the GA gets stuck at a local optimum in 15% of runs, where the input shape is partially matched to $m25$ instead of $m22$. The GA reached correct results in at most 254 generations, on average.

4 Conclusions

A new approach for shape recognition is developed, which utilizes genetic algorithms and attributed string representation. Outline features of shapes are represented using attributed strings. Each feature is a line segment with two attributes: relative length and angle, providing rotation and size invariance. Experimental results demonstrate that this approach is computationally efficient and memory requirements are smaller than neural network models [Yang *et al.* (1993)]. The use of many-individual populations and evolutionary operators overcomes the primary problem faced by greedy algorithms that get stuck in locally optimal solutions. For shapes with curvilinear segments, a different representation would be more appropriate [Bruckstein *et al.* (1992)], but we expect that a GA can still be used for matching.

References

[1] M. DiIanni, R. Diekmann, R. Luling, J. Schulze and S. Tschoeke (1996), *Simulated Annealing and Genetic Algorithms for Shape Detection*, **Control and Cybernetics**, vol. 25, No. 1, pp. 159-175.

[2] E. Ozcan and C. K. Mohan (1996), *Shape Recognition Using Genetic Algorithms*, **Proc. IEEE Int. Conf. on Evol. Computation**, Nagoya (Japan), pp. 414-420.

[3] M. C. Yang, K. Mehrotra, C. K. Mohan, S. Ranka (1993), *A New Algorithm for Shape Matching*, **Proc. Artificial Neural Networks in Eng. Conf.**, St. Louis (MO), pp.523-528.

[4] A. M. Bruckstein, N. Katzir, M. Lindenbaum and M. Porat (1992), *Similarity-Invariant Signatures for Partially Occluded Planar Shapes*, **Int. J. of Computer Vision**, 7:3, pp. 271-285.

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

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

[7] N. Ansari and E. J. Delp (1990), *Partial shape recognition: A landmark-based approach*, **IEEE Trans. on Pattern Anal. Machine Intell.**, vol 12., no.5, pp. 489-497.

[8] W. -H. Tsai and S.-S. Yu (1985), *Attributed string matching with merging for shape recognition*, **IEEE Tran. on Pattern Anal. Machine Intell.**, vol. 7, pp 453-462.

[9] J. L. Turney, T. N. Mudge and R. A. Voltz (1985), *Recognizing partially occluded parts*, **IEEE Trans. on Pattern Anal. Machine Intell.**, vol. PAMI-7, pp. 410-421.

[10] J. H. Holland (1975), *Adaptation in Natural and Artificial Systems*, Univ. Mich. Press, Ann Arbor (MI).

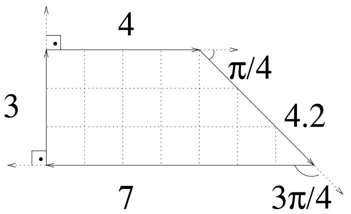

Figure 1: Polygon represented as $((1.33, \frac{\pi}{2}), (1.05, \frac{\pi}{4}), (1.67, \frac{3\pi}{4}), (0.43, \frac{\pi}{2}))$.

Noise Levels	0.05	0.10	0.15	0.20	0.25
Success Rate	0.96	0.90	0.74	0.94	0.55

Table 1: Test results for input image *i*4: Success rates, based on over 100 experiments, for different noise levels which indicates the fraction of features perturbed.

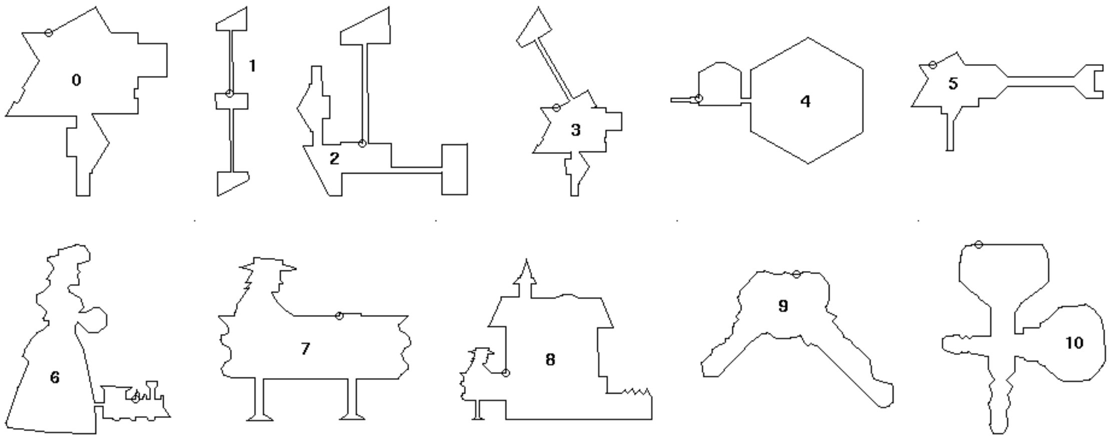

Figure 2: Input shapes used for the experiments.

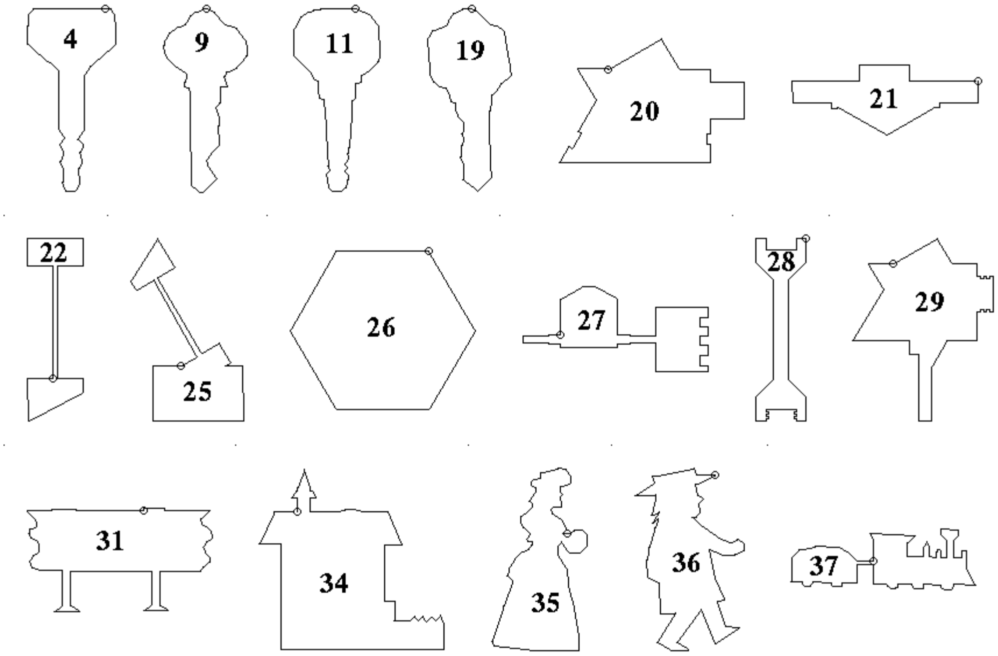

Figure 3: Model shapes used to construct the input shapes in the experiments.

Shape label	No. of features	Overlapping models	Frequency of correct matching	No. of generations μ	No. of generations σ	Time (sec.) μ	Time (sec.) σ
$i0$	32	$m20, m21$	1.00	82	65	3.95	3.04
$i1$	20	$m22, m22$	1.00	31	26	0.73	0.57
$i2$	37	$m21, m22, m22$	0.85	253	242	15.03	13.87
$i3$	41	$m20, m21, m22$	1.00	154	117	11.19	8.22
$i4$	24	$m26, m27$	1.00	138	169	4.27	5.11
$i5$	29	$m28, m29$	1.00	173	160	7.11	6.59
$i6$	113	$m35, m37$	1.00	148	135	57.64	48.57
$i7$	71	$m31, m36$	1.00	141	95	23.75	15.08
$i8$	86	$m31, m34, m36$	1.00	86	74	21.86	18.33
$i9$	67	$m9$,$m19$	1.00	53	42	9.07	7.07
$i10$	91	$m4$,$m11$	1.00	158	102	46.16	29.49

Table 2: Test results for 100 trials of GA (1000 generations), for shapes given in Figure 2: Averages (μ), and standard deviations (σ) are based on experiments in which the correct solution is found.