

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics

Peng-Yeng Yin

*Department of Information Management, National Chi Nan University
Taiwan*

1. Introduction

Representation of digital planar curves is an important step prior to many image analysis tasks, such as object recognition, image matching, target tracking, etc. Polygonal approximation is an important technique to digital curve representation since the main information of curves is preserved at the corner points, it is desired to approximate a digital curve by an appropriate polygon to reduce the memory storage and the processing time for subsequent analyses. The design of a polygonal approximation algorithm not only impacts on the compression ratio of the data volume but also affects the accuracy of the subsequent image analysis tasks. There are several possible criteria with which the polygonal approximation can be performed, one of the most broadly used can be described as "given a digital curve and an error tolerance, the algorithm approximates the curve with a polygon by taking a subset of the points on the curve as the vertices such that the number of vertices is minimized and the approximation error between the curve and the corresponding polygon is no more than the error tolerance." (Yin, 2006)

An exact method to the polygonal approximation problem is impractical due to the intensive computations involved. An attempt using the dynamic programming technique had been made (Dunham, 1986), however, it required a worst-case complexity of $O(N^4)$ where N is the number of data points. Early solutions to reduce the amount of computations rely on local search heuristics, namely the sequential scan-along approaches (Wall & Danielsson, 1984; Ray & Ray, 1993), split-and-merge approaches (Ansari & Delp, 1991; Ray & Ray, 1995), and dominant point detection approaches (Teh & Chin, 1989; Zhu & Chirlian, 1995). However, the quality of the approximation result depends upon the initial condition where the heuristics take place and the metric used to measure the curvature.

Metaheuristics are alternatives to solve complex combinatorial optimization problems. Fred Glover first coined the term *metaheuristic* as a strategy that guides another heuristic to search beyond the local optimality such that the search will not get trapped in local optima. Metaheuristics combine two components, an exploration strategy and an exploitation heuristic, in a framework. The exploration strategy searches for new regions, and once it finds a good region the exploitation heuristic further intensifies the search for this area. In this context, metaheuristics encompass several well-known approaches such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), scatter search (SS), ant colony optimisation (ACO), particle swarm optimisation (PSO), just to name a few. Most of the

central metaheuristics have been applied to the polygonal approximation problems and attained promising results. Instead of describing all the methods, this chapter will focus on the more recently proposed metaheuristics, ACO and PSO, and give their comparative evaluations.

The remainder of this chapter is organized as follows. Section 2 presents the formulation of the polygonal approximation problem. Section 3 renders the details of the ACO- and the PSO-based methods. In Section 4, we present the experimental results and discussions. Finally, a conclusion is given in Section 5.

2. Problem Formulation

Given a digital curve represented by a set of N points, $S = \{x_0, x_1, \dots, x_{N-1}\}$ where $x_{(i+1) \bmod N}$ is considered as the succeeding point of x_i . We define arc $\widehat{x_i x_j}$ as the collection of those points between x_i and x_j , and chord $\overline{x_i x_j}$ as the line segment connecting x_i and x_j . If we approximate $\widehat{x_i x_j}$ by $\overline{x_i x_j}$, the incurred approximation error, denoted by $e(\widehat{x_i x_j}, \overline{x_i x_j})$, can be measured by any distance norm; for here, the L_2 norm, i.e., the sum of squared perpendicular distance from every data point on $\widehat{x_i x_j}$ to $\overline{x_i x_j}$, is adopted. Thus a polygon with the vertex set $T = \{x_{p_0}, x_{p_1}, \dots, x_{p_{M-1}}\}$, where $T \subset S$ and $3 \leq M \leq N$, can approximate the given curve with a total error $E = \sum_{i=0}^{M-1} e(\widehat{x_{p_i} x_{p_{(i+1) \bmod M}}}, \overline{x_{p_i} x_{p_{(i+1) \bmod M}}})$, and our aim is to

construct a polygon with the minimal vertex set and the incurred approximation error is less than the pre-specified tolerance. Formally, the polygonal approximation problem can be formulated as follows.

$$\arg \min_{T \subset S} |T| \quad \text{subject to } 3 \leq |T| \leq N \text{ and } E \leq \varepsilon, \quad (1)$$

where $|T|$ denotes the cardinality of T and ε is the pre-specified error tolerance.

3. Polygonal Approximation Using Metaheuristics

Metaheuristics have shown many successful applications in diverse domains and the effectiveness and the malleability of metaheuristics are proven to be significantly better than most of the traditional local search heuristics. Metaheuristics are attractive to researchers because of their common features: natural metaphor, adaptivity, parallelism, easy implementation, and high quality result. In the following we illustrate the polygonal approximation application using two state-of-the-art metaheuristics: ant colony optimization (ACO) and particle swarm optimization (PSO).

3.1 ACO-based method

The basic framework of ant colony optimization (ACO) was first introduced in Dorigo's Ph.D. dissertation (Dorigo, 1992). Since then many ACO applications have been investigated such as the travelling salesman problem (Dorigo & Gambardella, 1997), quadratic assignment problem (Maniezzo et al., 1994), and combined heat and power economic dispatch problem (Song et al., 1999). The ACO is inspired by the research on the real ant

behavior. Ethologists observed that ants are able to construct the shortest feasible path from their colony to the feeding source by the use of pheromone trails. An ant leaves some quantities of pheromone on the ground and marks the path by a trail of this substance. The next ant then senses the pheromone laid on different paths and chooses one with a probability proportional to the amount of pheromone on it. The ant traverses the chosen path and leaves its own pheromone. This is an autocatalytic (positive feedback) process which favors the path along which more ants previously traversed. To apply the ACO to circumvent the problem, we need to define the path space and the pheromone field that play central roles in the algorithm (Yin, 2003).

3.1.1 Graph representation

Ideally, we can construct a graph $G = \langle S, E^* \rangle$, where S is the set of data points on the given curve and E^* is the ideal edge set that has the desired property that any closed circuit through E^* which originates and ends at the same node represents a feasible solution to the problem, i.e., the polygon consisting of the edges and nodes along the closed circuit should approximate the curve with $E \leq \epsilon$. However, it is impossible to generate E^* in practice. An alternative is to generate a pseudo-ideal edge set \hat{E} , such that, $E^* \subseteq \hat{E}$. For the constructed circuits which violate $E \leq \epsilon$, we can decrease the intensity of pheromone trails on the circuits to make them less attractive. \hat{E} is constructed as follows. First, an empty edge set is created, i.e., $\hat{E} = \emptyset$. For every node $x_i \in S$, we examine each of the remaining nodes, $x_j \in S$, in clockwise order. The directed edge $\overrightarrow{x_i x_j}$ is added to \hat{E} if the approximation error between the arc $\widehat{x_i x_j}$ and the line segment $\overline{x_i x_j}$ is no more than ϵ . The reason for using a directed edge is to avoid the ants walking backward. Now, the problem of polygonal approximation is equivalent to finding the shortest closed circuit on the directed graph $G = \langle S, \hat{E} \rangle$ such that $E \leq \epsilon$.

For the convenience of presentation, we define some notations as follows. Let the closed circuit completed by the k th ant be denoted $tour_k$, the number of nodes visited in $tour_k$ be $|tour_k|$, and the approximation error between the original curve S and the approximating polygon corresponding to $tour_k$ be $E(S, tour_k)$.

3.1.2 Starting node selection

Each ant chooses a starting node in the graph and sequentially constructs a closed path to finish its tour during each iteration. We establish a selection table for the starting node which is a linear array of N entries denoted by T_i , $i = 1, 2, \dots, N$. Initially, we let each $T_i = 1$. The probability with which the i th node is chosen as a starting node, denoted $Select_i$, is estimated as the entry value T_i divided by the sum of all entry values, $Select_i = T_i / \sum_{j=1}^N T_j$.

The ties with respect to $Select_i$ are broken randomly. Apparently, at the beginning of the first cycle, every node has equal probability of being chosen as a starting node since $Select_i = 1/N$. We then update the entry value of the selection table at the end of each cycle.

Let the set of ants which start with the i th node at the current cycle be Ant_Start_i , and the size of Ant_Start_i be $|Ant_Start_i|$. We update entry T_i based on a trade-off between the average quality of current solutions constructed by those ants in Ant_Start_i and the value of $Select_i$ derived from older cycles. Thus, we let

$$T_i \leftarrow \begin{cases} \frac{(1-r)}{|Ant_Start_i|} \sum_{j \in Ant_Start_i} \frac{1}{|tour_j|} + r Select_i, & \text{if the } i\text{th node was chosen as a} \\ T_i, & \text{starting node at current cycle} \end{cases} \quad (2)$$

otherwise,

where $r \in (0, 1)$ is the parameter which controls the relative contribution of each component.

3.1.3 Node transition rule

The node transition rule is a probabilistic one determined by the pheromone intensity τ_{ij} and the visibility value η_{ij} of the corresponding edge. In the proposed method, τ_{ij} is equally initialized to $1/N$ (actually, any small constant positive value will suffice), and is gradually updated at the end of each cycle according to the average quality of the solutions that contain this edge. On the other hand, the value of η_{ij} is determined by a greedy heuristic which encourages the ants to walk to the farthest accessible node in order to construct the longest possible line segment in a hope that an approximating polygon with fewer vertices is obtained eventually. This can be accomplished by setting $\eta_{ij} = |\widehat{x_i x_j}|$, where $|\widehat{x_i x_j}|$ is the number of points on $\widehat{x_i x_j}$. The value of η_{ij} is fixed during all the cycles since it considers local information only.

We now define the transition probability from node i to node j through directed edge $\overrightarrow{x_i x_j}$ as

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{\substack{\overrightarrow{x_i x_h} \\ \text{from } x_i}} (\tau_{ih})^\alpha (\eta_{ih})^\beta}. \quad (3)$$

Also, the ties with respect to p_{ij} are broken randomly.

3.1.4 Pheromone Updating Rule

The intensity of pheromone trails of an edge is updated at the end of each cycle by the average quality of the solutions that traverse along this edge. In particular, the pheromone intensity at directed edge $\overrightarrow{x_i x_j}$ is updated by

$$\tau_{ij} \leftarrow \rho \tau_{ij} + \max\left(\sum_{k=1}^m \Delta \tau_{ij}^k, 0\right), \tag{4}$$

where $\rho \in (0, 1)$ is the persistence rate of previous pheromone trails, and $\Delta \tau_{ij}^k$ is the quantity of new trails left by the k th ant and it is computed by

$$\Delta \tau_{ij}^k = \begin{cases} \frac{1}{|tour_k|}, & \text{if } \overrightarrow{x_i x_j} \in tour_k \\ & \text{and } E(S, tour_k) \leq \varepsilon; \\ -\frac{E(S, tour_k)}{\varepsilon N}, & \text{if } \overrightarrow{x_i x_j} \in tour_k \\ & \text{and } E(S, tour_k) > \varepsilon; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, more quantities of pheromone trails will be laid at the edges along which most ants have constructed shorter feasible tours. On the other hand, in the worst case, the edges will receive no positive rewards because either no ants walked through them or most passing ants constructed infeasible tours. As such, the proposed rule can guide the ants to explore better tours corresponding to high quality solutions.

3.2 PSO-based method

Particle swarm optimization (PSO) is a new metaheuristic developed in 1995 (Kennedy & Eberhart, 1995). It has exhibited effectiveness and malleability in many applications, such as evolving weights and structure for artificial neural networks (Eberhart & Shi, 1998), manufacture end milling (Tandon, 2000), and reactive power and voltage control (Yoshida et al., 1999). The development of PSO is inspired by the observation on the behaviors of bird flocking. A large number of birds flock synchronously, change direction suddenly, and scatter and regroup together. Each individual, called a particle, benefits from the experience of its own and that of the other members of the swarm during the search for food. The PSO models the social dynamics of flocks of birds and serves as an optimizer for nonlinear continuous functions. In order to deal with combinatorial optimization, the discrete version of PSO has also been introduced (Kennedy & Eberhart, 1997). However, in our experiments this discrete version does not show effective result for polygonal approximation problem. We conjecture that the deterioration is due to the linear combination of reference solutions which is often adopted in solving continuous function optimization. Thus, we add genetic features to enhance the search ability in combinatorial optimisation using the discrete PSO (Yin, 2006).

3.2.1 Particle representation and fitness evaluation

Since particles of the PSO correspond to candidate solutions of the underlying problem, we use the particle to represent the approximating polygon by a binary vector. For the i th particle, the corresponding representation is

$$P_i = (p_{i0}, p_{i1}, \dots, p_{i(N-1)}) \quad \text{subject to } \sum_{j=0}^{N-1} p_{ij} \geq 3 \text{ and } p_{ij} \in \{0, 1\}, \quad (6)$$

where $p_{ij} = 1$ if x_j is one of the vertices chosen to represent the polygon, and $p_{ij} = 0$ otherwise. Thus, the particle representation indicates which data points constitute the vertex set T of the polygon and $\sum_{j=0}^{N-1} p_{ij} = |T|$.

The fitness of the particle is evaluated in two ways. If the approximation error entailed by a candidate polygon exceeds the specified error tolerance, i.e., $E > \varepsilon$, the fitness of the corresponding particle will be assigned a negative value to express the infeasibility degree of this candidate solution, else the particle fitness is set to the inverse of the sum of particle bit values to assess the solution quality in terms of the number of vertices. More precisely, the fitness of particle P_i is determined by

$$\text{fitness}(P_i) = \begin{cases} -E/\varepsilon N & \text{if } E > \varepsilon, \\ 1/\sum_{j=0}^{N-1} p_{ij} & \text{otherwise.} \end{cases} \quad (7)$$

Therefore, there are two optimization goals in our setting. The first one is to move the particle from infeasible solution space to feasible regions, and the second one is to fly the particle to a new position which may result in a polygon with fewer vertices, i.e., with better merit in problem objective. The two optimization goals are pursued simultaneously since the PSO evolves with a swarm of particles and each of which may invoke different fitness evaluation depending on the entailed approximation error.

3.2.2 Genetic operations

PSO is a population-based search paradigm using a swarm of particles, it is natural to compare PSO with GA which is another population-based search algorithm and is well-known to the community. In PSO, each particle flies to a better position which is a randomized weighted sum of vectors based on its personal best (*pbest*) and the global best (*gbest*) positions, while in GA the quality of individual chromosome is improved by using two principal genetic operations: *selection* and *reproduction*. The selection operation picks the good individuals for survival to mimic the natural selection of the fittest and the reproduction operation provides a mechanism to exchange and recombine the information (building blocks) among good-quality individuals. The feature of genetic selection has been added to PSO for solving continuous function optimization problems (Angeline, 1998; Shigenori et al., 2003) and the experimental results demonstrated substantial improvement over the original version. In this chapter, we further devise the scheme for conducting the genetic reproduction with the discrete PSO.

Since the particle vector adjustment formulae are in fact a linear combination of critical vectors with quasi-random coefficients, the newly explored parameter values are bounded between experienced vectors to some extent. This is perhaps a desired property for continuous function value optimization problems, however, it hinders the solution exploration for discrete combinatorial optimization. For the latter one, the building blocks of

good quality solutions are segments of specific ordering or partial selections of elements, and the optimal solution may be obtained through recombination of those segments instead of a weighted sum of those values. Hence, we propose a new particle adjustment rule with genetic recombination for the j th bit of particle i as follows.

$$p_{ij} = w(0, w_1)rand(p_{ij}) + w(w_1, w_2)rand(pb_{ij}) + w(w_2, 1)rand(gbest_j), \quad (8)$$

where $0 < w_1 < w_2 < 1$ and $w(\bullet)$ and $rand(\bullet)$ are the threshold function and the probabilistic bit flipping function, respectively, and they are defined as follows.

$$w(a, b) = \begin{cases} 1 & \text{if } a \leq q_1 < b, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where $q_1 \in U(0, 1)$ is a randomly drawn real number. Therefore, only one of the three terms on the right hand side of Eq. (8) will remain depending on the value of q_1 .

$$rand(y) = \begin{cases} (y+1) \text{ modulo } 2 & \text{if } q_2 \leq t, \\ y & \text{otherwise.} \end{cases} \quad (10)$$

Thus, $rand(y)$ mutates the binary bit y with a small probability t (q_2 is another random number drawn from $U(0, 1)$). To relate the new particle adjustment rule to genetic reproduction, we analyze Eq. (8) in two aspects. *First*, the particle P_i derives its every single bit from either one of p_{ij} , pb_{ij} , or $gbest_j$, this operation corresponds to a 3-way uniform crossover among P_i , pb_{ij} , and $gbest_j$, such that the particle can exchange building blocks (segments of ordering or partial selections of elements) with personal and global experiences. *Second*, each bit attained in this way will be flipped with a small probability, analogous to the binary mutation performed in genetic algorithms. As such, the genetic reproduction, in particular, the crossover and mutation, have been added to the discrete PSO, and this new version is very likely more suitable to solve combinatorial optimization problems than the original one.

3.3 Hybrid strategy

Metaheuristics combine two elements, *exploration* and *exploitation*, in a framework. The exploration strategy searches for new regions, and once it finds a good region the exploitation heuristic further intensifies the search for this area. However, since the two strategies are usually inter-wound in the algorithm, the search is conducted to other regions before it finds the local optima. Many researchers have suggested to employ a hybrid strategy which embeds a local optimizer such as hill-climbing in between the iterations of the metaheuristics to enhance the searching ability. In the light of this, we propose to embed a local heuristic into the ACO- and the PSO-based approaches. To save the computational efforts, the local heuristic is only applied to the best candidate solution observed so far at each iteration.

The local heuristic, named the segment-adjusting-and-merging, takes into account the problem-specific knowledge that the approximation error may be further reduced if the positions of the vertices of the polygon are appropriately adjusted, and that the number of vertices is decreased if we merge two adjacent segments under the constraint that the

resulting new polygon still satisfies the error tolerance. The two solution-improving processes are performed repeatedly until the number of vertices cannot be further decreased.

4. Experimental Results and Discussions

In this section, we present the computational results and evaluate the performance of the algorithms. The platform of the experiments is a PC with a 1.8 GHz CPU and 192 MB RAM. The algorithms are coded in C++. A number of benchmark curves borrowed from relevant literature are used for testing.

4.1 Benchmark curves

Three synthesized benchmark curves (see Fig. 1) and two real image curves (see Figs. 2-3) which are broadly used in the literature to evaluate various algorithms for polygonal approximation are included in our experiments for testing. As such the readers can easily compare the proposed algorithms with existing works. Fig. 1(a) is a leaf curve with 120 points, Fig. 1(f) is a chromosome curve with 60 points, Fig. 1(k) is a semi-circle curve with 102 points, Fig. 2(a) is a plane contour image with 682 edge points, and Fig. 3(a) is a fish contour image with 700 edge points.

4.2 Competing metaheuristics

In addition to evaluating the ACO-based and the PSO-based algorithms presented in Section 3, we compare the results with those obtained using two other major metaheuristics: GA (Yin, 1999) and TS (Yin, 2000). The GA-based approach used the same solution representation scheme as that of the PSO-based method (see Eq. (6)). It applied a fitness function as $k - \sum_{j=0}^{N-1} p_{ij} - \max(E-\varepsilon, 0)$ where k is a constant. Besides using the traditional genetic operators (selection, crossover, mutation), a learning strategy is employed to improve the best chromosome observed so far at each iteration. The TS-based approach also followed Eq. (6) to generate its solution configuration. Three kinds of moves are defined: vertex-addition, vertex-deletion, and vertex-adjustment. As such the bounded neighborhood space is well defined. The tabu moves are enforced in order to prevent the current solution configuration getting into a subregion already visited. However, appropriate aspiration criteria are applied to resume a tabu move when it results in a better solution status than the ones observed so far.

4.3 Comparative performances

All of these metaheuristics have been proved to significantly outperform traditional local heuristics in solving the polygonal approximation problem (Yin, 2003; Yin, 2006). We thus focus our comparison among these metaheuristics only. The experiments on the three synthesized curves using the competing metaheuristics are shown in Table 1. As these metaheuristics are stochastic and each separate run of the same program may yield a different result, we report the average number of vertices (M) on the finally obtained polygon and the average consumed times in seconds (t) over 10 independent runs. The standard deviation (σ_M) of M is calculated for measuring the stability of the metaheuristics. It is evident from Tables 1 that the ACO- and the PSO-based approaches have better

performance than those of the GA-based and the TS-based approaches in terms of minimizing the value of M . This is due to the fact that the ACO- and the PSO-based methods further intensify the search in the neighborhood of the best solution observed so far using the hybrid strategy. All of the four competing metaheuristics have small values of σ_M , this means that these methods are all malleable against various curves with different properties. As for the computational times, all of these methods can derive quality results very quickly because the number of data points on the curves is small.

Fig. 1 shows the visualization of the finally obtained approximating polygons with their specified error tolerance (ϵ) and the number of yielded vertices (M) using various metaheuristics. It is seen that GA and TS yield worse approximating polygons with redundant vertices while ACO and PSO produce the least number of vertices but still preserving the main corner information.

	ϵ	GA		TS		ACO		PSO	
		$M(\sigma_M)$	t	$M(\sigma_M)$	t	$M(\sigma_M)$	t	$M(\sigma_M)$	t
Leaf (N=120)	150	15.6 (0.6)	0.4	10.6 (0.5)	0.1	11.0 (0.0)	0.9	10.7 (0.5)	0.4
	100	16.3 (0.5)	0.3	13.7 (0.6)	0.1	12.6 (0.2)	0.8	12.4 (0.5)	0.3
	90	17.3 (0.5)	0.3	14.6 (0.5)	0.1	12.8 (0.3)	0.9	13.0 (0.0)	0.3
	30	20.5 (0.6)	0.3	20.1 (0.5)	0.1	16.6 (0.4)	0.9	16.6 (0.5)	0.3
	15	23.8 (0.6)	0.3	23.1 (0.5)	0.1	19.7 (0.3)	0.9	20.0 (0.0)	0.2
Chromosom (N=60)	30	7.3 (0.4)	0.2	6.7 (0.4)	0.1	6.0 (0.0)	0.4	6.0 (0.0)	0.2
	20	9.0 (0.6)	0.2	8.0 (0.3)	0.1	7.6 (0.3)	0.5	7.6 (0.7)	0.2
	10	10.2 (0.4)	0.2	11.0 (0.4)	0.1	10.0 (0.3)	0.5	10.5 (0.5)	0.1
	8	12.2 (0.5)	0.2	12.2 (0.5)	0.1	11.0 (0.4)	0.5	11.0 (0.0)	0.1
	6	15.2 (0.6)	0.2	14.4 (0.5)	0.1	12.2 (0.3)	0.5	12.4 (0.7)	0.1
Semicircle (N=102)	60	13.2 (0.4)	0.3	11.0 (0.4)	0.1	10.0 (0.0)	0.8	10.0 (0.0)	0.3
	30	13.9 (0.7)	0.3	13.6 (0.5)	0.1	12.0 (0.0)	0.8	12.1 (0.3)	0.3
	25	16.8 (0.7)	0.3	14.9 (0.6)	0.1	13.0 (0.0)	0.7	13.2 (0.4)	0.3
	20	19.2 (0.6)	0.3	16.2 (0.6)	0.1	15.8 (0.4)	0.7	14.6 (0.7)	0.2
	15	23.0 (0.9)	0.3	18.3 (0.7)	0.1	16.8 (0.4)	0.7	15.8 (1.2)	0.2

Table 1. The comparative results on synthesized curves using competing metaheuristics

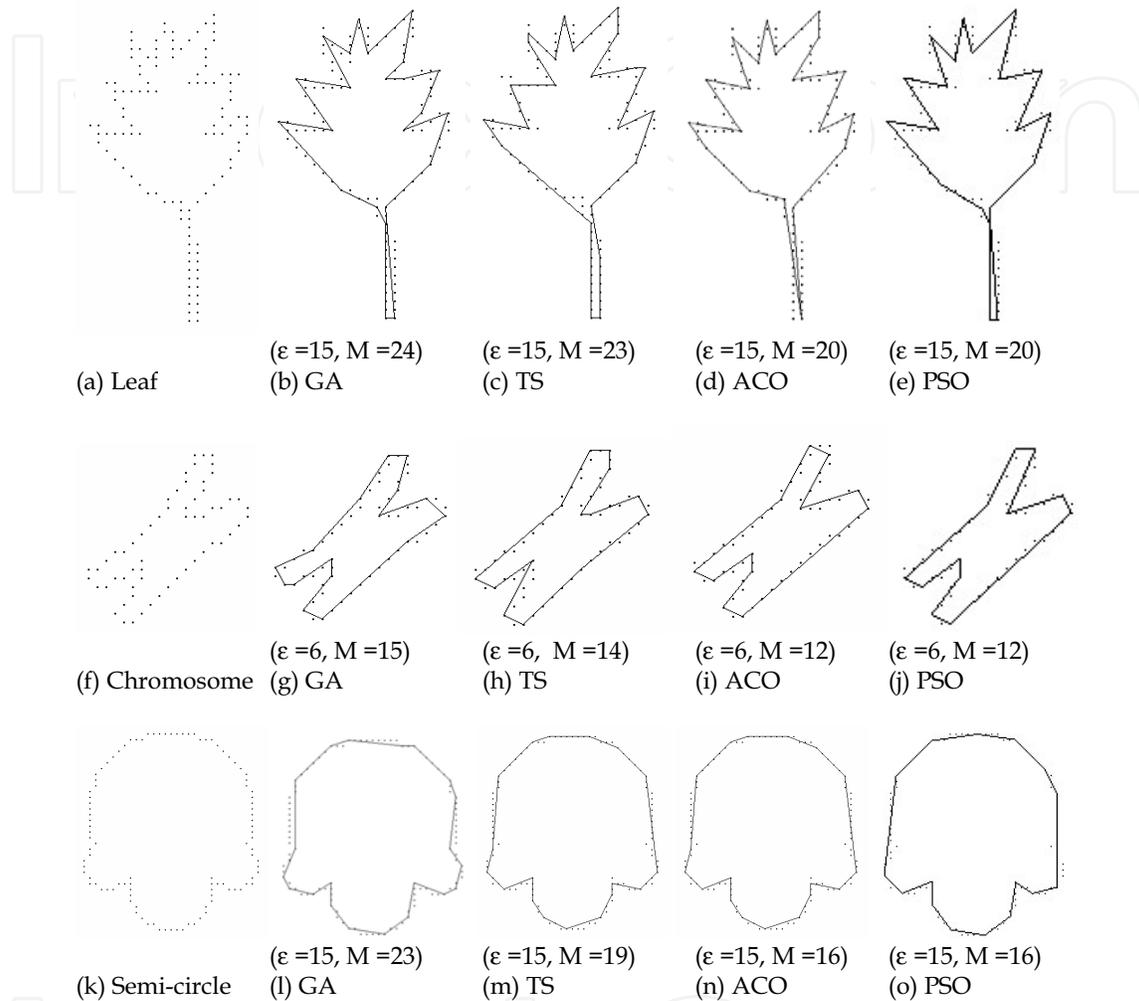


Fig. 1. Finally obtained approximating polygons on the synthesized curves with their specified error tolerance (\mathcal{E}) and the number of yielded vertices (M) using various metaheuristics

To demonstrate the feasibility of the metaheuristics for real-world applications, two real images containing a symbol of a plane and a fish, respectively, are further experimented with. The two images are binarized and the contour edge points are extracted by detecting the black-white transitions (see Figs. 2(a) and 3(a)). By specifying various values of error tolerance, the comparative performances obtained using the competing metaheuristics are summarized in Table 2. It is observed that the performance of the GA- and the TS-based methods deteriorates in the two real applications as the error tolerance decreases where the numbers of polygon vertices are significantly greater than those obtained by the ACO- and

the PSO-based approaches. However, the TS-based approach is the fastest one because it only uses one seed solution to conduct the search path while the others are population-based searching methods.

Figs. 2-3 show the finally obtained approximating polygons with their specified error tolerance (ϵ) and the number of yielded vertices (M) using various metaheuristics. Similarly, the ACO- and the PSO-based methods economically preserve the main corner information on the curve while the GA- and the TS-based methods may use multiple vertices to approximate some corners in a small region.

To justify the reason behind the performance difference observed, we disable the application of the hybrid strategy in the ACO- and the PSO-based approaches and reperform the experiments again. We found that the new results obtained by the ACO- and the PSO-based methods without hybrid strategy become comparable with that obtained by the GA- and the TS-based methods. Therefore, the problem-specific local heuristics such as the segment-adjusting-and-merging are the key-reason that results in the performance differences among these metaheuristics. It is worth further studying other appropriate problem-specific local heuristics, e.g., the scan-along search, split-and-merge process, and dominant-point detection, to be hybridized with these metaheuristics. Note that the learning strategy employed by the GA-based approach is a general strategy that may be not as efficient as the problem-specific heuristics in some complex problems but it is useful when the problem-specific heuristics are not easy to design.

	ϵ	GA		TS		ACO		PSO	
		$M(\sigma_M)$	t	$M(\sigma_M)$	t	$M(\sigma_M)$	t	$M(\sigma_M)$	t
Plane (N=682)	3000	14.2 (0.8)	2.5	13.0 (0.3)	0.4	12.1 (0.4)	5.0	12.3 (0.5)	6.4
	2000	15.1 (0.9)	2.4	14.4 (0.6)	0.4	13.0 (0.2)	4.7	13.0 (0.0)	6.1
	1000	17.4 (0.6)	2.3	16.7 (0.5)	0.4	14.0 (0.6)	4.8	15.3 (0.8)	5.6
	500	21.3 (0.8)	2.2	19.6 (0.6)	0.4	17.8 (0.5)	4.5	17.4 (0.5)	5.3
	100	33.8 (0.9)	2.4	31.3 (0.5)	0.4	28.1 (0.7)	4.6	24.0 (0.6)	4.5
Fish (N=700)	4000	16.5 (0.5)	2.3	14.0 (0.3)	0.5	12.2 (0.4)	5.7	15.8 (0.4)	5.5
	3000	17.4 (0.6)	2.2	16.0 (0.3)	0.5	14.6 (0.3)	5.9	16.9 (0.3)	5.1
	2000	22.1 (1.0)	2.2	21.2 (0.4)	0.4	17.1 (0.5)	5.5	18.6 (0.9)	4.9
	1000	32.4 (0.9)	2.3	29.1 (1.0)	0.5	26.8 (0.7)	5.6	25.3 (0.5)	4.0
	500	37.0 (1.1)	2.4	35.9 (1.2)	0.4	34.8 (0.7)	5.6	32.8 (0.6)	3.4

Table 2. The comparative results on real image curves using competing metaheuristics

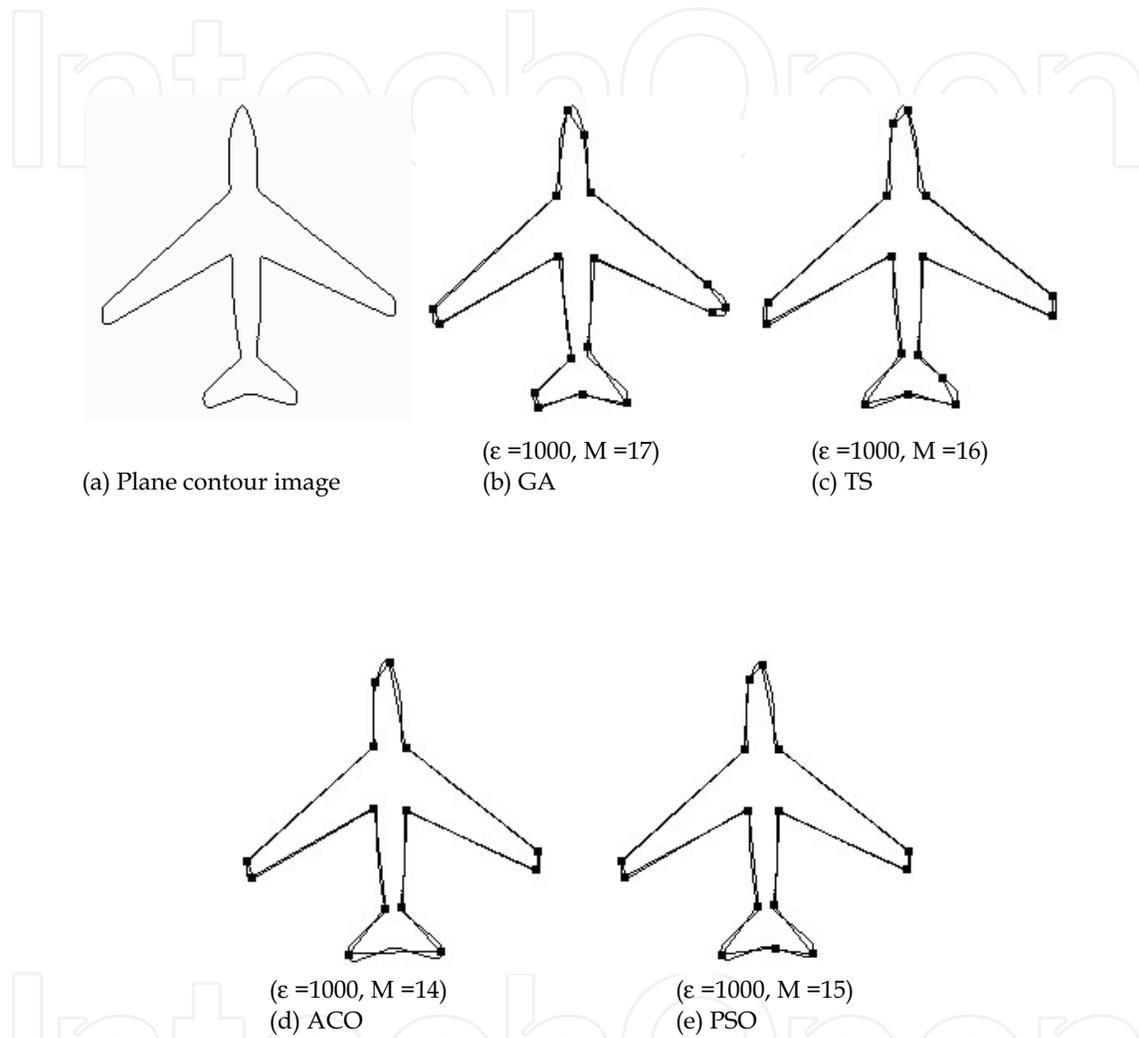


Fig. 2. Finally obtained approximating polygons one the plane image with their specified error tolerance (ϵ) and the number of yielded vertices (M) using various metaheuristics

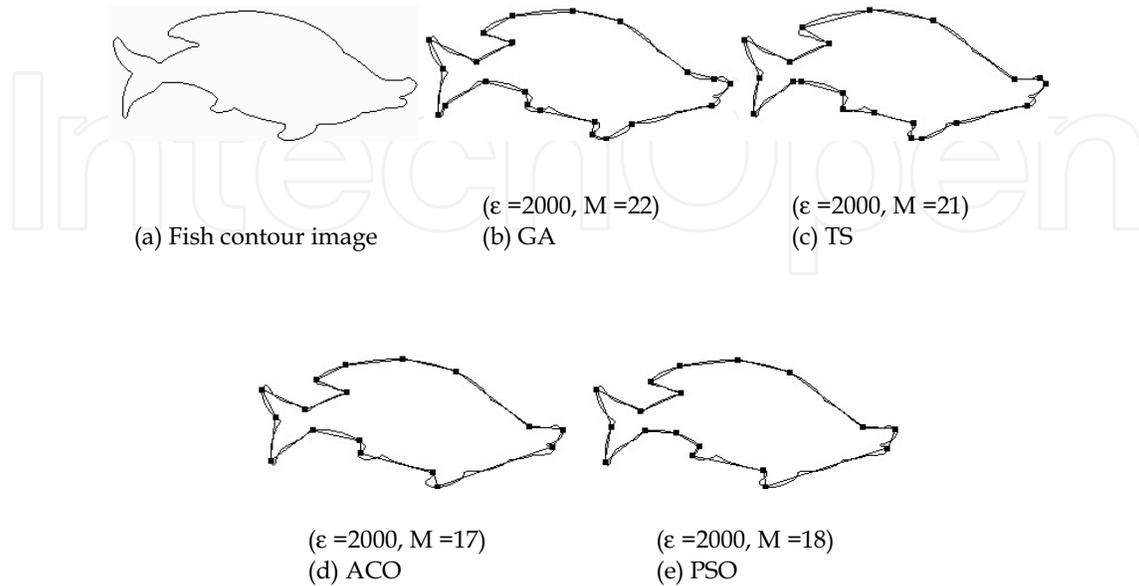


Fig. 3. Finally obtained approximating polygons on the fish image with their specified error tolerance (\mathcal{E}) and the number of yielded vertices (M) using various metaheuristics

5. Conclusion

In this chapter, we investigate the polygonal approximation problem which is fundamental to many image analysis tasks. Traditional problem-specific heuristics are not suitable to be applied alone because the quality of the obtained result depends on the initial setting of the algorithms and the properties of the curves. On the other hand, metaheuristic approaches can produce stable approximation quality for various kinds of curves. We have illustrated the implementations based on two newly developed metaheuristics, namely the ACO and the PSO. To circumvent the underlying problem, specific features have been introduced such as the ACO graph representation, PSO genetic operators, penalty functions, and the hybrid strategy. Experimental results on several benchmark curves have manifested that these new features can improve the performance of metaheuristics in solving the polygonal approximation problem.

6. References

- Angeline, P. (1998). Using selection to improve particle swarm optimization. *Proceedings IEEE Int'l. Conf. on Evolutionary Computation*, pp. 84-89
- Ansari, N. & Delp, E. J. (1991). On detection dominant points. *Pattern Recognition*, Vol. 24, pp. 441-450
- Dorigo, M. (1992). *Optimization, learning, and natural algorithms*. Ph.D. Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy

- Dorigo, M. & Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transaction on Evolutionary Computation*, Vol. 1, pp. 53-66
- Dunham, J. G. (1986). Optimum uniform piecewise linear approximation of planar curves, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 67-75
- Eberhart, R. C. & Shi, Y. (1998). Evolving artificial neural networks, *Proceedings Int'l. Conf. on Neural Networks and Brain*, PL5-PL13
- Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization, *Proceedings IEEE Int'l. Conf. on Neural Networks, IV*, , pp. 1942-1948
- Maniezzo, V.; Colormi, A. & Dorigo, M. (1994). The ant system applied to the quadratic assignment problem. Universite Libre de Bruxelles, Belgium, Technical Report IRIDIA/94-28
- Ray, B. K. & Ray, K. S. (1993). Determination of optimal polygon from digital curve using L1 norm. *Pattern Recognition*, Vol. 26, pp. 505-509
- Ray, B. K. & Ray, K. S. (1995). A new split-and-merge technique for polygonal approximation of chain coded curves. *Pattern Recognition Letters*, Vol. 16, pp. 161-169
- Shigenori, N.; Takamu, G.; Toshiku, Y. & Yoshikazu, F. (2003). A hybrid particle swarm optimization for distribution state estimation. *IEEE Transaction on Power Systems*, Vol. 18, pp. 60-68
- Song, Y. H.; Chou, C. S. & Stonham, T. J. (1999). Combined heat and power economic dispatch by improved ant colony search algorithm. *Electric Power Systems Research*, Vol. 52, pp. 115-121
- Tandon, V. (2000). Closing the gap between CAD/CAM and optimized CNC end milling, Master thesis, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis
- Teh, C. H. & Chin, R. T. (1989). On the detection of dominant points on digital curves, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 859-872
- Wall, K. & Danielsson, P. E. (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, Vol. 28, pp. 220-227
- Yin, P.Y. (1999). Genetic algorithms for polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 13, pp. 1-22
- Yin, P.Y. (2000). A tabu search approach to the polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, pp. 243-255
- Yin, P.Y. (2003). Ant colony search algorithm for optimal polygonal approximation of plane curves. *Pattern Recognition*, Vol. 36, pp. 1783-1797
- Yin, P.Y. (2006). Genetic particle swarm optimization for polygonal approximation of digital curves. *Pattern Recognition and Image Analysis*, Vol. 16, No. 2, pp. 223-233
- Yoshida, H.; Kawata, K.; Fukuyama, Y. & Nakanishi, Y. (1999). A particle swarm optimization for reactive power and voltage control considering voltage stability, *Proceedings Int'l. Conf. on Intelligent System Application to Power Systems*, , pp. 117-121
- Zhu, P. & Chirlian, P. M. (1995). On critical point detection of digital shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 737-748



Vision Systems: Segmentation and Pattern Recognition

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-05-9

Hard cover, 536 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

Research in computer vision has exponentially increased in the last two decades due to the availability of cheap cameras and fast processors. This increase has also been accompanied by a blurring of the boundaries between the different applications of vision, making it truly interdisciplinary. In this book we have attempted to put together state-of-the-art research and developments in segmentation and pattern recognition. The first nine chapters on segmentation deal with advanced algorithms and models, and various applications of segmentation in robot path planning, human face tracking, etc. The later chapters are devoted to pattern recognition and covers diverse topics ranging from biological image analysis, remote sensing, text recognition, advanced filter design for data analysis, etc.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Peng-Yeng Yin (2007). Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics, Vision Systems: Segmentation and Pattern Recognition, Goro Obinata and Ashish Dutta (Ed.), ISBN: 978-3-902613-05-9, InTech, Available from:

http://www.intechopen.com/books/vision_systems_segmentation_and_pattern_recognition/polygonal_approximation_of_digital_curves_using_the_state-of-the-art_metaheuristics

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen