Any Dimension Polygonal Approximation Based on Equidistance Principle

Costas Panagiotakis and George Tziritas

Computer Science Department, University of Crete, Greece, P.O. Box 2208

Abstract

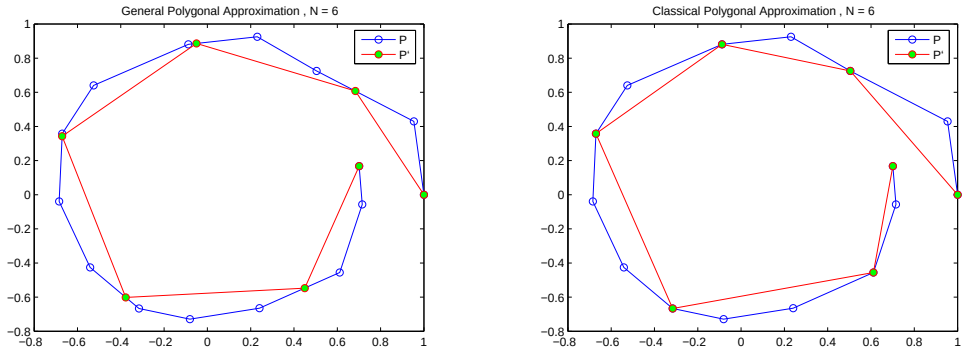
In this paper, we present a new and more general version of polygonal approximation problem (GPA). Given an $N-$vertex polygonal curve P in the n-dimensional space $\Re^n$, we approximate P by finding another $M-$vertex polygonal curve $\dot{P}$, such that the vertices of $\dot{P}$ are an ordered subsequence of the curve points along P. The definition of the classical polygonal approximation problem (PA) demands the $\dot{P}$ vertices to be a subset of P vertices. Therefore, the solutions of GPA problem approximates better the polygonal curve P than the solutions of PA problem. The optimal or a suboptimal solution of GPA is achieved when the approximation errors per line segment are equal. Our method is very flexible on changes of error criteria and on curve dimension yielding an alternative and in many cases better solution than the optimal PA methods with about the same computation cost.

1 Introduction

The polygonal approximation is an important topic in the area of pattern recognition, computer graphics and computer vision. A huge number of applications like object recognition, computational cartography, signal summarization and compression are based on polygonal approximation. The polygonal approximation process saves memory space, reduces the rendering time on graphics applications and gives a more compact representation of P.

Given an $N-$vertex polygonal curve P in the n-dimensional space, the curve approximation of P is to compute another $M-$vertex polygonal curve in the n-dimensional space that approximates the original curve, according to a predefined error criterion. Let $P = \{p_1, p_2, \cdots, p_N\}$ and $\dot{P} = \{\dot{p}_1, \dot{p}_2, \cdots, \dot{p}_M\}$ be the set of the vertex points of the given polygonal curve and its approximation, respectively. According to the general polygonal approximation problem (GPA), the vertices of $\dot{P}$ are an ordered subsequence of the curve

Email address: {cpanag, tziritas}@csd.uoc.gr (Costas Panagiotakis and George Tziritas).



(a) (b)

Fig. 1. Polygonal approximations (red polygon) with six segments of the same given curve (blue polygon). **(a)** A general polygonal approximation and **(b)** a classical polygonal approximation of the given curve.

points along P (Fig. 1(a)), for which it is not required to be a subset of P vertices as the the classical polygonal approximation (PA) demands (Fig. 1(b)). In addition, it holds that $\dot{p}_1 = p_1$ and $\dot{p}_M = p_N$.

Different error criteria have been proposed for polygonal approximation problems. One of the most used is the tolerance zone criterion [1], [2]. Let $\overline{\dot{p}_k\dot{p}_{k+1}}$, $k \in \{1, 2, \cdots, M-1\}$ be a segment of $\dot{P}$ and $S = [\dot{p}_k, p_m, p_{m+1}, \cdots, p_{m+s}, \dot{p}_{k+1}]$ be the corresponding subcurve of P. Under this criterion, the error between the segment $\overline{\dot{p}_k\dot{p}_{k+1}}$ and S is defined as the maximum distance in an L_h ($h \in \{1, 2, \infty\}$) metric between $\overline{\dot{p}_k\dot{p}_{k+1}}$ and each point on the subcurve S. Another frequently used error criterion is the local integral square error (LISE) [3], [4]. Under this criterion, the error between the segment $\overline{\dot{p}_k\dot{p}_{k+1}}$ and S is defined as the sum of squared Euclidean distances from each vertex point of subcurve S. Finally, according to these error criteria the approximation error between $\dot{P}$ and P is defined as the maximum error between the segments of $\dot{P}$ and their corresponding subcurves of P.

The polygonal approximation problem can be formulated in two ways.

- The problem of minimum error ($min - \varepsilon$), where the approximation error is minimized given the number of segments M.
- The problem of minimum number of segments ($min - \#$), where the approximation error is known (ε) and the goal is to find the minimum number of segments (M) that gives error lower than the given error.

1.1 Related Work

The problem of approximating a polygonal curve P (PA) has been studied extensively during the last two decades [5], [6], [7]. The methods, that have been developed, solve the problem by approximating the original polygonal curve P by another polygonal curve $\dot{P}$ under the constraint that the $\dot{P}$ vertex sequence is an ordered subsequence of the vertices along P. There are two well-known approaches for solving this problem: graph-theoretical and dynamic programming. Graph-theoretical methods generate directed acyclic graph on the vertices of P, and then compute the shortest path in the graph [1], [8], [9]. Dynamic programming generates the solution for the problem using results of the smaller problem instances [10], [11]. Concerning the 2-D $min - \#\ problem$ and the $min - \epsilon\ problem$ under the tolerance zone criterion, the lowest computation cost method [8] has cost $O(N^2)$ and $O(N^2 logN)$, respectively. The memory requirements can be reduced to $O(N)$ [2]. The 3-D, 4-D polygonal approximation problem require near-quadratic time and sub-cubic time, respectively [12]. When the L_1 or L_∞ metrics are used, the time requirements for $min - \#$ $problem$ and the $min - \epsilon\ problem$ are reduced to $O(N^2)$ and $O(N^2 logN)$ in any dimensional space [12], respectively. A 2-D monotone polygonal curve can be approximated by an $O(N^{\frac{4}{3}+\delta})$ time and space algorithm [13], where $\delta > 0$ is an arbitrarily small constant. Under the LISE criterion, the 3-D $min - \#\ problem$ and the $min - \epsilon\ problem$ can be solved [4] in $O(N^2)$ and $O(N^2 logN)$, respectively.

These approaches require high execution time and memory when the given polygon size is getting high. Thus in a lot of applications, the main objective is not to compute the optimal solution, but to find low computation cost algorithms that will give a sub-optimal solution [14], [15], [16]. One of the most widely used high-quality curve simplification algorithms is the heuristic method commonly called the Douglas-Peucker approximate algorithm [17], [18]. These approaches have time complexities ranging from $O(N)$ and $O(N^2)$. However, their quality remains less than 80% in comparison to that of the optimal solution [19]. The performance of polygonal approximation algorithms can be measured under variations in scale parameters and data [19].

The rest of the paper is organized as follows: Section 2 presents the proposed general polygonal approximation algorithm. The experimental results and comparisons with the existed classical PA methods are given in Section 3. Finally, conclusions and discussion are provided in Section 4.

Symbols	Definitions
$P = \{p_1, p_2, \cdots, p_N\}$	The given $N-$vertex polygonal curve, $A = p_1, B = p_N$
$\dot{P} = \{\dot{p}_1, \dot{p}_2, \cdots, \dot{p}_M\}$	The $M-$vertex approximating polygonal curve, $\dot{p}_1 = p_1, \dot{p}_M = p_N$
$dist(p, \overline{uv})$	The distance between point p and line $\overline{uv}$, $u, v \in P$
$D(u, v)$	The approx. error between $\overline{uv}$ and the corresponding subcurve of P
$C(t), t \in [0, 1]$	The polygonal curve P, $C(t_k) = p_k$, $t_k = \frac{k-1}{N-1}$, $k \in \{1, 2, \cdots, N\}$
$d(x, y)$	The approx. error between $\overline{C(x)C(y)}$ and the corr. subcurve of P

Table 1
Symbol table of GPA problem.

2 General Polygonal Approximation Algorithm

2.1 Problem Definition

The general polygonal approximation problem is defined as follows: Given an $N-$vertex polygonal curve P in the n-dimensional space $\Re^n$, we approximate P by finding another $M-$vertex polygonal curve $\dot{P}$ in the n-dimensional space $\Re^n$ according to a predefined error criterion, such that the $\dot{P}$ vertices are an ordered subsequence of the curve points along P. The goal is to solve the $min - \epsilon$ *problem* and the $min - \#$ *problem* under any predefined error criterion. Some useful symbols are defined on Table 1. Let u, v be points of polygonal curve P. Let $dist(p, \overline{uv})$ be the distance between point p and line $\overline{uv}$. Let $D(u, v)$ be the approximation error between the segment $\overline{uv}$ and the corresponding subcurve of P under a predefined criterion. We consider that the approximation error ($Error(P, \dot{P})$) between $\dot{P}$ and P is defined as the maximun error between the segments of $\dot{P}$ and their corresponding subcurves of P (Equation (1)).

$$Error(P, \dot{P}) = max_{k \in \{1,2,\cdots,M-1\}} D(\dot{p}_k, \dot{p}_{k+1}) \tag{1}$$

The PA problem can be solved by polynomial algorithms. However, the solutions of GPA problem approximate better the polygonal curve P than the solutions of PA problem. We are going to present an algorithm that in a lot of cases gives better results than the optimal solutions of PA problem with about the same computation cost.

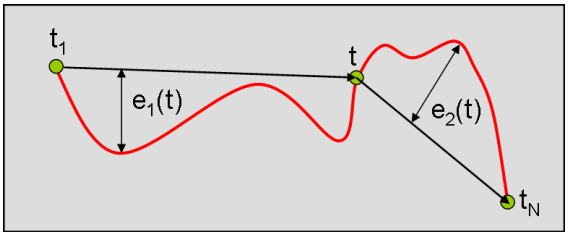


Fig. 2. The maximun of errors $e_1(t)$, $e_2(t)$ give the error of this approximation.

2.2 Reduction to Equipartition Problem

The optimal or a suboptimal solution of GPA is achieved when the approximation errors per line segment are equal, as the error is shared between all the segments (Equation (3)).

$$\epsilon = D(\dot{p}_1, \dot{p}_2) = D(\dot{p}_2, \dot{p}_3) = \cdots = D(\dot{p}_{M-1}, \dot{p}_M) \tag{2}$$

Let $\{t_1 = 0 > t_2 > \cdots > t_{N-1} > t_N = 1\} : C(t_k) = p_k, k \in \{1, 2, \cdots, N\}$ where $C(t), t \in [0, 1]$ denotes the polygonal curve P. Let $\{\dot{t}_1 = 0 > \dot{t}_2 > \cdots > \dot{t}_{M-1} > \dot{t}_M = 1\} : \dot{C}(\dot{t}_k) = \dot{p}_k, k \in \{1, 2, \cdots, M\}$ where $\dot{C}(t), t \in [0, 1]$ denotes the polygonal curve $\dot{P}$. Then the distance function $d(x, y)$ can be defined by the following equation.

$$d(x, y) = D(C(x), C(y))x, y \in [0, 1] \tag{3}$$

The equation (3) can be rewritten as:

$$\epsilon = d(\dot{t}_1, \dot{t}_2) = d(\dot{t}_2, \dot{t}_3) = \cdots = d(\dot{t}_{M-1}, \dot{t}_M) \tag{4}$$

The approximation error ϵ will be minimum or close to minimum. This is the equal errors (EE) criterion. We are going to study the case where $M = 3$ under the tolerance zone criterion. Let $C(t), t \in [t_1, t_N]$ denote a point of polygon P (see Fig. 2). Let $e_1(t)$ and $e_2(t)$ be the distances $d(t_1, t)$, $d(t, t_N)$ under the tolerance zone criterion with L_2 metric, respectively. The maximum of these errors $e(t) = max(e_1(t), e_2(t))$ is the error of the $\{C(t_1), C(t), C(t_N)\}$ approximation. It holds that $e_1(t_1) = e_2(t_N) = 0$, $e_1(t_N) = e_2(t_1)$. If the functions $e_1(t)$, $e_2(t)$ are monotone, which is true for high values of M (small segments Fig. 3(a)) or smooth given polygonal curve (Fig. 3(b)), then the maximum of error $e(t)$ appears when $e_1(t) = e_2(t)$. Otherwise, when $e_1(t) = e_2(t)$ the error will be a local minimum and it will be possibly very close to the global minimum error $min_t(e(t))$ (see Fig. 3(d), 3(e)).

The solution under the EE criterion can be computed approximately using the equipartition method (EP) [20]. The error of this method is extremely small, it decreases with $O(N^{-2})$

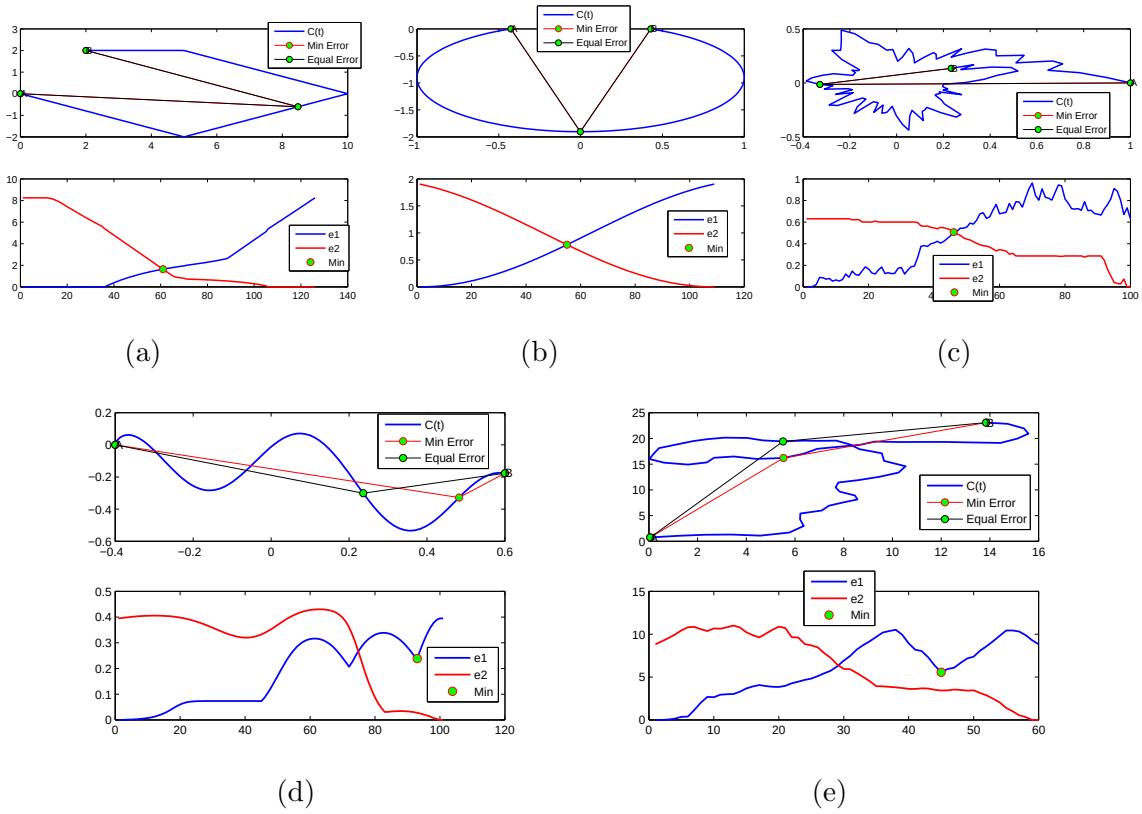


Fig. 3. The polygonal approximations of some given polygonal curves ($C(t)$) with two segments using minimun error criterion (red polygon), and the EE criterion (black polygon). **(a)**, **(b)**, **(c)** The solutions of the EE criterion and minimun error criterion are the same. **(d)**, **(e)** The errors between the corresponding solutions using these criteria are almost the same.

under the assumption that the length of polygonal curve P line segments are almost equal. Moreover, N should be greater than M, while our experiments show that it should be $> 2{\cdot}M$. The EP method accurancy decreases when M is getting high. This constraint does not affect our method results, since it is true for the most polygonal curve approximations. The error between the classical PA method and the optimal solution of GPA problem decreases with $O(N^{-1})$ [1]. The EP algorithm computes for a specific M, the M vertices of $\dot{P}$ under the EE criterion (Equation (3)) and a predefined error metric. The input of the EP algorithm is the M and the matrix $d(t_k, t_l), k, l \in \{1, 2, \cdots, N\}$. The $min - \epsilon$ *problem* is solved directly. We are going to present the constraints that the $d(x, y)$ should satisfy. Under these constraints, it has been proved that the EP problem has always at least one solution ([20]).

We can use as $d(x, y), x, y \in [0, 1]$ any smooth semimetric distance function. A semimetric distance function does not satisfy the triangular inequality property. As a smooth semimetric distance, $d(x, y)$ is characterized by the following properties:

(1) $d(x, y) = 0 \Leftrightarrow x = y$ (isolation).

[1] This error decreases with the length segments of P, which is proportional to N^{-1}.

(2) $d(x,y) = d(y,x)$ (symmetry).

(3) $d(x,y)$ inherits continuity from the polygonal curve P.

(4) $d(x,y)$ can be defined in any dimension ($P \in \Re^n$).

The most of the polygonal approximation error criteria (e.g. the LISE or the tolerance zone criterion) satisfy the above constraints apart from the first one. However, we can define a new distance function $f(x,y)$ satisfying all of the constraints without accurancy loss. Let w be a small constant value, e.g. $w = 10^{-4}$, then $f(x,y)$ can be defined by the following equation (5). In the next sections, we are going to keep using the symbol $d(x,y)$ instead of $f(x,y)$.

$$f(x,y) = w \cdot |x-y| \cdot e^{1-|x-y|} + d(x,y), \quad 0 < w << 1 \tag{5}$$

It holds that $|f(x,y) - d(x,y)| \leq w$. An alternative choice of $f(x,y)$ is given by the equation (7).

$$f(x,y) = w \cdot |x-y| + d(x,y), \quad 0 < w << 1 \tag{6}$$

2.3 The Proposed Algorithm

In this section, the proposed algorithm is presented. The straightforward implementation of the EP method solves the $min - \epsilon$ problem. The input of the method is the number of $\dot{P}$ vertices (M). In addition, it needs the values of matrix $d(t_k, t_l), k, l \in \{1, 2, \cdots, N\}$. This algorithm is described in [20]. There are two versions of the algorithm. The greedy version computes the total solutions in $O(M \cdot N^2)$ steps. However, at least one solution can be computed in $O(M \cdot N)$ steps. In these costs, we have not included the cost of the $d(t_k, t_l)$ computation. If there are more than one solutions, the solution with the minimum error is selected. A brief description of the EP algorithm is given next.

The method is inductive. Thus, when it is executed for M segments, it uses the precomputed results for $M-1$ segments. The major hypothesis of the method is that the function $d(x,y), x, y \in [0,1]$ can be approximated by a polygonal surface $\hat{d}(x,y)$. Thus, the $\hat{d}(x,y)$ is determined by $d(t_k, t_l), k, l \in \{1, 2, \cdots, N\}$. Let

$$D_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \subset [0,1]^2$$

with $x_i = y_i = t_i$, $i, j \in \{1, 2, \cdots, N\}$. The segment D_{ij} can be separated into two triangles: D_{ij}^1 where $x - x_i \geq y - y_j$ and D_{ij}^2 where $x - x_i < y - y_j$. Under our major hypothesis, we have considered that $\hat{d}(x, y), x, y \in D_{ij}^1$ or $x, y \in D_{ij}^2$ is a part of plane.

In each iteration step l, the algorithm computes the curves L_l so that if the point $(u, v) \in L_{l-1}, u > v$, then, it holds that $(z, u), z > u \in L_l \Leftrightarrow d(u, v) = d(z, u)$. These curves consist of line segments defined on D_{ij}^1, D_{ij}^2, so they can be computed by the line segments end points. For $l = 1$, it holds that,

$$L_1 = [(0, 0), (t_1, 0)] \cup [(t_1, 0), (t_2, 0)] \cup \cdots \cup [(t_{N-1}, 0), (1, 0)].$$

Let $(x, y) \in L_l$, $x > y$. Under the above definition, the equipartition of curve $C(t), t \in [0, x]$ into l segments can be done using the precomputed curves $L_l, L_{l-1}, \cdots, L_1$ (see Fig. 4). The equipartition of curve $C(t)$, $t \in [0, 1]$ into $l + 1$ segments can be done using the curves $L_l, L_{l-1}, \cdots, L_1$. Let $q_l(u, v) = d(u, v) - d(u, 1)$, $(u, v) \in L_l$, $u > v$. This function is partially linear. The roots of this function will give the last two points $(\dot{t}_{l-1}, \dot{t}_l)$ of the equipartition. The other points are estimated using the rule of Fig. 4.

It can be proven that for each step l there is a continuous curve $h_l \subset L_l$ starting from $[0, 0]$ and ending on axis $x = 1$ or $y = 1$ (see Fig. 4). We can compute at least one solution of the problem using these curves. The computation cost of h_l curves is $O(M \cdot N)$, because we can track them starting from their known end point $[0, 0]$.

Figure 5 illustrates the results of this proposed algorithm for different polygonal curves and values of M. The L_l curves converge to the diagonal $(y = x)$, as l increases. At least solution belongs on the h_l curve. However, in some cases, some solutions appear on other curves. Figure 5(c) illustrates such an example, where two solutions appear on other curves.

2.4 The $min - \#$ problem

The $min - \#$ problem can be solved by the EP method under the same time-space requirements as the $min - \epsilon$ problem. The algorithm is presented next: We execute the EP method computing the function $q_l(u, v)$. Let $(u_k, v_k) \in L_l, u_k > u_k$ with $q_l(u_k, v_k) = 0$. The minimum approximation error of l segments is $min_k(d(u_k, v_k))$, as we have used the EE criterion. Let k_0 denote the $argmin_k(d(u_k, v_k))$. The method terminates when $d(u_{k_0}, v_{k_0}) < \epsilon$, where ϵ denotes the given approximation error. The (u_{k_0}, v_{k_0}) will be the $(\dot{t}_{l-1}, \dot{t}_l)$ point of

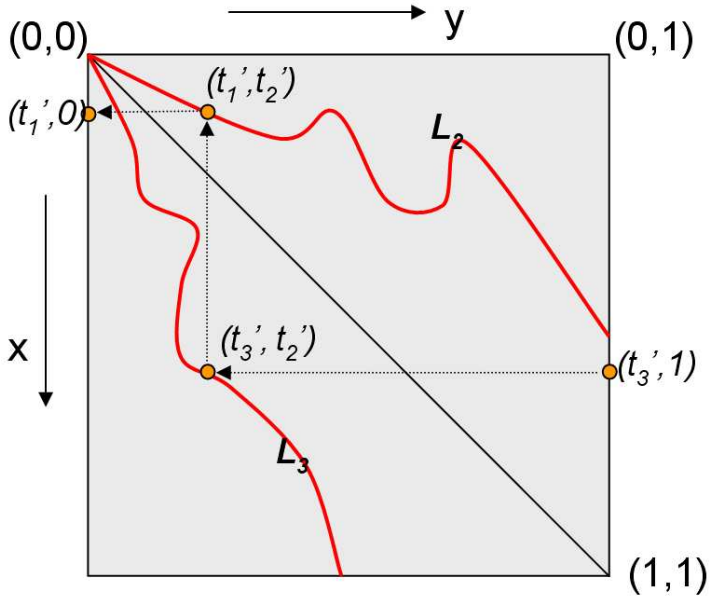
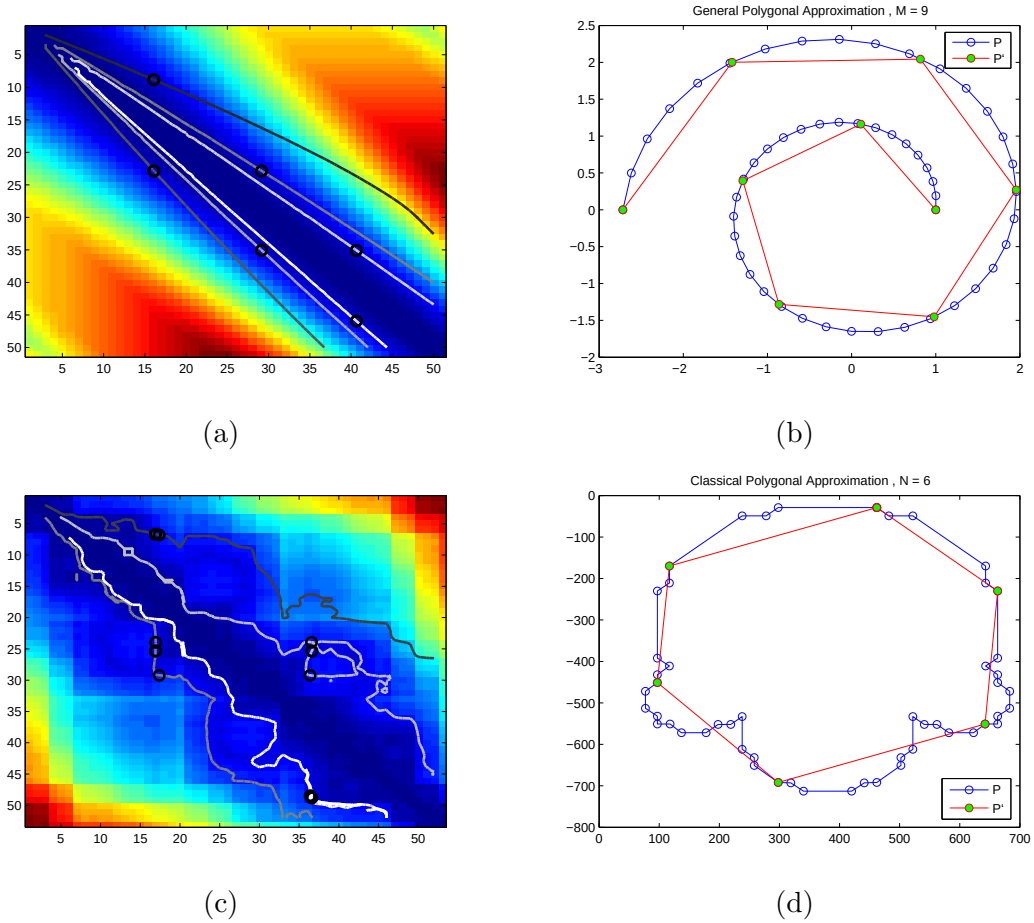


Fig. 4. An example of curve equipartition into 4 segments. It is shown the recursive computation of $\{\dot{t}_1, \dot{t}_2, \dot{t}_3\}$ and L_2, L_3 curves.



(a) (b)

(c) (d)

Fig. 5. Results of greedy version of the proposed algorithm. The estimated solutions are projected on $d(x, y)$ (left) with black cycles. The L_l curves are projected on $d(x, y)$, with gray colors, at both sides of diagonal $x = y$. On input polygonal curve P (right) is projected the estimated solution with minimum error ($\dot{P}$). **(c), (d)** The input curve P is the Teh and Chin curve [21].

the approximation.

2.5 Time and Space Requirements

The computation complexity of the greedy version of the EP method is $O(M \cdot N^2)$. However, we can compute at least one solution in $O(M \cdot N)$ steps using h_l curves. An important algorithm property is that the computation cost is independent of curve dimension n.

Concerning the computation cost of $d(p_k, p_l)$, it varies on the predefined error criterion and curve dimension. Under the LISE criterion and in the case of $n = 3$, the total greedy version cost is $O(M \cdot N^2)$, because the matrix $d(p_k, p_l)$ can be computed in $O(N^2)$ [4]. The worst case total cost of greedy version is $O(n \cdot N^3 + M \cdot N^2)$ and $O(n \cdot M \cdot N^2)$ for at least one solution computation.

The memory requirements of the algorithm are $O(M \cdot N)$, as we have to store intermediary results of the EP algorithm (the curves $L_l, l < M$). The storage of $d(p_k, p_l)$ matrix is optional and it needs $O(N^2)$ memory.

3 Experimental Results

In this section, the experimental results of the proposed algorithm are presented. The method has been implemented using Matlab and for our experiments, we have been using a Pentium 4 CPU at 2.8 GHz. In our experiments, the approximation error ϵ is normalized by the curve length. Moroever, in some test curves where the lengths of polygonal curve P line segments differ too much (e.g. Teh and Chin curve), we have added 50% more vertices on P curve making the lengths of polygonal curve P line segments to be almost equal. As we have discussed in section 2.2, the error between the proposed method and the solutions of the EE criterion descreases with $O(N^{-2})$ under the assumption that the length of P line segments are almost equal. So, we have developed a preliminary procedure of the proposed method. The preliminary procedure of adding more vertices on polygonal curve P is executed stepwise. In each iteration step, the new vertex a is added on P vertices so that a will be the mean point of the P line segment with the maximum length. The above procedure has cost $O(NlogN)$ and it does not affect the total algorithm computation cost. The above procedure is executed when the P line segments differ too much, i.e: their variance is higher than a predifined threshold.

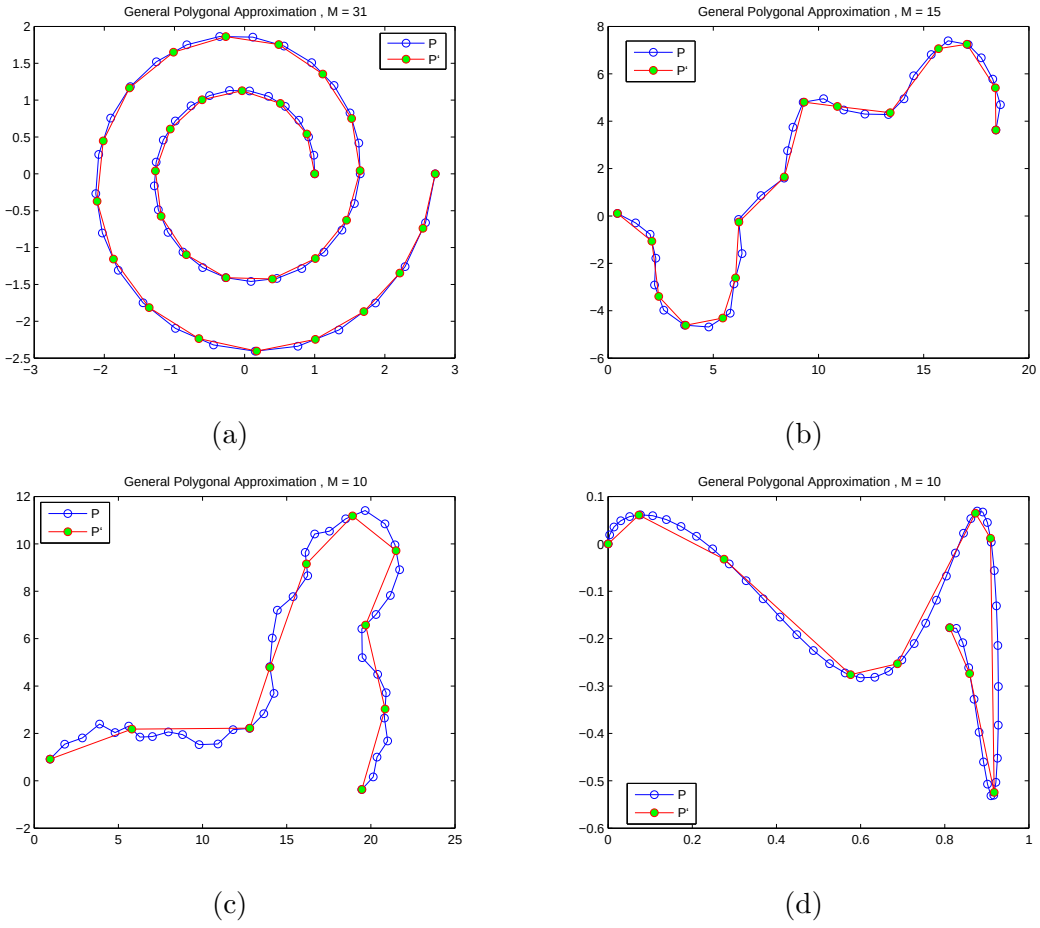


Fig. 6. $Min - \epsilon$ results of the proposed algorithm under the tolerance zone criterion for various curves and different values of M.

We have approximated various 2D polygonal curves under the tolerance zone or the LISE criterion (see Fig. 6, 7). The L_2 metric has been used. Results of the $Min - \epsilon$ problem are illustrated in Fig. 6, 7. The $\dot{P}$ vertices of Fig. 7 are close to P vertices. This can be explained as under the LISE criterion, the optimal solution of the PA problem is possibly the optimal solution of the GPA problem.

We compare our method with Douglas-Peuker line simplification algorithm (implementated by Matlab) and optimal PA method using the Efficiency criterion proposed by Rosin [19].

3.1 Comparison to Other Algorithms

In order to evaluate the quality of the proposed method, we have used the Efficiency criterion (Equation (7)) introduced by Rosin [19]. It measures how compact is the suboptimal polygonal representation of the curve, relative to the optimal polygon which incurs the same error.

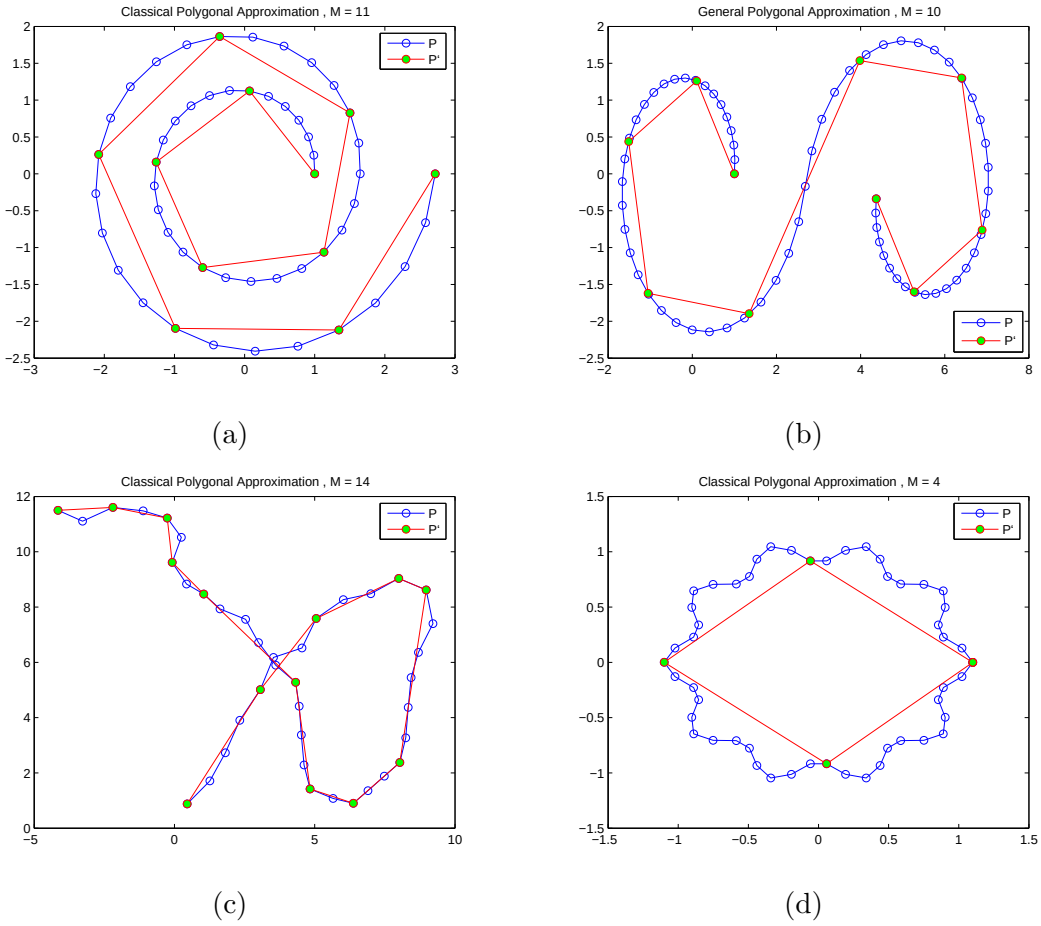


Fig. 7. $Min - \epsilon$ results of the proposed algorithm under LISE criterion for various curves and different values of M.

$$Efficiency = \frac{M_{opt}}{M_{approx}} \cdot 100 \tag{7}$$

where M_{approx} is the number of the approximating polygon vertices produced by the suboptimal algorithm and M_{opt} is the number of vertices that the optimal algorithm would require to produce the same error as the suboptimal algorithm. In a lot of cases, the proposed algorithm gives better results than the optimal PA algorithm, so the proposed algorithm becomes the optimal and its Efficiency is 100%.

We have compared the proposed GPA method with Douglas-Peuker line simplification algorithm and optimal PA method under the tolerance zone criterion with L_2 metric. Under the LISE criterion, the optimal solution of the PA problem is possibly the optimal solution of the GPA problem. Thus, we have not presented comparisons under this criterion as the results were almost the same.

The tables 2, 3, 4, 5, 6 and 7 present the Efficiency (E) results of the Douglas-Peucker, the optimal PA and the GPA algorithm for various curves under the tolerance zone criterion with L_2 metric. The mean value of Efficiency over the six tables is 81.06% for the Douglas-

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	17	88.2	14	78.6	13	76.9	12	75	7	100	6	83.3
Optimal PA	15	100	11	100	11	90.9	9	100	7	100	6	83.3
GPA	16	93.75	11	100	10	100	9	100	9	77.8	5	100

Table 2
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the Teh and Chin curve [21] (Fig. 8(a)). The mean value of Efficiency is 83.67% for the Douglas-Peucker algorithm, 95.7% for the optimal PA algorithm and 95.26% for the proposed GPA algorithm.

Peucker algorithm, 95.26% for the optimal PA algorithm and 97.49% for the proposed GPA algorithm.

Consequently, the proposed GPA algorithm approximates in a lot of cases better the given curve than the optimal PA algorithm. In addition, the Efficiency of the proposed algorithm was almost always higher than the Efficiency of the Douglas-Peucker algorithm, which solves the PA problem suboptimally. The performance of the proposed GPA algorithm depends on N, M and on the shape of the given curve P. When,

- the $\dot{P}$, derived by the EE criterion, is the optimal solution of the GPA problem
- and the EP method accurancy is high,

then the proposed solution will be better with a great probability than the optimal solution of the classical PA problem. The first constraint is satisfied for high values of M or low values of ϵ or smooth given polygonal curves. When ϵ is high or M is low, it is possible that the solution derived by the EE criterion to be a suboptimal solution of the GPA problem. The second constraint is satisfied for low values of M or smooth given polygonal curves. Therefore, when the given polygonal curve is smooth, the proposed algorithm yields with a great probability better results than the optimal PA method (see Table 4, 7). Otherwise, the result of which algorithm gives better solution is unpredictably changing with M (see Table 2, 3, 5, 6). Thus, in many cases the huge search space of proposed method and the high accurancy of the EP method under the EE criterion are sufficient to provide better solution than the optimal PA method (see Fig. 8). If the results have the same number of vertices, it does not hold that the solutions will be the same. An example of this situation is illustrated in Fig. 8(b).

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	14	78.6	11	90.9	10	70	10	60	8	75	5	80
Optimal PA	11	100	10	100	7	100	7	85.7	6	100	4	100
GPA	11	100	10	100	7	100	6	100	6	100	4	100

Table 3
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the curve of Fig. 8(b). The mean value of Efficiency is 75.75% for the Douglas-Peucker algorithm, 97.62% for the optimal PA algorithm and 100% for the proposed GPA algorithm.

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	19	84.2	17	76.5	15	73.3	13	76.9	11	81.8	10	70
Optimal PA	17	94.1	14	92.8	12	91.7	11	90.9	10	90	8	87.5
GPA	16	100	13	100	11	100	10	100	9	100	7	100

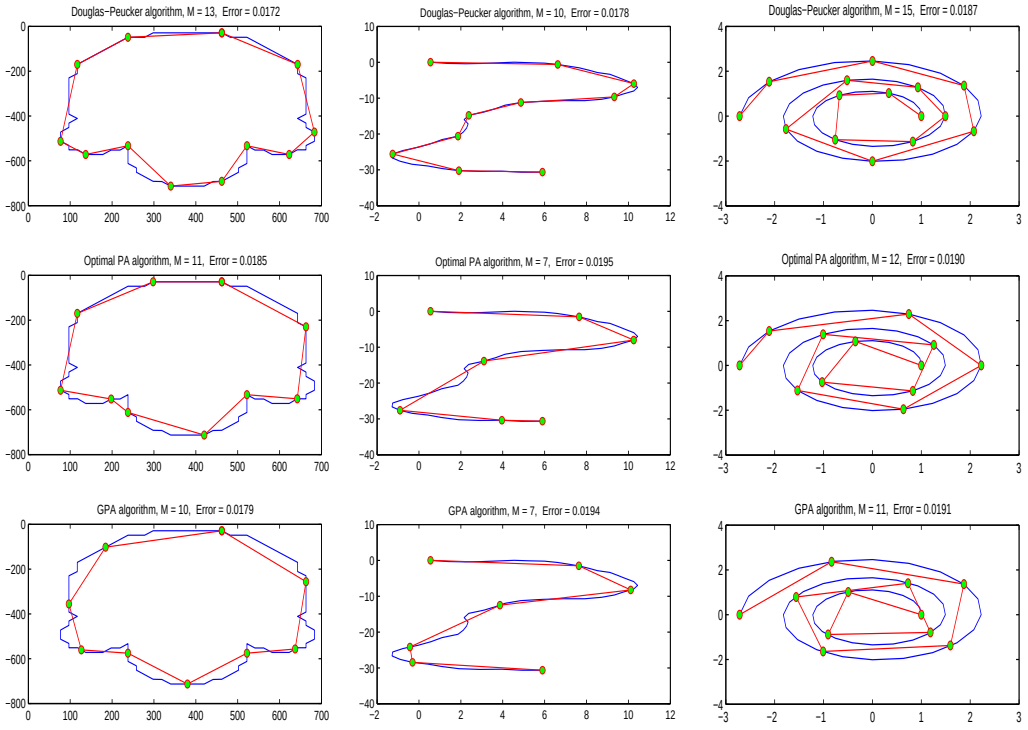
Table 4
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the curve of Fig. 8(c). The mean value of Efficiency is 77.12% for the Douglas-Peucker algorithm, 91.17% for the optimal PA algorithm and 100% for the proposed GPA algorithm.

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	18	77.8	14	71.43	9	88.9	8	75	7	85.7	6	83.3
Optimal PA	14	100	10	100	8	100	7	85.7	6	100	5	100
GPA	16	87.5	10	100	8	100	6	100	6	100	5	100

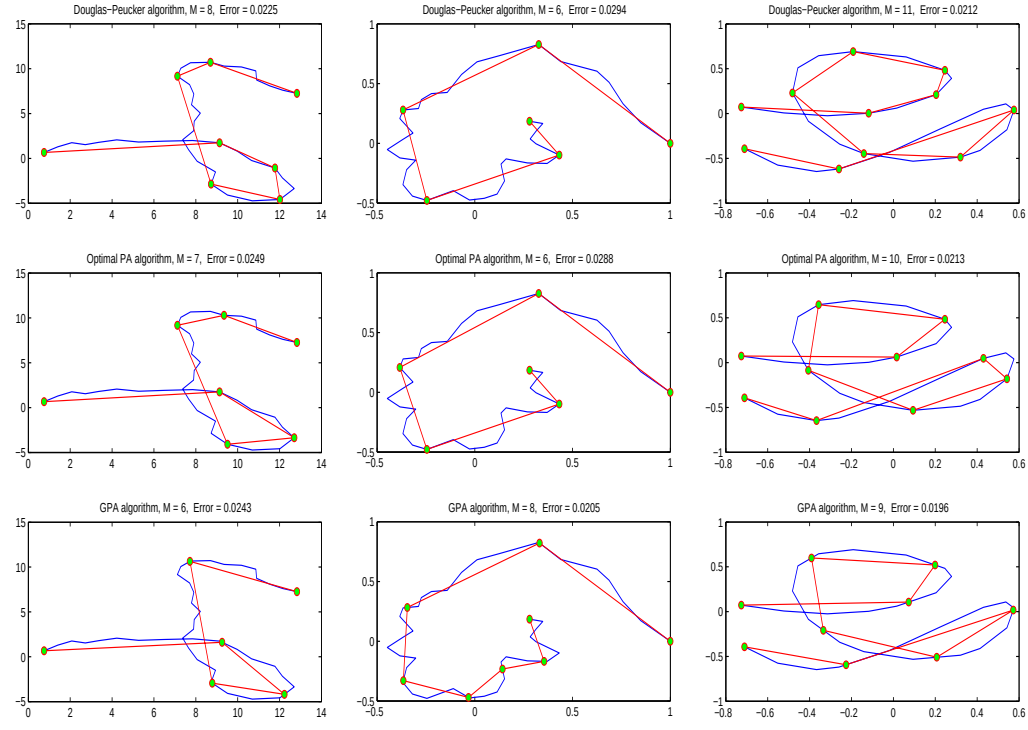
Table 5
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the curve of Fig. 8(d). The mean value of Efficiency is 80.35% for the Douglas-Peucker algorithm, 97.61% for the optimal PA algorithm and 97.92% for the proposed GPA algorithm.

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	21	85.7	14	85.7	13	69.2	9	88.9	6	100	6	100
Optimal PA	18	100	12	100	10	90	8	100	6	100	6	100
GPA	20	90	14	85.7	9	100	8	100	8	75	6	100

Table 6
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the curve of Fig. 8(e). The mean value of Efficiency is 88.25% for the Douglas-Peucker algorithm, 98.33% for the optimal PA algorithm and 91.78% for the proposed GPA algorithm.



(a) (b) (c)



(d) (e) (f)

Fig. 8. $Min - \#$ results of the Douglas-Peucker, optimal PA and proposed GPA algorithms under the tolerance zone criterion for different values of error ϵ. The curves P, $\dot{P}$ are projected with blue and green colors.

Algorithms	$\epsilon = 0.01$		$\epsilon = 0.015$		$\epsilon = 0.02$		$\epsilon = 0.025$		$\epsilon = 0.03$		$\epsilon = 0.05$	
	M	E	M	E	M	E	M	E	M	E	M	E
Douglas-Peucker	18	72.2	14	78.6	12	75	11	81.8	10	80	8	100
Optimal PA	14	77.8	11	100	10	90	10	90	9	88.9	7	100
GPA	13	100	11	100	9	100	9	100	8	100	7	100

Table 7
Comparison of the Efficiency (E) of the Douglas-Peucker, optimal PA and GPA algorithms for the curve of Fig. 8(f). The mean value of Efficiency is 81.27% for the Douglas-Peucker algorithm, 91.12% for the optimal PA algorithm and 100% for the proposed GPA algorithm.

4 Conclusions

In this paper, we have discussed the general polygonal approximation problem (GPA) in any dimensional space. We have proposed an algorithm based on the equipartition method. The solution, when the errors per segment are equal, is the optimal or a suboptimal of the GPA problem. The search space of PA problem (the vertices of P) is a subset of GPA problem search space (the total points of polygonal curve P). Therefore, the GPA problem solutions approximates better the given curve than the PA problem solutions.

In a lot of cases the huge search space of the proposed method and the high accurancy of the EP method under the EE criterion are sufficient to provide better solution than the optimal PA method with about the same computation cost. As our experiments show, in mean case, the proposed GPA algorithm Efficiency was 2% higher than the Efficiency of the optimal PA algorithm. However, there are cases where the best curve approximation is given by the optimal PA algorithm. It holds that if the $\dot{P}$, derived by the EE criterion, is the optimal solution of the GPA problem and the EP method accurancy is high, then the proposed solution will be better with a great probability than the optimal solution of the PA problem. This is getting true for smooth given polygonal curves. Otherwize, the result of which algorithm gives better solution is unpredictable.

Furthermore, our method is very flexible on changes of error criteria and given curve dimension, in contrast with the most of the proposed algorithms on the literature, which are strongly associated with a specific error criterion or curve dimension. Moreover, the method is inductive. Thus, when it is executed for M, it solves the problem for any number of segments less than M without additional computation cost. The $min - \epsilon$ *problem* and the $min - \#$ *problem* are solved with the same computation cost. The method can be used in cases where it is critical to minimize the approximation error obtaining better results than the optimal solutions of PA problem. An important feature of the method is that the equal

approximation error is almost the same per segment, so the $\dot{P}$ line segments are equivalent on the approximation.

References

[1] H. Imai, M. Iri, Polygonal approximations of a curve (formulations and algorithms), Computational Morphology (1988) 71–86.

[2] D. Chen, O. Daescu, Space-efficient algorithms for approximating polygonal curves in two-dimensional space,, in: Proc. of the 4th Ann. Int. Conf. on Computing and Combinatorics, 1998.

[3] B. Ray, K. Ray, A non-parametric sequential method for polygonal approximation of digital curves, Pattern Recognition Letters 15 (1994) 161–167.

[4] K.-L. Chung, W.-M. Yan, W.-Y. Chen, Efficient algorithms for 3-d polygonal approximation based on lise criterion, Pattern Recognition 35 (2002) 2539–2548.

[5] P. S. Heckbert, M. Garland, Survey of polygonal surface simplification algorithms, in: SIGGRAPH, 1997.

[6] A. Kolesnikov, Efficient algorithms for vectorization and polygonal approximation, Ph.D. thesis, University of Joensuu, Finland (2003).

[7] R. Weibel, Generalization of spatial data: principles and selected algorithms, Algorithmic Foundations of Geographic Information System.

[8] W. Chan, F. Chin, On approximation of polygonal curves with minimun number of line segments or minimun error, International Journal of Computational Geometry Applications (1996) 59–77.

[9] A. Katsaggelos, L. Kondi, F. Meier, J. Ostermann, G. Schuster, MPEG-4 and ratedistortion -based shape-coding techniques, Proceedings of the IEEE, Special Issue on Multimedia Signal Processing 86 (6) (1998) 1126–1154.

[10] J. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, Pattern Recognition Letters 15 (1994) 743–750.

[11] M. Salotti, An efficient algorithm for the optimal polygonal approximation of digitized curves, Pattern Recognition Letters 22 (2001) 215–221.

[12] G. Barequet, D. Chen, O. Daescu, M. Goodrich, J. Snoeyink, Efficiently approximating polygonal paths in three and higher dimensions, Pattern Recognition 35 (2002) 2539–2548.

[13] K. Varadarajan, Approximating monotone polygonal curves using uniform metric, in: Proc. 12th Ann. ACM Symp. on Computational Geometry, 1996.

[14] P. Agarwal, S. Har-Peled, N. Mustafa, Y. Wang, Near-linear time approximation algorithms for curve simplification in two and three dimensions, in: Proc. of the 10th European Symposium on Algorithms (ESA '02), 2002.

[15] A. Kolesnikov, P. Franti, Reduced-search dynamic programming for approximation of polygonal curves, Pattern Recognition Letters 24 (2003) 2243–2254.

[16] H. Zhang, J. Guo, Optimal polygonal approximation of digital planar curves using meta-heuristics, Pattern Recognition 34 (2001) 1429–1436.

[17] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Computer Graphics and Image Processing 1 (1972) 244–256.

[18] D. Douglas, T. Peucker, Algorithm for the reduction of the number of points required to represent a line or its caricature, The Canadian Cartographer 10 (1973) 112–122.

[19] P. Rosin, Assessing the behaviour of polygonal approximation algorithms, Pattern Recognition 36 (2003) 505–518.

[20] C. Panagiotakis, G. Georgakopoulos, G. Tziritas, The curve equipartition problem, submitted to Computational Geometry.
URL `http://www.csd.uoc.gr/~cpanag/papers/EP.pdf`

[21] C. Tex, R. Chin, On the detection of dominant points in digital curves, IEEE Trans. PAMI 11 (1989) 859–872.