Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1282–1293
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Polygonal approximation of closed discrete curves

Alexander Kolesnikov[a,b,*], Pasi Fränti[a]

[a] *Department of Computer Science, University of Joensuu, Joensuu, Finland*
[b] *Institute of Automation and Electrometry, Novosibirsk, Russia*

Received 8 December 2005; received in revised form 2 August 2006; accepted 5 September 2006

Abstract

Optimal polygonal approximation of closed curves differs from the case of open curve in the sense that the location of the starting point must also be determined. Straightforward exhaustive search would take *N* times more time than the corresponding optimal algorithm for an open curve, because there are *N* possible points to be considered as the starting point. Faster sub-optimal solution can be found by iterating the search and heuristically selecting different starting point at each iteration. In this paper, we propose to find the optimal approximation of a cyclically extended closed curve of double size, and to select the best possible starting point by search in the extended search space for the curve. The proposed approach provides solution very close to the optimal one using at most twice as much time as required by the optimal algorithm for the corresponding open curve.

Keywords: Polygonal approximation; Closed contour; Dynamic programming

1. Introduction

Approximation of polygonal discrete curves aims at finding a subset of the original vertices so that a given objective function is minimized. The problem can be formulated in two ways:

(a) *Minimum-distortion problem*: given an *N*-vertex polygonal curve *P*, approximate it by another polygonal curve *Q* with a given number of segments *M* so that the approximation error is minimized.
(b) *Minimum-rate problem*: given an *N*-vertex polygonal curve *P*, approximate it by another polygonal curve *Q* with minimum number of segments so that the approximation error does not exceed a given maximum tolerance ε.

Sometimes the problems are referred as *min-*ε and *min-# problem*, respectively [1]. In the case of *open* curve, the problems can be solved by dynamic programming (DP) algorithm [1–9], or by A*-search algorithm [10]. The time complexity of these algorithms varies from $O(N^2)$ to $O(N^3)$. Heuristic algorithms also exist: *split* [11–13], *merge* [14–20], *dominant points detection* [21–24], *sequential tracing* [25–27], *genetic algorithms* [28–31], *tabu search* [31,32], *ant colony methods* [33,34], *particle swarm method* [35]. These algorithms are fast but they lack the optimality.

In the case of *closed* curve, we have to find optimal allocation of all approximation nodes including also the starting point. A straightforward solution is to try all *N* vertices of *P* as a starting point for an algorithm designed for open curve, and to choose the one with the minimal number of segments [6] or the minimal error [8]. Actually, for the minimum-distortion problem the number of vertices to be tested is $(N-M)$ [8], but this is still $O(N)$ for small *M*.

It was shown in Ref. [2] that the minimum-rate problem for closed curve can be solved as the all-pairs shortest path problem for a graph of *N* vertices by an algorithm of complexity $O(N^3)$. Considering that the time complexity of the optimal minimum-rate algorithm for open curve is $O(N^2)$ [2], we have the same proportion between the processing time for closed and open curve, which is $O(N)$.

* Corresponding author. Tel.: +358 0 46 810 4853; fax: +358 0 13 251 7955.
E-mail addresses: koles@cs.joensuu.fi (A. Kolesnikov), franti@cs.joensuu.fi (P. Fränti).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.09.002

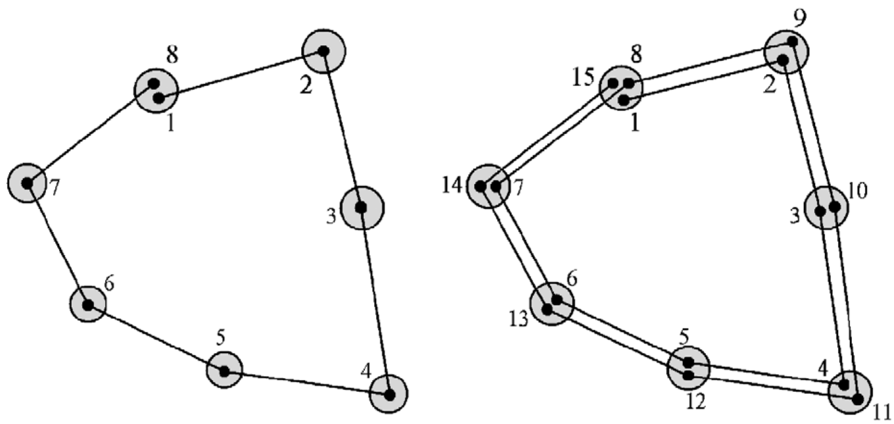

Fig. 1. Illustration of an input closed curve P_1 (left), and twofold curve P_2 constructed from the vertices of P_1 in a cyclic manner (right).

In Refs. [4,9] the minimum-rate problem with L_2 [4] and L_1 [9] error measures were considered, and algorithms for open curve based on the shortest path problem were proposed. For the closed curve, they proposed to test each vertex of a segment of the curve as a starting point. In this way, the number of vertices to be tested is reduced from $O(N)$ to $O(N/M)$, on average, but this is still large, especially for the problems with a small values of M. On the other hand, the optimal selection of the starting point is more critical especially for small values of M.

Two fast heuristic methods were proposed in Refs. [36,37] for the starting point selection for minimum-distortion problem. First, polygonal approximation with the optimal algorithm for an open curve [8] is performed using a random starting point. A new starting point is selected among the approximating nodes of Q based on certain rules, and this point is used for the second run of the optimal algorithm [8].

Another possibility would be to ignore the optimality at the first place and use the optimal algorithm developed for open curve without considering the problem of starting point.

To sum up, existing heuristics for starting point selection are sub-optimal [36,37] whereas the optimal choice is time consuming [2,4,6,8,9]. Thus, the problem of finding the optimal approximation efficiently for a closed contour is still unsolved. The existing heuristic approaches can be summarized by the fact that they iterate the search by using one of the approximation nodes as a new starting point for the subsequent iterations. After every run, however, the information from the previous iteration is lost, and the search starts from scratch.

In this paper, we propose to use the information from the previous run by performing dynamic programming search in an *extended state space* constructed for a twofold curve (see Fig. 1). All parts of the shortest path in the extended state space have be analyzed to find such a sub-path with *conjugate states* that provides minimum for the cost function in question (approximation error or the number of segments). Pair of *conjugate states* in the extended state space defined by two points on the twofold curve which are produced by the same point on the input closed curve. It is expected that the proposed approach provides better solution than the existing heuristic algorithms in a comparable time because of using all available information.

The proposed approach is illustrated in Fig. 1. Approximation with DP algorithm for open curve with the vertex 1 as the starting point is given by segments 1-2-4-7-8. Optimal starting point and the corresponding approximation of the original contour can now be found by analyzing the approximation of the twofold curve P_2. In this example, we obtain the solution as the segments 2-4-7-2.

The rest of the paper is organized as follows. In Section 2, we give the problem formulation, recall the dynamic programming approach of [8] for minimum-distortion approximation of open curves, including approximation algorithm with reduced search [38], and present a new algorithm for the optimization of the starting point by searching in the extended state space. In Section 3, we apply the proposed approach to the case of minimum-rate problem. Experimental results and discussions are then given in Section 4, and conclusions are drawn in Section 5.

2. Minimum-distortion problem for closed curve

2.1. Problem formulation

We define a closed N-vertex polygonal curve P in two-dimensional space as the ordered set of vertices $P = \{p_1, p_2, \ldots, p_N\} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where the last vertex coincides with the first one: $p_N = p_1$. The problem is stated as follows: approximate the closed polygonal curve P by another closed polygonal curve Q with a given number of linear segments M so that the total distortion (approximation error) $E(P, M)$ is minimized. The optimal approximation Q of a closed curve P is the set of nodes $\{q_1, q_2, \dots, q_{M+1}\}$ that minimizes the cost function $E(P, M)$ with L_2 measure:

$$E(P, M) = \min_{\{q_m\}} \left\{ \sum_{m=1}^{M} e_2(q_m, q_{m+1}) \right\}, \tag{1}$$

here $q_{M+1} = q_1 = p_1 = p_N$. The problem also can be referred as *rate-constrained problem*. Error of the approximation of curve segment $\{p_i, \dots, p_j\}$ with the corresponding linear segment (q_m, q_{m+1}) is the sum of squared Euclidean distances from each vertex $p_k \in \{p_i, \dots, p_j\}$ to the line segment (q_m, q_{m+1}):

$$e_2(p_i, p_j) = \sum_{k=i+1}^{k=j-1} d^2(k; p_i, p_j), \tag{2}$$

where $q_m = p_i$ and $q_{m+1} = p_j$.

The Euclidean distance $d(k; p_i, p_j)$ from a point $p_k = (x_k, y_k)$ to the approximating line segment $y = a_{i,j}x + b_{i,j}$ can be calculated by the following expression:

$$d(k; i, j) = \frac{|y_k - a_{i,j}x_k - b_{i,j}|}{\sqrt{1 + a_{i,j}^2}}. \tag{3}$$

The coefficients $a_{i,j}$ and $b_{i,j}$ of the line are calculated from the coordinates of the end points p_i and p_j:

$$a_{i,j} = (y_j - y_i)/(x_j - x_i),$$
$$b_{i,j} = y_i - a_{i,j}x_i. \tag{4}$$

To solve the optimization task we first recall the optimal dynamic programming algorithm of Perez and Vidal [8] for open curves.

2.2. Dynamic programming approach

Optimal approximation of an open N-curve P_1 with M line segments is the set nodes $\{q_2, q_3, \dots, q_M\}$ that minimizes the cost function $E(P_1, M)$:

$$E(P_1, M) = \min_{\{q_m\}} \left\{ \sum_{m=1}^{M} e_2(q_m, q_{m+1}) \right\}, \tag{5}$$

here we assume that the first and the last approximation nodes are fixed: $q_1 = p_1$, $q_{M+1} = p_N$.

Let us define a discrete two-dimensional *state space* $\Omega_1 = \{\omega(n, m) : n = 1, \dots, N;\ m = 0, \dots, M\}$. Every point $\omega(n, m)$ in the state space Ω_1 represents the subproblem of approximating of an n-vertex polygonal curve $\{p_1, p_2, \dots, p_n\}$ by m line segments. The state space Ω_1 is bounded by left $L(m)$, right $R(m)$, bottom $B(n)$ and top $T(n)$ borders as follows (see Fig. 2):

$$L(m) = \begin{cases} m + 1, & m = 0, 1, \dots, M - 1, \\ N, & m = M, \end{cases}$$

$$R(m) = \begin{cases} 1, & m = 0, \\ N - M + m, & m = 1, 2, \dots, M, \end{cases}$$

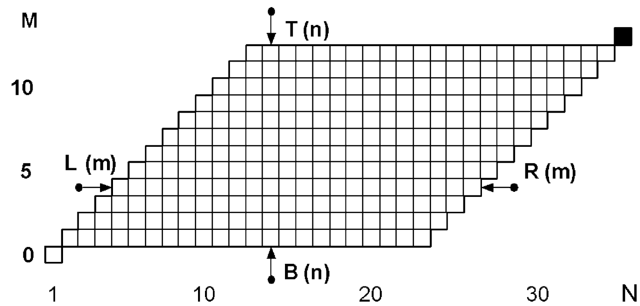

Fig. 2. Illustration of the state space Ω_1 for approximation of an N-vertex curve P_1 with M line segments.

$$B(n) = \begin{cases} 0, & n = 1, \\ 1, & n = 2, \dots, N - M, \\ n - N + M, & n = N - M + 1, \dots, N, \end{cases}$$

$$T(n) = \begin{cases} n - 1, & n = 1, \dots, M, \\ M - 1, & n = M + 1, \dots, N - 1, \\ M, & n = N. \end{cases}$$

The complete problem is represented by the *goal state* $\omega(N, M)$ (see dark grey square on Fig. 2). In the state space Ω_1, we define a cost function $C(n, m)$ as the error of the optimal approximation for the n-vertex curve $\{p_1, \dots, p_n\}$ by m linear segments.

For solving the problem in question we have to find the weighted shortest path from the start state $\omega(1, 0)$ to the goal state $\omega(N, M)$. The optimization problem can be solved by the dynamic programming algorithm with the following recursive expressions for cost function $C(n, m)$ for all $\omega(n, m) \in \Omega_1$:

$$C(n, m) = \min_{L(m-1) \leqslant j < n} \{C(j, m - 1) + e^2(p_j, p_n)\},$$
$$A(n, m) = \arg\min_{L(m-1) \leqslant j < n} \{C(j, m - 1, j) + e^2(p_j, p_n)\}. \tag{6}$$

Here $A(n, m)$ is the *parent state* that provides the minimum value of the cost function $C(n, m)$ for a state $\omega(n, m)$, and $e^2(p_j,\ p_n) = e^2(q_m,\ q_{m+1})$ is the approximation error of the curve segment $\{p_i, \dots, p_j\}$ with the corresponding line segment (q_m, q_{m+1}). The dynamic programming algorithm is detailed in Fig. 3. The optimal path $H(m)$ in the state space Ω_1 is restored by backtracking from the goal state $\omega(N, M)$ to the start state $\omega(1,0)$.

The time complexity of the DP algorithm is $O(M(N - M)^2)$. For the sake of simplicity, we refer it as $O(MN^2)$ because usually $M \ll N$. The space complexity is defined by the area of the state space, which is $O(MN)$.

2.3. Reduced search approach

The main idea behind the *reduced search* (RS) algorithm [38] is to reduce the time consuming search in the state space by exploring only a small but relevant part of the state space. To define which part should be explored, an initial solution is generated using any fast heuristic algorithm. The solution

```
Initialization:
C(1,0) = 0
FOR n = 2 TO  N  DO
   C(n,0) = ∞
END

Recursion:
FOR m = 1  TO  M  DO
   FOR n = L(m)  TO  R(m)  DO
      c_min =  ∞
      FOR  j = L(m–1) TO n-1 DO
         c = C(j, m–1-B(n)) + e^2(p_j, p_n)
         IF(c < c_min)
            C_min = c,
            J_min = j
         ENDIF
      END
      D(n, m – B(n)) = c_min
      A(n, m – B(n)) = j_min
   END
END
E = C(N,M)

Backtracking to find the optimal path H:
H(M+1)= N
FOR m = M+1 TO 2 DO
   H(m–1) = A(h(m), m – B(H(m)))
END
```

Fig. 3. Pseudo code of the dynamic programming algorithm for minimum-distortion problem.

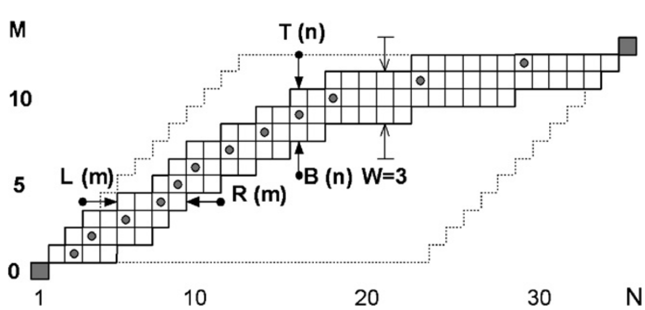

Fig. 4. Illustration of the bounding corridor of width $W = 3$ in the state space Ω_1. The reference path H_0 is marked with grey circles.

defines a *reference path* in the state space Ω_1. A *bounding corridor* of width W is then constructed along this path, and the minimum cost path is searched within the corridor using DP algorithm (see Figs. 4 and 5).

Any fast heuristic algorithm for the minimum-distortion problem can be used to construct the reference path H_0. We use algorithm *Merge*-L_2 [19,20] to construct a reference path because this algorithm gives smaller approximation error with measure L_2. For closed curve the advantage of the merge algorithm is that the result does not depend on the choice of the starting point because any two adjacent segments can be merged. The complexity of the merge algorithm is $O(N \log N)$.

The time complexity of the near-optimal RS algorithm is $O(W^2N^2/M)$, and the speed-up is proportional to $(W/M)^2$ in comparison to the time complexity of the full search. The space complexity is $O(WN)$. We set the corridor width as $W = \sqrt{M}$ to keep the processing time fixed to $O(N^2)$ for all M.

2.4. Solution for closed curve

In heuristic approach [36,37] for closed curve approximation, a new starting point is selected among the nodes obtained at the preliminary run of the optimal approximation algorithm [8]. This provides a solution that is close to the optimal one, especially for a large number of segments. However, for a small number of M, the solution can be far from the optimal. The main reason for non-optimality is that after the first run, the search starts again from scratch and loses the information of the previous run.

To solve the problem, we offer to perform the dynamic programming search in state space Ω_2 constructed for an open $(2N - 1)$-vertex curve P_2, obtained from the input curve P. In other words, we continue the search along closed contour P beyond the end point until we reach the end point second time. To reduce the processing time we apply the reduced search approach [38].

The algorithm consists of four steps:

(1) Find approximation Q_R for P with *Merge*-L_2 algorithm, select a starting point for P_2 among the nodes of Q_R, and get a reference solution;
(2) Construct the bounding corridor in the state space Ω_2 using the reference solution;
(3) Perform DP search in the bounding corridor to construct solutions for all sub-problems;
(4) For every state in the second half of the bounding corridor $(n \geqslant N)$ backtrack the shortest path to find a such sub-path with conjugate states that provides minimum difference of the cost function between the states. The starting point is the first node of the found sub-path.

Let us consider the last step of the algorithm in details. We backtrack the optimal path $H(\omega_g) = \{h(1), \ldots, h(m_g)\}$ from the state ω_g to the start state $\omega(1, 0)$; here $\omega_g = (n_g, m_g) \in \Omega_2$ and $n_g \geqslant N$; $m_g \geqslant M$. We refer two states $\omega_1 = \omega(n_1, m_1)$ and $\omega_2 = \omega(n_2, m_2)$ on the optimal path $H(\omega_g)$ as conjugate states if (1) they represent two vertices p_{n_1} and p_{n_2} of twofold curve P_2 produced by the vertex p_{n_1} in the input

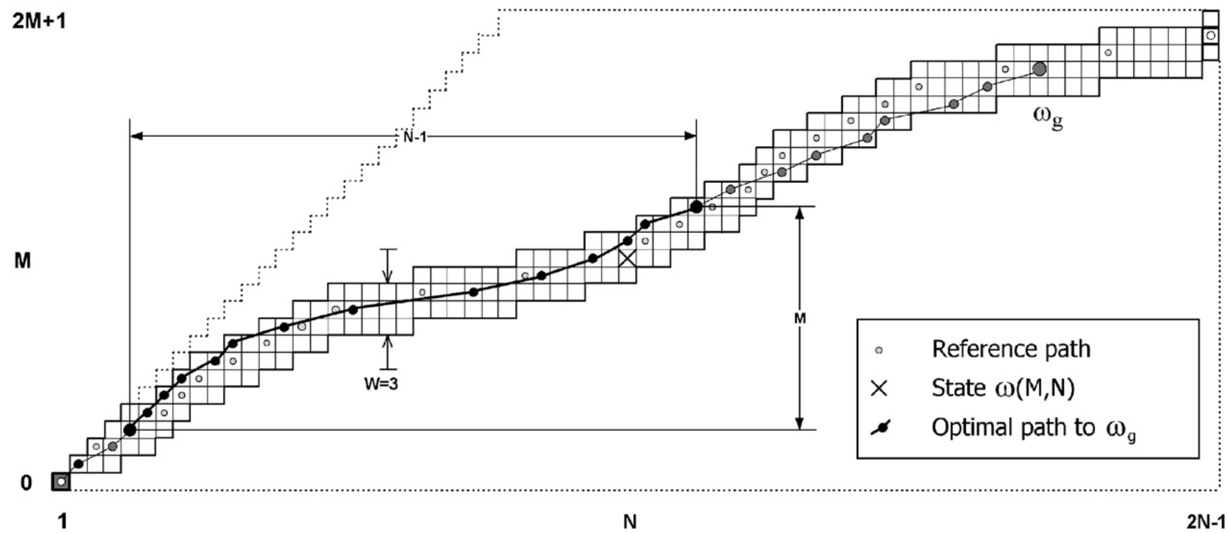

Fig. 5. Illustration of the bounding corridor in the state space Ω_2 with corridor width $W=3$. Sub-path of the optimal path $H(\omega_g)$ with conjugate states is emphasized.

closed curve P, and (2) the difference of the segments number between the states is M, which is the case if $n_2=n_1+N$, and $m_2=m_1+M$, (see Fig. 5). The sub-path of $H(\omega_g)$ between the two *conjugate states* corresponds to the optimal approximation Q of the closed curve P with M linear segments starting from the point n_1. The approximation error $E(\omega_{1,}\omega_2)$ for Q is then given by the difference of the cost function for the conjugate states ω_1 and ω_2:

$$E(\omega_{1,}\omega_2) = C(\omega_2) - C(\omega_1), \tag{7}$$

where $\omega_1 = \omega(h(m-M),\ m-M)$, and $\omega_2 = \omega(h(m), m)$.

To find the optimal starting point n_1, we have to find a pair of conjugate states on the path $H(\omega_g)$ for all possible goal states ω_g so that it minimizes the approximation error $E(\omega_1, \omega_2)$:

$$E_{\text{opt}} = \min_{\{\omega_g\}} \left\{ \min_{M \leqslant m \leqslant m_g} \{C(h(m), m) - C(h(m-M), m-M)\} \right\}, \tag{8}$$

$$n_{\text{opt}} = \arg\min_{\{\omega_g\}} \left\{ \min_{M \leqslant m \leqslant m_g} \{C(h(m), m) - C(h(m-M), m-M)\} \right\} - (N-1). \tag{9}$$

Among all shortest paths there is at least one path with conjugate states: path for the state $\omega_g = \omega(N, M)$; it is marked by '×' in Fig. 5.

The complexity of the search for optimal sub-path with conjugate states is $O(WMN)$, which is defined by the total number of goal states $O(NW)$, and the number of steps $O(M)$ to check a single path.

3. *Minimum-rate problem* for closed curve

3.1. Problem formulation

The *minimum-rate problem* is stated as follows: approximate the N-vertex closed polygonal curve P by another closed polygonal curve Q with a minimum number of line segments M so that the approximation error $E(P)$ is less than a given error bound ε. The approximation error E with measure L_∞ is defined as the maximum Euclidean distance from the vertices of the curve P to the approximation line segments of curve Q:

$$E(P) = \max_{1 \leqslant m \leqslant M} \{d(q_m, q_{m+1})\}, \tag{10}$$

where $d(p_i, p_j) = \max_{i \leqslant k \leqslant j}\{d(k; p_i, p_j)\}$, and $q_m = p_i$, $q_{m+1} = p_j$. The problem can be referred as *distortion-constrained problem*.

3.2. Solution for fixed end points

To find minimum-rate approximation for the closed curve P with a fixed starting point, a feasibility graph $G_1 = G(P, \varepsilon)$ is constructed on vertices of the curve P for the given error tolerance ε [1–7] (see Fig. 6). Nodes $V = \{v_1, v_2, \ldots, v_N\}$ of the graph G_1 are vertices $\{p_1, p_2, \ldots, p_N\}$ of the curve P. A pair of nodes v_i and v_j is connected by an edge $e_{i,j}$ if the approximation error $d(p_i, p_j)$ for the curve segment $\{p_i, p_{i+1}, \ldots, p_j\}$ by the line segment (p_i, p_j) is less than a given error tolerance: $d(p_i, p_j) \leqslant \varepsilon$.

The solution for the problem is the shortest path in the feasibility graph G_1. To find the shortest path in directed acyclic graph, we introduce one-dimensional discrete state space $\Omega_1 = \{\omega(n) : n = 1, \ldots, N\}$ of size N (see Fig. 7). Every point $\omega(n)$ in the state space Ω_1 represents the

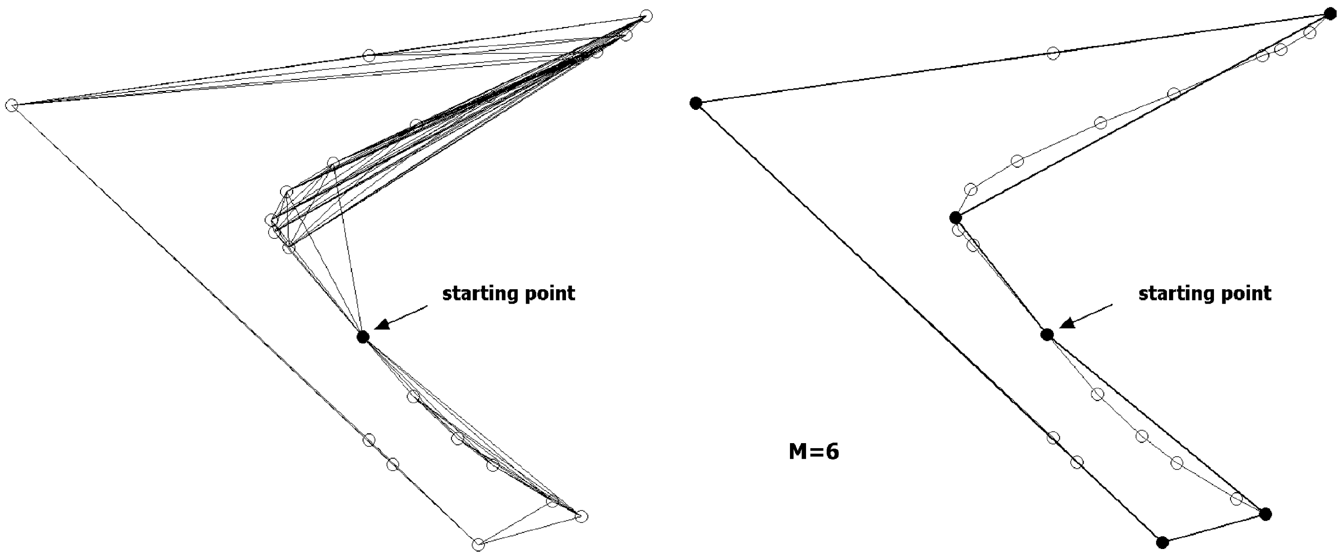

Fig. 6. Feasibility graph $G_1 = G(P, \varepsilon)$ constructed on the test shape #1 with a fixed starting point for error tolerance $\varepsilon = 30$ (left), and the solution with $M = 6$ line segments as the shortest path in the graph G_1 (right).

```
Initialization:
C(1) = 0
Recursion:
FOR n = 2 TO N  DO
    R (n) = ∞
    FOR j = n-1TO 1 DO
        IF edge(j,n) ∈ G
        THEN
            IF C(n) > C(j) +1
            THEN
                C(n) = C(j)+1
                A(n) = j
            ENDIF
        ENDIF
    END
END
Backtracking to find the optimal path H:
M=R(N)
H(M+1)=N
FOR m = M+1 TO 2 DO
    H(m–1) = A(H(m))
END
```

Fig. 7. Pseudo code of the dynamic programming algorithm for the shortest path in the feasibility graph G_1.

sub-problem of the shortest path finding from the first node v_1 of G_1 to the node v_n. The cost function $C(n) \equiv C(\omega(n))$ for state $\omega(n)$ is given as the minimum number of edges in the shortest path. The minimum number of approximating line segments M is given by cost function value for the *goal state* $\omega(N) : M = C(N)$. The cost function $C(n)$ is calculated for all $n = 1, \ldots, N$ by dynamic programming:

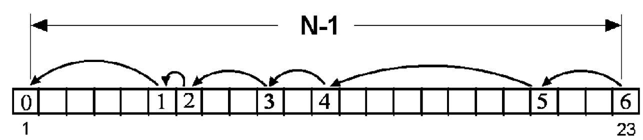

Fig. 8. State space Ω_1 for test shape #1. The shortest path is labelled with arrows and the values of the cost function $C(n)$. The minimum number of approximating segments is $M = C(23) - C(1) = 6 - 0 = 6$.

$$C(n) = \min_{\substack{1 \leqslant j < n \\ (v_j, v_n) \in G_1}} \{C(j) + 1\}, \tag{10a}$$

$$A(n) = \arg\min_{\substack{1 \leqslant j < n \\ (v_j, v_n) \in G_1}} \{C(j) + 1\}. \tag{10b}$$

After completing the DP search, optimal solution H is backtracked using the array of parent states $A(n)$ (see Figs. 7, 8). The complexity of the minimum-rate algorithm is defined by the complexity of the algorithm for the feasibility graph construction, which is $O(N^2)$ [2], and by the complexity of the shortest path construction, which is also $O(N^2)$.

3.3. Solution for closed curve

With heuristic approach [36,37] the optimal result can be obtained by two iterations of the algorithm for open curve, and the corresponding processing time is $T_H = 2T_1$, where T_1 is processing time for one run of the optimal algorithm for the open curve. To make the new algorithm competitive

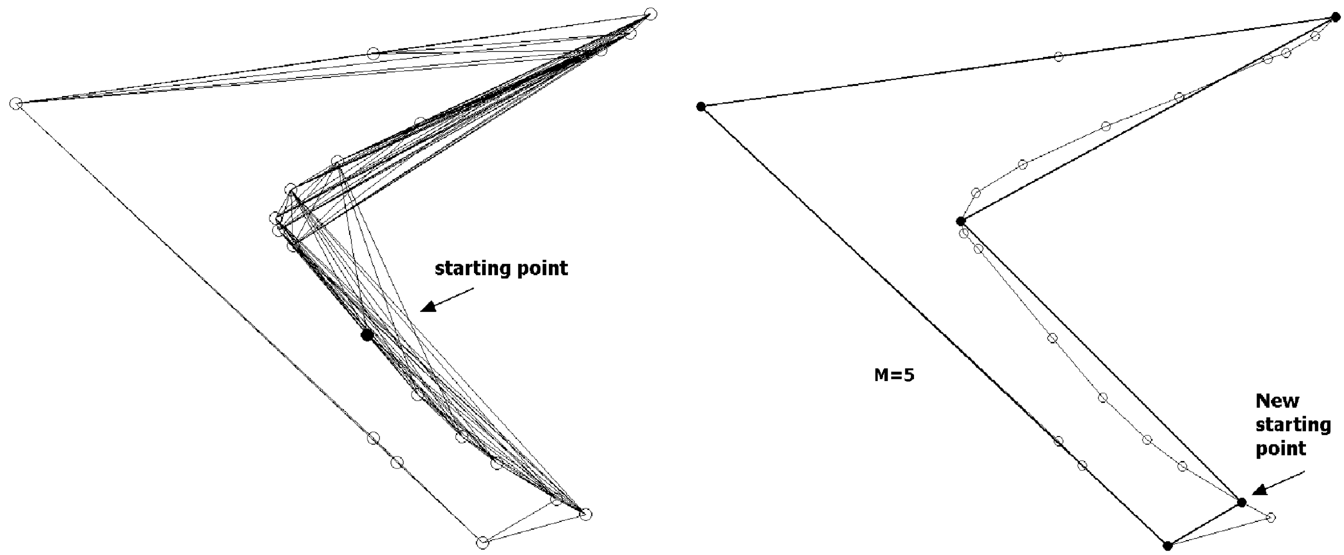

Fig. 9. Feasibility graph $G_2 = G(P_2, \varepsilon)$ constructed on the twofold curve P_2 for error tolerance $\varepsilon = 30$ (left), and the optimal approximation with $M = 5$ line segments as the optimal sub-path with conjugate state.

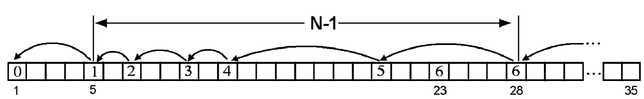

Fig. 10. Extended state space Ω_α for curve P_α for test shape #1. The shortest path is labelled with arrows and values of the cost function $C(n)$. Conjugate states $\omega(5)$ and $\omega(28)$ are emphasized. The minimum number of approximating segments is $M = C(28) - C(5) = 6 - 1 = 5$.

with the heuristic approaches we have to achieve equal or better result but faster.

Let us consider an α-fold curve P_α, and construct a feasibility graph G_α, where α is a *non-integer* number: $\alpha \geqslant 1$ (see Fig. 9). In other words, we continue the cyclic search along the curve P_2 up to $\lceil \alpha N \rceil$th vertex for the shortest path construction in the extended feasibility graph G_α. To bound the processing time T_α by an upper limit $2T_1$, the parameter α is restricted to $1 \leqslant \alpha \leqslant 2$.

To find the shortest path H in the directed acyclic graph G_α, we introduce an extended one-dimensional state space $\Omega_\alpha = \{\omega(n) : n = 1, \ldots, \lceil \alpha N \rceil\}$ of size $\lceil \alpha N \rceil$, and define cost function $C(n)$ in the space (see Fig. 10). We define two states $\omega(i)$ and $\omega(j)$ as *conjugate*, if there is exactly $(N-1)$ vertices between vertices p_i and p_j of $P_\alpha : j = i + N - 1$. In other words, the conjugate states $\omega(i)$ and $\omega(j)$ correspond to vertices p_i and p_{i+N-1} on α-fold curve P_α that are produced by the same vertex p_i of the input curve P. In the state space Ω_1, there is only one pair of conjugate states: $\omega(1)$ and $\omega(N)$, see Fig. 8.

The proposed algorithm for closed curve consists of three steps:

(1) Construct an extended feasibility graph G_α for α-fold curve P_α.
(2) Construct the shortest paths to all goal nodes in the graph G_α;
(3) For every goal node $v_k \in G_\alpha$ backtrack the shortest path $H(v_k)$ and find such a pair of nodes v_i and v_j on the path $H(v_k)$ that corresponding states $\omega(i)$ and $\omega(j)$ are *conjugate* and the number of edges in the sub-path connecting the nodes v_i and v_j is minimal.

Consider the shortest path $H(v_k)$ for some goal node $v_k \in G_\alpha$ and $k \geqslant N$. The number of edges in the sub-path between nodes v_i and v_j of the path is defined as the difference of the cost function for the corresponding states $\omega(i)$ and $\omega(j)$: $M = C(j) - C(i)$. In such a way, the number of approximating line segments for the closed curve P for starting point p_i is defined as the difference of cost function for conjugate states $\omega(j)$ and $\omega(j)$.

To solve the minimum-rate problem, we have to test the shortest paths to all goal states $v_k \in G_\alpha$ to find such conjugate states $\omega(i)$ and $\omega(i + N - 1)$ for nodes v_i, $v_{i+N-1} \in H(v_k)$ that the difference of cost function values is minimal:

$$M = \min_{v_i, v_{i+N-1} \in H(v_k)} \{C(i + N - 1) - C(i)\}.$$

The algorithm is illustrated by Fig. 10 for test shape #1. The found pair of conjugate states $\omega(5)$–$\omega(28)$ for v_5, $v_{28} \in H(v_k)$ gives the minimal difference of cost functions values: $M = C(28) - C(5) = 5$, the optimal starting point is the vertex p_5.

The complexity of the search for the conjugate states is defined by the number of goal states to be checked, which is $O(N)$, and the number of nodes in the shortest path, $O(M)$. In total, this yields to $O(MN)$, which is much smaller than the complexity of the core algorithm $O(N^2)$.

During the search, we have to backtrack shortest paths $H(v_k)$ for all goal nodes $v_k \in G_\alpha$ and $k > N$. If the fold-factor α is large enough, some node v_k^* can belong to the optimal solution $H^{(\mathrm{opt})}$. Following back from the goal node

v_k^* to the first node v_1 of the graph G_α we could restore the optimal solution $H^{(\mathrm{opt})}$ as a sub-path of the path $H(v_k^*)$. However, the first nodes of the restored path $H(v_k^*)$ do not necessarily belong to the globally optimal path $H^{(\mathrm{opt})}$ because its location can be distorted by non-optimal selection of the starting point, i.e. $v_1 \notin H^{(\mathrm{opt})}$.

The influence of the starting point selection on the subsequent approximating nodes is diminishing when we go further from the node v_1. In other words, when we continue the search in the extended graph G_α, we increase the probability for finding such a goal state $v_k^* \in H^{(\mathrm{opt})}$ that beginning from a node of $H(v_k^*)$ the location of other nodes is not affected anymore by non-optimality of the starting point, and the pair of conjugate states for nodes $v_i, v_{i+N-1} \in H(v_k^*)$ gives global minimum of the number of segments.

4. Results and discussion

Experiments are provided for the set of test shapes shown in Fig. 11. The test set includes images of different type (digitized contour and vector maps), smoothness and size. The shape #1 is used mainly to illustrate the algorithm. The shapes #2 and #3 are examples of a noisy and a smooth contour, respectively. The shape #4 is a large vector contour of France of size $N = 6663$.

The optimality of the solutions for minimum-distortion problem is measured by the fidelity (F), and for minimum-rate problem by the corresponding efficiency parameters [39]. For the shapes #2 and #3, the average fidelity and efficiency are calculated with respect to the optimal solution obtained by trying all possible starting points. For the shape #4, the optimal result have been estimated by testing 200 random starting points.

4.1. Minimum-distortion problem

We compare the following algorithms for the minimum-distortion approximation:

(1) FS-1: one run of the full search DP algorithm [8] with a random starting point;
(2) FS-2: two runs of full search DP algorithm [8] with the starting point selection as in [36];
(3) RS-Ω_2: proposed algorithm with bounded search in the extended state space.

In FS-1, the starting point is chosen randomly and the results presented here are averages over all tested starting points. In FS-2, the starting point for the second run is chosen heuristically as proposed in Ref. [36].

Experiments in Figs. 12–14 and Tables 1–3 show that the heuristic methods (FS-1 and FS-2) find the optimal result only occasionally, and the average fidelity is usually less than 100% for small M.

The results of the RS-Ω_2 are significantly better and the average fidelity of the solution is usually 100%. Thus, the algorithm finds the optimal solution with rare exceptions (see Fig. 12). The superiority of the proposed algorithm RS-Ω_2 over the heuristic ones is most noticeable for small values of M (see Figs. 12–14). As for greater number of segments ($M > 20$), when the heuristic algorithm FS-2 is more efficient, the proposed algorithm is much faster than the FS-2 and yet it provides 100% fidelity.

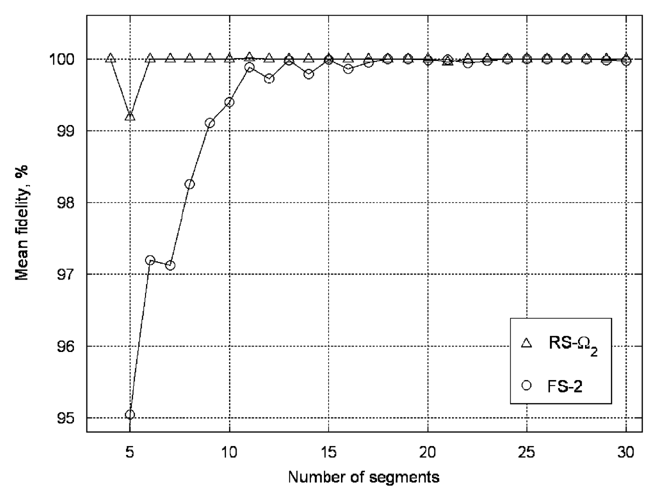

Fig. 12. The average fidelity of the minimum-distortion approximation for the shape #2 with the algorithms FS-2 and RS-Ω_2.

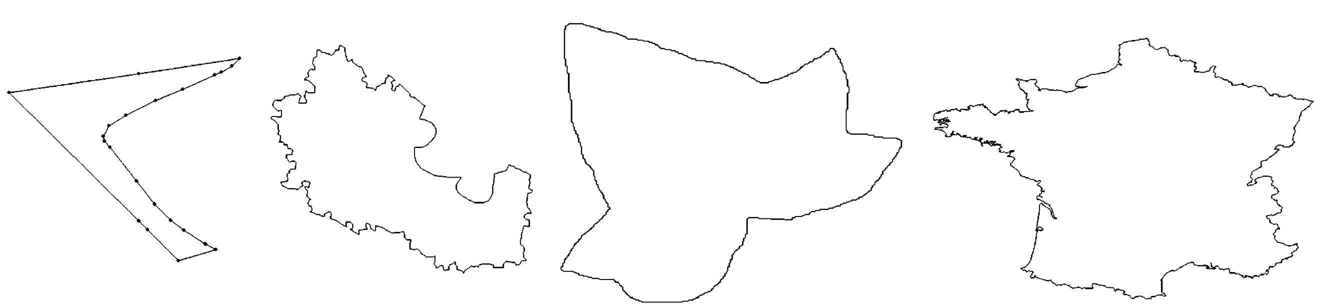

Fig. 11. The set of test shapes: #1: simulated contour; #2: simulated noisy contour; #3: digitized contour Leaf; #4: contour of France.

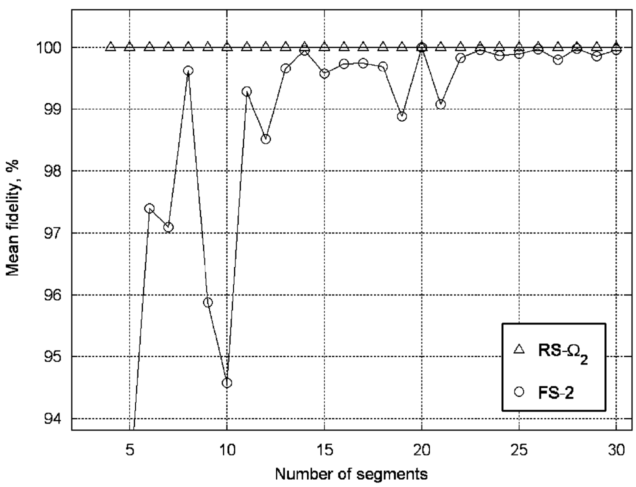

Fig. 13. Average fidelity of minimum-distortion approximation for the shape #3 with the algorithms FS-2 and RS-Ω_2.

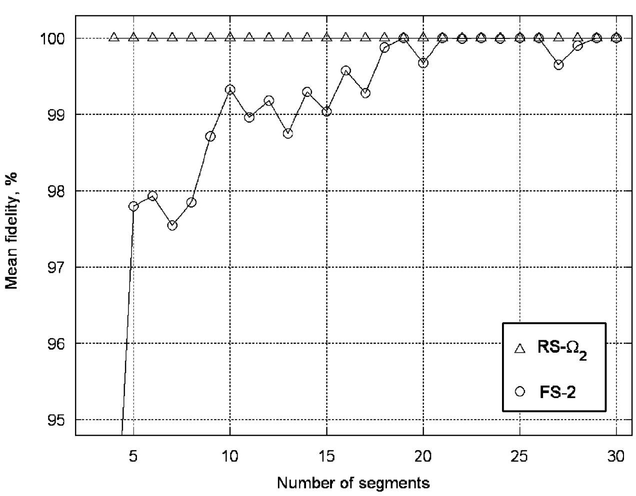

Fig. 14. The average fidelity of minimum-distortion approximation for the shape #4 with the algorithms FS-2 and RS-Ω_2.

Table 1
The shape #2: average fidelity (F) and processing time (T) for the algorithms FS-1, FS-2, and RS-Ω_2

M	FS-1		FS-2		RS-Ω_2	
	F (%)	T (s)	F (%)	T (s)	F (%)	T (s)
5	82.0	0.012	95.1	0.023	99.2	0.035
10	91.5	0.023	99.4	0.047	100	0.020
15	88.3	0.035	99.99	0.070	100	0.015
20	97.1	0.045	99.98	0.089	100	0.011
30	97.2	0.058	99.97	0.116	100	0.010

So, performing search in the extended state space for twofold curve we can analyse much more sub-solutions to find the best one. On the other hand, restricting search in the space by the bounding corridor we reduce the processing time.

Table 2
The shape #3: average fidelity (F) and processing time (T) for the algorithms FS-1, FS-2, and RS-Ω_2

M	FS-1		FS-2		RS-Ω_2	
	F (%)	T (s)	F (%)	T (s)	F (%)	T (s)
5	61.2	0.07	97.9	0.14	100	0.21
10	72.7	0.14	97.0	0.28	100	0.12
15	86.5	0.22	99.6	0.44	100	0.08
20	90.4	0.29	100.0	0.58	100	0.07
30	95.5	0.44	99.95	0.88	100	0.07

Table 3
The shape #4: average fidelity (F) and processing time (T) for the algorithms FS-1, FS-2, and RS-Ω_2

M	FS-1		FS-2		RS-Ω_2	
	F (%)	T_2 (s)	F (%)	T_2 (s)	F (%)	T (s)
6	65.5	3.5	98.1	6.9	100	7.9
10	91.8	5.4	99.4	10.8	100	5.3
15	88.9	8.8	99.9	17.6	100	3.7
20	93.6	12.8	99.7	25.6	100	2.9
50	98.0	34.3	99.99	68.5	100	2.8

The question whether we can obtain the optimal solution for any curve and for any value of M is still open. Even though this is rare and usually happens only with very small values of M, the search can get stuck in a local minimum. Even when extending the search to space Ω_K for K-fold curve ($K=3, 4, \ldots$) we cannot theoretically guarantee 100% fidelity of the solution. From practical point of view, however, the proposed algorithm gives solution very close to the optimal one for 500–5000-vertex curves in 0.1–10 s using our current computer with 3 GHz processor.

4.2. Minimum-rate problem

The second series of experiments was performed using the proposed algorithm for the minimum-rate problem. We compare the following algorithms:

(1) FS-1: one run of the full search DP in the state space Ω_1 for a random starting point in P;
(2) FS-2: two runs of the full search DP in the state space Ω_1 with the starting point as in [36];
(3) FS-Ω_α: proposed algorithm with the DP search in the extended state space Ω_α (see Fig. 10).

For the heuristic method FS-2 with two iterations of the DP algorithm, the processing time is always $2T_1$, where T_1 is the processing time of one run. For the tested shapes, the minimum number of segments was obtained in most cases.

In the case of the proposed algorithm, FS-Ω_α, the trade-off between the processing time and efficiency can be controlled

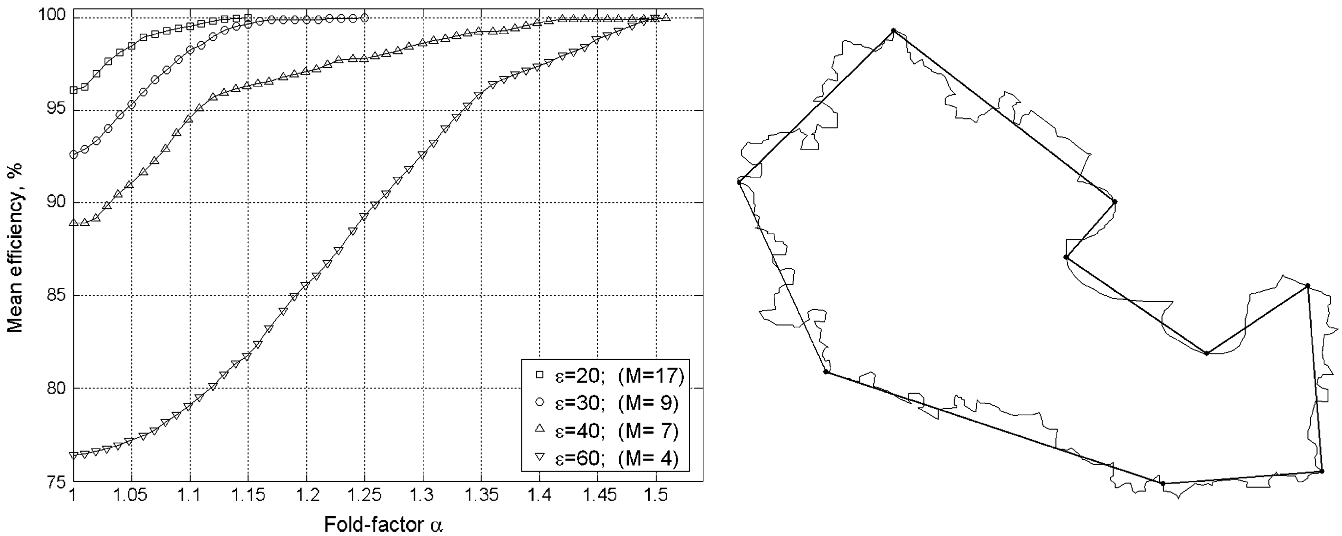

Fig. 15. Average efficiency of the proposed algorithm FS-Ω_α for the minimum-rate problem for the shape #2 (left), and the sample result for $\varepsilon = 30$ (right).

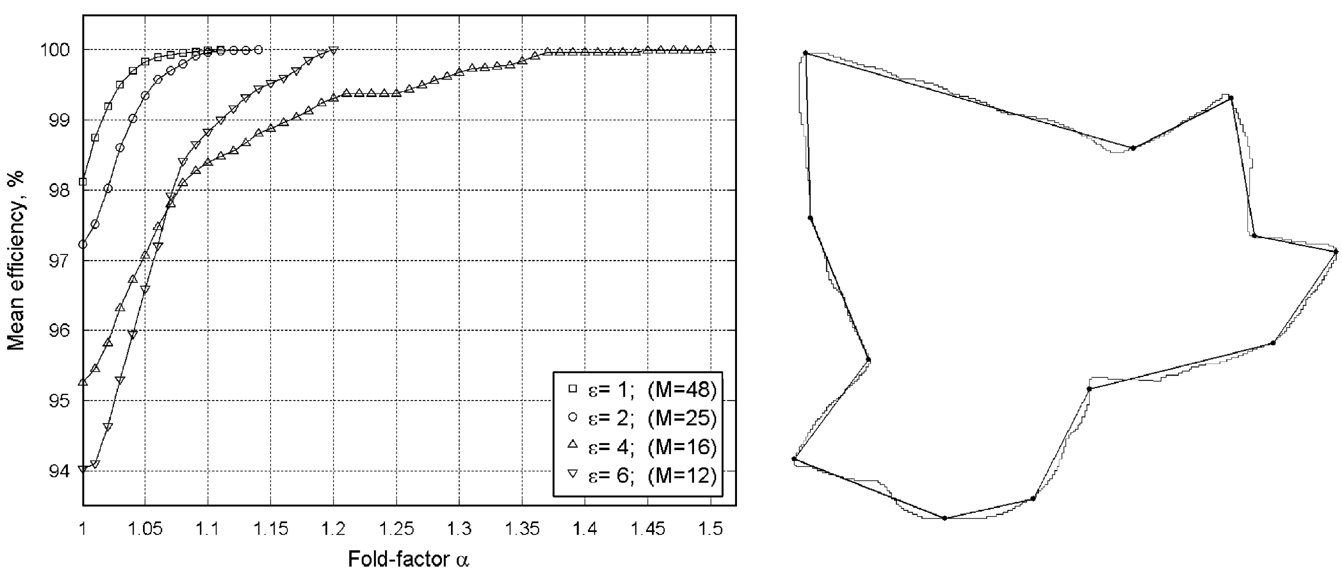

Fig. 16. Average efficiency of the proposed algorithm FS-Ω_α for the minimum-rate problem for the shape #3 (left), and the sample result for $\varepsilon = 6$ (right).

by changing the parameter α within the range from 1.0 (no optimization) up to 2.0. In practice, for the tested shapes the global minimum was achieved for $\alpha \leqslant 1.5$ including the worst case when a large error tolerance value ε was used (see Figs. 15–17 and Tables 4–6). For smaller values of ε, and large number of segments, respectively, the fold-factor α can be even less than 1.5 (see Figs. 15–17).

5. Conclusions

We have introduced a new approach for polygonal approximation of closed curves based on the corresponding optimal dynamic programming algorithm for open curves. The information from the previous run is exploited by performing dynamic programming search in an extended state space constructed for a twofold curve.

For minimum-distortion approximation, dynamic programming search in extended state space finds the optimal approximation with a high probability. On the other hand, reducing the search by bounding corridor along a reference solution we can keep the processing time small.

Then proposed approach, DP search in the extended state space, was also propagated on the case of minimum-rate problem. The processing time of the new algorithm for closed curve is about 1.5 of the processing time for the open curve. The trade-off between the processing time and efficiency can be controlled by changing the number of processed vertices in the second run along the curve.

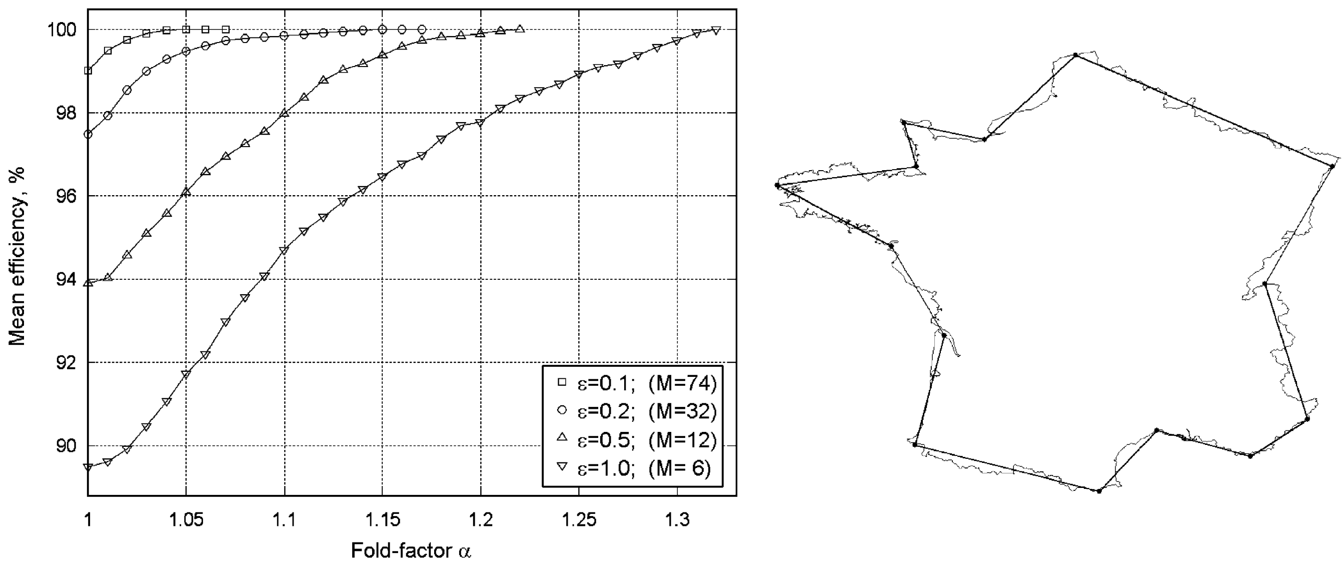

Fig. 17. Average efficiency of the proposed algorithm FS-Ω_α for the minimum-rate problem for the shape #4 (left), and the sample result for $\varepsilon = 0.4$ (right).

Table 4
The shape #2: average efficiency (*Eff.*) and processing time (T) for the algorithms FS-1, FS-2, and FS-Ω_α($\alpha = 1.5$)

ε	$M_{\min}$	FS-1		FS-2		FS-Ω_α	
		Eff. (%)	T_1 (s)	*Eff.* (%)	T_2	*Eff.* (%)	T (s)
20	17	96.1	0.009	100	0.017	100	0.014
30	9	92.6	0.017	100	0.036	100	0.027
40	7	88.9	0.023	100	0.048	100	0.038
60	4	76.5	0.040	97.1	0.080	100	0.065

Table 5
The shape #3: average efficiency (*Eff.*) and processing time (T) for the algorithms FS-1, FS-2, and FS-Ω_α($\alpha = 1.5$)

ε	$M_{\min}$	FS-1		FS-2		FS-Ω_α	
		Eff. (%)	T_1 (s)	*Eff.* (%)	T_2 (s)	*Eff.* (%)	T (s)
1.0	48	98.1	0.014	100	0.027	100	0.021
2.0	25	97.2	0.036	100	0.071	100	0.058
4.0	16	95.6	0.060	100	0.122	100	0.095
6.0	12	94.0	0.078	100	0.160	100	0.125

Table 6
The shape #4: average efficiency (*Eff.*) and processing time (T) for the algorithms FS-1, FS-2, and FS-Ω_α($\alpha = 1.5$)

ε	$M_{\min}$	FS-1		FS-2		FS-Ω_α	
		Eff. (%)	T_1 (s)	*Eff.* (%)	T_2 (s)	*Eff.* (%)	T (s)
0.1	74	98.9	1.5	100	3.0	100	2.3
0.2	32	97.2	7.5	100	15.0	100	13.2
0.5	12	94.3	34.7	100	69.7	100	47.5
1.0	6	89.2	97.9	100	188.3	100	159.1

In general, the proposed approach cannot guarantee 100% optimality of the obtained approximation solution, but with the analysis of the extended state space, we can get optimal or very close to the optimal solution in relatively short time.

References

[1] H. Imai, M. Iri, Polygonal approximation of a curve (formulations and algorithms), in: G.T. Toussaint (Ed.), Computational Morphology, North-Holland, Amsterdam, 1988, pp. 71–86.

[2] W.S. Chan, F. Chin, On approximation of polygonal curves with minimum number of line segments or minimum error, Int. J. Comput. Geometry Appl. 6 (1996) 59–77.

[3] D.Z. Chen, O. Daescu, Space efficient algorithm for approximating polygonal curves in two dimensional space, Proc. Fourth International Conference on Computing and Combinatorics, 1998, pp. 55–64.

[4] J.G. Dunham, Optimum uniform piecewise linear approximation of planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 67–75.

[5] H. Imai, M. Iri, Computational-geometric methods for polygonal approximations of a curve, Comput. Vision Image Process. 36 (1986) 31–41.

[6] H. Imai, M. Iri, An optimal algorithm for approximating a piecewise linear function, J. Inform. Process. 9 (1986) 159–162.

[7] G. Papakonstantinou, Optimal polygonal approximation of digital curves, Signal Process. 8 (1985) 131–135.

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

[9] A. Pikaz, I. Dinstein, Optimal polygonal approximation of digital curves, Pattern Recognition 28 (3) (1995) 373–379.

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

[11] U. Ramer, An iterative procedure for polygonal approximation of plane curves, Comput. Graphics Image Process. 1 (1972) 244–256.

[12] D.H. Douglas, T.K. Peucker, Algorithm for the reduction of the number of points required to represent a line or its caricature, The Canadian Cartographer 10 (2) (1973) 112–122.

[13] H.Z. Shu, L.M. Luo, J.D. Zhou, X.D. Bao, Moment-based methods for polygonal approximation of digitized curves, Pattern Recognition 32 (2002) 421–434.
[14] M. Visvalingam, J. Whyatt, Line generalization by repeated elimination of points, Cartographic J. 30 (1993) 46–51.
[15] L. Boxer, C.-S. Chang, R. Miller, A. Rau-Chaplin, Polygonal approximation by boundary reduction, Pattern Recognition Lett. 14 (1993) 111–119.
[16] A. Pikaz, I. Dinstein, An algorithm for polygonal approximation based on iterative point elimination, Pattern Recognition Lett. 16 (1995) 557–563.
[17] K.M. Ku, P.K. Chisu, Polygonal approximation of digital curve by graduate iterative merging, Electron. Lett. 31 (1995) 444–446.
[18] L.J. Latecki, R. Lakämper, Convexity rule for shape decomposition based on discrete contour evolution, Comput. Vision Image Understanding 73 (1999) 441–454.
[19] A. Kolesnikov, P. Fränti: Multiresolution polygonal approximation of digital curves, Proc. Int. Conf. Pattern Recognition-ICPR'04, Cambridge, UK, August, vol. 2, 2004, pp. 855–858.
[20] A. Kolesnikov, P. Fränti, Data reduction of large vector graphics, Pattern Recognition 38 (2005) 381–394.
[21] C.-H. Teh, R.T. Chin, On the detection of dominant points on digital curves, IEEE Trans. Pattern Anal. Mach. Intell. 11 (8) (1989) 859–872.
[22] P. Zhu, P.M. Chirlian, On critical point detection of digital shapes, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 737–748.
[23] J.M. Iñesta, M. Buendí, M.A. Sarti, Reliable polygonal approximations of imaged real objects through dominant point detection, Pattern Recognition 31 (1998) 685–697.
[24] A. Garrido, N.P. de la Blanca, M. Garcia-Silvente, Boundary simplification using a multiscale dominant-point detection algorithm, Pattern Recognition 31 (1998) 791–804.
[25] J. Sklansky, V. Gonzalez, Fast polygonal approximation of digitized curves, Pattern Recognition 12 (1980) 327–331.
[26] Y. Kurozumi, W.A. Davis, Polygonal approximation by the minimax method, Computer Vision, Graphics Image Process. 19 (1982) 248–264.
[27] B.K. Ray, K.S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, Pattern Recognition Lett. 15 (1994) 161–167.
[28] P.-Y. Yin, A new method for polygonal approximation using genetic algorithms, Pattern Recognition Lett. 19 (1998) 1017–1026.
[29] S.-C. Huang, Y.-N. Sun, Polygonal approximation using genetic algorithms, Pattern Recognition 32 (1999) 1017–1026.
[30] S.-Y. Ho, Y.-C. Chen, An efficient evolutionary algorithm for accurate polygonal approximation, Pattern Recognition 34 (2001) 2305–2317.
[31] H. Zhang, J. Guo, Optimal polygonal approximation of digital planar curves using meta-heuristics, Pattern Recognition 34 (2001) 1429–1436.
[32] P.-Y. Yin, A tabu search approach to the approximation of digital curves, Int. J. of Pattern Recognition Artif. Intell. 14 (2000) 243–255.
[33] U. Vallone, Bidimensional shapes polygonalization by ACO, Proceedings of the third International Workshop on Ants Algorithms, Brussels, Belgium, 2002, pp. 296–297.
[34] P.-Y. Yin, Ant colony search algorithms for optimal approximation of plane curve, Pattern Recognition 36 (2003) 1783–1797.
[35] P.-Y. Yin, A discrete particle swarm algorithm for optimal polygonal approximation of digital curves, J. Visual Commun. and Image Representation 15 (2004) 241–260.
[36] J.-H. Horng, J.T. Li, An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, Pattern Recognition Lett. 23 (2002) 171–182.
[37] J.-H. Horng, Improving fitting quality of polygonal approximation by using the dynamic programming technique, Pattern Recognition Lett. 23 (2002) 1657–1673.
[38] A. Kolesnikov, P. Fränti, Reduced search dynamic programming for approximation of polygonal curves, Pattern Recognition Lett. 24 (2003) 2243–2254.
[39] P.L. Rosin, Techniques for assessing polygonal approximations of curves, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1997) 659–666.

About the Author—ALEXANDER KOLESNIKOV received the M.Sc. degree in Physics in 1976 from the Novosibirsk State University, USSR, and the Ph.D. degree in Computer Science in 2003 from the University of Joensuu, Finland. From 1976 to 2003 he was a Senior Research Fellow with the Institute of Automation and Electrometry, Russian Academy of Sciences, Novosibirsk, Russia. In 2003, he joined the Department of Computer Science at the University of Joensuu, Joensuu, Finland. His main research areas are in signal and image processing, vector map processing and compression.

About the Author—PASI FRÄNTI received his M.Sc. and Ph.D. degrees in Computer Science in 1991 and 1994, respectively, from the University of Turku, Finland. From 1996 to 1999 he was a postdoctoral researcher of the Academy of Finland. Since 2000, he has been a professor in the University of Joensuu, Finland. His primary research interests are in image compression, clustering, and speech technology.