Polygonal approximation of closed contours

Alexander Kolesnikov and Pasi Fränti

Department of Computer Science, University of Joensuu
80101 Joensuu, Finland
koles@cs.joensuu.fi, franti@cs.joensuu.fi

Abstract. Optimal 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 algorithm for open curve. We propose to approximate a cyclically extended contour of double size, and to select the best possible starting point by analyzing the state space. This takes only twice of the time required by the algorithm for open curve.

1 Introduction

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

a) *min-ε 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) *min-# problem*: Given an *N*-vertex polygonal curve *P*, approximate it by another polygonal curve *Q* with minimum number of segments so that the error does not exceed a given maximum tolerance Δ.

The problem can be solved by dynamic programming (DP) approach [3,5-7,9,14,16], by A*-search [11], or by algorithms developed for the shortest path problem in digraph [1,2,4,10,15]. The time complexity of these algorithms varies from $O(N^2)$ to $O(N^3)$.

In the case of closed contours, we have to find optimal allocation of all approximation vertices including the starting point. A straightforward solution is to try all vertices as starting points, and to choose the one with minimal error [9]. The complexity of the straightforward algorithm for *N*-vertex contour is *N* times that of the algorithm for open curve: between $O(N^3)$ and $O(N^4)$. In [1] it was shown that the *min-# problem* for closed curve can be solved in $S(N)$ time, where $S(N)$ is the time for solving the all-pairs shortest path problem for graph of *N* vertices.

There also exist a number of heuristic approaches for selecting the starting point [5,10,11,14-16]. Sato [14] chooses the farthest point from the center of gravity as a starting point. The worst case complexity of the algorithm is $O(N^3)$. In the case of the error metrics L_1, Ray and Ray [11] proposed an algorithm to extend the cycle beyond the current starting point until to the first approximation vertex to revise the choice of the starting point. Pikaz and Dinstein [10] considered *min-# problem* with city-block error metrics, and proposed an algorithm based on the shortest path problem for open and closed curves.

Zhu and Seneviratne [16] considered the approximation problem with L_∞ metrics, and proposed an $O(N^2)$ time algorithm for open curves. An iterative procedure for optimizing the choice of the starting point was also suggested for the case of closed curve. Horng and Li [5] proposed a two-step method to select the starting point: at the first step, optimal approximation with any starting point is performed. At the second step, the approximation is performed again using the most distant vertex as the new starting point. Schroeder and Laurent [15] perform preliminary approximation until the first approximation vertex is reached, and use this vertex as a new starting point for the next iteration.

To sum up, the proposed heuristic approaches for closed curves are sub-optimal whereas the optimal choice of the starting point is time consuming. Thus, the problem still remains unsolved. Several of the heuristic approaches perform preliminary approximation by using one of the approximation vertices as a new starting point for the final approximation [5,11,14,15,16]. After the first run, however, the search starts from scratch and loses the information of the previous run.

We propose to approximate with dynamic programming algorithm the double-size curve obtained from the closed contour in wrapped manner. Then the state space of the solution is analyzed for selecting the best possible starting point. The proposed approach is illustrated in Fig 1. Approximation with DP algorithm for open curve with the vertex 1 as the starting point is shown left. The corresponding solution for double-size curve is shown right. Optimal starting point and the corresponding approximation of the original contour can now be found by analyzing the approximation of the double-size curve. In the example, we obtain the solution as the segments 2-3-4-5.

The rest of the paper is organized as follows. In Section 2, we give problem formulation, recall the dynamic programming approach of [9] for approximation of open curves, and then present the new algorithm for the optimization of the starting point. We extend the approach also to the iterative reduced search recently presented in [6,7]. We then briefly consider the approach in the case of *min-# problem*. Experimental results and discussion are given in Section 3, and conclusions are drawn in Section 4.

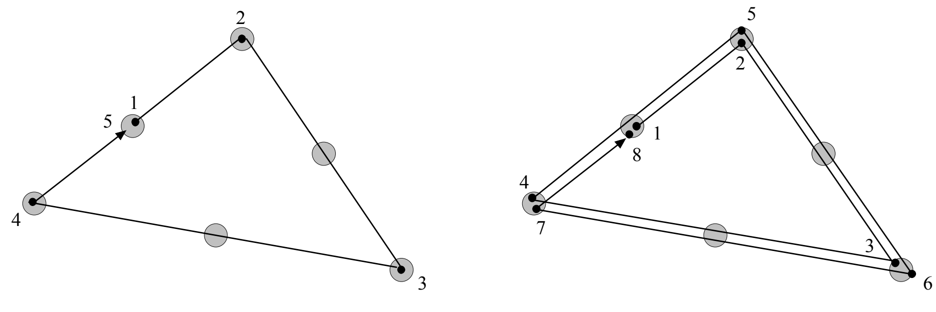

Figure 1: Illustration of the approach for open curve problem (left), and for the closed curve problem (right). The numbers indicate the approximation points.

2. Algorithm for closed contours

Let us define an closed *N-vertex polygonal curve P* in 2-dimensional space as the ordered set of vertices $P = \{ p_1, p_2, ..., p_N;\ p_N = p_1\} = \{(x_1, y_1), (x_2, y_2), \ldots,(x_N, y_N);$ and $(x_N, y_N) = (x_1, y_1) \}$. The problem is stated as follows: approximate the closed polygonal curve (contour) P by another closed polygonal curve Q with a given number of linear segments M so that total approximation error $E(P, M)$ is minimized. The output polygonal curve Q consists of $(M+1)$ vertices: $Q = \{q_1, \ldots, q_{M+1};\ _{M+1}=q_1\}$, where the set of vertices $\{q_1, ..., q_{M+1}\}$ a subset of vertices of P.

The L_2-optimal approximation of contour P is the set of vertices $\{q_1, ..., q_{M+1}\}$ of Q that minimizes the cost function $E(P, M)$:

$$E(P,M) = \min_{\{q_m\}} \sum_{m=1}^{M} e^2(q_m, q_{m+1})$$

Error of the approximation of curve segment $\{p_i, ..., p_j\}$ with the corresponding linear segment (q_m, q_{m+1}) is defined here as the sum of squared Euclidean distances from each vertex of $\{p_i, ..., p_j\}$ to the linear segment (q_m, q_{m+1}); here $q_m=p_i$ and $q_{m+1}=p_j$. To solve the optimization task we first recall the full search optimal dynamic programming algorithm of Perez and Vidal [9].

2.1 Dynamic programming approach

Let us define a discrete 2-dimensional *state space* $\Omega=\{(n, m):\ n = 1, \ldots, N;\ m= 0, \ldots, M\}$. Every point (n, m) in the state space Ω represents the sub-problem of approximating of an n-vertex polygonal curve $\{p_1, p_2, \ldots, p_n\}$ by m line segments. The complete problem is represented by the goal state (N, M).

An output polygonal curve Q can be represented as a *path* $H(m)$ in the state space Ω from the start state (1,0) to the goal state (N, M). In the state space, we can also define a function $D(n, m)$ as the cost function value of the optimal approximation for the n-vertex curve by m linear segments.

For solving the *min-ε problem* under question we have to find the optimal path from the start state (1,0) to the goal state (N, M). The optimization problem can be solved by the dynamic programming algorithm [9] with the following recursive expressions for states $(n, m)\in\Omega$:

$$D(n,m) = \min_{L(m-1)\le j<n} \left\{ D(j,m-1) + e^2(p_j, p_n) \right\};$$
$$A(n,m) = \arg\min_{L(m-1)\le j<n} \left\{ D(j,m-1,j) + e^2(p_j, p_n) \right\}.$$

Here $A(n, m)$ is the *parent state* that provides the minimum value for the cost function $D(n, m)$. The time complexity of the algorithm is $O(MN^2)$.

2.2 Solution for closed contours

We next extend the full-search dynamic programming approach on the case of closed contours. The main idea is to perform approximation of the wrapped input closed contour cyclically and make analysis of the state space to select a starting point that provides minimal approximation error. The proposed algorithm consists of four steps:

Step 1: Create a double-size closed curve P_2 of size ($2N$-1) from vertices of the N-vertex input closed contour P_1: $P_2 = \{p_1, p_2, \ldots, p_{N-1}, p_1, p_2, \ldots, p_N\}$.

Step 2: Construct bounded state space Ω_2 for the contour P_2 (see Fig. 2). The configuration of Ω_2 is explained by the fact that this is a state space for the double-size cyclically wrapped curve P, not those for arbitrary curve of size $2N$-1.

Step 3: Perform search in the bounded state space Ω_2 with dynamic programming algorithm.

Step 4: Make analysis of optimal paths $H(m)$ for all goal states (n_g, m_g) in Ω_2, where $N \le n_g \le 2N-1$ and $M \le m_g \le 2M$.

Let us restore the optimal path $H(m)$ for some goal state (n_g, m_g), and check all pairs of states (($H(m)$, m), ($H(m-M)$, $m-M$)) on the path $H(m)$, where $m=M, \ldots, m_g$. We call two states ($H(m)$, m) and ($H(m-M)$, $m-M$) on the path $H(m)$ *conjugate*, if $H(m)-H(m-M)=N-1$. Sub-path of the path $H(m)$ from a state ($H(m-M)$, $m-M$) to the conjugate one ($H(m)$, m) corresponds to approximation of the closed curve P of size N with M linear segments for starting point $n=H(m-M)$. Approximation error for the sub-path is defined as the difference of the cost function values of the conjugate states: $E(m)=D(H(m), m)-D(H(m-M), m-M)$. To find the optimal starting point n_{opt} we have to find two conjugate states, which provide minimum of the approximation error $E(m)$ for all goal states (n_g, m_g) in Ω_2, where $N \le n_g \le 2N-1$ and $M \le m_g \le 2M$:

$$n_{\text{opt}} = \arg \min_{M \le m \le M_g} \{D(H(m), m) - D(H(m-M), m-M)\} - (N-1).$$

As we know, in DP algorithm for approximation of *open* curve a solution is constructed under the condition $q_{M+1}=p_N$. Propagating DP search in state space Ω_2 beyond the end point of P, we remove the restriction $q_{M+1}=p_N$ and do the approximation vertex q_{M+1} "free". Then we analyze solutions of all relevant sub-tasks with the free vertex q_{M+1} to find the best location for the starting point.

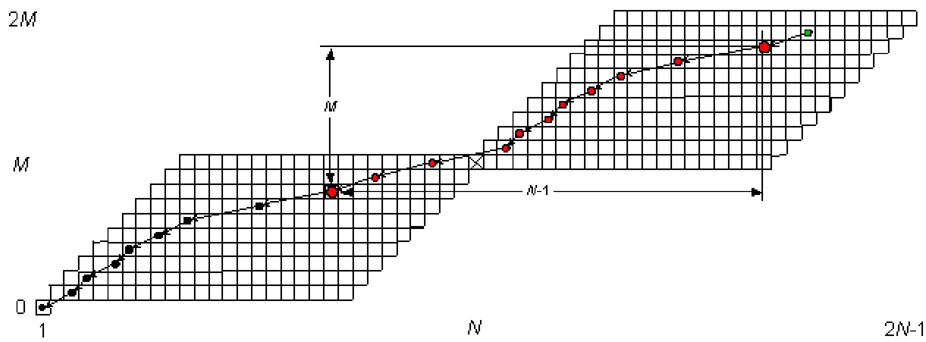

Fig. 2: Illustration of the state space Ω_2 for double-size contour P_2. The optimal path $H(m)$ for a goal state (n_g, m_g,) is marked by dots. Two *conjugate* states in the path are emphasized.

Due to the special configuration of the state space Ω_2 processing time T_2 for curve P_2 is only twice bigger than that of the curve P_1 in the state space Ω_1: $T_2 \approx 2T_1$.

It is noted that the optimality of the solution cannot be guaranteed in general, it is nevertheless expected that the proposed approach can provide better solution than the other heuristic algorithm for the selection of the starting point. In principle, we could continue the search in the state space Ω_k constructed for the k-size curve P_k, where k=3,4,…, but experiments show that the further search would provides negligible improvement of the solution (if any) in comparison to the search in the space Ω_2.

2.3 Reduced-search for closed contours

The complexity of the full-search approximation algorithm is $O(MN^2)$. To solve the approximation tasks for big N in practice, we introduced an iterative reduced-search algorithm for the case of open curves [6,7]. With this algorithm, we can get optimal (or near optimal) solution in $O(N)..O(N^2)$ time.

Instead of the full search in state space, only a small but relevant part of the state space is processed in a bounding corridor along the current solution. We can use the iterative nature of this algorithm and apply heuristic approach as mentioned in the introduction: select a new starting point among the approximation vertices after every iteration. In this way, with every run of the reduced search we improve the location of all non-start points; selecting a new start point after the run, we improve location of the starting point. The method provides very good results but the result is not always optimal. To improve the efficiency of the heuristic approach we apply the algorithm of Section 2.2 and analyse the state space Ω_2 for double-size curve P_2.

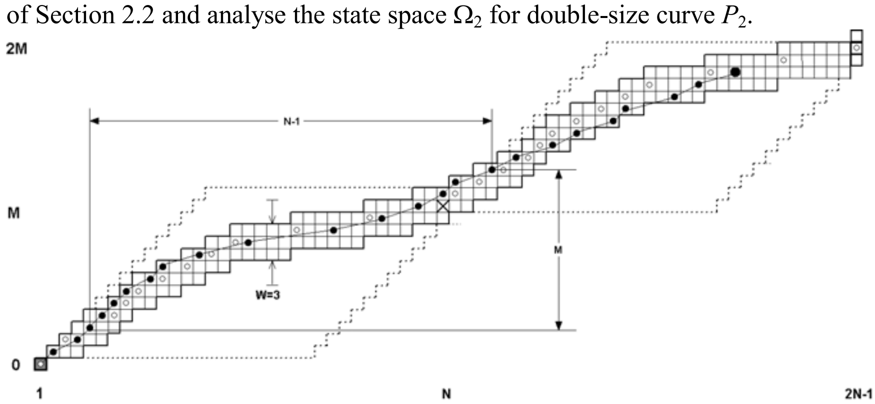

Fig. 3: Illustration of the bounding corridor in the state space Ω_2 with the corridor width of W=3. The reference path is marked by circles, and the state (N, M) by symbol '×'. The optimal path $H(m)$ from (0,1) to goal state (n_g, m_g) is marked by dots and lines. Two sample *conjugate* states in the path $H(m)$ are emphasized.

Approximation for the closed contours can be obtained with the following algorithm:

a) $M \leq 20$: perform approximation of the double-size curve P_2 with a single iteration of the reduced search, and analysis of the state space Ω_2;

b) M>20: perform approximation of the curve P with iterative reduced search in Ω_1, and by selecting a new starting point after every iteration until no further improvement is registered. Then perform approximation of the double-size curve P_2 with analysis of the state space Ω_2. If processing time is limited, we can perform only two iterations of the reduced search.

2.4 Solution for *min-# problem*

To solve the *min-# problem* for closed curves, we have to find approximation of the closed contour P by another closed contour Q with minimal number of line segments with an approximation error within given error tolerance level: $d(P)\leq\Delta$. The approximation error $d(P)$ with measure L_∞ is given as the maximum Euclidean distance from the vertices of P to the approximation linear segments.

To solve the *min-# problem* a digraph is constructed on the vertices of the input contour P. In the digraph, a pair of vertices p_i and p_j are connected with an edge if the approximation error of the curve segment between the vertices is less than a given error tolerance: $d(p_i,p_j)<\Delta$. The optimal solution is then a given by solving the shortest path in the digraph. This can be solved by using DP algorithm for the shortest path problem in the digraph [3,8,15].

To find the optimal approximation for *closed* contour we shall follow the approach introduced in this paper: perform approximation of the wrapped double-size closed contour P_2 and then analyze the state space. In the case of *min-# problem*, the analysis of the space is reduced to the analysis of the solutions for the relevant sub-problems: we have to found that part of the approximation polygonal curve Q_2, which have the same start and end points.

Let us define $M_1=\lfloor M_2/2\rfloor$, where M_2 is the number of segments of Q_2 found with the approximation algorithm. The number of approximation segments M for the input curve P is supposed to be equal to M_1 or M_1-1. To find the optimal starting point n_{opt} we have to check the first M_1 vertices q_k of Q_2, where k=0, .., M. If the approximation vertices q_{k+M} or q_{k+M-1} correspond to the same vertex p_n of the input contour P as the vertex q_k, the vertex p_n is the optimal starting point. If the pair is not found, we have to use the current starting point.

3. Results and discussion

Results of experiments for 3500-vertex test shape "France" are shown in Fig.4 and 5. The optimality of the obtained solutions for *min*-ε *problem* is defined by *fidelity* (*F*) and for *min-# problem* by *efficiency* (*Eff*) parameters [12]. The experiments for the shape "France" and other test shapes have shown that for the *min*-ε *problem* with relatively small values of M (e.g. $M\leq 20$) the optimal starting point can usually be found by the reduced search algorithm and the analysis of the state space Ω_2. For big values of M, the iterative reduced search in the state space Ω_1 can be used along with the reduced search, and the analysis in the state space Ω_2.

For comparison, *min*-ε approximation of the test shape "France" by the method of Horng [5] with two iterations of the full search DP algorithm [9] takes about 100 s with fidelity F=89..100%. The proposed algorithm, on the other hand, obtains the optimal result in 10 s. The *min*-# approximation of the test shape "France" was performed by a modified version of Schroeder and Laurent algorithm [13]: L_2 and L_∞ are used jointly [8] instead of using only L_∞.

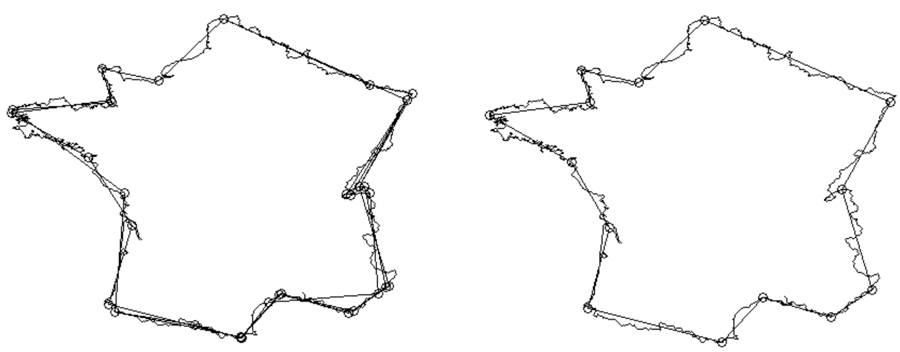

Fig. 4: Results of the *min*-ε approximation of the shape "France" with 14 line segments using random starting point (100 repeats): **a**) by two iterations of the reduced search algorithm (W=5) for the open curves [6,7]. The quality of the result varies from F=88..100%, and the processing time is T=8.0 s, on average. **b**) by the proposed algorithm for the closed contours (W=5) with the results of F=100%, and T=10.4 s.

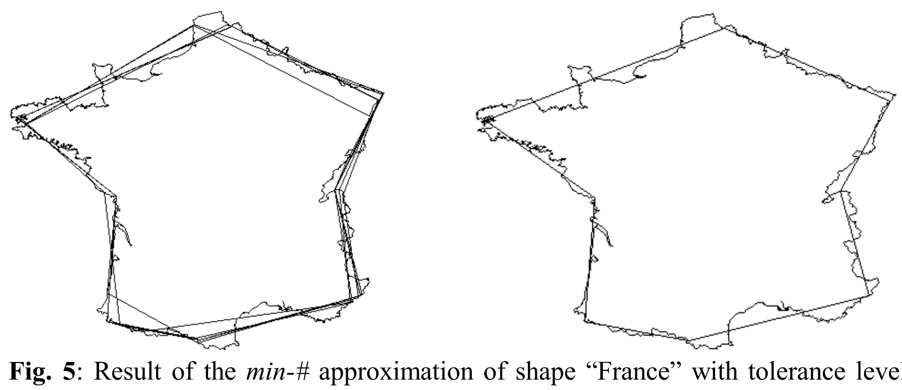

Fig. 5: Result of the *min*-# approximation of shape "France" with tolerance level Δ=0.75 using 50 random starting points: **a**) by two iterations of the algorithm for the open curves [15]: Eff=99.5%, T=88 s; **b**) by the proposed algorithm for the closed contours: M=8, Eff=100%, T=91 s.

4. Conclusions

We have introduced a new approach for *min*-ε and *min*-# approximation of closed contours based on dynamic programming method for open curves. It performs approximation of the cyclically wrapped doubled-size contour and then makes analysis of the state space to select the best starting point. The processing time is double of that of the approximation of the corresponding open curve. The time complexity of the algorithms is defined by the complexity of approximation algorithms for open curves in use. For solving the *min*-ε *problem* the suggested method can be used with iterative reduced search algorithm introduced earlier with the time complexity between $O(N)$ and $O(N^2)$. In most cases, it does provide optimal solution although the question of optimality remains an open question.

References

1. Chan, W.S., Chin, F.: On approximation of polygonal curves with minimum number of line segments or minimum error. Int. J. on Comput. Geometry and Applications. 6 (1996) 59-77.
2. Chen, D.Z., Daescu, O.: Space efficient algorithm for polygonal curves in two dimensional space, Proc. 4th Int. Conf. Computing and Combinatorics. (1998) 55-64.
3. Dunham, J.G.: Optimum uniform piecewise linear approximation of planar curves. IEEE Trans on Pattern Analysis and Machine Intelligence. 8(1) (1986) 67-75.
4. Imai, H., Iri.M.: Computational-geometric methods for polygonal approximations of a curve. Computer Vis. Image Processing. 36 (1986) 31-41.
5. Horng, J.-H., Li, J.T.: An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. Pattern Recognition Letters. 23 (2002) 171-182.
6. Kolesnikov, A., Fränti, P.: A fast near-optimal algorithm for approximation of polygonal curves. Proc. Int. Conf. Pattern Recognition-ICPR'02. 4 (2002) 335-338.
7. Kolesnikov, A., Fränti, P.: Reduced search dynamic programming for approximation of polygonal curves. Pattern Recognition Letters. 2003. (in press)
8. Papakonstantinou, G., Tsanakas, P., Manis, G.: Parallel approaches to piecewise linear approximation. Signal Processing. 37 (1994) 415-423.
9. Perez, J.C., Vidal, E.: Optimum polygonal approximation of digitized curves, Pattern Recognition Letters. 15 (1994) 743-750.
10 Pikaz, A., Dinstein, L.: Optimal polygonal approximation of digital curves. Pattern Recognition. 28 (3) (1995) 373-379.
11. Ray, B.K., Ray, K.S.: Determination of optimal polygon from digital curve using L_1 norm. Pattern Recognition. 26 (1993) 505-509.
12. Rosin, P.L.: Techniques for assessing polygonal approximations of curves, IEEE Trans. Pattern Analysis and Machine Intelligence. 14 (1997) 659-666.
13. Salotti, M.: An efficient algorithm for the optimal polygonal approximation of digitized curves, Pattern Recognition Letters. 22 (2001) 215-221.
14. Sato, Y.: Piecewise linear approximation of planar curves by perimeter optimization. Pattern Recognition. 25 (12), (1992) 1535-1543.
15. Schroeder, K., Laurent, P.: Efficient polygon approximations for shape signatures. Proc. Int. Conf. Image Processing-ICIP'99. 2 (1999) 811-814.
16. Zhu, Y, Seneviratne, L.D.: Optimal polygonal approximation of digitized curves. IEE Proc. - Vision, Image and Signal Processing. 144 (1997) 8-14.