Path Simplification under Difference Area Measure

Shervin Daneshpajouh
Department of Computer Engineering
Sharif University of Technology,
Tehran, Iran.
Email: daneshpajouh@ce.sharif.edu

Alireza Zarei
Department of Computer Engineering
Sharif University of Technology,
Tehran, Iran.
Email: zarei@mehr.sharif.edu

Mohammad Ghodsi
Sharif University of Technology
Computer Engineering Department &
School of Computer Science
Institute for Research in Fundamental
Sciences (IPM), Tehran, Iran.
Email: ghodsi@sharif.edu

***Abstract*—In this paper, we consider path simplification problem under difference area (diff-area) measure. Diff-area measure is defined as $|A_A(Q) - A_B(Q)|$, where $A_A(Q)$ is the area under Q and above P and $A_B(Q)$ is the area above Q and under P (see Figure 1). Bose *et al.* [1] presented an approximation algorithm for finding a simplified path with at most k vertices that minimizes the diff-area measure which only works on x-monotone paths. The constraint of being x-monotone is restrictive in some applications like tracking bird migration paths or map boundary simplification. Here, we extend the method of Bose *et al.* [1] and present algorithms with the same time complexities as theirs for general paths.**

I. INTRODUCTION

Path simplification, also known as line simplification or generalization in some literatures, is a fundamental problem in cartography , imaging, computational geometry and geographic information systems (GIS). We are given sequence of input points defining a path $P = \langle p_1, p_2, \ldots, p_n \rangle$ and asked to simplify P by another path $Q = \langle q_1 = p_1, q_2, \ldots, q_k = p_n \rangle$ with $k < n$. The similarity between P and Q is measured using different metrics (error functions).

There are two main versions of this problem. In the *restricted* version, the vertices of Q should be a subsequence of the vertices of P. In the *unrestricted* version, there is no such restriction.

The error of a simplification Q under an error function m is represented by $\text{Error}_m(Q)$. This simplification error is either defined to be $\max_{i=1}^{k-1} \text{Error}_m(q_i q_{i+1})$ or $\sum_{i=1}^{k-1} \text{Error}_m(q_i q_{i+1})$ which are respectively referred by $\mathcal{P}rob^{\text{m}}_{\text{max-simp}}$ and $\mathcal{P}rob^{\text{m}}_{\text{sum-simp}}$. Assuming that $q_i q_{i+1}$ is the simplification of sub-path $P(s,t) = \langle p_s = q_i, p_{s+1}, \ldots, p_t = q_{i+1} \rangle$, $\text{Error}_m(q_i q_{i+1})$ is the associated error of approximating $P(s,t)$ by link $q_i q_{i+1}$ under error function m. The main simplification metrics are retained length, angular change, perpendicular distance, Fréchet distance, and areal displacement [2]. A survey and comparison of these metrics can be found in [3], [4].

There are two optimization goals for this problem: (1) min-k, where for a given error threshold δ, the goal is to find a simplification with the minimum number of vertices and error of at most δ, and (2) min-δ, where for a given number k, the goal is to find a simplification of at most k vertices and with the minimum simplification error.In this paper, we consider the restricted version of the min-k problem.

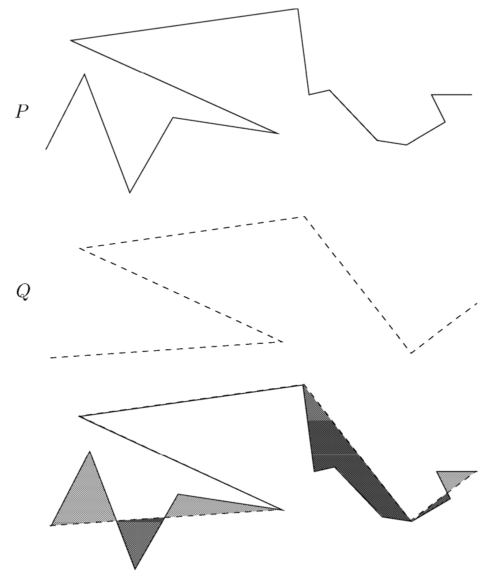

Fig. 1. Path P is simplified by path Q. $A_A(Q)$ is shown by dark gray and $A_B(Q)$ is shown by light gray.

*a) **Related work.**:* There are different results for line simplification under the area measure [1], [5], [6], [7], [8], [9], [10]. It was first studied by McMaster [5], [6]. Aronov *et al.* [10] studied the unrestricted version of min-k simplification under L_1 and L_2 metrics. They presented an approximation algorithm of $O(kn^4\epsilon^{-4}\log^4(n+\epsilon^{-1}))$ time complexity and $1+\epsilon$ error. Bose *et al.* [1] studied three kind of area measures: Sum-area , Max-area and Diff-area. They proposed simplification algorithms for these three measures on x-monotone paths. For $\mathcal{P}rob^{\text{sum-area}}_{\text{sum-simp}}$, they presented a polynomial time algorithm for min-δ simplification. They showed that $\mathcal{P}rob^{\text{max-area}}_{\text{sum-simp}}$ and $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$ are NP-hard and therefore presented approximation algorithms for these measures. Daneshpajouh *et al.* [11] studied the unrestricted version of min-k simplification and presented an optimal $O(n^3)$ algorithm and an $O(n^2)$ approximation algorithm for $\mathcal{P}rob^{\text{sum-area}}_{\text{max-simp}}$.

*b) **Difference Area Measure.**:* For a simplified path Q of a given path P the value of diff-area measure for $\mathcal{P}rob^{\text{diff-area}}_{\text{max-simp}}$ is maximum values of $|A_A(Q) - A_B(Q)|$ of all edges of Q. This measure is defined as sum $|A_A(Q)-A_B(Q)|$ values of all

Proceedings of the 14th International CSI Computer Conference (CSICC'09)
978-1-4244-4262-1/09/$25.00 ©2009 IEEE

edges of Q for $\mathcal{P}rob^{\text{diff-are}}_{\text{sum-simp}}$. This measure has applications in map boundary simplifications where $A_A(Q)$ and $A_B(Q)|$ areas are exchangeable [1].

*c) **Our results.**:* In this paper, we present two algorithms for path simplification under diff-area measure which work on general paths. First, we present a method for finding error of all n^2 possible simplification links in $O(n^2)$ time. Having this result, the $\mathcal{P}rob^{\text{diff-area}}_{\text{max-simp}}$ is solved in $O(n^2)$ time using the general DAG approach of Imai and Iri [12]. Then, we present an $O(n^2k^2/\gamma)$ time algorithm for $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$ which is based on dynamic programming techniques.

II. Simplification Algorithms

In this section, we first present an algorithm for computation of errors of links in $O(n^2)$ time. Then, we use Imai and Iri's approach for solving min-k for $\mathcal{P}rob^{\text{diff-area}}_{\text{max-simp}}$. Finally, we employ dynamic programming for solving min-δ for $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$.

A. Computing the links errors

This process can be done in $O(n^3)$ using a naive algorithm. There are $O(n^2)$ possible links for which the unified error must be computed. Computation of $\text{Error}_{diff-area}(p_ip_j)$ can be done in $O(j-i)$ time by a linear trace on the sub-path $P(i,j)$. Therefore, we can do this computation for all p_ip_j links in $O(n^3)$. Here, we present an $O(n^2)$ algorithm for this computation.

There are different methods for computing the area of a simple polygon [13], [14]. *Polar formula* is one of these methods. We employ this method and show how this method can be used for diff-area measure optimal computation.

For each edge $\overrightarrow{p_ip_j}$, $A(\overrightarrow{p_ip_j})$ is defined as $A = x(p_i)y(p_i) - x(p_j)y(p_j)$ in which $x(p)$ is x coordinate of p and $y(p)$ is its y. It is proved that $A(\overrightarrow{p_ip_j})$ is twice the area of the triangle having p_i, p_j and $(0,0)$ vertices [13] if $p_j \leq p_i$ and otherwise it is negative.

Therefore, the area of a simple polygon $P = \{p_0, p_1, ..., p_n\}$ with $e_i = p_ip_{i+1} : 0 \leq i < n$ and $e_n = p_np_0$ can be computed using following formula:

$$A(P) = |\frac{1}{2}\sum_{i=0}^{n} A(e_i)| \quad (1)$$

An example of this computation is shown in Figure 2. We can use this method for computation of $\text{Error}_{diff-area}(p_ip_j)$. Let $P(i,j) = \{p_i, p_{i+1}, ..., p_j\}$ be a non-simple polygon. It is clear that if this polygon is a simple one, then $A(P(i,j)) = \text{Error}_{diff-area}(p_ip_j)$. Next, we show that this equation is true, in case that p_jp_i intersects some other edges.

Consider path p_ip_j in figure 3.a. In the first step, we connect the origin to p_i in a way that it does not intersect itself and path p_ip_j using a sequence of points $S = \{s_0, s_1, ..., p_k\}$. We do the same for connecting p_j to origin using a sequence of points $T = \{t_0, t_1, ..., t_l\}$. In this way, we build a simple

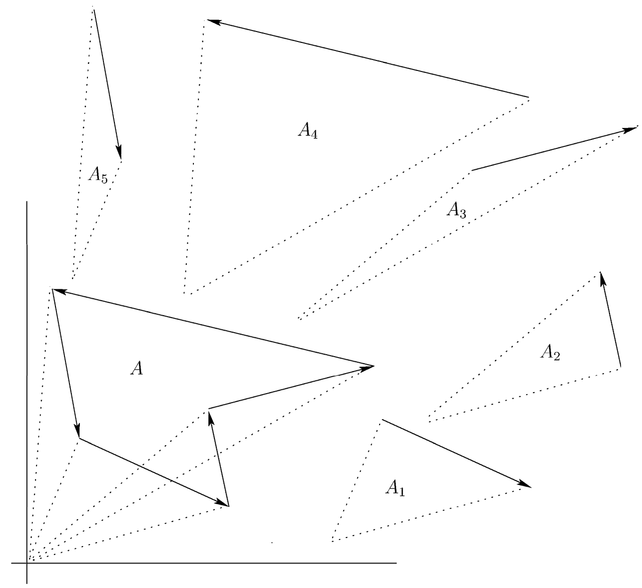

Fig. 2. $A = ||A_1| - |A_2| + |A_3| - |A_4| + |A_5||$

polygon $P' = \{s_0, s_1, ..., s_k, p_i, p_{i+1}, ..., p_j, t_0, t_1, ..., t_l\}$ and have the following equation (See figure 3.b):

$$A(P') = \frac{1}{2}\sum_{i=0}^{n+k+l+2} A(e_i) = \Phi \quad (2)$$

In the second step, we do this computation for $P'' = \{t_l, t_{l-1}, ...t_0, p_j, p_i, s_k, s_{k-1}, ..., s_0\}$ path(See figure 3.c):

$$A(P'') = \frac{1}{2}\sum_{i=0}^{l+k+2} A(e_i) = \Gamma \quad (3)$$

By equation 2 and 3 we have:

$$\begin{aligned}
|A(P^{'}) + A(P^{''})| &= |\Phi + \Gamma| = ||\Phi| + |\Gamma|| \\
&= |A(\{s_0, ..., s_k, p_i\}) + A(P) \\
&= +A(s_0, p_j, t_0, ..., t_l) \\
&\quad +A(\{s_0, t_l, t_{l-1}, ..., t_0, p_j\}) \\
&\quad +A(\{s_0, p_j, p_i\}) \\
&\quad +A(\{s_0, p_i, s_k, s_{k-1}, ..., s_1\})| \\
&= |A(P) + A(\{s_0, p_j, p_i\})| \\
&= \text{Error}_{diff-area}(p_i, p_j)
\end{aligned}$$

Now, using the partial sum we show how error weight computation can be done in $O(n^2)$. Assume that for $0 \leq i < j \leq n$ we have:

$$S_{ij} = \frac{1}{2}\sum_{k=i}^{j-1} A(p_k, p_{k+1}) \quad (4)$$

Using this approach, for each starting point i, we can do all the related computations of $S_{i,j}$ in linear time. Consequently, the total time for computations of n^2 different cases of $S_{i,j}$

will be $O(n^2)$. On the other hand:

$$\begin{aligned}\text{Error}_{diff-area}(p_ip_j) &= A(P(i,j)) \\ &= |S_{i,j} + A(p_jp_i)|\end{aligned}$$

Thus, for all p_ip_j edges, we can compute $\text{Error}_{diff-area}(p_ip_j)$ in $O(n^2)$ time by having $S_{i,j}$ values. More complex paths can be divided to simple parts and be solved in a similar way. From these facts we obtain following lemma

Lemma 1 *The computation of* $\text{Error}_{diff-area}(Q)$ *for all* n^2 *possible shortcuts* $e_{i,j}$ *of general path* P, $i,j \in P$, *can be done in* $O(n^2)$ *time.*

Algorithm 1 Approx Path Simplification Algorithm

```
procedure COMPUTEAPPROXPATH(P:point-set; n,k:integer; γ:real)
    for i ← 2, n do
        for s ← 1, k do
            for r ← ⌊−2k/γ⌋, ⌊2k/γ⌋ do
                if s = 1 then
                    D[i, 1, r] ← 0
                    if w̄(p_1, p_i) = r then
                        D[i, 1, r] ← 1
                    end if
                else
                    tmp ← 0
                    for g ← 2, i − 1 do
                        if D[g, s−1, r−w̄(p_g, p_i)] = 1 then
                            tmp ← 1
                        end if
                    end for
                    D[i, s, r] ← D[i, s − 1, r] ∨ tmp
                end if
            end for
        end for
    end for
return D
end procedure
```

B. Algorithms

There is an efficient general algorithm for restricted, min-k version of the line simplification algorithm [12]. We plug our error function into this algorithm and solve $\mathcal{P}rob^{\text{diff-area}}_{\text{max-simp}}$ optimally. First, a directed acyclic graph G is built over path $P = p_0, p_1, ..., p_n$ vertices. The min-k problem can be solved as follows:

All edges with $\text{Error}_{area}(p_ip_j) > \delta$ are removed from the DAG G. Remaining edges weights are set to 1. Running a shortest path algorithm from p_1 to p_n returns the optimal min-k simplification. Therefore, knowing lemma 1, we have,

Theorem 2 *The optimal min-k simplification for* $\mathcal{P}rob^{\textit{diff-area}}_{\textit{max-simp}}$ *can be computed in* $O(n^2)$ *time and* $O(n^2)$ *space complexities.*

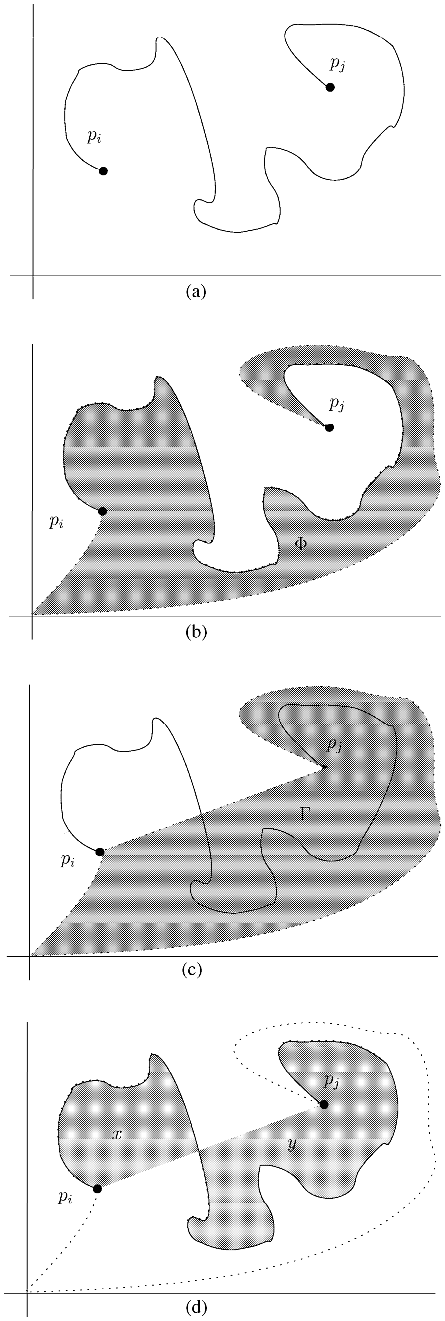

Fig. 3. $\text{Error}_{diff-area}(p_ip_j)$ is equal to $|\Phi - \Gamma| = |x - y|$

Now we use dynamic programming to find an approximation path Q for $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$. This technique has already been employed by other authors too [1].

Let γ be a given factor, H be area of convex hull P, $\Gamma = \gamma H/2k$, $w(e) = \text{Error}_{diff-area}(e)$, $\overline{w}(e)$ be $\lfloor w(e)/\Gamma \rfloor$.

Also, let $W(Q) = \sum_{e \in Q} w(e)$ and $\overline{W}(Q) = \sum_{e \in Q} \overline{w}(e)$ be for path Q. We define Q_{aprx} to be an approximation path that minimizes $|\overline{W}(Q_{aprx})|$ and Q_{optm} to be a path that minimizes $|W(Q_{optm})|$. As it is easy to show that for any approximation path (Q), $||W(Q)| - \Gamma|\overline{W}(Q)|| \leq k\Gamma$, we omit it. Using this fact we can argue that $|W(Q_{aprx})| \leq$

$|W(Q_{optm})| + \gamma H$.

The only thing that remains is to show how dynamic programming works. We use algorithm 1 for computing approximation path Q with at most k points and weight at most γH bigger than the optimal for diff-area. Algorithm returns matrix D. Construction of approximation path will be easy using this matrix. Clearly, running time of this algorithm is $O(n^2k^2/\gamma)$. We conclude all this in following theorem,

Theorem 3 *The min-δ for $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$ and given path P and parameter γ, can be computed approximately in $O(n^2k^2/\gamma)$ time and $O(nk^2/\gamma)$ space complexities with error cost γH larger than the optimal, where H is convex hull of path P.*

III. Conclusion

In this paper, we studied the restricted version of path simplification problem under diff-area measure. Previous results of this problem only worked on x-monotone paths. We have presented optimal algorithm for min-k for $\mathcal{P}rob^{\text{diff-area}}_{\text{max-simp}}$ and approximation algorithm for min-δ for $\mathcal{P}rob^{\text{diff-area}}_{\text{sum-simp}}$. In comparison to current solutions, our algorithms can be used to simplify general paths while keeping the time and space complexities the same as previous methods. It would be interesting to see if better results are possible in the latter case.

References

[1] P. Bose, S. Cabello, O. Cheong, J. Gudmundsson, M. Kreveld, B. Speckmann. *Area-Preserving Approximations of Polygonal Paths.* Journal of Discrete Algorithms, 4(4):554-566, 2006.

[2] R. Weibel. *Generalization of Spatial Data - Principles and Selected Algorithms.* In: Algorithmic Foundations of Geographic Information Systems, 1997.

[3] R.B. McMaster. *Automated Line Generalization.* Cartographica, 24(2):74-111 (1987).

[4] L. Buzer. *Optimal Simplification of Polygonal Chain for Rendering.* 23rd ACM Symp. on Computational Geometry (SoCG), pages 168–174, (2007).

[5] R.B. McMaster. *A Quantitative Analysis of Mathematical Measures in Linear Simplification.* Ph.D. Thesis, Dept. of Geography and Meteorology, University of Kansas, Lawrence, Kansas., (1983).

[6] R.B. McMaster. *The Geometric Properties of Numerical Generalization.* Geographical Analysis, 19(4) : 330-346 (1987).

[7] M. Visvalingam, J.D. Whyatt. *Line Generalisation by Repeated Elimination of Points.* Cartographic Journal, 30(1): 46-51 (1993).

[8] H. Veregin. *Line Simplification, Geometric Distortion, and Positional Error.* Cartographica, v36(1). 25-39 (1999).

[9] L. Chen, J. Wang, J. Xu. *Asymptotically Optimal and Linear-time Algorithm for Polygonal Curve Simplification.* IEEE Transactions on Pattern Analysis and Machine Intelligence, (2005).

[10] B. Aronov, T. Asano, N. Katoh, K. Mehlhorn, T. Tokuyama. *Polyline fitting of planar points under min-sum criteria.* International Journal of Computational Geometry and Applications, 16 (2006) 97-116.

[11] Sh. Daneshpajouh, A. Zarei, M. Ghodsi, *On Realistic Line Simplification under Area Measure.* IAENG International Conference on Computer Science (ICCS'09), 2009.

[12] H. Imai, M. Iri. *Polygonal approximations of a curve-formulations and algorithms.* Computational Morphology, North-Holland, pages 71–86, (1988)

[13] F. Contreras-Alcala. *Cutting polygons and a problem on illumination of stages.* Master thesis, Dept. Comput. Sci. University of Ottawa., Ottawa Ont, Canada, 1998.

[14] P. Boland, J. Urrutia. *Polygon Area Problems.* Canadian Conference on Computational Geometry, Canada, 2008.