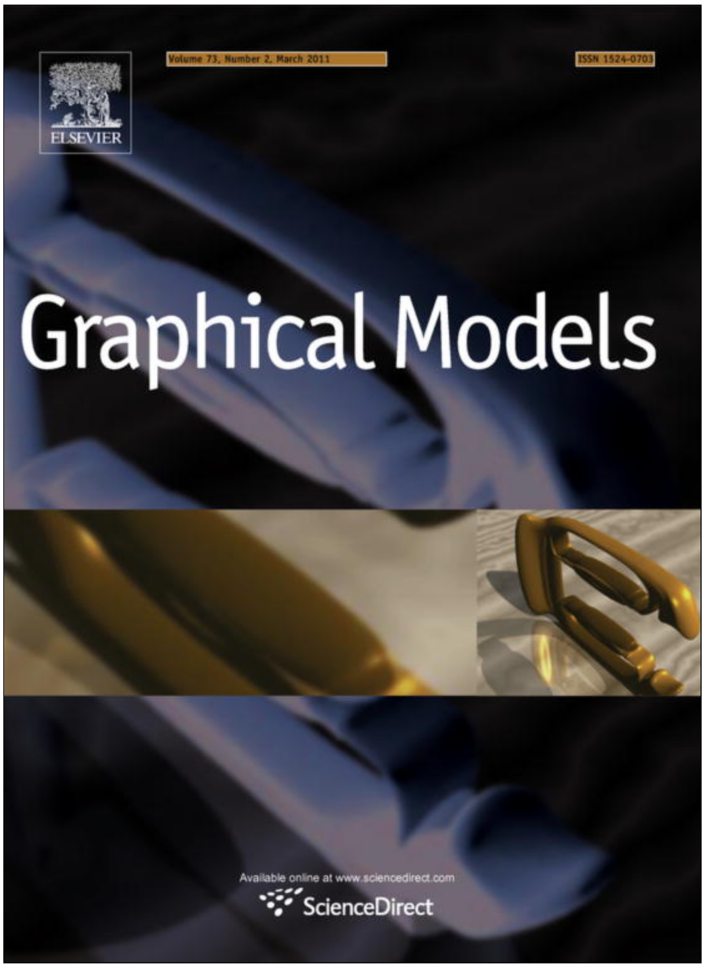
Volume 73, Number 2, March 2011
ISSN 1524-0703
ELSEVIER
Graphical Models
Available online at www.sciencedirect.com
ScienceDirect

Graphical Models 74 (2012) 283–289

Contents lists available at SciVerse ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Computing polygonal path simplification under area measures ☆

Shervin Daneshpajouh [a,*], Mohammad Ghodsi [a], Alireza Zarei [b]

[a] Computer Engineering Department, Sharif University of Technology, Iran
[b] Mathematics and Computer Science Department, Sharif University of Technology, Iran

ARTICLE INFO

Article history:
Received 23 March 2011
Received in revised form 8 April 2012
Accepted 17 April 2012
Available online 26 April 2012

Keywords:
Computational geometry
Linear model simplification
Area difference

ABSTRACT

In this paper, we consider the restricted version of the well-known 2D line simplification problem under area measures and for restricted version. We first propose a unified definition for both of sum-area and difference-area measures that can be used on a general path of n vertices. The optimal simplification runs in $O(n^3)$ under both of these measures. Under sum-area measure and for a realistic input path, we propose an approximation algorithm of $O\left(\frac{n^2}{\epsilon}\right)$ time complexity to find a simplification of the input path, where ϵ is the absolute error of this algorithm compared to the optimal solution. Furthermore, for difference-area measure, we present an algorithm that finds the optimal simplification in $O(n^2)$ time. The best previous results work only on x-monotone paths while both of our algorithms work on general paths. To the best of our knowledge, the results presented here are the first sub-cubic simplification algorithms on these measures for general paths.

1. Introduction

Line simplification, also referred to as line generalization or curve simplification in some literatures, is a basic problem in imaging, cartographic, computational geometry, and geographic information systems (GISs). In this problem, there is a sequence of n input points defining a simple (non-intersecting) path $P = \langle p_1, p_2, \ldots, p_n \rangle$ and we are asked to approximate this by another path $Q = \langle q_1 = p_1, q_2, \ldots, q_k = p_n \rangle$ of smaller number of vertices. This problem has many applications wherever data reduction is needed for space and complexity purposes. Examples are map representation, path tracking and geometric shape modeling.

There are two main versions of this problem. In the *restricted* version, the vertices q_i of Q must be a subsequence of the vertices of P, while in the *unrestricted* version, the vertices q_i can be anywhere in the plane. Some results on the unrestricted version can be found in [12,13,17].

In this paper, we consider the restricted version. For this problem, two optimization goals have been proposed: (1) min-k, in which the goal is to find a simplification with minimum number of vertices and error of at most δ, and (2) min-δ, in which the goal is to find a simplification of at most k vertices with the minimum simplification error. Here, we focus on the min-k problem.

The error of a simplification Q using an error measure m, denoted by $\mathrm{E_m}(Q)$, is either defined to be $\max_{i=1}^{k-1} \mathrm{E_m}(q_i q_{i+1})$ or $\sum_{i=1}^{k-1} \mathrm{E_m}(q_i q_{i+1})$, where $q_i q_{i+1}$ is the simplification of the sub-path $P(s,t) = \langle p_s = q_i, p_{s+1}, \ldots, p_t = q_{i+1} \rangle$ and $\mathrm{E_m}(q_i q_{i+1})$ is the associated error of approximating $P(s,t)$ by $q_i q_{i+1}$ under the error measure m. The main simplification error measures are Hausdorff distance, Fréchet distance, and areal displacement. A survey and comparison of these error measures can be found in [6]. In this paper, we only consider the area error measure and use $\max_{i=1}^{k-1} \mathrm{E_m}(q_i q_{i+1})$ as the simplification error.

1.1. Related work

The oldest and most popular approach for line simplification was proposed by Douglas and Peucker which is known as Douglas–Peucker algorithm [11]. A basic implementation

☆ This paper has been recommended for acceptance by Tamal Dey and Pierre Alliez.

* Corresponding author.
E-mail address: daneshpajouh@ce.sharif.edu (S. Daneshpajouh).

1524-0703/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.gmod.2012.04.006

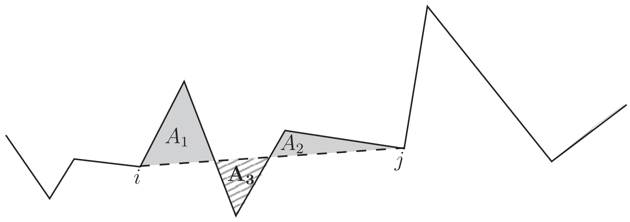

Fig. 1. The sub-path $P(i, j)$ has been simplified by p_ip_j. The gray and hatched areas show the above and below areas enclosed by the sub-path and its shortcut, respectively.

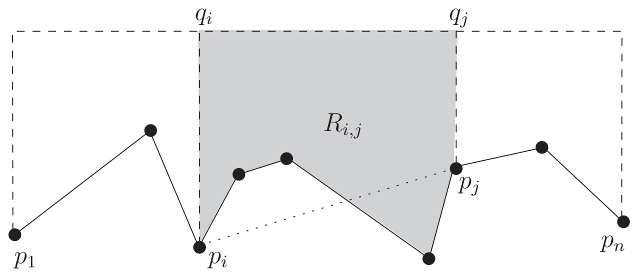

Fig. 2. Computation of the diff-area error measure for x-monotone paths.

of this algorithm for orthogonal distance error measure runs in $O(n^2)$ time. Other implementations improved the running time to $O(n\log n)$ [15] and $O(n\log^* n)$ [16]. However, this algorithm is a heuristic without any guarantees about the quality of the resulting approximation.

The first general algorithm was proposed by Imai and Iri [18] which produces the optimal simplification. They modeled the problem by a directed acyclic graph and showed that solving the shortest path on this graph is equivalent to the optimal simplification. Moreover, they showed how this graph can be constructed for orthogonal distance measure in $O(n^2\log n)$ time. This running time was improved to quadratic or near quadratic [7,20], and $O(n^{4/3})$ [2] for L_1 and uniform metrics. Finally, a near linear time approximation algorithm was proposed in [1] for L_2 orthogonal distance.

The line simplification under the Fréchet distance was first studied in [14]. For this error measure, the optimal solution can be obtained using the results from [18,3].

1.2. The Area measure

Assume that a sub-path $P(i, j)$ has been simplified by the link p_ip_j which connects the endpoints of this sub-path. For such a simplification, as shown in Fig. 1, there are regions enclosed by $P(i, j)$ and p_ip_j on both sides of p_ip_j. For link p_ip_j in Fig. 1 these regions are $\{A_1, A_2, A_3\}$. Using the area of union of these regions is a natural parameter to be used as the error of simplification of p_ip_j. This error measure is called *sum-area* and has been considered before [5,23,24]. In some applications, like simplification of borders between two neighboring countries, the difference between the areas of the regions defined by p_ip_j and $P(i, j)$ on both sides of p_ip_j are important. This error measure is called *diff-area* and has been considered in [5]. Diff-area for link p_ip_j in Fig. 1 is $(A_1 + A_2) - A3$.

The sum-area and diff-area error measures are natural measures in graphic applications. In such applications, the area of a simplified region should be as close as possible to the area of the original shape. This is natural while the area is an important factor from which the similarity or difference between two regions can be conceived.

The first optimal simplification algorithm under the sum-area measure was proposed by Veregin [24] which computes the error of all simplifications built on all possible combinations of the vertices of the path. However, this algorithm is exponential and is not useful in practice. There are several approximation or heuristic algorithms, like the method presented by Visvalingam and Whyatt [23]. The main drawback of these methods is that there is no guarantee on the deviation of their results from the optimal solution.

Recently, Bose et al. [5] revisited the line simplification problem under area error measures and proposed simplification algorithms for x-monotone paths. For sum-area error measure, they bounded the input path by two vertical lines at p_1 and p_n and a horizontal segment lying above P according to Fig. 2. Then, they recursively partition this bounded area R and compute the area above and below P for every sub-path $P(i, j)$ using Langerman's algorithm [19]. For a given simple polygon Z with n vertices and m line queries, Langerman's algorithm computes the area of Z on both sides of each line in time $O(m^{2/3}n^{2/3+\epsilon} + (n + m)polylog\ n)$ for any $\epsilon > 0$. Using this approach, they compute and build the directed acyclic graph G of the Imai and Iri's method [17] in $O(n^{2+\epsilon})$ time and space. For diff-area error measure, they again bounded the input path as Fig. 2. Let $T_{i,j}$ be the trapezoid described by p_i, p_j, q_j, q_i. Clearly, $E_d(p_ip_j) = Ar(R_{i,j}) - Ar(T_{i,j})$ and $Ar(R_{i,j}) = Ar(R_{1,j}) - Ar(R_{1,i})$, where $Ar(S)$ is the area of a closed region S. Using these facts, and computing values of $R_{1,j}$,$E_d(p_ip_j)$ for all possible links p_ip_j can be obtained in $O(n^2)$ time. Obviously, these methods work only for x-monotone paths and cannot be used for general paths. In many applications like map rendering, paths are not x-monotone. Therefore, it is interesting and important to study this problem in general cases.

1.3. Our results

Trying to use the sum-area and diff-area error measures for line simplification, we faced different definitions for these measures none of which was well applicable on a general path. Therefore, we first describe a unified definition for the sum-area and diff-area error measures that can be applied on general paths. Further, for this unified area measure, we employ Imai and Iri's general approach and propose an $O(n^3)$ simplification algorithm to obtain the optimal min-k simplifications under both sum-area and diff-area error measures.

Running time complexity of this algorithm is too high to be used in practical applications. We resolve this problem by proposing quadratic or near quadratic algorithms for the unified definition of sum-area and diff-area error measures. For sum-area, we propose a near quadratic algorithm that approximately solves the min-k simplification. Precisely, the running time of our algorithm is $O\left(\frac{n^2}{\epsilon}\right)$ and the error of the resulting simplification is at most ϵL^2

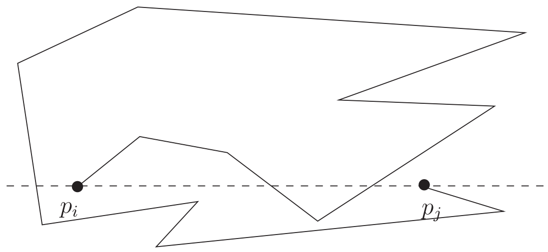

Fig. 3. A complex sub-path simplified by p_ip_j.

farther than the error of the optimal min-k simplification, where L is the length of the longest shortcut of the simplification. We assume that in a realistic application the path lies entirely inside a region of bounded edge length. Then, the length of the longest link of any simplification is constant which means that L is bounded by some constant value. Therefore, this near quadratic time algorithm can be used to obtain arbitrarily close approximations to the optimal simplification. For diff-area error measure we propose an algorithm that computes the diff-area error of all possible shortcuts in $O(n^2)$ time. Consequently, we have an optimal simplification algorithm that runs in $O(n^2)$ time and minimizes $\max_{i=1}^{k-1} \mathrm{E_d}(q_iq_{i+1})$. To the best of our knowledge, these algorithms are the best ones that can be used to simplify general paths under the sum-area and diff-area error measures.

In the rest of this paper, in Section 2, we first describe a unified method for computing the area shaped between a chain and a line segment, to be used for computing the sum-area and diff-area error measures. Then, we propose optimal simplification algorithms for these error measures. In Section 3, we propose the near quadratic approximation algorithm that simplifies general paths under the sum-area error measure. In Section 4, we present the optimal simplification algorithm under the diff-area error measure. We offer the conclusion in Section 5.

2. Optimal simplification under area measures

In this section, we first revisit the definition of the area enclosed by a path and a line segment and propose a uniform definition which covers any path. We show that current line simplification algorithms can be used under this definition of sum-area and diff-area error measures.

2.1. The area measure: revisited

Assume that we have a sub-path $P(i,j) = \langle p_i, p_{i+1}, \ldots, p_j \rangle$ simplified by the link p_ip_j. The error of this simplification under the area measure, depends on the area of the region enclosed by $P(i, j)$ and p_ip_j. In general, the sub-path $P(i, j)$ may intersect p_ip_j. The enclosed region may be too complex to identify and compute its area. An example of such complex paths is shown in Fig. 3. Therefore, we need a definition that covers all paths.

We first distinguish the areas lying to the left and to the right of a link. If we are at point p_i and look toward point p_j, some parts of the sub-path lie on our left and the other parts lie on our right. Hence, we have two values defining

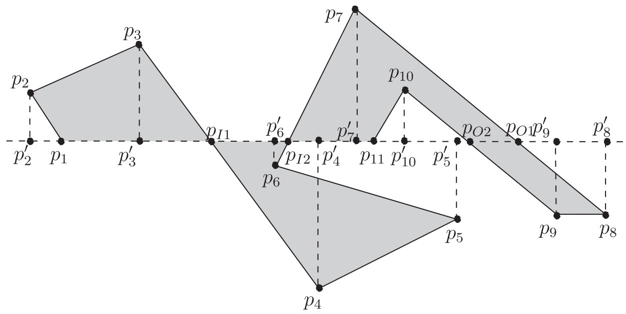

Fig. 4. The area enclosed by sub-path $P(1, 11)$ and the link p_1p_{11}.

the area of the region enclosed by p_ip_j and $P(i, j)$: the left area and the right area which are respectively denoted by $\mathrm{Ar}_l(p_ip_j)$ and $\mathrm{Ar}_r(p_ip_j)$. Then, the error of link p_ip_j in terms of sum-area and diff-area $\mathrm{E_s}(p_ip_j) = \mathrm{Ar}_l(p_ip_j) + \mathrm{Ar}_r(p_ip_j)$ and $\mathrm{E_d}(p_ip_j) = |\mathrm{Ar}_l(p_ip_j) - \mathrm{Ar}_r(p_ip_j)|$ where $|\cdot|$ denotes the absolute value.

Now, we describe how to compute $\mathrm{Ar}_l(p_ip_j)$ and $\mathrm{Ar}_r(p_ip_j)$. Assume that $P(i, j)$ intersects link p_ip_j in points $\langle p_{I1}, p_{I2}, \ldots, p_{Ik} \rangle$. Then, p_ip_j is divided into $U = \langle p_ip_{I1}, p_{I1}p_{I2}, \ldots, p_{Ik-1}p_{Ik}, p_{Ik}, p_j \rangle$ segments. Note that we distinguish between the intersection points of $P(i, j)$ and p_ip_j and the intersection points of $P(i, j)$ and the supporting line of p_ip_j (the line that passes through p_i and p_j). Each segment $p_{Ix}p_{Iy} \in U$ and its related sub-path $P(Ix, Iy)$ define a polygon. This polygon is simple because $P(Ix, Iy))$ does not intersect itself and p_{Ix} and p_{Iy} are two consecutive points in U. We denote this polygon by $\Delta(Ix, Iy)$. We say that $\Delta(Ix, Iy)$ lies on the left (right) side of p_ip_j, if each edge of $\Delta(Ix, Iy)$ can be connected to the left (right) side of $p_{Ix}p_{Iy}$ by a path inside the polygon $\Delta(Ix, Iy)$. To calculate the values of $\mathrm{Ar}_l(p_ip_j)$ and $\mathrm{Ar}_r(p_ip_j)$, we calculate the area of each $\Delta(Ix, Iy)$ for all $p_{Ix}p_{Iy} \in U$ and based on their sides, we add these values to $\mathrm{Ar}_l(p_ip_j)$ or $\mathrm{Ar}_r(p_ip_j)$.

The area of each $\Delta(Ix, Iy)$, for each $p_{Ix}p_{Iy} \in U$ is defined as follows:

$$\sum_{p_sp_t \in P(Ix,Iy)} \text{Signed-Area}(p'_sp_sp_tp'_t)$$

where p_sp_t is an edge of $P(Ix, Iy)$, $x \leqslant s < t \leqslant y$ and p'_s and p'_t are respectively the orthogonal projection of points p_s and p_t on the supporting line of $p_{Ix}p_{Iy}$. If p_sp_t intersects the supporting line of $p_{Ix}p_{Iy}$, each part of this edge is treated separately. The absolute value of $\text{Signed-Area}(p'_sp_sp_tp'_t)$ is the area of the trapezoid $p'_sp_sp_tp'_t$ and its sign is defined as follows:

$$\operatorname{sign}(\text{Signed-Area}(p'_sp_sp_tp'_t)) = \begin{cases} + & \begin{array}{l} \text{if } \overrightarrow{p_{Ix}p_{Iy}} \text{ and } \overrightarrow{p'_sp'_t} \text{ have the same directions} \\ \text{and } p_sp_t \text{ and } \Delta(Ix,Iy) \text{ lie on the same} \\ \text{side of } p_{Ix}p_{Iy} \\ \textit{or} \\ \text{if } \overrightarrow{p_{Ix}p_{Iy}} \text{ and } \overrightarrow{p'_sp'_t} \text{do not have the} \\ \text{same directions and } p_sp_t \text{ and } \Delta(Ix,Iy)) \\ \text{lie on the opposite sides of } p_{Ix}p_{Iy}. \end{array} \\ - & \text{otherwise} \end{cases}$$

It is easy to prove by induction on the length of the sub-path $P(Ix, Iy)$ that the above method is equal to the area of the simple polygon $\Delta(Ix, Iy)$. Note that, the sign of the computed area is negative if the direction of the edges in $P(Ix, Iy)$ are not the same as the direction of those edges in $P(i, j)$.

As an example, assume that p_1p_{11} is the simplification of sub-path $P(1, 11)$ in Fig. 4. The area of a polygon $p_i p_{i+1} \ldots p_j$ is denoted by $\mathrm{Ar}(p_ip_{i+1} \cdots p_j)$. According to our definition,

$$\begin{aligned}\mathrm{Ar}_l(p_1p_{11}) &= \mathrm{Ar}(\Delta(1, I1)) + \mathrm{Ar}(\Delta(I2, 11)) \\ &= \big(-\mathrm{Ar}(p_1p_2p_2') + \mathrm{Ar}(p_2'p_2p_3p_3') + \mathrm{Ar}(p_3'p_3p_{I1})\big) \\ &\quad + \big(\mathrm{Ar}(p_{I2}p_7p_7') + \mathrm{Ar}(p_7'p_7p_{O1}) - \mathrm{Ar}(p_{O1}p_8'p_8) \\ &\quad + \mathrm{Ar}(p_8p_8'p_9'p_9) + \mathrm{Ar}(p_9'p_9p_{O2}) - \mathrm{Ar}(p_{O2}p_{10}p_{10}') \\ &\quad - \mathrm{Ar}(p_{10}'p_{10}p_{11})\big),\end{aligned}$$

and

$$\begin{aligned}\mathrm{Ar}_r(p_1p_{11}) &= \mathrm{Ar}(\Delta(I1, I2)) \\ &= \mathrm{Ar}(u_1p_4p_4') + \mathrm{Ar}(p_4'p_4p_5p_5') - \mathrm{Ar}(p_5'p_5p_6p_6') \\ &\quad + \mathrm{Ar}(p_6p_6'p_{I2})\end{aligned}$$

It is simple to verify that this computation is equal to the area of the gray regions in Fig. 4.

2.2. Optimal simplification algorithm

The definition of the area measures presented in Section 2.1 can be used as a unified and general definition and can be applied on any paths. We plug our error functions into the general algorithm of Imai and Iri [18] and solve the min-k version of the problem optimally. First, we compute $\mathrm{E_s}(p_ip_j)$ or $\mathrm{E_d}(p_ip_j)$ by a linear trace on the sub-path $P(i, j)$ in $O(j - i)$ time. There are $O(n^2)$ possible links for which the error must be computed. Consequently, we can do this computation for all p_ip_j links in $O(n^3)$ time. We build a directed acyclic graph G over the vertices of path $P = p_0, p_1, \ldots, p_{n-1}$ and solve the min-k problem as follows:

All edges whose weights (the error of the corresponding link which is $\mathrm{E_s}(p_ip_j)$ or $\mathrm{E_d}(p_ip_j)$) are greater than the given δ are removed from the DAG. Weights of the remaining edges are set to 1. Running a shortest path algorithm from p_1 to p_n returns the optimal min-k simplification. Therefore,

Theorem 1. *The optimal min-k simplification under the sum-area and diff-area error measures can be computed in* $O(n^3)$ *time and* $O(n^2)$ *space.*

3. An approximation algorithm for simplification under sum-area error measure

The time complexity of the optimal $O(n^3)$ algorithm is too high to be used in practical applications. In this section we propose a near quadratic time algorithm to compute the simplification under sum-area error measure. However, the resulting simplification is not optimal.

The idea of this approximation algorithm is to use the information resulted in computing $\mathrm{E_s}(p_ip_j)$ to efficiently compute $\mathrm{E_s}(p_ip_{j+1})$. This is done by computing and maintaining the error of the current sub-path for a set of canonical lines C drawn from the start vertex of the path (here p_i). For the next point, p_{j+1}, we determine the two canonical lines where p_{j+1} lies between them (from now on, we call these two lines c and c'). We approximate the error of p_ip_{j+1} by the errors of these two lines.

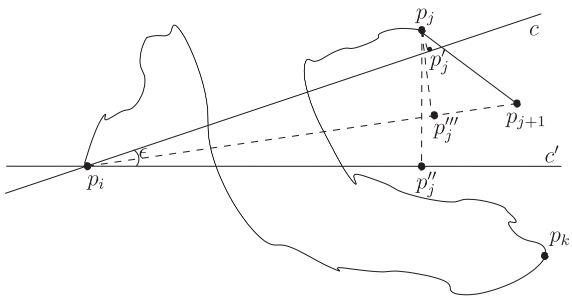

Fig. 5. Approximating the sum-area error of a link.

Assume that for a sub-path $P(i, j)$ we have the exact values of $\mathrm{Ar}_l\left(p_ip_j'\right)$, $\mathrm{Ar}_r\left(p_ip_j'\right)$, $\mathrm{Ar}_l\left(p_ip_j''\right)$ and $\mathrm{Ar}_r\left(p_ip_j''\right)$ where p_j' and p_j'' are respectively the orthogonal projections of p_j on lines c and c' drawn from p_i (See Fig. 5).

For the next point p_{j+1}, we use $\left|\mathrm{Ar}_l\left(p_ip_j'\right) + \mathrm{Ar}_r\left(p_ip_j''\right) + \text{Signed-Area}^*\left(p_jp_j'''p_{j+1}\right)\right|$ as an approximation for $\mathrm{E_s}(p_ip_{j+1})$ where c lies on the left of p_ip_{j+1}, c' lies on the right of p_ip_{j+1}, p_j''' is the orthogonal projection of p_j on the supporting line of p_ip_{j+1} and $\text{Signed-Area}^*\left(p_jp_j'''p_{j+1}\right)$ is signed area of the triangle $p_jp_j'''p_{j+1}$ with a sign that is positive if and only if $\overrightarrow{p_ip_{j+1}}$ and $\overrightarrow{p_j'''p_{j+1}}$ have the same directions.

We incrementally compute the area above and below of each canonical line $c \in C$. Precisely, for each $c \in C$ and each p_jp_{j+1}, we compute a signed area for trapezoid $p_jp_{j+1}p_{j+1}'p_j'$ denoted by $\nabla(c, i, j)$ where p_j' and p_{j+1}' are orthogonal projections of p_j and p_{j+1} on c, respectively. The sign of this area is positive if and only if $\overrightarrow{p_jp_{j+1}}$ and $\overrightarrow{p_j'p_{j+1}'}$ have the same directions. It is easy to see that $\sum_{p_sp_t \in P(i,j)} \nabla(c, s, t)$ is equal to the area computed using our uniform method $\sum_{p_sp_t \in P(i,j)} \text{Signed-Area}(p_s'p_sp_tp_t')$.

We denote this approximated value by $\mathrm{E_s^*}(p_ip_{j+1})$.

Lemma 2. *We have the following relation between the approximation and exact values of sum-area error of a link* p_ip_{i+1}.

$$\mathrm{E_s}(p_ip_{j+1}) - \frac{\epsilon|p_ip_k|^2}{2} \leqslant \mathrm{E_s^*}(p_ip_{j+1})$$

and

$$\mathrm{E_s^*}(p_ip_{j+1}) \leqslant \mathrm{E_s}(p_ip_{j+1}) + \frac{\epsilon|p_ip_k|^2}{2}$$

where ϵ *is the angle between c and* c' *containing point* p_{j+1} *and* p_k *is the farthest point of P(i, j) from* p_i.

Proof. Assuming that ϵ is small enough, we can conclude that $\mathrm{Ar}_l\left(p_ip_j'\right) - \frac{\epsilon|p_ip_k|^2}{2} \leqslant \mathrm{Ar}_l(p_ip_{j+1})$ and $\mathrm{Ar}_r\left(p_ip_j''\right) - \frac{\epsilon|p_ip_k|^2}{2} \leqslant \mathrm{Ar}_r(p_ip_{j+1})$.

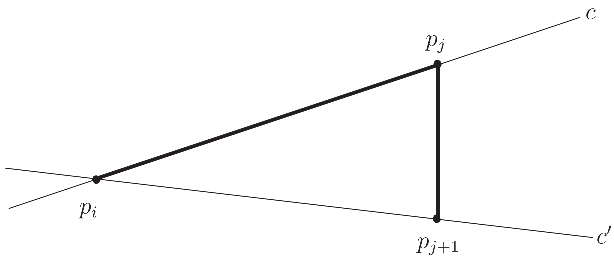

Fig. 6. A tight example for the approximation of $E_s(p_ip_{j+1})$.

This difference is related to the area that lies between c and c'. According to the definition of p_k; this area is at most $\frac{\epsilon|p_ip_k|^2}{2}$ when ϵ is arbitrarily small. Fig. 6 shows a tight example. □

Therefore, if we have these canonical lines for an arbitrarily small value of ϵ, we can approximate the sum-area error of the next point in constant time, using the left and right errors of the path $P(i,j)$ on these canonical lines. However, we need to update the left and right areas of these canonical lines against the newly received vertex to be able to approximate the error of the next point.

Lemma 3. *There is an $O(\frac{2\pi}{\epsilon}n)$ time algorithm that approximately computes the sum-area error of all links p_1p_i for $1 < i \leqslant n$.*

Proof. We have $\frac{2\pi}{\epsilon}$ canonical lines from p_1 and on receiving a new point, values of Ar_l and Ar_r are updated for these lines. Then, the approximated error of the new link is computed in constant time. □

For any vertex p_i we can apply the above method. Then, we can find the approximated error of all links p_ip_j in $O(\frac{2\pi}{\epsilon}n^2)$ time. As mentioned before, in a realistic scene we are working in a bounded region. Then the distance between any two points is smaller than a constant value. Therefore, we can omit the $|p_ip_k|$term in Lemma 2. Combining these results, we have:

Theorem 4. *There is an $O\left(\frac{n^2}{\epsilon}\right)$ time algorithm that finds a near optimal simplification under the sum-area error measure. The error of the resulting simplification differs from the error of the optimal simplification by $O(\epsilon)$ in a realistic scene.*

4. Efficient simplification algorithm for diff-area error measure

In this section, we present an algorithm for efficient computation of error of all shortcuts under diff-area measure. In this method, we compute the $E_d(p_ip_j)$ for all links p_ip_j in $O(n^2)$ time. Then, we can use the general simplification algorithm described in Section 2.2 to find optimal simplification under the diff-area error measure in $O(n^2)$ time.

There are different methods for computing the area of a simple polygon [9]. Our algorithm is based on a method called *Polar formula*. Let $\overrightarrow{p_ip_j}$ be a directed edge from p_i to p_j. For each edge $\overrightarrow{p_ip_j}$, $\mathrm{Ar}(\overrightarrow{p_ip_j})$ is defined as $x(p_i)y(p_j) - x(p_j)y(p_i)$ in which $x(p)$ is x coordinate of p and $y(p)$ is its y coordinate. It is proved that $|\mathrm{Ar}(\overrightarrow{p_ip_j})|$ is twice the area of the triangle formed by vertices p_i, p_j and (0, 0) [9].

Using this fact, the area of a simple polygon $Y = \langle y_1, \ldots, y_n\rangle$ with $e_i = y_iy_{i+1}$:$0 \leqslant i < n$ and $e_n = y_ny_1$ can be computed using the following formula [9]:

$$\mathrm{Ar}(Y) = \left|\frac{1}{2}\sum_{i=0}^{n}\mathrm{Ar}(e_i)\right| \tag{1}$$

An example of this computation is shown in Fig. 7. We use this result to compute $E_d(p_ip_j)$. Precisely, we show that if the input path P is simple, then $\mathrm{Ar}(Y = P(i,j)) = E_d(p_ip_j)$ which means that the Eq. (1) is compatible with our diff-area error measure defined in Section 2.1. In the first step, we connect the origin o to p_i using some extra points in such a way that they do not intersect path $P(i, j)$. Let $S = \langle s_0, s_1, \ldots, p_k\rangle$ be the set of these extra points. We do the same to connect p_j to o using a sequence of points $T = \langle t_0, t_1, \ldots, t_l\rangle$ (See Fig. 8b) and build a simple polygon $P' = \langle o, s_0, s_1, \ldots, s_k, p_i, p_{i+1}, \ldots, p_j, t_0, t_1, \ldots, t_l\rangle$ which has a vertex at origin. Let Φ be the area of the polygon described by P'. From Eq. (1) we have:

$$\Phi = \mathrm{Ar}(P') = \frac{1}{2}\sum_{i=0}^{j-i+k+l+2}\mathrm{Ar}(e'_i) \tag{2}$$

Now, consider the polygon $P'' = \langle t_l, t_{l-1}, \ldots, t_0, p_j, p_i, s_k, s_{k-1}, \ldots, s_0\rangle$ formed by the shortcut $\overrightarrow{p_jp_i}$ and the points in T and S (see Fig. 8c). Let Γ be the area of the polygon described by P''. Applying Eq. (1) we have:

$$\Gamma = \mathrm{Ar}(P'') = \frac{1}{2}\sum_{i=0}^{l+k+4}\mathrm{Ar}(e''_i) \tag{3}$$

It is simple to see that,

$$\Gamma = \Phi - \mathrm{Ar}_l(p_ip_j) + \mathrm{Ar}_r(p_ip_j) \tag{4}$$

Therefore, we have:

$$|\Gamma - \Phi| = |\mathrm{Ar}_r(p_ip_j) - \mathrm{Ar}_l(p_ip_j)| = E_d(p_ip_j)$$

Using this result, we propose a simple method to compute all $O(n^2)$ values of $E_d(p_ip_j)$ in $O(n^2)$ time which is an $O(n)$ improvement over the naïve $O(n^3)$ method. Assume that for $0 \leqslant i < j \leqslant n$, we have all values of:

$$S_{i,j} = \frac{1}{2}\sum_{k=i}^{j-1}\mathrm{Ar}(p_kp_{k+1}) \tag{5}$$

We fix the starting point i and compute the values for all $S_{i,j}$, for all possible $j, j > i$, in linear time. Consequently, the total running time of computing the $O(n^2)$ different $S_{i,j}$ values is $O(n^2)$. Knowing $S_{i,j}$ values, we can compute $E_d(p_ip_j)$ in $O(n^2)$ time for all p_ip_j shortcuts:

$$E_d(p_ip_j) = |S_{i,j} + \mathrm{Ar}(p_jp_i)| \tag{6}$$

Consequently, we obtain the following lemma:

Lemma 5. *The computation of $E_d(p_ip_j)$ for all $O(n^2)$ possible shortcuts of a general path P can be done in $O(n^2)$ time.*

Therefore, we have the following theorem:

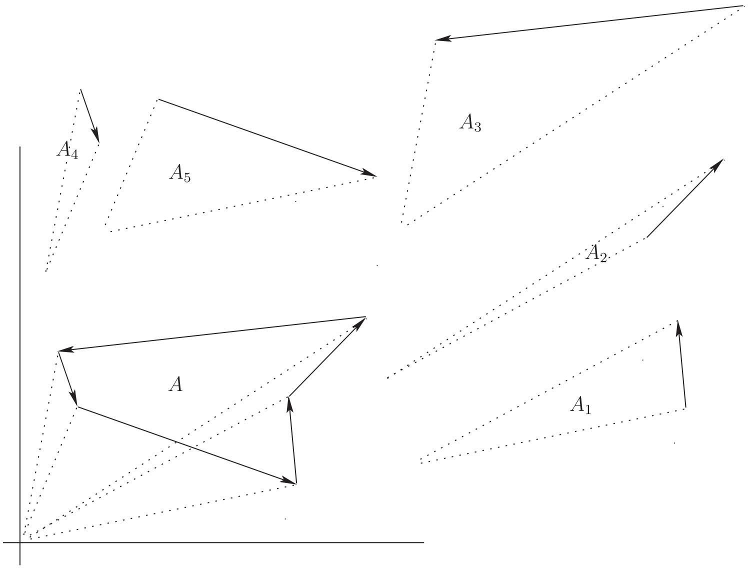

Fig. 7. $\text{Ar}(A) = |\text{Ar}(A_1)| + |\text{Ar}(A_2)| + |\text{Ar}(A_3)| - |\text{Ar}(A_4)| - |\text{Ar}(A_5)|$.

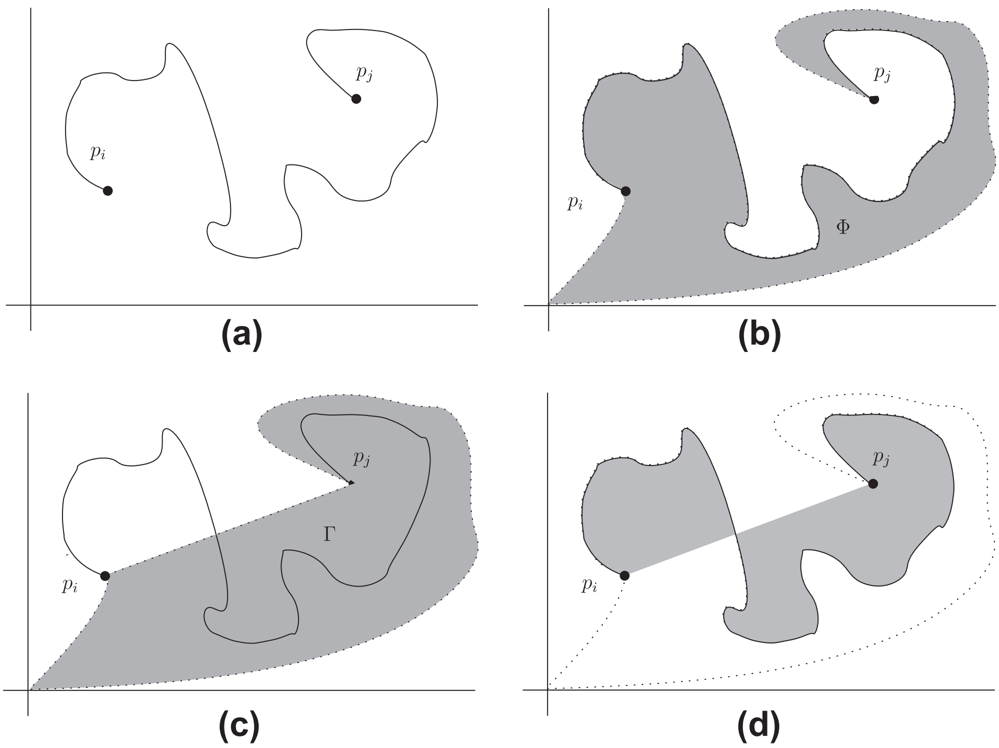

Fig. 8. Computation of $\text{E}_\text{d}(p_ip_j)$.

Theorem 6. *There is an $O(n^2)$ time algorithm that can be used to find an optimal simplification under diff-area error measure.*

Remark 1. For simplification under $\sum_{i=1}^{k-1}\text{E}_\text{m}(q_iq_{i+1})$ error measure and min-δ, we can combine our presented algorithms for sum-area and diff-area with the dynamic programming technique used in [5] and achieve approximation simplification. It will take $O(n^2k^2/\gamma + n^2/\epsilon)$ and $O(n^2k^2/\gamma)$ time for sum-area and diff-area problems where k is the size of the simplification and γ and ϵ are the given approximation factors. Note that it has been proved that the min-δ simplification under $\sum_{i=1}^{k-1}\text{E}_\text{s}(q_iq_{i+1})$ and $\sum_{i=1}^{k-1}\text{E}_\text{d}(q_iq_{i+1})$ are NP-Hard problems [5]. □

5. Conclusion

In this paper, we considered the well-known line simplification problem under the sum-area and diff-area error measures. Previous optimal algorithms are either too costly to be used in real applications or work only on the special case of x-monotone paths. Therefore, heuristic and non-optimal solutions are always used in real applications.

Because of the non-uniform definitions of the area error measures, we first proposed a unified definition for sum-area and diff-area error measures that can be used on general paths. For this definition, we described an $O(n^3)$ time algorithm that obtains an optimal solution for the min-k simplification under both sum-area and diff-area error measures. Furthermore, we proposed a near quadratic approximation algorithm that can be used for simplifying a general 2D path under the sum-area error measure. Moreover, we presented an efficient simplification algorithm for the problem under diff-area error measure.

The current algorithms are still super linear and proposing sub-quadratic or near linear approximation algorithms is an open direction for extending this work.

References

[1] P.K. Agarwal, S. Har-Peled, N.H. Mustafa, Y. Wang, Near-linear time approximation algorithms for curve simplification, Algorithmica 42 (2005) 203–219.

[2] P.K. Agarwal, K.R. Varadarajan, Efficient algorithms for approximating polygonal chains, Discrete Computational Geometry 23 (2000) 273–291.

[3] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, International Journal of Computational Geometry and Applications 5 (1995) 75–91.

[4] P. Bose, S. Cabello, O. Cheong, J. Gudmundsson, M. Kreveld, B. Speckmann, Area-preserving approximations of polygonal paths, Journal of Discrete Algorithms 4 (4) (2006) 554–566.

[5] L. Buzer, optimal simplification of polygonal chain for rendering, in: 23rd ACM Symposium on Computational Geometry (SoCG), 2007, pp. 168–174.

[6] W.S. Chan, F. Chin, Approximation of polygonal curves with minimum number of line segments, in: Proceeding of 3rd Annual International Symposium on Algorithms and Computation, LNCS, vol. 650, 1992, pp. 378–387.

[7] F. Contreras-Alcala, Cutting Polygons and a Problem on Illumination of Stages. Master thesis, Dept. Comput. Sci. University of Ottawa., Ottawa Ont, Canada, 1998.

[8] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122.

[9] M.T. Goodrich, Efficient piecewise-linear function approximation using the uniform metric, Journal of Discrete Computational Geometry 14 (4) (1995) 45–462.

[10] L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell, J.S. Snoeyink, Approximating polygons and subdivisions with minimum link paths, International Journal of Computational Geometry & Applications 3 (1993) 383–415.

[11] M. Godau, A natural metric for curves: computing the distance for polygonal chains and approximation algorithms, in: Proceeding of 8th annual symposium on theoretical aspects of computer science (STACS), 1991, pp. 127–136,.

[12] J. Hershberger, J. Snoeyink, An $O(n\log n)$ implementation of the Douglas–Peucker algorithm for line simplification, in: Proceeding of 10th ACM Symposium on Computational Geometry, 1994, pp. 383–384.

[13] J. Hershberger, J. Snoeyink, Cartographic line simplification and polygon CSG formulae in $O(n\log^* n)$ time, in: Proceeding of 5th International Workshop on Algorithms and Data Structures (WADS), LNCS, vol. 1272, 1997, pp. 93–103.

[14] H. Imai, M. Iri, An optimal algorithm for approximating a piecewise linear function, Journal of Information Processing 9 (3) (1986) 159–162.

[15] H. Imai, M. Iri, Polygonal Approximations of a Curve-Formulations and Algorithms, Computational Morphology, North-Holland, 1988, pp. 71–86.

[16] S. Langerman, The complexity of halfspace area queries, Discrete & Computational Geometry 30 (4) (2003) 639–648.

[17] A. Melkman, J. ORourke, On polygonal chain approximation, in: G.T. Toussaint (Ed.), Computational Morphology, North-Holland, 1988, pp. 87–95.

[18] M. Visvalingam, J.D. Whyatt, Line generalisation by repeated elimination of points, Cartographic Journal 30 (1) (1993) 46–51.

[19] H. Veregin, Line simplification, geometric distortion, and positional error, Cartographica 36 (1) (1999) 25–39.