On Optimal Polyline Simplification using the Hausdorff and Fréchet Distance

Marc van Kreveld[1], **Maarten Löffler**[1], **and Lionov Wiratma**[1,2]

1 **Dept. of Inform. and Computing Sciences, Utrecht University, the Netherlands**
`[m.j.vankreveld|m.loffler|l.wiratma]@uu.nl`

2 **Dept. of Informatics, Parahyangan Catholic University, Indonesia**
`lionov@unpar.ac.id`

Abstract

We revisit the classical polygonal line simplification problem and study it using the Hausdorff distance and Fréchet distance. We use these measures in its pure form, namely: for a given $\varepsilon > 0$, choose a minimum size subsequence of the vertices of the input such that the Hausdorff or Fréchet distance between the input and output polylines is at most ε.

We analyze how the Douglas-Peucker and Imai-Iri simplification algorithms perform compared to the optimum possible. We prove that it is NP-hard to compute the optimal simplification under (undirected) Hausdorff distance. Under the Fréchet distance, the optimal simplification of a polygonal line consisting of n vertices can be computed in $O(kn^5)$ time and $O(kn^2)$ space, where k is the output complexity of the simplification.

1 Introduction

Line simplification (a.k.a. polygonal approximation) is one of the oldest and best studied applied topics in computational geometry. A simplification should have a similar shape as the input, and hence we need a similarity or distance measure to specify when a simplification is acceptable. The *Hausdorff distance* and the *Fréchet distance* are probably the best known distance measures used for shape similarity in computational geometry.

Among the well-known simplification algorithms, the ones by Douglas and Peucker [4] and by Imai and Iri [7] are frequently implemented and cited. For a given constant $\varepsilon > 0$, both algorithms start with a polygonal line (henceforth *polyline*) as an input, specified by a sequence of points $\langle p_1, \ldots, p_n \rangle$, and compute a subsequence starting with p_1 and ending with p_n, representing a simplified polyline which is within a distance of ε from the input.

The Douglas-Peucker algorithm [4] is a simple procedure that starts with a simplification $\overline{p_1 p_n}$, determines the furthest vertex p_k, and if it is further than ε, adds p_k to the simplification. This gives two subproblems with $\overline{p_1 p_k}$ and $\overline{p_k p_n}$ that are solved recursively in the same way and then merged. Hershberger and Snoeyink [6] provide an $O(n \log n)$ time implementation of this algorithm. The Imai-Iri algorithm [7] takes a different approach. It determines for every *link* $\overline{p_i p_j}$ $(i < j)$ if it lies within distance ε from the vertices $p_{i+1}, \ldots, p_{j-1}$ and if so, deems it *valid*. The graph G with all vertices $p_1, \ldots, p_n$ as nodes and all valid links as edges can then be constructed, and a minimum link path from p_1 to p_n represents an optimal simplification. By the implementation of Chan and Chin [3], this algorithm runs in $O(n^2)$ time.

The Imai-Iri algorithm is considered an optimal line simplification algorithm, because it minimizes the number of vertices in the output. It also guarantees the Hausdorff distance between the input and the simplification of at most ε. However, the simplification is not optimal for the Hausdorff distance, because there are simple examples where a simplification with fewer vertices have the Hausdorff distance at most ε to the input. This comes from the fact that the algorithm uses the Hausdorff distance *between a link* $\overline{p_i p_j}$ *and the sub-polyline* $\langle p_i, \ldots, p_j \rangle$, and not an overall Hausdorff distance.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.

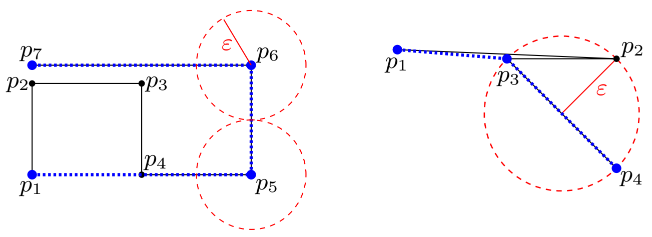

Figure 1 The Douglas-Peucker and Imai-Iri algorithms do not simplify the inputs for the Hausdorff distance (left) or the Fréchet distance (right). The optimal simplifications are shown dotted in blue.

Note that we can easily adapt the Imai-Iri algorithm to guarantee the Fréchet distance of at most ε: we deem a link $\overline{p_i p_j}$ valid if its Fréchet distance to the sub-polyline $\langle p_i, \ldots, p_j \rangle$ is at most ε [1]. This simple variation of the Imai-Iri algorithm does not yield the optimal simplification within the Fréchet distance of ε, because *it requires us to match a vertex p_i in the input to the vertex p_i in the output in the parametrizations, if p_i is used in the output.* This restriction on the parametrizations limits the simplification in undesirable ways.

The examples in Figure 1 show that under the Hausdorff distance (left) and Fréchet distance (right) the Douglas-Peucker and Imai-Iri simplifications are both equal to P itself and may use more vertices than an optimal simplification using these measures.

The discussion begs the following questions: How much worse do the known algorithms and their variations perform in theory, when compared to the optimal Hausdorff and Fréchet simplifications? What if the optimal Hausdorff and Fréchet simplifications use a smaller value than ε? How efficiently can the optimal Hausdorff simplification and the optimal Fréchet simplification be computed (when using the input vertices)?

Organization and results. In Section 2 we show that the optimal simplification has fewer vertices than the Imai-Iri output, both under the Hausdorff and the Fréchet distance (we ignore the Douglas-Peucker method from now on because it never yields fewer vertices than the Imai-Iri method). In particular, we analyze how much worse the output of the Imai-Iri algorithm can be for the two measures. In Section 3 we show that the optimal simplification under the undirected Hausdorff distance is NP-hard to compute. In Section 4 we show that simplification can be done optimally in polynomial time for the Fréchet distance.

2 Approximation Quality of Imai-Iri Simplification

We denote the simplification by the Imai-Iri algorithm under the Hausdorff distance as $II_H(P, \varepsilon)$, and will leave out the arguments P and/or ε if they are understood. We refer to the simplification from the adapted Imai-Iri algorithm using the Fréchet distance as $II_F(P, \varepsilon)$. We denote the optimal simplification using the Hausdorff distance by $OPT_H(P, \varepsilon)$, and using the Fréchet distance by $OPT_F(P, \varepsilon)$. The example in Figure 1 shows that to let II_H use as few vertices as OPT_H, we must use 2ε instead of ε when the example is stretched horizontally. For the Fréchet distance, the enlargement factor needed in the example approaches $\sqrt{2}$ if we put p_1 far to the left. In this section we analyze how the approximation enlargement factor relates to the number of vertices in the Imai-Iri simplification and the optimal ones.

Hausdorff Distance To show that II_H may use many more vertices than OPT_H, even if we enlarge ε, we give a construction where this occurs in Figure 2 that applies for both the directed and undirected Hausdorff distance.

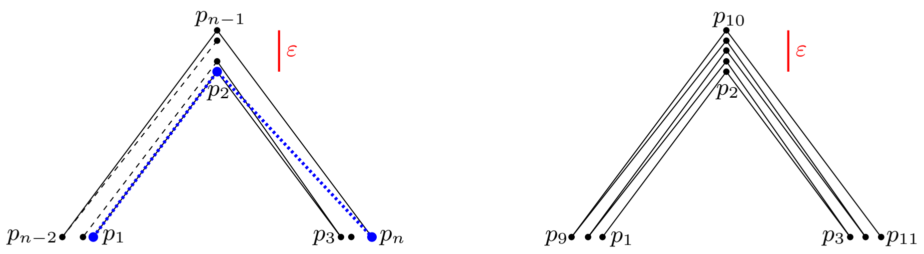

Figure 2 The Imai-Iri algorithm may not be able to simplify $\langle p_1, \ldots, p_n \rangle$ at all. The optimal Hausdorff simplification (dotted, blue) has three vertices. Right, an example input with 11 vertices.

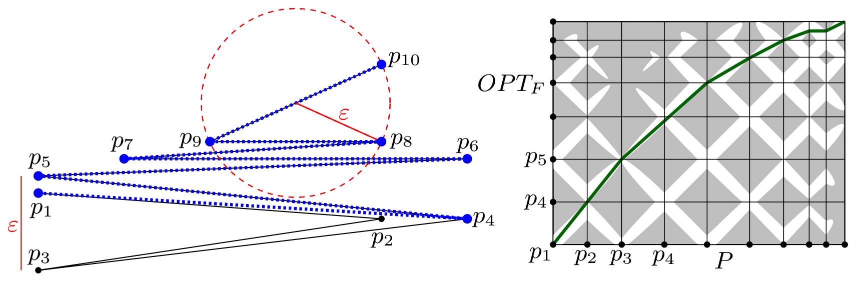

Figure 3 The optimal simplification can skip p_2 and p_3; in the parametrizations witnessing the Fréchet distance, OPT_F "stays two vertices behind" on the input until the end. Right, the free space diagram of P and OPT_F.

An optimal simplification is $\langle p_1, p_i, p_n \rangle$ where i is any even number between 1 and n. Since the only valid links are the ones connecting two consecutive vertices of P, II_H is P itself. If the triangle is large enough with respect to ε, this remains true even if we give the Imai-Iri algorithm a much larger error threshold than ε.

▶ **Theorem 2.1.** *For any $c > 1$, there exists a polyline P with n vertices and an $\varepsilon > 0$ such that $II_H(P, c\varepsilon)$ has n vertices and $OPT_H(P, \varepsilon)$ has 3 vertices.*

Fréchet Distance We give another input polyline P in Figure 3 to show that II_F does not approximate OPT_F even if II_F is allowed to use ε that is larger by a constant factor. Our main construction has ten vertices placed in such a way that II_F has all ten vertices, while OPT_F has only eight of them, see Figures 3. We can append multiple copies of this construction together with a suitable connection in between. We obtain:

▶ **Theorem 2.2.** *There exist constants $c_1 > 1$, $c_2 > 1$, a polyline P with n vertices, and an $\varepsilon > 0$ such that $|II_F(P, c_1\varepsilon)| > c_2|OPT_F(P, \varepsilon)|$.*

By a result of Agarwal et al. [1], we know that the theorem is not true for $c_1 \geq 4$.

3 Algorithmic Complexity of Optimal Simplification using the Hausdorff Distance

The results in the previous section lead us to the following question: *Is it possible to compute the optimal Hausdorff or Fréchet simplification in polynomial time?*

We first consider the undirected (or bidirectional) Hausdorff distance; that is, we require both the maximum distance from the initial polyline P to the simplified polyline Q and the maximum distance from Q to P to be at most ε.

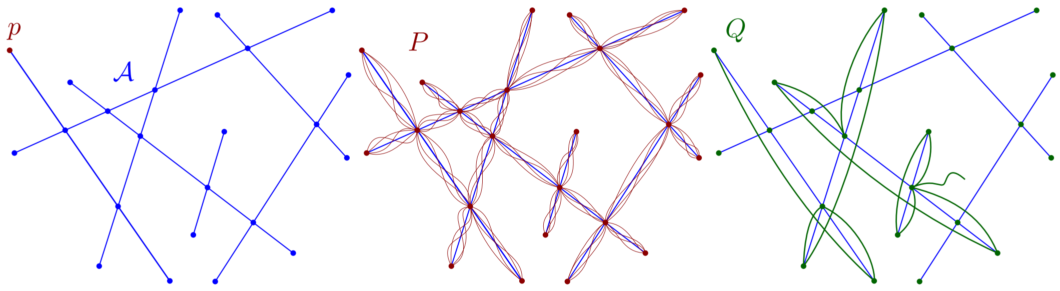

Figure 4 The construction: $\mathcal{A}$ is the arrangement of a set of segments S. We build an input path P that "paints" over S completely, and we are looking for an output path Q that corresponds to a Hamiltonian cycle. In this case, there is no Hamiltonian cycle, and the path gets stuck.

▶ **Theorem 3.1.** *Given a polyline $P = \langle p_1, p_2, \ldots, p_n \rangle$ and a value ε, the problem of computing a minimum length polyline Q defined by a subsequence of the vertices of P such that the undirected Hausdorff distance between P and Q is at most ε is NP-hard.*

Our proof uses a reduction from Hamiltonian cycle in segment intersection graphs. Since deciding if a Hamiltonian cycle exists is NP-complete in planar graphs [5], and planar graphs are included in segment intersections graphs [2], it follows that Hamiltonian cycle in segment intersections graphs is NP-complete. Let S be a set of n line segments in $\mathbb{R}^2$, and assume all intersections are proper (if not, extend the segments slightly). Let G be its intersection graph. Assume that G is connected; otherwise, there is no Hamiltonian cycle in G.

We first construct an initial polyline P as follows (see Figure 4). Let $\mathcal{A}$ be the arrangement of S, let p be some endpoint of a segment in S, and let π be any path on $\mathcal{A}$ that starts and finishes at p and visits all vertices and edges of $\mathcal{A}$. Then P is simply $3n + 1$ copies of π appended to each other. We now set ε to a sufficiently small value. Then, an output polyline Q with Hausdorff distance at most ε to P must also visit all vertices and edges of $\mathcal{A}$, and stay close to $\mathcal{A}$. If ε is sufficiently small, there will be no benefit for Q to ever leave $\mathcal{A}$.

▶ **Lemma 3.2.** *A solution Q of length $3n + 1$ exists iff G admits a Hamiltonian cycle.*

Proof. Clearly, any simplification Q will need to visit the $2n$ endpoints of the segments in S, and—since it starts and ends at the same point p—will need to have length at least $2n + 1$. Furthermore, Q will need to have at least two internal vertices on every segment $s \in S$: once to enter and once to leave the segment (we cannot enter or leave a segment at an endpoint since all intersections are proper intersections). This means the theoretical minimum number of vertices possible for Q is $3n + 1$.

Now, if G admits a Hamiltonian cycle, it is easy to construct a simplification with $3n + 1$ vertices. We start at p, an endpoint of the segment s_1, and collect the other endpoint. Then we follow the Hamiltonian cycle to segment s_2; by definition $s_1 s_2$ is an edge in G so their corresponding segments intersect, and we use the intersection point to leave s_1 and enter s_2. We proceed in this fashion until we reach s_n, which intersects s_1, and finally return to p.

On the other hand, any solution with $3n + 1$ vertices must necessarily be of this form and therefore imply a Hamiltonian cycle: in order to have only 3 vertices per segment the vertex at which we leave s_1 must coincide with the vertex at which we enter some other segment, which we call s_2, and we must continue until we visited all segments and return to p. ◀

For completeness, we also state the results for simplification using the *directed* Hausdorff distance, in both directions. If we require the distance from the input to the simplification to be at most ε, then an optimal simplification using the (directed) Hausdorff distance is NP-hard to compute. However, if we require the distance from the simplification to the input to be at most ε, an optimal simplification can be computed in polynomial time. We give the proofs in the full paper.

4 Algorithmic Complexity of Optimal Simplification using the Fréchet Distance

In this section, we show that for a given polyline $P = \langle p_1, p_2, ..., p_n \rangle$ and an error ε, the optimal simplification $Q = OPT_F(P, \varepsilon)$ can be computed in polynomial time using a dynamic programming approach. First, we define π, a parameterization of P as a continuous mapping: $\pi : [0, 1] \rightarrow \mathbb{R}^2$ where $\pi(0) = p_1$ and $\pi(1) = p_n$. We also write $P[s, t]$ for $0 \leq s \leq t \leq 1$ to be the subcurve of P starting at $\pi(s)$ and ending at $\pi(t)$, also writing $P[t] = P[0, t]$ for short.

For the dynamic programming approach to work, we might imagine to store, for each vertex p_i and value k, the point $\pi(\alpha)$ which is the farthest along P such that a simplification of $\langle p_1, ..., p_i \rangle$ using k links has Fréchet distance at most ε to $P[\alpha]$. However, this is not sufficient to ensure that we find an optimal solution (see the full paper for details). Instead, we argue that if we maintain the set of *all* points at P that can be "reached" by a simplification up to each vertex, then we can make dynamic programming work. We now make this precise and argue that the complexity of these sets of reachable points is never worse than linear.

We say that a point $\pi(t)$ can be *reached* by a (k, i)-simplification for $0 \leq k < i \leq n$ if there exists a simplification of $\langle p_1, \ldots, p_i \rangle$ using k links which has Fréchet distance at most ε to $P[t]$. We let $\rho(k, i, t) = \texttt{true}$ in this case, and `false` otherwise. With slight abuse of notation we also say that t itself is reachable, and that an interval I is reachable if all $t \in I$ are reachable (by a (k, i)-simplification).

▶ **Observation 4.1.** *A point $\pi(t)$ can be reached by a (k, i)-simplification if and only if there exist a $0 < h < i$ and a $0 \leq s \leq t$ such that $\pi(s)$ can be reached by a $(k-1, h)$-simplification and the segment $\overline{p_h p_i}$ has Fréchet distance at most ε to $P[s, t]$.*

Proof. Follows directly from the definition of the Fréchet distance. ◀

Observation 4.1 immediately suggests a dynamic programming algorithm: for every k and i we store a subdivision of $[0, 1]$ into intervals where ρ is true and intervals where ρ is false, and we calculate them for increasing values of k. We simply iterate over all possible values of h, calculate which intervals can be reached using a simplification via h, and then take the union over all those intervals. For this, the only unclear part is how to calculate these intervals. We argue that, for any given k and i, there are at most $n - 1$ reachable intervals on $[0, 1]$, each contained in an edge of P. Indeed, every (k, i)-reachable point $\pi(t)$ must have distance at most ε to p_i, and since the edge e of P that $\pi(t)$ lies on intersects the disk of radius ε centered at p_i in a line segment, every point on this segment is also (k, i)-reachable. We denote the farthest point on e which is (k, i)-reachable by $\hat{t}$.

Furthermore, we argue that for each edge of P, we only need to take the farthest reachable point into account during our dynamic programming algorithm.

▶ **Lemma 4.2.** *If k, h, i, s, and t exist such that $\rho(k-1, h, s) = \rho(k, i, t) = \texttt{true}$, and $\overline{p_h p_i}$ has Fréchet distance $\leq \varepsilon$ to $P[s, t]$, then $\overline{p_h p_i}$ also has Fréchet distance $\leq \varepsilon$ to $P[\hat{s}, \hat{t}]$.*

Proof. By the above argument, $P[s, \hat{s}]$ is a line segment that lies completely within distance ε from p_h, and $P[t, \hat{t}]$ is a line segment that lies completely within distance ε from p_i.

We are given that the Fréchet distance between $\overline{p_h p_i}$ and $P[s,t]$ is at most ε; this means a mapping $f : [s,t] \to \overline{p_h p_i}$ exists such that $|\pi(x) - f(x)| \le \varepsilon$. Let $q = f(s')$. Then $|p_h - \pi(\hat{s})| \le \varepsilon$ and $|q - \pi(\hat{s})| \le \varepsilon$, so the line segment $\overline{p_h q}$ lies fully within distance ε from $\hat{s}$.

Therefore, we can define a new ε-Fréchet mapping between $P[\hat{s},\hat{t}]$ and $\overline{p_h p_i}$ which maps $\hat{s}$ to the segment $\overline{p_h q}$, the curve $P[\hat{s},t]$ to the segment $\overline{q p_i}$ (following the mapping given by f), and the segment $\overline{\pi(t)\pi(\hat{t})}$ to the point p_i. ◀

Now, we can compute the optimal simplification by maintaining a $k \times n \times n$ table storing $\rho(k,i,\hat{t})$, and calculate each value by looking up n^2 values for the previous value of k, and testing in linear time for each combination whether the Fréchet distance between the new link and $P[\hat{s},\hat{t}]$ is within ε or not.

▶ **Theorem 4.3.** *Given a polyline $P = \langle p_1, ..., p_n \rangle$ and a value ε, we can compute the optimal polyline simplification of P that has Fréchet distance at most ε to P in $O(kn^5)$ time and $O(kn^2)$ space, where k is the output complexity of the optimal simplification.*

5 Future Work

A number of challenging open problems remain. First, we would like to know whether the problem of computing an optimal simplification using the Hausdorff distance remains NP-hard when the simplification may not have self-intersections. Second, we are interested in the computational status of the optimal simplification when the simplification need not use the vertices of the input. Finally, we may consider optimal polyline simplifications using the weak Fréchet distance.

Acknowledgements. M.v.K is supported by the Netherlands Organisation for Scientific Research (NWO). M.L. is supported by the Netherlands Organisation for Scientific Research (NWO). L.W. is supported by the Ministry of Research, Technology and Higher Education of Indonesia (138.41/E4.4/2015).

References

1 Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42(3):203–219, 2005.

2 Jérémie Chalopin and Daniel Gonçalves. Every planar graph is the intersection graph of segments in the plane: Extended abstract. In *Proceedings 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 631–638, New York, NY, USA, 2009. ACM.

3 W.S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *International Journal of Computational Geometry & Applications*, 06(01):59–77, 1996.

4 David Douglas and Thomas Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973.

5 M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976.

6 John Hershberger and Jack Snoeyink. An $O(n \log n)$ implementation of the Douglas-Peucker algorithm for line simplification. In *Proceedings 10th Annual Symposium on Computational Geometry*, SCG '94, pages 383–384, New York, NY, USA, 1994. ACM.

7 Hiroshi Imai and Masao Iri. Polygonal approximations of a curve - formulations and algorithms. In Godfried T. Toussaint, editor, *Computational Morphology: A Computational Geometric Approach to the Analysis of Form.* North-Holland, Amsterdam, 1988.