曲線の最小 Fréchet 距離近似に関するアルゴリズムの実装

結城 匡人　　徳山 豪
東北大学大学院　情報科学研究科

概要

地図製作やコンピュータグラフィックスの分野では、曲線を表現するのに、その上のサンプル点の列を使用するが、データ量を圧縮するために、曲線の単純化が必要となる。多角形曲線を単純化する近似アルゴリズムとして、Fréchet distance を利用した FrechetSimp 法を使用し、二分探索法を用いて、$O(n \log n)$ で計算できる。FrechetSimp 法を実装し、複雑に折れ曲がった曲線でも、その特徴を残したまま単純化できることを紹介する。

Implementation of an approximation algorithm for minimum Fréchet distance of polygonal curve

Masato Yuki　　Takeshi Tokuyama
GSIS, Tohoku University

Abstract

FrechetSimp is an approximation algorithm for that applies the concept of Fréchet distance. The algorithm is based on binary search and its complexing is $O(n \log n)$. We implement this algorithm and report some experimental results.

1 はじめに

情報をビジュアルに表現するとき、データの削減を必要とする場合がしばしばある。例えば、地図製作やコンピュータグラフィックスの分野では、データを圧縮して地形の特徴を残したまま単純化する必要があり、その際に用いられる手段として、曲線の単純化がある。

ここで、曲線単純化問題がどのようなものであるかについて説明する。平面上において、頂点系列が、$p_1, p_2, \ldots, p_n$ である多角形曲線を P とする。ここで、n は頂点の数とする。これに対し、頂点系列が $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ である多角形曲線を P' とし、$P' \subseteq P$、$1 = i_1 < \cdots < i_k = n$ であるとする。このとき、P' は P を単純化すると呼ぶ。

単純化の質の尺度を与えよう。頂点 $p_i, p_j \in P$ を結んでできる線分 p_ip_j に対し、ある誤差測定法 M を用いたときの p_ip_j と P の誤差を $\delta_M(p_ip_j, P)$ とする。これを用いて、P' と P の誤差 $\delta_M(P', P)$ は以下の式で定義される。

$$\delta_M(P', P) = \max_{1 \le j < k} \delta_M(p_{i_j}p_{i_j+1}, P)$$

$\delta_M(P', P) \le \varepsilon$ であるとき、P' を P の ε-*simplification* と呼ぶ。ここで、ε は非負実数のしきい値である。このような P' のうち、頂点数が最小となるものを計算することを曲線単純化問題といい、そのときの頂点数を $\kappa_M(P, \varepsilon)$ とする。

ここでは、誤差測定法として、Fréchet 誤差測定法を用いる。その際に測定するのが、Fréchet distance である。Fréchet distance は、単調な曲線だけでなく、複雑に折れ曲がった曲線にも対応でき、曲線の特徴を失うことなくデータの圧縮が可能となる。本論文では、文献［APMW 02］に基づき、Fréchet 誤差測定法を用いた場合における、曲線単純化の近似アルゴリズムを紹介する。

このアルゴリズムの計算時間は、任意の曲線で $O(n \log n)$ である。また、単純化された曲線の頂点数は、最適な $\varepsilon/2$-simplification の頂点数以下であ

り、単純化された頂点数を保証する。アルゴリズムを実装し，文献［APMW 02］の手法の実効性を確認した。

2 Fréchet 誤差測定法による曲線単純化

Fréchet 誤差測定法による曲線単純化の最適解を求めるためにかかる計算時間は、$O(n^3)$ であることが知られている ([II88])。ここでは、$O(n \log n)$ で計算する近似アルゴリズムを紹介する。

2.1 Fréchet Distance

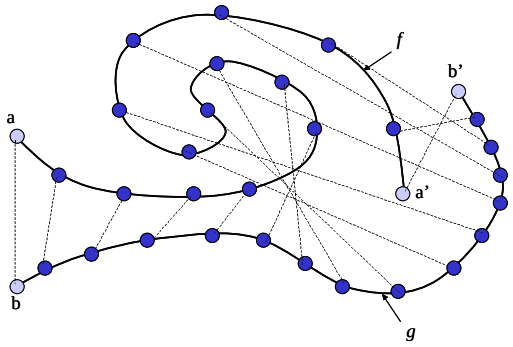


図 1: f,g 上を動く 2 点間の距離

2 つの曲線 $f : [a, a']$ と $g : [b, b']$ を考える。α, β を連続かつ増加関数とし、パラメータを $t \in [0, 1]$ とすると、$\alpha(t), \beta(t)$ はそれぞれ、f, g 上を動く。ただし、$\alpha(0) = a, \alpha(1) = a', \beta(0) = b, \beta(1) = b'$ とする。これらを表したのが、図 1 である。

f と g の間の Fréchet distance である $F_D(f, g)$ は以下の式で定義される。

$$F_D(f,g) = \inf_{\substack{\alpha:[0,1]\to[a,a'] \\ \beta:[0,1]\to[b,b']}} \max_{t\in[0,1]} \{d(f(\alpha(t)), g(\beta(t)))\}$$

人間が犬を散歩に連れて行く光景を思い出してほしい。いま、犬を鎖につけて散歩すると考える。人間は f 上の道を a から a' まで戻らずに歩き、犬は g 上の道を b から b' まで戻らずに歩くとし、道をはみ出すことはないとする。また、人間は鎖から手を離さないとする。そのときに、鎖が伸びきらないように最後まで歩くことができるような、最小の鎖の長さが、f と g の間の Fréchet distance に相当する。

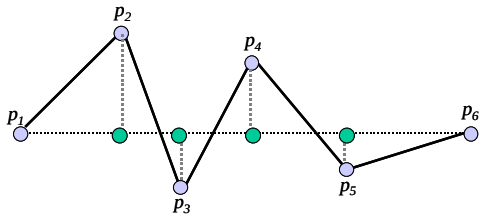


図 2: x-単調な折れ線と水平線分の間の Fréchet distance

線分と線分方向に単調な折れ線 P の場合を考えてみる。図 2 のように f を P、g を線分 p_1p_6 とすると、Fréchet distance は、P の各頂点から線分 p_1p_6 への最短距離の中で、最大のものとなる。ここでは、p_2 から線分 p_1p_6 へ下ろした垂線が、Fréchet distance となる。

従って、片方が線分で他方が単調である場合は、通常の $Hausdorff$ 距離と同一である。

2.2 Fréchet 誤差測定法

2 つの頂点 $p_i, p_j \in P, i < j$ に関して、線分 p_ip_j と P の Fréchet 誤差は以下の式で定義される。

$$\delta_F(p_ip_j, P) = F_D(\pi(p_i, p_j), p_ip_j)$$

$\pi(p_i, p_j)$ は、p_i から p_j までの P の部分曲線である。図 3 のような破線と実線からなる P を考えると、$\pi(p_i, p_j)$ は、実線部分がそれに相当する。

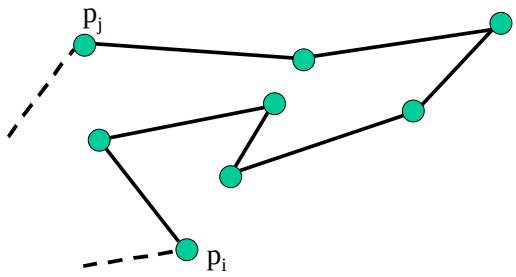


図 3: $\pi(p_i, p_j)$

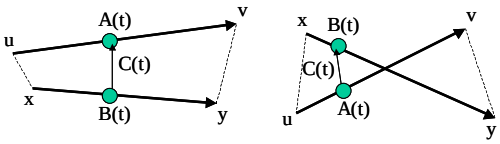


図 4: 有向辺 uv,xy

ここで、以下の補題が成立する。

補題 2.1　2 つの有向辺である uv と xy に関して、以下の補題が成立する。

$$F_D(uv, xy) = \max\{d(u, x), d(v, y)\}$$

証明　$\delta = \max\{d(u, x), d(v, y)\}$ とおく。まず、F_D は定義式より、u と x、v と y が対応する。$d(u, x)$ と $d(v, y)$ は F_D の値の候補となるため、$F_D(uv, xy) \geq \delta$ が成立する。

一方、$t \in [0, 1]$ において、$A(t) = u + t(v - u), B(t) = x + t(y - x)$ とし、$C(t) = A(t) - B(t)$ とすると、$C(t) = (1 - t)(u - x) + t(v - y)$ となる。ここで、

$$\begin{aligned} ||C(t)|| &= ||(1 - t)(u - x) + t(v - y)|| \\ &\leq (1 - t)||u - x|| + t||v - y|| \\ &\leq (1 - t) \cdot \delta + t \cdot \delta = \delta \end{aligned}$$

したがって、$F_D(uv, xy) \leq ||C(t)|| \leq \delta$ となる。以上より、$F_D(uv, xy) = \delta$　□

2.3　FrechetSimp 法

Fréchet 誤差測定法のもとで P の ε-simplification である P' を計算する近似アルゴリズムである FrechetSimp 法は、以下の手順で行われる。

アルゴリズム

- 入力：$P = \langle p_1, p_2, \ldots, p_n \rangle$ と誤差 $\varepsilon(> 0)$
- 出力：頂点系列 P'

1. $P' = \langle p_{i_1} = p_1 \rangle$ とし、$k = 2$ とする。
2. P' に含まれる点のなかで、添え字が最大のものを p_{i_j} とすると、$\delta_F(p_{i_j} p_k, P) \leq \varepsilon, \delta_F(p_{i_j} p_{k+1}, P) > \varepsilon$ となる k を見つけ、p_k を P' に含める。
3. $k > n$ となるまで手順 2 を繰り返し、最後に p_n を P' に含める。

2.4　FrechetSimp 法の計算時間

FrechetSimp 法の計算時間が、$O(n \log n)$ であることを示す。それを示すために、距離判定問題と二分探索法の 2 つについて述べる。

2.4.1　距離判定問題

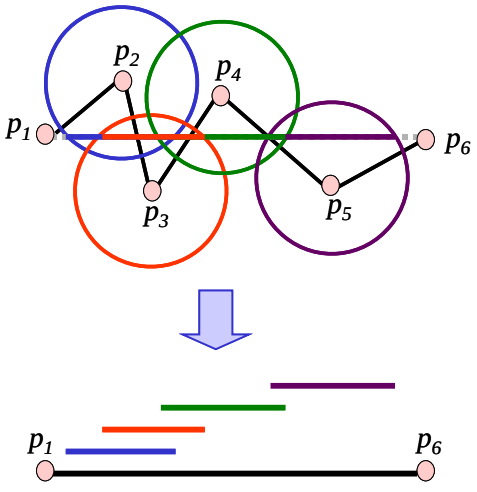


図 5: 各頂点から ε 以内にある区間の作成

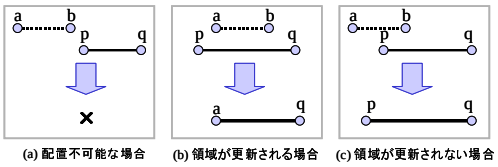


図 6: 現在の区間と考慮する区間との関係

手順 2 で $\delta_F(p_{i_j}p_k, P)$ が ε より大きいかどうかの判定に関して、以下のように行う。$p_{i_j}p_{i_j+1},\ldots,p_{k-1}p_k$ であるように、$\pi(p_{i_j}, p_k)$ を線分に分割して考える。線分 $p_{i_j}p_k$ に関して、$\pi(p_{i_j}, p_k)$ 上の $p_{i_j}, p_{i_j+1},\ldots,p_k$ と対応する点 $p'_{i_j}, p'_{i_j+1},\ldots,p'_k$ を、それぞれ決める。補題 2.1 より、$F_D(p_{i_j}p_k, \pi(p_{i_j}, p_k))$ は、$d(p_{i_j}, p'_{i_j}),\ldots,d(p_k, p'_k)$ の最大値となるから、$d(p_{i_j}, p'_{i_j}),\ldots,d(p_k, p'_k)$ の最大値がすべて ε 以下で、かつ、点 $p'_{i_j}, p'_{i_j+1},\ldots,p'_k$ が後戻りしないように配置することができれば、$F_D(p_{i_j}p_k, \pi(p_{i_j}, p_k))$ の値は ε 以下となり、そうでなければ、ε よりも大きくなる。

これらを計算するために、図 5 のように、点 $p_{i_j}, p_{i_j+1},\ldots,p_k$ を中心とする半径 ε の円が含む、線分 $p_{i_j}p_k$ の各区間を求める。この作業は定数時間でできる。この得られた区間内に関して、点 p_{i_j} から順番に点を配置できるかどうか判定する。

まず、初期区間を線分 $p_{i_j}p_k$ の両端点とする。次に、点 p_{i_j+1} がつくる区間に点 p'_{i_j+1} を配置する。この際、図 6 のような関係が考えられる。現在の区間を $[p, q]$、考慮する区間を $[a, b]$ とする。(a) の場合は、順番に点を配置することができない。(b) の場合は、区間を $[a, q]$ に更新する。(c) の場合は、区間は変更されず、そのままである。

この処理を p_k まで繰り返していき、すべての点を順番に配置することができれば、ε より小さい値になると分かる。そうでなければ、ε より大きくなる。したがって、この配置を行うのに、$O(k - i_j)$ の時間がかかる。

関数 d は定数時間で計算できるので、$\delta_F(p_{i_j}p_k, P)$ を計算するのに要する時間は、$O(k - i_j)$ となる。従って、次の補題を得る。

補題 2.2　$\delta_F(p_{i_j}p_k, P)$ が ε より大きいかどうか判定するのにかかる時間は、$O(k - i_j)$ である。

2.4.2　二分探索法

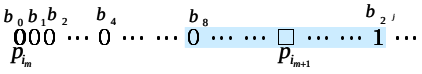


図 7: 最初に b_ρ が 1 になる j

頂点 $p_{i_m} \in P'$ の次に、P' に含まれる $p_{i_{m+k}}$ を計算する方法を考える。これには、二分探索法を用いる。

まず、b_ρ を 1 ビットの値とし、$\delta(p_{i_m}p_{i_{m+\rho}}) > \varepsilon$ ならば 1、そうでなければ 0 とする。b_ρ を計算するのにかかる時間は、補題 2.2 より、$O(\rho)$ となる。

$b_k = 0, b_{k+1} = 1$ となる、連続した 2 ビットの値を探すのだが、最初に探索する範囲を決定する。$b_0 = 0$ とし、$b_1, b_2, b_4, \ldots, b_{2^j}$ と飛び飛びに計算し、最初に値が 1 となる j の値を求め、$b_{2^{j-1}}$ から b_{2^j} を探索の区間に定める。なお、$2^j > n$ となった場合は、探索の終点を頂点が p_n に相当する b_ρ とする。この作業にかかる時間は、$O((i_{m+k} - i_m)\log(i_{m+k} - i_m))$ となる。図 7 はこの作業をイメージしたものだが、数字が書いていない場所は、まだ判定が行われていないことを表している。図 7 では、b_{16} が b_{2^j} に対応している。

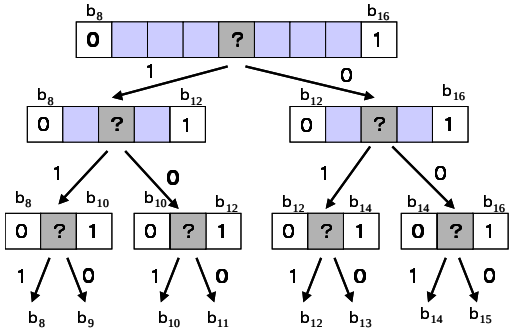


図 8: 二分探索の例

この区間内には、01 の連続した 2 ビットが存在する。これを探すのに二分探索を行う。最初は、

$b_{(2^j+2^{j-1})/2}$ を計算し、その値が 0 ならば、左半分の区間で、1 ならば右半分の区間で同様の処理を行う。この処理を繰り返して、区間の幅が 1 になったら、そのときの位置、または 1 つ後ろの位置が求める k の値となる。図 8 は、図 7 のときに二分探索を行った場合を示している。このときにかかる時間は、二分探索に $O(j-1) = O(\log(i_{m+1} - i_m))$、$b_\rho$ の値の判定に $O(i_{m+1} - i_m)$ かかるので、$O((i_{m+1} - i_m)\log(i_{m+1} - i_m))$ となる。

したがって、FrechetSimp 法に要する計算時間は、$O(n \log n)$ となる。

2.5 FrechetSimp 法による頂点数の上限

FrechetSimp 法によってできる P' について、以下の補題が成立する。

補題 2.3 *FrechetSimp* 法は、$\delta_F(P', P) \leq \varepsilon$ となる P' を計算する。

証明 $P' = \langle p_1 = p_{i_1}, p_{i_2}, \ldots, p_{i_m} = p_n \rangle \in P$ とする。*FrechetSimp* 法より、$\delta_F(p_{i_1}p_{i_2}, \pi(p_{i_1}, p_{i_2})) \leq \varepsilon, \ldots, \delta_F(p_{i_m-1}p_{i_m}, \pi(p_{i_m-1}, p_{i_m})) \leq \varepsilon$ となる。

よって、補題 *2.1* より、$\delta_F(P', P) \leq \varepsilon$ となる。 □

次に、FrechetSimp 法によってできる P' の頂点数の上限について考えるが、その前にそれを考えるために必要な補題を示す。

補題 2.4 P と 2 つの線分 uv, xy に関して、以下の式が成立する。

$$F_D(xy, P) \leq F_D(uv, P) + F_D(xy, uv)$$

証明 $t \in [0,1]$ において、$A(t) = u + t(v - u), B(t) = x + t(y - x)$ とする。すべての $t \in [0,1]$ について、三角不等式より、

$$d(B(t), P(t)) \leq d(A(t), P(t)) + d(B(t), A(t))$$

が成立する。

よって、$F_D(xy, P) \leq F_D(uv, P) + F_D(xy, uv)$ □

補題 2.5 $P = \langle p_1, p_2, \ldots, p_n \rangle$ とする。$l \leq i \leq j \leq m$ なる、i, j, l, m について、以下の式が成立する。

$$\delta_F(p_ip_j, P) \leq 2 \cdot \delta_F(p_lp_m, P)$$

証明 δ'=$\delta_F(p_lp_m, P)$ とする。*Fréchet distance* の定義より、$\delta_F(p_lp_m, P)$=$F_D(p_lp_m, \pi(p_l, p_m))$ であり、$\pi(p_l, p_m)$ 上の p_i, p_j と対応する点をそれぞれ、$p'_i, p'_j \in p_lp_m$ とする。そうすると、$F_D(p'_ip'_j, \pi(p_i, p_j)) \leq \delta'$ が成立する。これに関連して、$d(p_i, p'_i) \leq \delta', d(p_j, p'_j) \leq \delta'$ が成立する。ここで、補題 *2.1* より、$F_D(p_ip_j, p'_ip'_j) = \max\{d(p_i, p'_i), d(p_j, p'_j)\} \leq \delta'$ となる。よって、補題 *2.4* より、

$$\begin{aligned}
\delta_F(p_ip_j, P) &= F_D(p_ip_j, \pi(p_i, p_j)) \\
&\leq F_D(p'_ip'_j, \pi(p_i, p_j)) + F_D(p'_ip'_j, p_ip_j) \\
&\leq F_D(p_ip_j, \pi(p_i, p_j)) + \delta' \\
&\leq \delta' + \delta' = 2\delta'
\end{aligned}$$

したがって、$\delta_F(p_ip_j, P) \leq 2 \cdot \delta_F(p_lp_m, P)$ となる。 □

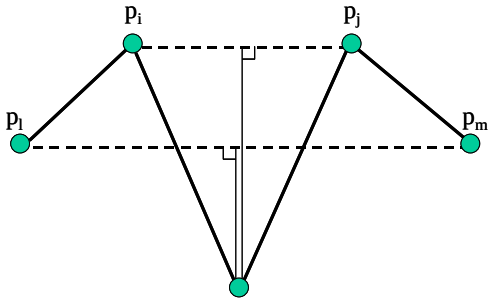


図 9: $\delta_F(p_lp_m, P)$ と $\delta_F(p_ip_j, P)$

FrechetSimp 法によってできる P' の頂点数の上限について、以下の補題が成り立つ。

補題 2.6 *FrechetSimp* 法は、$|P'| \leq \kappa_F(\varepsilon/2, P)$ を満たす P' を作成する。

証明 $P' = \langle p_1 = p_{i_1}, \ldots, p_{i_k} = p_n \rangle$ とし、$Q = \langle p_1 = p_{j_1}, \ldots, p_{j_l} = p_n \rangle$ を、P の *$\varepsilon/2$-simplification* とする。ただし、$1 \leq m \leq k$ に

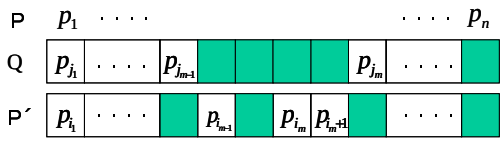


図 10: P' と Q の関係

おいて、$1 \leq i_m \leq n$ とし、$1 \leq m \leq l$ において、$1 \leq j_m \leq n$ とする。すべての m において、$i_m \geq j_m$ であることを示す。

図 *10* のように、$p_{j_{m-1}}, \ldots, p_{j_m}$ の中に $p_{i_{m-1}}, \ldots, p_{i_m}, p_{i_m+1}$ が含まれていると仮定する。P' について、*FrechetSimp* 法より、$\delta_F(p_{i_{m-1}}p_{i_m}, P) \leq \varepsilon, \delta_F(p_{i_{m-1}}p_{i_m+1}, P) > \varepsilon$ となる。Q は P の *ε/2-simplification* であるから、補題 *2.5* より、$\delta_F(p_{i_{m-1}}p_{i_m+1}, P) \leq 2 \cdot \delta_F(p_{j_{m-1}}p_{j_m}, P) \leq 2 \cdot \varepsilon/2 = \varepsilon$ となり、矛盾する。よって、$i_1 = j_1 = 1$ であることから、$i_m \geq j_m$ となることが分かる。

したがって、$j_k \leq i_k = n$ となるから、$k \leq l$ が成立する。以上より、$|P'| \leq \kappa_F(\varepsilon/2, P)$ が成立する。 □

3 実験

Fréchet 誤差測定法による曲線単純化の近似アルゴリズムである FrechetSimp 法を実装した。このアルゴリズムを用いて、様々な P に関して実験を行った。

まず、P を正弦波曲線を 0.05[rad] 刻みに取った頂点集合とする。誤差 ε=0.5 と固定し、頂点数を変えて計算時間に関する実験を行った。

図 11 を見ると、$O(n \log n)$ に近いグラフが得ることができた。このことより、計算時間に関しては $O(n \log n)$ であることが確かめられる。

次に、P を図 12 のようなリサージュ曲線とし、誤差 ϵ の値を変えて実行した。頂点数は 628 個である。また、P を Hilbert 曲線とし、一辺の長さが 2 の正方形に囲まれた、図 15 の場合で、誤差 ϵ の値を変えて実行した。

図 13、図 14 を見ると、ϵ=0.1、ϵ=0.01 の場合

頂点数 n	実行後の頂点数	時間 (sec)
1000	19	0.02
2000	35	0.03
3000	51	0.04
4000	66	0.05
5000	82	0.07
6000	98	0.08
7000	114	0.09
8000	130	0.11
9000	146	0.12
10000	162	0.13

表 1: 単純化したときの頂点数と実行時間

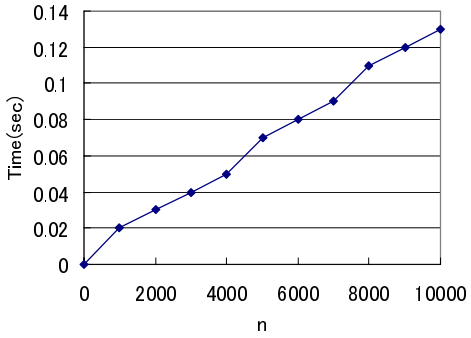


図 11: n と計算時間の関係

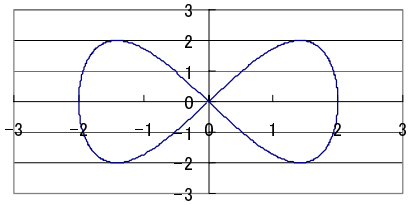


図 12: リサージュ曲線

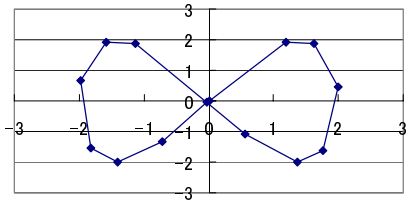


図 13: $\epsilon = 0.1$ の近似曲線

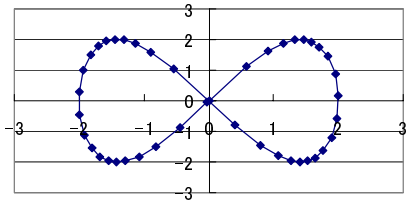


図 14: $\epsilon = 0.01$ の近似曲線

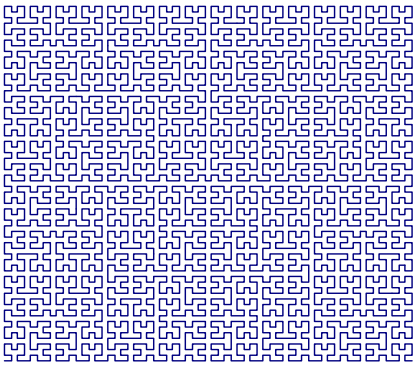
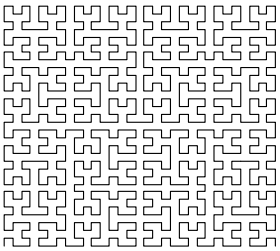


図 15: Hilbert 曲線 H_6



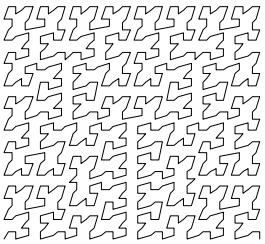
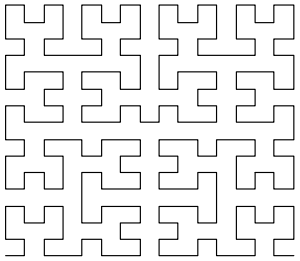


図 16: Hilbert 曲線 H_5 と H_6 の $\epsilon = 0.05$ での近似曲線



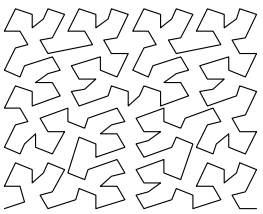


図 17: Hilbert 曲線 H_4 と H_6 の $\epsilon = 0.1$ での近似曲線

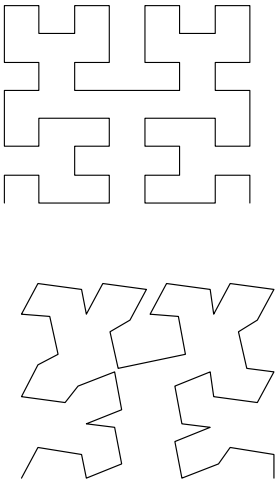


図 18: Hilbert 曲線 H_3 と $\epsilon = 0.25$ の近似曲線

の頂点数は、それぞれ 14 個、42 個であり、いずれも、頂点数を減らして、オリジナルに近い曲線を得ることができている。

また、図 16、図 17、図 18 を見ると、H_6 の近似曲線として、H_5、H_4、H_3 に近い図形が得られていることが分かる。Hilbert 曲線 H_n は、H_{n-1} を利用して再帰的に作られているので、誤差 ϵ で近似すると、$H_k(k \leq n)$ の曲線が得られることが推測される。これらの結果より、複雑な曲線でも、FrechetSimp 法の実効性を確認することができる。

4 まとめ

Fréchet 誤差測定法に二分探索法を導入したことで、計算時間を $O(n \log n)$ に改善し、複雑な曲線に関しても、その実効性を確かめることに成功した。今後の課題としては、データ構造を活用することで、より高速なアルゴリズムを構築すること、より良い誤差測定法を提案することが挙げられる。

参考文献

[APMW 02] Pankaj Agarwal, Sariel Har-Peled, Nabil H.Mustafa, Yusu Wang. Near-Linear Time Approximation Algorithms for Curve Simplification in Two and Three dimensions. *Proc. of the 10th European Symposium on Algorithms (ESA'02)*,2002.

[II 88] H.Imai and M.Iri. Polygonal approximations of a curve-formulations and algorithms. In G.T.Toussaint, editor, *Computational Morphology*, pages 71-86. North-Holland, Amsterdam, Netherlands, 1988.

[God 91] M.Godau. A natural metric for curves: Computing the distance for polygonal chains and approximation algorithms. In proc. *of the 8th Annual Symposium on Theoretical Aspects of Computer Science*, pages 127-136, 1991.

[HS 94] J.Hershberger and J.Snoeyink. An $O(n \log n)$ implementation of the Douglas-Peucker algorithm for line simplification. In Proc. *10th Annual ACM Symposium on Computational Geometry*, pages 383-384, 1994.