Optimal Approximation of Monotone Curves on a Grid*
(Extended Abstract)

Tetsuo Asano[1] Naoki Katoh[2] Elena Lodi[3] Thomas Roos[4]

[1] Dept. of Applied Electronics, Osaka Electro-Communication Univ., Neyagawa, Osaka 572, Japan (asano@djinni.osakac.ac.jp)

[2] Kobe University of Commerce, Kobe, Japan (naoki@kobeuc.ac.jp)

[3] University of Siena, Siena, Italy (lodi@di.unipi.it)

[4] Theoretische Informatik, ETH Zentrum, CH-8092 Zürich, Switzerland (roos@inf.ethz.ch; Fax: +41 1 632 1172)

Abstract. This paper presents efficient algorithms for approximating a curve by a polygonal chain with vertices on a grid. We are given an x-monotone curve $y = f(x)$ consisting of N pieces on some $m \times n$ grid. Our goal is to compute an approximation of the given curve by an x-monotone polygonal chain whose vertices are points of the grid. For that, we discuss several optimization criteria such as minimizing the area of the region bounded by the given curve and the polygonal chain.

Our approach is based on a reduction to the computation of minimum-cost paths. We present an $O(N + n^2)$ time and $O(N + n)$ space algorithm for computing an optimal x-monotone polygonal chain. If we add the restriction that the horizontal lengths of the line segments used for the approximation is bounded by k, the time complexity can further be reduced to $O(N + kn)$. In both cases, the time complexity does not depend on the range m of the function. Applications of this problem can be found, e.g., in the area of computer graphics.

1 Introduction

The approximation of complex geometric objects by simpler ones that capture the relevant features of the originals is a very important problem which appears in many areas of computer science since more than 20 years. The real need for simplification can be elegantly motivated by line simplification/generalization in geography where one often deals with different levels of abstraction. Since the early works by Douglas/Peucker [3] and Ramer [18], many different approximation strategies have been studied, some of which are even $\mathcal{NP}$-hard [6].

In the area of cartography & GIS, line simplification methods are often based on the detection of critical points [2, 16, 21]; however, they always have to "incorporate information about the geometric structure of geographic phenomena" [12, 14] meaning that a good approximation can only obtained with high-level geographic information. A statistical approach describing the benefits of geometric simplification in geography can be found in [15]. Similar techniques appear in computer graphics when approximating digitized images by polygons [11, 19, 20]. In computational geometry, Imai and Iri [8, 9, 10] and Natarajan [17] gave linear time algorithms for the problem of approximating a piecewise linear function with a minimum number of line segments according to a given tolerance ε. Recent surveys of approximation techniques can be found in [4, 5, 6, 7].

In this paper, we study a new variant of this problem: the optimal approximation of an x-monotone planar curve on some $m \times n$ grid by an x-monotone polygonal chain of line segments. Notice that any planar curve can be decomposed into x-monotone pieces. We measure the quality of the approximation with respect to the area of the region bounded by the curve and the polygonal chain, among other

* Part of this work was done while the fourth author visited the Osaka Electro-Communication University and while the first author was with the University of Siena. The first author acknowledges the partial support of the Grant for Promotion of International Joint Work of Osaka Electro-Communication University and of the Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan. The fourth author acknowledges the partial support of this work by the Swiss National Science Foundation (SNF) under grant 21-39328.93.

approximation schemes. The existing straight-forward algorithm for computing an approximation simply computes points by rounding the intersections between the curve and the grid. The quality of this method, however, is not always satisfying. Actually, there are several imaginable approximation criteria. Nevertheless, in the description of our approach, we mainly focus our attention to the so-called squared-area measure, although our algorithms apply to other approximation criteria, as well.

We solve this problem by reduction to minimum-cost paths in some networks. Our first (simple) algorithm computes an optimal approximation in $O(N + n^2m^2)$ time and $O(N + n + m)$ space. Our second and more sophisticated approach runs in $O(N + n^2)$ time and $O(N + n)$ space. By adding the restriction that the horizontal length of each approximating line segment is bounded by k, the time complexity can further be reduced to $O(N+kn)$. Notice that in the latter, the time and space complexity does not depend on the range m of the function.

A similar problem, the approximation of a given polygonal chain by another simpler chain has been studied by Imai and Iri [9, 10]. Their algorithm runs in optimal $O(n)$ time.

2 Preliminaries

At the beginning, we are given a continuous function

$$f : [0, n-1] \to [0, m-1], \quad x \mapsto f(x)$$

on the domain $D := [0, n-1] \times [0, m-1]$ which is covered by a regular grid

$$G := \{(x, y) \mid x = 0, \ldots, n-1 \text{ and } y = 0, \ldots, m-1\}$$

We only assume that some basic computations (e.g. the integral) depending on the measure of approximation are available in constant time. Of course, any function f is x-monotone; here x-monotonicity means that the intersection of any vertical line with f is connected.

Our goal is to approximate (in a certain way) this given function by an x-monotone polygonal chain under the restriction that the vertices of the chain have to be taken from the grid G (see Figure 1). Thus, the output of the algorithm will be a chain $\mathcal{P}$ of length l:

$$\mathcal{P} = (P_0, \ldots, P_l)$$

with $P_i = (\bar{x}_i, \bar{y}_i) \in G$ for all $i = 0, \ldots, l$.

As $\mathcal{P}$ is x-monotone, the x-coordinates of the vertices of the chain form a monotonously increasing sequence: $0 = \bar{x}_0 \leq \ldots \leq \bar{x}_l = n-1$ that spans the interval $[0, n-1]$.

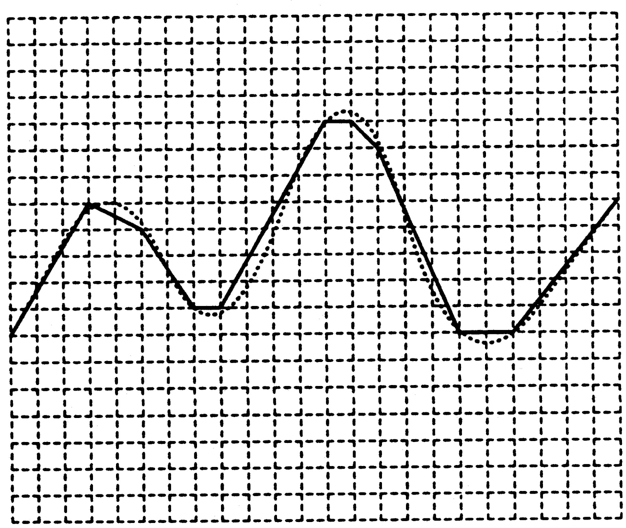

Fig. 1. Approximating an x-monotone planar curve by a polygonal chain.

There are several imaginable optimization criteria, among them very intuitive ones such as minimizing the area of the region bounded by the curve f and the polygonal chain $\mathcal{P}$:

$$\min_{\mathcal{P}} \int_{x=0}^{n-1} |f(x) - \mathcal{P}(x)| \, dx$$

or the so-called squared-area measure

$$\min_{\mathcal{P}} \int_{x=0}^{n-1} \Big[f(x) - \mathcal{P}(x)\Big]^2 dx$$

that measures the integral of the squared function difference (not the square of the area, of course). Yet another optimization criterion (among many others) could be the minimization of the maximum vertical distance

$$\min_{\mathcal{P}} \max_{x \in [0, n-1]} |f(x) - \mathcal{P}(x)|$$

In the following investigation, we will focus our attention on the squared-area criterion. Concerning the description of the input function f, we assume that it is specified as a continuous chain of N x-monotone curves $f_0, \ldots, f_{N-1}$ where f_i is defined over an interval $[\tilde{x}_i, \tilde{x}_{i+1}]$, for $i = 0, \ldots, N-1$, and $0 = \tilde{x}_0 < \ldots < \tilde{x}_N = n-1$. At first, in a preprocessing step, we compute all integrals

$$\int_0^i f(x)\, dx, \quad \int_0^i x \cdot f(x)\, dx \quad \text{and} \quad \int_0^i f(x)^2\, dx$$

for each $i = 1, \ldots, n-1$. It is easy to see that these computations can be performed in $O(N+n)$ time by merging the sorted lists $(0, \ldots, n-1)$ and $(\tilde{x}_0, \ldots, \tilde{x}_N)$.

Once these values have been computed, all integrals $\int_i^j f(x)\,dx$, $\int_i^j x \cdot f(x)\,dx$ and $\int_i^j f^2(x)\,dx$ can be calculated by differences in constant time. Notice that in the case of the squared-area criterion, we do not have to compute intersections between the curve f and the polygonal chain, while in the case of minimizing the area, these intersections would be required.

3 Reduction to Minimum-Weight Paths

Our first algorithm for computing an optimally approximating polygonal chain is based on a directed graph defined as follows:

- The points of the grid G define the vertices of the graph.
- Any two vertices $(x_i, y_i), (x_j, y_j) \in G$ with $x_i < x_j$ share a directed edge $\big((x_i, y_i), (x_j, y_j)\big)$ with an associated weight

$$\delta(x_i, y_i, x_j, y_j) := \int_{x_i}^{x_j} \Big[f(x) - a_{ij}x - b_{ij}\Big]^2 dx$$

  where $y = a_{ij}x + b_{ij}$ is the equation for the line passing through the two points, that is,

$$a_{ij} = \frac{y_j - y_i}{x_j - x_i} \quad \text{and} \quad b_{ij} = y_i - a_{ij}x_i$$

- Furthermore, we add a directed edge with zero weight between any pair of vertices on the same vertical line.

This provides a graph with $n \cdot m$ vertices, $\Theta(n^2m^2)$ edges and non-negative edge weights. Now, it is obvious that an optimal x-monotone polygonal chain corresponds to a minimum-weight path starting from vertex $(0,0)$ to vertex $(n-1, 0)$. The special structure of the graph allows the following minimum-weight path algorithm:

[Algorithm 1]
$d[0] = 0$;
for $x = 1$ **to** $n-1$ **do**
  for $y = 0$ **to** $m-1$ **do**
    $v[y] = \infty$;
    for $x' = x-1$ **downto** 0 **do**
      for $y' = 0$ **to** $m-1$ **do**
        evaluate the squared-area difference $\delta(x', y', x, y)$
          for the line segment between (x', y') and (x, y);
        if $d[x'] + \delta(x', y', x, y) < v[y]$
          then $v[y] = d[x'] + \delta(x', y', x, y)$;
  let $d[x]$ be the minimum among all $v[y]$, for $y = 0, 1, \ldots, m-1$;

Theorem 1. *Algorithm 1 computes an optimally approximating x-monotone polygonal chain in $O(N + n^2m^2)$ time and $O(N + m + n)$ space.*

Proof. The correctness and the time complexity of the algorithm is obvious. The space complexity can be achieved if we avoid constructing the entire graph explicitly. Notice that except of the vertices with zero transitions (which can be compacted into one node, each), the graph allows a topological ordering.

Of course, the algorithm can easily be modified such that it returns not only the minimum weight but also a description of the underlying minimum-weight path.

4 An Efficient Algorithm

A significant disadvantage of Algorithm 1 is that its time complexity depends on m, the value range of the input function. As we'll see in this section, this disadvantage can be avoided. Doing this, we present an algorithm which has no such dependency. The most important observation comes from the following problem.

[One-Segment-Approximation Problem]
Given an x-monotone planar curve $y = f(x)$ and two integers x_1 and x_2 ($x_1 < x_2$), find two integers y_1 and y_2 such that the line segment connecting the two points (x_1, y_1) and (x_2, y_2) minimizes the squared-area difference:

$$F(y_1, y_2) = \int_{x_1}^{x_2} \left[f(x) - \left(\frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2 \right) \right]^2 dx$$

Now, $F(y_1, y_2)$ is minimized if y_1 and y_2 simultaneously satisfy the equation $\frac{\partial F}{\partial y_1}(y_1, y_2) = \frac{\partial F}{\partial y_2}(y_1, y_2) = 0$. A straightforward calculation shows that the optimal point (y_1^*, y_2^*) is given by

$$y_1^* = \frac{2}{(x_2 - x_1)^2} \left[(2x_2 + x_1) \int_{x_1}^{x_2} f(x)\, dx - 3 \int_{x_1}^{x_2} x f(x)\, dx \right] \quad \text{and}$$

$$y_2^* = \frac{-2}{(x_2 - x_1)^2} \left[(2x_1 + x_2) \int_{x_1}^{x_2} f(x)\, dx - 3 \int_{x_1}^{x_2} x f(x)\, dx \right]$$

If y_1^* and y_2^* are both integers, we are done. Otherwise, we need to find an integer point $(\hat{y}_1, \hat{y}_2)$ that minimizes $F(y_1, y_2)$. In order to compute this point, we present the following lemma that guarantees that only a constant number of lattice points near the optimal point (y_1^*, y_2^*) are potential candidates for the optimal integer point $(\hat{y}_1, \hat{y}_2)$.

Lemma 2. *An optimal solution of the One-Segment-Approximating Problem can be computed in $O(1)$ time.*

Proof. Consider the surface $\mathcal{F}$ defined by $z = F(y_1, y_2)$. $\mathcal{F}$ allows a description

$$z = A(y_1^2 + y_1 y_2 + y_2^2) + By_1 + Cy_2 + D$$

with some constants $A, B, C, D \in \mathbb{R}$. Thus, $\mathcal{F}$ forms an elliptic paraboloid in 3-space with the unique minimum (y_1^*, y_2^*). Therefore, all intersections of $\mathcal{F}$ with hyperplanes $z = z_0 \geq F(y_1^*, y_2^*)$ create ellipses with center (y_1^*, y_2^*, z_0). These ellipses originate from the standard ellipse by a clockwise rotation of 45 degrees – due to symmetry.

In order to find an integer point $(\hat{y}_1, \hat{y}_2)$ that minimizes the function $F(y_1, y_2)$, we consider the ellipse defined by the set of points (y_1, y_2) with $F(y_1, y_2) \leq F(y_1^* + 0.5, y_2^* + 0.5)$. This set is guaranteed to contain at least one and at most a constant number of integer points (y_1, y_2). In addition, the desired optimal integer point $(\hat{y}_1, \hat{y}_2)$ must be among them.

Now the new algorithm can easily be described. Let d_{ij} denote the squared-area difference of an optimal segment approximating the input curve in the interval $x \in [i, j]$. In addition, we let D_j denote the squared-area difference of an optimal polygonal chain in the interval $x \in [0, j]$. Then, the so-called minimum-weight path optimality conditions (see [1]) imply $D_j = \min_{i=0,\ldots,j-1} D_i + d_{ij}$.

For each $j \in \{1, \ldots, n-1\}$, we store its predecessor i for which the equation above is fulfilled. Doing this, we can explicitly reconstruct the minimum-weight path starting from D_{n-1}.

Thus, we obtain the following algorithm.

[Algorithm 2]
$D[0] = 0$;
for $x = 1$ **to** $n - 1$ **do**
    for $x' = x$ **to** $n - 1$ **do**
    solve the One-Segment-Approximation Problem for $x, x' \rightsquigarrow y, y'$ and d_{ij}
apply a single-source shortest-path algorithm $\rightsquigarrow D(1), \ldots, D(n-1)$.

It is easy to see that all d_{ij}'s can be computed in $O(n^2)$ time. This can further be reduced to $O(kn)$ time if we bound the horizontal length of the approximating line segments. In particular, if k is some constant, the total running time is linear. The running time of the single-source shortest-path algorithm is linear in the number of edges – due to the special structure of the underlying graph.

Theorem 3. *Algorithm 2 computes an optimally approximating x-monotone polygonal chain in $O(N + kn)$ time and $O(N + n)$ space.*

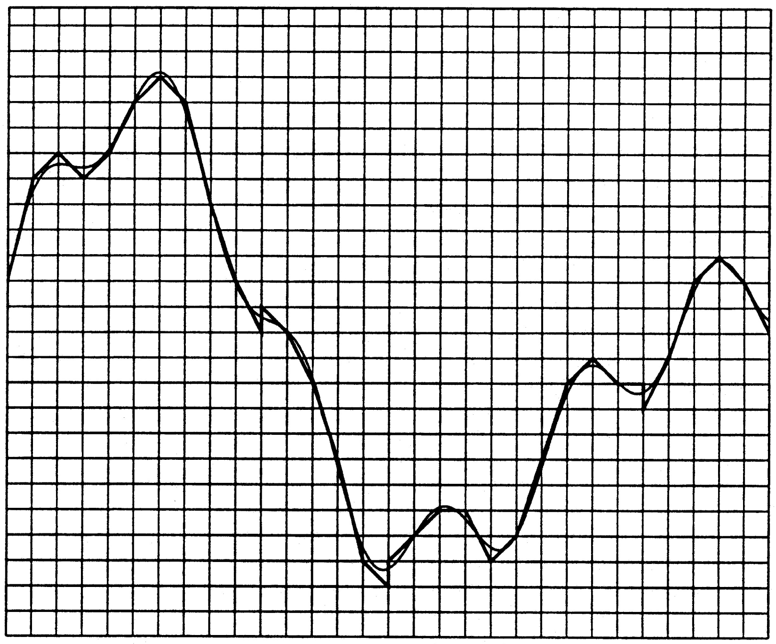

Fig. 2. The optimal approximating polygonal chain ($k = 1$).

5 Experimental Results

We have implemented the algorithms in C for planar curves using classical numerical integration methods. Figure 2 displays the experimental results for an input planar curve defined by

$$y = 17\sin(x/10.0 + 0.2) + 8\sin(x/7 + 0.3) + 4\sin(x/2 + 0.5), \quad 0 \leq x \leq 70$$

together with its optimally approximating x-monotone polygonal chain (for $k = 1$). For most curves and grids processed, a satisfying quality of approximation could be achieved for constant k.

6 Remarks and Open Problems

Although our investigations focused on approximations by means of straight line segments, it is not hard to extend the algorithm so that polygonal curves of fixed degree (e.g. parabolas) can also be incorporated without increasing the computational complexity.

One major open problem arises when we drop the constraint that the approximating polygonal chain must be x-monotone. In fact, it is not hard to construct an example where a polygonal chain that minimizes the squared-area difference is not x-monotone. For that, consider the input curve defined by the following polygonal chain $((0,0),(7/2,21/8),(7/2,0),(5,0))$ which is best approximated by the polygonal chain $((0,0),(4,3),(3,0),(5,0))$ which is not x-monotone. It is not clear how to solve the problem without assuming monotonicity.

Acknowledgement

Special thanks go to Clemens Luz [13] for implementing a workbench for approximating monotone functions and experimentally verifying the claimed properties of the algorithm. The authors are also grateful to Peter Remmele for carefully reading the manuscript.

References

1. R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network flows: theory, algorithms, and applications*, Prentice Hall, 1993
2. R.G. Cromley, *A vertex substitution approach to numerical line simplification*, Proc. 3^{rd} Spatial Data Handling, pp 57–64, 1988
3. D.H. Douglas and T.K. Peucker, *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*, Canadian Cartographer, Vol. 10, No. 2, pp 112–122, 1973
4. D. Eu and G. Toussaint, *On approximating polygonal curves in two and three dimensions*, Tech. Rep. SOCS 92.15, McGill Univ., Montreal, 1992
5. R. Fleischer, K. Mehlhorn, G. Rote, E. Welzl, and C.K. Yap, *Simultaneous inner and outer approximation of shapes*, Algorithmica, Vol. 8, pp 365–389, 1992
6. L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell, and J. Snoeyink, *Approximating polygons and subdivisions with minimum-link paths*, Proc. 2^{nd} Ann. SIGAL Internat. Sympos. Alg., LNCS 557, pp 151–162, 1991
7. J.E. Hershberger and J. Snoeyink, *An $O(n \log n)$ implementation of the Douglas-Peucker algorithm for line simplification*, Proc. 10^{th} Ann. ACM Symp. on Comp. Geometry, pp 383–384, 1994
8. H. Imai and M. Iri, *Computational-geometric methods for polygonal approximations of a curve*, Computer Vision, Graphics, and Image Processing, Vol. 36, pp 31–41, 1986
9. H. Imai and M. Iri, *An optimal algorithm for approximating a piecewise linear function*, Journal of Information Processing, Vol. 9, No. 3, pp 159–162, 1986
10. H. Imai and M. Iri, *Polygonal approximations of a curve – formulations and algorithms*, in G.T. Toussaint (Ed.), Computational Morphology, Elsevier Science Publishers, pp 71–86, 1988
11. Y. Kurozumi and W.A. Davis, *Polygonal approximation by the minimax method*, Computer Graphics and Image Processing, Vol. 19, pp 248–264, 1982
12. Z. Li and S. Openshaw, *Algorithms for automated line generalization based on a natural principle of objective generalization*, Intern. Journal of Geogr. Info. Systems, Vol. 6, No. 5, pp 373–389, 1992
13. C. Luz, *Approximation of x-monotone functions* (in German), Semesterarbeit, ETH Zurich, 1994
14. D.M. Mark, *Conceptual basis for geographic line generalization*, Proc. 9^{th} AUTOCARTO, pp 68–77, 1988
15. R.B. McMaster, *A statistical analysis of mathematical measures for line simplification*, Amer. Cartog., Vol. 13, pp 103–116, 1986
16. R.B. McMaster, *Automated line generalization*, Cartographica, Vol. 24, No. 2, pp 74–111, 1987
17. B.K. Natarajan, *On piecewise approximations to curves*, Tech. Rep. HPL-91-36, Hewlett Packard Labs., Palo Alto, 1991
18. U. Ramer, *An iterative procedure for the polygonal approximation of plane curves*, Comput. Vis. Graph. Image Process, Vol. 1, pp 244–256, 1972
19. B.K. Ray and K.S. Ray, *A new approach to polygonal approximation*, Pattern Recognition Letters, Vol. 12, pp 229–234, 1991
20. J. Roberge, *A data reduction algorithm for planar curves*, Computer Vision, Graphics and Image Processing, Vol. 29, 168–195, 1985
21. M. Visvalingam and J.D. Whyatt, *Line generalization by repeated elimination of points*, personal communications, 1994

This article was processed using the LaTeX macro package with LLNCS style