アルゴリズム 42−10
(1994. 11. 18)

Optimal Approximation of a Curve by a Polygonal Chain with Vertices on Grid Points

浅野哲夫[1]、加藤直樹[2]、Elena Lodi[3], Thomas Roos[4]
[1]大阪電気通信大学（寝屋川市初町）、[2]神戸商科大学（神戸市西区学園西町）
[3]University of Siena, Italy, [4]ETH, Zürich, Switzerlands

本論文では、平面曲線を格子点を結ぶ多角形チェインで近似する効率のよいアルゴリズムを提案する。すなわち、$m \times n$ の格子平面上に x 方向に単調な曲線 $y = f(x)$ が与えられたとき、この曲線を格子点を直線で結ぶ多角形チェインによって近似するのが問題である。与えられた曲線と近似多角形チェインによって囲まれた領域の面積や面積の２乗を最小化するなどの最適化基準について論じる。

入力の曲線は幾つかの x 方向に単調な曲線をつなげたものとして定義されていてもよい。近似に用いる線分の（水平方向の）長さの上限を k としたとき、提案するアルゴリズムは (kn) 時間と $O(n)$ の記憶量で最適解を見つけることができる。ただし、与えられた曲線の積分値などの基本的な計算が定数時間でできることを仮定している。この時間複雑度は入力関数の値域のサイズを表わす m に依存しないことに注意されたい。最小コストパス問題に還元するというのが基本的な考え方である。

Optimal Approximation of a Curve by a Polygonal Chain with Vertices on Grid Points

Tetsuo Asano[1]、Naoki Katoh[2]、Elena Lodi[3], Thomas Roos[4]
[1]Osaka Electro-Communication University, Japan
[2]Kobe University of Commerce, Kobe, Japan
[3]University of Siena, Italy, [4]ETH, Zürich, Switzerlands

This paper presents efficient algorithms for approximating a curve by a polygonal chain with vertices on grid points. Precisely, we are given an $x-$monotone curve $y = f(x)$ in some $m \times n$ grid plane. We want to approximate the curve by an $x-$monotone polygonal chain whose vertices are grid points. For that, we discuss several optimization criteria such as minimizing the area or squared area of the region bounded by the given curve and the polygonal chain.

Notice that the curve can be defined as a chain of $x-$monotone curves. When the horizontal length of a line segment to be used for approximation is restricted to be at most k, the proposed algorithm finds an optimal $x-$monotone polygonal chain in $O(kn)$ time and $O(n)$ space if we assume that some basic computations (e.g. the integral of a given curve) can be computed in constant time. This time complexity does not depend on m, the size of the value range of the input function. The key idea behind it is the reduction to that of finding a minimum-cost path.

1 Introduction

In this paper we consider the following problem. Given an $x-$monotone planar curve on some $m \times n$ grid plane, find an approximating $x-$monotone polygonal chain whose vertices are grid points according to some optimization criteria such as, e.g., minimizing the area or squared area of the region bounded by the curve and the polygonal chain.

This kind of problem arises in computer graphics when digitalizing a planar curve. The existing naive algorithm simply computes a grid point for each vertical line of the grid by rounding the intersection between the curve and this vertical line. However, the quality of this simple approximation is often not sufficient.

In order to qualify this simple approximation approach, we could consider several optimization criteria. However, we will mainly focus our attention on a simple and intuitive criterion, i.e., minimizing the squared area of the region bounded by the given curve and the resulting polygonal chain, although our approach could be generalized to more sophisticated criteria.

At first glance, the problem looks rather difficult. Surprisingly, the problem allows a quite simple solution. In fact, we present two efficient algorithms. One runs in $O(m^2n^2)$ time and $O(m+n)$ space. The key idea behind the algorithm is the reduction to that of finding a minimum-cost path in a directed graph. The other algorithm is more efficient. It can find an optimal solution in $O(n^2)$ time and $O(n)$ space. It should be noticed that it does not depend on m, the size of the value range of the input curve. Furthermore, when the length of a segment to be used is restricted to be at most k, the time complexity is reduced to $O(kn)$ time, which is $O(n)$ if k is some constant.

A related topic is the polygonal approximation of a curve studied by Imai and Iri[1], [3], [2]. The problem there is to approximate a given finer piecewise linear curve by another coarse piecewise linear curve consisting of fewer line segments. More specifically, it is to find a minimum width w of rectangles such that a sequence of n points can be covered by at most m rectangles with width w. Their algorithm runs in $O(mn \log^2 n)$ time.

2 Preliminaries

At the beginning, we are given a planar $m \times n$ grid $G = \{(x,y) \mid x = 0,1,...,n-1,\ y = 0,1,...,m-1\}$ and the corresponding continuous domain $D = \{(x,y) \mid 0 \leq x \leq n-1,\ 0 \leq y \leq m-1\}$. In addition, we are given a function

$$f : x \in [0, n-1] \mapsto y = f(x) \in [0, m-1]$$

in the domain D. We are not going to make strong assumptions on the type of function that fits into our approach, but we will assume some basic computations (e.g. the integral over f) to be available in constant time.

Now, we want to approximate (in a certain way) this given function by an x-monotone polygonal chain. In other words, the intersection of any vertical line in D with the chain is connected. Thereby, the vertices of this chain are restricted to the grid points of G. Thus, the output of our algorithm will be a polygonal chain

$$\mathcal{P} = P_0, \ldots, P_k,\ k \geq 1, \text{ with } P_i = (\bar{x}_i, \bar{y}_i) \in G \text{ for all } i = 0, \ldots, k.$$

Thereby, due to the x-monotonicity the x-coordinates of the vertices of the chain form a monotonously increasing sequence $\bar{x}_1 = 0 \leq \ldots \leq \bar{x}_k = n-1$ extending from the left to right side of the grid G.

There are several imaginable optimization criteria, among them are very intuitive ones such as minimizing the area or the squared area of the region bounded by the curve itself and the polygonal chain, i.e.

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

or

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

or minimizing the maximum vertical distance

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

among many others. We will focus our attention to the minimum area criterion.

Let $G = \{(x,y)|y = 0,1,...,m-1,\ x = 0,1,...,n-1\}$ be an $m \times n$ grid plane and $D = \{(x,y)|\ 0 \le y \le m-1, 0 \le x \le n-1\}$ is its corresponding continuous domain. Suppose that we are given an $x-$monotone curve $y = f(x)$ in the domain D where we allow vertical connection. In that sense we abuse the symbol $\int$. In what follows, $\int_a^b f(x)dx$ means $\sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} f(x)dx$, $x_0 = a, ..., x_{N+1} = b$ when the function $f(x)$ is defined as a chain of $x-$monotone functions defined in the intervals $(x_i, x_{i+1}], i = 0, ..., N$.

An input curve may be specified in two different ways: a polygonal chain whose vertices may have real coordinates except for the first and last $x-$coordinates (0 and $n-1$), and a chain of $x-$monotone curves. More specifically, an input curve is specified in the form:

$(x_0 = 0, f_0(x), x_1, f_1(x), x_2, \ldots, f_{N-1}(x), x_N = n-1)$.

This means a sequence of monotone curves $y = f_i(x),\ i = 0, \ldots, N-1$ each defined in the interval $[x_{i-1}, x_i]$. For simplicity we assume the continuity of an input curve, i.e., $f_i(x_{i+1}) = f_{i+1}(x_{i+1})$ for every $i = 0, \ldots, N-2$. This assumption is, however, easily removed if we allow to abuse the symbol of integral.

Given an input curve $y = f(x)$, we first calculate the integrals $f[i] = \int_0^i f(x)dx$, $xf[i] = \int_0^i xf(x)dx$, and $f2[i] = \int_0^i f(x)^2 dx$ for each $i = 0, 1, \ldots, n-1$. It is rather easy to see that this calculation is done in $O(N+n)$ time, where N and n are input complexity and the horizontal length of the grid plane. Once these values are computed, integral $\int_i^j f(x)dx$, for example, is obtained as $f[j] - f[i]$ in constant time.

Noteworthy is that we do not need to compute intersections between the curve $y = f(x)$ and polygonal chain. Notice that if the optimization criterion is to minimize the area of the region bounded by two curves then their intersections are required to compute the area. When an input planar curve is a polynomial function of degree more than five, there is no way to compute intersections exactly. In our case, on the other hand, it will be shown that there is no need to compute those intersections but it suffies to compute the integrals described above. Thus, our algorithm is implemented in a framework of the algebraic decision tree computation model.

Given an input $x-$monotone planar curve $y = f(x)$, we want to find an $x-$monotone polygonal chain with vertices on grid points that minimizes the squared area of the region bounded by the input curve and the polygonal chain.

3 Algorithm 1: Naive Algorithm

The first algorithm for finding an optimal polygonal chain approximating an input curve is based on a directed graph defined as follows: Each grid point in G corresponds to a vertex of the graph. Two grid points (x_i, y_i) and (x_j, y_j), $x_i < x_j$ are joined by a directed edge weighted by the value

$$\int_{x_i}^{x_j} (f(x) - a_{ij}x - b_{ij})^2 dx,$$

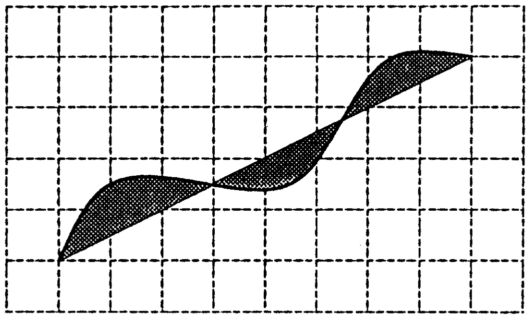

Figure 1: One-segment approximation.

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

$a_{ij} = (y_j - y_i)/(x_j - x_i)$,

$b_{ij} = y_i - a_{ij}x_i$.

Furthermore, we add a directed edge with zero weight between each pair of grid points on the same vertical line.

Now it is easy to see that an optimal polygonal chain corresponds to a minimum weight path starting from the lower left corner $(0,0)$ to the lower right corner $(n-1,0)$.

Theorem 3.1 *The weight of a minimum-weight path can be computed in $O(m^2n^2)$ time and $O(m+n)$ space in the $m \times n$ grid plane.*

Proof Notice that to evluate the squared area difference for a line segment (x, y, x', y') we need tha values of $\int_x^{x'} xf(x)dx$, $\int_x^{x'} f(x)dx$ and $\int_x^{x'} f(x)^2dx$. Apparently they take time $O(x-x')$ if we compute them without using any previous knowledge. To evaluate the square area difference $\delta(x, y, x', y')$ for the line segment between (x, y) and (x', y') we can use the integrals computed to evaluate $\delta(x, y, x'+1, y'+1)$. Thus, the computation is done in constant time.

There are $O(mn)$ vertices, and for each vertex of a location (i, j) there are $O(im)$ different ways to choose its preceding vertex. Therefore, the total time complexity is $O(m^2n^2)$. If we constructed the graph explicitly it would take $O(m^2n^2)$ edges. So, we do not do so but just examine each edge in a systematic manner. An important observation is that once we complete the computation of minimum-cost path to each vertex on some $x-$coordinate i then we take their minimum value and throw them away while keeping the minimum. Thus, we need only $O(m+n)$ space instead of $O(m^2n^2)$ space. □

It is an easy exercise to modify the algorithm so that not only the minimum weight but also a description of the path is obtained.

4 Algorithm 2: Efficient Algorithm

One big disadvantage of Algorithm 1 is that its time complexity depends on m, the value range of an input function. So, we next 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) minimized the squared area difference (see Figure 1):

$$\int_{x_1}^{x_2} [f(x) - \{\frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2\}]^2 dx.$$

We define

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

Rewriting the above expression using the equality between $y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2$ and $y = \frac{x - x_1}{x_2 - x_1}y_2 - \frac{x - x_2}{x_2 - x_1}y_1$, we have

$$F(y_1, y_2) = \frac{1}{3}(x_2 - x_1)(y_1^2 + y_1 y_2 + y_2^2) + \frac{2}{x_2 - x_1}y_1 \int_{x_1}^{x_2} xf(x)dx - \frac{2x_2}{x_2 - x_1}y_1 \int_{x_1}^{x_2} f(x)dx$$

$$-\frac{2}{x_2 - x_1}y_2 \int_{x_1}^{x_2} xf(x)dx + \frac{2x_1}{x_2 - x_1}y_1 \int_{x_1}^{x_2} f(x)dx + \int_{x_1}^{x_2} f(x)^2 dx.$$

Because of the form of the expression, $F(y_1, y_2)$ is minimized when y_1 and y_2 satisfy the simultaneous equations: $\frac{\partial F}{\partial y_1} = \frac{\partial F}{\partial y_2} = 0$. Let (y_1^*, y_2^*) be such a pair of values. That is,

$$y_1^* = \frac{6}{(x_2 - x_1)^2} \int_{x_1}^{x_2} xf(x)dx - \frac{2x_2 + 4x_1}{(x_2 - x_1)^2} \int_{x_1}^{x_2} f(x)dx,$$

and

$$y_2^* = -\frac{6}{(x_2 - x_1)^2} \int_{x_1}^{x_2} xf(x)dx + \frac{4x_2 + 2x_1}{(x_2 - x_1)^2} \int_{x_1}^{x_2} f(x)dx.$$

If y_1^* and y_2^* are both integers, we are done. Otherwise, we need to find integer lattice points $(\hat{y}_1, \hat{y}_2)$ that minimizes the $F()$ value. The following lemma guarantees that only a constant numbr of lattice points near the point (y_1^*, y_2^*) can be candidates for $(\hat{y}_1, \hat{y}_2)$.

Lemma 4.1 *When the function $F(y_1, y_2)$ is minimized when $y_1 = y_1^*$ and $y_2 = y_2^*$, an optimal solution (y_1^*, y_2^*) to the One-Segment-Approximating Problem can be found as a point among some constant number of points in the neighborhood of the point (y_1^*, y_2^*) in the y_1y_2-plane.*

Proof Consider a curve surface $z = F(y_1, y_2)$ in the y_1y_2z-space. It is evidently a paraboloid and it has a bottom point. Let (y_1^*, y_2^*) be the bottom point. When an arbitrary $z-$value z_0 is specified so that $z_0 > F(y_1^*, y_2^*)$, the intersection of the surface with the plane $z = z_0$ forms an ellipse centered at (y_1^*, y_2^*). The form of the expression for $F(y_1, y_2)$ implies that the curve $F(y_1, y_2) = z_0$ in the y_1y_2-plane is an ellipse resulting after rotating a standard ellipse by 45 degrees in the clockwise direction. See Figure 2 for illustration.

Let z_0 be such a $z-$value that the curve $z_0 = F(y_1, y_2)$ passes through the point $(y_1^* - 0.5, y_2^* - 0.5)$. Then, because of the symmetry it also passes through the point $(y_1^* + 0.5, y_2^* + 0.5)$.

What we want to find here is a lattice point $(\hat{y}_1, \hat{y}_2)$ to minimize the value $F(y_1, y_2)$. It is easy to see that the ellipse $F(y_1, y_2) = z_0 = F(\hat{y}_1, \hat{y}_2)$ does not contain any lattice point in its interior. Therefore, the point $(\hat{y}_1, \hat{y}_2)$ must be contained in or on the ellipse $F(y_1, y_2) = z_0$ defined above.

It is not so hard to see that the ellipse $z_0 = F(y_1, y_2)$ touches the lines $y_1 = y_1^* \pm 1$ and $y_2 = y_2^* \pm 1$. Thus, if we draw lines $y_1 = y_1^* - 1, y_1 = y_1^* - 0.5, y_1 = y_1^*, y_1 = y_1^* + 0.5, y_1 = y_1^* + 1, y_2 = y_2^* - 1, y_2 = y_2^* - 0.5, y_2 = y_2^*, y_2 = y_2^* + 0.5, y_2 = y_2 * +1$, the interior of the ellipse is covered by 14 small squares of side 0.5. Especially, the four squares which share the point (y_1^*, y_2^*) as their vertices are totally included in the ellipse. It is easy to see that the square of

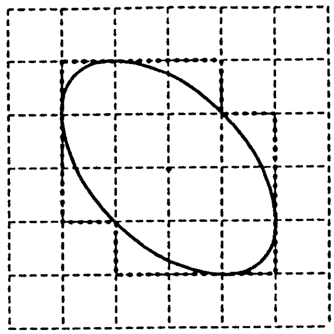

Figure 2: An ellipse defined by $F(y_1, y_2) = F(y_1^* + 0.5, y_2^* + 0.5)$.

side 1 contains only one lattice point unless the four corners are all lattice points. Depending on which quadrant contains a lattice point, we can enumerate at most four such small square regions of side 0.5 each that can contain an integer lattice point. Thus, we have the lemma. □

Now the algorithm is described easily. Let D_i denote the squared area difference of an optimal polygonal chain from $x = 0$ to $x = i$, and d_{ji} denote that of an optimal segment approximating the input curve in the interval $[x = j, x = i]$. Then, we have

$$D_i = \min_{j=i-1,i-2,...,0} D_j + d_{ji}.$$

In addition to D_i we calculate a structured information $p_i = (j, y_j; i, y_i)$ such that $D_i = D_j + d_{ji}$ and $(j, y_j; i, y_i)$ is a solution to the one-segment approximation problem in the interval $[j, i]$, for $i = 1, 2, ..., n-1$.

Obviously the minimum weight is given by D_{n-1} and the path itself is obtained by following the pointer $p_i's$ starting from p_{n-1}. If some distinction happens at some $x-$coordinate, we join those points by a vertical line segment. It takes $O(i-j)$ time to compute d_{ji} since the input curve may be defined as a chain of size $O(i-j)$ in the interval $[x = j, x = i]$ if we compute them independently. Thus, the time complexity becomes $O(n^3)$. However, we can save much time by a simple trick. Notice that when we need the value d_{ji} we have already computed $d_{j+1,i}$. It is easy to see that d_{ji} can be computed in constant time using the informations needed for computing $d_{j+1,i}$ since the hard parts are calculations of integrals $\int_{x=j}^{x=i} xf(x)dx$, etc. Obviously, $d_{i-1,i}$ can be calculated in constant time. Therefore, the total time we need is just $O(n^2)$. Furtherfore, if the length of the longest line segment used in the approximation is upper bounded by k, the total running time is $O(kn)$. In particular, if k is some constant, the total running time is linear in n.

Theorem 4.2 *Given an $x-$monotone curve $y = f(x), 0 \le x \le n-1$ consisting of N monotone curves and an integer $k < n$, we can find in $O(N + kn)$ time and $O(N + n)$ space an optimal polygonal chain that minimizes the squared area difference.*

5 Experimental Results

We have implemented the algorithm in C for various planar curves. Figure 3 shows the experimental results. Figure 3(a) shows an input planar curve defined by $y = 17sin(x/10.0 + 0.2) +$

$8sin(x/7+0.3)+4sin(x/2+0.5)$, $0 \leq x \leq 70$. The optimal polygonal chain that approximates the curve is given in (b) together with the input curve. It is seen that the approximation is good enough. The polygonal chain itself is shown in Figure 3(c).

6 Conclusion

In this short paper we have presented an efficient algorithm for approximating an $x-$monotone curve by a polygonal chain that minimizes the squared area difference between the two curves. We have implemented the algorithm in C and the experimantal results are quite promising from a point of human view.

In this article we had a constraint that the approximating polygonal must also be $x-$monotone although vertical connection is allowed. It is not so hard to construct an example for which a polygonal chain that minimizes teh squared area difference is not $x-$monotone. In fact, a polygonal chain $((0,0),(7/2,21/8),(7/2,0),(5,0))$ is best approximated by a polygonal chain $((0,0),(4,3),(3,0),(5,0))$ which is not $x-$monotone. It has been left as an open problem to solve the problem without assuming monotonicity. It is also open to approximate a closed planar curve.

7 Acknowledgement

This joint work was done while the third author visited the first author and the the first author visited the second author under the support of the Grant for Promotion of International Joint Work of Osaka Electro-Communication University.

This work was also partially supported by Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan.

References

[1] H. Imai and M. Iri: "Computational-Geometric Methods for Polygonal Approximations of a Curve," *Computer Vision, Graphics, and Image Processing*, 36, pp.31-41 (1986).

[2] H. Imai and M. Iri: "An Optimal Algorithm for Approximating a Piecewise Linear Function," *Journal of Information Processing*, 9, 3, pp.159-162 (1986).

[3] H. Imai and M. Iri: "Polygonal Approximations of a Curve — Formulations and Algorithms," *Computational Morphology*, edited by G. T. Toussaint, Elsevier Science Publishers, pp.71-86 (1988).

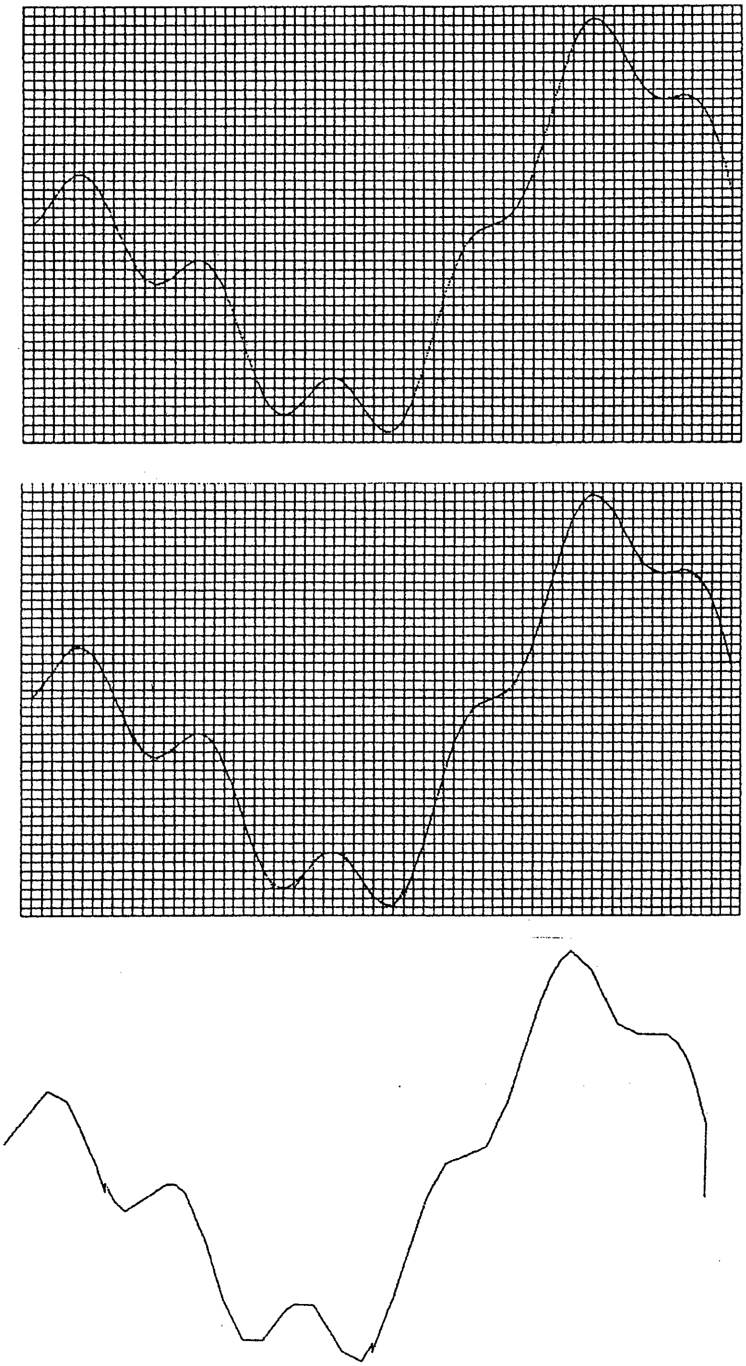

Figure 3: An experimental result. (a) Input curve. (b) Approximated polygonal chain and input curve. (c) Approximated polygonal chain.