On the Curve Equipartition Problem: a brief exposition of basic issues

Costas Panagiotakis * George Georgakopoulos * George Tziritas *

Abstract

We describe briefly the problem of partitioning a continuous curve into N parts with equal chords. (The length of a chord may be defined by any smooth distance metric applied on its endpoints-the Euclidean metric being one of them.) A have proved that a decision variation of this problem is NP-complete, yet for any continuous curve and any N there always exists at least one equipartition. In this work, we propose an approximate algorithm and also a steepest descent method that converges to an exact solution.

1 Introduction

The curve segmentation problem is a challenging problem of computational geometry. A huge number of applications, like object recognition and tracking, signal summarization and compression, curve simplification and computer graphics applications, are based on curve segmentation. Many computer graphics applications are based on curve segmentation problem, like surface simplification and 3D modelling. Most polygonal surface simplification methods employ triangles as their approximating elements when constructing a surface [3]. One of the most popular triangulation methods is Delaunay triangulation [5], [11].

On computer vision applications the curve segmentation problem also appears. Signal summarization and key frames detection methods using an appropriate feature set reduce the initial problem into a curve segmentation problem [4]. Methods for non articulated motion tracking are based on solutions of the curve equipartition problem [10].

Another example of such segmentation approach is the 2D or 3D polygonal approximation [2] or convex polygons [7]. This problem asks for computing another polygonal curve that approximates the original curve. The problem can be formulated in two ways [6], [1] : The problem of minimum error ($Min-\varepsilon$) and the the problem of minimum number of line segments ($Min-\#$).

*Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece {cpanag, ggeo, tziritas}@csd.uoc.gr

Symbols	Definitions
$C(t)$	The given curve, $t \in [0,1]$
N	The number of equal length chords
$d(x,y)$	$d(x,y) = \|C(x) - C(y)\|_2$
$U_{M\cdots K}$	$\{M, M+1, \cdots, K\}$

Table 1: Proof symbol table.

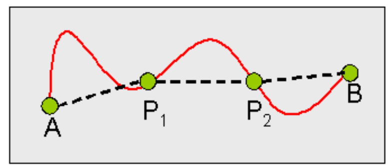


Fig. 1: An EP example for $N=3$, $|AP_1| = |P_1P_2| = |P_2B|$.

1.1 Problem Definition

The equipartition problem is defined as follows: Let $C(t), t \in [0,1]$ be a 2D acyclic curve[1] that starts on $A = C(0)$ and ends on $B = C(1)$. We have to compute $N-1$ sequential curve points $P_i, i \in U_{1\cdots N-1}$, $P_0 = A$, $P_N = B$ under the constraint $d(P_{i-1}, P_i) = d(P_i, P_{i+1})$, $i \in U_{1\cdots N-1}$. The problem is the curve partitioning into N parts with equal chords, so that the first starts from A and the last ends on B (Fig. 1). Some useful symbols are defined on Table 1.

The solution is obvious for $N = 1$, as we have one chord, AB. When $N = 2$, we have to locate a curve point P_1, so that $|AP_1| = |P_1B|$. This point can be given as the intersection of the curve with the AB segment bisector. When N is higher than two, there is not a trivial method to compute the equal length chords. The above problem can have more than one solutions depending on curve shape and the value of N. As N tends to infinity the problem solution (equal length chords) will be unique and it will approximate the curve (Fig. 2). More examples can be found in [9]. By our analysis, the EP problem admits always a solution, thus the decision version of EP is certainly not NP-complete. Yet, at

[1]We suppose that the curve is piecewise-algebraic. The problem can be defined in the same way in any dimension ($C(t) \in \Re^n$).

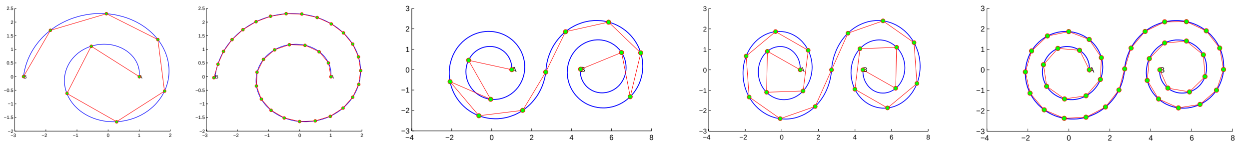


Fig. 2: EP examples for different N. The higher the N, the better the curve approximation.

the time, we cannot give a guarantee that it admits always a succint solution, thus we cannot assert that the functional version of it belongs to the TFNP class. Finally consider the following 'verification' version of the EP: given a curve C and a distance ρ is it possible to equipartition C into N parts, so their N chords are all equal to ρ? For this version, we do know something positive: we can prove that this version of EP is NP-complete (by a reduction from Knapsack).

1.2 An Equivalent Definition of the Problem

The smooth function $d(x, y) = |C(x) - C(y)|_2, \;\; x, y \in [0, 1]$ gives a different view and an equivalent definition of the problem. We can use as $d(x, y)$ any smooth metric like Euclidean distance. As a smooth metric distance, $d(x, y)$ is characterized by the following properties:

1. $d(x, y) = 0 \Leftrightarrow x = y$ (isolation).
2. $d(x, y) = d(y, x)$ (symmetry).
3. $d(x, y)$ inherits continuity existence from $C(t)$.
4. $d(x, y)$ can be defined in any dimension ($C(t) \in \Re^n$).

An equivalent problem definition can be derived using $d(x, y)$. A problem solution $\{0, t_1, t_2, \cdots, t_{N-1}, 1\}$ of curve $C(t)$, corresponds to the surface $d(x, y)$ as a point sequence, $(0, t_1), (t_1, t_2), \cdots, (t_{N-1}, 1)$. The length r of each chord is given by the following equation:

$$r = d(0, t_1) = d(t_1, t_2) = \cdots = d(t_{N-1}, 1) \qquad (1)$$

An alternative problem definition will be the determination of $\{t_1, t_2, \cdots, t_{N-1}\}$, so that Equation (1) will be satisfied. Under this definition we prove that the problem has at least one solution.

The rest of the paper is organized as follows: A brief description of the proof of solution existence for each chord number is presented in Section 2. The proposed algorithms that solve the problem are presented in Sections 3 and 4. Conclusions and discussion are provided in Section 5.

2 Existence Proof

In this section we are going to give a brief description of the proof that there exists at least one solution for each N. We are going to analyze the case of $N = 3$. The cases of $N > 3$ are faced with a generalization of the used methodology for $N = 3$ and their proof can be done inductively. The proof is presented with more details in [8].

The function $f_2(x, y) = d(x, y) - d(x, 0), \;\; x \in [0, 1], y \geq x$, is continuous and partially monotonous function. The null plane curves[2] of $f_2(x, y)$ will be continuous and partially monotonous. The total solutions of equal length chords for $N = 2$ are given by the points (x, y) of these curves, because $d(x, 0) = d(x, y)$ and $y \geq x$. Let $h_2(s) = [a_2(s), b_2(s)], s \in [0, 1]$ be the curve of $f_2(x, y)$ null plane, that starts from $[0, 0]$ ($h_2(0) = [0, 0]$). Then $a_2(s) \leq b_2(s)$, because the points of $h_2(s)$ are points of f_2 domain. This curve exists as $f_2(0, 0) = 0$. It can be proved that $h_2(s)$ ends on $y = 1$, equivalently $b_2(1) = 1$. We consider the continuous function $q(s)$ (Equation (2)). Using this function, we will find $\{t_1, t_2\}$ with $t_2 \geq t_1$ satisfying Equation (1) and the proposition will have been proved for $N = 3$.

$$q(s) = d(a_2(s), b_2(s)) - d(1, b_2(s)), \;\; s \in [0, 1] \qquad (2)$$

It holds that,

- $q(0) = d(0, 0) - d(1, 0) = -d(1, 0) < 0$ and
- $q(1) = d(a_2(s), 1) - d(1, 1) = d(a_2(s), 1) > 0$

At least a $s_2 \in (0, 1)$ exists (applying the Bolzano theorem) so that $q(s_2) = 0$. This means $d(a_2(s_2), b_2(s_2)) = d(1, b_2(s_2))$. Let $t_2 = b_2(s_2)$ and $t_1 = a_2(s_2), \Rightarrow t_2 \geq t_1$ and $d(t_1, t_2) = d(t_2, 1)$. The (t_1, t_2) is a point of $h_2 \Rightarrow d(t_1, t_2) = d(0, t_1)$. Thus we have found $\{t_1, t_2\}$ with $t_2 \geq t_1$ satisfying the equation (1). Finally, the problem has been proved for $N = 3$.

3 Iso-Level Algorithm (ILA)

The iso-level algorithm is based on the equivalent problem definition. It computes at least one solution or all the solutions (greedy version). It is inductive. Thus, when it is executed for N, it solves the problem for any number of parts (with equal chords) less than N.

[2] The null plane curves of $f(x, y)$ are defined by the equation $f(x, y) = 0$.

The major hypothesis of the method is that the function $d(x, y), x, y \in [0, 1]$ can be approximated by a polygonal surface $\hat{d}(x, y)$. Thus, the $\hat{d}(x, y)$ is determined by $d(m_k, m_l), k, l \in \{1, 2, \cdots, M\}$. Let

$$D_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \subset [0, 1]^2$$

with $x_i = y_i = m_i$, $i, j \in \{1, 2, \cdots, M\}$. The segment D_{ij} can be separated into two triangles: D^1_{ij} where $x - x_i \geq y - y_j$ and D^2_{ij} where $x - x_i < y - y_j$. Under our major hypothesis, we have considered that $\hat{d}(x, y), x, y \in D^1_{ij}$ or $x, y \in D^2_{ij}$ is a part of plane.

In each iteration step l, the algorithm computes the curves L_l so that if the point $(u, v) \in L_{l-1}, u > v$, then, it holds that $(z, u), z > u \in L_l \Leftrightarrow d(u, v) = d(z, u)$. These curves consist of line segments defined on D^1_{ij}, D^2_{ij}, so they can be computed from the line segments end points. For $l = 1$, it holds that,

$$L_1 = [(0, 0), (m_1, 0)] \cup \cdots \cup [(m_{M-1}, 0), (1, 0)].$$

Let $(x, y) \in L_l$, $x > y$. Under the above definition, the equipartition of curve $C(t), t \in [0, x]$ into l chords can be done using the precomputed curves $L_l, L_{l-1}, \cdots, L_1$ (see Fig. 3). The equipartition of curve $C(t), t \in [0, 1]$ into $l+1$ chords can be done using the curves $L_l, L_{l-1}, \cdots, L_1$. Let $q_l(u, v) = d(u, v) - d(u, 1)$, $(u, v) \in L_l$, $u > v$. This function is piecewise linear. The roots of this function will give the last two points $(\dot{t}_{l-1}, \dot{t}_l)$ of the equipartition. The other points are estimated using the rule of Fig. 3.

It can be proved that for each step there is a continuous curve $h_l \subset L_l$ starting from $[0, 0]$ and ending on axis $x = 1$ or $y = 1$ (see Fig. 3). We can compute at least one solution of the problem using these curves. The computation cost of h_l curves is $O(M \cdot N)$, because we can track them starting from their known end point $[0, 0]$. We can estimate a normalized error (NE) of an estimated equipartition of length chords by getting the standard deviation of the estimated length chords of this equipartition divided by the mean length chord of this equipartition ([8]). NE is decreased as M increases. It can be proved that NE is decreased by the factor $O(\frac{1}{M^2})$.

Figure 4 illustrates the results of this proposed algorithm for different curves and values of N. The null plane curves converge to the diagonal ($y = x$), as N increases (see Fig. 4(e)), and there exist exactly one solution. At least one solution belongs on the $h_N(s)$ null plane curve. However, in some cases, more solutions appear on other null plane curves (see Fig. 4(a), 4(c)).

4. Steepest Descent based Method

The steepest descent based method (SDM) converges to the closest solution to an initial equipartition, given this

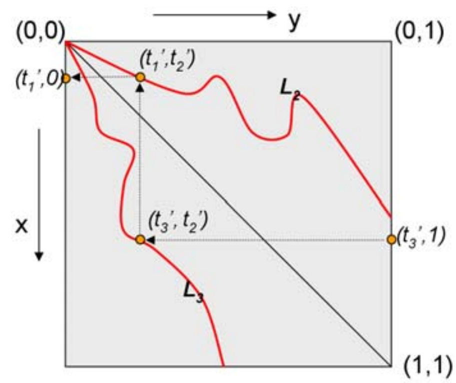


Fig. 3: An example of curve equipartition into 4 chords. It is shown the recursive computation of $\{\dot{t}_1, \dot{t}_2, \dot{t}_3\}$ and L_2, L_3 curves.

initial equipartition. The major advantage of this method is that the computed chords will have exactly the same length, as the end of the last chord is converging to B. In some cases the algorithm can not converge as there may appear local minima or jumps (loops) between different solutions. These phenomena are increased, when the initialization is far enough from an existing solution. But, when the problem has a unique solution, which is usually observed for high N, then the algorithm will converge. A pseudocode of this procedure is given hereafter.

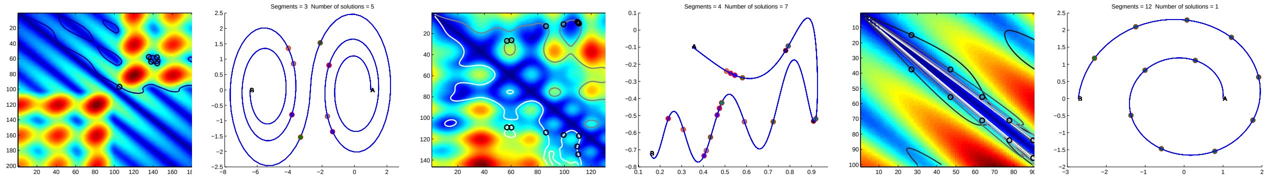
Steepest Descent based Algorithm

$s = r_0$
$P_0 = A$
Repeat
  for i=1:N
    $P_i = C(t_i) : (t_i > t_{i-1}) \wedge (|P_{i-1}P_i| = s)$
  end

$$s = \begin{cases} s + \lambda \frac{|B - P_N|}{N}, & P_N \in \text{inside of curve } C(t) \\ s - \lambda \frac{|B - P_N|}{N}, & P_N \in \text{ouside of curve } C(t) \end{cases}$$

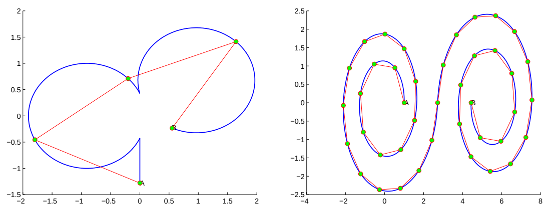
Until $|P_N - B| < T$

The learning rate λ determines the number of steps which are needed for convergence. However, when λ is set to a high value ($\lambda > 0.5$), it can cause instability and not convergence. For better convergence, we can start the method with $\lambda \approx 0.5$ and we decrease it to $\lambda \approx 0.05$. Conclusively, if we want to solve the problem for lower N, where there are possibly many solutions and local minima, it is better to execute first the approximate algorithm, getting a good initialization for the steepest descent based algorithm. For higher N, where the problem might have a unique solution, the proposed method will converge towards the exact solution, even if the initialization is not close to the solution. The time complexity of the algorithm is $O(N \cdot S)$, where S denotes the number of steps that are needed for



(a) (b) (c) (d) (e) (f)

Fig. 4: Results of greedy version of ILA. The estimated solutions are projected on $d(x, y)$ (left) with black cycles and on input curve $C(t)$ (right) with the same color points belonging to the same equipartition. The null plane curves are projected on $d(x, y)$, with gray colors, at both sides of diagonal $x = y$.



(a) (b)

Fig. 5: Steepest descent based algorithm results.

convergence. S depends on curve shape and how close to the final solution the initialization is. Results of the steepest descent based algorithm are shown in Fig. 5.

5 Conclusions

In this paper, we have discussed the curve equipartition problem (EP). If the chord length is part of the instance then the problem becomes NP-complete, but if the chord-length can be chosen at will it can be proved that there always exists at least one solution for any piecewise linear curve and thus for any continuous curve. An equivalent definition of the problem (using a distance metric on $[0..1] \times [0..1]$) leads also to an existence proof and to an approximate algorithm. The ILA can compute at least one solution or all the solutions using a greedy version of the algorithm. The output of this algorithm can be used to initialize a steepest descent based algorithm that converges to an exact solution.

References

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

[2] K.-L. Chung, W.-M. Yan, and W.-Y. Chen. Efficient algorithms for 3-d polygonal approximation based on lise criterion. *Pattern Recognition*, 35:2539–2548, 2002.

[3] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32:111–142, 1999.

[4] A. D. Doulamis, N. Doulamis, and S. Kollias. Non-sequential video content representation using temporal variation of feature vectors. *IEEE Trans. on Consumer Electronics*, 46:758–768, 2000.

[5] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. In *SIGGRAPH*, 1997.

[6] Y. Kurozumi and W. Davis. Polygonal approximation by the minimax method. *Computer Vision, Graphics and Image Processing*, pages 248–264, 1982.

[7] M. A. Lopez and S. Reisner. Hausdorff approximation of convex polygons. *Computational Geometry*, 32:139–158, 2005.

[8] C. Panagiotakis, G. Georgakopoulos, and G. Tziritas. The curve equipartition problem. *submitted to Computational Geometry, url = http://www.csd.uoc.gr/ cpanag/papers/EP.pdf*, 2005.

[9] C. Panagiotakis, G. Georgakopoulos, and G. Tziritas. Curve segmentation into equal segments. Technical Report CSD-TR-05-02, Computer Science Department, University of Crete, 2005 (in greek).

[10] C. Panagiotakis and G. Tziritas. Construction of animal models and motion synthesis in 3D virtual environments using image sequences. In *Proc. of the second Int. Symp. on 3DPVT*, 2004.

[11] C. Sohler. Fast reconstruction of delaunay triangulations. *Computational Geometry*, 31:166–178, 2005.