Computer Science Depatrment of The University of Auckland
CITR at Tamaki Campus (http://www.tcs.auckland.ac.nz)

CITR-TR-35 November 1998

Linear Time Calculation of 2D Shortest Polygonal Jordan Curves

Nan Yang [1] and Reinhard Klette [1]

Abstract

The length of curves may be measured by numeric integration if the curves are given by analytic formulas. Not all curves can or should be described parametrically. In this report we use the alternative grid topology approach. The shortest polygonal Jordan curve in a simple closed one-dimensional grid continuum is used to estimate a curve's length. An O(n) algorithm for finding the shortest polygonal Jordan curve is introduced, and its correctness and complexity is discussed.

[1] The University of Auckland, Computer Science Department, CITR,
Tamaki Campus (Building 731), Glen Innes, Auckland, New Zealand

Linear Time Calculation of 2D Shortest Polygonal Jordan Curves

Nan Yang and Reinhard Klette

Department of Computer Science, The University of Auckland
CITR, Tamaki Campus
Private Bag 92019, Auckland, New Zealand

Abstract: The length of curves may be measured by numeric integration if the curves are given by analytic formulas. Not all curves can or should be described parametrically. In this paper we use the alternative grid topology approach. The shortest polygonal Jordan curve in a simple closed one-dimensional grid continuum is used to estimate a curve's length. An $O(n)$ algorithm for finding the shortest polygonal Jordan curve is introduced, and its correctness and complexity is discussed.

Keywords: Multigrid convergence, shortest polygonal Jordan curve, curve length, grid continuum, shortest path, Yin-yang symbol

1 Introduction

In image analysis we have to deal with curves which are given in digitized pictorial form, and where a parametric description is typically not a final goal of the analysis process. Here, topological approaches for curve definitions may be more relevant in general, and may allow time-efficient and space-efficient solutions.

This paper deals with *simple closed one-dimensional grid continua* specifying a topological definition of curves in the orthogonal grid. The *length* of such grid continua in $\mathbf{R}^2$ is identified with the length of a shortest (polygonal) Jordan curve in a polygonally bounded set. It is theoretically proven that the length of the shortest polygonal Jordan curve converges towards the correct value when the gridding resolution approximates infinity [2].

An $O(n)$ time complexity algorithm for solving the shortest path problem in one-dimensional grid continua in the plane is introduced in Section 3. The algorithm is based on gridding techniques and the notion of a shortest path in a polygonally bounded set. Two soundness properties of the algorithm: *convergence* and *convergence towards the correct value*, are discussed with respect to different grid resolutions. The correctness and the complexity of the algorithm is also discussed.

2 Definitions and Theorems

In this section we introduce definitions and theorems which form the theoretical background for the algorithms discussed in Section 3. This introduction follows [1, 2, 5].

Consider an orthogonal grid in R^2. For $r = 0, 1, 2, \ldots$, and for each tuple (w_1, w_2) of integers let

$$N^r_{(w_1,w_2)} := \{(x_1, x_2) \in R^2 \mid w_i 2^{-r} \le x_i \le (w_i + 1)2^{-r}, i = 1, 2\}. \quad (1)$$

The set $N^r_{(w_1,w_2)}$ represents the *topological unit* of an orthogonal grid in R^2 with grid constant 2^{-r}. Let $\mathcal{A} \subset \mathcal{Z}^2$ and

$$M_r := \bigcup_{(w_1,w_2)\in A} N^r_{(w_1,w_2)} \quad (2)$$

be a compact set. A set $M_r \subset R^2$ which consists of at least two $N^r_{(w_1,w_2)}$ elements is called *edge connected* if each element of $M_r \subset R^2$ possesses an edge connected neighbour. An edge connected set $M_r \subset R^2$

is called a *planar grid continuum.* A few important definitions regarding the special case of simple one-dimensional planar grid continua follow:

Definition 1 *A planar grid continuum M_r is called a simple closed planar one-dimensional grid continuum if each typological unit of this set has exactly two edge connected neighbours.*

A simple closed planar one-dimensional grid continuum M_r represents a polygonally bounded compact set with boundary $\partial M_r = L_1 \cup L_2$, where L_1, L_2 are simple closed polygonal Joran curves, $L_1 \subset I(L_2)$, for which $dist(L_1, L_2) = 2^{-r}$, where 2^{-r} is the edge size of the $N^r_{(w_1,w_2)}$ unit. $I(.)$ specifies the topological interior, the distance function $dist$ is defined as the Hausdorff-Chebyshev distance between L_1 and L_2.

Definition 2 *The length of a simple closed planar one-dimensional grid continuum M_r with boundary $\partial M_r = L_1 \cup L_2$, where L_1, L_2 are simple closed polygonal Jordan curves, $L_1 \subset I(L_2)$, is defined as the length of the shortest polygonal Jordan curve in M_r encircling L_1.*

Definition 3 *Let $I_r^+(\Theta)$ be the union of all units for which $N_q^r \cap \Theta \neq \emptyset$. Let $I_r^-(\Theta)$ be the union of all units for which $N_q^r \subseteq I(\Theta)$. $I_r^+(\Theta)$ denotes the* outer interior *of Θ and $I_r^-(\Theta)$ is the* inner interior *of Θ.*

Theorem 1 *For any non-empty compact set $\Theta \subset R^2$, it holds that*

$$I_r^-(\Theta) \subset I(\Theta) \subset \Theta \subset I_r^+(\Theta)$$

where $I(\Theta)$ is the topological interior of Θ.

According to Theorem 1, the boundary of Θ must lie in the difference set between $I_r^+(\Theta)$ and $I_r^-(\Theta)$.

Theorem 2 *Assume that P_{L_1}, P_{L_2} are polygons in R^2 with boundaries L_1, L_2, respectively, and $L_1 \subset I(P_{L_2})$. Then there exists an uniquely defined Jordan curve L in $P_{L_2} \setminus I(P_{L_1})$ with minimum length, which is the boundary of a polygon P_L.*

The set $P_{L_2} \setminus I(P_{L_1})$ is polygonally bounded and compact, and the *shortest Jordan curve* $L = \partial P_L$ lies between the inner polygonal border L_1 and the outer polygonal border L_2. The vertices of the minumum Jordan curve L belong to the convex vertices of L_1, and the concave vertices of L_2 [2].

Theorem 3 *Let M_r be a simple closed planar one-dimensional grid continuum with $\partial M_r = L_1 \cup L_2$, where $L_1 \subset I(L_2)$, and L_1, L_2 are both simple closed polygonal Jordan curves. Assume that M_r contains a convex Jordan curve γ: $[0, d(\gamma)] \rightarrow R_2$ encircling L_1 of length $d(\gamma)$ parametrised by arclength. Then*

$$d(CH(L_1)) \leq d(\gamma) < d(CH(L_1)) + 8 \cdot 2^{-r}, for\ r = 0, 1, \dots,$$

where $d(CH(L_1))$ is the length of the boundary of the convex hull of L_1 and 2^{-r} is the edge size of the topological unit $N^r_{(w_1,w_2)}$.

This theorem [2] guarantees convergence of curve length in this special case of convex Jordan curve γ.

Assume polygonal Jordan curves L_1, L_2, then a ray $\overrightarrow{pq}$ *hits* L_1 *first* iff the ray starts in p, passes through q, and intersects L_1 afterwards at a point r, and does not intersect L_2 between q and r. Similiarly we can define ray $\overrightarrow{pq}$ *hits* L_2 *first.* It holds [2]:

Theorem 4 *Let L_1, L_2 be polygonal Jordan curves, the boundaries of polygons P_{L_1}, P_{L_2}, respectively, such that $P_{L_1} \subset I(P_{L_2})$. Let p_0 be a vertex of the shortest polygonal Jordan curve encircling L_1 in $G := P_{L_2} \setminus I(P_{L_1})$, with $\partial G = L_1 \cup L_2$. Then $p_0, p_1, \dots, p_{n-1}$ is the shortest polygonal Jordan curve in G encircling L_1 iff the ray $\overrightarrow{p_i p_{i+1(mod n)}}$ hits L_1 first if $p_{i+1(mod n)} \in L_2$, or the ray $\overrightarrow{p_i p_{i+1(mod n)}}$ hits L_2 first if $p_{i+1(mod n)} \in L_1$, for $i = 0, 1, \dots, n-1$.*

Theorem 2 (uniqueness), Theorem 3 (convergence) and Theorem 4 (shortest path) together provide the theoretical background for calculations of the length of a digitized Jordan curve in R^2.

3 Linear Algorithm for Shortest Polygonal Jordan Curves

Based on the notions and theorems discussed in Section 2, we introduce a shortest Jordan cruve algorithm which finds the shortest (polygonal) Jordan curve in a given simple closed one-dimensional grid continuum M_r. We assume that the first point of M_r is given, and M_r is represented in an $N \times N$ grid of r-grid points. The algorithm has linear run time, thus improving the $O(nlogn)$ algorithms presented in [2]. It has the following four steps:

Step (1): Trace the inner border L_1, and the outer border L_2. Find all extremal convex vertices of P_{L_1}, and all extremal concave vertices of P_{L_2}.
Step (2): Partition L_1 into convex subcurves and cusps, and G into corresponding pseudomonotone polygons.
Step (3): Find the shortest path in each pseudomonotone polygon.
Step (4): Combine the shortest paths to generate the shortest Jordan curve.

Steps (2) and (3) are recursively applied where L_1 and L_2 alternate in Algorithm 3.

The convexity of a polygonal Jordan curve is characterised by the turning angles of its edges. If we traverse the polygonal curve, and find all those "turning" points, we can partition the curve into subcurves of convexity and concavity. We call a subcurve that only consists of concave points a *cusp*.

Let L_1 and L_2 be polygonal Jordan curves, with $L_1 \subseteq P_{L_2}$. The vertices of the shortest polygonal Jordan curve in $G = P_{L_2} \setminus I(P_{L_1})$ belong to the convex vertices of L_1 or to the concave vertices of L_2. This property and the uniqueness of the shortest path in a polygonally bounded compact set enable us to partition the polygonal curve encircling L_1 at the turning convex vertices of L_1. After partiton, L_1 consists of convex subcurves and cusps. Informally speaking, this results into a partition of G into pseudomonotone polygons. Combining the shortest paths in all pseudomonotone polygons gives the shortest path for G.

We discuss the algorithms used in all the four steps and their time complexity.

Step (1). Algorithm 1 is used to track the border of the digital image and to classify the vertices.

Input: r-grid points in M_r
Output: labelled grid points in M_r according to convexity and concavity
take a first point of M_r
$currentPoint \Leftarrow firstPoint$
$currentDirection \Leftarrow firstDirection$
while $currentPoint \neq firstPoint \vee currentDirection \neq firstDirection$ **do**
visit $N_4(currentPoint)$ counter-clockwise {N_4 is the 4-neighbourhood}
if next point $\in M_r$ **then**
if $nextDirection - currentDirection = 90^o$ **then**
label currentPoint as CONVEX
else if $nextdirection - currentdirection = 270^o$ **then**
label currentPoint as CONCAVE
else {only norm border point}
label currentPoint as BORDER
$currentPoint \Leftarrow nextPoint$
$currentDirection \Leftarrow nextDirection$
end if
end if
end while

Algorithm 1: Contour tracing and classification algorithm

Vertices are classfied according to the changing of directions from the previous step to the next. The border is built in the counter-clockwise direction. A vertex is a *convex point* if it makes a left turn, or a vertex is a *concave point* if it makes a right turn, otherwise it is only an ordinary border point. Since the first point of M_r is given, and the test of each vertex can be accomplished in constant time, thus time complexity of the algorithm is $O(n)$. The convex points of L_1 and the concave points of L_2 identified by this algorithm will be used by Algorithm 3.

Step (2). Let C be a polygonal Jordan curve in R^2 space (convex vertices of L_1, or concave vertices of L_2), cusplist be the list of ending vertices of all the cusps in C, and length(C) be the number of vertices in C. The procedure to partition polygonal Jordan curve C into convex and concave subcurves (cusps) is described in algorithm 2.

The algorithm partitions C according to turning of edges. It traverses the curve C twice. In the first pass, it labels each vertices in C according to its turning. In the second pass, for each vertex that make a positive turn, if its predecessor or successor made a negative turn, then this vertex is a turning point, and is added to the cusp list.

```
Input: C, cusplist
Output: cusplist
  cusplist ⇐ ∅
  for i = 0 to i = length(C) do
    v1 = C(prev(i)), v2 = C(i), v3 = C(next(i))
    if (k · (v1v2 × v2v3) > 0) then
      turn(C(i)) = 1
    else {a negative turn, or no turning}
      turn(C(i)) = −1
    end if
  end for
  for i = 0 to i = length(C) do
    if turn((C(i)) = 1) ∧ (turn(C(prev(i))) = −1 ∨ turn(C(next(i))) = −1) then
      cusplist = cusplist ⊕ C(i)
    end if
  end for
```

Algorithm 2: Polygonal Jordan curve partition

Since only the traversal of polygonal curve is involved in this algrithm, and the test of turning is done in linear time, the time complexity of the algorithm is $O(n)$.

Step (3). The cusplist achieved by Algorithm 2 partitions L_1 into convex subcurves and cusps, and G into corresponding pseudomonotone polygons (or L_2 into concave subcurves and cusps). Finding the shortest path in a pseudomonotone polygon corresponding to a convex subcurve of L_1 (or concave subcurve of L_2) is trival, we simply connect all the vertices of the subcurve. Algorithm 3 finds the shortest path in a pseudomonotone polygon corresponding to a cusp of L_1. A dual procedure has to be used for an iteration step where a cusp of L_2 has to be considered.

Let b_1, b_2 are the two ending points of cusp, $next(b_1)$ is the first concave vertice of L_2 that are on the right of b_1, and $prev(b_2)$ is the first concave vertice of L_2 that are on the left of b_2. The shortest path in a cusp is denoted as *cuspsp*.

```
Input: b1, b2, L2, cuspsp ⇐ ∅
Output: cuspsp
  if b1b2 intersects L2 ∧ b1 ≠ b2 then
    b'1 ⇐ next(b1)
    b'2 ⇐ prev(b2)
    cusp_stpath(L2, b'1, b'2, cuspsp)
  end if
  cuspsp ⇐ b1 ⊕ cuspsp ⊕ b2
```

Algorithm 3: $cusp_{stpath}(L_2, b_1, b_2, cuspsp)$, algorithm to find shortest path in a cusp

The convex points of L_1 and the concave points of L_2 are identified by Algorithm 1. The correctness of the algorithm can be proven with Theorem 4, and since the procedure only involves connecting the convex points of L_1 and concave points of L_2, it runs in linear time.

The above algorithms are illustrated in Fig. 1.

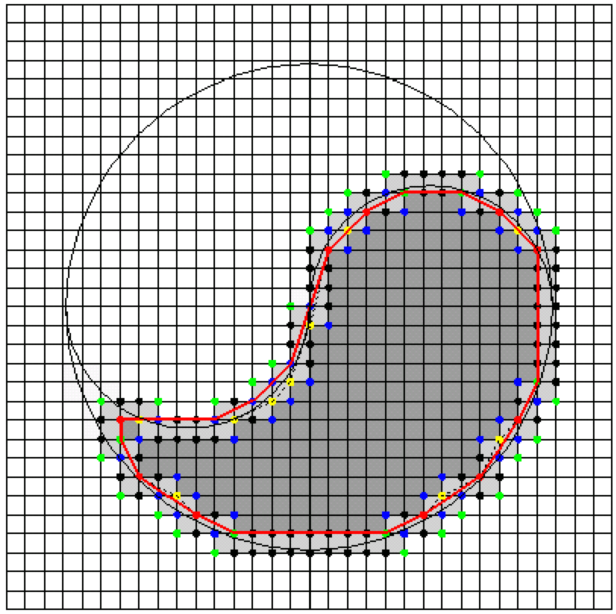

Figure 1: Shortest Jordan curve approximation of Yin-yang symbol

Since every step of the shortest Jordan curve algorithm runs in linear time, its time complexity is $O(n)$.

4 Generation of Test Data Sets

In this section we generate test data sets, i.e. simple closed planar one-dimensional grid continua, by the following steps:

Step (A): Digitize a given set Θ in the real plane by *intersection and inclusion digitization*.
Step (B): Shrink $I_r^+(\Theta)$ to get a simple closed one-dimensional continuum as difference set to $I_r^-(\Theta)$.

Step (A). We digitize Θ by *intersection and inclusion digitization*. After initializing $I^+(\Theta), I^-(\Theta)$ to empty sets and sampling the R^2 space into a regular $N \times N$ grid, we add grid cells to $I^+(\Theta)$, $I^-(\Theta)$ according to the number of grid cell vertices inside or on the boundary of object Θ. For each grid cell this only involves checking its four vertices against the boundary of Θ, thus the procedure can be completed in $O(n)$ time.

Step (B). Algorithm 4 is used to eliminate the extra elements of the continuum to produce a simply closed planar one-dimensional grid continuum. Let $Grid(i,j)$ be the grid point (i,j), $Cell(Grid(i,j))$ be the grid cell represented by $Grid(i,j)$, and $vertices(Grid(i,j))$ be the four vertices of the grid cell represented by $Grid(i,j)$.

Input: $N, I^+(\Theta), I^-(\Theta)$, Grid
Output: simple one-dimensional grid continuum achieved
  for $j = 0$ to $j = N$ **do**
    for $i = 0$ to $i = N$ **do**
      if $vertices(Grid(i,j)) \in I^+(\Theta) \wedge (vertices(Grid(i,j)) \notin I^-(\Theta)$ **then**
        $I^+(\Theta) \Leftarrow I^+(\Theta) \setminus Cell(Grid(i,j))$
      end if
    end for
  end for

Algorithm 4: Shrinking by direct elimination

Since the algorithm only requires one pass of the grid in the **for** loop, and each step within the loop involves testing 4 vertices of the grid cell against two interiors respectively, thus the time complexity of the algorithm is $O(n)$.

5 Conclusion

The given algorithm can be used in image analysis for estimating the length of boundaries of digital objects. The given object boundary may be interpreteted to be the polygonal boundary L_1, and the boundary L_2 may be generated by delation. The given algorithm also can be used to approximate the length of curves assuming that these curves are given in the real plane. Examples for the second approach can be found in [10].

References

[1] F. Sloboda, B. Zatko, R. Klette: On the Topology of Grid Continua. CITR-TR-22, The University of Auckland, CITR, June 1998.

[2] F. Sloboda, B. Zatko and J. Stoer: On approximation of planar one-dimensional continua. In: *Advances in Digital and Computational Geometry* (eds. R. Klette, A. Rosenfeld and F. Sloboda), Springer, Singapore 1998, pp 113 - 160.

[3] F. Sloboda: On approximation of Jordan surfaces - a topological approach. Internal Memo, CITR group, Tamaki, The University of Auckland, May 1997.

[4] R. Klette, Wu, Zhou: Multigrid Convergence of Surface Approximations Technical Report, The University of Auckland, CITR, June 1998.

[5] R. Klette: Approximation and representation of 3D objects. In: *Advances in Digital and Computational Geometry* (eds. R. Klette, A. Rosenfeld and F. Sloboda), Springer, Singapore 1998, pp 161 - 194.

[6] F. Sloboda, B. Zatko: Piecewise linear approximation of planar Jordan curves and arcs – theory and applications. In: *Theoretical Foundations of Computer Vision* (eds. W. Kropatsch, R. Klette, F. Solina), Springer, Wien 1996, pp 183 - 200.

[7] F. Sloboda, J. Stoer: On piecewise linear approximation of planar Jordan curves. *Comput. Appl. Math.* 55, 1994, pp 369 - 383.

[8] P. Ferianc, F. Sloboda, B. Zatko: On properties of the shortest path problem solution in a polygonally bounded compact set. Technical Report of the Inst. of Control Theory and Robotics, Bratislava, 1993.

[9] F. Sloboda and Ľ. Bvačík: On one-dimensional grid continua in R^3. Report of the Inst. of Control Theory and Robotics, Bratislava, 1996.

[10] N. Yang, R. Klette: Measurements of Arc Length's by Shortest Polygonal Jordan Curves. CIRT technical report, The University of Auckland, CITR, July 1998.