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

CITR-TR-55 December 1999

Digital Curves in 3D Space and a Linear-Time Length Estimation Algorithm

Thomas Bülow[1], and Reinhard Klette[2]

Abstract

We consider simple digital curves in a 3D orthogonal grid as special polyhedrally bounded sets. These digital curves model digitized curves or arcs in three-dimensional euclidean space. The length of such a simple digital curve is defined to be the length of the minimum-length polygonal curve fully contained and complete in the tube of this digital curve. So far no algorithm was known for the calculation of such a shortest polygonal curve. This paper provides an iterative algorithmic solution, including a presentation of its foundations and of experimental results.

[1] Institute of Computer Science, Christian Albrechts University, Preusserstrasse 1-9, 24105 Kiel, Germany

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

Digital Curves in 3D Space and a Linear-Time Length Estimation Algorithm

Thomas Bülow [(1)] and Reinhard Klette [(2)]

[(1)] Institute of Computer Science, Christian Albrechts University
Preusserstrasse 1–9, 24105 Kiel, Germany

[(2)] CITR, University of Auckland, Tamaki Campus, Building 731
Auckland, New Zealand

Abstract. We consider simple digital curves in a 3D orthogonal grid as special polyhedrally bounded sets. These digital curves model digitized curves or arcs in three-dimensional euclidean space. The length of such a simple digital curve is defined to be the length of the minimum-length polygonal curve fully contained and complete in the tube of this digital curve. So far no algorithm was known for the calculation of such a shortest polygonal curve. This paper provides an iterative algorithmic solution, including a presentation of its foundations and of experimental results.

1 Introduction

The analysis of digital curves in 3D space is of increasing practical relevance in volumetric image data analysis. A digital curve is the result of a process (3D skeleton, 3D thinning etc.) which maps captured 'curve-like' objects into well-defined digital curves (see definition below). The length of a simple digital curve in three-dimensional euclidean space is based on the calculation of the shortest polygonal curve in a polyhedrally bounded compact set [8, 9]. This paper presents an algorithm for calculating such a polygonal curve with measured time complexity in $\mathcal{O}(n)$, where n denotes the number of grid cubes of the given digital curve.

Any grid point $(i, j, k) \in \mathbb{R}^3$ is assumed to be the center point of a *grid cube* with *faces* parallel to the coordinate planes, with *edges* of length 1, and *vertices*. *Cells* are either cubes, faces, edges or vertices. The intersection of two cells is either empty or a joint *side* of both cells.

We consider a non-empty finite set K of cells such that for any cell in K it holds that any side of this cell is also in K. Such a set K is a special finite *euclidean complex* [6]. Let $dim(a)$ denote the dimension of a cell a, which is 0 for vertices, 1 for edges, 2 for faces and 3 for cubes. Then $[K, \subset, dim]$ is also a *cell complex* [4, 6, 10] with properties such as **(i)** $\subset$ is transitive on K, **(ii)** dim is monotone on K with respect to $\subset$, and **(iii)** for any pair of cells $a, b \in K$ with $a \subset b$ and $dim(a) + 1 < dim(b)$ there exists a cell $c \in K$ with $a \subset c \subset b$. Cell b *bounds* a cell a iff $a \subset b$, and b is a *proper side* of a in this case. Two cells a and b are *incident* iff a bounds b, or b bounds a.

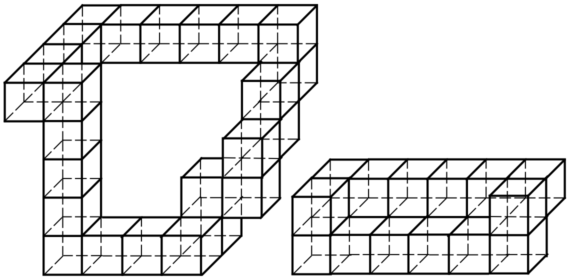

Fig. 1. Two cube-curves in 3D space.

We define *digital curves* g in 3D space with respect to such a euclidean complex as special sequences $(z_0, z_1, ..., z_m)$ of cells where z_i is incident with z_{i+1}, and $|dim(z_i) - dim(z_{i+1})| = 1$, for $i + 1 \ (mod \ m + 1)$. There are (at least) three different options which may depend upon an application context, or upon a preference of either a grid-point model or a cellular model which are dual approaches [2]. Let $n \geq 1$.

(i) An *edge-curve* is a sequence $g = (v_0, e_0, v_1, e_1, ..., v_n, e_n)$ of vertices v_i and edges e_i, for $0 \leq i \leq n$, such that vertices v_i and v_{i+1} are sides of edge e_i, for $0 \leq i \leq n$ and $v_{n+1} = v_0$. It is *simple* iff each edge of g has exactly two bounding vertices in g. It follows that a vertex or edge is contained at most once in a simple edge curve. [1]

(ii) A *face-curve* is a sequence $g = (e_0, f_0, e_1, f_1, ..., e_n, f_n)$ of edges e_i and faces f_i, for $0 \leq i \leq n$, such that edges e_i and e_{i+1} are sides of face f_i, for $0 \leq i \leq n$ and $e_{n+1} = e_0$. It is *simple* iff $n \geq 4$, and for any two faces f_i, f_k in g with $|i-k| \geq 2 \ (mod \ n+1)$ it holds that if $f_i \cap f_k \neq \emptyset$ then $|i-k| = 2 \ (mod \ n+1)$ and $f_i \cap f_k$ is a vertex.

(iii) A *cube-curve* is a sequence $g = (f_0, c_0, f_1, c_1, ..., f_n, c_n)$ of faces f_i and cubes c_i, for $0 \leq i \leq n$, such that faces f_i and f_{i+1} are sides of cube c_i, for $0 \leq i \leq n$ and $f_{n+1} = f_0$. It is *simple* iff $n \geq 4$, and for any two cubes c_i, c_k in g with $|i - k| \geq 2 \ (mod \ n + 1)$ it holds that if $c_i \cap c_k \neq \emptyset$ then either $|i - k| = 2 \ (mod \ n + 1)$ and $c_i \cap c_k$ is an edge, or $|i - k| = 3 \ (mod \ n + 1)$ and $c_i \cap c_k$ is a vertex. A *tube* **g** is the union of all cubes contained in a cube-curve g. It is a polyhedrally-bounded compact set in $\mathbb{R}^3$, and it is homeomorphic with a torus in case of a simple cube-curve. [2]

This paper deals exclusively with simple cube-curves. The cube-curve on the left of Fig. 1 is simple, and the cube-curve on the right is not. The latter

[1] This definition is consistent with, eg., the definition of a *4-curve* in [7] (see proposition 2.3.3) for 2D grids where our edges are 'hidden' in a neighborhood definition, or of a *closed simple path* in [11] (see page 7) for undirected graphs.

[2] *Closed simple one-dimensional grid continua* [8, 9] are defined such that each cube of g has exactly two bounding faces in g.

example shows that the polyhedrally-bounded compact set $\mathbf{g}$ of a cube-curve g is not necessarily homeomorphic with a torus if each cube of this cube-curve g has exactly two bounding faces in g. A (Jordan) curve is *complete in* $\mathbf{g}$ iff it has a non-empty intersection with any cube contained in g.

Definition 1. A *minimum-length polygon* (MLP) of a simple cube-curve g is a shortest polygonal simple curve $\mathcal{P}$ which is contained and complete in tube $\mathbf{g}$.

Following [8, 9], the *length* of a simple cube-curve g is defined to be the length $l(\mathcal{P})$ of an MLP of g.

A simple cube-curve g is *flat* iff the center points (i, j, k) of all cubes contained in g are in one plane parallel to one of the coordinate planes. A non-flat simple cube-curve in $\mathbb{R}^3$ specifies exactly one minimum-length polygonal simple curve (MLP, minimum-length polygon) which is contained and complete in its tube [8]. The MLP is not uniquely specified in flat simple cube-curves. Flat simple cube-curves may be treated as square-curves in the plane, and square-curves in the plane are extensively studied, see, eg. [3]. It seems there is no straightforward approach to extend known 2D algorithms to the 3D case. An important reason for that may be that 2D algorithms for (multigrid-convergent) perimeter estimation [3] may be such that all calculated vertices are grid points or vertices, but in the 3D case we are faced with a qualitatively new situation for the calculated vertices. The minimum-length polygon considered in this paper leads to vertices with real coordinates (not just multiples of integers as in the 2D case), ie. the model of cell complexes is considered as being embedded into the euclidean space. However, independent upon the dimension global information has to be taken into account for length calculation of digital curves to ensure multigrid convergence.

The paper is organized as follows: Section 2 informs about theoretical fundamentals relevant for the length calculation of cube-curves. Section 3 specifies our algorithm for length calculation, including theoretical results concerning the initialization process. Section 4 informs about a specific algebraic solution for one step in the algorithm to support a constant-time solution of this processing step. Section 5 informs about experimental results with respect to time complexity and length convergence. The conclusions also contain two open problems.

2 Simple cube-curves

This section contains fundamentals used in our algorithm for calculating the length of a simple cube-curve. Let g be a simple cube-curve, and $\mathcal{P} = (p_0, p_1, ..., p_m)$ be a polygonal curve complete and contained in $\mathbf{g}$, with $p_0 = p_m$.

Lemma 2. *It holds $m \geq 3$, for any polygon $\mathcal{P} = (p_0, p_1, ..., p_m)$ complete and contained in a simple cube-curve. If $m \geq 3$ then it holds that two line segments cannot be complete in any simple cube-curve.*

Proof. Cases $m = 0$ and $m = 1$ would be a point, and $m = 2$ would be a straight line segment. Both cases are excluded because our simple cube-curves

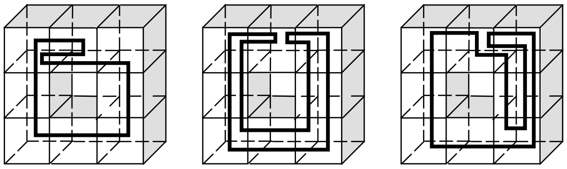

Fig. 2. Curves complete and contained in a tube.

are homeomorphic to the torus. The case $m = 3$ (a triangle) is possible, eg. for the simple cube-curve shown in Fig. 2. However, in this minimum case of $m = 3$ it holds that no side of the triangle may be completely contained within one of the cubes. □

Following [9], the curves on the left and on the right in Fig. 2 are *non-contractible in* **g**, and the curve in the middle is *contractible in* **g** by continuous contractions into a single point, ie. there are continuously repeated topological (ie. continuous and bijective) transformations of this curve into a family of disjoint curves, all contained in **g**, which converge towards a single point [5].

A Jordan curve γ *passes through* a face f iff there are parameters t_1, t_2, T such that $\{\gamma(t) : t_1 \leq t \leq t_2\} \subseteq f$, and $\gamma(t_1 - \varepsilon) \notin f$, $\gamma(t_2 + \varepsilon) \notin f$, for all ε with $0 < \varepsilon \leq T$. During a traversal along curve γ we *enter* a cube c at point $\gamma(t_1) \in c$ if $\gamma(t_1 - \varepsilon) \notin c$, and we *leave* c at point $\gamma(t_2) \in c$ if $\gamma(t_2 + \varepsilon) \notin c$, for all ε with $0 < \varepsilon \leq T$. A traversal is defined by the starting vertex p_0 of the curve and the given orientation.

We consider polygonal curves $\mathcal{P}$. Let $\mathcal{C}_\mathcal{P} = (c_0, c_1, ..., c_n)$ be the sequence of cubes in the order how they are entered during curve traversal. If $\mathcal{P}$ is complete and contained in a tube **g** then it follows that $\mathcal{C}_\mathcal{P}$ contains all cubes of g, and there are no further cubes in $\mathcal{C}_\mathcal{P}$.

Lemma 3. *For an MLP $\mathcal{P}$ of a simple cube-curve g it holds that $\mathcal{C}_\mathcal{P}$ contains each cube of g just once.*

Proof. Assume that $\mathcal{P}$ enters the same cube c of g twice, say at point q_1 first and at point q_2 again. Both points may be on one face of c, see Fig. 2 on the left and on the right, or on different faces of c, see Fig. 2 middle.

First consider the case that both entry points q_1 and q_2 of c are on one face f of cubes c and c'. Assume the number of passes of $\mathcal{P}$ through f is odd. We insert points q_1 and q_2 into $\mathcal{P}$ as new vertices which split the resulting polygonal curve into two polygonal chains, $\mathcal{P}_1 = (q_2, ..., q_1)$ and $\mathcal{P}_2 = (q_1, ..., q_2)$ such that the union of both is $\mathcal{P}$. The length of $\mathcal{P}_i$ exceeds the length of the straight line segment q_1q_2, for $i = 1, 2$. W.l.o.G. let $\mathcal{P}_1$ be the chain which does not pass through f. It follows that $\mathcal{P}_1$ is complete in **g**. Because the cube c is convex it also contains the straight line segment q_1q_2. We replace the polygonal sequence $\mathcal{P}_2$ by q_1q_2, ie. we replace $\mathcal{P}$ by $\mathcal{Q} = (q_1, q_2, ..., q_1)$. Curve $\mathcal{Q}$ is still complete and contained in g, but shorter than $\mathcal{P}$ which contradicts our assumption that $\mathcal{P}$ is an MLP of g.

Now assume that the number of passes of $\mathcal{P}$ through f is even, it enters c at q_1, then it passes f and enters c' at r_1, then it passes f again and enters c at q_2, then it passes f again and enters c' at r_2. There may be a further even number of passes of $\mathcal{P}$ through f before the curve returns to q_1. We insert points q_1, r_1, q_2, r_2 into $\mathcal{P}$ as new vertices which split the resulting polygonal curve into four polygonal chains, $\mathcal{P}_1 = (q_1, ..., r_1)$, $\mathcal{P}_2 = (r_1, ..., q_2)$, $\mathcal{P}_3 = (q_2, ..., r_2)$ and $\mathcal{P}_2 = (r_2, ..., q_1)$ such that the union of all four is $\mathcal{P}$. It follows that

$$\mathcal{C}_{\mathcal{P}_1} \subseteq \mathcal{C}_{\mathcal{P}_3} \vee \mathcal{C}_{\mathcal{P}_3} \subseteq \mathcal{C}_{\mathcal{P}_1} ,$$

and an analog conclusion for $\mathcal{P}_2$ and $\mathcal{P}_4$. W.l.o.g. let $\mathcal{C}_{\mathcal{P}_1} \subseteq \mathcal{C}_{\mathcal{P}_3}$. Then we replace in $\mathcal{P}$ the polygonal chain $\mathcal{P}_1$ by the straight line segment $q_1 r_1$ which is in f. The length of $\mathcal{P}_1$ exceeds the length of the straight line segment $q_1 r_1$. Thus the resulting polygonal curve is still complete and contained in g, but shorter than $\mathcal{P}$ which also contradicts our assumption that $\mathcal{P}$ is an MLP of g.

We consider the second case that both points q_1 and q_2 are on different faces of cube c, say q_1 on face f_1 and q_2 on face f_2. Because q_2 is a re-entry point to cube c there must be a point q_{ex} in f_2 where we leave c before entering c again at q_2. If there is another re-entry point on face f_2 then we are back to case one. It follows that $\mathcal{P}$ leaves c once and enters c once. Assume that f_2 is also a face of cube $c' \neq c$ of g. If $\mathcal{P}$ would not intersect the second face of c' contained in g then we may replace the polygonal subsequence $(q_{ex}, ..., q_2)$ (which is contained in c' but not in f_2) by the shorter straight line segment $q_{ex}q_2$ which is contained in f_2 and thus in c', ie. the resulting polygonal curve would be shorter and still contained and complete in g. It follows that the curve $\mathcal{P}$ has to leave cube c' through its second face contained in g. Tracing g around means that we arrive at the cube $c'' \neq c$ which is also incident with face f_1, and we leave c'' (and enter c) at a point which may be equal to q_1, and we enter c'' again through f_1. Thus $\mathcal{P}$ contains two polygonal subsequences which are both contained and complete in g. This contradicts the shortest-length constraint. □

Now we consider a special transformation of polygonal curves. Let $\mathcal{P} = (p_0, p_1, ..., p_m)$ be a polygonal curve contained in a tube g. A polygonal curve $\mathcal{Q}$ is a g-*transform* of $\mathcal{P}$ iff $\mathcal{Q}$ may be obtained from $\mathcal{P}$ by a finite number of steps, where each step is a replacement of a triple a, b, c of vertices by a polygonal sequence $a, b_1, ..., b_k, c$ such that the polygonal sequence $a, b_1, ..., b_k, c$ is contained in the same set of cubes of g as the polygonal sequence a, b, c. The case $k = 0$ characterizes the deletion of vertex b, the case $k = 1$ characterizes a move of vertex b within g, and cases $k \geq 2$ specify a replacement of two straight line segments by a sequence of $k + 1$ straight line segments, all contained in g.

Lemma 4. *Let $\mathcal{P}$ be a polygonal curve complete and contained in the tube* g *of a simple cube-curve g such that $\mathcal{C}_{\mathcal{P}}$ is without repetitions of cells. Then it holds that any* g*-transform of $\mathcal{P}$ is also complete and contained in* g.

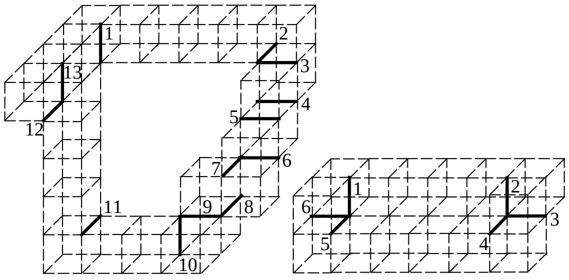

Fig. 3. Critical edges of two cube-curves.

Proof. By definition of the $\mathbf{g}$-transform it follows that this curve is also contained in $\mathbf{g}$. Because $\mathcal{C}_{\mathcal{P}}$ is without repetitions of cells it holds that $\mathcal{P}$ traces $\mathbf{g}$ cell by cell, starting with one vertex in one cell and returning to the same vertex. From Lemma 2 we know that $\mathcal{P}$ has at least three vertices, ie. at least three line segments, and that for the minimum case of $m = 3$ it holds that two line segments cannot be complete in g, ie. there is at least one cube not intersected by these two line segments. Thus a replacement of two line segments (within the same set of cells of g) cannot transform $\mathcal{P}$ into a curve contractible in $\mathbf{g}$, ie. the curve remains complete in $\mathbf{g}$. □

An edge contained in a tube $\mathbf{g}$ is *critical* iff this edge is the intersection of three cubes contained in the cube-curve g. Figure 3 illustrates all critical edges of the cube-curves shown in Fig. 1. Note that simple cube-curves may only have edges contained in three cubes at most. For example, the cube-curve consisting of four cubes only (note: there is one edge contained in four cubes in this case) was excluded by the constraint $n \geq 4$.

Theorem 5. *Let g be a simple cube-curve. Critical edges are the only possible locations of vertices of a shortest polygonal simple curve contained and complete in tube* $\mathbf{g}$.

Proof. We consider arbitrary (flat or non-flat) simple cube-curves g, ie. the MLP may not be uniquely defined.

Let $\mathcal{P} = (p_0, p_1, ..., p_m)$ be a shortest polygonal simple curve contained and complete in tube $\mathbf{g}$, with $p_0 = p_m$ and $m \geq 3$. We consider w.l.o.g. the polygonal subsequence (p_0, p_1, p_2) of such a shortest polygonal simple curve contained and complete in tube $\mathbf{g}$. We will show that p_1 is on a critical edge. According to Lemma 3 we know that $\mathcal{C}_{\mathcal{P}}$ is without repetitions, ie. we may apply Lemma 4 for this curve $\mathcal{P}$ and tube $\mathbf{g}$.

We can exclude the case that p_1 is colinear with p_0 and p_2, because p_1 would be no vertex of a polygon in such a case. Three non-colinear points p_0, p_1, and p_2 define a triangular region $\triangle(p_0, p_1, p_2)$ in a plane $\mathcal{E}$ in $\mathbb{R}^3$. The following

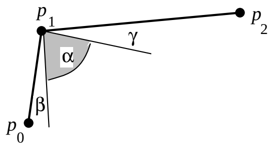

Fig. 4. Sketch of point p_1.

considerations are all for geometric configurations within this plane $\mathcal{E}$. In this proof, a *boundary point* is a point on the boundary $\partial \mathbf{g}$.

At first we ask whether p_1 may be moved into a new point p_{new} within the triangle $\triangle(p_0, p_1, p_2)$ towards line segment p_0p_2 such that a resulting polygonal subsequence $(p_0, ..., p_{new}, ..., p_2)$ remains to be contained in $\mathbf{g}$. This describes a $\mathbf{g}$-transform of $\mathcal{P}$, and the resulting curve would be complete and contained in $\mathbf{g}$. It can be of shorter length if the intersection of an ε-neighborhood of p_1 with $\triangle(p_0, p_1, p_2)$ is in $\mathbf{g}$, for $\varepsilon > 0$. It follows that such a move of p_1 is impossible, ie. it follows that for any $\varepsilon > 0$ there is at least one boundary point q in an ε-neighborhood of p_1 and on one of the line segments p_0p_1 or p_1p_2, avoiding such a move of p_0 into the triangle $\triangle(p_0, p_1, p_2)$. It follows that p_1 itself is a boundary point.

The situation of an ε_0-neighborhood at point p_1 is illustrated in Fig. 4. Angle α represents the region not in $\mathbf{g}$. Angles β and γ are just inserted to mention that they may be zero, and their actual value is not important in the sequel. It holds $\alpha < \pi$ because it is bounded by an inner angle of the triangle $\triangle(p_0, p_1, p_2)$.

A boundary point may be a point within a face, or on an edge. Assume first that boundary point p_1 is within a face f. Plane $\mathcal{E}$ and face f either intersect in a straight line segment, or face f is contained in $\mathcal{E}$. The straight line situation would contradict that $\alpha < \pi$ in the ε_0-neighborhood at point p_1, and $f \subset \mathcal{E}$ would allow to move p_1 into a new point p_{new} within $\triangle(p_0, p_1, p_2)$ towards line segment p_0p_2 which contradicts our MLP assumption.

There are three different possibilities for an edge contained in $\mathbf{g}$: we call it an *uncritical edge* if it is only in one cube contained in g, it is an *ineffective edge* if it is in exactly two cubes contained in g, and it is a critical edge (as defined

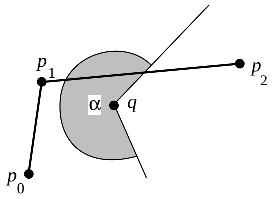

Fig. 5. Intersection with an uncritical edge.

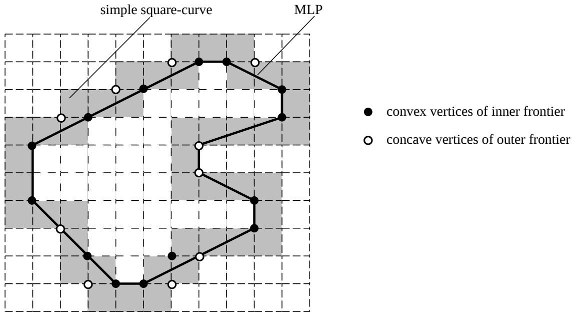
simple square-curve
MLP
● convex vertices of inner frontier
○ concave vertices of outer frontier

Fig. 6. Convex vertices of inner and concave vertices of outer frontier of the tube of a simple square curve in the plane.

above) in case of three cubes. Point p_1 cannot be on an ineffective edge such that it is also not on a critical or uncritical edge, because this corresponds to the situation being within a face as discussed before. Point p_1 also cannot be on an uncritical edge such that it is also not on a critical edge. Figure 5 illustrates an intersection point q with an uncritical edge in plane $\mathcal{E}$ assuming that this edge is not coplanar with $\mathcal{E}$. The resulting angle $\alpha > \pi$ (region not in $\mathbf{g}$ in an ε-neighborhood of q) does not allow that p_1 is such a point. If the uncritical edge is in $\mathcal{E}$ then angle α would be equal to π, what is excluded at p_1 as well. So there is only one option left. Point p_1 has to be on a critical edge (in fact, the angle α is less than π for such an edge). □

Note that this theorem also covers flat simple cube-curves with a straightforward corollary about the only possible locations of MLP vertices within a simple square-curve in the plane (see 6): such vertices may be convex vertices of the inner frontier or concave vertices of the outer frontier only because these are the only vertices incident with three squares of a simple square-curve.

3 Rubber-band algorithm

Our algorithm is based on the following physical model: Assume a rubber band is laid through the tube $\mathbf{g}$. Letting it move freely it will contract to the MLP which is contained and complete in $\mathbf{g}$ (assumed the band is slippery enough to slide across the critical edges of the tube). The algorithm consists of two subprocesses: at first an initialization process defining a simple polygonal curve $\mathcal{P}_0$ contained and complete in the given tube $\mathbf{g}$ and such that $\mathcal{C}_{\mathcal{P}_0}$ contains each cube of g just once (see Lemma 3), and second an iterative process (a $\mathbf{g}$-transform, see Lemma 4) where each completed run transforms $\mathcal{P}_t$ into $\mathcal{P}_{t+1}$ with $l(\mathcal{P}_t) \geq l(\mathcal{P}_{t+1})$, for $t \geq 0$. Thus the obtained polygonal curve is also complete and contained in g.

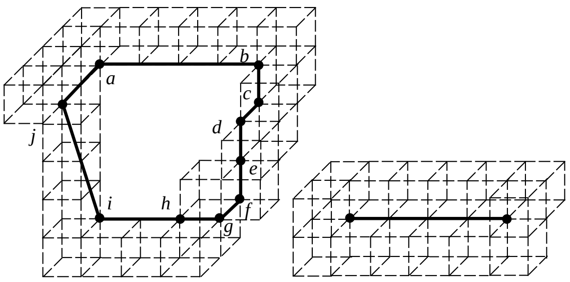

Fig. 7. Curve initializations ('clockwise').

3.1 Initialization

The initial polygonal curve will only connect vertices which are end points of consecutive critical edges. For curve initialization, we scan the given curve until the first pair (e_0, e_1) of consecutive critical edges is found which are not parallel or, if parallel, not in the same grid layer (see Fig. 2 (right) for a non-simple cube-curve showing that searching for a pair of non-coplanar edges would be insufficient in this case). For such a pair (e_0, e_1) we start with vertices (p_0, p_1), p_0 bounds e_0 and p_1 bounds e_1, specifying a line segment p_0p_1 of minimum length (note that such a pair (p_0, p_1) is not always uniquely defined). This is the first line segment of the desired initial polygonal curve $\mathcal{P}_0$.

Now assume that $p_{i-1}p_i$ is the last line segment on this curve $\mathcal{P}_0$ specified so far, and p_i is a vertex which bounds e_i. Then there is a uniquely specified vertex p_{i+1} on the following critical edge e_{i+1} such that p_ip_{i+1} is of minimum length. Length zero is possible with $p_{i+1} = p_i$; in this case we skip p_{i+1}, ie. we do not increase the value of i. Note that this line segment p_ip_{i+1} will always be included in the given tube because the centers of all cubes between two consecutive critical edges are colinear. The process stops by connecting p_n on edge e_n with p_0 (note that it is possible that a minimum-distance criterion for this final step may actually prefer a line between p_n and the second vertex bounding e_0, ie. not p_0). See Table 1 for a list of calculated vertices for the cube-curve on the left in Figs. 2 and 4. The first row lists all the critical edges shown in Fig. 2. The second row contains the vertices of the initial polygon shown in Fig. 4 (initialization =

critical edge	1	2/3	4	5	6/7	8	9	10	11	12/13
1st run (initialization)	a	b	c	d	e	f	g	h	i	j
2nd run	a	b	D	D	e	D	D	h	i	j

Table 1. Calculated points on edges.

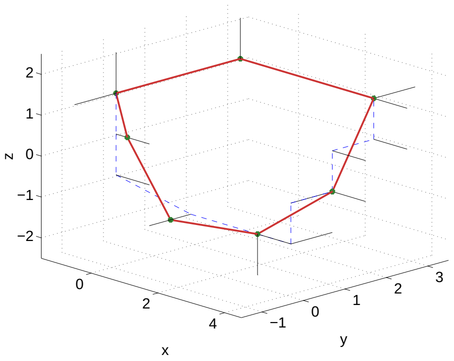

Fig. 8. Assume the initialization starts below on the left. Then the final step of the initialization process would prefer the second vertex of the first edge.

first run of the algorithm). For example, vertex b is on edge 2 and also on edge 3, so there is merely one column for (2/3) for these edges.

This initialization process calculates a polygonal curve $\mathcal{P}_0$ which is always contained and complete in the given tube. Note that traversals following opposite orientations or starting at different critical edges may lead to different initial polygons. For example, a 'counterclockwise' scan of the cube-curve shown in Fig. 2 (left), starting at edge 1, selects edges 11 and 10 to be the first pair of consecutive critical edges, and the generated 'counterclockwise' polygon would differ from the one shown in Fig. 4. Figure 8 shows a curve where the defined initialization does not return to the starting vertex.

Initialization results are shown in Fig. 7, and the curve on the right is already an MLP for this non-simple cube-curve. In case of flat cube-curves the process will fail to determine the specified first pair of critical edges, and in this case a 2D algorithm may be used to calculate the MLP of a corresponding square-curve.

3.2 Iteration steps

In this iterative procedure we move pointers addressing three consecutive vertices of the (so far) calculated polygonal curve around the curve, until a completed run $t+1$ does only lead to an improvement which is below an a-priori threshold τ i.e. $l(\mathcal{P}_t) - \tau < l(\mathcal{P}_{t+1})$. We cannot wait until there is no change at all since in some cases the algorithm would take infinite time to converge. However, in all of our experiments the algorithm converges fast for a practically reasonable value of τ.

Assume a polygonal curve $\mathcal{P}_t = (p_0, p_1, ..., p_m)$, and three pointers addressing vertices at positions $i-1$, i, and $i+1$ in this curve. There are three different options that may occur which define a specific $\mathbf{g}$-transform.

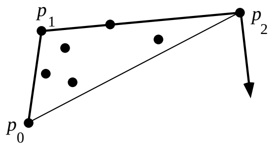

Fig. 9. Intersection points with edges.

($\mathbf{O_1}$) Point p_i can be deleted iff $p_{i-1}p_{i+1}$ is a line segment within the tube. Then subsequence (p_{i-1}, p_i, p_{i+1}) is replaced in our curve by (p_{i-1}, p_{i+1}). In this case we continue with vertices $p_{i-1}, p_{i+1}, p_{i+2}$.

($\mathbf{O_2}$) The closed triangular region $\triangle(p_{i-1}p_ip_{i+1})$ intersects more than just the three critical edges of p_{i-1}, p_i and p_{i+1} (see Fig. 9), ie. simple deletion of p_i would not be sufficient anymore. This situation is solved by calculating a convex arc (note: a convex polygon is the shortest curve encircling a given finite set of planar points [1]) and by replacing point p_i by the sequence of vertices q_1,...,q_k on this convex arc between p_{i-1} and p_{i+1} iff the sequence of line segments $p_{i-1}q_1, \ldots, q_kp_{i+1}$ lies within the tube. Because the vertices are ordered we may use a fast linear-time convex hull routine in case of ($\mathbf{O_2}$). Barycentric coordinates with basis $\{p_{i-1}, p_i, p_{i+1}\}$ may be used to decide which of the intersection points is inside the triangle or not. [3] In this case we continue with a triple of vertices starting with the calculated new vertex q_k. If ($\mathbf{O_1}$) and ($\mathbf{O_2}$) do not lead to any change, the third option may lead to an improvement.

($\mathbf{O_3}$) Point p_i may be moved on its critical edge to obtain an optimum position p_{new} minimizing the total length of both line segments $p_{i-1}p_{new}$ and $p_{new}p_{i+1}$. That's a *move on a critical edge* and an $\mathcal{O}(1)$ solution is given below. Then subsequence (p_{i-1}, p_i, p_{i+1}) is replaced in our curve by $(p_{i-1}, p_{new}, p_{i+1})$. In this case we continue with vertices $p_{new}, p_{i+1}, p_{i+2}$.

In Table 1 we sketch the fact that the second run starts with the polygonal curve $\mathcal{P}_1 = (a, b, c, d, e, f, g, h, i, j)$. For the first triple (a, b, c) we have: none of the cases ($\mathbf{O_1}$) to ($\mathbf{O_3}$) leads to a better location of b. For triple (b, c, d) we delete c according to ($\mathbf{O_1}$) (symbol 'D'). Then triple (b, d, e) leads to the deletion of d, etc., finally (j, a, b) does not delete or move a. In the following round nothing changes.

There are no other possibilities besides ($\mathbf{O_1}$)...($\mathbf{O_3}$). The process stops if one run has not led to a 'significant modification' defined by a threshold τ. In our experiments we chose $\tau = l(\mathcal{P}_t) \cdot 10^{-6}$.

Figures 10 and 11 show the initial polygon $\mathcal{P}_1$ dashed. The solid line represents the final polygon. The short line segments are the critical edges of the given tube.

[3] In the majority of such cases we found $k = 1$, i.e. p_i is replaced by q_1.

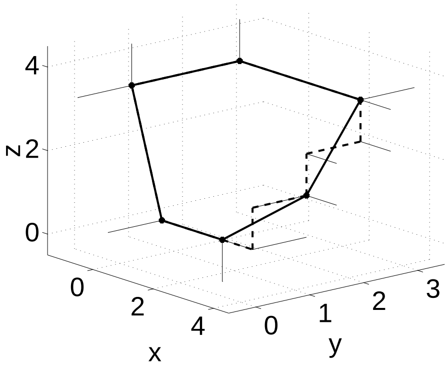

Fig. 10. Initial polygon (dashed) and MLP. Critical edges are shown as short line segments. The rest of the tube is not shown.

4 Move on a critical edge

We consider situation $(\mathbf{O_3})$, ie. p_i lies on a critical edge, say e, and is not colinear with $p_{i-1}p_{i+1}$. Let l_e be the line containing the edge e. First, we find the point $p_{opt} \in l_e$ such that

$$|p_{opt} - p_{i-1}| + |p_{i+1} - p_{opt}| = \min_{p \in l_e}(|p - p_{i-1}| + |p_{i+1} - p|).$$

If p_{opt} lies on the closed critical edge e we simply replace p_i by p_{opt}. If it does not replace p_i by that vertex bounding e and lying closest to p_{opt}.

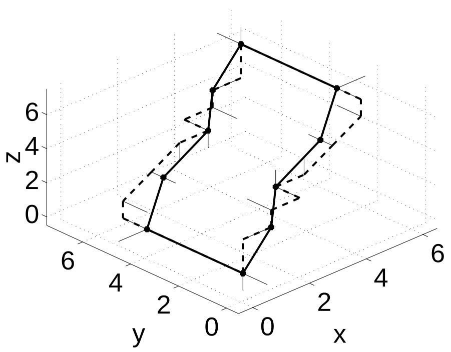

Fig. 11. Initial polygon (dashed) and MLP.

We describe how the point p_{opt} can be found. The set of all points $p \in \mathbb{R}^3$ for which

$$|p - p_{i-1}| + |p_{i+1} - p| = k \;, \tag{1}$$

for some real constant $k \geq |p_{i+1} - p_{i-1}|$ is a prolate spheroid $\mathcal{S}$, ie. an ellipsoid, the two shorter principle axes of which have the same length. In the following we use three Cartesian coordinate systems of $\mathbb{R}^3$ which are interrelated by rigid motions.

(**I**) The coordinates in the coordinate system used up to now are denoted by lower case letters.

(**II**) Coordinates in the coordinate system attached to $\mathcal{S}$ are denoted by upper case letters $(X, Y, Z) \in \mathbb{R}^3$. In these coordinates $\mathcal{S}$ is represented by

$$\frac{X^2}{c^2 + \varepsilon} + \frac{Y^2 + Z^2}{\varepsilon} - 1 = 0, \tag{2}$$

with $c := |p_{i+1} - p_{i-1}|/2$ and $\varepsilon := k^2/4 - c^2$.

(**III**) The coordinate system (x', y', z') is chosen such that l_e is identical to the z'-axis.

Depending upon the choice of ε, the line l_e has two intersection points with $\mathcal{S}$, is a tangent to $\mathcal{S}$, or does not share any point with $\mathcal{S}$. In the case of two intersection points q_1 and q_2 all the points $q \in l_e$ which lie between q_1 and q_2 lie inside $\mathcal{S}$. Thus,

$$\begin{aligned} |q_1 - p_{i-1}| + |p_{i+1} - q_1| &= |q_2 - p_{i-1}| + |q_{i+1} - q_2| \\ &> |q - p_{i-1}| + |p_{i+1} - q| \;, \end{aligned}$$

and neither q_1 nor q_2 is the point p_{opt} which we are looking for. However, if l_e is tangent to $\mathcal{S}$, the tangent point is p_{opt}, since there is no point $q \in l_e$ lying inside $\mathcal{S}$. Thus, we first have to find the value of ε such that l_e is tangent to $\mathcal{S}$. Afterwards we identify the intersection point of l_e and $\mathcal{S}$ as p_{opt}.

Representing (2) in coordinate system (**III**) and intersecting $\mathcal{S}$ with l_e by setting $x' = y' = 0$ yields a quadric equation in z':

$$a(\varepsilon){z'}^2 + b(\varepsilon)z' + c(\varepsilon) = 0 \;. \tag{3}$$

The coefficients a and b contain ε linearly, while c is quadratic in ε. The constraint that l_e be tangent to $\mathcal{S}$ is identical to the condition that (3) has a double solution. This is the case iff

$$a(\varepsilon)c(\varepsilon) - b^2(\varepsilon) = 0 \;. \tag{4}$$

Equation (4) is cubic in ε and has one real positive and two real negative solution. Since geometrically only the positive solution makes sense, we have a unique solution. Replacing ε into (3), solving that equation, and representing the solution in coordinate system (**I**) yields p_{opt}.

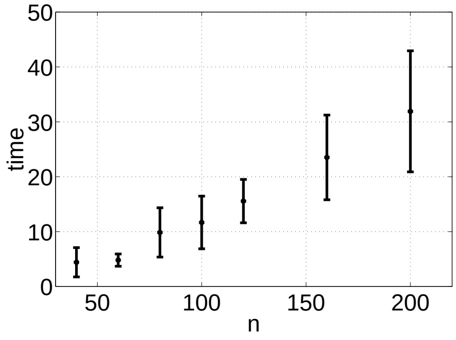

Fig. 12. CPU-time in seconds in dependence of the number of cubes (n)

5 Time complexity and convergence of algorithm

We give an estimate of the complexity of the rubber-band algorithm in dependency of the number of cubes n of the given cube-curve.

The algorithm completes each run in $\mathcal{O}(n)$ time. The described move of point p_i on a critical edge requires constant time. The given value of τ ensured that the measured time complexity is in $\mathcal{O}(n)$, ie. the number of runs does not depend upon n. Figures 12 and 13 show the time needed until the algorithm stops in dependency of the number of cubes n and in dependency of the number of critical edges, respectively. The test set contained 70 randomly generated simple cube-curves. In Fig. 12 each error bar shows the mean convergence time and the standard deviation for a set of 10 digital curves.

The algorithm is iterative and since by definition $l(\mathcal{P}_{t+1}) < l(\mathcal{P}_t)$ (otherwise, if equal, the algorithm stops) and there exists a lower bound for $l(\mathcal{P}_t)$ (namely

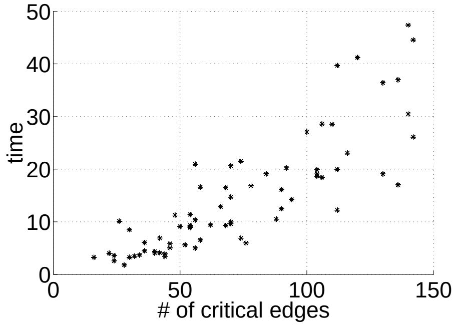

Fig. 13. CPU-time in seconds in dependence of the number of critical edges

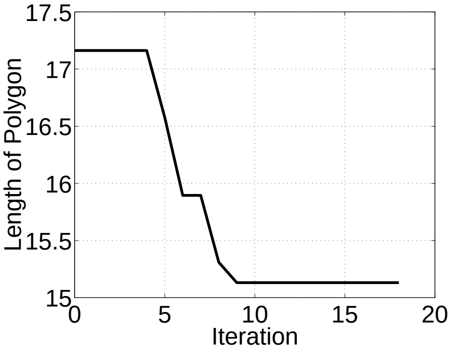

Fig. 14. Length convergence for the curve shown in Fig. 10

the length of the MLP) the algorithm converges. However, the open problem remains: convergent towards which polygon? In all of our experiments it converges towards the MLP. Lemma 3 and Lemma 4 give a partial answer: always towards a polygon complete and contained in **g**.

Figures 14 and 15 show two examples of measured changes in the length of the polygons within the iterative process (in this diagram: 'iteration' = number of runs so far). The simple cube-curves are those from Fig. 10 and Fig. 11.

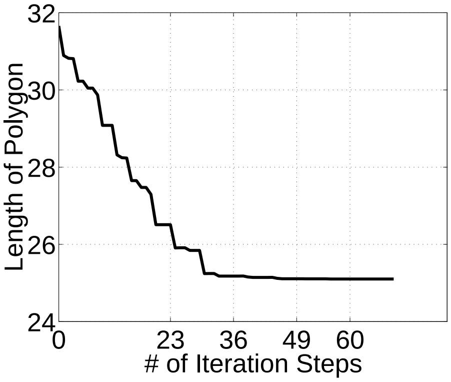

Fig. 15. Length convergence for the curve shown in Fig. 11. The tick-marks on the iteration axis are set after each completed run.

6 Conclusions

The given algorithmic solution provides a polygonal approximation and length measurement for simple cube-curves in 3D space with a measured time in $\mathcal{O}(n)$. Actually it was successfully used for a wider class of cube-curves also allowing cases such as the cube-curve on the right in Fig. 1: cube-curves g where each cube in g has exactly two bounding faces in g. This definition allows a simpler test of a given cube-curve, and also a simpler generation of test examples.

The curve used in the initialization step **(A)** might be replaced by other curves such as sequences of straight segments between midpoints of consecutive cubes, or between midpoints of consecutive critical edges. These two curves would also be curves complete and contained in a given tube of a simple cube-curve. However, the initial curve as defined in our algorithm leads to a much faster convergence in general compared to these two other options.

Tow open problems are stated: the time complexity might be provable always in $\mathcal{O}(n)$, and the convergence might be provable always towards the MLP.

Acknowledgment

This paper was written during a stay of the second author at UWA Nedlands, Perth. The support of UWA (Gledden scholarship) is acknowledged as well as discussions with *Lyle Noakes* and *Ryszard Kozera* to the subject of this paper. The paper was initiated during a stay of the second author at Christian Albrechts University, Kiel, and the support of this university is acknowledged as well.

References

1. R. Busemann and W. Feller. Krümmungseigenschaften konvexer Flächen. *Acta Mathematica*, 66:27–45, 1935.
2. R. Klette. M-dimensional cellular spaces. Tech. Report TR-1256, Univ. of Maryland, Computer Science Dep., March 1983.
3. R. Klette, V. Kovalevsky, and B. Yip. On the length estimation of digital curves. *SPIE Conference Proceedings Vision Geometry VIII*, 3811:52–63, 1999.
4. V. Kovalevsky. Finite topology and image analysis. *Advances in Electronics and Electron. Physics*, 84:197–259, 1992.
5. J.B. Listing: Der Census räumlicher Complexe oder Verallgemeinerungen des Euler'schen Satzes von den Polyëdern. Abhandlungen der Mathematischen Classe der Königlichen Gesellschaft der Wissenschaften zu Göttingen **10** (1861 and 1862) 97–182.
6. W. Rinow. *Topologie.* Deutscher Verlag der Wissenschaften, Berlin, 1975.
7. A. Rosenfeld. *Picture Languages.* Academic Press, New York, 1979.
8. F. Sloboda and Ľ. Bačík. *On one–dimensional grid continua in R^3*. Report of the Institute of Control Theory and Robotics, Bratislava, 1996.
9. F. Sloboda, B. Zaťko, and R. Klette. On the topology of grid continua. *SPIE Conference Proceedings Vision Geometry VII*, 3454:52–63, 1998.
10. A.W. Tucker. An abstract approach to manifolds. *Annals of Math.*, 34:191–243, 1933.
11. K. Voss. *Discrete Images, Objects, and Functions in Z^n*. Springer, Berlin, 1993.