# Shortest Paths in a Cuboidal World

Fajie Li and Reinhard Klette

Computer Science Department, The University of Auckland
Auckland, New Zealand

**Abstract.** Since 1987 it is known that the Euclidean shortest path problem is NP-hard. However, if the 3D world is subdivided into cubes, all of the same size, defining obstacles or possible spaces to move in, then the Euclidean shortest path problem has a linear-time solution, if all spaces to move in form a simple cube-curve. The shortest path through a simple cube-curve in the orthogonal 3D grid is a minimum-length polygonal curve (MLP for short). So far only one general and linear (only with respect to measured run times) algorithm, called the *rubberband algorithm*, was known for an approximative calculation of an MLP. The algorithm is basically defined by moves of vertices along critical edges (i.e., edges in three cubes of the given cube-curve). A proof, that this algorithm always converges to the correct MLP, and if so, then always (provable) in linear time, was still an open problem so far (the authors had successfully treated only a very special case of simple cube-curves before). In a previous paper, the authors also showed that the original rubberband algorithm required a (minor) correction.
This paper finally answers the open problem: by a further modification of the corrected rubberband algorithm, it turns into a provable linear-time algorithm for calculating the MLP of any simple cube-curve.
The paper also presents an alternative provable linear-time algorithm for the same task, which is based on moving vertices within faces of cubes.
For a disticntion, we call the modified original algorithm now the *edge-based rubberband algorithm*, and the second algorithm is the *face-based rubberband algorithm*; the time complexity of both is in $\mathcal{O}(m)$, where $m$ is the number of critical edges of the given simple cube-curve.

## 1 Introduction

A cube-curve $g$ is a loop of face-connected grid cubes in the 3D orthogonal grid; the union $\mathbf{g}$ of those cubes defines the *tube* of $g$. The paper discusses Euclidean shortest paths in such tubes, which are defined by minimum-length polygonal (MLP) curves (see Figure 1).

The Euclidean shortest path problem is as follows: Given a Euclidean space which contains (closed) polyhedral obstacles; compute a path which *(i)* connects two given points in the space, *(ii)* does not intersect the interior of any obstacle, and *(iii)* is of minimum Euclidean length. This problem (starting with dimension 2) is known to be NP-hard [2].

There are algorithms solving the approximate Euclidean shortest path problem in 3D in polynomial time, see [3]. Shortest paths or path planning in todays

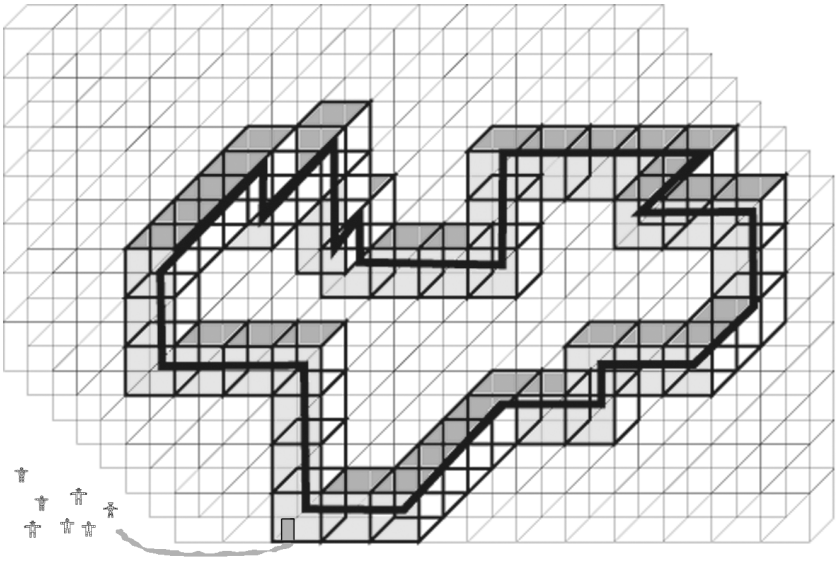

**Fig. 1.** A cuboidal world: seven robots at the lower left corner, and the bold curve is an initial guess for a 3D walk (or flight) through the given loop of shaded cubes. The length of the 3D walk needs to be minimized, which defines the MLP.

3D robotics (see, for example, [15], or the annual ICRA conferences in general) seems to be dominated by heuristics rather than by general geometric algorithms. If a cuboidal world can be assumed then this paper provides two general and linear-time shortest path algorithms.

3D MLP calculations generalize MLP computations in 2D; see, for example, [8, 17] for theoretical results and [5, 21] for 2D robotics scenarios. Shortest curve calculations in image analysis also use graph metrics instead of the Euclidean metric; see, for example, [20].

Interest in 3D MLPs was also raised by the issue of multigrid-convergent length estimation for digitized curves. The length of a simple cube-curve in 3D Euclidean space can be defined by that of the MLP; see [18, 19], which can be characterized as a 'global approach'. A 'local approach' for 3D length estimation, allowing only weighted steps within a restricted neighborhood, was considered in [7]. Alternatively to the MLP, the length of 3D digital curves can also be measured (within time, linear in the number of grid points on the curve) based on DSS-approximations [4].

The computation of 3D MLPs was first time published in [1], proposing a 'rubberband' algorithm[1]. This iterative algorithm was experimentally tested and showed linear run-time behavior. It also was correct for all the tested inputs (where correctness was tested manually!). However, in this publication, no math-

[1] Not to be confused with a 2D image segmentation algorithm of the same name [16].

ematical proof was given for linear run time or general correctness (i.e., that its solution iterates, for any simple cube-curve, to the MLP). This *original rubberband algorithm* is also published in the monograph [10]. Recent applications of this algorithm are in 3D medical imaging; see, for example, [6, 22].

The authors approached the correctness and linearity problem of the rubberband algorithm along the following steps:

[11] only considered a very special class of simple cube-curves and developed a provable correct MLP algorithm for this class. The main idea was to decompose a cube-curve of that class into arcs at "end angles" (see Definition 3 in [11]), that means, the cube-curves have to have end-angles, that the algorithm can be applied.

[12] constructed an example of a simple cube-curve whose MLP does not have any of its vertices at a corner of a grid cube. It followed that any of cube-curve with this property does not have any end angle, and this means that we cannot use the MLP algorithm as proposed in [11]. This was the basic importance of the result in [12]: we showed the existence of cube-curves which require further algorithmic studies.

[14] showed that the original rubberband algorithm requires a modification (in its Option 3) to guarantee that calculated curves are always contained in the tube $\mathbf{g}$. This *corrected rubberband algorithm* achieves (as the original rubberband algorithm) minimization of length by moving vertices along *critical edges* (i.e., grid edges incident with three cubes of the given simple cube-curve).

This paper now (finally) extends the corrected rubberband algorithm into the *edge-based rubberband algorithm* and shows, that it is correct for any (!) simple cube-curve. The paper also presents a totally new algorithm, the *face-based rubberband algorithm*, and shows that it is also correct for any simple cube-curve. We prove that both, the edge-based and the face-based rubberband algorithm, have time complexity in $\mathcal{O}(m)$ time, where $m$ is the number of critical edges in the given simple cube-curve.

Further (say, 'more elegant') algorithms for calculating MLPs in simple cube-curves may exist; this way this article may be just the starting point for more detailed performance evaluations. Also, the given modifications of the original rubberband algorithm might be not always necessary, or the simplest ones.

The paper is organized as follows: Section 2 describes the concepts used in this paper. Section 3 provides mathematical fundamentals for our two algorithms. Section 4 describes the edge-based and face-based rubberband algorithm, and discusses their time complexity. Section 5 presents an example illustrating how the edge-based and face-based rubberband algorithms are converging to identical results (i.e., to the $MLP$). Section 6 gives our conclusions.

## 2 Definitions

Following [10], a grid point $(i, j, k) \in \mathbb{Z}^3$ 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* at its corners. *Cells* are either cubes, faces, edges, or vertices. The

intersection of two cells is either empty or a joint *side* of both cells. A *cube-curve* is an alternating sequence $g = (f_0, c_0, f_1, c_1, \ldots, f_n, c_n)$ 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 $|i-k| \geq 2 \pmod{n+1}$, if $c_i \bigcap c_k \neq \phi$ then either $|i-k| = 2 \pmod{n+1}$ and $c_i \bigcap c_k$ is an edge, or $|i-k| \geq 3 \pmod{n+1}$ and $c_i \bigcap c_k$ is a vertex.

A *tube* $\mathbf{g}$ is the union of all cubes contained in a cube-curve $g$. A tube is a compact set in $\mathbb{R}^3$; its frontier defines a polyhedron. A curve in $\mathbb{R}^3$ is *complete* in $\mathbf{g}$ iff it has a nonempty intersection with every cube contained in $g$. Following [18, 19], we define:

**Definition 1.** *A* minimum-length curve *of a simple cube-curve $g$ is a shortest simple curve $P$ which is contained and complete in tube $\mathbf{g}$. The length $\mathcal{L}(g)$ of $g$ is defined to be the length $\mathcal{L}(P)$.*

It turns out that such a shortest curve $P$ is always a polygonal curve, called MLP for short; it is uniquely defined if the cube-curve is not contained in a single layer of cubes of the 3D grid (see [18, 19]). If it is contained in just one layer then the MLP is uniquely defined up to a translation orthogonal to that layer. We speak about *the* MLP of a simple cube-curve.

Figure 2 shows a simple-cube curve and (as bold polygonal curve) its MLP; grid edges containing vertices of the MLP are also shown in bold.

**Definition 2.** *A critical edge of a cube-curve $g$ is a grid edge which is incident with exactly three different cubes contained in $g$. If $e$ is a critical edge of $g$ and $l$ is a straight line such that $e \subset l$, then $l$ is called a* critical line of $e$ in $g$ *or* critical line *for short. If $f$ is a face of a cube in $g$ and one of $f$'s edges is a critical edge $e$ in $g$ then $f$ is called a* critical face of $e$ in $g$ *or* critical face *for short.*

**Definition 3.** *A simple cube-curve $g$ is called* first-class *iff each critical edge of $g$ contains exactly one vertex of the MLP of $g$.*

Figure 3 shows a first-class simple cube-curve. The cube-curve shown in Figure 2 is not first-class because there are no vertices of the $MLP$ on the following critical edges: $e_1$, $e_4$, $e_5$, $e_6$, $e_8$, $e_9$, $e_{10}$, $e_{11}$ and $e_{14}$ (will be later shown in the experiments, summarized in Table 4).

Unfortunately, we need also a few rather technical definitions:

**Definition 4.** *Let $e$ be a critical edge of a simple cube-curve $g$ and $f_1$, $f_2$ be two critical faces of $e$ in $g$. Let $c_1$, $c_2$ be the centers of $f_1$, $f_2$ respectively. Then a polygonal curve can go in the direction from $c_1$ to $c_2$, or from $c_2$ to $c_1$, to visit all cubes in $g$ such that each cube is visited exactly once. If $e$ is on the left of line segment $c_1c_2$, then the orientation from $c_1$ to $c_2$ is called* counter-clockwise orientation *of $g$. $f_1$ is called* the first *critical face of $e$ in $g$. If $e$ is on the right of line segment $c_1c_2$, then the direction from $c_1$ to $c_2$ is called* clockwise orientation *of $g$.*

Figure 2 shows all critical edges ($e_0$, $e_1$, $e_2$, ..., $e_{18}$) and their first critical faces ($f_0$, $f_1$, $f_2$, ..., $f_{18}$) of a simple cube-curve $g$.

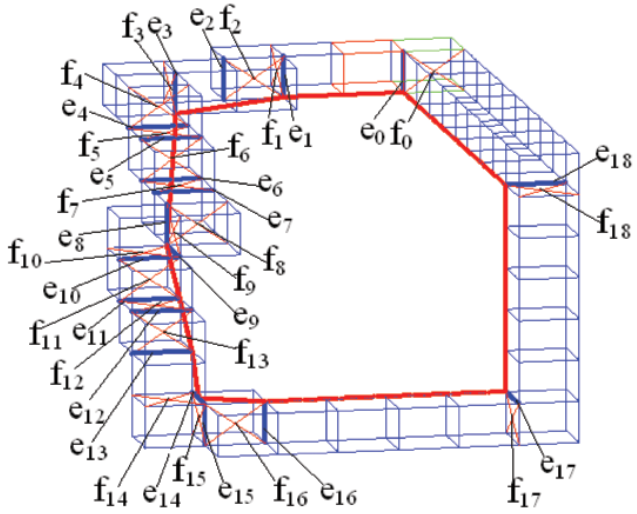


**Fig. 2.** A simple cube-curve and its $MLP$ (see also Table 1).

| Critical edge | $x_{i1}$ | $y_{i1}$ | $z_{i1}$ | $x_{i2}$ | $y_{i2}$ | $z_{i2}$ |
|---|---|---|---|---|---|---|
| $e_0$ | -0.5 | 1 | -0.5 | -0.5 | 1 | 0.5 |
| $e_1$ | -0.5 | 2 | -0.5 | -0.5 | 2 | 0.5 |
| $e_2$ | -1.5 | 3 | -0.5 | -1.5 | 3 | 0.5 |
| $e_3$ | -2.5 | 3 | -0.5 | -2.5 | 4 | -0.5 |
| $e_4$ | -3.5 | 3 | -0.5 | -3.5 | 4 | -0.5 |
| $e_5$ | -3.5 | 3 | -1.5 | -3.5 | 4 | -1.5 |
| $e_6$ | -4.5 | 3 | -1.5 | -4.5 | 4 | -1.5 |
| $e_7$ | -5.5 | 4 | -2.5 | -5.5 | 4 | -1.5 |
| $e_8$ | -6.5 | 4 | -2.5 | -5.5 | 4 | -2.5 |
| $e_9$ | -6.5 | 4 | -2.5 | -6.5 | 5 | -2.5 |
| $e_{10}$ | -6.5 | 4 | -3.5 | -6.5 | 5 | -3.5 |
| $e_{11}$ | -7.5 | 4 | -3.5 | -7.5 | 5 | -3.5 |
| $e_{12}$ | -7.5 | 4 | -4.5 | -7.5 | 5 | -4.5 |
| $e_{13}$ | -8.5 | 4 | -5.5 | -7.5 | 4 | -5.5 |
| $e_{14}$ | -8.5 | 4 | -6.5 | -8.5 | 4 | -5.5 |
| $e_{15}$ | -8.5 | 3 | -6.5 | -8.5 | 3 | -5.5 |
| $e_{16}$ | -9.5 | -1 | -5.5 | -8.5 | -1 | -5.5 |
| $e_{17}$ | -8.5 | -2 | -0.5 | -8.5 | -1 | -0.5 |
| $e_{18}$ | -0.5 | -1 | -0.5 | -0.5 | -1 | 0.5 |

**Table 1.** Coordinates of endpoints of critical edges shown in Figure 2 (also used later in an experiment).

**Definition 5.** *A* minimum-length pseudo polygon *of a simple cube-curve $g$, denoted by $MLPP$, is a shortest curve $P$ which is contained and complete in tube $\boldsymbol{g}$ such that each vertex of $P$ is on the first critical face of a critical edge in $g$.*

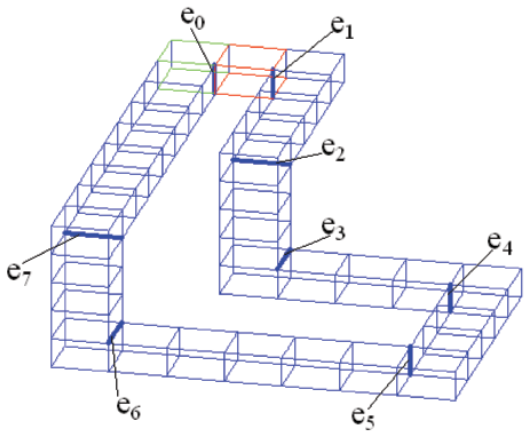


**Fig. 3.** A first-class simple cube-curve.

From results in [19] it follows that the $MLPP$ of a simple cube-curve $g$ is unique. The number of vertices of an $MLPP$ is the number of all critical edges of $g$. $p_{4_0}p_{4_1}\cdots p_{4_{18}}$ (see Table 3) is the $MLPP$ of $g$ as shown in Figure 2.

Let $f_{i1}$, $f_{i2}$ be two critical faces of $e_i$ in $g$, $i = 1, 2$. Let $c_{i1}$, $c_{i2}$ be the centers of $f_{i1}$, $f_{i2}$ respectively, for $i = 1, 2$. Obviousely, the counter-clockwise orientation of $g$ defined by $c_{11}$, $c_{12}$ is identical to the one defined by $c_{21}$, $c_{22}$.

**Definition 6.** *Let $e_0$, $e_1$, $e_2$, ... $e_m$ and $e_{m+1}$ be all consecutive critical edges of $g$ in the counter-clockwise orientation of $g$. Let $f_i$ be the first critical face of $e_i$ in $g$, and $p_i$ be a point on $f_i$, where $i$ = 0, 1, 2, ..., $m$ or $m + 1$. Then the polygonal curve $p_0p_1\cdots p_mp_{m+1}$ is called an* approximate minimum-length pseudo polygon *of $g$, denoted by $AMLPP$.*

The polygonal curve $p_{1_0}p_{1_1}\cdots p_{1_{18}}$ (see Table 2) is an $AMLPP$ of $g$ shown in Figure 2.

**Definition 7.** *Let $p_1$, $p_2$ and $p_3$ be three consecutive vertices of an $AMLPP$ of a simple cube-curve $g$. If $p_1$, $p_2$ and $p_3$ are colinear, then $p_2$ is called a* trivial *vertex of the $AMLPP$ of $g$. $p_2$ is called a* non-trivial *vertex of the $AMLPP$ of $g$ if it is not a trivial vertex of that $AMLPP$ of $g$.*

A simple *cube-arc* is an alternating sequence $a = (f_0, c_0, f_1, c_1, \ldots, f_k, c_k, f_{k+1})$ of faces $f_i$ and cubes $c_i$ with $f_{k+1} \neq f_0$, denoted by $a = (f_0, f_1, \ldots, f_{k+1})$ or $a(f_0, f_{k+1})$ for short, which is a consecutive part of a simple cube-curve. A subarc of an arc $a = (f_0, f_1, \ldots, f_{k+1})$ is an arc $(f_i, f_{i+1}, \ldots, f_j)$, where $0 \leq i \leq j \leq k$.

**Definition 8.** *Let a polygonal curve $P = p_0p_1\cdots p_mp_{m+1}$ be an $AMLPP$ of $g$ and $p_i \in f_i$, where $f_i$ is a critical face of $g$, $i$ = 0, 1, 2, ..., $m$ or $m + 1$. A cube-arc $\rho = (f_i, f_{i+1}, \ldots, f_j)$ is called*

- *a* (2,3)-cube-arc *with respect to* $P$ *if each vertex* $p_k$ *is identical to* $p_{k-1}$ *or* $p_{k+1}$*, where* $k = i$ *+1, ..., j - 1,* [2]
- *a* maximal (2,3)-cube-arc *with respect to* $P$ *if it is a (2,3)-cube-arc and* $p_i$ *is not identical to* $p_{i+1}$ *and* $p_{i-1}$*, and* $p_j$ *is not identical to* $p_{j-1}$ *and* $p_{j+1}$*,*
- *a* 3-cube-arc unit *with respect to* $P$ *if it is a (2,3)-cube-arc such that* $j = i + 4 \pmod{m + 2}$ *and* $p_{i+1}$, $p_{i+2}$, $p_{i+3}$ *are identical.*
- *a* 2-cube-arc *with respect to* $P$ *if it is a (2,3)-cube-arc and no three consecutive vertices of* $P$ *on a are identical,*
- *a* maximal 2-cube-arc *with respect to* $P$ *if it is both a maximal (2,3)-cube-arc and a 2-cube-arc as well,*
- *a* 2-cube-arc unit *with respect to* $P$ *if it is a 2-cube-arc such that* $j = i + 3 \pmod{m + 2}$ *and* $p_{i+1}$ *is identical to* $p_{i+2}$*,*
- *a* regular cube-arc unit *with respect to* $P$ *if* $a = (f_i, f_{i+1}, f_j)$ *such that* $p_i$ *is not identical to* $p_{i+1}$ *and* $p_j$ *is not identical to* $p_{i+1}$*,*
- *a* cube-arc unit *with respect to* $P$ *if a is a regular cube-arc unit, 2-cube-arc unit or 3-cube-arc unit, or*
- *a* regular cube-arc *with respect to* $P$ *if no two consecutive vertices of* $P$ *on a are identical.*

Let $P_{18_i} = p_{i_0} p_{1_1} \cdots p_{1_{18}}$ (see Table 2), where $i = 1, 2, 3, 4$. Then there are four maximal 2-cube-arcs with respect to $P_{18_i}$: $(p_{i_{18}}, p_{i_0}, p_{i_1}, p_{i_2})$, $(p_{i_2}, p_{i_3}, p_{i_4}, p_{i_5})$, $(p_{i_7}, p_{i_8}, p_{i_9}, p_{i_{10}})$ and $(p_{i_{12}}, p_{i_{13}}, p_{i_{14}}, p_{i_{15}})$ in total, where $i = 1, 2, 3$. They are also maximal 2-cube-arcs and 2-cube-arc units with respect to $P_{18_i}$, where $i = 1, 2, 3$. There are no 3-cube-arc units with respect to $P_{18_i}$, where $i = 1, 2, 3$. $(p_{i_1}, p_{i_2}, p_{i_3})$ is a regular cube-arc unit with respect to $P_{18_i}$ and $(p_{i_4}, p_{i_5}, p_{i_6}, p_{i_7}, p_{i_8})$ is a regular cube-arc with respect to $P_{18_i}$, where $i = 1, 2, 3$.

There are three maximal 2-cube-arcs with respect to $P_{18_4}$: $(p_{4_{18}}, p_{4_0}, p_{4_1}, p_{4_2})$, $(p_{4_2}, p_{4_3}, p_{4_4}, p_{4_5})$, and $(p_{4_{12}}, p_{4_{13}}, p_{4_{14}}, p_{4_{15}})$ in total. They are also maximal 2-cube-arcs and 2-cube-arc units with respect to $P_{18_4}$. $(p_{4_6}, p_{4_7}, p_{4_8}, p_{4_9}, p_{4_{10}}, p_{4_{11}}, p_{4_{12}})$ is a (2,3)-cube-arc with respect to $P_{18_4}$. $(p_{4_6}, p_{4_7}, p_{4_8}, p_{4_9}, p_{4_{10}})$ is a unique 3-cube-arc unit with respect to $P_{18_4}$.

**Definition 9.** *Let* $\rho = (f_i, f_{i+1}, \ldots, f_j)$ *be a simple cube-arc and* $p_k \in f_k$*, where* $k = i, j$*. A* minimum-length arc *with respect to* $p_i$ *and* $p_j$ *of* $\rho$*, denoted by* $MLA(p_i, p_j)$*, is a shortest arc (from* $p_i$ *to* $p_j$*) which is contained and complete in* $\rho$ *such that each vertex of* $MLA(p_i, p_j)$ *is on the first critical face of a critical edge in* $\rho$*.*

## 3 Basics

We provide mathematical fundamentals to be used in the following. We start with citing a theorem from [9]:

**Theorem 1.** *Let* $g$ *be a simple cube-curve. Critical edges are the only possible locations of vertices of the MLP of* $g$*.*

[2] Note that it is impossible that four consecutive vertices of $P$ on $\rho$ are identical.

Let $d_e(p, q)$ be the Euclidean distance between points $p$ and $q$.

Let $e_0$, $e_1$, $e_2$, ..., $e_m$ and $e_{m+1}$ be $m+2$ consecutive critical edges in a simple cube-curve $g$, and let $l_0$, $l_1$, $l_2$, ..., $l_m$ and $l_{m+1}$ be the corresponding critical lines. We express a point $p_i(t_i) = (x_i + k_{x_i}t_i, y_i + k_{y_i}t_i, z_i + k_{z_i}t_i)$ on $l_i$ in general form, with $t_i \in \mathbb{R}$, where $i = 0, 1, \ldots,$ or $m + 1$.

Let $e_i$, $e_j$, and $e_k$ be three (not necessarily consecutive) critical edges in a simple cube-curve.

**Lemma 1.** ([11], Lemma 1) *Let $d_j(t_i, t_j, t_k) = d_e(p_i, p_j) + d_e(p_j, p_k)$. It follows that $\frac{\partial^2 d_j}{\partial {t_j}^2} > 0$.*

By elementary geometry, we also have:

**Lemma 2.** *Let $P$ be a point in $\triangle ABC$ such that $P$ is not on any of the three line segments $AB, BC$ and $CA$. Then $d_{PA} + d_{PB} < d_{CA} + d_{CB}$.*

The following Lemma is straightforward but useful in our description of the edge-based rubberband algorithm in Section 4.

**Lemma 3.** *Let $p_i$ and $p_{i+1}$ be two consecutive vertices of an $AMLPP$ of $g$. If $p_i$ is identical to $p_{i+1}$ then $p_i$ and $p_{i+1}$ are on a critical edge of $g$.*

**Lemma 4.** ([13], Lemma 4) *The number of $MLPP$s of a first-class simple cube-curve $g$ is finite.*

Let $p_i \in f_i$, where $f_i$ is the first critical face of $e_i$ in $g$, $i = 0, 1, 2, \ldots, m$ or $m + 1$. Let $P$ be a polygonal curve $p_0p_1 \cdots p_mp_{m+1}$ .

**Corollary 1.** *The number of $MLPP$s of a simple cube-curve $g$ is finite.*

*Proof.* If there is a vertex $p_i \in f_i$ such that $p_i$ is not on an edge of $f_i$ (i.e, $p_i$ is a trivial vertex of $MLPP$), then $p_{i-1}$, $p_i$ and $p_{i+1}$ are colinear. In this case, $p_i$ can be ignored because it is defined by $p_{i-1}$ and $p_{i+1}$. Therefore, without loss generality, we can assume that each $p_i$ is on one edge of $f_i$, where $i = 0, 1, 2, \ldots, m$ or $m+1$. In this case, the proof of this lemma is exactly the same as that of Lemma 4. □

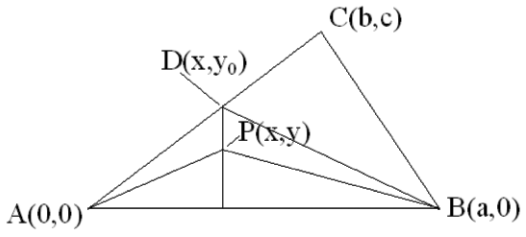


**Fig. 4.** Illustration for the proof of Lemma 2.

**Lemma 5.** ([13], Lemma 14) *Each first-class simple cube-curve $g$ has a unique $MLPP$.*

Analogous to the proof of Lemma 5, we also have

**Lemma 6.** *Each simple cube-curve $g$ has a unique $MLPP$.*

**Theorem 2.** *$P$ is an $MLPP$ of $g$ iff for each cube-arc unit $a = (f_i, f_{i+1}, \ldots, f_j)$ with respect to $P$, the arc $(p_i, p_{i+1}, \ldots, p_j)$ is equal to $MLA(p_i, p_j)$.*

*Proof.* The necessarity is straightforward. The sufficiency is by Lemma 6. □

Analogously to the proof of Theorem 1 we also obtain

**Lemma 7.** *If a vertex $p$ of an $AMLPP$ of $g$ is on a first critical face $f$ but not on any edge of it, then $p$ is a trivial vertex of the $AMLPP$.*

## 4 Algorithms

We present two algorithms which are both linear-time and provable convergent to the $MLP$ of a simple cube-curve. We start to describe some useful procedures which will be used in those two algorithms (as subroutines).

### 4.1 Procedures

Given a critical $e$ in $g$, and two points $p_1$ and $p_3$ in $g$, by Procedure 1, we can find a unique point $p_2$ in $f$ such that $d_{p_1p_2} + d_{p_3p_2} = \min\{d_{p_1p} + d_{p_3p} : p \in e\}$.

**Procedure 1**
Let the two endpoints of $e$ be $a$ and $b$. Then by Lemma 6 of [13], $p_2 = a + t * (b - a)$, where $t = -(A_1B_2 + A_2B_1)/(B_2 + B_1)$, $A_1$, $A_2$, $B_1$ and $B_2$ are functions of the coordinates of $p_1$, $p_3$, $a$ and $b$.

Given a critical face $f$ of a critical edge in $g$, and two points $p_1$ and $p_3$ in $g$, by Procedure 2, we can find a point $p_2$ in $f$ such that $d_{p_1p_2} + d_{p_3p_2} = \min\{d_{p_1p} + d_{p_3p} : p \in f\}$.

**Procedure 2**
*Case 1.* $p_1p_3$ and $f$ are on the same plane. *Case 1.1.* $p_1p_3 \cap f \neq \phi$. In this case, $p_1p_3 \cap f$ is a line segment. Let $p_2$ be the end point of this segment such that it is close to $p_1$. *Case 1.2.* $p_1p_3 \cap f = \phi$. By Lemmas 2, $p_2$ must be on the edges of $f$. By Lemmas 1, $p_2$ must be uniquely on one of the edges of $f$. Apply Procedure 1 on the four edges of $f$, denoted by $e_1$, $e_2$, $e_3$ and $e_4$, we get $p_{2i}$ such that $d_{p_1p_{2i}} + d_{p_3p_{2i}} = \min\{d_{p_1p} + d_{p_3p} : p \in e_i\}$, where $i = 1, 2, 3, 4$. Then we can find a point $p_2$ such that $d_{p_1p_2} + d_{p_3p_2} = \min\{d_{p_1p_{2i}} + d_{p_3p_{2i}} : i = 1, 2, 3, 4.\}$.

*Case 2.* $p_1p_3$ and $f$ are not on the same plane. *Case 2.1.* $p_1p_3 \cap f \neq \phi$. It follows that $p_1p_3 \cap f$ is a unique point. Let $p_2$ be this point. *Case 2.2.* $p_1p_3 \cap f = \phi$. In this case, $p_2$ can be found exactly the same way as in Case 1.2.

The following procedure is used to convert an $MLPP$ into an $MLP$.

**Procedure 3**

Given a polygonal curve $p_0p_1 \cdots p_mp_{m+1}$ and three pointers addressing vertices at positions $i$ - 1, $i$, and $i + 1$ in this curve. Delete $p_i$ if $p_{i-1}$, $p_i$ and $p_{i+1}$ are colinear. Next, the subsequence ($p_{i-1}$, $p_i$, $p_{i+1}$) is replaced in the curve by ($p_{i-1}$, $p_{i+1}$). Then, continue with vertices ($p_{i-1}$, $p_{i+1}$, $p_{i+2}$) until $i + 2$ is $m + 1$.

Let $p_i \in l_i \subset f_i$ ,. . ., $p_j \in l_j \subset f_j$ be some consecutive vertices of the $AMLPP$ of $g$, where $f_i$, . . ., $f_j$ are some consecutive critical faces of $g$, and $l_k$ is a line segment on $f_k$, $k = i$, $i + 1$, . . ., $j$. Let $\epsilon = 10^{-10}$ (this value defines the accuracy of the output of this algorithm). We can apply the method of Option 3 of rubberband algorithm (page 967, [1], and see correction in [14]) on cube-arc $\rho = (f_i, f_{i+1}, \ldots, f_j)$ to find an approximate $MLA(p_i, p_j)$ as follows:

**Procedure 4**

1. Calculate the length of arc $p_ip_{i+1} \cdots p_{j-1}p_j$, denoted by $L_1$;
2. Let $k = i$+1;
3. Take two points $p_{k-1} \in f_{k-1}$ and $p_{k+1} \in f_{k+1}$;
4. For line segment $l_k$ on a critical face $f_k$ in $g$, and points $p_{k-1}$ and $p_{k+1}$ on $l_{k-1}$ and $l_{k+1}$, respectively, apply Procedure 1 to find a point $q_k \in l_k$ such that $d_{p_{k-1}q_k} + d_{p_{k+1}q_k} = \min\{d_{p_{k-1}p} + d_{p_{k+1}p} : p \in l_k\}$. Let $p_k = q_k$.
5. $k = k + 1$;
6. If $k = j$, calculate the length of arc $p_ip_{i+1} \cdots p_{j-1}p_j$, denoted by $L_2$.
7. If $L_1$ - $L_2 > \epsilon$, let $L_1 = L_2$ and go to Step 2. Otherwise, output the arc $p_ip_{i+1} \cdots p_{j-1}p_j$.

Let $e_0$, $e_1$, $e_2$, . . . $e_m$ and $e_{m+1}$ be all consecutive critical edges of $g$ in the counter-clockwise orientation of $g$. Let $f_i$ be the first critical face of $e_i$ in $g$, and $c_i$ be the center of $f_i$, where $i = 0$, 1, 2, . . ., $m$ or $m + 1$. All indices of points, edges and faces are taken *mod* $m + 2$. Let $\epsilon = 10^{-10}$. By Procedure 5, we can compute an $AMLPP$ of $g$ and its length.

**Procedure 5**

1. Let $P$ be a polygonal curve $p_0p_1 \cdots p_mp_{m+1}$;
2. Calculate the length of $P$, denoted by $L_1$;
3. Let $i = 0$;
4. Take two points $p_{i-1} \in f_{i-1}$ and $p_{i+1} \in f_{i+1}$;

5. For the critical face $f_i$ of an critical edge $e_i$ in $g$, and points $p_{i-1}$ and $p_{i+1}$ in $f_{i-1}$ and $f_{i+1}$, respectively, apply Procedure 2 to find a point $q_i$ in $f_i$ such that $d_{p_{i-1}q_i} + d_{p_{i+1}q_i} = \min\{d_{p_{i-1}p} + d_{p_{i+1}p} : p \in f_i\}$. Let $p_i = q_i$.

6. $i = i + 1$;

7. If $i = m + 3$, calculate the length of the polygonal curve $p_0p_1 \cdots p_mp_{m+1}$, denoted by $L_2$.

8. If $L_1$ - $L_2 > \epsilon$, let $L_1 = L_2$ and go to Step 2. Otherwise, output the polygonal curve $p_0p_1 \cdots p_mp_{m+1}$ as an $AMLPP$ of $g$ and its length $L_2$.

Given an $n$-cube-arc unit ($f_i$, ..., $f_j$) with respect to a polygonal curve $P$ of $g$, where $n = 2$ or 3. Let $p_i \in f_i$ and $p_j \in f_j$. We can find an $MLA(p_i, p_j)$ by the following procedure.

**Procedure 6**

1. Compute the set $E = \{e$: $e$ is an edge of $f_k$, $k = i + 1, \ldots, j - 1\}$;

2. Let $I = 1$ and $L = 100$;

3. Compute the set $SE = \{S$: $S \subseteq E$ and $|S| = I\}$;

4. Go through each $S \in SE$, input $p_i$, $e_1$, ..., $e_l$, $p_j$ to Procedure 4 to compute an approximate $MLA(p_i, p_j)$ such that it has minimal length with respect to all $S \in SE$, denoted by $AMLA(I, SE)$, where $e_k \in S$, $k = 1, 2, \ldots, l$ and $l = |S|$. If the length of $AMLA(I, SE) < L$, let $MLA(p_i, p_j) = AMLA(I, SE)$ and $L =$ the length of $AMLA(I, SE)$;

5. Let $I = I + 1$.

6. If $I < n$ then go to Step 3. Otherwise, stop.

**Lemma 8.** *For each cube-arc unit $\rho = (f_i, f_{i+1}, \ldots, f_j)$ with respect to $P$, $MLA(p_i, p_j)$ can be computed in $\mathcal{O}(1)$.*

*Proof.* If $\rho$ is a regular cube-arc unit, then $MLA(p_i, p_j)$ can be found by Procedure 2, which has complexity $\mathcal{O}(1)$. Otherwise, $\rho$ is an $n$-cube-arc unit, where $n = 2$ or 3. Then, by Lemma 7, $MLA(p_i, p_j)$ can be found by Procedure 6, which can be computed in $\mathcal{O}(1)$ because $n = 2$ or 3. □

### 4.2 Algorithms

The original rubberband algorithm was published in [1] and slightly corrected in [14]. We now extend this corrected rubberband algorithm into the following (provable correct) algorithm.

**The Edge-Based Rubberband Algorithm**

1. Let $P_0$ be the polygon obtained by the (corrected) rubberband algorithm;

2. Find a point $p_i \in f_i$ such that $p_i$ is the intersection point of an edge of $P_0$ with $f_i$, where $i = 0, 1, 2, \ldots, m$ or $m + 1$. Let $P$ be a polygonal curve $p_0p_1 \cdots p_mp_{m+1}$;

3. Apply Procedure 6 to all cube-arc units of $P$. If for each cube-arc unit $\rho = (f_i, f_{i+1}, \ldots, f_j)$ with respect to $P$, the arc $(p_i, p_{i+1}, \ldots, p_j) = MLA(p_i, p_j)$,

then $P$ is the $MLPP$ of $g$ (by Theorem 2), and go to Step 4. Otherwise, go to Step 3.

4. Apply Procedure 3 to obtain the final $MLP$.

**The Face-Based Rubberband Algorithm**

1. Take a point $p_i \in f_i$, where $i = 0, 1, 2, \ldots, m$ or $m+1$;

2. Apply Procedure 5 to find an $AMLPP$ of $g$, denoted by $P$;

3. Find all maximal 2-cube-arcs with respect to $P$, apply Procedure 4 to update the vertices of the $AMLPP$, which are on one of the 2-cube-arcs. (By Lemma 3, the input line segments of Procedure 4 are critical edges.) Repeat this step until the length of the updated $AMLPP$ is sufficiently accurate ( i.e., previous length minus current length $< \epsilon$);

4. Apply Procedure 5 to update the current $AMLPP$;

5. Find all maximal (2,3)-cube-arcs with respect to the current $P$, apply Procedure 4 to update the vertices of the current $AMLPP$, which are on one of the (2,3)-cube-arcs. The input line segments of Procedure 4 can be found such that they are on the critical face and parallel or perpendicular to the critical edge of the face. Repeat this step until the length of the updated $AMLPP$ is sufficiently accurate;

6. Apply Procedure 5 to update the current $AMLPP$.

7. Apply Procedure 6 to all cube-arc units of $P$. If for each cube-arc unit $\rho = (f_i, f_{i+1}, \ldots, f_j)$ with respect to $P$, the arc $(p_i, p_{i+1}, \ldots, p_j) = MLA(p_i, p_j)$, then $P$ is the $MLPP$ of $g$ (by Theorem 2), and go to Step 8. Otherwise, go to Step 3.

8. Apply Procedure 3 to obtain the final $MLP$.

### 4.3 Computational Complexity

It is obvious that Procedures 1 and 2 can be computed in $\mathcal{O}(1)$, and Procedure 3 can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$. The main operation of Procedure 4 is Step 4, which can be computed in $\mathcal{O}(n)$, where $n$ is the number of points of the arc. Analogously, Procedure 5 can be computed in $\mathcal{O}(m)$, where $m$ is the number of points of the polygonal curve.

[14] has proved that the (corrected) rubberband algorithm can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$. The main additional operation of the edge-based rubberband algorithm is Step 3 which can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$ (by Lemma 8). It follows that the edge-based rubberband algorithm can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$.

For the face-based rubberband algorithm, Step 1 is trivial; Steps 2, 4, 6 have the same complexity as Procedure 5. Step 3 can be computed in $N_1(\epsilon)\mathcal{O}(m)$, where $N_1(\epsilon)$ depends on the accuracy $\epsilon$, where $m$ is the number of points of the polygonal curve. Analogously, Step 5 can be computed in $N_2(\epsilon)\mathcal{O}(m)$, where $N_2(\epsilon)$ depends on the accuracy $\epsilon$ (Note that there is a constant number of different combinations of input line segments of Procedure 4). By Lemma 8, Step 7

can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$. Therefore, the face-based rubberband algorithm can be computed in $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$.

| $p_{1i}$ | $x_{1i}$ | $y_{1i}$ | $z_{1i}$ | $p_{2i}$ | $x_{2i}$ | $y_{2i}$ | $z_{2i}$ |
|---|---|---|---|---|---|---|---|
| $p_{1_0}$ | -0.5 | 1 | 0 | $p_{2_0}$ | -0.5 | 1 | -0.21 |
| $p_{1_1}$ | -0.5 | 1 | 0 | $p_{2_1}$ | -0.5 | 1 | -0.21 |
| $p_{1_2}$ | -1.5 | 3 | -0.34 | $p_{2_2}$ | -1.5 | 3 | -0.34 |
| $p_{1_3}$ | -2.5 | 3.29 | -0.5 | $p_{2_3}$ | -2.5 | 3.23 | -0.5 |
| $p_{1_4}$ | -2.5 | 3.29 | -0.5 | $p_{2_4}$ | -2.5 | 3.23 | -0.5 |
| $p_{1_5}$ | -3.5 | 3.5 | -1.11 | $p_{2_5}$ | -3.5 | 3.45 | -1.11 |
| $p_{1_6}$ | -4.15 | 3.64 | -1.5 | $p_{2_6}$ | -4.15 | 3.64 | -1.5 |
| $p_{1_7}$ | -5.5 | 3.94 | -2.32 | $p_{2_7}$ | -5.5 | 3.94 | -2.32 |
| $p_{1_8}$ | -5.8 | 4 | -2.5 | $p_{2_8}$ | -5.69 | 4 | -2.5 |
| $p_{1_9}$ | -5.8 | 4 | -2.5 | $p_{2_9}$ | -5.69 | 4 | -2.5 |
| $p_{1_{10}}$ | -6.5 | 4 | -3.32 | $p_{2_{10}}$ | -6.5 | 4 | -3.32 |
| $p_{1_{11}}$ | -6.65 | 4 | -3.5 | $p_{2_{11}}$ | -6.65 | 4 | -3.5 |
| $p_{1_{12}}$ | -7.5 | 4 | -4.5 | $p_{2_{12}}$ | -7.5 | 4 | -4.5 |
| $p_{1_{13}}$ | -7.95 | 4 | -5.5 | $p_{2_{13}}$ | -8 | 4 | -5.5 |
| $p_{1_{14}}$ | -7.95 | 4 | -5.5 | $p_{2_{14}}$ | -8 | 4 | -5.5 |
| $p_{1_{15}}$ | -8.5 | 3 | -5.5 | $p_{2_{15}}$ | -8.5 | 3 | -5.5 |
| $p_{1_{16}}$ | -8.5 | -1 | -5.5 | $p_{2_{16}}$ | -8.5 | -1 | -5.5 |
| $p_{1_{17}}$ | -8.5 | -1 | -0.5 | $p_{2_{17}}$ | -8.5 | -1 | -0.5 |
| $p_{1_{18}}$ | -0.5 | -1 | -0.1 | $p_{2_{18}}$ | -0.5 | -1 | -0.1 |

**Table 2.** Comparison of results of steps of the face-based rubberband algorithm. $p_{1_0}$, $p_{1_1}$, ..., $p_{1_{18}}$ are the results of Step 2, and $p_{2_0}$, $p_{2_1}$, ..., $p_{2_{18}}$ are the results of Step 3.

| $p_{3i}$ | $x_{3i}$ | $y_{3i}$ | $z_{3i}$ | $p_{4i}$ | $x_{4i}$ | $y_{4i}$ | $z_{4i}$ |
|---|---|---|---|---|---|---|---|
| $p_{3_0}$ | -0.5 | 1 | -0.21 | $p_{4_0}$ | -0.5 | 1 | -0.5 |
| $p_{3_1}$ | -0.5 | 1 | -0.21 | $p_{4_1}$ | -0.5 | 1 | -0.5 |
| $p_{3_2}$ | -1.5 | 3 | -0.41 | $p_{4_2}$ | -1.5 | 3 | -0.5 |
| $p_{3_3}$ | -2.5 | 3.23 | -0.5 | $p_{4_3}$ | -2.5 | 3.22 | -0.5 |
| $p_{3_4}$ | -2.5 | 3.23 | -0.5 | $p_{4_4}$ | -2.5 | 3.22 | -0.5 |
| $p_{3_5}$ | -3.5 | 3.47 | -1.13 | $p_{4_5}$ | -3.5 | 3.48 | -1.17 |
| $p_{3_6}$ | -4.09 | 3.62 | -1.5 | $p_{4_6}$ | -4 | 3.61 | -1.5 |
| $p_{3_7}$ | -5.5 | 3.95 | -2.38 | $p_{4_7}$ | -5.5 | 4 | -2.5 |
| $p_{3_8}$ | -5.69 | 4 | -2.5 | $p_{4_8}$ | -5.5 | 4 | -2.5 |
| $p_{3_9}$ | -5.69 | 4 | -2.5 | $p_{4_9}$ | -5.5 | 4 | -2.5 |
| $p_{3_{10}}$ | -6.5 | 4 | -3.4 | $p_{4_{10}}$ | -6.5 | 4 | -3.5 |
| $p_{3_{11}}$ | -6.59 | 4 | -3.5 | $p_{4_{11}}$ | -6.5 | 4 | -3.5 |
| $p_{3_{12}}$ | -7.5 | 4 | -4.5 | $p_{4_{12}}$ | -7.5 | 4 | -4.5 |
| $p_{3_{13}}$ | -8 | 4 | -5.5 | $p_{4_{13}}$ | -8 | 4 | -5.5 |
| $p_{3_{14}}$ | -8 | 4 | -5.5 | $p_{4_{14}}$ | -8 | 4 | -5.5 |
| $p_{3_{15}}$ | -8.5 | 3 | -5.5 | $p_{4_{15}}$ | -8.5 | 3 | -5.5 |
| $p_{3_{16}}$ | -8.5 | -1 | -5.5 | $p_{4_{16}}$ | -8.5 | -1 | -5.5 |
| $p_{3_{17}}$ | -8.5 | -1 | -0.5 | $p_{4_{17}}$ | -8.5 | -1 | -0.5 |
| $p_{3_{18}}$ | -0.5 | -1 | -0.27 | $p_{4_{18}}$ | -0.5 | -1 | -0.5 |

**Table 3.** Comparison of results of steps of the face-based rubberband algorithm. $p_{3_0}$, $p_{3_1}$, ..., $p_{3_{18}}$ are the results of Step 4, and $p_{4_0}$, $p_{4_1}$, ..., $p_{4_{18}}$ are the results of Step 7.

## 5 An Example

We approximate the MLP of the simple cube-curve $g$, shown in Figure 2. Table 1 lists all coordinates of critical edges of $g$. We take the centers of the first critical faces of $g$ to produce an initial polygonal curve for the face-based rubberband algorithm. The updated polygonal curves are shown in Tables [3] 2 and 3. We take the middle points of each critical edge of $g$ for the initialization of the polygonal curve of the (corrected) rubberband algorithm. The resulting polygon is shown in Table 4. Table 5 shows that the edge-based and face-based rubberband algorithms converge to the same $MLP$ of $g$.

| $final_{p_i}$ | $x_i$ | $y_i$ | $z_i$ |
|---|---|---|---|
| $p_{4_0}$ | -0.5 | 1 | -0.5 |
| $p_{4_2}$ | -1.5 | 3 | -0.5 |
| $p_{4_3}$ | -2.5 | 3.22 | -0.5 |
| $p_{4_7}$ | -5.5 | 4 | -2.5 |
| $p_{4_{12}}$ | -7.5 | 4 | -4.5 |
| $p_{4_{13}}$ | -8 | 4 | -5.5 |
| $p_{4_{15}}$ | -8.5 | 3 | -5.5 |
| $p_{4_{16}}$ | -8.5 | -1 | -5.5 |
| $p_{4_{17}}$ | -8.5 | -1 | -0.5 |
| $p_{4_{18}}$ | -0.5 | -1 | -0.5 |

**Table 4.** Results of the edge-based rubberband algorithm. $p_{4_0}$, $p_{4_1}$, ..., $p_{4_{18}}$ are the vertices of the $MLP$ of the simple cube-curve shown in Figure 2.

| step | initial | 2 | 3 | 4 | 8 | edge-based rubberband algorithm |
|---|---|---|---|---|---|---|
| length | 35.22 | 31.11 | 31.08 | 31.06 | 31.01 | 31.01 |

**Table 5.** Lengths of calculated curves at different steps of the face-based rubberband algorithm, compared with the length calculated by the edge-based rubberband algorithm.

## 6 Conclusions

We have presented an edge-based and a face-based rubberband algorithm and have shown that both are provable correct for any simple cube-curve. We also have proved that their time complexity is $\mathcal{O}(m)$, where $m$ is the number of critical edges of $g$. The presented algorithms followed the basic outline of the original rubberband algorithm [1].

## References

1. T. Bülow and R. Klette. Digital curves in 3D space and a linear-time length estimation algorithm. *IEEE Trans. Pattern Analysis Machine Intell.*, **24**:962–970, 2002.

[3] Two digits are used only for displaying coordinates. Obviously, in the calculations it is necessary to use higher precision.

2. J. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In Proc. *IEEE Conf. Foundations Computer Science*, pages 49–60, 1987.
3. J. Choi, J. Sellen, and C.-K. Yap. Approximate Euclidean shortest path in 3-space. In Proc. *ACM Conf. Computational Geometry*, ACM Press, pages 41–48, 1994.
4. D. Coeurjolly, I. Debled-Rennesson, and O. Teytaud. Segmentation and length estimation of 3D discrete curves. In Proc. *Digital and Image Geometry*, pages 299–317, LNCS 2243, Springer, 2001.
5. M. Dror, A. Efrat, A. Lubiw, and J. Mitchell. Touring a sequence of polygons. In Proc. *STOC*, pages 473–482, 2003.
6. E. Ficarra, L. Benini, E. Macii, and G. Zuccheri. Automated DNA fragments recognition and sizing through AFM image processing. *IEEE Trans. Inf. Technol. Biomed.*, **9**:508–517, 2005.
7. A. Jonas and N. Kiryati. Length estimation in 3-D using cube quantization, *J. Math. Imaging and Vision*, **8**: 215–238, 1998.
8. M. I. Karavelas and L. J. Guibas. Static and kinetic geometric spanners with applications. In Proc. *ACM-SIAM Symp. Discrete Algorithms*, pages 168–176, 2001.
9. R. Klette and T. Bülow. Critical edges in simple cube-curves. In Proc. *Discrete Geometry Comp. Imaging*, LNCS 1953, pages 467–478, Springer, Berlin, 2000.
10. R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
11. F. Li and R. Klette. Minimum-length polygon of a simple cube-curve in 3D space. In Proc. *Int. Workshop Combinatorial Image Analysis*, LNCS 3322, pages 502–511, Springer, Berlin, 2004.
12. F. Li and R. Klette. The class of simple cube-curves whose MLPs cannot have vertices at grid points. In Proc. *Discrete Geometry Computational Imaging*, LNCS 3429, pages 183–194, Springer, Berlin, 2005.
13. F. Li and R. Klette. Minimum-Length Polygons of First-Class Simple Cube-Curve. In Proc. *Computer Analysis Images Patterns*, LNCS 3691, pages 321–329, Springer, Berlin, 2005.
14. F. Li and R. Klette. Analysis of the rubberband algorithm. Technical Report CITR-TR-175, Computer Science Department, The University of Auckland, Auckland, New Zealand, 2006 (www.citr.auckland.ac.nz).
15. T.-Y. Li, P.-F. Chen, and P.-Z. Huang. Motion for humanoid walking in a layered environment. In Proc. *Conf. Robotics Automation*, Volume 3, pages 3421–3427, 2003.
16. H. Luo and A. Eleftheriadis. Rubberband: an improved graph search algorithm for interactive object segmentation. In Proc. *Int. Conf. Image Processing*, Volume 1, pages 101–104, 2002.
17. J. Sklansky and D. F. Kibler. A theory of nonuniformly digitized binary pictures. *IEEE Trans. Systems Man Cybernetics*, **6**:637–647, 1976.
18. F. Sloboda, B. Zatko, and R. Klette. On the topology of grid continua. In Proc. *Vision Geometry*, SPIE 3454, pages 52–63, 1998.
19. F. Sloboda, B. Zatko, and J. Stoer. On approximation of planar one-dimensional grid continua. In R. Klette, A. Rosenfeld, and F. Sloboda, editors, *Advances in Digital and Computational Geometry*, pages 113–160. Springer, Singapore, 1998.
20. C. Sun and S. Pallottino. Circular shortest path on regular grids. CSIRO Math. Information Sciences, CMIS Report No. 01/76, Australia, 2001.
21. M. Talbot. A dynamical programming solution for shortest path itineraries in robotics. *Electr. J. Undergrad. Math.*, **9**:21–35, 2004.
22. R. Wolber, F. Stäb, H. Max, A. Wehmeyer, I. Hadshiew, H. Wenck, F. Rippke, and K. Wittern. Alpha-Glucosylrutin: Ein hochwirksams Flavonoid zum Schutz vor oxidativem Stress. *J. German Society Dermatology*, **2**:580–587, 2004.