Segmentation and Length Estimation of 3D Discrete Curves

David Coeurjolly[1], Isabelle Debled-Rennesson[2] and Olivier Teytaud[3]

{dcoeurjo,oteytaud}@univ-lyon2.fr,debled@loria.fr
[1] Laboratoire ERIC, 5, avenue Pierre-Mendes-France F69676 Bron CEDEX
[2] LORIA (Laboratoire LOrrain de Recherche en Informatique et ses Applications),
Campus Scientifique, B.P. 239, F54506 Vandœuvre-les-Nancy
[3] ISC, 67 Bd Pinel, F69675 Bron CEDEX

Abstract. We propose in this paper an arithmetical definition of 3-D discrete lines as well as an efficient construction algorithm. From this notion, an algorithm of 3-D discrete lines segmentation has been developed. It is then used to calculate the length of a discrete curve. A proof of the multigrid convergence of length estimators is presented.
Keywords: 3D discrete line, Segmentation, Length estimation, Discrete curve.

1 Introduction

A formal definition of a 3-D discrete straight line is mandatory in numerous applications using objects based on voxels [31, 16, 14, 15, 6]. For example, drawing 3-D lines between two points of the discrete space is fundamental in discrete ray tracing. The existing methods may be sorted in two categories, a summary is given in [6, 13]:

- "by projections" methods, where two projections of the 3-D segment are independently computed on two basic planes [1]
- "by direct calculus" methods obtained by extension of the 2-D and 3-D algorithms [16, 5]

Several authors have proposed definitions of 3-D 26-connected lines [17, 29] and, as in 2D, it seems to be interesting to propose a simple arithmetical definition of 3-D discrete lines including the different structures obtained by the known drawing algorithms and suggesting a controlled thickness [7].

A more general arithmetical definition is given in [12] but the proposed definition in this paper is sufficient for the presented work and allows to get a simple algorithm for segmentation and 3D discrete curves length estimation. In the second section of this paper, we give an arithmetical definition of 3-D lines as well as a structure theorem allowing the control of the connectivity of 3-D discrete lines so defined. Moreover, a very simple scanning algorithm is proposed which is equivalent to those already known using "by projections" methods. It only relies on calculations on integers. In the third section, a recognition algorithm of naive 3-D lines segments is given ; it relies on the functionality property of a naive 3-D line and the recognition algorithm of 2-D naive lines segments [9]. Then, a linear segmentation algorithm of 3-D discrete curves is directly deduced from the recognition algorithm of 3-D lines segments. In the last section, an application of this algorithm to the calculation of 3-D discrete curves length is proposed as well as the comparison with other methods. The convergence of this length estimation technique is demonstrated and several examples are given.

2 Definitions and main properties

2.1 3-D discrete line

To simplify the writing, we give a definition of a 3-D discrete line whose main vector is $(a, b, c) \in \mathbb{Z}^3$ with $a \geq b \geq c$. The definitions for coefficients ordered in a different way are obtained by permuting x, y, z as well as the coefficients. We suppose in the following that $a \geq b \geq c$, a condition to which it is easy to come back by symmetries.

Definition 1. *The* **3-D discrete line***, called $\mathcal{D}$, whose main vector is $V(a, b, c)$, with $(a, b, c) \in \mathbb{Z}^3$, such that $a \geq b \geq c$ is defined as the set of points (x, y, z) of $\mathbb{Z}^3$ verifying the following diophantine inequalities:*

$$\mathcal{D} \begin{cases} \mu \leq cx - az < \mu + e & (1) \\ \mu' \leq bx - ay < \mu' + e' & (2) \end{cases}$$

with $\mu, \mu', e, e' \in \mathbb{Z}$. μ and μ' are called the **lower bounds** *of $\mathcal{D}$, moreover, e and e' are named the* **arithmetical thickness** *of $\mathcal{D}$. Such a line is noted $\mathcal{D}(a, b, c, \mu, \mu', e, e')$.*

Remarks:

1. For a given main vector, there is a large number of different representations according to the lower bounds and thicknesses chosen (see Fig. 1. and Fig. 3.).
2. In the plane Oxz (resp. Oxy), the double inequality (1) (resp. (2)) represents a 2-D arithmetical line [26] with lower bound μ (resp. μ') and arithmetical thickness e (resp. e'), noted $\mathcal{D}_{2D}(c, a, \mu, e)$ (resp. $\mathcal{D}_{2D}(b, a, \mu', e')$).
3. The double diophantine inequalities (1) and (2) define two discrete planes of $\mathbb{Z}^3$ ([2, 8, 7]).

Definition 2. *We call* **3-D naive lines***, whose main vector is $V(a, b, c)$, with $(a, b, c) \in \mathbb{Z}^3$, such that $a \geq b \geq c$, the 3-D discrete lines whose thickness verifies $e = e' = a$, they are named $\mathcal{D}(a, b, c, \mu, \mu')$.*

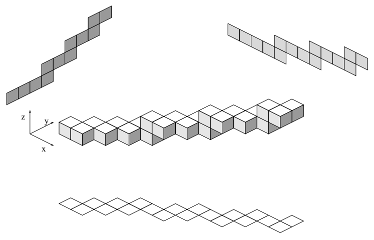

Fig. 1. Representation in voxels of $\mathcal{D}(10, 7, 3, 0, -9, 13, 17)$, a 6-connected line where a point (x, y, z) of $\mathcal{D}$ is represented by a unit cube centered at the integer point (x, y, z)

Remark: A 3-D naive line is projected into the plane Oxz and Oxy in two 2-D naive lines [26, 9], noted $\mathcal{D}_{2D}(c, a, \mu)$ and $\mathcal{D}_{2D}(b, a, \mu')$.

Theorem 1. *A 3-D naive line $\mathcal{D}(a, b, c, \mu, \mu')$ with $a \geq b \geq c$ is functional in x.*

Proof. Indeed, let us consider the naive line $\mathcal{D}(a, b, c, \mu, \mu')$, $\mathcal{D}$ is defined by

$$\begin{cases} \mu \leq cx - az < \mu + a \\ \mu' \leq bx - ay < \mu' + a \end{cases} \tag{3}$$

which is equivalent to

$$\begin{cases} z = \left[\frac{cx-\mu}{a}\right] \\ \\ y = \left[\frac{bx-\mu'}{a}\right] \end{cases} \tag{4}$$

where $[\ldots]$ denotes integer part of x.

2.2 Relation between arithmetical thicknesses and connectivity of 3-D discrete lines

In order to get a better understanding of the 3-D discrete line notion, here is the structure theorem which establishes a link between the arithmetical thicknesses and the connectivity of a 3-D discrete line.

Theorem 2. *Let us consider $\mathcal{D}(a, b, c, \mu, \mu', e, e')$ with $a > b > c$,*
If $e \geq a + c$ and $e' \geq a + b$, $\mathcal{D}$ is 6-connected (see Fig 1.).
If $e \geq a+c$ and $a \leq e' < a+b$ or $e' \geq a+b$ and $a \leq e < a+c$, $\mathcal{D}$ is 18-connected.
If $a \leq e < a + c$ and $a \leq e' < a + b$, $\mathcal{D}$ is 26-connected (see Fig 3.).
If $e < a$ or $e' < a$, $\mathcal{D}$ is disconnected.

Proof. In order to demonstrate these results, we use the fact that projections of $\mathcal{D}$ in the planes Oxy and Oxz are 2-D discrete lines whose connectivity is known [26].
For example, let us consider the third case ; $a \leq e < a + c$ and $a \leq e' < a + b$, in the plane Oxz, we move from a pixel to the next one in the discrete line $\mu \leq cx - az < \mu + e$ through a 4-connected or 8-connected move, the same in the plane Oxy for the line $\mu' \leq bx - ay < \mu' + e'$. All possible combinations between two pixels of planes Oxy and Oxz to get two voxels of $\mathbb{Z}^3$ have been studied on the figure 2.

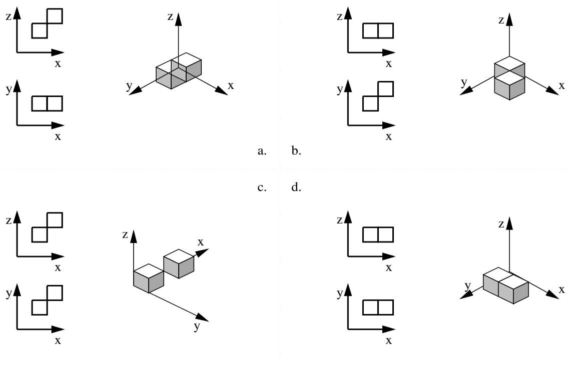

Fig. 2.

a. an 8-connected move in the plane Oxz and a 4-connected move in the plane Oxy lead to two voxels linked by an 18-connected move,
b. a 4-connected move in the plane Oxz and an 8-connected move in the plane Oxy lead to two voxels linked by an 18-connected move,
c. an 8-connected move in the plane Oxz and an 8-connected move in the plane Oxy lead to two voxels linked by a 26-connected move,
d. a 4-connected move in the plane Oxz and a 4-connected move in the plane Oxy lead to two voxels linked by a 6-connected move.

Therefore we show that the line $\mathcal{D}$ is 26-connected.
The other results can be demonstrated in a similar way by watching all combinations of two voxels obtained from the possible ones of two pixels in the planes Oxz and Oxy.

Remark: The naive lines are 26-connected and we go from a floor to the next one only through an edge or a vertex, we can see it on the figure 3.

The next theorem states that there is a link between the 3-D GIQ (Grid Intersect Quantization) [13] and the 3-D straight line defined above. We prove that

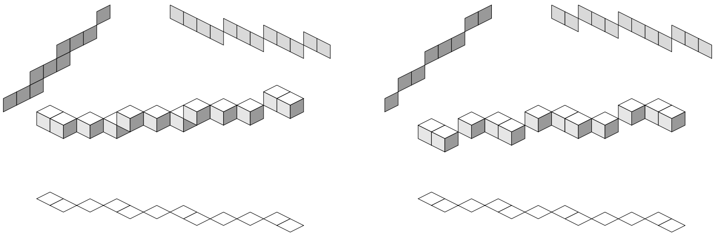

Fig. 3. On the left $\mathcal{D}(10,7,3,0,0)$, on the right $\mathcal{D}(10,7,3,-5,0)$

.

this digitization process digitizes an Euclidean straight line into a 26-connected set of voxels such that 2 of its 3 canonical projections are 2-D discrete straight lines. Note that in a general case, Jonas et al. [13] show that none of the classical digitizations into 26-connected curve has the *three*-projection property (*i.e* projections of a 3-D chain onto each axis planes are identical to the digitization of the continuous curve projections).

Theorem 3. *The 3-D GIQ (Grid Intersect Quantization) of an Euclidean straight line is a 3-D discrete straight line defined above.*

Proof. Without loss of generality, we consider $\mathcal{D}_{euc}$ a 3-D Euclidean straight line whose main vector is $(a,b,c) \in \mathbb{Z}^3$ with $a \geq b \geq c$, defined by:

$$\mathcal{D}_{euc} = \begin{cases} z = \frac{cx-r}{a} \\ \\ y = \frac{bx-r'}{a} \end{cases} \tag{5}$$

We consider the Euclidean planes $\mathcal{P}_1 : z = \frac{cx-r}{a}$ and $\mathcal{P}_2 : y = \frac{bx-r'}{a}$. The GIQ-digitization of these planes gives the following arithmetical naive planes:

$$\begin{array}{l} GIQ(\mathcal{P}_1) = \mathcal{P}(c,0,-a,r+\left[\frac{a}{2}\right]) \\ GIQ(\mathcal{P}_2) = \mathcal{P}(b,-a,0,r'+\left[\frac{a}{2}\right]) \end{array} \tag{6}$$

where $\mathcal{P}(u,v,w,\mu) : \mu \leq ux + vy + wz < \mu + max(|u|,|v|,|w|)$ [2,8]. In the following, we consider a free ambiguity GIQ process [13] (we assume a global strategy to remove pathological cases).

The intersection between this two discrete naive planes defines a 26-connected discrete line according to definition 1 and the structure theorem 2:

$$\mathcal{D} = \begin{cases} r + \left[\frac{a}{2}\right] \leq cx - az < r + \left[\frac{a}{2}\right] + a \\ r' + \left[\frac{a}{2}\right] \leq bx - ay < r' + \left[\frac{a}{2}\right] + a \end{cases} \tag{7}$$

Let v be a voxel of $\mathcal{D}$, we consider $GIQ(\mathcal{D}_{euc})$ the GIQ-digitization of $\mathcal{D}_{euc}$. If v belongs to $\mathcal{D}$, according to the GIQ process of the planes, we have (d denotes the classical Euclidean distance):

$$d(v,\mathcal{P}_1) \leq \sqrt{2} \text{ and } d(v,\mathcal{P}_1) \leq \sqrt{2} \Leftrightarrow d(v,\mathcal{D}_{euc}) \leq \sqrt{2} \tag{8}$$

This is equivalent to:

$$v \in \mathcal{D} \Rightarrow v \in GIQ(\mathcal{D}_{euc}) \tag{9}$$

and thus

$$\mathcal{D} \subset GIQ(\mathcal{D}_{euc}) \tag{10}$$

To prove the symmetric inclusion we consider $v' \in GIQ(\mathcal{D}_{euc})$, if $v' \notin \mathcal{D}$ we have two possibilities:

- $d(v',\mathcal{P}_1) > \sqrt{2}$ or $d(v',\mathcal{P}_2) > \sqrt{2}$ that is equivalent with $d(v',\mathcal{D}_{euc}) > \sqrt{2}$ which leads to a contradiction with the GIQ process of $\mathcal{D}_{euc}$;
- $d(v',\mathcal{D}_{euc}) \leq \sqrt{2}$ but since the GIQ is free ambiguity, v' disconnects $\mathcal{D}$ which leads to a contradiction with the structure theorem 2.

Finally:

$$\mathcal{D} = GIQ(\mathcal{D}_{euc}) \tag{11}$$

□

2.3 A scanning algorithm of 3-D naive discrete lines

Let us consider $A(x_A, y_A, z_A)$ and $B(x_B, y_B, z_B)$ two points of $\mathbb{Z}^3$. The algorithm given below draws a naive line segment between the two points A and B whose main vector is $\boldsymbol{AB} = (v_x, v_y, v_z)$. According to the choice of the lower bound, different structures are obtained. In particular, for $\mu = \mu' = -[\frac{max(|v_x|,|v_y|,|v_z|)}{2}] + 1$, the generated voxels are the same as those defined by the algorithm of Bresenham extended to 3D [4, 31].
The principle of the algorithm consists in computing in parallel in both planes Oxy and Oxz the pixels of the 2-D naive lines which are the projections of the 3-D line in these two planes. So we recalculate the three coordinates (x, y, z) of the corresponding voxel.

Algorithm for the construction of a naive discrete line segment with lower bounds μ and μ' between two points $A(x_A, y_A, z_A)$ and $B(x_B, y_B, z_B)$ of $\mathbb{Z}^3$ such that $x_B - x_A > y_B - y_A$ and $y_B - y_A > z_B - z_A$.

```
$v_x = x_B - x_A$; $v_y = y_B - y_A$; $v_z = z_B - z_A$;
$r_1 = v_z * x_A - v_x * z_A - \mu$;
$r_2 = v_y * x_A - v_x * y_A - \mu'$;
$x = x_A$; $y = y_A$; $z = z_A$;
While $x < x_B$ repeat
    $x = x + 1$;
    $r_1 = r_1 + v_z$;
    $r_2 = r_2 + v_y$;
    If $r_1 < 0$ or $r_1 \geq v_x$ then
        $z = z + 1$;
        $r_1 = r_1 - v_x$;
    Endif
    If $r_2 < 0$ or $r_2 \geq v_x$ then
        $y = y + 1$;
        $r_2 = r_2 - v_x$;
    Endif
    Print the point $(x, y, z)$;
EndRepeat
```

This algorithm is linear in the number of points to be computed and very simple.

3 Algorithm for 3-D naive line segment recognition

The recognition of 3-D naive line segments is a direct outcome of the recognition of 2-D naive line segments [9, 7]. Indeed, a 3-D naive line segment is bijectively projected into at least two orthogonal planes as two 2-D naive discrete straight lines. Consequently, recognizing a 3-D naive line segment consists in recognizing both 2-D naive line segments of its functional projections. Hence the linear algorithm proposed in [9, 7] is particulary adapted to this problem of 3D curve segmentation into naive lines, even more than other linear algorithms of 2D curve segmentation like [11, 25].

Algorithm for 3-D naive line segment recognition

```
Input: 26-connected sequence of voxels to be analysed, named $\mathcal{S}$
Output: True if $\mathcal{S}$ is 3-D naive line segment, False if not

If the voxels of $\mathcal{S}$ may not be bijectively projected on at least two orthogonal planes in order to create two curves of pixels $C_1$ and $C_2$ then $\mathcal{S}$ is not a 3-D line segment, return False; Endif
Apply the algorithm of 2-D naive line segments recognition on $C_1$ and $C_2$;
If $C_1$ and $C_2$ are two 2-D naive line segments then
    $\mathcal{S}$ is a 3-D naive line segment, return True;
Else $\mathcal{S}$ is not a 3-D naive line segment, return False;
Endif
```

This algorithm is linear in number of points of the sequence to be analysed and all necessary calculations are exclusively done using integers.

The characteristics of the recognized line segment are determined as a function of the characteristics of the 2-D line segments obtained. For example, let us suppose that $\mathcal{S}$ is bijectively projected on the plane Oxy curves C_1 and C_2. The recognition algorithm of 2-D naive line segments recognizes C_1 as a segment of $\mathcal{D}_{2D}(c_1, a_1, \mu_1)$ and C_2 as a segment of $\mathcal{D}_{2D}(b_2, a_2, \mu_2)$. Let us consider m the smallest common multiple of a_1 and a_2 such that $m = k_1 a_1 = k_2 a_2$ then $\mathcal{S}$ is a 3-D naive line segment whose main vector is $V(m, k_2 b_2, k_1 c_1)$ and with lower bounds μ_1 and μ_2.

4 Segmentation of 3D discrete curves

The segmentation of a 26-connected discrete curve, named $\mathcal{C}$, consists in “splitting” $\mathcal{C}$ into maximal length segments of 3-D naive lines. To do so, we use the same principle as the algorithm presented in the previous section, but, to achieve a linear complexity, the search of functional projection planes is not executed in the same way. We proceed by increment ; three 2-D discrete curves are built and segmented along the recognition, they correspond to the projections of the curve part already scanned in the three coordinates planes.

The algorithm given below uses the same bases as the segmentation algorithm of 2-D discrete curves presented in [9, 7]. A sequence $\mathcal{C}$ of 26-connected voxels is given as input.

```
Algorithm of segmentation of 26-connected discrete curves

M = first voxel of $\mathcal{C}$
While $\mathcal{C}$ has not entirely been scanned repeat
    In each coordinate planes $Oxy$, $Oxz$, $Oyz$, init the characterictics of new segments $S_1$,
$S_2$ and $S_3$;
    SEGMENT=true;
    While the added voxel $M$ possesses at least two projections in which the projected
     points of $M$ can be added to the current 2-D segments [9, 7] and SEGMENT
     repeat
        If the projections of the following voxel of $\mathcal{C}$ in the two coordinates planes used
        belong to the same octants as the current 2-D segments then
            M = following point of $\mathcal{C}$;
            Add by symmetry the projected pixels of M, to the 2-D segments concerned;
        Else
             SEGMENT = false;
        Endif
    End repeat
According to symmetries, compute the characteristics of the naive lines of the two 2-D
segments obtained in the initial octants, then deduce the main vector of the recognized
naive line segment;
End repeat
```

Remarks:
1. The first point of a new segment is the last of the previous segment.

2. An algorithm for 4-connected 2D line segment recognition is presented in [7] and it's easy to use it (as presented before) for the segmentation of 18-connected discrete curves.

Example: On the figure 4 we can see an example of the segmentation of a discrete curve, the starting and ending segment voxels are in black. The projection planes used are Oxy and Oyz, the segments resulting from the 2-D recognition are drawn in light gray on the projections in these two planes of the considered voxels set. Two naive line segments are obtained:

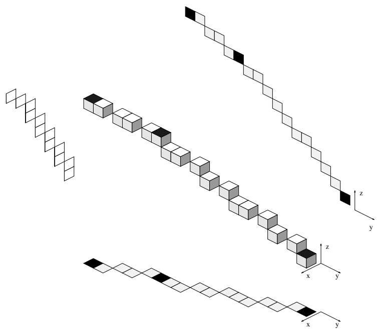

Fig. 4.

Segment 1 which contains 13 points. The inequalities of the discrete naive lines of the 2-D segments obtained are $-4 \leq -4y - 5z < 1$ and $-2 \leq -2y - 5x < 3$. Consequently, the 3-D naive line of the segment 1 has $V(2, -5, 4)$ as main vector. Segment 2 length is 6. The inequalities of the discrete naive straigth lines of the 2-D segments obtained are $0 \leq y - 2z < 2$ and $0 \leq y - 2x < 2$. Consequently, the 3-D naive line of the segment 2 has $V(1, -2, 1)$ as main vector.
Other examples of segmentation of 26-connected curves are given in the figure 5.

5 Length estimation algorithm

In 2D, a lot of approaches have been proposed for the length estimation of a discrete curve. These approaches can be *sorted* as follows:

- Local approaches [23, 10]: estimators are based on local or fixed size window characteristics of the discrete curve. An example of such estimators is based on counting the number of odd and even codes in the Freeman chain code of the curve and returning the length estimation $l = \alpha n_{odd} + \beta n_{even}$ where α and β are computed using statistical analysis (*e.g* $(\alpha, \beta) = (1.183, 1.059)$ in [10]). Such estimators exist for 3D curves [19, 18] in a same approach : find the optimal weights associated to different chain codes. In the following, in spite of advantages of such approaches, we only deal with global algorithm.

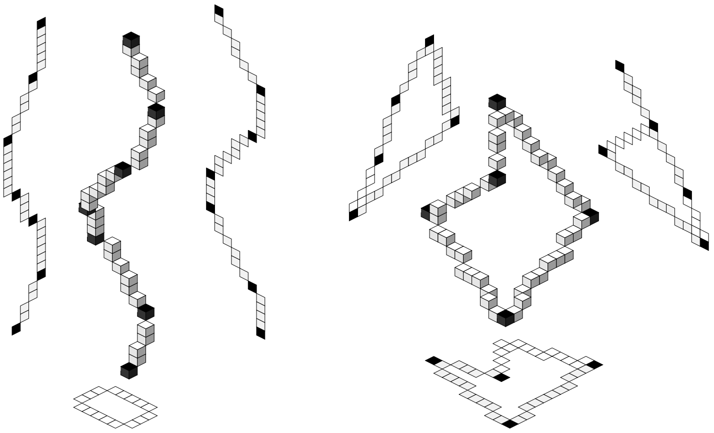

Fig. 5. The black voxels indicate the starting and ending segment. The 2-D segmented curves are in light gray.

- Discrete Straight Segments (DSS) approaches [24]: these methods are defined by a curve partitioning into digital straight line segments of maximum length each. Hence, the length estimator is based on the obtained polyline length.
- Minimum Length Polygon approaches (MLP) [27,28,21]: these methods compute a minimum length polygon that belongs to an open boundary of a digital object.
- Euclidean Path approaches: in these approaches, a semi-continuous representation of the discrete curve is computed using Euclidean points located in a open square of size one centered on each discrete points. Since this Euclidean path provides a good approximation of the underlying real curve, a length estimator on this path can be defined either as the path polyline length [30] or embeded in an MLP process [3].

In the 3D case, Klette and Bulow [20] propose a 3D version of the MLP approach. The input of the algorithm is a closed *cube*-curve that is a strictly 6-connected closed curve, the minimum length polygons is computed in two steps (see figure 5):

1. The vertices of the candidate polygon are first initialized to the *inner* vertex of edges of the cube-curve that are adjacent to three distinct cubes; such edges are called critical edges;
2. local operations (move or delete) are computed on the polygon vertices to converge to the minimum length polygon.

This first 3D global algorithm for the length estimation leads to several problems: the first one is purely algorithmic since this algorithm is not linear in the number of points of the discrete curve. This non-linear complexity is due to the local optimization processes that introduce backtrack on the polygon vertex chain. Since a vertex of a candidate ML-polygon can be moved along a critical

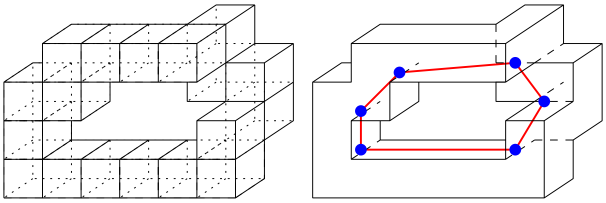

Fig. 6. An example of a cube-curve with its ML-polygon.

edge, the calculated vertices of the polygon might not belong to the discrete grid and we have to introduce an τ-MLP that is the minimum length polygon up to a threshold τ (*i.e.* $|l_{\tau-MLP}(C) - l_{MLP}(C)| \leq \tau$ where C denotes the cube-curve).

Furthermore, there is no multigrid convergence proof of this MLP 3D version.

In the following, we propose a length estimator based on the 3D naive discrete straight line segmentation. The interesting points of such an algorithm are:

- a purely discrete and optimal in time algorithm;
- multigrid convergence of the length estimator whatever the curve is closed or not and whatever the dimension;
- since the segmentation process holds whatever *thickness* has the curve, we have a general length estimator whatever the topology of the curve.

We first describe the algorithm based on the Naive Straigth line Segmentation (NSS):

```
NSS length estimation algorithm

Input: 26-connected sequence of voxels to be analysed, named $\mathcal{S}$
Output: the estimated length of $\mathcal{S}$

Compute the segmentation using the above algorithm of $\mathcal{S}$
Let $\mathcal{P}$ =$\{S_i\}_{0...n}$ denotes the polyline returned by the segmentation. The vertices of this polyline are the starting and the ending points of the naive line segments.
Return $\sum_{i=0}^{n} l(S_i)$ (where $l(S_i)$ denotes the Euclidean length of the segment $S_i$)
```

For the sake of clarity, this algorithm is designed as a post-processing of the segmentation but it can easily be embedded in the segmentation process since it is purely greedy.

In the next section, we present a proof of the multigrid convergence of a slight variant of NSS.

5.1 Proof of the multigrid convergence

In [24, 22], Kovalevsky and Klette prove the multigrid convergence of the 2D DSS approach. The main result of this proof is that the error between the boundary length, denoted by $\mathcal{B}_C$, of an Euclidean convex object and the estimated length using a DSS segmentation of the digitized boundary, denoted by B_{C_δ} on a grid of size δ, is linear with δ:

$$|l(\mathcal{B}_C) - l_{DSS}(B_{C_\delta})| \leq O(\delta) \tag{12}$$

In the following, we propose a proof of this asymptotic convergence under the curvature hypothesis without the convexity one, and that remains true in any dimension. In the sequel we assume the following:

1. $\varGamma : [0,1] \rightarrow \mathbb{R}^d$ denotes the underlying continuous curve whose curvature is bounded by $C = \frac{1}{R}$.
2. $\varGamma$ is included in the δ-enlargements of tubes T_1, ... ,T_N of diameter δ and lengths $L(T_1)$, ... ,$L(T_N)$. The δ-enlargement of a tube is the union of this tube and two parts as described in figure 7 (for negative values of δ, the notion of enlargement is similarly defined).
3. T_1 starts at $\varGamma(0)$, within $K_1\delta$ error.
4. T_N ends at $\varGamma(1)$, within $K_2\delta$ error.
5. T_i's end is T_{i+1}'s start.
6. $L(T_i) \geq K_3\sqrt{\frac{\delta}{C}}$, for any $i < N$.

We approximate the length of $\varGamma$ by the length sum of the T_i's. This tube approach is a general framework for discrete polygonalization schemes.

The begin and end of a tube are the intersection of the axis with the lateral faces (extremities). The pertinence of the last hypothesis is discussed later in the case of particular discretizations. Moreover, we assume that δ is small enough to avoid pathological cases such as half-turns in a tube ($\frac{\delta}{2} < R$). The goal is a bound on the error, linear in δ. Notice that the hypothesis (2) implies that $\varGamma$ gets out of the $(-\delta)$-enlargements of the T_i's through lateral faces.

We first give an overview of the proof: Lemma 1 shows that the angle of $\varGamma$ within a tube is small enough to achieve a linear error in δ *in* a tube (Corollary 1). In Lemma 2, we show an upper bound on the angle between two adjacent tubes that leads to an upper bound on the error while *joining* two tubes. Under the sixth hypothesis, we have a sufficiently small number of tubes to achieve a linear bound on the summed error of joinings (Corollary 2). Corollaries 1, 2 lead to a global linear bound on the error.

Lemma 1. *The angle between $\varGamma$[1] and the axis of T_i in the $(-\delta)$-enlargement of T_i is lower than $B = \arcsin(\frac{\delta}{L-\delta}) + L(T_i)/R$. If α such that $\tan\alpha = \sqrt{\frac{\delta}{2R-\delta}}$ verifies $R\sin(\alpha) \leq L(T_i)$, then the angle is bounded by $B = \alpha$.*

Proof. See figure 7: this is the case of largest negative derivative for a projection on a plane including the direction of the tube. The following holds:

$$\varOmega \leq L/R \tag{13}$$

$$\sin\theta \leq \frac{\delta}{L-\delta} \tag{14}$$

[1] This angle is the maximal one between a tangent of $\varGamma$ and the tube axis.

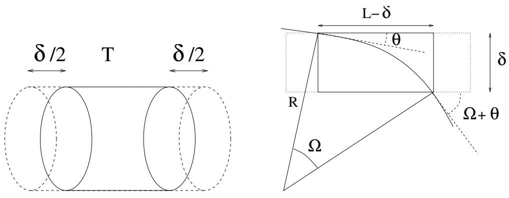

Fig. 7. *Left*: δ-enlargement of tube T. *Right*: schema for proof of lemma 1: L is the tube length, the dashed line is the tube contour, the solid line is the $(-\delta)$-enlargement of the tube.

This yields the first result. The particular case in which $R\sin(\alpha) \leq L(T_i)$ for $\tan\alpha \leq \sqrt{\frac{\delta}{2R-\delta}}$ is illustrated in figure 8; in this case, L is large, and the derivative must be small enough to keep Γ in the enlargement of the tube. This proves that the angle is bounded by $\arctan(\sqrt{\frac{\delta}{2R-\delta}})$.$\Box$

This bound is $O(\sqrt{\delta C})$ under the above assumptions. Moreover, this proves that the derivative of Γ in a parametrization colinear to the axis of the tube is bounded by $\sqrt{1+(d-1)p^2}$ with $p = sin(B)$. This bounds the difference between the length of Γ in T_i and the length of T_i by $L(T_i)(\sqrt{1+(d-1)p^2}-1)$:

Corollary 1. *The difference between the length of T_i and the length of Γ in T_i is bounded by $L(T_i)(\sqrt{1+(d-1)p^2}-1)$. This is linear in $L(T_i)$ and δC.*

This is linear in δ, but does not complete the proof: we have a linear error in δC for segmentations, but "joining" the segments leads to small errors (see figure 8): some parts could be taken into account zero or two times. These errors have to be bounded too. For this we will need bounds on the derivative of parametrizations of Γ "later" than in the $(-\delta)$-enlargements; in the following lemma we work on δ enlargements, with similar results.

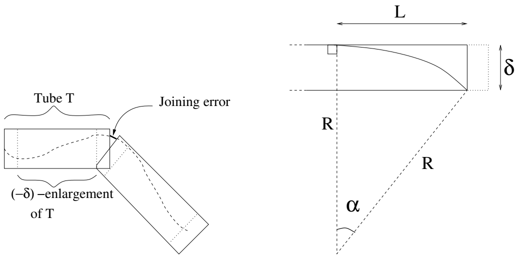

Fig. 8. *Left*: Errors out of segments. *Right*: Case L large in Lemma 1.

Lemma 2. *If the angle between Γ and the axis is lower than θ in the $-\delta$-enlargement, then it is lower than $\theta' = \arcsin(\sin(\theta)+\delta C)$ in the δ-enlargement.*

Moreover, the distance to the axis is lower than $R\cos(\theta) - R\cos(\theta')$ if Γ gets out of the $-\delta$ enlargement at the lateral faces (extremities).

Proof. See Fig. 9. This is the 2D projection of the worst case. One can easily derive the followings:

$$R\cos(\theta') + A = R\cos(\theta) \tag{15}$$

$$R\sin(\theta') = R\sin(\theta) + \delta \tag{16}$$

This yields the result. □

This implies that Γ has variations of angle within θ'. The angle Ω between T_i and T_{i+1} is bounded by $2\theta'$. Thus the length lost or counted two times between T_i and T_{i+1} is bounded by the Γ length between getting out of T_i and getting in T_{i+1}, which is bounded by $\sqrt{1+(d-1)p^2}\frac{\Omega\delta}{2}$, with Ω angle between T_i and T_{i+1}, with $p = sin(\theta')$.

This leads to the following important corollary:

Corollary 2. *The error committed in joining two tubes is $O(\delta\Omega) = O(\delta^{3/2}\sqrt{C})$. As the number of such errors is bounded by $O(\frac{C}{\delta})^{\frac{1}{2}}$, the total error resulting of joins is $O(\delta C)$.*

Hence, using corollary 1, we obtain the expected result. The error is linear under the above hypothesis. Constants have not been explicited as they are far from optimal by the proof below. More refined results would be useful. All hypothesizes are verified except the last one for usual segmentations, with δ linear in the resolution. We now have to study the hypothesis according to which $L(T_i) \geq K_3\sqrt{\delta/C}$, for any $i < N$.

This hypothesis is particularly true whenever the discretization and polygonalization verify that any curve which lies in a tube of radius ϵr with the resolution r is included in a segment (this can easily be proved by considering the fact that for bounding curvature C, the minimal length of Γ before ϵr-deviation from the tangent is $\Theta(\sqrt{\frac{\epsilon r}{C}})$). This will be refered in the sequel as an *ϵ-tolerant algorithm.* Experiments below have been made with a straightforward algorithm which doesn't have the algorithmic properties of the initial algorithm.

Figure 9 illustrates the fact that with 8-connectivity and with segments only 8-connected, large errors can be made. One can have similar bad behaviour with 4-connectivity; as explained above, a solution to avoid such bad results is adding a tolerance in the algorithm (this is simply an increase of the width parameters of bands assimilated to segments). As such pathological cases can be considered as negligible, perhaps the initial NSS algorithm would have a similar behavior in practical cases. Notice that such counter-examples do not mislead MLP. We present experiments illustrating proved results; Fig. 10 shows linear convergence for a circle (2D, convex, convergence proved in [24, 22]), and Fig. 12 shows linear convergence of the ϵ-tolerant variant for an helix (3D, with bounded curvature, proof above). Then Fig. 13 shows that in practical 3D-cases, linear convergence can be achieved with an ellipse even without ϵ-tolerance; however, convergence is much less regular than with ϵ-tolerance.

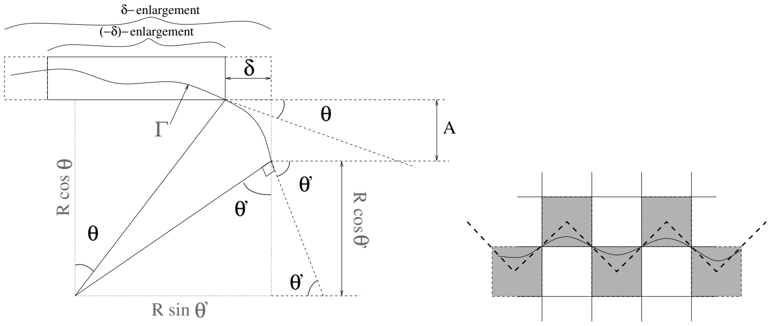

Fig. 9. *Left*: Figure for lemma 2.*Right*: $\Gamma(t) = (t, r^2 cos(> pt/r))$. The curvature is bounded by 1 for any value of the resolution r. The length computed (the length of the dashed curve) by a discretization algorithm in this case does not converge to the length of the curve for small values of r.

Note that, in $\mathbb{R}^2$, the proof using convexity (see [24, 22]) can be extended to a finite number of inflexions (or finite number of areas with null curvature), with bounded curvature. This is possible without the hypothesis above about the length of segments. Convexity avoids particular cases as in Fig. 9 (right).

5.2 Experiments

In our experimentations, we compute the NSS length of 3D mathematical curves (circles, ellipses and helixes) on which the theoretical length is known, using the above algorithm. Hence we consider a digitization of those Euclidean objects, using the GIQ process, on grids of increasing resolution in order to check the asymptotic convergence. In Fig. 10, we present the multigrid error graph and the execution time graph in 2D for a circle, and in Fig. 12, we present the multigrid error graph with an ϵ-tolerant algorithm for an helix in 3D. In Fig. 13, we show that in the particular case of a 3D ellipse, the NSS length algorithm converges. The interest of such a test is to check if it is possible, in concrete cases, to achieve a linear decrease of the error without ϵ-tolerance and thus use the algorithmic advantages of the NSS approach. The error is defined as the percentage of the absolute value of the difference between the true length and estimated length, to the true length. First, these results show an experimental asymptotic convergence since the error decreases with the resolution. If we compute the error inverse, we verify the linear nature of the convergence. Since the starting point of a closed curve in the segmentation algorithm may lead to different polygonalizations, we have plot for the circles curve the 95% confidence intervals (vertical bars) when we compute all possible segmentations. Hence, we can see that this starting point problem does not affect the results. The execution graph shows the linear complexity of the segmentation process in the number of points of the discrete curve.

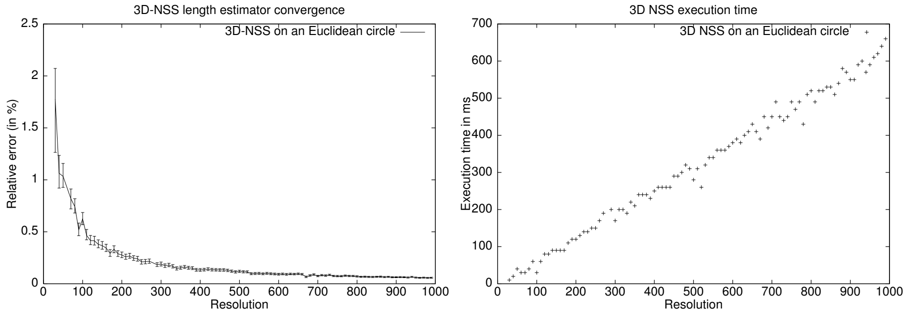

Fig. 10. These graphs present the error of the estimator on both 3D circles and the execution time in ms on a Sun Ultra SparcWorkstation.

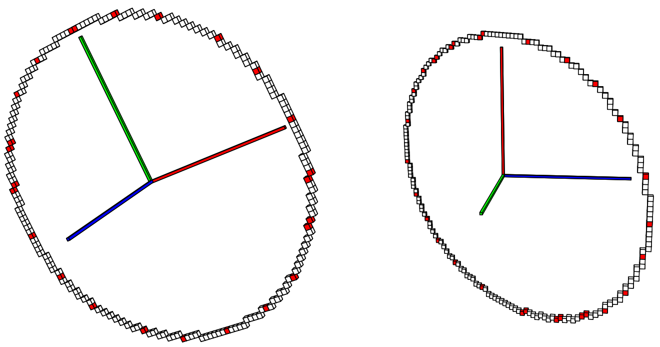

Fig. 11. An example of a NS-segmentation of a circle and an ellipse.

In [21], Klette et al. have already compaired the DSS and the MLP approaches in the 2D case and the comparisons both in execution time, speed of convergence and accuracy show that the DSS approach gives better results. In the 3D case, since the 3D MLP algorithm suffers from several problems, and since the 3D DSS does not, theoretically and practically, differ to the 2D one, we are convinced that the comparison will lead to the same results.

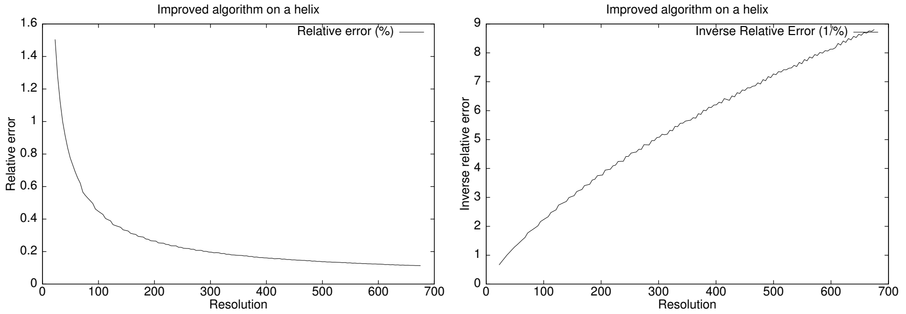

Fig. 12. *Left*: multigrid convergence of the ϵ-tolerant algorithm in the case of an helix in $\mathbb{R}^3$. *Right*: illustration of the error decrease.

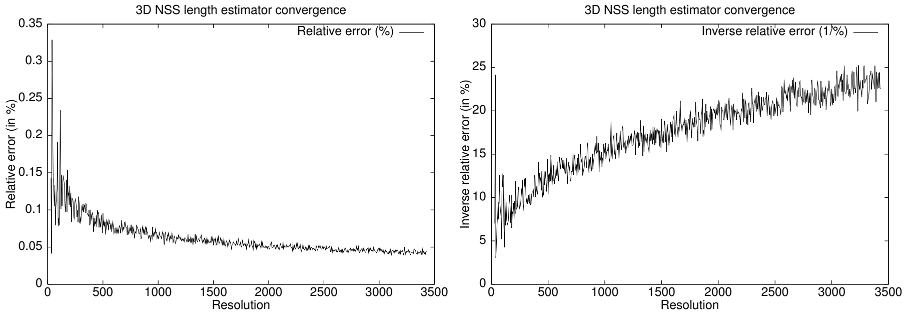

Fig. 13. *Left*: multigrid convergence of the 3D-NSS on a 3D ellipse of parameter $(1, \sqrt{2})$. *Right*: illustration of the linear decrease of the error.

6 Conclusion

In this article, we have first presented an arithmetical definition of 3D discrete lines and an extension to the third dimension of the discrete naive straight line recognition algorithm. Moreover we have shown a segmentation algorithm which is purely discrete and optimal in time ($O(n)$ in time and $O(1)$ in space). Based on this algorithm, we then proposed a 3D discrete curve length estimator with the same complexity. In the last section, we have presented a proof of the multigrid convergence of length estimators based on a general *tube* framework in any dimension. We have shown that the discrete naive straight line segmentation verifies the proof hypothesis under a slight refinement and therefore the length estimator based on this algorithm also converges. The error bound we found is the same as in the literature concerning the DSS approaches but with less hypothesis on the underlying Euclidean curve. We suggest that this general framework could be used in the case of the 3D MLP approach: bounding the number of critical edges as in hypothesis (6) would be sufficient.

References

1. J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographic's 87*, pages 3–12, 1987.
2. E. Andres. Le plan discret. In *Colloque en géométrie discrète en imagerie: fondements et applications*, Septembre 1993.
3. T. Asano, Y. Kawamura, R. Klette, and K. Obokata. A new approximation scheme for digital objects and curve length estimations. Technical Report CITR-TR-65, Computer Science Departement of The University of Auckland, September 2000.
4. J.E. Bresenham. Algorithm for computer control of a digital plotter. In *IBM System Journal*, volume 4, pages 25–30, 1965.
5. D. Cohen and A. Kaufman. Scan-conversion algorithms for linear and quadratic objects. In *IEEE Computer Society Press, Los Alamitos, Calif*, pages 280–301, 1991.
6. D. Cohen-Or and A. Kaufman. 3d line voxelization and connectivity control. In *IEEE Computer Graphics and Applications*, pages 80–87, 1997.
7. I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets.* PhD thesis, Thèse. Université Louis Pasteur, Strasbourg, 1995.
8. I. Debled-Rennesson and J.P. Reveillès. A new approach to digital planes. In *In Vision Geometry III, SPIE, Boston*, volume 2356, pages 12–21, 1994.
9. I. Debled-Rennesson and J.P. Reveillès. A linear algorithm for segmentation of digital curves. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 9, pages 635–662, 1995.
10. L. Dorst and A. W. M. Smeulders. Length estimators for digitized contours. *Computer Vision, Graphics, and Image Processing*, 40(3):311–333, December 1987.
11. L. Dorst and A.W.M. Smeulders. Decomposition of discrete curves into piecewise straight segments in linear time. In *Contemporary Mathematics*, volume 119, 1991.
12. O. Figueiredo and J.P. Reveillès. A contribution to 3d digital lines. In *Proc. DCGI'5*, pages 187–198, 1995.
13. A. Jonas and N. Kiryati. Digital representation schemes for 3d curves. *Pattern Recognition*, 30(11):1803–1816, 1997.
14. A. Kaufman. An algorithm for 3-d scan conversion of polygons. In *Proc. Eurographic's 87*, pages 197–208, 1987.
15. A. Kaufman. Efficient algorithms for 3-d scan conversion of parametric curves, surfaces, volumes. In *Computer Graphic 21, 4*, pages 171–179, 1987.

16. A. Kaufman and E. Shimony. 3-d scan conversion algorithms for voxel-based graphics. In *ACM Workshop on Interactive 3D Graphics, ACM Press, NY*, pages 45–75, 1986.
17. C.E. Kim. Three-dimensional digital line segments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 5, pages 231–234, 1983.
18. N. Kiryati and A. Jonas. Length estimation in 3d using cube quantization. *Journal of Mathematical Imaging and Vision*, 8:215–2138, 98.
19. N. Kiryati and O. Kubler. On chain code probabilities and length estimators for digitized three dimensional curves. 1995.
20. R. Klette and T. Bülow. Minimum-length polygons in simple cube-curves. *Discrete Geometry for Computer Imagery*, pages 467–478, 2000.
21. R. Klette, V. Kovalevsky, and B. Yip. Lenth estimation of digital curves. In *Vision Geometry VIII - SPIE*, pages 117–129, Denver, July 1999.
22. R. Klette and J. Zunic. Convergence of calculated features in image analysis. Technical Report CITR-TR-52, Computer Science Departement of The University of Auckland, 1999.
23. J. Koplowitz and A. M. Bruckstein. Design of perimeter estimators for digitized planar shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(6):611–622, jun 1989.
24. V. Kovalevsky and S. Fuchs. Theoritical and experimental analysis of the accuracy of perimeter estimates. In *Robust Computer Vision*, pages 218–242, 1992.
25. M. Lindenbaum and A. Bruckstein. On recursive, o(n) partitioning of a digitized curve into digital straigth segments. *IEEE Transactions on PatternAnalysis and Machine Intelligence*, PAMI-15(9):949–953, september 1993.
26. J.P. Reveillès. *Géométrie discrète, calculs en nombre entiers et algorithmique*. PhD thesis, Thèse d'état, Université Louis Pasteur, Strasbourg, 1991.
27. F. Sloboda, B. Zatko, and P. Ferianc. *Advances in Digital and Computational Geometry*, chapter On approximation of planar one-dimensional continua. Springer, 1998.
28. F. Sloboda, B. Zatko, and R. Klette. On the topology of grid continua. In *Vision Geometry VII*, volume SPIE Volume 3454, pages 52–63, San Diego, July 1998.
29. I. Stojmenović and R. Tosić. Digitization schemes and the recognition of digital straight lines, hyperplanes and flats in arbitrary dimensions. In *Vision Geometry,contemporary Mathematics Series*, volume 119, pages 197–212, American Mathematical Society, Providence, RI, 1991.
30. A. Vialard. Geometrical parameters extraction from discrete paths. *Discrete Geometry for Computer Imagery*, 1996.
31. B. Vidal. *Vers un lancer de rayon discret.* PhD thesis, Thèse de Doctorat, Lille, 1992.