



HAL
open science

Curvature and torsion estimators for 3D curves

Thanh Phuong Nguyen, Isabelle Debled-Rennesson

► **To cite this version:**

Thanh Phuong Nguyen, Isabelle Debled-Rennesson. Curvature and torsion estimators for 3D curves. 4th International Symposium on Advances in Visual Computing - ISVC 2008, Dec 2008, Las Vegas, United States. pp.688-699, 10.1007/978-3-540-89639-5_66 . hal-00345708

HAL Id: hal-00345708

<https://hal.science/hal-00345708v1>

Submitted on 9 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Curvature and torsion estimators for 3D curves ^{*}

Thanh Phuong NGUYEN and Isabelle DEBLED-RENNESON

LORIA

Campus Scientifique - BP 239
54506 Vandoeuvre-lès-Nancy Cedex, France
{nguyentp,debled}@loria.fr

Abstract. We propose a new torsion estimator for spatial curves based on results of discrete geometry that works in $O(n \log^2 n)$ time. We also present a curvature estimator for spatial curves. Our methods use the 3D extension of the 2D blurred segment notion [1]. These estimators can naturally work with disconnected curves.

1 Introduction

Geometric properties of curves are important characteristics to be exploited in geometric processing. They directly lead to applications in machine vision [2] and computer graphics [3]. In the planar case, many applications are based on the curvature property in domains such as curve approximation [4], geometry compression [3], and particularly in corner detection after the pioneer paper of Attneave [5].

In 3D space, torsion and curvature are the most important properties that permit to describe how a spatial curve bends. Several methods have been proposed for torsion estimation. Mokhtarian [6] used Gaussian smoothing to estimate it directly from torsion formula. Similarly, Kehtarnavaz et. al. [7] used B-spline smoothing technique; Lewiner et al. [3] utilized weighted least-squares fitting techniques. Raluben Medina et al. [8] proposed two methods to estimate torsion and curvature values at each point of the curve. The first one utilized Fourier transform, the second one is based on the least squares fitting. These methods are applied for description of arteries in medical imaging.

We propose in this paper a novel method for the estimation of local geometric parameters of a spatial curve. It uses a geometric approach and relies on results of discrete geometry on decomposition of a curve into maximal blurred segments [9, 1, 10]. This paper presents an extension to 3D of these results. The 3D curvature estimator given in [11] is extended here with the notion of blurred segment and it permits to study curves possibly noisy or disconnected. We also propose a new approach to the discrete torsion estimation.

We recall, in the Section II, 2D definitions and results [10] that we use. Section III presents how to extend these ideas into 3D space. Sections IV and V respectively

^{*} This work is supported by ANR in the framework of the GEODIB project, *BLAN06 – 2_134999*

propose a curvature and a torsion estimator. Last sections give experiments and conclusions.

2 Maximal 2D blurred segment of width ν

The notion of blurred segments relies on the arithmetical definition of discrete lines [12] where a line, with slope $\frac{a}{b}$, is the set of integer points (x, y) verifying $\mu \leq ax - by < \mu + \omega$ (a, b, μ and ω being integer and $\gcd(a, b) = 1$). Such a line is denoted by $\mathcal{D}(a, b, \mu, \omega)$. The notion of 2D blurred segment extends the notion of segment of a discrete line and permits more flexibility in operations such as recognition, segmentation of discrete curves. Let us recall definitions [1, 10] that we use in this paper (see also the Fig. 1).

Definition 1. Let \mathcal{S}_b be a sequence of digital points.

A discrete line $\mathcal{D}(a, b, \mu, \omega)$ is said **optimal** for \mathcal{S}_b if each point of \mathcal{S}_b belongs to \mathcal{D} and if its vertical width, $\frac{\omega-1}{\max(|a|, |b|)}$, is equal to the vertical width of the convex hull of \mathcal{S}_b (see the Fig. 1.a).

\mathcal{S}_b is a **blurred segment of width ν** iff there exists an optimal discrete line $\mathcal{D}(a, b, \mu, \omega)$ of \mathcal{S}_b such that $\frac{\omega-1}{\max(|a|, |b|)} \leq \nu$.

Let \mathcal{C} be a discrete curve and $C_{i,j}$ a sequence of points of \mathcal{C} indexing from i to j . Suppose that the predicate " $C_{i,j}$ is a blurred segment of width ν " is denoted by $BS(i, j, \nu)$.

Definition 2. $C_{i,j}$ is called a **maximal blurred segment of width ν** and noted $MBS(i, j, \nu)$ iff $BS(i, j, \nu)$, $\neg BS(i, j + 1, \nu)$ and $\neg BS(i - 1, j, \nu)$ (see the Fig. 1.b).

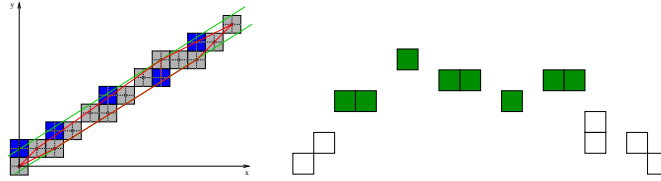


Fig. 1. From left to right: a. $\mathcal{D}(5, 8, -8, 11)$ (blue and grey points) is the optimal discrete line of the sequence of grey points, b. the set of black points is a maximal blurred segment (MBS) of width 2.

In [10] an algorithm is proposed to decompose a planar curve into maximal blurred segments for a given width and the theorem below is proved. The algorithm relies on operations of insertion (or deletion) of a point to (or from) the convex hull of the current studied segment. We have proven that the decomposition of a planar curve with n points into maximal blurred segments of width ν can be done in time $O(n \log^2 n)$.

3 Maximal 3D blurred segment of width ν

3.1 3D blurred segment of width ν

The notion of 3D discrete line (see the references [13, 14]) is defined as follows:

Definition 3. A **3D discrete line** [43], denoted $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$, with a main vector (a, b, c) such that $(a, b, c) \in \mathbb{Z}^3$, and $a \geq b \geq c$ is defined as the set of points (x, y, z) from \mathbb{Z}^3 verifying:

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

with $\mu, \mu', e, e' \in \mathbb{Z}$. e and e' are called arithmetical width of \mathcal{D} .

According to the definition, it is obvious that a 3D discrete line is bijectively projected into two projection planes as two 2D arithmetical discrete lines. Thanks to that property, we naturally define the notion of 3D blurred segment by using the notion of 2D blurred segment and by considering the projections of the sequence of studied points in the coordinate planes (see the Fig. 2.a).

Definition 4. Let $\mathcal{S}f_{3D}$ be a sequence of points of \mathbb{Z}^3 , $\mathcal{S}f_{3D}$ is a **3D blurred segment of width ν** with a main vector (a, b, c) such that $(a, b, c) \in \mathbb{Z}^3$, and $a \geq b \geq c$ if it possesses a said optimal discrete line, named $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$, such that

- $\mathcal{D}(a, b, \mu', e')$ is optimal for the sequence of projections of points of $\mathcal{S}f_{3D}$ in the plane (O, x, y) and $\frac{e'-1}{\max(|a|, |b|)} \leq \nu$,
- $\mathcal{D}(a, c, \mu, e)$ is optimal for the sequence of projections of points of $\mathcal{S}f_{3D}$ in the plane (O, x, z) and $\frac{e-1}{\max(|a|, |c|)} \leq \nu$.

A **linear algorithm of 3D blurred segment recognition** may be deduced from that definition. Indeed, we only need to use an algorithm of 2D blurred segment recognition in each projection plane.

3.2 Maximal 3D blurred segment of width ν

In this section, we present an algorithm to obtain the sequence of 3D maximal blurred segments of width ν in time $O(n \log^2 n)$ for any noisy 3D discrete curve \mathcal{C} . This sequence is noted $MBS_\nu(\mathcal{C}) = \{MBS_i(B_i, E_i, \nu)\}_{i=0 \text{ to } m-1}$ with B_i (resp. E_i) the index of the first (resp. last) point of the i th maximal blurred segment, MBS_i , of \mathcal{C} . This algorithm uses an algorithm to determine the 2D maximal blurred segment (see [10]) of the projections in the coordinate planes of the points of the studied curve.

Algorithm 1: Algorithm for the segmentation of a curve C into maximal 3D blurred segments of width ν

Data: C - discrete curve with n points, ν - width of the segmentation
Result: MBS_ν - the sequence of maximal blurred segments of width ν of C

```

begin
  k=0;  $S_b = \{C_0\}$ ;  $MBS_\nu = \emptyset$ ;  $a = 0$ ;  $b = 1$ ;  $\omega = b$ ,  $\mu = 0$ ;
  while the widths of 2 blurred segments obtained by projecting the points of
   $S_b$  in the coordinate planes are  $\leq \nu$  do
    k++;  $S_b = S_b \cup C_k$ ;
    Determine  $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$ , optimal discrete line of  $S_b$ ; (*)
  end
  bSegment=0; eSegment=k-1;
   $MBS_\nu = MBS_\nu \cup MBS(bSegment, eSegment, \nu)$ ;
  while  $k < n - 1$  do
    while the widths of 2 blurred segments obtained by projecting the points
    of  $S_b$  in the coordinate planes are  $> \nu$  do
       $S_b = S_b \setminus C_{bSegment}$ ; bSegment++;
      Determine  $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$ , optimal discrete line of  $S_b$ ; (*)
    end
    while the widths of 2 blurred segments obtained by projecting the points
    of  $S_b$  in the coordinate planes are  $\leq \nu$  do
      k++;  $S_b = S_b \cup C_k$ ;
      Determine  $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$ , optimal discrete line of  $S_b$ ; (*)
    end
    eSegment=k-1;  $MBS_\nu = MBS_\nu \cup MBS(bSegment, eSegment, \nu)$ ;
  end
end

```

(*) To determine the optimal discrete line of the current 3D blurred segment S_b , we consider the characteristic of the two 2D blurred segments obtained in the planes of projection and combine them to obtain the characteristics of the optimal 3D discrete line of S_b . As the whole process is done in dimension 2, this algorithm has the same complexity as the one in dimension 2. So, we have the following result :

The decomposition of a 3D curve into maximal blurred segments of width ν can be done in time $O(n \log^2 n)$.

4 3D discrete curvature of width ν

In this section, we're interested in curvature estimation based on osculating circle. Deducing from [10], we present below the notion of discrete curvature of width ν at each point of a 3D curve (see the Fig. 2.b). This method extends the one proposed in [11] to the blurred segments and is adapted to noisy curves thanks to the width parameter.

Let C be a 3D discrete curve, C_k is a point of the curve. Let us consider the

points C_l and C_r of \mathcal{C} such that : $l < k < r$, $BS(l, k, \nu)$ and $\neg BS(l - 1, k, \nu)$, $BS(k, r, \nu)$ and $\neg BS(k, r + 1, \nu)$. The points C_l and C_r for a given point C_k of \mathcal{C} are deduced from the sequence of maximal blurred segments of width ν of \mathcal{C} . The estimation of the **3D curvature of width ν at the point C_k** is determined thanks to the radius of the circle passing through the points C_l , C_k and C_r . To determine the radius $R_\nu(C_k)$ of the circumcircle of the triangle $[C_l, C_k, C_r]$, we use the formula given in [15] as follows:

Let $s_1 = \|\overrightarrow{C_k C_r}\|$, $s_2 = \|\overrightarrow{C_k C_l}\|$ and $s_3 = \|\overrightarrow{C_l C_r}\|$, then

$$R_\nu(C_k) = \frac{s_1 s_2 s_3}{\sqrt{(s_1 + s_2 + s_3)(s_1 - s_2 + s_3)(s_1 + s_2 - s_3)(s_2 + s_3 - s_1)}}$$

Then, the curvature of width ν at the point C_k is $\mathcal{C}_\nu(C_k) = \frac{s}{R_\nu(C_k)}$ with $s = \text{sign}(\det(\overrightarrow{C_k C_r}, \overrightarrow{C_k C_l}))$ (it indicates concavities and convexities of the curve).

Thanks to the sequence of maximal blurred segments of width ν (MBS_ν) of a 3D curve \mathcal{C} , an algorithm for curvature estimation at each point of a 3D curve can be directly deduced from [10].

Algorithm 2: Width ν curvature estimation at each point of ζ

Data: \mathcal{C} 3D discrete curve of n points, ν width of the segmentation

Result: $\{\mathcal{C}_\nu(C_k)\}_{k=0..n-1}$ - Curvature of width ν at each point of \mathcal{C}

begin

 Build $MBS_\nu = \{MBS_i(B_i, E_i, \nu)\}_{i=0 \text{ to } m-1}$ (See the Algo. 1); $m = |MBS_\nu|$; $E_{-1} = -1$; $B_m = n$;

for $i = 0$ **to** $m - 1$ **do**

for $k = E_{i-1} + 1$ **to** E_i **do** $L(k) = B_i$;

for $k = B_i$ **to** $B_{i+1} - 1$ **do** $R(k) = E_i$;

end

for $i = 0^{(*)}$ **to** $n - 1^{(*)}$ **do**

$R_\nu(C_i) = \text{Radius of the circumcircle to } [C_{L(i)}, C_i, C_{R(i)}]$;

$\mathcal{C}_\nu(C_i) = \frac{s}{R_\nu(C_i)}$;

end

end

(*) The bounds mentioned in the algorithm 2 are correct for a closed curve. In case of an open curve, the instruction becomes: *for* $i = l$ *to* $n - 1 - l$ with l fixed to a constant value.

5 Discrete torsion of width ν

5.1 Preliminary

The 3D curvature is not sufficient to characterize the local property of a 3D curve. This parameter only measures how rapidly the direction of the curve changes. In case

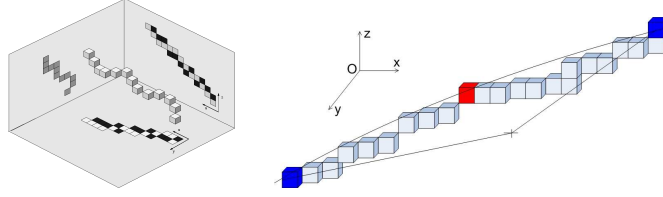


Fig. 2. From left to right: a. $\mathcal{D}_{3D}(45, 27, 20, -45, -81, 90, 90)$ optimal discrete line of the grey points, b. The curvature at the red point is defined as the inverse of circumcircle (passing through both blue points and the red point) radius.

of a planar curve, the osculating plane does not change. For 3D curves, torsion is a parameter that measures how rapidly the osculating plane changes. To clarify this notion, we recall below some definitions and results in differential geometry (see [16] for more details).

Definition 5. Let $r : I \rightarrow \mathbb{R}^3$ be a regular unit speed curve parameterized by t .

- i $T(t)$ (resp. $N(t)$) a unit vector in direction $r'(t)$ (resp. $r''(t)$). So, $N(t)$ is a normal vector to $T(t)$. $T(t)$ (resp. $N(t)$) is called the unit **tangent vector** (resp. **normal vector**) at t .
- ii $|T'(t)| = k(t)$ is called the **curvature** of r at t .
- iii The plane determined by the unit tangent and normal vectors ($T(t)$ and $N(t)$), is called the **osculating plane** at t . The unit vector $B(t) = T(t) \wedge N(t)$ is normal to the osculating plane and is called the **binormal vector** at t .
- iv $|B'(t)| = \tau(t)$ is called the **torsion** of curve at t .

Theorem 1 Let $r : I \rightarrow \mathbb{R}^3$ be a spatial curve parameterized by t .

- i The curvature of r at $t \in I$: $k(t) = \frac{|r' \wedge r''|}{|r'|^3}$
- ii The torsion of r at $t \in I$: $\tau(t) = \frac{(r' \wedge r'').r'''}{|r' \wedge r''|^2}$

Thanks to theorem 3, the torsion value at a point is 0 if the curvature value at this point is 0.

5.2 Discrete torsion

Discrete torsion was studied in [3, 6, 8, 7]. In this section, we propose a new geometric approach for the problem of torsion estimation that uses the definitions and results presented in the previous sections.

DEFINITIONS

Let ζ be a 3D discrete curve, C_k is k^{th} point of the curve. Let us consider the points C_l and C_r of ζ such that $l < k < r$, $BS(l, k, \nu) \& \neg BS(l - 1, k, \nu)$ and $BS(k, r, \nu) \& \neg BS(k, r + 1, \nu)$. Let's recall that the curvature of width ν (see section 4.) is estimated by circumcircle of triangle $\triangle C_l C_k C_r$. If $\overrightarrow{C_k C_l}$ and $\overrightarrow{C_k C_r}$ are colinear,

the curvature value at C_k is 0, therefore the torsion value at C_k is 0. So, without loss of generality, we suppose that $\overrightarrow{C_l C_k}$ and $\overrightarrow{C_k C_r}$ are not colinear. In addition, the plane defined by $\overrightarrow{C_l C_k}$ and $\overrightarrow{C_k C_r}$ is noted (C_l, C_k, C_r) , and we propose the definition below.

Definition 6. The *osculating plane of width ν* at C_k is the plane (C_l, C_k, C_r) .

The osculating plane (C_l, C_k, C_r) has two unit tangent vectors : $\overrightarrow{t1} = \frac{\overrightarrow{C_l C_k}}{|\overrightarrow{C_l C_k}|}$ and $\overrightarrow{t2} = \frac{\overrightarrow{C_k C_r}}{|\overrightarrow{C_k C_r}|}$. Therefore, we have the binormal vector at the k^{th} point: $\overrightarrow{b_k} = \overrightarrow{t1} \wedge \overrightarrow{t2} = (b_x, b_y, b_z)$ So, we propose the following definition of discrete torsion of width ν .

Definition 7. The *discrete torsion of width ν* at C_k is the derivation of $\overrightarrow{b_k}$

TORSION ESTIMATOR Our proposed method for torsion estimation is based on the definition 7. Let us remark that the set $\{\overrightarrow{b_k}\}_{k=0}^{n-1}$ can be constructed from the set of maximal blurred segments in $O(n \log^2 n)$ time. So, we can obtain torsion value by calculating the derivative at each position of $\{\overrightarrow{b_k}\}_{k=0}^{n-1}$. The traditional method for derivation estimation of discrete sequence is utilizing Gaussian kernel [17]. We propose below a geometric approach method to this problem.

Let us consider the curve $\zeta_1 = \{P\}_{i=0}^n$ that is constructed by this rule: $\overrightarrow{P_i P_{i+1}} = \overrightarrow{b_i}$, $i = 0, \dots, n - 1$ (see the Fig. 3).

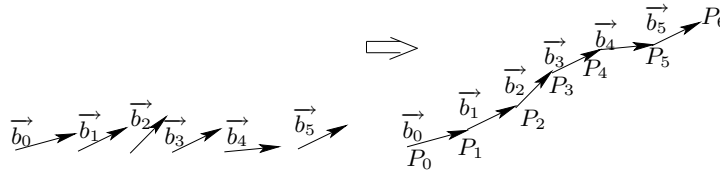


Fig. 3. The curve ζ_1 is constructed from the sequence of binormal vector

Proposition 1 The tangent vector at each point P_i of the curve ζ_1 is $\overrightarrow{b_i}$ ($i = 0, \dots, n - 1$).

Proof. In differential geometry, the tangent vector of a curve $r(t)$ at the point $P_{t_0} = r(t_0)$ is defined as: $t(t_0) = r'(t_0) = \lim_{h \rightarrow 0} \frac{r(t_0+h) - r(t_0)}{h} = \lim_{h \rightarrow 0} \frac{\overrightarrow{P_{t_0+h} P_{t_0}}}{h}$. Therefore, in discrete space, the tangent vector at the point $P_i = \alpha(i)$ can be estimated as $t(i) = \frac{r(i+1) - r(i)}{1} = \frac{\overrightarrow{P_{i+1} P_i}}{1} = \overrightarrow{P_i P_{i+1}} = \overrightarrow{b_i}$.

Proposition 2 The torsion value at each point of curve ζ corresponds to curvature value of ζ_1 curve.

Proof. Thanks to definition 8, the discrete torsion at C_k of ζ curve is the derivation of $\overrightarrow{b_k}$. In addition, $\overrightarrow{b_k}$ is the tangent vector at the k^{th} point of ζ_1 curve. So, this value is also curvature value at the k^{th} point of ζ_1 curve.

Therefore, by using these two propositions, we can estimate torsion value at each point of ζ curve by determining curvature value at corresponding point of ζ_1 curve. Our proposed method is presented in the algorithm 3, it uses the curvature estimator presented in section 3.3.

(*) Same remark as for the Algorithm 2.

Algorithm 3: Width ν torsion estimation at each point of ζ

Data: ζ 3D discrete curve of n points, ν width of the segmentation

Result: $\{\mathcal{T}_\nu(C_k)\}_{k=0..n-1}$ - Torsion of width ν at each point of ζ

begin

 Build $MBS_\nu = \{MBS_i(B_i, E_i, \nu)\}_{i=0 \text{ to } m-1}$;

$m = |MBS_\nu|$; $E_{-1} = -1$; $B_m = n$;

for $i = 0$ **to** $m - 1$ **do**

for $k = E_{i-1} + 1$ **to** E_i **do** $L(k) = B_i$;

for $k = B_i$ **to** $B_{i+1} - 1$ **do** $R(k) = E_i$;

end

for $i = 0^{(*)}$ **to** $n - 1^{(*)}$ **do**

$\vec{t1} = \frac{C_i C_{L(i)}}{|C_i C_{L(i)}|}$; $\vec{t2} = \frac{C_i C_{R(i)}}{|C_i C_{R(i)}|}$; $\vec{b}_i = \vec{t1} \wedge \vec{t2}$;

end

 Construct $\zeta_1 = \{P_k\}_{k=0}^n$, with $\overrightarrow{P_k P_{k+1}} = \vec{b}_k$;

 Estimate the curvature value of width ν at each point of the ζ_1 curve as torsion value of corresponding point of the ζ curve (see the Algo. 2);

end

6 Experiments

We introduce some experiments of our method on some ideal 3D curves : helix, Viviani's, spheric, horopter and hyper helix curves. The tests are done after a process of discretisation of these 3D curves (see the Fig. 6). We have tested our methods on this computer configuration: CPU Pentium 4 with 3.2GHz of speed, 1Gb of memory RAM, linux kernel 2.6.22-14 operating system.

We introduce three criteria for measuring error: mean relative error (meanRE), max relative error (maxRE) and quadratic relative error (QRE). Let's consider 2 sequences: $\{IR_i\}_{i=1}^n$ (resp. $\{RR_i\}_{i=1}^n$) the ideal result (resp. estimated result) at each position. So, we have:

$$\text{meanRE} = \frac{1}{n} \sum_{i=1}^n \frac{|RR_i - IR_i|}{IR_i}, \quad \text{maxRE} = \max \left\{ \frac{|RR_i - IR_i|}{IR_i} \right\}, \quad i = 1, \dots, n \quad \text{and} \quad \text{QRE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{|RR_i - IR_i|}{IR_i} \right)^2}.$$

Because the estimated result is not correct for the beginning and the end of the open curve (see the bounds mentioned in the algorithms 2 and 3), during the phase of error estimation we use the border parameter to eliminate this influence.

In most cases of the studied curves (see the Table 1 and Fig. 6), the mean relative errors do not overtake 0.15, and the quadratic relative errors do not overtake 0.015. If

the ideal torsion of the input curve has a value which is close to 0 at some positions, the obtained result is not very good. Let's see the case of Viviani's curve in the Table 1. In this case, the maximal relative error is high (15.6036). In spite of that, the mean relative error is acceptable (0.628899). In particular cases, if most of input curves has a torsion value which is close to 0, the obtained result is the worst (see the Fig. 4).

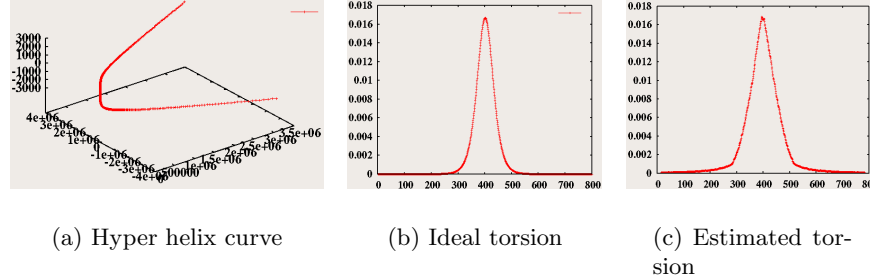


Fig. 4. Most of the hyper helix curve has a torsion value close to 0. So in this case, the obtained result is the worst.

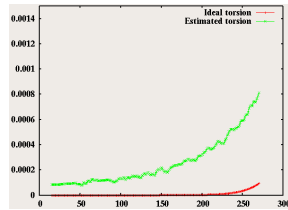


Fig. 5. Relatif error

Let's consider the case of an hyper helix curve (see the Fig. 4). The problem is that the torsion approximation is not good at nearly-0 values. In spite of that, the approximation value is also close to 0 but relative rate between approximation value and ideal value is very high. In Fig. 5, we show the relation between approximation torsion and ideal torsion of an hyper helix curve from the index 15 and to the index 250. In this index interval, the ideal torsion is close to 0. So, the relative error between approximation torsion and ideal torsion is very high, in spite of that the approximation value does not overtake 0.006. So, we propose to consider only the points whose ideal torsion value is greater than the threshold. In the error calculus, n

is replaced by number of points whose ideal torsion value is greater than a threshold. The table 2 shows the approximated error with a threshold equal to 0.0005.

7 Conclusions

We have presented in this paper 2 methods to estimate curvature and torsion of a 3D curve. These methods benefit from the amelioration of curvature estimator in planar case [10], so they're efficient. These estimators permit to discover local properties of spatial curve. We hope to identify and classify 3D objects by using these estimators. In the future, we will compare our methods with other existing methods [6, 11, 2, 3] for curvature and torsion estimation. Moreover, we will work with real discrete data of biology or medical area. We intend to present these works in a journal version.

<i>Curves</i>	<i>No of points</i>	<i>Border</i>	<i>meanRE</i>	<i>maxRE</i>	<i>QRE</i>	<i>Time (ms)</i>
Spheric	255	30	0.13174	0.34281	0.164156	280
Horopter	239	30	0.0827392	0.186243	0.0979267	300
Helix	760	30	0.0481056	0.514517	0.0813778	920
Viviani	274	30	0.628899	15.6036	1.62278	290
Hyper helix	740	30	8551.24	154378	24704.2	720

Table 1. Error estimation on the estimated torsion result

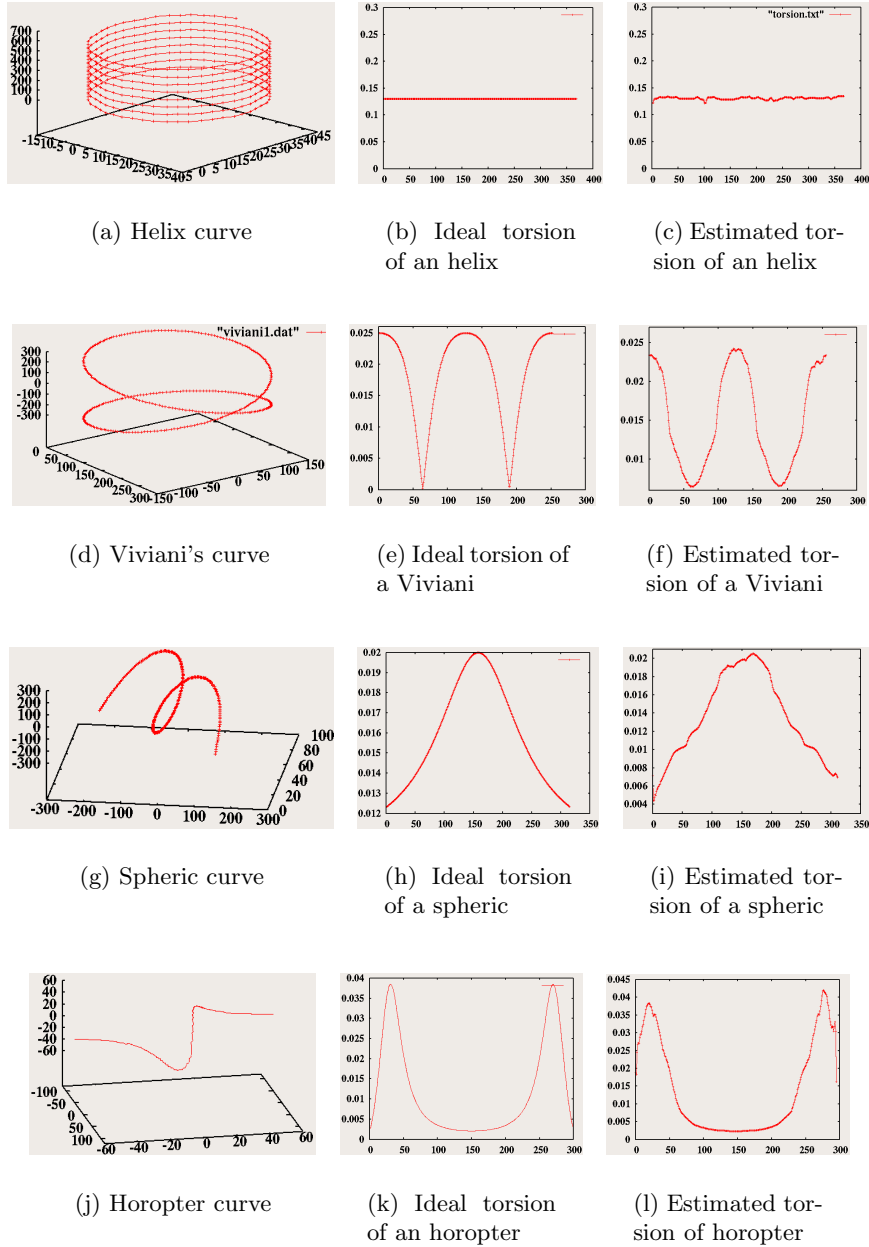


Fig. 6. Experiments with width $\nu=2$

<i>Curves</i>	<i>No points</i>	<i>No considered points</i>	<i>Border</i>	<i>meanRE</i>	<i>maxRE</i>	<i>QRE</i>
Spheric	255	255	30	0.13174	0.34281	0.164156
Horopter	239	239	30	0.0827392	0.186243	0.0979267
Helix	760	760	30	0.0481056	0.514517	0.0813778
Viviani	274	270	30	0.456954	3.51278	0.615561
Hyper helix	740	196	30	1.15266	3.80191	1.62102

Table 2. Error estimation on the estimated torsion result, threshold = 0.0005

References

1. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. *Computers & Graphics* **30** (2006)
2. Poyato, Á.C., García, N.L.F., Carnicer, R.M., Madrid-Cuevas, F.J.: A method for dominant points detection and matching 2d object identification. In: *ICIAR* (1). (2004) 424–431
3. Lewiner, T., Jr., J.D.G., Lopes, H., Craizer, M.: Curvature and torsion estimators based on parametric curve fitting. *Computers & Graphics* **29** (2005) 641–655
4. Salmon, J.P., Debled-Rennesson, I., Wendling, L.: A new method to detect arcs and segments from curvature profiles. In: *ICPR* (3). (2006) 387–390
5. Attneave, E.: Some informational aspects of visual perception. *Psychol. Rev.* **61** (1954)
6. Mokhtarian, F.: A theory of multiscale, torsion-based shape representation for space curves. *Computer Vision and Image Understanding* **68** (1997) 1–17
7. Kehtarnavaz, N.D., de Figueiredo, R.J.P.: A 3-d contour segmentation scheme based on curvature and torsion. *IEEE Trans. Pattern Anal. Mach. Intell.* **10** (1988) 707–713
8. Medina, R., Wahle, A., Olszewski, M.E., Sonka, M.: Curvature and torsion estimation for coronary-artery motion analysis. In: *SPIE Medical Imaging*. Volume 5369. (2004) 504–515
9. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: Application to the curvature. In: *DGCI*. Volume 1568 of *LNCS*. (1999) 31–40
10. Nguyen, T.P., Debled-Rennesson, I.: Curvature estimation in noisy curves. In: *CAIP*. (2007) 474–481
11. Coeurjolly, D., Svensson, S.: Estimation of curvature along curves with application to fibres in 3d images of paper. *LNCS* **2749** (2003) 247–254
12. Reveillès, J.P.: *Géométrie discrète, calculs en nombre entiers et algorithmique* (1991) Thèse d'état. Université Louis Pasteur, Strasbourg.
13. Debled-Rennesson, I.: *Reconnaissance des droites et plans discrets*. PhD thesis, Université Louis Pasteur (1995)
14. Coeurjolly, D., Debled-Rennesson, I., Teytaud, O.: Segmentation and length estimation of 3d discrete curves. In: *Digital and Image Geometry*. Volume 2243 of *LNCS*, Springer (2000) 299–317
15. Harris, J., Stocker, H.: *Handbook of mathematics and computational science*, Springer-Verlag. (1998)
16. Oprea, J.: *Differential geometry and its applications*. (2007)
17. Worring, M., Smeulders, A.W.M.: Digital curvature estimation. *Computer Vision Graphics Image Processing: CVIU* **58** (1993) 366–382