

# Geometric measures on arbitrary dimensional digital surfaces

Jacques-Olivier Lachaud and Anne Vialard

LaBRI, Univ. Bordeaux 1, 351 cours de la Libération, 33405 Talence, France.  
lachaud@labri.fr, vialard@labri.fr

**Abstract.** This paper proposes a set of tools to analyse the geometry of multidimensional digital surfaces. Our approach is based on several works of digital topology and discrete geometry: representation of digital surfaces, bel adjacencies and digital surface tracking, 2D tangent computation by discrete line recognition, 3D normal estimation from slice contours. The main idea is to notice that each surface element is the crossing point of  $n - 1$  discrete contours lying on the surface. Each of them can be seen as a 4-connected 2D contour. We combine the directions of the tangents extracted on each of these contours to compute the normal vector at the considered surface element. We then define the surface area from the normal field. The presented geometric estimators have been implemented in a framework able to represent subsets of  $n$ -dimensional spaces. As shown by our experiments, this generic implementation is also efficient.

## 1 Introduction

Many applications in the image analysis field need to represent and manipulate regions defined as subsets of  $n$ -dimensional images. Moreover, it is often necessary to perform geometric measurements on these regions and on the digital surfaces that form their boundaries. Classically, geometric estimators are defined over frontiers in 2D or 3D images. In this paper, we present a set of tools for the analysis of the geometry of arbitrary dimensional digital surfaces. This work is based on a concise coding of the cells of  $n$ -dimensional finite regular grids [8]. This coding induces a generic and efficient framework for implementing classical digital topology data structures and algorithms. We show here that this framework is also suited to defining the geometry of digital surfaces, namely by a careful use of digital surface tracking. Note that we do not compare our work with geometric definitions based on a continuous approximation of digital sets. Our topological and geometric definitions are purely discrete. Furthermore, they are much easier to define and compute in arbitrary dimension.

Some authors define arbitrary dimensional digital surfaces as set of *spels* (pixels in 2D, voxels in 3D,  $n$ -cells in  $n$ D) with specific properties [11]. However, frontiers of regions in images are generally not digital surfaces in this sense. Moreover, it is not clear how to extend classical 2D and 3D discrete geometry estimators to these surfaces. This paper is concerned with digital surfaces that

are defined as subsets of the cellular decomposition of  $\mathbb{R}^n$  into a regular grid (pixel edges in 2D, voxel faces or surfels in 3D). This space was introduced in image analysis by Kovalevsky [6].

There are several approaches to defining discrete geometric estimators on digitized objects (e.g. see [3] for a recent survey). Our approach for tangent and normal estimation follows the basic idea of “slice” decomposition proposed by Lenoir *et. al.* [10]. Intuitively, there are  $n - 1$  orthogonal 2D planes containing the point of interest. The intersection of each of those planes with the digitized object forms a contour on which a 2D tangent is computed. Lenoir builds the 3D normal as the vector product of the two extracted tangents. Tellier and Debled–Renneson [13] proposed a similar technique where the tangent is defined as a discrete line segment. This paper extends these two works to arbitrary dimensional digital surfaces. We use the obtained normal estimator to compute the area of a digital surface. This definition coincides with the one proposed by Lenoir [9] in 3D.

The paper is organized as follows. First we show how to represent boundaries of digital objects in arbitrary dimension as a set of surface elements (*surfels*) with a topology. This representation allows the definition of  $n - 1$  contours around each surfel. Secondly we define a discrete tangent at a surfel on each of these contours and detail its computation algorithm. In the last section we combine these 2D information to obtain  $n$ D estimators (normal vector, elementary area, surface area). The presented material has been implemented in  $n$ D. We show its efficiency on some experiments. All the necessary information to reimplement it are provided.

## 2 Representation and properties of digital surfaces

In this paper, we are interested in computing geometric characteristics of (oriented) digital surfaces that are boundaries of sets of spels. However, all the presented material is adaptable to any kind of digital surfaces (open or not, orientable or not) with little work. In this section, we assume we are working in a finite  $n$ -dimensional image forming a parallelepiped in  $\mathbb{Z}^n$ . We denote by  $M^i$  the inclusive upper bound for the  $i$ -th coordinate of any spel. All coordinates have 0 as lower bound.

### 2.1 Cell coding

There is an isomorphism between the cellular decomposition  $\mathbb{C}^n$  of  $\mathbb{R}^n$  into a regular grid and the  $n$ -dimensional Khalimisky space  $\mathbb{K}^n$  [5]. This space is the cartesian product of  $n$  connected ordered topological spaces (COTS). A COTS can be seen as a set of ordered discrete points, like  $\mathbb{Z}$ , whose topology alternates closed points and open points. If we define even points of  $\mathbb{Z}$  as closed and odd points of  $\mathbb{Z}$  as open, each point of  $\mathbb{K}^n$  is then identified by its  $n$  integer coordinates, whose parities define its topological properties.

Consequently, any cell  $c$  of  $\mathbb{C}^n$  has exactly one corresponding point in  $\mathbb{K}^n$  with coordinates  $(x_K^0, \dots, x_K^{n-1})$ . We propose to code any (unoriented) cell  $c$  as one binary word  $\boxed{\alpha | x^{n-1} | \dots | x^i | \dots | x^0}$ , called the *unsigned code* of  $c$ , as follows:

- The  $i$ -th coordinate  $x_K^i$  is coded by its binary decomposition after a rightshift ( $x^i = x_K^i \text{ div } 2$ ). We say that  $x^i$  is the  *$i$ -th digital coordinate* of  $c$ .
- All coordinates are packed as one binary word (from  $x^{n-1}$  to  $x^0$ ). Every coordinate is allocated a fixed number of bits  $N_i$  given by  $N_i = \log_2(M^i) + 1$ .
- The parities of all coordinates are also packed as an  $n$ -bits word  $\alpha$  with  $\alpha = \sum_i (x_K^i \text{ mod } 2)2^i$ . The word  $\alpha$  is called the *topology* of  $c$ .

According to the isomorphism, cells of  $\mathbb{C}^n$  that are  $k$ -dimensional (or  *$k$ -cells*) have a topology word composed of  $k$  1's. The coordinate where a surfel  $c$  has a 0 in its topology word is called the coordinate *orthogonal* to the surfel  $c$  and is denoted by  $\perp(c)$ .

The cell topology (dimension, open or closed along a coordinate, adjacent and incident cells) and geometry (coordinates in  $\mathbb{Z}^n$ , centroid, trivial normal and tangent vectors) can be computed from the code without any further information. It has been shown in [7] that most basic operations on cells (e.g. adjacence, incidence) have an efficient implementation that is independent from the dimension of space. All specific subsets of  $\mathbb{C}^n$  (e.g., objects, digital surfaces, cubical complexes) have then an efficient and compact representation.

## 2.2 Oriented cells, boundary operators, bels, boundary of an object

To define some geometric characteristics (e.g. normal vector), a digital surface must be oriented (at least locally). It is thus convenient to associate an orientation to each cell of  $\mathbb{C}^n$ . We therefore define the *signed code* of an oriented cell  $c$  by adding an *orientation bit*  $s$  (0 for positive orientation and 1 for negative orientation) to its unsigned code as follows:  $\boxed{\alpha | s | x^{n-1} | \dots | x^i | \dots | x^0}$ . The *opposite cell*  $-c$  of  $c$  is the same cell as  $c$  but with opposite orientation.

With oriented cells, we can define boundary operators, which represent at the same time how cells are incident with each others and how orientations are propagated from one cell to another.

**Definition 1.** Let  $c = \boxed{i_k \dots i_j \dots i_0 | s | x^{n-1} | \dots | x^{i_j} | \dots | x^0}$  be any cell with topology bits set to 1 on the coordinates  $i_k, \dots, i_j, \dots, i_0, n-1 \geq i_k > \dots > i_j > \dots > i_0 \geq 0$  and the others bits set to 0. The symbol  $\hat{i}_j$  means that the bit  $i_j$  is set to 0. Let  $\tau = (-1)^{(k-j)}$ . The set  $\Delta_{i_j} c$  composed of the two oppositely signed cells  $\boxed{i_k \dots \hat{i}_j \dots i_0 | \tau s | x^{n-1} | \dots | x^{i_j} | \dots | x^0}$  and  $\boxed{i_k \dots \hat{i}_j \dots i_0 | -\tau s | x^{n-1} | \dots | x^{i_j} + 1 | \dots | x^0}$ , is called the lower boundary of the cell  $c$  along coordinate  $i_j$ . The lower boundary  $\Delta c$  of  $c$  is then the set of cells  $\cup_{l=0, \dots, k} \Delta_{i_l} c$ .

The lower boundary of a  $k$ -cell  $c$  thus corresponds to the set of  $k - 1$ -cells low incident to  $c$  with specific orientations (e.g. on Figure 1,  $+b'$  is the positively oriented 0-cell low incident to the 1-cell  $b$  along coordinate  $x$ ). The *upper boundary*  $\nabla$  of a cell is defined symmetrically (the upper boundary is taken on topology

bits set to 0). It can be shown that this definition of boundary operators induces that any cubical cell complex is a polyhedral complex.

In the remainder of the paper, the set  $O$  is an object of the image  $I$  with an empty intersection with the border of  $I$ . Assume that all spels of  $O$  are oriented positively. We merge the sets  $\Delta p$  with  $p \in O$  with the rule that two identical cells except for their orientation cancel each other. The resulting set of oriented surfels is called the *boundary of  $O$* , denoted by  $\partial O$ . It is an oriented digital surface, whose elements are called *bels* of  $O$ . This surface separates the object  $O$  from its complement [7].

### 2.3 Followers of surfel, bel adjacency, digital surface tracking

The bel adjacency defines the connectedness relations between bels bounding an object. It has two nice consequences: (i) the boundary of an object can be extracted by tracking the bels throughout their bel adjacencies [1]; (ii) sets of surfels can be considered as classical Euclidean surfaces, where one can move on the surface in different orthogonal directions (2 in 3D). The second property is thus essential for defining the geometry of digital surfaces. We start by defining which surfels are potentially adjacent to a given bel with the notion of follower. We then define two kinds of bel adjacency for each pair of coordinates.

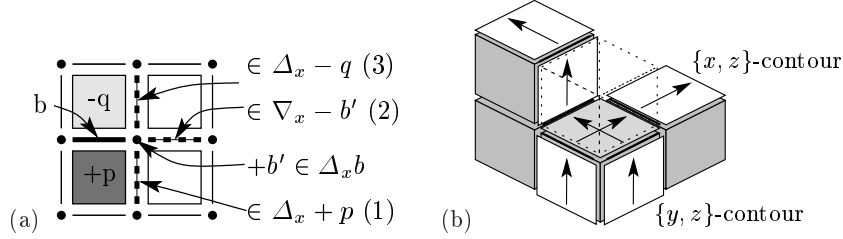
**Definition 2.** *We say that an oriented  $r$ -cell  $q$  is a direct follower of an oriented  $r$ -cell  $p$ ,  $p \neq \pm q$ , if  $\Delta p$  and  $\Delta q$  have a common  $r - 1$ -cell, called the direct link from  $p$  to  $q$ , such that this cell is positively oriented in  $\Delta p$  and negatively oriented in  $\Delta q$ . The cell  $p$  is then an indirect follower of  $q$ .*

It is easy to check that any surfel has 3 direct followers and 3 indirect followers along all coordinates except the one orthogonal to the surfel. We order the followers consistently for digital surface tracking (see Figure 1a).

**Definition 3.** *Let  $b$  be an oriented  $n - 1$ -cell with  $\nabla b = \{+p, -q\}$ . Let  $j$  be a coordinate with  $j \neq \perp(b)$ . The three direct followers of  $b$  along  $j$  are ordered as follows: (1) the first direct follower belongs to  $\Delta_j + p$ , (2) the second direct follower belongs to  $\nabla_j - b'$  with  $+b'$  direct link in  $\Delta_j b$ , (3) the third direct follower belongs to  $\Delta_j - q$ .*

Intuitively, when tracking a digital surface, you have 3 different possibilities for a move along a given coordinate. This is true for arbitrary dimension. The following definition shows which one to choose at each step. It is in agreement with the definitions of bel adjacencies proposed by Udupa [14], but easier to implement in our framework.

**Definition 4.** *Let  $b$  be a bel of  $\partial O$ , such that  $\nabla b = \{+p, -q\}$  (thus  $p \in O$  and  $q \notin O$ ). For any coordinate  $j \neq \perp(b)$ , the bel  $b$  has one interior direct adjacent bel (resp. exterior direct adjacent bel) which is the first (resp. last) of the three ordered direct followers of  $b$  along coordinate  $j$  that is a bel of  $\partial O$ . The bel adjacency is the symmetric closure of the direct bel adjacency.*



**Fig. 1.** (a) Direct followers of a surfel  $b$  along coordinate  $x$ . (b) The two direct contours crossing at a given surfel in 3D.

In 3D, the interior (resp. exterior) bel adjacency along all coordinates induces the classical (6,18) bel-adjacency (resp. (18,6) bel-adjacency). Interior and exterior bel adjacencies can be mixed for different coordinate pairs. This might be useful in an application where the image data are not isotropic (e.g., some CT scan images, confocal microscopy).

## 2.4 Contours over digital surfaces

The following definition is consistent since a direct follower  $c$  of a surfel  $b$  along a coordinate  $j \neq \perp(b)$  satisfies  $\perp(c) \in \{\perp(b), j\}$ .

**Definition 5.** Let  $S$  be a set of oriented surfels and  $i, j$  two distinct coordinates. A sequence of distinct surfels  $p_0, \dots, p_k$  in  $S$  is called a direct  $\{i, j\}$ -contour over  $S$  iff: (i)  $\forall 0 \leq l \leq k, \perp(p_l) \in \{i, j\}$ , and (ii)  $\forall 0 \leq l < k, p_{l+1}$  is a direct follower of  $p_l$  along the coordinate  $i$  or  $j$  different from  $\perp(p_l)$ .

The next propositions state that contours can be defined over boundaries of objects for any pair of coordinates and that these contours can be seen as 4-connected paths of pixels in the 2-dimensional plane that “contains” the contour (see Figure 1b for a 3D illustration). Proofs can be found in [7].

**Proposition 1.** Let  $b$  be any bel in  $\partial O$  and  $j$  any coordinate different from  $\perp(b)$ . The sequence  $(p_l)_{0 \leq l \leq k}$  of direct interior adjacent bels starting from  $b$  and going along either  $j$  or  $\perp(b)$  is a direct  $\{\perp(b), j\}$ -contour over  $\partial O$ . Note that  $p_0$  is the direct interior adjacent bel of  $p_k$ .

**Proposition 2.** Given a direct  $\{i, j\}$ -contour  $C$  over a set of oriented surfels  $S$  with  $C = (p_l)_{0 \leq l \leq k}$ , then the sequence  $D = (q_l)_{0 \leq l < k}$  of direct links from  $p_l$  to  $p_{l+1}$  has the following properties:

- (1) the  $n - 2$ -cells  $q_l$  are closed along coordinates  $i$  and  $j$ ,
- (2.i) for any coordinate  $d \notin \{i, j\}$ , the  $d$ -th coordinates of  $q_l$  and of  $q_{l+1}$  are equal,
- (2.ii) either the  $i$ -th coordinates of  $q_l$  and of  $q_{l+1}$  are equal and their  $j$ -th coordinates differ of  $\pm 1$ , or the  $j$ -th coordinates of  $q_l$  and of  $q_{l+1}$  are equal and their  $i$ -th coordinates differ of  $\pm 1$ .

As a corollary,  $D$  can be seen as a 4-connected contour in  $\mathbb{Z}^2$  if we forget all the coordinates different from  $i$  and  $j$  in the cells of  $D$ .

### 3 Discrete 2D tangent over a 4-connected contour

From the last proposition, we can trace from any bel  $b$   $n - 1$  contours  $C_j(b)$ ,  $j \neq \perp(b)$ , on the boundary  $\partial O$ . Each of these contours is a 4-connected contour composed of edges and points in the 2D plane it spans. Bels are then contour edges and links are contour points. For each contour  $C_j(b)$ , we define a discrete 2D tangent  $(\beta_j(b), \alpha_j(b))$  at  $b$  using a discrete line segment recognition algorithm.

#### 3.1 Recognition of a 4-connected discrete line segment

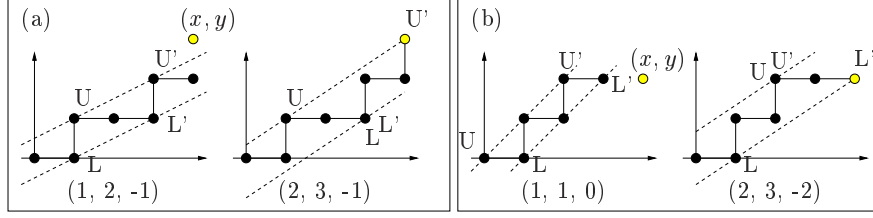
An incremental algorithm was proposed for recognizing 8-connected line segments by Debled-Rennesson and Reveilles [4]. It was adapted for the recognition of 8 or 4-connected discrete tangent lines by Vialard [2] and by Tellier and Debled-Rennesson [13]. We recall here the principle of this line recognition algorithm in the case of 4-connected contours.

A *4-connected discrete line* of characteristics  $(a, b, \mu) \in \mathbb{Z}^3$  can be defined as the following set of discrete points [12]:  $\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\}$ . The slope of the line is given by  $\frac{a}{b}$  while  $\mu$  describes its location in the 2D plane. The real lines of equations  $ax - by = \mu$  and  $ax - by = \mu + |a| + |b| - 1$  are called the upper and lower *leaning lines*. A point belonging to the upper (resp. lower) leaning line is called an upper (resp. lower) *leaning point*.

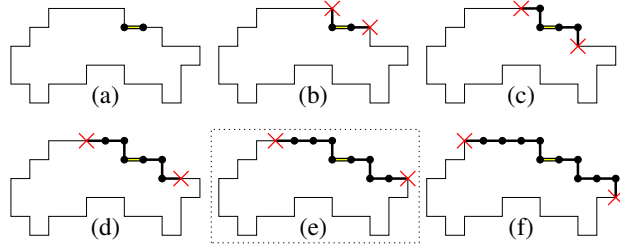
Let us now consider a discrete line segment. We denote by  $U$  (resp.  $U'$ ) the upper leaning point of minimum (resp. maximum) abscissa of this segment. In the same way, we denote by  $L$  (resp.  $L'$ ) the lower leaning point of minimum (resp. maximum) abscissa of this segment.

Given a starting point on the contour, we orient the  $x$ -axis in the direction of the following point. The initial characteristics of the line segment are  $(0, 1, 0)$  and  $U = L = (0, 0)$  and  $U' = L' = (1, 0)$ . Now assume that the characteristics of the line segment are  $(a, b, \mu)$  after adding  $m$  successive contour points. When adding the next contour point  $(x, y)$ , we update the characteristics of the line according to the value  $r = ax - by$  with the rules defined in the following table. In the three first cases the point  $(x, y)$  extends the segment without changing its characteristics  $(a, b, \mu)$ . The new point may just become a leaning point of maximum abscissa. In cases (4) and (5) the segment plus the point  $(x, y)$  is still a line segment. In case (4) (resp. (5)) the slope of the extended line segment is greater (resp. lower) than the slope of the initial line segment. These two last cases are illustrated in Figure 2. Any other value of  $r$  indicates that the current line segment completed by point  $(x, y)$  is no longer a line segment.

$r = ax - by$	U	U'	L	L'	a	b	$\mu$
(1) $\mu < r < \mu +  a  +  b  - 1$							
(2) $r = \mu$		$(x, y)$					
(3) $r = \mu +  a  +  b  - 1$				$(x, y)$			
(4) $r = \mu - 1$		$(x, y)$	$L'$		$y_{U'} - y_U$	$x_{U'} - x_U$	$ax_{U'} - by_{U'}$
(5) $r = \mu +  a  +  b $	$U'$			$(x, y)$	$y_{L'} - y_L$	$x_{L'} - x_L$	$ax_{L'} - by_{L'} -  a  -  b  + 1$



**Fig. 2.** Recognition of a 4-connected line. (a) Slope increase - (b) Slope decrease.



**Fig. 3.** Tangent line computation. (a) Initialization. (b-e) Growth of the tangent line segment. (f) The contour piece is no more a discrete line segment. The tangent line is thus the discrete line segment obtained at the previous step.

### 3.2 Discrete tangent computation

The tangent line segment at a contour edge  $e$  can be defined as the longest line segment corresponding to the contour and centered on  $e$ . This definition is a slight adaptation of the one given in [2]: here the discrete tangent is centered on a contour edge instead of a contour point. Computing the discrete tangent at  $e$  is performed by adding successively pairs of points, one of negative abscissa and one of positive abscissa, to a discrete segment. The preceding line segment recognition algorithm is therefore slightly adapted so that points are added alternately to the front and to the back of the segment. The rules for adding a point to the back are very similar to the ones presented in the previous table. Figure 3 illustrates the tangent computation algorithm.

**Definition 6.** Given a bel  $b$  and a coordinate  $j \neq \perp(b)$ , the 2D tangent vector is defined as  $(\beta_j(b), \alpha_j(b)) = (b, a)$  where  $(a, b, \mu)$  are the characteristics of the tangent line segment computed over the contour  $C_j(b)$ .

## 4 Geometric measures

In this section, we define the normal vector to and the area of a bel from its  $n-1$  2D tangent vectors. We assume that  $(e_0, \dots, e_{n-1})$  is the trivial orthonormal basis of  $\mathbb{R}^n$ .

#### 4.1 Tangent vectors and plane at a bel; normal vector at a bel

The orientations of the tangent vectors in the following definitions come from the definition of boundary operators (see computation of  $\tau$  in Definition 1) and from the fact that contours are implicitly oriented by the sequence of direct links.

**Definition 7.** Let  $b$  be a bel of  $\partial O$ . Let  $i = \perp(b)$  and  $j$  a coordinate different from  $i$ . The  $j$ -th tangent vector  $\mathbf{t}_j(b)$  at  $b$  is the  $n$ -dimensional vector  $(-1)^{n-1-j}\beta_j(b)\mathbf{e}_j + (-1)^{n-i}\alpha_j(b)\mathbf{e}_i$ .

Those  $n - 1$  tangent vectors at  $b$  span an  $n - 1$ -dimensional plane since they are linearly independent. We define the *tangent plane* at  $b$  as the affine plane parallel to these vectors and containing the centroid of  $b$ . It is now easy to define the normal vector at  $b$ .

**Definition 8.** The normal vector  $\mathbf{n}(b)$  at bel  $b$  on  $\partial O$  is the unit vector orthogonal to any vector of the tangent plane at  $b$  and pointing outside the object  $O$ . It is easy to find that  $\mathbf{n}(b) = \frac{\mathbf{u}(b)}{\|\mathbf{u}(b)\|}$  with  $\forall j \neq \perp(b), \mathbf{u}(b) \cdot \mathbf{e}_j = (-1)^{n-j} \frac{\alpha_j(b)}{\beta_j(b)}$ , and for  $i = \perp(b), \mathbf{u}(b) \cdot \mathbf{e}_i = (-1)^{n-i-1}$ .

#### 4.2 Elementary area of a bel; area of a boundary

As the boundary of an object is made of bels, each bel has a given contribution to the area of the whole boundary.

**Definition 9.** The elementary area  $d\sigma(b)$  of a bel  $b$  is defined as  $d\sigma(b) = 1/(\sum_{d=0}^{n-1} |\mathbf{n}(b) \cdot \mathbf{e}_d|)$ . The area of the boundary of  $O$  is then the sum  $\sum_{b \in \partial O} d\sigma(b)$ .

The following theorem justifies the previous definition by examining the elementary area of each bel of a 3D plane.

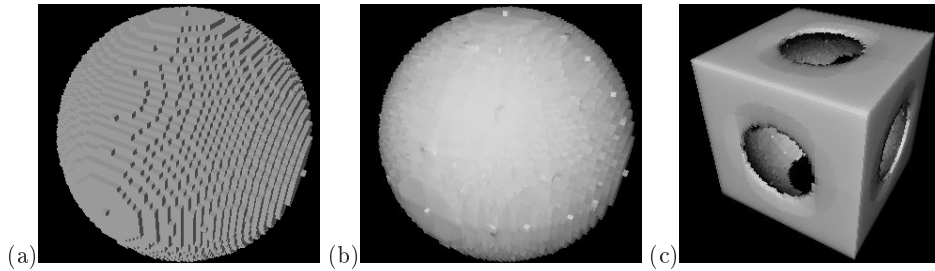
**Theorem 1.** Let  $U = (bc, 0, 0), V = (0, ac, 0), W = (0, 0, ab)$  be three points of  $\mathbb{R}^3$  with  $a, b, c$  positive integer numbers. The continuous plane  $P$  containing the triangle  $UVW$  follows the equation  $ax + by + cz = abc$  and its normal vector  $\mathbf{n}$  is thus  $\frac{1}{\sqrt{a^2+b^2+c^2}}(a, b, c)$ . The digital plane  $Q$ , digitized version of  $P$ , follows the equation  $abc \leq ax + by + cz < abc + a + b + c$ , and forms the vertices of a set of bels in  $\mathbb{C}^3$ . Then the elementary contribution to the area of each bel of  $Q$  is  $1/(\mathbf{n} \cdot \mathbf{e}_0 + \mathbf{n} \cdot \mathbf{e}_1 + \mathbf{n} \cdot \mathbf{e}_2)$ .

*Proof.* Each bel of  $Q$  can be projected onto the plane parallel to it and going through the origin. We restrict  $Q$  to the bels included in the positive octant. The number of bels  $m$  of  $Q$  is therefore obtained by counting the projected bels on each of the three planes of projection. There are  $acab/2$  bels projected on  $x = 0$ ,  $bcab/2$  bels on  $y = 0$  and  $bcac/2$  bels on  $z = 0$  so that  $m = \frac{abc}{2}(a + b + c)$ . Now this subset of  $Q$  corresponds exactly to the triangle  $UVW$ . The elementary area of each bel of  $Q$  is equal to the total area of the triangle  $UVW$  divided by the number of bels of  $Q$  that are part of the digitization of  $UVW$ . The area of  $UVW$  is given by the identity  $UV \wedge UW = abc(a, b, c) = 2\text{area}(UVW)\mathbf{n}$ . A short computation concludes the proof.  $\square$



Object	2D ball		3D ball			4D ball	
	r=50	r=1250	r=20	r=50	r=100	r=10	r=30
nb surfels	404	10004	7542	47070	188502	33352	904648
normal computation time (ms)	0	380	170	1230	6210	1070	29960
mean value of $\phi$	2.24°	0.22°	3.82°	2.19°	1.51°	6.75°	3.98°
std. dev. of $\phi$ from 0	6.47°	1.30°	5.76°	3.46°	2.34°	8.09°	5.15°
area / expected area	1.011	1.000	0.994	0.997	0.998	1.042	1.042

**Table 1.** Comparison between discrete geometric estimators and expected geometric measures. The object under consideration is a ball of increasing radius. For each ball  $b$ , we measure the positive angle  $\phi$  in degree between the expected normal vector to the sphere and the discrete normal  $\mathbf{n}(b)$ . Mean value and standard deviation of  $\phi$  from 0 are listed for 2D, 3D and 4D balls of increasing radii. The estimated area of the discrete balls, their number of surfels, and the computation time of the normal vector fields are also listed (tests made on a Celeron 400Mhz with 128Mbytes of memory).



**Fig. 4.** 3D examples of normal vector computations. Surfaces are rendered with flat shading. (a) Sphere of radius 30 with trivial normals of belts. (b) Same object but with discrete normals. (c) Cube minus sphere with discrete normals.

This theorem can be extended to arbitrary dimension using the  $n$ -dimensional external product, since it also provides an area measurement of  $n - 1$ -dimensional parallelograms. The preceding exposition is sufficient to understand the link between normal vector and elementary area of a belt without too cumbersome notations.

A corollary to this theorem is that if the discrete object of interest is the digitization of a continuous object with good properties (boundary  $C^1$ ), then the area of the discrete object tends toward the area of the continuous object as the discretization resolution increases. Our experiments have confirmed this theoretical result. Table 1 shows that the proposed discrete estimators of normal and area are consistent with expected values. Figure 4 illustrates the computation of normal vector field for two different objects.

## 5 Conclusion

We have defined several geometric measures (tangent plane, normal vector, elementary area, surface area) for boundaries of  $n$ -dimensional objects and we have shown how to compute them efficiently. An immediate extension of this work is the definition and computation of the mean curvature field of  $n - 1$ -dimensional digital surfaces. Our main motivation is the development of a multidimensional discrete deformable model for image segmentation. Local area and curvature measurements are used so as to maintain a regular and smooth shape during the evolution of the model towards boundaries of image components.

## References

1. E. Artzy, G. Frieder, and G.T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Graphics and Image Processing*, 15:1–24, 1981.
2. J.P. Braquelaire and A. Vialard. Euclidean paths: a new representation of boundary of discrete regions. *Graphical Models and Image Processing*, 61:16–43, 1999.
3. D. Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lumière Lyon 2, France, dec 2002.
4. I. Debled-Rennesson and J.P. Reveilles. A linear algorithm for segmentation of discrete curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9:635–662, 1995.
5. T. Y. Kong, R. D. Kopperman, and P. R. Meyer. A topological approach to digital topology. *Am. Math. Monthly*, 98:901–917, 1991.
6. V. A. Kovalevsky. Finite Topology as Applied to Image Analysis. *Computer Vision, Graphics, and Image Processing*, 46(2):141–161, May 1989.
7. J.-O. Lachaud. Coding cells of multidimensional digital spaces to write generic digital topology and geometry algorithms. Research Report 1283-02, LaBRI, University Bordeaux 1, Talence, France, 2002.
8. J.-O. Lachaud. Coding cells of digital spaces: a framework to write generic digital topology algorithms. In *Proc. Int. Work. Combinatorial Image Analysis (IW-CIA'2003), Palermo, Italy*, ENDM. Elsevier, 2003. To appear.
9. A. Lenoir. *Des Outils pour les Surfaces Discrètes. Estimation d'Invariants Géométriques. Préservation de la Topologie. Tracé de Géodésiques. Visualisation*. PhD thesis, Université de Caen, France, sep 1999.
10. A. Lenoir, R. Malgouyres, and M. Revenu. Fast computation of the normal vector field of the surface of a 3D discrete object. In *Proc. of 6th Discrete Geometry for Computer Imagery (DGCI'96), Lyon, France*, volume 1176 of *LNCS*, pages 101–112. Springer-Verlag, 1996.
11. G. Malandain. On topology in multidimensional discrete spaces. Research Report 2098, INRIA, France, 1993.
12. J.P. Reveilles. *Géométrie discrète, Calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Pasteur, Strasbourg, France, 1991.
13. P. Tellier and I. Debled-Rennesson. 3D discrete normal vectors. In *Proc. of 8th Discrete Geometry for Computer Imagery (DGCI'99), Marne-la-Vallée, France*, volume 1568 of *LNCS*, pages 447–457. Springer-Verlag, 1999.
14. J. K. Udupa. Multidimensional Digital Boundaries. *CVGIP: Graphical Models and Image Processing*, 56(4):311–323, July 1994.