



HAL
open science

3D Noisy Discrete Objects: Segmentation and Application to Smoothing

Laurent Provot, Isabelle Debled-Rennesson

► **To cite this version:**

Laurent Provot, Isabelle Debled-Rennesson. 3D Noisy Discrete Objects: Segmentation and Application to Smoothing. *Pattern Recognition*, 2009, 42 (8), pp.1626-1636. 10.1016/j.patcog.2008.11.032 . hal-00439569

HAL Id: hal-00439569

<https://hal.science/hal-00439569>

Submitted on 7 Dec 2009

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.

3D Noisy Discrete Objects: Segmentation and Application to Smoothing[★]

L. Provot^a and I. Debled-Rennesson^a

^a*LORIA Nancy
Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy Cedex, FRANCE*

Abstract

We propose in this paper a segmentation process that can deal with noisy discrete objects. A flexible approach considering arithmetic discrete planes with a variable width is used to avoid the over-segmentation that might happen when classical segmentation algorithms based on regular discrete planes are used to decompose the surface of the object. A method to choose a seed and different segmentation strategies according to the shape of the surface are also proposed, as well as an application to smooth the border of convex noisy discrete objects.

Key words: segmentation, discrete planes, noisy discrete objects

1 Introduction

Three-dimensional discrete objects are widely used in several areas using data coming from acquisition processes such as scanner, magnetic resonance imaging (MRI), ... Due to their internal structure and their huge size, the manipulation of such objects is not an easy task. Rendering algorithms for instance, cannot apply usual techniques to obtain a nice visualization of the objects. A general idea to address this problem is to transform the discrete volume into a Euclidean polyhedron. The segmentation of the border of such objects into discrete primitives such as discrete planes is thus a natural first step and we

[★] This work is supported by the ANR in the framework of the GEODIB project, BLAN06-2_134999

Email addresses: provot@loria.fr (L. Provot), debled@loria.fr (I. Debled-Rennesson).

URL: <http://www.loria.fr/~provot/> (L. Provot).

briefly describe hereafter the main methods proposed within the framework of discrete geometry.

In [9], L. Papier and J. Françon propose a segmentation of the surface of a discrete object into pieces of standard arithmetic discrete planes. These standard planes are recognized using a Fourier-Motskin elimination algorithm [10] and are forced to be homeomorphic to a topological disk in order to be used in a polyhedrization process. They state that the resulting segmentation heavily depends on the choice of the seed to start the recognition of a new face and the tracking order of the points chosen for enlarging the current face, but do not address these problems.

In the framework of surface area estimation [12], R. Klette and H.J. Sun have proposed a segmentation of the surface into digital planar segments (DPS) – which are actually pieces of standard discrete planes. To incrementally check whether a set of points is a DPS, they compute the convex hull of this set to retrieve a specific pair of parallel planes for which the main diagonal distance has to be less than $\sqrt{3}$. A breadth-first search of the surfel graph representing the surface is used to incrementally add points into the DPS.

In [18], I. Sivignon *et al.* have compared different tracking processes to decompose the surface of a discrete object into naive discrete planes. A dual-space approach is used to incrementally recognize the pieces of planes [20]. In [19] the authors have also studied the relation between a segmentation into naive and standard discrete planes depending on the considered surface definition.

Moreover, theoretical works have proved that the problems of polyhedrization and segmentation are computationally hard [17,2,3].

The previously mentioned methods of segmentation are reversible. They generally use naive and standard discrete planes recognition algorithms and behave well with regular discrete objects. However the discrete data coming from acquisition processes are often distorted and these methods might lead to an over-segmentation (see Fig. 6(b)). In this paper we propose a new and original method to segment discrete objects into pieces of discrete planes which is adapted to this possibly distorted data:

- it is ***more flexible*** and addresses the problem by considering a segmentation into pieces of discrete planes with a variable width (parameter fixed by the user) – a flexible recognition algorithm [13] is used,
- it ***considers the shape of the object*** to choose the seeds and to guide the incremental growth of the segments during the segmentation process.

To do this, a first pre-processing step is done to compute geometric features of the surface of the discrete object.

3D noisy discrete objects have been synthesized from regular ones in order to test our method and its adaptability to the level of data distortion. The method is presented in section 2. In section 3, after recalling the definition of blurred pieces of discrete planes, we summarize results from [14] about geometric features for noisy discrete surfaces. The different steps of the segmentation process are then described in section 4, followed by some results on synthesized 3D noisy and real objects. In section 5 an application to smooth the border of convex noisy discrete objects is proposed. The paper ends up with a conclusion and some perspectives in section 6.

This paper is an extended version of a conference paper [15] presented at IWCIA'08.

2 Noisy Discrete Objects

In this paper, we are dealing with three-dimensional noisy discrete objects. The aim is to propose a more flexible segmentation algorithm to avoid the over-segmentation that might happen when regular discrete primitives are used. We would obtain a decomposition into segments which correspond to the natural faces of the objects (for instance six segments, for the six faces of a cube). To test the robustness of our method and its adaptability, we have to observe how it behaves according to the distortion levels of data. We thus have to generate noisy discrete objects from regular counterparts, with different level of distortion. For this purpose, we use a method proposed by T. Kanungo [11] to simulate local distortions that arise after scanning or photocopying processes. Kanungo's noise model accounts for the pixel inversion (due to light intensity fluctuations, pixel sensitivity and thresholding level) and the blur that occurs due to point-spread function of the optical system of scanners. Although this method was primarily designed for 2D images, the different concepts involved can easily be extended to 3D.

The probability of a voxel flipping from background to foreground – or object – (and vice versa) is modeled as an exponential function of its distance from the nearest object voxel (resp. background voxel). Thus, the probability $P(o|b, d, \alpha_0, \alpha)$ to flip a background voxel into an object voxel is $\alpha_0 e^{-\alpha d^2} + \eta$. The parameter α_0 is the initial value for the exponential and the decay speed of the exponential is controlled by the parameter α . The flipping probability $P(b|o, d, \beta_0, \beta) = \beta_0 e^{-\beta d^2} + \eta$ of an object voxel into a background voxel is similarly controlled by β_0 and β . The parameter η represents the constant probability of flipping for all voxels.

The object voxel and background voxel distance d can be computed using a distance transform (*DT*) algorithm [1]. An algorithm that simultaneously

computes the distance transform of object voxels and background voxels is proposed in [7].

To take into account the effect of the point-spread function of the system, a morphological closure operation is applied.

To sum up, the algorithm to synthesize a noisy object from an initial binary object works as follows:

- (1) Compute the distance d of each voxel from the object boundary using DT
- (2) Invert background voxels with probability $P(o|b, d, \alpha_0, \alpha) = \alpha_0 e^{-\alpha d^2} + \eta$
- (3) Invert object voxels with probability $P(b|o, d, \beta_0, \beta) = \beta_0 e^{-\beta d^2} + \eta$
- (4) Perform a morphological closing operation

Fig. 1 shows a noisy discrete object synthesized using different values of α and β . $\alpha_0 = \beta_0 = 1$, $\eta = 0$ and we chose to set $\alpha = \beta$ to generate a symmetrical noise. Fig. 2 and 1(b) allows to see the influence of the DT . And Fig. 3 shows the results obtained using different structuring elements.

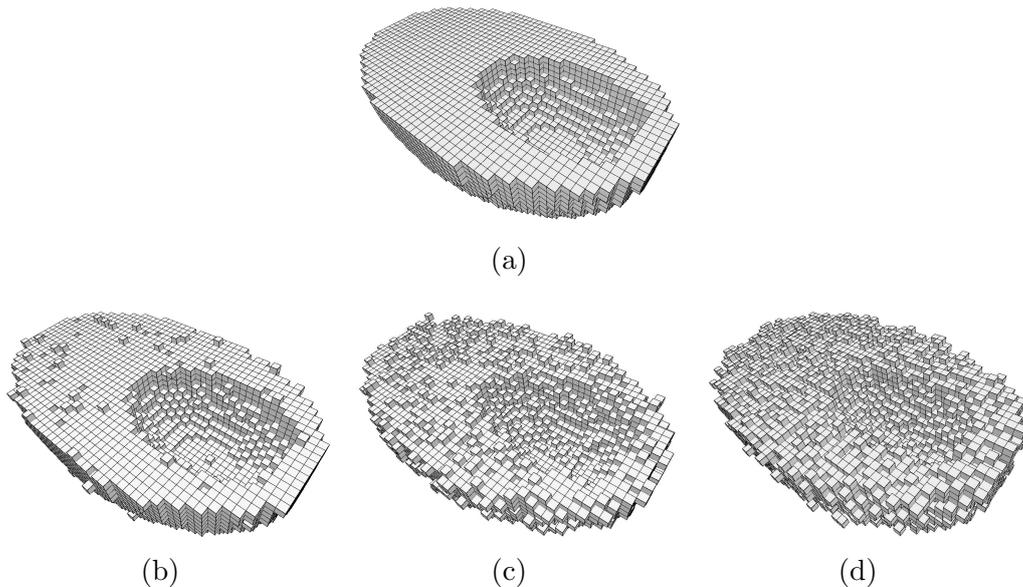


Fig. 1. Noisy half hollowed ellipsoids synthesized from (a) using Kanungo’s model with a $DT = \langle 3, 4, 5 \rangle$ and (b) $\alpha = \beta = 0.4$, (c) $\alpha = \beta = 0.2$ and (d) $\alpha = \beta = 0.1$. The structuring element for the closure is made up of a voxel and its 6-neighborhood.

One can notice that this method may produce non-connected objects (especially with small α, β values or small structuring elements), but we can easily extract a maximal connected component¹ and keep it as the noisy object.

¹ By visiting all the n -connected voxels ($n \in \{6, 18, 26\}$), for instance.

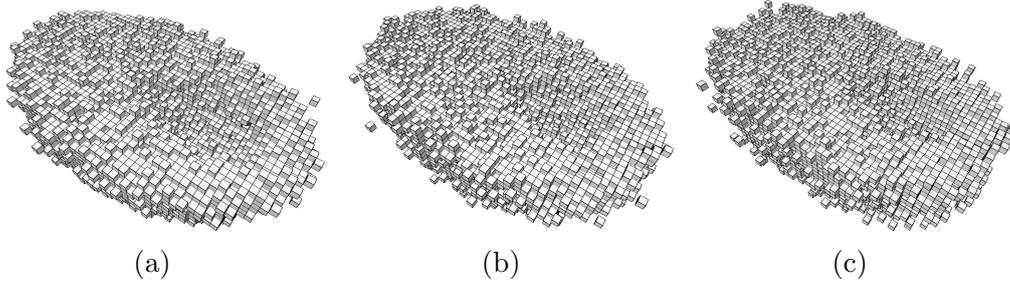


Fig. 2. Noisy half hollowed ellipsoids synthesized from Fig. 1(a) using Kanungo’s model with $\alpha = \beta = 0.4$ and (a) $DT^6 = \langle 1, \infty, \infty \rangle$, (b) $DT^{18} = \langle 1, 1, \infty \rangle$ and (c) $DT^{26} = \langle 1, 1, 1 \rangle$. The structuring element for the closure is made up of a voxel and its 6-neighborhood.

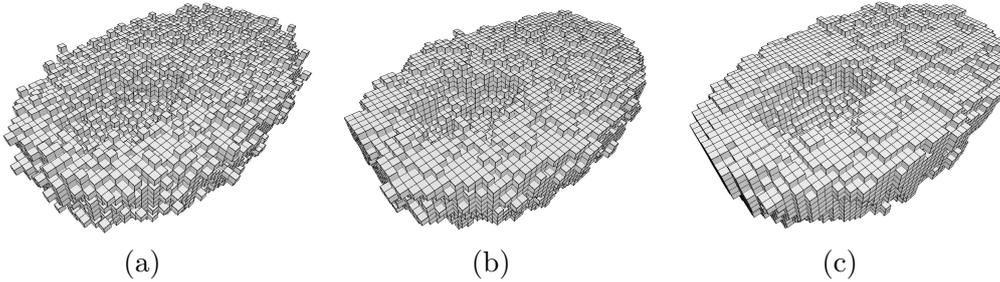


Fig. 3. Noisy half hollowed ellipsoids synthesized from Fig. 1(a) using Kanungo’s model with a $DT = \langle 3, 4, 5 \rangle$, $\alpha = \beta = 0.05$ and a structuring element for the morphological closure made up of a voxel and (a) its 6-neighborhood, (b) its 18-neighborhood and (c) its 26-neighborhood.

Another way to synthesize noisy discrete objects from regular ones is to randomly move inward or outward – by unit moves – border voxels. Thus, generated objects remain connected. In addition, the modification is very local, which enables to easily check that some properties are preserved. For instance, if the original surface of the object is a two-dimensional combinatorial manifold, we can check that a random move does not invalidate this property by verifying that the 26-neighborhood of the new voxel does not contain a forbidden configuration (see [8] for a list of allowed configurations in two-dimensional combinatorial manifolds). If it is not the case, we do not move the voxel.

In this paper, the noisy objects have been synthesized using Kanungo’s method. But we also present results with scanned versions of real objects (see Section 4.4).

3 Background

We recall in this section the definition of a width- ν blurred piece of discrete plane, an arithmetical discrete primitive introduced in [13], that allows to deal

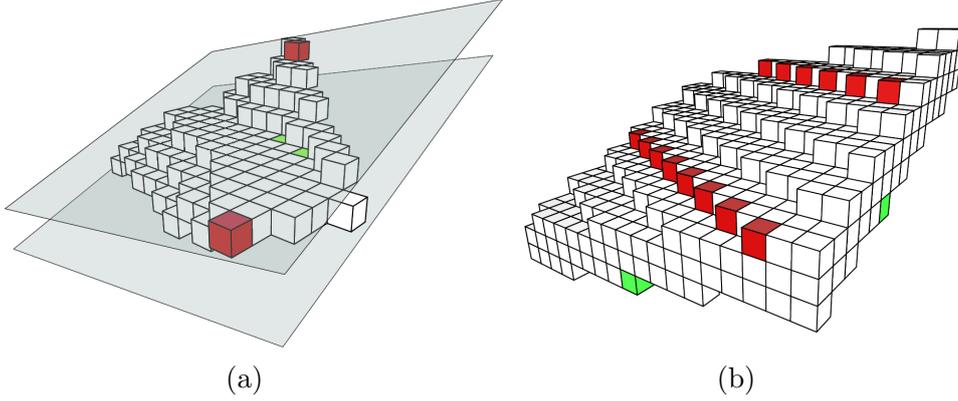


Fig. 4. (a) A width-3 blurred piece of discrete plane and (b) a piece of its optimal bounding plane $\mathcal{P}(4, 8, 19, -80, 49)$, using the Euclidean norm.

with noisy discrete data. Relying on this primitive, we present the notion of a width- ν patch centered at a border point of a discrete object and some features of the border obtained from this patch. More details about the construction of the patch and the study of different features of the border can be found in [14].

3.1 Blurred Pieces of Discrete Planes

One can see a blurred piece of discrete plane as an arithmetic discrete plane for which some points are missing. More formally:

Definition 1 Let N be a norm on \mathbb{R}^3 and \mathcal{E} a set of points in \mathbb{Z}^3 . We say that the discrete plane $\mathcal{P}(a, b, c, \mu, \omega)$ ² is a **bounding plane of \mathcal{E}** if all the points of \mathcal{E} belong to \mathcal{P} , and we call **width of $\mathcal{P}(a, b, c, \mu, \omega)$** , the value $\frac{\omega-1}{N(a,b,c)}$.

A bounding plane of \mathcal{E} is said **optimal** if its width is minimal.

Definition 2 A set \mathcal{E} of points in \mathbb{Z}^3 is a **width- ν blurred piece of discrete plane** if and only if the width of its optimal bounding plane is less than or equal to ν .

Two recognition algorithms of blurred pieces of discrete planes have been proposed in [13]. The first one considers the Euclidean norm and, for a set of points P in \mathbb{Z}^3 , it solves the recognition problem by using the geometry of the convex hull of P . The second one considers the infinity norm and uses methods from linear programming to solve the recognition problem.

² An arithmetic discrete plane $\mathcal{P}(a, b, c, \mu, \omega)$ is the set of integer points (x, y, z) verifying $\mu \leq ax + by + cz < \mu + \omega$, where $(a, b, c) \in \mathbb{Z}^3$ is the normal vector of the plane. $\mu \in \mathbb{Z}$ is named the translation constant and $\omega \in \mathbb{Z}$ the arithmetical thickness.

Thereafter, we denote by \mathcal{O}_b a possibly noisy 6-connected discrete object. We call *surface* or *border* of \mathcal{O}_b the set of points \mathcal{B}_b which have a 6-neighbor that does not belong to \mathcal{O}_b . All the results we present on this type of objects have been obtained by considering the geometrical approach which uses the Euclidean norm.

3.2 Width- ν Discrete Patches

If we are working on a noisy discrete surface and need to extract some of its local geometric features, such as the normal vector or the curvature, it is wise to use estimators that take into account the irregularity of this surface to compute these kinds of features. A way to achieve this goal at a point p of the surface is to gather the information of points lying in an extended neighborhood of p . The notion of patch we present hereafter takes place in this framework, considering an adaptive neighborhood around p .

Definition 3 Let \mathcal{B}_b be the border of a discrete object, p a point in \mathcal{B}_b and ν the greatest real value allowed. Let d be a distance. At each point $q \in \mathcal{B}_b$ we associate the weighting factor $d_p(q) = d(p, q)$. We call **width- ν patch centered at p** , and denote by $\Gamma_\nu(p)$, a width- ν blurred piece of discrete plane incrementally recognized from p by adding points q of \mathcal{B}_b following the increasing values of $d_p(q)$.

About the Incremental Recognition: We construct a *width- ν patch centered at p* using the incremental recognition algorithm of blurred pieces of discrete planes introduced in [13]. We add the points following the increasing values of d_p and, as soon as the width of the blurred piece of discrete plane becomes greater than ν , we stop the recognition process.

About the Distance d : To uniformly spread the patch in all directions, the best solution would be to use a geodesic distance³. Nevertheless, for efficiency, we have chosen to rely on a distance based on a chamfer mask $\langle 3, 4, 5 \rangle$ which is a good approximation of the geodesic distance [1]. The aim is to have a well-balanced patch around p which looks almost circular. With this method we obtain patches like those in Fig. 5.

³ The geodesic distance between two voxels v and w of the border \mathcal{B}_b of a discrete object is the length of the shortest path $\{v = p_1, p_2, \dots, p_n = w\}$ from v to w such that $p_i \in \mathcal{B}_b, 1 \leq i \leq n$.

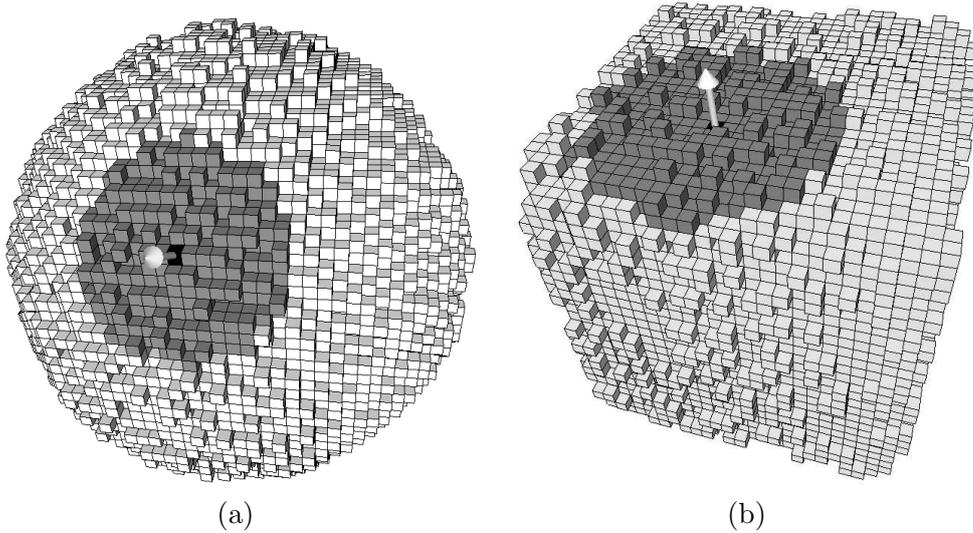


Fig. 5. An example of width-2 patches spread on the surface of different noisy objects. (a) A sphere of radius 20 and (b) a cube of edge 25.

3.3 Patch Features

A patch $\Gamma_\nu(p)$, as previously defined, characterizes the planarity of the surface around p (with respect to the width ν). Thus, the more the patch is spread, the less the surface around p is bent.

In addition, if the growth of $\Gamma_\nu(p)$ stopped, it means that the close neighboring points outside $\Gamma_\nu(p)$ would bend the patch too much if they were added. In that case the patch could no longer be regarded as flat. Therefore, it is possible to deduce a conformation of the discrete surface around p by studying the patches centered along the points of the outline of $\Gamma_\nu(p)$.

The following definitions give a formal quantization of all these observations.

3.3.1 Width- ν Normal

With the previous intuition we can see that the normal vector of $\Gamma_\nu(p)$ is a good estimation of the normal at p . Thus, assimilating the normal vector of $\Gamma_\nu(p)$ to the normal of the surface at p , we define a normal vector estimator for each point of the surface of a possibly noisy discrete object.

Definition 4 Let \mathcal{B}_b be the border of a discrete object and p a point of \mathcal{B}_b . We call **width- ν normal at p** the normal vector

$$\vec{n}_\nu(p) = \vec{n}(\Gamma_\nu(p))$$

where $\vec{n}(\Gamma_\nu(p))$ is the normal vector of the patch $\Gamma_\nu(p)$.

3.3.2 Width- ν Patch Area

Given a Euclidean surface \mathcal{S} and its normal vector field $\{\vec{n}\}$, in the continuous space we can compute the area of \mathcal{S} with the formula:

$$\mathcal{A}(\mathcal{S}) = \int_{\mathcal{S}} \vec{n}(s) ds$$

The discrete version of this equation given in [5] has been adapted as follows to compute the area of the surface of a width- ν patch:

$$E_{\mathcal{A}}(\Gamma_{\nu}(p)) = \sum_{s \in \mathcal{S}_{\Gamma_{\nu}(p)}} \vec{n}_{\nu}(p) \cdot \vec{n}_{el}(s) = \vec{n}_{\nu}(p) \cdot \sum_{s \in \mathcal{S}_{\Gamma_{\nu}(p)}} \vec{n}_{el}(s)$$

where $\mathcal{S}_{\Gamma_{\nu}(p)}$ is the set of surfels⁴ of the patch surface, and $\vec{n}_{el}(s)$ the elementary normal vector of s .

3.3.3 Shape Estimator

An estimator that enables the characterization of the shape (hollow-shaped, knob-shaped or flat) of the surface around a border point of a possibly noisy discrete object has been developed. It is based on the study of the conformation of the patches which are centered on points belonging to the outline of $\Gamma_{\nu}(p)$.

Definition 5 (Patch Outline) *Let \mathcal{B}_b be the border of a discrete object \mathcal{O}_b . We denote by \mathcal{S}_b the set of surfels of \mathcal{B}_b which are incident to a point that does not belong to \mathcal{O}_b , and $\mathcal{S}_{\Gamma_{\nu}(p)}$ the subset of \mathcal{S}_b that belongs to $\Gamma_{\nu}(p)$. A point q belongs to the outline of $\Gamma_{\nu}(p)$ if the voxel representation of q has a surfel $s \in \mathcal{S}_{\Gamma_{\nu}(p)}$ and if there exists a surfel $s' \in \mathcal{S}_b \setminus \mathcal{S}_{\Gamma_{\nu}(p)}$ such that s and s' are adjacent by edge.*

Let C be the set of points that belong to the outline of $\Gamma_{\nu}(p)$. Our shape estimator of the surface around a point p is then given by the formula:

$$\mathcal{F}_{\nu}(p) = \frac{1}{|C|} \sum_{\forall q \in C} (\widehat{\vec{n}_{\nu}(p), \vec{n}_{\nu}(q)}) \cdot \frac{E_{\mathcal{A}}(\Gamma_{\nu}(q))}{E_{\mathcal{A}}(\Gamma_{\nu}(p))}$$

where $(\widehat{\vec{n}_{\nu}(p), \vec{n}_{\nu}(q)})$ is the oriented angle value between the two normal vectors. So, the estimator $\mathcal{F}_{\nu}(p)$ is a weighted mean of the angle values between $\vec{n}_{\nu}(p)$ and the $\vec{n}_{\nu}(q_i)_{1 \leq i \leq |C|}$.

$\mathcal{F}_{\nu}(p)$ is positive when the surface around p is rather knob-shaped and $\mathcal{F}_{\nu}(p)$ is negative when the surface around p is rather hollow-shaped. An increasing

⁴ Faces of a voxel are called *surfels*

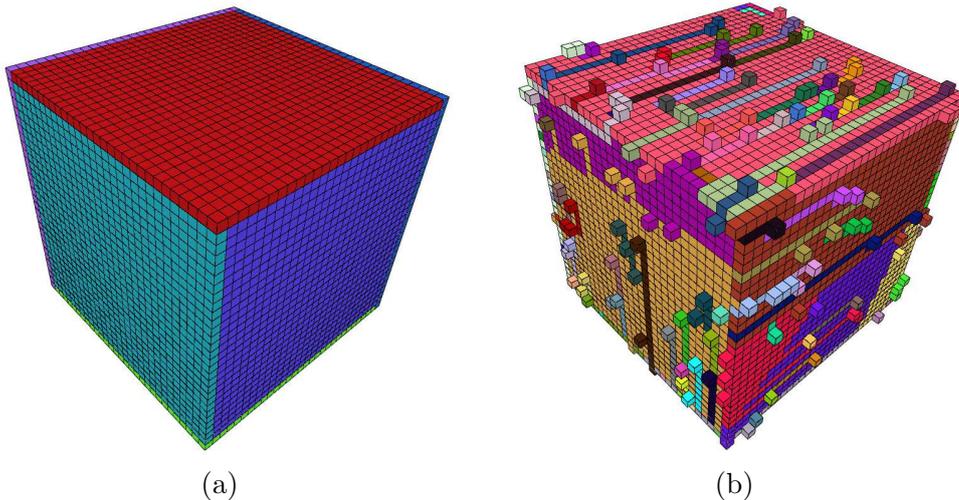


Fig. 6. Segmentation of (a) a regular cube of edge 25 and (b) a noisy counterpart by using the DSD algorithm (<http://liris.cnrs.fr/isabelle.sivignon/DSD.html>).

value of $|\mathcal{F}_\nu(p)|$ means that the surface around p is more strongly bent. Moreover, if $\Gamma_\nu(p)$ is big, a value $\mathcal{F}_\nu(p)$ close to zero means that the area around p is almost flat (according to the width ν we chose). If $\Gamma_\nu(p)$ is small, then the area around p is strongly distorted, but in a way we can neither qualify as hollow- or knob-shaped.

4 Segmentation

4.1 Introduction

The segmentation of a three-dimensional discrete object we will describe in this section consists in partitioning the border of the object into pieces of discrete planes. Some studies have been led on the subject [12,18,19,9] but they all consider regular planes with a fixed width (mainly naive or standard arithmetic discrete planes). Although these methods give good results with regular discrete objects (Fig. 6(a)), it is not always the case when we have to deal with irregular or noisy discrete objects (Fig. 6(b)). In particular, irregularities force to create lots of small segments. The approach we present hereafter is more flexible and considers a segmentation into pieces of planes with a variable width, width- ν blurred pieces of discrete planes to be specific (denoted $BPDP_\nu$ in the sequel), to deal with noisy data.

4.2 Segmentation into Blurred Pieces of Discrete Planes

Firstly, a pre-processing step is done on the border \mathcal{B}_b of the discrete object we want to segment. Given a real ν , for each point $p \in \mathcal{B}_b$ we compute a width- ν patch centered at p as explained in section 3.2. At each point p we can thus associate the features presented in section 3.3, that is:

- the normal vector $\vec{n}_\nu(p)$,
- the area factor $E_{\mathcal{A}}(\Gamma_\nu(p))$,
- and the shape factor $\mathcal{F}_\nu(p)$.

Our segmentation process can be summed up to the following steps: a *seed* is chosen among points of \mathcal{B}_b to start a first $BPDP_\nu$ *recognition* that grows through a process of accretion. An adjacent point is selected and added to the $BPDP_\nu$ if it satisfies some *required criteria*. The $BPDP_\nu$ eventually stops growing when there are no more adjacent points that can be added without contradicting the criteria. This procedure is repeated from a new seed until all points of \mathcal{B}_b belong to a $BPDP_\nu$.

In the following paragraphs we will discuss more in detail the different key points of this segmentation algorithm.

Seed Selection: The easiest way to choose a seed is to randomly pick a border point which does not belong to a $BPDP_\nu$ and to start the recognition process from there. The problem with this approach is that we have no control over the segmentation. To segment a cube for instance, a bad choice would be to start from seeds that lie near an edge of the cube. This would result in an over-segmentation as shown in Fig 7.

A better choice is to start from seeds that are lying in flat areas. It is indeed more meaningful to give a higher priority to flat areas than to bent areas since the underlying primitive of a $BPDP_\nu$ is an arithmetical discrete plane. Chances to have a better approximation are thus higher. Therefore we have chosen to rely on the area estimator $E_{\mathcal{A}}(\Gamma_\nu(p))$ to find the seeds. The idea is to pick the border point p (not yet processed) which has the highest $E_{\mathcal{A}}(\Gamma_\nu(p))$ value as the next seed.

$BPDP_\nu$ Recognition: The algorithm used to incrementally recognize width- ν blurred pieces of discrete planes is the geometrical one proposed in [13] by considering the Euclidean norm.

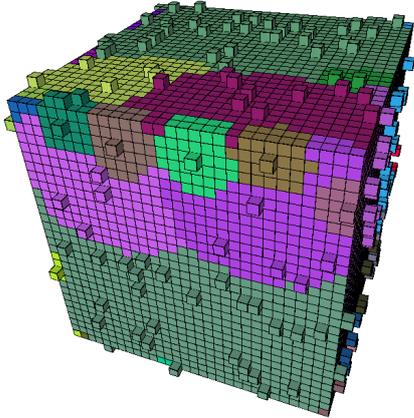


Fig. 7. Over-segmentation due to randomly chosen seeds.

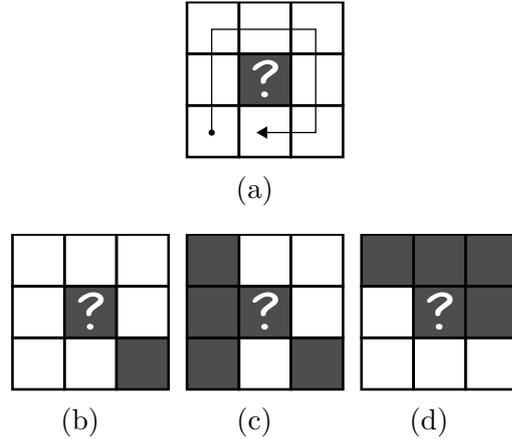


Fig. 8. The points that belong to the $BPDP_\nu$ are in grey (a) Processing order of the neighborhood. (b-d) Some possible configurations when we try to add the point with the question mark: (b) it cannot be added because it is not 4-connected to another grey point; (c) it cannot be added because it creates a hole; and (d) it can be added.

The spreading of a $BPDP_\nu$ heavily depends on the way the neighborhood of the seed is visited as explained in [18]. For the same reasons as before, the value $E_{\mathcal{A}}(\Gamma_\nu(p))$ is used in the accretion process. Points p adjacent to the evolving $BPDP_\nu$ are added according to their decreasing $E_{\mathcal{A}}(\Gamma_\nu(p))$ values.

To implement this behaviour we use a priority queue Q . We start by pushing the seed into the queue with a weight equals to its area factor and mark this seed as visited. Then, while Q is not empty, we pop out of Q the point p with the highest weight w and we add p to the evolving $BPDP_\nu$ if it satisfies the required criteria presented in the following paragraph. We then add the non-visited 26-neighbours of p which belong to the border and their associated area factor into the priority queue Q and mark them as visited.

Using this technique the $BPDP_\nu$ does not stop growing if a point cannot be added.

Required Criteria: The first criterion that has to be satisfied is that the width of the evolving $BPDP_\nu$ must not exceed ν when a point p is added. But this is implicitly checked in the recognition algorithm.

Moreover, as we plan to use the segmentation in a future work to develop a polyhedrization algorithm for noisy discrete objects, the $BPDP_\nu$ segments have to satisfy some constraints of good formation. In particular we want a

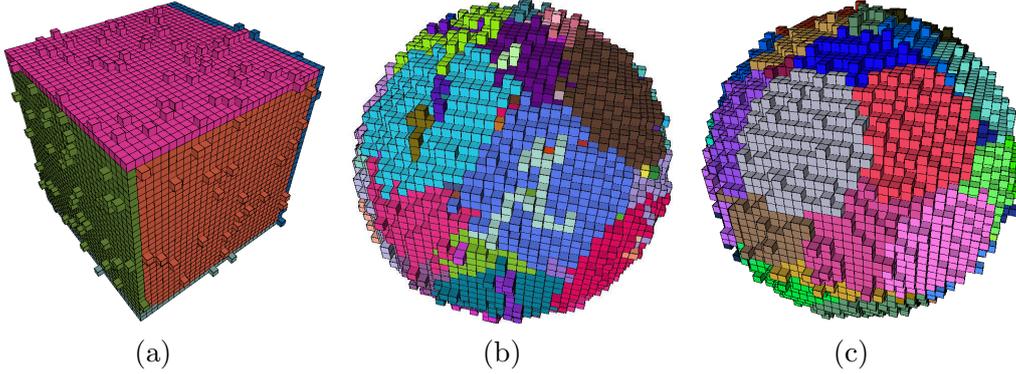


Fig. 9. Width-2 segmentation of a (a) noisy cube of edge 30; (b) noisy sphere of radius 20 with the method presented in section 4.2. (c) Segmentation of the same sphere using only width-2 patches.

$BPDP_\nu$ segment to be 4-connected and without holes, i.e. homeomorphic to a topological disk, according to the main direction of the normal vector of its seed. To check these constraints we use a simplified version of a method proposed in [16] (p.153). We work in the projection plane associated to the normal vector of the seed. We consider the 8-neighborhood of the point we are trying to add in the evolving $BPDP_\nu$ and process the 8-neighbors in the order shown in Fig. 8(a). During the processing a zero-initialized counter is incremented at each time we go from a point which belongs to $BPDP_\nu$ to a point which does not, and vice-versa. At the end, if the counter value is greater than two it means that the point cannot be added without creating a hole. At the same time we check that at least one 4-neighbor belongs to $BPDP_\nu$. If a point does not pass these tests it is marked as non-visited to give the tracking process the opportunity to visit it later on.

Some results obtained with this segmentation method are given in Fig. 9(a) and 9(b). On the one hand, if we look at the cube in Fig. 9(a), we can see that the segmentation is rather good and partitions the object into six segments which correspond to the six faces of the cube. On the other hand, the segmentation of the sphere in Fig. 9(b) could be better. The problem with the sphere is its curved border. The tracking process of points described in section 4.2 has the opportunity to skirt round the points that cannot be added and on curved parts it tends to create rough-crescent-shaped $BPDP_\nu$. If we now use a width- ν patch-based segmentation, as described in section 3.2, we can see that the result is better (see Fig. 9(c)). This is due to the chamfer-mask-based tracking process used to grow the patches.

4.3 Hybrid Method

Relying on the previous observations, we have developed an hybrid segmentation method. For seeds that lie in a flat part of the object we develop a $BPDP_\nu$ segment as described in 4.2 and for seeds that lie in curved parts we develop a patch $\Gamma_\nu(p)$ (see section 3.2).

To distinguish between flat parts and curved parts we use the shape factor $\mathcal{F}_\nu(p)$. As previously explained, an increasing value of $|\mathcal{F}_\nu(p)|$ means that the surface around p is more strongly bent. Thus, given a threshold value σ and a seed s , if $|\mathcal{F}_\nu(s)| < \sigma$ we develop a $BPDP_\nu$ segment, otherwise we develop a patch $\Gamma_\nu(s)$. Results are presented in the next paragraph.

4.4 Results and Comparisons

Results obtained with the hybrid method are shown in Fig. 10 and 11.

In Fig. 10 synthetic objects with different shapes have been segmented at different widths. We can see in Fig. 10(g) and 10(h) that the method still works for non-noisy objects. Furthermore, due to the hybrid approach, both flat and curved areas of the noisy half hollowed ellipsoid in Fig 10(c), 10(f) and 10(i), are well segmented.

To emphasize the improvements brought by the hybrid method in case of distorted data, the number of segments obtained with the method proposed by I. Sivignon⁵, called DSD algorithm, and with the Hybrid Method (HM) are shown in Table 1.

Table 1

Number of segments obtained on synthesized objects.

Noise level parameters $DT = \langle 3, 4, 5 \rangle$		Cube			Half hollowed ellipsoid			Sphere		
		None	Weak $\alpha = 0.28$ $\beta = 0.28$	Strong $\alpha = 0.1$ $\beta = 0.06$	None	Weak $\alpha = 0.4$ $\beta = 0.4$	Strong $\alpha = 0.05$ $\beta = 0.05$	None	Weak $\alpha = 0.2$ $\beta = 0.2$	Strong $\alpha = 0.05$ $\beta = 0.05$
DSD algo		6	223	237	133	199	362	215	662	488
HM	Width $\nu = 1$	6	206	224	93	177	327	115	603	491
	$\nu = 2$	6	6	36	33	47	110	50	115	119
	$\nu = 3$	6	10	15	27	30	41	26	58	50

We can see that the results between the Hybrid Method at $\nu = 1$ and the DSD algorithm are really close. But for increasing width values, the number of segments significantly decreases.

⁵ <http://liris.cnrs.fr/isabelle.sivignon/DSD.html>

We can however notice that on the cube with a weak level of noise, there are more segments for $\nu = 3$ than for $\nu = 2$. It is due to the fact that a segment on one side of the cube may “overflow” on the other sides, as shown in Fig 10(d). In that case, some points on these sides cannot be added because they do not satisfy the required criteria (Section 4.2) and then create tiny segments.

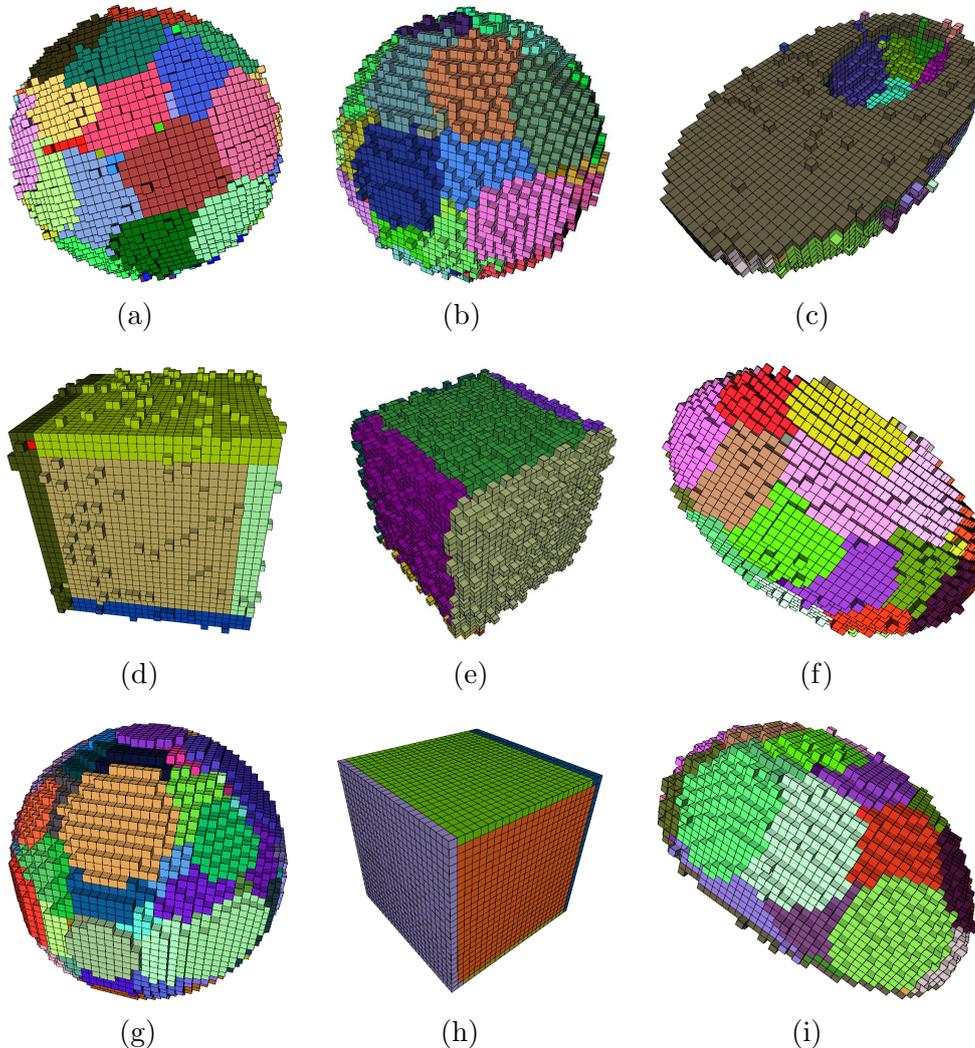


Fig. 10. Results of different segmentations by using the hybrid method on synthetic objects: (a, b) a width-2 segmentation of a weakly noisy sphere of radius 20; (d, e) a width-3 segmentation of noisy cubes of edge 30 and 25; (g, h) a width-1 segmentation of non-noisy objects; and (c, f, i) a width-2 segmentation of an half hollowed ellipsoid.

In Fig. 11 two scanned real-life objects – an old Dodge car and a spaceship, available on the TC18 website⁶ – have been segmented with both DSD algorithm proposed by I. Sivignon and hybrid approach. We can notice that, with the hybrid approach, the flat areas, i.e the roof, the parts of the hood and the windshield of the car are well segmented, with respect to the shape of the car

⁶ http://www.cb.uu.se/~tc18/code_data_set/3D_images.html

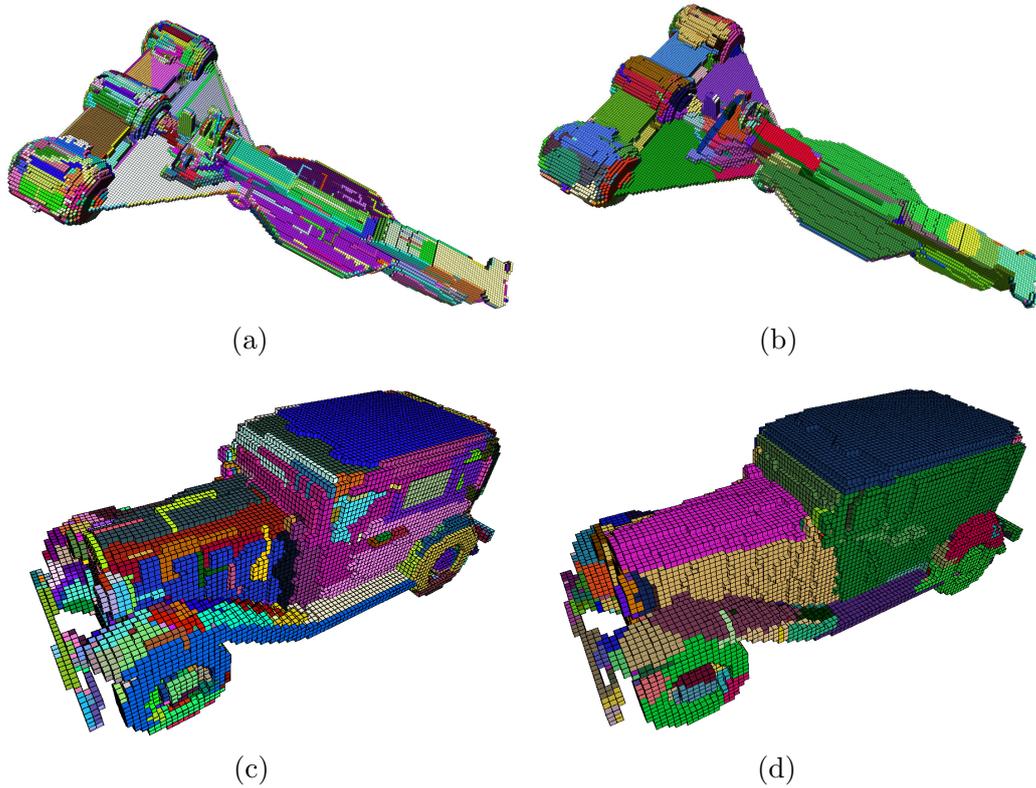


Fig. 11. The segmentation of a car (a, c) by using the DSD algorithm; and (b, d) by using the width-2 hybrid method.

Table 2
Number of segments obtained on real scanned objects.

	Dodge car	Spaceship
DSD algo	565	913
Hybrid Method, $\nu = 2$	259	430

and with a small number of segments. It is also the case for the wings of the spaceship. In curved areas the two segmentations are close, but it is difficult to decide what is a “good” segmentation in these areas.

The number of segments obtained with both Hybrid Method and DSD algorithm are shown in Table 2.

Note that, at this time, the width ν and the threshold have to be set manually, but we would like to investigate more to find a way to automatically choose an appropriate value for these parameters.

5 Application to Smoothing

In this section we present an application of the previous segmentation to smooth the border of noisy discrete objects. The approach we propose only works for convex objects.

Another method [7] has been proposed which relies on distance transform (*DT*). In order to remove protrusions, cavities and components of negligible size, the *DT* of the whole binary 3D image – i.e. both the object and the background – is computed. Then all voxels having a distance less than a chosen threshold are set to zero. This creates a kind of “hollow space” around the border of the object. The *DT* of this hollow space is then computed and voxels having equal distances to object and background are assigned as the new border of the object.

Our smoothing method directly exploits the results obtained by the segmentation algorithm. To smooth a noisy discrete object \mathcal{O}_b , the following steps are performed:

- (1) Compute a segmentation of \mathcal{O}_b for a given width ν . Let L_S be the list of all the segments.
- (2) For each segment $S_i \in L_S$ ($1 \leq i \leq |L_S|$), choose a Euclidean representative plane P_i of S_i
- (3) Let B_i be the set of all integer points that lie in the negative half-space of the representative of P_i . Compute the resulting smoothed object $\mathcal{O}_s = \bigcap_i B_i$

Point (3) states that the smoothed object is the digitization of a convex set. This explains why our method makes sense only when we want to smooth objects for which the result is expected to be convex.

Furthermore different candidates can be chosen to be the Euclidean representative of a segment. Let (a, b, c, μ, ω) be the characteristics of the width- ν blurred piece of discrete plane associated to a segment S of the segmentation. We have chosen as Euclidean representative of S the Euclidean plane $ax + by + cz = \mu + \omega/2$, that is, the plane at middle distance between the two leaning planes of S . This choice seems reasonable for objects synthesized with kanungo’s noise model.

As it stands, this method can still produce unsatisfactory results, as shown in Fig. 12, due to the maximal-extension strategy used to develop the $BPDP_\nu$ ’s. Indeed, if we add points that lie in discontinuous areas of the object (e.g.

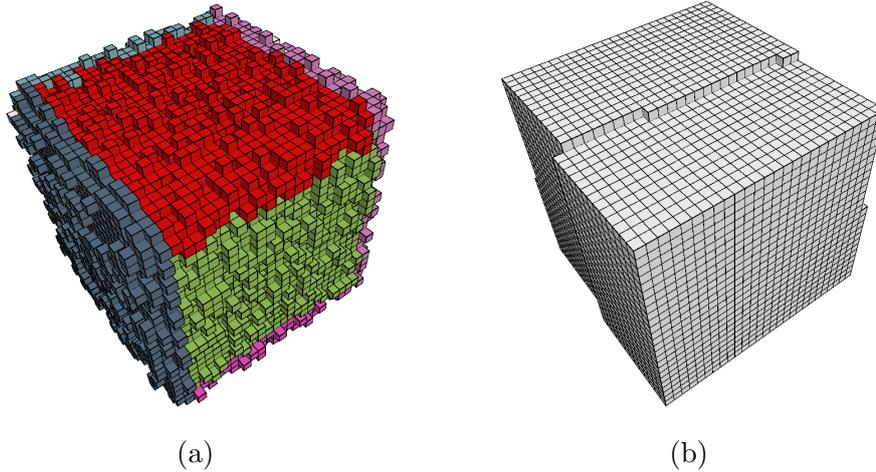


Fig. 12. (a) Example of an undesired segmentation for the smoothing. Points near the edge of the cube have been added into, e.g., the red segment (due to the maximal-extension strategy) which has originated a step in (b) the smoothed cube.

vertices and edges, for a cube) it tends to tilt the segments, which can disturb the smoothing. To avoid this scenario we must ensure that the segments are reliable enough. That is to say, we should not add points that are in strong discontinuous areas of the object. We can detect these points thanks to the shape estimator $\mathcal{F}_\nu(p)$. Since increasing values of $|\mathcal{F}_\nu(p)|$ means that the surface around p is more strongly bent, we do not add points that have a shape factor value greater than a given threshold. Fig. 13(c) shows a gradient colormap – from green (flat areas) to red (knob-shaped areas) – of the shape factor $\mathcal{F}_\nu(p)$ for a noisy dodecahedron and Fig. 13(d) shows the reliable segments obtained when we do not add the points that lie in the discontinuous areas of the object (i.e. the edges, depicted in red in the colormap, for the dodecahedron).

Fig. 13 sums up the different states of a dodecahedron, from (a) the regular original version to (b) a noisy counterpart to (e) its smoothed version, which turns out to be very close to the original one. The digitization of the convex hull of the noisy object is also shown in (f). We can see that the smoothed version obtained with our method is more regular. It is because the border of this object is composed of pieces of digital planes, which correspond to the digitization of the twelve Euclidean representatives of the reliable segments, whereas the convex hull of the noisy object is made up of a greater number of facets due to the noise.

Fig. 14 shows different noisy versions of regular mathematical objects and their smoothed counterparts. For each object the expected number of reliable segments has been found and we have obtained a smoothed version very close to the original one.

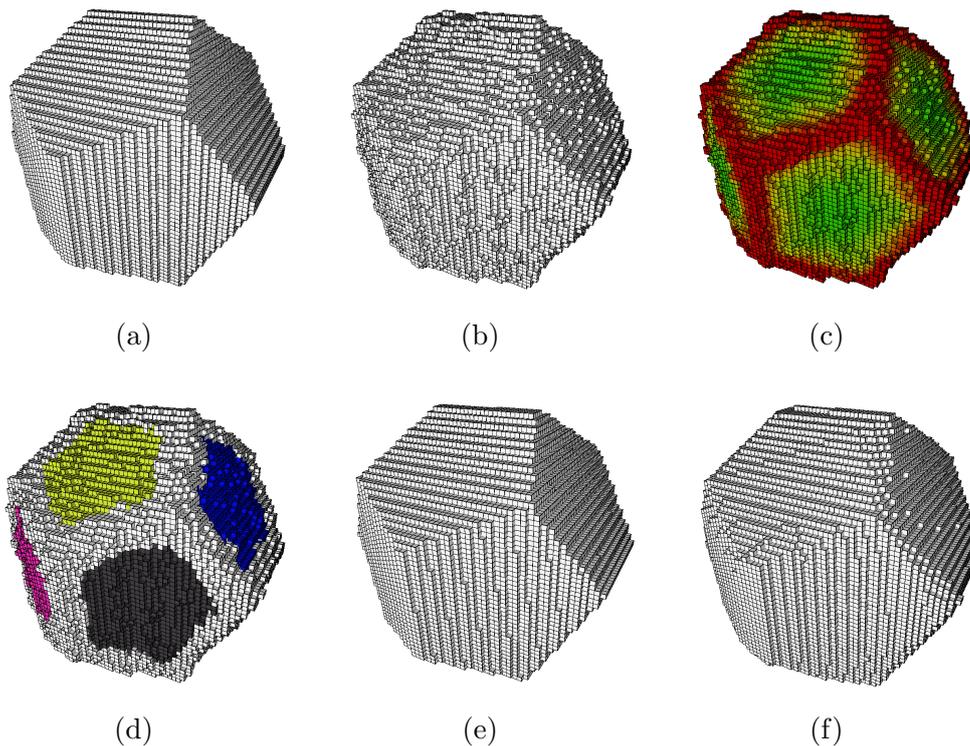


Fig. 13. Different states of a discrete objects. (a) The regular discrete dodecahedron, (b) a noisy counterpart, (c) the colormap of the shape estimator – from flat (green) to knob-shaped (red), (d) the reliable segments, (e) the smoothed object and (f) a digitization of the convex hull of (b).

6 Conclusion

In this paper we have presented a segmentation method to decompose a possibly noisy discrete object into pieces of discrete planes with a variable width. Different segmentation strategies have been proposed, guided by geometric features of the border of the object, computed in a pre-process step. Good results have been obtained for both noisy and non-noisy objects, but we still have to investigate more to automatically choose appropriate values for the different parameters of the method. An application to smooth the border of convex noisy discrete objects has also been proposed.

In a future work, we intend to improve the smoothing algorithm so that it also works for non convex objects. We also want to use the segmentation to develop a polyhedrization algorithm for noisy discrete objects. But some work has to be done to propose a good definition for facets and to study the way to group them together to build a Euclidean polyhedron. The strategies used in [6] and in [4] could help us in that way.

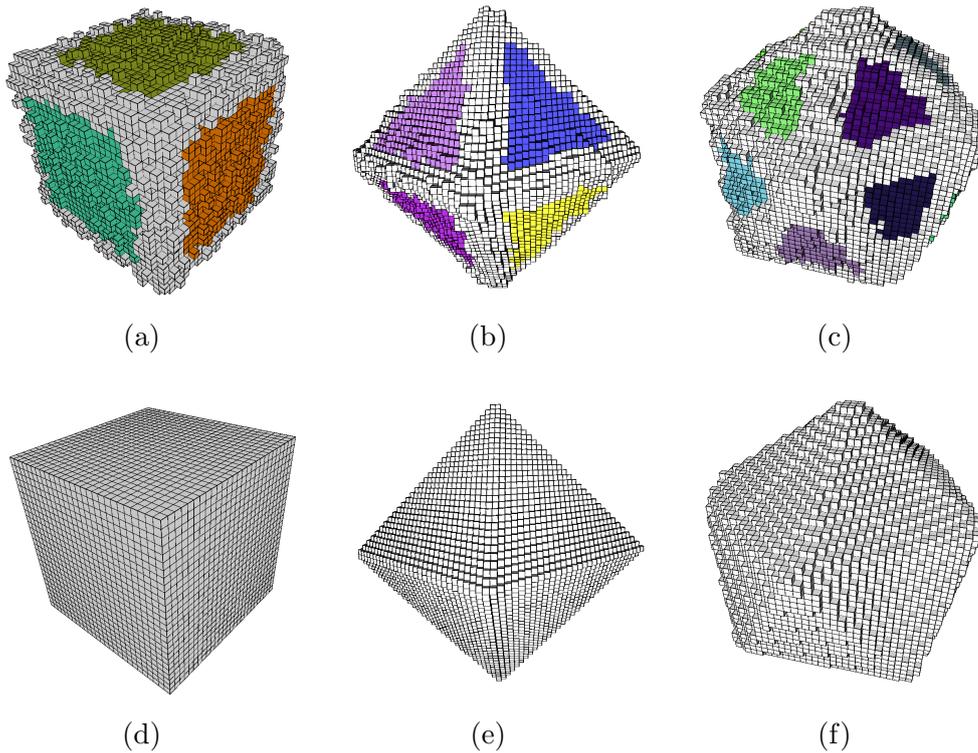


Fig. 14. Different noisy objects with their reliable segments (a) a cube, (b) an octahedron, (c) an icosahedron and (d), (e), (f) their respective smoothed version.

References

- [1] G. Borgefors, On digital distance transforms in three dimensions, *Computer Vision and Image Understanding* 64 (3) (1996) 368–376.
- [2] V. E. Brimkov, Discrete volume polyhedrization: Complexity and bounds on performance, in: *Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications. Proceedings of the International Symposium CompIMAGE 2006, 2007.*
- [3] V. E. Brimkov, Scaling of plane figures that assures faithful digitization, in: *12th International Workshop on Combinatorial Image Analysis*, vol. 4958 of LNCS, Buffalo, NY, USA, 2008.
- [4] D. Coeurjolly, F. Dupont, L. Jospin, I. Sivignon, Optimization schemes for the reversible discrete volume polyhedrization using marching cubes simplification, in: *13th International Conference on Discrete Geometry for Computer Imagery*, vol. 4245 of LNCS, Szeged, Hungary, 2006.
- [5] D. Coeurjolly, F. Flin, O. Teytaud, L. Tougne, Multigrid convergence and surface area estimation, in: *Theoretical Foundations of Computer Vision “Geometry, Morphology, and Computational Imaging”*, vol. 2616 of LNCS, 2003.

- [6] M. Dexet, D. Coeurjolly, E. Andres, Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction, in: 2nd International Symposium on Visual Computing, vol. 4292 of LNCS, Lake Tahoe, Nevada, USA, 2006.
- [7] G. S. di Baja, S. Svensson, Editing 3d binary images using distance transforms, International Conference on Pattern Recognition 2 (2000) 6030.
- [8] J. Françon, Discrete combinatorial surfaces, Graphical Models and Image Processing 57 (1) (1995) 20–26.
- [9] J. Françon, L. Papier, Polyhedrization of the boundary of a voxel object, in: 8th International Conference on Discrete Geometry for Computer Imagery, vol. 1568 of LNCS, Marne-la-Vallee, France, 1999.
- [10] J. Françon, J.-M. Schramm, M. Tajine, Recognizing arithmetic straight lines and planes, in: 6th International Workshop on Discrete Geometry for Computer Imagery, vol. 1176 of LNCS, Lyon, France, 1996.
- [11] T. Kanungo, R. M. Haralick, I. Phillips, Nonlinear local and global document degradation models, International Journal of Imaging Systems and Technology 5 (3) (1994) 220–230.
- [12] R. Klette, H. J. Sun, Digital planar segment based polyhedrization for surface area estimation, in: 4th International Workshop on Visual Form, vol. 2059 of LNCS, Capri, Italy, 2001.
- [13] L. Provot, L. Buzer, I. Debled-Rennesson, Recognition of blurred pieces of discrete planes, in: 13th International Conference on Discrete Geometry for Computer Imagery, vol. 4245 of LNCS, Szeged, Hungary, 2006.
- [14] L. Provot, I. Debled-Rennesson, Geometric feature estimators for noisy discrete surfaces, in: 14th IAPR International Conference on Discrete Geometry for Computer Imagery, vol. 4992 of LNCS, Lyon, France, 2008.
- [15] L. Provot, I. Debled-Rennesson, Segmentation of noisy discrete surfaces, in: V. E. Brimkov, R. P. Barneva, H. A. Hauptman (eds.), Combinatorial Image Analysis, vol. 4958 of LNCS, Springer, Berlin, 2008, 160-171.
- [16] I. Sivignon, De la caractérisation des primitives à la reconstruction polyédrique de surfaces en géométrie discrète, Thèse, I.N.P. de Grenoble (2004).
- [17] I. Sivignon, D. Coeurjolly, Minimal decomposition of a digital surface into digital plane segments is np-hard, in: 13th International Conference on Discrete Geometry for Computer Imagery, vol. 4245 of LNCS, Szeged, Hungary, 2006.
- [18] I. Sivignon, F. Dupont, J.-M. Chassery, Decomposition of a three-dimensional discrete object surface into discrete plane pieces, Algorithmica 38 (1) (2004) 25–43.
- [19] I. Sivignon, F. Dupont, J.-M. Chassery, Discrete surface segmentation into discrete planes, in: 10th International Workshop on Combinatorial Image Analysis, vol. 3322 of LNCS, Auckland, New Zealand, 2004.

- [20] J. Vittone, J.-M. Chassery, Recognition of digital naive planes and polyhedrization, in: 9th International Conference on Discrete Geometry for Computer Imagery, vol. 1953 of LNCS, Uppsala, Sweden, 2000.