



HAL
open science

Segmentation of discrete surfaces into plane segments based on a distance map

Loïc Drieu La Rochelle, Rita Zrour, Gaëlle Largeteau-Skapin, Éric Andrès,
Olena Tankyevych, Catherine Cheze Le Rest

► **To cite this version:**

Loïc Drieu La Rochelle, Rita Zrour, Gaëlle Largeteau-Skapin, Éric Andrès, Olena Tankyevych, et al..
Segmentation of discrete surfaces into plane segments based on a distance map. Computer Science
Research Notes, Jun 2024, Prague (Czech Republic), Czech Republic. 10.24132/CSRN.3401.18 .
hal-04782751

HAL Id: hal-04782751

<https://hal.science/hal-04782751v1>

Submitted on 14 Nov 2024

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.

Segmentation of discrete surfaces into plane segments based on a distance map

Loic Drieu La Rochelle¹
loic.drieu.la.rochelle@univ-
poitiers.fr

Rita Zrour¹
rita.zrour@univ-poitiers.fr

Gaëlle
Largeteau-Skapin¹
gaelle.largeteau.skapin@univ-
poitiers.fr

Eric Andres¹
eric.andres@univ-
poitiers.fr

Olena Tankyevych^{2 3}
Olena.Tankyevych@chu-
poitiers.fr

Catherine Cheze Le
Rest^{2 3}
Catherine.Cheze-Le-
Rest@chu-poitiers.fr

¹Université de Poitiers, Univ. Limoges, CNRS, XLIM, Poitiers, France

²Centre Hospitalier Universitaire de Poitiers (CHUP), Poitiers, France.

³LaTIM, INSERM, UMR 1101, Univ Brest, Brest, France.

ABSTRACT

In this paper, we present a method for segmenting 3D discrete objects into discrete plane segments. This segmentation is the first step in obtaining a polyhedrization of a discrete object with the reversibility property. This constraint requires that the discretization result for polyhedrization be exactly the same as the initial set of points. One of our objectives is to reduce the number of planes in our segmentation and achieve more efficient surface analysis algorithms. In 3D space, direction and starting point are common issues. Our method attempts to achieve segmentation by considering surfels one after the other and agglomerating them with their neighbours based on a distance ranking. This method enables the recognition of critical points on the boundary of a plane segment. A medical application is illustrated by the presentation of a tumour segmentation.

Keywords

Polyhedrization - Segmentation - Discrete Geometry - Discrete Plane - Medical Imaging.

1 INTRODUCTION

Discrete surface polyhedrization has received increasing interest in discrete geometry and has been studied in different papers [BB06, CDJS06, LMR20, LRC22, SC03, SDC04]. The reconstruction of a discrete 3D surface is a challenging problem, especially when a reduced number of polygons is expected and even more, when a property of reversibility is required. Reconstruction is reversible when the digitization of the reconstructed surface corresponds to the original discrete surface. The reversibility property ensures that no information is created or lost during the reconstruction process which is an important property in sensitive ap-

plications such as medical imaging. A reconstruction with fewer polygons helps for fluid and interactive visualization in medical imaging where the volume of data can be important. It represents a bottleneck in medical virtual reality: in [PCD21], the authors noted that for Virtual Reality "[...] results suggest that prerequisites such as real-time performance [...] pose the greatest limitations for clinical adoption and need to be addressed".

One of the widely used methods for 3D reconstruction is the Marching Cubes isosurface extraction algorithm [LC87] that proposes reversible solutions, however, it generates an important number of triangles that is proportional to the number of voxels or surface elements (surfels) of the discrete surface. In the discrete geometry community, several algorithms were proposed for discrete surface polyhedrization. These approaches are usually decomposed into two steps: segmentation of a discrete surface into plane segments [BF94, DA09, DCA06, KS01, Pap99, SDC04, VC00], and reconstruction where each of the plane segments is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

replaced by one or more continuous polygons [CDJS06, CGS04, DA09, DCA06]. Segmentation into discrete planes is an important step since it can guarantee the reversibility property and potentially diminish the number of polygons in the resulting reconstruction. This paper is focused on this first step. There is however a major obstacle to this segmentation process: a segmentation in a minimal number of plane segments has been shown to be an NP-hard problem [SC09].

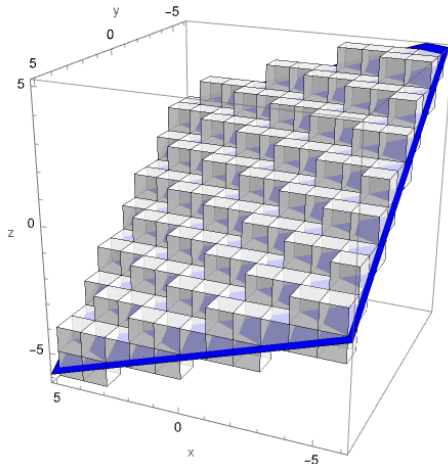


Figure 1: The standard discretization of a Euclidean plane (in blue) of equation $x + 2y + 3z = 1$

In this work, we seek a solution that would yield fewer discrete plane segments in most cases than previously proposed methods [CDJS06, CGS04, DGT]. Our method considers surface elements (surfels) instead of voxels for the reconstruction. This means that we are going to work with *standard* plane segments (i.e. 6-connected planes segments) [And03]. The Figure 1 shows what a Standard plane looks like and how surfels are connected (here face to face). When a segmentation is computed, the starting point and the direction of traversal has an important impact on the results. While, in 2D, on curves, only two directions are possible, in 3D, each surfel has 4 neighbours and selecting a specific direction introduces a bias in the segmentation process. We propose thus labelling on the surfels of our surface, which depends on the planarity of the surfel by determining the biggest disk centred on the surfel that would belong to the same plane. We deduce from that a queue of plane segment centres. Additionally, we adapted the notion of k-cuspal cells proposed by M. Rodriguez et al [RLA08] to propose a geometric criterion for a somewhat subjective question that has not been considered often so far: some reconstruction lead to shapes that *may feel more natural* than others (see Figure 2 for an example).

This paper is organized as follows: section 2 states some definitions and notations. Section 3 presents a state of the problem and of the art on some existing segmentation and reconstruction methods. Section 4

details our segmentation method and section 5 shows some results. We finally end with some conclusions and perspectives in section 6.

2 DEFINITIONS AND NOTATIONS

In this section, we review some notions and definitions of discrete geometry that will be used in the following sections. A definition of a discrete plane, for all points $(x, y, z) \in \mathbb{Z}^3$, has been given by JP. Reveilles in [Rev91] as follow :

$$-\frac{\omega}{2} \leq ax + by + cz + \mu < \frac{\omega}{2}$$

with ω the thickness of the plane, the *intersect* μ and (a, b, c) the normal of the plane. There are classically two types of discrete planes that are considered in discrete surface segmentation problems: *naive* discrete planes are the thinnest 18-connected planes without 6-connected tunnels that can be analytically characterized by a thickness $\omega = \max(|a|, |b|, |c|)$ [AAS97] ; *Standard* discrete planes are the thinnest 6-connected planes without tunnels that can be analytically characterized by a thickness $\omega = |a| + |b| + |c|$ [And03, AAS97]. In the literature (see next section), two discrete surface paradigms have been considered: a *voxel* and a *surfel* paradigm. For the voxel paradigm, the discrete surface of a three-dimensional object is considered to be constituted by the 18-connected voxel outer layer of the object. Those voxels are then segmented into naive planes segments. A discrete plane segment is a finite set of connected points that belong to a discrete plane. In the surfel paradigm, each voxel is considered to be a cubical complex where the 2-dimensional elements are called *surfel*, the 1-dimensional elements are called *linel* and the 0-dimensional elements *pointel*. The surface of an object is composed of the outer layer of surfels of the object (all the surfels that belong to a voxel of the object and to the complementary), and the associated linels and pointels that form the boundary of the surfels. Those surfels are then segmented into standard planes where each pointel of the surfels is seen as a discrete point of the discrete plane segments.

3 STATE OF THE PROBLEM AND STATE OF THE ART

3.1 State of the problem

In dimension two, the segmentation of a curve (open or closed) into discrete line segments has been solved with a solution that is both linear in time and optimal in the number of line segments [FT05]. However, in dimension three, decomposing a discrete surface into a minimal number of discrete planes has been shown to be NP-hard [SC09]. This is not completely surprising since there is no natural order in which discrete points can be added in a seed-based approach (which is the

most common approach as we will discuss next) in dimension three. Sometimes, there is no better solution than a trivial one for an object such as the one presented on the left of Figure 2. All the optimal decompositions into discrete planes will have a number of plane segments that are proportional to the number of voxels/surfels. You cannot do anything here that would be better than a simple straightforward local segmentation. This means that in some cases, globally or locally, one cannot expect a solution that is not proportional to the number of voxels/surfels of the surface. The minimality in the decomposition is not the only relevant question. One aspect of the decomposition of discrete surfaces that has not often been taken into account is that some decomposition *feel* less natural than others. Both the middle and the left image of Figure 2 present an optimal segmentation of a surfel cube into six planes, however, the middle image segmentation may be considered more natural/expected.

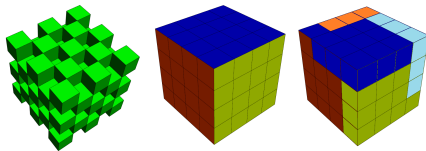


Figure 2: Left: limit configuration. Centre: desired optimal decomposition. Right: undesired optimal decomposition.

3.2 State of the art

Let us now discuss some of the previously proposed approaches to surface segmentation. In his thesis, L. Papier [FP99, Pap99], first proposed a method that creates only segments of naive planes that are rectangular (with a rectangular projection). In a second method, with standard discrete plane segments, he used an algorithm based on Fourier-Motzkin elimination. He imposed geometric and topologic constraints for the plane segments that are commonly taken into account now: connected faces that are homeomorphic to disks. The authors discussed a number of problems with this line of methods such as how to choose the starting point for a face and how to select adjacent surfels to a face and in which order.

In [KS01], R. Klette et al. proposed a segmentation algorithm based on the resolution of a system of inequalities. Surfels are processed by a breadth first traversal of the graph of surfel adjacency on the surface. However, no constraints on the recognized plane segment were imposed which led to thin and awkwardly shaped segments.

In her thesis [Siv04], I. Sivignon proposed several segmentation algorithms. She proposed to use the notion of tricubes (planar sets of 3×3 voxels) to reduce the number of small plane segments that can be recognized.

She also considered plane segments that are topological disks which resulted in a smaller number of plane segments than previous approaches. Lastly, a side effect of using tricubes is to make configurations such as the central image of Figure 2 impossible to reconstruct.

In [CGS04], authors proposed a polyhedrization algorithm that has the property of reversibility with the warranty that the obtained polyhedron is topologically correct; it is based on the simplification of the Marching Cube surface. This algorithm has then been extended in [CDJS06] using linear programming techniques to reduce the number of plane segments.

A last approach [LMR20] uses an arithmetical approach to incrementally compute the normal of a plane segment. Although promising, this approach has not yet resulted in a plane segmentation algorithm.

4 OUR METHOD

In this section, we present our segmentation method to decompose a discrete surface into discrete planes segments.

For the recognition algorithm, we are using an incremental algorithm (COBA) that consists in adding discrete points one after another [CB08]. The recognition problem in \mathbb{Z}^3 is transformed into a feasibility problem in \mathbb{Z}^2 . The function corresponds to the parameter of the two parallel planes enclosing points, and the diagonal distance between these two planes. The time complexity of this algorithm in the worst case is $O(n \log(n))$. It is thus an efficient algorithm for discrete plane recognition. This algorithm does not provide the *preimage* of the set of discrete points but gives the two supporting planes of the segment. In this work, we recognize standard discrete planes using the pointels (in the boundary of surface surfels) of our object.

We work with face-connected voxel objects. From this set of voxels, we compute a surfel-adjacency graph of the surface of our object. The discrete surface that we obtain is closed, thus each surfel has exactly four neighbours.

4.1 Context

Firstly, we need to retrieve the surface of a discrete object composed of voxels to obtain an unoriented graph of surfels. If the surface is closed, each vertex of the graph has exactly four neighbours, and we limit our work to this type of object (i.e. manifold). For the recognition of discrete plane segments, we use the Standard model to have standard surfaces, edges and vertices [And03].

We recognize standard discrete planes considering the four pointels of each surfel of our graph as the discrete points of these planes. A surfel can be added to only one

plane segment, thus the boundaries between our different plane segments are formed of lignels and pointels. The method implemented in our algorithm consists in two passes over our set of surfels. The first is used to compute surfel weights by computing an approximation of the size of the maximal plane it can belong to (see section 4.2). The second pass then computes the segmentation based on the weights while taking care of special cases (see section 4.3).

4.2 First pass

In this subsection, we present the way we compute the weights. The weight of each surfel corresponds to the radius of the biggest Manhattan disk centred on the surfel that corresponds to a standard plane segment. It is computed with a Breadth-First Search order, with a method akin to classical distance map computation algorithms.

A surfel priority queue sorted by decreasing surfel weights is maintained throughout the process.

Figure 3 presents the weight of the surfels on a cube, an octahedron, and a sphere with a colour gradient from green to red. Red surfels represent surfels where it is impossible to create a large plane segment and dark green are those where a larger plane segment can be created. We can see that some *edges* are recognizable in red. Red points are akin to 3D discrete cusp introduced by M. Rodriguez [RLA08]. In 2D, a point is a discrete cusp of a discrete curve iff a line segment of length five centred on this point is not a discrete segment [BSDA03]. M. Rodriguez [RLA08] generalized this notion to dimension three.

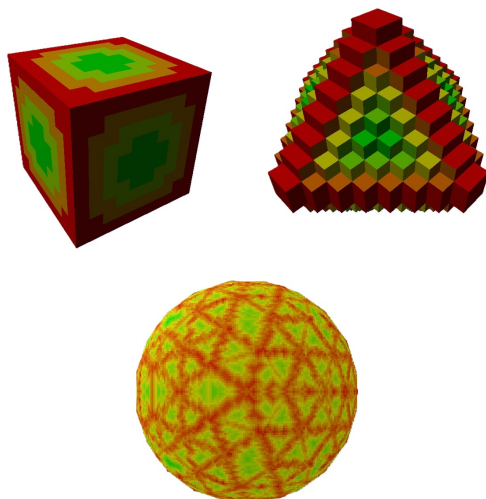


Figure 3: The weighted surfels of our surface on: a cube, an octahedron, a digitized mesh of a sphere

4.3 Second pass

In this second pass, we take the surfels one by one in the surfel priority queue (surfels with the biggest weights

first). The four direct neighbours of the surfel are tested and depending on the configuration (see Figures 4 and 5), different type decisions are taken. These different configurations can be described as follows:

- First configurations: The current surfel has no neighbour already inside a discrete plane segment where it can be added: either because the neighbours have not yet been treated (Figure 4a) or because the current surfel cannot be added to any of the already existing planes (as they wouldn't be planes anymore. See Figure 4b). For these configurations, a new plane segment is created.
- Second configuration: The current surfel has one and only one discrete plane segment already constructed in its neighbourhood where it can be added. The surfel is simply added to this plane (Figure 4c).
- Third configuration: The current surfel has more than one discrete plane segment in its neighbourhood to which it can be added. To select the best plane segment to add the surfel to, we choose to compare the normals of the different plane segments. We calculate a scalar product between the normals before and after adding the surfel to each plane and retain the solution that modifies the normal the least. The union of the point sets of two discrete plane segments can also be a discrete plane segment. Once the segment on which the surfel is to be added has been selected, we then check whether other plane segments in the neighbourhood can be merged with the selected segment. This configuration potentially reduces the number of recognized planes at this step (Figure 4).

Figure 4a and 4b describe two different cases related to the first configuration. In Figure 4a, the yellow surfel is the one with the biggest weight when compared to its four neighbours; it cannot be added to an existing plane since no plane exists yet in his neighbourhood. In Figure 4b, the yellow surfel cannot be added to the already existing red plane since the red plane is no more a standard plane when adding this surfel. In these two cases for configuration 1, a new plane segment is created that can then aggregate other surfels.

Figure 4c shows an example of the second configuration. The yellow surfel has one of its neighbours in the green plane it can be added to. This plane is the only current plane segment in its neighbourhood, it will thus be added to this plane.

Figure 5a shows an example of the third configuration where the yellow surfel can be added to two different plane segments in its neighbourhood, the green plane and the blue plane. In this case, the green and the blue plane segments along with the yellow surfel can

be merged. This is what happens in this case. Figure 5b shows an example of the third configuration where the yellow surfel can be added to two different plane segments in its neighbourhood (green and blue) that cannot be merged into one plane segment. In this case, the yellow surfel will be added to the green plane as it doesn't modify the normal of the green plane, while it would modify the normal of the blue plane.

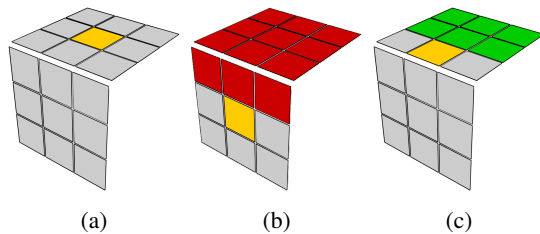


Figure 4: The two first configurations. (a) first configuration: no neighbours have already been treated, (b) first configuration: the current surfel (yellow) cannot be added to the red plane segment, (c) second configuration when a surfel has exactly one neighbouring plane to which it can be added.

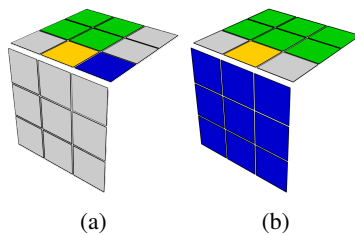


Figure 5: The third configuration when the surfel can be added to at least two plane segments. (a) Merging the planes is possible and will be done. (b) Merging is not possible and the normal is used to decide in which plane segment the surfel is added (here the green one).

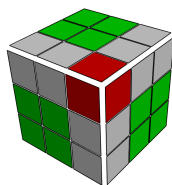


Figure 6: An example of the particular case where a surfel of distance 1 (in red) should not be computed before the grey ones also of distance 1.

There is however one special case to consider: it occurs when a surfel and all its neighbours of a surfel have a weight equal to 1. In Figure 6, the red surfels could form an autonomous plane, although each of those surfels can be added to other planes that may be created later on. The idea here is to wait before such surfels are handled. In this example, the red surfels will be treated after the grey ones, and then end up being added to three different planes. To avoid this case from happening, the

weight of such surfels can be diminished to let neighbour surfel (grey in Figure 5) be computed first.

Algorithm 1: Second pass

input : The *queue* resulting from the first pass
and G the graph of adjacency of surfels
output: a segmentation of the set of surfels into a discrete plane segment

```

segmentation  $\leftarrow$  {}
while queue  $\neq$  {} do
  surfel  $\leftarrow$  queue[0]
  queue  $\leftarrow$  queue \ surfel
  potential  $\leftarrow$  GOOD_NEIGHBORS(surfel)
  if size(potential) = 0 then
    Create a new discrete plane segment  $P$ 
    segmentation  $\leftarrow$  segmentation  $\cup$   $P$ 
     $P \leftarrow P \cup$  surfel
  else
    if size(potential) = 1 then
      potential[0]  $\leftarrow$  potential[0]  $\cup$  surfel
    else
      foreach  $P$  in potential do
         $P\_temp \leftarrow P \cup$  surfel
        pre_n  $\leftarrow$  normal of  $P$ 
        post_n  $\leftarrow$  normal of  $P\_temp$ 
        scalar  $\leftarrow$  pre_n  $\cdot$  post_n
         $B \leftarrow P$  with the minimal scalar
         $B \leftarrow B \cup$  surfel
      foreach  $P_i$  in potential \  $B$  do
        Try to merge  $P_i$  with  $B$ 

```

Algorithm 1 shows the second pass of our algorithm. The *GOOD_NEIGHBORS* function retrieves all the direct plane segments in the neighbourhood of the current surfel.

5 RESULT

This section shows the results of applying our algorithm on different 3D surfaces. The first three synthetic objects denoted *Cube*, *Octahedron* and *Sphere Mesh* are all obtained from the digitization of a mesh; the last object denoted *Parametric Sphere* is a parametric sphere voxelized using Gaussian digitization. These four objects are described as follows:

- *Cube*: a cube (6 face, 8 vertices) of size $10 \times 10 \times 10$ voxels.
- *Octahedron*: an octahedron (8 triangle, 6 vertices) of height 16 voxels.
- *Sphere Mesh*: a low poly sphere (320 triangles, 162 vertices) of radius 64 voxels.
- *Parametric Sphere*: a sphere with a radius of 64 voxels.

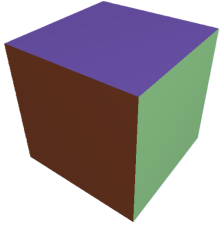
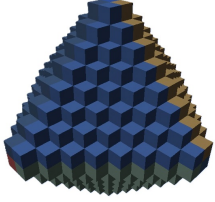
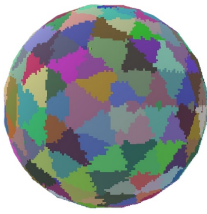
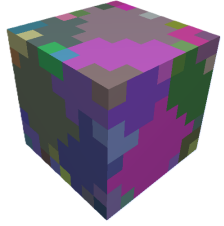
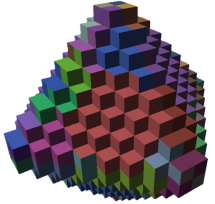
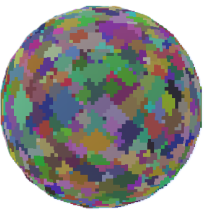
Algorithm	Cube	Octahedron	Sphere Mesh
Our Algorithm			
DGtal Algorithm			

Table 1: The segmentation of the surface of a cube; an octahedron; a digitized sphere mesh using our algorithm and using the DGtal library algorithm.

Table 1 and Figure 7 show the results of applying our algorithm and the algorithm presented in DGtal library, on the *Cube*, *Octahedron*, *Sphere Mesh* and *Parametric Sphere*. Table 2, highlights some details about these results. In this table, we discuss, for each object, the number of faces (triangles or polygons) in the mesh before digitization (Nb Faces) and the number of surfels after digitization (Nb Surfels).

We compare our algorithm to the segmentation method present in DGtal (DGtal is an open-source library where state-of-the-art algorithms in digital geometry are integrated) [DGT]. As this method allows the recognition of thick planes, we limit ourselves to standard planes, in order to maintain a criterion of reversibility. In addition, in order to maintain homogeneity with our method, each surfel is treated as a set of 4 pointels.

The results obtained using this algorithm are listed in Table 2. The results are compared on several metrics: the number of plane segments (Nb PS) generated by each segmentation method, the number of surfels inside the biggest (PS Max Size) and smallest plane segments (PS Min Size), the average (PS Mean Size) and median size (PS Median Size) of a plane segment. We also take the number of plane segments smaller than a defined size, here segments smaller than three and smaller than five in order to see if a method tends to produce small plane segments. These metrics are present in the literature and can be retrieved in several publications [Siv04].

For the *Sphere Mesh*, the number of segments recognized is greater than the number of triangles present in the initial mesh which is what we expected to retrieve. Furthermore, a lot of little segments are present in the segmentation, the presence of these little segments in

the segmentation is a well-known problem and appears at the end of the algorithm when segments are strongly constrained and new surfels cannot fit the actual discrete plane recognition. However, 98,7% of the surfels are in the 320 biggest plane segments. A post-processing stage could reduce those cases, as discussed in the perspectives.

For the *Parametric Sphere* (see Figure7) we have a greater number of plane segments for a roughly similar number of surfels. This result was to be expected since the parametric sphere is not derived from a mesh and has no plane in its geometry. However, results are competitive with previously proposed methods even if some surfels are embedded in plane segments. The percentage of small segments (less than 5 surfels) for the *Parametric Sphere* (28%) is just above the one for the *Mesh Sphere* (24%).

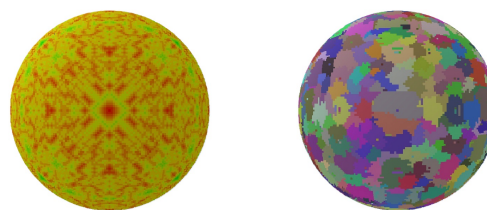


Figure 7: Our algorithm on a parametric sphere (left: first pass, right: second pass).

Overall, our method generates fewer plane segments than the method proposed in the DGtal library, particularly in terms of small pieces of planes.

As noted in section 1, the reconstruction of 3D surfaces can be really useful in medical imaging. Figure 8 shows a head-and-neck tumour semi-automatically segmented from PET image with the algorithm FLAB [HLRT⁺09] and Figure 9 shows our segmentation and the recon-

Object	Cube		Octahedron		Sphere Mesh		Parametric Sphere	
Nb Faces	6		8		320			
Nb Surfels	600		864		77 270		77 118	
Algorithm	Our	DGtal	Our	DGtal	Our	DGtal	Our	DGtal
Nb PS	6	85	8	81	456	2233	1183	2552
PS Max Size	100	41	108	45	551	354	356	159
PS Min Size	100	1	108	1	1	1	1	1
PS Mean Size	100	7	108	10	169	34	65	30
PS Median Size	100	8	108	9	203	31	52	39
Nb surfels ≤ 3	0	24	0	5	83	313	276	370
Nb surfels ≤ 5	0	39	0	29	100	427	335	554

Table 2: Quantitative information obtained after applying our algorithm and the algorithm presented in DGtal.

struction of the tumour. From the clinical point of view, the tumor shape is important in cancer prognosis and therapy choice analysis [HLRA⁺21], therefore it is crucial to have its realistic 3D representation, which provides more precise shape descriptors. Such shape information combined with other image-based characteristics (often called "radiomics"), such as intensity and texture features, can be used to create prediction models for therapeutic choices and for survival prediction via machine and deep learning approaches [TTL⁺22, HLRA⁺21, BTT⁺20].

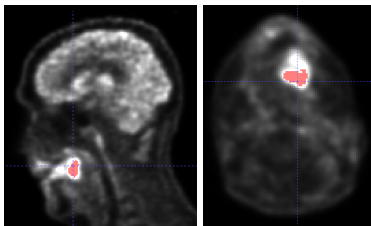


Figure 8: A head-and-neck tumour semi-automatically segmented from PET image with the algorithm FLAB (left : sagittal plane, right : axial plane).

6 CONCLUSION

In this paper, we have presented a new segmentation method for 3D digitized objects. Our method uses critical point detection based on surfel analyses based on a distance map. It retrieves fewer plane segments that are better distributed on the object than the state-of-the-art methods [FP99, KS01, Pap99, Siv04].

Surfel analysis however needs to be improved: some surfels having very different neighborhoods are labelled in the same way which is not accurate. We want to further analyse the topological configuration of surfel neighbourhoods and adjust the weight computation accordingly to better discriminate between cases.

Moreover, we have to study the criteria used to decide, when a surfel can be added to more than one plane and which one is the "best". We may, for those particular surfels, use a post-treatment to swap them from one

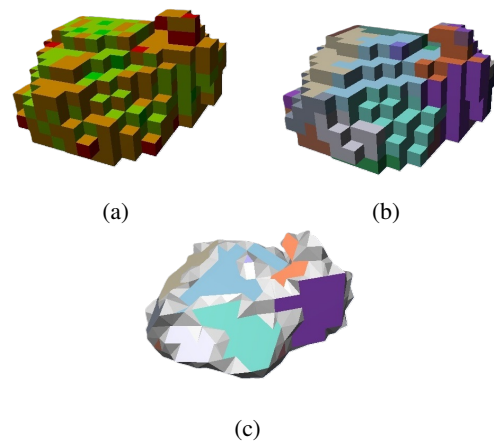


Figure 9: Polyhedrization and our algorithm surface decomposition on a head-and-neck cancer tumour. (a) The first pass, (b) the segmentation on discrete plane segment, (c) a reconstruction using Marching cube optimisation algorithm [CGS04] on our segmentation (b).

plane segment to another to optimize the reconstruction. A way to reach our goal may be to define a Machine learning model to optimize our results.

Other post-treatments can be considered: cutting the large plane segments to add smaller segments to homogenize the size of the polygons for example.

7 REFERENCES

- [AAS97] E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, 59(5):302–309, 1997.
- [And03] E. Andres. Discrete linear objects in dimension n: the standard model. *Graphical Models*, 65(1):92 – 111, 2003.
- [BB06] Valentin E. Brimkov and Reneta Barneva. Polyhedrization of discrete convex volumes. In *Advances in Visual Computing*, pages 548–557, 2006.

- [BF94] P. Borianne and J. Françon. Reversible polyhedrization of discrete volumes. In *4th Discrete Geometry for Computer Imagery*, pages 157–168, Grenoble, France, 1994.
- [BSDA03] Rodolphe Breton, Isabelle Sivignon, Florent Dupont, and Eric Andres. Towards an invertible euclidean reconstruction of a discrete object. In Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors, *Discrete Geometry for Computer Imagery*, pages 246–256, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [BTT⁺20] Moran Berraho, Olena Tankyevych, Gaele Tachon, HATT Mathieu, Dimitris Visvikis, and Catherine Cheze Le Rest. Use of pet derived features to predict mutational status in lung adenocarcinomas, 2020.
- [CB08] Emilie Charrier and Lilian Buzer. An Efficient and Quasi Linear Worst-Case Time Algorithm for Digital Plane Recognition. In *Discrete Geometry for Computer Imagery*, pages 346–357, 2008.
- [CDJS06] David Coeurjolly, Florent Dupont, Laurent Jospin, and Isabelle Sivignon. Optimization Schemes for the Reversible Discrete Volume Polyhedrization Using Marching Cubes Simplification. In *Discrete Geometry for Computer Imagery*, pages 413–424, 2006.
- [CGS04] David Coeurjolly, Alexis Guillaume, and Isabelle Sivignon. Reversible discrete volume polyhedrization using Marching Cubes simplification. In *Vision Geometry XII*, pages 1 – 11. International Society for Optics and Photonics, SPIE, 2004.
- [DA09] Martine Dexet and Eric Andres. A generalized preimage for the digital analytical hyperplane recognition. *Discret. Appl. Math.*, pages 476–489, 2009.
- [DCA06] Martine Dexet, David Coeurjolly, and Eric Andres. Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction. In *Advances in Visual Computing, Second International Symposium*, pages 514–523, 2006.
- [DGT] DGTAL. Digital geometry tools and algorithms library. <http://libdgtal.org>.
- [FP99] Jean Françon and Laurent Papier. Polyhedrization of the Boundary of a Voxel Object. In *Discrete Geometry for Computer Imagery*, pages 425–434, Berlin, Heidelberg, 1999.
- [FT05] F. Feschet and L. Tougne. On the min DSS problem of closed discrete curves. *Discrete Applied Mathematics*, pages 138–153, 2005.
- [HLRA⁺21] Mathieu Hatt, Catherine Cheze Le Rest, Nils Antonorsi, Florent Tixier, Olena Tankyevych, Vincent Jaouen, Francois Lucia, Vincent Bourbonne, Ulrike Schick, Bogdan Badic, et al. Radiomics in pet/ct: current status and future ai-based evolutions. *Seminars in Nuclear Medicine*, 51(2):126–133, 2021.
- [HLRT⁺09] Mathieu Hatt, Catherine Cheze Le Rest, Alexandre Turzo, Christian Roux, and Dimitris Visvikis. A fuzzy locally adaptive bayesian segmentation approach for volume determination in pet. *IEEE transactions on medical imaging*, 28(6):881–893, 2009.
- [KS01] Reinhard Klette and Hao Jie Sun. Digital Planar Segment Based Polyhedrization for Surface Area Estimation. In *Visual Form 2001*, pages 356–366, Berlin, Heidelberg, 2001.
- [LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, pages 163–169, 1987.
- [LMR20] Jacques-Olivier Lachaud, Jocelyn Meyron, and Tristan Roussillon. An optimized framework for plane-probing algorithms. *J. Math. Imaging Vis.*, 62(5):718–736, 2020.
- [LRC22] Jui-Ting Lu, Tristan Roussillon, and David Coeurjolly. A new lattice-based plane-probing algorithm. In *Discrete Geometry and Mathematical Morphology*, pages 366–381, 2022.
- [Pap99] Laurent Papier. *Polyédration et visualisation d’objets discrets tridimensionnels*. PhD thesis, Louis Pasteur, Strasbourg, 1999.
- [PCD21] F. Pires, C. Costa, and P. Dias. On the use of virtual reality for medical imaging visualization. *J Digit Imaging*, page 1034–1048, 2021.
- [Rev91] Jean-Pierre Reveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Paster, 1991.

- [RLA08] Marc Rodríguez, Gaëlle Largeteau-Skapin, and Eric Andres. Local non-planarity of three dimensional surfaces for an invertible reconstruction: k-cuspal cells. In *Advances in Visual Computing, 4th International Symposium, ISVC*, Lecture Notes in Computer Science, pages 925–934, 2008.
- [SC03] Isabelle Sivignon and David Coeurjolly. From Digital Plane Segmentation to Polyhedral Representation. In *Geometry, Morphology, and Computational Imaging*, pages 356–367, Berlin, Heidelberg, 2003.
- [SC09] Isabelle Sivignon and David Coeurjolly. Minimum decomposition of a digital surface into digital plane segments is NP-hard. *Discrete Applied Mathematics*, pages 558–570, 2009.
- [SDC04] Isabelle Sivignon, Florent Dupont, and Jean-Marc Chassery. Decomposition of a Three-Dimensional Discrete Object Surface into Discrete Plane Pieces. *Algorithmica*, pages 25–43, 2004.
- [Siv04] Isabelle Sivignon. *De la caractérisation à la reconstruction polyédrique de surfaces en géométrie discrète*. PhD thesis, Grenoble, Institut national polytechnique de Grenoble, 2004.
- [TTL⁺22] Olena Tankyevych, Flora Troussel, Claire Latappy, Moran Berraho, Julien Dutilh, Jean Pierre Tasu, Corinne Lamour, and Catherine Cheze Le Rest. Development of radiomic-based model to predict clinical outcomes in non-small cell lung cancer patients treated with immunotherapy. *Cancers*, 14(23):5931, 2022.
- [VC00] Joëlle Vittone and Jean-Marc Chassery. Recognition of Digital Naive Planes and Polyhedrization. In *Discrete Geometry for Computer Imagery*, pages 296–307, Berlin, Heidelberg, 2000.

