Reversible Polygonalization of a 3D Planar Discrete Curve: Application on Discrete Surfaces

Isabelle Sivignon[1], Florent Dupont[2], and Jean-Marc Chassery[1]

[1] Laboratoire LIS,
Domaine universitaire Grenoble - BP46,
38402 St Martin d'Hères Cedex, France
{sivignon, chassery}@lis.inpg.fr
[2] Laboratoire LIRIS - Université Claude Bernard Lyon 1,
Bâtiment Nautibus - 8, boulevard Niels Bohr,
69622 Villeurbanne cedex, France
florent.dupont@liris.cnrs.fr

Abstract. Reversible polyhedral modelling of discrete objects is an important issue to handle those objects. We propose a new algorithm to compute a polygonal face from a discrete planar face (a set of voxels belonging to a discrete plane). This transformation is reversible, *i.e.* the digitization of this polygon is exactly the discrete face. We show how a set of polygons modelling exactly a discrete surface can be computed thanks to this algorithm.

1 Introduction

Since a few years now, many methods have been proposed in order to compute a polyhedral representation of a discrete object, or more precisely of its surface. Indeed, such a transformation has many useful properties such as:

- a compression of the discrete data: a polyhedral representation suppresses the redundancy of the discrete representation;
- modelling of the discrete object
- better visualization, ...

Two kinds of reconstruction exist: either we only need an approximation of the discrete surface, or the reversibility of the transformation is desired.

A first naive method consists in computing the convex hull of the discrete points composing the discrete object. This approach simply needs an efficient convex hull algorithm, which is a very well known problem (see [1] for instance). Nevertheless, the polyhedral surface reconstructed is close to the discrete object only in case of convex, or nearly convex objects.

The Marching-Cubes algorithms [2, 3] are the most popular methods to get a polyhedral (triangulated in this case) surface from a discrete object. This transformation is reversible along a classical digitization scheme (OBQ). Nevertheless, this construction is based on local configurations: this induces that the number

E. Andres et al. (Eds.): DGCI 2005, LNCS 3429, pp. 347–358, 2005.

of faces of the polyhedral surface depends directly on the number of surface voxels of the discrete object. In order to compute a surface composed of a number of faces not related to the number of discrete points, a global study of the discrete object geometry is needed.

Then, another class of methods is based on the following outline: first, the discrete object surface is decomposed into pieces of discrete planes, and second, a polyhedral representation of the discrete object based on those discrete faces is computed. Such a framework has been used for instance by Borianne and Françon in [4] where a pair digitization/reconstruction is proposed. The authors conjecture that this pair defines a reversible transformation, but this has not been proven yet. In [5], Françon and Papier also proposed an algorithm based on this scheme. Nevertheless, in this case, they directly transform the discrete faces into polygonal non coplanar faces, which is not a satisfactory modelling of the object.

The last two methods we recall in this short state of the art compute an approximation of the discrete surface. The first one was proposed by Burguet and Malgouyres in [6] and uses a "topological Voronoi Diagram". This diagram is used in order to decompose the discrete surface into regions, which are triangulated to get the polyhedral representation. Finally, Yu and Klette [7] use the minimum length polygon algorithm on each slice of the discrete object and sue those polygons together to obtain an approximation of the discrete object surface.

The algorithm presented in this paper is based on a segmentation of the discrete object surface into pieces of discrete planes. We present an algorithm that computes, for each discrete face, a planar polygon containing the voxels of the discrete face in its digitization. Such a transformation is achieved *via* an analytical modelling of the discrete face, which defines a compact description of the discrete object itself.

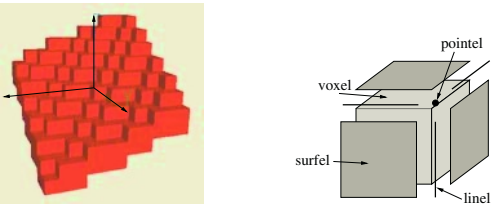
This paper is composed of three sections. In Section 2, we present the general framework of our algorithm, defining the notions of discrete plane, surface, connectivity and segmentation we use together with the dual spaces. The third section deals with the description of our algorithm and finally, application results of this algorithm over each discrete face of discrete surfaces are proposed in Section 4.

2 Framework and Tools

2.1 Preliminaries

First of all, we define the framework used in this paper. Our algorithm takes in input a discrete surface that has already been segmented into pieces of discrete planes.

The definitions of discrete plane and discrete surface used for the segmentation process are induced by the reconstruction we propose. Indeed, our transformation defines a discrete polygon (analytical description) from a discrete face (defined by a set of discrete points). The notion of discrete polygon was introduced by Andrès in [8] using the standard digitization model. This digitization



(a) (b)

Fig. 1. (a) Example of a discrete standard plane. (b) Decomposition of a voxel into lower dimensional elements

scheme is based on the supercover digitization which states that any pixel (or voxel) crossed by the object belongs to the digitized object. Supercovers may contain "bubbles", *i.e.* many digitization pixels for one point (points with half-integer coordinates for instance). To cope with that problem, an orientation convention is defined and leads to the standard digitization scheme.

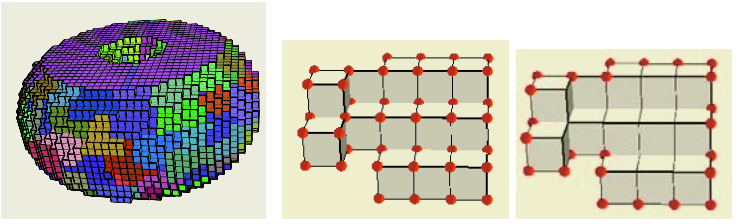
Consequently, standard planes must be used for the segmentation process. A standard plane is the thinnest 6-connected discrete plane without tunnels: any path (6-connected, 18-connected or 26-connected) joining the two background sides of a plane contains at least one voxel of the plane. An illustration of a standard plane is given in Figure 1(a).

Concerning the definition of discrete surfaces, two main approaches exist: the surface elements are either object voxels or object voxels' faces. In this work, we will define the object surface as the set of voxels' faces (called *surfels*, see Figure 1(b)) belonging to an object and a background voxel. In other words, the surface is composed of the faces visible when the object is displayed. This definition of surface is well-adapted for standard planes segmentation: in this case, discrete (lattice) points are not the object voxels but the vertices of those voxels (called *pointels*, see Figure 1(b)). It is easy to see that those vertices are linked along the 6-connectivity, which is consistent with the use of standard planes.

Using standard planes also induces the connectivities we consider for the object. Standard planes have a combinatorial structure of 2-dimensional manifolds [9, 10]. Thus, the discrete surface we work on should have the properties of a $2D$ combinatorial manifold as well, which implies that 6-connectivity has to be considered for the discrete object.

At this point, we have defined all the elements needed for a segmentation process: discrete plane and surface, connectivity. Now let us describe the properties the segmentation must fulfill:

1. pointels adjacent to a common surfel belong to a common discrete face
2. the projection of each discrete face along its main direction (direction given by the maximum parameter of its normal vector) is a set of 4-connected pixels
3. each discrete face is homeomorphic to a topological disk.



(a) (b) (c)

Fig. 2. Example of a segmentation result required for the reconstruction: (a) on a torus, (b) detail of a discrete face and (c) of its boundary

Those three conditions imply that any discrete face is a combinatorial 2-manifold with boundary, and that this boundary can be described as a 6-connected $3D$ discrete curve.

Figure 2(a) gives an example of the result we get with a segmentation algorithm fulfilling those three conditions (see [11]). Note that the top of the torus is decomposed into two discrete faces instead of one, so that each discrete face is homeomorphic to a disk. On Figure 2(b), a discrete face is depicted, and each pointel belonging to this face is marked by a small sphere. The boundary of this discrete face can be described as a 6-connected curve, as illustrated in Figure 2(c).

2.2 Principle

Since each discrete face is composed of discrete points belonging to a standard plane, there exist a Euclidean plane crossing all the discrete face points. The overall algorithm consists in computing for each discrete face f crossed by a plane p, a polygonal line embedded in p and crossing all the boundary points of f. This polygonal line defines a polygon containing exactly the discrete points of f in its standard digitization.

The result of such an algorithm is a set of polygons, one for each discrete face, such that the standard digitization of each polygon is exactly a standard face, and finally, the standard digitization of the set of polygons is exactly the initial discrete object surface.

To compute a polygonal line from a discrete face boundary, we propose the following outline, that we present in [12] in the case of 2D discrete curves and non coplanar 3D discrete curves. Consider a 6-connected discrete curve S described as an ordered set of discrete points $\{v_1, v_2, \dots v_n\}$. A Euclidean point r_1 is chosen inside the first point v_1, and the following voxels are added one by one (they define a discrete segment s_1) while there exist a Euclidean line going through r_1, through all the voxels of s_1 and embedded in the carrier plane p. In other words, s_1 is incrementally extended while:

– s_1 is a 3D discrete segment
– among the lines which contain s_1 in their digitization, there exist at least one line that is embedded in p and that goes through the fixed point r_1.

When one of those two conditions is no more fulfilled, the first real segment endpoint r_2 is computed as a common point of the computed line and s_1's last pixel. The fixed extremity of the next real segment is set to r_2 and this process starts over.

2.3 Dual Spaces and Preimages

Dual Spaces. In order to polygonalize discrete faces boundaries, the question " does this set of voxels belong to a discrete segment ?" will need an answer, and one method to solve this problem is to rewrite it in a dual space. The main idea is that a line in the Euclidean space is represented by a point in the dual space, and conversely, a point in the Euclidean space corresponds to a line in the dual space.

This tranformation is very similar to the Hough Transform [13, 14] that is classically used in image analysis for shape detection problems. The main difference between the Hough transform and the transform we use here is that the uncertainty related to the discrete nature of the data is not handled during a quantification step but during the tranform itself. Many works in discrete geometry use this transform for 2D discrete line or plane recognition [15, 16, 17] but we will not provide a full state of the art on this point here.

An illustration of the mapping we use is given in Figure 3. Note that in this figure, the dual representation of the line defined by the equation $ax - by + r = 0$ is the point $(\frac{a}{b}, \frac{r}{b})$, and thus, that this representation is based on a normalization along one direction (direction y in this case). Consequently, two dual spaces can

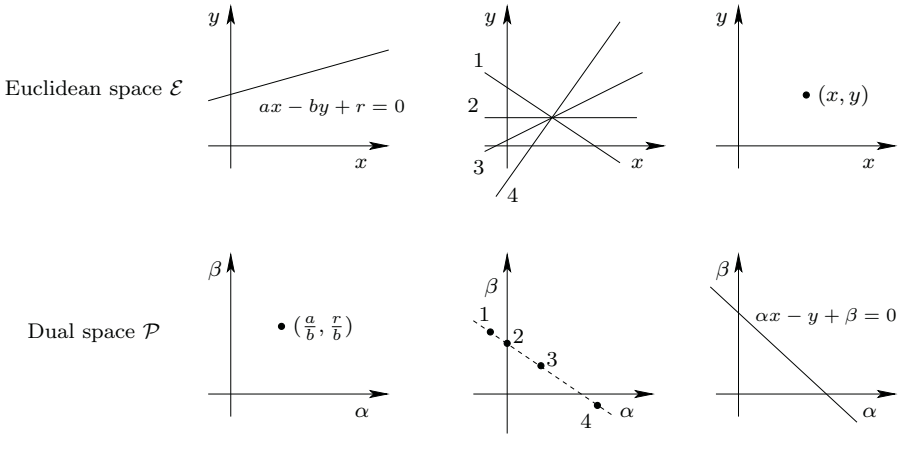


Fig. 3. Representation of the links between the Euclidean (top) and the dual spaces (bottom) for elementary geometric objects

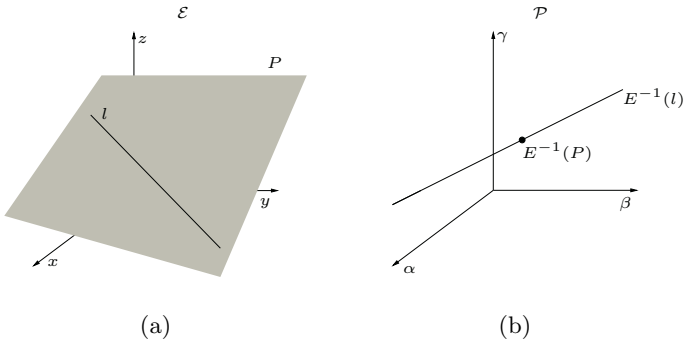


Fig. 4. Representation of a 3D line embedded in a plane in the Euclidean (a) and the dual (b) spaces

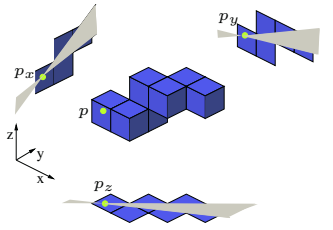
be defined in the 2D space, one for each direction. Since line parameters are represented in dual spaces, those spaces are also called parameter spaces.

Similarly, three dual spaces can be defined in 3D. One plane in the Euclidean space is represented by one point in the dual space and conversely. One 3D line in the Euclidean space $\mathcal{E}$ is represented by another 3D line in the parameter space $\mathcal{P}$. In the following, we will denote by E the operator which transforms one element of the parameter space into its corresponding element in the Euclidean space.

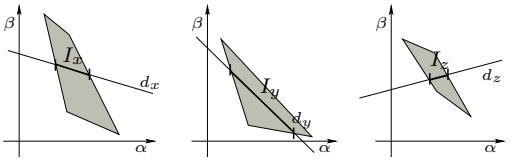
One important point for this work is how to represent in the parameter space the embedding of a 3D line into a given plane. It is actually easy to see that since a 3D line l maps to another 3D line $E^{-1}(l)$ (each point of $E^{-1}(l)$ corresponds to a plane containing l), and since a plane P maps to the point $E^{-1}(P)$, then l is embedded in P if and only if $E^{-1}(l)$ goes through $E^{-1}(P)$ (see Figure 4).

Preimages. Consider a set of pixels ϵ and a digitization scheme D. We call preimage of ϵ the set of Euclidean lines containing ϵ in their digitization. This set is represented in the dual space as a set of points.

Let us consider for instance the line l defined by $ax - by + r = 0$ where $a > 0$ and $b > 0$. Then, the standard digitization of l is the set of discrete points (x, y) fulfilling the inequalities $-\frac{a+b}{2} \leq ax - by + r < \frac{a+b}{2}$. The lines containing the point (x_0, y_0) in their digitization are those of parameters (α, β) (defined by the equation $\alpha x - y + \beta = 0$) fulfilling the inequalities $-\frac{\alpha+1}{2} \leq \alpha x_0 - y_0 + \beta < \frac{\alpha+1}{2}$. Thus, each discrete point defines two half-spaces in the parameter space, and the intersection of those half-spaces is the set of parameters of the lines containing this discrete point in their digitization. Given a set of discrete points, we call preimage of this set the convex polygon of the parameter space defined by the intersection of the constraints related to the discrete points (see Figure 5).



(a)



(b)

Fig. 5. Recognition of the three projections of a given set of voxels with a fixed point p

3 Reversible Polygonalization of a Planar 3D Discrete Curve

In this section, we present a new algorithm to compute a polygonal planar curve from a 3D discrete planar curve in a reversible way. This algorithm is based on three steps, described in the following three paragraphs.

3.1 Recognition of a 3D Standard Segment

The first step is to define an algorithm to recognize a 3D standard segment, *i.e.* the standard digitization of a 3D line segment. A standard discrete line can be defined in an analytical way:

Definition 1. *Let us consider a 3D straight line of directional vector* (a, b, c) *going through the point* (x_0, y_0, z_0)*. Then the standard digitization of this line is the set of integer points fulfilling the conditions given by the following double inequalities:*

$$\begin{aligned} -\tfrac{|a|+|b|}{2} &\leq bx - ay + ay_0 - bx_0 < \tfrac{|a|+|b|}{2} \\ -\tfrac{|a|+|c|}{2} &\leq cx - az + az_0 - cx_0 < \tfrac{|a|+|c|}{2} \\ -\tfrac{|b|+|c|}{2} &\leq cy - bz + bz_0 - cy_0 < \tfrac{|b|+|c|}{2} \end{aligned}$$

where $(b > 0$ *or* $(b = 0$ *and* $a > 0))$ *and* $(c > 0$ *or* $(c = 0$ *and* $a > 0))$ *and* $(c > 0$ *or* $(c = 0$ *and* $b > 0))$ *(otherwise, the strict and large inequalities of those equations are exchanged, see [8]).*

From this definition, we derive that if a set of voxels is a 3D standard segment, then the three projections of this set of voxels are 2D standard segments.

Consequently, in order to ensure that the three projections of the set of voxels are 2D standard segments, we compute the three preimages of those sets of pixels. If the three preimages are not empty, then this condition is fulfilled, otherwise, the set of voxels is not a 3D standard segment. Moreover, we said in Paragraph 2.2 that before the recognition step, a point is fixed inside one voxel of the set considered. Thus, the only lines which are interesting for us are the ones which go through this fixed point. As illustrated in Figure 5(a), the projection of this fixed point p onto the three coordinate planes defines three points p_x, p_y and p_z that are represented by three lines in the parameter spaces. Thus, the preimages we work one are no more polygons but simply segments denoted by I_x, I_y and I_z (see Figure 5(b)).

Nevertheless, this condition over the three projections is not sufficient to define a 3D standard segment, and a compatibility condition between the parameters of the three projections needs to be added (see [12], where non planar curves are studied, for details). In the case of planar curves, this compatibility condition does not need to be handled since we will see that it is ensured while considering the embedding of the curve into a plane.

3.2 Ensure Coplanarity

In the following, we consider the general case where the carrier plane P is defined by the equation $ax + by + cz + \mu = 0$, with a, b and c not equal to zero.

It is easy to see that any of the three preimage segments I_x, I_y and I_z can be represented in two out of the three dual spaces $\mathcal{P}_x$, $\mathcal{P}_y$ and $\mathcal{P}_z$. Indeed, those preimage segments are embedded in the planes $\alpha = 0$ or $\beta = 0$ in those dual spaces. For instance, consider the dual space $\mathcal{P}_x$ where the two segments I_z and I_y can be represented. In this space, the carrier plane P is represented by a point $E^{-1}(P)$. Thus, the 3D lines l embedded in P and containing the set of voxels considered in their digitization are those such that $E^{-1}(l)$ crosses $E^{-1}(P)$, I_y and I_z.

A reduction process of the segments I_z and I_y according to $E^{-1}(P)$ is done, as illustrated in Figure 6. The grey cone drawn on this figure represents all the lines going through one point of I_z and the point $E^{-1}(P)$. Thus, the points of I_y which do not belong to this cone must be deleted since there does not exist a line going through I_z, $E^{-1}(P)$ and those points (depicted between brackets on Figure 6). After the reduction of I_y, the reduction of I_z is computed.

This pair of reductions is computed in each dual space $\mathcal{P}_x$, $\mathcal{P}_y$ and $\mathcal{P}_z$, such that each preimage I_x, I_y and I_z is reduced twice. Finally, we have the following result:

Proposition 1. *After the six reductions presented above, the preimages of the three projections of the set of voxels S represent exactly the set of 3D lines that are solution for S and embedded in the carrier plane.*

The proof of this proposition is directly derived from the fact that one point in the preimage of one of S' projections defines a unique 3D line in the Euclidean

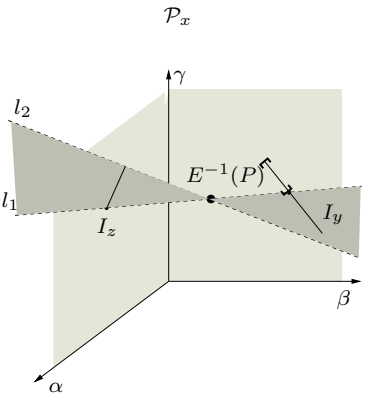


Fig. 6. Example of dual segments reductions according to the carrier plane P

space. Indeed, such a point represents a 2D line which, together with the carrier plane, defines a 3D line.

Moreover, those reductions ensure that the preimages of the projections are compatible, *i.e.* that there exist a 3D line such that its projections contain the projected set of pixels in their standard digitization (see previous paragraph).

3.3 Choice of Fixed Extremities

In Section 2.2, we saw that the first step of our algorithm is to choose an initial point in the first voxel of the curve and in the carrier plane P. This point belongs to the intersection between a voxel and a Euclidean plane. From the definition of standard plane, we know that the voxels cut by a given plane are those belonging to the standard digitization of this plane. The geometry of this intersection has been studied by Reveillès [18] and Andrès et al. [19] who show that the only five geometric shapes possible are a triangle, a trapezoid, a pentagon, a parallelogram or an hexagon (Figure 7). They moreover characterize completely the shape of the intersection between a plane and a voxel according to the position of the voxel in the corresponding standard plane. In [18], Reveillès gives the arithmeticalexpression of intersection vertices coordinates. Thus, the initial point chosen in our algorithm is simply the barycentre of the vertices computed thanks to Reveillès [18] and Andrès et al. [19] results.

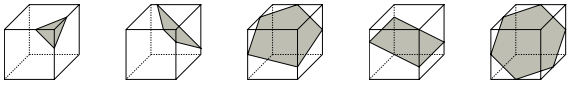
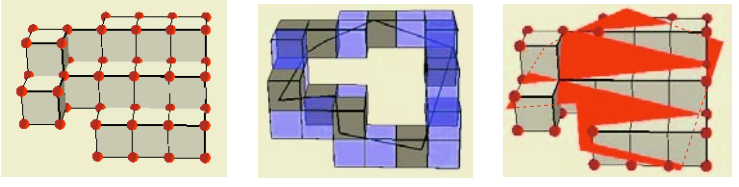


Fig. 7. The five possible intersections between a voxel and a plane

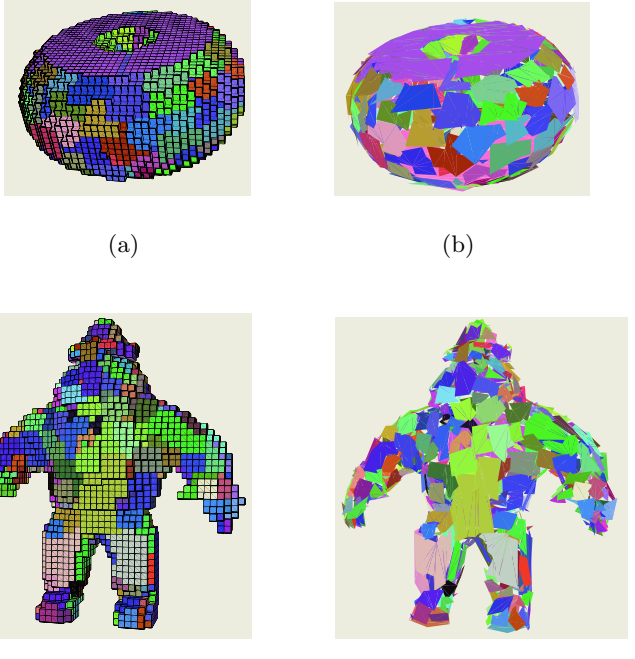
4 Application on Discrete Faces: Results

The result of this algorithm over a single discrete face is represented in Figure 8. In (a), the pointels belonging to this face are labelled. In (b), a polygonal curve embedded in the Euclidean plane that is solution for the discrete face is computed: on this figure, the boundary pointels of (a) are replaced by unit cubes centered on the pointels, so that it is easier to check that the computed line is



(a) (b) (c)

Fig. 8. Different steps of the polygonalization of a discrete face: (a) the discrete face pointels, (b) computation of the polygonal line, (c) final polygon



(a) (b)

(c) (d)

Fig. 9. Results of the polygonalization algorithm over complex objects surfaces

entirely included in the curve. Finally, the polygon computed is represented in (c), together with the discrete face it corresponds to: the standard digitization of the polygon is exactly the set of pointels of the discrete face.

On the complexity point of view, each discrete point of the face boundary is seen only once, computing the reduction of the preimages intervals according to a new discrete point is done in logarithmic time (using Grabiner algorithm, see [17]) and the reductions of the intervals presented in Proposition 1 is done in constant time. Finally, the overall complexity of this algorithm is in $\mathcal{O}(n \log n)$ where n denotes the number of boundary points for a given curve.

If we consider complex objects, this algorithm is applied over each discrete face computed by the segmentation algorithm. A set of polygons (one for each face) is computed such that the standard digitization of this set of polygons is exactly the surface pointels of the initial discrete object. Figure 9 presents two results over a torus and the image named "Al": the initial discrete object is depicted on the left, and the set of polygons computed is represented on the right.

5 Conclusion

In this paper we described a new algorithm to transform a discrete face into a discrete polygon and a Euclidean polygon. The discrete polygon is defined analytically thanks to the decomposition of the discrete face boundary into discrete analytical segments. The transformation into a Euclidean polygon is done incrementally at the same time, fixing the first extremity of each segment as the last extremity of the previous segment. Next we applied this algorithm onto each discrete face defined by a discrete surface segmentation algorithm. We get a set of polygons modelling the discrete surface in a reversible way: the standard digitization of each polygon is exactly a discrete face of the segmentation.

An interesting future work would be to improve this modelling by sewing the different polygons in order to get a surface while preserving the reversibility property. This problem may be related to the polygonal reconstruction of several adjacent discrete regions in 2D.

References

1. (Qhull) http://www.qhull.org.
2. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. In Stone, M.C., ed.: SIGGRAPH '87 Conference Proceedings, Computer Graphics, Volume 21, Number 4 (1987) 163–170
3. Kenmochi, Y., Imiya, A., Ezquerra, N.F.: Polyhedra generation from lattice points. In Miguet, S., Montanvert, A., Ubéda, S., eds.: Discrete Geometry for Computer Imagery. Volume 1176 of LNCS., Springer-Verlag (1996) 127–138
4. Borianne, P., Françon, J.: Reversible polyhedrization of discrete volumes. In: Discrete Geometry for Computer Imagery, Grenoble, France (1994) 157–167

5. Françon, J., Papier, L.: Polyhedrization of the boundary of a voxel object. In Bertrand, G., Couprie, M., Perroton, L., eds.: Discrete Geometry for Computer Imagery. Volume 1568 of LNCS., Springer-Verlag (1999) 425–434
6. Burguet, J., Malgouyres, R.: Strong thinning and polyhedrization of the surface of a voxel object. In Borgefors, G., Nyström, I., Sanniti di Baja, G., eds.: Discrete Geometry for Computer Imagery. Volume 1953 of LNCS., Uppsala, Sude, Springer-Verlag (2000) 222–234
7. Yu, L., Klette, R.: An approximative calculation of relative convex hulls for surface area estimation of 3D digital objects. In Kasturi, R., Laurendeau, D., Suen, C., eds.: IAPR International Conference on Pattern Recognition. Volume 1., Québec, Canada, IEEE Computer Society (2002) 131–134
8. Andrès, E.: Defining discrete objects for polygonalization : the standard model. In Braquelaire, A., Lachaud, J.O., Vialard, A., eds.: Discrete Geometry for Computer Imagery. Volume 2301 of LNCS., Springer-Verlag (2002) 313–325
9. Françon, J.: Discrete combinatorial surfaces. Graphical Models and Image Processing **51** (1995) 20–26
10. Françon, J.: Sur la topologie d'un plan arithmétique. Theoretical Computer Science **156** (1996) 159–176
11. Sivignon, I., Dupont, F., Chassery, J.M.: Discrete surface segmentation into discrete planes. In Klette, R., Zunic, J., eds.: International Workshop on Combinatorial Image Analysis. Volume 3322 of LNCS., Springer-Verlag (2004) 458–473
12. Sivignon, I., Breton, R., Dupont, F., Andrès, E.: Discrete analytical curve reconstruction without patches. Image and Vision Computing **23** (2005) 191–202
13. Hough, P.: Method and means for recognizing complex patterns. United States Patent, $n°3$, 069, 654 (1962)
14. Maître, H.: Un panorama de la transformation de Hough. Traitement du Signal **2** (1985) 305–317
15. McIlroy, M.D.: A note on discrete representation of lines. AT&T Technical Journal **64** (1985) 481–490
16. Lindenbaum, M., Bruckstein, A.: On recursive, $\mathcal{O}(n)$ partitioning of a digitized curve into digital straight segments. IEEE Trans. on Pattern Anal. and Mach. Intell. **15** (1993) 949–953
17. Vittone, J., Chassery, J.M.: Recognition of digital naive planes and polyhedrization. In Borgefors, G., Nyström, I., Sanniti di Baja, G., eds.: Discrete Geometry for Computer Imagery. Volume 1953 of LNCS., Springer-Verlag (2000) 296–307
18. Reveillès, J.P.: The geometry of the intersection of voxel spaces. In Fourey, S., Herman, G.T., Kong, T.Y., eds.: International Workshop on Combinatorial Image Analysis. Volume 46 of Electronic Notes in Theoretical Computer Science., Philadeplhie, Elsevier (2001)
19. Andrès, E., Sibata, C., Acharya, R., Shin, K.: New methods in oblique slice generation. In: SPIE Medical Imaging. Volume 2707 of Proceedings of SPIE. (1996) 580–589