Optimization schemes for the reversible discrete volume polyhedrization using Marching Cubes simplification

David Coeurjolly, Florent Dupont, Laurent Jospin and Isabelle Sivignon

Laboratoire LIRIS - Université Claude Bernard Lyon 1
Bâtiment Nautibus - 8, boulevard Niels Bohr
F-69622 Villeurbanne cedex, France
david.coeurjolly@liris.cnrs.fr
florent.dupont@liris.cnrs.fr
isabelle.sivignon@liris.cnrs.fr

Abstract. truc bidul machin

1 Introduction

3D discrete volumes are more and more used especially in the medical area since they result from MRI and scanners. As 2D images are composed of pixels, these 3D images are composed of voxels. This structure induces many difficulties in the exploitation and study of these objects: as each cube is stored, the volume of data is very huge which is a problem to get a fluent interactive visualization; the facet structure (voxels' faces) of the discrete object induces many problems to get a nice visualization that is necessary for medicines, as no rendering nor texture algorithm can be applied. The general idea to solve those problems is to transform discrete volumes into Euclidean polyhedra. An important property that must fulfill the Euclidean polyhedron is its reversibility up to a given digitization process (e.g. the result of the digitization must be the original discrete volume itself). In other words, no information are neither created nor lost during the transformation.

Many research activities have already been achieved to find solutions to this problem, using Euclidean geometry or discrete geometry . To get a good visualization of discrete volumes, classical methods use the Marching Cubes algorithm [11, 9], which considers local voxel configurations to replace them by small triangles. Even if these methods offer a good visualization, it does not provide a good data compression (huge number of facets) but we have a first reversible solution. Digital geometry solutions deal with a first step that segments the object boundary into pieces of digital plane [2, 4, 15, 7, 8, 13]. The digital plane is a fundamental object for this problem because reversibility properties exist. The next step consists in associating a polygon to each piece of digital plane and finally to construct the Euclidean polyhedron while sewing the polygons. The major problem of these methods is to ensure both the reversibility and the correct topology of the polyhedron.

In [3], we have proposed a polyhedrization algorithm with the following properties: it computes a reverse polyhedrization of the input digital object with the warranty that the obtained polyhedron is topologically correct. More precisely, the final polyhedron is a combinatorial 2-manifold. This algorithm is based on a simplification of the Marching-Cubes surface with digital plane segmentation information. In the following, we extend this algorithm using linear programming techniques to reduce the number of facets of the final object while preserving both the reversibility of the surface and its topology.

In section 2, we describe the preliminaries with a review of existing algorithms. In section 3, we detail the Marching-Cubes based simplification algorithm and its optimizations to obtain a polyhedron from a discrete object with a small number of facets.

2 Digital plane recognition and Marching-Cubes surface BOFFF

2.1 The Marching-Cubes algorithm

Let us assume a discrete 3D image that maps a value $V(x, y, z) \in \mathbb{R}$ to each grid point $(x, y, z) \in \mathbb{Z}^3$. The image V can also be considered as a density function on a subset of $\mathbb{Z}^3$. The Marching-Cubes (MC for short) algorithm was first introduced by Lorensen and Cline [11] to extract a triangulated surface from V corresponding to an iso-density value. The first application of this work was the visualization of iso-density surfaces in medical imaging. We first consider cubic cells of coordinate (x, y, z) whose vertices are placed on the 8 input samples $(x + i, y + j, z + k)$ of the volume data, with $i,\ j,\ k \in \{0, 1\}$. The triangulated iso-surface given by the Marching-Cubes algorithm is locally computed according to the way of the surface intersects each cell of V using a look-up table with 14 possible configurations (see figure 1). The coordinates of the MC vertices along an edge of a cell is given by an interpolation process between the values of V and the chosen iso-level.

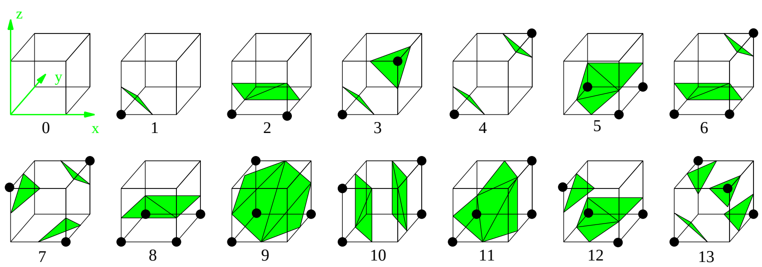

Fig. 1. The 14 different standard triangulations of the Marching-Cubes algorithm.

Note that some of original Lorensen and Cline's configurations may lead to ambiguities in the reconstruction and thus construct surfaces with holes. To have results on the topology of the reconstruction, we need a process that disambiguates the configurations according to the topology of the input discrete surface. The configurations presented in the figure 1 correspond to a $(18, 6)-$surface [9, 10]. Hence, if the binary object is 6-connected, the triangulated surface is a combinatorial 2-manifold, *i.e.* closed, oriented and without self crossing [9, 10]. If a binary object is considered, *i.e.* if $V(x, y, z) \in \{0, 1\}$, for all x, y and z, from [3], we have the following lemma (see figure 2):

GAUSSS et OBQ!!!

Lemma 1 ([3]). *The Marching-Cubes surface of a digital object, obtained with a an iso-level in* $]0, 1[$*, is a reversible polyhedrization of the binary object according to the Object Boundary Quantization model.*

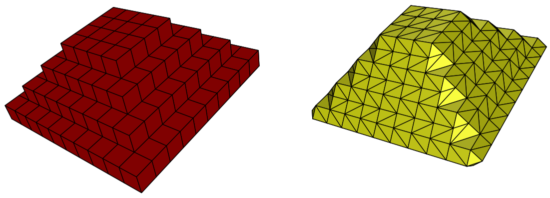

Fig. 2. A binary 3D object and the obtained Marching-Cubes surface.

In the following, the reversible polyhedrization we propose is based on a simplification of the Marching-Cubes surface.

2.2 Digital Plane segmentation of a discrete surface

Notations $\{p, q\}$ pour un surfel, $]p, q[$

-def plan/preimage -def segm -liens avec]p,q[- 1 sommetMC = 1 surfel -enoncer le theo : si on deplace sur]pq[ca change rien

2.3 The reversible reconstruction problem

From the literature, given a segmentation of a discrete surface into digital planes, we classify the polyhedrization algorithms as follows:

Top-down approaches: we first associate to each piece of digital plane an Euclidean 3D polygon. Hence, the reversibility is assured on these facets by the digital plane definition. Then, a complex task is performed to glue all the 3D polygons in order to obtain a well defined surface, while maintaining the reversibility property on edges and vertices [7, 5, 12, 14]. During this step, some patches, *i.e.* extra 3D polygons, may be locally inserted to sue two polygons. Another solution to ensure that the reversibility property is maintained on the edges is to change the DPS recognition process and to consider subsets of the preimage [6].

Bottom-up approaches: in this case, we start from a reversible and topologically correct surface (*e.g.* the Marching-Cubes) and we reduce the huge number of facets using digital plane segmentation information [3]. To ensure the reversibility and the topology properties of the final polyhedron, we just have to prove that each elementary modification of the MC surface do not violate the initial properties.

In the following, we consider the second class of algorithms.

3 Marching-Cubes simplification and optimization

Since there is a one-to-one and onto mapping between the MC vertices and the surfels of the input discrete object, we introduce a label on MC triangles as follows:

Definition 1 (Homogeneous and non-homogeneous triangle). *Let T be a triangle of the MC surface, T is homogeneous (H for short) if its three vertices are associated to surfels belonging to the same digital plane. Otherwise, T is called non-homogeneous, NH for short). If T is homogeneous, T is labelled with the digital plane segment label of its vertices.*

Furthermore, we can define the 2-NH triangle (resp. 3-NH triangle) if the number of distinct discrete plane segments associated to its vertices is exactly 2 (resp. 3).

In the following, we introduce a projection process of a MC vertex onto an Euclidean plane: let v be a MC vertex and $p,\ q$ be the two voxels (p belongs to the object and q to the background) such that v is associated the surfel $\{p, q\}$. Thus, only the projection of v onto an Euclidean plane P according to the $\boldsymbol{pq}$ direction is considered.

3.1 Homogeneous triangles case

Using [3], we have the following result on H-triangles:

Lemma 2 ([3]). *Let v be a vertex of an H-triangle, let P be an Euclidean plane from the preimage of the discrete plane associated to the triangle. The projection of v onto P does not change neither the reversibility nor its topological properties of the global surface.*

This lemma can easily be proved by definition and properties of the discrete plane segmentation process and using lemma REF VERS LE LEMME DANS LA PARTIE DPS.

In [3], the authors design a simplification algorithm based on the previous lemma to remove the homogeneous triangles: let S be a connected set of H-triangle with the same label, they extract from the DPS preimage associated to S an Euclidean plane P. Then, if we project all vertices of S onto P, triangles in S become coplanar. Finally, a post-processing step converts all connected sets of H-triangles with the same label into a single facet. At each step of this algorithm, we ensure the reversibility property and the final surface is still a combinatorial 2-manifold. Note that no assumption is needed during the choice of the plane P.

As presented in Figure XXXXX, for each connected set of H-triagnle with the same label, we have obtain a facet. NH-triangles allow to sue together all the facets maintaining the topological property of the polyherdon.

In the next section, we present a linear programming framework to extract, from the preimage, an appropriate Euclidean plane P in order to remove NH-triangles.

3.2 Non-Homogeneous triangles case

The basic idea to remove the NH-triangles consists in adding linear constraints in the DPS preimages. Then, the choice of the Euclidean plane P is made by a linear inequality system solver.

However, to have an efficient algorithm, we restrict the problem using the following two heuristics:

Local analysis: let us examine the 2D reconstruction presented in the figure 3. If we consider the OBQ scheme, both polygons are correct regarding to the reversibility property. However, the visual aspect of the dashed polygon compared to the initial binary object is worse than the bold one. Hence, to restrict our reconstruction to a polyhedron defined in the cells defined by the MC surface. More precisely, when we a modification of an NH triangle is performed, the result must belong to the MC cell associated to the triangle. This heuristic is a restriction on the possible reconstruction but it allows to design efficient algorithms since the surface properties (reversibility and topology) can be ensure using local analysis. Other arguments justifying this approach are based on the fact that the OBQ digitization scheme associated to MC surfaces is not a complete digitization model [1].

Linear programming problem in dimension 3: during the DPS recognition process, we have used linear programming algorithms in dimension 3 to compute the preimages. In this optimization process, the dimension of the linear constraint system that conducts the NH triangle simplification must be bounded by 3. Even if this choice influence and reduce the scope of the algorithm, we limit the computational cost of the linear programming solver. Futhermore this process is still consistent with the DPS preimage parameter space.

Using these heuristics, the process can summarized as follows: when a NH triangle T is considered, two different cases occur if we want to simply the surface (see Figure 4), we can:

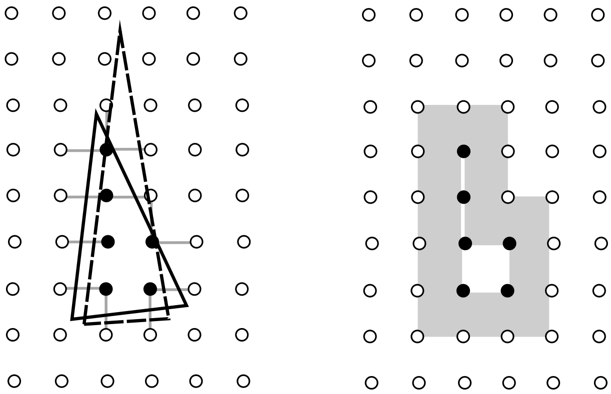

Fig. 3. *(left):* Two possible polygonalizations of a binary object (dark grey dots). The grey segment represent the $]pq[$ intervals in the OBQ scheme. *(right):* The light grey area define the allowed location of the polygon vertices we use (hence only the bold polygon in the left figure would be considered in our algorithm).

- remove an edge of T : in that case, the edge is collapsed into a point. Furthermore, such a point belong to a face of the MC cell containing T. Hence, a 2D processing is used to constraint the new point to be in the MC cell (see figure 5).
- Remove a triangle : the triangle is collapsed into a single point and we have to ensure that the point belong to the MC cell.

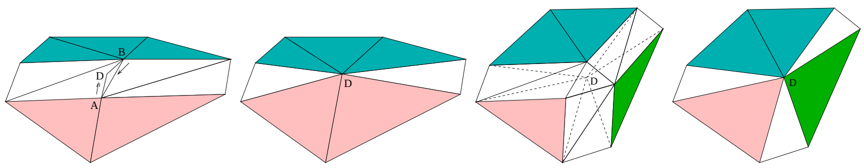

Fig. 4. Illustration of the removal of an edge and a triangle of the MC surface.

Let T be a NH-triangle, to check if an edge of T can be removed, we consider the three MC cell faces on which T edges are defined (see figure 5). From the three edges of T, at least one is such that its vertices do not belong to the same discrete plane segment. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the two preimages associated to such edge e. The edge e can be removed if for all $P_1 \in \mathcal{P}_1$ and $P_2 \in \mathcal{P}_2$, the intersection of P_1 and P_2 belongs to the MC cell face associated to e. It is not possible to linearly express those conditions without changing the dimension of the linear programming problem. To solve that point, we consider two approaches to obtain sufficient conditions on the intersection of P_1 and P_2.

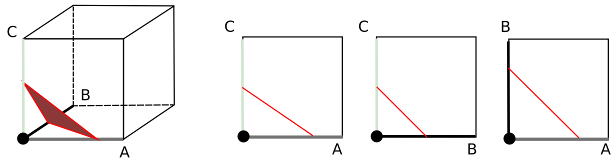

Fig. 5. Illustration of the 2-D decomposition of a MC cell into its faces in order to decide if an edge of a cell triangle can be removed.

Global simplification First of all, we have a global simplification process to remove NH elements. In this step, we only consider the simple MC configurations, *i.e.* the configurations with a single surface patch (1, 2, 3, 5, 8, 9, 11 in the figure 1). In the other configurations, we have to check the intersection of the two surface patches and we cannot add linear constraints to ensure the topology during the global simplification. The analysis of these configurations is done during the greedy simplification.

To obtain sufficient conditions on the intersection of P_1 and P_2, we can list three cases (see Figure 6), depending of how many voxels belong to the object on the considered face of the MC cell.

If we have only one voxel A belonging to the object on a face $ABCD$, then we will have the plane P_1 associated to surfel $\{A, B\}$ cross the segment CD and P_2 associated to surfel $\{A, D\}$ cross the segment BC. Thus we ensure that the intersection of P_1 and P_2 is inside the square. If we consider the case where two voxels belongs to the object on a face, then their is none interesting linear constraints. If we consider the case where only one voxel C does not belong to the object on a face $ABCD$, then we will have the plane P_1 associated to surfel $\{D, C\}$ cross the segment AB and the plane P_2 associated to surfel $\{B, C\}$ cross the segment AD. As in the first case, those conditions ensure that the intersection of the two planes is inside the square (see Figure 6). Finally, these constraints lead to simple linear constraints in dimension 3 that reduce both the preimages $\mathcal{P}_1$ and $\mathcal{P}_2$ to preimages $\mathcal{P}'_1$ and $\mathcal{P}'_2$. Hence, if $\mathcal{P}'_1$ and $\mathcal{P}'_2$ are not empty, whatever $P_1 \in \mathcal{P}'_1$ and $P_2 \in \mathcal{P}'_2$, the intersection of P_1 and P_2 belongs to the face $ABCD$ of the MC cell, ensuring the reversibility of the modified surface. If one of the two preimages is empty, the edge is not removed.

Greedy simplification This step consists in fixing planes one by one, to have more flexible constraints on the preimage of the remaining planes, and to be able to handle more cases. So the scheme is to fix one Euclidean plane P_1 (arbitrarily chosen in its associated preimage $\mathcal{P}_1$). Then, if T is a NH triangle edge associated to the DPS represented by the Euclidean plane P_1 and another DPS with preimage $\mathcal{P}_2$, we insert linear constraints on $\mathcal{P}_2$ to control the intersection between P_1 and P_2. Since p_1 is given, the intersection point in the MC cell face can be given in a linear form it the $\mathcal{P}_2$ parameter space. Indeed let us consider that a plane P_1 and a mobile plane P_2 on a face $ABCD$ (see figure 7), if I is the intersection of P_1 and P_2, to ensure that I is inside the square $ABCD$, we have

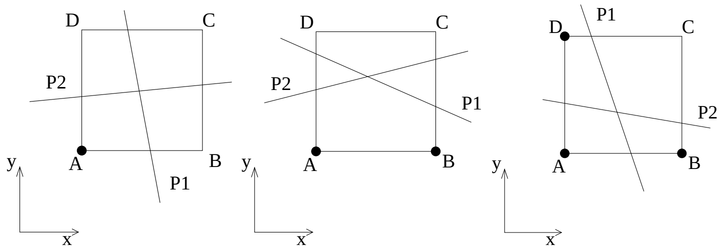

Fig. 6. The three possible cases to define sufficient conditions to remove an edge of a NH triangle.

the constraints:

$$\begin{cases} x_A < x_I < x_A + 1 \\ y_A < y_I < y_A + 1 \end{cases}$$

As x_A and y_A are constants and x_I, y_I only depend on the $\mathcal{P}_2$ parameters, this equations give linear constraints.

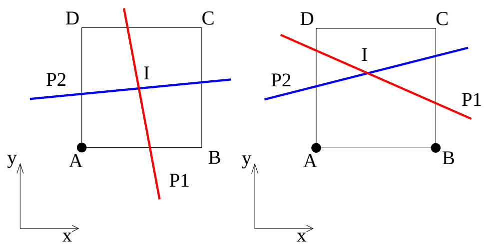

Fig. 7. Illustration of the greedy simplification approach.

Finally, if we fix a plane P_1 for a DPS, we propagate this piece of information to each neighboring DPS preimages. This process is greedy since we do not backtrack on the choice of P_1. Once all neighboring DPS have been considered, the greedy step can choose another Euclidean plane in another preimage and the process repeats.

AUTRES CAS !!!!!

3.3 Overall algorithm

In this section, we skecth the overall simplification algorithm based on the two approaches presented above.

1. Compute the MC surface
2. Segment the discrete object into digital plane segment
3. Optimization on NH triangles, i.e. find an Euclidean plane in each DP preimage:

(a) Step 1 : global optimization
(b) Step 2 : greedy optimization to remove other cases
4. Vertices displacements and simplification of coplanar triangles.

Lemma 3. *The algorithm presented above construct a reversible polyhedron which is a combinatorial 2-manifold.*

Proof. The proof is straightforward according to lemma XXX(lemme pq). To prove the topology, since the MC surface is a combinatorial 2-manifold [9, 10] and since we can locally prover that treatments on both H triangle and NH triangle to not change the topology, the final overall surface is still a combinatorial 2-manifold (see [3] for details on the H triangle treatment). Furthermore, since each new elements (facets and vertices) belongs to the MC cells in which the surface is defined, the OBQ digitization of the final polyhedron exactly corresponds to the input set of voxels. Note that since the topology is preserved, the polyhedral surface is still oriented and the OBQ digitization scheme is still well defined.□

4 Experiments and results

The figure 8 and the table 1 show some experiments. We can notice that in all presented cases, the global removal rate is always greeter than 75% which also holds for most experimented objects. The NH triangle removal shows an improvment of at least 30% and up to 60% compared to the initial algorithm.

object	# MC triangles	# NH	# removed NH	global(and NH) removal rate
`pyramid_4`	512	170	106	87% (62%)
`sphere_5`	824	326	164	80% (50%)
`rd_cube_7`	2024	304	271	98% (89%)
`sphere_10`	3656	1456	535	75% (37%)
`catenoid_10`	7928	2161	750	82% (35%)

Table 1. Some results of the presented work

5 Conclusion

References

1. E. Andrès. Discrete linear objects in dimension n: the standard model. *Graphical models*, 65(1–3):92–111, May 2003.
2. P. Borianne and J. Francon. Reversible polyhedrization of discrete volumes. In *4th Discrete Geometry for Computer Imagery*, pages 157–168, Grenoble, France, sept 1994.

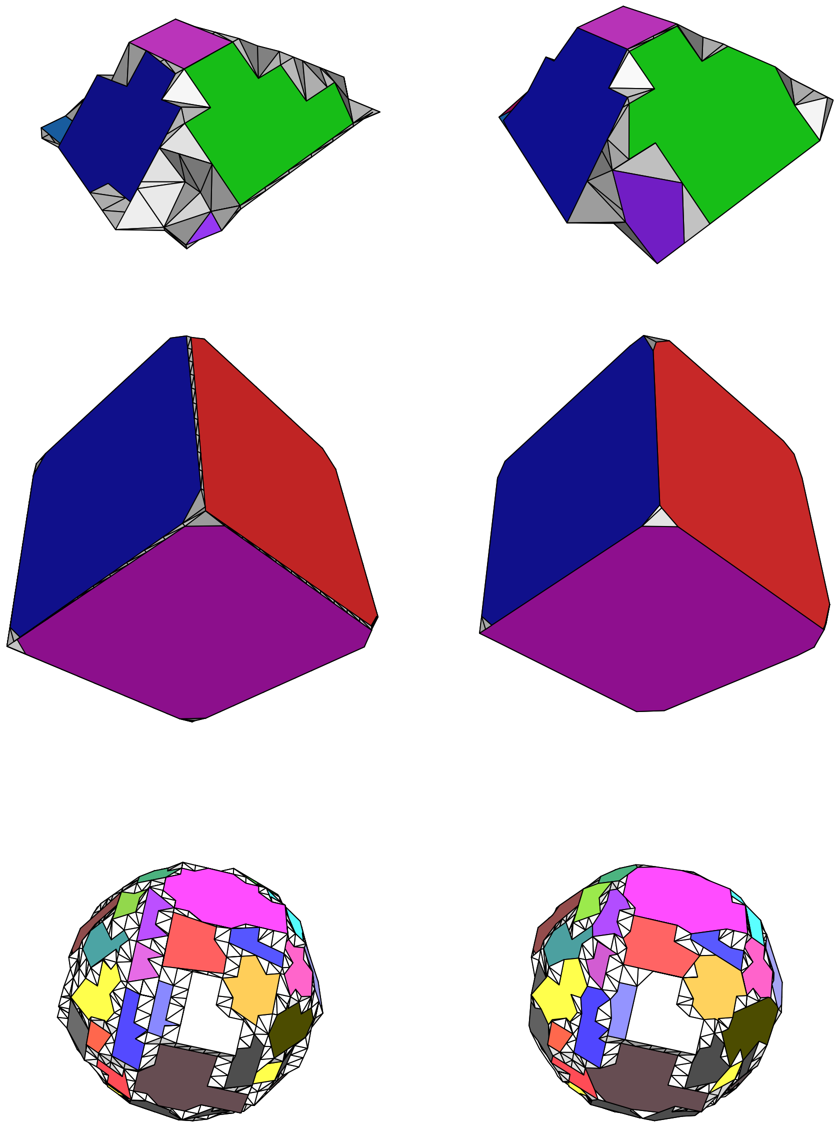

Fig. 8. Comparaison between the normal simplification, and the simplification including NH trianges removal

3. D. Coeurjolly, Alexis Guillaume, and I. Sivignon. Reversible discrete volume polyhedrization using marching cubes simplification. In *SPIE Vision Geometry XII*, volume 5300, pages 1–11, San Jose, USA, 2004.
4. I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets.* PhD thesis, Université Louis Pasteur, 1995.
5. I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets.* PhD thesis, Université Louis Pasteur, 1995.
6. M. Dexet. *Design of a topology based geometrical discrete modeler and reconstruction methods in 2D and 3D.* PhD thesis, Laboratoire SIC, Université de Poitiers, December 2005. (in French).
7. J. Francon and L. Papier. Polyhedrization of the boundary of a voxel object. In *8th International Workshop in Discrete Geometry for Computer Imagery*, pages 425–434. Springer-Verlag, LNCS, 1568, 1999.
8. R. Klette and H. J. Sun. Digital planar segment based polyhedrization for surface area estimation. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *International Workshop on Visual Form 4*, volume 2059 of *Lect. Notes Comput. Sci.*, pages 356–366. Springer-Verlag, 2001.
9. J.-O. Lachaud. Topologically defined isosurfaces. In *6th Discrete Geometry for Computer Imagery*, pages 245–256. LNCS 1176, Springer-Verlag, 1996.
10. J.-O. Lachaud and A. Montanvert. Continuous analogs of digital boundaries: A topological approach to iso-surfaces. *Graphical Models and Image Processing*, 62:129–164, 2000.
11. W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. In M. C. Stone, editor, *SIGGRAPH '87 Conference Proceedings (Anaheim, CA, July 27–31, 1987)*, pages 163–170. Computer Graphics, Volume 21, Number 4, July 1987.
12. I. Sivignon, F. Dupont, and J. M. Chassery. Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1):25–43, 2003.
13. I. Sivignon, F. Dupont, and J.-M. Chassery. Decomposition of a 3D discrete object surface into discrete plane pieces. *Algorithmica, Special Issue on Shapes Algorithmics*, 38, number 1, 2004.
14. I. Sivignon, F. Dupont, and J.-M. Chassery. Reversible polygonalization of a 3D planar discrete curve: Application on discrete surfaces. In *International Conference on Discrete Geometry for Computer Imagery*, pages 347–358, 2005.
15. J. Vittone and J.-M. Chassery. Recognition of digital naive planes and polyhedization. In *Discrete Geometry for Computer Imagery*, number 1953 in LNCS, pages 296–307. Springer, 2000.