

Polyhedrization of the Boundary of a Voxel Object

Jean Françon and Laurent Papier

Laboratoire des Sciences de l'Image, d'Informatique et de Télédétection
Pôle API, 67400 Illkirch, France
tel. (33)0388416343
{francon, papier}@dpt-info.u-strasbg.fr

Abstract. A voxel object (finite set of voxels) is considered in the cuberille approach (more precisely, the 3D cell complex approach). Its boundary is a set of surfels (faces of voxels). We assume, without loss of generality, that this set of surfels is a polyhedron whose faces are surfels. These faces can be agglomerated in such a way that the boundary is a polyhedron whose faces are topological disks of standard arithmetic planes; this new kind of polyhedron is called a *discrete standard polyhedron*. Thus, these new faces are generally much bigger than one surfel, and a discrete standard polyhedron has generally a less smaller space complexity than the starting set of surfels. This process, called polyhedrization or facetization, is the 3D extension of the known polygonalization of 2D discrete curves. The other main properties of this polyhedrization are the non-uniqueness, and the reversibility, i.e. starting from the discrete standard polyhedron, the boundary can be exactly computed back again. A polyhedrization algorithm is presented in this paper. It uses a recent algorithm for recognizing standard arithmetic planes. Examples of polyhedrizations of synthetic and natural objects are given. Examples of application to the visualization of the boundary of a voxel object are also given.

1 Introduction

Let us consider a finite set of voxels, called a *voxel object*. A *voxel* is a unit cube, or *3-cell* in the 3D cell complex approach of Kovalevsky [10] [11], centered at a point of integer coordinates. A face of a voxel is called a *surfel* (which is a unit square), or *2-cell*, an edge is called *linel*, or *1-cell*, and a vertex is called *pointel*, or *0-cell*. For technical purposes in the present paper, a voxel is translated so that it is centered at a point of half integer coordinates; thus, any pointel have integer coordinates and any integer coordinates point is a pointel. The (*surfel*) *boundary* of a voxel object is the set of surfels incident to a voxel of the object and to a voxel of its complementary. This is the well known boundary concept in the "cuberille" approach of 3D discrete imaging, and of the cell complex approach. We assume, without loss of generality, this boundary is connected, and, moreover, it is a polyhedron whose faces are surfels; that is, the boundary is

a connected *combinatorial 2-manifold* without boundary whose faces are unit squares (each line of the boundary is incident to exactly two surfels, and each point of the boundary is incident to exactly one umbrella of surfels, see definitions in [6] [7] [8]). But we know from [7] [8] that a standard arithmetic plane is an infinite orientable combinatorial 2-manifold without boundary whose faces are unit squares. Thus, the boundary of a voxel object can be decomposed (or encoded) into a set of topological disks of standard planes; that is, this boundary can be decomposed into a new kind of polyhedron, here called *discrete standard polyhedron* : A polyhedron whose vertices are pointels, and whose faces are not always exactly Euclidean planar segments, but always topological disks of standard arithmetic planes.

This decomposition (or encoding) is called *polyhedrization*, or *facetization*. It is obviously non unique. It is reversible, *i.e.* the boundary can be exactly back computed by starting from the discrete standard polyhedron. This reversibility property is not difficult to prove; but it is not addressed here. Note that it is not the case for the various methods, *e.g.* the "marching cube" algorithm, that produce a Euclidean polyhedron approximation of the boundary. Another important property of the discrete standard polyhedron is that it generally needs less information than a set of surfels for representing the boundary.

The authors of [1] were the first to address themselves to the same problem. Another polyhedrization (of a discrete sphere only) was done in [4] [5], with naive arithmetic planes instead of standard planes. The solution of [1] was not entirely satisfactory, partly because it was not completely discrete (it uses a classical least square fit algorithm). The solution given in the present paper uses the standard plane recognition algorithm presented in [9]; this algorithm makes sure that a given set of pointels belongs to an arithmetic plane and computes the parameters of the equation of that plane.

Note that the problem of polyhedrization of a set of surfels is a 3D extension of the well studied problem of polygonalization of a 2D discrete curve; see [2], [5], [15], [16], [11], and their references. The present paper has the same goals for 3D images than [11] for 2D images; thus, it is a partial 3D extension of [11]; in [11], the study of the reversibility and of the complexity reduction was well developed; it is not the case here, and left to another paper. In the present paper, we restrict ourselves to the study of the feasibility of our polyhedrization.

In section 2 the definitions and the basic used properties are given. In section 3 the problem is precisely stated, and the designed polyhedrization algorithm is presented and discussed. Experimental results on synthetic as well as a natural object are given in section 4. Application to the rendering of the boundary of voxel objects is shown in section 5. A prospective conclusion ends the paper.

2 Definitions and basic properties

The notion of surface used in the present paper is that of a two-dimensional combinatorial manifold in the classical combinatorial topology [6]. The basic notions

are that of *combinatorial 2-manifold without boundary (or closed)*, *2-manifold with boundary (or open)*, whose elements are *vertices*, *edges*, *faces*, their *incidence relations*, their *adjacency relations*, *umbrella*, *loop* or *boundary* of a face, *connectivity*, *orientability*. We use here these notions, and the notion of a *topological disk* which is an orientable combinatorial 2-manifold with a boundary of one loop only (or equivalently, of Euler characteristics one). A face is a topological disk.

An *arithmetic plane* is the set of points (x, y, z) of the discrete space \mathbb{Z}^3 satisfying the inequalities

$$\mu \leq ax + by + cz < \mu + \omega$$

where all parameters are integers and $\omega > 0$. The parameter ω is called the (*arithmetic*) *width* of the plane. The triple (a, b, c) is the *normal* to the plane; a, b, c, μ are the *coefficients* of the plane. These planes are basic in the arithmetical geometry introduced by Jean-Pierre Reveillès [14]. A *standard arithmetic plane* is an arithmetic plane Π of width $\omega = |a| + |b| + |c|$. Let us recall the result of [7] [8]. Let G be the graph of the 6-connectivity relation in Π , let F be the set of squares, i.e. cycles of 4 vertices, of G . The pair (G, F) is an orientable two-dimensional combinatorial 6-connected 2-manifold without boundary.

We call *discrete standard polyhedron* a combinatorial orientable 2-manifold without boundary whose any face is a topological disk of a standard arithmetic plane, called a *standard face*. The vertices of a *discrete standard polyhedron* are integer points (here pointels), and called *standard vertices*; its edges are 6-connected paths of points called *standard edges*, whose extremities are standard vertices and not containing other standard vertices. The figure 1 shows a view of the standard edges of a polyhedrization of a discrete hexahedron, which is a two times chamfered cube; the linels of the boundary of the discrete standard polyhedron are displayed like 3D unit segments.

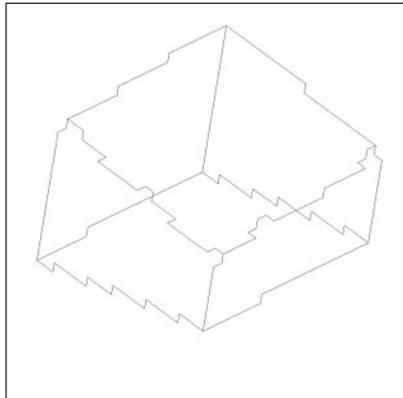


Fig. 1. Hexahedron

3 Polyhedrization: Problem and algorithm

Let \mathbf{V} be a voxel object whose voxels are centered at half-integer coordinates points, and let \mathbf{B} its boundary (see definitions in section 1). Thus the pointels of \mathbf{B} have integer coordinates only. Assume \mathbf{B} is connected (if not, we treat separately its connected components). Moreover, assume \mathbf{B} is a polyhedron whose faces are surfels, that is a combinatorial 2-manifold without boundary whose faces are surfels and whose vertices are pointels (generally, a line of \mathbf{B} can be incident to 4 surfels of \mathbf{B} , instead of 2, and a pointel of \mathbf{B} can be incident to 2 umbrellas of surfels of \mathbf{B} , instead of just one). This last assumption is made in order to simplify this paper, and can easily be dropped by different methods discussed in [12]. Because any surfel is a topological disk of a standard arithmetic plane (an orthotropic plane), \mathbf{B} is a discrete standard polyhedron whose standard faces are surfels, standard vertices are pointels, and standard edges are linels. The problem called *polyhedrization* is to agglomerate surfels of \mathbf{B} in order to make a discrete standard polyhedron with larger faces. The reduction of space complexity is obvious, as well the non uniqueness of the solution.

A simple, nevertheless instructive, example is that of an object of one voxel only, centered at $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Its boundary has 6 surfels; the 3 ones incident to the pointel $(0, 0, 0)$ (resp. the pointel $(1, 1, 1)$) define an umbrella of the standard plane

$$0 \leq x + y + z < 3 \quad (\text{resp. } 0 \leq x + y + z - 1 < 3).$$

Thus, the boundary of the one voxel object can be decomposed into two topological disks of standard planes (note that it is not possible for a Euclidean polyhedron to be reduced to two faces only); moreover, these planes are different and parallel (same normal), which is not possible for two adjacent faces of a Euclidean polyhedron.

The design of a polyhedrization algorithm is rather straightforward. Here, \mathbf{B} is a connected set of surfels. The set of faces of the searched discrete standard polyhedron is denoted by \mathbf{P} . A face of \mathbf{P} is represented by the quadruple of the coefficients of its standard plane, and by its boundary which is a loop of integer points (pointels). Any surfel of \mathbf{B} must belong to one and only one face of \mathbf{P} . The problem is : Given \mathbf{B} , compute \mathbf{P} .

The (standard) faces of \mathbf{P} have to be computed one by one, starting from a seed of one surfel (but for heuristic reasons we prefer to start with a seed of one umbrella, as long as possible). Thus some traversing through the surfels of \mathbf{B} has to be done. For each surfel of \mathbf{B} we try to affect it to the current standard face under construction \mathbf{f} . If we can't add any surfel of \mathbf{B} to \mathbf{f} , we start a new current face with (arbitrarily) the next non affected surfel with respect to the order of traversal of \mathbf{B} . In order to assign the current surfel \mathbf{s} to the current standard face \mathbf{f} we must check whether by adding \mathbf{s} to \mathbf{f}

1. we keep a topological disk;
2. the pointels of \mathbf{s} and of \mathbf{f} belong to a standard plane.

The answer to the first condition is obtained by checking whether the boundary of the union of \mathbf{s} and \mathbf{f} is a loop; it is easy and purely topological. The answer

to the second condition is obtained by the standard plane recognition algorithm presented in [9], based on the known Fourier's algorithm for solving a set of inequalities. This algorithm provides not only a yes or no answer, but also the cone of the solutions, which is a set of inequalities for the coefficients of the standard plane solution, if it exists. Moreover, a numerical solution can be easily computed, especially a numerical normal, which is practically needed.

Because of the first condition, the current surfel \mathbf{s} is chosen in the set $\mathbf{q}(\mathbf{f})$ of non assigned surfels that are not in \mathbf{f} and are adjacent to a surfel of \mathbf{f} (thus a linel of \mathbf{s} belongs to the boundary of \mathbf{f}). If \mathbf{s} is affected to \mathbf{f} defining a new current face \mathbf{f}' , the update of $\mathbf{q}(\mathbf{f})$ to compute $\mathbf{q}(\mathbf{f}')$ is by deletion of \mathbf{s} from $\mathbf{q}(\mathbf{f})$ and by adjonction of a set \mathbf{t} , maybe empty, of surfels. The set \mathbf{t} is the set of all surfels adjacent to \mathbf{s} that are not in \mathbf{f} , and non assigned. To each surfel of \mathbf{t} is associated an integer, called its *depth*, which is one more the maximal depth in $\mathbf{q}(\mathbf{f})$ (the surfels of $\mathbf{q}(\mathbf{g})$, where \mathbf{g} is the seed, are initialized to 1). We have decided to constraint the choice of the current \mathbf{s} in $\mathbf{q}(\mathbf{f})$ to be restricted to those surfels of minimal depth, a kind of breadth first search; in this way, the growing of the current face looks like the propagation of a circular wave.

The output of the algorithm is a set of standard faces; each standard face is defined by its standard plane and by a circular permutation of the linels (or the pointels) of its boundary. These linels (resp. pointels) can be considered as the standard edges (resp. vertices) of the computed polyhedron. Another possible way is to call standard vertices only those previous vertices that are incident to at least three standard faces; then, a standard edge is a path of linels bounding two adjacent standard faces, ending with two standard vertices, and not incident to other standard vertices.

It is clear that the computed \mathbf{P} heavily depends on several arbitrary choices : The ordering of the traversing through \mathbf{B} , the chosen seed for starting a new current face, the chosen surfel \mathbf{s} in \mathbf{q} , for enlarging the current face. Several questions occur naturally: How is it possible to control the number of faces of \mathbf{P} , or the sizes of the faces, or to eliminate the smaller faces, ..., by acting on these choices. We do not know any answer.

For some practical use in medical imaging (Philippe Borianne, private communication), the limitation of the size of the standard faces could be required; another interesting limitation is that of the depth of the surfels in \mathbf{q} . This can be easily done. All details of the algorithm are given in [12]. More constrained variants are given in [12], and not detailed here; one variant is in the spirit of [16], in order to get a better retrieval of voxel objects defined by a discretization on a Euclidean polyhedron.

4 Results

The polyhedrization algorithm of section 3 was programmed and experimented. Some results are now given and discussed. All the figures are obtained by the classical methods of rendering of polygons (flat shading or Gouraud shading with a one light illumination model); all the surfels of a standard face are displayed

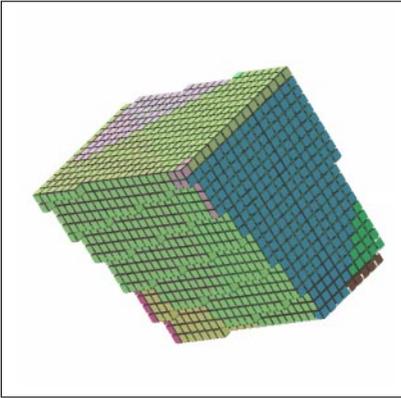


Fig. 2. Hexahedron

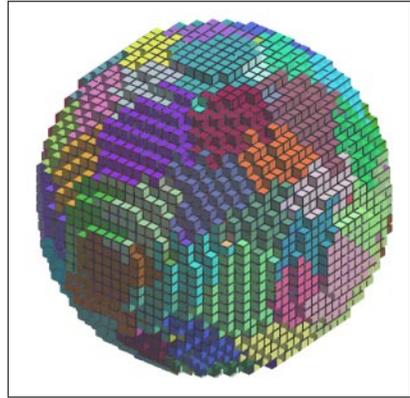


Fig. 3. Sphere ($R=15$)

with the same color, more or less darkened by the used model; adjacent standard faces have different colors.

The figure 2 shows the standard faces obtained for a discrete hexahedron which is a twice chamfered cube (the voxel object is the intersection of six half-spaces). In this example, the number of computed standard faces is not 6, but 11. The number of surfels (*size*) of the boundary is 1962; thus, the average size of the standard faces is 178.36; the maximum size is 475, the minimum is 13.

The same is performed in figure 3 for a discrete sphere of radius 15 (the voxel object is a bowl), and in figure 4 for the boundary of a X-ray 3D scanner image of a mushroom.

For the sphere, the 48 symmetries are lost. The number of computed standard faces is 135. The number of surfels of the boundary is 4494; thus, the average size of the standard faces is 33.29; the maximum size is 94, the minimum is one; there are 48 standard faces of size 5 or less; the distribution of sizes is rather scattered.

For the mushroom, the number of computed standard faces is 1019. The number of surfels of the boundary is 26392; thus, the average size of the standard faces is 25.90; the maximum size is 332, the minimum is one; there are 475 standard faces of size 5 or less; the distribution of the sizes is rather scattered.

A general observation is that the number of small standard faces is rather large. Decreasing this number, if possible, is an open problem.

5 Application to visualization

The surfel boundary of a voxel object have been rendered, in section 4, like a set of Euclidean polygons, by the classical methods of rendering of computer graphics, because surfels are 3D Euclidean polygons. Methods for smoothing the resulting visual staircase effects have been studied (see [13] for references and a new method). The same is true for a discrete standard polyhedron, although the

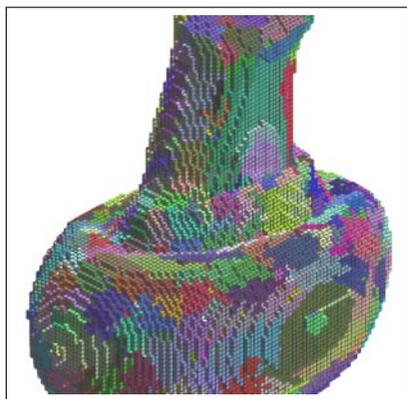


Fig. 4. Mushroom

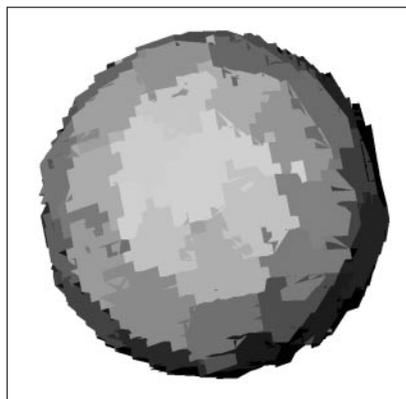


Fig. 5. Sphere ($R=15$) standard faces with flat shading

vertices of a standard face do not necessary all belong to a Euclidean plane; this is because some classical softwares of rendering begin by a triangulation of the entered polygons, and then display each obtained triangle by using an entered or a computed normal which can differ from the true normal of the triangle. Thus, it is possible to use these softwares with a standard face represented by a normal and a sequence of pointels of its boundary.

We present here a few results (obtained with the rendering library OpenGL), and a short discussion. The practical interest of this new way of visualization of discrete 3D data requires a much more developed study, which will be done in another full paper.

Figure 5 (resp. 8, 11) shows a flat shading view of the radius 15 discrete sphere (resp. the mushroom, the hexahedron). The line length is large enough for showing up the non Euclidean planarity of the standard faces. Obviously, the visual defects disappear if the length of a line is about that of a pixel of the screen.

In order to avoid some possible defects of the previous visualization method, the following variant has been experimented. Instead of entering the rendering software with the full standard edges of a standard face, we replace any standard edge by the Euclidean 3D segment defined by the two extremities of the edge; in other words, we enter the rendering software with, for each standard face, a cyclic permutations of its standard vertices. The obtained polyhedron is called a *semi discrete* polyhedron. Note that its faces are generally not Euclidean planar.

Figures 6 and 7 (resp. 9 and 10) show respectively a flat and a Gouraud shading view of the semi discrete polyhedron of the radius 15 discrete sphere (resp. the mushroom).

The more constrained variant (see end of section 3) designed for polyhedra provides better results than those of figure 2. Figure 1 shows a vectorial view of the computed standard polyhedron, that have the same number of faces, edges, and vertices as the starting Euclidean polyhedron. In this case the reduction of

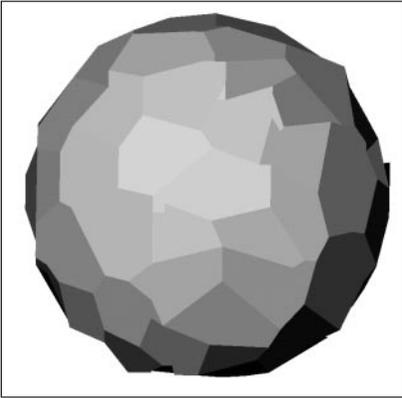


Fig. 6. Sphere (R=15) semi discrete faces with flat shading

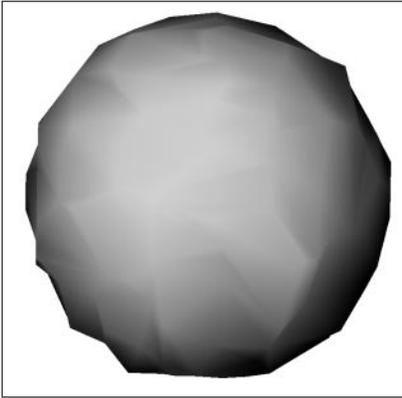


Fig. 7. Sphere (R=15) semi discrete faces with Gouraud shading



Fig. 8. Mushroom standard faces with flat shading



Fig. 9. Mushroom semi discrete faces with flat shading

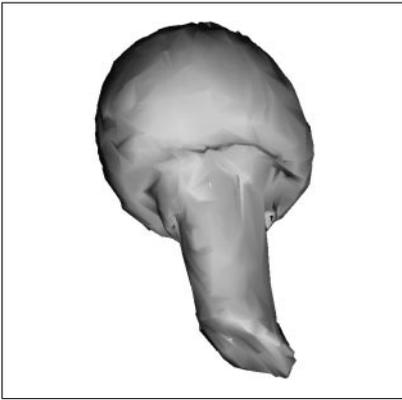


Fig. 10. Mushroom semi discrete faces with Gouraud shading

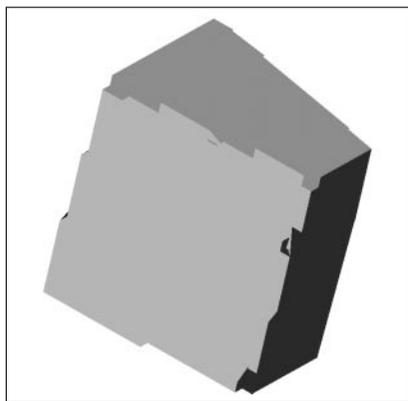


Fig. 11. Hexahedron standard faces with flat shading

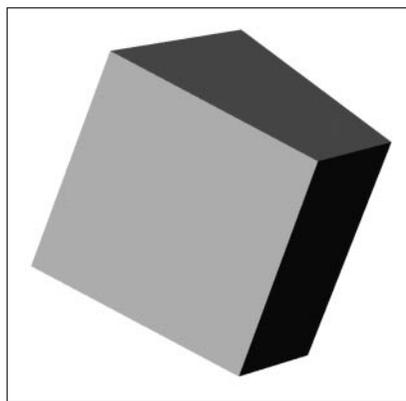


Fig. 12. Hexahedron semi discrete faces with flat shading

complexity of the polyhedrization is maximal. Figure 11 shows a flat shading view of this standard polyhedron, and figure 12 shows a flat shading view of the associated semi discrete polyhedron; this last result can be said very satisfactory.

6 Conclusion

We have defined a discrete standard polyhedron, a new kind of polyhedron, with faces that are planar for standard arithmetic planes. We have designed and experimented a polyhedrization algorithm that produces a discrete standard polyhedron, starting from the surfel boundary of a voxel object. Two properties of this process are very interesting: Reduction of space complexity, and reversibility. Examples of application to the visualization of a voxel object have been given. All these results are quite encouraging for further progress.

Now, a lot of future work has to be done:

- Theoretical and experimental study of the space complexity reduction;
- Design of reversion algorithms;
- Extension to the polyhedrization (encoding) of a discrete scene, and of all the objects of a 3D image (as was done for 2D images in [11]);
- Exploration of the numerous variants of the algorithm (varying the arbitrary decisions, defining more or less supplementary constraints), and their consequences on the statistics of the faces, and on their geometric properties;
- Full development of the visualization of discrete objects;
- Development of other applications, especially the use of discrete standard polyhedra like the use of Euclidean polyhedra in the field of topology based geometric modeling and in CAD, their use in discrete modelers, in medical imaging ...

A part of this future work is in progress.

Acknowledgements.

Thanks are due to the groups of discrete geometry and of topology based geometric modeling of Strasbourg and of Poitiers, and especially to Philippe Borianne (CIRAD Corpus 2000 Project), for numerous interesting discussions. The mushroom data are due to Philippe Borianne. Several improvements of this paper are due to comments of referees.

References

1. Borianne, Ph., Françon, J.: Reversible polyhedrization of discrete volumes. 4th DGCI Conference, Grenoble, September 19-21, 1994.
2. Debled, I., Reveillès, J.-P.: Un algorithme linéaire de polygonalisation de courbes discrètes. 2nd DGCI Conference, Grenoble, 17-18 September, 1992.
3. Debled, I., Reveillès, J.-P.: A new approach to digital planes. Vision Geometry III, Boston, 31 October - 4 November 1994.
4. Debled, I., Reveillès, J.-P.: An incremental algorithm for digital plane recognition. 4th DGCI Conference, Grenoble, September 19-21, 1994.
5. Debled, I.: Etude et reconnaissance de droites et plans discrets. Thèse de l'Université Louis Pasteur, Strasbourg, December 1995.
6. Françon, J.: Discrete combinatorial surfaces. Graphical Models and Image Processing, 57, pp. 20-26, 1995.
7. Françon, J.: Arithmetic planes and combinatorial manifolds. 5th DGCI Conference, Clermont-Ferrand, September 25-27, 1995.
8. Françon, J.: Sur la topologie d'un plan arithmétique. Theor. Comp. Sc. 156, pp. 159-176, 1996.
9. Françon, J., Schramm, J.-M., Tajine, M.: Recognizing arithmetic straight lines and planes. Proc. Int. Workshop on Discrete Geometry for Computer Imagery (6th DGCI Conference), LNCS 1176, Springer, 1996.
10. Kovalevsky, V.: Finite topology as applied to image processing. CVGIP 46, pp. 141-161, 1989.
11. Kovalevsky, V.: Applications of digital straight segments to economical image encoding. Proc. Int. Workshop on Discrete Geometry for Computer Imagery (7th DGCI Conference), LNCS 1347, Springer, 1997.
12. Papier, L.: Polyédrisation et visualisation du bord des objets voxels. Thèse de l'Université Louis Pasteur, Strasbourg (in preparation).
13. Papier, L., Françon, J.: Evaluation de la normale au bord d'un objet discret 3D. Revue internationale de CFAO et d'Infographie, to appear.
14. Reveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. Thèse d'état soutenue à l'Université Louis Pasteur, Strasbourg, December 1991.
15. Vialard, A.: Geometrical parameters extraction from discrete pathes. Proc. Int. Workshop on Discrete Geometry for Computer Imagery (6th DGCI Conference), LNCS 1176, Springer, 1996.
16. Vialard, A.: Chemins euclidiens: Un modèle de représentation des contours discrets. Thèse de l'Université de Bordeaux I, December 1996.