From digital plane Segmentation to Polyhedral representation

Isabelle Sivignon[1] and David Coeurjolly[2]

[1] Laboratoire LIS
961, rue de la Houille Blanche - BP46
38402 St Martin d'Hères Cedex
isabelle.sivignon@lis.inpg.fr
[2] Laboratoire ERIC - Université Lumière Lyon 2
5, av. Pierre Mendès-France
69676 BRON Cedex
david.coeurjolly@eric.univ-lyon2.fr

Abstract. Many applications, manipulation or just visualization of discrete volumes are time consuming problems. The general idea to solve these difficulties is to transform, in a reversible way, those volumes into Euclidean polyhedra. A first step of this process consists in a Digital Plane Segmentation of the discrete object's surface. In this paper, we present an algorithm to construct an optimal, in the number of vertices, discrete volume polyhedral representation (*i.e.* vertices and faces adjacencies).

1 Introduction

3D discrete volumes are more and more used especially in the medical area as they are the result of MRI (Magnetic Resonance Imagery) 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 that is needed is very huge which is a problem to get a fluent interactive visualization ; the facet structure (voxels's faces) of the discrete object induces many problems to get a nice visualization that is necessary for medicines.

The general idea to solve those problems is to transform, in a reversible way, those discrete volumes into Euclidean polyhedra. The first step to achieve this transformation is to split the surface of any discrete object into pieces of discrete planes. Processing this step requires an algorithm to recognize pieces of discrete planes and an algorithm to apply the former on a discrete surface. Many discrete plane recognition algorithms have been proposed using convex hull reconstruction [1, 2], linear system resolution [3–5], arithmetical plane definition [6], or parameter space transformation [7]. Although discrete plane recognition is a solved problem, the application of such algorithms on a surface is still an open problem as it lacks a criterion to judge if one solution is better than another.

We can nevertheless mention many works that propose different methods for the voxels tracking order and the plane germs [8–11].

After those operations, we get a digital plane segmentation of the surface, but not a polyhedrization. In order to obtain such polyhedrization, each piece of this segmentation have to be converted into a polygon, and therefore, it is necessary to determine where edges and vertices have to be placed.

In this paper, we propose a method to determine where an edge or a vertex has to be placed, using technics that are similar to those used in [12] for uncertain geometry. From a given digital plane segmentation, we build an adjacency graph, from which we deduce the object vertices, and then edges.

2 Digital Plane Segmentation

In this part, we have a insight in the basic definitions and methods that are used to construct the digital plane segmentation of a digital surface.

2.1 Digital Plane definition

A digital plane (DP for short) is the result of the digitization of an Euclidean plane with a given digitization scheme. Many definitions have been proposed to characterize those planes using supporting planes [13] and then arithmetic geometry to generalize those definitions to hyperplanes in n-dimensional spaces [14].

Definition 1. *A* **digital plane** *of normal vector* (a, b, c)*, translation parameter* r *and arithmetical thickness* $\omega \in \mathbb{N}$ *is defined as the set of points* $M(x, y, z) \in \mathbb{Z}^3$ *satisfying the double inequality:*

$$0 \leq ax + by + cz + r < \omega$$

where a, b, c *are not null all together and satisfy* $gcd(a, b, c) = 1$.

The thickness ω determines the connectivity of the digital plane. For instance, if $\omega = \max(|a|, |b|, |c|)$ we get the thinnest plane without holes, called *naive plane* (cf Figure 1), and for $\omega = |a| + |b| + |c|$ the thinnest 6-connected plane, called *standard plane*. Naive and standard planes are the most used for the construction of a DP segmentation: indeed, as naive planes are thin, doing a naive plane recognition on a object only requires the voxels that are the interface between the object and the background. Using standard plane can also be easy if the recognition is done on the surface voxels pointels.

2.2 Digital Plane Recognition and Segmentation

The problem of DP recognition can be formulated as following: let V be a set of voxels, does there exist real values $(\alpha, \beta, \gamma, \delta)$ such that the digitization of the plane $(\alpha x + \beta y + \gamma z + \delta = 0)$ contains all the voxels of V ?

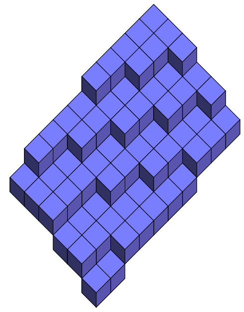


Fig. 1. A discrete naive plane with parameters (6,13,27,0)

From this problem, another one can be derived: let V be a set of voxels, what is the set of parameters $(\alpha, \beta, \gamma, \delta)$ such that the digitization of the plane $(\alpha x + \beta y + \gamma z + \delta = 0)$ contains all the voxels of V ?

With the first formulation, the only information wanted is the voxels coplanarity, whereas the second one requires the search of all the solutions. Many algorithms have been designed to solve those two problems [1–7] using the different DP definitions we presented in the previous section. Most of those algorithms only provide a coplanarity test, but the last two ones use the parameter space to get the whole set of solutions. For the problem we are dealing with, we only need a coplanarity test for the voxels, and therefore any of the existing algorithms can be used.

The second part is to apply this algorithm on a discrete surface in order to get a segmentation of the object surface into DP segments. In the following, the surface of an object is the set of object voxels that share a surfel with the background. We recall that a surfel is the square surface element present on a voxel face. The aim of this operation is to label each voxel with the numbers of the DP it belongs to. Many strategies are possible to do so. Indeed, to apply this algorithm, we have to choose the DP germs and the tracking order of the surface voxels, which can be done in many different ways. Different strategies have been proposed [8–11] but the comparison of the results can only be done visually on simple objects (cube, chamfer cube, etc), as no other criterion has been proposed for the moment.

The aim of this work is not to find another polyhedrization strategy from the discrete object but to find an optimal polyhedrization for a given segmentation. So, any segmentation that gives as a result a labelling of each surface voxels is appropriated. From this voxel labelling, we derive a pointel labelling leading to the adjacency graph presented in the next section.

3 Polyhedral representation

In this section, we present a method to determine where an edge or a vertex should be placed for a given DP segmentation, minimizing the number of vertices. This method is composed of five steps: first the construction of the adjacency graph, next a minimum clique covering, then a reduction of those cliques and a minimum cycle covering on this reduced graph, and finally a second reduction to get the polyhedral representation.

3.1 Digital plane Segmentation adjacency graph

First of all, we define the *Euclidean polyhedron adjacency graph*:

Definition 2. *Let P be an Euclidean polyhedron. Then we define the* **adjacency graph** *of P as follows:*

- *vertices are labelled with (i, j, k) where i, j, k are three planes indices, and those planes have at least one point in common ;*
- *an edge is drawn between two vertices if the vertices' labels have two planes in common.*

In the same way, we can define such a structure from a discrete volume. The construction of this graph is then based on a DP segmentation. As an input data, we have a labelling of all the voxels with the DP labels they belong to. We introduce the notion of pointels as one of the 8 vertices of a voxel. To build the graph, we derive this voxel labelling into a pointel labelling as follows:
Let p be a pointel and $S = \{\text{voxel } V \mid p \subset V\}$ be the set of voxels p belongs to. We note $L(v)$ v's label. Then $L(p) = \bigcup_{V \in S} L(V)$.

With this operation, at each pointel, we get a label corresponding to the planes it belongs to. Then the adjacency graph is defined as following:

Definition 3. *Let L be a labelling of all pointels of the discrete object surface S using a DP segmentation process. The* **adjacency graph** *of S is defined as follows:*

- *vertices are labelled with (i, j, k) where i, j, k are three plane indices such that there exists a pointel p which label contains i, j and k ;*
- *an edge is drawn between two vertices if the vertices' labels have two plane numbers in common.*

An example of such a graph is proposed in Figure 4.

In this graph, a vertex is possibly a vertex of the polyhedron as it is common to three different planes, so we can talk about "vertex" either for the graph or for the polyhedron. But we have to keep in mind that a vertex of the graph can be defined by many pointels of the object (if many pointels share same three planes). To sum up, a vertex of the graph is in fact a set of pointels of the object.

In the graph, two vertices are linked by an edge when they have two planes in common, hence the corresponding vertices in the polyhedron will either be the two extremities of an edge or be confounded. Studying this graph and grouping the vertices, we will extract the polyhedron's vertices that are possibly the intersection of more than three planes. Consequently, those vertices will be defined by a set of pointels of the object surface. Moreover, this grouping according given structures will allow us to find the minimum number of vertices for a given DP segmentation.

The definition of the structures to look for in the DP segmentation adjacency graph is done studying the Euclidean polyhedron adjacency graph. In fact, in this graph we can easily extract the different structures that can correspond to the polyhedron vertices. Those structures are general to any polyhedron. Then, looking for such structures in the DP segmentation adjacency graph enables to find the reconstructed polyhedron vertices.

3.2 Vertices and edges extraction

According to the adjacency graph construction, we first define the structure in this graph that corresponds to vertices using classical graph definitions (see [15] for an introduction to graph theory).

As we saw in the previous section, any vertex of the graph is a candidate to be a vertex of the polyhedron as it represents the intersection of three planes. A straight forward extension of this fact is the definition given below:

Definition 4. *Let G be the adjacency graph of a DP segmentation, a* **candidate vertex** *of the polyhedron is a clique (complete subgraph) of G.*

At this point, there is no equivalence between cliques and polyhedron vertices. The link between those two objects is not straight forward, as shown by the two following properties (see illustration Figure 2).

Proposition 1. *Let C be a clique of size n. Let p be the number of planes contained in the clique's vertices labels. Then, C is a polyhedron vertex or "valid clique" if and only if $n = \binom{p}{3}$.*

Proof. In fact, suppose that a clique does not satisfy this property. Then this means that $n < \binom{p}{3}$. So there exist 3 planes such that they do not share any pointel. Those three planes should not be part of the same vertex.

Proposition 2. *A vertex of the polyhedron that is the intersection of more than 4 planes is not represented by a clique in the adjacency graph.*

Proof. Let v be a vertex of the polyhedron corresponding to the intersection of p planes, with $p > 4$. Then in the corresponding graph, there are $\binom{p}{3}$ vertices that represent all the possible triplets with those p planes. As $p > 4$, there exist two vertices v_1 and v_2 which labels do not have two planes in common. Therefore, the edge (v_1, v_2) does not exist in the graph. Consequently, this graph cannot be a clique.

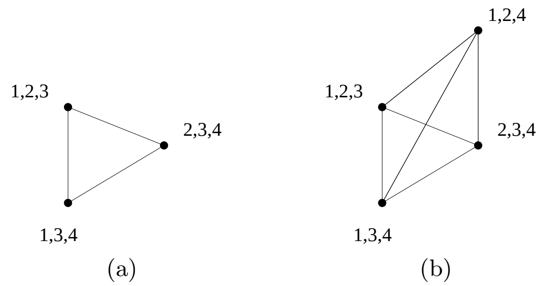


Fig. 2. Illustration of Proposition 1: (a) a non valid clique with 4 different planes but only $3 \neq \binom{4}{3}$ vertices ; (b) a valid clique with 4 planes and $4 = \binom{4}{3}$ vertices.

As a consequence, a clique search in the graph will never find correct cliques with more than 4 vertices. Then, with such an algorithm, we never reconstruct some vertices which are adjacent to more than 4 planes. We have to characterize those vertices adjacent to more than 4 planes in the graph in order to find an algorithm to detect them.

Proposition 3. *Let v be a polyhedron's vertex adjacent to p planes, with $p > 4$. Then the adjacency graph related to this vertex contains p cliques of size 4 that are not disjoint and such that if we define a new adjacency relation as "having one vertex in common", those p cliques form a cycle.*

Proof. Let v be such a vertex, and note $(0, 1, 2, 3, 4, 5, \ldots p)$ the planes it belongs to. In the adjacency graph, all the quadruples $(n,\ n+1 \pmod p,\ n+2 \pmod p,\ n+3 \pmod p)$, $0 \leq n \leq p$ are cliques of size 4 with vertices $(n,\ n+1 \pmod p,\ n+2 \pmod p)$, $(n+1 \pmod p, n+2 \pmod p, n+3 \pmod p)$, $(n+2 \pmod p, n+3 \pmod p, n)$ and $(n+3 \pmod p, n, n+1 \pmod p)$. Two consecutive quadruples have three planes in common, so the corresponding cliques have one vertex in common in the graph. If we introduce the adjacency relation exposed in the proposition, and reduce each clique into one point, we then get a cycle of size p.

The cycles containing all the quadruples given above are called *valid cycles*. This proposition gives a characterization of the vertices that are adjacent to more than 4 planes. With this characterization, we are now able to propose an algorithm to determine vertices and edges.

Algorithm 1 extracts from the adjacency graph the structures that correspond to the polyhedron's vertices. Let us see each step in details.

The first step extracts from the graph the basic element of any candidate vertex, *i.e.* the valid cliques of size 4. The result of this operation is uniquely determined by the adjacency graph, and so by the DP segmentation:

Proposition 4. *The result of the clique extraction done in the first step of the algorithm* MINIMUM_VERTICES *is uniquely determined by the initial DP segmentation.*

Algorithm 1 Minimum number of vertices Polyhedral representation

MINIMUM_VERTICES(G)

1: Extract the cliques of size 4 in the adjacency graph, keeping only the valid cliques. Two cliques can have a common vertex, but no common edge.

2: Reduce the graph a first time:

   – vertices are the cliques ;
   – two types of edges that correspond to two different adjacency relations:
     • the former edges remain in the reduced graph after contraction of the cliques (edges of type 1).
     • a new adjacency relation is introduced: two vertices are linked by an edge when the corresponding cliques have a common vertex (edges of type 2).

3: In this graph, extract the minimum number of valid cycles using only edges of type 2.

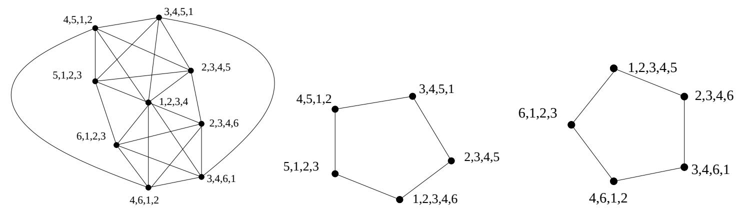
4: Reduce the graph a second time:

   – vertices are the cycles ;
   – edges of type 1 remain.

Proof. Suppose that there exists two different cliques covers. This means that a choice between two distinct cliques C_1 and C_2 is done during the clique covering algorithm. Such choice implies that the clique C_1 must invalidate the clique C_2, or conversely. In other words, since two cliques of the decomposition can share a vertex, C_1 and C_2 must share at least an edge. However, according to the definition of valid cliques, two valid cliques cannot share an edge. Therefore, C_1 or C_2 is not valid and the clique covering algorithm has no choice. Hence we prove the proposition.

During the second step of the algorithm, a first reduction is done: each clique found during step 1 is reduced to one new vertex of the graph. During this reduction, the labels of all the vertices of each clique are given to the representing vertex. An edge of type 1 is drawn between two vertices when the cliques corresponding to those vertices were linked by at least one edge. Moreover, new edges are added: edges of type 2 are the ones that take part in the cycles introduced in Proposition 3: they represent the fact that two cliques that share three planes should be part of the same vertex.

After the reduction of step 2, the graph with edges of type 2 should be disconnected, and each connected component should contain a unique cycle (so a vertex, cf Proposition 3). We note that if one connected component does not contain a cycle or contains many cycles, after reduction we get many different vertices that can share 3 planes. That is in contradiction with the definition of a vertex. This case is due to the discrete structure of the objects and an example of such a case is shown in Figure 6. The cycle covering extracts disjoint cycles and the decomposition we get is not unique, once more because of the discrete structure of the initial object (see Figure 3 for an example).



(a) (b) (c)

Fig. 3. The cycle extraction is not unique: (a) graph before cycle extraction; (b) and (c) graph after cycle extraction and reduction with two different strategies

In the last step, a second reduction is done to get to the final graph representation of the polyhedron's structure. Edges of type 1 are the ones that represent the fact that two vertices share two planes. So after the second reduction done in step 4, the remaining edges correspond to the polyhedron's edges.

The final graph obtained after this algorithm is a description of the corresponding polyhedron: the graph's vertices are the polyhedron's vertices, and the graph's edges are the polyhedron's edges. Moreover, the vertices labels provide the planes adjacent to each vertex.

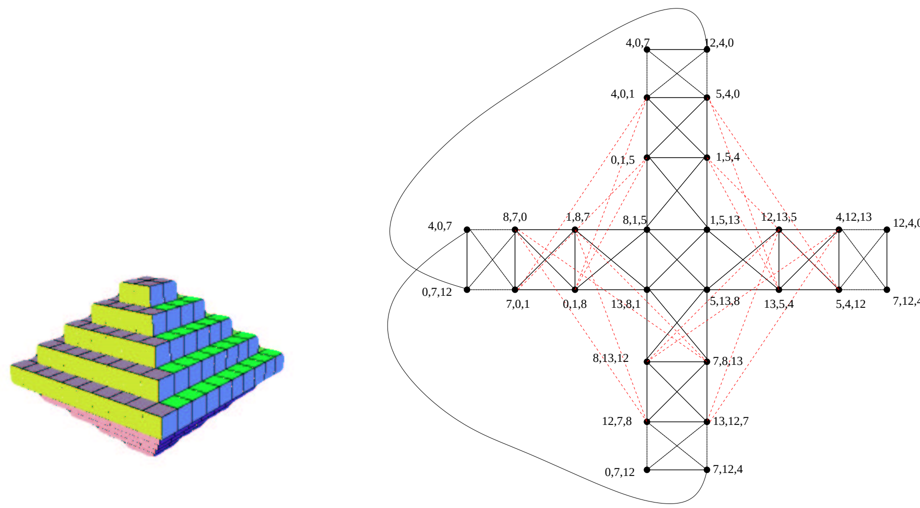
If we apply this algorithm to an Euclidean polyhedron adjacency graph, after the second step we get many connected components, each one corresponding to one polyhedron's vertex, and containing exactly one valid cycle. Hence, from a Euclidean adjacency graph, after reduction of the cycles, there is a bijection between graph and polyhedron vertices and edges. If we consider now a discrete surface, since all reduction steps consider the minimum number of cliques and cycles, no adjacency graph with less vertices than the final graph can be built. Hence, the obtained polyhedron is optimal, according to the number of vertices.

In a computational cost point of view, both the minimum clique covering of step 1 and the minimum vertex disjoint cycle cover of step 3 are NP-complete in the general case [15, 16]. In other words, no exact solution can be found in a polynomial time in the number of vertices in G. However, these two steps are equivalent to classical graph coloring problems [15] and thus, many approximated solutions can be found using efficient algorithms. For example, polynomial in time algorithm exists that approximates the solution of the minimum clique covering problem at a factor 2 of the optimal solution [17].

4 Examples

In this section, we present two examples of the algorithm different steps. The two objects studied are synthesized objects.

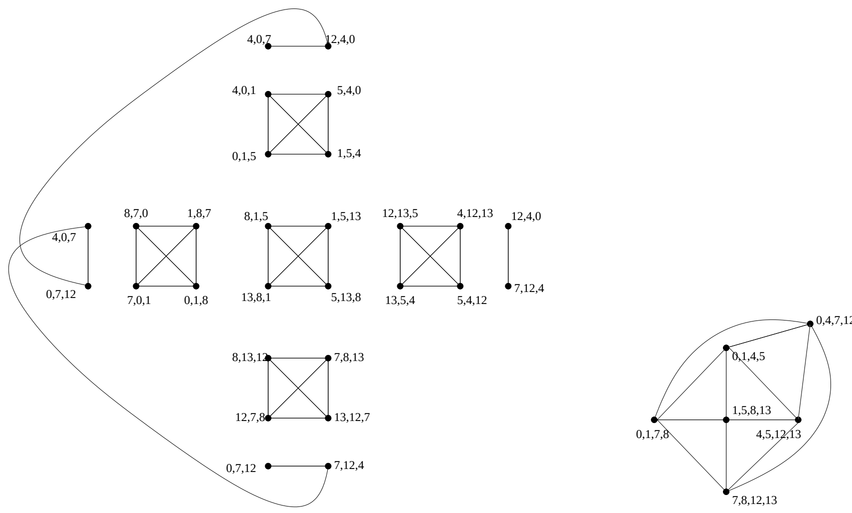
The first object represented on Figure 4 is composed of two pyramids with square basis that have a common basis. The DP segmentation finds 8 pieces of discrete planes, which fit with the object's 8 faces. Figure 4 (b) represents the adjacency graph corresponding to this DP segmentation. In the Figure 5 (a), the first step of the algorithm has been processed, and only the valid cliques found are drawn. The last graph (Figure 5 (b)) is the one obtained after step 2, 3 and 4. Indeed, as the cliques have no vertex in common, there is no edge of type 2 in the reduced graph. Then step 3 and 4 are useless, and the reduced graph is the final graph. This graph is composed of 6 vertices and 12 edges which also are the polyhedron's vertices and edges. The graph vertices's labels contain the information about the planes adjacent to each polyhedron's vertex.



(a) (b)

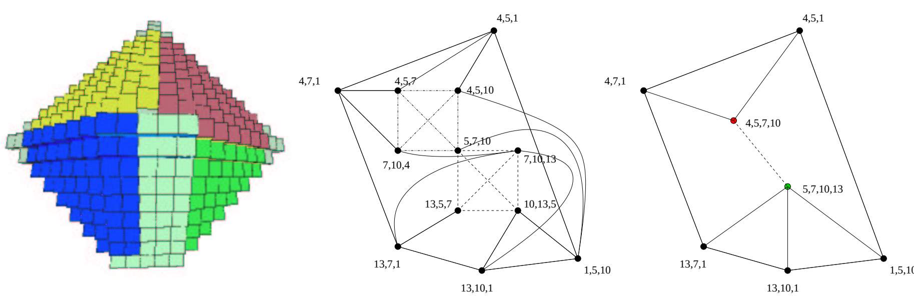
Fig. 4. Example of the construction of the graph: (a) DP segmentation of the object ; (b) corresponding adjacency graph

The second example is a synthesized pyramid with five lateral faces. The DP segmentation retrieve those 5 faces and the basis as shown on the Figure (one color for each plane). The first graph (Figure 6 (b)) is the adjacency graph. Step 1 determines 7 cliques on this graph: 2 cliques of size 4 and 5 isolated points. With the reduction done in step 2, we get the graph drawn on Figure 6 (c): there is only one edge of type 2 between the two vertices obtained after reduction of the two cliques of size 4. This edge is not a valid cycle, so step 3 doesn't modify the graph. Finally, we get a graph with 7 vertices and 11 edges. As we already noticed, this graph highlights some problems that are tightly linked to the discrete structure of the object: indeed, in this graph, we get 2 vertices that have three planes in common in their labels. This type of problem will have to be solved for the embedding in the Euclidean space.



(a) (b)

Fig. 5. Vertices and edges extraction: (a) after the minimum clique covering and sorting out ; (b) reduction of the cliques to detect edges



(a) (b) (c)

Fig. 6. Example of the construction of the graph: (a) DP segmentation of the object ; (b) corresponding adjacency graph and (c) final polyhedral representation

5 Conclusion

In this paper, we have presented both theoretical and practical aspects of the discrete volume polyhedrization problem. The objective was to define and locate vertices of the optimal discrete object polyhedral representation. The optimality criterion we have defined is based on the minimum number of vertices.

Hence, given a digital plane segmentation of a discrete volume surface, we have presented an algorithm based on classical graph theory tools in order to extract from this segmentation, an optimal polyhedral representation, *i.e.* vertices and faces adjacencies.

The next step of this work consists in the embedding of this polyhedral representation into the Euclidean space in order to give a complete reversible polyhedron with minimal number of vertices associated to the discrete object.

References

1. Kim, C., Rosenfeld, A.: Convex digital solids. IEEE Trans. on Pattern Anal. Machine Intell. **PAMI-4** (1982) 612–618
2. Kim, C., Stojmenović, I.: On the recognition of digital planes in three dimensionnal space. Pattern Recognition Letters **32** (1991) 612–618
3. Veelaert, P.: Digital planarity of rectangular surface segments. IEEE Trans. Pattern Anal. Machine Intell. **PAMI-16** (1994) 647–653
4. Franon, J., Schramm, J.M., Tajine, M.: Recognizing arithmetic straight lines and planes. In A. Montanvert, S.M., Ubéda, S., eds.: DGCI'96. Volume 1176 of Lect. Notes of Comp. Sci., Springer Verlag (1996) 141–150
5. Buzer, L.: An incremental linear time algorithm for digital line and plane recognition using a linear incremental feasibility problem. In Braquelaire, A., Lachaud, J.O., Vialard, A., eds.: DGCI'02. Volume 2301 of Lect. Notes of Comp. Sci., Springer Verlag (2002) 372–381
6. Debled-Rennesson, I., Reveillès, J.P.: An incremental algorithm for digital plane recognition. In: DGCI'94. (1994) 207–222
7. Vittone, J.: Caractérisation et reconnaissance de droites et de plans en géométrie discrète. PhD thesis, Université Joseph Fourier, Grenoble, France (1999)
8. Borianne, P., Françon, J.: Reversible polyhedrization of discrete volumes. In: DGCI'94, Grenoble, France (1994) 157–168
9. Papier, L., Françon, J.: Polyhedrization of the boundary of a voxel object. In G. Bertrand, M.C., Perroton, L., eds.: DGCI'99. Volume 1568 of Lect. Notes of Comp. Sci., Springer Verlag (1999) 425–434
10. Chassery, J.M., Dupont, F., Sivignon, I., Vittone, J.: Recognition of digital naive planes. In: ICIAP'01 11th International Conference on Image Analysis and Processing. (2001) 662–636
11. Klette, R., Sun, H.J.: Digital planar segment based polyhedrization for surface area estimation. In Arcelli, C., Cordella, L.P., di Baja, G.S., eds.: International Workshop on Visual Form 4. Volume 2059 of Lect. Notes Comput. Sci., Springer-Verlag (2001) 356–366
12. Veelaert, P.: Concurrency of line segments in uncertain geometry. In Braquelaire, A., Lachaud, J.O., Vialard, A., eds.: DGCI'02. Volume 2301 of Lect. Notes of Comp. Sci., Springer Verlag (2002) 289–300

13. Klette, R.: 1. In: Digital Geometry - The birth of a new discipline. (2001) Retirement of A. Rosenfeld.
14. Reveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. PhD thesis, Université Louis Pasteur (1991)
15. West, D.: Introduction to Graph Theory. 2 edn. Prentice Hall (2001)
16. Thomassen, C.: On the complexity of finding a minimum cycle cover of a graph. SIAM Journal on Computing **26** (1997) 675–677
17. Hochbaum, D., ed.: Approximation algorithms for NP-hard problems. PWS Publishing Company (1997)