



Removal and contraction operations to define combinatorial pyramids: application to the design of a spatial modeler

Guillaume Damiand, Martine Dexet, Pascal Lienhardt, Eric Andres

► To cite this version:

Guillaume Damiand, Martine Dexet, Pascal Lienhardt, Eric Andres. Removal and contraction operations to define combinatorial pyramids: application to the design of a spatial modeler. Image and Vision Computing, 2005, 23 (2), pp.259-269. 10.1016/j.imavis.2004.06.016 . hal-00211776

HAL Id: hal-00211776

<https://hal.science/hal-00211776>

Submitted on 21 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Removal and contraction operations to define combinatorial pyramids: application to the design of a spatial modeler

Guillaume Damiand^{a,*} Martine Dexet-Guiard^a
Pascal Lienhardt^a Eric Andres^a

^a*IRCOM-SIC, bât SP2MI, BP 30179, 86962 Chasseneuil Cedex, France*

Abstract

Removal and contraction are basic operations for several methods conceived in order to handle irregular image pyramids, for multi-level image analysis for instance. We give the definitions of removal and contraction operations in the generalized maps framework. We propose a first experimentation of irregular pyramid as a basis for a discrete geometrical modeler that can handle both discrete and continuous representations of geometrical objects. This modeler is based on a pyramidal kernel with four coexisting levels between the discrete and the Euclidean representations. We describe how this pyramid can be constructed and updated.

Key words: removal, contraction, irregular pyramids, n-G-map, modeler, discrete reconstruction

1 Introduction

Hierarchical representations are the bases of several applications in the field of discrete imagery. Our first goal is the study of basic problems related to the definition of hierarchical structures. We study the definition of removal and contraction operations, from which hierarchical structures can be derived (cf. [35]).

* Corresponding author.

Email addresses: damiand@sic.univ-poitiers.fr (Guillaume Damiand),
dexet@sic.univ-poitiers.fr (Martine Dexet-Guiard),
lienhardt@sic.univ-poitiers.fr (Pascal Lienhardt),
andres@sic.univ-poitiers.fr (Eric Andres).

Then we study a first experimentation of these notions. A four level hierarchical structure is proposed in order to handle coherent discrete and continuous representations of geometrical objects. The other levels of the structure correspond to other representations which make a link between the previous ones, mainly a discrete analytical representation.

Many works deal with regular (cf. e.g. [20]) or irregular (cf. e.g. [38,39,30]) image pyramids for multi-level image analysis and processing (cf. also [35]). For irregular pyramids, it is necessary to handle a (topological) representation and basic operations, for instance dual graphs and removal and contraction operations for 2D images [33,34]. Similar problems about multi-level representations arise also in geometric modeling (e.g. for CAD applications, architectural or geological modeling, . . . [10,24,5]).

Our goal is to build a theoretical framework for the definition and handling of n -dimensional irregular pyramids, in order to get coherent definitions of data structures and operations for any dimension. So we study the definition of removal and contraction of i -dimensional cells within n -dimensional objects (see also [23]), in order to rigorously define the relations between two consecutive levels of a pyramid.

We choose to study the definitions of removal and contraction operations for n -dimensional generalized maps, since this notion enables us to unambiguously represent the topology of quasi-manifolds, which is a well-defined class of subdivisions [37]. Note that several models based on combinatorial maps [26,29,21] have been proposed for handling 2-dimensional [15,28] and 3-dimensional segmented or multi-level images [12,7,8,11,22]. For instance, Brun and Kropatsch revisit works about graphs [16–19] in order to define 2-dimensional combinatorial map pyramids.

We prefer to use generalized maps instead of combinatorial maps, since their algebraic definition is homogeneous and so we can provide simpler definitions of data structures and operations with generalized maps, and so more efficiency for the conception of softwares (Note that several kernels of geometric modeling softwares are based upon data structures derived from this notion). Last, we know how to deduce combinatorial maps from generalized maps, so the results presented in this paper can be extended for combinatorial maps. Precise relations between generalized and combinatorial maps, and other classical data structures are presented in [36].

We propose a first experiment of the definition of n -dimensional irregular pyramids to define a four level hierarchical structure. This pyramid is the kernel of a discrete geometrical modeler. The goal of this modeler is to handle both discrete and continuous representations of geometrical objects. Operations in the discrete or in the continuous world are quite different. Many of these oper-

ations are best performed in one or the other world (morphological operations, intersections, transformations, ...). Our goal is to perform an operation in the world where it is best defined. There are applications where both worlds can be needed. One that comes to mind is medical imaging where acquired data is discrete. However, the popularity of techniques such as the Marching cubes show that there are many applications in medical imaging where a continuous representation of discrete data is useful.

The particularity of our modeler is that for each object a discrete and a Euclidean representation coexist in the modeler. These representations must be consistent: the discretization¹ of the Euclidean representation must be equal to the discrete object. Thus, we also want to be able to update all representations in case of modification in order to ensure the coherence. For instance, each operation applied on the discrete representation (such as Boolean or morphological operations) must automatically update the Euclidean representation and vice versa.

We add some intermediate levels between the discrete and the Euclidean representations in order to divide the updating operation into several basic steps. So, we define a pyramid in order to represent all different representations in a same data structure. This is a four level pyramid: level 0 is the discrete representation (pixels); level 1 is the representation of the borders of regions; level 2 is the discrete analytical representation and level 3 is the Euclidean representation.

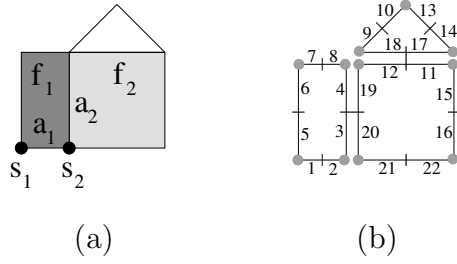
We recall in Sec. 2 the notion of generalized maps. Then we define removal of one i -dimensional cell in Sec. 3, and contraction by duality (in Sec. 4). In Sec. 5, we show that it is possible to simultaneously remove and contract several cells of any dimensions, if and only if cells are disjoint. We present in Sec. 6 the four levels pyramid, basis of the discrete geometrical modeler. Then, we present all conversion operations between consecutive levels of the pyramid in Sec. 7. Last we conclude and give some perspectives in Sec. 8.

2 Generalized Maps Recall

An n -dimensional generalized map is a set of abstract elements, called darts, and applications defined on these darts:

Definition 1 (Generalized map) *Let $n \geq 0$. A n -dimensional generalized map (or n -G-map) is $G = (B, \alpha_0, \dots, \alpha_n)$ where:*

¹ The discretization model chosen here is the discrete analytical model called *Standard Model*.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
α_0	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15	18	17	20	19	22	21
α_1	5	3	2	8	1	7	6	4	18	13	15	19	10	17	11	22	14	9	12	21	20	16
α_2	1	2	20	19	5	6	7	8	9	10	17	18	13	14	15	16	11	12	4	3	21	22

Fig. 1. (a) A 2D subdivision. (b) The corresponding 2-G-map (involutions are given explicitly in the array). Darts are represented by numbered black segments. Two darts in relation by α_0 share a little vertical segment (ex. darts 1 and 2). Two darts in relation by α_1 share a same point (ex. darts 2 and 3). Two distinct darts in relation by α_2 are parallel and close to each other (ex. darts 3 and 20); otherwise, the dart is its own image by α_2 (ex. dart 2). Dart 1 corresponds to (s_1, a_1, f_1) , dart 2 = $1\alpha_0$ corresponds to (s_2, a_1, f_1) , 3 = $2\alpha_1$ corresponds to (s_2, a_2, f_1) , and 20 = $3\alpha_2$ corresponds to (s_2, a_2, f_2) . The vertex incident to dart 2 is $\langle \alpha_1, \alpha_2 \rangle (2) = \{2, 3, 20, 21\}$, the edge incident to dart 3 is $\langle \alpha_0, \alpha_2 \rangle (3) = \{3, 4, 19, 20\}$, and the face incident to dart 9 is $\langle \alpha_0, \alpha_1 \rangle (9) = \{9, 10, 13, 14, 17, 18\}$.

- (1) B is a finite set of darts;
- (2) $\forall i, 0 \leq i \leq n, \alpha_i$ is an involution² on B ;
- (3) $\forall i, j, 0 \leq i < i + 2 \leq j \leq n, \alpha_i \alpha_j$ is an involution.

Let G be an n -G-map, and S be the corresponding subdivision. Intuitively, a dart of G corresponds to an $(n + 1)$ -tuple of cells (c_0, \dots, c_n) , where c_i is an i -dimensional cell that belongs to the boundary of c_{i+1} (cf. [14] and Fig. 1). α_i associates darts corresponding with (c_0, \dots, c_n) and (c'_0, \dots, c'_n) , where $c_j = c'_j$ for $j \neq i$, and $c_i \neq c'_i$ (α_i swaps the two i -cells that are incident to the same $(i - 1)$ and $(i + 1)$ -cells). When two darts b_1 and b_2 are such that $b_1 \alpha_i = b_2$ ($0 \leq i \leq n$), b_1 is said *i-sewn* with b_2 .

G-maps represent cells in an implicit way:

Definition 2 (i-cell) Let G be an n -G-map, b a dart and $i \in N = \{0, \dots, n\}$. The i -cell incident to b is the orbit³

² An involution f on S is a one to one mapping from S onto S such that $f = f^{-1}$.

³ Let $\{\Pi_0, \dots, \Pi_n\}$ be a set of permutations on B . The orbit of an element b relatively to this set of permutations is $\langle \Pi_0, \dots, \Pi_n \rangle (b) = \{\Phi(b), \Phi \in \langle \Pi_0, \dots, \Pi_n \rangle\}$, where $\langle \Pi_0, \dots, \Pi_n \rangle$ denotes the group of permutations generated by Π_0, \dots, Π_n .

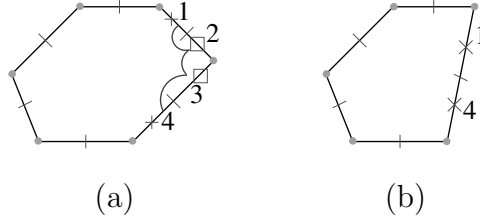


Fig. 2. 0-removal in 1D. (a) Initial 1-G-map. (b) Result. $C = \langle \alpha_1 \rangle (2) = \{2, 3\}$ (darts marked with empty squares), $C\alpha_0 = \{1, 4\} = B^S$ (darts marked with crosses). 0-removal consists in setting $1\alpha'_0 = 1(\alpha_0\alpha_1)\alpha_0 = 4 \in B^S$ and $4\alpha'_0 = 4(\alpha_0\alpha_1)\alpha_0 = 1 \in B^S$.

$$\langle \rangle_{N-\{i\}}(b) = \langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle (b)$$

Intuitively, an i -cell is the set of all darts which can be reached starting from b , by using any combination of all involutions except α_i . The set of i -cells is a partition of the darts of the G-map, for each i between 0 and n . Two cells are disjoint if their intersection is empty, i.e. when no dart is shared by the cells. More precisions about G-maps are provided in [37].

3 Removal

Intuitively and in a general way for an n -dimensional space, the removal of an i -cell consists in removing this cell and in merging its two incidents $(i+1)$ -cells: so removal can be defined for $0 \dots (n-1)$ -cells.

3.1 Dimension 1: 0-Removal

For dimension 1, only the 0-removal exists, which consists in removing a vertex and in merging its two incident edges. Let $C = \langle \alpha_1 \rangle (b)$ be a vertex, let $C\alpha_0$ be the “neighbor” darts of C for α_0 , i.e. $C\alpha_0 = \{b' \mid \exists b' \in C \text{ such that } b'\alpha_0 = b\}$, and let $B^S = C\alpha_0 - C$ be the “neighbor” darts of C for α_0 that do not belong to C (see Fig. 2). The G-map resulting from the 0-removal of C is obtained by redefining α_0 for the darts of B^S as follows: $\forall b' \in B^S, b'\alpha'_0 = b'(\alpha_0\alpha_1)^k\alpha_0$, where k is the smallest integer such that $b'(\alpha_0\alpha_1)^k\alpha_0 \in B^S$. Note that α_1 is not modified by 0-removal.

3.2 Dimension 2

There are two different removal operations (0 and 1-removal) for dimension 2.

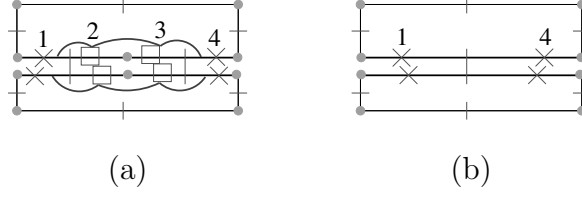


Fig. 3. 0-removal in 2D. (a) Initial 2-G-map. (b) Result. $C = \langle \alpha_1, \alpha_2 \rangle (2)$ (darts marked with empty squares), $C\alpha_0 = B^S$ (darts marked with crosses). For instance, $1\alpha'_0 = 1(\alpha_0\alpha_1)\alpha_0 = 4 \in B^S$

3.2.1 0-Removal

It consists in removing a 0-cell $C = \langle \alpha_1, \alpha_2 \rangle (b)$. Let $B^S = C\alpha_0 - C$ ($C\alpha_0$ is defined as above). This operation can be applied only if the following precondition is satisfied: $\forall b' \in C, b'\alpha_1\alpha_2 = b'\alpha_2\alpha_1$. This constraint corresponds, in the general case, to the fact that the degree of the vertex is equal to 2 (2 edges are incident to the vertex). If this constraint is not satisfied, we do not know how to join the cells incident to C , and it is then impossible to define the removal in a simple way. [27] proposes a generalization of this operation, but it is complex and cannot be used for an automatic process, in particular in automatic image processing.

The G-map resulting from 0-removal is obtained by redefining α_0 for the darts of B^S as follows: $\forall b' \in B^S, b'\alpha'_0 = b'(\alpha_0\alpha_1)^k\alpha_0$ where k is the smallest integer such that $b'(\alpha_0\alpha_1)^k\alpha_0 \in B^S$. Note that this redefinition of α_0 is the same as for dimension 1 but concerns different darts, since it is a 0-cell within a 2D object: cf. Fig. 3 (intuitively, in the general case, this operation consists in applying twice the 0-removal defined for dimension 1).

3.2.2 1-Removal

It consists in removing a 1-cell $C = \langle \alpha_0, \alpha_2 \rangle (b)$. This can be achieved without any precondition. Let $B^S = C\alpha_1 - C$. The resulting G-map is obtained by redefining α_1 for the darts of B^S as follows: $\forall b' \in B^S, b'\alpha'_1 = b'(\alpha_1\alpha_2)^k\alpha_1$, where k is the smallest integer such that $b'(\alpha_1\alpha_2)^k\alpha_1 \in B^S$. Examples of 1-removal are presented in Fig. 4, and 5. For this last example, $k = 2$ since the removed edge is incident twice to the same vertex.

3.3 Dimension n

The general definition of i -cell removal for an n -dimensional G-map is an obvious extension of the previous cases. Let C be an i -cell to remove; when

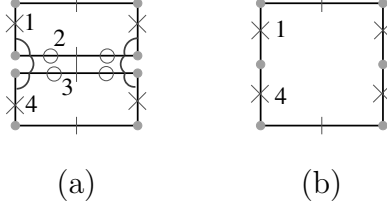


Fig. 4. 1-removal in 2D in the general case. Darts of the edge to remove are marked with circles. (a) Initial 2-G-map. (b) Result.

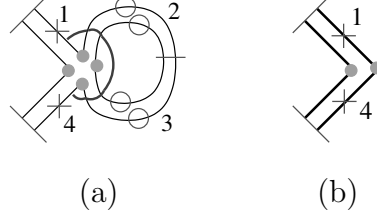


Fig. 5. 1-removal in 2D of a loop. (a) Initial 2-G-map. (b) Result. For instance, $1\alpha'_1 = 1(\alpha_1\alpha_2)(\alpha_1\alpha_2)\alpha_1 = 4 \in B^S$ (since $1(\alpha_1\alpha_2)\alpha_1 \notin B^S$, this dart belongs either to C and to $C\alpha_1$).

$i < n - 1$ the operation can be applied only when, informally⁴ the degree of C is equal to 2 (a vertex incident to exactly two edges or an edge incident to two faces or a face incident to two volumes...). The i -removal consists then in redefining α_i for the darts of $B^S = C\alpha_i - C$ in the following way: $b'\alpha'_i = b'(\alpha_i\alpha_{i+1})^k\alpha_i$, where k is the smallest integer such that $b'(\alpha_i\alpha_{i+1})^k\alpha_i \in B^S$. We obtain so the general definition of the i -removal operation:

Definition 3 (i-cell removal) Let $G = (B, \alpha_0, \dots, \alpha_n)$ be an n -G-map, $i \in \{0, \dots, n-1\}$ and $C = \langle \rangle_{N-\{i\}}$ (b) be an i -cell, such that: $\forall b' \in C, b'\alpha_{i+1}\alpha_{i+2} = b'\alpha_{i+2}\alpha_{i+1}$. Let $B^S = C\alpha_i - C$, the set of darts i -sewn to C that do not belong to C . The n -G-map resulting from the removal of this i -cell is $G' = (B', \alpha'_0, \dots, \alpha'_n)$ defined by:

- $B' = B - C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \alpha'_j = \alpha_j|_{B'}$; ⁵
- $\forall b' \in B' - B^S, b'\alpha'_i = b'\alpha_i$;
- $\forall b' \in B^S, b'\alpha'_i = b'(\alpha_i\alpha_{i+1})^k\alpha_i$,
where k is the smallest integer such that $b'(\alpha_i\alpha_{i+1})^k\alpha_i \in B^S$.

Note that G' can contain only one n -cell, and may even be empty if G contains only one i -cell.

4 Contraction

Informally, i -contraction consists in contracting an i -cell into an $(i - 1)$ -cell. Contraction is the dual of the removal operation. Informally, the dual of a subdivision is a subdivision of the same space, in which an $(n - i)$ -cell is associated with each initial i -cell, and incidence relations are kept. A nice

⁴ The formal precondition is: $\forall b' \in C, b'\alpha_{i+1}\alpha_{i+2} = b'\alpha_{i+2}\alpha_{i+1}$. Note that if $i = n - 1$ this condition doesn't apply and we can always remove any $(n - 1)$ -dimensional cell.

⁵ α'_j is equal to α_j restricted to B' , i.e. $\forall b \in B', b\alpha'_i = b\alpha_i$

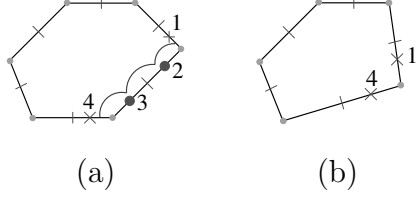


Fig. 6. 1-contraction in 1D.
Darts of the edge to contract
are marked with black disks.
(a) Initial 2-G-map. (b) Result.

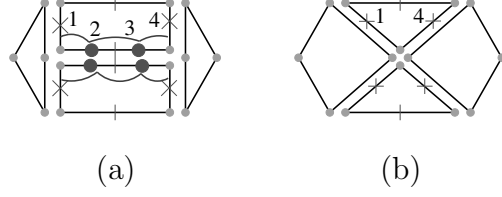


Fig. 7. 1-contraction in 2D. (a) Initial
2-G-map. (b) Result.

property of G-maps is the fact that the dual G-map of $G = (B, \alpha_0, \dots, \alpha_n)$ is $G' = (B, \alpha_n, \dots, \alpha_0)$: we just need to reverse the involution order.

We can thus easily deduce the definition of i -contraction from the general definition of i -removal. We just have to replace '+' by '-' for indices of involutions for preconditions and operations, i.e. $\alpha_{i+1}\alpha_{i+2} \rightarrow \alpha_{i-1}\alpha_{i-2}$ and $\alpha_i\alpha_{i+1} \rightarrow \alpha_i\alpha_{i-1}$. (see two examples of contraction in Fig. 6 and 7).

Definition 4 (i-cell contraction) *Let $G = (B, \alpha_0, \dots, \alpha_n)$ be an n -G-map, $i \in \{1, \dots, n\}$ and $C = \langle \rangle_{N-\{i\}}$ (b) be an i -cell, such that⁶: $\forall b' \in C, b'\alpha_{i-1}\alpha_{i-2} = b'\alpha_{i-2}\alpha_{i-1}$. Let $B^S = C\alpha_i - C$, the set of darts i -sewn to C that do not belong to C . The n -G-map resulting from the contraction of this i -cell is $G' = (B', \alpha'_0, \dots, \alpha'_n)$ defined by:*

- $B' = B - C$;
- $\forall j \in \{0, \dots, n\} - \{i\}, \alpha'_j = \alpha_j|_{B'}$;
- $\forall b' \in B' - B^S, b'\alpha'_i = b'\alpha_i$;
- $\forall b' \in B^S, b'\alpha'_i = b'(\alpha_i\alpha_{i-1})^k\alpha_i$,
where k is the smallest integer such that $b'(\alpha_i\alpha_{i-1})^k\alpha_i \in B^S$.

5 Generalisations

Previous definitions enable us to remove or to contract a single cell. For some applications, it could be more efficient to simultaneously apply several operations. Concretely, let G be an n -G-map. Assume that each dart belonging to a removed or contracted cell is marked with the dimension and type of the corresponding operation. Operations can be simultaneously applied if and only if:

- the cells are disjoint (involving that a dart has at most a unique mark and

⁶ Note that this condition doesn't apply for $i = 1$, so we can always contract any edge.

- thus that there is no ambiguity when redefining α_i for each $i, 0 \leq i \leq n$);
- the preconditions of the corresponding operations are satisfied.

Indeed, when a precondition is satisfied before a set of operations, this precondition is still satisfied after each step of the operation (the converse is obviously false), because cells are disjointed. This allows us to apply simultaneously or successively a set of operations and to obtain the same result.

We now present this generalization in several steps. First, we show that it is possible to simultaneously perform removals (resp. contractions) of several i -cells for a given i ($0 \leq i \leq n$).

Generalisation 1 *We can easily prove that the previous definition of removal (resp. contraction) stands for the removal (resp. contraction) of a set of cells of same dimension i . The (possible) precondition of the initial operation has to be satisfied for each cell (cf. Fig. 8(a)).*

Moreover, removing (resp. contracting) simultaneously several i -cells or applying successively, in any order, the initial operation for any removed cell, produce the same result. The main idea of the proof is: each α_i redefinition consists in: $\forall b \in B^{Si}, b\alpha'_i = b(\alpha_i\alpha_{i+1})^k\alpha_i$. The darts of this path can be partitioned depending on the removed cells they belong to, i.e. $b\alpha'_i = b(\alpha_i\alpha_{i+1})^{k_1}(\alpha_i\alpha_{i+1})^{k_2} \dots (\alpha_i\alpha_{i+1})^{k_p}\alpha_i$. Each subpath corresponds to a single removal and so order is not important, and each removal does not depend on other operations.

We retrieve here the notion of *connecting walk* of Brun and Kropatsch [17]. Darts of B^{Si} are surviving darts and α'_i put in relation two darts of B^{Si} by passing through some non-surviving darts (darts crossed by $(\alpha_i\alpha_{i+1})^k$).

Generalisation 2 *The previous generalization can be directly extended for simultaneously removing and contracting cells of same dimension i . A cell is either removed or contracted, but not both at the same time. The (possible) precondition of the corresponding initial operation has to be satisfied for each cell (cf. Fig. 8(b)).*

More precisely, let CS^i (resp. CC^i) be a set of i -cells to remove (resp. contract), such that $CS^i \cap CC^i = \emptyset$ and such that the (possible) precondition of i -removal (resp. i -contraction) operation is satisfied for each cell of CS^i (resp. CC^i). Let $B^{Si} = (CC^i \cup CS^i)\alpha_i - (CC^i \cup CS^i)$. As before, α_i is redefined for these darts: $\forall b \in B^{Si}, b\alpha'_i = b' = b(\alpha_i\alpha_{k_1}) \dots (\alpha_i\alpha_{k_p})\alpha_i$ where p is the smallest integer such that $b' \in B^{Si}$ and $\forall j, 1 \leq j < p$, if $b_c = b(\alpha_i\alpha_{k_1}) \dots (\alpha_i\alpha_{k_{j-1}})\alpha_i \in CS^i$ then $k_j = i + 1$ else ($b_c \in CC^i$) $k_j = i - 1$.

Generalisation 3 *The previous generalization can be directly extended for the removal and/or contraction of a set of disjointed cells of any dimension. The (possible) precondition of the corresponding initial operation has to be*

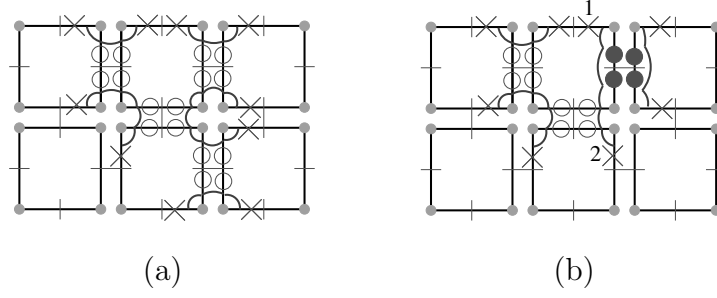


Fig. 8. (a) Simultaneous 1-removals in 2D. (b) Simultaneous 1-removals (dart marked with circles) and 1-contractions (dart marked with black disks) in 2D. For instance, $1\alpha'_1 = 1(\alpha_1\alpha_0)(\alpha_1\alpha_2)\alpha_1 = 2$ since the edge incident to $1\alpha_1$ is contracted and the edge incident to $1(\alpha_1\alpha_0)\alpha_1$ is removed.

satisfied for each cell.

This last generalization is possible because the set of cells are disjoint. If we consider two disjoint cells, there are two possibilities: the two cells have the same dimension and they can be adjacent or not; the two cells have different dimensions and they can not be incident. The first case is covered by the previous generalizations. The second one is easy to consider since the cells are not incident: so some surviving darts exist between the two cells, and α_i is redefined only for these darts. The other cells are not modified and so other preconditions are still valid when applying any subset of these operations.

6 A pyramidal data structure for a spatial modeler

6.1 Introduction

For a few years we are interested in the design of a discrete modeling software [4]. With this modeler, our goal is to create, import and handle different representations of a same geometrical object (mainly discrete and continuous representations). Modeling softwares exist that can handle discrete representations or Euclidean representations but rarely both. There are however situations where both representations could be useful. Let us give two examples to illustrate the long term goals of such a modeling system: Let us consider the general problem of computer vision as a first example. It is clear now that computer vision (for surveillance and security system applications for instance) requires low-end image processing techniques but also the construction of a high-level semantic model of the image content. There lies an important bottleneck in defining and constructing such a virtual high-level model. It would seem natural that a high-level model shouldn't be unrelated to virtual reality models as proposed in "continuous modeling". Being able to have both

a continuous “high” level representation and a discrete “low” level one could be an asset.

As a second example, let us consider the problem of augmented reality with an image containing real world objects (acquired by photography for instance) and virtual, synthetic, ones. Having both coexist in a single image means mixing geometrically the objects, mixing artificial and natural light and shadow effects, etc. Augmented reality has many applications in movie and advertising, virtual surgery, architecture, etc. One bottleneck lies in the virtual models that haven’t been conceived to handle low-end pixel or voxel information. Again, a close collaboration between both worlds should bring new insight in such problems.

In a first release of our modeler, the discrete and Euclidean forms of an object were coexisting, but were not directly linked. Then, in case of modification, each representation had to be entirely recomputed in order to keep coherent, the different representations. However, most operations are local ones; the effect of a local modification of a representation will be a local modification of the other representation. This requires the existence of connections between the different representations of an object.

In order to solve these problems, we propose the use of a pyramidal structure. Each level corresponds to a particular representation of the object: Euclidean, discrete analytical, borders of regions and pixel representations. Each level is linked with the level above and below it. This ensures the coherence between all the representations.

We present here our structure and some basic operations. For the moment, we deal only with 2D objects, because the discrete analytical reconstruction of a Euclidean object from a discrete object is still under investigation in 3D, while the 2D case has been fully solved [13]. Note that we can extend the pyramidal structure and the discretization process of a Euclidean object in any dimension.

6.2 *Our structure: a pyramid of G-maps*

Our pyramidal structure consists of four inter-dependent levels (see Fig. 9):

- Level 0: the discrete level
This level corresponds to the discrete matrix representation which is composed of pixels. A label, such as grey scale or color, is associated with each pixel.
- Level 1: the discrete border level

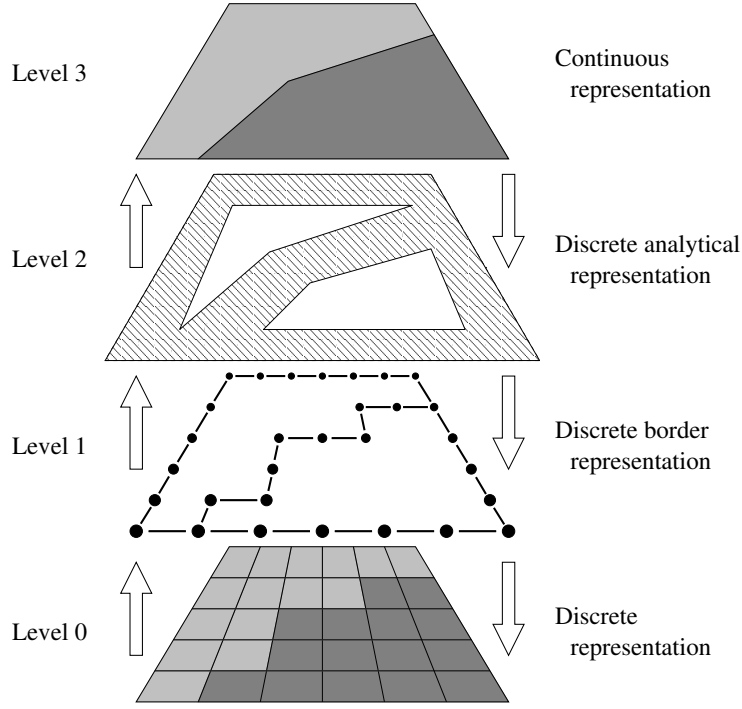


Fig. 9. The pyramidal structure.

The discrete border level corresponds to the contours obtained for each uniform labelled 4-connected region. These contours are the interpixel ones [32,31] (note that these contours are a *Cellular Complex* as defined by [1]). We can see on Fig. 9 that these contours are made up of 0-cell and 1-cell series.

- Level 2: the discrete analytical level
This level corresponds to the discrete analytical description of the region's contours. More precisely, each contour is described as a discrete analytical polygon.
- Level 3: the Euclidean level
In this level, each region is described as a Euclidean polygon.

For each level, there are structure and shape information. More precisely, each level is composed of a generalized map which represents the topological information. Specific geometrical information (called *embeddings*) are associated with specific orbits of each level in order to define the shape of the object.

New levels could be added without major upheaval in the structure. Note that the levels of the pyramid are chosen in order to obtain progressive evolution between the generalized maps of the first and the last levels: each intermediary map can be obtained from the previous one by minor modifications.

6.2.1 Links between levels

Since our goal is to quickly reflect a modification of a level of the pyramid on the other levels, we set up bidirectional connections between the different levels of the pyramid, and more precisely between darts of the levels (surviving darts). Figure 10 shows some connections between darts of the pyramid (this is only a partial representation). The advantage of these connections is that all pyramid's embeddings become accessible for all levels. The information redundancy inside the pyramid is thus limited. For example, color embeddings are only associated with level 0 map, and each level can access these information through existing connections.

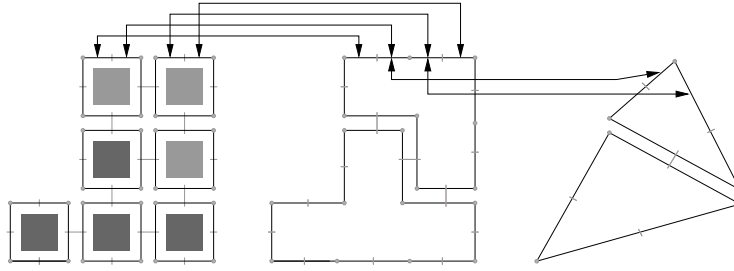


Fig. 10. Some connections between levels.

6.2.2 Level 0: the discrete level

The G-map of this level corresponds to a space subdivision into pixels (see Fig. 11(a)). Each pixel is represented by a square topological face associated with a label embedding. Moreover, integer coordinates are attached to each topological vertex.

6.2.3 Level 1: the discrete border level

The G-map of this level corresponds to a space subdivision into regions (see Fig. 11(b)), in which each topological edge corresponds to a 1-cell. This map is obtained by merging pixels of level 0 having the same label, by applying 1-removals (see Sec. 7). There are no embeddings at this level since level 0 embeddings (integer coordinates of 0-cells) can be accessed by using the links between levels of the pyramid.

6.2.4 Level 2: the discrete analytical level

The G-map of this level corresponds to a space subdivision into regions (see Fig. 11(c)), in which each topological edge corresponds to a discrete analytical segment. The discretization model used here is the discrete analytical model

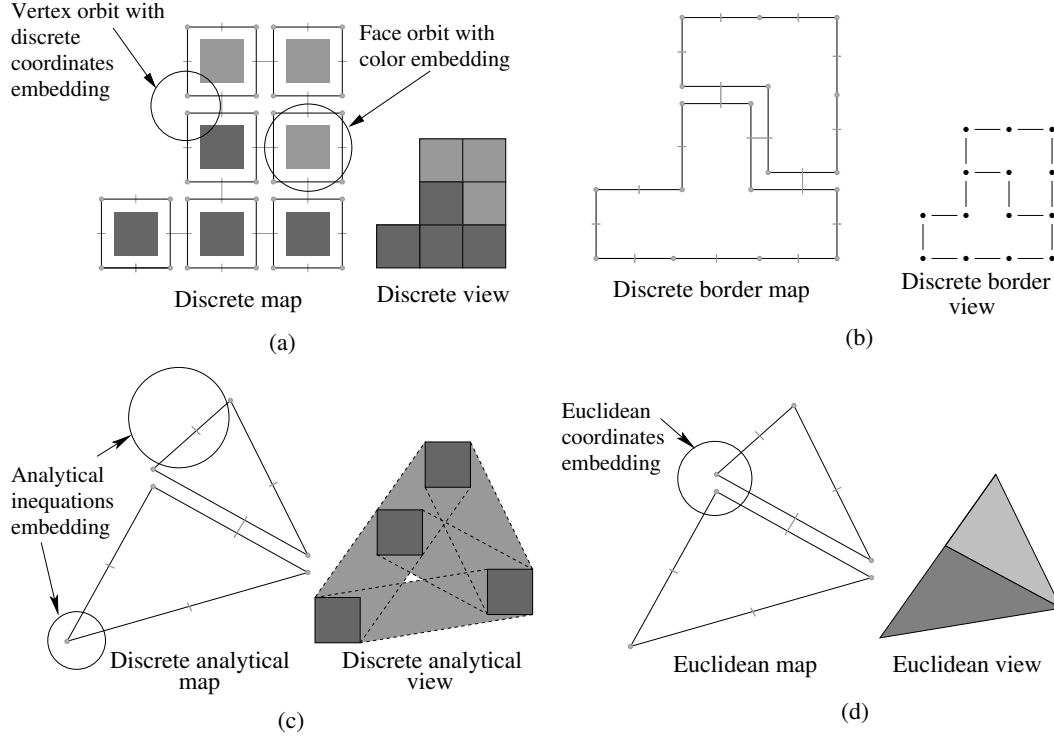


Fig. 11. The four levels of the pyramid.

called *Standard Model* [2,3]. We use this model because this is the only one that allows the analytical discretization of 4-connected polygons (segments and points). Moreover, this model is defined for any dimension.

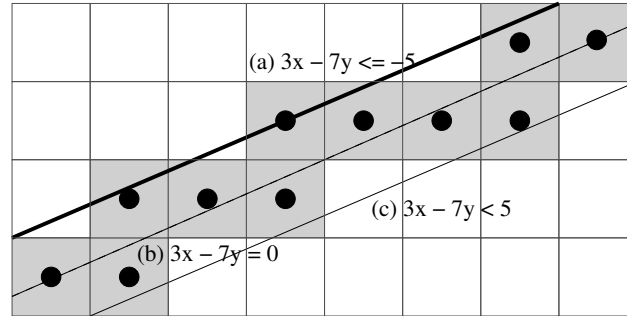


Fig. 12. Example of standard line and its analytical description. Pixels in gray correspond to the standard discretization of (b), i.e. they are pixels crossed by the line. (a) and (c) give the two opposite inequations of the analytical description. Note that the inequation (c) is a strict one. These inequations are satisfied by the pixel centers.

We can see in Fig. 12 an example of a standard line and its analytical description. A discrete standard line of parameters a, b, c is the set of integer points (x, y) verifying $-\omega \leq ax + by + c < \omega$ with $\omega = \frac{|a|+|b|}{2}$ and $a \geq 0$ or $a = 0$ and $b \geq 0$.

Level 2 map is obtained from level 1 map by applying 0-removals (see Sec. 7).

The embeddings of this level consist in a set of inequations. Each discrete segment is described by two opposite inequations associated with a topological edge, and each discrete vertex is described by four inequations associated with a topological vertex. To ensure the coherence between this level and the two previous ones, analytical inequations of level 2 must be satisfied by all 0-cells of level 1 (pixel centers of level 0).

6.2.5 Level 3: the Euclidean level

There are many different works [6,9,25,40] about modeling of Euclidean objects with maps. The G-map of this level corresponds to a generalized map with vertex embeddings that are 2D Euclidean coordinates (see Fig. 11(d)). The standard discretization of this level must correspond to the level 1.

Note that this level is not unique. Indeed, a discrete object can be seen as an equivalence class of Euclidean objects which have the same discretization. In our software we consider only one member of this class. It is thus not possible to ensure that, if we discretize a Euclidean object and then reconstruct it, we recover the same Euclidean object. On the other hand, a discrete object that is reconstructed, then discretized will remain unchanged. This supposes that the reconstruction process is deterministic. Since it depends on the starting point, a starting point choice convention is set up. Basically, the discrete point that has the smallest x value and among those the one with the smallest y value is the starting point.

7 Construction steps between consecutive levels

In this section, we present all algorithms we use to build a level from a neighbor level. Indeed, depending on the application, we want to build a level i given a level $i+1$ (discretization) or in the other way build a level $i+1$ given a level i (continuation).

7.1 Conversions between level 0 and level 1

- Level 1 is obtained by merging all incident faces of level 0 having same label. This corresponds to the use of 1-removal operations (see Sec. 3). The algorithm is the following: first, all edges between two pixels having the same label are marked. Then, these edges are removed during a single scan by applying simultaneous 1-removals (see Sec. 5).

We can see on Fig. 13 an example of discrete border level, as well as some

connections between the two levels. Note that each border dart of level 0 (i.e. without adjacent dart) is connected with one dart of level 1. Integer coordinates embeddings of level 0 can thus be accessed by level 1 through these connections.

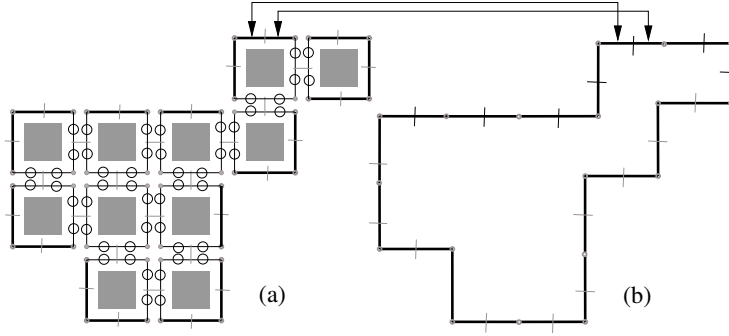


Fig. 13. Example of discrete (level 0) and border discrete (level 1) levels. (a) Level 0 map with marked darts (marks are represented by black circles). (b) Level 1 map obtained after several 1-removals. Arrows show some connections between darts, and all black darts are connected ones.

- To build level 0 from level 1, we need to reconstruct the pixels, for example by using a flood-fill algorithm on a matrix of 1-cells corresponding to level 1. We can see in Fig. 14(b) the level 0 obtained from Fig. 14(a). Note that the discrete space represented in Fig. 14(a) and Fig. 14(b) are not the same, but are duals.

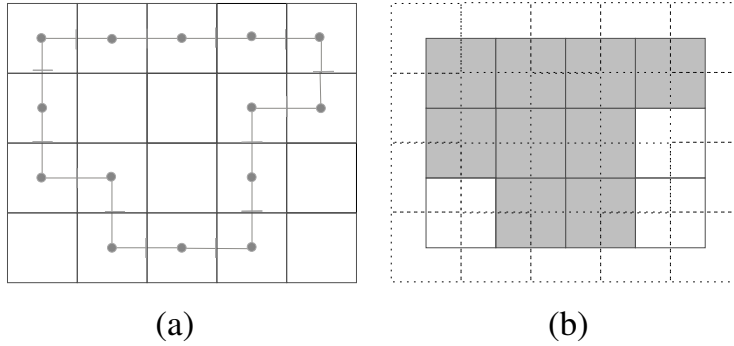


Fig. 14. (a) A discrete border level (level 1) and (b) its associated discrete view (level 0).

7.2 Conversions between level 1 and level 2

- To obtain level 2 from level 1, we use an operation called *analytical recognition* [13]. During the analytical recognition step, all vertices that belong to the same discrete segment are removed (by using 0-removal operation) in order to

keep only the two edge extremities. We can see in Fig. 15 the simplification made after a discrete segment recognition.

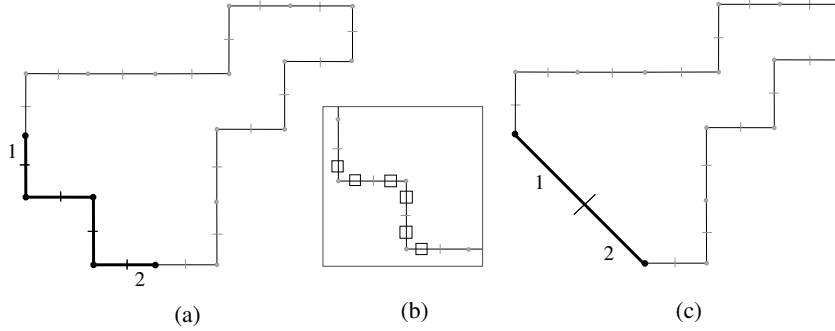


Fig. 15. Building level 2 from level 1. (a) Black darts belong to the recognized discrete segment. (b) Marked darts are removed using 0-removals. (c) Result.

At the end of the process, we have recognized all edges of the initial object. We can see in Fig. 16 the final result obtained for level 2. The generalized map has been simplified and each topological edge corresponds to a recognized discrete segment. Analytical inequations computed during the analytical recognition are linked with each edge and with each vertex of the map.

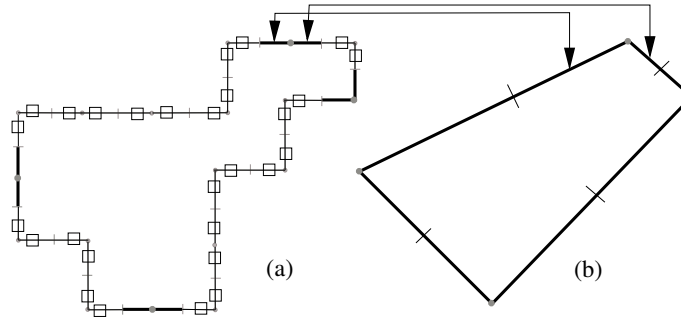


Fig. 16. Example of discrete border (level 1) and discrete analytical (level 2) levels. (a) Level 1 with marked darts. (b) Level 2 obtained after several 0-removals. Upper arrows show some connections between darts of level 1 and level 2.

- To build level 1 starting from level 2, we need to discretize the contours of level 2 (the discretization model used is the standard model, see Sec. 6). For each edge of level 2, if the standard discretization of the two edge end points differ from each other (see Fig. 17), the edge is incrementally discretized from one end point to the other, and a new vertex is inserted into the edge while the other end point is not reached. If the standard discretization of the two edge end points are equal (see Fig. 18), all darts of the edge are marked. Then, all marked edges are contracted using simultaneous 1-contractions (see Sec. 5).

However, some problems can arise such as degenerated faces (see Fig. 19). These faces disappear in the discrete level due to the loss of information in-

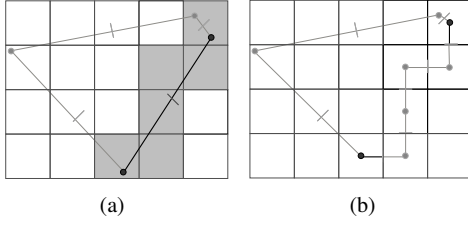


Fig. 17. First case. (a) The black edge is the processed one. Each filled pixel is crossed by the corresponding Euclidean segment. (b) Incremental discretization of the edge.

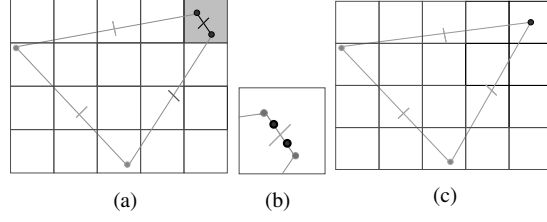


Fig. 18. Second case. (a) The back edge is the processed one. The filled pixel is the unique pixel crossed by the corresponding Euclidean segment. (b) The darts marked with black disks are contracted using 1-contraction. (c) Result.

herent to the discretization process. Thus, we chose to delete these faces mainly using 2-contractions (see Sec. 4).

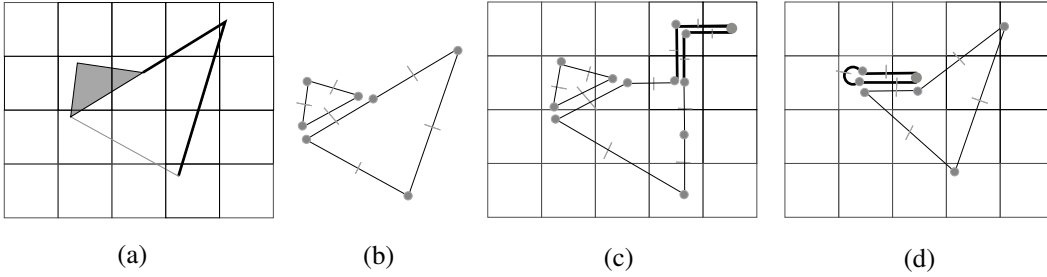


Fig. 19. Example of degenerated faces. (a) Euclidean object. (b) Map corresponding to the object in (a). (c) Standard discretization of the two black edges in (a). Black darts are contracted using two 2-contractions. (d) Standard discretization of the gray filled face in (a). Black darts are contracted using one 2-contraction.

7.3 Conversions between level 2 and level 3

Level 2 is topologically quite similar to level 3, but the main difference between these two levels consists in embeddings. Indeed, level 2 has analytical inequation embeddings whereas level 3 has Euclidean coordinate embeddings.

- To build level 3 from level 2, we make a copy of level 2, and compute the Euclidean coordinate embeddings during the *analytical reconstruction* step (see [13] for more details on particular cases that can occur during the analytical reconstruction).

- When we build level 2 starting from level 3, we also start with a copy of level 3, but we can eventually make some simplifications. For example, the gray square in Fig. 20(a) shows a set of vertices having the same standard discretization. All these vertices, except the two extremities, can thus be removed

using 0-removals. We keep the two extremities in order to keep the inequations of the other edges, that are computed from the Euclidean coordinates.

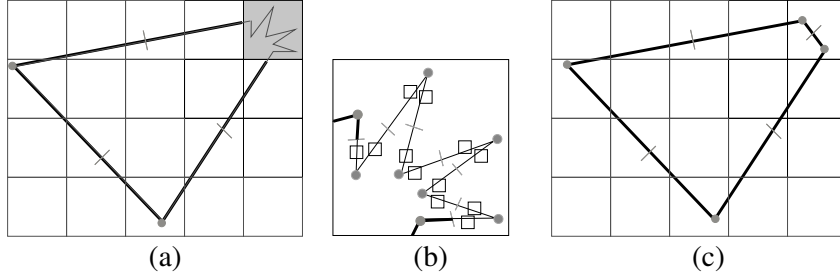


Fig. 20. Simplification of the discrete analytical level. (a) Euclidean object. (b) The darts marked with black squares are removed by 0-removals. (c) Discrete analytical level. Note that all black darts are connected ones.

8 Conclusion

In this paper, we have defined removal and contraction operations, that can be applied to any cells of any G-maps, whatever their respective dimensions. Moreover, we have studied how to simultaneously perform different operations. These definitions are homogeneous for any dimension. Since combinatorial maps [12,16] can be easily deduced from orientable generalized maps [37], these operations can also be defined on combinatorial maps.

We intend to revisit the works of Brun and Kropatsch for handling irregular pyramids of n -dimensional generalized maps. Properties of removal and contraction operations would enable us to establish relations between two contiguous levels within a pyramid, and thus between any levels. Efficient data structures could be deduced, taking these relations into account.

We have shown in a first experiment that we can easily define a combinatorial pyramid, thanks to the basic operations. This is a 2D pyramid, but it can be extended for higher dimensions, due to the general definition of removal and contraction operations for any dimension.

Moreover, this experiment shows a new way for using combinatorial pyramids. Indeed, a pyramid is generally used in order to represent an object at different levels of details. Here, we handle different representations of a same object, from continuous to discrete representation. This allows us to work with the best representation for a given operation. Moreover, we can progressively modify our structure, through the different intermediate levels. This simplifies the algorithms and the operations which maintain the coherence between the different levels of the pyramid.

Now we are studying the definition of several operations, such as modification or consultation algorithms. Classical Euclidean operations can be translated into discrete operations, by discretizing level 3. We are now considering typical discrete operations that will be translated into Euclidean operations by analytical reconstruction and analytical recognition. The main goal of our work is now to study the way of updating a pyramid after a given operation, in order to maintain the coherence between the different levels.

References

- [1] P. Alexandroff. Diskrete raüme. *Mat, Sbornik*, 2:501–518, 1937.
- [2] E. Andres. *Modélisation analytique discrète d’objets géométriques*. Thèse d’habilitation, Laboratoire IRCOM-SIC, University of Poitiers, France, december 2000.
- [3] E. Andres. Discrete linear objects in dimension n: the standard model. *Graphical Models*, 65:92–111, 2003.
- [4] E. Andres, R. Breton, and P. Lienhardt. Spamod: design of a spatial modeling tool. In *Digital and Image Geometry*, number 2243 in LNCS, pages 91–107, 2001.
- [5] S. Aubry and V. Hayward. Building hierarchical solid models from sensor data. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 196–201, Scottsdale, AZ, 1989.
- [6] Y. Bertrand. *Spécification algébrique et réalisation d’un modelleur interactif d’objets géométriques volumiques*. PhD thesis, Université Louis-Pasteur de Strasbourg, 1992.
- [7] Y. Bertrand, G. Damiand, and C. Fiorio. Topological encoding of 3d segmented images. In *Discrete Geometry for Computer Imagery*, number 1953 in LNCS, pages 311–324, Uppsala, Sweden, december 2000.
- [8] Y. Bertrand, G. Damiand, and C. Fiorio. Topological map: minimal encoding of 3d segmented images. In *Workshop on Graph based representations*, pages 64–73, Ischia, Italy, may 2001. IAPR-TC15.
- [9] Y. Bertrand and J.-F. Dufourd. Algebraic specification of a 3d-modeler based on hypermaps. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 56(1):29–60, January 1994.
- [10] G-P Bonneau and A. Gerussi. Hierarchical decomposition of datasets on irregular surface meshes. In *Computer Graphics International*, pages 59–65, Hannover, 1998.
- [11] J.P. Braquelaire, P. Desbarats, and J.P. Domenger. 3d split and merge with 3-maps. In *Workshop on Graph based representations*, pages 32–43, Ischia, Italy, may 2001. IAPR-TC15.

- [12] J.P. Braquelaire, P. Desbarats, J.P. Domenger, and C.A. Wüthrich. A topological structuring for aggregates of 3d discrete objects. In *Workshop on Graph based representations*, pages 193–202, Austria, may 1999. IAPR-TC15.
- [13] R. Breton, I. Sivignon, F. Dupont, and E. Andres. Towards an invertible euclidean reconstruction of a discrete object. In *Discrete Geometry for Computer Imagery*, number 2886 in LNCS, pages 246–256, Naples, Italy, november 2003.
- [14] E. Brisson. Representing geometric structures in d dimensions: topology and order. *Discrete and Computational Geometry*, 9(1):387–426, 1993.
- [15] L. Brun. *Segmentation d’images couleur à base topologique*. PhD thesis, Université de Bordeaux I, december 1996.
- [16] L Brun and W.G. Kropatsch. Dual contraction of combinatorial maps. In *Workshop on Graph based representations*, pages 145–154, Austria, may 1999. IAPR-TC15.
- [17] L Brun and W.G. Kropatsch. Pyramids with combinatorial maps. Technical report 57, Institute of Computer Aided Automation, Vienna University of Technology, Austria, december 1999. URL: <http://www.prip.tuwien.ac.at/>.
- [18] L Brun and W.G. Kropatsch. The construction of pyramids with combinatorial maps. Technical report 63, Institute of Computer Aided Automation, Vienna University of Technology, Austria, june 2000. URL: <http://www.prip.tuwien.ac.at/>.
- [19] L Brun and W.G. Kropatsch. Contraction kernels and combinatorial maps. In *Workshop on Graph based representations*, pages 12–21, Ischia, Italy, may 2001. IAPR-TC15.
- [20] P. Burt, T.-H. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man and Cybernetics*, 11(12):802–809, 1981.
- [21] R. Cori. *Un code pour les graphes planaires et ses applications*. PhD thesis, Université de Paris VII, 1973.
- [22] G. Damiand. *Définition et étude d’un modèle topologique minimal de représentation d’images 2d et 3d*. PhD thesis, Université de Montpellier II, december 2001.
- [23] G. Damiand and P. Lienhardt. Removal and contraction for n-dimensional generalized maps. In *Discrete Geometry for Computer Imagery*, number 2886 in Lecture Notes in Computer Science, pages 408–419, Naples, Italy, november 2003.
- [24] L. de Floriani and B. Falcidieno. A hierarchical boundary model for solid object representation. *ACM Transactions on Graphics*, 7(1):42–60, 1998.
- [25] J.-F. Dufourd and F. Puitg. Functional specification and prototyping with combinatorial maps. *Computational Geometry - Theory and Applications*, 16:129–156, 2000.

- [26] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.
- [27] H. Elter. *Etude de structures combinatoires pour la représentation de complexes cellulaires*. PhD thesis, Université Louis-Pasteur, Strasbourg, september 1994.
- [28] C. Fiorio. *Approche interpixel en analyse d'images : une topologie et des algorithmes de segmentation*. PhD thesis, Université Montpellier II, 24 novembre 1995.
- [29] A. Jacques. Constellations et graphes topologiques. In *Combinatorial Theory and Applications*, volume 2, pages 657–673, 1970.
- [30] J. Jolion and A. Montanvert. The adaptive pyramid : a framework for 2d image analysis. *Computer Vision, Graphics and Image Processing*, 55(3):339–348, may 1992.
- [31] E. Khalimsky, R. Kopperman, and P.R. Meyer. Boundaries in digital planes. *J. of Applied Mathematics and Stochastic Analysis*, 3:27–55, 1990.
- [32] V.A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46:141–161, 1989.
- [33] W.G. Kropatsch. Building irregular pyramids by dual graph contraction. Technical report PRIP-TR-35, Dept. for Pattern Recognition and Image Processing, Institute for Automation, Technical University of Vienna, Austria, july 1994.
- [34] W.G. Kropatsch. Building irregular pyramids by dual-graph contraction. *Vision, Image and Signal Processing*, 142(6):366–374, december 1995.
- [35] W.G. Kropatsch. Abstraction pyramids on discrete representations. In *Discrete Geometry for Computer Imagery*, number 2301 in LNCS, pages 1–21, Bordeaux, France, april 2002.
- [36] P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer Aided Design*, 23(1):59–82, 1991.
- [37] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3):275–324, 1994.
- [38] P. Meer. Stochastic image pyramids. *Computer Vision, Graphics and Image Processing*, 45:269–294, 1989.
- [39] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, april 1991.
- [40] F. Vidil. Développement d'un modeleur à base topologique. Technical report, Stage de 3^{ème} année ENSEiRB, 2001.