SIGNAL-IMAGE-COMMUNICATIONS
FRE 2731 CNRS

nD generalized map pyramids: three equivalent representations

C. Grasset-Simon, G. Damiand, P. Lienhardt

Research Report

n° 2005-03

September 2005

SIC, Université de Poitiers, France

SIC, UFR SFA, Université de Poitiers,
Bât. SP2MI, Téléport 2, Boulevard Marie et Pierre Curie, BP 30179, 86962 FUTUROSCOPE CHASSENEUIL Cedex, FRANCE
Tél : (+33)(0)5 49 49 65 99
Fax : (+33)(0)5 49 49 65 70

Contents

1 Introduction

In [1] we have defined the notion of pyramids of n-dimensional generalized maps. Generalized maps represent the topology of any partition of any n-dimensional quasi-manifold with or without boundaries [2]. So, they can represent any partition of any nD image without loss of topological information (all cells are represented, together with all adjacency and incidence relations between cells and topological order). Moreover, generalized maps are homogeneously defined for any dimension: this is another advantage of this structure for which the definition of generic operations and algorithms is easier. The definition of n-G-map pyramids is based upon a general operation for removing and contracting cells of any dimension [3]. More precisely, each level of a pyramid is a "reduction" of the previous one, computed by applying this operation. From this "principle" we deduce the definition of n-G-map pyramids, several useful notions and properties [4] and [1]. So, with n-G-map pyramids, 2D, 3D or 4D images (3D plus time for instance) can be processed using a single formalism, without loss of topological information.

The main drawback of n-G-map pyramids is the fact that they can be very expensive (in memory space). So we study here three equivalent generic representations for such pyramids, and conversion operations between these representations.

Section 2 is a reminder of the notion of generalized map, the general operation of cell contraction and removal and the notion of generalized map pyramids. We study in section 3 three different representations of generalized map pyramids. We present in section 4 conversion algorithms between these representations. And we prove in section 5 their equivalence. Conclusion and further issues are discussed in section 6.

2 Recalls: n-G-map, cell removal and contraction and n-G-map pyramids

n-dimensional generalized maps (or n-G-maps) make it possible to describe each cell of a subdivision, and more generally any subdivision of any quasi-manifold [5]. An n-G-map is a set of abstract elements (called darts), together with applications defined on these darts.

Definition 1 (n-G-map) : *Let $n \geq 0$. An n-dimensional generalized map $G = (D, \alpha_0, \dots, \alpha_n)$ is defined by:*

1. *D a finite set of darts;*
2. *$\forall k$, $0 \leq k \leq n$, α_k an* involution[1] *on D;*
3. *$\forall k, j$, $0 \leq k < k+2 \leq j \leq n$, $\alpha_k \alpha_j$ is an involution.*

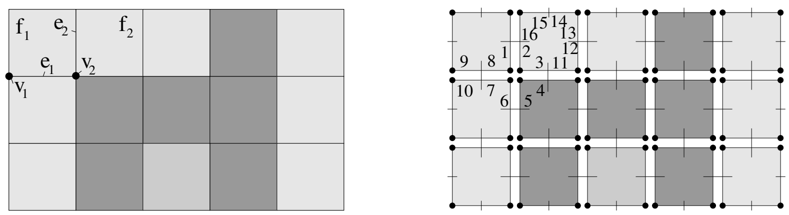
Let G be an n-G-map, and S be the corresponding subdivision. Intuitively, a dart of G corresponds to an $(n+1)$-tuple of cells $(c_0, \dots, c_n)$, where c_i is an i-dimensional cell that belongs to the boundary of c_{i+1} (cf. [6] and Fig. 1). α_i associates darts corresponding with $(c_0, \dots, c_n)$ 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).

Cells are implicitly described as sets of darts through the notion of orbit (see [2] for more details).

Definition 2 (orbit and i-cell) : *Let $\{\Pi_0, \dots, \Pi_n\}$ be a set of permutations on D. The orbit of an element $d \in D$ related to this set of permutations is $< \Pi_0, \dots, \Pi_n > (d) = \{\Phi(d), \Phi \in < \Pi_0, \dots, \Pi_n >\}$, where $< \Pi_0, \dots, \Pi_n >$ denotes the group of permutations generated by $\Pi_0, \dots, \Pi_n$. Let $d \in D$, $N = \{0, 1, \dots, n\}$ and let $i \in N$. The i-cell incident to d is the orbit*

$$<>_{N-\{i\}} (d) = < \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n > (d).$$

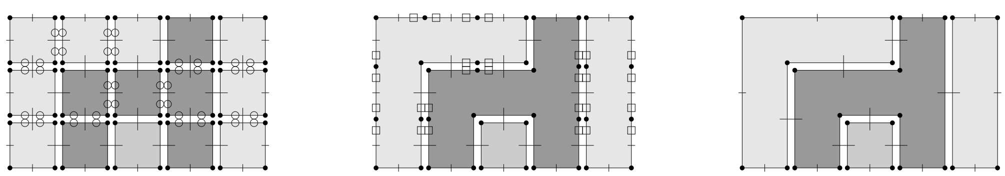
[1] An involution f on a finite set S is a one to one mapping from S onto S such that $f = f^{-1}$.



(a) (b)

Figure 1: (a) A 2D image. (b) The corresponding 2-G-map. Darts are represented by numbered black segments. Two darts related by α_0 share a small vertical segment (ex. darts 14 and 15). Two darts related by α_1 share a same point (ex. darts 2 and 3). Two distinct darts related by α_2 are parallel and close to each other (ex. darts 3 and 4); otherwise, the dart is its own image by α_2 (ex. dart 14). Dart 9 corresponds to (v_1, e_1, f_1), dart $8 = 9\alpha_0$ corresponds to (v_2, e_1, f_1), $1 = 8\alpha_1$ corresponds to (v_2, e_2, f_1), and $2 = 1\alpha_2$ corresponds to (v_2, e_2, f_2). The vertex (0-cell) incident to dart 2 is $< \alpha_1, \alpha_2 > (2) = \{1, 2, 3, 4, 5, 6, 7, 8\}$, the edge (1-cell) incident to dart 9 is $< \alpha_0, \alpha_2 > (9) = \{7, 8, 9, 10\}$, and the face (2-cell) incident to dart 2 is $< \alpha_0, \alpha_1 > (2) = \{2, 3, 11, 12, 13, 14, 15, 16\}$.

In order to define n-G-map pyramids, Damiand and Lienhardt have defined the operation of "simultaneous removals and contractions of cells of any dimension" [3] which allows to contract and remove a set of cells of any dimension in a simultaneous manner (see Fig. 2 for examples and [3] for more details). With this operation we can merge different regions (using removals of 1-cells in 2D or 2-cells in 3D), or simplify region boundaries (using removals of 0-cells in 2D or 0-cells and 1-cells in 3D).



a b c

Figure 2: An example of simultaneous removals of cells. (a) A 2-G-map corresponding to an image. A $\circ$ marks a dart of a 1-cell to be removed in order to merge certain 2-cells. (b) The resulting 2-G-map. A $\square$ marks a dart of a 0-cell to be removed in order to simplify the boundary of regions. (c) The resulting 2-G-map where the region boundaries are simplified.

The formal and general definition of this operation is:

Definition 3 (Simultaneous removal and contraction of any cells) : *Let $G = (D, \alpha_0, \ldots, \alpha_n)$ be an n-G-map, $R_0, \ldots, R_{n-1}$ be sets of 0-cells,…, $(n-1)$-cells to be removed and $C_1, \ldots, C_n$ be sets of 1-cells,…, n-cells to be contracted*[2]. *Let $R = \cup_{i=0}^{n-1} R_i$ and $C = \cup_{i=1}^{n} C_i$. Two preconditions have to be satisfied:*

$(C1)$ cells are disjoint (i.e. $\forall c, c' \in C \cup R$, $c \cap c' = \emptyset$),
$(C2)$ "the degree of each cell is locally[3] *2", i.e.:*

[2] $R_n = \emptyset$ and $C_0 = \emptyset$ since it is not possible to remove n-cells nor to contract 0-cells
[3] The degree of an i-cell is the number of distinct incident $(i + 1)$-cells. The local degree of an i-cell is its degree when we consider the cell locally. For example, the degree of a vertex incident to a loop is 1 but its local degree is 2 since locally two edges are incident to this vertex.

- $\forall i,\ 0 \le i \le n-2,\ \forall d \in R_i,\ d\alpha_{i+1}\alpha_{i+2} = d\alpha_{i+2}\alpha_{i+1}$;
- $\forall i,\ 2 \le i \le n,\ \forall d \in C_i,\ d\alpha_{i-1}\alpha_{i-2} = d\alpha_{i-2}\alpha_{i-1}$.

$\forall i \in N$, let $SD_i = (R_i \cup C_i)\alpha_i - (R_i \cup C_i)$ (it is the set of surviving darts "neighbor" of removed and contracted cells). The resulting n-G-map is $G' = (D', \alpha'_0, \ldots, \alpha'_n)$ defined by:

- $D' = D - (C \cup R)$;
- $\forall i \in N,\ \forall d \in D' - SD_i,\ d\alpha'_i = d\alpha_i$;
- *$\forall i \in N,\ \forall d \in SD_i,\ d\alpha'_i = d' = d(\alpha_i\alpha_{k_1})\ldots(\alpha_i\alpha_{k_p})\alpha_i$, where p is the smallest integer such that $d' \in SD_i$, and $\forall j, 1 \le j < p$, if $d_j = d(\alpha_i\alpha_{k_1})\ldots(\alpha_i\alpha_{k_{j-1}})\alpha_i \in R_i$ then $k_j = i+1$ else ($d_j \in C_i$) $k_j = i-1$.*

An n-G-map pyramid is a hierarchical structure, each level of which is an n-G-map. The first level describes the initial data; the other levels describe successive reductions of the previous ones by removing and/or contracting some cells.

Definition 4 (n-G-map pyramid) : *Let $n,\ m \ge 0$. An $(m+1)$-level pyramid $\mathcal{P}$ of n-dimensional generalized maps is the set $\mathcal{P} = \{G^k\}_{0 \le k \le m}$ where:*

1. *$\forall k,\ 0 \le k \le m,\ G^k = (D^k, \alpha^k_0, \ldots, \alpha^k_n)$ is an n-G-map;*
2. *For each k, $0 \le k < m$, for each $i \in N$, let R^k_i (resp. C^k_i) be sets of i-cells and $R^k = \bigcup_{i=0}^n R^k_i$ (resp. $C^k = \bigcup_{i=0}^n C^k_i$) with $R^k_n = C^k_0 = \emptyset$. The two preconditions (C1) and (C2) of the definition 3 have to be satisfied;*
3. *$\forall k,\ 0 < k \le m$, G^k is obtained from G^{k-1} by removing the cells of R^{k-1} and contracting the cells of C^{k-1}.*

Each level of a pyramid is a "reduction" of the previous one, computed by applying the previous operation and generaly the first level corresponds to the initial data. Examples of 2D and 3D pyramids are provided in Fig. 3 and Fig. 4.

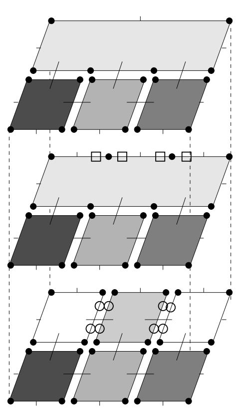


Figure 3: A 2-G-map pyramid composed of three levels. A □ (resp. a ○) marks a dart of a 0-cells (resp. 1-cells) to be removed.

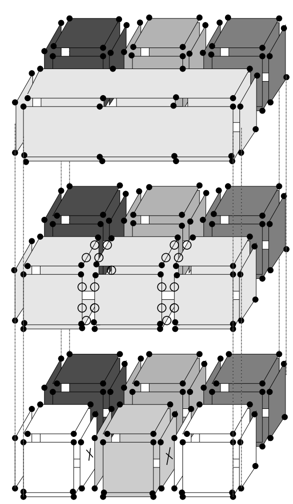


Figure 4: A 3-G-map pyramid composed of three levels. The second level is obtained by removing the 2 faces between the 3 volumes in the foreground. The third level is obtained by removing 6 edges around the volume in the foreground.

Two major properties of n-G-map pyramids are:

Proposition 1 :

1. *Each dart which belongs to a removed or a contracted cell of level k does not belong to another removed or contracted cell at any level. More formally: $\forall i, j \in N$, $\forall k, l \in [0..m-1]$ we have:*

$$\begin{cases} R_i^k \cap C_j^l = \emptyset, \\ R_i^k \cap R_j^l = \emptyset, \text{ with } i \neq j \text{ or } k \neq l, \\ C_i^k \cap C_j^l = \emptyset, \text{ with } i \neq j \text{ or } k \neq l. \end{cases}$$

2. *Let k, $0 \leq k < m$. A one to one mapping φ^k exists between the surviving darts of G^k (i.e. the darts which are not removed nor contracted), and the darts of G^{k+1} ($\varphi^k : D^k - (R^k \cup C^k) \longrightarrow D^{k+1}$). φ^k is called the* successor *relation and $(\varphi^{k-1})^{-1}$ the* predecessor *relation.*

Remarque 1 : *In order to simplify the notations, a dart of G^k and its image in G^{k+1} are denoted by the same name. So, $D^{k+1} = D^k - (R^k \cup C^k)$.*

These properties can be easily deduced from the definition of simultaneous removals and contractions operation [3], and from the definition of n-G-map pyramids. Moreover these properties are useful for the definition of different representations of n-G-map pyramids.

3 Different representations of n-G-map pyramids

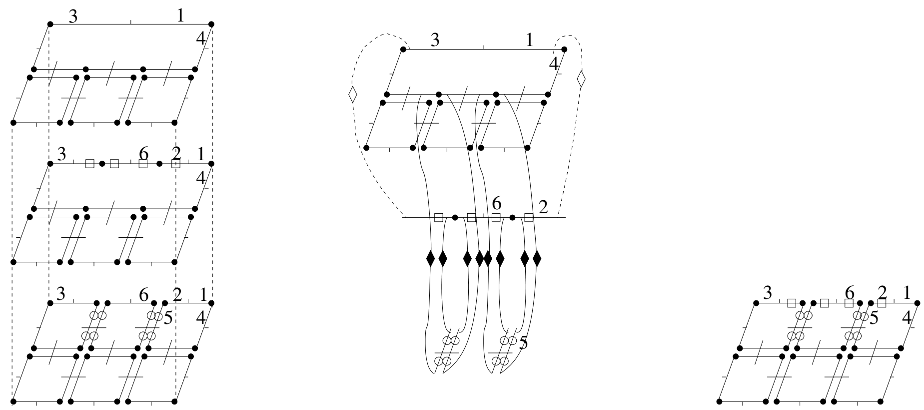
An n-G-map pyramid can be described more or less explicitly according to the expected space/time complexity. In this section we present three possible representations: *explicit*, *hierarchical* and *implicit* (see Fig. 5). The first one immediately follows the definition of n-Gmap pyramids, the other two are proposed in order to reduce the cost in memory space. These last two representations generalize similar existing structures proposed in [7, 8]. Note that depending on particular needs, it could be possible to propose other representations.

3.1 Description of the representations

Explicit n-G-map pyramid: All levels and one to one mappings between the levels are explicitly represented (see Fig. 5-a). The representation thus contains $m+1$ n-G-maps, and each dart is linked to its predecessor (except the darts of level 0) and to its successor (except the darts of the last level and the darts which belong to a removed or contracted cell). Moreover, for each level, two marks are associated with each dart which belongs to a removed or contracted cell: the type of the operation (contraction or removal) and the dimension of the cell.

Hierarchical n-G-map pyramid: This representation contains a single set of darts, i.e. the darts of level 0 map (cf. remark 1). All involutions are explicitly represented for all levels. More precisely, for each dart a table gives the images of this dart by all involutions (see Fig. 5-b). Two marks are associated with each dart which belongs to a removed or contracted cell in the same way than for the explicit representation. A possible optimization is the following one: for each dart we only represent distinct images for all involutions and the levels where images change. A structure based upon a similar principle has been proposed in order to model complex architectural environments [8].

Implicit n-G-map pyramid: This representation contains the map of level 0 with additional information which makes possible to compute the other levels. For this three marks are associated with each dart that has disappeared: the type of the operation (removal or contraction), the dimension of the removed or contracted incident cell, and the level at which the cell disappears. A similar representation is proposed by Brun and Kropatsch [7] for 2D combinatorial pyramids.



level 0	α_0^0	α_1^0	α_2^0
1	2	4	1
2	1	5	2

level 1	α_0^1	α_1^1	α_2^1
1	2	4	1
2	1	6	2

level 2	α_0^2	α_1^2	α_2^2
1	3	4	1

a

	level 0			*level* 1			*level* 2		
	α_0^0	α_1^0	α_2^0	α_0^1	α_1^1	α_2^1	α_0^2	α_1^2	α_2^2
1	2	4	1	2	4	1	3	4	1
2	1	5	2	1	6	2	–	–	–

b

	α_0	α_1	α_2
1	2	4	1
2	1	5	2

c

Figure 5: Three representations of a same pyramid. For each representation, the corresponding array shows images by involutions α for the two darts 1 and 2. In the explicit representation, each dart is duplicated. In the hierarchical representation, darts are not duplicated, but involutions exist at each level. The implicit representation describes only the darts and the involutions of the first level. (a) *Explicit.* A □ (resp. a ∘) marks a dart of a 0-cell (resp. a 1-cell) to be removed. (b) *Hierarchical.* Removed cells are marked in the same way as in the explicit representation. Each dart is drawn in the last pyramid level where it exists. When two darts linked by α_i are drawn in the same level, their link α_i is drawn in the usual way. Otherwise, the links α_0^1 and α_0^0 between two darts of two different levels are represented by lines with ♢ and the link α_1^0 by line with ♦. (c) *Implicit.* A □ (resp. a ∘) marks a dart of a 0-cell of level 0 (resp. a 1-cell of level 1) to be removed.

3.2 Definition of the representations

Following these descriptions we can express a definition for each representation.

In order to do that we need the connecting walk notion. A connecting walk is the sequence of darts at a given level which separates a surviving dart and its successor in the next level. $Ch_{(i,a,a+1)}(d)$ denotes the connecting walk between levels a and $a+1$, for any dart $d \in D^a$ neither contracted nor removed ($d \in D^{a+1}$) and dimension i.

The notion of connecting walk between two consecutive levels is formally defined in the following way:

Definition 5 (Connecting walk) : *Let $i \in N$ and a be such that $0 \leq a < m$.*
For each dart $d \in D^a$ which is not contracted or removed, $Ch_{(i,a,a+1)}(d)$ is defined by:
if $d \notin SD_i^a$: $Ch_{(i,a,a+1)}(d) = (d.\alpha_i^a)$,
else: $Ch_{(i,a,a+1)}(d) = (d_1, d_2, \ldots, d_p)$,
where $\forall u,\ 2 \leq u \leq p$:

$$d_1 = d\alpha_i^a$$

$$d_u = d_{u-1}.\alpha_{k_{u-1}^a}$$

$$k_u = \begin{cases} i \text{ if } u \text{ is even} \\ i+1 \text{ if } u \text{ is odd and } d_u \in R_i^a, \\ i-1 \text{ if } u \text{ is odd and } d_u \in C_i^a, \end{cases}$$

and p is the smaller integer such that $d' \in SD_i^a$.

$d.LDC_{(i,a,a+1)}$ (or $LDC_{(i,a,a+1)}(d)$) denotes the last dart of the connecting walk $Ch_{(i,a,a+1)}(d)$.
According to the previous definition, we have:
if $d \notin SD_i^a$: $d.LDC_{(i,a,a+1)} = d.\alpha_i^a$,
else: $d.LDC_{(i,a,a+1)} = d' = d.(\alpha_i^a.\alpha_{k_1}^a)\ldots(\alpha_i^a.\alpha_{k_p}^a).\alpha_i^a$

$$\text{where: } \forall u,\ 1 \leq u \leq p,\ k_u = \begin{cases} i+1 \text{ if } d_u \in R_i^a, \\ i-1 \text{ if } d_u \in C_i^a, \end{cases}$$

and p is the smaller integer such that $d' \in SD_i^a$.

Definition 6 (Explicit representation) : *The explicit representation of an n-G-map pyramid with $m+1$ levels is $E = ((G^a)_{0 \leq a \leq m}, (succ^a)_{1 \leq a \leq m})$ where:*

- *$\forall a,\ 0 \leq a \leq m,\ G^a = (D^a, (\alpha_i^a)_{0 \leq i \leq n})$ is an n-G-map;*

such that:

- *$\forall a,\ 0 \leq a < m,\ D^a = S^a \cup (R^a \cup C^a)$ where R^a and C^a respect the second item of the n-G-map pyramid definition (definition 4 page 6);*
- *$\forall a,\ 0 < a \leq m,\ succ^a : S^{a-1} \longrightarrow D^a$ is a one-to-one mapping which verifies: $\forall d \in S^{a-1}$, $\forall i$, $0 \leq i \leq n$, $d.succ^a.\alpha_i^a$ is the image by $succ^a$ of $d.LDC_{(i,a-1,a)}$.*

NB: $\forall a,\ 0 < a \leq m$, $pred^a$ denotes $(succ^a)^{-1}$.

Definition 7 (Hierarchical representation) : *The hierarchical representation of an n-G-map pyramid with $m+1$ levels is $H = (D, (\alpha_i^a)_{0 \leq i \leq n, 0 \leq a \leq m})$ where:*

- *$D = D^m \cup \bigcup_{a=0}^{m-1}(R^a \cup C^a)$ with $\forall a,\ 0 \leq a < m,\ R^a = \bigcup_{i=0}^{n} R_i^a$, $C^a = \bigcup_{i=0}^{n} C_i^a$, $R_n^a = \emptyset$ and $C_0^a = \emptyset$;*

Let D^a be $D - \bigcup_{k=0}^{a-1}(R^k \cup C^k)$ $\forall a$ $1 \leq a \leq m$ and $D^0 = D$, H is such that:

- *$\forall a,\ 0 \leq a \leq m,\ G^a = (D^a, (\alpha_i^a)_{0 \leq i \leq n})$ is an n-G-map;*
- *$\forall a,\ 0 \leq a < m$ R^a and C^a respect the second item of the n-G-map pyramid definition (definition 4 page 6);*
- *$\forall a,\ 0 < a \leq m,\ \forall d \in D^a,\ \forall i,\ 0 \leq i \leq n,\ d.\alpha_i^a$ is $d.LDC_{(i,a-1,a)}$.*

Definition 8 (Implicit representation) : *The implicit representation of an n-G-map pyramid with $m+1$ levels is $I = (G^0)$ where:*

- *$G^0 = (D^0, (\alpha_i^0)_{0 \leq i \leq n})$ is an n-G-map;*
- *$D^0 = D^m \cup \bigcup_{a=0}^{m-1}(R^a \cup C^a)$ where $\forall a,\ 0 \leq a < m,\ R^a = \bigcup_{i=0}^{n} R_i^a$, $C^a = \bigcup_{i=0}^{n} C_i^a$ $R_n^a = \emptyset$ and $C_0^a = \emptyset$;*

Let $\forall a,\ 1 \le a \le m$*,*

> $D^a = D - \bigcup_{k=0}^{a-1}(R^k \cup C^k)$*,*
> $G^a = (D^a, (\alpha_i^a)_{0\le i\le n})$ *such that* $\forall d \in D^a$*,* $d.\alpha_i^a$ *is* $d.LDC_{(i,a-1,a)}$ *(we can show that for* $1 \le a \le m$*,* G^a *is an n-G-map),*

I is such that $\forall a,\ 0 \le a < m$*,* R^a *and* C^a *respect the second item of the n-G-map pyramid definition (definition 4 page 6).*

3.3 Comparison

As we will see in the two following sections, we can move from one representation to another (explicit → hierarchical → implicit → explicit) and these three representations are equivalent (in particular, the constraints of the pyramid definition can be easily retrieved for each representation).

These three representations have different space complexities. Let p be the number of darts of level 0, n be the dimension of the space and m be the number of pyramid levels. Since we have at most pm darts in the *explicit* representation, and since a dart contains the information concerning its neighbors for all n involutions, the space complexity of this representation is $\mathcal{O}(mnp)$. In the *hierarchical* representation, we have p darts and for each dart we have at most mn neighbors. Then, the space complexity of this representation is $\mathcal{O}(mnp)$. But note that if we consider the proposed optimization (for each dart we only represent the distinct images for each involution), most of the time, the average number l of different neighbors for each involution and each dart will be smaller than m and depends of the considered pyramid. Indeed, $l = m$ means that the dart has a different neighbor at each level (i.e. it is the neighbor of a disappeared cell at each level). But, it is not possible for all the darts to be both survivors and neighbors of a disappearing cell since if a dart is the neighbor of a disappearing cell necessarily a cell disappears and so a dart does not survive. Therefore, l is strictly smaller than m (if $1 < m$). So, with the optimization, this representation has a cost $\mathcal{O}(lnp)$ (with $1 \le l < m$).
In the *implicit* representation, we have p darts and for each of them n involutions. So this representation has a cost $\mathcal{O}(np)$.
This implies that the memory space of the explicit representation is larger than the hierarchical one, which is itself larger than the implicit one. This is due to the fact that the explicit representation is characterized by an important redundancy of information, since all darts and all involutions are present at each level. The information contained in the hierarchical representation is less redundant since the darts are not duplicated. Finally, there is no redundant information in the implicit representation since only one level is explicitly represented.

Most operations (construction, features computation . . .) are realized by exploring some orbits at a level of the pyramid (for example, in 2D, the color of a region is retrieved by exploring the face orbits which have been merged into this region in order to use the pixel colors). An orbit is explored, dart by dart, with a breath first search algorithm which use all the involutions of the orbit. Retrieval the neighbor of a dart for an involution occurs in constant time in the explicit and hierarchical representations since the information is explicitly described. But it is not the case in the implicit representation where it is necessary to compute it by following the "path" of disappeared darts. So the complexity of this computing depends on the number of darts of the "path". Consequently, most operations have a complexity proportional to the number of darts of the examined orbits for the explicit and hierarchical representations, while this complexity is multiplied by the average length of the "paths" for the implicit representation.

We can summarize the main characteristics of the three representations by giving here their main advantage and drawback and their main complexities in Tab. 1:

- *explicit*: it has, at each level, a real n-G-map which can be considered independently. It allows to use swapping techniques in order to retain in memory only the information necessary for the current process. But it takes up important memory space;
- *hierarchical*: it takes less memory space than the explicit representation while allowing

to directly access any level by using involutions of this level. But the operations are not necessarily local since darts are shared by different levels;

- *implicit*: it minimizes the required memory space, and some modifications are easier. For instance, removing or contracting a given cell is simple, since we do not have to propagate modifications. But a level can not be directly accessed, it is necessary to compute it when it is required.

Representation / Complexity	Explicit	Hierarchical	Implicit
Memory space	$\mathcal{O}(mnp)$	$\mathcal{O}(mnp)$	$\mathcal{O}(np)$
Exploration of orbits	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(kl)$

Table 1: Recapitulation of the complexities for each representation. p is the number of darts of level 0, n is the dimension of the space, m is the number of pyramid levels, k is the number of darts of the orbits and l is the average length of "paths" (l depends on the pyramid).

4 Conversion algorithms

Here we propose three algorithms allowing to construct any representation given another one among the three ones presented before. These algorithms are local ones: the output representation (darts, links, and marks) is built by processing each input dart in two steps: first, one or several copies of the dart are created and second, all necessary information is added. These algorithms use different functions dealt within subsection 4.4.

4.1 Explicit to hierarchical representation

Algorithm 1 builds the hierarchical representation given the explicit one. The idea consists in "merging" each dart with all its successors. The principle of algorithm 1 as follows: for each dart, create a copy of level 0, then copy the involutions at each level and finally report the different marks on the copy. The creation of the representation is achieved dart by dart and in three steps:

1. The creation of the corresponding dart in the hierarchical representation;
2. The linking of this corresponding dart to its neighbor for each involution (α_i with $i \in N$) and each level if it already exists;
3. If the dart disappears at a level, we mark the dart with the same two marks as those contained in the initial dart in order to indicate how it disappears: removal (R) or contraction (C), and the dimension of the cell ($i \in N$).

This algorithm links each dart of the first pyramid level to its neighbor for each involution at each level. In order to compute the last successor of a dart (the top most dart), we explore at most all the levels. So this algorithm has a cost $\mathcal{O}(pmn)$, p being the number of darts of the first pyramid level, n the dimension of the G-map and m the number of pyramid levels.

4.2 Hierarchical to implicit representation

Algorithm 2 builds the implicit representation given the hierarchical one. The principle of this algorithm for each dart is:

1. to create a copy;

Algorithm 1: Construction of the hierarchical representation given the explicit one.

```
Input: Explicit representation (whose different levels are G^0, ..., G^m)
Output: Hierarchical representation (whose unique set of darts is D')
foreach dart d ∈ G^0 do
    d' ← create_copy(d, D') ;
    lev_d ← last_level_for_dart(d) ;
    for lev ← 0 to lev_d do
        for i ← 0 to n do
            if yet_create_copy(dα_i^lev) then
                link_by_α_i(d',the_copy_of(dα_i^lev), lev) ;
    d^{lev_d} ← top_most_dart(d) ;
    if is_marked_by_R(d^{lev_d}) then
        mark_by_R_i(d',dim_of_rem_cell(d^{lev_d})) ;
    else if is_marked_by_C(d^{lev_d}) then
        mark_by_C_i(d',dim_of_con_cell(d^{lev_d})) ;
```

2. to copy the level 0 involutions, i.e. to link the copy of the dart to its neighbor for each involution (α_i with $i \in N$) if it already exists;
3. then to put on the copy the information contained in the initial dart: the mark if the dart belongs to a removed or a contracted cell, the dimension of the cell, and the level at which the cell disappears (this last mark is not a copy, it is computed).

Algorithm 2: Construction of the implicit representation given the hierarchical one.

```
Input: Hierarchical representation (whose unique set of darts is D')
Output: Implicit representation (whose unique level is G)
foreach dart d ∈ D' do
    d' ← create_copy(d, G) ;
    for i ← 0 to n do
        if yet_create_copy(dα_i^0) then
            link_by_α_i(d', the_copy_of(dα_i^0)) ;
    lev_d ← last_level_for_dart(d) ;
    if is_marked_by_R(d) then
        mark_by_R_i(d',dim_of_rem_cell(d^{lev_d}),lev_d) ;
    else if is_marked_by_C(d) then
        mark_by_C_i(d',dim_of_con_cell(d^{lev_d}),lev_d) ;
```

This algorithm links each dart of the implicit representation to its neighbor for each involution. The cost to compute the last level at which a dart exists is $\mathcal{O}(m)$. So this algorithm has a cost $\mathcal{O}(pmn)$.

4.3 Implicit to explicit representation

Algorithm 3 allows to build the explicit representation given the implicit one. The idea is to distribute the information through the different levels. The principle of algorithm 3 for each dart is:

1. to construct all darts of the different levels;

2. to link them by the successor and predecessor relations;

3. to link the copies to their neighbors for each involution;

4. and then to put the information contained in the initial dart on the dart in the last level in which it exists.

Algorithm 3: Construction of the explicit representation given the implicit one.

```
Input: Implicit representation (whose unique level is G)
Output: Explicit representation (whose different levels are G^0, ..., G^m)
foreach dart d ∈ G do
    lev_d ← last_level_for_dart(d) ;
    for lev ← 0 to lev_d do
        d'^lev ← create_copy(d, G^lev);
        if lev ≠ 0 then
            link_pred_succ(d'^(lev−1), d'^lev) ;
        for i ← 0 to n do
            d_i^lev ← d.α_i^lev ;
            if yet_create_copy(d_i^lev) then
                link_by_α_i(d'^lev, the_copy_of(d_i^lev)) ;
    if is_marked_by_R(d) then
        mark_by_R_i(d'^lev_d, dim_of_rem_cell(d)) ;
    else if is_marked_by_C(d) then
        mark_by_C_i(d'^lev_d, dim_of_cont_cell(d)) ;
```

This algorithm links each dart with its neighbor for each involution at each level. In order to compute the neighbor of a dart for a given involution, we follow the "path" of disappeared darts between a dart and its neighbor. This path is composed of at most k darts, k being the number of disappeared darts in the pyramid. So this algorithm has a cost $\mathcal{O}(pkmn)$. This is the worst case complexity. In fact, this algorithm has a cost $\mathcal{O}(pxmn)$ (with $1 \leq x \leq k \leq p$), with x being the average length of the "paths". But the average length depends on the construction of the pyramid and so it depends on the application.

So this third algorithm is a slightly more costly ($\mathcal{O}(pkmn)$) than the first two ones ($\mathcal{O}(pmn)$). Indeed, since the involutions are explicitly represented in the explicit and hierarchical representations, the neighbor of a given dart for a given involution is directly obtained, while the levels are not explicitly represented in the implicit representation and so the neighbor of a dart for a given involution must be computed.

4.4 The different functions used in the algorithms

There are some functions used by the conversion algorithms which are common to the three representations:

- create_copy creates a copy of a dart. It has a cost $\mathcal{O}(1)$;
- yet_create_copy indicates whether the copy of a dart has been yet created. It has a cost $\mathcal{O}(1)$;
- the_copy_of gives the copy of a dart. It has a cost $\mathcal{O}(1)$;
- link_by_α_i links two darts by an involution. It has a cost $\mathcal{O}(1)$;

- is_marked_by_R (resp. is_marked_by_C) indicates whether a dart is removed (resp. or contracted). It has a cost $\mathcal{O}(1)$;
- mark_by_R_i (resp. mark_by_C_i) marks a disappearing dart by two (in the hierarchical and explicit representations) or three marks (in the implicit representation). It has a cost $\mathcal{O}(1)$;

Other functions are more specific to a particular representation:

- top_most_dart indicates the upper copy of a dart in the explicit representation. This function uses successor links until it reaches a dart with no successor. It has a cost $\mathcal{O}(m)$;
- last_level_for_dart returns the last level of the pyramid at which a dart exists. It has a cost $\mathcal{O}(1)$ in the implicit representation, since the information is contained in the dart. It has a cost $\mathcal{O}(m)$ in the explicit and hierarchical representations, since the information is not contained in the dart: it is necessary to use the successor-predecessor relations until a dart without a successor is reached in the explicit representation, and it is necessary to examine each level in order to find the level at which the dart has no neighbor for an involution in the hierarchical representation;
- link_pred_succ links two darts of two consecutive levels by the successor and predecessor relations in the explicit representation. It has a cost $\mathcal{O}(1)$;
- $d.\alpha_i^{lev}$ gives the neighbor of dart d by α_i at level lev. It has a cost $\mathcal{O}(1)$ in the explicit and hierarchical representations. The neighbors of darts are not explicitly represented at each level in the implicit representation. This function follows the "path" of darts, which disappeared between level 0 and lev, that separate dart d and its neighbor by using the removal and contraction rules (this is similar to the notion of connecting walk [4]). For a given dart, the complexity of this function depends on the length of the "path". This length is bounded by k, the number of darts that disappeared in the pyramid. So it has a cost $\mathcal{O}(k)$.

5 Equivalence of these three representations

As we have said before, the explicit, implicit and hierarchical representations are equivalent. Here, these equivalences are proved.

In order to do that the notion of connecting walk between two any levels is necessary. Such a connecting walk is a sequence of darts in a lower level that separates two darts of an upper level (linked by an involution). Intuitively, a connecting walk at a given level is obtained by concatenating connecting walks of the previous levels concerned by removals or contractions. $Ch_{(i,a,b)}(d)$ denotes the connecting walk between levels a and b ($a \leq b$), for any dimension i and any dart $d \in D^a$ which is neither contracted nor removed in all levels between levels a and b.

Definition 9 (Connecting walk) : *Let $i \in N$, a and b be such that $0 \leq a \leq b \leq m$.*
For each dart $d \in D^a$ which is not contracted or removed between the levels a and b, $Ch_{(i,a,b)}(d)$ is defined by:
if $b = a$: $Ch_{(i,a,b)}(d) = (d.\alpha_i^a)$,
else if $d \notin SD_i^{b-1}$: $Ch_{(i,a,b)}(d) = Ch_{(i,a,b-1)}(d)$,
else: $Ch_{(i,a,b)}(d) = C = \Big(Ch_{(k_1,a,b-1)}(d_1), \ldots, Ch_{(k_p,a,b-1)}(d_p)\Big)$,
where: $d_1 = d$,
$$\forall u,\ 1 \leq u < p,\ d_{u+1} \text{ is the last dart of } Ch_{(k_u,a,b-1)}(d_u),$$
$$\forall u,\ 1 \leq u \leq p,\ k_u = \begin{cases} i \text{ if } u \text{ is odd,} \\ i+1 \text{ if } u \text{ is even and } d_u \in R_i^{b-1}, \\ i-1 \text{ if } u \text{ is even and } d_u \in C_i^{b-1}, \end{cases}$$
and p is the smaller integer such that the last dart of C belongs to SD_i^{b-1}.

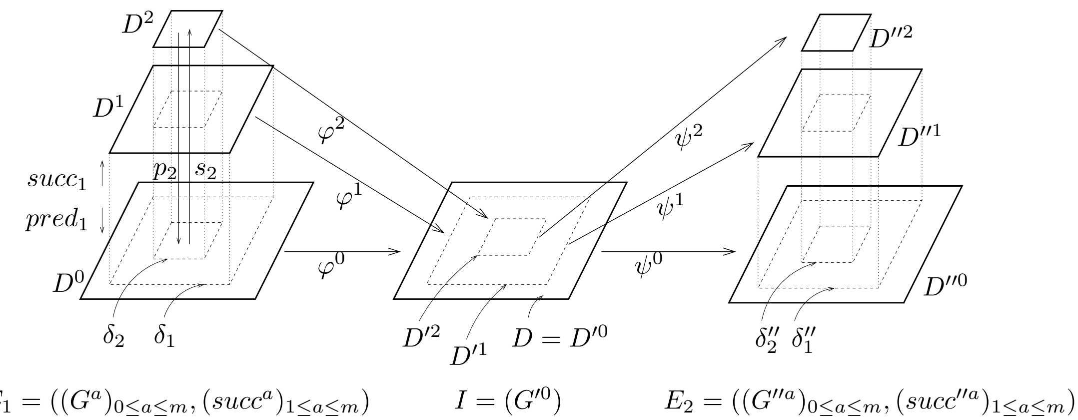


Figure 6: Three representations of a same pyramid. The first one is an explicit representation. The second is an implicit representation and is deduced from the precedent one by applying the algorithm of conversion 4. The third is an explicit representation too and it is deduced from the second one by applying the algorithm 3. The isomorphisms φ^a and ψ^a $(0 \leq a \leq 2)$ are illustrated here.

NB: We note $d.LDC_{(i,a,b)}$ (or $LDC_{(i,a,b)}(d)$) the last dart of the connecting walk $Ch_{(i,a,b)}(d)$.

5.1 Equivalence between the explicit and the implicit representations

In order to prove that the explicit and implicit representations are equivalent:

1. we prove that the representation obtained thanks to algorithm 4 is an implicit one, and that, $\forall a$, $0 \leq a \leq m$, we can construct an isomorphism φ^a between the level a n-G-map of the explicit representation and the level a n-G-map of the implicit representation of a same pyramid;

2. we prove that the representation obtained thanks to algorithm 3 is an explicit one, and that, $\forall a$, $0 \leq a \leq m$, we can construct an isomorphism ψ^a between the level a n-G-map of the implicit representation and the level a n-G-map of the explicit representation of a same pyramid;

3. we prove that, $\forall a$, $0 \leq a \leq m$, $\gamma^a = \varphi^a\psi^a$ is an isomorphism between the level a n-G-map of two explicit representations of a same pyramid (and respects the successor relation). Figure 6 illustrates the notations used in this proof.

Proposition 2 : *Let E be the explicit representation of a pyramid, and I be the representation obtained thanks to Algorithm 4 (cf.apendix) applied to E. The representation I is so an implicit one. And we can construct an isomorphism φ^a between level a of E and level a of I , $\forall a$, $0 \leq a \leq m$.*

Proof:

1. E is the explicit representation of a pyramid, so $E = ((G^a)_{0\leq a\leq m}, (succ^a)_{1\leq a\leq m})$ with:

   - $\forall a$, $0 \leq a \leq m$, $G^a = (D^a, (\alpha_i^a)_{0\leq i\leq n})$;
   - $\forall a$, $0 \leq a < m$, $D^a = S^a \cup (R^a \cup C^a)$;

   and E verifies the preconditions expressed in def 6.

2. definition of s^a:

- by definition, $\forall a, 1 \leq a \leq m$, $succ^a$ is an isomorphism between $(S^{a-1}, (LDC_{(i,a-1,a)})_{1\leq i\leq n})$ and $(D^a, (\alpha_i^a)_{0\leq i\leq n})$;
- Let δ^a be the set of darts of D^0 which are linked by a succession of successor (or predecessor) relations to a dart of D^a. We can expressed δ^a as $\{d \in D^0 | d.s^a \ exists\}$. And we can define δ^a as the image of D^a by $pred^a pred^{a-1} \dots pred^1$.
  Let $s^a = succ^1 succ^2 \dots succ^a$.
  By using the definition of connecting walk we have:

$$\begin{aligned} s^a : \qquad \delta^a(\subseteq D^0) &\longrightarrow D^a \\ d^0 &\longrightarrow d^a \\ d^0.LDC_{(i,0,a)} &\longrightarrow d^a.\alpha_i^a, \ \forall i, 0 \leq i \leq n \end{aligned} , \qquad (1)$$

  So s^a is an isomorphism between $(\delta^a, (LDC_{(i,0,a)})_{0\leq i\leq n})$ and $(D^a, (\alpha_i^a)_{0\leq i\leq n})$.
- moreover we can define $\forall a, 0 \leq a < m$, $p^a = (s^a)^{-1}$ ($p^a = pred^a pred^{a-1} \dots pred^1$). p^a is so an isomorphism between $(D^a, (\alpha_i^a)_{0\leq i\leq n})$ and $(\delta^a, (LDC_{(i,0,a)})_{0\leq i\leq n})$.

3. we construct a representation I starting from E and by using the algorithm 4 page 24. Let $I = \{G'^0\}$ with:

   - $G'^0 = \{D'^0, (\alpha_i'^0)_{0\leq i\leq n}\}$ is an n-G-map;
   - $D'^0 = D'^m \cup \bigcup_{a=0}^{m-1}(R'^a \cup C'^a)$ where $\forall a, 0 \leq a < m$, $R'^a = \bigcup_{i=0}^{n} R_i'^a$ and $C'^a = \bigcup_{i=0}^{n} C_i'^a$;

   In the following, we show that I is an implicit representation, and we construct φ^a for $0 \leq a \leq m$:

   - For each dart d of G^0, this algorithm:
     (a) copies d in G'^0;
     (b) for each involution α_i^0 ($0 \leq i \leq n$), links with $\alpha_i'^0$ the copy of d and the copy of $d.\alpha_i^0$, if this copy already exists;
     (c) by using the successor operator, follows the darts of the higher levels that correspond to the dart until the last one (until to have a dart without successor). If this last dart contains a mark indicating that it is removed or contracted at this level, the algorithm copies this information on the copy of the initial dart, and add an other mark in order to indicate at which level the removal or contraction takes place.
   - In order to prove that I is an implicit representation, we show that we can identify in I all the elements of the definition 8.
   - By using the steps (a) and (b) of the algorithm, we know that each dart of level 0 is copied and at the end all links between the darts of this level are copied too. So it exists an isomorphism φ^0 between G^0 and G'^0.
   - According to step (c) of the algorithm, if the last successor of a dart contains a mark indicating it is contracted or removed, the copy of the dart is marked too. So, if $d \in D^0$ is such that $\exists a$ and i ($0 \leq a < m$ and $0 \leq i \leq n$), $d.s^a \in R_i^a$ (resp. $d.s^a \in C_i^a$) we have: $d.\varphi^0 \in R_i'^a$ (resp. $d.\varphi^0 \in C_i'^a$).
     So $\forall i, 0 \leq i \leq n$ and $\forall a, 0 \leq a < m$

$$\begin{aligned} R_i'^a &= (\delta^a)_{/R}\varphi^0 \quad \text{with } (\delta^a)_{/R} = \{d \in D^0 | d.s^a \in R_i^a\} \\ C_i'^a &= (\delta^a)_{/C}\varphi^0 \quad \text{with } (\delta^a)_{/C} = \{d \in D^0 | d.s^a \in C_i^a\} \end{aligned} \qquad (2)$$

     And so $\forall a, 0 \leq a < m$:
     $\varphi^0 : \{d \in D^0 | \forall c, 0 \leq c < a, d.s^c \notin \bigcup_{i=0}^{n}(R_i^c \cup C_i^c)\} \longrightarrow \{d' \in D'^0 | \forall c, 0 \leq c < a, d' \notin$

$\bigcup_{i=0}^{n}(R_i'^c \cup C_i'^c)\}$
this corresponds to $\forall a$, $0 \leq a < m$: $D'^a = \delta^a.\varphi^0$;

- let $\varphi^a = p^a.\varphi^0$, $\forall a$, $0 < a \leq m$. φ^a is an isomorphism between G^a and G'^a; since $\forall i$, $0 \leq i \leq n$ and $\forall a$, $0 \leq a < m$:

$$\begin{aligned} R_i'^a &= R_i^a.\varphi^a \\ C_i'^a &= C_i^a.\varphi^a \end{aligned}. \tag{3}$$

  it exists a one to one mapping between $R_i'^a$ and R_i^a (resp. between $C_i'^a$ and C_i^a), and so between R'^a and R^a (resp. between C'^a and C^a)

- since the preconditions concerning the sets R^a and C^a of the def 6 are verified, R'^a and C'^a respect the preconditions of the def 8. So representation I obtained by the algorithm is an implicit one.

□

Proposition 3 : *Let I be the implicit representation of a pyramid, and E be the representation obtained thanks to Algorithm 3 (cf. previous section) applied to I. The representation E is so an explicit one. And we can construct an isomorphism ψ^a between level a of I and level a of E , $\forall a$, $0 \leq a \leq m$.*

Proof:

1. I is the implicit representation of a pyramid, so $I = (G^0)$ with:
   - $G^0 = (D^0, (\alpha_i^0)_{0 \leq i \leq n})$ is an n-G-map;
   - $D^0 = D^m \cup \bigcup_{a=0}^{m-1}(R^a \cup C^a)$ where $\forall a$, $0 \leq a < m$, $R^a = \bigcup_{i=0}^{n} R_i^a$, and $C^a = \bigcup_{i=0}^{n} C_i^a$;

   and I verifies the preconditions expressed in def 8.

2. we construct a representation E starting from I and by using the algorithm 3 page 13. Let $E = ((G'^a)_{0 \leq a \leq m}, (succ'^a)_{1 \leq a \leq m})$ with $\forall a$, $0 \leq a \leq m$, $G'^a = (D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$. In the following, we show that E is an explicit representation, and we construct ψ^a for $0 \leq a \leq m$:
   - For each dart d, this algorithm:
     (a) copies d in representation E at each level where it implicitly exists. So if d implicitly exists in the first $c+1$ levels (is in $D^0, \ldots, D^c$), a copy is created at each level ($D'^0 \ldots D'^c$) in explicit representation E;
     (b) all copies of d are linked by the predecessor and successor relations;
     (c) for each level c ($0 \leq c \leq$ last level d) and each α_i^c ($0 \leq i \leq n$), the copy of d and the copy of the neighbor of d are linked by $\alpha_i'^c$, if this last copy already exists;
     (d) If d contains a mark indicating that it is removed or contracted at a level, the algorithm copies this information on the copy of d located at the level indicating by the mark.
   - In order to prove that E is an explicit representation, we show that we can identify in E all the elements of the definition 6.
   - By using the steps (a) and (c) of the algorithm, we show that it exists an isomorphism ψ^0 between G^0 and G'^0.

- Moreover, by using step (b), we can deduce that $\forall a$, $1 \leq a \leq m$, $succ'^a$ is an isomorphism between $\left(D'^{a-1} - \bigcup_{i=0}^{n}(R_i'^{a-1} \cup C_i'^{a-1}), (LDC'_{(i,a-1,a)})_{0 \leq i \leq n}\right)$ and $(D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$.

  Let $s'^a = succ'^1 succ'^2 \ldots succ'^a$, and $\delta'^a = D'^0 - \bigcup_{i=0}^{n} \bigcup_{c=0}^{a-1}(R_i'^c \cup C_i'^c)$, $\forall a$, $1 \leq a \leq m$. s'^a is so an isomorphism between $(\delta'^a, (LDC'_{(i,0,a)})_{0 \leq i \leq n})$ and $(D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$.

- According to step (d), we have $\forall a$, $0 \leq a < m$, and $\forall i$, $0 \leq i \leq n$, $R_i'^a = R_i^a.\psi^0.s^a$ and $C_i'^a = C_i^a.\psi^0.s^a$. So $\forall a$, $0 \leq a < m$, $R'^a = R^a.\psi^0.s^a$ and $C'^a = C^a.\psi^0.s^a$. In consequence, we have a one to one mapping between $R_i'^a$ and R_i^a (resp. $C_i'^a$ and C_i^a), and R'^a and R^a (resp. C'^a and C^a). So R'^a, C'^a, $R_i'^a$ and $C_i'^a$ $(0 \leq i \leq n)$ respect the preconditions of the definition 6.

3. Let $\psi^a = \psi^0.s'^a$, $\forall a$, $1 \leq a \leq m$. ψ^a is so an isomorphism between G^a and G'^a. Moreover, we can prove that $G'^a = (D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$ is an n-G-map.

So representation E obtained by the algorithm is an explicit one.

□

Proposition 4 : *Let two explicit representations E_1 and E_2 (of a same pyramid) where E_2 is obtained by using algorithms 4 and then 3 starting from E_1. We can construct an isomorphism γ^a between level a of E_1 and level a of E_2 (which respects the successor relation), $\forall a$, $0 \leq a \leq m$.*

Proof:

1. let I be the implicit representation obtained by using algo 4 starting from E_1.
   Since E_1 is an explicit representation we have $E_1 = ((G^a)_{0 \leq a \leq m}, (succ^a)_{1 \leq a \leq m})$ with:

   - $\forall a$, $0 \leq a \leq m$, $G^a = (D^a, (\alpha_i^a)_{0 \leq i \leq n})$;
   - $\forall a$, $0 \leq a < m$, $D^a = S^a \cup (R^a \cup C^a)$;

   and E_1 verifies the preconditions expressed in def 6.
   According to proposition 2 we know that I is an implicit representation and we have: $I = (G'^0)$ with:

   - $G'^0 = (D'^0, (\alpha_i'^0)_{0 \leq i \leq n})$ is an n-G-map;
   - $D'^0 = D'^m \cup \bigcup_{a=0}^{m-1}(R'^a \cup C'^a)$ where $\forall a$, $0 \leq a < m$, $R'^a = \bigcup_{i=0}^{n} R_i'^a$, and $C'^a = \bigcup_{i=0}^{n} C_i'^a$;

   and I verifies the preconditions expressed in def 8.

   According to proposition 3, we know that E_2 is an explicit representation and we have: $E_2 = ((G''^a)_{0 \leq a \leq m}, (succ''^a)_{1 \leq a \leq m})$ with:

   - $\forall a$, $0 \leq a \leq m$, $G''^a = (D''^a, (\alpha_i''^a)_{0 \leq i \leq n})$;
   - $\forall a$, $0 \leq a < m$, $D''^a = S''^a \cup (R''^a \cup C''^a)$;

   and E_2 verifies the preconditions expressed in def 6.

   According to proposition 2 (resp proposition 3 we know that $\forall a$, $0 \leq a \leq m$, it exists an isomorphism φ^a (resp. ψ^a) between G^a and G'^a (resp. G'^a and G''^a).

   Let $\gamma^a = \varphi^a.\psi^a$, $\forall a$, $0 \leq a \leq m$. γ^a is an isomorphism between G^a and G''^a, $\forall a$, $0 \leq a \leq m$.

2. we verify that $succ^a.\gamma^a = \gamma^{(a-1)}.succ''^a$:
   let $d^{(a-1)} \in D^{(a-1)}$ a dart of E_1 such as $d^a = d^{(a-1)}.succ^a$ exists.
   We have $d^{(a-1)}.\varphi^{(a-1)} = d^a.\varphi^a = d'$ and $d' \in D'^a \subseteq D'^{(a-1)}$ (since in the implicite representation, the set of darts wich represents a level is a subset of dart set of the previous levels).
   In the same way, since $d' \in D'^a \subseteq D'^{(a-1)}$, we have $d'.\psi^{(a-1)} = d''^{(a-1)}$ and $d'.\psi^a = d''^a$ with $d''^a = d''^{(a-1)}.succ''^a$.
   So $d^a.\gamma^a = d''^a$ and $d^{(a-1)}.\gamma^{(a-1)} = d''^{(a-1)}$.
   And in consequence $d^{(a-1)}.succ^a.\gamma^a = d^{(a-1)}.\gamma^{(a-1)}.succ''^a$.
   So $succ^a.\gamma^a = \gamma^{(a-1)}.succ''^a$.

So we can construct an isomorphism γ^a between level a of E_1 and level a of E_2 (which respects the successor relation), $\forall a,\ 0 \leq a \leq m$.

□

So the explicit and implicit representations are equivalent.

5.2 Equivalence between the explicit and the hierarchical representations

In this subsection we apply the same methode as in the previous one.
In order to prove that the explicit and hierarchical representations are equivalent:

1. we prove that the representation obtained thanks to algorithm 1 is a hierarchical one, and that, $\forall a,\ 0 \leq a \leq m$, we can construct an isomorphism φ^a between the level a n-G-map of the explicit representation and the level a n-G-map of the hierarchical representation of a same pyramid.
2. we prove that the representation obtained thanks to algorithm 5 is an explicit one, and that, $\forall a,\ 0 \leq a \leq m$, we can construct an isomorphism ψ^a between the level a n-G-map of the hierarchical representation and the level a n-G-map of the explicit representation of a same pyramid.
3. we prove that, $\forall a,\ 0 \leq a \leq m$, $\gamma^a = \varphi^a.\psi^a$ is an isomorphism between the level a n-G-maps of two explicit representations of a same pyramid (and respects the successor relation).

Proposition 5 : *Let E be the explicit representation of a pyramid, and H be the representation obtained thanks to algorithm 1 applied to E. The representation H is so a hierarchical one. And we can construct an isomorphism φ^a between level a of E and level a of H , $\forall a,\ 0 \leq a \leq m$.*

Proof:

1. E is the explicit representation of a pyramid, so $E = ((G^a)_{0 \leq a \leq m}, (succ^a)_{1 \leq a \leq m})$ with:
   - $\forall a,\ 0 \leq a \leq m,\ G^a = (D^a, (\alpha_i^a)_{0 \leq i \leq n})$;
   - $\forall a,\ 0 \leq a < m,\ D^a = S^a \cup (R^a \cup C^a)$;

   and E verifies the preconditions expressed in def 6.
2. definition of s^a:
   - by definition, $\forall a,\ 1 \leq a \leq m$, $succ^a$ is an isomorphism between $(S^{a-1}, (LDC_{(i,a-1,a)})_{0 \leq i \leq n})$ and $(D^a, (\alpha_i^a)_{0 \leq i \leq n})$;

- Let δ^a be the set of darts of D^0 which are linked by a succession of successor relations to a dart of D^a. We can expressed δ^a as $\{d \in D^0 | d.s^a \ exists\}$. And we can define δ^a as the image of D^a by $pred^a pred^{a-1} \dots pred^1$.
Let $s^a = succ^1 succ^2 \dots succ^a$.
By using the definition of connecting walk we have:

$$\begin{array}{rrcl} s^a: & \delta^a(\subseteq D^0) & \longrightarrow & D^a \\ & d^0 & \longrightarrow & d^a \\ & d^0.LDC_{(i,0,a)} & \longrightarrow & d^a\alpha_i^a, \ \forall i, 0 \leq i \leq n \end{array}, \tag{4}$$

So s^a is an isomorphism between $(\delta^a, (LDC_{(i,0,a)})_{0\leq i\leq n})$ and $(D^a, (\alpha_i^a)_{0\leq i\leq n})$.
- moreover we can define $\forall a$, $0 \leq a < m$, $p^a = (s^a)^{-1}$ ($p^a = pred^a pred^{a-1} \dots pred^1$). p^a is so an isomorphism between $(D^a, (\alpha_i^a)_{0\leq i\leq n})$ and $(\delta^a, (LDC_{(i,0,a)})_{0\leq i\leq n})$.

3. we construct a representation H starting from E and by using the algorithm 1 page 12. Let $H = \{D', (\alpha_i'^a)_{0\leq i\leq n, 0\leq a\leq m}\}$ with

   - $D' = D'^m \cup \bigcup_{a=0}^{m-1}(R'^a \cup C'^a)$;
   - $\forall a$, $0 \leq a < m$, $D'^{a+1} = D' - \bigcup_{k=0}^{a}(R'^k \cup C'^k)$ and $D'^0 = D'$,
   $R'^a = \bigcup_{i=0}^{n} R_i'^a$ and $C'^a = \bigcup_{i=0}^{n} C_i'^a$;

   In the following, we show that H is a hierarchical representation, and we construct φ^a for $0 \leq a \leq m$:

   - For each dart d of G^0, this algorithm:
     (a) copies d in D';
     (b) for each level c ($0 \leq c \leq$ last level for d) and each α_i^c ($0 \leq i \leq n$), it links with $\alpha_i'^c$ the copy of d and the copy of $d.\alpha_i^c$, if this last copy already exists;
     (c) by using the successor operator, follows the darts of the higher levels that correspond to d until the last one (until to have a dart without successor). If this last dart contains a mark indicating that it is removed or contracted at this level, the algorithm copies this information on the copy of d.
   - In order to prove that H is a hierarchical representation, we show that we can identify in H all the elements of the definition 7.
   - By using the steps (a) and (b) of the agorithm, we know that each dart of level 0 is copied and at the end all links between the darts of this level are copied too. So it exists an isomorphism φ^0 between G^0 and G'^0.
   - According to step (c) of the algorithm, if the last successor of a dart contains a mark indicating it is contracted or removed, the copy of the dart is marked too. So, if $d \in D^0$ is such that $\exists a$ and i ($0 \leq a < m$ and $0 \leq i \leq n$), $d.s^a \in R_i^a$ (resp. $d.s^a \in C_i^a$) we have: $d.\varphi^0 \in R_i'^a$ (resp. $d.\varphi^0 \in C_i'^a$).
   So $\forall i$, $0 \leq i \leq n$ and $\forall a$, $0 \leq a < m$

$$\begin{array}{ll} R_i'^a = (\delta^a)_{/R}\varphi^0 & \text{with } (\delta^a)_{/R} = \{d \in D^0 | d.s^a \in R_i^a\} \\ C_i'^a = (\delta^a)_{/C}\varphi^0 & \text{with } (\delta^a)_{/C} = \{d \in D^0 | d.s^a \in C_i^a\} \end{array} \tag{5}$$

   And so $\forall a$, $0 \leq a < m$:
   $\varphi^0 : \{d \in D^0 | \forall c, 0 \leq c < a, d.s^c \notin \bigcup_{i=0}^{n}(R_i^c \cup C_i^c)\} \longrightarrow \{d' \in D'^0 | \forall c, 0 \leq c < a, d' \notin \bigcup_{i=0}^{n}(R_i'^c \cup C_i'^c)\}$
   this corresponds to $\forall a$, $0 \leq a < m$: $D'^a = \delta^a.\varphi^0$;

- let $\varphi^a = p^a.\varphi^0$, $\forall a$, $0 < a \leq m$. φ^a is an isomorphism between G^a and G'^a; since $\forall i$, $0 \leq i \leq n$ and $\forall a$, $0 \leq a < m$:

$$\begin{array}{l} R_i'^a = R_i^a.\varphi^a \\ C_i'^a = C_i^a.\varphi^a \end{array}. \tag{6}$$

  it exists a one to one mapping between $R_i'^a$ and R_i^a (resp. between $C_i'^a$ and C_i^a), and so between R'^a and R^a (resp. between C'^a and C^a)

- since the preconditions concerning the sets R^a and C^a of the def 6 are verified, R'^a and C'^a respect the preconditions of the def 7. So representation H obtained by the algorithm is a hierarchical one.

□

Proposition 6 : *Let H be the hierarchical representation of a pyramid, and E be the representation obtained thanks to algorithm 5 applied to H. The representation E is so an explicit one. And we can construct an isomorphism ψ^a between level a of H and level a of E , $\forall a$, $0 \leq a \leq m$.*

Proof:

1. H is the hierarchical representation of a pyramid, so $H = \{D, (\alpha_i^a)_{0 \leq i \leq n, 0 \leq a \leq m}\}$ with

   - $D = D^m \cup \bigcup_{a=0}^{m-1}(R^a \cup C^a)$;
   - $\forall a$, $0 \leq a < m$, $D^{a+1} = D - \bigcup_{k=0}^{a}(R^k \cup C^k)$ and $D^0 = D$, $R^a = \bigcup_{i=0}^{n} R_i^a$ and $C^a = \bigcup_{i=0}^{n} C_i^a$;

   and such that the preconditions expressed in def 7 are verified;

2. we construct a representation E starting from H and by using the algorithm 3 page 13. Let $E = ((G'^a)_{0 \leq a \leq m}, (succ'^a)_{1 \leq a \leq m})$ with $\forall a$, $0 \leq a \leq m$, $G'^a = (D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$. In the following, we show that E is an explicit representation, and we construct ψ^a for $0 \leq a \leq m$:

   - For each dart d, this algorithm:
     (a) copies d in the representation E at each level where it implicitly exists. So if d implicitly exists in the first $c + 1$ levels (is in $D^0, \ldots, D^c$), a copy of d is created at each level $(G'^0 \ldots G'^c)$ in the explicit representation E;
     (b) all copies of d are linked by the predecessor and successor relations;
     (c) for each level c ($0 \leq c \leq$ last level for d) and each α_i^c ($0 \leq i \leq n$), it links with $\alpha_i'^c$ the copy of d and the copy of the neighbor of d, if this last copy already exists;
     (d) If d contains a mark indicating that it is removed or contracted at a level, the algorithm copies this information on the copy of d located at the highest level .

   - In order to prove that E is an explicit representation, we show that we can identify in E all the elements of the definition 6.
   - By using the steps (a) and (c) of the agorithm, we show that it exists an isomorphism ψ^0 between G^0 and G'^0.

   - Moreover, by using step (b), we can deduce that $\forall a$, $1 \leq a \leq m$, $succ'^a$ is an isomorphism between $\left(D'^{a-1} - \bigcup_{i=0}^{n}(R_i'^{a-1} \cup C_i'^{a-1}), (LDC'_{(i,a-1,a)})_{0 \leq i \leq n}\right)$ and $(D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$.

     Let $s'^a = succ'^1 succ'^2 \ldots succ'^a$, and $\delta'^a = D'^0 - \bigcup_{i=0}^{n} \bigcup_{c=0}^{a-1}(R_i'^c \cup C_i'^c)$, $\forall a$, $1 \leq a \leq m$. s'^a is so an isomorphism between $(\delta'^a, (LDC'_{(i,0,a)})_{0 \leq i \leq n})$ and $(D'^a, (\alpha_i'^a)_{0 \leq i \leq n})$.

- According to step (d), we have $\forall a,\ 0 \leq a < m$, and $\forall i,\ 0 \leq i \leq n$, $R'^a_i = R^a_i.\psi^0.s^a$ and $C'^a_i = C^a_i.\psi^0.s^a$. So $\forall a,\ 0 \leq a < m$, $R'^a = R^a.\psi^0.s^a$ and $C'^a = C^a.\psi^0.s^a$. In consequence, we have a one to one mapping between R'^a_i and R^a_i (resp. C'^a_i and C^a_i), and R'^a and R^a (resp. C'^a and C^a). So R'^a, C'^a, R'^a_i and C'^a_i $(0 \leq i \leq n)$ respect the preconditions of the definition 6.

3. Let $\psi^a = \psi^0.s'^a$, $\forall a,\ 1 \leq a \leq m$. ψ^a is so an isomorphism between G^a and G'^a. Moreover, we can prove that $G'^a = (D'^a, (\alpha'^a_i)_{0 \leq i \leq n})$ is an n-G-map.

So representation E obtained by the algorithm is an explicit one.

□

Proposition 7 : *Let two explicit representations E_1 and E_2 (of a same pyramid) where E_2 is obtained by using algorithms 1 and then 5 starting from E_1. We can construct an isomorphism γ^a between level a of E_1 and level a of E_2 (which respects the successor relation), $\forall a,\ 0 \leq a \leq m$.*

Proof:

1. let H be the hierarchical representation obtained by using algorithm 1 starting from E_1. Since E_1 is an explicit representation we have $E_1 = ((G^a)_{0 \leq a \leq m}, (succ^a)_{1 \leq a \leq m})$ with:

   - $\forall a,\ 0 \leq a \leq m$, $G^a = (D^a, (\alpha^a_i)_{0 \leq i \leq n})$;
   - $\forall a,\ 0 \leq a < m$, $D^a = S^a \cup (R^a \cup C^a)$;

   and E_1 verifies the preconditions expressed in def 6.

   According to proposition 5 we know that H is a hierarchical representation and we have: $H = \{D', (\alpha'^a_i)_{0 \leq i \leq n, 0 \leq a \leq m}\}$ with

   - $D' = D'^m \cup \bigcup_{a=0}^{m-1} (R'^a \cup C'^a)$;
   - $\forall a,\ 0 \leq a < m$, $D'^{a+1} = D' - \bigcup_{k=0}^{a} (R'^k \cup C'^k)$ and $D'^0 = D$,
     $R'^a = \bigcup_{i=0}^{n} R'^a_i$ and $C'^a = \bigcup_{i=0}^{n} C'^a_i$;

   and such that the preconditions expressed in def 7 are verified.

   According to proposition 6, we know that E_2 is an explicit representation and we have: $E_2 = ((G''^a)_{0 \leq a \leq m}, (succ''^a)_{1 \leq a \leq m})$ with:

   - $\forall a,\ 0 \leq a \leq m$, $G''^a = (D''^a, (\alpha''^a_i)_{0 \leq i \leq n})$;
   - $\forall a,\ 0 \leq a < m$, $D''^a = S''^a \cup (R''^a \cup C''^a)$;

   and E_2 verifies the preconditions expressed in def 6.

   According to proposition 5 (resp proposition 6 we know that $\forall a,\ 0 \leq a \leq m$, it exists an isomorphism φ^a (resp. ψ^a) between G^a and G'^a (resp. G'^a and G''^a).

   Let $\gamma^a = \varphi^a.\psi^a$, $\forall a,\ 0 \leq a \leq m$. γ^a is an isomorphism between G^a and G''^a, $\forall a,\ 0 \leq a \leq m$.

2. we verify that $succ^a.\gamma^a = \gamma^{(a-1)}.succ''^a$:
   let $d^{(a-1)} \in D^{(a-1)}$ a dart of E_1 such as $d^a = d^{(a-1)}.succ^a$ exists.
   We have $d^{(a-1)}.\varphi^{(a-1)} = d^a.\varphi^a = d'$ and $d' \in D'^a \subseteq D'^{(a-1)}$.
   In the same way, since $d' \in D'^a \subseteq D'^{(a-1)}$, we have $d'.\psi^{(a-1)} = d''^{(a-1)}$ and $d'.\psi^a = d''^a$ with $d''^a = d''^{(a-1)}.succ''^a$.
   So $d^a.\gamma^a = d''^a$ and $d^{(a-1)}.\gamma^{(a-1)} = d''^{(a-1)}$.
   And in consequence $d^{(a-1)}.succ^a.\gamma^a = d^{(a-1)}.\gamma^{(a-1)}.succ''^a$.
   So $succ^a.\gamma^a = \gamma^{(a-1)}.succ''^a$.

So we can construct an isomorphism γ^a between level a of E_1 and level a of E_2 (which respects the successor relation), $\forall a$, $0 \leq a \leq m$.

□

So the explicit and hierarchical representations are equivalent.

6 Conclusion and Perspectives

n-G-map pyramids have several advantages. Mainly, n-G-maps pyramids describe the topological information about n-dimensional multi-level subdivided objects and their definition is homogeneous for any dimension. The main drawback is the fact that n-G-map pyramids can be very expensive in memory space.

n-G-map pyramids can be represented in different ways. We have defined here three generic representations: *explicit*, *hierarchical* and *implicit* and proposed different conversion algorithms. We have presented advantages and drawbacks of these representations. This is particularly important in order to choose an efficient representation according to the needs of an application (complexity in memory space and/or in time). Moreover we have shown that these three representations are equivalent.

Now we are conceiving operations for handling this structure. A common problem is the propagation of modifications from a level to other ones. For instance, if we add a new cell within a given level, it is necessary to propagate this modification to the bottom of the pyramid. But if we remove or contract a cell, we have to propagate this modification to the top.

We intend to study the use of n-G-map pyramids for 3D and 4D image processing, leading to the study of more specific operations. Our goal is to develop a computer software based on n-G-map pyramids which groups many functionalities: multi-level image segmentation, modification of a given region at a particular level by an expert, extraction of topological and geometrical characteristics…

7 Appendix

We have proposed in section 3 three algorithms allowing to move from one representation to another (explicit → hierarchical → implicit → explicit). In order to have the algorithms allowing to convert diectly any of the three proposed representations into another one, we give here the three other conversion algorithms (explicit → implicit → hierarchical → explicit).

7.1 Explicit to implicit representation

Algorithm 4 builds the implicit representation given the explicit one. The principle of this algorithm is, for each dart of level 0 :

1. to create a copy;
2. to copy the level 0 involutions, i.e. to link the copy of the dart to its neighbor for each involution (α_i with $i \in N$) if it already exists;
3. then to put on the copy the information contained in the initial dart: the mark if the dart belongs to a removed or a contracted cell, the dimension of the cell, and the level at which the cell disappears (this last mark is not a copy, it is computed).

This algorithm links each dart of the implicit representation to its neighbor for each involution. The cost to compute the last level at which a dart exists is $\mathcal{O}(m)$. So this algorithm has a cost $\Theta(p(m+n))$, p being the number of darts of the first pyramid level map, n the dimension of the G-map and m the number of pyramid levels.

Algorithm 4: Construction of the implicit representation given the explicit one.

```
Input: Explicit representation (whose different levels are G^0, ..., G^m)
Output: Implicit representation (whose unique level is G)
foreach dart d ∈ G^0 do
    d' ← create_copy(d, G) ;
    for i ← 0 to n do
        if yet_create_copy(dα_i^0) then
            link_by_α_i(d', the_copy_of(dα_i^0)) ;
    d^{lev_d} ← dart_of_last_level_for_dart(d) ;
    if is_marked_by_R(d^{lev_d}) then
        mark_by_R_i(d',dim_of_rem_cell(d^{lev_d}),lev_d) ;
    else if is_marked_by_C(d^{lev_d}) then
        mark_by_C_i(d',dim_of_con_cell(d^{lev_d}),lev_d) ;
```

7.2 Hierarchical to explicit representation

Algorithm 5 builds the explicit representation given the hierarchical one. The principle of this algorithm is the same than the one of algo. 3. For each dart:

1. to construct all darts of the different levels;
2. to link them by the successor and predecessor relations;
3. to link the copies to their neighbors for each involution;
4. and then to put the information contained in the initial dart on the dart in the last level in which it exists.

Algorithm 5: Construction of the explicit representation given the hierarchical one.

```
Input: Hierarchical representation (whose unique set of darts is D')
Output: Explicit representation (whose different levels are G^0, ..., G^m)
foreach dart d ∈ D' do
    lev_d ← last_level_for_dart(d) ;
    for lev ← 0 to lev_d do
        d'^{lev} ← create_copy(d,G^{lev}) ;
        for i ← 0 to n do
            if yet_create_copy(dα_i^{lev}) then
                link_by_α_i(d', the_copy_of(dα_i^{lev})) ;
    if is_marked_by_R(d, lev_d) then
        mark_by_R_i(d'^{lev_d},dim_of_rem_cell(d,lev_d));
    else if is_marked_by_C(d, lev_d) then
        mark_by_C_i(d'^{lev_d},dim_of_con_cell(d,lev_d)) ;
```

This algorithm links each dart to its neighbor for each involution at each level. It computes the last level in which a dart exists by looking if it has a neighbor for an involution in at most m levels. So this algorithm has a cost $\Theta(pmn)$, p being the number of darts of the first pyramid level map, n the dimension of the G-map and m the number of pyramid levels.

7.3 Implicit to hierarchical representation

Algorithm 6 builds the hierarchical representation given the implicit one. The principle of this algorithm is the same than the one of Algorithm 1. For each dart:

1. To create a copy,
2. to link the copy of the dart to its neighbor for each involution (α_i with $i \in N$) and each level if it already exists;
3. then to put on the copy the information contained in the initial dart: the mark if the dart belongs to a removed or a contracted cell and the dimension of the cell.

Algorithm 6: Construction of the hierarchical representation given the implicit one.

Input: Implicit representation (whose unique level is G)
Output: Hierarchical representation (whose unique set of darts is D')

```
foreach dart d ∈ G do
    lev_d ← last_level_for_dart(d) ;
    d' ← create_copy(d, D') ;
    for lev ← 0 to lev_d do
        for i ← 0 to n do
            d_i^lev ← d.α_i^{lev_d} ;
            if yet_create_copy(d_i^lev) then
                link_by_α_i(d',the_copy_of(d_i^lev)) ;
    if is_marked_by_R(d, lev_d) then
        mark_by_R_i(d'^{lev_d}, dim_of_rem_cell(d)) ;
    else if is_marked_by_C(d, lev_d) then
        mark_by_C_i(d'^{lev_d}, dim_of_cont_cell(d)) ;
```

Algorithm 6 links each dart to its neighbor for each involution at each level. In the same way as in Algorithm 3, in order to compute the neighbor of a dart for a given involution, we follow the "path" of disappeared darts between a dart and its neighbor. This path is composed by at most k darts, k being the number of disappeared darts in the pyramid. So this algorithm has a cost $\mathcal{O}(pkmn)$, p being the number of darts of the first pyramid level, n the dimension of the G-map and m the number of pyramid levels. This is the worst case complexity. In fact, this algorithm has a cost $\mathcal{O}(pxmn)$ (with $1 \leq x \leq k \leq p$), with x being the average length of the "paths". But the average length depends on the construction of the pyramid and so it depends on the application.

References

[1] C. Grasset-Simon, G. Damiand, P. Lienhardt, Pyramides de cartes généralisées, chemins de connection et orbites généralisées, Tech. Rep. 2, SIC, Université de Poitiers, http://www.sic.sp2mi.univ-poitiers.fr/grasset (2005).

[2] P. Lienhardt, N-dimensional generalized combinatorial maps and cellular quasi-manifolds, in: International Journal of Computational Geometry and Applications, Vol. 4, Hamburg, Germany, 1994, pp. 275–324.

[3] G. Damiand, M. Dexet-Guiard, P. Lienhardt, E. Andres, Removal and contraction operations to define combinatorial pyramids: application to the design of a spatial modeler, Image and Vision Computing 23 (2) (2005) 259–269.

[4] C. Grasset-Simon, G. Damiand, P. Lienhardt, Receptive fields for generalized map pyramids: the notion of generalized orbit, in: Discrete Geometry for Computer Imagery, no. 3429 in LNCS, Poitiers, France, 2005, pp. 56–67.

[5] P. Lienhardt, Topological models for boundary representation: a comparison with n-dimensional generalized maps, Computer-Aided Design 23 (1) (1991) 59–82.

[6] E. Brisson, Representing geometric structures in d dimensions: topology and order, Discrete Comput. Geom. 9 (1) (1993) 387–426.

[7] L. Brun, W. Kropatsch, Implicit encoding of combinatorial pyramids, in: O. Drbohlav (Ed.), Proceedings of the Computer Vision Winter Workshop, Valtice, Czech Republic, 2003, pp. 49–54.

[8] D. Fradin, Modélisation et simulation d'éclairage à base topologique: application aux environnements architecturaux complexes, Phd thesis, Université de Poitiers, France (December 2004).