

# Introduction to Combinatorial Pyramids

Luc Brun<sup>†</sup> and Walter Kropatsch<sup>\*‡</sup>

<sup>†</sup> Laboratoire d'Études et de Recherche en Informatique(EA 2618)  
Université de Reims - France

and

<sup>‡</sup> Institute for Computer-aided Automation  
Pattern Recognition and Image Processing Group  
Vienna Univ. of Technology- Austria

<sup>†</sup> brun@leri.univ-reims.fr, <sup>‡</sup> krw@prip.tuwien.ac.at

**Abstract.** A pyramid is a stack of image representations with decreasing resolution. Many image processing algorithms run on this hierarchical structure in  $\mathcal{O}(\log(n))$  parallel processing steps where  $n$  is the diameter of the input image. Graph pyramids are made of a stack of successively reduced graphs embedded in the plane. Such pyramids overcome the main limitations of their regular ancestors. The graphs used in the pyramid may be region adjacency graphs or dual graphs. This paper reviews the different hierarchical data structures and introduces a new representation named combinatorial pyramid.

## 1 Introduction

Regular image pyramids have been introduced in 1981/82 [10] as a stack of images with decreasing resolutions. Such pyramids present interesting properties for image processing and analysis such as [4]: The reduction of noise, the processing of local and global features within the same frame, the use of local processes to detect global features at low resolution and the efficiency of many computations on this structure. Regular pyramids are usually made of a very limited set of levels (typically  $\log(n)$  where  $n$  is the diameter of the input image). Therefore, if each level is constructed in parallel from the level below, the whole pyramid may be built in  $\mathcal{O}(\log(n))$  steps. Since 1981, regular pyramids have been widely used in image segmentation, shape analysis, surface reconstruction and motion analysis (see e.g. [25] for a survey of these applications). However, the rigidity of regular pyramids induces several drawbacks such as the shift-dependence problem and the limited number of regions encoded at a given level of the pyramid [4]. Irregular pyramids overcome these negative properties while keeping the main advantages of their regular ancestors [23]. These pyramids are defined as a stack of successively reduced graphs. Each graph is built from the graph below by selecting a set of vertices named *surviving vertices* and mapping each

---

\* Thanks to L. Lucas and M. Mokhtari for proofreading. This Work was supported by the Austrian Science Foundation under P14445-MAT.

non surviving vertex to a surviving one [23]. Typically, such graphs represent the pixel neighborhood, the region adjacency, or the semantical context of image objects. Since images are samplings of a plane, the graphs are embedded in the plane by definition. Hence they are planar and the dual graph is well defined.

Two kinds of graphs are usually used within the pyramid framework: the simple graph data structure encodes each relation between two vertices by a single edge. The dual graph data structure is based on an explicit encoding of both the initial graph and its dual.

*Combinatorial map* first introduced in 1960 by Edmonds [12] may be considered as a planar graph with an explicit encoding of the neighborhood's orientation of each vertex. This graph encoding induces several advantages besides the other graph representations such as the explicit encoding of the orientation of the plane or an implicit encoding of the dual graph.

This paper presents the usual hierarchical data structures in Section 2. Then the combinatorial maps are presented together with their main properties in Section 3. A combination of the combinatorial and Irregular Pyramid frameworks is then presented in Section 4.

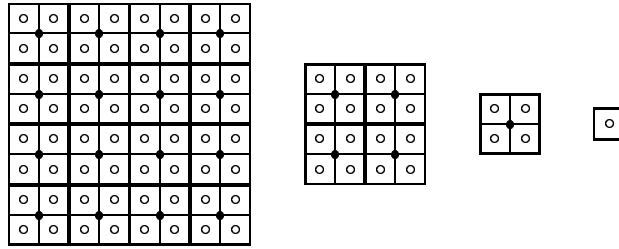
## 2 Hierarchical Data Structures

### 2.1 Regular Pyramids

A regular pyramid is defined as a sequence of images with exponentially reduced resolution. Each image of this sequence is called a *level* of the pyramid. The lowest level corresponds to the original image while the highest one corresponds to the weighed average of the original image. Using the neighborhood relationships defined on each image the *Reduction window* relates each pixel of the pyramid with a set of pixels defined in the level below. The pixels belonging to a reduction window are the *children* of the pixel which defines it. Such a pixel is the *father* (filled circles in Fig. 1) of the pixels belonging to its reduction window. This father-child relationship maybe extended by transitivity to any level of the pyramid. The set of children of one pixel in the base level image is named its *receptive field*. Within the regular pyramid framework the shape and the cardinal of the reduction window remains the same for all pixels of the pyramid.

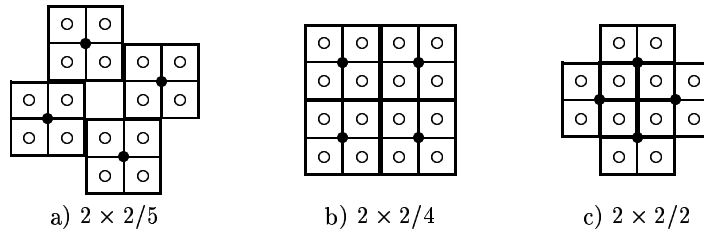
Another parameter of a regular pyramid is its *Reduction factor* which encodes the ratio between the size of two successive images in the pyramid. This ratio remains constant between any two levels of a regular pyramid. A *reduction function* computes the contents of a father from the contents of the pixels in the reduction window. A regular pyramid is thus formally defined by the ratio  $N \times N/q$ , where  $N \times N$  denotes the size of the reduction window while  $q$  denotes the reduction factor. Different types of pyramids may be distinguished according to the ratio between the size of the reduction window and the reduction factor:

- If  $N \times N/q < 1$ , the pyramid is named a *non-overlapping holed pyramid*. Within such pyramid, some pixels have no fathers [15] (e.g., the center pixel in Fig. 2(a)).



**Fig. 1.** A  $2 \times 2/4$  regular pyramid

- If  $N \times N/q = 1$ , the pyramid is called a *non overlapping pyramid without hole*(see e.g. in Fig. 2(b)). Within such pyramids, each pixel in the reduction window has exactly one father [15].
- If  $N \times N/q > 1$ , the pyramid is named an *Overlapping pyramid* (see e.g. in Fig. 2(c)). Each pixel of such a pyramid has several *potential parents* [10]. If each child selects one parent, the set of children in the reduction window of each parent is re-arranged. Consequently the receptive field may take any form inside the original receptive field.



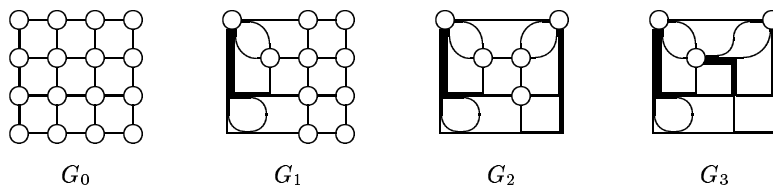
**Fig. 2.** Three different types of regular pyramids.

## 2.2 Irregular Pyramids

Beside their interesting properties mentioned in Section 1, regular pyramids have several drawbacks. Indeed, the fixed size and shape of the reduction window together with the fixed value of the reduction factor induce a poor ability of regular pyramids to adapt their structure to the data. The rigidity of regular pyramids induces several drawbacks such as the shift-dependence problem and the limited number of regions encoded at a given level of the pyramid [4].

Irregular pyramids overcome these negative properties while keeping the main advantages of their regular ancestors [23]. These pyramids are defined as a stack

of successively reduced graphs ( $G_0, G_1, G_2, G_3$  in Fig. 3). Each graph is built from the graph below by selecting a set of vertices named surviving vertices and mapping each non-surviving vertex to a surviving one [23]. Therefore each non-surviving vertex is the child of a surviving one which represents all the non surviving vertices mapped to it and becomes their father. If the initial graph  $G_0 = (V_0, E_0)$  is defined from the regular grid, we may associate one pixel to each vertex (in  $V_0$ ) and link vertices by edges in  $E_0$  according to the adjacency relationships defined on the regular grid ( $G_0$  in Fig. 3).



**Fig. 3.** An irregular pyramid using the 4-adjacency of pixels.

The reduction operation was first introduced by Meer [22, 23] as a stochastic process. Using such framework, the graph  $G_{l+1} = (V_{l+1}, E_{l+1})$  defined at level  $l + 1$  is deduced from the graph defined at level  $l$  by the following steps:

1. The selection of the vertices of  $G_{l+1}$  among  $V_l$ . These vertices are the surviving vertices of the decimation process.
2. A link of each non surviving vertex to a surviving one. This step defines a partition of  $V_l$ .
3. A definition of the adjacency relationships between the vertices of  $G_{l+1}$  in order to define  $E_{l+1}$ .

### 2.3 Selection of surviving vertices

In order to obtain a fixed decimation ratio between each level, Meer [22] imposes the following constraints on the set of surviving vertices:

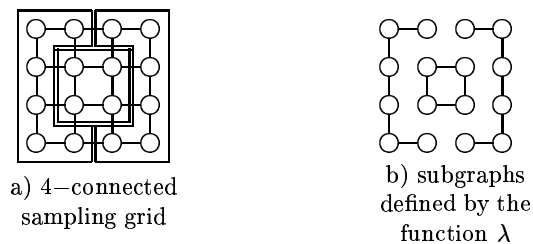
$$\forall v \in V_l - V_{l+1} \exists v' \in V_{l+1} : (v, v') \in E_l \quad (1)$$

$$\forall (v, v') \in V_{l+1}^2 : (v, v') \notin E_l \quad (2)$$

Constraint (1) insures that each non-surviving vertex is adjacent to at least a surviving one. Constraint (2) insures that two adjacent vertices cannot both survive. These constraints define a *maximal independent set* (MIS).

Meer [22] defines a set of surviving vertices fulfilling the maximal independent set requirements thanks to an iterative stochastic process: The outcome of a random variable uniformly distributed between  $[0, 1]$  is associated to each vertex. Then each vertex associated to a local maximum of the random variable survives.

This process is iterated on the remaining non-surviving vertices until the set of surviving vertices fulfills the maximal independent set requirements. This purely random process has been adapted by Montanvert [23] to the connected component analysis scheme by restricting the decimation process to a set of subgraphs of the graph  $G_l$ . Note that, using such restriction, the MIS is defined separately on each subgraph. Therefore, two surviving vertices may be adjacent in  $G_l$  if they do not belong to the same subgraph. The definition of the subgraphs is equivalent to the definition of a function  $\lambda$  from  $E_l$  to  $\{0, 1\}$ . For each  $(v, v') \in E_l$ ,  $\lambda((v, v'))$  is set to 1 if  $v$  and  $v'$  belong to the same subgraph and 0 otherwise. Applied to the connected component analysis framework, the function  $\lambda$  is simply defined by setting  $\lambda((v, v'))$  to 1 if  $v$  and  $v'$  belong to the same component (each surrounded by a closed curve in Fig. 4).



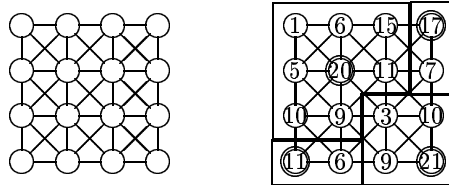
**Fig. 4.** The connected components defined by function  $\lambda$ .

Applied to the segmentation scheme, the function  $\lambda$  should be adapted at each level in order to take into account the greater variability of the input data. Jolion [14] improves the adaptability of the decimation process by using the local maxima of an interest operator instead of the local maxima of the random variable. For example, within the segmentation framework, Jolion defines the operator of interest as a decreasing function of the gray level variance computed in the neighborhood of each vertex. This operator provides a location of surviving vertices in homogeneous regions.

#### 2.4 Parent-Child definition

Given the set of surviving vertices, constraint (1) of a MIS insures that each non-surviving vertex is adjacent to at least one surviving vertex. Using the stochastic decimation process of Meer [22] and Montanvert [23], each non surviving vertex is attached to its adjacent surviving neighbor with the greatest value of the random variable. Jolion [14] uses a contrast measure such as the gray level difference, to link each non-surviving vertex to its least contrasted surviving neighbor. Each surviving vertex is thus the father of all non-surviving vertices attached to it. The set of children of one father is called its reduction window by reference to Regular pyramid notations. Fig. 5 represents the outcomes of the random variable on a graph defined from the  $4 \times 4$  8-connected sampling grid. The surviving vertices

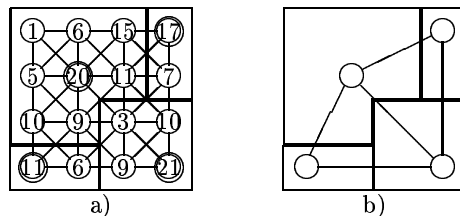
induced by the outcomes of this random variable are represented by an extra circle. The boundary of each reduction window is superimposed to the figure.



**Fig. 5.** Construction of a Maximal Independent Set using a random variable. The outcome of the random variable are displayed inside each vertex.

## 2.5 Connecting surviving vertices

The last step of the decimation process consists in connecting surviving vertices in  $G_{t+1}$ . Meer [22] joins two fathers by an edge if they have adjacent children (e.g. vertices labeled 15 and 7 in Fig. 6(a)). Since each child is attached directly to its father, two adjacent fathers at the reduced level are connected in the level below by paths of length less than three. Let us call these paths *connecting paths*. Using Constraint (2) of a MIS, two surviving vertices cannot be adjacent in  $G_t$ . Therefore, the length of the connecting paths is between 2 and 3. Using the restriction of the decimation process to subgraphs [23, 14] defined by the function  $\lambda$ , two surviving vertices may be adjacent by an edge mapped to 0 by  $\lambda$ . In such cases, the length of the connecting path is equal to 1.



**Fig. 6.** The reduced graph deduced from the maximal independent set (a) and the Parent-Children relationships (b).

## 2.6 Reaching the top of the Pyramid

The irregular pyramid is thus built recursively from  $G_0$  (the original sampling grid) to the apex  $G_t$ . The apex of the pyramid cannot be reduced by the decimation process. This case occurs when  $G_t$  is reduced to a single vertex or when no

decimation rule may be applied on the graph. For example, within the connecting component analysis scheme, the pyramid may be reduced until all connected components of the initial graph are reduced to a single vertex [23]. Within the segmentation framework, the decimation process may also stop when the difference between the value of any adjacent vertices exceeds a threshold. Another strategy consists in decimating the initial graph until it reduces to a single vertex and to select a set of roots within the pyramid. Jolion [14] defines a root as a vertex whose contrast measure with its parent exceeds a threshold. This strategy is also used within the regular pyramid framework [10] and allows us to select roots at different levels of the pyramid.

## 2.7 The need for Multi-edges and self-loops

The stochastic pyramids used by Meer [22] and Montanvert [23] or the adaptive pyramids defined by Jolion [14] combine the advantages of their regular ancestors with a greater adaptability to the input data. However, both the stochastic and adaptive reduction process utilize simple graphs e.g. graphs without multiple edges and self-loops. Therefore, if the graphs of the pyramid are used to encode a partition, an edge between two vertices in a simple graph pyramid encodes disconnected boundaries between the associated regions. Moreover, the lack of self-loops does not allow to differentiate inclusions from adjacencies relationships.

Fig. 7 illustrates these inadequacies. The left and right regions of the left image share two distinct boundaries. However, a reduction process such as the one defined in the stochastic pyramid framework provides a final graph encoding these multiple boundaries by a single edge.



**Fig. 7.** Inadequacy of the stochastic or adaptive decimation process.

This last drawback is induced by the definition of the edges between the surviving vertices. Indeed, two surviving vertices having several adjacent children will be connected by a single edge. A better understanding of this drawback may be achieved by interpreting the decimation process in terms of *edge contractions* and *edge removals*. The contraction of one edge  $(v, v')$  consists to identify  $v$  and  $v'$  into a new vertex  $v''$  and to remove the edge  $(v, v')$ . Note that, if  $v$  and  $v'$  are connected by an other edge, this last one becomes a self-loop doubly incident to  $v''$ . Since the contraction operation first identifies two distinct vertices, the contraction of self-loops is not defined. The removal of one edge consists to remove it from the set of edges of the graph.

Using the edge contractions and removal operations the decimation process performed by Meer [22], Montanvert [23] and Jolion [14] is equivalent to the application of the following steps:

1. The selection of a set  $S$  of surviving vertices.
2. The definition of a set of edges  $N$  linking each non-surviving vertex to its father.
3. The contraction of the set of edges  $N$ .
4. The removal of all multiple edges and self-loops.

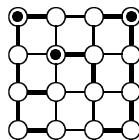
The definitions of the sets  $S$  and  $N$  vary according to the considered method. However, within the stochastic [22, 23] or adaptive [14] framework, the step 4 of the decimation process removes all multiple edges and self-loops between surviving vertices.

## 2.8 Contraction Kernels

Kropatsch [15] encodes the step 3 of the decimation process using a *Contraction Kernel*. A Contraction Kernel is defined on a graph  $G = (V, E)$  by a set of surviving vertices  $S$ , and a set of non surviving edges  $N$  (represented by bold lines in Fig. 8) such that:

- $(V, N)$  is a spanning forest of  $G$ ,
- Each tree of  $(V, N)$  is rooted by a vertex of  $S$ .

Since the set of non surviving edges  $N$  forms a forest of the initial graph, no self loop may be contracted and the contraction operation is well defined. If the graph is deduced from the initial sampling grid, each vertex on the border of this grid has to be adjacent to a special vertex encoding the background of the grid and called the *background vertex*. In order to preserve the image boundary, any edge encoding an adjacency relationship between a vertex of the graph and the background vertex must be excluded from  $N$ . Therefore, an initial graph  $G_0$  deduced from the sampling grid, may at most be reduced up to a graph  $G_t$  defined by two vertices connected by a single edge. If each vertex of  $G_0$  is associated to one pixel of the sampling grid, the two vertices encode respectively the image and its background. The edge between the two vertices corresponds to the image boundary.

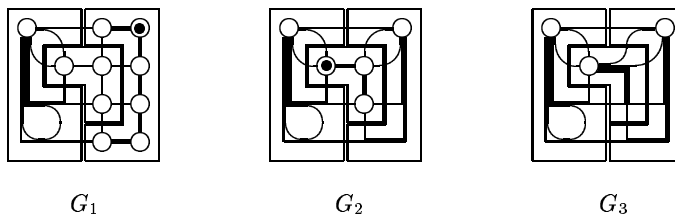


**Fig. 8.** A contraction Kernel  $(S, N)$  composed of three trees. Vertices belonging to  $S$  are represented with a filled circle inside (●).



The decimation of a graph by Contraction Kernels differs from the Stochastic [22, 23] or Adaptive [14] decimation processes on the two following points:

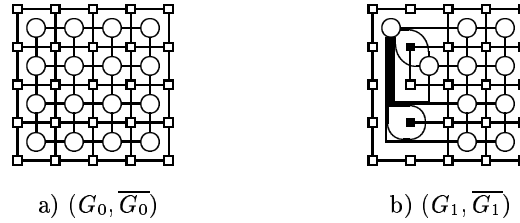
- First, using Contraction Kernels, the set of surviving vertices is not required to form a MIS. Therefore, two surviving vertices may be adjacent in the contracted graph. This last property avoids the use of the function  $\lambda$  defined by Montanvert [23] and Jolion [14].
- Secondly, using Contraction Kernels a non surviving vertex is not required to be directly linked to its father but may be connected to it by a branch of a tree (see Fig 8). The set of children of a surviving vertex may thus vary from a single vertex to a tree with any height.



**Fig. 9.** The successive contractions of the trees defined in Fig. 8

Fig. 9 illustrates the decimation performed by the Contraction Kernel displayed in Fig. 8 by contracting successively each tree of the forest  $(V, N)$ . Note that all the contractions may be performed in parallel.

The contraction of a graph reduces the number of vertices while maintaining the connections to other vertices. As a consequence, the decimation of a graph by a Contraction Kernel may induce the creation of some redundant edges. The characterization of these edges requires a better description of the topological relationships between the objects described by the graph. Since images are often projections of the reality into a plane, the image structure is planar and can be described by planar graphs. Moreover, if the initial graph is deduced from an initially planar grid (like 4-neighborhood) the initial graph and its reduced versions are planar. Given an initial planar graph  $G$ , the vertices of its dual  $\overline{G}$  are located inside every face of  $G$ . The edges of  $\overline{G}$  connect those dual vertices of which the corresponding faces are adjacent (lines between boxes in Fig. 10(a)). Note that the construction of the dual graph induces a one to one mapping between the edges of both graphs. Therefore, if  $N$  denotes a set of edges of the initial graph, we can denote by  $\overline{N}$  the set of associated edges in the dual graph. A well known result of graph theory [13] states that if  $G'$  is obtained from  $G = (V, E)$  by contracting a set  $N \subset E$ , the dual of  $G'$  may be obtained from  $\overline{G}$  by removing  $\overline{N}$  in  $\overline{G}$ . Conversely, if  $G'$  is obtained from  $G$  by removing a set  $N \subset E$ ,  $\overline{G'}$  is obtained from  $\overline{G}$  by contracting the edges  $\overline{N}$  in  $\overline{G}$ .



**Fig. 10.** The initial planar grid with the associated dual graph(a), and characterization of redundant edges(b). Dual vertices incident to redundant dual edges are represented by filled boxes.

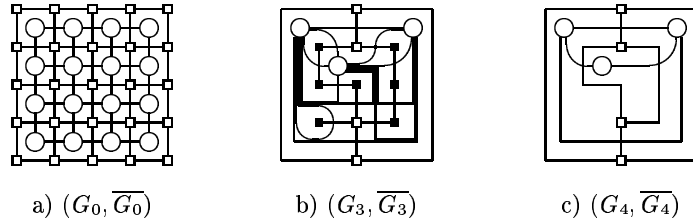
The characterization of redundant edges using the dual of the contracted graph is illustrated in Fig 10. If the initial graph is deduced from the initial 4-connected sampling grid, each dual vertex encodes a corner of a pixel and each edge of the dual graph encodes a side of a pixel. The upper filled box of Fig 10(b) represents a dual vertex with a degree equal to two. The two vertices defining the face of this dual vertex correspond to two regions of the initial image sharing a boundary artificially split in two subpaths by the dual vertex. The contraction of any of the two dual edges incident to this dual vertex concatenates the two paths. The contraction of this dual edge has to be followed by the removal of the associated edge in the initial graph to maintain the duality between both graphs. In the same way, the lower filled box of Fig 10(b) represents a dual vertex with a degree one. The empty self-loop associated to this dual vertex encodes an adjacency relationship between two vertices contracted in a same vertex by the Contraction Kernel. This useless<sup>1</sup> adjacency relation is deleted by removing the self-loop and contracting the associated dual edge.

The *level generation* from a pair of dual graphs  $(G_0, \overline{G_0})$  using a Contraction Kernel  $(S, N)$  is thus performed in two steps respectively illustrated in Fig. 11(b) and Fig. 11(c):

1. A set of edge contractions on  $G_0$  encoded by the Contraction Kernel  $(S, N)$ . The dual of the contracted graph  $G_1$  is computed from  $\overline{G_0}$  by removing the dual of the edges contained in  $N$ .
2. The removal of redundant edges encoded by a Contraction Kernel applied on the dual graph. Let us call this kernel a *Removal Kernel*. The edge contractions performed in the dual graph has to be followed by edge removals in the initial one in order to preserve the duality between the reduced graphs.

If each vertex of the original graph  $G_0$  encodes a pixel of the sampling grid, each vertex of the contracted graphs encodes a connected region of this grid. The preservation of the meaningful edges by the two step strategy defined by Kropatsch induces a one to one mapping between the edges of the graph and the region boundaries. Experiences with connected component analysis [18], with

<sup>1</sup> A self-loop is 'useless' if it does not surround any surviving part of the graph



**Fig. 11.** Two steps of the graph reduction operation encoded by Contraction Kernels

universal segmentation [16], and with topological analysis of line drawings [17] show the great potential of this concept.

### 3 Combinatorial Maps

Combinatorial maps and generalized combinatorial maps define a general framework which allows to encode any subdivision of nD topological spaces orientable or non-orientable with or without boundaries. The concept of maps has been first introduced by Edmonds [12] in 1960 and later extended by several authors such as Lienhardt [20], Tutte [26] and Cori [11]. This model has been applied to several fields of computer imagery such as geometrical modeling [3], 2D segmentation [1, 5] and graph labeling [21]. An exhaustive comparison of combinatorial maps with other boundary representations such as cell-tuples and quad-edges is presented in [20]. Recent trends in combinatorial maps apply this framework to the segmentation of 3D images [6, 2] and the encoding of hierarchies [8, 9].

The remaining of this paper will be based on 2D combinatorial maps which will be just called combinatorial maps. A Combinatorial map may be deduced from a planar graph by splitting each edge into two half edges called darts. The relation between two darts  $d_1$  and  $d_2$  associated to the same edge is encoded by the permutation  $\alpha$  which maps  $d_1$  to  $d_2$  and vice-versa. The permutation  $\alpha$  is thus an involution and its orbits are denoted by  $\alpha^*(d)$  for a given dart  $d$ . These orbits encode the edges of the graph. Moreover, each dart is associated to a unique vertex. The sequence of darts encountered when turning around a vertex is encoded by the permutation  $\sigma$ . Using a counter-clockwise orientation, the orbit  $\sigma^*(d)$  encodes the set of darts encountered when turning counter-clockwise around the vertex encoded by the dart  $d$ . A combinatorial map can thus be formally defined by:

**Definition 1. Combinatorial map**

*A combinatorial map  $G$  is the triplet  $G = (\mathcal{D}, \sigma, \alpha)$ , where  $\mathcal{D}$  is the set of darts and  $\sigma, \alpha$  are two permutations defined on  $\mathcal{D}$  such that  $\alpha$  is an involution:*

$$\forall d \in \mathcal{D} \quad \alpha \circ \alpha(d) = d$$

Note that, if the darts are encoded by positive and negative integers (as in Fig 12(a)), the involution  $\alpha$  may be implicitly encoded by:

$$\forall d \in \mathcal{D} \quad \alpha(d) = -d$$

This convention is often used for practical implementations of combinatorial maps [5, 7] where the permutation  $\sigma$  is simply implemented by an array of integers (see Fig. 12 (d)).

Using the combinatorial map formalism, the dual  $\overline{G}$  of a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  is defined as  $\overline{G} = (\mathcal{D}, \varphi = \sigma \circ \alpha, \alpha)$ . The orbits of the permutation  $\varphi$  encode the set of darts encountered when turning counter-clockwise around the vertices of the dual graph (Fig 12(b)). Such orbits may also be interpreted as the sequence of darts encountered when turning around a face. Note that due to our orientation conventions, each dart of an orbit have its associated face on its right (e.g. the  $\varphi$ -orbit (1, 3, 2) in Fig 12(a)).

A combinatorial map and its dual being deduced one from the other by a basic transformation, many properties of one graph remain true for its dual. For example, if the initial combinatorial map is connected its dual is also connected [8]. Moreover, many particular configurations such as self-loops (see Table 1) remain particular configurations in the dual combinatorial map (see [8] for a demonstration of these results).

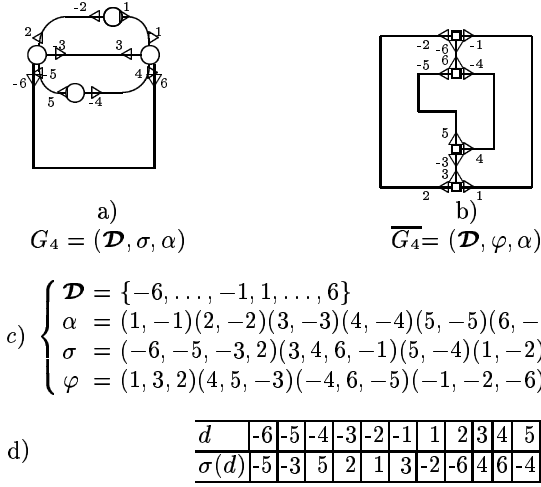
**Table 1.** Relationships between particular configurations in the original and dual combinatorial maps

	$(\mathcal{D}, \sigma, \alpha)$	$(\mathcal{D}, \varphi, \alpha)$
self-loop	$\alpha(d) \in \sigma^*(d)$	bridge
bridge	$\alpha(d) \in \varphi^*(d)$	self-loop
self-direct-loop	$(\sigma(d) = \alpha(d) \text{ or } \sigma(\alpha(d)) = d)$	pendant dart
pendant dart	$(\sigma(d) = d \text{ or } \sigma(\alpha(d)) = \alpha(d))$	self-direct-loop

Contraction kernels described in Section 2.8 may induce the creation of some redundant edges. Using the dual graph framework, such redundant edges are characterized by their associated dual edges which are incident to dual vertices with a degree lower than 3. Using the combinatorial map framework one of the dart of a redundant edge  $\alpha^*(d)$  is either a self-direct-loop (see Table 1) or belongs to an orbit of  $\varphi$  with a cardinal equal to 2. This last condition characterizes dual vertices with a degree 2 and may be checked by:  $\varphi^2(d) = d$  or  $\varphi(\sigma(d)) = \alpha(d)$ .

Combinatorial maps have thus the following interesting properties:

- The darts are ordered around each vertex and face. Note that, this information is not encoded by the simple graph data structure used by Meer [22], Montanvert [23] and Jolion [14] nor explicitly available in dual graph data structures.



**Fig. 12.** The combinatorial maps corresponding to  $\overline{G}_4$  and its dual. The lower part of this figure represents an encoding of the permutation  $\sigma$  by an array of integers.

- The simplicity and the efficiency of the computation of the dual combinatorial map avoids an explicit encoding of the dual graph.
- The combinatorial map formalism may be extended to any dimensions [19].

## 4 Combinatorial Pyramids

The aim of combinatorial pyramids is to combine the advantages of combinatorial maps with the reduction scheme defined by Kropatsch [15]. A combinatorial pyramid will thus be defined by an initial combinatorial map successively reduced by a sequence of contraction or removal kernels defined in the combinatorial map framework. The definition of combinatorial pyramids requires thus a formal definition of the contraction and removal operations in the combinatorial map framework.

Using combinatorial maps, the removal operation (see Definition 2) is encoded as an update of the permutation  $\sigma$ . The analytic form of Definition 2 is given by proposition 1. Note that this last operation may be performed efficiently if  $\sigma$  is encoded by an array of integers.

### Definition 2. Removal Operation

Given a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  and a dart  $d \in \mathcal{D}$ . If  $\alpha^*(d)$  is not a bridge, the combinatorial map  $G' = G \setminus \alpha^*(d) = (\mathcal{D}', \sigma', \alpha)$  is defined by:

- $\mathcal{D}' = \mathcal{D} \setminus \alpha^*(d)$  and
- $\sigma'$  is defined as:

$$\forall d \in \mathcal{D}' \quad \sigma'(d) = \sigma^n(d) \text{ with } n = \text{Min}\{p \in \mathbb{N}^* / \sigma^p(d) \notin \alpha^*(d)\}$$

Note that the bridges are excluded from Definition 2 in order to keep the number of connected components of the combinatorial map. Moreover, since both the dart  $d$  and its opposite  $\alpha(d)$  are removed, the involution  $\alpha$  of  $G'$  is restricted to the reduced dart set  $\mathcal{D} \setminus \alpha^*(d)$ . Although, the involution  $\alpha$  of the reduced graph is mathematically different from the one of the original graph, the implicit encoding of the permutation  $\alpha$  by the sign (see Section 3) may remain unchanged in both graphs. We thus consider that the involution  $\alpha$  is unchanged by the removal operation.

**Proposition 1.** *Given a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  and a dart  $d \in \mathcal{D}$  which is neither a bridge nor a self-direct-loop, the combinatorial map  $G \setminus \alpha^*(d) = (\mathcal{D} \setminus \alpha^*(d), \sigma', \alpha)$  is defined by:*

$$\begin{cases} \forall d' \in \mathcal{D} \setminus \sigma^{-1}(\alpha^*(d)) & \sigma'(d') = \sigma(d') \\ \sigma'(\sigma^{-1}(d)) & = \sigma(d) \\ \sigma'(\sigma^{-1}(\alpha(d))) & = \sigma(\alpha(d)) \end{cases}$$

As stated in Section 2.8, the removal of one edge on a planar graph induces the contraction of its associated edge in its dual. Conversely, the contraction of one edge on the initial graph is equivalent to the removal of the associated dual edge in the dual graph (Fig. 12). This last property allows us to define the contraction operation as a removal performed in the dual combinatorial map.

**Definition 3. Contraction operation**

*Given a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  and a dart  $d \in \mathcal{D}$  which is not a self-loop. The contraction of dart  $d$  creates the combinatorial map  $G/\alpha^*(d)$  defined by:*

$$G/\alpha^*(d) = \overline{G \setminus \alpha^*(d)}$$

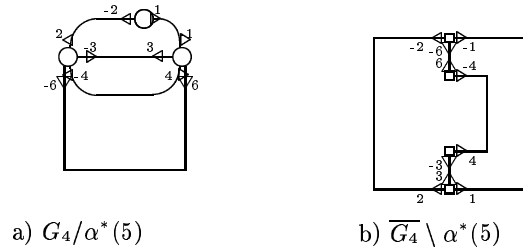
Note that this operation is well defined since  $d$  is a self-loop in  $G$  iff it is a bridge in  $\overline{G}$ . Thus, any sequence of removal or contraction operations will preserve the number of connected components of the initial graph. This last property is useful in the irregular pyramids framework because it attempts to simplify the initial planar map while preserving its essential topological properties. Moreover, since the involution  $\alpha$  is not modified by the removal operation and remains the same for one combinatorial map and its dual (see Section 3) it is also unchanged by the contraction operation.

Since the contraction operation is equivalent to a removal operation performed on the dual combinatorial map, the analytic form of Definition 3 may be deduced from the analytic form of the removal operation by substituting  $\varphi$  by  $\sigma$  in Proposition 1. This analytical form, rewritten in terms of modifications of the permutation  $\sigma$  is provided in Proposition 2. The proof of this result together with the description of the special cases excluded from Propositions 1 and 2 may be found in [8].

**Proposition 2.** *Given a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  and a dart  $d \in \mathcal{D}$  which is neither a pendant edge nor a self loop. The combinatorial map*

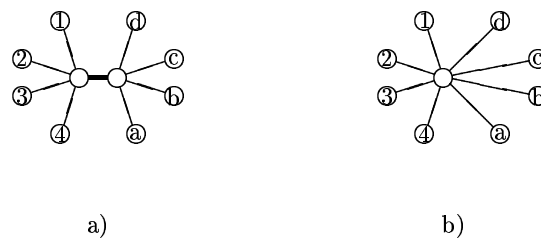
$G/\alpha^*(d) = (\mathcal{D} \setminus \alpha^*(d), \sigma', \alpha)$  is defined by:

$$\begin{cases} \forall d' \in \mathcal{D} \setminus \sigma^{-1}(\alpha^*(d)) & \sigma'(d) = \sigma(d) \\ \sigma'(\sigma^{-1}(d)) & = \sigma(\alpha(d)) \\ \sigma'(\sigma^{-1}(\alpha(d))) & = \sigma(d) \end{cases}$$



**Fig. 13.** Contraction of the edge  $\alpha^*(5)$  from  $G_4$  and the associated dual removal.

Note that, since the dual graph is implicitly encoded, any modification of the initial combinatorial map will also modify its dual. Therefore, using combinatorial maps the dual combinatorial map is both implicitly encoded and updated. Moreover, the updates of the permutation  $\sigma$  by the contraction operation preserves the orientation of the original combinatorial map. For example, let us suppose that the empty vertices displayed in Fig 14(a) represents two adjacent regions surrounded by the regions 1,2,3,4,a,b,c,d. This last sequence encodes the order in which the adjacent regions will be encountered when turning counter-clockwise around the two adjacent regions. The order of this sequence is preserved when the two regions are merged in Fig. 14(b), by the contraction of their common edge.



**Fig. 14.** Preservation of the orientation using combinatorial maps

Definitions 2 and 3 define the value of the permutation  $\sigma$  after the removal or the contraction of one edge. Such definitions may be extended to the removal or the contraction of a set of edges [8]. However, in order to avoid testing the preconditions of these operations for each edge we have to adapt the concept of contraction kernel (see section 2.8) to the combinatorial map framework.

**Definition 4. Contraction Kernel**

Given a connected combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$  and a non-empty set of darts  $\mathcal{B} \subset \mathcal{D}$  encoding the adjacency relationship between  $G$  and its background. The set  $K \subset \mathcal{D}$  will be called a contraction kernel iff:

1.  $K$  is a forest of  $G$ ,
2.  $K$  does not intersect  $\mathcal{B}$ .

The set  $\mathcal{SD} = \mathcal{D} - K$  is called the set of surviving darts.

Note that the set of surviving darts of a contraction kernel cannot be empty since it contains at least  $\mathcal{B}$ . Unlike dual graph contraction kernels, contraction kernels defined within the combinatorial map framework do not specify the surviving vertices. Within such framework, the surviving vertices are implicitly defined by the orbits of the updated permutation  $\sigma$ .

Given an initial combinatorial map  $G_0 = (\mathcal{D}, \sigma, \alpha)$  and a contraction kernel  $K$ , the contracted combinatorial map  $G' = G/K = (\mathcal{SD}, \sigma', \alpha)$  may be deduced from  $G$  by the iterative contraction of each dart contained in  $K$  using proposition 2. However, this last solution does not allow a straightforward parallel implementation and induces unnecessary updates of the permutation  $\sigma$ . The parallel computation of the contracted combinatorial map is based on the concept of *connecting walk*:

**Definition 5. Connecting walk**

Given a combinatorial map  $G = (\mathcal{D}, \sigma, \alpha)$ , a contraction kernel  $K$  and a dart  $d \in \mathcal{SD}$ , the connecting walk associated to  $d$  is equal to:

$$CW(d) = d, \varphi(d), \dots, \varphi^{n-1}(d) \text{ with } n = \text{Min}\{p \in \mathbb{N}^* \mid \varphi^p(d) \in \mathcal{SD}\}$$

Note that, only the first dart of each connecting walk survives. Each connecting walk  $CW(d)$  connects the surviving darts  $d$  and  $\varphi^n(d)$  by a sequence of non surviving darts.

Connecting walks may be considered as an extension of the connecting paths used by Jolion [14] and Montanvert [23](see Section 2.5). However, using contraction kernels, the MIS requirements are substituted by the more general concept of forest requirement. Therefore, the paths defined by Montanvert and Jolion have to be replaced by the more general concept of walks [13]. Moreover, since the trees of the forest may have any depth, the length of a connecting walk is not bounded by 3. Finally, using combinatorial maps, a connecting walk is an ordered sequence of darts which preserves the order defined on  $\mathcal{D}$  by the permutation  $\varphi$ . More precisely, given a surviving dart  $d$  and its connecting walk we have [9]:

$$\forall d \in \mathcal{SD} \quad \varphi'(d) = \varphi^n(d) \tag{3}$$



Where  $n = \text{Min}\{p \in \mathbb{N}^* \mid \varphi^p(d) \in \mathcal{SD}\}$  and  $\varphi' = \sigma' \circ \alpha$  denotes the  $\varphi$  permutation of the contracted graph  $G/K$ . The permutation  $\alpha$  being an involution, the permutation  $\sigma'$  may be retrieved from  $\varphi'$  by  $\sigma' = \varphi' \circ \alpha$ . We have thus:

$$\forall d \in \mathcal{SD} \quad \sigma'(d) = \varphi^n(\alpha(d)) \text{ with } n = \text{Min}\{p \in \mathbb{N}^* \mid \varphi^p(\alpha(d)) \in \mathcal{SD}\}$$

Note that,  $\varphi^{n-1}(\alpha(d))$  is the last dart of  $CW(\alpha(d))$ . Therefore, the permutation  $\alpha$  remains unchanged in the contracted combinatorial map while the permutation  $\sigma'$  maps each surviving dart  $d$  to the  $\varphi$ -successor of the last dart of  $CW(\alpha(d))$ . Therefore, the computation of the  $\sigma$ -successor of a dart  $d$  in the contracted combinatorial map requires to traverse  $CW(\alpha(d))$ . Sequential algorithm 1 computes the  $\sigma$ -successor of all the surviving darts in the contracted combinatorial map. Since the set of connecting walks forms a partition of  $\mathcal{D}$  (see proof in [9]), Algorithm 1 has to traverse  $|\mathcal{D}|$  darts. Its complexity is thus equal to  $\mathcal{O}(|\mathcal{D}|)$ .

```

1  dart contracted_sigma( $G = (\mathcal{D}, \sigma, \alpha), K$ )
2  {
3      For each  $d \in \mathcal{SD} = \mathcal{D} - K$ 
4      do
5           $d' = \varphi(\alpha(d)) = \sigma(d)$  // Second dart of  $CW(\alpha(d))$ 
6
7          while( $d' \in K$ ) // computation of  $CW(\alpha(d))$ 
8               $d' = \varphi(d')$ 
9               $\sigma'(d) = d'$ 
10         done
11     }
```

**Algorithm 1:** Computation of the permutation  $\sigma'$  of the contracted combinatorial map  $G' = (\mathcal{SD}, \sigma', \alpha)$

Using the dual graph pyramid scheme, a removal kernel is defined as a contraction kernel applied on the dual combinatorial map. Therefore, all results already obtained on contraction kernels remain valid in the dual combinatorial map for removal kernels [9]. The connecting walks defined by removal kernels may be constructed as a sequence of  $\sigma$ -successors of a surviving dart:

$$\forall d \in \mathcal{SD} \quad CW_R(d) = d.\sigma(d) \dots \sigma^{n-1}(d) \text{ with } n = \text{Min}\{p \in \mathbb{N}^* \mid \sigma^p(d) \in \mathcal{SD}\}.$$

The  $\sigma$ -successor of any surviving dart  $d$  in the removed combinatorial map is then defined as  $\sigma^n(d)$  where  $\sigma^{n-1}(d)$  is the last dart of  $CW_R(d)$ . The updated permutation  $\sigma'$  may thus be computed from the original combinatorial map by a traversal of the connecting walks defined by the removal kernel. This traversal may be performed by a slightly modified version of Algorithm 1 where line 8 is replaced by the instruction  $d' = \sigma(d)$ .

Using the same scheme as dual graph pyramids (see section 2.8), the level generation using a contraction kernel  $K$  is performed in two steps:

1. One set of edge contractions encoded by a contraction kernel  $K_1$ .
2. The removal of redundant edges encoded by a removal kernel  $K_2$ . Such redundant edges are incident to faces with a degree one or two (see Section 2.8 and 3).

## 5 Conclusion

In this paper we have described different hierarchical models and some of the basic processes working on them: regular, irregular and combinatorial pyramids. All three possess the following important properties [4]:

- Independence to resolution,
- global to local interactions,
- $\log(\text{image\_diameter})$  parallel complexity.

In the remaining of this section, we want to differentiate the advantages and drawbacks of the three types of pyramids discussed in the paper. The first part of Table 2 describes the properties of the basic entities represented at each level of the structure. In the second part, we consider the properties of the projection of these basic entities into the image plan.

Each level of the hierarchy encodes a set of resolution cells (“region”) and their respective adjacencies. A resolution cell is explicitly encoded by a pixel or a vertex of the graph structure while it is implicitly coded by a  $\sigma$ -orbit of the combinatorial map. An adjacency relation between two resolution cells is implicitly encoded by a crack between two 4-adjacent pixels while it is explicitly coded as an edge of the graph or a couple of darts  $(d, \alpha(d))$  in the other hierarchies. A point is defined where three or more resolution cells meet. It is implicitly defined as a pixel corner in the regular structure and as a  $\varphi$ -orbit in the combinatorial map while it is not encoded by the simple graph pyramids and explicitly encoded as a dual vertex in the dual graph. A combinatorial map is the only structure which explicitly codes the orientation of the resolution cells local arrangements by the permutation  $\sigma$ .

The receptive field of a resolution cell defines its projection into the image plane. The regular pyramid is the only structure which restricts the form of the receptive fields to regular shapes. Using hierarchies, the segmentation of the image plane is achieved by selecting a set of roots (Section 2.6). The receptive field of each root defines a region of the image plane. The pyramid linking [10] algorithm adapts the shape of the regular pyramid receptive fields to the image content. All other hierarchies may use the receptive fields obtained from the pyramid construction scheme. Moreover, the pyramid linking algorithm is the only segmentation scheme which may produce non connected regions (e.g. [24]). A boundary is defined as the image embedding of an adjacency relation. This boundary may be not connected using regular or simple graph hierarchies.

Dual graph and Combinatorial pyramids are thus the only hierarchies which provide both connected regions and connected boundaries. Combinatorial pyramids combine the advantages of dual graph pyramids with an explicit orientation

**Table 2.** Basic hierarchical entities and properties of their image embedding.

Pyramids	Regular	Irregular		Combinatorial
		Simple graph	Dual graph	
Region	pixel	vertex	vertex	$\sigma$ -orbit
Adjacency	crack	edge	edge	$(d, \alpha(d))$
Point orientation	pixel corner	-	dual vertex	$\varphi$ -orbit
	-	-	implicit	$\sigma, \varphi$
Properties of image embedding				
Receptive field	regular shape	arbitrary shape	arbitrary shape	arbitrary shape
Segmentation scheme	linking	adaptive decimation	adaptive dual contraction	adaptive dual contraction
Region(vertex)	not connected	connected	connected	connected
Boundaries (edge)	not connected	not connected	connected	connected

of the boundary segments of the embedded object thanks to the permutation  $\sigma$ . Moreover, using combinatorial maps, the dual graph is both implicitly encoded and updated. Finally, the combinatorial map formalism is defined in any dimensions.

## References

- [1] E. Ahronovitz, J. Aubert, and C. Fiorio. The star-topology: a topology for image analysis. In *5th DGCI Proceedings*, pages 107–116, 1995.
- [2] Y. Bertrand, G. Damiand, and C. Fiorio. Topological map: Minimal encoding of 3d segmented images. In J. M. Jolion, W. Kropatsch, and M. Vento, editors, *3<sup>rd</sup> Workshop on Graph-based Representations in Pattern Recognition*, pages 64–73, Ischia(Italy), May 2001. IAPR-TC15, CUEN.
- [3] Y. Bertrand and J. Dufourd. Algebraic specification of a 3D-modeler based on hypermaps. *CVGIP: Graphical Models and Image Processing*, 56(1):29–60, Jan. 1994.
- [4] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognit Letter.*, 11(9):605–617, Sept. 1990.
- [5] J. P. Braquelaire and L. Brun. Image segmentation with topological maps and inter-pixel representation. *Journal of Visual Communication and Image representation*, 9(1), 1998.
- [6] J. P. Braquelaire, P. Desbarats, and J. P. Domenger. 3d split and merge with 3-maps. In J. M. Jolion, W. Kropatsch, and M. Vento, editors, *3<sup>rd</sup> Workshop on Graph-based Representations in Pattern Recognition*, pages 32–43, Ischia(Italy), May 2001. IAPR-TC15, CUEN.
- [7] L. Brun. *Segmentation d'images couleur à base Topologique*. PhD thesis, Université Bordeaux I, 351 cours de la Libération 33405 Talence, December 1996.
- [8] L. Brun and W. Kropatsch. Dual contractions of combinatorial maps. Technical Report 54, Institute of Computer Aided Design, Vienna University of Technology, Istr. 3/1832,A-1040 Vienna AUSTRIA, January 1999.

- [9] L. Brun and W. Kropatsch. Pyramids with combinatorial maps. Technical Report PRIP-TR-057, PRIP, TU Wien, 1999.
- [10] 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, December 1981.
- [11] R. Cori. *Un code pour les graphes planaires et ses applications*. PhD thesis, Université Paris VII, 1975.
- [12] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices American Society*, 7, 1960.
- [13] F. Harary. *Graph Theory*. Addison-Wesley, 1972.
- [14] J. Jolion and A. Montanvert. The adaptative pyramid : A framework for 2d image analysis. *Computer Vision, Graphics, and Image Processing*, 55(3):339–348, May 1992.
- [15] W. G. Kropatsch. From equivalent weighting functions to equivalent contraction kernels. In E. Wenger and L. I. Dimitrov, editors, *Digital Image Processing and Computer Graphics (DIP-97): Applications in Humanities and Natural Sciences*, volume 3346, pages 310–320. SPIE, 1998.
- [16] W. G. Kropatsch and S. BenYacoub. Universal Segmentation with PIRRamids. In A. Pinz, editor, *Pattern Recognition 1996, Proc. of 20th ÖAGM Workshop*, pages 171–182. OCG-Schriftenreihe, Österr. Arbeitsgemeinschaft für Mustererkennung, R. Oldenburg, 1996. Band 90.
- [17] W. G. Kropatsch and M. Burge. Minimizing the Topological Structure of Line Images. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'98 and SPR'98*, volume Vol. 1451 of *Lecture Notes in Computer Science*, pages 149–158, Sydney, Australia, August 1998. Springer, Berlin Heidelberg, New York.
- [18] W. G. Kropatsch and H. Macho. Finding the structure of connected components using dual irregular pyramids. In *Cinquième Colloque DGCI*, pages 147–158. LLAIC1, Université d'Auvergne, ISBN 2-87663-040-0, September 1995.
- [19] P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Annual ACM Symposium on Computational Geometry, all*, volume 5, 1989.
- [20] P. Lienhardt. Topological models for boundary representations: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82, 1991.
- [21] J. Marchandier, S. Michelin, and Y. Egels. A graph labelling approach for connected feature selection. In A. Amin, F. J. Ferri, P. Pudil, and F. J. Iñesta, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'2000 and SPR'2000*, volume Vol. 1451 of *Lecture Notes in Computer Science*, pages 287–296, Alicante, Spain, August 2000. Springer, Berlin Heidelberg, New York.
- [22] P. Meer. Stochastic image pyramids. *Computer Vision Graphics Image Processing*, 45:269–294, 1989.
- [23] 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.
- [24] P. F. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. *Pattern Recognition*, 28(6):907–920, June 1995.
- [25] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer Verlag, Berlin, 1984.
- [26] W. Tutte. *Graph Theory*, volume 21. Addison-Wesley, encyclopedia of mathematics and its applications edition, 1984.