Defining regions within the Combinatorial Pyramid framework

Luc Brun[†] and Walter Kropatsch[‡*]

† Laboratoire d'Études et de Recherche en Informatique(EA 2618)

Université de Reims - France

and

‡ Institute for Computer-aided Automation

Pattern Recognition and Image Processing Group

Vienna Univ. of Technology- Austria

Abstract

Irregular Pyramids are defined as a stack of successively reduced graphs. Each vertex of a reduced graph is associated to a set of vertices in the base level graph named its receptive field. If the initial graph is deduced from a planar sampling grid its reduced versions are planar and each receptive field is a region of the initial grid. Combinatorial Pyramids are defined as a stack of successively reduced combinatorial maps. Combinatorial maps are based on half edges named darts and the receptive field of a dart is a sequence of darts in the base level combinatorial map. We present in this paper preliminary results showing how to define regions from the receptive fields of the darts.

1 Introduction

A *Region* is defined as a connected set of pixels. The regions defined by segmentation algorithms fulfill some homogeneity criterion and usually encode either the projections of the different objects of a scene or the main parts of some of these objects. Regions are lot more informative than pixels and a wide variety of internal properties such that the shape, the texture or the set of colors may be extracted from them. External properties such as the adjacency or the inclusion relationships between regions also provide meaningful information about a scene.

Image partitions into region may be defined in parallel using hierarchical data structures. These data structures encode additionally the levels of details of a partition. For example, using such data structures, the hierarchical relation between one region encoding a face and the

*This Work was supported by the Austrian Science Foundation under P14445-MAT.

regions encoding the different parts of this face (e.g. the eyes and the hears) may be encoded explicitly. The *Regular image pyramids* is a hierarchical data structure introduced in 1981/82 [4] as a stack of images with exponentially reduced resolution. 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 one reduction window are the *children* of the pixel which defines it. This father-child relationship may be extended by transitivity down to the base level image. The set of children of one pixel in the base level is named its *receptive field* (RF) and defines the embedding of this pixel on the original image.

Using regular pyramid the receptive fields are not necessarily connected [1] and may thus contradict the usual definition of regions. This drawback may be overcome by using irregular pyramids defined as a stack of successively reduced graphs. The *Simple graph* and *Dual graph Pyramids* respectively introduced by Meer [6] and Kropatsch [5] define each level of the pyramid by selecting a set of vertices named surviving vertices and mapping all non surviving vertices to surviving ones. The father-child relationship induced by this mapping defines the reduction window of each surviving vertex. The transitive closure of this relation defines, as in regular pyramids, the receptive field of a surviving vertex. Using such a reduction scheme, if the initial graph is defined from a planar sampling grid all the reduced versions of the grid are planar. Moreover, each initial vertex may be associated to one pixel and the receptive field of a surviving vertex is defined as a connected set of pixels.

Combinatorial pyramids are defined as a stack of combinatorial maps successively reduced by contraction and removal operations. Combinatorial Pyramids are equivalent to dual graph pyramids with the exception that they represent the orientation explicitly. The expected advantages of such hierarchies within the image analysis framework are presented in [3]. Combinatorial maps are based on darts. Hence the the reduction window and the receptive fields of Combinatorial Pyramids are expressed in terms of darts. However, using either simple graph or dual graph pyramids the basic entity is the vertex/pixel. Therefore, the receptive fields may be interpreted as regions of an initial image. We present in this paper preliminary results showing how to define regions within the combinatorial pyramid framework.

2 Combinatorial maps

A combinatorial map may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. Figure 1(a) demonstrates the derivation of a combinatorial map from a plane graph. First edges are split into two half edges called *darts*, each dart having its origin at the vertex it is attached to. The fact that two half-edges (darts) stem from the same edge is recorded in the reverse permutation α. A second permutation σ encodes the set of darts encountered when turning counterclockwise around a vertex.

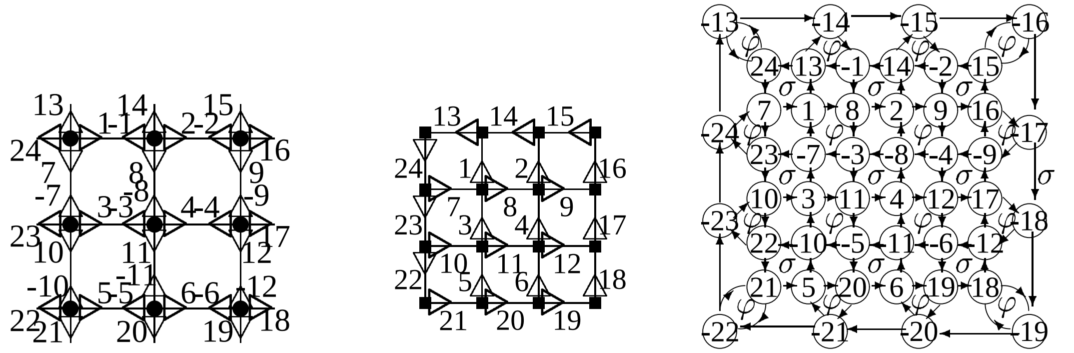
A combinatorial map is thus defined as a triplet $G = (\mathcal{D}, \sigma, \alpha)$, where $\mathcal{D}$ is the set of darts and σ, α are two permutations defined on $\mathcal{D}$ such that α is an involution:

$$\forall d \in \mathcal{D} \quad \alpha^2(d) = d \tag{1}$$

Note that, if the darts are encoded by positive and negative integers, the involution α may be implicitly encoded by the sign (Figure 1(a)).

The symbols $\alpha^*(d)$ and $\sigma^*(d)$ stand, respectively, for the α and σ orbits of the dart d. More generally, if d is a dart and π a permutation we will denote the π-orbit of d by $\pi^*(d)$.

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, its dual is defined by $\overline{G} = (\boldsymbol{\mathcal{D}}, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$. The orbits of the permutation φ encode the set of darts encountered when turning around a face. Note that, using a counter-clockwise orientation for permutation σ, each dart of a φ-orbit has its associated face on its right (see e.g. the φ-orbit $\varphi^*(1) = (1, 8, -3, -7)$ in Figure 1(a)).



(a) G (b) $\overline{G}$ (c) $\mathcal{OG}$

Figure 1: A 3×3 grid encoded by a combinatorial map

Figure 1 illustrates the encoding of a 3×3 4-connected discrete grid by a combinatorial map. Each vertex of the initial combinatorial map (Figure 1(a)) encodes a pixel of the grid. The σ-orbit of one vertex encodes its adjacency relationships with neighboring vertices (see e.g. the σ-orbit $(-8, -3, 11, 4)$ encoding the central vertex). The α successors of the darts 13 to 24 are not represented in Figure 1(a) in order to not overload it. These darts encode the adjacency relationships between the external pixels of the grid and its background. The σ orbit of the background vertex is equal to the sequence of darts form -13 to -24 : $(-13, -14, \ldots, -23, -24)$. The dual combinatorial map is represented in Figure 1(b). We also did not represent the α-successor of the positive darts on this Figure to not overload it. Each vertex of this dual map may be associated to a corner of a pixel. Moreover, each of its dart may be understood as an oriented crack, i.e. as a side of a pixel with an orientation. For example, the dart 1 in Figure 1(b) encodes the left side of the upper-left pixel oriented from bottom to top. The dart -1 encodes the same crack with an orientation from top to bottom. Using the above interpretation of darts, the σ-orbit of each pixel defines the sequence of cracks which surrounds it. For example, the upper left pixel is encoded in Figure 1(b) by the σ-orbit $(1, 13, 24, 7)$. In the same way, the pixel located on the first line, second column, is encoded by the σ-orbit $(2, 14, -1, 8)$. The fact that these two pixels share a same crack with a different orientation is recorded by the darts 1 and -1 which belong to a same edge.

Figures 1(c) illustrates the $\sigma - \varphi$ representation of a combinatorial map. Within this alternative representation, a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ is represented by an oriented planar graph $\mathcal{OG} = (V, E)$. The set V of vertices of $\mathcal{OG}$ is equal to the set of darts $\boldsymbol{\mathcal{D}}$ and an oriented edge $e \in E$ connects two vertices d_1 and d_2 iff either $d_2 = \sigma(d_1)$ or $d_2 = \varphi(d_1)$. Using this representation, the σ and φ orbits of the combinatorial map are represented by the faces of the oriented graph $\mathcal{OG}$. Note that each vertex of $\mathcal{OG}$ has two incoming arcs (its σ and φ predecessors) and two outgoing ones (its σ and φ successors).

3 Combinatorial Pyramids

The aim of combinatorial pyramids is to combine the advantages of combinatorial maps with the reduction scheme defined by Kropatsch [5] (Section 1). A combinatorial pyramid is thus defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations.

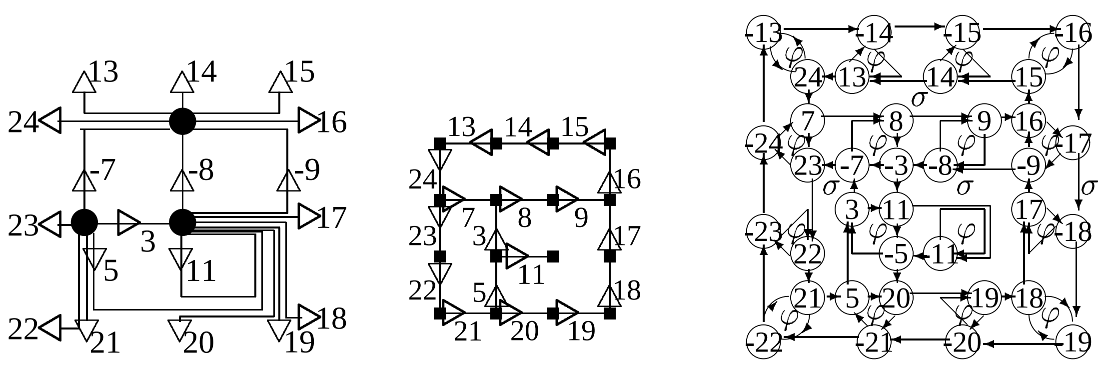
In order to preserve the number of connected components of the initial combinatorial map, the contraction of self-loops must be avoided. This last requirement may be satisfied if the set of edges to be contracted forms a forest of the initial combinatorial map. A forest is defined as a set of non connected trees. Within the combinatorial map framework, a tree may be characterized as a combinatorial map with only one φ-orbit (i.e. only one face). A more formal definition may be found in [2][Def. 4]. A set of edges to be contracted satisfying the above requirement is called a contraction kernel:

Definition 1 Contraction Kernel

Given a connected combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$ the set $K \subset \mathcal{D}$ will be called a contraction kernel iff K is a forest of G.

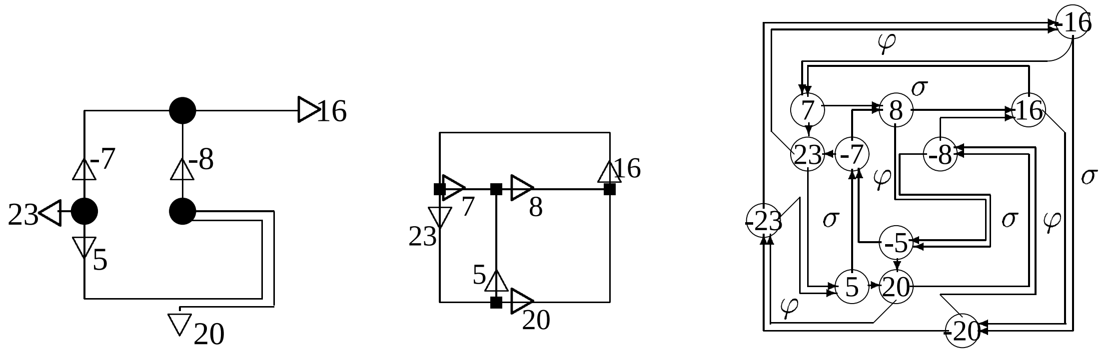
The set $\mathcal{SD} = \mathcal{D} - K$ is called the set of surviving darts.

Given a contraction kernel K, we denote by $\mathcal{CC}(K)$ its set of connected components. Since K is a forest, each $\mathcal{T} \in \mathcal{CC}(K)$ is a tree. Intuitively, a tree $\mathcal{T} \in \mathcal{CC}(K)$ collapses in one vertex a connected set of vertices of the initial combinatorial map. Since each initial vertex is associated to one pixel, the contracted vertex encodes a region.



(a) $G_1 = G/K_1$ (b) $\overline{G_1} = \overline{G/K_1}$ (c) $\mathcal{OG}_1$

Figure 2: Reduction of the initial grid displayed in Figure 1 by the contraction kernel K_1



(a) $(G/K_1) \setminus K_2$ (b) $\overline{(G/K_1) \setminus K_2}$ (c) $\mathcal{OG}_2$

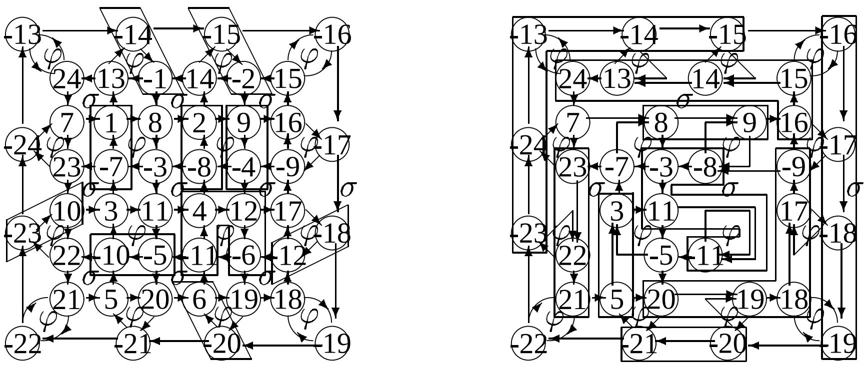
Figure 3: Reduction of the contracted combinatorial map displayed in Figure 2 by the removal kernel K_2

Figure 2 illustrates a contraction of the initial combinatorial map represented in Figure 1 by a contraction kernel K_1 defined by the trees $\alpha^*(1,2), \alpha^*(4,12,6)$ and $\alpha^*(10)$. Since each initial vertex is incident to a contracted edge this forest spans the initial combinatorial map and we obtain 3 surviving vertices encoding 3 regions.

One can note on Figure 2 that many edges encode redundant boundaries. For example the edges $\alpha^*(8)$ and $\alpha^*(9)$ encode a same adjacency relationship between the top vertex and the center one. Such edges correspond to an artificial split of a boundary between two regions (see Figure 2(b)). These edges, named double edges, may be characterized by the relationship $\varphi^2(d) = d$ where d is one of the dart of the double edge. We have for example on Figure 2(b), $\varphi(9) = -8$ and $\varphi(-8) = 9$, thus $\varphi^2(9) = 9$ and $\alpha^*(9)$ is a double edge. Another type of redundant edge is the direct self-loop, characterized by the relationship $\sigma(d) = \alpha(d)$ where d is one of the darts of the direct self-loop $\alpha^*(d)$. We have for example, on Figure 2, $\sigma(11) = -11$. Such edges may be interpreted in the dual combinatorial map as inner-boundaries (see edge $\alpha^*(11)$ in Figure 2(b)). Such redundant edges are removed by a removal kernel defined as a forest of the dual combinatorial map. This last constraint insures that no self-loop will be contracted in the dual combinatorial map and thus that no bridge may be removed in the initial one. Figure 3 represents the simplified combinatorial map deduced from the one represented in Figure 2 by the removal kernel $K_2 = \{\alpha^*(15,14,13,24), \alpha^*(9), \alpha^*(11,3), \alpha^*(19,18,17), \alpha^*(22,21)\}$. Note that given a sequence of double edges, the choice of the surviving edge is arbitrary. For example, a choice of the tree $\alpha^*(20,19,18)$ instead of $\alpha^*(19,18,17)$ would lead to an equivalent simplified combinatorial map with a surviving edge equal to $\alpha^*(17)$ instead of $\alpha^*(20)$.

Contraction and removal kernels specify the set of edges which must be contracted or removed. The creation of the reduced combinatorial map from a contraction or a removal kernel is performed in parallel by using connecting walks [3]. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, a kernel K and a surviving dart $d \in \mathcal{SD} = \boldsymbol{\mathcal{D}} - K$, the connecting walk associated to d is either equal to:

$$CW(d) = d, \varphi(d), \ldots, \varphi^{n-1}(d) \text{ with } n = Min\{p \in \mathbb{N}^* \mid \varphi^p(d) \in \mathcal{SD}\} \quad (2)$$



(a) connecting walks in $\mathcal{OG}$ defined by K_1

(b) Connecting walks in $\mathcal{OG}_1$ defined by K_2

Figure 4: Connecting walks defined by K_1 (a) and K_2(b).

if K is a contraction kernel and

$$CW(d) = d, \sigma(d), \ldots, \sigma^{n-1}(d) \text{ with } n = Min\{p \in \mathbb{N}^* \mid \sigma^p(d) \in \mathcal{SD}\} \tag{3}$$

If K is a removal kernel.

Figure 4 represents the connecting walks defined by K_1 and K_2 superimposed to the oriented graphs $\mathcal{OG}$ (Figure 1(c)) and $\mathcal{OG}_1$ (Figure 2(c)) respectively associated to $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and $G_1 = G/K_1 = (\mathcal{SD}_1, \sigma_1, \alpha)$. Let us consider the surviving dart -5 of G_1 (Figure 4(a)). Since K_1 is a contraction kernel $CW(-5)$ is equal to the sequence of non-surviving φ successors of -5. Since $\varphi(-5) = -10 \in K_1$ and $\varphi(-10) = 3 \in \mathcal{SD}_1$, we have $CW(-5) = -5. - 10$. In the same way, let us now consider the combinatorial map $G_2 = G_1 \setminus K_2 = (\mathcal{SD}_2, \sigma_2, \alpha)$ and the surviving dart $5 \in \mathcal{SD}_2$ (Figure 4(b)). Since K_2 is a removal kernel, the connecting walk of 5 is defined as the sequence of non-surviving σ-successors of 5. Since $\sigma_1(5) = 3 \in K_2$ and $\sigma_1(3) = -7 \in \mathcal{SD}_2$ (Figure 2) we have: $CW(5) = 5.3$.

Given a kernel K and a surviving dart $d \in \mathcal{SD}$, such that $CW(d) = d.d_1 \ldots d_p$, the successor of d within the reduced combinatorial map $G' = (\mathcal{SD}, \sigma', \alpha)$ is retrieved from $CW(d)$ by the following equations [3]:

$$\begin{array}{rcll} \varphi'(d) & = & \varphi(d_p) & \text{if } K \text{ is a contraction kernel} \\ \sigma'(d) & = & \sigma(d_p) & \text{if } K \text{ is a removal kernel} \end{array} \tag{4}$$

Using Figure 4, we have for example $\varphi_1(-5) = \varphi(-10) = 3$ (Figure 4(a)) and $\sigma_2(5) = \sigma_1(3) = -7$ (Figure 4(b) see also Figure 3(c)).

Note that, if K is a contraction kernel, the connecting walk $CW(d)$ allows to compute $\varphi'(d)$. The σ-successor of d within the contracted combinatorial maps may be retrieved from $CW(\alpha(d)) = \alpha(d).d'_1, \ldots, d'_p$. Indeed, we obtain by using equations 1 and 4: $\varphi'(\alpha(d)) = \sigma'(\alpha(\alpha(d))) = \sigma'(d) = \varphi(d'_p)$. We may alternatively consider the sequence $d.CW^*(\alpha(d))$ where $CW^*(\alpha(d))$ denotes the sequence $CW(\alpha(d))$ without its first dart $\alpha(d)$. In this case, using equation 4 the σ successor of a surviving dart d is provided by the last dart of $CW(d)$ if K is a removal kernel and by the last dart of $d.CW^*(\alpha(d))$ if K is a contraction kernel.

If K is a removal kernel (resp. a contraction kernel), $CW(d)$ (resp. $CW^*(\alpha(d))$) defines the sequence of non surviving darts which are mapped to the surviving dart d in the reduced combinatorial map. Such sequences encode thus the notion of reduction window within the combinatorial pyramid framework. In the following section we show how such sequences may be combined to define higher level objects such as regions.

4 Regions

The definition of regions within the combinatorial pyramid framework supposes first to express the notion of a connected set of pixels in terms of darts. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, we define a connected sequence of darts as a sequence $P = d_1. \dots, d_n$ such that all darts of the sequence are distinct and each dart $(d_i)_{i\in\{2,\dots,n\}}$ is either the σ or φ successor of d_{i-1}. Intuitively, two darts of such a sequence belong either to the same vertex or to adjacent vertices. If d_1 is the σ or φ successor of d_n, such a sequence is called a cycle. Note that a connected sequence of darts defines either a path of a cycle in the oriented graph $\mathcal{OG}$(Section 2). Given the connected sequence of darts, we can define the notion of connected set of darts. This notion is stronger than the usual notion of connected set of vertices since one can easily show that the set of vertices defined by a connected set of darts is connected. However, a region is usually defined as a connected set of pixels rather than a connected set of darts. Therefore a connected set of darts must contain all the darts of its vertices in order to be called a region. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and a connected set of darts $R \subset \boldsymbol{\mathcal{D}}$, this last condition may be written: $\sigma^*(R) = R$. The above considerations are resumed in the following definition:

Definition 2 Region

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, a set of darts $R \subset \boldsymbol{\mathcal{D}}$ is called a region of G iff:

- *R is connected: Given any two darts $(d, d') \in R^2$ it exists one connected sequence of darts P included in R which connects either d to d' or d' to d.*
- *R contains its vertices: $\sigma^*(R) = R$*

Let us consider an initial combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and a contracted one $G' = (\mathcal{SD}, \sigma', \alpha)$ deduced from G by a contraction kernel K. Each tree of K contracts a connected set of vertices into one surviving vertex. Let us consider such a surviving vertex $\sigma'^*(d_1) = (d_1, \dots, d_p)$. Since each dart d_i of this σ'-orbit is connected in G to $d_{i+1} = \sigma'(d_i)$ by $d_i.CW^*(\alpha(d_i))$ (see Section 3), the reduction window of the vertex $\sigma'^*(d_1)$ is encoded by:

$$R_{\sigma'^*(d_1)} = d_1 CW^*(\alpha(d_1)). \dots .d_p CW^*(\alpha(d_p)) \tag{5}$$

In the same way, if G' is deduced from G by a removal kernel, each dart d_i of $\sigma'^*(d_1)$ is connected in G to d_{i+1} by $CW(d_i)$. Therefore, the reduction window associated to this vertex is equal to:

$$R_{\sigma'^*(d_1)} = CW(d_1). \dots .CW(d_p) \tag{6}$$

Since each vertex of the initial combinatorial map G corresponds to one pixel, the vertex-reduction windows defined above should correspond to the usual notion of region. However, it remains to show that these regions fulfill the requirements of Definition 2.

If K is a removal kernel, any connecting walk is included in a σ orbit by definition (equation 3). Moreover, each connecting walk $CW(d_i)$ is connected to $CW(d_{i+1})$ by σ (equations 4 and 6). Thus $R_{\sigma'^*(d_1)}$ is a sequence of σ successors included in $\sigma^*(d_1)$. Moreover, the σ-successor of the last dart of $CW(d_p)$, is equal to d_1 (equation 4). Therefore, $R_{\sigma'^*(d_1)}$ is a σ-orbit included in $\sigma^*(d_1)$. By definition of an orbit we have: $R_{\sigma'^*(d_1)} = \sigma^*(d_1)$. The region $R_{\sigma'^*(d_1)} = \sigma^*(d_1)$ is thus trivially connected and contains all its vertices.

If K is a contraction kernel, each sequence $CW^*(\alpha(d_i))$ is connected (equation 2). Moreover, each dart d_i is the σ-successor of the second dart of $CW(\alpha(d_i))$ (equation 2 and 1). The sequence $d_iCW^*(\alpha(d_i))$ is thus connected. Finally, the φ successor of the last dart of each sequence $d_iCW^*(\alpha(d_i))$ is equal to d_{i+1}(equation 4). Each sequence $d_iCW^*(\alpha(d_i))$ in $R_{\sigma'^*(d_1)}$ is thus connected to the following one and $R_{\sigma'^*(d_1)}$ is connected.

Note that using the circular order defined on $\sigma'^*(d_1)$, d_1 is either the σ or φ successor of the last dart of $R_{\sigma'^*(d_1)}$. The region $R_{\sigma'^*(d_1)}$ corresponds thus to a closed connected sequence of darts which defines a cycle in the oriented graph $\mathcal{OG}$ associated to G.

The proof that $R_{\sigma'^*(d)}$ contains its vertices, is based on a study of the connections between the trees of a contraction kernel and the connecting walks. We have in particular the following properties:

Proposition 1 *Given a contraction kernel K, an initial combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and the contracted combinatorial map $G' = (\mathcal{SD} = \boldsymbol{\mathcal{D}} - K, \sigma', \alpha)$, the trees of K satisfy:*

$$\begin{aligned} \forall \mathcal{T} \in \mathcal{CC}(K), \quad \forall d_1 \in \sigma^*(\mathcal{T}) \cap \mathcal{SD} \quad \sigma'^*(d_1) &= \sigma^*(\mathcal{T}) \cap \mathcal{SD} && (7) \\ \mathcal{T} &= \cup_{j=1}^{p} CW^*(\alpha(d_j)) && (8) \end{aligned}$$

with $\sigma'^*(d_1) = (d_1, \ldots, d_p)$.

Equation 7 is demonstrated in [2]. Equation 8 may be deduced from equation 7 using the fact that $\cup_{j=1}^{p} CW^*(\alpha(d_j))$ is a connected set of non surviving darts and is thus included in a particular tree $\mathcal{T}$ of K.

Equation 7 may be understood as follows: The set of surviving darts belonging to the leafs of a tree $\mathcal{T}$ define the adjacency relationships between $\mathcal{T}$ and the other vertices of G. Note that we have a circular order on the σ'-orbit $\sigma'^*(d_1)$. Therefore, the set of surviving darts encoding the adjacency relationships of the tree is ordered according to counter-clockwise orientation when turning around the tree.

Each surviving dart d_i of $\sigma^*(T) \cap \mathcal{SD} = \sigma'^*(d)$ is connected in G to $d_{i+1} = \sigma'(d_i)$ by $d_i.CW^*(\alpha(d_i))$ (Section 3). Therefore, the two surviving darts d_i and d_{i+1} are connected in G by a sequence $CW^*(\alpha(d_i))$ of non surviving darts. Equation 8 shows that the union of such sequences cover the whole tree.

Let us consider one surviving vertex $\sigma'^*(d_1) = (d_1, \ldots, d_p)$ and one tree $\mathcal{T} = \cup_{j=1}^{p} CW^*(\alpha(d_j))$. Since $\mathcal{T}$ is a connected component of K, a dart d belonging to $\sigma^*(\mathcal{T}) \cap K$ must belong to $\mathcal{T}$, otherwise, $\mathcal{T}$ would be connected to an other tree of K. We have thus $\sigma^*(\mathcal{T}) \cap K = \mathcal{T}$. Moreover, since $\mathcal{SD} = \boldsymbol{\mathcal{D}} - K$, we have $\boldsymbol{\mathcal{D}} = \mathcal{SD} \cup K$ and:

$$\begin{array}{rcll} \sigma^*(\mathcal{T}) & = & \sigma^*(\mathcal{T}) \cap \boldsymbol{\mathcal{D}} = \sigma^*(\mathcal{T}) \cap (\mathcal{SD} \cup K) & \\ & = & (\sigma^*(\mathcal{T}) \cap \mathcal{SD}) \cup (\sigma^*(\mathcal{T}) \cap K) & \\ & = & \sigma'^*(d_1) \cup \mathcal{T} & \text{(equation 7)} \\ & = & \sigma'^*(d_1) \cup \bigcup_{j=1}^{p} CW^*(\alpha(d_j)) & \text{(equation 8)} \end{array}$$

where $\sigma'^*(d_1) = (d_1, \ldots, d_p)$.

Therefore the region $R_{\sigma'^*(d_1)}$ associated to the surviving vertex $\sigma'^*(d_1)$ defines an order on the set $(d_1, \ldots, d_p) \cup \bigcup_{j=1}^{p} CW^*(\alpha(d_j)) = \sigma^*(\mathcal{T})$ (see equation 5). Since the operator σ^* is idempotent we have:

$$\sigma^*(R_{\sigma'^*(d)}) = \sigma^*(\sigma^*(\mathcal{T})) = \sigma^*(\mathcal{T}) = R_{\sigma'^*(d)}$$

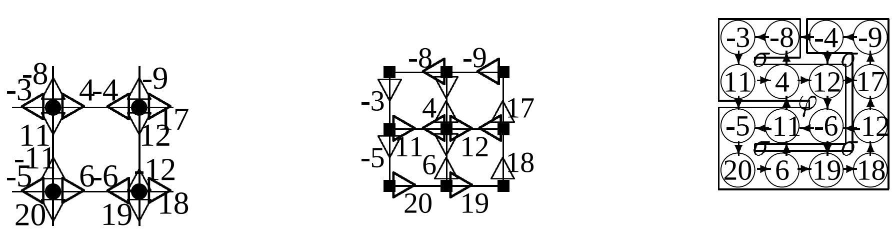
The sequence of darts $R_{\sigma'^*(d)}$ is thus connected and contains its vertices. It is thus a region which defines a connected set of vertices.

Figure 5 illustrates three alternative representations of the region associated to the contracted vertex $\sigma'^*(-8)$ (see central vertex in Figure 2). Note that, all darts in $R_{\sigma'^*(-8)}$ are associated to a triangle in Figure 5(b). However, the name of the darts -4, -6, -11 and -12 is not displayed in this figure in order to not overload it. The vertex $\sigma'^*(-8)$ is defined by the sequence of darts -8, -3, 11,-11,-5,20,19, 18, 17 and -9(Figure 2). Using equation 5 the region $R_{\sigma'^*(-8)}$ is defined as:

$$R_{\sigma'^*(-8)} = \boxed{-8}.\boxed{-3}.\boxed{11.4.12.-6}.\boxed{-11}.\boxed{-5}.\boxed{20.6}.\boxed{19}.\boxed{18.-12}.\boxed{17}.\boxed{-9.-4} \qquad (9)$$

where each box surrounds a sequence $d.CW^*(\alpha(d))$ with $d \in \sigma'^*(-8)$ (Figure 4(a)).

The region $R_{\sigma'^*(-8)}$ is composed of 4 pixels with 4 cracks defining inner boundaries and 8 cracks defining the boundary of the region. We can note on equation 9 that all the edges defining inner boundaries are included in $R_{\sigma'^*(-8)}$. We have indeed, $\alpha^*(4, 11, 6, 12) \subset R_{\sigma'^*(-8)}$. This property is a direct consequence of the fact that a region contains its vertices (Definition 2). Indeed, if an edge $\alpha^*(d)$ defines an inner boundary of a region R both $\sigma^*(d)$ and $\sigma^*(\alpha(d))$ must



(a) $R_{\sigma_1^*(-8)}$ in G (b) $R_{\sigma_1^*(-8)}$ in $\overline{G}$ (c) $R_{\sigma_1^*(-8)}$ in $\mathcal{OG}$

Figure 5: The region $R_{\sigma_1^*(-8)}$ defined by K_1 (see Figure 2)

be included in R. Therefore, d and $\alpha(d)$ must belong to R. Conversely, all darts defining the boundary of a region R cannot have their α-successor in R (see e.g. the darts 17 to 20 in equation 9 and Figure 5). We can also note, on equation 9 (see also Figure 5(b)) that the sequence of darts $-8, -3, -5, 20, 19, 18, 17, -9$ defining the boundary of $R_{\sigma'^*(-8)}$ with a counter-clockwise orientation is included in $R_{\sigma'^*(-8)}$. Moreover, the order of this sequence is respected in $R_{\sigma'^*(-8)}$. This last property has been verified in all our experiments but is not yet fully demonstrated.

5 Conclusion

We have defined in this paper the notion of regions within the combinatorial pyramid framework. This result allows us to either draw the image partition associated to one level of the pyramid or to extract parameters from regions. Moreover, the reduction window R of any high level pixel has been found to form a directed Hamiltonian circuit in the sub-graph of $\mathcal{OG}$ restricted to R. However, the region defined in this paper are based on connecting walks which correspond to the notion of reduction window within the pyramid framework. A general definition based on the receptive fields of darts should be studied. Moreover, we also plan to study finer properties of regions. Such results should allow us to retrieve the set of pixels of one region or its boundary without traversing all the darts of the region. Finally, combinatorial maps being formally defined in any dimensions, these results should be extended to higher dimensions in order to define nD Combinatorial Pyramids.

References

[1] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognit Letter.*, 11(9):605–617, Sept. 1990.

[2] Luc Brun and Walter Kropatsch. Pyramids with combinatorial maps. Technical Report PRIP-TR-057, PRIP, TU Wien, 1999.

[3] Luc Brun and Walter Kropatsch. Contraction kernels and combinatorial maps. In Jean Michel Jolion, Walter Kropatsch, and Mario Vento, editors, 3^{rd} *IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, pages 12–21, Ischia Italy, May 2001. IAPR-TC15, CUEN.

[4] Peter Burt, Tsai-Hong Hong, and Azriel Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchial computation. *IEEE Transactions on Sustems, Man and Cybernetics*, 11(12):802–809, December 1981.

[5] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142(No. 6):pp. 366–374, December 1995.

[6] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, APRIL 1991.