A causal extraction scheme in top-down pyramids for large images segmentation

Romain Goffe, Guillaume Damiand, Luc Brun

▶ **To cite this version:**

Romain Goffe, Guillaume Damiand, Luc Brun. A causal extraction scheme in top-down pyramids for large images segmentation. In 13th International Workshop On Structural and Syntactic Pattern Recognition (SSPR'10), Aug 2010, Cesme, Turkey. pp.264-274, 10.1007/978-3-642-14980-1_25 . hal-00567656v2

HAL Id: hal-00567656
https://hal.science/hal-00567656v2

Submitted on 26 Apr 2011

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

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

A causal extraction scheme in top-down pyramids for large images segmentation

Romain Goffe[1], Guillaume Damiand[2], and Luc Brun[3]

[1] SIC-XLIM, Université de Poitiers, CNRS, UMR6172, F-86962, Futuroscope Chasseneuil, France
goffe@sic.univ-poitiers.fr
[2] LIRIS, Université de Lyon, CNRS, UMR5205, F-69622, Villeurbanne, France
guillaume.damiand@liris.cnrs.fr
[3] GREYC, ENSICAEN, CNRS, UMR6072, 6 Boulevard du Maréchal Juin, F-14050, Caen, France
luc.brun@greyc.ensicaen.fr

Abstract Applicative fields based on the analysis of large images must deal with two important problems. First, the size in memory of such images usually forbids a global image analysis hereby inducing numerous problems for the design of a global image partition. Second, due to the high resolution of such images, global features only appear at low resolutions and a single resolution analysis may loose important information. The tiled top-down pyramidal model has been designed to solve this two major challenges. This model provides a hierarchical encoding of the image at single or multiple resolutions using a top-down construction scheme. Moreover, the use of tiles bounds the amount of memory required by the model while allowing global image analysis. The main limitation of this model is the splitting step used to build one additional partition from the above level. Indeed, this step requires to temporary refine the split region up to the pixel level which entails high memory requirements and processing time. In this paper, we propose a new splitting step within the tiled top-down pyramidal framework which overcomes the previously mentioned limitations.

Key words: Irregular pyramid; Topological model; Tiled data structure; Combinatorial map;

1 Introduction

High resolution image analysis usually entails memory issues that prevent them from being processed by common models. Moreover, in multi-resolution images, the amount of details at full resolution is likely to mask global features which only appear at lower resolutions. For instance, applicative fields such as whole slide microscopic imaging produce large multi-resolution images with resolutions up to $40\,000 \times 40\,000$: low resolutions let appear global features such as tissues delimitations while high resolutions allow to discern the different phases of mitosis within cells. As a result, analyzing such images implies a hierarchical representation with memory constraint.

The segmentation of an image defines an image partition into connected regions. Models for such partitions usually encode either geometrical or topological features of the partition. Operations involving both types of information are thus hard or costly to implement. For example, RAG-based data structures lack efficient access to regions' geometry. This drawback has entailed the design of topological maps [BDM03, DBF04] for an efficient representation of both geometrical and topological information while allowing modifications of a partition through split and merge operations. Yet, they cannot apply to multi-resolution images since they do not encode a hierarchy of partitions.

Quadtrees and regular pyramids' frameworks provide a multi-resolution description of the image and a hierarchical segmentation scheme [BCR90] inducing a hierarchy of regions that may be defined onto such models. However, both models present several drawbacks: a given regular pyramid may fail to encode connected regions of any size and shape nor provide an efficient access to the neighborhood of a region. Moreover, both quadtrees and regular pyramids do not ensure that connected regions defined at a given level remain connected at the level below. The irregular pyramid framework has been introduced by [Mee89, MMR91] to overcome these limitations with different segmentation schemes such as [JM92, Kro95]. Finally, in order to access both geometrical and topological information, [BK03, GSDL06] proposed a model of irregular pyramids composed of combinatorial maps. When applied to high resolution images, the bottom-up construction scheme of combinatorial pyramids raises at least two issues: memory usage and relevance of extracted information. Indeed, a bottom-up scheme starts from an explicit encoding of the whole initial partition: for large images this requires a large amount of memory especially if additional levels must also be computed. Moreover, in hierarchical data analysis, extracted information is usually more relevant if the construction scheme allows to use a region to influence the way its children (defined at a higher resolution) are processed. As a result, [GDB09] have introduced the tiled top-down framework for combinatorial pyramids.

A tiled top-down pyramid is a hierarchical model based on topological maps [BDM03, DBF04] and thus, provides an efficient access to both geometrical and topological information. A top-down pyramidal model allows to reduce memory usage by encoding upper levels in the pyramid and refining only areas of interest. Moreover, the subdivision in tiles allows to bound the required amount of memory. Yet, its main drawback comes from its construction process [GBD09]: in order to refine a region, a first step splits it into basic regions enclosing single pixels before the application of a merging step. This step may thus require a large amount of available memory to make temporary regions. Since the main operation in a top-down scheme consists in regions' splitting, we have explored alternative splitting techniques for combinatorial models. Different approaches have been proposed such as insertion operations [BD99] or incremental extractions [DBF04, BDM03] but those methods are not designed for a causal hierarchical model [GCM06]. This causal property is fundamental within the tiled top-down pyramidal framework since it ensures the

existence of a hierarchy.

In this paper, we propose a new method for the construction of tiled top-down pyramids which avoids the temporary split of a region into basic regions enclosing a single pixel while preserving the causality of the pyramid. In Section 2, we present the different topological models used to define a tiled top-down pyramid. In Section 3, we detail our causal extraction for such pyramids. Finally, experiments and segmentation results are proposed in Section 4 in order to emphasize the advantage of our method.

2 Recalls

2.1 Combinatorial Maps

In two dimensions, a combinatorial map (noted 2-map) is a set of vertices, edges and faces that encodes the subdivision and incidence relationships of a topological space [Lie89]. A complete decomposition of an image results in a set of abstract basic elements called *darts*. We introduce two operators noted β_i, $i \in \{1, 2\}$ that apply on darts in order to represent adjacency relationships (Figure 1).

Definition 1 (2-dimensional combinatorial map)**.** *A two-dimensional combinatorial map M (or 2-map) is a triplet $M = (D, \beta_1, \beta_2)$ where:*

(1) D is a finite set of darts;

(2) β_1 is a permutation[1] *on D;*

(3) β_2 is an involution[2] *on D.*

Intuitively, we can consider a map as a planar graph where β_i operators explicitly define the relationships between edges and where darts allow to differentiate the two extremities of an edge (a dart may be assimilated to a half-edge). In practice, the β_1 permutation allows to turn around a face: it links a dart of a face to the next one encountered while turning clockwise around it. The β_2 involution separates two adjacent faces: it links a dart to the other dart that belongs to the same edge but has an opposite orientation. For instance, in Figure 1(d), $\beta_1(3) = 4$ and $\beta_2(4) = 5$. As a result, a 2-map is a connected set of cells of 0, 1, and 2 dimensions. For $i \in \{0, 1, 2\}$, an i-cell respectively denotes a vertex, an edge and a face. The degree of an i-cell is its number of distinct incident $(i + 1)$-cells.

2.2 Topological Maps

Since combinatorial maps only describe topological relationships, an extension of the model which also encodes geometrical information has been introduced for a full

[1] A *permutation* is a one to one mapping from S onto S.

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

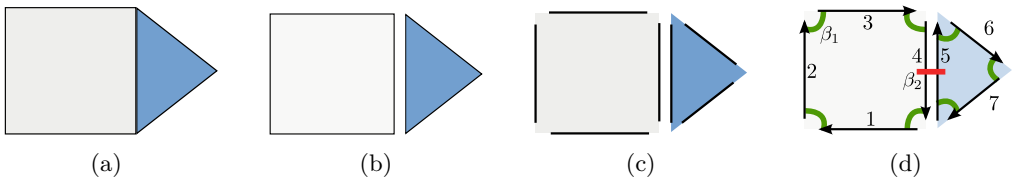


Figure 1: Combinatorial maps: construction by successive decompositions. (a) Original image; (b) Decomposed faces; (c) Decomposed edges; (d) 2-Map: arrows represent darts, β_1 and β_2 operators are respectively represented by arcs and segments.

representation of a partition: the model of topological map [BDM03, DBF04]. A topological map combines three distinct models: a 2-map that encodes topological relationships, a matrix of *interpixel elements* [Kov89, KKM90] that encodes the geometry of the partition elements, and a tree of regions for inclusion relationships. These three models are illustrated in Figure 2 and described below.

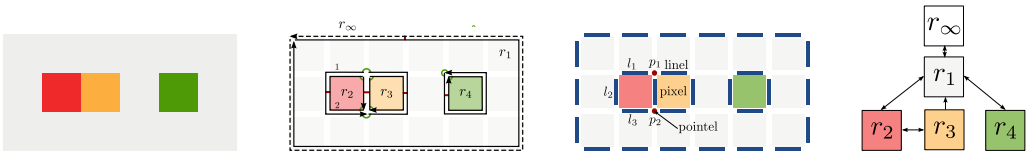
Minimal combinatorial map As illustrated in Figure 2(b), a 2-map encodes topological relationships through β_1 and β_2 operators (Section 2.1). The combinatorial map is *minimal* in number of cells: there is not any vertex with a degree lower or equal to 2 and therefore, the removal of any element would change the topology. For implementation purposes, darts and regions are linked together: a dart knows the region it belongs to and a region knows a representative dart (arbitrary chosen on the external border of the region). Note that the infinite region may be omitted in some figures for visibility reasons.

Matrix of interpixel elements Pointels, linels and pixels [KKM90] represent the geometry of a partition. Associating geometrical information to a topological element is an operation called *embedding*. Similarly to vertices, edges and faces, we respectively refer to pointels, linels and pixels as i-cells, $i \in \{0, 1, 2\}$. We respectively denote by $pointel(d)$ and $linel(d)$ the first pointel and linel of the embedding of a dart d. For example, in Figure 2(c), the embedding of the edge (1,2) is the sequence of linels (l_1, l_2, l_3); $linel(1) = l_1$, $pointel(2) = p_2$; $degree(p_1) = degree(p_2) = 3$.

Tree of regions The tree of regions describes inclusion relationships: a region is the father of the regions it contains. In Figure 2(d), r_1 contains r_2, r_3 and r_4, r_2 and r_3 are adjacent. The root of the tree encodes the background of the image and is called the infinite region (noted r_∞).

2.3 Tiled Top-down Framework for Combinatorial Pyramids

A hierarchical extension of the topological map model is proposed by [GBD09]. Contrary to bottom-up methods, this framework uses a top-down approach which



(a) (b) (c) (d)

Figure 2: Topological map: three complementary models for image representation. (a) Original image; (b) Combinatorial map for topological relationships. Dotted arrow denotes the dart of the infinite region; (c) Interpixel matrix for geometrical encoding: pointels and linels are represented by bold circles and segments; (d) Tree of regions.

induces a segmentation process based on a rough partition refined at further levels: it results in a major memory reduction since regions may only be encoded at the top level of the pyramid. Moreover, it allows to take advantage of the *focus of attention* over interesting regions: the segmentation of a region can be adapted according to the features of its parent. Despite this memory reduction, [GDB09] proposes a subdivision of the levels into topological tiles in order to bound the amount of required memory.

A topological tile is a topological map with an additional involution β'_2 which applies on darts belonging to a border shared by two adjacent tiles [GDB09] to ensure their topological connection. The juxtaposition of topological tiles composes a tiled topological map. Such a map may contain fictive elements along the tiles' borders when, according to a given merging criterion, pixels on both sides of a tile's border belong to a same region. Linels encoding these fictive borders are marked by a flag indicating their fictive state. Tiled combinatorial maps (Definition 2) redefine the operators β_1 and β_2 [GDB09] to abstract those fictive elements (Figure 3(c)).

Definition 2 (Tiled combinatorial map)**.** *Let T be a set of connected topological tiles $T = \{t(i,j)\}_{(i,j)\in\{0,\dots,W\}\times\{0,\dots,H\}}$. Let D be the set of darts of T with a real embedding. A tiled combinatorial map M is a triplet $M = (D, \delta_1, \delta_2)$ where, $\forall d \in D$:*

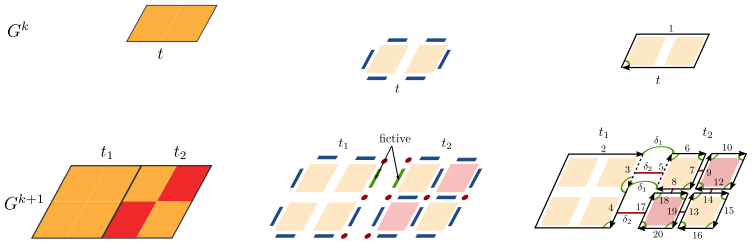
(1) δ_1 is a permutation on D such as:

$$\delta_1(d) = \beta_1((\beta'_2\beta_1)^n(d)) \text{ with } n = min\{p \in \mathbb{N} \mid linel(\beta_1((\beta'_2\beta_1)^p(d)) \text{ is real}\}$$

(2) δ_2 is an involution on D such as:

$$\delta_2(d) = \begin{cases} \beta'_2(d) & \text{if } \beta'_2(d) \text{ exists} \\ \beta_2(d) & \text{otherwise} \end{cases}$$

A tiled top-down topological pyramid is a stack of finer and finer partitions denoted by $P = \{G^k\}_{k\in\{0,\dots,n\}}$ where G^{k+1} is a tiled combinatorial map (Definition 2) deduced from G^k by splitting operations. Within a pyramid, a tile is denoted by $t(i,j,k)$ where (i,j,k) are the coordinates (i,j) of t at level G^k. Besides, the pyramid



dart	δ_1	δ_2	*up*	*down*
1	1	∞	-	2
2	6	∞	1	-
3	4	5	-	-
4	2	17	-	-
5	6	3	-	-
6	7	∞	1	-
7	8	9	-	-
8	4	18	-	-
9	10	7	-	-
10	11	∞	1	-
11	12	∞	1	-
12	9	14	-	-
13	14	19	-	-
14	15	12	-	-
15	16	∞	1	-
16	13	∞	1	-
17	18	4	-	-
18	19	8	-	-
19	20	13	-	-
20	17	∞	1	-

Figure 3: Representation of a top-down pyramid composed of two levels G^k and G^{k+1}. (a) Original image: a tile t is decomposed in two tiles t_1 and t_2 ($ratio = 2 \times 2$); (b) Interpixel matrix: fictive borders appear between t_1 and t_2; (c) Tiled combinatorial maps: δ_1 and δ_2 are represented by arcs and segments. Dotted arrows denote darts with a fictive embedding; (d) Relationships between darts.

may swap or load tiles between memory and disk and spread global modifications: if an operation modifies a tile's border, adjacent tiles that are either on disk or in memory should be updated. Note that a stack of successively refined partitions differs from the notion of resolution used within the regular pyramid framework: top-down pyramids can be constructed either from single or multi-resolution images. Although, the pyramid mixes both regular and irregular notions, the top-down model remains an irregular pyramid since it handles fictive borders between the tiles. The resulting sequence of partitions is a *causal structure* [GCM06] where hierarchical relationships are encoded through *up/down* relationships between tiles, darts and regions.

A first strategy to build the pyramid is to start from a single region and refine it according to segmentation criteria. The operation is performed in three steps. First, the last level is duplicated and *up/down* relationships are set. Second, a *splitting criterion* indicates the regions to refine. Those regions are split into a set of basic regions enclosing a single pixel. Third, the basic regions are merged according to a *merging criterion*. Since any couple of adjacent regions may be merged, this refinement step may encode any subdivision of the parent region. Yet, this solution presents a major drawback: the splitting step in one region per pixel is a bottom-up refinement that encodes every pixel of the split region. Such an operation implies useless calculus and important memory requirements.

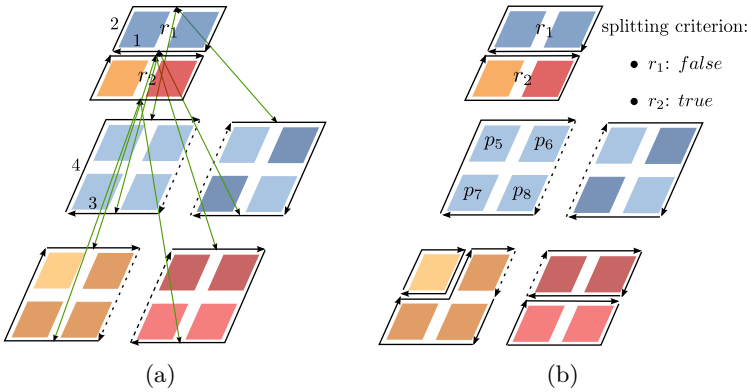


Figure 4: Extraction constraints for a tiled top-down pyramid. G^k is a single tile t decomposed in G^{k+1} into 4 subtiles ($ratio = 2 \times 2$). (a) Causal constraint: preserves existing borders; (b) Top-down constraint: the focus of attention only refines regions whose *splitting criterion* is *true*. Dotted arrows denote fictive borders.

3 A Causal Extraction Scheme for Tiled Top-down Pyramids

Our main objective consists in avoiding the bottom-up refinement step temporary creating one region per pixel. We propose with Algorithm 1 a hierarchical extension of the extraction scheme introduced by [DBF04]. In order to adapt it to the tiled top-down pyramidal framework, we must fulfill two main constraints:

- the *causal constraint* entails that existing borders are preserved from one level to an other: any border defined in G^k must exist in G^{k+1} and darts and regions of G^k must be connected to their children in G^{k+1}.
- the *top-down constraint* (focus of attention): since we use a splitting criterion which determines whether a region should be refined in the next level, no border should be created within regions whose splitting criterion is *false*.

Those two constraints are illustrated in Figure 4. In Figure 4(a), the causality is ensured as each border defined in G^k exists in G^{k+1} with *up/down* relationships set accordingly. For example, when the darts 3 and 4 are created, they must be linked with their respective parents 1 and 2. In Figure 4(b), the focus of attention for the top-down construction is respected as regions whose splitting criterion is set to *false* are not refined: no border is inserted in $down(r_1)$ while r_2 is refined in G^{k+1}.

The global scheme of the extraction algorithm is the following. All the pixels of a tile t in G^{k+1} are traversed with a scanline traversal from top-left to bottom-right corner. For each pixel p, we create the region enclosing it (line 1 of Algorithm 1) which results in the insertion of two new borders between the p and its *top* and *left* neighbors. Then, we determine whether those borders should be kept or removed

Algorithm 1: Causal extraction algorithm.

Data: A tile t in G^k.
Result: Extraction of the sons of t in G^{k+1}.
$S_1 \leftarrow$ *embedding structure* of t;
$S_2 \leftarrow$ *regions structure* of t;
foreach *son* t' *of* t *in* G^{k+1} **do**
  foreach *pixel* $p(x, y)$ *in* t' **do**
    $p \leftarrow p(x, y)$, $p' \leftarrow p(x-1, y)$, $p'' \leftarrow p(x, y-1)$;
1   Create region r enclosing p;
2   **if** $SameRegion(p, p', S_1, S_2)$ **then**
      Remove border between $region(p)$ and $region(p')$;
3   **if** $SameRegion(p, p'', S_1, S_2)$ **then**
      Remove border between $region(p)$ and $region(p'')$;
4   Set *up/down* relationships for r and its darts;
5 Simplify and compute the tree of regions for t';

(line 2-3) by calling Algorithm 2. First, our two constraints must be respected: a border is kept if it corresponds to an existing border in G^k (line 1 of Algorithm 2) and is removed if it corresponds to a forbidden refinement (line 2). Once those two conditions are verified, a merging criterion determines if the border is kept (line 3). Since a topological map aims at representing regions, the merging criterion has to define a partition: we may use a quantization of the tile *up* but conversely, a criterion based on a minimum gradient would lead to the creation of an inconsistent map with dangling edges. The last step finalizes the extraction by removing degree two vertices and computing the tree of regions (line 5 of Algorithm 1) as described in [DBF04]. In the following, we implicitly use projections between levels: if p is a pixel of G^{k+1}, $region(p)$ in G^k refers to the region of the projection of pixel p in G^k. In order to answer the first two conditions of Algorithm 2, two external structures related to $up(t)$ called *embedding structure* and *regions structure* are required and detailed below.

Embedding structure This structure establishes the correspondence between darts and their embedding in a tile by mapping each linel to its dart. Thus, we can determine from the embedding of a dart d in G^{k+1} if there exists a dart d' in G^k whose embedding is down-projected onto the one of d. In this case, the border is kept and we establish *up/down* relationships between $d = down(d')$ and $d' = up(d)$ (line 4 of Algorithm 1).

Regions structure Similarly to an image of labels, this structure maps each tile's pixel to its region. It allows to know the splitting criterion of a pixel p in the above level. For instance in Figure 4(b), we can determine that $region(p_7)$ in G^k is r_1

Algorithm 2: SameRegion.

Data: Two adjacent pixels p and p' of a tile t at G^{k+1}.
embedding structure and *regions structure* of $up(t)$ at G^k.
Result: true if p and p' belong to the same region.
$r \leftarrow region(p)$ in G^k, $r' \leftarrow region(p')$ in G^k;
if $r \neq r'$ **then**
1 **return** false;
if $splitting_criterion(r) = false$ **then**
2 **return** true;
3 **return** $merging_criterion(p, p')$;

Table 1: Memory usage and runtime comparisons between the burst-merge refinement construction and the causal extraction process for a multi-resolution histological image.

level size (pixels)	number of darts	number of regions	burst-merge runtime (s)	burst-merge ram (MB)	causal extract. runtime (s)	causal extract. ram (MB)
4 000×3 036	483 662	187 033	181	104	105	90
8 000×6 072	962 368	392 395	354	106	272	90
32 000×24 291	4 670 978	1 767 890	4 611	100	3 869	92

whose splitting criterion is *false* (r_1 must not be refined) so we do not keep the border between p_7 and p_5.

4 Experiments

This section has two main objectives: show the advantage of our causal extraction algorithm compared to the initial burst-merge refinement and present first segmentation results obtained on histological images.

In Table 1, we provide runtime[3] and memory usage for our causal extraction algorithm (column 6-7) compared to a burst-merge refinement (column 4-5) with a fixed tile size of 512×512. We can notice that our extraction method globally improves the computational time from 15 to 40% with a reduction of memory usage around 10%. The computational time is mostly due to pixel traversal (to get colorimetric information for the regions) and segmentation criteria computation.

We illustrate the results of our extraction in Figure 5. This example shows the tiled map of the same small area of an histological image at different resolutions $(4\,000 \times 3\,036, 8\,000 \times 6\,072, 32\,000 \times 24\,291)$. The segmentation is based on an image quantization algorithm in two classes [KMN+02].

[3]The model is implemented in C++ and computations are carried out on an Intel E5300@2GHz with 2GB RAM.

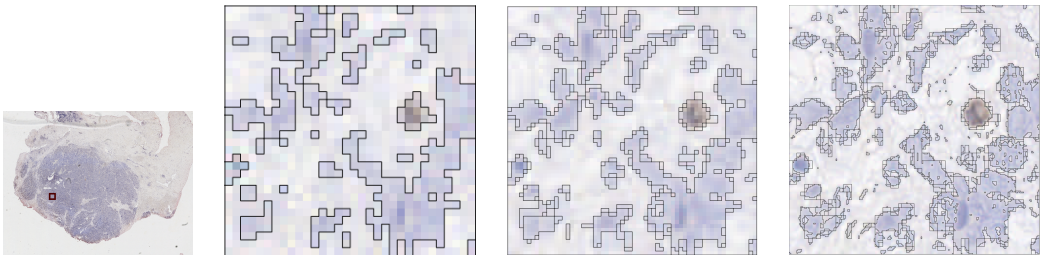


Figure 5: Causal extraction of a tiled top-down pyramid from a multi-resolution histological image. From left to right: low resolution of the original image, three different levels of the pyramid with increasing resolutions encoding a same small area.

5 Conclusion

In this paper, we have presented an alternative construction scheme for the extraction of tiled top-down pyramids that overcomes the main drawbacks of the previous method. First, our method requires less computational time and memory usage. Second, it still preserves the top-down hierarchical relationships of the model. Therefore, our method can favorably act as a replacement for the burst and merge refinement step during the construction of a tiled top-down pyramid. Finally, our causal extraction algorithm presents interesting perspectives for parallel computation that we plan to implement in our future work.

Acknowledgements

This research is part of the FoGrImMi project, supported by the ANR foundation under grant ANR-06-MDCA-008-01/FOGRIMMI.

References

[BCR90] M. Bister, Jan Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617, 1990.

[BD99] J.P. Braquelaire and J.P. Domenger. Representation of segmented images with discrete geometric maps. *Image and Vision Computing*, 17(10):715 – 735, 1999.

[BDM03] Luc Brun, Jean-Philippe Domenger, and Myriam Mokhtari. Incremental modifications of segmented image defined by discrete maps. *J. Visual Communication and Image Representation*, 14(3):251–290, 2003.

[BK03] Luc Brun and Walter G. Kropatsch. Combinatorial pyramids. In *ICIP (2)*, pages 33–36, 2003.

[DBF04] Guillaume Damiand, Yves Bertrand, and Christophe Fiorio. Topological model for two-dimensional image representation: definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2):111–154, 2004.

[GBD09] Romain Goffe, Luc Brun, and Guillaume Damiand. A top down construction scheme for irregular pyramids. In *VISSAPP (1)*, pages 163–170, 2009.

[GCM06] Laurent Guigues, Jean Pierre Cocquerez, and Hervé Le Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.

[GDB09] Romain Goffe, Guillaume Damiand, and Luc Brun. Extraction of tiled top-down irregular pyramids from large images. In *13th International Workshop on Combinatorial Image Analysis (IWCIA'09)*, pages 123–137. RPS, Singapore, 2009.

[GSDL06] Carine Grasset-Simon, Guillaume Damiand, and Pascal Lienhardt. n-d generalized map pyramids: Definition, representations and basic operations. *Pattern Recognition*, 39(4):527–538, 2006.

[JM92] Jean-Michel Jolion and Annick Montanvert. The adaptative pyramid: A framework for 2d image analysis. *CVGIP*, 55(3):339–348, May 1992.

[KKM90] E. Khalimsky, R. Kopperman, and P.R. Meyer. Boundaries in digital planes. *Journal of Applied Mathematics and Stochastic Analysis*, 3(1):27–55, 1990.

[KMN+02] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. on PAMI*, 24(7):881–892, 2002.

[Kov89] V. A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46(2):141–161, 1989.

[Kro95] Walter Kropatsch. Building irregular pyramids by dual graph contraction. *IEE Proceedings - Vision, Image, and Signal Processing*, 142:366–374, December 1995.

[Lie89] Pascal Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Symposium on Computational Geometry*, pages 228–236, 1989.

[Mee89] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, 1989.

[MMR91] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):307–316, 1991.