COMBINATORIAL PYRAMIDS

Luc Brun

LERI
I.U.T. de Reims 51.059 Reims
France

*Walter Kropatsch**

Institute for Computer-aided Automation
Pattern Recognition and Image Processing Group
Vienna Univ. of Technology
Austria

ABSTRACT

An irregular pyramid consists of a stack of successively reduced graphs. Each smaller graph is deduced from the preceding one by the contraction or the removal of a set of edges. Using a fixed decimation ratio we need approximately $\mathcal{O}(\log(\text{image size}))$ graphs to encode the whole pyramid. A combinatorial map encodes a planar graph thanks to two permutations encoding the edges and their orientation around the vertices. We present in this article an encoding of a combinatorial pyramid which allows to fold the whole pyramid in the base level layer and provides at the same time a measure of the relevance of every pixel. This encoding is used to retreive any reduced combinatorial map of the pyramid from its base and to compute the borders of the partition encoded by the combinatorial maps.

1. INTRODUCTION

Irregular pyramids are widely used in image segmentation and image analysis frameworks to encode a hierarchy of partitions. 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 non surviving vertex to a surviving one [1]. If the initial graph is planar its reduced versions are also planar. Moreover, given an image, if each vertex of the initial graph is associated to one pixel, the set of initial vertices mapped to a surviving vertex defines a region of the image.

The boundaries between two adjacent regions are encoded by the edges of the reduced graphs. Using simple graphs (without multiple edges between vertices nor self-loops) multiple boundaries between two regions are mapped into only one edge. This drawback may be overcome by using the Dual graph reduction scheme [2]. Within this framework, the reduction operation is performed in two steps: First, the contraction of a set of edges identifies a set of vertices. This operation may create redundant edges such as empty self-loops or double edges [2]. These redundant edges are characterized in the dual of the graph and removed by a set of edge removals. Using such a reduction scheme each edge in the reduced graph corresponds to one boundary between two regions. Moreover, inclusion relationships may be differentiated from adjacency ones in the dual graph.

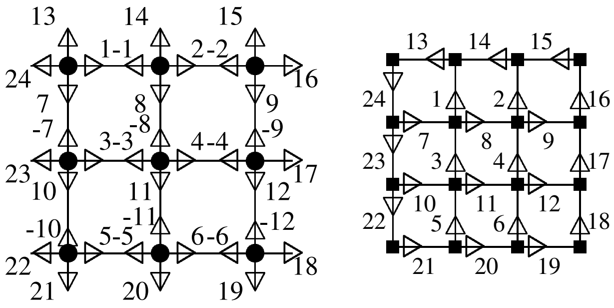
Combinatorial Pyramids inherit all the useful properties from the dual graph pyramids with the addition that they also preserve the local orientation of edges around vertices and faces. The expected advantages of such hierarchies within the image analysis framework are presented in [3].

The remaining of this paper is as follows: We present in section 2 the combinatorial map model together with its main properties. In section 3 we define the contraction and removal operations within the combinatorial map framework. In Section 4 we present a folding of the Combinatorial Pyramid based on an encoding of the maximal level where an edge survives. Using such a folding the relevance of an edge according to a particular reduction scheme is stored in the base level combinatorial map. We thus combine local and global information within a same representation. This encoding is used to build any combinatorial map of the pyramid directly from the base level and to compute the borders of the associated partition.

2. COMBINATORIAL MAPS

A combinatorial map [4] may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. Fig. 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. The orbits of a second permutation σ encode the set of darts encountered when turning counterclockwise

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



(a) $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ (b) $\overline{G} = (\boldsymbol{\mathcal{D}}, \varphi, \alpha)$

Fig. 1. A 3×3 grid encoded by a combinatorial map G (a). The dual combinatorial map is represented in (b).

around a vertex. The central vertex in Fig. 1(a) is for example encoded by the σ-orbit $(-8, -3, 11, 4)$. We have thus $\sigma(-8) = -3$, $\sigma(-3) = 11$, $\sigma(11) = 4$ and $\sigma(4) = -8$.

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 $\pi^*(d)$.

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

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

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

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, its dual is defined by $\overline{G} = (\boldsymbol{\mathcal{D}}, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$ (Fig. 1(b)). The orbits of the permutation φ encode the set of darts encountered when turning around a face of G or a vertex of $\overline{G}$ (see e.g. the φ-orbit $(1, 8, -3, -7)$ encoding the top-left face in Fig. 1(a)). Note that, if the initial combinatorial map G is associated to a $m \times n$ planar sampling gird, its dual $\overline{G}$ is associated to a $(m + 1) \times (n + 1)$ planar sampling grid [3] (Fig. 1).

3. COMBINATORIAL PYRAMIDS

As in the dual graph pyramid scheme [2] (Section 1) a combinatorial pyramid is 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, we forbid the removal of bridges and the contraction of self-loops. Such contractions may be avoided by using a contraction kernel defined as a forest of the initial combinatorial map. Given a contraction kernel defined by a set K of darts to be contracted, the set of surviving darts denoted by $\mathcal{SD}$ is equal to $\boldsymbol{\mathcal{D}} - K$ where $\boldsymbol{\mathcal{D}}$ denotes the initial set of darts. In the same way, the removal of bridges may be avoided by using a removal kernel defined as a forest of the dual combinatorial map. Note that, while a contraction kernel is application dependent a removal kernel is automatically defined from one combinatorial map. Indeed, within our reduction scheme a contraction kernel specifies a set of regions to be merged while a removal kernel is restrained to the removal of double edges and empty self-loops defining redundant boundaries between the merged regions.

The creation of the reduced combinatorial map from a contraction or a removal kernel is performed in parallel by using dart's reduction window [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 reduction window of d is either equal to:

$$RW(d) = d, \sigma(d), \ldots, \sigma^{n-1}(d)$$

with $n = Min\{p \in \mathbb{N}^* \mid \sigma^p(d) \in \mathcal{SD}\}$ if K is a removal kernel or

$$RW(d) = d, \varphi(\alpha(d)), \ldots, \varphi^{n-1}(\alpha(d))$$

with $n = Min\{p \in \mathbb{N}^* \mid \varphi^p(\alpha(d)) \in \mathcal{SD}\}$, if K is a contraction kernel.

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

$$\sigma'(d) = \begin{cases} \sigma(d_p) & \text{if } K \text{ is a removal kernel} \\ \varphi(d_p) & \text{if } K \text{ is a contraction kernel} \end{cases} \tag{2}$$

Note that the reduction window of a surviving dart d connects $d \in G'$ to a sequence of darts in the initial combinatorial map G. The notion of dart's reduction window connects thus two successive levels of the pyramid and corresponds to the usual notion of reduction window [1].

4. FOLDING AND UNFOLDING THE PYRAMID

Dart's reduction windows allow us to reduce a combinatorial map either by contraction or removal kernels. Starting from an initial combinatorial map $G_0 = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and given a sequence of kernels $K_1, \ldots, K_n$ we can thus build the sequence of reduced combinatorial maps $G_0, G_1, \ldots, G_n$ encoding explicitly the pyramid as a stack of successively reduced combinatorial maps. The aim of this section is to define an implicit encoding of the pyramid by additional attributes stored in the initial combinatorial map G_0.

As mentioned in Section 3, the notion of darts reduction window establishes a father-child relationship between two successive levels of the pyramid. The transitive closure of this father-child relationship defines the notion of receptive field. The receptive field at level i: $RF_i(d) = d_1.\ldots.d_p$ of a dart d belonging to $G_i = (\mathcal{SD}_i, \sigma_i, \alpha)$ is defined by [3] :

$$\begin{array}{l} d_1 = d,\ d_2 = \sigma(d) \text{ and} \\ \text{for each } j \text{ in } \{2, \ldots, p\} \\ d_j = \left\{ \begin{array}{ll} \varphi(d_{j-1}) & \text{if } d_{j-1} \text{ has been contracted} \\ \sigma(d_{j-1}) & \text{if } d_{j-1} \text{ has been removed} \end{array} \right. \end{array} \quad (3)$$

Dart's receptive fields may be understood as the transitive closure of the hierarchical relationship defined by reduction windows. Both sequences should thus satisfy similar properties. Indeed, given one dart $d \in \mathcal{SD}_i$, such that $RF_i(d) = d.d_1, \ldots, d_p$ we have [3]:

$$\sigma_i(d) = \left\{ \begin{array}{ll} \varphi(d_p) & \text{if } d_p \text{ has been contracted} \\ \sigma(d_p) & \text{if } d_p \text{ has been removed} \end{array} \right. \quad (4)$$

Note that equations 4 is similar to equation 2 defined for reduction windows.

The receptive field $RF_i(d) = d.d_1.\ldots.d_p$ of a dart $d \in G_i$ connects d to a sequence of darts in the base level combinatorial map. This notion corresponds thus to the usual notion of receptive field. We additionally showed [3] that given the dart's receptive field $RF_i(d) = d.d_1.\ldots.d_p$, with $d \in \mathcal{SD}_i$, we have $\{d_1, \ldots, d_p\} \subset \cup_{j=1}^{i} K_j$. The first dart of a receptive field is thus the only one which survives up to level i.

4.1. Folding the Pyramid

According to equation 3, a dart's receptive field may be traversed if we know the initial dart in $\mathcal{SD}_i$ defining it and if we determine the operation which has reduced each dart of the sequence. The above remark suggests an encoding of the pyramid by two functions:

1. one function $state$ from $\{1, \ldots, n\}$ to the binary states $\{Contracted, Removed\}$ which specifies the type of each kernel.

2. One function $level$ defined for all darts in $\mathcal{D}$ such that $level(d)$ is equal to the maximal level where d survives:

$$\forall d \in \mathcal{D}\ level(d) = \ Max\{i \in \{1, \ldots, n+1\} \mid d \in \mathcal{SD}_{i-1}\}$$

a dart d surviving up to the top level has thus a level equal to $n + 1$. Note that the level of an initial vertex (i.e. a pixel) may be implicitly defined as the minimal level of its darts.

One encoding of the pyramid based on the functions $level$ and $state$ is called an implicit encoding of the pyramid.

Using a pyramid made of n levels whose apex is reduced to a self-loop, the memory requirements of an implicit and explicit encoding of the pyramid are respectively equal to [3]: $\frac{1}{2}|\mathcal{D}| \log_2(n)$ and $2|\mathcal{D}|n \log_2(k)$ bits where $\mathcal{D}$ denotes the initial set of darts and k the reduction factor. The memory requirement of an implicit encoding of the pyramid varies thus as a logarithmic function of its height while the memory requirement of an explicit encoding of all the combinatorial maps is linear according to the height n of the pyramid.

4.2. Unfolding the Pyramid

Given the function $level$, the sequence of kernels and the set of surviving darts may be retrieved by the following equations [3]:

$$\forall i \in \{1, \ldots, n\} \left\{ \begin{array}{lll} K_i & = & \{d \in \mathcal{D} \mid level(d) = i\} \\ \mathcal{SD}_i & = & \{d \in \mathcal{D} \mid level(d) > i\} \end{array} \right. \quad (5)$$

Moreover, we can note that in equation 3 a dart d_{j-1} is contracted if $state(level(d_{j-1})) = Contracted$ and removed if $state(level(d_{j-1})) = Removed$. This remark allows us to traverse the receptive field of any dart $d \in \mathcal{SD}_i$ using equation 3 and the functions $level()$ and $state()$.

Given a dart's receptive field $RF_i(d) = d.d_1 \ldots, d_p$, if we define d_{p+1} by $\varphi(d_p)$ if d_p is contracted and $\sigma(d_p)$ is d_p is removed, d_{p+1} is equal to $\sigma_i(d)$ (equation 4). We have thus in both cases, $d_{p+1} \in \mathcal{SD}_i$ and $level(d_{p+1}) > i$ (equation 5). This last remark allows us to determine the last dart of dart's receptive fields as the one whose successor defined by equation 3 has a level strictly greater than i. Using this last property we designed [3] an algorithm based on equation 3 which traverses the receptive field of each dart surviving up to level i. Note that using equation 4, this algorithm allows to retrieve any combinatorial map of the pyramid.

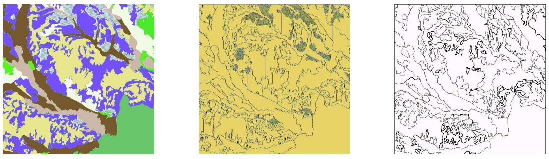
4.3. Embedding

The receptive field of a dart encodes its embedding in the base level combinatorial map. If this initial combinatorial map is associated to a planar sampling grid, each initial vertex corresponds to one pixel and a vertex defined at level i of the pyramid encodes a connected set of vertices, i.e. a region. Within the combinatorial map framework a vertex surviving up to level i is defined by its σ_i orbit. The embedding of a vertex $\sigma_i^*(d) = (d_1, \ldots, d_p)$ in the initial combinatorial map is called a vertex's receptive field and is defined as the concatenation of the dart's receptive fields belonging to its σ_i orbit:

$$R_{\sigma_i^*(d)} = \bigodot_{j=1}^{p} RF_i(d_j)$$

where $\bigodot$ denotes the concatenation operator.

A connected set of pixels is encoded within the combinatorial map framework by a connected set of vertices, each vertex being encoded by its σ orbit (Fig. 1(a)). The set of darts belonging to the σ orbits of one region encodes both



(a) image (b) Backup (c) Borders

Fig. 2. Backup (b) of a combinatorial pyramid encoding the connected components of image (a). The borders of the partition are represented in (c).

the adjacency relationships between adjacent couple of pixels inside the region and between the pixels of the region and their adjacent neighbors outside the region. The former darts are called internal boundaries while the second ones encode the external boundary of the region.

Let us consider for example, the first row of Fig. 1(a). This connected region is defined by the set of darts {1, -1, 2, -2, 7, 8, 9,13,14,15}. The darts {1,-1,2,-2} correspond to internal boundaries while $\{7, 8, 9, 13, 14, 15\}$ correspond to external ones.

Let $\mathcal{B}_i$ be the set of external boundary darts corresponding to the partition defined at level i. Given a dart $d \in \mathcal{SD}_i$, we denote by $\partial RF_i(d) = RF_i(d) \cap \mathcal{B}_i$ the sequence of external boundary darts belonging to the receptive field $RF_i(d)$. The order in $\partial RF_i(d)$ is deduced from the one defined on $RF_i(d)$. Given one dart $d \in \mathcal{SD}_i$, the external boundary of the vertex $\sigma_i^*(d) = (d_1, \ldots, d_p)$ is defined by the concatenation of the sequences $\partial RF_i(d_j)$ with $j \in \{1, \ldots, p\}$: $\partial R_{\sigma_i^*(d)} = \bigodot_{j=1}^{p} \partial RF_i(d)$

The sequence $\partial R_{\sigma_i^*(d)}$ encodes the set of external boundary darts of the connected set of vertices $R_{\sigma_i^*(d)}$.

Given one dart $d \in \mathcal{SD}_i$, the sequence $\partial R_{\sigma_i^*(d)} = d_1, \ldots, d_p$ may be retrieved by [3]:

$$\forall j \in \{1, \ldots, p-1\} \; d_{j+1} = \varphi^{n_j}(\alpha(d_j)) \qquad (6)$$

with $n_j = Min\{k \in \mathbb{N}^* \mid \varphi^k(\alpha(d_j)) \in \mathcal{B}_i\}$

Note that using a 4 connected grid the rank of each φ-orbit is equal to 4. The number of iterations required by equation 6 is thus bounded by 4.

We designed [3] an algorithm which traverses the sequence of external boundary darts of a vertex's receptive field. Each of this dart is associated to one pixel belonging to the border of the region. This sequence provides thus the boundary of the region. The sequential complexity of this algorithm is linear in the total length of the borders of the partition.

Since both darts of an edge are reduced at a same level, we have $level(d) = level(\alpha(d))$ for any dart $d \in \mathcal{D}$. Moreover, using alternatively contraction and removal operations the state of each level may be deduced from its parity. We use the above properties to perform a backup of a pyramid: Given a combinatorial map G encoding a $m \times n$, 4-connected sampling grid, we create a $(m+1) \times (n+1)$ image and associate one pixel to each vertex of the dual combinatorial map $\overline{G}$ (Section 2). Then, we store in each pixel of the image the level of the two darts encoding the adjacency of the associated vertex with its right and down neighbors. Using Fig. 1(b), the top-left pixel of the image encode the level of the darts -13 and 24. The level of the remaining darts is retrieved using the relationship: $level(d) = level(\alpha(d))$. The level of each dart being encoded on 12 bits the backup of the pyramid may be encoded as a 24 bits $(m+1) \times (n+1)$ color image.

Fig. 2(a) represents a satellite image[1] whose connected components are encoded by a combinatorial pyramid. The backup of this pyramid is represented in Fig 2(b). The border of the partition encoded on the top of the pyramid are represented in Fig 2(c).

5. CONCLUSION

A combinatorial pyramid can be folded into its base level combinatorial map by two functions $state()$ and $level()$. Using these functions and the local orientation of edges around each vertex any reduced combinatorial map may be unfolded directly from the base level. Our method stores in the base level combinatorial map the relevance of each edge according to a particular decimation process and allows thus to combine local and global information into a same representation.

6. REFERENCES

[1] Annick Montanvert, Peter Meer, and Azriel Rosenfeld, "Hierarchical image analysis using irregular tessellations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 307–316, APRIL 1991.

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

[3] Luc Brun, *Traitement d'images couleur et pyramides combinatoires*, Habilitation à diriger des recherches, Université de Reims, 2002.

[4] W.T. Tutte, *Graph Theory*, vol. 21, Addison-Wesley, encyclopedia of mathematics and its applications edition, 1984.

[1]Data provided by the institute of Surveying Remote Sensing and Land Information, BOKU Vienna