Hierarchical Image Partitioning using Combinatorial Maps

Adrian Ion[†], Yll Haxhimusa[†], Walter G. Kropatsch[†], and Luc Brun[‡]

[†] [*]Pattern Recognition and Image Processing Group 183/2
Institute for Computer Aided Automation, Vienna University of Technology
Favoritenstr. 9, A-1040 Vienna, Austria

[‡] Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (UMR 6072)
Université de Caen Basse - Normandie
6 Boulevard Maréchal Juin, 14050 Caen, France

[†]{ion,yll,krw}@prip.tuwien.ac.at, [‡]luc.brun@greyc.ensicaen.fr

Abstract

We present a hierarchical partitioning of images using a pairwise similarity function on a combinatorial map based representation. We used the idea of minimal spanning tree to find region borders quickly and effortlessly in a bottom-up way, based on local differences in a color space. The result is a hierarchy of partitions with multiple resolutions suitable for further goal driven analysis. The algorithm can handle large variation and gradient intensity in images. Dual graph pyramid representations lack the explicit encoding of edge orientation around vertices i.e they lack an explicit encoding of the orientation of planes, existing in combinatorial maps. Moreover with combinatorial maps, the dual must not be explicitly represented because one map is enough to fully characterize the partition.

1 Introduction

The authors in [12] asked: ”How do we bridge the representational gap between image features and coarse model features?” They identify the 1-to-1 correspondence between salient image features (pixels, edges, corners,...) and salient model features (generalized cylinders, polyhedrons,...) as a limiting assumption that makes prototypical or generic object recognition impossible. They suggested to bridge and not to eliminate the representational gap, and to focus efforts on *region segmentation*, *perceptual grouping*, and *image abstraction*. The union of regions forming the group is again a region with both internal and external properties and relations. Low-level cue image segmentation cannot and should not produce a complete final *good* segmentation, because there is *an intrinsic ambiguity* in the exact location of region boundaries in digital images. Problems emerge because homogeneity of low-level cues will not map to the semantics [12], and the degree of homogeneity of a region is in general quantified by threshold(s) for a given measure [7]. The low-level coherence of brightness, color, texture or motion attributes should be used to come up sequentially with hierarchical partitions [20].

[*]This paper has been supported by the Austrian Science Fund under grants FSP-S9103-N04.

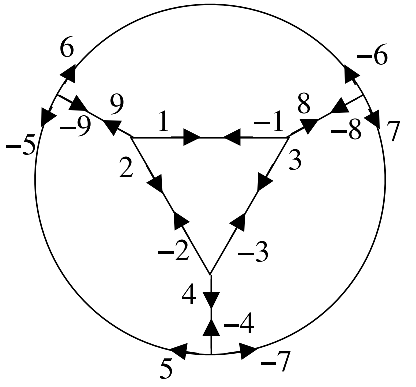
A grouping method should have the following properties [6]: capture perceptually important groupings or regions which reflect global aspects of the image, be highly efficient, running in time linear in the number of image pixels (e.g Minimal spanning tree), and creating hierarchical partitions [20].

In a regular image pyramid the number of pixels at any level l, is r times higher than the number of pixels at the next reduced level $l + 1$. The so called reduction factor r is greater than one and it is the same for all levels l. If s denotes the number of pixels in an image I, the number of new levels on top of I amounts to $log_r(s)$. Thus, the regular image pyramid may be an efficient structure for fast grouping and access to image objects in top-down and bottom-up processes. However, regular image pyramids are confined to globally defined sampling grids and lack shift invariance [1]. Bister et.al. [1] concludes that regular image pyramids have to be rejected as general-purpose segmentation algorithms. In [18, 11] it was shown how these drawbacks can be avoided by irregular image pyramids, the so called adaptive pyramids, where the hierarchical structure (vertical network) of the pyramid was not *a priori* known but recursively built based on the data. Moreover in [5, 17], was shown that irregular pyramids can be used for segmentation and feature detection.

Each level represents a partition of the pixel set into cells, i.e. connected subsets of pixels. The construction of an irregular image pyramid is iteratively local [16, 10]. This means that we use only local properties to build the hierarchy of the pyramid. On the base level (level 0) of an irregular image pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4 (8)-connectivity of the pixels. A cell on level $l + 1$ (parent) is a union of neighboring cells on level l (children). This union is controlled by so called contraction kernels (decimation parameters, see [14]). Every parent computes its values independently of other cells on the same level. This implies that an image pyramid is built in $O[log(image\_diameter)]$ time. Neighborhoods on level $l + 1$, are derived from neighborhoods on level l. Two cells c_1 and c_2 are neighbors if there exist pixels p_1 in c_1 and p_2 in c_2 such that p_1 and p_2 are 4-neighbors. We assume that on each level $l + 1$ $(l \geq 0)$ there exists at least one cell not contained in level l. In particular, there exists a highest level h . In general the top of the pyramid can have one vertex, i.e. an apex.

Region adjacency graphs (RAG), dual graphs [9] and combinatorial maps have been used before [3] to represent the partitioning of 2D space. From these 3 structures, the combinatorial map seems to be the most adequate because, RAGs cannot correctly encode multiple boundaries and inclusions, and dual graphs lack the explicit encoding of edge orientation around vertices, present in a combinatorial map [3]. Moreover with combinatorial maps, the dual must not be explicitly represented because one map is enough to fully characterize the partition, and also the dual can be easily deduced anytime.

In this paper we present the hierarchical image partitioning method introduced in [9] using combinatorial maps (Section 2) and combinatorial pyramids. The building of the minimum weight spanning tree (MST) using Borůvka's algorithm [2], which is an efficient parallel method, and (dual) combinatorial map contraction is presented in Section 3. We will end with some experimental results (Section 4) and conclusions (Section 5).



$$\sigma = (1,2,9)(-1,8,3)(-2,-3,4)$$
$$(-4,-7,5)(6,-9,-5)(-6,7,-8)$$

$$\alpha(d) = -d$$

Figure 1: Example combinatorial map

2 Combinatorial Maps

We will recall here some information about combinatorial maps: definitions and some properties. For a more detailed description of combinatorial maps see [4].

Definition 1 Combinatorial map. *A combinatorial map G is the triplet $G = (\mathcal{D}, \sigma, \alpha)$, where $\mathcal{D}$ is a set called the set of darts and σ, α are two permutations defined on $\mathcal{D}$ such that α is an involution:*

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

If the darts are encoded by positive and negative integers, the permutation α can be implicitly encoded by $\alpha(d) = -d$ (see Figure 1). In the following, we will use alternatively both notations. A combinatorial map may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. Thus all graph definitions used in irregular pyramids [13] such as end vertices, self loops, or degrees may be retrieved easily. 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 $\pi^*(d)$. The cardinal of this orbit will be denoted $|\pi^*(d)|$.

Definition 2 End vertices *Given a dart d, we call the end vertices of the edge $\alpha^*(d) = (d, -d)$ the orbits $\sigma^*(d)$ and $\sigma^*(-d)$.*

Some topological notions can be defined on G:

- $\alpha^*(d)$ is a *self loop* iff $-d \in \sigma^*(d)$
- $\alpha^*(d)$ is a *self-directed loop* iff $\sigma(d) = -d$ or $\sigma((\alpha(d)) = d$
- $\alpha^*(d)$ is a *bridge* iff $\alpha(d) \in \phi^*(d)$.
- $\alpha^*(d)$ is a *pendant edge* iff $\sigma(d) = d$ or $\sigma((\alpha(d)) = d$

Using combinatorial maps each vertex is implicitly defined by its set of darts. Thus a vertex partition of a combinatorial map may be defined by encoding each vertex by one of its dart (see definition 3).

Definition 3 Partition. *Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, $\mathcal{D}_1, \dots, \mathcal{D}_n \subset \mathcal{D}$ is a vertex-partition of G iff:*

1. $\forall i \in \{1, \dots, n\} \quad \mathcal{D}_i \neq \emptyset$. *All $\mathcal{D}_i$ are non-empty.*
2. $\forall d \in \mathcal{D} \quad \exists i \in \{1, \dots, n\}, \quad \exists d' \in \mathcal{D}_i \quad | \quad d \in \sigma^*(d')$. *Each vertex may be retrieved thanks to a dart in one $\mathcal{D}_i$.*
3. $\forall i, k \in \{1, \dots, n\}^2 \quad \sigma^*(\mathcal{D}_i) \cap \sigma^*(\mathcal{D}_k) = \emptyset$. *The set of darts of one vertex is included in only one $\mathcal{D}_i$.*

Note that we do not have $\bigcup_{i=1}^n \mathcal{D}_i = \mathcal{D}$. Condition 2, only requires that each vertex has at least one of its darts in one $\mathcal{D}_i$.

Lemma 1 The Restriction Operator. *Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$ and $\mathcal{D}' \subset \mathcal{D}$ the application :*

$$p_{\mathcal{D},\mathcal{D}'} \left(\begin{array}{lll} \mathcal{D}' & \rightarrow & \mathcal{D} \\ d & \mapsto & \sigma^{n-1}(d) \text{ with } n = Min\{p \in \mathbb{N}^* \mid \sigma^p(d) \in \mathcal{D}'\} \end{array} \right.$$

is an injective function.

Definition 4 Partition into Connected Components. *Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, and a partition $\mathcal{D}_1, \dots, \mathcal{D}_n$. This partition will be called a partition into connected components iff:*

$$\forall i \in \{1, \dots, n\} \left\{ \begin{array}{l} G_i = (\mathcal{D}_i, \sigma \circ p_{\mathcal{D},\mathcal{D}_i}, \alpha) \text{ is connected} \\ \sigma^*(\mathcal{D}_i) = \alpha(\sigma^*(\mathcal{D}_i)) \end{array} \right.$$

The second equality means that there is no edge which connects $\mathcal{D}_i$ to $\mathcal{D}_j$.

Definition 5 Dual Combinatorial Map. *Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, the combinatorial map $\overline{G} = (\mathcal{D}, \varphi, \alpha)$ is called the dual of G. The permutation φ is defined by:*

$$\varphi = \sigma \circ \alpha$$

The orbits of φ encode the faces of G. Note that the function φ is a permutation, since it is the composition of two permutations on the same set.

The connectivity is preserved by the dual transformation [4].

Definition 6 Removal Operation. *Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$ and $d \in \mathcal{D}$. If $\alpha^*(d)$ is not a bridge, the combinatorial map $G' = G \setminus \alpha^*(d)$ is the sub-map defined by:*

- $\boldsymbol{\mathcal{D}}' = \boldsymbol{\mathcal{D}} - \alpha^*(d)$ *and*
- $\sigma' = \sigma \circ p_{\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{D}}'}$.

This operation will be denoted $\boldsymbol{R_d}$.

Definition 7 Contraction operation. *Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and one dart d, in $\boldsymbol{\mathcal{D}}$ which is not a self loop. The contraction of dart d creates the graph:*

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

This operation will be denoted $\boldsymbol{C_d}$. Note that this operation is well defined since d is a self-loop in G iff it is a bridge in $\overline{G}$.

Definition 8 Contraction Kernel. *Given a connected combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, the forest [4] $F = (\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_n)$ will be called a contraction kernel N iff:*

$$\mathcal{SD} = \boldsymbol{\mathcal{D}} - \bigcup_{i=1}^{n} \boldsymbol{\mathcal{D}}_i \neq \emptyset$$

The set $\mathcal{SD}$ is called the set of surviving darts.

Definition 9 Equivalent Contraction Kernel. *Given a combinatorial map $G_0 = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, a contraction kernel N_1 of G_0, the contracted combinatorial map $G_1 = G_0/N_1$, and N_2 a contraction kernel of G_1, the contraction kernel K_3 of G_0 for witch $G_0/N_3 = (G_0/N_1)/K_2$, is called the* **equivalent contraction kernel** *of N_1 and N_2.*

The successive application of N_1 and N_2 forms a new operator on G_0 denoted by $N_2 \circ N_1$.

A **combinatorial pyramid** is a stack of successively reduced combinatorial maps, where each map is build from the one below by selecting a set of vertices named *surviving vertices* and mapping each non surviving vertex to a surviving one.

3 Image Partitioning

Let $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ be a given attributed combinatorial map with the vertex set $V = \sigma^*(\boldsymbol{\mathcal{D}})$ and edge set $E = \alpha^*(\boldsymbol{\mathcal{D}})$. We will discuss later about the attributes. The goal is to find partitions $P = \{CC_1, CC_2, ..., CC_n\}$ such that these elements satisfy certain properties. The author in [6] defines the function, $Comp(\cdot, \cdot)$, which measures the difference along the boundary of two components relative to a measure of the differences of components' internal differences. This definition tries to encapsulate the intuitive notion of contrast: a contrasted zone is a region containing two connected components whose inner differences (**internal contrast**) are less than differences within it's context (**external contrast**). We define an external contrast measure between two components and an internal contrast measure of each component. These measures are defined in [6, 8], analogously.

Let $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha, attr_v, attr_e)$ be a given attributed combinatorial map with the vertex set $V = \sigma^*(\boldsymbol{\mathcal{D}})$ and edge set $E = \alpha^*(\boldsymbol{\mathcal{D}})$ on the base level (level 0). Vertices $v \in V$ and edges

$e \in E$ are attributed, i.e. $attr_v : V \to \mathbb{R}^+$ and $attr_e : E \to \mathbb{R}^+$. One possible way to attribute the edges is given in Section 4. The combinatorial map on level k of the pyramid is denoted by $G_k(\boldsymbol{\mathcal{D}}_k, \sigma_k, \alpha_k)$. Every vertex $u \in V_k = \sigma_k^*(\boldsymbol{\mathcal{D}}_k)$ is a representative of a component CC_i of the partition P_k. The equivalent contraction kernel of a vertex $u \in V_k$, $N_{0,k}(u)$ is e set of darts (a subtree) of the base level $d \in \boldsymbol{\mathcal{D}}$ that are contracted; i.e. applying equivalent contraction kernel on the base level, one contracts the sub combinatorial map $G' \subseteq G$ onto the vertex u.

The **internal contrast** measure of the $CC_i \in P_k$ is the **largest dissimilarity** measure of the component CC_i i.e. the largest edge weight of the $N_{0,k}(u)$ of a vertex $u \in V_k$:

$$Int(CC_i) = max\{attr_e(e), e \in \alpha^*(N_{0,k}(u))\}. \tag{1}$$

Let $u_i, u_j \in V_k$ be the end vertices of an edge $e \in E_k$. The **external contrast** measure between two components $CC_i, CC_j \in P_k$ is the **smallest dissimilarity** measure between component CC_i and CC_j i.e. the smallest edge weight connecting $N_{0,k}(u_i)$ and $N_{0,k}(u_j)$ of vertices $u_i \in CC_i$ and $u_j \in CC_j$:

$$Ext(CC_i, CC_j) = min\{attr_e(e), e = (d, -d) : d \in N_{0,k}(u_i) \wedge -d \in N_{0,k}(u_j)\}. \tag{2}$$

The pairwise comparison function $Comp(\cdot, \cdot)$ between two connected components CC_i and CC_j can now be defined as:

$$Comp(CC_i, CC_j) = \begin{cases} \text{True} & \text{if } Ext(CC_i, CC_j) > PInt(CC_i, CC_j), \\ \text{False} & \text{otherwise,} \end{cases} \tag{3}$$

where $PInt(CC_i, CC_j)$ is the minimum internal contrast difference between two components:

$$PInt(CC_i, CC_j) = min(Int(CC_i) + \tau(CC_i), Int(CC_j) + \tau(CC_j)). \tag{4}$$

For the function $Comp(CC_i, CC_j)$ to be true i.e. for the border to exist, the external contrast difference must be greater than the internal contrast differences. The reason for using a threshold function $\tau(CC)$ in Equation (4) is that for small components CC, $Int(CC)$ is not a good estimate of the local characteristics of the data, in extreme case when $|CC| = 1$, $Int(CC) = 0$. Any non-negative function of a single component CC, can be used for $\tau(CC)$ [6]. One can define τ to be function of the size of CC: $\tau(CC) = \alpha/|CC|$, where $|CC|$ denotes the size of the component CC and α is a constant. More complex definition of $\tau(CC)$, which is large for certain shapes and small otherwise would produce a partitioning which prefers certain shapes, e.g. using ratio of perimeter to area would prefer components that are not long and thin.

Let $P_k = CC_i^k, CC_j^k, ..., CC_n^k$ be the partitions on the level k of the pyramid i.e P_k is the combinatorial map $G_k(\boldsymbol{\mathcal{D}}_k, \sigma_k, \alpha_k)$. Algorithm 1 shows how to build the hierarchy of partitions.

4 Experiments on Image Maps

In the following paragraphs we will comment about some implementation issues and results. Because having a combinatorial map, one can always deduce/obtain it's dual. There is no need of storing both. All the details here (as in the whole paper) are given for the map in which vertices represent regions. In it's dual, instead of vertices, faces are used to represent regions,

Algorithm 1 – Construct Hierarchy of Partitions

Input: Attributed combinatorial map G_0.

```
$k = 0$
**repeat**
    **for all** vertices $u \in V_k = \sigma_k^*(\boldsymbol{\mathcal{D}}_k)$ **do**
        $E_{min}(u) = argmin\{attr_e(e) \mid e = (d, -d) \in E_k$ and $u = \sigma_k^*(d)\}$
    **end for**
    **for all** $e = (d, -d) \in E_{min}, u_{k,i} = \sigma_k^*(d), u_{k,j} = \sigma_k^*(-d)$ with $Ext(CC_i^k, CC_j^k) \leq PInt(CC_i^k, CC_j^k)$ **do**
        include $d$ and $-d$ in contraction kernel $N_{k,k+1}$
    **end for**
    contract combinatorial map $G_k$ with contraction kernel, $N_{k,k+1}$: $G_{k+1} = C[G_k, N_{k,k+1}]$.
    **for all** $e_{k+1} \in E_{k+1} = \alpha_{k+1}^*(\boldsymbol{\mathcal{D}}_{k+1})$ **do**
        set edge attributes $attr_e(e_{k+1}) = min\{attr_e(e_k) \mid e_{k+1} = C[e_k, N_{k,k+1}]\}$
    **end for**
    $k = k + 1$
**until** $G_k = G_{k-1}$
```

Output: A region adjacency combinatorial map at each level of the pyramid.

and also instead of 'end vertices of an edge' we are interested in the neighboring faces of an edge (the 2 faces separated by that edge).

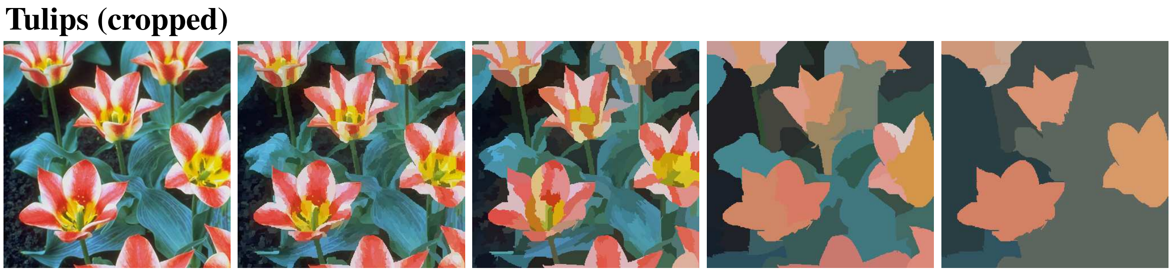
Choosing which map to represent is more a subjective matter because both store the same information in a not so different way (the same framework could be used to store and manage both). Because the base entity in a combinatorial map is the dart, we cannot have a map containing only one vertex and no edges. So one notable difference would be that if vertices are chosen to represent the regions, a one region map, without self loops is not possible until the background (the infinite region) is represented explicitly (darts have to be specially added for that). On the dual, where regions are represented by faces, the background/infinite region also exists, but no special darts have to be added to accommodate it, so a one region map would be made out of 2 darts that are 'sewed' together by α and by σ.

We start with the trivial partition, where each pixel is a homogeneous region. The attributes of edges are defined as the difference of its end point vertices. The attributes of edges can be defined as the difference between end point features of end vertices , $attr_e(u_i, u_j) = |F(u_i) - F(u_j)|$, where F is some feature. F could be defined as $F(u_i) = I(u_i)$, for gray value intensity images, or $F(u_i) = [v_i, v_i \cdot s_i \cdot \sin(h_i), v_i \cdot s_i \cdot \cos(h_i)]$, for color images in HSV color distance [20]. However the choice of the definition of the weights and the features to be used is in general a hard problem, since the grouping cues could conflict each other [15].

For our experiments we used as attributes of edges the euclidean distance between pixel RGB values,

$$attr_e(u_i, u_j) = \sqrt{red(u_i) * red(u_j) + green(u_i) * green(u_j) + blue(u_i) * blue(u_j)}. \quad (5)$$

We choose this simple color distance to study the properties of the algorithm. To compute the



(a) 0 (160 000) (b) 21 (2387) (c) 27 (321) (d) 37 (41) (e) 42 (9)

Figure 2: Some levels of the partitioning of "Tulips(cropped)": level (number of components).

hierarchy of partitions we also need to define $\tau(CC) = \alpha/|CC|$, where $\alpha = const$ and $|CC|$ is the number of elements in CC, i.e. the size of the region. The algorithm has one running parameter α, which is used to compute the function τ. A larger constant α sets the preference for larger components. A more complex definition of $\tau(CC)$, which is large for certain shapes and small otherwise would produce a partitioning which prefers certain shapes, e.g. using ratio of perimeter to area would prefer components that are compact, e.g. not long and thin. For computational efficiency the internal contrast $PInt()$ and the size of the connected component $|CC|$ (receptive field) is stored.

Image *Tulips (cropped)* is a cropped version (top-left 400 by 400 pixels) of the *Tulips* image from the Waterloo image database and *Obj18_355*, and *Obj59_0* from the Coil 100 image database. We found that $\alpha = 300$ produces the best hierarchy of partitions of the images shown in *Tulips.cropped* [1] Figure 2, *Obj18_355* [2] Figure 3, and *Obj59_0* [2] Figure 4. Figures 2, 3 and 4 show some of the partitions on different levels of the pyramid and the number of components. In general the top of the pyramid will consist of one vertex, an apex, which represents the whole image.

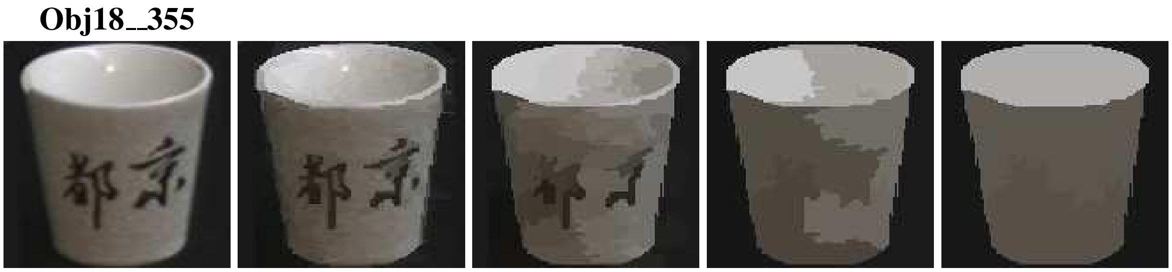
Note that in all images there are regions of large intensity variability and gradient. This algorithm copes with this kind of gradient and variability. In contrast to [6][3] the result is a hierarchy of partitions with multiple resolutions, suitable for further goal driven, domain specific analysis [4]. On the lower level of the pyramid the image is over segmented (partitioned) whereas in upper it is under segmented (partitioned), the help of mid and high level knowledge would select the proper partitioning. Since the algorithm preserves details in low-variability regions, a noisy pixel would survive through the hierarchy. Of course, image smoothing in low variability regions would overcome this problem. We, however do not smooth the images, as this would introduce another parameter into the method. The hierarchy of partitions can also be built from an over segmented image to overcome the problem of noisy pixels. Note that the influence of τ in decision criterion is smaller as the region gets bigger for a constant α. The constant α is

[1] Waterloo image database

[2] Coil 100 image database

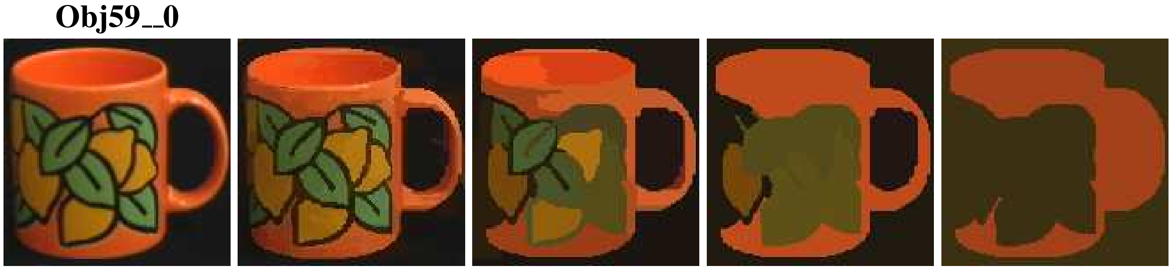
[3] In [19] results of different segmentation methods, including the ones in [6] and [15], are shown and compared.

[4] Please note that a whole class of partitions is created, where a partition is not limited to a certain level of the pyramid, but can be constructed of components from different levels (the receptive fields of the vertices of a multilevel partition occupy the whole image, and do not overlap)



(a) 0 (16 384) (b) 23 (512) (c) 32 (65) (d) 39 (10) (e) 42 (4)

Figure 3: Some levels of the partitioning of "Obj18__355": level (number of components).



(a) 0 (16 384) (b) 24 (291) (c) 34 (23) (d) 37 (6) (e) 39 (2)

Figure 4: Some levels of the partitioning of "Obj59__0": level (number of components).

used to produce a kind of the over segmented image and the influence of τ is smaller after each level of the pyramid. For an overexerted image, where the size of regions is large, the algorithm becomes parameterless.

5 Conclusion

In this paper we presented a method for building hierarchical image partitions using Borůvka's minimal spanning tree algorithm. The hierarchy is presented as a combinatorial pyramid, where each level is a 2D combinatorial map. Combinatorial maps are defined in any dimension, thus the current work should lead the way to segmentation of digital video streams using contraction in 3D combinatorial maps/pyramids. It was shown that the algorithm can handle large variation and gradient intensity in images. Even though the algorithm makes greedy decisions locally, it produces perceptually important partitions in a bottom-up way based only on local differences. A drawback is that maximum and minimum criterion is very sensitive to noise, although in practice it has a small impact. To overcome the problem of noise, one could start with an oversegmented image produced by a robust method e.g. watershed method. A comparison between the classes of partitions produced by the presented method, and some of the well known methods (e.g. [6]) is planned.

References

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

[2] O. Borůvka. O jistém problému minimálnim. *Práce Mor. Přírodvěd. Spol. v Brně (Acta Societ. Scienc. Natur. Moravicae)*, 3(3):37–58, 1926.

[3] L. Brun and W. Kropatsch. Contraction kernels and combinatorial maps. *Pattern Recogn. Lett.*, 24(8):1051–1057, 2003.

[4] L. Brun and W. G. Kropatsch. Dual Contraction of Combinatorial Maps. Technical Report PRIP-TR-54, Institute f. Computer Aided Automation 183/2, PRIP, TU Wien, Austria, 1999. http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr54.ps.gz.

[5] K. Cho and P. Meer. Image Segmentation from Consensus Information. *CVGIP: Image Understanding*, 68 (1):72–89, 1997.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Image Segmentation Using Local Variation. In *Proc. of IEEE Conference on CVPR*, 98–104, 1998.

[7] C.-S. Fu, W. Cho, S, and K. Essig. Hierarchical color image region segmentation for content-based image retrival system. *IEEE Trans. on Img. Proc.*, 9(1):156–162, 2001.

[8] L. Guigues, L. M. Herve, and J.-P. Cocquerez. The Hierarchy of the Cocoons of a Graph and its Application to Image Segmentation. *Patt. Rec. Lett.*, 24(8):1059–1066, 2003.

[9] Y. Haxhimusa and W. G. Kropatsch. Segmentation Graph Hierarchies. In A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, and D. de Ridder, eds, *Procc of J. Inter. Work. on SS+SPR 2004*, LNCS 3138:343–351, 2004.

[10] J.-M. Jolion. Stochastic pyramid revisited. *Patt. Rec. Lett.*, 24(8):1035–1042, 2003.

[11] J.-M. Jolion and A. Montanvert. The adaptive pyramid, a framework for 2D image analysis. *Comp. Vis., Graphics, and Im. Proces.: Im. Underst.*, 55(3):339–348, 1992.

[12] Y. Keselman and S. Dickinson. Generic model abstraction from examples. In *Proc. of CVPR*, 1:856–863, 2001.

[13] W. G. Kropatsch. Parallel graph contraction for dual irregular pyramids. Tech. Rep. PRIP-TR-28, Institute f. Computer Aided Automation 183/2, PRIP, TU Wien, Austria, Janury 1990. http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr28.ps.gz.

[14] W. G. Kropatsch, A. Leonardis, and H. Bischof. Hierarchical, Adaptive and Robust Methods for Image Understanding. *Surveys on Mathem. for Ind.*, 9:1–47, 1999.

[15] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue integration in image segmentation. In *Intern. Conf. on Comp. Vis.*, 918–925, 1990.

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

[17] P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld. Consensus vision. In *AAAI-90 Workshop on Qualitative Vision*, 111–115, 1990.

[18] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tesselations. *IEEE Trans. on PAMI*, 13(4):307–316, 1991.

[19] M. Pavan and M. Pelillo. Dominiant Sets and Hierarchical Clustering. *ICCV*, 2003.

[20] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *CVPR*, 731–737, 1997.