In Proceedings of the 16^{th} IT&T SPIE Annual Symposium, *Computational Imaging*, January, 2004 ©.

Grouping and Segmentation in a Hierarchy of Graphs

Walter G. Kropatsch and Yll Haxhimusa

Pattern Recognition and Image Processing Group 183/2,
Institute for Computer Aided Automation, Vienna University of Technology, Vienna, Austria

ABSTRACT

We review multilevel hierarchies under the special aspect of their potential for segmentation and grouping. The one-to-one correspondence between salient image features and salient model features are a limiting assumption that makes prototypical or generic object recognition impossible. The region's internal properties (color, texture, shape, ...) help to identify them and their external relations (adjacency, inclusion, similarity of properties) are used to build groups of regions having a particular consistent meaning in a more abstract context. Low-level cue image segmentation in a bottom-up way, cannot and should not produce a complete final "good" segmentation. We present a hierarchical partitioning of images using a pairwise similarity function on a graph-based representation of an image. This function measures the difference along the boundary of two components relative to a measure of differences of the components' internal differences. Two components are merged if there is a low-cost connection between them. We use this idea to find region borders quickly and effortlessly in a bottom-up way, based on local differences in a specific feature. The aim of this paper is to build a minimum weight spanning tree (MST) in order to find region borders quickly in a bottom-up 'stimulus-driven' way based on local differences in a specific feature.

Keywords: Hierarchical graph-based image partitioning, irregular graph pyramids, topology preserving contraction.

1. INTRODUCTION

Kesselman and Dickinson[1] asked the following question referring to several research issues: "How do we bridge the representational gap between image features and coarse model features?" They identify the one-to-one correspondence between salient image features (pixels, edges, corners,...) and salient model features (generalized cylinders, polyhedrons, invariant models,...) as **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**. Let us take these goals as a guideline to consider multiresolution representations under the special viewpoint of segmentation and grouping. Kropatsch[2] considers the multiresolution representation under the abstraction viewpoint.

Wertheimer[3] has formulated the importance of wholes and not of its individual elements as: "There are wholes, the behaviour of which is not determined by that of their individual elements, but where the part-processes are themselves determined by the intrisinic nature of the whole",[4] and introduced the importance of perceptual grouping and organization in visual perception. Regions as aggregations of primitive pixels play an extremely important role in nearly every image analysis task. Their internal properties (color, texture, shape, ...) help to identify them and their external relations (adjacency, inclusion, similarity of properties) are used to build groups of regions having a particular meaning in a more abstract context. 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 it is not clear what a "good" segmentation is. Without prior knowledge, segmentation based on low-level cues will not be able to extract semantics in generic images. The segmentation process results in "homogeneity" regions w.r.t the low-level cues using some similarity measures. Problems emerge because (i) homogeneity of

Further author information: (Send correspondence to Walter G. Kropatsch)
Walter G. Kropatsch : E-mail: krw@prip.tuwien.ac.at, Telephone: +43 1 58801 183 50
Yll Haxhimusa: E-mail: yll@prip.tuwien.ac.at, Telephone: +43 1 58801 183 70

low-level cues will not map to the semantics[1] and (ii) the degree of homogeneity of a region is in general quantified by threshold(s) for a given measure.[5] The low-level coherence of brightness, color, texture or motion attributes should be used to come up sequentially with hierarchical partitions.[6] Mid and high level knowledge can be used to either confirm these groups or select some further attention. A wide range of computational vision problems could make use of segmented images, were such segmentation rely on efficient computation e.g. motion estimation requires an appropriate region of support for finding correspondence. Higher-level problems such as recognition and image indexing can also make use of segmentation results in the problem of matching. It is important that a grouping method has following properties[7]: **i)** capture perceptually important groupings or regions, which reflect global aspects of the image, **ii)** be highly efficient, running in time linear in the number of image pixels, and **iii)** creates hierarchical partitions.[6]

The aim of this paper is to build a minimum weight spanning tree (MST) in order to find region borders quickly. The aim is reached by using the selection method for contraction kernels, proposed in Haxhimusa[8] to achieve logarithmic tapering, local construction and shift invariance. Borůvka's algorithm[9] with dual graph contraction algorithm[10] builds in a hierarchical way a MST (of the region) preserving the proper topology. The topological relation seems to play an even more important role for vision tasks in natural systems than precise geometrical position. Rather than trying to have just one "good" segmentation the method produces a stack of (dual) graphs (a graph pyramid), which down-projected on the base level will give a multi-level segmentation.

After presenting related work and the pyramid representation, we recall the main idea of the Borůvka's MST algorithm in Sec. 4. In Sec. 5 we give the definition of internal and external contrast and the merging decision criteria. Building the hierarchy of image is given in Sec. 5. Sec. 6 reports results.

1.1. Related Work

A graph-theoretical clustering algorithm consists in searching for a certain combinatorial structure in the edge weighted graph, such as a minimum spanning tree,[7, 11] a minimum cut[6, 12] and, among these methods a classic approach to clustering (the *complete linkage clustering* algorithm[13]) reduces to a search for a complete subgraph i.e. the maximal clique.[14]

Hierarchical structures describing data for clustering purposes[13] or image segmentation[15] have been studied very early. Horowitz and Pavlidis[15] define a consistent homogeneity criterion over a set as a boolean predicate over its parts that verifies the consistency property. In image analysis this states that the subregions of a homogeneous region are also homogeneous. Thus the joint use of hierarchy and homogeneity criteria allow to define a partitioning in a natural way. A regular image pyramid may be an efficient structure for fast grouping and access to image objects in top-down and bottom-up processes.[16] However, the regular image pyramids are confined to globally defined sampling grids and lack shift invariance. Bister[17] concludes that regular image pyramids have to be rejected as general-purpose segmentation algorithms. These drawbacks can be avoided by irregular image pyramids,[18, 19] the so called adaptive pyramids, where the hierarchical structure (vertical network) of the pyramid is not known *a priori* but recursively built on the data. Meer[20] in his *consensus vision* uses n irregular pyramids to produce an image segmentation.

The clustering community has produced agglomerative and divisive algorithms[21]; in image segmentation the region-based merging and splitting algorithms exist. Early graph-based methods[22] use fixed thresholds and local measures in computing a segmentation, i.e MST is computed. The segmentation criterion is to break the MST edges with the largest weight. The idea behind is that edges in the MST reflect the low-cost connection between two elements. The work of Urquhart[23] attempts to overcome the problem of fixed threshold by normalizing the weight of an edge using the smallest weight incident on the vertices touching that edge. The methods of Felzenszwalb[7] and Guigues[11] use an adaptive criterion that depend on local properties rather than global ones. Recent works have the MST as the base algorithm.[7, 11, 24, 25]

Gestalt grouping factors, such as proximity, similarity, continuity and symmetry, are encoded and combined in pairwise feature similarity measures.[6, 12, 26, 27] Another method of segmentation is that of splitting and merging region based on how well the regions fulfill some criterion. These methods[28, 29] use a measure of uniformity of a region. Felzenszwalb[7] and Guigues[11] use, in contrast, in a graph-based method a pairwise region comparison rather than applying a uniformity criterion to each individual region. It has been demonstrated that complex grouping phenomena can emerge from simple computation on these local cues.[30, 31]

Our method is related to the work of Felzenszwalb[7] and Guigues[11] in the sense of pairwise comparison of region similarity. We create a hierarchy of graphs i.e of partitions. At each level of the pyramid a region adjacency graph (RAG) is created, in an agglomerative way by edge contraction taking the proper topology into account. A vertex of RAG is a representative of a region on the base level (it's receptive field), creation of which takes into consideration the integrity of the region.

2. VISUAL ABSTRACTION

By definition abstraction extracts essential features and properties while it neglects unnecessary details. Two types of unnecessary details can be distinguished: **redundancies** and **data of minor importance**. Details may not be necessary in different contexts and under different objectives which reflect in different types of abstraction.

In general, three different types of abstraction are distinguished: **isolating abstraction**: important aspects of one or more objects are extracted from their original context; **generalizing abstraction**: typical properties of a collection of objects are emphasized and summarized; **idealizing abstraction**: data are classified into a (finite) set of ideal models, with parameters approximating the data and with (symbolic) names/notions determining their semantic meaning.

These three types of abstraction have strong associations with well known tasks in computer vision: recognition and object detection tries to **isolate** the object from the background; perceptual grouping needs a high degree of **generalization**; and classification assigns data to **'ideal'** classes disregarding noise and measurement inaccuracies. In all three cases abstraction drops certain data items which are considered less relevant. Hence the **importance** of the data needs to be computed to decide which items to drop during abstraction. The importance or the relevance of an entity of a (discrete) description must be evaluated with respect to the purpose or the goal of processing. The system may also change its focus according to changing goals after knowing certain facts about the actual environment, other aspects that were not relevant at the first glance may gain importance. Representational schemes must be flexible enough to accommodate such attentional shifts in the objectives. For an overview see Kropatsch.[2]

3. PYRAMIDS

In this section we summarize the concepts developed for building and using multiresolution pyramid[16,32] and put the existing approaches into a general framework. The focus of the presentation is the representational framework, its components and the processes that transfer data within the framework.

A pyramid (Fig. 1a,b) describes the contents of an image at multiple levels of resolution. The base level is a high resolution input image. Successive levels reduce the size of the data by a constant **reduction factor** $\lambda > 1.0$ while local **reduction windows** relate one cell at the reduced level with a set of cells in the level directly below. Thus local independent (and parallel) processes propagate information up and down in the pyramid. The contents of a lower resolution cell is computed by means of a **reduction function** the input of which are the descriptions of the cells in the reduction window. Sometimes the description of the lower resolution needs to be extrapolated to the higher resolution. This function is called the **refinement** or **expansion function**. It is used in Laplacian pyramids[33] and wavelets[34] to identify redundant information in the higher resolution and to reconstruct the original data. The number of levels n is limited by the reduction factor λ: $n \leq \log(image\_size)/\log(\lambda)$. The main computational advantage of image pyramids is due to this logarithmic complexity. The reduction window and the reduction factor relate two successive levels of a pyramid. In order to interpret a derived description at a higher level this description should be related to the original input data in the base of the pyramid. This can be done by means of the **receptive field** (RF) of a given pyramidal cell c_i: $RF(c_i)$ collects all cells (pixels) in the base level of which c_i is the ancestor . This is the base of several pyramidal approaches, one of which is chosen as representative: irregular graph pyramids.

3.1. Irregular Graph Pyramid Representation

In irregular pyramids, each level represents an arbitrary partition of the pixel set into cells, i.e. connected subsets of pixels. The construction of an irregular pyramid is iteratively local[8,36]: **i)** the cells have no information about their global position, **ii)** the cells are connected only to (direct) neighbors, and **iii)** the cells cannot distinguish

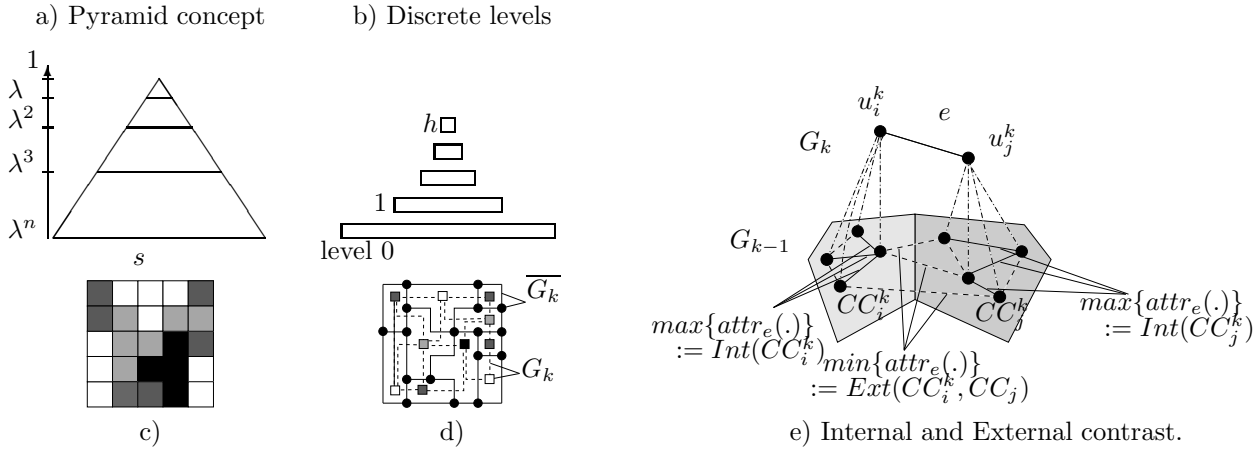

Figure 1. (a,b) Multiresolution pyramid. (c) Partition of pixel set into cells. (d) Representation of the cells and their neighborhood relations by a dual pair $(G_k, \overline{G_k})$.

the spatial positions of the neighbors. 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 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[10]). Every parent computes its values independently of other cells on the same level. This implies that an image pyramid is built in $O[log(image\_diameter)]$ parallel steps. Neighborhoods on level $k+1$ are derived from neighborhoods on level k. 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 neighbors (Fig. 1c). In the base level (level 0) pixels are the vertices and two vertices are related by an edge if the two corresponding pixels are neighbors. On each level $k+1$ ($k \geq 0$) there exists at least one cell not contained in level k. In particular, there exists a highest level h . In general the top of the pyramid can have one vertex, i.e. an apex.

A graph pyramid is a pyramid where each level is a graph $G(V, E)$ consisting of vertices V and of edges E relating two vertices. In order to correctly represent the embedding of the graph in the image plane[35] we additionally store the dual graph $\overline{G}(\overline{V}, \overline{E})$ at each level. We represent the levels as **dual pairs** $(G_k, \overline{G_k})$ of plane graphs G_k and $\overline{G_k}$ (Fig. 1d). The vertices of G_k represent the cells and the edges of G_k represent the neighborhood relations of the cells on level k, depicted with square vertices and dashed edges in Figure 1d. This graph is also called the **region adjacency graph**. The edges of $\overline{G_k}$ represent the borders of the cells on level k, depicted with solid lines in Figure 1d, possibly including so called pseudo edges needed to represent the neighborhood relation to a cell completely surrounded by another cell. Finally, the vertices of $\overline{G_k}$, the circles in Figure 1d, represent meeting points of at least three edges from G_k, solid lines in Figure 1d. Let us denote the original graph as the **primal graph**. The sequence $(G_k, \overline{G_k})$, $0 \leq k \leq h$ is called (dual) **graph pyramid** (Figure 1b). Moreover the graph is attributed, $G(V, E, attr_v, attr_e)$, where $attr_v : V \rightarrow \mathbb{R}^+$ and $attr_e : E \rightarrow \mathbb{R}^+$, i.e. content of the graph is stored in attributes attached to both vertices and edges. Initially only the attributes of the vertices receive the gray values of the pixels. We use a weight for $attr_e$ measuring the difference between the two end points.

In general a graph pyramid can be generated bottom-up as in Alg. 1. The complete formalism of **dual graph contraction** is described by Kropatsch etal.[32] Let us explain it here by means of our image example (Fig. 1c). The second step determines what information in the current top level is important and what can be dropped. A contraction kernel is a (small) sub-tree of the top level the root of which is chosen to survive (rooted trees in Fig. 2b). Fig. 2a shows the window and Fig. 2b the selected contraction kernels $N_{0,1}$. Selection criteria in this case contract only edges inside connected components having the same gray value.

All the edges of the contraction trees are dually contracted during step 3. Dual contraction of an edge e (formally denoted by $G/\{e\}$) consists of contracting e and removing the corresponding dual edge $\overline{e}$ from the

Algorithm 1 – Contructing Graph Pyramid

Input: Graphs $(G_0, \overline{G_0})$

1: **while** further abstraction is possible **do**
2: determine contraction kernels, $N_{k,k+1}$.
3: perform dual graph contraction and simplification of dual graph, $(G_{k+1}, \overline{G_{k+1}}) = C[(G_k, \overline{G_k}), N_{k,k+1}]$.
4: apply reduction functions to compute content $attr : G_{k+1} \to \mathbb{R}^+$ of new reduced level.
5: **end while**

Output: Graph pyramid – $(G_k, \overline{G_k})$, $0 \leq k \leq h$.

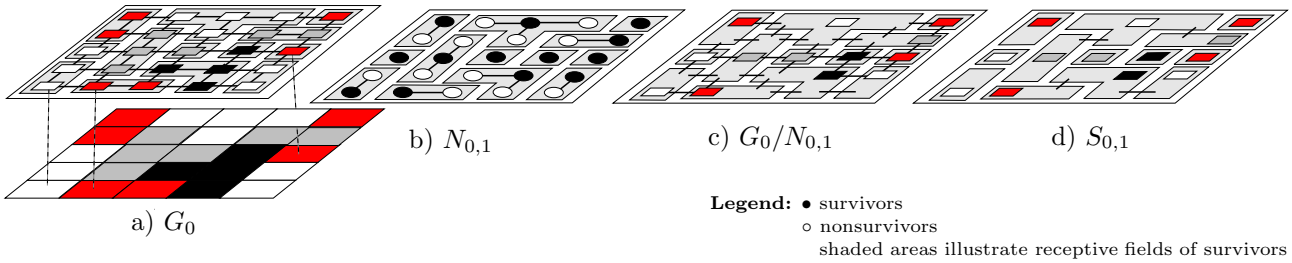

Figure 2. a) Neighborhood graph G_0, b) contraction kernel $N_{0,1}$, c) edge contractin $G_0/N_{0,1}$ and d) redundant edges $S_{0,1}$.

dual graph (formally denoted by $\overline{G} \setminus \{\overline{e}\}$). In our example the graph shown in Fig. 2c is created. This preserves duality and the dual graph needs not be constructed from the contracted primal graph G' at the next level. Since the contraction of an edge may yield multi-edges and self-loops there is a second phase of step 3 which removes all redundant multi-edges and self-loops (edges $S_{0,1}$ in Fig. 2d). Note that not all such edges can be removed without destroying the topology of the graph: if the cycle formed by the multi-edge or the self-loop surrounds another part of the data its removal would corrupt the connectivity! Fortunately this can be decided locally by the dual graph since **faces of degree two** (having the double-edge as boundary) and **faces of degree one** (boundary = self-loop) cannot contain any further elements in its interior. Since removal and contraction are dual operations, the removal of a self-loop or a double edge can be done by contracting the corresponding dual edges in the dual graph, wich are not depicted in our example for the simplicity of figures. The dual contraction of our example graph G_0 remains a simple graph G_1 without self-loops and multi-edges (Fig. 3a).

Step 3 generates a reduced pair of dual graphs. Their contents is derived in step 4 from the level below using the reduction function. In our example reduction is very simple: the surviving vertex inherits the color of its son. The result of the dual contraction is shown as graph G_2 in Fig. 4. The selection rules and the reduction function are the same as in the first iteration. The result shows that the regions with the same color are brought together. This fact could be used in a top-down verification step which checks the reliability of merging criterion in a more general context. The result of the algorithm applied for this simply merging criterion is shown in Fig. 4,(the dual graphs are not shown) together with contraction kernels and the equivalent contraction kernel. See Sec. 5.2 for a complex merge criterion. By contracting the edges of the equivalent contraction kernel $N_{0,2}$ one can reach G_2 directly from the base.

There are lots of useful properties of the resulting graph pyramids. If the plane graph is transformed into a

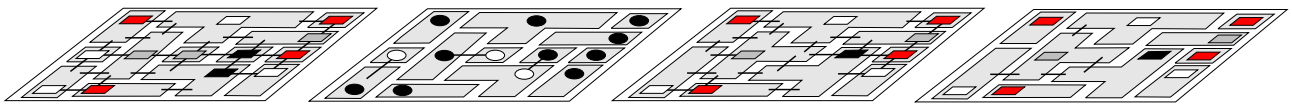

Figure 3. a) Region adjacency graph G_1 after dually contracting G_0, b) contraction kernel $N_{1,2}$, c) edge contraction of graph $G_1/N_{1,2}$ and d) redundant edges $S_{1,2}$.

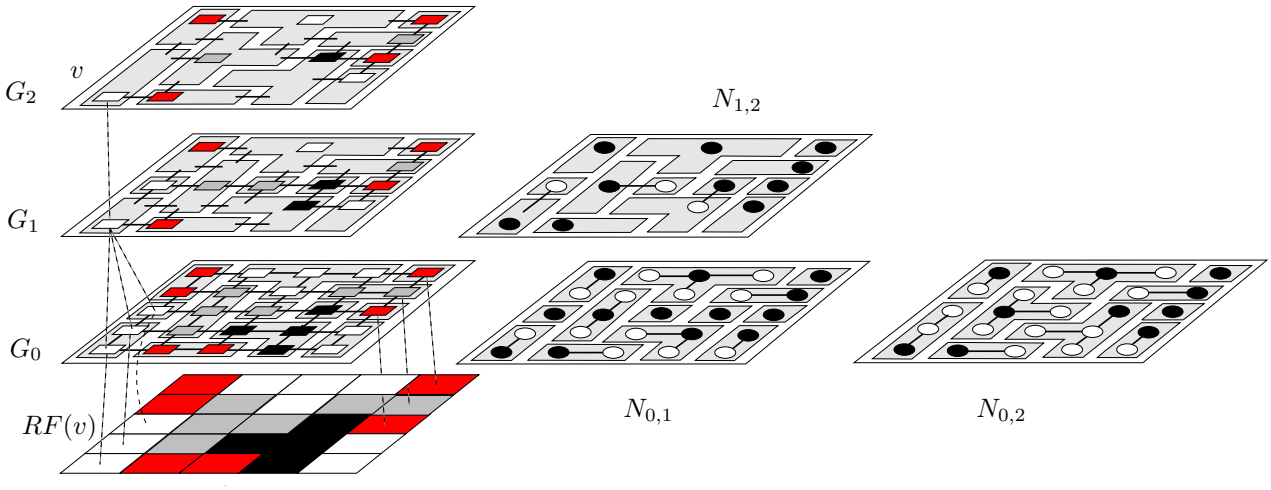

Figure 4. The graph pyramid with contraction kernels $N_{0,1}, N_{1,2}$ and the equivalent contraction kernel $N_{0,2}$. The receptive field $RF(v)$ of a vertex $v \in G_2$ is shown for illustration.

combinatorial map the transcribed operations form the combinatorial pyramid.[37,38] This framework allowed us to prove several of the above mentioned properties and link dual graph pyramids with topological maps which extend the scope to three dimensions. The following table summarizes dual graph contraction in terms of the control parameters used for abstraction and the conditions to preserve topology:

Level	representation	contract / remove	conditions
0	$(G_0, \overline{G_0})$		
	$\downarrow$	contraction kernel $N_{0,1}$	forest, depth 1
	$(G_0/N_{0,1}, \overline{G_0} \setminus \overline{N_{0,1}})$		
	$\downarrow$	redundant edges $S_{0,1}$	$\deg \overline{v} \leq 2$
1	$(G_1 = G_0/N_{0,1} \setminus S_{0,1}, \overline{G_1} = \overline{G_0} \setminus \overline{N_{0,1}}/\overline{S_{0,1}})$		
	$\downarrow$	contraction kernel $N_{1,2}$	forest, depth 1
	$\vdots$		

4. MINIMUM WEIGHT SPANNING TREE: BORŮVKA'S ALGORITHM

Let $G = (V, E)$ be the undirected connected plane graph consisting of the finite set of vertices V and the finite set of edges E. Each edge $e \in E$ is identified with a pair of vertices $v_i, v_j \in V$. Let each edge $e \in E$ be associated with a non-negative unique real weight $w(e) := w(v_i, v_j)$. The problem is formulated as construction of a MST of G. A deterministic solution is proposed by Borůvka,[9] Kruskal and Prim[39] as the widely known greedy algorithms. The weight of the subgraph of G is the sum of edge weights of subgraph, i.e. for $T \subseteq G$, the weight of a the subgraph is $w(T) := \sum_{e \in T} w(e)$.

Theorem 1. Consider a vertex v in a weighted connected graph G. Among all the edges incident on v, let e be one of minimum weight. Then, G has a minimum weight spanning tree that contains e.

Theorem 2. Let T be an acyclic subgraph of a weighted connected graph G such that there exists a minimum weight spanning tree containing T. If G' denotes the graph obtained by contracting all the edges of T, and T'_{min} is a minimum weight spanning tree of G', then $T'_{min} \cup T$ is a minimum weight spanning tree.

These two theorems provide the basis of the MST Algorithm 2, see Thulasiraman[40] for proofs. The idea of the Borůvka[9] is similar to Prim's algorithm but executed simulteonosly for the whole graph. We use Borůvka's algorithm to build MST in parallel.[41] This algorithm constructs a spanning tree as shown in Alg. 2. First

Algorithm 2 – Borůvka's Algorithm

Input: Graph $G(V, E)$.

1: $MST :=$ empty edge list.
2: all vertices $v \in V$ make a list of trivial trees L.
3: **while** there is more than one tree in L **do**
4: each tree $T \in L$ finds the edge e with the minimum weight which connects T to $G \setminus T$ and add edge e to MST.
5: using edge e merge pairs of trees in L.
6: **end while**

Output: Minimum weight spanning tree - MST.

create a list L of trivial trees, each a single vertex $v \in V$. For each tree T of L find the edge e with the **smallest weight**, which connects T to $G \setminus T$. The trees T are then connected to $G \setminus T$ with the edge e. In this way the number of trees in L is reduced, until there is only one, the MST.

Theorem 3. Borůvka's algorithm constructs a minimum weight spanning tree.

The proof that this algorithm builds the minimum weight spanning tree is analogous to the proof of the Kruskal's MST algorithm given in.[40] Algorithm 2 may fail to build MST if the edge weights are not distinct, since the set of selected edges may contain cycles. If there are edges with minimal weight that touch a vertex v, then choose the edge with the smallest random number, under the assumption that random numbers are unique.

In 3^{rd} step of the Algorithm 2, each tree $T \in L$ finds the edge with the minimal weight, and as trees become larger, the process of finding the edge with the minimal weight for each tree T takes longer. A contraction of the edge e, in the 4^{th} step of Algorithm 2 will speed up the process of searching for minimum weight edges in Borůvka's algorithm, since the search for the edge with the minimum weight would be a local search.

5. HIERARCHY OF PARTITIONS

The goal is to find partitions $P_k := \{CC_1^k, CC_2^k, ..., CC_n^k\}$ in k^{th} level of the pyramid such that these elements satisfy certain properties. We compare pairwise neighboring vertices, i.e. partitions to check for similarities.[7, 11] Felzenszwalb[7] defines a pairwise group merge criterion $Comp(CC_i^k, CC_j^k)$ that judges whether there is evidence for a boundary between two partitions $CC_i^k, CC_j^k \in P$. Note that $Comp(CC_i^k, CC_j^k)$ is a boolean merge criterion for pairs of partitions and it is not defined yet. Definition of $Comp(CC_i^k, CC_j^k)$ depends on the application. $Comp(CC_i^k, CC_j^k)$ is true, if there is no evidence for a boundary between CC_i^k and CC_j^k, and false otherwise.

This criterion measures the difference along the boundary of two components relative to a measure of 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, like in Felzenszwalb[7] and Guigues.[11]

5.1. Internal and External Contrast

Let $G(V, E, attr_v, attr_e)$ be a given attributed graph with vertex set V and edge set E 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 6. The graph on level k of the pyramid is denoted by G_k. Every vertex $u^k \in G_k$ is a representative of a component CC_i^k of the partition P_k. The equivalent contraction kernel[32] of a vertex $u^k \in G_k$, $N_{0,k}(u^k) \in E$ is e set of edges of the base level that are contracted; i.e. applying equivalent contraction kernel on the base level, one contracts the subgraph $G' \subseteq G$ onto the vertex u^k .

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

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

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

$$Ext(CC_i^k, CC_j^k) := min\{attr_e(e), e := (v, w) : v \in N_{0,k}(u_i^k) \wedge w \in N_{0,k}(u_j^k)\}. \quad (2)$$

In Fig. 1e a simple example of $Int(CC_i^k)$ and $Ext(CC_i^k, CC_j^k)$ is given. $Int(CC_i^k)$ of the component CC_i^k is the *maximum* of weights of the solid line edges, whereas $Ext(CC_i^k, CC_j^k)$ is the *minimum* of weights of the dashed line edges (bridges) connecting component CC_i^k and CC_j^k on the base level G_0. Vertices u_i^k and u_j^k are representative of the components CC_i^k and CC_j^k. By contracting the edges $N_{0,k}(u_i^k)$ one arrives to the vertex u_i^k, analogously $N_{0,k}(u_j^k)$ for the vertex u_j^k.

The pairwise merge criterion $Comp(\cdot, \cdot)$ between two connected components CC_i^k and CC_j^k can now be defined as:

$$Comp(CC_i^k, CC_j^k) := \begin{cases} \text{True} & \text{if } Ext(CC_i^k, CC_j^k) \leq PInt(CC_i^k, CC_j^k), \\ \text{False} & \text{otherwise,} \end{cases} \quad (3)$$

where $PInt(CC_i^k, CC_j^k)$ is the minimum internal contrast difference between two components $PInt(CC_i^k, CC_j^k) := min(Int(CC_i^k) + \tau(CC_i^k), Int(CC_j^k) + \tau(CC_j^k))$. For the merge criterion $Comp(CC_i^k, CC_j^k)$ to be false i.e. for the border to exist, the external contrast difference must be greater than the internal contrast differences. The threshold function $\tau(CC^k)$ controls the size of the components CC^k, since $Int(CC^k)$ is not a good estimate of the local characteristics of the data, in extreme case when $|CC^k| := 1$, $Int(CC^k) := 0$. Any non-negative function of a single component CC^k, can be used for $\tau(CC^k)$.[7] Note that we have not defined the function $\tau(CC^k)$ yet, this will be done in Sec. 6.

5.2. Construct a Hierarchy of Partitions

A consequence of contracting the edge e, which connects T and $G \setminus T$ in the 4^{th} step of Algorithm 2 will speed up the search for minimum weight edges in Borůvka's algorithm. Since each tree (on level k) after edge contraction will be represented by a single vertex (in the level $k+1$), the edge with the minimum weight would be in a local neighborhood. The dual graph contraction algorithm[10] contracts edges and creates "super" vertices with father-son relations between vertices in subsequent levels (vertical relation) and at the same time preserves the topology, whereas Borůvka's algorithm is used to create son-son relation between vertices in the same level (horizontal relation). Internal and external contrast definition are used to find region borders quickly in a bottom-up based on local differences.

Let $P_k := CC_i^k, CC_j^k, ..., CC_n^k$ be the partitions at level k of the pyramid. Alg. 3 collects smallest weighted edges e (4^{th} step) that could be part of MST, and then checks if the edge weight $attr_e(e)$ is smaller than the internal contrast of both of the components (MST of end vertices of e) (6^{th} step). If these conditions are fulfilled then these two components will be merged (7^{th} step). Two regions will be merged if the internal contrast, which is represented by its MST, is larger than the external contrast, represented by the weight of the edge, $attr_e(e)$. All the edges to be contracted form the contraction kernels $N_{k,k+1}$, which are used to create the graph $G_{k+1} := C[G_k, N_{k,k+1}]$,[32] so that all neighborhood and inclusion relations are preserved. In general $N_{k,k+1}$ is a forest. We update the attributes of those edges $e_{k+1} \in G_{k+1}$ with the minimum attribute of the edges $e_k \in E_k$ that are contracted into e_{k+1} (11^{th} step). This means that we do not recompute the attributes of the edges but simply inherit it. The output of the algorithm is a pyramid where each level represents a set of sub trees of the MST. Each vertex of these RAGs is the representative of a MST of a region in the image. In general the top of the pyramid which represents the whole image. The algorithm is greedy since it collects only the nearest neighbor with the minimal edge weights and merges them if the merge criterion holds (Eq. 3).

If we assume that the steps 6 to 8 of the Alg. 3 are left out, it can be shown, that this algorithm produces a MST (Theorem 4). Each vertex $u_k \in G_k$ represents a connected region CC^k on the base level of the pyramid, and since the presented algorithm is based on Borůvka's algorithm,[9] it builds a $MST(u_k)$ of each region, i.e $N_{0,k}(u_k) = MST(u_k)$.

Algorithm 3 – Construct Hierarchy of Partitions

Input: Attributed graph G_0.

$k := 0$
repeat
  for all vertices $u \in G_k$ **do**
    $E_{min}(u) := argmin\{attr_e(e) \,|\, e := (u,v) \in E_k \text{ or } e := (v,u) \in E_k\}$
  end for
  for all $e := (u_{k,i}, u_{k,j}) \in E_{min}$ with $Ext(CC_i^k, CC_j^k) \leq PInt(CC_i^k, CC_j^k)$ **do**
    include e in contraction edges $N_{k,k+1}$
  end for
  contract graph G_k with contraction kernels, $N_{k,k+1}$: $G_{k+1} = C[G_k, N_{k,k+1}]$.
  for all $e_{k+1} \in G_{k+1}$ **do**
    set edge attributes $attr_e(e_{k+1}) := min\{attr_e(e_k) \,|\, e_{k+1} := C[e_k, N_{k,k+1}]\}$
  end for
  $k := k + 1$
until $G_k := G_{k-1}$

Output: A region adjacency graph (RAG) at each level of the pyramid.

Theorem 4. The **equivalent contraction kernel** of a vertex u_i^k in the level k of the graph pyramid is the **minimum spanning tree** of its receptive field on the base level.

For a graph $G(V, E)$ where $|V| := n$ and $|E| := m$ follows

Theorem 5. Alg. 3 finds the minimum spanning tree after **at most** $\log n$ **iterations** each of which involves $O(m + n)$ operations.

It can be shown that the Alg. 3 produces a **hierarchy** over V and **partitions on each level** which are **invariant** under any monotone transformation of dissimilarity measure $attr_e$ and that the **hierarchy** over V is **invariant** under monotone transformation. See Haxhimusa[42] for more details and for the proofs of the theorems. The presented algorithm collects only the nearest neighbor partitions with the minimal edge weights and merges two trees if they fulfill the criterion in Eq. 3, this is known as *single linkage clustering.*[13]

6. EXPERIMENTS ON IMAGE GRAPHS

We start with the trivial partition, where each pixel is a homogeneous region. The attributes of edges can be defined as the difference between features of end vertices, $attr_e(u_i, u_j) := |F(u_i) - F(u_j)|$, where F is some feature. Other distances could be used as well e.g.,[6] $attr_e(u_i, u_j) := e^{\frac{-||F(u_i)-F(u_j)||_2^2}{\sigma_I}}$, where F is some feature, and σ_I is a parameter which controls the scale of proximity measures of F. 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 space.[6] 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.[31]

For our experiments we use as feature $F(u)$ pixel intensities $I(u)$, for color images $F(u)$ is the luminance in the color space. To compute the hierarchy of partitions we define $\tau(CC^k)$ to be function of the size of CC^k e.g. $\tau(CC^k) := \alpha/|CC^k|$, where $|CC^k|$ denotes the size of the component CC^k and α is a constant. A larger constant α sets the preference for larger components. 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.

We use indoor and outdoor RGB images. We found that $\alpha = 300$ produces the best hierarchy of partitions of the images shown in *Monarch*[*1] Fig. 5, and *Object*45[2] Fig. 6 after the average intensity attribute of vertices is down-projected onto the base grid. Fig. 5 and Fig. 6 show some of the partitions on different levels of pyramid and the number of components. Note that the number of component decreases. In these images there are

[*1] 1) Waterloo image database and 2) Coil 100 image database.

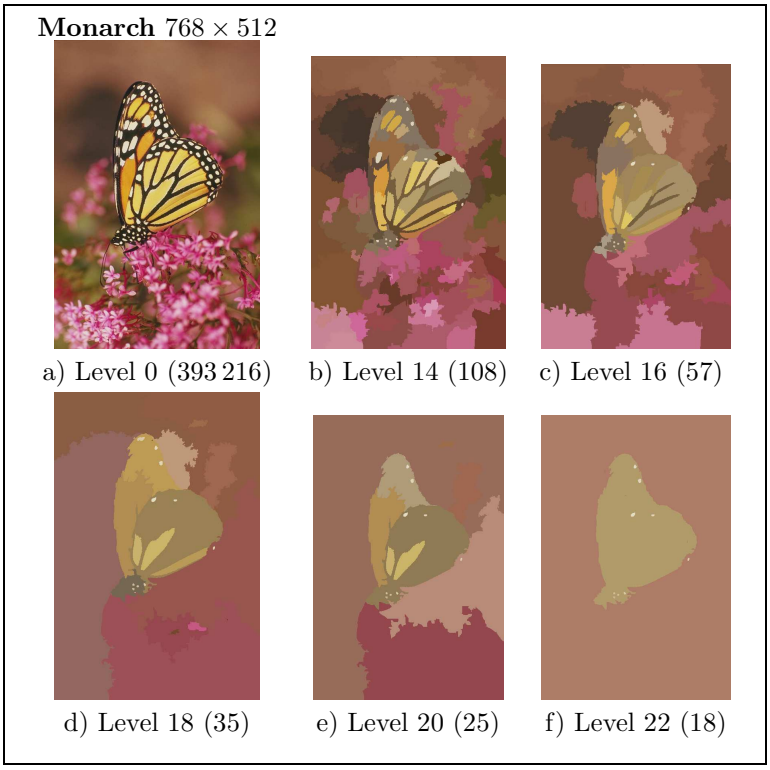

Figure 5. Some levels of the partitioning of the image *Monarch*. In parenthesis the number of components.

regions of large intensity variability (area of flowers in Figure 5a) and gradient (monarch wings in Figure 5a or in the cup example Fig. 6a). This algorithm copes with this kind of gradient and variability. In contrast to Felzenszwalb[7] the result is a hierarchy of partitions at multiple resolutions suitable for further goal driven, domain specific analysis. On lower levels of the pyramid the image is over segmented (partitioned) whereas in higher levels it is under segmented (partitioned). The help of mid and high level knowledge could 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 oversegmented image to overcome the problem of noisy pixels. The influence of τ in the decision criterion is smaller as the region gets bigger for a constant α. The constant α is used to produce a kind of the oversegmented image and the influence of τ decays after each level of the pyramid. For an oversegmeted image, where the size of regions is large, the algorithm becomes parameterless.

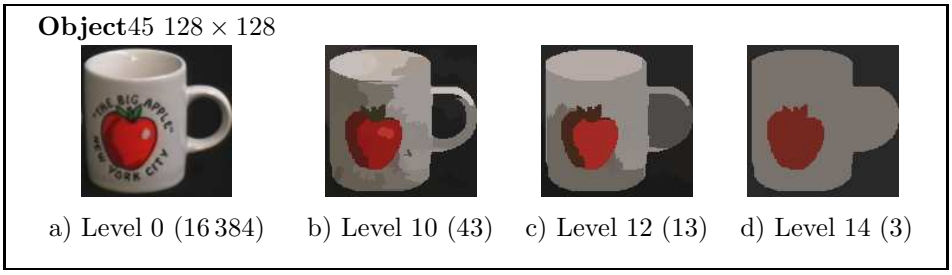

Figure 6. Some levels of the partitioning of the image *Object*45 . In parenthesis the number of components.

7. CONCLUSION AND OUTLOOK

In this paper we introduced a method to build a hierarchy of graphs of an image by comparing in a pairwise manner the difference along the boundary of two components relative to the differences of the components' internal differences. Internal properties, in our case color, helps to identify regions and their external relations to build groups of regions having a particular meanining in a more abstract context. Even though the algorithm makes simple greedy decisions locally, it produces perceptually important groupings in a bottom-up 'stimulus-driven' way based only on local differences. It was shown that the algorithm can handle large variation and gradient intensity in images. Since our framework is general enough, we can use *RAG*s of any oversegmented image and build the hierarchy of graphs afterwards. External knowledge can help in a top-down segmentation technique. A drawback is that the maximum and minimum criterion is very sensitive to noise, although in practice it has a small impact. However, other criteria like median would lead to an NP-complete algorithm. The algorithm has only one running parameter which controls the size of the regions. Our future work is to automatically extract this parameter from the image and also to define different comparison functions which will prefer learned regions of specific shapes. Being aware of the inherent limitations of pure bottom-up approach we further investigate in methods fo experience-based top-down decomposition of receptive fields with a goal to produce globally consistent interpretations.

ACKNOWLEDGMENTS

This paper has been supported by the Austrian Science Fund under grants P14445-MAT and P14662-INF. We would like to thank Michael Schreier for the software support.

REFERENCES

1. Y. Keselman and S. Dickinson, "Generic model abstraction from examples," in *Proc. IEEE Comp. Socie. Conf. on CVPR*, **1**, pp. 856–863, 2001.
2. W. G. Kropatsch, "Abstract pyramid on discrete represtations," in *DGCI 2002 LNCS*, **Vol. 2301**, I. J. O. Lachaud, A. Braquelaire, and A. Vialard, eds., pp. 1–21, Springer Verlag, 2002.
3. M. Wertheimer, "Über Gestaltheorie," *Philosoph. Zeitsch. für Forsch. und Aussp.* **1**, pp. 30–60, 1925.
4. D. E. Willis, *Source Book of Gestalt Psychology*, Harcourt, Brace and Co. New York., 1938, rep. by Gestalt Journal Press, New York 1997.
5. C.-S. Fu, W. Cho, S, and K. Essig, "Hierarchical color image region segmentation for content-based image retrival system," *IEEE Trans. on Image Proc.* **9(1)**, pp. 156–162, 2001.
6. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," in *Proc. IEEE Conf. CVPR*, pp. 731–737, 1997.
7. P. F. Felzenszwalb and D. P. Huttenlocher, "Image Segmentation Using Local Variation," in *Proc. of IEEE Conf. on CVPR*, pp. 98–104, 1998.
8. Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, and W. G. Kropatsch, "Logarithmic Tapering Graph Pyramid," in *Proc. of 24th DAGM Symposium LNCS* , **Vol. 2449**, L. van Gool, ed., pp. 117–124, 2002.
9. O. Borůvka, "O jistém problému minimálnim," *Práce Mor. Přírodvěd. Spol. v Brně (Acta Societ. Scienc. Natur. Moravicae)* **III**(3), pp. 37–58, 1926.
10. W. G. Kropatsch, "Building Irregular Pyramids by Dual Graph Contraction," *IEE-Proc. Vis., Im. and Sig. Proc.* **Vol. 142**, pp. 366–374, 1995.
11. L. Guigues, L. M. Herve, and J.-P. Cocquerez, "The Hierarchy of the Cocoons of a Graph and its Application to Image Segmentation," *Pattern Recognition Letters* **24(8)**, pp. 1059–1066, 2003.
12. Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. on PAMI* **Vol. 15**, pp. 1101–1113, 1993.
13. J. Lance and W. Williams, "A General Theory of Classificatory Sorting Strategies," *Comput. J.* **No. 9**, pp. 51–60, 1967.
14. M. Pavan and M. Pelillo, "A New Graph-Theoretic Approach to Clustring and Segmentation," in *CVPR03* **Vol. I** pp.145–152, 2003.

15. S. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *J. Assoc. Compt. Math.* **Vol. 2**(23), pp. 368–388, 1976.
16. J.-M. Jolion and A. Rosenfeld, *A Pyramid Framework for Early Vision*, Kluwer, 1994.
17. M. Bister, J. Cornelis, and A. Rosenfeld, "A critical view of pyramid segmentation algorithms," *Pattern Recognition Letters* **Vol. 11**, pp. pp. 605–617, 1990.
18. A. Montanvert, P. Meer, and A. Rosenfeld, "Hierarchical image analysis using irregular tesselations," *IEEE Trans. on PAMI* **Vol. 13**, pp.307–316, 1991.
19. J.-M. Jolion and A. Montanvert, "The adaptive pyramid, a framework for 2D image analysis," *Comp. Vis., Graph., and Im. Proc. Im. Under.* **55**, pp. 339–348, 1992.
20. P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld, "Consensus vision," in *AAAI-90 Workshop on Qualitative Vision*, pp. 111–115, 1990.
21. A. K. Jain and R.C. Doubes, *Algorithms for Clustering Data*, Prentice Hall, Berlin, 1988.
22. C. Zahn, "Graph-theoretical methods for detecting and describing gestal clusters," in *IEEE Trans. Comput.*, **20**, pp. 68–86, 1971.
23. R. Urquhart, "Graph Theoretical Clustering Based on Limited Neighborhood Sets," *Pattern Recognition* **Vol. 13(3)**, pp. 173–187, 1982.
24. F. Meyer, "Graph based morphological segmentation," in *2nd IAPR-TC-15 Workshop on GbR*, W. G. Kropatsch and J.-M. Jolion, eds., pp. 51–60, OCG-Schriftenreihe, Band 126, ÖCG, 1999.
25. Y. Haxhimusa and W. G. Kropatsch, "Hierarchical Image Partitioning with Dual Graph Contraction," in *Proc. of 25th DAGM Symposium LNCS*, **Vol. 2781**, B. Milaelis and G. Krell, eds., pp. 338–345, 2003.
26. P. Perona and W. Freeman, "A Factorization Approach to Grouping," in *LNCS*, H. Burkhardt and B. Neumann, eds., **Vol. 1406**, pp. 655–670, Computer Vision Springer Verlag, 1998.
27. Y. Gdalyahu, D Weinshall, and M. Werman, "A Randomized Algorithm for Pairwise Clustering," *Neural Information Processing Systems* **NIPS 11**, pp. 424–430, 1998.
28. M. Cooper, "The Tractibility of Segmentation and Scene Analysis," in *IJCV*, **Vol. 30(1)**, pp. 27–42, 1998.
29. T. Pavlidis, *Structural Pattern Recognition*, Springer Verlag, 1977.
30. G. Guy and G. Medioni, "Inferring Global Perceptual Contours for Pairwise Clustering," *Inter. Jour. of Comp. Vis.* , pp. 424–430, 1996.
31. J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Intern. J. on Comp. Vis.* **43(1)**, pp. 7–27, 2001.
32. W. G. Kropatsch, A. Leonardis, and H. Bischof, "Hierarchical, Adaptive and Robust Methods for Image Understanding," *Surveys on Mathematics for Industry* **No. 9**, pp. 1–47, 1999.
33. P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Tran. on Commun.* **Vol. COM-31**, pp. 532–540, 1983.
34. S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. on PAMI* **Vol. PA-11**, pp. 674–693, 1989.
35. R. Glantz and W. G. Kropatsch, "Plane embedding of dually contracted graphs," in *DGCI 2000*, G. Borgefors, I. Nyström, and G. Sanniti di Baja, eds., *LNCS* **Vol. 1953**, pp. 348–357, 2000.
36. P. Meer, "Stochastic image pyramids," *Comp. Vis., Graph., and Im. Proc.* **Vol. 45**, pp. 269–294, 1989.
37. L. Brun and W. G. Kropatsch, "Contraction Kernels and Combinatorial Maps," in *Graph-based Repres. Patt. Recog., GbR 2001*, J.-M. Jolion, W. G. Kropatsch, and M. Vento, eds., pp. 12–21, 2001.
38. L. Brun and W. G. Kropatsch, "Introduction to Combinatorial Pyramids," in *Dig. and Im. Geom.*, G. Bertrand, A. Imiya, and R. Klette, eds., pp. 108–128, 2001.
39. T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, 1990.
40. K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, Wiley-Interscience, 1992.
41. R. Cole, K. P. N., and T. R. E., "A Linear-Work Parallel Algorithm for Finding Minimum Spanning Trees," in *Proc. SPAA: Annual ACM Symp. on Parall. Alg. and Arch.*, pp. 11–15, 1994.
42. Y. Haxhimusa and W. G. Kropatsch, "Hierarchical Image Partitioning with Dual Graph Contraction," Tech. Rep. No.81, PRIP, Vienna University of Technology, http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr81.pdf, 2003.