

Segmentation Graph Hierarchies*

Yll Haxhimusa and Walter Kropatsch

Pattern Recognition and Image Processing Group 183/2
Institute for Computer Aided Automation, Vienna University of Technology, Austria
{yll,krw}@prip.tuwien.ac.at

Abstract. The region's internal properties (color, texture, ...) help to identify them and their external relations (adjacency, inclusion, ...) 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. The aim of this paper is to build a minimum weight spanning tree (*MST*) of an image in order to find region borders quickly in a bottom-up 'stimulus-driven' way based on local differences in a specific feature.

1 Introduction

The authors in [16] 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 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: i) **region segmentation**, ii) **perceptual grouping**, and iii) image abstraction. In this paper, these goals are taken to consider multiresolution representations under the viewpoint of segmentation and grouping. The multiresolution is considered under the abstraction viewpoint in [18].

The union of regions forming the group is again a region with both internal and external properties and relations. The segmentation process results in "homogeneous" regions w.r.t the low-level cues using some similarity measures. Problems emerge because i) homogeneity of low-level cues will not map to the semantics [16] and ii) the degree of homogeneity of a region is in general quantified by threshold(s) for a given measure [6]. The low-level coherence of brightness, color, texture or motion attributes should be used to come up sequentially with hierarchical partitions [29]. It is important that a grouping method has following properties [5]: 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 pixels, and iii) creates **hierarchical partitions** [29].

* This paper has been supported by the Austrian Science Fund under grants P14445-MAT, P14662-INF and FSP-S9103-N04.

The aim is to build an *MST* of an image by combining the advantage of regular pyramids (logarithmic tapering) with the advantages of irregular graph pyramids (their purely local construction and shift invariance). The aim is reached by using the method for selecting contraction kernels to achieve logarithmic tapering, local construction and shift invariance [12]. Borůvka's algorithm [2] with dual graph contraction (*DGC*) algorithm [17] builds in a hierarchical way an *MST* (of the region) preserving the proper topology. After presenting related works and the pyramid representation, we recall the Borůvka's *MST* algorithm in Sec. 3. In Sec. 4 are given the definition of internal and external contrast, the merging criteria and building the hierarchy of an image. Sec. 5 reports results.

1.1 Related Works

A graph-theoretical clustering algorithm consists in searching for a certain combinatorial structure in the edge weighted graph, such as an *MST* [5, 8], a minimum cut [31, 29] and, the complete linkage clustering algorithm [19], reduces to a search for a complete subgraph i.e. the maximal clique [24]. Early graph-based methods [33] use fixed thresholds and local measures in finding a segmentation, i.e. *MST* is computed. The segmentation criterion is to break the *MST* edges with the largest weight, which reflect the low-cost connection between two elements. To overcome the problem of fixed threshold, [30] attempts by normalizing the weight of an edge using the smallest weight incident on the vertices touching that edge. The methods in [5, 8] use an adaptive criterion that depends on local properties rather than global ones. The methods based on minimum cuts in a graph are designed to minimize the similarity between pixels that are being split [31, 29, 7]. A cut criterion in [31] is biased toward finding small components. The normalized cut criterion [29] is defined to avoid this problem, which takes into consideration self-similarity of regions. In contrast with the simple graph-based methods, such as breaking edges in the *MST*, cut-criterion methods capture the non-local properties of the image. It also produces a divisive hierarchical tree, the dendrogram. However, they provide only a characterization of such cut rather than of final segmentation as provided in [5]. A minimum normalized cut approximation method [29] is computationally expensive and the error in these approximation is not well understood. The clustering algorithms based on maximal clique work on unweighted graphs derived from the weighted graphs by means of some thresholding [15]. In [24] the concept of maximal clique is generalized to weighted graphs. Discrete replicator dynamics are used to find the maximal cliques, which is an instance of the relaxation labeling algorithm [27].

Gestalt grouping factors, such as proximity, similarity, continuity and symmetry, are encoded and combined in pairwise feature similarity measures [29, 26, 7, 32, 28]. Another method of segmentation is that of splitting and merging region based on how well the regions fulfill some criterion. Such methods [4, 25] use a measure of uniformity of a region. In contrast, [5, 8] uses a pairwise region comparison rather than applying a uniformity criterion to each individual region. Complex grouping phenomena can emerge from simple computation on these local cues [20].

Our method is related to the works in [5, 8] in the sense of pairwise comparison of region similarity. Rather than trying to have just one “good” segmentation [5], the method produces a stack of (dual) graphs (a graph pyramid), which down-projected on the base level will give a multi-level segmentation (a class of segmentation). The segmentation result of [5] is also included in our hierarchy.

2 Irregular Pyramid Representation

Hierarchical structures for description of the data for clustering purposes are studied very early in [19], or for image segmentation in [11]. In a regular image pyramid the number of pixels at any level k , is λ times higher than the number of pixels at the next reduced level $k + 1$. The so called reduction factor λ is greater than one and it is the same for all levels k . If s denotes the number of pixels in an image I , the number of new levels on top of I amounts to $\log_\lambda(s)$ (Figure 1a). Thus, the regular image pyramid is an efficient structure for fast grouping and access to image objects in top-down and bottom-up processes [14], because of the apriori known structure. However, the regular image pyramids are confined to globally defined sampling grids and lack shift invariance and have to be rejected as general-purpose segmentation algorithms [1]. These drawbacks are avoided by irregular image pyramids (adaptive pyramids) [23, 13], where the hierarchical structure (vertical network) of the pyramid is recursively built on the data. In [22] is shown that irregular pyramid can be used for segmentation.

In irregular pyramids, 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 [21, 12]. 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 -connectivity of the pixels. A cell on level $k + 1$ (parent) is a union of neighboring cells on level k (children). This union is controlled by so called contraction kernels (decimation parameters) [17]. Every parent computes its values independently of other cells on the same level. This implies that an image pyramid is built in $O[\log(\text{image_diameter})]$ parallel steps. We represent the levels as dual pairs (G_k, \overline{G}_k) of plane primal graphs G_k and duals \overline{G}_k . The vertices of G_k represent the cells, and edges the neighborhood relations of the cells on level k . The edges of \overline{G}_k represent the borders of the cells on level k and vertices meeting points of at least three edges from G_k . The sequence (G_k, \overline{G}_k) , $0 \leq k \leq h$ is called (dual) **graph pyramid**. See [17] for the complete formalism. In order to simplify the paper presentation only graph G_k is used afterwards.

3 Minimum Weight Spanning Tree: Borůvka’s Algorithm

Let $G_0(V, E, attr_v, attr_e)$ be a given undirected, connected, attributed plane graph consisting of the finite vertex set V and of the finite edge set E on the base level (level 0) of the pyramid, $attr_v : v \in V \rightarrow \mathbb{R}^+$ and $attr_e : e \in E \rightarrow \mathbb{R}^+$. Let each edge $e \in E$ be associated with a nonnegative **unique** real attribute

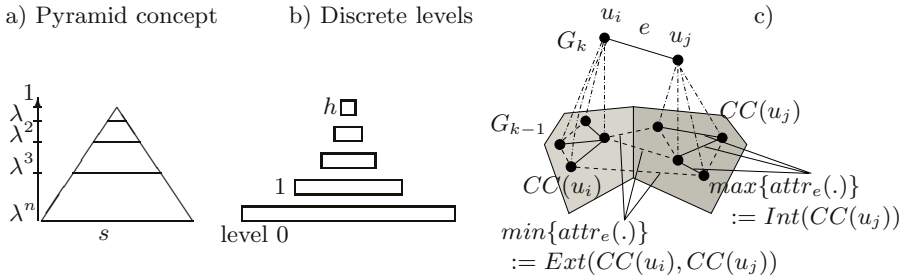


Fig. 1. (a,b) Multiresolution pyramid. (c) Internal and external contrast.

Algorithm 1 – Borůvka’s Algorithm.

Input: Attributed graph $G(V, E)$.

- 1: $MST :=$ empty edge list.
- 2: $\forall v \in V$ **do** { make a list of 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 .

Output: Minimum weight spanning tree - MST .

(weight). The problem is formulated as construction of an MST of G . A deterministic solution is proposed by Borůvka [2]. Borůvka algorithm is similar to Prim’s algorithm but executed simultaneously on the whole graph. We use Borůvka’s algorithm to build MST in parallel [3]. The proof that Alg. 1 builds the MST is analogous with the MST Kruskal’s proof [10].

Observation 1 *In 4th step of the Algorithm 1, 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.*

4 Hierarchy of Partitions

Hierarchies are a significant tool for image partitioning as they naturally mix with homogeneity criteria [11]. The goal is to find partitions $P_k := \{CC(u_1), CC(u_2), \dots, CC(u_n)\}$ in k^{th} level of the pyramid such that these elements satisfy certain properties. We compare pairwise of neighboring vertices, i.e. partitions to check for similarities [5, 8]. In [5] a pairwise group merge criterion $Comp(CC(u_i), CC(u_j))$ is defined, that judges whether there is evidence for a boundary between two partitions $CC(u_i), CC(u_j) \in P_k$. 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**).

4.1 Internal and External Contrast

Let G_k be the graph on level k of the pyramid. Every vertex $u_i \in G_k$ is a representative of a **connected component** $CC(u_i)$ of the partition P_k . The equivalent contraction kernel (*ECK*) $N_{0,k}(u_i)$ [17] of a vertex $u_i \in G_k$, is a set of edges of the base level $e \in E_0$ that are contracted; i.e. applying *ECK* on the base level, one contracts the subgraph $G' \subseteq G_0$ onto the vertex u_i . The **internal contrast** of the $CC(u_i) \in P_k$ is the **largest dissimilarity** of component $CC(u_i)$, i.e. the largest edge weight of the $N_{0,k}(u_i)$ of vertex $u_i \in G_k$:

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

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

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

In Fig. 1c an example of $Int(CC(u_j))$ and $Ext(CC(u_i), CC(u_j))$ is given. The $Int(CC(u_j))$ of the component $CC(u_j)$ is the *maximum* of weights of the solid line edges, whereas $Ext(CC(u_i), CC(u_j))$ is the *minimum* of weights of the dashed line edges (bridges) connecting component $CC(u_i)$ and $CC(u_j)$ on the base level G_0 . By contracting the edges $N_{0,k}(u_j)$ one arrives to the vertex u_j . The pairwise merge criterion $Comp(CC(u_i), CC(u_j))$ between two connected components $CC(u_i)$ and $CC(u_j)$ can be defined as:

$$Comp(CC(u_i), CC(u_j)) \\ := \begin{cases} \text{T} & \text{if } Ext(CC(u_i), CC(u_j)) \leq PInt(CC(u_i), CC(u_j)), \\ \text{F} & \text{otherwise,} \end{cases} \\ PInt(CC(u_i), CC(u_j)) := \min(Int(CC(u_i)) + \tau(CC(u_i)), \\ Int(CC(u_j)) + \tau(CC(u_j))). \quad (3)$$

$PInt(\cdot, \cdot)$ is the minimum internal contrast difference between two components. For the merge criterion $Comp(\cdot, \cdot)$ to be false i.e. for the border to exist, the external contrast difference must be greater than the internal contrast differences. Any non-negative function $\tau(CC)$ of a single component CC can be used [5].

4.2 Construct Hierarchy of Partitions

A consequence of Obs. 1 is that a contraction of the edge e , which connects T and $G \setminus T$ in the 4th step of Alg. 1 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 *DGC* algorithm [17] contracts edges and creates "super" vertices with father-son relations between vertices in

Algorithm 2 – Construct Hierarchy of Partitions.

Input: Attributed graph G_0 .

```

1:  $k := 0$ 
2: repeat
3:   for all { vertices  $u \in G_k$  } do
4:      $E_{min}(u) := \operatorname{argmin}\{attr_e(e) \mid e := (u, v) \in E_k \text{ or } e := (v, u) \in E_k\}$ 
5:   for all {  $e := (u_i, u_j) \in E_{min}$  with
      $Ext(CC(u_i), CC(u_j)) \leq PInt(CC(u_i), CC(u_j))$  } do
6:     include  $e$  in contraction edges  $N_{k,k+1}$ 
7:   contract graph  $G_k$  with contraction kernels,  $N_{k,k+1}$ :  $G_{k+1} = C[G_k, N_{k,k+1}]$ .
8:   for all {  $e_{k+1} \in G_{k+1}$  } do
9:     set edge attributes  $attr_e(e_{k+1}) := \min\{attr_e(e_k) \mid e_{k+1} := C[e_k, N_{k,k+1}]\}$ 
10:   $k := k + 1$ 
11: until {  $G_k = G_{k-1}$  }

```

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

subsequent levels, whereas Borůvka’s algorithm is used to create son-son relation between vertices in the same level (horizontal relation).

The algorithm to build the hierarchy of partitions is shown in Alg. 2. Each vertex $u_i \in G_k$ defines a **connected region** $CC(u_i)$ on the base level of the pyramid. Since the presented algorithm is based on Borůvka’s algorithm [2], it builds an $MST(u_i)$ of each region, i.e $N_{0,k}(u_i) = MST(u_i)$ [9]. The idea is to collect the smallest weighted edges e (4th step) that could be part of the MST , and then to check if the edge weight $attr_e(e)$ is smaller than the internal contrast of both of the components (MST of end vertices of e) (5th step). If these conditions are fulfilled then these two components are merged (7th step). Two regions will be merged if their internal contrast is larger than the external contrast, represented by the weight $attr_e(e)$ of the connecting edge. All the edges to be contracted form the contraction kernels $N_{k,k+1}$, which are then used to create the graph $G_{k+1} = C[G_k, N_{k,k+1}]$ [17]. 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} (9th step). This means the edge attributes are inherited. It can be shown, that Alg. 2 produces an MST [10]. At each level of the pyramid a region adjacency graph (*RAG*) is created, in an agglomerative way by topology preserving edge contraction. Each vertex of these *RAGs* is the representative of a sub-trees of MST . This greedy algorithm collects only the nearest neighbor with the minimum edge weights, known as *single linkage clustering*, and merges them if the pairwise comparison (Eq. 3) evaluates to “false”. See [9, 10, 5] for other properties of this algorithm.

5 Experiments on Image Graphs

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 features of

end vertices, $attr_e(u_i, u_j) := |F(u_i) - F(u_j)|$, where F is some feature. Other attributes could be used as well e.g. [29] $attr_e(u_i, u_j) := exp\{\frac{-||F(u_i) - F(u_j)||_2^2}{\sigma_I}\}$, 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 distance [29]. 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 [20]. For our experiments we use the difference between pixel intensities $F(u_i) := I(u_i)$, i.e. $attr_e(u_i, u_j) := |I(u_i) - I(u_j)|$. For color images we run the algorithm by computing the distances (weights) in *RGB* color space. We define threshold function $\tau(CC)$ to be function of the size of CC e.g. $\tau(CC) := \alpha/|CC|$, where $|CC|$ is the size of the component CC and α is a constant. This function controls the influence of the size of the components CC . The algorithm has one running parameter α . This constant is used to produce a kind of the over-segmented image. The influence of τ in Eq.3 decays after each level of the pyramid, since the $|CC|$ gets bigger. A larger α sets the preference for larger components. A complex definition of $\tau(CC)$, which is large for certain shapes would produce a partitioning which prefers certain shapes.

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, *Object45*² and *Object11*² Fig.2(1,3,4) and $\alpha := 1000$ for the image in Fig.2(2), after the average intensity attribute of vertices is down-projected onto the base grid. Fig. 2 show some of the partitions on different levels of pyramid and the number of components. In all images there are regions of large intensity variability and gradient. This algorithm copes with this kind of images.

In contrast to [5] 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 whereas in higher levels it is under-segmented. Since the algorithm preserves details in low-variability regions, a noisy pixel would survive through the hierarchy. Image smoothing in low variability regions would overcome this problem, and it is not done, since 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. For an over-segmented image, where the size of regions is large, the algorithm becomes parameterless.

6 Conclusion and Outlook

In this paper we introduced a method to build a hierarchy of partitions of an image by comparing in a pairwise manner the difference along the boundary of two components relative to the differences of components' internal differences. Even though the algorithm takes simple greedy decisions locally, it produces perceptually important partitions in a bottom-up 'stimulus-driven' way based only on local differences. It was shown that the algorithm can handle large variation

¹ 1)Waterloo image database and 2) Coil 100 image database

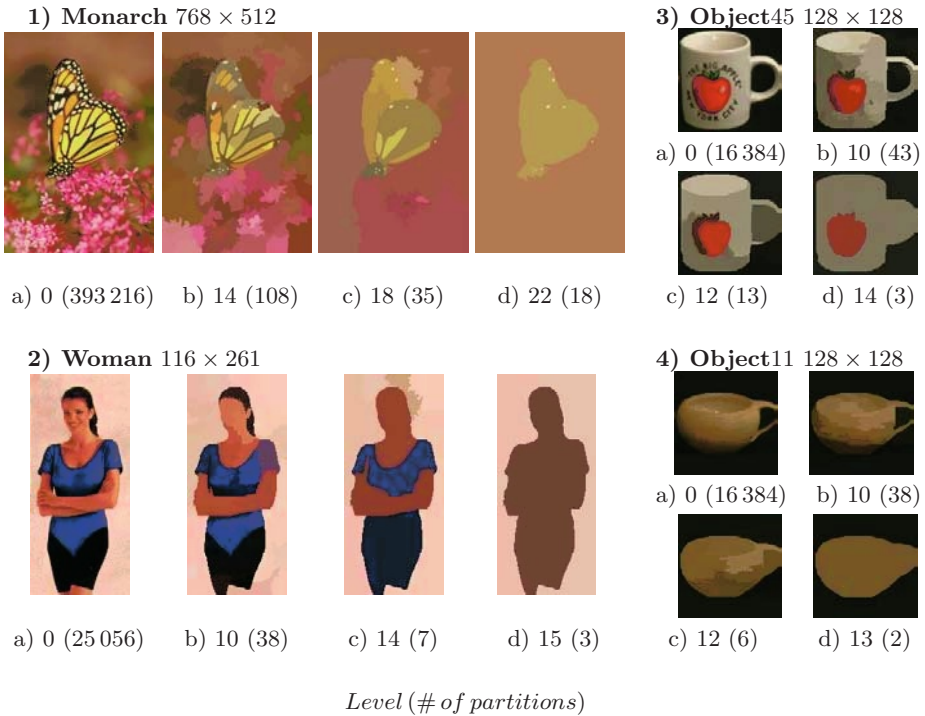


Fig. 2. Partitioning of images.

and gradient intensity in images. Since our framework is general enough, we can use *RAGs* of any over-segmented image and build the hierarchy of partitions. External knowledge can help in a top-down segmentation technique. A drawback is that the maximum and minimum criterion are very sensitive to noise, although in practice it has a small impact. Other criteria like median would lead to an *NP*-complete algorithm [5]. Our future work is to define different comparison functions which will prefer learned regions of specific shapes.

References

1. M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *PRL*, 11(9):605–617, 1990.
2. O. Borůvka. O jistém problému minimálním. *Práce Mor. Přírodověd. Spol. v Brně (Acta Societ. Scienc. Natur. Moravicae)*, III(3):37–58, 1926.
3. R. Cole, K. P. N., and T. R. E. A Linear-Work Parallel Algorithm for Finding Minimum Spanning Trees. *Annual SPAA*, 11–15, 1994.
4. M. Cooper. The Tractability of Segmentation and Scene Analysis. *IJCV*, 30(1):27–42, 1998.
5. P. F. Felzenszwalb and D. P. Huttenlocher. Image Segmentation Using Local Variation. *CVPR*, 98–104, 1998.
6. C.-S. Fu, W. Cho, S., and K. Essig. Hierarchical color image region segmentation for content-based image retrieval system. *IP*, 9(1):156–162, 2001.

7. Y. Gdalyahu, D. Weinshall, and M. Werman. Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization. *PAMI*, 23(10):1053–1074, 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. *PRL*, 24(8):1059–1066, 2003.
9. Y. Haxhimusa and W. G. Kropatsch. Hierarchical Image Partitioning with Dual Graph Contraction. B. Milaelis and G. Krell, eds., *LNCS 2781*, 338–345, 2003.
10. Y. Haxhimusa and W. G. Kropatsch. Hierarchical Image Partitioning with Dual Graph Contraction. Technical Report No.81, PRIP, Vienna University of Technology, <http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr81.pdf>, July 2003.
11. S. Horowitz and T. Pavlidis. Picture Segmentation by a Tree Traversal Algorithm. *J. Assoc. Compt. Math.*, 2(23):368–388, 1976.
12. Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, and W. G. Kropatsch. Logarithmic Tapering Graph Pyramid. L. van Gool, ed., *LNCS 2449*, 117–124, 2002.
13. J.-M. Jolion and A. Montanvert. The adaptive pyramid, a framework for 2D image analysis. *CVGIP: Im. Under.*, 55(3):339–348, 1992.
14. J.-M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision*. 1994.
15. A. K. Jain and R. C. Dubes *Algorithms for Clustering Data*. 1988.
16. Y. Keselman and S. Dickinson. Generic model abstraction from examples. *CVPR*, 1:856–863, 2001.
17. W. G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vis. Im and Sig. Proc.*, 142(6):366–374, 1995.
18. W. G. Kropatsch. Abstract pyramid on discrete representations. I. J. O. Lachaud, A. Braquelaire, and A. Vialard, eds., *LNCS 2301*, 1–21, 2002.
19. J. Lance and W. Williams. A General Theory of Classificatory Sorting Strategies. *Comput. J*, 9:51–60, 1967.
20. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and Texture Analysis for Image Segmentation. *Int. J. on Com. Vis.*, 43(1):7–27, 2001.
21. P. Meer. Stochastic image pyramids. *CVGIP*, 45(3):269–294, 1989.
22. P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld. Consensus vision. *AAAI-90 Work. on Qual. Vis.*, 111–115, 1990.
23. A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *PAMI*, 13(4):307–316, 1991.
24. M. Pavan and M. Pelillo. Dominant Sets and Hierarchical Clustering. *ICCV*, 2003.
25. T. Pavlidis. *Structural Pattern Recognition*. 1977.
26. P. Perona and W. Freeman. A Factorization Approach to Grouping. H. Burkhardt and B. Neumann, eds., *LNCS 1406*, 655–670, 1998.
27. A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *Sys. Man and Cyb.*, 6(6):420–433, 1976.
28. E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. *CVPR*, 70–77, 2000.
29. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *CVPR*, 731–737, 1997.
30. R. Urquhart. Graph Theoretical Clustering Based on Limited Neighborhood Sets. *Patt.Rec.*, 13(3):173–187, 1982.
31. Z. Wu and R. Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *PAMI*, 15(11):1101–1113.
32. S. Yu and J. Shi. Segmentation with pairwise attraction and repulsion. *ICCV*, 52–58, 2001.
33. C. Zahn. Graph-theoretical methods for detecting and describing gestal clusters. *IEEE Comput.*, 20:68–86, 1971.