Available online at www.sciencedirect.com
SCIENCE DIRECT®

Pattern Recognition 39 (2006) 1430 – 1451

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Pyramid segmentation algorithms revisited☆

R. Marfil*, L. Molina-Tanco, A. Bandera, J.A. Rodríguez, F. Sandoval

Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga, Campus de Teatinos s/n, 29071 Málaga, Spain

Received 28 July 2005; received in revised form 17 January 2006; accepted 1 February 2006

Abstract

The main goal of this work is to compare pyramidal structures proposed to solve segmentation tasks. Segmentation algorithms based on regular and irregular pyramids are described, together with the data structures and decimation procedures which encode and manage the information in the pyramid. In order to compare the different segmentation algorithms, we have employed three types of quality measurements: the shift variance measure, the F function and the Q function.

Keywords: Image segmentation; Irregular pyramids; Regular pyramids; Combinatorial map; Graphs; Data structures; Decimation schemes

1. Introduction

Segmentation is an important task in image processing that plays a chief role in understanding images. It can be defined as the process of decomposing an image into regions which are homogeneous according to some criteria. In this process, the internal properties of the region help to identify it, while its external properties (inclusion, adjacency, ...) are used to group regions having a "*particular consistent meaning*" [1]. The segmentation algorithm must adapt the resulting regions to the contents of the image. It is also interesting for the majority of applications that the segmentation algorithm assures the spatial consistency or intra-connectivity of these regions. Pyramids are hierarchical structures which have been widely used in segmentation tasks [2,3]. Instead of performing image segmentation based on a single representation of the input image, a pyramid segmentation algorithm describes the contents of the image using multiple representations with decreasing resolution. In this hierarchy, each representation or level is built by computing a set of local operations over the level below, being the original image the level 0 or base level of the hierarchy. Pyramid segmentation algorithms exhibit interesting properties with respect to segmentation algorithms based on a single representation. Thus, local operations can adapt the pyramidal hierarchy to the topology of the image, allowing the detection of global features of interest and representing them at low resolution levels [4–6]. Bister et al. [4] emphasize the reduction of noise and the processing of local and global features within the same framework. Finally, the hierarchical nature of the pyramid algorithm allows a reduction in the complexity of the image segmentation task [4,6].

The general principle of the pyramidal approach was briefly described by Jolion and Montanvert [7]: "a global interpretation is obtained by a local evidence accumulation". In order to accumulate this local evidence, a pyramid represents the contents of an image at multiple levels of abstraction. Each level of this hierarchy is at least defined by a set of vertices V_l connected by a set of edges E_l. These edges define the horizontal relationships of the pyramid and represent the neighbourhood of each vertex at the same level (intra-level edges). Another set of edges define the vertical relationships by connecting vertices between adjacent pyramid levels (inter-level edges). These inter-level edges establish a dependency relationship between each vertex of level l+1 and a set of vertices at level l (reduction window). The vertices belonging to one reduction window are

☆ This work has been partially granted by Spanish Ministerio de Educación y Ciencia (MEC) project no. TIN2004-05961 and by EURON project VISOR.

* Corresponding author. Tel.: +34 952137223.
E-mail address: rebeca@dte.uma.es (R. Marfil).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.02.017

the sons of the vertex which defines it. The value of each parent is computed from the set of values of its sons using a reduction function. The ratio between the number of vertices at level *l* and the number of vertices at level *l*+1 is the reduction factor. Using this general framework, the local evidence accumulation is achieved by the successive building of level $G_{l+1} = (V_{l+1}, E_{l+1})$ from level $G_l = (V_l, E_l)$. This procedure consists of three steps:

(1) *Selection of the vertices of G_{l+1} among V_l*: This selection step is a decimation procedure and selected vertices V_{l+1} are called the surviving vertices.
(2) *Inter-level edges definition*: Each vertex of G_l is linked to its parent vertex in G_{l+1}. This step defines a partition of V_l.
(3) *Intra-level edges definition*: The set of edges E_{l+1} is obtained by defining the adjacency relationships between the vertices V_{l+1}.

The parent–son relationship defined by the reduction window may be extended by transitivity down to the base level. The set of sons of one vertex in the base level is named its receptive field. The receptive field defines the embedding of this vertex in the original image. Global properties of a receptive field with a diameter d can be computed in $O(\log(d))$ parallel processing steps using this parent–son relationship. In a general view of the pyramid hierarchy, the vertices of the bottom pyramidal level (level 0, also called base level) can be anything from an original image pixel via some general numeric property to symbolic information, e.g. a vertex can represent an image pixel grey level or an image edge. Corresponding to the generalization of the vertex contents, the intra-level and inter-level relations of the vertices are also generalized.

After building the pyramidal structure, the segmentation of the input image can be achieved either by selecting a set of vertices from the whole hierarchy as region roots, or by choosing as roots all the vertices which constitute a level of this hierarchy. In any case, this selection process depends on the final application and it must be performed by a higher level task. The efficiency of a pyramid to solve segmentation tasks is strongly influenced by two related features that define the intra-level and inter-level relationships. These features are the data structure used within the pyramid and the decimation scheme used to build one graph from the graph below [8]. The choice of a data structure determines the information that may be encoded at each level of the pyramid. It defines the way in which edges E_{l+1} are obtained. Thus, it roughly corresponds to setting the horizontal properties of the pyramid. On the other hand, the reduction scheme used to build the pyramid determines the dynamics of the pyramid (height, preservation of details, etc.). It determines the surviving vertices of a level and the inter-level edges between levels. It corresponds to the vertical properties of the pyramid. Taking into account these features, pyramids have been roughly classified as regular and irregular pyramids. A regular pyramid has a rigid structure where the intra-level relationships are fixed and the reduction factor are constant. In these pyramids, the inter-level edges are the only relationships that can be changed to adapt the pyramid to the image layout. The inflexibility of these structures has the advantage that the size and the layout of the structure are always fixed and well-known. However, regular pyramids can suffer several problems [4,9]: non-connectivity of the obtained receptive fields, shift variance, or incapability to segment elongated objects. In order to avoid these problems, irregular pyramids were introduced. In the irregular pyramid framework, the spatial relationships and the reduction factor are not constant. Original irregular pyramids presented a serious drawback with respect to computational efficiency because they gave up the well-defined neighbourhood structure of regular pyramids. Thus, the pyramid size cannot be bounded and hence neither can the time to execute local operations at each level [10]. This problem has been resolved by recently proposed strategies [8,11–14].

The main goal of this paper is to explain and compare pyramid structures proposed to solve segmentation tasks. It must be stressed that our aim is not to evaluate the performance of these segmentation algorithms. The ill-defined nature of the segmentation problem prevents us from making such evaluation. As several authors have previously pointed out, it is not possible to assure that an algorithm provides a final "good" segmentation of the image. In general, "it is not clear what a "good" segmentation is" [6]. Fortunately, although segmentation evaluation and comparison are closely related, they are in fact different matters. In this paper, our goal is to rank the performance of several pyramid segmentation algorithms using different measures. Specifically, the chosen measurements are the shift variance (SV) proposed by Prewer and Kitchen in Ref. [15], the F function proposed in Ref. [16] and the Q function proposed in Ref. [17]. The first descriptor was introduced to evaluate the stability of a segmentation algorithm when minor shifts occur [15]. The F and Q functions were defined to take into account some of the criteria suggested by Haralick and Shapiro [18]: (i) segmentation regions must be homogeneous and without too many small holes in their interiors and (ii) adjacent regions must present significantly different values for homogeneous properties.

The rest of the paper is organized as follows: Sections 2 and 3 describe segmentation algorithms based on regular and irregular pyramids, respectively. These sections introduce data structures and decimation procedures proposed to encode and manage the information at the pyramid. Pyramid segmentation algorithms are compared in Section 4. Finally, Section 5 summarizes the main conclusions obtained from the experimental results.

2. Regular pyramids

Regular pyramids can be explained as a graph hierarchy. However, it is more usual to represent them as a hierarchy of

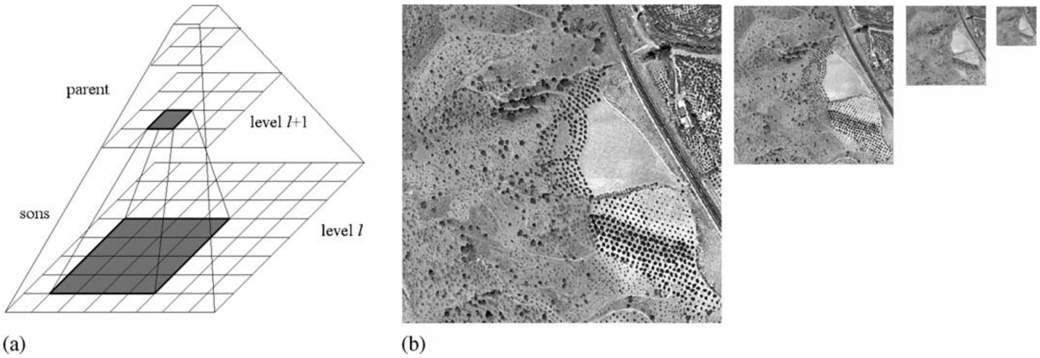

Fig. 1. Regular pyramids: (a) A $4 \times 4/4$ regular pyramid; (b) different levels of a $2 \times 2/4$ pyramid.

image arrays due to their rigid structure (see Fig. 1). Thus, a node of a regular pyramid can be defined by its position (i, j, l) in the hierarchy, being l the level of the pyramid and (i, j) its (x, y) coordinates within the level. In each of these arrays two nodes are neighbours if they are placed in adjacent positions of the array. The possibility to express the regular pyramids as a hierarchy of image arrays with well-defined neighbourhood relationships is the main advantage of these kind of pyramids, because it allows to build and traverse them with a low computational cost.

2.1. Regular pyramid data structure

The usefulness of pyramidal structures in image processing was firstly pointed out in Refs. [19,20]. In these pyramids, inter- and intra-level relationships are fixed, so the structure only reduces the resolution of the image in successive levels. On the base level of the pyramid, the vertices represent single pixels and the neighbourhood of the vertices is defined by the 4- or 8-connectivity of the pixels (Fig. 1a). Each pyramid level is recursively obtained by processing its underlying level. Fig. 1b shows that these pyramids generate a set of bandpass-filtered versions of an image, and they do not exploit their intrinsic capability to reliable delineate the significant features in an image [21,22]. The son–parent relationships are fixed and for each vertex in level l+1, there is a $N \times N$ reduction window of sons at level l. A regular pyramid is thus defined by the ratio $N \times N/q$, where $N \times N$ is the size of the reduction window and q the reduction factor or fixed ratio between the sizes of two consecutive levels of the pyramid [23]. When the ratio $N \times N/q$ is greater than one, reduction windows are overlapped, and the parent selection scheme can be easily modified: each vertex v_i at level l could now be linked to any of its potential parents, which are the set of vertices at level l+1 whose reduction window includes v_i. Therefore, in a regular pyramidal structure, inter-level relationships could adapt itself to the image layout. Since the inter-level edges determine the partition of the set of vertices V_l, image segmentation is implemented as the process that selects a single legitimate parent for each vertex from the vertex's candidate parents. This legitimate parent could be the candidate with the most similar value to that of the vertex itself.

2.2. Segmentation algorithms based on regular pyramids

Chen and Pavlidis [24] proposed the first pyramidal segmentation algorithm. In this approach, they define son–parent edges, which constitute vertical relationships in the pyramid, and brother–brother edges (horizontal relationships). The rigidity of this pyramid structure may give rise to artefacts [25,26]. Specially, the difficulty of handling long-shaped features in an image was closely related to the limitations of image pyramids [27]. To compensate for these artefacts, different regular pyramids were proposed. Thus, Shneier [28] focused his research on extraction of linear features from an image. However, Shneier only uses multi-resolution images to define local thresholds in a classical local thresholding method, which is not really a pure pyramid segmentation technique. Other approaches control the resolution reduction by the local image content [29,30]. These approaches recalculate the son–parent relationships in order to achieve the adaptation of the pyramidal structure to the image layout. Particularly, the son–parent relationships are refined over several iterations, so these approaches are named global iterative approaches [31]. A typical iteration may consist of a bottom-up linking process, a bottom-up recomputation of vertex values and a top-down reassignment of vertex values. After several iterations, the inter-level edges will normally have stabilized and the segmented image is obtained from the base level vertex values. Although these global iterative approaches to pyramidal segmentation exhibit superior performance over the classical top-down approaches, this performance advantage must be considered against the greater computational requirements of the iterative algorithms [31]. Still, global iterative approaches can be considered as the main type of regular pyramidal structures and are explained below.

In other pyramidal approaches, the pyramid is built using several types of Gaussian filters. Ping et al. [32] use a Gaussian filter function with changeable filter scales. By modifying the filter scale, this algorithm changes the window

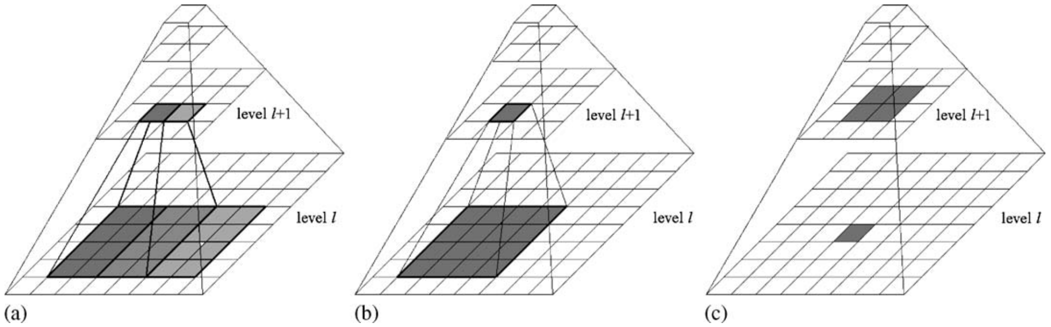

Fig. 2. Linked pyramid: (a) overlapped at the linked pyramid; (b) the sixteen grey vertices in level l are the candidate sons for the grey vertex in level $l+1$; and (c) the four grey vertices in level $l+1$ are the candidate parents for the grey vertex in level l.

size between pyramidal levels. When applied to segmentation, this algorithm searches for vertices in the structure that can be regarded as roots of segmented regions at the base level. Regular pyramids normally employ a square window but there are regular structures with triangular and hexagonal windows [33,34]. Another possible modification consists in reducing the size only by half between pyramidal levels [35].

2.2.1. Pyramid linking approach (PLA)

Burt et al. [29] originally proposed the Linked Pyramid in 1981. In this pyramid, during the first iteration, each 4×4 set of vertices within a level generates a new vertex in the upper level by averaging the local image property values of the vertices in the reduction window ($4 \times 4/4$ pyramid). Each level has a size four times smaller than the level below, because the 4×4 windows are 50% overlapped, as it is shown in Fig. 2a. For each vertex at level l there is a 4×4 subarray of "candidate son" vertices at level $l-1$ (Fig. 2b). The vertex itself is a member of four such subarrays for level $l+1$ vertex. On each iteration, the whole of the structure is covered and every vertex is linked to the most similar candidate parent from the higher level (Fig. 2c). After linking, each vertex will have between 0 and 16 legitimate sons. The local image property value of each parent is recalculated by averaging the local image property of its sons. This process continues until the son–parent edges do not vary. Finally, in order to perform the image segmentation, a level of the pyramid (called working level) is selected as the level in charge to generate the segmentation. Each working level vertex is linked to a set of vertices at the base of the structure. These vertices represent its receptive field and define a segmented region. The local image property values from the working level vertices are propagated to their corresponding regions at the base. These regions constitute the segmented image. The selection of the working level is very important because it sets the number of resulting segmented regions, which is approximately equal to the number of vertices at the working level (there could exist vertices at the working level with null receptive field). It must be noted that the correct working level depends on the content of the image to segment, being unknown at the beginning of the process. The accuracy of the final segmentation depends on the correct selection of the working level. Because of its apparent flexibility, this adaptive hierarchical structure has been investigated by other researchers [36,37].

The linked pyramid, as originally proposed in Ref. [29] presents four main problems. The first is the aforementioned need of choosing the working level. The other three are related to the inflexibility of the structure [4]:

- The region connectivity is not preserved: In the son–parent relinking process, this structure does not take into account adjacency criteria in the original image; hence, adjacent nodes in a level do not necessarily originate from adjacent segmented regions at the base level. When these vertices are grouped into a new vertex in the upper level, the new vertex is associated to a disjoint region at the base.
- Non-adaptability to the image layout: Due to the use of a fixed size rectangular reduction window, the shape of elongated regions cannot be represented in the segmented image.
- The structure is variant to small shifts, rotations or scale modifications in the original image. This problem is commonly named the SV problem.

On the other hand, this pyramid does not need any threshold to compute similarity between nodes. This is probably its main advantage.

It must be noted that the notion of working level is not mandatory. In Ref. [29], the final segmentation is defined by pre-selecting the final level in the linking process (the working level). The working level has to be defined manually and determines the approximate number of final segmented regions obtained. Large regions with similar homogeneous local image property values usually persist through the linking process up to the highest levels, whereas smaller homogeneous regions may merge with their surroundings in a lower level. The need for pre-selecting the final level can be removed by introducing unforced linking [9] which allows the exclusion of some vertices from the linking scheme. Thus, region roots can be selected in the pyramid at different levels if the set of receptive fields forms a partition of the initial image. This exclusion procedure, referred as seeding, allows the number of segmented regions to automatically adapt to the image content, which results in extraction of small homogeneous regions as well as large ones. Seeding rules determine whether a vertex is excluded and, therefore, they have a great influence on the result of the segmentation.

2.2.2. *Modified pyramid linking approach (MPLA)*

Ziliani and Jensen [38] modified the classic pyramid linking approach to avoid the generation of disjoint segmented regions and the selection of a working level. The goal of the modified pyramidal linking approach (MPLA) was to achieve spatial consistency and to obtain a parameter-free algorithm. Another difference with previous works [39,40] is that the redefinition of the son–parent relationships is carried out consecutively between two levels before computing the initial values of the consecutive level. This modification increases the processing speed. In order to achieve spatial consistency, the vertex linking process is restricted to vertices that represent adjacent regions at the base level. To do that, Ziliani and Jensen [38] define a neighbourhood for each vertex. This neighbourhood specifies which vertices in the same level cover neighbouring areas in the base level. The algorithm presents a problem in the first iteration because it has not neighbourhood information yet. To avoid the selection of a working level, the algorithm uses two seeding rules. The island seeding rule assumes that a vertex that covers an entire region is surrounded by vertices with artificial features due to the overlapping of windows. Based on the neighbourhood information defined previously, the algorithm assumes a vertex to be an island if it has only one neighbour. In addition to this exclusion rule, the "parent-seeding" rule excludes all vertices for which an adjacent parent to link to cannot be found, because this indicates that no similar vertex is available.

Although the modified linking approach avoids the selection of the working level and achieves spatial consistency in the segmentation, it kept the other important problems of the linked pyramid: SV and non-adaptability to the image layout.

2.2.3. *Weighted linked pyramidal segmentation approaches*

Hong et al. [30] developed this type of regular pyramid. The structure is similar to the Linked Pyramid but each vertex retains all the edges with its parents, so every vertex has four parent-edges, one for each parent. Every edge carries a weight value that depends on the son–parent similarity. The value of a parent vertex is calculated as the average of its 16 sons, weighting each son value with its corresponding edge weight. Edge weights are recalculated at each iteration based on the new values of the vertices. Once the pyramid has converged, the final segmentation is achieved by using the edge with the highest weight for each vertex. The use of weights and the retention of all edges avoid forced choice in the edge modification process. This structure obtains slightly better results than the Linked one, as shown in Ref. [15]. However, it also shares its rigidity problems.

Depending on the type of weights used, there are two different kinds of Weighted Linked pyramids [15]: Fuzzy Linked Pyramid and Possibilistic Linked Pyramid. In the Fuzzy Linked Pyramid [30,41] weights are always positive and the parent edges of a vertex sum to one. In these algorithms, each vertex has only a parent edge with a value close to one after each iteration, while the rest of parent edges have a value close to zero. In these structures, it is fairly natural to use only the parent edge with the largest weight to define the preferred region for each vertex and thus perform the final segmentation of the image.

The Possibilistic Linked Pyramid [26] uses a non-normalized set of weights. Thus, some vertices link strongly with multiple parents, some link moderately with multiple parents, while some link only feebly to all their parents. Hong and Rosenfeld [26] continue to use the parent edges with the largest weight to define the final segmentation. However, this structure can be used to perform a different type of segmentation: the soft segmentation [15]. In contrast with classical segmentation, called crisp segmentation, in the soft segmentation each pixel can belong to more than one region. This segmentation avoids mistakes in region boundaries. Pixels near region boundaries usually are intermediate in value between the regions, and they can be placed in either of them during the crisp segmentation. In the soft segmentation these pixels belong to both regions. Prewer and Kitchen [15] perform the soft segmentation looking at the tree of possibilistic edge weights as a fuzzy decision tree. They determine a membership value for each of the vertices below the root by using a minimax approach, where each path to the root vertex has assigned the minimum value of the weights on that path, and each vertex takes the maximum value of its paths to the root as its degree of membership of that root. After this, each of the base level vertices has a membership value for each of the regions to which it links.

Finally, the Principal Component Pyramid (PCA pyramid) proposed by Bischof et al. [45,46] must also be mentioned here. In the PCA Pyramid, principal components analysis is used within a regular pyramid to obtain the reduction function used to compute the value of each parent from the set of values of its sons. Thus, known learning algorithms can be translated from neural networks to the pyramidal framework [47]. The PCA-pyramid has the same parent–son relationships as a regular pyramid and all the possible sons of each parent can be weighted to obtain the corresponding parent value.

3. Irregular pyramids

Irregular pyramids were introduced in order to solve the problems of the regular pyramids derived from their lack of flexibility. In contrast to regular pyramids, irregular ones have variable data structures and decimation processes which dynamically adapt to the image layout. Thus, the reduction factor between adjacent levels is not fixed; the size of each level and the height of the structure are unknown. Consequently, the well-defined and easy to compute neighbourhood relationships among nodes of the regular structures are lost in the irregular ones.

Initial attempts to develop adaptive hierarchical structures were done in the eighties (i.e. custom-made pyramids [42] and Voronoi tesselation-based approaches [43,44]). The first irregular pyramid to be applied in image analysis was proposed by Montanvert et al. [48]. They employed a stochastic decimation algorithm [27] to construct irregular tessellations and generate a hierarchical representation of the input image. This representation was built bottom-up and adapted to the content of the input image.

Irregular pyramids allow coarse-to-fine strategies by encoding a hierarchy of successively reduced graphs. Level l is represented by a graph $G_l = (V_l, E_l)$ consisting of vertices $v \in V_l$ and edges $e \in E_l$. In this hierarchy, each graph G_{l+1} is built from G_l by selecting a subset of V_l. The selected vertices are called surviving vertices. Non-surviving vertices of V_l are linked to surviving ones. Thus, each vertex v of G_{l+1} has associated a set of vertices of G_l, the reduction window of v, which includes itself and all non-surviving vertices linked to it [48]. This is a decimation process which requires rules for:

- The selection of the vertices V_{l+1} among V_l. These vertices are the surviving vertices of the decimation process.
- The allocation of each non-surviving vertex of level l to a survivor, which generates the son–parent edges.
- The creation of edges E_{l+1} by defining the adjacency relationships among the surviving vertices of level l.

The receptive field of one surviving vertex is defined by the transitive closure of the parent–son relationship and must be a connected set of vertices in the base level. Rules for the definition of the set of surviving vertices and the set of edges connecting each non-surviving vertex to its parent vary according to the considered decimation algorithm used within the irregular pyramid [49]. Therefore, the reduction procedure used to build one graph from the one below strongly influences the efficiency of the pyramid. On the other hand, each level of the hierarchy is encoded by a graph and, since many graph algorithms suffer from a high computational complexity, the efficiency of the irregular pyramid is also influenced by the selected graph encoding. Next subsections present different graph encodings and decimation algorithms used within the irregular pyramid framework.

3.1. Irregular pyramid data structures

Irregular pyramid data structures can be classified as:

- *Simple Graphs* [48]: This is the simplest data structure where the pyramid is defined as a stack of successively reduced simple graphs. This type of structures have two main drawbacks for image processing tasks: (i) they do not permit to know if two adjacent receptive fields have one or more common boundaries, and (ii) they do not allow to differentiate an adjacency relationship between two receptive fields from an inclusion relationship.
- *Dual Graphs* [10]: This structure solves the drawbacks of the simple graph approach representing each level of the pyramid as a dual pair of graphs and computing contraction and removal operations within them. The problem of this structure is the high increase of memory requirements and execution times since two data structures need to be stored and processed. In the results section of this paper, the computational time necessary to segment an image using this pyramid is compared with the time of other structures.
- *Combinatorial Maps* [50]: The combinatorial map is an efficient implementation of the dual graph approach which solves its aforementioned drawbacks. To do that, the combinatorial map approach uses a planar graph to represent each level of the pyramid, which encodes explicitly the orientation of edges around the vertices instead of a pair of dual graphs. In this planar graph it is possible to perform the contraction and removal operations using a set of permutations within the graph. In the results section of this paper, the reduction of computational time of this structure compared with the dual graph one is shown.

These irregular pyramid data structures are explained in detail in the following sections.

3.1.1. Simple graph

A simple graph is a non-weighted and undirected graph containing no self-loops. In this hierarchy, a pyramidal level l is defined by a graph $G_l = (V_l, E_l)$, where the set of vertices V_l represents a partition of the image into connected subsets of pixels. The graph edges E_l represent adjacency relationships among pyramidal vertices of the level l. Two vertices are connected if there exists a connecting path in level $l-1$ that joins them. A path in G_{l-1} is a connecting path of two surviving vertices $v, v' \in V_l$ if it satisfies one of the following conditions [49]:

- v and v' are connected by an edge $e \in E_{l-1}$.
- v and v' are connected by a path (e_1, v_i, e_2), where v_i is a non-surviving vertex connected to v and v' by edges $e_1 \in E_{l-1}$ and $e_2 \in E_{l-1}$, respectively.
- v and v' are connected by a path $(e_1, v_i, e_i, v_j, e_2)$, where v_i and v_j are two non-surviving vertices connected to v and v', respectively.

Simple graphs encode the adjacency between two vertices by only one edge, although their receptive fields may share several boundary segments. Therefore, a graph edge may thus encode a non-connected set of boundaries between the

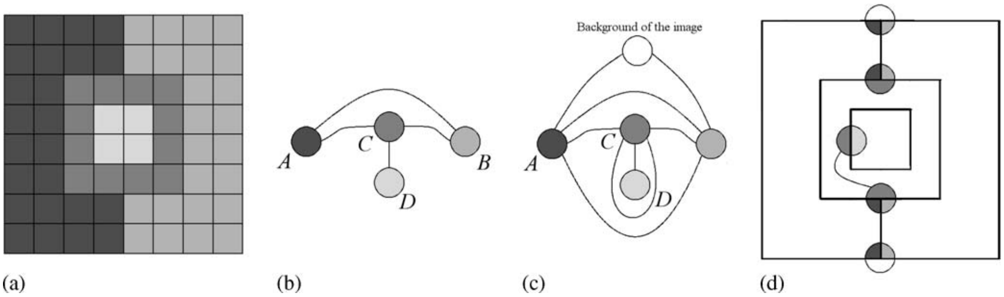

Fig. 3. Codification of connected components by several irregular pyramid data structures: (a) 8 × 8 image layout; (b) encoding by a simple graph pyramid; and (c,d) encoding by a dual graph or combinatorial pyramids (reprinted from Ref. [23], with permission from Elsevier).

associated receptive fields. Moreover, the lack of self-loops in simple graphs does not allow to differentiate an adjacency relationship between two receptive fields from an inclusion relationship. These facts are shown in Fig. 3b, which represents the top of a simple graph pyramid encoding the connected components of Fig. 3a.

3.1.2. Dual graph

In a dual graph pyramid, a level consists of a dual pair $(G_l, \bar{G}_l)$ of planar graphs G_l and $\bar{G}_l$. If level l defines a partition of the image into a connected subsets of pixels, then the vertices of G_l are the representatives of these subsets and the edges of G_l represent their neighbourhood relationships. The edges of $\bar{G}_l$ represent the boundaries of these connected subsets in level l and the vertices of $\bar{G}_l$ define meeting points of boundary segments of $\bar{G}_l$. Fig. 3c represents the top of a dual graph pyramid encoding the connected components of Fig. 3a. Fig. 3d shows the dual graph corresponding to 3c.

Within the dual graph pyramid framework, the set of edges that define the adjacency relationships among pyramidal vertices of the level $l+1$ is generated in two steps. First, the set of edges that connects each non-surviving vertex to its parent is contracted using a contraction kernel. A contraction kernel of a level l is the set of surviving vertices of l and the edges that connect each non-surviving vertex with its parent. The edge contraction operation collapses two adjacent vertices into one vertex, removing the edge between them. This operation may create redundant edges such as empty self-loops or double edges. The removal of these redundant edges constitutes the second step of the creation of the set of edges E_{l+1}. These redundant edges are characterized in the dual of the graph and removed by a set of edge removal kernels [51]. The key idea of the dual graphs is that a contraction in a graph implies a removal in its dual, and vice versa, in order to maintain the duality between the newly generated graphs. Thus, the generation of the edges in level $l+1$ can be resumed as follows:

(1) Contraction of edges in G_l which connect non-surviving vertices with their parents. Removal of their corresponding edges in $\bar{G}_l$. Fig. 4b shows the reduction performed by the contraction kernel in Fig. 4a.
(2) Contraction of redundant edges in $\bar{G}_l$ and removal of their corresponding edges in G_l. In Fig. 4c, the dual vertex a has a face defined by vertices **A** and **B**. The boundary between the regions defined by these vertices is artificially split by this dual vertex. Then, the two dual edges incident to this dual vertex (e'_1 and e'_2) can be contracted. The contraction of these dual edges has to be followed by the removal of one associated edge (e_1 or e_2) in order to maintain the duality between both graphs. In the same way, the dual vertex b encodes an adjacency relationship between two vertices contracted in the same vertex. This relationship can be removed by eliminating this direct self-loop and contracting the associated dual edge.

Using such a reduction scheme each edge in the reduced graph corresponds to one boundary between two regions. Moreover, inclusion relationships may be differentiated from adjacency ones in the dual graph.

3.1.3. Combinatorial map

A combinatorial map may be defined as a planar graph encoding explicitly the orientation of edges around a given vertex [23]. Fig. 5 illustrates the derivation of a combinatorial map from a plane graph. Firstly, edges are split where their dual edges cross (see Fig. 5b). These half-edges are called darts and have the origin at the vertex they are attached to (Fig. 5c). A combinatorial pyramid is defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations [8].

A combinatorial map can be expressed as $G = (\mathscr{D}, \sigma, \alpha)$, where $\mathscr{D}$ is the set of darts and σ and α are two operations defined on $\mathscr{D}$. α allows to know which two darts stem from the same edge and is called "reverse permutation". σ is used to compute which darts are around a given vertex and it is named "successor permutation". Another important operation φ is defined over the combinatorial map which allows to know which darts are around a given face of G. This operation is the same permutation as σ but calculated in the dual graph $\bar{G}$. The advantage of this representation of graphs using α, σ and φ is that φ can be also computed over G as a

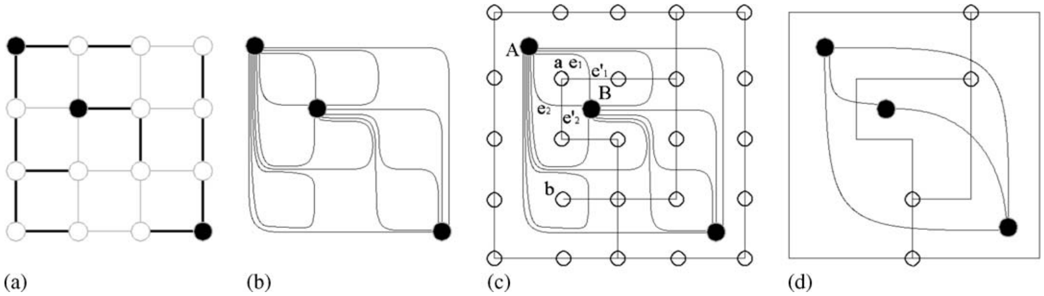

Fig. 4. Contraction and removal kernels: (a) contraction kernel composed of three vertices (surviving vertices are marked in black); (b) reduction performed by the equivalent contraction kernel in (a); (c) redundant edges characterisation; and (d) dual graph pair $(G, \bar{G})$ after dual decimation step.

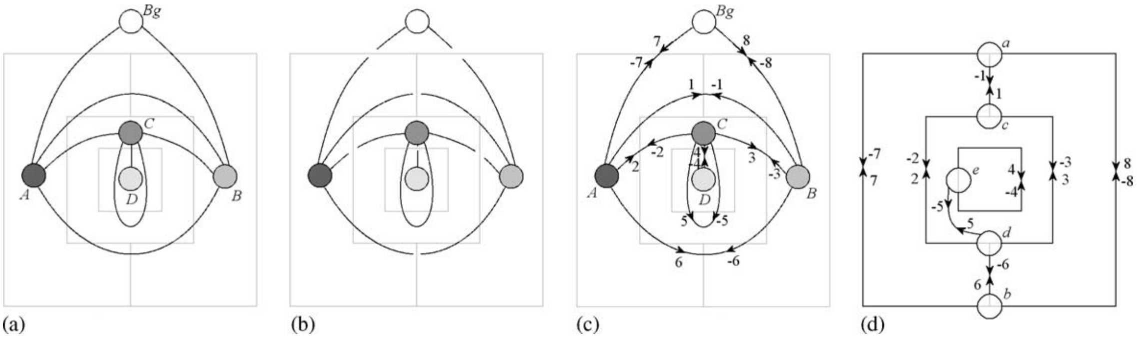

Fig. 5. Combinatorial map: (a) a plane graph; (b) edges splitting; (c) combinatorial map G; and (d) dual map of G.

combination of α and σ: $\varphi = \sigma \circ \alpha$. Thus, the dual graph is implicitly encoded in the combinatorial map G.

The advantage of the combinatorial map based pyramid representation is that the contraction and removal operations can be performed knowing only the combinatorial map G and the permutation operations α, σ and φ. $\bar{G}$ is not needed. So, the advantages of the dual graph representation are kept without a high increase of the computational cost.

3.2. Irregular pyramid decimation schemes

Although original irregular pyramids overcome the drawbacks of regular ones, their main drawback is that they only grow to a reasonable height as long as the base level is small. If the base level size gets larger, the reduction factor cannot be bound because the progressive deviation from the regular base favours configurations that slow down the contraction process [52]. This height increasing degrades the efficiency of irregular pyramids. Recent work has resolved this problem by new selection mechanisms which guarantee logarithmic heights [53]. Next subsections deal with different reduction schemes used to build the irregular pyramid. These schemes determine the height of the pyramid and the properties that arise from the decimation process.

3.2.1. Stochastic decimation process

If $G_l = (V_l, E_l)$ represents the level l of the hierarchy, where V_l defines the set of vertices of the graph and E_l the set of edges, the stochastic decimation process introduced by Meer [27] imposes two constraints on the set of surviving vertices, V_{l+1}:

(1) Any non-surviving vertex v of level l has at least one surviving vertex in its neighbourhood, v'.
(2) Two neighbour vertices v and v' at level l cannot both survive.

These rules define a maximal independent set (MIS). In order to build this MIS the decimation algorithm uses three variables for each vertex v_i: two binary-state variables p_i and q_i, and a random variable x_i uniformly distributed between [0,1]. The surviving vertices are chosen by an iterative local process. A vertex v_i in V_l survives if, at the end of the algorithm—iteration k—, its $p_i(k)$ state value is true. In the first iteration:

- $p_i^{l+1}(1)$ of a vertex v_i is set to 1 (true) if its x_i value is the maximum x value in its neighbourhood (local maximum). It must be noted that the local maximum nodes are selected as surviving vertices in the first iteration.
- $q_i^{l+1}(1)$ is set to 1 if v_i is not a local maximum and there is not a local maximum (node with $p^{l+1}(1) = 1$) in its vicinity.

In the rest of iterations the nodes with $q_i^{l+1}(n-1) = 1$ are studied. Thus, a node with $q_i^{l+1}(n-1) = 1$ is set to

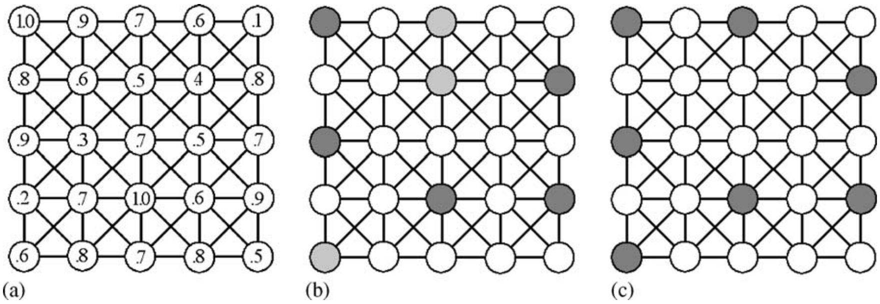

Fig. 6. Stochastic decimation procedure: (a) 8-connected valuated graph; (b) extraction of local maxima (dark grey vertices) and their neighbours (white vertices); and (c) complete specification of the set of surviving vertices (grey vertices).

$p_i^{l+1}(n) = 1$ and $q_i^{l+1}(n) = 0$ if it is the local maximum among its neighbours with $q_i^{l+1}(n-1) = 1$. This process is iterated until $q_i^{l+1}(n)$ is false for all vertex v_i. The set of sons are defined in G_l only after the vertices of G_{l+1} (their parents) have been chosen.

In Fig. 6 the stochastic decimation process is shown. In Fig. 6a the x value of each vertex is represented. The first iteration of the stochastic decimation procedure is shown in Fig. 6b. The dark vertices are the vertices with $p(2)=1$ (surviving vertices) and the white vertices are their neighbours. Grey vertices are the vertices with $q(2) = 1$. In Fig. 6c the second iteration is presented. This iteration is the last one in this case because all the vertices have a surviving vertex in their neighbourhood.

3.2.2. Connectivity preserving relinking approach

Nacken [54] describes a decimation process that adapts the classical relinking rules proposed by Burt et al. [29] for an irregular data structure. In order to create V_{l+1}, the set of vertices of the lower level in the hierarchy, V_l, is partitioned into a number of connected reduction windows. Reduction windows are computed by applying the following iterative process:

(1) Every vertex v_i which does not belong to any reduction window is given a label γ_i.
(2) Every vertex whose label is larger than that of all of its neighbours is selected as a surviving vertex (centre of a new reduction window).
(3) For each newly selected surviving vertex v, a maximal subset of the neighbours of v, containing no dissimilar pairs, is added to complete the reduction window. Dissimilarity of adjacent vertices must be defined using an edge strength measure.

The label γ_i can be a random number, although some image-dependent value can also be employed. The difference with the stochastic decimation procedure is in the order of the steps. In stochastic decimation, the computation of a MIS by repeated selection of local maxima is completed before the reduction windows are computed by assignments of neighbours; in this approach, a number of reduction windows are computed in each selection of local maxima.

The parent–son edges created in this step have the same role as the regular structure in the classical relinking scheme [29]: they serve as an initial configuration which is adapted by relinking. Then, the algorithm performs an iterative relinking process that preserves the connectivity. This process is applied vertex by vertex. For each vertex v, a set of allowed candidate parents is computed, depending on the actual structure of the hierarchy. This set plays the same role as the fixed set of candidate parents in the classical relinking scheme, with the particularity that linking v to any of this allowed parent assures that connectivity is preserved [54]. Then, a new parent is chosen from the set of allowed candidate parents. The vertex is relinked to the new parent and the graph structure and attributes of vertices are updated accordingly. This process is repeated until a stable configuration is reached. When the relinking process finishes, the next level of the hierarchy can be built.

3.2.3. Dual graph contraction

In Ref. [49], the building of irregular pyramids by dual-graph contraction is described. In this work, a contraction kernel is defined on a graph $G_l=(V_l, E_l)$ by a set of surviving vertices V_{l+1} and a set of non-surviving edges N_l such that:

- (V_l, N_l) is a spanning forest of G_l. A spanning forest of G_l is a subgraph that contains all the vertices of G_l and that contains no cycles. Fig. 4a shows a spanning forest of a graph.
- Each tree of (V_l, N_l) is rooted by a vertex of V_{l+1}.

Therefore, the decimation of a graph by contraction kernels differs from the stochastic decimation process in that two surviving vertices may be adjacent in the contracted graph. Also a non-surviving vertex may be connected to its parent by a branch of a tree.

3.2.4. Specified rate and prioritized sampling approaches

In the stochastic pyramid framework, the ratio between the number of surviving vertices and the total number of vertices (the sampling rate, which is the inverse of the reduction factor) may be different in different parts of the graph. This is because different parts of the graph consist of vertices having different numbers of neighbours. Hence, parts of the graph where vertices have a smaller number of neighbours on average can accommodate more surviving vertices than other parts of the graph. The specified rate sampling approach [55] replaces the iterative decimation process of the stochastic approach by a single step process. The decimation algorithm uses two variables for each vertex v_i: a binary-state variable p_i and a random variable x_i uniformly distributed between [0,1]. Initially, the state variable p_i of all vertices is set to 0. Then,

$$p_i^{l+1} \quad \Leftrightarrow \quad x_i < \omega. \tag{1}$$

That is, a vertex is selected as a surviving vertex based on a fixed probability. The variable ω determines the sampling rate and can be specified by the user. The constraints imposed by the stochastic decimation process [27] may be violated. Thus, any non-surviving vertex can have no surviving vertex in its neighbourhood, and two neighbouring vertices can be selected as surviving vertices.

Another way to increase the reduction factor is to allow some vertices to have a higher priority over others in being selected as surviving vertices. If these prioritized vertices have larger numbers of neighbours (the number of neighbours of a node is called "degree" of the node), a larger number of vertices will become non-surviving vertices. Then, the number of selected surviving vertices is reduced. To introduce priority in the decimation scheme, a ranking approach is proposed by Ip and Lam [55]. The range [0,1] is divided into n sub-ranges. The value n is an estimated constant denoting the maximum degree of a vertex. A random variable x' is generated in the range $[0, 1/n]$. The random variable x associated with a vertex is then set by

$$x = x' + (r-1)/n \quad \text{if } r < n, \tag{2}$$
$$x = x' + (n-1)/n \quad \text{if } r \geqslant n, \tag{3}$$

where r is the degree of the node. Thus, for any two neighbouring vertices with different degree, the one which has a larger degree will be usually assigned a higher priority in being chosen as a surviving vertex.

3.2.5. Data driven decimation scheme

One of the disadvantages of the stochastic decimation process is that vertices extracted as local maxima in the first iteration must wait until the graph is complete in successive iterations [56]. These iterations are used only to complete the MIS. In the data driven decimation process (D3P), a vertex v_i of G_l survives if and only if it is a local maximum (p_i^{l+1} = true) or does not have yet any surviving vertex in its neighbourhood (q_i^{l+1} = true). Therefore, it is assumed that being a local maximum is of importance and no correction is performed in subsequent iterations. In areas where there is no real maxima, the process still tries to extract submaxima but without slowing down the decimation procedure in other areas of the graph. The procedure is not iteratively run.

The graph G_{l+1} defined by the D3P is slightly different to the one defined by the stochastic pyramid because two neighbours in V_l can both survive in V_{l+1}. Thus, D3P achieves faster convergence and better fits the distribution of the values associated with the vertices of the initial graph [56]. However, as for the stochastic decimation process, the D3P cannot guarantee a constant reduction factor between pyramid levels [52].

3.2.6. MIES and MIDES algorithms

Although stochastic pyramids overcome the drawbacks of regular ones, they grow higher than the base diameter for large input images. As a consequence of the greater height, the efficiency of pyramids degrades. This problem has been resolved in dual graph pyramids by selection mechanisms which guarantee logarithmic heights by replacing the selection method proposed in Ref. [27] by two new iteratively local methods: maximal independent edge set algorithm (MIES) [52] and maximal independent directed edge set (MIDES) [11].

The MIES algorithm has been developed to be applied in the dual graph framework. Its goal is to find a set of contraction kernels in a plane graph G_l such that each vertex of G_l is contained in exactly one contraction kernel, and each contraction kernel contains at least two vertices. Thus, the number of vertices between consecutive graph levels is reduced to half or less and a reduction factor of at least 2 can be guaranteed. The MIES algorithm consists of three steps [52,53]:

(1) Find a maximal matching M from G_l. A matching is a set of edges in which no pair of edges has a common end vertex (Fig. 7a).
(2) Enlarge M to forest $M+$ by connecting isolated vertices of G_l to the maximal matching M (Fig. 7b).
(3) $M+$ is reduced by breaking up trees of diameter three into trees of depth one. A tree is a set of edges connected at their ends containing no closed loops (cycles) (Fig. 7c).

A maximal matching of G_l is equivalent to a maximal independent vertex set on the edge graph [52]. Therefore, the maximal matching can be obtained by applying the MIS algorithm in $\bar{G}_l$. The second and third steps of the MIES algorithm permit to obtain a set of contraction kernels where each vertex belongs to a tree of depth one.

The MIES algorithm can be used either in a dual graph framework or for connected component analysis [57]. However, its main disadvantage is that it is only applicable where there are no constraints on direction of contraction [11,53]. As it is shown in Fig. 7d, there are certain contraction kernels

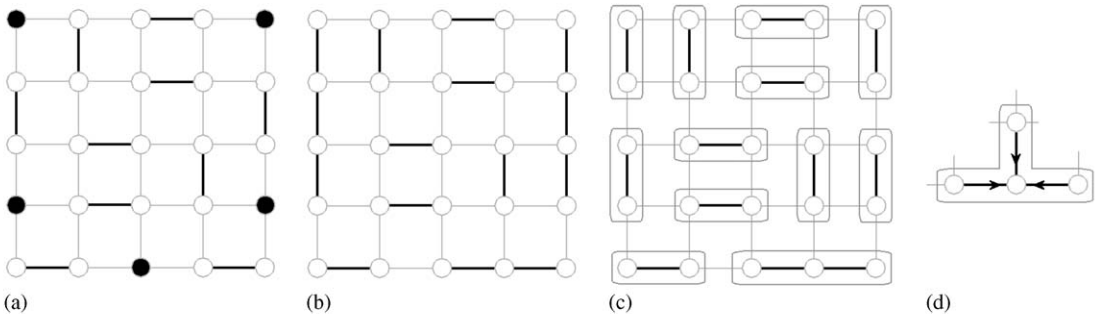

Fig. 7. MIES algorithm: (a) maximal matching M (isolated vertices are black coloured); (b) enlarged matching $M+$; (c) reduced matching $M+$ and contraction kernels; and (d) restriction to choose the surviving vertex and direction of contraction of a contraction kernel.

that impose the only possible surviving vertex and, therefore, the direction of contraction.

MIDES algorithm can be applied in oriented graphs, such as the graph applied to line image analysis [58]. In an oriented graph the relations between pairs of vertices are not symmetric, so that each edge has a directional character. Besides, this edge direction is unique (i.e., edges cannot be bi-directed). In these graphs, an edge e with source s_e and target t_e, $e=(s_e, t_e)$, must be contracted from s_e to t_e, only if the attributes of the edge e and of its source and target vertices fulfil a certain rule. The set of edges that fulfils the rule are called pre-selected edges [11]. Only these pre-selected edges are considered as candidates for contraction and the goal is to build contraction kernels with a high reduction factor. In order to perform the contractions in parallel, a vertex disjoint union of contraction kernels is needed [53]. The MIDES algorithm defines such a union in terms of independent directed edges. Two directed edges are independent if they do not belong to the same neighbourhood. The neighbourhood of a directed edge e, N_e, is defined by all directed edges with the same source s_e, targeting the source s_e or emanating from t_e [53]. Then, the contraction kernels can be found as in MIS, but dealing with edges instead of vertices. This algorithm shows better reduction factor than MIS or MIES [11,53].

3.2.7. Union-find techniques

The union-find algorithm was proposed by Tarjan [59] as a general method for keeping track of disjoint sets. Basically, it allows performing of set-union operations on sets which are in some way equivalent, while ensuring that the end product of such a union is disjoint from any other set.

Brun and Kropatsch [8] propose to use the union-find algorithm to design a contraction kernel in the combinatorial pyramid framework. Union-find algorithms use tree structures to represent sets. Each non-root vertex in the tree points to its parent, while the root is flagged in some way. Therefore, each tree of a contraction kernel is encoded by storing in each vertex a reference to its parent. Initially, the parent of each vertex v is itself. Then, the union-find algorithm performs the following operations over any dart d in $\mathcal{D}_l$:

- A find operation is applied on the origin vertices of $\alpha(d)$ defined over the same edge. These operations return the roots, r_d and r_{-d}, of the trees containing these two vertices.
- If r_d and r_{-d} are different and they must be merged, a union operation merges the corresponding two trees into one. This union is performed by setting one of the roots to be the parent of the other root. The edge which contain d and $-d$ is included in the contraction kernel.

The union-find algorithm has proven to be very efficient, especially when it is run on sequential machines.

3.3. Irregular pyramid-based segmentation approaches

In this work, we propose to evaluate and compare the efficiency and performance of several pyramid-based segmentation approaches. To do that, we have firstly presented several data encodings and decimation processes. All decimation processes described in Section 3.2 have been applied to segmentation purposes. In this section, several segmentation approaches are reviewed.

3.3.1. Segmentation with a hierarchy of region adjacency graphs (RAG) and the adaptive pyramid

The simple graph hierarchy and the stochastic decimation procedure supposed a great novelty for hierarchical processing. These tools permitted a hierarchical data structure to adapt itself to the image layout, since the proposed hierarchy was not restricted to a rigid sampling structure. The stochastic decimation procedure was successfully applied to multiscale smoothing of chain-coded curves [60] and segmentation of grey level images [48]. In this last case, a hierarchy of region adjacency graphs (RAG) is generated. The RAG hierarchy performs the stochastic decimation within classes. These classes or similarity subgraphs must be generated before graph contraction is made and they are derived from the RAG by local decisions. Thus, contrary to the original stochastic decimation idea, the resulting decimation procedure is dependent on the image data.

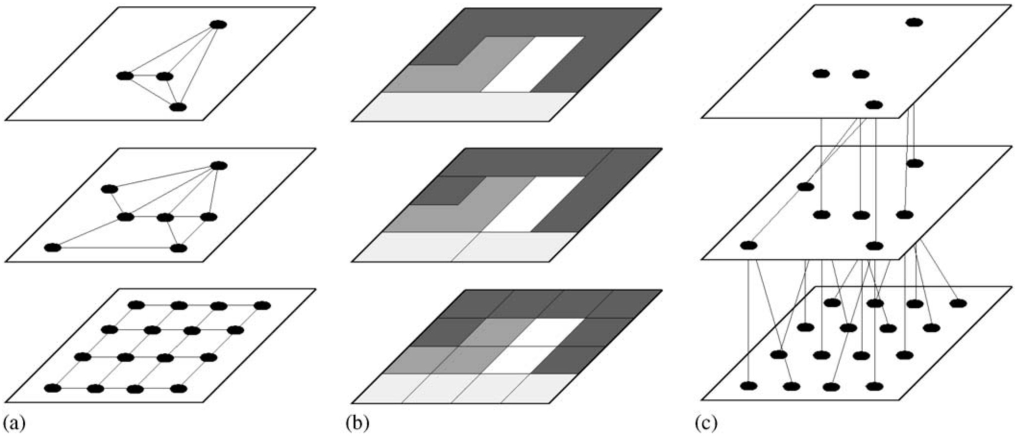

Fig. 8. Top-down segmentation based on the RAG hierarchy: (a) region adjacency graphs; (b) receptive fields pyramid; and (c) corresponding tree structure.

The algorithm works as follows:

(1) Graph G_0 is defined by the 8-connected square sampling grid on the level 0, where each vertex is a pixel of the original image.
(2) Classes at level l are defined. To do that, each vertex v_i at level l has associated a value g_i characterizing its region of the image (e.g. average grey level). For each v_i, which does not belong to any class yet, every neighbour v_j is examined and a decision is made on whether or not it belongs to the same class of v_i. This decision is based on g_i and g_j values, which are compared using a similarity function.
(3) Surviving vertices of every class are chosen applying the stochastic decimation algorithm into the class.

In order to define the classes, several approaches have been experimentally proven [48]. The simplest approach is to define class membership by thresholding the grey level differences between a vertex and their neighbours. This symmetric class membership criterion does not achieve satisfactory results because it strongly influences the structure of the hierarchy and therefore the final segmentation of the image. To overcome this problem, a non-symmetric class membership criterion based on the maximum averaged contrast method was also proposed. Finally, in order to deal with the problem of root detection, a root measure can be defined [48]. Although the results presented in Ref. [48] show that the RAG hierarchy correctly reflects the structure of the image, the stochastic concept inherent to the method causes changes in the segmentation results when the algorithm is successively applied to the same input image.

Then, the main drawback of the stochastic decimation process is that different outcomes of the random variable produce different structures and segmentations. Thus, the segmentation of an image varies between executions with the same input parameters. Besides, the decimation process should be controlled in order to assure that there exists at least a root for each interest region of the original image. Jolion and Montanvert [7] propose to modify the decimation process in order to bias toward vertices with high information value. Instead of a random variable, the adaptive pyramid uses an interest variable in the decimation process: the grey level variance gv_i from the receptive field of a pyramid vertex v_i. After survivor extraction, non-surviving vertices are linked to the most similar surviving vertex of its neighbourhood. In the adaptive pyramid the use of the interest variable avoids the definition of classes before the decimation process.

Jolion and Montanvert [7] introduce a root extraction process into the algorithm. A vertex is the root of an original image region if it satisfies the following conditions: (i) a root vertex must be very different to the surviving vertices of its neighbourhood; and (ii) the size of a region defined by a root must be large enough to avoid local variations due to noise; a small region must compensate its low size with a high contrast with all its neighbours.

Using the classical RAG pyramid, Bertolino and Montanvert [61] propose to generate distinct segmentations of an image at different resolutions by using the tree structure represented by the graph hierarchy. Thus, a region of any level can be recursively split into subregions at the level below. Fig. 8 shows a graph hierarchy and its corresponding tree structure. Starting at the highest structure level, a homogeneity criterion is evaluated for each region: the standard deviation of every region is compared with a threshold σ_M to decide if the region must be split or not. Depending on σ_M the segmentation preserves more or less detail. Each region of the original image is extracted in the level where its representation is optimum. Since the standard deviation could be only suitable for certain kinds of images, other scale parameters may be used [61].

3.3.2. Segmentation with the localized pyramid

The graph pyramid is usually initialized with as many vertices as the number of pixels in the original input image.

In the localized pyramid [12], only a subset of the image pixels are segmented (undefined zones), while the rest of image pixels is associated to one or several vertices called roots. During the segmentation process, each pixel of the undefined zones is a vertex which can be merged with other vertices associated to undefined zones or with a root. To initialize the local pyramid and to determine which pixels of the original image belong to the undefined zones and which not, a local homogeneity analysis can be performed. Thus, Huart and Bertolino [12] propose to compute a homogeneity image (H-image) from the CIE L*a*b* colour space. This H-image is a grey-scale image whose pixel values are the H-values representing the image discontinuities according to a homogeneous feature. Low values correspond to homogeneous regions (roots) and high values correspond to possible discontinuities (undefined zones). The pixels of the undefined zones are segmented using a simple graph data structure and a modified stochastic decimation process. This pyramid allows that, during the segmentation process, the pixels of the undefined zones merge together and/or with a neighbouring root according to a similarity criterion.

When the segmentation has been locally performed, segmented regions are grouped using Gestalt criteria (perception-based image segmentation). In this region grouping process the local pyramid is extended with additional levels. The final result is a stack of partitions with very few objects [12].

3.3.3. Consensus image segmentation

Cho and Meer [5] propose a new approach for unsupervised segmentation based on RAG. This approach is derived from the consensus of a set of different segmentation outputs on one input image. The probabilistic component of the RAG pyramid-based segmentation implies that each time the algorithm is run the obtained result is slightly different. Differences are more important in the neighbourhoods where the piecewise constancy is less valid. In order to extract this information, local homogeneity is determined by collecting statistics for every pair of adjacency pixels, instead of statistics characterizing the spatial structure of the local neighbourhood of a pixel. The proposed segmentation algorithm works as follows:

- Given the input image, N different segmentations are obtained by exploiting the probabilistic component of the hierarchical RAG pyramid-based technique [48]. An example of the variation in the structure of hierarchy is illustrated in Fig. 9. Figs. 9a and d show that different surviving vertices were obtained because of the different random number assignation.
- The N segmented images are registered on the 8-connected mesh of the input image. Therefore, every pixel has N values associated. For every adjacent pixel pair a co-occurrence probability, i.e. the probability of belonging to the same delineated region, is derived. The set of all co-occurrence probabilities defines the co-occurrence field of the input image studied under the homogeneity criterion which defines the class distribution.
- Since the co-occurrence probabilities are derived from the initial image segmentations, they capture global information about the image at the local (pixel pair) level. The final segmentation of the input image is obtained by processing the co-occurrence probability field with a weighted RAG pyramid technique. This new graph is needed because each edge of the 8-connected mesh of the co-occurrence probability field has now a co-occurrence probability associated to it. Then, pixel pairs with high co-occurrence probability are grouped together based on the consensus about local homogeneity.

3.3.4. Image segmentation by connectivity preserving relinking

Segmentation by relinking [29] is performed by iteratively updating the class membership of pyramid vertices, i.e. by adapting parent–son edges. This technique, originally proposed for regular pyramids (see Section 2.2.1), presented serious drawbacks, the main of which is that classes represented by a vertex need not correspond to connected regions. Nacken [54] modifies the original relinking procedure and applies it to a RAG pyramid. The decimation algorithm proposed by Nacken [54] is briefly described in Section 3.2.2. In this section, we deal with its application to segmentation purposes.

In this pyramid, to create the vertices of level $l+1$, the vertices of level l are partitioned in a number of connected regions, as explained in Section 3.2.2. In order to apply this scheme to segmentation purposes [54], used as γ_i-value of a vertex v_i the area of the receptive field of this vertex. He defined two dissimilarity measures S_1 and S_2 between nodes:

$$S_1(v,w)=|g(v)-g(w)|-\tfrac{1}{2}(\sigma(v)+\sigma(w)), \tag{4}$$

$$S_2(v,w)=\frac{|g(v)-g(w)|}{1+\frac{1}{2}(\sigma(v)+\sigma(w))}, \tag{5}$$

being $g(v)$ the average grey value within the receptive field of a vertex v and $\sigma(v)$ the standard deviation of the grey value.

Once the vertices of G_{l+1} have been selected from G_l, and the son–parent edges between each non-surviving vertex and the survivor in its vicinity have been established, the connectivity-preserving relinking procedure is performed. For each vertex of level l in turn, a new parent is chosen from a set of candidate parents which preserve structure connectivity. The selected new parent could be the parent which minimizes the grey level difference. Another way to select the most suitable parent is to minimize the following energy function in each level:

$$E_{region}[l]=\sum_{v\in V_l} n(v)[g(v)-g(\pi(v))]^2, \tag{6}$$

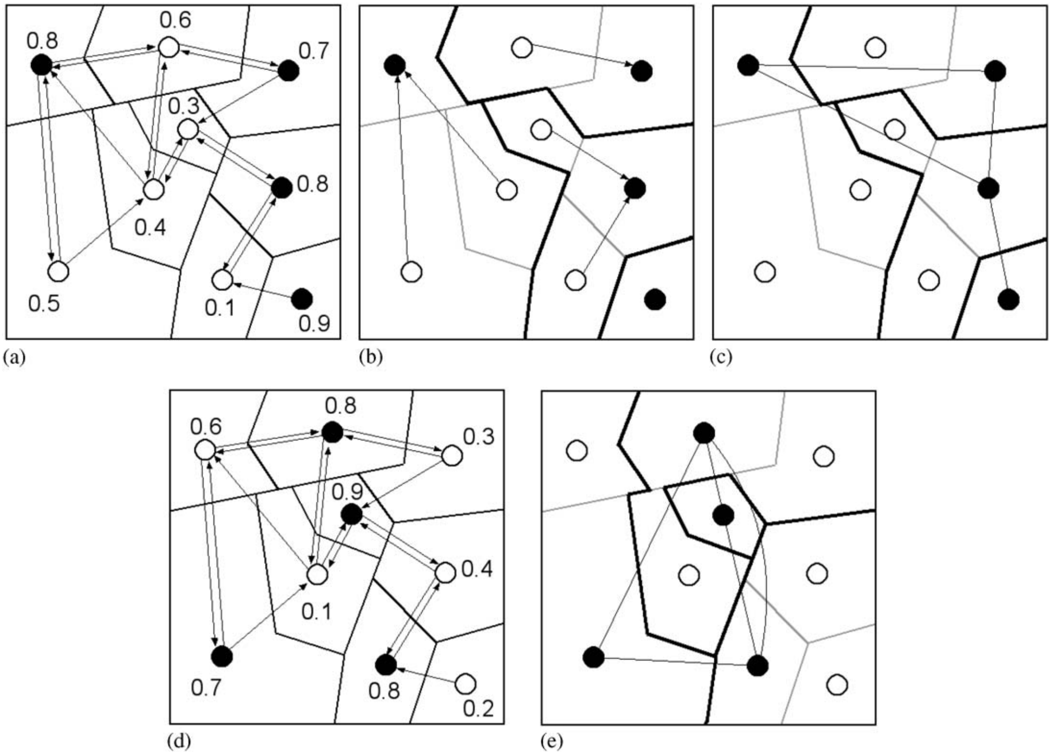

Fig. 9. Influence of the random component on the structure of the RAG pyramid: (a) The RAG at level l. Arrows show the decomposition of RAG into classes using a non-symmetric class membership. Surviving vertices into each class are marked in black; (b) non-surviving vertices allocation; (c) the RAG at level $l+1$ from (a,b); (d) the RAG at level l with different random numbers assigned to the vertices; and (e) the RAG at level $l+1$ from (d).

V_l being the set of vertices at level l, $n(v)$ the receptive field of v, $g(v)$ its grey level value and $\pi(v)$ its parent.

Finally, Nacken [54] proposes to combine region and boundary information in the segmentation process (edges of the RAG correspond to the boundaries between receptive fields in the input image). The proposed boundary-based relinking criterion is based on the minimization of the energy:

$$E_{boundary}[l] = \sum_{v \in V_{l-1}} \eta(R(v)), \tag{7}$$

where $\eta(R(v))$ is the average response of an edge detection filter along the receptive field of a vertex v. New edge strength measures are defined based on boundary information. The combination of boundary and region information is then performed in a combined edge strength measure, which takes into account the previously defined measures.

3.3.5. Region growing stopping based on spatial autocorrelation

The application of the decimation process to a graph hierarchy to obtain segmentation requires defining a criterion to stop this reduction procedure when the best segmentation is obtained. In Ref. [13], a statistical test to control the region growing process is proposed. This test is applied to the adaptive pyramid [7]. Let $F_{l,l+1}$ be the decimation graph defined by

$$F_{l,l+1} = (V_l, E_{l,l+1}) \subset G_l, \tag{8}$$

where $E_{l,l+1}$ are the inter-level edges between the levels l and $l+1$.

Thus, an edge in the decimation graph stands for the merging of two regions. If it is assumed that G_l is the best segmentation, then any edge in $F_{l,l+1}$ must be inappropriate. Therefore, it can be seen as an edge of a random graph that does not correctly correlate the associated vertices [13]. On the contrary, if G_l does not define the best segmentation, it must have an edge with a significant correlation between the associated vertices. Then, spatial autocorrelation can be used to control the decimation procedure, e.g. the region growing. Therefore, Lallich et al. [13] uses one of the most popular indicators to measure global spatial autocorrelation: the Moran's test [62].

3.3.6. Segmentation with the bounded irregular pyramid

The bounded irregular pyramid (BIP) [14] is a hierarchical structure that merges characteristics from regular and irregular pyramids. Its data structure is a combination of the simplest regular and irregular structures: the $2 \times 2/4$ regular one and the simple graph irregular representation. The algorithm firstly tries to work in a regular way by generating, from level l, a $2 \times 2/4$ new level $l+1$. However, only

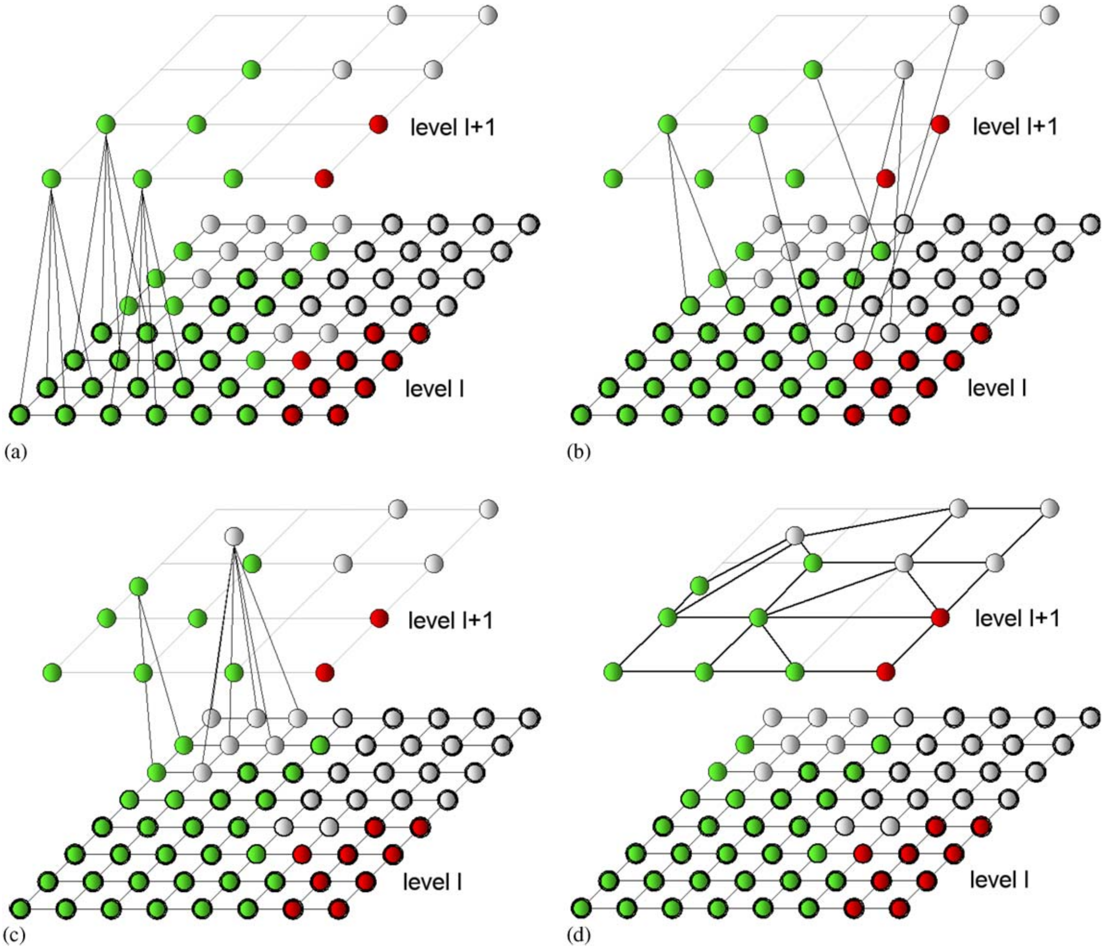

Fig. 10. BIP level generation: (a) regular step (non-orphan vertices of level l have been marked); (b) parent search; (c) intra-level twining at level l; and (d) intra-level edge definition at level $l+1$ (marked in black).

the 2×2 homogeneous arrays of V_l generate a new vertex of V_{l+1}. Therefore, this step creates an incomplete regular level $l+1$ which only presents vertices associated to homogeneous regions at the level below. Vertices of level l which generate a new vertex in V_{l+1} are linked to this vertex (son–parent edges). Fig. 10a shows this step (only the son–parent edges associated to three parents are presented).

Then, all vertices without parent (orphan vertices) of level l search for a neighbour vertex with a parent in level $l+1$ whose colour will be similar to the orphan vertex's colour (parent search step). If there are several candidate parents, the orphan vertex is linked to the most similar parent (Fig. 10b). Finally, the irregular part of the BIP is built. In this step, orphan vertices, of level l, search for all neighbour orphan vertices at the same level. Among the set of candidates, they are linked with the most similar. When two orphan vertices are twined, a new parent is generated at level $l+1$ (intra-level twining step). This parent is a node of the irregular part of the BIP (Fig. 10c). The algorithm performs these two steps simultaneously. Thus, if an orphan vertex does not find a parent in the parent search stage, it will search for an orphan neighbour to link to it (intra-level twining). In the parent search stage an orphan vertex can be linked with the irregular parent of a neighbour. Once this is completed, intra-level edges are generated at level $l+1$ (Fig. 10d). Fig. 11 shows the generation of a new level from level $l+1$ of Fig. 10d. The decimation process stops when it is no longer possible to generate new vertices in the regular part of the BIP. When all the levels are generated, homogeneous vertices without parent are regarded as roots and their corresponding receptive fields constitute the segmented image.

3.3.7. *Hierarchy of partitions by internal and external contrast measures*

The aim of the hierarchy of partitions defined by Haxhimusa and Kropatsch [1] is to build a minimum weight spanning tree (MST) of the input image [63]. This MST will allow to find the region borders in a bottom-up way and, thus, to perform the image segmentation. Although the used data structure is the dual graph and the employed decimation process is the MIES proposed by Haxhimusa et al. [52], in Ref. [1] the construction of the dual graph is formulated

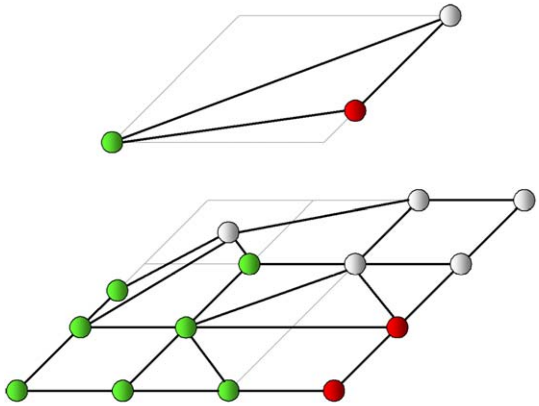

Fig. 11. BIP level generation.

as the building of a MST of the input image (level 0 of the graph hierarchy). Thus, an algorithm based on Borůvka's proposal [64] is used to build in a hierarchical way a MST preserving the image topology. The method is based on a previous work of Felzenszwalb and Huttenlocher [63].

This MST is built as follows:

In a hierarchy of graphs, where G_l defines the graph on level l of the hierarchy, every vertex u_i of G_l has a receptive field in the base level $CC(u_i)$. In each level l the union of the receptive fields of the vertices in the level defines a partition $P_l = \{CC(u_i)\}_{i=1\dots n}$. Then, to build the level $l+1$ from the level l, the goal is to find a partition P_{l+1} by merging members of P_l. Haxhimusa and Kropatsch [1] define the following pairwise merge criterion:

$$Comp(CC(u_i), CC(u_j)) \\ \begin{cases} 1 & \text{if } Ext(CC(u_i), CC(u_j)) \\ & \quad \leqslant PInt(CC(u_i), CC(u_j)), \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where $PInt(\cdot,\cdot)$ and $Ext(\cdot,\cdot)$ are the minimum internal contrast difference and the external contrast between two receptive fields, respectively [63]. $Ext(CC(u_i), CC(u_j))$ is the smallest dissimilarity between $CC(u_i)$ in P_l and $CC(u_j)$ in P_l. $PInt(\cdot,\cdot)$ is defined as

$$PInt(CC(u_i), CC(u_j)) = \min(Int(CC(u_i)) + \tau(CC(u_i)), \\ Int(CC(u_j)) + \tau(CC(u_j))). \tag{10}$$

$Int(CC(u_i))$ being the internal contrast of the $CC(u_i)$ in P_l. This contrast measure is defined as the largest dissimilarity of component $CC(u_i)$. The threshold function τ controls the degree to which the external variation can actually be larger than the internal variations and still have the receptive fields be considered similar [63].

This hierarchical partitioning algorithm has been applied to the combinatorial pyramid framework by Haxhimusa et al. [65]. Results show that the algorithm can handle large variations and gradient intensity in images.

3.3.8. *Segmentation based on combinatorial pyramids and union-find algorithm*

In Ref. [8], a segmentation application based on the combinatorial pyramid and the union-find-based decimation algorithm is proposed. The segmentation algorithm works on grey level images and can be briefly summarized as follows:

(1) The original image is quantized into K grey levels. Each vertex in level 0 of the graph hierarchy encodes a connected component of the pixels whose grey level values are mapped onto a same interval. The background of the image is determined by selecting the largest region adjacent to the exterior of the image.
(2) All regions included in the background whose size is less than a given threshold T are merged with the background (level 1).
(3) In order to perform the union-find process and to build a level $l+1$ from l the mean grey level of each vertex of l is used to initialize a grey-level histogram. The frequency $h(i)$ of one entry i of the histogram is set to the number of vertices whose mean grey level is equal to i. This histogram is then quantized into K values. The algorithm merges any couple of adjacent vertices whose mean grey values are mapped into the same interval. The vertices which are merged together generate a new vertex of the level $l+1$.

The last step is iterated until no more merge occurs. It must be noted that the quantization process only provides a partition of the range of grey values. The encoding of the partition and the merge operations are performed using the combinatorial pyramid model [8].

4. Evaluation of segmentation results

4.1. *Introduction*

Until a few years ago, most of the segmentation methods worked with grey level images due to the large amount of data necessary to process colour images. Nowadays, colour image segmentation approaches are arising, thanks to the increase of the computational capability. The evaluation of segmentation results described in this paper deals with colour images. It must be noted that “colour cue image segmentation in a bottom-up way cannot and should not produce complete final image decomposition into meaningful objects, but it can provide a hierarchical partitioning of the image into homogeneous coloured regions” [1].

There are two main types of evaluation methods to measure the quality of a given segmentation algorithm: qualitative methods and quantitative ones. Qualitative methods are based on the opinion of a human expert who decides the accuracy of the studied algorithm. Although this measure depends on the human intuition and could be different for distinct observers, it could be very useful to evaluate some characteristics of the algorithms. On the other hand, quantitative methods are based on numerical data. According to the previous work of Zhang [66], quantitative segmentation evaluation methods can be classified into two categories: analytical and empirical methods. Analytical methods directly examine and assess the segmentation algorithms by analyzing their principles and properties. Some properties to be evaluated are the processing strategy, the processing complexity and efficiency and the segmentation resolution. These properties could be useful for selecting suitable algorithms in particular applications. But usually, the segmentation results are used in more complex image processing or computer vision tasks, where the accuracy of the results is usually more important than the performance of the algorithm, which can be improved later. Hence, the empirical methods are to be preferred. These methods indirectly judge the segmentation algorithms by applying them to test images and measuring the quality of segmentation results. Empirical methods can be classified into two types: goodness methods and discrepancy methods. Goodness methods measure some desirable properties of segmented images by "goodness" parameters. These methods have the problem that these parameters depend on the human intuition. Discrepancy methods compute the ideal segmentation first and then the segmentation obtained with the algorithm is compared with the ideal one by counting their differences. These methods present the problem that having a previous ideal segmentation is necessary, which depends on the human intuition too.

4.2. *Selected evaluation measurements*

Two of the main drawbacks of the regular pyramids were qualitatively evaluated by Bister et al. [4]. Region connectivity is not preserved in regular pyramids because the structure does not take into account adjacency information when the pyramid is built. Figs. 12b and c represent the different classes resulting from the segmentation of the image in Fig. 12a. False colour has been used to distinguish each class from the rest. The linked pyramid divides up the background region into different classes (Fig. 12b). Besides, it fuses different regions into the same class, creating non-connected segmented regions. On the other hand, the bounded irregular pyramid correctly segment the original image into five classes: four rectangles and the background (Fig. 12c). The second drawback is related with the presence of elongated objects. The inflexibility of the structure of regular pyramids makes the adaptation of such a structure to this type of objects difficult. Fig. 12d includes a set of elongated objects presenting different aspect ratios. It is easy to note that the linked pyramid (Fig. 12e) cannot handle elongated shapes. Fig. 12f shows that the bounded irregular pyramid is capable to adapt its structure to correctly segment this type of objects.

Quantitative evaluation methods are also useful when subjective contribution is not desired or permitted. We have selected three empirical methods: the SV proposed by Prewer and Kitchen [15], the F function proposed in Ref. [16] and the Q function proposed in Ref. [17]. These methods can be regarded as goodness methods, but they do not require any user-set parameter for the evaluation of the performance of the segmentation. The F function takes into account the following guidelines:

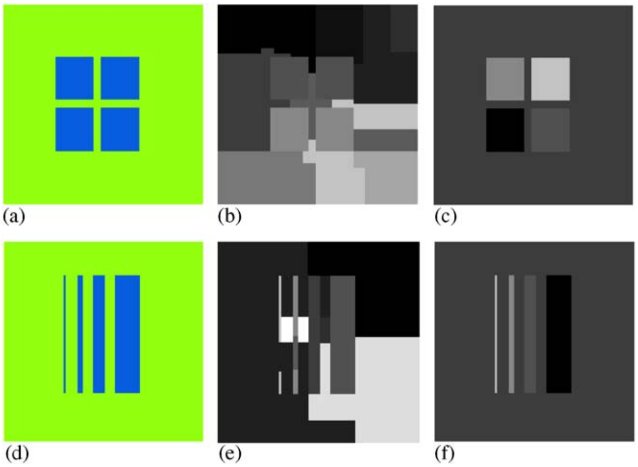

Fig. 12. Qualitative evaluation of regular pyramid drawbacks: (a) Input image #1; (b) linked pyramid segmentation result of (a); (c) BIP segmentation result of (a); (d) input image #2; (e) linked pyramid segmentation result of (d); and (f) BIP segmentation result of (d).

- Regions must be uniform and homogeneous.
- The interior of the regions must be simple, without too many small holes.
- Adjacent regions must present significantly different values for uniform characteristics.

The F function is computed as follows:

$$F(I) = \frac{1}{1000(N \cdot M)} \sqrt{R} \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}}, \tag{11}$$

I being the segmented image, $N \times M$ the image size and R the number of segmented regions. A_i and e_i are the area of the region i and its average colour error, respectively.

The Q function takes into account the same guidelines, but penalizes the existence of small regions in a more rigid way

$$Q(I) = \frac{1}{1000(N \cdot M)\sqrt{R}\sum_{i=1}^{R}\left[\frac{e_i^2}{1+\log A_i} + \left(\frac{R(A_i)}{A_i}\right)^2\right]}, \tag{12}$$

$R(A_i)$ being the number of segmented regions with area equal to A_i.

SV means that the segmentation produced by pyramidal segmentation algorithms varies when the base of the pyramid is shifted slightly. This is an undesirable effect for a segmentation method. Thus, the SV can be taken as a measurement of an algorithm quality. The previous methods compare an original image with its segmented image. In contrast, this method compares the segmentation of an image by a given algorithm with the segmentation produced by the same algorithm on slightly shifted versions of the same image. To do that, we have taken a 128 × 128 pixel window in the center of the original image. We have compared the segmentation of this subimage with each segmented image obtained by shifting the window a maximum shift of 11 pixels to the right and 11 pixels down. Thus, there is a total of 120 images to compare with the original one. In order to do each comparison between a segmented shifted image j and the segmented original one, the root mean square difference is calculated:

$$RMSD_j = \sqrt{\frac{\sum d_i^2}{128 \cdot 128}}, \quad SV = \frac{1}{120}\sum_{j=1}^{120} RMSD_j, \qquad (13)$$

d_i being the pixel-to-pixel colour difference between the segmented images. It must be noted that the smaller the value of these parameters (F, Q and SV), the better the segmentation result should be.

4.3. A comparative study

In order to perform the comparisons we have implemented two segmentation algorithms based on regular pyramids. One is an implementation of the linked pyramid proposed by Burt et al. [29] (LinRPyr), and the other one is a weighted linked pyramid with possibilistic linking (WeiRPyr). The weighted linked pyramid has been slightly modified by including a root extraction process that avoids the need for choosing a working level. Vertices that link only weakly to all their parents have been selected as root vertices. Unforced linking [9] has been used in the linked pyramid to select region roots at different pyramid levels. We also include comparisons with six segmentation algorithms based on irregular pyramids: the classical RAG hierarchy employed by Bertolino and Montanvert [61] (ClaIPyr), the localized pyramid [12] (LocIPyr); the segmentation algorithm proposed by Lallich et al. [13] (MorIPyr), the bounded irregular pyramid [14] (BouIPyr), the hierarchy of image partitions by dual graph contraction [1,6] (HieIPyr) and the hierarchical segmentation algorithm based on combinatorial pyramids proposed by Brun and Kropatsch [8] (ComIPyr). The two regular pyramid-based algorithms and the bounded irregular pyramid employ the HSI colour distance to perform the segmentation. The algorithm proposed by Lallich et al. [13] has been modified by the authors to deal with HSI colour images.

Several experiments have been conducted in order to compare the efficiency of the different segmentation algorithms. Particularly, 30 colour images from Waterloo and Coil 100 databases have been chosen. All these images have been resized to 256×256 pixels. In these experiments, we have employed a 766 MHz Pentium PC, i.e. a sequential processor. Algorithms proposed by Lallich et al. [13] and Haxhimusa and Kropatsch [1] are based on decimation procedures that have been mainly designed for parallel computing. Therefore, they do not efficiently run in this sequential computer. However, the algorithms proposed by Marfil et al. [14] and Brun and Kropatsch [8] are based on decimation techniques more suited to sequential computing. Specially, the union-find process has proven to be very efficient when run on sequential machines. Although it employs a decimation kernel designed for parallel computing, another fast algorithm is the LocIPyr which only processes a part of the image. In this case, the computational time associated to the local homogeneity analysis has been taken into account. The processing times are shown in Table 1. It can be noted that the fastest algorithms are based on regular pyramids. In these experiments, iterative relinking has been bounded to a maximum value of 10 iterations per level (LinRPyr). BouIPyr is also very fast because a large part of the image is processed following a classical regular pyramid approach. Table 1 also presents the maximum height associated to the hierarchical representation employed to perform the segmentation. The vertices of the pyramid level associated to this height define the segmentation in the ClaIPyr, LocIPyr, HieIPyr, MorIPyr and ComIPyr algorithms. In the rest of algorithms, roots can be defined in different levels of the hierarchy. In any case, it must be noted that this height does not correspond to the apex of the hierarchical representation, i.e. the pyramid level that only contains one vertex. According to the obtained data (Table 1), it can be appreciated that the two regular representations and the BouIPyr and HieIPyr irregular pyramids present the minimum heights. On the contrary, the ComIPyr, the ClaIPyr and the MorIPyr irregular pyramids present the maximum height values. Finally, Table 1 also shows the number of regions obtained by the different segmentation algorithms. It can be noted that the different values are very similar.

Fig. 13 shows two image tests used in our experiments, and the results obtained from all compared segmentation algorithms. Table 2 presents the comparison measurements among methods. The selection of the parameters of all algorithms has been conducted to obtain the best results according to the Q function. This table shows that all irregular pyramids obtain better segmentation results than regular ones. Only connected regions have been considered. For the regular pyramids, unconnected regions have been split into several smaller regions. It can be also noted that the MorIPyr and the ComIPyr present the best global results. In this last algorithm, the background region growing has been limited because this produce worse results. It must be also noted that the implicit partitioning of the initial graph using union-find-based process allows the execution times required by this algorithm to be greatly reduced. In MorIPyr, the test based on Moran's spatial autocorrelation coefficient allows the control of the decimation process both globally and locally at the same time. At a global level, it determines

Table 1
Processing times, height of the hierarchy employed by the segmentation algorithm and number of obtained regions. Average values have been obtained from 30 different images

	Processing times (s)			Hierarchy height			Number of regions		
	t_{min}	t_{ave}	t_{max}	h_{min}	h_{ave}	h_{max}	NR_{min}	NR_{ave}	NR_{max}
LinRPyr	0.94	1.37	1.81	9	9	9	17	81.6	203
WeiRPyr	0.31	0.40	0.58	9	9	9	19	79.7	148
ClaIPyr	2.51	3.96	7.68	17	36.7	72	9	84.1	210
LocIPyr	1.71	2.78	6.13	8	25.4	51	12	73.8	210
MorIPyr	2.43	3.47	4.47	13	33.3	62	45	107.7	201
BouIPyr	0.65	0.76	0.84	5	6.1	7	4	72.2	198
HieIPyr	4.07	4.29	4.91	10	11.6	18	23	76.2	149
ComIPyr	1.32	2.88	12.8	9	74.4	202	25	91.6	238

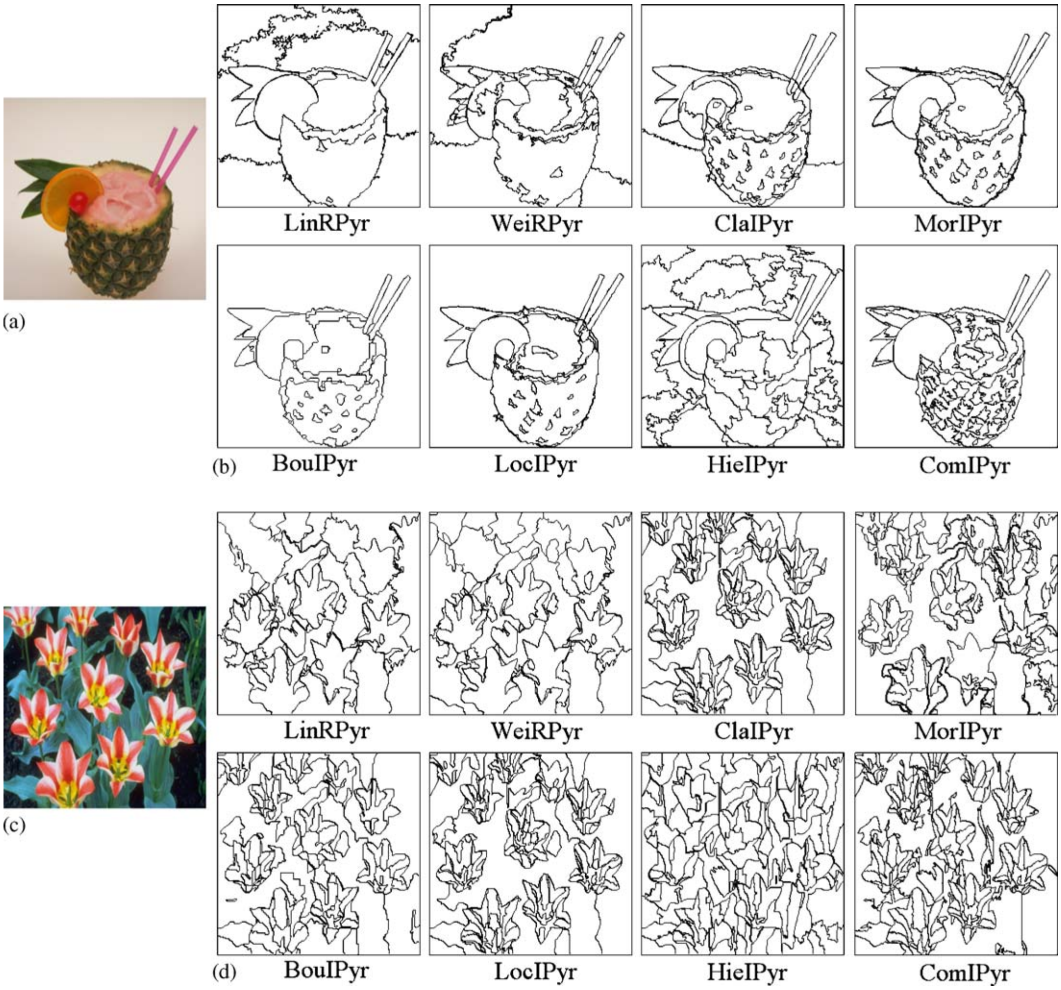

Fig. 13. (a) Input image #2; (b) segmentation images associated to (a) using different algorithms; (c) input image #3; and (d) segmentation images associated to (c) using different algorithms.

if the merging process must stop and, at a local level, this test extracts outliers in the distribution of regions merging candidates. Assuming the normal distribution of the errors [13], a vertex could be considered as an outlier if the distance between the tendency line of the Moran's scatterplot and the vertex is greater than a percentage tail of the distribution of error. In our experiments, the F and Q functions penalize the existence of small regions. However, outlier detection results in a more detailed segmentation with more regions. Therefore, a high threshold value has been employed

Table 2
F, Q and shift variance values. Average values have been obtained from 30 different images

	F			Q			SV		
	F_{min}	F_{ave}	F_{max}	Q_{min}	Q_{ave}	Q_{max}	SV_{min}	SV_{ave}	SV_{max}
LinRPyr	765.8	1070.4	1515.5	1052.1	1524.9	2105.4	37.8	66.9	83.5
WeiRPyr	791.2	1072.8	1428.2	1133.7	1480.6	2034.2	49.6	69.9	98.5
ClaIPyr	329.3	840.2	1290.0	479.1	1062.7	1590.3	18.0	28.8	42.8
LocIPyr	213.6	746.1	1345.6	489.4	1002.5	1327.4	20.8	31.7	46.7
MorIPyr	290.4	646.6	1043.7	360.5	817.6	1292.5	19.3	30.1	42.4
BouIPyr	198.6	711.7	1556.1	339.4	1086.7	1919.8	26.4	44.1	84.5
HieIPyr	201.7	689.2	1201.6	458.3	957.8	1521.5	18.5	27.1	35.9
ComIPyr	234.3	618.8	934.9	415.5	878.5	1294.5	21.3	30.7	42.8

(the 20% tail of the distribution of error). Using this threshold value, the behaviour of the MorIPyr is excellent, although it is the method that provides the highest number of obtained segmentation regions. On the contrary, the results obtained with the BouIPyr and the LocIPyr present the lowest number of regions. Compared to the results provided by the ClaIPyr, the results obtained with the LocIPyr present less regions but with better performance in terms of the F and Q functions. The results obtained by the BouIPyr are very similar to the ones obtained by the ClaIPyr or the LocIPyr. In order to reduce the number of small regions, several methods employ a threshold. In our experiments, this threshold has been set to 20 pixels in all algorithms (in the ComIPyr framework, this cleaning procedure implies that the hierarchy presents additional levels). The result associated to the HieIPyr is a hierarchy of partitions with multiple resolutions that is performed until the pyramid apex is reached. Although this hierarchy is suitable for further analysis, a hierarchy level must be selected in order to obtain an unique image segmentation. In this case, the level that provides the better Q has been chosen. Fig. 13b shows that this algorithm preserves details in low-variability regions (in this case, the background of the image). Image smoothing in low-variability regions would solve this problem[1]. In any case, this method provides perceptually important partitions in a bottom-up way based only on local differences. The height of the hierarchy is one of the lowest among the irregular approaches (in fact, only the BouIPyr has a lower height), so it is specially suitable to describe the image structure. Finally, it can be noted that the SV measure is high in the regular pyramids and in the BouIPyr approach, due to the regular-based reduction of great part of the image. In the rest of irregular pyramids, SV measures are very similar.

5. Conclusions

Previous work on evaluating the capability of pyramid approaches to segment an image conclude that classical pyramids have to be rejected as general-purpose image segmentation algorithms [4,15]. Specially, the work of Bister and Prewer and Kitchen show that classical regular pyramids are shift-, rotation- and scale-variant and hence that the results of segmentation are very data-dependent and not reproducible. In this work, these conclusions have been confirmed. In addition, Bister pointed out that some of the shortcomings of the classical pyramid segmentation algorithms might be attenuated or even avoided by the irregular pyramid segmentation scheme. The search for accurate hierarchical representation has lead in the last decade to the introduction of several new pyramids that try to solve these shortcomings. However, a comparison between these new structures and the regular ones is needed. In Ref. [15], the authors show that fuzzy regular pyramid approaches display less SV than the classical linked and weighted pyramids, but they do not develop conclusions about the performance of irregular pyramids. This paper has presented a review of pyramid structures employed in image segmentation that includes classical and recently proposed irregular pyramid structures. In this work, we have described several of these approaches. As expected, irregular pyramids yield better results than the regular ones. Besides, if the classical irregular pyramids achieves these results in exchange for computational efficiency, the new approaches have increased their efficiency and performance. Thus, the height reduction achieved by the hierarchy of partitions [1], the computational efficiency of the combinatorial pyramid [8], the efficient region growing control achieved with the Moran test [13] or the combination of different procedures for uniform or non-uniform regions [12,14] are several recently proposed strategies that have increased the global performance of irregular pyramid-based segmentation approaches.

Acknowledgements

The authors would thank Dr. Bister, Dr. Meer and Dr. Bertolino for their unselfish help. Special thanks are due to Dr. J. M. Jolion for his aid to implement the colour irregular pyramid and provides us the source code of the algorithm proposed in Ref. [13]. This source code has been easily

modified to deal with HSI images. Source code for the combinatorial pyramid-based algorithm [8] has been downloaded from http://www.greyc.ensicaen.fr/luc/PYRAMIDES/. Finally, the source code of the algorithm proposed by Haxhimusa and Kropatsch [1] has been downloaded from http://www.prip.tuwien.ac.at/yll/hierarchy/. These source codes have been slightly modified to provide the performance measures.

References

[1] Y. Haxhimusa, W.G. Kropatsch, Segmentation graph hierarchies, in: Proceedings of the Joint IAPR International Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition (SSPR & SPR), 2004, pp. 343–351.

[2] A. Rosenfeld, Multiresolution Image Processing and Analysis, Springer, Berlin, 1984.

[3] J.M. Jolion, A. Rosenfeld, A Pyramid Framework for Early Vision, Kluwer, Dordrecht, 1993.

[4] M. Bister, J. Cornelis, A. Rosenfeld, A critical view of pyramid segmentation algorithms, Pattern Recognition Lett. 11 (1990) 605–617.

[5] K. Cho, P. Meer, Image segmentation from consensus information, Comput. Vision Image Understanding 68 (1997) 72–89.

[6] W.G. Kropatsch, Y. Haxhimusa, Grouping and segmentation in a hierarchy of graphs, in: C.A. Bouman, E.L. Miller (Eds.), Computational Imaging II, SPIE, vol. 5299, 2004, pp. 193–204.

[7] J.M. Jolion, A. Montanvert, The adaptive pyramid, a framework for 2D image analysis, Comput. Vision Graphics Image Process.: Image Understanding 55 (3) (1992) 339–348.

[8] L. Brun, W.G. Kropatsch, Construction of combinatorial pyramids, in: E. Hancock, M. Vent. (Eds.), Graph Based Representations in Pattern Recognition, Lecture Notes in Computer Science, vol. 2726, Springer, Berlin, 2003, pp. 1–12.

[9] H.J. Antonisse, Image segmentation in pyramids, Comput. Graphics Image Process. 19 (1982) 367–383.

[10] D. Willersinn, W.G. Kropatsch, Dual graph contraction for irregular pyramids, in: 12th IAPR International Conference on Pattern Recognition, vol. 3, 1994, pp. 251–256.

[11] Y. Haxhimusa, R. Glantz, W.G. Kropatsch, Constructing stochastic pyramids by MIDES—maximal independent directed edge set, in: E. Hancock, M. Vento (Eds.), Fourth IAPR-RC15 Workshop on GbR in Pattern Recognition, Lecture Notes in Computer Science, vol. 2726, Springer, Berlin, 2003, pp. 35–46.

[12] J. Huart, P. Bertolino, Similarity-based and perception-based image segmentation, in: IEEE International Conference on Image Processing—ICIP 2005, 2005.

[13] S. Lallich, F. Muhlenbach, J.M. Jolion, A test to control a region growing process within a hierarchical graph, Pattern Recognition 36 (2003) 2201–2211.

[14] R. Marfil, J.A. Rodriguez, A. Bandera, F. Sandoval, Bounded irregular pyramid: a new structure for colour image segmentation, Pattern Recognition 37 (3) (2004) 623–626.

[15] D. Prewer, L.J. Kitchen, Soft image segmentation by weighted linked pyramid, Pattern Recognition Lett. 22 (2) (2001) 123–132.

[16] J. Liu, Y. Yang, Multi-resolution color image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 16 (7) (1994) 689–700.

[17] M. Borsotti, P. Campadelli, R. Schettini, Quantitative evaluation of color image segmentation results, Pattern Recognition Lett. 19 (1998) 741–747.

[18] R.H. Haralick, L.G. Shapiro, Image segmentation techniques, Comput. Vision Graphics Image Process. 29 (1985) 100–132.

[19] E.M. Riseman, A.R. Hanson, Design of a semantically directed vision processor, Technical Report 74(2–1), Computer and Information Science, University of Massachusets, Amherst, MA, 1974.

[20] S.L. Tanimoto, T. Pavlidis, A hierarchical data structure for picture processing, Comput. Graphics Image Process. 4 (2) (1975) 104–119.

[21] J.L. Crowley, R.M. Stern, Fast computation of the difference of low-pass transform, IEEE Trans. Pattern Anal. Mach. Intell. 6 (2) (1984) 212–222.

[22] A. Rosenfeld, Some pyramid techniques for image segmentation, Technical Report CAR-TR-203, Center for Automation Research, University of Maryland at College Park, 1986.

[23] L. Brun, W.G. Kropatsch, Receptive fields within the combinatorial pyramid framework, Graphical Models 65 (2003) 23–42.

[24] P.C. Chen, T. Pavlidis, Image segmentation as an estimation problem, Comput. Graphics Image Process. 12 (1980) 153–172.

[25] A. Gross, A. Rosenfeld, Multiresolution object detection and delineation, Comput. Vision Graphics Image Process. 39 (1) (1987) 102–115.

[26] T. Hong, A. Rosenfeld, Compact region extraction using weighted pixel linking in a pyramid, IEEE Trans. Pattern Anal. Mach. Intell. 6 (2) (1984) 222–229.

[27] P. Meer, Stochastic image pyramids, Comput. Vision Graphics Image Process. 45 (1989) 269–294.

[28] M. Shneier, Extracting linear features from images using pyramids, IEEE Trans. Syst. Man Cybern. 12 (1982) 569–572.

[29] P. Burt, T. Hong, A. Rosenfeld, Segmentation and estimation of image region properties through cooperative hierarchical computation, IEEE Trans. Syst. Man Cybern. 11 (12) (1981) 802–809.

[30] T. Hong, K. Narayane, S. Peleg, A. Rosenfeld, T. Silberberg, Image smoothing and segmentation by multiresolution pixel linking: further experiments and extensions, IEEE Trans. Syst. Man Cybern. 12 (5) (1982) 611–623.

[31] J.A. Hird, D.F. Wilson, A comparison of target detection and segmentation techniques, in: A.H. Lettington (Ed.), Optical Systems for Space and Defense, vol. 1191, 1989, pp. 375–386.

[32] Y. Ping, W. Runsheng, L. Diannong, A new image segmentation approach based on linked pyramid, in: International Conference on Signal Processing, 1996, pp. 1118–1121.

[33] P. Burt, Tree and pyramid structures for coding hexagonally sampling binary images, Comput. Graphics Image Process. 14 (1980) 171–180.

[34] N. Ahuja, On approaches to polygonal decomposition for hierarchical representation, Comput. Graphics Image Process. 24 (1983) 200–214.

[35] W.G. Kropatsch, A pyramid that grows by powers of 2, Pattern Recognition Lett. 3 (1985) 315–322.

[36] M. Spann, C. Horne, H. du Buf, The detection of thin structures in images, Pattern Recognition Lett. 10 (3) (1989) 175–179.

[37] S. Baronti, A. Casini, F. Lotti, L. Favaro, V. Roberto, Variable pyramid structures for image segmentation, Comput. Vision Graphics Image Process. 49 (3) (1990) 346–356.

[38] F. Ziliani, B. Jensen, Unsupervised image segmentation using the modified pyramidal linking approach, in: International Conference on Image Processing, vol. 3, 1998, pp. 303–307.

[39] M. Pietikäinen, A. Rosenfeld, Gray level pyramid linking as an aid in texture analysis, IEEE Trans. Syst. Man Cybern. 12 (1982) 422–429.

[40] J.M. Cibulskis, C.R. Dyer, An analysis of node linking in overlapped pyramids, IEEE Trans. Syst. Man Cybernet. 14 (1984) 424–436.

[41] B. Sabata, F. Arman, J. Aggarwal, Convergence of fuzzy-pyramid algorithms, J. Math. Imaging Vision 4 (1994) 291–302.

[42] S. Peleg, O. Federbusch, R. Hummel, Custom-made pyramids, in: Parallel Computer Vision, 1998, pp. 125–146.

[43] H. Rom, S. Peleg, Image representation using voronoi tessellation: adaptive and secure, IEEE Comput. Vision Pattern Recognition (1988) 282–285.

[44] J.M. Chassery, A. Montavert, A segmentation method in a Voronoi diagram environment, in: Scandinavian Conference on Imaging Processing, 1989, pp. 408–415.

[45] H. Bischof, K. Hornik, 1995. PCA-pyramids for image compression, in: Advances in Neural Information Processing Systems, vol. 7, 1995, pp. 941–948.

[46] H. Bischof, Pyramidal Neural Networks, Lawrence Erlbaum, New York, 1995.

[47] A. Weingessel, H. Bischof, K. Hornik, F. Leisch, Adaptive combination of PCA and VQ networks, IEEE Trans. Neural Networks 8 (5) (1997) 1208–1211.

[48] A. Montanvert, P. Meer, A. Rosenfeld, Hierarchical image analysis using irregular tessellations, IEEE Trans. Pattern Anal. Mach. Intell. 13 (4) (1991) 307–316.

[49] W.G. Kropatsch, Building irregular pyramids by dual graph contraction, IEE Proc. Vision Image Signal Process. 142 (6) (1995) 366–374.

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

[51] W.G. Kropatsch, From equivalent weighting functions to equivalent contraction kernels, in: E. Wenger, L.I. Dimitrov (Eds.), Digital Image Processing and Computer Graphics: Applications in Humanities and Natural Sciences, vol. 3346, SPIE, 1998, pp. 310–320.

[52] Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, W.G. Kropatsch, Logarithmic tapering graph pyramid, in: L. van Gool (Ed.), Proceedings of the 24th German Association for Pattern Recognition Symposium, Lecture Notes in Computer Science, vol. 2449, Springer, Berlin, 2002, pp. 117–124.

[53] W.G. Kropatsch, Y. Haxhimusa, Z. Pizlo, G. Langs, Vision pyramids that do not grow too high, Pattern Recognition Lett. 26 (3) (2005) 319–337.

[54] P. Nacken, Image segmentation by connectivity preserving relinking in hierarchical graph structures, Pattern Recognition 28 (6) (1995) 907–920.

[55] H. Ip, S. Lam, Alternative strategies for irregular pyramid construction, Image Vision Comput. 14 (4) (1996) 297–303.

[56] J.M. Jolion, Stochastic pyramid revisited, Pattern Recognition Lett. 24 (8) (2003) 1035–1042.

[57] W.G. Kropatsch, H. Macho, Finding the structure of connected components using dual irregular pyramids, in: Cinquieme Colloque DGCI, LLAIC1, Universite d'Auvergne, 1995, pp. 147–158.

[58] M. Burge, W.G. Kropatsch, A minimal line property preserving representations of line images, Comput. Devoted Issue Image Process. 62 (1999) 355–368.

[59] R.E. Tarjan, Efficiency of a good but not linear set union algorithm, J. ACM 22 (1975) 215–225.

[60] P. Meer, C. Sher, A. Rosenfeld, The chain pyramid: hierarchical contour processing, IEEE Trans. Pattern Anal. Mach. Intell. 12 (4) (1990) 363–376.

[61] P. Bertolino, A. Montanvert, Multiresolution segmentation using the irregular pyramid, in: International Conference on Image Processing, vol. 1, 1996, pp. 257–260.

[62] P.A.P. Moran, The interpretation of statistical maps, J. R. Stat. Soc. Ser. B (1948) 246–251.

[63] P.F. Felzenszwalb, D.P. Huttenlocher, Image segmentation using local variation, Comput. Vision Pattern Recognition (1998) 98–104.

[64] O. Borùvka, O jistém problému minimálnim, Acta Soc. Sci. Natur. Moravicae 3 (1926) 37–58.

[65] Y. Haxhimusa, A. Ion, W.G. Kropatsch, L. Brun, Hierarchical image partitioning using combinatorial maps, in: D. Chetverikov, L. Czúni, M. Vincze (Eds.), Proceedings of the Joint Hungarian-Austrian Conference on Image Processing and Pattern Recognition, pp. 179–186.

[66] Y. Zhang, A survey on evaluation methods for image segmentation, Pattern Recognition 29 (8) (1996) 1335–1346.