On morphological hierarchical representations for image processing and spatial data clustering

Pierre Soille, Laurent Najman

HAL Id: hal-00733251
https://hal.science/hal-00733251

Submitted on 18 Sep 2012

On morphological hierarchical representations for image processing and spatial data clustering[*]

Pierre Soille[1] and Laurent Najman[2]

[1]Institute for the Protection and Security of the Citizen,
Joint Research Centre, European Commission,
Via E. Fermi 2749, I-21027 Ispra (VA), Italy
[2]Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge
Equipe A3SI, ESIIE, France

Abstract. Hierarchical data representations in the context of classification and data clustering were put forward during the fifties. Recently, hierarchical image representations have gained renewed interest for segmentation purposes. In this paper, we briefly survey fundamental results on hierarchical clustering and then detail recent paradigms developed for the hierarchical representation of images in the framework of mathematical morphology: constrained connectivity and ultrametric watersheds. Constrained connectivity can be viewed as a way to constrain an initial hierarchy in such a way that a set of desired constraints are satisfied. The framework of ultrametric watersheds provides a generic scheme for computing any hierarchical connected clustering, in particular when such a hierarchy is constrained. The suitability of this framework for solving practical problems is illustrated with applications in remote sensing.

Keywords image representation, segmentation, clustering, ultrametric, hierarchy, graphs, connected components, constrained connectivity, watersheds, min-tree, alpha-tree.

1 Introduction

Most image processing applications require the selection of an image representation suitable for further analysis. The suitability of a given representation can be evaluated by confronting its properties with those required by the application at hand. In practice, images are often represented by decomposing them into primitive or fundamental elements that can be more easily interpreted. Examples of decomposition (or simply representation) schemes are given hereafter:

- A functional decomposition decomposes the image into a sum of elementary functions. The most famous functional decomposition is the Fourier transform which decomposes the image into a sum of cosine functions with a given frequency, phase, and amplitude. This proves to be a very effective

---

[*] A preliminary version of this paper was presented at the workshop WADGMM 2010 [1] held in conjunction with ICPR 2010, Istanbul, August 2010.

representation for applications requiring to target structures corresponding to well-defined frequencies;

- A pyramid decomposition relies on a shrinking operation which applies a low-pass filter to the image and downsamples it by a factor of two and an expand operation which upsamples the image by a factor of two using a predefined interpolation method. Such a scheme is extremely efficient in situations where the analysis can be initiated at a coarse resolution and refined by going through levels of increasing resolution;
- A multi-scale representation consists of a one-parameter family of filtered images, the parameter indicating the degree (scale) of filtering. This scheme is appropriate for the analysis of complex images containing structures at various scales;
- A skeleton representation consists in representing the image by a thinned version. It is useful for applications where the geometric and topological properties of the image structures need to be measured;
- The threshold decomposition decomposes a grey tone image into a stack of binary images corresponding to its successive threshold levels. This decomposition is useful as a basis for some hierarchical representations (see below) and from a theoretical point of view for generalising operations on binary images to grey tone images;
- A hierarchical representation of an image can be viewed as an ordered set or tree (acyclic graph) with some elementary components defining its leaves and the full image domain defining its root. Examples of elementary components are the regional minima/maxima/extrema, or the flat zones of the input image. This approach is interesting in all applications where the tree encoding the hierarchy offers a suitable basis for revealing structural information for filtering or segmentation purposes.

Note that these schemes are not mutually exclusive. A case in point is the skeleton representation defined in terms of maximal inscribed disks since it fits the multi-scale representation (with morphological openings with disks of increasing size as structuring elements) as well as the functional decomposition (with spatially localised disks as elementary functions that are unioned to reconstruct the original pattern).

A given representation scheme can be further characterised by considering the properties of the operations it relies on. For example, a representation is linear if it is based on operations invariant to linear transformations of the input image. The multi-scale representation with Gaussian filters of increasing size fulfils this property. Morphological representations are non-linear representations relying on morphological operations. For example, a granulometry is a morphological multi-scale representation originally proposed by Matheron in his seminal study on the analysis of porous media [2]. The representation does not need to rely exclusively on morphological operations to be considered as morphological. For example, the non-linear scale-space representation with levellings [3] is based on self-dual geodesic reconstruction using Gaussian filters of increasing size as geodesic mask.

This paper deliberately focuses on hierarchical image representations for image segmentation with emphasis on morphological methods. Note that the development of hierarchical representations appeared first in taxonomy in the form of hierarchical clustering methods (see for example [4] for an old but excellent review on classification including a discussion on hierarchical clustering). In fact, hierarchical image segmentation can be seen as a hierarchical clustering of spatial data. Graph theory is the correct setting for formalising clustering concepts as already recognised in [5] and [6], see also the enlightening paper [7] as well as the detailed survey and connections between graph theory and clustering in [8] (and [9] for clustering on directed graphs). For this reason, Sec. 2 presents briefly background notions and notations of graph theory used throughout this paper. Then, fundamental concepts of hierarchical clustering methods where the spatial location of the data points is usually not taken into account are reviewed in Sec. 3. Hierarchical image segmentation methods where the spatial location of the observations (i.e., the pixels) plays a central role are presented in a nutshell in Sec. 4. Recent recent paradigms developed for the hierarchical representation of images in the framework of mathematical morphology known as constrained connectivity and ultrametric watersheds are then developed in Sec. 5 while highlighting their links with hierarchical clustering methods. The framework of ultrametric watersheds provides a generic scheme for computing any hierarchical connected clustering, in particular when such a hierarchy is constrained. Before concluding, the problem of transition pixels is set forth in Sec. 6.

2   Background definitions and notations on graphs

The objects under study (specimens in biology, galaxies in astronomy, or pixels in image processing) are considered as the nodes of a graph. An edge is then drawn between all pairs of objects that need to be compared. The comparison often relies on a dissimilarity measure that assigns a weight to each edge. Following the notations of [10], we summarise hereafter graph definitions required in the context of clustering.

A *graph* is defined as a pair $X = (V, E)$ where V is a finite set and E is composed of unordered pairs of V, *i.e.*, E is a subset of $\{\{p, q\} \subseteq V \mid p \neq q\}$. Each element of V is called a *vertex or a point (of X)*, and each element of E is called an *edge (of X)*. If $V \neq \emptyset$, we say that X is *non-empty*.

As several graphs are considered in this paper, whenever this is necessary, we denote by $V(X)$ and by $E(X)$ the vertex and edge set of a graph X.

Let X be a graph. If $u = \{p, q\}$ is an edge of X, we say that p and q are *adjacent (for X)*. Let $\pi = \langle p_0, \ldots, p_\ell \rangle$ be an ordered sequence of vertices of X, π is *a path from p_0 to p_ℓ in X (or in V)* if for any $i \in [1, \ell]$, p_i is adjacent to p_{i-1}. In this case, we say that p_0 and p_ℓ are *linked for X*. We say that *X is connected* if any two vertices of X are linked for X.

Let X and Y be two graphs. If $V(Y) \subseteq V(X)$ and $E(Y) \subseteq E(X)$, we say that *Y is a subgraph of X* and we write $Y \subseteq X$. We say that Y is a *connected component of X*, or simply a *component of X*, if Y is a connected subgraph of X

3

which is maximal for this property, *i.e.*, for any connected graph Z, $Y \subseteq Z \subseteq X$ implies $Z = Y$.

Clustering methods generally work on a complete graph $(V, V \times V)$. In this case, the notion of connected component is not an important one, as any subset is obviously connected. On contrary, this notion is fundamental for image segmentation.

Let X be a graph, and let $S \subseteq E(X)$. The *graph induced by S* is the graph whose edge set is S and whose vertex set is made of all points that belong to an edge in S, *i.e.*, $(\{p \in V(X) \mid \exists u \in S, p \in u\}, S)$.

In the sequel of this paper, $X = (V, E)$ denotes a connected graph, and the letter V (*resp. E*) will always refer to the vertex set (*resp.* the edge set) of X. We will also assume that $E \neq \emptyset$. Let $S \subset E$. In the following, when no confusion may occur, the graph induced by S is also denoted by S. If $S \subset E$, we denote by $\overline{S}$ *the complementary set of S in E*, i.e., $\overline{S} = E \setminus S$.

Typically, in applications to image segmentation, V is the set of picture elements (pixels) and E is any of the usual adjacency relations, *e.g.*, the 4- or 8-adjacency in 2D [11]. In all examples, 4-adjacency is used.

We consider in this paper weighted graphs, and either the vertices or the edges of a graph can be weighted. We denote the weight on the vertives of V by f, and the weights on the edges of E by F. For application to image processing, f is generally some information on the pixels (e.g., the grey level of the considered pixel), and F represents a dissimilarity (e.g., $F(\{p, q\}) = |f(p) - f(q)|$).

3   Hierarchical clustering

Clustering can be defined as a method for grouping objects into homogeneous groups (called clusters) on the basis of empirical measures of similarity among those objects. Ideally, the method should generate clusters maximising their internal cohesion and external isolation. Analogously to the categorisation of classification methods proposed in [12], any clustering methodology can be characterised by three main properties. The first concerns the relation between object properties and clusters. It indicates whether the clusters are monothetic or polythetic. A cluster is monothetic if and only if all its members share the same common property or properties. The second property regards the relation between objects and clusters. It indicates whether the clusters are exclusive (i.e., non-overlapping) or overlapping. Non-overlapping clustering methods can be defined as partitional in the sense that they realise a partition of the input objects (a partition of a set is defined as division of this set in disjoint non-empty subsets such that their union is equal to this set). Non-partitional clustering allows for overlap between clusters, see [13] for an early reference on this topic and [14] for recent developments. The third property refers to the relation between clusters. It indicates whether the clustering method is hierarchical (also called ordered) or non-hierarchical (unordered).

Because we are chiefly interested in image segmentation applications, we focus on clustering methods that are monothetic, partitional, and hierarchical.

The term *hierarchical clustering* was first coined in [15]. A hierarchical clustering can be viewed as a sequence of nested clusterings such that a cluster at a given level is either identical to a cluster already existing at the previous level or is formed by unioning two or more clusters existing at the previous level. It is convenient to represent this hierarchy in the form of a tree called *dendrogram* [16] or *taxonomic tree* (see [17] for this latter terminology as well as a procedure which in essence already defined the concept of hierarchical clustering). The first detailed study about the use of trees in the context of hierarchical clustering appeared in [18].

By construction, a hierarchical clustering is parameterised by a non-negative real number λ indicating the level of a given clustering in the hierarchy. At the bottom level, this number is equal to zero and each object correspond to a cluster so that the finest possible partition is obtained. At the top level only one cluster containing all objects remains. Given any two objects, it is possible to determine the minimum level value for which these two objects belong to the same cluster. A key property of hierarchical clustering is that the function that measures this minimum level is an *ultrametric*. An ultrametric is a measurement that satisfies all properties of a metric (distance) plus a condition stronger than the triangle inequality and called ultrametric inequality. It states that the distance between two objects is lower than or equal to the maximum of the distances calculated from (i) the first object to an arbitrary third object and (ii) this third object to the second object. Denoting by d the ultrametric function and p, q, and r respectively the first, second and third objects, the ultrametric inequality corresponds to the following inequality:

$$d(p,q) \leq \max\{d(p,r), d(r,q)\}.$$

The ultrametric property of hierarchical clustering was discovered simultaneously in [15,19], see also [20] for a thorough study on ultrametrics in classification. An example of dendrogram is displayed in Fig. 1.
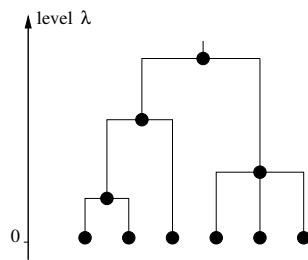
Fig. 1. An example of dendrogram starting from 6 objects at the bottom of the hierarchy (level $\lambda = 0$). At the top of the hierarchy, there remains only one cluster containing all objects.

The measure of similarity between the input objects requires the selection of a dissimilarity measurement. A dissimilarity measurement between the elements of a set V is a function $d^\star$ from $V \times V$ to the set of nonnegative real numbers satisfying the three following conditions: (i) $d^\star(p,q) \geq 0$ for all $p, q \in V$ (i.e., positiveness), (ii) $d^\star(p,p) = 0$ for all $p \in V$, and (iii) $d^\star(p,q) = d^\star(q,p)$ for all $p, q \in V$ (i.e., symmetry). Starting from an arbitrary dissimilarity measurement, it is possible to construct a hierarchical clustering: if the dissimilarity is increasing with the merging order, an ultrametric distance between any two objects (or clusters) can be defined as the dissimilarity threshold level from which these two objects (or clusters) belong to the same cluster; if if the dissimilarity is *not* increasing with the merging order, then any increasing function of the merging order can be used.

In practice, the hierarchy is constructed by an iterative procedure merging first the object pair(s) with the smallest dissimilarity value so as to form the first non-trivial cluster(s) (i.e., non reduced to one object). To proceed, the dissimilarity measurement between objects needs to be extended so as to be applicable to clusters. Let C_i and C_j denote two clusters obtained at a given iteration level. The dissimilarity between between these two clusters is naturally defined as a function of the dissimilarities between the objects belonging to these clusters:

$$d^\star(C_i, C_j) = f\{d^\star(p,q)) \mid p \in C_i \text{ and } q \in C_j\}.$$

Typical choices for the function f are the minimum or maximum. The maximum rule leads to the complete-linkage clustering (sometimes called maximum method) and dates back to [21]. Complete-linkage is subject to ties in case the current smallest dissimilarity value is shared by two or more clusters. Consequently, one of the possible merge must be chosen and often this can only be achieved by resorting to some arbitrary (order dependent or random) selection. By construction, complete-linkage favours compact clusters. On the other hand, the minimum rule is not subject to ties (and is therefore uniquely defined) and does not favour compact clusters. The resulting clustering is called the single-linkage clustering[1] (sometimes called minimum method). Indeed, only the pair (link) with the smallest dissimilarity value is playing a role.

The single-linkage clustering is closely related to the minimum spanning tree [23], defined as follows. To any edge-weighted graph X, the number $F(X) = \sum_{u=\in E(X)} F(u)$ is the *weight* of the graph. A spanning tree of a connected graph X is a graph whose vertex set is equal to $V(X)$ and whose edge set is a subset of $E(X)$ such that no cycles are formed. A spanning tree of X with minimum weight is called a *minimum spanning tree* of X.

Indeed, the hierarchy underlying the single-linkage clustering is at the root of the greedy algorithm of Kruskal [24] for solving the minimum spanning tree

[1] The concept of single-linkage and its use for classification purposes were apparently suggested for the first time in [22] while the terminology single-linkage seems to be due to Sneath, see [16, p. 180] where it is also called Sneath's method.

problem[2]. In this algorithm, referred to as 'construction A' in [24], the edges of the graph are initially sorted by increasing edge weights (in a clustering perspective, the nodes of the graph are the objects and the edge weights are defined by the dissimilarity measurements between the objects). Then, a minimum spanning tree MST is defined recursively as follows: the next edge is added to MST if and only if together with MST it does not form a circuit. That is, there is a one-to-one correspondence between (i) the clusters obtained for a given dissimilarity level and (ii) the subtrees obtained for a distance equal to this level in Kruskal's greedy solution to the minimum spanning tree problem.

While the single-linkage is not subject to ties, it is sensitive to the presence of objects of intermediate characteristics (transitions) that may occur between two clearly defined populations, see [27] for a detailed discussion as well as Sec. 6. This effect is sometimes called 'chaining-effect' although this latter terminology is somewhat misleading for chaining is the very principle of single-linkage [28].

4 Hierarchical image segmentation

After a brief discussion on the definition of image segmentation and hierarchical image segmentation (see Sec. 4.1), methods relying on graph representations are presented (Sec. 4.2) and then those developed in MM (Sec. 4.2).

4.1 From image segmentation to hierarchical image segmentation

A segmentation of the definition domain V of an image is usually defined as a partition of V into disjoint connected subsets $V_i, \ldots, V_n$ (called segments) such that there exists a logical predicate P returning true on each segment but false on any union of adjacent segments [29,30]. That is, a series of subsets V_i of the definition domain V of an image forms a segmentation of this image if and only if the following four conditions are met (i) $\cup_i(V_i) = X$, (ii) $V_i \cap V_j = \emptyset$ for all $i \neq j$, (iii) $P(V_i) =$ true for all i, and (iv) $P(V_i \cup V_j) =$ false if V_i and V_j are adjacent. The first condition requires that every picture element (pixel) must belong to a segment. The second condition requires that each segment does not overlap any other segment. The third condition determines what kind of properties each segment must satisfy, i.e., what properties the image pixels must satisfy to be in the same segment. The fourth condition ensures that the segments are maximal in the sense that specifies that any merging of any adjacent regions would violate the third condition.

Note that uniqueness of the resulting segmentation given a predicate is not required. If uniqueness is desired, the predicate should rely on an equivalence relation owing to the one-to-one correspondence between the unique partitions of a set and the equivalence relations on it, see for example [31, p. 48]. Interestingly, the relation 'is connected' is an equivalence relation since it is reflexive (a point

---

[2] The first explicit formulation of the minimum spanning tree problem is attributed to [25], see detailed account on the history of the problem in [26].

is connected to itself by a path of length 0), symmetric (if a point p is connected to a point q then q is connected to p since the reversal of a path is path), and transitive (if p is connected to q and q to r then p is connected to r since the concatenation of two paths is a path). Any given connectivity relation partitions the set of pixels of a given input image into equivalent classes called connected components [32]. They are maximal subsets of pixels such that every pair of pixels belonging to such a subset is connected. The resulting partition meets therefore all conditions of a segmentation.

The segments resulting from a segmentation procedure are analogous to the clusters obtained when clustering data. Clustering techniques can be applied to image data for either classification or segmentation purposes. In the former case, the spatial position of the pixels does not necessarily play a role for clusters are searched in a parametric space such as the multivariate histogram. The resulting clusters partition the parametric space into a series of classes and this partition is used as a look-up-table to indicate the class of each pixel of the input image. An example of this approach using morphological clustering is proposed in [33]. Contrary to data clustering applied to non-spatial data, the dissimilarity measurements between the data samples (i.e., the pixels) are not measured between all possible pairs. Indeed, the spatial position of the pixels plays a key role so that measurements are only performed between adjacent pairs of pixels. That is, the full dissimilarity matrix is very sparse: for a image of $m \times n$ pixels, there are $2mn - m - n$ entries in the $(mn)^2 \times (mn)^2$ dissimilarity matrix when considering 4-adjacency relation.

By analogy with hierarchical clustering, hierarchical segmentation can be defined as a family of fine to coarse image partitions (i.e., family of ordered partitions) parameterised by a non-negative real number indicating the level of a given partition in the hierarchy. Hierarchical segmentation is useful to help the detection of objects in an image. In particular, it can be used to simplify the image in such a way that the elementary picture elements are not anymore the pixels but connected sets of pixels. Indeed, in image data, analogues to phonemes and characters correspond to structural primitives that compress the data to a manageable size without eliminating any possible final interpretations [34]. It should be emphasised that a hierarchical segmentation does not necessarily deliver segments directly corresponding to the searched objects. This happens for instance when an object is not characterised by some homogeneity/separation criteria but from the consideration of an a priori model of the whole object (e.g. perceptual grouping and Gestalt theory).

There exists a fundamental difference between segmentation and classification. Indeed, contrary to classification, segmentation requires the explicit definition of an adjacency graph or, more generally, a connection [35,36]. Typically, the k-nearest neighbouring graph with k equal to 4 or 8 is used for processing 2-dimensional images. With classification, a decision about the class (i.e., label) of each pixel can be reached without using its spatial context (position) so that it does not necessarily need the definition of an adjacency graph. Nevertheless, any classification can be used to generate a segmentation. Indeed, once an adjacency

graph is added to the classified image, the maximal connected regions of pixels belonging to the same class generate a segmentation of the image definition domain. If the considered adjacency graph is the complete graph, a one-to-one correspondence between the classes and the resulting connected components is obtained.

Hereafter, a selection of techniques achieving hierarchical image segmentation is proposed, extending the initial survey proposed in [37]. We start with generic methods based on graph representations and then proceed with specific methods developed in the context of mathematical morphology. Recent developments related to constrained connectivity and ultrametric watersheds are discussed in Sec. 5.

4.2   Methods based on graph representations

Horowitz and Pavlidis [29,38] are among the first to suggest a formulation of hierarchical image segmentation in a graph theoretical framework. It is based on the split-and-merge algorithm. Because their implementation relies on a regular pyramid data structure with square blocks, it is not translation invariant and it favours blocky edges owing to the initial regular split of the image. In addition, the grouping stage of split-and-merge algorithms is order dependent, a drawback of all procedures updating the features of a region once new points are added to it.

The idea of applying the single-linkage clustering method to produce hierarchical image segmentation was implemented for the first time by Nagao [39,40] for processing aerial images using grey level differences between adjacent pixels as dissimilarity measurement. For colour images, the resulting dissimilarity vector led to the notion of differential threshold vector in [41]. The application of single-linkage clustering to image data are further developed in [42] using a graph theoretic framework. This latter paper also details a minimax SST (Shortest Spanning Tree) segmentation allowing for the initial minimum spanning tree to be partitioned into n subtrees by recursively splitting the subtree with the larger cost into 2 subtrees (see also recursive SST segmentation into n regions). Note that single-linkage clustering based on grey level difference dissimilarity was rediscovered much later in morphological image processing under the term quasi-flat zones [43,3]. More recently, the more general and appropriate term of α-connected component was proposed in [37] to refer to any connected component (i.e., maximal set of connected pixel) of pixels such that any pair of pixels of this connected component can be linked by a path such that the dissimilarity value between two successive pixels of the path does not exceed a given dissimilarity threshold value (see details in Sec. 5.1). The ultrametric behind the single-linkage hierarchical image segmentation is analogous to the one defined for single-linkage clustering, see Sec. 3.

The hierarchy of graphs (irregular pyramids) proposed recently in [44,45] builds on the graph weighted partitions developed in [46,47] and inspired by the seminal work of Zahn [7] on point data clustering and its extension to graph cut image segmentation in [48,49]. It relies on weighted graphs where each element

9

of the edge set is given a weight corresponding to the range of the values of its two nodes. The internal contrast of a connected component corresponds to the largest weight of all edges belonging to this connected component (an edge belongs to a connected component if its corresponding nodes belong to it or, alternatively, to a spanning tree of minimum sum of edge weights). The external contrast is defined as the smallest weight of the edges linking a pixel of the considered connected component to another one. The hierarchy is achieved by defining a dissimilarity measure accounting for both the internal and external contrasts. The successive levels of the hierarchy are then obtained by iteratively merging the adjacent connected components of minimum dissimilarity. An up-to-date survey (including comparisons) of both regular and irregular pyramidal structures can be found in [50]. A survey on graph pyramids for hierarchical segmentation is proposed in [51].

The hierarchical image segmentation based on the notion of the cocoons of a graph relies on a complete-linkage hierarchy and its corresponding ultrametric [52]. The same authors introduced the notion of scale-sets [53] where the dissimilarity measurement is replaced by a two-term energy minimization process where the first term accounts for the amount of information required to encode the deviation of the data against the region model (typically taken as the mean of the region) and the second term is proportional to the amount of information required to encode the shape of the model (typically taken as the boundary length of the region).

In [54], the extrema mosaic (influence zones of the image regional extrema) is considered as the base level of the hierarchy. The dissimilarity between the segments is defined as the average gray level difference along the common boundary of these segments. This dissimilarity is increasing with the merging order and is therefore an ultrametric. Generic ultrametric distances obtained by integrating local contour cues along the regions boundaries and combining this information with region attributes are proposed in [55].

4.3   Methods developed in mathematical morphology

Mathametical morphology relies on the notion of lattices, and a theory devoted to segmentation in this context recently appears [35,36]. From a practical point of view, most of the application schemes use either a watershed-based approach or a tree-based approach.

Watershed based The waterfall algorithm [56,57,58] can be considered as the first morphological hierarchical image segmentation method. The elementary components of the base level of the tree underlying the waterfall hierarchy are the catchment basins of the gradient of the image. Each basin is then set to the height of the lowest watershed pixel surrounding this basin while the watershed pixels keep their original value. The watersheds of the resulting image delivers basins corresponding to the subsequent level of the hierarchy. The procedure is then iterated until only one basin matching the image domain is obtained. This

hierarchy of partitions can be implemented directly on graph data structures as detailed in [59].

Watershed hierarchies using the notion of contour dynamic is proposed in [60]. The arcs of the watersheds of the gradient of the original image are valued by their contour dynamic. More precisely, the contour dynamic of an arc of a watershed separating two basins is defined as the height difference between the lowest point of this arc and the height of the highest regional minimum associated with these two basins. The contour dynamic is a dissimilarity that satisfies all properties of an ultrametric. The resulting contour dynamic map is a saliency map representing a hierarchy. Indeed, a fine to coarse family of partitions is obtained by thresholding the contour dynamic map for increasing contour dynamic values. By associating other dissimilarity measures to the arcs of the watersheds, other partition hierarchies are obtained.

Note that, if one wants to obtain theoretical results associating definitions and properties [61], one has to work on edge-weighted graphs with the watershed-cut definition [62] that links the watershed with the minimum spanning tree as initially pointed out in [63].

Tree based Another type of hierarchy is obtained by considering the flat zones of the image as the finest partition and then iteratively merging the most similar flat zones. This resulting tree is called binary partition trees in [64]. The tree always represents a hierarchy indexed by the merging order and not always the dissimilarity since the one used in [64] is not an ultrametric.

Another tree, known as the component tree [65,66] of the vertices (called max-tree or min-tree in [67] depending on whether its leaves are matching the image maxima or minima) represents the hierarchy of the level sets of the image and are therefore not directly representing a hierarchy of partitions of the image definition domain.

However, when defined not on the vertices but on the edges, we will see below that the component tree is indeed a dendrogram representing a hierarchy of connected partitions.

Reviews on hierarchical methods developed in mathematical morphology based on watersheds are presented in [68,69], and on trees in [70,71]. Recent developments related to constrained connectivity and ultrametric watersheds are developed in the next section.

5   Constrained connectivity and ultrametric watersheds

5.1   Constrained connectivity

Preliminaries Let us first recall the notion of α-connectivity that corresponds to single-linkage clustering applied to image data, see Sec. 4.2. Two pixels p and q of an image f are α-connected if there exists a path going from p to q such that the dissimilarity between any two successive pixels of this path does not exceed the value of the local parameter α. By definition, a pixel is α-connected to itself.

Accordingly, the α-connected component of a pixel p is defined as the set of image pixels that are α-connected to this pixel. We denote this connected component by α-CC(p): α-CC$(p) = \{p\} \cup \big\{ q \mid$ there exists a path $\mathcal{P} = \langle p = p_1, \ldots, p_n = q \rangle$, $n > 1$, such that $F(\{p_i, p_{i+1}\}) \leq \alpha$ for all $1 \leq i < n \big\}$. In the case of grey level images and when considering the absolute intensity difference as dissimilarity measure, the α-connected components of an image are equivalent to its quasi-flat zones [43,3]. Note that the edges of the connected graph corresponding to a given α-connected component is defined by the pairs of adjacent pixels belonging to this α-connected component such that their associated dissimilarity (weight) does not exceed α.

Definitions and properties The constrained connectivity paradigm [72,37] originated from the need to develop a method preventing the formation of α-connected components whose range values exceed that specified by the local range parameter α (assuming that the dissimilarity between two pixels is the absolute difference of their intensity values, see [73,74] for other examples of dissimilarity measures). This is simply achieved by looking for the largest α-connected components satisfying a global range constraint referred to as the global range parameter denoted by ω:

$$(\alpha, \omega)\text{-CC}(p) = \bigvee \Big\{ \alpha_i\text{-CC}(p) \;\Big|\; \alpha_i \leq \alpha \text{ and } \mathsf{R}\Big(\alpha_i\text{-CC}(p)\Big) \leq \omega \Big\},$$

where the range function R calculates the difference between the maximum and the minimum values of a nonempty set of intensity values. Note that the (α, ω)-connected components for $\alpha \geq \omega$ are equivalent to those obtained for $\alpha = \omega$. That is, when $\alpha \geq \omega$ the local range parameter does not play a role. This leads to the concept of (ω)-connected component[3]:

$$(\omega)\text{-CC}(p) = (\alpha \geq \omega, \omega)\text{-CC}(p) = \bigvee \Big\{ \alpha_i\text{-CC}(p) \mid \mathsf{R}\Big(\alpha_i\text{-CC}(p)\Big) \leq \omega \Big\}.$$

The corresponding global dissimilarity measurement $d_\Omega^\star$ between two pixels is defined by the smallest range of the α-connected components containing these two pixels. This dissimilarity measurement satisfies also the ultrametric inequality. Accordingly, we obtain the following equivalent definition of a (ω)-connected component: (ω)-CC$(p) = \{q \mid d_\Omega^\star(p, q) \leq \omega\}$. In contrast to what happens with the local dissimilarity measurement $d_A^\star$, the range of the values of arbitrary pairs of pixels belonging to the same (ω)-connected component is limited, the maximal value of this range being equal to ω. Therefore, the resulting clustering bears some resemblance to the complete linkage clustering suggested in [21] but, contrary to the latter procedure, it is unequivocal (see [16, pp. 181-182] for an account on the equivocality of the complete linkage clustering). The generalisation of the concept of constrained connectivity to arbitrary constraints is presented in [72].

---

[3] The parenthesis is not dropped to avoid confusion with α-connected components when the Greek letters are replaced by a numerical value indicating the actual value of the corresponding range parameter.

Separation value The separation value $\Delta^\wedge$ of an iso-intensity connected component (flat-zone) can be defined in terms of grey tone hit-or-miss transforms [75] with *adaptive* composite structuring elements. The adaptive hit-or-miss transform of a pixel with the composite structuring element containing the origin $\mathbf{o}$ for the foreground component and its direct neighbours having a strictly lower value $N^<(\mathbf{o})$ for the background component outputs the difference between the input pixel value and that of its largest lower neighbour(s) if the set of its lower neighbours is non-empty, 0 otherwise. This adaptive hit-or-miss transform is denoted by $\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))}$:

$$[\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))}](p) = \begin{cases} f(p) - \vee\{f(q) \mid q \in N^<(p)\} \text{ if } N^<(p) \neq \emptyset, \\ 0 \qquad\qquad\qquad\qquad\qquad \text{otherwise.} \end{cases}$$

Similarly, the adaptive hit-or-miss transform $\mathrm{HMT}_{(N^>(\mathbf{o}),\mathbf{o})}$ of a pixel outputs the difference between the value of its smallest greater neighbour(s) and that of the pixel itself, if the set of its greater neighbours $N^>(\mathbf{o})$ is non-empty, 0 otherwise:

$$[\mathrm{HMT}_{(N^>(\mathbf{o}),\mathbf{o})}](p) = \begin{cases} \wedge\{f(q) \mid q \in N^>(p)\} - f(p) \text{ if } N^>(p) \neq \emptyset, \\ 0 \qquad\qquad\qquad\qquad\qquad \text{otherwise.} \end{cases}$$

The non-zero values of the point-wise minimum between the two hit-or-miss transforms corresponds to the transition pixels in the sense that these pixels have simultaneously lower and greater neighbours (and the point-wise minimum image indicates the minimum height of the transition). The binary mask of transition pixels can therefore be obtained by the following operator denoted by TP:

$$\mathrm{TP} = T_{>0}[\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))} \wedge \mathrm{HMT}_{(N^>(\mathbf{o}),\mathbf{o})}].$$

In [76], the same mask is obtained by considering the non-zero values of the point-wise minimum of the gradients by erosion and dilation with the elementary neighbourhood (the pixel and its direct neighbours) as structuring element. In this latter case, the point-wise minimum image indicates the maximum height of the transition.

The minimum separation value of a pixel of an image is defined as the minimum intensity difference between a pixel and its neighbour(s) having a different value from this pixel if such neighbour(s) exist, 0 otherwise. It is denoted by $[\Delta^\wedge(f)](p)$ and can be calculated as follows:

$$[\Delta^\wedge(f)](p) = \begin{cases} [\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))}(f)](p) \text{ if } [\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))}(f)](p) < [\mathrm{HMT}_{(N^>(\mathbf{o}),\mathbf{o})}(f)](p) \\ \qquad\qquad\qquad \text{and } [\mathrm{HMT}_{(\mathbf{o},N^<(\mathbf{o}))}(f)](p) \neq 0, \\ [\mathrm{HMT}_{(N^>(\mathbf{o}),\mathbf{o})}(f)](p) \text{ otherwise.} \end{cases}$$

The minimum separation value of an iso-intensity connected component 0-CC is then defined as the smallest (minimum) separation value of its pixels:

$$\Delta^\wedge(\text{0-CC}) = \wedge\{\Delta^\wedge(q) \mid q \in \text{0-CC and } \Delta^\wedge(q) \neq 0\}.$$

13

It is equivalent to the smallest α value such that α-CC $\neq$ 0-CC. Similarly, the operator that sets each pixel of the image to the minimum separation value of the iso-intensity connected component it belongs to is defined as follow:

$$[\Delta^\wedge(0\text{-CC}(f))](p) = \wedge\{\Delta^\wedge(q) \mid q \in 0\text{-CC}(p) \text{ and } \Delta^\wedge(q) \neq 0\}.$$

It can be viewed as an adaptive operation where the output value at a given pixel depends on the iso-intensity component of this pixel and the neighbouring pixels of this component. By replacing the $\wedge$ operation with the $\vee$ operation in the minimum separation definitions, we obtain the definitions for maximal separations. Figure 2 illustrates the map of minimal separation of the pixels and iso-intensity connected components of a synthetic image.

1	3	8	7	8	8	2
2	1	9	8	8	9	1
1	0	4	1	1	2	5
1	1	9	3	4	2	6
3	2	7	9	9	1	1
1	0	8	4	9	6	7
0	2	9	3	8	5	9

1	2	1	1	1	1	1
1	1	1	1	1	1	1
1	1	3	2	1	1	1
2	1	2	1	1	1	1
1	1	1	2	5	1	5
1	1	1	1	1	1	1
1	2	1	1	1	1	2

1	2	1	1	1	1	1
1	1	1	1	1	1	1
1	1	3	1	1	1	1
1	1	2	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	2	1	1	1	1	2

Fig. 2. Left: a synthetic 7×7 image f with its intensity values [37, Fig. 2a]. Middle: the map of separation value of its pixels $\Delta^\wedge(f)$. Right: the map of separation value of its flat zones $\Delta^\wedge(0\text{-CC}(f))$.

The regional maxima RMAX of $\Delta^\wedge(0\text{-CC}(f))$ can be used to flag the flat zones that are the most isolated. Conversely, the regional minima RMIN of $\Delta^\wedge(0\text{-CC}(f))$ can be used to flag the flat zones from which an immersion simulation should be iniated to compute the successive levels of the hierarchy of constrained components. By doing so, an algorithm similar to the watershed by flooding simulation [77] can be designed.

Alpha-tree representation Constrained connectivity relies on the definition of α-connectivity. The later boils down to the single-linkage clustering of the image pixels given the underlying dissimilarity measure between adjacent pixel pairs. The corresponding single-linkage dendrogram was described as a *spatially rooted tree* in [37]. This spatially rooted tree was introduced as the *alpha-tree* in [78,79]. It represents the fine to coarse hierarchy of partitions for an increasing value of the dissimilarity threshold α. The alpha-tree can also be seen as a component tree representing the ordering relations of the α-connected components of the image. The representation in terms of min-tree is developed in Sec. 5.2.

In the case of constrained connectivity, a given (α, ω)-partition corresponds to the highest cut of the alpha-tree such that all the nodes below this cut satisfy

the α and ω constraints. Usually this cut is not horizontal. A given (ω)-partition corresponds to the highest cut of the alpha-tree such all the nodes below the cut satisfy the ω constraint. Alternatively, a (ω)-partition can be obtained by performing a horizontal cut in the dendrogram based on the ultrametric $d_\Omega^\star$ (i.e., the omega-tree). An example of omega-tree is given [80]. Note however that the set of all (α, ω)-partitions is itself not ordered given the absence of order between arbitrary pairs of local and global dissimilarity threshold values.

Edge-weighted graph setting and minimum spanning tree By construction, the connected components of the graph $G[\alpha] = (V, \{\{p, q\} \in E \mid F(\{p, q\}) \leq \alpha\})$ are equivalent to the α-connected components of f. Since α-connectivity corresponds to single-linkage clustering, there is an underlying minimum spanning tree associated to it (see also section 3 and [42] for equivalent image segmentations based on the direct computation of a minimum spanning tree). More precisely, the minimum spanning tree of the edge-weighted graph of an image is a tree spanning its pixels and such that the sum of the weights associated with the edges of the tree is minimal. Denoting by $E_{\min}$ the edge set of a minimum spanning tree of the edge-weighted graph of an image, the connected components of the graph $(V, \{\{p, q\} \in E_{\min} \mid F(\{p, q\}) \leq \alpha\})$ are equivalent to those of $G[\alpha]$ (equivalent in the sense that given any node, the set of nodes of the connected component of $(V, \{\{p, q\} \in E_{\min} \mid F(\{p, q\}) \leq \alpha\})$ containing this node is identical to the set of nodes of the connected component of $G[\alpha]$ containing this very node). Since the minimum spanning tree representation contains less edges than the initial edge-weighted graph, it is less memory demanding for further computations such as global range computations. However, not all computations can be done on the minimum spannning tree (for example, connectivity constraints relying on the computation of a connectivity index [37] cannot be derived from it).

5.2 Ultrametric watersheds: from hierarchical segmentations to saliency maps

We have several different ways to deal with hierarchies: dendrograms and minimum spanning trees. In the case where a hierarchy is made of connected regions, then we can also use its connected component tree, e.g., min-tree, max-tree or alpha-tree. None of these three tools allows for an easy visualisation of a given hierarchy as an image. We now introduce ultrametric watershed [81,82] as a tool that helps visualising a hierarchy: we stack the contours of the regions of the hierarchy; thus, the more a contour of a region is present in the hierarchy, the more visible it is. Ultrametric watershed is the formalisation and the caracterisation of a notion introduced under the name of *saliency map* [60].

Ultrametric watersheds The formal definition of ultrametric watershed relies on the topological watershed framework [83].

15

Let X be a graph. An edge $u \in \overline{E(X)}$ is said to be *W-simple (for X)* if X has the same number of connected components as $X + u = (V(X), E(X) \cup \{u\})$.

An edge u such that $F(u) = \lambda$ is said to be *W-destructible (for F) with lowest value λ_0* if there exists λ_0 such that, for all λ_1, $\lambda_0 < \lambda_1 \leq \lambda$, u is W-simple for $G[\lambda_1]$ and if u is not W-simple for $G[\lambda_0]$.

A *topological watershed (on G)* is a map that contains no W-destructible edges.

An *ultrametric watershed* is a topological watershed F such that $F(v) = 0$ for any v belonging to a minimum of F.

There exists a bijection between ultrametric distances and hierarchies of partitions [15]; in other word, to any hierarchy of partitions is associated an ultrametric, and conversely, any ultrametric yields a hierarchy of partitions, see also Sec. 3. Similarly, there exists a bijection between the set of hierarchies of *connected partitions* and the set of ultrametric watersheds [81,82]. In [84], it is proposed a generic algorithm for computing hierarchies and their associated ultrametric watershed.

Usage: gradient and dissimilarity Constrained connectivity is a hierarchy of flat zones of f, in the sense where the 0-connected components of f are the zones of f where the intensity of f does not change. In a continuous world, such zones would be the ones where the gradient is null, *i.e.* $\nabla f = 0$. However, the space we are working with is discrete, and a flat zone of f can consist in a single point. In general, it is not possible to compute a gradient on the points or on the edges such that this gradient is null on the flat zones. To compute a gradient on the edges such that the gradient is null on the flat zones, we need to "double" the graph, for example we can do that by doubling the number of points of V and adding one edge between each new point and the old one.

More precisely, if we denote the points of V by $V = \{p_0, \ldots, p_n\}$, we set $V' = \{p'_0, \ldots, p'_n\}$ (with $V \cap V' = \emptyset$), and $E' = \{\{p_i, p'_i\} \mid 0 \leq i \leq n\}$. We then set $V_1 = V \cup V'$ and $E_1 = E \cup E'$.

By construction, as $G = (V, E)$ is a connected graph, the graph $G_1 = (V_1, E_1)$ is a connected graph. We also extend f to V', by setting, for any $p' \in V'$, $f(p') = f(p)$, where $\{p, p'\} \in E'$.

We set, as in section 5.1, $F(\{p, q\}) = |f(p) - f(q)|$. The map F can be seen as the "natural gradient" of f [85]. We can then apply the same scheme on this F as in section 5.1 to find the hierarchy of α-connected components.

We denote by $L(G_1)$ the edge graph (also called line graph) of G_1. That is, each vertex of $L(G_1)$ represents an edge of G_1 and two vertices of $L(G_1)$ are adjacent if and only if their corresponding edges in G_1 share a common endpoint in G_1. While the edges of $L(G_1)$ are not weighted, the weights of its nodes are given by the weights of the corresponding edges of G_1. It follows that the minima of $L(G_1)$ are equivalent to the 0-connected components of G_1. More generally, the alpha-tree of G_1 is contained in the min-tree of $L(G_1)$. Interestingly, the min-tree of $L(G_1)$ can be computed efficiently thanks to the quasi-linear algorithm described in [86]. Hence, the morphological framework of attribute filtering [87]

can be applied to this min-tree [65,67,66], similarly to the segmentation of an image into k regions proposed in [88]. This is in particular useful when the filtering is performed before computing a watershed and this is illustrated in the next paragraph for the computation of a hierarchy based on constrained connectivity.

Finding the (α, ω)-CCs can be done by filtering the ultrametric watershed W of F with R that acts as a flooding on the topological/ultrametric watershed W of F, and then finding a (topological) watershed of the filtered image. Repeating these steps for a sequence of ordered (α, ω) vectors, we build a constrained connectivity hierarchy. In effect, we are viewing a hierarchy as an image (edge-weighted graph) and transforming it into another hierarchy/image.

Thus, classical tools from mathematical morphology can be applied to constrain any hierarchy. Similar examples exist in the literature, for example [53], where the authors compute what they called a non-horizontal cut in the hierarchy, in other words, they compute a flooding on a watershed. In their framework, the flooding is controlled by an energy.

The advantages of using an ultrametric watershed are numerous. Let us mention the two following ones:

1. an ultrametric watershed is visible. A dendrogram or a component tree can be drawn, but less information is available from such a drawing, and visualising a MST is not really useful;
2. an ultrametric watershed allows the use any information in the contours between regions; such information is not available on the component tree, and is only partially available with a MST (which contains only the pass between regions).

Let us note that those concepts are theoretically equivalent: even their respective computational time is in practice nearly identical; thus we can choose the one the most adapted to the desired usage.

Visualising the hierarchy of constrained connectivity as an ultrametric watershed allows ones to assess some of its qualities. One can notice in Fig. 3.c a large number of transition regions (small undesirable regions that persist in the hierarchy), which is the topic of the next section.

6 Transition pixels

Constrained connectivity prevents the formation of connected components that would otherwise be created in case samples of intermediate value (transition pixels) between two populations (homogeneous image structures) are present. Indeed, these components would violate the global range or other appropriate constraint. However, sometimes the formation of two distinct connected components cannot occur at all. In the extreme case represented in Fig. 4. either each pixel is a connected component (flat zone) or there is a unique connected component. One way to address this problem is to propose a definition of transition pixels and perform some pre-processing to suppress them. This approach

(a) Original image

(b) W^1(logarithmic grey-scale)

(c) W^2

(d) Area-filtering ultrametric watershed
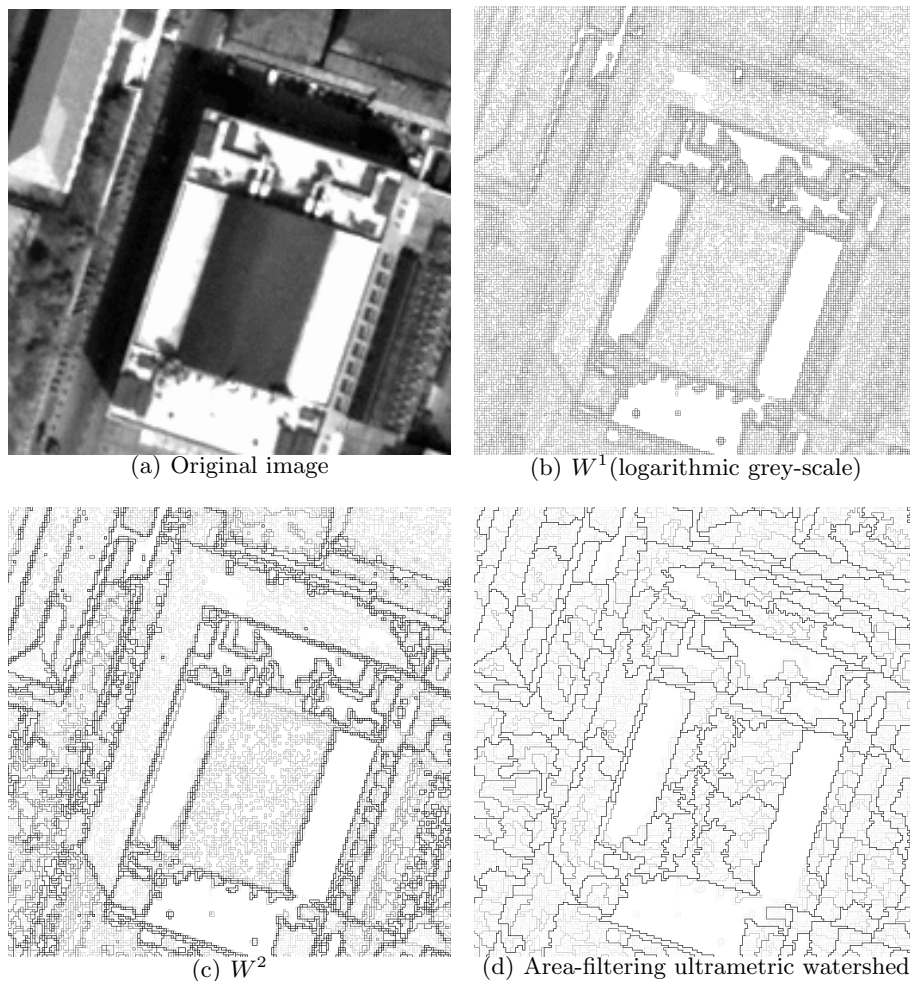
Fig. 3. Constrained connectivity and ultrametric watersheds. (a) Original image (extract from the panchromatic channel of a Quickbird Imagery © DigitalGlobe Inc., 2007, distributed by Eurimage). (b) Ultrametric watershed W^1 for the α-connectivity (the grey level of a contour corresponds to the α value above which the contour disappears in the α-hierarchy. (c) Ultrametric watershed W^2 for the constrained connectivity (the grey level of a contour corresponds to the $\alpha = \omega$ value above which the contour disappears in the $(\alpha, \omega = \alpha)$-hierarchy). (d) Ultrametric watersheds corresponding to one of the possible hierarchies of area-filterings on W^2.

is advocated in [76,80]. For example, assuming that *local* extrema correspond to non-transition pixels, they are extracted on then considered as seeds whose values are propagated in the input image using a seeded region growing algorithm [89]. Note that this approach is linked with contrast enhancement techniques since it

0	1	0	7	8	7	8
1	0	1	6	7	8	7
0	1	0	5	8	7	8
1	0	1	4	7	8	7
0	1	0	3	8	7	8
1	0	1	2	7	8	7
0	1	0	1	8	7	8
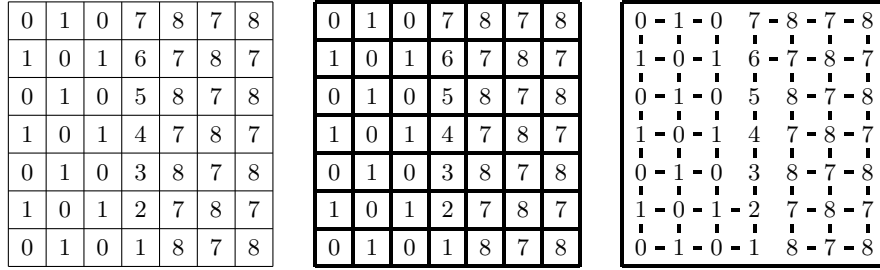
Fig. 4. A synthetic sample image with its intensity values and its two possible partitions into constrained connected components *whatever* the considered constraints in case standard α-connectivity is used in the definitions. The two homogeneous regions show intensity variations of 1 level while the ramp between the two regions also proceeds by steps of 1 intensity level. In the image at the right, adjacent pixels are linked by an edge if and only if their range does not exceed 1.

aims at increasing the external isolation of the obtained connected components. A number of classical morphological schemes (*e.g.*, area filtering of the ultrametric watershed) can be used to remove those transition zones (see Fig. 3.d for an example).

Another approach is to substitute the α-connectivity with a more restrictive connectivity. Indeed, the local range parameter α defined in [37] as the intensity difference between adjacent pixels can be viewed as a special case of dissimilarity measurement. Although this measurement is the most natural, other dissimilarity measurements may be considered. For example, the following alternative definition of alpha-connectivity may be considered to tackle the problem of transition regions. Let the α-degree of a pixel (node) be defined as the number of its adjacent pixels that are within a range equal to α:

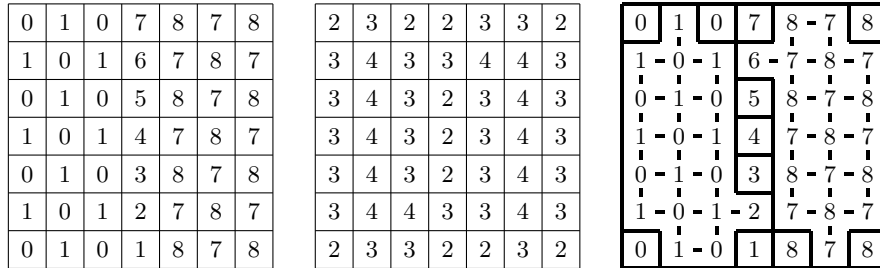$$\alpha^\star\text{-deg}(p) = \#\{q \mid \{p, q\} \in E \text{ and } |f(q) - f(p)| \leq \alpha\}.$$

Then two pixels p and q are said to be α_n-connected if and only if there exists an α-path connecting them such that every pixel of the path has a α-degree greater of equal to n. We obtain therefore the following definition for the α_n-connected component of a pixel p:

$$\alpha_n\text{-CC}(p) = \{p\} \cup \{q \mid \text{ there exists a path } \langle p = p_1, \ldots, p_n = q \rangle, \, n > 1,$$
$$\text{such that } |f(p_i) - f(p_{i+1}| \leq \alpha \text{ and } \alpha^\star\text{-deg}(p_i) \geq n\}.$$

If necessary, other constraints can be considered. Note that α-connectivity is a special case of α_n-connectivity obtained for $n = 1$. In addition, the following nesting property holds:

$$\alpha_{n'}\text{-CC}(p) \subseteq \alpha_n\text{-CC}(p),$$

where $n \leq n'$. α_n-connectivity satisfies all properties of an equivalence relation and therefore also partitions the image definition domain into unique maximal connected components. An example is provided in Fig. 5. In this example,

0	1	0	7	8	7	8
1	0	1	6	7	8	7
0	1	0	5	8	7	8
1	0	1	4	7	8	7
0	1	0	3	8	7	8
1	0	1	2	7	8	7
0	1	0	1	8	7	8

2	3	2	2	3	3	2
3	4	3	3	4	4	3
3	4	3	2	3	4	3
3	4	3	2	3	4	3
3	4	3	2	3	4	3
3	4	4	3	3	4	3
2	3	3	2	2	3	2

0	1	0	7	8 - 7	8
1 - 0 - 1	6 - 7 - 8 - 7				
0 - 1 - 0	5	8 - 7 - 8			
1 - 0 - 1	4	7 - 8 - 7			
0 - 1 - 0	3	8 - 7 - 8			
1 - 0 - 1 - 2	7 - 8 - 7				
0	1 - 0	1	8	7	8

Fig. 5. A synthetic sample image with its intensity values, the corresponding 1-deg map, and 1_3-connected components.

the non singleton 1_3-connected components match the core of the two homogeneous regions. Singleton connected components correspond to pixels whose degree is smaller than 3. Non-singleton connected components can be used as seeds for coarsening the obtained partition. Special care is needed to produce connected components matching one-pixel thick non-transition regions. Alternative approaches to tackle the problem of transition regions are also presented in [73] using a dissimilarity value taking into account the values of the gradient by erosion and dilation at the considered adjacent pixels and in [74] using image statistics.

7 Conclusion and perspectives

In this paper, we have presented several equivalent tools dealing with hierarchies of connected partitions. Such a review invites us to look more closely at links between what have been done in different research domains as, for example, between clustering and lattice theory [90]. A first step in that direction is [91], and there is a need for in-depth study of operators acting on lattices of graphs [92] (or the one of complexes [93]). The question of transition pixels is not only a theoretical one, regarding its significance for applications. Finally, we want to stress the importance of having frame work allowing a generic implementation of existing algorithms, not limited to the pixel framework, but also able to deal transparently with edges, or, more generally, with graphs and complexes [94].

Finally, when dealing with very large images such as those encountered in remote sensing or biomedical imaging, the computation of the min-tree of the edge graph of an image may be prohibitive in terms of memory needs (without mentioning the additional cost of doubling the graph to make sure that each flat zone of the original image is matched by a minimum of the edge graph). In this situation, the direct computation of the alpha-tree of the image may be a valid alternative. An efficient implementation based on the union-find as originally presented for the computation of component trees [86] is presented in [79].

References

1. Köthe, U., Montanvert, A., Soille, P., eds.: Proc. of ICPR Workshop on Applications of Discrete Geometry and Mathematical Morphology. IAPR, Istanbul (August 2010)
2. Matheron, G.: Eléments pour une théorie des milieux poreux. Masson, Paris (1967)
3. Meyer, F., Maragos, P.: Nonlinear scale-space representation with morphological levelings. Journal of Visual Communication and Image Representation **11**(3) (2000) 245–265
4. Cormack, R.: A review of classification (with discussion). Journal of the Royal Statistical Society A **134** (1971) 321–367
5. Estabrook, G.: A mathematical model in graph theory for biological applications. Journal of Theoretical Biology **12** (1966) 297–310
6. Matula, D.: Cluster analysis via graph theoretic techniques. In Mulin, R., Reid, K., Roselle, P., eds.: Proc. Louisiana Conference on Combinatorics, Graph Theory, and Computing, Winnipeg, University of Manitoba (1970) 199–212
7. Zahn, C.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers **C-20**(1) (January 1971) 68–86
8. Hubert, L.: Some applications of graph theory to clustering. Psychometrika **39**(3) (1974) 283–309
9. Hubert, L.: Min and max hierarchical clustering using asymetric similaritly measures. Psychometrika **38** (1973) 63–72
10. Diestel, R.: Graph Theory. Graduate Texts in Mathematics. Springer (1997)
11. Kong, T., Rosenfeld, A.: Digital topology: Introduction and survey. Comput. Vision Graph. Image Process. **48**(3) (1989) 357–393
12. Spærck Jones, K.: Some thoughts on classification for retrieval. Journal of Documentation **26**(2) (1970) 571–581
13. Jardine, N., Sibson, R.: A model for taxonomy. Mathematical Biosciences **2**(3–4) (1968) 465–482
14. Barthélemy, J.P., Brucker, F., Osswald, C.: Combinatorial optimization and hierarchical classifications. 4OR: A Quaterly Journal of Operations Research **2**(3) (2004) 179–219
15. Johnson, S.: Hierarchical clustering schemes. Psychometrika **32**(3) (September 1967) 241–254
16. Sokal, R., Sneath, P.: Principles of Numerical Taxonomy. W. H. Freeman and Company, San Fransisco and London (1963)
17. Sneath, P.: The application of computers in taxonomy. Journal of general Microbiology **17** (1957) 201–226
18. Hartigan, J.: Representation of similarity matrices by trees. American Statistical Association Journal (1967) 1140–1158
19. Jardine, C., Jardine, N., Sibson, R.: The structure and construction of taxonomic hierarchies. Mathematical Biosciences **1**(2) (1967) 173–179
20. Benzécri, J.P.: L'analyse des données. Volume 1 La taxinomie. Dunod, Paris (1973)
21. Sørensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its applications to analyses of the vegetation of Danish commons. Biologiske Skrifter **5**(4) (1948) 1–34
22. Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S.: Sur la liaison et la division des points d'un ensemble fini. Colloquium Mathematicum **2** (1951) 282–285

23. Gower, J., Ross, G.: Minimum spanning trees and single linkage cluster analysis. Applied Statistics **18**(1) (1969) 54–64
24. Kruskal, J.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society **7**(1) (February 1956) 48–50
25. Boruvka, O.: O jistém problému minimálním [On a certain minimal problem]. Acta Societatis Scientiarum Naturalium Moravicae **III**(3) (1926) 37–58
26. Graham, R., Hell, P.: On the history of the minimum spanning tree problem. Ann. History Comput. **7**(1) (January 1985) 43–57
27. Wishart, D.: Mode analysis: a generalization of nearest neighour which reduced chain effect. In Cole, A., ed.: Numerical taxonomy. Academic Press, New York (1968) 282–311
28. Jardine, N., Sibson, R.: The construction of hierarchic and non-hierarchic classifications. The Computer Journal **11** (1968) 177–184
29. Horowitz, S., Pavlidis, T.: Picture segmentation by a directed split-and-merge procedure. In: Proc. Second Int. Joint Conf. Pattern Recognition. (1974) 424–433
30. Zucker, S.: Region growing: childhood and adolescence. Computer Graphics and Image Processing **5** (1976) 382–399
31. Jardine, N., Sibson, R.: Mathematical Taxonomy. Wiley, London (1971)
32. Rosenfeld, A.: Fuzzy digital topology. Information and Control **40** (1979) 76–87
33. Soille, P.: Morphological partitioning of multispectral images. Journal of Electronic Imaging **5**(3) (July 1996) 252–265
34. Ahuja, N.: On detection and representation of multiscale low-level image structure. ACM Computing Surveys **27**(3) (September 1995) 304–306
35. Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision **24**(1) (2006) 83–130
36. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision **32**(2) (oct. 2008) 97–105
37. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(7) (July 2008) 1132–1145
38. Horowitz, S., Pavlidis, T.: Picture segmentation by a tree traversal algorithm. Journal of the ACM **23**(2) (April 1976) 368–388
39. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing **10**(3) (July 1979) 195–223
40. Nagao, M., Matsuyama, T.: A Structural Analysis of Complex Aerial Photographs. Plenum, New York (1980)
41. Baraldi, A., Parmiggiani, F.: Single linkage region growing algorithms based on the vector degree of match. IEEE Transactions on Geoscience and Remote Sensing **34**(1) (January 1996) 137–148
42. Morris, O., Lee, M., Constantinides, A.: Graph theory for image analysis: an approach based on the shortest spanning tree. IEE proceedings **133**(2) (1986) 146–152
43. Meyer, F., Maragos, P.: Morphological scale-space representation with levelings. Lecture Notes in Computer Science **1682** (September 1999) 187–198
44. Nacken, P.: Image segmentation by connectivity preserving relinking in hierarchical graph structures. Pattern Recognition **28**(6) (1995) 907–920
45. Kropatsch, W., Haxhimusa, Y.: Grouping and segmentation in a hierarchy of graphs. In Bouman, C., Miller, E., eds.: Proc. of the 16th IS&T SPIE Annual Symposium, Computational Imaging II. Volume SPIE-5299. (May 2004) 193–204

46. Felzenszwalb, P., Huttenlocher, D.: Image segmentation using local variations. In: Proc. of IEEE Int. Conf. on Comp. Vis. and Pat. Rec. (CVPR). (1998) 98–104

47. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based segmentation. IJCV **59**(2) (2004) 167–181

48. Wu, Z., Leahy, R.: An optimal graph-theoretic approach to data clustering: theory and its applications to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **15**(11) (November 1993) 1101–1113

49. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (August 2000) 888–905

50. Marfil, R., Molina-Tanco, L., Bandera, A., Rodriguez, J., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition **39**(8) (2006) 1430–1451

51. Kropatsch, W.G., Haxhimusa, Y., Ion, A.: Multiresolution image segmentations in graph pyramids. Studies in Computational Intelligence **52** (2007) 3–41

52. Guigues, L., Le Men, H., Cocquerez, J.P.: The hierarchy of the cocoons of a graph and its application to image segmentation. Pattern Recognition Letters **24**(8) (2003) 1059–1066

53. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. IJCV **68**(3) (2006) 289–317

54. Arbeláez, P., Cohen, L.: Energy partition and image segmentation. Journal of Mathematical Imaging and Vision **20** (2004) 43–57

55. Arbeláez, P.: Boundary extraction in natural images using ultrametric contour maps. In: Proc. of Computer Vision and Pattern Recognition Workshop, Los Alamitos, CA, USA, IEEE Computer Society (2006)

56. Beucher, S.: Segmentation d'images et morphologie mathématique. PhD thesis, Ecole des Mines de Paris (June 1990)

57. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In Serra, J., Soille, P., eds.: Mathematical Morphology and its Applications to Image Processing. Kluwer Academic Publishers (1994) 69–76

58. Beucher, S., Meyer, F.: The morphological approach to segmentation: the watershed transformation. In Dougherty, E., ed.: Mathematical Morphology in Image Processing. Volume 34 of Optical Engineering. Marcel Dekker, New York (1993) 433–481

59. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6) (June 1991) 583–598

60. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(12) (1996) 1163–1173

61. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: 10th International Symposium on Mathematical Morphology (ISMM'11). LNCS (2011)

62. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: thinnings, shortest-path forests and topological watersheds. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(5) (May 2010) 925–939

63. Meyer, F.: Minimum spanning forests for morphological segmentation. In Serra, J., Soille, P., eds.: Mathematical Morphology and its Applications to Image Processing. Kluwer Academic Publishers (1994) 77–84

64. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing **9**(4) (April 2000) 561–576

65. Jones, R.: Component trees for image filtering and segmentation. In Coyle, E., ed.: Proc. of IEEE Workshop on Nonlinear Signal and Image Processing, Mackinac Island (September 1997)
66. Jones, R.: Connected filtering and segmentation using component trees. Comput. Vis. Image Underst. **75**(3) (1999) 215–228
67. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing **7**(4) (April 1998) 555–570
68. Meyer, F.: An overview of morphological segmentation. International Journal of Pattern Recognition and Artificial Intelligence **15**(7) (November 2001) 1089–1118
69. Meyer, F., Najman, L.: Segmentation, minimum spanning tree and hierarchies. In Najman, L., Talbot, H., eds.: Mathematical morphology: from theory to applications. Wiley-ISTE (2010) 255–287
70. Salembier, P., Wilkinson, M.: Connected operators: A review of region-based morphological image processing techniques. IEEE Signal Processing Magazine **26**(6) (November 2009) 136–157
71. Salembier, P.: Connected operators based on tree pruning strategies. In Najman, L., Talbot, H., eds.: Mathematical Morphology: from theory to applications. Wiley-ISTE (2010) 205–221
72. Soille, P.: On genuine connectivity relations based on logical predicates. In: Proc. of 14th Int. Conf. on Image Analysis and Processing, Modena, Italy, IEEE Computer Society Press (September 2007) 487–492
73. Soille, P.: Preventing chaining through transitions while favouring it within homogeneous regions. Lecture Notes in Computer Science **6671** (2011) 96–107
74. Gueguen, L., Soille, P.: Frequent and dependent connectivities. Lecture Notes in Computer Science **6671** (2011) 120–131
75. Soille, P.: Advances in the analysis of topographic features on discrete images. Lecture Notes in Computer Science **2301** (March 2002) 175–186
76. Soille, P., Grazzini, J.: Constrained connectivity and transition regions. Lecture Notes in Computer Science **5720** (2009) 59–69
77. Soille, P., Vincent, L.: Determining watersheds in digital pictures via flooding simulations. In Kunt, M., ed.: Visual Communications and Image Processing'90. Volume 1360., Bellingham, Society of Photo-Instrumentation Engineers (1990) 240–250
78. Ouzounis, G., Soille, P.: Pattern spectra from partition pyramids and hierarchies. Lecture Notes in Computer Science **6671** (2011) 108–119
79. Ouzounis, G., Soille, P.: Attribute-constrained connectivity and alpha-tree representation. IEEE Transactions on Image Processing (2011)
80. Soille, P.: Constrained connectivity for the processing of very high resolution satellite images. International Journal of Remote Sensing **31**(22) (July 2010) 5879–5893
81. Najman, L.: Ultrametric watersheds. In Wilkinson, M., Roerdink, J., eds.: Proc. ISMM 2009. Volume 5720 of Lecture Notes in Computer Science. (2009) 181–192
82. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. Journal of Mathematical Imaging and Vision **40**(3) (July 2011) 231–247
83. Bertrand, G.: On topological watersheds. J. Math. Imaging Vis. **22**(2-3) (2005) 217–230
84. Cousty, J., Najman, L.: Incremental algorithms for hierarchical minimum spanning forests and saliency of watershed cuts. In Soille, P., Pesaresi, M., Ouzounis, G., eds.: Proc. of ISMM 2011. Volume 6671 of Lecture Notes in Computer Science., Springer-Verlag (2011) 271–283

85. Mattiussi, C.: The Finite Volume, Finite Difference, and Finite Elements Methods as Numerical Methods for Physical Field Problems. Advances in Imaging and Electron Physics **113** (2000) 1–146

86. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing **15**(11) (2006) 3531–3539

87. Breen, E., Jones, R.: Attribute openings, thinnings, and granulometries. Comput. Vis. Image Underst. **64**(3) (1996) 377–389

88. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: minimum spanning forests and the drop of water principle. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(8) (August 2009) 1362–1374

89. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(6) (1994) 641–647

90. Hubert, L.: Some extension of Johnson's hierarchical clustering. Psychometrika **37** (1972) 261–274

91. Cousty, J., Najman, L., Serra, J.: Raising in watershed lattices. In: 15th IEEE ICIP'08, San Diego, USA (October 2008) 2196–2199

92. Cousty, J., Najman, L., Serra, J.: Some morphological operators in graph spaces. In: ISMM 2009. Volume 5720 of LNCS., Springer-Verlag (August 2009) 149–160

93. Dias, F., Cousty, J., Najman, L.: Some Morphological Operators on Simplicial Complex Spaces. In Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P., eds.: 16th Discrete Geometry for Computer Imagery (DGCI'11). Volume 6607 of Lecture Notes in Computer Science., Nancy, Springer-Verlag (April 2011) 441–452

94. Levillain, R., Géraud, T., Najman, L.: Writing reusable digital topology algorithms in a generic image processing framework. Volume [This volume] of Lecture Notes in Computer Science. (2012)