An overview of max-tree principles, algorithms and applications

Roberto Souza, Luís Tavares, Letícia Rittner, Roberto Lotufo
Department of Computer Engineering and Industrial Automation
School of Electrical and Computer Engineering - FEEC
University of Campinas - UNICAMP
Campinas, Brazil
{roberto.medeiros.souza, xtluis, lrittner, robertoalotufo}@gmail.com

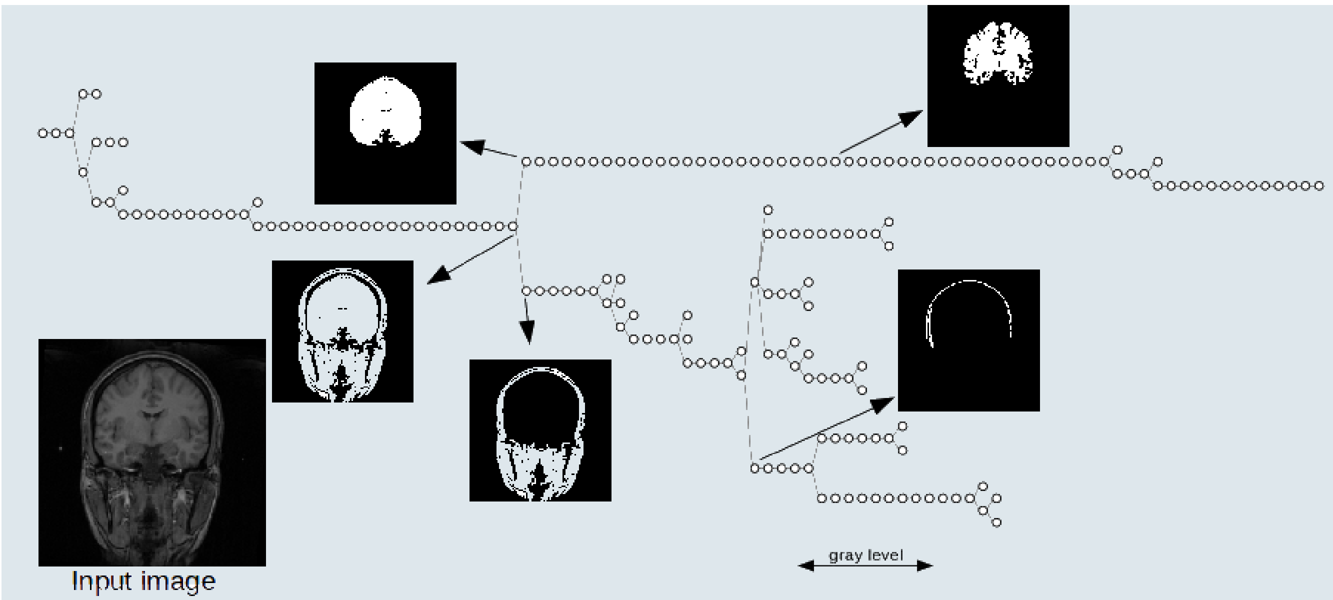


Fig. 1. Simplified illustration of a max-tree. The arrows point to the connected components that each node represents.

***Abstract*—The max-tree is a mathematical morphology data structure that represents an image through the hierarchical relationship of connected components resulting from different thresholds. It was proposed in 1998 by Salembier et al., since then, many efficient algorithms to build and process it were proposed. There are also efficient algorithms to extract size, shape and contrast attributes of the max-tree nodes. These algorithms allowed efficient implementation of connected filters like attribute-openings and development of automatic and semi-automatic applications that compete with the state-of-the-art. This paper reviews the max-tree principles, algorithms, applications and its current trends.**

***Keywords*-max-tree; component tree; filtering; algorithms;**

I. Introduction

Connected operators [1] are morphological filters that do not blur the image. They act by merging flat-zones [1] in the image and can be efficiently implemented on hierarchical region-based representations of the input image [2]. Among these representations, the max-tree [3] is arguably the most popular. Some authors consider it as being a synonym of the component tree [4]. Although both structures allow to do the same operations and analysis, they are different in their definition. The max-tree can be seen as an efficient and compact representation of the component tree [5].

Max-trees can be computed, and processed fast [6], [7], [8], [9], [10]. The best algorithm depends on the image data type and the number of cores available for processing. There are many different structures used to represent a max-tree: array-based [6], [7], array-based node-oriented [10], and linked-list node-oriented [11].

Max-trees allow the efficient implementation of many connected filters like area-open [12], hmax [3], ultimate opening [13], statistical attribute filters [14], vector attribute filters [15], among others. They have been used in a wide range of automatic applications like object recognition [5], scale and rotation invariant image classification [16], detection, tracking and recognition of license plates [17], recognition of text in natural scenes [18], 3D magnetic resonance (MR) brain segmentation [19], angiographic image segmentation [20], dermatological image segmentation [21], medical image registration [22], remote sensing [23], [24], [25], [26], detection of local features like Maximally Stable Extremal Regions (MSER) [27] and Morse Regions [28], among many others.

The max-tree has also been used in interactive applications. Westenberg et al. [29] proposed an interactive volumetric filtering and visualization of 3D images using the max-tree. Ouzounis and Gueguen [30] proposed a technique for interactive collection of training samples from the max-tree nodes. Naegel and Passat [31] proposed an interactive segmentation method, where the user selects regions of interest in the image and the method returns the set of max-tree nodes corresponding to the segmentation result. In the interactive methods previously cited, the interaction occurs by setting filtering parameters or by selecting regions in the input image. The work by Tavares et al. [32] was the first interactive method, where the interaction occurs in the max-tree domain. They provided a graphical representation of the max-tree, where the user is able to navigate and select nodes to perform interactive segmentation, filtering and collection of training samples. In a subsequent work, Tavares et al. [33] extended their interactive method to work with 3D images and to use a color map to display attribute values of the nodes in their max-tree graphical representation.

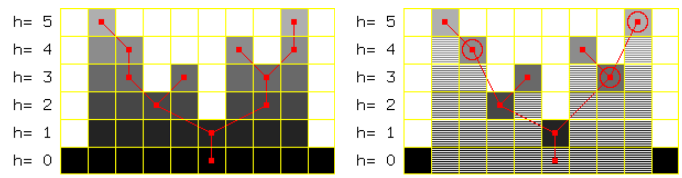
This paper reviews the max-tree literature, pointing to interesting works and analyzing current trends. The paper is organized as follows: Section II reviews the max-tree theory and its difference to the component tree. Section III discusses the max-tree construction and processing algorithms. Section IV describes max-tree applications. Section V presents the current research trends in the field. Finally, Section VI concludes the paper.

II. Theory

A. Max-tree vs component tree

Informally speaking, suppose we have a gray-scale image. An upper threshold ($\geq$) results in a binary image, where each "white island" (value 1) is called a connected component [1] (CC). The higher the threshold value, the smaller the size of the CCs, *i.e.* there is an inclusion relationship between the connected components obtained at different threshold values. The component tree represents an image through this hierarchical property of threshold decomposition. For every CC resulting of every possible threshold of the image, there is a component tree node and this node stores all the pixels of the connected component. This results in a redundant representation, since most pixels will belong to more than one node. Also, there are CCs that do not change from one threshold to the other, therefore being redundant to use more than one node to represent them. The max-tree is a compact representation for the component tree, its nodes store only the pixels that are visible in the image at that corresponding gray-level, therefore CCs that stayed the same for a sequence of thresholds are represented in a single node called composite node [5]. We think in terms of the component tree when designing methods and applications, but we process using the max-tree, since it is a more compact and efficient structure. That is the reason why many authors treat component trees and max-trees as being synonyms.

The component tree and max-tree representations for a 1D image $I = [0, 5, 4, 2, 3, 1, 4, 3, 5, 0]$ are depicted in Fig. 2. The composite nodes are represented by double circles and the filled pixels in each horizontal CC represent the pixels each node stores. For a more formal definition of max-trees and component trees, the readers are referred to [3], [4], [5], [35].



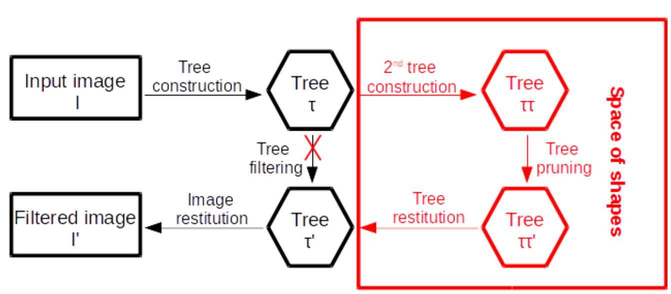
(a) (b)

Fig. 2. (a) Component tree and (b) max-tree of the 1D image $I = [0, 5, 4, 2, 3, 1, 4, 3, 5, 0]$.

B. Filtering

It is possible to make an analogy between max-trees and reversible transforms, like the Fourier Transform. When filtering an image using a reversible transform, the first step is to compute the the transform of the signal. Then, in the transform domain different operations like convolutions and multiplications are performed. Finally, the inverse transform gives the filtered image.

Filtering an image using max-trees is the same. First, the image max-tree is built using the appropriate connectivity rule. Then, some of its nodes are removed (contracted) according to the filtering criteria, usually size and shape criteria are used. From the filtered max-tree representation, the filtered image is reconstructed. The max-tree filtering pipeline is represented by the black path in Fig. 3. One difference between filtering images with max-trees and using image transforms is that the image reconstruction from the max-tree is considerably faster than building the max-tree (see Table I), while using image transforms like the Fourier Transform, the times to go to the transform domain and back to the image domain are similar.



(a)

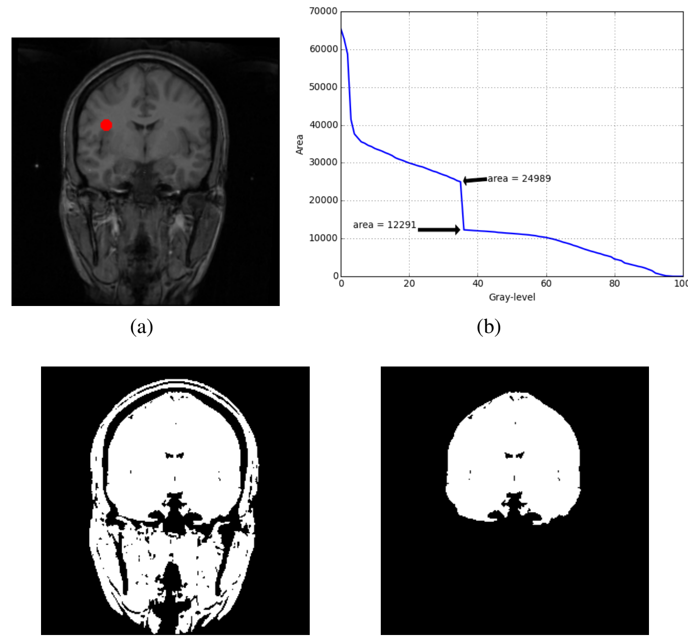
Fig. 3. Max-tree filtering pipeline (black path) and space of shapes filtering pipeline (black + red paths).

1 Actually the "island" definition depends on the connectivity rule [34].

The max-tree filters are based on the attributes that can be extracted from its nodes, like size, contrast and shape attributes. Filtering an image using a max-tree is a connected anti-extensive filter [3]. In order to obtain extensive filters, the min-tree should be employed, which consists in the same idea of the max-tree, but it is based on a lower threshold decomposition. Equivalently, the duality property [36] can be employed to obtain extensive filters, *i.e.* build the max-tree of the negative of the image.

There are two rules for filtering the max-tree: the direct rule [3] and the subtractive rule [37]. The direct rule preserves the gray-level of the descendants of the nodes removed, while the subtractive rule lowers the gray-level of the descendants of the nodes removed, preserving image contrast.

There are also two strategies for filtering the max-tree: pruning and non-pruning strategies. Pruning strategies usually result of thresholding an increasing attribute, such as area, and non-pruning strategies result from thresholding a non-increasing attribute, such as most shape attributes like eccentricity and circularity. Some authors consider non-pruning strategies unstable, therefore when dealing with a non-increasing criterion many authors use the *min*, *max* or *Viterbi* algorithms [3] to transform the non-pruning filter into a pruning. Other authors prefer to use non-pruning strategies in order to simplify more the max-tree, and still preserve its topology [5].



(a) (b)

(c) (d)

Fig. 4. (a) Original image. The red dot corresponds to a regional maximum (max-tree leaf). (b) Area signature. (c) Node reconstruction before the sudden drop in the area signature value and (d) node reconstruction after.

C. Max-tree attribute signature analysis

One important concept related to the max-tree is the attribute signature proposed by Jones [4]. The max-tree signature consists in analyzing an attribute variation starting at a leaf node and going towards the root, composite nodes must be considered in this analysis. Signature analysis is a powerful tool to inspect how the connected components evolve when you go through many threshold values. For instance, when there is a sudden change in the area signature value, it may represent a connected component that split in two or more significant connected components. A connected component whose area remains practically unchanged for a sequence of thresholds is a MSER region. An example using a brain image showing where the brain and the scalp split is displayed in Fig. 4. The CC immediately before the sudden change in the signature (Fig. 4(c)) has an area of 24989 pixels and it represents the brain and the scalp connected, while the CC immediately after the sudden change (Fig. 4(d)) has an area of 12291 pixels and it represents the brain after disconnecting from the scalp.

Any attribute can be used used in signature analysis. There are many size, shape and energy attributes that can be efficiently computed from the max-tree [38], [10]. Also, the combination of signatures of different attributes can be used for more complex analysis.

III. Algorithms

There are three groups of max-tree construction algorithms: immersion algorithms [39], [9], flooding algorithms [3], [40] and merge-based algorithms [6], [30]. Carlinet and Géraud [7] presented a comparative review with the state-of-the-art algorithms. They proposed a decision tree to choose the best algorithm based on image quantization, memory limitation and availability of multiple cores in the system. In order to compare the different algorithms, they standardized their outputs, *i.e.*, the max-tree representation, to be the same. Instead of simply using an array that stores the parent relationship, they propose to use two arrays: one to store the parent relationship and a sorting array that allows to do tree traversals from the root to the leaves and vice versa more easily. These data structures used to represent the max-tree can be seen as a pixel oriented max-tree representation, since the max-tree nodes, called level-roots, cannot be accessed directly. The pixel oriented max-tree representation of a 3×4 image is depicted in Fig. 5. The tree has seven nodes and it was built using a 8-connectivity rule.

More recently, Souza et al. [10] proposed a data structure that also uses two arrays called node index and node array to represent the max-tree. Node index is an array with the same shape as the image that tells to which node each pixel belongs and node array encodes the parent relationship of the tree and the gray level of each node. The nodes are identified by the line (or column) index used to access node array. Node array has the property that if traversing from the first line (column) towards the last line (column) of the array, the ancestors of the node being processed will already have been processed. The node array/node index structure is node-oriented, i.e. provides direct access to the max-tree nodes, and is more memory

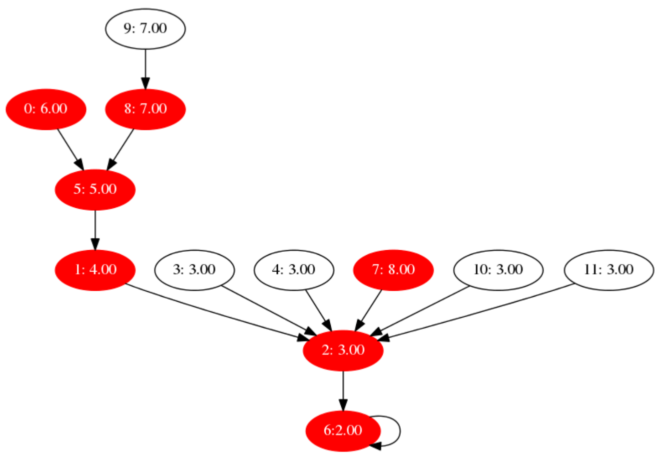
Image

0:6	1:4	2:3	3:3
4:3	5:5	6:2	7:8
8:7	9:7	10:3	11:3

(a)

5	2	6	2	2	1	6	2	5	8	2	2	**Parent**
6	2	11	10	4	3	7	1	5	8	0	9	**S**

(b)



(c)

Fig. 5. (a) Original image (pixel ID: gray-value). (b) Parent and sorting (S) arrays with the level-roots highlighted in red. (c) Pixel oriented max-tree graphical representation.

Node array

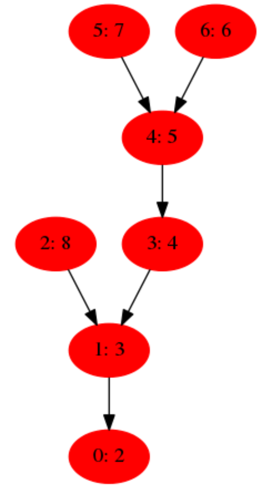
0	2
0	3
1	8
1	4
3	5
4	7
4	6

parent gray-level

Node index

6	3	1	1
1	4	0	2
5	5	1	1

(a)



(b)

Fig. 6. (a) Node array and node index max-tree representation of the image in Fig. 5(a). (b) Node-oriented max-tree graphical representation.

efficient than the representation used by Carlinet and Géraud [7] for storing 8-bit images as long the ratio between the number of image pixels and the number of max-tree nodes is larger than 1.6, which is usually true in practical cases [10]. The node-oriented max-tree of the image in Fig. 5(a) is illustrated in Fig. 6.

There are also max-tree representations based on linked lists [11], but they have not gained much popularity in the mathematical morphology community, probably due the fact that it is harder to manage linked lists than arrays.

Filtering the max-tree is very fast, when compared to the time necessary to build it. There are three main algorithms for filtering the max-tree. Wilkinson et al. [6] proposed a parallel algorithm that uses only the parent array structure. Carlinet and Géraud [7] proposed a sequential algorithm that uses the parent and the sorting arrays to traverse the tree from the root to the leaves and perform the filtering. They say that their algorithm is faster and simpler than [6], but without experimental evidence. Souza et al. [10] proposed a parallel filtering algorithm that uses the node array and node index structures. Their experiments showed a superior performance compared to [7], specially when performing successive filtering steps.

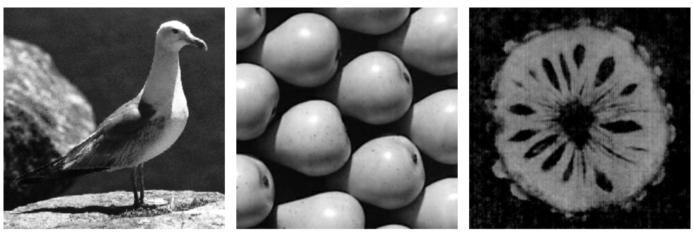
Another subject of interest is the efficient extraction of attributes from max-tree nodes. Xu et al. [38] proposed efficient algorithms for incrementally computing information on region, contour, and context. Bertrand [41] proposed an efficient algorithm to compute the dynamics [42]. Silva and Lotufo [43] proposed a generalization of Bertrand's algorithm to compute extinction values [44], which is a generalization of the dynamics, on the max-tree.

There are a few max-tree libraries. The SDC morphology toolbox for MATLAB and Python [36] has a limited number of max-tree processing functions. It is not public and the source code is not available. The quasi-linear component tree [9] with bindings for MATLAB has only the max-tree construction algorithm implemented, no max-tree filters and feature extraction methods are available. Carlinet and Géraud [7] provide the code of many max-tree computation algorithms. The Milena implementation [45] in C++ with Python bindings is not focused on max-trees, but its advantage is that it works with generic data structures and not just images. The PINK image processing library [46] was developed for research and teaching purposes. It contains implementations of over 200 algorithms but it is not focused on max-trees, but mathematical morphology in general. More recently, the iamxt toolbox[2] was proposed. It was written using an array programming style using Python in association with the Numpy [47] library and C++. This toolbox is devoted solely to max-trees and it has the implementation of many filters, attribute extraction and visualization routines.

In order to illustrate the order of magnitude of the max-tree processing time, we used the iamxt toolbox implementation. The toolbox uses the node array/node index structure and the max-tree construction implemented in the toolbox is the sequential union-find with level compression algorithm [7]. The filtering and image reconstruction algorithms are the

[2]https://github.com/rmsouza01/iamxt

ones described in [10]. The measurements were performed on a 4-core virtual machine running in the Intel Xeon X5675 server with clock of 3.06 GHz. We used the three 256×256 pixels sample images shown in Fig. 7. We measured the average times for building the max-tree using a 8-neighborhood connectivity rule, filtering the max-tree using an area-open filter set to remove nodes with area smaller than 500, and reconstructing the image. The images were interpolated up to 1280×1280 pixels. The average processing times are summarized in Table I. We can see that for 1 megapixel images the total processing time is close to 0.2 seconds, and more than 90% of the total processing time is related to the max-tree construction, which could be improved even further by using a parallel implementation if multiple cores are available.



(a) (b) (c)

Fig. 7. Sample images.

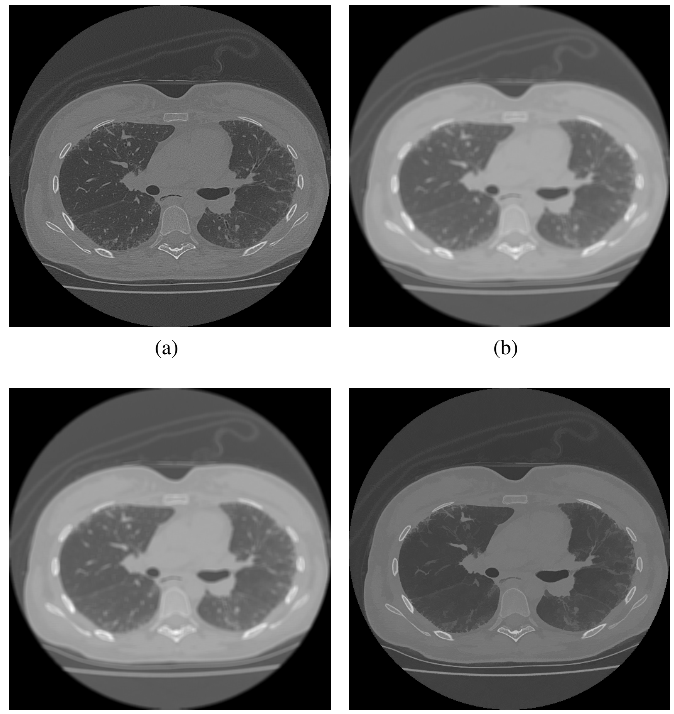
TABLE I
AVERAGE MAX-TREE CONSTRUCTION, FILTERING AND IMAGE RESTITUTION PROCESSING TIMES IN MILLISECONDS.

Dimensions	Construction	Filtering	Restitution	Total
256×256	11.71	9.14	0.12	20.97
512×512	47.26	9.35	0.42	57.03
768×768	111.10	10.23	0.93	122.26
1024×1024	198.98	9.49	1.56	210.03
1280×1280	310.42	9.56	2.64	322.62

IV. APPLICATIONS

The fact that the max-tree can be computed efficiently combined with its information concerning size and shape of image structures allow the development of many applications. Entire applications can be build using almost solely the max-tree or it can be used in combination with other methods in a larger pipeline. It is specially well suited for medical image segmentation problems, where intervals of the size and a model of the shape of the structures of interest are usually known. Also, because filtering images using the max-tree does not change the contours of the structures in the image, which is a very important feature for medical applications. In this section, we illustrate some possibilities of using max-trees in real applications. For applications detailed in more depth, see the references listed in Section I.

A basic use of max-trees is image filtering. They are better suited than classical filters like mean and median filters, because they do not blur the image. A lung computer tomography (CT) image and the result of applying mean, median and area-open filters are depicted in Fig. 8. It is visible that the mean and median filters cause considerable blurring to the image, while the area-open implemented on the max-tree filters the image without producing that undesirable effect. The structural similarity index [48] between the original image and the filtered with the mean filter is 0.88, with the median filter is 0.90, and with the area-open is 0.99. The closest to 1 the higher is the structural similarity between the images.



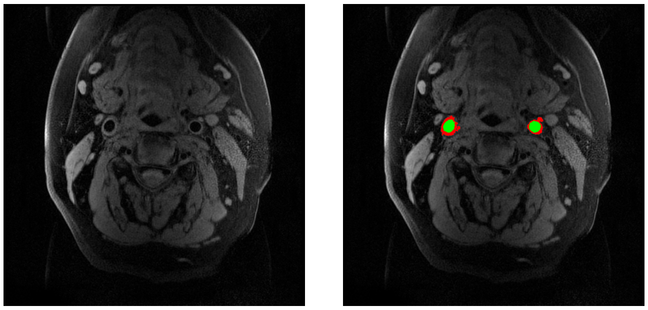
(a) (b) (c) (d)

Fig. 8. (a) Lung CT image. (b) Mean filter and (c) median filter using a 7×7 window. (d) Area-open removing structures with area smaller than 25.

Another important use of max-trees filtering is segmentation. For example, we can use contrast, size and shape criteria to filter all structures in the image, but the ones we are interested. For instance, the carotid artery is usually imaged using MR systems. The shape of the carotid is circular and its diameter range is known. The segmentation of the carotid artery wall and its interior can be done solely through max-tree filtering steps. A carotid wall and its interior segmentation is depicted in Fig. 9. This segmentation was obtained solely by filtering the max-tree for the artery wall and the min-tree for its interior using a combination of height extinction filter [49], bounding-box and circularity filtering.

Another max-tree application is its use to select nodes to be provided as input markers for a segmentation technique, such as the Watershed Transform [50]. For instance, based on average values and standard-deviations of the brain white-matter volume and its shape description, it is possible to select a max-tree node that corresponds to most of the brain white-matter (Fig. 10). This node can be used as a marker for brain segmentation.

Another application of max-trees is vessel segmentation in



(a) (b)

Fig. 9. (a) Carotid MR image. (b) Segmentation result using size and shape filtering. Carotid wall shown in red and its interior in green.

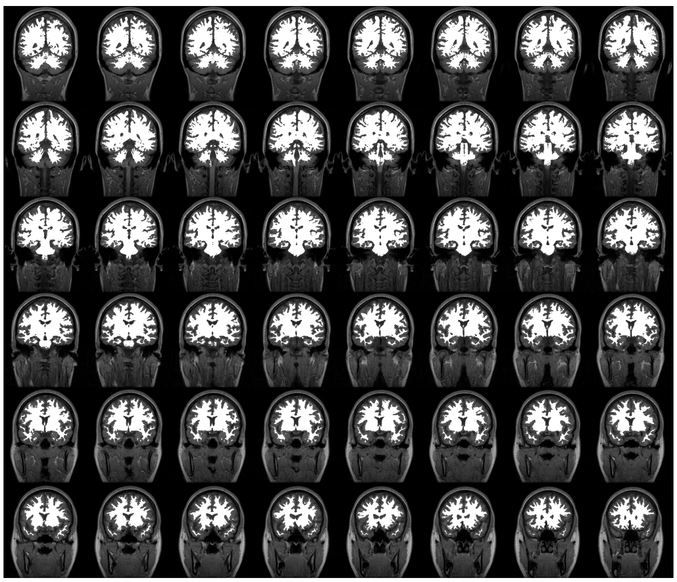


Fig. 10. Brain MR and max-tree node corresponding to the brain white-matter overlaid in white. Only some slices of the volume are displayed.

images of retina. A supervised classifier can be trained to tell which max-tree nodes correspond to vessels, and which nodes do not. The composition of the vessels' nodes corresponds to the final segmentation. A retinal image from the DRIVE dataset [51] and its vessel segmentation using the methodology described is depicted in Fig. 11.

One drawback of applying classification methods directly to the max-tree nodes is that max-trees corresponding to natural images may have tens of thousand of nodes making the classification procedure slow. The max-tree graphical representation of the retina image in Fig. 11(a) is depicted in Fig. 12 using the Scalable Force Directed Placement (SFDP) algorithm [52] to position all the 32938 nodes of this max-tree.

In order to simplify the max-tree without losing relevant information and preserving its topology, the maximal max-tree simplification (MMS) methodology [5] may be employed. It consists in applying an extinction filter followed by the maximal max-tree simplification filter. This methodology guarantees that at the end the number of max-tree nodes is bounded

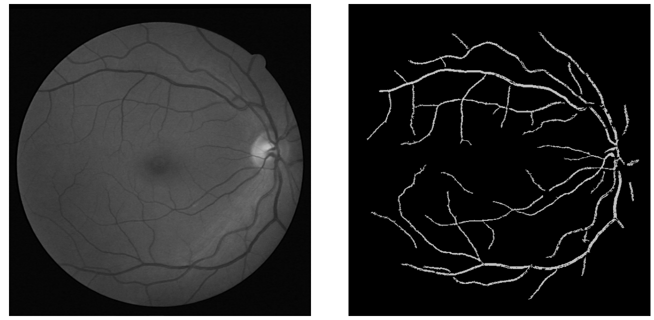


Fig. 11. (a) Retinal image and (b) max-tree segmentation result.

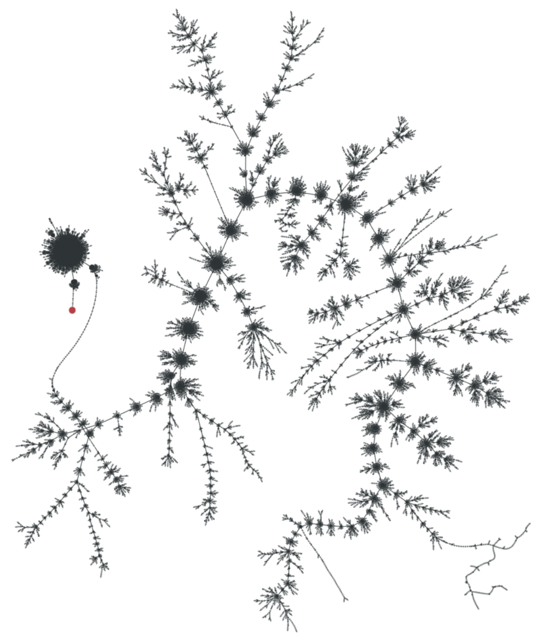
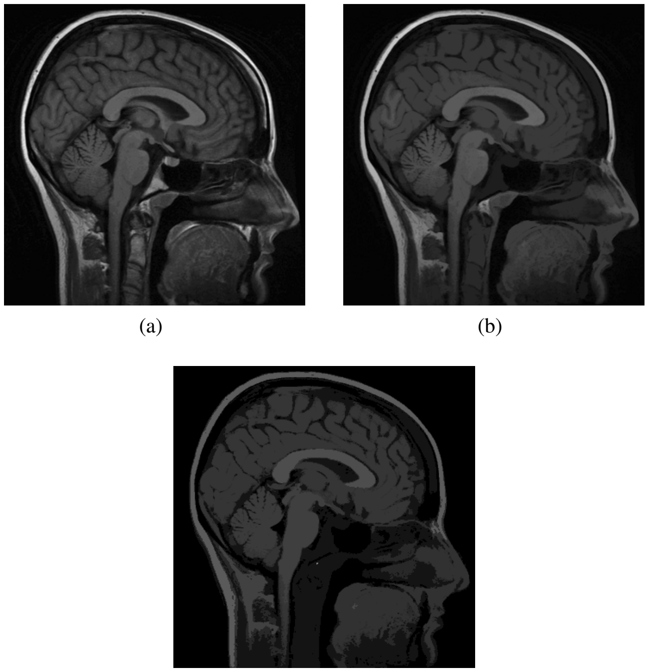


Fig. 12. Max-tree graphical representation of the retina image in Fig. 11(a). The red node is the max-tree root.

between the number of tree leaves plus one and two times the number of leaves. The application of this methodology to a sagittal brain MR slice is illustrated in Fig. 13. The original max-tree has 14312 nodes, after the area extinction filter set to preserve 8 leaves, the tree has 614 nodes. After the MMS filter, the resulting max-tree has only 16 nodes and many relevant structures like scalp, brain stem, cerebellum, and the corpus callosum are still present in the image as can be seen in Fig. 14. This methodology is very useful for simplifying the max-tree, so its nodes can be used in association with machine learning techniques faster than using the entire set of nodes of the original max-tree.

Another option is to display a simplified max-tree for processing purposes. For instance, Fig. 15 illustrates the layout of the interactive max-tree visualization tool proposed by Tavares et al. [33]. It displays a reduced number of max-tree nodes, but through features for zooming specific portions of



(a) (b) (c)

Fig. 13. (a) Original image. (b) Result of the area extinction filter set to preserve 8 leaves. (c) Result of the MMS filter.

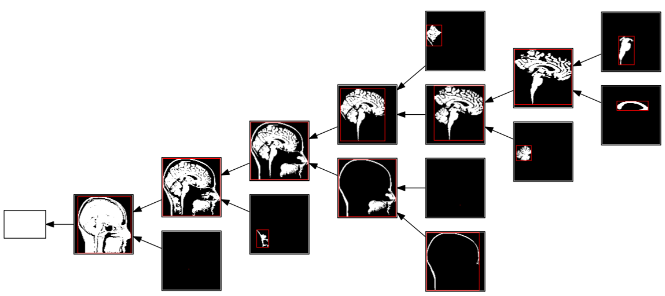
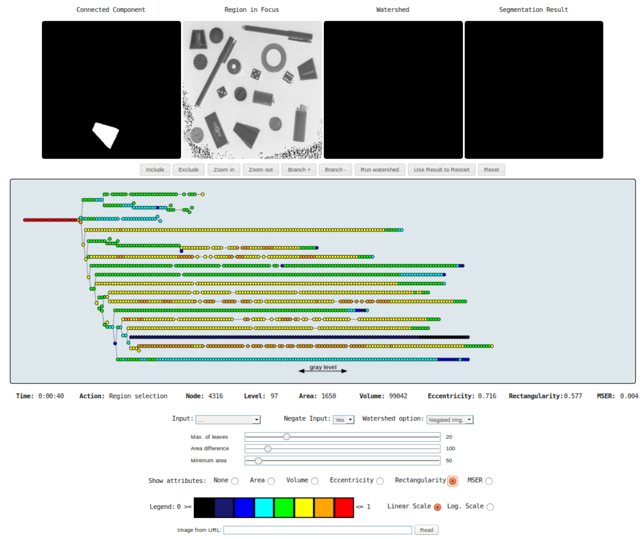


Fig. 14. Max-tree after the MMS methodology.

the tree, the user is able to access all max-tree nodes. The tool also implements an attribute color map, that the user can use as a guide for selecting the most relevant nodes for the problem at hand. This max-tree interactive tool is useful for semi-automatic image segmentation. The user can select a set of nodes to be the segmentation or select nodes to be used as input markers for another segmentation technique. A demo of this tool is available at http://adessowiki.fee.unicamp.br/adesso/wiki/code/MainPage/view/.

Six examples of max-trees applications were described in this section. The first was a simple filtering step to mitigate noise in the image without blurring it. The second showed that a filtering step combining size and shape information can be used for segmentation purposes. The third application illustrated how to use the max-tree to select markers to be used as input for other segmentation techniques. The fourth application illustrated the combination of the max-tree information



(a)

Fig. 15. Interactive max-tree tool.

with machine learning techniques for image segmentation. The fifth used the MMS methodology to simplify the max-tree and speed-up the nodes classification. Finally, the sixth application illustrated the use of the interactive max-tree visualization tool for segmentation purposes.

V. Current Trends

The max-tree was one of the first tree-based image representations proposed. After its popularization, many new tree-based image representations were proposed. Salembier and Garrido [53] proposed the Binary Partition Tree, which is a compact multiscale representation of the image composed of meaningful regions that can be extracted from an image. It is devoted to image processing, segmentation and information retrieval applications. Ouzounis and Wilkinson [54], [55] proposed the dual-input max-tree and fast algorithms to compute it. Morimutsu et al. [56] proposed an incremental algorithm to compute families of max-trees with connectivity rules of increasing neighborhoods. Jalba and Westenberg [57] compared the filtering procedure on the max-tree with the filtering procedure on the watershed tree. Monasse and Guichard [58] proposed a contrast-invariant image representation called tree of shapes, which represents the hierarchical relationship between the shapes in the image. Géraud et al. [59] proposed a quasi-linear algorithm to build the tree of shapes, which can be seen as an extension of the max-tree union-find [60] construction algorithms. There is also some work trying to extend these tree structures to multivalued images [61], [62], [63], the main difficulty is that there is no strict order relationship between pixel values when working with multivalued images.

Xu et al. [64] proposed to build max-trees of tree-based image representations, i.e. build a max-tree of a max-tree or a max-tree of a tree of shapes. This second tree construction takes the image to the space of shapes allowing the creation of a novel class of connected operators from the leveling family and more complex morphological analysis, such as the computation of extinction values for non-increasing attributes. This pipeline is illustrated by the black and red paths in Fig. 3. This methodology was used for blood vessels segmentation, a generalization of constrained connectivity [65], and hierarchical segmentation [66].

Max-trees of real images usually have thousands of nodes, therefore it may be useful to simplify it trying to preserve the most relevant nodes according to the problem being analyzed [49]. Also, the tree topology may be relevant, since under controlled circumstances, such as in brain MR images, it is expected that a common topology be present, some max-tree filters like the maximal max-tree simplification [5] try to simplify the max-tree and preserve its topology at the same time.

VI. Conclusions

This paper presented an overview of the max-tree principles, algorithms and applications. We pointed to interesting works on the field and illustrated how the max-tree can be used in different applications. We tried to emphasize the fact that the size and shape information that the max-tree provides about the structures in the image is well suited for applications, such as medical imaging applications, where we have prior knowledge of size and shape of the structures of interest. Since the max-tree is based on a threshold decomposition, which is sensitive to noise, it is expected that with the advance of imaging techniques and consequent improvement of images quality, the max-tree based applications will be even more accurate.

Acknowledgment

This project was supported by FAPESP CEPID-BRAINN (2013/07559-3) and CAPES PVE (88881.062158/2014-01). Roberto Lotufo thanks CNPq (311228/2014-3) for sponsorship. Roberto Souza thanks FAPESP grant (2013/23514-0) for the support.

References

[1] P. Salembier and J. Serra, “Flat zones filtering, connected operators, and filters by reconstruction,” *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1153–1160, 1995.

[2] P. Salembier and M. Wilkinson, “Connected operators,” *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, November 2009.

[3] P. Salembier, A. Oliveras, and L. Garrido, “Antiextensive connected operators for image and sequence processing,” *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.

[4] R. Jones, “Connected filtering and segmentation using component trees,” *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 215–228, 1999.

[5] R. Souza, L. Rittner, R. Machado, and R. Lotufo, “Maximal max-tree simplification,” in *22nd International Conference on Pattern Recognition*, Aug 2014, pp. 3132–3137.

[6] M. Wilkinson, H. Gao, W. Hesselink, J. Jonker, and A. Meijster, “Concurrent computation of attribute filters on shared memory parallel machines,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1800–1813, Oct 2008.

[7] E. Carlinet and T. Géraud, “A comparative review of component tree computation algorithms,” *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3885–3895, Sept 2014.

[8] P. Matas, E. Dokladalova, M. Akil, T. Grandpierre, L. Najman, M. Poupa, and V. Georgiev, “Parallel algorithm for concurrent computation of connected component tree.” in *ACIVS*, ser. Lecture Notes in Computer Science, vol. 5259. Springer, 2008, pp. 230–241.

[9] L. Najman and M. Couprie, “Building the component tree in quasi-linear time,” *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3531–3539, Nov 2006.

[10] R. Souza, L. Rittner, R. Lotufo, and R. Machado, “An array-based node-oriented max-tree representation,” in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 3620–3624.

[11] X. Huang, M. Fisher, and D. Smith, “An efficient implementation of max tree with linked list and hash table.” in *DICTA*. CSIRO Publishing, 2003, pp. 299–308.

[12] L. Vincent, “Morphological area openings and closings for grey-scale images,” in *Shape in Picture*. Springer Berlin Heidelberg, 1994, vol. 126, pp. 197–208.

[13] J. Fabrizio and B. Marcotegui, *Fast Implementation of the Ultimate Opening*. Springer Berlin Heidelberg, 2009, pp. 272–281.

[14] P. Teeninga, U. Moschini, S. Trager, and M. Wilkinson, *Improved Detection of Faint Extended Astronomical Objects Through Statistical Attribute Filtering*. Springer International Publishing, 2015, pp. 157–168.

[15] F. Kiwanuka and M. Wilkinson, *Cluster Based Vector Attribute Filtering*. Springer International Publishing, 2015, pp. 277–288.

[16] E. Urbach, J. Roerdink, and M. Wilkinson, “Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 272–285, 2007.

[17] M. Donoser, C. Arth, and H. Bischof, “Detecting, tracking and recognizing license plates,” in *Proceedings of the 8th Asian conference on Computer vision - Volume Part II*, ser. ACCV’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 447–456.

[18] C. Merino-Gracia, K. Lenc, and M. Mirmehdi, “A head-mounted device for recognizing text in natural scenes,” in *Proceedings of the 4th international conference on Camera-Based Document Analysis and Recognition*, ser. CBDAR’11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 29–41.

[19] P. Dokládal, I. Bloch, M. Couprie, D. Ruijters, R. Urtasun, and L. Garnero, “Topologically controlled segmentation of 3d magnetic resonance images of the head by using morphological operators,” *Pattern Recognition*, vol. 36, no. 10, pp. 2463–2478, 2003.

[20] B. Caldairou, B. Naegel, and N. Passat, *Segmentation of Complex Images Based on Component-Trees: Methodological Tools*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 171–180.

[21] B. Naegel, N. Passat, N. Boch, and M. Kocher, “Segmentation using vector-attribute filters: Methodology and application to dermatological imaging,” in *Proceedings of the Eighth International Symposium on Mathematical Morphology (ISMM’07)*, 2007, pp. 239–250.

[22] M. Richard, M. Fleute, L. Desbat, S. Lavallee, and J. Demongeot, “Registration of medical images for surgical action: use of optical sensors and matching algorithm,” pp. 125–139, 2000.

[23] P. Ghamisi, R. Souza, J. Benediktsson, X. Zhu, L. Rittner, and R. Lotufo, “Extinction profiles for the classification of remote sensing data (accepted for publication),” *IEEE Transactions on Geosciences and Remote Sensing*, 2016.

[24] J. Benediktsson, L. Bruzzone, J. Chanussot, M. D. Mura, P. Salembier, and S. Valero, *Hierarchical Analysis of Remote Sensing Data: Morphological Attribute Profiles and Binary Partition Trees*. Springer Berlin Heidelberg, 2011, pp. 306–319.

[25] M. D. Mura, J.Benediktsson, and L. Bruzzone, *Self-dual Attribute Profiles for the Analysis of Remote Sensing Images*. Springer Berlin Heidelberg, 2011, pp. 320–330.

[26] M. Wilkinson, P. Soille, M. Pesaresi, and G. Ouzounis, *Concurrent Computation of Differential Morphological Profiles on Giga Pixel Images*. Springer Berlin Heidelberg, 2011, pp. 331–342.

[27] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” in *British Machine Vision Conference*, 2002, pp. 384–393.

[28] Y. Xu, P. Monasse, T. Géraud, and L. Najman, “Tree-based morse regions: A topological approach to local feature detection,” *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5612–5625, 2014.

[29] M. Westenberg, J. Roerdink, and M. Wilkinson, “Volumetric attribute filtering and interactive visualization using the max-tree representation.”

IEEE Transactions on Image Processing, vol. 16, no. 12, pp. 2943–2952, 2007.

[30] G. Ouzounis and L. Gueguen, "Interactive collection of training samples from the max-tree structure," in *18th IEEE International Conference on Image Processing (ICIP)*, Sept 2011, pp. 1449–1452.

[31] B. Naegel and N. Passat, "Interactive Segmentation Based on Component-trees," *Image Processing On Line*, pp. 89–97, 2014.

[32] L. Tavares, R. Souza, L. Rittner, R. Machado, and R. Lotufo, "Interactive max-tree visualization tool for image processing and analysis," in *International Conference on Image Processing Theory, Tools and Applications (IPTA), 2015*, Nov 2015, pp. 119–124.

[33] ——, "A max-tree simplification proposal and applications for the interactive max-tree visualization tool," in *Brazilian Symposium on Computer Graphics and Image Processing*, October 2016.

[34] R. Gonzalez and R. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[35] L. Najman and H. Talbot, *Mathematical Morphology: from theory to applications*, N. Laurent and T. Hugues, Eds. ISTE-Wiley, Jun. 2010.

[36] E. Dougherty and R. Lotufo, *Hands-on Morphological Image Processing (SPIE Tutorial Texts in Optical Engineering Vol. TT59)*. SPIE Publications, 2003.

[37] E. Urbach and M. Wilkinson, "Shape-Only Granulometries and Grey-Scale Shape Filters," in *Proceedings of the VIth International Symposium on Mathematical Morpholgy*, vol. 6, 2002.

[38] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Efficient computation of attributes and saliency maps on tree-based image representations," in *Mathematical Morphology and Its Application to Signal and Image Processing*, ser. Lecture Notes in Computer Science Series, vol. 9082. Springer, 2015, pp. 693–704.

[39] C. Berger, T. Geraud, R. Levillain, N. Widynski, A. Baillard, and E. Bertin, "Effective component tree computation with application to pattern recognition in astronomical imaging," in *2007 IEEE International Conference on Image Processing*, vol. 4, Sept 2007, pp. IV – 41–IV – 44.

[40] M. Wilkinson, "A fast component-tree algorithm for high dynamic-range images and second generation connectivity," in *18th IEEE International Conference on Image Processing (ICIP)*, Sept 2011, pp. 1021–1024.

[41] G. Bertrand, "On the dynamics," *Image and Vision Computing*, vol. 25, no. 4, pp. 447–454, apr 2007.

[42] M. Grimaud, "New measure of contrast: the dynamics," pp. 292–305, 1992.

[43] A. Silva and R. Lotufo, "New extinction values from efficient construction and analysis of extended attribute component tree," in *XXI Brazilian Symposium on Computer Graphics and Image Processing, 2008. SIBGRAPI '08.*, 2008, pp. 204–211.

[44] C. Vachier, "Extinction value: a new measurement of persistence," in *IEEE Workshop on Nonlinear Signal and Image Processing*, vol. I, 1995, pp. 254–257.

[45] R. Levillain, T. Géraud, and L. Najman, "Milena: Write generic morphological algorithms once, run on many kinds of images," in *Mathematical Morphology and Its Application to Signal and Image Processing*, ser. Lecture Notes in Computer Science, vol. 5720. Springer Berlin / Heidelberg, aug 2009, pp. 295–306.

[46] M. Couprie, L. Marak, and H. Talbot, "The pink image processing library," in *Euroscipy*, Paris, 2011.

[47] S. van der Walt, S. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, March 2011.

[48] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Proc.*, vol. 13, no. 4, pp. 600–612, April 2004.

[49] R. Souza, L. Rittner, R. Machado, and R. Lotufo, *A Comparison Between Extinction Filters and Attribute Filters*. Springer International Publishing, 2015, pp. 63–74.

[50] J. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundam. Inf.*, vol. 41, no. 1,2, pp. 187–228, 2000.

[51] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[52] Y. Hu, "Efficient, High-Quality Force-Directed Graph Drawing," *The Mathematica Journal*, vol. 10, no. 1, pp. 37–71, 2006.

[53] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, 2000.

[54] G. Ouzounis and M. Wilkinson, "A parallel implementation of the dual-input max-tree algorithm for attribute filtering," in *Proceedings of ISMM 2007, 8th International Symposium on Mathematical Morphology*, vol. 1. Instituto Nacional de Pesquisas Espaciais, oct 2007, pp. 449–460.

[55] ——, "Mask-based second-generation connectivity and attribute filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 990–1004, 2007.

[56] A. Morimitsu, W. Alves, and R. Hashimoto, *Incremental and Efficient Computation of Families of Component Trees*. Springer International Publishing, 2015, pp. 681–692.

[57] A. C. Jalba and M. A. Westenberg, *A Comparison of Two Tree Representations for Data-Driven Volumetric Image Filtering*. Springer Berlin Heidelberg, 2011, pp. 405–416.

[58] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[59] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, *A Quasi-linear Algorithm to Compute the Tree of Shapes of nD Images*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 98–110.

[60] R. Tarjan, "Efficiency of a good but not linear set union algorithm," *J. ACM*, vol. 22, no. 2, pp. 215–225, 1975.

[61] F. Tushabe and M. Wilkinson, "Color processing using max-trees: A comparison on image compression," in *International Conference on Systems and Informatics (ICSAI), 2012*, 2012, pp. 1374–1380.

[62] C. Kurtz, B. Naegel, and N. Passat, "Connected filtering based on multivalued component-trees," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5152–5164, 2014.

[63] E. Carlinet and T. Géraud, *A Color Tree of Shapes with Illustrations on Filtering, Simplification, and Segmentation*. Springer International Publishing, 2015, pp. 363–374.

[64] Y. Xu, T. Géraud, and L. Najman, "Morphological filtering in shape spaces: Applications using tree-based image representations," in *2012 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 485–488.

[65] ——, "Two applications of shape-based morphology: blood vessels segmentation and a generalization of constrained connectivity," in *Mathematical Morphology and Its Application to Signal and Image Processing*, ser. Lecture Notes in Computer Science Series, vol. 7883. Springer, 2013, pp. 390–401.

[66] Y. Xu, E. Carlinet, T. Geraud, and L. Najman, "Hierarchical segmentation using tree-based shape space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.