New Extinction Values From Efficient Construction and Analysis of Extended Attribute Component Tree

Alexandre Gonçalves Silva
Universidade do Estado de Santa Catarina
Departamento de Ciência da Computação
alexandre@joinville.udesc.br

Roberto de Alencar Lotufo
Universidade Estadual de Campinas
Faculdade de Eng. Elétrica e de Computação
lotufo@unicamp.br

Abstract

A gray-level image can be interpreted as a topographical surface, and represented by a component tree, based on the inclusion relation of connected components obtained by threshold decomposition. Relations between plateaus, valleys or mountains of this relief are useful in computer vision systems. An important definition to characterize the topographical surface is the dynamics, introduced by Grimaud (1992), associated to each regional minimum. This concept has been extended, by Vachier and Meyer (1995), by the definition of extinction values associated to each extremum of the image. This paper proposes four new extinction values – two based on the topology of the component tree: (i) *number of descendants and* (ii) *sub-tree height; and two geometric:* (iii) *height and* (iv) *width of a level component bounding box. This paper describes efficient computation of these extinction values based on the incremental determination of attributes from the component tree construction in quasi-linear time, compares the computation time of the method and illustrates the usefulness of these new extinction values from real examples.*

1. Introduction

The location and segmentation of regions or objects of interest in a more direct possible way is a feature desired by many applications of computer vision. In mathematical morphology, the definition of contours of these objects is frequently obtained by watershed transformation [3] based on the influence zone of selected markers. Considering a gray-level image as a topographical surface (pixel intensity is an altitude value), the plateaus (flat zones) at the bottom of valleys (regional minima) or at the summits (regional maxima) are typically used as markers. However, images acquired by cameras or scanners normally provide a great number of these extrema points which can correspond to undesirable parts or noise, generating a watershed over-segmentation. The selection of significant minima (darker regions) or maxima (lighter regions) for selection of good markers can be done by dynamics [8, 2, 13]. The dynamics of a regional minimum is a contrast measure and can be seen as the minimum height one has to overcome starting from a valley to reach another deeper valley, as illustrated in Figure 1.

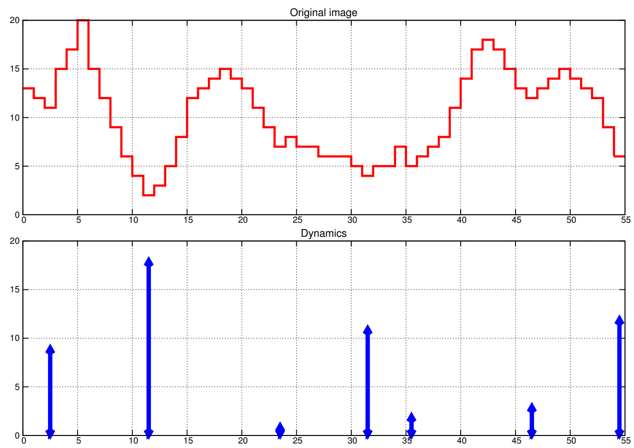


Figure 1. Dynamics of the regional minima.

The dynamics concept has been extended, by Vachier and Meyer (1995), by the definition of extinction values associated to each extremum of the image. The extinction value of a regional extremum (minimum or maximum) for any increasing attribute (height, area, volume, etc) is the maximal size of an attribute filter [5] such that this extremum still exists after the filtering [21]. Dynamics is a particular case of extinction value when the attribute is height. In this work, extinction value of maxima will be used and the interpretation, in this case, relates to an attribute filtering sufficiently great (removal of considerable summit land) for vanishing a hill from image relief.

Image pixels can be grouped under certain similarities in a set of regions. Understanding the image as an organization of regions promotes semantic information and enables the implementation of extinctions. In this sense, the Component Tree [16, 12, 10, 4, 14] is a structure based on the relief formed by the decomposition of a gray-level image by thresholds useful for extinction value determination [2]. It is especially interesting by presenting features as: *(i)* there are algorithms in quasi-linear time [14] and it requires only the adjacency definition for its construction; *(ii)* all level components are unique, corresponding to the lowest number of connected regions, obtained from decomposition by thresholding without redundancy of information; *(iii)* the results of operations on nodes, as pruning (removing branches of the extremity nodes) or graft[1] (removal of intermediary nodes), are connected anti-extensive operators that does not create new contours in the image; *(iv)* its hierarchical organization of regions enables a considerable disposal of nodes in subsequent processing steps (such as shape detection or matching of images). *(v)* the possibility of simultaneous filtering of multiple attributes, successive filtering in tree domain (image rendering only at the end), allowing the implementation of efficient algorithms (number of nodes or regions to examine is lesser than the number of pixels), besides other operations based on topology.

Other names for the component tree are found in the literature for the same (or similar) representation: dendrogram [6], connectivity tree [19], confinement tree [12, 11] or Max-tree[2] [17, 7, 15]. In this work, the latter name is preferably used, being represented by the symbol MT_I indicating the Max-tree of an image I. Some efficient algorithms for construction of this structure, are based on *union-find* [14] or hierarchical flood [7].

In this paper, new attributes are proposed for each node of the tree – number of descendants, sub-tree height, coordinates of the top-left and bottom-right corner of bounding box of the level component associated – and determined incrementally in the same algorithm for the Max-tree construction, maintaining running time as fast as the recent solutions in the literature, and increasing the possibilities for differentiated filtering or segmentation. The main contribution, however, is to establish new extinction values – beyond the height [8], area and volume [21, 20] –, that can be used as single technique or auxiliary tool in the identification of objects of interest in gray-level images. The proposal is to add four extinctions based on the new attributes mentioned – *(i)* extinction of descendants; *(ii)* extinction of topological height; *(iii)* extinction of height of the bounding box; and *(iv)* extinction of width of the bounding box – in a fast way from this suitable image hierarchical representation. Section 2 describes some preliminary definitions such as the dynamics, Max-tree, and extinction values. Section 3 details the proposed algorithms and shows tests of performance. Section 4 suggests some applications. Finally, we have conclusions and future works in Section 5.

1 This designation is not usual and refers to the removal of internal tree nodes (that are not root or leaves).

2 Max-tree refers to the algorithm introduced by Salembier et al. (1998) for an efficient implementation of the component tree.

2. Basic definitions

A *gray-scale image* is a rectangular matrix I of *pixels* in a domain[3] $E \subset \mathbb{N}^2$. The intensity of a pixel x is denoted by $I(x) \in \mathbb{N}$ such that $0 \leq I(x) \leq n_{\max}, \forall x \in E, n_{\max} \in \mathbb{N}^*$. It is *binary image* if $n_{\max} = 1$. A *negative image* of I is $\bar{I}(x) = n_{\max} - I(x)$, $\forall x \in E$. Let $\mathscr{N}_E(x)$ a *neighborhood* of x, a *path* from x_1 to x_n is defined as a sequence $P_{x_1,x_n} = (x_1, x_2, \ldots, x_n)$, where $x_{i+1} \in \mathscr{N}_E(x_i)$, $\forall i \in [1, n)$. A *connected component* is a maximal subset of pixels $\mathcal{C} \subseteq E$, where there is always a path P_{x_a,x_b} entirely inside $\mathcal{C}, \forall x_a, x_b \in \mathcal{C}$. A *flat zone* is a connected component Z of the image, such that, $\forall x_a, x_b \in Z, I(x_a) = I(x_b)$. A *regional maximum* is a flat zone M such that $I(x_m) > I(x_n)$, $\forall x_m \in M$, for every pixel x_n in the neighborhood of M. A *threshold decomposition* is defined by a set of binary images $X_h(I)$ representing the thresholding of I at each possible level h, or $X_h(I) = \{x \in E | I(x) \geq h\}, \forall h \in [0, n_{\max}]$. *Level component*[4] is an image $\mathcal{C}_h^k(x) = 1$ if x belongs to the connected component of $X_h(I)$ labeled with $k \in \mathbb{N}^*$, and $\mathcal{C}_h^k(x) = 0$ otherwise, $\forall x \in E$. A *connected operator* acts by eliminating or merging of level components, preserving contours. An *attribute* μ of a level component is *increasing* [5] if $\mathcal{C}_{h_a}^i \subseteq \mathcal{C}_{h_b}^j \Rightarrow \mu(\mathcal{C}_{h_a}^i) \leq \mu(\mathcal{C}_{h_b}^j)$. An *attribute opening* $\Upsilon_{\mu,\lambda}(I)$ is a connected operator on image I and consists in to keep only its level components $\mathcal{C}_h^k$ whose increasing attribute μ exceeds a threshold λ [5], or $\Upsilon_{\mu,\lambda}(I)(x) = \max_{(\forall h, \forall k)}\{h{\cdot}\mathcal{C}_h^k(x) \,|\, \mu(\mathcal{C}_h^k) \geq \lambda\}, \forall x \in E$.

2.1. Max-tree

Max-tree is a hierarchical structure formed from the relationship between connected regions obtained by threshold decomposition. In order to organize the level components $\mathcal{C}_h^k$ in a hierarchy, a node $\mathcal{N}$ is associated with each. The descendant relationship between nodes $\mathcal{N}_a$ and $\mathcal{N}_b$ occurs if and only if $\mathcal{C}_{h_a}^i$ concerning to $\mathcal{N}_a$ contains $\mathcal{C}_{h_b}^j$ referring to $\mathcal{N}_b$, where $h_a < h_b$ [7]. $\mathcal{N}_r$ is the root node if related to unique component $\mathcal{C}_{\min(I)}^1$, where $\min(I)$ is the lowest image intensity. $\mathcal{N}_l$ is a leaf node if there is not another level component inside of its component (regional maximum). If $\mathcal{C}_{h_a}^i \equiv \mathcal{C}_{h_b}^j$, and $h_a < h_b$, a node $\mathcal{N}_b$ is only defined for $\mathcal{C}_{h_b}^j$ (the allocation of a node for $\mathcal{C}_{h_a}^i$ is redundant). In other

3 $\mathbb{N}$ is the set of natural numbers (efficient component tree can also be constructed for images with intensities in the real domain [1]).

4 Also known as *peak component*, *k-component* or just *connected component* in the literature.

words, a node is only defined for a level component which presents a flat zone (visible) at the image.

Figure 2 illustrates the Max-tree building process of an image with a line and eight columns. We can imagine that the topographical surface from the gray-level image is completely submerged initially. As water flows, islands, whose summit plateaus correspond to the regional maxima (tree leaves), emerges. Assuming the constant flow of water, a new plateau (flat zone) can appear (new parent-child relationship) or two or more islands can unite (definition of a parent for two or more children) successively until there is only one "block of land" (root of the tree). This is the method interpretation based on *union-find* [18, 22, 14]. However, this structure can be formed with equivalent efficiency, from *recursive hierarchical flood* from the regional minimum with lowest intensity [17, 7]. In this case, however, the interpretation is not so simple, but can be seen as the elevation of water toward the maxima, from its injection into a mountain at a time (for more details, see Algorithm 2 in Section 3).

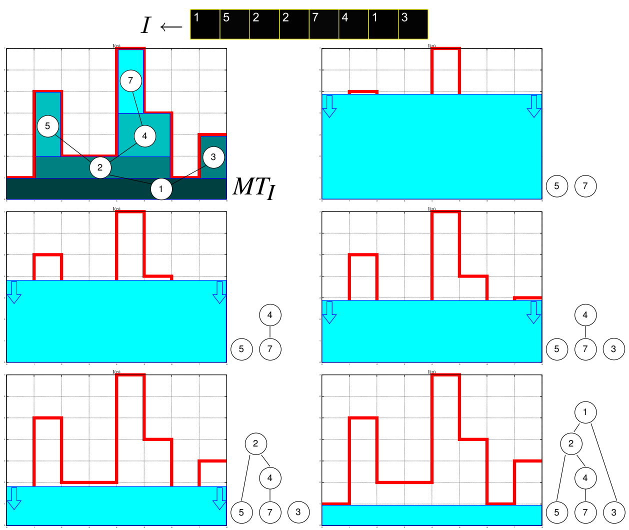


Figure 2. Max-tree construction.

2.2. Dynamics

Path dynamic, assuming P_{x_a,x_b} linking the pixels x_a and x_b of an image I, is the difference of levels between the points of highest and lowest intensity in this path, or $D_P(x_a,x_b) = \{\max(|I(x_i) - I(x_j)|) \mid x_i, x_j \in P_{x_a,x_b}\}$. *Dynamics between two pixels*, x_a x_b, is equal to the lesser path dynamic between them, or, $D_X(x_a,x_b) = \{\min(D_P(x_a,x_b)) \mid P_{x_a,x_b}$ is one of the possible paths between x_a and $x_b\}$. *Dynamics of a regional maximum M*

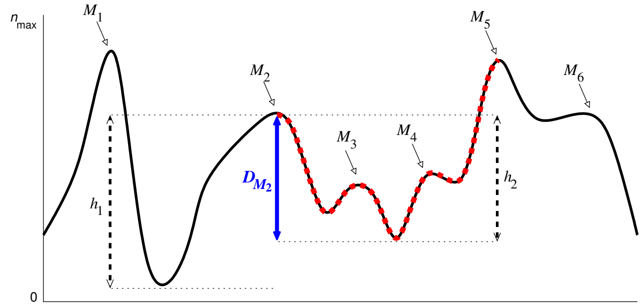


Figure 3. Dynamics of regional maximum M_2.

is the minimum altitude (difference of gray levels) that we should fall, given a path P_{x_a,x_b}, from a pixel x_a of M, to achieve a pixel x_b of another regional maximum M_V higher than M, or $D_M = \{\min(D_X(x_a,x_b)) \mid x_a \in M, x_b \in M_V, I(x_a) < I(x_b)\}$[5]. Figure 3 illustrates this idea. M_i, $i \in [1,5]$, are the regional maxima. To determine the dynamics[6] of M_2, there are two higher maxima, M_1 and M_5. But it is necessary to climb down h_1 in the first path and h_2, in the second. Being $h_2 < h_1$, the second path (dotted) is preferred, and $D_{M_2} = h_2$. Two situations may occur in the determination of the dynamics: *(i)* it is possible that there is no other higher maximum, and in this case, the dynamics is defined as the height of the relief as a whole (or infinite); *(ii)* tie, where two or more regional maxima have same dynamics and are positioned on a same greater mountain (analogous to minima dynamic tie under same valley described by Grimaud (1992)). In this case, an infinitesimal difference of height should be considered and the original dynamics must be assigned to only one of them. The others maxima take lower dynamics in relation to another initially tied maximum with gray-level infinitesimally higher. In this manner, there is an augment of the influence region of the most representative maxima, reducing the occurrence of possible markers for an object localization further analysis. Dynamic is a measure of contrast concise and powerful for the identification of regions of interest in the image (examples will be presented in the following sections).

2.3. Extinction values

In the previous section, the dynamic was discussed. In short, this measures the smallest decrease in altitude, from a regional maximum, to achieve other higher regional max-

5 Grimaud (1992) presents dynamics of regional minima with inverse logic in relation to this definition. The use of regional maxima agrees with the characteristics of the Max-tree. Problems can also be modeled in the Max-tree of the negative image or Min-tree directly.

6 In the absence of specification, it refers to *dynamics of regional maximum*.

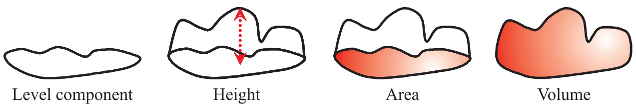


Figure 4. Attributes from the relief top of a region of the image.

imum. In other words, it refers to the *height extinction* of an hill or mountain in the relief (or sub-tree at the Max-tree) for λ_h sufficiently large in an attribute opening $\Upsilon_{height,\lambda_h}(I)$ (Max-tree pruning). This concept can be extended to other attributes, in addition to height (related to the difference of gray levels), since they are increasing from regional maxima (leaves) to the lowest gray-level (root) of the image (tree). Figure 4 shows some possible measurements on a level component (associated to a node $\mathcal{N}$ of sub-tree). Above of this component (descendant nodes of $\mathcal{N}$) there is a "stack of land" of the image relief where area and volume can be determined, in addition to height. All these attributes increase if a level component is obtained in a lower gray-level thresholding (for example, the parent of a node $\mathcal{N}$ presents larger height, area and volume). Thus, as well as height extinction, we can define *area extinction* – from a maximum, the final area of the hill to achieve another hill (from another regional maximum) whose base has larger area or $\Upsilon_{area,\lambda_a}(I)$ with λ_a sufficiently large for the vanishing of the first hill – and *volume extinction* – from a maximum, the final volume of the hill to achieve another hill (another regional maximum) with larger volume or $\Upsilon_{volume,\lambda_v}(I)$ with λ_v sufficiently large for the vanishing of the first. Vachier and Meyer (1995) suggests extinction values for area and volume of regional minima. However, the regional maxima are used in this work because are easily extracted from the efficient Max-tree construction.

3. New extinction values proposed

Once checked the relationship between Max-tree structure and determination of extinction, by pruning (attribute opening), any increasing attribute associated with a node can be used as extinction value. Algorithm 1 is proposed, in this respect, for the generalization of extinction value calculation, given any attribute increasing μ, based on information from the Max-tree. Broadly speaking, from a tree leaf $\mathcal{N}^L$, a path toward root is initiated. When a parent node of N^A appears with more than one child (line 7), or either, if a branching tree appears, a verification is done on each sibling of this node: if sibling already visited, and if there is extinction tie, or if attribute the sibling is higher (lines 9-11), then the attribute of $\mathcal{N}^A$ is defined as the extinction of $\mathcal{N}^L$ (line 19). Finally, the image is rendered with assigning of extinction to each level component $\mathcal{C}_{\mathcal{N}^L}$ of regional maximum represented by a leaf node $\mathcal{N}^L$ (line 21). The attribute μ can be height (difference of gray level), area or volume, already well established in the literature. Now, the definition of new attributes, associated to Max-tree nodes (level components), is proposed, in order to use as extinction values in the generic algorithm presented.

Algorithm 1: Generic algorithm to determine the extinction values using the Max-tree.

INPUT: MT_I, μ
OUTPUT: E_μ

EXTINCTION()
1 $continue \leftarrow$ **true**
2 **for each** $\mathcal{N}^L \in$ (leaves of MT_I)
3 $extinction \leftarrow \infty$
4 $\mathcal{N}^A \leftarrow \mathcal{N}^L$
5 $\mathcal{N}^P \leftarrow$ parent of $\mathcal{N}^A$
6 **while** $continue$ **and** $\exists \mathcal{N}^P$
7 **if** (number of children of $\mathcal{N}^P$) > 1
8 **for each** $\mathcal{N}^C \in$ (children of $\mathcal{N}^P$) **and** $continue$
9 **if** (($\mathcal{N}^C$ already visited) **and**
10 $\mathcal{N}^C \neq \mathcal{N}^A$ **and** $\mathcal{N}^C_\mu = \mathcal{N}^A_\mu$)
11 **or** ($\mathcal{N}^C \neq \mathcal{N}^A$ **and** $\mathcal{N}^C_\mu > \mathcal{N}^A_\mu$)
12 $continue \leftarrow$ **false**
13 $\mathcal{N}^C$ is marked as visited
14 **if** $continue$
15 $\mathcal{N}^A \leftarrow \mathcal{N}^P$
16 $\mathcal{N}^P \leftarrow$ parent of $\mathcal{N}^A$
17 $continue \leftarrow$ **true**
18 **if** $\exists \mathcal{N}^P$
19 $extinction \leftarrow \mathcal{N}^A_\mu$
20 $\mathcal{N}^L_{ext} \leftarrow extinction$
21 $E_\mu \leftarrow \max\{\mathcal{N}^L_{ext} \cdot \mathcal{C}_{\mathcal{N}^L} \mid \forall \mathcal{N}^L \in$ (leaves of MT_I)$\}$
22 **return** E_μ

Some measures can be added incrementally in Max-tree construction time. Thus, the algorithm remain with the same complexity of the original Max-tree algorithm in which it is based (there are quasi-linear solutions for this issue). In the set of information associated with each node $\mathcal{N}_x$, it is possible to include:

- **Number of descendants** of the sub-tree rooted in $\mathcal{N}_x$ or, in other words, cardinality of the set of nodes of this sub-tree.
- **Topological height** of the sub-tree rooted in $\mathcal{N}_x$ or, otherwise, the maximum quantity of edges on the path (maximum path length) from this root node to any other descending node (certainly a leaf) in this sub-tree.
- **Height and width of bounding box** of the level component $\mathcal{C}_{\mathcal{N}_x}$, or difference between the higher and lower line (for height), and between higher and lower column (for width), considering all the pixels belonging to the level component.

Figure 5 helps in the understanding of these definitions. For instance, the node C presents 6 descendants (F, G, H,

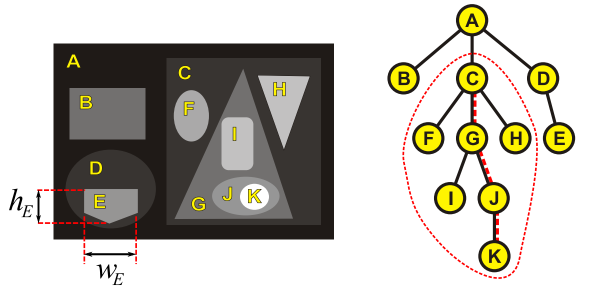


Figure 5. Image (left) and its Max-tree (right) marked with attributes under discussion.

I, J e K) and its topological height is 3, corresponding to the maximum length of the path to another descending node (in this case, way up to K). Regarding the bounding box, for example, the level component concerning the region E presents height h_E and width w_E.

These attributes, when applied to Algorithm 1, define two topological extinctions (in the tree domain): descendants, E_{desc}, and sub-tree height, E_{htop}; and two geometric extinctions (in the image domain): height and width of the bounding box, E_{hbbox} and E_{wbbox}, respectively.

Algorithm 2 clarifies the incremental calculation of these attributes in the Max-tree construction process. It modifies the method based on hierarchical flood of Salembier et al. (1998), adding determination the coordinates of the top left corner $x_\ulcorner$ and bottom right corner $x_\lrcorner$ of the bounding box (initialization in the lines 8-9 and tests in 12-13) for each pixel p visited (line 11). Signs $\gtrsim$ or $\lesssim$ compares (greater or lesser) both coordinates (line **and** column) for bounding box corner determination. LINK (lines 25 and 27) establishes the connection between the nodes parent and child represented by a tuple (gray-level and label), in addition to the current values of $x_\ulcorner$ and $x_\lrcorner$ that should be updated considering the child pixels included in the parent's component.

Algorithm 3 details this linking of nodes (line 25 or 27 of Algorithm 2), conforming the hierarchical flood, with the determination the number of descendants and topological height incrementally. Firstly, a search by nodes parent and child is done through an auxiliary hash table (lines 1-2). If a node is not found, then it is created (lines 4 or 15). Case there is only the parent node (line 6), then this and all its ancestors receive more one descendant (lines 7-11) considering the newly created child. If both nodes already exist (but are not yet linked), then are added the descendants of the child more one to the parent's descendants (line 13). In any case, the sub-tree height of the parent is updated with the sub-tree height of the child plus one (line 16). The linking between the nodes is finished (line 17). Finally, the corners of the bounding box are again observed and updated when they are necessary (lines 18-21).

Algorithm 2: Max-tree construction, based on Salembier et al. (1998), with support for new attributes.

INPUT: $I, \mathcal{N}_E$
OUTPUT: MT_I
INITIALIZATION:
1 $level[k] \leftarrow$ **false**, $\forall k \in [0, n_{\max}]$ *//current gray levels*
2 $label[k] \leftarrow 0, \forall k \in [0, n_{\max}]$ *//current labels*
3 $queue[k] \leftarrow \emptyset, \forall k \in [0, n_{\max}]$ *//hierarchical queues*
4 $queue[\min(I)]$.insert(x_m) such that $I(x_m) = \min(I)$
5 $status[x] \leftarrow 0, \forall x \in E \subset \mathbb{N}^2$ *//persistent labels*
6 $MT_I \leftarrow \emptyset$
7 FLOOD$(\min(I))$

FLOOD(n)
8 $x_\ulcorner \leftarrow \{\infty, \infty\}$ *//bounding box top left corner*
9 $x_\lrcorner \leftarrow \{0, 0\}$ *//bounding box bottom right corner*
10 **while** $queue[n] \neq \emptyset$
11 $p \leftarrow queue[n]$.remove()
12 **if** $p \lesssim x_\ulcorner \Rightarrow x_\ulcorner \leftarrow p$
13 **if** $p \gtrsim x_\lrcorner \Rightarrow x_\lrcorner \leftarrow p$
14 $status[p] \leftarrow label[n] + 1$
15 **for each** $q \in \mathcal{N}_E(p)$
16 **if** $status[q] = 0$ *//not analyzed*
17 $m \leftarrow I[q]$; $queue[m]$.insert(q); $level[m] \leftarrow$ **true**
18 $status[q] \leftarrow -1$ *//in queue*
19 **while** $m > n$
20 $m \leftarrow$ FLOOD(m)
21 $m \leftarrow n - 1$
22 **while** $m \geq 0$ **and** (**not** $level[m]$)
23 $m \leftarrow m - 1$
24 **if** $m \geq 0$
25 MT_I.LINK$(\{m, label[m] + 1\}, \{n, label[n] + 1\}, x_\ulcorner, x_\lrcorner)$
26 **else**
27 MT_I.LINK$(\{-1, 1\}, \{min(I), 1\}, x_\ulcorner, x_\lrcorner)$
28 $level[n] \leftarrow$ **false**
29 $label[n] \leftarrow label[n] + 1$
30 **return** m

Figure 6 compares Max-tree construction time of four different algorithms[7]: a_1 using hash table [9]; a_2 using *union-find* [14]; a_3 toolbox[8] based on Salembier (1998); a_4 referent to Algorithms 2 and 3 presented[9]. At least the algorithms a_2, a_3 and a_4 calculate height, area and volume attributes. a_3 and a_4 also determine the bounding box. Algorithm a_4 proposed also adds number of descendants and topological height incrementally. Each point in the plotting consists in the time $t(n)$ (milliseconds) for the Max-tree construction of a random image I_R with n pixels. $I_R(x) = \text{rand}(n_{\max}), \forall x \in E$, with dimensions $N \times N$ ($n = N^2$ pixels) for $N = 100k$ such that $k = [1, 15]$ (15 images). The proposed implementation, as well as adds new attributes, has performed better than the other solutions. This was possible due the data structures used, involving a hashing allocated from image histogram information (see its use at lines 1 and 2 of Algorithm 3) and memory allocation for new nodes (lines 4 and 15) in block (a node is available from a vector of nodes previously allocated and, if there are not more free nodes, a new vector of

7 All tests were done on a Mobile Pentium© 4, 3.2GHz, 512MB, and the four algorithms were implemented in C/C++.
8 http://mmorph.com
9 http://code.google.com/p/extinction-values

Algorithm 3: Algorithm for insertion of parent-child relationship and calculation of new attributes.

INPUT: $MT_I, \{n_{parent}, l_{parent}\}, \{n_{child}, l_{child}\}, x_\ulcorner, x_\lrcorner$
OUTPUT: MT_I

LINK()
$\mathcal{N}^P \leftarrow MT_I.\text{HASH\_TABLE}(\{n_{parent}, l_{parent}\})$
$\mathcal{N}^C \leftarrow MT_I.\text{HASH\_TABLE}(\{n_{child}, l_{child}\})$
if $\nexists \mathcal{N}^P$
  $\mathcal{N}^P \leftarrow MT_I.\text{NEW\_NODE}(\{n_{parent}, l_{parent}\}, x_\ulcorner, x_\lrcorner)$
else
  if $\nexists \mathcal{N}^C$
    $\mathcal{N}^P_{desc} \leftarrow \mathcal{N}^P_{desc} + 1$
    $\mathcal{N}^A \leftarrow$ parent of $\mathcal{N}^P$
    while $\mathcal{N}^A$
      $\mathcal{N}^A_{desc} \leftarrow \mathcal{N}^A_{desc} + 1$
      $\mathcal{N}^A \leftarrow$ parent of $\mathcal{N}^A$
  else
    $\mathcal{N}^P_{desc} \leftarrow \mathcal{N}^P_{desc} + \mathcal{N}^C_{desc} + 1$
if $\nexists \mathcal{N}^C$
  $\mathcal{N}^C \leftarrow MT_I.\text{NEW\_NODE}(\{n_{child}, l_{child}\}, x_\ulcorner, x_\lrcorner)$
if $\mathcal{N}^P_{htop} \leq \mathcal{N}^C_{htop} \quad \Rightarrow \quad \mathcal{N}^P_{htop} \leftarrow \mathcal{N}^C_{htop} + 1$
link $\mathcal{N}^C$ as child of $\mathcal{N}^P$
if $x_\ulcorner \hat{<} \mathcal{N}^C_{x_\ulcorner} \quad \Rightarrow \quad \mathcal{N}^C_{x_\ulcorner} \leftarrow x_\ulcorner$
if $x_\lrcorner \check{>} \mathcal{N}^C_{x_\lrcorner} \quad \Rightarrow \quad \mathcal{N}^C_{x_\lrcorner} \leftarrow x_\lrcorner$
if $\mathcal{N}^C_{x_\ulcorner} \hat{<} \mathcal{N}^P_{x_\ulcorner} \quad \Rightarrow \quad \mathcal{N}^P_{x_\ulcorner} \leftarrow \mathcal{N}^C_{x_\ulcorner}$
if $\mathcal{N}^C_{x_\lrcorner} \check{>} \mathcal{N}^P_{x_\lrcorner} \quad \Rightarrow \quad \mathcal{N}^P_{x_\lrcorner} \leftarrow \mathcal{N}^C_{x_\lrcorner}$

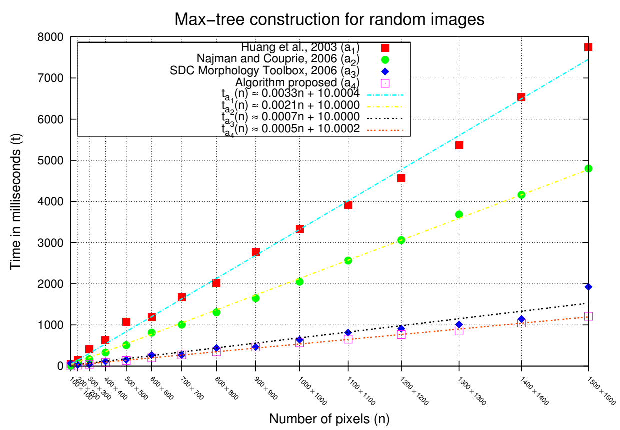


Figure 6. Comparison of Max-tree construction time.

same size as the previous is allocated), beyond the calculation of incremental attributes (updated throughout the iterations of the Salembier's algorithm) that has little influence in the total time of the Max-tree construction.

In relation to determination of extinctions, it was observed that the four proposals have similar time since they are implemented from the same Algorithm 1. Addition, the time of each extinction is not far from the Max-tree construction time, conforming Figure 7, following the same linearity. Reviewing Algorithm 1, we can see that its complexity depends on the number of leaves (line 2), beyond the number of steps, from each leaf, toward the tree root (line 6). And, once found a branch (line 7), still is necessary a scan in the children of the last node for verification of the dynamics (line 8). Considering several tests done, the number total the iterations of these three loops increases linearly with respect to the number of nodes N_{nodes} of the input Max-tree (there are around $1.8 \times N_{nodes}$ iterations for 15 random images independent of their dimensions). The number of nodes depends on the image content. An analysis for the maximum and minimum quantity of nodes in relation to the number of pixels is done in [7] (Chapter 3).

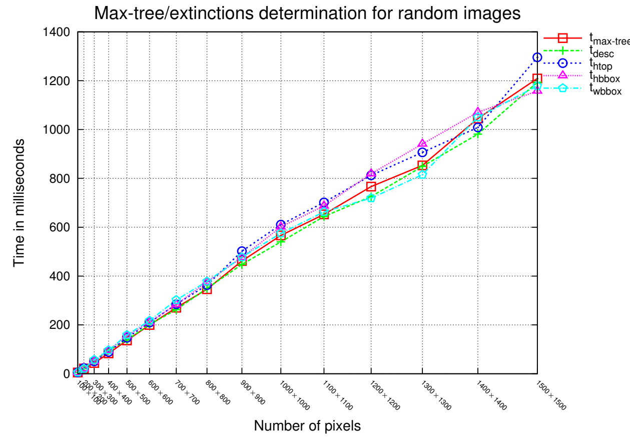


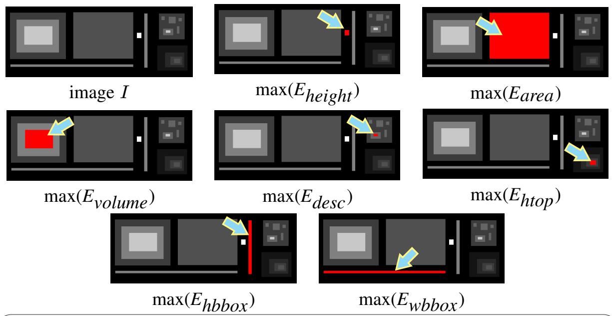
Figure 7. Running time for Max-tree and extinctions proposed.

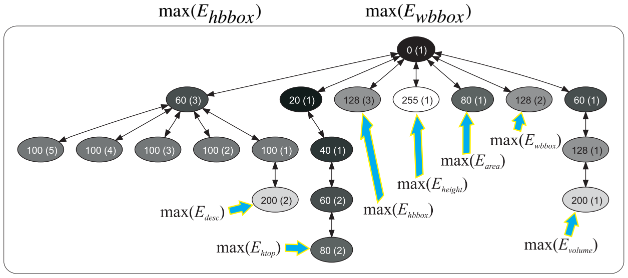
3.1. Examples

Figure 8 illustrates a synthetic image and its Max-tree with the maximum extinction values (components in red indicated by arrows) obtained from traditional attributes (height, area and volume) and the new values proposed in this work (descendants, topological height, bounding box height and width). The table, in turn, shows the extinctions of all maxima, highlighting the highest in bold. The added resources signal potential applications where the tree topology or the component geometry can be explored.

4. Computer vision applications

In this section, some applications of extinction values are presented, emphasizing the new attributes proposed. The regional maxima with significant extinctions are highlighted (in red). Only the regional maxima, whose extinction values are above a certain value or in a range of values, are selected. All tests can be reproduced using the source code and images of this work available at `http://code.google.com/p/extinction-values`.





MT_I and selected maxima

Node	E_{height}	E_{area}	E_{volume}	E_{desc}	E_{htop}	E_{hbbox}	E_{wbbox}
128 (3)	128	1056	135168	1	1	**172**	8
255 (1)	**255**	143	36465	1	1	13	11
80 (1)	80	**78948**	1328880	1	1	113	147
128 (2)	128	1806	231168	1	1	6	**459**
200 (1)	200	15481	**3870261**	3	3	113	137
80 (2)	80	4514	139020	4	**5**	61	74
100 (5)	40	72	2880	1	1	9	8
100 (4)	40	64	2560	1	1	8	8
100 (3)	40	272	10880	1	1	16	17
100 (2)	40	108	4320	1	1	63	6
200 (2)	200	3969	281380	**19**	3	16	63

Extinctions for each regional maximum

Figure 8. Illustration of the maximum extinction values for a synthetic image I (459×172).

4.1. Indirect counting

Vision systems can be used to avoid efforts of manual counting of pieces, objects, fruit, cells, etc, with greater speed and, in some cases, with greater precision, assuming no interference as fatigue, for instance. Figure 9 shows some examples where significant extinctions of descendants and topological height have adapted for the counting of tomatoes, beans, *brigadeiros*, and blood cells (the last two applied on the negative image).

4.2. Quality inspection

Control of industrial processes is another vision application. For example, the verifying of label integrity, the presence of all pieces in a set, the automatic inspection of shapes, sizes, colors, among others. Figure 10 shows significant extinctions of bounding box height and width of tablets, with application in the pharmaceutical industry.

5. Conclusion

The extinction values are powerful tools in applications of computer vision, indicating the placement of objects, according to their topological or geometrical features. Even with noisy images and wide variation in the range of extinction values, the results of the regional maxima selection with highlighted extinctions remain robust. The extinction value selection by threshold was sufficient to detect and count objects of interest in the experimentation reported in this paper. The Max-tree implementation of the component tree is based on Salembier et al. (1998), modified to compute the attributes incrementally. The speed efficiency was significant when compared to other algorithms.

Four new extinction values were proposed. The descendant extinction value signals the complexity of a region, in the sense of having many hills and summits, which can be useful in texture detection. The topological height indicates the amount of overlapping of "layers of land", which can be useful in detecting of spherical or pyramidal shaped objects. The height or width of a level

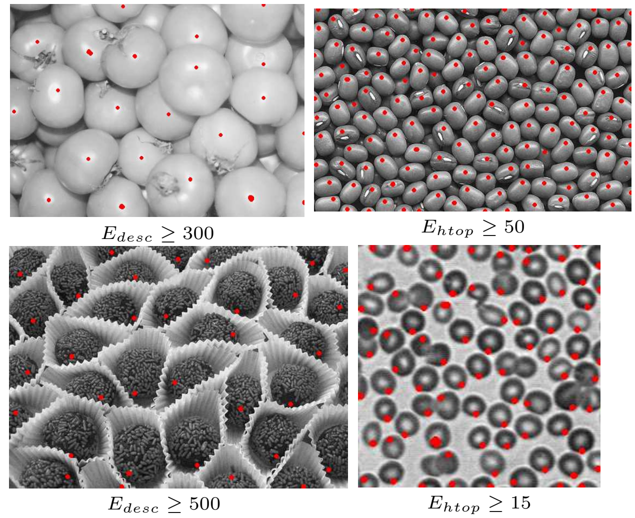


Figure 9. Indirect counting from descendants and topological height extinction.

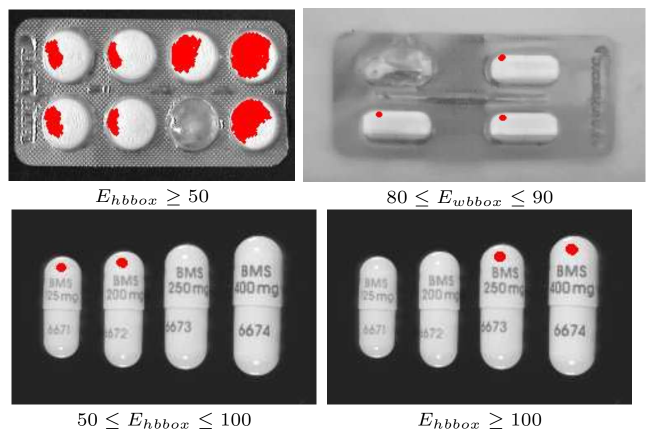


Figure 10. Quality inspection from bounding box extinction.

component bounding box reflects the approximate size, in pixels, of a object (lighter than its background) in the image.

A more careful research of new applications must be established. As future work, it would be interesting to focus in a single problem, examining, in more details, all the characteristics that this proposal differs in relation to other solutions, for the same purpose, in the literature.

Acknowledgments

The authors are grateful to Laurent Najman, Xiaoqiang Huang and SDC Morphology Toolbox for the release of their implementations of Max-tree. Thanks to UDESC and UNICAMP for supporting this research.

References

[1] 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 *IEEE International Conference on Image Processing*, volume 4, pages 41–44, San Antonio, USA, September 2007.

[2] G. Bertrand. On the dynamics. *Image and Vision Computing*, 25(4):447–454, 2007.

[3] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, New York, 1993.

[4] U. Braga-Neto and J. Goutsias. Grayscale Level Connectivity: Theory and Applications. *IEEE Transactions on Image Processing*, 13(12):1567–1580, 2004.

[5] E. J. Breen and R. Jones. Attribute Openings, Thinnings, and Granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, November 1996.

[6] L. Chen, M. W. Berry, and W. W. Hargrove. Using Dendronal Signatures for Feature Extraction and Retrieval. *International Journal of Imaging Systems and Technology*, 11(4):243–253, 2000.

[7] L. O. Garrido. *Hierarchical region based processing of images and video sequences: application to filtering, segmentation and information retrieval*. PhD thesis, Department of Signal Theory and Communications - Universitat Politècnica de Catalunya, Barcelona, April 2002.

[8] M. Grimaud. A New Measure of Contrast: the Dynamics. In *Image Algebra and Morphological Image Processing III*, volume 1769, pages 292–305. SPIE–The International Society for Optical Engineering, 1992.

[9] X. Huang, M. Fisher, and D. Smith. An Efficient Implementation of Max Tree with Linked List and Hash Table. In *VIIth Digital Imaging Computing: Techniques and Applications*, pages 299–308, Sydney, December 2003.

[10] R. Jones. Connected Filtering and Segmentation Using Component Trees. *Computer Vision and Image Understanding*, 75:215–228, 1999.

[11] J. Mattes and J. Demongeot. Efficient Algorithms to Implement the Confinement Tree. In 9th *International Conference on Discrete Geometry for Computer Imagery*, pages 392–405, London, UK, 2000. Springer-Verlag.

[12] J. Mattes, R. Mathieu, and J. Demongeot. Tree Representation for Image Matching and Object Recognition. In 8th *International Conference on Discrete Geometry for Computer Imagery*, pages 298–312, Marne-la-Vallee, France, March 1999.

[13] F. Meyer. The dynamics of minima and contours. In *Mathematical Morphology and Its Applications to Image Processing*, pages 329–336. Kluwer Academic Publishers, 1996.

[14] L. Najman and M. Couprie. Building the Component Tree in Quasi-Linear Time. *IEEE Transactions on Image Processing*, 15(11):3531–3539, 2006.

[15] G. K. Ouzounis and M. H. F. Wilkinson. Mask-Based Second-Generation Connectivity and Attribute Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):990–1004, 2007.

[16] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for filtering, segmentation and information retrieval. In *IEEE Int. Conference on Image Processing*, Proc. of ICIP'98, pages 4–7, Chicago (IL), USA, October 1998.

[17] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive Connected Operators for Image and Sequence Processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

[18] R. E. Tarjan. Efficiency of a Good But Not Linear Set Union Algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[19] C. Tzafestas and P. Maragos. Shape Connectivity: Multiscale Analysis and Application to Generalized Granulometries. *Journal of Mathematical Imaging and Vision*, 17:109–129, 2002.

[20] C. Vachier. *Extraction de caractéristiques, segmentation d'images et morphologie mathématique*. PhD thesis, École des Mines, Paris, 1995.

[21] C. Vachier and F. Meyer. Extinction value: a new measurement of persistence. In *IEEE Workshop on Nonlinear Signal and Image Processing*, volume I, pages 254–257, 1995.

[22] M. Wilkinson and J. Roerdink. Fast morphological attribute operations using tarjan's union-find algorithm. In *Mathematical Morphology and its Applications to Image and Signal Processing, Kluwer*, pages 311–320, 2000.