HAL open science

Component Tree Loss Function: Definition and Optimization

Benjamin Perret, Jean Cousty

To cite this version:

Benjamin Perret, Jean Cousty. Component Tree Loss Function: Definition and Optimization. DGMM 2022: Discrete Geometry and Mathematical Morphology, 2022, Strasbourg, France. pp.248–260. hal-03115362v2

HAL Id: hal-03115362

https://hal.science/hal-03115362v2

Submitted on 5 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Component Tree Loss Function: Definition and Optimization

Benjamin Perret[1 [0000−0003−0933−8342]] and Jean Cousty[1 [0000−0002−2163−9714]]

LIGM, Univ Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée
{benjamin.perret,jean.cousty}@esiee.fr

Abstract. In this article, we propose a method for designing loss functions based on component trees that can be optimized by gradient descent algorithms and are therefore usable in conjunction with recent machine learning approaches such as neural networks. The nodes of this tree are the connected components of the upper level sets of an image and the leaves represent the regional maxima (or regional minima if the dual tree is considered) of the image, *i.e.*, connected sets of bright pixels surrounded by darker pixels. The proposed loss function is thus defined at the level of connected components rather than at the level of individual pixels, which allows for the optimization of higher semantic level quantities such as topological features. We show how the altitudes associated with the nodes of such hierarchical image representations can be differentiated with respect to the values of the image pixels. This property is used to design a generic loss function that can select or discard image maxima based on various attributes, such as extinction values based on the contrast or the size of the maxima. The possibilities of the proposed method are demonstrated on simulated and real image filtering.

Keywords: max-tree · connected filters · topological loss · continuous optimization · mathematical morphology

1 Introduction

Component-trees are hierarchical image representations that are used to perform connected image analysis and filtering [19,13]. In such methods, an image is seen as the collection of the connected components of its level sets, thus offering a representation based on elements of higher semantic level, connected components instead of pixels, to design new image analysis methods. These approaches provide efficient solutions to many image analysis problems such as feature detection [8,23], segmentation [19,13,7,22,18], or object detection [11,20].

However, those methods, based on topological decompositions, do not play well with recent machine learning approaches such as neural networks as their combinatorial nature is, at first sight, not well suited to optimization strategies based on gradient descent. In this context, some authors have recently proposed topological loss functions [5,12,4,9] that enable to enforce topological constraints in continuous optimization frameworks using notions coming from the persistent

homology theory. It has also been shown that hierarchies of segmentations can also be used in such context with the introduction of an ultrametric layer [3].

In this article, we propose a novel approach to use component trees, and more specifically max-trees, within continuous optimization methods. This approach is based on the observation that, in such trees, the altitude of a node (the level of the level-set where it first appears) is directly linked to the value of some pixels of the image. Hence, we study how we can back-propagate any slight modification of the altitude of a node of the tree to a slight modification of the initial image. We then design a component tree loss function that enforces the presence of a prescribed number of maxima in the image based on maxima measures. We study how extinction values [21], maxima measures notably used in mathematical morphology to define hierarchical watersheds [6,17], can be used to modify the behavior of the proposed loss function. Finally, the method has been implemented in Pytorch with the hierarchical graph processing library Higra [16] and we provide preliminary results demonstrating the use of the proposed approach on simulated and real images.

This article is organized as follows. The definition of max-trees is recalled in Sec. 2. Then, Sec. 3 presents how max-trees can be used in gradient descent algorithms and formalizes the optimization problem which we address. In Sec. 4, we define a component tree loss function used for maxima selection in the max-trees and we introduce different maxima measures. The experiments are presented in Sec. 5. Finally, Sec. 6 concludes the work and gives some perspectives.

2 Max-trees

In this section, we recall the definition of max-trees [19,13] which is based on the decomposition of every upper thresholds of an image into connected components.

In the following, the image domain is represented by a finite nonempty set $V = \{v_i\}_{i \in [\![1,n]\!]}$ of cardinality n. The elements of V are called *pixels*. Given any vector $\mathbf{v}$ of $\mathbb{R}^m$ with $m \in \mathbb{N}^+$, the i-th component of $\mathbf{v}$ is denoted $\mathbf{v}_i$. An *image* is represented by a vector $\mathbf{f} \in \mathbb{R}^n$ and, for any $i \in [\![1,n]\!]$, f_i is called *the value of the pixel* v_i. Note that any image can be represented as a vector by choosing an arbitrary ordering of the pixels (*e.g.*, a raster scan for 2d images) and that this choice does not change the results of the proposed method. In order to simplify notations, when we have a vector $\mathbf{f} \in \mathbb{R}^n$ and an element x of a family $\{x_i\}_{i \in [\![1,n]\!]}$ indexed from 1 to n, there exists a single integer $k \in [\![1,n]\!]$ such that $x = x_k$ and we will write $\mathbf{f}_x$ instead of $\mathbf{f}_k$.

Let X be a subset of V, the set of connected components of X is denoted by $\mathcal{CC}(X)$ where connected components may be defined by any appropriate mean: *e.g.*, by path connectivity in a graph. In this article, all the examples involving 2d images are based on a classical 8-adjacency relation on a regular square grid of pixels. Let $\mathbf{f} \in \mathbb{R}^n$ be an image, the set of connected components of $\mathbf{f}$, denoted by $\mathcal{CC}(\mathbf{f})$, is defined by $\mathcal{CC}(\mathbf{f}) = \bigcup_{\lambda \in \mathbb{R}} \{\mathcal{CC}([\mathbf{f}]_\lambda)\}$ where, for any $\lambda \in \mathbb{R}$, $[\mathbf{f}]_\lambda$ is *the upper level set of* $\mathbf{f}$ *of level* λ: $[\mathbf{f}]_\lambda = \{v_i \in V \mid \mathbf{f}_i \geq \lambda\}$. Note that the set $\mathcal{CC}(\mathbf{f})$ is finite and can thus be indexed by integers: $\mathcal{CC}(\mathbf{f}) = \{C_i\}_{i \in [\![1,m]\!]}$, where m is

the number of connected components of $\mathbf{f}$. Let C_i in $\mathcal{CC}(\mathbf{f})$, the *altitude of* C_i is defined as the largest level λ in $\mathbb{R}$ such that C_i is a connected component of the upper level set of $\mathbf{f}$ at level λ: *i.e.*, $\max\left\{\lambda \in \mathbb{R} \mid C_i \in \mathcal{CC}([\mathbf{f}]^\lambda)\right\}$.

Let $\mathbf{f} \in \mathbb{R}^n$ be an image. The *max-tree* $\mathrm{MT}(\mathbf{f})$ *of* $\mathbf{f}$ is the pair $(\{C_i\}_{i\in[\![1,m]\!]}, \mathbf{a})$ where $\{C_i\}$ is the set of connected components of $\mathbf{f}$ and where $\mathbf{a}$ is a vector of $\mathbb{R}^m$ such that $\mathbf{a}_i$ is equal to the altitude of C_i. The first element of the pair, denoted by $\mathrm{MT}_1(\mathbf{f})$, is called the *hierarchy of* $\mathrm{MT}(\mathbf{f})$. The second element of the pair, denoted by $\mathrm{MT}_2(\mathbf{f})$, is called the *altitude vector of* $\mathrm{MT}(\mathbf{f})$. An example is given in Fig. 1. An element of the hierarchy $\mathcal{H}$ of $\mathrm{MT}(\mathbf{f})$ is called *a node of* $\mathcal{H}$. The node V includes every node of $\mathcal{H}$ and is called the *root*. Let C_i and C_j be two distinct nodes of $\mathcal{H}$. We say that C_i is an ancestor of C_j if C_j is included in C_i. Furthermore, if C_i is an ancestor of C_j, we say that C_i *is a parent of* C_j and that C_j *is a child of* C_i if any ancestor $C_k \neq C_i$ of C_j is also an ancestor of C_i. Any non-root node C_i of $\mathcal{H}$ has a unique parent which is denoted by $\mathrm{par}(C_i)$. The set of children of a node C_i of $\mathcal{H}$ is denoted by $\mathrm{Ch}(C_i)$. A node C_i of $\mathcal{H}$ is called a *leaf* if it has no child. There is a bijection between the leaf nodes of the hierarchy of $\mathrm{MT}(\mathbf{f})$ and the (regional) maxima of $\mathbf{f}$. For any node C_i of $\mathcal{H}$, a pixel v in C_i that is not contained in any child of C_i is called *a proper pixel of* C_i. Any element v of V is a proper pixel of a unique node C_i denoted $\mathrm{par}(v)$.

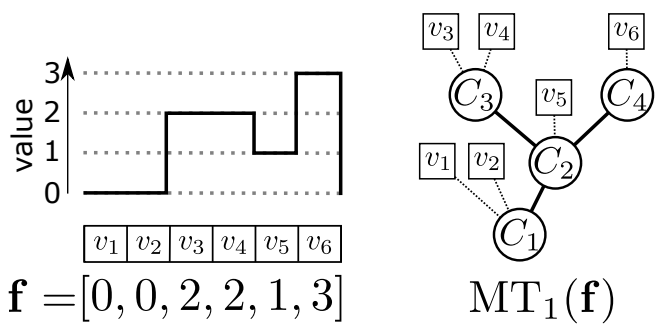

Fig. 1. Max-tree example. The left figure shows a 1d image $\mathbf{f} \in \mathbb{R}^6$ defined on the domain $v_1, \ldots, v_6$. Each of the four level sets at levels 0, 1, 2, and 3, has a single connected component $C_1, \ldots, C_4$. Those components are the nodes of hierarchy $MT_1(\mathbf{f})$ (circles) shown on the right image. The plain lines represent the parent relations between nodes. The proper elements of each node are depicted by squares and the dashed lines show the parent relation between those proper elements and their respective nodes, for example we have $\mathrm{par}(v_4) = C_3$. The altitude vector $\mathbf{a} = \mathrm{MT}_2(\mathbf{f})$ of the max-tree of $\mathbf{f}$ is equal to $[0, 1, 2, 3]$, meaning for example that the altitude of the node C_2 is equal to 1. The two maxima of $\mathbf{f}$ correspond to the leaf nodes C_3 and C_4 of the hierarchy $\mathrm{MT}_1(\mathbf{f})$.

3 Optimization with differentiable max-trees

In this section, we first study how the altitude vector of a max-tree can be sub-differentiated with respect to the pixel values, then we state the general formulation of the optimization problem which is addressed.

Differentiable max-trees. Trees, as combinatorial structures, are generally not suited to gradient-based optimization. However, in max-trees, the altitude of a component is mapped to the value of some pixels of the base image: its proper pixels. Then, intuitively, a small modification of the values of those proper pixels will not change the hierarchy associated with the max-tree of the image and will produce the exact same modification of the altitude of the corresponding node of the hierarchy.

Property 1 *Let $\mathbf{f} \in \mathbb{R}^n$ be an image. Let $\epsilon \in \mathbb{R}^n$ such that $\mathrm{MT}_1(\mathbf{f}) = \mathrm{MT}_1(\mathbf{f} + \epsilon)$. Then, for any node C_i of $\mathrm{MT}_1(\mathbf{f})$, the altitude of C_i in $\mathrm{MT}(\mathbf{f} + \epsilon)$ is equal to $\mathbf{a}_i + \epsilon_j$ where $\mathbf{a}_i$ is the altitude of C_i and where v_j is any proper pixel of C_i.*

This property indicates that the Jacobian of the function MT_2 can be written as the matrix composed of the indicator column vectors giving the index of the node associated with any pixel of V by the parent mapping (its proper elements):

$$\frac{\partial \mathrm{MT}_2(\mathbf{f})}{\partial \mathbf{f}} = \left[\mathbb{1}_{\mathrm{par}(v_1)}, \ldots, \mathbb{1}_{\mathrm{par}(v_n)} \right], \tag{1}$$

where $\mathbb{1}_{C_k}$ is the column vector of $\mathbb{R}^m$ equals to 1 in position k, and 0 elsewhere. In a back-propagation algorithm, this means that if we have an error measure e and we have already computed $\frac{\partial e}{\partial \mathbf{a}}$, *i.e.*, how the altitude vector $\mathbf{a} = \mathrm{MT}_2(\mathbf{f})$ of the max-tree of $\mathbf{f}$ should be modified in order to minimize e, we can then back-propagate through MT with the chain rule $\frac{\partial e}{\partial \mathbf{f}} = \frac{\partial \mathbf{a}}{\partial \mathbf{f}} \frac{\partial e}{\partial \mathbf{a}}$ leading to the simple formula $\left(\frac{\partial e}{\partial \mathbf{f}}\right)_i = \left(\frac{\partial e}{\partial \mathbf{a}}\right)_{\mathrm{par}(i)}$ telling how $\mathbf{f}$ should be modified to minimize e.

For example, the transpose of the Jacobian of the altitude vector $\mathrm{MT}_2(\mathbf{f})$ of the max-tree shown in Fig. 1 is equal to

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 & \mathbf{f}_4 & \mathbf{f}_5 & \mathbf{f}_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

This matrix indicates, how the image $\mathbf{f}$ should be modified in order to reflect a modification of the altitude vector $\mathbf{a}$ of the nodes of the max-tree of $\mathbf{f}$. For example, in order to increase the altitude $\mathbf{a}_3$ of the component C_3 by a small value ϵ, one must increase the value of $\mathbf{f}_3$ and $\mathbf{f}_4$ by this same value ϵ.

Optimization problem. We now state a general formulation of the optimization problem that we want to solve. Let $\mathbf{y} \in \mathbb{R}^n$ be an image representing an observation. We are interested in solving the following optimization problem:

$$\underset{\mathbf{f} \in \mathbb{R}^n}{\text{minimize}}\ J(\mathbf{f}; \mathbf{y}), \qquad \text{s.t. } 0 \leq \mathbf{f} \leq 1, \tag{2}$$

where J is a differentiable cost function involving the altitude vector $\mathrm{MT}_2(\mathbf{f})$. As this altitude vector $\mathrm{MT}_2(\mathbf{f})$ is differentiable with respect to the image $\mathbf{f}$, a local optimum of the above problem can be found by gradient descent algorithm. The constraint to keep the image between the values 0 and 1 will prevent it to shrink (resp. grow) towards $-\infty$ (resp. ∞) when we try to remove (resp. increase) some image features. A similar effect could be achieved by imposing margins on how much a feature can shrink or grow.

4 Maxima loss

In the following, we study how to define a component tree loss imposing a topological criterion, by prescribing how many maxima should be present in the result. The proposed approach relies on two features characterizing the maxima:

- *a measure of saliency*: increasing this measure should reinforce the maximum and decreasing it should make it disappear; and
- *a measure of (relative) importance*: which provides a ranking of the maximum to identify those that should be reinforced and those which should disappear.

We first introduce a loss function to *select* a given number of maxima and to *discard* the others according to these two measures. Then, we present several measures to assess the importance and the saliency of maxima.

4.1 Ranked selection loss

Assume that the hierarchy $\mathcal{H}$ of $\mathrm{MT}(\mathbf{f})$ contains k maxima $\{M_i\}_{i=[\![1,k]\!]}$ (its leave nodes). Let $\ell \in \mathbb{N}^+$ be a target number of maxima. Let $\mathbf{sm} \in {\mathbb{R}^+}^k$ and $\mathbf{im} \in {\mathbb{R}^+}^k$ represent respectively a saliency and an importance measure on the maxima $\{M_i\}$. Let $p \in \mathbb{R}^{+*}$ and $q \in \mathbb{R}^{+*}$ be 2 strictly positive numbers controlling the growth/shrink pressure applied on the maxima. The ranked selection function will seek to maximize the saliency of the ℓ maxima with the largest importance values and decrease the saliency of the others:

$$J_{p,q}(\mathbf{sm}, \mathbf{im}; \ell) = -\sum_{i=1}^{i \leq \ell} \mathbf{sm}_{\mathbf{r}_i}^p + \sum_{i=\ell+1}^{i \leq k} \mathbf{sm}_{\mathbf{r}_i}^q \text{ with } \mathbf{r} = \mathrm{argsort}(\mathbf{im}), \tag{3}$$

where argsort is the function that associates any vector $\mathbf{v}$ of $\mathbb{R}^k$ with a permutation vector $\mathbf{r} \in [\![1, n]\!]^k$ sorting the elements of $\mathbf{v}$ in decreasing order, *i.e.*, such that for any i, j in $[\![1, n]\!]$, we have $i < j \Rightarrow v_{\mathbf{r}_i} \geq v_{\mathbf{r}_j}$.

4.2 Maxima measures

We now define maxima measures that will be used as saliency and/or importance measures in the previous loss function. Recall that the leaves of the max-tree

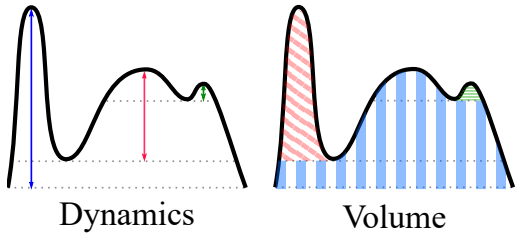

Fig. 2. Illustration of the dynamics and the volume associated with the maxima of a 1d function. The dynamics of a maximum is equal to the difference of altitude between the top of the maximum and the closest level that contains another maximum of greater altitude. Similarly, the volume of the maximum is equal to the surface between the top of the maximum and the closest level that contains another maximum of greater volume. With the dynamics, the most important maximum is the sharp peak on the left while, with the volume, the most important maximum is the large mount in the middle.

of an image $\mathbf{f}$ corresponds to the maxima of this image and assume that the hierarchy $\mathcal{H}$ of $\mathrm{MT}(\mathbf{f})$ contains k maxima $\{M_i\}_{i=[\![1,k]\!]}$; a measure on $\{M_i\}$ is then a positive vector of $\mathbb{R}^k$.

Maxima altitude: A simple way to measure the importance and the saliency of a maximum is to look at its altitude, *i.e.*, to the altitude $\mathbf{a}_M$ of the maximum M of $\mathcal{H}$. The altitude of the maxima of $\mathbf{f}$ is denoted $\mathbf{alt}(\mathbf{f})$.

Extinction values: Extinction values are classical maxima measures known for their robustness [21]. Given a family of image filters $\{\sigma_k\}_k$ whose activity increases with k (for any $k_1 \leq k_2$, we have $\sigma_{k_1} \geq \sigma_{k_2}$). The extinction value of a maximum M of $\mathbf{f}$ is equal to the smallest k such that M is not contained in any maximum of $\sigma_k(\mathbf{f})$. Typical examples of extinction values are the *dynamics* and the *volume*. The dynamics and volume associated with the maxima of a function are illustrated in Fig. 2. In the following, we will show how extinction values can be defined and computed based on the max-tree representation.

Any extinction value relies on an increasing attribute measuring the importance of regional maxima. Such attribute will be represented by a vector $\mathbf{v} \in \mathbb{R}^m$ associating a value to each node of the max-tree such that for any two nodes C_i and C_j, $C_i \subseteq C_j \Rightarrow \mathbf{v}_i \leq \mathbf{v}_j$. The idea to compute the extinction value of a maximum for the attribute $\mathbf{v}$ is then to find, for any maximum M, the closest ancestor of M that contains another maximum whose attribute value is greater than the one of M: this node is called the saddle node associated with M for $\mathbf{v}$.

Formally, let N be a node of $\mathcal{H}$ and let A be an ancestor of N. There exists a single branch rooted in A that contains the node N, the child of A in this branch is denoted by $\mathrm{ch}(A)_{\to N}$; in other words $\mathrm{ch}(A)_{\to N}$ is the only child C of A that contains N. The saddle node associated with the node N for the attribute $\mathbf{v}$, denoted by $\mathrm{saddle}_{\mathbf{v}}(N)$, is the closest ancestor A of N such that there exists a child C of A with $\mathbf{v}_{\mathrm{ch}(A)_{\to N}} < \mathbf{v}_C$. If no such ancestor exists, then the saddle node of N for $\mathbf{v}$ is defined as the empty-set $\emptyset$. The base node

associated with the node N for the attribute $\mathbf{v}$, denoted by $\text{base}_{\mathbf{v}}(N)$, is then equal to $\text{ch}(\text{saddle}_{\mathbf{v}}(N))_{\rightarrow N}$ if $\text{saddle}_{\mathbf{v}}(N) \neq \emptyset$ and the root of the max-tree otherwise.

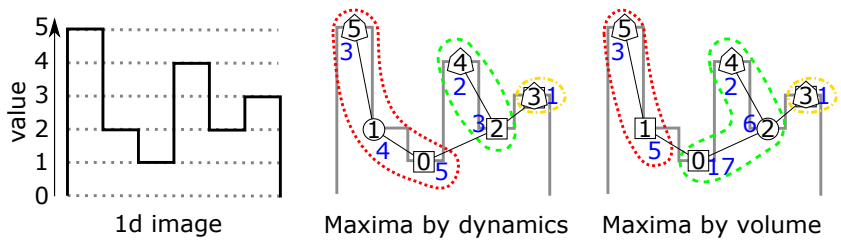

Fig. 3. Regional maximum associated with each maximum for the dynamics and the volume. We consider a 1d image on the left with 3 maxima. The second (resp. third) figure shows how the 1d image is decomposed into 3 regional maxima according to the dynamics (resp. the extinction by volume). In those two images, we see the max-tree of the 1d image where the nodes are depicted by circles, squares and hexagons. The label of each node corresponds to its index, and the blue value beside it corresponds to its attribute value: its depth for the dynamics and its volume for the extinction by volume. The hexagons are the maxima, *i.e.*, the leaves of the tree. Each maximum is associated with a branch of the tree, circled in red (dotted line), green (dashed line) and orange (dot-dash line) whose base node is depicted by a square. For example, for the dynamics (resp. the extinction by volume), the maximum of index 5 extends to its base node 0 (reps. 1); its dynamics (resp. extinction by volume) is thus equal to 5, *i.e.*, the attribute value of this base node.

Thus, each regional maximum extends from its top node M to its base node $\text{base}_{\mathbf{v}}(M)$, and the saddle node $\text{saddle}_{\mathbf{v}}(M)$ is the first ancestor of M that belongs to another maximum according to the attribute $\mathbf{v}$. The extinction value of the maximum M for the attribute $\mathbf{v}$ is then defined as the attribute value of its base node $\mathbf{v}_{\text{base}_{\mathbf{v}}(M)}$, see Figure 3. Note that the definition of the extinction value for the attribute $\mathbf{v}$ is just a value selection process in a vector, as in a max-pooling layer, and it can thus be used in the definition of a loss function suitable for gradient descent optimization.

In the following, we consider two different attributes: the depth, whose associated extinction value is usually called the dynamics, and the volume. Both attributes will be defined as a function of the max-tree altitudes, so that any error on the extinction value can be translated as an error on those altitudes.

Let N be a node of $\mathcal{H}$, the depth of N, denoted by $\text{depth}(N)$, is defined by:

$$\text{depth}(N) = \max\{\mathbf{a}_C, C \in \mathcal{H} \mid C \subset N\} - \mathbf{a}_{\text{par}(N)}. \tag{4}$$

The depth of N is thus equal to the difference between the largest altitude in the subtree rooted in N and the altitude of the parent of N. The extinction values of the maxima of $\mathcal{H}$ for the attribute depth is called the dynamics and will be denoted by $\mathbf{dyn}$, see Figure 3.

Let N be a node of $\mathcal{H}$, the volume of N, denoted by $\mathrm{vol}(N)$, is defined recursively by:

$$\mathrm{vol}(N) = |N| \cdot \left(\mathbf{a}_N - \mathbf{a}_{\mathrm{par}(N)}\right) + \sum_{C \in \mathrm{Ch}(N)} \mathrm{vol}(C), \tag{5}$$

where $|N|$ denotes the cardinal of N, *i.e.*, the number of pixels in the node N. The volume of N is thus equal to the volume of the cylinder defined by the node N and its parent, plus the volume of its children. The extinction values of the maxima of $\mathcal{H}$ for the attribute vol is called the extinction value by volume and will be denoted by **vol**, see Figure 3.

Effect of modifying the saliency: The ranked selection loss (3) will try to increase the saliency measure of the selected maxima and decrease the one of the others. In order to better understand how this will affect the result, we propose to study how a single maximum is modified when we try to increase/decrease its saliency according to one of the proposed saliency measures.

In the case of the saliency based on the maxima altitudes, this effect is simple, as increasing (resp. decreasing) the altitude of a maximum simply means increasing (resp. decreasing) the altitude of the leaf node that corresponds to this maximum in the max-tree.

With the dynamics, the saliency of a maximum is determined by the altitude of the leaf node that corresponds to this maximum in the max-tree and by the altitude of the saddle node associated with this leaf for the depth attribute. In this case, increasing (resp. decreasing) the dynamics of a maximum means increasing (resp. decreasing) the altitude of its leaf node and decreasing (resp. increasing) the altitude of the saddle node (see Figure 4). Note that the altitudes of all the nodes between the leaf node and the saddle node are not modified.

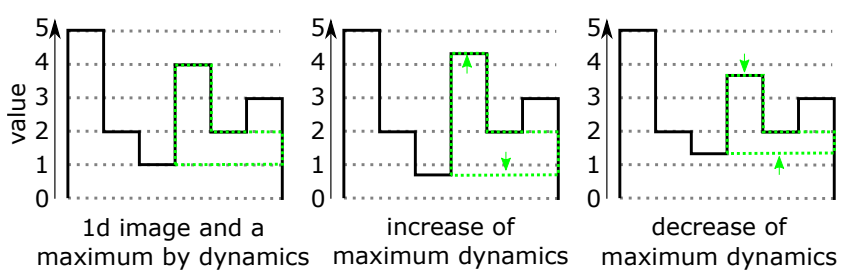

Fig. 4. Effect of increasing or decreasing the saliency measure of a maximum based on the dynamics **dyn**. The first figure shows a 1d image and a regional maximum for the dynamics (see Figure 2). The second (resp third) figure shows the effect of increasing (resp decreasing) the measure for this maximum.

Finally, with the extinction value by volume, the saliency of a maximum is determined by the altitudes of all the nodes in the branch going from the leaf node that corresponds to this maximum in the max-tree to the saddle node

associated with this leaf for the volume attribute. In this case, increasing (resp. decreasing) the extinction value by volume of a maximum means increasing (resp. decreasing) the altitudes of all the nodes in the branch going from the leaf node to the base node and decreasing (resp. increasing) the altitude of the saddle node (see Figure 5).

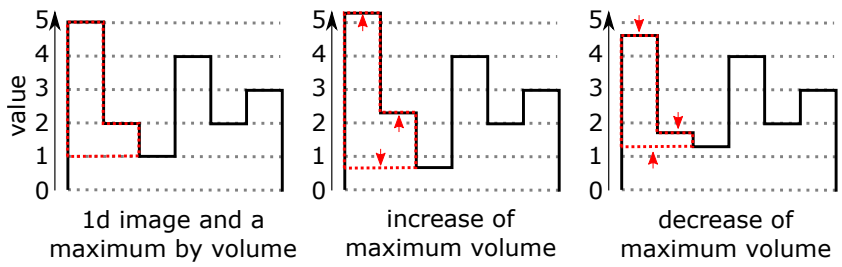

Fig. 5. Effect of increasing or decreasing the saliency measure of a maximum based on the extinction value by volume **vol**. The first figure shows a 1d image and a regional maximum for the volume extinction value (see Figure 2). The second (resp third) figure shows the effect of increasing (resp decreasing) the measure for this maximum.

Note that in both cases, modifying the saliency of a single maximum usually preserves the ordering of the nodes in the tree; an inversion between the top node and the base node may still happen when a maximum collapses. However, when the saliency of several maxima is modified at the same time, it becomes more probable that the ordering of the nodes in the tree changes, leading to more complex topological modifications in the image domain.

5 Experiments

We demonstrate the behavior of the proposed method and the various maxima measures on a simulated image, and we show how it can be combined with classical loss functions to process real images. The method is implemented using the library Higra [16] for hierarchical graph analysis in combination with the continuous optimization framework Pytorch [15]. In all the experiments, an Adam optimizer [14] is used and the input image $\mathbf{y}$ is used as the initial solution. A Jupyter notebook containing the presented experiments is available online[1].

Simulated image filtering: The effect of the optimization of the component tree loss $J_{1,2}(\mathbf{sm}, \mathbf{im}; 2)$ with the proposed importance and saliency maxima measures is demonstrated on a simulated image in Fig. 6. The test image contains four maxima with different altitudes, contrast, and volumes. We can see that the altitude measure is not robust to noise and fails to select perceptually significant maxima: both are located in the bottom left blob. On the other

[1] `https://www.esiee.fr/~perretb/notebooks/Component_Tree_Loss.zip`

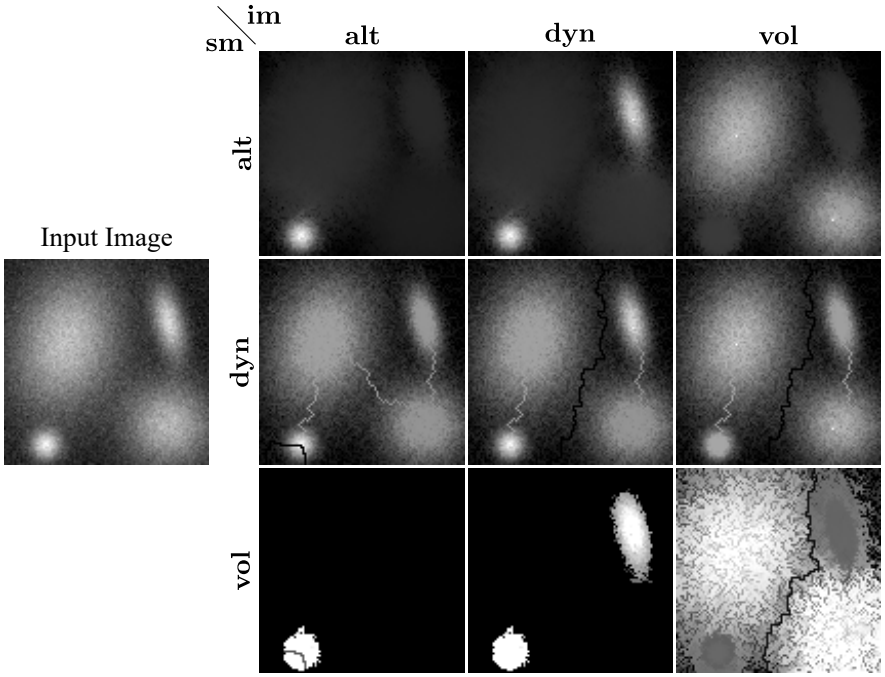

Fig. 6. Optimization of the component tree loss $J_{1,2}(\mathbf{sm}, \mathbf{im}; 2)$ on a simulated image with the objective of selecting 2 maxima for various combinations of saliency measures (**sm**) and importance measures (**im**).

hand, the two measures based on extinction values, the dynamics and the volume, both manage to select significant maxima: with the dynamics, the two brightest maxima are selected (bottom left and the top right blobs) while with the volume, the two largest maxima are selected (top left and bottom right blobs). Regarding saliency measures, we can see that the optimization of the saliency based on maxima altitudes leads to increasing the altitudes of the top node of the selected maxima and to raising discarded maxima. The optimization of the dynamics saliency measure is more complex, as increasing/decreasing the dynamics of the maxima involves increasing/decreasing the altitude of its top node and decreasing/increasing the altitudes of its saddle node: this leads to the creation of "bridges" between some maxima. Finally, the optimization of the volume saliency has a more global effect on the maxima as, contrarily to the dynamics, its definition involves the altitudes of all the nodes between the base and the top nodes of a maximum.

In order to work with the saliency measure based on volume, we have observed that raising the power q to 2 inside the loss function can help to erase the unwanted maxima whose contribution to the loss value and hence to the gradient tends to be smaller than the one of the largest maxima.

Note that optimizing J with the dynamics used both as the saliency and the importance measure of maxima is similar [2] to optimizing the barcode length of the connected components used in the "topological loss" based on persistent homology [5,12,4,9]. However, as our approach does not require computing the full persistence diagram associated with the image at each iteration of the optimization algorithm, we observe that it is faster than a classical implementation of the topological loss[2].

Real image filtering: Finally, in Fig. 7, we show how the proposed loss function can be combined with classical loss functions used in image analysis: here we optimize the term

$$||\mathbf{f} - \mathbf{y}||_2^2 + \lambda_1 J_{1,1}(\mathbf{dyn}(\mathbf{f}), \mathbf{dyn}(\mathbf{f}), 1) + \lambda_2 ||\nabla \mathbf{f}||_2^2, \tag{6}$$

which combines our loss based on the max-tree to enforce the presence of a single maximum with a L2 data attachment term and a total variation regularization term. We can see that we are able to successfully reconnect the different branches of the neurite.

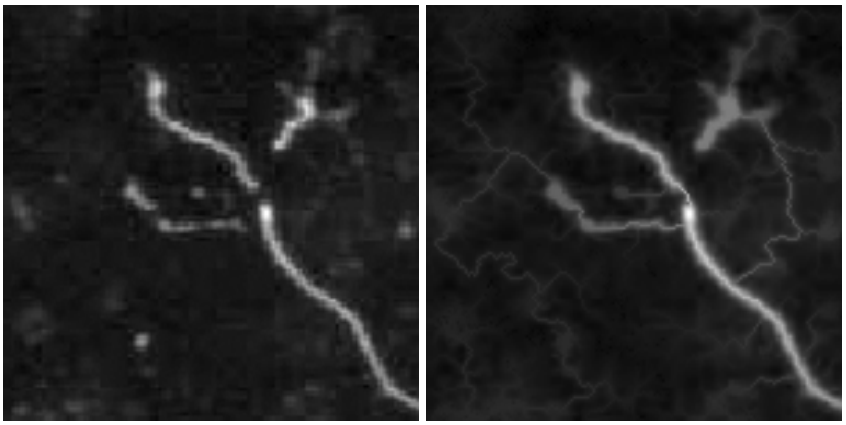

Image Result

Fig. 7. Reconnection of a neurite image with a combination of the proposed loss J_r to enforce a single maximum, a L2 data attachment term and a TV2 regularization term.

6 Conclusion

We have proposed a continuous optimization framework based on the hierarchical image representation called the max-tree. We showed how it can be used to design a component tree loss, *i.e.*, a regularization term, enabling to select or discard maxima in an image based on various measures. This approach can be generalized

[2] `https://github.com/bruel-gabrielsson/TopologyLayer`

immediately to other hierarchical representations, such as the min-tree or the tree-of-shapes [1,10]. In future works, we plan to explore more general component tree loss functions based on such hierarchical representations and their use in supervised learning methods involving deep networks.

Acknowledgements This work was supported by the French ANR grant ANR-20-CE23-0019.

References

1. Ballester, C., Caselles, V., Monasse, P.: The tree of shapes of an image. ESAIM: Control, Optimisation and Calculus of Variations **9**, 1–18 (2003)
2. Boutry, N., Géraud, T., Najman, L.: An equivalence relation between morphological dynamics and persistent homology in 1d. In: ISMM. pp. 57–68. Springer (2019)
3. Chierchia, G., Perret, B.: Ultrametric fitting by gradient descent. In: NeurIPS. pp. 3181–3192 (2019)
4. Clough, J., Byrne, N., Oksuz, I., Zimmer, V., Schnabel, J., King, A.: A topological loss function for deep-learning based image segmentation using persistent homology. IEEE TPAMI (2020)
5. Clough, J., Oksuz, I., Byrne, N., Schnabel, J., King, A.: Explicit topological priors for deep-learning based image segmentation using persistent homology. In: IPMI. pp. 16–28. Springer (2019)
6. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: ISMM. pp. 272–283. Springer (2011)
7. Dalla Mura, M., Benediktsson, J., Waske, B., Bruzzone, L.: Morphological attribute profiles for the analysis of very high resolution images. IEEE TGRS **48**(10), 3747–3762 (2010)
8. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In: IEEE CVPR. vol. 1, pp. 553–560 (2006)
9. Gabrielsson, R., Nelson, B., Dwaraknath, A., Skraba, P.: A topology layer for machine learning. In: AISTATS. pp. 1553–1563. PMLR (2020)
10. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of nd images. In: ISMM. pp. 98–110. Springer (2013)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CVPR. pp. 580–587 (2014)
12. Hu, X., Li, F., Samaras, D., Chen, C.: Topology-preserving deep image segmentation. In: NeurIPS. pp. 5657–5668 (2019)
13. Jones, R.: Connected filtering and segmentation using component trees. CVIU **75**(3), 215–228 (1999)
14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2015)
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS. pp. 8026–8037 (2019)
16. Perret, B., Chierchia, G., Cousty, J., Guimarães, S., Kenmochi, Y., Najman, L.: Higra: Hierarchical graph analysis. SoftwareX **10**, 100335 (2019)

17. Perret, B., Cousty, J., Guimaraes, S., Maia, D.: Evaluation of hierarchical watersheds. IEEE TIP **27**(4), 1676–1688 (2017)
18. Robic, J., Perret, B., Nkengne, A., Couprie, M., Talbot, H.: Self-dual pattern spectra for characterising the dermal-epidermal junction in 3d reflectance confocal microscopy imaging. In: ISMM. pp. 508–519. Springer (2019)
19. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE TIP **7**(4), 555–570 (1998)
20. Teeninga, P., Moschini, U., Trager, S., Wilkinson, M.: Statistical attribute filtering to detect faint extended astronomical sources. Mathematical Morphology-Theory and Applications **1** (2016)
21. Vachier, C., Meyer, F.: Extinction value: a new measurement of persistence. In: IEEE Workshop on nonlinear signal and image processing. vol. 1, pp. 254–257 (1995)
22. Xu, Y., Carlinet, E., Géraud, T., Najman, L.: Hierarchical segmentation using tree-based shape spaces. IEEE TPAMI **39**(3), 457–469 (2016)
23. Xu, Y., Monasse, P., Géraud, T., Najman, L.: Tree-based Morse regions: A topological approach to local feature detection. IEEE TIP **23**(12), 5612–5625 (2014)