RISF: Recursive Iterative Spanning Forest for Superpixel Segmentation

Felipe L. Galvão
and Alexandre X. Falcão
Institute of Computing
University of Campinas
Campinas, SP, Brazil

Ananda S. Chowdhury
Dept. of Electronics and
Telecommunication Engineering
Jadavpur University
Kolkata, India

***Abstract*—Methods for superpixel segmentation have become very popular in computer vision. Recently, a graph-based framework named ISF (*Iterative Spanning Forest*) was proposed to obtain connected superpixels (supervoxels in 3D) based on multiple executions of the Image Foresting Transform (IFT) algorithm from a given choice of four components: a seed sampling strategy, an adjacency relation, a connectivity function, and a seed recomputation procedure. In this paper, we extend ISF to introduce a unique characteristic among superpixel segmentation methods. Using the new framework, termed as *Recursive Iterative Spanning Forest* (RISF), one can recursively generate multiple segmentation scales on region adjacency graphs (i.e., a hierarchy of superpixels) without sacrificing the efficiency and effectiveness of ISF. In addition to a hierarchical segmentation, RISF allows a more effective geodesic seed sampling strategy, with no negative impact in the efficiency of the method. For a fixed number of scales using 2D and 3D image datasets, we show that RISF can consistently outperform the most competitive ISF-based methods.**

I. Introduction

The partition of an image into a disjoint set of connected regions, named *superpixel segmentation*, has become a very active research topic in computer vision and image processing [1]. Such a simplified image representation can improve the performance of higher level image processing methods for several applications including: medical image segmentation [2], multi-class object detection and segmentation [3], [4], spatiotemporal saliency detection [5], target tracking [6], and depth estimation [7]. The challenge here is to preserve the object boundaries, such that an object of interest can be defined by the union of its superpixels, with low computational time. Due to that, boundary recall [8] and undersegmentation error [9] are often used as boundary adherence measures.

Many superpixel segmentation methods cannot guarantee connected superpixels without post-processing operations (e.g., [8], [10]–[13]), which affect the number of desired superpixels. Recently, a graph-based framework named ISF (*Iterative Spanning Forest*) [14], [15] was proposed to obtain connected superpixels (supervoxels in 3D) based on multiple executions of the Image Foresting Transform (IFT) algorithm [16], [17] for improved sets of seed pixels. ISF consists of four components: (a) a seed sampling strategy for the first execution of the IFT algorithm, (b) an adjacency relation that usually interprets the image as a 4-connected graph in 2D and 6-connected graph in 3D, (c) a connectivity function that defines the costs of the paths from the seed set to any node (pixel) of the graph, and (d) a seed recomputation procedure that estimates an improved seed set for each subsequent execution of the IFT algorithm. ISF essentially represents each connected superpixel as one *optimum-path tree* rooted at each seed pixel. By choice of those four components, the user can design different superpixel segmentation methods.

In this paper, we add a unique characteristic to ISF — the ability to construct a hierarchy of superpixels for exploitation in high-level applications. It can now recursively generate multiple segmentation scales on Region Adjacency Graphs (RAGs) with no negative impact in efficiency and effectiveness. The new framework is termed *Recursive Iterative Spanning Forest* (RISF). In addition to a hierarchical segmentation, RISF allows a more effective geodesic seed sampling strategy, with no negative impact in the efficiency of the method. The efficiency of ISF with that strategy would not be viable due to the number of graph nodes. For a fixed number of scales and using 2D and 3D datasets of natural and medical images, we show that RISF can consistently outperform the most competitive ISF-based methods.

II. Basic definitions and IFT

An n-dimensional image is a pair $(\mathcal{I}, \mathbf{I})$ that assigns a finite set $\mathbf{I}(t)$ of physical properties to each pixel (voxel, superpixel, supervoxel) $t \in \mathcal{I} \subset \mathcal{Z}^n$. For a given *adjacency relation* $\mathcal{A} \subset \mathcal{I} \times \mathcal{I}$, an image can be interpreted as a graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ where $\mathcal{I}$ is the set of nodes, $\mathcal{A}$ is the set of arcs, and $\mathbf{I}(t)$ assigns physical properties to each node $t \in \mathcal{I}$. In this work, for $n = 2$, $\mathcal{A}$ may define a 4-connected pixel graph or a Region Adjacency Graph (RAG), when the superpixels in $\mathcal{I}$ contain pixels that share one edge. Similarly, for $n = 3$, $\mathcal{A}$ may define a 6-connected voxel graph or a RAG, when the supervoxels in $\mathcal{I}$ contain voxels that share one face. Given two nodes $s, t \in \mathcal{I}$, we indicate that s is incident on t by $(s, t) \in \mathcal{A}$.

A *path* with terminus t is defined as a sequence of nodes $\pi_t = \langle t_1, t_2, \ldots, t_k = t \rangle$, where $(t_i, t_{i+1}) \in \mathcal{A}$ for $1 \leq i \leq k - 1$, being *trivial* when $k = 1$. A *path-cost function* $f(\pi_t)$ assigns a value to any path π_t. A path π_t is said to be optimum if for any other path π'_t we have $f(\pi_t) \leq f(\pi'_t)$. We cover specific instances of path-cost functions in Section III.

A *predecessor map* is a function that assigns each node $t \in \mathcal{I}$ to another node s such that $(s,t) \in \mathcal{A}$, or to a distinct marker *nil* $\notin \mathcal{I}$. In the latter case, t is said to be a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles. For any node $t \in \mathcal{I}$, a spanning forest P defines a path π_t^P recursively as $\langle t \rangle$ if $P(t) = nil$ and as the extension $\pi_s^P \cdot \langle s,t \rangle$ of π_s^P by an arc $(s,t) \in \mathcal{A}$, if $P(t) = s \neq nil$.

The Image Foresting Transform (IFT) [16] takes a graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ with path-cost function f and returns an *optimum-path forest* — a spanning forest where π_t^P is an optimum path for any $t \in \mathcal{I}$. The conditions over f to guarantee that P is optimum are defined in [17]. However, even when this is not the case, P is still useful in several applications [14], [18].

In superpixel segmentation, the roots of the spanning forest are forced to be a seed set $\mathcal{S} \subset \mathcal{I}$ by defining the following initial condition for the cost values of trivial paths:

$$f(\pi_t = \langle t \rangle) = \begin{cases} 0 & \text{if } t \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \quad (1)$$

This definition allows us to control the number $|\mathcal{S}|$ of generated spanning trees, each one being assigned to a distinct superpixel label.

Here we are interested in executing the IFT multiple times from the same graph while updating the seed set $\mathcal{S}$ in an efficient way to better capture image information (like other iterative methods do [8], [10]). For that, we rely on the differential IFT algorithm (DIFT) proposed in [19] for non-monotonically incremental path-cost functions. This algorithm updates the spanning forest for each seed set by processing only the nodes whose path costs change. We present a slightly modified version of this algorithm in Section V.

III. ISF AND RISF

In the Iterative Spanning Forest (ISF) framework [15], superpixel segmentation is formulated within the more general Image Foresting Transform (IFT) framework together with an iterative scheme similar to SLIC [8], as first presented in [14]. An ISF-based method consists primarily of: an initial sampling strategy, an adjacency relation, a path-cost function for the DIFT computation, and a seed recomputation procedure.

As defined in [15], an ISF-based method can operate over either pixels or voxels. Here we present a more general formalization which extends ISF to Region Adjacency Graphs (RAGs). We call this formalization Recursive Iterative Spanning Forest (RISF) as it calls ISF multiple times with a reducing number of regions (scales), creating a new RAG from the resulting regions for each subsequent execution, up to achieving the desired number of regions. The obtained regions (superpixels/supervoxels) per scale define a hierarchical segmentation for exploitation in a future work.

Given the first execution of ISF over the pixel/voxel graph, the resulting image partition into simply-connected regions is used to build the first RAG $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$, where each node $s_i \in \mathcal{I}$ now represents one region $\mathcal{R}_i$ of the previous pixel/voxel set $\mathcal{I}$. This process continues as described above, updating the graph G from region representations that become coarser after each execution of ISF. Assuming we are working in the CIELAB color space, each pixel $p = (x, y, z)$ has color $\mathbf{I}(p) = [l \;\; a \;\; b]$ and we extend it for regions as $\mathbf{I}(s_i) = \frac{1}{|\mathcal{R}_i|} \sum_{p \in \mathcal{R}_i} \mathbf{I}(p)$ (the mean color in region $\mathcal{R}_i$). In a similar manner, we define the geometric coordinate of a region as its centroid. It is possible to incorporate more complex region features (e.g., color histogram), but here we only consider the mean color for simplicity.

The summary of each ISF call is as follows. The initial sampling strategy determines the first seed set $\mathcal{S} \subset \mathcal{I}$. Given the path-cost function f, each seed $s \in \mathcal{S}$ becomes root of a spanning tree (Equation 1). The DIFT algorithm using G and f (Equation 2) computes a spanning forest, which is either used to obtain an improved seed set $\mathcal{S}'$ (Equations 3-4) or taken as final segmentation with the desired number of regions.

The standard initial seed sampling strategy consists of selecting seeds on a regular grid, similarly to various superpixel segmentation methods [8], [10], [20]. An alternative based on Shannon Entropy was considered in [15]. In this paper we present a novel seeding method detailed in Section IV.

For non-trivial paths, we consider the same path-cost function used in both [14] and [15]:

$$f(\pi_s \cdot \langle s,t \rangle) = f(\pi_s) + (\alpha \|\mathbf{I}(t), \mathbf{I}(R(s))\|)^\beta + \|s,t\|, \quad (2)$$

where $\alpha \geq 0$, $\beta \geq 1$, $\|\cdot,\cdot\|$ is the Euclidean norm, $R(s)$ is the root of node s (i.e., the first node in the path π_s), and $\mathbf{I}(R(s))$ is the mean color inside the region of the previous iteration that is now represented by $R(s)$ (or the color of $R(s)$ in the first iteration). The parameter β balances the regularity imposed by an additive path-cost function ($\beta = 1$) and the boundary adherence a $\max$-based path-cost function ($\beta \to \infty$). Similarly to [14] and [15], it is assumed a fixed $\beta = 12$. While β weights the edges along the whole path $\pi_s \cdot \langle s,t \rangle$, the parameter α provides a balance between the parametric and geometric terms of the edge weight, acting as a compactness parameter; lower α means more compact regions.

The seed recomputation procedure is equivalent to the one in [15]. Let $s_j \in \mathcal{I}$ be a seed at iteration j defining the region $\mathcal{R}_j$ of a spanning tree on the graph G. The seed s_j will be swapped by a new seed s_{j+1}, which is the node in $\mathcal{R}_j$ closest to its centroid, if:

$$\|\mathbf{I}(s_j), \mathbf{I}(s_{j+1})\| > \sqrt{\frac{1}{|\mathcal{R}_j|} \sum_{t \in \mathcal{R}_j} \|\mathbf{I}(s_j), \mathbf{I}(t)\|} \quad (3)$$

or

$$\|s_j, s_{j+1}\| > \sqrt{\frac{1}{|\mathcal{R}_j|} \sum_{t \in \mathcal{R}_j} \|s_j, t\|}. \quad (4)$$

Otherwise s_j is kept for the next iteration.

IV. Geodesic seed sampling

An important aspect of any seed-based algorithm is the initial location of each seed, which is defined by the seed sampling strategy in RISF. Inspired by the work in [21], which uses the Eikonal-based framework to iteratively select seeds with the maximum geodesic distance to the previous ones, we propose a new seed sampling strategy that explores the IFT framework in a similar manner.

The seed sampling strategy, which we refer to as *geodesic sampling*, is defined by the following recursive procedure: given a seed set $\mathcal{S}_i \subset \mathcal{I}$, an optimum-path forest P is computed over the graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ for path-cost function f_g,

$$\begin{aligned} f_g(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \in S_i, \\ +\infty & \text{otherwise,} \end{cases} \\ f_g(\pi_s \cdot \langle s, t \rangle) &= f_g(\pi_s) + \|\mathbf{I}(R(s)), \mathbf{I}(t)\|, \end{aligned} \tag{5}$$

and then the node $t = \arg\max_{t \in \mathcal{I}}\{f(\pi_t^P)\}$ is selected to build a new set $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{t\}$. We use $\mathbf{I}(R(s))$ as the color of the root $R(s)$ in Equation 5. The initial set $\mathcal{S}_0 \neq \emptyset$ may consist of any arbitrary node taken as seed (e.g., we use the node closest to the center of the image). The procedure is repeated until a desired number of seeds is reached.

Being additive, the path-cost function f_g prioritizes the selection of seeds that are properly spread over the image domain. At the same time, having the arc weight defined on the parametric space, it selects seeds in heterogeneous image regions, avoiding multiple seeds in regions with no image borders.

One observed side-effect of choosing seeds based on maximum cost is an increased chance of placing them over outlier nodes in the feature space. While the seed recomputation in RISF naturally fixes part of the problem, we note that the final segmentation tends to present some small regions representing those outliers. For that reason we also pair the geodesic sampling with an extra iteration of RISF where spanning trees with size below five percent of the average target region size (in number of underlying elements) are removed. This parameter was empirically observed to be a good compromise between capturing those outliers and not removing actual borders from the superpixel segmentation.

V. The RISF algorithm

In this section, we present the RISF algorithm using a slightly modified version of the differential IFT proposed in [19], also including the algorithm for our geodesic sampling strategy. While the IFT in Section II is fully described in terms of building a predecessor map P, additional maps are actually used to obtain other attributes of the forest. The cost map $C : \mathcal{I} \to \mathbb{R}$ stores the current best path cost $C(s) = f(\pi_s^P)$. The root map $R : \mathcal{I} \to \mathcal{I}$, briefly mentioned in Equations 2 and 5, keeps track of the root of the spanning tree that contains each node $s \in \mathcal{I}$, which means $R(s) = r$ when $\pi_s^P = \langle r, \ldots, s \rangle$. The RISF algorithm also allows the use of root-based path-cost functions that do not guarantee optimum paths, but provide effective region segmentation.

Input: Image $(\mathcal{I}, \mathbf{I})$, adjacency relation $\mathcal{A}$, path-cost function f, sequence $\langle n_1, n_2, \ldots, n_K \rangle$ with the number n_k of regions per scale, where $n_k > n_{k+1}, k = 1, 2, \ldots, K-1$ and n_K is the desired number of regions in the final segmentation, number of iterations T for ISF.
Output: Labeled image L with n_K regions.

1: **for** $k = 1$ to K **do**
2: **if** $k = 1$ **then**
3: Build graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ from $(\mathcal{I}, \mathbf{I})$ and $\mathcal{A}$
4: **else**
5: Build RAG $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ from $(\mathcal{I}, \mathbf{I})$ and L
6: $L \leftarrow ISF(G, f, T, n_k)$
7: **return** L

Figure 1. RISF algorithm

Input: Graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$, path-cost function f, number of iterations T, and number of regions n
Output: Labeled image L with n regions.

1: Create maps C, P and R from G
2: $\mathcal{S} \leftarrow \emptyset$, $\mathcal{S}' \leftarrow SeedSamplingStrategy(G, n)$
3: $\mathcal{K} \leftarrow \emptyset$
4: **for all** $s \in \mathcal{I}$ **do**
5: $C(s) \leftarrow +\infty$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$
6: **for** $iter = 1$ to T **do**
7: $(C, P, R, \mathcal{K}) \leftarrow DIFT(G, f, C, P, R, \mathcal{K}, \mathcal{S}, \mathcal{S}')$
8: $\mathcal{S} \leftarrow \mathcal{S}'$
9: Recompute $\mathcal{S}'$ with Equations 3 and 4
10: Convert root map R to a labeled image L
11: **return** L

Figure 2. ISF algorithm

The RISF algorithm is presented in Fig. 1. Each iteration of the loop in Lines 1-7 builds a new segmentation scale using the ISF algorithm. In Lines 2 and 3 a pixel (voxel) image graph is built for the first iteration following the standard ISF. In Lines 4 and 5 the segmentation result from the previous iteration is used to build a RAG according to RISF. In Line 6 the label map L is generated with ISF, representing the current segmentation scale.

The ISF algorithm is presented in Fig. 2. Lines 1-5 initialize the DIFT, ensuring that the first iteration runs the standard IFT algorithm. In Lines 6-9 the loop of computing a new spanning forest with DIFT and performing the seed recomputation is performed. In Line 10 the final segmentation label map is obtained, noting that each spanning tree can be represented and labelled according to its root.

In Fig. 3 the actual the DIFT algorithm is covered, which efficiently calls the IFT described in Section II over multiple iterations. Lines 1 initializes the priority queue $\mathcal{Q}$. Lines 2 and 3 isolate the set of removed and new seeds as S^- and S^+, respectively. The loop in Lines 4 and 5 resets all spanning trees rooted in S^-, while the one in Lines 6-8 initializes the new seeds in S^+ that will form new spanning trees. Lines 9-21 perform the actual computation of the new spanning forest.

Input: Graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$, path-cost function f, cost map C, predecessor map P, root map R, set $\mathcal{K}$, previous seed set $\mathcal{S}$, updated seed set $\mathcal{S}'$
Output: Updated maps C, P, R, and set $\mathcal{K}$

1: $\mathcal{Q} \leftarrow \emptyset$
2: $\mathcal{S}^+ \leftarrow \mathcal{S}' \setminus \mathcal{S}$
3: $\mathcal{S}^- \leftarrow \mathcal{S} \setminus \mathcal{S}'$
4: **for all** $s \in \mathcal{S}^-$ **do**
5: $\quad RemoveSubTree(G, C, P, R, \mathcal{Q}, \mathcal{K}, s)$
6: **for all** $s \in \mathcal{S}^+$ **do**
7: $\quad C(s) \leftarrow 0$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$
8: $\quad$ Insert s into $\mathcal{Q}$
9: **while** $\mathcal{Q}$ is not empty **do**
10: $\quad$ Remove s from $\mathcal{Q}$ such that $C(s)$ is minimum
11: $\quad \mathcal{K} \leftarrow \mathcal{K} \cup \{s\}$
12: $\quad$ **for** $t \in \mathcal{I}$ such that $(s,t) \in \mathcal{A}$ **and** $t \notin \mathcal{K}$ **do**
13: $\quad\quad tmp \leftarrow f(\pi_s^P \cdot \langle s,t \rangle)$
14: $\quad\quad$ **if** $tmp < C(t)$ **then**
15: $\quad\quad\quad C(t) \leftarrow tmp$, $P(t) \leftarrow s$, $R(t) \leftarrow R(s)$
16: $\quad\quad\quad$ **if** $t \notin \mathcal{Q}$ **then**
17: $\quad\quad\quad\quad$ Insert t into $\mathcal{Q}$
18: $\quad\quad$ **else**
19: $\quad\quad\quad$ **if** $P(t) = s$ **then**
20: $\quad\quad\quad\quad$ **if** $tmp > C(t)$ **or** $R(s) \neq R(t)$ **then**
21: $\quad\quad\quad\quad\quad RemoveSubTree(G, C, P, R, \mathcal{Q}, \mathcal{K}, t)$

Figure 3. DIFT algorithm

Line 10 removes the node with lowest cost from the priority queue, at which point we know it already has optimum cost for this DIFT iteration, marked by including it in set $\mathcal{K}$ (Line 11). Lines 12-17 perform the standard IFT behavior of extending optimum paths to the node's neighbors when a better cost is possible. Lines 18-21 perform the test to detect and remove inconsistent sub-trees [19]. We do not differentiate the special case of $tmp = C(t) \wedge R(s) \neq R(t)$, as proposed in [19], since it rarely happens in practice with our defined path-cost functions, being an unnecessary optimization.

In Fig. 4 the auxiliary procedure `RemoveSubTree` is shown, responsible for removing either inconsistent sub-trees [19] or sub-trees rooted at eliminated seeds from the previous DIFT execution. Line 1 initializes the auxiliary sets H and H' which will hold the nodes from the removed sub-tree and the immediate neighboring nodes (frontier set), respectively. The loop in Lines 2-11 covers all the nodes from the sub-tree, with Lines 3-6 being responsible for re-initializing them as never-visited nodes. The loop in Lines 7-11 updates H and H' with the node's neighbors, similar to a breadth-first search. Lines 12-15 reinsert finished nodes from the frontier set into the priority queue $\mathcal{Q}$ so they can revisit nodes from the removed sub-trees.

In Fig. 5 the proposed geodesic sampling procedure is defined. It follows an operation analogous to the ISF algorithm but, instead of a fixed seed size that is updated over a fixed number iterations, it builds an increased seed set by selecting

Input: Graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$, cost map C, predecessor map P, root map R, priority queue $\mathcal{Q}$, set $\mathcal{K}$ and node u (root of the sub-tree being removed).
Output: Updated maps C, P, R, set $\mathcal{K}$, and queue $\mathcal{Q}$

1: $H \leftarrow \{u\}$, $H' \leftarrow \emptyset$
2: **while** $H \neq \emptyset$ **do**
3: $\quad$ Remove s from H and from $\mathcal{K}$
4: $\quad C(s) \leftarrow +\infty$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$
5: $\quad$ **if** $s \in \mathcal{Q}$ **then**
6: $\quad\quad$ Remove s from $\mathcal{Q}$
7: $\quad$ **for** $t \in \mathcal{I}$ such that $(s,t) \in \mathcal{A}$ **and** $t \in \mathcal{K}$ **do**
8: $\quad\quad$ **if** $P(t) = s$ **then**
9: $\quad\quad\quad H = H \cup \{t\}$
10: $\quad\quad$ **else**
11: $\quad\quad\quad H' = H' \cup \{t\}$
12: **while** $H' \neq \emptyset$ **do**
13: $\quad$ Remove s from H'
14: $\quad$ **if** $C(s) \neq +\infty$ **and** $s \notin \mathcal{Q}$ **then**
15: $\quad\quad$ Insert s into $\mathcal{Q}$

Figure 4. RemoveSubTree algorithm

Input: Graph $G = (\mathcal{I}, \mathcal{A}, \mathbf{I})$ and number of seeds n.
Output: Seed set $\mathcal{S}$.

1: $\mathcal{S} \leftarrow \emptyset$, $\mathcal{S}' \leftarrow$ one arbitrary element from $\mathcal{I}$.
2: $\mathcal{K} \leftarrow \emptyset$
3: **for all** $s \in \mathcal{I}$ **do**
4: $\quad C(s) \leftarrow +\infty$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$
5: **while** $|\mathcal{S}'| < n$ **do**
6: $\quad (C, P, R, \mathcal{K}) \leftarrow DIFT(G, f_g, C, P, R, \mathcal{K}, \mathcal{S}, \mathcal{S}')$
7: $\quad$ Remove s from C such that $C(s)$ is maximum
8: $\quad \mathcal{S} \leftarrow \mathcal{S}'$
9: $\quad \mathcal{S}' \leftarrow \mathcal{S}' \cup \{s\}$
10: **return** $\mathcal{S}'$

Figure 5. Geodesic seed sampling algorithm

the node with maximum path cost until the desired number of seeds is reached.

VI. Experimental results

We evaluate RISF for natural and medical images by using two seed sampling strategies, including the geodesic one, in comparison with the best ISF-based methods in [15]. The instances of ISF and RISF, image datasets, and results are presented next.

A. Instances of ISF and RISF

In [15], the authors define several ISF-based methods and the best ones use either grid (GRID) or mixed (MIX) seed sampling, depending on the dataset, with two possible definitions for $\mathbf{I}(R(s))$ in Equation 2: either it is the color of the root $R(s)$ (ROOT) or it is the mean color of the region of the previous iteration that is now represented by $R(s)$ (MEAN). The mixed sampling is a combination between grid sampling and an entropy-based approach that generates more seeds

in heterogeneous regions of the image. The combination of those options results into four ISF-based methods: ISF-GRID-ROOT, ISF-MIX-ROOT, ISF-MIX-MEAN, and ISF-GRID-MEAN, the last one being previously presented as IFT-SLIC in [14]. We used the source code provided by the authors. [1]

In RISF, grid sampling selects the closest regions to the pixels of the grid as seeds, and geodesic sampling is presented by the algorithm in Fig. 5. This results two RISF-based methods: RISF-GRID and RISF-GEODESIC. Both start with ISF-GRID-MEAN for $n_1 = 10000$ superpixels in 2D and $n_1 = 25000$ supervoxels in 3D, and then reduce the number n_2 of regions to the desired one shown in the resulting graphs. The parameter α in Equation 2 is set to generate more compact superpixels in this use of ISF-GRID-MEAN to exploit the performance benefits of constraining the amount of updates during the DIFT computation.

B. Datasets

For natural images, we consider two datasets of 2D images: the Berkeley dataset (BSD300) [22] and the Birds dataset [23]. BSD300 is a standard superpixel segmentation benchmark which consists of 300 images with a wide variety of outdoor scenes. Each image has resolution 321×481 and has multiple hand-labeled segmentation ground truths indicating the borders perceived by multiple subjects. The Birds dataset has 50 public domain images of birds which are challenging to segment due to their thin and elongated parts. Each image has resolution up to 640×640 and a manually generated ground truth segmentation of a single bird in the scene.

As for medical images, the datasets have been provided by the authors in [15]: the 2D Liver dataset and the 3D Brain dataset. The Liver dataset is composed of 40 computed tomography (CT) slice images from the abdomen of four patients. Each image has resolution 512×512 and a manually generated ground truth segmentation of the liver in each slice. The Brain dataset comprises of 19 T1-weighted magnetic resonance (MRT1) brain images from different control patients. Each volumetric image is isotropic and has an order of ten million voxels, which poses a challenge in terms of processing time. The segmentation ground truths were obtained by a hybrid system of automatic model-based segmentation followed by interactive correction, separating the brain's cerebellum and each hemisphere as different objects.

C. Results

Each method is evaluated in terms of boundary recall (as implemented in [8]), undersegmentation error (as implemented in [9]), and execution time. Figs. 6, 7, 8 and 9 show those evaluation metrics on all datasets based on the target number of regions. In all datasets, RISF-GEODESIC displays better boundary adherence than the ISF-based methods, while RISF-GRID shows comparable performance to its non-recursive counterpart ISF-GRID-MEAN.

The RISF-based methods also show an improvement in terms of efficiency. Despite the theoretical complexity of all the ISF and RISF methods being $O(|\mathcal{I}| \log |\mathcal{I}|)$, those methods get faster in practice when the target number of regions increases due to a reduced number of tree updates during the successive DIFT iterations. Moreover, the majority of the execution time spent on RISF-based methods occurs when computing the initial segmentation scale with a higher number of regions using ISF-GRID-MEAN, resulting in the observed improvement. The behavior of ISF-GRID-MEAN being faster as we increase the number of regions is clear from its downward slope in all time plots. We also note that on the Brain dataset, the efficiency benefit is not as pronounced due to the computational overhead of storing and handling a big number of supervoxels[2].

Additionally, it is important to consider the overhead introduced by non-trivial seed sampling functions (e.g., geodesic sampling). The overhead introduced by the proposed geodesic sampling is observed by comparing the processing time of RISF-GEODESIC and RISF-GRID. Even with the speed improvement of using the DIFT-based implementation, we note that the cost of executing the geodesic seed sampling method over pixels would be prohibitive as compared to its use in RISF. This explains the absence of ISF with geodesic seed sampling in the experiments.

Besides the quantitative benefits of the RISF-based methods, we also highlight the fact it does so with the added benefit of providing a multi-scale segmentation that can be used by other algorithms.

Finally, the experiments were executed on an Intel Core i7-7700 3.6GHz processor with 8GB RAM.

VII. Conclusion

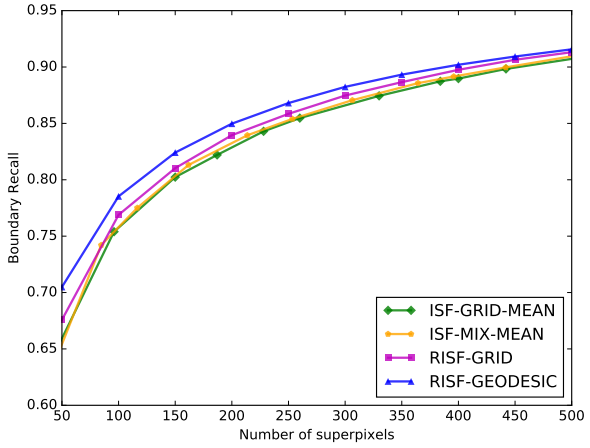
In this work, we introduce a new recursive version of the iterative spanning forest (ISF) [15] framework that defines the use of ISF methods over region adjacency graphs (RAGs) to generate a multi-scale superpixel segmentation. This version, called recursive iterative spanning forest (RISF), keeps all the benefits from the original framework while being able to exploit the usual reasons for operating over a RAG, namely, reduced computational cost and being able to work with higher level superpixel features. We also present a new improved sampling function based on the Image Foresting Transform (IFT) that is only viable in the context of RISF.

Our experiments show that a RISF-based method using the new sampling function can generate a superpixel segmentation faster and with better boundary adherence than existing ISF-based methods. The experiments also show that even a straightforward recursive version of an ISF method can be used to improve its efficiency while maintaining comparable segmentation results.

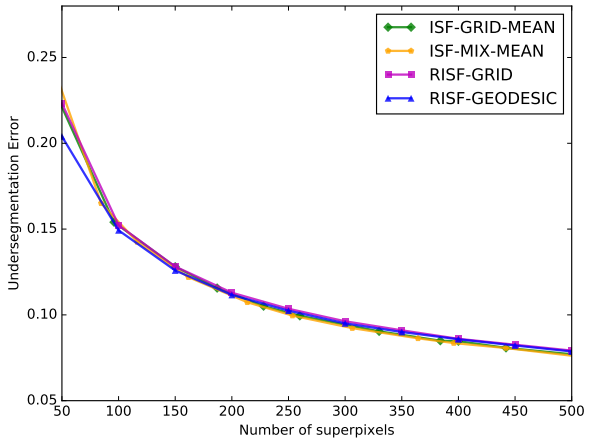
Future work includes exploring more ways to generate superpixel hierarchies with the RISF framework, such as using more scales and different superpixel features, and its use for higher level applications, in particular for active learning, image classification, and object segmentation.

[1] www.ic.unicamp.br/~afalcao/downloads.html

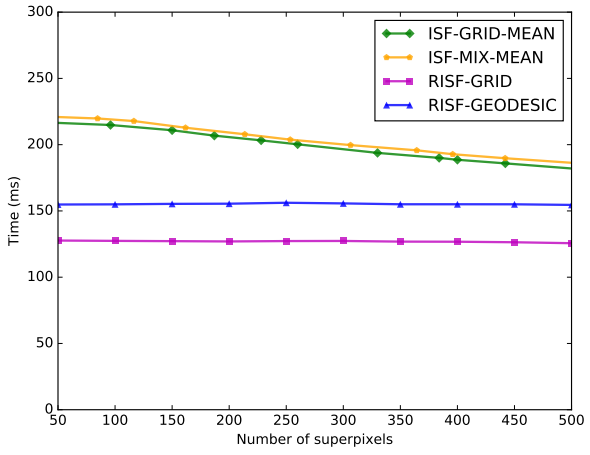
[2] Beyond $n_1 = 10000$ regions there is a negligible reduction in the processing time of ISF.



(a)

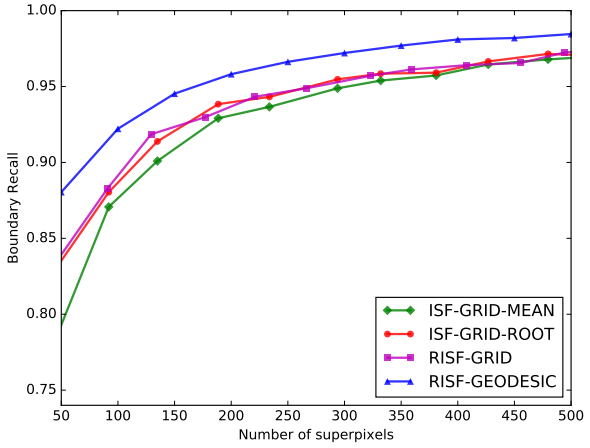


(b)

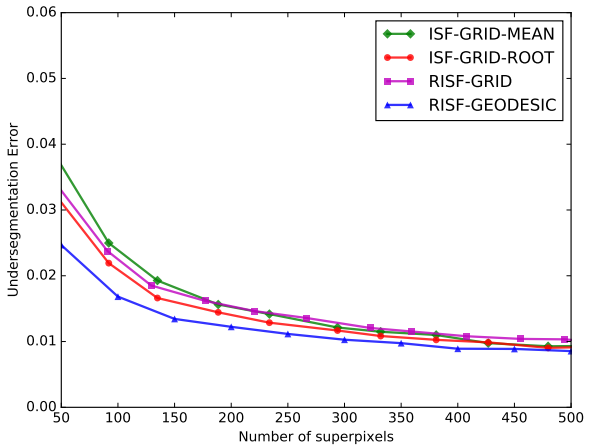


(c)

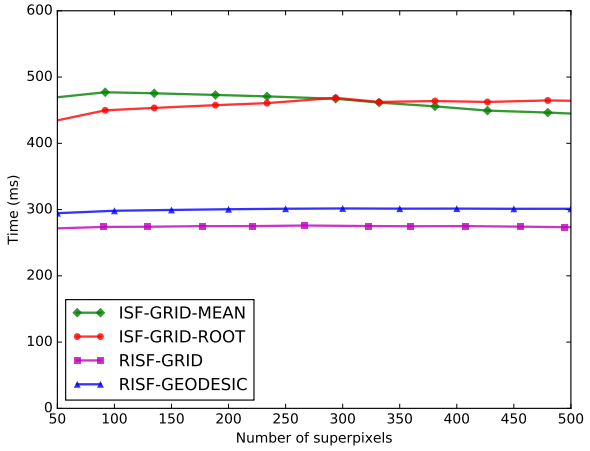
Figure 6. Comparison of boundary adherence and execution time for ISF-GRID-MEAN and the best ISF method against our proposed methods on the **Berkeley** dataset. Both RISF methods generate an initial segmentation scale of $n = 10000$ superpixels with ISF-GRID-MEAN ($\alpha = 0.12$), and then perform the final superpixel segmentation using the indicated sampling strategy and $\alpha = 0.5$. Both indicated ISF methods use $\alpha = 0.5$.



(a)



(b)

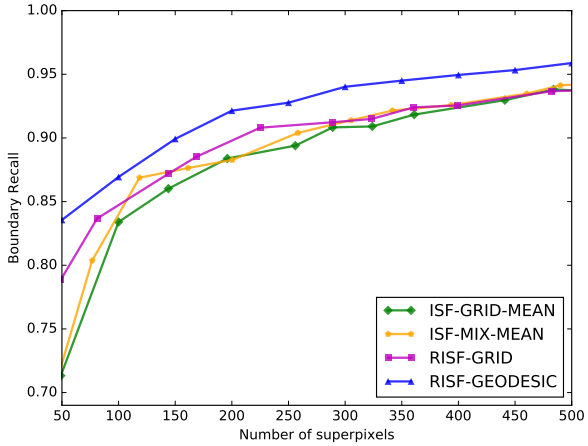


(c)

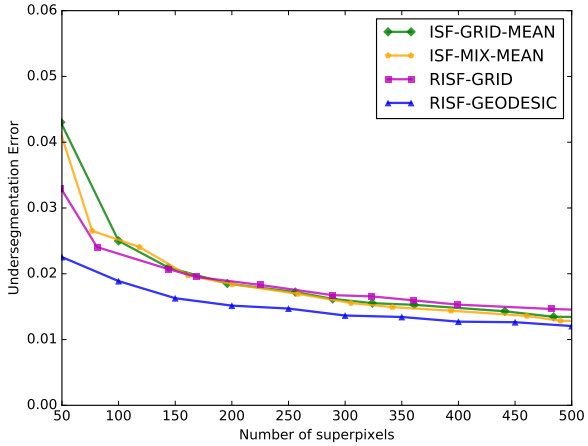
Figure 7. Comparison of boundary adherence and execution time for ISF-GRID-MEAN and the best ISF method against our proposed method on the **Birds** dataset. Both RISF methods generate an initial segmentation scale of $n = 10000$ superpixels with ISF-GRID-MEAN ($\alpha = 0.12$), and then perform the final superpixel segmentation using the indicated sampling strategy and $\alpha = 0.5$. Both indicated ISF methods use $\alpha = 0.5$.

VIII. Acknowledgements

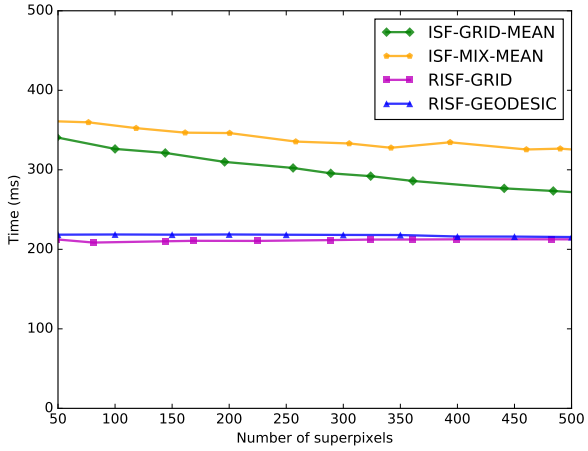
This work was supported by CAPES, and FAPESP (grant 2014/12236-1). We thank Dr. J.K. Udupa (MIPG-UPENN) for the Liver images and Dr. F. Cendes (FCM-UNICAMP) for the Brain images.



(a)

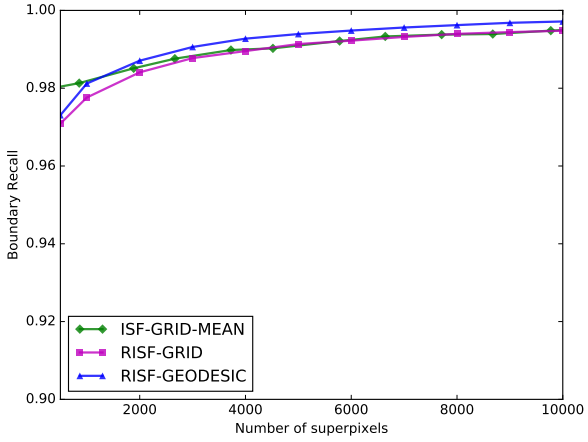


(b)

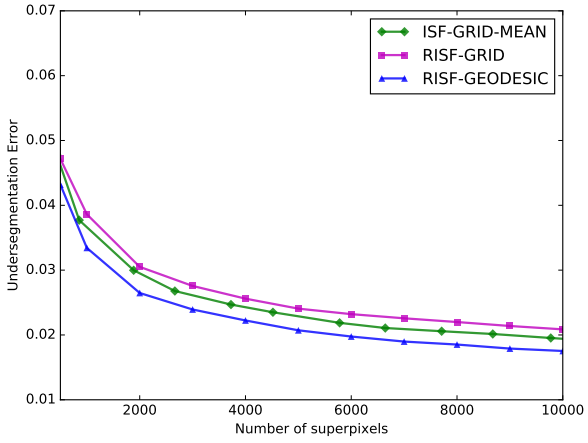


(c)

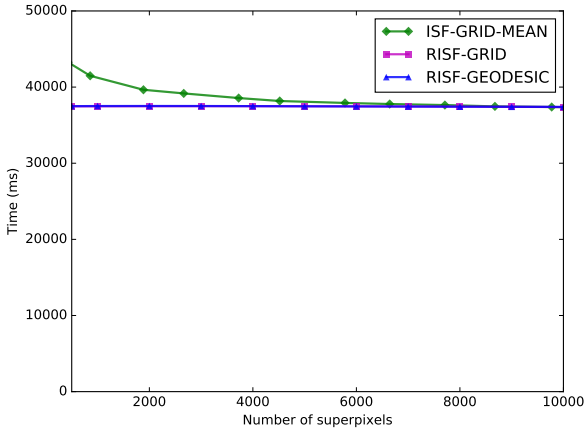
Figure 8. Comparison of boundary adherence and segmentation time for ISF-GRID-MEAN and the best ISF method against our proposed method on the **Liver** dataset. Both RISF methods generate an initial segmentation scale of $n = 10000$ superpixels with ISF-GRID-MEAN ($\alpha = 0.12$), and then perform the final superpixel segmentation using the indicated sampling strategy and $\alpha = 0.12$. ISF-GRID-MEAN and ISF-MIX-MEAN use $\alpha = 0.12$ and $\alpha = 0.5$, respectively.



(a)



(b)



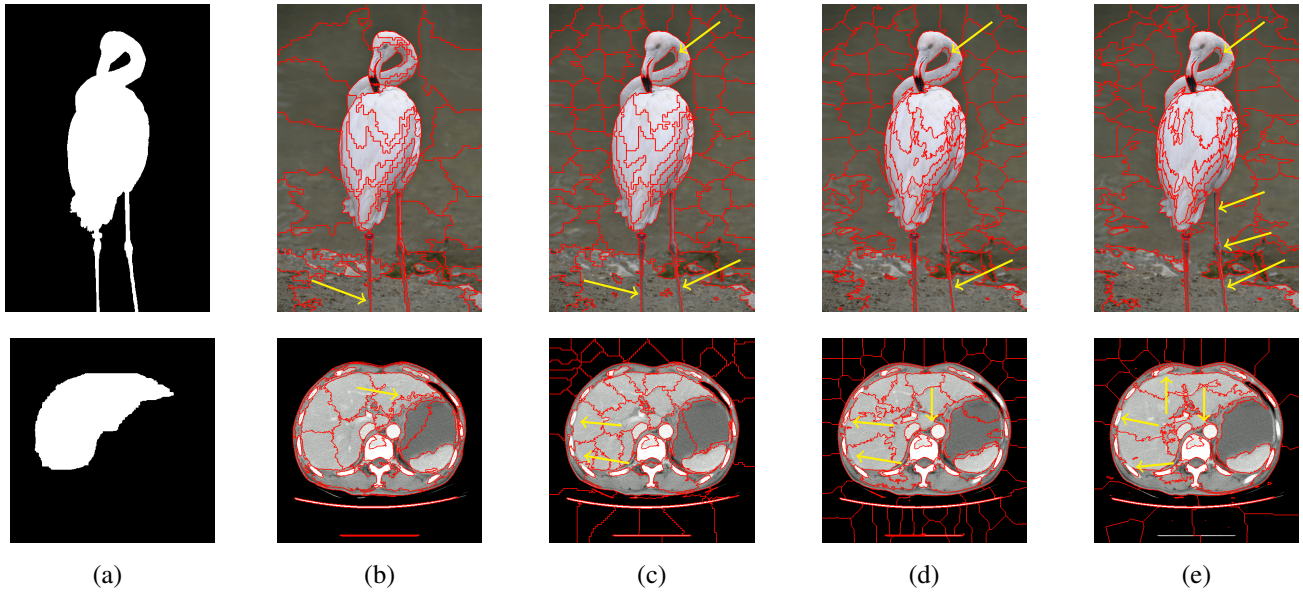
(c)

Figure 9. Comparison of boundary adherence and segmentation time for ISF-GRID-MEAN (which is the best ISF method) against our proposed method on the **Brain** dataset. Both RISF methods generate an initial segmentation scale of $n = 25000$ superpixels with ISF-GRID-MEAN ($\alpha = 0.01$), and then perform the final superpixel segmentation using the indicated sampling strategy and $\alpha = 0.01$. ISF-GRID-MEAN uses $\alpha = 0.01$.

REFERENCES

[1] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Computer Vision and Image Understanding*, vol. 166, pp. 1 – 27, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314217300589

[2] W. Wu, A. Y. C. Chen, L. Zhao, and J. J. Corso, "Brain tumor detection



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

Figure 10. Segmentation results for $n = 100$ superpixels on images from **Birds** (first row) and **Liver** (second row). Ground truth (a) and evaluated methods: (b) RISF-GEODESIC, (c) RISF-GRID, (d) ISF-GRID-MEAN and (e) best ISF method. The compactness parameter α is set according to Section VI-A. Yellow arrows indicate regions overlapping both object and background.

and segmentation in a CRF (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 2, pp. 241–253, 2014.

[3] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE International Conf. on Computer Vision (ICCV)*, Sept 2009, pp. 670–677.

[4] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes, "Layered object models for image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1731–1743, 2012.

[5] Z. Liu, X. Zhang, S. Luo, and O. L. Meur, "Superpixel-based spatiotemporal saliency detection," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 9, pp. 1522–1540, 2014.

[6] F. Yang, H. Lu, and M. Yang, "Robust superpixel tracking," *IEEE Trans. Image Processing*, vol. 23, no. 4, pp. 1639–1651, 2014.

[7] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 49–65, 2007.

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[9] P. Neubert and P. Protzel, "Superpixel benchmark and comparison," in *Proc. Forum Bildverarbeitung*, 2012, pp. 1–12.

[10] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3317–3330, 2017.

[11] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE International Conf. on Computer Vision and Pattern Recognition, (CVPR), Colorado Springs, CO, USA*, 2011, pp. 2097–2104.

[12] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations," *IEEE Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1241–1247, 2012.

[13] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. on Image Processing*, vol. 25, no. 12, pp. 5933–5942, Dec 2016.

[14] E. B. Alexandre, A. S. Chowdhury, A. X. Falcão, and P. A. V. Miranda, "IFT-SLIC: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform," in *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*. IEEE, 2015, pp. 337–344.

[15] J. E. Vargas-Muñoz, A. S. Chowdhury, E. B. Alexandre, F. L. Galvão, P. A. V. Miranda, and A. X. Falcão, "An iterative spanning forest framework for superpixel segmentation," *arXiv preprint arXiv:1801.10041*, 2018.

[16] A. X. Falcão, J. Stolfi, and R. de Alencar Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[17] K. C. Ciesielski, A. X. Falcão, and P. A. V. Miranda, "Path-value functions for which dijkstra's algorithm returns optimal mapping," *Journal of Mathematical Imaging and Vision*, Feb 2018. [Online]. Available: https://doi.org/10.1007/s10851-018-0793-1

[18] L. A. Mansilla, P. A. V. Miranda, and F. A. Cappabianco, "Image segmentation by image foresting transform with non-smooth connectivity functions," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. IEEE, 2013, pp. 147–154.

[19] M. A. T. Condori, F. A. M. Cappabianco, A. X. Falcão, and P. A. V. De Miranda, "Extending the differential image foresting transform to root-based path-cost functions with application to superpixel segmentation," in *Graphics, Patterns and Images (SIBGRAPI), 2017 30th SIBGRAPI Conference on*. IEEE, 2017, pp. 7–14.

[20] P. Buyssens, I. Gardin, S. Ruan, and A. Elmoataz, "Eikonal-based region growing for efficient clustering," *Image and Vision Computing*, vol. 32, no. 12, pp. 1045–1054, 2014.

[21] P. Buyssens, M. Toutain, A. Elmoataz, and O. Lézoray, "Eikonal-based vertices growing and iterative seeding for efficient graph-based segmentation," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4368–4372.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[23] L. A. C. Mansilla and P. A. V. Miranda, "Oriented image foresting transform segmentation: Connectivity constraints with adjustable width," in *29th SIBGRAPI: Conf. on Graphics, Patterns and Images*. IEEE, 2016, pp. 289–296.