An Iterative Spanning Forest Framework for Superpixel Segmentation

John E. Vargas-Muñoz, Ananda S. Chowdhury, *Senior Member, IEEE*, Eduardo B. Alexandre, Felipe L. Galvão, Paulo A. Vechiatto Miranda, and Alexandre X. Falcão, *Member, IEEE*

***Abstract*—Superpixel segmentation has emerged as an important research problem in the areas of image processing and computer vision. In this paper, we propose a framework, namely *Iterative Spanning Forest* (ISF), in which improved sets of connected superpixels (supervoxels in 3D) can be generated by a sequence of Image Foresting Transforms. In this framework, one can choose the most suitable combination of ISF components for a given application — i.e., i) a seed sampling strategy, ii) a connectivity function, iii) an adjacency relation, and iv) a seed pixel recomputation procedure. The superpixels in ISF structurally correspond to spanning trees rooted at those seeds. We present five ISF-based methods to illustrate different choices for those components. These methods are compared with a number of state-of-the-art approaches with respect to effectiveness and efficiency. Experiments are carried out on several datasets containing 2D and 3D objects with distinct texture and shape properties, including a high-level application, named *sky image segmentation*. The theoretical properties of ISF are demonstrated in the supplementary material and the results show ISF-based methods rank consistently among the best for all datasets.**

***Index Terms*—Image Foresting transform, spanning forests, mixed seed sampling, connectivity function, superpixel/supervoxel segmentation.**

I. INTRODUCTION

SUPERPIXEL generation has evolved as an important research topic in image processing and computer vision with a variety of target applications like medical image segmentation [1], sky image segmentation [2], motion segmentation [3], multi-class object segmentation [4], [5], object detection [6], spatiotemporal saliency detection [7], target tracking [8], and depth estimation [9]. A superpixel may be conceived as a region of similar and connected pixels, which makes the superpixel-based image representation computationally much more efficient than its pixel-based counterpart. For the sake of effectiveness and efficiency, it is also expected that objects of interest for a given application can be defined by the union of a reduced number of superpixels. This goal poses the main challenge in superpixel segmentation because those objects may exhibit different shape and texture properties (e.g., compact, 2D/3D, elongated, thin, color/gray-scale and so on). Given such variability, a flexible framework where one can design the most suitable superpixel segmentation method for a given application is necessary. Indeed, most of the successful methods [10], [11] start from (1) a seed sampling strategy, followed by multiple executions of (2) a superpixel delineation algorithm, and (3) a seed recomputation procedure for superpixel segmentation. However, they cannot usually be considered as a framework, in which the above components can be changed to improve effectiveness for distinct applications [12].

In view of the above observation, the main contribution of this paper is a superpixel segmentation framework, named *Iterative Spanning Forest* (ISF), in which one can choose the most suitable combination of independent components for a given application — i.e., (i) a seed sampling strategy, (ii) a connectivity function, (iii) an adjacency relation, and (iv) a seed recomputation procedure. For instance, a compromise between boundary adherence and shape regularity (i.e., superpixels with compact shapes) can be achieved by choice of the connectivity function. The ISF algorithm relies on a sequence of Image Foresting Transforms (IFTs) [13] from distinct seed sets, yielding to connected superpixel sets along the execution paths until convergence is achieved (see the supplementary material).

The ISF-based methods have the property of generating connected superpixels — i.e., each superpixel is a spanning tree rooted at a seed pixel. Popular clustering-based methods [14], [10], [15], [11] for superpixel segmentation do not necessarily have this important property, which in turn can negatively affect the object representation as a union of a set of superpixels. Figure 1, for instance, presents simple objects which are close to each other having similar colors and elongated shapes. We compare three methods in Figure 1, namely, an ISF-based method, SLIC (Simple Linear Iterative Clustering) [10], and LSC (Linear Spectral Clustering) [11]. The results indicate that even this simple example can be quite challenging for SLIC and LSC whereas the ISF-based method achieves good segmentation.

In order to illustrate the ISF framework, we present (i) a mixed seed sampling strategy based on normalized Shannon entropy, the standard grid sampling, and a regional-minima-based sampling; (ii) three connectivity functions; (iii) two adjacency relations, 4-neighborhood in 2D and 6-neighborhood in 3D; and (iv) two seed recomputation procedures. The mixed sampling strategy aims at estimating a higher number of seeds in more heterogeneous regions in order to improve boundary

John E. Vargas Muñoz, Felipe L. Galvão and Alexandre X. Falcão are with the Dept. of Information Systems, Institute of Computing, University of Campinas, Campinas, Brazil. E-mail: afalcao@ic.unicamp.br (corresponding author)

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

Eduardo B. Alexandre and Paulo A Vechiatto Miranda are with the Dept. of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil.

This paper has supplementary material with theorectical results available at http://ieeexplore.ieee.org. For further inquires contact afalcao@ic.unicamp.br.

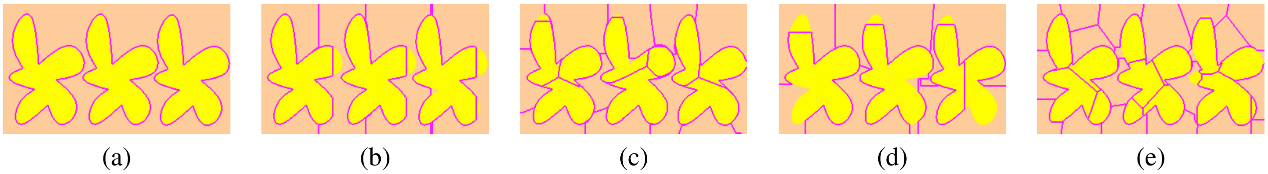


Fig. 1. (a) ISF can segment the objects with only 4 superpixels. LSC and SLIC have difficulties when objects with similar colors are close to each other and/or present elongated parts. (b-c) It shows that LSC cannot segment the three objects correctly with 7 and even 29 superpixels. (d-e) It shows that SLIC cannot segment the three objects correctly with 8 and even 30 superpixels.

adherence. Grid sampling tends to produce more regularly distributed superpixels and the regional-minima-based strategy aims at solving superpixel segmentation in a single IFT iteration. Two connectivity functions allow to control the balance between boundary adherence and superpixel regularity, and the third one maximizes boundary adherence regardless of superpixel regularity. Both adjacency relations guarantee the connectivity between pixels and their corresponding seeds (i.e., a result consistent with the superpixel definition). For seed recomputation, we present procedures that exploit color and spatial information, and color information only. At each execution, the IFT algorithm propagates paths from each seed to pixels that are more closely connected to that seed than to any other, according to a given connectivity function. The resulting superpixels are spanning trees rooted at those seeds.

The Berkeley dataset [16] has been widely used to evaluate superpixel segmentation algorithms. However, it does not contain a reasonable number of objects with distinct shape and texture properties. A more suitable testing environment should contain a reasonable number of compact objects, objects with thin and elongated parts, 2D and 3D objects, and objects in color and gray-scale images. In this paper, we extensively compare ISF-based methods with five state-of-the-art superpixel segmentation methods [10], [11], [14], [15], [17] on six image datasets of natural and medical images. We also evaluate one ISF-based method in comparison with SLIC and LSC on a high-level application, namely *sky image segmentation* (a seventh dataset). Altogether these datasets contain 949 images, comprising an environment that satisfies the aforementioned distinct object properties. The results show ISF-based methods consistently rank among the best for all datasets, which clearly demonstrate the generalization power of the ISF framework.

The rest of the paper is organized as follows. In Section II, we discuss related works and emphasize the importance of this contribution by describing recently published extensions of the ISF framework. Section III presents the ISF framework with the general algorithm and five ISF-based methods, discusses implementation issues, and provides a link to its code. The experimental results are discussed in Section IV and the ISF theoretical properties are demonstrated in the supplementary material. Section V states conclusion and provides directions for future work.

II. Related Work

Most superpixel segmentation approaches adopt a clustering algorithm and/or a graph-based algorithm to address the problem in one or multiple iterations of seed estimation. Several of these methods cannot guarantee connected superpixels: SLIC (Simple Linear Iterative Clustering) [10], LSC (Linear Spectral Clustering) [11], Vcells (Edge-Weighted Centroidal Voronoi Tessellations) [18], LRW (Lazy Random Walks) [15], ERS (Entropy Rate Superpixels) [14], DBSCAN (Density-based spatial clustering of applications with noise) [19] and GMM (Superpixel Segmentation Using Gaussian Mixture Model) [20]. Connected superpixels in these methods are usually obtained by merging regions, as a post-processing step, which can reduce the number of desired superpixels.

Some representative graph-based algorithms include Normalized Cuts [21], an approach based on minimum spanning tree [22], a method using optimal path via graph cuts [23], an energy minimization framework, which can also yield supervoxels [24], the watershed transform from seeds [25], [26], [17], and approaches based on random walk [14], [15]. Normalized cuts can generate more compact and more regular superpixels. However, as shown in [10], its performance in boundary adherence is inferior with respect to other methods. The problem with the algorithm in [22] is exactly the opposite. The resulting superpixels can conform to object boundaries, but they are very irregular in size and shape. Similar results are observed in graph-based watershed algorithms [25], [26]. An exception is the waterpixel approach [17] that enforces compactness by using a modified gradient image. However, these algorithms try to solve the segmentation problem from a single seed set (e.g., seeds are selected from the regional minima of a gradient image). Due to the absence of seed recomputation and/or quality of the image gradient, they usually miss important object boundaries. The performance of the methods described in [23] and [27] depends on the pre-computed boundary maps, which is not guaranteed to be the best in all cases. Authors in [24] actually suggest two methods for generating compact and constant-intensity superpixels. In [14], the authors use the entropy rate of a random walk on a graph and a balancing term for superpixel segmentation. The method yields good segmentation results, but it involves a greedy strategy for optimization. In [15], the authors show that the lazy random walk produces better results, but the method has initialization and optimization steps, both requiring the computation of the commute time, which tends to adversely affect the total execution time.

ISF falls in the category of graph-based algorithms as a particular case of a more general framework [13] — the Image Foresting Transform (IFT). IFT is a framework for the design of image operators based on connectivity, such as distance and geodesic transforms, morphological reconstructions, multiscale skeletonization, image description, region- and boundary-based image segmentation methods [28], [25], [29], [30], [31], [32], [33], [34], [35], with extensions to clustering and classification [36], [37], [38], [39], [40]. As the authors discuss in [41], by choice of its connectivity function, the IFT algorithm can output a watershed transform from a set of seeds that corresponds to a graph cut in which the minimum gradient value in the cut is maximum. From [26], it is known that the watershed transform from seeds is equivalent to a cut in a minimum-spanning tree (MST). That is, the removal of the arc with maximum weight from the single path in the MST that connects each pair of seeds results in a minimum-spanning forest (i.e., a watershed cut). Such a graph cut tends to be better than the normalized cut in boundary adherence, but worse in superpixel regularity.

In the evolution of superpixel segmentation methods, Mean-Shift [42], Quick-Shift [43], turbopixels [44], SLIC [10], geometric flow [45], LSC [11], and DBSCAN [19] are all worth-mentioning. Mean-Shift method produces irregular and loose superpixels whereas the Quick-Shift algorithm does not allow an user to choose the number of superpixels. The turbopixel-based approaches can produce good superpixels, but are computationally complex. Çiğla and Atalan [46] used connected k-means algorithm with convexity constraints to achieve superpixel segmentation via speeded-up turbopixels. The method is still bit slow, and, as claimed by the authors, fails to provide good boundary recall for complex images. SLIC is by far the most commonly used superpixel method [10]. It uses a regular grid for seed sampling. Once chosen, the seeds are transferred to the lowest gradient position within a small neighborhood. Finally, a modified k-means algorithm is used to cluster the remaining pixels. This algorithm was shown to perform better than many other methods (e.g., [44], [21], [22], [24], [43]). However, the k-means algorithm searches for pixels within a $2S \times 2S$ window around each seed, where S is the grid interval. For a non-regular seed distribution, some pixels may not be reached by any seed. Indeed, this might happen from the second iteration on and this labeling inconsistency problem is only solved by post-processing. In [45], Wang *et al.* proposed a geometric-flow-based method of superpixel generation. The method has high computational complexity as it involves computation of the geodesic distance and several iterations. LSC [11] and DBSCAN [19] are among the most recent approaches. LSC models the segmentation problem using Normalized Cuts, but it applies an efficient approximate solution using a weighted k-means algorithm to generate superpixels. DBSCAN performs fast pixel grouping based on color similarity with geometric restrictions and then merges small clusters to ensure connected superpixels.

ISF allows the design of distinct methods for superpixel segmentation and the first example was presented in [47], before we conceive ISF as a framework. Since then new applications and extensions of the ISF framework has been proposed and published in conference papers. For instance, the authors in [35] address the object segmentation problem as follows. Whenever automatic object segmentation fails, the only alternative is interactive correction. The differential IFT algorithm [48] allows to correct 3D object segmentation in sublinear time (i.e., it can provide real-time response to the user's actions), but it requires an optimum-path forest as input. The contribution in [35] is then a new connectivity function for ISF, which attracts the boundaries of the supervoxels towards the boundary of a given 3D object segmentation mask, such that an optimum-path forest for interactive segmentation correction can be derived from those supervoxels. More recently, we propose in [49] new methods for seed sampling and seed recomputation, using a variant of the connectivity function presented in [35] to attract the superpixel boundaries to high-intensity transitions in a given object saliency map (visual attention map). The paper actually represents a new paradigm for superpixel segmentation, in which superpixels can be created based on image and object information rather than image properties only. In [50], we propose a hierarchical superpixel segmentation framework by applying ISF recursively on the subsequent superpixel graphs (region adjacency graphs). It allows the use of new seed sampling strategies and contextual superpixel properties that can lead to an improvement in effectiveness over the simplest ISF-based methods presented in the current work. Therefore, our aim here is to introduce the ISF framework and show that its simplest ISF-based methods can already perform consistently well for diverse datasets.

III. The ISF framework

An ISF-based method results from the choice of each component: initial seed selection, connectivity function, adjacency relation, and seed recomputation strategy. The ISF algorithm is a sequence of Image Foresting Transforms (IFTs) from improved seed pixel sets (Section III-A). For initial seed selection, we propose either grid or mixed entropy-based seed sampling (hereafter referred to as GRID and MIX, respectively) as effective strategies (Section III-B). Additionally, we evaluate a seed sampling strategy that moves the seeds obtained by grid sampling to the closest minima of a gradient image, attempting to solve the problem in a single iteration. Examples of connectivity functions and adjacency relations for 2D and 3D segmentations are presented in Sections III-C and III-D respectively. Two strategies for seed recomputation are described in Section III-E. The ISF algorithm is presented in Section III-G and its theoretical properties are demonstrated in the supplementary material. Section III-H discusses implementation issues and provides a link to the code.

A. Image Foresting Transform

An image can be interpreted as a graph $G = (\mathcal{I}, \mathcal{A})$, whose pixels in the image domain $\mathcal{I} \subset Z^n$ are the nodes and pixel pairs (s, t) that satisfy the *adjacency relation* $\mathcal{A} \subset \mathcal{I} \times \mathcal{I}$ are the arcs (e.g., 4-neighbors when $n = 2$). We use $t \in \mathcal{A}(s)$ and $(s, t) \in \mathcal{A}$ to indicate that t is adjacent to s.

For a given image graph $G = (\mathcal{I}, \mathcal{A})$, a path $\pi_t = \langle t_1, t_2, \ldots, t_n = t \rangle$ is a sequence of adjacent pixels with terminus t. A path is *trivial* when $\pi_t = \langle t \rangle$. A path $\pi_t = \pi_s \cdot \langle s, t \rangle$ indicates the extension of a path π_s by an arc (s, t). When we want to explicitly indicate the origin of a path, the notation $\pi_{s \rightsquigarrow t} = \langle t_1 = s, t_2, \ldots, t_n = t \rangle$ is used, where s stands for the origin and t for the destination node. A *predecessor map* is a function P that assigns to each pixel t in $\mathcal{I}$ either some other adjacent pixel in $\mathcal{I}$, or a distinctive marker nil not in $\mathcal{I}$ — in which case t is said to be a *root* of the map. A *spanning forest* (image segmentation) is a predecessor map which contains no cycles — i.e., one which takes every pixel to nil in a finite number of iterations. For any pixel $t \in \mathcal{I}$, a spanning forest P defines a path π_t^P recursively as $\langle t \rangle$ if $P(t) = nil$, and $\pi_s^P \cdot \langle s, t \rangle$ if $P(t) = s \neq nil$.

A *connectivity (path-cost) function* computes a value $f(\pi_t)$ for any path π_t, including trivial paths $\pi_t = \langle t \rangle$. A path π_t is *optimum* if $f(\pi_t) \leq f(\tau_t)$ for any other path τ_t in Π_G (the set of paths in G). By assigning to each pixel $t \in \mathcal{I}$ one optimum path with terminus t, we obtain an optimal mapping C, which is uniquely defined by $C(t) = \min_{\forall \pi_t \text{ in } \Pi_G} \{f(\pi_t)\}$. *Image Foresting Transform* (IFT) [13] takes an image graph $G = (\mathcal{I}, \mathcal{A})$, and a connectivity function f; and assigns one optimum path π_t to every pixel $t \in \mathcal{I}$ such that an *optimum-path forest* P is obtained — i.e., a spanning forest where all paths are optimum. However, f must satisfy certain conditions, as described in [51], otherwise, the paths may not be optimum.

In ISF, all seeds are forced to be the roots of the forest by choice of f, in order to obtain a desired number of superpixels. For any given seed set $\mathcal{S}$, each superpixel will be represented by its respective tree in the spanning forest P as computed by the IFT algorithm.

B. Seed Sampling Strategies

Any natural image contains a lot of heterogeneity. There are parts of the image that can have really small variations in intensity whereas other parts in the image can show significant variations. So, it is but natural to choose more seeds from a more non-uniform region of an image. However, having a grid structure for the seeds is also essential to conform to the regularity of the superpixels. The proposed mixed sampling strategy achieves both the goals. We use a two-level quad-tree representation of an input 2D image. The heterogeneity of each quadrant (Q) is captured using Normalized Shannon Entropy (NSE(Q)). This is given by

$$NSE(Q) = -\frac{\sum_{i=1}^{n} p_i log_2(p_i)}{log_2 n}. \quad (1)$$

Here n denotes the total number of intensity levels in the quadrant Q and p_i is the probability of occurrence of the intensity i in the quadrant Q. For color images, we deem the lightness component in the Lab color model as the intensity of a pixel. Normalizing the entropy ensures that the $NSE(Q) \in [0, 1]$. At the first level in the quad-tree, we compute the normalized Shannon entropies for each quadrant and also obtain the mean $\mu(NSE)$ and the standard deviation $\sigma(NSE)$ of the four values. If the value of entropy for any quadrant exceeds the mean by one standard deviation, i.e., if $|NSE(Q) - \mu(NSE)| > \sigma(NSE)$, then we further divide the region in the next level into four quadrants. We then compute the NSE values for the new quadrants at the second level. Once, the two-level quad-tree representation is complete, we assign the number of seeds to be selected from each region as proportional to their NSE values. Finally, the seeds from each region are picked based on the grid sampling strategy. So, we essentially perform local grid sampling for each leaf node in the two-level quad-tree. This procedure may improve boundary recall with respect to grid sampling, depending on the dataset. In addition to grid and mixed sampling strategies, we have also evaluated seed selection based on the reduction of the seed set generated by grid sampling to the set of the closest regional minima in a gradient image.

C. Connectivity Functions

We consider the computation of the IFT with two path-cost functions that only guarantee a spanning forest, f_1 (Equation 3) and f_2 (Equation 4), and a third one, f_3 (Equation 5), that guarantees an optimum-path forest. The spanning forest in f_1 and f_2 might not be optimum, because the path costs depend on path-root properties [51]. However, these functions can efficiently deal with the problem of intensity heterogeneity [29].

The seed sampling approach (e.g. grid or mixed) defines an initial seed set $\mathcal{S}$, such that for each seed pixel $s_j \in \mathcal{S}$ at coordinate (x_j, y_j), its color representation in the Lab color space is given $I(s_j) = [l_j \;\; a_j \;\; b_j]^T$. A path-cost function f is defined by a trivial-path cost initialization rule and an extended-path cost assignment rule. We present three instances of f, denoted as f_1, f_2 and f_3, with trivial-path initialization rule given by

$$f_*(\pi_t = \langle t \rangle) \quad = \quad \begin{cases} 0 & \text{if } t \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \quad (2)$$

They differ in the extended-path cost assignment rule, as follows.

$$f_1(\pi_{s_j \rightsquigarrow s} \cdot \langle s, t \rangle) = f_1(\pi_s) + (\|I(t) - I(s_j)\|\alpha)^\beta + \|s, t\|, \quad (3)$$

where $\alpha \geq 0$, $\beta \geq 1$, and $I(t) = [l_t \; a_t \; b_t]^T$ is the color vector at pixel t.

$$f_2(\pi_{s_j \rightsquigarrow s} \cdot \langle s, t \rangle) = f_2(\pi_s) + (\|I(t) - M(s_j)\|\alpha)^\beta + \|s, t\|, \quad (4)$$

where $M(s_j)$ is the mean color, computed inside the superpixel of the previous iteration, which contains the new seed s_j ($M(s_j) = I(s_j)$ at the first iteration).

$$f_3(\pi_{r \rightsquigarrow s} \cdot \langle s, t \rangle) = \max\{f_3(\pi_s), D(t)\}, \quad (5)$$

where $D(t)$ is the value of the gradient image in the pixel t.

At the end of the IFT algorithm, each superpixel will be represented by its respective tree in the spanning forest P. After that, an update step adjusts the roots (new seeds) of the spanning trees.

For paths $\pi_{t_1 \rightsquigarrow t_n} = \langle t_1, t_2, \ldots, t_n \rangle$, $n > 1$, and additive path-cost function $f(\pi_{t_1 \rightsquigarrow t_n}) = \sum_{i=1,2,\ldots,n-1}\{w(t_i, t_{i+1})\}$, $w(t_i, t_{i+1}) \geq 0$, the minimization of the cost map imposes too much shape regularity on superpixels, by avoiding adherence to image boundaries. On the other hand, $f(\pi_{t_1 \rightsquigarrow t_n}) = \max_{i=1,2,\ldots,n-1}\{w(t_i, t_{i+1})\}$ (Equation 5, for $w(t_i, t_{i+1}) = D(t_{i+1})$) provokes high adherence to image boundaries, but also possible leakings when delineating poorly defined parts of the boundaries. The path-cost function $f(\pi_{t_1 \rightsquigarrow t_n}) = \sum_{i=1,2,\ldots,n-1}\{w(t_i, t_{i+1})^\beta\}$, $\beta > 1$, represents a compromise

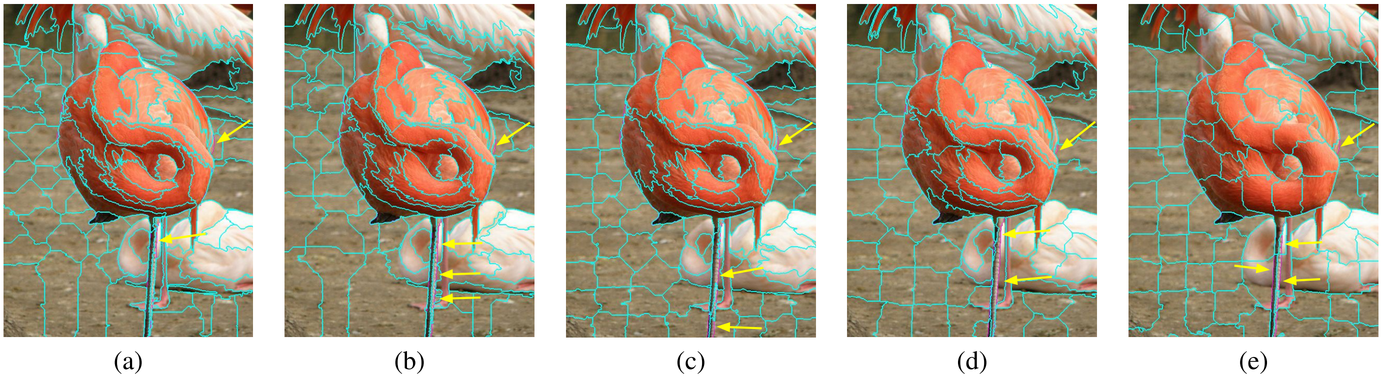


Fig. 2. Segmentation results of an image from Birds [52] for five ISF methods (a) ISF-GRID-ROOT (BR = 0.93, UE = 0.01), (b) ISF-MIX-ROOT (BR = 0.89, UE = 0.02), (c) ISF-GRID-MEAN (BR = 0.90, UE = 0.02), (d) ISF-MIX-MEAN (BR = 0.86, UE = 0.02), and (e) ISF-REGMIN (BR = 0.82, UE = 0.02). Yellow arrows indicate leaking between object and background.

between the previous two. We fix $\beta = 12$ in all experiments to approximate the effect of high adherence to image boundaries with considerably reduced leaking in superpixel segmentation. The arc weight $w(t_i, t_{i+1}) = \|I(t_{i+1}) - I(s_j)\|\alpha$ (Equation 3 for $s_j = t_1$), or $w(t_i, t_{i+1}) = \|I(t_{i+1}) - M(s_j)\|\alpha$ (Equation 4 for $s_j = t_1$), penalizes paths that cross image boundaries, but the choice of α provides the compromise between the shape regularity on superpixels, as imposed by the spatial connectivity component $\|t_{n-1}, t_n\|$ in Equations 3 and 4, and the high boundary adherence of $\sum_{i=1,2,\ldots,n-1}\{w(t_i, t_{i+1})^\beta\}$ for $\beta = 12$. The choice of α is then optimized as described in the experimental section.

D. Adjacency Relation

The popular choices for adjacency relation are 4- or 8-neighborhood in 2D and 6- or 26-neighborhood in 3D in order to ensure connected superpixels (supervoxels). We prefer simple symmetric adjacency of 4-neighborhood in 2D and 6-neighborhood in 3D. This choice helps in the regularity of the superpixels/supervoxels.

E. Seed Recomputation

We next discuss the automated seed recomputation strategy. Let s_i^t be the i^{th} superpixel root (seed) at iteration t and its feature vector defined as $[l_i^t\ a_i^t\ b_i^t\ x_i^t\ y_i^t]^T$. We select s_i^t either as the pixel of the superpixel whose color is the most similar to the mean color of the superpixel or as the pixel of the superpixel that is the closest to its geometric center. During the subsequent IFT computations, we only recompute the seed s_i^{t+1} if:

$$\|[l_i^t\ a_i^t\ b_i^t] - [l_i^{t+1}\ a_i^{t+1}\ b_i^{t+1}]\| > \sqrt{\mu_c} \tag{6}$$

or

$$\|[x_i^t\ y_i^t] - [x_i^{t+1}\ y_i^{t+1}]\| > \sqrt{\mu_s}, \tag{7}$$

where μ_c and μ_s are the average color and spatial distances to seed s_i^t.

F. Five Different ISF Methods

We present five ISF methods. The first two use function f_1, ISF-GRID-ROOT is based on grid sampling and ISF-MIX-ROOT is based on mixed sampling. They recompute seeds as the pixel inside each superpixel whose color is the closest to the mean color of the superpixel. The third and fourth methods use function f_2, ISF-GRID-MEAN is based on grid sampling and ISF-MIX-MEAN is based on mixed sampling. They recompute seeds as the pixel inside each superpixel whose position is the closest to the geometric center of the superpixel. The method presented in [47] is called here ISF-GRID-MEAN.

We now discuss the fifth superpixel generation method, called ISF-REGMIN, that uses path-cost function f_3. ISF-REGMIN is designed to be fast, as it uses only a single iteration of the IFT algorithm with no seed recomputation. This method initially performs grid sampling to set the seeds. Then, the seeds are substituted by any pixel at the closest regional minimum, computed in the gradient image.

It is important to note that the ISF methods do not require a post-processing step as the connectivity is already guaranteed by design.

Figure 2 presents the segmentation results of the five ISF methods on an image of Birds [52]: ISF-GRID-ROOT, ISF-MIX-ROOT, ISF-GRID-MEAN, ISF-MIX-MEAN and ISF-REGMIN. For this dataset, with thin and elongated object parts, ISF-GRID-ROOT obtains the best result.

G. The ISF Algorithm

Algorithm 1 presents the Iterative Spanning Forest procedure.

Algorithm 1. – ITERATIVE SPANNING FOREST

INPUT:	Image $\hat{I} = (\mathcal{I}, I)$, adjacency relation $\mathcal{A}$, initial seed set $\mathcal{S} \subset \mathcal{I}$, the parameters $\alpha \geq 0$ and $\beta \geq 1$, and the maximum number of iterations $MaxIters \geq 1$.
OUTPUT:	Superpixel label map L_s.
AUXILIARY:	State map S, priority queue Q, predecessor map P, cost map C, root map R and superpixel mean color array M.

1. $iter \leftarrow 0$

2. **While** $iter < MaxIter$, **do**
3. **For each** $t \in \mathcal{I}$, **do**
4. $P(t) \leftarrow nil$, $R(t) \leftarrow t$
5. $S(t) \leftarrow White$, $C(t) \leftarrow +\infty$
6. $label \leftarrow 1$
7. **For each** $t \in \mathcal{S}$, **do**
8. $C(t) \leftarrow 0$
9. $L_s(t) \leftarrow label$, $label \leftarrow label + 1$
10. *Insert t in Q*, $S(t) \leftarrow Gray$
11. **If** $iter = 0$, **then**
12. $M(t) \leftarrow I(t)$
13. **While** $Q \neq \emptyset$, **do**
14. *Remove s from Q such that* $C(s)$ *is minimum*
15. $S(s) \leftarrow Black$
16. **For each** $t \in \mathcal{A}(s)$, *such that* $S(t) \neq Black$, **do**
17. $c \leftarrow C(s) + (\|I(t) - M(R(s))\|\alpha)^{\beta} + \|s,t\|$
18. **If** $c < C(t)$, **then**
19. *Set* $P(t) \leftarrow s$, $R(t) \leftarrow R(s)$
20. *Set* $C(t) \leftarrow c$, $L_s(t) \leftarrow L_s(s)$
21. **If** $S(t) = Gray$, **then**
22. *Update position of t in Q*
23. **Else**
24. *Insert t in Q*
25. $S(t) \leftarrow Gray$
26. $\mathcal{S}, M \leftarrow RecomputeSeeds(\mathcal{S}, \hat{I}, L_s)$
27. $iter \leftarrow iter + 1$
28. *Return* L_s

Line 1 initializes the auxiliary variable $iter$ (iteration number). The loop in Line 2 stops when the maximum number of iterations is achieved. Lines 3-5 initialize the values for the predecessor, root, state and cost maps for all image pixels. The state map S indicates by $S(t) = White$ that a pixel t was never visited (never inserted in the priority queue Q), by $S(t) = Gray$ that t has been visited and is still in Q, and by $S(t) = Black$ that t has been processed (removed from Q). Lines 7-12 initialize the cost and label maps and insert the seeds in Q. The seeds are labeled with consecutive integer numbers in the superpixel label map L_s. Lines 13-25 perform the label propagation process. First, we remove the pixels s that have minimum path cost in Q. Then the loop in Lines 16-25 evaluates if a path with terminus s extended to its adjacent t is cheaper than the current path with terminus t and cost $C(t)$. If that is the case, s is assigned as the predecessor of t and the root of s is assigned to the root of t (Line 19). The path cost and the label of t are updated. If t is in Q, its position is updated, otherwise t is inserted into Q. After the label propagation stage, the function $RecomputeSeeds$ returns the new seed set and the new mean color values M for the superpixels. Note that in the first iteration the feature vector of the superpixel root is the seed pixel color (Line 11-12). The tasks of label propagation and seed recomputation are performed until the condition of Line 2 is achieved. The algorithm returns the label map L_s (superpixel segmentation). Note that the algorithm describes the method ISF-MIX-MEAN if we use mixed sampling as seed initialization strategy. It uses the path-cost function f_2 (see Equation 4) in Line 17. By replacing Line 17 with the path-cost function f_1 (see Equation 3), we obtain the algorithm for the method ISF-MIX-ROOT. Finally, by replacing mixed sampling by grid sampling in ISF-MIX-ROOT, we obtain the method ISF-GRID-ROOT.

H. Implementation issues and available code

In general, using a priority queue as a binary heap, each execution of the IFT algorithm takes time $O(N \log N)$ for $N = |\mathcal{I}|$ pixels (linearithmic time). Given that the time to recompute seeds is linear, the complexity of the ISF framework using a binary heap is linearithmic, independently of the number of superpixels. For integer path costs, such as in ISF-REGMIN, it is possible to reduce the IFT execution time to $O(N)$ using a priority queue based on bucket sorting [28].

For efficient implementation, we use a new variant, as proposed in [53], of the Differential Image Foresting Transform (DIFT) algorithm [48]. This algorithm is able to update the spanning forest by revisiting only pixels of the regions modified in a given iteration $iter > 1$. The efficient implementation of ISF is available at www.ic.unicamp.br/~afalcao/downloads.html.

IV. Experimental Results

In this section, we evaluate the methods based on their effectiveness on 2D and 3D image datasets according to Boundary Recall (BR), as implemented in [10], and Undersegmentation Error (UE), as implemented in [54], for a varying number of superpixels; effectiveness in terms of F-score (Dice) for 3D brain and 2D sky image segmentation; and their efficiency.

A. Datasets

For evaluation of the ISF-based methods, we use seven datasets with 2D and 3D objects in natural and medical images. Berkeley [16], Birds [52], Grabcut [55], Insects [52], Liver, and Sky are 2D datasets. Brain is a 3D dataset with MR-T1 images of the brain and three objects of interest — left and right brain hemispheres, and cerebellum, without pons and medulla, which poses a great challenge to superpixel segmentation due to the absence of boundary information in some parts. These images contain about 10 millions of voxels each. Birds, Insects, and Sky can be downloaded with their annotations from http://www.vision.ime.usp.br/~pmiranda/downloads.html. They contain publicly available images from *Pixabay* and Caltech. Liver and Brain are private. Berkeley and Grabcut are available from https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ and http://www.robots.ox.ac.uk/~vgg/data/iseg/, respectively. Table I describes their main characteristics.

B. Effectiveness on 2D datasets

The ISF-based methods are compared with five approaches from the state-of-the-art on 2D datasets: SLIC (Simple Linear Iterative Clustering) [10] [1], LSC (Linear Spectral Clustering) [11] [2], ERS (Entropy Rate Superpixel) [14], LRW (Lazy Random Walk) [15] [3], and Waterpixels [17]. In order to avoid busy and confusing plots, we present the effectiveness of the two best ISF-based methods (10 iterations), ISF-GRID-ROOT and ISF-MIX-MEAN, and the fastest one (ISF-REGMIN) for

[1] http://ivrl.epfl.ch/supplementary_material/RK_SLICSuperpixels/
[2] http://jschenthu.weebly.com/projects.html
[3] https://github.com/shenjianbing/lrw14/

TABLE I
DATASETS USED FOR EVALUATION OF THE ISF-BASED METHODS.

Dataset	Description	Number of images
Berkeley [16]	Color images with borders of 2D objects	500
Birds [52]	Color images of birds, 2D objects with thin and elongated parts	150
Grabcut [55]	Color images with mostly 2D compact objects	50
Insects [52]	Color images of insects, 2D objects with thin and elongated parts	130
Liver	CT gray-scale images of the liver, compact 2D objects in slice images of the abdomen	40
Brain	MR-T1 gray-scale images of the brain, 3D compact objects with absence of boundary information	19
Sky	Color images containing the sky, 2D objects with holes due to the presence of airplanes	60

each 2D dataset. Note that, ISF-GRID-MEAN [47] is not among them, but it was the best ISF-based approach for 3D brain image segmentation and we use it for the experiments with these datasets. The parameter α of the ISF-based methods has been adjusted based on 10 images from the Berkeley dataset and then kept fixed for all 2D datasets.

We maintain ISF-REGMIN in the plots because it (a) uses an integer path-cost function, which allows fast computation in time proportional to the number of pixels and independent of the number of seeds (superpixels), (b) does not require seed recomputation, and even being the simplest among the ISF-based methods, (c) it shows consistently better effectiveness than its counterpart, Waterpixels [17]. We also include a fast hybrid approach, namely SLIC-ISF, that combines 10 iterations of SLIC for faster seed estimation, followed by 2 iterations of ISF, to show that it is competitive to the others. Figures 3–7 show the results of this first round of experiments, using $\alpha = 0.5$ and $\beta = 12$ for the ISF-based methods based on f_1 or f_2.

Although LSC presents the best performance (the highest BR and the lowest UE) on Berkeley, the same is not observed for the other datasets. On Insects, for instance, ERS presents the best performance followed by ISF-GRID-ROOT. ISF-based methods are the best for Birds and Liver, being competitive for Grabcut and among the second best for Berkeley and Insects. ISF-REGMIN is consistently better than Waterpixels in both BR and UE for all datasets. ERS performs well in Berkeley and Insects, but its performance is not competitive in the other three datasets. Although SLIC is the fastest and most used method, its performance is far from being competitive in all datasets. Among the baselines, LSC is the most competitive. However, its performance in UE can be negatively affected for objects with thin and elongated parts, such as Birds and Insects. Except for Berkeley and Insects, SLIC-ISF presents better performance than ERS in BR and UE.

In conclusion, one cannot say that there is a winner for all datasets, because the objects may be very different from image to image. However, the results clearly show that ISF can produce methods with consistently high effectiveness in different datasets. This important aspect of robustness is not observed in the others. This also shows the importance of obtaining connected superpixels with no need for post-processing. The performance of LSC in UE is usually inferior when compared to its performance in BR. Birds and Insects are clearly a case in the point. Indeed, LSC produces less regular superpixels with high BR. For sky image segmentation, as we will see, this property of LSC considerably impairs its effectiveness. Between ISF-GRID-ROOT and ISF-MIX-MEAN, we can say that ISF-MIX-MEAN provides better results in most datasets. We believe this is related to the advantages in the effectiveness of mix sampling over grid sampling.

Figure 8 then illustrates the quality of segmentation in images from the five datasets using the best ISF-based method and the most competitive baseline for each case. The examples show that ISF can produce considerably better results in images from all datasets.

C. Effectiveness on 3D images

Given that the 3D extension of ISF simply requires a different choice of adjacency relation, we present a comparison among the best ISF-based method for this application (ISF-GRID-MEAN with $\alpha = 0.1$), the only baseline with 3D implementation (SLIC), and the hybrid approach (SLIC-ISF) on the Brain dataset. The parameter α of ISF has been determined based on its F-score (Dice) in a single 3D image, the rest of the images being used for testing. Figure 9a shows the three objects of interest: left and right brain hemispheres, and cerebellum, without pons and medulla. Segmentation creates supervoxels as shown in Figure 9b. Supervoxels with more than 50% of their voxels inside a particular object are labeled as belonging to that object, otherwise they are considered as part of the background or other objects. Effectiveness is measured by the F-score of this decision for three supervoxel resolutions, given the usual image sizes: low ($N = 1000$), medium ($N = 5000$), and high ($N = 10000$). Table II shows the results of this experiment, using a 64 bit, Core(TM) i7-3770K Intel(R) PC with CPU speed of 3.50GHz. It is not a surprise that ISF outperforms SLIC in effectiveness. However, SLIC is exploiting parallel computing [4] and given that SLIC-ISF is twice faster than ISF, their equivalence in performance above medium superpixel resolution is an excellent result. Another interesting observation is that ISF performs better for a value of α ($\alpha = 0.1$) lower than 0.5 (i.e., more regular supervoxels).

[4]Without parallel computing, SLIC would take from 19s-23s of processing time for $N = 1000$ to $N = 10000$ supervoxels.

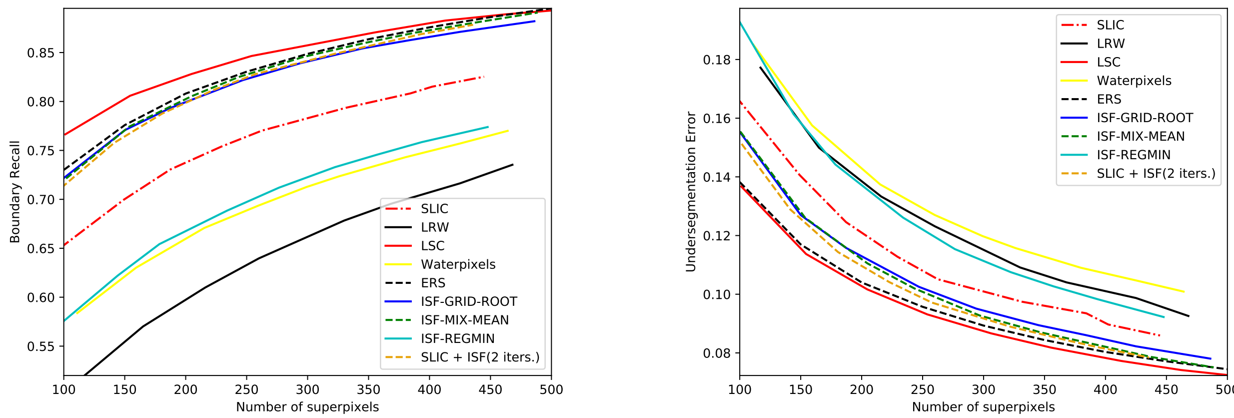


Fig. 3. Variations of BR, UE with number of superpixels for ISF-MIX-MEAN, ISF-GRID-ROOT, ISF-REGMIN, SLIC, the combination of SLIC and ISF (two iterarions), LRW, ERS, Waterpixels and LSC methods on **Berkeley**. We use the parameters $\alpha = 0.5$ for ISF variants, $m = 10$ (compactness parameter) for SLIC variants, $\alpha = 0.999999$ for LRW, $k = 8$ for Waterpixels and $ratio = 0.075$ for LSC.

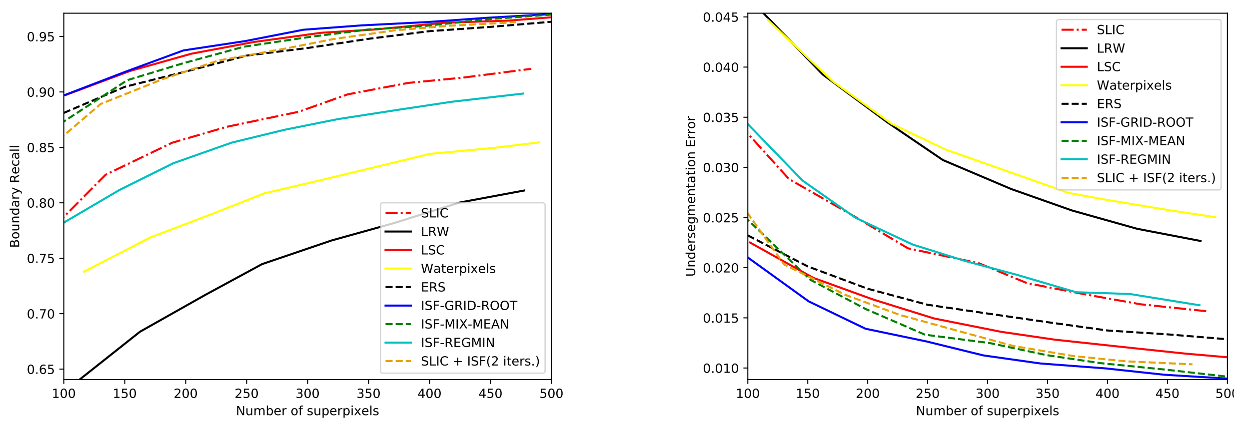


Fig. 4. Variations of BR, UE with number of superpixels for ISF-MIX-MEAN, ISF-GRID-ROOT, ISF-REGMIN, SLIC, the combination of SLIC and ISF (two iterarions), LRW, ERS, Waterpixels and LSC methods on **Birds**. We use the parameters $\alpha = 0.5$ for ISF variants, $m = 10$ (compactness parameter) for SLIC variants, $\alpha = 0.999999$ for LRW, $k = 8$ for Waterpixels and $ratio = 0.075$ for LSC.

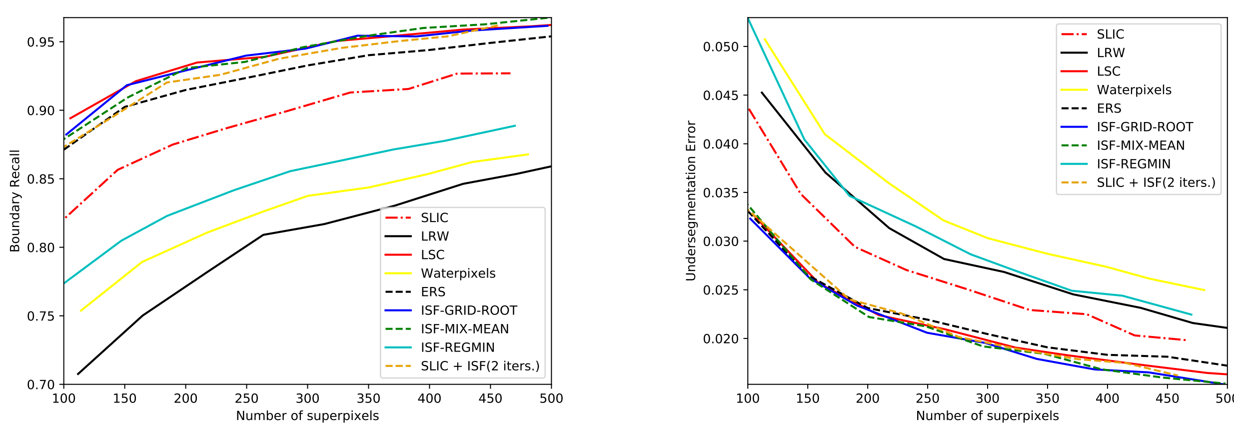


Fig. 5. Variations of BR, UE with number of superpixels for ISF-MIX-MEAN, ISF-GRID-ROOT, ISF-REGMIN, SLIC, the combination of SLIC and ISF (two iterarions), LRW, ERS, Waterpixels and LSC methods on **Grabcut**. We use the parameters $\alpha = 0.5$ for ISF variants, $m = 10$ (compactness parameter) for SLIC variants, $\alpha = 0.999999$ for LRW, $k = 8$ for Waterpixels and $ratio = 0.075$ for LSC.

Figures 9c-d show another example using ISF-GRID- MEAN, where the specification of 10 supervoxels using $\alpha =$

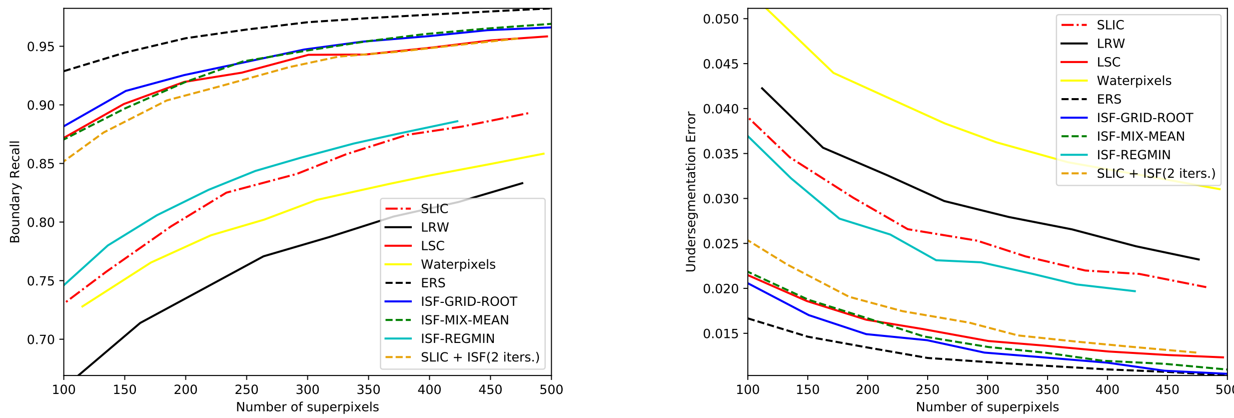


Fig. 6. Variations of BR, UE with number of superpixels for ISF-MIX-MEAN, ISF-GRID-ROOT, ISF-REGMIN, SLIC, the combination of SLIC and ISF (two iterarions), LRW, ERS, Waterpixels and LSC methods on **Insects**. We use the parameters $\alpha = 0.5$ for ISF variants, $m = 10$ (compactness parameter) for SLIC variants, $\alpha = 0.999999$ for LRW, $k = 8$ for Waterpixels and $ratio = 0.075$ for LSC.

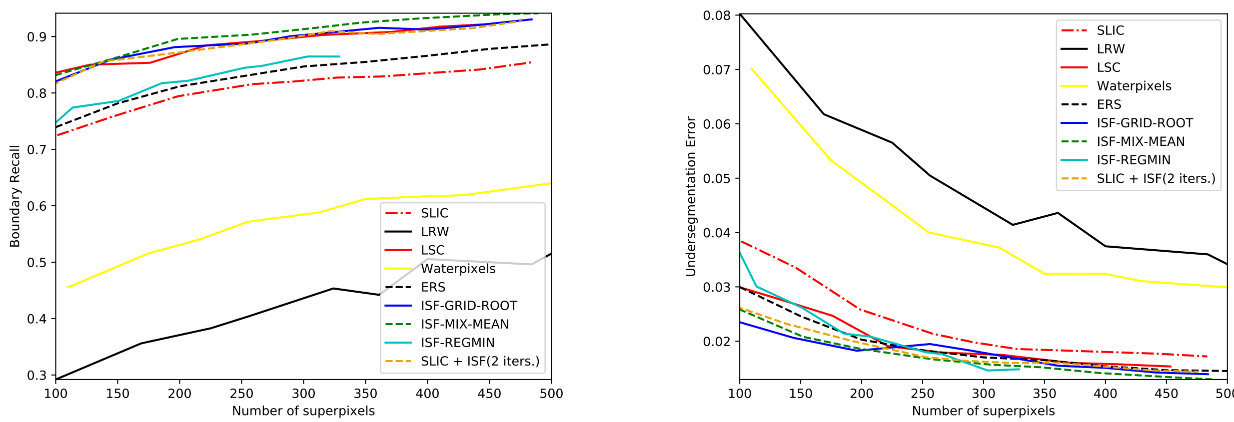


Fig. 7. Variations of BR, UE with number of superpixels for ISF-MIX-MEAN, ISF-GRID-ROOT, ISF-REGMIN, SLIC, the combination of SLIC and ISF (two iterarions), LRW, ERS, Waterpixels and LSC methods on **Liver**. We use the parameters $\alpha = 0.5$ for ISF variants, $m = 10$ (compactness parameter) for SLIC variants, $\alpha = 0.999999$ for LRW, $k = 8$ for Waterpixels and $ratio = 0.075$ for LSC.

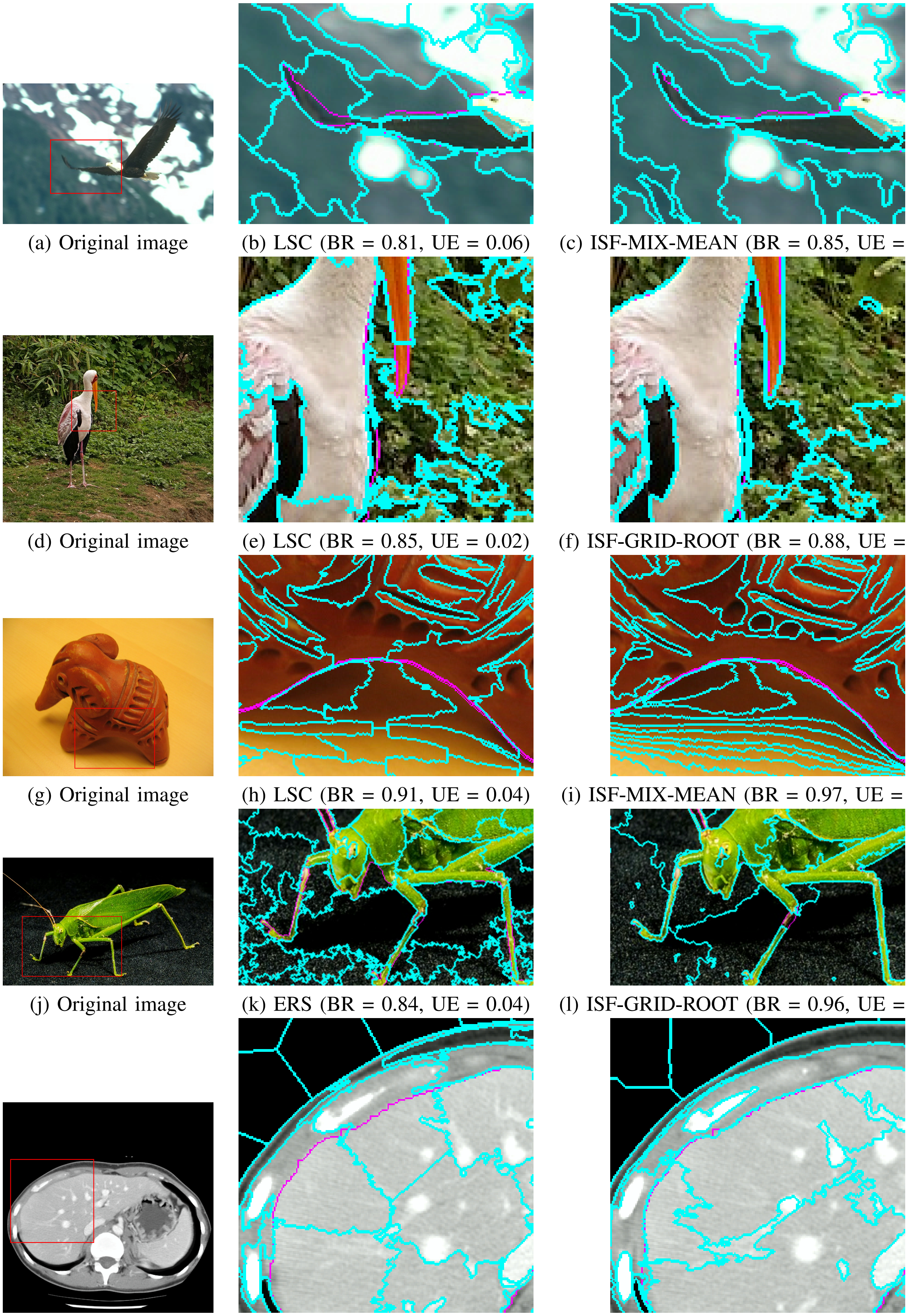
TABLE II
F-SCORE (MEAN +/- STD. DEVIATION) FOR CEREBELLUM, LEFT AND RIGHT BRAIN HEMISPHERES IN 3D MR-T1 IMAGES.

	$N = 1000$			$N = 5000$			$N = 10000$		
Method	F-score	Stdev	Time(sec)	F-score	Stdev	Time(sec)	F-score	Stdev	Time(sec)
SLIC	0.8584	0.0110	6.1	0.9194	0.0075	7.0	0.9369	0.0039	7.2
ISF-GRID-MEAN	0.8815	0.0129	31.8	0.9321	0.0069	30.3	0.9459	0.0051	29.9
SLIC + ISF (two iterations)	0.8686	0.0138	17.3	0.9305	0.0072	18.0	0.9444	0.0044	18.0

0.5 segments the patella bone as one of the supervoxels.

D. *Effectiveness on a high-level application*

When considering a high-level application, such as object segmentation based on superpixel labeling, the label assignment follows some independent and automatic rule. In this section, we evaluate the performance of the best ISF-based method (ISF-MIX-MEAN) for this application, namely *sky image segmentation*, in comparison with the fastest method (SLIC) and the most competitive baseline (LSC). We use a simple yet effective sky segmentation algorithm, as proposed in [2]. This algorithm uses the mean color of the superpixels and a threshold defined in the Lab color space to merge superpixels. The region (set of superpixels) at the top of the image that contains the larger number of pixels is selected as the sky region. Figure 10 shows the results of F-score for this experiment for a varying number of superpixels. Again, ISF with $\alpha = 0.08$ (more regular superpixels) performs better than the others. The parameter α has been found based on F-score and 5 training images, the 55 remaining images being



(a) Original image
(b) LSC (BR = 0.81, UE = 0.06)
(c) ISF-MIX-MEAN (BR = 0.85, UE = 0.05)
(d) Original image
(e) LSC (BR = 0.85, UE = 0.02)
(f) ISF-GRID-ROOT (BR = 0.88, UE = 0.01)
(g) Original image
(h) LSC (BR = 0.91, UE = 0.04)
(i) ISF-MIX-MEAN (BR = 0.97, UE = 0.02)
(j) Original image
(k) ERS (BR = 0.84, UE = 0.04)
(l) ISF-GRID-ROOT (BR = 0.96, UE = 0.01)
(m) Original image
(n) LSC (BR = 0.80, UE = 0.04)
(o) ISF-MIX-MEAN (BR = 0.93, UE = 0.01)

Fig. 8. Examples of superpixel segmentation on Berkeley (first row), Birds (second row), Grabcut (third row), Insects (fourth row), and Liver (fifth row). The first column shows the original images with a region of interest (in red rectangle). The second and third columns show the zoomed segmentation results of the most competitive baseline and the best ISF-based method for the region of interest in the corresponding dataset respectively. The superpixel borders are presented in cyan and the ground-truth borders in magenta (i.e., errors appear in magenta).

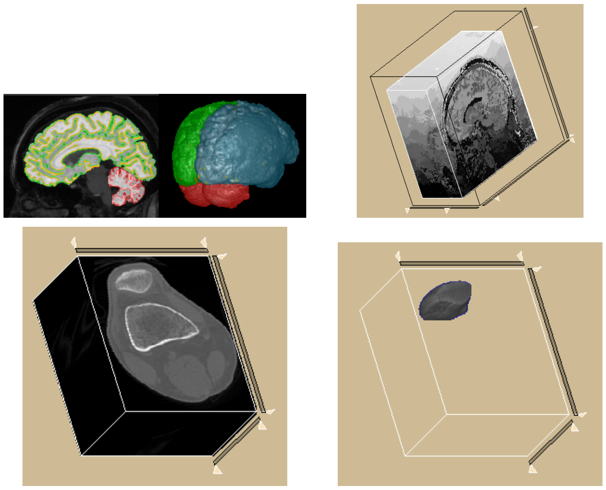


Fig. 9. (a) Cerebellum, left and right brain hemispheres from an MR image of the brain (top left). (b) Resulting supervoxels for one MR image of the brain (top right). (c) CT image of a knee (bottom left). (d) For a segmentation of 10 supervoxels, the patella bone is obtained as one of them (bottom right).

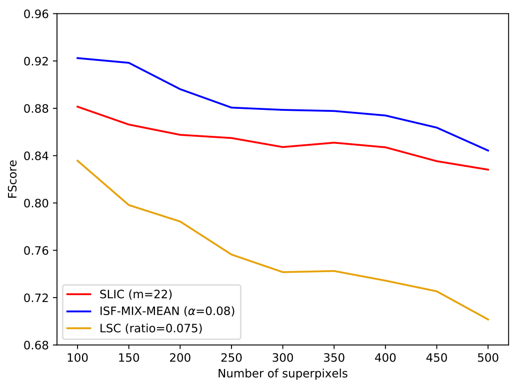


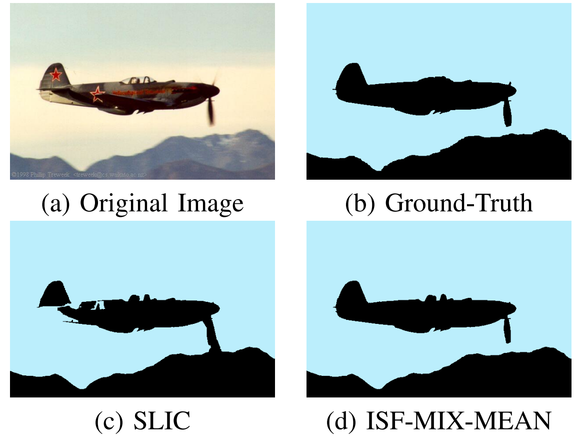
Fig. 10. Performance in F-score (Dice) for sky image segmentation: ISF-MIX-MEAN, SLIC, and LSC. Each method uses its best parameter values.

used for testing. Figure 11 presents one example of sky image segmentation by using SLIC and ISF-MIX-MEAN. The example illustrates the superiority of ISF-MIX-MEAN over SLIC for this application.

The use of lower values of α in the segmentation of 3D MR images of the brain and in this application strongly suggests that superpixel regularity has some importance as well as boundary adherence. It is also interesting to observe that SLIC outperforms LSC in this application.

E. Efficiency

SLIC is acknowledged as one of the fastest superpixel segmentation methods [54]. In this section, we compare the processing times on one of the datasets (Berkeley) for the ISF methods used in Section IV-B for different superpixel resolutions and values of the parameter α, SLIC, LSC, and ERS (the two most competitive methods in Berkeley). Table III shows the average processing time in seconds of the methods, without taking into account the I/O operations and pre-processing (e.g. RGB to Lab conversion), and using the



(a) Original Image (b) Ground-Truth

(c) SLIC (d) ISF-MIX-MEAN

Fig. 11. One example of sky image segmentation. (a) Original image, (b) ground-truth, and the segmentation results by using (c) SLIC and d) ISF-MIX-MEAN.

same machine specification used for Table II. Note that the optimized code of ISF can run faster with higher number of superpixels and lower value of α (more regular superpixels). This can be explained by the use of the new differential image foresting transform [53], whose processing time is $O(N \log N)$ where N is the number of pixels in the modified regions of the image. As the number of superpixels increases and their shapes become more compact, the sizes of the modified regions per iteration reduce. Note that ISF can be more efficient than LSC and ERS in general, and depending on the choices of α and number of superpixels, ISF can achieve processing time competitive with SLIC.

V. Conclusion

In this paper, we present an iterative spanning forest (ISF) framework, based on sequences of image foresting transforms (IFTs) for the generation of superpixels. The proposed framework provides us a lot of flexibility in terms of different choices of seed sampling strategies, connectivity functions, adjacency relations, and seed recomputation strategies. We also introduce a new seed sampling strategy, which can provide better results than grid sampling for most datasets, and new connectivity functions.

In the supplementary material, we prove that ISF converges and outputs connected superpixels — a property that avoids the post-processing step required in several other approaches. We also demonstrate by extensive experiments that the ISF superpixels can be computed fast with high value of boundary recall and low value of undersegmentation error. Different from the baselines, ISF can produce efficient methods with highly effective superpixel delineation independent of the dataset. In most cases, including high-level applications, such as sky segmentation, the methods can be competitive or superior to several state-of-the-art methods.

As shown, the compromise between boundary adherence and superpixel regularity in ISF can be controlled by properly choosing the parameter α in Equations 3 and 4. Indeed, more superpixel regularity has shown to be important for sky image segmentation and 3D MR image segmentation of the brain.

TABLE III
AVERAGE PROCESSING TIME FOR SUPERPIXEL SEGMENTATION IN THE BERKELEY DATASET.

	$N = 250$	$N = 500$	$N = 1000$	$N = 5000$
Method	Time (sec)	Time (sec)	Time (sec)	Time (sec)
ISF-MIX-MEAN ($\alpha = 0.5$)	0.248	0.227	0.199	0.127
ISF-MIX-MEAN ($\alpha = 0.12$)	0.158	0.129	0.101	0.067
ISF-MIX-MEAN ($\alpha = 0.04$)	0.075	0.066	0.057	**0.049**
ISF-GRID-ROOT ($\alpha = 0.5$)	0.250	0.249	0.243	0.201
ISF-GRID-ROOT ($\alpha = 0.12$)	0.257	0.253	0.236	0.159
ISF-GRID-ROOT ($\alpha = 0.04$)	0.244	0.235	0.210	0.127
SLIC	**0.036**	0.038	0.041	0.042
SLIC + ISF (two iterations)	0.104	0.105	0.108	0.109
ISF-REGMIN	0.055	0.056	0.057	0.057
LSC	0.257	0.259	0.262	0.267
ERS	0.952	1.012	1.065	1.224

This result requires further and more careful investigation. We also plan to pursue the ideas of incorporating object information in superpixel delineation [49] and of exploiting hierarchical superpixel segmentation [50] in high-level applications. The combination between deep-learning-based approaches and superpixel delineation is also a promising path, whenever a large number of annotated images is available [56].

ACKNOWLEDGMENTS

Ananda S. Chowdhury was supported through a FAPESP Visiting Scientist Fellowship under the grant $2015/01186-6$. The authors thank the financial support of CAPES, CNPq (grants $479070/2013-0$, $302970/2014-2$ and $308985/2015-0$) and FAPESP (grants $2016/14760-5$ and $2014/12236-1$). We thank Dr. J.K. Udupa (MIPG-UPENN) for the CT images of the liver and the project FAPESP-BRAINN for the MR images of the brain.

REFERENCES

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

[2] J. Kostolansky, "Sky segmentation using Slic superpixels," 2016. [Online]. Available: http://vgg.fiit.stuba.sk/2015-02/sky-detection-using-slic-superpixels/

[3] A. Ayvaci and S. Soatto, "Motion segmentation with occlusions on the superpixel graph," in *Proc. of the Workshop on Dynamical Vision, Kyoto, Japan*, October 2009, pp. 727–734.

[4] 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.

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

[6] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA*, 2013, pp. 3721–3727.

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

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

[9] 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.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

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

[12] Z. Tian, L. Liu, Z. Zhang, and B. Fei, "Superpixel-based segmentation for 3d prostate mr images," *IEEE Trans. on Medical Imaging*, vol. 35, no. 3, pp. 791–801, March 2016.

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

[14] 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.

[15] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Processing*, vol. 23, no. 4, pp. 1451–1462, 2014.

[16] 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. IEEE International Conf. on Computer Vision (ICCV)*, vol. 2, July 2001, pp. 416–423.

[17] V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, and E. Decencière, "Waterpixels," *IEEE Trans. on Image Processing*, vol. 24, no. 11, pp. 3707–3716, Nov 2015.

[18] 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.

[19] 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.

[20] Z. Ban, J. Liu, and L. Cao, "Superpixel segmentation using gaussian mixture model," *IEEE Trans. on Image Processing*, vol. 27, no. 8, pp. 4105–4117, Aug 2018.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[22] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[23] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. International Conf. on Computer Vision Pattern Recognition (CVPR) Anchorage, Alaska, USA*, 2008, pp. 1–8.

[24] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels

in an energy optimization framework," in *Proc. European Conf. on Computer Vision (ECCV), Heraklion, Crete, Greece*, 2010, pp. 211–224.

[25] R. A. Lotufo and A. X. Falcão, "The ordered queue and the optimality of the watershed approaches," in *Mathematical Morphology and its Applications to Image and Signal Processing*. Kluwer, Jun 2000, vol. 18, pp. 341–350.

[26] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Minimum spanning forests and the drop of water principle," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1362–1374, Aug 2009.

[27] C. Wilms and S. Frintrop, "Edge adaptive seeding for superpixel segmentation," in *German Conf. on Pattern Recognition*. Springer, 2017, pp. 333–344.

[28] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: live wire on the fly," *IEEE Trans. on Medical Imaging*, vol. 19, no. 1, pp. 55–62, Jan 2000.

[29] L. A. C. Mansilla, P. A. V. Miranda, and F. A. M. Cappabianco, "Image segmentation by image foresting transform with non-smooth connectivity functions," in *2013 XXVI Conf. on Graphics, Patterns and Images*, Aug 2013, pp. 147–154.

[30] R. da S. Torres, A. X. Falcão, and L. da F. Costa, "A graph-based approach for multiscale shape analysis," *Pattern Recognition*, vol. 37, no. 6, pp. 1163 – 1174, 2004.

[31] P. A. V. Miranda and L. A. C. Mansilla, "Oriented image foresting transform segmentation by seed competition," *IEEE Trans. on Image Processing*, vol. 23, no. 1, pp. 389–398, Jan 2014.

[32] T. V. Spina, P. A. V. de Miranda, and A. X. Falcão, "Hybrid approaches for interactive image segmentation using the live markers paradigm," *IEEE Trans. on Image Processing*, vol. 23, no. 12, pp. 5756–5769, Dec 2014.

[33] A. Freitas, R. da S. Torres, and P. Miranda, "TSS & TSB: Tensor scale descriptors within circular sectors for fast shape retrieval," *Pattern Recognition Letters*, vol. 83, pp. 303 – 311, 2016, efficient Shape Representation, Matching, Ranking, and its Applications.

[34] A. X. Falcão, C. Feng, J. Kustra, and A. Telea, *Multiscale 2D medial axes and 3D surface skeletons by the image foresting transform*. Academic Press, 2017, ch. 2, pp. 43–67.

[35] A. C. M. Tavares, P. A. V. Miranda, T. V. Spina, and A. X. Falcão, *A Supervoxel-Based Solution to Resume Segmentation for Interactive Correction by Differential Image-Foresting Transforms*. Springer International Publishing, 2017, pp. 107–118.

[36] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[37] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[38] J. P. Papa, A. X. Falcão, V. H. C. de Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, Jan. 2012.

[39] W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, pp. 72–85, Dec. 2016.

[40] J. P. Papa, S. E. N. Fernandes, and A. X. Falcão, "Optimum-path forest based on k-connectivity: Theory and applications," *Pattern Recognition Letters*, vol. 87, pp. 117 – 126, 2017, advances in Graph-based Pattern Recognition.

[41] P. A. Miranda and A. X. Falcão, "Links between image segmentation based on optimum-path forest and minimum cut in graph," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 128–142, Oct 2009.

[42] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[43] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. European Conf. on Computer Vision (ECCV), Marseille, France*, 2008, pp. 705–718.

[44] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[45] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 1–21, 2013.

[46] C. Çiğla and A. A. Alatan, "Efficient graph-based image segmentation via speeded-up turbo pixels," in *Proc. IEEE International Conf. on Image Processing (ICIP), Hong Kong, China*, 2010, pp. 3013–3016.

[47] 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 *28th SIBGRAPI: Conf. on Graphics, Patterns and Images*. IEEE, 2015, pp. 337–344.

[48] A. X. Falcão and F. P. G. Bergo, "Interactive volume segmentation with differential image foresting transforms," *IEEE Trans. on Medical Imaging*, vol. 23, no. 9, pp. 1100–1108, 2004.

[49] F. de C. Belém, S. J. F. Guimarães, and A. X. Falcão, "Superpixel segmentation by object-based iterative spanning forest." in *23rd Iberoamerican Congress on Pattern Recognition (CIARP)*, 2018, to appear.

[50] F. L. Galvão, A. S. Chowdhury, and A. X. Falcão, "Recursive iterative spanning forest for superpixel segmentation," in *28th SIBGRAPI: Conf. on Graphics, Patterns and Images*. IEEE, 2018, to appear.

[51] 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*, vol. 60, no. 7, pp. 1025–1036, Sep 2018.

[52] 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.

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

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

[55] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM trans. on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[56] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz, "Learning superpixels with segmentation-aware affinity loss," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.