



The Role of Optimum Connectivity in Image Segmentation: Can the Algorithm Learn Object Information During the Process?

Alexandre Falcão^(✉)  and Jordão Bragantini^(✉)

University of Campinas, Campinas, SP 13083-852, Brazil
afalcao@ic.unicamp.br, jordao.bragantini@gmail.com

Abstract. Image segmentation is one of the most investigated research topics in Computer Vision and yet presents challenges due to the difficulty of modeling all possible appearances of objects in images. In this sense, it is important to investigate methods that can learn object information before and during delineation. This paper addresses the problem by exploiting optimum connectivity between image elements (pixels and superpixels) in the image domain and feature space to improve segmentation. The study uses the Image Foresting Transform (**IFT**) framework to explain and implement all methods and describes some recent advances related to superpixel and object delineation. It provides a guideline to learn prior object information from the target image only based on seed pixels, superpixel clustering, and classification, evaluates the impact of using object information in several connectivity-based delineation methods using the segmentation by a deep neural network as baseline, and shows the potential of a new paradigm, namely *Dynamic Trees*, to learn object information from the target image only during delineation.

Keywords: Superpixel and object segmentation ·
Seed-based delineation algorithms · Image Foresting Transform

1 Introduction

The definition of the exact spatial extent of relevant image segments for shape and texture analysis, namely *segmentation*, is certainly one of the most challenging problems in Computer Vision and Image Processing. Such segments may be *superpixels*—connected regions in which the pixels share common texture properties—or *objects*—connected components with characteristic shapes.

Superpixel segmentation aims at considerably reducing the number of image elements for analysis, such that the objects of interest can still be represented by the union of their superpixels (Fig. 1a). The problem requires an accurate

FAPESP 2014/12236-1, 2018/08951-8, and CNPq 302970/2014-2.

© Springer Nature Switzerland AG 2019

M. Couprie et al. (Eds.): DGCI 2019, LNCS 11414, pp. 180–194, 2019.

https://doi.org/10.1007/978-3-030-14085-4_15

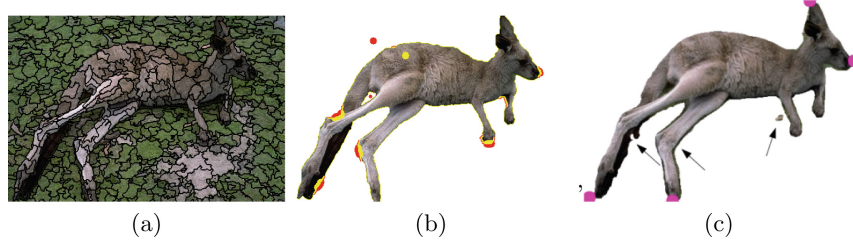


Fig. 1. (a) Superpixel segmentation by recursive spanning forests [21]. Object segmentation by (b) live markers [36] with internal (yellow) and external (red) seeds, some created from live-wire segments [20]. (c) Deep Extreme Cut (**DEC**) [25] with the four extreme points (magenta) needed for object localization. Arrows indicate three delineation errors in (c). (Color figure online)

delineation algorithm from seed pixels with one seed per superpixel. The first set of seed pixels *localizes* the superpixels and a few iterations of superpixel delineation and seed recomputation refine segmentation [1, 5, 9, 21]. Superpixels are then expected to increase the efficiency of higher level operations without compromising their effectiveness. One example is object segmentation using a superpixel graph [32].

Localization and delineation are also crucial tasks for object segmentation. Localization requires to specify the whereabouts of the object in the image—e.g., by providing a bounding box around the object [34], selecting points on its boundary [20], or drawing scribbles inside and outside the object [15, 16, 32]. Humans can more easily locate the object than machines. From effective localization, a delineation algorithm can define the spatial extent of the object in the image (Fig. 1b). Machines are more precise, not subject to fatigue, and so better than humans for object delineation.

A single iteration of localization and delineation does not usually solve object segmentation, which often requires multiple user interventions. Prior object information—i.e., texture [10, 24, 25] and shape [18, 26, 31, 39] models built from a training set with interactively segmented images—have been proposed to automate the segmentation process. Without connectivity-based delineation algorithms, these models may force their prior knowledge ignoring the appearance of the object in the image. For example, after a user indicates four extreme points to localize the object, a texture model based on deep learning [25] can provide an impressive object approximation (Fig. 1c). However, the problem is not solved in a single iteration and the method does not allow further user intervention. This suggests the combination of texture and shape models with connectivity-based delineation algorithms [10, 31], methods that can automatically adapt the model to the target image [26], and methods that can learn object information during delineation [8].

In this paper, we demonstrate the role of optimum connectivity in image segmentation through the challenge of learning object information from the target

image only. The strategy aims at improving object delineation with minimum user intervention. The *Image Foresting Transform* (**IFT**) is the framework [19] of choice since it can unify several methods used for delineation [5, 8, 21, 32, 36], clustering [33], and classification [2, 3, 30], by exploiting connectivity between a seed set and the remaining image elements. We combine some of these methods to learn object information before and during segmentation, and evaluate their impact on object delineation.

Section 2 presents the **IFT** framework with examples of adjacency relations and connectivity functions for superpixel and object segmentation. Sections 3 and 4 cover some recent advances in **IFT**-based superpixel and object segmentation, by calling attention to dynamic trees [8]—a new paradigm that learns object information during the delineation process. Section 4.3 then explains how to use superpixel segmentation, clustering, and classification to learn object information from the target image before object delineation. Section 5 evaluates the results of incorporating object information in connectivity-based delineation algorithms. Finally, Sect. 6 states conclusion and suggests research directions related to image segmentation.

2 Image Foresting Transform (IFT)

In the **IFT** framework [19], an image is interpreted as a graph in which the nodes may be pixels, superpixels, or any other element from the image domain, each one being a point in some n -dimensional feature space. The arcs of the graph must satisfy some *adjacency relation*, which may be defined in the image domain and/or feature space. The **IFT** uses Dijkstra’s algorithm, with more general *connectivity functions* and multiple sources [12], to transform the image graph into an optimum-path forest rooted at the minima of a resulting cost map (e.g., a seed set for delineation), then reducing the image operator to a local processing of its attributes (optimum paths, their costs, and root labels).

2.1 Images as Graphs

Let $\hat{I} = (D_I, \mathbf{I})$ be a 2D natural image in which $\mathbf{I}(p) \in \mathbb{R}^3$ represents the CIE Lab color components of each pixel $p \in D_I \subset \mathbb{Z}^2$. Image \hat{I} may be interpreted as a weighted graph $(\mathcal{N}, \mathcal{A}, w)$ in several ways such that the nodes $s, t \in \mathcal{N}$ may be pixels or superpixels, the arcs $(s, t) \in \mathcal{A}$ may connect nodes in the image domain and/or feature space, depending on the *adjacency relation* $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$, and the arc weights $w(s, t)$ may be defined from image properties and object information.

Let Π be the set of all possible paths in $(\mathcal{N}, \mathcal{A}, w)$. A path $\pi_{r \rightarrow s}$ is a sequence $\langle r = t_1, t_2, \dots, t_n = s \rangle$ of adjacent nodes $(t_i, t_{i+1}) \in \mathcal{A}$, $i = 1, 2, \dots, n - 1$, being π_s any path with terminus s , $\pi_s = \langle s \rangle$ a *trivial* path, and $\pi_s \cdot \langle s, t \rangle$ the concatenation of π_s and $\langle s, t \rangle$ with the joining instances of s merged into one. For a suitable *connectivity function* $f: \Pi \rightarrow \mathbb{R}$, the **IFT** algorithm on $(\mathcal{N}, \mathcal{A}, w)$ can output an *optimum-path forest* P —i.e., a spanning forest (acyclic map) that

assigns to all nodes $t \in \mathcal{N}$ a predecessor $P(t) = s \in \mathcal{N}$ in the optimum path with terminus t or a marker $P(t) = \text{nil} \notin \mathcal{N}$ when t is a *root* of the map—by minimizing a path-cost map V (also called connectivity map).

$$V(t) = \min_{\forall \pi_t \in \Pi} \{f(\pi_t)\}. \quad (1)$$

The map V can also be maximized depending on the connectivity function and the sufficient conditions for optimal cost mapping have been revised recently [12]. Even when these conditions are not satisfied, the algorithm can output a spanning forest useful for superpixel [5] and object [28] segmentation, being the minimization criterion a graph-cut measure in the latter.

2.2 Adjacency Relations

Let $\mathcal{K}(s, k)$ be the set of the k -closest nodes $t \in \mathcal{N}$ of s in some feature space (e.g., the CIELab color space) according to the Euclidean metric $\|\cdot, \cdot\|$. Examples of adjacency relations used in this work are given below.

$$\mathcal{A}_1 = \{s, t \in \mathcal{N} \mid \|s, t\| \leq d\}, \quad (2)$$

$$\mathcal{A}_2 = \{s, t \in \mathcal{N} \mid t \in \mathcal{K}(s, k), 1 \leq k \ll |\mathcal{N}|\}, \quad (3)$$

$$\mathcal{A}_3 = \{s, t \in \mathcal{N} \mid s \neq t\}, \quad (4)$$

where \mathcal{A}_1 with $d = 1$ and $\mathcal{N} = D_I$ (4-neighbor graph) is used for superpixel and object delineation (Sects. 3 and 4), \mathcal{A}_2 (k -nearest neighbor graph) is used for clustering of superpixels (Sect. 4.3), and \mathcal{A}_3 (complete graph) is used to create an *object membership map* O based on pixel classification from a small training subset $\mathcal{N} \subset D_I$ (Sect. 4.3).

2.3 Algorithm and Connectivity Functions

The **IFT** algorithm starts with all nodes $s \in \mathcal{N}$ being trivial paths $\langle s \rangle$ with values $f(\langle s \rangle) = V_o(s) \geq f(\pi_s)$ for all $\pi_s \in \Pi$. This step sets $V(s) \leftarrow V_o(s)$ and $P(s) \leftarrow \text{nil}$, and inserts all nodes in a priority queue Q . At each iteration, a node s with minimum cost $V(s)$ is removed from Q and it offers the extended path $\pi_s \cdot \langle s, t \rangle$ to its adjacent nodes in Q . Whenever $V(t) > f(\pi_s \cdot \langle s, t \rangle)$, the node s conquers the node t by substituting its current path π_t by $\pi_s \cdot \langle s, t \rangle$. This step updates $V(t) \leftarrow f(\pi_s \cdot \langle s, t \rangle)$ and $P(t) \leftarrow s$, and the process stops after $|\mathcal{N}|$ iterations. The roots of the forest can be forced to be a *seed set* $\mathcal{S} \subset \mathcal{N}$, when $V_o(s)$ is defined as

$$V_o(s) = \begin{cases} 0 & \text{if } s \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \quad (5)$$

Otherwise, the root set \mathcal{R} derives from the minima (flat zones) of V (a subset of the flat zones of V_o). It is also possible to guarantee a single root per minimum if the nodes are inserted in Q with costs $f(\langle s \rangle) = V_o(s) + \delta$, where

$0 < \delta = \min_{\forall (s,t) \in \mathcal{A}} \{f(\pi_s \cdot \langle s, t \rangle) - f(\pi_s)\}$ for monotonically incremental path-cost functions [19]. By that, when a node s is removed from Q with $P(s) = \text{nil}$, it has never been conquered by another node and so it is immediately identified as a root in \mathcal{R} with cost updated to $V_o(s)$. This is equivalent to define

$$f(\langle s \rangle) = \begin{cases} V_o(s) & \text{if } s \in \mathcal{R}, \\ V_o(s) + \delta & \text{otherwise.} \end{cases} \quad (6)$$

Examples of connectivity functions for non-trivial paths are given below. They usually use Eq. 5 for trivial paths, but Sect. 4.3 shows one example of f_1 with trivial-path cost given by Eq. 6.

$$f_1(\pi_s \cdot \langle s, t \rangle) = \max\{f_1(\pi_s), w(s, t)\}, \quad (7)$$

$$f_2(\pi_s \cdot \langle s, t \rangle) = w(s, t), \quad (8)$$

$$f_3(\pi_{r \rightarrow s} \cdot \langle s, t \rangle) = f_3(\pi_{r \rightarrow s}) + aw^b(r, t) + \|s, t\|, \quad (9)$$

where r is the root of s , $a > 0$ and $b \geq 1$. Clearly, the success of the image operations strongly depends on the *arc-weight function* w . Next sections illustrate different choices of w and connectivity functions for superpixel and object segmentation, clustering, and classification.

3 Superpixel Segmentation

Superpixel segmentation has evolved as an important topic with several applications, such as object detection, tracking, and segmentation. A superpixel is a connected image region in which pixels present similar texture properties. However, some methods cannot guarantee connected superpixels during delineation [1, 9], except by post-processing. In [40], the authors propose a framework, named *Iterative Spanning Forest (ISF)*, for the generation of connected superpixels by choice of four independent components: (a) a seed sampling strategy, (b) an adjacency relation, (c) a connectivity function, and (d) a seed recomputation procedure. They present and evaluate different choices for those components. Similar to some popular approaches, for a desired number of superpixels, the **ISF** algorithm starts by finding an initial seed set \mathcal{S} that represents meaningful locations for those superpixels. Subsequently, the **IFT** algorithm followed by seed recomputation is executed multiple times to improve the seed set \mathcal{S} and refine superpixel delineation. Connected superpixels require the adjacency relation in Eq. 2 with $d = 1$ for 2D and 3D images (4- and 6-neighbor graphs, respectively). Equations 5 and 9 represent one example of connectivity function in which $\|s, t\|$ contributes to obtain compact superpixels, $w^b(r, t) = \|\mathbf{I}(r), \mathbf{I}(t)\|^b$ increases the boundary adherence for higher values of b (e.g., $b = 12$), and a (e.g., $a \in \{0.08, 0.1, 0.2, 0.5\}$) controls the compromise between boundary adherence and compactness of the superpixels. In **ISF**, each connected superpixel is a spanning tree rooted at one seed pixel $r \in \mathcal{S}$.

In [5], the authors present the first **ISF**-based method and the authors in [38] propose a connectivity function that increases the boundary adherence of superpixels to the boundary of a given object mask created by automated segmentation. The method can be used to transform any object segmentation into an optimum-path forest rooted at superpixel seeds for correction by differential **IFTs** [17]. By adapting the connectivity function in [38], the authors in [6] incorporate object information in superpixel delineation such that **ISF** can considerably reduce the number of superpixels while the object representation is preserved. The method is called *Object-based ISF (OISF)*. **ISF** can also be executed recursively on superpixel graphs to create a hierarchy of segmentations [21] for distinct applications. This hierarchy is explored in [21] to introduce a geodesic seed sampling strategy, which can considerably improve superpixel delineation.

At each execution of the **IFT** algorithm in the **ISF**-based methods, the seed set partially changes by preserving the most stable seeds from previous executions, eliminating the unstable ones, and adding new seeds for replacement. For the sake of efficiency, this requires the update of the spanning forest in a differential way. However, the differential **IFT** algorithm [17] does not apply for non-monotonically incremental connectivity functions [19]. A solution for root-based connectivity functions (e.g., Eq. 9) has been presented in [14].

In this work, the superpixel representation reduces the number of image elements from 481×321 pixels (the resolution of the images in the GrabCut dataset [34]) to about 800 superpixels. We then use the method in [33] to find groups in the superpixel set much faster and with higher efficiency than using the pixel set (Sect. 4.3). For pixel sets, this algorithm is usually applied to a considerably smaller subset of randomly selected pixels and then the group labels are propagated to the remaining pixels. In our case, the algorithm can use the entire image content as represented by the superpixel set.

4 Object Segmentation

In the **IFT** framework, object localization is usually solved by selecting seeds on the object's boundary [20, 27, 36] or inside and outside the object [11, 13, 19, 28, 29, 32], being some approaches hybrid (e.g., Fig. 1b [36]). Region-based approaches can be naturally extended to 3D images and so more easily combined with shape models for automated segmentation [26, 31]. In this section, our focus is on region-based object delineation algorithms from a seed set $\mathcal{S} = \mathcal{S}_o \cup \mathcal{S}_b$ with internal (\mathcal{S}_o) and external (\mathcal{S}_b) pixels. These algorithms run on 4-neighbor graphs and output two sets in a label map L , the object set $\mathcal{O} = \{p \in D_I, L(p) = 1\}$ and the background set $\mathcal{B} = \{p \in D_I, L(p) = 0\}$, such that $\mathcal{O} \cup \mathcal{B} = D_I$ and $\mathcal{O} \cap \mathcal{B} = \emptyset$. The partition defines a graph cut $\mathcal{C} = \{\forall (p, q) \in \mathcal{A} \mid p \in \mathcal{O}, q \in \mathcal{B}\}$. We are interested in algorithms that define connected sets \mathcal{O} and \mathcal{B} , with $\mathcal{S}_o \subset \mathcal{O}$ and $\mathcal{S}_b \subset \mathcal{B}$, and optimize one of the two graph-cut measures below [13].

$$\mathbf{GC}_{\max} = \max_{\forall \mathcal{C} \subset \mathcal{A}} \left\{ \min_{\forall (p,q) \in \mathcal{C}} \{w(p,q)\} \right\}, \quad (10)$$

$$\mathbf{GC}_{\text{sum}} = \min_{\forall \mathcal{C} \subset \mathcal{A}} \left\{ \sum_{\forall (p,q) \in \mathcal{C}} w(p,q) \right\}. \quad (11)$$

For \mathbf{GC}_{sum} , we use the min-cut/max-flow algorithm [7] and the **IFT** algorithm [12] is used for \mathbf{GC}_{\max} . Note that, $w(p,q)$ must be higher on the object's boundary than elsewhere for \mathbf{GC}_{\max} and lower on the object's boundary than elsewhere for \mathbf{GC}_{sum} . In order to avoid small cuts in \mathbf{GC}_{sum} , we must adopt an exponent $\beta > 1$ when computing $w(p,q)$. Note that, for $\beta \gg 1$, \mathbf{GC}_{sum} should provide similar results to \mathbf{GC}_{\max} [29].

\mathbf{GC}_{sum} extends the image graph by adding binary arc weights, with values either 0 or $+\infty$, between pixels and the source (object) and sink (background) seeds in \mathcal{S} , while the **IFT** algorithm uses Eqs. 5 and 7 to satisfy the constraints. In the **IFT** algorithm, the seeds in \mathcal{S}_o and \mathcal{S}_b compete among themselves such that the object \mathcal{O} is defined in P by the optimum-path trees rooted in \mathcal{S}_o .

4.1 Localization by a User Robot

In order to avoid user subjectivity in selecting seeds, we use a geodesic user robot [22], which uses the ground-truth segmentation to simulate the user knowledge about the object. For the first iteration of object delineation, the robot selects one marker (i.e., a pixel and its four neighbors) at the geodesic center of each, object and background. At each subsequent iteration, the largest error components of both, object and background, are identified and the robot selects one marker at the geodesic center of each one. The markers must be at least two pixels from the object's boundary and the process proceeds for 15 iterations, but it can halt earlier when the error components are small to contain markers. As it occurs in practice, distinct methods require seeds in different locations to correct segmentation. In [8], we use benchmarks of seed sets, but the methods are compared for a single execution only.

4.2 Arc-Weight Assignment

We consider the following arc-weight assignment functions.

$$w_1(p,q) = \|\mathbf{I}(p), \mathbf{I}(q)\|, \quad (12)$$

$$w_2(p,q) = (1 - \alpha)w_1(p,q) + \alpha|O(p) - O(q)|, \quad (13)$$

$$w_3(p,q) = \left(1 - \frac{w_1(p,q)}{dI_{\max}}\right)^\beta, \quad (14)$$

$$w_4(p,q) = \left[(1 - \alpha) \left(1 - \frac{w_1(p,q)}{dI_{\max}}\right) + \alpha \left(1 - \frac{|O(p) - O(q)|}{dO_{\max}}\right)\right]^\beta, \quad (15)$$

$$w_5(p,q) = \|\mu_{R(p)}, \mathbf{I}(q)\|, \quad (16)$$

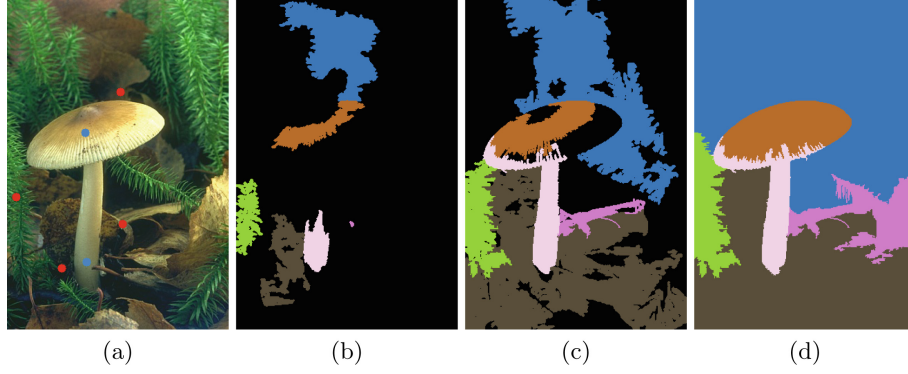


Fig. 2. (a) Original image with object (blue) and background (red) markers. (b-c-d) Dynamic tree growth during segmentation. Each color is a different optimum-path tree. (Color figure online)

$$w_6(p, q) = (1 - \alpha)w_5(p, q) + \alpha|O(p) - O(q)|, \quad (17)$$

$$w_7(p, q) = w_5(p, q) + \|\mathbf{I}(p), \mathbf{I}(q)\|, \quad (18)$$

$$w_8(p, q) = (1 - \alpha)w_7(p, q) + \alpha|O(p) - O(q)|, \quad (19)$$

$$w_9(p, q) = \min_{\forall r \in \mathcal{S} | L(r) = L(p)} \{\|\mu_r, \mathbf{I}(q)\|\}, \quad (20)$$

$$w_{10}(p, q) = (1 - \alpha)w_9(p, q) + \alpha|O(p) - O(q)|, \quad (21)$$

$$w_{11}(p, q) = w_9(p, q) + \|\mathbf{I}(p), \mathbf{I}(q)\|, \quad (22)$$

$$w_{12}(p, q) = (1 - \alpha)w_{11}(p, q) + \alpha|O(p) - O(q)|, \quad (23)$$

where O is an object map (Sect. 4.3), $dI_{max} = \max_{\forall(p,q) \in \mathcal{A}_1} \{\|\mathbf{I}(p), \mathbf{I}(q)\|\}$, $dO_{max} = \max_{\forall(p,q) \in \mathcal{A}_1} \{|O(p) - O(q)|\}$, $L(p) \in \{0, 1\}$ indicates which set p belongs to, either \mathcal{O} or \mathcal{B} , by the time $w(p, q)$ is estimated, α controls the balance between object and image information (e.g., $\alpha = 0.25$), β avoids small cuts in \mathbf{GC}_{sum} (e.g., $\beta = 70$), and $\mu_{R(p)}$ is the mean CIELab color of the optimum-path tree that contains p with root $R(p) \in \mathcal{S}$ at the moment the arc weight $w(p, q)$ is computed. In the **IFT** algorithm, at this moment the optimum-path trees have already been assigned to either \mathcal{O} or \mathcal{B} . Note that, Eqs. 12, 13, 16–23 are used for \mathbf{GC}_{max} while Eqs. 14 and 15 are used for \mathbf{GC}_{sum} . A new paradigm, named *dynamic trees* [8], is defined by Eqs. 16–23. In this paradigm, object information is extracted from the growing optimum-path trees to estimate the arc weights during the process. Figure 2 illustrates the growing process of those trees for three distant iterations. One can see that, as they grow, it is possible to obtain shape and texture information. We have evaluated two simple measures so far—the mean color μ_{Rp} of the tree that contains p (Eqs. 16–19) and, for the object/background forest that contains p , the mean color μ_r of the tree closest

to the color of p (Eqs. 20–23). Although we are addressing binary segmentation, \mathbf{GC}_{\max} is extensive to multiple object segmentation with no additional cost.

4.3 Learning Prior Object Information from Input Markers

Equations 16 and 20 use only object information as extracted and represented by $\mu_{R(p)}$ and μ_r , respectively, during delineation. In addition to that, object information can also be extracted from input markers by executing another **IFT** prior to object delineation. The process takes as input the markers $\mathcal{S} = \mathcal{S}_o \cup \mathcal{S}_b$ on the target image and outputs an *object map* O , which is used for arc-weight assignment in Eqs. 13, 15, 17, 19, 21, and 23. Other knowledge about objects, such as a hierarchy among them [23] and border polarity [20, 28], may also be used.

An object map can be created from markers inside and outside the object when those markers are interpreted as training sets for a pattern classifier [29]. By defining the graph $(\mathcal{N}, \mathcal{A}, w)$ such that $\mathcal{N} \subset \mathcal{S}$ (*some representative seeds*) in \mathcal{S}_o and \mathcal{S}_b , \mathcal{A} is given by Eq. 4, and $w = \|\mathbf{I}(t), \mathbf{I}(s)\|$, the **IFT** algorithm for path-cost function f_1 (Eqs. 5 and 7) can output two optimum-path forests, one rooted in \mathcal{S}_o and the other rooted in \mathcal{S}_b , with the paths first computed in \mathcal{S} and then propagated to the remaining pixels in $D_I \setminus \mathcal{S}$ [30]. The object map O is defined as $O(p) = \frac{V^{(b)}(p)}{V^{(o)}(p) + V^{(b)}(p)}$ from the costs $V^{(o)}(p)$ and $V^{(b)}(p)$ of the optimum paths that reach each pixel $p \in D_I$ from its roots in \mathcal{S}_o and \mathcal{S}_b , respectively [29]. However, object and background markers selected in regions with similar properties can impair the object map. The problem is solved in [35] by a method that eliminates seeds with similar properties and distinct labels that fall in a same cluster of the color space. Figure 3 illustrates the process using markers drawn by real users from the database in [4]. Intelligent seed selection selects groups (Fig. 3a) with a minimum percentage (e.g., 80%) of seeds from the same label (Fig. 3b). Without seed selection, the object map (above) is impaired.

For clustering, we use a graph $(\mathcal{N}, \mathcal{A}, w)$ such that \mathcal{N} is a superpixel set with 800 superpixels [21], \mathcal{A} is given by Eq. 3, and $w = \|\mathbf{I}(t), \mathbf{I}(s)\|$, being $\mathbf{I}(s)$ the mean CIELab color of superpixel s . The **IFT** algorithm for path-cost function f_1 (Eqs. 6 and 7) can output one optimum-path tree (cluster) rooted at each minimum of $V_o(s)$, being $V_o(s)$ the complement of a probability density function (**pdf**) [33]. The parameter k in $\mathcal{K}(s, k)$ is found by optimization in $[1, k_{\max}]$ (e.g., $k_{\max} = 24$) as the one that produces a minimum normalized cut in the graph [33]. This way each dome of the estimated pdf becomes one cluster.

5 Experimental Results

In order to evaluate object delineation using \mathbf{GC}_{\max} and \mathbf{GC}_{sum} , we first randomly divided the GrabCut dataset [34] in two parts: 50% of the images to optimize the parameters of the methods and 50% of the images to evaluate the

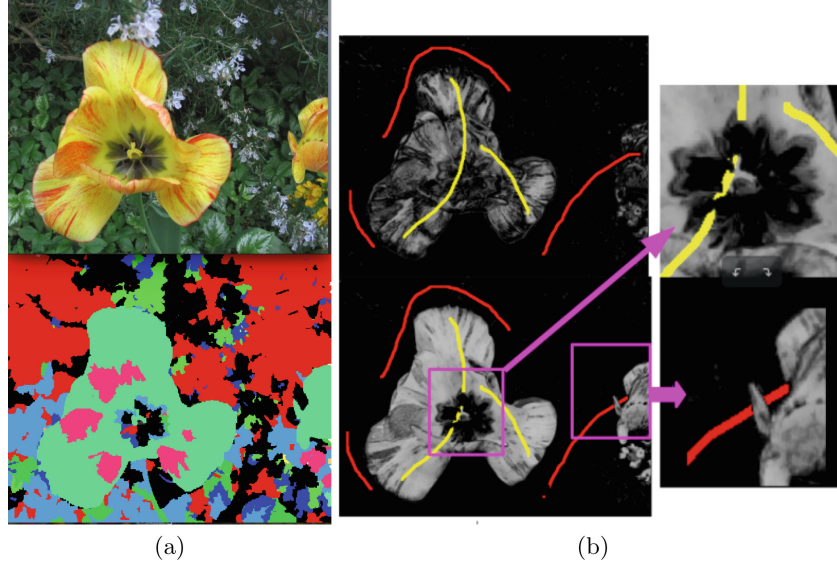


Fig. 3. (a) Original image (top) and clusters of superpixels (bottom). (b) Object map O without (top) and with (bottom and zoomed regions on the right) intelligent seed selection.

methods. The number of superpixels was found in $[100, 1000]$, by keeping $a = 0.5$ and $b = 12$ fixed in Eq. 9 [40], using two hierarchy levels [21], being the first level with three times more superpixels, and classifying as part of the object the superpixels with at least 80% of the pixels inside the ground-truth object mask. The number 800 was the minimum to obtain a segmentation accuracy (Dice coefficient) above 98%. The clustering parameter k_{max} [33] was found in $[8 - 80]$ and the purity of the clusters for seed selection was found in $[50\% - 95\%]$ by using the real user markers from [4] to generate an object map and then measuring the accuracy of segmentation based on the binary classification of the map at 50%. The best values were $k_{max} = 24$ and purity equal to 80%. Afterwards, α and β in Eqs. 12–23 were searched in $[0.0 - 0.95]$ and $[50 - 150]$, respectively, based on the accuracy of delineation. The best values were $\alpha = 0.25$ for w_2 , $\alpha = 0.1$ for w_4 , w_{10} and w_{12} , $\alpha = 0.2$ for w_6 and w_8 , which indicates that the color of the pixels are more important than the object map, and $\beta = 70$ for \mathbf{GC}_{sum} . We have also included Deep Extreme Cut (DEC) [25] as baseline and optimized the threshold on its probability map. We found the best threshold at 85% in $[50\%, 95\%]$ based on the accuracy of segmentation.

Table 1 presents the experimental results on the test images when the geodesic user robot is used for at most 15 iterations. In general, methods based on \mathbf{GC}_{max} with dynamic arc-weight assignment—i.e., Dynamic Trees (DT) [8]—

Table 1. Mean dice coefficient, standard deviation, and number of iterations required for convergence using each method, \mathbf{GC}_{\max} , $\mathbf{GC}_{\max}\text{-DT}$ (Dynamic Trees), \mathbf{GC}_{sum} , and \mathbf{DEC} (Deep Extreme Cut).

Arc-weight	Mean accuracy (dice)	Standard dev.	Iterations
$GC_{\max}\text{-DT } w_{10}$	0.970	0.0237	11.3
$GC_{\max}\text{-DT } w_8$	0.968	0.0295	11.8
$GC_{\max}\text{-DT } w_6$	0.966	0.0364	12.0
$GC_{\max} w_2$	0.961	0.0421	12.6
$GC_{\max}\text{-DT } w_7$	0.961	0.0491	11.9
$GC_{\max}\text{-DT } w_9$	0.959	0.0498	10.9
$GC_{\max}\text{-DT } w_5$	0.958	0.0458	11.8
$GC_{\max}\text{-DT } w_{12}$	0.958	0.0878	11.4
$GC_{\max}\text{-DT } w_{11}$	0.957	0.0599	11.5
\mathbf{DEC}	0.942	0.0814	4.00
$GC_{\max} w_1$	0.933	0.0854	13.8
$GC_{\text{sum}} w_4$	0.932	0.0845	13.3
$GC_{\text{sum}} w_3$	0.918	0.1080	13.6

provide the best results, not only in accuracy but also in the number of iterations, which can be related to less user intervention. \mathbf{DEC} reduces user intervention to four extreme points (4 iterations), which are inserted by a robot based on the ground truth, but it does not allow user corrections. The object map plays an important role to improve the results, making \mathbf{GC}_{\max} with w_2 more competitive. The good performance of $\mathbf{GC}_{\max}\text{-DT}$ with w_9 and w_5 indicates that Dynamic Trees can learn object information from the tree and forest that contain p at the moment $w(p, q)$ is estimated. On the other hand, the local distance $\|\mathbf{I}(p), \mathbf{I}(q)\|$ seems to have a negative influence on $\mathbf{GC}_{\max}\text{-DT}$, when using the forest that contains p (w_9 and w_{11}), but the same is not observed when using the tree that contains p (w_5 and w_7). The object map can also improve \mathbf{GC}_{sum} , but it is clear that the small size of the markers placed by the robot affects more \mathbf{GC}_{sum} than \mathbf{GC}_{\max} [13, 29]. Given that \mathbf{DEC} needs training from several segmented images, these results are a strong indication that optimum connectivity is crucial not only for delineation but also to extract object information from target images with no prior training.

Finally, a better idea about these numbers can be obtained in Fig. 4 when comparing $\mathbf{GC}_{\max}\text{-DT}$ with w_{10} and w_9 against \mathbf{DEC} and \mathbf{GC}_{sum} with w_3 .

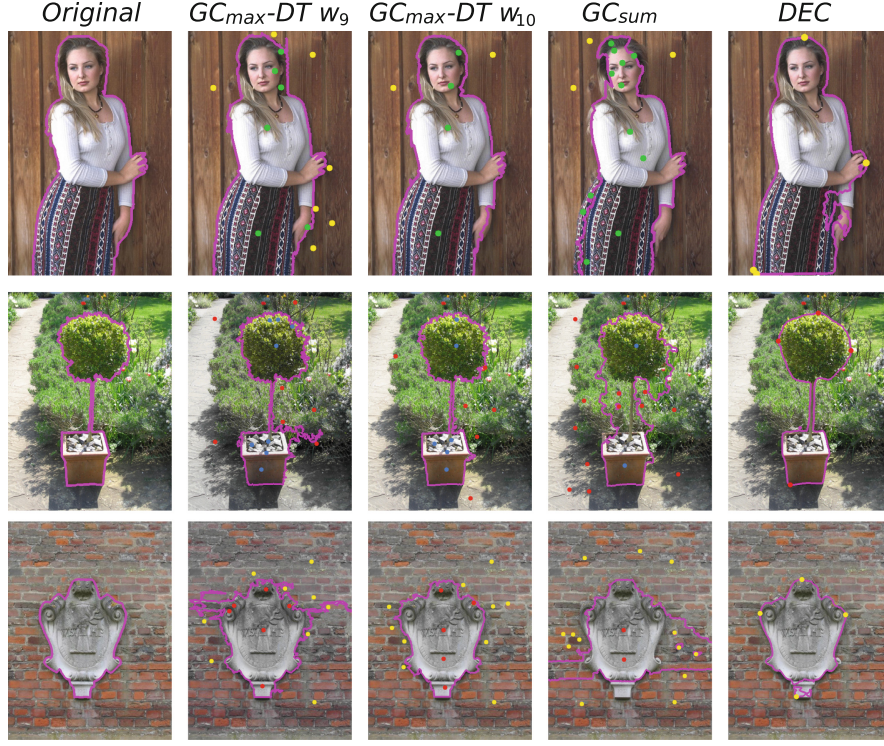


Fig. 4. From left to right, original images with ground-truth border (magenta), segmentation results with their respective methods written above, object (extreme points) and background markers with enhanced size to improve visualization. (Color figure online)

6 Conclusion and Research Directions

We described the **IFT** framework¹, some of its recent results, and applications to delineation, clustering, and classification. We also provided a guideline to extract object information from the target image before and during delineation. Experimental results showed significant improvements in delineation when object information is used, calling attention to a new paradigm—Dynamic Trees (**GC_{max}-DT**) [8].

As research directions, one can directly extend **GC_{max}-DT** to 3D images and multiple objects, learn other object properties from dynamic trees, and integrate it with shape [26] and texture models [24] for automated segmentation. When it fails, methods that transform the result into an optimum-path forest [38] for user correction by differential **IFTs** [14, 19] are needed. Interactive machine learning

¹ Segmentation framework binaries available at <https://github.com/JoOkuma/PyIFT>.

and collaborative segmentation [37] are also important to increase intelligence and reliability in segmentation, creating large annotated image databases for research.

References

1. Achanta, R., et al.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012)
2. Amorim, W.P., Falcão, A.X., Papa, J.P., Carvalho, M.H.: Improving semi-supervised learning through optimum connectivity. *Pattern Recogn.* **60**, 72–85 (2016)
3. Amorim, W.P., Falcão, A.X., Papa, J.P.: Multi-label semi-supervised classification through optimum-path forest. *Inf. Sci.* **465**, 86–104 (2018)
4. Andrade, F., Carrera, E.V.: Supervised evaluation of seed-based interactive image segmentation algorithms. In: *Symposium on Signal Processing, Images and Computer Vision*, pp. 1–7 (2015)
5. Barreto, A.E., Chowdhury, A.S., Falcão, A.X., Miranda, P.A.V.: IFT-SLIC: a general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. In: *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 337–344 (2015)
6. Belém, F., Guimarães, S., Falcão, A.X.: Superpixel segmentation by object-based iterative spanning forest. In: *23rd Iberoamerican Congress on Pattern Recognition* (2018, to appear)
7. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004)
8. Bragantini, J., Martins, S.B., Castelo-Fernandez, C., Falcão, A.X.: Graph-based image segmentation using dynamic trees. In: *23rd Iberoamerican Congress on Pattern Recognition* (2018, to appear)
9. Chen, J., Li, Z., Huang, B.: Linear spectral clustering superpixel. *IEEE Trans. Image Process.* **26**(7), 3317–3330 (2017)
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2018)
11. Ciesielski, K.C., et al.: Joint graph cut and relative fuzzy connectedness image segmentation algorithm. *Med. Image Anal.* **17**(8), 1046–1057 (2013)
12. Ciesielski, K.C., et al.: Path-value functions for which Dijkstra’s algorithm returns optimal mapping. *J. Math. Imaging Vis.* **60**(7), 1–12 (2018)
13. Ciesielski, K., Udupa, J., Falcão, A., Miranda, P.: Fuzzy connectedness image segmentation in graph cut formulation: a linear-time algorithm and a comparative analysis. *J. Math. Imaging Vis.* **44**(3), 375–398 (2012)
14. Condori, M.A.T., Cappabianco, F.A.M., Falcão, A.X., Miranda, P.A.V.: Extending the differential image foresting transform to root-based path-cost functions with application to superpixel segmentation. In: *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 7–14 (2017)
15. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watershed: a unifying graph-based optimization framework. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(7), 1384–1399 (2011)

16. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: minimum spanning forests and the drop of water principle. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(8), 1362–1374 (2009)
17. Falcão, A.X., Bergo, F.P.G.: Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. Med. Imaging* **23**(9), 1100–1108 (2004)
18. Falcão, A.X., Spina, T.V., Martins, S.B., Phellan, R.: Medical image segmentation using object shape models. In: *Invited Lecture in Proceedings of the ECCOMAS Thematic (VipIMAGE)*, pp. 9–15. CRC Press (2015)
19. Falcão, A.X., Stolfi, J., de Lotufo, R.A.: The image foresting transform: theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1), 19–29 (2004)
20. Falcão, A.X., Udupa, J.K., Samarasekera, S., Sharma, S., Hirsch, B.E., Lotufo, R.A.: User-steered image segmentation paradigms: live wire and live lane. *Graph. Models Image Process.* **60**(4), 233–260 (1998)
21. Galvão, F.L., Falcão, A.X., Chowdhury, A.S.: RISF: recursive iterative spanning forest for superpixel segmentation. In: *Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE Xplore (2018, to appear)
22. Gulshan, V., et al.: Geodesic star convexity for interactive image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3129–3136 (2010)
23. Leon, L.M.C., Miranda, P.A.V.: Multi-object segmentation by hierarchical layered oriented image foresting transform. In: *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 79–86 (2017)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
25. Maninis, K.K., et al.: Deep extreme cut: from extreme points to object segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2018)
26. Martins, S.B., Spina, T.V., Yasuda, C., Falcão, A.X.: A multi-object statistical atlas adaptive for deformable registration errors in anomalous medical image segmentation. In: *SPIE on Medical Imaging: Image Processing*, p. 101332G (2017)
27. Miranda, P.A.V., Falcão, A.X., Spina, T.V.: Riverbed: a novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Trans. Image Process.* **21**(6), 3042–3052 (2012)
28. Miranda, P.A.V., Mansilla, L.A.C.: Oriented image foresting transform segmentation by seed competition. *IEEE Trans. Image Process.* **23**(1), 389–398 (2014)
29. Miranda, P.A., Falcão, A.X.: Links between image segmentation based on optimum-path forest and minimum cut in graph. *J. Math. Imaging Vis.* **35**(2), 128–142 (2009)
30. Papa, J., Falcão, A., de Albuquerque, V., Tavares, J.: Efficient supervised optimum-path forest classification for large datasets. *Pattern Recogn.* **45**(1), 512–520 (2012)
31. Phellan, R., Falcão, A.X., Udupa, J.K.: Medical image segmentation via atlases and fuzzy object models: improving efficacy through optimum object search and fewer models. *Med. Phys.* **43**(1), 401–410 (2016)
32. Rauber, P.E., Falcão, A.X., Spina, T.V., Rezende, P.J.: Interactive segmentation by image foresting transform on superpixel graphs. In: *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 131–138 (2013)
33. Rocha, L., Cappabianco, F., Falcão, A.: Data clustering as an optimum-path forest problem with applications in image analysis. *Int. J. Imaging Syst. Technol.* **19**(2), 50–68 (2009)

34. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **23**, 309–314 (2004)
35. Spina, T.V., Miranda, P.A.V., Falcão, A.X.: Intelligent understanding of user interaction in image segmentation. *Int. J. Pattern Recogn. Artif. Intell.* **26**(02), 1265001 (2012)
36. Spina, T.V., Miranda, P.A.V., Falcão, A.: Hybrid approaches for interactive image segmentation using the live markers paradigm. *IEEE Trans. Image Process.* **23**(12), 5756–5769 (2014)
37. Spina, T.V., Stegmaier, J., Falcão, A.X., Meyerowitz, E., Cunha, A.: SEGMENT3D: a web-based application for collaborative segmentation of 3D images used in the shoot apical meristem. In: *International Symposium on Biomedical Imaging (ISBI)*, pp. 391–395 (2018)
38. Tavares, A.C.M., Miranda, P.A.V., Spina, T.V., Falcão, A.X.: A supervoxel-based solution to resume segmentation for interactive correction by differential image-foresting transforms. In: Angulo, J., Velasco-Forero, S., Meyer, F. (eds.) *ISMM 2017. LNCS*, vol. 10225, pp. 107–118. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57240-6_9
39. Udupa, J.K., et al.: Body-wide hierarchical fuzzy modeling, recognition, and delineation of anatomy in medical images. *Med. Image Anal.* **18**(5), 752–771 (2014)
40. Vargas-Muñoz, J.E., et al.: An iterative spanning forest framework for superpixel segmentation. *IEEE Trans. Image Process.* (2019, to appear)