Journal of Mathematical Imaging and Vision
https://doi.org/10.1007/s10851-023-01158-7

Differential Oriented Image Foresting Transform and Its Applications to Support High-level Priors for Object Segmentation

Marcos A. T. Condori[1] · **Paulo A. V. Miranda**[1]

Received: 9 January 2023 / Accepted: 8 July 2023

Abstract
Image foresting transform (IFT) is a graph-based framework to develop image operators based on optimum connectivity between a root set and the remaining nodes, according to a given path-cost function. Oriented image foresting transform (OIFT) was proposed as an extension of some seeded IFT-based segmentation methods to directed graphs, enabling them to support the processing of global object properties, such as connectedness, shape constraints, boundary polarity, and hierarchical constraints, allowing their customization to a given target object. OIFT lies in the intersection of generalized graph cut and general fuzzy connectedness frameworks, inheriting their properties. Its returned segmentation is optimal, with respect to an appropriate graph cut measure, among all segmentations satisfying the given constraints. In this work, we propose differential oriented image foresting transform, which allows multiple OIFT executions for different root sets, making the processing time proportional to the number of modified nodes. Experimental results show considerable efficiency gains over the sequential flow of OIFTs in image segmentation, while maintaining a good treatment of tie zones. We also demonstrate that the differential flow makes it feasible to incorporate the prior knowledge about the maximum allowable size for the segmented object, thus avoiding false positive errors in the segmentation of multi-dimensional images. We also propose an algorithm to efficiently create a hierarchy map that encodes area-constrained OIFT results for all possible thresholds, facilitating the quick selection of the object of interest.

Keywords Oriented image foresting transform · Image segmentation in directed graphs · Generalized graph cut · Differential algorithms

1 Introduction

In graph-based methods, image segmentation can be seen as a graph partition problem between sets of seed pixels. Oriented image foresting transform (OIFT) [1] and oriented relative fuzzy connectedness (ORFC) [2] are extensions to directed weighted graphs of some methods from the generalized graph cut (GGC) framework [3], including fuzzy connectedness [4] and watersheds [4, 5]. OIFT generates an optimal cut in the graph according to an appropriate graph cut measure, while having a lower computational complexity compared to the min-cut/max-flow algorithm [6].[1]

OIFT also belongs to a class of general fuzzy connectedness (GFC) algorithms described in [8], and so, has several good theoretical properties, like robustness for seed placement [9]. Indeed, it has been experimentally verified that OIFT has a good balance between accuracy and robustness when compared to other similar methods in graphs [10]. OIFT has also some relationship with the extension of max-tree algorithm to directed graphs from [11]. However, the segmentation approach described in [11] is more

✉ Marcos A. T. Condori
mtejadac@ime.usp.br

✉ Paulo A. V. Miranda
pmiranda@ime.usp.br

[1] Institute of Mathematics and Statistics, University of São Paulo, R. do Matão 1010, São Paulo, SP 05508–090, Brazil

[1] Concerning the computational complexity, OIFT can be implemented in $\mathcal{O}((M+N)\log N)$ (linearithmic time), where N is the number of vertices in the graph and M is the number of arcs, if a binary heap is used for its priority queue. This computational complexity can be improved to $\mathcal{O}(M+N\times K)$, when the weights are integers in a small interval of size K, by using bucket sorting. The graph cut computational complexity is $\mathcal{O}(\sqrt{M}*N^2) = O(N^{2.5})$ for a sparse graph, which is more than quadratic-time using a push-relabel based on the highest label node selection rule [7].

Published online: 05 August 2023

(a) (b)

Fig. 1 Example of a leaf segmentation: **a** Method from [27] with models trained on a combination of COCO and LVIS datasets with four clicks. **b** OIFT with shape constraints from [28] with two markers

strongly related to the ORFC method from [2]. OIFT has been used in practice, for example, to segment hundreds of three-dimensional images to create an anatomical atlas to regularize the electrical impedance tomography problem [12].

OIFT's energy formulation on digraphs makes it a very versatile method, supporting several high-level priors for object segmentation, including global properties such as connectedness [13, 14], shape constraints [9, 15–17], boundary polarity [1, 18], and hierarchical constraints [19, 20], which allow the customization of the segmentation to a given target object [21]. On the other hand, the integration of machine learning techniques with methods built on the strong formalism of graph partitions has become a very relevant research topic [22–24]. There are different ways to integrate probability maps by convolutional neural networks (CNN) with optimization frameworks in graphs, such as the learning of graph weights and to define seeds or other constraints [25, 26]. In this sense, although this is not the subject of this work, the proposed OIFT extensions can be used at the top of a segmentation pipeline combining with deep learning techniques, guaranteeing the theoretical establishment of the formal properties of the generated objects, increasing the robustness of the obtained results. For example, Fig. 1 presents a comparison of interactive segmentation of natural images of a deep learning based method from [27] with an OIFT equipped with shape constraints for leaf segmentation from [28]. Note that the network would have to be retrained for this particular type of object under consideration in order to generate better results, whereas the high-level constraints supported by OIFT can provide good results.

In interactive region-based segmentation from markers (i.e., set of seeds), the user can add markers to and/or remove markers from previous interactions in order to improve the results. In the context of image foresting transform (IFT) [29], which is based on propagating paths from seeds, instead of starting over the segmentation for each new set of seeds, differential image foresting transform (DIFT) algorithm [30] can be employed to update the segmentation in a differential manner, by correcting only the wrongly labeled parts of the optimum-path forest in time proportional to the size of the modified regions in the image (i.e., in sublinear time). This greatly increases efficiency, which is crucial to obtain interactive response times in the segmentation of large 3D volumes. However, DIFT [30] requires that the path-cost function be *monotonically incremental* (MI), consequently not supporting the OIFT path-cost functions.

More recently, a novel differential IFT algorithm, named *Generalized DIFT* (GDIFT) [31], has been proposed, which extends the original DIFT algorithm to handle connectivity functions with root-based increases (which can be non-monotonically incremental), avoiding segmentation inconsistencies (e.g., disconnected regions) in applications to superpixel segmentation [32–34]. However, there were still no studies on the differential computation for the case of the OIFT path-cost functions. This work aims to close this gap by testing three alternatives for differential oriented image foresting transform (DOIFT). Our experimental results show considerable efficiency gains of the differential flow of DOIFTs over the sequential flow of OIFTs in image segmentation of medical images, while maintaining a good treatment of tie zones for two of the presented solutions. We also demonstrate that the differential flow makes it feasible to incorporate an area constraint in OIFT, as a new high-level prior for object segmentation in multi-dimensional images, which is useful for getting regions of interest in the image with less user interaction. These area/volume constraints can also be used to build a hierarchy of OIFT segmentations by varying the area threshold. This facilitates the quick selection of the object of interest, as each threshold value implies the automatic selection of several extra seeds, simplifying the segmentation process.

DOIFT was first presented in a conference paper [35]. In this extended work, we provide: (i) A more formal and complete description of the DOIFT algorithms; (ii) the usage of DOIFT to incorporate a new high-level prior in the OIFT segmentation, by proposing a new algorithm of OIFT subject to area constraints, including a formal definition of the obtained objects and a formal mathematical proof of the correctness of the algorithm; (iii) the proposal of a new algorithm to efficiently create a hierarchy map that encodes area-constrained OIFT results for all possible thresholds; and (iv) an extensive evaluation of the computed hierarchy, showing the improvements in segmentation with reduced user interaction, including 3D volumes.

The next section presents the required notations and the related graph-based methods. The proposed DOIFT algorithms are then presented in Sect. 3. Section 4 discusses the application of DOIFT for the incorporation of a new high-level constraint limiting the maximum area/volume of the segmented object and the hierarchy of OIFT segmentations by varying the area threshold. This section presents new algorithms that have not been presented in the conference version

of the paper [35]. Finally, the experiments and conclusions are shown in Sects. 5 and 6, respectively.

2 Background

A multi-dimensional and multi-spectral image $\hat{I}$ is a pair $\langle \mathcal{I}, \mathbf{I} \rangle$, where $\mathcal{I} \subset \mathbb{Z}^n$ is the image domain and $\mathbf{I}(t)$ assigns a set of c scalars $I_i(t)$, $i = 1, 2, \ldots, c$, to each pixel $t \in \mathcal{I}$, with c denoting the number of image channels. The subindex i is removed when $c = 1$, representing a grayscale image.

An image can be interpreted as a weighted digraph $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, whose nodes $\mathcal{V}$ are the image pixels in its image domain $\mathcal{I} \subset \mathbb{Z}^n$ or superpixels, and whose arcs are the ordered pixel pairs $\langle s, t \rangle \in \mathcal{A}$ (e.g., 4-neighborhood or 8-neighborhood, in case of 2D images) or are defined by a Region Adjacency Graph (RAG). The digraph G is symmetric if for any of its arcs $\langle s, t \rangle \in \mathcal{A}$, the pair $\langle t, s \rangle$ is also an arc of G. Each arc $\langle s, t \rangle \in \mathcal{A}$ has a weight $\omega(s, t) \geq 0$, such as a dissimilarity measure between pixels s and t (e.g., $\omega(s, t) = \|I(t) - I(s)\|$).

For a given image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, a path of length $\ell \geq 0$ is a sequence $\pi = \langle t_0, t_1, \ldots, t_\ell \rangle$ of adjacent nodes (i.e., $\langle t_i, t_{i+1} \rangle \in \mathcal{A}$, $i = 0, 1, \ldots, \ell - 1$) with no repeated vertices ($t_i \neq t_j$ for $i \neq j$). Other Greek letters, such as τ, can also be used to denote different paths. A path $\pi_t = \langle t_0, t_1, \ldots, t_\ell = t \rangle$ is a path with terminus at a vertex t. When we want to explicitly indicate the origin of the path, the notation $\underset{s \rightsquigarrow t}{\pi} = \langle t_0 = s, t_1, \ldots, t_\ell = t \rangle$ may also be used, where s stands for the origin and t for the destination node. More generally, we can use $\underset{\mathcal{S} \rightsquigarrow t}{\pi} = \langle t_0, t_1, \ldots, t_\ell = t \rangle$ to indicate a path with origin restricted to a set $\mathcal{S}$ (i.e., $t_0 \in \mathcal{S}$). 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 $\langle s, t \rangle$. The set of all possible paths in a graph G with terminus at a node t is denoted as Π_t^G and Π^G denotes all paths in G (i.e., $\Pi^G = \bigcup_{t \in \mathcal{V}} \Pi_t^G$).

A *predecessor map* is a function $P : \mathcal{V} \to \mathcal{V} \cup \{nil\}$ that assigns to each vertex t in $\mathcal{V}$ either some other adjacent node in $\mathcal{V}$, or a distinctive marker nil not in $\mathcal{V}$, in which case t is said to be a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles, i.e., one which takes every node to nil in a finite number of iterations. For any vertex $t \in \mathcal{V}$, 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 $P(t) = s \neq nil$.

2.1 Image Foresting Transform (IFT)

The image foresting transform (IFT) algorithm (Algorithm 1) is a generalization of Dijkstra's algorithm for multiple sources (root sets) and more general connectivity functions [29, 36]. A *connectivity function* $f : \Pi^G \to \mathbb{R}$ computes a value $f(\pi_t)$ for any path π_t, usually based on arc weights. A path π_t is *optimum* if $f(\pi_t) \leq f(\tau_t)$ for any other path $\tau_t \in \Pi_t^G$. By taking to each vertex $t \in \mathcal{V}$ one optimum path with terminus at t, we obtain the optimum-path value $V_{opt}^f(t)$, which is uniquely defined by $V_{opt}^f(t) = \min_{\pi_t \in \Pi_t^G} \{f(\pi_t)\}$. The *image foresting transform* (IFT) [29] takes an image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, and a path-cost function f; and assigns one optimum path to every vertex $t \in \mathcal{V}$ such that an *optimum-path forest* P is obtained, i.e., a spanning forest where all paths π_t^P for $t \in \mathcal{V}$ are optimum. However, f must satisfy the conditions indicated in [36], otherwise, the paths π_t^P of the returned spanning forest may not be optimum.

For the sake of simplicity, in this section we will present a particular version of the IFT algorithm, focused on seed-based segmentation problems, such that we constrain the search to paths that start in a given set $\mathcal{S} \subseteq \mathcal{V}$ of seed nodes. We can model this constraint by considering a seeded path-cost function $f^{\mathcal{S}}$, such that $f^{\mathcal{S}}(\underset{s \rightsquigarrow t}{\pi}) = +\infty$ when $s \notin \mathcal{S}$. The seeded path-cost functions are usually defined by an initialization rule for trivial paths and an extension rule for non-trivial paths. For example, consider the case of the path-cost function $f_{\max}^{\mathcal{S}}$:

$$f_{\max}^{\mathcal{S}}(\langle t \rangle) = \begin{cases} -1 & \text{if } t \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases}$$

$$f_{\max}^{\mathcal{S}}(\pi_s \cdot \langle s, t \rangle) = \max\{f_{\max}^{\mathcal{S}}(\pi_s), \omega(s, t)\}. \quad (1)$$

In the context of multiple object segmentation, let $\mathcal{L} = \{0, \ldots, k\}$ denote a label set, where k is the number of objects to be segmented. IFT can be used to divide an image into $k+1$ partitions $\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_k$ ($\bigcup_{l \in \mathcal{L}} \mathcal{O}_l = \mathcal{V}$ and $\mathcal{O}_i \cap \mathcal{O}_j = \emptyset$ for $i, j \in \mathcal{L}$ with $i \neq j$), by assigning a label $L(t) \in \mathcal{L}$ to each vertex t, such that $L(t) = 0$ is used to indicate the background and $L(t) = l > 0$ is used to represent the corresponding object $\mathcal{O}_l = \{t \in \mathcal{V} \mid L(t) = l\}$. In segmentation problems, a partial labeling of the image $\lambda : \mathcal{S} \to \mathcal{L}$ is usually provided, defining labels for a subset of seeds $\mathcal{S} \subset \mathcal{V}$, such that $\mathcal{S} = \mathcal{S}_0 \cup \ldots \cup \mathcal{S}_k$ and $\lambda(t) = l$ if and only if $t \in \mathcal{S}_l$. During the seeded IFT computation, this initial labeling of the seeds in $\mathcal{S}$ is propagated to all other unlabeled nodes in $\mathcal{V} \setminus \mathcal{S}$, such that each vertex $t \in \mathcal{V} \setminus \mathcal{S}$ receives the same label as the origin of the computed path π_t^P, that is, $L(t) = l$ if $\pi_t^P = \langle r, \ldots, t \rangle$ and $r \in \mathcal{S}_l$.

In Algorithm 1, the root map $R : \mathcal{V} \to \mathcal{V}$ stores the origin of the paths ending at each node (i.e., $R(t) = r$ if $\pi_t^P = \langle r, \ldots, t \rangle$) and the path-cost map V converges to V_{opt}^f, when $f^{\mathcal{S}}$ satisfies the conditions indicated in [36].

Algorithm 1 – SEEDED IFT ALGORITHM

INPUT: Image graph $\langle \mathcal{V}, \mathcal{A}, \omega \rangle$, a seeded path-cost function $f^{\mathcal{S}}$, seed set $\mathcal{S}$ and an initial labeling function $\lambda : \mathcal{S} \to \mathcal{L}$.

OUTPUT: The label map $L : \mathcal{V} \to \mathcal{L}$, root map $R : \mathcal{V} \to \mathcal{V}$, path-cost map $V : \mathcal{V} \to \mathbb{R}$ and the spanning forest $P : \mathcal{V} \to \mathcal{V} \cup \{nil\}$.

AUXILIARY: Priority queue Q, variable tmp and an array of status $S : \mathcal{V} \to \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.

1. **For each** $t \in \mathcal{V}$, **do**
2. Set $S(t) \leftarrow 0$ and $R(t) \leftarrow t$.
3. Set $P(t) \leftarrow nil$ and $V(t) \leftarrow +\infty$.
4. **For each** $t \in \mathcal{S}$, **do**
5. Set $L(t) \leftarrow \lambda(t)$ and $V(t) \leftarrow f^{\mathcal{S}}(\langle t \rangle)$.
6. Insert t in Q.
7. **While** $Q \neq \emptyset$, **do**
8. Remove s from Q such that $V(s) = \min_{t \in Q}\{V(t)\}$.
9. Set $S(s) \leftarrow 1$.
10. **For each** node t such that $\langle s, t \rangle \in \mathcal{A}$, **do**
11. **If** $S(t) \neq 1$, **then**
12. Compute $tmp \leftarrow f^{\mathcal{S}}(\pi_s^P \cdot \langle s, t \rangle)$.
13. **If** $tmp < V(t)$, **then**
14. **If** $t \in Q$, **then** remove t from Q.
15. Set $P(t) \leftarrow s$ and $V(t) \leftarrow tmp$.
16. Set $R(t) \leftarrow R(s)$ and $L(t) \leftarrow L(s)$.
17. Insert t in Q.

2.2 Oriented Image Foresting Transform (OIFT)

In a symmetric digraph, each object $\mathcal{O} \subset \mathcal{V}$ has two possible cuts: An outer cut composed of the arcs in the set $\mathcal{C}_{\text{out}}(\mathcal{O})$. An inner cut composed of the arcs in the set $\mathcal{C}_{in}(\mathcal{O})$.

$$\mathcal{C}_{\text{out}}(\mathcal{O}) = \{\langle s, t \rangle \in \mathcal{A} \mid s \in \mathcal{O} \text{ and } t \notin \mathcal{O}\} \tag{2}$$

$$\mathcal{C}_{in}(\mathcal{O}) = \{\langle s, t \rangle \in \mathcal{A} \mid s \notin \mathcal{O} \text{ and } t \in \mathcal{O}\} \tag{3}$$

In this work, OIFT will be presented based on the optimality of the outer-cut energy,[2] but note that the inner cut is a dual case that can be obtained by inverting the seed labels in $\mathcal{L} = \{0, 1\}$, that is $\mathcal{C}_{\text{out}}(\mathcal{O}) = \mathcal{C}_{in}(\mathcal{V} \backslash \mathcal{O})$.

In its first version [1], OIFT was built on the IFT framework for the binary segmentation problem with $\mathcal{L} = \{0, 1\}$ by considering the following seeded path-cost function in a symmetric digraph with integer weights (i.e., $\omega \colon \mathcal{A} \to \mathbb{Z}$):

$$f_1^{\sigma}(\langle t \rangle) = \begin{cases} -1 & \text{if } t \in \mathcal{S}_1 \cup \mathcal{S}_0 \\ +\infty & \text{otherwise} \end{cases}$$

$$f_1^{\sigma}(\underset{r \rightsquigarrow s}{\pi} \cdot \langle s, t \rangle) = \begin{cases} \text{If } r \in \mathcal{S}_1\text{:} \\ \max\{f_1^{\sigma}(\underset{r \rightsquigarrow s}{\pi}), 2 \times \omega(s, t) + 1\} \\ \\ \text{Else if } r \in \mathcal{S}_0\text{:} \\ \max\{f_1^{\sigma}(\underset{r \rightsquigarrow s}{\pi}), 2 \times \omega(t, s)\} \end{cases} \tag{4}$$

Later, a second version [18], with a better handling of energy ties and real weights $\omega \colon \mathcal{A} \to \mathbb{R}$, was proposed based on the following alternative seeded path-cost function, designed to be used with FIFO tie-breaking policy for the priority queue Q in Algorithm 1:

$$f_2^{\sigma}(\langle t \rangle) = f_1^{\sigma}(\langle t \rangle)$$

$$f_2^{\sigma}(\underset{r \rightsquigarrow s}{\pi} \cdot \langle s, t \rangle) = \begin{cases} \omega(s, t) & \text{if } r \in \mathcal{S}_1 \\ \omega(t, s) & \text{else if } r \in \mathcal{S}_0 \end{cases} \tag{5}$$

The segmented object $\mathcal{O}_1 = \{t \in \mathcal{V} \mid L(t) = 1\}$ by OIFT is defined from the label map L computed by Algorithm 1 with f_2^{σ} (or f_1^{σ}). See Fig. 2a–c.

The path-cost functions f_1^{σ} and f_2^{σ} are non-monotonically incremental connectivity functions, as described in [1, 18]. The optimality of $\mathcal{O}_1$ by OIFT is supported by an energy criterion of cut in graphs involving arcs from object to background nodes in $\mathcal{C}_{out}(\mathcal{O}_1)$ (Fig. 2d–e), according to Theorem 1 from [1, 18].

$$E(\mathcal{O}) = \min_{\langle s,t \rangle \in \mathcal{C}_{out}(\mathcal{O})} \omega(s, t) \tag{6}$$

Theorem 1 *(Outer-cut optimality by OIFT) For two given sets of seeds $\mathcal{S}_1$ and $\mathcal{S}_0$, let $\mathcal{U}(\mathcal{S}_1, \mathcal{S}_0) = \{\mathcal{O} \subseteq \mathcal{V} \mid \mathcal{S}_1 \subseteq \mathcal{O} \subseteq \mathcal{V} \backslash \mathcal{S}_0\}$ denote the universe of all possible objects satisfying the seed constraints. Any label map L computed by Algorithm 1 for function f_1^{σ} (or f_2^{σ}) defines a segmented object $\mathcal{O}_1 = \{t \in \mathcal{V} \mid L(t) = 1\}$ that maximizes E (Eq. 6) among all possible segmentation results in $\mathcal{U}(\mathcal{S}_1, \mathcal{S}_0)$. That is, $E(\mathcal{O}_1) = \max_{\mathcal{O} \in \mathcal{U}(\mathcal{S}_1, \mathcal{S}_0)} E(\mathcal{O})$.*

2.3 Differential Image Foresting Transform (DIFT)

Let a sequence of IFTs be represented as $\langle IFT_{(\mathcal{S}^1)}, IFT_{(\mathcal{S}^2)}, \ldots, IFT_{(\mathcal{S}^n)} \rangle$, where n is the total number of IFT executions on the image. At each execution, the seed set $\mathcal{S}^i$ is modified by adding and/or removing seeds to obtain a new set $\mathcal{S}^{i+1}$. We define a scene $\mathcal{G}^i$ as the set of maps $\mathcal{G}^i = \{L^i, R^i, V^i, P^i\}$, resulting from the *ith* iteration in a sequence of IFTs.

The DIFT algorithm [30, 31] allows to efficiently compute a scene $\mathcal{G}^i$ from the previous scene $\mathcal{G}^{i-1}$, a set $\Delta^+_{\mathcal{S}^i} = \mathcal{S}^i \backslash \mathcal{S}^{i-1}$ of new seeds for addition, and a set $\Delta^-_{\mathcal{S}^i} = \mathcal{S}^{i-1} \setminus \mathcal{S}^i$ of seeds marked for removal. In the execution flow by DIFT, after the first execution of $IFT_{(\mathcal{S}^1)}$, we

[2] As will be shown, the ♂ sign is used for the OIFT path-cost functions to indicate that they are related to the optimality of the outer-cut boundary (see Theorem 1), since it resembles an arc pointing to the exterior of an object.

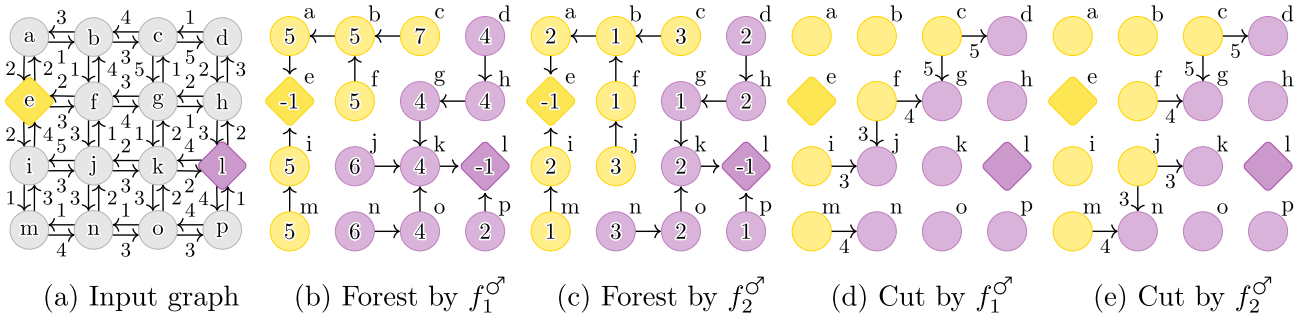


Fig. 2 **a** Input graph with marked seeds $\mathcal{S}_1 = \{e\}$ and $\mathcal{S}_0 = \{l\}$. **b–c** The computed forests by OIFT with $f_1^{\sqrt{}}$ and $f_2^{\sqrt{}}$, respectively. The values within the nodes indicate the costs of the paths and the arrows point to the predecessor of each node. **d–e** The corresponding outer cuts of the objects generated by OIFT with $f_1^{\sqrt{}}$ and $f_2^{\sqrt{}}$, respectively, with both having an energy value of 3 by Eq. 6

have that the scenes $\mathcal{G}^i$ for $i \geq 2$ are calculated based on the scene $\mathcal{G}^{i-1}$, taking advantage of the trees that were computed in the previous iteration, thus reducing the processing time. Hence, we have the following differential flow: $\langle IFT_{(\mathcal{S}^1)}, DIFT_{(\Delta^+_{\mathcal{S}^2},\Delta^-_{\mathcal{S}^2},\mathcal{G}^1)}, DIFT_{(\Delta^+_{\mathcal{S}^3},\Delta^-_{\mathcal{S}^3},\mathcal{G}^2)}, \ldots, DIFT_{(\Delta^+_{\mathcal{S}^n},\Delta^-_{\mathcal{S}^n},\mathcal{G}^{n-1})}\rangle$.

3 Differential OIFT (DOIFT)

Figure 3 shows that the generalized DIFT (GDIFT) algorithm [31] with $f_2^{\sqrt{}}$, to differentially compute the sequence $\langle IFT_{(\mathcal{S}^1)}, IFT_{(\mathcal{S}^2)}\rangle$, where $\mathcal{S}^1 = \mathcal{S}^1_1 \cup \mathcal{S}^1_0 = \{a\} \cup \{i, l\}$ and $\mathcal{S}^2 = \mathcal{S}^2_1 \cup \mathcal{S}^2_0 = \{a\} \cup \{i\}$, may generate a result not predicted by $IFT_{(\mathcal{S}^2)}$ via Algorithm 1. The problem occurs because nodes b and g are initially processed in a given order during the first run of the IFT (Fig. 3b), but later become frontier nodes, i.e., neighboring nodes of removed trees/subtrees (Fig. 3c), that can be reprocessed in a different order than the original (Fig. 3d). Due to the strictly minor inequality of Line 13 of Algorithm 1, in the case of ties in offered costs, we have that the node that first sees its contested neighbor will win the dispute. Therefore, multiple processing orders affect the conquest of neighboring nodes (such as nodes c and f in Fig. 3).

The DIFT algorithms [30, 31] do not attempt to address this issue, as they assume that the usage of the "$\leq$" comparison on Line 13 of Algorithm 1 would also be perfectly valid. However, in the case of functions such as $f_2^{\sqrt{}}$, in which the cost along the path is not a non-decreasing function, these problems in the processing order of frontier nodes are severely aggravated and can generate solutions that would never be obtained in the sequential flow. To resolve these issues, it would be necessary to explicitly store the processing order of the nodes, to ensure that later, the frontier nodes would be reprocessed in the same previous order. However, in addition to spending more memory, it would be complex to ensure the consistency in maintaining this new map of order over several iterations.

In order to address these issues without compromising the execution time of the algorithms, we chose to develop solutions for the differential OIFT focused only on the issue of generating segmentation labels that are consistent with the sequential flow labeling (consequently ensuring an optimal cut as in Theorem 1), even though the predecessor map might be different. Note that this decision does not adversely affect any of the OIFT applications to date, because minor topology details of the resulting forest are irrelevant to the segmentation task.

The first proposed solution, denoted as $DOIFT_1$, is simply to consider the usage of the generalized DIFT (GDIFT) algorithm from [31] with the $f_1^{\sqrt{}}$ path-cost function. Note that $f_1^{\sqrt{}}$ is a function with non-decreasing costs along the path, with cost variations depending only on the root label and the arc weights $\omega(s,t)$ and $\omega(t,s)$, which perfectly fits the conditions required in [31]. Note that problems like the one reported in Fig. 3 do not occur with $f_1^{\sqrt{}}$, since there are no cost ties between object and background in this formulation, as they are treated as odd and even numbers, respectively, and the background is always favored. Figure 4a-c shows an example of $DOIFT_1$. The disadvantage of this solution is the unbalanced treatment in favor of the background in the case of ties, when there are multiple solutions with the same optimal cutting energy by Eq. 6 (Fig. 4c).

The second proposed solution, denoted as $DOIFT_2$, is to use Algorithm 2, which considers for each path π_t a lexicographical path-cost function with two components $\langle F_2^{\sqrt{}}(\pi_t), T(\pi_t)\rangle$, where $F_2^{\sqrt{}}(\pi = \langle t_0, \ldots, t_n\rangle) = \max_{i=0,1,\ldots,n}\{f_2^{\sqrt{}}(\langle t_0, \ldots, t_i\rangle)\}$ and $T(\pi_t)$ is related to the number of maximum valued arcs crossed along the path, aiming at a better handling of tie zones, but we use odd numbers in $T(\pi_t)$ for paths from the background seeds and even numbers for the object, so that there are no ties in the second

component between object and background:

$$T(\langle t\rangle) = \begin{cases} \lambda(t)+1 & \text{if } t \in \mathcal{S}_1 \cup \mathcal{S}_0 \\ +\infty & \text{otherwise} \end{cases}$$

$$T(\pi_t = \underset{r\rightsquigarrow s}{\pi} \cdot \langle s,t\rangle) = \begin{cases} \text{If } r \notin \mathcal{S}_1 \cup \mathcal{S}_0\text{:} \\ +\infty \\ \\ \text{Else if } f_2^{\vec{\sigma}}(\pi_t) \neq F_2^{\vec{\sigma}}(\pi_t)\text{:} \\ T(\underset{r\rightsquigarrow s}{\pi}) \\ \\ \text{Else if } F_2^{\vec{\sigma}}(\underset{r\rightsquigarrow s}{\pi}) = F_2^{\vec{\sigma}}(\pi_t)\text{:} \\ T(\underset{r\rightsquigarrow s}{\pi})+2 \\ \\ \text{Else:} \\ \lambda(r)+1 \end{cases} \quad (7)$$

Figure 4d–e illustrates an example of $DOIFT_2$.

Algorithm 2 – ALGORITHM $DOIFT_2$

INPUT:	Image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega\rangle$, the set Δ_S^+ of seeds for addition, set Δ_S^- of seeds for removal, the maps L, V and P initialized with the result from the previous OIFT/DOIFT execution, and an initial labeling function $\lambda : \Delta_S^+ \to \{0,1\}$ for the new seeds. We consider $V(t) = \langle V_1(t), V_2(t)\rangle$ as we work with lexicographical costs.
OUTPUT:	The updated maps L, V and P.
AUXILIARY:	Priority queue Q, and variables tmp_1 and tmp_2.

1. *Set* $Q \leftarrow \emptyset$.
2. **If** $\Delta_S^- \neq \emptyset$, **then**
3. └ *DOIFT-RemoveSubTrees*($G, L, V, P, Q, \Delta_S^-$)
4. **For each** $s \in \Delta_S^+$, **do**
5. │ *Set* $L(s) \leftarrow \lambda(s)$ *and* $P(s) \leftarrow nil$.
6. │ *Set* $V(s) \leftarrow \langle -1, L(s)+1\rangle$.
7. └ **If** $s \notin Q$, **then** *insert* s *in* Q.
8. **While** $Q \neq \emptyset$, **do**
9. │ *Remove* s *from* Q *such that* $V(s) \overset{\text{lex}}{\leq} V(r)$ *for all* $r \in Q$.
10. │ **For each** *node* t *such that* $\langle s,t\rangle \in \mathcal{A}$, **do**
11. │ │ *Compute* $tmp_1 \leftarrow F_2^{\vec{\sigma}}(\pi_s^P \cdot \langle s,t\rangle)$.
12. │ │ **If** $tmp_1 \neq f_2^{\vec{\sigma}}(\pi_s^P \cdot \langle s,t\rangle)$, **then**
13. │ │ └ *Set* $tmp_2 \leftarrow V_2(s)$.
14. │ │ **Else If** $V_1(s) = tmp_1$, **then**
15. │ │ └ *Set* $tmp_2 \leftarrow V_2(s)+2$.
16. │ │ **Else** , **then**
17. │ │ └ *Set* $tmp_2 \leftarrow L(s)+1$.
18. │ │ **If** $\langle tmp_1, tmp_2\rangle \overset{\text{lex}}{<} V(t)$, **then**
19. │ │ │ *Set* $L(t) \leftarrow L(s)$ *and* $P(t) \leftarrow s$.
20. │ │ │ *Set* $V(t) \leftarrow \langle tmp_1, tmp_2\rangle$.
21. │ │ └ **If** $t \notin Q$, **then** *insert* t *in* Q.
22. │ │ **Else If** $s = P(t)$, **then**
23. │ │ │ **If** $tmp_1 \neq V_1(t)$ *or* $tmp_2 > V_2(t)$ *or* $L(t) \neq L(s)$, **then**
24. │ │ │ │ *DOIFT-RemoveSubTrees*($G, L, V, P, Q, \{t\}$)
25. └ └ └ └ ***Break; #GOTO LINE 9***

Procedure DOIFT-RemoveSubTrees in Algorithm 3 releases the entire subtrees, converting its nodes to trivial trees of infinite cost, and transforms all of its neighboring nodes into frontier vertices, inserting them in Q, assuming that the graph is symmetric. It plays the role of both DIFT-RemoveSubTree and DIFT-TreeRemoval from [31], but has been modified to not rely on the use of a root map to save memory.

Algorithm 3 – PROCEDURE DOIFT- REMOVESUBTREES

INPUT:	Image graph G, the maps L, V and P, the priority queue Q, and a set $\mathcal{R}$ of roots of the subtrees to be removed.
OUTPUT:	The updated maps L, V and P, and the updated priority queue Q passed by reference.
AUXILIARY:	Queue J and a set $\mathcal{F}$.

Set $J \leftarrow \emptyset$, $\mathcal{F} \leftarrow \emptyset$.
For each $t \in \mathcal{R}$, **do**
│ **If** $t \in Q$, **then** *remove* t *from* Q.
│ *Set* $V(t) \leftarrow \langle\infty,\infty\rangle$, $P(t) \leftarrow nil$.
└ *Insert* t *in* J.
While $J \neq \emptyset$, **do**
│ *Remove* s *from* J.
│ **For each** *node* t *such that* $\langle s,t\rangle \in \mathcal{A}$, **do**
│ │ **If** $s = P(t)$, **then**
│ │ │ *Insert* t *in* J.
│ │ │ **If** $t \in Q$, **then** *remove* t *from* Q.
│ │ └ *Set* $V(t) \leftarrow \langle\infty,\infty\rangle$, $P(t) \leftarrow nil$.
│ │ **Else If** $V(t) \neq \langle\infty,\infty\rangle$ *and* $t \notin Q$, **then**
└ └ └ *Insert* t *in* $\mathcal{F}$.
While $\mathcal{F} \neq \emptyset$, **do**
│ *Remove* t *from* $\mathcal{F}$.
│ **If** $V(t) \neq \langle\infty,\infty\rangle$ *and* $t \notin Q$, **then**
└ └ *Insert* t *in* Q.

Other differences of Algorithm 2 in relation to GDIFT [31], are the absence of the state map used in [31], which proved to be unnecessary for functions with non-decreasing costs along the paths, as for the lexicographical cost $\langle F_2^{\vec{\sigma}}(\pi_t), T(\pi_t)\rangle$, and modifications to avoid using the root map to save memory. Another difference is the inclusion of Line 25 in Algorithm 2, to immediately break the innermost loop, thus avoiding the repeated processing of part of the neighborhood.

The third proposed version of DOIFT is a variant of the second, which was designed to better mimic the behavior of the sequential flow with $f_2^{\vec{\sigma}}$ by Algorithm 1. More specifically, it was modified so that disputed nodes with the same cost (from the perspective of $f_2^{\vec{\sigma}}$) are given to the first processed neighbor, so as to respect Proposition 1, that will be defined next.

For any function $f(\pi)$, let $F(\pi)$ denote the maximum cost along the path:

$$F(\pi = \langle t_0, \ldots, t_n\rangle) = \max_{i=0,1,\ldots,n} \{f(\langle t_0, \ldots, t_i\rangle)\} \quad (8)$$

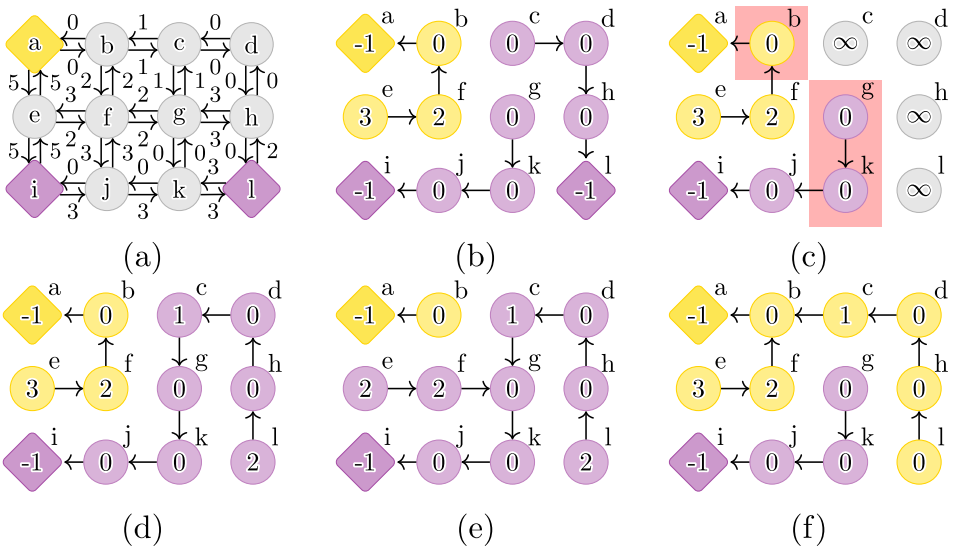


Fig. 3 **a** Input graph with marked seeds $\mathcal{S}_1^1 = \{a\}$ and $\mathcal{S}_0^1 = \{i, l\}$. **b** Initial computed forest by OIFT with $f_2^{\sphericalangle}$, assuming node b was processed first than node g. The values within the nodes indicate the costs of the paths and the arrows point to the predecessor of each node. **c** The tree of node l is marked for removal and its nodes are made available for a new dispute between the frontier nodes of neighboring trees (marked with a pink background). **d** A possible result of the differential flow, where the frontier node g was processed before b, thus gaining c, but leading to a result that cannot be generated by the sequential flow via Algorithm 1. **e–f** The two possible outcomes of sequential flow for $f_2^{\sphericalangle}$ with $\mathcal{S}_1^2 = \{a\}$ and $\mathcal{S}_0^2 = \{i\}$

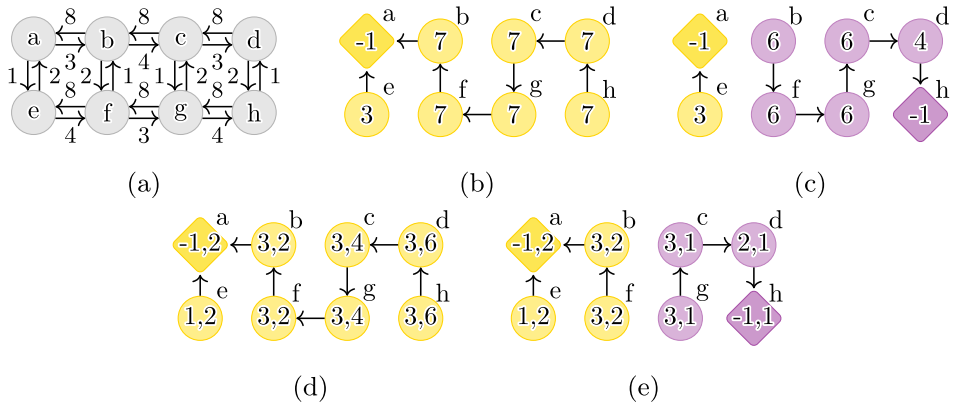


Fig. 4 **a** Input graph. **b** Initial forest by OIFT with $f_1^{\sphericalangle}$ for $\mathcal{S}^1 = \mathcal{S}_1^1 = \{a\}$. The values within nodes reflect the costs of $f_1^{\sphericalangle}$ and the arrows point to the predecessor of each node. **c** The updated result by the first proposed differential solution for DOIFT, using GDIFT with $f_1^{\sphericalangle}$, as a new background seed h is inserted, so that $\mathcal{S}^2 = \mathcal{S}_1^2 \cup \mathcal{S}_0^2 = \{a\} \cup \{h\}$. **d–e** Alternative differential flow through the usage of Algorithm 2, leading to a better balanced solution. The pairs of values inside the nodes indicate the lexicographical costs $\langle F_2^{\sphericalangle}(\pi_t^P), T(\pi_t^P)\rangle$. **d** Initial forest by Algorithm 2 for $\mathcal{S}^1 = \{a\}$. **(e)** The updated result by Algorithm 2 for $\mathcal{S}^2 = \{a, h\}$. Note that the cost offered to node f by $\pi_g^P \cdot \langle g, f\rangle$ would be $\langle 3, 3\rangle$, which is worse than $\langle 3, 2\rangle$ offered by seed a

Consider the following lemma:

Lemma 1 *Let P be a predecessor map computed by Algorithm 1 for a seeded path-cost function f. For any two paths $\delta_t^P = \langle t_0, t_1, \ldots, t_n = t\rangle$ and $\tau_s^P = \langle s_0, s_1, \ldots, s_m = s\rangle$, defined by P, if $F(\delta_t^P) < F(\tau_s^P)$, then we have that node t was removed before s from $\mathcal{Q}$ on Line 8 of Algorithm 1.*

Proof Let s_k be a node in τ_s^P, such that $f(\langle s_0, \ldots, s_k\rangle) = F(\tau_s^P)$. From Eq. 8, we have that $f(\langle t_0, \ldots, t_i\rangle) \leq F(\delta_t^P)$, $i = 0, 1, \ldots, n$. From the assumptions of Lemma 1, we may conclude that $F(\delta_t^P) < f(\langle s_0, \ldots, s_k\rangle)$. Thus, $f(\langle t_0, \ldots, t_i\rangle) < f(\langle s_0, \ldots, s_k\rangle)$, $i = 0, 1, \ldots, n$.

From the dynamic of execution of Algorithm 1, we know that paths δ_t^P and τ_s^P stored in the map P are gradually computed by the removal from $\mathcal{Q}$ of nodes with minimum cost (Line 8). After s_k gets inserted in $\mathcal{Q}$ with cost $V(s_k) = f(\langle s_0, \ldots, s_k\rangle)$, it won't be removed from $\mathcal{Q}$ before

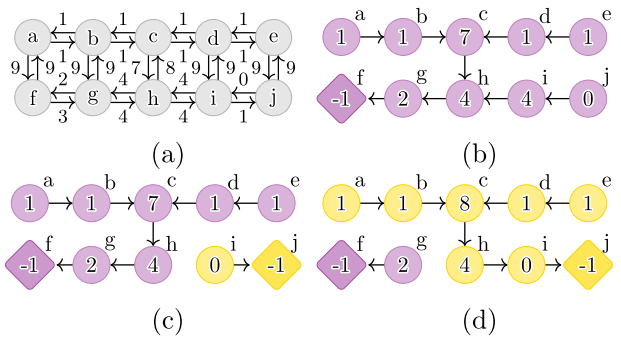


Fig. 5 **a** Input graph. **b** Initial forest by OIFT with $f_2^{\sum}$ for $\mathcal{S}^1 = \mathcal{S}_0^1 = \{f\}$. **c** The updated result by Algorithm 2, as a new object seed j is inserted, so that $\mathcal{S}^2 = \{f, j\}$. The values within nodes reflect the costs of $f_2^{\sum}$. **d** The correct result by Proposition 1 with respect to $f_2^{\sum}$. Note that node i leaves $\mathcal{Q}$ prior to g, since $F_2^{\sum}(\pi_i^P) = 0 < F_2^{\sum}(\pi_g^P) = 2$. So, π_h^P cannot be $\pi_g^P \cdot \langle g, h\rangle$ according to Proposition 1

all nodes t_i, $i = 0, 1, \ldots, n$, are consecutively processed in $\mathcal{Q}$, with lower costs $V(t_i) = f(\langle t_0, \ldots, t_i\rangle)$. Therefore, we have that $t = t_n$ is removed prior to s from $\mathcal{Q}$. □

From Lemma 1, we can also conclude the following proposition:

Proposition 1 *Let P be a predecessor map computed by Algorithm 1 for a seeded path-cost function f. For any two paths δ_s^P and $\tau_{s'}^P$, $s \neq s'$, defined in P, if $F(\tau_{s'}^P) < F(\delta_s^P)$ and $f(\delta_s^P \cdot \langle s, t\rangle) = f(\tau_{s'}^P \cdot \langle s', t\rangle)$, then we have that $\pi_t^P \neq \delta_s^P \cdot \langle s, t\rangle$.*

Proof Algorithm 1 will assign t to the first optimum path that reaches it, because of the strict inequality in Line 13. According to Lemma 1, we have that s' leaves $\mathcal{Q}$ before s. Consequently, the path $\tau_{s'}^P \cdot \langle s', t\rangle$ is evaluated before $\delta_s^P \cdot \langle s, t\rangle$, offering the same cost (i.e., $f(\delta_s^P \cdot \langle s, t\rangle) = f(\tau_{s'}^P \cdot \langle s', t\rangle)$). Therefore, we have that π_t^P cannot be $\delta_s^P \cdot \langle s, t\rangle$. □

Figure 5 discusses the consequences of Proposition 1 in the differential execution of OIFT. Note that the predecessor map P computed by Algorithm 2 does not satisfy Proposition 1 with respect to function $f_2^{\sum}$. To correct this issue, the condition of Line 18 of Algorithm 2 must be changed to a much more complex condition: $tmp_1 < V_1(t)$or $(tmp_1 = V_1(t)$ and $((tmp_2 < V_2(t)$ and not$H_2)$ or $H_1))$ where X, H_1 and H_2 are Boolean variables defined as:

$$X \leftarrow V_1(t) = f_2^{\sum}(\pi_s^P \cdot \langle s, t\rangle) > V_1(s) \text{ and } V_1(t) > V_1(P(t))$$

$$H_1 \leftarrow P(t) \neq nil \text{ and } X \text{ and } V(s) \overset{\text{lex}}{<} V(P(t))$$

$$H_2 \leftarrow P(t) \neq nil \text{ and } X \text{ and } V(s) \overset{\text{lex}}{>} V(P(t))$$

With these modifications, we have the third version of the DOIFT algorithm, denoted as $DOIFT_3$.

4 OIFT with area/Volume Constraint Aided by DOIFT

The *boundary* of an object $\mathcal{O}$ is defined as

$$\text{bd}(\mathcal{O}) = \{s \in \mathcal{O} \mid \exists \langle s, t\rangle \in \mathcal{A} \text{ such that } t \notin \mathcal{O}\}.$$

Let $G[\mathcal{O}]$ be the subgraph of G induced by $\mathcal{O} \subseteq \mathcal{V}$ and $\underset{\mathcal{S}\rightsquigarrow t}{\pi}[\mathcal{O}]$ denote a path from a node in $\mathcal{S}$ to t in the subgraph $G[\mathcal{O}]$. Consider the following universe of solutions:

$$\widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0) = \{\mathcal{O} \in \mathcal{U}(\mathcal{S}_1, \mathcal{S}_0) \mid \forall t \in \overline{\mathcal{O}}\ \exists\ \underset{\mathcal{S}_0\rightsquigarrow t}{\pi}[\overline{\mathcal{O}}]\},$$

where $\overline{\mathcal{O}} = \mathcal{V} \setminus \mathcal{O}$. This set defines any solution that satisfies the seed constraints with its background being connected to the seeds in $\mathcal{S}_0$. In this section, we show how to compute an optimum segmentation subject to area constraints by OIFT according to the following theorem.

Theorem 2 *For two given sets of seeds $\mathcal{S}_1$ and $\mathcal{S}_0$, and an area threshold T (such that $T \geq |\mathcal{S}_1|$), any label map L computed by Algorithm 4 defines a segmented object $\mathcal{O}_1 = \{t \in \mathcal{V} \mid L(t) = 1\}$ that maximizes E (Eq. 6) among all possible segmentations results in $\widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0) = \{\mathcal{O} \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0) \mid |\mathcal{O}| \leq T\}$. That is, $E(\mathcal{O}_1) = \max_{\mathcal{O} \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)} E(\mathcal{O})$ and $\mathcal{O}_1 \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)$.*

Algorithm 4 – AREA- CONSTRAINED OIFT

INPUT:	Image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega\rangle$, seed sets $\mathcal{S}_1$ and $\mathcal{S}_0$, initial labeling function $\lambda : \mathcal{S} \to \{0, 1\}$, and area threshold T.
OUTPUT:	The label map $L : \mathcal{V} \to \{0, 1\}$.
AUXILIARY:	Energy map $E : \mathcal{V} \to \mathbb{R}$, path-cost map $V : \mathcal{V} \to \mathbb{R}^2$, and the spanning forest $P : \mathcal{V} \to \mathcal{V} \cup \{nil\}$.

1. $(L, R, E, P) \leftarrow IFT(G, f_{\max}^{\mathcal{S}_1}, \mathcal{S}_1, \lambda)$.
2. **For each** $t \in \mathcal{V}$, **do**
3. └ *Set* $P(t) \leftarrow nil$ *and* $V(t) \leftarrow \langle \infty, \infty\rangle$.
4. *Set* $\Delta_{\mathcal{S}}^+ \leftarrow \mathcal{S}_1 \cup \mathcal{S}_0$ *and* $\mathcal{S}^1 \leftarrow \mathcal{S}_1 \cup \mathcal{S}_0$.
5. *Set* $\Delta_{\mathcal{S}}^- \leftarrow \emptyset$.
6. *Set* $i \leftarrow 1$ *and* $area \leftarrow \infty$.
7. **While** $area > T$, **do**
8. | $DOIFT(G, \Delta_{\mathcal{S}}^+, \Delta_{\mathcal{S}}^-, L^i, V, P, \lambda)$.
9. | *Set* $\mathcal{O}_1^i = \{t \in \mathcal{V} \mid L^i(t) = 1\}$.
10. | *Set* $s \leftarrow \arg\max_{t \in \text{bd}(\mathcal{O}_1^i)} E(t)$.
11. | *Set* $\Delta_{\mathcal{S}}^+ \leftarrow \{s\}$ *and* $\mathcal{S}^{i+1} \leftarrow \mathcal{S}^i \cup \{s\}$.
12. | *Define* $\lambda(s) := 0$.
13. | *Set* $area \leftarrow \sum_{t \in \mathcal{V}} L^i(t)$.
14. └ *Set* $L^{i+1} \leftarrow L^i$ *and* $i \leftarrow i + 1$.
15. *Return* L^i.

Let $E^A = \max_{\mathcal{O} \in \mathcal{U}(\mathcal{S}_1, A)} E(\mathcal{O})$ denote the optimum energy value by Eq. 6 of a segmentation by OIFT using set A as background seeds in Theorem 1. In order to prove Theorem 2, we need first to establish some supporting propositions.

Proposition 2 *The optimum energy $E^{A\cup B}$, among all objects in $\mathcal{U}(\mathcal{S}_1, A \cup B)$, satisfies $E^{A\cup B} = \min\{E^A, E^B\}$.*

Proof Since the energy $E^{A\cup B}$ is optimum, we have:

$$E^{A\cup B} = \max_{\mathcal{O}\in\mathcal{U}(\mathcal{S}_1, A\cup B)} E(\mathcal{O}) \tag{9}$$

Given that $\mathcal{U}(\mathcal{S}_1, A \cup B) \subset \mathcal{U}(\mathcal{S}_1, A)$, we have that $\max_{\mathcal{O}\in\mathcal{U}(\mathcal{S}_1,A\cup B)} E(\mathcal{O}) \leq \max_{\mathcal{O}\in\mathcal{U}(\mathcal{S}_1,A)} E(\mathcal{O})$. Thus, $E^{A\cup B} \leq E^A$. By the same arguments, we also have $E^{A\cup B} \leq E^B$. So we can conclude that:

$$E^{A\cup B} \leq \min\{E^A, E^B\} \tag{10}$$

Let $\mathcal{O}_A \in \mathcal{U}(\mathcal{S}_1, A)$ and $\mathcal{O}_B \in \mathcal{U}(\mathcal{S}_1, B)$, such that $E(\mathcal{O}_A) = E^A$ and $E(\mathcal{O}_B) = E^B$. The object $\mathcal{D} = \mathcal{V}\backslash(\overline{\mathcal{O}}_A \cup \overline{\mathcal{O}}_B)$ is a valid solution in $\mathcal{U}(\mathcal{S}_1, A \cup B)$ (i.e., $\mathcal{S}_1 \subseteq \mathcal{D}$, $A \cap \mathcal{D} = \emptyset$ and $B \cap \mathcal{D} = \emptyset$). From Equation 9, since $\mathcal{D} \in \mathcal{U}(\mathcal{S}_1, A \cup B)$, we may conclude:

$$E(\mathcal{D}) \leq E^{A\cup B} \tag{11}$$

By Eq. 6 we have:

$$\begin{aligned}
&E(\mathcal{D}) = \min_{\langle s,t\rangle\in\mathcal{C}_{out}(\mathcal{D})} \omega(s,t) = \\
&\min_{\langle s,t\rangle\in\mathcal{C}_{out}(\mathcal{V}\backslash(\overline{\mathcal{O}}_A\cup\overline{\mathcal{O}}_B))} \omega(s,t) = \\
&\min_{\langle s,t\rangle\in\mathcal{C}_{in}(\overline{\mathcal{O}}_A\cup\overline{\mathcal{O}}_B)} \omega(s,t) = \\
&\min\left\{\min_{\langle s,t\rangle\in\mathcal{C}_{in}(\overline{\mathcal{O}}_A)|s\notin\overline{\mathcal{O}}_B} \omega(s,t), \min_{\langle s,t\rangle\in\mathcal{C}_{in}(\overline{\mathcal{O}}_B)|s\notin\overline{\mathcal{O}}_A} \omega(s,t)\right\} \geq \\
&\min\left\{\min_{\langle s,t\rangle\in\mathcal{C}_{in}(\overline{\mathcal{O}}_A)} \omega(s,t), \min_{\langle s,t\rangle\in\mathcal{C}_{in}(\overline{\mathcal{O}}_B)} \omega(s,t)\right\} = \\
&\min\left\{\min_{\langle s,t\rangle\in\mathcal{C}_{out}(\mathcal{O}_A)} \omega(s,t), \min_{\langle s,t\rangle\in\mathcal{C}_{out}(\mathcal{O}_B)} \omega(s,t)\right\} = \\
&\min\{E^A, E^B\}
\end{aligned}$$

By the above equation, we have $E(\mathcal{D}) \geq \min\{E^A, E^B\}$, which combined with Eq. 11 leads to $E^{A\cup B} \geq \min\{E^A, E^B\}$. But since $E^{A\cup B} \leq \min\{E^A, E^B\}$ according to Eq. 10, we have $E^{A\cup B} = \min\{E^A, E^B\}$. □

Proposition 3 *For a given strongly connected and symmetric digraph G, and sets of seeds $\mathcal{S}_1$ and $\mathcal{S}_0$, such that $\mathcal{S}_0 = \{t\}$, we have that $E^{\{t\}} = V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t)$, where $f_{\max}^{\mathcal{S}_1}$ is the path-cost function from Eq. 1, but being computed only from the object seeds in $\mathcal{S}_1$.*

Proof Consider the set of cutting arcs $\mathcal{C}_{out}(\mathcal{O}_{opt})$ of an optimum solution $\mathcal{O}_{opt} \in \mathcal{U}(\mathcal{S}_1, \{t\})$, i.e., $E^{\{t\}} = E(\mathcal{O}_{opt})$. By Eq. 6, we have that $\omega(a, b) \geq E^{\{t\}}$ for all $\langle a, b\rangle \in \mathcal{C}_{out}(\mathcal{O}_{opt})$. An optimum path $\underset{\mathcal{S}_1\rightsquigarrow t}{\pi}$, from $\mathcal{S}_1$ to t, must necessarily pass through some arc of $\mathcal{C}_{out}(\mathcal{O}_{opt})$. So its connectivity value $f_{\max}^{\mathcal{S}_1}(\underset{\mathcal{S}_1\rightsquigarrow t}{\pi}) = V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t)$ cannot be lower than $E^{\{t\}}$, i.e., $V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t) \geq E^{\{t\}}$ **(C1)**.

To conclude the proof, let us consider the following statement:

(⊛) There is a path $\underset{\mathcal{S}_1\rightsquigarrow t}{\tau} = \langle v_0, v_1, \ldots, v_\ell\rangle$ from $\mathcal{S}_1$ to $v_\ell = t$ in G, which is composed only by arcs $\langle v_i, v_{i+1}\rangle$, such that $\omega(v_i, v_{i+1}) \leq E^{\{t\}}$, $i = 0, 1, \ldots, \ell - 1$.

This statement can be proven by contradiction. Let R be the set of nodes that are still reachable from $\mathcal{S}_1$ by paths after removing all arcs $\langle a, b\rangle$ such that $\omega(a, b) > E^{\{t\}}$. From the contradiction hypothesis we assume that $t \notin R$, thus $R \in \mathcal{U}(\mathcal{S}_1, \{t\})$. Note that the cutting arcs $\langle a, b\rangle \in \mathcal{C}_{out}(R)$ all have $\omega(a, b) > E^{\{t\}}$ in G. Consequently, $\mathcal{C}_{out}(R)$ has a better cut value than $E^{\{t\}}$ ($E(R) > E^{\{t\}}$), which leads to a contradiction by the definition of $E^{\{t\}}$.

From statement **(⊛)**, we may conclude that the connectivity value $V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t)$ of an optimum path from $\mathcal{S}_1$ to t cannot be higher than $E^{\{t\}}$, i.e., $V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t) \leq f_{\max}^{\mathcal{S}_1}(\underset{\mathcal{S}_1\rightsquigarrow t}{\tau}) \leq E^{\{t\}}$ **(C2)**.

From the above conditions **(C1)** and **(C2)**, we may conclude that the only valid configuration is $V_{opt}^{f_{\max}^{\mathcal{S}_1}}(t) = E^{\{t\}}$. □

Algorithm 4 defines an optimal object of maximum energy via OIFT but having area/volume less than or equal to a given threshold T. It assumes that the defined background must be connected to the originally selected background seeds (i.e., the solution belongs to $\widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$). If the object by OIFT has an area above the threshold (Line 7), we can reduce its size by inserting new background seeds in its boundary (Lines 10-12). The energies of background nodes can be computed by IFT with $f_{\max}^{\mathcal{S}_1}$ from the object seeds in $\mathcal{S}_1$, according to Proposition 3 (Line 1). In order to get an optimal object, at each iteration we must then select a new background seed at the highest energy node of the object's boundary (Line 10). We can then repeat this procedure until the area of the resulting object falls below or equals the given threshold. We therefore have a sequence of OIFTs, with a new OIFT for each new inserted seed, that can be calculated faster by DOIFT (Line 8). Next, we present a proof of Theorem 2.

Proof The proof is based on the following loop invariant of Algorithm 4:

(⋆) At each iteration of the main loop of Algorithm 4, we have an object $\mathcal{O}_1^i = \{t \in \mathcal{V} \mid L^i(t) = 1\}$ (Line 9), such that $\mathcal{O}_1^i \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$ and $\exists\ \mathcal{O}^* \subseteq \mathcal{O}_1^i$, $i > 0$, where

$\mathcal{O}^*$ is an optimal solution for the conditions established in Theorem 2 (i.e., $E(\mathcal{O}^*) = \max_{\mathcal{O} \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)} E(\mathcal{O})$ and $\mathcal{O}^* \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)$).

This invariant can be proved by induction. For the induction basis ($i = 1$), observe that $\mathcal{O}_1^1$ is equivalent to the result of a regular OIFT from $\mathcal{S}_1$ and $\mathcal{S}_0$. So we have $\mathcal{O}_1^1 \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$, due to the propagation of labels from seeds by OIFT. To prove the existence of an optimal solution $\mathcal{O}^*$ contained in $\mathcal{O}_1^1$, note that $E(\mathcal{O}_1^1) \geq E(\mathcal{O}^*)$, since $\widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0) \subseteq \mathcal{U}(\mathcal{S}_1, \mathcal{S}_0)$ and $\mathcal{O}_1^1$ is optimal in $\mathcal{U}(\mathcal{S}_1, \mathcal{S}_0)$ (Theorem 1). So we have $\omega(s,t) \geq E(\mathcal{O}^*)$ for all $\langle s,t \rangle \in \mathcal{C}_{out}(\mathcal{O}_1^1)$. Assume that $\mathcal{O}^* \nsubseteq \mathcal{O}_1^1$ and consider $\mathcal{O}' = \mathcal{O}^* \cap \mathcal{O}_1^1$. Note that $\mathcal{O}' \neq \emptyset$, since $\mathcal{S}_1 \subseteq \mathcal{O}^*$ and $\mathcal{S}_1 \subseteq \mathcal{O}_1^1$. Now notice that $\mathcal{C}_{out}(\mathcal{O}') = \mathcal{X} \cup \mathcal{Y}$, where $\mathcal{X} = \mathcal{C}_{out}(\mathcal{O}') \cap \mathcal{C}_{out}(\mathcal{O}_1^1)$ and $\mathcal{Y} = \mathcal{C}_{out}(\mathcal{O}') \cap \mathcal{C}_{out}(\mathcal{O}^*)$. Given that $\min_{\langle s,t \rangle \in \mathcal{X}} \omega(s,t) \geq E(\mathcal{O}_1^1) \geq E(\mathcal{O}^*)$ and $\min_{\langle s,t \rangle \in \mathcal{Y}} \omega(s,t) \geq E(\mathcal{O}^*)$, we can conclude that $E(\mathcal{O}') = \min_{\langle s,t \rangle \in \mathcal{X} \cup \mathcal{Y}} \omega(s,t) \geq E(\mathcal{O}^*)$. It remains only to prove that $\mathcal{O}' \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)$. Note that $|\mathcal{O}'| \leq T$ since $\mathcal{O}' \subseteq \mathcal{O}^*$, so it suffices to prove that $\mathcal{O}' \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$. Now notice that $\overline{\mathcal{O}}' = \overline{\mathcal{O}^* \cap \mathcal{O}_1^1} = \overline{\mathcal{O}}^* \cup \overline{\mathcal{O}}_1^1$. Given that $\mathcal{O}^*, \mathcal{O}_1^1 \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$, then we have $\forall t \in \overline{\mathcal{O}}^* \; \exists \underset{\mathcal{S}_0 \rightsquigarrow t}{\pi} [\overline{\mathcal{O}}']$ and $\forall t \in \overline{\mathcal{O}}_1^1 \; \exists \underset{\mathcal{S}_0 \rightsquigarrow t}{\pi} [\overline{\mathcal{O}}']$, so we can conclude $\forall t \in \overline{\mathcal{O}}' \; \exists \underset{\mathcal{S}_0 \rightsquigarrow t}{\pi} [\overline{\mathcal{O}}']$. So we can take $\mathcal{O}'$ as an optimal solution contained in $\mathcal{O}_1^1$, since $E(\mathcal{O}') = \max_{\mathcal{O} \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)} E(\mathcal{O})$.

For the induction step, we must prove that if the statement ($\star$) holds for i, then it also holds for $i+1$. The condition that $\mathcal{O}_1^{i+1} \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$ is easy to prove, as it is enough to observe that the new additional seed s in $\mathcal{S}^{i+1} \backslash \mathcal{S}^i$ (Line 11) is selected on the boundary of $\mathcal{O}_1^i$, so for all new nodes t conquered to the background in $\overline{\mathcal{O}}_1^{i+1} \backslash \overline{\mathcal{O}}_1^i$, we have that there is a path $\underset{\mathcal{S}_0 \rightsquigarrow t}{\pi} [\overline{\mathcal{O}}_1^{i+1}]$. By the induction hypothesis, there is an optimal solution $\mathcal{O}^*$, such that $\mathcal{O}^* \subseteq \mathcal{O}_1^i$. We must prove that there is also an optimal solution contained in $\mathcal{O}_1^{i+1}$. The object $\mathcal{O}_1^{i+1}$ is only computed when $|\mathcal{O}_1^i| > T \geq |\mathcal{O}^*|$. Among all the ways to reduce $\mathcal{O}_1^i$ to another smaller object in $\widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$, in Algorithm 4, the choice leading to the highest possible energy is performed (Line 10), according to Propositions 2 and 3.[3] Therefore, we have that $E(\mathcal{O}_1^{i+1}) \geq E(\mathcal{O}^*)$. The remainder of the proof is analogous to the arguments used for the base case and therefore we will present it here only briefly. Assume that $\mathcal{O}^* \nsubseteq \mathcal{O}_1^{i+1}$ and consider $\mathcal{O}' = \mathcal{O}^* \cap \mathcal{O}_1^{i+1}$. Note that $\mathcal{O}' \neq \emptyset$, since $\mathcal{S}_1 \subseteq \mathcal{O}^*$ and $\mathcal{S}_1 \subseteq \mathcal{O}_1^{i+1}$. Also note that $|\mathcal{O}'| \leq T$ since $\mathcal{O}' \subseteq \mathcal{O}^*$. Now notice that $\mathcal{C}_{out}(\mathcal{O}') = \mathcal{X}' \cup \mathcal{Y}'$, where $\mathcal{X}' = \mathcal{C}_{out}(\mathcal{O}') \cap \mathcal{C}_{out}(\mathcal{O}_1^{i+1})$ and $\mathcal{Y}' = \mathcal{C}_{out}(\mathcal{O}') \cap \mathcal{C}_{out}(\mathcal{O}^*)$. Given that $\min_{\langle s,t \rangle \in \mathcal{X}'} \omega(s,t) \geq E(\mathcal{O}_1^{i+1}) \geq E(\mathcal{O}^*)$ and $\min_{\langle s,t \rangle \in \mathcal{Y}'} \omega(s,t) \geq E(\mathcal{O}^*)$, we can conclude that $E(\mathcal{O}') = \min_{\langle s,t \rangle \in \mathcal{X}' \cup \mathcal{Y}'} \omega(s,t) \geq E(\mathcal{O}^*)$. So we can take $\mathcal{O}'$ as an optimal solution contained in $\mathcal{O}_1^{i+1}$.

Now that we have proved the loop invariant of Algorithm 4; we can conclude the proof of Theorem 2. Note that by ($\star$) we always have an object $\mathcal{O}_1^i \in \widetilde{\mathcal{U}}(\mathcal{S}_1, \mathcal{S}_0)$, $i = 1, \ldots, n$, which contains an optimal solution to Theorem 2. As object $\mathcal{O}_1^i$ decreases in size for increasing values of i, the loop in Line 7 will break for $i = n$ such that $|\mathcal{O}_1^n| \leq T$. Note that $\mathcal{O}_1^n \in \widetilde{\mathcal{U}}_T(\mathcal{S}_1, \mathcal{S}_0)$. Also note that $\mathcal{O}_1^n$ equals the result of an OIFT with additional background seeds, so we have that $\mathcal{O}_1^n \in \mathcal{U}(\mathcal{S}_1, \mathcal{S}^n \backslash \mathcal{S}_1)$ has optimal energy among all solutions in $\mathcal{U}(\mathcal{S}_1, \mathcal{S}^n \backslash \mathcal{S}_1)$. By the invariant ($\star$) we have that there is $\mathcal{O}^* \subseteq \mathcal{O}_1^n$. Since $\mathcal{O}^* \in \mathcal{U}(\mathcal{S}_1, \mathcal{S}^n \backslash \mathcal{S}_1)$, we conclude that $E(\mathcal{O}_1^n) \geq E(\mathcal{O}^*)$. So the computed result $\mathcal{O}_1^n$ is optimal as we wanted to prove. □

A problem with Algorithm 4 is the fact that it has to identify and process the entire object boundary on Line 10 for every generated object $\mathcal{O}_1^i$. Another problem is that if the threshold T changes (for example, as in an interactive segmentation), we would have to run the algorithm again. Next, we present Algorithm 5, which generates a hierarchy map $H : \mathcal{V} \rightarrow \{|\mathcal{S}_1|, \ldots, |\mathcal{V}|\}$ that encodes area-constrained OIFT results for all possible thresholds T, so that $\mathcal{O}_H^T = \{t \in \mathcal{V} \mid H(t) \leq T\}$ equals the result of Algorithm 4. Algorithm 5 is also modified to update the boundary nodes of the generated objects $\text{bd}(\mathcal{O}_1^i)$ differentially during the DOIFT computation, keeping boundary nodes in a priority queue Q_E. For this purpose, Algorithm 5 uses a modified DOIFT given by Algorithm 6. The main differences between Algorithm 2 and Algorithm 6 are the inclusion of Lines 9-12 and Lines 28-31 in Algorithm 6. Lines 9-11 assign values to map H for the regions conquered by the background in the current DOIFT, considering only regions not yet included in the hierarchy ($H(s) = 0$). Line 12 and Lines 28-31 update the boundary nodes stored in Q_E. The priority queue Q_E is used to allow the efficient selection of the boundary element of maximum energy in the map E (Line 10 of Algorithm 5).

[3] Note that in Line 10 of Algorithm 4, a new seed s is selected from $\text{bd}(\mathcal{O}_1^i)$ such that $E^{\{b\}} \leq E^{\{s\}}$ for any $b \in \text{bd}(\mathcal{O}_1^i)$, which implies that $E^{\mathcal{S}_0^i \cup \{b\}} \leq E^{\mathcal{S}_0^i \cup \{s\}}$ according to Proposition 2, where $\mathcal{S}_0^i$ denotes the background seeds of the ith iteration. Therefore, the new set of background seeds $\mathcal{S}_0^{i+1} = \mathcal{S}_0^i \cup \{s\}$ is the one leading to the highest possible energy.

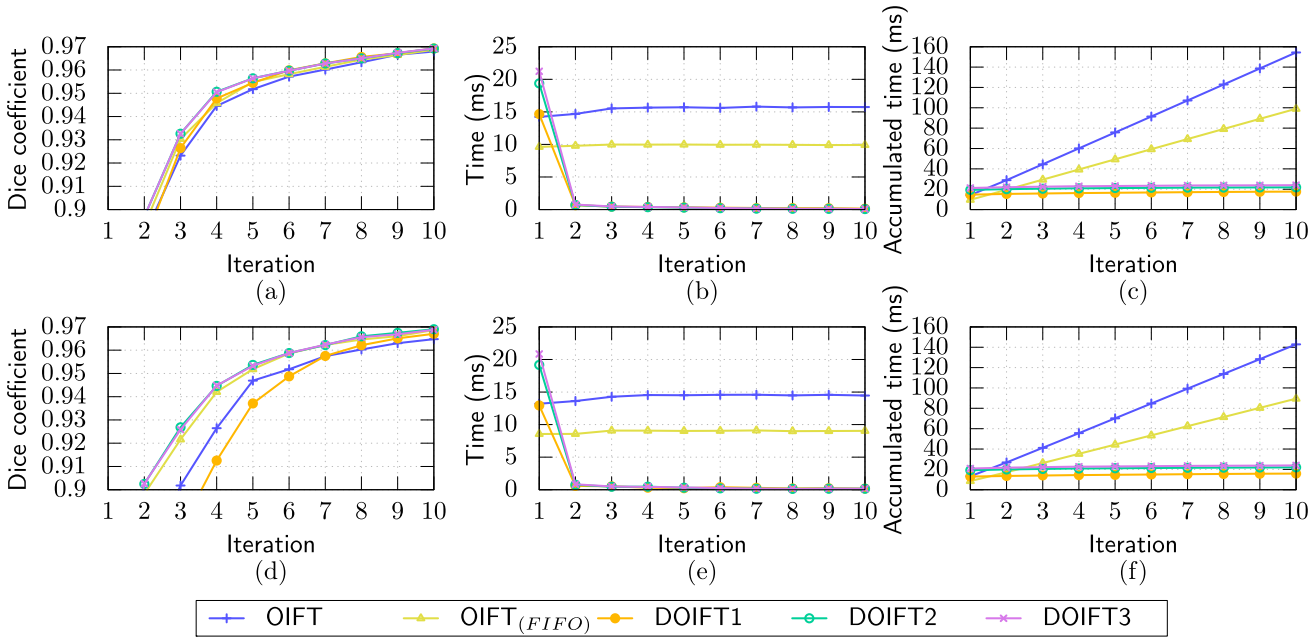


Fig. 6 The mean curves of accuracy, time, and accumulated time. $OIFT$ stands for a $f_2^{\sum}$ implementation with binary heap for the priority queue Q, while $OIFT(FIFO)$ considers $f_2^{\sum}$ with a bucket sorting for Q and with FIFO tie-breaking policy

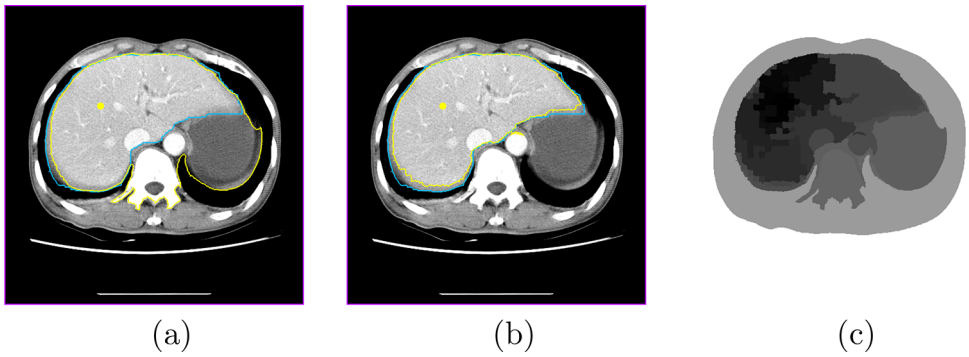


Fig. 7 Example of the liver segmentation using area-constrained OIFT for different values of T and 2500 superpixels. The blue line indicates the gold standard and the yellow line the segmentation results. **a** The result with $T = 740$. **b** An improved result is obtained by lowering the threshold to $T = 440$, which is done by simply sliding a slider in our graphical interface. **c** The hierarchy map H computed by Algorithm 5

Algorithm 5 – OIFT AREA HIERARCHY

INPUT: Image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, the seed sets $\mathcal{S}_1$ and $\mathcal{S}_0$, initial labeling function $\lambda : \mathcal{S} \to \{0, 1\}$.
OUTPUT: Hierarchy map $H : \mathcal{V} \to \{|\mathcal{S}_1|, \ldots, |\mathcal{V}|\}$.
AUXILIARY: Priority queue Q_E, the energy map $E : \mathcal{V} \to \mathbb{R}$, the label map $L : \mathcal{V} \to \{0, 1\}$, path-cost map $V : \mathcal{V} \to \mathbb{R}^2$, the spanning forest $P : \mathcal{V} \to \mathcal{V} \cup \{nil\}$, and variable $area$.

1. $(L, R, E, P) \leftarrow IFT(G, f_{\max}^{\mathcal{S}_1}, \mathcal{S}_1, \lambda)$.
2. *Set* $Q_E \leftarrow \emptyset$.
3. **For each** $t \in \mathcal{V}$, **do**
4. | *Set* $P(t) \leftarrow nil$ *and* $V(t) \leftarrow \langle \infty, \infty \rangle$.
5. | *Set* $H(t) \leftarrow 0$.
6. *Set* $area \leftarrow |\mathcal{V}|$.
7. *Set* $\Delta_{\mathcal{S}}^{+} \leftarrow \mathcal{S}_1 \cup \mathcal{S}_0$.
8. **While** $area > |\mathcal{S}_1|$, **do**
9. | *H_DOIFT*$(G, \Delta_{\mathcal{S}}^{+}, L, V, P, \lambda, Q_E, H, area)$.
10. | *Remove* s *from* Q_E *such that* $E(s) = \max_{t \in Q_E}\{E(t)\}$.
11. | *Set* $\Delta_{\mathcal{S}}^{+} \leftarrow \{s\}$.
12. | *Define* $\lambda(s) := 0$.
13. **For each** $t \in \mathcal{S}_1$, **do**
14. | *Set* $H(t) \leftarrow |\mathcal{S}_1|$.
15. *Return* H.

Algorithm 6 – H_DOIFT: ALGORITHM DOIFT FOR HIERARCHY CONSTRUCTION

INPUT: Image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, the set Δ_S^+ of seeds for addition, the maps L, V and P initialized with the result from the previous OIFT/DOIFT execution, an initial labeling function $\lambda : \Delta_S^+ \to \{0, 1\}$ for the new seeds, the priority queue Q_E, the hierarchy map $H : \mathcal{V} \to \{0, \ldots, |\mathcal{V}|\}$, and variable $area$. We consider $V(t) = \langle V_1(t), V_2(t) \rangle$ as we work with lexicographical costs.
OUTPUT: The updated maps L, V, P, H, priority queue Q_E, and variable $area$ passed by reference.
AUXILIARY: Priority queue Q, and variables tmp_1, tmp_2 and $hlevel$.

1. *Set* $hlevel \leftarrow area$.
2. *Set* $Q \leftarrow \emptyset$.
3. **For each** $s \in \Delta_S^+$, **do**
4. *Set* $L(s) \leftarrow \lambda(s)$ *and* $P(s) \leftarrow nil$.
5. *Set* $V(s) \leftarrow \langle -1, L(s) + 1 \rangle$.
6. **If** $s \notin Q$, **then** *insert* s *in* Q.
7. **While** $Q \neq \emptyset$, **do**
8. *Remove* s *from* Q *such that* $V(s) \overset{\text{lex}}{\leq} V(r)$ *for all* $r \in Q$.
9. **If** $L(s) = 0$, **then**
10. **If** $H(s) = 0$, **then**
11. *Set* $H(s) \leftarrow hlevel$ *and* $area \leftarrow area - 1$.
12. **If** $s \in Q_E$, **then** *remove* s *from* Q_E.
13. **For each** *node* t *such that* $\langle s, t \rangle \in \mathcal{A}$, **do**
14. *Compute* $tmp_1 \leftarrow F_2^{\sigma}(\pi_s^P \cdot \langle s, t \rangle)$.
15. **If** $tmp_1 \neq f_2^{\sigma}(\pi_s^P \cdot \langle s, t \rangle)$, **then**
16. *Set* $tmp_2 \leftarrow V_2(s)$.
17. **Else If** $V_1(s) = tmp_1$, **then**
18. *Set* $tmp_2 \leftarrow V_2(s) + 2$.
19. **Else**, **then**
20. *Set* $tmp_2 \leftarrow L(s) + 1$.
21. **If** $\langle tmp_1, tmp_2 \rangle \overset{\text{lex}}{<} V(t)$, **then**
22. *Set* $L(t) \leftarrow L(s)$ *and* $P(t) \leftarrow s$.
23. *Set* $V(t) \leftarrow \langle tmp_1, tmp_2 \rangle$.
24. **If** $t \notin Q$, **then** *insert* t *in* Q.
25. **Else If** $s = P(t)$ *and* $(tmp_1 \neq V_1(t)$ *or* $tmp_2 > V_2(t)$ *or* $L(t) \neq L(s))$, **then**
26. *DOIFT-RemoveSubTrees*$(G, L, V, P, Q, \{t\})$
27. ***Break;*** *#GOTO LINE 8*
28. **Else If** $L(s) = 0$ *and* $L(t) = 1$, **then**
29. **If** $t \notin Q_E$, **then** *insert* t *in* Q_E.
30. **Else If** $L(s) = 1$ *and* $L(t) = 0$ *and* $H(t) > 0$, **then**
31. **If** $s \notin Q_E$, **then** *insert* s *in* Q_E.

5 Experimental Results

Figure 6 shows the experimental curves for the segmentation of the talus bone using 40 slices of 256 × 256 pixels from MR images of the foot by using a robot user.[4] In the first row, the arc weights were defined as $\omega(s, t) = \|I(t) - I(s)\|$ and

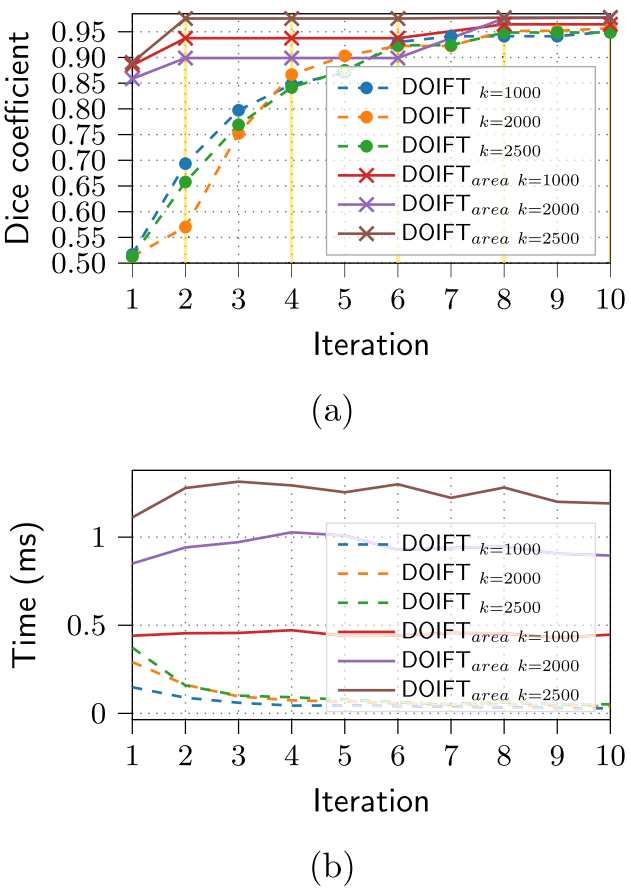


Fig. 8 The mean experimental curves for the liver segmentation: **a** Accuracy by Dice coefficient. **b** Execution time in ms

with boundary polarity parameter defined as -50% (see [1]). In the second row, we repeat the experiment but with the arc weights quantized in a smaller range of values, corresponding to a quarter of the original range. $DOIFT_1$ and $OIFT$ (with f_2^{σ} and a heap priority queue) had a performance drop in the second case, due to their worse handling of tie zones. $DOIFT_2$ and $DOIFT_3$ had an accuracy performance consistent with the $OIFT$ with FIFO tie-breaking policy using f_2^{σ}. In case of $OIFT(FIFO)$, we considered a bucket sorting for Q and a binary heap was used for all other cases. Even using a slower queue Q, differential approaches were faster than $OIFT(FIFO)$ with the exception of the first iteration. The times were measured on an Intel Core i5-10210U CPU @ 1.60GHz×8 machine with 8 GiB memory.

In the second experiment, we aim to demonstrate the effectiveness of using high-level priors via the maximum allowable size T by Algorithm 5, in order to achieve a good segmentation of the desired object with less user effort. We used 40 computed tomography (CT) slices of size 512 × 512 pixels, obtained from thoracic studies of 10 individuals, to segment the liver (Fig. 7). We used a region adjacency graph of superpixels and compared the Dice coefficients, for dif-

[4] We used a robot user [37] to simulate user interaction by placing brush strokes automatically to iteratively perform the segmentation task. At each iteration, the robot user selects a new corrective seed in the largest connected component of mislabeled pixels, placed at a point farthest from the boundary of the component, in order to imitate the behavior of a real user.

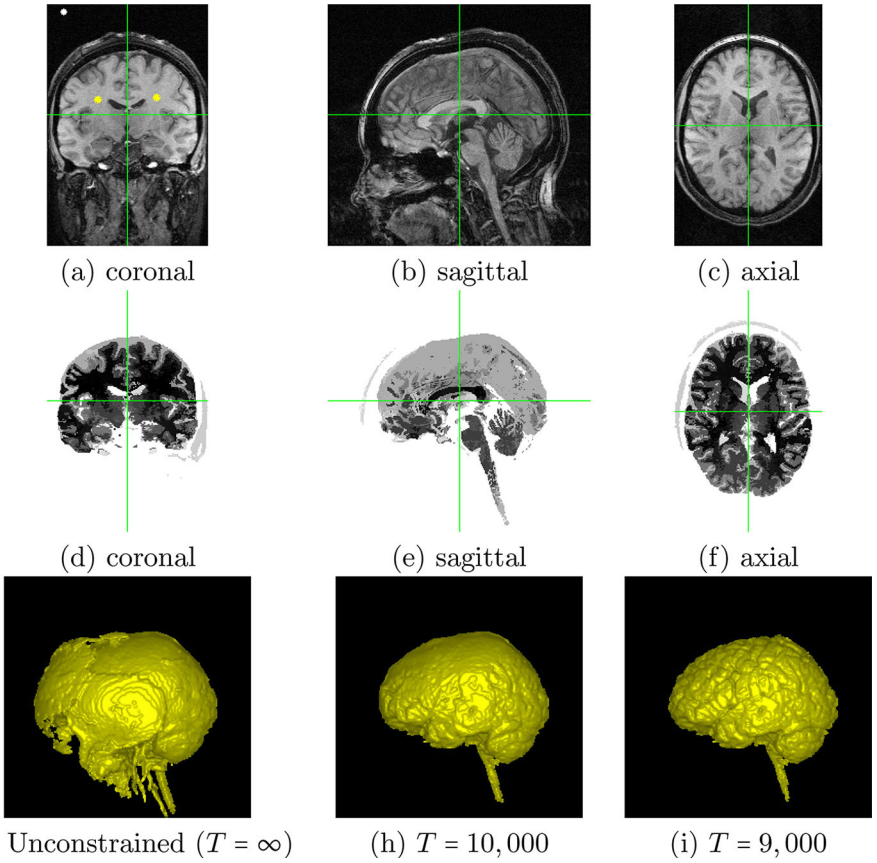


Fig. 9 Brain segmentation in an MR image. **a**–**c** The coronal, sagittal and axial planes. Only three markers were selected in the indicated coronal slice. **d**–**f** The coronal, sagittal and axial planes of the hierarchy map H computed by Algorithm 5. **g**–**i** Area-constrained OIFT segmentation results computed by Algorithm 4 for different values of T

ferent values of k (number of superpixels). In Fig. 8, the $DOIFT$ curves represent the segmentation results of 10 interactions of the robot user, with a new corrective seed being selected for each robot interaction and without area constraints. In $DOIFT_{area}$ curves, the robot interactions occur in pairs, with the selection of a new seed by the robot user in odd interactions, followed by an even interaction for the selection of the area threshold T that best approximates the desired object. Therefore, the process of searching for the best area threshold was considered a user intervention. Note that, in practice, a human user could easily make this selection by sliding a slider in the program's interface with immediate visual feedback, because the computed hierarchy map H encodes all possible area-constrained outcomes by its thresholding. In the first iteration, we consider the automatic selection of T, as the one that generates the object with the closest size to a fixed average area expected for the dataset.[5] Note that from the second interaction of $DOIFT_{area}$, by tuning the threshold, it is already possible to obtain very good results (Fig. 8a). We also computed the standard Number of Clicks (NoC) measure, which is commonly used to report the results of recent deep learning-based algorithms for interactive segmentation [27]. NoC measures the number of interactions required to achieve a predefined intersection over union (IoU) threshold between predicted and ground truth masks. We denote NoC with IoU threshold set to 85% and 90% as NoC@85 and NoC@90, respectively. For the liver segmentation, $DOIFT_{area}$ has achieved a value of NoC@90=1.3, which is very close to the optimal limit value of NoC for interactive methods, so the presented results are competitive with the state of the art. It also obtained NoC@85 = 1.0, which is the best possible value for this measure.

In order to get an idea of the impact of applying DOIFT in the context of large 3D MR images, we also carried out experiments in a 3D MR-T1 image with volume size of $256 \times 256 \times 157$ voxels to segment the brain, using the area-constrained OIFT segmentation from Sect. 4. The image was acquired with a 2T Elscint scanner and at a voxel size of $0.98 \times 0.98 \times 1.00\ mm^3$. We considered a region adjacency graph of supervoxels by [32] with an average of 100 voxels per region. The arc weights were defined as $\omega(s,t) = \|I(t) - I(s)\|$ (where $I(t)$ represents the average intensity of each supervoxel t) and with boundary polarity parameter defined as 50% (see [1]). Figure 9 shows the results obtained for different values of the maximum volume threshold T, which is expressed in number of supervox-

[5] In the first interaction of the robot user, a seed is selected for the object to start the process. The background seed is not counted as a user interaction as we use a fixed set of background seeds at the image border for all images.

els. To measure the time gain obtained by using DOIFT in Algorithm 4, we compared its execution time with the time that would be spent if a regular OIFT had been used in Line 8 of Algorithm 4. So, regarding the execution time of Algorithm 4, for $T = 9,000$ we had 1 sec if $DOIFT_2$ (Algorithm 2) is employed and 75 sec if a regular OIFT is employed with heap. For $T = 10,000$, we had 0.86 sec for $DOIFT_2$ and 64 sec for OIFT. On the other hand, the computation of the full hierarchy map H via Algorithm 5 is even faster, taking only 127 ms and allowing us to obtain the same results via the simple thresholding of H (Fig. 9d–f). The times were measured on an Intel Core i5-10210U CPU @ 1.60GHz×8 machine with 8 GiB memory.

6 Conclusion

We have successfully tested different approaches to implement the differential OIFT, as well as the use of DOIFT for implementing an area/volume constraint in OIFT and for creating a hierarchy of OIFT segmentations by varying the area threshold. The computed hierarchy map helps the user to quickly select the appropriate area threshold, improving segmentation considerably, without the need to select multiple markers.

As future works, we intend to investigate other applications and operators based on DOIFT. In this work, we considered the area constraint as an upper bound, but note that we could equally implement it as a lower bound, by requiring the resulting generated object to have an area above the given threshold. How to calculate an optimal result with area comprised in a range of allowed values is a topic for investigation.

Acknowledgements Thanks are due to Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – (Grant 407242/2021-0, 313087/2021-0, 465446/2014-0, 166631/2018-3), CAPES (88887.136422/2017-00) and FAPESP (2014/12236-1, 2014/50937-1).

Author Contributions The authors contributed equally to this work.

Funding This work was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico—CNPq—(Grant 407242/2021-0, 313087/2021-0, 465446/2014-0, 166631/2018-3), CAPES (88887.136422/2017-00) and FAPESP (2014/12236-1, 2014/50937-1).

Availability of data and materials Not applicable.

Code Availability Our code is available upon request to the corresponding author.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

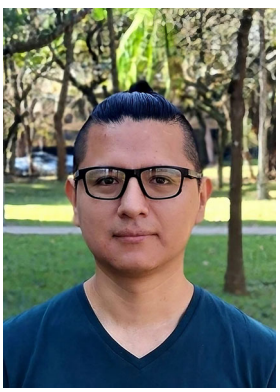
Ethics Approval This paper is an extended version of a conference paper presented at International Conference on Discrete Geometry and Mathematical Morphology (DGMM22), as an invited paper to a special issue of JMIV.

Consent to participate This research did not directly involve human participants or animals. The medical images used in this paper were provided by the hospital of Unicamp and the University of Pennsylvania preserving the anonymity of their patients and subjects.

References

1. 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)
2. Bejar, H.H.C., Miranda, P.A.V.: Oriented relative fuzzy connectedness: Theory, algorithms, and its applications in hybrid image segmentation methods. EURASIP J. Image Video Process. **2015**(21) (2015)
3. Ciesielski, K.C., Udupa, J.K., Falcão, A.X., Miranda, P.A.V.: A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them. In: Proc. of SPIE on Med. Imaging: Image Process., vol. 8314 (2012)
4. Ciesielski, K.C., Udupa, J.K., Saha, P.K., Zhuge, Y.: Iterative relative fuzzy connectedness for multiple objects with multiple seeds. Comput. Vis. Image Underst. **107**(3), 160–182 (2007)
5. 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 (2008)
6. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Intl. Jrnl. of Comp. Vision **70**(2), 109–131 (2006)
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. Ciesielski, K.C., Herman, G.T., Kong, T.Y.: General theory of fuzzy connectedness segmentations. J. Math. Imaging Vision **55**(3), 304–342 (2016)
9. de Moraes Braz, C., Miranda, P.A.V., Ciesielski, K.C., Cappabianco, F.A.M.: Optimum cuts in graphs by general fuzzy connectedness with local band constraints. J. Math. Imaging Vis. 62, 659–672 (2020)
10. Tavares, A.C.M., Bejar, H.H.C., Miranda, P.A.V.: Seed robustness of oriented image foresting transform: core computation and the robustness coefficient. In: Angulo, J., Velasco-Forero, S., Meyer, F. (eds.) Math. Morphol. Appl. Signal Image Process., pp. 119–130. Springer, Cham (2017)
11. Perret, B., Cousty, J., Tankyevych, O., Talbot, H., Passat, N.: Directed connected operators: asymmetric hierarchies for image filtering and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **37**(6), 1162–1176 (2015)
12. Choi, J., Ueda, E.K., Duran, G.C., Miranda, P.A.V., Tsuzuki, M.S.G.: Automatic lung segmentation with seed generation and ROIFT algorithm for the creation of anatomical atlas. In: ICGG 2022 - Proceedings of the 20th International Conference on Geometry and Graphics, pp. 636–647. Springer, Cham (2023)
13. Mansilla, L.A.C., Miranda, P.A.V., Cappabianco, F.A.M.: Oriented image foresting transform segmentation with connectivity constraints. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 2554–2558 (2016)
14. Mansilla, L.A.C., Miranda, P.A.V.: Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In: 29th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 289–296 (2016)

15. de Moraes Braz, C.: Graph-based segmentation with local band constraints. In: Couprie, M., Cousty, J., Kenmochi, Y., Mustafa, N. (eds.) Discrete Geometry for Computer Imagery. Springer, Cham (2019)
16. Mansilla, L.A.C., Miranda, P.A.V.: Image Segmentation by Oriented Image Foresting Transform with Geodesic Star Convexity. In: 15th Intl. Conference on Computer Analysis of Images and Patterns (CAIP), vol. 8047. York, UK, pp. 572–579 (2013)
17. de Moraes Braz, C., Miranda, P.A.V.: Image segmentation by image foresting transform with geodesic band constraints. In: Image Processing (ICIP), 2014 IEEE International Conference On, pp. 4333–4337 (2014)
18. Mansilla, L.A.C., Miranda, P.A.V.: Image Segmentation by Oriented Image Foresting Transform: Handling Ties and Colored Images. In: 18th International Conference on Digital Signal Processing, Greece, pp. 1–6 (2013)
19. Leon, L.M.C., Ciesielski, K.C., Miranda, P.A.V.: Efficient hierarchical multi-object segmentation in layered graphs. Math. Morphol. Theory Appl. **5**(1), 21–42 (2021). https://doi.org/10.1515/mathm-2020-0108
20. Leon, L.M.C., Miranda, P.A.V.D.: Multi-object segmentation by hierarchical layered oriented image foresting transform. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 79–86 (2017)
21. Lézoray, O., Grady, L.: Image Processing and Analysis with Graphs: Theory and Practice. CRC Press, California, USA (2012)
22. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling Trans. On Pattern Anal. Mach. Intell. **35**, 1915–1929 (2013)
23. Belém, F., Borlido, I., João, L., Perret, B., Cousty, J., Guimarães, S.J.F., Falcão, A.: Fast and effective superpixel segmentation using accurate saliency estimation. In: Baudrier, É., Naegel, B., Krähenbühl, A., Tajine, M. (eds.) Discrete Geometry and Mathematical Morphology, pp. 261–273. Springer, Cham (2022)
24. Belém, F., Perret, B., Cousty, J., Guimarães, S.J.F., Falcão, A.: Efficient Multiscale Object-based Superpixel Framework. arXiv (2022). https://arxiv.org/abs/2204.03533
25. Oliveira, D.E.C., Demario, C.L., Miranda, P.A.V.: Image segmentation by relaxed deep extreme cut with connected extreme points. In: Lindblad, J., Malmberg, F., Sladoje, N. (eds.) Discrete Geometry and Mathematical Morphology, pp. 441–453. Springer, Cham (2021)
26. Wolf, S., Schott, L., Kothe, U., Hamprecht, F.: Learned Watershed: End-to-End Learning of Seeded Segmentation. In: International Conference on Computer Vision (ICCV), pp. 2030–2038 (2017)
27. Sofiiuk, K., Petrov, I.A., Konushin, A.: Reviving iterative training with mask guidance for interactive segmentation. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 3141–3145 (2022)
28. Braz, C.d.M., Santos, L.F.D., Miranda, P.A.V.: Graph-based image segmentation with shape priors and band constraints. In: Baudrier, É., Naegel, B., Krähenbühl, A., Tajine, M. (eds.) Discrete Geometry and Mathematical Morphology, pp. 287–299. Springer, Cham (2022)
29. Falcão, A.X., Stolfi, J., Lotufo, R.A.: The image foresting transform: Theory, algorithms, and applications. IEEE TPAMI **26**(1), 19–29 (2004)
30. Falcão, A.X., Bergo, F.P.G.: Interactive volume segmentation with differential image foresting transforms. IEEE Trans on Medical Imaging **23**(9), 1100–1108 (2004)
31. Condori, M.A.T., Cappabianco, F.A.M., Falcão, A.X., Miranda, P.A.V.: An extension of the differential image foresting transform and its application to superpixel generation. J. Vis. Commun. Image Rep. **71**, 102748 (2020)
32. Vargas-Muñoz, J.E., Chowdhury, A.S., Alexandre, E.B., Galvão, F.L., Miranda, P.A.V., Falcão, A.X.: An iterative spanning forest framework for superpixel segmentation. IEEE Trans. Image Process. **28**(7), 3477–3489 (2019)
33. Galvão, F.L., Falcão, A.X., Chowdhury, A.S.: Risf: Recursive iterative spanning forest for superpixel segmentation. In: 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 408–415 (2018)
34. Alexandre, E.B., 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: 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 337–344 (2015)
35. Condori, M.A.T., Miranda, P.A.V.: Differential oriented image foresting transform segmentation by seed competition. In: Baudrier, É., Naegel, B., Krähenbühl, A., Tajine, M. (eds.) Discrete Geometry and Mathematical Morphology, pp. 300–311. Springer, Cham (2022)
36. Ciesielski, K.C., Falcão, A.X., Miranda, P.A.V.: Path-value functions for which dijkstra's algorithm returns optimal mapping. J. Math. Imaging Vis. **60**(7), 1025–1036 (2018)
37. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: Proceeding of Computer Vision and Pattern Recognition, pp. 3129–3136 (2010)



Marcos A. T. Condori received the M.Sc. degree in Computer Science from the Institute of Mathematics and Statistics, University of São Paulo, SP, Brazil in 2017 where he is currently a Ph.D. candidate. His main research interests are image segmentation, data visualization and computer vision

Paulo A. V. Miranda is currently professor at the Institute of Mathematics and Statistics (IME) of the University of São Paulo (USP), SP, Brazil. He received a B.Sc. in Computer Engineering (2003) and a M.Sc. in Computer Science (2006) from the University of Campinas (UNICAMP), SP, Brazil. During 2008-2009, he was with the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, USA, where he worked on image segmentation for his doctorate. He got his doctorate in Computer Science from the University of Campinas (UNICAMP) in 2009. After that, he worked as a post-doctoral researcher in a project in conjunction with the professors of the Department of Neurology, UNICAMP. He has experience in computer science, with emphasis on computer vision, image processing, and pattern recognition.