Differential Oriented Image Foresting Transform Segmentation by Seed Competition

Marcos A. T. Condori and Paulo A. V. Miranda(✉)

Institute of Mathematics and Statistics, University of São Paulo, São Paulo, SP 05508-090, Brazil
{mtejadac,pmiranda}@ime.usp.br

Abstract. The 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 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 the Generalized Graph Cut and the 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 the Differential Oriented Image Foresting Transform (DOIFT), 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 area constraints in OIFT segmentation of multi-dimensional images.

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) [14] and Oriented Relative Fuzzy Connectedness (ORFC) [1] are extensions to directed weighted graphs of some methods from the Generalized

Thanks 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).

Graph Cut (GGC) framework [3], including Fuzzy Connectedness [4] and Watersheds [7]. 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 [2].

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 [12], shape constraints [15], boundary polarity [13,14], and hierarchical constraints [11], which allow the customization of the segmentation to a given target object.

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) [9], 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 [8] 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 [8] 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) [6], 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 [10,16]. However, there are still no studies of 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 area constraints in OIFT segmentation of multi-dimensional images, which is useful for getting regions of interest in the image with less user interaction.

2 Background

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

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 $\mathcal{I} \subset \mathbb{Z}^n$, 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). 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 $\omega(s,t)$, 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 $\pi = \langle t_1, t_2, \ldots, t_n\rangle$ is a sequence of adjacent pixels (i.e., $\langle t_i, t_{i+1}\rangle \in \mathcal{A}$, $i = 1, 2, \ldots, n-1$) 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 $\pi_t = \langle t_1, t_2, \ldots, t_n = t\rangle$ is a path with terminus at a pixel t. When we want to explicitly indicate the origin of the path, the notation $\pi_{s\rightsquigarrow t} = \langle t_1 = s, t_2, \ldots, t_n = t\rangle$ may also be used, where s stands for the origin and t for the destination node. 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 $\langle s, t\rangle$.

A *predecessor map* is a function $P\colon \mathcal{V} \rightarrow \mathcal{V} \cup \{nil\}$ that assigns to each pixel t in $\mathcal{V}$ either some other adjacent pixel 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 pixel to nil in a finite number of iterations. For any pixel $t \in \mathcal{V}$, 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 [5,9]. A *connectivity function* 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 τ_t in G. By taking to each pixel $t \in \mathcal{V}$ one optimum path with terminus at t, we obtain the optimum-path value $V_{opt}^f(t)$, which is uniquely defined by $V_{opt}^f(t) = \min_{\forall \pi_t \text{ in } G}\{f(\pi_t)\}$. The *image foresting transform* (IFT) [9] 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 pixel $t \in \mathcal{V}$ 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 [5], otherwise, the paths π_t^P of the returned spanning forest may not be optimum.

The cost of a trivial path $\pi_t = \langle t\rangle$ is usually given by a handicap value $H(t)$. For example, $H(t) = 0$ for all $t \in \mathcal{S}$ and $H(t) = \infty$ otherwise, where $\mathcal{S}$ is a seed set. The costs for non-trivial paths follow a path-extension rule. For example:

$$f_{\max}(\pi_s \cdot \langle s, t\rangle) = \max\{f_{\max}(\pi_s), \omega(s,t)\} \tag{1}$$

In Algorithm 1, the root map R stores the origin of the paths and the path-cost map V converges to V_{opt}^f, when f satisfies the conditions indicated in [5].

Algorithm 1 – IFT ALGORITHM

INPUT:	Image graph $\langle \mathcal{V}, \mathcal{A}, \omega\rangle$, path-cost function f and an initial labeling function $\lambda : \mathcal{V} \rightarrow \{0, \ldots, l\}$.
OUTPUT:	The label map $L : \mathcal{V} \rightarrow \{0, \ldots, l\}$, root map $R : \mathcal{V} \rightarrow \mathcal{V}$, path-cost map $V : \mathcal{V} \rightarrow \mathbb{R}$ and the spanning forest $P : \mathcal{V} \rightarrow \mathcal{V} \cup \{nil\}$.
AUXILIARY:	Priority queue Q, variable *tmp* and an array of status $S : \mathcal{V} \rightarrow \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.

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

2.2 Oriented Image Foresting Transform (OIFT)

In its first version [14], OIFT was built on the IFT framework by considering the following path-cost function in a symmetric digraph with integer weights:

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

$$f_1^{\male}(\pi_{r \rightsquigarrow s} \cdot \langle s,t \rangle) = \begin{cases} \max\{f_1^{\male}(\pi_{r \rightsquigarrow s}), 2 \times w(s,t) + 1\} & \text{if } r \in \mathcal{S}_1 \\ \max\{f_1^{\male}(\pi_{r \rightsquigarrow s}), 2 \times w(t,s)\} & \text{if } r \in \mathcal{S}_0 \end{cases} \tag{2}$$

Later, a second version [13] with a better handling of ties was proposed based on the following path-cost function:

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

$$f_2^{\male}(\pi_{r \rightsquigarrow s} \cdot \langle s,t \rangle) = \begin{cases} \omega(s,t) & \text{if } r \in \mathcal{S}_1 \\ \omega(t,s) & \text{otherwise} \end{cases} \tag{3}$$

The segmented object $\mathcal{O}^P$ by OIFT is defined from the forest P computed by Algorithm 1, with $f_2^{\male}$ (or $f_1^{\male}$), by taking as object pixels the set of pixels that were conquered by paths rooted in $\mathcal{S}_1$ (i.e., $t \in \mathcal{O}^P$ if and only if $R(t) \in \mathcal{S}_1$).

The functions $f_1^{\male}$ and $f_2^{\male}$ are non-monotonically incremental connectivity functions, as described in [13,14]. The optimality of $\mathcal{O}^P$ by OIFT is supported by an energy criterion of cut in graphs involving arcs from object to background pixels $\mathcal{C}(\mathcal{O}^P)$ (outer-cut boundary), according to Theorem 1 from [13,14].

$$\mathcal{C}(\mathcal{O}) = \{\langle s,t \rangle \in \mathcal{A} \mid s \in \mathcal{O} \text{ and } t \notin \mathcal{O}\} \tag{4}$$

$$E(\mathcal{O}) = \min_{\langle s,t \rangle \in \mathcal{C}(\mathcal{O})} \omega(s,t) \tag{5}$$

Theorem 1 (Outer-cut optimality by OIFT). *For two given sets of seeds $\mathcal{S}_1$ and $\mathcal{S}_0$, let $\mathcal{U}(\mathcal{S}_1, \mathcal{S}_0) = \{\mathcal{O} \subseteq \mathcal{V} \mid \mathcal{S}_1 \subseteq \mathcal{O} \subseteq \mathcal{V} \setminus \mathcal{S}_0\}$ denote the universe of all possible objects satisfying the seed constraints. Any spanning forest P computed by Algorithm 1 for function $f_1^{\male}$ (or $f_2^{\male}$) defines a segmented object $\mathcal{O}^P$ that maximizes E (Eq. 5) among all possible segmentation results in $\mathcal{U}$. That is, $E(\mathcal{O}^P) = \max_{\mathcal{O} \in \mathcal{U}(\mathcal{S}_1, \mathcal{S}_0)} E(\mathcal{O})$.*

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 $\mathcal{G}^i = \{P^i, V^i, L^i, R^i\}$, resulting from the *ith* iteration in a sequence of IFTs.

The DIFT algorithm [6,8] allows to efficiently compute a scene $\mathcal{G}^i$ from the previous scene $\mathcal{G}^{i-1}$, a set $\Delta^+_{\mathcal{S}^i} = \mathcal{S}^i \setminus \mathcal{S}^{i-1}$ 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 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 1 shows that the Generalized DIFT (GDIFT) algorithm [6] with $f_2^{\male}$, to differentially compute the sequence $\langle IFT_{(\mathcal{S}^1)}, IFT_{(\mathcal{S}^2)} \rangle$, where $\mathcal{S}^1 = \mathcal{S}_1^1 \cup \mathcal{S}_0^1 = \{a\} \cup \{i, l\}$ and $\mathcal{S}^2 = \mathcal{S}_1^2 \cup \mathcal{S}_0^2 = \{a\} \cup \{i\}$, 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. 1b), but later become frontier nodes, i.e., neighboring nodes of removed trees/subtrees (Fig. 1c) that can be reprocessed in a different order than the original (Fig. 1d). Due to the strictly minor inequality of Line 12 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. 1).

The DIFT algorithms [6,8] do not attempt to address this issue, as they assume that the usage of the "$\leq$" comparison on Line 12 of Algorithm 1 would also be perfectly valid. However, in the case of functions such as $f_2^{\male}$, 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), without worrying about minor topology details of the resulting forest, that are irrelevant to the segmentation task.

The first proposed solution is simply to consider the usage of the Generalized DIFT (GDIFT) algorithm from [6] with the $f_1^{\male}$ path-cost function. Note that $f_1^{\male}$ 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 [6]. Note that problems like the one reported in Fig. 1 do not occur with $f_1^{\male}$, 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.

The second proposed solution is to use Algorithm 2, which considers for each path π_t a lexicographical path-cost function with two components $\langle F_2^{\male}(\pi_t), T(\pi_t)\rangle$, where $F_2^{\male}(\pi = \langle t_1, \ldots, t_n\rangle) = \max_{i=1,2,\ldots,n}\{f_2^{\male}(\langle t_1, \ldots, t_i\rangle)\}$ 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.

Algorithm 2 – Algorithm DOIFT

```
INPUT:     Image graph $G = \langle \mathcal{V}, \mathcal{A}, \omega\rangle$, the set $\Delta_{\mathcal{S}}^{+}$ of seeds for addition, set $\Delta_{\mathcal{S}}^{-}$ 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
           $\lambda : \Delta_{\mathcal{S}}^{+} \rightarrow \{0,1\}$ 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 $Q \leftarrow \varnothing$.
2.  If $\Delta_{\mathcal{S}}^{-} \neq \varnothing$, then
3.      $(L, V, P, Q) \leftarrow$ DOIFT-RemoveSubTrees($G, L, V, P, Q, \Delta_{\mathcal{S}}^{-}$)
4.  For each $s \in \Delta_{\mathcal{S}}^{+}$, do
5.      Set $L(s) \leftarrow \lambda(s)$, $P(s) \leftarrow nil$, $V(s) \leftarrow \langle -1, L(s)+1\rangle$
6.      If $s \notin Q$, then insert $s$ in $Q$.
7.  While $Q \neq \varnothing$, do
8.      Remove $s$ from $Q$ such that $V(s) \overset{\text{lex}}{\leq} V(r)$ for all $r \in Q$.
9.      For each node $t$ such that $\langle s,t\rangle \in \mathcal{A}$, do
10.         Compute $tmp_1 \leftarrow F_2^{\male}(\pi_s^P \cdot \langle s,t\rangle)$.
11.         If $tmp_1 \neq f_2^{\male}(\pi_s^P \cdot \langle s,t\rangle)$, then
12.             Set $tmp_2 \leftarrow V_2(s)$.
```

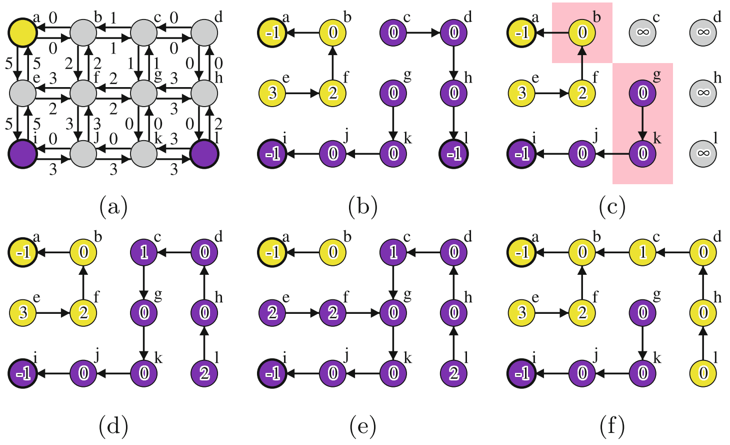


Fig. 1. (a) Input graph with marked seeds $\mathcal{S}_1^1 = \{a\}$ and $\mathcal{S}_0^1 = \{i, l\}$. (b) Initial computed forest by OIFT with $f_2^{\male}$, assuming node b was processed first than node g. The values within the nodes indicate the costs of the paths and the arrows point to the predecessor 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 first than 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 $f_2^{\male}$ with $\mathcal{S}_1^2 = \{a\}$ and $\mathcal{S}_0^2 = \{i\}$.

```
13.         Else If $V_1(s) = tmp_1$, then
14.             Set $tmp_2 \leftarrow V_2(s) + 2$.
15.         Else , then
16.             Set $tmp_2 \leftarrow L(s) + 1$.
17.         If $\langle tmp_1, tmp_2 \rangle \overset{\text{lex}}{<} V(t)$, then
18.             Set $P(t) \leftarrow s$, $V(t) \leftarrow \langle tmp_1, tmp_2 \rangle$, $L(t) \leftarrow L(s)$.
19.             If $t \notin Q$, then insert $t$ in $Q$.
20.         Else If $s = P(t)$, then
21.             If $tmp_1 \neq V_1(t)$ or $tmp_2 > V_2(t)$ or $L(t) \neq L(s)$, then
22.                 $(L, V, P, Q) \leftarrow$ DOIFT-RemoveSubTrees$(G, L, V, P, Q, \{t\})$
23.                 Break; #GOTO LINE 8
```

Procedure DOIFT-RemoveSubTrees in Algorithm 3, releases the entire subtrees, converting its pixels to trivial trees of infinite cost, and transforms all of its neighboring pixels into frontier pixels, inserting them in Q, assuming that the graph is symmetric. It plays the role of both DIFT-RemoveSubTree and DIFT-TreeRemoval from [6], 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 $Q$.
AUXILIARY: Queue $J$ and a set $\mathcal{F}$.

1.  Set $J \leftarrow \varnothing$, $\mathcal{F} \leftarrow \varnothing$.
2.  For each $t \in \mathcal{R}$, do
3.      If $t \in Q$, then remove $t$ from $Q$.
4.      Set $V(t) \leftarrow \langle\infty,\infty\rangle$, $P(t) \leftarrow nil$.
5.      Insert $t$ in $J$.
6.  While $J \neq \varnothing$, do
7.      Remove $s$ from $J$.
8.      For each node $t$ such that $\langle s,t\rangle \in \mathcal{A}$, do
9.          If $s = P(t)$, then
10.             Insert $t$ in $J$.
11.             If $t \in Q$, then remove $t$ from $Q$.
12.             Set $V(t) \leftarrow \langle\infty,\infty\rangle$, $P(t) \leftarrow nil$.
13.         Else If $V(t) \neq \langle\infty,\infty\rangle$ and $t \notin Q$, then
14.             Insert $t$ in $\mathcal{F}$.
15. While $\mathcal{F} \neq \varnothing$, do
16.     Remove $t$ from $\mathcal{F}$.
17.     If $V(t) \neq \langle\infty,\infty\rangle$ and $t \notin Q$, then
18.         Insert $t$ in $Q$.
```

Other differences of Algorithm 2 in relation to GDIFT [6], are the absence of the state map used in [6], which proved to be unnecessary for functions with non-decreasing costs along the paths, as for the lexicographical cost $\langle F_2^{\male}(\pi_t), T(\pi_t)\rangle$, and modifications to avoid using the root map to save memory. Another difference is the inclusion of Line 23 in Algorithm 2, to immediately break the innermost loop, thus avoiding the repeated processing of part of the neighborhood.

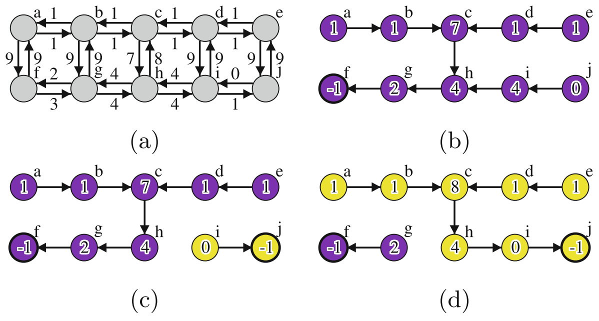


Fig. 2. (a) Input graph. (b) Initial forest by OIFT with $f_2^{\male}$ for $\mathcal{S}^1 = \{f\}$. (c) The updated result by Algorithm 2, as a new object seed j is inserted, so that $\mathcal{S}^2 = \{f, j\}$. The values within nodes reflect the costs of $f_2^{\male}$. (d) The correct result by Proposition 1.

The third proposed version of DOIFT is a variant of the second, modified so that disputed nodes with the same cost 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:

$$F(\pi = \langle t_1, \ldots, t_n \rangle) = \max_{i=1,2,\ldots,n} \{f(\langle t_1, \ldots, t_i \rangle)\} \tag{6}$$

Consider the following lemma:

Lemma 1. *Let P be a predecessor map computed by Algorithm 1. For any two paths $\delta_t^P = \langle t_1, t_2, \ldots, t_n = t \rangle$ and $\tau_s^P = \langle s_1, s_2, \ldots, s_m = s \rangle$, 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 7 of Algorithm 1.*

Proof. Let s_k be a node in τ_s^P, such that $f(\langle s_1, \ldots, s_k \rangle) = F(\tau_s^P)$. From Eq. 6, we have that $f(\langle t_1, \ldots, t_i \rangle) \leq F(\delta_t^P)$, $i = 1, 2, \ldots, n$. From the assumptions of Lemma 1, we may conclude that $F(\delta_t^P) < f(\langle s_1, \ldots, s_k \rangle)$. Thus, $f(\langle t_1, \ldots, t_i \rangle) < f(\langle s_1, \ldots, s_k \rangle)$, $i = 1, 2, \ldots, n$.

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 7). After s_k gets inserted in $\mathcal{Q}$ with cost $V(s_k) = f(\langle s_1, \ldots, s_k \rangle)$, it won't be removed from $\mathcal{Q}$ before all nodes t_i, $i = 1, 2, \ldots, n$, are consecutively processed in $\mathcal{Q}$, with lower costs $V(t_i) = f(\langle t_1, \ldots, t_i \rangle)$. 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 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 12. 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 2 discusses the consequences of Proposition 1 in the differential execution of OIFT. Note that Algorithm 2 does not satisfy Proposition 1. To correct this issue, the condition of Line 17 of Algorithm 2 must be changed to a much more complex condition:

$$tmp_1 < V_1(t) \text{ or } (tmp_1 = V_1(t) \text{ and } ((tmp_2 < V_2(t) \text{ and not} H_2) \text{ or } H_1))$$

where X, H_1 and H_2 are boolean variables defined as:

$$\begin{aligned} X \leftarrow V_1(t) = f_2^{\male}(\pi_s^P \cdot \langle s, t \rangle) > V_1(s) \text{ and } V_1(t) > V_1(P(t)) \\ H_1 \leftarrow P(t) \neq nil \text{ and } X \text{ and } V(s) \overset{\text{lex}}{<} V(P(t)) \\ H_2 \leftarrow P(t) \neq nil \text{ and } X \text{ and } V(s) \overset{\text{lex}}{>} V(P(t)) \end{aligned} \tag{7}$$

With these modifications, we have the third version of the DOIFT algorithm.

4 OIFT with Area/volume Constraint

Let $E_A = \max_{\mathcal{O} \in \mathcal{U}(A,\mathcal{S}_0)} E(\mathcal{O})$ denote the optimum energy value by Eq. 5 of a segmentation by OIFT using set A as internal seeds in Theorem 1. In order to introduce the idea of the incorporation of a size constraint in OIFT, we need first to establish some supporting propositions.

Proposition 2. *The optimum energy $E_{A \cup B}$ among all objects in $\mathcal{U}(A \cup B, \mathcal{S}_0)$, satisfies $E_{A \cup B} = \min\{E_A, E_B\}$.*

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

The proofs of Proposition 2 and 3 are given in [12].

Suppose we want to define an optimal object of maximum energy via OIFT but having area/volume below a given threshold. Let's assume that the defined background must be connected to the originally selected background seeds. If the object has an area above the threshold, we can reduce its size by inserting new background seeds in its boundary. In order to apply Propositions 2 and 3, we can temporarily invert the object and background labels, in order to take advantage of the analogous and symmetrical problem. In this complementary problem, the energies of background nodes could be computed by the IFT with $f_{\max}$ from the object seeds $\mathcal{S}_1$ in the original graph. 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. We can then repeat this procedure until the area of the resulting object falls below the given threshold. We therefore have a sequence of OIFTs for each new seed inserted that can be calculated faster by DOIFT.

5 Experimental Results

Figure 3 shows the experimental curves for the segmentation of the talus bone using 40 slices from MR images of the foot using a robot user. In the first row, the arc weights were defined as $\omega(s,t) = |I(t) - I(s)|$ and with boundary polarity parameter defined as -50% (see [14]). 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 $f_2^{\male}$ 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 $f_2^{\male}$. 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.

We also carried out experiments in a 3D MR image. We consider the accumulated time over the iterations of the automatic seed selection via the area

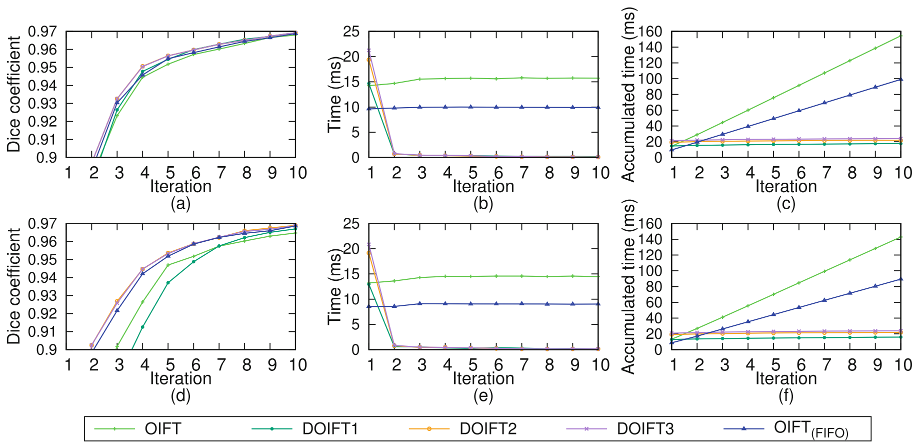
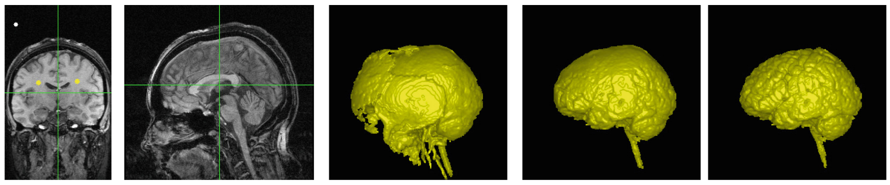


Fig. 3. The mean curves of accuracy, time, and accumulated time.



(a) coronal (b) sagittal (c) Unconstrained (d) $T_a = 10,000$ (e) $T_a = 9,000$

Fig. 4. Brain segmentation in MR images. Only three markers were selected in the indicated coronal slice.

procedure described in Sect. 4 to segment the brain. We considered a region adjacency graph of supervoxels by [16] with an average of 100 voxels per region. Figure 4 shows the results obtained for different values of the maximum volume threshold T_a, which is expressed in number of supervoxels. Regarding the execution time, for $T_a = 9,000$ we had 1 s for the differential flow by $DOIFT_2$ and 75 s for the sequential flow by OIFT with heap. For $T_a = 10,000$, we had 0.86 s for $DOIFT_2$ and 64 s for OIFT.

6 Conclusion

We have successfully tested different approaches to implement the differential OIFT and its use in implementing an area/volume constraint in OIFT. The use of area constraints can help to improve segmentation considerably, without the need to select multiple markers. As future works we intend to evaluate other applications for DOIFT and to create a hierarchy of OIFT segmentations by varying the area threshold.

References

1. Bejar, H.H., Miranda, P.A.: Oriented relative fuzzy connectedness: theory, algorithms, and its applications in hybrid image segmentation methods. EURASIP J. Image Video Process. **2015**(21) (2015)
2. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vision **70**(2), 109–131 (2006)
3. Ciesielski, K., Udupa, J., Falcão, A., Miranda, P.: A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them. In: Proceedings of SPIE on Medical Imaging: Image Processing, vol. 8314 (2012)
4. Ciesielski, K., Udupa, J., Saha, P., Zhuge, Y.: Iterative relative fuzzy connectedness for multiple objects with multiple seeds. Comput. Vision Image Underst. **107**(3), 160–182 (2007)
5. Ciesielski, K.C., Falcão, A.X., Miranda, P.A.V.: Path-value functions for which Dijkstra's algorithm returns optimal mapping. J. Math. Imaging Vision **60**(7), 1025–1036 (2018)
6. Condori, M.A., Cappabianco, F.A., Falcão, A.X., Miranda, P.A.: An extension of the differential image foresting transform and its application to superpixel generation. J. Visual Commun. Image Represent. **71**, 102748 (2020)
7. 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)
8. Falcão, A.X., Bergo, F.P.: Interactive volume segmentation with differential image foresting transforms. IEEE Trans. Med. Imaging **23**(9), 1100–1108 (2004)
9. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: theory, algorithms, and applications. IEEE TPAMI **26**(1), 19–29 (2004)
10. 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)
11. Leon, L.M., Ciesielski, K.C., Miranda, P.A.: 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
12. 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)
13. Mansilla, L., Miranda, P.: 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)
14. Miranda, P., Mansilla, L.: Oriented image foresting transform segmentation by seed competition. IEEE Trans. Image Process. **23**(1), 389–398 (2014)
15. de Moraes Braz, C., Miranda, P.A., Ciesielski, K.C., Cappabianco, F.A.: Optimum cuts in graphs by general fuzzy connectedness with local band constraints. J. Math. Imaging Vision **62**, 659–672 (2020)
16. 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)