How to complete any segmentation process interactively via Image Foresting Transform

Paulo A.V. Miranda, Alexandre X. Falcão, Guilherme C.S. Ruppert
Information Systems Department
LIV – Institute of Computing – Unicamp
CP 6176, 13084-971, Campinas, SP, Brazil
Email: {pavm,afalcao,ruppert}@ic.unicamp.br

***Abstract*—The segmentation of poorly defined structures in medical imaging and heterogeneous objects in natural images usually call for considerable user assistance. Consequently, automatic results are often far from desirable and interactive repairs become an essential feature to consider. However, how to import automatic results obtained from external processes and complete their segmentation interactively is an issue, since different tools are based on different optimization criteria.**

Another simpler related problem concerns how to continue a previous segmentation obtained by the same interactive tool. This ability to stop and later resume interactive segmentation sessions is specially important for tridimensional images and video. However, very often crucial data (e.g., the history of user input) are no longer available; or are no longer reliable, as consequence of some post-processing.

How to offer a comprehensive recovery and resume capability, comprising all these different scenarios, under the framework of the "Image Foresting Transform" (IFT) is the central focus of this paper.

***Keywords*-image foresting transform; robustness in image segmentation; and graph search algorithms;**

I. Introduction

In image processing and computer vision, there are several situations in which user interaction becomes essential in obtaining effective image segmentation. The high-level, application domain specific knowledge of the user is often required in medical image analysis [1] because of poorly defined structures, and in the digital matting of natural scenes [2], because of their heterogeneous nature.

As a consequence, automatic segmentation techniques always tend to present some sort of errors and may even fail under critical circumstances. Hence, without the necessary corrections by edition, the results generated by automatic segmentation tools may become inappropriate in any more rigorous study (e.g., medical image analysis). Manual editing can always be adopted to make corrective repairs, since it does not depend on any additional data besides the segmentation mask, but it is an extremely time-consuming and tedious task. On the other hand, interactive tools usually depend on the whole history of user input. Thus, in order to interactively correct an arbitrary segmentation, we must first solve an inverse problem, i.e., how to guess the missing user input from the given segmentation result.

Another simpler related problem concerns the availability of means to resume a previous segmentation session in a specific interactive tool. In operator-assisted segmentation tools, the user usually adds/removes markers (seed pixels, anchor boundary points) for recognition, while subsequent delineation is performed by the computer in interactive time. Accuracy becomes a compromise between the user's patience for verification and correction, and the quality of delineation. In practice, the user tends to stop the corrective actions when the efforts needed to improve the results increase too much relative to the returned improvement in accuracy. In the context of 3D medical image and video segmentation, the user is tempted to stop even earlier due to the weariful and hard work. In this sense, it is highly desirable that segmentation tools provide the ability to stop and later resume segmentation sessions at will.

However, many times crucial data like the history of user inputs (e.g., the sequence of selected markers) are no longer available or valid. Usually, such features are not supported by open file formats and are encoded only in proprietary formats, or are stored in separate files, potentially leading to consistency problems while transferring data to different locations. Moreover, the image may have been altered by filtering (e.g., Gaussian blur, radiometric transformation), or by some spatial transformation (e.g., interpolation, registration) after the last interactive session. As a consequence, the task of making corrections in existing databases becomes complex.

Discrete Mathematics provides an elegant framework for image processing, rich of efficient algorithms with proofs of correctness. As a consequence, many image segmentation methods have been modeled as graph-search problems [2], [3], [4], [5]. In these approaches, a graph derived from the image is computed and the user indicates hard constraints by selecting some of its nodes as seeds. An optimal graph partition satisfying this supplied set of constraints is computed and displayed. Corrections can then be performed by new seed addition or seed removal.

Under this scenario, we have the *image foresting transform* (IFT) [4] — a tool for the design of image processing operators based on *connectivity functions* (path-value func-

tions) in graphs derived from the image. The IFT algorithm minimizes/maximizes a connectivity map by taking into account all paths with terminus at each pixel, such that an optimum-path forest is computed from the graph.

More recently, it was shown that IFT segmentation methods can lead to a minimum cut in the graph according to some appropriate graph-cut measures [5]. Indeed, it was shown that procedures adopted to circumvent the existing bias of the *min-cut/max-flow* algorithm [3], [6] lead to approximations of the IFT with internal and external seed competition [5]. Hence, the IFT with seed competition (IFT-SC) is our option of choice in this work. It is especially suitable for our purposes, since it allows corrections to be performed in sublinear time by its differential version [1].

The central focus of this paper is to devise a comprehensive recovery and resume capability, comprising all aforementioned cases, under the framework of the IFT [4]. However, as already mentioned, in order to complete an arbitrary segmentation we have to first find a suitable set of seeds that assembles it. This could be accomplished by the method described in [7], which computes a set with minimum number of seeds under some constraints. However, it usually places too many seeds along weak and ambiguous segments of the boundary, making further corrections to behave almost manual over these regions. In the present paper, we theoretical extend these results to a more general and flexible case leading to fewer seeds.

For the sake of completeness in presentation, Sections II and III include an overview of concepts on image graph and a revision of the IFT. Section IV shows how to complete any segmentation process interactively using the IFT. Then, Section V treats some particular cases about how to resume a previous segmentation by IFT when more information is available. Our conclusions are stated in Section VI.

II. Basic Concepts on Image Graphs

A multi-dimensional and multi-spectral image $\hat{I}$ is a pair $(\mathcal{I}, \vec{I})$ where $\mathcal{I} \subset Z^n$ is the image domain and $\vec{I}(t)$ 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 *adjacency relation* $\mathcal{A}$ is a binary relation on $\mathcal{I}$. We use $t \in \mathcal{A}(s)$ and $(s,t) \in \mathcal{A}$ to indicate that t is adjacent to s. Once the adjacency relation $\mathcal{A}$ has been fixed, the image $\hat{I}$ can be interpreted as a graph $(\mathcal{I}, \mathcal{A})$ whose nodes (or vertices) are the image pixels in $\mathcal{I}$ and whose arcs are the pixel pairs (s,t) in $\mathcal{A}$. In this work, we are interested in irreflexive and symmetric relations. For example, one can take $\mathcal{A}$ to consist of all pairs of pixels (s,t) in the Cartesian product $\mathcal{I} \times \mathcal{I}$ such that $d(s,t) \leq \rho$ and $s \neq t$, where $d(s,t)$ denotes the Euclidean distance and ρ is a specified constant (e.g., 4-neighborhood, when $\rho = 1$, and 8-neighborhood, when $\rho = \sqrt{2}$, in case of 2D images).

Each arc $(s,t) \in \mathcal{A}$ has a fixed weight $w(s,t) \geq 0$ which may be computed from local image and object properties extracted from $\vec{I}$ [8]. In this work, higher arc weights across the object's boundary (i.e., a dissimilarity measure between pixels s and t) will be considered without loss of generality. For example, one may use the mean gradient magnitude (i.e., $\frac{I(s)+I(t)}{2}$ for a gradient image $\hat{I}$). We also consider only undirected and weighted graphs. That is, the adjacency relation is symmetric and $w(s,t) = w(t,s)$ for all $(s,t) \in \mathcal{A}$.

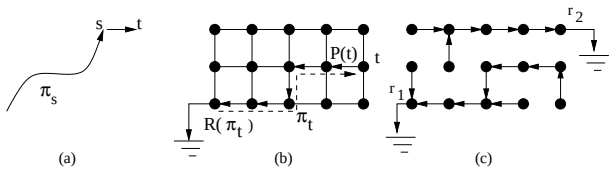

Figure 1. (a) Path $\pi_t = \pi_s \cdot \langle s,t \rangle$ indicates the extension of path π_s by an arc (s,t). (b) A 4-neighborhood graph showing a path π_t (dashed line) represented in backwards, where $P(t)$ is the predecessor node of t and $R(\pi_t)$ is the root pixel. (c) A spanning forest P with two root nodes, r_1 and r_2.

For a given image graph $(\mathcal{I}, \mathcal{A})$, a path $\pi_t = \langle t_1, t_2, \ldots, t \rangle$ is a sequence of adjacent pixels with terminus at a pixel t. A path is *trivial* when $\pi_t = \langle t \rangle$. A path $\pi_t = \pi_s \cdot \langle s,t \rangle$ indicates the extension of a path π_s by an arc (s,t) (Figure 1a). All paths considered in this work are simple paths, that is, paths with no repeated vertices (pixels).

A *predecessor map* is a function P that assigns to each pixel t in $\mathcal{I}$ either some other adjacent pixel in $\mathcal{I}$, or a distinctive marker nil not in $\mathcal{I}$ — in which case t is said to be a *root* of the map. A *spanning forest* is a predecessor map which contains no cycles — i.e., one which takes every pixel to nil in a finite number of iterations (Figures 1b and 1c, where $R(\pi_t)$ is a root node and $P(t)$ is the predecessor node of t in the path π_t). For any pixel $t \in \mathcal{I}$, a spanning forest P defines a path π_t recursively as $\langle t \rangle$ if $P(t) = nil$, and $\pi_s \cdot \langle s,t \rangle$ if $P(t) = s \neq nil$.

III. Image Foresting Transform (IFT)

For purposes of completeness in the presentation, several concepts introduced in [5] are included in this section. However, differently from [5], the IFT is presented here in its equivalent dual form, in accordance with the original IFT paper [4].

A *connectivity function* computes a value $f(\pi_t)$ for any path π_t, usually based on arc weights. Let $\Pi(\mathcal{I}, \mathcal{A}, t)$ be the set of all paths in the graph $(\mathcal{I}, \mathcal{A})$ with terminus at t. In this work, a path is *optimum* according to the following definition.

Definition 1 (Optimum path): A path π_t is *optimum* if $f(\pi_t) \leq f(\tau_t)$ for any other path $\tau_t \in \Pi(\mathcal{I}, \mathcal{A}, t)$.

By taking to each pixel $t \in \mathcal{I}$ one optimum path with terminus t, we obtain the optimum-path value $V(t)$, which

is uniquely defined by

$$V(t) \quad = \quad \min_{\forall \pi_t \in \Pi(\mathcal{I},\mathcal{A},t)} \{f(\pi_t)\}. \qquad (1)$$

The *image foresting transform* (IFT) algorithm solves the above optimization problem by dynamic programming [4]. The IFT takes an image $\hat{I}$, a path-value function f and an adjacency relation $\mathcal{A}$; and assigns one optimum path π_t to every pixel $t \in \mathcal{I}$ such that an *optimum-path forest* P is obtained — i.e., a spanning forest where all paths are optimum. However, f must be *smooth*, that is, it must satisfy Definition 2, as demonstrated in [4]. The attributes of the forest include the map V, the roots $R(\pi_t)$, root labels $L(t)$, and the predecessor $P(t)$ of t in the optimum path. The image operators are then reduced to a local processing of these attributes [4].

Definition 2 (Smooth path-value function): A path-value function f is *smooth* if for any pixel $t \in \mathcal{I}$, there is an optimum path π_t which either is trivial, or has the form $\tau_s \cdot \langle s,t \rangle$ where
(C1) $f(\tau_s) \leq f(\pi_t)$,
(C2) τ_s is optimum,
(C3) for any optimum path τ'_s, $f(\tau'_s \cdot \langle s,t \rangle) = f(\pi_t)$.

An interesting property of an optimum-path forest is that any path starting in a root node is also a complete optimum path (path-value function must be *smooth*), according to the following definition.

Definition 3 (Complete optimum path): A path $\pi_{t_n} = \langle t_1, t_2, \ldots, t_n \rangle$ is *complete optimum* if all paths $\pi_{t_i} = \langle t_1, t_2, \ldots, t_i \rangle$, $i = 1, 2, \ldots, n$ are optimum paths.

From this point on, we will use the notation $\bar{\pi}_t$ when we want to explicitly refer to a complete optimum path π_t.

A. General IFT Algorithm

Algorithm 1 obtains an optimum-path forest P, in which all paths satisfy conditions $(C1) - (C3)$, by minimizing a smooth path-value funtion f.

Algorithm 1: – GENERAL IFT ALGORITHM

INPUT: *Image $\hat{I} = (\mathcal{I}, \vec{I})$, adjacency $\mathcal{A}$, and path-value function f.*
OUTPUT: *Optimum-path forest P, the minimum path-value map V and label map L.*
AUXILIARY: *Priority queue Q, variable tmp, and an array of $status$.*

1. **For each** $t \in \mathcal{I}$, **do**
2. Set $P(t) \leftarrow nil$ and $V(t) \leftarrow f(\langle t \rangle)$.
3. Set $status(t) \leftarrow 0$.
4. **If** $V(t) \neq +\infty$, **then** insert t in Q.
5. **While** $Q \neq \emptyset$, **do**
6. Remove s from Q such that $V(s)$ is minimum.
7. Set $status(s) \leftarrow 1$.
8. **For each** $t \in \mathcal{A}(s)$, such that $status(t) = 0$, **do**
9. Compute $tmp \leftarrow f(\pi_s \cdot \langle s,t \rangle)$.
10. **If** $tmp < V(t)$, **then**
11. **If** $V(t) \neq +\infty$, **then** remove t from Q.
12. Set $P(t) \leftarrow s$, $V(t) \leftarrow tmp$.
13. $L(t) \leftarrow L(s)$ and insert t in Q.

Line 1 initializes maps and inserts pixels with finite trivial-path values in Q. The minima of the initial map V compete with each other and some of them become roots of the forest. The main loop computes optimum paths from the minima to every pixel s in a non-decreasing order of value (Lines 2–8). At each iteration, a path π_s of minimum value $V(s)$ is obtained in P when we remove its last pixel s from Q (Line 3). The rest of the lines evaluate if the path $\pi_s \cdot \langle s,t \rangle$ that reaches an adjacent pixel t through s is cheaper than the current path π_t in P and update Q, $V(t)$, $L(t)$, and $P(t)$ accordingly.

B. IFT segmentation with internal and external seeds

Although the results of this paper can be extended to multiple objects, we will focus on binary image segmentation for sake of simplicity. A binary segmentation of an image is represented by a labeled image $\hat{L} = (\mathcal{I}, L)$, where $L(t) = 1$ for object pixels and $L(t) = 0$ for background pixels. Hence, a binary segmentation corresponds to an image partition into two disjoint sets $\mathcal{O}_{\hat{L}} = \{t \in \mathcal{I} \mid L(t) = 1\}$ and $\mathcal{B}_{\hat{L}} = \{t \in \mathcal{I} \mid L(t) = 0\}$ representing object and background respectively. Each segmentation also defines an *induced cut boundary* $\mathcal{C}$ in the graph, which is the set $\mathcal{C}$ of arcs (s,t) such that $L(s) = 1$ and $L(t) = 0$.

We consider image segmentation from two seed sets, $\mathcal{S}_o$ and $\mathcal{S}_b$ ($\mathcal{S}_o \cap \mathcal{S}_b = \emptyset$), containing pixels selected inside and outside the object, respectively. A feasible segmentation must satisfy these sets of hard constraints (i.e., $L(t) = 1$ for all $t \in \mathcal{S}_o$ and $L(t) = 0$ for all $t \in \mathcal{S}_b$).

We are interested in a particular case of smooth path-value functions, the monotonically incremental path-value function $f_{\max}$. This function basically assigns to any path π_t the maximum arc-weight along π_t. Equation 2 presents it in the recursive form.

$$\begin{aligned} f_{\max}(\langle t \rangle) \quad &= \quad \begin{cases} -1 & \text{if } t \in \mathcal{S}_o \cup \mathcal{S}_b \\ +\infty & \text{otherwise} \end{cases} \\ f_{\max}(\pi_s \cdot \langle s,t \rangle) \quad &= \quad \max\{f_{\max}(\pi_s), w(s,t)\}, \end{aligned} \qquad (2)$$

Note that the search for optimum paths is constrained to start in $\mathcal{S}_o \cup \mathcal{S}_b$ (roots by imposition).

The internal and external seeds compete with each other for their most strongly connected pixels, such that the image is partitioned into two optimum-path forests — one rooted at the internal seeds, defining the object, and the other rooted at the external seeds, representing the background [1]. This method is sometimes referenced as IFT-SC (*IFT segmentation by Seed Competition*) [5]. Both the internal and external forests are encoded on the same predecessor map P returned

by the IFT. The segmentation $\hat{L}$ is defined as follows, where π_t is the optimum path with terminus t obtained from P.

$$L(t) = \begin{cases} 1 & \text{if } R(\pi_t) \in \mathcal{S}_o, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In fact, Algorithm 1 is already propagating the root labels to all graph nodes. Hence, we only have to set $L(t) = 1$ for all $t \in \mathcal{S}_o$ and $L(t) = 0$ for all $t \in \mathcal{S}_b$ before calling the algorithm.

As observed in reference [4], the optimum-path forest may not be unique. For example, if all paths have the same value, then any spanning forest will be optimum. Ties between paths π_t and τ_t from seeds $s_1 = R(\pi_t)$ and $s_2 = R(\tau_t)$ with the same label ($\{s_1, s_2\} \subset \mathcal{S}_o$ or $\{s_1, s_2\} \subset \mathcal{S}_b$) are never a problem, since they lead to exactly the same final segmentation result $\hat{L}$. Hence, any solution in this case is satisfactory. However, a special care has to be taken in the case of seeds with different labels, which constitute the basis of the real *tie zones* as follows.

Definition 4 (Tie-zone pixel): A pixel t is a *tie-zone pixel* if there exist two complete optimum paths π_t and τ_t such that $R(\pi_t) \in \mathcal{S}_b$ and $R(\tau_t) \in \mathcal{S}_o$.

In many implementations of Algorithm 1, ties are usually broken in Q using first-in-first-out (FIFO) policy. That is, when two optimum paths reach an ambiguous pixel s with the same minimum value, s is assigned to the first path that reached it [4]. But if we simply assign a fixed label (1 or 0) to all tie-zone pixels then a theorem stated in [5] holds. For our purposes, we just present here a particular version, although the theorem as proved in [5] is more general.

Theorem 1 (Optimum-path forest cut in IFT-SC): Any segmentation $\hat{L}$ defined by an optimum-path forest with path-value function $f_{\max}$ and with a single label value for all tie-zone pixels (Definition 4) maximizes the graph-cut measure E defined by Equation 4 among all possible segmentation results satisfying the hard constraints.

$$E(\hat{L}) = \min_{\forall (s,t) \in \mathcal{A} \mid L(s)=1, L(t)=0} w(s,t) \quad (4)$$

Another important result presented in [5] concerns the proof that the cut boundaries obtained by the IFT-SC are also piecewise optimum. That is, under the same conditions of Theorem 1, any part of a cut boundary is chosen as one that maximizes its minimum value E.

IV. How to complete any segmentation via IFT

Let $\hat{L}^*$ be a general segmentation obtained by any method with no additional information. We may start computing the image graph as usually, but the seeds are not known. Hence, to complete this segmentation via IFT, we first have to find a suitable set of seeds. These seeds must restore an optimum-path forest in a manner consistent with the segmentation $\hat{L}^*$. Once this is done, the corrections can then be done in sublinear time by using the *Differential IFT* algorithm [1].

A trivial solution to this problem would be simply to select all object pixels as being internal seeds (i.e., $t \in \mathcal{S}_o$ if $L^*(t) = 1$), and all background pixels as external seeds (i.e., $t \in \mathcal{S}_b$ if $L^*(t) = 0$). However the corrections in this case would degenerate into a manual segmentation process. Note that, in order to add new hard constraints for correction, other old constraints would necessarily be overwritten because $\mathcal{S}_o \cap \mathcal{S}_b = \emptyset$. Since we also have $\mathcal{S}_o \cup \mathcal{S}_b = \mathcal{I}$, the execution of Algorithm 1 would not change anything in the results.

Therefore we have to be as less invasive as possible in the choice of seeds, in order to avoid restricting too much the action of the algorithm during the corrections. So the best solution in theory is to choose a set of seeds with minimum cardinality that generates the same segmentation result. The work in [7] solves this problem in a particular case assuming that there are no tie-zones. We theoretically extend this result to two kinds of tie-breaking policies:

(P1) all tie-zone pixels are assigned to the object,
(P2) all tie-zone pixels are assigned to the background.

Thus, since the solution proposed in this work is more flexible, it usually produces fewer seeds than in [7]. That is, it is always possible to eliminate ties at the price of adding more seeds. Also note that under the conditions (P1-P2) we have that Theorem 1 is valid, and therefore many intermediate results will be explained here based on this concept of graph-cut energy introduced in [5].

In order to start we first must introduce the notion of *seed robustness*. The problem is to find the regions where seeds can be moved without altering the segmentation. In the case of tie-breaking policies last-in-first-out (LIFO) or first-in-first-out (FIFO), this problem becomes very complicated, because the results may vary depending on the order that the data is processed. Hence, in this case any movement of a seed connected to tie-zone regions can affect the results. However, the tie-breaking policy P1 (or P2) always leads to an unique result allowing us to devise a theoretical analysis.

The regions, where the seeds are free to move, are called in some works as the *cores* [7]. But since the approach here is more flexible, the cores as presented here will be eventually bigger than in [7].

Without loss of generality, we will constrain the analysis of robustness only to internal seeds, being the external seeds a completely symmetric problem. We will use the notation $\hat{L}_A$ to refer to a segmentation obtained using as internal seeds only the elements in a set A. In order to define our cores, we must first introduce the notions of seed equivalence and redundancy.

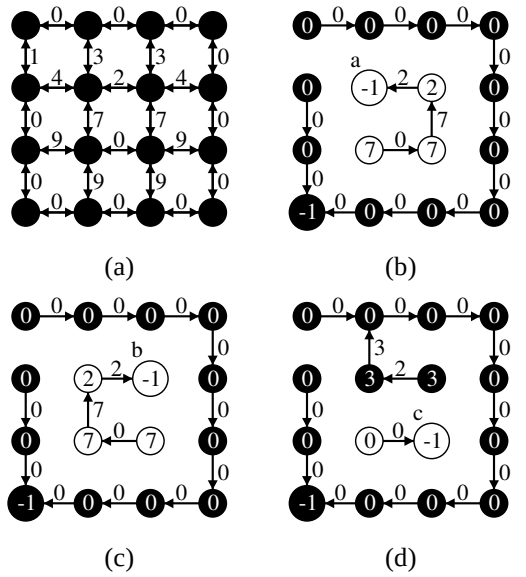

Figure 2. (a) A 4-neighborhood graph, where the numbers indicate the arc weights. (b-d) Results by IFT using $f_{\max}$ (Eq. 2) for different object's seeds in $\mathcal{S}_o = \{a, b, c\}$ and one fixed external seed (bigger black dot). The numbers inside the nodes indicate the values $V(t)$ (Eq. 1). Note that a and b are equivalent, and c is redundant in relation to a and b.

Definition 5 (Equivalent seeds): Two internal seeds s_1 and s_2 are said *equivalent* if they separately produce the same result. That is, for the given external seed set $\mathcal{S}_b$, the result $\hat{L}_{\{s_1\}}$ obtained from $\mathcal{S}_o = \{s_1\}$ is the same as $\hat{L}_{\{s_2\}}$ obtained from $\mathcal{S}_o = \{s_2\}$.

Definition 6 (Redundant seeds): Let $\hat{L}$ be a segmentation result by IFT-SC (under P1 or P2) with internal and external seeds given by $\mathcal{S}_o$ and $\mathcal{S}_b$. An internal seed set $B \subset \mathcal{S}_o$ is said *redundant* in relation to $\mathcal{S}_o \backslash B$ if it can be eliminated without affecting the segmentation $\hat{L}$. That is, the result $\hat{L}_B$ obtained using only B as internal seeds is contained within $\hat{L}$ which can be obtained by using as internal seeds the elements in $\mathcal{S}_o \backslash B$ (i.e., $\mathcal{O}_{\hat{L}_B} \subseteq \mathcal{O}_{\hat{L}}$).

If a seed $s_1 \in \mathcal{S}_o$ is equivalent to another seed $s_2 \in \mathcal{S}_o$ then $\{s_1\}$ is necessarily redundant in relation to $\mathcal{S}_o \backslash \{s_1\}$. But a redundant seed s_3 is not necessarily equivalent to any other seed in $\mathcal{S}_o$ (Figure 2).

In order to better understand the idea of redundancy, let's consider an example. Let $\hat{L}_A$ be the segmentation obtained using only A as internal seeds. Any additional set of object's seeds B selected inside the object's mask $\hat{L}_A$ (i.e., $B \subseteq \mathcal{O}_{\hat{L}_A}$) won't change the result (i.e., B is redundant in relation to A, see Figure 3). This is a direct consequence of the optimization of the the graph-cut measure (Theorem 1) under the condition P1 (or P2) which provides a single solution. The universe of possible cut boundaries using the set $A \cup B$ as internal seeds is more restricted and is contained within

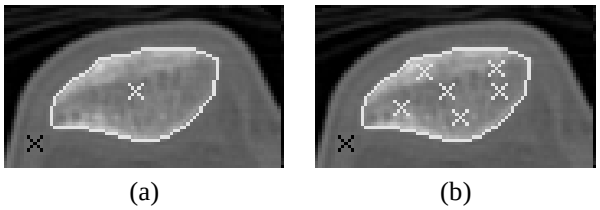

Figure 3. (a) A CT image of a patella segmented by IFT. (b) New seeds added to the object are all redundant under tie-breaking policy P1 (or P2).

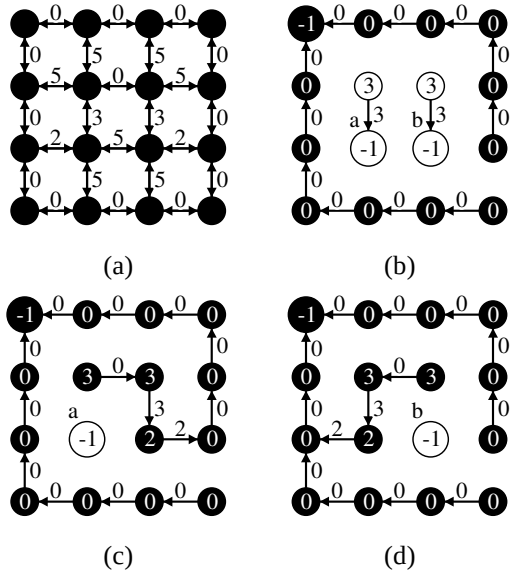

Figure 4. (a) A 4-neighborhood graph. (b) Example of an IFT with P2 policy showing the combined result of the seeds a and b. (c-d) The results of a and b separately.

the search space from A. Thus, the optimum cut boundary for $A \cup B$ can not be better than the best obtained for A. Therefore, since the solution $\hat{L}_A$ is feasible in relation to $A \cup B$ (i.e., $L_A(t) = 1$ for all $t \in A \cup B$) we have that it is also optimum for $A \cup B$.

Now to prove that $\mathcal{O}_{\hat{L}_B} \subseteq \mathcal{O}_{\hat{L}_{A \cup B}}$, we must only note that, by removing internal seeds the optimum-path values (Eq. 1) from object's seeds may only get worse. Hence, a segmentation with fewer seeds may only shrink under P1 (or P2). Figure 4 shows that, the union of the objects obtained for each internal seed separately may be smaller than the combined result of all internal seeds at once (i.e., $\bigcup_{s \in \mathcal{S}_o} \mathcal{O}_{\hat{L}_{\{s\}}} \subseteq \mathcal{O}_{\hat{L}_{\mathcal{S}_o}}$).

The notion of equivalent seeds introduced by Definition 5 is a binary relation $\equiv$ on the set $\mathcal{O}_{\hat{L}}$, i.e., $s_1 \equiv s_2$ if and only if s_1 and s_2 are equivalent. This relation is reflexive, symmetric and transitive, hence, it is indeed an *equivalence relation* as defined in mathematics. Therefore, the *core* of a seed s_1 is in fact the *equivalence class* of s_1 under $\equiv$, denoted $[s_1]$, which is defined as $[s_1] = \left\{t \in \mathcal{O}_{\hat{L}} \mid s_1 \equiv t\right\}$.

The notion of redundant seeds introduced by Definition 6

leads to a binary relation $\propto$ on the powerset $\mathcal{P}(\mathcal{O}_{\hat{L}})$ (i.e., the set of all subsets of $\mathcal{O}_{\hat{L}}$). In other words, we have $A \propto B$ if and only if the seed set A is redundant in relation to the seed set B. This relation is transitive, i.e., if $A \propto B$ and $B \propto C$ then we have that $A \propto C$. This is easy to verify, since $\mathcal{O}_{\hat{L}_A} \subseteq \mathcal{O}_{\hat{L}_B}$ and $\mathcal{O}_{\hat{L}_B} \subseteq \mathcal{O}_{\hat{L}_C}$ implies that $\mathcal{O}_{\hat{L}_A} \subseteq \mathcal{O}_{\hat{L}_C}$. Note also that, mutually redundant seeds are equivalent. That is, if $\{s_1\} \propto \{s_2\}$ and $\{s_2\} \propto \{s_1\}$ then we have that $s_1 \equiv s_2$. This is true because $\hat{L}_{\{s_1\}}$ must be contained in $\hat{L}_{\{s_2\}}$ and vice-versa. On a more general way, we also have that any cycle of redundancies implies in equivalence.

Next, we give a formal definition of redundancy and equivalence in terms of the values of optimum paths linking seeds and their energies. Let E_A be the value of the graph-cut energy $E(\hat{L}_A)$ (Eq. 4) where $\hat{L}_A$ is the segmentation obtained by Algorithm 1 with $f_{\max}$ under condition P1 (or P2), $\mathcal{S}_o = A$ and $\mathcal{S}_b$ fixed. If $A = \{s_1\}$, we have $E_{\{s_1\}}$ which is said to be the energy of the internal seed s_1.

The energy of a set $E_{A\cup B}$ is always less than or equal to the energies of the individual parts E_A and E_B. This is a direct consequence of the optimization of the the graph-cut measure, because an optimum cut in a smaller search space with more constraints (i.e., seeds) can not generate better energy values than a search in a larger space with fewer constraints. In fact, it is possible to prove that $E_{A\cup B} = \min(E_A, E_B)$.

If two internal seeds s_1 and s_2 are equivalent (i.e., $s_1 \equiv s_2$), by the Definition 5 we have that $E_{\{s_1\}} = E_{\{s_2\}}$. But this condition alone can not guarantee that $s_1 \equiv s_2$. Let's define the *pass-value* between two nodes a and b as

$$f(a,b) = \min_{\forall \pi : \pi = \langle t_1 = a, \ldots, t_n = b \rangle} \left\{ \max_{i=1,\ldots,n-1} w(t_i, t_{i+1}) \right\}. \quad (5)$$

If the best path between s_1 and s_2 has pass-value lower than $E_{\{s_1\}} = E_{\{s_2\}}$, then this implies that s_1 and s_2 are equivalent.

$$f(s_1, s_2) < E_{\{s_1\}} = E_{\{s_2\}} \quad \Rightarrow \quad s_1 \equiv s_2 \quad (6)$$

The value $E_{\{s_1\}}$ is the boundary energy of the best cut that separates s_1 from the background seeds. If $f(s_1, s_2) < E_{\{s_1\}}$, then s_2 is certainly inside the object obtained from s_1 (i.e., $L_{\{s_1\}}(s_2) = 1$). Hence, from the discussion of Figure 3, we may conclude that $\{s_2\} \propto \{s_1\}$. Since we also have $f(s_1, s_2) < E_{\{s_2\}}$, then by similar arguments we have $\{s_1\} \propto \{s_2\}$. Therefore, $s_1 \equiv s_2$ as we wanted to prove.

However, the converse of Equation 6 may not be true under tie-breaking policy P1 (Figure 5). That is why the cores may be bigger here than in [7]. In this sense, the IFT-SC under P1 (or P2) can not be less robust than the method RFC [9] discussed in [7].

Since mutually redundant seeds are equivalent, in order to fully characterize the seed equivalence relation we must

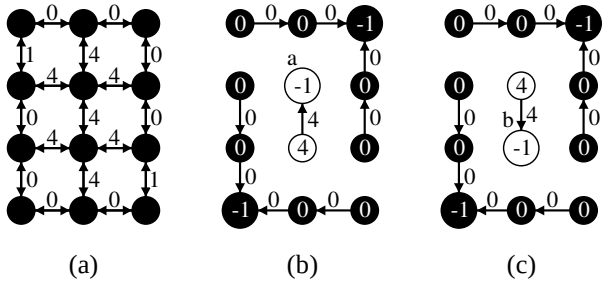

Figure 5. (a) A 4-neighborhood graph. (b-c) The results using P1 policy for the seeds a and b, respectively. (d) Note that they are equivalent, although $f(a,b) = E_{\{a\}} = E_{\{b\}} = 4$.

first formally define the notion of redundancy. From the Definition 6, we have that if set B is redundant in relation to a set A (i.e., $B \propto A$) then $\hat{L}_A = \hat{L}_{A\cup B}$. Thus, $E_A = E_{A\cup B} = \min(E_A, E_B)$. Therefore, we have that $E_B \geq E_A$. But this condition alone is not sufficient to prove the redundancy. If a set is redundant, then all its elements are too. Hence, we may restrict our attention only to sets of individual seeds (e.g., $\{s_1\}$).

Let $f^{\mathcal{O}_{\hat{L}}}(a,b)$ be the pass-value between two nodes a and b in the subgraph induced by $\mathcal{O}_{\hat{L}}$, or $+\infty$ if a and b are not connected in this subgraph. In other words, $f^{\mathcal{O}_{\hat{L}}}(a,b)$ is the best value of a path interconnecting nodes a and b with the constraint of having to pass entirely inside the object. We can extend this concept, and define the constrained pass-value from a set to a node as follows

$$F^{\mathcal{O}_{\hat{L}}}(A,b) \quad = \quad \min_{\forall a \in A} \left\{ f^{\mathcal{O}_{\hat{L}}}(a,b) \right\}. \quad (7)$$

In the case of condition P1, if $\hat{L} = \hat{L}_{A\cup\{b\}}$ then

$$F^{\mathcal{O}_{\hat{L}}}(A,b) \leq E_{\{b\}} \quad \Leftrightarrow \quad \{b\} \propto A. \quad (8)$$

Equation 8 with strictly decreasing inequality falls in a particular case, very similar to Equation 6, where b is equivalent to some node of A. In case of equality, the energy $E_{\{b\}}$ says that, in the best scenario, the background seeds will reach $\mathcal{O}_{\hat{L}_{\{b\}}}$ with this path value. Hence, since the pass-value from A has the same value, then by the policy P1 we have that $\{b\}$ will be part of $\mathcal{O}_{\hat{L}_A}$. Therefore, $\{b\}$ is redundant.

In the case of condition P2, the problem becomes more complicated, and Equation 8 will only remain valid if we redefine $F^{\mathcal{O}_{\hat{L}}}(A,b)$ as $+\infty$ when $\exists \bar{\pi}_b$ such that $\bar{\pi}_b$ is rooted in $\mathcal{S}_b$ and $\mathcal{S}_o = A \setminus \{b\}$. In other words, $+\infty$ must be considered if there is a complete optimum path from the background seeds that reaches b, when we disregard b as internal seed. That is, if such path exists then, accordingly to P2, we have that b will be assigned to the background, so it can not be redundant in relation to A.

Now that we have a better characterization of equivalence and redundancy, we may discuss how to get a set

of seeds with minimum cardinality. Let $\hat{L}^* = (\mathcal{I}, L^*)$ be the given segmentation, that we intend to resume. We may initially take all object pixels as internal seeds (i.e., $\mathcal{S}_o = \{t \in \mathcal{I} \mid L^*(t) = 1\}$) and all background pixels as external seeds (i.e., $\mathcal{S}_b = \{t \in \mathcal{I} \mid L^*(t) = 0\}$). At a first moment, we may then reduce this number of seeds by selecting just one seed per core. Hence, we end up with a smaller set $\bar{\mathcal{S}}_o$ containing only non-equivalent seeds. After that, we may then identify a seed that is redundant in relation to all others and remove it. We can repeat this process until there are no more redundant seeds. At this point, we have a minimal subset of $\bar{\mathcal{S}}_o$ with respect to the segmentation $\hat{L}^*$. But does it have minimum cardinality? In order to answer this question we must prove that the order of removal of redundant elements does not affect the final results. That is, if $\bar{\mathcal{S}}_o = \{a\} \cup B \cup \{c\}$, under the hypotheses

(H1) $\{a\} \propto B \cup \{c\}$,
(H2) $\{c\} \propto B \cup \{a\}$,
(H3) there are no equivalent nodes in $\bar{\mathcal{S}}_o$,

we must prove that $\{a\} \propto B$ and $\{c\} \propto B$.

Proof: From Eq. 8 and hypotheses H1,H3, we have that

$$\begin{aligned} E_{\{a\}} &= F^{\mathcal{O}_{\hat{L}}}(B \cup \{c\}, a) \\ &= \min\left\{F^{\mathcal{O}_{\hat{L}}}(B, a), F^{\mathcal{O}_{\hat{L}}}(\{c\}, a)\right\}. \end{aligned} \quad (9)$$

From Eq. 8 and hypotheses H2 and H3, we have that

$$\begin{aligned} E_{\{c\}} &= F^{\mathcal{O}_{\hat{L}}}(B \cup \{a\}, c) \\ &= \min\left\{F^{\mathcal{O}_{\hat{L}}}(B, c), F^{\mathcal{O}_{\hat{L}}}(\{a\}, c)\right\}. \end{aligned} \quad (10)$$

From Equation 9 we may conclude that

$$E_{\{a\}} = \begin{cases} F^{\mathcal{O}_{\hat{L}}}(\{c\}, a) & \text{if } F^{\mathcal{O}_{\hat{L}}}(B, a) > E_{\{a\}} \\ F^{\mathcal{O}_{\hat{L}}}(B, a) & otherwise \end{cases} \quad (11)$$

From Equation 10 we may conclude that

$$E_{\{c\}} = \begin{cases} F^{\mathcal{O}_{\hat{L}}}(\{a\}, c) & \text{if } F^{\mathcal{O}_{\hat{L}}}(B, c) > E_{\{c\}} \\ F^{\mathcal{O}_{\hat{L}}}(B, c) & otherwise \end{cases} \quad (12)$$

The only valid combination between Equations 11 and 12 is $E_{\{a\}} = F^{\mathcal{O}_{\hat{L}}}(B, a)$ and $E_{\{c\}} = F^{\mathcal{O}_{\hat{L}}}(B, c)$ which implies that $\{a\} \propto B$ and $\{c\} \propto B$ as we wanted to prove. The other combinations lead to contradictions like $a \equiv c$, or violations of transitivity. ■

Now that we have a general procedure to find a set of seeds with minimum cardinality, we may proceed with a discussion about implementation issues.

A first point concerns the problem of how to evaluate Equations 6 and 8 efficiently. The energies $E_{\{s\}}$ of all internal seeds $s \in \mathcal{O}_{\hat{L}}$ are easy to compute. They are the values $V(s)$ of the path-value map V computed by Algorithm 1 with $f_{\max}$, but using only the external seeds during its execution. In order to compute the *pass-value* $f(a, b)$ (Eq. 5) we may exploit its relation with a *minimum-spanning tree* (MST) [6]. A minimum-spanning tree — is a tree whose sum of arc weights is minimum. In a MST, if we take any node $a \in \mathcal{I}$, there is a single path connecting a to any other node b and this path always has $f(a, b)$ as its maximum arc weight. Therefore, a MST encodes all possible pass-values $f(a, b)$ [10].

In order to compute a MST, we may use Algorithm 1 with the non-smooth path-value function f_{mst} (Eq. 13). Despite this function not being smooth (Definition 2), Algorithm 1 still returns a spanning tree P. Although this tree is rooted (in an arbitrary starting point) and P is a directed graph, the arc orientations have no meaning. The minimum-spanning tree is P without arc orientations. That is, Algorithm 1 with f_{mst} becomes Prim's algorithm that computes the MST [6].

$$\begin{aligned} f_{mst}(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \text{ is the starting point} \\ +\infty & otherwise \end{cases} \\ f_{mst}(\pi_s \cdot \langle s, t \rangle) &= w(s, t) \end{aligned} \quad (13)$$

Similarly, we can calculate the pass-value $f^{\mathcal{O}_{\hat{L}}}(a, b)$ constrained in $\mathcal{O}_{\hat{L}}$ by analyzing the paths of a MST restricted to the object. That is, by computing a MST over the subgraph induced by $\mathcal{O}_{\hat{L}}$. Of course, if the object presents several disconnected parts, then a MST for each connected component will be needed. In this case each component can be analyzed separately from the others, because the seeds between different components are necessarily non-redundant in relation to each other.

To compute the cores, we may start by verifying Eq. 6 which is simpler. Note that, thanks to the transitive property, in order to check whether an element belongs to a core or not, we just have to compare it with a single representative element of the core. So it is not necessary to test all possible combinations between elements. Hence, this could be implemented in an *union-find* manner.

But in order to find the core by Eq. 6 of a given representative seed s, a better implementation would be simply to make a breadth-first search from s in the MST topology, and select all elements that are reached only through arcs with weights strictly less than $E_{\{s\}}$.

During this process, we can create a graph of representative seeds with arcs weighted by the pass-value between them. A MST over this very reduced graph, can then be computed in order to encode the pass-value between different cores. This greatly simplifies the evaluation of Eq. 8. Thus, we may then finish the process by the successive removal of redundant representative seeds.

Other possible implementation would be simply to first consider, the efficient algorithm proposed in [7], which will return a small number of seeds. We may then reduce it even more by following the proposed extension to P1 and P2 with the advantage of having to evaluate Eq. 8 only over a reduced set of points.

V. How to resume a previous IFT segmentation

The next subsections treat, in order of severity, the most common causes of problems that occur when we decide to continue a previous segmentation session using the IFT.

A. Dealing only with ambiguities

This is the simplest case to treat. The previous segmentation to resume was obtained by IFT-SC, for a given known graph and set seed. However, as shown in Section III-B, the segmentation may not be unique due to the possible presence of tie-zone pixels. Indeed, when popular tie-breaking policies such as LIFO or FIFO are adopted, the segmentation results may vary depending on the order that the data is processed.

Fortunately this case is very easy to solve. Let's denote by $\hat{L}^* = (\mathcal{I}, L^*)$ the previous segmentation (label map) that we intend to continue. Only a small modification is needed in order to obtain an optimum-path forest that is always consistent with the label map $\hat{L}^*$. By initially setting $L(t) = L^*(t)$ for all $t \in \mathcal{I}$, and adding the test $L^*(s) = L^*(t)$ in Line 8, Algorithm 1 with $f_{\max}$ shall do it.

B. The image graph has changed

This case usually happens when the image is altered by filtering (e.g., Gaussian blur, radiometric transformation) after the last interactive session that we intend to resume. Since the arc weights of the graph are computed based on image attributes, they will also be affected, although the seeds are preserved.

In this scenario, it is very unlikely that an execution from the given seeds will be able to generate the exactly same result. Fortunately, a minimum number of additional seeds can be found by using the procedure described in Section IV. The only difference is that, while choosing the representative seeds inside the cores, we must give preference to the seeds marked by the user. By doing this, we avoid new seeds that are equivalent to the user input. After removing all redundant seeds, the only remaining new seeds will be non-redundant seeds that are truly required.

C. The image graph and seeds have changed

This case is similar to the previous one, but the locations of the seeds may also have been affected. This usually happens when the image is altered by some spatial transformation (e.g., interpolation, registration).

In the new space, the location of the seeds may suffer from discretization problems. Moreover, by changing the spatial resolution, the interpolated version of the mask may lose thin regions. Hence, extra care must be taken in order to avoid object's seeds badly positioned over the background and vice versa. So the only difference in relation to the previous case is that, before starting, we must make sure to eliminate any inconsistent seeds.

VI. Conclusion

In this work, a comprehensive recovery and resume capability was developed comprising different scenarios under the framework of the *image foresting transform* (IFT) [4]. Theoretical advances in relation to the state of the art were presented with an extension to different tie-breaking policies (P1 and P2). This extension can also be elegantly combined with the previous work [7]. As future work, we intend to analyze practical aspects that were not covered here.

Acknowledgment

The authors thank FAPESP (2009/16428-4 and 07/52015-0), and CNPq (481556/2009-5, 201732/2007-6 and 302617/2007-8) for the financial support.

References

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

[2] X. Bai and G. Sapiro, "Distance cut: Interactive segmentation and matting of images and videos," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, San Antonio, Texas, 2007, pp. II – 249–II – 252.

[3] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

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

[5] P. Miranda and A. Falcão, "Links between image segmentation based on optimum-path forest and minimum cut in graph," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 128–142, 2009, doi:10.1007/s10851-009-0159-9.

[6] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. MIT, 1990.

[7] R. Audigier and R. Lotufo, "Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches," in *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*. Belo Horizonte, MG: IEEE CPS, Oct 2007, pp. 61–68.

[8] P. Miranda, A. Falcão, and J. Udupa, "Synergistic arc-weight estimation for interactive image segmentation using graphs," *Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 85–99, Jan 2010, doi: 10.1016/j.cviu.2009.08.001.

[9] J. Udupa, P. Saha, and R. Lotufo, "Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1485–1500, 2002.

[10] C. Allène, J. Audibert, M. Couprie, J. Cousty, and R. Keriven, "Some links between min-cuts, optimal spanning forests and watersheds," in *Proceedings of the 8th Int. Symposium on Mathematical Morphology (ISMM)*, 2007, pp. 253–264.