Differential and Relaxed Image Foresting Transform for Graph-Cut Segmentation of Multiple 3D Objects

Nikolas Moya[1], Alexandre X. Falcão[1],
Krzysztof C. Ciesielski[2,3], and Jayaram K. Udupa[3]

[1] Institute of Computing, University of Campinas, Campinas, SP, Brazil
[2] Department of Mathematics, West Virginia University, Morgantown, WV, USA
[3] Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA

Abstract. Graph-cut algorithms have been extensively investigated for interactive binary segmentation, when the simultaneous delineation of multiple objects can save considerable user's time. We present an algorithm (named DRIFT) for 3D multiple object segmentation based on seed voxels and Differential Image Foresting Transforms (DIFTs) with relaxation. DRIFT stands behind efficient implementations of some state-of-the-art methods. The user can add/remove markers (seed voxels) along a sequence of executions of the DRIFT algorithm to improve segmentation. Its first execution takes linear time with the image's size, while the subsequent executions for corrections take sublinear time in practice. At each execution, DRIFT first runs the DIFT algorithm, then it applies diffusion filtering to smooth boundaries between objects (and background) and, finally, it corrects possible objects' disconnection occurrences with respect to their seeds. We evaluate DRIFT in 3D CT-images of the thorax for segmenting the arterial system, esophagus, left pleural cavity, right pleural cavity, trachea and bronchi, and the venous system.

Keywords: Image segmentation, differential image foresting transform, boundary smoothing, graph-cut algorithms.

1 Introduction

This work studies the segmentation algorithm which, given an image (3D, medical) and $M \geq 1$ desired objects, returns a label map L from the image domain D into $\{0, \ldots, M\}$, where the label 0 designates the background and voxels in the ith object are assigned to label $1 \leq i \leq M$. Although some algorithms do not need more input (e.g., those based on the Mumford-Shah model [1]), they are neither efficient nor accurate. Therefore, we will focus on the algorithms that require the object location as input, in a format of the seed sets $\lambda^{-1}(0), \ldots, \lambda^{-1}(M)$ (see e.g. [2–5]), where λ is a function from $S \subset D$ into $i \in \{0, \ldots, M\}$.

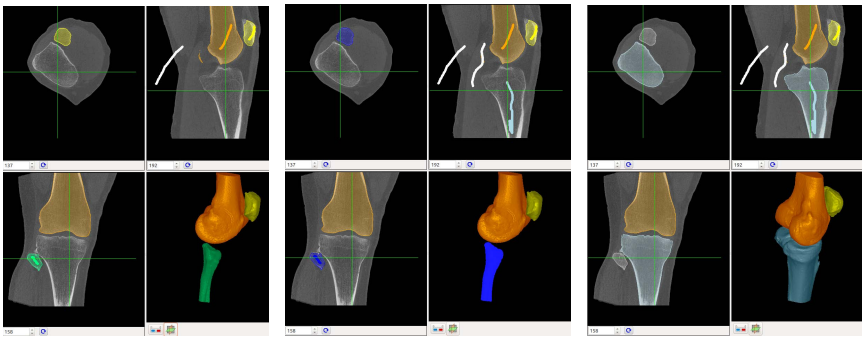
Even with seeds indicating object location, the state-of-the-art segmentation methods rarely provide satisfactory results in a single execution, asking for the user's assistance to add seeds and improve segmentation. These steps can be

P. Golland et al. (Eds.): MICCAI 2014, Part I, LNCS 8673, pp. 690–697, 2014.

repeated several times until user's satisfaction is reached. Most algorithms, however, are limited to binary segmentation, when the simultaneous segmentation of multiple objects can save the user's time considerably.

Figure 1 illustrates the interactive 3D segmentation of multiple objects using Differential Image Foresting Transforms (DIFTs) [6]. Markers (seed sets, represented by distinct colors) propagate their labels to the most strongly connected voxels in the image. The image is interpreted as a graph, with arcs given by an *adjacency relation* between voxels, and each voxel is conquered by a seed which offers an optimum path to it. The result is an *optimum-path forest* with labeled trees, where each object is the union of the trees painted with the same color (the label map). The forest is used to correct segmentation, since it connects markers to their influence zones in the image. This algorithm takes time proportional to the number of voxels in the first execution, but subsequent segmentation corrections (marker addition and/or removal) usually take time proportional to the size of the modified regions (i.e., sublinear time). The label map is also a graph cut that minimizes the maximum arc-weight along the cut, given seeds as constraints (i.e., the DIFT is a $GC_{\max}$ algorithm [7]). The DIFT is part of the Image Foresting Transform (IFT) methodology [8], which accepts as input either an image or an optimum-path forest resulting from a previous execution.

A crucial requirement in the above procedure is an interactive response time to the user's actions (i.e., a few seconds or, preferably, instantaneous response). The DIFT meets the speed requirement for large medical images and stands behind efficient implementations of $GC_{\max}$ methods, such as Iterative Relative Fuzzy Connectedness [7] and Watershed Transforms [6]. A negative aspect of



(a) (b) (c)

Fig. 1. (a) By examining orthogonal slices, the user draws green, yellow and orange markers (seed voxels) in three bones and white marker in the background. The bottom-right part displays the 3D rendition of the resulting label map. (b) The user can change mind, pick any voxel in the influence zone of the green marker to remove the bone (dark blue indicates marked for removal). At the same time, the user can insert new markers (cyan for a new bone and white for background) to adjust segmentation. (c) The algorithm removes the green bone and creates a new result with the other three selected bones.

such methods, however, is the lack of regularity of the segmented object boundaries (the "leaking problem"). Boundary smoothness can be enforced by post-processing. In [9], for instance, the IFT results were adjusted by the max-flow algorithm and in [10], a version of diffusion was used for boundary smoothing, resulting into the "relaxed IFT algorithm" (RIFT).

In this paper, we revisit the RIFT algorithm [10] to correct its inconsistency in segmentation and to propose its differential version with a considerably faster boundary smoothing process, which applies diffusion while the boundary between distinct labels is expanded inward the objects (and background) through a few iterations (e.g., 10). The initial boundary is also found during the DIFT computation, when wavefronts from distinct labels meet each other. Given that any post-processing for boundary smoothing might disconnect a part of an object from its most strongly connected seed, such inconsistency is corrected as follows: We find (in a fully automatic way) a modified set of seeds and use the DIFT algorithm with these seeds to create a new forest, which induces the segmentation closed to the one produced by the smoothing step, while removing any objects' disconnection occurrences. This second forest-induced segmentation is the one that the user examines and either accepts, or continues to improve.

One can use the boundary smoothing and connectivity correction (second DIFT module) steps after each application of the first DIFT module or only once, at the end of the process. In preliminary tests, we found the latter less accurate in most cases, since it does not give the chance to correct the process. Therefore, we evaluate the first solution, named *Differential and Relaxed IFT* (DRIFT), in comparison with the DIFT [6] and the Dynamic Graph-Cut (DGC) algorithms [11] for segmenting 3D CT-images of the thorax into: arterial system, esophagus, left pleural cavity, right pleural cavity, trachea and bronchi, and venous system. The segmentation of these objects is especially challenging due to their different shapes and lack of boundary information in several parts.

2 The Details of the Algorithm

The DRIFT Algorithm can be presented as follows.

Algorithm 1 – DRIFT Algorithm

Input:	The image $I\colon D \to \mathbb{R}^m$ and seeds' labeling $\lambda\colon S \to \{0,\ldots,M\}$.
Output:	The forest $\mathbb{F}$ in the image; labeling $L\colon D \to \{0,\ldots,M\}$ extending λ.
Auxiliary:	Termination variable *flag* initiated as *FALSE*, boundary set $B \subset D$, and dilated boundary set $B_d \subset D$.

1. *Run DIFT with I and λ to get the forest $\mathbb{F}$, label map L, and boundary set B.*
2. **While** *flag = FALSE* **do**
3. *Smooth L from B, returning new labeling L' and dilated set $B_d \supset B$.*
4. *Run DIFT from B_d to get new $\mathbb{F}$ and associated labeling L consistent with L'.*
5. **If** *User is satisfied with L* **then**
6. *Set flag $\leftarrow$ TRUE.*
7. **Else**

8. *Get correcting labeling λ' from the user.*
9. *Run DIFT for λ' to get new forest $\mathbb{F}$, labeling L, and set B.*

In the rest of these section we will explain in more details what stands behind the lines 1, 3, 4, 8, and 9.

Details on DIFT. We assume that $D \subset \mathbb{Z}^3$ and its voxels are the nodes of a graph G, whose arcs $\{s,t\} \subset \mathcal{A} \subset D \times D$ satisfy $0 < \|t-s\| \leq \sqrt{3}$ (adjacency relation). We also say that $\mathcal{A}(s)$ is the set $\{t \in D\colon 0 < \|t-s\| \leq \sqrt{3}\}$. The graph G is weighted on the arcs by $w(s,t) = K\omega(t)$, where $K > 0$ is an integer and ω is defined to assign lower values on object boundaries than elsewhere: $\omega(s) = 1 - \frac{1}{|\mathcal{A}(s)|}\sum_{t\in\mathcal{A}(s)} \frac{\|I(t)-I(s)\|}{I_{\max}}$, $I_{\max} = \max_{s\in D}\|I(s)\|$.

The $DIFT$ algorithm can be run for: any subgraph G' of G induced by the vertices $D' \subset D$; any set $S' \subset D'$ of seeds such that for each $t \in D'$ there is a path in G' from t to S'; any labeling $\lambda\colon S' \to \{0,\ldots,M\}$; and any initial seed-strength map $\mu\colon S' \to [0,K]$. Then, $DIFT$ returns a forest $\mathbb{F}$ in G' rooted at S', that is, a family of paths in G' such that: any initial segment of a $p \in \mathbb{F}$ is also in $\mathbb{F}$; every $p \in \mathbb{F}$ contains precisely one seed from S'; and for each $t \in D'$ there is a $p_t \in \mathbb{F}$ from $s_t \in S'$ to t. The paths in $\mathbb{F}$ are indicated by the predecessor map $P\colon D' \setminus S' \to D'$, where $P(t)$ is a predecessor of t in p_t. Since the map P uniquely determines $\mathbb{F}$, we will identify $\mathbb{F}$ with P, wherever convenient. The forest $\mathbb{F}$ returned by $DIFT$ is optimal in the following sense. For a path $\pi = \langle t_0,\ldots,t_n\rangle$ from S' we define its strength as $f(\pi) = \min_{i<n}\{\mu(t_0), w(t_i,t_{i+1})\}$. Then for any path $\pi = \langle t_0,\ldots,t_m,t\rangle$ in G' with $\langle t_0,\ldots,t_m\rangle \in \mathbb{F}$ and $t \notin S'$, either $f(p_t) > f(\pi)$ or both $f(p_t) = f(\pi)$ and $\lambda(s_t) \leq \lambda(t_0)$.

Line 1. We run $DIFT$ with $D' = D$, $S' = S$, λ provided by the user, and $\mu(s) = K$ for all $s \in S'$ and for $s \in D' \setminus S'$, $\mu(s) = -\infty$. The output labeling L is defined as $L(t) = \lambda(s_t)$ and the boundary set B is $\{s \in D\colon \exists t \in \mathcal{A}(s), L(t) \neq L(s)\}$.

Line 3. The smoothing is a diffusion filtering on the label map L, starting from B, that takes a fix number T of iterations (e.g., $T = 10$). At each iteration, B generates a dilated set $B_d = B \cup \{t \in D \setminus B, t \in \mathcal{A}(s), s \in B\}$, which turns to be the set B for the next iteration. The final set B_d is also returned and it contains the voxels that might have changed labels during diffusion.

For every $s \in D$, put $W(s) = \frac{1}{1+\alpha(1-\omega(s))}$, where $\alpha \in [0,1]$ is a fixed smoothing factor, and define a normalization factor as $N(s) = \sum_{t\in\mathcal{A}(s)} W(t)$. (For sake of efficiency, the maps W and N should be computed only once and used as input).

Put $L_0 = L$ and for every $t \in B_d$ and $l \in \{0,1,\ldots,M\}$ define $\mu_0(l,t) = 1$ if $l = L_0(t)$ and $\mu_0(l,t) = 0$, otherwise. Then, recursively, for $i \in \{1,\ldots,T\}$ and $s \in B$, we find $\mu_i(l,s) = \frac{1}{N(s)}\sum_{t\in\mathcal{A}(s),\, L_{i-1}(t)=l} \mu_{i-1}(l,t)W(t)$. Note that $\sum_{l=1}^{M}\mu_i(l,s) = 1$ for every $i \in \{0,\ldots,T\}$ and $s \in B$. Define the labeling L_i by putting $L_i(s) = l$ when for every $l' \in \{0,1,\ldots,M\}$ either $\mu_i(l,s) > \mu_i(l',s)$ or both $\mu_i(l,s) = \mu_i(l',s)$ and $l \leq l'$. At the end, $L' = L_T$.

Line 4. Let $U_0 = \{t \in B_d \colon L'(t) \neq L(t)\}$ be the set of voxels that changed labeling and U be the subtrees of the voxels in U_0. That is, U is the set of all $t \in D$ for which the path $p_t \in \mathbb{F}$ contains a voxel from U_0. The voxels in U were influenced (w.r.t. $\mathbb{F}$) by the change of labeling and they need to be "reconquered." Define the frontier set S' as $s \in D \setminus U$ for which there exists a $t \in \mathcal{A}(s) \cap U$ with $L(s) = L'(t)$. This is the new set of seeds, that will compete for the voxels in U.

For this we put $D' = U \cup S'$ and define, for every $s \in S'$, $\lambda(s) = L(s)$ and $\mu(s) = f(p_s)$, where $p_s \in \mathbb{F}$, and for every $s \in U$, $\mu(s) = -\infty$. Run DIFT with this setup.

Of course, the forest $\mathbb{F}'$ that is returned is defined only on D'. However, if $P' \colon U \to D'$ is a predecessor map for $\mathbb{F}'$, then the new predecessor map defined as P' on U, as an old predecessor map on $D \setminus U$, determines full predecessor map on D and the new forest $\mathbb{F}$ (for, possibly decreased, set $S \setminus U$ of seeds). It is worth to notice that, this new forest need not to be optimal in the sense discussed above.

Line 8. The correcting labeling λ' is a map from $W \subset D$ into $\{-1, 0, \ldots, M\}$. The meaning of a label $\lambda'(s) \in \{0, \ldots, M\}$ is straightforward: it means that the user assigns s as a seed for the ith object (background). (Note that assignment makes sometimes sense even when we previously had $\lambda(s) = i$, as this changes the connectivity strength between s and the seed set.) The assignment $\lambda(s) = -1$, on the other hand, is for removing the marker, whose influence zone (its forest) contains s. Given that the user can make mistakes, marker deletion is a desirable feature.

Line 9. Let $W_0 = \{s \in W \colon \lambda'(s) = -1\}$ and $W_1 = W \setminus W_0$. From W_0, the predecessor map identifies a set U_0 with the roots of all trees selected for removal and, as for line 4, let U be the subtrees of the voxels in U_0.

Define the frontier set F as those $s \in D \setminus U$ for which there exists a $t \in \mathcal{A}(s) \cap U$. Put $S' = W_1 \cup F$ and $D' = D$. Define λ and μ on S' as follows. For $s \in W_1$ we put $\lambda(s) = \lambda'(s)$ and $\mu(s) = K$. For $s \in F$ we put $\lambda(s) = L(s)$ and $\mu(s) = f(p(s))$. For $s \in U \setminus W_1$ we put $\mu(s) = -\infty$. Then, we run $DIFT$ with this setup to get a forest $\mathbb{F}'$ on D'. We recreate from it the new forest $\mathbb{F}$ and label map L, as in Line 4, and the boundary set B, as in Line 1.

3 Experiments

The experiments evaluate the accuracy and efficiency of the DRIFT (as an IRFC implementation [7]), DIFT [6], and DGC [11] algorithms running on the same input graph (nodes are voxels) from 40 3D CT-images of the thorax to segment: the arterial system (AS), esophagus (E), left pleural cavity (LPS), right pleural cavity (RPS), trachea and bronchi (TB), and venous system (VS). Since DGC is constrained to segment object by object, we present the total mean number of executions and the sum of the mean times spent per execution on each object. The ground-truth (GT) images were created by several experts, as described

in [12]. The images were acquired with voxel size $0.5 \times 0.5 \times 5mm^3$, so they were interpolated to $158 \times 158 \times 116$ voxels with $2.5 \times 2.5 \times 2.5mm^3$ each.

Seeking to reduce costs and biases associated with evaluation by real users, we used geodesic robot users to generate the seed sets [13]. Given the GT image, the geodesic robot adds a given number of seeds along the executions in the error components, until they are too small for new marker selection. The markers were spheres whose radius varied proportionally with the size of the error components within $[1, 10]$ voxels. A maximum of 56 makers per execution was used (8 markers per object plus 8 for the background), the number of smoothing iterations $T = 10$, and the smoothing factor $\alpha = 0.5$.

The error values in Table 1 are in mm and they correspond to the mean error over the 40 images according to the average symmetric absolute surface distance between GT and segmentation[1].

Table 1. Mean error and standard deviation in mm per object after convergence, using 8 seeds per object (background) and 10 smoothing iterations

Algorithm	AS	E	LPS	RPS	TB	VS	Mean
DRIFT	2.6±2.19	1.3±0.58	1.0±1.91	0.8±1.36	0.8±0.33	1.6±0.86	1.4±0.95
DIFT	3.4±1.43	2.2±2.14	0.7±0.24	0.7±0.24	0.7±0.24	2.1±1.25	1.6±0.52
DGC	4.8±0.95	3.4±0.63	1.2±0.13	1.1±0.15	1.3±0.74	4.2±0.72	2.0±0.57

For a statistical significance level of 95%, DRIFT is more accurate than DGC in all cases and than DIFT for AS, E, and VS (sparse objects). On average, it required less executions for convergence (i.e., it should not affect the user's control over segmentation) and its response time was about 4.6s per execution (Table 2).

Table 2. Mean number of executions and mean time per execution (s) and their standard deviations

Algorithm	# of executions			Time per execution (s)		
	DRIFT	DIFT	DGC	DRIFT	DIFT	DGC
Thorax	13.8±4.95	15.5±5.0	48.1 ± 13.16	4.6 ±0.96	2.5±0.61	8.5 ± 2.4

Figure 2 also illustrates for the most accurate methods, DIFT and DRIFT, that irrespective to their accuracy differences, the results with boundary smoothing seem to match better with the users' expectations. All these aspects involving real users need to be investigated.

[1] `http://mbi.dkfz-heidelberg.de/grand-challenge2007/sites/eval.htm`

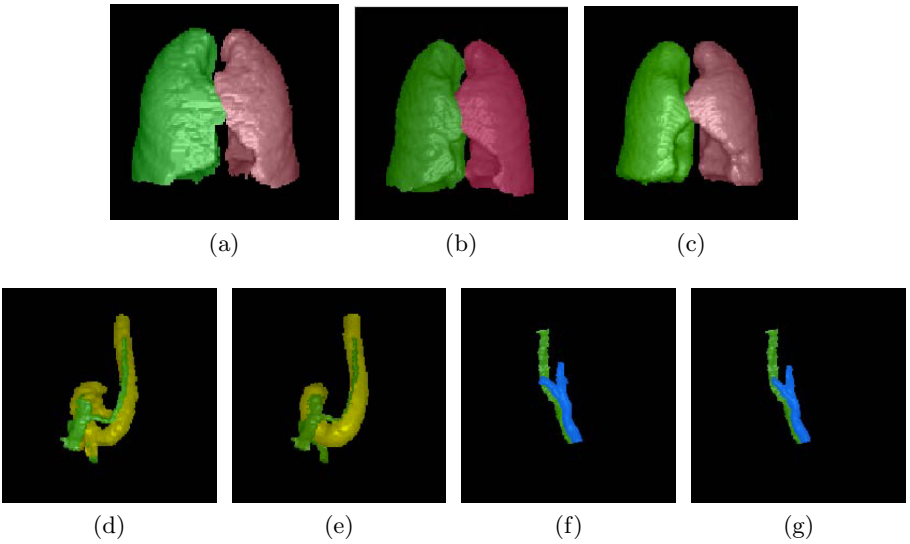


Fig. 2. RPS and LPS: (a) GT, (b) DIFT, and (c) DRIFT. AS and VS: (d) DIFT and (e) DRIFT. E and TB: (f) DIFT and (g) DRIFT.

4 Conclusion

We presented the DRIFT algorithm for interactive segmentation of multiple 3D objects using optimum-path forests and boundary smoothing. DRIFT can provide better accuracies than the DIFT and DGC algorithms, with interactive response time to the user's actions, being much faster and correct with respect to the RIFT algorithm. DRIFT can be used to devise new segmentation methods, by choice of other connectivity functions [8], as well as to implement state-of-the-art methods [7] more efficiently. The algorithm was tested on 40 3D CT-images of the thorax, presenting accuracy gains and visual results that seem to better match with the users' expectations. From Table 2, DRIFT required only about 63s of computational time, on average, to complete segmentation of 6 objects, considerably outperforming the DGC algorithm in accuracy and speed.

Future research is required to extend DRIFT to supervoxel graphs for further speed-up gains and assess it on other datasets, possibly involving distinct imaging modalities, with different arc-weight functions, and in practical situations involving multiple experts.

Acknowledgments. The authors thank FAPESP (2013/17991-0), CAPES, CNPq (303673/2010-9,479070/2013-0).

References

[1] Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 42(5), 577–685 (1989)

[2] Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: International Conference on Computer Vision, vol. 1(1), pp. 105–112 (2001)

[3] Sinop, A., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: Proceedings of ICCV (2007)

[4] Ciesielski, K., Udupa, J., Saha, P., Zhuge, Y.: Iterative relative fuzzy connectedness for multiple objects, allowing multiple seeds. Computer Vision and Image Understanding 107(3), 160–182 (2007)

[5] Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: A unifying graph-based optimization framework. IEEE Transactions Pattern Analysis and Machine Inteligence 33(7), 1384–1399 (2011)

[6] Falcão, A., Bergo, F.: Interactive volume segmentation with differential image foresting transforms. IEEE Transactions on Medical Imaging 23(9), 1100–1108 (2004)

[7] Ciesielski, K., Udupa, J., Falcão, A., Miranda, P.: Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis. Journal of Mathematical Imaging and Vision 44(3), 375–398 (2012)

[8] Falcão, A., Stolfi, J., Lotufo, R.A.: The image foresting transform: theory, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 19–29 (2004)

[9] Ciesielski, K., Udupa, J., Miranda, P., Falcão, A.: Joint graph cut and relative fuzzy connectedness image segmentation algorithm. Medical Image Analysis 17(8), 1046–1057 (2013)

[10] Malmberg, F., Nyström, I., Mehnert, A., Engstrom, C., Bengtsson, E.: Relaxed image foresting transforms for interactive volume image segmentation. In: Proceedings of SPIE-The International Society for Optical Engineering, vol. 23(1), pp. 40–51 (2010)

[11] Kohli, P., Torr, P.H.S.: Dynamic graph cuts for efficient inference in markov random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(12), 2079–2088 (2007)

[12] Udupa, J., Odhner, D., Tong, Y., Matsumoto, M.M.S., Ciesielski, K.C., Vaideeswaran, P., Ciesielski, V., Saboury, B., Zhao, L., Mohammadianrasanani, S., Torigian, D.: Fuzzy model-based body-wide anatomy recognition in medical images. In: Proceedings SPIE on Medical Imaging, vol. 8671, pp. 86712B–86712B–7 (2013)

[13] Kohli, P., Nickisch, H., Rother, C., Rhemann, C.: User-centric learning and evaluation of interactive segmentation systems. Int. J. Computer Vision 100(3), 261–274 (2012)