

Leissi M.C. Leon, Krzysztof C. Ciesielski, and Paulo A.V. Miranda*

Efficient Hierarchical Multi-Object Segmentation in Layered Graphs

<https://doi.org/10.1515/mathm-2020-0108>

Received August 11, 2020; accepted June 1, 2021

Abstract: We propose a novel efficient seed-based method for the multi-object segmentation of images based on graphs, named Hierarchical Layered Oriented Image Foresting Transform (HLOIFT). It uses a tree of the relations between the image objects, with each node in the tree representing an object. Each tree node may contain different individual high-level priors of its corresponding object and defines a weighted digraph, named as layer. The layer graphs are then integrated into a hierarchical graph, considering the hierarchical relations of inclusion and exclusion. A single energy optimization is performed in the hierarchical layered weighted digraph leading to globally optimal results satisfying all the high-level priors. The experimental evaluations of HLOIFT, on medical, natural, and synthetic images, indicate promising results comparable to the related baseline methods that include structural information, but with lower computational complexity. Compared to the hierarchical segmentation by the min-cut/max-flow algorithm, our approach is less restrictive, leading to globally optimal results in more general scenarios, and has a better running time.

Keywords: Multi-object segmentation, hierarchical image segmentation on graphs, medical image segmentation

AMS: 68U10

1 Introduction

Image segmentation is one of the most fundamental and challenging problems in image processing and computer vision. It is the task of partitioning an input image into objects of interest by assigning distinct labels to their composing pixels. Hence, image segmentation can be interpreted as a classification problem at the pixel level. Not surprisingly, machine learning-based methods are among the most prominent solutions to the segmentation problem, especially after the advent of deep learning techniques [21]. In this context, a state-of-the-art method in grabcut-style [41] for interactive segmentation with minimal user involvement is Deep Extreme Cut [28]. It considers extreme points (left-most, right-most, top and bottom pixels at the object boundary) as 4 clicks for each object, and the CNN produces the segmented masks. This type of user input is usually a good choice for single-shot algorithms, since from the 4 indicated points we can infer object's bounding box, and consequently, all points outside the box belong to its background. At first glance, this seems to be better than other types of user inputs, since arbitrary points do not allow us to draw any further conclusions. However, it is not possible in general to get a segmentation result with ground truth quality using only a 4 points selection, since considerable residual errors all around the boundary are quite common (Figures 1a and c). Moreover, user input in grabcut-style is usually not appropriate to perform further cor-

Leissi M.C. Leon: University of São Paulo, Institute of Mathematics and Statistics, CEP 05508-090, São Paulo, SP, Brazil, E-mail: leissicl@ime.usp.br

Krzysztof C. Ciesielski: Department of Mathematics, West Virginia University, Morgantown, USA, E-mail: Krzysztof.Ciesielski@mail.wvu.edu

***Corresponding Author: Paulo A.V. Miranda:** University of São Paulo, Institute of Mathematics and Statistics, CEP 05508-090, São Paulo, SP, Brazil, E-mail: pmiranda@ime.usp.br

rections, such that a different type of corrective action is usually employed in approaches like GrabCut [41], resulting in a less user-friendly interface due to the multiple types of input. On the other hand, deep learning techniques require a lot of annotated data for training the network and for many applications the availability of gold standard is quite limited.

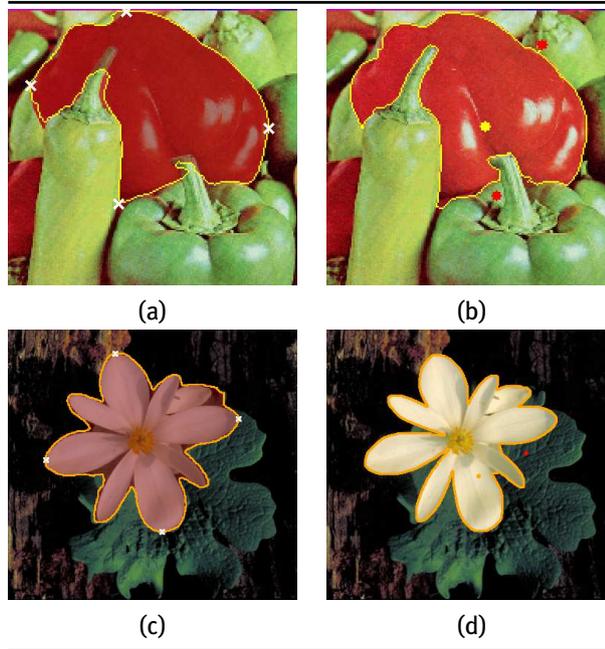


Figure 1: The results of segmentations on two natural images by Deep Extreme Cut and by a particular case of HLOIFT, in single object setting and with no high-level priors. (a,c) The results by Deep Extreme Cut with 4 clicks. (b,d) The results by HLOIFT with arc-weight estimation from [13] using only 3 and 2 clicks, respectively. Notice, that Deep Extreme Cut has poor boundary adherence in the pepper segmentation while in the flower segmentation it invades the surrounding green area of the leaves.

In this work, we focus on the interpretation of the image segmentation as a graph partition problem subject to hard constraints, given by seed pixels selected in the image domain, such that a refined segmentation can be obtained with an arbitrary level of precision, leading to a high-quality ground-truth data. The intended methods in graphs are very flexible. Besides being easily extensible for multidimensional images, they allow several customizations, such as the usage of different image elements as its nodes (e.g., superpixels [24]), the learning of graph weights by means of machine learning techniques [13] (see Figures 1b and d), or even the incorporation of learning by deep learning techniques [55, 56]. For the sake of simplicity and space constraints, here we focus only on the base form of the proposed method at the pixel level, without the use of training images, and discuss the use of high-level priors that can be easily set based on the application knowledge.

Examples of seed-based methods are watershed [12], random walks [19], fuzzy connectedness [6], graph cuts (GC) [2], grow cut [25], minimum barrier distance [9], and image foresting transform (IFT) [7, 18]. Some methods, including the min-cut/max-flow algorithm, can provide global optimal solutions according to a graph-cut measure in graphs and can be described in a unified manner according to a common framework, to which we refer as Generalized GC (GGC) [5]. *Oriented Image Foresting Transform* (OIFT) [35] and *Oriented Relative Fuzzy Connectedness* (ORFC) [1] are the GGC methods designed for directed weighted graphs, which have low computational complexity as compared to the min-cut/max-flow algorithm [2].

Global object properties, such as connectedness [29, 54], shape constraints [20, 22, 32, 53], and boundary polarity [35, 42], are potentially useful high-level priors for object segmentation, allowing the customization of the segmentation to a given target object.

In the context of multiple object segmentation, each object may present its own distinctive features, requiring different high-level priors. In order to obtain a good segmentation result, the segmentation method must attend all individual object priors and capture the contextual or structural relations between them. However, many existing methods either do not include any form of structural information or include only the high-level priors for single object segmentation [20, 29, 35, 54]. Consequently, they may be inappropriate in the context of multiple objects.

Most of the methods for multi-object segmentation that include structural information are based on graph-cut optimization and are performed by a min-cut/max-flow algorithm [15, 16, 52]. These methods usually use priors, based on inclusion or exclusion interactions between objects. However, their globally optimal results are restricted only to some particular cases and they usually have a high computational cost. The methods based in LOGISMOS (*Layered Optimal Graph Image Segmentation for Multiple Objects and Surfaces*) [38, 57] require an approximated pre-segmentation whenever the objects present complex shapes, in order to get information about the topological structures of the target objects. Note that the fast segmentation obtained by our proposed method could also be used as a starting point for LOGISMOS. In the context of segmentation by the *Image Foresting Transform* (IFT) framework, in order to incorporate structural information among objects, Fuzzy Object Models (FOMs) [33, 36, 37, 40, 44, 47–51] are usually employed. However, these approaches are based on separate IFT executions per object, that do not incorporate contextual information and the high-level priors of all objects into a single energy optimization, limiting their potential.

In this work, we circumvent the aforementioned problems, by proposing a hierarchical layered graph-based approach for multiple object segmentation, named as *Hierarchical Layered OIFT* (HLOIFT), which performs a single energy optimization subject to individual and structural high-level priors. A graph used in HLOIFT for a segmentation of m objects is formed with m layers, copies of image scene graph, each associated with an object. The hierarchy considers inclusion and exclusion constraints between objects as in [15], but each object in the image can also have its own set of high-level priors. Further, we formulate the integration of individual object constraints and structural priors from layers, within a single energy optimization, overcoming the mentioned limitations from previous works and conserving the low computational cost of OIFT.

Specifically, our main contributions are as follows. **Theoretical:** We propose a new seed-based method for multi-object segmentation allowing high-level priors for image objects and the hierarchical constraints between them. **Generality:** Our approach is less restrictive than most methods in use and leads to the globally optimal results in more general scenarios. **Complexity:** Our method has low computational complexity as compared to methods based on min-cut/max-flow.

HLOIFT was first presented in a conference paper [23] and later demonstrated for use in graphs of superpixels [24]. In this extended work, we provide more details in the presentation of the method, including a formal definition of the obtained objects, the complete algorithm, a formal mathematical proof of the correctness of the algorithm, and more experiments, quantitatively comparing with the hierarchical graph cut algorithm [15, 16], one of the few existing methods that can also use a full hierarchy prior as an input.

Our work is organized as follows: In Section 2, we introduce some required definitions and review the OIFT method. Our new algorithm HLOIFT is described in Section 3 and the proof of its correctness is given in Section 4. The experimental evaluation of HLOIFT is presented in Section 5, while our conclusions are stated in Section 6.

2 Background

An (n -dimensional) image is a pair (\mathcal{J}, I) , where \mathcal{J} is a finite set of *pixels* identified with the n -tuples of integers (i.e., $\mathcal{J} \subset \mathbb{Z}^n$) and I is a mapping that assigns to each pixel $t \in \mathcal{J}$ its intensity $I(t)$, that is, either a real number (in case of monochromatic image) or an appropriate vector in \mathbb{R}^k . An image can be interpreted as a weighted digraph (i.e., directed graph) $G = (\mathcal{N}, \mathcal{A}, \omega)$ whose nodes (vertices) are the pixels in the image domain $\mathcal{N} = \mathcal{J}$, arcs/edges listed in \mathcal{A} are the pairs $(s, t) \in \mathcal{J}^2$ of pixels (usually, in 2D images, identified with either 4- or

8-neighborhoods), and the weight map ω associates to each arc $(s, t) \in \mathcal{A}$ a value $\omega(s, t) \in [-\infty, \infty]$ (usually defined as $\omega(s, t) = \|I(t) - I(s)\|$).¹ We use the notations $t \in \mathcal{A}(s)$ and $(s, t) \in \mathcal{A}$ to indicate that t is adjacent to s . A digraph G is *symmetric* if for all $(s, t) \in \mathcal{A}$, the pair (t, s) is also an arc of G , that is, $(t, s) \in \mathcal{A}$. All digraphs we consider below are symmetric. Notice, however, that in symmetric digraphs we may still have $\omega(s, t) \neq \omega(t, s)$.

2.1 Oriented Image Foresting Transform (OIFT)

Image segmentation can be formulated as a graph partition problem subject to hard constraints. In the case of binary segmentation whose goal is to separate an object O_1 from its background $O_0 = \mathcal{J} \setminus O_1$, we consider two non-empty disjoint seed sets $\mathcal{S}_0, \mathcal{S}_1 \subset \mathcal{J}$ indicating, respectively, O_0 and O_1 (i.e., aiming for $O_1 \supset \mathcal{S}_1$ and $O_0 \supset \mathcal{S}_0$). So, we assume that $\mathcal{S}_1 \cap \mathcal{S}_0 = \emptyset$. Actually, the object O_1 is identified with its *labeling* $L: \mathcal{J} \rightarrow \{0, 1\}$, which is the indicator (i.e., characteristic) function χ_{O_1} of O_1 . (That is, $O_1 = \{v \in \mathcal{J}: L(v) = 1\}$.) In particular, the seeds give us a partial labeling, $L(t) = 1$ for all $t \in \mathcal{S}_1$ and $L(t) = 0$ for all $t \in \mathcal{S}_0$, which is propagated to all unlabeled pixels during the OIFT algorithm [35].

In the case of directed weighted graphs, there are two important classes of energy formulations within the Generalized GC framework, the Max-Min² and Min-Sum optimizers [5]. The OIFT and ORFC algorithms are Max-Min optimizers while the min-cut/max-flow algorithm is a Min-Sum optimizer. The resulting segmentation by OIFT gives, subject to the seed constraints [35], a global optimum solution by maximizing the graph-cut measure ε_{min} defined as

$$\varepsilon_{min}(L) = \min\{\omega(s, t): (s, t) \in \mathcal{A} \ \& \ L(s) > L(t) = 0\}. \quad (1)$$

For the sake of simplicity, the OIFT algorithm is being presented here without the explicit forest calculation, from which its name comes from. We can do this, since the label map L without the forest is sufficient for the segmentation task. The OIFT segmentation, indicated by L , can be computed by Algorithm 1, in a connected and symmetric digraph G , as described in [31].

Algorithm 1. – OIFT ALGORITHM

INPUT: Image graph $(\mathcal{N}, \mathcal{A}, \omega)$, non-empty disjoint seed sets \mathcal{S}_0 and \mathcal{S}_1 .
OUTPUT: The label map $L: \mathcal{N} \rightarrow \{0, 1\}$.
AUXILIARY: Priority queue Q , variable tmp , the cost function $V: \mathcal{N} \rightarrow [-\infty, \infty]$, and a status function $S: \mathcal{N} \rightarrow \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.

```

1. For each  $t \in \mathcal{N}$ , do
2.   Set  $S(t) \leftarrow 0$  and  $V(t) \leftarrow \infty$ ;
3.   If  $t \in \mathcal{S}_0$ , then
4.      $V(t) \leftarrow -\infty$ ,  $L(t) \leftarrow 0$ , and insert  $t$  in  $Q$ ;
5.   If  $t \in \mathcal{S}_1$  then
6.      $V(t) \leftarrow -\infty$ ,  $L(t) \leftarrow 1$ , and insert  $t$  in  $Q$ .
7. While  $Q \neq \emptyset$  do
8.   Remove  $s$  from  $Q$  such that  $V(s)$  is minimum;
9.   Set  $S(s) \leftarrow 1$ ;
10.  For each  $(s, t) \in \mathcal{A}$  such that  $S(t) = 0$  do
11.    If  $L(s) = 1$  then  $tmp \leftarrow \omega(s, t)$ 
12.    Else  $tmp \leftarrow \omega(t, s)$ ;
13.    If  $tmp < V(t)$  then
14.      Set  $V(t) \leftarrow tmp$  and  $L(t) \leftarrow L(s)$ 
15.      If  $t \notin Q$  then insert  $t$  in  $Q$ .

```

¹ The symbol $\|\cdot\|$ denotes the standard Euclidean norm in \mathbb{R}^k .

² Min-Max optimizer is a dual equivalent problem.

Note that in line 12 of Algorithm 1, the arc weight $\omega(t, s)$ of the reversed arc (t, s) is used (rather than that of chosen $(s, t) \in \mathcal{A}$). That is why a symmetric digraph is required. The value $V(t)$ assigned in line 14 constitutes a contribution, to the energy $\varepsilon_{\min}(L)$ given by (1), that a consecutive change of labeling $L(t)$ from $L(s)$, already fixed, to $1 - L(s)$ would represent. For every node t considered in lines 10-15 the value $V(t)$ represents the worst (minimum) weight (w_0 or w_1 , as defined in Remark 1) of the arcs $(v, t) \in \mathcal{A}$ from already processed object nodes v (i.e., with $S(v) = 1$).

Remark 1. Notice, that if we define the weight functions w_0 and w_1 on \mathcal{A} as $w_1(s, t) = \omega(s, t)$ and $w_0(s, t) = \omega(t, s)$, then the execution of lines 11 and 12 in Algorithm 1 is equivalent to the execution of a single line: “ $\text{tmp} \leftarrow w_{L(s)}(s, t)$.” Also, if consecutive line 14 is executed, then $V(t)$ becomes $w_{L(s)}(s, t)$. The use of different affinities for different objects has a long history in Fuzzy Connectedness algorithms, as described in [8]. In fact, more recently, it has been proven that OIFT algorithm belongs to a class of General Fuzzy Connectedness algorithms and so, it has several good theoretical properties, like robustness for seed placement [14]. Algorithm 1 is also equivalent (in a sense defined in [10] and [11]) to a generalized Dijkstra algorithm from [7] used with an appropriate path cost function, being the map V related to the cost of paths [31]. As pointed in [39], OIFT algorithm is also related to directed connected filters. In particular, the marker based segmentation from [39], which takes the union of the selected directed connected components, corresponds to a segmentation by Oriented Relative Fuzzy Connectedness (ORFC) [1, 4]. ORFC is another Max-Min optimizer, whose result is always contained in the segmentation by OIFT [45, 46].

For a detailed example of the step-by-step execution of Algorithm 1, the reader should refer to [14].

To explore the boundary polarity in monochromatic images, the arcs weights $\omega(s, t)$ are defined as a combination of an undirected dissimilarity measure $\psi(s, t) \geq 0$ between neighboring pixels s and t , multiplied by an orientation factor, as follows:

$$\omega(s, t) = \begin{cases} \psi(s, t) \times (1 + \alpha) & \text{if } I(s) > I(t), \\ \psi(s, t) \times (1 - \alpha) & \text{if } I(s) < I(t), \\ \psi(s, t) & \text{otherwise,} \end{cases} \quad (2)$$

where $\alpha \in [-1, 1]$ and we usually have $\psi(s, t) = |I(t) - I(s)|$. Other options for $\psi(s, t)$ are discussed in [10, 13]. Note that, in general, we have $\omega(s, t) \neq \omega(t, s)$ for $\alpha \neq 0$. For $\alpha > 0$, the segmentation by OIFT favors transitions from bright to dark pixels, and $\alpha < 0$ favors the opposite orientation. To understand this, note that, for a given labeling L , if $I(s) > I(t)$ for all arcs (s, t) such that $L(s) > L(t)$, then increasing α will increase $\varepsilon_{\min}(L)$. On the other hand, if $I(s) < I(t)$ for all arcs (s, t) such that $L(s) > L(t)$, then decreasing α will increase $\varepsilon_{\min}(L)$. For the case of polarity involving color images, the reader should refer to [31].

Finally, as presented in the next section, for multi-object segmentation we consider multiples α_i values, each associated to a different object.

3 Hierarchical Layered OIFT

We propose the **Hierarchical Layered Oriented Image Foresting Transform** (HLOIFT) as a new seed-based method for multi-object segmentation. Figure 2 shows an overview of our framework. For a given input image, seeds sets for some objects, and the tree of relations between objects, the HLOIFT method has the following steps: (1) Each layer is constructed as a weighted digraph representing one object with its own priors (described in Section 3.2). (2) HLOIFT defines a setup for the inter-layer connections representing the hierarchical constraints, such as inclusion and exclusion relations (described in Section 3.3). (3) HLOIFT uses an extension of the OIFT algorithm to compute an optimal cut over the hierarchical layered digraph, giving as output a labeled image (described in Section 3.4).

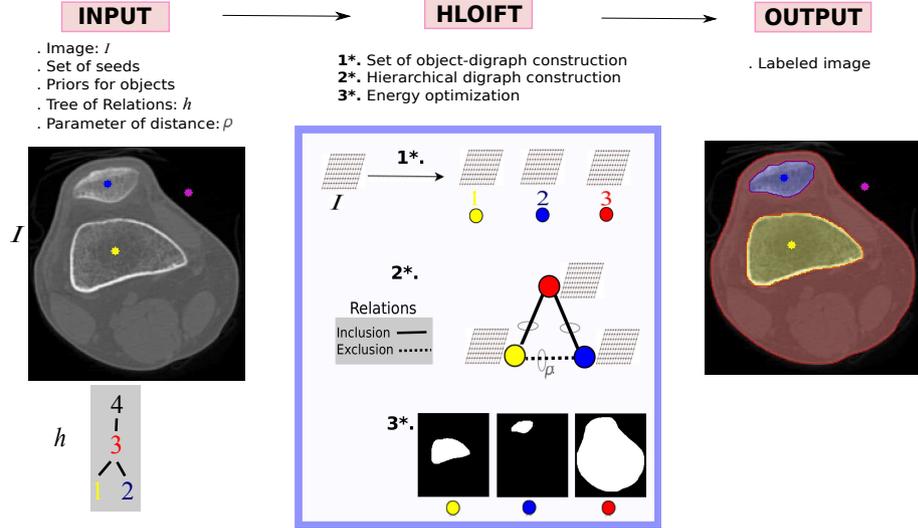


Figure 2: Overview of our framework. Given the input parameters, a hierarchical weighted digraph of layers (digraphs) is constructed using the *inclusion* (solid line) and *exclusion* (dashed line) hierarchical constraints between objects, and a graph-cut measure is optimized by our algorithm. Finally, we have a labeled image as output. (Compare to Figure 9. Notice, that only three seeds are used.)

3.1 Notations and definitions

Let $\mathcal{L} = \{1, \dots, m\}$ denote an index set, where each element in \mathcal{L} is associated with an object we consider and m is the number of objects to be segmented. The HLOIFT graph associated with \mathcal{L} and image (\mathcal{J}, I) will be defined on the set of nodes $\mathcal{N} = \mathcal{L} \times \mathcal{J}$. The HLOIFT resulted segmentation of the image will be identified with a binary variable $X: \mathcal{N} \rightarrow \{0, 1\}$, where, for $i \in \mathcal{L}$, the i th object O_i and the background O_0 are defined, respectively, as

$$O_i = \{t \in \mathcal{J}: X(i, t) = 1\} \quad \text{and} \quad O_0 = \mathcal{J} \setminus \bigcup_{i \in \mathcal{L}} O_i. \quad (3)$$

Each object/background object O_i , $i \in \mathcal{L} \cup \{0\}$, will be identified with a corresponding set $\mathcal{S}_i \subset \mathcal{J}$ of seeds, aiming for $\mathcal{S}_i \subseteq O_i$.

The hierarchy between the objects is understood as a prior knowledge on any pair $\langle O_i, O_j \rangle$ of objects we consider: either $O_i \cap O_j = \emptyset$, or one of them is properly contained in the other. This prior is represented as a function $h: \mathcal{L} \rightarrow \{1, \dots, m+1\}$, referred to as a *tree*, and defined as follows. If $O_{m+1} = \mathcal{J}$ (the image domain and the root of the tree), then $h(i) = j$ if, and only if, O_j is the smallest of the objects properly containing O_i . If $h(i) = j$, then we will refer to O_j as the *parent* of O_i (or say that the tree node j is the parent of i). See the left lower part of Figure 2 for a graphical representation of h . We say that the objects O_i and O_j (or just nodes i and j) are *siblings*, provided $i, j \in \mathcal{L}$ are distinct and $h(i) = h(j)$. In particular, any sibling objects are disjoint. Only the seed sets \mathcal{S}_i for the leaves in the tree of relations and background must be non-empty, since nested boundaries can be discovered by the inclusion relation as will be demonstrated in the experiments.

In addition, we will use the distance parameter $\rho \geq 0$, which indicates the minimum distance between the boundaries of siblings and of the parent-offspring pairs of objects. More specifically, we have the following consistency condition:

- (C) For siblings O_i and O_j we will assume that $\|s - t\| > \rho$ for every $s \in O_i$ and $t \in O_j$, while for parent-offspring pair $\langle O_j, O_i \rangle$ (i.e., with $h(i) = j$) that $t \in O_j$ whenever there exists an $s \in O_i$ with $\|s - t\| \leq \rho$.

The hierarchy h , with no parameter ρ , is understood as the consistency condition (C) with $\rho = 0$. For $\rho > 0$, we require that the assumptions of h are, in a sense, strongly satisfied.

3.2 Construction of the layers of our digraph

The first step of HLOIFT is to create a set of m layers, where each layer \mathcal{H}_i , $i \in \mathcal{L}$, is used to represent a single corresponding object O_i . A layer $\mathcal{H}_i = (\mathcal{N}_i, \mathcal{A}_i, \omega_i)$ is a weighted digraph, where $\mathcal{N}_i = \{i\} \times \mathcal{J}$ and each node $t = (i, v) \in \mathcal{N}_i$ corresponds to the image pixel $p(t) = v$. Thus, the node set \mathcal{N} of HLOIFT digraph is defined as $\mathcal{L} \times \mathcal{J} = \bigcup_{i \in \mathcal{L}} \mathcal{N}_i$ and $p: \mathcal{N} \rightarrow \mathcal{J}$ is the projection onto the second coordinate. See graphical representation of layers in Figures 4, 5, and 6. Also, $\lambda: \mathcal{L} \times \mathcal{J} \rightarrow \mathcal{L}$ will denote the projection onto the first coordinate, that is, $\lambda(t) = i$ means that t belongs to the i th layer of the graph.

Let $\mathcal{A}_\mathcal{J}$ be an affinity defined on the original image graph (with $\mathcal{N} = \mathcal{J}$) as in the case of OIFT. We will usually assume that, in the case of 2D images, $\mathcal{A}_\mathcal{J}$ is the 4- or 8-neighborhood adjacency. We define the intra-layer adjacency \mathcal{A}_i on \mathcal{N}_i as this $\mathcal{A}_\mathcal{J}$, that is, $(s, t) \in \mathcal{A}_i$ if, and only if, $(p(s), p(t)) \in \mathcal{A}_\mathcal{J}$. Similarly, if for an i th object we already have defined a weight function ω_i on the image (depending on the image intensities and, in some cases, according to a given higher level prior), then the intra-layer weight function ω_i is defined, for every $(s, t) \in \mathcal{A}_i$, as $\omega_i(s, t) = \omega_i(p(s), p(t))$. For HLOIFT to work properly we need to assume that the values of these intra-layer weight functions are the *finite* real numbers (unless they represent hard constraints, see star convexity constraint next), as opposed to the inter-layer weight, defined below, which will have infinite values.

Of course, ω_i should highlight the desired boundaries for O_i as clearly as possible and we would like to incorporate in its definition the higher level priors whenever it is appropriate. In particular, to utilize the object-contour orientations, that is, the **boundary polarity priors**, we use in our experiments the same scheme that was adopted by the regular OIFT method, defining ω_i via formula (2) with $\psi(s, t) = |I(s) - I(t)|$. (For other possible definitions of ψ see, e.g., [10, 13].) In this setting, each object O_i has its own constant $\alpha_i \in [-1, 1]$ (used in the expression (2)), so that we can favor the segmentation of O_i with transitions from *bright to dark* pixels with $\alpha_i > 0$, or the opposite orientation, with $\alpha_i < 0$. Note that $\alpha_i = 0$ can be used to indicate that O_i has no boundary polarity prior.

If the i th layer is created using the **Geodesic Star Convexity** prior (GSC), then we will be prioritizing the segmentation of O_i with more regular shape. A common shape prior is the *star shape prior* (SSP) [53], where a region has a star shape with respect to a given center c if for any point p inside the region, all points on the straight line segment between the center c and p also lie inside the region (Figures 3a-b). The *geodesic star convexity prior* (GSC) corresponds to a discrete version of SSP directly in the image domain, by considering shortest paths in the image graph, returned by IFT with the additive path-cost function (geodesic), as the line segments (Figure 3c). For each object, it considers all the given foreground seeds as center points, prioritizing the segmentation of a region with more regular shape. The geodesic star convexity prior for the i th layer is obtained by setting the weights ω_i of some arcs in \mathcal{H}_i to $-\infty$, according to the scheme proposed in [30] (Figure 3c). Moreover, it is still possible to simultaneously handle boundary polarity and shape priors [32].

3.3 Hierarchical digraph construction (setup of inter-layer connections)

In this step, HLOIFT generates a *hierarchical layered* weighted digraph \mathcal{H} as the union of all layered graphs \mathcal{H}_i , $i = 1, \dots, m$, with additional *inter-layer* arcs connecting only some of the distinct layers. The choice of these arcs, as well as their weights chosen among $-\infty$ and ∞ , is subjugated to a sole requirement of ensuring the agreement of a resulted segmentation with assumed hierarchical priors h and its related parameter ρ .

Specifically, the vertices $s \in \mathcal{N}_i$ and $t \in \mathcal{N}_j$, with distinct $i, j \in \mathcal{L}$, form an inter-layer arc whenever, according to the hierarchy consistency condition (C), a decision whether one of them belongs (or not) to the object in question may force such decision for the other vertex. Formally, this means that a pair $\langle s, t \rangle$ is an inter-layer arc in \mathcal{H} (i.e., $\langle s, t \rangle$ belongs to \mathcal{A} of \mathcal{H}) if, and only if, $\|p(s) - p(t)\| \leq \rho$ and either O_i and O_j are siblings, or one of them is the parent of the other. The set of all inter-layer arcs coming from the siblings is denoted by \mathcal{A}_s , while \mathcal{A}_p will stand for those associated with a parent/offspring pair. The weights of these inter-layer arcs $\langle s, t \rangle$, with $s \in \mathcal{N}_i$ and $t \in \mathcal{N}_j$ (or, equivalently, with $\lambda(s) = i$ and $\lambda(t) = j$), are defined as follows.

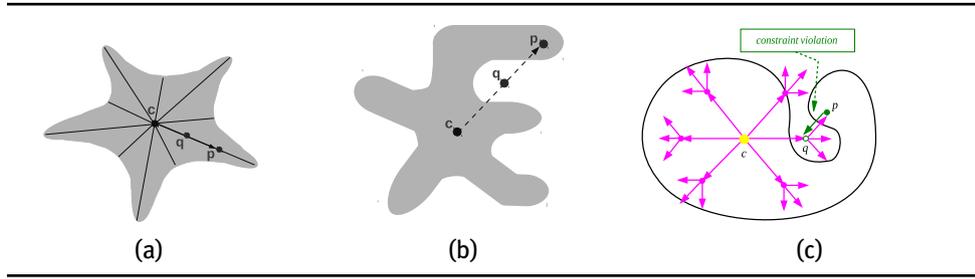


Figure 3: (a) A star shape object. (b) Violation of the star shape prior: Note that point q (in the straight line segment between the center c and p) is lying outside the object. (c) The geodesic star convexity prior: The line segments are approximated by geodesic paths from the center point c , as shown in magenta color, forming a geodesic tree (or a geodesic forest in case of multiple centers). A GSC constraint violation occurs for the object O_i when we have an arc (p, q) such that $p \in O_i$, $q \notin O_i$ and q is the predecessor of p in the geodesic forest. To implement the GSC constraint, we must set $\omega_i(p, q) = -\infty$ for such arcs, where q is the predecessor of p in the geodesic forest.

- **Inclusion.** If O_j is the parent of O_i (i.e., $h(i) = j$), then we define $\omega(t, s) = \infty$ and $\omega(s, t) = -\infty$. Figure 4 shows, in this case, the inter-layer arcs and their weights, with parameter $\rho = 1$, while the intra-layer arcs are formed according to a 4-neighborhood adjacency.

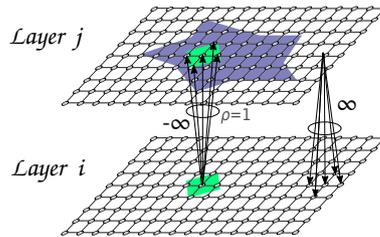


Figure 4: Illustration of inter-layer arc construction, involving two objects O_i and O_j , for the inclusion case (O_j is the parent of O_i , i.e., $h(i) = j$), where $\omega(s, t) = -\infty$ and $\omega(t, s) = \infty$ for $\lambda(s) = i$ and $\lambda(t) = j$.

- **Exclusion.** If the objects O_i and O_j are siblings, then we put $\omega(s, t) = \omega(t, s) = -\infty$. See Figure 5.

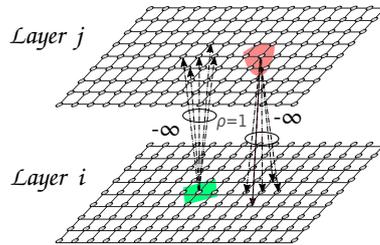


Figure 5: Illustration of inter-layer arc construction, involving two objects O_i and O_j , for the exclusion case (O_i and O_j are siblings), where $\omega(s, t) = -\infty$ for all inter-layer arcs (s, t) .

Notice that, in the inclusion case, the pair $\langle s, t \rangle$ contradicts the consistency requirement (C) if, and only if, the associated cost of the arc in the cut has value $-\infty$. This will be used in the specific formulation of our algorithm HLOIFT (especially extra condition in line 13) and in the proof of our correctness theorem. Basically, the weight $-\infty$ enforces that the vertex connected to one just processed will be examined with the highest possible priority, to ensure satisfaction of the consistency requirement (C).

In Figure 6, we give an example of the hierarchical layered digraph construction for a case that cannot be optimized under the graph cut framework [15]. This case combines the inclusion and exclusion hierarchical constraints, such that we have two mutually exclusive objects O_i and O_j , both contained within another object O_k . Globally optimal segmentation in this case cannot be modeled with graph cuts, because it cannot be converted to a submodular energy, more details in [15]. The proposed HLOIFT method can compute globally optimal results in this case and also in other more sophisticated cases, as described below.

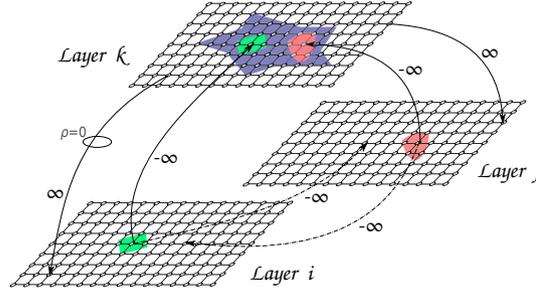


Figure 6: Illustration of the inter-layer arc construction, involving three objects O_i , O_j and O_k , where O_k is the parent of two sibling objects, O_i and O_j , i.e. $h(i) = h(j) = k$.

3.4 Energy optimization

In the final step, we execute the HLOIFT algorithm, which constitutes a modified OIFT algorithm from [31, 35]. It is applied to the hierarchical layered graph \mathcal{H} constructed above and its output *maximizes* a single energy ε_{min}^h , a modification of energy $\varepsilon_{min}(L) = \min\{\omega(s, t) : (s, t) \in \mathcal{A} \ \& \ L(s) > L(t) = 0\}$ defined in (1). This modification of ε_{min} is designed to ensure that the output satisfies also the hierarchical constraints imposed by h and ρ .

Specifically, for a binary map $X: \mathcal{N} \rightarrow \{0, 1\}$ described in Section 3.1 the energy ε_{min}^h of X is defined as

$$\varepsilon_{min}^h(X) = \min\{\varepsilon_{min}^{incl}(X), \varepsilon_{min}^{excl}(X)\}, \quad (4)$$

where

$$\varepsilon_{min}^{incl}(X) = \min\{\omega(s, t) : (s, t) \in \mathcal{A} \setminus \mathcal{A}_s \ \& \ X(s) > X(t)\},$$

$$\varepsilon_{min}^{excl}(X) = \min\{\omega(s, t) : (s, t) \in \mathcal{A}_s \ \& \ X(s) = X(t) = 1\},$$

when the set $\{(s, t) \in \mathcal{A}_s : X(s) = X(t) = 1\}$ is empty, $\varepsilon_{min}^{excl}(X)$ is defined as ∞ .

In other words, all arcs in the cut, except for the inter-layer arcs \mathcal{A}_s associated to siblings, are treated directionally, in the same way as in ε_{min} . This part is expressed by the energy ε_{min}^{incl} . However, the inter-layer arcs from \mathcal{A}_s must be treated differently. First of all, there is no directional information between them. Moreover, the siblings s and t influence one another, according to (C), only when they both belong to their respective objects, that is, when $X(s) = X(t) = 1$. This justifies the format of ε_{min}^{excl} . The formula for the combined energy ε_{min}^h is defined as a minimum of the costs of all relevant arcs in the cut, the same way as in ε_{min} (only the meaning of the term “relevant” being changed).

We say that a sequence $\langle \mathcal{S}_0, \dots, \mathcal{S}_m \rangle$ of seed sets is *consistent with (C)* (with respect to h and ρ) provided $\mathcal{S}_i \neq \emptyset$ for $i = 0$ and every $i \in \mathcal{L}$ with no offsprings and there exists a sequence $\langle O_0, \dots, O_m \rangle$ satisfying (C) such that $\mathcal{S}_i \subseteq O_i \subset \mathcal{J}$ for every $i \in \mathcal{L} \cup \{0\}$.

Theorem 1 (Cut optimality by HLOIFT). *For every image (J, I) , a hierarchy tree h , a distance parameter $\rho \geq 0$, and a sequence $\langle \mathcal{S}_0, \dots, \mathcal{S}_m \rangle$ of seed sets consistent with (C), the binary map $X: \mathcal{N} \rightarrow \{0, 1\}$ computed by Algorithm 2 maximizes the energy $\varepsilon_{\min}^h(X)$ given by (4) among all solutions satisfying the seed constraints and the consistency requirement (C).*

Algorithm 2. – HLOIFT ALGORITHM

INPUT: Hierarchical layered digraph $\mathcal{H} = (\mathcal{N}, \mathcal{A}, \omega)$, builded from an image according to the tree h of hierarchical constraints and the distance parameter ρ ; the seed sets $\langle \mathcal{S}_0, \dots, \mathcal{S}_m \rangle$ consistent with (C).

OUTPUT: The binary map $X: \mathcal{N} \rightarrow \{0, 1\}$ identifying segmentation given by (3).

AUXILIARY: Priority queue Q , variable tmp , the cost map $V: \mathcal{N} \rightarrow [-\infty, \infty]$, and an array of status $S: \mathcal{N} \rightarrow \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.

```

1. For each  $t \in \mathcal{N}$  and  $i \in \mathcal{L}$  do
2.   Set  $S(t) \leftarrow 0$  and  $V(t) \leftarrow \infty$ ;
3.   If  $p(t) \in \mathcal{S}_0$  then
4.      $V(t) \leftarrow -\infty$ ,  $X(t) \leftarrow 0$ , and insert  $t$  in  $Q$ ;
5.   If  $p(t) \in \mathcal{S}_i$  and  $\lambda(t) = i$  then
6.      $V(t) \leftarrow -\infty$ ,  $X(t) \leftarrow 1$ , and insert  $t$  in  $Q$ .
7. While  $Q \neq \emptyset$  do
8.   Remove  $s$  from  $Q$  such that  $V(s)$  is minimum;
9.   Set  $S(s) \leftarrow 1$ ;
10.  For each  $(s, t) \in \mathcal{A}$  such that  $S(t) = 0$  do
11.    If  $X(s) = 1$  then  $tmp \leftarrow \omega(s, t)$ 
12.    Else  $tmp \leftarrow \omega(t, s)$ ;
13.    If  $\neg [(s, t) \in \mathcal{A}_s \text{ and } X(s) = 0]$ , then
14.      If  $tmp < V(t)$ , then
15.        Set  $V(t) \leftarrow tmp$ ;
16.        Set  $X(t) \leftarrow X(s)$ ;
17.        If  $t \notin Q$  then insert  $t$  in  $Q$ ;
18.        If  $(s, t) \in \mathcal{A}_s$  and  $X(s) = 1$  then
19.           $X(t) \leftarrow 0$ .

```

In addition to the different input graph (as described in Sections 3.2 and 3.3), the main remaining differences from Algorithm 2 to Algorithm 1 correspond to lines 18-19 and the extra condition on line 13. More specifically, if we have only one object (i.e., $\mathcal{L} = \{1\}$), then $X = L$ and the condition from line 18 is never satisfied, so lines 18-19 can be removed. More generally, lines 18-19 and the extra condition on line 13 are redundant when tree h has no siblings. In this case, the optimality of HLOIFT regarding the maximization of $\varepsilon_{\min}^h(X)$ can be deducted from Theorem 2 from [31].

In line 18 the condition is satisfied when s and t are siblings and $p(s) \in O_{\lambda(s)}$ as $X(s) = 1$. Therefore, in line 19 we impose $X(t) = 0$, to ensure that $p(t) \notin O_{\lambda(t)}$, as required by (C). More specifically, lines 18-19 ensure that for the output X we have no $(s, t) \in \mathcal{A}_s$ with $X(s) = X(t) = 1$, that is, $\varepsilon_{\min}^{excl}(X) = -\infty$.

Concerning the computational complexity, HLOIFT is $\mathcal{O}(M + N)$, where N is the number of vertices in the graph \mathcal{H} and M is the number of arcs in the graph \mathcal{H} , when \mathcal{Q} is implemented using bucket sorting [18] and $\mathcal{O}(M + N \log N)$ (linearithmic time) if \mathcal{Q} is a heap. The Graph cut computational complexity is $\mathcal{O}(\sqrt{M} * N^2) = \mathcal{O}(N^{2.5})$ when \mathcal{H} is a sparse graph, which is more than quadratic-time using a push-relabel based on the highest label node selection rule [3].

As a final note, observe that an interesting variation of Algorithm 2 could be considered. Specifically, in the case of inclusion relation $h(i) = j$, we can additionally require that the seeds in \mathcal{S}_j belong exclusively to the parent object (i.e., $O_i \cap \mathcal{S}_j = \emptyset$). In other words, \mathcal{S}_j should be inserted as background seeds on its children's layers to additionally impose that the seeds in \mathcal{S}_j do not belong to the offspring objects of j (i.e., for all pixels $s \in \mathcal{S}_j$ we must set $X(t) = 0$ and must insert t in \mathcal{Q} with $V(t) = -\infty$, where $t = (i, s) \in \mathcal{N}_i$).

4 Proof of Theorem 1

Since the digraph \mathcal{H} is built from an image, it is symmetric and connected. So, it is easy to see that upon its termination HLOIFT returns X defined on the entire set \mathcal{N} . Therefore, to finish the proof, we need only to show that such X satisfies the seed constraints and the consistency requirements (C) with maximum energy.

All notions considered below are based on the variables from Algorithm 2. In particular, the values of these variables may change during the execution of the algorithm.

Define $D = \{t \in \mathcal{N} : S(t) = 1\}$. Then, at any time, the partial binary map X is already defined on D and, for every $t \in D$, the value of $X(t)$ remains unchanged during the rest of the execution of the algorithm. (Such change can occur only during the execution of lines 16 or 19, and it can be applied only for t for which, in line 10, we had $S(t) = 0$, that is, $t \notin D$.)

Notice also that if t in Q is not a seed, then there exists an $s \in D$ such that $(s, t) \in \mathcal{A}$, $X(t) = X(s)$, and $V(t) = w_{X(s)}(s, t)$, where we use notation w_i from Remark 1. In addition, the value $V(t)$ represents a “penalty” (i.e., a contribution to the final energy ε_{min}^h cost) of a potential change of the value $X(t)$ from its current value $X(s)$ to $1 - X(s)$.

Directly after the execution of the initialization loop 1-6, the queue Q will consist precisely of the nodes from the set $\mathcal{S} = \{(i, s) : i \in \mathcal{L} \ \& \ s \in \mathcal{S}_i\} \cup \{(i, s) : i \in \mathcal{L} \ \& \ s \in \mathcal{S}_0\}$, that is, seed representations in the HLOIFT graph \mathcal{N} . Let X_0 denote the map X defined on \mathcal{S} at this time. Since, at this time, we also have $V(t) = -\infty$ for every $t \in \mathcal{S}$, these values cannot ever be changed, as the condition in line 14 is never satisfied. Also, for any $X : \mathcal{N} \rightarrow \{0, 1\}$,

X satisfies the seed constraints if, and only if, X extends X_0 .

In particular, any $X : \mathcal{N} \rightarrow \{0, 1\}$ returned by HLOIFT has this property. Therefore, to finish the proof, it is enough to show that such X also satisfies the consistency requirements (C) with maximum energy.

By consistency of the seeds with (C), the family \mathcal{X} of all $X : \mathcal{N} \rightarrow \{0, 1\}$ satisfying the seed constraints and the consistency requirement (C) is non-empty. In particular, any $X \in \mathcal{X}$ extends X_0 and has energy $\varepsilon_{min}^h(X) > -\infty$ (as, by the choice of inter-layer weights, $\varepsilon_{min}^h(X) = -\infty$ only when X fails the consistency requirement (C)). Let E be the maximum of $\varepsilon_{min}^h(X)$ over all $X \in \mathcal{X}$ and let \mathcal{X}_0 be the family of all $X \in \mathcal{X}$ with $\varepsilon_{min}^h(X) = E$. Notice that $E > -\infty$.

The two key steps of our proof are the following closely related preservation properties (P) and (Q):

- (P) Assume that, directly before an execution of the line 8, the current X (defined on a $D \subset \mathcal{N}$) is a restriction of an $\bar{X} \in \mathcal{X}_0$. If, during the following execution of line 8, we remove t from Q with $V(t) < E$, then the extension of X to $D \cup \{t\}$ also agrees with \bar{X} , that is, $X(t) = \bar{X}(t)$.
- (Q) Assume that, directly before an execution of the line 8, the current X (defined on a $D \subset \mathcal{N}$) can be extended to an $\bar{X} \in \mathcal{X}_0$. If, during the following execution of line 8, we remove t from Q with $V(t) \geq E$, then the extension X to $D \cup \{t\}$ has a further (not necessarily proper) extension to an $\hat{X} \in \mathcal{X}_0$.

The above discussion shows that the assumptions of both of these properties are satisfied directly after the execution of the initialization loop. It is also clear that their satisfaction at algorithm termination ensures that the conclusion Theorem 1 is satisfied. Indeed, at that time, by (P) and (Q), the “partial” map X from D into $\{0, 1\}$ has an extension to an $\bar{X} \in \mathcal{X}_0$ (which by definition of \mathcal{X}_0 satisfies consistency requirement (C) and, among such maps, maximizes energy ε_{min}^h) while, as we noticed earlier, the algorithm does not stop until $D = \mathcal{N}$ (implying that $X = \bar{X} \in \mathcal{X}_0$). Therefore, to finish the proof of the theorem it is enough to show that (P) and (Q) are preserved during the execution of the algorithm.

Proof of preservation of the property (P). Indeed, if $t \in \mathcal{S}$, then, as we argued above, $\bar{X}(t) = X_0(t) = X(t)$. Otherwise, there is an $s \in D$ such that $(s, t) \in \mathcal{A}$ and the current values of $V(t)$ and $X(t)$ were fixed during the execution of lines 8-19, when our node s was removed from Q . This means that we could not have simultaneously $(s, t) \in \mathcal{A}_s$ and $X(s) = 0$.

If $(s, t) \in \mathcal{A}_s$ and $X(s) = 1$, then, during the execution of line 19, we have set $X(t) = 0$. At the same time, we must have $\bar{X}(t) = 0$, since otherwise we would have $\bar{X}(t) = 1 = X(s) = \bar{X}(s)$, making $\varepsilon_{\min}^h(\bar{X}) = \varepsilon_{\min}^{excl}(\bar{X}) = -\infty \neq E$, a contradiction. So, indeed, $X(t) = 0 = \bar{X}(t)$.

To finish the argument for (P) we can assume that $(s, t) \notin \mathcal{A}_s$. Then, the executions of lines 8-19 ensures that $X(t) = X(s)$. At the same time we must have that $\bar{X}(s) = \bar{X}(t)$, as otherwise

$$E = \varepsilon_{\min}^h(\bar{X}) \leq \varepsilon_{\min}^{incl}(\bar{X}) \leq w_{\bar{X}(s)}(s, t) = w_{X(s)}(s, t) = V(t) < E,$$

impossibility. Therefore, indeed $X(t) = X(s) = \bar{X}(s) = \bar{X}(t)$, finishing the argument for (P). \square

Proof of preservation of the property (Q). We have $V(t) > -\infty$, as $V(t) \geq E > -\infty$. For the main part of the argument for (Q) assume that we also have $V(t) < \infty$.

As $V(t)$ is finite, there must exist an $s \in D$ such that $(s, t) \in \mathcal{A}$ is an intra-arc and the current values of $V(t)$ and $X(t)$ were fixed during the execution of lines 8-19, when our node s was removed from Q . We need to show that the extension X to $D \cup \{t\}$ has a further extension to an $\hat{X} \in \mathcal{X}_0$. For $i \in \{0, 1\}$ let $R_i = \{d \in D : X(d) = i\}$.

If $X(t) = 0$, then the indicator (i.e., characteristic) function χ_{R_1} is the desired $\hat{X} \in \mathcal{X}_0$ extending X . Indeed, it certainly extends X (which, in turn, extends X_0) and has an energy $\varepsilon_{\min}^h(\hat{X}) = \min\{\varepsilon_{\min}^{incl}(\hat{X}), \varepsilon_{\min}^{excl}(\bar{X})\} \geq E > -\infty$, where the equation is justified by $\varepsilon_{\min}^{excl}(\bar{X}) = \varepsilon_{\min}^{excl}(\hat{X})$, while the inequality \geq by $\varepsilon_{\min}^{incl}(\hat{X}) \geq \min\{V(s) : s \in Q\} \geq V(t) \geq E$ and $\varepsilon_{\min}^{excl}(\bar{X}) \geq \varepsilon_{\min}^h(\bar{X}) = E$. Hence, $\hat{X} \in \mathcal{X}_0$.

So, assume that $X(t) = 1$ and notice that $(s, t) \notin \mathcal{A}_s$: for $X(s) = 0$ this is ensured by line 13, while for $X(s) = 1$ by line 18. In particular, we also have $X(s) = 1$. A first impulse could be to define \hat{X} as the indicator $\chi_{\mathcal{N} \setminus R_0}$. However, $\varepsilon_{\min}^h(\chi_{\mathcal{N} \setminus R_0}) \leq \varepsilon_{\min}^{excl}(\chi_{\mathcal{N} \setminus R_0})$ and it is likely that $\varepsilon_{\min}^{excl}(\chi_{\mathcal{N} \setminus R_0}) = -\infty$. So, in this case, \hat{X} needs to be defined more carefully. Specifically, let $i = \lambda(t)$ and let \mathcal{L}^i be the set of all ancestors of i : all $j \in \mathcal{L}$ such that, according to h , $O_i \subseteq O_j$. Let

$$R = R_1 \cup \{t \in \mathcal{N} \setminus R_0 : \lambda(t) \in \mathcal{L}^i\}$$

and put $\hat{X} = \chi_R$. Then, \hat{X} extends X which, in turn, extends X_0 . So, \hat{X} satisfies seeds constrains and, to show that $\hat{X} \in \mathcal{X}_0$, it is enough to prove that $\varepsilon_{\min}^h(\hat{X}) = E > -\infty$.

To see this, first notice that $\varepsilon_{\min}^{excl}(\hat{X}) = \infty$. Indeed, otherwise there would exist $u, v \in R$ with $(u, v) \in \mathcal{A}_s$ and $\hat{X}(u) = \hat{X}(v) = 1$. However, they cannot both belong to R_1 , since this would mean that $E = \varepsilon_{\min}^h(\bar{X}) = -\infty$. They cannot both belong to $R \setminus R_1$, since \mathcal{L}^i contains no siblings. Finally if one of these nodes belongs to R_1 , say $u \in R_1$, then the other cannot be in $R \setminus R_1$. Indeed, this would mean that $u \in D$ and at the time when u was removed from Q , the value $V(v) = -\infty$ was assigned. This $V(v) = -\infty$ cannot be ever increased during execution of HLOIFT. With $v \in Q$, this contradicts the minimality $V(t) \geq E$, as $E > -\infty = V(v)$. So, indeed, $\varepsilon_{\min}^{excl}(\hat{X}) = \infty$.

From this, $\varepsilon_{\min}^h(\hat{X}) = \min\{\varepsilon_{\min}^{incl}(\hat{X}), \varepsilon_{\min}^{excl}(\bar{X})\} \geq E > -\infty$, where the equation is justified by $\varepsilon_{\min}^{excl}(\hat{X}) = \infty = \varepsilon_{\min}^{excl}(\bar{X})$, while the inequality \geq by the facts that $\varepsilon_{\min}^{incl}(\hat{X}) \geq \min\{V(s) : s \in Q\} = E$ and $\varepsilon_{\min}^{excl}(\bar{X}) \geq \varepsilon_{\min}^h(\bar{X}) = E$. This completes the proof of (Q) in case $V(t) < \infty$.

To finish the proof of (Q) it is enough to show that $V(t) = \infty$ is impossible. Indeed, the consistency of the seeds ensures that either $\lambda(t)$ or some of its descendants, say j , contains a seed, say u . In particular, $s_0 = \langle j, u \rangle \in D$ and $X(s_0) = X_0(s_0) = 1$. Also, \mathcal{H} contains a path from s_0 to $s = \langle \lambda(t), u \rangle$ with each connecting arc having weight $-\infty$. Therefore, s must belong to D , as otherwise we would have $E = -\infty$. Since $s \in \mathcal{N}_{\lambda(t)}$ and $\mathcal{N}_{\lambda(t)}$ is connected and symmetric, it contains adjacent u and v with $u \in D$ and $v \in Q$. Hence, $V(t) \leq V(v) < \infty$, finishing the proof of (Q). \square

5 Experiments

In this section, an experimental evaluation of HLOIFT is presented. First, we analyze the effects of different high-level priors on the segmentation results, showing that HLOIFT is a versatile method, which allows us to customize the segmentation to some target objects according to their global properties (shape and polarity). Then, we assess the performance of HLOIFT against the previously existing method for multiple object

segmentation by IFT [34] and the segmentation of multiple objects by the hierarchical min-cut/max-flow algorithm [15], using medical and natural images. Our code and data are available upon request.

5.1 Setting the high-level priors

The proper configuration of the high-level priors for each object is essential to obtain the desired segmentation of the target objects. Figure 7 shows a two-object segmentation problem for a synthetic image composed of four nested boundaries. We present the results using one internal seed for the object O_1 , an external seed for the background, the inclusion relation $h(1) = 2$, $\rho = 1.5$ and ω_i defined via formula (2) with $\psi(s, t) = |I(s) - I(t)|$. HLOIFT with different priors for each object can provide any desired result, allowing the easy customization of the segmentation to any particular result.

In Figure 7, the superscript codes bd and db for each object O_i indicate polarity from bright to dark pixels ($\alpha_i > 0$) and polarity from dark to bright pixels ($\alpha_i < 0$), respectively, while their absences indicate neutral polarity ($\alpha_i = 0$). The superscript code g indicate the usage of the geodesic star convexity prior. Figure 7(a) shows the result when O_1 has boundary polarity setup from bright to dark and O_2 from dark to bright (i.e., O_1^{bd} and O_2^{db}). Note that in this case the addition of the GSC constraint for O_2 does not affect the result (i.e., O_1^{bd} and O_2^{g+db}). Figure 7(b) shows the result when we remove or reverse the polarity for O_1 , while Figure 7(c) reverses the polarity for O_2 . Figure 7(d) shows the result with reversed polarity for both objects in relation to Figure 7(a) and without the GSC constraint to get the stronger boundaries. In Figure 7(e), the GSC constraint is assumed for O_1 and the outermost boundary is assigned to O_2 by exclusion.

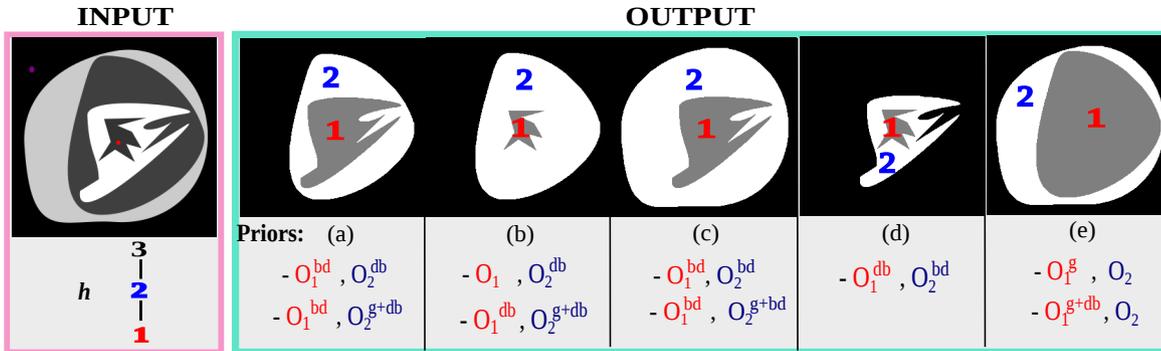


Figure 7: Example of two object segmentation by HLOIFT, where O_2 is parent of O_1 . Each object has different high-level priors –db: polarity from dark to bright pixels, bd: polarity from bright to dark pixels and g: geodesic star convexity prior.

Another important parameter is the ρ value, which controls the minimum distance between the object boundaries. In order to understand its effect on the results, Figure 8 shows HLOIFT with geodesic star convexity prior to segment an image of archaeological fragments (Figure 8a), with three sibling objects, using the simplest arc weights $\omega_i(s, t) = \|I(t) - I(s)\|$ (Figure 8b). For $\rho = 0$, we get a bad result with touching objects (Figure 8c), which can be avoided by using $\rho = 2$ (Figure 8d).

5.2 Qualitative comparison with the IFT method

In this section, we compare the results obtained by HLOIFT against the IFT method [34] for multi-object segmentation by seed competition, which encompasses the watershed transform from markers. OIFT is not included here since it is restricted only to binary segmentation [31]. We used for this experiment medical images, composed of multiple sophisticated objects.

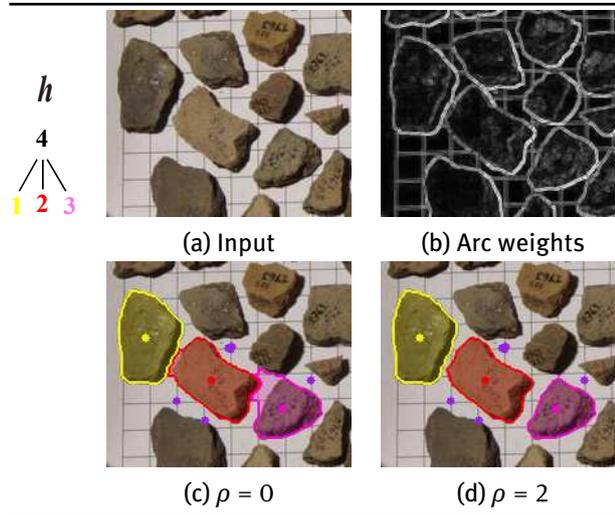


Figure 8: Example showing how changing the ρ value from 0 to 2 can improve the archaeological fragment segmentation by HLOIFT, avoiding a result with touching objects.

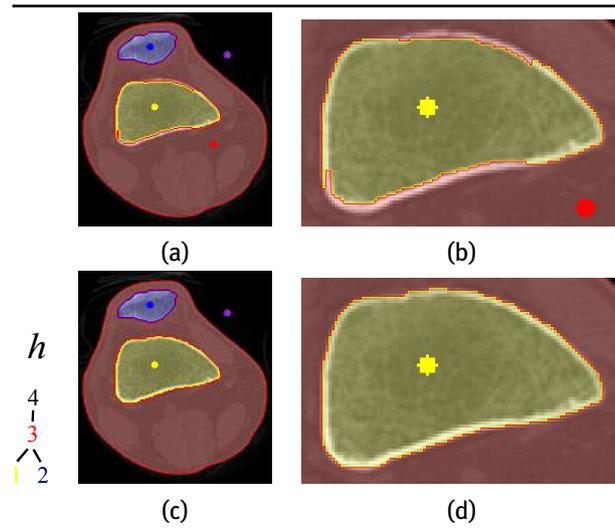


Figure 9: Knee segmentation composed of three objects in a CT image, including the patella in blue. (a-b) Result by IFT where the O_1 is mixing bright and dark boundaries. (c-d) An improved result for the central bone is obtained by HLOIFT with boundary polarity from bright to dark pixels, requiring fewer seeds.

Figure 9 shows an image segmentation task, using ω_i defined via formula (2) with $\psi(s, t) = |I(s) - I(t)|$, where the goal is to segment a gray-scale CT image composed of three objects, O_1 (central bone), O_2 (patella) and O_3 (knee), which are shown in yellow, blue and red color, respectively. The first row shows the results for the IFT method by seed competition, with user provided markers for the three objects, plus one additional external marker for the background. In this example, HLOIFT requires markers only for the two inner objects and for the background, as shown on the second row. For HLOIFT, we considered $\rho = 1.5$ and the geodesic star convexity with boundary polarity from bright to dark ($g + bd$) for all objects ($\alpha = 0.75$). In the second column, we can observe the differences highlighted in the O_1 's contour. IFT gives a poor result for O_1 , mixing bright and dark boundaries, because it does not support the boundary polarity prior. On the other hand, HLOIFT can be seen as an extension of OIFT for multiple object segmentation, allowing different priors for each object, giving better results and requiring fewer seeds.

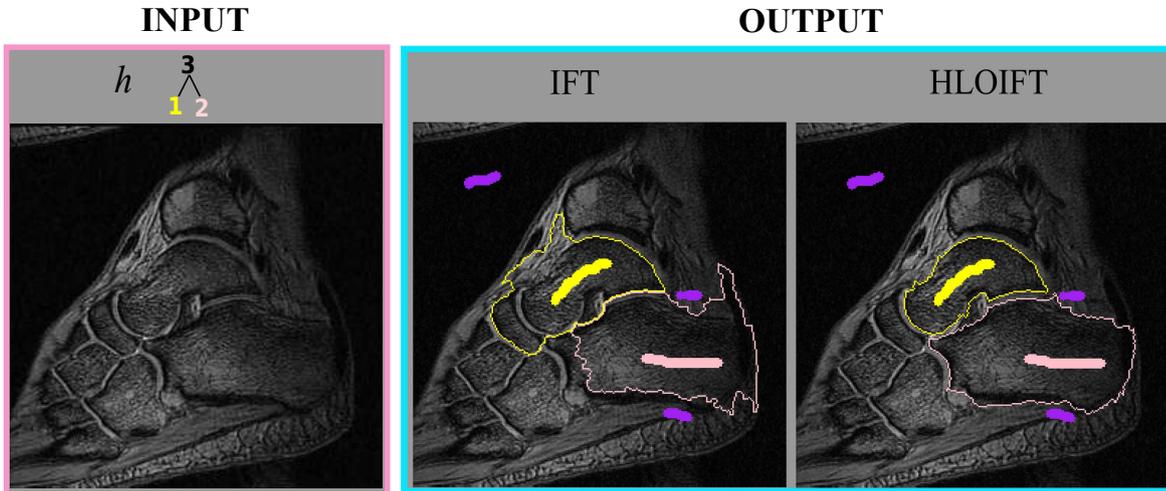


Figure 10: Talus (O_1) and calcaneus (O_2) segmentation. The two objects are sibling objects. For HLOIFT, we used $\rho = 0$, the geodesic star convexity and boundary polarity ($\alpha = -0.75$).

Figure 10 shows the segmentation of a MRI slice of a real foot, using ω_i defined via formula (2) with $\psi(s, t) = G(s) + G(t)$, where $G(\cdot)$ denotes the magnitude of the Sobel gradient. The objects of interest are the two large bones in the peritalar complex, namely, the talus (O_1) and the calcaneus (O_2). These objects are modeled as two siblings objects, i.e., both have the same parent (image domain). We used for HLOIFT, the exclusion relation $h(1) = h(2)$, $\rho = 0$, the geodesic star convexity and boundary polarity ($\alpha = -0.75$). IFT gives worse results, capturing good part of the background as part of the objects, while HLOIFT gives a more regular and correct segmentation.

Figure 11 shows a segmentation task of a CT image composed of three objects, liver (O_1), aorta (O_2) and the abdominal region (O_3), which are shown in yellow, cyan and red color, respectively. The liver (O_1) and aorta (O_2) are modeled as two sibling objects contained in the abdomen (O_3), as defined by h . We used ω_i defined via formula (2) with $\psi(s, t) = G(s) + G(t)$, where $G(\cdot)$ denotes the magnitude of the Sobel gradient. For the same user provided seeds, the IFT gets unsatisfactory results, while HLOIFT can effectively explore the inclusion and exclusion relations with $\rho = 0$, the geodesic star convexity, and the boundary polarity from bright to dark pixels ($\alpha = 0.75$) for the three objects leading to superior results.

In Figure 12, we use an axial cross section of a thoracic-abdominal CT image extracted from an open database called 3D-IRCADb-02 [43] to segment six objects: right lung (O_1), liver (O_2), heart (O_3), left lung (O_4), aorta (O_5) and the thoracic-abdominal region (O_6). As input we have the image, the tree of relations (h) and some user seeds. We used ω_i defined via formula (2) with $\psi(s, t) = |I(s) - I(t)|$. For HLOIFT, we used $\rho = 3.5$, boundary polarity from dark to bright pixels for O_1 and O_4 , shape constraint by geodesic star convexity for O_2 and O_3 , and boundary polarity from bright to dark pixels for O_5 and O_6 . It can be seen, the results obtained by HLOIFT are closer to the ground-truth compared to the IFT results.

5.3 Comparison with multi-object segmentation by min-cut/max-flow algorithm

In this section, we compare HLOIFT with the multi-object segmentation by the min-cut/max-flow algorithm in layered graphs [15]. For the sake of simplicity, we considered only the inclusion relation, since many cases combining the inclusion and exclusion hierarchical constraints (such as when we have two mutually exclusive objects O_i and O_j , both contained within another object O_k) cannot be converted to a submodular energy to perform the optimization by the graph-cut framework [15]. We used the max-flow/min-cut code from [3].

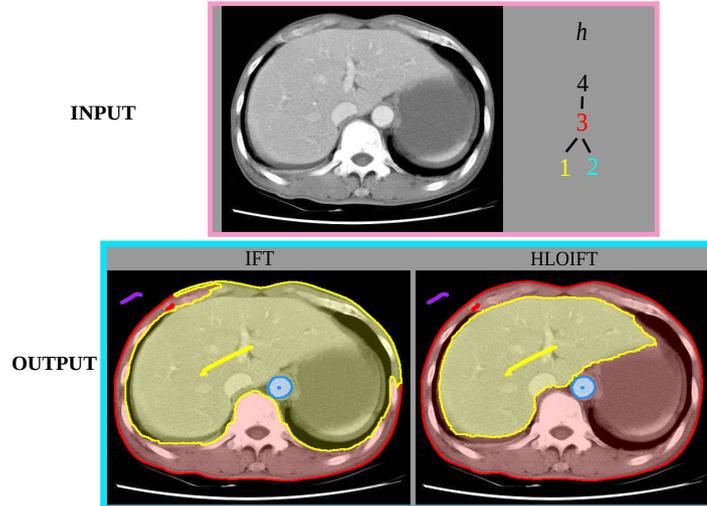


Figure 11: Liver (O_1), aorta (O_2) and abdomen (O_3) segmentation on a CT slice image. For HLOIFT, we used $\rho = 0$, $h(1) = h(2) = 3$ and the geodesic star convexity shape prior with the bright to dark boundary polarity ($\alpha = 0.75$).

Table 1: The running times for the flower segmentation by HLOIFT and the min-cut/max-flow algorithm in layered graphs using different image sizes.

| Image size (pixels) | Time of HLOIFT (ms) | Time of min-cut/max-flow (ms) |
|------------------------|------------------------|----------------------------------|
| 380×320 | 114.65 | 323.61 |
| 760×640 | 488.62 | 1,798.91 |
| 1520×1280 | 1,823.55 | 19,021.71 |

Figure 13 shows a flower segmentation using $\omega_i(s, t) = G(s) + G(t)$, where $G(\cdot)$ denotes the maximum magnitude of the Sobel gradient for the three image channels. We used $\rho = 1.5$ and no region constraints (shape constraints/boundary polarity). In general the results are similar for both methods. The min-cut/max-flow algorithm generates results with a smoother contour, but sometimes this can lead to errors in some finer parts of the object, such as the petals (Figure 13b). Besides being able to compute globally optimal results with arbitrary hierarchy constraints, the proposed HLOIFT method also has a better running time compared to the min-cut/max-flow algorithm. Table 1 shows the running times for the flower segmentation using different image resolutions in a laptop Intel Core i3-5005U CPU 2.00GHz $\times 4$.

5.4 Quantitative accuracy experiments

For quantitative accuracy results, we compared HLOIFT, with the multi-object segmentation by IFT [34], its improved version by the relaxation procedure proposed by Malmberg et al. [26], and the hierarchical layered graph cut [15, 16], denoted as HLGC, using two datasets of medical images.

For fair comparison, only methods with the same type of user input were considered here, in the form of a partial labeling.³ In order to stress the methods to their limit, in a more challenging situation, we only consider

³ Note that drawing scribbles for a partial labeling with a large brush can be quite effective to quickly mark several object/background pixels, while accurately selecting extreme boundary points requires more user attention and caution. Therefore comparing these different types of constraints is quite subjective.

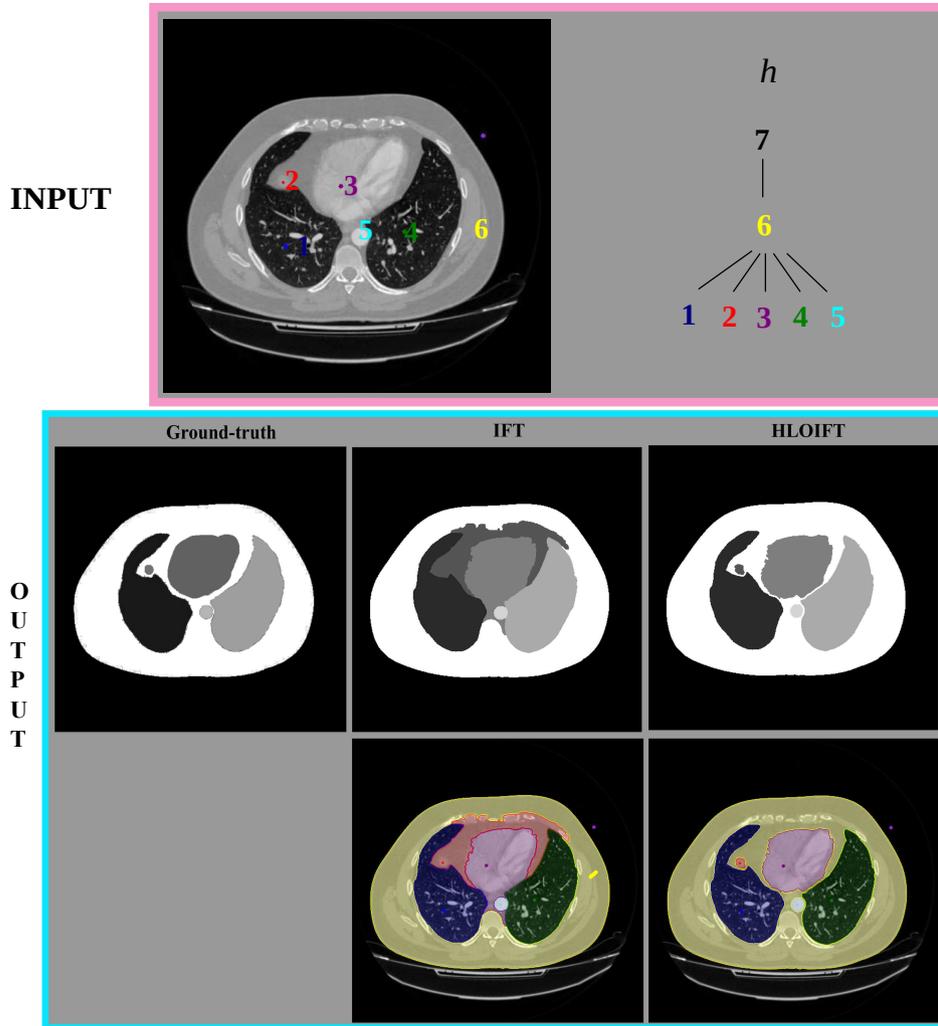


Figure 12: An axial cross section of a thoracic-abdominal CT image segmentation. We consider the following objects: right lung (O_1), liver (O_2), heart (O_3), left lung (O_4), aorta (O_5) and the thoracic-abdominal region (O_6). HLOIFT obtained a result similar to the given ground-truth (manual segmentation), in contrast to the output by IFT.

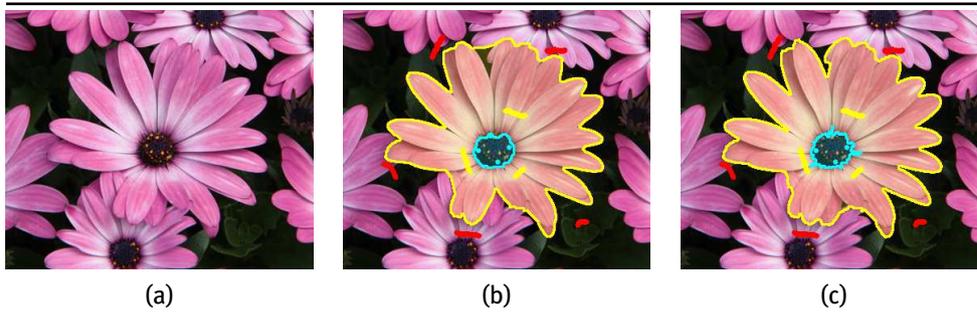


Figure 13: Flower segmentation in two objects, the central part in cyan and the petals in yellow, using the inclusion relation. (a) The input image. (b) Result by the min-cut/max-flow algorithm in layered graphs. (c) Result by HLOIFT.

simple arc weights with $\psi(s, t) = |I(s) - I(t)|$, so that no a priori information on the brightness (or color/texture) distribution was taken into account [13], nor enhancements based on deep learning techniques [55, 56].

For the first dataset, we used 40 slices from thoracic CT studies of size 512×512 to segment the liver and the abdomen as its parent object. The second dataset was composed of 40 real MR images to segment the talus and calcaneus bones as siblings, taking the foot region as their parent. The same seeds were used for all the methods, which were progressively obtained by eroding the ground truth objects and background for different radius sizes (Figure 14a). Figure 14 illustrates an example of the segmentation of the liver and abdomen for different methods. For the bones we used $\alpha = -0.5$ and $\rho = 3$ pixels, while for the liver we used $\alpha = 0.9$ and $\rho = 5$ pixels. HLGC could not exploit the exclusion relation for the bones, since it cannot be converted to a submodular energy for these sibling objects inside the foot.

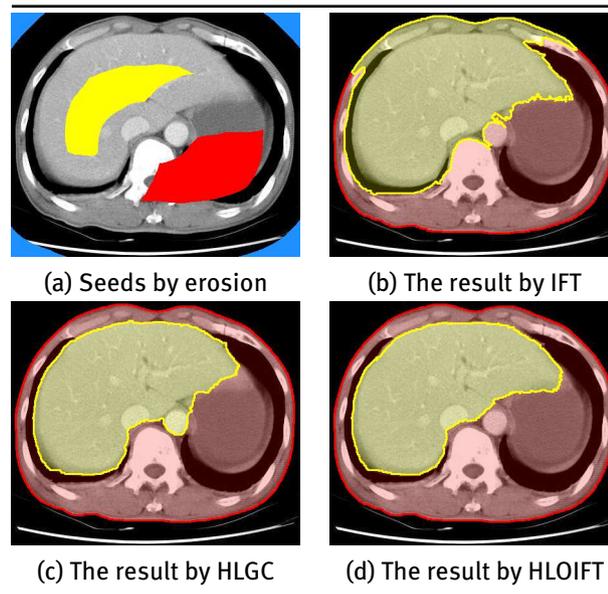


Figure 14: (a) Seeds obtained by eroding the ground truth of each object/background with 48 pixels radius. (b) Multi-object result by IFT. (c) The result by hierarchical layered graph cut (HLGC) mistakenly assigns the aorta as part of the liver. (d) A better result is obtained by HLOIFT.

In order to disambiguate the benefit coming from the boundary polarity (already present in OIFT) from the new hierarchical constraints, we also include quantitative experiments to measure which gains come from oriented boundaries and from hierarchical constraints by ablation analysis. For this purpose, we also include the results of HLOIFT with α set to 0 (no boundary polarity) and the results of considering the multi-objects segmentation problem as multiple independent binary segmentations by OIFT computed separately for each object (taking for each object its internal seeds as foreground and the other seeds as background). In the experimental curves, we denote this latter approach as OIFTs, since it represents a naive multi-objects OIFT. Note that, it can also be seen as a degenerate variant of HLOIFT without the inter-layer arcs, since there is no consistency constraints between the different binary problems. An example of the advantages of having a complete HLOIFT over these simplified versions ($\alpha = 0$ and OIFTs) for the segmentation of the liver and abdomen is shown in Figure 15.

The mean accuracy curves according to the Dice coefficient are shown in Figure 16, being the results in the left column obtained without shape constraints and the ones in the right with Geodesic Star Convexity (GSC). The accuracy curves for the parent objects (foot and abdomen) are not shown, since they had almost perfect results (Dice > 99%). HLOIFT had the best results in most cases. Note also that it could benefit more from the usage of the shape constraint by GSC for the liver compared to HLGC, since this latter approach was already producing an object with regular shape and therefore was almost not affected by the inclusion of GSC.



(a) The result by OIFTs (b) The result by HLOIFT with $\alpha = 0$ (c) The result by HLOIFT with $\alpha = 0.9$

Figure 15: (a) The result by naive multi-objects OIFT with $\alpha = 0.9$, denoted as OIFTs. (b) HLOIFT with $\alpha = 0$ (no boundary polarity). (c) A better result is obtained by HLOIFT with $\alpha = 0.9$. Note that OIFTs cannot guarantee the requirement of the minimum distance between the object boundaries as HLOIFT with $\rho > 0$.

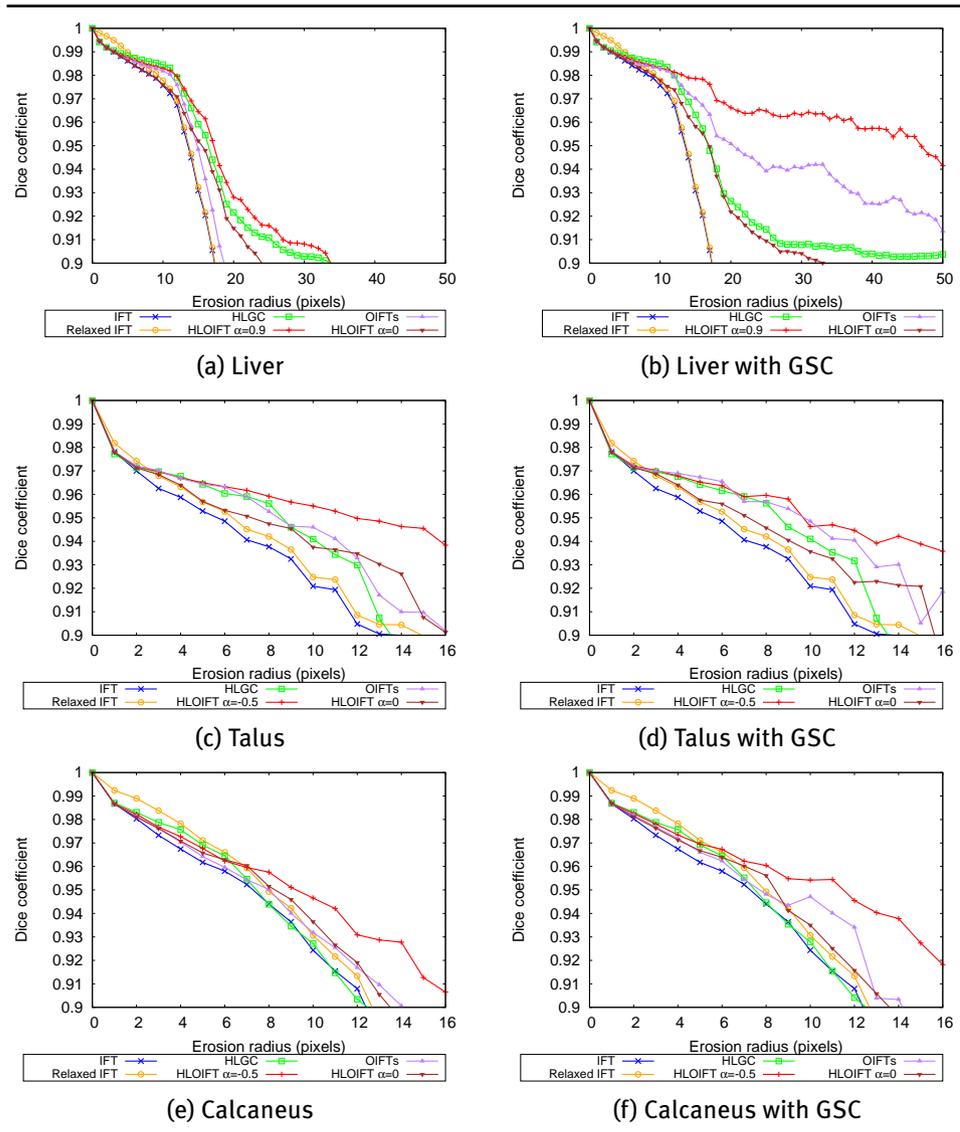


Figure 16: The mean curves of Dice accuracy for different methods.

6 Conclusion

We proposed a new graph-based algorithm, named as HLOIFT, for multi-object segmentation, allowing the integration of the individual high-level priors of each object and the geometric constraints between them into a single energy optimization. The HLOIFT algorithm was described in details, including its proof of correctness. Besides the theoretical contribution in the context of the multi-object segmentation problem, our experiments show that good segmentation results can be obtained, even when considering a simple measure of intensity dissimilarity. Besides being faster than hierarchical min-cut/max-flow based approaches, it is also less restrictive, allowing globally optimal results for arbitrary hierarchies.

As future works, we intend to combine some training information by deep learning approaches in HLOIFT, similar to what was done in [55, 56], and to incorporate in HLOIFT the relaxation procedure for directed graphs from [17], in order to circumvent the irregular boundaries from the max-norm energy of HLOIFT. We also intend to analyze and compare HLOIFT to the polynomial time graph labeling algorithms from [27]. At last, we would like also to handle relations of partial intersection by using a negative parameter ρ .

Acknowledgements: Thanks to CNPq (313554/2018-8, 486988/2013-9, 308985/2015-0, 486083/2013-6, FINEP 1266/13), FAPESP (2011/50761-2, 2016/21591-5), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and NAP eScience - PRP - USP for funding. This research is also part of the FAPESP Thematic Research Project [proc. 2014/12236-1] and part of the INCT of the Future Internet for Smart Cities funded by CNPq [proc. 465446/2014-0], CAPES [proc. 88887.136422/2017-00], and FAPESP [proc. 2014/50937-1].

References

- [1] Hans Harley Ccacyahuilca Bejar and Paulo AV Miranda. Oriented relative fuzzy connectedness: theory, algorithms, and its applications in hybrid image segmentation methods. *EURASIP Journal on Image and Video Processing*, 2015(1):21, 2015.
- [2] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.
- [3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004.
- [4] H. H. Ccacyahuilca Bejar and P. A. V. Miranda. Oriented relative fuzzy connectedness: Theory, algorithms, and applications in image segmentation. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 304–311, 2014.
- [5] K.C. Ciesielski, J.K. Udupa, A.X. Falcão, and P.A.V. Miranda. A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them. In *Proc. of SPIE on Medical Imaging: Image Processing*, volume 8314, 2012.
- [6] K.C. Ciesielski, J.K. Udupa, P.K. Saha, and Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Computer Vision and Image Understanding*, 107(3):160–182, 2007.
- [7] Krzysztof Chris Ciesielski, Alexandre Xavier Falcão, and Paulo A. V. Miranda. Path-value functions for which dijkstra’s algorithm returns optimal mapping. *Journal of Mathematical Imaging and Vision*, Feb 2018.
- [8] Krzysztof Chris Ciesielski, Gabor T. Herman, and T. Yung Kong. General theory of fuzzy connectedness segmentations. *Journal of Mathematical Imaging and Vision*, 55(3):304–342, Jul 2016.
- [9] Krzysztof Chris Ciesielski, Robin Strand, Filip Malmberg, and Punam K. Saha. Efficient algorithm for finding the exact minimum barrier distance. *Computer Vision and Image Understanding*, 123:53 – 64, 2014.
- [10] Krzysztof Chris Ciesielski and Jayaram K. Udupa. Affinity functions in fuzzy connectedness based image segmentation i: Equivalence of affinities. *Comput. Vis. Image Underst.*, 114(1):146–154, January 2010.
- [11] Krzysztof Chris Ciesielski and Jayaram K. Udupa. A framework for comparing different image segmentation methods and its use in studying equivalences between level set and fuzzy connectedness frameworks. *Computer Vision and Image Understanding*, 115(6):721 – 734, 2011.
- [12] Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):925–939, 2010.
- [13] P.A.V. de Miranda, A.X. Falcão, and J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 114(1):85 – 99, 2010.
- [14] Caio de Moraes Braz, Paulo A. V. Miranda, Krzysztof Chris Ciesielski, and Fábio A. M. Cappabianco. Optimum cuts in graphs by general fuzzy connectedness with local band constraints. *Journal of Mathematical Imaging and Vision*, 62:659-672, 2020.

- <https://doi.org/10.1007/s10851-020-00953-w>
- [15] Andrew Delong and Yuri Boykov. Globally optimal segmentation of multi-region objects. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 285–292. IEEE, 2009.
- [16] Andrew Delong, Lena Gorelick, Olga Veksler, and Yuri Boykov. Minimizing energies with hierarchical costs. *International journal of computer vision*, 100(1):38–58, 2012.
- [17] C.L. Demario and P.A.V. Miranda. Relaxed oriented image foresting transform for seeded image segmentation. In *IEEE International Conference on Image Processing (ICIP)*, pages 1520–1524, Sep 2019.
- [18] Alexandre X Falcão, Jorge Stolfi, and Roberto de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):19–29, 2004.
- [19] Leo Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006.
- [20] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proceedings of Computer Vision and Pattern Recognition*, pages 3129–3136, 2010.
- [21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.
- [22] Hossam Isack, Olga Veksler, Milan Sonka, and Yuri Boykov. Hedgehog shape priors for multi-object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2434–2442, 2016.
- [23] L. M. C. Leon and P. A. V. D. Miranda. Multi-object segmentation by hierarchical layered oriented image foresting transform. In *2017 30th SIBGRAP Conference on Graphics, Patterns and Images (SIBGRAP)*, pages 79–86, Oct 2017.
- [24] L.M. Castañeda Leon and P.A. Vechiatto de Miranda. Efficient interactive multi-object segmentation in medical images. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 705–710, Cham, 2019. Springer International Publishing.
- [25] Xiaoqiang Li, Jingsong Chen, and Huafu Fan. Interactive image segmentation based on grow cut of two scale graphs. In Wenjun Zhang, Xiaokang Yang, Zhixiang Xu, Ping An, Qizhen Liu, and Yue Lu, editors, *Advances on Digital Television and Wireless Multimedia Communications*, pages 90–95, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [26] F. Malmberg, I. Nyström, A. Mehnert, C. Engstrom, and E. Bengtsson. Relaxed image foresting transforms for interactive volume image segmentation. In *Proc.SPIE*, volume 7623, pages 7623 – 7623 – 11, 2010.
- [27] Filip Malmberg and Krzysztof Chris Ciesielski. Two polynomial time graph labeling algorithms optimizing max-norm-based objective functions. *Journal of Mathematical Imaging and Vision*, 62:737–750, Jun 2020.
- [28] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 616–625, 2018.
- [29] L. A. C. Mansilla, P. A. V. Miranda, and F. A. M. Cappabianco. Oriented image foresting transform segmentation with connectivity constraints. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2554–2558, Sept 2016.
- [30] L.A.C. Mansilla, M.P. Jackowski, and P.A.V. Miranda. Image foresting transform with geodesic star convexity for interactive image segmentation. In *IEEE International Conference on Image Processing (ICIP)*, pages 4054–4058, Melbourne, Australia, Sep 2013.
- [31] L.A.C. Mansilla and P.A.V. Miranda. Image segmentation by oriented image foresting transform: Handling ties and colored images. In *18th International Conference on Digital Signal Processing (DSP)*, pages 1–6, Santorini, Greece, Jul 2013. IEEE.
- [32] L.A.C. Mansilla and P.A.V. Miranda. Image segmentation by oriented image foresting transform with geodesic star convexity. In *Computer Analysis of Images and Patterns (CAIP)*, volume 8047, pages 572–579, York, UK, Aug 2013.
- [33] P. A. V. Miranda, A. X. Falcao, and J. K. Udupa. Cloud bank: A multiple clouds model and its use in mr brain image segmentation. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 506–509, June 2009.
- [34] Paulo AV Miranda and Alexandre X Falcão. Links between image segmentation based on optimum-path forest and minimum cut in graph. *Journal of Mathematical Imaging and Vision*, 35(2):128–142, 2009.
- [35] Paulo AV Miranda and Lucy AC Mansilla. Oriented image foresting transform segmentation by seed competition. *Image Processing, IEEE Transactions on*, 23(1):389–398, 2014.
- [36] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. CLOUDS: A model for synergistic image segmentation. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 209–212, Paris, France, May 2008.
- [37] Seyedmehrdad Mohammadianrasanani. The use of a body-wide automatic anatomy recognition system in image analysis of kidneys. Master's thesis, School of Technology and Health, Royal Institute of Technology, Flemingsberg, Sweden, 2013.
- [38] Ipek Oguz and Milan Sonka. LOGISMOS–B: layered optimal graph image segmentation of multiple objects and surfaces for the brain. *IEEE transactions on medical imaging*, 33(6):1220–1235, 2014.
- [39] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat. Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1162–1176, 2015.
- [40] Leticia Rittner, Jayaram K. Udupa, and Drew A. Torigian. Multiple fuzzy object modeling improves sensitivity in automatic anatomy recognition. In *In Proceedings of SPIE on Medical Imaging: Image Processing*, volume 9034, San Diego, California, USA, 2014.
- [41] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

- [42] Dheeraj Singaraju, Leo Grady, and René Vidal. Interactive image segmentation via minimization of quadratic energies on directed graphs. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [43] Luc Soler, Alexandre Hostettler, Vincent Agnus, Arnaud Charnoz, Jean-Baptiste Fasquel, Johan Moreau, Anne-Blandine Oswald, Mourad Bouhadjar, and Jacques Marescaux. 3D image reconstruction for comparison of algorithm database : A patient-specific anatomical and medical image database. 2012.
- [44] Kaiqiong Sun, Jayaram K. Udupa, Dewey Odhner, Yubing Tong, and Drew A. Torigian. Automatic thoracic anatomy segmentation on ct images using hierarchical fuzzy models and registration. In *In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 9036, San Diego, California, USA, 2014.
- [45] Anderson C. M. Tavares, Hans H. C. Bejar, and Paulo A. V. Miranda. Seed robustness of oriented relative fuzzy connectedness: core computation and its applications. In Martin A. Styner and Elsa D. Angelini, editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 336 – 345. International Society for Optics and Photonics, SPIE, 2017.
- [46] Anderson Carlos Moreira Tavares, Hans Harley Ccacyahuilca Bejar, and Paulo André Vechiatto Miranda. Seed robustness of oriented image foresting transform: Core computation and the robustness coefficient. In Jesús Angulo, Santiago Velasco-Forero, and Fernand Meyer, editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 119–130, Cham, 2017. Springer International Publishing.
- [47] Yubing Tong, J. K. Udupa, D. Odhner, Sanghun Sin, and R. Arens. Abdominal adiposity quantification at MRI via fuzzy model-based anatomy recognition. In *In Proceedings of SPIE on Medical Imaging: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 8672, Orlando, Florida, USA, 2013.
- [48] Jayaram K. Udupa, Dewey Odhner, Yubing Tong, Monica M. S. Matsumoto, Krzysztof C. Ciesielski, Pavithra Vaideeswaran, Victoria Ciesielski, Babak Saboury, Liming Zhao, Syedmehrdad Mohammadianrasanani, and Drew Torigian. Fuzzy model-based body-wide anatomy recognition in medical images. volume 8671, pages 86712B–86712B–7, 2013.
- [49] Jayaram K. Udupa, Dewey Odhner, Liming Zhao, Yubing Tong, Monica M.S. Matsumoto, Krzysztof C. Ciesielski, Alexandre X. Falcao, Pavithra Vaideeswaran, Victoria Ciesielski, Babak Saboury, Syedmehrdad Mohammadianrasanani, Sanghun Sin, Raanan Arens, and Drew A. Torigian. Body-wide hierarchical fuzzy modeling, recognition, and delineation of anatomy in medical images. *Medical Image Analysis*, 18(5):752 – 771, 2014.
- [50] J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, S. Mishra, G.J. Grevera, B. Saboury, and D.A. Torigian. Automatic anatomy recognition via fuzzy object models. In *In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 8316, San Diego, California, USA, 2012.
- [51] J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, P. Vaideeswaran, S. Mishra, G.J. Grevera, B. Saboury, and D.A. Torigian. Fuzzy object modeling. In *Proc. SPIE*, volume 7964, pages 7964 – 7964 – 10, 2011.
- [52] Johannes Ulén, Petter Strandmark, and Florian Kahl. An efficient optimization framework for multi-region segmentation based on lagrangian duality. *Medical Imaging, IEEE Transactions on*, 32(2):178–188, 2013.
- [53] Olga Veksler. Star shape prior for graph-cut image segmentation. *Computer Vision–ECCV 2008*, pages 454–467, 2008.
- [54] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE conference on*, pages 1–8. IEEE, 2008.
- [55] S. Wolf, L. Schott, U. Köthe, and F. Hamprecht. Learned watershed: End-to-end learning of seeded segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2030–2038, Oct 2017.
- [56] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred A. Hamprecht. The mutex watershed and its objective: Efficient, parameter-free image partitioning. Available: <https://arxiv.org/abs/1904.12654>, 2019.
- [57] Yin Yin, Xiangmin Zhang, Rachel Williams, Xiaodong Wu, Donald D Anderson, and Milan Sonka. LOGISMOS—layered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint. *IEEE transactions on medical imaging*, 29(12):2023–2037, 2010.