Efficient Hierarchical Layered Graph Approach for Multi-Region Segmentation

Leissi Margarita Castañeda Leon

Text submitted to Institute of Mathematics and Statistics from University of São Paulo for Grade of Doctor in Computer Science

Program: Computer Science

Advisor: Prof. Dr. Paulo A. Vechiatto de Miranda.

This document is presented for the author's PhD degree.

São Paulo, September 2019

Efficient Hierarchical Layered Graph Approach for Multi-Region Segmentation

This thesis version contains corrections and changes suggested by the Committee in Charge during the defense of the original version of the work, held at 03/15/2019. A copy of the original and correction versions are available at the Institute of Mathematics and Statistics of the University of São Paulo.

Committee in Charge:

- ProfTM. DrTM. Paulo André Vechiatto de Miranda IME-USP
- Prof. Dr. Fábio Augusto Menocci Cappabianco UNIFESP
- Prof. Dr. Silvio Jamil Ferzoli Guimarães PUC-MG
- Prof. Dr. Thiago Vallin Spina CNPEM
- Prof. Dr. Ronaldo Fumio Hashimoto IME-USP

Acknowledgements

I am profoundly grateful to my family, my mother Ilsa, my father Jose and brothers Roy and Edwin, who are always in my mind and my heart. To my husband Jorge for his extensive and endless support and love, and to my lovely children: Christopher and my unborn baby Fabrizio who are my main motivation in everything.

My deepest appreciation and sincere thanks to my supervisor Professor Dr. Paulo André Vechiatto de Miranda for his strong support, expertise, guidance and persistent help, greatly conducting me throughout my PhD.

I would like to thank Prof. Dr. Krzysztof Chris Ciesielski from Department of Mathematics, West Virginia University, USA, who collaborated in the formalism of our proposed method.

My gratitude to Prof. Dr. Fábio A. M. Cappabianco, Prof. Dr. Silvio J. F. Guimarães, Prof. Dr. Thiago V. Spina, and Prof. Dr. Ronaldo F. Hashimoto, who were involved in the examination committee, for their comments during the thesis defense.

I would like to thank as well my friends from the lab, specially to Leandro and Jihan. Also, a special thank to Professor Dr. Roberto Hirata Jr., who always supported me and helped me in different academic and personal circumstances.

I would like to thank CNPq [308985/2015-0, 486083/2013-6, 486988/2013-9, FINEP 1266/13], FAPESP [2011/50761-2, 2016/21591-5], CAPES, 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].

Abstract

LEON, LEISSI M. C. Efficient Hierarchical Layered Graph Approach for Multi-Region Segmentation. 78 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

Image segmentation refers to the process of partitioning an image into meaningful regions of interest (objects) by assigning distinct labels to their composing pixels. Images are usually composed of multiple objects with distinctive features, thus requiring distinct highlevel priors for their appropriate modeling. In order to obtain a good segmentation result, the segmentation method must attend all the individual priors of each object, as well as capture their inclusion/exclusion relations. However, many existing classical approaches do not include any form of structural information together with different high-level priors for each object into a single energy optimization. Consequently, they may be inappropriate in this context. We propose a novel efficient seed-based method for the multiple 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, being each object represented by a node. Each tree node may contain different individual high-level priors 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 and its extensions, on medical, natural and synthetic images, indicate promising results comparable to the state-of-the-art methods, but with lower computational complexity. Compared to 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: Multiple object segmentation; hierarchical image segmentation; image segmentation based on graphs; interactive segmentation; superpixels; medical image segmentation.

Resumo

LEON, LEISSI M. C. Abordagem Eficiente baseada em Grafo Hierárquico em Camadas para a Segmentação de Múltiplas Regiões. 78 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

A segmentação de imagem refere-se ao processo de particionar uma imagem em regiões significativas de interesse (objetos), atribuindo rótulos distintos aos seus pixels de composição. As imagens geralmente são compostas de vários objetos com características distintas, exigindo, assim, restrições de alto nível distintas para a sua modelagem apropriada. Para obter um bom resultado de segmentação, o método de segmentação deve atender a todas as restrições individuais de cada objeto, bem como capturar suas relações de inclusão / exclusão. No entanto, muitas abordagens clássicas existentes não incluem nenhuma forma de informação estrutural, juntamente com diferentes restrições de alto nível para cada objeto em uma única otimização de energia. Consequentemente, elas podem ser inapropriadas nesse contexto. Estamos propondo um novo método eficiente baseado em sementes para a segmentação de múltiplos objetos em imagens baseado em grafos, chamado *Hierarchical* Layered Oriented Image Foresting Transform (HLOIFT). Ele usa uma árvore das relações entre os objetos de imagem, sendo cada objeto representado por um nó. Cada nó da árvore pode conter diferentes restrições individuais de alto nível, que são usadas para definir um dígrafo ponderado, nomeado como *camada*. Os grafos das camadas são então integrados em um grafo hierárquico, considerando as relações hierárquicas de inclusão e exclusão. Uma otimização de energia única é realizada no dígrafo hierárquico em camadas, levando a resultados globalmente ótimos, satisfazendo todas as restrições de alto nível. As avaliações experimentais do HLOIFT e de suas extensões, em imagens médicas, naturais e sintéticas, indicam resultados promissores comparáveis aos métodos do estado-da-arte, mas com menor complexidade computacional. Comparada à segmentação hierárquica pelo algoritmo mincut/max-flow, nossa abordagem é menos restritiva, levando a resultados globalmente ótimos em cenários mais gerais e com melhor tempo de execução.

Palavras-chave: Segmentação de múltiplos objetos; segmentação hierárquica de imagens; segmentação de imagens baseada em grafos; segmentação interativa; superpixels; segmentação de imagens médicas.

Contents

A	cknov	vledgements	i	
Li	List of Acronyms ix			
Li	List of Symbols xi			
Li	st of	Figures xi	iii	
Li	st of	Tables xv	'ii	
1	Intr	oduction	1	
	1.1	Interactive graph-based segmentation	2	
	1.2	Multi-object segmentation	5	
	1.3	Proposal	6	
		1.3.1 Why optimization by OIFT?	7	
		1.3.2 Contributions	7	
	1.4	Outline	7	
2 Background		kground	9	
	2.1	Definitions	9	
		2.1.1 Image	9	
		2.1.2 Image as digraph \ldots	9	
		2.1.3 Symmetric digraph	9	
		2.1.4 Image segmentation $\ldots \ldots \ldots$	10	
	2.2	Related methods	10	
		2.2.1 Image Foresting Transform (IFT)	10	
		2.2.2 Oriented Image Foresting Transform (OIFT)	1	
	2.3	High-level priors	13	
		2.3.1 Boundary polarity	13	

		2.3.2	Geodesic star convexity	14	
		2.3.3	Geometric interactions	15	
3	Pro	posed	method	17	
	3.1	Notati	ons and definitions	19	
	3.2	Hierar	chical Layered OIFT (HLOIFT)	19	
		3.2.1	Construction of the layers of our digraph	20	
		3.2.2	Hierarchical digraph construction (setup of inter-layer connections) .	21	
		3.2.3	Energy optimization	22	
	3.3	Extens	sions of HLOIFT	25	
		3.3.1	3D HLOIFT	25	
		3.3.2	HLOIFT with superpixels	25	
4	Experimental results 2				
	4.1	2D exp	perimental evaluation	29	
		4.1.1	Setting the high-level priors	29	
		4.1.2	Qualitative comparison with the IFT method	31	
		4.1.3	Comparison with multi-object segmentation by min-cut/max-flow al-		
			gorithm	34	
		4.1.4	Quantitative accuracy experiments	36	
	4.2	3D exp	perimental analysis	39	
	4.3	Experi	imental evaluation with superpixels	41	
5	Conclusions 4				
	5.1	Final of	considerations	43	
	5.2	Public	ations	43	
	5.3	Future	works	44	
A	App	oendix		47	
	A.1	Proof	of Theorem 1	47	
Bi	bliog	raphy		51	
In	dex			58	

List of Acronyms

IFT	Image Foresting Transform
MRI	Magnetic Resonance Image
CT	Computed Tomography
OIFT	Oriented Image Foresting Transform
ORFC	Oriented Relative Fuzzy Connectedness
SSP	Star Shape Prior
GSC	Geodesic Star Convexity Constraints
HLOIFT	Hierarchical Layered Oriented Image Foresting Transform
GC	Graph Cut
GGC	Generalized Graph Cut
FOMs	Fuzzy Object Models
CNN	Convolutional Neural Networks

List of Symbols

- \mathcal{I} Image domain
- I Digital image
- I(t) Image intensity
- \mathbb{Z} Integer set
- \mathbb{R} Real set
- G Graph
- \mathcal{N} Graph nodes
- \mathcal{A} Graph adjacency
- ω Arc weights in the graph
- (s,t) An arc from s to t
- i, j, k Indexes
- O_i The *i*th object
- O_0 The background region
- *m* Number of regions to be segmented
- \mathcal{S} Seeds set
- \emptyset Empty set
- L Label map
- π Path in a graph
- Π Set of paths in a graph
- π_t A path with terminus at pixel t
- *P* Predecessor map or spanning forest
- f Connectivity function
- Q Priority queue
- V Cost map
- α Orientation value
- \mathcal{L} Set of label indexes
- h Tree of hierarchies
- ρ A minimal distance between two region contours

- \mathcal{H}_i The *i*th layer (graph)
- ${\cal H}$ Hierarchical graph of layers
- h(i) Indicates the parent index of the *i*th layer.
- \mathcal{A}_s Interlayer arcs for siblings layers
- \mathcal{A}_p Interlayer arcs for parent/offspring layers
- X Binary map
- ε Energy function
- G(.) Magnitude of the Sobel Gradient.

List of Figures

1.1	Examples of image segmentation. (a) A binary segmentation of a horse from	
	its background and (b) a multi-object segmentation of birds from their back-	
	ground.	1
1.2	Example of interactive image segmentation by IFT. (a) An input CT image of	
	the liver with the initial segmentation result from the user selected seeds (pur-	
	ple stroke for background and yellow stroke for the object). (b) An improved	
	segmentation result for the liver is obtained by adding more seeds	3
1.3	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	
	hight-level priors. (a,c) The results by Deep Extreme Cut with 4 clicks. (b,d)	
	The results by HLOIFT with arc-weight estimation from (de Miranda <i>et al.</i> ,	
	2010) 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	4
1.4	Example of multi-object segmentation by IFT. (a) A given input CT image	
	with its initial segmentation result from the user selected seeds, where purple	
	stroke stands for background, while yellow, cyan and red colors stand for the	
	liver, a orta and the abdominal region respectively. (b) By adding more seeds	
	an improved result is obtained, but it still has many imperfections	6
2.1	Example of segmentation by OIFT. (a) A given input image with the user	
	selected seeds, where the red color stands for background, while yellow stands	
	for the object (central bone). (b) A segmentation result by OIFT favoring	
	transitions from bright to dark pixels; and (c) a segmentation by OIFT favor-	
	ing transitions from dark to bright pixels	14
2.2	Star Convexity: (a) A star-convex object. (b) Violation of the star-convexity	
	prior, where p is a point of the given object and c is the star center considered.	14

2.3	(a) Input image with seeds selected by the user. (b) Segmentation result by OIFT without the Geodesic Star Convexity. (c) Result by OIFT with Geodesic	
2.4	Star Convexity, favoring a segmentation with a more regular shape Example of segmentation using geometric interactions. Given an input image (a) involving three objects: liver (O_1) , aorta (O_2) and abdominal region (O_3) with selected seeds in yellow, blue and red colors respectively. We have O_1 excluded from O_2 , and both are included in O_3 . Then, the segmentation result by our proposed method is shown in (b)	15 16
3.1	Overview of our framework. Given the input parameters, a hierarchical weighted digraph of layers (digraphs) is constructed using the <i>inclusion</i> (solid line) and <i>exclusion</i> (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 13. Notice, that only three souds are used.)	18
3.2	as output. (Compare to Figure 4.13. Notice, that only three seeds are used.) . 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 $\lambda(t) = j$	21
3.3	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) .	22
3.4	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. $h(i) = h(j) = k$.	22
4.1	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	
4.2	prior	30
4.3	ing objects	30
4.4	HLOIFT with boundary polarity from bright to dark pixels, requiring fewer seeds	32
	$(\alpha = -0.75)$	32

4.5	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 ($\alpha = 0.75$)	33
4.6	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.	34
4.7	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. $\ . \ .$	35
4.8	(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 hi-	
	erarchical layered graph cut (HLGC) mistakenly assigns the aorta as part of	
	the liver. (d) A better result is obtained by HLOIFT	37
4.9	The mean curves of Dice accuracy for different methods	38
4.10	(a) Given the input image with six regions and seeds for O_1 (in yellow), O_2 (in	
	blue) and background (in red), we obtain the result for O_1 and O_2 included in	
	O_3 , where O_1 has boundary polarity from bright to dark and O_2 from dark to	
	bright. In (b) a slice view is shown and in (c) the 3D renderings of the objects.	39
4.11	(a) Given the input image with six regions and seeds for O_1 (in yellow), O_2 (in	
	blue) and background (in red), we obtain the result for O_1 and O_2 included	
	in O_3 , where O_1 and O_2 have boundary polarity from dark to bright, and O_3	
	has the opposite orientation from bright to dark. In (b) a slice view is shown	
	and in (c) the 3D renderings of the objects	40
4.12	Example of 3D HLOIFT segmentation in a real MR image. (a) The brain. (b)	
	The lateral ventricles	40
4.13	The segmentation of a CT image of the knee for different superpixel sizes	42
5.1	Example of a thoracic-abdominal image from 3D-IRCADb-02 dataset (Soler <i>et al</i>	•,
	2012)	44
5.2	Example of cells with overlapping regions. In the left we have the input image,	
	and in the right a desired segmentation.	45

List of Tables

4.1	The running times for the flower segmentation by HLOIFT and the min-	
	cut/max-flow algorithm in layered graphs using different image sizes. $\ . \ . \ .$	35
4.2	Time in ms for the different methods and image resolutions	41

Introduction

Image segmentation is the task of partitioning an input image into regions of interest by assigning distinct labels to their composing pixels. Hence, all of the pixels sharing a same label in the image form an object, see Figure 1.1. The image segmentation involving the partitioning of image into a foreground and a background (representing every other part of the image that is not the foreground) is known as binary segmentation, see Figure 1.1(a). It can only be used to segment one object at a time. However, in practice, we often need to segment multiple objects from a single image simultaneously, where each label corresponds to a different object in the image, as shown in Figure 1.1(b).



Figure 1.1: Examples of image segmentation. (a) A binary segmentation of a horse from its background and (b) a multi-object segmentation of birds from their background.

The image segmentation task is not only one of the most fundamental and challenging problems in image processing and computer vision, but also has impact in different research areas such as Ecology, Medicine, Neurology and Artificial Intelligence (Gordon *et al.*, 2016; Toennies, 2012; Visser *et al.*, 2016). For example, in medical imaging, image segmentation can be used to isolate the objects corresponding to different organs in Magnetic Resonance Images (MRI), helping to analyze their forms, volumes and textures for the diagnostic of pathologies (Jackowski e Goshtasby, 2005). However, to achieve an ideal segmentation in medical imaging, we usually have critical obstacles to face such as noise, inhomogeneity effects, low contrast and complexity of objects. The incorporation of prior knowledge into the segmentation algorithm is useful for more accurate results.

The segmentation task can fall into one of three categories: manual segmentation process, automatic segmentation and interactive segmentation. The manual segmentation process can be a tedious work for humans, being time-consuming and prone to errors due to the large amount of data. The automatic segmentation algorithms usually model the objects of interest, through a training process on a large amount of manually labeled data. On the other hand, an interactive segmentation method combines computers and humans' expertise. Seedbased methods can be employed for interactive segmentation, where a set of "seed" pixels (or scribbles), specifying the regions of interest and their respective labels, is indicated by the user. This partial labeling is then propagated to the other pixels, allowing the user to evaluate the result and edit the final segmentation by adding more seeds if it is not satisfactory, until no further modification is needed.

Our work belongs to the seed-based class of segmentation methods, which requires as input some seeds/scribbles representing the localization of the regions of interest as constraints. We mainly focus on the interactive graph-based segmentation approach for multi-object segmentation. In the context of interactive segmentation, our method has the following pipeline: 1) A user/method provides an initial input; 2) the algorithm produces its resulting segmentation; 3) the user may provide further constraints based on the previous results, and go back to step 2. This process repeats until the results are satisfactory.

In this chapter, we briefly start presenting some state-of-the-art methods and discuss how the usage of prior information from the target objects may improve the multi-object segmentation results. Then we highlight our proposal and give the supporting reasons for choosing *Oriented Image Foresting Transform* (OIFT) (Miranda e Mansilla, 2014) for the energy optimization. Finally we provide a summary of our main contributions and briefly outline the organization of this thesis.

1.1 Interactive graph-based segmentation

Currently, interactive graph-based methods are commonly used in image segmentation tasks, where the image is modeled as a connected graph. Graph-based image segmentation is popular, because graphs can naturally represent image parts and their relationships (Golodetz *et al.*, 2017). Here, the image segmentation task can be interpreted as a graph partition problem subject to hard constraints, such as seed pixels selected in the image domain for the foreground regions and background (anywhere outside the foreground). Examples of interactive graph-based methods are watershed from markers (Cousty *et al.*, 2010), random walks (Grady, 2006), fuzzy connectedness (Ciesielski *et al.*, 2007), graph cuts (GC) (Boykov e Funka-Lea, 2006), grow cut (Li *et al.*, 2012), minimum barrier distance (Ciesielski *et al.*, 2014) and image foresting transform (IFT) (Ciesielski *et al.*, 2018; Falcão *et al.*, 2004). Figure 1.2 shows an example of interactive image segmentation by IFT (Ciesielski *et al.*, 2018; Falcão *et al.*, 2004) for the single object segmentation of the liver. (a) The user provides seeds as hard constraints representing foreground and background; and then an initial graph partition is obtained by energy optimization. (b) The initial segmentation can be further edited by adding new foreground and/or background seeds in order to improve the results.



Figure 1.2: Example of interactive image segmentation by IFT. (a) An input CT image of the liver with the initial segmentation result from the user selected seeds (purple stroke for background and yellow stroke for the object). (b) An improved segmentation result for the liver is obtained by adding more seeds.

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, which we refer to as, Generalized GC (GGC) (Ciesielski *et al.*, 2012). Oriented Image Foresting Transform (OIFT) (Falcão *et al.*, 1998; Miranda e Mansilla, 2014) and Oriented Relative Fuzzy Connectedness (ORFC) (Bejar e Miranda, 2015) are GGC methods designed for directed weighted graphs, which have low computational complexity compared to the min-cut/maxflow algorithm (Boykov e Funka-Lea, 2006).

Image segmentation can also 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 (He *et al.*, 2016). In this context, a state-of-the-art method in grabcutstyle (Rother *et al.*, 2004) for interactive segmentation with minimal user involvement is Deep Extreme Cut (Maninis *et al.*, 2018). It considers extreme points (left-most, right-most, top and bottom pixels at the object boundary) as 4 clicks for each object, and the *Convolutional Neural Network* (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 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 1.3a and c). Moreover, user input in grabcut-style is usually not appropriate to perform further corrections, such that a different type of corrective action is usually employed in approaches like GrabCut (Rother *et al.*, 2004), 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 1.3: 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 hight-level priors. (a,c) The results by Deep Extreme Cut with 4 clicks. (b,d) The results by HLOIFT with arc-weight estimation from (de Miranda et al., 2010) 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 (Leon e de Miranda, 2019)), the learning of graph weights by means of machine learning techniques (de Miranda *et al.*, 2010) (see Figures 1.3b and d), or even the incorporation of learning by deep learning techniques (Wolf *et al.*, 2017; Wolf *et al.*, 2019). 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.

1.2 Multi-object segmentation

Most successful segmentation methods usually incorporate some image prior information. In the context of multiple object segmentation, each object may present its own distinctive features, requiring different high-level priors. Priors or global object properties refer to some prior knowledge used to guide the segmentation process. For example, priors such as connectedness (Mansilla *et al.*, 2016; Vicente *et al.*, 2008), shape constraints (Gulshan *et al.*, 2010; Isack *et al.*, 2016; Mansilla e Miranda, 2013b; Veksler, 2008), convexity prior (Gorelick *et al.*, 2017), and boundary polarity (Miranda e Mansilla, 2014; Singaraju *et al.*, 2008) are potentially useful high-level priors for object segmentation, allowing the customization of the segmentation to a given target object.

Also, it is advantageous for the segmentation method to explore the structural interaction/relations between the different objects in the image, whenever it is possible. 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 do not include any form of structural information or only include high-level priors for single object segmentation (Gulshan *et al.*, 2010; Mansilla *et al.*, 2016; Miranda e Mansilla, 2014; Vicente *et al.*, 2008). 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 (Delong e Boykov, 2009; Delong *et al.*, 2012; Ulén *et al.*, 2013). 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. For example, they cannot represent the inclusion of a pair of adjacent objects in a third object. Also they usually have a high computational cost.

The methods based in Layered Optimal Graph Image Segmentation for Multiple Objects and Surfaces (LOGISMOS) (Oguz e Sonka, 2014; Yin et al., 2010) require an approximated pre-segmentation whenever the objects present complex shapes. 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, the methods usually employ Fuzzy Object Models (FOMs) (Miranda *et al.*, 2009; Mohammadianrasanani, 2013; Rittner *et al.*, 2014; Sun *et al.*, 2014; Tong *et al.*, 2013; Udupa *et al.*, 2013, 2014, 2011, 2012). However, these approaches are based on separate IFT executions per object, that do not incorporate structural information and the high-level priors of all objects into a single energy optimization, limiting their potential.

In Figure 1.4, we have an example of multi-object segmentation by IFT in a *Computed Tomography* (CT) image composed of three objects representing the liver, aorta and the abdominal region, which are shown in yellow, cyan and red color, respectively. Initially, IFT gets unsatisfactory results (Figure 1.4a). By adding more seeds the segmentation result is improved, however, it still presents many errors (Figure 1.4b). Our hypothesis is that by the usage of high-level priors, such as the geodesic star convexity and the boundary polarity from bright to dark pixels for the three regions, superior segmentation results can be obtained. Unfortunately, the existing IFT based approaches for multi-object segmentation do not support the incorporation of these constraints related to global object properties.



Figure 1.4: Example of multi-object segmentation by IFT. (a) A given input CT image with its initial segmentation result from the user selected seeds, where purple stroke stands for background, while yellow, cyan and red colors stand for the liver, aorta and the abdominal region respectively. (b) By adding more seeds an improved result is obtained, but it still has many imperfections.

1.3 Proposal

In this work, we circumvent the aforementioned problems, by proposing a hierarchical layered graph-based approach for the multiple object segmentation problem, named as Hi-erarchical Layered OIFT (HLOIFT), using a single energy optimization. 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 Delong e Boykov (2009), 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.

1.3.1 Why optimization by OIFT?

The proposed method in this work uses a modified OIFT algorithm (Miranda e Mansilla, 2014) as its main optimization algorithm, as justified for the following reasons:

- The OIFT algorithm has a good balance between accuracy and robustness in practice (Tavares *et al.*, 2017).
- It achieves globally optimal results according to a graph-cut measure (Miranda e Mansilla, 2014).
- It has low computational cost (linear for most applications).
- It is flexible, allowing the incorporation of different high-level priors, such as shape constraints (de Moraes Braz *et al.*, 2019) and boundary polarity, into a single energy for object segmentation.

1.3.2 Contributions

Therefore, 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 globally optimal results in more general scenarios compared to Delong e Boykov (2009).
- **Complexity:** Our method has lower computational complexity as compared to methods based on the min-cut/max-flow algorithm.

1.4 Outline

Our work is organized as follows: In Chapter 2, we introduce some required definitions and review the OIFT method with some commonly used priors. Our new algorithm HLOIFT, is described in Chapter 3 and the proof of its correctness is given in Appendix A. In Chapter 4, we present the experimental results, including the 3D case and the usage of superpixels for the nodes. The usage of superpixels allows us to reduce the running time of HLOIFT while conserving its accuracy. There is also some quantitative experimental evaluation of HLOIFT, comparing it against IFT and hierarchical segmentation by the min-cut/max-flow algorithm. Finally, the remaining conclusions are stated in Chapter 5.

Background

This chapter provides the theoretical foundations of our proposed method. Here, we describe the image and digraph definitions, the related methods and algorithms, and the high-level priors to be used in HLOIFT. The details provided here are necessary for fully understanding our proposal and the results presented in subsequent chapters.

2.1 Definitions

2.1.1 Image

An (2-dimensional) image is a pair (\mathcal{I}, I) , where \mathcal{I} is a finite set of *pixels* identified with the 2-tuples of integers (i.e., $\mathcal{I} \subset \mathbb{Z}^2$) and I is a mapping that assigns to each pixel $t \in \mathcal{I}$ its intensity I(t), that is, either a real number (in case of monochromatic image) or an appropriate vector in \mathbb{R}^k .

2.1.2 Image as digraph

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 $\mathcal{N} = \mathcal{I}$, arcs/edges listed in \mathcal{A} are the ordered pairs $(s,t) \in \mathcal{I}^2$ 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.

2.1.3 Symmetric digraph

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 in this work are symmetric. Notice, that in the symmetric weighted digraphs we may still have $\omega(s,t) \neq \omega(t,s)$.

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

2.1.4 Image segmentation

In binary segmentation, whose goal is to separate an object O_1 from its background $O_0 = \mathcal{I} \setminus O_1$, we consider two non-empty disjoint seed sets $\mathcal{S}_0, \mathcal{S}_1 \subset \mathcal{I}$ 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* $L: \mathcal{I} \to \{0, 1\}$, which is the indicator (i.e., characteristic) function of O_1 . (That is, $O_1 = \{v \in \mathcal{I}: L(v) = 1\}$). 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 algorithm. All this notation may be easily extended for multiple object segmentation.

Image segmentation can be formulated as a graph cut (GC) problem subject to hard constraints. 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 (Ciesielski *et al.*, 2012). The algorithms presented in this work are Max-Min optimizers while the classical min-cut/max-flow algorithm is a Min-Sum optimizer.

2.2 Related methods

2.2.1 Image Foresting Transform (IFT)

The Image Foresting Transform (IFT) (Falcão *et al.*, 2004) is a modification of Dijkstra's shortest paths algorithm (Dijkstra, 1959) allowing multiple sources and general connectivity functions (Monotonically Increasing (MI) functions). IFT uses definitions such as *path*, connectivity function, spanning forest and optimum path value. For a given image graph G, a path is a sequence of pixels $\pi = \langle t_1, t_2, \ldots, t_k \rangle$, where $(t_i, t_{i+1}) \in \mathcal{A}$, for $1 \leq i \leq k - 1$. We use π_t to indicate that pixel t is the terminus of the path. To explicitly indicate the origin of the path, we use the notation $\pi_{s \to t} = \langle s = t_1, t_2, \ldots, t_k = t \rangle$, where s indicates the origin and t the terminus. A path is trivial when k = 1. If π_s and $\tau = \langle s, t \rangle$ are both paths, we denote as $\pi_s \cdot \tau$ the concatenation of the two paths. We denote as $\Pi(G, t)$ the set of all paths in the graph G with terminus t, while $\Pi(G) = \bigcup_{t \in \mathcal{I}} \Pi(G, t)$ denotes all possible paths in G.

A spanning forest is a function P that assigns to each pixel t in \mathcal{I} either some other adjacent pixel in \mathcal{I} , or a distinctive marker nil not in \mathcal{I} , with no containing cycles. Thus, for any pixel $t \in \mathcal{I}$, a spanning forest P defines a path π_t^P recursively as $\langle t \rangle$ if P(t) = nil (root node), and $\pi_s^P \cdot \langle s, t \rangle$ if $P(t) = s \neq nil$ (predecessor node of t).

A connectivity function is a function $f : \Pi(G) \to \mathbb{R}$, that assigns to each path π a path cost value $f(\pi)$. Usually, the path cost depends on the arc weights ω along the path. In this work, we use connectivity functions (f) constrained to paths starting in a given set of *seed* pixels $S \subseteq \mathcal{I}$. This constraint is modeled by defining a new path-cost function $f^S(\pi)$, which is equal to $f(\pi)$ when the origin of π belongs to S, and it is equal to $+\infty$ otherwise.

 $^{^{2}}$ Min-Max optimizer is a dual equivalent problem.

A path π_t is optimum if $f(\pi_t) \leq f(\tau_t)$ for any other path $\tau_t \in \Pi(G, t)$. The optimum path value $V_{opt}(t)$ is obtained by taking to each pixel t the cost of one optimum path with terminus at t, which is defined by $V_{opt}(t) = \min_{\forall \pi_t \in \Pi(G,t)} \{f(\pi_t)\}$. The IFT algorithm (Algorithm 1) solves this path cost optimization problem. It receives an image graph G, a MI connectivity function f and a set of seeds S, and it computes a label map L, a spanning forest $P : \mathcal{I} \to \mathcal{I} \cup \{nil\}$ of optimum paths π_t^P associated to each pixel $t \in \mathcal{I}$, and a connectivity map V, such that $V(t) = f(\pi_t^P)$. For the sake of simplicity, Algorithm 1 is presented here for binary segmentation.

Algorithm 1. – GENERAL IFT ALGORITHM

INPUT:	mage graph $G = (\mathcal{N}, \mathcal{A}, \omega)$, non-empty disjoint seed sets $\mathcal{S}_0, \mathcal{S}_1 \subset \mathcal{N}$, and the
	path-cost function $f: \Pi(G) \to \mathbb{R}$.

- OUTPUT: Optimum-path forest $P : \mathcal{N} \to \mathcal{N} \cup \{nil\}$, the path-cost map $V : \mathcal{N} \to [-\infty, +\infty]$ and label map $L : \mathcal{N} \to \{0, 1\}$.
- AUXILIARY: Priority queue Q, variable tmp, and a status function $S: \mathcal{N} \to \{0, 1\}$, where S(t) = 1 for processed nodes and S(t) = 0 for unprocessed nodes.

```
1. For each t \in \mathcal{N}, do
```

2.		Set $S(t) \leftarrow 0$, $P(t) \leftarrow nil and V(t) \leftarrow f(\langle t \rangle)$;
3.		If $t \in \mathcal{S}_0$, then
4.		$ L(t) \leftarrow 0, and insert t in Q; $
5.		If $t \in \mathcal{S}_1$ then
6.		$L(t) \leftarrow 1$, and insert t in Q.
7.	Whi	ile $Q eq \emptyset$ 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.		Compute $tmp \leftarrow f(\pi_s^P.\langle s, t \rangle)$
12.		If $tmp < V(t)$ then
13.		Set $P(t) \leftarrow s, V(t) \leftarrow tmp and L(t) \leftarrow L(s)$
14.		

In the case of non-MI functions, the IFT algorithm produces a spanning forest P, but the paths π_t^P may not be optimum (i.e., we may have $f(\pi_t^P) = V(t) \neq V_{opt}(t)$). However, the computed spanning forest P may be optimal according to other optimality criteria, as proved for the cost function of Oriented Image Foresting Transform (OIFT) (Miranda e Mansilla, 2014), as presented next.

2.2.2 Oriented Image Foresting Transform (OIFT)

The Oriented Image Foresting Transform (OIFT) method is build upon the IFT framework by exploring the object-contour orientation in digraphs, using non-MI connectivity functions (Miranda e Mansilla, 2014). It improves the segmentation results because it helps to distinguish between two similar and nearby boundary segments with opposite orientations from distinct objects.

The OIFT algorithm is a Max-Min optimizer, that is, its resulting segmentation gives a global optimum solution subject to the seed constraints by maximizing the graph-cut measure ε_{min} defined as:

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

$$(2.1)$$

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 2, in a connected and symmetric digraph G, as described in Mansilla e Miranda (2013a). Alternatively, its segmentation can also be obtained by Algorithm 1, using the following non-MI connectivity function f^* :

$$f^{\star}(\langle t \rangle) = \begin{cases} -\infty & \text{if } t \in \mathcal{S}_0 \cup \mathcal{S}_1 \\ +\infty & \text{otherwise} \end{cases}$$
$$f^{\star}(\pi_{r \rightsquigarrow s} \cdot \langle s, t \rangle) = \begin{cases} \omega(s, t) & \text{if } r \in \mathcal{S}_1 \\ \omega(t, s) & \text{otherwise} \end{cases}$$
(2.2)

Algorithm 2. – OIFT ALGORITHM

J

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

```
1. For each t \in \mathcal{N}, do
```

```
Set S(t) \leftarrow 0 and V(t) \leftarrow \infty;
2.
               If t \in S_0, then
3.
                  \downarrow V(t) \leftarrow -\infty, L(t) \leftarrow 0, and insert t in Q;
4.
               If t \in S_1 then
5.
                  \ \ V(t) \leftarrow -\infty, \ L(t) \leftarrow 1, \ and \ insert \ t \ in \ Q. 
6.
7.
      While Q \neq \emptyset do
8.
               Remove s from Q such that V(s) is minimum;
9.
               Set S(s) \leftarrow 1;
               For each (s,t) \in \mathcal{A} such that S(t) = 0 do
10.
11.
                       If L(s) = 1 then tmp \leftarrow \omega(s, t)
12.
                       Else tmp \leftarrow \omega(t,s);
13.
                       If tmp < V(t) then
                               Set V(t) \leftarrow tmp \text{ and } L(t) \leftarrow L(s)
14.
                               If t \notin Q then insert t in Q.
15.
```

Note that in line 12 of Algorithm 2, 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 (2.1), 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 2 is equivalent to the execution of a single line: "tmp $\leftarrow w_{L(s)}(s,t)$ ". Also, if consecutive line 14 is executed, then V(t) becomes $w_{L(s)}(s,t)$. In this case, the path-cost function from Equation 2.2 can also be expressed as $f^*(\langle v_0, \ldots, v_\ell \rangle) = w_{L(v_0)}(v_{\ell-1}, v_\ell)$ for $\ell > 0$.

In Section 2.3.1, we explain how the boundary polarity prior of OIFT is treated.

2.3 High-level priors

The usage of some prior knowledge of the objects is usually necessary in order to obtain successful segmentation results. In this section, we briefly describe some individual and structural priors used, such as boundary polarity, shape priors and geometric interactions.

2.3.1 Boundary polarity

To explore the boundary polarity, the arc weights $\omega(s, t)$ are defined as a combination of an undirected dissimilarity measure $\psi(s, t)$ between neighboring pixel s and t, multiplied by an orientation factor, as follows:

$$\omega(s,t) = \begin{cases} \psi(s,t) \times (1+\alpha) & \text{if } I(s) > I(t), \\ \psi(s,t) \times (1-\alpha) & \text{if } I(s) < I(t), \\ \psi(s,t) & \text{otherwise,} \end{cases}$$
(2.3)

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 (Ciesielski e Udupa, 2010; de Miranda *et al.*, 2010). Note that, in general, we have $\omega(s, t) \neq \omega(t, s)$ for $\alpha \neq 0$. As shown in Figure 2.1(b), for $\alpha > 0$, the segmentation by OIFT favors transitions from bright to dark pixels, and $\alpha < 0$ favors the opposite orientation, as shown in Figure 2.1(c).

Finally, for multi-object segmentation we consider multiples α_i values, each associated to a different object.



Figure 2.1: Example of segmentation by OIFT. (a) A given input image with the user selected seeds, where the red color stands for background, while yellow stands for the object (central bone). (b) A segmentation result by OIFT favoring transitions from bright to dark pixels; and (c) a segmentation by OIFT favoring transitions from dark to bright pixels.

2.3.2 Geodesic star convexity

Shape priors have been widely utilized in medical image segmentation to improve segmentation accuracy and robustness (Bai *et al.*, 2017). A common shape prior is the **star shape prior** (SSP) (Veksler, 2008). An object has a star shape with respect to a given center c if for any point p inside the object, all points on the straight line segment between the center c and p also lie inside it (Figure 2.2).



Figure 2.2: Star Convexity: (a) A star-convex object. (b) Violation of the star-convexity prior, where p is a point of the given object and c is the star center considered.

The geodesic star convexity prior (GSC) corresponds to a discrete version of SSP defined 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. It considers all the given object seeds as center points, prioritizing the segmentation of an object with more regular shape (Mansilla *et al.*, 2013). OIFT subject to the geodesic star convexity prior can be obtained by setting the weights of some arcs in the created digraph to $-\infty$, according to the scheme proposed in Mansilla e Miranda (2013b), prior to computing the OIFT. For the natural image presented in Figure 2.3a, Figure 2.3b shows a segmentation result without any shape priors, while the corresponding segmentation with the GSC prior is shown in Figure 2.3c.



Figure 2.3: (a) Input image with seeds selected by the user. (b) Segmentation result by OIFT without the Geodesic Star Convexity. (c) Result by OIFT with Geodesic Star Convexity, favoring a segmentation with a more regular shape.

It is also possible to simultaneously handle boundary polarity and shape priors in OIFT, according to the results as presented in Mansilla e Miranda (2013b).

2.3.3 Geometric interactions

The contribution of Delong e Boykov (2009) for multi-object segmentation is a binary multi-layered formulation, defining a layer for each object, encoding the geometric interactions between the different objects in the image. These interactions refer to the *inclusion* of one object within another object or the *exclusion* between them, while enforcing a minimal distance between the object contours in different layers.

However, as a Min-Sum optimizer of data terms is used in Delong e Boykov (2009), their globally optimal results are restricted only to some particular cases, because it cannot always be converted to a submodular energy. For example, Figure 2.4 illustrates an example using geometric interactions between three objects: liver (O_1) , aorta (O_2) and abdominal region (O_3) , with yellow, blue and red colors respectively, where O_1 is excluded from O_2 , and both are included in O_3 . Global optimum results of the Min-Sum optimizer by Delong e Boykov (2009) cannot be obtained in this case, while our proposed approach solves this problem for the Max-Min optimizer with global optimum results (Figure 2.4b).



(a) Input image



(b) Segmentation by our proposed method

Figure 2.4: Example of segmentation using geometric interactions. Given an input image (a) involving three objects: liver (O_1) , aorta (O_2) and abdominal region (O_3) with selected seeds in yellow, blue and red colors respectively. We have O_1 excluded from O_2 , and both are included in O_3 . Then, the segmentation result by our proposed method is shown in (b).
Proposed method

We propose the *Hierarchical Layered Oriented Image Foresting Transform* (HLOIFT) as a new seed-based method for multi-object segmentation (Leon e Miranda, 2017). Figure 3.1 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 3.2.1). (2) HLOIFT defines a setup for the interlayer connections representing the hierarchical constraints, such as inclusion and exclusion relations (described in Section 3.2.2). (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 3.2.3).

We first present some required notations and definitions.



Figure 3.1: 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 4.13. Notice, that only three seeds are used.)

3.1 Notations and definitions

Let $\mathcal{L} = \{1, \ldots, m\}$ 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{I}, I) will be defined on the set of nodes $\mathcal{N} = \mathcal{L} \times \mathcal{I}$. The HLOIFT resulted segmentation of the image will be identified with a binary variable $X : \mathcal{N} \to \{0, 1\}$, where, for $i \in \mathcal{L}$ the *i*th object O_i and the background O_0 are defined, respectively, as

$$O_i = \{ t \in \mathcal{I} : X(i, t) = 1 \} \quad \text{and} \quad O_0 = \mathcal{I} \setminus \bigcup_{i \in \mathcal{L}} O_i.$$
(3.1)

Each object/background O_i , $i \in \mathcal{L} \cup \{0\}$, will be identified with a corresponding set $S_i \subset \mathcal{I}$ of seeds, aiming for $S_i \subseteq O_i$. Only the seed sets 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.

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 $h: \mathcal{L} \to \{1, \ldots, m+1\}$, referred to as a *tree*, and defined as follows. If $O_{m+1} = \mathcal{I}$ (the image domain and the root of the tree), then h(i) = jif, 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 3.1 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.

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,

(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 $||s - t|| \le \rho$.

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 Hierarchical Layered OIFT (HLOIFT)

We present next the steps of our proposed method.

3.2.1 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 $\mathcal{N}_i = \{i\} \times \mathcal{I}$ and each node $t = (i, v) \in \mathcal{N}_i = \{i\} \times \mathcal{I}$ correspond to the image pixel p(t) = v. Thus, the node set \mathcal{N} of HLOIFT digraph is defined as $\mathcal{L} \times \mathcal{I} = \bigcup_{i \in \mathcal{L}} \mathcal{N}_i$ and $p: \mathcal{N} \to \mathcal{I}$ is the projection onto the second coordinate. See graphical representation of layers in Figures 3.2, 3.3, and 3.4. Also, $\lambda: \mathcal{L} \times \mathcal{I} \to \mathcal{L}$ will denote the projection onto the first coordinate, that is, $\lambda(t) = i$ means that t belongs to the ith layer of the graph.

Let $\mathcal{A}_{\mathcal{I}}$ be an affinity defined on the original image graph (with $\mathcal{N} = \mathcal{I}$) as in the case of OIFT. We will usually assume that, in the case of 2D images, $\mathcal{A}_{\mathcal{I}}$ is the 4- or 8-neighborhood adjacency. We define the intra-layer adjacency \mathcal{A}_i on \mathcal{N}_i as this $\mathcal{A}_{\mathcal{I}}$, that is, $(s,t) \in \mathcal{A}_i$ if, and only if, $(p(s), p(t)) \in \mathcal{A}_{\mathcal{I}}$. 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 (2.3) with $\psi(s,t) = ||I(s) - I(t)||$. (For other possible definitions of ψ see, e.g., (Ciesielski e Udupa, 2010; de Miranda *et al.*, 2010).) In this setting, each object O_i has its own constant $\alpha_i \in [-1, 1]$ (used in (2.3)), 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) (Veksler, 2008), 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. 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 functions, as the line segments. It considers all the given seeds as center points of a region, prioritizing the segmentation of the 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 Mansilla *et al.* (2013). Moreover, it is still possible to simultaneously handle boundary polarity and shape priors (Mansilla e Miranda, 2013b).

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

In this step, HLOIFT generates a *hierarchical layered* weighted digraph $\mathcal{H} = (\mathcal{N}, \mathcal{A}, \omega)$ as the union of all layered graphs \mathcal{H}_i , $i = 1, \ldots, m$, with additional *interlayer* 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 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.

Inclusion. If O_j is the parent of O_i (i.e., h(i) = j), then we define ω(t, s) = ∞ and ω(s,t) = -∞. Figure 3.2 shows, in this case, the arcs and their weights, where the inter-layer arcs are formed according to a 4-neighborhood adjacency and the parameter ρ is equal 1.



Figure 3.2: Illustration of inter-layer arc construction, involving two objects O_i and O_j , for the inclusion case $(O_j \text{ is the parent of } O_i, \text{ i.e., } h(i) = j)$, 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 3.3.

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



Figure 3.3: Illustration of inter-layer arc construction, involving two objects O_i and O_j , for the exclusion case $(O_i \text{ and } O_j \text{ are siblings})$, where $\omega(s,t) = -\infty$ for all inter-layer arcs (s,t).

vertex connected to one just processed will be examined with the highest possible priority, to ensure satisfaction of the consistency requirement (C).

In Figure 3.4, we give an example of the hierarchical layered digraph construction for a case that cannot be optimized under the graph cut framework of Delong e Boykov (2009). 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 Delong e Boykov (2009). The proposed HLOIFT method can compute globally optimal results in this case and also in other more sophisticated cases, as described below.



Figure 3.4: 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.

3.2.3 Energy optimization

In the final step, we execute the HLOIFT algorithm (Algorithm 3), which constitutes a modified OIFT algorithm from (Mansilla e Miranda, 2013a; Miranda e Mansilla, 2014). It is applied to the hierarchical layered graph \mathcal{H} constructed above and its output *maximizes*

a single energy ε_{\min}^{h} (i.e., the solution X by HLOIFT is such that $X = \operatorname{argmax}_{X'} \varepsilon_{\min}^{h}(X')$, among all solutions X' satisfying the seed constraints), a modification of energy $\varepsilon_{\min}(L) = \min\{\omega(s,t): (s,t) \in \mathcal{A} \& L(s) > L(t) = 0\}$ defined in (2.1). This modification of ε_{\min} is designed to ensure that the output satisfies also the hierarchical constraints imposed by hand ρ .

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

$$\varepsilon_{\min}^{h}(X) = \min\{\varepsilon_{\min}^{incl}(X), \varepsilon_{\min}^{excl}(X)\}, \qquad (3.2)$$

where

$$\varepsilon_{\min}^{incl}(X) = \min\{\omega(s,t) \colon (s,t) \in \mathcal{A} \setminus \mathcal{A}_s \& X(s) > X(t)\},\$$
$$\varepsilon_{\min}^{excl}(X) = \min\{\omega(s,t) \colon (s,t) \in \mathcal{A}_s \& X(s) = X(t) = 1\}.$$

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 not 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 $\langle S_0, \ldots, S_m \rangle$ of seed sets is consistent with (C) (with respect to h and ρ) provided $S_i \neq \emptyset$ for i = 0 and every $i \in \mathcal{L}$ with no offsprings and there exists a sequence $\langle O_0, \ldots, O_m \rangle$ satisfying (C) such that $S_i \subseteq O_i \subset \mathcal{I}$ for every $i \in \mathcal{L} \cup \{0\}$.

Theorem 1 (Cut optimality by HLOIFT). For every image (\mathcal{I}, I) , a hierarchy tree h, a distance parameter $\rho \geq 0$, and a sequence $\langle S_0, \ldots, S_m \rangle$ of seed sets consistent with (C), the binary map $X : \mathcal{N} \to \{0, 1\}$ computed by Algorithm 3 maximizes the energy $\varepsilon_{\min}^h(X)$ given by (3.2) among all solutions satisfying the seed constraints and the consistency requirement (C).

Algorithm 3. – 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
	$\langle \mathcal{S}_0, \ldots, \mathcal{S}_m \rangle$ consistent with (C).

OUTPUT: The binary map $X: \mathcal{N} \to \{0, 1\}$ identifying segmentation given by (3.1).

AUXILIARY: Priority queue Q, variable tmp, the cost map $V \colon \mathcal{N} \to [-\infty, \infty]$, and an array of status $S \colon \mathcal{N} \to \{0, 1\}$, 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.		$\downarrow V(t) \leftarrow -\infty, X(t) \leftarrow 0, and insert t in Q;$	
5.		If $p(t) \in S_i$ and $\lambda(t) = i$ then	
6.		$ \downarrow 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 $tmp < V(t)$ and $\neg [(s,t) \in \mathcal{A}_s \text{ and } X(s) = 0]$, then	
14.		Set $V(t) \leftarrow tmp and X(t) \leftarrow X(s);$	
15.		If $t \notin Q$ then insert t in Q;	
16.		If $(s,t) \in \mathcal{A}_s$ and $X(s) = 1$ then	
17.			

Algorithm 3 is essentially identical to Algorithm 2 after we disregard lines 16-17 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 16 is never satisfied, so lines 16-17 can be removed. More generally, lines 16-17 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 Mansilla e Miranda (2013a).

In line 16 the condition is satisfied when s and t are siblings and $p(s) \in O_{\lambda(s)}$ as X(s) = 1.¹ Therefore, in line 17 we impose X(t) = 0, to ensure that $p(t) \notin O_{\lambda(t)}$, as required by (C). More specifically, lines 16-17 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 (Falcão *et al.*, 2004) and $\mathcal{O}(M+N\log N)$ (linearithmic

¹In fact, the check condition X(s) = 1 on line 16 of Algorithm 3 is not strictly necessary. We are keeping it only to facilitate the understanding.

time) if \mathcal{Q} is a heap. The Graph cut computational complexity is $\mathcal{O}(\sqrt{M} * N^2) = O(N^{2.5})$ 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 (Boykov e Kolmogorov, 2004).

As a final note, observe that an interesting variation of Algorithm 3 could be considered. Specifically, in the case of inclusion relation h(i) = j, we can additionally require that the seeds in S_j belong exclusively to the parent object (i.e., $O_i \cap S_j = \emptyset$). In other words, S_j should be inserted as background seeds on its children's layers to additionally impose that the seeds in S_j do not belong to the offspring objects of j.

3.3 Extensions of HLOIFT

3.3.1 3D HLOIFT

A 3D image is a pair (\mathcal{I}, I) , where \mathcal{I} is a finite set of voxels identified with the 3-tuples of integers (i.e., $\mathcal{I} \subset \mathbb{Z}^3$) and I is a mapping that assigns to each voxel $t \in \mathcal{I}$ its intensity I(t), that is, either a real number (in case of monochromatic image) or an appropriate vector in \mathbb{R}^k .

From the implementation point of view, the only required changes of HLOIFT for the segmentation of 3-dimensional images (volumes) are that now each layer has intra-layer arcs in a 3D neighborhood and that the parameter ρ defines a sphere in space.

3.3.2 HLOIFT with superpixels

Here, we propose a **superpixel**-based adaptation of the HLOIFT method, leading to a more efficient and adequate solution for multi-object segmentation in large images.

Superpixels can group pixels into perceptually meaningful atomic regions of similar and connected pixels. Superpixel primitives are computationally much more efficient than their pixel counterparts. For instance, superpixels can be used to create a *Region Adjacency Graph* (RAG) greatly reducing the number of graph elements.

The superpixels are given by an unsupervised segmentation from a given image. The most popular method for superpixel generation is SLIC, which starts with a selection of k initial cluster centers, which are sampled on a *regular grid* spaced $\sqrt{|\mathcal{I}|/k}$ nodes apart. Then, for the assignment step of each pixel to a cluster/superpixel, an adaptive k-means clustering is used for SLIC.

In this work we used IFT-SLIC (Alexandre *et al.*, 2015), which combines benefits from both IFT (*Image Foresting Transform*) and SLIC (*Simple Linear Iterative Clustering*) (Achanta *et al.*, 2012) to compute superpixels from a given input image.

IFT-SLIC converts the color space of the input image to CIELAB, in the case of colored images, and starts with a selection of k initial cluster centers similar to SLIC. Then, for the assignment step of each pixel to a cluster, instead of using an adaptive k-means clustering

such as SLIC, IFT-SLIC considers IFT computations with a non-MI connectivity function, featuring a parameter that controls the boundary adherence and compactness of the generated clusters. Then each cluster center is updated to be the closest pixel to its mean position and the process is repeated for a total of n iterations.

Similarly to HLOIFT at the pixel level described before, the HLOIFT at superpixel level is composed of three steps and produces a labeled image as result.

- Set of object-digraph construction We first create a set of m weighted digraphs of superpixels \mathcal{H}_i , $i = 1, \ldots, m$, where each digraph corresponds to a single object O_i of an (n-dimensional) image (\mathcal{I}, I) . Each graph $\mathcal{H}_i = (\mathcal{N}_i, \mathcal{A}_i, \omega_i)$ is a triple consisting of a vertex set \mathcal{N}_i , a directed edge set \mathcal{A}_i and a weight function ω_i . Each pair $(\mathcal{N}_i, \mathcal{A}_i)$ is an isomorphic copy of a Region Adjacency Graph (RAG) of the given image (\mathcal{I}, I) segmented in superpixels by IFT-SLIC (Alexandre *et al.*, 2015), while ω_i is a defined weight function for every $(s, t) \in \mathcal{A}_i$, being p(s) and p(t) the corresponding superpixels of nodes s and t. For example, we may consider $\omega_i(s, t) = |I(p(s)) I(p(t))|$, where I(p(t)) is the mean intensity inside superpixel p(t). Of course, ω_i should also highlight the priors for each object O_i whenever it is appropriate. For this purpose, we consider the same modification scheme of the weight assignment that was adopted by the regular OIFT method (Miranda e Mansilla, 2014) for *boundary polarity* priors, where the polarity of each O_i is defined to highlight boundary transitions from *bright to dark* superpixels or from *dark to bright*.
- **Hierarchical digraph construction** In this step, we generate a *hierarchical* weighted digraph $\mathcal{H} = (\mathcal{N}, \mathcal{A}, \omega)$ as the union of all object-digraphs of superpixels $\mathcal{H}_i, i = 1, \dots, m$, with additional arcs connecting only some of the distinct object-digraphs, based on the priors given by the parent tree h and the parameter $\rho \geq 0$ representing the minimal distance between the object boundaries. The hierarchical prior between any pair $\langle O_i, O_j \rangle$ of objects is understood as an exclusion case when $O_i \cap O_j = \emptyset$, or as an inclusion case when one of them is properly contained in the other. For convenience, object $O_{m+1} = \mathcal{I}$ is considered to be the root of the tree representing the image domain. The notation h(i) = j indicates that $O_i \subset O_j$, being O_j the *parent* of O_i and we say that two objects O_i and O_j are siblings if h(i) = h(j), meaning that both have the same parent. The weights of the arcs for the *inclusion case* are given by $\omega(s,t) = -\infty$ and $\omega(t,s) = \infty$ for h(i) = j, $s \in \mathcal{N}_i$ and $t \in \mathcal{N}_j$, whenever the superpixels p(s) and p(t) have pixels with a distance smaller than ρ . For the exclusion case we consider $||p-q|| > \rho$ for every pixels $p \in O_i$ and $q \in O_i$. Therefore, for the exclusion case, we have special arcs to avoid overlapping between sibling objects, defining $\omega(s,t) = \omega(t,s) = -\infty$, whenever superpixels p(s) and p(t) have pixels with a distance smaller than or equal to ρ . Under this scheme, it is possible to have many different and sophisticated cases of hierarchical constraints which cannot be easily modeled with graph cuts (Delong e Boykov, 2009).

Energy optimization Finally, we execute our proposed method using an algorithm similar

to Algorithm 3, as presented in Chapter 3, but running on the hierarchical weighted digraph of superpixels \mathcal{H} as constructed here. Its output maximizes a single energy defined to ensure that the output satisfies all the constraints, including the constraints imposed by h and ρ , according to the theoretical result presented in Leon *et al.* and Appendix A.

Experimental results

In this chapter, 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 (Miranda e Falcão, 2009) and the segmentation of multiple objects by the hierarchical min-cut/max-flow algorithm (Delong e Boykov, 2009)

We also evaluate the usage of HLOIFT in 3-dimensional images (volumes). The main challenges correspond to the required space and execution time, since in multidimensional data the memory consumption grows exponentially with the increase in the number of dimensions. Here we present the results of applying HLOIFT to 3D volumes, through the direct application of its proposed algorithm, without taking any additional precautions to get a more moderate usage of memory, in order to test its viability in this more challenging scenario.

Finally, we present experimental results for the **superpixel**-based adaptation of the HLOIFT method, leading to a more efficient and adequate solution for multi-object segmentation in large images, and show the improvement in the running time.

4.1 2D experimental evaluation

4.1.1 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 4.1 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 (2.3) 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. For example, Figure 4.1(a) shows the result when O_1 has boundary polarity setup from bright to dark and O_2 from dark to bright, while Figure 4.1(d) shows the result for the opposite orientation.



Figure 4.1: 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 4.2 shows HLOIFT with geodesic star convexity prior to segment an image of archaeological fragments (Figure 4.2a), with three sibling objects, using the simplest arc weights $\omega_i(s,t) = ||I(t)-I(s)||$ (Figure 4.2b). For $\rho = 0$, we get a bad result with touching objects (Figure 4.2c), which can be avoided by using $\rho = 2$ (Figure 4.2d).



Figure 4.2: 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.

4.1.2 Qualitative comparison with the IFT method

In this section, we compare the results obtained by HLOIFT against the IFT method (Miranda e Falcão, 2009) 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 (Mansilla e Miranda, 2013a). We used for this experiment medical images, composed of multiple sophisticated objects.

Figure 4.3 shows an image segmentation task, using ω_i defined via formula (2.3) with $\psi(s,t) = |I(s) - I(t)|$, where the goal is to segment a gray-scale CT image composed of three regions, 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 4.4 shows the segmentation of a MRI slice of a real foot, using ω_i defined via formula (2.3) with $\psi(s,t) = G(s) + G(t)$, where G(.) 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 siblings, 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 4.5 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 (2.3) with $\psi(s,t) = G(s) + G(t)$, where G(.) 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.



Figure 4.3: 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 4.4: Talus (O_1) and calcaneus (O_2) segmentation. The two objects are siblings. For *HLOIFT*, we used $\rho = 0$, the geodesic star convexity and boundary polarity ($\alpha = -0.75$).

In Figure 4.6, we use an axial cross section of a thoracic-abdominal CT image extracted from an open database called 3D-IRCADb-02 (Soler *et al.*, 2012) 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



Figure 4.5: 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 ($\alpha = 0.75$).

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 (2.3) 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 , 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.



Figure 4.6: 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.

4.1.3 Comparison with multi-object segmentation by min-cut/maxflow algorithm

In this section, we compare HLOIFT with multi-object segmentation by the min-cut/maxflow algorithm in layered graphs (Delong e Boykov, 2009). For the sake of simplicity, we considered only the inclusion relation, since many cases combining the inclusion and exclusion hierarchical constraints, such that 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 (Delong e Boykov, 2009). We used the max-flow/min-cut code from (Boykov e Kolmogorov, 2004).

Figure 4.7 shows a flower segmentation using $\omega_i(s,t) = G(s) + G(t)$, where G(.) 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 4.7b).



Figure 4.7: 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.

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 4.1 shows the running times for the flower segmentation using different image resolutions in a laptop Intel Core i3-5005U CPU 2.00GHz \times 4.

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

Table 4.1: The running times for the flower segmentation by HLOIFT and the min-cut/max-flow algorithm in layered graphs using different image sizes.

4.1.4 Quantitative accuracy experiments

For quantitative accuracy results, we compared HLOIFT, with the multi-object segmentation by IFT (Miranda e Falcão, 2009), its improved version by the relaxation procedure proposed by Malmberg *et al.* (2010), and the hierarchical layered graph cut (Delong e Boykov, 2009; Delong *et al.*, 2012), 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.¹ In order to stress the methods to their limit, in a more challenging situation, we only consider 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 (de Miranda *et al.*, 2010), nor enhancements based on deep learning techniques (Wolf *et al.*, 2017; Wolf *et al.*, 2019).

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 4.8a). Figure 4.8 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.

The mean accuracy curves according to the Dice coefficient are shown in Figure 4.9, 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.

¹Note 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 4.8: (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.



Figure 4.9: The mean curves of Dice accuracy for different methods.

4.2 3D experimental analysis

We present an experimental analysis of HLOIFT in 3-dimensional images (volumes) to assess its performance considering different high-level priors. We use a synthetic image of size $150 \times 150 \times 150$ voxels composed of six distinct regions: Two dark cubes contained in a brighter ball, which in turn is contained inside a bigger cube, surrounded by a white background, and a small ball contained in one of the inner cubes. Then, we want to segment it into three objects, where O_1 and O_2 are *included* in O_3 . By changing the boundary polarity prior from "bright to dark" (bd) to "dark to bright" (db) for O_1 and O_3 , we get different results for the same input image with the expected global properties, demonstrating that our algorithm is working correctly, as depicted in the slice views and 3D renderings of the objects in Figures 4.10 and 4.11.



Figure 4.10: (a) Given the input image with six regions and seeds for O_1 (in yellow), O_2 (in blue) and background (in red), we obtain the result for O_1 and O_2 included in O_3 , where O_1 has boundary polarity from bright to dark and O_2 from dark to bright. In (b) a slice view is shown and in (c) the 3D renderings of the objects.

In order to test our 3D HLOIFT in a real MR image, we reduced the image size by interpolation to $120 \times 120 \times 90 = 1296000$ voxels. We segmented the brain (Figure 4.12a) and the lateral ventricles (Figure 4.12b) using the inclusion relation. The running time was 4.2 sec in a laptop IntelCore i3-5005U CPU 2.00GHz4.

The main challenges correspond to the required space and execution time, since in multidimensional data the memory consumption grows exponentially with the increase in the number of dimensions. Here we present the results of applying HLOIFT to 3D volumes, through the direct application of its proposed algorithm, without taking any additional precautions to get a more moderate usage of memory, in order to test its viability in this more



Figure 4.11: (a) Given the input image with six regions and seeds for O_1 (in yellow), O_2 (in blue) and background (in red), we obtain the result for O_1 and O_2 included in O_3 , where O_1 and O_2 have boundary polarity from dark to bright, and O_3 has the opposite orientation from bright to dark. In (b) a slice view is shown and in (c) the 3D renderings of the objects.



Figure 4.12: Example of 3D HLOIFT segmentation in a real MR image. (a) The brain. (b) The lateral ventricles.

challenging scenario. One possible idea, as future work, would be to avoid the explicit creation of the inter-layer arcs, since they have repeated weights $-\infty$ or ∞ . When the intra-layer weights of different layers differ only in relation to their values of α_i , the base dissimilarity values $\psi(s,t)$ could be stored only once, being the orientation procedure of Equation 2.3 applied on-the-fly to save storage space. Another alternative solution is via the usage of supervoxels, since they considerably reduce the execution time as it will be demonstrated for superpixels in next section.

4.3 Experimental evaluation with superpixels

This section presents an experimental evaluation of the usage of superpixels in HLOIFT to assess its performance gains. In our first experiment, presented in Table 4.2, we show the execution time gains of the proposed approach in comparison to Image Foresting Transform (IFT) (Miranda e Falcão, 2009) and the multiple object segmentation by the regular HLOIFT (Leon *et al.*) without superpixels, for different image resolutions and superpixel sizes, for the segmentation of three objects in a CT image of the knee with inclusion and exclusion relations. The usage of superpixels in HLOIFT significantly reduced the size of the graph, resulting in a great saving of memory and computation time, thus compensating the additional cost of the three object-digraphs (layers) of HLOIFT in relation to IFT that has a single layer. Moreover, the segmentation results for different superpixel sizes were similar to those obtained by HLOIFT at the pixel level demonstrating the robustness of the proposed method (Figure 4.13).

	171×193	342×386	684×772	1368×1544
IFT (Miranda e Falcão, 2009)	8.46	29.26	106.61	333.13
HLOIFT (Leon <i>et al.</i>)	54.55	200.44	724.73	2,878.91
HLOIFT superpixel size 10×10	0.52	1.88	8.08	33.05
HLOIFT superpixel size 5×5	1.61	8.14	24.78	91.29
HLOIFT superpixel size 3×3	4.37	17.25	62.93	260.24

Table 4.2: Time in ms for the different methods and image resolutions.

We also compared the running time between our approach and the hierarchical mincut/max-flow algorithm with the inclusion case from Delong e Boykov (2009). Our approach has the best running time. For the flower segmentation from Figure 4.7, composed of two objects with the inclusion relation and size of 1520×1280 pixels, our algorithm takes 158.35ms, 58.06ms and 20.38ms for superpixels of size 3×3 , 5×5 and 10×10 , respectively, while the regular HLOIFT takes 1,823.55ms and the hierarchical min-cut/max-flow takes 19,021.71ms. All experiments presented in this section were conducted in a laptop Intel Core i3-5005U CPU 2.00GHz ×4.

SUPERPIXELS SIZE	HLOIFT	SUPERPIXELS
without superpixels (at the pixel level)		
3×3		
5×5		
10×10		

Figure 4.13: The segmentation of a CT image of the knee for different superpixel sizes.

Conclusions

5.1 Final considerations

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.

Finally, we presented how the HLOIFT method can be easily extended to work with 3D images. However, we note that under this last scenario, the usage of supervoxels can play an important role to circumvent memory concerns in machines with limited memory resources. Therefore, we also proposed a **superpixel**-based adaptation of the HLOIFT method, leading to a more efficient and adequate solution for multi-object segmentation in large images.

5.2 Publications

As results of this thesis, we have one conference publication, one submitted journal, one workshop publication in a book and three workshop poster presentations. These contributions are listed below in chronological order of submission:

- A conference paper (Leon e Miranda, 2017), in the 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) held in Niteroi, Rio de Janeiro. Qualis CAPES B1. Oral presentation. This paper was selected as one of the top papers of the event and we were invited to submit an extended version to a journal.
- 2. Submission of a journal paper (Leon *et al.*) at the special issue of the Journal of Visual Communication and Image Representation: *Special Issue on Feature representations*

for medical images and activity understanding. Qualis CAPES A2. This extended version was made in cooperation with Krzysztof Chris Ciesielski from Department of Mathematics, West Virginia University, USA.

- 3. A book chapter (Leon e de Miranda, 2019), from the workshop of Women in Computer Vision at European Conference in Computer Vision (WiCV@ECCV) held in Munich, Germany. Published in the ECCV Workshop proceedings. Poster presentation.
- 4. A workshop poster (Leon e Miranda, 2018), in the Latinx in AI Coalition at NeurIPS (LXAI@NeurIPS) held in Montréal, Canada. Poster presentation.
- 5. A workshop poster (Leon *et al.*, 2019b) in the Women in Computer Vision at CVPR (WiCV@CVPR), and a workshop poster (Leon *et al.*, 2019a) in the Latinx in AI Coalition at ICML (LXAI@ICML), both held in Long Beach, United States. Poster presentation.

5.3 Future works

As future work, we are interested in:

- 1. To study and test other high-level priors such as connectivity priors, and spatial interactions between objects, like their relative spatial positioning (up, down, left, right);
- 2. To evaluate our 3D image segmentation, also considering different high-level priors, in open databases like 3D-IRCADb-02 (Soler *et al.*, 2012), which contains axial cross sections of CT thoracic-abdominal images (see Figure 5.1).



Figure 5.1: Example of a thoracic-abdominal image from 3D-IRCADb-02 dataset (Soler et al., 2012).

3. We would like also to handle relations of partial intersection of objects, as shown in Figure 5.2, by using a negative parameter ρ ;



Figure 5.2: Example of cells with overlapping regions. In the left we have the input image, and in the right a desired segmentation.

- 4. We are interested in using a machine learning method to generate improved weighted graphs and to adapt our method for an automatic medical image segmentation task.
- 5. We also intend to combine some training information by deep learning approaches in HLOIFT, similar to what was done in Wolf *et al.* (2017); Wolf *et al.* (2019), and to incorporate in HLOIFT the relaxation procedure for directed graphs from Demario e Miranda (2019), in order to circumvent the irregular boundaries from the max-norm energy of HLOIFT.

Appendix

A.1 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).

All notions considered below are based on the variables from Algorithm 3. 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 reminder of the execution of the algorithm. (Such change can occur only during the execution of lines 14 or 17, 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 $\varepsilon_{min}^h \cos t$) 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 consists precisely of the nodes from the set $S = \{\langle s, i \rangle : i \in \mathcal{L} \& s \in S_i\}$, that is, seed representations in the HLOIFT graph \mathcal{N} . Let X_0 denote the map X defined on S at this time. Since, at this time, we also have $X(t) = -\infty$ for every $t \in S$, these values cannot ever be changed, as the condition in line 13 is never satisfied. Also, for any $X : \mathcal{N} \to \{0, 1\}$,

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

In particular, any $X : \mathcal{N} \to \{0, 1\}$ 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).

By consistency of S with (C), the family \mathcal{X} of all $X \colon \mathcal{N} \to \{0, 1\}$ 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 versions of preservation properties (P) and (Q). The above discussion shows that the assumptions of both of these properties are satisfied directly after the execution of the initialization loop.

The property (P) reads as follows.

(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, $X(t) = \bar{X}(t)$.

Indeed, if $t \in S$, 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 8-17, 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 17, 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 8-17 ensures that X(t) = X(s). At the same time we must have that $\bar{X}(s) = \bar{X}(t)$, as otherwise

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

impossibility.

Therefore, indeed $X(t) = X(s) = \overline{X}(s) = \overline{X}(t)$, finishing the argument for (P). The second preservation property is as follows.

(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 $\overline{X} \in \mathcal{X}_0$. 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 extension to an $\hat{X} \in \mathcal{X}_0$.

We have $V(t) > -\infty$, as $V(t) \ge E > -\infty$. 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 8-17, 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 $\varepsilon_{\min}^h(\hat{X}) = \min\{\varepsilon_{\min}^{incl}(\hat{X}), \varepsilon_{\min}^{excl}(\bar{X})\} \ge E > -\infty$, where the equation is justified by $\varepsilon_{\min}^{excl}(\bar{X}) =$ $\varepsilon_{\min}^{excl}(\hat{X})$ (which holds, as we have $\{s \in \mathcal{N} : \bar{X}(s) = 1\} = \{s \in \mathcal{N} : \hat{X}(s) = 1\}$) while the inequality \ge by $\varepsilon_{\min}^{incl}(\hat{X}) \ge \min\{V(s) : s \in Q\} \ge V(t) \ge E$ and $\varepsilon_{\min}^{excl}(\bar{X}) \ge \varepsilon_{\min}^h(\bar{X}) = E$. 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 16. 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

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

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 $\varepsilon_{\min}^h(\hat{X}) > -\infty$.

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).

Bibliography

- Achanta et al.(2012) Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua e Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120. URL http://dx.doi.org/10.1109/TPAMI.2012.120. Citado na pág. 25
- Alexandre et al.(2015) Eduardo Barreto Alexandre, Ananda Shankar Chowdhury, Alexandre Xavier Falcao e Paulo A Vechiatto Miranda. IFT-SLIC: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. Em 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), páginas 337–344. Citado na pág. 25, 26
- Bai et al.(2017) Junjie Bai, Abhay Shah e Xiaodong Wu. Optimal multi-object segmentation with novel gradient vector flow based shape priors. *CoRR*, abs/1705.10311. URL http://arxiv.org/abs/1705.10311. Citado na pág. 14
- Bejar e Miranda(2015) Hans Harley Ccacyahuillea Bejar e 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. Citado na pág. 3
- Boykov e Funka-Lea(2006) Yuri Boykov e Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. International journal of computer vision, 70(2):109–131. Citado na pág. 3
- Boykov e Kolmogorov(2004) Yuri Boykov e 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. Citado na pág. 25, 35
- Ciesielski et al.(2007) K.C. Ciesielski, J.K. Udupa, P.K. Saha e Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Computer Vision and Image Understanding*, 107(3):160–182. ISSN 1077-3142. Citado na pág. 3
- Ciesielski et al. (2012) K.C. Ciesielski, J.K. Udupa, A.X. Falcão e P.A.V. Miranda. A unifying graph-cut image segmentation framework: algorithms it encompasses and equiv-

alences among them. Em *Proc. of SPIE on Medical Imaging: Image Processing*, volume 8314. doi: 10.1117/12.911810. Citado na pág. 3, 10

- Ciesielski e Udupa(2010) Krzysztof Chris Ciesielski e Jayaram K. Udupa. Affinity functions in fuzzy connectedness based image segmentation i: Equivalence of affinities. Comput. Vis. Image Underst., 114(1):146–154. ISSN 1077-3142. doi: 10.1016/j.cviu.2009.09.006. URL http://dx.doi.org/10.1016/j.cviu.2009.09.006. Citado na pág. 13, 20
- Ciesielski et al.(2014) Krzysztof Chris Ciesielski, Robin Strand, Filip Malmberg e Punam K. Saha. Efficient algorithm for finding the exact minimum barrier distance. Computer Vision and Image Understanding, 123:53 – 64. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2014.03.007. URL http://www.sciencedirect.com/science/ article/pii/S107731421400068X. Citado na pág. 3
- Ciesielski et al.(2018) Krzysztof Chris Ciesielski, Alexandre Xavier Falcão e Paulo A. V. Miranda. Path-value functions for which dijkstra's algorithm returns optimal mapping. Journal of Mathematical Imaging and Vision. ISSN 1573-7683. doi: 10.1007/ s10851-018-0793-1. URL https://doi.org/10.1007/s10851-018-0793-1. Citado na pág. 3
- **Cousty** et al.(2010) Jean Cousty, Gilles Bertrand, Laurent Najman e Michel Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *Pattern* Analysis and Machine Intelligence, IEEE Transactions on, 32(5):925–939. Citado na pág. 2
- de Miranda et al.(2010) P.A.V. de Miranda, A.X. Falcão e J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. Computer Vision and Image Understanding, 114(1):85 – 99. ISSN 1077-3142. doi: https://doi. org/10.1016/j.cviu.2009.08.001. URL http://www.sciencedirect.com/science/article/pii/ S1077314209001246. Citado na pág. xiii, 4, 5, 13, 20, 36
- de Moraes Braz et al.(2019) Caio de Moraes Braz, Paulo A. V. Miranda, Krzysztof Chris Ciesielski e Fábio A. M. Cappabianco. Graph-based segmentation with local band constraints. Em Michel Couprie, Jean Cousty, Yukiko Kenmochi e Nabil Mustafa, editors, Discrete Geometry for Computer Imagery, páginas 155–166, Cham. Springer International Publishing. ISBN 978-3-030-14085-4. Citado na pág. 7
- Delong e Boykov(2009) Andrew Delong e Yuri Boykov. Globally optimal segmentation of multi-region objects. Em Computer Vision, 2009 IEEE 12th International Conference on, páginas 285–292. IEEE. Citado na pág. 5, 7, 15, 22, 26, 29, 34, 35, 36, 41
- **Delong** et al.(2012) Andrew Delong, Lena Gorelick, Olga Veksler e Yuri Boykov. Minimizing energies with hierarchical costs. International journal of computer vision, 100(1): 38–58. Citado na pág. 5, 36
- **Demario e Miranda(2019)** C.L. Demario e P.A.V. Miranda. Relaxed oriented image foresting transform for seeded image segmentation. Em *IEEE International Conference on Image Processing (ICIP)*. Accepted, to appear. Citado na pág. 45
- **Dijkstra(1959)** E. W. Dijkstra. A note on two problems in connexion with graphs. *NU-MERISCHE MATHEMATIK*, 1(1):269–271. Citado na pág. 10
- Falcão et al.(2004) Alexandre X Falcão, Jorge Stolfi e Roberto de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(1):19–29. Citado na pág. 3, 10, 24
- Falcão et al.(1998) Alexandre X. Falcão, Jayaram K. Udupa, Supun Samarasekera, Shoba Sharma, Bruce Elliot Hirsch e Roberto de Alencar Lotufo. User-steered image segmentation paradigms: Live wire and live lane. Graphical Models and Image Processing, 60(4): 233–260. Citado na pág. 3
- Golodetz et al. (2017) Stuart Golodetz, Irina Voiculescu e Stephen Cameron. Simpler editing of graph-based segmentation hierarchies using zipping algorithms. *Pattern Recognition*, 70:44–59. Citado na pág. 2
- Gordon et al.(2016) Shiri Gordon, Irit Dolgopyat, Itamar Kahn e Tammy Riklin Raviv. Co-segmentation of multiple images into multiple regions: Application to mouse brain MRI. Em 13th IEEE International Symposium on Biomedical Imaging, ISBI 2016, Prague, Czech Republic, April 13-16, 2016, páginas 399–402. IEEE. ISBN 978-1-4799-2349-6. doi: 10.1109/ISBI.2016.7493292. URL http://dx.doi.org/10.1109/ISBI.2016.7493292. Citado na pág. 1
- Gorelick et al.(2017) Lena Gorelick, Olga Veksler, Yuri Boykov e Claudia Nieuwenhuis. Convexity shape prior for binary segmentation. *IEEE Trans. Pattern Anal. Mach. In*tell., 39(2):258–271. doi: 10.1109/TPAMI.2016.2547399. URL https://doi.org/10.1109/ TPAMI.2016.2547399. Citado na pág. 5
- Grady(2006) Leo Grady. Random walks for image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(11):1768–1783. Citado na pág. 3
- Gulshan et al.(2010) V. Gulshan, C. Rother, A. Criminisi, A. Blake e A. Zisserman. Geodesic star convexity for interactive image segmentation. Em Proceedings of Computer Vision and Pattern Recognition, páginas 3129–3136. Citado na pág. 5
- He et al.(2016) K. He, X. Zhang, S. Ren e J. Sun. Deep residual learning for image recognition. Em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 770–778. doi: 10.1109/CVPR.2016.90. Citado na pág. 3
- **Isack** et al.(2016) Hossam Isack, Olga Veksler, Milan Sonka e Yuri Boykov. Hedgehog shape priors for multi-object segmentation. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 2434–2442. Citado na pág. 5
- Jackowski e Goshtasby(2005) Marcel Jackowski e Ardeshir Goshtasby. A Computer-Aided Design System for Segmentation of Volumetric Images, páginas 251–272. Springer US, Boston, MA. ISBN 978-0-306-48608-1. doi: 10.1007/0-306-48608-3 7. Citado na pág. 2
- Leon e Miranda(2017) L. M. C. Leon e P. A. V. D. Miranda. Multi-object segmentation by hierarchical layered oriented image foresting transform. Em 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), páginas 79–86. doi: 10.1109/ SIBGRAPI.2017.17. Citado na pág. 17, 43
- Leon e Miranda(2018) Leissi M. Castaneda Leon e Paulo A. V. Miranda. A hierarchical layered graph approach for multi-label segmentation in 2d medical images. Citado na pág. 44
- Leon et al.() Leissi M. Castaneda Leon, K.C. Ciesielski e Paulo A. V. Miranda. Efficient hierarchical multi-object segmentation in layered graph (submitted). página https://www.math.wvu.edu/ kcies/SubmittedPapers/SS29.HLOIFT.pdf. Citado na pág. 27, 41, 43

- Leon et al.(2019a) Leissi M. Castaneda Leon, K.C. Ciesielski e Paulo A. V. Miranda. Extensions of the hierarchical graph approach in multi-region segmentation. LatinX in AI Workshop @ International Conference in Machine Learning (ICML) 2019, 2019a. Citado na pág. 44
- Leon et al.(2019b) Leissi M. Castaneda Leon, K.C. Ciesielski e Paulo A. V. Miranda. Extensions of the hierarchical graph approach in multi-region segmentation. Women in Computer Vision Workshop @ Computer Vision and Pattern Recognition (CVPR) 2019, 2019b. Citado na pág. 44
- Leon e de Miranda(2019) L.M. Castañeda Leon e P.A. Vechiatto de Miranda. Efficient interactive multi-object segmentation in medical images. Em Laura Leal-Taixé e Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, páginas 705–710, Cham. Springer International Publishing. ISBN 978-3-030-11018-5. Citado na pág. 5, 44
- Li et al.(2012) Xiaoqiang Li, Jingsong Chen e Huafu Fan. Interactive image segmentation based on grow cut of two scale graphs. Em Wenjun Zhang, Xiaokang Yang, Zhixiang Xu, Ping An, Qizhen Liu e Yue Lu, editors, Advances on Digital Television and Wireless Multimedia Communications, páginas 90–95, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-642-34595-1. Citado na pág. 3
- Malmberg et al.(2010) F. Malmberg, I. Nyström, A. Mehnert, C. Engstrom e E. Bengtsson. Relaxed image foresting transforms for interactive volume image segmentation. Em *Proc.SPIE*, volume 7623, páginas 7623 7623 11. Citado na pág. 36
- Maninis et al. (2018) K.-K. Maninis, S. Caelles, J. Pont-Tuset e L. Van Gool. Deep extreme cut: From extreme points to object segmentation. Em In IEEE Conf. on Computer Vision and Pattern Recognition, páginas 616–625. Citado na pág. 3
- Mansilla et al.(2016) L. A. C. Mansilla, P. A. V. Miranda e F. A. M. Cappabianco. Oriented image foresting transform segmentation with connectivity constraints. Em 2016 IEEE International Conference on Image Processing (ICIP), páginas 2554–2558. doi: 10.1109/ICIP.2016.7532820. Citado na pág. 5
- Mansilla e Miranda(2013a) L.A.C. Mansilla e P.A.V. Miranda. Image segmentation by oriented image foresting transform: Handling ties and colored images. Em 18th International Conference on Digital Signal Processing (DSP), páginas 1–6, Santorini, Greece. IEEE. Citado na pág. 12, 22, 24, 31
- Mansilla e Miranda(2013b) L.A.C. Mansilla e P.A.V. Miranda. Image segmentation by oriented image foresting transform with geodesic star convexity. Em Computer Analysis of Images and Patterns (CAIP), volume 8047, páginas 572–579, York, UK. Citado na pág. 5, 14, 15, 20
- Mansilla et al.(2013) L.A.C. Mansilla, M.P. Jackowski e P.A.V. Miranda. Image foresting transform with geodesic star convexity for interactive image segmentation. Em IEEE International Conference on Image Processing (ICIP), páginas 4054–4058, Melbourne, Australia. Citado na pág. 14, 20
- Miranda et al.(2009) P. A. V. Miranda, A. X. Falcao e J. K. Udupa. Cloud bank: A multiple clouds model and its use in mr brain image segmentation. Em 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, páginas 506–509. doi: 10.1109/ISBI.2009.5193095. Citado na pág. 6

- Miranda e Falcão(2009) Paulo AV Miranda e 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. Citado na pág. 29, 31, 36, 41
- Miranda e Mansilla(2014) Paulo AV Miranda e Lucy AC Mansilla. Oriented image foresting transform segmentation by seed competition. *Image Processing, IEEE Transactions* on, 23(1):389–398. Citado na pág. 2, 3, 5, 7, 11, 22, 26
- Mohammadianrasanani(2013) Seyedmehrdad Mohammadianrasanani. The use of a body-wide automatic anatomy recognition system in image analysis of kidneys. Dissertação de Mestrado, School of Technology and Health, Royal Institute of Technology, Flemingsberg, Sweden. Citado na pág. 6
- Oguz e Sonka(2014) Ipek Oguz e 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. Citado na pág. 5
- Rittner et al.(2014) Leticia Rittner, Jayaram K. Udupa e Drew A. Torigian. Multiple fuzzy object modeling improves sensitivity in automatic anatomy recognition. Em In Proceedings of SPIE on Medical Imaging: Image Processing, volume 9034, San Diego, California, USA. Citado na pág. 6
- Rother et al.(2004) Carsten Rother, Vladimir Kolmogorov e Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3): 309–314. ISSN 0730-0301. doi: 10.1145/1015706.1015720. Citado na pág. 3, 4
- Singaraju et al.(2008) Dheeraj Singaraju, Leo Grady e René Vidal. Interactive image segmentation via minimization of quadratic energies on directed graphs. Em Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, páginas 1–8. IEEE. Citado na pág. 5
- Soler et al.(2012) Luc Soler, Alexandre Hostettler, Vincent Agnus, Arnaud Charnoz, Jean-Baptiste Fasquel, Johan Moreau, Anne-Blandine Osswald, Mourad Bouhadjar e Jacques Marescaux. 3d image reconstruction for comparison of algorithm database : A patient-specific anatomical and medical image database. URL https://www.ircad.fr/research/3d-ircadb-02/. Citado na pág. xv, 32, 44
- Sun et al.(2014) Kaioqiong Sun, Jayaram K. Udupa, Dewey Odhner, Yubing Tong e Drew A. Torigian. Automatic thoracic anatomy segmentation on ct images using hierarchical fuzzy models and registration. Em In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling, volume 9036, San Diego, California, USA. Citado na pág. 6
- Tavares et al. (2017) Anderson Carlos Moreira Tavares, Hans Harley Ccacyahuillca Bejar e Paulo André Vechiatto Miranda. Seed robustness of oriented image foresting transform: Core computation and the robustness coefficient. Em Jesús Angulo, Santiago Velasco-Forero e Fernand Meyer, editors, Mathematical Morphology and Its Applications to Signal and Image Processing, páginas 119–130, Cham. Springer International Publishing. ISBN 978-3-319-57240-6. Citado na pág. 7
- **Toennies(2012)** Klaus D Toennies. Guide to medical image analysis: methods and algorithms. Springer Science & Business Media. Citado na pág. 1

- Tong et al.(2013) Yubing Tong, J. K. Udupa, D. Odhner, Sanghun Sin e R. Arens. Abdominal adiposity quantification at MRI via fuzzy model-based anatomy recognition. Em In Proceedings of SPIE on Medical Imaging: Biomedical Applications in Molecular, Structural, and Functional Imaging, volume 8672, Orlando, Florida, USA. Citado na pág. 6
- Udupa et al.(2013) Jayaram K. Udupa, Dewey Odhner, Yubing Tong, Monica M. S. Matsumoto, Krzysztof C. Ciesielski, Pavithra Vaideeswaran, Victoria Ciesielski, Babak Saboury, Liming Zhao, Syedmehrdad Mohammadianrasanani e Drew Torigian. Fuzzy model-based body-wide anatomy recognition in medical images. volume 8671, páginas 86712B-86712B-7. doi: 10.1117/12.2007983. URL http://dx.doi.org/10.1117/12.2007983. Citado na pág. 6
- Udupa et al.(2014) 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 e Drew A. Torigian. Body-wide hierarchical fuzzy modeling, recognition, and delineation of anatomy in medical images. Medical Image Analysis, 18(5): 752 771. ISSN 1361-8415. doi: http://dx.doi.org/10.1016/j.media.2014.04.003. URL http://www.sciencedirect.com/science/article/pii/S1361841514000498. Citado na pág. 6
- Udupa et al.(2011) J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, P. Vaideeswaran, S. Mishra, G.J. Grevera, B.Saboury e D.A. Torigian. Fuzzy object modeling. Em *Proc.SPIE*, volume 7964, páginas 7964 – 7964 – 10. doi: 10.1117/12.878273. Citado na pág. 6
- Udupa et al.(2012) J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, S. Mishra, G.J. Grevera, B. Saboury e D.A. Torigian. Automatic anatomy recognition via fuzzy object models. Em In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling, volume 8316, San Diego, California, USA. Citado na pág. 6
- Ulén et al.(2013) Johannes Ulén, Petter Strandmark e Florian Kahl. An efficient optimization framework for multi-region segmentation based on lagrangian duality. Medical Imaging, IEEE Transactions on, 32(2):178–188. Citado na pág. 5
- Veksler (2008) Olga Veksler. Star shape prior for graph-cut image segmentation. Computer Vision-ECCV 2008, páginas 454-467. Citado na pág. 5, 14, 20
- Vicente et al. (2008) Sara Vicente, Vladimir Kolmogorov e Carsten Rother. Graph cut based image segmentation with connectivity priors. Em Computer vision and pattern recognition, 2008. CVPR 2008. IEEE conference on, páginas 1–8. IEEE. Citado na pág. 5
- Visser et al.(2016) Eelke Visser, Max C. Keuken, Gwenaelle Douaud, Veronique Gaura, Anne-Catherine Bachoud-Levi, Philippe Remy, Birte U. Forstmann e Mark Jenkinson. Automatic segmentation of the striatum and globus pallidus using mist: Multimodal image segmentation tool. NeuroImage, 125:479 – 497. ISSN 1053-8119. doi: http://dx.doi.org/10. 1016/j.neuroimage.2015.10.013. URL http://www.sciencedirect.com/science/article/pii/ S1053811915009143. Citado na pág. 1
- Wolf et al.(2017) S. Wolf, L. Schott, U. Köthe e F. Hamprecht. Learned watershed: Endto-end learning of seeded segmentation. Em 2017 IEEE International Conference on Computer Vision (ICCV), páginas 2030–2038. doi: 10.1109/ICCV.2017.222. Citado na pág. 5, 36, 45

- Wolf et al.(2019) Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe e Fred A. Hamprecht. The mutex watershed and its objective: Efficient, parameter-free image partitioning. Available: https://arxiv.org/abs/1904.12654, 2019. Citado na pág. 5, 36, 45
- Yin et al.(2010) Yin Yin, Xiangmin Zhang, Rachel Williams, Xiaodong Wu, Donald D Anderson e 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. Citado na pág. 5

Index

 $\mathrm{CNN},\,3$

CNN, see Convolutional Neural Networks

CT, 6

 CT , $\ see$ Computed Tomography

FOMs, 6

FOMs , see Fuzzy Object Models

GC, **3**

 GC , $\ see$ Graph Cut

GGC, 3

 ${\rm GGC}$, $\ see$ Generalized Graph Cut

GSC, 14

 GSC , see Geodesic Star Convexity Constraints

HLOIFT, 6

HLOIFT , *see* Hierarchical Layered Oriented Image Foresting Transform

IFT, 3, 6, 10, 11

 IFT , $\ see \ \operatorname{Image}$ Foresting Transform

MRI, 2, 31

MRI, see Magnetic Resonance Image

OIFT, 2, 3, 7, 11, 13

OIFT , see Oriented Image Foresting Transform

ORFC, 3

 ORFC , see Oriented Relative Fuzzy Connectedness

SSP, 14

 SSP , $\ see$ Star Shape Prior