Segmentação de imagens baseada em grafos com modelos de forma e restrições de banda local

Segmentação com Restrição de Banda Local

Caio de Moraes Braz

Thesis presented to the
Institute of Mathematics and Statistics
of the University of São Paulo
in partial fulfillment
of the requirements
for the degree of
Doctor of Science

Program: Ciência da Computação
Advisor: Prof. Dr. Paulo André Vechiatto de Miranda

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

São Paulo
March, 2023

Segmentação de imagens baseada em grafos com modelos de forma e restrições de banda local

Segmentação com Restrição de Banda Local

Caio de Moraes Braz

This version of the thesis includes the corrections and modifications suggested by the Examining Committee during the defense of the original version of the work, which took place on March 3, 2023.

A copy of the original version is available at the Institute of Mathematics and Statistics of the University of São Paulo.

Examining Committee:

Prof. Dr. Paulo André Vechaitto de Miranda (orientador) — IME-USP
Prof. Dr. Roberto Hirata Jr. — IME-USP
Prof. Dr. Fabio Augusto Faria — UNIFESP
Prof. Dr. Luis Gustavo Nonato — ICMC-USP
Prof. Dr. Silvio Jamil Ferzoli Guimarães — PUC-MG

Agradecimentos

Contra a estupidez os próprios deuses lutam em vão
— Friedrich Schiller

Primeiramente, agradeço aos meus pais, Sandra Regina e José Luis, que sempre me guiaram e apoiaram durante toda essa jornada, sem o apoio deles, nada disso seria possível.

Agradeço aos primeiros professores, lá da época da escola, que me fornceram a base sólida e fértil na qual todo o conhecimento acadêmico adquirido pôde florescer livremente.

Agradeço aos meus grandes amigos que tive e tenho a honra de compartilhar minhas alegrias, conquistas, medos e frustrações, todos eles foram e são importantes cada um à sua maneira, então todos também tem uma parte neste trabalho.

Não posso deixar de agradecer ao Instituto de Matemática e Estatística da USP, que proporcionou a estrutura necessária para que eu pudesse evoluir como estudante, acadêmico e principalmente como ser humano. Também quero incluir todos os funcionários que sempre dão o melhor para atender todos, vocês merecem o mundo!

Agradeço também a todos os docentes nos quais pude me espelhar durante toda essa trajetória, em especial aos do departamento de ciência da computação, tenho um carinho especial por todos, mas gostaria de direcionar agradecimentos especiais ao Prof. Coelho, que além de ter orientado meu TCC da graduação, sempre esteve aberto a conversas e foi um dos melhores professores que tive a honra de ter aulas; ao Prof. Hirata, que durante a maior parte do tempo que estive no mestrado e doutorado, foi o coordenador do programa e sempre esteve disponível para quaisquer problemas que surgissem; e à Profª. Yoshiko, que mesmo sem saber foi uma das minhas maiores referências acadêmicas e foi super importante lá no começo da minha graduação, quando aquele jovem perdido interessado em ser monitor foi gentilmente recebido por ela para explicar como tudo funcionava. Talvez tenha sido apenas mais um dia comum para ela, para mim valeu o mundo.

Claro, agradecimentos mais do que especiais ao meu orientador, Prof. Paulo Miranda, que me “adotou” como orientando em 2013 para o mestrado, e cá estamos terminando este ciclo. Não tenho como colocar em palavras o quanto foram importantes todo o apoio, conversas, viagens, aprendizado que tive a honra de compartilhar, a partir do fechamento deste ciclo, não serei mais seu orientando, mas espero poder ter a oportunidade de trabalhar contigo novamente no futuro.

Por último, mas não menos importante, meu agradecimento à minha psicóloga, que desde 2017 tem sido parte essencial da minha vida, sem ela, a depressão da qual fui e ainda estou sendo vítima, teria me corroído muito antes, mas acabou me “pegando” de vez nesta reta final do doutorado, tendo dificultado muito o progresso final, sem o apoio dela, não sei como isso teria terminado. Cuidem da saúde mental de vocês, é muito importante!

Resumo

Caio de Moraes Braz. **Segmentação de imagens baseada em grafos com modelos de forma e restrições de banda local: *Segmentação com Restrição de Banda Local***. Tese (Doutorado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2023.

O objetivo deste trabalho é descrever um algoritmo eficiente para encontrar uma segmentação binária de uma imagem tal que: o objeto indicado satisfaz uma nova restrição de alto nível, chamada restrição de banda local, LB; a segmentação devolvida é ótima, em respeito a uma medida de corte em grafos apropriada, entre todas as segmentações que satisfaçam a restrição LB dada. O novo algoritmo tem dois estágios: expandir o número de arcos de um grafo tradicional de imagem; aplicando a este novo grafo com pesos, um algoritmo conhecido como Transformada Imagem-Floresta Orientada, OIFT. Em nossos trabalhos teóricos, discutimos as relações teóricas da LB com outras restrições de forma e provamos que o algoritmo da OIFT pertence a uma classe de algoritmos gerais de conexão difusa (*General Fuzzy Connectedness*) e, portanto, possui várias propriedades teóricas, como robustez ao posicionamento de sementes. A extensão do grafo construído no primeiro estágio garante, como provamos, que o objeto resultante realmente satisfaz a restrição LB dada. Para efeitos de eficiência computacional, consideramos o menor número de arcos possível necessário para garantir a restrição. Esta construção de grafo é flexível o suficiente para permitir combiná-la com outras restrições de alto nível. Para o caso particular da LB com raio infinito, caso este chamado de restrição de banda, também apresentamos um algoritmo eficiente, com prova de corretude, que pode ser aplicado diretamente sobre o grafo de imagem original. Finalmente, demonstramos experimentalmente que a restrição LB possui resultados competitivos quando comparada com a convexidade geodésica em estrela, restrição de banda de borda e *Hedgehog Shape Prior*, todos implementados dentro do arcabouço da OIFT e aplicados a vários cenários envolvendo imagens naturais e médicas.

Palavras-chave: restrição de banda. hedgehog shape prior. transformada imagem-floresta. segmentação por corte em grafos.

Abstract

Caio de Moraes Braz. **Segmentação de imagens baseada em grafos com modelos de forma e restrições de banda local:** ***Segmentação com Restrição de Banda Local***. Thesis (Doctorate). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2023.

The goal of this work is to describe an efficient algorithm for finding a binary segmentation of an image such that: the indicated object satisfies a novel high-level prior, called Local Band, LB, constraint; the returned segmentation is optimal, with respect to an appropriate graph cut measure, among all segmentations satisfying the given LB constraint. The new algorithm has two stages: expanding the number of arcs of a standard edge-weighted graph of an image; applying to this new weighted graph an algorithm known as an Oriented Image Foresting Transform, OIFT. In our theoretical investigations, we discuss the theoretical relationships of LB with other shape constraints and prove that OIFT algorithm belongs to a class of General Fuzzy Connectedness algorithms and so, has several good theoretical properties, like robustness for seed placement. The extension of the graph constructed in the first stage ensures, as we prove, that the resulted object indeed satisfies the given LB constraint. For purposes of computational efficiency, we consider the least number of arcs needed to guarantee the constraint. This graph construction is flexible enough to allow combining it with other high-level constraints. For the particular case of LB with infinite radius, this case called Band constraint, we also present an efficient algorithm, with proof of correctness, which can be applied directly to the original image graph. Finally, we experimentally demonstrate that the LB constraint gives competitive results as compared to Geodesic Star Convexity, Boundary Band, and Hedgehog Shape Prior, all implemented within OIFT framework and applied to various scenarios involving natural and medical images.

Keywords: restrição de banda. hedgehog shape prior. transformada imagem-floresta. segmentação por corte em grafos.

List of Figures

List of Tables

List of Programs

Contents

Chapter 1

Introduction

In the recent decades, the volume of collected digital image data is increasing exponentially, reaching such a volume, that it becomes impossible to interpret it without computer tools that essentially reduce human input in such task. It is especially an acute problem with medical images, in which the three dimensional structure of medical volumes usually requires that radiologists, who interpret the images, to have many years of study to ensure the correct image interpretation necessary for correct diagnosis. Thus, it is imperative to create for them the computer tools that reduce as much as possible the time necessary for the radiologists to interpret the images. One of the most useful computer tools that are created towards this goal are image segmentation algorithms [70].

Image segmentation is one of the most fundamental and challenging problems in image processing and computer vision, with applications beyond medical and biological image analysis, including the digital matting [2] and machine perception. In many scenarios, the high-level, application-domain specific knowledge of the user is often required in the segmentation process because of the presence of heterogeneous backgrounds, objects with ill-defined borders, field inhomogeneity, noise, artifacts, partial volume effects, and their interplay [53].

Image segmentation may be thought of as consisting of two related processes; object recognition and delineation [27]. Recognition is the task of determining an object's approximate whereabouts in the image. Delineation completes segmentation by defining the exact spatial extent of that object. While the user can often solve the recognition problem by simple point (seed) selection or by an effective initialization action, precise manual delineation is challenging due to the intra and inter operator variation and drudgery, and human failure. On the other hand, computers can be very precise, even when they are not accurate.

The segmentation problem can be interpreted as a graph partition problem subject to hard constraints, such as seed pixels selected in the image domain for object recognition, by modelling neighborhood relations of picture elements from digital images [9]. Examples of seed-based methods are watershed [19], random walks [35], fuzzy connectedness [15], graph cuts (GC) [4], grow cut [50], minimum barrier distance [18], and image foresting transform (IFT) [26, 16]. 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 Graph Cuts (GGC) [13]. *Oriented Image Foresting Transform* (OIFT) [67] and *Oriented Relative Fuzzy Connectedness* (ORFC) [3] are extensions of some GGC methods for directed weighted graphs and have lower computational complexity compared to the min-cut/max-flow algorithm [4].

Image segmentation can also be interpreted as a classification problem at the pixel level, given that image partitions are obtained by assigning distinct labels to their composing pixels. Not surprisingly, machine learning-based methods are among the most prominent solutions to the segmentation problem, especially after the advent of deep learning techniques [39, 55, 96, 74, 63, 52, 49, 12, 8].

The integration of machine learning techniques with methods built on the strong formalism of graph partitions has become a very relevant research topic, as attested by the popularity of the method proposed in [28] that achieves state-of-the-art results in various databases using a combination of deep learning and hierarchical approaches by watershed-based techniques. Besides being easily extensible for multidimensional images, graph-based methods allow several customizations, such as the usage of different image elements as its nodes (e.g., superpixels [92]), the learning of graph weights by means of machine learning techniques [85, 66], or even the incorporation of learning by deep learning techniques [99, 98]. There are different ways to integrate probability maps by Convolutional Neural Networks (CNN) with optimization frameworks in graphs, such as to estimate arc weights and to define seeds or other constraints [71].

In this work, we are interested in solving the delineation problem with a high accuracy by fast seed-based methods in graphs to efficiently deal with large amounts of data, but which must also be versatile enough to support the inclusion of high-level constraints from prior object knowledge. The proposed methods, developed in the strong formalism of graphs, could also be used as an additional layer in a segmentation pipeline, guaranteeing the theoretical establishment of the formal properties of the generated objects, increasing the robustness of the obtained results[1]. We are especially interested in developing hard shape constraints, as opposed to soft constraints. A soft constraint imposes a penalty on certain labeling assignments rather than prohibiting them. A hard constraint corresponds to a limit case of the penalty value, ensuring the desired high-level features (e.g., shape constraints), which can be learned from a training dataset.

Segmentation methods using shape constraints can be divided into two groups:

1. Methods that execute simple delineation algorithms, generating a list of candidate segmentations and then use shape constraints in a later recognition step, in order to filter unwanted results from the list and select some for further attention.

2. Methods that perform more complex delineation algorithms that treat shape constraints in an integrated way as a priori knowledge, aiming to handle more challenging cases.

[1] Such objectives are aligned with explainable AI (XAI), which is a huge obstacle towards making AI more trustable and interpretable, enabling it to be used in real-world problems.

Methods of the first group generally require well-controlled environments and/or objects with strong contrast at their boundaries. As an example of a method from the first group, we can cite the algorithm by Garrido-Jurado et al. [31] used for detecting ArUco markers, which are binary square fiducial markers that can be used for camera pose estimation. In [31], a local adaptive thresholding approach is used for delineation and a contour extraction is performed on the thresholded image to produce a list of candidate contours. Then, a polygonal approximation is performed using the Douglas–Peucker algorithm [25] for the removal of irrelevant contours. Since markers are enclosed in rectangular contours, those that are not approximated to 4-vertex polygons and that do not define a convex polygon are discarded by shape constraint. The remaining selected contours are then inspected for extracting their internal binary code for identification. Another example of the first group corresponds to the filtering of connected components by threshold decomposition by the attribute of circularity, which is based on the incremental computation of the attributes of area and perimeter of a component tree [83].

In this work we are interested in developing new methods belonging to the second group, where shape constraints are used as a priori knowledge to circumvent problems due to weak edges and the presence of multiple objects with similar intensity profiles. Although there are several methods that directly or indirectly use shape constraints in their formulations, including atlas-guided brain segmentation [30, 72] and active contour models [46, 78, 20, 21, 11, 75, 87], here we will restrict our scope to optimization methods on graphs subject to shape constraints aiming at globally optimal segmentations. Although some high-level shape constraints have been previously successfully incorporated in other classes of optimal segmentation algorithms in graphs, like those based on min-cut/max-flow algorithm [95, 54, 29, 22, 84, 41, 40], these classes of algorithms have higher computational complexity than the Dijkstra-like algorithm OIFT we plan to use.

In the works [95, 29], for each execution of the min-cut/max-flow algorithm, the shape prior is considered as a soft constraint, only penalizing segmentations that differ from a given binary template of the expected object. Soft constraints are also considered in [54], but the template is estimated as a fuzzy map from a training set of aligned binary shapes using kernel PCA. In [29], the shape prior is embedded in the edge weights of neighboring pixels in the graph (pairwise term), being the penalty value proportional to a distance function from the boundary of the binary template, while in [95, 54] the shape prior is embedded in the weights of arcs interconnecting pixels with terminal nodes (unary term). Increasing the penalty value in order to obtain a hard shape constraint in these methods from [95, 29] is not useful, as it would lead to a segmentation practically identical to the given binary template, disregarding all information present in the image. Hedgehog Shape Prior [41, 40] is more strongly related to the objectives of our work, since it corresponds to the definition of a genuine hard shape constraint. Hedgehog defines different classes of shapes, according to its configuration parameters. Hedgehog will be explained in more detail in the next chapter. Compact Shape Prior [22] also defines a class of shapes by a hard constraint, but it is very restrictive and does not generalize to highly variable shapes, having a very limited scope of applications. Regarding the reference [84], although the elliptical shape prior can model a wide variety of anatomical structures like blood vessels and lymph nodes, the heuristic solution presented in [84] via an iterative graph cut method does not have any theoretical guarantees that it will present satisfactory results in more

challenging scenarios.

1.1 Motivation

The processing of global object properties, such as connectedness [56, 59, 94], shape constraints [57, 5, 48], boundary polarity [58, 3] and hierarchical constraints [45, 44], have been successfully incorporated in OIFT algorithm. These high-level priors are potentially useful for object segmentation, allowing the customization of the segmentation to a given target object. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason, considerable care should be taken to improve the likelihood of obtaining an accurate segmentation [34]. Shape constraints can be used to eliminate undesirable intricate forms, improving the segmentation of objects with more regular contour, or to enforce the result to be similar to a given template. Some shape constraints demand more sophisticated algorithms, such as the Boundary Band Constraint [5]. The OIFT with the Boundary Band Constraint allows the segmentation to follow a pre-established template of shapes, with variances within a range of permitted deformations around an arbitrary scale, while other approaches handle scale inefficiently based on brute force, by computing the graph cut for each level of a gaussian pyramid [29].

Despite the progress made so far, there are still many issues not fully covered in this research topic. The sensitivity of the methods with respect to seed position/template positioning is a problem to be addressed and the development of new classes of more robust shape constraints are necessary. Also, other constraints, such as the Hedgehog Shape Prior [41, 40], have not yet been exploited in the OIFT method and their relations, from an experimental point of view, with the Boundary Band Constraint still need to be analyzed. Lastly, the new developed shape constraints should also be easy to generalize for multidimensional images, such as in the segmentation of three-dimensional magnetic resonance imaging and computed tomography, and in the case of large volumes, concerns about memory consumption must be taken into account.

1.2 Overview of the Contributions

In this research, we propose a novel boundary constraint for object segmentation to be used in the Generalized Graph Cut framework, denoted as *Local Band Constraint* (LB), which in its limit case is strongly related to Boundary Band Constraint [5]. In order to validate this novel high-level prior, we instantiate it through a new efficient algorithm for finding a binary segmentation of an image such that:

- The indicated object satisfies the Local Band, LB, constraint;
- The returned segmentation is optimal, with respect to an appropriate graph cut measure, among all segmentations satisfying the given LB constraint.

The new algorithm has two stages: expanding the number of arcs of a standard edge-weighted graph of an image; applying to this new weighted graph an algorithm known as an Oriented Image Foresting Transform, OIFT. The extension of the graph constructed

in the first stage ensures, as we prove, that the resulted object indeed satisfies the given LB constraint. For purposes of computational efficiency, we consider the least number of arcs needed to guarantee the constraint in the expanded graph. This graph construction is flexible enough to allow combining it with other high-level constraints, that can be embedded in the arc-weight formulation. For instance, it can be easily combined with other high-level priors already supported by OIFT, such as the boundary polarity, considerably advancing the targeted segmentation [47]. In theory, it can also be exploited in the layer graphs of *Hierarchical Layered Oriented Image Foresting Transform* (HLOIFT) [45].

From a theoretical point of view, we have the following contributions:

- We prove that OIFT algorithm belongs to a class of *General Fuzzy Connectedness* algorithms (GFC) [17] and so, has several good theoretical properties, like robustness for seed placement. This contribution was conducted in cooperation with researcher Krzysztof C. Ciesielski from Department of Mathematics, West Virginia University, USA, who is one of the authors of GFC [17].

- We propose a new shape constraint, entitled Local Band constraint (LB), aiming at a greater robustness in relation to the positioning of its template, and we discuss its theoretical relationships with other shape constraints, such as the Boundary Band constraint [5] and its variants.

- We propose an efficient algorithm, as discussed above, that returns optimal segmentation, among all segmentations satisfying the LB constraint, with the fewest arcs needed [69].

- For the particular case of LB with infinite radius, this case called Band constraint, we also present an efficient algorithm to compute an optimal segmentation, with proof of correctness, which can be applied directly to the original image graph. The previous solution via the usage of an expanded graph would not be feasible in this case, due to the excessive number of arcs [7].

From the experimental point of view, we also introduce the usage of the Hedgehog constraint [41, 40] in the OIFT method. The LB constraint demonstrates competitive results, being less sensitive to the seed/template positioning for high accuracy values, compared to Boundary Band [5], Hedgehog [41, 40] and Geodesic Star Convexity [37], when applied to various scenarios involving natural and medical images. We also discuss how to create shape templates by Gielis equation, which has been used to simulate many natural shapes, such as, diatoms, eggs, cross sections of plants, snowflakes and starfish [81].

As a secondary contribution, we also propose a new algorithm for the efficient computation of the OIFT segmentation based on energy competition of two wavefronts. Compared to native IFT-based implementations, the new algorithm only computes priority queue operations whenever it is strictly necessary and uses a simple stack data structure otherwise. In the case of priority queue by binary heap, the gains are more pronounced, since many pixels are processed by depth-first search taking linear time in the size of the graph, rather than log-linear time. This is especially important when we do not want to restrict the arc weights to be small integers.

This thesis resulted in the following publications, listed below in chronological order of publication:

- 2019 Discrete Geometry for Computer Imagery - DGCI, Paris, France:
  Graph-Based Segmentation with Local Band Constraints [69]
- June 2020 issue of Journal of Mathematical Imaging and Vision (JMIV):
  Optimum Cuts in Graphs by General Fuzzy Connectedness with Local Band Constraints [68]
- 2022 Discrete Geometry and Mathematical Morphology - DGMM, Strasbourg, France:
  Graph-Based Image Segmentation with Shape Priors and Band Constraints [7]

1.3 Text Outline

In this section, we present the distribution of topics to be covered in the next chapters.

- In Chapter 2, we explore the previous works based on the Oriented Image-Foresting Transform and shape priors, to set a starting point for our research.
- In Chapter 3, we prove that OIFT algorithm belongs to a class of General Fuzzy Connectedness algorithms and present our proposed algorithm for computing the OIFT subject to the Local Band constraint, with the formal definitions, structures, theoretical relationships with other shape constraints and experimental results.
- In Chapter 4, we present the secondary contribution of our proposed algorithm for computing the Fast-OIFT.
- In Chapter 5, we state our conclusions and give suggestions for future works.

Chapter 2

Background

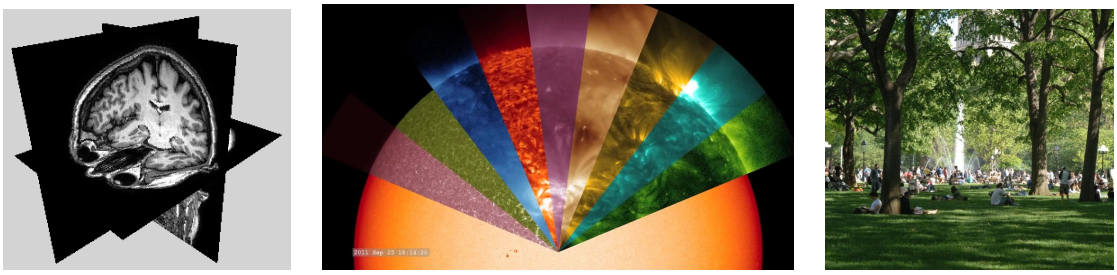
In this chapter, we expose the background of our research project, giving an overview of the concepts and related works.

2.1 Basic Image Concepts

Digital images come in a wide variety of kinds and structures. They are formed by elements called pixels, which have associated values that determine some types of intensity measures, such as brightness, color, depth, and transparency.

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

Some important and usual images structures are shown in Figure 2.1. An MR image usually is a monochrome ($m = 1$) three-dimensional image ($n = 3$). In this case, the picture elements are called *voxels* and the image is also named as a *volume* (Fig. 2.1 (a)). A multi-spectral image (e.g., satellite images) has one channel for each spectral band analyzed ($m > 1$), such as infra-red, ultra-violet, and x-rays (Fig. 2.1 (b)). A traditional bidimensional ($n = 2$) RGB image (Fig. 2.1 (c)) has one channel for each color ($m = 3$). Other valuable examples include video (images along the time) and stereoscopic images (overlapping images, taken from different view points).



(a) 3D MR image (b) Multi-spectral satellite image (c) 2D RGB image

Figure 2.1: *Some distinct types of images.*

It is important to note that greater values of n imply in an exponential increase of data required to store the images, making efficiency a key point when dealing with image processing.

2.2 Image as a Graph and Image Segmentation

In this section, we introduce the mathematical concepts for modelling digital images as a graph and the definitions needed to understand image segmentation.

2.2.1 Image as a Graph

An image can be interpreted as a directed graph (digraph) $G = \langle \mathcal{N}, \mathcal{A} \rangle$ whose nodes/vertices in $\mathcal{N}$ are the image pixels in its image domain $\mathcal{I} \subset Z^n$, and whose arcs, elements of $\mathcal{A}$, are the ordered pixel pairs $\langle s, t \rangle$ of vertices that are adjacent, that is, spatially close (e.g., 4-neighborhood, or 8-neighborhood, in case of 2D images). We use $t \in \mathcal{A}(s)$ and $\langle s, t \rangle \in \mathcal{A}$ to indicate that t is adjacent to s. We will usually assume also that our image graph G is arc-weighted, that is, that each arc $\langle s, t \rangle \in \mathcal{A}$ has a fixed weight $\omega(s, t) \in [-\infty, \infty]$ (e.g., $\omega(s, t) = |I(t) - I(s)|$ for a single channel image with values given by $I(t)$). An arc-weighted digraph will be denoted as $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$.

A digraph G is symmetric if, for all $\langle s, t \rangle \in \mathcal{A}$, the pair $\langle t, s \rangle$ is also an arc of G. Note that in the symmetric graphs we can still have $\omega(s, t) \neq \omega(t, s)$. In this work, all considered graphs are symmetric and connected.

A *path (in G)* of length $\ell \geq 0$ is any sequence $\pi_v = \langle v_0, \ldots, v_\ell \rangle$ of vertices, with *terminus* $v = v_\ell$, such that $\langle v_j, v_{j+1} \rangle \in \mathcal{A}$ for any $j < \ell$; it is *from* $\mathcal{S} \subset \mathcal{N}$ *to* $v \in \mathcal{N}$ when $v_0 \in \mathcal{S}$ and $v_\ell = v$; if $\langle s, t \rangle \in \mathcal{A}$, then $\pi_s \hat{} t$ denotes the path $\langle v_0, \ldots, v_\ell, t \rangle$ with $s = v_\ell$ (Figure 2.2a). All paths considered in this work are simple paths, that is, they pass through each vertex at most once. We use the notation $\pi_v = \langle v \rangle$ to indicate a path composed of a single vertex, which we denote as a *trivial path*.

Let Π_G be the family of all paths in G and consider a path-cost function $\psi : \Pi_G \to [-\infty, \infty]$ (e.g., Eq. 2.1, also known as geodesic). The cost $\psi(\pi_v)$ of a path π_v usually depends on local image properties, such as color, gradient, and vertex position along the path.

The cost of a trivial path $\pi_v = \langle v \rangle$ is usually given by a handicap value $H(v)$, while the connectivity functions for non-trivial paths follow a path-extension rule. As a classic example of connectivity functions, we have the additive path-cost function ψ_{sum}, with $\omega(s, t) \geqslant 0$, which is used to compute shortest paths (geodesic):

$$\begin{aligned} \psi_{sum}(\langle t \rangle) &= H(t) \\ \psi_{sum}(\pi_s \hat{} t) &= \psi_{sum}(\pi_s) + \omega(s, t) \end{aligned} \tag{2.1}$$

A *predecessor map* is a function P that associates with each pixel $t \in \mathcal{I}$ some other adjacent pixel in $\mathcal{I}$, or a distinct marker $nil \notin \mathcal{I}$. In the latter case t is a root node (Figure 2.2 (b)). A *spanning forest* is a map P of predecessors that does not contain cycles, i.e., when

returning by the map P from any vertex, we get to *nil* in a finite number of iterations (Figure 2.2(c)). Thus, for any pixel $t \in \mathcal{I}$, a forest P defines a path π_t^P recursively as $\langle t \rangle$, if $P(t) = nil$ and $\pi_s^P \hat{} t$ if $P(t) = s \neq nil$. We use $R(\pi_t^P)$ to denote the root node of the path π_t^P.

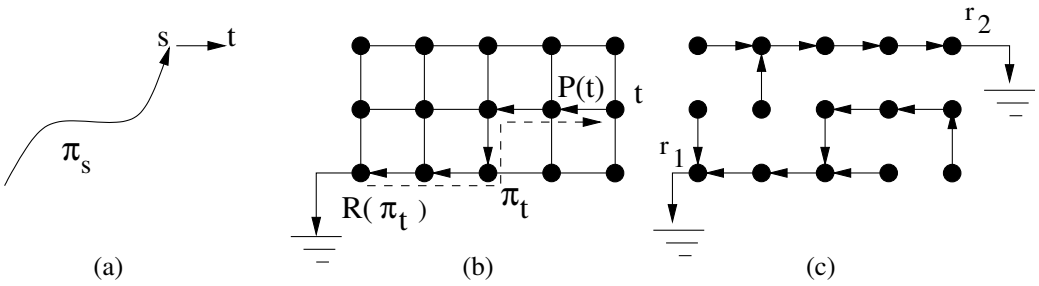


Figure 2.2: *(a) Extension of a path π_s by arc $\langle s, t \rangle$. (b) Neighborhood-4 graph showing a path π_t (dashed line) represented in reverse direction, where $P(t)$ is the predecessor node of t and $R(\pi_t)$ is the root pixel of the path π_t. (c) Spanning Forest P with two root nodes, r_1 and r_2.*

Let $\Pi_G(t)$ be the set of all possible paths in the graph $G = \langle \mathcal{N}, \mathcal{A} \rangle$ ending at t. A path π_t is optimal if $\psi(\pi_t) \leqslant \psi(\pi_t')$ for any other path $\pi_t' \in \Pi_G(t)$, that is, any other path that ends in t, regardless of its origin.

The value $V(t)$ of an optimal path for each node $t \in \mathcal{N}$ is defined as:

$$V(t) \quad = \quad \min_{\pi_t \in \Pi_G(t)} \{\psi(\pi_t)\}. \tag{2.2}$$

A path $\pi_{t_n} = \langle t_0, t_1, \ldots, t_n \rangle$ is *hereditarily optimum* if every path $\pi_{t_i} = \langle t_0, t_1, \ldots, t_i \rangle$, for $i = 0, 2, \ldots, n$, is an optimal path. Note that every path π_t^P defined by an optimal-path forest P is a *Hereditarily-optimum path.*

The Image Foresting Transform (IFT) algorithm [26] is an extension of Dijkstra algorithm for computing minimum cost paths [24], slightly modified to allow multiple sources and more general cost functions, that is, it obtains an optimal-path forest P for a given cost function ψ, which must satisfy some admissibility criteria [16].

2.2.2 Image Segmentation and Oriented Image Foresting Transform (OIFT)

Image segmentation can be formulated as a graph partition problem subject to hard constraints. In the case of binary segmentation (object/background), we consider two non-empty disjoint seed sets, $\mathcal{S}_1$ and $\mathcal{S}_0$ ($\mathcal{S}_1 \cap \mathcal{S}_0 = \emptyset$), containing pixels selected inside the object $\mathcal{O}$ and in its exterior, respectively. A label, $L(t) = 1$ for all $t \in \mathcal{S}_1$ and $L(t) = 0$ for all $t \in \mathcal{S}_0$, is propagated to all unlabeled pixels during the OIFT algorithm [67]. The same algorithm can also be used for multi-object segmentation by computing it in a hierarchical layered digraph [44]. For a label map $L : \mathcal{N} \rightarrow \{0, 1\}$ the object $\mathcal{O}$ identified with it is defined as the set $L^{-1}(1)$, where $L^{-1}(i) := \{t \in \mathcal{N} : L(t) = i\}$.

There are two important classes of energy formulations within the Generalized Graph Cut framework, the Max-Min [1] and Min-Sum optimizers [13]. OIFT and ORFC algorithms are Max-Min optimizers while the min-cut/max-flow algorithm is a Min-Sum optimizer. The resulting segmentation by OIFT gives a global optimum solution by maximizing the graph-cut measure $\varepsilon_{\min}$ (Eq. 2.3) subject to the seed constraints [67].

$$\varepsilon_{\min}(L) \quad = \quad \min_{(s,t)\in\mathcal{A}|L(s)>L(t)} \omega(s,t) \tag{2.3}$$

Oriented Image Foresting Transform (OIFT) [67] is a particular case of the IFT algorithm. The segmentation L by OIFT can be computed by Algorithm 1, as described in [58].

Algorithm 1. – *Segmentation Algorithm OIFT*

Input: Symmetric edge-weighted image digraph $\langle \mathcal{N}, \mathcal{A}, \omega \rangle$ and non-empty disjoint seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$.
Output: The label map $L : \mathcal{N} \to \{0, 1\}$.
Auxiliary: An initially empty set Q, variable tmp, and an array of status $S : \mathcal{N} \to \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes. The value $V(t)$ represents a potential penalty that a change of $L(t)$ would contribute to $\varepsilon_{\min}(L)$.

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

OIFT allows the segmentation favoring borders with transitions ranging from darker to brighter pixels or vice-versa (boundary polarity). For this purpose, we define the arc weights $\omega(s, t)$ as a combination of an undirected dissimilarity measure $\delta(s, t)$ (e.g., $\delta(s, t) = |I(t)-I(s)|$ for a single channel image with values given by $I(t)$), multiplied by an orientation factor α (Eq. 2.4), as illustrated in Figure 2.3. An example of the usage of the boundary polarity by OIFT, favoring transitions from darker to brighter pixels, in the case of a real MR image of the foot is shown in Figure 2.4.

[1] Min-Max optimizer is a dual equivalent problem.

$$\omega(s,t) = \begin{cases} \delta(s,t) \cdot (1+\alpha), & \text{if } I(s) > I(t), \\ \delta(s,t) \cdot (1-\alpha), & \text{if } I(s) < I(t), \\ \delta(s,t), & \text{otherwise.} \end{cases} \tag{2.4}$$

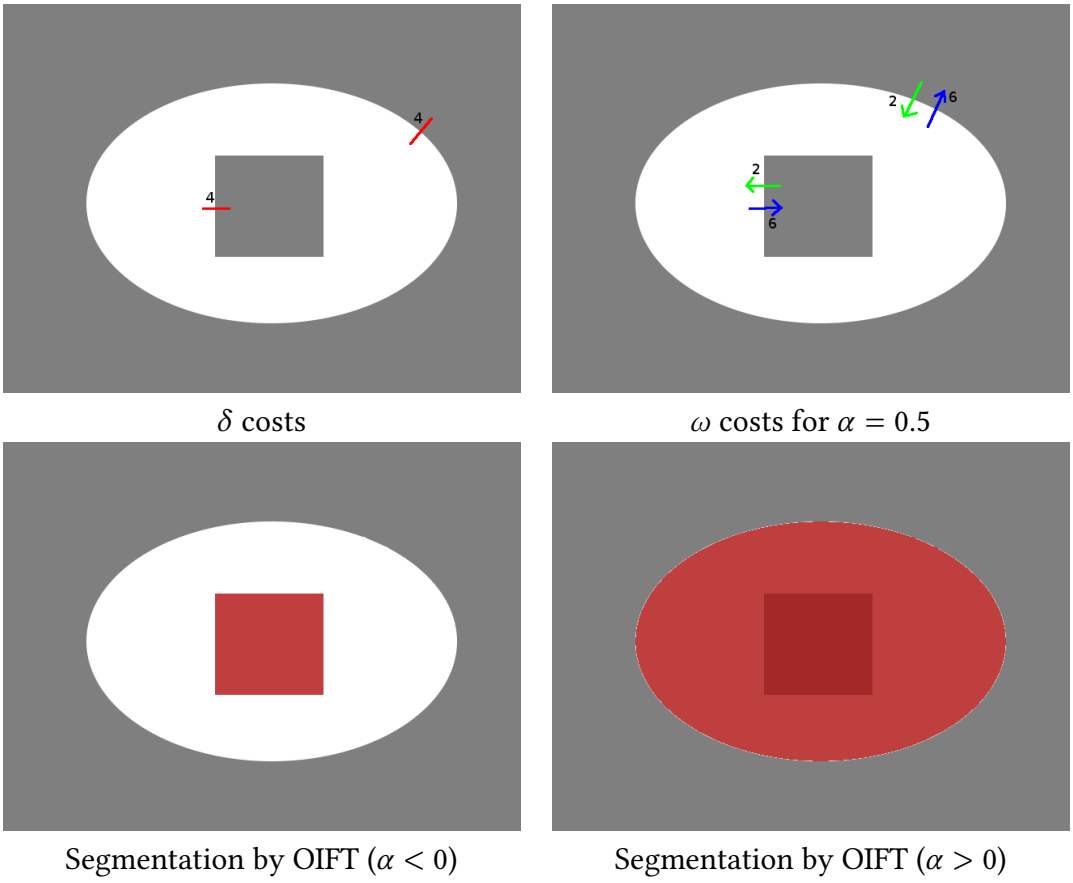


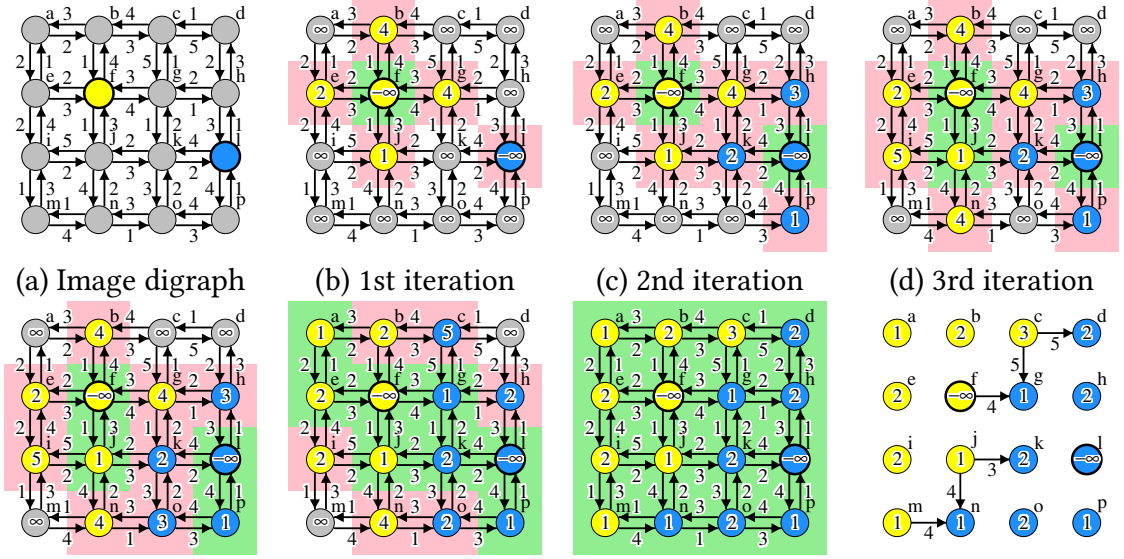
Figure 2.3: *Illustrative OIFT example, segmenting dark to bright transitions ($\alpha < 0$) and bright to dark ($\alpha > 0$).*

Figure 2.4: *Segmentation of the talus bone in an MR image of the foot. In (a), no boundary polarity was used ($\alpha = 0$), while in (b) we favored darker to brighter transitions ($\alpha < 0$).*

Proposition 1. [Mansilla, Miranda 2013] *Let $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$ be a symmetric edge-weighted image digraph. Let L be a segmentation returned by Algorithm 1 applied to G and non-empty disjoint seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$. Then L satisfies the seed constraints and maximizes the energy $\varepsilon_{\min}$, given by (2.3), among all segmentations satisfying these constraints.*

Notice that in line 12 of OIFT Algorithm 1 the weight $\omega(t, s)$ of the reversed parallel arc $\langle t, s \rangle$ is used (rather than that of chosen $\langle s, t \rangle \in \mathcal{A}$). That is why a symmetric digraph is required.

The OIFT algorithm is based on a label propagation from the seeds, such that the growing object avoids traversing high-weighted arcs. The node propagation priority is governed by map V, such that the lower the value $V(t)$ the higher its priority. Initially, seeds are initialized with maximum priority $V(t) = -\infty$ on lines 1 to 6. During propagation (lines 7 to 15), nodes with $S(t) = 1$ reach their final label assignment and can no longer be modified. Figures 2.5 (a-g) show a step-by-step illustration of the execution of Algorithm 1 and its resulting optimal cut is shown in Figure 2.5 (h).



(a) Image digraph (b) 1st iteration (c) 2nd iteration (d) 3rd iteration

(e) 4th iteration (f) 8th iteration (g) 16th iteration (h) The resulting cut

Figure 2.5: *An example of the step-by-step execution of Algorithm 1. (a) An image digraph with 4-neighborhood and seeds $\mathcal{S}_0 = \{l\}$ (in blue) and $\mathcal{S}_1 = \{f\}$ (in yellow). (b) The label propagation after one iteration of the main loop (Lines 7 to 15), where the node colors blue and yellow indicate the labels $L(t) = 0$ and $L(t) = 1$, respectively, a pink-colored background indicates pixels in Q, while a green background indicates processed nodes in $\{t \in \mathcal{N} : S(t) = 1\}$. The values $V(t)$ are indicated inside the nodes. (c-g) The results of the next iterations. (h) The arcs in the resulting optimal cut $\{\langle s, t \rangle \in \mathcal{A} : L(s) > L(t)\}$ with $\varepsilon_{\min}(L) = \omega(j, k) = 3$.*

Regarding the computational complexity of Algorithm 1, the nodes in Q should be organized according to an appropriate data structure of a priority queue in order to support the efficient removal of its element with minimum value of $V(t)$ from line 8. If a binary heap is used for Q, then Algorithm 1 can be implemented in $O((m + n) \log n)$, where $n = |\mathcal{N}|$, $m = |\mathcal{A}|$, and $|\cdot|$ denotes the set cardinality. Note that in this case the position of a node t in the binary heap must be updated whenever we change its value of $V(t)$ in line 14. This computational complexity can be improved to $O(m + n \times K)$, when the weights are

integers in a small interval of size K, by using bucket sorting. Note also that the special configuration of infinite-weight arcs is not a problem as they can be assigned to a special bucket.

The OIFT Algorithm 1 can also be adapted for multi-object segmentation by computing a related variant in a hierarchical layered digraph [44].

In the next section, we will show that OIFT belongs to the GGC general algorithmic framework.

2.3 OIFT as a Generalized Graph Cut algorithm

Algorithm 2. – *OIFT* Algorithm*

Input:	Image graph $\langle \mathcal{N}, \mathcal{A} \rangle$, weight maps w_0 and w_1, seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$.
Output:	The label map $L : \mathcal{N} \to \{0, 1\}$ and an array $\pi[\,]$ such that if $S(t) = 1$, then $\pi[t]$ is a path from $\mathcal{S}_{L(t)}$ to t.
Auxiliary:	An initially empty set 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***
2. ⎸ *Set* $S(t) \leftarrow 0$, $V(t) \leftarrow -\infty$, *and* $\pi[t] \leftarrow \langle t \rangle$;
3. ⎸ ***If*** $t \in \mathcal{S}_0$, ***then***
4. ⎸ ⎸ $V(t) \leftarrow \infty$, $L(t) \leftarrow 0$, *and insert* t *in* Q;
5. ⎸ ***If*** $t \in \mathcal{S}_1$ ***then***
6. ⎸ ⎸ $V(t) \leftarrow \infty$, $L(t) \leftarrow 1$, *and insert* t *in* Q.
7. ***While*** $Q \neq \emptyset$ ***do***
8. ⎸ *Remove* s *from* Q *with* $V(s) \geq V(t)$ *for all* $t \in Q$;
9. ⎸ *Set* $S(s) \leftarrow 1$;
10. ⎸ ***For each*** $\langle s, t \rangle \in \mathcal{A}$ *such that* $S(t) = 0$ ***do***
11. ⎸ ⎸ $tmp \leftarrow w_{L(s)}(s, t)$;
12. ⎸ ⎸ ***If*** $tmp > V(t)$ ***then***
13. ⎸ ⎸ ⎸ *Set* $V(t) \leftarrow tmp$, $\pi[t] \leftarrow \pi[s]\hat{}t$ *and* $L(t) \leftarrow L(s)$.
14. ⎸ ⎸ ⎸ ***If*** $t \notin Q$ ***then*** *insert* t *in* Q.

The biggest difference between the OIFT Algorithm 1 and the algorithms in the GGC framework [14] is that in the former case we *maximize* the energy function, while in the latter case we *minimize* its analog. To represent OIFT as a minimization problem it is enough to reverse in it all inequalities, exchange terms "∞" with "$-\infty$" and "minimum" with "maximum," and replace the weight function $\omega(s, t)$ with a function[2] $\bar{\omega}(s, t) := e^{-\omega(s,t)}$. Specifically, we represent OIFT as OIFT* Algorithm 2, for which we have the following result.

Proposition 2. *OIFT Algorithm 1 applied to* $\langle \mathcal{N}, \mathcal{A}, \omega \rangle$ *and the seed sets* $\mathcal{S}_0$ *and* $\mathcal{S}_1$ *can return the label map* L *if, and only if,* L *can be returned by OIFT* Algorithm 2 applied to the*

[2] In fact, we can use $h(\omega(s, t))$ in place of $e^{-\omega(s,t)}$ when h is any strictly decreasing function from $\mathbb{R}$ into $[0, \infty)$.

same graph, seed sets, and the weight functions w_0 and w_1 (on $\mathcal{A}$) defined as $w_0(s,t) = \bar{\omega}(t,s)$ and $w_1(s,t) = \bar{\omega}(s,t)$.

Now, let

$$X_L := \{\langle s,t\rangle \in \mathcal{A} : \ L(s) > L(t)\}$$

be the (standard) graph cut associated with the partition $\langle L^{-1}(1), L^{-1}(0)\rangle$ and let us define the functional[3] $F_L : \mathcal{A} \to [0,\infty)$ by putting, for every$\langle s,t\rangle \in \mathcal{A}$,

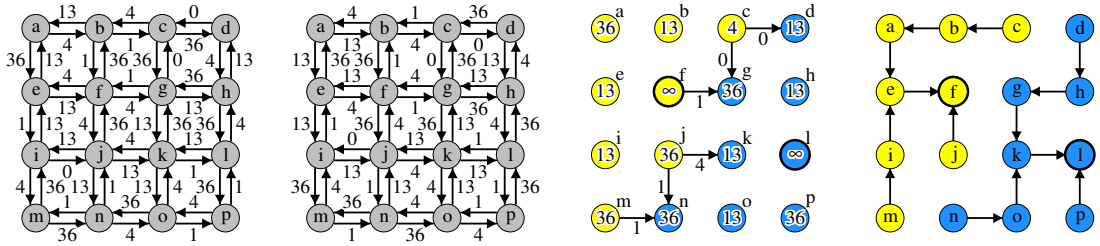
$$F_L(s,t) := \begin{cases} e^{-\omega(s,t)} & \text{for } \langle s,t\rangle \in X_L, \\ 0 & \text{otherwise.} \end{cases}$$

Then, OIFT* Algorithm 2 minimizes the energy

$$\|F_L\|_\infty := \max\{F_L(s,t) : \ \langle s,t\rangle \in \mathcal{A}\},$$

that is, the L_∞ norm of the functional F_L. This puts OIFT*, which is equivalent to OIFT, within the framework of GGC, see e.g. [14]. (Recall, that the usual graph cut minimization, associated with the max-flow/min-cut theorem, is defined as L_1 norm of the functional F_L, defined as $\|F_L\|_1 := \sum_{\langle s,t\rangle\in\mathcal{A}} F_L(s,t)$.)

A detailed example of OIFT* Algorithm 2, based on the weights $\omega(s,t)$ of Figure 2.5a, is presented in Figure 2.6.



(a) Weights $w_0(s,t)$ (b) Weights $w_1(s,t)$ (c) Resulting cut (d) Paths $\pi[t]$

Figure 2.6: *Illustration of OIFT* Algorithm 2. (a-b) The weight functions $w_0(s,t) = \bar{\omega}(t,s)$ and $w_1(s,t) = \bar{\omega}(s,t)$, using $\bar{\omega}(s,t) = 100 \cdot e^{-\omega(s,t)}$, truncated to integers for display purposes, where $\omega(s,t)$ are the weights from Figure 2.5 (a). (c) The OIFT* result using the same seeds $\mathcal{S}_0 = \{l\}$ and $\mathcal{S}_1 = \{f\}$ from Figure 2.5a, where the node colors blue and yellow indicate the labels $L(t) = 0$ and $L(t) = 1$, respectively, and the values $V(t)$ are indicated inside the nodes. The depicted arcs represent the resulting optimal cut X_L with $\|F_L\|_\infty = e^{-\omega(j,k)} \approx 0.04979$. (d) The computed paths $\pi[t]$ from $\mathcal{S}_{L(t)}$ to t represented in backwards, where the arrows indicate the predecessor of each node in the paths. For instance, we have $\pi[p] = \langle l,p\rangle$, $\pi[n] = \langle l,k,o,n\rangle$, $\pi[d] = \langle l,k,g,h,d\rangle$, $\pi[c] = \langle f,e,a,b,c\rangle$, $\pi[j] = \langle f,j\rangle$ and $\pi[m] = \langle f,e,i,m\rangle$.*

2.4 Related Shape Constraints

The importance of shape constraints in OIFT is due to the fact that its segmentation results usually present irregular boundaries, due to the lack of regularization measures,

[3] Shortly, $F_L := \bar{\omega} \cdot \chi_{X_L}$, where $\chi_{X_L} : \mathcal{A} \to \{0,1\}$ is the characteristic function of X_L.

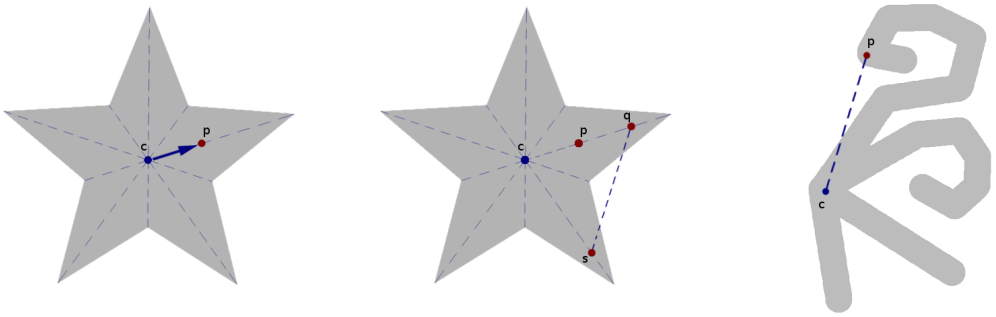
i.e. some kind of context awareness to avoid minor blemishes on the image to become full-fledged leaks, for instance. The imposition of shape constraints on the object helps to filter unwanted shapes, regularizing its boundaries.

2.4.1 Geodesic Star Constraint

The star convexity can be interpreted as a visibility constraint, that is, an object is star convex in relation to a center point c, if for every point p in the object, p is visible to c via the line segment connecting them, which must also be part of the object (the background is considered as an obstruction to the "light" emitted by p to the observer c). Note that the set of all shapes which are star-convex to a center c is a superset of the set of all convex shapes containing the point c (Figure 2.7). Furthermore, an object $\mathcal{O}$ is convex if and only if it is star-convex for any center $c \in \mathcal{O}$.

A possible extension to several star centers, considers that the points of the object must be visible in relation to at least one of the centers. However this definition is computationally intractable, being related to an NP-complete problem [37]. Thus, a computationally feasible solution alternative is to consider the argument of the shortest distance. This alternative can be defined in a discrete version in the image domain $\mathcal{I}$, where the line segments correspond to shortest paths (geodesic) in the graph derived from the image, as follows.

[Geodesic Star Convexity (GSC) [37]] A point p is said to be visible to c via a set $\mathcal{O} = \{t \in \mathcal{I} : L(t) = 1\}$ if the shortest path joining p to c lies in the set $\mathcal{O}$. An object $\mathcal{O}$ is star-convex with respect to center c, if every point $p \in \mathcal{O}$ is visible to c via $\mathcal{O}$ [93]. In the case of multiple stars in the image domain, the set of star centers $C = \{c_1, c_2, \ldots, c_n\}$ is taken to coincide with the internal seeds (i.e., $C = \mathcal{S}_1$), and the shortest paths form a spanning forest rooted at the internal seeds in the graph [37].

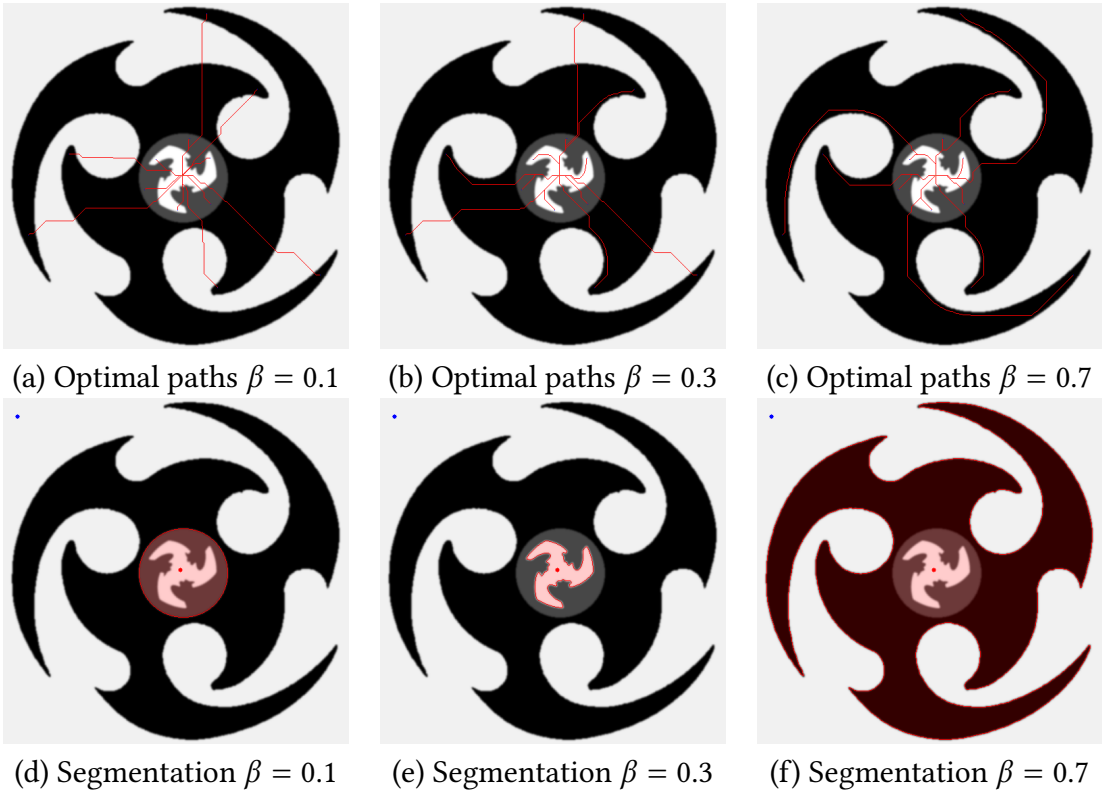


(a) Star Convex shape. (b) Non-convex shape. (c) Non-star convex shape.

Figure 2.7: *Examples of the classification of shapes according to convexity constraints. Note that in (a) and (b), we illustrate how the same shape can be star convex, but not convex in the usual definition.*

Depending on how the arc weights are defined, the paths of the spanning forest may follow the euclidean distance or may perform curves, adapting to the shapes present in the image, according to a control parameter β (Equation 2.5), as shown in Figure 2.8.

$$\omega(s,t) = [\delta(s,t) + 1]^{\beta} - 1 + \|t - s\| \tag{2.5}$$



(a) Optimal paths $\beta = 0.1$ (b) Optimal paths $\beta = 0.3$ (c) Optimal paths $\beta = 0.7$

(d) Segmentation $\beta = 0.1$ (e) Segmentation $\beta = 0.3$ (f) Segmentation $\beta = 0.7$

Figure 2.8: *Effects of the parameter β on the optimal-path forest topology and their respective segmentation results. We can see how the adaptability controlled by the β parameter influences the final segmentation result.*

2.4.2 Boundary Band Constraint

The main idea of the boundary band is to establish a maximum reference cost variation between the points of the boundary of the object to be segmented, in order to prevent the generated segmentation to be irregular in relation to the shapes of the level curves of the provided cost map.

[Boundary Band (BB) [5]] For a given labeled segmentation, consider the set of object pixels $\mathcal{O} = \{t \in \mathcal{I} : L(t) = 1\}$, and the set of pixels in its boundary $\text{bd}(\mathcal{O}) = \{t \in \mathcal{O} : \exists s \in \mathcal{A}(t)$ such that $s \notin \mathcal{O}\}$. An object $\mathcal{O}$ satisfies the Boundary Band Constraint with band size Δ and reference cost map $C : \mathcal{I} \to [0, \infty)$ if $\forall t \in \mathcal{O}$ we have $C(t) < \min_{s \in \text{bd}(\mathcal{O})} C(s) + \Delta$. Note that it restricts all boundary pixels in $\text{bd}(\mathcal{O})$ to be inside a band $B(\mathcal{O}) = \{b \in \mathcal{I} : C(b) \in [m, m + \Delta)\}$, where $m = \min\{C(s) : s \in \text{bd}(\mathcal{O})\}$.

The boundary band restricts the pixels in $\text{bd}(\mathcal{O})$ so that they are within a band $B(\mathcal{O})$ (i.e., $\text{bd}(\mathcal{O}) \subseteq B(\mathcal{O})$) of fixed width Δ, preventing the boundary pixels found from having cost variations greater than this value, therefore regularizing its shape [5].

In the context of OIFT-based segmentation, during the label propagation from the seeds, the growing object $\mathcal{O}'$ changes the position of its band $B(\mathcal{O}')$, allowing its adaptation to the image content (Figure 2.9) [4]. Note that only its position is changed while its width Δ

[4] During the computation of Algorithm 1, the growing object is defined as $\mathcal{O}' = \{t \in \mathcal{N} : S(t) = 1 \wedge L(t) = 1\}$.

is kept fixed.

Regarding the reference cost map C, one option is to take the costs of the shortest paths from the internal seeds as C, by computing one IFT with an additive path-cost function (Eq. 2.1) of the spatial distance of neighboring pixels, similar to the shortest-path forest used in the Geodesic Star Convexity, so that the band size becomes a spatial size constraint. Other options based on templates of shapes are discussed in Section 3.2.5 (see also reference [6]) and will be considered for evaluation on Section 3.3.1.

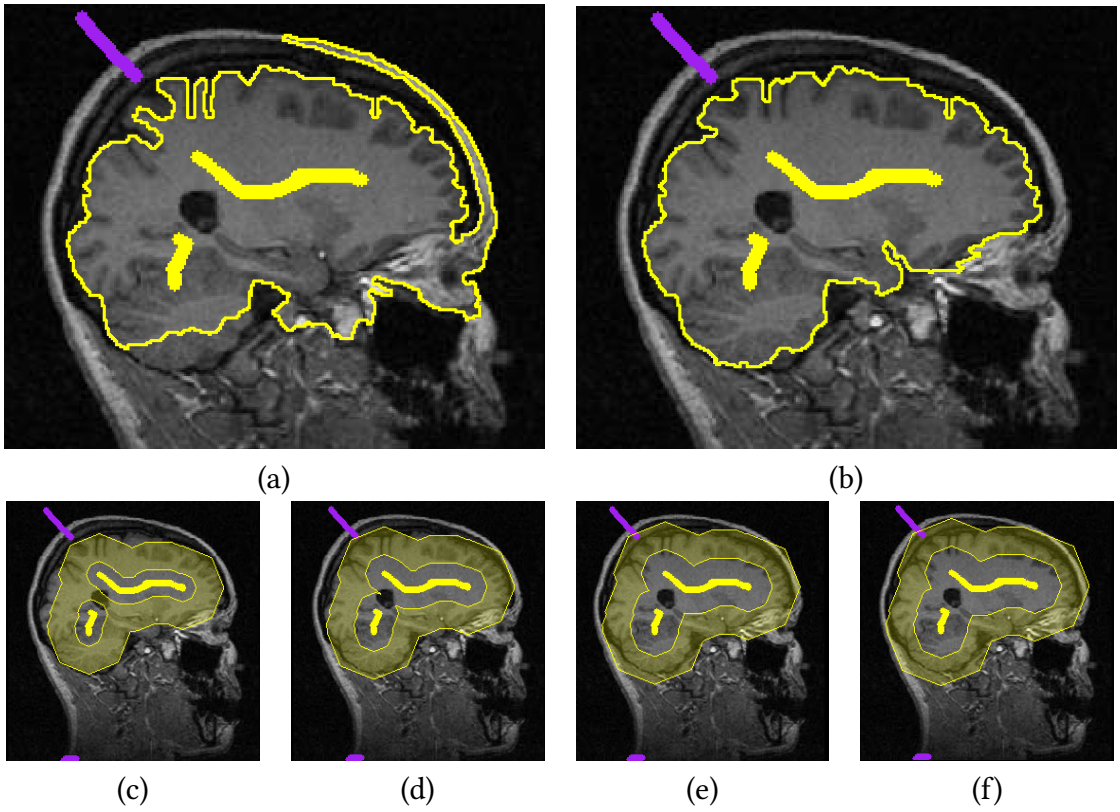


Figure 2.9: *Brain segmentation example in MRI exam. (a) and (b): Segmentation results by OIFT without and with the BB constraint, respectively. (c) to (f): The BB fixed size band evolves from the seeds, adapting to the image contents. Note that the segmentation boundary achieved in (b) resides within the band area in (f).*

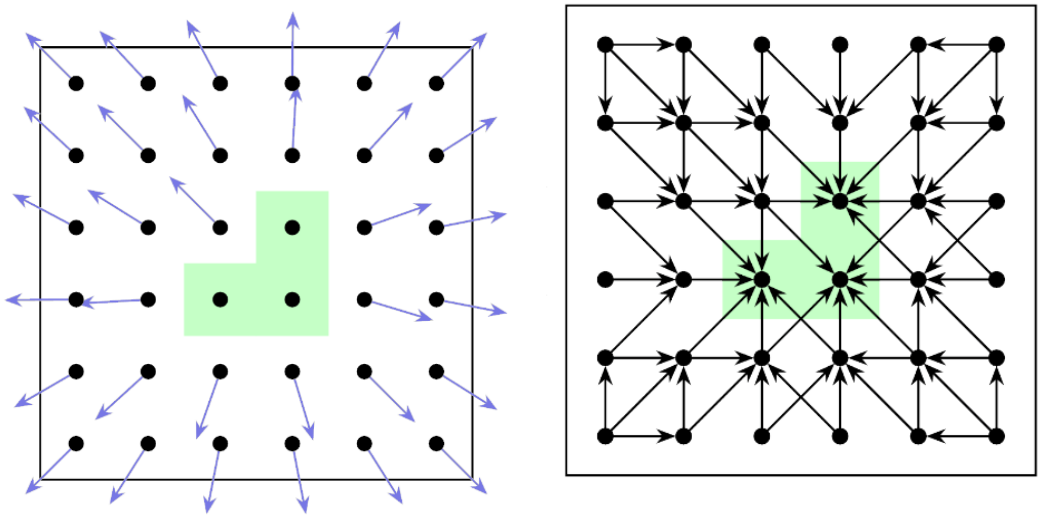
2.4.3 Hedgehog Shape Prior

In a nutshell, hedgehog shape prior for a given object uses a vector field (Figure 2.10a) to restrict the object's surface normals by setting or creating new infinity weighted arcs $\langle p, q \rangle$, in a circular sector centered at p, resulting in a final graph with additional arcs defined by an translation-variant adjacency relation (Figure 2.10b). When the vector field is taken as the gradient of the distance transform from the internal seeds, we obtain a segmentation that better aligns with the levelsets of the distance transform, which is beneficial in case of clutter and noise.

[Hedgehog Shape Prior (HSP) [41, 40]] Each hedgehog shape has its surface normals locally constrained by an arbitrary given vector field (e.g., gradient of the distance

transform from internal seeds) and angular threshold θ. This hedgehog constraint is obtained by setting the worst cutting cost to all arcs adjacent to a pixel p whose directions agree with polar cone $\hat{C}_\theta(p) = \left\{\langle p, y\rangle : [p, y] \cdot \vec{z} \leqslant 0 \ \forall \vec{z} \in C_\theta(p)\right\}$, that is, the arcs in $H(\theta) = \left\{\langle p, q\rangle \in \mathcal{A} : \langle p, q\rangle \in \hat{C}_\theta(p) \text{ or } \langle p, q\rangle \in -\hat{C}_\theta(q)\right\}$, where $C_\theta(p)$ is the cone of allowed surface normals at some point p (Figure 2.11).

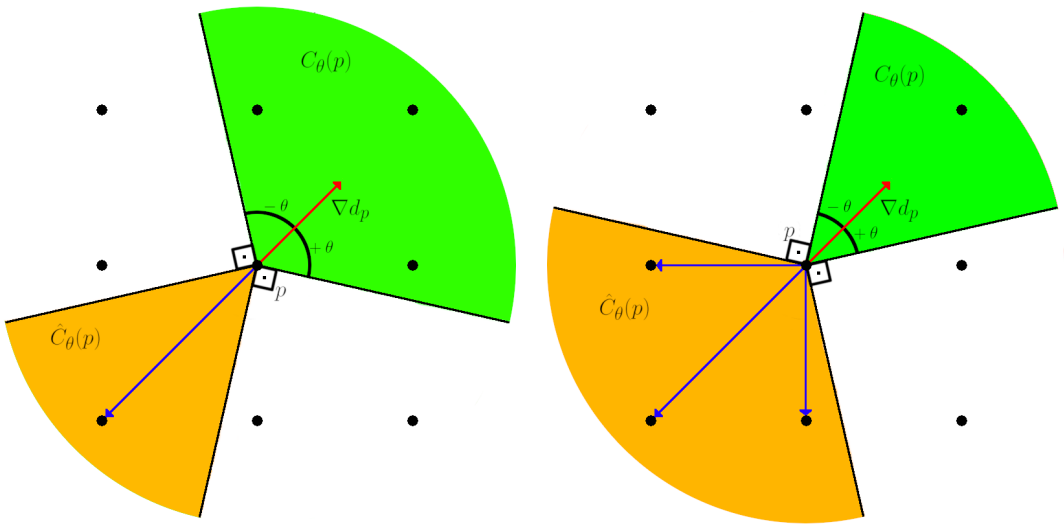
Hedgehog constraints are better approximated by large neighbor systems. We will use $\mathcal{A}_R$ to denote the circular adjacency relation used to define the circular sectors. $\mathcal{A}_R$ is composed of all pairs of pixels $\langle s, t\rangle$ in the Cartesian product $\mathcal{I} \times \mathcal{I}$ such that $\|s - t\| \leqslant R$, $s \neq t$ and R is a specified constant.



(a) gradients ∇d (b) graph arcs $H(\frac{\pi}{4})$

Figure 2.10: *Example of the Hedgehog constraint. (a) The internal seeds (green) and the corresponding distance map gradients ∇d (vector field). (b) The infinity cost arcs in $H(\theta)$ selected as in Figure 2.11.*

The usage of a different vector field other than the gradient of the distance transform leads to a different set of allowed shapes by the hedgehog prior. For example, the gradient of the shape templates used as reference cost map for the BB constraint could be used as vector field, as discussed in Section 3.2.5.



(a) Wide cone $C_\theta(p)$ (b) Tight cone $C_\theta(p)$

Figure 2.11: *Approximating hedgehog constraint at point p. Cone $C_\theta(p)$ of allowed normals (green) is enforced by forbidding the arcs $\langle p, q \rangle$, for q inside the $\hat{C}_\theta(p)$ cone (orange).*

Chapter 3

Contributions

3.1 OIFT within General Fuzzy Connectedness framework

In the previous chapter, we have seen that OIFT Algorithm 1 belongs to the GGC framework. Here, we will argue that it can be also viewed as belonging to a class of General Fuzzy Connectedness, GFC, algorithms [17]. This will allow us to deduce that OIFT has the properties that all algorithms in GFC are known to have.

In what follows, for a fixed digraph $\langle \mathcal{N}, \mathcal{A} \rangle$, weight maps w_0 and w_1, and the seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$, define the path costs:

$$\begin{aligned} \psi_{\min}(\langle v_0, \ldots, v_\ell \rangle) &:= \min_{1 \le j \le \ell} w_{L(v_0)}(v_{j-1}, v_j) \\ \psi_{\text{last}}(\langle v_0, \ldots, v_\ell \rangle) &:= w_{L(v_0)}(v_{\ell-1}, v_\ell) \end{aligned}$$

for $\ell > 0$ and

$$\psi_{\text{last}}(\langle v_0 \rangle) := \psi_{\min}(\langle v_0 \rangle) := \begin{cases} \infty & \text{for } v_0 \in \mathcal{S}_0 \cup \mathcal{S}_1, \\ -\infty & \text{otherwise.} \end{cases}$$

The map $\psi_{\min}$ is the standard FC cost [16], while ψ_{last}, explicitly defined in [58] (using symbols $f_{i,\omega}$ and $f_{o,\omega}$), is naturally associated with OIFT.

To place OIFT in the GFC framework, we will first represent OIFT* of Algorithm 2 as the MOFS* Algorithm 3, which is a version of MOFS algorithm from [17]. The key result here is the following theorem, which is considerably less clear than Proposition 2, since the conditions in lines 8 and 12 of the algorithms have different forms.

Theorem 3 (OIFT* in GFC format)**.** *Any output of OIFT* Algorithm 2 is identical to that of MOFS* Algorithm 3. In particular, the algorithms MOFS* and OIFT are equivalent.*

Algorithm 3. – *MOFS* Algorithm*

Input: Image graph $\langle \mathcal{N}, \mathcal{A} \rangle$, affinities w_0 and w_1, seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$.
Output: The label map $L : \mathcal{N} \to \{0, 1\}$ and an array $\pi[\,]$ such that if $S(t) = 1$, then $\pi[t]$ is a path from $\mathcal{S}_{L(t)}$ to t.
Auxiliary: An initially empty set 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***
2. $\quad$ *Set* $S(t) \leftarrow 0$, $V(t) \leftarrow -\infty$, *and* $\pi[t] \leftarrow \langle t \rangle$;
3. $\quad$ ***If*** $t \in \mathcal{S}_0$, ***then***
4. $\quad\quad$ $V(t) \leftarrow \infty$, $L(t) \leftarrow 0$, *and insert* t *in* Q;
5. $\quad$ ***If*** $t \in \mathcal{S}_1$ ***then***
6. $\quad\quad$ $V(t) \leftarrow \infty$, $L(t) \leftarrow 1$, *and insert* t *in* Q.
7. ***While*** $Q \neq \emptyset$ ***do***
8. $\quad$ *Remove from* Q *an* s *in* $M = \{u \in Q : \psi_{\min}(\pi[u]) = \max_{t \in Q} \psi_{\min}(\pi[t])\}$ *such that* $V(s) \geq V(u)$ *for all* $u \in M$;
9. $\quad$ *Set* $S(s) \leftarrow 1$;
10. $\quad$ ***For each*** $\langle s, t \rangle \in \mathcal{A}$ *such that* $S(t) = 0$ ***do***
11. $\quad\quad$ $tmp \leftarrow w_{L(s)}(s, t)$;
12. $\quad\quad$ ***If*** $\psi_{\min}(\pi[s]\hat{}t) > \psi_{\min}(\pi[t])$ *or* $[\psi_{\min}(\pi[s]\hat{}t) = \psi_{\min}(\pi[t])$ *and* $tmp > V(t)]$ ***then***
13. $\quad\quad\quad$ *Set* $V(t) \leftarrow tmp$, $\pi[t] \leftarrow \pi[s]\hat{}t$ *and* $L(t) \leftarrow L(s)$;
14. $\quad\quad\quad$ ***If*** $t \notin Q$ ***then*** *insert* t *in* Q.

We will postpone the proof of Theorem 3 to the end of this section.

Notice, that although OIFT* Algorithm 2 has a format of the MOFS algorithm from the GFC framework, it is not precisely of this format. The first difference is that the main GFC algorithm MOFS, when it removes a vertex s from Q, does not have the secondary condition "$V(s) \geq V(u)$ for all $u \in M$" as we have in line 8. But this just means, that in MOFS* we are just a bit more precise, when making such choice.

The bigger difference is that MOFS allows some overlap of the object and background. Specifically, they overlap on the tie zone set TZ defined as the set of all $v \in \mathcal{N}$ for which MOFS, whose output is unique, produces the paths of the same strength from the object and the background. The issue of how to deal with the set TZ is discussed in details in [17]. In particular, if $w_0(s, t) \neq w_1(u, v)$ for all edges $\langle s, t \rangle$ and $\langle u, v \rangle$, then TZ is empty and the object returned by MOFS* (or OIFT*) is identical to that of MOFS output. Other solutions of the "overlapping problem" are also discussed in [17]. The reader should be warned, however, that a simple minded removal of TZ from the MOSF object (with overlap) may create a set with vertices that are not connected, within the object, to the seeds.

3.1.1 Proof of Theorem 3

First notice that, during the execution of OIFT* Algorithm 2, for any $u \in Q$ either u is a seed or $\pi[u] = \pi[w]\hat{}u$ for some $w \in \mathcal{N}$ with $S(w) = 1$.

To prove the theorem, it is enough to show that during the execution of OIFT* Algorithm 2, the condition from line 8 holds, if and only if, the condition from line 8 of MOFS*

Algorithm 3 holds. Similarly, for the conditions from line 12.

To see this, we will prove that, at any time of the execution of OIFT* Algorithm 2 past the line 6 of the code, the following holds for every $u, v \in \mathcal{N}$:

(i) if $S(u) = 1$ and $S(v) = 0$, then $\psi_{\min}(\pi[u]) \geq \psi_{\min}(\pi[v])$;

(ii) if $S(u) = S(v) = 0$ and $V(u) \geq V(v)$, then $\psi_{\min}(\pi[u]) \geq \psi_{\min}(\pi[v])$.

After the for loop on lines 1-6, this holds directly by design, thus it is enough to show that these properties are preserved by any consecutive single execution of the while loop, that is, of lines 7-14.

So, fix $u, v \in \mathcal{N}$ for which we will be showing preservation of (i) and (ii). If u is a seed, then after the initialization we have $\psi_{\min}(\pi[u]) = V(u) = \infty$, so (i) and (ii) hold. So, we will assume that u is not a seed. Next, assume that during our execution of lines 8-14 we have taken s from Q.

To see that (ii) is preserved, assume that, after the execution of lines 8-14, we have $S(u) = S(v) = 0$. During the execution, the values of either $V(v)$ or $\pi[v]$ can change only in line 13, when $v = t$ for t chosen in line 10 and, during the execution of line 12, we have $w_{L(s)}(s, v) = w_{L(s)}(s, t) = tmp > V(t)$. Hence, the execution of line 13 results with $V(v) = V(t)$ becoming $w_{L(s)}(s, v)$ and $\pi[v] = \pi[t]$ becoming $\pi[s]\hat{}v = \pi[s]\hat{}t$ so that

$$\psi_{\min}(\pi[v]) = \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s, v)\}.$$

The similar analysis holds when either of the values $V(u)$ or $\pi[u]$ are changed during the execution of lines 8-14.

Now, consider 4 cases:

- If none of the values $V(v)$, $\pi[v]$, $V(u)$, or $\pi[u]$ changes during the execution of lines 8-14, then clearly (ii) is preserved.
- If, during the execution, we applied the changes in line 13 to both u and v, then $V(u) \geq V(v)$ implies that $w_{L(s)}(s, u) \geq w_{L(s)}(s, v)$ and so,

$$\begin{aligned} \psi_{\min}(\pi[u]) &= \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s, u)\} \\ &\geq \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s, v)\} \\ &= \psi_{\min}(\pi[v]) \end{aligned}$$

  giving desired (ii).
- If, during the execution, we applied the changes in line 13 only to u, then $\psi_{\min}(\pi[u]) = \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s, u)\} \geq \psi_{\min}(\pi[s]) \geq \psi_{\min}(\pi[v])$, where the last inequality is implied by $V(s) \geq V(v)$, ensured by the choice of s from Q, and the recursive assumption (ii). Thus, indeed (ii) is preserved.
- Finally, if, during the execution, we applied the changes in line 13 only to v, then $V_{\text{old}}(u) = V_{\text{new}}(u) \geq V_{\text{new}}(v) = w_{L(s)}(s, v)$. Also, since u is not a seed, we have $\pi[u] = \pi[w]\hat{}u$ for some $w \in \mathcal{N}$ with $S(w) = 1$. By (i), used just before we have taken s from

Q, we have $\psi_{\min}(\pi[w]) \geq \psi_{\min}(\pi[s])$. Therefore,

$$\begin{aligned} \psi_{\min}(\pi[u]) &= \min\{\psi_{\min}(\pi[w]), V(u)\} \\ &\geq \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s,v)\} \\ &= \psi_{\min}(\pi[v]) \end{aligned}$$

finishing the proof of preservation of (ii).

Next, we will prove preservation of (i). So, assume that after the execution of lines 8-14, we have $S(u) = 1$ and $S(v) = 0$. Then, by (i), used just before we have taken s from Q, we have $\psi_{\min}(\pi[u]) \geq \psi_{\min}(\pi[s])$. Thus, it is enough to show that, right after the execution of lines 8-14, we have $\psi_{\min}(\pi[s]) \geq \psi_{\min}(\pi[v])$. This clearly holds if the values $V(v)$ or $\pi[v]$ were not changed. So, assume that they have been changed. Then, as before, we see that

$$\psi_{\min}(\pi[s]) \geq \min\{\psi_{\min}(\pi[s]), w_{L(s)}(s,v)\} = \psi_{\min}(\pi[v])$$

finishing the proof of preservation of (i) and of the theorem.

3.2 The Local Band constraint (LB)

Let $C : \mathcal{N} \to [0,\infty)$ be a fixed vertex cost function associated with an image digraph $G = \langle \mathcal{N}, \mathcal{A} \rangle$. Usually $C(t)$ is defined as a minimum of all possible path cost functions for the paths from $\mathcal{S}_1$ to t. The path cost can be its geodesic length (i.e., $\psi_{sum}(\langle v_0, \ldots, v_\ell \rangle) := \sum_{1 \leq j \leq \ell} \|v_{j-1} - v_j\|$), as used in Geodesic Star Convexity, but other path costs are also useful. It can also be based on templates of shapes as discussed in Section 3.2.5, which will also be considered for evaluation in Section 3.3.

The goal of this section is to construct an extension of an edge-weighted digraph $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$, discussed above to the edge-weighted digraph $G' = (\mathcal{N}, \mathcal{A}', \omega')$ so that the application of OIFT (Algorithm 1) to G' produces an optimized object satisfying the Local Band constraint defined below.

To relate Local Band constraint to Boundary Band constraint introduced in [5], we first introduce the following notion of *Local Boundary Band constraint,* LBB. In this definition, the symbol $\|\cdot\|$ denotes the standard Euclidean L_2 norm on $\mathcal{N} \subset \mathbb{Z}^2$. The *boundary* of an object $\mathcal{O}$ is defined as

$$\mathrm{bd}(\mathcal{O}) = \{t \in \mathcal{O} : \exists s \in \mathcal{A}(t) \text{ such that } s \notin \mathcal{O}\}.$$

Definition 1 (Local Boundary Band (LBB))**.** *For $\Delta, R > 0$ and a cost map $C : \mathcal{N} \to [0,\infty)$, a pixel $t \in \mathcal{O}$ is LBB^R_Δ (satisfies Local Boundary Band Constraint with band size Δ and parameter R) provided $C(t) < C(s) + \Delta$ for all $s \in \mathrm{bd}(\mathcal{O})$ such that $\|s - t\| \leq R$. An object $\mathcal{O}$ is LBB^R_Δ provided every $t \in \mathcal{O}$ is LBB^R_Δ.*

Definition 2 (Boundary Band constraint (BB))**.** *For $\Delta > 0$, an object $\mathcal{O}$ is BB_Δ (satisfies Boundary Band constraint with band size Δ) provided it is $\mathrm{LBB}^\infty_\Delta$, that is, when $C(t) < C(s) + \Delta$ for all $t \in \mathcal{O}$ and $s \in \mathrm{bd}(\mathcal{O})$.*

Note that Definition 2 is equivalent to the definition given in Section 2.4.2.

As a consequence of Definition 2, we have that $\mathrm{bd}(\mathcal{O})$ is contained in the band $\{s \in \mathcal{N} : C(s) \in (m-\Delta, m]\}$, where $m = \max\{C(t) : t \in \mathcal{O}\}$. In particular, we have $|C(s)-C(t)| < \Delta$ for all $s, t \in \mathrm{bd}(\mathcal{O})$. Consequently, this regularizes the shape of $\mathrm{bd}(\mathcal{O})$, see [5]. Therefore, the idea of BB is to establish a maximum possible variation of the cost C between the boundary points $\mathrm{bd}(\mathcal{O})$ of the object $\mathcal{O}$ to be segmented. This is expected to prevent the generated segmentation to be irregular in relation to the C-level sets [5]. During the OIFT computation subject to BB, the band changes its reference level set, allowing a better adaptation to the image content, while its width is kept fixed (Figure 2.9). Note that this bears some resemblance to narrow band level set [77] and to the regional context of a level line used in [100].

In BB, however, local changes in a part of the object can generate constraint violations in any other part of its boundary, usually resulting in greater sensitivity to the initialization of the cost map C and to the positioning of internal seeds, while in LBB its consistency checks are limited locally, leading to a more flexible solution. Clearly, every BB_Δ object is LBB^R_Δ, but the converse is not true. Nevertheless, for every C and Δ, there exists an $R \in (0, \infty)$ such that the property LBB^R_Δ implies BB_Δ (this certainly holds for any $R \geq \max\{\|s-t\| : s, t \in \mathcal{N}\}$). Thus, BB_Δ can be considered as a limit, as $R \to \infty$, of LBB^R_Δ.

In order to facilitate the implementation, we consider an approximate alternative definition, named the *Local Band constraint* (LB), in order to avoid the continuous analysis of the dynamic set of boundary pixels inside the disks of radius R at runtime, but keeping the main idea of locally restricting the band effects. This effort resulted in the following similar definition.

Definition 3 (Local Band constraint (LB))**.** *For $\Delta, R > 0$ and a cost map $C : \mathcal{N} \to [0, \infty)$, a pixel $t \in \mathcal{O}$ is LB^R_Δ (satisfies Local Band constraint with band size Δ and parameter R) provided $C(t) < C(s) + \Delta$ for all $s \in \mathcal{N} \setminus \mathcal{O}$ such that $\|s - t\| \leq R$. An object $\mathcal{O}$ is LB^R_Δ provided every $t \in \mathcal{O}$ is LB^R_Δ.*

In other words, if $\mathcal{O}$ is LB^R_Δ, then for any pair of pixels s and t such that $\|s - t\| \leq R$ and $C(t) - C(s) \geq \Delta$, we have that $t \in \mathcal{O}$ implies $s \in \mathcal{O}$. Note that neither of the statements "$\mathcal{O}$ is LB^R_Δ" and "$\mathcal{O}$ is LBB^R_Δ" implies the other. Nevertheless, they are closely related, as shown in the next subsection.

Similar to establishing BB_Δ as the limiting case of LBB^R_Δ, as $R \to \infty$, we can also consider the limiting case of LB^R_Δ as $R \to \infty$, which results in the following definition.

Definition 4 (Band constraint (B))**.** *For $\Delta > 0$, an object $\mathcal{O}$ is B_Δ (satisfies Band constraint with band size Δ) provided it is $\mathrm{LB}^\infty_\Delta$, that is, when $C(t) < C(s) + \Delta$ for all $t \in \mathcal{O}$ and $s \in \mathcal{N} \setminus \mathcal{O}$.*

3.2.1 Theoretical relationships with other shape constraints

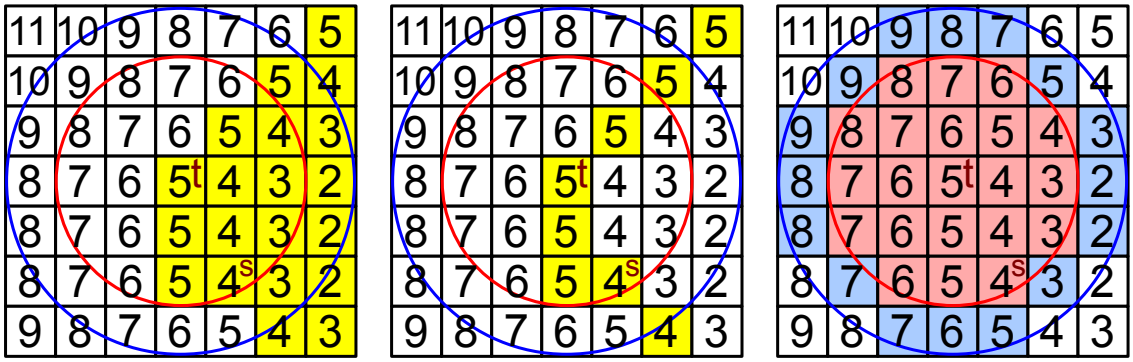
In this section we discuss the theoretical relationships between Local Band constraint and other shape constraints. Consider the following proposition:

Proposition 4. *Let $r = \max_{\langle s,t\rangle \in \mathcal{A}} \|s - t\|$ and $\delta = \max_{\langle s,t\rangle \in \mathcal{A}} |C(t) - C(s)|$. If $\Delta, R > 0$ and $\mathcal{O}$ is LB^{R+r}_Δ, then $\mathcal{O}$ is $\mathrm{LBB}^R_{\Delta+\delta}$.*

Proof. Choose a $t \in \mathcal{O}$. Then $C(t) < C(s) + \Delta$ for all $s \in \mathcal{N} \setminus \mathcal{O}$ such that $\|s - t\| \leq R + r$. We need to show that t is $\mathrm{LBB}^R_{\Delta+\delta}$, that is, that $C(t) < C(u) + \Delta + \delta$ for all $u \in \mathrm{bd}(\mathcal{O})$ such that $\|u - t\| \leq R$. So, take such u. Then, there is an $s \in \mathcal{N} \setminus \mathcal{O}$ with $\langle u, s \rangle \in \mathcal{A}$. Notice that $\|s - t\| \leq \|s - u\| + \|u - t\| \leq r + R$. Using this and the definition of δ, we get $C(t) < C(s) + \Delta \leq C(u) + \Delta + |C(s) - C(u)| \leq C(u) + \Delta + \delta$, as needed. □

Figure 3.1 illustrates Proposition 4, showing a pixel t that is both $\mathrm{LB}^{R+r}_{\Delta}$ and $\mathrm{LBB}^R_{\Delta+\delta}$.

Since usually numbers δ and r are small, so should be the difference between the objects with properties LB^R_{Δ}, $\mathrm{LB}^{R+r}_{\Delta}$, $\mathrm{LBB}^R_{\Delta+\delta}$, or LBB^R_{Δ} and, for large R, each approximates BB_{Δ}.



(a) $\mathcal{O}$ (yellow) (b) $\mathrm{bd}(\mathcal{O})$ (yellow) (c) Disks R & $R + r$.

Figure 3.1: *Example of Proposition 4, where "t is* $\mathrm{LB}^{R+r}_{\Delta}$*" and "t is* $\mathrm{LBB}^R_{\Delta+\delta}$*" for* $R = 2.5$, $r = 1.0$, $\Delta = 1$ *and* $\delta = 1$. *(a)* $\mathcal{O}$, *(b)* $\mathrm{bd}(\mathcal{O})$, *and (c) the disks of radii* R *and* $R + r$.

The following additional theoretical relationships can be established for these shape constraints.

Properties For $\Delta, \delta, R, r > 0$, we have:

(*P*1) If $\delta \leq \Delta$, then $\mathcal{O}$ is $\mathrm{LBB}^R_{\delta} \implies \mathcal{O}$ is LBB^R_{Δ},

(*P*2) If $r \leq R$, then $\mathcal{O}$ is $\mathrm{LBB}^R_{\Delta} \implies \mathcal{O}$ is LBB^r_{Δ},

(*P*3) If $\delta \leq \Delta$, then $\mathcal{O}$ is $\mathrm{LB}^R_{\delta} \implies \mathcal{O}$ is LB^R_{Δ},

(*P*4) If $r \leq R$, then $\mathcal{O}$ is $\mathrm{LB}^R_{\Delta} \implies \mathcal{O}$ is LB^r_{Δ}.

Proof. These properties follow immediately from the shape constraint definitions.

For P1, $\mathcal{O}$ is LBB^R_{δ} requires $C(t) < C(s) + \delta$ in Definition 1, which implies that $C(t) < C(s) + \delta \leq C(s) + \Delta$, so consequently we have that $\mathcal{O}$ is LBB^R_{Δ}.

For P2, $\mathcal{O}$ is LBB^R_{Δ} requires $C(t) < C(s) + \Delta$ for all $s \in \mathrm{bd}(\mathcal{O})$ such that $\|s - t\| \leq R$ in Definition 1, so it is also satisfied for $\|s - t\| \leq r$ and consequently we have that $\mathcal{O}$ is LBB^r_{Δ}.

For P3, $\mathcal{O}$ is LB^R_δ requires $C(t) < C(s) + \delta$ in Definition 3, which implies that $C(t) < C(s) + \delta \leq C(s) + \Delta$, so consequently we have that $\mathcal{O}$ is LB^R_Δ.

For P4, $\mathcal{O}$ is LB^R_Δ requires $C(t) < C(s) + \Delta$ for all $s \in \mathcal{N} \setminus \mathcal{O}$ such that $\|s - t\| \leq R$ in Definition 3, so it is also satisfied for $\|s - t\| \leq r$ and consequently we have that $\mathcal{O}$ is LB^r_Δ. □

For small/appropriate values of r and δ, the LBB and LB constraints are closely related according to Proposition 4. Note however that the converse of this statement may not be true. The relationship in the opposite direction normally requires extra assumptions on map C and $\mathcal{O}$, as exemplified in the proposition below.

Proposition 5. *Let $\mathcal{O}$ be a star-convex object with respect to center c and $C(t) = \|t - c\|$ be the Euclidean distance from c. If $\Delta, R > 0$ and $\mathcal{O}$ is LBB^R_Δ, then $\mathcal{O}$ is LB^R_Δ.*

Proof. Let $B(t) = \{s \in \mathcal{N} : \|t - s\| \leq R\}$ be the closed ball centered at t with radius R. Consider a boundary pixel $b \in \mathrm{bd}(\mathcal{O})$ inside $B(t)$, such that $C(b) = \|b - c\| = d_{\min}$, where $d_{\min} = \min_{x \in \mathrm{bd}(\mathcal{O}) \cap B(t)} C(x)$, as indicated in Figure 3.2a. Since $\mathcal{O}$ is LBB^R_Δ, we know that $C(t) < C(b) + \Delta = d_{\min} + \Delta$. To prove that $\mathcal{O}$ is LB^R_Δ, we must show that the following condition holds:

(∗) $C(u) \geq d_{\min}$ for all $u \in (\mathcal{N} \setminus \mathcal{O}) \cap B(t)$.

Note that this condition given by (∗) implies that $C(t) < d_{\min} + \Delta \leq C(u) + \Delta$.

For each $u \in (\mathcal{N} \setminus \mathcal{O}) \cap B(t)$, we have two cases to consider:

1. The line segment $\overline{uc}$ with endpoints u and c intercepts a pixel $p \in \mathrm{bd}(\mathcal{O}) \cap B(t)$ (Figure 3.2a). In this case, we have $C(p) < C(u)$. By the definition of $d_{\min}$, we know that $d_{\min} \leq C(p)$. Hence we have that $d_{\min} \leq C(p) < C(u)$.
2. The line segment $\overline{uc}$ with endpoints u and c intercepts a pixel $p \in \mathrm{bd}(B(t))$ and $p \notin \mathcal{O}$ (Figure 3.2b). In this case, moving from p through the circular arc $\overset{\frown}{pr}$ towards the point r, which is the ball's pixel closest to c, there will always be a point $q \in \mathrm{bd}(\mathcal{O})$ and $C(q) \leq C(p) \leq C(u)$. Note that this is true since $\mathcal{O}$ is assumed to be a star-convex object with respect to center c, so that the segment $\overline{tc}$ is always contained within the object and $r \in \mathcal{O}$. By the definition of $d_{\min}$, we know that $d_{\min} \leq C(q)$. Hence we have that $d_{\min} \leq C(q) \leq C(u)$.

□

Although Proposition 5 requires a star-convex object with respect to center c and $C(t) = \|t - c\|$, in practice, we have close results by similar arguments when we consider the Geodesic Star Convexity [57] and the cost map C is the geodesic length [16] (i.e., $\psi_{sum}(\langle v_0, \ldots, v_\ell \rangle) := \sum_{1 \leq j \leq \ell} \|v_{j-1} - v_j\|$), from a compact and connected set of seeds $\mathcal{S}_1$ in $G = (\mathcal{N}, \mathcal{A})$.

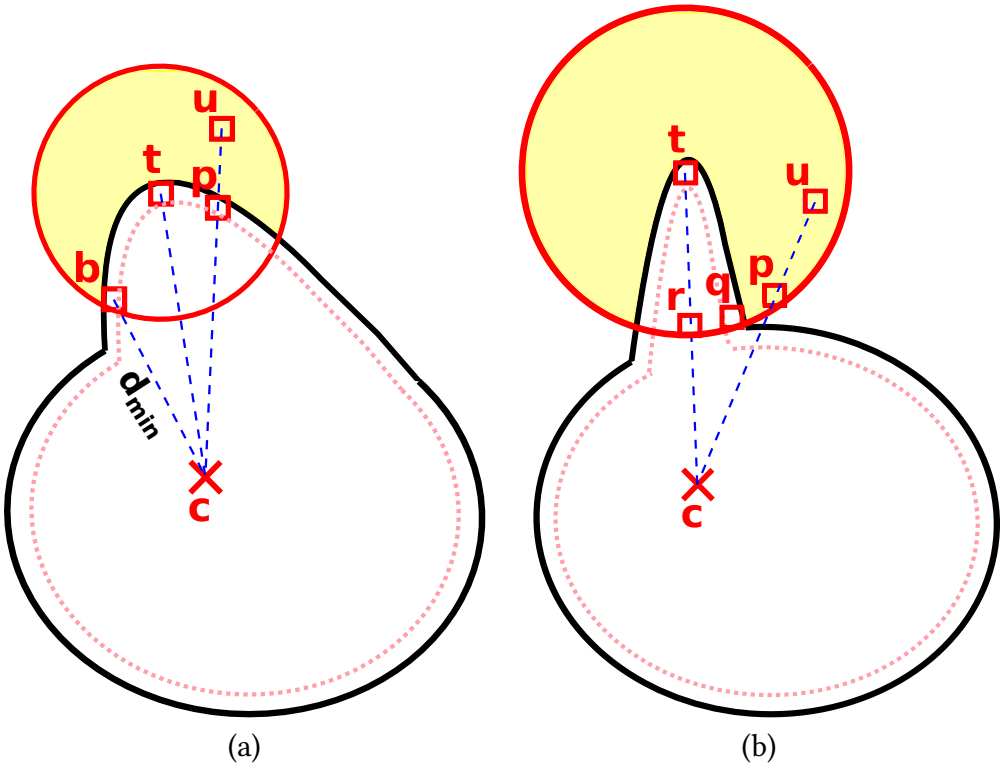


Figure 3.2: *Schematic illustration of Proposition 5. (a) The line segment $\overline{uc}$ with endpoints u and c intercepts a pixel $p \in \mathrm{bd}(\mathcal{O}) \cap B(t)$. (b) The line segment $\overline{uc}$ with endpoints u and c intercepts a pixel $p \in \mathrm{bd}(B(t))$ and $p \notin \mathcal{O}$.*

Now let's consider the limiting cases of Propositions 4 and 5, as $R \to \infty$. For Proposition 4, we have that if $\Delta > 0$ and $\mathcal{O}$ is B_Δ, then $\mathcal{O}$ is $\mathrm{BB}_{\Delta+\delta}$. Under the assumptions of Proposition 5, we have that $\mathcal{O}$ is $\mathrm{BB}_\Delta \implies \mathcal{O}$ is B_Δ. Note also that the limiting case of Property (P1), as $R \to \infty$, implies that $\mathcal{O}$ is $\mathrm{BB}_\Delta \implies \mathcal{O}$ is $\mathrm{BB}_{\Delta+\delta}$. Therefore the Boundary Band constraint (BB) and the Band constraint (B) are strongly correlated.

3.2.2 LB Algorithm and its proof of correctness

The LB constraint can be implemented, as proposed in Algorithm 4 for OIFT, by considering a modified graph G' with the LB constraint embedded in its arcs. In general, the worst cost should be ∞ for Min-Sum optimizers (i.e., min-cut/max-flow algorithm) and $-\infty$ for Max-Min optimizers. In order to maintain a symmetric graph, we also create anti-parallel arcs with the best cutting cost (zero for Min-Sum and ∞ for Max-Min optimizers) if they do not exist (line 5 in Algorithm 4). Note that in G' the set of displacement vectors $D(s) = \{t - s : t \in \mathcal{A}'(s)\}$ varies for different positions of s, leading therefore to a translation-variant adjacency relation.

Algorithm 4. – *Segmentation by OIFT subject to the* LB *constraint*

Input: Symmetric edge-weighted image digraph $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$, non-empty disjoint seed sets $\mathcal{S}_1$ and $\mathcal{S}_0$, cost map $C : \mathcal{N} \to [0, \infty)$, and parameters $R > 0$ and $\Delta > 0$.
Output: The label map $L : \mathcal{N} \to \{0, 1\}$.
Auxiliary: Edge-weighted digraph $G' = \langle \mathcal{N}, \mathcal{A}', \omega' \rangle$ with $\mathcal{A} \subset \mathcal{A}'$.

1. *Set* $\mathcal{A}' \leftarrow \mathcal{A}$ *and* $\omega' \leftarrow \omega$.
2. ***For each*** $\langle s, t \rangle \in \{\langle p, q \rangle \in \mathcal{N} \times \mathcal{N} : \|p - q\| \leq R \;\&\; C(p) \geq C(q) + \Delta\}$ ***do***
3. ⎸ ***If*** $\langle s, t \rangle \notin \mathcal{A}'$ ***then*** *Set* $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{\langle s, t \rangle\}$ *and define* $\omega'(s, t) := -\infty$.
4. ⎸ ***Else*** *Redefine* $\omega'(s, t) := -\infty$.
5. ⎸ ***If*** $\langle t, s \rangle \notin \mathcal{A}'$ ***then*** *Set* $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{\langle t, s \rangle\}$ *and define* $\omega'(t, s) := \infty$.
6. *Compute, by Algorithm 1,* $L : \mathcal{N} \to \{0, 1\}$ *for* G' *and seed sets* $\mathcal{S}_1$ *and* $\mathcal{S}_0$.
7. *Return L.*

Theorem 6. *Let* $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$ *be a symmetric edge-weighted image digraph with* $\omega : \mathcal{A} \to \mathbb{R}$. *Let L be a segmentation returned by Algorithm 4 applied to G, non-empty disjoint seed sets* $\mathcal{S}_1$ *and* $\mathcal{S}_0$, *cost map* $C : \mathcal{N} \to [0, \infty)$, *and parameters* $R > 0$ *and* $\Delta > 0$. *Assume that* $\mathcal{S}_1$ *and* $\mathcal{S}_0$ *are* LB^R_Δ*-consistent, that is, that*

(⋆) there exists a labeling satisfying seeds and LB^R_Δ *constraints.*

Then L satisfies seeds and LB^R_Δ *constraints and maximizes the energy* $\varepsilon_{\min}$, *given by (2.3) w.r.t. G, among all segmentations satisfying these constraints.*

Proof. In this proof $\varepsilon^G_{\min}$ and $\varepsilon^{G'}_{\min}$ denote the energy $\varepsilon_{\min}$ with respect to G and G', respectively. Let $\mathcal{L} := \{\langle p, q \rangle \in \mathcal{N} \times \mathcal{N} : 0 < \|p - q\| \leq R \;\&\; C(p) \geq C(q) + \Delta\}$ and $\mathcal{M} := \{\langle s, t \rangle : (s, t) \in \mathcal{L}\} \setminus \mathcal{A}$. It is easy to see that after the execution of lines 1-5 we have $\mathcal{A}' = \mathcal{A} \cup \mathcal{L} \cup \mathcal{M}$ and

$$\omega'(s, t) = \begin{cases} -\infty & \text{for } \langle s, t \rangle \in \mathcal{L}, \\ \infty & \text{for } \langle s, t \rangle \in \mathcal{M}, \\ \omega(s, t) & \text{otherwise, that is for } \langle s, t \rangle \in \mathcal{A} \setminus \mathcal{L}. \end{cases}$$

Also, by Proposition 1, after the execution of line 6 the labeling L satisfies the seed constraints and maximizes the energy $\varepsilon^{G'}_{\min}$ among all segmentations satisfying seed constraints. We need to show that L satisfies also LB^R_Δ constraints an that it maximizes $\varepsilon^G_{\min}$ among all segmentations satisfying these constraints.

To see this, let $L' : \mathcal{N} \to \{0,1\}$ be an arbitrary labeling satisfying seeds and LB^R_Δ constraints. It exists by ($\star$). Then, by the definition of LB^R_Δ constraints, the set $T' := \{\langle p,q\rangle \in \mathcal{A}' : L'(p) > L'(q)\}$ is disjoint with $\mathcal{L}$. In particular,

$$\begin{aligned}\varepsilon^{G'}_{\min}(L) &\geq \varepsilon^{G'}_{\min}(L')\\ &= \min\{\omega'(s,t) : \langle s,t\rangle \in \mathcal{A}' \;\&\; L'(s) > L'(t)\}\\ &> -\infty.\end{aligned}$$

Hence

$$\begin{aligned}\varepsilon^{G'}_{\min}(L) &= \min\{\omega'(s,t) : \langle s,t\rangle \in \mathcal{A}' \;\&\; L(s) > L(t)\}\\ &> -\infty,\end{aligned}$$

so that the set $T := \{\langle p,q\rangle \in \mathcal{A}' : L(p) > L(q)\}$ must be also disjoint with $\mathcal{L}$. This means that L satisfies LB^R_Δ constraints. To finish the proof we need to show that $\varepsilon^G_{\min}(L) \geq \varepsilon^G_{\min}(L')$. For this notice first that

$$\varepsilon^{G'}_{\min}(L') = \varepsilon^G_{\min}(L'). \tag{3.1}$$

Indeed, $T' \cup T$ is disjoint with $\mathcal{L}$, so $\langle s,t\rangle \in \mathcal{A}'$ and $L'(s) > L'(t)$ implies that $\langle s,t\rangle \in (\mathcal{A} \setminus \mathcal{L}) \cup \mathcal{M}$. Thus, since $\omega' = \omega$ on $\mathcal{A} \setminus \mathcal{L}$ and $\omega' = \infty$ on $\mathcal{M}$,

$$\begin{aligned}&\varepsilon^{G'}_{\min}(L')\\ &= \min\{\omega'(s,t) : \langle s,t\rangle \in \mathcal{A}' \;\&\; L'(s) > L'(t)\}\\ &= \min\left(\{\omega'(s,t) : \langle s,t\rangle \in \mathcal{A} \setminus \mathcal{L} \;\&\; L'(s) > L'(t)\} \cup \{\infty\}\right)\\ &= \min\left(\{\omega(s,t) : \langle s,t\rangle \in \mathcal{A} \;\&\; L'(s) > L'(t)\} \cup \{\infty\}\right)\\ &= \varepsilon^G_{\min}(L'),\end{aligned}$$

as needed. Finally, using (3.1) for L and L', we obtain

$$\varepsilon^G_{\min}(L) = \varepsilon^{G'}_{\min}(L) \geq \varepsilon^{G'}_{\min}(L') = \varepsilon^G_{\min}(L'),$$

finishing the proof. □

3.2.3 Transitive Reduction

The nodes $\mathcal{N}$ and the set of arcs

$$\{\langle p,q\rangle \in \mathcal{A}' : \|p-q\| \leq R \;\&\; C(p) \geq C(q) + \Delta\}$$

from Line 2 of Algorithm 4 form a *Directed Acyclic Graph* (DAG). Let's denote this DAG as LB_{DAG}.[1] Several of these $-\infty$-valued arcs in ω' that represent the LB constraint may actually be redundant, i.e., they are just "shortcuts" linking already connected nodes,

[1] Note that the sequence of vertices $\langle v_1, \ldots, v_N\rangle$, such that $C(v_i) \geq C(v_j)$ for $i < j$, corresponds to a topological ordering of LB_{DAG}.

without offering a better value, so we can apply a *transitive reduction* [1] to remove them from this DAG. A transitive reduction of a directed graph D is another directed graph with the same vertices and as few arcs as possible, such that if there is a (directed) path from vertex v to vertex w in D, then there is also such a path in the reduction. The transitive reduction of a finite DAG is unique and is a subgraph of the given graph [1].

Some examples are shown in Figure 3.3. Following the letter naming of the vertices, as shown in Figure 3.3a, note that the arcs $\langle c, j\rangle, \langle h, o\rangle, \langle m, t\rangle, \langle p, w\rangle, \langle u, B\rangle$ and $\langle v, C\rangle$ in Figure 3.3c are all redundant and have been removed in Figure 3.3d. For instance, the arc $\langle c, j\rangle$ is redundant, given that from the node c it is possible to reach j by an alternative route, following the arcs $\langle c, i\rangle$ and $\langle i, j\rangle$ in Figure 3.3c.

The fastest known worst-case time bound for transitive reduction in dense graphs is $O(N^{2.3729})$, where N is the number of vertices. For sparse graphs, transitive reductions can also be found in time $O(NM)$, where N is the number of vertices and M is the number of arcs in a directed acyclic graph. To do so, apply a linear time longest path algorithm [76] in the given directed acyclic graph, for each possible choice of starting vertex. From the computed longest paths, keep only those of length one (single arc); in other words, keep those arcs $\langle u, v\rangle$ for which there exists no other path from u to v.

In this section, we present an algorithm that can be used to eliminate all redundant arcs representing the LB constraint from the graph G' used in Algorithm 4. All examples of transitive reduction shown in Figure 3.3 were computed by Algorithm 5. This algorithm takes advantage of the prior knowledge of our particular application in order to have a better performance, close to linear, for LB_{Δ}^{R} configurations defining more localized arcs in the image domain, that is, for small radius values of R and large values of Δ. In case of the worst scenario, i.e., large radius values of R and small values of Δ, its complexity becomes $O(NM)$, where N is the number of vertices and M is the number of arcs of G'.

Algorithm 5. – *Transitive Reduction for the* LB *constraint*

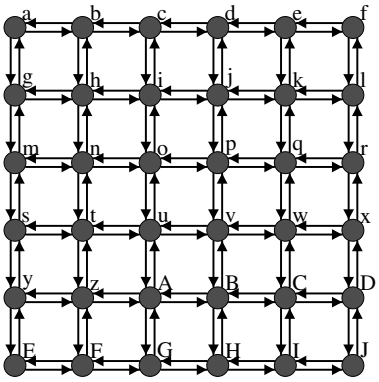
Input: Symmetric edge-weighted digraph $G' = \langle \mathcal{N}, \mathcal{A}', \omega' \rangle$ as used in Algorithm 4 from [68].

Output: An updated version of the graph G' with fewer arcs.

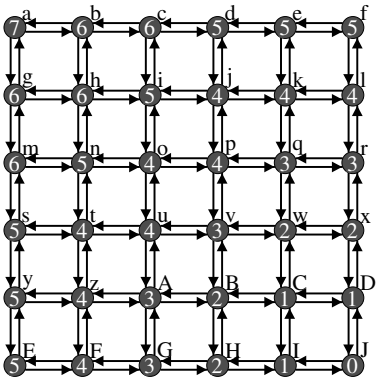
Auxiliary: An initially empty set R, an initially empty stack S for traversing the graph by depth-first search, and a binary array $B : \mathcal{N} \to \{0, 1\}$, where $B(t) = 1$ for processed nodes and $B(t) = 0$ for unprocessed nodes.

```
1.  For each p ∈ N, do
2.      Set c_min ← ∞;
3.      For each ⟨p, q⟩ ∈ A' such that ω'(p, q) = −∞, do
4.          Set B(q) = 1 and insert q in S;
5.          Set c_min ← min{c_min, C(q)}.
6.      While S is not empty, do
7.          Pop a node s from the stack S;
8.          Insert s in R;
9.          For each ⟨s, t⟩ ∈ A' such that ω'(s, t) = −∞, do
10.             If ⟨p, t⟩ ∈ A' and ω'(p, t) = −∞ and ω'(t, p) = ∞ then
11.                 Remove arcs ⟨p, t⟩ and ⟨t, p⟩ from G'.
12.             Else If C(t) ≥ c_min + Δ and B(t) = 0 then
13.                 Set B(t) = 1 and insert t in S.
14.     While R is not empty, do
15.         Remove r from R and set B(r) = 0.
```

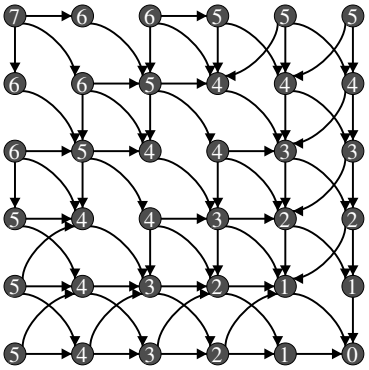
For each node p of the main loop of Algorithm 5, in the first inner loop (Lines 3-5), we consider the arcs $\langle p, q \rangle$ to immediate neighbors of p in LB_{DAG} to find the lowest value of $C(q)$ in its neighborhood. The loop of Lines 6-13 uses a stack to perform a local depth-first search looking for longer alternative routes for the neighbors t of p in LB_{DAG}. Whenever a longer route is found, then the direct arc $\langle p, t \rangle$ is removed from the graph (Line 11). The condition on Line 12 restricts the search based on the map C and Δ, in order to maintain a local search, pruning unpromising search subspaces as early as possible, and the binary map B is used to prevent a node from being repeatedly processed. Finally, the final inner loop of Lines 14-15 is used to reset the values of the auxiliary search map B.



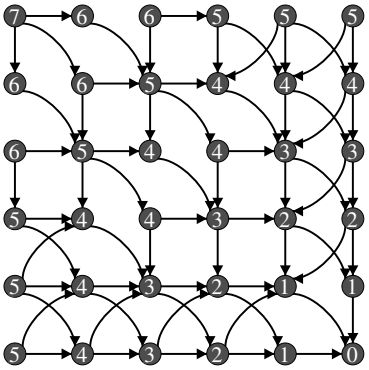
(a) A 4-neighborhood graph ($|\mathcal{A}| = 120$)



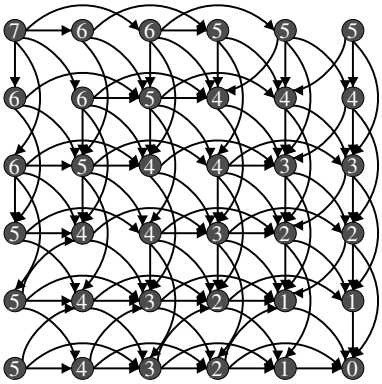
(b) The values $C(t)$ are indicated inside the nodes



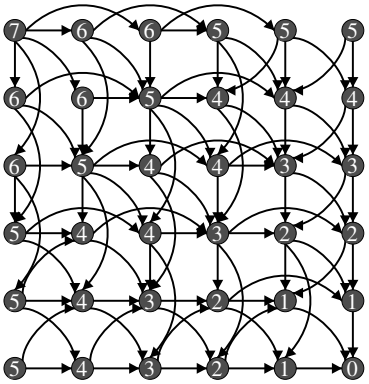
(c) LB^R_Δ with $\Delta = 1$ and $R = 1.5$ ($|\mathcal{A}'| = 190$)



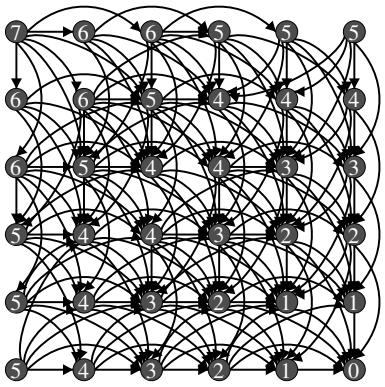
(d) Transitive reduction of Fig. 3.3c ($|\mathcal{A}'| = 178$)



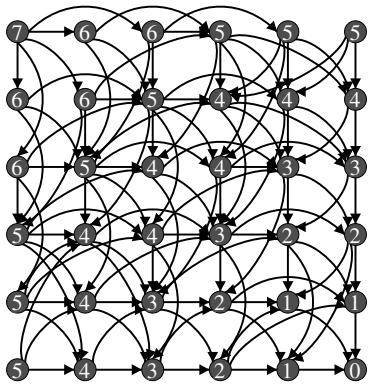
(e) LB^R_Δ with $\Delta = 1$ and $R = 2.0$ ($|\mathcal{A}'| = 278$)



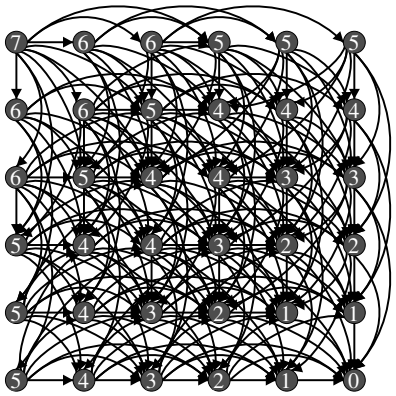
(f) Transitive reduction of Fig. 3.3e ($|\mathcal{A}'| = 226$)



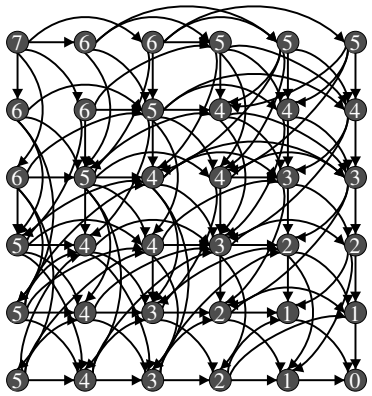
(g) LB_Δ^R with $\Delta = 1$ and $R = 2.5$ ($|\mathcal{A}'| = 410$)



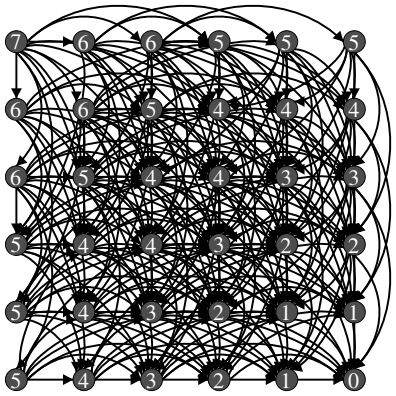
(h) Transitive reduction of Fig. 3.3g ($|\mathcal{A}'| = 274$)



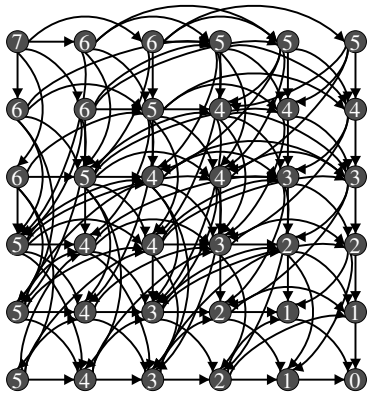
(i) LB_Δ^R with $\Delta = 1$ and $R = 3.0$ ($|\mathcal{A}'| = 530$)



(j) Transitive reduction of Fig. 3.3i ($|\mathcal{A}'| = 306$)



(k) LB_Δ^R with $\Delta = 1$ and $R = 3.5$ ($|\mathcal{A}'| = 642$)



(l) Transitive reduction of Fig. 3.3k ($|\mathcal{A}'| = 334$)

Figure 3.3: *(a) A 4-neighborhood graph. (b) The values $C(t)$ are indicated inside the nodes. (c-l) Display of the $-\infty$-valued arcs in G' for LB_Δ^R with $\Delta = 1$ for different radius values, before and after transitive reduction (first/second columns, respectively). The total number of arcs $|\mathcal{A}'|$ in G' before and after transitive reduction are shown for each graph, where the symbol $|\cdot|$ denotes the cardinality of a set. Note that the bigger the radius, the greater will be the reduction of the total number of arcs by transitive reduction, reaching 47.97% for $R = 3.5$.*

3.2.4 Special case: The Band constraint algorithm

Now let's consider the limit case of the segmentation by OIFT subject to the LB constraint, as $R \to \infty$. Although in theory Algorithm 4 could be used to handle this case, in practice it would become unfeasible due to the excessive amount of arcs created for the graph G', even considering the transitive reduction by Algorithm 5. In this section we present an alternative and efficient algorithm (Algorithm 6) for solving this particular case.

According to Definition 4, as $R \to \infty$, the Local Band constraint (LB^R_Δ) becomes the Band constraint (B_Δ), which implies that $C(t) < C(s) + \Delta$ for all $t \in \mathcal{O}$ and $s \in \mathcal{N} \setminus \mathcal{O}$.

Algorithm 6. – *Segmentation by OIFT subject to the* B *constraint*

Input: Symmetric edge-weighted image digraph $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$ and non-empty disjoint seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$, cost map $C : \mathcal{N} \to [0, \infty)$, and parameter $\Delta > 0$.
Output: The label map $L : \mathcal{N} \to \{0, 1\}$.
Auxiliary: Initially empty sets Q_0,Q_1,Q_x and Q, and an array of status $S : \mathcal{N} \to \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes. The value $V(t)$ represents a potential penalty that a change of $L(t)$ would contribute to $\varepsilon_{\min}(L)$.

1. **For each** $t \in \mathcal{N}$, **do**
2. Set $S(t) \leftarrow 0$ and $V(t) \leftarrow \infty$;
3. Insert t in Q_x;
4. **If** $t \in \mathcal{S}_0$, **then**
5. $V(t) \leftarrow -\infty$, $L(t) \leftarrow 0$, and insert t in Q;
6. **If** $t \in \mathcal{S}_1$, **then**
7. $V(t) \leftarrow -\infty$, $L(t) \leftarrow 1$, and insert t in Q.
8. **While** $Q \neq \emptyset$ **do**
9. Remove s from Q such that $V(s)$ is minimum;
10. Set $S(s) \leftarrow 1$;
11. Propagate(s,G,Q,V,L,S) by Algorithm 7;
12. Remove s from Q_x;
13. **If** $L(s) = 0$, **then**
14. Insert s in Q_0;
15. **While** $Q_x \neq \emptyset \wedge (\max_{a \in Q_x} C(a) - \min_{b \in Q_0} C(b) \geq \Delta)$ **do**
16. Remove t from Q_x such that $C(t)$ is maximum;
17. Insert t in Q_0 and set $L(t) \leftarrow 0$;
18. Set $S(t) \leftarrow 1$;
19. **If** $t \in Q$, **then** remove t from Q.
20. Propagate(t,G,Q,V,L,S) by Algorithm 7.
21. **Else If** $L(s) = 1$, **then**
22. Insert s in Q_1;
23. **While** $Q_x \neq \emptyset \wedge (\max_{a \in Q_1} C(a) - \min_{b \in Q_x} C(b) \geq \Delta)$ **do**
24. Remove t from Q_x such that $C(t)$ is minimum;
25. Insert t in Q_1 and set $L(t) \leftarrow 1$;
26. Set $S(t) \leftarrow 1$;
27. **If** $t \in Q$, **then** remove t from Q.
28. Propagate(t,G,Q,V,L,S) by Algorithm 7.
29. Return L.

Algorithm 7. – *Propagate*

Input: Pixel $s \in \mathcal{N}$, graph $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$, set Q, cost map V, label map $L : \mathcal{N} \to \{0,1\}$, and the array of status $S : \mathcal{N} \to \{0,1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.
Output: The updated priority queue Q and the updated maps V and L.
Auxiliary: Variable tmp.

1. ***For each*** $\langle s,t \rangle \in \mathcal{A}$ *such that* $S(t) = 0$ ***do***
2. ***If*** $L(s) = 1$, ***then*** $tmp \leftarrow \omega(s,t)$.
3. ***Else*** $tmp \leftarrow \omega(t,s)$;
4. ***If*** $tmp < V(t)$, ***then***
5. *Set* $V(t) \leftarrow tmp$ *and* $L(t) \leftarrow L(s)$.
6. ***If*** $t \notin Q$, ***then*** *insert* t *in* Q.

Algorithm 6 corresponds to a modified OIFT algorithm, similar to the one used to compute the Boundary Band (BB) [5, 6], but slightly simpler. The auxiliary function "Propagate" (Algorithm 7) corresponds to the same code of Lines 10-15 of Algorithm 1.

Theorem 7. *Let* $G = \langle \mathcal{N}, \mathcal{A}, \omega \rangle$ *be a symmetric edge-weighted image digraph with* $\omega : \mathcal{A} \to \mathbb{R}$. *Let* L *be a segmentation returned by Algorithm 6 applied to* G, *non-empty disjoint seed sets* $\mathcal{S}_1$ *and* $\mathcal{S}_0$, *cost map* $C : \mathcal{N} \to [0,\infty)$, *and parameter* $\Delta > 0$. *Assume that* $\mathcal{S}_1$ *and* $\mathcal{S}_0$ *are* B_Δ*-consistent, that is, that*

(⋆) there exists a labeling satisfying seeds and B_Δ *constraints.*

Then L *satisfies seeds and* B_Δ *constraints and maximizes the energy* $\varepsilon_{\min}$, *given by (2.3) w.r.t.* G, *among all segmentations satisfying these constraints.*

Proof. During the computation of Algorithm 6, three sets of pixels are indirectly defined, a growing object $\mathcal{O}' = \{t \in \mathcal{N} : S(t) = 1 \wedge L(t) = 1\}$, a growing background $\mathcal{B}' = \{t \in \mathcal{N} : S(t) = 1 \wedge L(t) = 0\}$ and an unprocessed zone $\mathcal{U}' = \{t \in \mathcal{N} : S(t) = 0\}$, initially given by $\mathcal{N}$, which is constantly reducing in size. From these sets, two candidate objects $\mathcal{O}_1 = \mathcal{O}'$ and $\mathcal{O}_2 = \mathcal{N} \setminus \mathcal{B}'$ are defined.

At each iteration of the main loop at Lines 9-10, Algorithm 6 always modifies the candidate object ($\mathcal{O}_1$ or $\mathcal{O}_2$) that has the lowest energy (given by $V(s)$ at Line 9), through the arc with the worst cost at its cut boundary. This is an event that should not be undone, since it is the only change of a pixel that can improve the candidate object with the worst energy. Changing other pixels will not improve it. However, this change may lead to a violation of the Band constraint for the corresponding candidate object ($\mathcal{O}_1$ or $\mathcal{O}_2$).

(V1) Violations for $\mathcal{O}_1$: The candidate object $\mathcal{O}_1$ is modified by the acquisition of a pixel s at Lines 9-10. Violations can only occur when $C(s) \geq C(t) + \Delta$ for some $t \in \mathcal{N} \setminus \mathcal{O}_1$. Since s cannot be modified, the only way to correct this violation is by adding t to $\mathcal{O}_1$ (Lines 25-26). This new insertion of t cannot in turn generate other unforeseen violations in a chain of events since $C(s) > C(t)$. However, there may be multiple pixels in a condition similar to that of node t with respect to s. They are handled in a similar way through the loop of Lines 23-28.

(V2) Violations for $\mathcal{O}_2$: The candidate object $\mathcal{O}_2$ is modified by the removal of a pixel s at Lines 9-10. Violations can only occur when $C(t) \geq C(s) + \Delta$ for some $t \in \mathcal{O}_2$. Since s cannot be modified, the only way to correct this violation is by removing t from $\mathcal{O}_2$ (Lines 17-18). This new removal of t cannot in turn generate other unforeseen violations in a chain of events since $C(t) > C(s)$. However, there may be multiple pixels in a condition similar to that of node t with respect to s. They are handled in a similar way through the loop of Lines 15-20.

Algorithm 6 has the following invariant:

Proposition 8. *At the end of each iteration of the main loop (Lines 8-28), we have that $\mathcal{O}_1$ and $\mathcal{O}_2$ satisfy the Band constraint (i.e., $\mathcal{O}_1$ is B_Δ and $\mathcal{O}_2$ is B_Δ) and $\mathcal{O}_1 \subseteq \mathcal{O}_2$.*

Condition $\mathcal{O}_1 \subseteq \mathcal{O}_2$ is guaranteed by construction of Algorithm 6[2].

$\mathcal{O}_1$ and $\mathcal{O}_2$ satisfying the Band constraint can be proved by induction. The base case is assured by the fact that $\mathcal{S}_1$ and $\mathcal{S}_0$ are assumed to be B_Δ-consistent. Therefore, in the first iterations of the main loop, the seeds will be processed and the handling of violations V1 and V2, as explained previously, will establish the base case, being the processing order of seeds irrelevant. For the induction step, we assume that $\mathcal{O}_1$ is B_Δ and $\mathcal{O}_2$ is B_Δ before executing the body of the main loop and we prove that they still satisfy the Band constraint at the end of the loop. The induction step is guaranteed by the handling of violations V1 and V2, as explained previously, in Lines 23-28 and Lines 15-20 of Algorithm 6 respectively. However, there are two important issues that need to be discussed.

(i) Lines 23-28 only consider pixels t from the set $Q_x = \mathcal{U}'$, whereas from the discussion of V1 we should actually consider $t \in \mathcal{N} \setminus \mathcal{O}_1 = \mathcal{U}' \cup \mathcal{B}'$.

(ii) Lines 15-20 only consider pixels t from the set $Q_x = \mathcal{U}'$, whereas from the discussion of V2 we should actually consider $t \in \mathcal{O}_2 = \mathcal{U}' \cup \mathcal{O}_1$.

For case (i), we must prove that there are no violations of the Band constraint of $\mathcal{O}_1$ for $t \in \mathcal{B}'$ during the computation of Lines 23-28, and therefore, there is no need to test pixels from this set $\mathcal{B}'$. We can carry out a proof by contradiction, assume that there is $t \in \mathcal{B}'$ such that $C(s) \geq C(t) + \Delta$. Note that $t \in \mathcal{B}'$ implies that $t \notin \mathcal{O}_2$. Since s corresponds to a pixel removed from Q in Line 9, before computing Line 10, it previously had $S(s) = 0$ and therefore we have $s \in \mathcal{O}_2$. However, $t \notin \mathcal{O}_2$, $s \in \mathcal{O}_2$ and $C(s) \geq C(t) + \Delta$ indicate a violation of the Band constraint for $\mathcal{O}_2$, leading to a contradiction, since $\mathcal{O}_2$ is B_Δ by the induction hypothesis.

For case (ii), we must prove that there are no violations of the Band constraint of $\mathcal{O}_2$ for $t \in \mathcal{O}_1$ during the computation of Lines 15-20, and therefore, there is no need to test pixels from this set $\mathcal{O}_1$. We can carry out a proof by contradiction, assume that there is

[2] Note that, in Lines 9-10, since s was removed from set Q on Line 9, we can conclude that it previously had $S(s) = 0$. Therefore, prior to the execution of Lines 9-10, we had $s \notin \mathcal{O}_1$ and $s \in \mathcal{O}_2$. During the execution of Lines 9-10, if $L(s) = 1$ then s is inserted into $\mathcal{O}_1$, but it was already in $\mathcal{O}_2$. On the other hand, if $L(s) = 0$ then s is removed from $\mathcal{O}_2$, but $s \notin \mathcal{O}_1$. A pixel t can also be removed from $\mathcal{O}_2$ on Lines 17-18, but according to Line 16, this pixel t was previously in the set Q_x, which stores unprocessed pixels, so t was not in $\mathcal{O}_1$. A pixel t can also be inserted into $\mathcal{O}_1$ on Lines 25-26, but according to Line 24, this pixel t was previously in the set Q_x, which stores unprocessed pixels, so $t \in \mathcal{O}_2$ since $t \notin \mathcal{B}'$.

$t \in \mathcal{O}_1$ such that $C(t) \geq C(s) + \Delta$. Since s corresponds to a pixel removed from Q in Line 9, before computing Line 10, it previously had $S(s) = 0$ and therefore we have $s \notin \mathcal{O}_1$. However, $t \in \mathcal{O}_1$, $s \notin \mathcal{O}_1$ and $C(t) \geq C(s) + \Delta$ indicate a violation of the Band constraint for $\mathcal{O}_1$, leading to a contradiction, since $\mathcal{O}_1$ is B_Δ by the induction hypothesis.

Following the invariant of Proposition 8 and the fact that the set $Q_x = \mathcal{U}'$ is always decreasing in size, in the end, all pixels will have been removed from set Q and objects $\mathcal{O}_1$ and $\mathcal{O}_2$ will have converged to the same set of pixels that satisfy the Band constraint. A global maximum of the energy function $\varepsilon_{\min}$ (Eq. 2.3) subject to the Band constraint is guaranteed by the energy competition between the two candidate objects $\mathcal{O}_1$ and $\mathcal{O}_2$ throughout the process, always carrying out operations strictly necessary to improve the object with lower energy and conserving the one with higher energy. □

For implementation purposes, the sets Q_0,Q_1,Q_x and Q must be implemented using appropriate priority queue data structures in order to support the efficient removal of their extreme value elements (maximum or minimum). The queues Q_0,Q_1 and Q_x represent the sets $\mathcal{B}'$, $\mathcal{O}'$ and $\mathcal{U}'$, respectively. Regarding the computational complexity of Algorithm 6, if a binary heap is used for the queues, then Algorithm 6 can be implemented in $O((m + n) \log n)$, where $n = |\mathcal{N}|$, $m = |\mathcal{A}|$, and $|\cdot|$ denotes the set cardinality, that is, the same complexity as Algorithm 1.

3.2.5 Shape Templates

We can generate the cost map C in other ways, this includes using other cost functions instead of using the geodesic length in the first step or using predefined cost maps obtained externally. In the latter case, we will be directing the segmentation result to follow the shapes of the level curves of the provided cost map.

In several works [29, 101], the shape prior is constructed from a fixed instance of the target object taken as a reference (Figures 3.4a and d), by calculating the *Euclidean Distance Transform* (EDT) in relation to its contour (Figures 3.4b and e). Note that, in this case, the zero level curve corresponds exactly to the given object contour. In order to adapt this scheme for use in our approaches as the cost map C, we must consider a signed distance function (assuming negative values for distances within the object as in [101]) and then subtract its most negative value from it to shift the brightness range to consider only positive values (Figures 3.4c and f). However, the problem with this approach is that the level curves farther from the initial contour suffer deformations, for example, their salient parts become rounded on larger scales, as indicated by the sampled level curves as highlighted in Figures 3.4c and f. Therefore, this approach is only viable when we have a good a priori idea of an appropriate scale for the object of interest. This may be true for some applications, such as the tracking of objects in videos when the changes between subsequent frames are small and we can take the size of the object in the previous frame to set the scale for the next frame [101]. In some cases, we can also build these templates based on the average shape learned via training, or using the inverted membership image of *Fuzzy Object Models* (FOMs) [90, 91, 88, 89, 64, 65, 43, 42, 73, 61, 60].

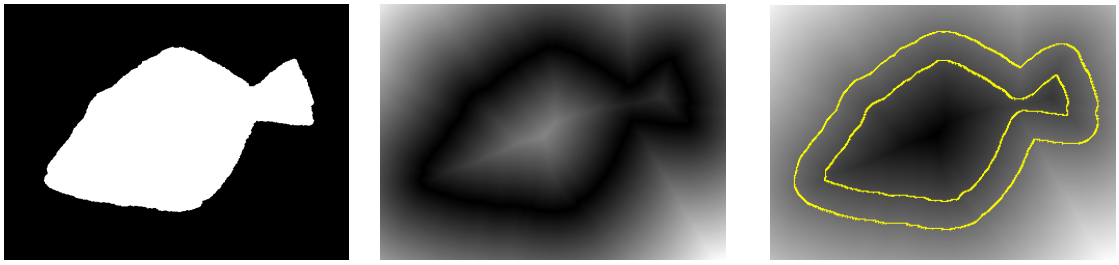
For the creation of a cost template invariant to scale changes of a given predefined shape, it is necessary that its scale variations relative to a fixed reference point keep the smaller scale shapes entirely contained within the upscaled shapes. Note that star-convex shapes meet this constraint with respect to their star centers, as seen in Figure 3.5. When the considered shape is non-star convex, but it can be satisfactorily approximated by a similar star-convex object, one option is to build the cost template based on this approximated object and then handle the approximation errors by increasing the tolerance to deformations via the parameter Δ of the LB constraint.

Many natural shapes can be described by the Gielis geometric equation and its generalized version, including shapes of some starfish and plant leaves [81]. As described below, all these shapes that can be formulated by Gielis equation are star convex shapes that can be used immediately to create scale invariant templates for the LB constraint.

Before introducing Gielis equation, let's discuss the equation of Lamé curves [36, 33], which can describe several classical geometric equations (circles and ellipses, as well as squares and rectangles) as special cases. It also has been shown to be valid for describing the actual cross-sections of tree rings and bamboo shoots [32, 79, 97, 38]. Its mathematical expression is defined by Equation 3.2:

$$\left|\frac{x}{A}\right|^n + \left|\frac{y}{B}\right|^n = 1 \tag{3.2}$$

where A,B and n are real numbers.



(a) Flatfish (b) EDT (c) Flatfish in multiple scales

(d) Bat (e) EDT (f) Bat in multiple scales

Figure 3.4: *(a & d) Sample shapes taken from MPEG-7 shape database[a]. (b & e) The unsigned Euclidean distance transform as used in [29]. (c & f) The corresponding adapted cost maps to be used as shape templates, with some level curves highlighted in yellow.*

[a] https://dabi.temple.edu/external/shape/MPEG7/dataset.html

This equation in polar coordinates can be rewritten as a function $r : [0, 2\pi) \to \mathbb{R}$, given by Equation 3.3, where $r(\varphi)$ and φ are the polar radius and the angle between the straight line where the polar radius lies and the x-axis, respectively.

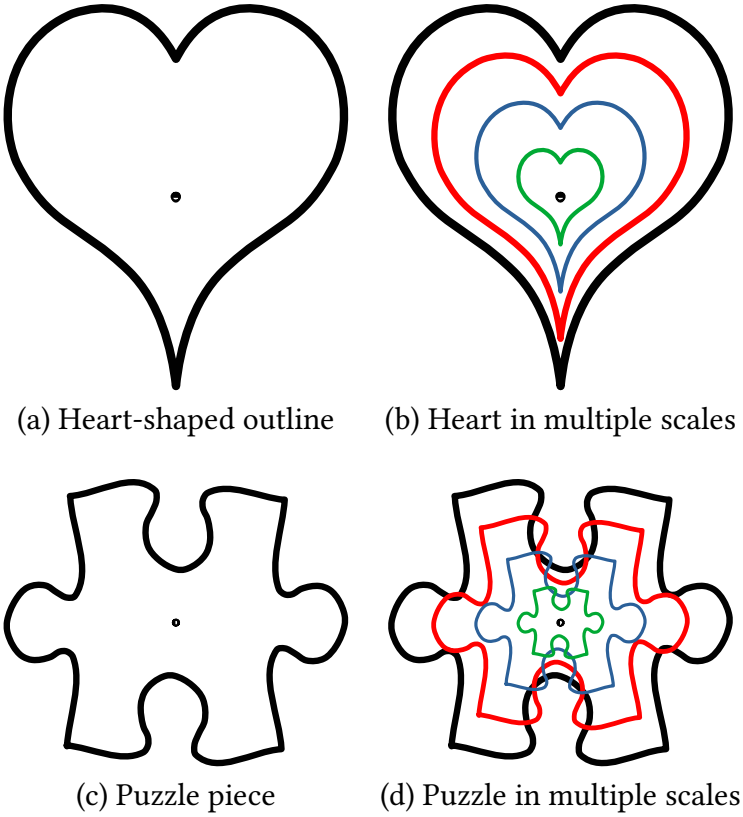
$$r(\varphi) = \left(\left| \frac{1}{A} \cos(\varphi) \right|^n + \left| \frac{1}{B} \sin(\varphi) \right|^n \right)^{-1/n} \tag{3.3}$$

Gielis proposed a more general polar equation that can reflect more complex natural shapes [33, 32]:

$$r(\varphi) = \left(\left| \frac{1}{A} \cos\left(\frac{m}{4}\varphi\right) \right|^{n_2} + \left| \frac{1}{B} \sin\left(\frac{m}{4}\varphi\right) \right|^{n_3} \right)^{-1/n_1} \tag{3.4}$$

where n_1, n_2 and n_3 are real constants, and the positive integer m was introduced to enhance the flexibility of Lamé curves, so that it can now produce regular, or at least very approximately regular polygons [10, 62]. Given that $r(\varphi) > 0$ and we have a single point on the curve for each angle φ in the interval $[0, 2\pi)$, we can conclude that all shapes described by Equation 3.4 are star convex.

Gielis equation has been used to simulate many natural shapes, e.g., diatoms, eggs,



(a) Heart-shaped outline (b) Heart in multiple scales

(c) Puzzle piece (d) Puzzle in multiple scales

Figure 3.5: *(a-b) Example of a shape that can be used as a template. From (b) it is clear that the shape is star convex, since it is possible to generate a predefined cost map so that the smaller shapes are always inside the larger shapes. Note that the multiple colored contours representing different scales do not intersect with each other. (c-d) Example of a shape that cannot be used as a template. From (d) it is clear that we have a non-star convex shape in relation to the center. For instance, notice that the red contour crosses the black contour, implying that distinct scales cannot be properly represented as different level curves of a cost map.*

cross sections of plants, snowflakes and starfish [33, 32]. For instance, it has shown its validity in describing leaf shapes of *Hydrocotyle vulgaris L.*, *Polygonum perfoliatum L.*, seed planar projections of *Ginkgo biloba L.* [82, 86], and the actual leaf shapes of 46 bamboo species [32, 80, 51]. A more generalized version of Gielis geometric equation was also proposed by Peijian Shi et al. [81]. This study was motivated by the wide diversity of shapes of starfish, in order to support pincushion starfish of genus *Culcita* and biscuit starfish of genus *Tosia*.

Algorithm 8 computes a template C of shapes given by Gielis equation that is appropriate for the LB constraint and other shape constraints previously discussed in Section 2.4.

Algorithm 8. – *Template by Gielis Equation*

Input: The parameters of the Gielis equation (A, B, n_1, n_2, n_3, and m), the coordinates (c_x, c_y) of the model center, and the template dimensions T_x and T_y.
Output: The template (cost map) $C: \mathcal{N} \rightarrow [0, \infty)$, where $\mathcal{N} \subset \mathbb{Z}^2$ and for $(x, y) \in \mathcal{N}$ we have $0 \leq x < T_x$ and $0 \leq y < T_y$.
Auxiliary: Variables φ,r,s,d,d_x,d_y, $s_{\min}$ and $d_{\min}$.

1. *Set* $s_{\min} \leftarrow \infty$ *and* $d_{\min} \leftarrow \infty$;
2. ***For each*** $(x, y) \in \mathcal{N}$ *such that* $x = 0 \vee x = T_x - 1 \vee y = 0 \vee y = T_y - 1$, ***do***
3. *Set* $d_x \leftarrow x - c_x$ *and* $d_y \leftarrow y - c_y$;
4. *Compute* $\varphi \leftarrow \text{arctan2}\,(d_y, d_x)$;
5. ***If*** $\varphi < 0$, ***then*** $\varphi \leftarrow \varphi + 2\pi$.
6. *Compute* $r \leftarrow \text{GielisEquation}\,(\varphi, A, B, n_1, n_2, n_3, m)$;
7. *Set* $d \leftarrow \sqrt{d_x^2 + d_y^2}$;
8. *Set* $s \leftarrow d/r$;
9. ***If*** $s < s_{\min} \vee (s = s_{\min} \wedge d < d_{\min})$ ***then***
10. *Set* $s_{\min} \leftarrow s$ *and* $d_{\min} \leftarrow d$.
11. ***For each*** $p = (x, y) \in \mathcal{N}$, ***do***
12. *Set* $d_x \leftarrow x - c_x$ *and* $d_y \leftarrow y - c_y$;
13. *Compute* $\varphi \leftarrow \text{arctan2}\,(d_y, d_x)$;
14. ***If*** $\varphi < 0$, ***then*** $\varphi \leftarrow \varphi + 2\pi$.
15. *Compute* $r \leftarrow \text{GielisEquation}\,(\varphi, A, B, n_1, n_2, n_3, m)$;
16. *Set* $d \leftarrow \sqrt{d_x^2 + d_y^2}$;
17. *Set* $s \leftarrow d/r$;
18. *Set* $C(p) \leftarrow \text{Round}\,((s/s_{\min}) \cdot d_{\min})$.
19. *Return* C.

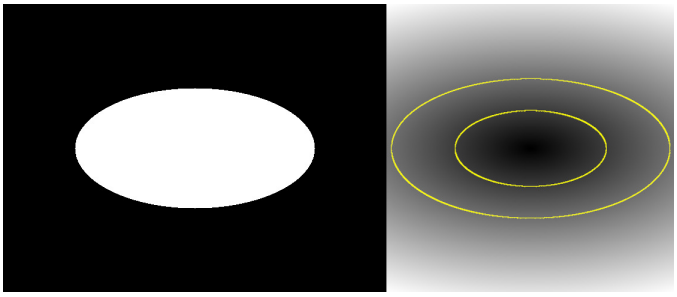
The (c_x, c_y) image coordinates are used to define the origin for the purpose of calculating the Gielis equation. In Lines 4 and 13 of Algorithm 8, the function $\text{arctan2}\,(d_y, d_x)$ is used to compute φ by the two-argument variant of arctangent with a range of $(-\pi, \pi]$. Gielis function $r(\varphi)$ by Equation 3.4 is then computed on Lines 6 and 15.

The d/r ratios stored in s from Lines 8 and 17 are used to indicate the different scale values for the template, but these values must be normalized and quantized into an appropriate range for the final digital cost map C. To this end, the first loop (Lines 2-10) is used to find an appropriate scale value $s_{\min}$ (and corresponding distance $d_{\min}$) for the purpose of normalization by inspecting the various directions along the image border. After that, in the second loop (Lines 11-18), the scale values of all pixels are calculated and the final normalization/quantization is done on Line 18.

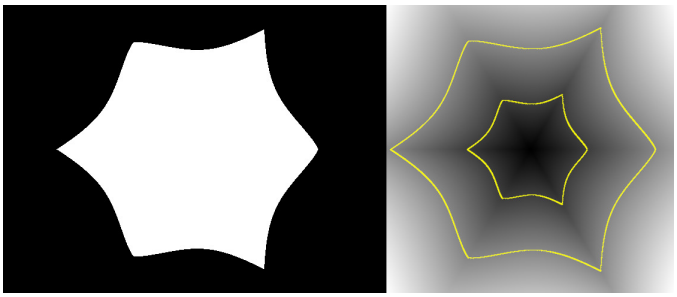
Figure 3.6, in its left column, presents some examples of shapes that can be modeled via Equation 3.4. The corresponding templates C obtained by Algorithm 8 are shown on its right column, where some level curves (isolines) of the templates are shown as yellow outlines to improve the visualization of the results. For each example, the corresponding values of the parameters used (i.e., A,B,n_1,n_2,n_3 and m) are indicated just below the figure. We considered $(c_x, c_y) = (T_x/2, T_y/2)$ for all examples, except for the last template, where the center was shifted to the left in order to make a better use of the image area.

There are still the problems of how to find the proper translation and rotation of the template in relation to the desired object present in the image. In [29], rotation and trans-

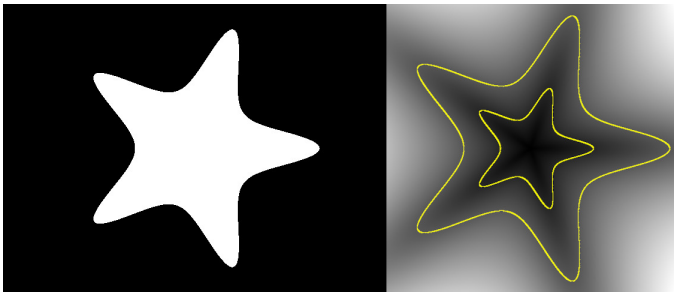
lation are estimated based on the given object markers. In [95], a sequence of intermediate segmentations is produced, starting with a segmentation free of shape constraints and the penalties for shape constraints are gradually increased, with translation and rotation being estimated between iterations based on the previous segmentation of the sequence. Although strategies similar to these can be adapted to our case, unfortunately they do not guarantee that good results will be achieved. Accurately estimating rotation based on object markers requires a reasonable amount of markers preferably distributed close to the object boundary. This fact is evident from the examples given in [29]. Progressively raising the shape constraint penalty, with the template pose being estimated based on the previous segmentation, will only work in easier cases, when a reasonable approximation of the object can be obtained for the first segmentations of the sequence. This is clear in the examples presented in [95], in which the objects show little variation in brightness (with the exception of regions with occlusion), so that the data penalty component generates a good approximation for the object. In this work, we focus on the more formal aspects of shape constraints, avoiding the usage of these heuristic procedures. Here, we only fix the translation based on the object seeds and study the impact of template positioning errors for the different methods, being the treatment of rotations left for future works. Once we obtain a predefined cost map externally, we can for instance center its zero-valued pixel with the center of mass of the object seeds and execute Algorithm 4 normally.



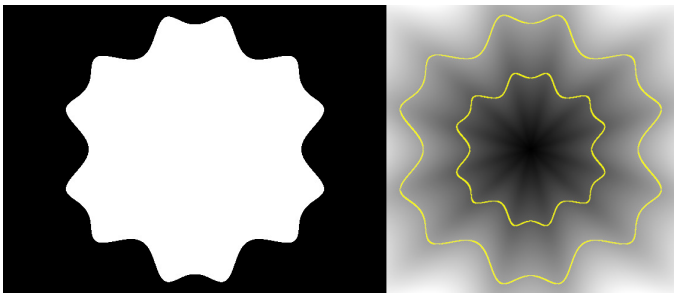
(a) $A = 200$, $B = 100$, $n_1 = 2$, $n_2 = 2$, $n_3 = 2$, $m = 4$



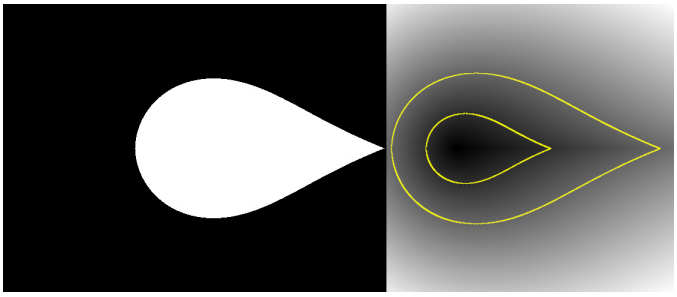
(b) $A = 350$, $B = 100$, $n_1 = 1.1$, $n_2 = 1$, $n_3 = 1.3$, $m = 6$



(c) $A = 3000$, $B = 1000$, $n_1 = 3$, $n_2 = 2$, $n_3 = 2$, $m = 10$



(d) $A = 500$, $B = 3000$, $n_1 = 6$, $n_2 = 5$, $n_3 = 4$, $m = 12$



(e) $A = 100000$, $B = 10000$, $n_1 = 2$, $n_2 = 1$, $n_3 = 1$, $m = 2$

Figure 3.6: *Shape templates by Gielis geometric equation.*

3.3 Experimental results

In this section, we compare LB with shape constraints commonly employed in graph-based segmentation: Geodesic Star Convexity [37], Boundary Band [5], and Hedgehog Shape Prior [41, 40]. We opted to compare them using Max-Min optimizers because BB is not yet supported by Min-Sum optimizers [5].

Regarding Local Boundary Band (LBB) from Definition 1, we still don't have an implementation to optimize a segmentation subject to this constraint. However, its limiting case BB, as $R \to \infty$, is included in the experiments. The results of the Band constraint from Definition 4 (Algorithm 6), which results from the limiting case of LB_{Δ}^{R} as $R \to \infty$, are not shown, as they are practically redundant in relation to those obtained by BB, given the theoretical similarities between these approaches as presented at the end of Section 3.2.1.

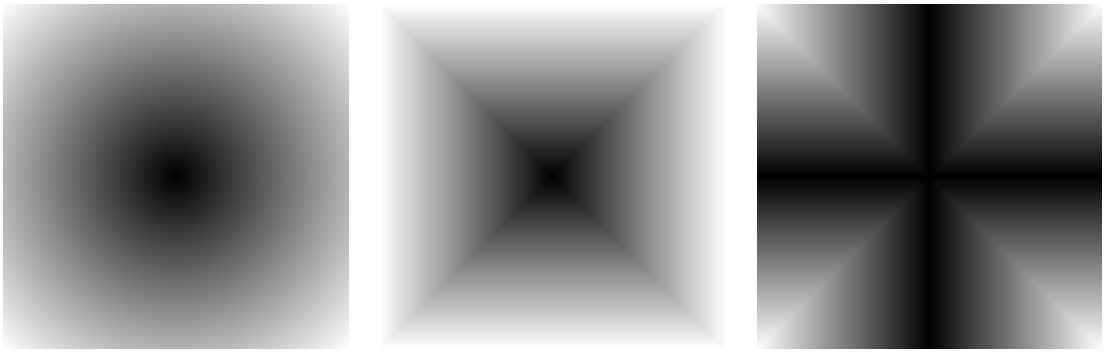
3.3.1 Shape Templates

In this section, we present the results of the methods using the shape templates shown on Figure 3.7 as the reference cost map. Figures 3.7a-b show the simpler templates used for describing circles and squares. Figure 3.7c shows a template used to describe cross-shaped objects, but without fixing the proportion of the cross dimensions (i.e., the ratio of the width of the cross arm to its height) in order to increase its flexibility. In Figure 3.7d, the template of a vertically oriented ellipse with an eccentricity value of 0.617 is shown. The template of Figure 3.7e is the same from Figure 3.6e, which was obtained by Algorithm 8 to describe the natural shape of a leaf by Gielis equation.

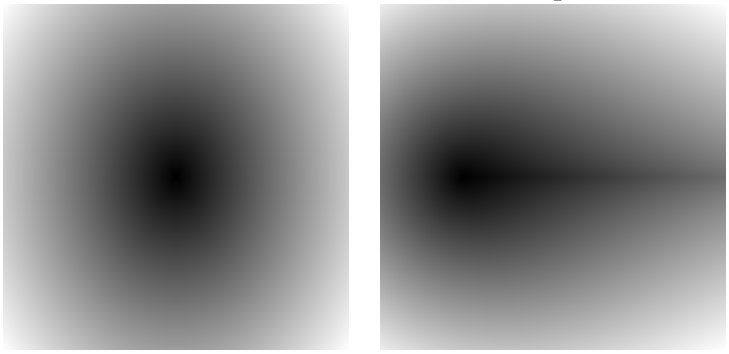
Figure 3.8 shows the results of coin segmentation by OIFT using different methods, $\omega(s,t) = \|\vec{I}(t) - \vec{I}(s)\|$ and a circle template, as reference cost map, centered on the center of mass of the internal seeds. The BB constraint fails to give good results compared to Local Band, due to its greater sensitivity to the template positioning.

Figure 3.9 shows some results of a tile segmentation using a square template and $\omega(s,t) = \|\vec{I}(t) - \vec{I}(s)\|$. In order to measure the sensitivity of the most promising methods for different seed positioning, in Figure 3.10 we show the accuracy curves using internal seeds in a circular brush of radius 5 pixels with horizontal displacements relative to the object's center and background seeds at the image frame. Note that, for the coin segmentation, LB ($R = 3.5$ and $\Delta = 2$) had slightly more stable results compared to Hedgehog, giving almost perfect results for 68.2% of the maximum possible horizontal shift in the coin (radius 44 pixels). BB constraint with $\Delta = 10$ had significantly lower robustness to seed displacements (11.4% of the maximum shift). For higher delta values, BB became unstable. It surprisingly had better results for a left-shifted position to avoid false positives on its right side. For the wall tile segmentation, LB ($R = 3.5$ and $\Delta = 2$) had the most accurate results, giving good results for 10.3% of the maximum possible horizontal shift in the wall tile (radius 145 pixels). BB constraint with $\Delta = 10$ had worse robustness to seed displacements. For higher values of delta, it was possible to increase its robustness, but at the price of sacrificing its accuracy.

Regarding the computational time, for the tile segmentation (Figure 3.9), the max-min



(a) Circle (b) Square (c) Cross



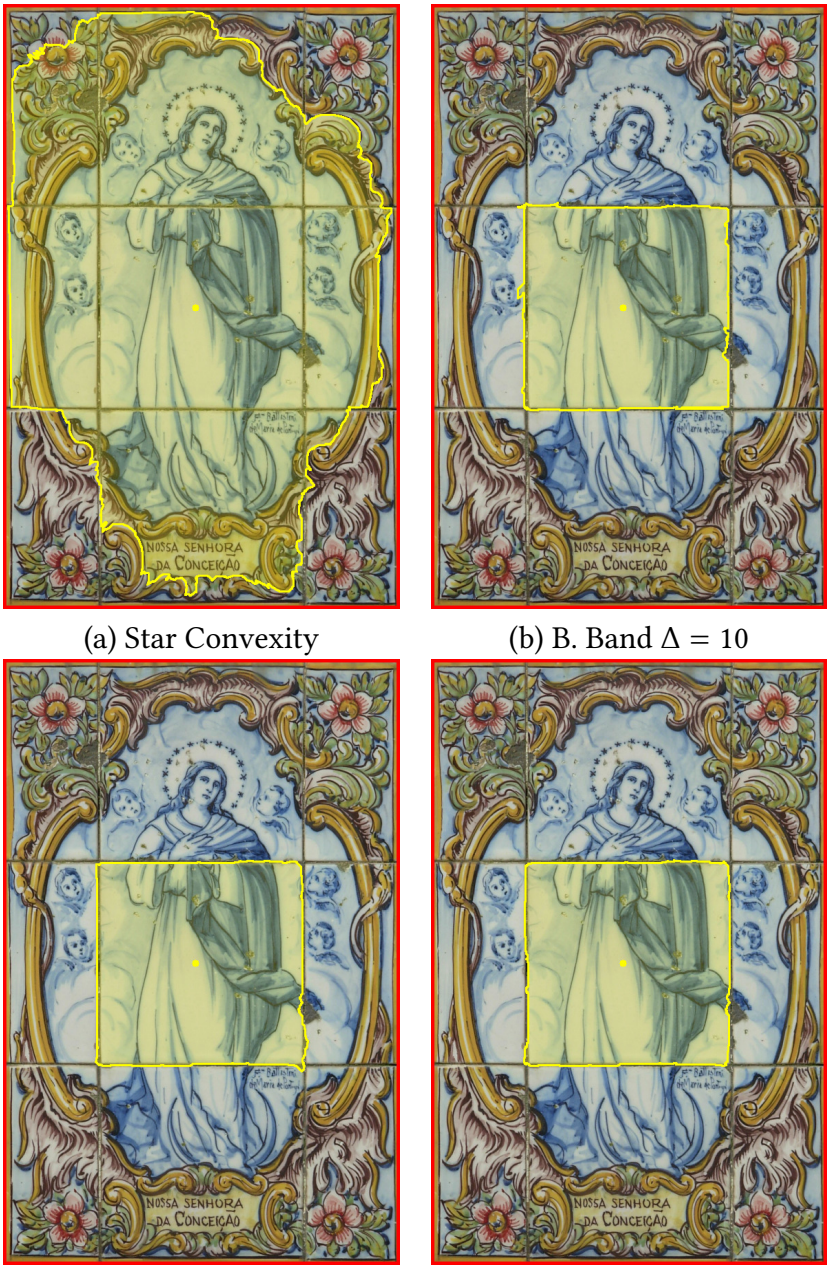
(d) Ellipse (e) Leaf

Figure 3.7: *Some shape templates used for segmentation.*

(a) Star Convexity (b) B. Band $\Delta = 10$

(c) Hedgehog $\theta = 45^{\circ}$ (d) Local Band $\Delta = 2$

Figure 3.8: *Coin segmentation by OIFT with a circle template (Fig. 3.7 (a)) in a* 250×185 *image.*



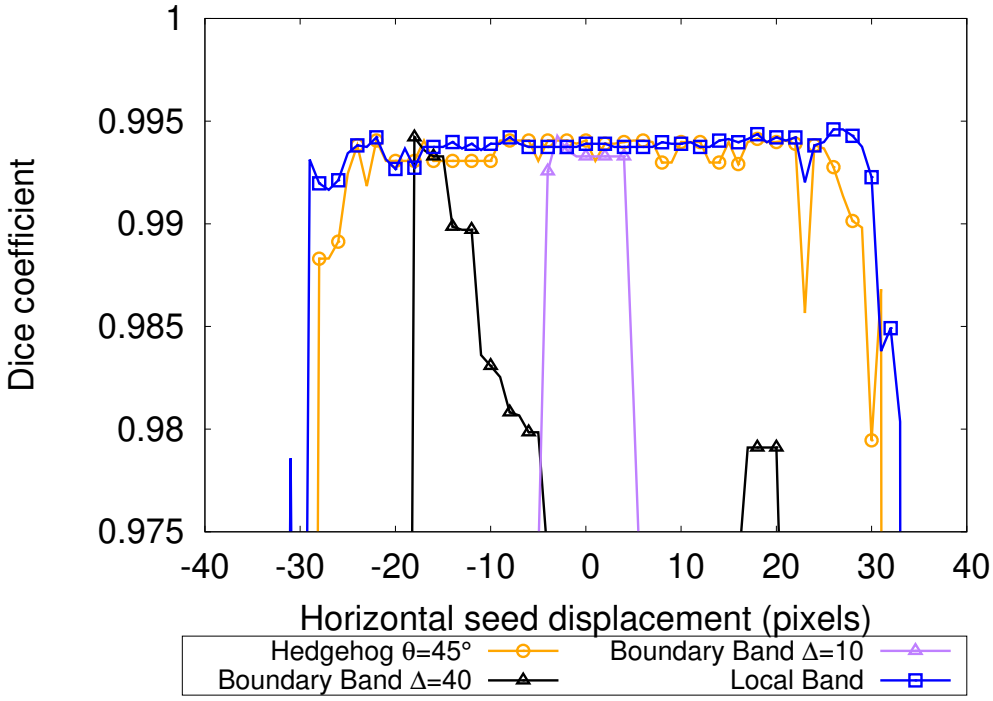
(a) Star Convexity (b) B. Band $\Delta = 10$

(c) Hedgehog $\theta = 45^{\circ}$ (d) Local Band $\Delta = 2$

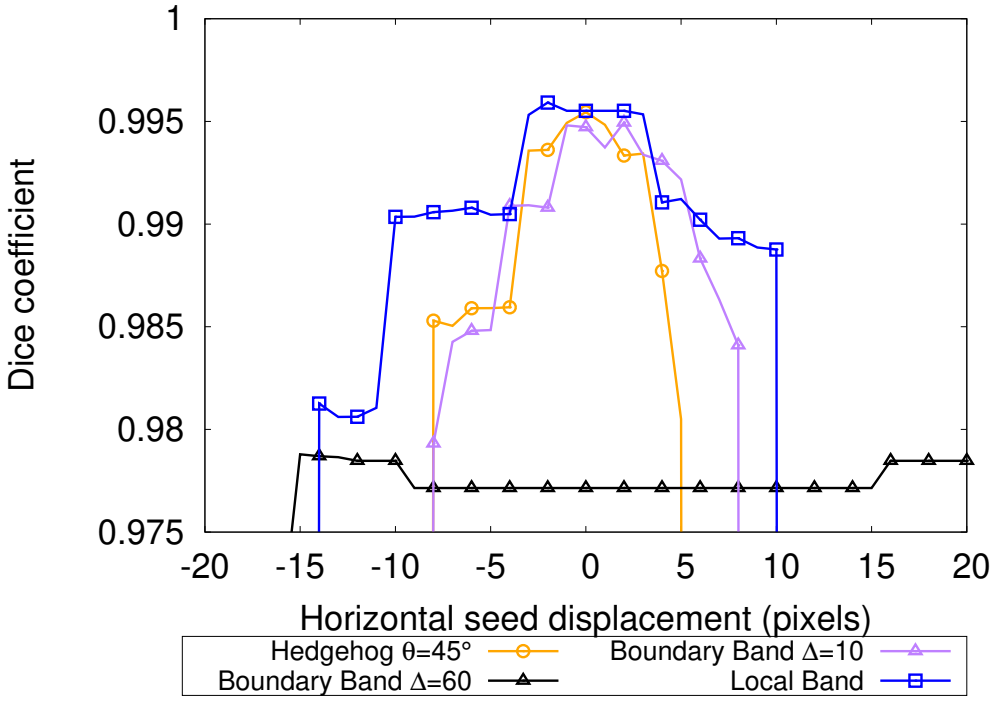
Figure 3.9: *Wall tile segmentation by OIFT with a square template (Fig. 3.7 (b)) in a* 576×881 *image.*

optimization on G' with LB ($R = 3.5$ and $\Delta = 2$) took 147.6 ms running on an Intel Core i3-5005U CPU 2.00GHz×4. In relation to the number of arcs, we have $|\mathcal{A}| \approx 4 \times 10^6$ for G and $|\mathcal{A}'| \approx 12 \times 10^6$ for G' in Algorithm 4. Tables 3.1 and 3.2 show the effect of increasing the radius size on the computational time of the max-min optimization on G' and the resulting number of arcs of G', respectively. Note that the computational cost increases considerably for larger radius values, since its computational complexity is $O(|\mathcal{A}'| + |\hat{\mathcal{N}}| \times K)$, when the finite arc weights are integers in a small interval of size K. The nodes $\hat{\mathcal{N}}$ and the set of arcs

$$\{\langle p, q\rangle \in \mathcal{A}' : \|p - q\| \leq R \;\&\; C(p) \geq C(q) + \Delta\}$$



(a) coin



(b) wall tile

Figure 3.10: *The accuracy curves for different horizontal displacements of the internal seeds.*

from Line 2 of Algorithm 4 form a *Directed Acyclic Graph* (DAG). As discussed earlier in Section 3.2.3, several of these $-\infty$-valued arcs in ω' that represent the LB constraint may actually be redundant and we can apply a *transitive reduction* [1] to eliminate them from

	Time (ms) without TR	Time (ms) with TR	percent gain
$\Delta = 1, R = 6$	356.2	116.6	67.3%
$\Delta = 1, R = 8$	581.8	175.7	69.8%
$\Delta = 1, R = 10$	886.5	186.8	78.9%
$\Delta = 1, R = 12$	1,438.8	218.4	84.8%
$\Delta = 2, R = 6$	329.0	248.6	24.4%
$\Delta = 2, R = 8$	521.2	295.7	43.3%
$\Delta = 2, R = 10$	823.7	340.8	58.6%
$\Delta = 2, R = 12$	1,429.7	361.8	74.7%

Table 3.1: *The computational time of Algorithm 1 on G', without and with Transitive Reduction (TR), for the tile segmentation (Figure 3.9), for different* LB *parameters on an Intel Core i3-5005U CPU 2.00GHz×4 and the percentage gain of using TR.*

	$\|\mathcal{A}'\|$ without TR	$\|\mathcal{A}'\|$ with TR	percent gain
$\Delta = 1, R = 6$	51,342,944	12,189,774	76.3%
$\Delta = 1, R = 8$	91,391,300	16,260,890	82.2%
$\Delta = 1, R = 10$	149,234,880	20,349,830	86.4%
$\Delta = 1, R = 12$	208,879,992	24,420,058	88.3%
$\Delta = 2, R = 6$	43,204,080	26,245,554	39.3%
$\Delta = 2, R = 8$	79,181,320	34,360,266	56.6%
$\Delta = 2, R = 10$	132,935,960	42,510,658	68.0%
$\Delta = 2, R = 12$	188,510,844	50,623,754	73.1%

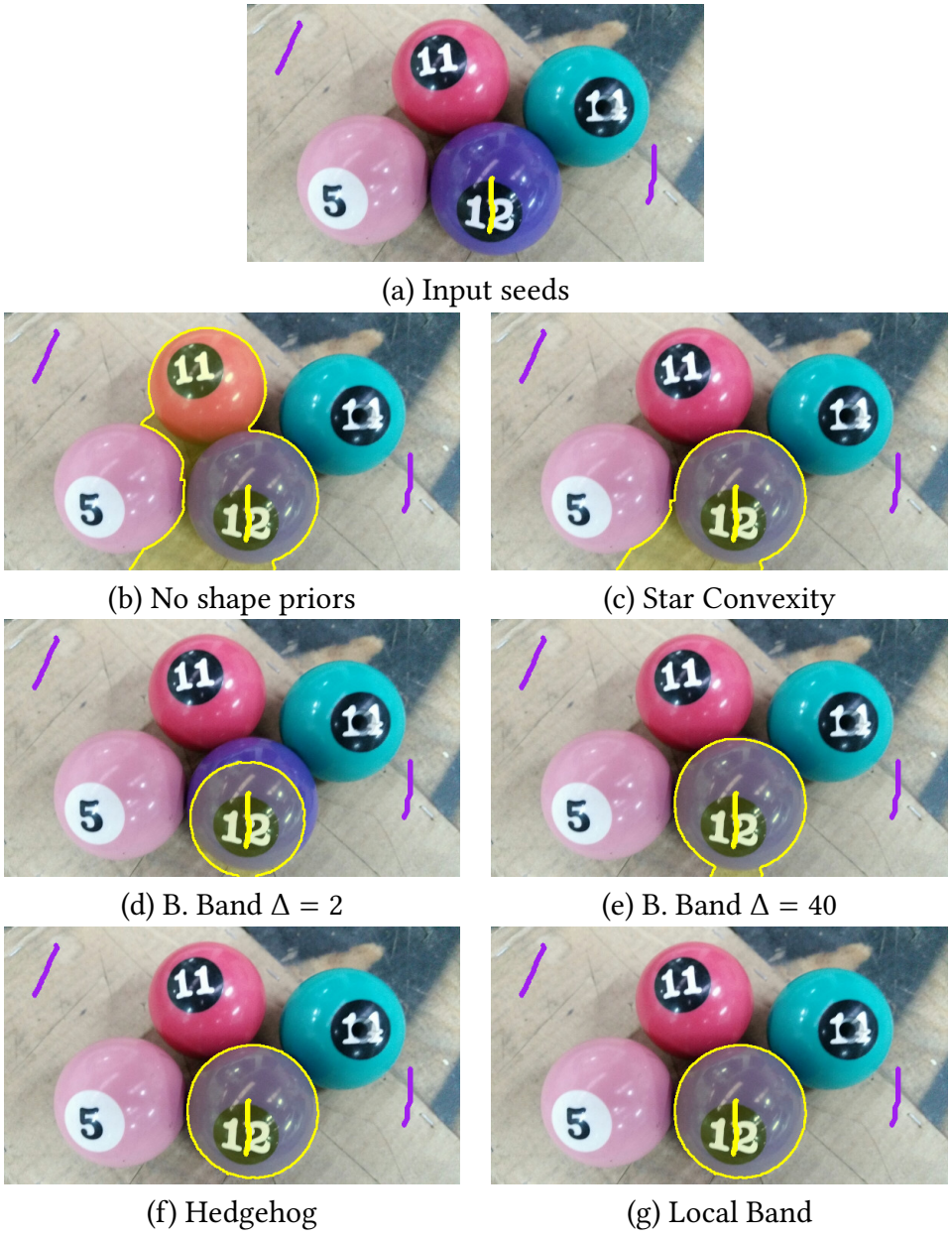
Table 3.2: *The number of arcs of G', without and with Transitive Reduction (TR), for the tile segmentation (Figure 3.9), for different* LB *parameters and the percentage gain of using TR.*

this DAG.

The redundant arcs from the DAG can be eliminated from $\mathcal{A}'$, as well as their opposite infinite weighted arcs from Line 5 of Algorithm 4. Whenever the cost map C is fixed (e.g., based on a prior template of shapes), the transitive reduction can be computed offline in advance to save computational time. The updated runtime on G' and the cardinality of $\mathcal{A}'$ after removing all redundant arcs are shown in the second columns of Tables 3.1 and 3.2.

Figures 3.11, 3.12, and 3.13 show other examples of segmentation results by OIFT using different shape constraints of circle, square and cross templates, respectively, placed on the center of mass of the internal seeds.

The way these templates are made and used can yield useful tools to segment a wide variety of similar shaped objects. One example is the cross shape template, that can be used to segment a wide variety of crosses, even when their crossing points are not at their centers of mass and their bars have different lengths, provided that the template is centered approximately at their crossing points (Figure 3.13). Another example is the usage of the circle shape template to segment regular hexagonal objects in any scale or rotation, by



(a) Input seeds

(b) No shape priors

(c) Star Convexity

(d) B. Band $\Delta = 2$

(e) B. Band $\Delta = 40$

(f) Hedgehog

(g) Local Band

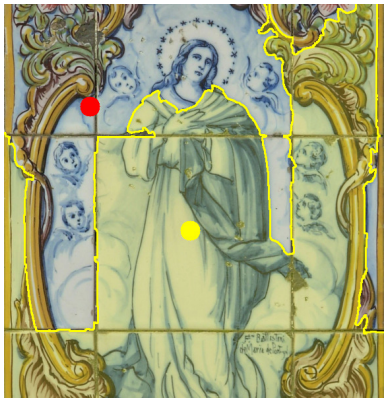
Figure 3.11: *Pool ball segmentation by OIFT with a circle template (Fig. 3.7 (a)).*

just increasing the deformation tolerance controlled by the parameter Δ and/or reducing the value of R (Figure 3.14).

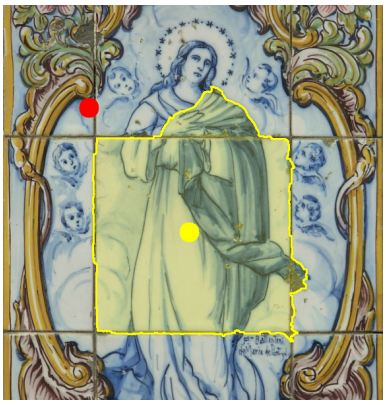
The Boundary Band constraint fails to give good results compared to Local Band and Hedgehog, due to its greater sensitivity to the template positioning. This problem is observable on Figures 3.14 and 3.15, where a minor shift of the internal seeds changes the positioning of the shape template. When the shape is well centered, the compared methods perform well, but we can observe that this is not the case when the template is moved away from the center, thus making the more sensitive methods more prone to fail.

Figure 3.16 shows the segmentation results by OIFT using different shape constraints

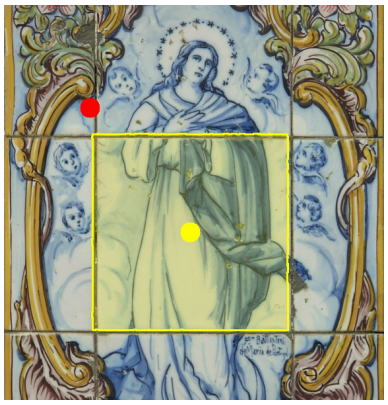
with the leaf template by Gielis equation from Figure 3.7e. The template is positioned such that its zero-valued pixel is aligned with the center of mass of the object seeds. In this example, the Boundary Band constraint required a high value of $\Delta = 160$ to be able to produce a reasonable result. Hedgehog failed to produce a minimally acceptable result for $\theta = 45^{\circ}$, requiring $\theta = 50^{\circ}$ to work. Both results indicate the instability of these methods in the selection of their parameters. On the other hand, the LB constraint obtained good results for its standard parameters of $R = 3.5$ and $\Delta = 2$.



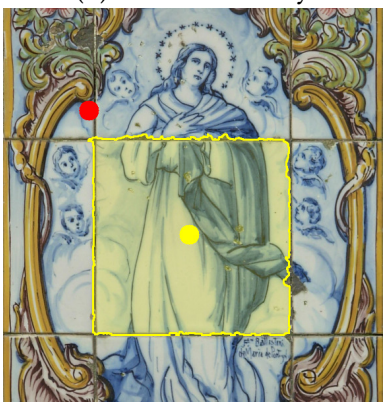
(a) No shape priors



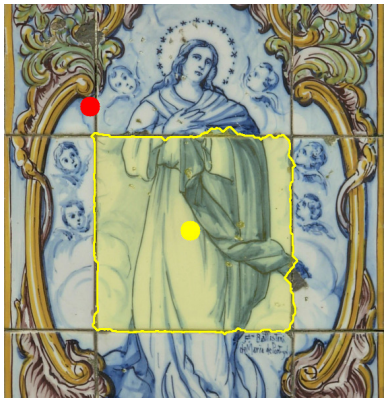
(b) Star Convexity



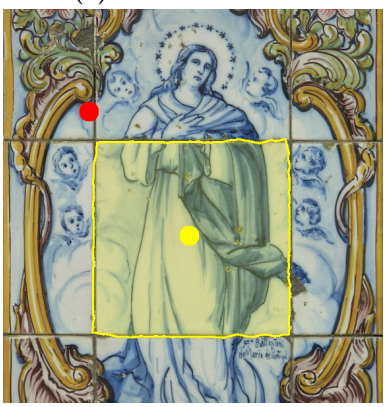
(c) B. Band $\Delta = 2$



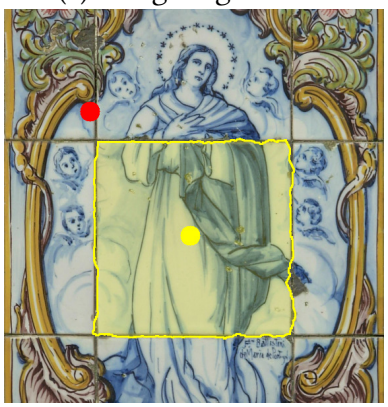
(d) B. Band $\Delta = 10$



(e) Hedgehog $\theta = 45^{\circ}$

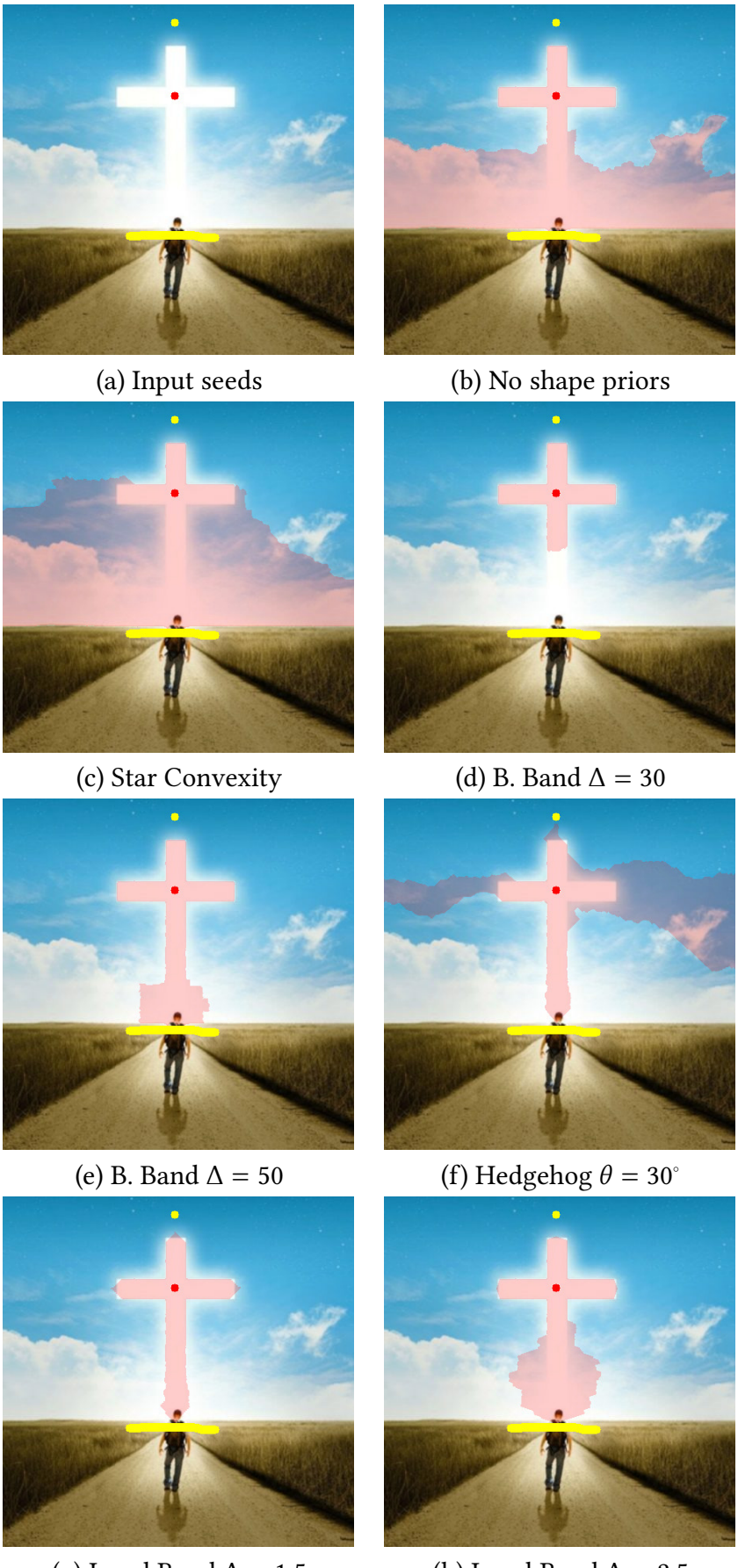


(f) Local Band $\Delta = 1$



(g) Local Band $\Delta = 2$

Figure 3.12: *Wall tile segmentation by OIFT with a square template (Fig. 3.7 (b)).*



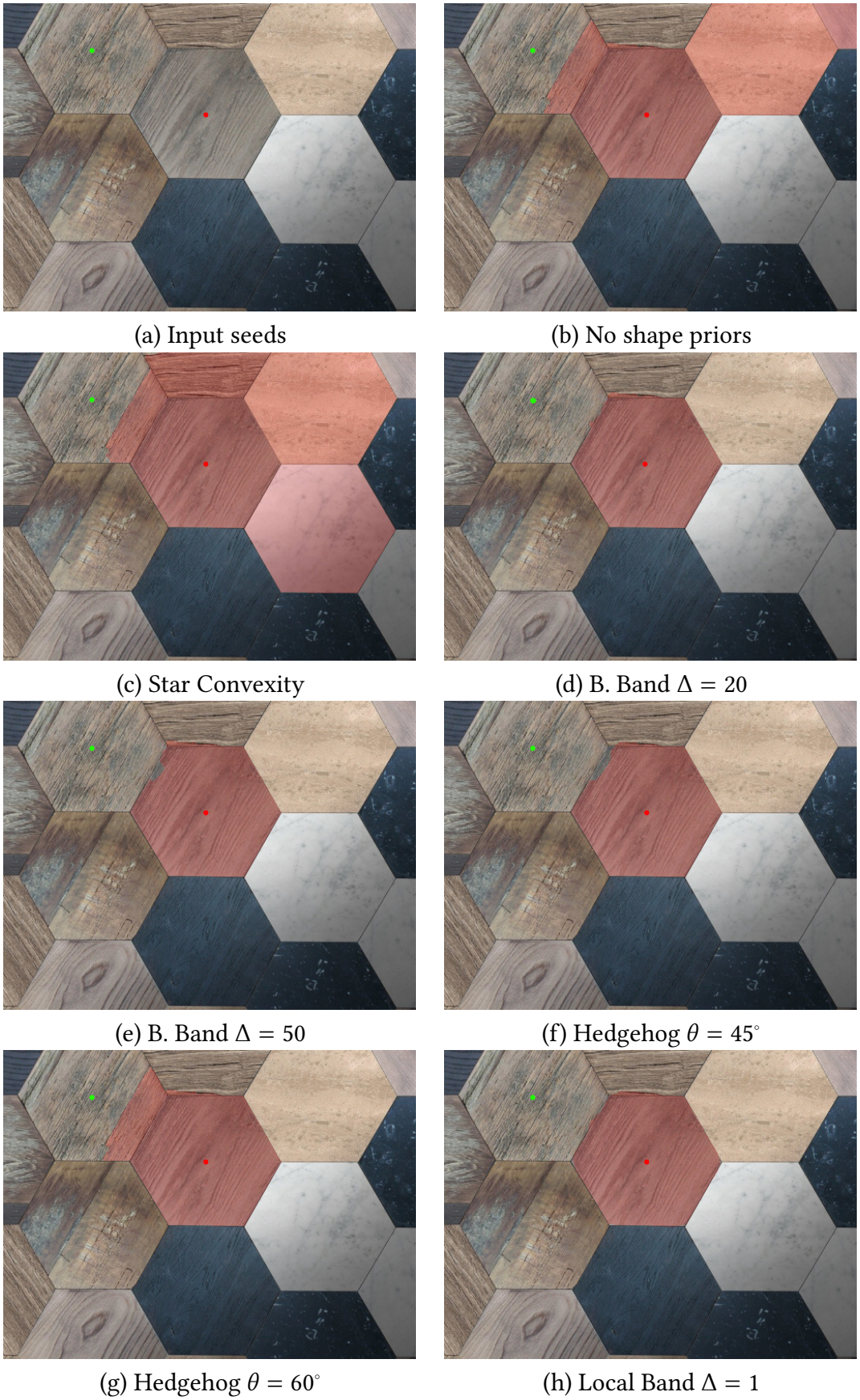
(a) Input seeds (b) No shape priors

(c) Star Convexity (d) B. Band $\Delta = 30$

(e) B. Band $\Delta = 50$ (f) Hedgehog $\theta = 30^{\circ}$

(g) Local Band $\Delta = 1.5$ (h) Local Band $\Delta = 2.5$

Figure 3.13: *Cross segmentation by OIFT with a cross template (Fig. 3.7 (c)).*



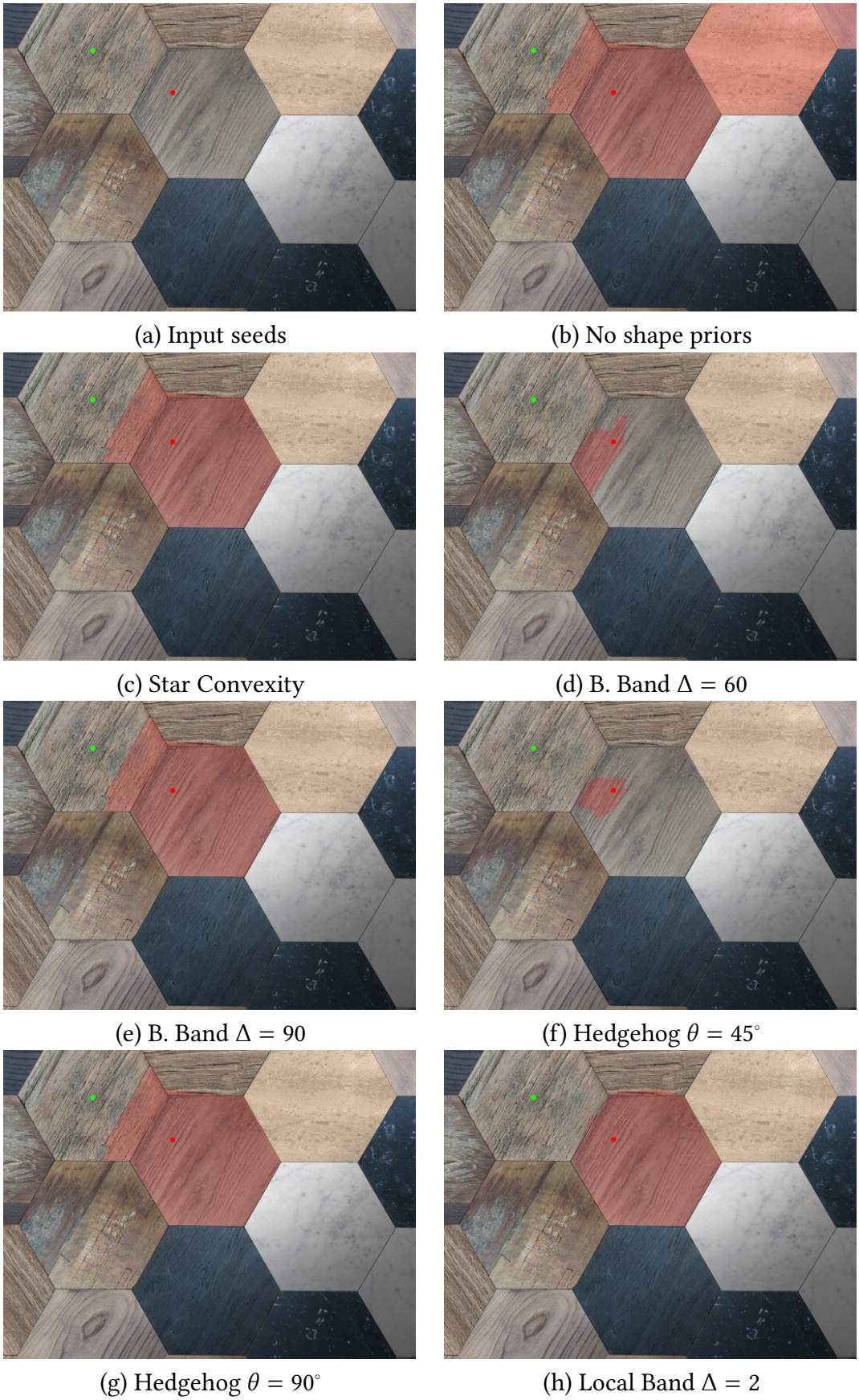
(a) Input seeds (b) No shape priors

(c) Star Convexity (d) B. Band $\Delta = 20$

(e) B. Band $\Delta = 50$ (f) Hedgehog $\theta = 45^{\circ}$

(g) Hedgehog $\theta = 60^{\circ}$ (h) Local Band $\Delta = 1$

Figure 3.14: *Hexagon floor tile segmentation by OIFT with a circle template positioned on the center of the tile.*



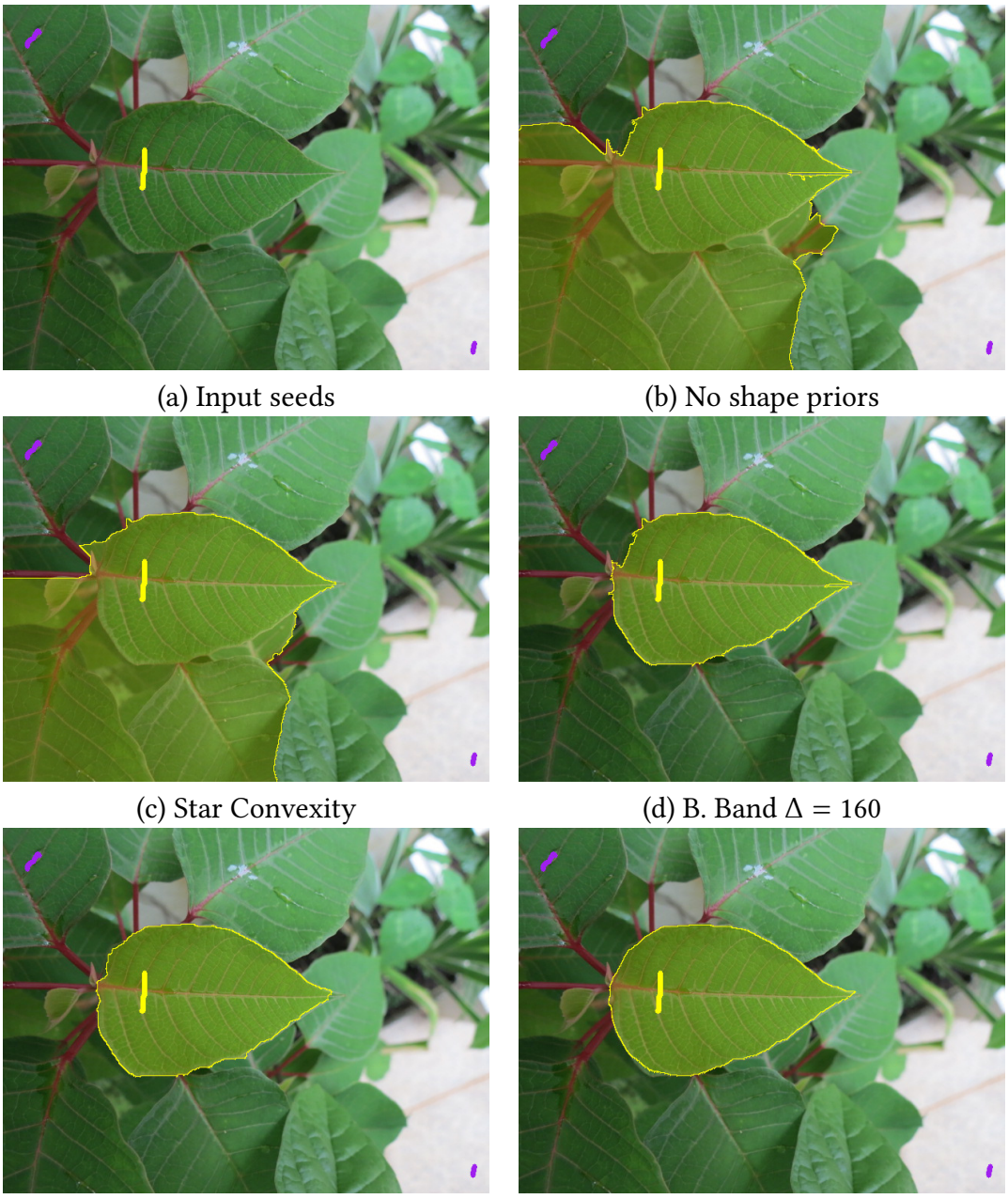
(a) Input seeds (b) No shape priors

(c) Star Convexity (d) B. Band $\Delta = 60$

(e) B. Band $\Delta = 90$ (f) Hedgehog $\theta = 45^{\circ}$

(g) Hedgehog $\theta = 90^{\circ}$ (h) Local Band $\Delta = 2$

Figure 3.15: *Hexagon floor tile segmentation by OIFT with a circle template positioned off the center of the tile.*



(a) Input seeds (b) No shape priors

(c) Star Convexity (d) B. Band $\Delta = 160$

(e) Hedgehog $\theta = 50^{\circ}$ (f) Local Band $\Delta = 2$

Figure 3.16: *Leaf segmentation by OIFT with the template by Gielis equation from Figure 3.7e positioned such that its zero-valued pixel is aligned with the center of mass of the object seeds.*

We also conducted experiments with the elliptical template from Figure 3.7d, using the central slices of 3D magnetic resonance imaging (MRI) scans of the brain (Figure 3.17). We used a dataset composed of MRI-T1 volumes acquired with a 1.5T Elscint scanner at a voxel size of $0.98 \times 0.98 \times 1.00mm^3$ of 106 normal subjects from both genders, in the age range from 17 to 65 years.

We considered the task of classifying the slices in the sagittal, coronal or axial categories automatically, given that the volumes considered did not have this information available in their image header. As the sagittal slice has neither horizontal nor vertical symmetry in relation to its central point (indicated by the green cross in Figure 3.17a), it can be easily identified as it does not pass a symmetry test. So the problem comes down to deciding between the two remaining slices, as coronal or axial. Since several boundaries present in an axial slice have an approximately elliptical shape (Figure 3.17b), a segmentation subject to a shape constraint by the template given in Figure 3.7d will be more successful in an axial slice when compared to a coronal slice (Figure 3.17c). Therefore, we expect the energy $\varepsilon_{\min}$, given by Equation 2.3 w.r.t. G, of a segmentation of an axial slice to be greater than that of a coronal slice.

Figure 3.18 shows some segmentation results of axial and coronal slices by OIFT subject to the LB constraint with the elliptical template. We used $\omega(s,t) = |I(t) - I(s)|$ and $\alpha = 0.5$ for the boundary polarity. Table 3.3 presents the average energy values of $\varepsilon_{\min}$ for axial and coronal slices using Local Band and Hedgehog constraints with the elliptical template and the corresponding percentage of volumes with their correct slicing identification. Local Band obtained maximum accuracy, misidentifying a single volume, and having the largest energy separation interval $\Delta\bar{\varepsilon}_{\min}$ between the mean energies of axial and coronal slices. For each volume, we considered as the classification rule for the identification of the slicing orientation, the assignment of the axial slice as the one with the highest energy between the two remaining slice planes. Table 3.3 also shows the results without the use of any shape constraints. Clearly, in this case the OIFT energy does not show any clear preference in favor of any particular slicing orientation, achieving a 48.11% success rate, which is no better than random guess. On the other hand, the energies with the usage of elliptical constraints clearly showed a strong indication of the orientation of the volume slices. Therefore, we have that the OIFT energy accompanied by appropriate constraints can be used for the purpose of feature extraction.

	$\bar{\varepsilon}_{\min}$ **(axial)**	$\bar{\varepsilon}_{\min}$ **(coronal)**	$\Delta\bar{\varepsilon}_{\min}$	**Success rate of classification**
Local Band $\Delta = 2$	54.73	18.98	35.75	105/106 = 99.06%
Hedgehog $\theta = 40^\circ$	50.55	17.92	32.63	105/106 = 99.06%
Hedgehog $\theta = 45^\circ$	53.02	19.38	33.64	105/106 = 99.06%
Hedgehog $\theta = 50^\circ$	57.54	24.22	33.32	103/106 = 97.17%
No shape priors	79.15	77.56	1.59	51/106 = 48.11%

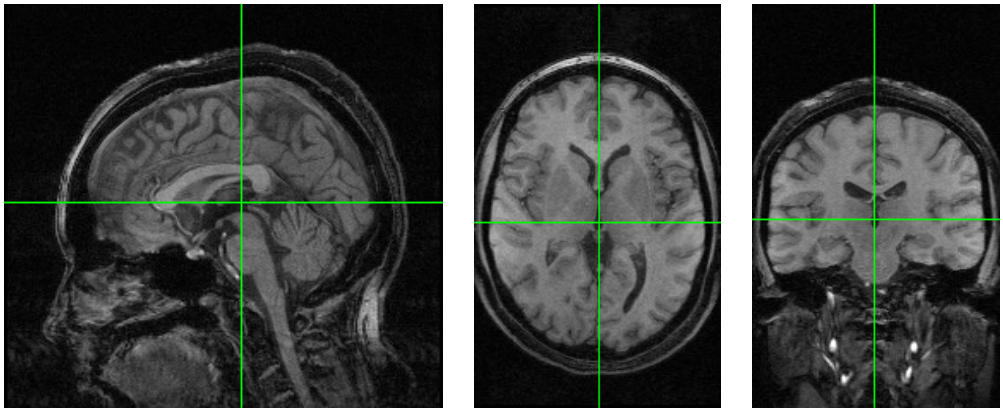
Table 3.3: *Average energy values for axial and coronal slices for different shape constraints and the percentage of volumes with their correct slicing identification for the 106 MRI-T1 volumes of 1.5 Tesla.*

We also repeated these experiments now using a second dataset of 3 Tesla MRI composed of 10 volumes, reaching similar conclusions (Table 3.4). In this case, the result with

no shape priors obtained a low accuracy of only 80%, but which is much higher than the previously obtained result of 48.11% from Table 3.3. This variation is mainly due to the small size of the second dataset. Although these results from Tables 3.3 and 3.4 involve 3D volumes, only 2D segmentations of the central slices were performed. Actual 3D experiments will be left for Section 3.3.3.

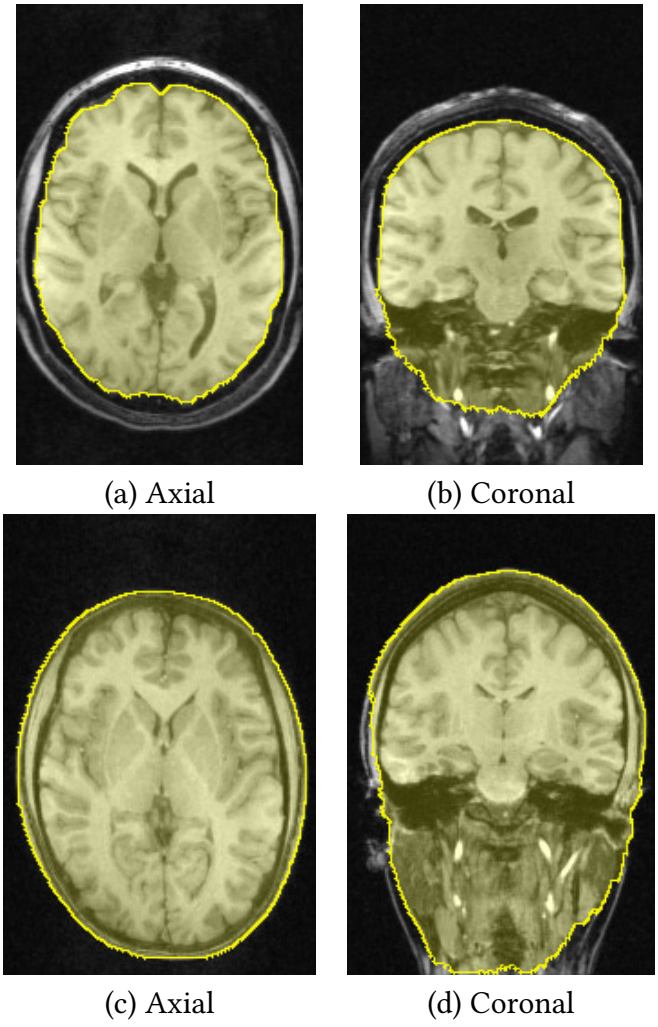
	$\bar{\varepsilon}_{\text{min}}$ **(axial)**	$\bar{\varepsilon}_{\text{min}}$ **(coronal)**	$\Delta\bar{\varepsilon}_{\text{min}}$	**Success rate of classification**
Local Band $\Delta = 2$	17.90	4.20	13.70	10/10 = 100.00%
Hedgehog $\theta = 40^{\circ}$	18.10	4.50	13.60	10/10 = 100.00%
Hedgehog $\theta = 45^{\circ}$	18.10	4.90	13.20	10/10 = 100.00%
Hedgehog $\theta = 50^{\circ}$	18.10	5.50	12.60	10/10 = 100.00%
No shape priors	18.40	13.60	4.80	8/10 = 80.00%

Table 3.4: *Average energy values for axial and coronal slices for different shape constraints and the percentage of volumes with their correct slicing identification for the 10 MRI-T1 volumes of 3 Tesla.*



(a) Sagittal (b) Axial (c) Coronal

Figure 3.17: *Sagittal, axial and coronal central slices of a 3D magnetic resonance imaging (MRI) scan of the brain.*



(a) Axial (b) Coronal

(c) Axial (d) Coronal

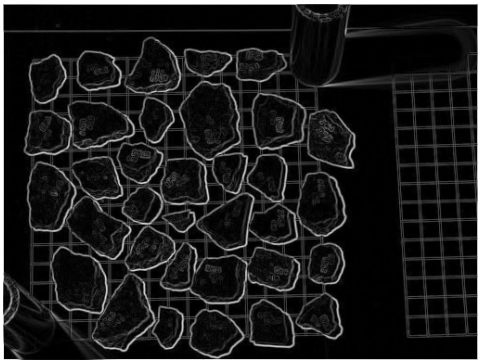
Figure 3.18: *Segmentation results of brain slices by OIFT subject to the* LB *constraint ($R = 3.5$ and $\Delta = 2$) with the elliptical template given in Figure 3.7d. Note that the segmentation with elliptical constraint of coronal slices only partially matches the edges of the structures present in the image, leading to lower energy values.*

3.3.2 Cost map as the geodesic length

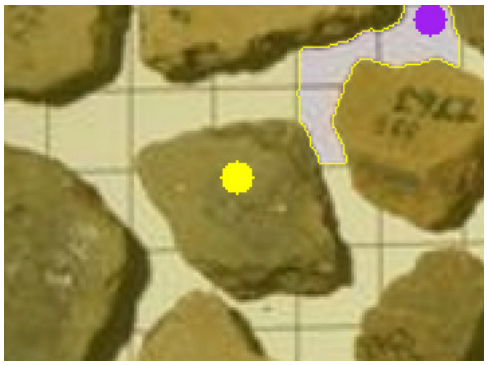
From the IFT [26] perspective, when the cost map C is the geodesic length (i.e., $\psi_{sum}(\langle v_0, \ldots, v_\ell \rangle) := \sum_{1 \leq j \leq \ell} \|v_{j-1} - v_j\|$), from $\mathcal{S}_1$ in $G = (\mathcal{N}, \mathcal{A})$, the previous constraints are based on different attributes of a previously computed minimal forest in G rooted at $\mathcal{S}_1$: Geodesic Star Convexity uses the predecessor map [57], BB and LB constraints exploit the cost map directly, and Hedgehog uses the gradient of the cost map as vector field.

We tested the robustness of the methods in relation to different image resolutions by quantitative experiments, to segment archaeological fragments in seven different resolutions with the geodesic cost. In order to make the experiment more challenging, the simple arc weight $\omega(s,t) = G(s) + G(t)$ was used, disregarding any prior color information, where $G(t)$ denotes the magnitude of Sobel gradient, such that we have several false boundaries (Figure 3.19). Figure 3.20 shows the mean values of the Dice coefficient for segmenting ten fragments for each image resolution, totalizing 70 executions for each method. The overall best results were obtained by LB using $R = 3.5$ and $\Delta = 2$. Hedgehog for different θ values and the same radius presented unstable results (Figure 3.19d).

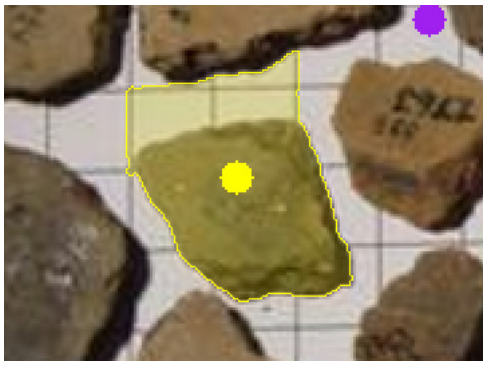
Finally, we conducted experiments with the geodesic cost to segment the liver in medical images of 40 slices of thoracic CT studies of size 512×512, using regular weights $\omega(s,t) = |I(t) - I(s)|$ and seed sets progressively obtained by eroding the ground truth and its background with twice the radius size (Figure 3.21). Although this scenario is apparently advantageous for the BB constraint, in view of the well-distributed and centralized seeds, LB ($R = 3.5$ and $\Delta = 2$) demonstrated good results with the highest accuracy for a large part of the curve (Figure 3.22a). We repeated the experiments, but now with the internal seeds shifted by 5 pixels to the left (25% of the maximum possible displacement in the central part of the curves) whenever possible. In this new scenario, the results clearly show that LB is more robust than BB in relation to seed positioning (Figure 3.22b). The performance of OIFT with LB is also superior to that revealed by classical techniques for the same image graph G and seeds (Figures 3.21d-f).



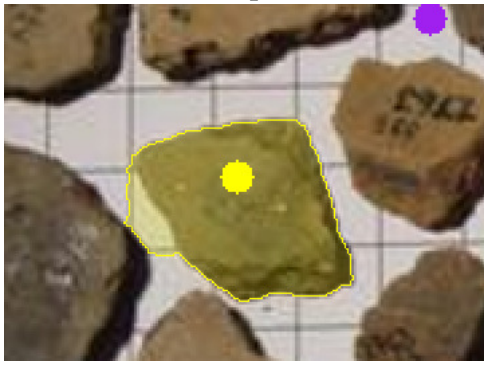
(a) Sobel gradient



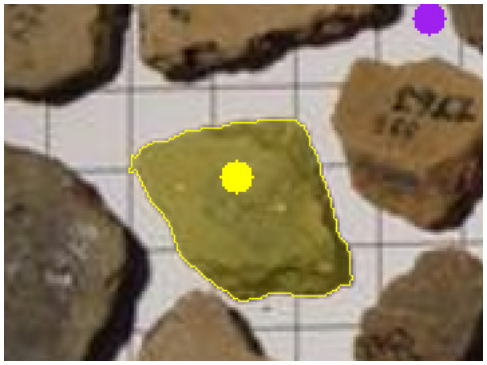
(b) No priors



(c) B. Band

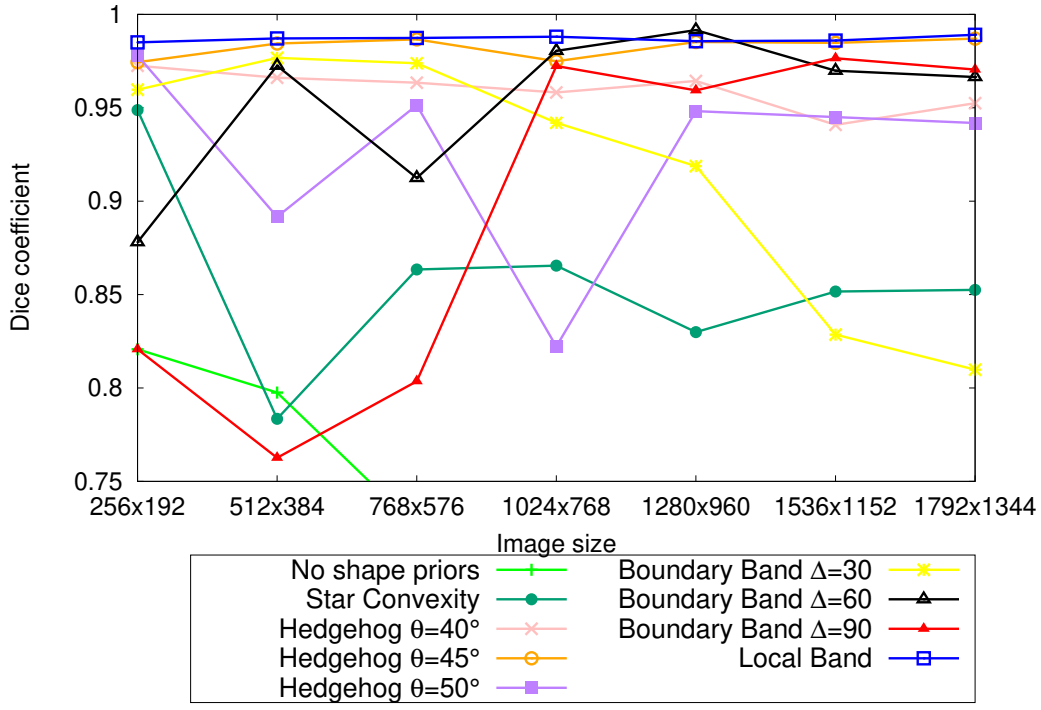


(d) Hedgehog

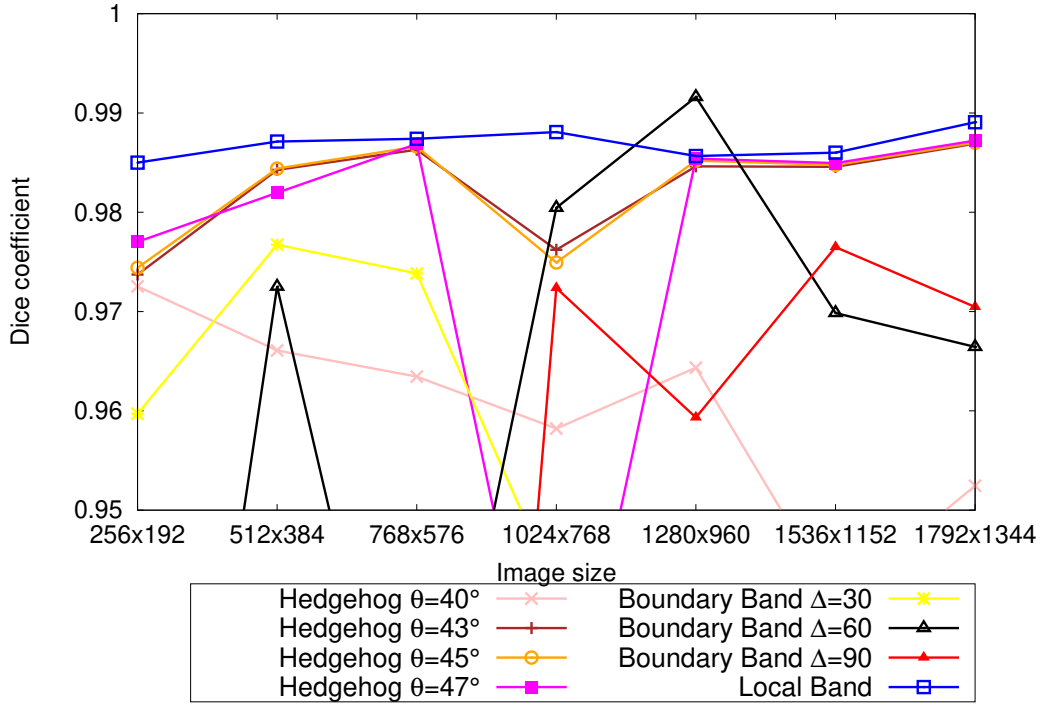


(e) Local Band

Figure 3.19: *Archaeological fragment segmentation*

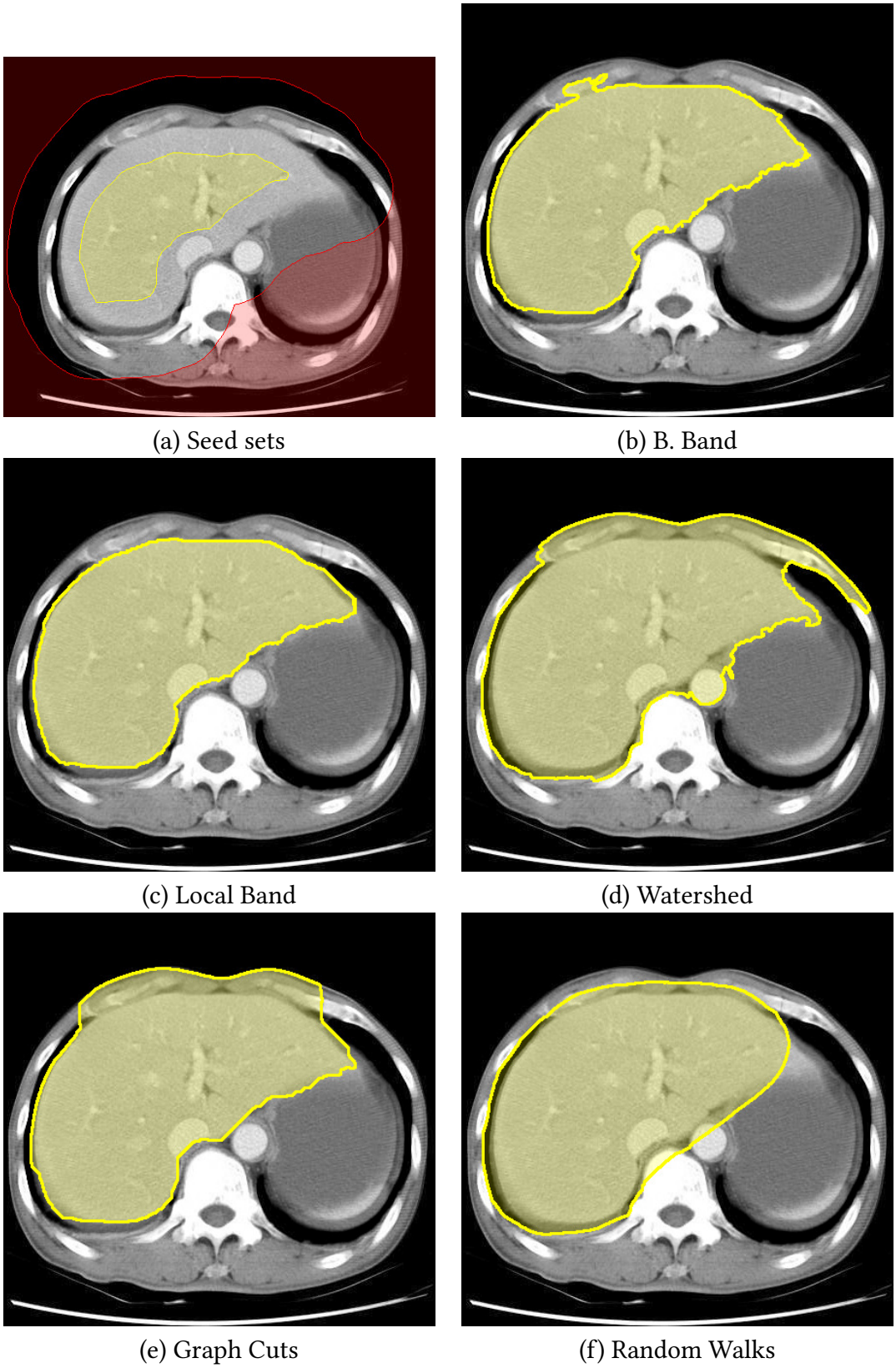


(a)



(b)

Figure 3.20: *(a) The mean accuracy values to segment the archaeological fragments for different image resolutions. (b) Zoomed results (accuracy* $\geq 95\%$*).*

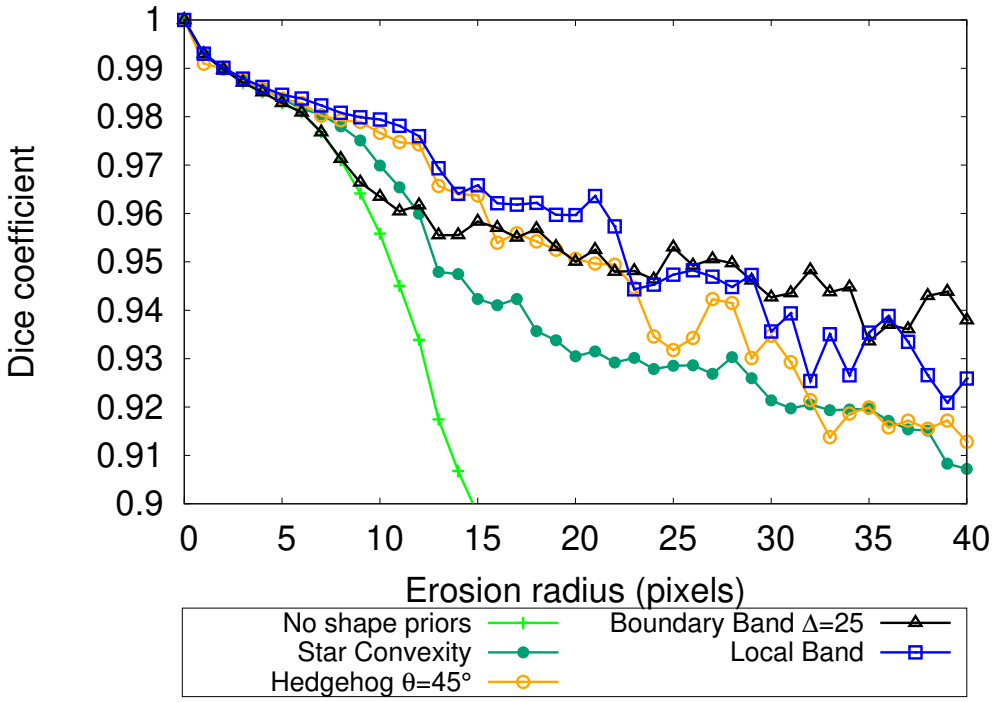


(a) Seed sets (b) B. Band

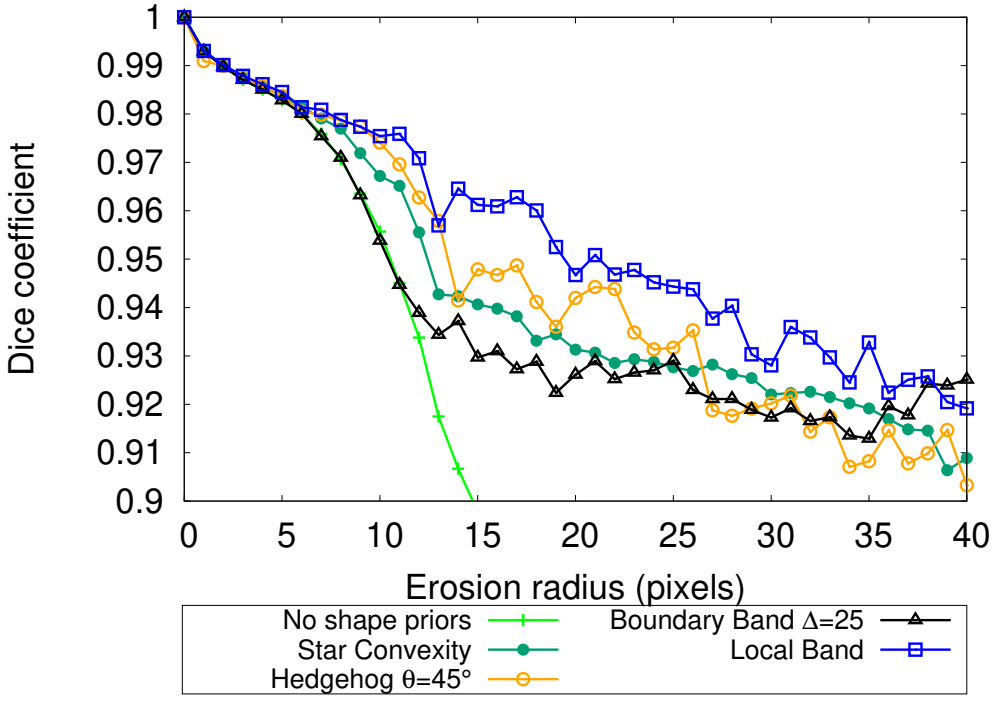
(c) Local Band (d) Watershed

(e) Graph Cuts (f) Random Walks

Figure 3.21: *A thoracic CT image of* 512×512 *pixels. (a) Seed sets obtained by eroding the ground truth of the liver (in yellow) and its background with twice the radius size (in red). (b-c) Liver segmentation samples by OIFT subject to different shape constraints. (d-f) Segmentation results by classical techniques for the given image graph and seeds.*



(a) S_1 centralized



(b) S_1 shifted to the left

Figure 3.22: *The mean accuracy curves to segment the liver for seed sets obtained by erosion.*

3.3.3 Examples in three-dimensional images

In this section we present some examples of the proposed LB shape constraint in synthetic volumes and a real 3D magnetic resonance imaging of the brain. In these examples, we considered the cost map as the Euclidean Distance Transform (EDT, for short) from the internal seeds.

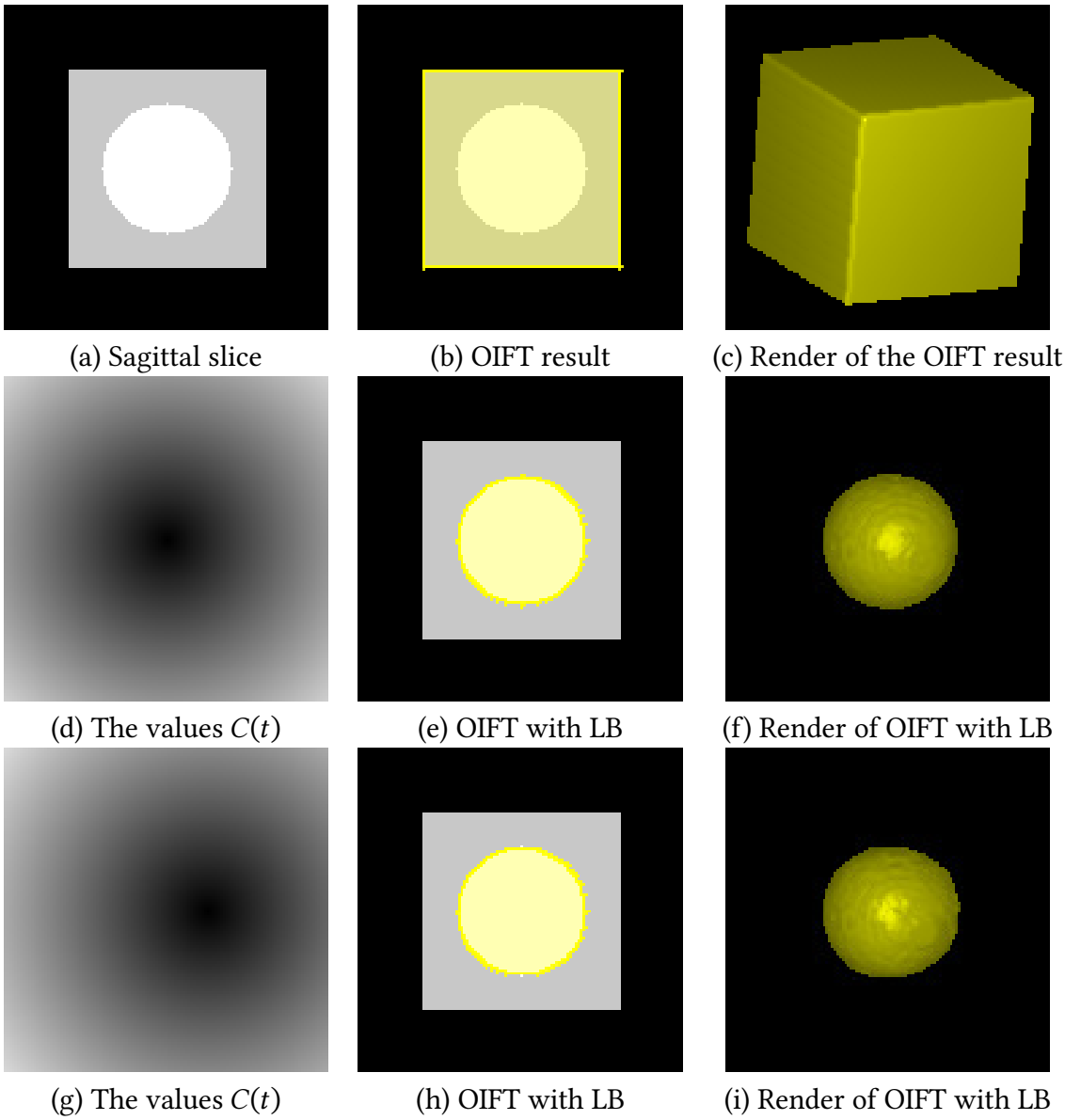
Figure 3.23 shows the segmentation of a synthetic volume of $120 \times 120 \times 120$ voxels containing a sphere of radius 24 and brightness 255 within a gray cube of intensity 200 and a black background (Figure 3.23a).

Considering a single object seed at the center of the sphere and a background seed at the $(0, 0, 0)$ position of the volume, the OIFT segmentation (Figures 3.23b-c) selects the cube since it has a stronger intensity transition on its boundary and it is not possible to differentiate the sphere from the cube by using the boundary polarity. The LB constraint favors the segmentation of the sphere, since the distance transform from a single point generates a spherical template (Figures 3.23d-f). In order to measure the robustness in relation to template placement, we repeated the segmentation by shifting the internal seed to the right. We observe that up to a displacement of 15 voxels (62.5% of the maximum internal displacement) we can still get the sphere segmentation (Figures 3.23g-i). The number of arcs $|\mathcal{A}'|$ of G' for the LB constraint with $\Delta = 1$ and $R = 2.5$ is 86,327,188, which can be reduced to 75,026,776 by employing the Transitive Reduction, while the original number of arcs $|\mathcal{A}|$ of G is 20,563,200. Regarding the computational time, Algorithm 1 on G has a computational time of 0.52 sec, while Algorithm 1 with LB on G' has 1.50 sec, which can be reduced to 1.36 sec by employing the Transitive Reduction, on an Intel Core i5-10210U CPU 1.60GHz×8.

To measure the impact of the volume size on the LB computation, Tables 3.5 and 3.6 show the computational time of Algorithm 1 with LB on G' and the number of arcs $|\mathcal{A}'|$ of G' for the LB parameters $\Delta = 1$ and $R = 3$ by varying the size of the synthetic volume from $80 \times 80 \times 80$ voxels to $160 \times 160 \times 160$ voxels. On the other hand, Tables 3.7 and 3.8 present the results for variations in the constraint parameters (Δ and R) and with a fixed volume size of $50 \times 50 \times 50$ voxels. The gains of the transitive reduction are more accentuated for low values of Δ and high values of R, as expected.

	Time (ms) without TR	**Time (ms) with TR**	**percent gain**
$80 \times 80 \times 80$	577.4	492.1	14.8%
$100 \times 100 \times 100$	1,141.2	974.6	14.6%
$120 \times 120 \times 120$	1,992.9	1,680.6	15.7%
$140 \times 140 \times 140$	3,201.7	2,809.4	12.3%
$160 \times 160 \times 160$	4,854.3	4,056.8	16.4%

Table 3.5: *The computational time of Algorithm 1 on G' for the* LB *parameters $\Delta = 1$ and $R = 3$, without and with Transitive Reduction (TR), for the segmentation of synthetic 3D volumes with different volume sizes in voxels (similar to Figure 3.23) on an Intel Core i5-10210U CPU 1.60GHz×8 and the percentage gain of using TR.*



(a) Sagittal slice (b) OIFT result (c) Render of the OIFT result

(d) The values $C(t)$ (e) OIFT with LB (f) Render of OIFT with LB

(g) The values $C(t)$ (h) OIFT with LB (i) Render of OIFT with LB

Figure 3.23: *Example of a three-dimensional image using a synthetic volume of a sphere of radius 24 and brightness 255 within a gray cube of intensity 200, with* $120 \times 120 \times 120$ *voxels. (a) Sagittal slice of the synthetic volume. (b-c) OIFT segmentation result and its 3D rendering. (d) The cost map as the Euclidean Distance Transform from the internal seed. (e-f) The OIFT subject to the* LB_{Δ}^{R} *with* $\Delta = 1$ *and* $R = 2.5$*. (g-i) The updated cost map and result by OIFT with the* LB *constraint (*$\Delta = 1$ *and* $R = 2.5$*) after shifting the internal seed to the right by a displacement of 15 voxels. Note the robustness of the segmentation for seed displacements.*

	$\lvert\mathcal{A}'\rvert$ **without TR**	$\lvert\mathcal{A}'\rvert$ **with TR**	**percent gain**
$80 \times 80 \times 80$	38,664,908	28,687,448	25.8%
$100 \times 100 \times 100$	76,231,392	56,548,236	25.8%
$120 \times 120 \times 120$	132,574,216	98,355,352	25.8%
$140 \times 140 \times 140$	211,473,716	156,905,732	25.8%
$160 \times 160 \times 160$	316,746,204	235,003,512	25.8%

Table 3.6: *The number of arcs of G' for the* LB *parameters $\Delta = 1$ and $R = 3$, without and with Transitive Reduction (TR), for the segmentation of synthetic 3D volumes with different volume sizes in voxels (similar to Figure 3.23) and the percentage gain of using TR.*

	Time (ms) without TR	**Time (ms) with TR**	**percent gain**
$\Delta = 1, R = 4$	244.8	183.4	25.1%
$\Delta = 1, R = 5$	443.4	301.4	32.0%
$\Delta = 1, R = 6$	756.7	351.4	53.6%
$\Delta = 1, R = 7$	1,010.8	429.9	57.5%
$\Delta = 2, R = 4$	148.9	141.9	4.7%
$\Delta = 2, R = 5$	294.9	273.1	7.4%
$\Delta = 2, R = 6$	522.6	403.7	22.8%
$\Delta = 2, R = 7$	830.7	560.5	32.5%

Table 3.7: *The computational time of Algorithm 1 on G', without and with Transitive Reduction (TR), for the segmentation of a synthetic 3D volume of $50 \times 50 \times 50$ voxels (similar to Figure 3.23), for different* LB *parameters on an Intel Core i5-10210U CPU 1.60GHz×8 and the percentage gain of using TR.*

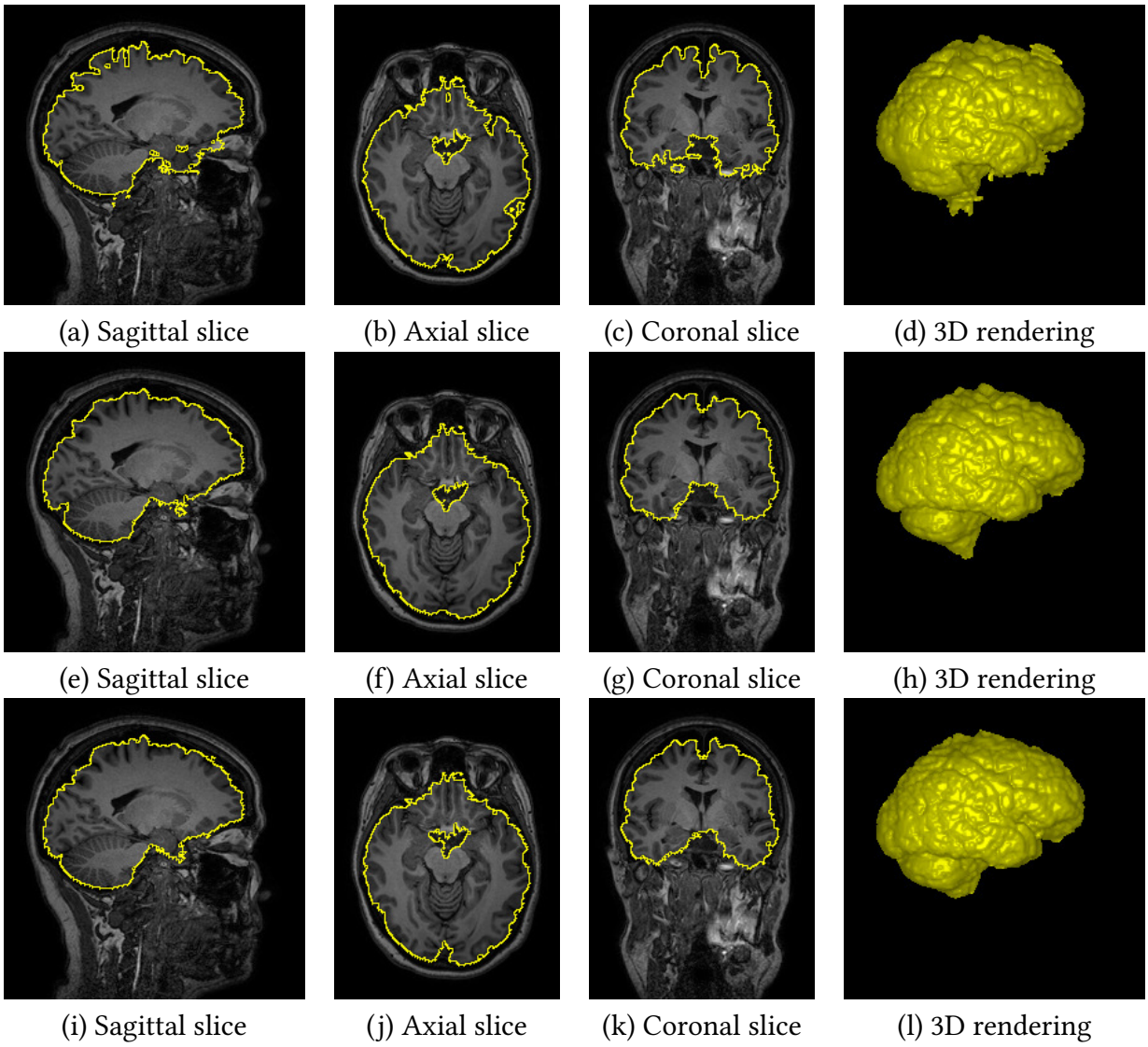
	$\lvert\mathcal{A}'\rvert$ **without TR**	$\lvert\mathcal{A}'\rvert$ **with TR**	**percent gain**
$\Delta = 1, R = 4$	20,055,390	11,185,682	44.2%
$\Delta = 1, R = 5$	41,905,614	17,657,468	57.9%
$\Delta = 1, R = 6$	77,109,568	25,866,678	66.5%
$\Delta = 1, R = 7$	119,656,756	34,082,958	71.5%
$\Delta = 2, R = 4$	10,433,764	10,429,906	0.04%
$\Delta = 2, R = 5$	25,772,758	22,645,552	12.1%
$\Delta = 2, R = 6$	52,742,962	38,285,476	27.4%
$\Delta = 2, R = 7$	87,064,366	54,370,946	37.6%

Table 3.8: *The number of arcs of G', without and with Transitive Reduction (TR), for the segmentation of a synthetic 3D volume of $50 \times 50 \times 50$ voxels (similar to Figure 3.23), for different* LB *parameters and the percentage gain of using TR.*

Figure 3.24 shows an example of the proposed LB shape constraint in a real 3 Tesla MRI-T1 three-dimensional image with severe inhomogeneity problems to segment the brain (task known as skull stripping). The volume was acquired at a voxel size of $1.0 \times 1.0 \times 1.0$ mm^3 and with a size of $240 \times 240 \times 180$ voxels.

The brain is composed of gray matter (GM) and white matter (WM) tissues, and it is surrounded by cerebrospinal fluid (CSF). In order to attenuate the strong transitions from WM to GM, thus favoring transitions from GM to CSF, we first computed the square root $I'(t) = \sqrt{I(t)}$ of the image intensity values $I(t)$ and then computed the magnitude of the Sobel 3D gradient $G(t) = |\nabla I'(t)|$ of the resulting filtered volume. We then set the weights of the arcs as $\omega(s,t) = G(s) + G(t)$, using 6-neighbors for voxel adjacency. The seed sets were obtained by eroding the ground truth and its background by a radius of 15 voxels.

Figures 3.24a-d show the brain segmentation result by OIFT with boundary polarity favoring transitions from bright to dark voxels ($\alpha = 0.75$), having a Dice accuracy of 97.08% relative to the gold standard. Figures 3.24e-h show the result by OIFT with $\alpha = 0.75$ and subject to the LB_Δ^R with $\Delta = 3$ and $R = 4.0$, having a Dice accuracy of 97.57%. Figures 3.24i-l show the result by OIFT with $\alpha = 0.75$ and subject to the LB_Δ^R with $\Delta = 3.5$ and $R = 4.5$, having a Dice accuracy of 97.65%. In addition to the higher accuracy, it is clear that the LB constraint leads to results with more regular boundaries and fewer false positive errors, such as in the optic nerve region. The number of arcs $|\mathcal{A}'|$ of G' for the LB constraint is 171,122,992 for $\Delta = 3$ and $R = 4.0$ and 210,205,334 for $\Delta = 3.5$ and $R = 4.5$, while the original number of arcs $|\mathcal{A}|$ of G is 65,161,516. Regarding the computational time, Algorithm 1 on G has a computational time of 1.35 sec, while Algorithm 1 with LB on G' has 3.55 sec for $\Delta = 3$ and $R = 4.0$ and 4.05 sec for $\Delta = 3.5$ and $R = 4.5$, on an Intel Core i5-10210U CPU 1.60GHz×8.



(a) Sagittal slice (b) Axial slice (c) Coronal slice (d) 3D rendering

(e) Sagittal slice (f) Axial slice (g) Coronal slice (h) 3D rendering

(i) Sagittal slice (j) Axial slice (k) Coronal slice (l) 3D rendering

Figure 3.24: *(a-d) OIFT segmentation result with boundary polarity favoring transitions from bright to dark voxels ($\alpha = 0.75$). (e-h) The OIFT subject to the* LB_{Δ}^{R} *with* $\Delta = 3$ *and* $R = 4.0$. *(i-l) The OIFT subject to the* LB_{Δ}^{R} *with* $\Delta = 3.5$ *and* $R = 4.5$.

Chapter 4

Secondary contributions

The execution times of Algorithm 4 reported in the previous sections consider a priority queue by bucket sorting in the implementation of Algorithm 1. The problem with using bucket sorting is that it limits the arc weights to be small integers. In this chapter, we describe our secondary contributions, involving the Fast-OIFT algorithm, which leads to a more time-efficient implementation of OIFT in practice, by avoiding unnecessary sorting of costs whenever possible and without imposing restrictions on the weights of the arcs.

4.1 Energy-based Max-Min optimizer — Fast-OIFT

In this section, we propose an improved algorithm for the efficient computation (in practice) of the OIFT segmentation based on energy competition of two wavefronts (Algorithm 9). The wavefronts are represented by two separate priority queues Q_1 and Q_0, for object and background respectively, which process the costs of the cutting arcs of candidate segmentations obtained by the region growing from object and background seeds.

More specifically, during Algorithm 9, the region growing from object seeds defines a candidate object $\mathcal{O}_{obj} = L^{-1}(1) \cap \mathcal{P}$, while the growth of background seeds defines an alternative result given by $\mathcal{O}_{bkg} = \mathcal{N} \setminus (L^{-1}(0) \cap \mathcal{P})$, where $\mathcal{P} = \{t \in \mathcal{N} : S(t) = 1\}$ is the set of processed nodes. In Lines 11-12 of Algorithm 9, we have $V(o) = \varepsilon_{\min}(\chi_{\mathcal{O}_{obj}})$ and $V(b) = \varepsilon_{\min}(\chi_{\mathcal{O}_{bkg}})$, where $\chi_{\mathcal{O}} : \mathcal{N} \to \{0, 1\}$ is the characteristic function of object $\mathcal{O} \subset \mathcal{N}$. The idea is to always grow the candidate object with the lowest energy through its output arc with minimum weight, but processing more quickly any subsequent nodes that are reachable through weights smaller than the best energy found so far.

Algorithm 9. – *Segmentation by OIFT via competing wavefronts*

Input: Symmetric edge-weighted image digraph $\langle \mathcal{N}, \mathcal{A}, \omega \rangle$ and non-empty disjoint seed sets $\mathcal{S}_0$ and $\mathcal{S}_1$.
Output: The label map $L : \mathcal{N} \to \{0, 1\}$.
Auxiliary: Priority queues Q_0 and Q_1, stack Q, set $\mathcal{T}$, variables tmp, o, b, e and l, cost functions $V : \mathcal{N} \to [-\infty, \infty]$ and $V' : \mathcal{N} \to [-\infty, \infty]$, and an array of status $S : \mathcal{N} \to \{0, 1\}$, where $S(t) = 1$ for processed nodes and $S(t) = 0$ for unprocessed nodes.

```
1.  For each t ∈ N, do
2.      Set S(t) ← 0.
3.      If t ∈ S_0, then
4.          Set V'(t) ← 0, V(t) ← 0, L(t) ← 0 and insert t in Q_0.
5.      Else If t ∈ S_1, then
6.          Set V'(t) ← 0, V(t) ← 0, L(t) ← 1 and insert t in Q_1.
7.      Else , then
8.          Set V'(t) ← ∞, V(t) ← ∞ and L(t) ← 0.
9.  Set l ← 1.
10. While Q_0 ≠ ∅ and Q_1 ≠ ∅, do
11.     Let b ∈ Q_0 such that V(b) is minimum.
12.     Let o ∈ Q_1 such that V(o) is minimum.
13.     If V(o) < V(b), then
14.         Set e ← V(b), r ← o and remove r from Q_1.
15.     Else If V(o) > V(b), then
16.         Set e ← V(o), r ← b and remove r from Q_0.
17.     Else , then
18.         e ← V(o).
19.         If l = 1, then
20.             Set l ← 0, r ← o and remove r from Q_1.
21.         Else , then
22.             Set l ← 1, r ← b and remove r from Q_0.
23.     Set T ← ∅.
24.     Set S(r) ← 1 and insert r at the top of the stack Q.
25.     While Q ≠ ∅, do
26.         Remove s from the top of the stack Q.
27.         For each ⟨s, t⟩ ∈ A, such that S(t) = 0, do
28.             If L(s) = 1, then tmp ← ω(s, t).
29.             Else If L(s) = 0, then tmp ← ω(t, s).
30.             If tmp < e, then
31.                 If t ∈ Q_0, then remove t from Q_0.
32.                 If t ∈ Q_1, then remove t from Q_1.
33.                 Set S(t) ← 1.
34.                 Set L(t) ← L(s) and insert t in Q.
35.             Else If tmp < V'(t), then
36.                 If V'(t) ≥ V(t), then
37.                     Set L(t) ← L(s) and insert t in T.
38.             Set V'(t) ← tmp.
39.     For each t ∈ T, do
40.         If S(t) = 0, then
41.             Set V(t) ← V'(t).
42.             If L(t) = 1, then
43.                 If t ∈ Q_0, then remove t from Q_0.
```

44. ***If*** $t \notin Q_1$, ***then*** *insert* t *in* Q_1.
45. ***Else*** , ***then***
46. ***If*** $t \in Q_1$, ***then*** *remove* t *from* Q_1.
47. ***If*** $t \notin Q_0$, ***then*** *insert* t *in* Q_0.
48. ***For each*** $r \in Q_1$, ***do***
49. *Set* $S(r) \leftarrow 1$ *and insert* r *in* Q.
50. ***While*** $Q \neq \emptyset$, ***do***
51. *Remove* s *from* Q.
52. ***For each*** $\langle s, t \rangle \in \mathcal{A}$, *such that* $S(t) = 0$, ***do***
53. *Set* $S(t) \leftarrow 1$, $L(t) \leftarrow L(s)$ *and insert* t *in* Q.

The main advantage lies in Lines 30 to 34, where priority queue operations were replaced by simple stack operations, whenever the processed arc weights are worse than the best detected energy so far, meaning that they will not improve the overall energy value, regardless of their order of processing. Otherwise (Lines 35-38), the new costs are registered in an auxiliary map V' (with their respective pixels in $\mathcal{T}$) and the priority queue updates are postponed out of the loop of Lines 25-38, being made only when strictly necessary (Lines 39-47). The reason why we didn't update map V directly is that, since it is associated with priority queues Q_0 and Q_1 in Lines 11-12, updates to this map would result in internal changes in the structure of priority queues, requiring a call to the up-heap operation (also known as bubble-up) in case of binary heaps. Lastly, the codes in Lines 48 to 53 perform faster stack operations for any remaining object pixels.

In our implementation of Algorithm 9, we considered the usage of binary heaps for the priority queues Q_0 and Q_1, in order to avoid restricting the arc weights to be small integers as in bucket sorting. Furthermore, as the queues Q_0 and Q_1 never have repeated elements (i.e., $Q_0 \cap Q_1 = \emptyset$) and their union cannot have more elements than the total number of nodes in the graph (i.e., $Q_0 \cup Q_1 \subseteq \mathcal{N}$), we allocated both queues in a single array in memory, starting each queue at one of the two extremities and growing them in opposite directions.

In order to compare the time efficiency of the different OIFT algorithms, we repeated the quantitative experiments to segment archaeological fragments from Figure 3.19 using images in seven different resolutions. Figure 4.1 shows the execution times of Algorithm 9 ($OIFT_{\text{heap+stack}}$) compared to Algorithm 1 for different data structures of the priority queues (bucket sorting and binary heap for $OIFT_{\text{bucket}}$ and $OIFT_{\text{heap}}$, respectively). In Figures 4.1a-b, we considered $\omega(s,t) = G(s) + G(t)$, where $G(t)$ denotes the magnitude of Sobel gradient. In this case, the gains of Algorithm 9 are more pronounced in relation to Algorithm 1 with priority queue by binary heap. In Algorithm 9 many pixels are processed by depth-first search taking linear time in the size of the graph, rather than log-linear time. This explains why it has a competitive time performance when compared to bucket sorting in Algorithm 1. In fact, it even performed superiorly.

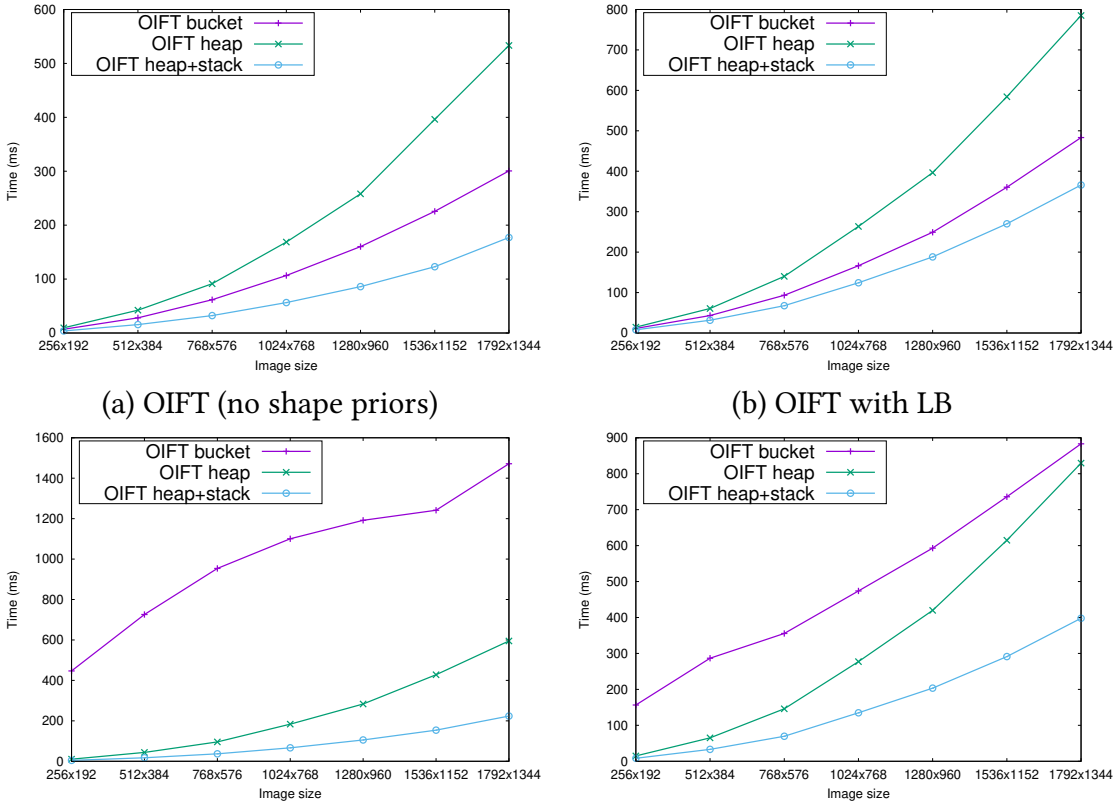
In Figures 4.1c-d, we raised the gradient values to a power of $\beta = 2.5$ to assess the impact of increasing transformations on the runtimes. In this latter case, Algorithm 1 with priority queue by bucket sorting became the slowest. Since the number of buckets is given

by the range of possible finite arc-weight values[1], increasing transformations, like the power value, can lead to a very large memory consumption in the priority queue, which can make Algorithm 1 with bucket sorting unfeasible for larger values of β. Moreover, several empty buckets may have to be traversed when removing a lowest cost pixel from the priority queue on Line 8 of Algorithm 1, making it slower. On the other hand, the other two algorithms ($OIFT_{\text{heap+stack}}$ and $OIFT_{\text{heap}}$) are practically unaffected by these transformations.

In summary, Algorithm 9 generated the fastest results in all cases. It also allows for a better and more balanced treatment of tie zones compared to Algorithm 1 with binary heap, due to the alternation provided in Lines 17-22. Lastly, its running time is unaffected by increasing transformations to arc weights. Hence, Algorithm 9 ($OIFT_{\text{heap+stack}}$) is especially important to avoid restricting the arc weights to be small integer values. Indeed, in recent hybrid methods that explore the joint use of OIFT with other techniques that require the usage of increasing transformations [23], the new algorithm would allow the early application of these weight transformations, simplifying the calculations and avoiding the later increase of computational burden.[2]

[1] Infinite cost values $-\infty$ and ∞ can be assigned to separate specific buckets.

[2] The usage of bucket sorting in OIFT prevents the early application of these weight transformations. So they have to be applied on-the-fly over and over again for each execution of the hybrid pipeline increasing the computational burden.



(a) OIFT (no shape priors)

(b) OIFT with LB

(c) OIFT (no shape priors)

(d) OIFT with LB

Figure 4.1: *The curves of mean execution time of Algorithm 1 for different data structures of the priority queue ($OIFT_{bucket}$ and $OIFT_{heap}$ for bucket sorting and binary heap, respectively) and Algorithm 9 with binary heap operations exchanged for simple stack operations whenever possible ($OIFT_{heap+stack}$). In the first column we have the results by OIFT with no shape priors, while the second column shows us the results by OIFT subject to the LB_{Δ}^{R} with $\Delta = 2$ and $R = 3.5$. In the first row, we used $\omega(s,t) = G(s) + G(t)$ for the arc weights, where $G(t)$ denotes the magnitude of Sobel gradient. In the second row, we repeated the experiments by raising the gradient values to a power of $\beta = 2.5$ (i.e., $\omega(s,t) = G(s)^{\beta} + G(t)^{\beta}$) in order to measure the effects of this increasing transformation on the running times.*

Chapter 5

Conclusions

We have proposed the Local Band (LB) shape constraint, which in its limit case known as Band constraint (i.e., $R \to \infty$) is strongly related to Boundary Band (BB) constraint [5]. We also demonstrated that OIFT lies in the intersection of the Generalized Graph Cut and the General Fuzzy Connectedness frameworks, inheriting their properties. Although the proposed LB constraint can be applied to any method belonging to Generalized Graph Cut framework, in this work it is instantiated for the OIFT method, in order to enable its comparison with BB constraint, which is currently restricted to OIFT. To the best of our knowledge, we are also the first to report OIFT with the Hedgehog shape prior [41, 40].

Our experimental results indicate that LB constraint is less sensitive to the template positioning for high accuracy values compared to BB constraint [5] and Hedgehog shape prior [41, 40], with BB constraint being the most sensitive method. Therefore, in applications where you want to obtain a good segmentation result even in the presence of uncertainties in the template positioning, LB constraint is the recommended method. However, if the objective is to execute the method several times varying the position of the template on the image in order to identify its best positioning, BB becomes a more interesting option, as its sensitivity can help us to isolate the best position. But in this case a similar result could also be obtained by the proposed Band constraint (B) as discussed in Chapter 3.2.1, which can be efficiently implemented by Algorithm 6 from Chapter 3.2.4.

The elimination of redundant arcs for the LB constraint by transitive reduction given by Algorithm 5 from Chapter 3.2.3, combined with the more efficient implementation of OIFT from Chapter 4, which is especially important to avoid restricting the arc weights to small integer values, can allow us to run the method on larger data, such as 3D images, high-resolution photos, among others without a great time overhead.

As future work, we intend to incorporate the LB constraint into the layers of a hierarchical layered digraph [44, 45] to perform multi-object segmentation and to test LB in 3D medical applications for the segmentation of multidimensional magnetic resonance imaging and computed tomography. For instance, in medical images we usually have corresponding mirrored objects, such as organs, due to the symmetry of the human body.

The segmentation of these homologous objects could certainly benefit by incorporating constraints, related to their relative size and similar mirrored shape. How to effectively treat rotation of shape templates to segment objects in arbitrary poses is another challenge. These problems can be mitigated by considering other types of graphs at the superpixel level [92], in order to enable multiple executions at different angles at a low cost. Our future research will follow in these directions.

References

[1] A. V. Aho, M. R. Garey, and J. D. Ullman. “The Transitive Reduction of a Directed Graph”. In: *SIAM Journal on Computing* 1.2 (1972), pp. 131–137. DOI: 10.1137/0201008. URL: https://doi.org/10.1137/0201008 (cit. on pp. 31, 48).

[2] X. Bai and G. Sapiro. “Distance Cut: Interactive Segmentation and Matting of Images and Videos”. In: *Proc. of the IEEE Intl. Conf. on Image Processing*. Vol. 2. 2007, pp. II - 249-II –252 (cit. on p. 1).

[3] Hans H.C. Bejar and Paulo A.V. Miranda. “Oriented Relative Fuzzy Connectedness: Theory, algorithms, and its applications in hybrid image segmentation methods”. In: *EURASIP Journal on Image and Video Processing* 2015.21 (July 2015). DOI: 10.1186/s13640-015-0067-4 (cit. on pp. 2, 4).

[4] Y. Boykov and G. Funka-Lea. “Graph Cuts and Efficient N-D Image Segmentation”. In: *Intl. Jrnl. of Comp. Vision* 70.2 (2006), pp. 109–131. ISSN: 0920-5691 (cit. on pp. 1, 2).

[5] Caio de M. Braz and Paulo A.V. Miranda. “Image segmentation by image foresting transform with geodesic band constraints”. In: *2014 IEEE Int. Conf. Image Process.* IEEE, Oct. 2014, pp. 4333–4337. ISBN: 978-1-4799-5751-4. DOI: 10.1109/ICIP.2014.7025880. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7025880 (cit. on pp. 4, 5, 16, 24, 25, 36, 45, 77).

[6] Caio de Moraes Braz. “Segmentação de imagens pela transformada imagem-floresta com faixa de restrição geodésica”. MA thesis. São Paulo, Brasil: Instituto de Matemática e Estatística, Universidade de São Paulo, 2016. DOI: 10.11606/D.45.2016.tde-01062016-104354 (cit. on pp. 17, 36).

[7] Caio de Moraes Braz, Luiz Felipe D. Santos, and Paulo A. V. Miranda. “Graph-Based Image Segmentation with Shape Priors and Band Constraints”. In: *Discrete Geometry and Mathematical Morphology*. Ed. by Étienne Baudrier et al. Cham: Springer International Publishing, 2022, pp. 287–299. ISBN: 978-3-031-19897-7 (cit. on pp. 5, 6).

[8] T. D. Bui, J. Shin, and T. Moon. *3D densely convolutional networks for volumetric segmentation*. Available: https://arxiv.org/abs/1709.03199. 2017 (cit. on p. 2).

[9] K. S. Camilus and V. K. Govindan. “A Review on Graph Based Segmentation”. In: *Intl. Journal of Image, Graphics and Signal Processing* 4.5 (2012), pp. 1–13. ISSN: 2074-9074 (cit. on p. 1).

[10] D. Caratelli et al. “Fourier-Hankel solution of the Robin problem for the Helmholtz equation in supershaped annular domains”. In: *Boundary Value Problems* 2013.253 (2013). DOI: https://doi.org/10.1186/1687-2770-2013-253 (cit. on p. 40).

[11] T. Chan and Wei Zhu. "Level set based shape prior segmentation". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. 2005, 1164–1170 vol. 2. DOI: 10.1109/CVPR.2005.212 (cit. on p. 3).

[12] S. Chen, K. Ma, and Y. Zheng. *Med3D: Transfer learning for 3D medical image analysis*. Available: https://arxiv.org/abs/1904.00625. 2019 (cit. on p. 2).

[13] K.C. Ciesielski et al. "A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them". In: *Proc. of SPIE on Medical Imaging: Image Processing*. Vol. 8314. 2012. DOI: 10.1117/12.911810 (cit. on pp. 2, 10).

[14] K.C. Ciesielski et al. "Fuzzy Connectedness image segmentation in Graph Cut formulation: A linear-time algorithm and a comparative analysis". In: *Journal of Mathematical Imaging and Vision* 44.3 (Nov. 2012), pp. 375–398 (cit. on pp. 13, 14).

[15] K.C. Ciesielski et al. "Iterative Relative Fuzzy Connectedness for Multiple Objects with Multiple Seeds". In: *Computer Vision and Image Understanding* 107.3 (2007), pp. 160–182. ISSN: 1077-3142 (cit. on p. 1).

[16] Krzysztof Chris Ciesielski, Alexandre Xavier Falcão, and Paulo A. V. Miranda. "Path-Value Functions for Which Dijkstra's Algorithm Returns Optimal Mapping". In: *Journal of Mathematical Imaging and Vision* (Feb. 2018). ISSN: 1573-7683. DOI: 10.1007/s10851-018-0793-1. URL: https://doi.org/10.1007/s10851-018-0793-1 (cit. on pp. 1, 9, 21, 27).

[17] Krzysztof Chris Ciesielski, Gabor T. Herman, and T. Yung Kong. "General Theory of Fuzzy Connectedness Segmentations". In: *Journal of Mathematical Imaging and Vision* 55.3 (July 2016), pp. 304–342. ISSN: 1573-7683. DOI: 10.1007/s10851-015-0623-7 (cit. on pp. 5, 21, 22).

[18] Krzysztof Chris Ciesielski et al. "Efficient algorithm for finding the exact minimum barrier distance". In: *Computer Vision and Image Understanding* 123 (2014), pp. 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 (cit. on p. 1).

[19] J. Cousty et al. "Watershed Cuts: Thinnings, Shortest Path Forests, and Topological Watersheds". In: *Trans. on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 925–939. ISSN: 0162-8828. DOI: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.71 (cit. on p. 1).

[20] Daniel Cremers, Timo Kohlberger, and Christoph Schnörr. "Nonlinear Shape Statistics in Mumford-Shah Based Segmentation". In: *Proceedings of the 7th European Conference on Computer Vision-Part II*. ECCV '02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 93–108. ISBN: 3540437444 (cit. on p. 3).

[21] Daniel Cremers, Stanley J. Osher, and Stefano Soatto. "Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation". In: *Int. J. Comput. Vision* 69.3 (Sept. 2006), pp. 335–351. ISSN: 0920-5691. DOI: 10.1007/s11263-006-7533-5. URL: https://doi.org/10.1007/s11263-006-7533-5 (cit. on p. 3).

[22] P. Das and O. Veksler. "Semiautomatic Segmentation with Compact Shapre Prior". In: *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. 2006, pp. 28–28. DOI: 10.1109/CRV.2006.63 (cit. on p. 3).

[23] Caio L. Demario and Paulo A. V. Miranda. "Relaxed Oriented Image Foresting Transform for Seeded Image Segmentation". In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 1520–1524. DOI: 10.1109/ICIP.2019.8803080 (cit. on p. 74).

[24] E.W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1 (1959), pp. 269–271 (cit. on p. 9).

[25] David H. Douglas and Thomas K. Peucker. "ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE". In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (1973), pp. 112–122 (cit. on p. 3).

[26] A.X. Falcão, J. Stolfi, and R.A. Lotufo. "The Image Foresting Transform: Theory, Algorithms, and Applications". In: *IEEE TPAMI* 26.1 (2004), pp. 19–29 (cit. on pp. 1, 9, 60).

[27] A.X. Falcão et al. "User-steered Image Segmentation Paradigms: Live-wire and Live-lane". In: *Graph. Models and Image Proc.* 60 (1998), pp. 233–260 (cit. on p. 1).

[28] Clement Farabet et al. "Learning Hierarchical Features for Scene Labeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1915–1929. DOI: 10.1109/TPAMI.2012.231 (cit. on p. 2).

[29] Daniel Freedman and Tao Zhang. "Interactive graph cut based segmentation with shape priors". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE. 2005, pp. 755–762 (cit. on pp. 3, 4, 39, 40, 42, 43).

[30] K. A. Ganser et al. "A deformable digital brain atlas system according to Talairach and Tournoux". In: *Medical Image Analysis* 8.1 (Mar. 2004), pp. 3–22 (cit. on p. 3).

[31] S. Garrido-Jurado et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0031320314000235 (cit. on p. 3).

[32] J. Gielis. *The Geometrical Beauty of Plants.* Atlantis Press: Paris, France, 2017. ISBN: 978-94-6239-150-5. DOI: 10.2991/978-94-6239-151-2 (cit. on pp. 39–41).

[33] Johan Gielis. "A generic geometric transformation that unifies a wide range of natural and abstract shapes". In: *American Journal of Botany* 90.3 (2003), pp. 333–338. DOI: https://doi.org/10.3732/ajb.90.3.333. eprint: https://bsapubs.onlinelibrary.wiley.com/doi/pdf/10.3732/ajb.90.3.333. URL: https://bsapubs.onlinelibrary.wiley.com/doi/abs/10.3732/ajb.90.3.333 (cit. on pp. 39–41).

[34] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN: 013168728X (cit. on p. 4).

[35] L. Grady. "Random Walks for Image Segmentation". In: *IEEE Trans. Pattern Anaysis and Machine Intelligence* 28.11 (2006), pp. 1768–1783. ISSN: 0162-8828. DOI: http://dx.doi.org/10.1109/TPAMI.2006.233 (cit. on p. 1).

[36] N. T. Gridgeman. "Lamé Ovals". In: *The Mathematical Gazette* 54.387 (1970), pp. 31–37. DOI: 10.2307/3613154 (cit. on p. 39).

[37] V. Gulshan et al. "Geodesic Star Convexity for Interactive Image Segmentation". In: *Proc. of Computer Vision and Pattern Recognition.* San Francisco, CA, 2010, pp. 3129–3136 (cit. on pp. 5, 15, 45).

[38] Lin Guo et al. “Morphological dissection and cellular and transcriptome characterizations of bamboo pith cavity formation reveal a pivotal role of genes related to programmed cell death”. In: *Plant Biotechnology Journal* 17.5 (2019), pp. 982–997. DOI: https://doi.org/10.1111/pbi.13033. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/pbi.13033. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/pbi.13033 (cit. on p. 39).

[39] K. He et al. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 2).

[40] H. Isack et al. “Hedgehog Shape Priors for Multi-Object Segmentation”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 2434–2442 (cit. on pp. 3–5, 17, 45, 77).

[41] Hossam N. Isack, Yuri Boykov, and Olga Veksler. “A-expansion for multiple "hedgehog" shapes”. In: *CoRR* abs/1602.01006 (2016). arXiv: 1602.01006. URL: http://arxiv.org/abs/1602.01006 (cit. on pp. 3–5, 17, 45, 77).

[42] S. Kobashi and J.K. Udupa. “Fuzzy object model based fuzzy connectedness image segmentation of newborn brain MR images”. In: *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. Oct. 2012, pp. 1422–1427. DOI: 10.1109/ICSMC.2012.6377934 (cit. on p. 39).

[43] Syoji Kobashi et al. “Neonatal brain segmentation using 4-D fuzzy object model”. In: *Informatics, Electronics Vision (ICIEV), 2014 International Conference on*. May 2014, pp. 1–7. DOI: 10.1109/ICIEV.2014.6850710 (cit. on p. 39).

[44] L. M. C. Leon and P. A. V. D. Miranda. “Multi-Object Segmentation by Hierarchical Layered Oriented Image Foresting Transform”. In: *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. Oct. 2017, pp. 79–86. DOI: 10.1109/SIBGRAPI.2017.17 (cit. on pp. 4, 9, 13, 77).

[45] Leissi M.C. Leon, Krzysztof C. Ciesielski, and Paulo A.V. Miranda. “Efficient Hierarchical Multi-Object Segmentation in Layered Graphs”. In: *Mathematical Morphology - Theory and Applications* 5.1 (2021), pp. 21–42. DOI: doi:10.1515/mathm-2020-0108. URL: https://doi.org/10.1515/mathm-2020-0108 (cit. on pp. 4, 5, 77).

[46] M.E. Leventon, W.E.L. Grimson, and O. Faugeras. “Statistical shape influence in geodesic active contours”. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2000, pp. 316–323. DOI: 10.1109/CVPR.2000.855835 (cit. on p. 3).

[47] O. Lézoray and L. Grady. *Image Processing and Analysis with Graphs: Theory and Practice*. California, USA: CRC Press, 2012. ISBN: 1439855072 (cit. on p. 5).

[48] Guodong Li et al. “Automatic liver segmentation based on shape constraints and deformable graph cut in CT images”. In: *IEEE Transactions on Image Processing* 24.12 (2015), pp. 5315–5329 (cit. on p. 4).

[49] Wenqi Li et al. “On the Compactness, Efficiency, and Representation of 3D Convolutional Networks: Brain Parcellation as a Pretext Task”. In: *Information Processing in Medical Imaging*. Ed. by Marc Niethammer et al. Cham: Springer International Publishing, 2017, pp. 348–360. ISBN: 978-3-319-59050-9 (cit. on p. 2).

[50] Xiaoqiang Li, Jingsong Chen, and Huafu Fan. "Interactive Image Segmentation Based on Grow Cut of Two Scale Graphs". In: *Advances on Digital Television and Wireless Multimedia Communications*. Ed. by Wenjun Zhang et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 90–95. ISBN: 978-3-642-34595-1 (cit. on p. 1).

[51] Shuyan Lin et al. "A geometrical model for testing bilateral symmetry of bamboo leaf with a simplified Gielis equation". In: *Ecology and Evolution* 6.19 (2016), pp. 6798–6806. DOI: https://doi.org/10.1002/ece3.2407. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ece3.2407. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.2407 (cit. on p. 41).

[52] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298965 (cit. on p. 2).

[53] A. Madabhushi and J.K. Udupa. "Interplay Between Intensity Standardization and Inhomogeneity Correction in MR Image Processing". In: *IEEE Transactions on Medical Imaging* 24.5 (2005), pp. 561–576 (cit. on p. 1).

[54] James Malcolm, Yogesh Rathi, and Allen Tannenbaum. "Graph Cut Segmentation with Nonlinear Shape Priors". In: *2007 IEEE International Conference on Image Processing*. Vol. 4. 2007, pp. IV - 365-IV –368. DOI: 10.1109/ICIP.2007.4380030 (cit. on p. 3).

[55] K.-K. Maninis et al. "Deep extreme cut: From extreme points to object segmentation". In: *In IEEE Conf. on Computer Vision and Pattern Recognition*. 2018, pp. 616–625 (cit. on p. 2).

[56] L. A. C. Mansilla, P. A. V. Miranda, and F. A. M. Cappabianco. "Oriented image foresting transform segmentation with connectivity constraints". In: *2016 IEEE International Conference on Image Processing (ICIP)*. Sept. 2016, pp. 2554–2558 (cit. on p. 4).

[57] L.A.C. Mansilla and P.A.V. Miranda. "Image Segmentation by Oriented Image Foresting Transform with Geodesic Star Convexity". In: *15th International Conference on Computer Analysis of Images and Patterns (CAIP)*. Vol. 8047. York, UK, Aug. 2013, pp. 572–579 (cit. on pp. 4, 27, 60).

[58] L.A.C. Mansilla and P.A.V. Miranda. "Image Segmentation by Oriented Image Foresting Transform: Handling Ties and Colored Images". In: *18th Intl. Conf. on Digital Signal Processing*. Greece, July 2013, pp. 1–6 (cit. on pp. 4, 10, 21).

[59] Lucy AC Mansilla and Paulo AV Miranda. "Oriented image foresting transform segmentation: Connectivity constraints with adjustable width". In: *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*. IEEE. 2016, pp. 289–296 (cit. on p. 4).

[60] Monica M. S. Matsumoto and Jayaram K. Udupa. "Optimal hierarchies for fuzzy object models". In: *In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*. Vol. 8671. Orlando, Florida, USA, 2013 (cit. on p. 39).

[61] Monica M. S. Matsumoto et al. "Automatic localization of IASLC-defined mediastinal lymph node stations on CT images using fuzzy models". In: *In Proceedings of SPIE on Medical Imaging: Computer-Aided Diagnosis*. Vol. 9035. San Diego, California, USA, 2014 (cit. on p. 39).

[62] M. Matsuura. "Gielis' superformula and regular polygons". In: *Journal of Geometry* 106 (2015), pp. 383–403. DOI: https://doi.org/10.1007/s00022-015-0269-z (cit. on p. 40).

[63] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79 (cit. on p. 2).

[64] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. "Cloud Bank: A Multiple Clouds Model and its use in MR Brain Image Segmentation". In: *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*. Boston, MA, 2009, pp. 506–509 (cit. on p. 39).

[65] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. "*CLOUDS*: A Model for Synergistic Image Segmentation". In: *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*. Paris, France, May 2008, pp. 209–212 (cit. on p. 39).

[66] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. "Synergistic Arc-Weight Estimation for Interactive Image Segmentation using Graphs". In: *Computer Vision and Image Understanding* 114.1 (Jan. 2010), pp. 85–99. ISSN: 1077-3142 (cit. on p. 2).

[67] P.A.V. Miranda and L.A.C. Mansilla. "Oriented Image Foresting Transform Segmentation by Seed Competition". In: *IEEE Transactions on Image Processing* 23.1 (Jan. 2014), pp. 389–398 (cit. on pp. 2, 9, 10).

[68] C. de Moraes Braz et al. "Optimum Cuts in Graphs by General Fuzzy Connectedness with Local Band Constraints". In: *Journal of Mathematical Imaging and Vision* 62 (June 2020), pp. 659–672. DOI: 10.1007/s10851-020-00953-w (cit. on pp. 6, 32).

[69] Caio de Moraes Braz et al. "Graph-Based Segmentation with Local Band Constraints". In: *Discrete Geometry for Computer Imagery*. Ed. by Michel Couprie et al. Cham: Springer International Publishing, 2019, pp. 155–166. ISBN: 978-3-030-14085-4 (cit. on pp. 5, 6).

[70] S.D. Olabarriaga and A.W.M. Smeulders. "Interaction in the Segmentation of Medical Images: A Survey". In: *Medical Image Analysis* 5.2 (June 2001), pp. 127–142 (cit. on p. 1).

[71] Débora E. C. Oliveira, Caio L. Demario, and Paulo A. V. Miranda. "Image Segmentation by Relaxed Deep Extreme Cut with Connected Extreme Points". In: *Discrete Geometry and Mathematical Morphology*. Ed. by Joakim Lindblad, Filip Malmberg, and Nataša Sladoje. Cham: Springer International Publishing, 2021, pp. 441–453. ISBN: 978-3-030-76657-3 (cit. on p. 2).

[72] M. Ortega et al. "Deformable brain atlas: Validation of the location of subthalamic nucleus using T1-weighted MR images of patients operated on for Parkinson's". In: *Medical Image Analysis* 32.5 (2008), pp. 367–378 (cit. on p. 3).

[73] Leticia Rittner, Jayaram K. Udupa, and Drew A. Torigian. "Multiple fuzzy object modeling improves sensitivity in automatic anatomy recognition". In: *In Proceedings of SPIE on Medical Imaging: Image Processing*. Vol. 9034. San Diego, California, USA, 2014 (cit. on p. 39).

[74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4 (cit. on p. 2).

[75] Mikael Rousson and Nikos Paragios. "Shape Priors for Level Set Representations". In: *Computer Vision — ECCV 2002*. Ed. by Anders Heyden et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 78–92. ISBN: 978-3-540-47967-3 (cit. on p. 3).

[76] Robert Sedgewick and Kevin Daniel Wayne. *Algorithms (4th ed.)* Addison-Wesley Professional, 2011, pp. 661–665. ISBN: 9780321573513 (cit. on p. 31).

[77] J.A. Sethian. "A fast marching level set method for monotonically advancing fronts". In: *Proceedings of the National Academy of Sciences of the USA* 93.4 (1996), pp. 1591–5 (cit. on p. 25).

[78] Dinggang Shen and C. Davatzikos. "An adaptive-focus deformable model using statistical and geometric information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 906–913. DOI: 10.1109/34.868689 (cit. on p. 3).

[79] Pei-Jian Shi et al. "Capturing spiral radial growth of conifers using the superellipse to model tree-ring geometric shape". In: *Frontiers in Plant Science* 6 (2015). ISSN: 1664-462X. DOI: 10.3389/fpls.2015.00856. URL: https://www.frontiersin.org/article/10.3389/fpls.2015.00856 (cit. on p. 39).

[80] Pei-Jian Shi et al. "Comparison of dwarf bamboos (Indocalamus sp.) leaf parameters to determine relationship between spatial density of plants and total leaf area per plant". In: *Ecology and Evolution* 5.20 (2015), pp. 4578–4589. DOI: https://doi.org/10.1002/ece3.1728. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ece3.1728. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.1728 (cit. on p. 41).

[81] Peijian Shi, David A. Ratkowsky, and Johan Gielis. "The Generalized Gielis Geometric Equation and Its Application". In: *Symmetry* 12.4 (2020). ISSN: 2073-8994. DOI: 10.3390/sym12040645. URL: https://www.mdpi.com/2073-8994/12/4/645 (cit. on pp. 5, 39, 41).

[82] Peijian Shi et al. "Proportional Relationship between Leaf Area and the Product of Leaf Length and Width of Four Types of Special Leaf Shapes". In: *Forests* 10.2 (2019). ISSN: 1999-4907. DOI: 10.3390/f10020178. URL: https://www.mdpi.com/1999-4907/10/2/178 (cit. on p. 41).

[83] Dennis J. Silva et al. "Efficient incremental computation of attributes based on locally countable patterns in component trees". In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3738–3742. DOI: 10.1109/ICIP.2016.7533058 (cit. on p. 3).

[84] G. Slabaugh and G. Unal. "Graph cuts segmentation using an elliptical shape prior". In: *IEEE International Conference on Image Processing 2005*. Vol. 2. 2005, pp. II–1222. DOI: 10.1109/ICIP.2005.1530282 (cit. on p. 3).

[85] Italos Estilon de Souza, Barbara C. Benato, and Alexandre Xavier Falcão. "Feature Learning from Image Markers for Object Delineation". In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2020, pp. 116–123. DOI: 10.1109/SIBGRAPI51738.2020.00024 (cit. on p. 2).

[86] F. Tian et al. “Comparison of seed morphology of two ginkgo cultivars”. In: *Journal of Forestry Research* 31 (2020), pp. 751–758. DOI: https://doi.org/10.1007/s11676-018-0770-y (cit. on p. 41).

[87] A. Tsai et al. “A shape-based approach to the segmentation of medical imagery using level sets”. In: *IEEE Transactions on Medical Imaging* 22.2 (2003), pp. 137–154. DOI: 10.1109/TMI.2002.808355 (cit. on p. 3).

[88] J.K. Udupa et al. “Automatic Anatomy Recognition via Fuzzy Object Models”. In: *In Proceedings of SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*. Vol. 8316. San Diego, California, USA, 2012 (cit. on p. 39).

[89] J.K. Udupa et al. “Fuzzy Object Modeling”. In: *In Proceedings of SPIE on Medical Imaging: Visualization, Image-Guided Procedures, and Modeling*. Vol. 7964. doi: 10.1117/12.878273. 2011 (cit. on p. 39).

[90] Jayaram K. Udupa et al. “Body-wide hierarchical fuzzy modeling, recognition, and delineation of anatomy in medical images”. In: *Medical Image Analysis* 18.5 (2014), pp. 752–771. ISSN: 1361-8415. DOI: http://dx.doi.org/10.1016/j.media.2014.04.003 (cit. on p. 39).

[91] Jayaram K. Udupa et al. *Fuzzy model-based body-wide anatomy recognition in medical images*. 2013. DOI: 10.1117/12.2007983. URL: http://dx.doi.org/10.1117/12.2007983 (cit. on p. 39).

[92] John E. Vargas-Muñoz et al. “An Iterative Spanning Forest Framework for Superpixel Segmentation”. In: *IEEE Transactions on Image Processing* 28.7 (2019), pp. 3477–3489. DOI: 10.1109/TIP.2019.2897941 (cit. on pp. 2, 78).

[93] O. Veksler. “Star shape prior for graph-cut image segmentation”. In: *European Conference on Computer Vision (ECCV)*. Vol. 5304. 2008, pp. 454–467 (cit. on p. 15).

[94] S. Vicente, V. Kolmogorov, and C. Rother. “Graph cut based image segmentation with connectivity priors”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. June 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587440 (cit. on p. 4).

[95] N. Vu and B. S. Manjunath. “Shape prior segmentation of multiple objects with graph cuts”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. June 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587450 (cit. on pp. 3, 43).

[96] Guotai Wang et al. “Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning”. In: *IEEE Transactions on Medical Imaging* 37.7 (2018), pp. 1562–1573. DOI: 10.1109/TMI.2018.2791721 (cit. on p. 2).

[97] Qiang Wei et al. “Exploring key cellular processes and candidate genes regulating the primary thickening growth of Moso underground shoots”. In: *New Phytologist* 214.1 (2017), pp. 81–96. DOI: https://doi.org/10.1111/nph.14284. eprint: https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/nph.14284. URL: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/nph.14284 (cit. on p. 39).

[98] S. Wolf et al. “Learned Watershed: End-to-End Learning of Seeded Segmentation”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 2030–2038. DOI: 10.1109/ICCV.2017.222 (cit. on p. 2).

[99] Steffen Wolf et al. *The Mutex Watershed and its Objective: Efficient, Parameter-Free Image Partitioning*. Available: https://arxiv.org/abs/1904.12654. 2019 (cit. on p. 2).

[100] Y. Xu, T. Géraud, and L. Najman. “Context-based energy estimator: Application to object segmentation on the tree of shapes”. In: *2012 19th IEEE International Conference on Image Processing*. Sept. 2012, pp. 1577–1580 (cit. on p. 25).

[101] Tao Zhang and Freedman. “Tracking objects using density matching and shape priors”. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 1056–1062 vol.2. DOI: 10.1109/ICCV.2003.1238466 (cit. on p. 39).