
Segmentation and Tracking in High-Throughput Bioimaging

Ricard Delgado Gonzalo

Thèse N° 5657 (Mars 2013)

*Thèse présentée à la faculté des sciences et techniques de l'ingénieur
pour l'obtention du grade de docteur ès sciences
et acceptée sur proposition du jury*

Prof. Pierre Vandergheynst, *président*
Prof. Michael Unser, *directeur de thèse*
Dr. Jean-Christophe Olivo-Marin, *rapporteur*
Prof. Mathews Jacob, *rapporteur*
Prof. Dimitri Van De Ville, *rapporteur*

École polytechnique fédérale de Lausanne—2013

Cover design by Annette Unser
Printing and binding by Repro-EPFL
Typeset with L^AT_EX
Copyright © 2013 by Ricard Delgado Gonzalo
Available at <http://bigwww.epfl.ch/>

Don't Forget To Be Awesome

HANK AND JOHN GREEN

To my parents and brother.

Abstract

This thesis addresses the problem of detecting and segmenting biological objects and tracking them over time. This is far from trivial due to the fact that, in biology, the objects of interest are usually indistinguishable from each other and can appear tightly packed and in various configurations. Since we focus on objects that have a constrained shape and move according to specific patterns, it seems natural to approach the detection, segmentation, and tracking problems with model-based techniques.

We present a class of parametric active contours that use a novel kind of B-splines as basis functions. We prove analytically that our new bases have the shortest-possible support, subject to some design constraints. While the resulting active contours are versatile and able to closely approximate any closed curve in the plane, their most important feature is the fact that they admit ellipses within their span. Thus, they are able to represent exact circular and elliptical shapes and are particularly appropriate to delineate cross sections of cylindrical-like conduits and to outline blob-like objects. Then, we extend our model to a fully parametric 3D design. The resulting active surface can approximate smooth blob-like objects with good accuracy and can perfectly reproduce spheres and ellipsoids of any position and orientation.

Finally, we make use of our active contours to segment and track mitotic cells in large-scale time-lapse images. Due to their optimally short support, our active contours are computationally efficient. Moreover, we designed a highly parallelizable image analysis toolkit to further increase the throughput rate.

Keywords: Time-lapse, high-throughput microscopy, exponential B-spline, interpolation, parameterization, multiresolution, Fourier descriptor, segmentation, active contour, active surface, shape prior, ellipse, tracking, mitosis, crowd, particle filter, ImageJ.

Résumé

Cette thèse aborde le problème de la détection et segmentation d'objets biologiques, ainsi que de leur poursuite temporelle. Ce problème est rendu difficile par le fait que, en biologie, les objets d'intérêt ne sont généralement pas différenciables les uns des autres, peuvent s'agglutiner et apparaître dans plusieurs configurations. Nous nous concentrons sur des objets qui ont une forme préétablie et se déplacent selon des trajectoires prévisibles ; dès lors, il semble naturel que détection, segmentation et poursuite soient traitées selon des techniques fondées sur des modèles.

Nous présentons une classe de contours actifs paramétriques qui utilisent un nouveau type de B-splines comme fonctions de base. Nous démontrons analytiquement que nos nouvelles bases ont un support minimal tout en satisfaisant certaines contraintes de conception. Alors que les contours actifs qui en résultent sont polyvalents et capables d'approcher toute courbe fermée du plan, leur caractéristique majeure est le fait que le sous-espace qu'ils engendrent contient les ellipses. Ainsi, ils sont en mesure d'offrir une représentation exacte de toute forme circulaire ou elliptique, et sont donc particulièrement propices pour délimiter les sections de tubes cylindriques et celles d'objets globuleux. Ensuite, nous étendons notre modèle à une conception paramétrique 3D. La surface active qui en résulte peut approcher avec bonne précision les objets lisses et globuleux. Elle peut reproduire de façon exacte les sphères et ellipsoïdes de n'importe quelle position et orientation.

Enfin, nous tirons parti de nos contours actifs pour segmenter et poursuivre des cellules mitotiques dans une volumineuse séquence d'images. En raison de leur support minimal, nos contours actifs sont efficaces d'un point de vue calculatoire. En outre, nous avons conçu un outil d'analyse d'images qui se prête bien à la parallélisation, dans le but d'augmenter encore le débit de calcul.

Mots-clé : Vidéo-microscopie, B-splines exponentielles, interpolation, multirésolution, paramétrisation, descripteurs de Fourier, segmentation, contours actifs, surfaces actives, ellipses, contraintes de forme, suivi de cellules, mitose, cultures cellulaires, filtres à particules, ImageJ.

Acknowledgements

This thesis would not have materialized without the help and cooperation of many people. I take this opportunity to express my gratitude to all of them. First and foremost, I thank my advisor Prof. Michael Unser. Terribly lost in details to start with, it was his patient guidance that eventually helped me strike the right chord. He has always encouraged fresh and original ideas, and helped refine them at the same time. I particularly value the advise and training I received from him concerning the exposition of scientific ideas. Michael has also been very kind and considerate.

I express my sincere thanks to the president of the thesis jury, Prof. Pierre Vandergheynst, and the official referees, Prof. Mathews Jacob, Prof. Dimitri Van De Ville, and Dr. Jean-Christophe Olivo-Marin, for reviewing the thesis.

I take this opportunity to thank the fellow present and past lab members of the Biomedical Imaging Group (BIG), Dr. Jean-Charles Baritoux, Prof. İlker Bayram, Ayush Bhandari, Emrah Bostan, Aurélien Bourquard, Julien Fageot, Dr. Matthieu Guerquin-Kern, Ulugbek Kamilov, Dr. Djano Kandaswamy, Dr. Ildar Khalidov, Dr. Hagai Kirshner, Dr. Stamatis Lefkimmatis, Dr. Florian Luisier, Ramtin Madani, Dr. Simona Maggio, Junhong Min, Masih Nilchian, Dr. Chiara Olivieri, Pedram Pad, Zsuzsanna Püspöki, Dr. Sathish Ramani, Dr. Daniel Sage, Prof. Chandra Sekhar Seelamantula, Dr. Pouya Tafti, Raquel Terrés Cristofani, Dr. Philippe Thévenaz, Prof. Dimitri Van De Ville Dr. Cédric Vonesch, and Dr. John Paul Ward. And more particularly, I thank my present and past office mates Dr. François Aguet, Dr. Arash Amini, Dr. Kunal N. Chaudhury, Dr. Nicolas Chenouard, and Virginie Uhlmann. I really appreciate the efforts of Virginie to increase the geekiness of the laboratory. I am also grateful to Manuelle Mary for helping me out with various administrative matters inside and outside EPFL.

I must say it was a real pleasure interacting with the members of the BIG group. A special thanks to Chandra and Philippe for introducing me to the world of snakes, Pouya Tafti for the stimulating discussions on myriads of topics, and François, Philippe and Daniel for sharing their knowledge on various technical matters related to ImageJ. I particularly thank Nicolas for helping me with my

ACKNOWLEDGEMENTS

first baby steps into Icy.

I thank my collaborators, Prof. Sebastian Maerkl, Dr. Nicolas Denervaud, and Jean-Bernard Nobs from the DynamiX project for their help and cooperation understanding the processes laying behind the natural images I helped to analyze. Moreover, I also want to thank my collaborator from the WingX project Thomas Schaffter for showing me the good practices of collaborative software development.

I also take this opportunity to express my deep gratitude to all the member of the BIG Snake Project: Cédric, Emrah, Masih, Nicolas, Ramtin, Ulugbek and Virginie. Together we made our snakes reach a totally new level of awesomeness.

I thank all my friends from Spain, giving special attention to the ones that have been giving me their support and friendship since childhood: Alberto Ramírez, Dídac Farré, Xavier Garriga, and Gerard Salom. I do not want to forget all my friends from my universities (FME, ETSETB, and IIT). Without them, my geekiness would not be so strong. I am also very gratefully to Raquel for her tremendous support and affection. I am certain that this thesis would not have been possible without her patience and support.

Finally, I wish to thank my family members, especially my parents and my brother. I dedicate this thesis to them.

Table of Contents

Abstract	v
Résumé	vii
Acknowledgements	ix
Table of Contents	xi
1 Introduction	1
1.1 The Quest for Quantitative Microscopy	1
1.2 Image Analysis in Bioimaging	2
1.3 The DynamiX Project	2
1.4 Developement of Principalized Image Analysis Tools for Systems Biology	3
1.5 Overview of the Contributions	4
1.6 Organization of this Thesis	5
2 State-of-the-art in Image Analysis	7
2.1 A Survey on Segmentation	7
2.1.1 Snakes, a Perfect Fit for Image Segmentation	8
2.2 Image Analysis Software	14
2.2.1 Software Design	14
2.2.2 Open Image Analysis Platforms	15
3 Spline Bases for Representation of Curves	19
3.1 Parametric Curves	20
3.1.1 Generic Curves	20
3.1.2 Closed Curves	21
3.1.3 Desirable Properties of Bases in the Periodic Settings	22
3.1.4 Approximation and Reproduction Properties in Periodic Settings	23
3.2 Reproduction of Exponential Polynomials	24

3.2.1	Preliminary Definitions	24
3.2.2	Reproduction Conditions	24
3.2.3	Exponential B-Splines	26
3.2.4	Distributional Decomposition	27
3.2.5	Minimal-Support Generating Functions	31
3.2.6	Interpolator	33
3.3	Multiresolution and Subdivision	33
3.3.1	Classical Multiresolution of Exponential B-Splines	33
3.3.2	Subdivision Scheme	34
3.3.3	Multiresolution-Reproduction Capabilities	35
3.4	Applications	36
3.4.1	Reproduction of Ellipses	36
3.4.2	Reproduction of Higher-Order Harmonics	44
3.5	Conclusions	49
4	2D Spline Snakes	51
4.1	Parametric Snakes	52
4.1.1	Parametric Representation of Closed Curves	52
4.1.2	Desirable Properties for the Basis Functions	53
4.2	Reproduction of Ellipses	55
4.2.1	Minimum-Support Ellipse-Reproducing Basis	56
4.2.2	Approximation Properties	56
4.2.3	Approximation Order	57
4.2.4	Best Constant and Ellipse Fitting	58
4.2.5	Expansion of Sinusoids	59
4.3	Energies and Implementation	61
4.3.1	Snake Energy	61
4.3.2	Fast Energy Computation	64
4.3.3	Sampling	65
4.3.4	Optimization	65
4.4	Experiments and Simulations	66
4.4.1	Approximation of Sinusoids	67
4.4.2	Accuracy and Robustness to Noise	67
4.4.3	Medical Data	72
4.4.4	Real Data	77
4.5	Conclusions	78
	Appendices	80
4.A	Implementation Details	80
4.A.1	Image Energy	80
4.A.2	Partial Derivatives of the Image Energy	82
5	Extension to 3D Spline Snakes	85
5.1	Spline Surfaces	86

5.1.1	Parametric Representation of Surfaces	86
5.1.2	Desirable Properties of Basis Functions	87
5.1.3	3D Snake Model	90
5.2	Energies and Implementation	98
5.2.1	Snake Energy	98
5.2.2	Fast Energy Computation	102
5.2.3	Sampling	102
5.2.4	Optimization	103
5.2.5	Self-Intersection Detection	103
5.2.6	User Interaction	104
5.2.7	Initialization	104
5.3	Experiments and Simulations	105
5.3.1	Twisting the Snake	105
5.3.2	Robustness to Noise	107
5.3.3	Segmentation of Overlapping Objects	108
5.3.4	Approximation of Shapes	110
5.3.5	Segmentation of 3D Confocal Microscopic Images	112
5.4	Conclusions	115
	Appendices	116
5.A	Implementation Details	116
5.A.1	Image Energy	116
6	Snake-Based Algorithm for Tracking Mitotic Cells	119
6.1	Single Mitotic Cell Tracker	120
6.1.1	Particle-Filter Framework	120
6.1.2	Variational Importance Sampling	123
6.1.3	Application to Time-Lapse Microscopy	124
6.2	High-Throughput Multi-Target Tracker	128
6.2.1	Notation	128
6.2.2	Probabilistic Graph Formulation	129
6.2.3	Efficient Graph-Based Algorithm	130
6.2.4	Motion Model	132
6.2.5	Application to Time-Lapse Microscopy	133
6.3	Conclusions	135
7	Conclusion	137
7.1	Technical Contributions	137
7.2	Contributions to Research Projects in Life Sciences	139
7.2.1	The DynamiX Project	139
7.2.2	The WingX Project	140
7.2.3	Drosophila Fly Locomotion Study	141
7.2.4	Estimation of Local Aortic Elastic Properties with MRI	142
7.2.5	Assessment of Chromosomal Size Variation in CHO Cells	143

TABLE OF CONTENTS

7.3 Outlook for Future Research	144
Bibliography	147
Curriculum Vitæ	163

Chapter 1

Introduction

1.1 The Quest for Quantitative Microscopy

By their nature, biological systems are dynamic. In recent years, there has been an increasing interest in getting a proper understanding of the underlying cellular and molecular processes [1, 2]. One of the major challenges of current biomedical research is to characterize not just the spatial organization of these complex systems, but their spatio-temporal relationships as well [3].

Thanks to substantial improvements in optics [4, 5], imaging sensors [6, 7, 8], and florescence labeling methods [9, 10, 11], microscopy has matured to the point that it enables sensitive time-lapse imaging of cells *in vivo* and even of single molecules [12, 13]. Microscopy was initially a qualitative technique, but the transition to computerized microscopy enables one to extract meaningful quantitative data from images [14, 15]. Making microscopy more quantitative will bring important scientific benefits in the form of new applications and improved performance and reproducibility.

A direct consequence of the advances in high-throughput microscopy is that the size and complexity of image data are increasing. Datasets generated in time-lapse experiments commonly consist of hundreds to thousands of images, each containing hundreds to thousands of objects to be analyzed [16, 17]. Such huge amounts of data cannot be analyzed by visual inspection or manual processing within any reasonable amount of time. Automated methods are therefore necessary, not only to handle the growing rate at which images are acquired, but also to provide a level of sensitivity and objectivity that human observers cannot match.

1.2 Image Analysis in Bioimaging

The aim of image analysis in bioimaging is to use cutting-edge techniques from the fields of Image Processing and Computer Vision to achieve insights into biological problems through analysis of large-scale image datasets [18].

The domain of action of the tools provided by these fields is very large. It actually begins during the image acquisition process itself. All imaging modalities introduce a certain degree of distortion in the captured images, which are already intrinsically noisy. These deformations can range from simple smoothing with a point spread function (PSF), to optical aberrations, or non-linearities in the acquisition process [19].

The quality of an acquired image from an optical imaging system can be limited by factors such as imperfections or misalignment in the lenses. However, there is a fundamental maximum to the magnification of any optical system which is due to diffraction [19]. Shannon's sampling theory from Signal Processing provides the conditions for setting the optimal resolution of the camera during the acquisition process to match the magnification of the optical device [20]. As for the next step, a large variety of image restoration algorithms exist to facilitate the extraction of the information of interest. Among them, the most used ones are deconvolution [21] and denoising [22] algorithms. Then, the last major image analysis challenge is to reliably segment thousands of individual biological objects and to track them over time. This is far from trivial due to the dependence on the imaging modality and the fact that the cells can be tightly packed in the growth chamber and may appear in various configurations making them difficult to segregate.

1.3 The DynamiX Project

The research in this thesis is part of the larger interdisciplinary DynamiX project within the SystemsX.ch consortium.¹ SystemsX is the the research initiative underway in Switzerland with the mandate of promoting Systems Biology.

The aim of the DynamiX project is to advance in the state-of-the-art of protein biochemistry and live cell imaging by applying highly-integrated microfluidic devices, advanced image processing, and computational biology to two central aspects of cell function: the cell cycle and growth control. Together these methods enable a single scientist to gather thousands of precision measurements on protein expression dynamics, promoter architecture, or molecular interactions in a single experiment. The measurements provide insights into network function on all levels, including cis-regulatory networks, transcriptional regulatory networks, and protein expression dynamics, thus permitting the development

1. <http://www.systemsx.ch/>

1.4. Development of Principalized Image Analysis Tools for Systems Biology

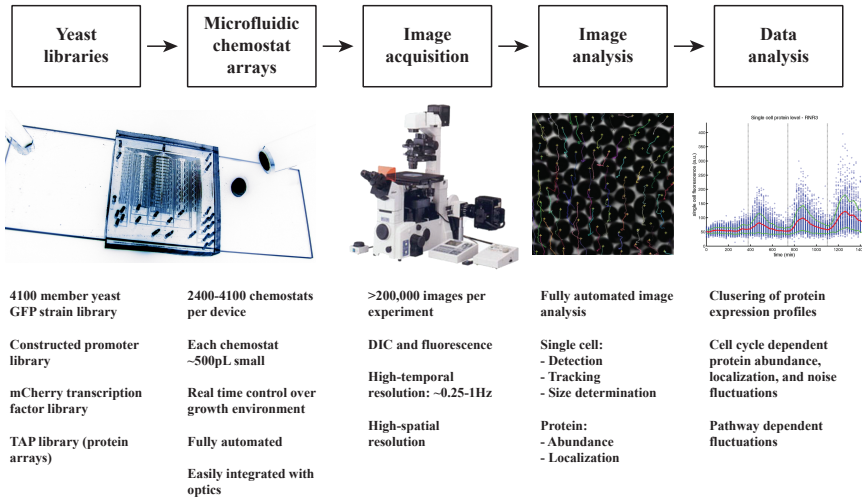


Figure 1.1: Description of the DynamiX work-flow. Microfluidic devices are programmed with yeast libraries and imaged on an automated microscope. The acquired images are then analyzed and the resulting data interpreted. The images on the bottom show, from left to right, a highly-integrated microfluidic device, yeast cells grown in pico-chemostats using device on left, a fully automated fluorescence microscope, a yeast cell expressing a GFP tagged protein, a colony of yeast cells and their trajectories, and a statistical temporal analysis of the colony of yeast cells expressing RNR3.

of quantitative models of specific sub-network function, such as ribosome biogenesis or DNA damage response. The final goal of the project is to decipher promoter architecture to understand how a given promoter DNA sequence regulates gene expression levels. We show in Figure 1.1 the general pipeline of the DynamiX project from the image acquisition process to the final data analysis. Within the DynamiX project, we tackle the block of image analysis shown in Figure 1.1.

1.4 Development of Principalized Image Analysis Tools for Systems Biology

Many algorithms exist in the literature that perform cell detection, segmentation or tracking. We review the state of the art with regard of this matter in Chapter 2. These algorithms are usually ad-hoc, and strongly dependent upon the acquisition technique. This makes it difficult to reuse image analysis tools

from different imaging modalities, or even tools developed for the same imaging modality, but designed to detect objects with clear morphological differences (e.g., an elliptic detector designed to segment cell nuclei is not suited to segment chromosomes even though both can be imaged with fluorescence microscopy).

In this thesis, we aim at designing tools that can be used in a variety of situations and are easily extensible to a broad range of imaging modalities. For that reason, our approach to design segmentation and tracking algorithms is principled, in contrast to many of ad-doc designs that can be found in the literature. Our aim is two-fold: first of all, we want our methods to be rooted in a consistent and flexible framework in which our algorithms can be properly analyzed. This will enable us to derive optimality results and make concrete statements about the efficiency and performance of the derived algorithms. Moreover, it will also provide a clear methodology to extend or adapt our algorithms to new imaging modalities. Secondly, we want to produce useful tools for the bioimaging community. Thus, special attention is given to the user-friendliness and interactivity of our resulting software.

Usually, the general perception of the shape of an object is independent of its location, orientation and size [23]. These abstract attributes, which seem to come from Plato's world of Ideas, can be made precise using the appropriate formalism. We identify the areas of functional analysis and differential geometry as the ones that provide an elegant methodology to design shape descriptions with explicit parametrization. Moreover, computational geometry provides us with strategies to implement the shape descriptors and create routines that perform quantitative analyses of biological images. We use the machinery of these disciplines to identify the strengths and limitations of the classical B-spline representation model.

1.5 Overview of the Contributions

Specifically, this thesis addresses the problem of detecting and segmenting biological objects and tracking them over time in high-throughput microscopy. This is far from trivial due to the fact that, in biology, the objects of interest are usually indistinguishable from each other and can appear tightly packed and in various configurations making them difficult to segregate. We focus on objects that have a constrained shape and move according to specific patterns. It seems natural, then, to approach the detection, segmentation and tracking problems with model-based techniques that enforce the topology.

We revisit the spline-based framework for active contours, and tailor it to cell segmentation. We identify the key elements of the framework that have an important impact on the efficiency, speed, robustness, and usability. We address them providing optimal solutions derived from functional analysis. The most

important design factor throughout the thesis is the efficiency of the derived algorithms. For that, a proper definition of shape descriptors is imperative. Every module of the segmentation and tracking algorithms must be efficient in order to be able to apply them to high-throughput microscopy.

The main contributions of this thesis can be summarized as

- The proposal of a parametric curve representation model using generalized B-splines that can perfectly replicate ellipses as well as higher-order algebraic curves. For that purpose, we fully characterize the family of basis functions with shortest support that allows one to reproduce exponential polynomials. We show that the minimal-support of these functions has a crucial role in terms of efficiency.
- A new 2D segmentation method using our B-spline parametric curve model. It is versatile enough to provide a good approximation of any closed curve in the plane. Furthermore, its most important feature is that it can perfectly generate circular and elliptical shapes. These features are appropriate to delineate cross sections of cylindrical-like conduits and to outline blob-like objects.
- The extension of our 2D segmentation method to 3D obtaining a fully parametric B-spline surface model with a sphere-like topology. This surface can approximate any blob-like structure with arbitrary precision and reproduce spheres and ellipsoids perfectly.
- A framework that is capable of generating fast and intuitive interactions of the user with the segmentation algorithms due to our B-spline representation of the 2D and 3D segmentation methods. The modification of one parameter in the model affects a limited region of the active curve/surface, which allows us to provide feedback to the user in terms of live updating display.
- The design of an image analysis toolkit that performs large-scale spatio-temporal analysis of mitotic cells using our segmentation algorithms as building blocks. This is possible due to the efficiency of each individual segmentation routine and possibility of high level of parallelization.

1.6 Organization of this Thesis

The thesis proceeds with a review in Chapter 2 of different segmentation methods, among which we highlight the active contours framework. Special effort is given to categorize methodologies that have emerged from this framework using different shape representations.

We present in Chapter 3 the mathematical concepts that are used extensively throughout the work. Special attention is given to the parametric representation of curves in a basis composed of integer shifts of a generating function (*i.e.*,

uniform B-spline representation). We also prove our optimality theorems in which the methods designed in the subsequent chapters are based on.

In Chapter 4, we present a new class of parametric active contours using the special kind basis functions designed in Chapter 3. We force our bases to have the shortest possible support subject to some design constraints to maximize efficiency. While the resulting snakes are versatile enough to provide a good approximation of any closed curve in the plane, their most important feature is the fact that they admit ellipses within their span. Thus, they can perfectly generate circular and elliptical shapes. We address the implementation details and illustrate the capabilities of our snake with synthetic and real data.

An extension to 3D active contours is presented in Chapter 5. We introduce a fully parametric 3D design relying on the basis functions of Chapter 3. Once more, we design our bases to have the shortest possible support subject to some constraints that maximize computational efficiency. The proposed 3D snake can approximate blob-like objects with \mathcal{C}^1 smoothness, with good accuracy and can perfectly reproduce spheres and ellipsoids irrespective of their position and orientation. The optimization process is remarkably fast for a volumetric method thanks to the use of Gauss' theorem within our energy computation scheme. Our technique yields successful segmentation results, even for challenging datasets where object contours are not well defined. This is due to our parametric approach that allows us to favor prior shapes.

Finally, in Chapter 6, we make use of the active contours designed in previous chapters to design a segmentation and tracking method that performs large-scale time-lapse analysis of mitotic cells. The demonstrated efficiency of our active contours allows us to use them as building blocks in a highly parallelizable image analysis toolkit.

Chapter 2

State-of-the-art in Image Analysis

In this chapter, we first review the state of the art in image-analysis algorithms. Then, we address how implementation standards allow one to provide users with appropriate tools.

First, we focus on image segmentation. We review different approaches, among which we highlight the active contours framework. Then, we present the current paradigm in image analysis for software development.

We describe the current good practices of the bioimage analysis community. To conclude, we briefly review the most popular open image analysis available platforms.

2.1 A Survey on Segmentation

The first step of Biomedical image analysis is often to identify objects in images that are relevant to a specific application. These objects are typically anatomical structures (*e.g.*, organs, vessels or other conduits) in medical imaging, and different cell structures in automated microscopy.

In image processing and computer vision, the process of separating the desired object (or objects) of interest from the background in an image is called segmentation. More precisely, it is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. Ultimately, the goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful to analyze.

A variety of techniques can be used to do this. The literature contains hundreds of segmentation techniques [24]. They range from simple pixel-wise operations (such as thresholding or masking) to more complex continuous models (such as active contours). There is no single method that can be considered good for all

images, nor are all methods equally good for a particular type of image.

Segmentation methods vary depending on the imaging modality, application domain, the level of automation, and other specific factors. Beside manual segmentation, the simplest method for separating objects from a background is *intensity thresholding* [25, 26, 27]. This involves defining one or several threshold parameters whose value can be set manually or derived automatically from the data based on the intensity histogram [25]. This approach would be successful if the objects to segment and the background are well separated and their intensity levels differ significantly from each other. Unfortunately, these methods do not maintain object integrity since they do not include neighborhood relations, and are also sensitive to noise.

A more elaborated approach consists in using a predefined *intensity profile*, also referred to as a *template*, to be matched to the image data [28, 29]. This method has been shown to work well as long as the shape to segment does not change significantly across different experiments (*i.e.*, rigid transformations) [28]. In order for the algorithm to gain flexibility and generality, a large number of different templates must be considered. This usually makes a good algorithm design impractical [30].

Another segmentation strategy is to apply the *watershed transform* [31]. The image is considered as a topographic relief map and progressively flooded starting from its local minima. This transform subdivides the image into regions (catchment basins) with delimiting contours (watersheds). However, the basic algorithm has several drawbacks such as sensitivity to noise and a tendency toward oversegmentation [32].

Finally, the most used statistical segmentation methods rely on Markov Random Field Models (MRF). MRF modeling itself is not a segmentation method but a statistical model that can be used within segmentation methods. MRFs model spatial interactions between neighboring pixels [33]. MRFs are often incorporated into clustering segmentation algorithms such as the K-means [34, 35, 36]. The segmentation is then obtained by maximizing the *a posteriori* probability of the segmentation, given the image data. The major difficulty associated with MRF models is the proper selection of the parameters controlling the strength of spatial interactions [33]. Moreover, MRF-based segmentation methods are usually computationally intensive algorithms.

2.1.1 Snakes, a Perfect Fit for Image Segmentation

In recent years, there has been an increasing interest in using deformable models in segmentation [37, 38, 39, 40] since they provide the best tradeoff between flexibility and efficiency. Within this category, active contours (also named snakes) are the most popular tools for image segmentation. More precisely, an

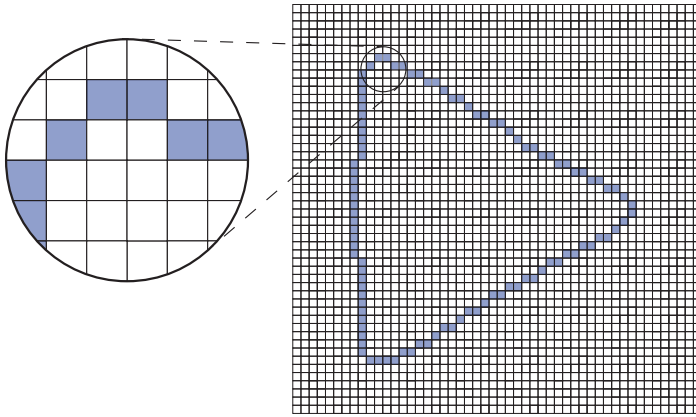


Figure 2.1: Graphical representation of the discrete curve defined by a point-snake over the grid associated to a discrete image model. The shaded pixels represent the snake points joined by an 8-connected digital topology.

active contour is a curve within a 2D image that evolves from an initial position towards the boundary of the object of interest. Its extension to 3D images is an evolving surface. The initial position of the snake is usually specified by the user, or it is provided by an auxiliary detection algorithm. The evolution of the snake is formulated as a minimization problem; the associated cost function is usually referred as *snake energy*. Snakes have become popular because it is possible for the user to interact with them, not only when specifying its initial position, but also during the segmentation process. This interaction is usually implemented by allowing the user to specify control points the snake must go through.

Research in this area has been fruitful and has resulted in many snake variants [37, 38, 41, 42]. They differ in the type of representation and in the choice of the energy term. In the rest of this chapter we provide a categorization of snakes in terms of representation, and offer a description of the overall snake energies.

2.1.1.1 Snake Representations Snakes can be broadly categorized based on the type of representation used:

1. *Point-snakes*. These snakes are based on the simplest representation of discrete curves (or surfaces): by using an ordered collection of points [43, 44, 45]. A pair of snake points are considered adjacent if some topological relations are satisfied [46]. In Figure 2.1, we show a 2D point-snake overlaid on the grid associated to a discrete image model. The discrete curve is displayed as gray pixels, and has an 8-neighbor connectivity.

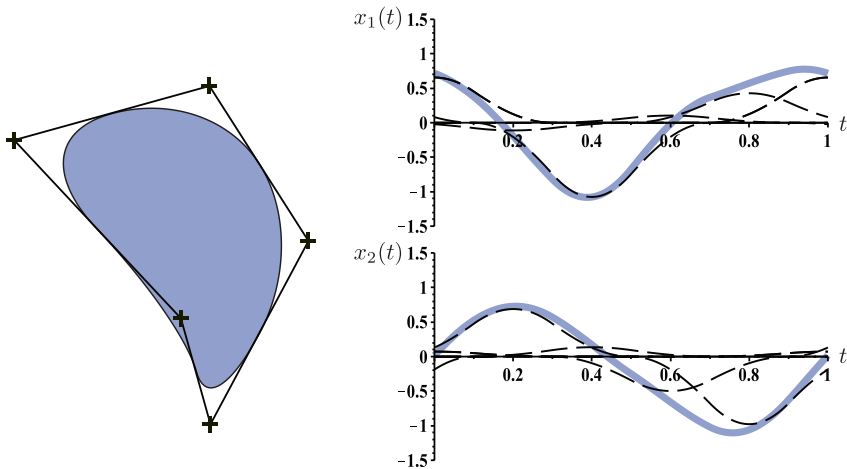


Figure 2.2: Graphical representation of the continuous curve defined by a snake parameterized by a B-spline basis. The snake contour is shown as a solid line enclosing a shaded region, while the '+' elements are the spline control points. The parametric functions $x_1(t)$ and $x_2(t)$ are displayed in solid lines, and the dashed lines indicate the weighted basis functions.

This approach does not ensure smoothness of the contour due to the discrete nature of the representation. However, some degree of smoothness (in a discrete sense) is usually introduced by adding extra constraints in the energy functional [43]. This discrete representation requires many parameters to encode a simple shape (two for each snake point in 2D and three for each snake point in 3D). The large number of parameters to be estimated abate the robustness of the overall segmentation algorithm, and results in a high computational complexity.

2. *Parametric Snakes.* The snake is described continuously by some coefficients [47, 48, 49, 50, 51]. Parametric snakes are usually built in a way that continuity and smoothness are ensured. This provides the algorithm the capability to segment at an arbitrary resolution, which may yield subpixel accuracy. Moreover, when compared with the discrete approaches, they require much fewer coefficients and result in faster optimization schemes than their discrete counterpart.

There are many different techniques for representing continuous curves. For a complete review, refer to [52, 53]. In computer graphics, curves and surfaces are often represented using non-uniform or uniform B-spline functions [54] and, more recently, NURBS (Non-Uniform Rational B-Splines)

[55]. NURBS are the preferred approach in computer graphics since these functions are closed under perspective transformations. On the other hand, curve and surface parameterizations based on Fourier descriptors [48, 56] and uniform B-spline functions [50, 57, 58] are popular in image processing due to the existence of efficient signal-processing algorithms, and their invariance to affine transformations. Of these, the B-spline curves have the extra advantage of locality of control which favors a more user-friendly interaction: a change in one of the snake points will only affect a small region of the curve or surface. We show in Figure 2.2 a curve parameterized with a B-spline basis, its spline control points as well as its corresponding coordinate functions. We discuss the B-spline representation of curves in detail in the next chapter.

In the case of B-spline parameterizations, it can be shown that the computation complexity of the snake energy and, therefore, the speed of the optimization algorithms, is related to the size of the support of the basis functions [57]. It is therefore critical to minimize this support while designing snakes.

Since the curve or surface of parametric snakes is represented explicitly, it is easy to introduce smoothness and shape constraints [47]. It is also straightforward to accommodate user interaction. This is often achieved by allowing the user to specify some anchor points the curve should go through [43]. The downside of the method is that the topology of the curve is imposed by the parameterization. This makes parametric snakes less suitable for handling topological changes, although solutions have been proposed for specific cases [59, 60].

3. *Geodesic Snakes*. Geodesic approaches have obtained a lot of attention during the last decade [61, 62, 63, 64]. The representation of these snakes is implicit and described as the zero level-set of a higher-dimensional manifold. Formally, the snake contour is given by $\Phi^{-1}(0) = \{\mathbf{p} \in \mathbb{R}^n | \Phi(\mathbf{p}) = 0\}$, where Φ is a scalar function defined all over the image domain. This method is based on the ideas developed by Osher and Sethian to model propagating solid/liquid interfaces with curvature-dependent speeds [65]. The interface (front) is a closed, nonintersecting, hypersurface flowing along its gradient field with constant speed or a speed that depends on the curvature. It is moved by solving a Hamilton-Jacob type equation written for a function in which the interface is a particular level-set.

These methods offer great flexibility as far as the curve topology is considered. A single geodesic snake (evolving under the appropriate energy functional), has the ability to split freely to segment multiple objects within an image. This flexibility is convenient when segmenting complex shapes, which include shapes with significant protrusions, and to situations where no *a priori* assumption about the topology of the object is made [66]. More-

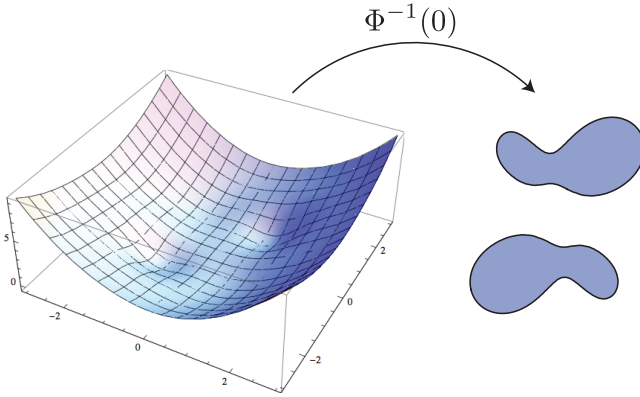


Figure 2.3: Graphical representation of the continuous curves generated as the result of computing the zero level-set ($\Phi^{-1}(0)$) of a scalar function Φ .

over, level-set methods can be extended to any dimension, which is more challenging for the case of point-snakes and parametric snakes.

However, they tend to be computationally more expensive since they evolve a manifold with a higher number of dimensions than the actual contour to segment. We show in Figure 2.3 a set of curves generated as the result of computing $\Phi^{-1}(0)$.

2.1.1.2 Snake Energies In this thesis we follow the standard paradigm introduced by Kass *et al.* [43] and formulate the snake evolution as an energy minimization. The snake energy is typically a linear combination of three terms:

- the *image energy*, which is the responsible for guiding the snake toward the boundary of interest;
- the *internal energy*, which ensures that the segmented region has smooth boundaries;
- the *constraint energy*, which provides a means for the user to interact with the snake.

The total energy of the snake is written as

$$E_{\text{snake}}(\Theta) = E_{\text{image}}(\Theta) + E_{\text{int}}(\Theta) + E_{\text{c}}(\Theta), \quad (2.1)$$

where Θ encodes the snake representation (snake points, parameters, or manifolds). Then, the optimal Θ is formally obtained as

$$\Theta_{\text{opt}} = \arg \min_{\Theta} E_{\text{snake}}(\Theta).$$

The energy minimization process is nothing but an optimization procedure, where we iteratively update the snake representation so as to reach the minimum of the energy function from a starting position. Many methods exist to minimize the energy functional (gradient descent, PDEs, DP, etc.), and each optimization scheme is usually linked to a particular snake representation.

The image energy is the most important of the three terms in (2.1) since is the one that guides the snake to the object of interest. Traditional snakes rely on edge maps derived from the image [43, 47]. These edge-based energies can provide a good localization of the contour of the object to segment. However, they have a narrow basin of attraction, making critical a good initialization. Traditional point-snakes and parametric snakes were very sensitive to initialization. This was in part due to the fact that the underlying internal energy of these methods was purely based on edge maps. Several authors have developed alternative solutions to this issue. Among them the most important ones are the introduction of balloon forces [67], the introduction of gradient vector-fields defined everywhere on the image domain [44], or multiresolution approaches [50].

More image energies use statistical information to distinguish different homogeneous regions [49, 68, 69]. The region-based energies have a larger basin of attraction and can converge even if explicit edges are not present [70]. However, it does not provide a good localization as the edge-based image energies.

The internal energy is responsible for ensuring the smoothness of the snake. In its original definition, it is composed by a linear combination of the length of the contour and the curvature of the snake [43]. Despite the fact that this particular expression was the first one to be introduced, it still corresponds to the most widely used [38]. Some authors also incorporate prior knowledge as shape constraints in this energy [71].

The constraint energy provides a means for the user to interact with the snake. Usually, this is obtained by introducing an energy functional that behaves as virtual springs that pull the snake towards the desired points [57]. Some implementations obviate the constraint energy while accommodating the user interaction as a hard constraint and leaving the parameters related to the point out of the optimization routine [72].

An alternative minimization framework to (2.1) is the multipurpose Mumford-Shah functional [73]. In this framework, the image is modeled as a piecewise-smooth function. The functional penalizes the distance between the model and the input image, the lack of smoothness of the model within the sub-regions, and the length of the boundaries of the sub-regions. This approach is quite popular in the context of geodesic snakes [70, 74, 75, 76].

The quality of segmentation is determined by the choice of the energy terms; it is generally agreed that specific image energies need to be defined for each particular imaging device. For this reason, we define in the subsequent chapters particular energies for each application.

2.2 Image Analysis Software

The step for converting algorithms to good and usable bioimage analysis software is also of great importance. In this section, we present the current good practices for software development to ensure a successful conversion, and we review the history and current state of the most popular open image analysis platforms.

2.2.1 Software Design

The primary users of image analysis software are biologists with little or no programming training and who are operating their own microscopes and analyzing their own data. They require user-friendly, well-supported, and flexible software to easily fulfill their particular needs [77].

It is generally agreed that the following good practices must be followed in order to create software that is usable and helpful to a broad segment of bioimaging community [78]:

1. *User-friendliness*: The software should be intuitive and easy-to-use. Moreover, it should be accompanied with clear usage manuals and offer feedback mechanisms (e.g., forums, mailing lists, bug report systems) [79]. We show in Figure 2.4 an intuitive interface of an image analysis software running in a Tablet PC.
2. *Developer-friendliness*: A good documentation of the structure of the code is crucial since it provides developers the capability to understand what and how the program works. Open-source software is a good example of developer-friendly software.
3. *Interoperability*: It is important to make software that communicates using the available open standards. In this way, different software can easily interact without having to define complementary components to translate the data. A successful example is the Bio-Formats project, a Java library for reading and writing life sciences image file formats [80].
4. *Modularity*: The implicit modularity of object-oriented design is key when maintaining a large piece of software. The use of modular structures with common interfaces allows developers to update their software at a minimum effort.



Figure 2.4: Samsung Slate PC Series 7 running the open image analysis software Icy [81] and one of the plug-ins described in this thesis. This is the result of the efforts of the open-source community of developers to produce an user-friendly image analysis software.

5. *Validation and Quality Control*: The software should be tested in ways that are relevant to the user. Moreover, for the benefit of making research reproducible, it must be possible to replicate the same exact computations and quantitative results that the developers advertise.

2.2.2 Open Image Analysis Platforms

The established paradigm in science is to ask and answer scientific questions by making observations and doing experiments. In order to properly analyze the experiments and draw conclusions, the scientist must be aware of how his tools work. Simply pressing a button in a piece of software and interpreting the results without understanding what the software does is obviously not good scientific practice. Open-source software provides the necessary transparency, giving scientists the opportunity to fully understand the computational methods behind their tools.

Among all open-source bioimage analysis tools, the one that has had the most impact so far is ImageJ [82]. It was initiated by Wayne Rasband at the National Institutes of Health (NIH) under the name of NIH Image. The idea was to develop a low-cost image-processing platform for the Apple Macintosh II. This piece of software was coded in Pascal, and had add-on capabilities in the form of

	Initiated	Status	Language	License
NIH Image ¹	1987	Discontinued	Pascal	Public domain
ImageJ ²	1997	Active	Java	Public domain
μ Manager ³	2005	Active	C++	BSD, Lesser GPL
CellProfiler ⁴	2006	Active	Python	GNU
Fiji ⁵	2007	Active	Java	GNU
ImageJ2 ⁶	2009	Under development	Java	Simplified BSD
Icy ⁷	2011	Active	Java	GPL

Table 2.1: Summary of open-source image-processing platforms.

expansion slots in order to enable other developers to easily extend the software for their own applications.

In the mid-nineties, the programming language Java was created by Sun Microsystems. Java applications are typically compiled to bytecode that can run on any machine regardless of the architecture. This allowed developers to write their software independently of the platform. Rasband ported NIH Image to Java in the late-nineties under the name of ImageJ. As a result the base of NIH Image users and developers was extended to PC and Unix.

ImageJ upgraded the expansion slots of NIH Image into the more modular concept of plug-ins. Since its creation, ImageJ has enjoyed a great popularity, and resulted in the development of a wide variety of plug-ins for very diverse applications [83].

Besides the core application, another popular distribution is Fiji. It is a more user-friendly distribution of ImageJ together with Java, Java 3D and the most prominent plug-ins as well as transparent installation and updates [84].

The largest upgrade of ImageJ since NIH Image is being prepared involving several research laboratories under the name of ImageJ2. It involves a full rewrite of the source code using new architectures in order to overcome the limitations of ImageJ.

Recently, other open-source related platforms are emerging. Among them, we can find: μ Manager, a software package for the control of automated microscopes [85]; CellProfiler, a software specialized in measuring phenotypes automatically within images [86]; and Icy, a full integrated easy-to-use platform

1. <http://rsb.info.nih.gov/nih-image/>

2. <http://rsbweb.nih.gov/ij/>

3. <http://www.micro-manager.org/>

4. <http://www.cellprofiler.org/>

5. <http://fiji.sc/>

6. <http://developer.imagej.net/>

7. <http://icy.bioimageanalysis.org/>

extensible with plug-ins [81]. We summarize all these open-source projects in Table 2.1 [87].

Due to the possibility that all the aforementioned image-processing packages diverge and interoperability becomes an issue, the Open Bio Image Alliance⁸ (OBIA) was constituted in 2012. Its primary mission is to *provide biologists and researchers in the life sciences with the highest quality public-domain software resources and a corresponding knowledge base to analyze and quantitate their image data in a sound and reproducible fashion, and to strengthen the interaction between biologists, imaging scientists and developers of bio-image analysis software and algorithms.*

OBIA capitalizes on the existence of highly successful software packages such as ImageJ. However, it also faces substantial challenges relating to the long-term support of existing software, its improvement, the quantity and diversity of available plug-ins, the documentation and organization of the modules, as well as compatibility issues. OBIA promotes long-term availability and backward compatibility, federates the harmonious community-based development of interoperable software, and promotes good software development practices.

The methods described in this thesis have been programmed as plug-ins for ImageJ and Icy. Both are free open-source multi-platform Java image-processing platforms. Our plug-ins are independent of any imaging hardware and, thanks to ImageJ and Icy, any common file format may be used. The plug-ins and the source codes are freely available at the respective official repositories.

8. <http://www.openbioimage.org/>

Chapter 3

Spline Bases for Representation of Curves

The generation of curves under geometric restrictions is an important area of research in Computer-Aided Geometric Design (CAGD). Considerable effort has been expended over the last forty years in this field in order to develop efficient and flexible representations of complex shapes. Since Bézier curves in the early sixties, B-spline curves in the mid seventies, and subdivision schemes in the late seventies, the search for representations that overcome the topological limitations of the classical approaches has not ceased. Research in this area has been fruitful and has resulted in many different methodologies [88, 89]. They can be broadly categorized in terms of curve representation as

- subdivision schemes, where the curve is described as the limit of a refinement process [90, 91, 92];
- parametric schemes, where the curve is described continuously by some coefficients using basis functions [53, 93, 94, 95].

A subdivision scheme is a set of rules that recursively define new points on finer grids starting from a set of initial points on a coarse grid. If the same rule is kept for all iterations, the scheme is called stationary [96, 97, 98]. If a different rule is used at each refinement level, the scheme is called nonstationary [99, 100]. Research is continually moving toward the investigation of refinement rules able to combine desirable reproduction properties under some geometrical constraints. In particular, schemes capable of reproducing circles were proposed in [101, 102, 103, 104, 105], and, more recently, schemes based on exponential B-splines made possible the reproduction of conic sections [106, 107, 108, 109, 110] and exponential polynomials [111, 112].

For certain applications, it is more convenient to represent the curve in an explicit parametric form instead of representing it as the limit of a subdivision

process, the reason being that the parameters provide a direct way of evaluating any point on the curve. For computational reasons, short basis functions are preferable because the evaluation of a single point on the curve then depends on fewer coefficients.

In this chapter, we design a parametric curve representation model that can perfectly replicate ellipses as well as higher-order algebraic curves. To achieve this, we select basis functions that have the capability of reproducing specific families of exponential polynomials. We prove a factorization theorem that links the reproduction properties of a given basis function and its support. The theorem shows that any compact-support basis function that reproduces that subspace can be expressed as the convolution of an exponential B-spline and a compact-support distribution. As a corollary of this result, we obtain a full characterization of the minimal-support basis functions with the required reproduction properties; these basis functions were first identified by Ron using a different approach [113]. This explicit characterization gives us the opportunity to identify interesting candidates within the family, and to construct nonstationary subdivision schemes that share the same reproduction properties.

This chapter is organized as follows: In Section 3.1, we state the general parametric curve model through an expansion with compact-support basis functions, and discuss the requirements these bases should fulfill. In Section 3.2, we construct a family of basis functions that reproduce exponential polynomials and prove that these bases have minimal support. In Section 3.3, we exhibit the multiresolution properties of our basis functions and propose a subdivision scheme that shares the same reproduction properties within the family. Finally, we illustrate the versatility of our model in Section 3.4 by identifying a basis from the family that contains ellipses and higher-order harmonics within its span.

3.1 Parametric Curves

3.1.1 Generic Curves

A curve $\mathbf{r}(t)$ on the plane can be described by a pair of Cartesian coordinate functions $x_1(t)$ and $x_2(t)$, where $t \in \mathbb{R}$ is a continuous parameter. We choose to parameterize the one-dimensional functions x_1 and x_2 by linear combinations of suitable basis functions. Among all possible bases, we focus on those derived from a compactly supported generator and its integer shifts $\{\varphi(\cdot - k)\}_{k \in \mathbb{Z}}$. This allows us to take advantage of fast and stable interpolation algorithms [114, 115, 116]. The parametric representation of the curve is then given by the vectorial equation

$$\mathbf{r}(t) = \sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi\left(\frac{t}{T} - k\right), \quad (3.1)$$

where $\{\mathbf{c}[k]\}_{k \in \mathbb{Z}}$ is a sequence of control points and T a sampling step.

We want our parametric curve to be defined in terms of the coefficients in such a way that unicity of representation of the coordinate functions x_1 and x_2 is satisfied. Furthermore, for computational purposes, we ask the interpolation procedure to be numerically stable. A generating function φ is said to satisfy the Riesz-basis condition if and only if there exist two constants $0 < A \leq B < \infty$ such that

$$A \|\mathbf{c}\|_{\ell_2(\mathbb{Z})}^2 \leq \left\| \sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(\cdot - k) \right\|_{L_2(\mathbb{R})}^2 \leq B \|\mathbf{c}\|_{\ell_2(\mathbb{Z})}^2, \quad (3.2)$$

for all $\mathbf{c} \in \ell_2(\mathbb{Z})$. A direct consequence of the lower inequality is that the condition $\sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(\frac{t}{T} - k) = 0$ for all $t \in \mathbb{R}$ implies that $\mathbf{c}[k] = 0$ for all $k \in \mathbb{Z}$. Moreover, $\mathbf{c}[k] = 0$ for all $k \in \mathbb{Z}$ trivially implies that $\sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(\frac{t}{T} - k) = 0$ for all $t \in \mathbb{R}$. Therefore, the basis functions are linearly independent and every function is uniquely specified by its coefficients. Moreover, the upper inequality ensures the stability of the interpolation process [116, 117]. Condition (3.2) can be expressed [117] in the Fourier domain, where the following equivalent form must hold for every $\omega \in \mathbb{R}$:

$$A \leq \sum_{n=-\infty}^{\infty} |\hat{\varphi}(\omega + 2\pi n)|^2 \leq B.$$

The curve model in (3.1) has been shown to be very versatile since it can approximate any curve when the sampling step T decreases while keeping the same basis function φ . The minimum requirement for this to happen is that φ should be able to reproduce constants, which we formalize by

$$\sum_{k=-\infty}^{\infty} \varphi(\cdot - k) = 1. \quad (3.3)$$

In the literature of approximation theory, this constraint is often named the *partition-of-unity* condition [118].

3.1.2 Closed Curves

We are especially interested in the case when \mathbf{r} is closed. In this context, the two coordinate functions are periodic, with the same period. We normalize it to unity so that $\mathbf{r}(t) = \mathbf{r}(t + 1)$ for all $t \in \mathbb{R}$, and divide it into M segments, which is equivalent to choosing the sampling step $T = \frac{1}{M}$. Under these conditions, we can reduce the infinite summation in (3.1) to a finite one with M terms involving

periodized basis functions. We write

$$\begin{aligned}
 \mathbf{r}(t) &= \sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(M t - k) \\
 &= \sum_{n=-\infty}^{\infty} \sum_{k=0}^{M-1} \mathbf{c}[M n + k] \varphi(M (t - n) - k) \\
 &= \sum_{k=0}^{M-1} \mathbf{c}[k] \underbrace{\sum_{n=-\infty}^{\infty} \varphi(M (t - n) - k)}_{\varphi_{\text{per}}(M t - k)}, \tag{3.4}
 \end{aligned}$$

where M is the number of control points, the sequence $\{\mathbf{c}[k]\}_{k \in \mathbb{Z}}$ is M -periodic, and φ_{per} is the M -periodization of the basis function φ . In the periodic setting, it has also been shown that this parametric curve model is very versatile [119], and we can approximate any closed curve as accurately as we want by increasing the number of control points M . Under some mild refinability conditions, it has been shown that this model naturally leads to a stationary subdivision scheme [90].

3.1.3 Desirable Properties of Bases in the Periodic Settings

Now, we enumerate the conditions that our parametric closed curve model should satisfy, and introduce the corresponding mathematical formalism.

1. *Unique and Stable Representation.* We want our closed parametric curve to be defined in terms of the coefficients in such a way that unicity of representation is satisfied, and we want the interpolation procedure to be numerically stable. A generating function φ is said to satisfy the periodic Riesz-basis condition if and only if there exist two constants $0 < A \leq B < \infty$ such that

$$A \|\mathbf{c}\|_{\ell_2([0 \dots M-1])}^2 \leq \left\| \sum_{k=0}^{M-1} \mathbf{c}[k] \varphi_{\text{per}}(M \cdot -k) \right\|_{L_2([0,1])}^2 \leq B \|\mathbf{c}\|_{\ell_2([0 \dots M-1])}^2 \tag{3.5}$$

holds true for all M -periodic and bounded sequences \mathbf{c} . The interpretation of this condition is in all points similar to the non-periodic case. We also note that (3.5) is automatically satisfied if φ_{per} is defined as in (3.4), and (3.2) holds true for φ .

2. *Affine Invariance.* Since we are interested in representing shapes irrespective of their position and orientation, we would like our model to be invariant

to affine transformations, which we formalize as

$$\mathbf{A}\mathbf{r}(t) + \mathbf{b} = \sum_{k=0}^{M-1} (\mathbf{A}\mathbf{c}[k] + \mathbf{b}) \varphi_{\text{per}}(Mt - k), \quad (3.6)$$

where \mathbf{A} is a (2×2) matrix and \mathbf{b} is a two-dimensional vector. From (3.6), it is easy to show that the affine invariance is satisfied if and only if

$$\sum_{k=0}^{M-1} \varphi_{\text{per}}(M \cdot -k) = 1. \quad (3.7)$$

This last equality is a direct implication of the partition-of-unity condition stated in (3.3).

3.1.4 Approximation and Reproduction Properties in Periodic Settings

The parametric closed-curve model (3.4) can be used to approximate any closed curve \mathbf{s} as accurately as desired by increasing the number of control points M . Formally, we write that

$$\lim_{M \rightarrow \infty} \|\mathbf{s} - \mathcal{P}_M \mathbf{s}\|_{L_2([0,1])} = 0,$$

where $\mathcal{P}_M \mathbf{s}$ denotes a projection of \mathbf{s} onto $\{\varphi(M \cdot -k)\}_{k \in \mathbb{Z}}$, or, equivalently, onto $\{\varphi_{\text{per}}(M \cdot -k)\}_{k=0 \dots M-1}$, since both allow for alternative representations of the same space. In order to be able to select a suitable basis function, it is important to know the rate at which the error decreases as a function of M . The open-curve case reduces to the well-known Strang-and-Fix framework in approximation theory [120, 121], the results of which are transposable to the closed-curve case as well [119].

In addition to desirable approximation properties, our main interest lies in the situation where the curve \mathbf{r} can reproduce desirable shapes exactly. For this purpose, we select for each $M \geq M_0$ a specific basis function capable of reproducing the shapes of interest with M vector coefficients, and denote it φ_M . Its M -periodization is written as $\varphi_{M,\text{per}}$. Using a different basis function φ_M for each value of M obviously leads to a subdivision scheme that is nonstationary. The existence of such a scheme depends on some refinability conditions over φ_M . In particular, the conditions of Section 3.1.3 have to hold for each φ_M individually.

In the nonstationary case, the approximation error of curve is $\|\mathbf{s} - \mathcal{P}_M \mathbf{s}\|_{L_2([0,1])}$, where now $\mathcal{P}_M \mathbf{s}$ denotes the projection of \mathbf{s} onto the space $\{\varphi_M(N \cdot -k)\}_{k \in \mathbb{Z}}$, with $N = M$. Inspired by [100], which discusses asymptotically equivalent binary subdivision schemes, we show in Section 3.4.1.5 that the rate of decay of the approximation error as a function of $N = M$ is equivalent to that of the stationary case.

3.2 Reproduction of Exponential Polynomials

The main aim of this section is to introduce a family of functions that reproduce exponential polynomials, and prove that these functions have minimal support. To achieve this goal, we start by formalizing the concept of the reproduction of exponential polynomials. Next, we define the exponential B-splines and list their relevant properties. This allows us to give a full parameterization of the family of functions of interest: they happen to be combinations of exponential B-splines and their derivatives. Note that, in this section, we consider spline functions on a cardinal grid on the real line. The case of periodic spline functions corresponding to closed curves follows directly from this theory by the argument given in Section 3.1.3, but the theory we develop here is more general and can also be used to design basis functions that reproduce non-periodic functions, for instance, open curves or surfaces.

3.2.1 Preliminary Definitions

A function P_α^N of the variable $t \in \mathbb{R}$ is called an *exponential polynomial* of degree N and exponent $\alpha \in \mathbb{C}$ when it takes the form

$$P_\alpha^N(t) = e^{\alpha t} \left(a[0] + \sum_{n=1}^N a[n] t^n \right), \quad (3.8)$$

where $\{a[n]\}_{n \in [0..N]}$ is a sequence of $(N + 1)$ complex coefficients such that $a[N] \neq 0$. A finite linear combination of exponential polynomials takes the form

$$\sum_{m=1}^M p[m] P_{\alpha_m}^{N_m}. \quad (3.9)$$

A generating function φ is said to *reproduce* a function f if and only if there exists a sequence $\{c[k]\}_{k \in \mathbb{Z}}$ such that

$$f(t) = \sum_{k=-\infty}^{\infty} c[k] \varphi(t - k)$$

holds almost everywhere.

3.2.2 Reproduction Conditions

A fundamental result in approximation theory is that there is an equivalence between the ability of a generating function to reproduce polynomials of a certain degree and the order of decay of the approximation error as the step size

goes to zero [122]. Strang and Fix showed in [121] that a generating function $\varphi \in L_2(\mathbb{R})$ has an approximation error that decays with order N if and only if

$$\int_{-\infty}^{\infty} \varphi(t) dt \neq 0$$

and there exists a finite constant $C_n \in \mathbb{C}$ such that

$$\sum_{k=-\infty}^{\infty} (t-k)^n \varphi(t-k) = C_n$$

holds for almost every $t \in \mathbb{R}$, and for $n \in [0 \dots N-1]$. Moreover, the generating function reproduces polynomials up to degree $(N-1)$.

An extension of the Strang-and-Fix conditions was presented by Vonesch *et al.* in [123] in the context of the reproduction of exponential polynomials. Here, we provide a reformulation suited to our needs.

Proposition 1. *A compact-support generating function $\varphi \in L_2(\mathbb{R})$ reproduces exponential polynomials of degree up to $(N-1)$ and exponent α if and only if*

$$\int_{-\infty}^{\infty} e^{-\alpha t} \varphi(t) dt \neq 0 \tag{3.10}$$

and there exists a finite constant $C_n \in \mathbb{C}$ such that

$$\sum_{k=-\infty}^{\infty} (t-k)^n e^{-\alpha(t-k)} \varphi(t-k) = C_n \tag{3.11}$$

holds for almost every $t \in \mathbb{R}$, and for $n \in [0 \dots N-1]$.

This proposition is a direct consequence of the generalized Strang-and-Fix conditions from [123] and the fact that φ is compactly supported.

Another way of approaching the problem is offered in [124] where the authors show that the reproduction properties of generating functions are preserved through convolution. We summarize here their proposition for completeness.

Proposition 2 (Unser and Blu, 2005). *Given a generating function φ_α that reproduces exponential polynomials of exponent α and degree up to N , then, for any ψ such that $\int_{-\infty}^{\infty} e^{-\alpha t} \psi(t) dt \neq 0$, the composite function $(\varphi_\alpha * \psi)$ also reproduces exponential polynomials of exponent α and degree up to N .*

The formulation proposed by the authors also requires two mild technical conditions over ψ and $(\varphi_\alpha * \psi)$ to ensure that moments are well-defined.

Proposition 2 provides a constructive procedure to build generating functions using simpler functions with known reproduction properties. In the next section, we present the exponential B-splines, which will provide us with the appropriate building blocks to reproduce exponential polynomials.

3.2.3 Exponential B-Splines

As their name suggests, exponential B-splines are the exponential counterpart of the well-known polynomial B-splines [124, 125, 126]. They have the property of reproducing exponential polynomials, polynomials being recovered as a particular case by setting $\alpha = 0$ in (3.8). An exponential B-spline of order N and poles $\alpha = (\alpha_1, \dots, \alpha_N)$ is defined in the Fourier domain as

$$\hat{\beta}_\alpha(\omega) = \prod_{m=1}^N \frac{1 - e^{-j(\omega - \alpha_m)}}{j\omega - \alpha_m}. \quad (3.12)$$

Note that the exponential B-splines are entirely specified by the collection α ; the ordering of the poles α_m is irrelevant. We illustrate in Figure 3.1 several exponential B-splines, where we see that a wide range of behaviors can be obtained by varying N and α .

The most relevant properties of exponential B-splines for our purposes are

- The exponential B-splines are always well-defined (*i.e.*, bounded and compactly supported), and form a Riesz basis if and only if $(\alpha_{m_1} - \alpha_{m_2}) \notin 2\pi j\mathbb{Z}$ for all pairs such that $m_1 \neq m_2$.
- Exponential B-splines of order N are compactly supported within the interval $[0, N]$.
- The convolution of two exponential B-splines yields another B-spline of augmented order

$$\beta_{\alpha_1} * \beta_{\alpha_2} = \beta_{\alpha_1 \cup \alpha_2}$$

where $(\alpha_1 \cup \alpha_2)$ is the concatenation of the elements of α_1 and α_2 .

- The exponential B-splines of first order with parameter (α) reproduce the exponential function with exponent α

$$e^{\alpha t} = \sum_{k=-\infty}^{\infty} e^{\alpha k} \beta_{(\alpha)}(t - k).$$

- Exponential B-splines reproduce exponential polynomials of degree up to $(N_{m_1} - 1)$ and exponent α_{m_1} if and only if α_{m_1} appears exactly N_{m_1} times in α and, for all other distinct α_{m_2} , we have that $(\alpha_{m_1} - \alpha_{m_2}) \notin 2\pi j\mathbb{Z}$.

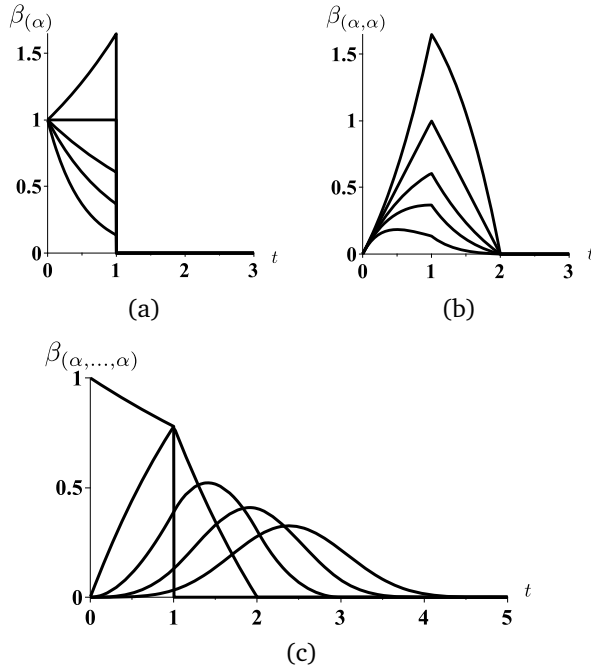


Figure 3.1: Examples of exponential B-splines. (a) First-order exponential B-splines with $\alpha \in \{(-2), (-1), (-\frac{1}{2}), (0), (\frac{1}{2})\}$. (b) Second-order exponential B-splines $\beta_{(\alpha, \alpha)}$ with $\alpha \in \{(-2, -2), (-1, -1), (-\frac{1}{2}, -\frac{1}{2}), (0, 0), (\frac{1}{2}, \frac{1}{2})\}$. (c) N -th order exponential B-splines $\beta_{(\alpha, \dots, \alpha)}$ with $\alpha = -\frac{1}{4}$ and $N \in [1 \dots 5]$.

The three last properties provide us with a constructive procedure for building generating functions capable of reproducing exponential polynomials of a given degree and exponent. By construction, the support of the resulting generating functions corresponds to the order of the exponential B-spline. We refer to [124] for additional aspects of exponential B-splines.

3.2.4 Distributional Decomposition

Our first goal is to characterize the functions that reproduce exponential polynomials. To that end, we are able to prove a converse version of Proposition 2; we prove that any compact-support function with the required reproduction properties must contain an exponential B-spline convolution factor with the same reproduction properties.

Theorem 1. Let φ be supported within $[a, b]$ and let it reproduce finite linear combinations of exponential polynomials (3.9) such that $(\alpha_{m_1} - \alpha_{m_2}) \notin 2\pi j\mathbb{Z}$ for $m_1 \neq m_2$. That is, φ satisfies (3.10) and (3.11) for each pair (N_m, α_m) . Then, a distribution ψ exists such that

$$\varphi = \beta_\alpha * \psi, \quad (3.13)$$

where ψ satisfies (3.10) for all α_m , each α_m appears N_m times in α , and ψ is compactly supported within $[a, b - N]$ with $N = \sum_{m=1}^M N_m$.

Proof. We proceed by induction over the order N_{m_1} of each α_{m_1} to show that we can factor out N_{m_1} times an exponential B-spline of first order for each α_{m_1} from the generating function φ . The process can be repeated for each exponent until the remaining kernel cannot reproduce any exponential polynomial anymore. Then, it is enough to show that, for a given α_{m_2} , there exists a distribution ψ such that

$$\varphi = \beta_{(\alpha_{m_2})} * \psi, \quad (3.14)$$

where ψ satisfies the following properties:

1. it is compactly supported within $[a, b - 1]$;
2. it reproduces exponential polynomials of degree up to $(N_{m_2} - 2)$ and exponent α_{m_2} ;
3. it reproduces exponential polynomials of degree up to $(N_{m_1} - 1)$ and exponent α_{m_1} for all $m_1 \neq m_2$.

Since the definition of ψ provided in (3.14) is implicit, we need to verify that this distributional kernel exists and is well-defined. We show this constructively. For a given $m_2 \leq M$, we define the function

$$\psi(t) = \sum_{k=0}^{\infty} e^{\alpha_{m_2} k} (D - \alpha_{m_2} I)\varphi(t - k), \quad (3.15)$$

where D is the derivative operator in the sense of distributions, and I is the identity. The infinite sum in (3.15) is well-defined since, for every t , the sum has only a finite number of elements because φ has compact support. From (3.15), we write that

$$\psi(t) - e^{\alpha_{m_2}} \psi(t - 1) = (D - \alpha_{m_2} I)\varphi(t). \quad (3.16)$$

Taking the Fourier transform of (3.16) leads to the factorization

$$\hat{\varphi}(\omega) = \frac{1 - e^{-(j\omega - \alpha_{m_2})}}{j\omega - \alpha_{m_2}} \hat{\psi}(\omega) = \hat{\beta}_{(\alpha_{m_2})}(\omega) \hat{\psi}(\omega)$$

which corresponds to the implicit definition of ψ given in (3.14).

To prove Point 1), we recall that φ reproduces exponential polynomials of degree up to $N_{m_2} - 1 \geq 0$ and exponent α_{m_2} . Thus, by setting $n = 0$ in (3.11) and applying the differential operator $(D - \alpha_{m_2} I)$, we have that

$$\sum_{k=-\infty}^{\infty} e^{\alpha_{m_2} k} (D - \alpha_{m_2} I) \varphi(t - k) = 0,$$

in the distributional sense. Thanks to this last equality and using the explicit formula of ψ given in (3.15), we can also write that

$$\psi(t) = - \sum_{k=-\infty}^{-1} e^{\alpha_{m_2} k} (D - \alpha_{m_2} I) \varphi(t - k).$$

According to this last expression, the support of ψ is contained within $(-\infty, b - 1]$. But, according to the definition (3.15), we also have that the support of ψ is contained within $[a, +\infty)$. Hence, we conclude that the support of ψ is contained within $[a, b - 1]$.

We deal with a modified version of (3.11) to prove Point 2). By linearity, and since φ reproduces exponential polynomials of degree up to $N_{m_2} - 1 \geq 0$ and exponent α_{m_2} , we can write that

$$\sum_{k=-\infty}^{\infty} P(t - k) e^{-\alpha_{m_2} (t-k)} \varphi(t - k) = C_P, \quad (3.17)$$

where P is any polynomial of degree no greater than $(N_{m_2} - 1)$, and C_P is a constant that only depends on the polynomial P and not on t . Then, the application of $(D - \alpha_{m_2} I)$ to (3.17) leads to

$$\begin{aligned} 0 &= \sum_{k=-\infty}^{\infty} P(t - k) e^{-\alpha_{m_2} (t-k)} \underbrace{(D - \alpha_{m_2} I) \varphi(t - k)}_{\psi(t-k) - e^{\alpha_{m_2}} \psi(t-k-1)} \\ &\quad + \underbrace{\sum_{k=-\infty}^{\infty} \dot{P}(t - k) e^{-\alpha_{m_2} (t-k)} \varphi(t - k)}_{C_{\dot{P}}}, \end{aligned}$$

where we have used (3.16) to rewrite the first term, and where the second term is equal to the constant $C_{\dot{P}}$ since \dot{P} is a polynomial of degree no greater than $(N_{m_2} - 2)$. Since ψ has a compact support, we can rearrange the terms as

$$\sum_{k=-\infty}^{\infty} Q(t - k) e^{-\alpha_{m_2} (t-k)} \psi(t - k) = -C_{\dot{P}}, \quad (3.18)$$

where $Q(t) = P(t) - P(t + 1)$.

Since P is a polynomial of degree no greater than $(N_{m_2} - 1)$, it follows that Q is a polynomial of degree no greater than $(N_{m_2} - 2)$. This also means that, for all polynomials Q of degree no greater than $(N_{m_2} - 2)$, there exists a constant C_Q such that $\sum_{k=-\infty}^{\infty} Q(t-k) e^{-\alpha_{m_2}(t-k)} \psi(t-k) = C_Q$. In particular, if $P(t) = t$, then $Q(t) = -1$. Because \dot{P} is a polynomial of degree lesser than that of P , it also satisfies (3.17). Then, we can substitute P by $\dot{P} = 1$ in (3.17), which we combine with (3.18) and $Q = -1$ to obtain the system

$$\begin{cases} \sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \varphi(t-k) = C_{\dot{P}} \\ -\sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \psi(t-k) = -C_{\dot{P}}, \end{cases}$$

which leads to

$$\sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \psi(t-k) = \sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \varphi(t-k).$$

Integrating the last expression of t over the interval $[0, 1]$, and rearranging the terms, yields

$$\int_{-\infty}^{\infty} e^{-\alpha_{m_2} t} \psi(t) dt = \int_{-\infty}^{\infty} e^{-\alpha_{m_2} t} \varphi(t) dt.$$

Thus, since φ satisfies (3.10), so does ψ . Therefore, ψ reproduces exponential polynomials of degree up to $(N_{m_2} - 2)$ and exponent α_{m_2} .

Finally, to prove Point 3), we proceed in the same manner. We recall that, for $m_1 \neq m_2$, the function φ reproduces exponential polynomials of degree up to $N_{m_1} - 1 \geq 0$ and exponent α_{m_1} . Thus, if we use (3.17) with parameter α_{m_1} and apply the differential operator $(D - \alpha_{m_2} I)$, then we obtain

$$\begin{aligned} (\alpha_{m_1} - \alpha_{m_2}) C_P = & \\ & \sum_{k=-\infty}^{\infty} P(t-k) e^{-\alpha_{m_1}(t-k)} \underbrace{(D - \alpha_{m_2} I) \varphi(t-k)}_{\psi(t-k) - e^{\alpha_{m_2} t} \psi(t-k-1)} \\ & + \underbrace{\sum_{k=-\infty}^{\infty} \dot{P}(t-k) e^{-\alpha_{m_1}(t-k)} \varphi(t-k)}_{C'_{\dot{P}}}, \end{aligned}$$

where we have used again (3.16) to rewrite the first term, and where the second term is equal to the constant $C'_{\dot{P}}$ since \dot{P} is a polynomial of degree no greater than

$(N_{m_1} - 2)$. Since ψ has compact support, we can rearrange the terms to obtain

$$\sum_{k=-\infty}^{\infty} Q(t-k) e^{-\alpha_{m_1}(t-k)} \psi(t-k) = (\alpha_{m_1} - \alpha_{m_2}) C_P - C'_P, \quad (3.19)$$

where $Q(t) = P(t) - e^{\alpha_{m_2} - \alpha_{m_1}} P(t+1)$.

Since P is a polynomial of degree no greater than $(N_{m_1} - 1)$, and $e^{\alpha_{m_2} - \alpha_{m_1}} \neq 1$, then Q is a polynomial of degree $(N_{m_1} - 1)$, too. This also means that, for all polynomials Q of degree no greater than $(N_m - 1)$, there exists a constant C_Q such that $\sum_{k=-\infty}^{\infty} Q(t-k) e^{-\alpha_{m_1}(t-k)} \psi(t-k) = C_Q$. In addition, we see that, if $P(t) = 1$, then $Q(t) = 1 - e^{\alpha_{m_2} - \alpha_{m_1}}$ and $C'_P = 0$. Now, by setting $P(t) = 1$ in (3.17) and $Q(t) = 1 - e^{\alpha_{m_2} - \alpha_{m_1}}$ in (3.19), we have the system

$$\begin{cases} \sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \varphi(t-k) = C_P \\ \sum_{k=-\infty}^{\infty} (1 - e^{\alpha_{m_2} - \alpha_{m_1}}) e^{-\alpha_{m_1}(t-k)} \psi(t-k) = (\alpha_{m_1} - \alpha_{m_2}) C_P, \end{cases}$$

which leads to

$$\sum_{k=-\infty}^{\infty} e^{-\alpha_{m_1}(t-k)} \psi(t-k) = \frac{\alpha_{m_1} - \alpha_{m_2}}{1 - e^{-(\alpha_{m_1} - \alpha_{m_2})}} \sum_{k=-\infty}^{\infty} e^{-\alpha_{m_2}(t-k)} \varphi(t-k).$$

Integrating the last expression of t over the interval $[0, 1]$, and rearranging the terms, yields

$$\int_{-\infty}^{\infty} e^{-\alpha_{m_1} t} \psi(t) dt = \frac{\alpha_{m_1} - \alpha_{m_2}}{1 - e^{-(\alpha_{m_1} - \alpha_{m_2})}} \int_{-\infty}^{\infty} e^{-\alpha_{m_2} t} \varphi(t) dt.$$

Thus, since φ satisfies (3.10) for α_{m_2} , so does ψ for α_{m_1} . Therefore, ψ reproduces exponential polynomials of degree up to $(N_{m_1} - 1)$ and exponent α_{m_1} . \square

3.2.5 Minimal-Support Generating Functions

As a direct consequence of Theorem 1, we show that appropriate combinations of exponential B-splines define the whole family of functions of minimal support that reproduce exponential polynomials. This family was first identified in [113] by independent means.

Theorem 2. *The size of the smallest-support kernel $\varphi \in L_2(\mathbb{R})$ that reproduces exponential polynomials of degree up to $(N_m - 1)$ and parameter α_m for $m \in \{1 \dots M\}$ is*

$$N = \sum_{m=1}^M N_m, \quad (3.20)$$

provided that $(\alpha_{m_1} - \alpha_{m_2}) \notin 2\pi j\mathbb{Z}$ for $m_1 \neq m_2$. Moreover, every minimal-support function φ can be written as

$$\varphi(t) = \sum_{n=0}^{N-1} \lambda_n \frac{d^n}{dt^n} \beta_{\alpha}(t-a), \quad (3.21)$$

where a is an arbitrary shift parameter that determines the lower extremity of the support of φ . In (3.21), each α_m appears exactly N_m times within the collection α and the collection of λ_n satisfies $\sum_{n=0}^{N-1} \lambda_n \alpha_m^n \neq 0$.

Proof. By Theorem 1, we can write

$$\varphi = \beta_{\alpha} * \psi,$$

where ψ is a distribution with support $[a, b-N]$ that satisfies (3.10) for all α_m . Finally, each α_m appears N_m times within the collection α . Conversely, if we take a distribution ψ that satisfies (3.10) for all α_m and is supported within $[a, b']$, then $\varphi = \beta_{\alpha} * \psi$ is supported within $[a, b'+N]$ and reproduces exponential polynomials of degree up to $(N_m - 1)$ and parameter α_m for $m \in [1 \dots M]$. Now, minimizing the support of φ means finding the smallest b such that ψ exists. Of course, this is possible only if $b' = b - N \geq a$, which yields ψ as a single-point distribution. This shows that the minimum size of the support of φ is $b - a = N$.

We know from distribution theory that the only distributions that have a support of zero-measure are finite linear combinations of the Dirac distribution and of its derivatives [127, Th. XXXV]. Thus, if φ has minimal support, then there exist constants λ_n such that

$$\psi(t) = \sum_{n=0}^{\infty} \lambda_n \delta^{(n)}(t-a). \quad (3.22)$$

This means that

$$\varphi(t) = \sum_{n=0}^{\infty} \lambda_n \frac{d^n}{dt^n} \beta_{\alpha}(t-a).$$

Since we restrict ourselves to $L_2(\mathbb{R})$, the summation has to run from 0 to $(N - 1)$.

Finally, since ψ satisfies (3.10) for all α_m , we have that

$$\begin{aligned}
 0 &\neq \int_{-\infty}^{\infty} e^{-\alpha_m t} \psi(t) dt && \text{by hypothesis} \\
 &= \int_{-\infty}^{\infty} e^{-\alpha_m t} \sum_{n=0}^{N-1} \lambda_n \delta^{(n)}(t) dt && \text{by (3.22)} \\
 &= \sum_{n=0}^{N-1} \lambda_n \int_{-\infty}^{\infty} e^{-\alpha_m t} \delta^{(n)}(t) dt && \text{by linearity} \\
 &= \sum_{n=0}^{N-1} \lambda_n \langle \delta^{(n)}(t), e^{-\alpha_m t} \rangle && \text{by definition} \\
 &= \sum_{n=0}^{N-1} \lambda_n \alpha_m^n,
 \end{aligned}$$

which proves the last result. □

3.2.6 Interpolator

It is also possible to constrain φ to be an interpolator. That is,

$$\forall k \in \mathbb{Z} : \varphi(t) \Big|_{t=k} = \delta[k].$$

Due to the size of the support of φ , the interpolation condition can add up to N constraints, depending on the value of a . This number of constraints matches the N degrees of freedom that result from the choice of λ_n in (3.21). A general study of the appropriate choice of λ_n to satisfy the interpolation condition lies out of the scope of this thesis. However, we propose a case-by-case approach that will be exemplified in Section 3.4.

3.3 Multiresolution and Subdivision

We have characterized the complete family of functions with minimal support that reproduce exponential polynomials in order to build parametric curves. In this section, we emphasize the connection with the subdivision world using the classical multiresolution properties of exponential B-splines. Moreover, we also specify another type of multiresolution scheme in terms of reproduction capabilities. In this section we focus on our case of interest: closed curves.

3.3.1 Classical Multiresolution of Exponential B-Splines

An important observation concerning the family of minimal-support basis functions in (3.21) is that it is constructed with exponential B-splines and their

derivatives of equal parameter α . Thanks to this property and under appropriate circumstances, the basis functions in (3.21) inherit the multiresolution properties of the exponential B-splines. It has been shown in [124, 128] that an exponential B-spline and its derivatives with parameter $\alpha = (\alpha_1, \dots, \alpha_N)$ satisfy the nonstationary set of dilation relations

$$\begin{aligned}
 \beta_\alpha\left(\frac{t}{2}\right) &= \sum_{k=-\infty}^{\infty} h_{\frac{\alpha}{2}}[k] \beta_\alpha(t-k) \\
 \frac{d}{dt} \beta_\alpha\left(\frac{t}{2}\right) &= 2 \sum_{k=-\infty}^{\infty} h_{\frac{\alpha}{2}}[k] \frac{d}{dt} \beta_\alpha(t-k) \\
 &\vdots \\
 \frac{d^n}{dt^n} \beta_\alpha\left(\frac{t}{2}\right) &= 2^n \sum_{k=-\infty}^{\infty} h_{\frac{\alpha}{2}}[k] \frac{d^n}{dt^n} \beta_\alpha(t-k), \tag{3.23}
 \end{aligned}$$

where $n \leq (N-1)$, $\frac{\alpha}{2} = (\frac{\alpha_1}{2}, \dots, \frac{\alpha_N}{2})$ is the collection of roots divided by 2, and $h_{\frac{\alpha}{2}}$ is the mask whose symbol is given by

$$H_{\frac{\alpha}{2}}(z) = \frac{1}{2^{N-1}} \prod_{m=1}^N \left(1 + e^{\frac{\alpha_m}{2}} z^{-1}\right).$$

3.3.2 Subdivision Scheme

We have now all the ingredients in hand to define a multiresolution hierarchy of spaces of closed curves. We define the spline space at resolution M as

$$V_{\alpha, M} = \left\{ \mathbf{r}(t) = \sum_{k=0}^{M-1} \mathbf{c}_M[k] \varphi_{M, \text{per}}(M t - k) \right\},$$

where M is the number of control points, and $\varphi_{M, \text{per}}$ is the M -periodization of (3.21) with defining parameter $\frac{\alpha}{M}$. Note that the parameters $\{\lambda_n\}_{n=0 \dots N-1}$, which are used to define $\varphi_{M, \text{per}}$ through (3.21) and (3.4), depend on M . In order to find the equivalent scaling expression for our generating function φ_M , we proceed in the Fourier domain where the explicit expression of φ_M in terms of exponential B-splines is

$$\hat{\varphi}_M(\omega) = \Lambda_M(j\omega) \hat{\beta}_{\frac{\alpha}{M}}(\omega) e^{-j\omega\alpha}. \tag{3.24}$$

There, the Fourier-domain function $\Lambda_M(j\omega) = \lambda_0[M] + \sum_{n=1}^{N-1} \lambda_n[M] (j\omega)^n$ is a polynomial in $(j\omega)$ of degree no greater than $(N-1)$. To derive the scaling

relation, we take $\varphi_M(\frac{t}{2})$ and $\varphi_{2M}(t)$ to the Fourier domain. We have that

$$\frac{2\hat{\varphi}_M(2\omega)}{\hat{\varphi}_{2M}(\omega)} = \frac{2\Lambda_M(j2\omega)\hat{\beta}_{\frac{\alpha}{M}}(2\omega)e^{-j2\omega a}}{\Lambda_{2M}(j\omega)\hat{\beta}_{\frac{\alpha}{2M}}(\omega)e^{-j\omega a}}. \quad (3.25)$$

By identifying the Fourier symbol $H_{\frac{\alpha}{2M}}(e^{j\omega})$, we can rewrite (3.25) as

$$\frac{2\hat{\varphi}_M(2\omega)}{\hat{\varphi}_{2M}(\omega)} = \frac{\Lambda_M(j2\omega)}{\Lambda_{2M}(j\omega)} H_{\frac{\alpha}{2M}}(e^{j\omega}) e^{-j\omega a}. \quad (3.26)$$

Using this result, it is straightforward to verify that $V_{\alpha,M} \subset V_{\alpha,2M}$, provided that a in (3.21) is an integer and $\frac{\Lambda_M(j2\omega)}{\Lambda_{2M}(j\omega)}$ is a 2π -periodic function. If a is noninteger, a similar multiresolution embedding space scheme can be achieved by shifting the grid at each refinement level. In any case, the particular choice of the set of parameters $\{\lambda_n\}_{n=0\dots N-1}$ will determine if the basis function is refinable and, therefore, if the multiresolution spaces are nested or not. We analyze in Section 3.4.1.3 how (3.26) applies to the various bases proposed in this chapter. In addition, we illustrate in Section 3.4.1.4 a constructive procedure to determine a nontrivial set of $\{\lambda_n\}_{n=0\dots N-1}$ that satisfies (3.26) and generates refinable schemes.

In the case where the spaces are nested, the subdivision process for finding the sequence of coefficients \mathbf{c}_{2M} starting with the sequence \mathbf{c}_M is then carried out in the following two steps:

1. up-sampling of the original sequence \mathbf{c}_M with a factor of 2;
2. filtering of the up-sampled sequence with a smoothing filter $\tilde{h}_{\frac{\alpha}{2}}$ using periodic boundary conditions.

The filter \tilde{h}_{α} will depend on the particular choice of the parameters $\{\lambda_n\}_{n=0\dots N-1}$, and its construction will be exemplified in Section 3.4 for the case of centered basis functions. The sequence \mathbf{c}_{2M} of $2M$ coefficients represents exactly the same parametric curve as the original sequence \mathbf{c}_M of M coefficients. This process can be repeated indefinitely to obtain finer representations of the curve in a dyadic fashion.

3.3.3 Multiresolution-Reproduction Capabilities

An alternative multiresolution scheme emerges as we concatenate new elements to α for fixed M . Since the reproduction of exponential polynomials is fully determined by α , the incorporation of additional elements does not perturb the reproduction capabilities. This multiresolution scheme in the reproduction properties will be exemplified in the case of multiple harmonics in Section 3.4.2.

3.4 Applications

In this section, we make use of Theorem 2 to build basis functions with minimal support capable of reproducing sinusoids. We start with single-frequency sinusoids that lead to ellipses, and then we derive the basis functions for generating higher-order harmonics.

3.4.1 Reproduction of Ellipses

Circles and ellipses deserve a special attention since these simple shapes appear frequently in images in many fields, for example computer graphics and biomedical engineering. Since all ellipses can be obtained by applying an affine transformation to the unit circle, we focus on the reproduction of this simple shape. This allows us to take advantage of the requirement for affine invariance that we stated in Section 3.1.3.

A parametric curve defined by M vectorial coefficients and by an M -dependent generating function φ_M is said to reproduce the unit circle if there exist two M -periodic sequences $\{c_c[k]\}_{k \in \mathbb{Z}}$ and $\{c_s[k]\}_{k \in \mathbb{Z}}$ such that

$$\cos(2\pi t) = \sum_{k=0}^{M-1} c_c[k] \varphi_{M,\text{per}}(Mt - k) \quad (3.27)$$

$$\sin(2\pi t) = \sum_{k=0}^{M-1} c_s[k] \varphi_{M,\text{per}}(Mt - k). \quad (3.28)$$

We illustrate in Figure 3.2 the reproduction of sinusoids of unit period for each component. Note that, when (3.27) and (3.28) hold, it is possible to represent any sinusoid of unit period for an arbitrary initial phase using linear combinations of the two sequences of coefficients.

3.4.1.1 Minimal-Support Basis for Sinusoids with Maximum Smoothness We now particularize Theorem 2 for the case of sinusoids keeping the maximum degree of smoothness for φ_M . This particular case is of special interest to us. We show in Chapter 4 and Chapter 5 how to build active contours capable of reproducing ellipses and ellipsoids respectively.

Corollary 1. *The centered generating function with minimal support and maximal smoothness that satisfies all conditions in Section 3.1.3 and that reproduces sinusoids of unit period with M coefficients is*

$$\varphi_M^S(t) = \sum_{k=0}^3 (-1)^k c_M^S[k] \varsigma_M(t + \frac{3}{2} - k), \quad (3.29)$$

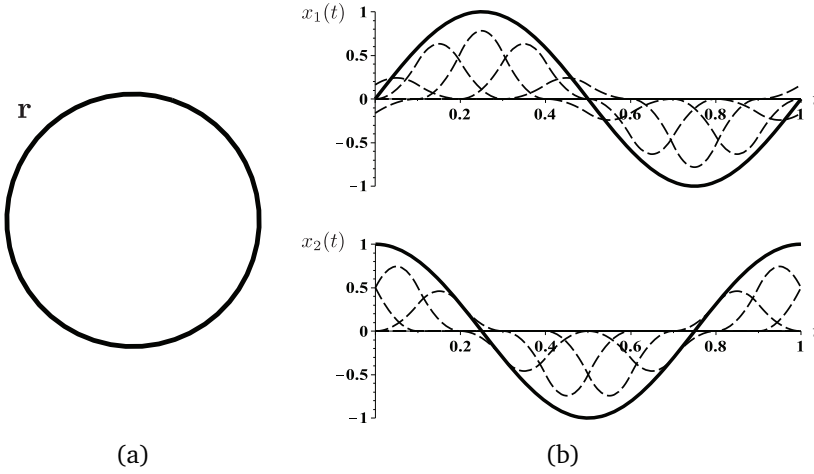


Figure 3.2: Parametric representation of the unit circle (a) and its coordinate functions (b) with exponential B-splines and $M = 10$. The dashed lines in (b) indicate the corresponding basis functions.

where

$$\begin{cases} \varsigma_M(t) &= \frac{1}{4} \operatorname{sgn}(t) \left(\frac{\sin(\frac{\pi}{M} t)}{\sin \frac{\pi}{M}} \right)^2 \\ c_M^S &= [1, 1 + 2 \cos \frac{2\pi}{M}, 1 + 2 \cos \frac{2\pi}{M}, 1]. \end{cases}$$

Proof. Using (3.21), we see that φ_M^S needs to be constructed from combinations of exponential B-splines with parameters $\alpha = (0, j \frac{2\pi}{M}, -j \frac{2\pi}{M})$, which leaves $N = 3$. Therefore, we have

$$\varphi_M^S(t) = \sum_{n=0}^2 \lambda_n^S[M] \frac{d^n}{dt^n} \beta_\alpha(t - a). \quad (3.30)$$

This ensures that φ_M^S is the shortest generating function that reproduces constants and all sinusoids of unit period with M coefficients. The reproduction of constants is a direct consequence of using $\alpha_1 = 0$, and the sinusoid-reproduction property comes from applying Euler's identity to $\alpha_2 = j \frac{2\pi}{M}$ and $\alpha_3 = -j \frac{2\pi}{M}$.

In order to maximize the smoothness of the resulting generating function, the coefficients $\lambda_1^S[M]$ and $\lambda_2^S[M]$ in (3.30) must vanish. Since φ_M^S reproduces constants, $\lambda_0^S[M]$ can be determined by imposing the partition-of-unity condition.

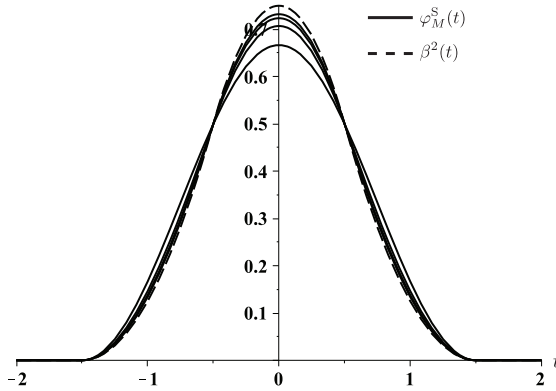


Figure 3.3: Plot of a quadratic B-spline β^2 and of the generating functions in (3.29) for $M = 3, 4, 5,$ and 6 . The function with the lowest peak at $t = 0$ corresponds to $M = 3$, and, as M increases, the central peak increases as well.

From (3.7), we have that

$$\lambda_0^S[M] = \left(\operatorname{sinc} \frac{1}{M} \right)^{-2}.$$

An exponential B-spline parameterized by α generates a Riesz basis if and only if

$$\alpha_{m_1} - \alpha_{m_2} \notin 2\pi j\mathbb{Z}$$

for all purely imaginary pairs such that $m_1 \neq m_2$. In our case, it is important to realize that this condition is satisfied if and only if $M \geq M_0 = 3$. In other words, at least three control points are needed to define our parametric curve.

Finally, a closed form for φ_M^S is obtained by computing the inverse Fourier transform of

$$\hat{\varphi}_M^S(\omega) = \lambda_0^S[M] e^{j\frac{3\omega}{2}} \frac{1 - e^{-j\omega}}{j\omega} \frac{1 - e^{-(j\omega - j\frac{2\pi}{M})}}{j\omega - j\frac{2\pi}{M}} \frac{1 - e^{-(j\omega + j\frac{2\pi}{M})}}{j\omega + j\frac{2\pi}{M}},$$

where we have set $a = -\frac{3}{2}$ in order to ensure that the basis function is centered. \square

We show in Figure 3.3 some members of this family of functions for several values of M . We observe that they are continuous, with finite support of length $W = 3$, and tend to be bump-like. Moreover, when $M \rightarrow \infty$, they converge to

the quadratic B-spline. We can see this by expanding in Maclaurin series ς_M . Then, we have that $\lim_{M \rightarrow \infty} \varsigma_M(t) = \frac{1}{4} \operatorname{sgn}(t) t^2$ and $\lim_{M \rightarrow \infty} c_M^S = [1, 3, 3, 1]$ immediately implies that $\lim_{M \rightarrow \infty} \varphi_M^S = \beta^2$. This is because a polynomial B-spline of degree n can be written as

$$\beta^n(t) = \sum_{k=0}^{n+1} (-1)^k \binom{n+1}{k} \varsigma^n\left(t + \frac{n+1}{2} - k\right),$$

where $\varsigma^n(t) = \frac{1}{2n!} \operatorname{sgn}(t) t^n$. Note that the convergence of φ_M^S to β^2 is point-wise. A piecewise expression of φ_M^S can be obtained by expanding (3.29) into

$$\varphi_M^S(t) = \frac{1}{1 - \cos \frac{2\pi}{M}} \begin{cases} \cos \frac{2\pi|t|}{M} \cos \frac{\pi}{M} - \cos \frac{2\pi}{M} & 0 \leq |t| < \frac{1}{2} \\ \left(\sin \frac{\pi(3/2-|t|)}{M} \right)^2 & \frac{1}{2} \leq |t| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |t|. \end{cases}$$

3.4.1.2 Minimal-Support Interpolating Basis for Sinusoids As was suggested in Section 3.2.6, the generating function φ_M can be tailored to satisfy the interpolating condition. We investigate now how this applies to the reproduction of ellipses and other trigonometry-related curves.

Corollary 2. *The centered interpolating generating function with minimal support that satisfies all conditions in Section 3.1.3 and that reproduces sinusoids of unit period with M coefficients is*

$$\begin{aligned} \varphi_M^I(t) &= \sum_{k=0}^3 (-1)^k c_M^S[k] \sec \frac{\pi}{M} \left(\varsigma_M\left(t + \frac{3}{2} - k\right) \right. \\ &\quad \left. - \frac{1}{16} \left(\sec \frac{\pi}{2M} \right)^2 \operatorname{sgn}\left(t + \frac{3}{2} - k\right) \right). \end{aligned} \quad (3.31)$$

Proof. Following the same approach as when constructing φ_M^S , we see that φ_M^I needs to be constructed from combinations of exponential B-splines with $\mathbf{a} = (0, j \frac{2\pi}{M}, -j \frac{2\pi}{M})$. Therefore, we have that

$$\varphi_M^I(t) = \sum_{n=0}^2 \lambda_n^I[M] \frac{d^n}{dt^n} \beta_{\mathbf{a}}(t - a). \quad (3.32)$$

In order to fulfill the interpolating condition, $\lambda_0^I[M]$, $\lambda_1^I[M]$, and $\lambda_2^I[M]$ must satisfy a linear system of equations. If we set $a = -\frac{3}{2}$ in order to ensure that the

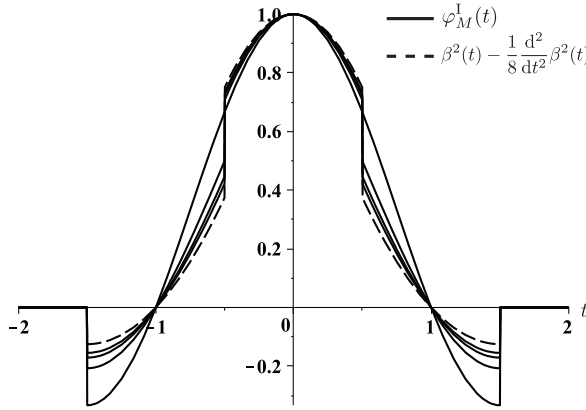


Figure 3.4: Plot of the third-order I-MOMS $(\beta^2 - \frac{1}{8} \ddot{\beta})$ and of the generating functions in (3.32) for $M = 3, 4, 5,$ and 6 . Among the different φ_M^I , the function with the least pronounced discontinuity at $t = \pm \frac{3}{2}$ corresponds to $M = 3$, and, as M increases, the jump of the discontinuity increases as well. For $M = 3$, φ_M^I is continuous at $t = \pm \frac{1}{2}$, but discontinuous at $t = \pm \frac{3}{2}$.

basis function is centered, we end up with

$$\begin{aligned} \lambda_0^I[M] &= 1 \\ \lambda_1^I[M] &= 0 \\ \lambda_2^I[M] &= \left(\frac{M}{2\pi}\right)^2 \left(1 - \sec \frac{\pi}{M}\right). \end{aligned}$$

In this case, the interpolating φ_M^I is a Riesz basis if and only if $M \geq 3$, a condition that we already encountered in the case of Corollary 1. Finally, a closed form for φ_M^I is obtained by applying an inverse Fourier transform to

$$\begin{aligned} \hat{\varphi}_M^I(\omega) &= \lambda_0^I[M] e^{j\frac{3\omega}{2}} \frac{1 - e^{-j\omega}}{j\omega} \frac{1 - e^{-(j\omega - j\frac{2\pi}{M})}}{j\omega - j\frac{2\pi}{M}} \frac{1 - e^{-(j\omega + j\frac{2\pi}{M})}}{j\omega + j\frac{2\pi}{M}} \\ &+ \lambda_2^I[M] (j\omega)^2 e^{j\frac{3\omega}{2}} \frac{1 - e^{-j\omega}}{j\omega} \frac{1 - e^{-(j\omega - j\frac{2\pi}{M})}}{j\omega - j\frac{2\pi}{M}} \frac{1 - e^{-(j\omega + j\frac{2\pi}{M})}}{j\omega + j\frac{2\pi}{M}}. \end{aligned}$$

□

We show in Figure 3.4 some members of this family of functions for several values of M . We observe that they share a finite support of length $W = 3$. As

we increase M , φ_M^I converges to $\beta^2 - \frac{1}{8}\beta^2$, which is the third-order I-MOMS described in [129]. A piecewise expression of φ_M^I can be obtained by expanding (3.31) into

$$\varphi_M^I(t) = \begin{cases} \frac{\cos \frac{2\pi t}{M} - \cos \frac{2\pi}{M}}{1 - \cos \frac{2\pi}{M}} & 0 \leq |t| < \frac{1}{2} \\ \frac{(2 \cos \frac{\pi}{M} + 1)^2}{8 \cos \frac{\pi}{M} (\cos \frac{\pi}{M} + 1)} & |t| = \frac{1}{2} \\ \frac{\cos \frac{\pi}{M} - \cos \frac{2\pi(3/2-t)}{M}}{2(1 - \cos \frac{2\pi}{M}) \cos \frac{\pi}{M}} & \frac{1}{2} \leq |t| < \frac{3}{2} \\ \frac{\cos \frac{\pi}{M} - \cos \frac{2\pi}{M}}{-1} & |t| = \frac{3}{2} \\ 0 & |t| > \frac{3}{2}. \end{cases}$$

3.4.1.3 Refinability of the Proposed Bases As discussed in Section 3.3.2, not all members of the family of functions given by Theorem 2 are refinable. Here, we show the multiresolution properties of the proposed basis functions that reproduce sinusoids.

When imposing maximal smoothness, it is straightforward to verify that the basis function φ_M^S is refinable since it is proportional to a refinable exponential B-spline. To build the associated refinement mask, we have to take into account that a is a half-integer. Therefore, there is a half-integer shift in the parameterization every time we apply the refinement. This means that a curve \mathbf{r}_M built with M coefficients and the same curve expressed with $2M$ coefficients satisfy

$$\begin{aligned} \mathbf{r}_M(t) &= \lambda_0^S[M] \sum_{k=-\infty}^{\infty} \mathbf{c}_M[k] \beta_a(Mt - k + \frac{3}{2}) \\ &= \lambda_0^S[2M] \sum_{k=-\infty}^{\infty} \mathbf{c}_{2M}[k] \beta_{\frac{a}{2}}(2Mt - k + \frac{3}{2} - \frac{1}{2}). \end{aligned}$$

The dependency between the two sequences of coefficients can be stated as

$$\begin{aligned} \mathbf{c}_{2M}[k] &= \frac{\lambda_0^S[M]}{\lambda_0^S[2M]} \sum_{l=-\infty}^{\infty} \mathbf{c}_M[l] h_{\frac{a}{2}}[k + 2 - 2l] \\ &= \left(\frac{\text{sinc} \frac{1}{2M}}{\text{sinc} \frac{1}{M}} \right)^2 \left((\mathbf{c}_M)_{\uparrow 2} * h_{\frac{a}{2}} \right)[k + 2], \end{aligned}$$

where $(\mathbf{c}_M)_{\uparrow 2}$ is the \mathbf{c}_M sequence upsampled by a factor of 2. It is interesting to note that the filter $h_{\frac{a}{2}}$ is equal to the sequence c_M^S in the expression of φ_M^S in (3.29). We identify the refinement filter $\tilde{h}_{\frac{a}{2}}$ described in Section 3.3.2 as a shifted and scaled version of the refinement filter $h_{\frac{a}{2}}$ of the exponential B-spline β_a .

When imposing the interpolation property, it can be shown that, for the particular choice $\lambda_0^I[M]$, $\lambda_1^I[M]$, and $\lambda_2^I[M]$ leads to a ratio $\frac{\Lambda_M(j2\omega)}{\Lambda_{2M}(j\omega)}$ that is not 2π -periodic. Thus, the multiresolution spaces are not nested, and no refinement mask exists. Meanwhile, φ_M^I is unique due to the restrictions introduced by the interpolatory condition, and there is no remaining degree of freedom to be used to increase the regularity or to improve the multiresolution properties of the basis function.

3.4.1.4 Additional Refinable Bases In this section, we illustrate a constructive procedure to design new refinable schemes. In particular, we focus on the particular case where the ratio $\frac{\Lambda_M(j2\omega)}{\Lambda_{2M}(j\omega)}$ is constant. This can be achieved by imposing scaling conditions over the N' roots $\{\gamma_n\}_{n=1\dots N'}$ of the polynomial $\Lambda_M(j\omega)$. Then, we have that

$$\Lambda(j\omega) = \lambda_{N'}[M] \prod_{n=1}^{N'} (j\omega - \gamma_n[M]),$$

where $N' < N$ and where we have made explicit the dependence of the roots with respect to M . Note that there is a one-to-one dependence between the elements of the set $\{\lambda_n\}_{n=0\dots N-1}$ and the roots of the polynomial $\{\gamma_n\}_{n=1\dots N'}$, up to a scaling factor. In particular, if we choose the roots such that

$$\gamma_n[2M] = \frac{\gamma_n[M]}{2} \tag{3.33}$$

for all n , then the quantity

$$\begin{aligned} \frac{\Lambda_M(j2\omega)}{\Lambda_{2M}(j\omega)} &= \frac{\lambda_{N'}[M] \prod_{n=1}^{N'} (j2\omega - \gamma_n[M])}{\lambda_{N'}[2M] \prod_{n=1}^{N'} (j\omega - \gamma_n[2M])} && \text{by definition} \\ &= \frac{\lambda_{N'}[M] 2^{N'} \prod_{n=1}^{N'} (j\omega - \gamma_n[M]/2)}{\lambda_{N'}[2M] \prod_{n=1}^{N'} (j\omega - \gamma_n[2M])} && \text{factoring} \\ &= 2^{N'} \frac{\lambda_{N'}[M]}{\lambda_{N'}[2M]} && \text{by (3.33)} \end{aligned}$$

is independent of ω and the resulting function φ is refinable. This particular multiresolution scheme where the roots of $\Lambda_M(j\omega)$ satisfy (3.33) is intimately related to the generalized exponential B-splines proposed in [130].

To build new refinable basis functions that reproduce sinusoids, we can choose the roots $\{\gamma_n^R\}_{n=1\dots N'}$ of $\Lambda_M(j\omega)$ such that $\gamma_n^R[2M] = \gamma_n^R[M]/2$. The number of roots N' determines which is the maximum non-zero element in the sequence $\{\lambda_n^R\}_{n=0\dots 2}$, and therefore the smoothness of the resulting basis function.

The particular choice of $\gamma_1^R[M] = -\gamma_2^R[M] = \frac{1}{M}$ and $a = -\frac{3}{2}$ defines a refinable, centered, and symmetric generating function with minimal support reproduces

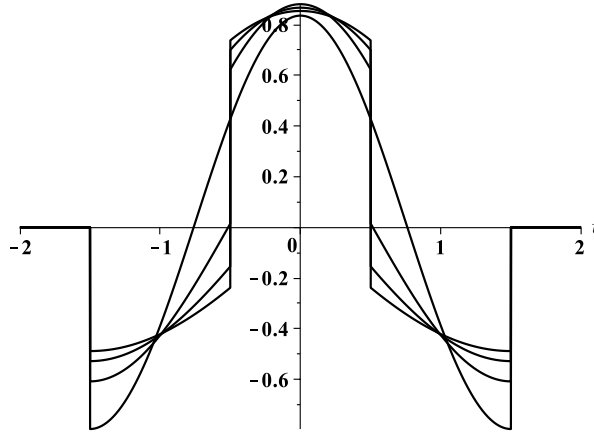


Figure 3.5: Plot of the generating functions in (3.34) for $M = 3, 4, 5,$ and 6 . Among the different φ_M^R , the function with the most pronounced discontinuity at $t = \pm\frac{3}{2}$ corresponds to $M = 3$, and, as M increases, the jump of the discontinuity decreases.

sinusoids of unit period with M coefficients. These roots determine the set of parameters $\{\lambda_n^R\}_{n=0\dots 2}$ up to a scaling constant as

$$\begin{aligned}\lambda_0^R[M] &= -\frac{\lambda_2^R[M]}{M^2} \\ \lambda_1^R[M] &= 0 \\ \lambda_2^R[M] &= \lambda_2^R[M].\end{aligned}$$

Then, the resulting generating function is

$$\varphi_M^R(t) = -\frac{\lambda_2^R[M]}{M^2} \beta_\alpha(Mt - k + \frac{3}{2}) + \lambda_2^R[M] \ddot{\beta}_\alpha(Mt - k + \frac{3}{2}). \quad (3.34)$$

We show in Figure 3.5 some members of this family of functions for several values of M . We choose $\lambda_2^R[M]$ such that the L_2 norm of $\varphi_M^R(t)$ is unitary. We observe that they share a finite support of length $W = 3$.

Our choice of $\{\gamma_n^R\}_{n=1\dots N'}$ is arbitrary and corresponds to one particular case where the resulting generating function is symmetric and non-smooth. Other choices would lead to asymmetric functions and other degrees of smoothness.

3.4.1.5 Order of Approximation The notion of order of approximation is crucial in approximation theory since it governs the rate of decrease of the approximation error as the sampling step vanishes. Specifically, in the periodic stationary case,

the approximation order is defined as the exponent L such that the difference between a function f and its projection $\mathcal{P}_M f$ onto $\{\varphi(M \cdot -k)\}_{k \in \mathbb{Z}}$, or equivalently in $\{\varphi_{\text{per}}(M \cdot -k)\}_{k=[0 \dots M-1]}$, tends to zero. In direct analogy with the classical Strang-and-Fix theory of approximation for the nonperiodic case, it has been shown in [119] that the error for the periodic case can be bounded by

$$\|f - \mathcal{P}_M f\|_{L_2([0,1])} \leq C_\varphi M^{-L} \|f\|_{L_2([0,1])},$$

where C_φ is a constant that only depends on the particular choice of φ . An analogous result for the nonstationary case can be obtained using the concept of asymptotically equivalent subdivision schemes presented in [100]. We say that φ_M and $\tilde{\varphi}$ define equivalent multiresolution schemes of order γ if and only if

$$\|\tilde{\mathcal{P}}_M f - \mathcal{P}_M f\|_{L_2([0,1])} = \mathcal{O}(M^{-\gamma}), \quad (3.35)$$

where $\mathcal{P}_M f$ denotes the projection of f onto $\{\varphi_M(N \cdot -k)\}_{k \in \mathbb{Z}}$ with $N = M$, and $\tilde{\mathcal{P}}_M f$ denotes the projection of f onto $\{\tilde{\varphi}(M \cdot -k)\}_{k \in \mathbb{Z}}$. In our setting, if we set $\tilde{\varphi}(t) = \lim_{M \rightarrow \infty} \varphi_M(t)$ for all $t \in \mathbb{R}$, it is straightforward to see that

$$\|f - \mathcal{P}_M f\|_{L_2([0,1])} \leq \|f - \tilde{\mathcal{P}}_M f\|_{L_2([0,1])} + \|\tilde{\mathcal{P}}_M f - \mathcal{P}_M f\|_{L_2([0,1])} = \mathcal{O}(M^{-\min(L, \gamma)}).$$

Therefore, if the φ_M and $\tilde{\varphi}$ define multiresolution schemes of order high enough, the rate of decay of the error is the same for the nonstationary and the stationary case.

By taking the limit $M \rightarrow \infty$ on φ_M^S and φ_M^I , we can observe that such functions converge to the classical quadratic B-spline β^2 and to the third-order I-MOMS $\beta^2 - \frac{1}{8}\beta^2$ derived in [129], respectively. Both generating functions are known to have the same order of approximation $L = 3$. The main difference between them lies in the constant that multiplies the M^{-3} factor. This factor is more favorable in the case of the quadratic B-spline than in the case of the third-order I-MOMS. Thus, in general, the approximation offered by the quadratic B-spline is more accurate than the one offered by the I-MOMS. This property carries over to φ_M^S and φ_M^I when $M \rightarrow \infty$.

3.4.2 Reproduction of Higher-Order Harmonics

We now present a constructive procedure to extend the ellipse-reproduction properties of our curves to higher-order harmonics. This problem was already approached using Fourier descriptors [56]. Since our basis functions are capable of perfectly reproducing sinusoids, the classical family of Fourier descriptors becomes a special class of our construction. It must be noted, though, that our bases have a finite support, a property which is lacking in Fourier descriptors.

We say that a parametric curve defined by M vectorial coefficients and by a generating function φ_M reproduces higher-order harmonics up to order L if there

	φ_M^S	φ_M^I
Parameters	$\lambda_0^S[M] = \left(\text{sinc}\frac{1}{M}\right)^{-2}$ $\lambda_1^S[M] = 0$ $\lambda_2^S[M] = 0$	$\lambda_0^I[M] = 1$ $\lambda_1^I[M] = 0$ $\lambda_2^I[M] = \left(\frac{M}{2\pi}\right)^2 \left(1 - \sec\frac{\pi}{M}\right)$
Smoothness	$\mathcal{C}^{-1}(\mathbb{R})$	$\mathcal{C}^1(\mathbb{R})$
Order of approximation	$\mathcal{O}(M^{-3})$	$\mathcal{O}(M^{-3})$
Limit ($M \rightarrow \infty$)	β^2	$\beta^2 - \frac{1}{8}\beta^2$
Refinable	YES	NO

 Table 3.1: Summary of the properties of φ_M^S and φ_M^I .

exist two M -periodic sequences $\{c_{l,c}[k]\}_{k \in \mathbb{Z}}$ and $\{c_{l,s}[k]\}_{k \in \mathbb{Z}}$ for every $1 \leq l \leq L$ such that

$$\cos(2\pi l t) = \sum_{k=0}^{M-1} c_{l,c}[k] \varphi_{M,\text{per}}(M t - k) \quad (3.36)$$

$$\sin(2\pi l t) = \sum_{k=0}^{M-1} c_{l,s}[k] \varphi_{M,\text{per}}(M t - k). \quad (3.37)$$

Such a curve is able to reproduce all modes up to order L for each component. Like in the case of the sinusoids, it is possible to represent any initial phase using linear combinations of the two sequences of coefficients in (3.36) and (3.37). We recall that, using Euler's identity and the multinomial theorem, related functions such as $(\cos(2\pi \cdot))^l$ and $(\sin(2\pi \cdot))^l$, with $1 \leq l \leq L$, can also be expressed as linear combinations of elements from $\{\cos(2\pi l \cdot), \sin(2\pi l \cdot)\}_{1 \leq l \leq L}$. This ensures that the functions $(\cos(2\pi \cdot))^l$ and $(\sin(2\pi \cdot))^l$ are expressible with the same basis functions φ_M or $\varphi_{M,\text{per}}$.

3.4.2.1 Minimal-Support Basis for Higher-Order Harmonics

Corollary 3. *The centered generating function with minimal support and maximal smoothness that satisfies all conditions in Section 3.1.3 and that reproduces higher-order harmonics up to order L with M coefficients is*

$$\varphi_M^S(t) = \lambda_0[M] \beta_\alpha \left(t + \frac{2L+1}{2}\right), \quad (3.38)$$

where α contains only $\{0\}$, $\{j \frac{2\pi}{M} k\}_{k \in [1 \dots L]}$, and $\{-j \frac{2\pi}{M} k\}_{k \in [1 \dots L]}$, and where the value $\lambda_0[M]$ is an appropriate normalizing constant.

Proof. The proof follows the same strategy as in Corollary 1. The choice of the collection α and the size of the support $N = 2L + 1$ is given by Theorem 2.

The parameters $\lambda_1[M], \dots, \lambda_L[M]$ are set to zero to maximize the smoothness of φ_M^S , and $\lambda_0[M]$ is fixed in such a way that φ_M^S satisfies the partition-of-unity condition, which yields

$$\lambda_0[M] = \frac{1}{\sum_{k=1}^{2L-1} \beta_\alpha(k + \frac{1}{2})}.$$

We recall that exponential B-splines parameterized by α form a Riesz basis if and only if $(\alpha_{m_1} - \alpha_{m_2}) \notin 2\pi j\mathbb{Z}$ for all pairs such that $m_1 \neq m_2$. In our case, this condition is satisfied if $M \geq 2L + 1$. Finally, the shift parameter is set to $a = -\frac{2L+1}{2}$ to ensure that the generating function is centered. \square

It should be noted that the smoothest basis function corresponds to a normalized trigonometric spline, which was defined as a piecewise trigonometric function by Schoenberg in [131].

3.4.2.2 Parametric Expansion of Higher-Order Harmonics Here, we determine the sequence of M vector coefficients that reproduce the higher-order harmonics using the generating function φ_M^S given in (3.38). We start by recalling the exponential-reproducing property of the exponential B-splines

$$e^{\alpha t} = \sum_{k=-\infty}^{\infty} e^{\alpha k} \beta_{(\alpha)}(t - k). \quad (3.39)$$

Setting $\alpha = j\frac{2\pi l}{M}$ with $1 \leq l \leq L$, we see that $\beta_{(j\frac{2\pi l}{M})}$ reproduces $e^{j\frac{2\pi l}{M}t}$. If we now convolve both sides of (3.39) with $\beta_{\alpha \setminus (j\frac{2\pi l}{M})}$, we get that

$$\left(\beta_{\alpha \setminus (j\frac{2\pi l}{M})} * e^{j\frac{2\pi l}{M}\cdot} \right) (t) = \sum_{k=-\infty}^{\infty} e^{j\frac{2\pi l}{M}k} \underbrace{\left(\beta_{(j\frac{2\pi l}{M})} * \beta_{\alpha \setminus (j\frac{2\pi l}{M})} \right) (t - k)}_{\frac{1}{\lambda_0[M]} \varphi_M^S(t - \frac{2L+1}{2} - k)},$$

where we have used the definition of φ_M^S from (3.38), along with the fact that the convolution operator commutes with the shift operator. To simplify the left-hand side, we invoke an important property of linear shift-invariant (LSI) systems: complex exponentials are eigenfunctions of LSI operators. By virtue of this property, if the complex exponential $e^{j\alpha t}$ is presented at the input of a system specified by the impulse response h , then its output is given by $\hat{h}(\alpha)e^{j\alpha t}$, where \hat{h} denotes the Fourier transform of h . If we consider $\beta_{\alpha \setminus (j\frac{2\pi l}{M})}$ as the impulse response of a LSI system, then

$$\left(\beta_{\alpha \setminus (j\frac{2\pi l}{M})} * e^{j\frac{2\pi l}{M}\cdot} \right) (t) = \hat{\beta}_{\alpha \setminus (j\frac{2\pi l}{M})}(\omega) \Big|_{\omega = \frac{2\pi l}{M}} e^{j\frac{2\pi l}{M}t}.$$

Therefore, we have that

$$e^{j\frac{2\pi l}{M}t} = \sum_{k=-\infty}^{\infty} e^{j\frac{2\pi l}{M}k} \frac{1}{\lambda_0[M] \hat{\beta}_{\alpha \setminus (j\frac{2\pi l}{M})}(\omega) \Big|_{\omega=\frac{2\pi l}{M}}} \varphi_M^S(t - \frac{2L+1}{2} - k).$$

By flipping the sign of α , we can easily obtain an analogous result for the reproduction of $e^{-j\frac{2\pi l}{M}t}$. Finally, by using both results, we have that

$$\cos\left(2\pi l\left(t + \frac{2L+1}{2M}\right)\right) = \sum_{k=-\infty}^{\infty} c_1[k] \varphi_M^S(Mt - k) \quad (3.40)$$

$$\sin\left(2\pi l\left(t + \frac{2L+1}{2M}\right)\right) = \sum_{k=-\infty}^{\infty} c_2[k] \varphi_M^S(Mt - k), \quad (3.41)$$

where

$$c_1[k] = \frac{1}{2\lambda_0[M]} \left(\frac{e^{j\frac{2\pi l}{M}k}}{\hat{\beta}_{\alpha \setminus (j\frac{2\pi l}{M})}(\omega) \Big|_{\omega=\frac{2\pi l}{M}}} + \frac{e^{-j\frac{2\pi l}{M}k}}{\hat{\beta}_{\alpha \setminus (-j\frac{2\pi l}{M})}(\omega) \Big|_{\omega=-\frac{2\pi l}{M}}} \right)$$

$$c_2[k] = \frac{1}{2j\lambda_0[M]} \left(\frac{e^{j\frac{2\pi l}{M}k}}{\hat{\beta}_{\alpha \setminus (j\frac{2\pi l}{M})}(\omega) \Big|_{\omega=\frac{2\pi l}{M}}} - \frac{e^{-j\frac{2\pi l}{M}k}}{\hat{\beta}_{\alpha \setminus (-j\frac{2\pi l}{M})}(\omega) \Big|_{\omega=-\frac{2\pi l}{M}}} \right).$$

Note that the sequences c_1 and c_2 can be considered M -periodic and that the summations in (3.40) and (3.41) can be reduced to finite ones if we make use of the periodized basis functions given in (3.4). We have expressed in (3.40) and (3.41) how to compute the vector coefficients for reproducing sinusoids and initial phase. The appropriate linear combination of c_1 and c_2 allows one to change arbitrarily the initial phase.

In order to illustrate the reproduction capabilities of the proposed model, we designed a basis function capable of reproducing some of the classical harmonic curves [132]. In particular, we tailored φ_M^S in (3.38) with $L = 4$ and $M = 9$, which lead to $\alpha = (0, j\frac{2\pi}{9}, -j\frac{2\pi}{9}, \dots, j\frac{8\pi}{9}, -j\frac{8\pi}{9})$. We show some members of the Lissajous, Hypotrochoid, and Epitrochoid families in Figures 3.6, 3.7, and 3.8, respectively. More singular examples like the Teardrop, the Deltoid, the Astroid, and the Cardioid are shown in Figure 3.9. The coefficients for each coordinate function can be found in Table 3.2.

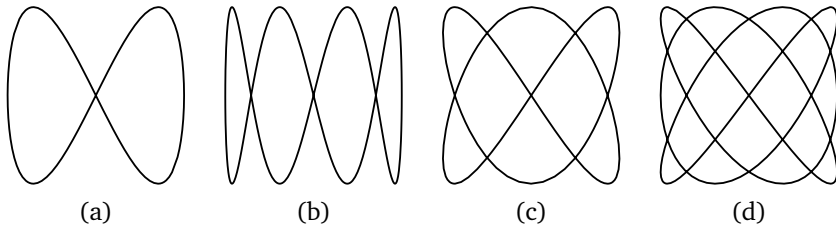


Figure 3.6: Lissajous curves.

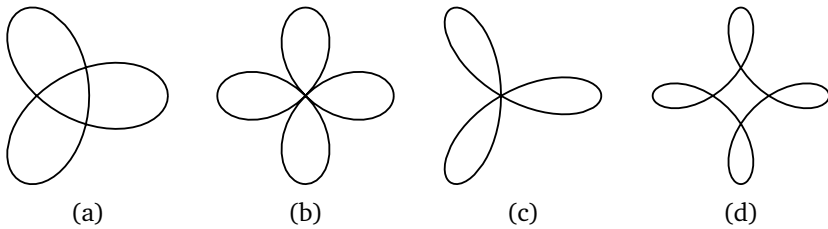


Figure 3.7: Hypotrochoid curves.

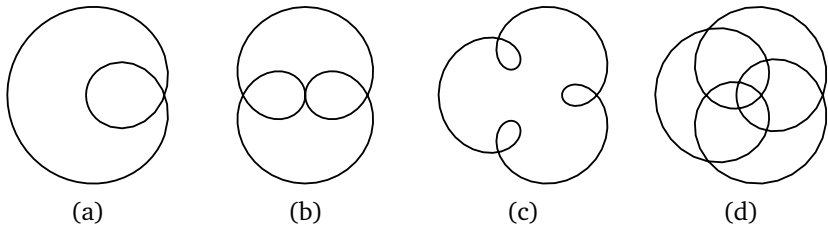


Figure 3.8: Epitrochoid curves.

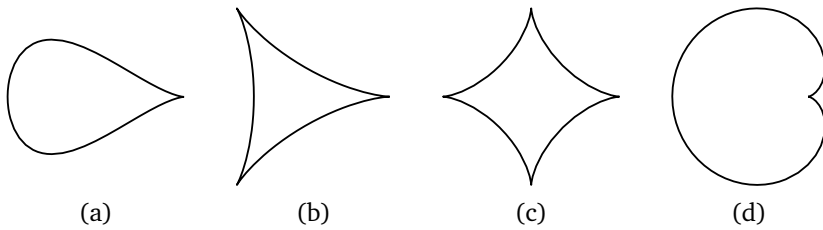


Figure 3.9: Other curves: (a) Teardrop, (b) Deltoid, (c) Astroid, (d) Cardioid.

Curve	c_{x_1}	c_{x_2}
Lissajous (a)	$c_{1,s}$	$c_{2,s}$
Lissajous (b)	$c_{1,s}$	$c_{4,s}$
Lissajous (c)	$c_{2,s}$	$c_{3,s}$
Lissajous (d)	$c_{3,s}$	$c_{4,s}$
Hypotrochoid (a)	$2c_{1,c} + 3c_{2,c}$	$2c_{1,s} - 3c_{2,s}$
Hypotrochoid (b)	$c_{1,c} + c_{3,c}$	$c_{1,s} - c_{3,s}$
Hypotrochoid (c)	$c_{1,c} + c_{2,c}$	$c_{1,s} - c_{2,s}$
Hypotrochoid (d)	$3c_{1,c} + 2c_{3,c}$	$3c_{1,s} - 2c_{3,s}$
Epitrochoid (a)	$2c_{1,c} - 3c_{2,c}$	$2c_{1,s} - 3c_{2,s}$
Epitrochoid (b)	$c_{1,c} - c_{3,c}$	$c_{1,s} - c_{3,s}$
Epitrochoid (c)	$2c_{1,c} - c_{4,c}$	$2c_{1,s} - c_{4,s}$
Epitrochoid (d)	$4c_{1,c} - 5c_{4,c}$	$4c_{1,s} - 5c_{4,s}$
Teardrop	$4c_{1,c}$	$2c_{1,s} - c_{2,s}$
Deltoid	$2c_{1,c} + c_{2,c}$	$2c_{1,s} - c_{2,s}$
Astroid	$3c_{1,c} + c_{3,c}$	$3c_{1,s} - c_{3,s}$
Cardioid	$2c_{1,c} - c_{2,c}$	$2c_{1,s} - c_{2,s}$

Table 3.2: Coefficients for the curves shown in Figure 3.6, Figure 3.7, Figure 3.8, and Figure 3.9.

3.5 Conclusions

In this chapter we have proposed a new family of basis functions that we use to represent planar curves. We were able to single out the basis of shortest support that allows one to reproduce exponential polynomials. Under the appropriate circumstances, these basis functions may form a natural multiscale hierarchy. In these cases, we specified multiresolution algorithms and subdivision schemes for the representation of geometric closed curves. We were able to characterize the order of approximation of such nonstationary multiresolution schemes. We exemplified our method by constructing minimal-support bases that reproduce ellipses and higher-order harmonics. In particular we tailored these bases to obtain maximal-smoothness basis functions, and interpolatory basis functions. In the forthcoming chapters, we take advantage of the theoretical developments of this chapter to build efficient active contours in 2D and 3D.

Chapter 4

2D Spline Snakes

In this chapter, we present a new class of continuously defined parametric snakes using the basis functions we designed in Chapter 3. While the resulting snakes are versatile enough to provide a good approximation of any closed curve in the plane, their most important feature is the fact that they admit ellipses within their span. Thus, they can perfectly generate circular and elliptical shapes. These features are appropriate to delineate cross sections of cylindrical-like conduits and to outline blob-like objects. We illustrate in Figure 4.1 how our snake can adopt the shape of a perfect ellipse (*i.e.*, reproduces the ellipse) as well as more refined shapes.

Segmenting circles and ellipses in images is a problem that arises in many fields, for example biomedical engineering [133, 134, 135, 136] or computer graphics [137, 138]. In medical imaging in particular, it is usually necessary to segment arteries and veins within tomographic slices [139]. Because those objects are physiological tubes, their sections show up as ellipses in the image. Ellipse-like objects are also present at microscopic scales. For instance, cell nuclei are known to be nearly circular [140] and water drops are similarly spherical thanks to surface-tension forces [141]. However, these elements deform and become elliptical when they are subject to stress forces.

In order to efficiently segment elliptical objects, a parametric snake named the Ovuscule was proposed in [69]. It is a minimalistic elliptical snake defined by three control points. Its main drawback was that it was unable to represent shapes different from circles and ellipses. Our goal here is to create a more versatile parametric snake whose basis functions are short, perfectly reproduce ellipses, and have good approximation properties. Our main contribution in this chapter is to fulfill this goal by selecting a special kind of exponential B-splines. We are actually able to prove that our basis functions are the ones with the shortest support among all admissible functions. Since the computational cost



Figure 4.1: Approximation capabilities of the proposed parametric snake. The thin solid line corresponds to an elliptical fit. The dashed thick line corresponds to a generalized shape.

of spline snakes is determined in part by the size of the support of the basis function, our use of the shortest possible support favors optimal performance.

The chapter is organized as follows: In Section 4.1 we review the general parametric snake model, fix the notation, and formalize our design constraints. The main contribution is described in Section 4.2, where we build an explicit expression for the underlying basis functions that fulfill our requirements, and we analyze in detail its reproduction and approximation properties. Implementation details such as energy functionals and discretization issues are addressed in Section 4.3. Finally, we perform report evaluations in Section 4.4.

4.1 Parametric Snakes

In this section we begin by recalling the formalism of B-spline parametric curves, and fix the notation for the rest of the chapter, which is a simplification of the one of Chapter 3.

4.1.1 Parametric Representation of Closed Curves

Following our formalism introduced in Section 3.1, a curve $\mathbf{r}(t)$ on the plane can be described by a pair of Cartesian coordinate functions $x_1(t)$ and $x_2(t)$, where $t \in \mathbb{R}$ is a continuous parameter. The one-dimensional functions x_1 and x_2

are efficiently parameterized by linear combinations of suitable basis functions. Among all possible bases, we focus on those derived from a compactly supported generator φ and its integer shifts $\{\varphi(\cdot - k)\}_{k \in \mathbb{Z}}$. This allows us to take advantage of the availability of fast and stable interpolation algorithms [116].

We are interested in closed curves specified by an M -periodic sequence of control points $\{\mathbf{c}[k]\}_{k \in \mathbb{Z}}$, with $\mathbf{c}[k] = \mathbf{c}[k + M]$. The parametric representation of the curve is then given by the vectorial equation

$$\mathbf{r}(t) = \sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(M t - k). \quad (4.1)$$

The number of control points M determines the degrees of freedom in the model (4.1). Small numbers lead to constrained shapes, and large numbers lead to additional flexibility and more general shapes.

Since the curve \mathbf{r} is closed, each coordinate function is periodic, and the period is common for both. For simplicity, in (4.1) we normalized this period to be unity. Under these conditions, we can reduce the infinite summation in (4.1) to a finite one involving periodized basis functions as

$$\begin{aligned} \mathbf{r}(t) &= \sum_{k=0}^{M-1} \sum_{n=-\infty}^{\infty} \mathbf{c}[M n + k] \varphi(M (t - n) - k) \\ &= \sum_{k=0}^{M-1} \mathbf{c}[k] \underbrace{\sum_{n=-\infty}^{\infty} \varphi(M (t - n) - k)}_{\varphi_M(M t - k)}, \end{aligned} \quad (4.2)$$

where φ_M is the M -periodization of the basis function φ .

This kind of curve parameterization is general. Using this model, we can approximate any closed curve as accurately as desired by using a higher number of vector coefficients $M_2 > M$, provided that φ satisfies some mild conditions [119].

4.1.2 Desirable Properties for the Basis Functions

We now enumerate the conditions that our parametric snake model should satisfy and introduce the corresponding mathematical formalism.

1. *Unique and Stable Representation.* We want our parametric curve to be defined in terms of the coefficients in such a way that unicity of representation of the coordinate functions x_1 and x_2 is satisfied. Furthermore, for computational purposes, we ask the interpolation procedure to be numerically stable.

A generating function φ is said to satisfy the Riesz basis condition if and only if there exist two constants $0 < A \leq B < \infty$ such that

$$A \|\mathbf{c}\|_{\ell_2} \leq \sqrt{M} \left\| \sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(M \cdot -k) \right\|_{L_2} \leq B \|\mathbf{c}\|_{\ell_2} \quad (4.3)$$

for all $\mathbf{c} \in \ell_2$. A direct consequence of the lower inequality is that the condition $\sum_{k=-\infty}^{\infty} \mathbf{c}[k] \varphi(M t - k) = 0$ for all $t \in \mathbb{R}$ implies that $\mathbf{c}[k] = 0$ for all $k \in \mathbb{Z}$. Thus, the basis functions are linearly independent and every function is uniquely specified by its coefficients. The upper inequality ensures the stability of the interpolation process [116].

It has been shown in [117] that, due to the integer-shift-invariant structure of the representation, the Riesz condition has the following equivalent expression in the Fourier domain:

$$A \leq \sum_{k=-\infty}^{\infty} |\hat{\varphi}(\cdot + 2\pi k)|^2 \leq B,$$

where $\hat{\varphi}(\omega) = \int_{\mathbb{R}} \varphi(x) e^{-j\omega x} dx$ denotes the Fourier transform of φ . Once expressed in the Fourier domain, the Riesz condition provides a practical way to verify if a given generating function φ satisfies (4.3).

2. *Affine Invariance.* Since we are interested in outlining shapes irrespective of their position and orientation, we would like our model to be invariant to affine transformations, which we formalize as

$$\mathbf{A}\mathbf{r}(t) + \mathbf{b} = \sum_{k=-\infty}^{\infty} (\mathbf{A}\mathbf{c}[k] + \mathbf{b}) \varphi(M t - k), \quad (4.4)$$

where \mathbf{A} is a (2×2) matrix and \mathbf{b} is a two-dimensional vector. From (4.4), it is easy to show that affine invariance is ensured if and only if

$$\forall t \in \mathbb{R} : \sum_{k=-\infty}^{\infty} \varphi(M t - k) = 1. \quad (4.5)$$

In the literature, this constraint is often named the *partition-of-unity* condition [116].

3. *Well-Defined Curvature.* The curvature of a parametric curve at a point $(x_1(t), x_2(t))$ is given by

$$\kappa(x_1, x_2) = \frac{\dot{x}_1 \ddot{x}_2 - \ddot{x}_1 \dot{x}_2}{(\dot{x}_1^2 + \dot{x}_2^2)^{3/2}},$$

where the dot denotes the derivative with respect to t . We would like κ to be a bounded function with respect to t . To do so, each coordinate function (or, equivalently, the basis φ) must be at least $\mathcal{C}^1(\mathbb{R})$ with bounded second derivative.

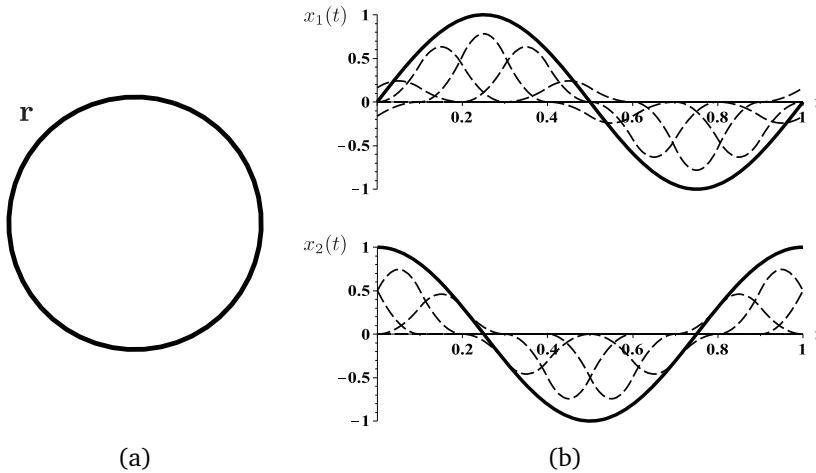


Figure 4.2: Parametric representation of the unit circle (a) and its coordinate functions (b) with exponential B-splines and $M = 10$. The dashed lines in (b) indicate the corresponding basis functions.

4.2 Reproduction of Ellipses

Since every ellipse can be obtained by applying an affine transformation to the unit circle, we focus on the reproduction of this simpler shape. This simplification is allowed whenever the affine-invariance requirement stated in Section 4.1.2 is satisfied.

A parametric snake defined by M vectorial coefficients and by a generating function φ is said to reproduce the unit circle if there exist two M -periodic sequences $\{c_c[k]\}_{k \in \mathbb{Z}}$ and $\{c_s[k]\}_{k \in \mathbb{Z}}$ such that

$$\cos(2\pi t) = \sum_{k=-\infty}^{\infty} c_c[k] \varphi(Mt - k) \quad (4.6)$$

$$\sin(2\pi t) = \sum_{k=-\infty}^{\infty} c_s[k] \varphi(Mt - k). \quad (4.7)$$

That is, we need to be able to reproduce sinusoids of unit period for each component of the parametric snake, as illustrated in Figure 4.2. Note that, when (4.6) and (4.7) hold, it is possible to represent any sinusoid of unit period for an arbitrary initial phase using linear combinations of the two sequences of coefficients.

4.2.1 Minimum-Support Ellipse-Reproducing Basis

We now provide an explicit expression for the minimum-support basis functions that reproduce sinusoids. These bases are a particular case of the broader family of basis functions investigated in Chapter 3.

By Corollary 1 in Section 3.4.1.1, we know that the centered generating function with minimal support that satisfies the Riesz basis condition, the partition-of-unity condition, is $\mathcal{C}^1(\mathbb{R})$ with bounded second derivative and reproduces sinusoids of unit period with M coefficients is

$$\varphi(t) = \frac{1}{1 - \cos \frac{2\pi}{M}} \begin{cases} \cos \frac{2\pi|t|}{M} \cos \frac{\pi}{M} - \cos \frac{2\pi}{M} & 0 \leq |t| < \frac{1}{2} \\ \left(\sin \frac{\pi(3/2-|t|)}{M} \right)^2 & \frac{1}{2} \leq |t| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |t|. \end{cases} \quad (4.8)$$

This result is a direct consequence of the Minimal-Support Generating Functions Theorem detailed in Section 3.2.5. This theorem provides a complete characterization of the family of basis functions with minimum-support that reproduce exponential polynomials expressed as combinations of exponential B-splines.

We recall that the basis function (4.8) form a Riesz basis if and only if $M \geq 3$. Therefore, at least three control points are needed to define our parametric snake. Moreover, they are one-time continuously differentiable and the second derivative is bounded. This ensures the well-definiteness of the curvature of the snake curve.

For the sake of completeness, we also show in Figure 4.3 the function φ for several values of M . We observe that they share with the quadratic B-spline a finite support of length $W = 3$, and all of them have a similar bump-like appearance.

4.2.2 Approximation Properties

Not only are we interested in reproducing ellipses, but we would also like our snake to be able to approximate any other shape \mathbf{s} . This is achieved by increasing the number of degrees of freedom afforded by the number M of nodes. In the Fourier domain, it is easy to see that φ converges to a quadratic B-spline as M increases (see Section 3.4.1.1). Therefore, we expect similar approximation properties for large values of M .

While φ leads to integer-shift invariance, the space spanned by the generating function φ is not shift-invariant in general. Hence, the approximation error using M vector coefficients is dependent upon a shift in the continuous parameter t of the 1-periodic function \mathbf{s} . The minimum-mean-square approximation error

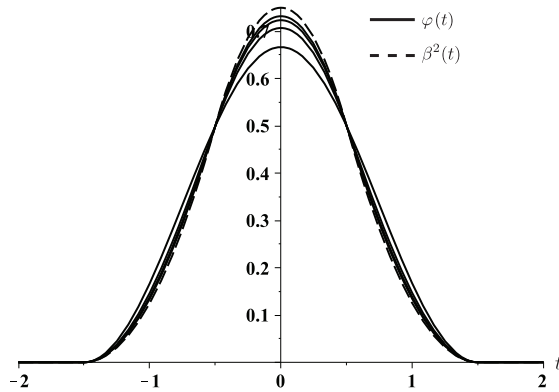


Figure 4.3: Plot of a quadratic B-spline β^2 and the resulting generating functions given in (4.8) for $M = 3, 4, 5$, and 6 . The function with the lowest peak at $t = 0$ corresponds to $M = 3$, and as M increases, the height of the central peak increases as well.

for a shifted function is given by

$$\begin{aligned} \gamma(\tau, M) &= \int_0^1 \|\mathbf{s}(t - \tau) - \mathbf{r}(t)\|^2 dt \\ &= \|\mathbf{s}(\cdot - \tau) - \mathbf{r}(\cdot)\|_{L_2([0,1])}^2, \end{aligned}$$

where \mathbf{r} is the best approximation within the span $\{\varphi(M \cdot -k)\}_{k \in \mathbb{Z}}$. Since τ is usually unknown, we measure the error averaged over all possible shifts as

$$\eta(M) = \left(\int_0^1 \gamma(\tau, M) d\tau \right)^{\frac{1}{2}}. \quad (4.9)$$

We give in Section 4.2.3 the decay of η as $M \rightarrow \infty$, following the method described in [119].

4.2.3 Approximation Order

In this section, we introduce the necessary formalism to compute the order of the approximation error associated to the best-possible approximation of a periodic vector function \mathbf{s} within the span of the basis $\{\varphi(M \cdot -k)\}_{k \in \mathbb{Z}}$, where φ is given by (4.8).

As explained in Section 4.2.2 about the approximation properties of φ , the space spanned by the generating function φ is not shift-invariant in general. Hence, as

a metric of dissimilarity between shapes, we use the averaged minimum-mean-square approximation error η .

Using the main result of [119], we obtain the asymptotic behavior of η as

$$\begin{aligned} \eta^2(M) &= C_1^2(M) \|\mathbf{s}\|_{L_2([0,1])}^2 M^{-2} \\ &\quad + C_2^2(M) \|\mathbf{s}\|_{L_2([0,1])}^2 M^{-4} + \mathcal{O}(M^{-6}), \end{aligned}$$

where $C_L = \frac{1}{L!} \sqrt{\left(\sum_{k \neq 0} |\hat{\varphi}^{(L)}(2\pi k)|^2\right)}$ and $\hat{\varphi}^{(L)}$ is the L -th derivative of the Fourier transform of φ . Following lengthy calculations, we get

$$C_1(M) = \frac{1}{12\pi} \left(18 (M_0 - M) (M_0 + 4M) + 30\pi^2\right)^{\frac{1}{2}} \quad (4.10)$$

$$\begin{aligned} C_2(M) &= \frac{1}{120\pi^2} \left(225 (2M_0^4 - 7M^2 M_0^2 - 15M^3 M_0 + 20M^4) \right. \\ &\quad \left. + 75 (8M_0^2 - 29M^2) \pi^2 + 170\pi^4\right)^{\frac{1}{2}}, \end{aligned} \quad (4.11)$$

where we defined $M_0 = \pi \cot \frac{\pi}{M}$. It can be shown that $C_1(M) = \mathcal{O}(M^{-2})$ and $C_2(M) = \mathcal{O}(M^{-2})$. Since the curve \mathbf{s} does not depend on M , we can also write that

$$\eta(M) = \left(\mathcal{O}(M^{-6})\right)^{\frac{1}{2}} = \mathcal{O}(M^{-3}),$$

which shows that the averaged quadratic mean error decays as M^{-3} —the same rate as the quadratic B-spline [142].

4.2.4 Best Constant and Ellipse Fitting

Since our snakes have the capability of perfectly reproducing ellipses, it is natural to ask which is the best ellipse that approximates the parametric curve \mathbf{r} defined by the M -periodic sequence $\{\mathbf{c}[k]\}_{k \in \mathbb{Z}}$. In other words, we are interested in finding the ellipse \mathbf{r}_e that minimizes

$$\|\mathbf{r} - \mathbf{r}_e\|_{L_2([0,1])}^2 = \int_0^1 \|\mathbf{r}(t) - \mathbf{r}_e(t)\|^2 dt.$$

Since \mathbf{r} is continuous and 1-periodic, we can expand it in a Fourier series as

$$\mathbf{r}(t) = \sum_{n=-\infty}^{\infty} \mathbf{R}[n] e^{j2\pi n t}. \quad (4.12)$$

The Fourier-series vector coefficients \mathbf{R} in (4.12) are given by

$$\begin{aligned}\mathbf{R}[n] &= \int_0^1 \mathbf{r}(t) e^{-j2\pi n t} dt \\ &= \frac{1}{M} \hat{\varphi}\left(\frac{2\pi n}{M}\right) \sum_{k=0}^{M-1} \mathbf{c}[k] e^{-j\frac{2\pi}{M}nk},\end{aligned}\quad (4.13)$$

where the parametric expression of \mathbf{r} has been used in the second equality.

From the classical theory of harmonic analysis, we know that the best ellipse approximation (component-wise sinusoids) of \mathbf{r} , in the $L_2([0, 1])$ sense, is the first-order truncation of the series (4.12), where only the terms $n = -1$, $n = 0$, and $n = 1$ are kept. Therefore, we have that

$$\begin{aligned}\mathbf{r}_e(t) &= \mathbf{R}[0] + (\mathbf{R}[1] + \mathbf{R}[-1]) \cos(2\pi t) \\ &\quad + j(\mathbf{R}[1] - \mathbf{R}[-1]) \sin(2\pi t),\end{aligned}\quad (4.14)$$

where $\mathbf{R}[0]$ is the center of gravity of the snake. The Fourier-series vector coefficients in (4.14) can easily be obtained from (4.13) as

$$\begin{aligned}\mathbf{R}[0] &= \frac{1}{M} \sum_{k=0}^{M-1} \mathbf{c}[k] \\ \mathbf{R}[1] + \mathbf{R}[-1] &= \sum_{k=0}^{M-1} h_c[k] \mathbf{c}[k] \\ j(\mathbf{R}[1] - \mathbf{R}[-1]) &= \sum_{k=0}^{M-1} h_s[k] \mathbf{c}[k],\end{aligned}$$

where

$$\begin{aligned}h_c[k] &= \frac{2}{M} \cos \frac{\pi}{M} \cos \frac{2\pi k}{M} \\ h_s[k] &= \frac{2}{M} \cos \frac{\pi}{M} \sin \frac{2\pi k}{M}.\end{aligned}$$

Since all sinusoids of unit period can be reproduced by the generating function φ and the appropriate M -periodic sequence of coefficients \mathbf{c} , the curve \mathbf{r}_e belongs to the span of φ . For the sake of completeness, we provide in the next section an explicit expansion of sinusoids in terms of φ .

4.2.5 Expansion of Sinusoids

Here, we explicitly find the sequence of M vector coefficients that reproduce sinusoids of unit period using the generating function φ given in (4.8). We start

by recalling the exponential-reproducing property of the exponential B-splines as

$$e^{\alpha t} = \sum_{k=-\infty}^{\infty} e^{\alpha k} \beta_{(\alpha)}(t - k). \quad (4.15)$$

Setting $\alpha = j \frac{2\pi}{M}$, we see that $\beta_{(j \frac{2\pi}{M})}$ reproduces the complex exponential $e^{j \frac{2\pi}{M} t}$, which is M -periodic. If we now convolve both sides of (4.15) with $\beta_{(0, -j \frac{2\pi}{M})}$, we get that

$$\left(\beta_{(0, -j \frac{2\pi}{M})} * e^{j \frac{2\pi}{M} \cdot} \right) (t) = \sum_{k=-\infty}^{\infty} e^{j \frac{2\pi}{M} k} \underbrace{\left(\beta_{(j \frac{2\pi}{M})} * \beta_{(0, -j \frac{2\pi}{M})} \right) (t - k)}_{\frac{2(1 - \cos \frac{2\pi}{M})}{\left(\frac{2\pi}{M}\right)^2} \varphi(t - \frac{3}{2} - k)}$$

where we have used the definition of φ from (4.8), along with the fact that the convolution operator commutes with the shift operator.

To simplify the left-hand side, we invoke an important property of linear shift-invariant (LSI) systems: complex exponentials are eigenfunctions of LSI operators. By virtue of this property, if the complex exponential $e^{j\alpha t}$ is presented at the input of a system specified by the impulse response h , then its output is given by $\hat{h}(\alpha) e^{j\alpha t}$, where \hat{h} denotes the Fourier transform of h . If we consider $\beta_{(0, -j \frac{2\pi}{M})}$ as the impulse response of a LSI system, then

$$\left(\beta_{(0, -j \frac{2\pi}{M})} * e^{j \frac{2\pi}{M} \cdot} \right) (t) = \underbrace{\hat{\beta}_{(0, -j \frac{2\pi}{M})}(\omega)}_{\lambda} \Big|_{\omega = \frac{2\pi}{M}} e^{j \frac{2\pi}{M} t}.$$

Therefore, we have that

$$e^{j \frac{2\pi}{M} t} = \sum_{k=-\infty}^{\infty} e^{j \frac{2\pi}{M} k} 2 \frac{1 - \cos \frac{2\pi}{M}}{\lambda \left(\frac{2\pi}{M}\right)^2} \varphi(t - \frac{3}{2} - k).$$

By flipping the sign of α we can easily obtain an analogous result for the reproduction of $e^{-j \frac{2\pi}{M} t}$. Finally, by using both results, we have that

$$\cos(2\pi t) = \sum_{k=-\infty}^{\infty} c_c[k] \varphi(M t - k) \quad (4.16)$$

$$\sin(2\pi t) = \sum_{k=-\infty}^{\infty} c_s[k] \varphi(M t - k), \quad (4.17)$$

where

$$c_c[k] = \frac{2 \left(1 - \cos \frac{2\pi}{M}\right)}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \cos \frac{2\pi k}{M}$$

$$c_s[k] = \frac{2 \left(1 - \cos \frac{2\pi}{M}\right)}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \sin \frac{2\pi k}{M}.$$

Note that the sequences c_c and c_s are M -periodic and that the summations in (4.16) and (4.17) can be reduced to finite ones if we make use of the periodized basis functions.

We have expressed in (4.16) and (4.17) how to compute the vector coefficients for reproducing sinusoids of unit period. The appropriate linear combination of c_c and c_s allows one to reproduce sinusoids of arbitrary initial phase.

4.3 Energies and Implementation

Since the presented parametric active contour is a spline snake, it is capable of handling all traditional energies applicable to point-snakes and parametric snakes. However, to illustrate the behavior of our parameterization in a real implementation, we performed our experiments with a specific snake energy that we designed to be versatile.

In this section, we first introduce the snake energy that drives the optimization process, and then we provide a description of the implementation details for the proposed snake. We construct the energy functional to detect dark objects on a brighter background.

4.3.1 Snake Energy

As it was exposed in Chapter 2, the snake evolution is driven by a chosen energy function. Thus, the quality of the segmentation depends on the choice of the energy term. In our model we obviated the constraint energy since we accommodated the user interaction as a hard constraint allowing the user to leave some control points outside the optimization routine.

4.3.1.1 Image Energy There are many construction strategies for the image energy. These can be categorized in two main families: 1) edge-based schemes, which use gradient information to detect contours [43, 47, 50] and 2) region-based methods, which use statistical information to distinguish different homogeneous regions [49, 68]. In order to benefit from the advantages of both strategies, a unified energy was proposed in [57]. In our case, we are going to follow a similar approach by using a convex combination of gradient and region energies, like in

$$E_{\text{image}} = \alpha E_{\text{edge}} + (1 - \alpha) E_{\text{region}} \quad (4.18)$$

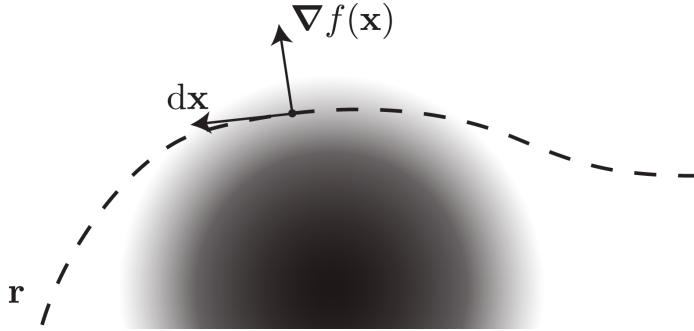


Figure 4.4: Schematic representation of a parametric snake \mathbf{r} (dashed line), of its interaction with an object constituted by a gray semicircle (representing low pixel values), of the vector $d\mathbf{x}$ tangent to the curve, and of the gradient vector ∇f of the image. The vector \mathbf{k} , which is mentioned in the text, is perpendicular to the image plane and points outwards, towards the reader.

where $\alpha \in [0, 1]$. The tradeoff parameter α balances the contribution of the edge-based energy and the region-based energy. Its value depends on the characteristics of each particular application.

For the gradient-based (or edge) energy, we consider the one described in [68] since it has the advantage of penalizing the snake when the orientation is inconsistent with the object to segment. Let \mathbf{r} be our parametric snake. The contour energy term is then given by

$$E_{\text{edge}} = - \oint_{\mathbf{r}} \mathbf{k}^T (\nabla f(x_1, x_2) \times d\mathbf{x}), \quad (4.19)$$

where $\mathbf{k} = (0, 0, 1)$ denotes the outward vector orthonormal to the image plane, where $\nabla f(x_1, x_2) = \left(\frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2}, 0 \right)$ is the within-plane gradient of the image f at (x_1, x_2) on the curve, where $d\mathbf{x}$ denotes the tangent vector of the curve in the three-dimensional space formed by the image plane and its orthogonal dimension, and where \times is the 3D cross product. In Figure 4.4, we present the configuration of the various quantities involved. The chirality of the system of coordinates will determine the sign of the integrand, as discussed in [57, 68].

For the region-based energy, we adopt a strategy similar to the Ovuscule in [69]. More precisely, our region-based energy discriminates an object from its background by building an ellipse \mathbf{r}_λ around the snake and maximizing the contrast between the intensity of the data averaged within the curve, and the intensity of the data averaged over the elliptical shell Ω_λ . When $\Omega \subset \Omega_\lambda$, the region energy

term can be expressed as

$$E_{\text{region}} = \frac{1}{|\Omega|} \left(\iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_\lambda \setminus \Omega} f(\mathbf{x}) dx_1 dx_2 \right), \quad (4.20)$$

where $|\Omega|$ is given by

$$|\Omega| = - \sum_{k=0}^{M-1} \sum_{n=0}^{M-1} c_1[k] c_2[n] \int_0^M \varphi_M(t-n) \dot{\varphi}_M(t-k) dt. \quad (4.21)$$

The normalization factor $|\Omega|$ can be interpreted as the signed area, defined as $|\Omega| = - \oint_{\mathbf{r}} x_2 dx_1$. The sign of the quantity $|\Omega|$ depends on the clockwise or anti-clockwise path followed on the curve \mathbf{r} . In this paper, we follow the usual convention whereby an anti-clockwise path leads to a positive sign. We enforce our criterion to remain neutral ($E_{\text{region}} = 0$) when f takes a constant value, for instance in flat regions of the image. To achieve this we set $|\Omega_\lambda| = 2 |\Omega|$.

The construction of the elliptic shell is performed using the best ellipse \mathbf{r}_e given in (4.14), and magnifying its axes by a factor λ to achieve

$$\begin{aligned} \mathbf{r}_\lambda(t) = & \mathbf{R}[0] + \lambda (\mathbf{R}[1] + \mathbf{R}[-1]) \cos(2\pi t) \\ & + j\lambda (\mathbf{R}[1] - \mathbf{R}[-1]) \sin(2\pi t), \end{aligned}$$

where $\lambda = \sqrt{2|\Omega|/|\Omega_e|}$ and $|\Omega_e|$ is the signed area enclosed by the curve \mathbf{r}_e , with

$$|\Omega_e| = - \frac{4\pi}{M^2} \cos \frac{\pi}{M} \sum_{k=0}^{M-1} \sum_{n=0}^{M-1} c_1[k] c_2[n] \sin \frac{2\pi(n-k)}{M}.$$

The elliptic shell \mathbf{r}_λ is fully determined by the sequence of control points. Thus, the optimization of the control points leads to an automatic readjustment of \mathbf{r} and \mathbf{r}_λ .

In Figure 4.5, we illustrate how we take advantage of the ideas presented in Section 4.2.4 to build the best ellipse approximation \mathbf{r}_e of an arbitrary snake \mathbf{r} .

4.3.1.2 Internal Energy The internal energy is responsible for ensuring the smoothness of the curve. In the original implementation by Kass *et al.* [43], the internal energy is composed of a linear combination of the length of the contour and the integral of the square of the curvature along the contour. This energy is the one that is most widely-used in applications.

In the framework of parametric snakes, most schemes rely on the smoothness of the representation, thus eliminating the need for an explicit internal energy term. However, these approaches can ensure a low value of the curvature only when the curves are parameterized at constant speed (proportional to arc-length). For example, a spline curve may be rough if some of the spline control

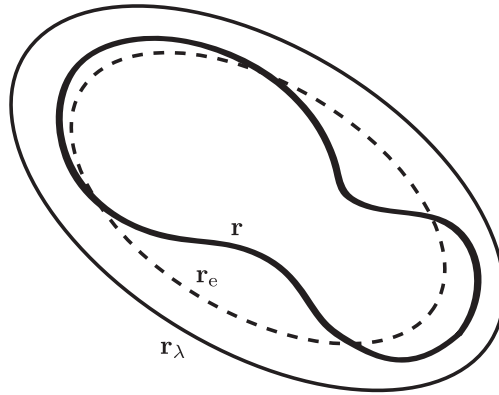


Figure 4.5: Representation of the parametric snake r , the best ellipse approximation r_e , and the corresponding enclosing shell r_λ used in E_{region} .

points accumulate at the same position. A practical workaround is to reparameterize the curve to constant arc-length after each step of the optimization algorithm, which is quite expensive [143]. Another approach is to substitute the curvature term of (5.21) by an energy term that penalizes the curve for not being in the curvilinear abscissa [57]. This energy is called *curvilinear reparameterization energy*. Minimizing this energy causes the control points to move tangentially to the snake, thus bringing it to curvilinear abscissa. The use of this energy yields the same results as reparameterizing the snake at each step, but with a much lower computational load.

In our implementation we obviated the internal energy term in order to allow our snake to segment objects with non-smooth boundaries. In Section 4.4.2 we quantify the accuracy of our snake while segmenting objects with non-smooth boundaries.

4.3.2 Fast Energy Computation

The computational cost is dominated by the evaluation of the surface integrals in (4.20). An efficient way to implement these operations is the use of pre-integrated images. Let g be the function we are integrating (Δf , f , or $-f$, respectively) and let Γ be the domain of integration (Ω or Ω_λ). Then, by Green's

theorem, we rewrite the surface integrals as the line integrals

$$\begin{aligned} \iint_{\Gamma} g(\mathbf{x}) dx_1 dx_2 &= \oint_{\partial\Gamma} g_1(x_1, x_2) dx_2 \\ &= - \oint_{\partial\Gamma} g_2(x_1, x_2) dx_1, \end{aligned}$$

where $\partial\Gamma$ is the boundary of Γ , and

$$g_1(x_1, x_2) = \int_{-\infty}^{x_1} g(\tau, x_2) d\tau \quad (4.22)$$

$$g_2(x_1, x_2) = \int_{-\infty}^{x_2} g(x_1, \tau) d\tau. \quad (4.23)$$

The use of Green's theorem to rewrite the surface integrals as line integrals reduces dramatically the computational load. This can only be achieved if the curve is defined continuously, like with the curves of Section 4.1.1. By contrast, this acceleration would not be available to methods such as point-snakes and level-sets, because their implementation ultimately relies on discretization.

In the interest of space, we show the derivation of the energies using pre-integrated images in Appendix 4.A.

4.3.3 Sampling

Despite the fact that we are assuming a continuously defined model for our functions, in a real-world implementation we only have at our disposal a sampled version of the functions we want to pre-integrate. To solve this inconsistency, we perform a bilinear interpolation of the sampled data and we store in lookup tables the values of (4.23) or (4.22) at integer locations.

4.3.4 Optimization

As mentioned before, the active contour extracts the final contour by finding the minimum of the energy functional. For that purpose, we iteratively update the value of the M free control points $\{\mathbf{c}[k]\}_{k \in [0..M-1]}$ using a generic unconstrained gradient-based optimizer. The optimization scheme is efficiently carried out by a Powell-like line-search method [144]. This method requires the derivatives of the energy function with respect to the parameters, and converges quadratically to the solution. The algorithm proceeds as follows: firstly, one direction within the parameter space is chosen depending on the partial derivatives of the energy. Secondly, a one-dimensional minimization is performed within the selected direction. Finally, a new direction is chosen using the partial derivatives of the energy function once more, while enforcing conjugation properties.

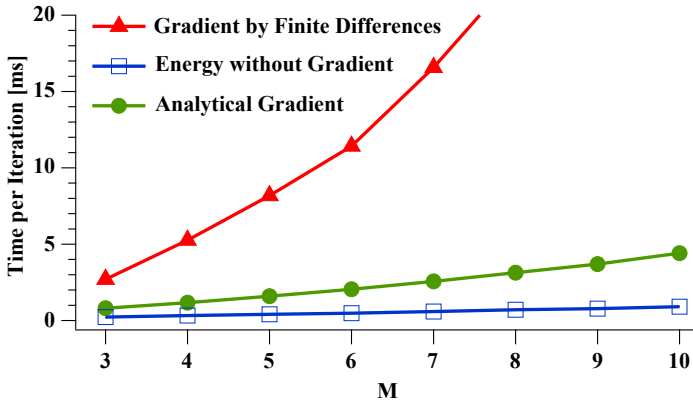


Figure 4.6: Mean time of one iteration in the snake evolution.

This scheme is repeated till convergence. Assuming a bilinear interpolation of the original function f , we were able to derive exact and closed expressions for these derivatives. In the interest of space, we show the derivation of these expressions in Appendix 4.A.

For spline snakes it has been shown that the evaluation of the partial derivatives of the energy of the form (4.18) depends quadratically on the number of parameters [57]. In Figure 4.6, we compare the computational cost of the snake during line minimization (simple update), and when the energy gradient is required to chose a new direction (gradient update). For the latter case, we contrast the computation time of an analytical computation of the gradient to that of a centered finite differences approach. For low values of M , the simple update and the gradient update using analytical energy gradient lead to a similar computational load. As the value of M increases, the quadratic behavior of the computation of the gradient makes the update cost increase. This quadratic behavior can be easily discerned in the topmost curve of Figure 4.6.

4.4 Experiments and Simulations

We present in this section four experimental setups. In the first one, we compare our choice in (4.8) against the classical quadratic B-spline when representing sinusoids. We move away from sinusoids in the second experiment, where we work with synthetic data and perform an objective validation of the segmentation properties of our snake in noiseless and noisy environments. In the third setup, we also perform a quantitative evaluation by segmenting real cardiac MRI data. Finally, in the last experiment, we illustrate some real applications of our

snake where the ground truth is not available.

4.4.1 Approximation of Sinusoids

By design, our basis function φ has the property of reproducing sinusoids exactly. By contrast, the classical polynomial B-splines do not enjoy this property. In this section, we are focusing on this aspect and exhibit the amount of error committed by B-splines when attempting to reproduce a sine function.

We start with exact reproduction by our basis. Using the result of Section 4.2.5, we determine the coefficients for the case $M = 3$ (smallest possible M). They are given by

$$\sin(2\pi t) = \sqrt{3} (\varphi_3(3t - 1) - \varphi_3(3t + 1)),$$

where φ_3 corresponds to the 3-periodization of the basis function (4.8), as in (4.2).

We continue with approximate reproduction by B-splines. For fairness, we choose a quadratic B-spline β^2 so that the size of the support of β^2 and φ is the same. The reproduction will be approximate, not because of the limited size of the support, but because the sine function does not lie in the span of polynomial B-splines of any degree. Nevertheless, we can compute the coefficients that best adjust the sinusoid with unit period in the least-squares sense. This yields

$$\sin(2\pi t) \approx \frac{1215}{26\pi^2} (\beta_3^2(3t - 1) - \beta_3^2(3t + 1)),$$

where

$$\beta^2(t) = \begin{cases} \frac{3}{4} - |t|^2 & 0 \leq |t| < \frac{1}{2} \\ \frac{1}{2} \left(\frac{3}{2} - |t| \right)^2 & \frac{1}{2} \leq |t| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |t| \end{cases} \quad (4.24)$$

is the quadratic B-spline and the subscript 3 indicates a 3-periodized basis function as in (4.2).

We observe in Figure 4.7 that both constructions result in sine-like functions. However, the reproduction is exact in the left part of Figure 4.7, while it is only approximate in the right part. This happens even though the support of β^2 is identical to the support of φ , even though the asymptotic approximation properties of β^2 and φ are identical, and even though β^2 and φ have the same degree of differentiability. We show in Figure 4.8 the amount of error committed by the parabolic approximation. We determine that $\text{MSE} = \frac{1}{2} - \frac{98415}{208\pi^6}$.

4.4.2 Accuracy and Robustness to Noise

In this section, two experiments are carried out. The first one consists in outlining different synthetic blob-like shapes in a noise-free environment. The second

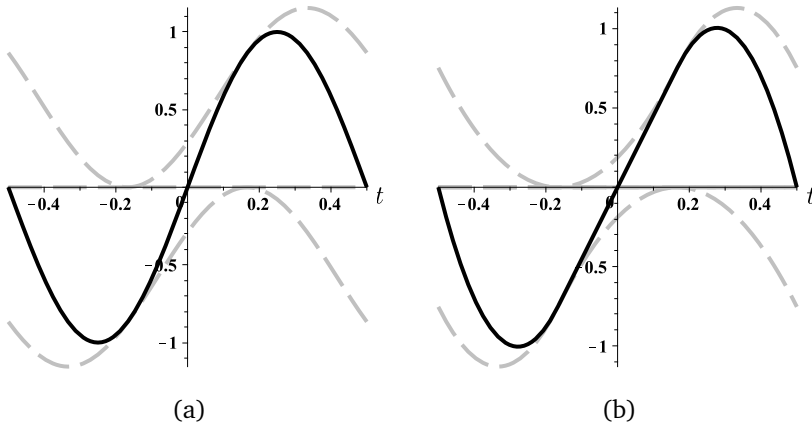


Figure 4.7: Approximations of a sin function with unit period. (a) Parametric representation (solid line) using φ_3 (dashed lines). (b) Best parametric approximation (solid line) using β^2 (dashed lines).

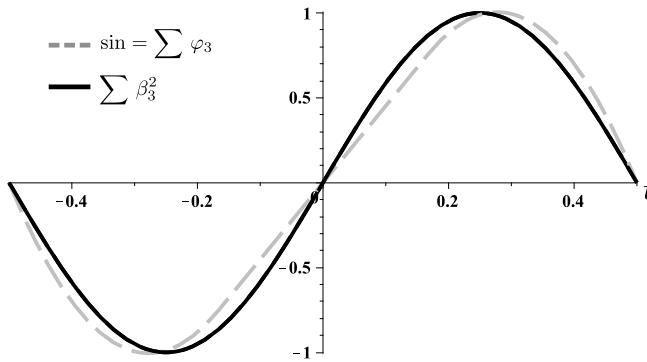


Figure 4.8: Sinusoid of period 3, its representation with our basis function (solid line), and its best quadratic B-spline approximation (dashed line).

experiment consists in outlining one specific target within an image, this time, in the presence of noise. In both experiments we set $\alpha = 0$, that is, we make use of the region energy only. This particular choice ensures that the snake is not misled by noisy boundaries in the presence of excessive noise.

In the first experiment, we generate 10 test images of size (512×512) by pixel-wise sampling of our shape of interest, which is built by intersecting or making the union of two circles of radius 50 pixel units. We illustrate these shapes in the header of Table 4.1. They are parameterized with the distance d , in pixel units, between the centers of the circles. For $d < 0$, the shape is built by the intersection of the two circles. For $d \geq 0$, they are parameterized by their union. The grayscale values of the images are 255 for the shape, and 0 for the background.

We used the Jaccard distance $J = 1 - |\Theta \cap \Omega| / |\Theta \cup \Omega|$ to measure as a percentage the dissimilarity between the two sets. There, Θ corresponds to the ground-truth region, and Ω corresponds to the region enclosed by the snake. We computed J with a pixel-wise discretization of the images.

In the simulations of Table 4.1, we investigated the dependence of J on the number M of coefficients and the distance d between the circles. We denoted with a dash (–) when the snake did not converge, and therefore, we could not compute the Jaccard distance. We initialized every snake as a circle with a radius of 75 pixel and a center that lay in the middle of the shape. We observe that the results in Table 4.1 tend to improve as the number M of control points is increased, especially for the non-elliptical shapes. However, the increase in the number of control points does not bring any further improvement when the shape to segment is a perfect circle. This result is expected since the circular shape is reproduced exactly for any $M \geq 3$. The residual error seen in Table 4.1 for $d = 0$ can be attributed to the discretization of Θ and Ω . We also observe that for $d = -80$ and $d = -64$ the Jaccard distance starts increasing severely for $M \geq 7$ and for $M \geq 9$, respectively. This is due to the fact that the sharp corners of the shape lead to loops in the curve during the optimization process. Such self-intersections violate the conditions of Green’s theorem in Section 4.3.1.

In the second experiment, we investigated the sensitivity to noise of our snake depending on the number of snake coefficients M . We generated 100 noisy realizations of a circle of radius 50 pixel units for different signal-to-noise ratios. We computed the power of the noise over a region of interest of size (200×200) . We illustrate a realization of the resulting images in the header of Table 4.2.

We show the percentage of success in Table 4.2. We considered that our snake succeeded in segmenting the circle when the optimization process led to a segmentation with $J < 1\%$. This criterion is very conservative as shown in Figure 4.9. We observe from the results that our snake is robust against noise since it is capable of giving a proper segmentation even for low signal-to-noise ratios.









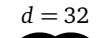
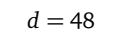
M	3	4	5	6	7	8	9	10
 $d = -80$	5.08	4.85	3.53	2.69	3.63	18.84	—	—
 $d = -64$	4.12	4.12	2.64	2.18	1.87	0.58	1.56	1.41
 $d = -48$	3.78	3.84	2.03	2.00	1.63	0.68	0.99	0.93
 $d = -32$	2.84	2.78	1.25	1.13	1.08	0.55	0.72	0.70
 $d = -16$	1.54	1.53	0.58	0.64	0.48	0.32	0.30	0.34
 $d = 0$	0.17	0.15	0.20	0.17	0.17	0.18	0.15	0.17
 $d = 16$	2.18	2.22	1.06	0.91	1.09	0.86	0.55	0.18
 $d = 32$	4.06	4.01	2.27	1.81	1.92	1.92	0.85	0.41
 $d = 48$	6.63	6.64	4.21	2.84	2.50	4.00	1.41	0.80
 $d = 64$	9.49	9.48	6.82	4.36	3.68	5.73	—	1.23

Table 4.1: Error percentage of our snake for noiseless synthetic data.

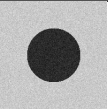
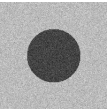
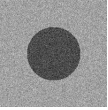
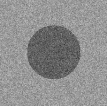
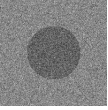
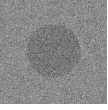
M	3	4	5	6	7	8	9	10
 SNR = 15dB	100	100	99	100	100	99	99	96
 SNR = 10dB	100	100	99	99	99	100	98	99
 SNR = 5dB	100	100	100	100	100	100	99	97
 SNR = 0dB	100	100	100	99	99	98	96	100
 SNR = -5dB	99	96	97	98	90	90	92	92
 SNR = -10dB	45	33	25	25	20	7	7	11

Table 4.2: Percentage of success rate of our snake for noisy synthetic data.

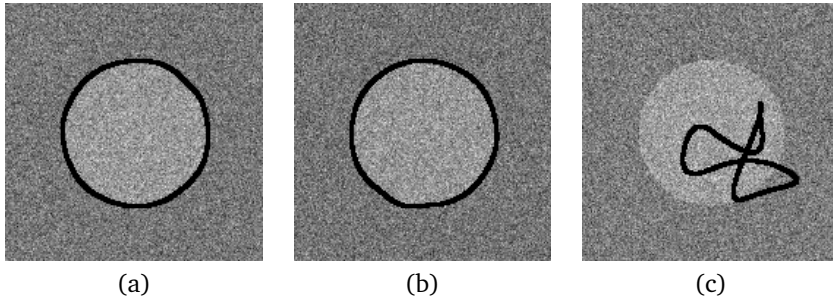


Figure 4.9: Segmentation results for noisy synthetic data with $\text{SNR} = -5\text{dB}$. (a) Barely accepted with $J = 0.853\%$. (b) Rejected with $J = 1.001\%$. (c) Rejected with $J = 81.065\%$.

Furthermore, the increased sensitivity to noise as we increase the number of vector coefficients M corresponds to the appearance of additional noise-related local minima in the energy of the snake. Therefore, M should be chosen as small as possible in order to avoid over-fitting of the noise, but large enough to be able to approximate the shape of interest.

4.4.3 Medical Data

Now, we move away from synthetic data. We compare our snake against other snake variants in terms of accuracy and speed. We quantify their accuracy at outlining the endocardial wall of the left ventricle within slices of 3D cardiac MR image sequences.

The data we used are short-axis cardiac MR image sequences from 33 subjects acquired in the Department of Imaging of the Hospital for Sick Children in Toronto, Canada [145]. For each subject, data consist of a time-series of 20 volumes. For each volume, the number of slices varies from 8 to 15. Each slice is a (256×256) image with a pixel spacing between 0.93 mm and 1.64 mm. The ground truth was obtained by manual annotation. In each segmented image 1,000 points (named *landmark points*) define a closed polygon outlining the endocardial wall.

4.4.3.1 Accuracy For each subject, we selected one slice guided by its anatomical structures along the long axis and its timing in the cardiac cycle. Since the region of interest is nearly elliptical, we used the minimalistic elliptical active contour named *Ovuscule* to provide a first estimate of the location and orientation of the left ventricle [69]. Then, we refined the segmentation of the endocardial wall using the general parametric active contour model (4.1) for different values of M and several basis functions. More specifically, we used linear and quadratic

B-splines, our function (4.8) that we refer to as third-order exponential spline, and an extended version of (4.8) that we refer to as fourth-order exponential spline. The linear B-spline basis function has a smaller support than our function (4.8). However, it can only adopt the form of polygons. The quadratic B-spline basis function has the same support and regularity than (4.8). However, it is unable to reproduce ellipses. Finally, the fourth-order exponential spline is an extended version of (4.8), with one more degree of regularity, but with a support one unit larger. The initialization provided by the Ovusculer could be carried over to (4.8) and to the fourth-order exponential spline. In the case of other types of snakes, the perfect ellipse of the Ovusculer cannot be reproduced but must be approximated. This approximation was achieved by sampling the outline of the Ovusculer.

In a preprocessing step, the images were magnified four times horizontally and vertically. Firstly, we evolved the Ovusculer on the magnified image. Secondly, we evolved more refined snakes, guided exclusively by the edge energy on a smoothed version of the magnified image. The smoothing was Gaussian, with a kernel of variance $\sigma^2 = 10^2$. We then measured the landmark error. We computed this error as the mean distance of the snake to the landmark points given by the ground truth, as was done in [145].

In Figures 4.10, 4.11, and 4.12, we show the mean, median, and maximum values of the landmark error, respectively. From these graphs, we validate that the Ovusculer provides a good and robust starting point to be refined by the snakes investigated in this paper. The polygonal snake does not reach the accuracy of the Ovusculer till $M = 7$, and exhibits a high variance across subject. The quadratic-spline snake and the third-order exponential-spline snake converge to similar accuracies starting with $M = 4$. This was expected, since we showed in Section 4.2.2 that our function does converge to a quadratic B-spline when M increases. However, for low values of M , the difference is noticeable, and the quadratic-spline snakes produce shapes that are not compatible with the region of interest. Finally, the fourth-order exponential-spline snakes produce equivalent results in terms of accuracy and stability than the third-order one, at a price of a larger support, and therefore, of a slower convergence.

In Figure 4.13b, we illustrate the initialization provided to the Ovusculer, and in Figure 4.13c the outcome of optimizing the Ovusculer, which will provide the initialization for further processing. We also show the result of several more elaborated snake variants, and how they compare with the ground truth. The fourth-order exponential-spline snake results in an outline that is visually indistinguishable from that of the third-order one, but comes at an increased cost.

4.4.3.2 Speed In terms of speed, we compared our proposed snake to some classic traditional snakes such as a Kass-like snake [146] and a traditional Geodesic Active Contour (GAC) model [62].

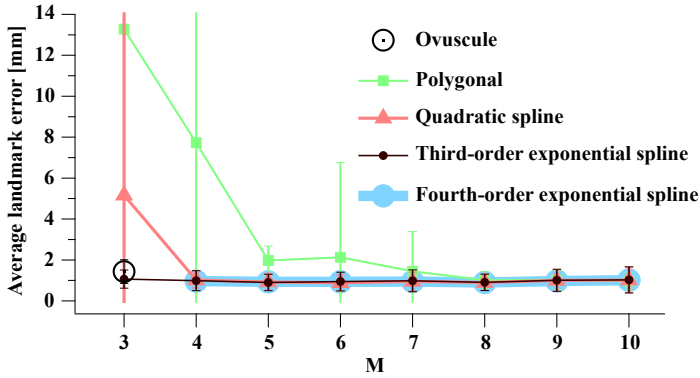


Figure 4.10: Mean and variance of the landmark error across all 33 patients.

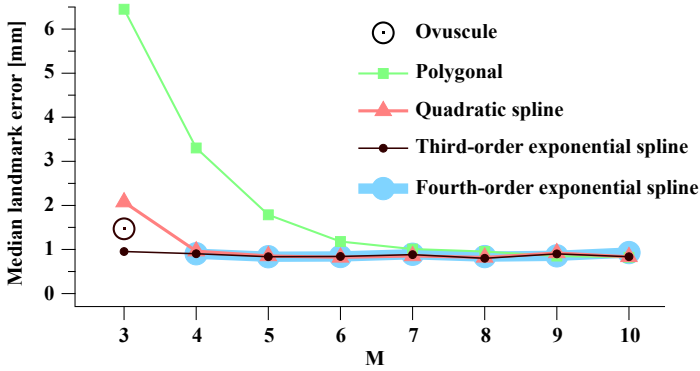


Figure 4.11: Median of the landmark error across all 33 patients.

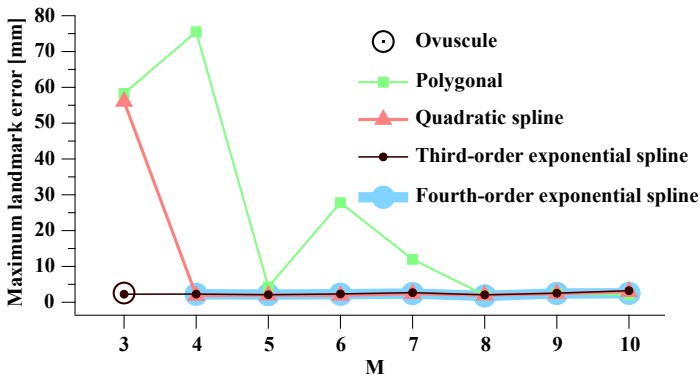


Figure 4.12: Maximum landmark error among all 33 patients.

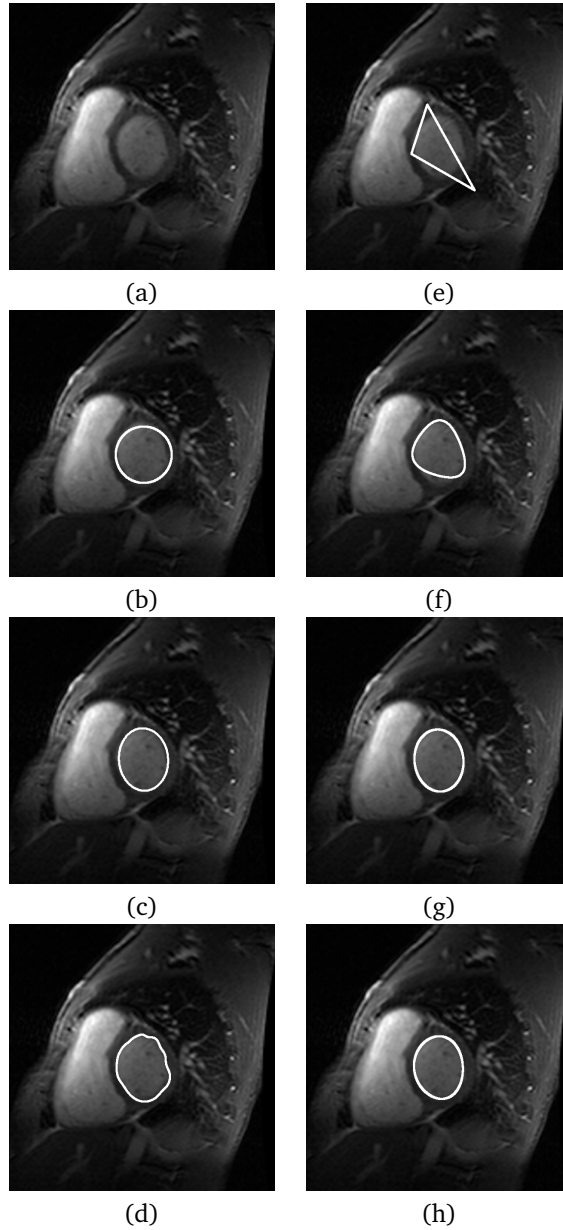


Figure 4.13: Outlining of the endocardial wall. (a) Raw data. (b) Initialization. (c) Ovuscule. (d) Ground truth. (e) Polygonal snake with $M = 3$. (f) Quadratic-spline snake with $M = 3$. (g) Third-order exponential-spline snake with $M = 3$. (h) Fourth-order exponential-spline snake with $M = 4$.

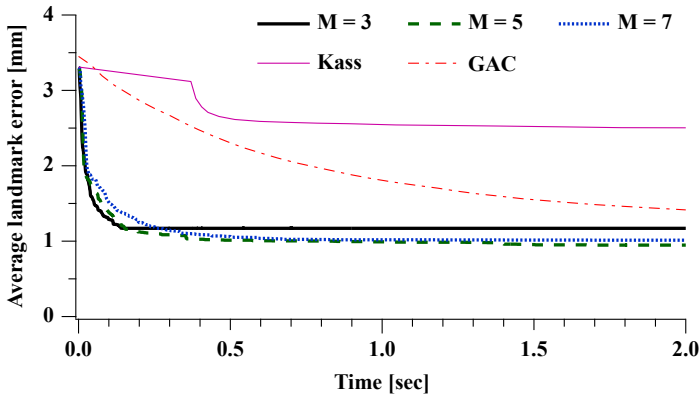


Figure 4.14: Temporal evolution of the Jaccard distance. During the 2 seconds of snake evolution, the proposed method with $M = 3$ performed 1479 iterations, with $M = 5$ it performed 1406 iterations, and with $M = 3$ it performed 889 iterations. The Kass snake performed 17 iterations, the first of which took 370ms, and the GAC performed 34 iterations.

In this analysis, we used the anatomical structures of the 33 patients of Section 4.4.3.1. However, we modified our initialization procedure to accommodate for the GAC model, since it fails unless the initial contour lies totally inside or outside of the boundary of interest. Therefore, we scaled down the initialization that was provided by the outcome of optimizing an Ovusculum in Section 4.4.3.1. By doing so, we guarantee that all initial contours lay inside the endocardial wall to be segmented. Unfortunately, neither the Kass-like snake nor the GAC model are able to reproduce the initial ellipse perfectly and their initialization must be approximated. This approximation was achieved by sampling the outline of the Ovusculum. Finally, we refined the segmentation of the endocardial wall either using our snake model for different values of M , the Kass-like snake, or the GAC.

This experiment was performed on a MacPro 3.1 with two Quad-Core Intel Xeon processors and 8GB of RAM memory running Mac OS X 10.6.8. The implementation of the Kass-like active contour was taken from [146], and the one of GAC model from the free open-source image-processing package Fiji¹ implementing the algorithm described in [62].

In Figure 4.14, we show the mean temporal evolution of the improvement of the Jaccard distance during the snake evolution process for the 33 patients. We can clearly see that the proposed snake reaches its optimum earlier than the classical

1. <http://fiji.sc/>

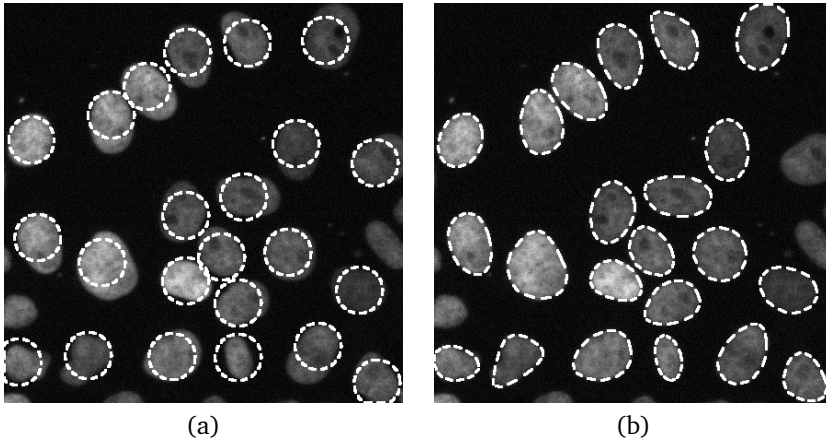


Figure 4.15: Outline of HeLa nuclei in a fluorescence microscopy image. The parametric snakes were built with $M = 5$. (a) The initial contour of the snake. (b) Result provided by our snake.

Kass-like snake and the GAC model. The Kass-like snake has a very costly first step, and then it cannot escape a local minimum. The GAC is executed with an advection value of 2.20, and a propagation value of 1. These parameters make the GAC succeed in overcoming the local minimum, but the convergence rate is still slower than that of the parametric case. It is important to notice that, for our proposed model, an increase in the number M of control points slows down the convergence. As pointed out in Section 4.3.4, this is due to the fact that larger values of M increase the computational load per iteration of the snake.

4.4.4 Real Data

Here, we illustrate the behavior of our snake and provide further insights into its capabilities. In the context of this section, the ground truth is missing, so we must relinquish quantitative assessments in favor of qualitative ones.

4.4.4.1 HeLa Nuclei We want to evaluate the success of our snake model at outlining ellipse-like targets in the context of automated time-lapse microscopy. We use (434×434) images of HeLa nuclei that express fluorescent core histone 2B on an RNAi live cell array. We show in Figure 4.15 the result of the optimization process with (4.8) and $M = 5$. This number of points is high enough to capture small departures from an elliptic shape.

We initialized every snake as a circle of radius of 25 pixel units, as shown in Figure 4.15. These initial circles were centered on the locations given by a maxima detector applied over a version of the image that was smoothed with a

Gaussian kernel of variance $\sigma^2 = 12^2$ pixel. A total number of 23 maxima were detected. We then proceed with an inverted version of the original, unsmoothed image to optimize the snakes. The optimization process converged in 22 cases. We show in Figure 4.15 the result of the outlining process. We observe that our snakes were successful in most of the cases.

4.4.4.2 Droplets As a second example, we show the outline of sprayed and deformed water droplets hitting a surface. The flight and the impact of the droplet was captured by a high-speed camera (Photron Fastcam) at a rate of 10,000 images/s. The shape of the droplet is changing during flight, at impact, and while bouncing. After cropping, the size of the image was (663×663) pixels.

We analyzed two frames. One was an image taken before the collision took place, the other was taken after the impact. In both cases, we initialized the snake as a circle with a position and size that we chose manually. These initializations are shown in Figure 4.16. In the image prior to the impact, which we show in the left part of Figure 4.16, a snake with $M = 5$ was used. We selected a small value for M because the droplet is nearly circular. In the image after the impact, which we show in the right part of Figure 4.16, five control points did not provide enough freedom to cope with the discontinuity created by the attachment to the surface. However, the outline was successfully retrieved when slightly increasing the number of nodes to $M = 8$.

4.5 Conclusions

Our contribution in this chapter is a new family of basis functions that we use to describe parametric contours in terms of a set of control points. We were able to single out the basis of shortest support that allows one to reproduce circles and ellipses. Those can be characterized exactly by as few as three control points but, by considering additional ones, our parametric contours can reproduce with arbitrary precision any planar closed curve. In particular, we have shown that the mean error of approximation decays in inverse proportion of the cube of the number of control points. We have used our ellipse-reproducing parametric curves to build snakes driven by a combination of contour and region-based energies. In the latter case, the energy depends on the contrast between two regions, one being delineated by the curve itself, and the other by an ellipse of double area. To determine this ellipse, we showed how to compute the best elliptical approximation, in a least-squares sense, of a contour described by an arbitrary number of control points. We were able to accelerate the implementation of our snakes by taking advantage of Green's theorem, which was facilitated by the availability of the explicit expressions of our basis. We have applied our snakes to a variety of problems that involve synthetic simulations and real data. We achieved excellent objective and subjective performance.

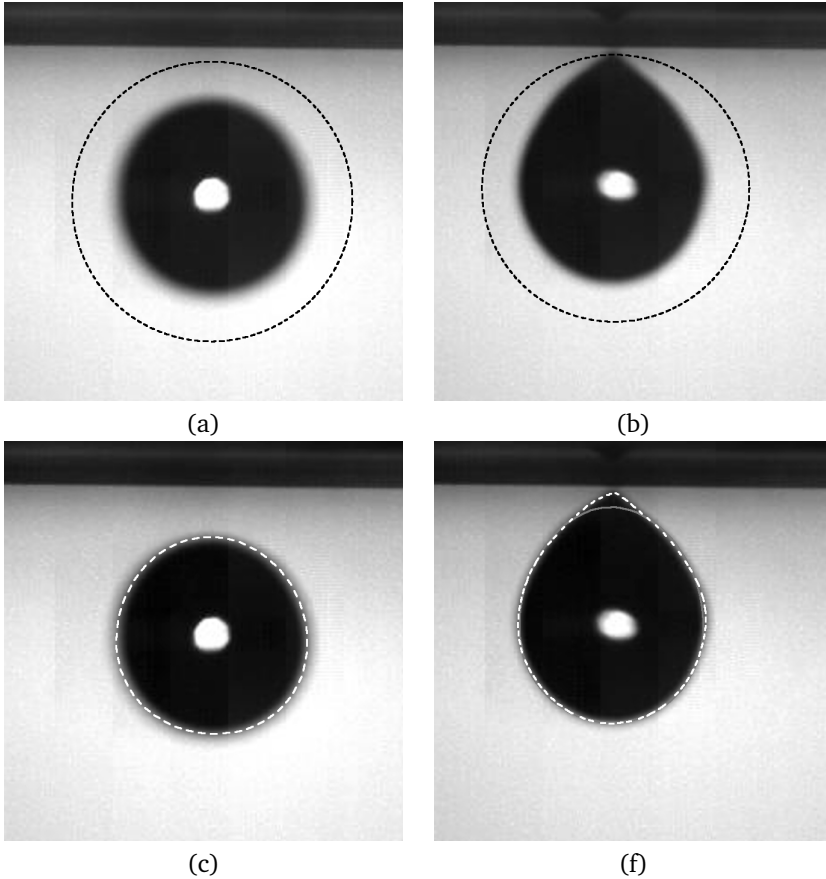


Figure 4.16: Sprayed droplets. (a) Prior to the impact: The initial contour of the snake is represented with a black dashed line. (b) After the impact: The initial contour of the snake is represented with a black dashed line. (c) Prior to the impact: The outline of our snake with $M = 5$ is represented with a white dashed line. (d) After the impact: The outline of the successful snake is represented with a white dashed line ($M = 8$), while the configuration with $M = 5$ is represented with a gray solid line. The droplet edges are partially out of focus, making them blurry and noisy.

Appendices

4.A Implementation Details

In Section 4.3, we provided the guidelines for an efficient implementation of our energy functionals. Here, we derive the explicit expressions of our image energies and their partial derivatives. These expressions are needed when implementing the snake optimization routine.

4.A.1 Image Energy

As described in Section 4.3.1, our image energy is composed of two terms: a contour (or edge) term and a region term.

4.A.1.1 Contour Image Energy Using Green's theorem, our contour energy (4.19) can be expressed as the surface integral

$$E_{\text{edge}} = - \iint_{\Omega} \Delta f(\mathbf{x}) dx_1 dx_2,$$

where $\mathbf{x} = (x_1, x_2)$ and Δf is the Laplacian of the image f . We express the surface integral of $g = -\Delta f$ over the region Ω enclosed by the curve \mathbf{r} as

$$E_{\text{edge}} = \oint_{\mathbf{r}} g_1(\mathbf{r}) dx_2 = \int_0^1 g_1(\mathbf{r}(t)) \frac{dx_2(t)}{dt} dt,$$

where g_1 is the pre-integrated image along the first dimension. Now, by the explicit parametric description of \mathbf{r} , we have

$$E_{\text{edge}} = \int_0^1 g_1(\mathbf{r}(t)) M \sum_{k=0}^{M-1} c_2[k] \dot{\varphi}_M(Mt - k) dt,$$

where $\mathbf{c} = (c_1, c_2)$. Finally, we approximate the integral by the sum

$$E_{\text{edge}} \approx \frac{1}{R} \sum_{i=0}^{MR} g_1 \left(\mathbf{r} \left(\frac{i}{MR} \right) \right) \sum_{k=0}^{M-1} c_2[k] \dot{\varphi}_M \left(\frac{i}{R} - k \right),$$

where R is discretization the sampling rate. Since $\dot{\varphi}$ is compactly-supported, the number of non-zero elements of the inner sum is small. We precompute and store in a lookup table the values of $\dot{\varphi}_M(i/R - k)$.

4.A.1.2 Region Image Energy Our region energy (4.20) can be expressed as

$$E_{\text{region}} = \frac{1}{|\Omega|} \left(2 \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_\lambda} f(\mathbf{x}) dx_1 dx_2 \right),$$

as long as $\Omega \subset \Omega_\lambda$. Then, computing the image energy reduces to the evaluation of two surface integrals over the regions delimited by \mathbf{r} and \mathbf{r}_λ (i.e., Ω and Ω_λ respectively).

We express the surface integral of f over Ω as

$$\oint_{\mathbf{r}} f_1(\mathbf{r}) dx_2 = \int_0^1 f_1(\mathbf{r}(t)) \frac{dx_2(t)}{dt} dt,$$

where f_1 is the pre-integrated image along the first dimension. Now, by the explicit parametric description of \mathbf{r} , we have

$$\int_0^1 f_1(\mathbf{r}(t)) M \sum_{k=0}^{M-1} c_2[k] \dot{\varphi}(Mt - k) dt.$$

Analogously, we can express the surface integral of f over the region Ω_λ enclosed by the curve $\mathbf{r}_\lambda = (x_{\lambda,1}, x_{\lambda,2})$ as

$$\oint_{\mathbf{r}_\lambda} f_1(\mathbf{r}_\lambda) dx_{\lambda,2} = \int_0^1 f_1(\mathbf{r}_\lambda(t)) \frac{dx_{\lambda,2}(t)}{dt} dt.$$

Now, by the explicit parametric description of \mathbf{r}_λ , we have

$$\int_0^1 f_1(\mathbf{r}_\lambda(t)) 2\pi\lambda \sum_{k=0}^{M-1} c_2[k] (h_s[k] \cos(2\pi t) - h_c[k] \sin(2\pi t)) dt.$$

We obtain an explicit expression for the region energy by combining both results

$$E_{\text{region}} = \frac{1}{|\Omega|} \left(2 \int_0^1 f_1(\mathbf{r}(t)) M \sum_{k=0}^{M-1} c_2[k] \dot{\varphi}(Mt - k) - f_1(\mathbf{r}_\lambda(t)) 2\pi\lambda \sum_{k=0}^{M-1} c_2[k] (h_s[k] \cos(2\pi t) - h_c[k] \sin(2\pi t)) dt \right).$$

Finally, we approximate the integral by the sum

$$E_{\text{region}} \approx \frac{1}{|\Omega|R} \left(2 \sum_{i=0}^{MR} f_1 \left(\mathbf{r} \left(\frac{i}{MR} \right) \right) \sum_{k=0}^{M-1} c_2[k] \dot{\varphi}_M \left(\frac{i}{R} - k \right) - \frac{2\pi}{M} \lambda f_1 \left(\mathbf{r}_\lambda \left(\frac{i}{MR} \right) \right) \sum_{k=0}^{M-1} c_2[k] \left(h_s[k] \cos \frac{2\pi i}{MR} - h_c[k] \sin \frac{2\pi i}{MR} \right) \right),$$

where R is the discretization sampling rate. We precompute and store in lookup tables the values of $\dot{\varphi}_M(i/R - k)$ and $(h_s[k] \cos \frac{2\pi i}{MR} - h_c[k] \sin \frac{2\pi i}{MR})$.

4.A.2 Partial Derivatives of the Image Energy

Our optimization scheme requires the partial derivatives of the energy function with respect to the parameters, that is, the sequence of control points $\{\mathbf{c}[k]\}_{k \in [0 \dots M-1]}$.

4.A.2.1 Partial Derivatives of the Contour Image Energy Since the edge-based image energy can be expressed as a surface integral of a function $g = -\Delta f$, we can compute the partial derivatives as

$$\begin{aligned}
 \frac{\partial E_{\text{edge}}}{\partial c_1[k]} &= \frac{\partial}{\partial c_1[k]} \oint_{\mathbf{r}} g_1(\mathbf{r}) dx_2 && \text{(by (4.22))} \\
 &= \frac{\partial}{\partial c_1[k]} \int_0^1 g_1(\mathbf{r}(t)) \frac{dx_1(t)}{dt} dt && \text{(line integral)} \\
 &= \int_0^1 \underbrace{\frac{\partial}{\partial x_1} \{g_1(\mathbf{r}(t))\}}_{g(\mathbf{r}(t))} \underbrace{\frac{\partial x_1(t)}{\partial c_1[k]}}_{\varphi_M(Mt-k)} \frac{dx_1(t)}{dt} dt && \text{(chain rule)} \\
 &= \int_0^1 g(\mathbf{r}(t)) \varphi_M(Mt-k) M \sum_{i=0}^{M-1} c_2[i] \dot{\varphi}_M(Mt-i) dt && \text{(by (4.2))} \\
 &= \sum_{i=0}^{M-1} c_2[i] M \underbrace{\int_0^1 g(\mathbf{r}(t)) \varphi_M(Mt-k) \dot{\varphi}_M(Mt-i) dt}_{Q_g[k,i]} && \text{(reordering).}
 \end{aligned}$$

Thus, we obtain the simplified expression

$$\frac{\partial E_{\text{edge}}}{\partial c_1[k]} = \sum_{i=0}^{M-1} c_2[i] Q_g[k, i].$$

In a similar manner, we get

$$\frac{\partial E_{\text{edge}}}{\partial c_2[k]} = - \sum_{i=0}^{M-1} c_1[i] Q_g[k, i].$$

To summarize, the computation of the partial derivatives of the edge energy reduces to the trivial computation of the finite sequence Q_g . Since φ and $\dot{\varphi}$ are compactly supported, $Q_g[k, i]$ differ from zero if and only if $\varphi_M(Mt-k)$ and $\dot{\varphi}_M(Mt-i)$ overlap. Formally, $Q_g[k, i] \neq 0$ if $\min\{|k-i|, |i-k|\} < N$, where N is the common support length of φ and $\dot{\varphi}$. Then, if M is large compared to N or if the length of the support of the basis functions is short, then most of the elements of $Q_g[k, i]$ are zero.

4.A.2.2 Partial Derivatives of the Region Image Energy We compute the partial derivatives of the region-based image energy as

$$\frac{\partial E_{\text{region}}}{\partial c_1[k]} = \frac{\partial}{\partial c_1[k]} \frac{1}{|\Omega|} \left(2 \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_\lambda} f(\mathbf{x}) dx_1 dx_2 \right).$$

By using differentiation properties and expanding the expression, we obtain

$$\begin{aligned} \frac{\partial E_{\text{region}}}{\partial c_1[k]} &= -\frac{E_{\text{region}}}{|\Omega|} \frac{\partial |\Omega|}{\partial c_1[k]} \\ &+ \frac{1}{|\Omega|} \left(2 \frac{\partial}{\partial c_1[k]} \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \frac{\partial}{\partial c_1[k]} \iint_{\Omega_\lambda} f(\mathbf{x}) dx_1 dx_2 \right). \end{aligned}$$

We can expand the first term using (4.21). We obtain,

$$-\frac{E_{\text{region}}}{|\Omega|} \frac{\partial |\Omega|}{\partial c_1[k]} = \frac{E_{\text{region}}}{|\Omega|} \sum_{n=0}^{M-1} c_2[n] \underbrace{\int_0^M \varphi_M(t-k) \dot{\varphi}_M(t-n) dt}_{Q_1[k,n]}.$$

The second term can be expanded following the strategy used in Section 4.A.2.1. We obtain

$$\frac{1}{|\Omega|} \left(\sum_{i=0}^{M-1} c_2[i] \left(2Q_f[k, i] - \tilde{Q}_f[k, i] \right) \right),$$

where \tilde{Q}_f contains the integration of f along the elliptical shell.

To summarize, the partial derivatives of the region-based image energy can be written as

$$\frac{\partial E_{\text{region}}}{\partial c_1[k]} = \frac{1}{|\Omega|} \sum_{i=0}^{M-1} c_2[i] \left(E_{\text{region}} Q_1[k, i] + 2Q_f[k, i] - \tilde{Q}_f[k, i] \right).$$

In a similar manner, we get

$$\frac{\partial E_{\text{region}}}{\partial c_2[k]} = -\frac{1}{|\Omega|} \sum_{i=0}^{M-1} c_1[i] \left(E_{\text{region}} Q_1[k, i] + 2Q_f[k, i] - \tilde{Q}_f[k, i] \right).$$

Chapter 5

Extension to 3D Spline Snakes

Many 3D snake variants have been proposed utilizing different types of surface representation and various energy terms. Implicit methods based on a level-set formulation of the Chan-and-Vese problem have been investigated extensively over the past decade [70]. Some effort was invested to obtain semi-parametric approaches using simplex meshes [147] and, more recently, 3D triangular meshes [148]. A first approach to fully parametric snakes named active geometric functions (AGF) was proposed in [149] using the variational framework and the Mumford-Shah energy functional. Then, a refinement of the AGF method was presented using polynomial B-splines [150].

In this chapter, we extend our 2D parametric snakes exposed in Chapter 4 and propose the first fully 3D spline-based parametric snake for the analysis of images in 3D microscopy in which we constrain the topology to segment ellipsoid-like objects of the type encountered in cell biology. Our snake surface is parameterized by few control points and uses as basis functions a special kind of exponential B-splines from the family investigated in Chapter 3. The most important feature of our basis is that it allows our 3D snake to perfectly reproduce ellipsoids. Our bases have the shortest-possible support given the aforementioned ellipsoid reproduction property. Because they are also refinable, they also provide a good approximation of any closed surface with a sphere-like topology.

The parameterization based on splines, and more precisely the use of our exponential B-splines, allows us to derive a fast algorithm for image segmentation. This is crucial for biological applications such as cell tracking in time-lapse sequences of 3D images, which produce tremendous amounts of data. We have investigated the efficiency of the proposed approach with the analysis of several sets of real microscopic images and are reporting real-time performance. We have designed edge and region energies that admit a fast implementation thanks to the use of pre-integrated images and Gauss' theorem. We also pro-

pose a simple method to detect self-intersections of the surface during the snake evolution.

The class of parametric snakes proposed here lends itself to a semi-automatic segmentation scheme that allows for user-friendly interaction. Because the snake is fully parameterized by only a few 3D control points, the user is able to easily guide and modify it by interacting with anchors in dedicated 2D and 3D image views. These views feature a live display of the snake and provides feedback to the user when a control point is modified. This ability is precious for crowded biological environments which may require user input and feedback. The software implementing our techniques is given as an open-source library in an effort to provide useful tools for the bioimaging community.

This chapter is organized as follows: In Section 5.1, we present an extension of the curve generation framework introduced in Chapter 3 in order to generate parametric surfaces using B-splines. Then, we formalize the mathematical conditions for the basis functions in order to make the parametric surfaces suitable for segmentation. Next, we specify a 3D snake model with a sphere-like topology. Implementation details such as energy functionals and discretization issues are addressed in Section 5.2. Finally, we illustrate the capabilities of our snake with synthetic and real data in Section 5.3.

5.1 Spline Surfaces

In this section, we extend the general framework of parametric curve representation investigated in Chapter 3 to parametric surfaces using B-splines as basis functions, and provide explicit expressions that characterize the surface points and the tangent spaces. Then, we provide a formal set of conditions for the basis functions to ensure unique and stable representation of the surfaces, affine invariance of the model, well-definiteness of the surface curvature, and some reproduction properties. Finally, we specify a 3D snake model with a sphere-like topology capable of perfectly reproducing ellipsoids irrespective of their position and orientation.

5.1.1 Parametric Representation of Surfaces

We consider a parametric representation of a surface $\sigma(u, v)$ in 3D space that is described by a triplet of Cartesian coordinate functions $x_1(u, v)$, $x_2(u, v)$ and $x_3(u, v)$, where $u, v \in \mathbb{R}$ are continuous parameters. The two-dimensional functions x , y and z are represented by linear combinations of suitable basis functions. Among all possible bases, we focus on those derived from a compactly supported generator $\Phi : \mathbb{R}^2 \mapsto \mathbb{R}$ and its multi-integer shifts $\{\Phi(u - k, v - l)\}_{(k,l) \in \mathbb{Z}^2}$.

Then, the representation of the surface is given by the vectorial equation

$$\sigma(u, v) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \Phi \left(\frac{u}{T_1} - k, \frac{v}{T_2} - l \right), \quad (5.1)$$

where $\{\mathbf{c}[k, l] \in \mathbb{R}^3\}_{(k,l) \in \mathbb{Z}^2}$ are the control points in 3D that define the shape of the surface, and $T_1, T_2 \in (0, \infty)$ are the sampling steps for each parametric dimension. We denote by $S \subset \mathbb{R}^3$ the set of points of the surface.

In view of the nature of the domain set defined by u and v , a common strategy is to consider using tensor-products for the construction of the base function Φ . Then, the generator can be written as

$$\Phi(u, v) = \phi_1(u) \phi_2(v). \quad (5.2)$$

This approach of representing surfaces using bases built on tensor-products of one dimensional functions has been studied by several authors [151, 152, 153], albeit not in the context of snakes. Various choices of ϕ_1 and ϕ_2 have been considered, such as polynomials, polynomial B-splines and trigonometric B-splines. Moreover, this tensor-product decomposition choice allows us to take advantage of fast and stable interpolation algorithms [114, 115, 116].

We define the tangent space at any point on the surface $\mathbf{p} = \sigma(u_0, v_0) \in S$ as the vector space generated by the tangent vectors to S at \mathbf{p}

$$\mathbf{T}_1 = \frac{\partial \sigma}{\partial u}(u, v)|_{(u_0, v_0)} \quad (5.3)$$

$$\mathbf{T}_2 = \frac{\partial \sigma}{\partial v}(u, v)|_{(u_0, v_0)}. \quad (5.4)$$

The tangent bundle is usually defined as the disjoint union of all tangent spaces indexed by the points on the surface $\mathbf{p} \in S$. The tangent bundle is said to be well defined if all tangent spaces have dimension equal to two; that is, they are planes. In this case, the surface S is said to be regular [154]. Requiring S to be regular implies that the surface should be smooth, not self-intersect, or have any border. Loosely speaking, S should locally *look like a plane*. Under these conditions, a normal vector to S at $\mathbf{p} = \sigma(u_0, v_0)$ can be computed by

$$\mathbf{n} = \mathbf{T}_1 \times \mathbf{T}_2, \quad (5.5)$$

where \times denotes the three-dimensional cross product.

5.1.2 Desirable Properties of Basis Functions

We now enumerate the conditions that our parametric surface model should satisfy for the purpose of shape segmentation and introduce the corresponding mathematical formalism. These conditions are the bivariate analogous to the ones we presented in Section 4.1.2 in the context of parametric curves.

1. *Unique and Stable Representation.* We want our parametric functions x_1 , x_2 and x_3 to be defined in terms of the coefficients in a unique fashion; that is, x_1 , x_2 and x_3 are uniquely determined by a single sequence of coefficients $\{\mathbf{c}[k, l]\}_{(k,l) \in \mathbb{Z}^2}$ for all $u, v \in \mathbb{R}$. Furthermore, for computational purposes, we ask the interpolation procedure to be numerically stable. A bivariate generating function Φ is said to satisfy the Riesz basis condition if there exist two constants $0 < A \leq B < \infty$ such that

$$A \|\mathbf{c}\|_{\ell_2} \leq \|\boldsymbol{\sigma}\|_{L_2} \leq B \|\mathbf{c}\|_{\ell_2} \quad (5.6)$$

for all $\mathbf{c} \in \ell_2$. A direct consequence of the lower inequality of (5.6) is that the condition $\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \Phi\left(\frac{u}{T_1} - k, \frac{v}{T_2} - l\right) = 0$ for all $(u, v) \in \mathbb{R}^2$ implies that $\mathbf{c}[k, l] = 0$ for all $(k, l) \in \mathbb{Z}^2$. Thus, the basis functions are linearly independent and every function is uniquely specified by its coefficients. The upper inequality ensures the stability of the interpolation process [116].

It has been shown in [117] that, due to the integer-shift-invariant structure of the representation, the Riesz basis condition has the following equivalent expression in the Fourier domain:

$$A \leq \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \left| \hat{\Phi}(\omega_1 + 2\pi k, \omega_2 + 2\pi l) \right|^2 \leq B \quad (5.7)$$

for all $(\omega_1, \omega_2) \in \mathbb{R}^2$, where $\hat{\Phi}(\omega_1, \omega_2) = \iint_{\mathbb{R}^2} \Phi(u, v) e^{-j(\omega_1 u + \omega_2 v)} du dv$ denotes the two-dimensional Fourier transform of Φ . Once expressed in the Fourier domain, the Riesz condition provides a practical way to verify whether a given generating function Φ satisfies (5.6) or not.

Given the fact that Φ is built from the tensor product of two one-dimensional functions, ϕ_1 and ϕ_2 , a sufficient condition to satisfy (5.7) is to require ϕ_1 and ϕ_2 to satisfy the one-dimensional Riesz condition. In particular, they should satisfy

$$\begin{aligned} A_1 &\leq \sum_{k=-\infty}^{\infty} \left| \hat{\phi}_1(\omega + 2\pi k) \right|^2 \leq B_1 \\ A_2 &\leq \sum_{l=-\infty}^{\infty} \left| \hat{\phi}_2(\omega + 2\pi l) \right|^2 \leq B_2, \end{aligned}$$

where $\hat{\phi}_1$ and $\hat{\phi}_2$ are the one-dimensional Fourier transforms of ϕ_1 and ϕ_2 , respectively, and A_1, B_1, A_2 and B_2 are the corresponding Riesz bounds.

2. *Affine Invariance.* Since we are interested in outlining shapes irrespective of their position and orientation, we would like our model to be invariant to affine transformations. We formalize this by requiring

$$\mathbf{A} \boldsymbol{\sigma}(u, v) + \mathbf{b} = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} (\mathbf{A} \mathbf{c}[k, l] + \mathbf{b}) \Phi\left(\frac{u}{T_1} - k, \frac{v}{T_2} - l\right),$$

where \mathbf{A} is a (3×3) matrix and \mathbf{b} is a three-dimensional vector. From (5.8), it is easy to show that affine invariance is ensured if and only if

$$\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \Phi \left(\frac{u}{T_1} - k, \frac{v}{T_2} - l \right) = 1 \quad (5.8)$$

for all $(u, v) \in \mathbb{R}^2$. In the literature, this constraint is often named the *partition-of-unity* condition [118].

Since Φ is built from the tensor product of two one-dimensional functions, ϕ_1 and ϕ_2 , (5.8) holds if and only if

$$\begin{aligned} \forall u \in \mathbb{R} & : \sum_{k=-\infty}^{\infty} \phi_1 \left(\frac{u}{T_1} - k \right) = 1 \\ \forall v \in \mathbb{R} & : \sum_{l=-\infty}^{\infty} \phi_2 \left(\frac{v}{T_2} - l \right) = 1, \end{aligned}$$

hold; that is, if both ϕ_1 and ϕ_2 satisfy the one-dimensional partition-of-unity condition.

3. *Well-Defined Gaussian Curvature.* The Gaussian curvature of a parametric surface at a point \mathbf{p} can be expressed as the ratio of the determinants of the second and first fundamental forms at the same point

$$K(\mathbf{p}) = \frac{\det \mathbf{II}}{\det \mathbf{I}}.$$

The first fundamental form at a point \mathbf{p} is the inner product on the tangent space of a surface in three-dimensional Euclidean space which is induced canonically from the dot product of \mathbb{R}^3 . It is usually expressed as a symmetric matrix

$$\mathbf{I} = \begin{pmatrix} \mathbf{T}_1 \cdot \mathbf{T}_1 & \mathbf{T}_1 \cdot \mathbf{T}_2 \\ \mathbf{T}_1 \cdot \mathbf{T}_2 & \mathbf{T}_2 \cdot \mathbf{T}_2 \end{pmatrix},$$

where \mathbf{T}_1 and \mathbf{T}_2 are the tangent vectors defined in (5.3) and (5.4) at a point \mathbf{p} , respectively.

The second fundamental form at a point \mathbf{p} is a quadratic form on the tangent plane in the three-dimensional Euclidean space,

$$\mathbf{II} = \begin{pmatrix} \frac{\partial^2 \boldsymbol{\sigma}}{\partial u^2} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \boldsymbol{\sigma}}{\partial u \partial v} \cdot \hat{\mathbf{n}} \\ \frac{\partial^2 \boldsymbol{\sigma}}{\partial u \partial v} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \boldsymbol{\sigma}}{\partial v^2} \cdot \hat{\mathbf{n}} \end{pmatrix},$$

where $\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|}$ denotes the normal unit vector, which can be computed using (5.5). Together with the first fundamental form, it serves to define extrinsic invariants of the surface [154].

We want our parametric surface to have a well-defined Gaussian curvature at every point on the surface. To do so, each coordinate function (or, equivalently, the functions ϕ_1 and ϕ_2) must be in \mathcal{C}^1 with bounded second derivative.

4. *Reproduction of Particular Shapes.* Ellipsoids are particular shapes that appear repeatedly in segmentation problems in 3D microscopy. For that reason, it is important that our surface model (5.1) perfectly reproduces them. Formally, we want a parametric description of all ellipsoids to lie within the span of (5.1). For that reason the basis $\{\Phi(u-k, v-l)\}_{(k,l) \in \mathbb{Z}^2}$ should reproduce the family of functions that describe any ellipsoid. Since our model is vectorial, we can impose this condition component-wise. In mathematical terms, we say that a bivariate generating function Φ reproduces a bivariate function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ if and only if there exists a sequence of coefficients $\{c[k, l] \in \mathbb{R}\}_{(k,l) \in \mathbb{Z}^2}$ such that

$$f(u, v) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} c[k, l] \Phi(u-k, v-l)$$

holds almost everywhere. The analysis is further simplified if the function f is separable in u and v , that is, there exists a decomposition such that $f(u, v) = f_1(u) f_2(v)$. In this situation, it is sufficient to approach the problem separately for each variable. Then, we say that Φ reproduces the function f if there exists sequences $\{c_1[k]\}_{k \in \mathbb{Z}}$ and $\{c_2[l]\}_{l \in \mathbb{Z}}$ such that

$$\begin{aligned} f_1(u) &= \sum_{k=-\infty}^{\infty} c_1[k] \phi_1(u-k) \\ f_2(v) &= \sum_{l=-\infty}^{\infty} c_2[l] \phi_2(v-l) \end{aligned}$$

holds almost everywhere. In addition, we say that ϕ_1 reproduces f_1 and that ϕ_2 reproduces f_2 .

5.1.3 3D Snake Model

We are especially interested in the case when S is a closed surface since we want our snake to segment blob-like objects within 3D images. We define our 3D snake model as a closed parametric surface σ following the parametric vectorial equation (5.1). Since the surface is closed, it is not necessary to consider the parameters (u, v) taking all possible values in \mathbb{R}^2 . It is enough to consider a domain that is a compact set $\Omega \subset \mathbb{R}^2$. By convention, we normalize the range of the parameters u, v to lie within $[0, 1]$, setting the domain to $\Omega = [0, 1]^2$.

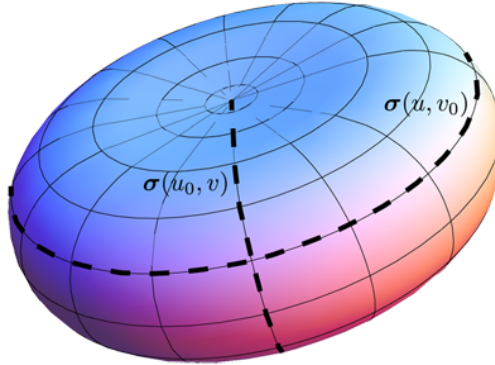


Figure 5.1: Rendering of the snake surface taking a form of an oblate spheroid flattened at the poles. The mesh has been obtained by sampling (5.9).

As discussed in Section 5.1.1, we consider among all possible bases those derived from the tensor product of two compactly supported generators ϕ_1, ϕ_2 . Then, the parametric representation of the surface is given by the vectorial equation

$$\sigma(u, v) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \phi_1(M_1 u - k) \phi_2(M_2 v - l), \quad (5.9)$$

where we have substituted the sampling steps T_1 and T_2 by the positive integers $M_1 = \frac{1}{T_1}$ and $M_2 = \frac{1}{T_2}$. The fact that we impose here M_1 and M_2 to be positive integers guarantees that the functions x_1, x_2 and x_3 are represented by an integer number of basis functions within their domain $\Omega = [0, 1]^2$. The larger these values are, the more basis functions come into the domain Ω . In this situation, σ has more degrees of freedom and can represent a larger variety of shapes. In other words, small numbers lead to constrained shapes for the snake, and large numbers lead to additional flexibility and more general shapes.

From the desired conditions in Section 5.1.2, the particular choice of ϕ_1 and ϕ_2 determines the properties of the surface generated by (5.9), such as smoothness, computational load of the resulting model, or reproduction of particular shapes.

We can force the surface generated by (5.9) to take the topology of an ellipsoid by imposing the appropriate boundary conditions on the sequence of control points $\{\mathbf{c}[k, l]\}_{(k,l) \in \mathbb{Z}^2}$, and we can make the snake reproduce all possible ellipsoids with the appropriate choice of ϕ_1 and ϕ_2 .

5.1.3.1 Topology To describe the parameterization, we shall adopt an earth-like cartographic terminology referring to meridians and circles of latitude. Then, the curves that are obtained when fixing the second parameter in (5.9) correspond

to circles of latitude, as shown in Figure 5.1. Formally, the curves obtained when v is constant in σ must be closed. As a consequence, the functions of each component of σ in u are 1-periodic when v is constant. In order to satisfy this condition, it is necessary to apply periodic boundary conditions along the first index of the sequence of control points. Therefore, the sequence of coefficients becomes M_1 -periodic and satisfies $\mathbf{c}[k, l] = \mathbf{c}[k + M_1, l]$. Under these conditions, we can reorganize the first infinite summation in (5.9) to a finite one involving periodized basis functions. Then, the parametric representation of the surface is expressed as

$$\sigma(u, v) = \sum_{k=0}^{M_1-1} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \phi_{1,\text{per}}(M_1 u - k) \phi_2(M_2 v - l), \quad (5.10)$$

where $\phi_{1,\text{per}}(u) = \sum_{n=-\infty}^{\infty} \phi_1(u - M_1 n)$ for all $u \in \mathbb{R}$. Moreover, continuing with our earth simile, the curves that are obtained when fixing u in (5.9) correspond to meridians; that is, open curves starting at the north pole \mathbf{c}_N and ending at the south pole \mathbf{c}_S .

5.1.3.2 Reproduction of Ellipsoids We are also interested in our snake being capable of perfectly reproducing ellipsoids irrespective of their size, position and orientation. Note that if ϕ_1 and ϕ_2 are chosen to satisfy the partition-of-unity condition, the snake model will satisfy the affine invariance property. Then, since every ellipsoid can be obtained by an affine transformation of a sphere of unit radius, we focus on the reproduction of this simpler shape.

The classical parametrization of a sphere with unit radius that goes with our earth-like description is given by

$$\begin{cases} x_1(u, v) &= \cos(2\pi u) \sin(\pi v) \\ x_2(u, v) &= \sin(2\pi u) \sin(\pi v) \\ x_3(u, v) &= \cos(\pi v), \end{cases} \quad (5.11)$$

where $u, v \in [0, 1]$.

All parametric equations in (5.11) are separable in u and v , and each part can be efficiently taken care of with our separable model (5.10). In order Φ to be able to reproduce (5.11) and satisfy the condition of Section 5.1.2, ϕ_1 must reproduce constants and sinusoids of unit period, and ϕ_2 must reproduce constants and sinusoids of double period.

We rely once more in the Minimal-Support Generating Functions Theorem detailed in Section 3.2.5 to determine the optimal choice of ϕ_1 and ϕ_2 . More precisely, by using Corollary 1 in Section 3.4.1.1, we know that the centered generating function with minimal support that satisfies the Riesz basis condition, the partition-of-unity condition, is $\mathcal{C}^1(\mathbb{R})$ with bounded second derivative

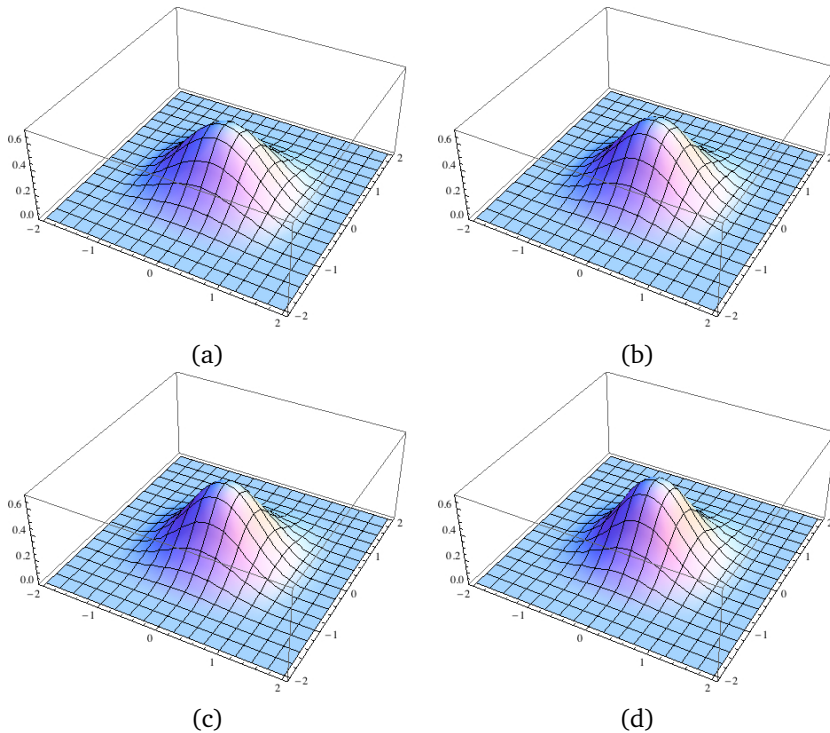


Figure 5.2: Members of the family of functions Φ indexed by $M = M_1 = 2M_2$. (a) $M = 3$. (b) $M = 4$. (c) $M = 5$. (d) $M \rightarrow \infty$. The function with the lowest peak at the origin corresponds to $M = 3$, and, as M increases, the central peak increases as well.

and reproduces sinusoids of unit period with M coefficients is

$$\varphi_M(t) = \frac{1}{1 - \cos \frac{2\pi}{M}} \begin{cases} \cos \frac{2\pi|t|}{M} \cos \frac{\pi}{M} - \cos \frac{2\pi}{M} & 0 \leq |t| < \frac{1}{2} \\ \left(\sin \frac{\pi(3/2-|t|)}{M} \right)^2 & \frac{1}{2} \leq |t| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |t|. \end{cases}$$

Therefore, we take $\phi_1(u) = \varphi_{M_1}(u)$ and $\phi_2(v) = \varphi_{2M_2}(v)$ in order to be able to reproduce all sinusoids in (5.11). Note that ϕ_1 and ϕ_2 are equal if and only if $M_1 = 2M_2$.

We show in Figure 5.2 some members of the family of functions Φ indexed by $M = M_1 = 2M_2$. They are continuous, have finite support of area (3×3) , and tend to be bump-like. Moreover, when $M \rightarrow \infty$, they converge to the tensor product of two quadratic B-splines. These functions allow the snake to per-

fectly replicate constants and sinusoids of the appropriate frequency at each component. This means that they can perfectly reproduce the parametric equations (5.11) with the appropriate configuration of control points. The most remarkable feature, though, is that the size of the support of these functions are the shortest possible to satisfy the conditions from Section 5.1.2. This results in a maximally efficient scheme due to the fact that the computation of each point on the surface of the snake depends on the minimum possible number of basis functions, and it also ensures a local control of the surface by modifying single control points.

The explicit expression of the control points that make our snake take the shape of a perfect a unit sphere is

$$\mathbf{c}[k, l] = \begin{pmatrix} c_{M_1}[k] s_{2M_2}[l] \\ s_{M_1}[k] s_{2M_2}[l] \\ c_{2M_2}[l] \end{pmatrix},$$

where

$$c_M[n] = \frac{2 \left(1 - \cos \frac{2\pi}{M}\right)}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \cos \frac{2\pi n}{M}$$

$$s_M[n] = \frac{2 \left(1 - \cos \frac{2\pi}{M}\right)}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \sin \frac{2\pi n}{M}.$$

The derivation of the specific coefficients to reproduce the sinusoids is equivalent to the one obtained in Section 4.2.5.

5.1.3.3 Smoothness The chosen basis functions are $\mathcal{C}^1(\mathbb{R})$, and their second derivatives are bounded. However, the parametric model (5.10) has two singular points where the continuity and smoothness are not guaranteed and require special attention: the poles \mathbf{c}_N and \mathbf{c}_S .

In order for the surface to be well-defined and continuous at the poles, we require the function $\boldsymbol{\sigma}(u, v)$ to be independent of u for $v = 0$ and $v = 1$. Moreover, to ensure that the tangent plane varies continuously, we need to make some assumptions about the partial derivatives of $\boldsymbol{\sigma}$. It was shown in [151] that a sufficient condition for continuity of the tangent plane is that the partial derivatives satisfy

$$\frac{\partial \boldsymbol{\sigma}}{\partial v}(u, v)|_{v=0} = \mathbf{T}_{1,N} \cos(2\pi u) + \mathbf{T}_{2,N} \sin(2\pi u) \quad (5.12)$$

$$\frac{\partial \boldsymbol{\sigma}}{\partial v}(u, v)|_{v=1} = \mathbf{T}_{1,S} \cos(2\pi u) + \mathbf{T}_{2,S} \sin(2\pi u), \quad (5.13)$$

where $\mathbf{T}_{1,N}, \mathbf{T}_{2,N}, \mathbf{T}_{1,S}, \mathbf{T}_{2,S} \in \mathbb{R}^3$ are free vectors.

Tensor product polynomial splines on the sphere have already been considered in [155] in the context of estimation techniques for fitting data on the sphere. However, no attempt was made to deal with the *pole problem* nor to take full advantage of the use of B-splines. Then, the sufficient conditions to obtain a continuously varying tangent plane, or equivalently $\mathcal{C}^1(\mathbb{R})$ on the surface, were first formulated in [151] within the same context of fitting data on the sphere using polynomial B-splines. However, the proposed scheme could only fulfill the conditions approximately, the main reason being that the conditions (5.12) and (5.13) can only be satisfied if the basis function associated to u is capable of reproducing sinusoids of unit period. An extension of this work was presented in [153], where the basis function associated to u was substituted by periodic trigonometric splines, being able to satisfy (5.12) and (5.13). Here, we use the full potential of the underlying exponential B-splines within our basis functions, and we can satisfy (5.12) and (5.13) thanks to the fact that the basis function ϕ_1 that we selected in Section 5.1.3.2 is capable of reproducing sinusoids. To the best of our knowledge, this has not been done previously.

We would like our parameterization include implicitly the exposed conditions. Now, we translate them as conditions over the control points. We categorize the required conditions in two types:

1. *Pole Interpolation Conditions.* All meridians originate at the north pole if and only if $\sigma(u, 0) = \mathbf{c}_N$ for all $u \in [0, 1]$. Likewise, all meridians terminate at the south pole if and only if $\sigma(u, 1) = \mathbf{c}_S$ for all $u \in [0, 1]$. The condition concerning the north pole can be rewritten in terms of the control points by evaluating (5.10) at $v = 0$. This yields

$$\mathbf{c}_N = \mathbf{c}[k, 1] \phi_2(-1) + \mathbf{c}[k, 0] \phi_2(0) + \mathbf{c}[k, -1] \phi_2(1),$$

for all $k \in [0 \dots M_1 - 1]$, where we have used the fact that ϕ_2 satisfies the partition-of-unity condition and that its support is limited to the interval $[-\frac{3}{2}, \frac{3}{2}]$. Likewise, the condition concerning the south pole can be rewritten in terms of the control points by evaluating (5.10) at $v = 1$:

$$\mathbf{c}_S = \mathbf{c}[k, M_2 + 1] \phi_2(-1) + \mathbf{c}[k, M_2] \phi_2(0) + \mathbf{c}[k, M_2 - 2] \phi_2(1),$$

for all $k \in [0 \dots M_1 - 1]$.

2. *Pole Smoothness Conditions.* The sufficient conditions (5.12) and (5.13) were stated for any generic parameterization. For our case, this translates

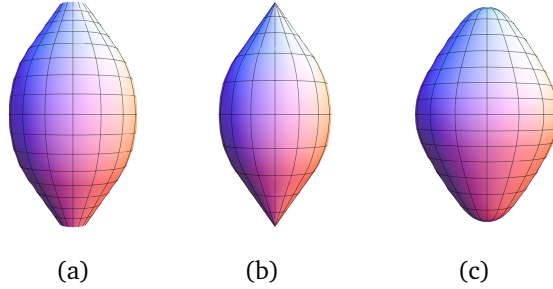


Figure 5.3: Surfaces generated by our snake model. (a) Interpolation conditions are not satisfied. (b) Smoothness conditions are not satisfied. (c) Interpolation and smoothness conditions are satisfied.

into

$$\begin{aligned}
 & \mathbf{T}_{1,N} \cos(2\pi u) + \mathbf{T}_{2,N} \sin(2\pi u) \\
 &= M_2 \sum_{k=0}^{M_1-1} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \phi_{1,\text{per}}(M_1 u - k) \dot{\phi}_2(-l) \\
 & \mathbf{T}_{1,S} \cos(2\pi u) + \mathbf{T}_{2,S} \sin(2\pi u) \\
 &= M_2 \sum_{k=0}^{M_1-1} \sum_{l=-\infty}^{\infty} \mathbf{c}[k, l] \phi_{1,\text{per}}(M_1 u - k) \dot{\phi}_2(M_2 - l)
 \end{aligned}$$

by evaluating the left-hand-side of (5.12) and (5.13). Here, $\mathbf{T}_{1,N}, \mathbf{T}_{2,N} \in \mathbb{R}^3$ represent two free vectors that determine the tangent plane at the north pole, and $\mathbf{T}_{1,S}, \mathbf{T}_{2,S} \in \mathbb{R}^3$ are two free vectors that determine the tangent plane at the south pole. The proposed smoothness conditions can be satisfied if and only if the model can reproduce sinusoids of unit period over u with a specified sampling rate. Since ϕ_1 was designed to reproduce sinusoids of unit period over u with a sampling rate of M_1 , we are guaranteed that smoothness conditions can be satisfied.

In Figure 5.3 we show some surfaces generated by (5.10) that fail to satisfy some of the conditions on the poles. If the pole interpolation conditions are not satisfied, the surface may not be completely closed; if the pole smoothness condition is not satisfied, the surface may have kinks at the poles, leaving an ill-defined tangent plane at these points.

The final step is to incorporate the exposed conditions into the parameterization and to obtain explicit formulas. We provide an explicit expression of (5.10) with a sphere topology and verify the pole interpolation and smoothness conditions.

Proposition 3. *A parametric spline-based surface with a sphere-like topology, $\mathcal{C}^1(\mathbb{R})$ continuity, and the capability of reproducing ellipsoids irrespective of their position and orientation can be expressed as*

$$\sigma(u, v) = \sum_{k=0}^{M_1-1} \sum_{l=-1}^{M_2+1} \mathbf{c}[k, l] \phi_{1,\text{per}}(M_1 u - k) \phi_2(M_2 v - l), \quad (5.14)$$

restricted to

$$\mathbf{c}[k, -1] = \mathbf{c}[k, 1] + \frac{\mathbf{T}_{1,N} c_{M_1}[k] + \mathbf{T}_{2,N} s_{M_1}[k]}{M_2 \dot{\phi}_2(1)} \quad (5.15)$$

$$\mathbf{c}[k, 0] = \frac{\mathbf{c}_N}{\phi_2(0)} + \frac{\phi_2(1)(\mathbf{c}[k, -1] + \mathbf{c}[k, 1])}{\phi_2(0)} \quad (5.16)$$

$$\mathbf{c}[k, M_2] = \frac{\mathbf{c}_S}{\phi_2(0)} - \frac{\phi_2(1)(\mathbf{c}[k, M_2 - 1] + \mathbf{c}[k, M_2 + 1])}{\phi_2(0)} \quad (5.17)$$

$$\mathbf{c}[k, M_2 + 1] = \mathbf{c}[k, M_2 - 1] - \frac{\mathbf{T}_{1,S} c_{M_1}[k] + \mathbf{T}_{2,S} s_{M_1}[k]}{M_2 \dot{\phi}_2(1)}, \quad (5.18)$$

where $\{\mathbf{c}[k, l]\}_{k \in [0 \dots M_1 - 1], l \in [1 \dots M_2 - 1]}$, \mathbf{c}_N , \mathbf{c}_S , $\mathbf{T}_{1,N}$, $\mathbf{T}_{2,N}$, $\mathbf{T}_{1,S}$, and $\mathbf{T}_{2,S}$ are free parameters. This adds up to a total number of $(M_1(M_2 - 1) + 4)$ free control points.

Proof. First, since the support of ϕ_2 is limited to the interval $[-\frac{3}{2}, \frac{3}{2}]$ and v lies within the interval $[0, 1]$, the second summation in (5.10) can be restricted to the indices $l \in [-1 \dots 1]$.

The restrictions (5.16) and (5.17) are obtained directly from the pole interpolation conditions. We rewrite the condition of the north pole as

$$\mathbf{c}[k, 0] = \frac{\mathbf{c}_N + \phi_2(1)(\mathbf{c}[k, -1] + \mathbf{c}[k, 1])}{\phi_2(0)}$$

for all $k \in [0 \dots M_1 - 1]$, where we have used the fact that $\phi_2(1) = \phi_2(-1)$ since ϕ_2 is symmetric, giving us (5.16). Likewise, the interpolation condition on the south pole is rearranged as

$$\mathbf{c}[k, M_2] = \frac{\mathbf{c}_S - \phi_2(1)(\mathbf{c}[k, M_2 - 1] + \mathbf{c}[k, M_2 + 1])}{\phi_2(0)}$$

for all $k \in [0 \dots M_1 - 1]$, giving us (5.17).

Now, we simplify the right-hand-side of the smoothness condition on the north pole to

$$\sum_{k=0}^{M_1-1} M_2 \left(\mathbf{c}[k, -1] \dot{\phi}_2(1) - \mathbf{c}[k, 1] \dot{\phi}_2(1) \right) \phi_{1,\text{per}}(M_1 u - k),$$

where we have used the fact that ϕ_2 is an antisymmetric function. We expand the sinusoids of the left-hand-side of the smoothness condition of the north pole using $\phi_{1,\text{per}}$. It results in

$$\begin{aligned} & \mathbf{T}_{1,N} \cos(2\pi u) + \mathbf{T}_{2,N} \sin(2\pi u) \\ &= \mathbf{T}_{1,N} \sum_{k=0}^{M_1-1} c_{M_1}[k] \phi_{1,\text{per}}(M_1 u - k) + \mathbf{T}_{2,N} \sum_{k=0}^{M_1-1} s_{M_1}[k] \phi_{1,\text{per}}(M_1 u - k) \\ &= \sum_{k=0}^{M_1-1} (\mathbf{T}_{1,N} c_{M_1}[k] + \mathbf{T}_{2,N} s_{M_1}[k]) \phi_{1,\text{per}}(M_1 u - k). \end{aligned}$$

By identification of the coefficients, we obtain

$$\mathbf{c}[k, -1] = \mathbf{c}[k, 1] + \frac{\mathbf{T}_{1,N} c_{M_1}[k] + \mathbf{T}_{2,N} s_{M_1}[k]}{M_2 \phi_2(1)}$$

for all $k \in [0 \dots M_1 - 1]$, which gives (5.15). The expression for (5.18) is obtained analogously using the pole smoothness condition in the south pole. \square

5.2 Energies and Implementation

5.2.1 Snake Energy

As exposed in Chapter 2, the snake evolution is driven by a application-dependent energy function. Moreover, the quality of the segmentation depends on the choice of the energy term. In our implementation, we obviated the constraint energy since we accommodated the user interaction as a hard constraint.

5.2.1.1 Image Energy There are many construction strategies to design the image energy in 3D snakes. They are usually an extension of their 2D counterparts. As shown in Section 4.3.1 in the context of 2D snakes, image energies can be categorized in two main families: 1) edge-based schemes [43, 47, 50] and 2) region-based methods [49, 68]. Both families have their own advantages and disadvantages. The first one gives a better localization of the contour near the boundaries of the object to segment at expense of a narrower basin of attraction. We follow the same strategy than in Section 4.3.1 by using a convex combination of gradient and region energies

$$E_{\text{image}} = \alpha E_{\text{edge}} + (1 - \alpha) E_{\text{region}},$$

where $\alpha \in [0, 1]$. The tradeoff parameter α balances the contribution of the edge-based energy and the region-based energy. Its value depends on the characteristics of each application.

For the gradient-based (or edge) energy, traditional snakes rely on edge maps to guide them. The most popular approach is based on the magnitude $\|\nabla f\|$ of the gradient [43]. Mathematically, they minimize

$$E_{\text{mag}} = - \iint_S \|\nabla f\| \, dS,$$

where the integration is performed on the closed surface S , and dS represents the differential of area. The major drawback of this approach is that the snake gets distracted by edges of nearby targets, since it does not distinguish between gradients generated by different objects in the image.

We solve this issue by considering the direction of the gradient as well and by imposing that the direction of the gradient and the normals of the surface be aligned. The parameterization (5.9) offers us the opportunity to choose the orientation of the normal vectors at initialization time. From now on, and without loss of generality, we assume it to be outwards. If we want to segment a bright object surrounded by a darker region, we expect the directions of the image gradients to be the same as the directions of the normal vectors when the energy reaches its minimum. We formalize this concept with the following image energy:

$$E_{\text{grad}} = - \iint_S \nabla f \cdot \mathbf{dS} = - \iint_S \left(\nabla f \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) \, dS, \quad (5.19)$$

where \mathbf{dS} represents the vector differential of area.

The advantage of this energy is that it uses the direction of the gradient to discriminate between edges of the same target and between different targets. Its minimization makes the surface of the snake stick to edges where the image gradient has similar direction as the normal vector \mathbf{n} and to be repelled from edges with different orientations.

In Figure 5.4, we present a 2D schematic with the configuration of the various quantities involved in the computation of (5.19) in two different setups. The first one corresponds to a snake represented by the dashed line S_1 , segmenting the gray circle (representing high pixel values) labeled as Ω_1 . The normal to the surface \mathbf{n} and the image gradient ∇f are pointing in the same direction, which will add a strong contribution to E_{grad} . The second case corresponds to a snake represented by the dashed line S_2 segmenting the same gray circle, but this time the dashed line is closer to the region labeled as Ω_2 . In this case, the normal to the surface \mathbf{n} and the image gradient ∇f are pointing in opposite directions, which adds a strong negative contribution to E_{grad} .

For the region-based energy, we adopt a strategy similar to the one we followed in the 2D case. We first build an enclosing shell S_{sh} around the snake. Then,

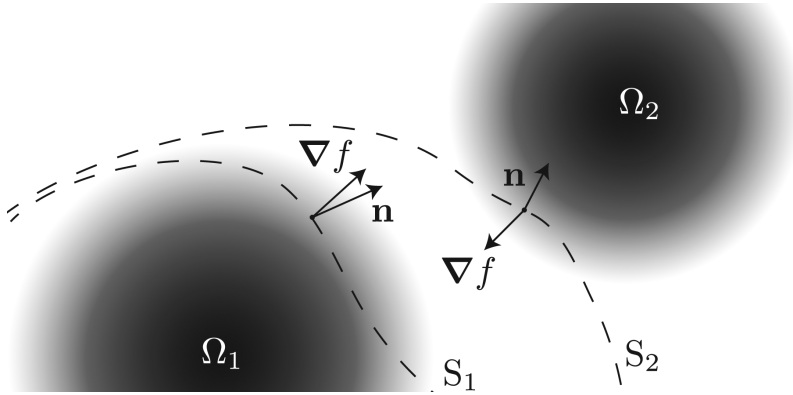


Figure 5.4: 2D schematic representation of two parametric snakes S_1 and S_2 (dashed lines), of their interaction with the objects Ω_1 and Ω_2 constituted by gray circles (representing high pixel values), of normal vectors \mathbf{n} , and of the gradient vectors ∇f of the image.

our region-based energy discriminates an object from its background by and maximizing the contrast between the intensity of the data averaged within the volume V enclosed by the snake, and the intensity of the data averaged within the volume V_{sh} enclosed by the shell. When $V \subset V_{\text{sh}}$, the region energy is expressed as

$$E_{\text{region}} = \frac{1}{|V|} \left(\iiint_V f \, dV - \iiint_{V_{\text{sh}} \setminus V} f \, dV \right), \quad (5.20)$$

where $|V|$ is the volume of the snake.

To enforce that the criterion remains neutral when f takes a constant value f_0 , we build the enclosing shell such that $|V_{\text{sh}}| = 2|V|$. Under these conditions, $E_{\text{region}}|_{f=f_0} = 0$ depends neither on the snake nor even on f_0 . We take full advantage of the affine invariance property of our snake model; we build the parameterization of the shell σ_{sh} as the affine transformation of the snake surface (5.14) with the same center of gravity and with volume $|V_{\text{sh}}|$. The explicit parametric expression is

$$\sigma_{\text{sh}}(u, v) = \sqrt[3]{2} \sigma(u, v) + \left(1 - \sqrt[3]{2}\right) \sigma_{\text{g}},$$

where

$$\sigma_{\text{g}} = \sum_{k=0}^{M_1-1} \sum_{l=-1}^{M_2+1} \mathbf{c}[k, l]$$

is the center of gravity of the snake. Since the shell is an affine transformation of the original snake, it is possible to express σ_{sh} with the same basis functions.

The shell σ_{sh} is fully determined by the sequence of control points. Thus, any modification on σ results in a readjustment of σ_{sh} .

5.2.1.2 Internal Energy The internal energy is responsible for ensuring the smoothness of the snake. It was first proposed in [43] in the context of 2D active contours as a linear combination of the length of the contour and the integral of the square of the curvature along the contour. This smoothness term is the one that is most widely-used in applications. Its direct extension to active surfaces gives

$$E_{\text{int}} = \lambda_1 |S| + \lambda_2 \iint_S |K|^2 dS, \quad (5.21)$$

where K is the Gaussian curvature of the surface.

This internal energy can also be expressed specifically in terms of the tangent vectors and the fundamental forms as

$$E_{\text{int}} = \lambda_1 \int_0^1 \int_0^1 \|\mathbf{T}_1 \times \mathbf{T}_2\| du dv + \lambda_2 \int_0^1 \int_0^1 \left| \frac{\det \mathbf{II}}{\det \mathbf{I}} \right|^2 \|\mathbf{T}_1 \times \mathbf{T}_2\| du dv,$$

where the first term makes the snake contract and the second favors smooth solutions. In the framework of active contours, most parametric schemes rely on the smoothness of the representation, thus eliminating the need for an explicit internal energy term. However, these approaches can ensure a low value of the curvature only when the curves are parameterized at constant speed (proportional to arc-length). For example, a spline curve may be rough if some of the spline knots accumulate at one section of the curve. Similar problems exist with Fourier and other parametric representations. A practical workaround is to reparameterize the curve to constant arc-length after each step of the optimization algorithm, which is quite expensive [143]. Another approach is to substitute the curvature term of (5.21) by an energy term that penalizes the curve for not being in the curvilinear abscissa [57]. This energy is called *curvilinear reparameterization energy*. Minimizing this energy causes the curve knots to move tangentially to the snake, thus bringing it to curvilinear abscissa. The use of this energy yields the same results as reparameterizing the snake at each step, but with a much lower computational load. We adopt a similar approach and modify the internal energy (5.21) to

$$E_{\text{int}} = \lambda_1 |S| + \lambda_2 \int_0^1 \int_0^1 (|\det \mathbf{I}| - |S|)^2 du dv. \quad (5.22)$$

Evolving the curve with such an energy term will cause the control points to distribute uniformly over the snake surface and avoids accumulation.

5.2.2 Fast Energy Computation

The computational cost of the evolution of the snake is dominated by that of the volume integrals in (5.20). An efficient way to implement these operations is the use of pre-integrated images. Let g be the function we are integrating, and let Ω be the domain of integration (V or V_{sh}). Then, by Gauss' theorem, we rewrite the volume integral as the surface integral in different ways

$$\begin{aligned} \iiint_{\Omega} g \, dV &= \oint_{\partial\Omega} g_1 \, dx_2 \wedge dx_3 \\ &= \oint_{\partial\Omega} g_2 \, dx_3 \wedge dx_1 \\ &= \oint_{\partial\Omega} g_3 \, dx_1 \wedge dx_2, \end{aligned}$$

where \wedge is the wedge product, ∂V is the boundary of V , and

$$\begin{aligned} g_1(x_1, x_2, x_3) &= \int_{-\infty}^{x_1} g(\tau, x_2, x_3) \, d\tau \\ g_2(x_1, x_2, x_3) &= \int_{-\infty}^{x_2} g(x_1, \tau, x_3) \, d\tau \\ g_3(x_1, x_2, x_3) &= \int_{-\infty}^{x_3} g(x_1, x_2, \tau) \, d\tau. \end{aligned}$$

All three possibilities are equivalent and can be stored in lookup tables to speed up the access to the data. The translation of volume integrals into surface integrals reduces the computational load dramatically. This can only be achieved if the surface is defined continuously, as in (5.1).

In the interest of space, we show the exact expressions of the energies using pre-integrated images in Appendix 5.A.

5.2.3 Sampling

Despite the fact that we are describing our surface continuously, in a real-world implementation we only have at our disposal a sampled version of the functions we want to pre-integrate. We therefore perform a trilinear interpolation of the sampled data and store the result in lookup tables.

5.2.4 Optimization

As mentioned before, the snake extracts the boundary of an object by finding the minimum of the energy functional. For that purpose, we iteratively update the value of the free control points $\{\mathbf{c}[k, l]\}_{k \in [0 \dots M_1 - 1], l \in [1 \dots M_2 - 1]}$, \mathbf{c}_N , \mathbf{c}_S , and the free control vectors $\mathbf{T}_{1,N}$, $\mathbf{T}_{2,N}$, $\mathbf{T}_{1,S}$, and $\mathbf{T}_{2,S}$ using a generic unconstrained gradient-based optimizer. The optimization scheme is efficiently carried out by a Powell-like line-search method [144]. This method requires the derivatives of the energy function with respect to the parameters, and converges quadratically to the solution. The algorithm proceeds as follows: firstly, one direction within the parameter space is chosen depending on the partial derivatives of the energy. Secondly, a one-dimensional minimization is performed within the selected direction. Finally, a new direction is chosen using the partial derivatives of the energy function once more, while enforcing conjugation properties. This scheme is repeated until convergence. This powerful optimization method is used here because the number of parameters that define the shape of the snake is very small. Assuming a trilinear interpolation of the original function f , we were able to derive exact and closed expressions for the energy functions. In the interest of space, we show the derivation of these expressions in Appendix 5.A.

5.2.5 Self-Intersection Detection

The optimization process can sometimes lead to self-intersecting surfaces. However, the probability of self-intersection is greatly reduced by the introduction of the new internal energy (5.22). Without this term, the control points can bunch together, eventually resulting in self-intersection.

Despite this refinement, self-intersection may still arise occasionally when the image energy forces some control points to move faster than others. This compromises our approach since we use Gauss' theorem, which assumes non self-intersecting closed surfaces. An extensive body of research can be found on the intersection problem, with numerous articles presenting different approaches for the intersection of freeform curves and surfaces [156]. Unfortunately, these methods are excessively time-consuming for our purpose.

As an alternative, we devised a fast method for self-intersection detection using the Gauss-Bonnet formula. This formula states that the Euler characteristic χ of a closed, non-intersecting surface S can be computed by integrating the Gaussian curvature as follows:

$$\chi(S) = \frac{1}{2\pi} \iint_S |K|^2 dS. \quad (5.23)$$

The Euler characteristic is a number that describes a topological space's shape or structure regardless of the way it is bent or deformed, as long as it does not self-

intersect. In the case of sphere-like topologies, we have $\chi(S) = 2$. Therefore, we know that the snake self-intersects if $\chi(S) \neq 2$.

Note that our criterion can give a correct value χ even if the surface is self-intersecting, which implies that it is not completely foolproof. In principle, it is possible to detect these cases by splitting the integral (5.23) over a series of smaller intervals and checking if there is a self-intersection in each of the subintervals. However, such cases are unlikely to occur in practice, and it was not necessary to implement such a finer level of detection.

5.2.6 User Interaction

By contrast with many other implicit and global parametric snakes, our snake is fully parameterized by only a few control points, which eases the interactions with the user. This has encouraged us to develop a dedicated graphical user interface that lets the user initialize the 3D snake position and refine it even after the optimization process. The user can intuitively manipulate the position of any point by selecting it in either the 2D or 3D mesh representations of the snake. The 2D view also provides 3D cues, as we color the mesh depending on the depth and set its transparency proportional to the distance of the slice displayed in the 2D viewer. With this strategy, only parts of the snake which are close to the displayed image are revealed. Editing the points is performed with simple mouse actions. The live update of the 2D and 3D views is very fast. This is due to the fact that the change in one control point affects the structure locally. Thus, just a limited region of the snake's surface has to be recomputed. As a result, a fast, user-friendly, and semi-automatic segmentation procedure, that loops between snake initialization, optimization, and correction, is made possible.

5.2.7 Initialization

Our method can be operated in an automated fashion. For that, the algorithm only requires an initial position. Like in all segmentation problems, this initialization is strongly dependent on the imaging modality. For example, in fluorescence microscopy, a blurring or a DoG filter followed by the detection of local maxima/minima suffices to provide an initial position (see Chapter 4 for examples of automatic initializations of 2D parametric snakes). In medical imaging, refined methods based on atlases can also be used [157].

When choosing the initial position of the snake, one must ensure that the object of interest is within the basin of attraction of the chosen image energy. The conditions for the convergence of our image energies are the following:

- E_{grad} is only responsive to image gradients. Therefore, the snake surface must be initialized close to the boundary of the object. This basin of attraction is

usually extended by preprocessing the input image with a smoothing or a distance transform.

- E_{region} is responsive to image contrasts between the snake core and the snake shell. Therefore, the snake surface must be initialized such that the core intersects the object and the shell intersects the background.

5.3 Experiments and Simulations

We present in this section five experiments. In the first one, we deform our snake to show how the self-intersection is detected by monitoring the Euler characteristic. In the second experiment, we investigate the sensitivity of our snake to the presence of noise. In the third setup, we perform a quantitative evaluation of our algorithm when segmenting neighboring targets; there the preservation of topology plays a crucial role. We move away from simulated data in the fourth experiment where we investigate the approximation properties of our snake with medical data of a spleen from a CT-scan. In the last experiment, we illustrate the application to real microscopic data where the ground truth is not available.

The calculations were performed on a MacPro 3.1 with two Quad-Core Intel Xeon processors, 12GB of RAM memory, and an NVIDIA GeForce 8800 GT with 512 MB running Mac OS X 10.8.2

5.3.1 Twisting the Snake

We deformed the snake away from a perfect sphere by rotating the central layer of control points by an angle θ , thereby creating a twist in the center of the structure. In particular, we set the snake control points to

$$\mathbf{c}[k, l] = \begin{pmatrix} c_{M_1}[k] s_{2M_2}[l] \\ s_{M_1}[k] s_{2M_2}[l] \\ c_{2M_2}[l] \end{pmatrix},$$

for $k \in [0 \dots M_1 - 1]$ and for all $l \neq \lfloor \frac{M_2}{2} \rfloor$, and we set

$$\mathbf{c}[k, l] = \begin{pmatrix} c_{M_1}[k] s_{2M_2}[l] \cos \theta - s_{M_1}[k] s_{2M_2}[l] \sin \theta \\ c_{M_1}[k] s_{2M_2}[l] \sin \theta + s_{M_1}[k] s_{2M_2}[l] \cos \theta \\ c_{2M_2}[l] \end{pmatrix},$$

for $k \in [0 \dots M_1 - 1]$ and for $l = \lfloor \frac{M_2}{2} \rfloor$.

We show in Figure 5.5 different surface configurations as a function of the rotation angle θ . We see that for small angles, the surface is slightly twisted but not self-intersecting. However, as we approach $\theta = \pi$, the central part of the

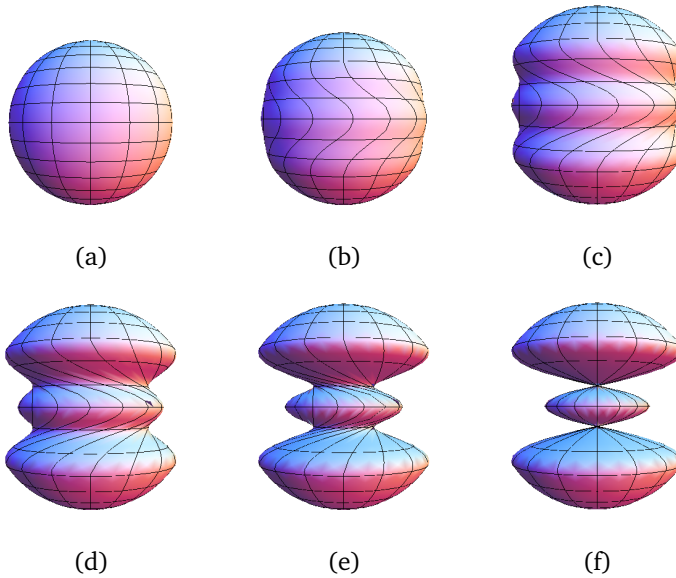


Figure 5.5: Deformation of our snake from a perfect sphere by rotating the central layer of control points by an angle θ , thereby creating a twist in the center of the structure: (a) $\theta = 0$; (b) $\theta = \frac{\pi}{5}$; (c) $\theta = \frac{2\pi}{5}$; (d) $\theta = \frac{3\pi}{5}$; (e) $\theta = \frac{4\pi}{5}$; (f) $\theta = \pi$.

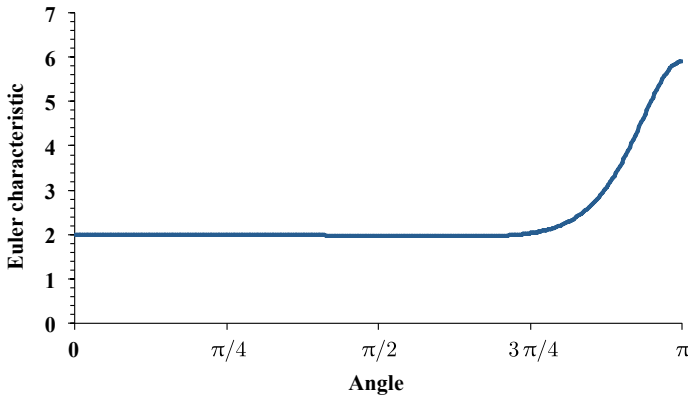


Figure 5.6: Value of the Euler characteristic $\chi(S)$ of the snake surface, computed using the Gauss-Bonnet formula when applying a rotation on the central layer of control points. As the layer rotates, the structure deforms and loses its sphere-like topology.

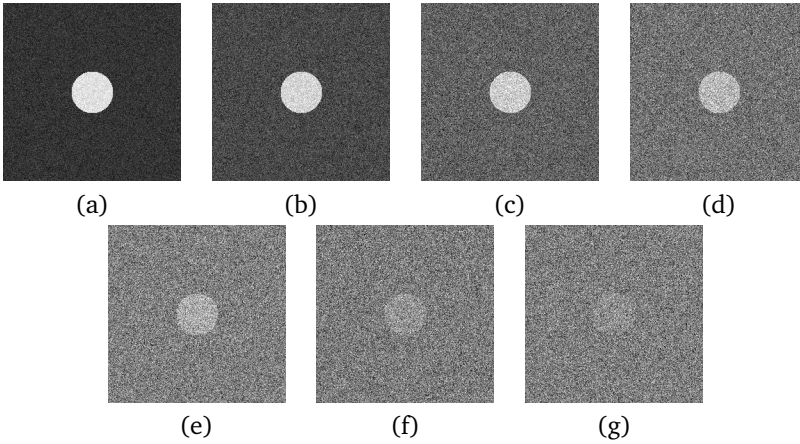


Figure 5.7: Cross-sections of the test images used in the assessment of robustness. Only one realization is shown for each PSNR value. (a) PSNR = 20dB. (b) PSNR = 15dB. (c) PSNR = 10dB. (d) PSNR = 5dB. (e) PSNR = 0dB. (f) PSNR = -5dB. (g) PSNR = -10dB.

structure collapses in two points, producing two self-intersections. We show in Figure 5.6 how the Euler characteristic, computed by the use of (5.23), varies in terms of θ . The predicted value is correct for $\theta \leq \frac{7\pi}{9}$ and starts increasing when the snake starts self-intersecting. Hence the conditions of the Gauss-Bonnet theorem are violated.

5.3.2 Robustness to Noise

Next, we investigate the sensitivity of our method to the presence of noise. We generated 100 realizations of a noisy sphere for each one of seven different peak-signal-to-noise ratios (PSNRs). Our test images were obtained by on-voxel-wise sampling a sphere of radius of 30 voxel units on a regular grid of $(256 \times 256 \times 256)$ voxels. We show in Figure 5.7 one cross-section of the noisy volumes for every PSNR value.

The initial shape is a sphere with a fixed radius of 50 voxel centered at a distance of 25 voxel from the real center of the object to segment. We constrained the number of control points to its minimum ($M_1 = M_2 = 3$). Then, we ran the optimization process until convergence using exclusively the region-based energy (*i.e.*, we set the tradeoff parameter $\alpha = 0$).

We used the Jaccard distance $J = 1 - |\Theta \cap \Omega| / |\Theta \cup \Omega|$ to quantify, as a percentage, the quality of the segmentation. This distance provides a measure of

PSNR [dB]	J [%]	σ_J [%]
20	0.20	0.16
15	0.44	0.26
10	0.61	0.35
5	2.16	2.68
0	12.16	10.05
-5	22.74	11.29
-10	34.69	9.82

Table 5.1: Jaccard distance when segmenting noisy spheres.

dissimilarity between two binary objects where a low value reflects an accurate segmentation. In the definition of the Jaccard distance, Θ corresponds to the ground-truth region and Ω corresponds to the region enclosed by the snake. We computed J following a on-voxel discretization of the data.

We show in Table 5.1 the value of J and its standard deviation σ_J across all noisy realizations. We observe from the results that our snake is robust against noise since it is capable of giving a proper segmentation even for low PSNRs. The quality of the segmentation deteriorates quickly when $\text{PSNR} \geq 0$ due to the presence of too much noise.

5.3.3 Segmentation of Overlapping Objects

In this section, we compare our snake against other segmentation methods in terms of accuracy and speed at the task of delineating different configurations of overlapping objects.

We generated 4 volumetric images ($256 \times 256 \times 256$) by voxel-wise sampling the union of two spheres of radius 50 pixel. We show a rendering of these shapes in Figure 5.8. They are parameterized with the distance d , in pixel units, between the centers of the spheres. For $d < 100$, the spheres intersect; for $d = 100$, the spheres share one single pixel; for $d > 100$, the spheres are disjoint. The grayscale values of the images are 255 for the shape and 0 for the background. We are interested in isolating each sphere. Without loss of generality, we focus on segmenting one of them.

We compared our snake to a traditional level-set method based on the formulation of Chan-and-Vese [70], and to the 3D active meshes of [148]. The implementation of the level-set method was taken from the free open-source image-processing package Fiji¹ implementing the algorithm described in [62]. The im-

1. <http://fiji.sc/>

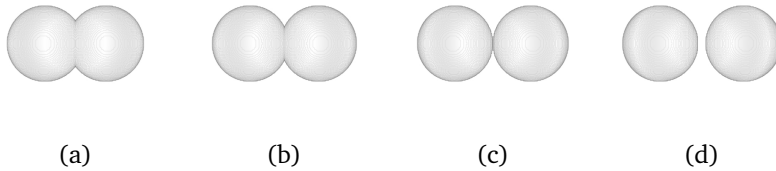


Figure 5.8: Renderings of the test images used in the analysis of performance when segmenting overlapping objects. (a) $d = 80$. (b) $d = 90$. (c) $d = 100$. (d) $d = 110$.

plementation of the active-meshes method was taken from the free open-source image-processing package Icy.²

We initialized the level-set method by providing a point seed (the only possible initialization afforded by the Fiji plug-in). The initial positions were determined by a detector of local maxima applied over a version of the image that was smoothed with a Gaussian kernel of with $\sigma = 10$. A total number of 2 local maxima were detected in all images. We discarded the rightmost detection since we are interested in segmenting the leftmost object. We initialized our snake as a perfect sphere of radius 60 pixel units. Finally, the active meshes cannot take the form of an ideal sphere but can approximate it. We initialized this method using the automatic tessellation of the sphere of radius 60 voxels.

We chose $M_1 = M_2 = 3$, which are low values that favor ellipsoid-like shapes during the segmentation process. Then, we ran the optimization process until convergence using the edge-based energy (*i.e.*, we set the tradeoff parameter $\alpha = 1$) and our internal energy with $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$. We executed the level-set with an advection value of 220 and a propagation value of 100. These values were chosen to accelerate the propagation of the evolving level-set front and to obtain a faster convergence without losing accuracy. For the active meshes, we set the time-evolution step to 0.01, the window size to 100, and we evolved the snake using the gradient criterion with weight 0.5 and regularization weight 0.01.

We show in Table 5.2 a comparison across all mentioned methods of the Jaccard distance J reached at the end of the optimization process as well as the time it took the algorithms to converge for the different test datasets. The times shown in Table 5.2 exclude the preprocessing stages of the three methods.

Clearly, the level-set method is the slowest. This is in agreement with the results shown in Chapter 4 for the 2D case. Meanwhile, our snake and the active meshes

2. <http://icy.bioimageanalysis.org/>

Method	J [%]	Time [s]	Segmented shape
$d = 80$			
Spline Snake	3.06	0.93	Left sphere
Level-Set	48.66	2808.00	Merged spheres
Active Meshes	48.91	6.32	Merged spheres
$d = 90$			
Spline Snake	2.61	0.91	Left sphere
Level-Set	49.69	2862.03	Merged spheres
Active Meshes	50.11	6.01	Merged spheres
$d = 100$			
Spline Snake	0.83	0.93	Left sphere
Level-Set	50.00	2889.13	Merged spheres
Active Meshes	2.79	3.96	Left sphere
$d = 110$			
Spline Snake	0.71	0.93	Left sphere
Level-Set	0.25	1412.09	Left sphere
Active Meshes	1.98	4.44	Left sphere

Table 5.2: Accuracy and efficiency of the mentioned segmentation algorithms when segmenting overlapping objects.

demonstrate a similar level of performance in terms of speed and accuracy, even though the active-meshes method takes advantage of the GPU present in the hardware, while our snake and the level-set method do not.

We can also see from Table 5.2 that the level-set method extracts a merged version of the two spheres as long as there exists a single pixel that connects them. This is due to the fact that the level-set method does not have any constraint on the topology and can leak through holes. On the other hand, the active meshes and our snake succeed in segmenting the left sphere alone even in the presence of some overlap, being the proposed method the most robust and the fastest.

5.3.4 Approximation of Shapes

In this section, we move away from numerical simulations, and we investigate the capabilities of our snake when approximating realistic shapes as a function of M_1 and M_2 . We quantify its accuracy at outlining the wall of a spleen within slices of a 3D CT-scan image sequence.

The data we used are part of the 3D-IRCAdB (3D image reconstruction for comparison of algorithm database). It includes several sets of medical images of patients and the manual segmentation of the various structures of interest, per-

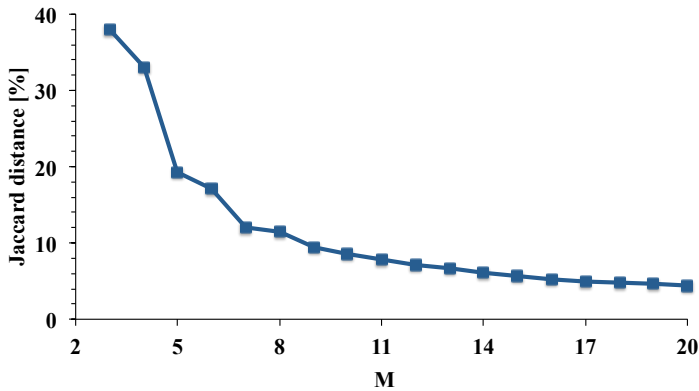


Figure 5.9: Evolution of the Jaccard distance as a function of $M = M_1 = M_2$ when approximating a spleen.

formed by clinical experts.³ For every patient under analysis, the ground truth is available as a triangular mesh where the vertex locations correspond to pixel positions. Moreover, the database provides a 3D voxel mask with the interior of the mesh. In the case of the mask, the volume consists of 166 slices with a spacing between slices of 1.8 mm. Each slice is a (512×512) image with a pixel spacing of 0.961 mm.

To approximate the spleen with our snake, we first detect the boundary pixels of the spleen mask for each slice. Then, for each slice, we fit a spline corresponding to a circle of latitude of σ . Using this approach we obtain a snake σ aligned in the vertical direction. The north pole is located at the apex of the spleen, and the south pole is located at its bottom. The circles of latitude are adapted to the shape of the spleen in the XY plane. Then, we refine the final fit with a global 3D optimization led by the edge-based energy (*i.e.*, we set the tradeoff parameter $\alpha = 1$), and our internal energy with $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$.

In the simulations of Figure 5.9, we investigated the dependence of the Jaccard distance J on $M_1 = M_2 = M$. Our results show that the error decreases quadratically, which demonstrates the ability of the proposed model to segment objects with an ellipsoid-like topology. In Figure 5.10, we show the voxelized mask we used as ground truth and the successive approximations of our snake for different values of M . We see that, for small values M (such as $M = 3$), the snake takes an almost ellipsoidal shape and is not capable of capturing every detail of the spleen structure. As we increase the number of control points, the snake captures the structure of the organ while providing a smooth surface.

3. <http://www.ircad.fr/software/3Dircadb/3Dircadb.php>

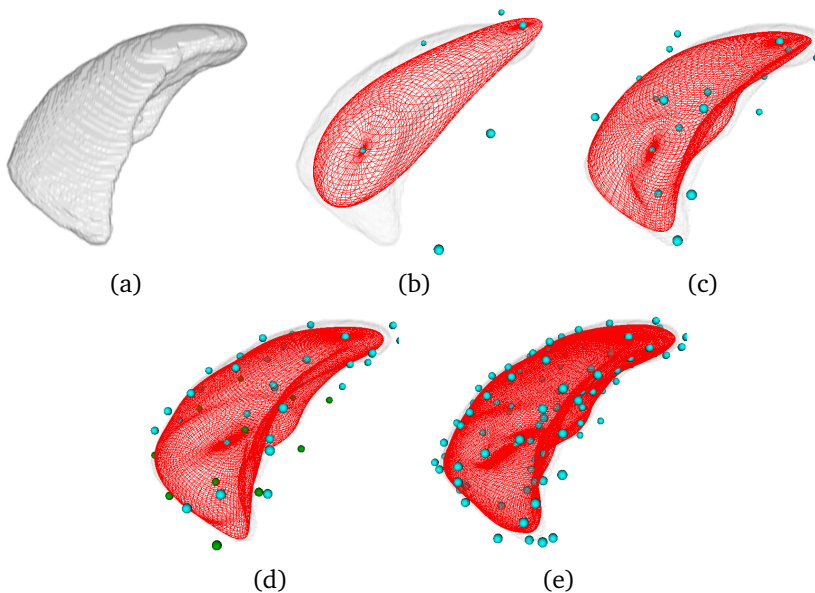


Figure 5.10: Approximation of a 3D CT-scan spleen for different values of $M = M_1 = M_2$. (a) Rendering of the spleen. (b) Approximation with $M = 3$. (c) Approximation with $M = 5$. (d) Approximation with $M = 7$. (e) Approximation with $M = 10$.

5.3.5 Segmentation of 3D Confocal Microscopic Images

We finally illustrate the behavior of our snake and provide further insights into its capabilities in real-world applications. In this section, the ground truth is missing, so we must relinquish quantitative assessments in favor of qualitative ones. Here, we initialized our snake manually using the interaction capabilities of our software.

5.3.5.1 Cell-Body Segmentation We processed a stack ($576 \times 504 \times 200$) of confocal ($\times 60$ magnification) images from the brain cortex of a rat, with YFP labeling for the neurons and GFP for the microglia.⁴ The challenge was to segment the body of the neuronal cells, despite their non-spherical shape, the lack of clear borders, and the presence of several surrounding objects. We set $M_1 = M_2 = 7$, roughly initializing the snake position around each cell body as a sphere, and ran the optimization process until convergence using exclusively the edge-based energy (*i.e.*, we set the tradeoff parameter $\alpha = 1$). One example of the resulting segmentation is shown in Figure 5.11. We observe that the snake was

4. <http://www.cellimagelibrary.org/images/27155/>

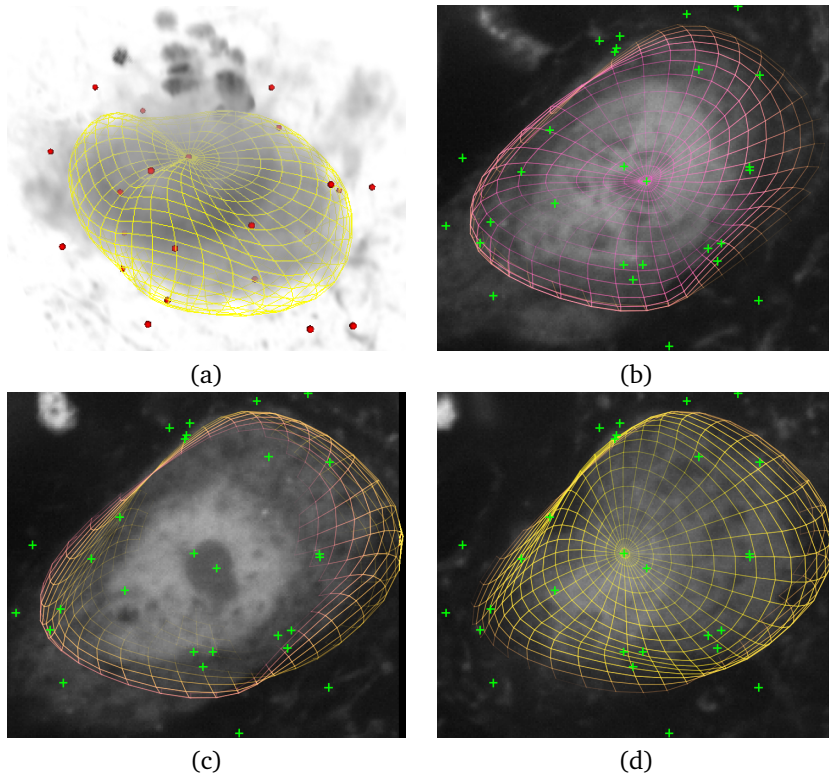


Figure 5.11: Segmentation of a cell in a 3D confocal image of a murine brain (courtesy of Sabine Scheibe and Sebastian Rhode at TILL Photonics). (a) 3D view of the segmented brain cell with the full snake mesh overlaid. From (b) to (d): 2D views of different slices of the dataset at different x_3 -axis positions. The snake mesh changes color and transparency depending on the position of the grid points to the displayed slice.

able to adapt well to the 3D cell shape: the surface is accurately fitted despite the limited degrees of freedom of the model, while irregularities are properly smoothed out. The optimization process took only 0.74 s, which is faster than the duration of the acquisition of such data (usually, much longer than 1 s).

5.3.5.2 Segmentation of Glomeruli We investigated the segmentation of olfactory glomeruli in the mouse brain which represent neuroglial functional units in olfactory information processing [158]. With Topro staining, glomeruli correspond to the dark areas delimited by fluorescent cell bodies. In Figure 5.12, they are visualized as bright areas with an inverted look-up-table. However,

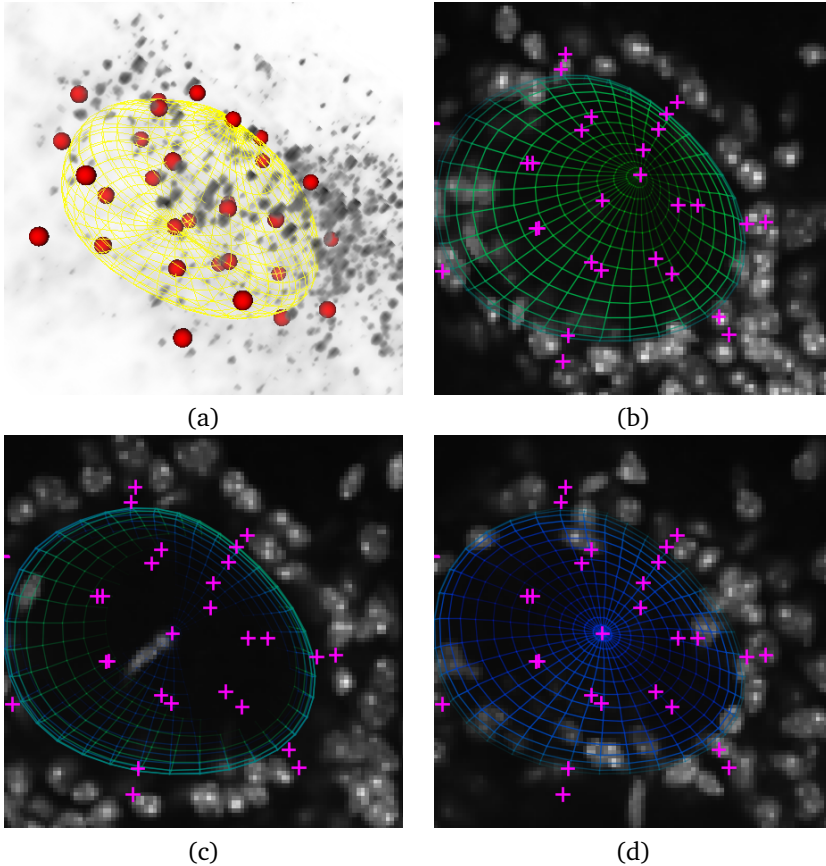


Figure 5.12: Segmentation of an olfactory glomerulus in a 3D confocal image stack ($256 \times 256 \times 67$) of a mouse brain (image courtesy of Lisa Roux at the Collège de France). (a) 3D view of the segmented glomeruli with the full snake mesh overlaid. From (b) to (d): 2D views of different slices of the dataset (inverted colors) at different x_3 -axis positions ($x_3 = 10, 17$ and $24 \mu\text{m}$). The snake mesh changes color and transparency depending on the position of the grid points to the displayed slice.

the surrounding fluorescent cells do not form continuous boundaries. This penalizes nonparametric active contours, as the snake may ‘leak out’ between the neighboring cells. By contrast, we show in Figure 5.12 that the proposed snake method (with $M_1 = M_2 = 7$) is able to accurately identify the glomerulus border. This is a key advantage of the proposed parametric model, as it allows one to exert *a priori* control over the regularity and topology of the snake. Here again, the optimization process was performed using the edge-based energy exclusively (*i.e.*, $\alpha = 1$). Convergence was reached after 1.74 s, which is remarkably fast.

5.4 Conclusions

Our contribution in this chapter is a new fully parametric snake with a sphere-like topology. It is constructed using the basis functions we investigated in Chapter 3, and it is therefore capable of reproducing any ellipsoid, irrespective of its position or orientation. Our snake is characterized by fewer control points than nonparametric snakes and can approximate any blob-like structure with arbitrary precision. The modification of one control point affects a limited region of the snake surface, which favors intuitive interactions with the user. Since our shape model is fully characterized by few control points, the design of customized shapes becomes possible by simple manipulation of these points in the same way that control points are used in the NURBS meshes typical of computer-aided industrial designs [55]. Moreover, the control points may be used to perform statistical learning/analysis of the segmented objects [159].

We designed an edge-based energy that is capable of maintaining the consistency of the segmentation in the presence of clutter. This is accomplished by penalizing mismatches in the directions of the image gradients. Furthermore, we combined it with a robust region-based energy. We were able to accelerate the implementation by taking advantage of Gauss’ theorem, which was facilitated by the availability of explicit expressions of our bases. Moreover, we introduced a novel technique to detect self-intersection in order to know when the snake loses the sphere-like topology. We have applied our snakes to a variety of problems that involve synthetic simulations and challenging real datasets, where the object contours were not well defined. As a result, various experiments have shown that the proposed 3D snake can approximate blob-like objects with good accuracy. Moreover, the optimization process is remarkably fast as we have designed our bases to have the shortest-possible support.

Appendices

5.A Implementation Details

In Section 5.2, we provided the guidelines for an efficient implementation of our energy functionals. Here, we derive the explicit expressions of our image energies. These expressions are needed when implementing the snake optimization routine. We approximated the partial derivatives of the energy functionals using centered finite differences. However, it is possible to obtain analytic close-form expressions following the same strategy as in Appendix 4.A

5.A.1 Image Energy

As described in Section 5.2.1.1, our image energy is composed of two terms: a contour (or edge) term and a region term.

5.A.1.1 Contour Image Energy By Gauss' (or divergence) theorem, our contour energy (5.19) can also be expressed as the volume integral

$$E_{\text{grad}} = - \iiint_V \text{div}(\nabla f) \, dV = - \iiint_V \Delta f \, dV,$$

where V is the volume enclosed by S , dV represents the differential of volume, and Δf is the Laplacian of the image f . We express the volume integral of $g = -\Delta f$ over the region V enclosed by S as

$$E_{\text{grad}} = \oiint_S g_1 \, dx_2 \wedge dx_3 = \int_0^1 \int_0^1 g_1(\boldsymbol{\sigma}) \left(\frac{\partial x_2}{\partial u} \frac{\partial x_3}{\partial v} - \frac{\partial x_2}{\partial v} \frac{\partial x_3}{\partial u} \right) \, du \, dv, \quad (5.24)$$

where g_1 is the pre-integrated image along the first dimension. Now, by the explicit parametric description of $\boldsymbol{\sigma}$, we have

$$\frac{\partial \boldsymbol{\sigma}}{\partial u}(u, v) = M_1 \sum_{k=0}^{M_1-1} \sum_{l=-1}^{M_2+1} \mathbf{c}[k, l] \dot{\phi}_{1,\text{per}}(M_1 u - k) \phi_2(M_2 v - l) \quad (5.25)$$

$$\frac{\partial \boldsymbol{\sigma}}{\partial v}(u, v) = M_2 \sum_{k=0}^{M_1-1} \sum_{l=-1}^{M_2+1} \mathbf{c}[k, l] \phi_{1,\text{per}}(M_1 u - k) \dot{\phi}_2(M_2 v - l) \quad (5.26)$$

where $\frac{\partial \boldsymbol{\sigma}}{\partial u} = \left(\frac{\partial x_1}{\partial u}, \frac{\partial x_2}{\partial u}, \frac{\partial x_3}{\partial u} \right)$ and $\frac{\partial \boldsymbol{\sigma}}{\partial v} = \left(\frac{\partial x_1}{\partial v}, \frac{\partial x_2}{\partial v}, \frac{\partial x_3}{\partial v} \right)$. Finally, we approximate the double integral (5.24) by a double sum by sampling u and v at a fixed sampling rate. Since the basis functions and their derivatives are compactly-supported, the number of non-zero elements in the sum is small. Moreover, we precompute their samples and store them in lookup tables .

5.A.1.2 Region Image Energy Our region energy (5.20) can be expressed as

$$E_{\text{region}} = \frac{1}{|V|} \left(2 \iiint_V f \, dV - \iiint_{V_{\text{sh}}} f \, dV \right),$$

as long as $V \subset V_{\text{sh}}$. Then, computing the image energy reduces to the evaluation of two volume integrals over the regions delimited by σ and σ_{sh} (i.e., V and V_{sh} respectively).

We express the volume integral of f over V as

$$\iiint_S f_1 \, dx_2 \wedge dx_3 = \int_0^1 \int_0^1 f_1(\sigma) \left(\frac{\partial x_2}{\partial u} \frac{\partial x_3}{\partial v} - \frac{\partial x_2}{\partial v} \frac{\partial x_3}{\partial u} \right) du \, dv,$$

where f_1 is the pre-integrated image along the first dimension. Analogously, we can express the surface integral of f over the region V_{sh} as

$$\iiint_{S_{\text{sh}}} f_1 \, dx_2 \wedge dx_3 = \int_0^1 \int_0^1 f_1(\sigma) \sqrt[3]{4} \left(\frac{\partial x_2}{\partial u} \frac{\partial x_3}{\partial v} - \frac{\partial x_2}{\partial v} \frac{\partial x_3}{\partial u} \right) du \, dv,$$

since

$$\begin{aligned} \frac{\partial \sigma_{\text{sh}}}{\partial u}(u, v) &= \sqrt[3]{2} \frac{\partial \sigma_{\text{sh}}}{\partial u}(u, v) \\ \frac{\partial \sigma_{\text{sh}}}{\partial v}(u, v) &= \sqrt[3]{2} \frac{\partial \sigma_{\text{sh}}}{\partial v}(u, v). \end{aligned}$$

We obtain an explicit expression for the region energy by combining both results

$$E_{\text{region}} = \frac{1}{|V|} \int_0^1 \int_0^1 \left(2f_1(\sigma) - \sqrt[3]{4}f_1(\sigma_{\text{sh}}) \right) \left(\frac{\partial x_2}{\partial u} \frac{\partial x_3}{\partial v} - \frac{\partial x_2}{\partial v} \frac{\partial x_3}{\partial u} \right) du \, dv.$$

Finally, we approximate the double integral by a double sum by sampling u and v at a fixed sampling rate. Since the basis functions and their derivatives are compactly-supported, the number of non-zero elements in the sums is small. Moreover, we precompute their samples and store them in lookup tables .

Chapter 6

Snake-Based Algorithm for Tracking Mitotic Cells

Because biological systems are dynamic, it is highly desirable to quantify their evolution through time in order to improve our understanding of their behavior. Large-scale time-lapse imaging of cells is nowadays performed routinely thanks to the automatization achieved in the field of light microscopy. The obtained datasets are such that it is not possible to analyze them manually within any tolerable amount of time.

Present-day efforts in cell tracking are mostly application-oriented and depend on different methodological approaches [3, 19, 160]. Among them, two paradigms can be identified: the Bayesian framework and variational methods. The former involves a probabilistic reasoning grounded in a motion model [161, 162, 163]. The latter localizes the target accurately at each frame by optimizing a cost function that depends exclusively on the current image, often employing a standard minimization algorithm [164, 165, 166, 167]. The most straightforward technique is to link each detected target in one frame with the closest one in the subsequent frame, where the used distance may include similarity measures. It has also been proposed to use graph-theoretic approaches. In this case, the solution is obtained using standard graph optimization algorithms on a weighted graph composed of the detected targets and all possible correspondences [168, 169].

The variational approach is usually preferred in bioimaging, while the Bayesian framework is quite popular in the Computer Vision. Nonetheless, several attempts have been made in the Computer Vision community to take advantage of Bayesian and variational methods simultaneously. Most of these methods make use of parametric active contours and rely on kernel density estimators, which incorporate many parameters that can degrade the overall performance of the

algorithm [170, 171]. By using active contours of the geodesic variety the use of such estimators was avoided in [172]. This resulted in a significantly slower algorithm than their parametric counterparts due to the high computational load inherent to geodesic active contours.

In this chapter we make use of the active contours of the previous chapters to design a segmentation and tracking method that performs large-scale time-lapse analysis of mitotic cells. The demonstrated efficiency of our active contours allows us to use them as building blocks in a highly parallelizable image analysis toolkit.

This chapter is organized as follows: In Section 6.1, we present a tracking algorithm that combines our active contours with the Bayesian tracking framework. As a result we obtain a parallelizable single cell tracking algorithm capable of handling cell mitosis, and multiple changing motion cell behaviors. Then, we describe a refinement algorithm in Section 6.2 in order to deal with imaging conditions where the cells are densely packed and interact with each other.

6.1 Single Mitotic Cell Tracker

Among the Bayesian methods, our interest lies in the particle filter, which performs a multimodal random search guided by a motion model [173]. The fact that the search is multimodal is important when modeling uncertainties of association in dividing targets. In the role of the parametric contours, we can use any of the methods developed in Chapter 4 or Chapter 5. We embed the snakes in the particle filter in a way that the importance sampling of the particle filter is defined implicitly by the optimization algorithm of the variational method, and the particle weights correspond to the optimal values of the energy function of each individual particle. This construction drastically reduces the number of particles needed to have an accurate description of the target. We make use of the shape information provided by the snake in order to detect the start and end of the mitotic stage within a simplified cell cycle, and use different motion models accordingly.

This section is structured as follows: We first recall elements of the particle-filter framework in Section 6.1.1. Then, we describe our algorithm in Section 6.1.2, and illustrate its capabilities by tracking mitotic HeLa cells and outlying their nuclei in Section 6.1.3.

6.1.1 Particle-Filter Framework

The Bayesian-tracking framework provides a methodology to infer the sequence of hidden states of a dynamic system $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$, using a sequence of noisy measurements $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$. Bayesian estimation is used to recur-

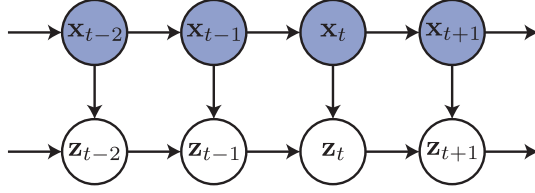


Figure 6.1: Belief network representation of a first-order state-space belief network with a layer of hidden variables. Round nodes denote that the variables are continuously defined, and shaded nodes indicate that the variable is latent/hidden.

sively compute a time-evolving posterior distribution $p(\mathbf{x}_t | \mathbf{z}_{1:t})$. This distribution can be estimated by assuming a Markovian model of the state evolution, $D(\mathbf{x}_t | \mathbf{x}_{t-1})$, and a likelihood that relates the noisy measurements to the hidden state $L(\mathbf{z}_t | \mathbf{x}_t)$.

In the statistical signal processing community, this framework for modeling time-series data that uses a latent, unobserved variables x_t , from which the observations z_t are generated, takes different names: latent Markov models, state-space models or HMMs [174]. We show in Figure 6.1 the corresponding belief network representation of a first-order Markov model. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. The round nodes indicate that the variables are continuous, the shaded nodes indicate latent/hidden variables and the absence of arrows indicate independence relationships [175].

Then, the probability density function (pdf) $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ is estimated in two steps: prediction of the state and update after the new measurement \mathbf{z}_t is available.

In the prediction step, the system model and the estimated posterior density from the previous frame are combined in the Chapman-Kolmogorov equation to obtain the prior density

$$p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) = \int D(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}. \quad (6.1)$$

Next, in the update step, Bayes' rule is used to modify the prior density and obtain the desired posterior

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \propto L(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}). \quad (6.2)$$

The solution of the problem defined by (6.1) and (6.2) is analytically tractable in a limited number of cases (*e.g.*, linear Gaussian models). For most practical

models, sequential Monte Carlo methods are used as an efficient approximation. In these methods, the posterior $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ is represented with a set of N_s random weighted samples, usually referred as particles, $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$ as

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}),$$

where δ is the Dirac delta and the sum of the weights is normalized to the unity. In the classical approach, the particles are chosen using the principle of importance sampling. This principle relies on the availability of an importance function $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t)$ that describes the state space. The idea is to sample the areas of the state space where the importance function is large and to avoid generating samples with low weights, since they provide a negligible contribution to the posterior. Thus, we generate a set of new particles using the importance function, that is

$$\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, \mathbf{z}_t). \quad (6.3)$$

Generally, the importance function can be chosen arbitrarily. The only requirements are the possibility to easily draw samples from it, and to have the same support as $p(\mathbf{x}_t | \mathbf{z}_{1:t})$. When using the importance density function $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t)$, the expectation of any function $f(\mathbf{x}_t)$ with respect to the probability $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ can be rewritten as

$$\int f(\mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t}) d\mathbf{x}_t = \int f(\mathbf{x}_t) \frac{p(\mathbf{x}_t | \mathbf{z}_{1:t})}{q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t)} q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t) d\mathbf{x}_t,$$

where the integration is performed only over the common support of the probability $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ and $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t)$. By drawing N_s samples as in (6.3), the expectation can be approximated as

$$\int f(\mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t}) d\mathbf{x}_t \approx \sum_{i=1}^{N_s} f(\mathbf{x}_t^{(i)}) w_t^{(i)}, \quad (6.4)$$

where

$$w_t^{(i)} \propto \frac{p(\mathbf{x}_t^{(i)} | \mathbf{z}_{1:t})}{q(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)}, \mathbf{z}_t)},$$

and $\sum_{i=1}^{N_s} w_t^{(i)} = 1$. Thus, the Chapman-Kolmogorov equation can be approximated using the right hand side of (6.4). Taking advantage of the fact that we have a robust observation model given by our snakes, we propose to replace the classical importance sampling function by the optimizer of the variational scheme. This novel approach drives the particles towards regions in the state space with high probability.

6.1.2 Variational Importance Sampling

In our setting, the snakes developed in Chapter 4 and Chapter 5 provide an accurate observation model that can be used to describe the shape of any blob-like cell. In such circumstances, the state vector $\mathbf{x}_t^{(i)}$ corresponds to control points of the snake, *i.e.*, $\mathbf{x}_t^{(i)} = (\mathbf{c}^{(i)}[0], \dots, \mathbf{c}^{(i)}[M_i - 1])$, and the measurement vector corresponds to the pixel values $\mathbf{z}_t = \{I_t\}$ of the image within the boundaries of the snake. Note that each particle can have a different number of control points.

At each frame, we propagate each particle from the previous frame following the state evolution model, which generates the predicted set of particles $\{\tilde{\mathbf{x}}_t^{(i)}\}_{i=1}^{N_s}$. Since each particle $\tilde{\mathbf{x}}_t^{(i)}$ is built from a snake, it can be associated with an energy value $E_{\text{snake}}^{(i)}(\mathbf{c}^{(i)})$ measuring the goodness of fit of the snake to the target being tracked. We optimize the energy value of the predicted set of particles following the gradient-based optimizer of the snake. This defines the optimized set of particles $\{\mathbf{x}_{t,\text{opt}}^{(i)}\}_{i=1}^{N_s}$ with an optimized set of snake energies $\{E_{\text{snake}}^{(i)}(\mathbf{c}_{\text{opt}}^{(i)})\}_{i=1}^{N_s}$. Following the principle of maximum entropy, we assume that $E_{\text{snake}}(\mathbf{c})$ is a random variable with exponential distribution, which leads to assign the particle weights $w_t^{(i)}$ to

$$w_t^{(i)} \propto e^{-\lambda E_{\text{snake}}^{(i)}(\mathbf{c}^{(i)})},$$

where λ is a parameter that controls the sharpness of $p(\mathbf{x}_t | \mathbf{z}_{1:t})$.

Using the proposed scheme, the importance sampling of the particle filter, usually performed by (6.3), is given implicitly by the optimization algorithm used in the variational method. This interpretation arises naturally since the role of the optimizer is to attract the snakes, and therefore the particles, to the target under inspection. As a consequence, the weights of the particles within the region of convergence of the optimizer will gain importance compared to the ones that are not. Therefore, a much smaller set of particles is necessary to describe the high-probability regions of the state space.

Finally, we perform a resampling step to eliminate particles that have small weights and to focus on particles with large weights. The resampling step involves generating a new set by sampling (with replacement) N_s times from $\{\mathbf{x}_{t,\text{opt}}^{(i)}\}_{i=1}^{N_s}$, which leads to the equiprobable set of particles $\{\mathbf{x}_t^{(i)}, \frac{1}{N_s}\}_{i=1}^{N_s}$. The estimation at each frame of the location and shape of the target being tracked at each frame can be carried out efficiently with the MAP estimator as follows:

$$\hat{\mathbf{x}}_t = \arg \max_{\mathbf{x}_t} \{p(\mathbf{x}_t | \mathbf{z}_{1:t})\} \approx \arg \max_i \{w_t^{(i)}\}.$$

Thus, the *maximum a posteriori* (MAP) estimation of the target corresponds to the optimized particle with highest weight. All these operations are summarized in Algorithm 1.

Algorithm 1: Snake-based particle filter

input: Particle set $\{\mathbf{x}_{t-1}^{(i)}\}_{i=1}^{N_s}$ and current image I_t
output: MAP estimation $\hat{\mathbf{x}}_t$ and particle set $\{\mathbf{x}_t^{(i)}\}_{i=1}^{N_s}$

for $i \leftarrow 1$ **to** N_s **do**
 $\tilde{\mathbf{x}}_t^{(i)} \leftarrow$ Propagate $\mathbf{x}_{t-1}^{(i)}$ with the motion model;
 $\{\mathbf{x}_{t,\text{opt}}^{(i)}, E_{\text{snake}}^{(i)}(\mathbf{c}_{\text{opt}}^{(i)})\} \leftarrow$ Adjust the snake to I_t ;
 $w_t^{(i)} \leftarrow e^{-\lambda E_{\text{snake}}^{(i)}(\mathbf{c}_{\text{opt}}^{(i)})}$;
end
for $i \leftarrow 1$ **to** N_s **do**
 $w_t^{(i)} \leftarrow w_t^{(i)} / \sum_j w_t^{(j)}$;
end
 $\hat{\mathbf{x}}_t \leftarrow \arg \max_i \{w_t^{(i)}\}$
 $\{\mathbf{x}_t^{(i)}\}_{i=1}^{N_s} \leftarrow \text{Resampling}(\{\mathbf{x}_{t,\text{opt}}^{(i)}, w_t^{(i)}\}_{i=1}^{N_s})$;

6.1.3 Application to Time-Lapse Microscopy

In this section, we use our snake-based particle filter to construct the lineage of migrating HeLa cells, and outline their nuclei. We used the 2D spline-based snake from Chapter 4 with $M = 3$ and region energy only.

6.1.3.1 Biphasic Motion Model The HMM is a classical signal processing model that is not capable of handling the different behavior that the cells exhibit during the cell division process. A more complex model is the switching linear dynamical systems (SLDS) which breaks the time series into segments, each modeled by a potentially different motion model. We show in the associated belief network in Figure 6.2.

For our particular application, two different motion models are considered depending on the state of the cell cycle. Both models are considered to be linear, with

$$\tilde{\mathbf{x}}_t = \mathbf{x}_{t-1} + \mathbf{n}_t(\mathbf{s}_t), \quad (6.5)$$

where \mathbf{n}_t is a random vector that depends on the state of the cell \mathbf{s}_t at time t . The two cell states are:

- Non-mitotic, where the nuclei are essentially circular, and move and deform without any preferred direction, as shown in Figure 6.3a and Figure 6.3b;
- Mitotic, where nuclei are more elongated and brighter than in the non-mitotic state, and where the movement during the splitting is fast and perpendicular to the main axis of the cell, as shown in Figure 6.3c and Figure 6.3d.

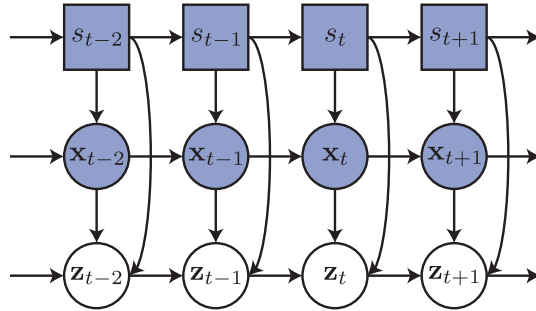


Figure 6.2: Belief network representation of the SLDS. Square nodes denote discrete switch variables s_t ; x_t are continuous latent/hidden variables and z_t continuous observed/visible variables. The discrete state s_t determines which linear dynamical systems (LDS) is operational at time t .

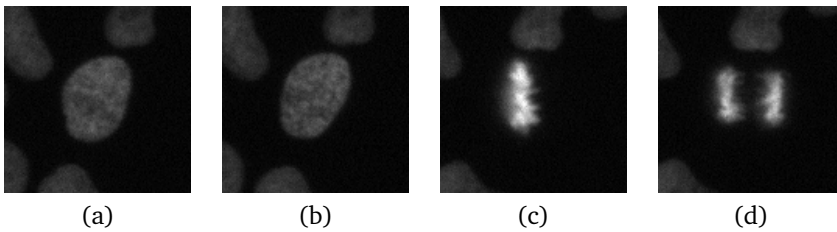


Figure 6.3: Migrating HeLa cell nuclei: (a) non-mitotic state at time $(t - 1)$, (b) non-mitotic state at time t , (c) mitotic state at time $(t - 1)$, (d) mitotic state at time t .

For the non-mitotic stage, the natural choice in (6.5) is to assume Gaussianity and independence for each component of \mathbf{n}_t . For the mitotic stage, we adopt a purely translational model perpendicular to the main orientation axis. A cell is considered to enter in the mitotic state if its MAP estimation is brighter and more eccentric than a certain threshold values. At that point, the motion model switches to the mitotic one, and eventually returns to the non-mitotic one once the values of the brightness and eccentricity get below the thresholds.

6.1.3.2 Experimental Results To illustrate our method, we applied our algorithm to a time-lapse sequence of images of HeLa nuclei expressing fluorescent core histone 2B on an RNAi live cell array.¹ We focused on building the cell lineage of a single cell. We only used a total of 20 particles. The thresholds, λ , and the standard deviations for \mathbf{n}_t were chosen empirically.

1. Courtesy of D. Gerlich, Institute of Biochemistry, ETHZ, Zürich.

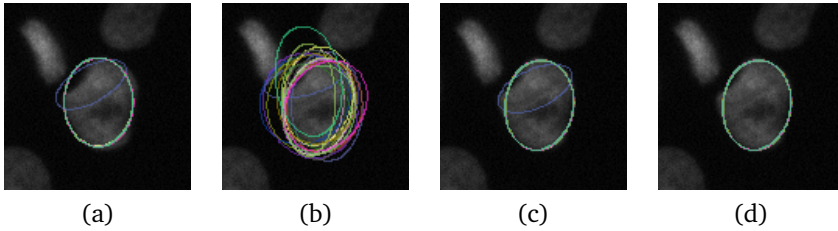


Figure 6.4: Different steps of Algorithm 1 with non-mitotic motion model. (a) Initial. (b) Propagated. (c) Optimized. (d) Resampled.

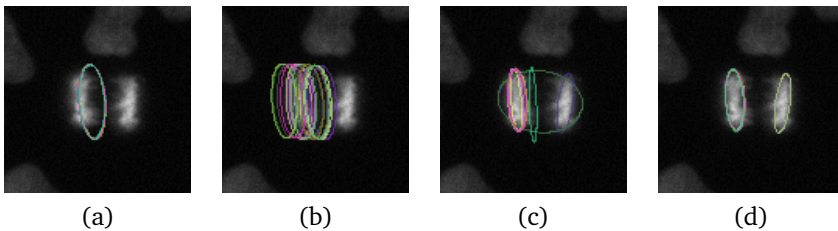


Figure 6.5: Different steps of Algorithm 1 with mitotic motion model. (a) Initial. (b) Propagated. (c) Optimized. (d) Resampled.

In Figure 6.4, we show the behavior of Algorithm 1 when a non-mitotic motion model is used. In particular, we observe in Figure 6.4a the outlines of the snakes representing the particles from the previous frame. These particles are propagated following the non-mitotic motion model to the locations shown in Figure 6.4b. After optimizing the snakes, we obtained the particles shown in Figure 6.4c, and, finally, after the resampling, the particles in Figure 6.4d. Note that, after the optimization, one snake converged to a local minima, but its weight was negligible compared to the others. Therefore, it was eliminated in the resampling step. In Figure 6.5, we show the behavior of Algorithm 1 when a mitotic motion model is used and when the cell division occurs. In particular, we observe in Figure 6.5a the outlines of the snakes representing the particles from the previous frame located at the same position. These particles are propagated following the mitotic motion model to the locations shown in Figure 6.5b. After running the snake optimizer we obtained the particles shown in Figure 6.5c, and after the resampling we obtained the particles shown in Figure 6.5d. Note that, after the optimization, some snakes converged to different targets, and this information was preserved in the resampling step.

We show in Figure 6.6 and Figure 6.7 the temporal evolution of the mean brightness intensity and the eccentricity of a single nucleus respectively. We can ob-

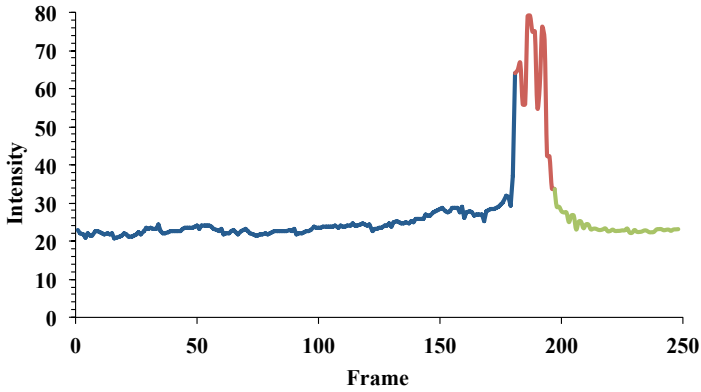


Figure 6.6: Mean intensity within the elliptical MAP estimation. Three stages can be clearly differentiated. The first one prior to frame 180 in which the cell is not in mitotic stage. The second one occurs between Frame 180 and 188 in which the cell becomes brighter and undergoes mitosis. Finally, the third one in which the cell returns to a non-mitotic stage.

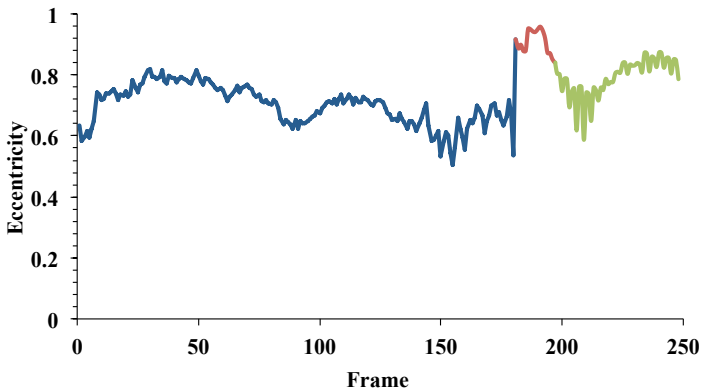


Figure 6.7: Eccentricity of the elliptical fit of the MAP estimation. Three stages can be clearly differentiated. The first one prior to frame 180 in which the cell is not in mitotic stage. The second one occurs between Frame 180 and 188 in which the cell becomes more elongated and undergoes mitosis. Finally, the third one in which the cell returns to a non-mitotic stage.

serve a simultaneous peak in both graphs between frames 180 and 188, which corresponds to the mitotic stage of the cell.

The use of our biphasic motion model would not have been possible if we had not used the optimized snake to obtain an accurate estimation of the orientation of the cell during the mitotic stage with a reasonable number of particles. Moreover, due to the capability of the particle filter to describe multimodal distributions, our algorithm is capable of building the cell lineage, which the snake could not have achieved on its own.

The computation time is usually directly related to the number of particles used in the particle filter. Since our variational importance sampling provides a better description of the high-probability regions of $p(\mathbf{x}_t | \mathbf{z}_{1:t})$, a reduced number of particles is necessary. Moreover, the optimization of each snake during the variational importance sampling stage can be carried out independently, which makes the algorithm fully parallelizable.

6.2 High-Throughput Multi-Target Tracker

The simplicity of tracking one cell is lost when tracking a whole population of densely packed cells. Joint optimization tracking techniques provide good results in such cases, but the computational complexity increases dramatically [161, 176]. This is partly due to the fact that the cells all interact with each other, and no independence relations in the probability model can be easily established. Therefore, the joint probability space of all targets must be used for tracking every single cell within the whole population.

Here, we consider high-density crowd scenes. It has been observed, in the context of human crowd tracking, that the locomotive behavior of an individual in a crowded scene is a function of collective patterns evolving from the space-time interactions of individuals among themselves [177].

In this section, we present a fast algorithm which is specialized for tracking biological cells within crowds. It relies on graph-theoretic techniques to minimize a cost functional that models the characteristic motion in highly packed scenes and imposes a certain level of coherence in the displacement field while being capable of handling large movements (*i.e.*, coarse temporal resolution). The output of this algorithm can be easily incorporated as a base displacement field in a motion model for the tracker introduced in Section 6.1.

6.2.1 Notation

We denote $M^k = \{\mathbf{m}_j^k\}_{j=1 \dots N_k}$ the set of measurements $\mathbf{m}_j^k = (\mathbf{x}_j^k, \mathbf{s}_j^k)$ at frame k , where each measurement contains information about the position \mathbf{x}_j^k and features \mathbf{s}_j^k of each target. In our snake framework, the features \mathbf{s}_j^k contain, but

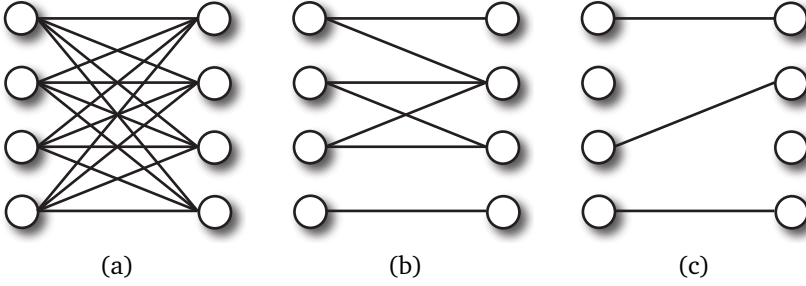


Figure 6.8: Bipartite graph representation of the sets M^{k-1} (left side), M^k (right side) with (a) all possible correspondences given by Ω , (b) the most likely correspondences given by G , and (c) a possible pairing A^k .

are not limited to, shape descriptors from the snake control points. Analogously, we define the set of positions and features at a given frame by $X^k = \{\mathbf{x}_j^k\}_{j=1\dots N_k}$ and $S^k = \{\mathbf{s}_j^k\}_{j=1\dots N_k}$.

We are interested in finding the best pairing between two consecutive frames given the measurements in these frames. Formally, a pairing is as a subset

$$A^k \subset \Omega = \{1 \dots N_{k-1}\} \times \{1 \dots N_k\},$$

where $(i, j) \in A^k$ indicates that the measurement \mathbf{m}_i^{k-1} in frame $(k-1)$ corresponds to the measurement \mathbf{m}_j^k in frame k . We restrict the admissible solution by disallowing multiple assignments to the same measurement. Given $(i, j), (l, n) \in A^k$, then $\mathbf{m}_i^{k-1} = \mathbf{m}_l^{k-1}$ if and only if $\mathbf{m}_j^k = \mathbf{m}_n^k$ (see Figure 6.8). Note that the cell-tracking problem is equivalent to finding the corresponding A^k at each transition interval.

6.2.2 Probabilistic Graph Formulation

We consider the measurement sets as random variables. This implies that we are also looking for a pairing which is a random element within the space of all possible pairings. We are interested in finding a pairing that maximizes its conditional probability given the known information, that is, the measurements

$$A_{\text{opt}}^k = \arg \max_{A^k \subset \Omega} \{P(A^k | M^{k-1}, M^k)\}, \quad (6.6)$$

where $P(A^k | M^{k-1}, M^k)$ is the probability associated to the pairing A^k conditioned to the sets of measurements M^{k-1} and M^k .

Using an approach similar to the one in the Bayesian filtering framework, and assuming mutual independence between the positions and the features, we restate the maximization criteria (6.6) as

$$A_{\text{opt}}^k = \arg \max_{A^k \subset \Omega} \{P(A^k)P(X^k|X^{k-1}, A^k)P(S^k|S^{k-1}, A^k)\}, \quad (6.7)$$

where $P(A^k)$ is the marginal probability of A^k , and reflects our prior knowledge about the preferred pairings, $P(X^k|X^{k-1}, A^k)$ corresponds to a measure of likelihood of the positions given that we know the position in the previous frame through the pairing A^k , and $P(S^k|S^{k-1}, A^k)$ corresponds to a measure of how likely the feature values are given the knowledge of the feature values in the previous frame through the pairing A^k .

Using the maximum-entropy principle, we finally model the probabilities as

$$P(X^k|X^{k-1}, A^k) \propto e^{-\sum_{v \in A^k} \mu_X(v, X^{k-1}, X^k, A^k)}$$

$$P(S^k|S^{k-1}, A^k) \propto e^{-\sum_{v \in A^k} \mu_S(v, S^{k-1}, S^k, A^k)},$$

where μ_X is a compatibility measure of the joint movement and μ_S is a feature compatibility measure.

6.2.3 Efficient Graph-Based Algorithm

We now cast the problem of finding the best pairing A_{opt}^k in (6.7) as a global graph-optimization problem. First of all, we consider the complete bipartite graph whose nodes represent the measurement sets M^{k-1} and M^k , and whose edges are represented by Ω . Our goal is to select a subgraph such that the sum of the weights w_{ij} of the edges is maximized subject to the restriction that all vertices in this graph have a degree less or equal to one, so that each target detection from set M^{k-1} is assigned to at most one target detection on M^k and *vice versa*. This set of edges receives the name of *matching* [178]. It exactly corresponds to the pairing A^k if we appropriately choose the edge weights w_{ij} . Despite the fact that we are maximizing a criterion based on graph edges, we are not biasing the solution toward high connectivity, as is usually the case for non-weighted graph matchings [179, 180].

This graph-optimization problem corresponds to a known problem in graph theory called *maximum weighted bipartite graph matching* [179, 180], which can easily be expressed in the context of binary linear programming (BLP). In this framework, the objective functional to maximize is

$$\sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} w_{ij} \chi_{ij} \quad (6.8)$$

subject to $\sum_{i=1}^{N_{k-1}} \chi_{ij} \leq 1$, $\sum_{j=1}^{N_k} \chi_{ij} \leq 1$, and $\chi_{ij} \in \{0, 1\}$.

The solution to the BLP would be either $\chi_{ij} = 0$ or $\chi_{ij} = 1$, where $\chi_{ij} = 0$ denotes $(i, j) \notin A^k$ and $\chi_{ij} = 1$ denotes $(i, j) \in A^k$.

In our case, we know beforehand that many pairings between measurements of M^{k-1} and M^k are unlikely. These incompatibilities are usually known when designing the experimental setup (e.g., a cell will not be able to move more than a certain distance, the shape of a target cannot change through time).

In our formulation, the probability of any matching that includes one of these unlikely edges is set to $P(A^k) = 0$. Therefore, the solution has to be found in a sparse-graph G (see Figure 6.8b), where the edges only connect possible correspondences (i.e., measurements located within a certain region). For all possible matchings within this sparse graph, we can choose a non-zero value for $P(A^k)$ such that the solution of (6.7) corresponds to the solution in (6.8) with

$$w_{ij} = -\mu_S((i, j), S^{k-1}, S^k, A^k) - \lambda \mu_X((i, j), X^{k-1}, X^k, A^k),$$

where λ is a positive tradeoff parameter between the compatibility measure of the joint movement and the feature compatibility measure. For $\lambda = 0$ the solution of the pairing is totally based on target features, and for high values of λ , the solution is dominated by the joint movement criteria. Note that the compatibility measure of the joint movement acts as a regularization term; that is, modifies the optimal solution of the problem given our prior knowledge of the flow behavior. Therefore, the compatibility measure of the joint movement cannot act independently and should always work together with the feature compatibility measure.

Note that it is necessary to have a pairing A^k to compute the edge weights w_{ij} . This is resolved by the following iterative algorithm:

Algorithm 2: Iterative multi-target tracker

Initialization of G, A^k ;

repeat

$\mu_S, \mu_X \leftarrow A^k$ (update compatibility measures);

$w_{ij} \leftarrow \mu_S, \mu_X$ (update edge weights);

$A^k \leftarrow$ **Maximum Weighted Graph Matching** (G, w_{ij});

until convergence;

return A^k

The most costly part of the algorithm corresponds to the execution of the *Maximum Weighted Graph Matching* function. Some graph algorithms can perform this function in $\mathcal{O}(NE + N^2 \log(N))$ steps, where $N = N_{k-1} + N_k$, and E is the number of edges, or equivalently, the number of possible correspondences.

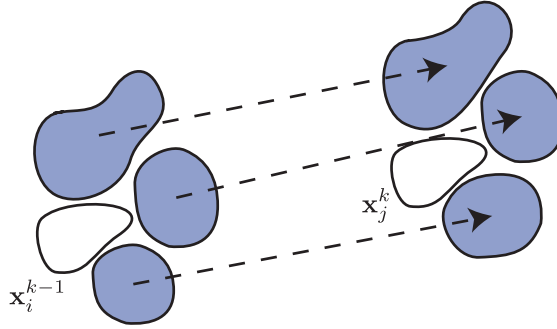


Figure 6.9: Schematic representation of the optimal configuration when pairing a set of neighboring targets from frame $(k-1)$ (left) to frame k (right) using the compatibility measure μ_X favouring joint displacements. Given an initial pairing A^k mapping the shaded targets, the optimal configuration is reached when \mathbf{x}_i^{k-1} is paired to \mathbf{x}_j^k .

6.2.4 Motion Model

Blocks of cells often tend to move together in heavy populated environments, so that the cells preserve a certain spatio-temporal continuity in their movement. We model this behavior using a displacement model in the compatibility measure of the joint movement μ_X , as follows:

$$\begin{aligned} \mu_X((i, j), X^{k-1}, X^k, A^k) \\ = \frac{1}{\mathcal{N}_i^{k-1}} \sum_{\mathbf{x}_i^{k-1} \in \mathcal{N}_i^{k-1}} \underbrace{\|(\mathcal{A}(\mathbf{x}_i^{k-1}) - \mathbf{x}_i^{k-1})\|}_{\text{neighbor } l \text{ displacement}} - \underbrace{\|(\mathbf{x}_j^k - \mathbf{x}_i^{k-1})\|}_{i \rightarrow j \text{ displacement}}, \end{aligned} \quad (6.9)$$

where \mathcal{N}_i^{k-1} is the non-empty set of neighbors of the detected cell at \mathbf{x}_i^{k-1} , and where \mathcal{A} is a mapping induced by A^k that satisfies $\mathcal{A}(\mathbf{x}_i^{k-1}) = \mathbf{x}_j^k$ whenever $(i, j) \in A^k$.

The regularization term μ_X measures the average relative displacement of the neighboring targets given a pairing A^k . As shown in Figure 6.9, the minimization of (6.9) imposes that neighboring target detections go to neighboring target detections. This term becomes crucial when all targets look alike, which is the case when imaging living cells.

6.2.5 Application to Time-Lapse Microscopy

In order to validate the efficacy of the method in tracking individual cells in large crowds, the algorithm was applied to a sequence of images obtained from yeast cell populations.

S. cerevisiae strains (ATCC 201388) were grown within a microfluidic chamber in YPD medium at 30°C. Monolayer-grown cell crowds were imaged with a Nikon Ti-E microscope (Nikon instruments inc., Melville, U.S.A.), a 60× objective (plan apo, 1.4 oil), and a 1.5× scope, resulting in a total 90× magnification. An iXonEM camera (Andor technology plc., Belfast, U.K.) controlled by a VB6 based software was used to acquire 14-bit images of (1024 × 1024) pixel with an exposure time of 50ms, an analogical gain of 2.4, and an EM gain of 2.

We used a watershed transform to split the image domain in non-overlapping regions and serve to us an automatic initialization for our active contours. This segmentation provides us with several important parameters for the construction of our graph G . In particular we identify the centroid of each watershed region as the position \mathbf{x}_i^{k-1} , and the grayscale values within a bounding box covering the watershed region as the features \mathbf{s}_i^{k-1} . The watershed segmentation also gives information about neighboring regions: regions that share dams with other regions are considered neighbors. We use this information to determine the non-empty sets of neighbors in (6.9). For the watershed region detector to succeed, we need dense areas of targets within crowds. Otherwise, some heuristic methods can be applied to eliminate regions that do not represent cells.

We define the similarity criteria μ_S between two targets as the mean-squared error of the grayscale values between the two corresponding cell shapes. In order to avoid misalignments of the images when computing the MSE, we used the Three-Step Search (TSS) block-matching algorithm [181]. Some typical results are presented in Figure 6.10. In the simulations presented in this section we used an initial matching A^k based on features only (*i.e.*, we run the Maximum Weighted Graph Matching function once with $\lambda = 0$).

To illustrate the adequacy of our motion model, we compared in Figure 6.11 the use of μ_S alone to the joint use of μ_S and μ_X . In this setup, all targets look very similar, and the μ_S measure is not able to discriminate well between all possible pairings. Introducing μ_X notably improves the solution even after one single iteration.

For the purpose of assessing tracking performances, we manually identified cell trajectories within an area of interest with an average population of 120 yeast cells over a span of 50 frames. We quantified the quality of the pairing process by computing the rate of successful connections made by the algorithm over the total number of connections in each frame. We summarize the results obtained by our method in Table 6.1.

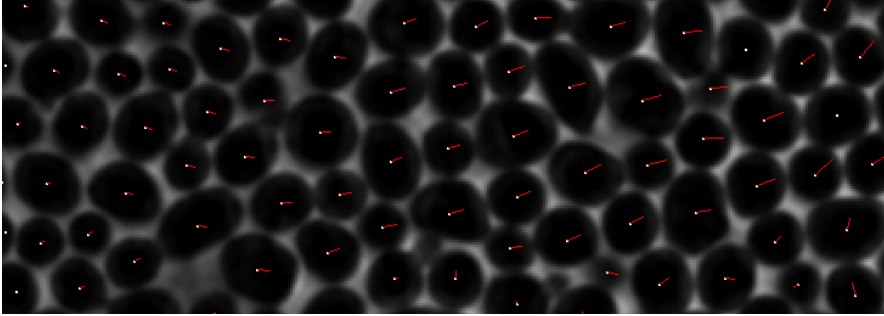


Figure 6.10: Flow-like behavior of the constrained movement of a crowd of cells. The red lines point towards the position in the subsequent frame.

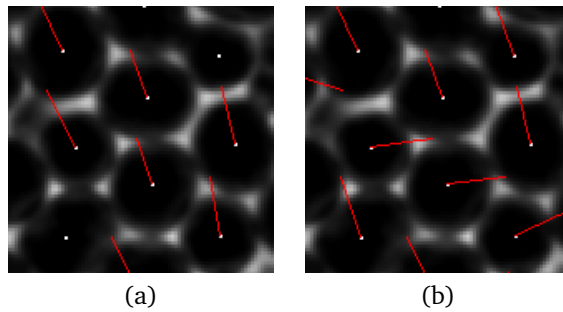


Figure 6.11: Pairing between two consecutive frames. The red lines point towards the location of the corresponding region in the subsequent frame. (a) Pairing obtained using only on the similarity measure μ_s ($\lambda = 0$). (b) Pairing obtained using a weighted combination of both measures after one iteration using (a) as initial matching ($\lambda = 100$).

λ	Max. rate [%]	Min. rate [%]	Average rate [%]
0	100.0	67.0	95.5
50	100.0	72.0	97.6
100	100.0	90.0	98.6

Table 6.1: Success rate when tracking yeast cells.

The similarity measure μ_S is sufficient for obtaining a correct pairing when cell movements are small (*i.e.*, past and present positions overlap). However, when pressure inside the device exceeds a critical level, cells tend to displace in clusters pushing the neighboring ones. This situation cannot be appropriately handled by using μ_S alone, and tracking is then improved with the introduction of μ_X . The appropriate choice of λ strongly depends on the behavior of the motion vector field and should be further investigated.

6.3 Conclusions

In the first part of this chapter, we have proposed a new methodology that fuses in a single tracker the two major tracking philosophies and that retains the advantages of both. We showed that, by using a robust variational method, it is possible to replace the importance sampling function of the particle filter and obtain an alternative scheme. The resulting algorithm is capable of creating an accurate segmentation of elliptic targets with a reduced number of particles, and capable of detecting and tracking cells undergoing mitosis.

In the second part of the chapter, we have presented a new algorithm for multi-cell tracking in crowded areas. It is fast and easy to implement. It runs in real time and is able to deal with thousands of cells. We have specifically designed model-based cost functions that take account of multiple cell interactions. The parameter λ of the algorithm needs to be chosen so as to strike a balance between the frequently conflicting goals of having flow-like trajectories and enforcing the similarity of individual cells across frames.

Chapter 7

Conclusion

In this chapter, we first recap all the technical contributions presented in this thesis. Then, we briefly show some life-sciences research projects in which our tools have been incorporated within larger image analysis pipelines. Finally, we comment on the future research possibilities.

7.1 Technical Contributions

Our schemes are centered on shape models that ease feature extraction of blob-like shapes. We have optimized the computational efficiency and have provided a quantitative and qualitative assessment of our methods. The main technical contributions of this thesis grouped by topic are as follows:

Contributions in the fields of approximation and spline theory.

- We fully characterized a new family of basis functions with shortest support that allows one to reproduce exponential polynomials.
- We provided subdivision schemes and nonstationary multiresolution algorithms based on these functions.
- We characterized the order of approximation of such nonstationary multiresolution schemes.
- We used these minimum-supported basis functions to design spline curve models that reproduce ellipses and higher-order harmonics. In particular, we tailored these bases to obtain maximal-smoothness basis functions, and interpolatory basis functions.

Contributions in the field of 2D image segmentation.

- We used the minimum-supported bases that reproduce ellipses to construct parametric active contours. These models can be characterized exactly by as

few as three control points, but they can reproduce any planar closed curve, with arbitrary precision, by adding a sufficient number of nodes.

- We also provided closed expressions to compute the best elliptical approximation, in a least-squares sense, of a contour described by an arbitrary number of control points.
- We applied our snakes to a variety of problems that involve synthetic simulations and real data.
- Our active contours are general, and compatible with any traditional energy functional. To exemplify our method, we designed a combination of contour and region-based energies. The former uses the gradient direction in order to impose consistency constraints during the segmentation process. This has proved to be very useful when segmenting or tracking cells in crowded environments. In the second case, the energy depends on the contrast between two regions, one being delineated by the curve itself, and the other by an ellipse of double area.
- We accelerated the implementation of our snakes by taking advantage of Green's theorem. This was possible thanks to the availability of explicit expressions for the basis functions.

Contributions in the field of 3D image segmentation.

- We extended our planar parametric snake to a fully 3D spline-based parametric snake with a sphere-like topology. Our snake can approximate any blob-like structure with arbitrary precision. Thanks to the underlying B-spline representation, the modification of one control point affects a limited region of the snake surface, which results in intuitive interactions with the user. This is made possible by a parameterization based on splines.
- We have applied our snakes to a variety of problems that involve synthetic simulations and challenging real datasets, where the object contours were not well defined.
- We designed an edge-based energy that is capable of maintaining the consistency of the segmentation in the presence of clutter. This is accomplished by penalizing mismatches in the directions of the image gradients. Furthermore, we combined it with a robust region-based energy.
- We accelerated the implementation of our snakes by taking advantage of Gauss' theorem in 3D. Again, this was possible thanks to the availability of explicit B-spline expressions.
- We introduced a novel technique to detect self-intersection in order to know when our 3D snake loses the sphere-like topology based on the computation of the Euler characteristic.

Contributions in the field of spatio-temporal object tracking.

- We have designed a new methodology that combines in a single algorithm the two major tracking philosophies and that retains the advantages of both using our snakes as building blocks. We were able to handle object division and track a large number of cells in crowded environments.

7.2 Contributions to Research Projects in Life Sciences

The design of the segmentation and tracking algorithms presented in this thesis was initially motivated for the analysis of blob-like biological objects. However, our algorithms are general enough to be applicable in a wider range of situations. Now, we show some research projects in life sciences where our tools have been incorporated within larger analysis pipelines. Some of the projects required particular customizations to the characteristics of each problem (e.g., specific energy functionals for the snakes). These modifications are just briefly described in order not to overload this thesis, and they will result in future joint publications.

7.2.1 The DynamiX Project

As it was mentioned in the introduction, the design of algorithms in this thesis had been motivated by the DynamiX Project (see Section 1.3 for a detailed description).

The microfluidic live cell imaging platform designed in the project can generate a large quantity of images in a short time. For example, an experiment with 1152 strains, running for 12 hours, generates more than 40000 images, grouped in 1152 time-lapse sequences of 36 frames [182]. Single cell analysis is not possible under such conditions within any reasonable amount of time. Thus, an efficient fully automated image analysis scheme is required. To respond to the specific needs of the datasets of the project, we implemented the custom image analysis pipeline shown in Figure 7.1.

In order to minimize the acquisition mechanism, two distinct cell chambers were imaged at the same time. The drift in the position of the chambers and the separation of the two chambers was performed in a preprocessing step. Then, the active contours described in Chapter 4 were used to obtain an accurate representation of the boundaries of the cells. This was performed in a multiresolution fashion: first an elliptic fit was performed, and then a refinement was obtained by increasing the number of control parameters of the model. The ellipse-reproducing property and the inherent properties of the basis functions designed in Chapter 3 were crucial in order to perform the refinement efficiently. At the next step of the analysis, the segmented objects were filtered based on several shape descriptors in order to remove dead cells and segmentation errors (i.e., size, circularity, snake energy). Then, the filtered set of cell boundaries

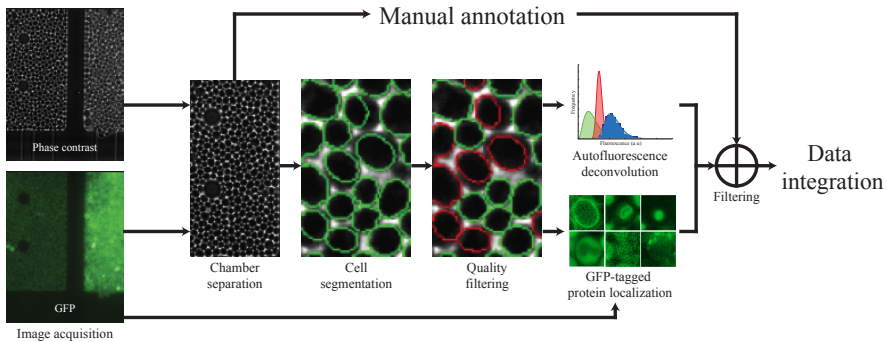


Figure 7.1: Schematic of the image analysis pipeline in the DynamiX project for computing single-cell protein abundance and protein localization.

were used for statistical single cell analysis and feature extraction. In particular, the protein abundance and localization were computed based on the amount and spatial distribution of the fluorescence. Finally, a few sequences were manually annotated in order to provide a cross validation set of features for the automatic algorithm.

This analysis pipeline can record protein abundance and noise, cell size and protein localization of the 40000 images (12 hours experiment) in less than a day.

7.2.2 The WingX Project

The WingX project is also part of the SystemsX.ch consortium. The aim of this project is to quantify the genetic program that governs the growth and shape of the wing of the *Drosophila* fly. This organ is a model uniquely suited for a systems biology study.

Advances in developmental genetics provide us today with the toolkit of organogenesis (e.g., morphogens, transcription factors). This toolkit will be used to build the *Drosophila* wing, in a reproducible fashion, determining the size and form. For that purpose, an interaction between the computational and the experimental stages of the project is crucial.

The goal is to provide a quantitative description of wing development at a multiscale systems level as determined by the interaction of processes at the molecular, cellular, and tissue level.

We embedded the active contours exposed in Chapter 4 within an image analysis tool specifically developed to automatically quantify morphological properties

of the wing and gene expression information on fluorescent confocal images. The parametric model of the snakes provides a convenient way to describe the structural properties of the wing and offers the final user an intuitive mechanism to edit the segmentation through manipulation of the control points of the snake (see Figure 7.2). Here, the snake model presented in Chapter 4 was modified by adding an internal cross-like structure to better fit the anatomy.

Evaluating the effect of a single mutation on the gene expression requires to analyze around 30 wings in order to obtain statistically meaningful results. We note that there are thousands of genes expressed in a biological system like the *Drosophila* fly. So far, hundreds of wings have been analyzed.

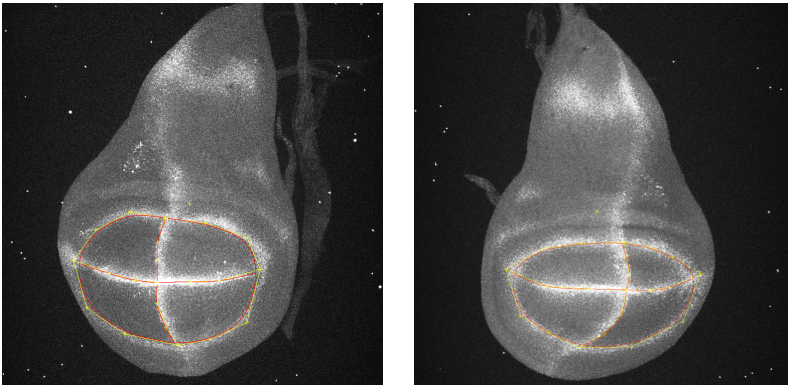


Figure 7.2: Segmentation of the wing pouch structure of the *Drosophila* fly using fluorescent confocal images of the gene expression *Wg-Ptc-AB*. The snake contour is shown as a smooth curve, while the '+' elements are the spline control points connected by a control polygon.

7.2.3 *Drosophila* Fly Locomotion Study

The aim of this project is to understand how voluntary locomotion is orchestrated by the brain. Addressing this fundamental question requires studying voluntary motion in an animal that can be experimentally manipulated to generate and test hypotheses. In the project, the *Drosophila* fly is used as a model due to its relatively simple nervous system and complex locomotor behaviour. It is possible to activate, inhibit, and kill specific neurons of interest in the fly to dissect their role in locomotor neural circuits. Furthermore, one can sequence and manipulate the entire genome of the fly to see what role genes play in behavior.

In order to study locomotion in *Drosophila*, one must be able to quantify their walking behaviors with high precision. This allows to measure the effects of

experimental perturbations such as neuron activation or knockdown as well as genetic manipulations on walking. Such experiments allow to test models of how the fly can produce complicated and robust locomotor sequences (e.g., traversing cluttered environments, escaping from traps).

Currently the state-of-the-art for *Drosophila* behavioral image processing is Ctrax (The Caltech Multiple Walking Fly Tracker), which can only determine the position and orientation of the body. We incorporated the active contours exposed in Chapter 4 within a new image analysis tool specifically developed for automatically segmenting and tracking the bodies of the *Drosophila* flies as well as their legs. The mechanism of modifying the snake curve through the manipulation of control points was a key feature to design prior shapes adapted to the flies (see Figure 7.3). Our tracking algorithms exposed in Chapter 6 are also being introduced in order to consider several motion models.

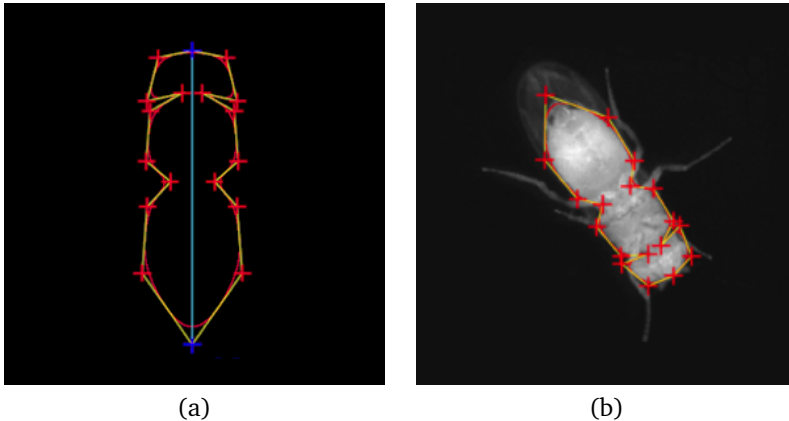


Figure 7.3: (a) Prior-shape model of the body of the *Drosophila* fly. (b) Segmentation of the body of the *Drosophila* fly. The snake contours are shown as smooth curves, while the '+' elements are the spline control points connected by a control polygon.

7.2.4 Estimation of Local Aortic Elastic Properties with MRI

The aim of this project was to use a non-invasive technique to measure elasticity properties of the aorta. More precisely, the parameters of interest were the Pulse Wave Velocity (PWV) and the aortic compliance. These parameters are considered as important determiners of heart load and clinically useful indices of cardiovascular risk.

For that purpose, we used 3D time-lapse sequences of flow-sensitive magnetic resonance imaging (MRI). In these sequences we obtain two different kind of measurements: the magnitude and the phase. The former provided us anatomical information that can be used to segment the aorta, and the latter provided us information concerning the blood flow through the imaged cross-section (see Figure 7.4).

We used the active contours exposed in Chapter 4 to segment and track a cross-section of the aorta. The ellipse-reproducing property of our snakes showed to be very useful for reducing the number of parameters involved in the model, increasing the performance of the segmentation and the tracking.

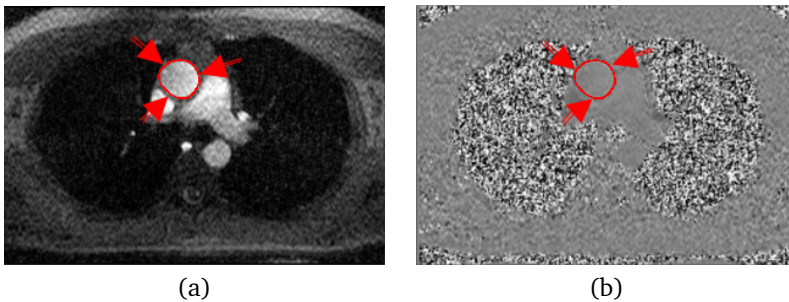


Figure 7.4: Segmentation of the cross-section of the aorta in a flow-sensitive MRI. (a) Magnitude image with the contour of the snake overlying the ascending aorta. (b) Phase image with the contour of the snake overlying the ascending aorta.

7.2.5 Assessment of Chromosomal Size Variation in CHO Cells

The aim of this project was to analyze metaphase chromosome length in Chinese hamster ovary (CHO) cells. CHO cells are widely used for the production of recombinant proteins. Currently, the pharmaceutical industry relies on stable CHO cell clones for therapeutic protein production. Most of the currently used methods to analyze clonality and stability in recombinant cell lines only take into consideration the cellular phenotype. However, few methods are available to study clonality and genomic stability in recombinant cell populations.

We customized the active contours exposed in Chapter 4 within an image analysis tool to segment the chromosomes, sort them according to size, and extract different length patterns. Again, the mechanism of modifying the snake curve through the manipulation of control points was a key feature to provide biologists with a user-friendly and agile platform to work with. Moreover, the explicit expression of our active contours made possible to introduce prior-shapes

and enforce topology. This is necessary when interaction of chromosomes occur (e.g., chromosomes touching or even laying on top of each other).

The method was applied to parental and recombinant CHO cells. Metaphase chromosome spreads were prepared from growth-arrested cells and visualized with a confocal microscope after fluorescence staining. In studies of seven different CHO cell lines (20 metaphase spreads for each cell line), it was noticed that the average chromosome number was not homogenous and each cell line had a specific chromosomal length pattern. This pattern can be used to identify the CHO cell lines and to assess the degree of homo- or heterogeneity in clonal populations [183].

7.3 Outlook for Future Research

The research presented here opens several interesting avenues for future investigations. Some of them are listed below.

- Interpolatory basis functions: The family of minimum-support basis functions introduced in Chapter 3 focuses on computational efficiency and reproduction properties, while leaving the interpolation property aside. This is not an issue when dealing with 2D curves due to the usage of control polygons. However, the interaction feels less natural for the 3D snakes. We would like to investigate the set of interpolatory and \mathcal{C}^k smooth basis functions with certain reproduction properties while maintaining a short support.
- Snakes with singularities: We can find in nature cells that are mostly smooth, but present some sharp kinks. This can be handled in our current model by stacking several control points at the place where kink is located. We would like to investigate how the multiplicity of control points affects the convergence of the optimization algorithm, and how this can be handled efficiently.
- Prior shapes and atlases: In regular biomedical segmentation problems, the shape of the target to segment is usually known up to some extend. We would like to provide a library with several shapes that our snake can use as reference models. The natural way to use the reference model would be by the introduction of an energy function that penalizes deformations from the reference shapes.
- Extend the supported snake topologies: We have limited ourselves to blob-like objects in this thesis. Therefore, the 2D snake curves presented in Chapter 4 are closed curves, and the 3D snake surfaces presented in Chapter 5 exhibit a sphere-like topology. We would like to account for other kind of topological properties such as open curves in 2D, or tori in 3D.

- Usage of GPUs: With the recent explosion of computation power provided by GPUs, we think it would be interesting to move part of the computation load to these kind of devices. Specially the part related to the user display.
- Application of the snakes to more projects in life-sciences: Our active contours have shown a remarkable capability to adapt to many different biomedical structures: endocardial heart walls, aorta cross sections, HeLa cell nuclei, water droplets, spleens, murine brain cells, olfactory glomerulus, yeast cells, and *Drosophila* wings and bodies. We believe that our active contours can be still used in many more applications by designing customized the energy functions.
- Moving on from on-line tracking: The current tracking algorithms used in bioimaging and computer vision are based in earlier algorithms from the radar community, where targets are tracked in real-time. This had led to the bioimaging community to believe that real-time on-line tracking is the only possibility. However, in biology, the acquisition and analysis stages take place in different moments in time. Therefore, all the time-lapse sequences of images are usually available at the time of the analysis. We would like to investigate the implications of using the whole time-lapse sequence during the tracking.

Bibliography

- [1] Z. Demou, “Time-lapse analysis and microdissection of living 3D melanoma cell cultures for genomics and proteomics,” *Biotechnology and Bioengineering*, vol. 101, pp. 307–316, October 2008.
- [2] K. Kirkegaard, I. Agerholm, and H. Ingerslev, “Time-lapse monitoring as a tool for clinical embryo assessment,” *Human Reproduction*, vol. 27, pp. 1277–1285, May 2012.
- [3] C. Zimmer, B. Zhang, A. Dufour, A. Thébaud, S. Berlemont, V. Meas-Yedid, and J.-C. Olivo-Marin, “On the digital trail of mobile cells,” *IEEE Signal Processing Magazine*, vol. 23, pp. 54–62, January 2006.
- [4] B. Wilt, L. Burns, E. Ho, K. Ghosh, E. Mukamel, and M. Schnitzer, “Advances in light microscopy for neuroscience,” *Annual Review of Neuroscience*, vol. 32, pp. 435–506, July 2009.
- [5] D. Schmolze, C. Standley, K. Fogarty, and A. Fischer, “Advances in microscopy techniques,” *Archives of Pathology and Laboratory Medicine*, vol. 135, pp. 255–263, February 2011.
- [6] F. Blais, “Review of 20 years of range sensor development,” *Journal of Electronic Imaging*, vol. 13, pp. 231–243, January 2004.
- [7] O. Levi, T. Lee, M. Lee, S. Smith, and J. Harris, “Integrated semiconductor optical sensors for cellular and neural imaging,” *Applied Optics*, vol. 46, pp. 1881–1889, April 2007.
- [8] D. Gramotnev and S. Bozhevolnyi, “Plasmonics beyond the diffraction limit,” *Nature Photonics*, vol. 4, pp. 83–91, February 2010.
- [9] B. Pollok and R. Heim, “Using GFP in FRET-based applications,” *Trends in Cell Biology*, vol. 9, pp. 57–60, February 1999.
- [10] I. Medintz, H. Uyeda, E. Goldman, and H. Mattoussi, “Quantum dot bioconjugates for imaging, labelling and sensing,” *Nature Materials*, vol. 4, pp. 435–446, June 2005.

- [11] B. Giepmans, S. Adams, M. Ellisman, and R. Tsien, "The fluorescent toolbox for assessing protein location and function," *Science*, vol. 312, pp. 217–224, April 2006.
- [12] A. Miyawaki, "Visualization of the spatial and temporal dynamics of intracellular signaling," *Developmental Cell*, vol. 4, pp. 295–305, March 2003.
- [13] N. Wu, X. Zhou, D. Czajkowsky, M. Ye, D. Zeng, Y. Fu, C. Fan, J. Hu, and B. Li, "In situ monitoring of single molecule binding reactions with time-lapse atomic force microscopy on functionalized DNA origami," *Nanoscale*, vol. 3, pp. 2481–2484, June 2011.
- [14] T. Mayhew and J. Lucocq, "Developments in cell biology for quantitative immunoelectron microscopy based on thin sections: A review," *Histochemistry and Cell Biology*, vol. 130, pp. 299–313, August 2008.
- [15] D. Muzzey and A. Van Oudenaarden, "Quantitative time-lapse fluorescence microscopy in single cells," *Annual Review of Cell and Developmental Biology*, vol. 25, pp. 301–327, November 2009.
- [16] S. Maerkl, "Integration column: Microfluidic high-throughput screening," *Integrative Biology*, vol. 1, pp. 19–29, January 2009. 11.
- [17] M. Oheim, "Advances and challenges in high-throughput microscopy for live-cell subcellular imaging," *Expert Opinion on Drug Discovery*, vol. 6, pp. 1299–1315, December 2011.
- [18] G. Myers, "Why bioimage informatics matters," *Nature Methods*, vol. 6, pp. 659–660, July 2012.
- [19] Q. Wu, F. Merchant, and K. Castleman, *Microscope Image Processing*. Elsevier, first ed., 2008.
- [20] M. Bertero and E. Pike, "Signal processing for linear instrumental systems with noise: A general theory with illustrations from optical imaging and light scattering problems," in *Handbook of Statistics*, vol. 10, pp. 1–46, Amsterdam, The Netherlands: Elsevier Science, 1993.
- [21] A. Griffa, N. Garin, and D. Sage, "Comparison of deconvolution software in 3D microscopy: A user point of view—Part 1," *G.I.T. Imaging & Microscopy*, vol. 12, pp. 43–45, March 2010.
- [22] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [23] D. Kendall, D. Barden, and H. Carne, T.K. Le, *Shape and Shape Theory*. Wiley, first ed., 1999.

-
- [24] A. Bovik, *The Essential Guide to Image Processing*. Academic Press, first ed., 2009.
- [25] R. Ghosh and W. Webb, "Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules," *Biophysical Journal*, vol. 66, pp. 1301–1318, May 1994.
- [26] N. Harder, F. Mora-Bermúdez, W. Godinez, J. Ellenberg, R. Eils, and K. Rohr, "Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences," in *Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, vol. 4190/2006, (Copenhagen, Denmark), pp. 840–848, October 1-6 2006.
- [27] M. Wang, X. Zhou, F. Li, J. Huckins, R. King, and S. Wong, "Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy," *Bioinformatics*, vol. 24, pp. 94–101, January 2008.
- [28] R. Baron, "Mechanisms of human facial recognition," *International Journal of Man-Machine Studies*, vol. 15, pp. 137–178, August 1981.
- [29] R. Brunelli and T. Poggio, "Template matching: Matched spatial filters and beyond," *Pattern Recognition*, vol. 30, pp. 751–768, May 1997.
- [30] K. Chang, K. Bowyer, and P. Flynn, "Multiple nose region matching for 3D face recognition under varying facial expression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1695–1700, October 2006.
- [31] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, second ed., 2003.
- [32] V. Grau, A. Mewes, M. Alcañiz, R. Kikinis, and S. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, 2004.
- [33] S. Li, "Markov random field models in computer vision," in *Proceedings of the Third European Conference on Computer Vision (ECCV'94)*, (Stockholm, Sweden), pp. 361–370, May 2-6, 1994.
- [34] J. Rajapakse, J. Giedd, and J. Rapoport, "Statistical approach to segmentation of single-channel cerebral MR images," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 176–186, April 1997.
- [35] T. Pappas and N. Murray Hill, "An adaptive clustering algorithm for image segmentation," *IEEE Transactions on Signal Processing*, vol. 40, pp. 901–914, April 1992.

- [36] K. Held, E. Kops, B. Krause, I. Wells, W.M., R. Kikinis, and H.-W. Muller-Gartner, "Markov random field segmentation of brain MR images," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 878–886, December 1997.
- [37] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.
- [38] A. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, vol. 71, no. 2, pp. 109–129, 1998.
- [39] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681–685, June 2001.
- [40] L. Zhu, Y. Chen, and A. Yuille, "Learning a hierarchical deformable template for rapid deformable object parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1029–1043, June 2010.
- [41] A. Britto and G. Ravindran, "Review of deformable curves-A retro analysis," *Information Technology Journal*, vol. 6, pp. 26–36, January 2007.
- [42] D. Jayadevappa, S. Kumar, and D. Murty, "Medical image segmentation algorithms using deformable models: A review," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 28, pp. 248–255, May 2011.
- [43] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, January 1987.
- [44] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, pp. 359–369, March 1998.
- [45] J. Gao, A. Kosaka, and A. Kak, "A deformable model for human organ extraction," in *Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP'98)*, (Chicago, IL, USA), pp. 323–327, October 4–7, 1998.
- [46] T. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Computer Vision, Graphics and Image Processing*, vol. 48, pp. 357–393, December 1989.
- [47] L. Staib and J. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 1061–1075, November 1992.
- [48] A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 859–870, December 1996.

-
- [49] M. Figueiredo, J. Leitão, and A. Jain, “Unsupervised contour representation and estimation using B-splines and a minimum description length criterion,” *IEEE Transactions on Image Processing*, vol. 9, pp. 1075–1087, June 2000.
- [50] P. Brigger, J. Hoeg, and M. Unser, “B-Spline snakes: A flexible tool for parametric contour detection,” *IEEE Transactions on Image Processing*, vol. 9, pp. 1484–1496, September 2000.
- [51] M. Gebhard, J. Mattes, and R. Eils, “An active contour model for segmentation based on cubic B-splines and gradient vector flow,” in *Proceedings of the 2001 Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’01)*, (Utrecht, The Netherlands), pp. 1373–1375, October 14-17, 2001.
- [52] V. Kindratenko, “On using functions to describe the shape,” *Journal of Mathematical Imaging and Vision*, vol. 18, pp. 225–245, May 2003.
- [53] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Academic Press, fourth ed., 1997.
- [54] R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric and Geometric Modeling*. Morgan Kaufmann, first ed., 1995.
- [55] L. Piegl and W. Tiller, *The NURBS Book*. Springer Berlin Heidelberg, second ed., 2010.
- [56] C. Zahn and R. Roskies, “Fourier descriptors for plane closed curves,” *IEEE Transactions on Computers*, vol. C-21, pp. 269–281, March 1972.
- [57] M. Jacob, T. Blu, and M. Unser, “Efficient energies and algorithms for parametric snakes,” *IEEE Transactions on Image Processing*, vol. 13, pp. 1231–1244, September 2004.
- [58] R. Delgado-Gonzalo, P. Thévenaz, and M. Unser, “Exponential splines and minimal-support bases for curve representation,” *Computer Aided Geometric Design*, vol. 29, pp. 109–128, February 2012.
- [59] T. McInerney and D. Terzopoulos, “T-snakes: Topology adaptive snakes,” *Medical Image Analysis*, vol. 4, pp. 73–91, June 2000.
- [60] F. Precioso and M. Barlaud, “B-Spline active contour with handling of topology changes for fast video segmentation,” *EURASIP Journal on Applied Signal Processing*, vol. 2002, pp. 555–560, June 2002.
- [61] R. Malladi, J. Sethian, and B. Vemuri, “Shape modeling with front propagation: A level set approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 158–175, February 1995.

- [62] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [63] K. Zhang, L. Zhang, H. Song, and W. Zhou, "Active contours with selective local or global segmentation: A new formulation and level set method," *Image and Vision Computing*, vol. 28, pp. 668–676, April 2010.
- [64] K. Zhang, H. Song, and L. Zhang, "Active contours driven by local image fitting energy," *Pattern Recognition*, vol. 43, pp. 1199–1206, April 2010.
- [65] S. Osher and J. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, November 1988.
- [66] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, first ed., 2006.
- [67] L. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, pp. 211–218, March 1991.
- [68] M. Jacob, T. Blu, and M. Unser, "A unifying approach and interface for spline-based snakes," in *Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'01)*, (San Diego, CA, USA), pp. 340–347, February 19-22, 2001.
- [69] P. Thévenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 382–393, February 2011.
- [70] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, pp. 266–277, February 2001.
- [71] M. Kamandar and S. Seyedin, "Procrustes - Based shape prior for parametric active contours," in *Proceedings of the International Conference on Machine Vision, (ICMV'07)*, (Islamabad, Pakistan), pp. 135–140, December 28-29, 2007.
- [72] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, "Fast parametric snakes for 3D microscopy," in *Proceedings of the Ninth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'12)*, (Barcelona, Spain), pp. 852–855, May 2-5, 2012.
- [73] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, July 1989.
- [74] A. Tsai, A. Yezzi Jr., and A. Willsky, "Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, vol. 10, pp. 1169–1186, August 2001.

- [75] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, pp. 109–131, November 2006.
- [76] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, pp. 151–167, June 2007.
- [77] A. Cardona and P. Tomancak, "Current challenges in open-source bioimage informatics," *Nature Methods*, vol. 9, pp. 661–665, July 2012.
- [78] A. Carpenter, L. Kametsky, and K. Eliceiri, "A call for bioimaging software usability," *Nature Methods*, vol. 6, pp. 666–670, July 2012.
- [79] D. Bolchini, A. Finkelstein, V. Perrone, and S. Nagl, "Better bioinformatics through usability analysis," *Bioinformatics*, vol. 25, pp. 406–412, February 2009.
- [80] M. Linkert, C. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. Eliceiri, and J. Swedlow, "Metadata matters: Access to image data in the real world," *Journal of Cell Biology*, vol. 189, pp. 777–782, May 2010.
- [81] F. De Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin, "Icy: An open bioimage informatics platform for extended reproducible reresearch," *Nature Methods*, vol. 9, pp. 690–696, July 2012.
- [82] M. Abràmoff, P. Magalhães, and S. Ram, "Image processing with ImageJ," *Biophotonics International*, vol. 11, pp. 36–41, July 2004.
- [83] T. Collins, "ImageJ for microscopy," *BioTechniques*, vol. 43, pp. 25–30, July 2007.
- [84] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: An open-source platform for biological-image analysis," *Nature Methods*, vol. 9, pp. 676–682, July 2012.
- [85] A. Edelstein, N. Amodaj, K. Hoover, R. Vale, and N. Stuurman, "Computer control of microscopes using μ manager," in *Current Protocols in Molecular Biology*, John Wiley & Sons, Inc., 2010.
- [86] M. Lamprecht, D. Sabatini, and A. Carpenter, "Cellprofiler™: Free, versatile software for automated biological image analysis," *BioTechniques*, vol. 42, pp. 71–75, January 2007.

- [87] C. Schneider, W. Rasband, and K. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, pp. 671–675, July 2012.
- [88] W. Böhm, G. Farin, and J. Kahmann, "A survey of curve and surface methods in CAGD," *Computer Aided Geometric Design*, vol. 1, pp. 1–60, July 1984.
- [89] B. Pham, "Offset curves and surfaces: A brief survey," *Computer-Aided Design*, vol. 24, pp. 223–229, April 1992.
- [90] C. Micchelli and H. Prautzsch, "Uniform refinement of curves," *Linear Algebra and its Applications*, vol. 114–115, pp. 841–870, March–April 1989.
- [91] G. Deslauriers and S. Dubuc, *Fractals, Dimensions Non Entières et Applications*, ch. Interpolation Dyadique, pp. 44–55. Masson, Paris, second ed., 1991.
- [92] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann, first ed., 2002.
- [93] G. Farin, G. Rein, N. Sapidis, and A. Worsley, "Fairing cubic B-spline curves," *Computer Aided Geometric Design*, vol. 4, pp. 91–103, July 1987.
- [94] L. Piegl, "On NURBS: A survey," *IEEE Computer Graphics and Applications*, vol. 11, pp. 55–71, January 1991.
- [95] L. Schumaker, *Spline Functions: Basic Theory*. Cambridge University Press, third ed., 2007.
- [96] A. Cavaretta, W. Dahmen, and C. Micchelli, *Stationary Subdivision*. American Mathematical Society, first ed., 1991.
- [97] N. Dyn, J. Gregory, and D. Levin, "Analysis of uniform binary subdivision schemes for curve design," *Constructive Approximation*, vol. 7, pp. 127–147, December 1991.
- [98] N. Dyn, "Subdivision schemes in computer-aided geometric design," in *Advances in Numerical Analysis* (W. Light, ed.), vol. 2, pp. 36–104, New York: Clarendon Press, Oxford, 1992.
- [99] G. Derfel, N. Dyn, and D. Levin, "Generalized refinement equations and subdivision processes," *Journal of Approximation Theory*, vol. 80, pp. 272–297, February 1995.
- [100] N. Dyn and D. Levin, "Analysis of asymptotically equivalent binary subdivision schemes," *Journal of Mathematical Analysis and Applications*, vol. 193, pp. 594–621, July 1995.
- [101] J. Zhang, "C-curves: An extension of cubic curves," *Computer Aided Geometric Design*, vol. 13, pp. 199–217, April 1996.

-
- [102] J. Zhang and F.-L. Krause, "Extending cubic uniform B-splines by unified trigonometric and hyperbolic basis," *Graphical Models*, vol. 67, pp. 100–119, March 2005.
- [103] M. Sabin and N. Dodgson, "A circle-preserving variant of the four-point subdivision scheme," in *Mathematical Methods for Curves and Surfaces*, (Tromsø, Norway), pp. 275–286, July 1-6, 2004.
- [104] C. Deng and G. Wang, "Incenter subdivision scheme for curve interpolation," *Computer Aided Geometric Design*, vol. 27, pp. 48–59, January 2010.
- [105] L. Romani, "A circle-preserving C2 Hermite interpolatory subdivision scheme with tension control," *Computer Aided Geometric Design*, vol. 27, pp. 36–47, January 2010.
- [106] C. Beccari, G. Casciola, and L. Romani, "A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics," *Computer Aided Geometric Design*, vol. 24, pp. 1–9, January 2007.
- [107] C. Beccari, G. Casciola, and L. Romani, "Shape controlled interpolatory ternary subdivision," *Applied Mathematics and Computation*, vol. 215, pp. 916–927, October 2009.
- [108] D. Sunita and P. Shunmugaraj, "An approximating C2 non-stationary subdivision scheme," *Computer Aided Geometric Design*, vol. 26, no. 7, pp. 810–821, 2009.
- [109] C. Conti and L. Romani, "Affine combination of B-spline subdivision masks and its non-stationary counterparts," *BIT Numerical Mathematics*, vol. 50, pp. 269–299, June 2010.
- [110] C. Conti, L. Gemignani, and L. Romani, "From approximating to interpolatory non-stationary subdivision schemes with the same generation properties," *Advances in Computational Mathematics*, vol. 35, pp. 217–241, November 2011.
- [111] N. Dyn, D. Levin, and A. Luzzatto, "Exponentials reproducing subdivision schemes," *Foundations of Computational Mathematics*, vol. 3, pp. 187–206, March 2008.
- [112] L. Romani, "From approximating subdivision schemes for exponential splines to high-performance interpolating algorithms," *Journal of Computational and Applied Mathematics*, vol. 224, pp. 383–396, February 2009.
- [113] A. Ron, "Factorization theorems for univariate splines on regular grids," *Israel Journal of Mathematics*, vol. 70, pp. 48–68, February 1990.
- [114] W. Boehm, "On de Boor-like algorithms and blossoming," *Computer Aided Geometric Design*, vol. 5, pp. 71–79, June 1988.

- [115] P. Costantini, "An algorithm for computing shape-preserving interpolating splines of arbitrary degree," *Journal of Computational and Applied Mathematics*, vol. 22, pp. 89–136, April 1988.
- [116] M. Unser, "Sampling—50 Years after Shannon," *Proceedings of the IEEE*, vol. 88, pp. 569–587, April 2000.
- [117] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory," *Numerical Functional Analysis and Optimization*, vol. 15, no. 1-2, pp. 1–21, 1994.
- [118] C. de Boor and R. DeVore, "Partitions of unity and approximation," *Proceedings of the American Mathematical Society*, vol. 93, pp. 705–709, April 1985.
- [119] M. Jacob, T. Blu, and M. Unser, "Sampling of periodic signals: A quantitative error analysis," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1153–1159, May 2002.
- [120] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2783–2795, October 1999.
- [121] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method," in *Constructive Aspect of Functional Analysis*, pp. 796–830, Rome, Italy: Edizioni Cremonese, 1971.
- [122] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Transactions on Signal Processing*, vol. 51, pp. 470–483, February 2003.
- [123] C. Vonesch, T. Blu, and M. Unser, "Generalized Daubechies wavelet families," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4415–4429, September 2007.
- [124] M. Unser and T. Blu, "Cardinal exponential splines: Part I—Theory and filtering algorithms," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1425–1438, April 2005.
- [125] H. Späth, "Exponential spline interpolation," *Computing*, vol. 4, pp. 225–233, September 1969.
- [126] C. de Boor and A. Ron, "The exponentials in the span of the multiinteger translates of a compactly supported function: Quasiinterpolation and approximation order," *Journal of the London Mathematical Society*, vol. 2, pp. 519–535, November 1992.
- [127] L. Schwarz, *Théorie des Distributions*. Hermann, second ed., 1966.
- [128] C. de Boor, R. A. DeVore, and A. Ron, "On the construction of multivariate (pre)wavelets," *Constructive Approximation*, vol. 9, no. 2–3, pp. 123–166, 1993.

-
- [129] T. Blu, P. Thévenaz, and M. Unser, “MOMS: Maximal-order interpolation of minimal support,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1069–1080, July 2001.
- [130] I. Khalidov and M. Unser, “From differential equations to the construction of new wavelet-like bases,” *IEEE Transactions on Signal Processing*, vol. 54, pp. 1256–1267, April 2006.
- [131] I. Schoenberg, “On trigonometric spline interpolation,” *Indiana University Mathematics Journal*, vol. 13, no. 5, pp. 795–825, 1964.
- [132] D. Von Seggern, *CRC Standard Curves and Surfaces*. CRC Press, first ed., 1993.
- [133] J. Blokland, A. Vossepoel, A. Bakker, and E. Pauwels, “Delineating elliptical objects with an application to cardiac scintigrams,” *IEEE Transactions on Medical Imaging*, vol. 6, pp. 57–66, March 1987.
- [134] P. Lipson, A. Yuille, D. O’Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal, “Deformable templates for feature extraction from medical images,” in *Proceedings of the First European Conference on Computer Vision (ECCV’90)*, (Antibes, France), pp. 413–417, April 23-27, 1990.
- [135] I. Adame, R. Van Der Geest, D. Bluemke, J. Lima, J. Reiber, and B. Lelieveldt, “Automatic vessel wall contour detection and quantification of wall thickness in in-vivo MR images of the human aorta,” *Journal of Magnetic Resonance Imaging*, vol. 24, pp. 595–602, September 2006.
- [136] J. Guerrero, S. Salcudean, J. McEwen, B. Masri, and S. Nicolaou, “Real-time vessel segmentation and tracking for ultrasound imaging applications,” *IEEE Transactions on Medical Imaging*, vol. 26, pp. 1079–1090, August 2007.
- [137] S. Tsuji and F. Matsumoto, “Detection of ellipses by a modified Hough transformation,” *IEEE Transactions on Computers*, vol. 27, pp. 777–781, August 1978.
- [138] N. Bennett, R. Burrige, and N. Saito, “A method to detect and characterize ellipses using the Hough transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 652–657, July 1999.
- [139] I. Adame, R. van der Geest, B. Wasserman, M. Mohamed, J. Reiber, and B. Lelieveldt, “Automatic segmentation and plaque characterization in atherosclerotic carotid artery MR images,” *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 16, pp. 227–234, April 2004.
- [140] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. Garland Science, fifth ed., 2007.

- [141] A. Stalder, G. Kulik, D. Sage, L. Barbieri, and P. Hoffmann, "A snake-based approach to accurate determination of both contact points and contact angles," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 286, pp. 92–103, September 2006.
- [142] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, pp. 22–38, November 1999.
- [143] F. Precioso and M. Barlaud, "B-spline active contours for fast video segmentation," in *Proceedings 2001 International Conference on Image Processing*, (Thessaloniki, Greece), pp. 777–780, October, 7-10, 2001.
- [144] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, third ed., 1986.
- [145] A. Andreopoulos and J. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI," *Medical Image Analysis*, vol. 12, pp. 335–357, June 2008.
- [146] P. Andrey and T. Boudier, "Adaptive active contours," in *First ImageJ User & Developer Conference*, (Luxembourg), pp. 1–5, May 18-19, 2006.
- [147] H. Delingette, "General object reconstruction based on simplex meshes," *International Journal of Computer Vision*, vol. 32, pp. 111–146, September 1999.
- [148] A. Dufour, R. Thibeaux, E. Labruyere, N. Guillen, and J.-C. Olivo-Marin, "3-D Active meshes: Fast discrete deformable models for cell tracking in 3-D time-lapse microscopy," *IEEE Transactions on Image Processing*, vol. 20, pp. 1925–1937, July 2011.
- [149] Q. Duan, E. Angelini, and A. Laine, "Surface function actives," *Journal of Visual Communication and Image Representation*, vol. 20, pp. 478–490, October 2009.
- [150] D. Barbosa, T. Dietenbeck, J. Schaerer, J. D'hooge, D. Friboulet, and O. Bernard, "B-Spline explicit active surfaces: An efficient framework for real-time 3-D region-based segmentation," *IEEE Transactions on Image Processing*, vol. 21, pp. 241–251, January 2012.
- [151] P. Dierckx, "Algorithms for smoothing data on the sphere with tensor product splines," *Computing*, vol. 32, pp. 319–342, April 1984.
- [152] R. Gmelig Meyling and P. Pfluger, "B-spline approximation of a closed surface," *IMA Journal of Numerical Analysis*, vol. 7, pp. 73–96, January 1987.

- [153] L. Schumaker and C. Traas, "Fitting scattered data on sphere-like surfaces using spherical splines," *Numerische Mathematik*, vol. 60, pp. 133–144, January 1991.
- [154] M. Do Carmo, *Differential Geometry of Curves and Surfaces*. Prentice-Hall, first ed., 1976.
- [155] L. Boydston, T. Armstrong, and F. Bookstein, "A comparison of three dimensional maximum reach estimation techniques," *Journal of Biomechanics*, vol. 13, no. 8, pp. 717–724, 1980.
- [156] C. Hoffmann, *Geometric and Solid Modeling*. Morgan Kaufmann, first ed., 1989.
- [157] L. Zhang, E. Hoffman, and J. Reinhardt, "Atlas-driven lung lobe segmentation in volumetric X-ray CT images," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1–16, January 2006.
- [158] L. Roux, K. Benchenane, J. Rothstein, G. Bonvento, and C. Giaume, "Plasticity of astroglial networks in olfactory glomeruli," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 45, pp. 18442–18446, 2011.
- [159] D. Cremers and C. Schnörr, "Statistical shape knowledge in variational motion segmentation," *Image and Vision Computing*, vol. 21, pp. 77–86, January 2003.
- [160] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Transactions on Image Processing*, vol. 14, pp. 1396–1410, September 2005.
- [161] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1805–1918, November 2005.
- [162] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, "Feature-aided particle tracking," in *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'08)*, (San Diego, CA, USA), pp. 1796–1799, October 12–15, 2008.
- [163] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, "Multiple hypothesis tracking in microscopy images," in *Proceedings of the 6th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'09)*, (Boston, MA, USA), pp. 1346–1349, June 28–July 1, 2009.
- [164] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J.-C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy

- with parametric active contours: A tool for cell-based drug testing,” *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1212–1221, January 2002.
- [165] K. Miura, “Tracking movement in cell biology,” *Advances in Biochemical Engineering/Biotechnology*, vol. 95/2005, pp. 267–295, June 2004.
- [166] J. Cui, N. Ray, S. Acton, and Z. Lin, “An affine transformation invariance approach to cell tracking,” *Computerized Medical Imaging and Graphics*, vol. 32, pp. 554–565, October 2008.
- [167] O. Dzyubachyk, W. van Cappellen, J. Essers, W. Niessen, and E. Meijering, “Advanced level-set-based cell tracking in time-lapse fluorescence microscopy,” *IEEE Transactions on Medical Imaging*, vol. 29, pp. 852–867, March 2010.
- [168] P. Vallotton, A. Ponti, C. Waterman-Storer, E. Salmon, and G. Danuser, “Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: A fluorescent speckle microscopy study,” *Biophysical Journal*, vol. 85, pp. 1289–1306, August 2003.
- [169] I. Sbalzarini and P. Koumoutsakos, “Feature point tracking and trajectory analysis for video imaging in cell biology,” *Journal of Structural Biology*, vol. 151, pp. 182–195, August 2005.
- [170] M. Isard and A. Blake, “ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework,” in *Proceedings of the 5th European Conference on Computer Vision (ECCV’98)*, (Freiburg, Germany), pp. 893–908, June 2-6, 1998.
- [171] M. Bray, E. Koller-Meier, and L. V. Gool, “Smart particle filtering for high-dimensional tracking,” *Computer Vision and Image Understanding*, vol. 106, pp. 116–129, April 2007.
- [172] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, “Tracking deforming objects using particle filtering for geometric active contours,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1470–1475, August 2007.
- [173] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, February 2002.
- [174] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. Springer-Verlag, first ed., 2005.
- [175] D. Barber and A. Cemgil, “Graphical models for time-series,” *IEEE Signal Processing Magazine*, vol. 27, pp. 18–28, November 2010.

-
- [176] K. Li, E. Miller, M. Chen, T. Kanade, L. Weiss, and P. Campbell, "Cell population tracking and lineage construction with spatiotemporal context," *Medical Image Analysis*, vol. 12, pp. 546–566, October 2008.
- [177] S. Ali and M. Shah, "Floor fields for tracking in high density crowd scenes," in *Proceedings of the 10th European Conference on Computer Vision (ECCV'08)*, (Marseille, France), pp. 1–14, October 12-18, 2008.
- [178] J. Bondy and U. Murty, *Graph Theory with Applications*. Elsevier Science Ltd, first ed., 1976.
- [179] M. Karpinski and W. Rytter, *Fast Parallel Algorithms for Graph Matching Problems*. Oxford University Press, first ed., 1998.
- [180] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.
- [181] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proceedings of the IEEE 1981 National Telecommunications Conference: Innovative Telecommunications - Key to the Future (NTC '81)*, (New Orleans, LA, USA), pp. G5.3/1–5, November 29-December 3 1981.
- [182] N. Dénervaud, *A Microfluidic Live-Cell Imaging Platform to Study Large Collections of Microbial Genotypes*. PhD thesis, Swiss Federal Institute of Technology Lausanne (EPFL), November 21, 2012.
- [183] P. Michel, R. Delgado-Gonzalo, M. Hubert, L. Baldi, D. Hacker, M. Unser, and F. Wurm, "Assessment of chromosomal size variation in CHO cells," in *Proceedings of the MipTec and BioValley Life Sciences Week (MTBVLS'12)*, (Basel BS, Switzerland), p. 117, September 24-27, 2012.

Curriculum Vitæ

Ricard Delgado Gonzalo



Personal Data

Birth date 13. August 1983
Nationality Spanish
E-Mail ricard.delgado@gmail.com
Web <http://www.ricarddelgado.com/>

Professional Experience

2008–Present **École Polytechnique Fédérale de Lausanne**, (*full-time - 4 years 8 months*).

Role: Project manager leading a research team developing bioimage analysis open-source software.

Achievements: Released two applications targeted to life science researchers. Invited talk to BioImage Informatics 2012.

Role: Researcher designing and developing a high throughput screening system for cell analysis in collaboration with the LBNC laboratory for the Swiss National Science Foundation.

Achievements: Fourteen scientific papers, eleven of which I presented in high impact international conferences.

Role: Teaching assistant in two master level courses, including lectures and programming labs.

Achievements: Defined and supervised twelve semester and master student projects, all of which involved the design and implementation of software solutions for image processing applications. Some projects were conducted in collaboration with other EPFL labs and external companies (software and source code is freely available).

2007–2008 **Deloitte**, (*full-time - 6 months*).

Role: Auditor in the ERS (Enterprise Risk Services) division.

Achievements: Supervision of a time-challenged migration project of the core-banking platform and databases of a Spanish savings bank.

2007 **Illinois Institute of Technology**, (*internship - 7 months*).

Role: Research assistant conceiving, designing and developing a new system to accelerate medical tomography using the new GPU technology.

Achievements: Three scientific papers presented in high impact international conferences. Initial benchmarks showed an improvement of up to 30% in efficiency and resource consumption. I defined the subsequent steps for the extension of the project.

2006–2007 **Institut de Robòtica i Informàtica Industrial**, (*internship - 8 months*).

Role: Research assistant designing new benchmarks for osteoporosis detection from tomographic data in collaboration with MD's from Universidad de Alicante (UA).

Achievements: I initiated the project, and facilitated the knowledge transfer to the medical team at UA for the continuation of the project.

- 2006–2007 **Universitat Politècnica de Catalunya**, (*internship - 11 months*).
Role: Research assistant working in the Antennas group on a new methodology for microwave imaging.
Achievements: Autodidact in a new and fast-evolving area establishing the basis for further research.

Education & Diplomas

- 2013 **Certified Associate in Project Management**, *Project Management Institute*, (Expected).
2008–2013 **PhD**, *École Polytechnique Fédérale de Lausanne, Switzerland*, specialized in biomedical image processing..
2002–2006 **Mathematics**, *Universitat Politècnica de Catalunya, Spain*, specialized in applied mathematics.
2001–2007 **Telecommunication Engineering**, *Universitat Politècnica de Catalunya, Spain*, final diploma project performed in the Illinois Institute of Technology, Chicago, USA.

Skills

- Soft Skills Creative, organized, meticulous, analytic, proactive, team player, empathic, negotiation skills, ability to summarize, adaptability of speech to audiences from different backgrounds, fast learner, capable of dealing with underdetermined problems.
- IT Java SE (developing image analysis apps for ImageJ and Icy), C, Objective-C (developed the iPhone/iPad app named iMondrian with over 300 downloads in the first 48 hours), OOAD, UML, VBA, Matlab, Maple, ImageJ, ICY, VTK, LabVIEW, Eclipse, SVN (Subclipse), Xcode, MSOffice/LibreOffice/iWork.

Languages

- Spanish & Catalan Native bilingual.
English C2 level.
French B1 level.
German A1 level.

Honors & Awards

- Nominated to the best PhD thesis award of EPFL 2013, (Decision pending).
Best poster award in the IEEE EMBS International Summer School on Biomedical Imaging 2012.
Best student paper award in the IEEE International Symposium on Biomedical Imaging 2011.

Publications

Papers Under Review

- N. Dénervaud, J. Becker, **R. Delgado-Gonzalo**, M. Unser, F. Naef, S.J. Maerkl, "A Microchemostat Array Enables the Spatio-Temporal Analysis of the Yeast Proteome in Flux".
- R. Delgado-Gonzalo**, N. Chenouard, M. Unser, "Spline-Based Deforming Ellipsoids for Interactive 3D Bioimage Segmentation".
- R. Delgado-Gonzalo**, M. Unser, "Spline-Based Framework for Interactive Segmentation in Biomedical Imaging".
- P.O. Michel, **R. Delgado-Gonzalo**, M. Hubert, L. Baldi, D.L. Hacker, M. Unser, F.M. Wurm, "Assessment of Chromosomal Size Variation in CHO Cells".

Journal Papers

R. Delgado-Gonzalo, P. Thévenaz, C.S. Seelamantula, M. Unser, "Snakes with an Ellipse-Reproducing Property," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1258-1271, March 2012.

R. Delgado-Gonzalo, P. Thévenaz, M. Unser, "Exponential Splines and Minimal-Support Bases for Curve Representation," *Computer Aided Geometric Design*, vol. 29, no. 2, pp. 109-128, February 2012.

P. Thévenaz, **R. Delgado-Gonzalo**, M. Unser, "The Ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382-393, February 2011.

Conference Papers

R. Delgado-Gonzalo, N. Chenouard, M. Unser, "Fast Parametric Snakes for 3D Microscopy," *Proceedings of the Ninth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'12)*, Barcelona, Spain, May 2-5, 2012.

P.D. Tafti, **R. Delgado-Gonzalo**, A. Stalder, M. Unser, "Variational Enhancement and Denoising of Flow Field Images," *Proceedings of the Eighth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'11)*, Chicago, USA, March 30-April 2, 2011. **Best Paper Award**.

R. Delgado-Gonzalo, N. Chenouard, M. Unser, "A New Hybrid Bayesian-Variational Particle Filter with Applications to Mitotic Cell Tracking," *Proceedings of the Eighth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'11)*, Chicago, USA, March 30-April 2, 2011.

R. Delgado-Gonzalo, N. Déneraud, S. Maerkl, M. Unser, "Multi-Target Tracking of Packed Yeast Cells," *Proceedings of the Seventh IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'10)*, Rotterdam, The Netherlands, April 14-17, 2010, pp. 544-547.

P.D. Tafti, **R. Delgado-Gonzalo**, A. Stalder, M. Unser, "Fractal Modelling and Analysis of Flow-Field Images," *Proceedings of the Seventh IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'10)*, Rotterdam, The Netherlands, April 14-17, 2010, pp. 49-52.

R. Delgado-Gonzalo, P.D. Tafti, M. Unser, "Fractional Laplacian pyramids," *Proceedings of the 2009 IEEE International Conference on Image Processing (ICIP'09)*, Cairo, Egypt, November 7-11, 2009, pp. 3809-3812.

R. Delgado Gonzalo, J.G. Brankov, "Mesh Model 2D Reconstruction Operator for SPECT," *Proceedings of the SPIE on Medical Imaging 2008: Physics of Medical Imaging*, San Diego, CA, USA, February 18, 2008, vol. 6913, pp. 69132L-1/69132L-8.

J.G. Brankov, **R. Delgado Gonzalo**, Y. Yang, M. Jin, M.N. Wernick, "Progress in Mesh Based Spatio-Temporal Reconstruction," *Proceedings of the SPIE on Computational Imaging VI*, San José, CA, USA, January 28, 2008, vol. 6814, pp. 68140V-1/68140V-9.

R. Delgado Gonzalo, J.G. Brankov, "Mesh Model Based Projection Operator for Emission Tomography," *IEEE Nuclear Science Symposium Conference Record, 2007 (NSS'07)*, Honolulu, Hawaii, USA, October 27-November 3, 2007, vol. 4, pp. 2760-2763.

Conference Abstracts

P.O. Michel, **R. Delgado-Gonzalo**, M. Hubert, L. Baldi, D.L. Hacker, M. Unser, F.M. Wurm, "Assessment of Chromosomal Size Variation in CHO Cells," *Proceedings of the MipTec and BioValley Life Sciences Week (MTBVL'S'12)*, Basel BS, Switzerland, September 24-27, 2012, paper no. P181, pp. 117.

M. Unser, **R. Delgado**, "Active Cells: An Exponential Spline Framework for the Interactive Segmentation of Microscopy Images in 2-D and 3-D," *Invited talk, Proceedings of BioImage Informatics 2012 (BI'12)*, Dresden, Germany, September 16-19, 2012, pp. 43.

R. Delgado-Gonzalo, M. Unser, "Exponential B-Splines and the Design of Active Contours and Surfaces for Biomedical Image Analysis," *Proceedings of the Workshop on New Trends in Subdivision and Related Applications (NTSRA'12)*, Milano, Italy, September 4-7, 2012, pp. 22.

R. Delgado-Gonzalo, N. Chenouard, M. Unser, "Parametric Snakes in Microscopy," *Euro-BioImaging First Workshop on Bioimage Analysis Software: Is There a Future Beyond ImageJ? (WBAS'12)*, Barcelona, Spain, April 30-May 1, 2012.

R. Delgado-Gonzalo, P.D. Tafti, C. Rabut, M. Unser, "Fractional Laplacians, Splines, Wavelets, and Fractal Processes," *Research Workshop on Subdivision and Refinability 2009 (SR'09)*, Pontignano, Italy, October 12-15, 2009.

Other Activities

- Sports Outdoors running (team leader of the Biomedical Imaging Group running team), group fitness classes, and golf.
- Other IEEE member, robotics, role-playing and real-time strategy games.

