Landmark Active Contours for Bioimage Analysis: A Tale of Points and Curves

Virginie Uhlmann

Thèse N° 7951 (décembre 2017)

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*
Prof. Dimitri Van De Ville, *rapporteur*
Prof. Carolina Wählby, *rapporteur*
Prof. Fred Hamprecht, *rapporteur*

École polytechnique fédérale de Lausanne—2017

Cover design by Annette Unser
Printing and binding by Repro-EPFL
Typeset with LaTeX

Available at **http://bigwww.epfl.ch/**

Abstract

The problem of identifying the outline of objects in images can be approached from two starting points, either by considering localized features (landmarks, keypoints or regions), or by searching for global contours. Features are regions or points of interest and usually include a description of the local properties of the image around them. The definition of a feature is flexible. Most often, it consists of a list of desirable properties inspired by the application at hand. Algorithms are then designed to robustly detect occurrences of the feature in the image under the effect of various geometrical transformations. Contours, on the other hand, are (portions of) curves that can be delineated using deformable models, for instance relying on spline curves. Splines are in particular at the core of a large family of such models called spline-based active contours, or designer snakes. These methods can be customized and adapted to outline a large variety of objects in many types of images.

In this thesis, we aim at unifying these two strategies by bridging automated feature detection and spline-based active contour segmentation for bioimage analysis. Our work proceeds in three steps. First, we introduce and characterize the Hermite spline interpolation framework, a model that allows incorporating local information at each node in the spline curve. Then, we study the design of custom feature detectors based on the steerable filters formalism. With these two ingredients, we propose a semiautomated segmentation algorithm called the *landmark snake*, which is defined relying on Hermite interpolation and evolves a curve in the image to outline objects of interest using information provided by steerable features detectors. The Hermite spline formalism allows for a direct correspondence between control points on the spline curve and landmarks, simplifying the algorithm design and allowing for user-friendly interactions. The approach is generic enough to be used in a wide variety of data, as will be illustrated through real bioimage analysis problems in the context of collaborative work with external laboratories.

Keywords: Bioimage analysis, segmentation, contour detection, outlining, feature detectors, keypoint detectors, active contours, parametric models, Hermite interpolation, spline-based methods, steerable filters.

Résumé

Lorsqu'il s'agit de délimiter le pourtour des objets dans des images, deux points de départs peuvent être identifiés. Le premier consiste à considérer des zones d'intérêt locales (région, point clé ou repère), le second à chercher des contours généraux. Les zones d'intérêt sont des régions ou des points et sont habituellement dotées d'information fournissant une description des spécificités de l'image dans leur voisinage proche. Leur définition est flexible. Le plus souvent, elle consiste en une liste de propriétés dépendant de l'application pratique considérée. Des algorithmes sont ensuite conçus pour identifier où de telles zones d'intérêt, possiblement sujettes à diverses transformations géométriques, se retrouvent dans l'image. Les contours, d'autre part, sont des (portions de) courbes qui peuvent être identifiés en utilisant des modèles déformables, par exemple construits à partir de courbes splines. Les splines sont notamment au cœur d'une grande famille de tels modèles, que l'on appelle parfois *snakes* (*i.e.*,"serpents"). Ces méthodes sont très modulaires et peuvent être facilement adaptées pour segmenter une grande variété d'objets dans de nombreux types de données d'imagerie.

Dans cette thèse, nous aspirons à unifier ces deux stratégies en rapprochant la détection automatique de zones d'intérêt et la segmentation à base de modèles déformables utilisant des splines, le tout dans un contexte d'analyse d'images biologiques. Notre travail se déploie en trois étapes. Dans un premier temps, nous posons le cadre théorique de l'interpolation hermitienne, un modèle permettant de construire des courbes splines avec des contraintes locales à chaque nœuds. Ensuite, nous étudions la conception de détecteurs personnalisables de zones d'intérêts en se basant sur le formalisme des filtres dits orientables. Avec ces deux ingrédients en main, nous proposons un algorithme semi-automatique pour la segmentation d'images. Il s'agit d'un modèle déformable construit par interpolation hermitienne qui évolue dans l'image pour trouver le contour des objets en utilisant de l'information provenant de détecteurs orientables de zones d'intérêt. L'interpolation hermitienne offre une correspondance directe entre les points de contrôle sur la courbe spline et les zones d'intérêt sur l'image, simplifiant la structure de l'algorithme et offrant à l'utilisateur la possibilité d'interagir avec la courbe de façon intuitive. Notre approche est suffisamment générale pour être utilisée avec de nombreux types de données. Nous l'illustrerons au travers de plusieurs problèmes d'analyse d'images biologiques émanant de travaux collaboratifs avec d'autres laboratoires de recherche.

Mots clefs : Analyse d'images biologiques, segmentation, détection de contours, délimitation, détecteurs de zones d'intérêt, détecteurs de points clés, contours actifs, modèles paramétriques, interpolation hermitienne, méthodes à base de splines, filtres orientables.

Ce sont des chercheurs,
alors ils cherchent
et parfois ils trouvent.

Y. Pradeau, *Algèbre*

To the one we trust without data.

Acknowledgement

A thesis is not only the fruit of some years of research in a lab, five in my case, but also the consequence of a longer life path. Lots of people crossed my road, inspired or encouraged me, provided me with love, friendship, discussions, attentive listening, or smiles. All of these contributions, regardless of their duration, were precious gems distributed along my way. It is unfortunately not possible to name and thank all these beautiful human souls individually in this acknowledgement section, although I would very much like to be able to. To those who won't find a mention of their name hereafter, be reassured, I did not forget you. There is fortunately much more space in my heart than on these few pages of paper.

I would first like to thank my mentors in science: Dr. Anne Carpenter, Prof. John McKinney and Prof. Michael Unser. They appeared very early in my academic life, and gave the young student I was a hint of what doing research and being a scientist means. They offered me time, opportunities and guidance, and trusted my abilities as I was not yet able to do so. John was probably the best teacher I ever had in a biology-related class and passed on to me his everlasting enthusiasm and curiosity for knowledge beyond the limits of his own scientific field. I am thankful for all our exchanges, from discussions over a coffee to e-mail conversations about philosophy books. Anne is and will remain my role model for her unbelievable abilities to simultaneously do amazing science, have a flawlessly upright scientific ethics, be an incredibly considerate manager to her team, and be a fantastic human in general. Our world is fortunate to host people like her, and I am lucky to have met her.All the aspects for which I am grateful to Michael won't fit here. He had already been a precious adviser before I started my PhD, and became a true scientific guide over the past years. His pure enthusiasm for research, his never-ending quest for mathematical beauty, his dedication to teaching and his infallible availability are some of the many qualities I admire in him. His unfailing support and encouragements allowed me to achieve far more than I thought I could. When I started my thesis, Michael told me to enjoy this time because, he said, PhD years are amazing. I can now say not only that he was right, but also that he is largely responsible for it. These three people have been and still are sources of inspiration. They made me grew up into a scientist and helped me develop ambitions, critical thinking and strong ethical principles. Most importantly, they gave me the support I needed to build self-confidence. I owe them lasting gratitude.

I am also grateful to my thesis jury, Prof. Pierre Vandergheynst, Prof. Dimitri Van De Ville, Prof. Carolina Wählby and Prof. Fred Hamprecht. Pierre was teaching signal processing to my 3rd-year-EPFL-Bachelor-student-self seven years earlier. We met again as I started the thesis and became teaching assistant for that same class. He has been an important character of my EPFL adventure ever since, and I am incredibly glad that he managed to find time to chair my thesis jury in spite of his crazily busy agenda. Dimitri also appeared early on in my EPFL life, as I was a Master student. At several occasion, in these early times and later on during my PhD, he dedicated time to share precious pieces of knowledge. I am grateful for his kindness and availability. Carolina and I shared an office at the Broad Institute in Cambridge, MA, during my Master's thesis. From there, our paths crossed again personally at various conferences, and scientifically as I started working on *C. elegans* segmentation. I admire both her scientific and human skills. Finally, Fred enabled one of my favorite collaborative projects to flourish into a beautiful piece of science. I learned a lot from this collaboration, and I am grateful for the continuous support, critical input and new ideas he kept on bringing into the project. It was a great honor to have these people, who are each important to me for a different reason, giving a critical look at my work. I am thankful to them for their time, comments and presence.

Beyond research, I woke up every morning of these five years looking forward to go to the lab for a good reason: the Biomedical Imaging Group (BIG) is a place full of amazing people. I want to thank warmly all members of the BIG, past and present, for making the north end of the corridor of the BM, 4th floor, a joyful place to be. It was a chance to be surrounded by such brilliant and friendly people, whether it involved working together, advising each other, recycling the meeting room to watch movies, teaming up to beat the rest of the world at Superbuzzer, sharing inappropriate comments in the well-named *BIG slackers* WhatsApp group, exploring local bars before and after (also sometimes during) conferences, running the Tour du Pays de Vaud, eating raclette or simply being there for each other when it was needed. I liked these folks so much that I stayed quite long around and got to know several generations of BIGgers, namely: the timeless wise men (Daniel and Philippe); the fairy godmothers, a.k.a secretaries (Manuelle, Nadia, Monika and Claudia); the pre-PhD era people (Ayush, Nicolas, Matthieu, Hagai, Stamatis, Jean-Charles, Pouya, Cédric); the primitive PhD era people (Emrah, Ricard, Ulugbek, Masih, Pedram, Zsuzsanna, Daniel, Martin, John Paul, Julien, Arash); and the late PhD era people (Emmanuel, Denis, Mike, Toulouse, Ferréol, Anaïs, Laurène, Harshit, Thanh-An, Shayan). A special thanks goes to the dream team of the BM4.141 office: Ricard, Adrien and Mike. The current president of the United States claims he has the best words, I can claim I have the best office mates.

I thank as well my numerous collaborators, who contributed to making my research meaningful and exciting. First, the colleagues from the BIG, Zsuzsana, Ricard, the two Daniels, Pedram, Adrien and Julien. Our respective projects and knowledge met at some point, and we walked for some miles together down the research road. I also had the chance to interact with the excellent image analysis team from the BIOP, Olivier and Romain. Their smart ideas, abundant collection of beautiful data and keen sense of humor made it always delightful to work with them. Then, researchers at the LMIC, including Neeraj, Olivia and Gaëlle, among others. They were a long-lasting source of challenging scientific questions and knowledge. The mycobacteria they image all day long are still bugging me. The worms dream-team Matteo (LMIS2, EPFL) and Laurent (LISP, EPFL) provided me with a gazillion of data that are as good-looking as the associated image analysis task is challenging. I am grateful to them for including me in their exciting project, for their patience and for their enthusiasm. A special thanks goes to Pavan (EPFL), with who I started my first serious collaboration on the fly tracking project back in 2012. Pavan has been the ideal collaborator: extremely motivated, eager to have the project move forward, and mindful of good communication. Now that the years told me how to drive collaborations to be successful, I realize how much I am indebted to him for keeping this project alive and believing in it. I learned a lot from interacting with him. Finally, Carsten (University of Heidelberg) brought his graphical models magic into the mycobacteria tracking project in 2015 and has been a infallible supply of great ideas and outstanding pieces of code ever since. Working with him has been an enlightening experience, yielding very neat research but also allowing me to discover new scientific tools, improve coding skills and enhance communication. I had the chance to visit Heidelberg at various occasions, upgrading our daily Skype meetings for real ones and having superproductive brainstorming sessions. In Heidelberg, with the help of Steffen and Philip, we extended our collaborative efforts to the subtle art of computer Jenga and the perilous exploration of bars. My gratitude goes both to the great collaborator and friend: I am thankful for all the fun we had, from cutting-edge science to epic kickertable games.

From 2012 to 2016, I got involved quite seriously into the teaching organization for the Signaux & Systèmes class given to the Life Sciences and Microengineering 3rd year Bachelor students. The team of teaching assistant featured seriously awesome elements who leave me fond memories of these times. In addition to my beloved BIG colleagues (Julien, Emrah, Ulugbek, Laurène, Anaïs, Emmanuel and Toulouse), I want to thank in particular the LTS2 members Johan, Yann, Lionel and Michaël. I am also grateful to the generations of EPFL students who made sure we did not get bored during exercise sessions and sometimes rewarded us with very creative reinterpretations of the fundamental aspects of signal processing.

I am very fortunate to be surrounded by beautiful souls in my private life. Some crossed my path during my student's years, like Fabian and Laure, Charlotte, Jalil and Nicolas, and some later, like Clément and Sara. There is as well some that form some kind of extended family: Zsuzsanna, Marie, Cynthia, Denis, Sylvain and Marie, and Gwen. These people have always been there in my life, or it feels exactly as if they did. No matter how far away our paths diverge, our common stories simply resume and move forward whenever life gives us chances to meet. To all of them, I am profoundly grateful for their friendship.

I want to address a particular thanks to my brothers and sisters from the Valdésia Lausannensis as well as from the Falkensteinerbund and Wingolfsbund, especially Teλλo, Paske//e, CheFa#, Pocalypse, Lupine, Alexander, Enade, Tallator, Lonrifl and Couetch. I joined Valdésia as a first year bachelor student in 2007. During the ten years that followed, we built together unforgettable memories in Lausanne, Bern, Basel, Zürich and in Germany. They helped me forge my critical thinking and taught me one of the most useful principle in life, namely that one should never ever take oneself too seriously. The precious collection of souvenirs I have with these people is a cry of protest against the common belief that student fraternities are obsolete.

I am infinitely grateful to my family: my father Dominique in Boston; my grand parents Gilbert and Josiane in Nyon; my brother Yvan, who basically is my second half, my stepfather Andr and my mother Sophie, who I love to the moon and back. I have a thought for my grand mother Elizabeth, who left this world too early to gave me the chance to explain her my thesis, but who would have surely been an attentive interlocutor. These people surround me with love and care, some of them since 28 years, encourage me in every adventure I undertake, and make it possible for me to follow a path that resembles me. Their endless love and unconditional support are the sources of my everyday enthusiasm and energy. A warm thanks also goes to my partner, Alexander, for being my peace haven regardless of the many kilometers standing between us.

Last but not least, a very special thanks goes to Julien. He started his thesis in the BIG at the same time as me, and we walked together on all possible paths where life took us during our respective PhDs. We passed on to each other millions of pages of research papers, essays, and french novels to read, listened to hours of music on repeat, ran over 100km together, scribbled on kilometers squared of notebook pages, spent an infinitely long time talking, exchanging ideas, fiercely debating, trying to understand science, the world, our loved ones and ourselves, and strived to become better humans, tiny steps by tiny steps. We both lifted each other up to reach beyond our own walls, regardless on how high and steep they appeared to be. Our scientific and personal interactions shaped the researcher and grown-up human I am today. I am thankful from the bottom of my heart to life for having brought our paths to cross. Our exchanges gave both of us the immensely rich chance to live two lives during five years, and I know many more are still to come. The whole story of this thesis cannot be conceived without him, it is also his in many ways.

Finally, I'd like to thank you, reader, who takes some of your precious time to go through my work. It is the outcome of a long initiatory journey, both scientific and personal. I hope it gives a taste of the joy of the research process and, maybe, inspires others to experience it themselves. Happy reading.

Contents

Chapter 1

Introductory Notes

1.1 The Bioimage Analysis Era

Observing, understanding and measuring biological phenomenons is at the heart of biological and medical research. Since a few decades, the constant advances achieved in the development of imaging methods keep pushing forward our abilities to obtain quantitative microscropy data of cellular and molecular processes [1]. It is nowadays possible to analyze both the spatial organization of complex systems as well as their spatiotemporal behavior [2, 3]. Modern techniques in microscopy go as far as enabling the time-lapse imaging of *in vivo* cells and of single molecules in a very precise manner [4]. As a consequence, measurements outputted from biological experiments take the form of images and videos in most of the cases.

In order to make statistically relevant observations, large amounts of quantitative measurements are required. This led to technological achievements such as high-throughput microscopy [5], which automatize the acquisition of gigantic image sets corresponding to various experimental conditions and duplicates thereof. As a direct consequence, the size and complexity of image data has kept on increasing. With nowadays digital storing capabilities and acquisition speed, terabytes of images can easily be obtained in a time frame of days. Experiments involving time-lapse data commonly generate hundreds to thousands of images, each possibly containing multiple objects to be analyzed. In this ocean of data, the analysis step itself stands out as a strict bottleneck. It is an unavoidable step to draw any conclusion from the data, but cannot, in the vast majority of cases, be performed by the experimenter because of time and resources constraints. This situation motivates the development of automated solutions for image analysis, which articulates around three different needs:

1. gaining time or, more modestly, making the analysis feasible by bringing down the time it requires to be performed to human time scales;
2. ensuring reproducibility by obviating unpredictable and inconsistent human mistakes;
3. enhancing objectivity by relying on precise and deterministic computer-based criteria and measurements, thus avoiding experimenter's bias.

These requirements have been driving the development of image analysis softwares and tools for many years. Unfortunately, coming up with computer-based solutions to extract quantitative information out of the rapidly expanding amount of acquired data remains a major challenge. The growing community of researchers working in this field made it evolve into a new area or research, referred to as *bioimage informatics* [6, 7, 8]. Research efforts have there led to tools that are now widespread among the neighboring community of biologist users, such as ImageJ [9], Fiji [10], ICY [11], CellProfiler [12], and Ilastik [13].

The approach followed in bioimage analysis could be summarized as thinking about what is happening when humans see. Although performed by all of us continuously through the day and without efforts, visual perception has kept neuroscientists, psychologists, image processing researchers and computer scientists busy for a long time [14]. Understanding and reproducing human performances is highly non-trivial and leads to many scientific and philosophic interrogations. The whole challenge of image analysis is to get enough comprehension of the process leading from simple visual inputs to complex scene understanding in order to subsequently reproduce it in a machine.

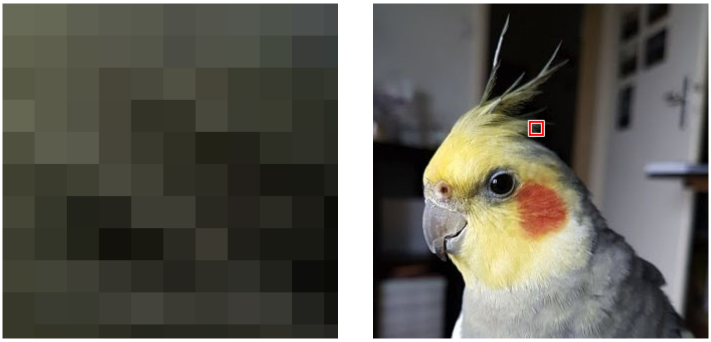
1.2 The Segmentation Problem

Segmentation is one of the standard problems arising when trying to design algorithms that reproduce visual perception [15]. It consists in identifying elements in a scene (*e.g.*, video, picture) or, more precisely, in managing to identify which parts of the scene belong to either the elements or the background. This usually boils down to finding boundaries of objects, or transition zones.

The main aspect making segmentation a challenge is the discrete, pixelized nature of images. Image data are "seen" by computers as arrays of values, without any global view. While humans rely on the "big picture" to understand images, computers only have access to local pixel information, as illustrated in Figure 1.1. It makes everything more difficult: since global context is lacking, the sensitivity to noise, or small perturbations in pixel values, is enhanced. Segmentation algorithms try to introduce or mimic some amount of global perspective on the scene being analyzed. To do so, they rely on a mixture of image-based and prior information. Understanding this problem gives all the elements required to explain why segmentation remains an area of open research: since local pixel information is generally not sufficient to understand the image, prior information on the nature of the elements being searched is incorporated in algorithms. This makes it very challenging to design all-purpose approaches that would remain flexible enough to perform segmentation of every possible object in every possible image. In fact, what allows our visual system to solve the segmentation problem so efficiently and with such amount of generality is more our processing capabilities (*i.e.*, the visual cortex in the brain) than the visual inputs we get. Our experience from the physics of the real world and the many examples of scenes we encountered during our lives allow us to swiftly perform segmentation in virtually any image. Keeping this in mind, it comes at no surprise that learning-based approaches for segmentation foster a lot of attention and start to produce impressive results that are slowly approaching human performances [16].

An ambivalent aspect of the segmentation problem must however be kept in mind. What is considered as a good segmentation result is often dependent on the interests of the observer. Given a scene, several segmentations might be considered as correct depending on what the observer is interested in. This further wards off the possibility for a unique, global segmentation approach as it is neither easy to design nor necessarily desirable in practice.

To conclude, it is worth mentioning that segmentation is not a research area that is specific to bioimage informatics. This problem is common to different research domains, including computer vision. Although the initial task is similar, the resulting challenges in these two fields differ significantly. In the standard computer vision setting, the elements to be segmented are often very specific (*e.g.*, cars,



(a) (b)

Figure 1.1: Comparison of computer- and human-based views of images. (a) The computer view, which consists in a collection of local pixel values (*i.e.*, color or grey level) without context. (b) The human view, which consists in a global consideration of the scene, complemented by additional prior experience of the real world. The area shown in (a) is indicated with a red square.

pedestrian, roads, houses). Lots of prior information can thus be incorporated in order to discriminate objects against background elements, and physical constraints can provide further help. As an example, in the same scene of road traffic, it is unlikely to have cars with wheels facing the road and wheels facing the sky. Once objects have been identified, individual characteristics can be extracted to discriminate them from each other (*e.g.*, car paint color, pedestrian clothes). These specific features provide important clues to solve cases of temporary occlusion for instance. In bioimage informatics, the objects of interest are most often all very similar (*e.g.*, cells, bacteria, white lab mice) up to some small biological diversity. Scenes are most commonly observed from above, reducing the possibility to rely on physics of the 3D world. Most importantly, the research problem often aims at identifying small deviations from the natural biological diversity, further narrowing down the amount of constraints that can be incorporated into the algorithms (*e.g.*, maximal and minimal sizes or average shape). Additional challenges appear from the microscopic nature of the objects and the physics of light in the optical settings being used. The general formulation of the segmentation problem is therefore at the same time shared among different fields, and very specific to the field being considered in the particular challenges it poses.

1.3 Roadmap of the Thesis

This thesis tells the story of the construction of a bioimage analysis method bringing together curve-based (global) and point-based (local) approaches. As curve-based approach, we consider continuous curve interpolation and, in particular, a type of spline construction called Hermite splines. By point-based approaches, we refer to keypoints or feature detectors. In this work, we focus on feature detectors designed from (possibly multiscale) steerable filters. From these two ingredients, we construct the *landmark active contour*. It consists in a curve modeled using Hermite splines, which evolves from an initial position in the image to fit the boundaries of objects of interest by minimizing a cost function, or energy, built from the output of feature detectors. This model is then used to segment bioimages. Every part of the thesis is an element of this global story. As a consequence, the content navigates through multiple research domains: approximation and spline theory, steerable filters, wavelets, active contours, and bioimage analysis. A global view of the different "modules" composing this work and of their interconnections is illustrated in Figure 1.2.

Each module is the subject of a dedicated chapter, in which we explore its topic in more details. In Chapter 2, we introduce the points-and-curves paradigm that inspires the development of the landmark snake. We also review state-of-the-art approaches that are relevant to it. Then, we first focus on curves and present our work in spline and approximation theory related to the Hermite spline construction (Chapter 3). We continue by discussing points, or feature, detection and study the design of multiscale steerable filters to achieve this task (Chapter 4). Our investigations in the two separate worlds of points and curves merge in Chapter 5, where we introduce the construction of the landmark snake. In Chapter 6, we present several bioimage analysis problems which benefited from the tools we developed. Finally, concluding remarks are provided in Chapter 7.

Although the content of all chapters are interconnected in the global story, many aspects in each of them can be understood individually. Our work on splines and on steerable filters can be appreciated independently from each other. The construction of our active contour model then uses these spline model and steerable

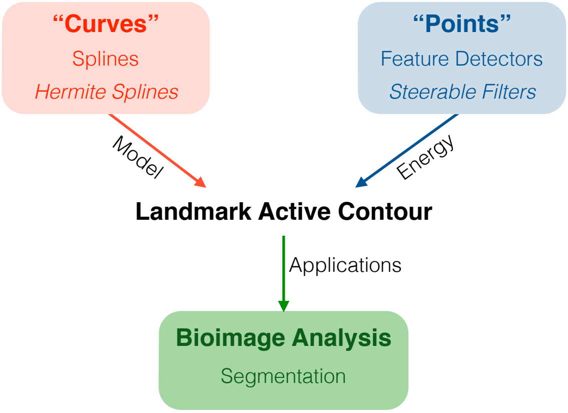


Figure 1.2: The content of the thesis in a nutshell. We propose the landmark active contour, a method that aims at bringing together curve-based (*e.g.*, splines) and point-based (*e.g.*, feature detectors) methods for segmenting objects in bioimages.

filters construction to bring together point- and curve-based approaches for object identification and segmentation. The bioimage analysis results are finally obtained from either one or several elements among the spline model, steerable filters, and active contour we designed. A possible walk through the thesis depending on the reader's interest could be as follows.

Theory-oriented tour focusing on the details of the more "formal" contributions. Start first by a walk through Chapter 3 to review the basics of Hermite spline interpolation (3.1) before moving on to novel results (3.2, 3.3 and 3.4). Then, head to Chapter 4 and (re)discover the theory of steerable filters (4.1). Continue to the new contributions in the design of feature detectors using steerable filters (4.2 and 4.3) and get a taste of future research directions (4.4). Finish with a visit to Chapter 5 (5.1 and 5.2) to see the landmark active contour construction resulting from the earlier theoretical wanderings.

Application-oriented tour focusing on the details of contributions targeted towards particular applications. Move directly to Chapter 5 and quickly throw a glance at the construction of the landmark active contour (5.1 and 5.2). Walk then through the practical aspects of the approach (5.3 and 5.4) and see illustrative examples of practical use. If curiosity calls, a small hike around the theoretical hills of Hermite splines (Chapter 3, essentially 3.1) and steerable filters (Chapter 4, essentially 4.1) can be of interest. Move then to Chapter 6 to get detailed examples of practical bioimage analysis problems solved with the proposed method, involving the analysis of fly motion (6.1), the tracking of mycobacteria (6.2), and the study of *C. elegans* nematodes dynamics (6.3).

Story-oriented tour focusing on the overall "points and curves" story. Begin the tour with Chapter 2 to get a sense of the motivation of the thesis. Pass by Chapters 3 (3.1) and Chapters 4 (4.1) to get a quick view of the two main theoretical ingredients on which our main contribution relies. Wander through Chapter 5 to discover the theoretical (5.1 and 5.2) and practical (5.3 and 5.4) aspects of landmark active contours based on Hermite interpolation. Finally, stroll around the introductory and results sections of Chapter 6 (6.1.2, 6.2.2 and 6.3.2) to get examples of practical use of the method through several successful collaborative projects in bioimage analysis.

All tours can be completed by the concluding remarks in Chapter 7, which summarize the work and discuss possible directions for further investigation.

1.4 Main Contributions in a Nutshell

Formally, the main goal of this work is to revisit the standard formulation of spline-based active contours using Hermite interpolation and to exploit the possibilities it opens to connect spline-based algorithms with feature detectors for bioimage segmentation. To motivate this objective, get a proper understanding of the tools we use, and formally describe the solution we propose, we provide the results of personal reflections on the related research topics, interpretations of existing results, outputs from collaborative efforts with other researchers, and individual work. In all chapters, we explicitly mention the name of collaborators in footnotes or directly in the text. In order to help the reader untangle the novelties from reinterpretations of existing material, we hereafter provide a short summary of our main scientific contributions.

Hermite splines: The formulation of the Hermite interpolation problem and of the corresponding basis functions in all generality is the work of I. J. Schoenberg in the 70's. However, the explicit connection of first order Hermite splines with classical B-spline models and Bézier representations had not yet been explicitly formulated. Similarly, the study of approximation error and optimality properties, as well as the derivation of exponential Hermite splines, are novel contributions.

Feature detectors: The multiscale steerable filters formalism was previously introduced by N. Chenouard and M. Unser. The design of steerable filters by either tuning the angular profile or by searching for optimally localized radial profiles are however both novel studies that have been performed in collaboration with other members of the Biomedical Imaging Group.

Landmark snake: The entire landmark snake algorithm is our most significant contribution, as both the formulation of the snake model and the presented energy terms have been developed in the context of this thesis.

Practical bioimage analysis problems: All considered practical problems are the fruits of collaborative efforts with biologists and computer science researchers from EPFL and elsewhere, and are thus original results.

Through our work, we commit ourselves to releasing working versions of the algorithms we propose as plugins for free, open-sources softwares widely used by the biology community such as ImageJ or Icy. As our various contributions get introduced, we indicate in footnotes the websites where the companion source code or compiled binaries can be found. These computational tools are also contributions on their own.

Chapter 2

Rambling Through Points and Curves

In this chapter, we introduce the main motivation for this thesis. We first formulate the problem we want to address, then review state-of-the-art approaches that are relevant to it, and finally enunciate our research plan to address it.

2.1 Synopsis

When segmenting images (*i.e.*, separating background from foreground, see Chapter 1), an important step is outlining the objects. Another essential aspect is the identification of features that characterize the objects of interest. Identifying features is required to distinguish actual objects from artifacts, to discern different instances of objects of interest, or to extract quantitative measurements from the object (*e.g.*, local orientation, head and tail location), among others. In most cases, neither the outline nor the location of feature elements are the final output of the analysis process. They rather are pieces of information that further get combined to extract other metrics for quantification purpose (*e.g.*, objects size, orientation, shape). Therefore, in order to characterize objects of interest from image-based information, both contours and features usually matter as they bring knowledge of different kind. One can formulate two different paradigms, which then shape the development of image analysis pipelines.

1. **Identification of features:** we identify localized salient points or regions in the image. From this, we then search for the associated objects outlines, *e.g.* by finding portions of curves linking the points. The initial step in this setting is the detection of a collection of specific interest points, landmarks or regions in the image, which is carried out by feature detectors.

2. **Tracing of contours:** we generally search for all contours (*i.e.*, portions of lines or curves) in the image. To distinguish which of them are relevant for the objects of interest, we then walk along them or investigate the portion of the image they enclose and look for features. It here all starts with the identification of meaningful outlines, which can efficiently be achieved for instance by methods based on deformable models.

In the following, we describe more precisely existing methods that can be used as starting point of each of these two paradigms and that pave the road to the research carried out in this thesis. We then formulate a unification proposal, which motivates and orients the research carried out in the rest of our work.

2.2 First Paradigm: Identifying Salient Points

Methods for identifying salient points in an image first have to deal with defining what a salient point is. Often referred to as landmarks, keypoints, or local features, they are distinctive location in the image, loosely defined as image patterns which differs from their immediate neighborhood [17]. This distinctiveness can be expressed in terms of intensity, color[1] or texture. The nature of the landmark itself is not so precisely defined. They can be points, segments, superpixels, small patches or regions of the image, representing blobs, portion of edges or corners, among others. The term “point” is therefore to be understood in a wide sense and does not necessarily mean “pixel”. Features can thus range from very specific patterns or objects to very general image content such as edges or ridges. A complete salient point identification algorithm usually return the location as well as a description of each detected feature, indicating for instance how strongly the portion of the image where the feature is located resembles it, how the feature is oriented, or how its

[1] Although, in this work, we will focus on grayscale images.

close neighborhood looks like. In addition, it is desirable for landmark identification algorithms to be robust under some set of transformations. The most common ones are translations, rotations and scaling. Detectors are usually tailored to match the properties of the application at hand.

Keypoints are very widely used in computer vision as they provide a small set of well-localized anchors that can then be used for matching points between different images. Typical applications are stereo matching, camera calibration or 3D reconstruction, where similar scenes are observed under different point of views. In such cases, the most important property that keypoints must satisfy is distinctiveness. This means that a particular keypoint should always be detected and properly identified (if possible uniquely, in the sense that each particular keypoint should be identified unambiguously) across images, even if the scene appears as distorted under the effect of perspective, rotation, scaling, or more complex geometrical transformations. Keypoints are usually not directly employed for segmentation, but as additional help. They provide discriminative clues so as to whether identified components correspond to objects of interest in the image or not, and provide seeds to initiate and guide the segmentation process. In such cases, what matters is the specificity of the keypoint, usually rather referred to as a feature. Few points should be detected in “uninteresting” parts of the image, while no “interesting” parts should be missed. The detection process should also be robust to slight variations, rotation and scaling of the features. We propose to group existing methods in two categories depending on whether they produce a set of unspecific but very distinctive points, or they specifically detect points falling into a particular category of similar features. Although the boundary between these two categories is loose, it allows capturing important differences and similarities between approaches. We give an illustrative example of each category in Figure 2.1.

2.2.1 Distinctive Points

Extensive research has been carried out on keypoint detectors in the field of computer vision [17]. Here, we provide a brief overview of the main trends by giving a few particular examples. In general, keypoint detection algorithms are built in a two-step fashion. First, the identification of salient points is carried out by a so-called *detector* which, as its name implies, indicates possible keypoint locations in the image. This detection step has to be robust under some set of transformations, such as scaling and rotation. It should also be robust against noise. Once a

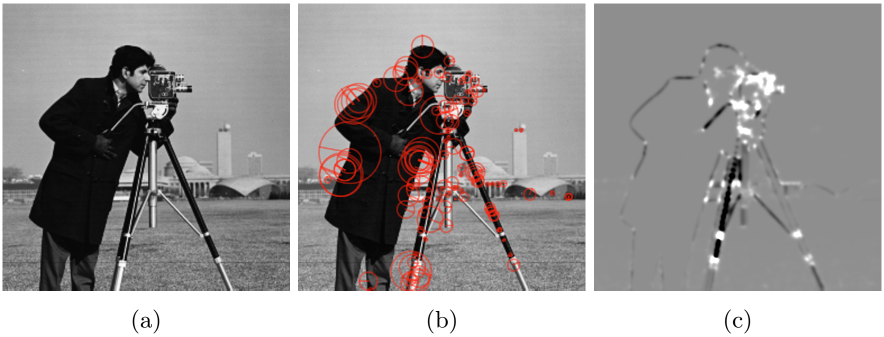


Figure 2.1: Illustration of different approaches for identifying salient points. (a) Input image. (b) Output of a detector providing a set of distinctive points (SIFT). It consists in a discrete list of points locations, scale and main orientation. Here, they are overlaid as circles on the input image, with the circle size depicting the scale and a radius inside it depicting the main orientation. One notices that keypoints are unspecific, *e.g.*, they are found both on corners and edges. (c) Output of a detector providing a set of points with common properties (Harris corner detector). It yields a magnitude map, which contains at each location on the image an measure of how much the desired feature is present. Here, the brighter an area, the more "corner-like" it is.

keypoint is identified, a *descriptor* is constructed to characterize the point and its immediate neighborhood. The descriptor can be composed of information on, *e.g.*, the distribution of pixel intensities, the local orientation around the point, or both. Considering that applications involving matching are targeted, the descriptors are often quite complex in order to decrease risks of ambiguity. In general, the overall performance depends on the combination of detector and descriptor . Keypoint algorithms can be seen as “super filtering” algorithms in the sense that they highlight some features and provide descriptive information about them. However, compared to filtering which usually highlights the presence of a particular feature in the image, these methods return a list of possibly very differently-looking keypoints in the form of a list of coordinates and descriptors.

The most famous keypoint detector method is called SIFT [18, 19] for Scale Invariant Feature Transform, and was found to be very efficient for many applications. In a nutshell, the algorithm steps are as follows.

Detector: First, intensity extrema are detected in a scale-space representation of the image built using an approximation of a Laplacian of Gaussian pyramid (difference of Gaussian). Extrema detected across several scales are considered as keypoint candidates. Keypoint location, scale and ratio of principal curvatures (computed based on the eigenvalues of the Hessian matrix) are extracted in order to reject poorly localized extrema and points with too low contrast. Finally, a local orientation value is assigned to each keypoint.

Descriptor: Gradient magnitude and orientation are computed at the scale of the keypoint based on finite-differences between pixels. An histogram is then made from the gradient orientations in a region around the keypoint and peaks are identified, corresponding to dominant orientations. The descriptor itself is obtained as follows. The local gradient vectors of pixels in a neighborhood around the keypoint is computed. To do so, pixel values are weighted by a Gaussian window and aggregated into four orientation histograms corresponding to each region of the neighborhood (top left, top right, bottom left, bottom right).

SIFT inspired the design of many variants aiming at improving the speed of the algorithm (SURF [20]) or its distinctiveness (GLOH [21]).

The development of most recent methods was mostly influenced by the need for fast computations. To achieve this goal, detectors such as FAST [22] have been proposed. There, keypoints are simply identified by evaluating and thresholding image intensity values, possibly relying on classifiers to recognize keypoint-like intensity patterns. A lot of effort has been put in the construction of binary descriptors, which further reduce the computational load and open the possibility for real-time matching. Such keypoint descriptors are built as binary strings formed by comparing image intensities between pair of pixels in a neighborhood around the keypoint. The question is then to establish how one selects the collection of pixel pairs. In BRIEF [23] and ORB [24], locations of the elements composing each pair are selected by randomly sampling a Gaussian around the keypoint. In BRISK [25], a circular pattern composed of concentric circles around the keypoint is used. Further efforts in this direction have been explored in FREAK [26], where circular patterns are designed in inspiration from the animal retinal system, in a way that the density of comparison points decreases exponentially as one gets away from the keypoint.

2.2.2 Points with Common Properties

We categorize now methods that allow identifying locations of the image where a specific common element is present. This is the core aspect that distinguishes them from methods of the first category. In keypoint algorithms for matching applications, the emphasis is put on invariance between sets of detected keypoints. It implies that, within the set of keypoints, elements can correspond to portions of the image of different nature. However, this set must be robust against deformations. In the present category, the emphasis is put on invariance within the set of keypoints in the sense that all detected points composing the set correspond to locations where the same specific image feature is found. Good feature detectors are usually filters designed or tuned to respond well to a particular pattern. The output of the detector can be a map of the same dimensions as the input image, since every point in the image generates a filter response. Compared to keypoint detectors, low-dimensional descriptors are often sufficient since the identification of landmark locations is generally the final goal. The descriptor part then takes the form of a measure indicating how much the specific feature is present, possibly containing some scale and orientation information in addition.

The Harris corner detector [27] is probably the most famous feature detector. It identifies corner points in the image relying on the eigenvalues of the structure tensor, or second moment matrix, which is derived from the gradient of the image. This matrix describes the distribution of the gradient in a local neighborhood of a point.

A measure of "cornerness" is assigned to each location in the image by estimating how the local gradient magnitude is split across its two othogonal components. The Harris corner detector is sometimes also considered as an example of naive keypoint detector, although it lacks a descriptor part. In fact, most early keypoint detector methods rely on the Harris corner detector and add some built-in descriptor on top of it [28, 29], further thinning the boundary between the two categories we consider. Targeting another type of features, the Laplacian of Gaussian [30] allows detecting blobs as extrema in the result of the convolution of the image with a mexican-hat-shaped filter. A computationally less intensive approximation of this method is used in the detection part of SIFT. Neither the Harris corner detector nor the Laplacian of Gaussian provide scale and orientation information per se. It is however possible to detect corners in a rotation-covariant way[2] for instance by running the Harris corner detector on a set of rotated version of the original image. Multiscale detection can be achieved relying on pyramids made of versions of the image that have been downsampled at different rate. Rotating or building the pyramid however introduces discretization artifacts yielding to detection errors, which call for the use of steerable filters to address the rotation-covariant part, and wavelets for the multiscale aspect. Steerable filters [31] are very appealing solutions for feature detection as they hold the potential for being tuned to detect various kind of features of interest with the same formalism. Although not multiscale in their usual formulation, they can be continuously rotated, avoiding most of the discretization artifacts, and provide smoothing by construction, making them more robust to noise. The local orientation estimation through the histogram binning step of SIFT can in fact be seen as an approximated version of steerable gradient filtering[3]. A formal introduction of the steerable filter framework is provided in the first part of Chapter 4.

[2] A rotation-covariant detector tuned to a specific feature has the following property. If the feature is subjected to a rotation by a given angle θ_0, detection can be performed using a version of the detector rotated by the same angle θ_0. The final detection result is then equivalent regardless of the value of θ_0.

[3] Binning the gradient response at different orientations into an histogram and searching for peaks indeed amounts to steering the gradient in a discrete fashion.

2.3 Second Paradigm: Capturing Contours

A family of algorithms called deformable models can provide a representation of meaningful contours in an image. It is the topic of our review paper [32]. Deformable models have gained popularity in segmentation applications since they provide an excellent trade-off between flexibility and efficiency [33]. Within this category, active contours (or *snakes*) [34, 32] are very popular methods. These algorithms consist in a curve evolving in the image from an initial position to fit the boundaries of the object of interest in such a way as to minimize an energy. Segmentation with snakes is therefore performed in two steps. An initial configuration of the contour is given, and the snake is optimized from this initial condition by minimizing a cost functional. Two aspects of the active contour thus play a significant role: the snake *model*, which describes the nature of the deformable curve; and the energy functional, referred to as snake *energy*, an appropriately defined cost function that dictates the evolution of the curve. The success of active contour in practice can be attributed to their combination of effectiveness and versatility. As we will see, these methods can easily be tuned to the particular characteristics of a given segmentation task. The choice and design of both the model and energy of the snake is thus highly dependent on the considered application and image modality.

Many different snake algorithms exist. They are usually grouped in three main categories, namely point-based, level sets and parametric snakes.

Point-based snakes [34] are historically the first active contours that have been proposed. They are constructed from the most elementary representation of discrete curves: an ordered collection of neighboring points within a grid [34]. We show in Figure 2.2a an example of 2D point-based snake overlaid on the grid associated to a discrete image model. The discrete curve, displayed as shaded pixels, here satisfies an 8-neighbor connectivity. The main drawback of this representation is that it relies on many parameters, even when encoding simple shapes. This impacts the robustness of the overall segmentation algorithm and results in a high computational complexity. Manual deformation of such curve is possible by simply updating the set of points composing it.

Level sets or geodesic snakes [35, 36] have been very popular in computer vision over the past decade [37, 38, 39, 40, 41, 42]. There, the curve is repre-

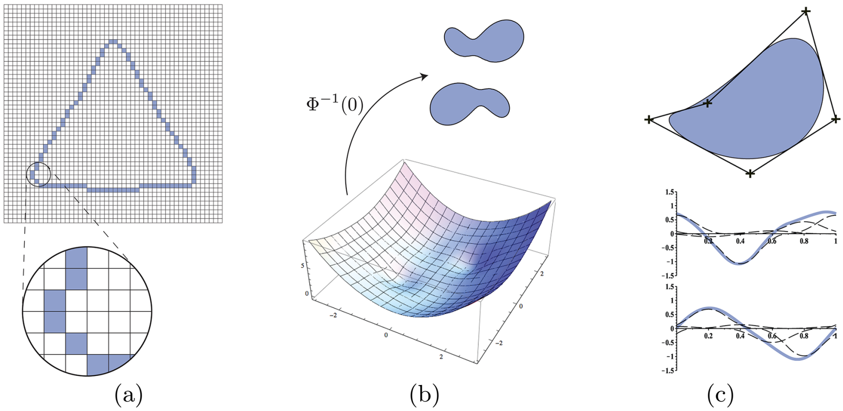


Figure 2.2: Depiction of the three main active-contour families. (a) Point-based snakes, defined as a discrete curve over the grid associated to a discrete image model. (b) Geodesic snakes, defined as a continuous curves corresponding to the zero level-set $\Phi^{-1}(0)$ of a scalar function Φ. (c) Parametric snakes, defined as continuous curve from a basis function, here a spline. The snake contour is shown as a solid line enclosing a shaded region, while the '+' elements are the control points. The corresponding parametric coordinate functions are displayed in solid lines, and the dashed lines indicate the weighted basis functions.

sented implicitly, described as the zero level-set of a higher-dimensional manifold. In the usual formalism, 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 over the image domain. A unique characteristic of geodesic methods is that they can be extended to any number of dimensions. In Figure 2.2b, we represent a set of curves generated as the result of computing $\Phi^{-1}(0)$. This approach originates from the work of Osher and Sethian, who aimed at modeling the propagation of solid-liquid interfaces with curvature-dependent speeds [43]. In this framework, the interface (or front) is represented as a closed, nonintersecting hypersurface flowing along its gradient field with either constant or curvature-dependent speed. Applying motion to it then amounts to solving a Hamilton-Jacobi-type equation written for a function in which the interface is a particular level-set. This type of active contour holds the interesting property of being particularly flexible in terms of topology. A single geodesic snake subject to the appropriate energy functional is indeed able to split freely in order to segment disconnected objects within an image. This plasticity is especially convenient when segmenting complex shapes (*e.g.*, involving significant protrusions) and when no prior assumption about the topology of the object is available. Useful geodesic models, however, have many degrees of freedom, which makes it difficult to constrain shapes and often leads to overfitting in practice. Another drawback of geodesic approaches is their very expensive computational needs, mainly due to the fact that they rely on the evolution of a manifold with a higher number of dimensions than the actual contour to segment. In summary, geodesic snakes based on level-sets are convenient when the shapes of the objects to segment exhibit high variability. When segmenting known shapes, geodesic snakes however tend to be suboptimal.

Parametric snakes [44, 45] are described by a discrete set of coefficients and a continuous parameter. Parametric snakes are usually built in a way that ensures continuity and smoothness. The continuous definition of parametric snakes implies that the segmentation task can be conducted at arbitrary resolution, hence enabling subpixel accuracy. The strength of parametric snakes comes from the fact that they are (i) composed of only a few parameters, (ii) very flexible, (iii) amenable to easy manual edits, and (iv) formed from continuously defined curves that permit refined data analysis. Since they require much fewer coefficients, they result in faster optimization schemes compared to point-based or geodesic snakes. Point-based snakes can in fact be seen as a sampled version of parametric snakes. User interaction is achieved by allowing the user to specify anchor points for the curve to go through [34]. Smoothness and shape constraints can easily be introduced [46], and are particularly suitable when the objects to segment have a reproducible shape that can be naturally encoded within the parameterization. The downside of parametric snakes is that the topology of the curve is imposed by the parameterization. Parametric snakes are thus less suitable than geodesic ones for accommodating changes of topology during optimization, although solutions have been proposed for specific cases [47]. For the interested reader, an extensive literature on parametric snakes has been published, both regarding snake model variants [48, 49, 50, 51] and associated snake energies [45, 52, 53, 54].

Spline-snakes are a subcategory of parametric snakes. They also benefit from a continuous-domain representation and are easy to handle analytically thanks to the spline formalism, which offers local control over the curve. Spline-snakes are also well-suited for semi-automated analysis pipelines and therefore hold a strong potential for the design of user-friendly segmentation frameworks. While remaining mostly automated, they allow for extensive user interaction in term of feedback and manual correction. This aspect plays an important role in their success for image segmentation in biological applications [55, 56]. They are the type of algorithms that are the most relevant to our work.

In the case of spline-snakes, the snake model generates a continuously-defined curve using one or several spline generator as basis function that interpolate a collection of discrete control points in the image. The snake energy which, upon minimization, drives the deformation of the snake curve to fit object boundaries, is generally composed of external and internal forces which attract the curve towards prominent image features (data fidelity) while constraining its rigidity (regularization). Out of these two aspects (*i.e.*, snake model and energy), a whole zoo of spline-snakes with different properties can be elaborated. In this way, spline-snakes can yield both multi-purpose segmentation methods as well as approaches specifically tuned to match the characteristics of particular problems. Hereafter, we give the generic expression of spline-snake models and illustrate different possible constructions through examples of basis functions and of energy terms.

2.3.1 Snake Models

Formally, the parameterization of spline-snakes is expressed as a curve $\mathbf{r}(t)$ on the plane. This curve corresponds to a pair of Cartesian coordinate functions $r_1(t)$ and $r_2(t)$, where $t \in \mathbb{R}$ is a continuous parameter. The one-dimensional functions $r_1(t)$ and $r_2(t)$ are efficiently parameterized by linear combinations of a compactly-supported *spline generator* $\phi : \mathbb{R} \mapsto \mathbb{R}$. The parametric representation of the active contour can be expressed as the vectorial equation

$$\mathbf{r}(t) = \begin{pmatrix} r_1(t) \\ r_2(t) \end{pmatrix} = \sum_{k\in\mathbb{Z}} \mathbf{c}[k]\, \phi(t-k), \tag{2.1}$$

with

$$\{\mathbf{c}[k]\}_{k\in\mathbb{Z}} = \left\{ \begin{pmatrix} c_1[k] \\ c_2[k] \end{pmatrix} \right\}_{k\in\mathbb{Z}} \tag{2.2}$$

a sequence of *control points.* The number M of control points determines the number of degrees of freedom in the model. Small numbers lead to constrained shapes; large numbers offer additional flexibility and yield more complex shapes. The choice of the basis function ϕ is dictated by the specific shapes the snake should be able to adopt and the amount of smoothness it should conserve. In this setting, fast and stable interpolation algorithms can be used to compute the curve [57].

In the parametric representation, snakes made of closed curves can be described with a periodic sequence of control points. In this case, both $r_1(t)$ and $r_2(t)$ are periodic with the same period. When normalized to unity such that $\mathbf{r}(t) = \mathbf{r}(t+1)$ for all $t \in \mathbb{R}$ and divided into M segments, the collection of control points is expressed as $\{\mathbf{c}[k]\}_{k\in\mathbb{Z}}$ with $\mathbf{c}[k] = \mathbf{c}[k+M]$ and $\mathbf{r}(t)$ can be expressed as a finite summation as

$$\mathbf{r}(t) = \sum_{k\in\mathbb{Z}} \mathbf{c}[k]\, \phi(Mt-k) = \sum_{k=0}^{M-1} \sum_{n\in\mathbb{Z}} \mathbf{c}[k+nM]\, \phi(Mt-(k+nM)) \tag{2.3}$$

$$= \sum_{k=0}^{M-1} \mathbf{c}[k] \sum_{n\in\mathbb{Z}} \phi(Mt-Mn-k) = \sum_{k=0}^{M-1} \mathbf{c}[k]\, \phi_{\text{per}}(Mt-k), \tag{2.4}$$

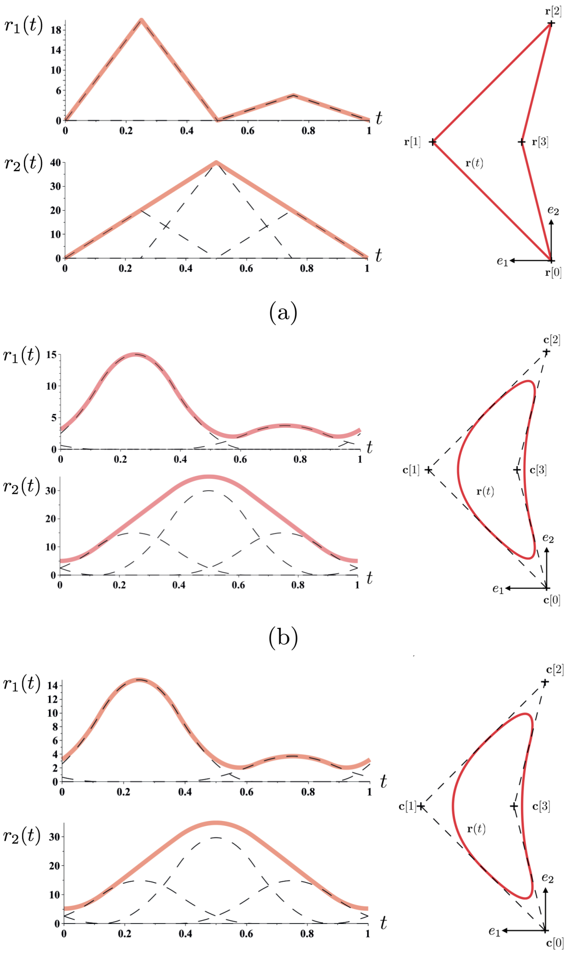
where $\phi_{\text{per}}(t) = \sum_{n\in\mathbb{Z}} \phi(t-Mn)$. In practice, parametric snakes are most commonly defined as closed curves (see [58] for a review), although models handling open curves for the segmentation of lines or boundaries also exist [59, 60, 61, 62, 63]. In the open curve case, the snake is obtained by considering a finite sequence of control points $\{\mathbf{c}[k]\}_{k\in\mathbb{Z}}$ with $\mathbf{c}[k]=0$ for all $k<0$ and $k\geq M$ and expressed as

$$\mathbf{r}(t)=\sum_{k=0}^{M-1}\mathbf{c}[k]\,\phi(Mt-k). \tag{2.5}$$

Example of Snake Models

We here illustrate different models that can be obtained by varying the spline generator ϕ, and how this choice affects the resulting snake properties. In Figure 2.3, we show three curves constructed using the same set of $M=4$ control points along with their two coordinate functions. The first one (Figure 2.3a) is generated using the linear B-spline $\phi(t)=\beta^1(t)=\mathrm{tri}(t)$. The resulting curve is piecewise-linear and of C^0 continuity at the knots, meaning that it is a continuous function, but none of its derivatives are continuous. It is said to interpolate the control points, meaning that the curve goes through them. By increasing the smoothness of the basis function and taking for instance the quadratic B-spline $\phi(t)=\beta^2(t)$ as generator, the resulting curve is now piecewise-quadratic (C^1 continuity at the knots, meaning that it is a continuous function with continuous first derivative) and loses the interpolation property (Figure 2.3b). B-spline generators thus generate piecewise-polynomial curves with different degrees of smoothness. A possible variation on this theme is to build curves that span other function spaces. One can for instance aim at representing ellipses or circles, which calls for basis that reproduce complex exponentials[4]. The third example (Figure 2.3c) is obtained from $\phi(t)=\beta_{\mathrm{E}}(t)$, an exponential spline [64]. The resulting curve is very similar to the one obtained with quadratic B-spline and has in fact the same continuity (C^1 at the knots), but is fundamentally different as it is piecewise-trigonometric. In this way, the choice of the generator can be adapted depending on the desired curve properties in terms of smoothness, approximation, reproduction or refinability properties.

[4] The reproduction of functions of the form $\mathrm{e}^{\pm\mathrm{j}\theta t}$ allows obtaining sines and cosines, which are the building blocks for representing ellipses and circles.



(a)

(b)

(c)

Figure 2.3: Comparison of curves generated by three different basis functions $\phi(t)$ from the same set of control points. (a) Linear B-spline, (b) quadratic B-spline, (c) minimum-support ellipse-reproducing spline [64].

2.3.2 Snake Energies

The evolution of the snake curve to fit object boundaries is formulated as an energy-minimization problem. The quality of the result is thus largely determined by the choice of the energy terms. Specific image energies therefore need to be defined according to the particularities of the image analysis problem. Kass *et al.* [34] originally formulated the snake energy as a linear combination of three terms:

- the *image energy*, E_{image}, which guides the snake towards the boundary of interest and is purely data driven;
- the *internal energy*, E_{internal}, which imposes prior shape information, such as ensuring that the segmented region has smooth boundaries, and serves as a regularizer;
- the *constraint energy*, $E_{\text{constraint}}$, which incorporates strict external conditions imposed by the user.

Let us consider the curve representation Θ which is, in the case of parametric snakes, entirely determined by the the set of control points $\{\mathbf{c}[k]\}_{k=0,\dots,M-1}$. A snake curve $\mathcal{C}(\Theta)$ is thus parameterized by $\mathbf{r} : \mathbb{R} \to \mathcal{C}$, where $\mathbf{r}$ is given by (2.1). The total energy of the snake is written as

$$E_{\text{snake}}(\Theta) = E_{\text{image}}(\Theta) + E_{\text{internal}}(\Theta) + E_{\text{constraint}}(\Theta), \tag{2.6}$$

where Θ is the curve representation. The optimal snake Θ_{opt} is formally obtained as

$$\Theta_{\text{opt}} = \arg\min_{\Theta} E_{\text{snake}}(\Theta). \tag{2.7}$$

Energy minimization is an optimization procedure where the snake representation is iteratively updated so as to reach the minimum of the energy function from its initial position. This starting position, or *initialization*, is usually specified by the user, although many application-dependent techniques exist to automatically provide a first estimate of the position of the target. Many methods exist to minimize the energy functional (*e.g.*, gradient descent, partial differential equations approaches, dynamic programming), and each optimization scheme is usually linked to a particular snake representation. The optimization requires to iteratively update the position of the control points, which is, in the case of spline-snakes, such a low dimensional problem that any minimizer is appropriate. In practice, the fastest approaches are usually gradient-based methods. There, the optimizing process relies on the computation of the partial derivatives of the energy with respect to the control points, $\frac{\partial E_{\text{snake}}}{\partial \mathbf{c}[k]}$.

Example of Snake Energies

We here provide examples of different spline-snake energies belonging to the image, internal and constraints categories.

Image energies incorporate image information to guide the snake to the object of interest. They can be classified into the subfamilies of contour- and region-based energies.

- *Contour-based energies* rely on edge or ridge maps derived from the image [34, 46, 65]. It for instance uses gradient magnitude only, such as in

$$E_{\text{edge}}(\Theta) = -\oint_{\mathcal{C}(\Theta)} |\nabla f(\mathbf{r})| \mathrm{d}r, \tag{2.8}$$

  where f is the input image, $\nabla I(\mathbf{r}) = \left(\frac{\partial I(\mathbf{r})}{\partial r_1}, \frac{\partial I(\mathbf{r})}{\partial r_2}\right)$ is the gradient of the image at position (r_1, r_2) and $\mathcal{C}$ is the snake curve determined by $\{\mathbf{c}[k]\}_{k=0,\dots,M-1}$. Alternatively, it can rely on both components of the gradient vector following

$$E_{\text{edge}}(\Theta) = -\oint_{\mathcal{C}(\Theta)} \langle \nabla I(\mathbf{r}), \mathbf{n}_{\mathbf{r}} \rangle \, \mathrm{d}r, \tag{2.9}$$

  where $\mathbf{n}_{\mathbf{r}}$ denotes the inward-pointing unit normal to the curve at (r_1, r_2). These edge-based energies can provide a good localization of the contour of the object. However, they have a narrow basin of attraction, making their success strongly dependent on the quality of the snake initialization. Several authors have developed alternative solutions to this lack of robustness. The most important ones are the introduction of balloon forces [66], the use of gradient vector-fields defined everywhere in the image domain [67], or multiresolution approaches [44, 68].

- *Region-based energies* aim at maximizing the difference of statistical properties between the inside of the snake and an outside shell [45, 69, 70]. Formally, they are expressed as

$$E_{\mathrm{region}}(\Theta) = \int\int_{\Omega(\Theta)} g(x,y)\mathrm{d}x\mathrm{d}y - \lambda \int\int_{\Omega_\mu(\Theta)\backslash\Omega(\Theta)} g(x,y)\mathrm{d}x\mathrm{d}y, \qquad (2.10)$$

  where g is a (possibly filtered) version of the input image, Ω is the region enclosed by the snake curve, Ω_μ is the region enclosed by a shell (with $\mu > 1$ a dilation parameter), which corresponds to a dilated version of the snake curve, and $\lambda > 0$ is a weighting parameter. Region-based energies have a larger basin of attraction and allow convergence even if explicit edges are not present [71]. However, they provide poor localization compared to edge-based image energies.

Internal energies are responsible for regularizing the snake optimization procedure. Internal energy terms incorporate the desired smoothness or shape properties that the curve must satisfy.

- *General regularization energies* promote smoothness, for instance as in [34] with

$$E_{\mathrm{internal}}(\Theta) = L(\Theta) + \lambda \oint_{\mathcal{C}(\Theta)} |\kappa(\mathbf{r})|^2 \mathrm{d}r, \qquad (2.11)$$

  where L is the length of the curve, κ its curvature, and $\lambda > 0$ a weighting parameter. Other ones ensure that the control points distribute evenly along the curve by enforcing curvilinear reparameterization [65] following

$$E_{\mathrm{internal}}(\Theta) = L(\Theta) + \lambda \oint_{\mathcal{C}(\Theta)} (\|\mathbf{r}'\| - L)^2 \mathrm{d}r. \qquad (2.12)$$

  The optimization process can also sometimes lead to self-intersecting snakes. This phenomenon arises mostly when the image energy forces some control points to move faster than others during the iterative optimization process. An extensive body of research on the intersection problem can be found, with numerous articles presenting different approaches to prevent the intersection of freeform curves and surfaces [72]. When a self-intersection is detected, some authors split their shape descriptor in a way that new smaller snakes are born [47]. Others preserve the topology by introducing self-repulsive forces [73] or stop the optimization routine and ask for user assistance [70]. As the likelihood and type of self-intersections is strongly dependent on the snake curve model at hand, there is no general strategy for preventing these events. The best strategy therefore remains to experiment and identify which solution performs the best for the considered snake curve.

- *Prior-shape energies* incorporate prior knowledge as shape constraints in the energy. They then prevent the snake from diverging too much from a set of given geometrical transformations of a reference shape [32]. They are expressed as
$$E_{\text{shape}}(\Theta) = \|\mathbf{r} - \mathbf{r}_p\|^2_{L_2}, \tag{2.13}$$
where $\mathbf{r}_p$ is the projection of the snake curve onto the space generated by the set of allowed transformations of the reference shape and $\| \cdot \|_{L_2}$ is the continuous L_2 norm.

Constraint energies consist in a set of strict external restrictions that are incorporated as hard constraints in the optimization procedure [70].

In practice, snake energies are composed of a combination of several energy terms belonging to each of these categories. The design of the energy functional is therefore not a straightforward task. It is inspired by the considered image analysis task. The parameters regulating each of the different term composing the energy and the trade-off between them are also fine-tuned to the particular application at hand. In automated pipelines, good parameter values can be searched for by training the full segmentation algorithm on real images for a particular application. This provides a range of acceptable values that guarantee robustness and accuracy in the overall segmentation algorithm.

2.4 A Unification Proposal

We propose to unify the two paradigms presented in Sections 2.2 and 2.3 by designing an image analysis approach that jointly incorporates salient points identification and contour detection. To do so, we need two building blocks:

1. **Hermite splines** (Chapter 3), which are generators yielding curve models that can easily incorporate local modifications;

2. **Steerable wavelet filters** (Chapter 4), which is a family of feature detectors providing multiscale orientation and location information, and which can be tuned to detect specific patterns of interest.

We incorporate these two elements into the *landmark active contour* (Chapter 5), a novel spline-snake algorithm which allows us to bridge the search for object contour and the extraction of feature-based information through the optimization of the snake energy.

On one hand, active contours are efficient in modeling contours in general. On the other hand, features give a sense of important locations in the image and of the properties of their local surroundings. It is thus tempting to bring these two approaches together in a unified algorithm. Features information could in fact be incorporated into other spline-snake algorithms, but the Hermite construction allows for a direct correspondence between control points and landmarks. As we will see, each control point of the landmark snake can be assigned to a landmark: it can be placed at salient locations to ensure that the contour goes through them and then adapt the local behavior of the curve by varying its local tangent. This in turns calls for landmark detection algorithms that provide a measure of how much landmark features are present at each image location and, when present, how they look like. By exploring the design of custom feature detectors using steerable filters and relying on Hermite splines, our landmark snake thus brings together features and contour extraction in a unified, flexible and robust algorithm for object detection.

Chapter 3

Taming the Hermite Splines

In this chapter, we present the Hermite spline interpolation scheme, which is the first building block of our approach mixing points and curves. Hermite interpolation was characterized by I. J. Schoenberg [74, 75]. The problem is stated as that of interpolating a function from samples at integer locations of its derivatives up to a given order. The existence and regularity of the resulting interpolant have been studied in all generality in these seminal papers.

In the following, we revisit the Hermite splines covered by the theory of Schoenberg and go beyond by introducing novel results uncovering some of their properties[1]. We first review the formulation of first-order Hermite interpolation, as it is the most relevant for our final application. This leads us to the definition of cubic Hermite splines, two functions with very particular properties. Given a spline scheme, a whole collection of characteristics in terms of approximation, optimality, or families of basis can be derived. Here, we provide some of these elements for the particular case of cubic Hermite splines. We study the approximation power, smoothness properties and minimum mean square error (MMSE) optimality of this interpolation scheme. Finally, we briefly propose an higher-level interpretation and discuss alternative families of Hermite splines in a spirit analogous to the general framework of L-splines proposed in [78, 79, 80].

[1] This chapter is composed of a mixture of published material and manuscripts under preparation. Published results include the content of Section 3.1, adapted from [76], and of Section 3.3, adapted from [77]

Through the chapter, we shall often use the notation $f(t)$ for the continuous function f with $t \in \mathbb{R}$ and $x[k]$ for the discrete sequence $\{x[k]\}_{k\in\mathbb{Z}}$. In addition, the sequence of samples of a continuous function f at the integers is denoted by $f(k) = f(t)|_{t=k}$. For typesetting reasons, we write column vectors in the text as $\mathbf{v}_{\mathrm{col}} = (v_1, \dots, v_N)$. In equations, they appear as true column vectors. Therefore,

$$\mathbf{v}_{\mathrm{col}} = (v_1, \dots, v_N) = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}.$$

In contrast, row vectors are denoted by $\mathbf{v}_{\mathrm{row}} = \begin{bmatrix} v_1 & \dots & v_N \end{bmatrix}$ both in equations and in the text. Also, given a function $f(t)$ with $t \in \mathbb{R}$, we denote its continuous-time Fourier transform by $\hat{f}(\omega)$ with $\omega \in \mathbb{R}$. For a discrete sequence $x[k]$ with $k \in \mathbb{Z}$, we refer to its discrete-time Fourier transform as $X(\mathrm{e}^{\mathrm{j}\omega})$, with $\mathrm{j} = \sqrt{-1}$ and $\omega \in \mathbb{R}$. We indicate the standard vector scalar product and associated norm as $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, respectively. Inner products and norms on function spaces are denoted in a similar way, with the function space indicated as a subscript. For instance, the classical L_2 inner product and associated norm are designated as $\langle \cdot, \cdot \rangle_{L_2}$ and $\|\cdot\|_{L_2}$, respectively. The conjugate of complex numbers and functions is denoted by a star exponent as $\cdot^*$.

We recall that splines are generally defined as piecewise polynomial functions with some amount of regularity at their knots. The knots correspond to the joining points where the polynomial pieces meet. Unless stated otherwise, we always remain in a uniform cardinal spline interpolation setting, meaning firstly that the spline basis are constructed as integer shifts of a single generator and, secondly, that knots are located either at the integers or at multiples of a fixed integer value.

3.1 Definition

In its most basic formulation, Hermite interpolation generates a continuously differentiable function from two discrete sequences that correspond to samples of a function and its derivative, respectively. The discrete sequences are continuously interpolated with Hermite-spline basis functions. Here, we first state the mathematical definition of Hermite interpolation and then discuss important properties of the resulting spline basis.

Formally, Schoenberg [74, 75] defines the cardinal cubic Hermite interpolation problem as follows. Knowing the discrete sequences of numbers $c[k]$ and $d[k]$, $k \in \mathbb{Z}$, we look for a continuously defined function $f_{\mathrm{Her}}(t)$, $t \in \mathbb{R}$, satisfying $f_{\mathrm{Her}}(k) = c[k]$, $f'_{\mathrm{Her}}(k) = d[k]$ for all $k \in \mathbb{Z}$, such that f_{Her} is piecewise polynomial of degree at most 3 and once differentiable with continuous derivative at the integers. The existence and uniqueness of the solution is guaranteed [74, Theorem 1] provided that $c[k]$ and $d[k]$ are in $S'(\mathbb{Z})$, which denotes the space of sequences of slow growth[2]. In addition, the spline function f_{Her}, which is the unique solution of the first-order cardinal Hermite interpolation problem, is explicitly given [74] by

$$\begin{aligned} f_{\mathrm{Her}}(t) &= \sum_{k\in\mathbb{Z}} \left(c[k]\phi_1(t-k) + d[k]\phi_2(t-k)\right) \\ &= \sum_{k\in\mathbb{Z}} \left(f_{\mathrm{Her}}(k)\phi_1(t-k) + f'_{\mathrm{Her}}(k)\phi_2(t-k)\right). \end{aligned} \tag{3.1}$$

The analytical expressions of the two *cubic Hermite splines* ϕ_1, ϕ_2 are found in [75, 81] as

$$\phi_1(t) = \begin{cases} (2\,|t|+1)\,(|t|-1)^2 & \text{for } 0 \le |t| \le 1 \\ 0 & \text{for } |t| > 1, \end{cases} \tag{3.2}$$

$$\phi_2(t) = \begin{cases} t\,(|t|-1)^2 & \text{for } 0 \le |t| \le 1 \\ 0 & \text{for } |t| > 1. \end{cases} \tag{3.3}$$

It is worth emphasizing that the existence of ϕ_1 and ϕ_2 is in itself a remarkable and non-trivial result. In addition to their fairly simple analytical expression, the cubic Hermite splines have other important properties. First, they are of finite support[3], being restricted to the $[-1, 1]$ interval. A very fundamental property of this construction is that the generating functions ϕ_1, ϕ_2 and their derivatives ϕ'_1, ϕ'_2 satisfy the joint interpolation conditions

$$\phi_1(k) = \delta[k], \quad \phi'_2(k) = \delta[k], \quad \phi'_1(k) = 0, \quad \phi_2(k) = 0, \tag{3.4}$$

for all $k \in \mathbb{Z}$, where $\delta[k]$ is the discrete unit impulse. The functions and their first derivative are depicted in Figure 3.1, where the interpolation properties can easily

[2] We say that a sequence $\{x[k]\}_{k\in\mathbb{Z}}$ is in $S'(\mathbb{Z})$ if there exist two real constants N and C such that $|x[k]| \le C(|k|^N + 1)$ for all $k \in \mathbb{Z}$.

[3] The *support* of a function is here defined as the smallest closed interval outside of which the function is equal to zero.

be observed. The functions ϕ_1 and ϕ_2 are deeply intricated. In (3.1), ϕ_1 interpolates points of the sequence c while the first derivative of ϕ_2 interpolates points of the sequence d. Considering $c[k] = f(k)$ and $d[k] = f'(k)$ for an input function f, f_{Her} is the solution to the problem of interpolating f and its first derivative at the integers, as exposed in [74]. Note that ϕ_1 and ϕ_2 are cubic polynomials in $[-1, 0]$ and $[0, 1]$ and, by extension, in each interval $[k, k+1]$ for $k \in \mathbb{Z}$, which is the motivation for referring to them as cubic Hermite splines. They are differentiable with continuous derivatives at the knots points $t = k$ (*i.e.*, at the integers).

The space of functions generated by Hermite splines is defined as

$$\left\{ \sum_{k\in\mathbb{Z}} (c[k]\phi_1(\cdot - k) + d[k]\phi_2(\cdot - k)) = \sum_{k\in\mathbb{Z}} \begin{bmatrix} c[k] \\ d[k] \end{bmatrix}^T \boldsymbol{\phi}(t-k) : c[k], d[k] \in S'(\mathbb{Z}) \right\}, \tag{3.5}$$

with $\boldsymbol{\phi} = (\phi_1, \phi_2)$. Due to the Hermite interpolation conditions, the expansion coefficients $c[k]$ and $d[k]$ in (3.5) coincide with $f(k)$, $f'(k)$, the samples of the function to be interpolated and its first derivative on the integer grid. The space (3.5) contains splines of degree 3 with regularity C^1 at the knots, which can also be interpreted as a space of cubic splines that can accommodate quadratic transitions. The generated functions are thus C^1-continuous piecewise-cubic polynomials with knots at integer locations. Such functions can also be represented by cubic B-splines with double knots at the integers. The effect of the double knot is to reduce by one the continuity (C^2) of the cubic B-splines. Note that, in the literature, the condition $c, d \in l_2(\mathbb{Z})$ is more commonly found. However, there is no conceptual difficulty in considering $c, d \in S'(\mathbb{Z})$ since the cubic Hermite basis functions ϕ_1 and ϕ_2 are compactly supported. Sequences in $S'(\mathbb{Z})$ are actually required in order to make the link with classical B-splines, as shown hereafter.

3.1.1 Connection with Classical Cubic and Quadratic B-Splines

In order to better understand the properties of cubic Hermite-splines, we highlight their connections to classical cubic and quadratic B-splines. As a reminder, B-spline interpolation generates piecewise-polynomial functions with C^{n-1} continuity, where n is the degree of the B-spline. The B-spline of degree n is defined as

$$\beta^n(t) = (\beta^{n-1} * \beta^0)(t) = (\beta^0 * \ldots * \beta^0)(t), \tag{3.6}$$

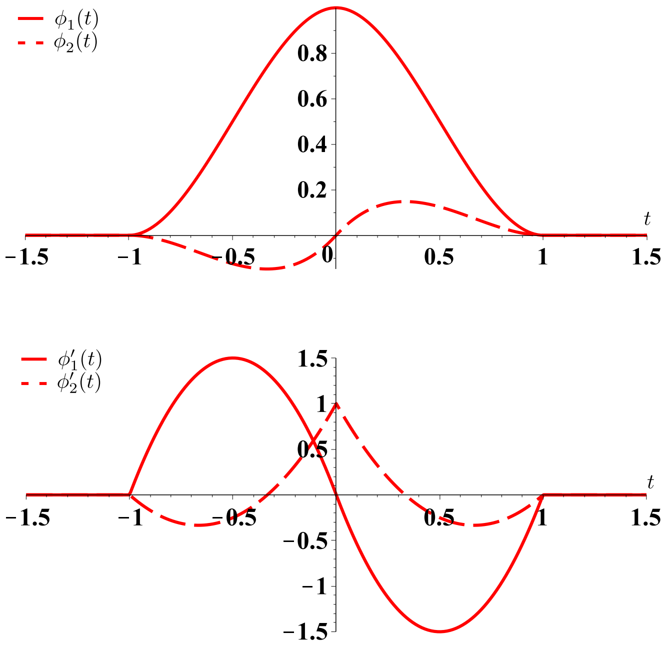


Figure 3.1: Cubic Hermite splines ϕ_1 and ϕ_2. The two functions and their derivatives are vanishing at the integers with the exception of $\phi_1(0) = 1$ and $\phi_2'(0) = 1$ (interpolation conditions). They are supported in $[-1, 1]$.

with

$$\beta^0(t) = \begin{cases} 1 & t \in \left[-\frac{1}{2}, \frac{1}{2}\right], \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

The causal B-spline of degree n, denoted as β^n_+, is obtained as $\beta^n_+(t) = \beta^n\left(t - \frac{n+1}{2}\right)$. A detailed review of the properties of B-splines can be found in [82, 83].

We establish the link between Hermite and B-splines through the Fourier transforms of the Hermite spline generators. This yields

$$\widehat{\boldsymbol{\phi}}(\omega) = \begin{bmatrix} \widehat{\phi}_1(\omega) \\ \widehat{\phi}_2(\omega) \end{bmatrix} = \begin{bmatrix} -\frac{12(\omega \sin\omega + 2\cos\omega - 2)}{\omega^4} \\ -\frac{4\mathrm{j}(2\omega - 3\sin\omega + \omega\cos\omega)}{\omega^4} \end{bmatrix}. \tag{3.8}$$

One remarks that the zero of the denominator is compensated by the numerator at $\omega = 0$. Therefore, $\widehat{\boldsymbol{\phi}}(0)$ is well-defined. These equations can be rewritten in matrix-vector form as

$$\widehat{\boldsymbol{\phi}}(\omega) = \widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})\ \widehat{\boldsymbol{\rho}}(\omega), \tag{3.9}$$

with

$$\widehat{\boldsymbol{\rho}}(\omega) = \begin{bmatrix} \widehat{\rho}_1(\omega) \\ \widehat{\rho}_2(\omega) \end{bmatrix} = \begin{bmatrix} \frac{1}{(\mathrm{j}\omega)^4} \\ \frac{1}{(\mathrm{j}\omega)^3} \end{bmatrix}, \tag{3.10}$$

and

$$\widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega}) = \begin{bmatrix} 24 - 12\mathrm{e}^{-\mathrm{j}\omega} - 12\mathrm{e}^{\mathrm{j}\omega} & -6\mathrm{e}^{-\mathrm{j}\omega} + 6\mathrm{e}^{\mathrm{j}\omega} \\ -6\mathrm{e}^{-\mathrm{j}\omega} + 6\mathrm{e}^{\mathrm{j}\omega} & -8 - 2\mathrm{e}^{-\mathrm{j}\omega} - 2\mathrm{e}^{\mathrm{j}\omega} \end{bmatrix}. \tag{3.11}$$

The vector $\boldsymbol{\rho} = (\rho_1, \rho_2)$, which is the inverse Fourier transform of $\widehat{\boldsymbol{\rho}}$, contains the Green's functions of the operators $\mathrm{L}_1 = \mathrm{D}^4 = \frac{\mathrm{d}^4}{\mathrm{d}t^4}$ and $\mathrm{L}_2 = \mathrm{D}^3 = \frac{\mathrm{d}^3}{\mathrm{d}t^3}$. The operators L_1 and L_2 generate the classical cubic and quadratic B-splines, respectively. Relying on the link between the Green's function of an operator and its associated spline [84], we state the relation between the Hermite spline generators $\boldsymbol{\phi} = (\phi_1, \phi_2)$ and $\boldsymbol{\rho} = (\rho_{\mathrm{D}^4}, \rho_{\mathrm{D}^3})$, which are the Green's functions of D^4 and D^3. Their explicit expressions are

$$\rho_{\mathrm{D}^4}(t) = \mathcal{F}^{-1}\left\{\frac{1}{(\mathrm{j}\cdot)^4}\right\}(t) = \frac{1}{12}t^3\mathrm{sgn}(t), \tag{3.12}$$

$$\rho_{\mathrm{D}^3}(t) = \mathcal{F}^{-1}\left\{\frac{1}{(\mathrm{j}\cdot)^3}\right\}(t) = \frac{1}{4}t^2\mathrm{sgn}(t). \tag{3.13}$$

By inverting the 2×2 Fourier matrix $\widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})$ from (3.11), we find that

$$\widehat{\boldsymbol{\rho}}(\omega) = \widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})^{-1}\ \widehat{\boldsymbol{\phi}}(\omega). \tag{3.14}$$

From the rules of matrix inversion, the entries of $\widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})^{-1} = \widehat{\mathbf{S}}(\mathrm{e}^{\mathrm{j}\omega})$ are ratios of trigonometric polynomials. Consequently, each of these entries can be reduced through partial fraction decomposition. Because it is composed of weighted sums of simple fractions, the discrete-time inverse Fourier transform of $\widehat{\mathbf{S}}$ is well-defined. It is also guaranteed to yield a unique sequence of matrices

$$\mathbf{S}[k] = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \widehat{\mathbf{S}}(\mathrm{e}^{\mathrm{j}\omega})\ \mathrm{e}^{\mathrm{j}\omega k}\ \mathrm{d}\omega \tag{3.15}$$

of slow growth. Hence, we conclude that

$$\boldsymbol{\rho}(t) = \sum_{k\in\mathbb{Z}} \mathbf{S}[k]\ \boldsymbol{\phi}(t-k), \tag{3.16}$$

which proves that the Green's functions ρ_{D^4} and ρ_{D^3}, as well as their integer shifts, can be reproduced by $\{\boldsymbol{\phi}(\cdot - k)\}_{k\in\mathbb{Z}}$. The specific form of (3.16) follows from the interpolation property of the generators,

$$s(t) = \sum_{k\in\mathbb{Z}} \big(s(k)\phi_1(t-k) + s'(k)\phi_2(t-k)\big), \tag{3.17}$$

which is valid for any function in the span of the cubic Hermite splines. In particular, we have that

$$\rho_{\mathrm{D}^4}(t) = \frac{1}{12}t^3\mathrm{sgn}(t) = \sum_{k\in\mathbb{Z}}\left(\frac{1}{12}k^3\mathrm{sgn}(k)\phi_1(t-k) + \frac{1}{4}k^2\mathrm{sgn}(k)\phi_2(t-k)\right). \tag{3.18}$$

The expansion (3.18) motivates our choice to consider sequences $c, d \in S'(\mathbb{Z})$ instead of $l_2(\mathbb{Z})$.

It is known [85, Chapter 6] that cubic B-splines admit a unique stable expansion of the type

$$s(t) = \sum_{k\in\mathbb{Z}} a[k]\rho_{\mathrm{D}^4}(t-k), \tag{3.19}$$

where a is a sequence of coefficients of slow growth. This in turn implies that the space of cubic Hermite splines includes the space of cubic splines as a subspace. The same holds true for the quadratic splines generated by $\{\rho_{\mathrm{D}^3}(\cdot - k)\}_{k\in\mathbb{Z}}$. Based on (3.9), we express ϕ_1 as

$$\phi_1 = 12\Delta_+^2 \rho_{\mathrm{D}^4} + 6\Delta_+ \rho_{\mathrm{D}^3}, \tag{3.20}$$

where Δ_+^2 is the second-order finite-difference operator with $\Delta_+ f(t) = f(t) - f(t-1)$. The expression of ϕ_2 in terms of ρ_{D^4} and ρ_{D^3} is obtained through an analogous calculation:

$$\phi_2 = 6\Delta_+ \rho_{\mathrm{D}^4} + 2(\Delta_+^2 \rho_{\mathrm{D}^3} - 6\rho_{\mathrm{D}^3}). \tag{3.21}$$

The Hermite spline space can thus be seen as the union of the cubic and quadratic spline spaces.

Causal cubic B-splines can be written in the two following equivalent ways

$$\begin{aligned} \beta_+^3(t) &= \Delta_+^4 \rho_{\mathrm{D}^4}(t) & (3.22)\\ &= \tfrac{1}{6}\phi_1(t-1) + \tfrac{4}{6}\phi_1(t-2) + \tfrac{1}{6}\phi_1(t-3) + \tfrac{1}{2}\phi_2(t-1) - \tfrac{1}{2}\phi_2(t-2), & (3.23) \end{aligned}$$

where Δ_+^4 is the 4th-order finite-difference operator with Δ_+ defined as above. Similarly, the Hermite interpolation property (3.17) can be used to obtain the corresponding expression for causal quadratic B-splines as

$$\beta_+^2(t) = \Delta_+^3 \rho_{\mathrm{D}^3}(t) = \tfrac{1}{2}\phi_1(t-1) + \tfrac{1}{2}\phi_1(t-2) + \phi_2(t-1) - \phi_2(t-2). \tag{3.24}$$

As cubic and quadratic B-splines can be expressed in terms of ϕ_1 and ϕ_2, it implies that they are included in the family of Hermite splines. The above expressions help understanding another remarkable property of the Hermite spline space. In the case of cubic splines, a compactly supported basis function (*i.e.*, a B-spline) is constructed by applying the discrete version Δ_+^4 of the operator D^4 to its Green's function ρ_{D^4}, as described in (3.22). Quadratic B-splines are constructed in the same way by applying the discrete version Δ_+^3 of the operator D^3 to the Green's function ρ_{D^3}. Either of the functions $\Delta_+^2\rho_{\mathrm{D}^4}$ and $\Delta_+\rho_{\mathrm{D}^3}$ is only partially localized and still includes a linear trend. However, the combination of both in (3.20) and (3.21) results in the cancellation of all residual polynomial components outside of the $[-1, 1]$ interval. As a consequence, the cubic Hermite basis functions have a support of size 2 which, remarkably, is shorter than that of cubic B-splines.

Finally, since cubic B-splines reproduce polynomials of degree up to 3, this property automatically extends to the Hermite spline space, yielding

$$t^m = \sum_{k\in\mathbb{Z}} \big(k^m \phi_1(t-k) + m\, k^{m-1}\phi_2(t-k)\big) \tag{3.25}$$

for $m = 0, 1, 2, 3$.

3.1.2 Uniqueness and Stability of the Representation

Uniqueness and stability of the interpolation result are guaranteed by the so-called Riesz-basis conditions for the generator $\boldsymbol{\phi}$. By definition (see for instance [85, Section 6.2.3]), the vector function $\boldsymbol{\phi} = (\phi_1, \phi_2)$ generates a Riesz basis if and only if there exist two constants $0 < A \leq B < +\infty$ such that

$$A\|\mathbf{a}\|_{\ell_2} \leq \left\| \sum_{k\in\mathbb{Z}} \mathbf{a}[k]^T \boldsymbol{\phi}(\cdot - k) \right\|_{L_2} \leq B\|\mathbf{a}\|_{\ell_2}, \tag{3.26}$$

for all $\mathbf{a}[k] = (c[k], d[k])$, with $c[k], d[k] \in \ell_2(\mathbb{Z})$. As a reminder, the ℓ_2 norm of a vector of sequences is defined as $\|\mathbf{a}[k]\|_{\ell_2}^2 = \sum_{i=1}^N \|a_i[k]\|_{\ell_2}^2$ for $\mathbf{a}[k] = (a_1[k], \ldots, a_N[k])$.

The Riesz basis property is easier to verify in the Fourier domain. To that end, we first compute the Fourier Gram matrix of the basis, which is given by

$$\begin{aligned}
\widehat{\mathbf{G}}(\mathrm{e}^{\mathrm{j}\omega}) &= \textstyle\sum_{k\in\mathbb{Z}} \widehat{\boldsymbol{\phi}}(\omega + 2\pi k)\widehat{\boldsymbol{\phi}}(\omega + 2\pi k)^{*T} \\
&= \begin{bmatrix} \displaystyle\sum_{k\in\mathbb{Z}} \langle \phi_1, \phi_1(\cdot - k)\rangle_{L_2} \mathrm{e}^{-\mathrm{j}\omega k} & \displaystyle\sum_{k\in\mathbb{Z}} \langle \phi_1, \phi_2(\cdot - k)\rangle_{L_2} \mathrm{e}^{-\mathrm{j}\omega k} \\ \displaystyle\sum_{k\in\mathbb{Z}} \langle \phi_2, \phi_1(\cdot - k)\rangle_{L_2} \mathrm{e}^{-\mathrm{j}\omega k} & \displaystyle\sum_{k\in\mathbb{Z}} \langle \phi_2, \phi_2(\cdot - k)\rangle_{L_2} \mathrm{e}^{-\mathrm{j}\omega k} \end{bmatrix} \qquad (3.27) \\
&= \begin{bmatrix} \frac{26}{35} + \frac{9\mathrm{e}^{-\mathrm{j}\omega}}{70} + \frac{9\mathrm{e}^{\mathrm{j}\omega}}{70} & -\frac{13\mathrm{e}^{-\mathrm{j}\omega}}{420} + \frac{13\mathrm{e}^{\mathrm{j}\omega}}{420} \\ \frac{13\mathrm{e}^{-\mathrm{j}\omega}}{420} - \frac{13\mathrm{e}^{\mathrm{j}\omega}}{420} & \frac{2}{105} - \frac{\mathrm{e}^{-\mathrm{j}\omega}}{140} - \frac{\mathrm{e}^{\mathrm{j}\omega}}{140} \end{bmatrix}.
\end{aligned}$$

The Gram matrix $\widehat{\mathbf{G}}(\mathrm{e}^{\mathrm{j}\omega})$ is Hermitian symmetric for every ω and is 2π-periodic in ω. Next, we recall that the Fourier equivalent of the Riesz-basis requirement (3.26) is

$$0 < A^2 = \min_{\omega\in[0,2\pi]} \lambda_{\min}(\mathrm{e}^{\mathrm{j}\omega}) \leq \max_{\omega\in[0,2\pi]} \lambda_{\max}(\mathrm{e}^{\mathrm{j}\omega}) = B^2 < +\infty, \tag{3.28}$$

where $\lambda_{\max}(\mathrm{e}^{\mathrm{j}\omega})$ and $\lambda_{\min}(\mathrm{e}^{\mathrm{j}\omega})$ denote the maximum and minimum eigenvalues of $\widehat{\mathbf{G}}(\mathrm{e}^{\mathrm{j}\omega})$ at frequency ω, respectively. Interestingly, the constants A and B obtained from (3.28) are the optimal ones. This result is classical in the case of a single generator φ (see for instance [86, 87]). In this situation, we have $\lambda_{\min}(\mathrm{e}^{\mathrm{j}\omega}) = \lambda_{\max}(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{k\in\mathbb{Z}}|\widehat{\varphi}(\omega+2\pi k)|^2$. The case of multiple generators, which is required in the Hermite scenario, is covered by [88, Theorem 2.1]. Knowing the Gram matrix (3.27), we simply compute the minimum and maximum eigenvalues and take the supremum and infimum among $\omega \in [0, 2\pi]$. This calculation yields the exact values of the Riesz bounds, namely

$$(A, B) = (210^{-\frac{1}{2}}, 1), \tag{3.29}$$

with the worst case of (3.28) obtained for $\omega = 0$.

3.1.3 Connection with Bézier Representations

The presence of explicit derivatives in the Hermite representation is reminiscent of Bézier curves, which are popular tools in computer graphics. The Hermite spline representation can in fact easily be converted to Bézier curves, but the converse is not true in general. Bézier curves can have different left and right derivative values by construction. They therefore have more degrees of freedom, but are also less regular. The additional smoothness granted by the Hermite spline representation is especially useful for practical applications, as it implies that less parameters have to be optimized. It also provides additional intrinsic regularity. In [76], we show how these two representations relate to each other through Bernstein polynomials. We also provide the expression of Hermite splines in terms of Bézier curves.

3.2 Approximation Error

Reconstruction using sequences of samples plays an essential role in communications and signal processing as it creates a bridge between discrete and analog signals. One of the fundamental results of this field is Shannon's sampling theorem, reviewed and extensively discussed in [57], which demonstrates that perfect reconstruction is possible for sampled bandlimited signals. In its initial formulation, Shannon's theory involves uniform samples of the original continuous-time signal. Papoulis [89] proposed an extension of this framework referred to as *generalized sampling*. The underlying idea is as follows. A signal can be fully characterized, and thus reconstructed, with many different kind of data, uniform samples being only one of them. In its initial formulation, the sampling problem is designed for bandlimited signals and involves ideal filters, making it mostly unusable in practice although theoretically beautiful [90]. Alternatively, extensions without bandlimited constraints relying on spline-based models for signal representation have been proposed [91]. The considered problem is then the reconstruction of a square-integrable function (possibly not bandlimited) from the outputs of M linear shift-invariant systems sampled at $1/M$ of the reconstruction rate. The reconstruction subspace is specified such that the final output corresponds to a bandlimited function, a spline, or a wavelet expansion. Another interesting aspect of this extensions is that approximation is considered in a broader sense, *i.e.*, not necessarily perfect but consistent, in the sense that it produces the same set of measurement as the original input of the system.

Here, we discuss[4] the performance of Hermite interpolation and compare it against other existing approximation schemes. The difference between a reference curve and its cubic Hermite spline interpolation defines the quality of the approximation. More precisely, we are interested in the rate at which the error decreases when approximating any smooth enough function f from $\boldsymbol{\phi} = (\phi_1, \phi_2)$ and its integer shifts, possibly as a function of the smoothness of f. As shown in (3.22) and (3.24), one can perfectly reproduce both quadratic and cubic B-splines with cubic Hermite splines. It means, in particular, that the approximation power of the Hermite spline space is at least as good as that of the space of cubic B-splines. It can actually be shown that the Hermite spline space corresponds to the direct sum of the cubic and quadratic B-spline spaces. The multifunction analysis from [92, Theorem 1] can then be applied to Hermite basis functions. By doing so, the cubic Hermite splines can be shown to have the same asymptotic approximation order as cubic B-splines. In the following, we derive such asymptotic constants assessing the quality of approximation results. The analysis we provide is both qualitative and quantitative—qualitative as it highlights the particularities that are inherent to each scheme, and quantitative as it provides the asymptotic form of the approximation error. Our goal is to put Hermite splines in the broader context of approaches approximating both a function and its first derivative from sequences

[4] This work was realized in close collaboration with J. Fageot, Biomedical Imaging Group, EPFL, Switzerland.

of their samples.

3.2.1 Generalized Signal Approximation Scheme

In order to obtain an approximation of a continuous signal from a collection of its samples in a generalized sampling scheme, the two following elements are needed.

1. A collection of *sampling functions* $\tilde{\boldsymbol{\varphi}} = (\tilde{\varphi}_1, \dots, \tilde{\varphi}_N)$ which are possibly generalized functions in $S'(\mathbb{R})$ and which describe how the samples are drawn from the input signal,

2. a collection of *interpolation functions* $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_N)$ which belong to $L^2(\mathbb{R})$ and which are used to reconstruct a continuous curve connecting the sample points,

with N the number of different pairs of sampling and interpolation functions being considered. The reason why sampling is said to be *generalized* lies in the fact that the sampling functions can yield sequences that are more complex than direct values of the signal at the sampling points. This set of pairs of sampling and interpolation functions fully characterize an approximation scheme.

The sampling and reconstruction problem is then formally defined as follows. The continuous-time input $f(t)$, $t \in \mathbb{R}$ is sampled through a collection of N filters with impulse responses $\tilde{\varphi}_i(-\cdot)$ for $i = 1, ..., N$, yielding the discrete measurements (*i.e.*, generalized samples) vector $\mathbf{g}[k] = (g_1[k], \dots, g_N[k])$. Each component of the measurement vector can be expressed as $g_i[k] = \langle f, \tilde{\varphi}_i(\cdot - Nk) \rangle_{L_2}$, where $\tilde{\varphi}_i(t)$ is the ith sampling function. When considering N sampling functions, the samples are picked at every integer multiple of N. In the following, we shall only consider functions f such that the inner product between f and $\tilde{\varphi}_i(\cdot - Nk)$ is well-defined for $i = 1, \dots, N$ and $k \in \mathbb{Z}$. This implies that the smoothness constraints imposed on f depend on the regularity of the sampling functions. For instance, if $N = 1$ and $\tilde{\varphi} = \delta'$, f should typically be differentiable with a continuous derivative. For simplicity, we choose not to discuss this aspect further and always assume that f is selected to be regular enough. With a slight abuse of notation, we keep the L_2 inner product subscript since the functions f we study will in particular live in $L_2(\mathbb{R})$, although being much smoother in practice to compensate for $\boldsymbol{\varphi}$[5]. Each resulting

[5] The notation $\langle \cdot, \cdot \rangle_{L_2}$ applies to functions in L_2 and is formally not correct when considering

measurement sequence must be well-defined in the l_2 sense, or, in other words, $\mathbf{g} \in (l_2(\mathbb{Z}))^N$. Measurements are then combined to reconstruct the approximation $\tilde{f}$ of f onto the subspace

$$V(\boldsymbol{\varphi}) = \left\{ \sum_{i=1}^{N} \sum_{k \in \mathbb{Z}} c_i[k] \varphi_i(x - Nk) : \mathbf{c}[k] \in (l_2(\mathbb{Z}))^N \right\} \tag{3.30}$$

with the collection of interpolation functions $\boldsymbol{\varphi}$. The approximation is not necessarily perfect as f might not belong to the span of $V(\boldsymbol{\varphi})$. The reconstruction scheme is usually at least required to be consistent, meaning that one should recover the original measurements $\{g_i[k]\}_{i=1,...,N, k \in \mathbb{Z}}$ from f when sampling the approximation $\tilde{f}$. This is formally expressed as

$$\langle \tilde{f}, \tilde{\varphi}_i(\cdot - Nk) \rangle_{L_2} = \langle f, \tilde{\varphi}_i(\cdot - Nk) \rangle_{L_2} = g_i[k] \tag{3.31}$$

for $k \in \mathbb{Z}$ and $i = 1, ..., N$. The approximation $\tilde{f}$ is given by

$$\tilde{f}(t) = \sum_{i=1}^{N} \sum_{k \in \mathbb{Z}} \langle f, \tilde{\varphi}_i(\cdot - Nk) \rangle_{L_2} \varphi_i(x - Nk) = \sum_{i=1}^{N} \sum_{k \in \mathbb{Z}} g_i[k] \varphi_i(x - Nk). \tag{3.32}$$

The best approximation is obtained when the pairs of sampling and interpolation functions are properly chosen such that $\tilde{f}$ is the orthogonal projection of f onto $V(\boldsymbol{\varphi})$. In general, it is worth noting that, for $N = 1$ and $\tilde{\varphi} = \delta$, one gets back to a classical interpolation problem since $\tilde{f}(t) = \sum_{k \in \mathbb{Z}} f(k) \varphi(t - k)$. For more details, we refer the interested reader to [91, 93]. We now provide some concrete examples of generalized sampling schemes.

Sampling and Reconstruction with Cubic B-splines

Reconstruction with cubic B-splines is the simplest case and provides a good introductory example [84]. In this setting, $N = 1$ and only one pair of sampling and interpolation functions is thus considered. The cubic B-spline reconstruction of a

sampling functions that belong to other function spaces. In cases where a $\tilde{\varphi} \notin L_2$ is considered, $\langle f, \tilde{\varphi} \rangle_{L_2}$ should be understood only for functions f that are smooth enough. For instance, if $\tilde{\varphi} \in \mathcal{S}'$, then only functions $f \in \mathcal{S}$ are allowed.

signal f is expressed as

$$\tilde{f}(t) = \sum_{k\in\mathbb{Z}} \left\langle f, \sum_{n\in\mathbb{Z}} (b^3)^{-1}[n]\delta(\cdot - n) \right\rangle_{L_2} \beta^3(t-k), \tag{3.33}$$

where $(b_1^3)^{-1}$ is the *direct B-spline filter* [84]. It is better expressed in the Z-transform-domain as

$$\frac{1}{B_1^3(z)} = \frac{6}{z + 4 + z^{-1}}. \tag{3.34}$$

Therefore, the sampling function corresponds to

$$\tilde{\varphi} = \sum_{k\in\mathbb{Z}} (b^3)^{-1}[k]\delta(\cdot - k), \tag{3.35}$$

and the interpolation function is β^3, the cubic B-spline.

Generalized Sampling Framework without Band-Limited Constraints

A generalized sampling framework without band-limited constraints is proposed in [91]. Defined as an extension of Papoulis' generalized sampling theory, it can be used in particular for approximating a function relying on its samples and samples of its first derivative, although the proposed framework is more general. It is of special interest for us as it offers a sensible candidate to compare against Hermite interpolation.

The basic ingredients are a collection of N sampling functions $\tilde{\boldsymbol{\varphi}}$ and a single *generating function* φ_G. The interpolation functions are constructed from φ_G following (3.36). Under proper assumptions on the $\tilde{\varphi}_i$ for $i = 0, \ldots, N$ and φ_G [91, Theorem 1], one can approximate a smooth enough function f from its measurements g_i. The best approximation is expressed as (3.32). The collection of N interpolation functions φ_i is constructed from φ_G as

$$\varphi_i(t) = \sum_{k\in\mathbb{Z}} q_i[k]\varphi_G(t-k) \text{ for } i = 1, ..., N. \tag{3.36}$$

Since (3.36) is a discrete convolution, it can be computed in Fourier domain as $\widehat{\varphi_i}(\omega) = \widehat{\varphi_G}(\omega)Q_i(\mathrm{e}^{\mathrm{j}\omega})$. The elements of the sequences q_i are given in the Z-transform-domain by

$$\begin{bmatrix} Q_1(z) & \ldots & Q_N(z) \end{bmatrix} = \begin{bmatrix} 1 & z^{-1} & \ldots & z^{-N-1} \end{bmatrix} \widehat{\mathbf{A}}^{-1}_{\tilde{\boldsymbol{\varphi}}\varphi_G}(z^N), \tag{3.37}$$

with $\mathbf{A}_{\tilde{\varphi}\varphi_G}[k]$ an $N \times N$ matrix which entries are equal to

$$[\mathbf{A}_{\tilde{\varphi}\varphi_G}]_{i,j}[k] = \langle \tilde{\varphi}_i(\cdot - Nk), \varphi_G(\cdot - j + 1)\rangle_{L_2}, \tag{3.38}$$

where $\tilde{\varphi}_i$ is the ith sampling function. The approximation $\tilde{f}$ is the orthogonal projection of f onto $V(\varphi_G)$, the space generated by linear combinations of translates of φ_G, built like (3.30). Therefore, perfect reconstruction (*i.e.*, $f = \tilde{f}$) is achieved for all f that are included in $V(\varphi_G)$.

In the particular case of reconstruction from samples of a function f and of its first derivative f', we set $N = 2$, $\tilde{\varphi}_1(t) = \delta(t)$ and $\tilde{\varphi}_2(t) = \delta'(t - \frac{1}{2})$. This particular case is referred to as *interlaced derivative sampling* [91, VI.B]. Therefore, $g_1[k] = f(2k)$ and $g_2[k] = f'(2k+1)$. We choose $\varphi_G(t)$ to be the cubic B-spline to allow for a fair comparison with cubic Hermite splines. We obtain

$$\widehat{\mathbf{A}}_{\tilde{\varphi}\varphi_G}(z) = \begin{bmatrix} \frac{2}{3} & \frac{z+1}{6z} \\ 0 & \frac{z-1}{2z} \end{bmatrix}, \tag{3.39}$$

and

$$\begin{bmatrix} Q_1(z) & Q_2(z) \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & \frac{(z-4)z+1}{2-2z^2} \end{bmatrix}. \tag{3.40}$$

The interpolation functions are obtained as

$$\widehat{\varphi}_1(\omega) = \frac{3\mathrm{e}^{-2\mathrm{j}\omega}\left(-1 + \mathrm{e}^{\mathrm{j}\omega}\right)^4}{2\omega^4}, \tag{3.41}$$

$$\widehat{\varphi}_2(\omega) = \frac{\mathrm{e}^{-2\mathrm{j}\omega}\left(-1 + \mathrm{e}^{\mathrm{j}\omega}\right)^4\left(1 + \mathrm{e}^{\mathrm{j}\omega}\left(-4 + \mathrm{e}^{\mathrm{j}\omega}\right)\right)}{\left(2 - 2\mathrm{e}^{2\mathrm{j}\omega}\right)\omega^4}. \tag{3.42}$$

Their expression is given in Fourier domain because φ_1 and φ_2 appear not to have a closed-form expression in the spatial domain. Upon inverting their Fourier transforms, one notices that these two functions are not of finite support. In addition, we have that $\varphi_1(k) = \varphi_2'(2k+1) = \delta[k]$, and $\varphi_1'(2k+1) = \varphi_2(k) = 0$ for all $k \in \mathbb{Z}$. This is reminiscent of the joint interpolation conditions (3.4) in the Hermite case.

Generalized Sampling using Hermite Splines

Hermite interpolation is also an instance of Papoulis' generalization without band limited constraints. In this framework, $N = 2$ and the sampling functions correspond to $\tilde{\varphi}_1(t) = \delta(t)$ and $\tilde{\varphi}_2(t) = \delta'(t)$. The interpolation functions are the cubic Hermite splines $\varphi_1(t) = \phi_1(\frac{t}{2})$ and $\varphi_2(t) = 2\phi_2(\frac{t}{2})$ given by (3.2) and (3.3), respectively. A function f is approximated from its measurements g_1 and g_2 as

$$\tilde{f}(t) = \sum_{i=1}^{2} \sum_{k\in\mathbb{Z}} g_i[k]\varphi_i(t-2k) = \sum_{k\in\mathbb{Z}} f(2k)\phi_1\left(\frac{t-2k}{2}\right) + 2f'(2k)\phi_2\left(\frac{t-2k}{2}\right). \tag{3.43}$$

3.2.2 Definition and Characterization of the Approximation Error

Now that we understand how to build an approximation scheme, we focus on how to evaluate and compare different ones. Any suitable pair of sampling and interpolation functions generates an approximation scheme. However, it comes at no surprise that some choices of pairs are better than others. The quality of the approximation is evaluated in terms of approximation error which, for a function f, is computed as the L_2 norm of the difference between the function and its approximation. As discussed in 3.2.1, we only consider functions f that are sufficiently regular such that $\langle f, \tilde{\varphi}_i\rangle_{L_2}$ is properly defined for $i = 1, \dots, N$.

In all generality, let us consider N sampling functions $\tilde{\boldsymbol{\varphi}} = (\tilde{\varphi}_1, \dots, \tilde{\varphi}_N) \in (S'(\mathbb{R}))^N$ and interpolation functions $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_N) \in (L_2(\mathbb{R}))^N$. As mentioned above, the sampling functions specify the measurements we have on a given function f. We consider the approximation space $V(\boldsymbol{\varphi})$ as in (3.30). The approximation of the function f is given by the projection

$$\mathcal{Q}f = \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} \langle f, \tilde{\varphi}_i(\cdot - Nk)\rangle_{L_2} \varphi_i(\cdot - Nk). \tag{3.44}$$

Since $V(\boldsymbol{\varphi})$ is a vector space, one can define the orthogonal projector which minimizes the approximation error. The weights yielding the least-square solution are obtained by imposing a particular condition [94] on the sampling functions $\tilde{\boldsymbol{\varphi}}$, namely that

$$\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d = \begin{pmatrix} \varphi_{1,d} \\ \vdots \\ \varphi_{N,d} \end{pmatrix} = \mathbf{G}_{\boldsymbol{\varphi}}(\omega)^{-1}\boldsymbol{\varphi}, \tag{3.45}$$

where $\mathbf{G}_{\boldsymbol{\varphi}}$ is the Gram matrix associated to $\boldsymbol{\varphi}$, which is given by

$$\mathbf{G}_{\boldsymbol{\varphi}}(\omega) = \sum_{k\in\mathbb{Z}} \widehat{\boldsymbol{\varphi}}(\omega+2k\pi)\widehat{\boldsymbol{\varphi}}^{*T}(\omega+2k\pi) \tag{3.46}$$

$$= \begin{pmatrix} \sum_{k\in\mathbb{Z}} \widehat{\varphi}_1(\omega+2k\pi)\widehat{\varphi}_1^*(\omega+2k\pi) & \sum_{k\in\mathbb{Z}} \widehat{\varphi}_1(\omega+2k\pi)\widehat{\varphi}_2^*(\omega+2k\pi) \\ \sum_{k\in\mathbb{Z}} \widehat{\varphi}_2(\omega+2k\pi)\widehat{\varphi}_1^*(\omega+2k\pi) & \sum_{k\in\mathbb{Z}} \widehat{\varphi}_2(\omega+2k\pi)\widehat{\varphi}_2^*(\omega+2k\pi) \end{pmatrix}. \tag{3.47}$$

This particular collection of sampling functions are called the *dual functions* associated to $\boldsymbol{\varphi}$ and are denoted by $\boldsymbol{\varphi}_d$. When $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d$, the operator $\mathcal{Q}$ is the orthogonal projector over $V(\boldsymbol{\varphi})$, and is denoted as $\mathcal{P}$. In this situation, the approximation of the function f is expressed as

$$\mathcal{P}f = \sum_{i=1}^{N}\sum_{k\in\mathbb{Z}} \langle f, \varphi_{i,d}(\cdot - Nk)\rangle_{L_2} \varphi_i(\cdot - Nk)\,. \tag{3.48}$$

We now have all the tools needed to formally define the approximation error. For a general pair of $\tilde{\boldsymbol{\varphi}}$ and $\boldsymbol{\varphi}$, it is expressed as

$$E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f) = \|f - \mathcal{Q}f\|_{L_2} \tag{3.49}$$

and, for the particular case of the orthogonal projector, as

$$E_{\boldsymbol{\varphi}}(f) = \|f - \mathcal{P}f\|_{L_2}. \tag{3.50}$$

This means that $E_{\boldsymbol{\varphi}}(f) = E_{\boldsymbol{\varphi}}^{\boldsymbol{\varphi}_d}(f)$. We also have that $E_{\boldsymbol{\varphi}}(f) \leq E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f)$, and the equality is only reached when $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d$.

We introduce an additional important parameter, the sampling step T, which defines a step size on the uniform sampling grid. Up to now, our approximation schemes were relying on (generalized) samples taken on the integer grid, corresponding to $T = 1$. Changing the sampling step T modifies the coarseness of the approximation. In particular, we expect reduced T to yield better approximation as the input function gets sampled more and more finely. At the limit case where T goes to zero, we expect the error to converge to zero for functions that are sufficiently smooth. We reformulate the approximation space for $T > 0$ as

$$V_T(\boldsymbol{\varphi}) = \left\{ \sum_{i=1}^{N}\sum_{k\in\mathbb{Z}} c_i[k]\varphi_i\left(\frac{\cdot}{T} - Nk\right) \ : \ \mathbf{c}[k] \in (\ell_2(\mathbb{Z}))^N \right\} \tag{3.51}$$

and the approximation of f from $\tilde{\boldsymbol{\varphi}}$ and $\boldsymbol{\varphi}$ as

$$\mathcal{Q}_T f = \sum_{i=1}^{N} \sum_{k \in \mathbb{Z}} \left\langle f, \frac{1}{T} \tilde{\varphi}_i \left(\frac{\cdot}{T} - Nk \right) \right\rangle_{L_2} \varphi_i \left(\frac{\cdot}{T} - Nk \right). \tag{3.52}$$

Similarly, the orthogonal projector is redefined as

$$\mathcal{P}_T f = \sum_{i=1}^{N} \sum_{k \in \mathbb{Z}} \left\langle f, \frac{1}{T} \varphi_{i,d} \left(\frac{\cdot}{T} - Nk \right) \right\rangle_{L_2} \varphi_i \left(\frac{\cdot}{T} - Nk \right) \tag{3.53}$$

where the $\varphi_{i,d}$ are given by (3.45). The errors become

$$E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f, T) = \| f - \mathcal{Q}_T f \|_{L_2}, \tag{3.54}$$

$$E_{\boldsymbol{\varphi}}(f, T) = \| f - \mathcal{P}_T f \|_{L_2}. \tag{3.55}$$

Given an approximation scheme, the first question is whether or not the approximation error $E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f, T)$ actually decreases to zero as $T \rightarrow 0$. This is ensured by imposing specific conditions on $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ [94] that we specify below. Assuming that the error goes to zero as the sampling step vanishes, additional conditions can be formulated to ensure that the error $E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f, T)$ decreases following $C_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f) T^L$. We refer to the parameter $L \geq 1$ as the *rate of decay*, while the constant $C_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f)$ is called the *asymptotic constant* of the approximation error. The asymptotic constant must be different than zero. It is quantified knowing the rate of decay L as

$$C_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f) = \lim_{T \rightarrow 0} T^{-L} E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f, T). \tag{3.56}$$

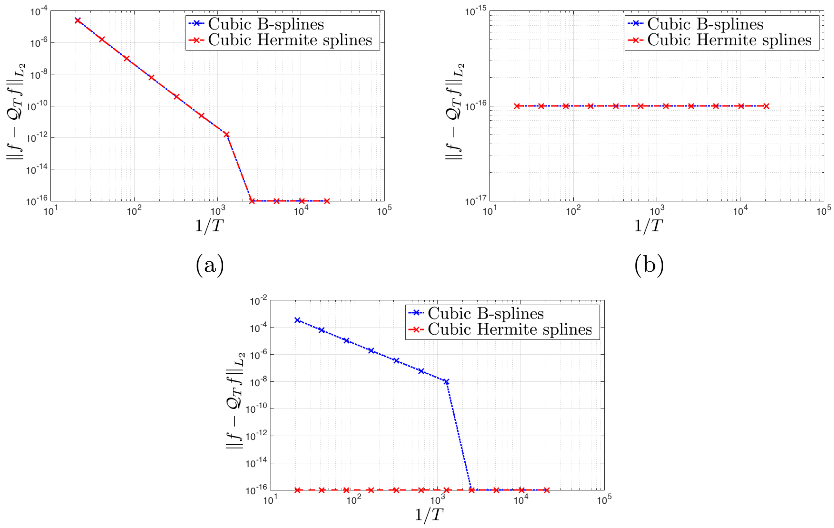
These different concepts are more easily understood visually. To do so, we consider the following simple experiment. Given a collection of M samples of a function f observed over the finite $[0, 1]$ interval, we construct two different approximations. The first one is the cubic B-spline interpolation of f from $\{f(k)\}_{k=1,\dots,M}$. The second one is the cubic Hermite interpolation of f from $\{f(2k), f'(2k)\}_{k=1,\dots,\frac{M}{2}}$. Note that considering only samples at even locations for the Hermite case allows having a comparable amount of information in the two schemes. We let T, the sampling step, reduce to 0. For each value of T, we compute the approximation error $\| f - \mathcal{Q}_T f \|_{L_2}$. Results are shown in log-log scale in Figure 3.2 for three interesting cases. The first one illustrates the error when approximating a function that is neither in the span of the cubic B-splines, nor in that of the cubic Hermite splines. The two approximation errors behave in a similar way and are indistinguishable. In the second case, a cubic polynomial function is approximated. Since such a function lives in the space generated by both the cubic B-splines and cubic Hermite splines, the approximation is perfect straight from the start. Finally, the third case shows the approximation error when considering a quadratic polynomial that is of C^1 but not C^2 regularity. More precisely, we consider

$$f(t) = \begin{cases} (t-0.5)^2 & t \geq 0.5, \\ 0 & \text{elsewhere.} \end{cases} \tag{3.57}$$

The function (3.57) is therefore in the span of the cubic Hermite splines, but not in that of the cubic B-splines. Provided that the sampling rate is reasonable (*i.e.*, that the function is not undersampled), it is immediately perfectly reconstructed with Hermite splines and the error remains null. Conversely, the error of the cubic B-spline approximation decreases to zero as T diminishes. The rate of decay of the approximation error can be read as the slope of the decreasing curve in the plots. Here, we are dealing with cubic B-splines and therefore observe a rate of decay of $L = 4$ [92]. The asymptotic constant corresponds to the value at the origin, obtained by extending the approximation error line to the left until it crosses the y-axis.

We now make a list of desired hypotheses on the interpolation functions $\boldsymbol{\varphi}$ and the sampling functions $\tilde{\boldsymbol{\varphi}}$ to study the errors $E_{\boldsymbol{\varphi}}(f,T)$ and $E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T)$ in terms of rate of decay and asymptotic constant. Such an analysis is provided in the case of a single pair of sampling and interpolation functions in [92], and in [94] for the general case of $N \geq 1$. While [92] is targeted towards a signal processing audience, [94] is written for approximation theorists and is thus more technical. In the following, we use the notations of [92] and reformulate the results of [94] for the sake of readability. We formally enunciate each of them and describe why they are of interest.

1. **Sensible approximation scheme.** First, $V_T(\boldsymbol{\varphi})$ (given by (3.51)) should be a well-defined subspace of $L_2(\mathbb{R})$. Each function belonging to this space must have a unique and stable representation, in the sense that there exists a unique sequence of coefficients describing it. This is guaranteed if we hypothesize that



(a)

(b)

(c)

Figure 3.2: Illustration of the order of approximation. Approximation error of cubic B-splines and cubic Hermite splines for decreasing sampling step size when considering (a) a trigonometric function, which does not belong to the span of any of the considered approximation functions; (b) a cubic polynomial function, which belongs to the span of both the cubic B-splines and cubic Hermite splines; (c) a function of C^1 but not C^2 regularity, which belongs to the span of the cubic Hermite splines but not of the cubic B-splines. Note that 10^{-16} corresponds to the machine precision and should therefore be understood as zero. Large jumps in the curves are imputed to rounding errors for values that are close to the machine precision.

the interpolation functions $\boldsymbol{\varphi}$ are in $L_2(\mathbb{R})$ and satisfy the Riesz condition

$$A \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} |c_i[k]|^2 \leq \left\| \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} c_i[k]\varphi_i(\cdot - Nk) \right\|_{L_2}^2 \leq B \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} |c_i[k]|^2 \quad (3.58)$$

for any $c_i[k] \in l_2(\mathbb{Z})$ with $i = 1, \ldots, N$ and $0 < A \leq B < \infty$.

2. **Rate of decay of the minimal error ($\mathcal{P}_T f$, orthogonal projector).** Now, assuming that the representation is stable, the approximation error should vanish as T goes to zero. For this, the optimal error (*i.e.*, the error obtained with the orthogonal projector $\mathcal{P}_T$) must at least have a rate of decay of $L = 1$. This is guaranteed if the so-called Strang and Fix condition of order 1 is satisfied [95]. More generally, the synthesis functions are at least *of order* L if and only if they satisfy the Strang and Fix condition of order L. The optimal error then decreases with at least an Lth-order decay.

Proposition 1 (Reformulation of [94, Lemma 1] using the notations of [92, II.D]). *Let $\boldsymbol{\varphi}$ be such that $\{\varphi_i(\cdot - Nk)\}_{i=1,\ldots,N,k\in\mathbb{Z}}$ is a Riesz basis. Then, for $L \geq 1$, the two following conditions are equivalent:*

- $E_{\boldsymbol{\varphi}}(f, T) = O(T^L)$,
- *One can reproduce polynomial functions up to order $L - 1$, i.e.,*

$$t^\ell = \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} c_{i,\ell}[k]\varphi_i(t - Nk) \text{ for } \ell = 0, \ldots, L - 1. \quad (3.59)$$

The first condition sets an upper bound on the optimal error using the orthogonal projector $E_{\boldsymbol{\varphi}}(f, T)$. It implies that the error decays at least as T^L, although it can be faster. The second condition is the definition of synthesis functions $\boldsymbol{\varphi}$ of order L, and corresponds to the Strang and Fix condition of order L. It states that a set of synthesis function of order L allows reproducing polynomials of degree up to $L - 1$. The Strang and Fix condition of order 1 therefore translates to the reproduction of constants, also known as partition-of-unity condition. An important consequence is that synthesis functions of order L are also of order $L-1, L-2, \ldots, 1$. The maximal value of L for which the synthesis functions still satisfy the Strang and Fix condition is referred to as the *maximal order*.

3. **Rate of decay of the general error ($\mathcal{Q}_T f$).** The approximation error from the orthogonal projector $\mathcal{P}_T$ is now under control. Our next desideratum is to consider other projectors $\mathcal{Q}_T$ which are possibly suboptimal, but still guarantee the same rate of decay of the error. This is satisfied if $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ are *quasi-biorthonormal* of order $L \geq 1$. The quasi-biorthonormality of order L corresponds to the combination of the Strang and Fix condition of order L as stated in (3.59) and the equality between the moments of $\tilde{\boldsymbol{\varphi}}$ and $\boldsymbol{\varphi}_d$ up to order L as given in (3.60) below.

   Proposition 2 (Reformulation of [94, Definition 2 and Theorem 3] using the notations of [92, II.D])**.** *Let $\boldsymbol{\varphi}$ be such that $\{\varphi_i(\cdot - Nk)\}_{i=1,\dots,N,k\in\mathbb{Z}}$ is a Riesz basis, and let us consider functions f that are bounded with bounded derivatives up to order L. We assume that $E_{\boldsymbol{\varphi}}(f,T) = O(T^L)$ for some $L \geq 1$. Then, the two following conditions are equivalent:*

   - $E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T) = O(T^L)$,
   - *For the dual function $\boldsymbol{\varphi}_d$ given by* (3.45),

   $$\int_{\mathbb{R}} t^\ell \tilde{\boldsymbol{\varphi}}(t)\,\mathrm{d}t = \int_{\mathbb{R}} t^\ell \boldsymbol{\varphi}_d(t)\,\mathrm{d}t \textit{ for all } \ell = 0,\dots,(L-1). \tag{3.60}$$

4. **Asymptotic constants.** The previous points allow characterizing the rate of decay of the errors from $\mathcal{P}_T$ and $\mathcal{Q}_T$. The last remaining step is the computation of the asymptotic constants. To do so, we first introduce the kernels

   $$\mathcal{E}_{\min}(\omega) = 1 + \widehat{\boldsymbol{\varphi}}^{*T}(\omega)\mathbf{G}_{\boldsymbol{\varphi}}^{-1}(\omega)\widehat{\boldsymbol{\varphi}}(\omega), \tag{3.61}$$

   $$\mathcal{E}_{\text{res}}(\omega) = (\widehat{\tilde{\boldsymbol{\varphi}}} - \widehat{\boldsymbol{\varphi}_d})^{*T}(\omega)\mathbf{G}_{\boldsymbol{\varphi}}(\omega)(\widehat{\tilde{\boldsymbol{\varphi}}} - \widehat{\boldsymbol{\varphi}_d})(\omega), \tag{3.62}$$

   $$\mathcal{E}(\omega) = \mathcal{E}_{\min}(\omega) + \mathcal{E}_{\text{res}}(\omega), \tag{3.63}$$

   where $\mathbf{G}_{\boldsymbol{\varphi}}$ is the Gram matrix (3.46). The kernel $\mathcal{E}_{\min}$ relates to the *minimum error* case achieved using the orthogonal projector (*i.e.*, $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d$), and $\mathcal{E}_{\text{res}}$ to the *residual error* arising when using sampling functions that differ from the dual functions. We furthermore remark that $\mathcal{E}_{\text{res}}(\omega) = 0$ when $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d$, as expected. The asymptotic constants are finally computed as follows.

   Proposition 3 (Reformulation of [94, Theorem 4] using the notations of [92, III])**.** *Let $\boldsymbol{\varphi}$ be such that $\{\varphi_i(\cdot - Nk)\}_{i=1,\dots,N,k\in\mathbb{Z}}$ is a Riesz basis, and let us* *consider functions f that are square integrable with square integrable derivatives up to order L. If the synthesis functions $\boldsymbol{\varphi}$ are of maximal order L, then $\mathcal{E}_{\min}^{(2L)}(0) \neq 0$ and*

$$E_{\boldsymbol{\varphi}}(f,T) \underset{T\to 0}{\sim} \sqrt{\frac{\mathcal{E}_{\min}^{(2L)}(0)}{(2\pi)^{2L}(2L)!}}\|f^{(L)}\|_{L_2}T^L. \tag{3.64}$$

In addition, if $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ are quasi-biorthonormal of order L, then $\mathcal{E}^{(2L)}(0) \neq 0$ and

$$E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T) \underset{T\to 0}{\sim} \sqrt{\frac{\mathcal{E}^{(2L)}(0)}{(2\pi)^{2L}(2L)!}}\|f^{(L)}\|_{L_2}T^L. \tag{3.65}$$

We hereafter provide a sketch of proof for Proposition 3. The asymptotic constant of the minimum error, (3.64), is obtained with the following reasoning. First, under the assumptions of Proposition 3, the functions $\mathcal{E}_{\min}$ and $\mathcal{E}_{\mathrm{res}}$ are $2L$ times differentiable at 0. The Taylor expansion of order L of $E_{\boldsymbol{\varphi}}^2$ is well-defined and is given by

$$E_{\boldsymbol{\varphi}}^2(f,T) = \sum_{\ell=0}^{L} \frac{\mathcal{E}_{\min}^{(2\ell)}(0)}{(2\pi)^{2\ell}(2\ell)!}\|f^{(\ell)}\|_{L_2}^2 T^{2\ell} + O(T^{2(L+1)}). \tag{3.66}$$

Since the $\boldsymbol{\varphi}$ are of maximal order L, the first $2(L-1)$ terms of (3.66) vanish, *i.e.*, $\mathcal{E}_{\min}^{(0)}(0) = \ldots = \mathcal{E}_{\min}^{(2(L-1))}(0) = 0$ and the first non-vanishing term is $\mathcal{E}_{\min}^{(2L)}(0)$. In the general case, we have that $\mathcal{E}^{(2\ell)}(0) = \mathcal{E}_{\min}^{(2\ell)}(0) + \mathcal{E}_{\mathrm{res}}^{(2\ell)}(0)$. We first notice that $\mathcal{E}_{\min}^{(2\ell)}(0) \geq 0$ and $\mathcal{E}_{\mathrm{res}}^{(2\ell)}(0) \geq 0$ for all $0 \leq \ell \leq L$. Additionally, when $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ are quasi-biorthonormal of order L, we have that $\mathcal{E}_{\mathrm{res}}^{(2\ell)}(0) = 0$ for all $\ell \leq L$. We then obtain the asymptotic constant of the general approximation error as (3.65). For the interested reader, all these arguments are fully developed in [94].

Choosing a pair of $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ that satisfy the quasi-biorthonormality condition of order L therefore ensures that the error of the resulting approximation scheme is decaying in a similar way as the minimum error case (*i.e.*, when $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_d$).

3.2.3 Numerical Comparisons

With these tools in hands, we proceed to compare different approximation schemes. Our focus is on the approximation of both f and its derivative f'. We first assume that both f and its first Lth derivatives belong to $L_2(\mathbb{R})$. In order to be of interest for us, an approximation scheme should approximate f and f' relying on $\mathcal{Q}_T f$ and $(\mathcal{Q}_T f)'$, respectively. We study the two components of the approximation error vector

$$\mathbf{E}_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T) = \begin{pmatrix} \|f - \mathcal{Q}_T f\|_{L_2} \\ \|f' - (\mathcal{Q}_T f)'\|_{L_2} \end{pmatrix}, \tag{3.67}$$

with $E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}} = \|f - \mathcal{Q}_T f\|_{L_2}$ and $E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}} = \|f' - (\mathcal{Q}_T f)'\|_{L_2}$.

The quality of the approximation is quantified by the rate of decay L and by the two asymptotic constants. Note that, in (3.56), the asymptotic constant depends on the function f being approximated. From (3.65), we are able to precisely identify the nature of this dependency. It can thus be relieved by considering

$$\mathbf{C}_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}} = \begin{pmatrix} \lim_{T\to 0} T^{-L} \|f^{(L)}\|_{L_2}^{-1} E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T) \\ \lim_{T\to 0} T^{-(L-1)} \|f^{(L-1)}\|_{L_2}^{-1} E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T) \end{pmatrix}. \tag{3.68}$$

We rely on these constants as criterion to compare the approximation power of different schemes having the same rate of decay L. The asymptotic constant of $E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$ can be computed with the same tools as that of $E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$. We indeed have that

$$(\mathcal{Q}_T f)' = \frac{1}{T} \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} \left\langle f, \frac{1}{T} \tilde{\varphi}_i \left(\frac{\cdot}{T} - k \right) \right\rangle_{L_2} \varphi_i' \left(\frac{\cdot}{T} - k \right). \tag{3.69}$$

This however does not fit our framework, as the approximation of the function must rely on samples of the function itself. In other words, the sampling functions $\tilde{\boldsymbol{\varphi}}$ must be applied to f'. Relying on integration by parts, (3.69) is modified as

$$(\mathcal{Q}_T f)' = \frac{1}{T} \sum_{i=1}^{N} \sum_{k\in\mathbb{Z}} \left\langle f', \frac{1}{T} \tilde{\varphi}_{i,\mathrm{int}} \left(\frac{\cdot}{T} - k \right) \right\rangle_{L_2} \varphi_i' \left(\frac{\cdot}{T} - k \right), \tag{3.70}$$

where the new sampling functions $\tilde{\varphi}_{i,\mathrm{int}}(t) = -\int_{-\infty}^{t} \tilde{\varphi}_i(x)\mathrm{d}x$ are better defined in the Fourier domain as

$$\widehat{\tilde{\varphi}_{i,\mathrm{int}}}(\omega) = -\frac{1}{\mathrm{j}\omega} \widehat{\tilde{\varphi}}_i(\omega). \tag{3.71}$$

From (3.70), one can see that the rate of decay of the error on the derivative is equal to $L-1$. One power of T indeed gets lost in the differentiation process.

The first considered scheme is cubic B-spline interpolation ($N=1$). The approximation is obtained as

$$\left(\mathcal{Q}_T f\right)(t)=\sum_{k\in\mathbb{Z}}\left\langle f,\sum_{n\in\mathbb{Z}}(b^3)^{-1}[n]\delta(\cdot-n)\right\rangle_{L_2}\beta^3\left(\frac{t}{T}-k\right), \tag{3.72}$$

with $(b^3)^{-1}$ defined in (3.34).

In the Hermite framework, sampling functions are taken to be $\tilde{\varphi}_1=\delta$ and $\tilde{\varphi}_2=\delta'$. The interpolation functions are the cubic Hermite splines, namely $\varphi_1=\phi_1$ from (3.2) and $\varphi_2=\phi_2$ from (3.3). The approximation scheme in the Hermite setting therefore corresponds to

$$\left(\mathcal{Q}_T f\right)(t)=\sum_{k\in\mathbb{Z}} f(2k)\,\phi_1\left(\frac{t}{2T}-k\right)+2f'(2k)\,\phi_2\left(\frac{t}{2T}-k\right). \tag{3.73}$$

As a generalized sampling scheme involving the function and its first derivative, interlaced derivative sampling is a good candidate for comparison. In this setting, the sampling functions are $\tilde{\varphi}_1=\delta$ and $\tilde{\varphi}_2=\delta'\left(\cdot-\frac{1}{2}\right)$. The interpolation functions are constructed as described in Section 3.2.1 and yield φ_1 and φ_2 given by (3.41) and (3.42). The resulting approximation is expressed as

$$\left(\mathcal{Q}_T f\right)(t)=\sum_{k\in\mathbb{Z}} f(2k)\,\varphi_1\left(\frac{t}{T}-2k\right)+f'(2k+1)\,\varphi_2\left(\frac{t}{T}-2k\right). \tag{3.74}$$

The shift of $\frac{1}{2}$ on the derivative sampling is chosen so as to optimally tune this scheme to get the lowest error when considering both $E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$ and $E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$ together. Other shift values are observed to result in larger overall approximation errors, as illustrated by those obtained with a shift of zero, corresponding to $\tilde{\varphi}_2=\delta'$.

In neither of these frameworks do the sampling functions correspond to the dual functions. For this reason, the relevant quantity is $E_{\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}(f,T)$. In all cases, results can however be compared to the corresponding optimal asymptotic constant obtained from (3.61). The main properties and asymptotic constants for cubic B-spline, interlaced derivative sampling and cubic Hermite splines are provided in Table 3.1. All methods have an order of approximation of $L=4$ on the function. One order

Table 3.1: Comparison of approximation methods.

Approximation Method	Cubic B-splines	Generalized Sampling (shift of $\frac{1}{2}$)	Generalized Sampling (shift of 0)	Hermite Splines
Interpolating	✗	✓	✓	✓
Finite Support	✓	✗	✗	✓
Closed-form expression	✓	✗	✗	✓
Rate of Decay ($E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	4	4	4	4
Asymptotic Constant ($E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	$\frac{1}{362880}$	$\frac{1}{362880}$	$\frac{1}{567000}$	$\frac{1}{362880}$
Ratio to optimal ($E_{1,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	$\frac{3}{10}$	$\frac{3}{10}$	$\frac{15}{32}$	$\frac{3}{10}$
Rate of Decay ($E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	3	3	3	3
Asymptotic Constant ($E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	$\frac{1}{30240}$	$\frac{1}{30240}$	$\frac{1}{23625}$	$\frac{1}{30240}$
Ratio to optimal ($E_{2,\boldsymbol{\varphi}}^{\tilde{\boldsymbol{\varphi}}}$)	1	1	$\frac{25}{32}$	1

gets lost when approximating the derivative, and the error on f' therefore decreases with an order of $L = 3$. With the exception of interlaced derivative sampling with a shift of 0, their behavior is equivalent in terms of the value of their asymptotic constant, both regarding the estimation of the function and of its first derivative. Hermite interpolation is in this sense not a unique way of approximating a function and its first derivative, even when one requires the error to remain close to optimal. The notable remaining difference is that the Hermite scheme provides functions that are simultaneously of finite support, which is not the case for interlaced derivative sampling (see (3.41) and (3.42)), and interpolating, which is not the case for cubic B-splines.

As a conclusion, there exists several approximation schemes given a function and its first derivative. However, the cubic Hermite construction is the only one which results in reconstruction functions that have a closed-form expression, a finite support, and that are interpolating.

3.3 Optimality

The optimality properties of B-splines have been the topic of several research work, which will be recalled in this chapter. We provide here analogous results for cubic Hermite splines.

In the generalized sampling framework mentioned in Section 3.2, reconstruction is based on variational principles [96]. In parallel, a statistical interpretation can be given to deterministic spline-fitting methods through the minimal mean square-error (MMSE) criterion, extending their use to the optimal reconstruction of random processes in the least-square sense. In the following, we study optimality properties of Hermite splines first in a purely variational formulation, and then in the statistical framework of MMSE estimation.

3.3.1 Variational Formulation: Smoothness Properties

Variational formulations are commonly used to describe regularization constraints in various kind of optimization procedures. The energy minimization properties of spline interpolation are a strong justification for using splines in signal estimation and reconstruction. Cubic splines can for instance be shown to have a minimum-curvature property [97, 98], which tends to favor solutions with a low average curvature, thus imposing some natural amount of smoothness. We hereafter show that the Hermite interpolant has similar properties.

Optimal Smoothness of Hermite Splines

Cubic B-splines are said to provide *optimal smoothness* because of their minimum-norm (or minimum-curvature) property. This result states that, given a set of nodes in the $[a, b]$ interval, the cubic spline interpolator is the one that minimizes

$$\int_a^b |f''(t)|^2 \, \mathrm{d}t \tag{3.75}$$

among all interpolating functions f. This result is shown with some variations in [99, Theorem 3.1.1] and [44, Theorem 1]. The optimal smoothness property of Hermite splines, described in Theorem 1, is closely related to that of the cubic B-splines.

Theorem 1. *Let $c, d \in \ell_2(\mathbb{Z})$. Among all possible functions $f : \mathbb{R} \to \mathbb{R}$ with f, f', $f'' \in L_2(\mathbb{R})$, the optimal one that minimizes*

$$\|f''\|_{L_2}$$

subject to $c[k] = f(k)$ *and* $d[k] = f'(k)$ *for* $k \in \mathbb{Z}$ *is the Hermite interpolator* f_{Her} *defined as*

$$f_{\mathrm{Her}}(t) = \sum_{k\in\mathbb{Z}} \left(c[k]\phi_1(t-k) + d[k]\phi_2(t-k)\right). \tag{3.76}$$

Proof. The existence and uniqueness of the Hermite interpolant given the sequences c and d is ensured by Schoenberg's work, as mentioned at the beginning of Section 3.1. We then have to show that, for any f satisfying the two interpolation constraints $f(k) = c[k]$ and $f'(k) = d[k]$, we have

$$\|\mathrm{D}^2 f\|_{L_2}^2 = \|\mathrm{D}^2 f_{\mathrm{Her}}\|_{L_2}^2 + \|\mathrm{D}^2(f - f_{\mathrm{Her}})\|_{L_2}^2, \tag{3.77}$$

where D^2 denotes the second derivative operator $\frac{\mathrm{d}^2}{\mathrm{d}t^2}$. If $f = f_{\mathrm{Her}}$, the second term in (3.77) vanishes, implying that f_{Her} is optimal among all solutions that interpolate f. Let us take $g = f - f_{\mathrm{Her}}$. Then,

$$\|\mathrm{D}^2 g + \mathrm{D}^2 f_{\mathrm{Her}}\|_{L_2}^2 = \|\mathrm{D}^2 f_{\mathrm{Her}}\|_{L_2}^2 + 2\langle \mathrm{D}^2 f_{\mathrm{Her}}, \mathrm{D}^2 g\rangle_{L_2} + \|\mathrm{D}^2 g\|_{L_2}^2, \tag{3.78}$$

and proving (3.77) reduces to demonstrating that

$$\langle \mathrm{D}^2 f_{\mathrm{Her}}, \mathrm{D}^2 g\rangle_{L_2} = 0. \tag{3.79}$$

By definition, f_{Her} is given by

$$\begin{aligned} f_{\mathrm{Her}}(t) &= \sum_{k\in\mathbb{Z}} \left(c[k]\phi_1(t-k) + d[k]\phi_2(t-k)\right) && (3.80)\\ &= \left(\phi_1 * \sum_{k\in\mathbb{Z}} c[k]\delta(\cdot - k)\right)(t) + \left(\phi_2 * \sum_{k\in\mathbb{Z}} d[k]\delta(\cdot - k)\right)(t), && (3.81)\end{aligned}$$

where $*$ denotes the continuous convolution and δ is the Dirac delta. In the Fourier domain,

$$\widehat{f_{\mathrm{Her}}}(\omega) = \widehat{\phi}_1(\omega)C(\mathrm{e}^{\mathrm{j}\omega}) + \widehat{\phi}_2(\omega)D(\mathrm{e}^{\mathrm{j}\omega}) = \widehat{\boldsymbol{\phi}}^T(\omega)\begin{bmatrix} C(\mathrm{e}^{\mathrm{j}\omega}) \\ D(\mathrm{e}^{\mathrm{j}\omega}) \end{bmatrix}, \tag{3.82}$$

with $C(\mathrm{e}^{\mathrm{j}\omega})$ and $D(\mathrm{e}^{\mathrm{j}\omega})$ the discrete Fourier transforms of the sequences c and d, respectively. Since, from (3.14),

$$\widehat{\boldsymbol{\phi}}(\omega) = \widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})\widehat{\boldsymbol{\rho}}(\omega) = \widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega})\begin{bmatrix} 1 \\ \mathrm{j}\omega \end{bmatrix}\widehat{\rho}_{\mathrm{D}^4}(\omega), \tag{3.83}$$

then

$$\widehat{f_{\mathrm{Her}}}(\omega) = \begin{bmatrix} 1 \\ \mathrm{j}\omega \end{bmatrix}^T \widehat{\rho}_{\mathrm{D}^4}(\omega)\widehat{\mathbf{R}}^T(\mathrm{e}^{\mathrm{j}\omega}) \begin{bmatrix} C(\mathrm{e}^{\mathrm{j}\omega}) \\ D(\mathrm{e}^{\mathrm{j}\omega}) \end{bmatrix}. \tag{3.84}$$

Let us remind that $\mathrm{j}\omega\widehat{g}(\omega) = \widehat{g'}(\omega)$. Then,

$$\begin{aligned}
\langle \mathrm{D}^2 f_{\mathrm{Her}}, \mathrm{D}^2 g\rangle_{L_2} &= \frac{1}{2\pi}\int_{\mathbb{R}} (\mathrm{j}\omega)^2 \widehat{f_{\mathrm{Her}}}(\omega)(\mathrm{j}\omega)^2 \widehat{g}(\omega)\,\mathrm{d}\omega & (3.85)\\
&= \frac{1}{2\pi}\int_{\mathbb{R}} \begin{bmatrix} 1 \\ \mathrm{j}\omega \end{bmatrix}^T \widehat{\mathbf{R}}^T(\mathrm{e}^{\mathrm{j}\omega}) \begin{bmatrix} C(\mathrm{e}^{\mathrm{j}\omega}) \\ D(\mathrm{e}^{\mathrm{j}\omega}) \end{bmatrix} \widehat{g}(\omega)\,\mathrm{d}\omega & (3.86)\\
&= \frac{1}{2\pi}\int_{\mathbb{R}} \left\langle \widehat{\mathbf{R}}^T(\mathrm{e}^{\mathrm{j}\omega}) \begin{bmatrix} C(\mathrm{e}^{\mathrm{j}\omega}) \\ D(\mathrm{e}^{\mathrm{j}\omega}) \end{bmatrix}, \begin{bmatrix} 1 \\ \mathrm{j}\omega \end{bmatrix} \widehat{g}(\omega) \right\rangle \mathrm{d}\omega & (3.87)\\
&= \frac{1}{2\pi}\int_{\mathbb{R}} \left\langle \widehat{\mathbf{R}}(\mathrm{e}^{\mathrm{j}\omega}) \begin{bmatrix} C(\mathrm{e}^{\mathrm{j}\omega}) \\ D(\mathrm{e}^{\mathrm{j}\omega}) \end{bmatrix}, \begin{bmatrix} \widehat{g}(\omega) \\ \widehat{g'}(\omega) \end{bmatrix} \right\rangle & (3.88)\\
&= \frac{1}{2\pi}\int_0^{2\pi} A(\mathrm{e}^{\mathrm{j}\omega}) \sum_{k\in\mathbb{Z}} \widehat{g}(\omega + 2k\pi)\,\mathrm{d}\omega \\
&\quad + \frac{1}{2\pi}\int_0^{2\pi} B(\mathrm{e}^{\mathrm{j}\omega}) \sum_{k\in\mathbb{Z}} \widehat{g'}(\omega + 2k\pi)\,\mathrm{d}\omega & (3.89)\\
&\leq \|A(\mathrm{e}^{\mathrm{j}\omega})\|_{L_2([0,2\pi])} \left\| \sum_{k\in\mathbb{Z}} \widehat{g}(\omega + 2k\pi) \right\|_{L_2([0,2\pi])} \\
&\quad \cdot \|B(\mathrm{e}^{\mathrm{j}\omega})\|_{L_2([0,2\pi])} \left\| \sum_{k\in\mathbb{Z}} \widehat{g'}(\omega + 2k\pi) \right\|_{L_2([0,2\pi])}, & (3.90)
\end{aligned}$$

where $A = \widehat{R}_{11}C + \widehat{R}_{12}D$ and $B = \widehat{R}_{21}C + \widehat{R}_{22}D$. In the development, (3.85) is obtained from Parseval's relation. Then, (3.86) comes from the definition of f_{Her} given in (3.84) and from the fact that $\widehat{\rho}_{\mathrm{D}^4}(\omega)(\mathrm{j}\omega)^4 = 1$. Finally, Cauchy-Schwarz inequality yields (3.90). The terms $\|A\|_{L_2([0,2\pi])}$ and $\|B\|_{L_2([0,2\pi])}$ are bounded since the four entries of $\widehat{\mathbf{R}}$ given by (3.14) have finite L_2 norms, and the $C(\mathrm{e}^{\mathrm{j}\omega})$, $D(\mathrm{e}^{\mathrm{j}\omega})$ are discrete-time Fourier transforms of sequences $c, d \in \ell_2(\mathbb{Z})$. Finally, since $g = f - f_{\mathrm{Her}}$ and $f(k) = f_{\mathrm{Her}}(k)$, $f'(k) = f'_{\mathrm{Her}}(k)$ for all $k \in \mathbb{Z}$, one obtains, by

Poisson formula, that

$$\sum_{k\in\mathbb{Z}} \widehat{g}(\omega+2k\pi) = \sum_{k\in\mathbb{Z}} g(k)\mathrm{e}^{-\mathrm{j}k\omega} = 0, \tag{3.91}$$

$$\sum_{k\in\mathbb{Z}} \widehat{g'}(\omega+2k\pi) = \sum_{k\in\mathbb{Z}} g'(k)\mathrm{e}^{-\mathrm{j}k\omega} = 0. \tag{3.92}$$

This concludes the proof. □

3.3.2 Statistical Interpretation: MMSE Estimation

We now consider a statistical framework where the signal s is modeled as a process in the continuous domain described by the linear stochastic differential equation $\mathrm{L}s = w$, with L a differential operator referred to as the whitening filter, and w a Lévy white noise [85]. This generic equation allows to easily describe an extended family of processes, including Brownian motion and Lévy flights. We focus on the statistical interpretation of spline optimality. In the stationary case, the generalized spline interpolator is demonstrated to be the linear minimum mean-square error (LMMSE) estimator of the stationary continuous-time stochastic process $s(t)$, $t \in \mathbb{R}$ whose whitening filter L is spline-admissible, given its discrete sequence of samples $\{c[k]\}_{k\in\mathbb{Z}}$ with $c[k] = s(k)$ [100]. Spline interpolation is therefore optimal for the estimation of a wide family of stationary random signals. Similar results have been demonstrated in [101, 102]: the optimal interpolator for first-order Lévy processes is always the piecewise linear B-spline. First-order Lévy processes, defined as s such that $\mathrm{D}s = w$, are non-stationary but (wide-sense) self-similar. We aim at extending these results to the Hermite spline interpolation framework, relying on the two sequences of samples $\{c[k], d[k]\}_{k\in\mathbb{Z}}$, where $c[k] = s(k)$ and $d[k] = s'(k)$.

Second-Order Lévy Processes

We consider continuous-time random processes s that are solutions of the stochastic differential equation

$$\mathrm{D}^2 s = w, \tag{3.93}$$

where D is the differential operator and w a Lévy white noise with finite variance and zero mean, and with boundary conditions $s(0) = s'(0) = 0$. The process s

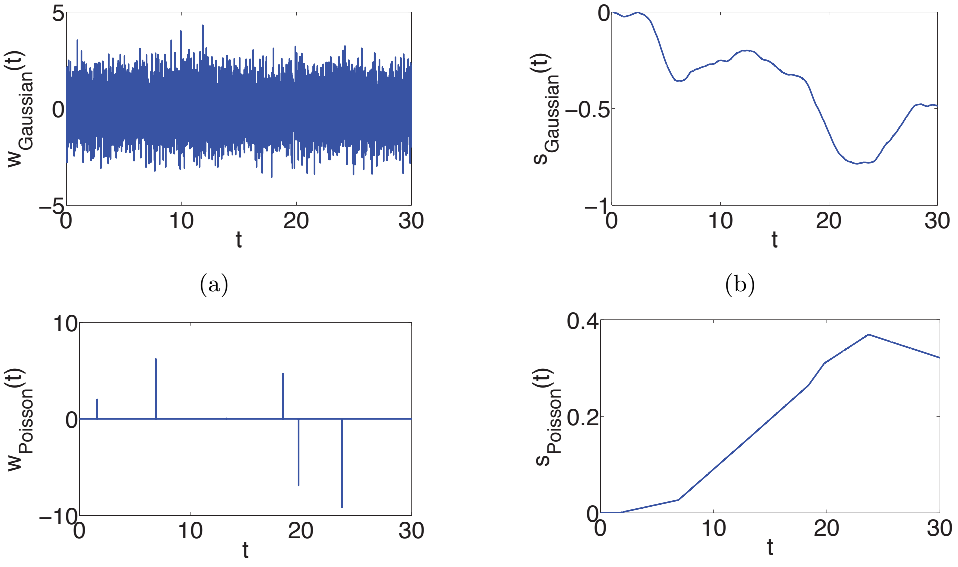
is called a *second-order Lévy process* [85]. As the integration of a Lévy process, second-order Lévy processes are smoother. Such processes include the integrated version of the Brownian motion, which corresponds to Gaussian white noise excitations (Figures 3.3a and 3.3b). They can as well be driven by compound-Poisson white noise (Figure 3.3c), which yields piecewise linear processes, as illustrated in Figure 3.3d. Second-order Lévy processes s are continuous and their derivatives s' are first-order Lévy processes [103]. The samples of s and s' at the integers are therefore well-defined. The Poisson process is for instance not differentiable at the jumps locations, but the intersection of the set of transitions and sampling locations (*i.e.*, the integers) is empty with probability 1. A throughout mathematical formulation of these objects can be found in [85, 104].

Let $c_{ss}(t,\tau) = \mathbb{E}\{s(t)s(\tau)\}$ be the autocorrelation function of s. The second-order Lévy process s is *wide-sense self-similar with scaling order* $H = \frac{3}{2}$, meaning that s and $a^{\frac{3}{2}} s\left(\frac{\cdot}{a}\right)$ have the same second-order statistics for all $a > 0$. Equivalently, the autocorrelation function satisfies $c_{ss}(t,\tau) = a^3 c_{ss}\left(\frac{t}{a}, \frac{\tau}{a}\right)$. In the Gaussian case, s is even *strongly self-similar*, which means that s and $a^{\frac{3}{2}} s\left(\frac{\cdot}{a}\right)$ have the same law. Moreover, s is non-stationary but has second-order stationary increments [85]. The explicit expression of the autocorrelation function c_{ss} is given in [105] as

$$c_{ss}(t,\tau) = \frac{\sigma^2}{4}\left(|t-\tau|^3 - |t|^3 - |\tau|^3 + 3t\tau(|t| + |\tau|)\right) \tag{3.94}$$

and is normalized such that $\mathrm{Var}\{s(t)\} = \sigma^2 |t|^3$.

Our goal is to study the estimation of s at a fixed time point t_0 given the sequences $\{c[k], d[k]\}_{k\in\mathbb{Z}}$. Since the process s is of slow growth [106], its samples fit the requirements enunciated in Section 3.1. Moreover, for a fixed $\tau = t_0$, the derivative of $c_{ss}(t, t_0)$ with respect to t can easily be computed and is always well-defined. Similarly, the sequences $\{c_{ss}(k,t_0)\}_{k\in\mathbb{Z}}$ and $\{c'_{ss}(k,t_0)\}_{k\in\mathbb{Z}}$ are well-defined for all $k \in \mathbb{Z}$ and are of slow growth. Hence, the function $t \mapsto c_{ss}(t, t_0)$ and its first derivative can be expanded in a Hermite basis.



(a) (b)

(c) (d)

Figure 3.3: Second-order Lévy processes for different types of noise. (a) Randomly generated Gaussian noise and (b) corresponding second-order Gaussian process. (c) Randomly generated Poisson noise with normally distributed jumps and (d) corresponding second-order Poisson process. We use the conventional way of representing white noises although they are not technically defined pointwise.

Statistical Optimality of Hermite Spline

For a fixed t_0, we determine a linear estimator of $s(t_0)$ given the random sequences $\{c[k], d[k]\}_{k\in\mathbb{Z}}$. The estimator is a random variable of the form

$$\tilde{s}(t_0) = \sum_{k\in\mathbb{Z}} \left(a_{t_0}[k]c[k] + b_{t_0}[k]d[k]\right), \tag{3.95}$$

where $a_{t_0}[k]$ and $b_{t_0}[k]$ are two infinite deterministic sequences of regression coefficients. The linear minimum mean-square error (LMMSE) estimator $\tilde{s}_{\mathrm{LMMSE}}(t_0)$ corresponds to the estimator $\tilde{s}(t_0)$ that minimizes the mean-square error $\mathbb{E}\{|s(t_0) - \tilde{s}(t_0)|^2\}$.

The random sequences $c[k] = s(k)$ and $d[k] = s'(k)$, for all $k \in \mathbb{Z}$, are in $S'(\mathbb{Z})$ almost surely [85]. We make sure that the summation in (3.95) is well-defined by restricting ourselves to sequences $a_{t_0}[k]$, $b_{t_0}[k] \in S(\mathbb{Z})$, the space of sequences that decay faster than any polynomial.

Theorem 2. *Let s be such that $\mathbb{E}\{s(t)\} = 0$ and $\mathbb{E}\{s(t)s(\tau)\}$ is equal to* (3.94)*. Then, the linear minimum mean-square error (LMMSE) estimators of $s(t)$ and $s'(t)$ at $t = t_0$ given the samples $\{c[k] = s(k), d[k] = s'(k)\}_{k\in\mathbb{Z}}$ are*

$$\tilde{s}_{\mathrm{LMMSE}}(t_0) = s_{\mathrm{Her}}(t_0) \tag{3.96}$$

$$\widetilde{s'}_{\mathrm{LMMSE}}(t_0) = s'_{\mathrm{Her}}(t_0), \tag{3.97}$$

where

$$s_{\mathrm{Her}}(t) = \sum_{k\in\mathbb{Z}} \left(c[k]\phi_1(t_0 - k) + d[k]\phi_2(t_0 - k)\right)$$

is the Hermite interpolation of $\{c[k], d[k]\}_{k\in\mathbb{Z}}$.

Proof. Note that, since boundary conditions impose that $s(0) = s'(0) = 0$, $c[0] = d[0] = 0$, we exclude the index $k = 0$ through the proof for the sake of convenience.

A linear estimator is built by projecting the element to be estimated onto the space spanned by the set of elements used for the estimation. The error is minimized if it is orthogonal to the subspace spanned by the elements used for the estimation or, in other term, if their inner products are equal to zero. This is referred to as the orthogonality principle [107] and yields for every $n \in \mathbb{Z}$

$$\mathbb{E}\{c[n](s(t_0) - \tilde{s}_{\mathrm{LMMSE}}(t_0))\} = 0, \tag{3.98}$$

$$\mathbb{E}\{d[n](s(t_0) - \tilde{s}_{\mathrm{LMMSE}}(t_0))\} = 0. \tag{3.99}$$

By definition of the autocorrelation function, we have that

$$\begin{aligned}\mathbb{E}\{s(t)s(\tau)\} &= c_{ss}(t,\tau) = c_{ss}(\tau,t), \quad (3.100)\\ \mathbb{E}\{s'(t)s(\tau)\} &= \partial_1 c_{ss}(t,\tau),\\ \mathbb{E}\{s(t)s'(\tau)\} &= \partial_2 c_{ss}(t,\tau) = \partial_1 c_{ss}(\tau,t). \quad (3.101)\end{aligned}$$

Rearranging (3.98) and plugging in (3.95), we obtain

$$\mathbb{E}\{c[n]s(t_0)\} = \mathbb{E}\{c[n]\tilde{s}_{\mathrm{LMMSE}}(t_0)\} = \sum_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]\mathbb{E}\{c[n]c[k]\} + b_{t_0}[k]\mathbb{E}\{c[n]d[k]\}) \tag{3.102}$$

which, from (3.100) and (3.101), can be rewritten as

$$c_{ss}(n,t_0) = \sum_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]c_{ss}(n,k) + b_{t_0}[k]\partial_2 c_{ss}(n,k))\,. \tag{3.103}$$

Doing similar operations with (3.99), we obtain the system

$$\begin{cases} c_{ss}(n,t_0) &= \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]c_{ss}(n,k) + b_{t_0}[k]\partial_2 c_{ss}(n,k)) \\ \partial_1 c_{ss}(n,t_0) &= \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]\partial_1 c_{ss}(n,k) + b_{t_0}[k]\partial_2\partial_1 c_{ss}(n,k))\,. \end{cases} \tag{3.104}$$

We now observe that

$$c_{ss}(n,t_0) = \frac{\sigma^2}{4}\left(|n-t_0|^3 - |n|^3 - |t_0|^3 + 3nt_0(|n| + |t_0|)\right) \tag{3.105}$$

and

$$\partial_1 c_{ss}(n,t_0) = \frac{3\sigma^2}{4}\left((n-t_0)|n-t_0| + n(2t_0\mathrm{sgn}(n) - |n|) + t_0|t_0|\right). \tag{3.106}$$

Therefore, c_{ss} is piecewise cubic with C^2 transitions, and $\partial_1 c_{ss}$ is piecewise cubic with C^1 transitions. They both belong to the space spanned by the cubic Hermite splines ϕ_1 and ϕ_2 and can be expanded following (3.1) as

$$c_{ss}(n,\cdot) \quad = \sum_{k\in\mathbb{Z}\setminus\{0\}} (c_{ss}(n,k)\phi_1(\cdot - k) + \partial_2 c_{ss}(n,k)\phi_2(\cdot - k)), \tag{3.107}$$

$$\partial_1 c_{ss}(n,\cdot) = \sum_{k\in\mathbb{Z}\setminus\{0\}} (\partial_1 c_{ss}(n,k)\phi_1(\cdot-k) + \partial_2\partial_1 c_{ss}(n,k)\phi_2(\cdot-k)). \tag{3.108}$$

The system to solve thus becomes

$$\begin{cases} \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (c_{ss}(n,k)\phi_1(t_0-k) + \partial_2 c_{ss}(n,k)\phi_2(t_0-k)) \\ \qquad = \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]c_{ss}(n,k) + b_{t_0}[k]\partial_2 c_{ss}(n,k)) \\ \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (\partial_1 c_{ss}(n,k)\phi_1(t_0-k) + \partial_2\partial_1 c_{ss}(n,k)\phi_2(t_0-k)) \\ \qquad = \sum\limits_{k\in\mathbb{Z}\setminus\{0\}} (a_{t_0}[k]\partial_1 c_{ss}(n,k) + b_{t_0}[k]\partial_2\partial_1 c_{ss}(n,k))\,. \end{cases} \tag{3.109}$$

Let the sequences u and v be such that $u[k] = \phi_1(t_0-k) - a_{t_0}[k]$ and $v[k] = \phi_2(t_0-k) - b_{t_0}[k]$, $k\in\mathbb{Z}\setminus\{0\}$. The system (3.109) can be reformulated in blocs matrix notation as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3.110}$$

where $[\mathbf{A}]_{n,k} = c_{ss}(n,k)$, $[\mathbf{B}]_{n,k} = \partial_2 c_{ss}(n,k)$, $[\mathbf{B}^T]_{n,k} = \partial_1 c_{ss}(n,k)$ (see (3.101)) and $[\mathbf{C}]_{n,k} = \partial_1\partial_2 c_{ss}(n,k)$, with a slight abuse of notation from the fact that these sequences and matrices are bi-infinite. To finish the proof, we observe that

$$\begin{bmatrix} u \\ v \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \mathrm{Var}\left\{ \sum_{k\in\mathbb{Z}\setminus\{0\}} (u[k]s(k) + v[k]s'(k)) \right\} = 0, \tag{3.111}$$

which implies that $\sum_{k\in\mathbb{Z}\setminus\{0\}} (u[k]s(k) + v[k]s'(k)) = 0$ almost surely. The only solution is therefore $u[k] = v[k] = 0$ for all $k\in\mathbb{Z}\setminus\{0\}$. Indeed, if one of the $u[k]$ or $v[k]$ is nonzero, say for instance $u[k_0]$, it implies that $s(k_0)$ can be deterministically evaluated from the values of all the $s(k)$ and $s'(k)$ except $s(k_0)$, which is absurd. Therefore,

$$\begin{aligned} a_{t_0}[k] &= \phi_1(t_0-k), \\ b_{t_0}[k] &= \phi_2(t_0-k), \end{aligned}$$

and the LMMSE is finally given by

$$\tilde{s}_{\mathrm{LMMSE}}(t_0) = s_{\mathrm{Her}}(t_0) = \sum_{k\in\mathbb{Z}} \left(s(k)\phi_1(t_0-k) + s'(k)\phi_2(t_0-k)\right). \tag{3.112}$$

The proof for the problem of estimating $s'(t_0)$ from $\{s(k), s'(k)\}_{k\in\mathbb{Z}}$ is obtained with a similar development starting from

$$\mathbb{E}\{s(n)(s'(t_0) - \widetilde{s'}_{\mathrm{LMMSE}}(t_0))\} = 0, \tag{3.113}$$

$$\mathbb{E}\{s'(n)(s'(t_0) - \widetilde{s'}_{\mathrm{LMMSE}}(t_0))\} = 0, \tag{3.114}$$

and the LMMSE is given by

$$\widetilde{s'}_{\mathrm{LMMSE}}(t_0) = s'_{\mathrm{Her}}(t_0) = \sum_{k\in\mathbb{Z}} \left(s(k)\phi'_1(t_0-k) + s'(k)\phi'_2(t_0-k)\right). \tag{3.115}$$

□

The LMMSE is the *linear* estimator that provides the minimum mean square error. In the Gaussian case, it can be shown from Bayes theorem that the LMMSE is actually the MMSE [107], which is the best *overall* estimator. This leads to Proposition 4.

Proposition 4. *If w is Gaussian, then the Hermite interpolation of $\{s(k), s'(k)\}_{k\in\mathbb{Z}}$ is also the MMSE of w.*

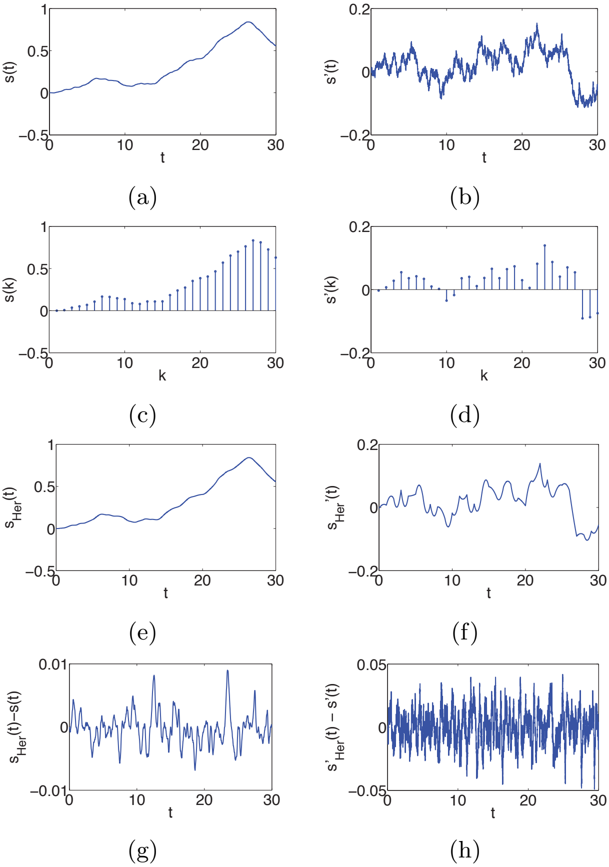
Second-order Lévy processes s such that $\mathrm{D}^2 s = w$ are an example of processes that satisfy the assumptions of Theorem 2. It is worth noting that, in the Poisson case, our result implies that the best reconstruction of a piecewise linear (second-order Poisson) process is a piecewise cubic (Hermite) function. This result, which appears counter intuitive at first, can be explained as follows. On one hand, the second-order Poisson process is generated by the double integration of an impulsive noise, with non-uniformly distributed impulse locations. On the other hand, the Hermite interpolation scheme is cardinal with knots placed at the integers. In this setting, one cannot simply linearly interpolate between the "knots" of the second-order Poisson process, as they are not located at the integers.

Theorem 2 and Proposition 4 provide solid statistical ground for using Hermite spline interpolation when dealing with second-order Lévy processes. The novelty of these results is twofold. First, they rely on a sampling scheme involving both $s(k)$ and $s'(k)$. Then, in addition to the optimal estimator of $s(t_0)$, they simultaneously give the optimal estimator of the derivative $s'(t_0)$, which happens to be the derivative of the Hermite spline interpolator. Our results are strongly linked to [100, 101] as they also highlight the optimality of splines for the estimation of random processes from their samples. Our contribution however differs in two ways: first from the fact that the samples $s'(k)$ are provided in addition to the $s(k)$, and then from the class of studied processes, which are non-stationary (unlike in [100], where the stationary case is investigated) and second-order (unlike in [101], where first-order Lévy processes are considered).

Illustrative Examples

We illustrate the Hermite spline reconstruction from samples of a second-order Lévy process in the Gaussian case. A realization of a continuous-time random process s satisfying $\mathrm{D}^2 s = w$ and its continuous-time derivative s' are represented in Figures 3.4a and 3.4b, respectively. As, formally, $s = \mathrm{D}^{-2} w$ (with suitable boundary conditions), the process s was generated by performing two rounds of integration on a Gaussian white noise while the derivative s' was obtained by integrating the noise only once. The process and its derivative were sampled at the integers, yielding the two discrete sequences $s(k)$ and $s'(k)$ represented in Figures 3.4c and 3.4d. Finally, the Hermite interpolation scheme (3.1) was applied to reconstruct the process s from its samples $\{s(k), s'(k)\}_{k\in\mathbb{Z}}$, yielding the continuous-time process s_{Her} displayed in Figure 3.4e. Similarly, the scheme was applied to reconstruct the continuously defined derivative s' from the samples $\{s(k), s'(k)\}_{k\in\mathbb{Z}}$, yielding s'_{Her} shown in Figure 3.4f. The absolute reconstruction error on the process, $|s_{\mathrm{Her}}(t) - s(t)|$, is smaller than 0.01 (Figure 3.4g), and that on its derivative, $|s'_{\mathrm{Her}}(t) - s'(t)|$, is smaller than 0.05, as seen in Figure 3.4h.

These optimality results hold for uniform samples. However, we expect similar findings in a non uniform sampling scheme, as it is already known in the case of reconstruction with mere function samples [102]. An extension of the problem would be to investigate the optimal estimation algorithm for continuous processes given collections of noise-corrupted measurements. There, we again expect to obtain results analogous to the known ones in the ordinary sampling case presented in [100].



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.4: Hermite spline reconstruction of a sampled second-order Gaussian process. (a) Realization of a second-order Gaussian process s and (b) its continuous derivative s'. (c) Discrete sequence of samples $s(k)$ drawn from s and (d) sequence of samples $s'(k)$ drawn from s', the first derivative of s. (e) Reconstruction s_{Her} of the original continuous-time process s using Hermite spline interpolation. (f) Reconstruction s'_{Her} of the original continuous-time derivative s' using Hermite spline interpolation. Reconstruction errors (g) $(s_{\mathrm{Her}} - s)$ and (f) $(s'_{\mathrm{Her}} - s')$.

3.4 Towards Hermite L-Splines

In the following, we discuss a generalization of cubic Hermite splines. We proceed in two steps. First, we consider Hermite constructions including sequences of samples from higher order derivatives. As a consequence, more spline generators are required. These higher order polynomial Hermite families are already covered by the theory of I. J. Schoenberg [74, 75], although the explicit expressions of the resulting bases are not necessarily provided. The second direction we explore is the development of novel Hermite spline constructions by considering other differential operators. The underlying idea is that every spline construction is associated to a particular differential operator, as introduced in [78] and [79, Chapter 10]. By considering different operators, alternative spline families can be constructed. This unifying theory, referred to as L-splines, is further developed in [80].

3.4.1 Higher Order Polynomial Hermite Splines

Higher-order families of Hermite splines are obtained by increasing the number of considered derivatives and, as a consequence, of generators. The immediate next construction after cubic Hermite splines yields quintic polynomial splines. Three generators ϕ_1, ϕ_2, and ϕ_3 are involved, and their supports are all restricted to the $[-1, 1]$ interval. The functions ϕ_1, ϕ_2 and ϕ_3 and their first and second derivatives ϕ_1', ϕ_2', ϕ_3', ϕ_1'', ϕ_2'', and ϕ_3'' satisfy the joint interpolation conditions

$$\begin{aligned}&\phi_1(k) = \delta[k], \quad \phi_2'(k) = \delta[k], \quad \phi_3''(k) = \delta[k],\\ &\phi_1'(k) = 0, \quad \phi_1''(k) = 0, \quad \phi_2(k) = 0, \quad \phi_2''(k) = 0, \quad \phi_3(k) = 0, \quad \phi_3'(k) = 0,\end{aligned} \tag{3.116}$$

for all $k \in \mathbb{Z}$. Their expressions are given by

$$\phi_1(t) = \begin{cases} 1 - 10|t|^3 + 15t^4 - 6|t|^5 & \text{for } 0 \le |t| \le 1 \\ 0 & \text{for } |t| > 1, \end{cases} \tag{3.117}$$

$$\phi_2(t) = \begin{cases} t - 6t^3 + 8t|t|^3 - 3t^5 & \text{for } 0 \le |t| \le 1 \\ 0 & \text{for } |t| > 1. \end{cases} \tag{3.118}$$

$$\phi_3(t) = \begin{cases} \frac{1}{2}\left(t^2 - 3|t|^3 + 3t^4 - |t|^5\right) & \text{for } 0 \le |t| \le 1 \\ 0 & \text{for } |t| > 1. \end{cases} \tag{3.119}$$

The functions ϕ_1, ϕ_2 and ϕ_3 along with their first and second derivatives are represented in Figure 3.5. They satisfy the second-order interpolation properties (3.116). Functions interpolated with this scheme are splines of degree 5 with C^2 regularity at the knots. In comparison, quintic B-splines are also of degree 5 but are of C^4 regularity at the knots.

The construction of higher-order polynomial Hermite spline families follows a path analogous to that of the design of B-splines of increasing order. These two strategies pursue similar goals under different sets of constraints. We propose the following unifying vision. The starting point is linear interpolation, or linear B-splines, which are the simplest way of interpolating a set of points. From there, several generalizations can be considered.

- One of them is to increase the smoothness of the scheme, related to the approximation power, while keeping a single basis function. This yields to the construction of B-splines, and comes at the cost of losing interpolation properties and having larger and larger supports for increasing smoothness.

- A second one is to increase the smoothness of the scheme but require that the interpolation conditions remain. A powerful way to increase smoothness is then to use several basis functions jointly. Different bases are then used for different samples types, *e.g.* one basis for even and one basis for odd sample indices. This corresponds to multigenerator constructions as studied in [91, 94].

- Another variation of the previous solution is to use the same basis for all samples, but to have different bases handling different orders of differentiation of the function. This corresponds to the Hermite construction.

Interestingly, it seems to be a general case that saving the interpolation conditions imposes for basis functions to remain of support 2. This was observed in all studied cases, although not demonstrated formally. The linear spline can thus be considered as the common starting point of both the classical and Hermite splines constructions.

3.4.2 Exponential Hermite Splines

A framework to build exponential splines as generalization of B-splines was introduced in [108]. There, a continuous-time function $s(t)$, $t \in \mathbb{R}$ is a called a cardinal

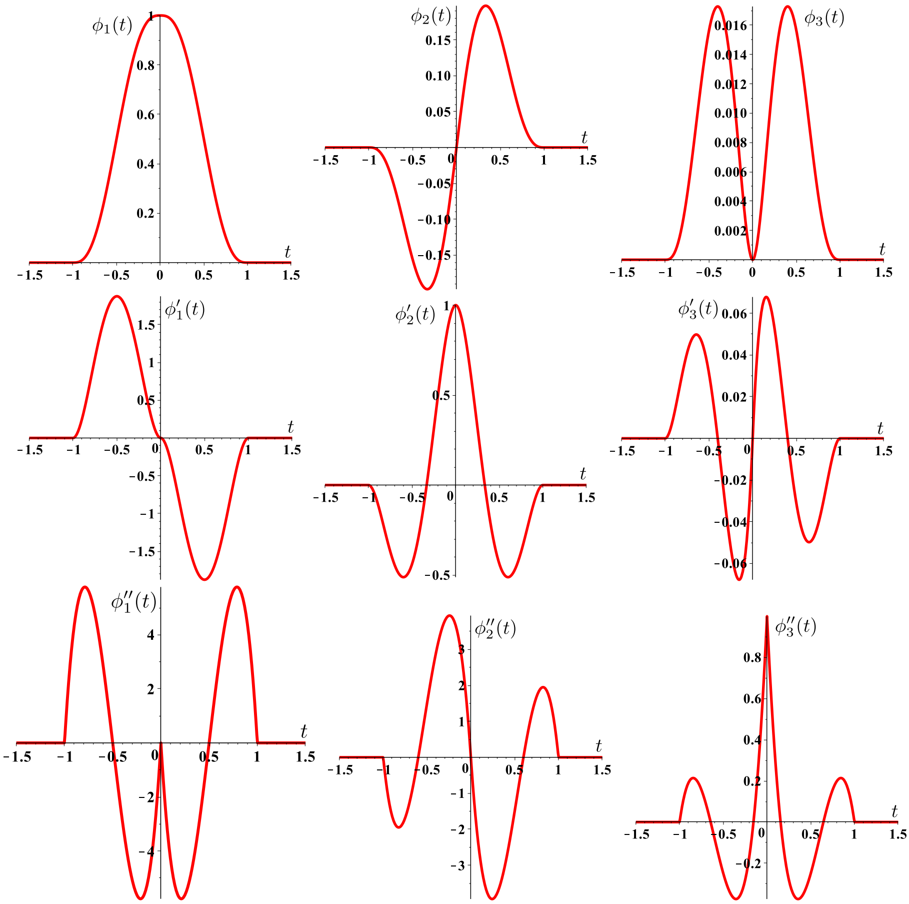


Figure 3.5: Second-order (quintic) polynomial Hermite spline basis functions and their first and second derivatives.

L-spline if and only if

$$\mathrm{L}s(t) = \sum_{k\in\mathbb{Z}} a[k]\delta(t-k), \tag{3.120}$$

with $a[k] \in S'(\mathbb{Z})$. In particular, B-splines are obtained when the considered differential operator corresponds to an nth order derivative, *i.e.*, $\mathrm{L} = \mathrm{D}^n$. The first-order differential operator, D, generates $\beta^0(t)$, the B-spline of degree 0, yielding piecewise-constant functions. Exponential splines are obtained when considering operators of the type $\mathrm{L} = \mathrm{D} - \alpha\mathrm{I}$, where $\alpha \in \mathbb{C}$ is a constant coefficient.

The Hermite construction allows for an analogous generalization. Although providing a complete theory in the spirit of [80] is not the goal of this thesis, we investigate an example of other Hermite spline family. We only consider differential operators of the form $\mathrm{L} = \alpha_0\mathrm{I} + ... + \alpha_N\mathrm{D}^N = P(\mathrm{D})$, where P denotes a polynomial function. The degree $N \geq 1$ of P is by extension called the degree of the operator. The particular case of cubic Hermite splines we considered up to now corresponds to $\mathrm{L} = \mathrm{D}^4$. Increasing the Hermite order amounts to consider operators of the form D^n for increasing values of n. The second-order Hermite splines introduced in 3.4.1 for instance correspond to $\mathrm{L} = \mathrm{D}^6$.

In this spirit, we construct the first-order exponential Hermite spline functions associated to the operator $\mathrm{D}-\alpha\mathrm{I}$, with $\alpha \in \mathbb{C}$ a constant coefficient. They generalize the first-order polynomial Hermite splines. Exponential splines are relevant to model circles and ellipses when α is chosen to be a purely imaginary number. We investigated this additional family in [109][6]. The considered differential operator is $\mathrm{L} = \mathrm{D}^4 + \frac{4\pi^2}{M^2}\mathrm{D}^2 = \mathrm{D}^2(\mathrm{D} - \frac{\mathrm{j}2\pi}{M}\mathrm{I})(\mathrm{D} + \frac{\mathrm{j}2\pi}{M}\mathrm{I})$. The corresponding exponential Hermite spline generators are specified as

$$\phi_{1,M}(t) = \begin{cases} g_{1,M}(t) & t \geq 0 \\ g_{1,M}(-t) & t < 0 \end{cases} \tag{3.121}$$

$$\phi_{2,M}(t) = \begin{cases} g_{2,M}(t) & t \geq 0 \\ -g_{2,M}(-t) & t < 0 \end{cases} \tag{3.122}$$

[6] This work has been carried out in collaboration with Profs. C. Conti, Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Florence, Italy, and L. Romani, Department of Mathematics and Applications, University of Milano-Bicocca, Minalo, Italy.

built using the functions

$$g_{1,M}(t) = \begin{cases} a_1(M) + b_1(M)t + c_1(M)\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}t} + d_1(M)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{M}t} & 0 \leq t \leq 1, \\ 0 & \text{elsewhere,} \end{cases} \tag{3.123}$$

$$g_{2,M}(t) = \begin{cases} a_2(M) + b_2(M)t + c_2(M)\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}t} + d_2(M)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{M}t} & 0 \leq t \leq 1, \\ 0 & \text{elsewhere,} \end{cases} \tag{3.124}$$

where M is a positive integer. The coefficients of the generators are expressed as

$$\begin{aligned} a_1(M) &= \tfrac{\mathrm{j}\frac{2\pi}{M}+1+\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}(\mathrm{j}\frac{2\pi}{M}-1)}{q(M)}, & b_1(M) &= -\tfrac{\mathrm{j}\frac{2\pi}{M}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}+1)}{q(M)}, \\ c_1(M) &= \tfrac{1}{q(M)}, & d_1(M) &= -\tfrac{\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}}{q(M)}, \\ a_2(M) &= \tfrac{p(M)}{\mathrm{j}\frac{2\pi}{M}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}-1)q(M)}, & b_2(M) &= -\tfrac{\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}-1}{q(M)}, \\ c_2(M) &= \tfrac{\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}-\mathrm{j}\frac{2\pi}{M}-1}{\mathrm{j}\frac{2\pi}{M}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}-1)q(M)}, & d_2(M) &= -\tfrac{\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}(\mathrm{j}\frac{2\pi}{M}-1)+1)}{\mathrm{j}\frac{2\pi}{M}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}-1)q(M)}, \end{aligned}$$

with

$$\begin{aligned} p(M) &= \mathrm{j}\frac{2\pi}{M} + 1 + \mathrm{e}^{\mathrm{j}\frac{4\pi}{M}}\left(\mathrm{j}\frac{2\pi}{M} - 1\right), \\ q(M) &= \mathrm{j}\frac{2\pi}{M} + 2 + \mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}\left(\mathrm{j}\frac{2\pi}{M} - 2\right). \end{aligned}$$

We display $\phi_{1,M}$ and $\phi_{2,M}$ along with their derivatives in Figure 3.6. Similarly as classical polynomial Hermite splines, the support of these two functions is limited to the $[-1,1]$ interval. The generating functions $\phi_{1,M}$ and $\phi_{2,M}$ and their derivatives $\phi'_{1,M}$ and $\phi'_{2,M}$ also satisfy the joint interpolation conditions

$$\phi_{1,M}(k) = \delta[k], \quad \phi_{2,M}(k) = 0, \quad \phi'_{1,M}(k) = 0, \quad \phi'_{2,M}(n) = \delta[k] \tag{3.125}$$

for all $k \in \mathbb{Z}$. The novelty is that these basis functions reproduce the function space $\{1, t, \mathrm{e}^{\mathrm{j}\frac{2\pi t}{M}}, \mathrm{e}^{-\mathrm{j}\frac{2\pi t}{M}}\}$. Hence, they generate ellipsoid and circular curves. Exponential Hermite splines can for instance represent perfect circles with M control points, whereas an infinity would be required with polynomial Hermite splines. Note that, as M grows to infinity, $\phi_{1,M}$ and $\phi_{2,M}$, the exponential Hermite splines, converge to ϕ_1 and ϕ_2, the polynomial Hermite splines. Additional mathematical properties of Hermite exponential splines, including their associated vector subdivision scheme, are investigated and discussed in [110].

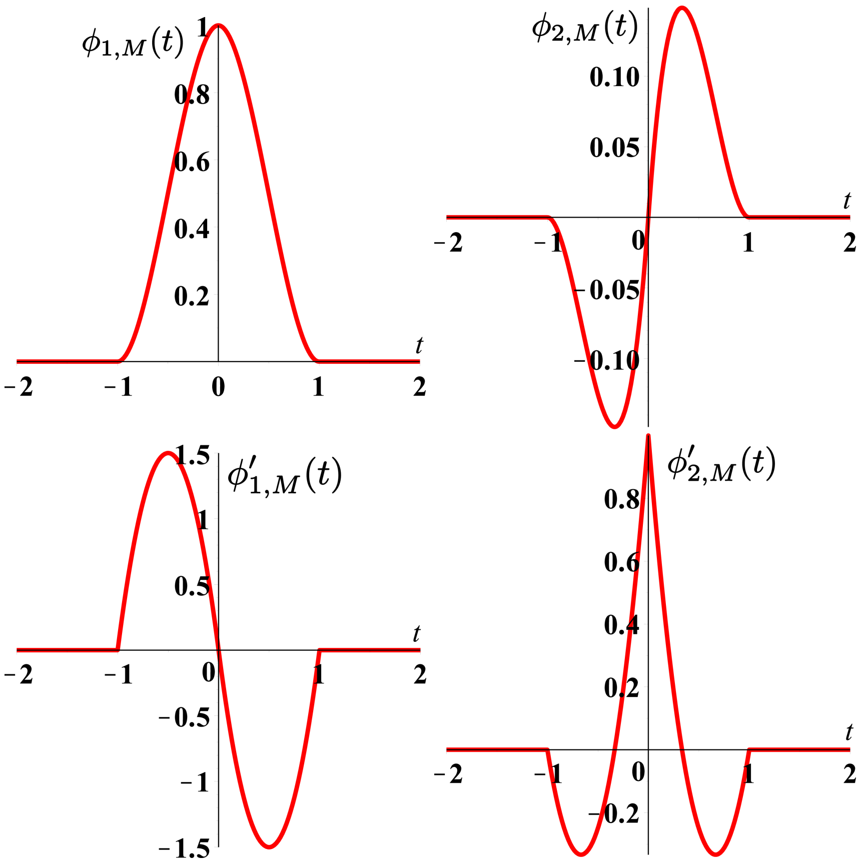


Figure 3.6: Ellipse-reproducing Hermite exponential B-spline basis functions and their first derivatives, represented for $M = 20$.

Chapter 4

Designing Feature Detectors

In this chapter, we focus on our second ingredient, namely detectors providing information about local features in the image. We study steerable filters as a possible way to design such tools. We first recall the steerable filter formalism focusing on the particular family of polar separable functions. We highlight two important elements in their construction, namely an isotropic, radial part and a set of circular harmonics determining the angular part. We then separately investigate how each of these two elements can be tuned and how it influences the properties of the resulting detector. Finally, we discuss a learning-based scheme that is currently under development. It allows creating custom steerable filters to detect features of interest from a template. The work presented in this chapter results from collaborations with Zs. Püspöki for the study of circular harmonics and angular profiles [111] and P. Pad for the investigation of radial profiles [112, 113]. The learning framework discussed at the end is ongoing research from A. Depeursinge and J. Fageot[1]. We included it for two main reasons. Firstly, it lies in the continuity of our work on the design of radial and angular profiles, and, secondly, it is being developed so as to be used in the landmark snake framework presented in Chapter 5.

Through the chapter, we denote the two-dimensional Fourier transform of a generalized function $f \in \mathcal{S}'(\mathbb{R}^2)$ as $\mathcal{F}\{f\} = \hat{f}$, which is computed, when f is in

[1] All collaborators mentioned here are with the Biomedical Imaging Group, EPFL, Lausanne, Switzerland.

$L_1(\mathbb{R}^2)$, as

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^2} f(\mathbf{x}) \mathrm{e}^{-\mathrm{j}\langle \mathbf{x}, \boldsymbol{\omega} \rangle} \mathrm{d}\mathbf{x}, \tag{4.1}$$

for $\mathbf{x} = (x, y)$ and $\boldsymbol{\omega} = (\omega_x, \omega_y)$. We indicate the standard vector scalar product and associated norm as $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. Inner products and norms on function spaces are denoted in a similar way, with the function space indicated as a subscript. For instance, the classical L_2 inner product and norm are designated as $\langle \cdot, \cdot \rangle_{L_2}$ and $\| \cdot \|_{L_2}$, respectively. We put the arguments of operators in curly brackets and the arguments of functionals in parentheses. When there is no ambiguity, we may choose to do away with brackets and parentheses, a typical case being the gradient ∇.

4.1 A Reminder on Steerable Filters

Feature detection refers to the problem of identifying particular patterns or elements in images. When performing feature detection, one is generally interested in identifying the feature of interest regardless of its orientation. This can be performed through rotated template matching. For a 2-dimensional image f and feature template h, the detection problem is formulated as

$$\theta_0^*(\mathbf{x}) = \arg\max_{\theta_0} \left(f * h(-\mathbf{R}_{\theta_0} \cdot)\right)(\mathbf{x}), \tag{4.2}$$

$$\mu(\mathbf{x}) = \big(f * h(-\mathbf{R}_{\theta_0^*(\mathbf{x})} \cdot)\big)(\mathbf{x}), \tag{4.3}$$

where $\mathbf{R}_\theta$ denotes a rotation matrix and $*$ is the convolution operation. The magnitude and orientation of the feature at position $\mathbf{x} = (x, y)$ are given by μ and θ_0^*, respectively. In its original formulation, the template matching approach is computationally heavy as it requires the discretization of the angle θ_0 and one filtering operation for each rotated template.

The concept of steerable filters has been introduced by Freeman and Adelson [114] as a mean to perform rotated template matching at a lighter computational cost. In their formulation, the feature template, or *detector*, is chosen such that any rotated version of it can be expressed with linear combinations of a finite number of basis filters. Formally, a steerable detector h can be expressed as

$$h(\mathbf{R}_{\theta_0}\mathbf{x}) = \sum_{n=1}^{N} c_n(\theta_0)h_n(\mathbf{x}), \tag{4.4}$$

where $\mathbf{R}_{\theta_0}$ is a rotation matrix, $h_1, ..., h_N$ are the basis filters, and $c_1(\theta_0), ..., c_N(\theta_0)$ are some coefficients. Importantly, the steerability property can be equivalently shown in the space and frequency domains. Remembering that applying a rotation in the spatial domain also corresponds to a rotation in the frequency domain provides a good intuition towards this result. The advantage of steerable filters then stems from the fact that no discretization is needed, and the filter can be continuously steered in θ_0 to find the angle of maximum filter response. A classical example is the class of detectors h of the form

$$h(\mathbf{x}) = \sum_{n=1}^{N}\sum_{i=0}^{n} a_{n,i}\,\frac{\partial^{n-i}}{\partial x^{n-i}}\frac{\partial^{i}}{\partial y^{i}}\,g(\mathbf{x}), \tag{4.5}$$

where g is an arbitrary isotropic window function and N is referred to as the *order* of the detector. Steerable edge, ridge and wedge detectors can be constructed relying on this formalism [31].

Let us consider any polar-separable function $h_{\mathrm{pol}}^{(n)}(r,\theta) = h_{\mathrm{rad}}(r)\mathrm{e}^{\mathrm{j}n\theta}$ with $r \in \mathbb{R}^+$, $\theta \in [0, 2\pi)$, and $n \in \mathbb{Z}$. The isotropic, radial part of every $h_{\mathrm{pol}}^{(n)}$ is similar and is denoted by h_{rad}. The angular part is a purely complex exponential that depends on the value of n following $\mathrm{e}^{\mathrm{j}n\theta}$. Then, $h_{\mathrm{pol}}^{(n)}(r,\theta)$ is steerable since, under the application of a rotation matrix $\mathbf{R}_{\theta_0}$, it is equal to $h_{\mathrm{pol}}^{(n)}(r,\theta+\theta_0) = h_{\mathrm{pol}}^{(n)}(r,\theta)\mathrm{e}^{\mathrm{j}n\theta_0}$. Consequently, functions expressed as linear combinations of polar-separable basis filters $h_{\mathrm{pol}}^{(1)}, ..., h_{\mathrm{pol}}^{(N)}$ following

$$h(r,\theta) = \sum_{n\in S} c_n h_{\mathrm{pol}}^{(n)}(r,\theta), \tag{4.6}$$

with S a finite set of elements $n \in \mathbb{Z}$, are steerable as well since

$$h(r,\theta+\theta_0) = \sum_{n\in S} \mathrm{e}^{\mathrm{j}n\theta_0} c_n h_{\mathrm{pol}}^{(n)}(r,\theta). \tag{4.7}$$

Note that, in this construction, h is polar-separable as well because all $h_{\text{pol}}^{(1)}, ..., h_{\text{pol}}^{(N)}$ have the same radial part, which does not depend on n. The form (4.6) characterizes the subclass of steerable filters that we study in Sections 4.3 and 4.2.

An additional refinement is the construction of steerable wavelets, introduced by E. Simoncelli with the steerable pyramid [115]. This opens the way to the design of rotation- and scale-covariant feature detectors. In this framework, steerable filters are built following (4.6): the $h_{\text{pol}}^{(n)}(r,\theta)$ are composed of a unique isotropic wavelet profile for the radial part (that is common for each of the $h_{\text{pol}}^{(n)}(r,\theta)$), which is then given angular selectivity. The isotropic profile allows performing classical multiscale wavelet analysis, and extracting isotropic image features as well as isolated singularities. The angular selectivity generates the multiorientation counterpart, which capture directional structures such as edges and curvilinear features. Building up on this idea, M. Unser and N. Chenouard proposed a general formulation for constructing tight steerable wavelet frames relying on the Riesz transform [116, 117]. There, the generalized Riesz transform of order n of a function f is defined as

$$\mathcal{R}^n f(\mathbf{x}) \xleftrightarrow{\mathcal{F}} \mathrm{e}^{\mathrm{j}\vartheta n} \hat{f}_{\text{pol}}(w,\vartheta), \tag{4.8}$$

with $w \in \mathbb{R}+$ and $\vartheta \in [0,2\pi)$. We refer the reader to [116] for an exhaustive list of properties of the generalized Riesz transform.

To build a steerable tight wavelet frame, let us first consider $\rho(\omega)$, a purely radial profile in the frequency domain such that

1. $\rho(\omega) = 0$ for all $\omega > \pi$, (4.9)
2. $\sum_{i\in\mathbb{Z}} |\rho(2^i\omega)|^2 = 1$, (4.10)
3. $\frac{\mathrm{d}^n\rho}{\mathrm{d}\omega^n}(\omega)\big|_{\omega=0} = 0$ for $n = 0, ..., N$, for a fixed N. (4.11)

Then, the basis functions are built from the isotropic generator ψ and given by

$$\psi_{i,\mathbf{k}}(\mathbf{x}) = \psi_i(\mathbf{x} - 2i\mathbf{k}) \text{ with } \psi_i(\mathbf{x}) = 2^{-i}\psi\left(\frac{\mathbf{x}}{2i}\right) \tag{4.12}$$

with $i \in \mathbb{Z}$, $\mathbf{k} \in \mathbb{Z}^2$, and $\hat{\psi}(\boldsymbol{\omega}) = \rho(\omega)$ where $\boldsymbol{\omega} = (w\cos(\vartheta), w\sin(\vartheta))$. These basis generate a tight wavelet frame of $L_2(\mathbb{R}^2)$ and have vanishing moments up to order

N [116]. Any function $f \in L_2(\mathbb{R}^2)$ can therefore be expanded as

$$f(\mathbf{x}) = \sum_{i \in \mathbb{Z}} \sum_{\mathbf{k} \in \mathbb{Z}^2} \langle f, \psi_{i,\mathbf{k}} \rangle_{L_2} \psi_{i,\mathbf{k}}(\mathbf{x}). \tag{4.13}$$

The generalized Riesz transform (4.8) can now be combined with the wavelet frame generated by ψ to yield steerable wavelets. To do so, let us first define

$$\xi^{(n)}(\mathbf{x}) = \mathcal{R}^n \psi(\mathbf{x}) \xleftrightarrow{\mathcal{F}} \widehat{\xi^{(n)}}(\boldsymbol{\omega}) = \rho(\omega) \mathrm{e}^{\mathrm{j}\vartheta n}. \tag{4.14}$$

The isotropic wavelet at scale $i \in \mathbb{Z}$ and location (grid point) $2i\mathbf{k}$, $\mathbf{k} \in \mathbb{Z}^2$ is of the form

$$\xi_{i,\mathbf{k}}^{(n)}(\mathbf{x}) = \mathcal{R}^n \psi_{i,\mathbf{k}}(\mathbf{x}) \tag{4.15}$$

and, in Fourier domain,

$$\begin{aligned} \widehat{\xi_{i,\mathbf{k}}^{(n)}}(\boldsymbol{\omega}) = & \ \mathcal{F}\{\mathcal{R}^n \psi_i(\cdot - 2i\mathbf{k})\}(\boldsymbol{\omega}) \\ = & \ \mathrm{e}^{\mathrm{j}n\vartheta} 2^i \hat{\psi}(2^i \boldsymbol{\omega}) \mathrm{e}^{-\mathrm{j}\langle 2i\mathbf{k}, \boldsymbol{\omega} \rangle} \\ = & \ 2^i \rho(2^i \omega) \mathrm{e}^{\mathrm{j}n\vartheta - \mathrm{j}\omega_{\mathbf{k}} \omega \cos(\vartheta - \vartheta_{\mathbf{k}})}, \end{aligned} \tag{4.16}$$

with $2^i \mathbf{k} = \omega_{\mathbf{k}} \mathrm{e}^{\mathrm{j}\vartheta_{\mathbf{k}}}$.

The primal isotropic wavelet frame $\{\psi_{i,\mathbf{k}}\}$, $i \in \mathbb{Z}$, $\mathbf{k} \in \mathbb{Z}^2$ of $L_2(\mathbb{R}^2)$ can be mapped into a steerable wavelet frame $\{\psi_{i,\mathbf{k}}^{(m)}\}$, $i \in \mathbb{Z}$, $m = 1, ..., M$, $\mathbf{k} \in \mathbb{Z}^2$ of $L_2(\mathbb{R}^2)$ through the multiorder generalized Riesz transform as

$$\begin{bmatrix} \psi^{(1)} \\ \vdots \\ \psi^{(M)} \end{bmatrix} = \mathbf{U} \begin{bmatrix} \xi^{(n_1)} \\ \vdots \\ \xi^{(n_N)} \end{bmatrix}, \tag{4.17}$$

where $\mathbf{U}$ is a (possibly complex-valued) shaping matrix of size $M \times N$ with $M \geq 1$ such that $\mathbf{U}^H \mathbf{U} = \mathrm{diag}(d_1, ..., d_N)$ is diagonal with $\sum_{n \in S} d_n = 1$ [116, Proposition 4.2]. Note that the exponent H denote the Hermitian transpose of a matrix, which corresponds to the complex conjugate of its transpose. The integers of the set $S = \{n_1, ..., n_N\}$ are referred to as *harmonics* and the $m = 1, ..., M$ as *channels.* The value of N appearing in (4.11) is chosen by considering the orders of the Riesz transform in the set of harmonics [116].

The steerability property can be verified by observing two things. First, the $\psi^{(m)}$ are polar-separable in the Fourier domain since they correspond to a linear combination of polar separable basis sharing the same radial part. Second, the number of elements in the sum is finite since S is of finite dimension. We indeed have

$$\psi^{(m)}(\mathbf{x}) = \sum_{n\in S} u_{m,n}\mathcal{R}^n\psi(\mathbf{x}) \xleftrightarrow{\mathcal{F}} \widehat{\psi^{(m)}}(\boldsymbol{\omega}) = \sum_{n\in S} u_{m,n}\widehat{\xi^{(n)}}(\boldsymbol{\omega}) \tag{4.18}$$

$$= \rho(\omega)\sum_{n\in S} u_{m,n}\mathrm{e}^{\mathrm{j}\vartheta n}, \tag{4.19}$$

where the $u_{m,n}$ are the elements of the m^{th} row of $\mathbf{U}$.

For a function $f(\mathbf{x}) \in L_2(\mathbb{R}^2)$, the corresponding steerable wavelet coefficients are given by

$$w_{m,i}[\mathbf{k}] = \langle f, \psi^{(m)}_{i,\mathbf{k}}\rangle_{L_2}, \tag{4.20}$$

and allow reconstructing the signal as

$$f(\mathbf{x}) = \sum_{i\in\mathbb{Z}}\sum_{\mathbf{k}\in\mathbb{Z}^2}\sum_{m=1}^{M} w_{m,i}[\mathbf{k}]\psi^{(m)}_{i,\mathbf{k}}(\mathbf{x}). \tag{4.21}$$

We now give specific example of interesting steerable wavelet transforms that can be generated with this framework.

- The two gradient wavelets $(\partial_x\psi_1, \partial_y\psi_1) = \nabla\psi_1$ for edge detection, obtained with $S = \{-1, 0, 1\}$ and

  $$\mathbf{U} = \begin{pmatrix} \frac{\mathrm{j}}{2} & 0 & \frac{\mathrm{j}}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, \tag{4.22}$$

  where $\psi_1 = (-\Delta)^{-\frac{1}{2}}\psi$ is a smoothed version of ψ. This results in filters than attenuate noise while having an edge detector effect, as expected from standard gradient approaches.

- The three Hessian wavelets $(\partial_{xx}\psi_2, \sqrt{2}\partial_{xy}\psi_2, \partial_{yy}\psi_2)$ for ridge detection, obtained with $S = \{-2, -1, 0, 1, 2\}$ and

  $$\mathbf{U} = \begin{pmatrix} -\frac{1}{4} & 0 & -\frac{1}{2} & 0 & -\frac{1}{4} \\ -\frac{\mathrm{j}}{2\sqrt{2}} & 0 & 0 & 0 & \frac{\mathrm{j}}{2\sqrt{2}} \\ \frac{1}{4} & 0 & -\frac{1}{2} & 0 & \frac{1}{4} \end{pmatrix}, \tag{4.23}$$

where $\psi_2 = (-\Delta)^{-1}\psi$ is a smoothed version of ψ like in the gradient case.

The two key elements of steerable filters constructed as such are thus the isotropic wavelet $\rho(\omega)$, which imposes the radial profile, and the parameters of the generalized Riesz transform, namely the set of harmonics S and the shaping matrix $\mathbf{U}$, which dictate the angular behavior[2]. In the two following sections, we investigate each of these two aspects.

4.2 Tuning the Angular Profile

In our steerable filter construction, the angular profile is imposed by the choice of the harmonics and of the shaping matrix. One can design filters to detect particular features by identifying the right set of harmonics and the corresponding shaping matrix coefficients. In this spirit, our work [111] focuses on the design of shaping matrices to generate detectors for different types of symmetric patterns. To do so, we define an angular energy concentration of the form $\sum_n u_n \xi^{(n)}$, where $\xi^{(n)} = \mathcal{R}^n \psi$ as in (4.14). This amounts to considering matrices $\mathbf{U}$ that are composed of only one row, and thus yield a single wavelet. We then optimize the energy with respect to the angular weights u_n that control its concentration along the desired pattern, with $\mathbf{u} = (u_1, u_2, ..., u_N)$ and $\mathbf{u}^H\mathbf{u} = 1$. The optimal solution can be found through an eigendecomposition, similarly as in the work of D. Slepian [118].

The radial profile of the filters is here not the topic of study. We thus choose a radial profile that remains fixed throughout this section. We pick Simoncellis isotropic wavelet [119], which is inspired from biological vision and has several well-established applications in image analysis. We recall its definition as

$$\rho(\omega) = \begin{cases} \cos\left(\frac{\pi}{2}\log_2\left(\frac{2\omega}{\pi}\right)\right) & \frac{\pi}{4} < |\omega| < \pi, \\ 0 & \text{otherwise.} \end{cases} \tag{4.24}$$

It is worth noting that Simoncelli's isotropic wavelet satisfies conditions (4.9), (4.10), and (4.11) for any value of $N \geq 0$ [120].

[2] The generalized Riesz-wavelet transform has been implemented for various useful shaping matrices as a Java-based plug-in for the open-source image-analysis software ImageJ [9].

4.2.1 Design of Specific Angular Profiles

We briefly review the core ideas introduced in [111] and refer the interested reader to the paper for more mathematical details.

K-fold Symmetries

To design a detectors of patterns with K-fold symmetries, we propose to impose the same K-fold-symmetric pattern on ψ. We achieve this by minimizing a well chosen energy functional. In the case of a unimodal detector, we are looking for the angular profile that is most concentrated around the angle 0. We are thus targeting a minimum-variance solution, which results in an energy that concentrates around a chosen axis. We can formulate this energy optimization either in the spatial or Fourier domain. Both designs are appropriate, each one having its own advantages and drawbacks.

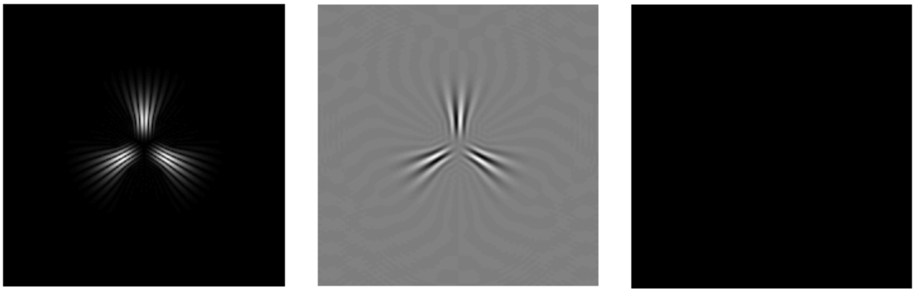
The spatial quadratic energy term we propose to minimize is of the form

$$E\{\psi\} = \frac{1}{2\pi}\int_0^\infty \int_{-\pi}^{\pi} |\psi(r,\theta)|^2 w(\theta)\mathrm{d}\theta r\mathrm{d}r, \tag{4.25}$$

where w is an K-fold-symmetric non-negative weighting function composed of K equidistant minima on the unit circle in order to favor K-fold-symmetric solutions. Minimizing E forces the solution ψ to be symmetrically localized near the K-fold minima. Once the wavelet ψ is found, its translations and dilations will naturally share the optimal angular profile around their center. Expanding ψ as $\sum_{n\in S} u_n\xi^{(n)}$ and imposing $\mathbf{u}$ to be normalized leads to a quadratic optimization problem with quadratic constraints that can be solved through an eigendecomposition. Noting that the $\xi^{(n)}$ are polar-separable, the problem can be reformulated as the minimization of

$$E = \mathbf{u}^H\mathbf{W}\mathbf{u} \tag{4.26}$$

with quadratic constraints ($\mathbf{u}^H\mathbf{u} = 1$). Each entry of the matrix $\mathbf{W}$ is composed of an angular part corresponding to the Fourier coefficients of $w(\theta)$ and of a radial part computed from the radial part of $\xi^{(n)}$, which results in Hankel-like integrals. For detailed computations and the precise definition of $\mathbf{W}$, we point the reader to [111, Equations (22) to (26)]. Since (4.26) is a standard eigenvalue problem, minimizing the energy functional is equivalent to finding the eigenvector that corresponds to the smallest eigenvalue of $\mathbf{W}$. We note that the matrix $\mathbf{W}$ is positive-semi-definite



(a) (b) (c)

Figure 4.1: (a) Magnitude, (b) real, and (c) imaginary parts of a three-fold-symmetric wavelet designed using the energy functional in the space domain and visualized in the space domain, with $S = \{3n : n = -8, ..., 8\}$ as harmonics. Note that, since the chosen set of harmonics is symmetric, the filter is purely real and its imaginary part is equal to zero.

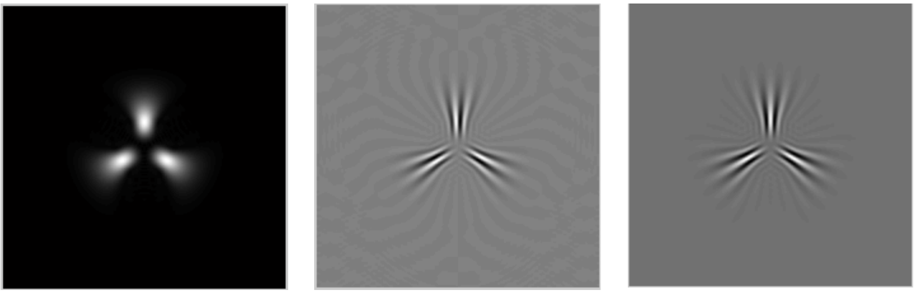
since it is derived from a non-negative weight function, which itself defines a non-negative energy. A particular example of a three-fold-symmetric wavelet obtained by the proposed method is shown in Figure 4.1. The weight function w is $\frac{2\pi}{3}$-periodic with $w(\theta) = \left(\frac{3\theta}{\pi}\right)^2$ for $|\theta| \leq \frac{\pi}{3}$ to achieve maximal energy concentration about the minima in the sense of variance. An example of a complex threefold-symmetric wavelet obtained by the proposed method is presented in Figure 4.2.

Since rotations and symmetries in the space domain carry over to the frequency domain, we can formulate our minimization problem in either one. The advantage of the frequency-domain formulation is to avoid the computation of Hankel-like integrals since, in the Fourier domain, the radial part of the wavelet has no effect on the optimization.

The Fourier-domain energy functional E is defined as

$$E\{\widehat{\psi}\} = \frac{1}{2\pi} \int_0^\infty \int_{-\pi}^{\pi} |\widehat{\psi}(\omega, \vartheta)|^2 w(\vartheta) \mathrm{d}\vartheta r \mathrm{d}r, \tag{4.27}$$

where $w \geq 0$ is an K-fold-symmetric weighting function with K equidistant minima on the unit circle. This forces the solution $\widehat{\psi}$ to be localized symmetrically near



(a) (b) (c)

Figure 4.2: (a) Magnitude, (b) real, and (c) imaginary parts of a three-fold-symmetric complex wavelet designed using the energy functional in the space domain and visualized in the space domain, with $S = \{3n : n = 0, ..., 9\}$ as harmonics.

the K-fold minima through the minimization of E. Relying again on the polar-separability of $\widehat{\psi}$, one observes that the radial part of the wavelet function can be factored out from the integral and thus plays no role in the optimization process. As a result, one can again formulate the problem as (4.26), where $\mathbf{W}$ is this time only composed of the coefficients of the Fourier serie of w ([111, Equations (31) and (32)]). We again end up with an eigenvalue problem. A particular example of a three-fold-symmetric wavelet obtained using the Fourier approach can be seen in Figure 4.3. The weight function w is $\frac{2\pi}{M}$-periodic with $w(\vartheta) = \left(\frac{M\vartheta}{\pi}\right)^2$ for $|\vartheta| \leq \frac{\pi}{M}$ to achieve maximal energy concentration around the minima in the sense of variance. Examples of higher-fold-symmetric wavelets can be seen in Figure 4.4.

It is worth noting that execution speed for both approaches pertains only to the initial design of the wavelets (*i.e.*, the calculation of the design matrix $\mathbf{U}$). Once the wavelets have been designed, the analysis and processing of images runs at the same speed for a given number of harmonics, independent on the design. Both approaches provide a comprehensive basis to detect symmetric junctions. The Fourier-domain design yields easier computations, since the radial part of the wavelet has no effect on the optimization process. The space-domain design is more cumbersome as it requires to handle the Hankel-like integrals associated with the radial part of the wavelet. In return, it sidesteps the limitations of the Fourier domain that forbid

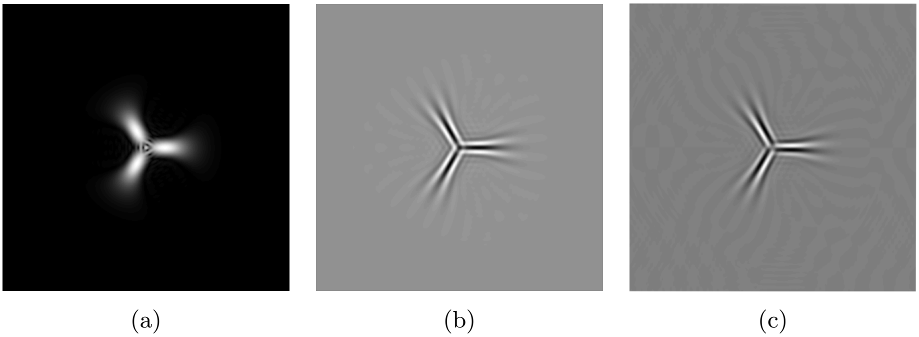


Figure 4.3: (a) Magnitude, (b) real, and (c) imaginary parts of a three-fold-symmetric wavelet designed using the energy functional in the Fourier domain and visualized in the space domain, with $S = \{3n : n = 0, ..., 9\}$ as harmonics.

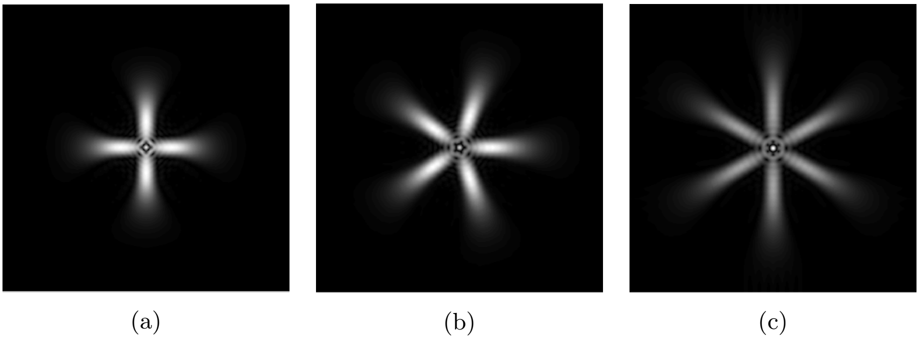


Figure 4.4: Modulus of (a) four-, (b) five-, and (c) six-fold-symmetric wavelets designed using the energy functional in the Fourier domain and visualized in the space domain, with $S = \{Mn : n = 0, ..., 9\}$ as harmonics.

wavelets that are asymmetric and real [111, Proposition 3]. While the Fourier design is not appropriate to obtain odd-symmetric real wavelets, it does not impair the detection of junctions since complex odd designs are still possible relying on positive harmonics alone. Also, defining the energy functional $E\{\psi\}$ in the space domain allows obtaining purely real wavelets.

Asymmetric Patterns

Our framework is flexible and easy to generalize to many classes of local structures and junctions. Here, we take another look at the angular weighting functions and propose to design asymmetric wavelets, targeting shapes like corners or T-junctions. The advantage, as before, is that we can steer these wavelets to any arbitrary angle by a systematic complex rescaling of the wavelet coefficients. The Fourier technique presented earlier is not suitable for the design of asymmetric wavelets since it relies on the equivalence of rotational symmetries between the spatial and Fourier domains. In the case of the symmetric three-fold junction from previous section, we had identified a profile that, after proper steering, was simultaneously concentrated around zero, $\frac{2\pi}{3}$, and $\frac{-2\pi}{3}$. Following our energy-minimization paradigm, we now choose w such that it has three identical minima at 0, $\frac{2\pi}{3}$, and $\frac{-2\pi}{3}$. This function is given by the parabola $\left(\frac{3\theta}{\pi}\right)^2$ over the interval $[-\frac{\pi}{3}, \frac{\pi}{3}]$ as well as its translates to the intervals $[\frac{\pi}{3}, \pi]$ and $[-\frac{\pi}{3}, -\pi]$. The specific expression of the resulting function w_{S3} on the $[-\pi, \pi]$ interval is given in [111, Equation (33)].

In the case of a T junction, we intend to identify profiles that, after proper steering, are simultaneously concentrated around zero, $\frac{\pi}{2}$, and $-\frac{\pi}{2}$. Similarly to the symmetric case, we take w to have three identical minima at 0, $\frac{\pi}{2}$, and $-\frac{\pi}{2}$, given by the parabola $\left(\frac{4\theta}{\pi}\right)^2$ over the interval $[-\frac{\pi}{4}, \frac{\pi}{4}]$ and its translates to the intervals $[\frac{\pi}{4}, \frac{3\pi}{4}]$ and $[-\frac{3\pi}{4}, -\frac{\pi}{4}]$. We keep w_T constant over the intervals $[\frac{3\pi}{4}, \pi]$ and $[-\pi, -\frac{3\pi}{4}]$. The expression of the function w_T on the $[-\pi, \pi]$ interval is given in [111, Equation (35)].

Finally, the perpendicular corner is obtained in a similar fashion. The weight function consists of two identical parabolas placed at $-\frac{\pi}{4}$ and $\frac{\pi}{4}$. We keep w_C constant over the intervals $[\frac{\pi}{2}, \pi]$ and $[-\pi, -\frac{\pi}{2}]$. This leads to the function given in [111, Equation (37)].

4.2.2 Detection Algorithm

Suppose that a K-fold symmetric steerable wavelet is available. Keypoints in the image correspond to maxima in the response of the wavelet detector. The wavelets thus have to be *steered* to search for the orientation yielding the largest response. The optimal steering angle at each point in the image can be found by maximizing a trigonometric polynomial in θ_0, which is achieved by computing the roots of its first-order derivative. The wavelet coefficients obtained when filtering the image are given by

$$\begin{aligned} w_i[\mathbf{k}] =& \langle f, \sum_{n\in S} u_n \xi_{i,\mathbf{k}}^{(n)} \rangle_{L_2} = \sum_{n\in S} u_n \langle f, \xi_{i,\mathbf{k}}^{(n)} \rangle_{L_2} && (4.28) \\ =& \sum_{n\in S} u_n \langle f, \mathcal{R}^n \psi_{i,\mathbf{k}} \rangle_{L_2} = \sum_{n\in S} u_n q_{i,n}[\mathbf{k}], && (4.29) \end{aligned}$$

where $\{q_{i,1}[\mathbf{k}], ..., q_{i,N}[\mathbf{k}]\}$ denote the wavelet coefficients for each of the N harmonics computed by analyzing the input image at scale i and position $\mathbf{k}$.

To perform the steering per se, we rotate the wavelet ψ at each $(i, \mathbf{k})$ and find the maximum of $|\langle \psi_{i,\mathbf{k}}(\cdot, \cdot + \theta_0), f\rangle_{L_2}|$ as a function of θ_0. For each $(i, \mathbf{k})$, we use (4.7) and rewrite the function to maximize as

$$Q(i, \mathbf{k}, \theta_0) = \left| \sum_{n\in S} \mathrm{e}^{\mathrm{j}n\theta_0} u_n \langle f, \mathcal{R}^n \psi_{i,\mathbf{k}} \rangle_{L_2} \right|^2 . \qquad (4.30)$$

The maximization of (4.30) can be achieved by performing root-finding on its derivative with respect to θ_0. The most basic method to solve the problem relies on brute-force computations, although alternative approaches can also be used to speed up the maximization process.

Finally, our algorithm to detect junctions of a given multiplicity along with their orientation can be decomposed in the three following steps.

1. **Wavelet analysis with optimally steered wavelets.** We decompose the image with the steerable wavelet. At each location and scale, the local orientation $\theta_0^*(i, \mathbf{k})$ that maximizes the detector response (4.30) is determined. The output of this first stage is a map of maximal steerable wavelet responses $\mu(i, \mathbf{k})$ and orientations at every scale and location in the wavelet decomposition of the image. These correspond to multiscale versions of (4.2) and (4.3).

2. **Maximal projection across scales.** We perform a maximum-intensity projection to aggregate the keypoints detected at different scales. The detectors have normalized energy across scales, which motivates the choice of a maximal projection. For each position, we keep the coefficient corresponding to the largest detector response as

$$\mu(\mathbf{k}) = \max_i \mu(i, \mathbf{k}). \tag{4.31}$$

3. **Thresholding and the detection of local maxima.** We assume that the pixels that correspond to features are sparse in the image. Points with a detector response smaller than a threshold are assigned as background pixels. We additionally apply local non-maximum suppression over a user-predefined window in order to prevent detecting the same feature multiple times.

4.2.3 Examples of Application

Our symmetry detection algorithm has been implemented as a plug-in for the open-source image processing software ImageJ [9]. There, we use Simoncellis isotropic wavelets [119] with dyadic scale progressions. To evaluate the performance of the algorithm, we tested it on a variety of synthetic images in the presence of noise and, finally, on real microscopic images. The aim of these experiments is to assess to which extent our method honors translation, rotation, and scale covariance, how robust it is to noise and how good its detection performance are.

A thorough evaluation of two versions of the space design (a real version with positive and negative harmonic pairs and a complex version with positive harmonics only) is provided in [111]. There, these results are compared with the ones given by the Fourier template. The robustness against noise, rotation invariance and multiscale properties of our design is studied in synthetic images. The resulting conclusions can be summarized as follows. First, the Fourier and the complex spatial design perform well in the presence of noise. The real spatial design has difficulties under heavy noise and exhibits small errors even in the noise-free case. This can be explained in two ways. First, the shape of the wavelet contains oscillations related to the Hankel functions. When matching, the algorithm tries to fit the junction with the oscillations, such that the maxima overlap. Second, when generating the wavelet, we have to use more harmonics to achieve the same detection range (17 in total). When steering the wavelets, these extra harmonics mean

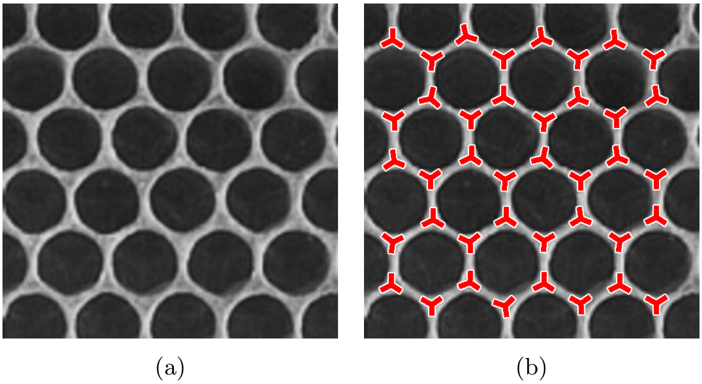
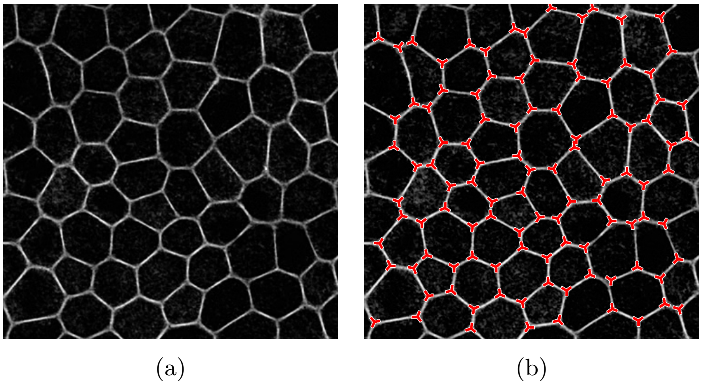


Figure 4.5: Three-fold junctions in a brightfield microscopic image of a honeycomb. (a) Original 256 × 256 pixels image. (b) Detection results.

that we have to find the roots of a polynomial of a much higher order (17 instead of 10), thus increasing numerical errors. The method is observed to perform well independently on the orientation of the junctions, as expected for a rotation-covariant approach. Finally, we could show that our method is essentially scale-covariant and yields good detection results, independently on the scale of the pattern.

We illustrate the use of our method in three types of real bioimages. First, Figure 4.5 features a microscopic view of a honeycomb. This biological structure is composed of a mass of cells that naturally exhibit a close-to-perfect hexagonal structure. The detection results of our method are shown on the right. They are accurate, which is expected from the regularity of the structure. Second, the image of Figure 4.6 contains hexagonal embryonic stem cells imaged by fluorescence microscopy featuring three-fold symmetric junctions. The detection task is more challenging than in the honeycomb image due to the variety of orientations and sizes of the junctions. In this case, the method yields again good results, which suggests that our algorithm works well even if the structures are only semi-regular and corrupted by photon counting noise. Finally, Figure 4.7 features endothelial cells of the cornea along with the detection results.



(a) (b)

Figure 4.6: Three-fold junctions in a fluorescence microscopic image of embryonic stem cells. (a) Original (512 × 512) image. (b) Detection results.

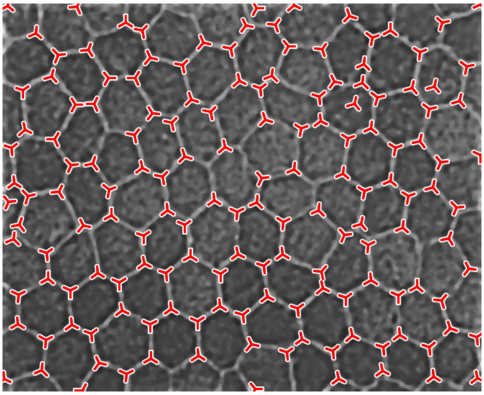


Figure 4.7: Result of the detection of three-fold junctions in an image of endothelial cells of the cornea.

4.3 Tuning the Radial Profile

The second key element of our steerable filter construction is the isotropic profile that defines the radial part of the detector. Studying the design of such isotropic profiles and how it affects results in different image processing tasks was the topic of our work [113].

In order to generate an isotropic wavelet transform, the underlying basis functions must satisfy several properties. The main ones are isotropy and perfect reconstruction of the image. Another desirable feature is that the basis functions form a tight frame. In this way, the wavelet transform is self-reversible, enabling simpler and faster algorithms. The isotropy and perfect-reconstruction properties are ensured by choosing a radially bandlimited wavelet profile that satisfies a partition of unity condition in the frequency domain [121, 122]. Many such bases have been proposed, including the Meyer [123], Papadakis [124], and Simoncelli [119] wavelets. The Simoncelli wavelet is the one implemented in the original version of the steerable pyramid. Due to its good performance in a wide range of practical applications, it is a commonly used profile.

Localization, either in the spatial or frequency domain, is an important consideration when specifying such wavelets. On the practical side, steerable wavelets are bandlimited with infinitely many vanishing moments, which tends to induce oscillations that can be visually displeasing. It is observed that more localized wavelets result in fewer oscillations and are less subject to truncation artifacts. Moreover, it has been theoretically shown that wavelets with better localization are more efficient for decoupling sparse signals [125]. It is worth mentioning that the Simoncelli wavelet is shown to be the most-localized wavelet in a specific sense [126].

We introduce a method based on calculus of variations to design localized radial profiles for steerable tight frames. Since the frequency response of steerable wavelets is polar-separable, we can concentrate on the task of optimizing the radial frequency profile. We focus on moment-based measures of localization and propose two different classes of criteria depending on whether we consider the localization in the spatial domain or in the wavelet domain. Two criteria can be derived within each class, depending on whether one considers localization over the whole space or in each radial direction. Ultimately, the choice of a particular criterion is guided by the application.

4.3.1 Preliminaries on Isotropic Wavelets

We here specify the class of wavelets that we consider. Let the mother wavelet ψ be a function from $\mathbb{R}^2$ to $\mathbb{R}$. The complete wavelet frame is given by

$$\psi_{i,\mathbf{k}}(\mathbf{x}) = 2^{-i}\psi\left(\frac{\mathbf{x}}{2^i} - \mathbf{k}\right), \tag{4.32}$$

in which $i \in \mathbb{Z}$ and $\mathbf{k} \in \mathbb{Z}^2$. It is a tight frame by construction, which implies that conditions (4.9), (4.10), and (4.11), also enunciated in [121, 122], are verified. We also assume that ψ is isotropic. Thus, following the notations we introduced at the beginning of this chapter, we write

$$\psi(\mathbf{x}) = \psi(r), \tag{4.33}$$

where $r = \|\mathbf{x}\|$. For simplicity, we use the notation ψ for both the wavelet and its radial profile. Now, let $\hat{\psi}$ be the two-dimensional Fourier transform of ψ as defined in (4.1). According to the properties of the Fourier transform, $\hat{\psi}$ is also isotropic. Hence, similarly as in (4.33), we write

$$\hat{\psi}(\boldsymbol{\omega}) = \rho(\omega), \tag{4.34}$$

where $\omega = \|\boldsymbol{\omega}\|$. According to [127], ψ and ρ are related through the Hankel transform as

$$\psi(r) = \mathcal{H}\{\rho\}(r) = \int_0^\infty \rho(s)\ J_0(rs)\ s\,\mathrm{d}s \tag{4.35}$$

for $r \geq 0$, with J_0 a Bessel function of the first kind. With these preliminaries, we are ready to propose a framework to find the most-localized wavelet among the ones that satisfy the present conditions. The following method can be used for a broad class of measures of localization.

4.3.2 Designing Localized Profiles

Our general framework relies on calculus of variations to find the optimal wavelet with respect to a given localization measure. We focus on two natural classes of moment-based measures.

Let us assume that the functional V is a given measure of localization. We recall that a functional is a mathematical object which takes as argument a function and associates a value to it. We shall consider that this measure operates in the Fourier domain. When V is Gâteaux differentiable, the natural method of minimizing it is to use a variation of the steepest-descent algorithm [128]. To do this, we first need the gradient of the localization criterion V. Since V is a functional on an infinite-dimensional space, we have to rely on calculus of variations to obtain its gradient. Then, during the optimization steps, we have to be careful not to leave the set of tight frames. We thus need to characterize $\mathcal{P}$, the orthogonal projector onto the space of tight wavelet frames. Technical details on how to obtain these two components, namely the gradient of V and the projector $\mathcal{P}$, are provided in [113, Section III.B]. The outline of the optimization algorithm is given in Algorithm 1, where ∇V denotes the gradient of V. It corresponds to the standard projected gradient descent algorithm in an infinite dimensional space [129].

Algorithm 1: Most Localized Wavelet

```
initialize:  $\rho \in L_2([0,\infty])$
initialize:  $\eta > 0$
repeat
    $\tilde{\rho} \leftarrow \rho - \eta \, \nabla V\{\rho\}$
    $\rho \leftarrow \mathcal{P}\{\tilde{\rho}\}$
until $\rho$ converges
return  $\rho$
```

Measures of Localization

We propose four measures of localization split in two natural classes. The first class consists of measures of the variance. Variance is the most well-known measure of localization as less variance implies more concentration around the center. In addition, we know from the uncertainty principle that the best achievable localization of a function is inversely proportional to the localization of its Fourier transform, and vice versa. More precisely, for a function $\psi \in L_2(\mathbb{R}^2)$ with $\psi : \mathbb{R}^2 \to \mathbb{R}$, we have [130]

$$\frac{\int_{\mathbb{R}^2} \|\mathbf{x}\|^2 \psi(\mathbf{x})^2 \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^2} \psi(\mathbf{x})^2 \mathrm{d}\mathbf{x}} \, \frac{\int_{\mathbb{R}^2} \|\boldsymbol{\omega}\|^2 |\mathcal{F}\{\psi\}(\boldsymbol{\omega})|^2 \mathrm{d}\boldsymbol{\omega}}{\int_{\mathbb{R}^2} |\mathcal{F}\{\psi\}(\boldsymbol{\omega})|^2 \mathrm{d}\boldsymbol{\omega}} \geq \frac{1}{16\pi^4}. \tag{4.36}$$

The first term of the left-hand side is the variance of the wavelet itself, and the second term is the variance of its Fourier transform. Thus, setting an upper bound on the variance in one domain imposes a lower bound on the variance in the other domain. In practice, we are interested in bandlimited wavelets, which implies that the variance of the wavelet in the Fourier domain is bounded from above. Thus, the variance of the wavelet in the space domain is bounded from below. Since we are interested in finding the wavelet profile that reaches the minimum value for the variance, our first measure of localization is given by

$$V_{2\mathrm{D}}(\psi) = \frac{\int_{\mathbb{R}^2} \|\mathbf{x}\|^2 \psi(\mathbf{x})^2 \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^2} \psi(\mathbf{x})^2 \mathrm{d}\mathbf{x}} = \frac{\int_0^\infty r^2 \psi(r)^2 \ r\mathrm{d}r}{\int_0^\infty \psi(r)^2 \ r\mathrm{d}r}. \tag{4.37}$$

We are also interested in isotropic wavelets used in a directional framework. This suggests that the variance of the one-dimensional radial profile of the isotropic wavelet can be another good candidate measure of localization. Hence, we propose the second variance-based measure of localization

$$V_{1\mathrm{D}}(\psi) = \frac{\int_0^\infty r^2 \psi(r)^2 \ \mathrm{d}r}{\int_0^\infty \psi(r)^2 \ \mathrm{d}r}, \tag{4.38}$$

which measures the spread of the wavelet along each radial line.

The second class of measures focuses on the localization of the wavelet coefficients rather than that of the wavelet profile. More precisely, the energy of a function computed over some spatial neighborhood should be well represented by the wavelet coefficients associated to that neighborhood and its vicinity. According to [126], as a wavelet $\psi_{i,\mathbf{k}}$ gets further from a position $\mathbf{m}$, the contribution of the local neighborhood of the image around $\mathbf{m}$ in the corresponding wavelet coefficient decays. Moreover, the rate of decay is controlled by a constant that corresponds to the unnormalized variance of the wavelet profile ψ. We propose this value as a third measure of localization, this time for the wavelet coefficients, as

$$U_{2\mathrm{D}}(\psi) = \int_{\mathbb{R}^2} \|\mathbf{x}\|^2 \psi(\mathbf{x})^2 \mathrm{d}\mathbf{x} = \int_0^\infty r^2 \psi(r)^2 \ r\mathrm{d}r. \tag{4.39}$$

Accounting for the fact that isotropic wavelets are often used in a directional setting, we define the unnormalized variance of the one-dimensional radial profile of the wavelet in analogy to (4.38) as our last measure of localization

$$U_{1\mathrm{D}}(\psi) = \int_0^\infty r^2 \psi(r)^2 \ \mathrm{d}r. \tag{4.40}$$

To summarize, the first class of measures targets the localization of the shape of the wavelet in the space domain while the second class of measures describes the localization of the wavelet coefficients. We thus expect the first class to be useful in applications that benefit from a local wavelet analysis. The second class should, on the contrary, be more appropriate in applications involving some form of wavelet-domain N-term approximation. For both classes of measures, the index 2D indicates that the spread of the wavelet is a measure over the whole space. The index 1D, conversely, indicates that the spread is measured along each radial direction. In detection tasks using steerable filters, we predict that profiles with the best radial localization will obtain the best performances.

Numerical Optimization

In practice, we are interested in a fast implementation of wavelet transforms relying on filter-banks. Focusing on wavelets that are supported on $[\frac{\pi}{4}, \pi]^3$, the highpass and lowpass filters of the filter-bank, denoted as ρ_H and ρ_L, respectively, are given by

$$\rho_H(\omega) = \begin{cases} 0 & \omega \leq \frac{\pi}{4} \\ \rho(\omega) & \frac{\pi}{4} < \omega < \frac{\pi}{2} \\ 1 & \frac{\pi}{2} \leq \omega \end{cases}, \tag{4.41}$$

$$\rho_L(\omega) = \begin{cases} 1 & \omega \leq \frac{\pi}{4} \\ \rho(2\omega) & \frac{\pi}{4} < \omega < \frac{\pi}{2} \\ 0 & \pi < \omega \end{cases}, \tag{4.42}$$

respectively. According to the tight-frame constraint (4.10), we automatically have that

$$\rho_L^2(\omega) + \rho_H^2(\omega) = 1. \tag{4.43}$$

Hence, when searching for the minimizer of a given criterion, we optimize the values of the function $\rho(\omega)$ over the $[\frac{\pi}{4}, \pi]$ interval. In this spirit, it is worth noting that a filter-bank construction for one-dimensional tight wavelet frames with arbitrary dilation has been proposed in [131]. The resulting wavelets have more frequency

[3] We refer the reader to [115] for a detailed explanation.

contents than the ones we consider in this work and are thus more localized in the space domain. However, there is no simple relation between the filters and the wavelet profiles, which hinders the optimization of the localization of such profiles.

We apply our numerical optimization algorithm to each of the proposed measures of localization. First of all, it can be shown analytically that the Simoncelli wavelet minimizes the criterion U_{2D} [126]. Thus, we already know the optimal profile with respect to this measure. To minimize V_{2D}, V_{1D}, and U_{1D}, we rely on Algorithm 1. The parameters and details of the optimization procedure are provided in [113]. At convergence, we obtain the minimum values of 1.73, 0.39, and 1.64 for $V_{\mathrm{2D}}(\rho)$, $V_{\mathrm{1D}}(\rho)$, and $U_{\mathrm{1D}}(\rho)$, respectively. However, due to the Gibbs phenomenon that results from the truncation of the Fourier transform, the wavelets exhibit ringing artifacts. In order to remove these effects and obtain smooth profiles for practical applications, we identify the closed-form formula that fits the best each numerically obtained wavelet. We propose four wavelets named $\rho_{V_{\mathrm{2D}}}$, $\rho_{U_{\mathrm{2D}}}$, $\rho_{V_{\mathrm{1D}}}$, and $\rho_{U_{\mathrm{1D}}}$, which correspond to each of the considered measures of localization. From (4.43), it is sufficient to specify the wavelet profile either over $[\frac{\pi}{4}, \frac{\pi}{2}]$ or over $[\frac{\pi}{2}, \pi]$ to describe it entirely. The expressions of $\rho_{V_{\mathrm{2D}}}$ and $\rho_{U_{\mathrm{2D}}}$ are easier when their high-pass filters are given as

$$\rho_{V_{\mathrm{2D}},H}(\omega) = \frac{\sqrt{6} - \sqrt{1 + 20\left(\frac{2\omega}{\pi} - 1\right)^2}}{\sqrt{6} - 1}, \tag{4.44}$$

$$\rho_{U_{\mathrm{2D}},H}(\omega) = \cos\left(\frac{\pi}{2}\log_2 \frac{2\omega}{\pi}\right). \tag{4.45}$$

We recall that $\rho_{U_{\mathrm{2D}}}$ is Simoncelli's wavelet. The profiles $\rho_{V_{\mathrm{1D}}}$ and $\rho_{U_{\mathrm{1D}}}$, conversely, are better expressed through their low-pass filters as

$$\rho_{V_{\mathrm{1D}},L}(\omega) = \left(\log_2 \frac{\pi}{\omega} - 0.005\sqrt{\frac{\pi}{\omega}}\sin\left(\pi \log_2 \frac{\pi}{\omega}\right)\right)^{\frac{2}{5}}, \tag{4.46}$$

$$\rho_{U_{\mathrm{1D}},L}(\omega) = \sqrt{\frac{\left(\log_2 \frac{\pi}{\omega} + 0.6\right)^4 - 0.6^4}{1.6^4 - 0.6^4}}. \tag{4.47}$$

The radial profiles of these wavelets are shown in Figures 4.8 and 4.9 in Fourier and space domains, respectively. We have that $V_{\mathrm{2D}}(\rho_{V_{\mathrm{2D}}}) = 1.74$, $V_{\mathrm{1D}}(\rho_{V_{\mathrm{1D}}}) = 0.40$, and $U_{\mathrm{1D}}(\rho_{U_{\mathrm{1D}}}) = 1.65$. These values are only marginally suboptimal. The values of

Table 4.1: Localization of different wavelets measured by V_{2D}, V_{1D}, U_{2D}, and U_{1D} ((4.37), (4.38), (4.39), and (4.40))

Wavelet type	Localization			
	V_{2D}	V_{1D}	U_{2D}	U_{1D}
$\rho_{V_{\mathrm{2D}}}$ (4.44)	**1.74**	0.44	3.88	2.19
$\rho_{V_{\mathrm{1D}}}$ (4.46)	$+\infty$	**0.40**	$+\infty$	2.03
$\rho_{U_{\mathrm{2D}}}$ (Simoncelli) (4.45)	1.84	0.46	**3.55**	1.93
$\rho_{U_{\mathrm{1D}}}$ (4.47)	$+\infty$	0.52	$+\infty$	**1.65**
Papadakis [124]	2.06	0.49	4.93	2.52
Meyer [123]	2.88	0.66	6.04	2.61
Shannon [116]	$+\infty$	$+\infty$	$+\infty$	$+\infty$

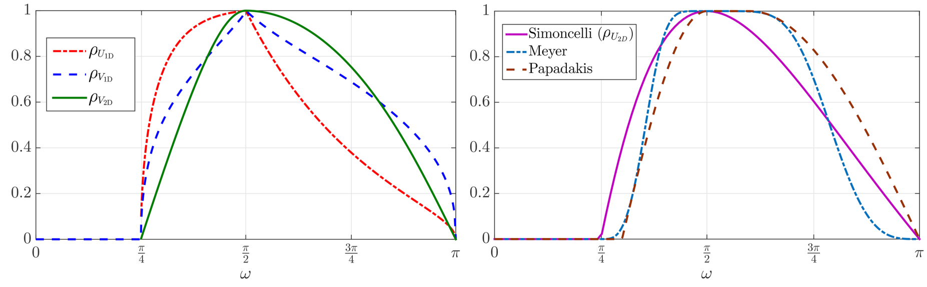
the different measures of localization for each of these wavelets as well as for more traditional ones are given in Table 4.1.

The measures V_{2D} and U_{2D} can also be interpreted as the normalized and unnormalized third-order moment of the radial profile of the wavelet, respectively, while V_{1D} and U_{1D} correspond to its normalized and unnormalized second-order moment. Since having finite values for higher-order moments in the space domain implies being smoother in the Fourier domain, the minimiziers of V_{2D} and U_{2D} necessarily have finite V_{1D} and U_{1D} values. The converse is however not always true.

As seen in Table 4.1, the localization values of the profiles minimizing V_{1D} and U_{1D} explode for V_{2D} and U_{2D}. This is in accordance with the roughness of the profiles at points $\frac{\pi}{4}$, $\frac{\pi}{2}$, and π, as seen in Figure 4.8. In the case of the Shannon wavelet, the discontinuity in the frequency-domain profile results in a slow decay in the spatial domain that brings V_{1D} and U_{1D} to infinity. From Table 4.1, we observe that the Simoncelli wavelet is not only optimal for the criterion U_{2D}, but also exhibits reasonable values for other measures of localization.

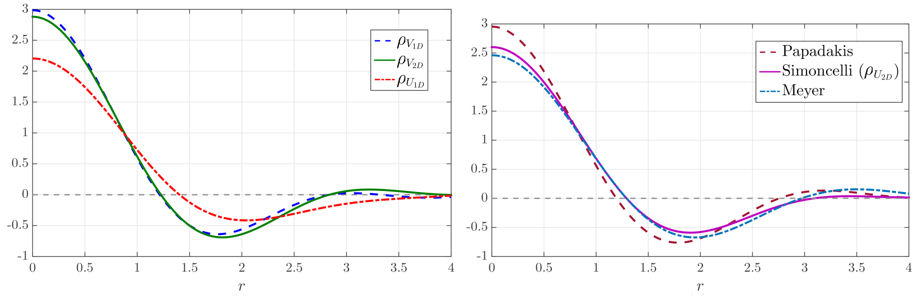
4.3.3 Examples of Application

We demonstrate the benefit of well-localized wavelet profiles in practical applications. In particular, we study the performance of our wavelets and compare them



(a) (b)

Figure 4.8: Wavelet profiles in Fourier domain of (a) the proposed optimal profiles, and (b) existing ones (Simoncelli, Meyer, and Papadakis).



(a) (b)

Figure 4.9: Wavelet profiles in space domain of (a) the proposed optimal profiles, and (b) existing ones (Simoncelli, Meyer, and Papadakis).

against other existing popular profiles for the problems of local orientation estimation and image reconstruction from edges.

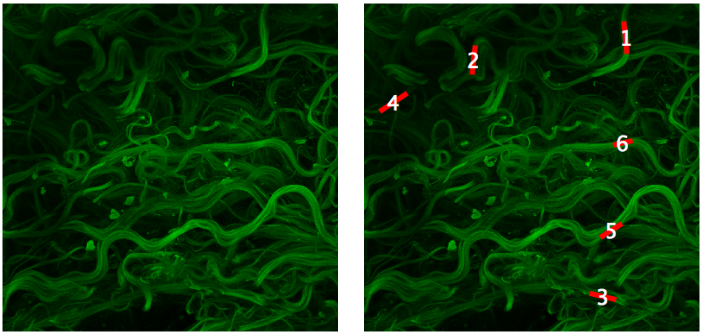
Estimation of Local Orientation

First, we focus on the task of estimating the local orientation of ridge-like objects (*e.g.*, filaments) using a steerable ridge detector. We construct Hessian-like wavelets by selecting an isotropic profile and applying the generalized Riesz-wavelet transform with the appropriate shaping matrix (4.23), as described in Section 4.1. The Hessian filters are steered to retrieve the orientation corresponding to the best response of the ridge detector at every point of the image. To perform a multiscale ridge detection at every location using the Hessian filter, we go through every scale of the wavelet pyramid and select the one where the strongest filter response is found, in a similar procedure as presented in Section 4.2.2. The final output of our experiment is therefore an angle map of the same dimensions than those of the input image, which contains at each pixel the estimated local orientation yielding the best ridge filter response[4].

In our experiment, we rely on a 512 × 512 pixels 8-bits image (Figure 4.10a) in which several regions of interest (ROIs) made of short line segments have been manually selected by an expert, and where local orientation should be estimated. The angle that each of the manually placed ROI form with the horizontal direction is considered as ground truth and corresponds to the orientation that shall be retrieved. To build our filters, we consider the trivial isotropic profile[5] (Shannon [116]), several popular isotropic wavelet profiles (Simoncelli [132], which corresponds to $\rho_{U_{\mathrm{2D}}}$ (4.45), Papadakis [124], and Meyer [123]), as well as with the wavelets we propose ($\rho_{U_{\mathrm{1D}}}$ (4.47), $\rho_{V_{\mathrm{2D}}}$ (4.44) and $\rho_{V_{\mathrm{1D}}}$ (4.46)). The estimation of the local orientation of each ROI is obtained by averaging the orientation estimates provided by the steerable filter under the ROI (*i.e.*, we average the values of the pixels that belong to the line segment composing the ROI). We indicate the estimated local orientation as well as the ground truth orientation in Table 4.2. We also report the absolute error between the ground truth and each of the estimates in Table 4.3. The experiment is conducted using 4 scales of wavelet decomposition.

[4] The extraction of local orientation estimation at each point of the input image has been implemented as a Java-based plug-in for the open-source image-analysis software ImageJ [9].

[5] Given by $\rho(\omega) = \begin{cases} 1 & \frac{\pi}{2} \leq \omega < \pi \\ 0 & \text{elsewhere} \end{cases}$.



(a) (b)

Figure 4.10: Estimation of local orientation. (a) Original filaments image from [133], and (b) overlaid regions of interest with their label.

Best results are obtained with $\rho_{U_{1\mathrm{D}}}$. The two classes of wavelets ρ_U and ρ_V are optimized for different applications. The profiles of $\rho_{V_{2\mathrm{D}}}$ and $\rho_{V_{1\mathrm{D}}}$ are most localized in the spatial domain as they optimize $V_{2\mathrm{D}}$ (4.37) and $V_{1\mathrm{D}}$ (4.38), while $\rho_{U_{2\mathrm{D}}}$ and $\rho_{U_{1\mathrm{D}}}$ optimally localize wavelet coefficients following $U_{2\mathrm{D}}$ (4.39) and $U_{1\mathrm{D}}$ (4.40). The estimation of local orientation is better when the wavelet response is strong and well localized, as ridges (here, the filaments) are then more accurately detected. In this experiment, a profile maximizing wavelet coefficients localization, and hence a criterion of the class U, is therefore preferable. This is confirmed by the results and the good performance obtained by $\rho_{U_{1\mathrm{D}}}$ and $\rho_{U_{2\mathrm{D}}}$. Also, among the class of ρ_U wavelets, $\rho_{U_{1\mathrm{D}}}$ outperforms $\rho_{U_{2\mathrm{D}}}$. A possible interpretation is that a 1-dimensional design is more suitable for steerable wavelets since they are inherently directional. In this application, wavelets align themselves with ridges so that the $U_{1\mathrm{D}}$ criterion, which measures the spread in each radial direction, is the most appropriate one.

Image Reconstruction from Edges

Our second experiment deals with image reconstruction from a reduced set of wavelet-based edges. First, a multiscale *primal sketch* [134], or *edge map* [117], is extracted from the set of wavelet coefficients of the image. An approximation of

Table 4.2: Estimation of local orientations for the regions of interest depicted in Figure 4.10b

Wavelet type	Angle [°]					
	1	2	3	4	5	6
Ground Truth	*95.36*	*129.09*	*83.88*	*166.50*	*33.02*	*13.24*
$\rho_{V_{2\mathrm{D}}}$ (4.44)	95.88	133.19	80.48	164.66	36.74	15.95
$\rho_{V_{1\mathrm{D}}}$ (4.46)	96.08	133.59	79.14	163.13	37.44	15.90
$\rho_{U_{2\mathrm{D}}}$ (Simonceli) (4.45)	96.42	133.75	78.37	164.55	37.72	16.67
$\rho_{U_{1\mathrm{D}}}$ (4.47)	**95.57**	**132.88**	**83.19**	**165.07**	**36.12**	**15.53**
Papadakis	97.31	134.91	79.46	163.39	41.65	18.04
Meyer	97.15	134.67	79.65	164.25	41.61	18.75
Shannon	96.78	136.61	76.97	136.10	70.73	22.16

Table 4.3: Error in local orientation estimates for the regions of interest depicted in Figure 4.10b

Wavelet type	Absolute error [°]					
	1	2	3	4	5	6
$\rho_{V_{2\mathrm{D}}}$ (4.44)	0.52	4.10	3.41	1.85	3.72	2.71
$\rho_{V_{1\mathrm{D}}}$ (4.46)	0.73	4.49	4.75	3.38	4.41	2.66
$\rho_{U_{2\mathrm{D}}}$ (Simonceli) (4.45)	1.07	4.66	5.52	1.95	4.69	3.43
$\rho_{U_{1\mathrm{D}}}$ (4.47)	**0.22**	**3.79**	**0.69**	**1.43**	**3.09**	**2.29**
Papadakis	1.95	5.82	4.42	3.11	8.62	4.80
Meyer	1.79	5.57	4.24	2.26	8.59	5.51
Shannon	1.43	7.51	6.91	30.41	37.71	8.92

the original image is then recovered from this small subset of coefficients, relying on constrained optimization.

To extract a multiscale edge map from the input image, we rely on a gradient-like wavelet framework. It is implemented using the generalized Riesz-wavelet transform with the appropriate shaping matrix (4.22) and yields a pair of x- and y-derivative wavelets as described in Section 4.1. Edges in the multiscale gradient signal are then detected based on a wavelet-domain version of the Canny edge detector, which includes non-maximum suppression and hysteresis thesholding [134]. Note that the Canny edge detector requires an estimation of the magnitude and orientation of the gradient at each point in the image, which is obtained by steering the gradient-like wavelets at every scale and location. The final edge map is composed of the wavelet coefficients retained by the multiscale edge detector. To preserve the graylevel information of the image, all coefficients of the lowpass residual of the wavelet decomposition are kept.

Reconstruction is then formulated as the constrained optimization problem

$$\text{minimize } \|\mathbf{z}\|_{\ell_1} \tag{4.48}$$

$$\text{subject to } \mathbf{z} = \mathbf{W}^H\tilde{\mathbf{f}} \tag{4.49}$$

$$z_\kappa = q_\kappa, \forall\, \kappa \in S, \tag{4.50}$$

where $\mathbf{W}^H$ is the wavelet analysis operator, $\tilde{\mathbf{f}}$ and $\mathbf{z}$ are the variables being optimized (the reconstructed image and its wavelet transform, respectively, represented as matrices), and S the set of indices of wavelet coefficients that are part of the edge map. Finally, q_κ denotes the wavelet coefficient of the original image $\mathbf{f}$ at location κ, where κ is an index of 2D position and scale (*i.e.*, a compact notation for the indices i, $\mathbf{k}$ we used earlier). The reconstructed image possesses the same coefficient as the original one for those that are part of the edge map, and the remaining ones are filled during the optimization. This formulation is motivated by two main principles. First, we aim at conserving the elements of the edge map in order to reconstruct the image. We refer to them as wavelet-based edges, as they are the output of a Canny edge detector applied on the wavelet transform of the image. This gives us constraint (4.50), which imposes the elements q_κ, $\kappa \in S$ to be fixed during the optimization process. Second, we want the estimated wavelet coefficients to project back onto an image. Knowing that images are sparse in the wavelet domain, we impose sparsity by minimizing the ℓ_1-norm of the wavelet transform $\mathbf{z}$ of the image, yielding (4.48). Our problem thus amounts to minimizing a convex functional under a finite set of linear constraints, which guarantees the existence of a feasible minimum that can be reached using appropriate optimization algorithms.

Hereafter, we propose an improvement of the reconstruction algorithm introduced in [117] that relies on a gradient descent of the augmented Lagrangian. Our algorithm is based on the alternating-direction method of multipliers (ADMM), which motivates the introduction of the auxiliary variable $\mathbf{z}$. ADMM is known to converge very fast to an acceptable solution and guarantees the residual to be brought to zero. In practice, it is observed that the fast and moderately good estimate provided after 30 iterations of the algorithm is already visually satisfactory. We refer the reader to [135] for a complete description of the method.

To reconstruct the image with ADMM, we first form the augmented Lagrangian

$$\mathcal{L}(\mathbf{z},\tilde{\mathbf{f}},\boldsymbol{\lambda}) = \|\mathbf{z}\|_{\ell_1} + \boldsymbol{\lambda}^T\left(\mathbf{z}-\mathbf{W}^H\tilde{\mathbf{f}}\right) + \frac{\sigma}{2}\left\|\mathbf{z}-\mathbf{W}^H\tilde{\mathbf{f}}\right\|_{\ell_2}^2, \tag{4.51}$$

where σ is a step size that can be adapted to influence the speed of convergence. We rewrite (4.51) in terms of the scaled dual variable $\mathbf{u} = \frac{\boldsymbol{\lambda}}{\sigma}$ in order to obtain simpler mathematical expressions, yielding

$$\mathcal{L}(\mathbf{z},\tilde{\mathbf{f}},\mathbf{u}) = \|\mathbf{z}\|_{\ell_1} + \frac{\sigma}{2}\left\|\mathbf{z}-\mathbf{W}^H\tilde{\mathbf{f}}+\mathbf{u}\right\|_{\ell_2}^2 - \frac{\sigma}{2}\|\mathbf{u}\|_{\ell_2}^2. \tag{4.52}$$

The ADMM algorithm for our problem thus consists of the three successive iterations

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z},\tilde{\mathbf{f}}^{(k)},\mathbf{u}^{(k)}), \tag{4.53}$$

$$\tilde{\mathbf{f}}^{(k+1)} = \arg\min_{\tilde{\mathbf{f}}} \mathcal{L}(\mathbf{z}^{(k+1)},\tilde{\mathbf{f}},\mathbf{u}^{(k)}), \tag{4.54}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \left(\mathbf{z}^{(k+1)}-\mathbf{W}^H\tilde{\mathbf{f}}^{(k+1)}\right). \tag{4.55}$$

The update of $\mathbf{z}$ can be rewritten in an element-wise manner as

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}}\left(\|\mathbf{z}\|_{\ell_1} + \frac{\gamma}{2}\left\|\mathbf{z}-\mathbf{W}^H\tilde{\mathbf{f}}^{(k)}+\mathbf{u}^{(k)}\right\|_{\ell_2}^2\right) \tag{4.56}$$

$$= \arg\min_{\mathbf{z}}\left(\sum_{\kappa\in I}|z_\kappa| + \frac{\gamma}{2}\sum_{\kappa\in I}\left|z_\kappa - v_\kappa^{(k)}\right|^2\right), \tag{4.57}$$

where we denote by I the set of indices of all wavelet coefficients in the reconstructed image (thus, $S \subset I$). We introduced $v_\kappa^{(k)} = [\mathbf{W}^H\tilde{\mathbf{f}}^{(k)}]_\kappa - u_\kappa^{(k)}$ for convenience. For all elements $\kappa \in S$, (4.50) imposes that $z_\kappa = q_\kappa$, and no further computations are required. For $\kappa \notin S$, z_κ and v_κ should be colinear in order to turn the second term of (4.57) into a constant. Plugging $z_\kappa = Cv_\kappa$ into (4.57) and solving for the optimal constant C brings us to the component-wise expression of the $\mathbf{z}$ update

$$z_\kappa^{(k+1)} = \begin{cases} q_\kappa, & \text{if } \kappa \in S \\ \left(1 - \frac{1}{\gamma\left|v_\kappa^{(k)}\right|}\right)_+ v_\kappa^{(k)}, & \text{if } \kappa \notin S, \end{cases} \tag{4.58}$$

where $(\cdot)_+ = \max(0, \cdot)$ corresponds to the shrinkage operation. Then, updating $\tilde{\mathbf{f}}$ boils down to an unconstrained quadratic optimization problem. It can hence be performed by taking the partial derivative of the augmented Lagrangian and solving for zero. This yields

$$\begin{aligned} \tilde{\mathbf{f}}^{(k+1)} &= \arg\min_{\tilde{\mathbf{f}}} \left\|\mathbf{z} - \mathbf{W}^H\tilde{\mathbf{f}} + \mathbf{u}\right\|_{\ell_2}^2 & (4.59) \\ &= \left(\mathbf{W}\mathbf{W}^H\right)^{-1}\mathbf{W}\left(\mathbf{z}^{(k+1)} + \mathbf{u}^{(k)}\right). & (4.60) \end{aligned}$$

As $\mathbf{W}$ forms a tight frame, (4.60) can be further simplified by observing that $\mathbf{W}\mathbf{W}^H = \mathbf{I}$. Finally, $\mathbf{u}$ is simply modified through a linear update.

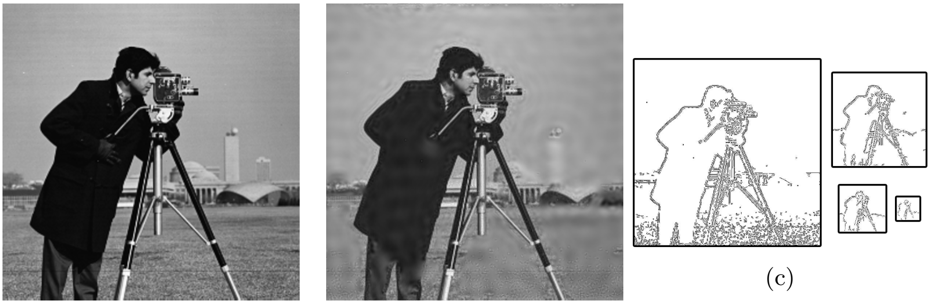
In practice, the algorithm is initialized with $\mathbf{z}^{(0)}$ composed of all the wavelet coefficients q_κ, $\kappa \in S$ retained in the edge map and the complete lowpass residual of the image to reconstruct. Then, $\tilde{\mathbf{f}}^{(0)}$ is initialized as an image entirely composed of pixels of zero value, and $\mathbf{W}^H\tilde{\mathbf{f}}^{(0)}$ is its wavelet transform. Finally, $\mathbf{u}^{(0)}$ is set as a pyramid of images composed only of zeros and having the same number of scales and dimensions as $\mathbf{W}^H\tilde{\mathbf{f}}^{(0)}$. The parameter σ is empirically set to 10^6. With these settings, from 30 to 50 ADMM iterations were observed to be sufficient to reach "visual convergence", which corresponds to a situation where additional iterations bring unnoticeable visual improvements. Note that, through the optimization process, the left multiplication by $\mathbf{W}^H$ or $\mathbf{W}$ amounts to performing wavelet analysis or synthesis, respectively. The algorithm can be executed in reasonable time as these two operations are performed efficiently with the help of a filterbank implementation. To illustrate this experiment, we give in Figure 4.11 the original Cameraman image, its reconstruction from 7% of the wavelet coefficients, as well as the mask containing the coefficients retained by our multiscale wavelet-based edge detector[6].

We gather results on a set of 7 standard test images, namely, House, Pirate, Peppers, Lena, Bridge, Cameraman, and Einstein, all being 512 × 512 pixels grayscale images. We run the same experiments with the trivial isotropic profile (Shannon [116]), several popular isotropic wavelet profiles (Simoncelli [132], which happens to be $\rho_{U_{\mathrm{2D}}}$ (4.45), Papadakis [124], and Meyer [123]), and finally with our wavelets ($\rho_{U_{\mathrm{1D}}}$ (4.47), $\rho_{V_{\mathrm{2D}}}$ (4.44) and $\rho_{V_{\mathrm{1D}}}$ (4.46)). We investigate the reconstruction performance of the different wavelets in terms of the PSNR of the reconstructed image. We start by retaining 7% of the total number of wavelet coefficients in the image. These 7% are selected from the set of wavelet-based edges identified by the multi-scale Canny edge detector[7]. Note that, as the test images all have the same size, the 7% threshold corresponds to the same absolute number of coefficients in each case. All experiments are conducted using 4 scales of decomposition. Reconstruction results obtained after 30 iterations of the ADMM algorithm are shown in Table 4.4. In order to allow for a visual comparison of the performance, we also show in Figure 4.12 close-ups of Lena reconstructed using the different wavelet profiles. We observe that $\rho_{U_{\mathrm{1D}}}$ outperforms the other wavelets. Further experiments of reconstruction using a set of edge coefficients with 1 to 7% of the total number of wavelet coefficients in the image allows us to reach similar conclusions, as presented in [113]. This confirms that the proposed $\rho_{U_{\mathrm{1D}}}$ profile is better for reconstruction than the other wavelets considered in this experiment. Comparing Tables 4.1 and 4.4, one can actually notice an interesting correlation between the average PSNR and the localization of the wavelet according to criterion U_{1D}.

Notice that, in this application, $\rho_{U_{\mathrm{1D}}}$ followed by $\rho_{U_{\mathrm{2D}}}$ outperforms in particular $\rho_{V_{\mathrm{2D}}}$ and $\rho_{V_{\mathrm{1D}}}$. The construction of the edge map relies on the same kind of framework as the filaments detection for local orientation estimation (namely, multiscale steerable filters). The same arguments therefore hold for explaining these results. The reconstruction task obtains better results when the edge map contains sharper elements, which yield better sets of edges. What matters most is therefore

[6] We implemented the Riesz-wavelet transform [120] as well as the edge-map extraction and subsequent image reconstruction using ADMM as a Java-based plug-in for the open-source image-analysis software ImageJ [9].

[7] In practice, we can adapt the number of coefficients identified by the multiscale Canny edge detector by changing the hysteresis thresholding parameters to make sure that the set contains enough elements .



(a)

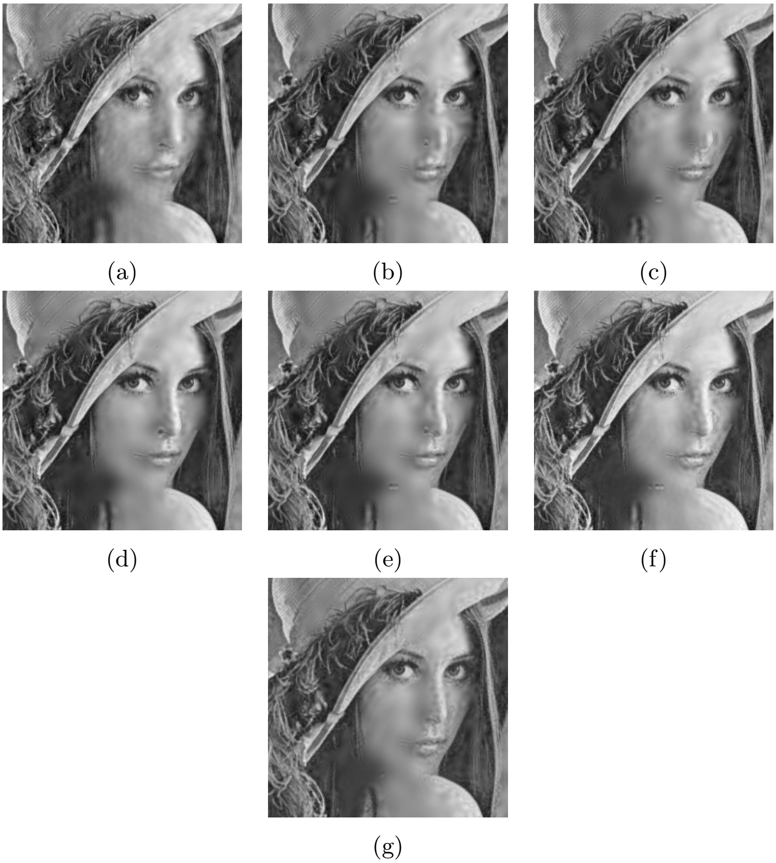
(b)

(c)

Figure 4.11: Wavelet-based edge reconstruction. (a) Original Cameraman image, (b) final result after reconstruction using $\rho_{U_{1\mathrm{D}}}$, and (c) binary masks featuring the wavelet coefficients saved for reconstruction at different scales. Here, 4 scales were used and 7% of the total number of coefficients was retained.

Table 4.4: Reconstruction from wavelet-based edges (7% of coefficients)

Wavelet type	PSNR [dB]						
	House	Pirate	Peppers	Lena	Bridge	Cameraman	Einstein
$\rho_{V_{2\mathrm{D}}}$ (4.44)	28.90	27.33	27.61	29.14	23.42	29.01	28.97
$\rho_{V_{1\mathrm{D}}}$ (4.46)	29.72	27.32	27.96	30.04	23.99	29.70	29.28
$\rho_{U_{2\mathrm{D}}}$ (4.45) (Simoncelli)	29.15	27.37	27.96	29.95	23.91	30.14	29.22
$\rho_{U_{1\mathrm{D}}}$ (4.47)	**30.01**	**28.24**	**28.42**	**30.37**	**24.07**	**30.67**	**29.40**
Papadakis [124]	27.98	26.55	27.34	28.99	23.67	28.52	28.89
Meyer [123]	27.72	26.29	26.70	28.88	23.50	28.97	28.65
Shannon [116]	26.80	24.91	25.53	27.20	23.03	25.60	27.61



(a) (b) (c)

(d) (e) (f)

(g)

Figure 4.12: Wavelet-based edge reconstruction. Close-up of the reconstruction of Lena relying on (a) Shannon, (b) Meyer, (c) Papadakis, (d) $\rho_{U_{\mathrm{1D}}}$, (e) $\rho_{U_{\mathrm{2D}}}$ (Simoncelli), (f) $\rho_{V_{\mathrm{1D}}}$, and (g) $\rho_{V_{\mathrm{2D}}}$. The best PSNR is achieved by $\rho_{U_{\mathrm{1D}}}$ shown in (d). Here, 4 scales were used and 7% of the total number of coefficients were retained.

again the optimal localization of the wavelet coefficients. This explains why best performances are obtained with profiles optimizing criterion of the class U ($\rho_{U_{1\mathrm{D}}}$ and $\rho_{U_{2\mathrm{D}}}$). Results can actually directly be interpreted from the values of $U_{1\mathrm{D}}$ provided in Table 4.1. Starting from the optimal $\rho_{U_{1\mathrm{D}}}$, the next most localized profiles sorted by distance to the optimum are Simoncelli ($\rho_{U_{2\mathrm{D}}}$), $\rho_{V_{1\mathrm{D}}}$, $\rho_{V_{2\mathrm{D}}}$, Papadakis and Meyer. The quality of reconstruction obtained with the different profiles follows the same pattern, with Papadakis and Meyer being the worst and followed by $\rho_{V_{2\mathrm{D}}}$ and $\rho_{V_{1\mathrm{D}}}$, $\rho_{U_{2\mathrm{D}}}$ (Simoncelli), and finally $\rho_{U_{1\mathrm{D}}}$, which achieves the best results.

In [113], we provide as additional example an experiment of image denoising using the BLS-GSM algorithm. There, we observe that $\rho_{V_{1\mathrm{D}}}$, followed by $\rho_{V_{2\mathrm{D}}}$, yields consistently better results than all the other profiles we tested, outperforming state-of-the-art results using the steerable pyramid. The most desirable feature in denoising, in comparison with local orientation estimation and image reconstruction from edges, is a spatially localized profile for the steerable pyramid. A transformation yielding very localized wavelet coefficients is actually even counterproductive as it concentrates the neighborhood of each coefficient, and thus reduces the amount of information that can be exploited by BLS-GSM. The profiles of choice for this application are hence of the class V. These different use-cases show that both classes of localization criterion are relevant depending on the kind of application being considered, and that the proposed wavelet profiles are interesting candidates for image processing tasks involving isotropic wavelets. The good or bad performance of the studied wavelet profiles can indeed be interpreted in the light of the localization metrics we introduced. This further hints at the fact that localization criterion are useful to study the localization of wavelets either in terms of their spatial profiles, or of the coefficients they generate. The two proposed localization criteria can thus serve as quick estimates to assess the relative performance of an isotropic tight-frame wavelet profile based on a simple calculation.

4.4 Learning Custom Feature Detectors

In Sections 4.2 and 4.3, we were studying the design of steerable filters optimizing properties such as the variance of their radial profile, or the shape of their angular profile. A further step is to formulate a procedure allowing to construct pattern-specific feature detectors from a given template. Such detectors can then be used to do template-matching-like analysis of images, while benefiting from all the advantages of the steerable filters framework. As research in this direction is ongoing [136], we here limit ourselves to providing a brief overview of the problem.

4.4.1 Designing Template-Specific Profiles

The template matching problem can be formulated as follows. A two-dimensional feature template h is swept over a two-dimensional input image f. At each location $\mathbf{x}$ in the image, we consider the normalized correlation between the image and the rotated template, expressed as

$$\mu_{\theta_0}(\mathbf{x}) = \frac{(f \, * \, h(-\mathbf{R}_{\theta_0}\cdot))\,(\mathbf{x})}{\|f\|_{L_2}\,\|h\|_{L_2}}, \tag{4.61}$$

where $\mathbf{R}_\theta$ denotes a rotation matrix and $*$ the convolution operation. The function μ_{θ_0} reaches maximum values at locations in the image that resemble the template rotated by an angle of θ_0. From there, the amplitude and orientation responses are obtained as

$$\mu(\mathbf{x}) = \max_{\theta_0} \mu_{\theta_0}(\mathbf{x}), \tag{4.62}$$

$$\theta_0^*(\mathbf{x}) = \arg\max_{\theta_0} \mu_{\theta_0}(\mathbf{x}), \tag{4.63}$$

respectively. Additionally, a pyramid can be constructed to add a scale component to the template matching procedure. To do so, the template is subjected to different amount of upsampling and downsampling, and the matching procedure is repeated such that an amplitude map μ_s and orientation map $\theta_{0,s}^*$ are obtained for each template scale s. The scale response map ς is obtained by identifying, for each location in the image, the scale where the largest amplitude response is found. This amounts to performing a maximum intensity projection of the amplitude response values across the scales of the pyramid. The final amplitude and orientation maps are constructed by reporting, at each location, the amplitude and corresponding orientation at the scale which responds the best (*i.e.*, which has the maximum value across the pyramid at that location). Formally, the multiscale version of template matching is thus expressed as

$$\mu(\mathbf{x}) = \max_{\theta_0, s} \mu_{\theta_0, s}(\mathbf{x}), \tag{4.64}$$

$$(\theta_0^*(\mathbf{x}), \varsigma(\mathbf{x})) = \arg\max_{\theta_0, s} \mu_{\theta_0,s}(\mathbf{x}), \tag{4.65}$$

with

$$\mu_{\theta_0,s}(\mathbf{x}) = \frac{\left(f \ast h_{\mathbf{R}_{\theta_0},s}(-\cdot)\right)(\mathbf{x})}{\|f\|_{L_2}\,\|h\|_{L_2}}, \tag{4.66}$$

where $h_{\mathbf{R}_{\theta_0},s}$ is the template h rotated by an angle of θ_0 and resized to correspond to scale s[8].

Template matching is unfortunately computationally heavy. One inner product between the template and the image must be computed for each particular value of location, scale, and angle. In practice, (4.61) is computed for a collection of θ_0 that uniformly discretizes the $[0, 2\pi]$ interval. Since the detection quality is strongly dependent on the fineness of the discretization angle, relying on a too coarse discretization step is not favorable. Detection with steerable filters is performed in a similar fashion, as described in Section 4.1. The main difference lies in the fact that the steerability property implies that any μ_{θ_0} can be computed as a linear combination of the basis filters. As a consequence, the computational cost for every rotation and scale is only a function of the number of considered basis filters. Considering a template matching procedure relying on an angle discretization of $\frac{\pi}{5}$ (*i.e.*, 10 angles are used to cover the $[0, 2\pi]$ interval), 10 inner products are required for a single scale. To improve the quality of the amplitude and orientation response maps, the discretization step must be reduced to finely sample the $[0, 2\pi]$ interval, thereby increasing the number of required computations. A steerable filter built from 10 basis functions is in contrast able to represent a continuum of rotation angles at the single cost of 10 inner products. To obtain the amplitude and orientation response maps at a given location in the image, the relevant quantities are therefore the number of (discrete) angles in the case of template matching, and the number of basis functions in the case of steerable filters. The advantage of each method can then be stated as follows. Template matching requires the computation of many inner products but, once done, returns directly the maximal amplitude and corresponding orientation. Steerable filters involve fewer inner product operations but, once these are computed, require to solve a minimization problem in θ_0 to recover the optimal amplitude and orientation.

[8] The actual resizing factor depends on the way the pyramid is built. Powers of two are often used, implying a rescaling factor of 2^s at scale s.

Our design goal is the following. Given an external feature template, one searches for the optimal radial profile for a given steerable filter framework. The angular profile of the filters are given by a set of harmonics of the Riesz transform. Considering the predefined finite set of harmonics S, the task translates to identifying the collection of optimal $\rho_{n,\mathrm{opt}}(\omega)$ such that the detector

$$\widehat{h}(\omega,\vartheta)=\sum_{n\in S}\rho_{n,\mathrm{opt}}(\omega)\mathrm{e}^{\mathrm{j}n\vartheta} \tag{4.67}$$

responds to (possibly rotated[9]) occurrences of the template in the input image. The work we presented in Sections 4.2 and 4.3 involved optimizing either the set of harmonics defining the angular part for a fixed radial one, or the common radial part for a fixed set of harmonics defining the angular one. Filters designed following (4.67) have in constrast a dedicated radial profile for each harmonic. This formulation is very rich: it makes it possible to go beyond polar separable filters and to represent any template in $L_2(\mathbb{R})$ by increasing the number of considered harmonics. The optimal radial profiles are obtained by identifying the detector $\widehat{h}$ which, when normalized, maximizes its inner product with the template. Several methods are under development in order to achieve this maximization. The most promising one consists in considering the spline expansion of the optimal radial profile. Additional refinements can be incorporated. For instance, the optimization problem can be reformulated to find the filter that responds the best to the template while responding the least to background. Although initially designed for templates featuring centered patterns surrounded by background, the approach can technically be used for any type of template under mild conditions. The number of considered harmonics is the main parameter. As detectors built from non-isotropic templates require in theory an infinite number of harmonics, the size of the set S is a function of the trade-off between the quality of the detector and the computational cost.

This procedure allows easily building custom feature detectors that yield amplitude response and orientation maps. For every pixel in the input image, one obtains a measure of how much the feature of interest is represented and how it is oriented at that location. It opens the way to more specific and refined feature or keypoint detectors built from steerable filters.

[9] The multiscale aspect is currently not covered by this theory.

Chapter 5

Theorizing Landmark Active Contours

In this chapter[1], we introduce the *landmark snake*, a novel active contour algorithm that relies on Hermite spline interpolation (Chapter 3) to perform segmentation relying on information extracted from local feature detectors (Chapter 4) and contours in the context of bioimage analysis. We present a general formulation of the landmark snake, from which a particular case (the open curve) has been briefly introduced in our previous work [138]. Our landmark snake model is reminiscent of B-spline based active contours [44, 65, 70] but benefits from direct control over tangents on the curve, which allows the locally adapt the curve according to keypoint descriptor-like information. To the best of our knowledge, this is the first time that a segmentation method bringing together spline-snakes and feature detectors is proposed. The essential ingredients of the snake model are the two complementary cubic Hermite spline basis functions that grant control on the curve and on its tangent field. This representation offers two interesting capabilities. First, modification of tangent magnitude allows generating local features such as sharp corners, or tips, which are harder to obtain with classical spline snakes. Second, controlling tangent orientation enables the design of energies enforcing that the snake locally

[1] This chapter is composed of a mixture of published material and manuscripts under preparation. Published results are adapted from our two papers [76] and [137].

adapts to the nature and orientation of features in the image.

The construction of (open and closed) snake curves that are able to generate corners and the definition of energy functionals based on local orientation information has also been studied by R. Kimmel *et al.* [59] with geodesic active contours. The construction of the model, the formulation of the energy, the associated optimization procedure and, ultimately, the capabilities and strengths of the resulting algorithms are however different from ours. Geodesic snake curves as proposed in [59] are defined implicitly. They allow for topological changes but have limited potential for manual interactions. By contrast, our landmark snakes are defined explicitly by construction. They allow for extensive manipulation through a general user interface, although the segmentation process remains mostly automated. The anchor points and tangent vectors can be manually tweaked in a user-friendly and precise manner. Such edit capabilities are of particular interest for complex image analysis problems where feedback from the user might be required. Our contribution is, firstly, to propose a new snake algorithm with very particular properties in terms of model and energies, differentiating it from existing active contours, and secondly to characterize the model and energies in a comprehensive manner through practical experiments so as to highlight their practical scope[2].

In the following, we denote two-dimensional vectors as

$$\mathbf{v} = (v_1, v_2) = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

We denote the standard vector scalar product and associated Euclidean norm as $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. As in previous chapters, inner products and norms on function spaces are denoted in a similar way, with the relevant function space indicated as a subscript.

5.1 Model

We now recall the general formulation of parametric snakes and specify our new snake model. We describe the active contour constructions (closed and open curves) obtained from our model.

[2] Open- and closed-snake variants have been implemented as plugins for the open-source image-analysis software ImageJ [9]. They are freely available online for use by the bioimage-analysis community at http://bigwww.epfl.ch/algorithms/hsnakes.

Let $\mathbf{r} = (r_1, r_2)$ be a two-dimensional curve in the image. It is described by the two Cartesian coordinate functions $r_1(t)$ and $r_2(t)$, with $t \in \mathbb{R}$ a continuous variable. The coordinate functions can be parameterized efficiently as the linear combination of integer shifts of basis functions weighted by a sequence of control points. In the case of spline snakes, bases are constructed as the integer shifts of a compactly supported spline generator ϕ. The sequence of distinct control points is denoted as $\{\mathbf{c}[k]\}_{k\in\mathbb{Z}}$ and the general expression of parametric spline snakes is therefore

$$\mathbf{r}(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \sum_{k\in\mathbb{Z}} \mathbf{c}[k]\phi(t-k). \tag{5.1}$$

For the interested reader, we elaborate on spline snake models in 2.3.1. In our case, the generators are the two cubic Hermite spline basis functions ϕ_1 and ϕ_2 discussed extensively in Chapter 3. The novel aspect of our construction comes from the fact that, given a curve, the function ϕ_1 interpolates point values on the curve, while the derivative of ϕ_2 interpolates tangents to the curve. Both ϕ_1 and ϕ_2 are compactly supported in $[-1, 1]$ and yield fast and stable interpolation methods (see [57] for more details). The analytic expressions of ϕ_1 and ϕ_2 as well as the mathematical formulation of their joint interpolation properties can be found in Section 3.1.

Our landmark snake model is particularly well adapted to both the open- and closed-curve scenarios. A closed curve can easily be constructed relying on periodized versions of the Hermite basis, in direct analogy with classical spline snakes [70]. The generation of open snakes from cubic B-splines, however, requires one to rely on virtual or invisible control points at the curve extremities. By contrast, cubic Hermite splines result in natural conditions at the ends of the landmark snake.

5.1.1 Open Landmark Snake

Parametric snake curves are continuously defined for $t \in \mathbb{R}$. However, for implementation convenience, we usually prefer to normalize them for $t \in [0, 1]$. To follow the convention used for classical spline-snakes, we denote the number of control points as M. Landmark snakes therefore have $2M$ parameters, since the local tangent can also be edited at every control point.

Open curves are specified by a sequence of M anchor points and their associated tangents vectors, defined respectively for $k = 0, ..., M-1$ as

$$\begin{array}{rcl} \mathbf{r}[k] & = & \mathbf{r}(t)\,|_{t=\frac{k}{M-1}}, \\ \mathbf{r}'[k] & = & \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}(t)\Big|_{t=\frac{k}{M-1}}. \end{array} \tag{5.2}$$

Due to the interpolation properties of the Hermite generators, control points and tangents directly correspond to the curve value at integer locations. The parametric representation of the open landmark snake is then given by

$$\mathbf{r}_{\text{open}}(t) = \sum_{k=0}^{M-1} \Big(\mathbf{r}[k]\,\phi_1((M-1)\,t-k) + \mathbf{r}'[k]\,\phi_2((M-1)\,t-k) \Big), \tag{5.3}$$

where $t \in [0,1]$. An example curve is depicted in Figure 5.1a.

5.1.2 Closed Landmark Snake

Closed curves are obtained by periodizing the control point sequence, as explained in 2.3.1. In such case, the active contour is entirely defined by its M-periodic sequence of anchor points $\{\mathbf{r}[k]\}_{k\in\mathbb{Z}}$ and tangent vectors $\{\mathbf{r}'[k]\}_{k\in\mathbb{Z}}$. In analogy to the open-curve case,

$$\begin{array}{rcl} \mathbf{r}[k] & = & \mathbf{r}[k+M] = \mathbf{r}(t)\,|_{t=\frac{k}{M}}, \\ \mathbf{r}'[k] & = & \mathbf{r}'[k+M] = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}(t)\Big|_{t=\frac{k}{M}}. \end{array} \tag{5.4}$$

The periodic nature of the control point sequences allows expressing the curve with periodized basis functions. We denote the M-periodized versions of the generators as

$$\phi_{1,\text{per}}(t) = \sum_{n=-\infty}^{\infty} \phi_1(t-Mn) \tag{5.5}$$

$$\phi_{2,\text{per}}(t) = \sum_{n=-\infty}^{\infty} \phi_2(t-Mn). \tag{5.6}$$

The continuously defined closed contour is 1-periodic from the normalization to $t \in [0,1]$. It is given by

$$\mathbf{r}_{\text{closed}}(t) = \sum_{k=0}^{M-1} \Big(\mathbf{r}[k]\, \phi_{1,\text{per}}(Mt-k) + \mathbf{r}'[k]\, \phi_{2,\text{per}}(Mt-k) \Big), \tag{5.7}$$

where $t \in [0,1]$ (see Figure 5.1b). Note that normalization is here achieved by multiplying t by M, whereas $M-1$ was used in (5.3). This is due to the fact that, when dealing with an open curve, considering M points only yields $M-1$ curve segments. In the closed curve case, a portion of curve connects the last control point to the first one, yielding M intervals.

The positive integer M, corresponding to the number of pairs of control points and tangent vectors, regulates the flexibility of the snake. Large values of M allow reproducing complex shapes, while small ones yield simple contours. It is possible to approximate any closed shape by considering an arbitrarily large amount of control points and associated tangent vectors.

Our model is an extension of cubic B-spline snakes. In the absence of ϕ_2, (5.3) and (5.7) indeed reduce to (5.1), the classical formulation of parametric active contours. The introduction of ϕ_2 allows controlling the tangents, which grants several features that classical spline snakes do not offer.

5.1.3 A Digression on the Generation of Corners

An important advantage of the landmark snake is its ability to reproduce corners with only one control point, something that is not achievable with classical spline snakes.

To better understand how the landmark snake generates corners, we start by considering what defines such features. A necessary condition for introducing a corner at curvilinear coordinate t_0 is to set the derivatives of the coordinate functions to

$$r_1'(t_0) = r_2'(t_0) = 0. \tag{5.8}$$

What we call corners are often referred to as *singular points* [139].

Provided that no two control points are identical, it can be shown that, in the cubic B-spline and exponential-spline snake cases, (5.8) can only be satisfied when

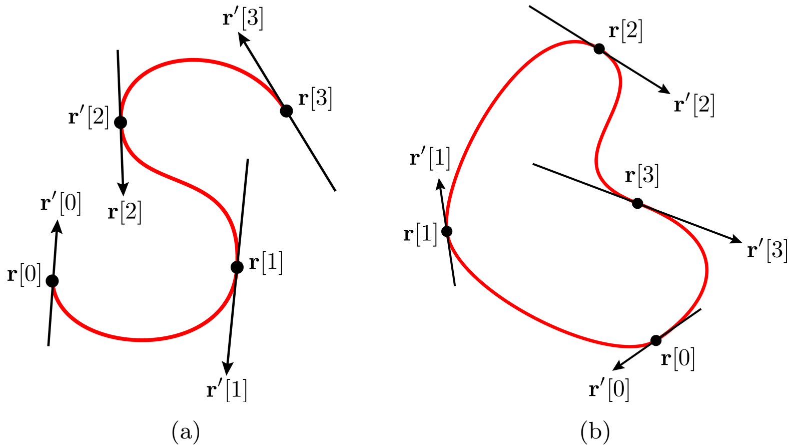


Figure 5.1: (a) Open and (b) closed landmark active-contour models. In both cases, the curve is represented by a set of control points and associated tangent vectors.

all control points lie on a straight line. We refer to this situation as a flat corner. It is obviously of limited practical interest. The landmark snake, however, allows enforcing (5.8) by setting the tangent vectors to zero at any arbitrary control point. When all tangent vectors in a landmark snake are set to zero (*i.e.*, $\mathbf{r}'[k] = 0$ for all k), it is not hard to show that $\frac{r_2(t)}{r_1(t)}$ is constant in $[k, k+1)$ for every k on the curve. This is equivalent to linear interpolation and control points are hence linked by straight lines. Non-null tangent vectors introduce smoothness in the curve at the corresponding control point location, as shown in Figure 5.2. Conversely, in a curve composed of mostly non-zero tangent vectors, setting one of them to zero results in a local roundish corner as depicted in Figure 5.3.

As a more precise explanation, two situations allow creating a true, non-flat corner. The local slopes in each coordinate functions around the corner point can either be different or equal to zero. The first situation with unequal local slopes around the corner point implies that the second derivative—of either coordinate function—contains a discontinuity. Cubic B-splines and exponential splines are C^2-continuous by construction at the joining points and hence necessarily have continuous second derivatives. Hermite spline curves, on the contrary, are C^1 at

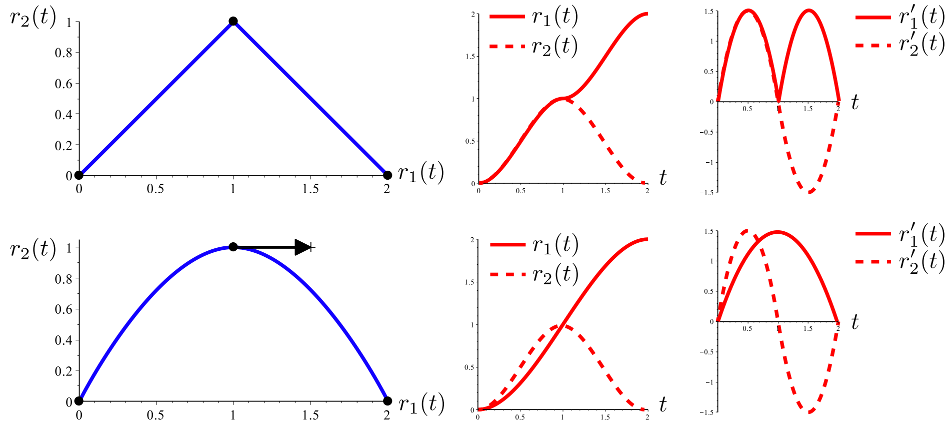


Figure 5.2: Generation of corners with the landmark snake. Two 2-D curves are depicted as solid lines along with a plot of their coordinate functions and their derivatives. The curves are parameterized by three control points (solid dots) and associated tangent vectors (arrows, set to vanish at $\mathbf{r}(0)$ and $\mathbf{r}(2)$). The control points of the two curves are identical, except for the tangent vector at $\mathbf{r}(1)$. When set to zero, it creates a discontinuity in r_1'' and yields a sharp corner. When it doesn not vanish, the discontinuity disappears and a smooth curve is obtained.

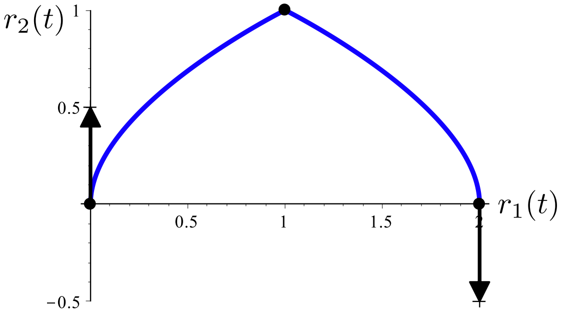


Figure 5.3: Generation of roundish corners with the landmark snake. Setting $r_1'(1) = r_2'(1) = 0$ creates a corner. Although the surrounding control points have non-zero derivatives (represented by the tangent vectors as arrows), the local behavior around the corner is conserved and yields a sharp discontinuity at $\mathbf{r}(1)$.

the joining points and therefore allow for discontinuous second derivatives. It is still possible to generate corners with cubic or exponential splines through the second scenario, where slopes around the corner point are vanishing. However, it implies that several control points accumulate at the same location on the 2D plane. This corresponds to the introduction of multiple knots in the spline curve, which does not fit the spline-snake formalism relying on sequences of distinct control points[3].

5.2 Energies

The evolution of active contours is driven by the optimization of a cost functional referred to as snake energy. Many energy terms have been proposed in the literature [65, 140], we point the reader to Section 2.3.2 for some examples. The specificity of the image analysis problem at hand usually drives the design of the energy function, as the energy dictates the quality of the final segmentation result.

While many approaches to detect feature points exist in the computer vision community (see Chapter 2 for more details), segmentation approaches relying on the

[3] The curve $\mathbf{r}(t) = (t^3, t^2)$ is an example of this situation, with a corner at $\mathbf{r}(0)$.

automated detection of landmarks are more rarely found in bioimage informatics. There, most methods rely on points that are defined manually [65]. The landmark snake is versatile enough to handle all traditional energy functionals designed for classical parametric and point-based snakes. However, in addition to that, the direct expression of tangents in the snake model enables the design of feature-based energies. As a consequence, constraints based on the output of feature detectors can be imposed locally on the tangent field of the contour. It allows linking automatic feature detection and active-contour methods.

5.2.1 Internal Energies

Cubic Hermite splines are optimally smooth in the sense that, for a function $f \in L_2(\mathbb{R})$, they minimize $\|f''\|_{L_2}$, as presented in more details in 3.3.1. This quantity is linked to the snake curvature [65]. It therefore holds the potential to eliminate the need for an explicit internal energy term. If required, low curvature can be guaranteed under some mild conditions on the parametric curve in a similar fashion as for cubic B-spline snakes [65].

5.2.2 Image Energies

Let us start by recalling that energies are functions of the snake curve that get minimized during the optimization process. Following the notations introduced in 2.3.1, we consider snake curves $\mathcal{C}(\Theta)$, where Θ is the snake representation. In our case, the snake is completely determined by $\Theta = \{\mathbf{r}[k], \mathbf{r}'[k]\}_{k=0,\dots,M-1}$ through $\mathbf{r}$ and is normalized such that $t \in [0,1]$. Thus, $\mathcal{C}$ is parameterized by $\mathbf{r} : [0,1] \to \mathcal{C}$. Optimizing an energy term amounts to adapting the values of the $\{\mathbf{r}[k], \mathbf{r}'[k]\}_{k=0,\dots,M-1}$ such that $E(\Theta)$ reaches a minimum value.

We propose three types of energies which directly use the control over tangents granted by the landmark snake. The underlying idea is to use feature detectors to obtain maps of possible landmarks in the image. These correspond to places where something particular happens. From the interpolation property of Hermite splines, the control points of the landmark snake lie directly on the curve. They can thus be used as a one-to-one correspondence with landmarks. Our feature-based energies then deforms the snake curve to attract control points to landmark regions. As feature detectors provide information on the nature of the local neighborhood of each feature, the energy then translates this information into constraints on the tangent vector so as to locally adapt the snake curve properties. Each tangent vector $\mathbf{r}'[k]$ has three different aspects that can be constrained, namely:

- its origin, which corresponds to the location of the associated control point $\mathbf{r}[k]$,
- its magnitude, which corresponds to $\|\mathbf{r}'[k]\|^2$ and which dictates the local smoothness of the curve at the tangent location,
- its orientation, which corresponds to the angle between the vector $\mathbf{r}'[k]$ and the horizontal axis and which influences the local orientation of the curve.

Our three feature-based energies target each of these properties, together or separately. First, the purely directional energy term imposes only orientation constraints on the tangent from the output of general-purpose filters. Then, the purely amplitude-based energy term based on corner detectors focuses on tangent magnitude only. Finally, our last formulation simultaneously constrains both tangent magnitude and orientation using the information provided by the output of template-based filters.

Purely Directional Energy

We define the continuous *directional snake energy* as

$$E_{\mathrm{directional}}(\Theta) = -\frac{1}{L(\Theta)} \oint_{\mathcal{C}(\Theta)} \left| \left\langle \mathbf{v}_{\theta_0^*(\mathbf{r})}, \frac{\mathbf{r}'}{\|\mathbf{r}'\|} \right\rangle \right| \mu(\mathbf{r})\,\mathrm{d}r, \tag{5.9}$$

where the two-dimensional vector functions $\mathbf{r} = (r_1, r_2)$ and $\mathbf{r}' = (r_1', r_2')$ correspond to the snake curve and its derivative, respectively. The elements θ_0^* and μ are some orientation and amplitude maps computed from the input image and having the same dimensions as the latter. By $\mu(\mathbf{r})$, we designate the values of the map μ under the image coordinates of the curve $\mathbf{r}$. Similarly, we define the unit vectors $\mathbf{v}_{\theta_0^*(\mathbf{r})} = (\cos(\theta_0^*(\mathbf{r})), \sin(\theta_0^*(\mathbf{r})))$ representing the local orientations θ_0^* under the image coordinates of the curve $\mathbf{r}$. The absolute value of the inner product between $\mathbf{v}_{\theta_0^*(\mathbf{r})}$ and the coordinates of $\mathbf{r}'$, which is the first term of the integrand, is maximized when the tangents of the snake are locally aligned with the orientation given by θ_0^*. The term μ acts as a weight that favors where the feature of interest is locally present, indicated by a large magnitude. In order to obtain a dimensionless energy, the line integral is finally normalized by the length of the curve $L(\Theta) = \oint_{\mathcal{C}(\Theta)} \mathrm{d}r = \int_0^1 \|\mathbf{r}'(t)\|\mathrm{d}t$. The whole expression is set with a minus sign since the snake optimization is defined as a minimization process.

The directional energy (5.9) is general enough to accommodate any input-image-sized maps θ_0^* and μ providing orientation and amplitude information, respectively. We here give two different examples thereof.

1. For an input image denoted by I, the orientation and amplitude maps are given by the classical two-dimensional gradient orientation and magnitude

$$\theta_0^*(\mathbf{x}) = \arctan\left(\frac{\frac{\partial I(\mathbf{x})}{\partial y}}{\frac{\partial I(\mathbf{x})}{\partial x}}\right), \tag{5.10}$$

$$\mu(\mathbf{x}) = \sqrt{\left(\frac{\partial I(\mathbf{x})}{\partial x}\right)^2 + \left(\frac{\partial I(\mathbf{x})}{\partial y}\right)^2}. \tag{5.11}$$

   The resulting snake energy tends to deform the snake towards regions of high gradient amplitude (*i.e.*, edges) and to align its tangent vectors with the local orientation of the gradient. As θ_0^* defined as (5.10) tends to be noisy, this choice is not the most robust in practice.

2. Steerable filters offer more refined orientation and magnitude response maps. From Chapter 4, orientation and magnitude can be defined as (4.2) and (4.3), respectively. The detection template h is given by

$$h(\mathbf{x}) = \sum_{n=0}^{N}\sum_{i=0}^{n} a_{n,i} \frac{\partial^{n-i}}{\partial x^{n-i}} \frac{\partial^i}{\partial y^i}\, g(\mathbf{x}), \tag{5.12}$$

   where g is a Gaussian window and N is the order of the detector. The kind of features detected by h can be modified by acting on N. Odd values of N yield edge detectors while even values of N produce ridge detectors. In addition, filters built with larger values of N have a higher SNR and better localization capabilities, at the cost of increasing computational expenses [31]. The orientation information obtained with steerable filters is generally less subject to noise and therefore more precise than that obtained with the Canny edge and Hessian ridge detectors.

Proof of Concept. The purely directional snake energy (5.9) differs from classical image-based energies by the addition of the term that enforces $\mathbf{r}'$ to be in the orientation given by θ_0^*. Constraining the behavior of the derivative of the snake through the tangent vectors increases the overall robustness of the segmentation. In cases where several patterns of interest are in close contact or when dealing with object having complex shapes, magnitude information might indeed not be sufficient to properly segment the contour. We generated synthetic images that are representative of situations where the contour of the object of interest contains rapid changes of orientation or has circonvolutions, and where nearby objects are likely to attract the snake and divert it from its target. Using the same initial curve, we optimized a cubic B-spline snake, an exponential snake [70], and a landmark snake. We carried out all experiments using a steerable filter designed to detect ridges constructed following (5.12). In the Hermite case, we took advantage of (5.9). For the other snakes, we used only magnitude information as

$$E_{\text{simple}}(\Theta) = -\frac{1}{L(\Theta)} \oint_{\mathcal{C}(\Theta)} \mu(\mathbf{r})\,\mathrm{d}r. \tag{5.13}$$

Traditional spline snakes do not include tangents in their parameterization. To perform a fair comparison and be consistent in the number of parameters, we therefore endow such snakes with at most twice as many control points as the landmark snake. Results are shown in Figure 5.4 for the closed-curve case and in Figure 5.5 for the open-curve case.

Purely Amplitude-Based Energy

Energies attracting parametric snakes towards a set of user-defined points have been proposed [65], as well as methods driving contour extremities to automatically detected landmarks [141]. We aim at extending this idea by not only imposing that automatically detected landmarks attract the control points, but also that the nature of the landmark constrains the local behavior of the curve through the tangent vectors. We propose the *amplitude-based energy* that enables the generation of controlled discontinuities at precise locations on the curve as

$$E_{\text{amplitude}}(\Theta) = \sum_{i=1}^{N_c} \|\mathbf{r}[k_i^*] - \mathbf{c}_i\|^2 + M^2 \frac{\|\mathbf{r}'[k_i^*]\|^2}{\|\mathbf{r}[k_i^*] - \mathbf{c}_i\|^2}, \tag{5.14}$$

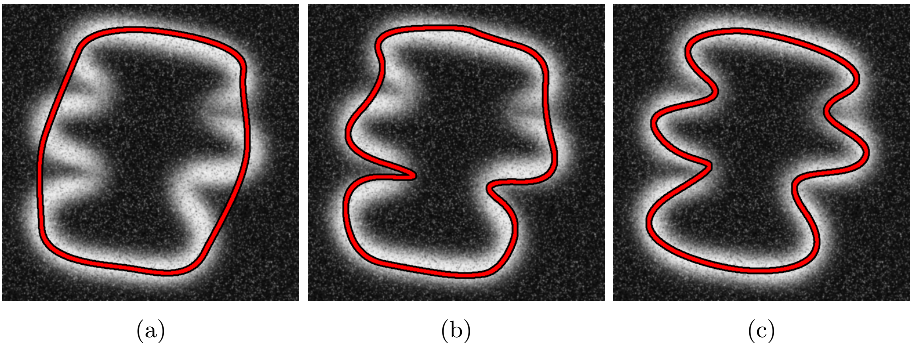


Figure 5.4: Segmentation with and without orientation information in the closed-curve case. (a) Snake initialization for (b) and (c). (b) Segmentation result after optimizing a 16-point exponential-spline snake with the ridge-based energy (5.13). (c) Segmentation result after optimizing an 8-point landmark snake with (5.9).

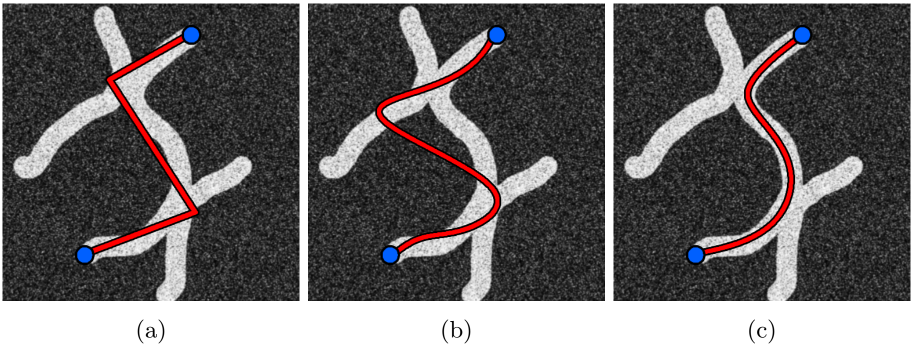


Figure 5.5: Segmentation with and without orientation information in the open-curve case. (a) Snake initialization for (b) and (c). (b) Segmentation result after optimizing an 8-point cubic-spline snake with the ridge-based energy (5.13). (c) Segmentation result after optimizing a 4-point landmark snake with (5.9). Fixed snake extremities are depicted as disks.

where

$$k_i^* = \underset{0 \leq k < M}{\operatorname{argmin}} \|\mathbf{r}[k] - \mathbf{c}_i\|^2.$$

The $\mathbf{r}[k]$ and $\mathbf{r}'[k]$ are the M control points and tangent vectors from (5.7), and $\mathbf{c}_i$ is the i-th element in the collection of N_c detected corner points.

The energy contribution of the corner point $\mathbf{c}_i$ is composed of two terms: the squared distance to the control point closest to the corner, and the length of the associated tangent vector divided by the aforementioned distance. In this way, the penalty imposed on the length of the tangent increases as the point approaches the corner location, forcing the curve to create a sharp corner. This result is weighted by M^2 to impose stronger constraints on the tangents for curves with more control points and, therefore, more degrees of freedom. In practice, an epsilon is added to the denominator to avoid numerical instabilities when control points get very close to or match corner locations.

Proof of Concept. We rely on a synthetic image to demonstrate the benefits of using the amplitude-based energy (5.14). We study the segmentation of a solid almond-shaped object. As depicted in Figure 5.6a, the pointy extremities can easily be detected, for instance by a Harris corner detector. This method identifies locations of sudden variation of the gradient in the image (see Section 2 for more details). More precisely, it relies on the eigenvalues (λ_1, λ_2) of the Hessian matrix to obtain a corner score R for each point in the image, defined as

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2).$$

The behavior of the detector can be tuned by acting on three main parameters. The first one is the so-called sensitivity parameter k, for which smaller values allow capturing sharper corners. The second is a minimal value for R, which sets a threshold on the accepted quality of corners. Finally, the support of the filter used to estimate the Hessian matrix can be tuned as well.

When the snake is not encouraged to place control points at the extremities of the object, it tries to generate a discontinuity by dramatically increasing the length of its tangents, resulting in unwanted loops (Figure 5.6c). Adding a landmark energy that does not involve tangents, such as the one proposed in [65], is still not sufficient (Figure 5.6d) as it fails to prevent the snake from diverging at

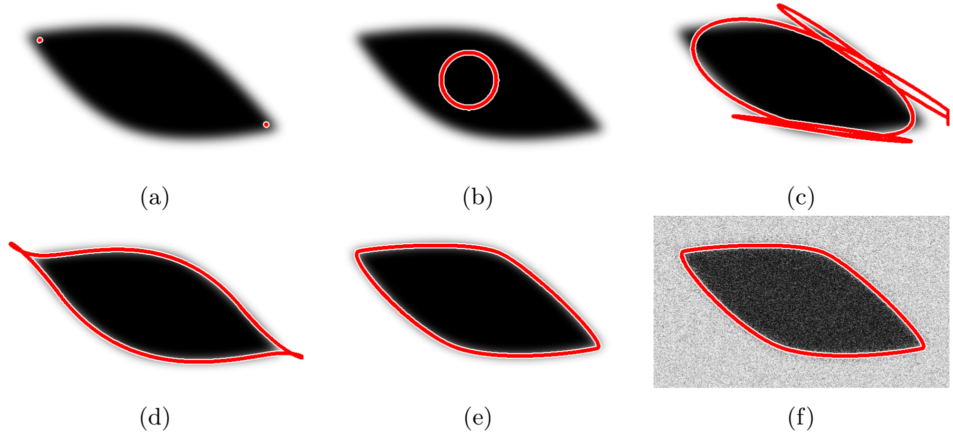


Figure 5.6: Examples of synthetic image. (a) Detected corner locations in red. (b) Initialization of the 4-points snake. Segmentation results after optimizing on (c) an edge-based energy only, (d) a combination of edge- and constraint-based [65] energies, (e) a combination of (5.14) and of an edge-based energy. (f) Results using a combination of (5.14) and of an edge-based energy on the image degraded by additive Gaussian noise.

the tips of the object. Attracting control points to corner locations and simultaneously encouraging small tangent values allows for a precise outline of the whole shape (Figure 5.6e). Moreover, both the corner detection method and the snake optimization are robust to noise (see Figure 5.6f). Results are shown for the same initial conditions depicted in Figure 5.6b in all cases.

Local Feature-Based Energy

The two energy terms we proposed so far were either purely directional or purely amplitude-based in the sense that we either imposed constraints only on the amplitude of the tangent or only on its direction. We now propose a local feature-based energy where both the magnitude and the direction of the tangent adapt to the nature of the features present in the image. The initial element we rely on is a template representing a type of features of interest. Let us consider a landmark snake curve with one control point that lies exactly on the feature (*i.e.*, in the center of the template). The way its tangent should be set in order to perfectly outline the feature locally is referred to as the *ideal tangent* and is denoted as $\boldsymbol{\tau}_f$. The template and ideal tangent generation can be replaced by a training step. This consists in manually annotating an instance of the object of interest with a landmark snake and flag the control points that should act as landmarks. Templates are then generated by extracting crops of the image around these control points, and ideal tangents are directly obtained as the tangent values at these locations.

In order to compute the local feature-based energy from this initial setting, an input image should be processed by a feature detector outputting three pieces of information, namely

1. where the feature occurs, or where it is found in the image;
2. at which scale the feature occurs;
3. how the feature is oriented.

These elements can come in two forms. The first one is a triplet of amplitude, orientation and scale response maps of the same dimensions than those of the input image, which contain at each location a measure of how strongly the feature is represented, how it is oriented and at which scale it occurs, respectively. For a feature f, the amplitude response map is referred to as $\mu_f(\mathbf{x})$. The orientation response map is denoted as $\theta^*_{0,f}(\mathbf{x})$ and the scale response map as $\varsigma_f(\mathbf{x})$. To build our local feature-based energy, we will rely on the image inverse of the amplitude response map, referred to as $\overline{\mu_f}(\mathbf{x})$. The image inverse corresponds to a reversed version of $\mu_f(\mathbf{x})$ and is similar to a photographic negative. As a consequence, areas of strong amplitude response correspond to peaks (*i.e.*, local maxima) in μ_f and pits (*i.e.*, local minima) in $\overline{\mu_f}$. The orientation and scale response maps can be used without any modification. The second form that feature detector outputs can take is a discrete list of coordinates indicating feature locations with a corresponding descriptor providing scale and orientation information, which is typical of keypoint detectors. As our energy is more easily expressed using input-image-sized maps, we propose the following strategy. For a discrete list of N_K keypoints $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{N_K}\}$, an inverted amplitude response map for the feature

f can be constructed as

$$\overline{\mu_f}(\mathbf{x}) = \sum_{i=1}^{N_K} w_i \mathrm{e}^{-\frac{\|\mathbf{x}-\mathbf{p}_i\|^2}{\sigma_i^2}}, \tag{5.15}$$

where $w_i \in [0,1]$ is a keypoint weight given by the descriptor (*i.e.*, a measure of how it is trusted to be a good keypoint), and σ_i is the size of the basin of attraction of the keypoint, which is an empirically-chosen value proportional to w_i. The quantity $\|\mathbf{x}-\mathbf{p}_i\|^2$ corresponds to the squared (Euclidean) distance from the current image location to the keypoint ${\mathbf{p}_i}^4$. The orientation and scale response maps can be constructed by filling small areas around each keypoint locations with their respective orientation and scale values extracted from the descriptor, and setting the remaining pixels to zero. Steerable filters learned from feature templates following the approach discussed in 4.4 are promising candidates as they provide all the aforementioned required pieces of information.

From there, the *local feature-based energy* is constructed to satisfy the following goals.

1. Attract control points to locations of high feature amplitude response (*i.e.*, to locations where the feature is present).
2. There, align the tangents to the local feature orientation.
3. Also encourage the tangent magnitude to correspond to that of the ideal tangent for that feature, up to a rescaling factor that depends on the scale at which the feature was detected.

The formulated energy is exclusively designed for landmark snakes as it exploits both the interpolating and tangent control properties of our model. Its expression is given by

$$E_{\text{single feature}}(\Theta) = \sum_{k=0}^{M-1} \overline{\mu_f}(\mathbf{r}[k]) + \frac{\alpha_f \left|C_{\varsigma_f(\mathbf{r}[k])}\|\boldsymbol{\tau}_f\| - \|\mathbf{r}'[k]\|\right|}{\overline{\mu_f}(\mathbf{r}[k])}, \tag{5.16}$$

[4] From a collection of discrete keypoint locations, a further possibility is to use the distance from the keypoints to the nearest control point as a direct measure of amplitude. Then, $\overline{\mu_f}(\mathbf{r}[k])$ in (5.16) is replaced by $\|\mathbf{r}[k]-\mathbf{p}_i\|^2$ in the spirit of (5.14).

with

$$\alpha_f = \cos^{-1}\left(\frac{\left|\langle\, \mathbf{R}_{\theta^*_{0,f}(\mathbf{r}[k])}\boldsymbol{\tau}_f,\, \mathbf{r}'[k]\,\rangle\right|}{\|\boldsymbol{\tau}_f\|\|\mathbf{r}'[k]\|}\right), \tag{5.17}$$

where $\mathbf{r}[k]$ is the location of the kth control point on the landmark snake curve, $\theta^*_{0,f}(\mathbf{r}[k])$ the local orientation of the feature at position $\mathbf{r}[k]$, $\overline{\mu_f}(\mathbf{r}[k])$ the image inverse of the local amplitude of the feature (where low values thus indicate that the feature is locally strongly present), and $\boldsymbol{\tau}_f$ the*ideal tangent* for this type of feature. Then, $C_{\varsigma_f(\mathbf{r}[k])}$ is a normalization factor corresponding to the scale $\varsigma_f(\mathbf{r}[k])$ at which the local feature was identified in the image coordinates of $\mathbf{r}[k]$. It thus adapts the magnitude of the ideal tangent to the actual feature scale. As explained earlier, the underlying idea is to favor local tangents that match the ideal one. This is in addition be enforced regardless of the direction of the parameterization (*i.e.*, curves with the same set of control points that are parameterized clockwise and counterclockwise are equivalent). The local and ideal tangents are therefore enforced to align up to an angle of π. This leads to the expression of the angle α_f, which corresponds to the angular discrepancy between the local tangent and the line obtained by extending the ideal tangent vector. It is derived from the nature of the scalar product, considering that

$$\frac{\left|\langle\, \mathbf{R}_{\theta^*_{0,f}(\mathbf{r}[k])}\boldsymbol{\tau}_f,\, \mathbf{r}'[k]\,\rangle\right|}{\|\boldsymbol{\tau}_f\|}$$

provides the orthogonal scalar projection of $\mathbf{r}'[k]$ onto the line determined by $\boldsymbol{\tau}_f$. This measure of orientation discrepancy is multiplied by the absolute difference in magnitude between the ideal and current tangents, taking into account possible scaling effects. The product of these two terms is divided by the image inverse of the local feature amplitude in order to relieve the constraints on tangent orientation and magnitude in areas where the feature is not present. When minimized, this energy term thus encourages the placement of control points at feature locations and, at these locations, favors a tangent orientation and magnitude that corresponds to the nature of the feature template.

A variation on this theme is to restrict the feature-based energy to specific control point indices. It amounts to specifying the nature of individual control points: some should act as landmark and be attracted to features, some others not. In this way, a particular snake parameterization can be imposed, *e.g.*, by ensuring that the kth control point is always acting as a landmark for a feature of interest. As a result, the energy is modified as

$$E_{\text{single feature},k}(\Theta) = \overline{\mu_f}(\mathbf{r}[k]) + \frac{\alpha_f \left| C_{\varsigma_f(\mathbf{r}[k])} \|\boldsymbol{\tau}_f\| - \|\mathbf{r}'[k]\| \right|}{\overline{\mu_f}(\mathbf{r}[k])}, \tag{5.18}$$

where $k \in \{0, \ldots, M-1\}$ is the index of a control point. The total energy of the snake curve is obtained by summing over the set K of control points that should act as landmarks, yielding

$$E_{\text{single feature, targeted}}(\Theta) = \sum_{k \in K} E_{\text{single feature},k}(\Theta). \tag{5.19}$$

Finally, different type of features can be combined in this spirit to yield a multiple local features-based energy. In this situation, each feature of interest has a corresponding template and ideal tangent. We denote the set of features of different nature as F. The input image is then processed to obtain one triplet of feature amplitude (and its corresponding image inverse), orientation and scale response maps for each considered feature type. The resulting energy is expressed as

$$E_{\text{multiple features}}(\Theta) = \sum_{i=1}^{N_F} \sum_{k=0}^{M-1} \overline{\mu_{f_i}}(\mathbf{r}[k]) + \frac{\alpha_{f_i} \left| C_{\varsigma_{f_i}(\mathbf{r}[k])} \|\boldsymbol{\tau}_{f_i}\| - \|\mathbf{r}'[k]\| \right|}{\overline{\mu_{f_i}}(\mathbf{r}[k])}, \tag{5.20}$$

with N_F the number of different feature types (*i.e.*, the cardinality of F) and the other elements defined as in (5.16). The tangents are hence adapted according to a mixture of ideal tangents, which are weighted by how strongly each corresponding feature is locally represented.

More generally, the multiple local features-based energy can be adapted to target precise control points in the same way as (5.19). Each control point can be assigned a dedicated energy term corresponding to the kind of feature it should represent. Let us consider again N_F different feature types. One denotes as K_i the set containing the indices of control points that should act as landmarks for feature

i, where $i = 1, \dots, N_F$. The resulting energy is expressed as

$$E_{\text{multiple features, targeted}}(\Theta) = \sum_{i=1}^{N_F} \sum_{k \in K_i} \overline{\mu_{f_i}}(\mathbf{r}[k]) + \frac{\alpha_{f_i} \left| C_{\varsigma_{f_i}(\mathbf{r}[k])} \|\boldsymbol{\tau}_{f_i}\| - \|\mathbf{r}'[k]\| \right|}{\overline{\mu_{f_i}}(\mathbf{r}[k])}. \tag{5.21}$$

The energy (5.20) is a particular case of (5.21) where $K_i = \{0, 1, \dots, M-1\}$ for all feature types $i = 1, \dots, N_F$ or, in other words, where control points are all indistinctively attracted to every type of feature. The energy (5.21) is the pinnacle of our points-and-curves story. It allows to explicitly associate an arbitrary set of control points to each type of features of interest. As a result, every individual control point acts as a feature-specific point seeker, deforming the curve it is attached to as it gets attracted to its target. The points and curves paradigms get united through this direct correspondence between snake control points and landmarks.

A possible unwanted effect of the feature-based energy term (5.21) is the accumulation of control points at a similar location in the image. The expression of the energy indeed encourages control points to be placed at landmark locations, but does not control that only one does so. To reduce the risk of having control points clustering at the same position, we propose to combine (5.21) with a self-repulsive energy given by

$$E_{\text{repulsive}}(\Theta) = \sum_{k=0}^{M-1} \frac{L_{\left[\frac{k-1}{M}, \frac{k+1}{M}\right]}(\Theta)}{2 \min\left(L_{\left[\frac{k-1}{M}, \frac{k}{M}\right]}(\Theta), L_{\left[\frac{k}{M}, \frac{k+1}{M}\right]}(\Theta)\right)}, \tag{5.22}$$

where $L_{[t_0,t_1]}(\Theta) = \int_{t_0}^{t_1} \|\mathbf{r}'(t)\| \mathrm{d}t$ is the arc length of the snake curve from t_0 to t_1. The energy is thus the ratio between half of the arc length from $\mathbf{r}[k-1]$ to $\mathbf{r}[k+1]$ to the minimum among the arc lengths from $\mathbf{r}[k]$ to $\mathbf{r}[k+1]$ and from $\mathbf{r}[k-1]$ to $\mathbf{r}[k]$. It is worth noting that $L_{\left[\frac{k-1}{M}, \frac{k+1}{M}\right]}$ is completely determined by the three pairs of control points and tangents in $k-1$, k and $k+1$. Similarly, $L_{\left[\frac{k-1}{M}, \frac{k}{M}\right]}$ and $L_{\left[\frac{k}{M}, \frac{k+1}{M}\right]}$ only depend on the two pairs of control points and tangents in $(k-1)$—k and k—$(k+1)$, respectively. This is a convenient consequence of the small support of the cubic Hermite splines (see Section 3.1 for more details). In the case of closed curves, the sequence of control points is periodic and no boundary conditions need to be considered. In the case of open curves, the set of indices in the summation is simply reduced from $k = 1$ to $M-2$, thereby excluding the two end points without any loss of generality. The effect of (5.22) is to introduce a repulsive force, which enforces that control points remain well distributed along the snake curve. Importantly, as it only consists of ratios, the energy is normalized. It therefore does not encourage the snake curve to grow endlessly. Moreover, the final energy used for optimization is always a combination of several terms. Other energy terms are required, *e.g.*, for the control points that do not act as landmarks. In addition to (5.22), these other terms also help avoiding that the snake curve collapses as control points cluster.

Proof of Concept. We demonstrate the efficiency of the local features-based energy (5.21) on synthetic images. We split our experiments in two parts. In the first one, we show how the quality of segmentation results can be improved by incorporating feature-based information in the segmentation process. In the second one, we illustrate the use of the feature-based energy targeted to specific control points in order to impose a common parameterization when segmenting several instances of similar objects in the image.

1. **Improved segmentation results on objects with prominent features.**

   We consider objects whose shapes imitate those of cells with filopodia-like protrusions. It consists in more or less sharp tips on a portion of the cell body, while the rest appears as smooth, as seen in Figure 5.8a. We search for two different feature types in these images, namely bumps and rounded tips. The corresponding templates are shown in Figure 5.7. Ideal tangents are obtained by setting a control point in the center of the template and manually adapting its tangent so as to obtain a curve portion that locally outlines the contour of the feature in the template. Test images are filtered by a pyramid of rotated matched filters built from each template. For the interested reader, this procedure is formulated in more details at the beginning of 4.4. Given a test image, three potential maps are obtained per feature type f_i: the amplitude response map μ_{f_i} containing at each location the highest response that could be obtained across the pyramid, the orientation response map θ^*_{0,f_i} containing the template rotation that provided this highest response, and the scale response map ς_{f_i} containing the scale of the pyramid at which the strongest response was found.

   We obtain segmentation results first by optimizing the landmark snake using

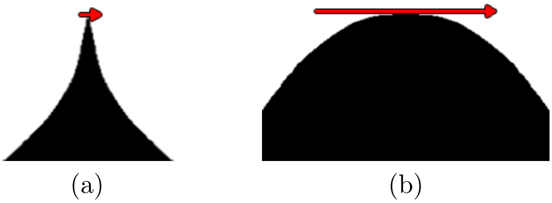


Figure 5.7: Feature templates for the filopodia test images. In both cases, the ideal tangent, overlaid as a red arrow, is that of a control point placed in the center of the template. It is set as an horizontal vector (*i.e.*, with an angle of zero), whose magnitude is then adapted to obtain a curve locally fitting the feature contours. (a) Tip-like template used to capture filopodia tip feature, (b) bump-like template used to detect the "back-end" of the cells.

a mixture of self-repulsive energy (5.22), constraint-based energy in the spirit of [65] and edge-based energy (5.9) (Figure 5.8c). The constraint-based energy used here only enforces that control points locate at area of high feature response, but does not affect the tangents. We then replace the constraint-based energy by our multiple local features-based energy (5.20) for bumps and rounded tips, while keeping the two other remaining terms (Figure 5.8d). Results are shown for the same initial conditions in all cases (Figure 5.8b). Encouraging control points to locate at precise places does not appear to be sufficient to obtain good segmentation results. When the tangents are not restricted, entangling occurs. It is worth noting that this effect could be seen as a counter-argument for using models involving tangents. However, much more control points are needed in order to obtain the same outline quality with classical spline-snake models. As a result, the model is more complicated and the risk of self-intersections increases. Our local feature-based energy precisely provides a good way to sufficiently constrain the landmark snake to benefit from the simplicity of its model when dealing with complicated shapes. Using this energy, precise segmentation results can be obtained in a reproducible and

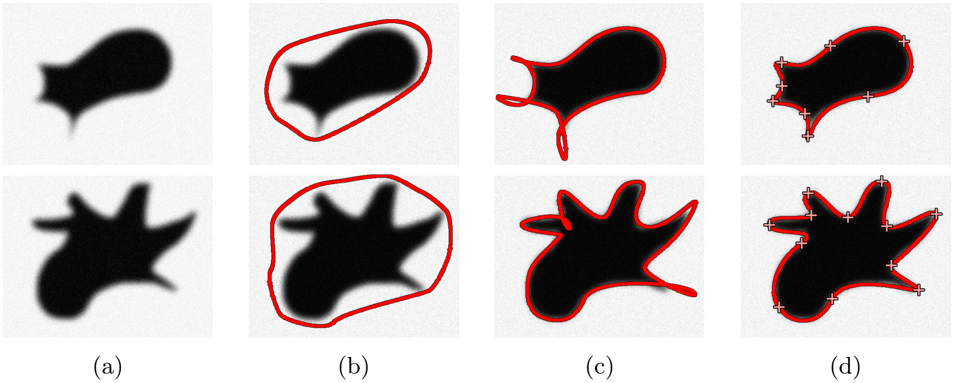


Figure 5.8: Illustration of the benefit of the local feature-based energy for constraining the landmark snake when segmenting complicated objects. (a) Synthetic images mimicking cells with filopodia-like protrusions, (b) landmark snake curve initialization, followed by segmentation results after optimizing on (c) a combination of edge- and constraint-based [65] energies, and (d) a combination of (5.20) and edge-based energies. The location of the control points in the final curve are indicated by crosses. In both cases the self-repulsive energy was used in order to avoid control points clustering.

stable manner. Although not depicted here, the local feature-based energy offers similar robustness to noise as the purely amplitude-based energy (5.14).

2. **Unique parameterization on different instances of similar objects.**

   The local feature-based energy can also be used as a way to enforce particular control points to be placed at locations where the feature is detected, making it possible to set a common origin for different snake curves. Imposing a unique parameterization of the segmentation curve is required to align and compare different instances of similar objects. To illustrate this, we provide a second proof-of-concept example using mock images of elongated rod-like objects that could be bacteria or nematodes (Figure 5.10a). These objects are smooth and standard spline-snake models can thus efficiently segment them (Figure 5.10c). However, there is then no easy way to favor particular

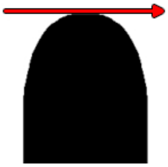


Figure 5.9: Feature templates for capturing extremities of rod-like shaped objects. The ideal tangent, overlaid as a red arrow, is that of a control point placed in the center of the template. It is set as an horizontal vector (*i.e.*, with an angle of zero), whose magnitude is then adapted to obtain a curve fitting the feature contours.

distributions of control points along the segmentation curve and, therefore, to impose a unique origin for the parameterization of each curve. This can be achieved by relying on (5.19) with a single feature type, namely rounded tip-like features. We use the template shown in Figure 5.9 and impose that the first ($\mathbf{r}[0]$) and middle ($\mathbf{r}\left[\frac{M}{2}\right]$) control points go to the object extremities, while the remaining ones evenly spread along both sides of the object, which translates to $K_1 = \left\{0, \frac{M}{2}\right\}$. As a result, solutions where these two particular control points locate at each object extremity are preferred (Figure 5.10d). It is then possible to align the outlines of the different objects, compare them and compute measures of difference in a meaningful manner.

In the above examples, we rely on a pyramid of rotated matched filters. This fairly simplistic feature detection method can be easily implemented with custom templates. However, it could be replaced by any feature detection method providing information about either the location or the magnitude, the local orientation and the local scale of the features in the image. For instance, steerable filters or convolutional neural networks tuned to detect instances of a given template appear as good alternatives. Due to the ideal tangent construction, the only critical aspect of the method is to be able to detect occurrences of the *same* pattern under the effect of rotation and scaling, and not a collection of different local features of different nature.

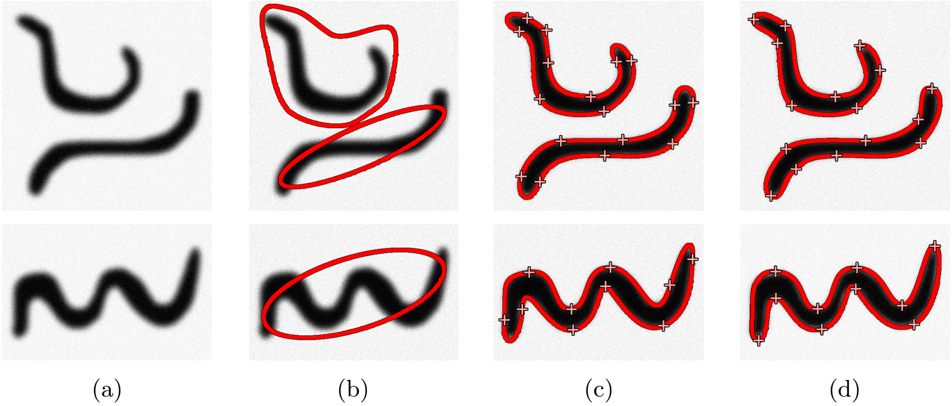


Figure 5.10: Illustration of the benefit of the local feature-based energy for imposing a particular snake parameterization by enforcing control points to lie at particular physical locations on the object. (a) Synthetic images featuring elongated rod-like structures., (b) landmark snake curve initialization, followed by segmentation results after optimizing on (c) an edge-based energy and (d) a combination of (5.19) and edge-based energies. The location of the control points in the final curve are indicated by crosses. In this experiment, the interest of using (5.19) lies in enforcing a given distribution of the control points along the object contour, not in improving the quality of the segmentation outline. In both cases the self-repulsive energy was used in order to avoid control points clustering.

The local feature-based energy unifies automated feature-detection methods and active contour segmentation. It can be seen as an extension of point-based only landmark energies for classical spline snakes. It does not only rely on feature detectors to attract control points at landmark locations, but also locally constrains the snake curve according to the nature, the orientation and the scale of the landmark. An interesting aspect of the method is that its robustness to noise boils down to the robustness of the feature detector. Similarly, the invariance properties of the feature detector used are inherited by the local feature-based energy.

5.3 Applicative Aspects

We now discuss additional aspects related to the implementation and user interactions with landmark snakes.

5.3.1 Implementation Details

The landmark snake plugin is implemented as a direct translation of the theory presented in Sections 5.1 and 5.2. More precisely, snake curves are directly constructed from (5.3) (for the open curve case) and (5.7) (for the closed curve case). Since our landmark snake model is continuously defined and practical implementations lie in the digital world, we discretize the continuous parameter t at a fixed sampling rate. To speed up execution time, samples of the cubic Hermite spline basis functions ϕ_1 and ϕ_2 are also precomputed and stored in lookup tables. This allows for a real-time response when interacting with the anchor points and tangent vectors of the snake.

5.3.2 Optimization

Automated optimization is carried out in an efficient way relying on Powell-like line-search methods [142], which are standard unconstrained optimization algorithms. These methods converge quadratically and require the computation of the derivatives of the energy function with respect to the parameters (*i.e.*, the coefficients of the Hermite spline). First, one direction is chosen according to the partial derivatives of the energy. Second, a one-dimensional minimization of the energy is performed along the selected direction. Finally, a new direction is chosen from the partial derivatives, while enforcing conjugation properties. These steps are repeated until convergence, and the position of the control points are updated as the energy gets minimized. Since the energy is continuously defined, centered finite differences a discrete approximation of the continuous derivative is used to estimate partial derivatives with respect to each of the control points. The convergence speed can sometimes be increased further by deriving a closed-form expression of the derivatives of the snake energy. However, while it has been shown in [70] that having an analytical expression for the energy gradient is especially useful when the number of parameters becomes large, our snake is defined by a small number of parameters (M anchor points plus M tangent vectors). Therefore, estimating the energy gradient using centered finite differences is satisfying in practice.

5.3.3 User Interaction

Our Hermite active-contour model facilitates interaction with the user in several ways. First, it relies on a smaller number of control points than many other parametric snake models. Then, since its construction involves basis functions of small support, the snake is only locally affected by structural changes. Modifications of one control point indeed merely affect the curve in a small local neighborhood. Finally, due to the interpolation properties of Hermite splines, anchor points truly lie on the actual contour on the snake, resulting in more intuitive interactions. We take advantage of this framework to provide a user-friendly interface in which the position and shape of the snake can be manually edited through manipulation of the control points and associated handles that correspond to their tangents. We also add the possibility to manually "freeze" control points and/or tangent handles at specific locations (corners, typically). When launching automated optimization, the "frozen" points are left out where they are while the rest of the curve automatically adapts to the outline of the object of interest. Our method can be deployed interactively by combining steps of snake initialization, optimization, and manual correction.

5.4 Practical Experiments

We carry out experiments to demonstrate that the landmark snake is well-suited to a wide range of segmentation tasks. For the quantitative evaluation of the snake performance, we rely on the Jaccard index $J \in [0, 1]$ defined as

$$J = \frac{|S \cap S_{\mathrm{ref}}|}{|S \cup S_{\mathrm{ref}}|}, \tag{5.23}$$

where S_{ref} is the ground-truth region and S the region segmented by the snake. It therefore measures the percentage of similarity between the two regions computed with a pixelwise discretization, 1 being perfect segmentation. Note that, because of discretization effects, a Jaccard index of 1 is almost never reached and values close to one can be considered as accurate segmentations.

We proceed in three steps. First, we discuss how the landmark snake compares to other state-of-the-art spline snakes. We discuss the introduction of sharp corners in the snake curve as well as local orientation constraints. We also study changes in the number of control points, dependence on initial conditions and robustness to noise. Finally, we show segmentation results in bioimages acquired using different microscopy modalities. In the literature, the snake energy is usually split into an *external* energy term, which contains all data-driven functionals, and an *internal* energy term encompassing the regularization energies. In the present case, snake optimization is only carried out relying on external energy terms, which are specified in each example.

5.4.1 Comparison with Existing Approaches

State-of-the-art spline snakes such as the ones built from cubic B-splines and exponential splines [70] adhere to the same principles as the landmark snake and are the most relevant competing methods. Here, we discuss how the two novel features of our model, namely the ability to influence tangent orientation through the design of directional energy functionals and that of reproducing corners by acting on tangent magnitude, are advantageous.

Reproduction of Corners

Cubic B-spline and exponential-spline snakes generate curves that are smooth by design. Sharp angles can be approximated but require a large number of distinct control points, as shown in Figures 5.11a and 5.11b. Conversely, the landmark snake is able to introduce sharp corners in curves with only one control point by setting the tangent magnitude to zero, as illustrated in Figure 5.11c. The cubic and exponential snakes are here composed of 9 control points and yield a poor approximation of the sharp tip. With 4 control points and associated tangent vectors, the landmark snake is able to perfectly recreate the tip of the drop-like shape. As the stability of the optimization process depends on the number of parameters, the landmark snake is also less likely to diverge.

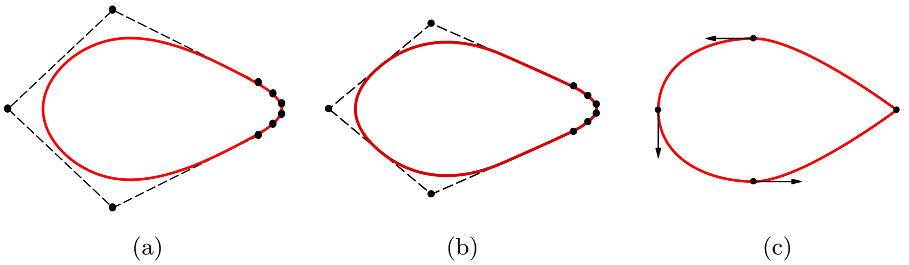


Figure 5.11: Generation of sharp corners using (a) cubic B-spline snake, (b) exponential-spline snake, and (c) landmark snake.

Constraints on the Local Orientation

Constraining the local orientation of the snake curve can serve two purposes. Firstly, it enhances robustness when segmenting complicated shapes. This aspect is discussed in 5.2.2 as a proof of concept of the usefulness of our purely directional energy. Energies taking into account, *e.g.*, the orientation of the gradient can be designed for classical spline snakes but involve artificially constructed unit normal vectors to the snake curve (see 2.3.2 for more details). Conversely, landmark snakes instantaneously provide such vectors: they simply are perpendicular to the tangents that directly appear in the snake model. As a result, energies incorporating local orientation information have simpler and more natural expressions.

Secondly, control over the local orientation allows locally adapting the snake curve to the presence of a feature. It is simply not possible to formulate terms such as the purely magnitude-based energy (5.14) or the feature-based energy (5.20) for classical spline snakes. These models do not provide a way to modify the local properties of the curve with a single control point. In classical spline snake models, control points only dictate where the curve goes, not how it looks like around them. A possible way to design local feature-based energies for classical spline snakes could be to impose constraints on groups of neighboring control points. The energy would enforce that the points behave together so as to locally shape the curve according to the nature and orientation of the feature. The number of required control points would however differ depending on the scale of the feature, resulting in overly complicated formulations. An added difficulty comes from the fact that control points of most smooth snakes (including cubic B-spline and exponential ones) do not lie on the curve. There is thus no correspondence between control points and landmarks, further complexifying the design of local feature-based energies.

5.4.2 Validation

Degrees of Freedom

The number M of control points is the main parameter of the Hermite-snake model. The choice of M is guided by the particular segmentation task being considered, keeping in mind that larger values of M grant additional flexibility to the curve and hence allow the reproduction of more complex shapes, but also enlarge the search space of the optimizer, which is therefore more likely to diverge or fall into local minima. A few control points are sufficient to segment smooth shapes, for which snakes with large M have a higher risk of getting entangled. For rougher, more complex contours, snakes with more control points and hence more degrees of freedom are required.

In the following experiments, we manually set the number of control points. There also exist ways to choose the number of control points automatically. For instance, one can apply a multiresolution strategy and perform several rounds of optimization starting from a snake with very few control points, optimizing it, resampling the resulting curve with more control points, further optimizing, and iterating in this way until the value of the snake energy at convergence stops decreasing. We do not provide a study of such approaches here, although it is used in the application presented in 6.3.

Dependence on Initial Conditions

Another important aspect is the initial snake curve from which the automated optimization process is started. A rough sketch should be sufficient since the optimization process takes care of precisely adapting it to the object boundary. Circular or oval shapes for closed snakes and broken lines for open ones are common initializations. From there, the curve obtained at convergence is a direct consequence of the nature of the chosen energy terms.

Robustness to Noise

We investigate the robustness of the landmark snake to noise in the image as a function of the number of coefficients M, when it is driven by a purely directional energy of the type (5.9). We created two synthetic images, one for the closed- and one for the open-curve case. We generated 100 realizations of these images in different PSNR conditions by adding a mixture of Gaussian and Poisson noises. Median Jaccard indices for each experimental setting are presented in Table 5.1. Snake initializations are overlaid in the thumbnails which depict the noise-corrupted images. In the closed-curve case, the performance of the snake degrades faster for large number of control points as the amount of noise increases. This can be explained by the presence of noise-induced local minima in snakes with large values of M. In the open-curve case, the decrease in performance for low PSNR is more gradual due to the fixed curve extremities, which reduces the risk of entanglement for large values of M.

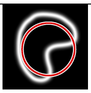
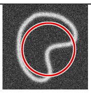
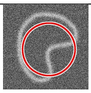
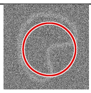
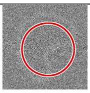
Considerations on Execution Time

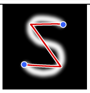
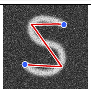
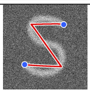
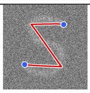
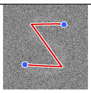
The automated optimization of the landmark snake from an initial curve runs in real time. The optimization process can either be clamped to a given number of iterations or can be left to run until a minimum is found. The time required for optimization is affected by the number of control points. The change in execution time is however hardly noticeable for a reasonable range of control point values (*i.e.*, $M = 2, ..., 20$). The number of manual clicks required to get an appropriate segmentation depends both on the initial shape and on the corrections possible required after automated optimization. It remains significantly lower than fully manual segmentation.

5.4.3 Segmentation of Bioimages

We provide insights into the capabilities of the landmark snake to segment bioimages. As we do not possess ground-truth information for the images presented here, we only rely on qualitative assessments for the quality of segmentation. For comparison purpose, we also provide results obtained with classical spline snakes. We rely on cubic B-spline snakes for the open curve case, and on exponential spline snakes [70] for the closed curve case. In all experiments, we use the same initializations for the different snake types and then let them evolve automatically until

Table 5.1: Segmentation results (Jaccard index) in noisy data using closed- and open-snakes, as a function of M.

						
	M	PSNR = ∞dB	PSNR = 11dB	PSNR = 8dB	PSNR = 6dB	PSNR = 5dB
Closed Snake	3	0.98	0.72	0.78	0.61	0.58
	4	0.98	0.72	0.75	0.59	0.48
	5	0.93	0.79	0.86	0.65	0.49
	6	0.96	0.85	0.69	0.51	0.37
	7	0.92	0.77	0.63	0.54	0.43
	8	0.93	0.62	0.59	0.42	0.29

						
	M	PSNR = ∞dB	PSNR = 11dB	PSNR = 8dB	PSNR = 6dB	PSNR = 5dB
Open Snake	4	0.93	0.87	0.75	0.64	0.46
	5	0.92	0.87	0.76	0.68	0.41
	6	0.95	0.95	0.75	0.66	0.42
	7	0.97	0.96	0.75	0.67	0.48
	8	0.92	0.92	0.72	0.65	0.43

convergence. To account for the fact that landmark snakes have twice more parameters per control points, we allow classical snakes with at most twice as many control points and display the best results we could obtain.

Segmentation of Phase-Contrast Microscopic Images

We segmented phase-contrast microscopic images of HeLa cells from [143]. Phase-contrast images are challenging for segmentation as they feature uneven gradients and halos around objects. We initialized closed landmark snakes with rough polygonal shapes around the cells, setting the number of control points as the number of edges in the polygon. For comparison, we initialized exponential snakes with a smoothed version of the polygons. Tangents at each points were initially set to zero. We then optimized the exponential snakes on (5.13) and the landmark snakes on

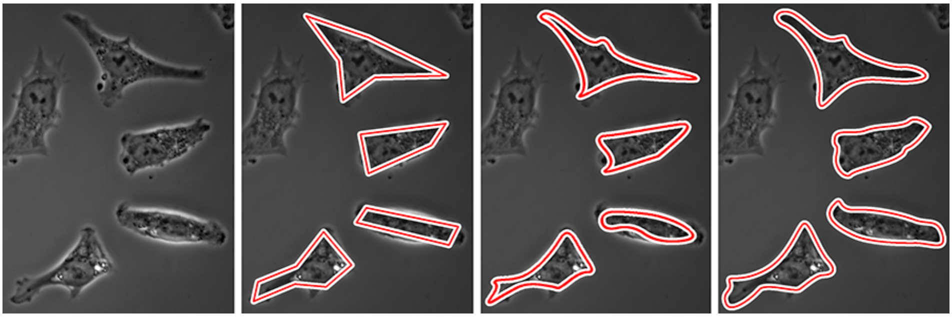


Figure 5.12: Phase-contrast images of HeLa cells. From left to right: original image, snake initialization, automated segmentation result using a classical spline-snake [70], automated segmentation result using landmark snakes.

the directional energy functional (5.9), both using an edge-sensitive steerable filter for the magnitude information. Results are displayed in Figure 5.12. Note that orientation information is of particular importance in these images as cell shapes exhibit fine details, in analogy with the synthetic example of Figure 5.4.

Segmentation of Differential-Interference-Contrast Images

Similar to phase-contrast microscopy, differential interference contrast (DIC) microscopy yields images where object boundaries are uneven due to shading effects. We obtained images of pancreatic acinar cells of live guinea pigs from the Cell Image Library [5] and initialized circular closed landmark snakes inside each cell. Each snake contained 5 control points and their associated tangents. We also generated exponential snakes from the same initializations. Relying on (5.13) for exponential snakes and on our directional energy functional (5.9), both using an edge-sensitive steerable filter, we obtained the segmentation results shown in Figure 5.13. Orientation information is crucial to constrain the overall shape of the snake in areas where cells touch each other and where little gradient information is available, similarly to the situation in Figure 5.5.

[5] Image corresponding to Figure 1 in [144], freely accessible from the Cell Image Library (http://www.cellimagelibrary.org/), accession number CIL:37314.

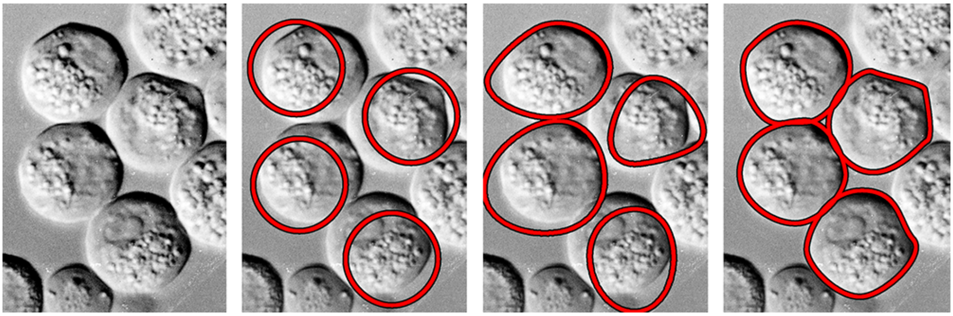


Figure 5.13: Differential-interference-contrast images of animal pancreatic acinar cells. From left to right: original image, snake initialization, automated segmentation result using a classical spline-snake [70], automated segmentation result using landmarks snakes.

Outline of Partly Polygonal Cells

We analyzed scanning-electron micrographs of the epidermal surface of lamprey larvae[6]. The image features microvilli that outline the polygonal borders between cells, while short microvilli cover the external surface in a reticular network. The motivation for using landmark snakes in these data is the presence of polygonal cells with sharp corners. We initialized closed landmark snakes inside the cells and manually imposed some critical corners. Note that the local feature-based energy cannot be used here because the corner locations are not discriminative enough with respect to other smooth parts of the cell contour. The number of control points was tuned depending on the complexity of each cell. We similarly initialized exponential snakes and imposed the same fixed points. The snakes were then automatically optimized. As shown in Figure 5.14, landmark snakes are able to properly segment both smooth and polygonal cells, and can therefore better accommodate biological variability than active contours that are uniformly smooth.

[6] Image from [145], freely accessible from the Cell Image Library, accession number CIL:11115.

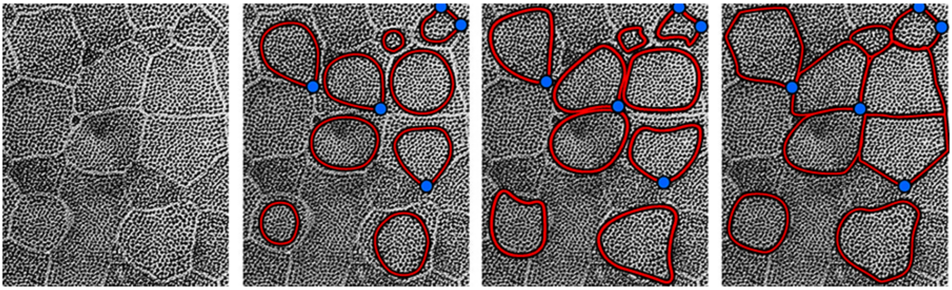


Figure 5.14: Epidermal cells in scanning-electron micrographs. From left to right: original image, snake initialization, automated segmentation result using a classical spline-snake [70], automated segmentation result using landmark snakes. User-defined fixed points are depicted as disks.

Segmentation of *Caenorhabditis Elegans*

From its ability to reproduce sharp angles, the landmark snake is well-suited to outline biological structures with pointy tips such as nematodes or protists. We segmented images from the *C. elegans* infection live/dead image set Version 1 provided by F. Ausubel (BBBC10) and available from the Broad Bioimage Benchmark Collection (BBBC, [146]). The living *C. elegans* nematodes appear in these images as elongated with pointy extremities and exhibit a variety of curved shapes. The difficulty when segmenting these data stems from the need to have a method that is flexible enough to accommodate the sinuous shapes of the nematodes while retaining the capability to capture sharp features. Sample results are shown in Figure 5.15. For comparison purpose, we also provide the best attainable results using exponential snakes [70] with at most twice more control points compared to the corresponding landmark snakes. Classical snakes like the exponential ones require control points to accumulate in order to generate pointy extremities, which increases the risk of artifacts as can be seen at the tips of some of the nematodes.

In Table 5.2, we report a quantitative comparison of our results against the manually annotated ground truth provided on the BBBC website[7]. For comparison,

[7] In Table 5.2, the nomenclature corresponds to the ground truth provided at http://www.broadinstitute.org/bbbc/BBBC010/.

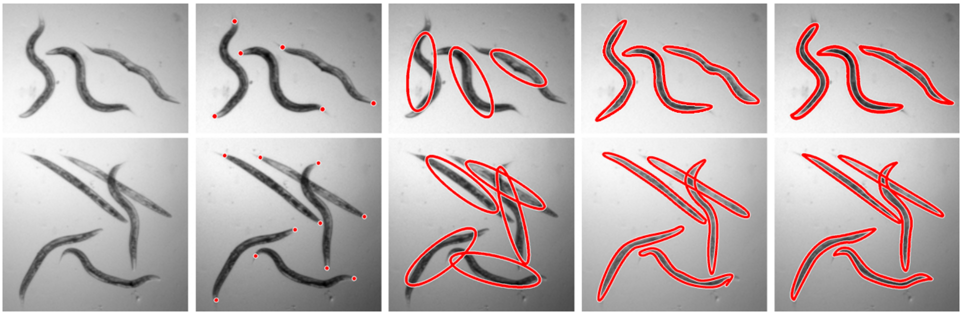


Figure 5.15: *Caenorhabditis elegans* nematodes. From left to right column: regions of interest from images B07 (top) and C03 (bottom) from the BBBC10 dataset, detected landmarks using a properly tuned Harris corner detector, active-contour initialization, automated segmentation result using a classical spline-snake [70], automated segmentation result using landmark snakes.

Table 5.2: Jaccard index—Snakes *versus* ground truth.

nematode #	1	4	5	6	10	12	14
B07 (Landmark snake)	0.79	n/a	0.83	0.79	n/a	n/a	n/a
B07 (Exponential snake [70])	0.77	n/a	0.80	0.79	n/a	n/a	n/a
C03 (Landmark snake)	0.84	0.87	n/a	n/a	0.81	0.88	0.84
C03 (Exponential snake [70])	0.79	0.83	n/a	n/a	0.80	0.82	0.81

we also give the same metrics for segmentation results obtained with exponential snakes [70]. A multiscale optimization scheme was used to boost robustness in such potentially crowded environment. Landmark active contours are first optimized on the purely amplitude-based energy (5.14) targeted to control points with index 0 and $\frac{M}{2}$ to capture the extremities of the nematodes. The locations of the extremities of the nematode are obtained from a properly tuned Harris corner detector. The multiscale steerable edge-based energy (5.9) is then used to fit the rest of the contour. Our segmentation results are robust to initial conditions provided that the initial snakes do not significantly overlap nearby nematodes.

Segmentation of *Heligmosomoides Polygyrus Bakeri*

We also tested our approach on images of unstained *Heligmosomoides polygyrus bakeri* (*H. bakeri*) nematodes[8]. This natural parasite of mice is a purely enteric nematode and appears as translucent with uneven interior. Here, we experiment both with the open and closed landmark snake for two different tasks.

The first task is to get a good approximation of the medial axis of the nematodes relying on open landmark snakes. Depending on experimental conditions, the nematodes are either "clean" or get cluttered by immune cells. By adapting the number of control points to the complexity of the nematode shape and initializing open landmark snakes with simple broken lines, we were able to accurately detect the medial axis, as shown in Figure 5.16. Snake optimization was carried out relying on (5.13) for cubic B-spline snakes, and on the directional energy functional (5.9) for landmark snakes, using in both cases properly tuned ridge-sensitive steerable filters. As they do not incorporate any directional information, the cubic B-spline snakes are more likely to get trapped in local minima and yield a less good estimation of the medial axis.

The second experiment consists in segmenting the complete nematode contour. *H. bakeri* appears as translucent and is therefore more difficult to segment than *C. elegans*. Moreover, as these nematodes are moving fast, their extremities are blurred in most images, although being pointy in reality. We optimized our landmark snakes on a combination of the purely directional energy functional (5.9) and the purely magnitude-based potential (5.14). Landmark locations, here corresponding to the extremities of the nematode, were obtained from a properly tuned Harris corner detector. Typical results are shown in Figure 5.17. Snakes with 4 to 8 control points were used, depending on the shape complexity. We had to initialize snakes close to the actual contour to get a satisfactory result because of the difficulty of the task. To provide comparison with existing approaches, we also show the best results we could obtain using exponential snakes [70] with at most twice more control points. In the absence of a corner potential, the extremities of the nematodes tend to get badly detected. Our rather simple corner detector sometimes fails to capture the correct location of the extremities in these data as observed in Figure 5.18, yielding imprecise segmentation of the tips. Such errors can easily be corrected in a semi-automated way by manually dragging the extremities while the rest of the curve

[8] Images courtesy of J. Esser, Laboratory of Intestinal Immunology, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland.

gets automatically adjusted.

Epidermal Cell Membrane Outline

We processed images of epidermal cells from *Drosophila* larvae. These open-access data [147] feature cells immunostained for Fasciclin III, which labels their basolateral surface. Detecting cell shape in these images is challenging for two reasons: first, images are low resolution and noisy. Second, due to the wound healing assay experimental setting, cells can be extensively deformed. Because of the poor data quality, active contour approaches solely driven by distance maps are practically unusable. They are too sensitive to initialization and therefore diverge most of the time. Relying on a properly designed steerable ridge detector following (5.12) and a custom multiscale symmetry center detector constructed following the procedure introduced in 4.3, we were able to outline cells using landmark snakes. The energy was composed of a mixture of the purely-directional energy (5.9) and of the local feature-based energy (5.16) targeted towards three-fold junctions. Results are shown in Figure 5.19. The addition of the local feature-based energy is crucial to make the optimization process less sensitive to initial conditions. In this framework, it is sufficient for the snake to lie inside the cell in order to get proper results. Seed points for the initialization can thus easily be automatically provided using for instance a watershed transform. It is worth pointing out that, although the local feature-based energy dramatically improves the robustness of the optimization, result quality does not only depend on it. As seen in Figure 5.19 on the right, the detector sometimes fails to catch a corner or badly locate some of them. Since the snake relies on a combination of ridge- and local feature-based energies, missing information about feature locations can be compensated by information from cell boundaries. In the same figure, we also illustrate the best results we could obtain using exponential snakes [70] with at most twice more control points. For these, several attempts were required in order to obtain segmentation results without loops and self intersections. Undesirable wiggling effects can be observed around the corners as several control points accumulate to try to properly outline these sharp features.

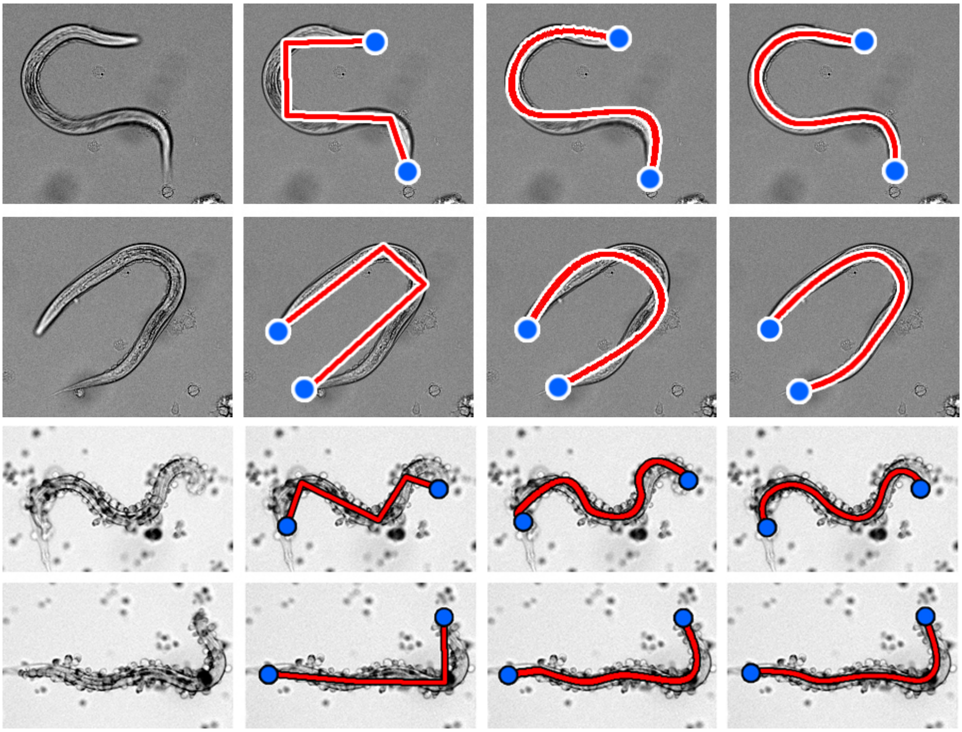


Figure 5.16: *H. bakeri* nematodes in clean (top row) and cluttered (bottom row) environments. From left to right column: original image, snake initialization, automated segmentation using a classical cubic B-splines snake, automated segmentation using landmark snakes. Fixed points at the extremities are depicted as disks.

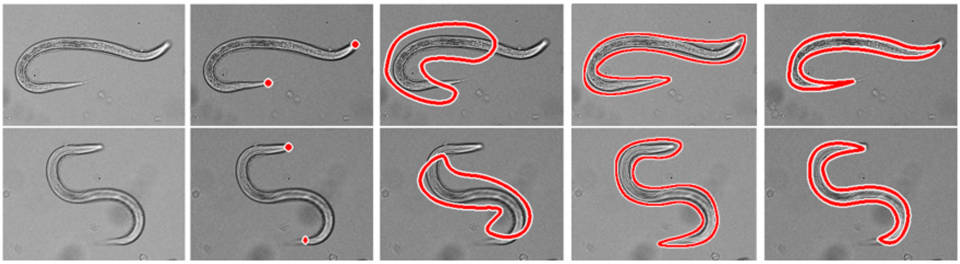


Figure 5.17: Translucent *H. bakeri* nematodes. From left to righ column: original image, detected landmarks using a properly tuned Harris corner detector, active-contour initialization, automated segmentation using an exponential spline snake [70], automated segmentation using landmark snakes.

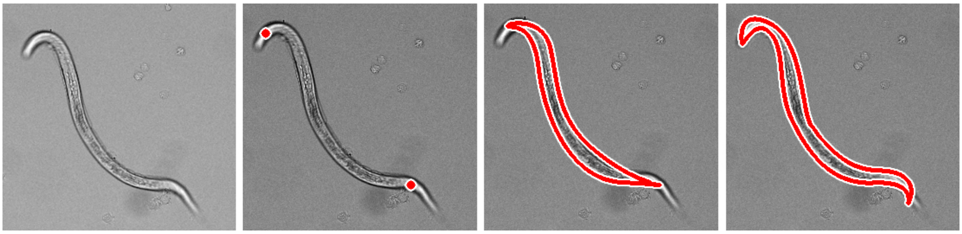


Figure 5.18: Example of badly located landmark at the tips of *H. bakeri* requiring manual correction. From left to right column: original image, detected landmark using a properly tuned Harris corner detector, segmentation result after automated optimization, segmentation result after manual correction, which simply amounts to dragging the snake extremities to match the real tips of the nematode.

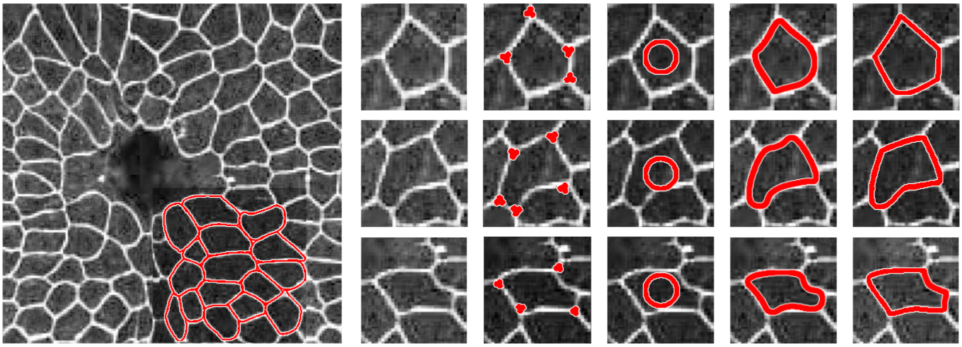


Figure 5.19: Epidermal cell membrane. Left: original image with final overlaid segmentation results for a region of interest. Right: step-by-step illustration of the segmentation process. For the three sample cells, from left to right column: original image, detected landmarks using a multiscale steerable symmetry center detector (see 4.3 for more details), active contour initialization, segmentation results after automated optimization of exponential snakes [70], segmentation results after automated optimization of landmark snakes. Landmark snakes with 5 control points and exponential snakes with up to 10 control points were used.

Chapter 6

Practicing Landmark Active Contours

In this chapter, we present three examples of bioimage analysis applications of the various tools we introduced up to now. First, we focus on the problem of analyzing fly leg and body motion. To do so, we rely on custom feature detectors built in the spirit of what we presented in Chapter 4. In our second example, we tackle a tracking problem involving *Mycobacterium smegmatis.* There, we require Hermite spline models to represent smooth open curves. Finally, in our third application, we study the dynamics of swimming *C. elegans* nematodes using the full potential of our landmark snake. The nematodes are first segmented with landmark snakes, mixing contour finding and landmark detection. Then, we analyze their motion with a novel procedure inspired from eigenshape analysis. The latter heavily relies on the Hermite spline representation of the segmented shapes and provides a good illustration of the many practical interests of the landmark snake model.

6.1 Fly Motion Analysis using Custom Feature Detectors

Many terrestrial animals rely on complex limb movements to locomote, groom, court, mate, and fight. Discovering how these and other fundamental behaviors are orchestrated by the nervous system requires manipulations of both the genome and the nervous system as well as quantitative measurements of behavior. The vinegar fly, *Drosophila melanogaster*, is an attractive model organism for uncovering the neural and genetic mechanisms underlying behavior. First, it provides formidable genetic tools that allow experimenters to remotely activate, silence, visualize and modulate specific gene function in identified neurons [148]. Second, a number of sophisticated methods have been developed that permit robust tracking of *Drosophila* body movements a promising set of tools for biological screens [149, 150, 151, 152, 153, 154]. By contrast, similarly robust methods with the precision required to semi-automatically track leg segments are largely absent. State-of-the-art approaches suffer from several drawbacks. For example, the most precise methods require the manual placement of visible markers on tethered animals [155] as well as sophisticated fluorescence-based optics ([156] provides another example in cockroaches). Marking insect leg segments is a time-consuming process that limits experimental throughput. On the other hand, the most high-throughput approach for marker-free leg tracking in freely behaving *Drosophila* uses complex optics to measure Total-Internal-Reflection Fluorescence (TIRF) when the distal leg tips (claws) of walking animals scatter light transmitted through a transparent floor [157]. An image-processing based method relying on hidden Markov models has been developed to track leg tips only [158]. Other approaches involve specialized setups that maintain the moving fly inside the camera frame by controlling the position of the x–y stage based on the camera image in real time, thus reducing the problem to simple registration (or image alignment) [159, 160]. Although these methods can resolve the claws of each leg, they cannot detect their segments. Thus, they provides only binary information about whether or not a leg is touching the surface and cannot resolve the velocity of joints during swing phases, stance adjustments, or non-locomotive limb movements such as reaching [161] or grooming [162]. Ultimately, such measurements will be necessary to gain a complete understanding of how the nervous system controls each limb.

The first practical application of our theory uses feature detection implemented with the steerable filter framework. In [163][1], we developed FlyLimbTracker, a method that permits semi-automated, marker-free tracking of the body and leg segments of freely walking *Drosophila*. Our approach relies purely on the analysis of high-speed, high-resolution videos and does not require complicated optical setups. FlyLimbTracker uses a combination of different parametric spline-snakes to process objects in high-frame-rate image sequences. Using this approach, we show that our method can semi-automatically track freely walking or grooming *D. melanogaster* in video data that span a wide range of spatial and temporal resolutions. FlyLimbTracker reduces the number of user clicks required a proxy for annotation speed by approximately 6-fold (see 6.1.2). We implemented this approach as a plug-in for the open-source software Icy, a community-maintained and user-friendly image processing environment for biological applications [11, 164, 165]. This makes it amenable to customization for behavioral measurements in flies with altered morphologies (*e.g.*, following leg removal) and, potentially, in other species (*e.g.*, stick insect, cockroach).

6.1.1 Description of the Method

Drosophila Behavior Experiments

We used adult female *Drosophila melanogaster* of the Canton-S strain at 2 to 4 days post-eclosion. Flies were raised on a 12h light:12h dark cycle at 25°C. Experiments were performed in the late afternoon (Zeitgeber time) after flies were starved for 4 to 6 hours in humidified 25°C incubators. During experiments, we placed the flies in a custom designed acrylic arena (pill shaped: 30mm × 5mm × 1.2mm) illuminated by a red ring light. We captured behavioral video using a high-speed (236 frames-per-second), high-resolution (2560×918 pixels) camera viewing animals from below.

Automated Body and Leg Tracking

FlyLimbTracker performs tracking in several steps. First, the user is asked to manually initialize the position of the body and leg segments of the fly in a single frame of the image sequence. To perform image segmentation, FlyLimbTracker uses

[1] This work has been carried out in collaboration with Prof. P. Ramdya, Neuroengineering Laboratory, EPFL, Lausanne, Switzerland.

a closed spline snake to segment the *Drosophila* body and open spline snakes to model each of the legs of the fly. The segmentation results from the initial frame are combined with image features to propagate body and leg segmentation to the frames immediately preceding or following in the sequence. At any time, the user can stop, edit, and restart automated segmentation proceeding forward or backward in time. Manual corrections are taken into account when tracking is resumed. Hereafter, we formalize and illustrate the construction of the segmentation models for the body and legs of the fly, respectively. We then describe how these models are propagated to track the fly through the image sequence.

***Drosophila* Body Model.** The body of the fly is defined as a two-dimensional closed spline snake curve $\mathbf{r}$ composed of M control points as in (2.3), that is

$$\mathbf{r}(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \sum_{k=0}^{M-1} \mathbf{c}[k]\phi_{\text{per}}(Mt-k), \tag{6.1}$$

with $t \in [0,1]$, where $\{\mathbf{c}[k]\}_{k\in\mathbb{Z}} = \{(c_1[k], c_2[k])\}_{k\in\mathbb{Z}}$ is the M-periodic sequence of control points and $\phi_{\text{per}}(t) = \sum_{n=-\infty}^{\infty} \phi(t-Mn)$ the M-periodization of a basis function ϕ. We use ellipse-reproducing splines [64] as basis functions (discussed in 2.3.1). Formally, the generator we rely on is given by

$$\phi(t) = \begin{cases} \frac{\cos\left(\frac{2\pi|t|}{M}\right)\cos\left(\frac{\pi}{M}\right) - \cos\left(\frac{2\pi}{M}\right)}{1-\cos\left(\frac{2\pi}{M}\right)}, & 0 \leq |t| < \frac{1}{2}, \\ \frac{1-\cos\left(\frac{2\pi\left(\frac{3}{2}-|t|\right)}{M}\right)}{2\left(1-\cos\left(\frac{2\pi}{M}\right)\right)}, & \frac{1}{2} \leq |t| < \frac{3}{2}, \\ 0, & |t| \geq \frac{3}{2}. \end{cases} \tag{6.2}$$

The model of the body of the fly is composed of $M = 18$ nodes.

To optimize the snake automatically from a coarse initial position to the precise boundaries of the body of the fly, we define a snake energy composed of three elements. It is a function of the snake representation Θ and is formulated as

$$E_{\text{body}}(\Theta) = E_{\text{edge}}(\Theta) + E_{\text{region}}(\Theta) + E_{\text{shape}}(\Theta). \tag{6.3}$$

The first element E_{edge} is a classical edge-based energy term relying on gradient information to detect the body contour already introduced in (2.9). It is formally expressed as

$$E_{\text{edge}}(\Theta) = -\oint_{\mathcal{C}(\Theta)} \langle \nabla I(\mathbf{r}), \mathbf{n}_{\mathbf{r}} \rangle \mathrm{d}r, \tag{6.4}$$

where $\nabla I(\mathbf{r}) = \left(\frac{\partial I(\mathbf{r})}{\partial r_1}, \frac{\partial I(\mathbf{r})}{\partial r_2} \right)$ is the gradient of the image at position (r_1, r_2), $\mathbf{n}_{\mathbf{r}}$ denotes the inward-pointing unit normal to the curve at (r_1, r_2), and $\mathcal{C}(\Theta)$ is the snake curve determined by the control points $\{\mathbf{c}[k]\}_{k\in\mathbb{Z}}$. The energy term is negative since it has to be minimized during the optimization process. From Green's theorem, we can transform the line integral into a surface integral

$$E_{\text{edge}}(\Theta) = -\int_{\Omega(\Theta)} \Delta I(\mathbf{r}) \mathrm{d}\mathbf{r}, \tag{6.5}$$

with Ω the region enclosed by the snake curve, and $\Delta I(\mathbf{r})$ the Laplacian of the image at position $\mathbf{r} = (r_1, r_2)$. The energy E_{edge} is minimized when the snake curve is aligned to the direction of the gradient.

The second term E_{region} uses region statistics to distinguish the object from the background, in the spirit of (2.10). Specifically, it is computed as the intensity difference between the region enclosed by the snake Ω and the region surrounding it $\Omega_\mu \backslash \Omega$ as

$$E_{\text{region}}(\Theta) = \frac{1}{|\Omega(\Theta)|} \left(\int_{\Omega(\Theta)} I(\mathbf{x}) \mathrm{d}\mathbf{x} - \int_{\Omega_\mu(\Theta) \backslash \Omega(\Theta)} I(\mathbf{x}) \mathrm{d}\mathbf{x} \right), \tag{6.6}$$

where I is the image, Ω the region enclosed by the snake curve, and $|\Omega|$ the signed area of the snake, which is defined as

$$|\Omega(\Theta)| = \oint_{\mathcal{C}(\Theta)} r_2 \, \mathrm{d}r_1, \tag{6.7}$$

with r_1, r_2 given by (6.1). Minimizing this term encourages the snake to maximize the contrast between the area it encloses and the background.

Finally, the last term E_{shape} is a shape-prior energy detailed in [166]. This term measures the similarity between the snake and its projection onto a given reference curve. It favors the convergence of the contour to an affine transformation of the reference shape. The smoothness and regularity of the reference shape are thus preserved. Moreover, this term prevents the formation of loops and aggregation of nodes during the optimization process. In our case, the reference shape is a symmetric fly body contour with 18 nodes (Figures 6.1A and 6.2B).

The algorithm depends on an initial user input to coarsely locate the fly in a frame of the image sequence. With a single mouse click, a two-step multiscale optimization scheme is initiated. A spherical active contour composed of three control points is first created, centered at the mouse position. The snake is optimized on $E_{\text{edge}} + E_{\text{region}}$ to form an elliptic curve surrounding the fly. In this way, the major axis of the elliptical snake aligns with the anteroposterior axis of the fly, and the minor axis is perpendicular to it. The 3-point elliptical snake fit to the body of the fly can be expressed following [70] as

$$\mathbf{r}(t) = \mathbf{R}_0 + \mathbf{R}_1 \cos(2\pi t) + \mathbf{R}_2 \sin(2\pi t), \tag{6.8}$$

where $t \in [0, 1]$ and

$$\mathbf{R}_0 = \frac{1}{3}\sum_{k=0}^{2} \mathbf{c}[k], \quad \mathbf{R}_1 = \sum_{k=0}^{2} h_c[k]\mathbf{c}[k], \quad \mathbf{R}_2 = \sum_{k=0}^{2} h_s[k]\mathbf{c}[k], \tag{6.9}$$

with

$$h_c[k] = \frac{2}{3}\cos\left(\frac{\pi}{3}\right)\cos\left(\frac{2\pi k}{3}\right), \quad h_s[k] = \frac{2}{3}\cos\left(\frac{\pi}{3}\right)\sin\left(\frac{2\pi k}{3}\right), \tag{6.10}$$

and $k \in \mathbb{Z}$. Note that the $\mathbf{c}[k]$ are the control points of the snake appearing in (6.1). Relating this to the general parametric equation of an ellipse of major axis a, minor axis b, and center (x_c, y_c) allows extracting the parameters of the 3-control points snake fit to the body of the fly. Namely, $(x_c, y_c) = \mathbf{R}_0$, $a = \max(\|\mathbf{R}_1\|, \|\mathbf{R}_2\|)$ and $b = \min(\|\mathbf{R}_1\|, \|\mathbf{R}_2\|)$. Knowing a, the orientation of the ellipse in the image can be computed. The ellipse fit is then replaced by an 18-node fly-shaped closed snake that has been rotated and dilated to match the ellipse's length and orientation (Figure 6.1A). An ambiguity remains since two potential snake models can be initialized for a given ellipse, with opposite anteroposterior axis orientation. To resolve it, snakes with both possible orientations are optimized in the image on E_{body} in addition to E_{edge} and E_{region}. The solution resulting in the lowest cost (*i.e.*, the lowest energy value at convergence) is kept.

***Drosophila* Leg Model.** Once the body of the fly is properly segmented, models for each of its legs are added. First, the positions of leg coxa-thorax attachment points (hereafter referred to as *anchors*) are automatically computed based on the body segmentation. The location of the six leg anchors with respect to the reference body model are empirically determined as linear combinations of three axes defined by the head-thorax junction, the thorax-abdomen junction, and the thorax length (Figure 6.1B). These reference locations are then adapted in accordance with deformations of the body model over the image sequence.

User input is required to initialize the position of each leg prior to tracking. Initialization is based on a single click per leg: the user indicates the claw (hereafter referred to as *tip*) of each leg with a mouse click. The click location is assigned to the most likely body anchor using a probabilistic formulation based on the distance and intersection with the body model of the fly and that of other leg models. Once a leg tip and a leg anchor have been paired, a dynamic programming method [167] is initiated to automatically trace the leg from the anchor to the tip. To be identified in a robust way, the legs of the fly are enhanced by processing the image frame with a ridge-sensitive steerable filter of the form (4.5).

Dynamic programming yields the globally optimal solution for a given separable problem. In particular, it can be used to implement algorithms solving shortest path problems. The Viterbi algorithm [168] is a particular instance of dynamic programming and relies on a graph-based representation: the shortest path is represented as a sequence of successive nodes in a graph that minimize a cost function. To trace a leg from its anchor to its tip, the graph is built by interpolating image pixels along two orthonormal axes. The first axis (axis $\mathbf{k}$, indexed by k) is defined by the unit vector along the straight segment linking the anchor of a leg to its tip. The second axis (axis $\mathbf{u}$, indexed by u) is perpendicular to the first axis and is thus defined by the normal vector to $\mathbf{k}$. On a path, we denote by u_k the index u corresponding to a given k. A path is described by a collection of nodes (k, u_k). The cost of the path at index $k+1$ along axis $\mathbf{k}$ is given by

$$C[k+1] = C[k] + \lambda \left(\frac{1}{L_S} \sum_{(x,y)\in S} I_{\text{feat}}(x,y) \right) + (1-\lambda)|u_k - u_{k+1}|, \tag{6.11}$$

where $C[k]$ is the cost of the path at location k on axis $\mathbf{k}$, S is the collection of image pixels (x,y) belonging to the segment between nodes (k, u_k) and $(k+1, u_{k+1})$, L_S is the pixel length of the considered segment, I_{feat} is the version of the current frame filtered by the steerable feature detector, and $\lambda \in [0,1]$ is a weighting coefficient. The first term corresponds to a discretized integral of the image along the segment linking nodes k and $k+1$, and tends to favor paths going through low pixel values. The second term is composed of the distance on axis u between two successive nodes and thus prevents large jumps along the u axis. As a result, the optimal path follows relatively dark regions in the image with respect to the background in accordance with the first term, while retaining a certain level of smoothness due to the second term. The relative contributions of each term are determined by λ.

In contrast to body segmentation, leg segmentation uses open snakes rather than closed ones. Fly legs are parameterized by a curve composed of $M = 4$ control points (Figure 6.1C and 6.2C). For each leg, the body anchor, $\mathbf{l}[0]$, is fixed. The discrete path obtained with dynamic programming is used to initialize the leg snake. The rationale behind this two-step procedure is two-fold. First, dynamic programming is very robust and can therefore effectively trace the leg from a body anchor to its tip. However, it is computationally expensive. By contrast, snake-based methods are more likely to diverge when initialized far from their target but are computationally inexpensive since only a few control points need to be stored and modified to update the curve. We combined these approaches by finding an initial path for each leg using dynamic programming once, and then transforming this path into a parametric curve for further optimization. The parametric representation of the leg snake curve is defined as

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = \sum_{k=0}^{M-1} \mathbf{l}[k]\phi(Mt-k), \tag{6.12}$$

where $t \in [0,1]$ and $\{\mathbf{l}[k]\}_{k\in\mathbb{Z}} = \{(l_1[k], l_2[k])\}_{k\in\mathbb{Z}}$ are the leg snake control points. Since *Drosophila* legs are composed of relatively straight segments between each joint, we use linear splines as basis functions ϕ. The leg control points are linked with linear interpolation, yielding a piecewise-linear curve. Each control point has a unique identifier that can be used for subsequent data processing (Figure 6.1D). The full process finally consists in taking a single raw image (Figure 6.2A) and using active contours to segment the body (Figure 6.2B) and legs (Figure 6.2C).

Segmentation Propagation (Tracking). High frame-rate videos ensure that the displacement of the body of the fly between successive frames is small. FlyLimbTracker takes advantage of this fact to propagate body and leg snakes from one

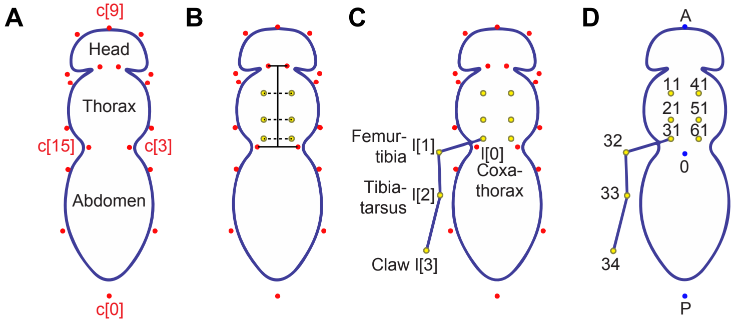


Figure 6.1: *Drosophila* body and legs model. (A) The body model is a closed snake consisting of 18 control points (**c**[0] to **c**[17]). Control points **c**[0] and **c**[9] correspond to the posterior-most position on the abdomen and the anterior-most position on the head, respectively. All other control points are symmetric along the anteroposterior axis of the body (*e.g.*, control points **c**[3] and **c**[15]). (B) Six leg anchor positions (yellow) between the coxa and thorax are defined empirically based on a linear combination of distances from the head-thorax boundary, the thorax-abdomen boundary, and a distance from the thoracic midline. These positions are then adapted depending on how the body model is deformed to fit the contours of a specific animal. (C) The leg model consists of four control points including a thorax-coxa attachment **l**[0], the femur-tibia joint **l**[1], the tibia-tarsus joint **l**[2], and the pretarsus/claw **l**[3]. For simplicity, control points are only shown for a single leg. (D) In total, 27 positions are extracted for each fly per frame: a centroid (0), anterior point (A), posterior point (P), as well as the body anchor, first intermediate, second intermediate and tip for each of the six legs. Our data labeling convention is as follows. Right and left legs are numbered 1 to 3 (front to rear) and 4 to 6 (front to rear), respectively. Each leg has four control points labeled 1 to 4 depending on whether they correspond to a body anchor (1), leg joints (2 and 3), or claw (4). In each label, the leg number is shown in the tenths digit and the control point in the units digit. For example, label 11 refers to the body anchor of the right prothoracic leg 1. For simplicity, only the control points of leg 3 are shown.

frame to the next during tracking. The body snake in frame $t+1$ is segmented by optimizing a contour initialized as the corresponding snake from frame t using the body snake energy (6.3). This approach is sufficient to obtain good segmentation provided that there is some overlap between the body of the animal in successive frames.

Compared to body movement, leg displacement can be larger between frames. Therefore, leg snakes require a more sophisticated algorithm to be properly propagated. First, the anchor of each leg is automatically computed from the newly propagated fly body. Since each leg is modeled as a 4-node snake, the three remaining leg snake control points are optimized using the snake energy

$$E_{\mathrm{leg}}(\Theta) = E_{\mathrm{feat}}(\Theta) + E_{\mathrm{EDT}}(\Theta) + E_{\mathrm{segments}}(\Theta) + E_{\mathrm{extremity}}(\Theta), \tag{6.13}$$

where Θ is the snake representation. The first term corresponds to the integral along the leg of the current frame filtered by a custom steerable filter tuned to detect leg features. It is expressed as

$$E_{\mathrm{feat}}(\Theta) = \oint_{\mathcal{C}(\Theta)} I_{\mathrm{feat}}(\mathbf{s})\mathrm{d}s = \int_0^1 I_{\mathrm{feat}}(\mathbf{s}(t))|\mathbf{s}'(t)|\mathrm{d}t, \tag{6.14}$$

where I_{feat} is the version of the current frame filtered by the steerable feature detector, and $\mathcal{C}(\Theta)$ is the snake curve determined by the control points $\{\mathbf{l}[k]\}_{k\in\mathbb{Z}}$ as described by (6.12). The second term is computed in a similar way as the integral along the leg of the Euclidean distance transform (EDT [169]) as

$$E_{\mathrm{EDT}}(\Theta) = \oint_{\mathcal{C}(\Theta)} I_{\mathrm{EDT}}(\mathbf{s})\mathrm{d}s = \int_0^1 I_{\mathrm{EDT}}(\mathbf{s}(t))|\mathbf{s}'(t)|\mathrm{d}t, \tag{6.15}$$

with E_{EDT} the Euclidean distance-transformed version of the current frame and $\mathbf{s}$ the snake curve as described by (6.12). Each individual linear segment in a leg should be roughly constant in length across a video, aside from changes introduced by projecting the three-dimensional leg onto a two-dimensional image. Taking this into account, the third term of the leg energy penalizes solutions where leg joint positions result in segments whose lengths vary considerably from one frame to the next. This prevents unrealistic configurations of the leg joints. Formally, E_{segments} is computed as the sum of absolute differences in length between each segment of the target leg at two successive frames. Finally, the fourth term is used to determine

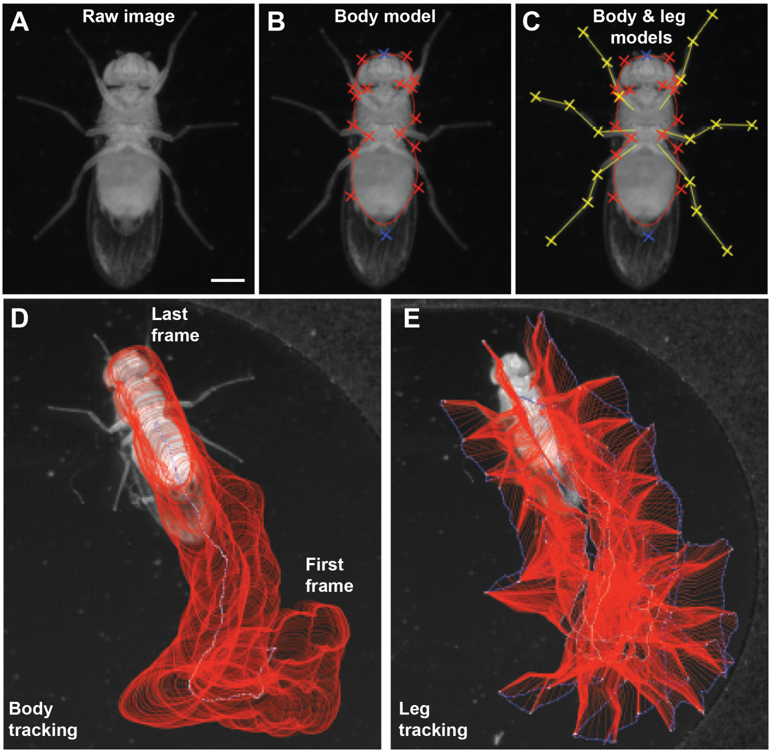


Figure 6.2: FlyLimbTracker uses active contour models to annotate *Drosophila* body and legs. (A) An example raw image of the ventral surface of a fly used for segmentation. (B) This image is first segmented using the parametric body snake consisting of 18 control points (red and blue crosses). (C) Subsequently, leg segmentation is initialized by automatic tracing from body anchor points to user-defined leg tips. From this initialization, open snakes consisting of four control points are generated (yellow crosses). (D) Body and (E) leg segment annotations for flies during a 455-frame (1.93s) sequence. Annotation results (red) and the centroid in D or leg tip positions in E (blue) for each frame are overlaid.

the leg tip position at time t, denoted by $\mathbf{l}_t[3]$ as it corresponds to the last control point of the leg snake. Since the distal tip of the leg may move considerably between successive frames, we designed a dedicated energy term to attract the tip toward candidate locations in the image. These candidate landmark locations are identified by filtering the image with a tip feature detector. A potential map P of points $\mathbf{p} = (p_x, p_y)$ that are tip candidates is then created according to

$$E_{\text{extremity}}(\Theta) = 1 - w_{\mathbf{p}^*} \mathrm{e}^{-\frac{\|\mathbf{l}_t[3]-\mathbf{p}^*\|^4}{\sigma^2}}, \tag{6.16}$$

where

$$\mathbf{p}^* = \underset{\mathbf{p}\in P}{\arg\min} \|\mathbf{l}_t[3] - \mathbf{p}\|^2 \tag{6.17}$$

is the tip candidate closest to $\mathbf{l}_t[3]$, $w_{\mathbf{p}^*} \in [0, 1]$ is its associated weight, and σ^2 is a fixed parameter determining the width of the basin of attraction of the tip candidate. The weight $w_{\mathbf{p}^*}$ is a measure of how tip-like $\mathbf{p}^*$ is, and is computed based on the magnitude of the response of the tip feature detector. A strong weight results in a deeper potential, and is therefore more likely to attract $\mathbf{l}_t[3]$. The energy (6.16) can be seen as a simplified version of our local feature-based energy. Here, landmarks are only used to attract control points. This is achievable because the curve is defined with linear B-splines. Control points therefore lie directly on the snake curve and can act as landmarks as in the Hermite spline case.

In summary, the four anchor points characterizing each leg are propagated as follows. First, the leg body anchors are determined using the body model. Second, the remaining three control points (two leg joints and tip) are updated by optimizing an energy that incorporates both image information (E_{feat} and E_{EDT}) and a temporal smoothness constraint (E_{segments}). Finally, the tip is attracted to nearby tip-like locations in the image. The complete tracking workflow is depicted in Figure 6.3.

Software and Data Availability

User instructions, FlyLimbTracker software, sample data, and a demonstration video of a complete data analysis pipeline can be found online[2].

[2] `http://bigwww.epfl.ch/algorithms/FlyLimbTracker/` and `https://doi.org/10.6084/m9.figshare.4688962.v1`.

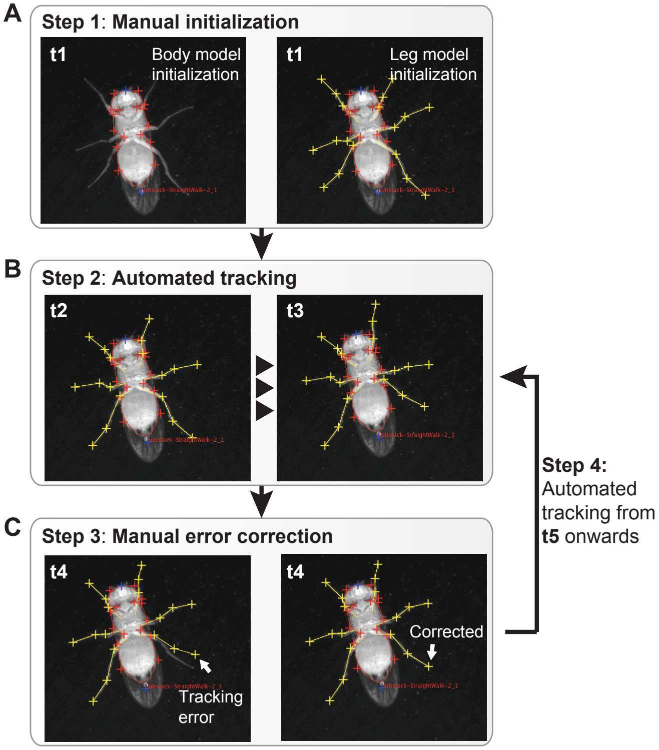

A
Step 1: Manual initialization
t1
Body model initialization
t1
Leg model initialization
B
Step 2: Automated tracking
t2
t3
C
Step 3: Manual error correction
t4
Tracking error
t4
Corrected
Step 4: Automated tracking from t5 onwards

Figure 6.3: FlyLimbTracker semi-automated body and leg tracking workflow. (A) The user manually indicates the approximate location of the body of the fly in an arbitrarily chosen initial video frame $t1$. FlyLimbTracker then optimizes a closed active contour that encapsulates the body of the fly in the correct orientation. The user manually indicates the location of the tip of each leg, and FlyLimbTracker optimizes an open active contour that outlines each leg. (B) These manual annotations are used to automatically propagate segmentation to prior or subsequent frames. (C) Either during or after automated tracking, the user can correct tracking errors. After manual editing, automated tracking can be resumed, taking into account and propagating user edits. In each image, the frame number is indicated.

Data Output. Once the full image sequence is annotated, data can be extracted as a CSV file for each fly. These measurements include the locations of the three reference points on the body of the fly (A, P, and 0), as well as each of the anchor points of the legs (see Figure 6.1 for details about the labeling convention). FlyLimbTracker is linked to the Track Manager plug-in of Icy (Publication Id: ICY-N9W5B7), allowing additional data to be extracted. In particular, segmentations of the body (Figure 6.2D) and legs (Figure 6.2E) of the fly can be visualized for the entire sequence at once, illustrating their complete trajectories. Each individual control point of the leg snakes as well as centroid of the body snake can be independently visualized. Tracks are also numbered according to the labeling convention described in Figure 6.1D.

Tracking multiple flies. FlyLimbTracker can track multiple flies in a single field of view. Additional flies are marked and tracked in a similar way as the first one. The tracking algorithm relies on a multithread implementation to avoid increased processing time. Since the location of the tracked flies is manually determined by user clicks in an initial frame, the presence of other objects in the field of view does not disturb the tracking algorithm as long as they do not occlude the flies. If occlusion occurs, the user can switch to manual mode to annotate problematic frames. Note that, from the design of the fly body model, each fly must be at least 10 pixels long and 8 pixels wide. The quality of segmentation and tracking strongly depends on resolution since higher resolution images contain more information. However, segmentation of large images also requires more computer memory and might thus be slower. This trade-off is further investigated and discussed in 6.1.2.

Cross-Platform Compatibility. Because the most memory-intensive step of the algorithm is image loading, FlyLimbTracker can be used in any computer capable of opening the image sequence of interest. Execution times are not expected to be strongly dependent on the operating system since the software is implemented in Java, a multiplatform language. Processing times reported in 6.1.2 can thus be used as a reference.

6.1.2 Results

Algorithm Robustness

FlyLimbTracker can be used to segment and track fly bodies and legs in videos spanning a wide range of spatial and temporal resolutions. Resolution determines the nature of the annotation process: tracking in high-resolution data is more automated, while low resolution data requires more user intervention. To quantify the dependence of computing time and number of user interventions on data quality, we systematically varied the spatial and temporal resolution of videos featuring five common *Drosophila* behaviors: walking straight (3 walking cycles using a tripod gait), turning ($> 90^\circ$ turn), foreleg grooming (3 leg rubs), head grooming (3 head rubs), and abdominal grooming (3 abdominal rubs). These five videos were derived from two longer movies: one movie of a fly walking straight and grooming its forelegs, and another movie of a different fly turning, grooming its head, and grooming its abdomen. Raw videos were originally captured at 236 fps and with a resolution of 2560×918 pixels.

First, we studied FlyLimbTracker's robustness to variations of spatial resolution. To cleanly isolate the effects of spatial resolution, instead of acquiring different datasets with lower resolution cameras, we downsampled each of the original five videos by a factor of N in space, meaning that $N \times N$ pixels patches were averaged into a single value. This resulted in image sequences that are N times smaller than the original one along both spatial dimensions but that have an identical temporal resolution of 236 fps (Figure 6.4A). Alternatively, to vary temporal resolution, we downsampled each video by a factor of N in time, meaning that only one frame in every N was retained. This resulted in image sequences of varying temporal resolution but consistently high spatial resolution of 2560×918 pixels (Figure 6.4B).

For each movie, body and leg snakes were manually initialized in the first image frame. Segmentation was then automatically propagated forward through the remainder of the image sequence. Whenever the automated tracker made a mistake, the process was interrupted and the user manually corrected the error. Automated tracking was then restarted from this frame until the next mistake was observed. In all cases, automated body tracking did not require any manual intervention. Therefore, we only took note of manual corrections in leg annotations.

The throughput of FlyLimbTracker can be determined by comparing the software results with fully manual annotation. Rather than the time spent annotating—

a metric that can dramatically vary between users—we quantify the number of required mouse clicks. Fully manual annotation of four control points for each of the six legs, and one additional click to move to the next frame, amounts to 25 mouse clicks per frame (24 clicks for the final frame). By contrast, FlyLimbTracker requires manual initialization in the first frame (four clicks to set up the body model and at least one click to initialize each leg model) and, in the worst-case scenario (head grooming), less than two corrections per frame with high spatial and temporal resolutions (Figure 6.4C, far left, 1× spatial downsampling). By conservatively assuming that there is one error in each frame, two mouse clicks per frame must be added to stop and then restart tracking. Therefore, using FlyLimbTracker it is reasonable to expect an average of four mouse clicks per frame. This is approximately a 6-fold increase in throughput when compared to fully manual annotation.

To quantify the performance of FlyLimbTracker across this range of spatial and temporal resolutions, we calculated two normalized quantities. First, we computed the average number of manual corrections per frame (Figure 6.4C and D). To do this, we measured the total number of user interventions needed to process an entire image sequence and normalized this quantity by the number of frames T, which contain each eighteen free parameters (six legs with three editable control points each). We also quantified the average time required to annotate a single image frame (Figure 6.4E and F). To do this, we recorded the total time required to annotate an image sequence and divided this value by the total number of frames. This normalized quantity combines both the computing time required for automated annotation, as well as the time required to manually correct annotation errors. Overall, we observed that reducing the spatial (Figure 6.4A, C, E) or temporal (Figure 6.4B, D, F) resolution resulted in an increase in the number of manual interventions (Figure 6.4C and D) as well as a longer time required for annotation (Figure 6.4E and F).

While the number of corrections was similar for videos with equivalent amounts of downsampling (up to 8-fold), annotation time was appreciably longer for sequences involving flies walking straight and turning. This reflects the importance of having overlapping images in successive frames for automated tracking, a feature that may be less common during locomotion. Notably, in a number of other cases (*e.g.*, grooming), the annotation time per frame flattens across spatial and temporal resolutions. This is probably due to the trade-off between resolution and speed. Resolution strongly influences the computing time required for automated tracking: smaller images or sequences composed of fewer frames are processed more quickly

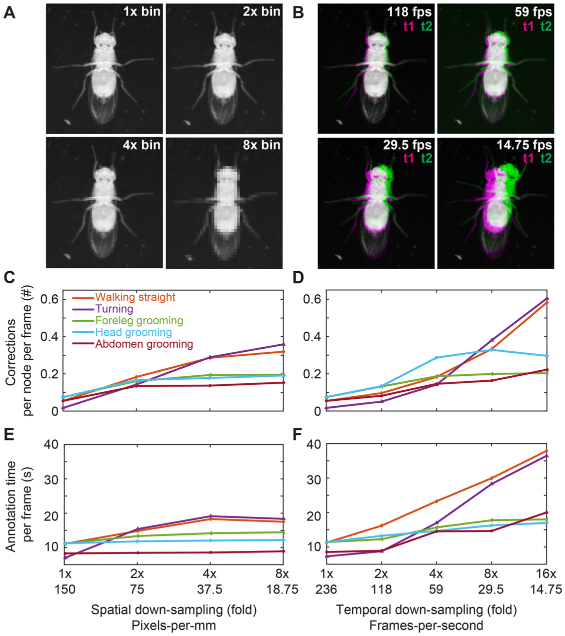


Figure 6.4: Sensitivity of leg tracking to changes in spatial or temporal video resolution. (A) Sample video image (top-left) after 2× (top-right), 4× (bottom-left), or 8× (bottom-right) spatial downsampling. Adult female flies are approximately 375, 187, 93, and 46 pixels in length in the 1×, 2×, 4×, and 8× spatial downsampled videos, respectively. (B) Representations of the difference between successive images in videos ($t1$ and $t2$ overlaid in magenta and green, respectively) after temporal downsampling. Number of required corrections per frame as a function of spatial resolution (C), or temporal resolution (D). Average time required to semi-automatically annotate a single frame as a function of spatial resolution (E), or temporal resolution (F). In (C) to (F), data for videos depicting a fly walking straight, turning, grooming its forelegs, head, or abdomen are shown in orange, purple, green, cyan, and red, respectively.

due to reduced demands on computer memory. However, a decrease in resolution also implies a reduction in the quantity of image information and an increase in the likelihood of errors. More user intervention is therefore required to correct mistakes. These interventions then in turn dominate the time required to annotate each frame. In summary, intermediate image resolutions are ideal for FlyLimbTracker since very low resolution images may require almost fully manual annotation while annotating very high resolution images can be prohibitively memory intensive.

Visualization and Analysis of Leg Segment Tracking Data

FlyLimbTracker provides a user-friendly interface that allows body and leg segment tracking data to be exported in a CSV file format, simplifying data analysis and visualization. Data interpretation and the number of experiments required to test statistical significance of biological observations are study- and experiment-dependent and are therefore out of the scope of this discussion.

First, within FlyLimbTracker itself, leg joint and/or body trajectories can be displayed overlaid upon the final raw video frame (Figure 6.5A1 to E1) using the TrackManager plug-in of Icy. This representation provides a way to project time-varying data onto a static image and illustrates the symmetric or asymmetric limb motions that control straight walking/grooming or turning, respectively. Second, leg segment trajectory data can be exported and processed externally using ad hoc Matlab or Python scripts. An example of such a script is provided in the FlyLimbTracker website. Second, these data can be rotated along with the frame of reference of the fly (Figure 6.5A2 to E2) for a direct comparison of leg segment movements across distinct actions. FlyLimbTracker permits to visualize previously inaccessible leg joints, as well as new, non-locomotive behaviors (*e.g.*, grooming or reaching) using a similar representation. In a third visualization, the speed of each claw is plotted to provide an exceptionally detailed characterization of locomotor gaits (Figure 6.5A3 to B3) or grooming movements in stationary animals (Figure 6.5C3 to E3). These are just a few examples of how tracking data can be analyzed. In addition, simple post-processing permits, for example, measurements of joint angles as well as the relative position of each joint with respect to any other annotated body part (*e.g.*, head, thorax, or abdomen).

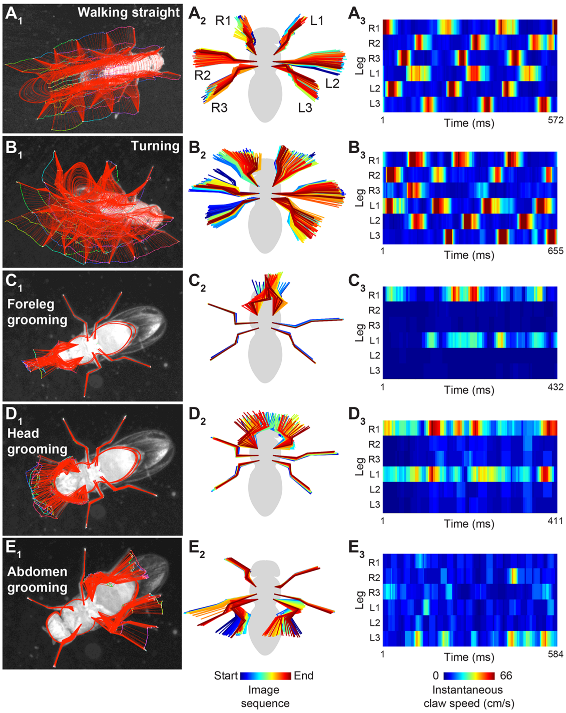


Figure 6.5: Visualizations of leg segment annotation results for videos of a fly (A) walking straight, (B) turning, (C) grooming its forelegs, (D) grooming its head, or (E) grooming its abdomen. First column: Leg segmentation results (red) and joint positions (color-coded by frame number) are overlaid in the final frame of the image sequence. Second column: Leg segment trajectories are rotated and color-coded by frame number. This alignment allows comparing leg movements across different datasets. Third column: Color-coded representation of the instantaneous speeds of each leg tip (claw).

6.1.3 Discussion

Existing methods for tracking insect leg segments rely on sophisticated optical equipment and/or laboriously-applied leg markers, often in tethered animals [155, 156, 157]. While these approaches are extremely valuable, they may potentially disrupt natural behaviors and cannot report the motions of multiple leg joints in untethered animals. Our method addresses these technical limitations. FlyLimbTracker only requires a single high-resolution, high-speed camera and no prior marking of leg segments. It can be used with video data across a range of spatial and temporal resolutions, permitting a flexible blend of automated and manual annotation. Importantly, when automation has difficulty segmenting low quality data, FlyLimbTracker remains a powerful tool for manual leg annotation since it relies on spline-snakes that can be easily manipulated and provides an interface for user-friendly data import and export. Exported data (*i.e.*, fly limb and body position in image coordinates) can serve as basis for computing a range of statistics describing fly motion. Of course, the nature of the statistics of interest depend entirely on the considered experimental setting and biological problem.

The open-source nature of FlyLimbTracker facilitates community-driven improvement and customization of the software. We envision a number of improvements that may be implemented. First, tracking currently requires overlap of the body of the fly in successive frames. This constraint places a lower bound on temporal resolution and could be improved using, for example, nearest-neighbor matching approaches to pair control points in successive frames. Second, additional leg control points may be added to FlyLimbTracker to more precisely annotate thorax-coxa-trochanter segments. The current system with a single camera projects three-dimensional joint position data onto two dimensions, making it difficult to accurately measure every joint angle. Additional camera views could allow full three-dimensional reconstruction of each limb's orientation and position in space. Third, FlyLimbTracker's requirement of user initialization makes it only semi-automated and restricts batch processing of multiple videos for high-throughput data analysis. This may be overcome using additional prior information to automatically identify and optimize the fly body and leg snakes. Fourth, the snake-based approach in FlyLimbTracker could easily be adapted for the study of other species (*e.g.*, mice, stick insects, and cockroaches) by modifying the prior shape imposed on the snake curves.

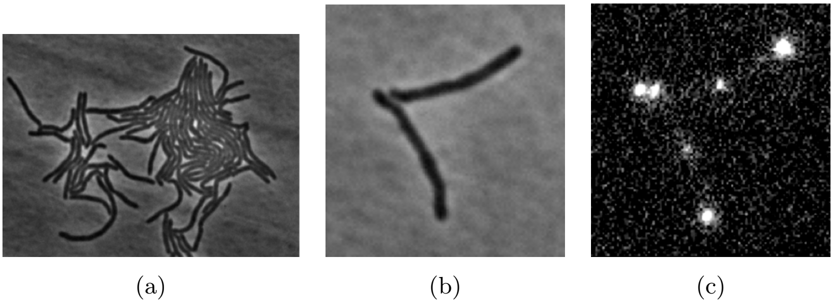
6.2 Detection and Tracking of Mycobacteria using Hermite Splines

Mycobacteria are the pathogenic agents responsible for tuberculosis, one of the top killing infectious diseases, especially in third-world countries [170]. Curing infections caused by *Mycobacterium tuberculosis* requires long-term antibiotic treatments in order to completely eradicate bacterial colonies. On top of that, the presence of mutant strains makes it necessary to alternate between different antibiotic compounds, further complexifying the curing process. In practice, irregular antibiotic intake and extended pauses in the treatment often lead to a fatal evolution of the disease [171]. Improper antibiotic treatments also favor the development of multi-resistant mutant strains, calling for the discovery of new chemical compounds that either kill or restrict the growth of mycobacteria. Studying the behavior of mycobacteria is the essential initial step towards a better understanding of the mechanisms of resistance and for investigating potential drug candidates. Basic parameters of interest include growth rate, interdivision time, and length at division among others. More advanced ones include all information coming from fluorescent protein reporters, ranging from quantifying the expression of a protein to monitoring its cytoplasmic localization over the cell cycle. For long, biological studies relied on observations at level of the cell population, or colony. It is however now known that cell populations are not normally distributed, neither at the genetic nor at the phenotypic level [172]. Significant individual cell-to-cell variations are observed between bacteria coming from a same colony, even when derived from a single ancestor [173]. The multimodal nature of cell populations rules out the use of colony-based statistics, which assume Gaussian or uniformly distributed individual characteristics. Analysis at the single-cell level is thus required in order to be able to identify small subpopulations of mutants. Also, the subcellular organization of bacteria is often dependent on the local environment and can undergo sophisticated spatial variations, further reducing the legitimacy of global population statistics. The most common raw material for microbiology studies are sequences of images of bacterial colonies growing and evolving over time in different experimental conditions and acquired with various microscopy modalities (fluorescence, brightfield, phase contrast). From these, individual cells are segmented (outlined) and tracked (followed over time) in the image data. It is required to rely on large enough datasets to produce statistically meaningful observations, meaning that the amount of data to process can be huge [174]. In addition, quantitative and reproducible results are needed to be able to capture possibly small variations in an objective way. For these two reasons, relying on manual, human-based efforts to perform this work is not only technically intractable for time and resources reasons, but also not desirable.

A plethora of generic softwares for image analysis of bacteria have been proposed. Among them, MicrobeJ [175], MicrobeTracker [176], Oufti [177], ObjectJ [178], Schnitzells [179], and MAMLE [180] can be mentioned as popular in the biology community. All of these methods have been mainly developed and tested for *E. coli*, *B. subtilis* or *C. crescentus*. These bacteria exhibit low variations in shape and size among individuals, exhibit clear boundaries and contours, or both at the same time. None of these aspects are present in mycobacteria, making existing approaches of limited to no use.

Automating the tracking of mycobacteria poses two main challenges, illustrated in Figure 6.6. First, the nature of the cells and of the colony they form (*e.g.*, tightly packed with bad contrast between neighboring cells, absence of strong shape prior as cells can adopt complicated morphologies) calls for the development of a method that solves the problem globally. Local ambiguities make it necessary to pick information from both upstream and downstream of a given frame to provide a solution in which errors are not propagated. The presence of many similar and closely located individuals calls for a globally optimal solution in order to avoid introducing order dependencies. Second, division events are non-standard: they are not visible on phase contrast data as cell division occurs much earlier than cell body separation [181]. One must rely on the fluorescence signal of a protein located at the cells' poles (*wag31* [181]) in order to accurately segment the cells. Complete segmentation thus requires the incorporation of joint information from signals of both the phase contrast and fluorescence channels. So far, the only published approach specifically dedicated to mycobacteria is BactImAS [182], a Java-based image analysis tool. It however has several strong limitations, including the fact that it can only incorporate information from phase contrast images, thus leading to imprecise division identification, and that it is mostly manual as all division events have to be manually annotated. Tracking is moreover performed frame-by-frame in a greedy manner, introducing order dependency and propagating errors as they occur.

Hereafter, we present an automated pipeline to analyze time-lapse microscopy images of mycobacteria. We propose a three-step approach relying on graphical



(a) (b) (c)

Figure 6.6: Challenges when analyzing images of mycobacteria. (a) Sample phase-contrast image illustrating poorly defined cell boundaries and high variety of shapes in term of length and bending. (b) Sample image of *Mycobacterium smegmatis* in phase-contrast and (c) fluorescence microscopy. The *wag31* protein, gathering at cell extremities, indicates that four cells are present in this image, although only two can be distinguished from the phase-contrast data. This is due to the fact that cytokinesis happens much earlier than physical cell separation when mycobacteria cells divide.

models for joint detection and tracking. In the first stage, we process each frame of the image sequence to generate a collection of candidate bacteria, or detection hypothesis. We link detections from successive frames to model cell transition and division, and obtain a graph. The problem annotating the sequence boils down to selecting the correct detections, transitions and divisions, considering the probabilistic graphical model whose nodes represent random variables with features describing the probability of each detection, transition or division event. The solution of the joint hypotheses selection and tracking problem is found as the optimal labeling of the graph based on these probabilities, as schematically represented in Figure 6.7. Such approaches have already been successful applied in similar problems [183, 184, 185]. The graphical model formulation allows solving the problem globally: the graph contains much more detection and linking possibilities than needed. They are then discarded or kept considering the information of the sequence as a whole. As image frames do not get processed sequentially, the need for taking decisions based on ambiguous frames is obviated and no order is imposed on the assignment of individual tracks. It is, to the best of our knowledge, the first automated approach addressing the image analysis challenges that are unique to mycobacteria. From its global spirit, our approach can be related to that of [186], where a Viterbi-like algorithm is used to track *E. coli* in an order-independent manner. The main differences lie in the already-mentioned particularities of mycobacteria, such as non-conventional division events and the impossibility to impose strong priors on shape and size.

This second practical bioimage analysis problem[3] illustrates a direct use of our landmark snake model. As we shall see shortly, candidate bacteria are represented by open spline curves.

6.2.1 Description of the Method

Time-lapse Sequences of Mycobacteria

Our data come from Prof. John McKinney's Laboratory of Microbiology and Microtechnology (LMIC) at EPFL, Lausanne, Switzerland. They feature *Mycobacterium smegmatis*, the classically used model for mycobacteria studies. The sequences are generally composed of two channels with 100 frames of 512×512 pixels

[3] This work has been carried out in collaboration with C. Haubold and Prof. F. Hamprecht, Heidelberg Collaboratory for Image Processing (HCI), University of Heidelberg, Germany.

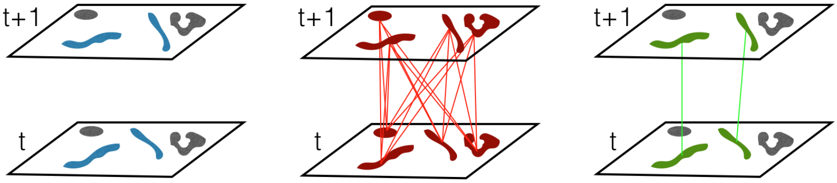


Figure 6.7: A schematic explanation of joint segmentation and tracking using graphical models. The original sequence features objects of interest (blue) and artifacts (grey). A set of detection candidates (dark red areas) and transition candidates (red lines) is generated. For the sake of simplicity, division events are here omitted. The set of detections and transitions is built in a very conservative way and therefore includes many wrong elements. All detections and transitions can take an on- or an off- state, with some attached probabilities. The solution is obtained by identifying the collection of states for each element in the set of transitions and detections that maximizes the overall probability. The result is visualized by retaining only elements that are in on-state (green) in the solution.

16 bits .tif images.

Image Analysis Pipeline

The pipeline we propose for the automated tracking of mycobacteria can be roughly decomposed into three main steps.

1. Generating detection hypotheses by identifying bacteria candidates.

2. Building a graphical model by linking detection hypotheses in consecutive frames through transition and division hypotheses if they are spatially close enough. Features are also extracted for all hypotheses and combined to train a per-hypotheses classifier, which is then used to reduce the dimensionality of the problem by pruning out too implausible hypotheses.

3. Finding the globally most probable configuration by solving an integer linear program (ILP) [187].

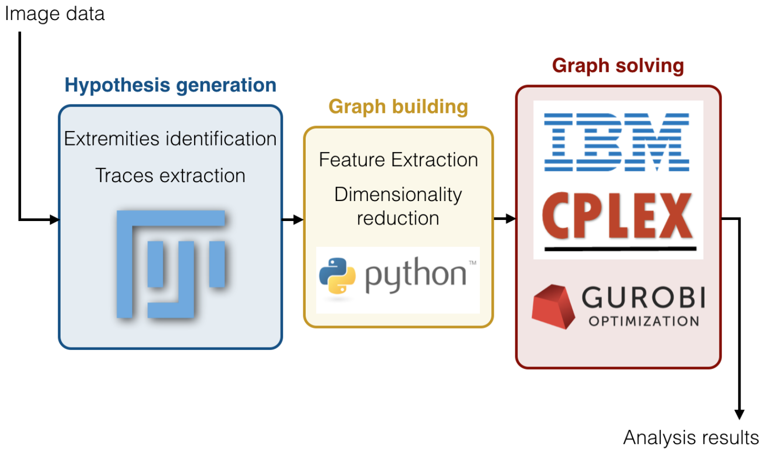


Figure 6.8: Schematic view of the image analysis pipeline. First, images are analyzed in Fiji or ImageJ in order to extract detection candidates. Then, the graphical model is constructed and the globally optimal solution is finally found relying on commercial ILP solvers.

A schematic view of the pipeline is provided in Figure 6.8. Hereafter we describe each step in detail, assuming that trained classifiers and tuned weights are available.

Extraction of Detection Hypotheses. In order to generate detection hypotheses, two sources of information are available for each time frame: the fluorescence channel, featuring blobs which correspond to cell extremities, and the phase contrast channel, where cell contours can be identified (Figure 6.9). These two sources of information must be used together to obtain a full segmentation of every cell. To simplify the problem, we search for the medial axis of each bacteria, which we refer to as a *trace* (Figure 6.10). The rationale behind this design choice is twofold. First, it makes it easier to handle information from both channels, as the bacteria identification problem is reduced to finding a meaningful path in the phase channel image between two locations corresponding to blobs in the fluorescence channel image. Second, traces provide a compact way of representing the complete cell. As mycobacteria grow mostly at the extremities and have a rather conserved cell

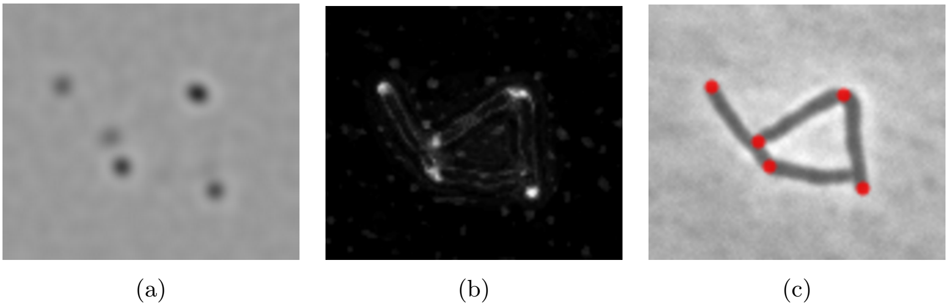


Figure 6.9: Dot selection pipeline. (a) Fluorescence channel image filtered with a Laplacian of Gaussian, (b) tip probability map from a two-stages autocontext Ilastik workflow, and (c) candidate extremities (overlaid as red dots) obtained by combining detections from the two previous sources and applying non-maxima suppression.

width, a good estimate of the cell contour can be obtained by searching for area of strong gradient around the trace and completing by interpolation in areas where no boundaries are available.

As a first step, using the open-source image processing software Ilastik [13], we interactively train a per-pixel Random Forests classifier [188] on one dataset to identify mycobacteria-looking parts of the images. This gives us a probability map for every frame in every dataset, indicating how likely each pixel is to belong to a bacterium. We refer to it as the *foreground probability map.* Similarly, using Ilastik, we also train a two-stage autocontext [189] pixel classification workflow to detect end-point-looking parts of the images. This results in a tip-looking probability map for every frame in every video, which we call *tip* or *endpoint probability map.* Each of the two probability maps alone is not sufficient to properly resolve the bacteria. For each frame, the three following steps are then performed.

1. *Identifying cell extremity candidates.* Cell extremity candidate locations are extracted from the fluorescence channel filtered by a Laplacian of Gaussian (*i.e.*, a blob detector) and from the endpoint probability map. Relying on these two sources increases the robustness of the algorithm. The fluorescence channel is required to identify divisions as long as cells have not yet physically

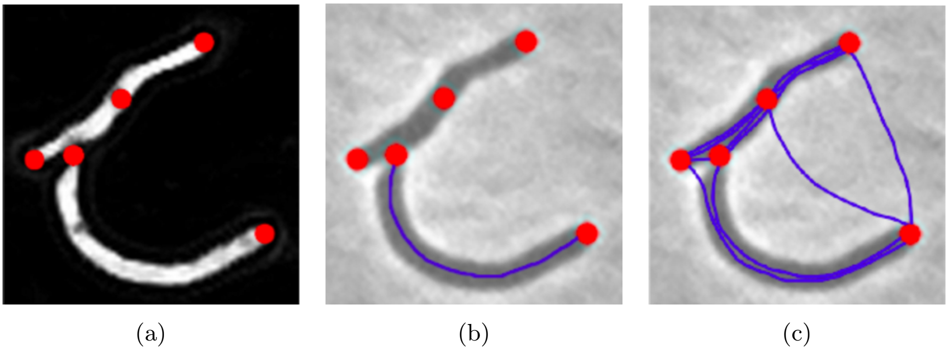


Figure 6.10: Traces generation pipeline. (a) Identified extremity locations overlaid as red dots on the foreground probability map. (b) Extremity locations are used as seed points for dynamic programming (DP) on the foreground probability map. An open landmark snake is fitted for the resulting DP path and optimized in order to obtain a smoother and more precise trace. (c) The procedure is repeated between all pairs of dots in the frame.

separated, and the probability map is needed to avoid losing extremities as the fluorescence signal of old tips decays over time. These two images are thresholded to identify strong local maxima of intensity, and detections from the two sources are combined while applying non-maxima suppression.

2. *Linking all pairs.* Dynamic programming is performed on the foreground probability map obtained from the phase contrast image to find the shortest path between each pair of endpoints. The dynamic programming graph is built as described in the *Drosophila* leg model of 6.1.1. The cost of a path is given as (6.11), where I_{feat} is replaced by $I_{\text{foreground prob}}$, the negative (*i.e.*, image inverse) of the foreground probability map. To increase the robustness of this search for the medial axis, a collection of M-diverse shortest paths is generated for each possible pairs of points following the approach we introduced in [190]. The actual value of M (the number of shortest paths) is not fixed *a priori*. Instead, the next best path subject to some diversity constraints keeps on being computed and its cost is monitored. The search is stopped when the cost of the most recently found Mth path differs by a too

large extent from the cost of the previously found $(M-1)$th path. In our implementation, the search for shortest paths continues as long as the cost of the newly found path does not exceed 10% of that of the previously found path.

3. *Refining traces.* We fit an open landmark snake with loose ends following (5.3), using the dynamic programming result as initialization. Using a snake model based on cubic Hermite splines instead of, *e.g.*, cubic B-splines, is preferable as the open nature of the curves make it necessary to handle boundary conditions. While cubic B-splines involve complicated constructions involving the introduction of artificial control points at the extremities of the curve, no difficulty arises with Hermite splines when considering open curves instead of closed ones. Our motivation for using a spline-snake here is two-fold. First, it refines the trace. We optimize the snake allowing only its endpoints (first and last control points) to move in order to fine-tune the localization of cell extremities. This corrects errors that might have been introduced when representing the blobs detected in the fluorescence channel with a single point location. To do so, an endpoint energy is computed as the integral of a circular area around the control points at the extremity of the snake curve on the endpoint probability map. The rest of the curve remains fixed, while the extremities search for the most tip-looking location around them. Second, the continuous spline curve representation offers the possibility to compute interesting features describing the bacteria.

Construction of the Graphical Model. Based on the extracted detections, we construct a factor graph [191], which is a particular type of probabilistic graphical model solved by relying on the factorization of individual terms. The graph covers all time frames of the input sequence. It is set up as a trellis graph in which each column contains nodes representing the detections of a specific time frame. Consecutive columns thus model subsequent time frames. Transition and division hypotheses are inserted between those columns as separate nodes connected to two or three detection nodes, respectively. Transition and division hypotheses outgoing from any detection i at time t are determined by considering the 20 detections j in the following frame $t+1$ that are the closest in terms of overlap. The overlap is estimated as the Jaccard index (5.23) between dilated versions of the spline traces, which provide good approximations of the full bacteria body. The whole process of extracting detections and generating transition and division hypotheses is illustrated in Figure 6.11.

The factor graph is composed of random variables for each detection, transition and division hypothesis, hard constraints and factors. Hard constraints allow to model requirements for a consistent solution. We use them to ensure flow conservation and mutually exclude detection hypotheses in the same frame based on whether and to which extent they cross and overlap. Factors encode local probabilities. In our model, the variables are binary and can take two possible states. A state of 1 implies that the hypothesis is conserved in the final tracking solution, while a state of 0 means that it is discarded. The factors attached to each of these nodes encode the probability for using (state 1) or not (state 0) the corresponding hypothesis. These values are obtained by first computing an extensive set of features for each detection, transition, and division hypothesis, and then relying on Random Forests classifiers to predict, based on these features, the probability for the hypothesis to be valid or not. Classifier predictions non-linearly combine features of possibly different dynamic ranges into a single value that can be readily interpreted thanks to its probabilistic nature. Because the initial sets of hypotheses are large enough to ensure that no correct solution is missed, we also use the predicted probabilities to prune out too implausible hypotheses. To do so, we apply very conservative thresholds and discard from our model detections with probabilities lower than 1%, as well as transitions and divisions with probabilities lower than 0.01%.

Optimal Tracking Solution. To formally enunciate the optimization problem that leads to the optimal tracking solution, let us first define the factor graph more precisely. As previously stated, all detection, transitions, and division hypotheses are represented by nodes. Nodes corresponding to divisions are connected to one detection at time t and two detections at time $t+1$, and nodes corresponding to transitions connect one detection at time t to one detection at time $t+1$. Detections can appear and disappear, which is modeled by special transition nodes with one unconnected endpoint, which could as well be connected to a virtual source or sink. Hence, in the equations below, transitions also include appearances and disappearances. Configurations of the graph are encoded in the form of a state vector $\mathbf{x}$, composed of as many binary components as there are nodes in the graph. The state vector thus contains one component for each detection, transition and division hypothesis. Each binary components of $\mathbf{x}$ indicates the state of the corresponding

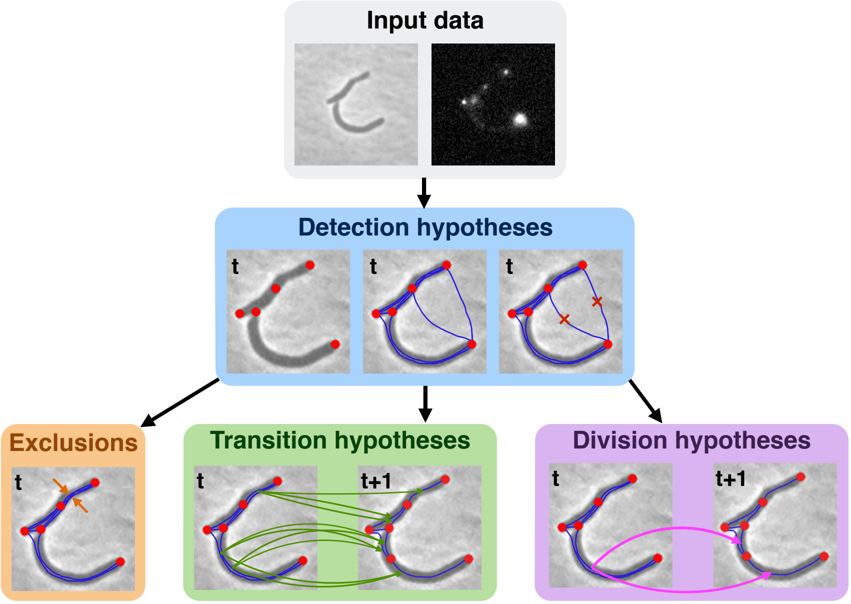


Figure 6.11: Pipeline for building the graphical model. Detection hypotheses are first extracted by linking the cell extremity candidates. From there, based on the amount of overlap and crossing between detection hypotheses, exclusion constraints forbidding the simultaneous use of incompatible detections are introduced. Then, transition hypotheses are generated between close pairs of detections in the successive frame. Finally, division hypotheses are constructed in a similar way between cells in a given frame and close-by pairs of cells in the successive frame.

detection, transition, or division node in the graph. We use sub- and superscripts to address the state of individual variables in $\mathbf{x}$ such as $x_{t,i}^{\mathrm{det}}$ for the state of the detection hypothesis i in frame t. The solution is then retrieved as the subset of nodes that are active in the state vector.

Any detection node i in any frame t is attached to a factor $\theta_{t,i}^{\mathrm{det}}$, which contains the cost of using ($x_{t,i}^{\mathrm{det}} = 1$) or not ($x_{t,i}^{\mathrm{det}} = 0$) this particular detection in the final solution. Factors are constructed similarly for transition and division nodes, with the difference that they respectively require 2 and 3 indices to be uniquely characterized. The probabilities predicted by the Random Forest classifiers are transformed into energies by applying a negative logarithm, which turns the product over all factors constituting the joint probability into a sum over energies. Thus, factors all take a form similar to

$$\theta_{t,i}^{\mathrm{det}}(x_{t,i}^{\mathrm{det}}) = -w^{\mathrm{det}} \log\left(\mathrm{P}^{\mathrm{det}}(x_{t,i}^{\mathrm{det}} = 1)\right), \tag{6.18}$$

with w^{det} a weighting term. The global objective function can then be formulated as

$$\begin{aligned} \min_{\mathbf{x}} \sum_{t=0}^{T} \sum_{i \in \mathcal{X}_t} \theta_{t,i}^{\mathrm{det}}(x_{t,i}^{\mathrm{det}}) + \sum_{t=0}^{T-1} \sum_{i \in \mathcal{X}_t} \sum_{j \in \mathcal{X}_{t+1}} \theta_{t,i,j}^{\mathrm{trans}}(x_{t,i,j}^{\mathrm{trans}}) \\ + \sum_{t=0}^{T-1} \sum_{i \in \mathcal{X}_t} \sum_{j \in \mathcal{X}_{t+1}} \sum_{\substack{k \in \mathcal{X}_{t+1} \\ k \neq j}} \theta_{t,i,j,k}^{\mathrm{div}}(x_{t,i,j,k}^{\mathrm{div}}), \end{aligned} \tag{6.19}$$

such that

$$\forall t, \forall i \in \mathcal{X}_t : x_{t,i}^{\mathrm{det}} - \sum_{j \in \mathcal{X}_{t+1}} x_{t,i,j}^{\mathrm{trans}} - \sum_{j \in \mathcal{X}_{t+1}} \sum_{k \in \mathcal{X}_{t+1}, k \neq j} x_{t,i,j,k}^{\mathrm{div}} = 0 \tag{6.20}$$

$$\forall t, \forall i \in \mathcal{X}_t : x_{t,i}^{\mathrm{det}} - \sum_{j \in \mathcal{X}_{t-1}} x_{t,i,j}^{\mathrm{trans}} - \sum_{j \in \mathcal{X}_{t-1}} \sum_{k \in \mathcal{X}_{t}, k \neq i} x_{t,j,i,k}^{\mathrm{div}} = 0, \tag{6.21}$$

and

$$\forall i, j \in C_t : x_{t,i}^{\mathrm{det}} + x_{t,j}^{\mathrm{det}} \leq 1, \tag{6.22}$$

where $\mathcal{X}_t$ denotes the set of all detection hypothesis at frame t. Ensuring in (6.20) that, given a detection, its value equals the sum of active outgoing transitions and divisions enforces flow conservation and guarantees that parent cells are not involved in more than one division at the time. Analogously, it must be ensured that every detection value also equals the sum of incoming active hypotheses, which is achieved through (6.21). Finally, the exclusion constraints are imposed in (6.22), where C_t denotes the set of mutually exclusive pairs of detections in frame t.

Since each individual term $\theta_i(x_i)$ can be written as

$$\theta_i(x_i) = \begin{cases} -w \log\left(\mathrm{P}(x_i = 1)\right) & x_i = 1, \\ \log\left(\mathrm{P}(x_i = 0)\right) & x_i = 0, \end{cases} \tag{6.23}$$

the optimization problem (6.19) is an instance of an ILP. Such problems can be solved by the commercial solvers Gurobi and CPLEX[4].

6.2.2 Results

We gathered results from our joint segmentation and tracking pipeline on five datasets for which a manually generated ground truth was available. Each dataset is composed of 100 image frames. Sequences always begin with a single bacteria in the field of view and end with approximately 30 individuals on the last frame.

Precision and recall scores of the segmentation and tracking results are provided in Table 6.1. Precision is defined as the ratio between the number of true positives to the total number of positives (*i.e.*, true and false altogether) and recall, or sensitivity, as the ratio between the number of true positives to the total number of theoretically correct hypotheses (*i.e.*, true positives and false negatives altogether). For all datasets, we observe that precision is generally close to perfect while recall scores remain slightly behind, although being also generally good. It means that our pipeline typically tends to miss correct hypotheses rather than selecting erroneous ones.

To get a better sense of the quality of these results, we also consider the acyclic oriented graphs matching (AOGM) measure described in [192]. The core idea behind this metric is to evaluate the resemblance between two joint segmentation and tracking result graphs. Considering a first tree corresponding to ground truth annotation and a second one corresponding to the results to be evaluated, the AOGM measures how much editing operations are required to turn the results tree into

[4] In our experiments, we rely on CPLEX.

Table 6.1: Precision and recall scores for five mycobacteria datasets.

Dataset ID		3	4	6	10	12	13	16
Detections	*Precision*	1	0.99	1	1	1	1	1
	Recall	0.97	0.98	0.99	1	0.99	0.93	0.99
Transitions	*Precision*	0.99	0.99	1	1	1	1	1
	Recall	0.97	0.97	0.98	0.99	0.98	0.91	0.99
Divisions	*Precision*	1	0.96	1	1	1	1	1
	Recall	0.88	0.96	1	0.96	0.85	0.77	1
Overall	*Precision*	0.99	0.99	1	1	1	1	1
	Recall	0.97	0.97	0.98	0.99	0.99	0.92	0.99

the ground truth one. As a first step, computing the AOGM requires to match the results and ground truth detections at each frame. The ith ground truth detection at frame t, denoted as $d_{t,i}^{\mathrm{GT}}$, is considered as matched to the jth results detection at frame t, designated as $d_{t,j}^{\mathrm{res}}$, if

$$|d_{t,i}^{\mathrm{GT}} \cap d_{t,j}^{\mathrm{res}}| > \; 0.5\, |d_{t,i}^{\mathrm{GT}}|. \tag{6.24}$$

An important consequence of this criterion is that each ground truth detection can be assigned to at most one results detection, whereas one results detection can have multiple ground truth detections assigned to it. From there, the following types of errors are considered. First, the missed detections, or false negatives, are denoted as FN. Similarly, the false detections, or false positives, are designated by FP. The missed transitions are denoted as EA since correcting them requires an *add edge* operation in the results tree, and the wrong transitions are labeled as ED since fixing them involves a *delete edge* operation. The detections that have more than one ground truth detection assigned are referred to as NS for *non-split vertices.* Their count is obtained as the difference between the number of true positives (TP) and matched detections (MD). Finally, the erroneous divisions are referred to as EC since editing them requires an *edge correction* operation in the results graph. The AOGM is computed as

$$\begin{aligned}\mathrm{AOGM} =& w_{\mathrm{FN}} C_{\mathrm{FN}} + w_{\mathrm{FP}} C_{\mathrm{FP}} + w_{\mathrm{EA}} C_{\mathrm{EA}} \\ &+ w_{\mathrm{ED}} C_{\mathrm{ED}} + w_{\mathrm{NS}} (C_{\mathrm{TP}} - C_{\mathrm{MD}}) + w_{\mathrm{EC}} C_{\mathrm{EC}},\end{aligned} \tag{6.25}$$

Table 6.2: AOGM scores for five mycobacteria datasets.

Dataset ID		3	4	6	10	12	13	16
Detections	*Missed*	0.033	0.010	0.007	0	0.075	0.006	0.001
	False	0.005	0.001	0.001	0	0.012	0.003	0
Transitions	*Missed*	0.040	0.018	0.014	0	0.101	0.013	0.003
	False	0	0.004	0.007	0.003	0.007	0.007	0.001
Divisions	*Missed*	0.154	0.037	0	0.035	0.258	0.238	0
	False	0.087	0.074	0	0	0.042	0.111	0
AOGM		285.5	111.5	81	2	835	58	14

where the $w_i \geq 0$ are weights and the C_i correspond to counts of the number of occurrences of events of type i. The percentage of missed (*i.e.*, FN) and false (*i.e.*, FP) detections, transitions and divisions, as well as the resulting AOGM scores, are given in Table 6.2. We have set the weights values as suggested in [192], namely $w_{\mathrm{NS}} = 5$, $w_{\mathrm{FN}} = 10$, $w_{\mathrm{FP}} = 1$, $w_{\mathrm{ED}} = 1$, $w_{\mathrm{EA}} = 1.5$, $w_{\mathrm{EC}} = 1$. This particular choice of values has been demonstrated to faithfully represent the manual workload required to fix each type of errors. The AOGM measure is then a good indication of how tedious the manual editing would be for a user who would correct the automatically obtained results. The AOGM measure is bounded from below by zero and reaches this value when the ground truth and results graphs are perfectly identical for the type of errors penalized by non-zero weights. It has no upper bound and can thus technically grow to infinity. Smaller AOGM values indicate results that are closer to the ground truth.

In all studied datasets, the amount of both missed and false events lie below 10% for detections and transitions. Division events obtain inferior scores, with as much as 25% missed divisions in some cases. It is however worth noting that the estimation of division error is exaggerated. Here, a division event detected one frame before or after the corresponding one in the ground truth is considered as missed, although it remains acceptable in practice. This translates to high FN scores. Comparatively, the amount of FP divisions lies in the same range as those of detections and transitions. Most of the division errors therefore actually correspond to slight mismatches in the time point where the division is identified. This is confirmed by visual inspection of the results. In Figure 6.12, we illustrate examples of end results in more or less crowded bacteria colonies. Individual mycobacteria are properly identified even when they are in close contact and exhibit no clear boundaries. The joint segmentation and tracking approach indeed relies on information from larger portions of the sequence (in fact, on the sequence as a whole) to resolve situations that are ambiguous when considering isolated frames. This mimics the natural approach adopted by humans, who tend to check frames before and after the one being processed to resolve ambiguities when manually annotating a sequence.

6.2.3 Discussion

Although its complete execution time might range from hours to a few days, the segmentation and tracking pipeline we designed is fully automated in the sense that no manual intervention is needed to process a dataset. Edits might however be required after the processing to correct erroneous results. The correction process is made intuitive through a dedicated interactive ImageJ plug-in.

Properly trained Random Forests as well as appropriate weighting terms w^{det}, w^{trans}, and w^{div} for the different types of factors (see (6.18)) were here assumed to be available. In practice, these ingredients must be carefully tuned from examples of annotated data so as to benefit from the full potential of the pipeline. To train the Random Forests used to turn the set of features for every detection, transition, or division hypotheses into a probability, positive and negative training examples must be provided. In our case, we used all samples from the training datasets. The first motivation for this choice is the large variety of cell density (*i.e.*, colony size) over the course of the sequences, which precludes the selection of few isolated time frames. The second reason is an attempt to increase the generalization power of the classifier through datasets variability, reducing the risk of batch effects [193]. The optimal weighting terms are searched by training a structured support vector machine (SSVM) [194] with maximal margin, similar to [195, 196].

The next experimental step consists in assessing the predictive power of the classifiers we rely on. This involves running the pipeline on additional, untested data. Having been designed to yield good results on our five training datasets, our hope is that the detection, transition and division classifiers as well as the weight values are general enough to accommodate sequences of similar nature. At the time of writing, these validation experiments were still under preparation. In the unfavorable event that the general classifiers fail to give sensible predictions on new datasets, additional user intervention would be needed in order to provide

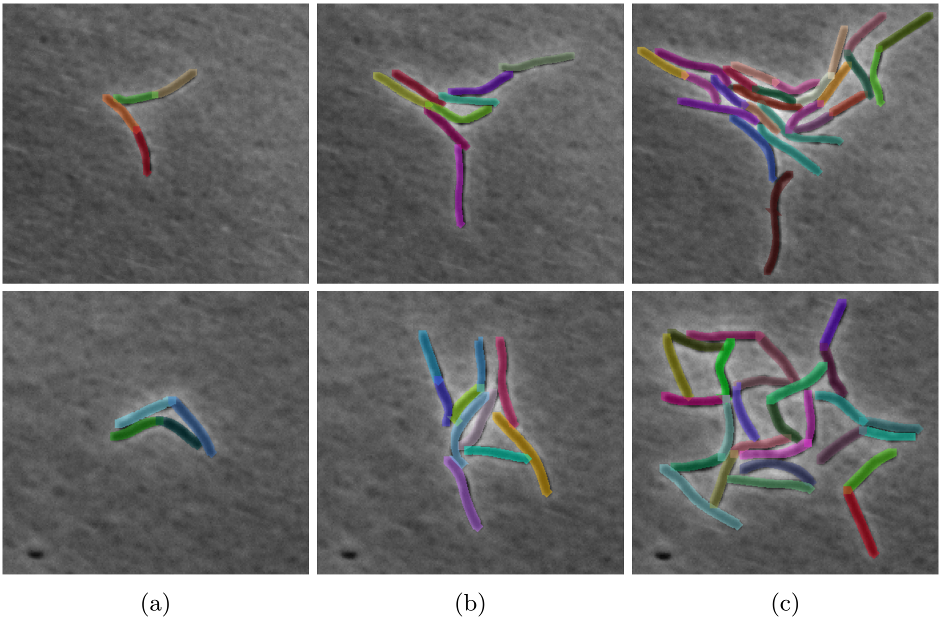


Figure 6.12: Example of results for dataset 6 (top row) and 16 (bottom row) for (a) small colonies corresponding to the first 50 frames of the sequences, (b) mid-sized colonies corresponding to the frame range 50-80, and (c) crowded environment towards the last frames of the sequences. The colors have been chosen so as to obtain good color contrast.

training examples for detections, transitions and divisions. The same ImageJ plug-in designed to edit segmentation and tracking results was also designed to be used for this purpose. It allows the user to select correct hypothesis among the set of all generated ones in a fast and user-friendly way. The minimal amount of training examples that should then be provided and, therefore, the minimal amount of annotation required to obtain satisfying results should be the topic of future studies. Although very encouraging, the results we provide here should thus be seen as a first step towards a fully automated solution for mycobacteria tracking. More validation is still needed in order to assess the generalization power of the approach and make it practically usable by the biologists.

6.3 Study of *C. elegans* Dynamics using Landmark Active Contours

The roundworm *C. elegans* has emerged as an attractive model organism in biomedical research, as it offers one of the best compromises between the simplicity of cell models and the complexity of vertebrate models [197, 198]. Some of the characteristics of this invertebrate model that contributed to its success include its sequenced genome that is easy to manipulate, its invariant and fully-described development, its relatively easy maintenance, its short and prolific life-cycle, and its small body size [199]. The nematode has additional favorable attributes, including the ability to generate large numbers of recombinant offspring, the ability to preserve and revive strains indefinitely through cryo-freezing, and the availability of ample phenotypic variations. These features, along with the conservation of many disease and stress response pathways between *C. elegans* and humans, have spurred the use of *C. elegans* in biomedical research, including but not restricted to the field of neuroscience. There, the study of *C. elegans* locomotion is crucial for many biological problems. Several solutions have already been proposed based on measures of deformation using geometrical features [200], on the construction of dictionaries, or on feature extraction and subsequent statistical analysis or classification [201, 202]. Most existing approaches rely on separation of background and foreground by thresholding and require an *a posteriori* reparameterization of the detected shape, as well as extra processing steps for identifying the orientation of the nematodes.

C. elegans locomotion is a particularly well-suited application for our landmark snake as nematodes feature pointy extremities and exhibit a variety of curved shapes. Although initially designed for static image analysis, our parametric model can be adapted to the problem of analyzing nematode motility. From the landmark snake construction, the segmentation outlines are automatically parameterized with landmarks matching the extremities, which facilitates further shape analysis and feature extraction. It thus provides a good example benefiting from joint feature detection and contour extraction. We present in the following our image analysis framework based on landmark active contours for nematode motility assays[5]. A novelty that stems from our approach is the formulation of a continuous shape-space, which allows to decompose nematode motion into a collection of "modes", or *eigenworms*, in the spirit of [203] to gain insight into the nature of the different locomotion patterns present in a dataset. As we shall see shortly, this is uniquely feasible thanks to the Hermite spline representation of the nematode contours. We present the derivation of the shape space and propose an efficient way of implementing it, before showing results on real datasets.

6.3.1 Description of the Method

C. elegans Nematodes on a Microfluidic Chip

Traditional *C. elegans* handling protocols rely almost entirely on manual manipulations and direct observation by the operator. There, *C. elegans* nematode culture is usually performed on the surface of nematode growth medium agar plates covered with *Escherichia coli* bacteria for feeding. Such manual protocols tend to be especially tedious when analyzing large numbers of animals. They require specialized and advanced manual skills and lack both reproducibility and high-throughput potential. To address this problem, the past decade has seen significant advances in culture conditions and assay development to facilitate the use of *C. elegans* in high-throughput and high-content screening [204]. However, automated image acquisition and data analysis proved to be non-trivial in *C. elegans* systems. While microplate readers can be adapted for capturing images of entire wells, automated analysis of such images remains very challenging. Assays requiring quantification

[5] This work has been carried out in collaboration with D. Schmitter, Biomedical Imaging Group, EPFL, Lausanne, Switzerland; M. Cornaglia and Prof. M. Gijs, Microsystems Laboratory 2, EPFL, Lausanne, Switzerland; and L. Mouchiroud, Nestl Chair in Energy Metabolism, EPFL, Lausanne, Switzerland.

of more complex phenotypes, such as those involving morphological changes in nematodes and differences in behavior or locomotion types appear to be very difficult to process [205]. Lab-on-a-chip technologies and microfluidics stand out as the most convenient approaches in term of functional exploration of *C. elegans* [206]. Microfluidics can in fact combine the advantages of high-throughput liquid-based well-plate screens with a fine-tuned control over genetic or pharmacological interventions, as well as detailed imaging and phenotypic analysis. It can also allow data acquisition at single-nematode resolution rather than at the population level. Furthermore, it opens the possibility to replace traditional error-prone manual culture, handling, and treatment of *C. elegans* by standardized operations integrated in a fully automated fashion.

In the following, we focus on brightfield time-lapse sequences of swimming nematodes placed in a prototype of microfluidics platform for multiplexed nematode culture and imaging designed at the Microsystems Laboratory 2 at EPFL. They thus have the characteristics expected from image data of high-throughput microfluidic experiments.

Automated Nematode Segmentation

Existing methods for analyzing image sequences of *C. elegans* can be broadly categorized into two groups depending on whether the focus is put on providing mostly tracking or segmentation information. Approaches of the first category usually allow studying multiple nematodes in images of medium or low resolution. They do not provide precise segmentation in the sense that they mostly keep track of nematode centroids and are, for this reason, often referred to as *centroid-based* methods. Such approaches do not allow the extraction of precise morphological parameters related to each nematode body, but do give precise tracking information for each individual. Examples of centroid-based image analysis solutions can be found in [207, 208]. The second category comprises methods where most efforts are allocated for segmentation in terms of accurate outlining of the nematode body. They require high-resolution images and can usually only handle a single nematode per image. In most cases, they rely on a custom optical setup where the camera moves so as to maintain the nematode in the center of the field of view, thereby entirely circumventing the tracking problem. For this reason, issues related to background noise are avoided as well. Segmentation most often consists in standard thresholding and mathematical morphology techniques, followed by the identification of the nematode centerline (called the skeleton). A spline curve is finally fitted onto the skeleton in order to facilitate feature extraction. Most algorithms extract the centerline only, although recent approaches have been proposed to obtain full body outlines. Example of such so-called *skeleton-based* methods are reviewed in [200, 209, 210]. A few skeleton-based methods handle the segmentation of multiple nematodes. When available, they are restricted to slow swarming motion [211]. It is generally worth noting that most existing approaches have been tailored for swarming nematodes [212]. Swimming *C. elegans* are slowly getting attention but remain understudied [205]. We can get a sense of this problem by looking at the complexity of sequences of swimming over those of swarming nematodes. When swimming, *C. elegans* can bend into more complicated shapes involving occluded body parts, as it is freely moving in a three-dimensional volume.

Overall, the development of image analysis methods for high-content screening in microfluidic chips presents the five following requirements.

1. Efficiently tracking multiple nematodes: the use of a custom camera setup following individual nematodes in each well is excluded.

2. Providing precise nematode outline at high resolution: segmentation of the whole nematode body is required to accurately extract, *e.g.*, motility phenotypes.

3. Handling swimming nematodes.

4. Discriminating nematodes against possibly complex and dynamic background.

5. Providing maximal amount of automation.

To the best of our knowledge, there exists no proposed solution for this problem. An image analysis approach for the processing of high content data of swimming *C. elegans* therefore remains to be developed. Ideally, image analysis should be flexible enough to deal with nematodes at different life stages, refined enough to handle multiple individuals, occlusion events, and complex background at high resolution, and to provide a data-driven approach to extract discriminative phenotypical features. In the following, we do not claim to fully address all these problems, but rather to show that our landmark snake holds a lot of potential for this application. The proof-of-concept examples we provide hereafter show that our method would be well-suited in a global image analysis pipeline for the segmentation, tracking and analysis of swimming *C. elegans*.

To perform *C. elegans* segmentation, we use our closed landmark snake model given by (5.7) and optimize it on a combination of feature energies. In each frame, a local feature-based energy (5.21) relying on the output of a fine-tuned tip detector is used to attract selected control points at the extremities of the nematodes. In combination with this, an edge-sensitive steerable filter is designed following (4.5), and the purely directional energy (5.9) is used to distribute the remaining control points along the nematode body while their tangents adapt to the local orientation of the contour. The total number of control points composing the snake curve is adapted through a multiresolution approach. First, we start by optimizing an $M = 4$ points snake. Its first and third control points are attracted to the head and tail of the nematode, while the others approach the nematode contour. In this way, all segmentation curves share a common parameterization imposed by the two nematodes extremities (head and tail), which correspond to the first ($k = 0$) and mid-curve ($k = M/2$) control point, respectively. This allows aligning and comparing the segmentation curves obtained at different frames. We then increase the number of control points by introducing pairs in the portions of the snake curve that are along each sides of the nematode, such that M ranges from 4 to 12. This amount to having from 1 to 5 control points along each side of the nematode. Before adding a new pair of control points, we record the segmentation outline and corresponding energy at convergence. At the end of the procedure, the snake curve corresponding to the number of control points that yields the lowest energy is kept. This approach increases the robustness of the algorithm and lowers the risk of self-intersections on complicated nematode shapes as the snake gets optimized step by step with increasingly large number of parameters. In this way, no other energy term is actually needed to obtain proper segmentation results. Tracking nematodes in successive frames is not needed *per se* as, in the sequences we consider, the time resolution is good enough for the nematode bodies to overlap from one frame to the next. We thus simply reuse the segmentation curve from the previous frame as initialization for the new one. An example of landmark snake at convergence overlaid on a nematode is shown in Figure 6.13.

Once the entire image sequence is processed, segmentation errors are subsequently corrected by the user through an interactive interface allowing manual edits on the snake curve. In Figure 6.14, we show segmentation results for a sample frame of each of the three considered sequences.

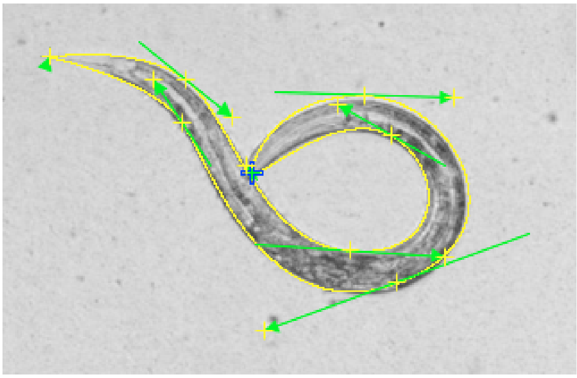


Figure 6.13: A sample *C. elegans* nematode overlaid with a landmark snake at convergence, here composed of $M = 8$ control points. Notice that the first ($\mathbf{r}[0]$) and middle (here, fifth, corresponding to $\mathbf{r}[4]$) control points act as landmarks for the extremities of the nematode.

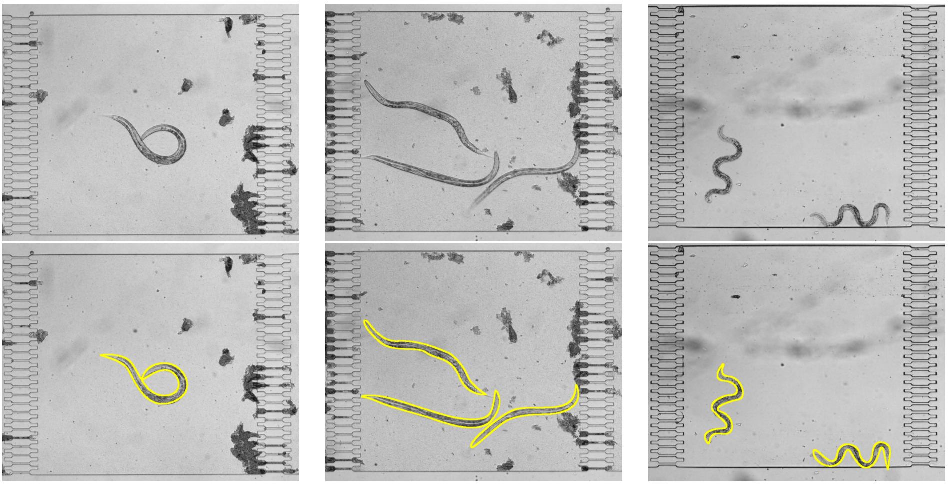
Computation of Nematode Features. The spline nature of segmentation outlines obtained with landmark snakes offers many advantages for the extraction of features. The spline formalism provides a closed-form expression of each of the contours. The coordinates of the control points and the expression of the basis functions is sufficient to retrieve the continuous spline contour $\mathbf{r}$ following (5.7) in a resolution-independent manner and without pixel or discretization artifacts. The continuous length

$$L = \int_0^1 \left(r_1'(t)^2 + r_2'(t)^2\right)^{\frac{1}{2}} \mathrm{d}t \tag{6.26}$$

and instantaneous curvature

$$\kappa = \int_0^1 \left(\frac{r_1''(t)r_2'(t) - r_2''(t)r_1'(t)}{(r_1'(t)^2 + r_2'(t)^2)^{\frac{3}{2}}}\right)^2 \mathrm{d}t \tag{6.27}$$

are examples of physically-meaningful features that can be directly computed from $\mathbf{r}$.



(a) First sequence

(b) Second sequence

(c) Third sequence

Figure 6.14: Raw images (top row) and segmentation results (bottom row) of sample frames of the three considered image sequences featuring (a) a single nematode swimming and coiling, (b) three nematodes swimming, and (c) two nematodes undulating in a swarming-like motion.

Eigenworms Analysis

Identifying the main modes of variation in a collection of similar shapes is at the heart of shape-space analysis. The idea of extracting the main *modes* of motion from a collection of nematode outlines (actually, skeletons) was initially proposed in [203]. There, the main eigenshapes, called *eigenworms*, are obtained using a Fourier descriptors-based analysis. This approach shines new light on dynamics patterns and motility phenotypes as it offers a rich yet simple description of motion [203]. In particular, it allows easily identifying and clustering different motion types. The eigenworms analysis method yielded very promising results, although it is, in its original formulation, limited to the case of swarming nematodes.

The spline nature of our outlines opens the way to a novel shape-space formulation, which also permits to decompose *C. elegans* motion into a collection of modes. The starting point of our method is a dictionary of nematode shapes, that is, a collection of outlines. From this dictionary, the average shape or the shape capturing most of the variation can be extracted. The latter is what we refer to as an eigenworm. Each eigenworm is associated with the amount of variance it is able to capture in the dictionary. In real datasets, one usually observes that the first one or two eigenworms are sufficient to capture more than 90% of the variance. The eigenworms decomposition provides useful information on the nature of a dataset. It allows assessing how similar shapes composing a dictionary are, and estimating how different dictionaries resemble each other. As we shall see, dictionaries can also be built from shapes of the same nematode at different time frames, giving valuable information on the evolution of the dynamics of a single individual.

Formulation of the Shape-Space. We hereafter describe how our novel, continuous spline shape-space is constructed. In the original formulation of shape-space analysis, which dates back to the late eighties [213, 214, 215], shapes specified by discrete sets of points are considered. There, a collection of N shapes is described by N image vectors (*i.e.*, collection of point coordinates that represent the shape contour) denoted by $\mathbf{s}_i \in \mathbb{R}^{2M}$, $i = 1, ..., N$, where M is the number of (many) discrete two-dimensional landmarks defining the shape contour. Each shape vector $\mathbf{s}_i$ is built as $\mathbf{s}_i = (x_{i,0}, y_{i,0}, \dots, x_{i,M}, y_{i,M})$, where $(x_{i,j}, y_{i,j})$ are the coordinates of the jth landmark of the ith shape. First, a geometric normalization step is performed on each $\mathbf{s}_i \in \mathbb{R}^{2M}$, $i = 1, ..., N$ to obtain an "aligned" shape $\widehat{\mathbf{s}}_i = \widehat{\mathbf{T}}_i \mathbf{s}_i + \widehat{\mathbf{b}}_i$,

with

$$\left(\widehat{\mathbf{T}}_i, \widehat{\mathbf{b}_i}\right) = \underset{\mathbf{T},\mathbf{b}}{\arg\min} \|\mathbf{s}_{\mathrm{ref}} - \mathbf{T}\mathbf{s}_i - \mathbf{b}\|^2, \tag{6.28}$$

where $\mathbf{s}_{\mathrm{ref}} \in \mathbb{R}^{2M}$ is a reference shape, $\mathbf{T} \in \mathbb{R}^{2M\times 2M}$ is a transformation matrix, and $\mathbf{b} \in \mathbb{R}^{2M}$ is a translation vector of the form $\mathbf{b} = (t_x, t_y, \dots, t_x, t_y)$ with $t_x, t_y \in \mathbb{R}$. Principal components analysis (PCA) is then performed on the collection of aligned shapes $\{\widehat{\mathbf{s}}_i\}$. This amounts to extracting the eigenvectors of $\mathbf{A}\mathbf{A}^*$, where $\mathbf{A} = [\,\widehat{\mathbf{s}}_1 \dots \widehat{\mathbf{s}}_N\,]$ and $\mathbf{A}^*$ is the unique adjoint of $\mathbf{A}$ such that $\mathbf{s}_i^T(\mathbf{A}\mathbf{s}_j) = (\mathbf{A}^*\mathbf{s}_i)^T\mathbf{s}_j$ [216]. Since $\mathbf{A}$ is a matrix, its adjoint is the transpose of the complex conjugate of $\mathbf{A}$. The matrix $\mathbf{A}$, composed of all aligned shapes, is referred to as the data matrix.

In our approach, shapes are characterized as continuous Hermite spline curves instead of collections of discrete point coordinates. Our dictionary is composed of N shapes, which are Hermite splines $\mathbf{r}_i \in L_2^2([0,1])$, $i = 1, ..., N$. We recall that $L_2^2([0,1])$ is the Hilbert space associated with the inner product

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{L_2^2} = \int_0^1 \mathbf{r}_i(t)^T \mathbf{r}_j(t)\mathrm{d}t = \int_0^1 \left(r_{1,i}(t)r_{1,j}(t) + r_{2,i}(t)r_{2,j}(t)\right)\mathrm{d}t. \tag{6.29}$$

We restrict ourselves to the set of similarity transformations, *i.e.*, $\mathbf{T} = a\mathbf{Q}_\theta$ where $a \in \mathbb{R}$ is a scaling parameter and $\mathbf{Q}_\theta : L_2^2([0,1]) \to L_2^2([0,1])$ the operator applying a rotation of angle θ. This requires two modifications in the standard formulation enunciated above. Each $\mathbf{r}_i \in L_2^2([0,1])$, $i = 1, ..., N$, must be "aligned" to obtain $\widehat{\mathbf{r}}_i \in L_2^2([0,1])$, $i = 1, ..., N$. However, since our shapes lie in the continuous domain, we cannot do point-to-point normalization like in the discrete case. We thus perform the alignment by projecting our shapes onto the spline-based shape-space for similarity transformation, as described in [217]. The projected curve $\widehat{\mathbf{r}}_i = \widehat{a}_i\mathbf{Q}_{\widehat{\theta}_i}\mathbf{r}_i + \widehat{\mathbf{b}}_i$ is obtained through the minimization of the continuous L_2^2 norm as

$$\left(\widehat{a}_i, \widehat{\theta}_i, \widehat{\mathbf{b}}_i\right) = \underset{a,\theta,\mathbf{b}}{\arg\min} \|\mathbf{r}_{\mathrm{ref}} - a\mathbf{Q}_\theta\mathbf{r}_i - \mathbf{b}\|_{L_2^2}^2, \tag{6.30}$$

where $\mathbf{b} \in \mathbb{R}^2$ is a vector of the form $\mathbf{b} = (t_x, t_y)$ with $t_x, t_y \in \mathbb{R}$, which allows translating the curve. The reference shape $\mathbf{r}_{\mathrm{ref}}$ is chosen to be a good representative (*i.e.*, an "average") of the N shapes composing the dictionary. In that way, we take into account the continuous nature of the curves, and we are also able to obtain closed-form solutions in the continuous domain.

Then, we extract the principal components of $\{\widehat{\mathbf{r}}_i\}$ using AA^*, where $A = (A(i,t)) = (\widehat{\mathbf{r}}_i(t))_{i=1,\dots,N,\, t\in[0,1]}$. In the same way that a matrix $\mathbf{A} \in \mathbb{R}^{2M\times 2M}$ specifies a linear mapping from $\mathbb{R}^{2M}$ to $\mathbb{R}^{2M}$, A defines the linear mapping $A : \mathbb{R}^N \to L_2^2([0,1])$. More precisely, for $\mathbf{x} = (x_i)_{i=1,\dots,N} \in \mathbb{R}^N$, we have that $A : \mathbf{x} \mapsto \mathbf{r} = A\mathbf{x}$, where $\mathbf{r} \in L_2^2([0,1])$ and $\mathbf{r}(t) = \sum_{i=1}^N A(i,t)x_i = \sum_{i=1}^N x_i\widehat{\mathbf{r}}_i(t)$ for all $t \in [0,1]$. In particular, for $\mathbf{e}_i = (0,\dots,0,1,0,\dots,0)$, where the 1 is the ith component of $\mathbf{e}_i$, we have that $A\mathbf{e}_i = \widehat{\mathbf{r}}_i$. It is worth noting that, although A seems to be of "$\infty\times N$" dimensions, it actually maps $\mathbb{R}^N$ to $\mathrm{Span}(\widehat{\mathbf{r}}_i)$, which is a finite-dimensional subspace of $L_2^2([0,1])$. The unique adjoint A^* of A is the operator from $L_2^2([0,1])$ to $\mathbb{R}^N$ that satisfies $\langle A\mathbf{x}, \mathbf{r}\rangle_{L_2^2} = \langle \mathbf{x}, A^*\mathbf{r}\rangle_{\mathbb{R}^N}$ for every $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{r} \in L_2^2([0,1])$. Also, $AA^* : L_2^2([0,1]) \to L_2^2([0,1])$ and $A^*A : \mathbb{R}^N \to \mathbb{R}^N$. Importantly, the linear mapping A^*A is therefore a matrix in $\mathbb{R}^{N\times N}$ which coefficients correspond to the L_2^2 inner products between all possible pairs of $\widehat{\mathbf{r}}_i$ for $i = 1,\dots,N$. To perform PCA, we rely on Proposition 5, which is the extension of the known result for $M \times N$ matrices to the "$\infty \times N$" case. For the sake of simplicity, we drop in the following the hat notation and use $\{\mathbf{r}_i\}$, assuming that we are from now on working with aligned shapes.

Proposition 5 (Computation of continuous-domain PCA)**.** *Let $\mathbf{u}_1,\dots,\mathbf{u}_N$ with $\mathbf{u}_i \in \mathbb{R}^N$ be the orthonormal set of eigenvectors of the symmetric positive semidefinite $N \times N$ Gram matrix*

$$A^*A = \begin{bmatrix} \langle \mathbf{r}_1, \mathbf{r}_1\rangle_{L_2^2} & \dots & \langle \mathbf{r}_1, \mathbf{r}_N\rangle_{L_2^2} \\ \vdots & & \vdots \\ \langle \mathbf{r}_N, \mathbf{r}_1\rangle_{L_2^2} & \dots & \langle \mathbf{r}_N, \mathbf{r}_N\rangle_{L_2^2} \end{bmatrix} \tag{6.31}$$

corresponding to the eigenvalues $\lambda_1 \geq \dots \geq \lambda_N \geq 0$. Let $0 \leq N_1 \leq N$ be the integer such that λ_{N_1} is the smallest strictly positive eigenvalue of A^*A. Then, $AA^* : L_2^2([0,1]) \to L_2^2([0,1])$ admits the eigenvalue decomposition

$$AA^* = \sum_{i=1}^{N_1} \lambda_i \tilde{\mathbf{u}}_i \otimes \tilde{\mathbf{u}}_i \tag{6.32}$$

where $\otimes$ is the tensor product. The eigenfunctions $\tilde{\mathbf{u}}_i \in L_2^2([0,1])$, $i = 1,\dots,N_1$ of AA^* satisfy

$$AA^*\{\tilde{\mathbf{u}}_i\} = \lambda_i\{\tilde{\mathbf{u}}_i\} \tag{6.33}$$

and form an orthonormal set, that is, $\langle\, \tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j \,\rangle_{L_2^2} = \delta[i-j]$*. They are in addition related to the eigenvectors* $\mathbf{u}_i$ *of* A^*A *as*

$$\tilde{\mathbf{u}}_i = \frac{1}{\sqrt{\lambda_i}} A\mathbf{u}_i. \tag{6.34}$$

Proof. We first assume that $\lambda_i > 0$ and show that $\tilde{\mathbf{u}}_i$ is an eigenfunction of AA^* as

$$AA^*\tilde{\mathbf{u}}_i = A\left(A^*A\right)\mathbf{u}_i/\sqrt{\lambda_i} = A\lambda_i\mathbf{u}_i/\sqrt{\lambda_i} = \lambda_i\tilde{\mathbf{u}}_i, \tag{6.35}$$

where we used (6.34) and the fact that $\mathbf{u}_i$ is the eigenvector of A^*A associated to the eigenvalue λ_i. Next, we show the orthogonality of the $A\mathbf{u}_i$ as

$$\langle\, A\mathbf{u}_i, A\mathbf{u}_j \,\rangle_{L_2^2} = \langle\, A^*A\mathbf{u}_i, \mathbf{u}_j \,\rangle_{L_2^2} = \langle\, \lambda_i\mathbf{u}_i, \mathbf{u}_j \,\rangle_{L_2^2} = \lambda_i\delta[i-j]. \tag{6.36}$$

The orthonormality of the $\tilde{\mathbf{u}}_i$ for $i = 1, ..., N_1$ (*i.e.*, for $\lambda_i > 0$) in $L_2^2([0,1])$ immediately follows by substituting (6.34) in the above. The considered collection of shapes might contain duplicates or, more likely, shapes that very closely resemble each other although not being exact duplicates. In such cases, A^*A is ill-conditioned and $N_1 < N$, which makes it worth addressing the possibly remaining $N - N_1$ vanishing eigenvalues. For them, we have

$$\langle\, A\mathbf{u}_i, A\mathbf{u}_i \,\rangle_{L_2^2} = \|A\mathbf{u}_i\|_{L_2^2}^2 = \lambda_i = 0, \tag{6.37}$$

which implies that $A\mathbf{u}_i = 0$ for $i = N_1 + 1, ..., N$.

In the situation where none of the eigenvalues are vanishing (*i.e.*, when $N_1 = N$), we have

$$\begin{aligned}\sum_{n=1}^{N} \lambda_i\tilde{\mathbf{u}}_i \otimes \tilde{\mathbf{u}}_i =& \sum_{n=1}^{N} \lambda_i \frac{A\mathbf{u}_i}{\sqrt{\lambda_i}} \otimes \frac{A\mathbf{u}_i}{\sqrt{\lambda_i}} = \sum_{n=1}^{N} A\mathbf{u}_i \otimes A\mathbf{u}_i \\ =& A\left(\sum_{n=1}^{N} \mathbf{u}_i \otimes \mathbf{u}_i\right) A^* = AA^*,\end{aligned} \tag{6.38}$$

which gives the eigendecomposition of AA^*. The last step is obtained from the orthonormality of the $\mathbf{u}_i$, which implies that $\mathbf{u}_i \otimes \mathbf{u}_i = \mathbf{I}_N$, where $\mathbf{I}_N$ is the identity matrix of size N. In cases where $N_1 < N$, we have from (6.37) that

$$\sum_{n=N_1+1}^{N} A\mathbf{u}_i \otimes A\mathbf{u}_i = 0. \tag{6.39}$$

Therefore, we can write as well

$$\sum_{n=1}^{N_1} \lambda_i \tilde{\mathbf{u}}_i \otimes \tilde{\mathbf{u}}_i = \sum_{n=1}^{N_1} A\mathbf{u}_i \otimes A\mathbf{u}_i = \sum_{n=1}^{N} A\mathbf{u}_i \otimes A\mathbf{u}_i, \tag{6.40}$$

which completes the proof. □

The eigenvectors obtained from the above continuous-domain PCA are our eigenworms. It is important to notice that, in our setting, the eigenworms are composed of contributions of all dictionary atoms.

Efficient Computation of Inner-Products. To extract continuous spline eigenworms with our approach, the only required ingredient is an eigendecomposition of a finite-dimension matrix. However, this matrix is composed of continuous $L_2^2([0,1])$ inner products. An efficient way of computing them is therefore needed. To do so, we adapt the method of [218] to Hermite splines.

Let $c_{k,l}[m,n] = \int_0^1 \phi_{k,\mathrm{per}}(t-m)\phi_{l,\mathrm{per}}(t-n)\mathrm{d}t$ with $k,l \in \{1,2\}$ and where $\phi_{1,\mathrm{per}}$ and $\phi_{2,\mathrm{per}}$ are the M-periodized versions of ϕ_1 and ϕ_2, given by (5.5) and (5.6), respectively. Then, the inner product between any two closed landmark snake curves defined as (5.7) is given by

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{L_2^2} = \int_0^1 \mathbf{r}_i(t)^T \mathbf{r}_j(t)\mathrm{d}t \tag{6.41}$$

$$\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} \big(\mathbf{r}_i[m]^T\mathbf{r}_j[n]c_{1,1}[m,n] + \mathbf{r}'_i[m]^T\mathbf{r}_j[n]c_{2,1}[m,n] + \mathbf{r}_i[m]^T\mathbf{r}'_j[n]c_{1,2}[m,n] + \mathbf{r}'_i[m]^T\mathbf{r}'_j[n]c_{2,2}[m,n]\big). \tag{6.42}$$

Now let us set

$$\mathbf{B}_{kl} = \begin{bmatrix} \mathbf{\Phi}_{kl} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{kl} \end{bmatrix}, \tag{6.43}$$

where $[\mathbf{\Phi}_{kl}]_{m,n} = c_{k,l}[m,n]$, and

$$\tilde{\mathbf{r}}_i = \begin{bmatrix} r_{1,i}[0] \\ \vdots \\ r_{1,i}[M-1] \\ r_{2,i}[0] \\ \vdots \\ r_{2,i}[M-1] \end{bmatrix}, \quad \tilde{\mathbf{s}}_i = \begin{bmatrix} r'_{1,i}[0] \\ \vdots \\ r'_{1,i}[M-1] \\ r'_{2,i}[0] \\ \vdots \\ r'_{2,i}[M-1] \end{bmatrix}. \tag{6.44}$$

Note that $\mathbf{B}_{kl}$ is easily precomputable in closed form and can be stored in computer memory. It is in fact extremely sparse due to the small support of ϕ_1 and ϕ_2. The vectors $\tilde{\mathbf{r}}_i$ and $\tilde{\mathbf{s}}_i$ only contain the coordinates of the control points and are therefore directly given by the position of the active contours in the image.

The inner product can finally be computed as

$$\begin{aligned} \langle \mathbf{r}_i, \mathbf{r}_j \rangle_{L_2^2} =& \tilde{\mathbf{r}}_i^T \mathbf{B}_{11} \tilde{\mathbf{r}}_j + \tilde{\mathbf{r}}_i^T \mathbf{B}_{12} \tilde{\mathbf{s}}_j + \tilde{\mathbf{s}}_i^T \mathbf{B}_{21} \tilde{\mathbf{r}}_j + \tilde{\mathbf{s}}_i^T \mathbf{B}_{22} \tilde{\mathbf{s}}_j \\ =& \begin{bmatrix} \tilde{\mathbf{r}}_i^T & \tilde{\mathbf{s}}_i^T \end{bmatrix} \mathbf{B} \begin{bmatrix} \tilde{\mathbf{r}}_j \\ \tilde{\mathbf{s}}_j \end{bmatrix}, \end{aligned} \tag{6.45}$$

with

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}. \tag{6.46}$$

We note that $\mathbf{B}$ is of dimension $4M \times 4M$ and the two vectors around it are $1 \times 4M$ and $4M \times 1$, respectively. The inner product of any two closed Hermite spline curves thus boils down to simple and computationally inexpensive linear algebra.

6.3.2 Results

We illustrate eigenworm analysis results for the three microfluidic time-lapse image datasets introduced in Figure 6.14. The sequences are all composed of 188 frames of 800×640 pixels. The first sequence features a single nematode swimming and coiling, the second one contains three nematodes swimming, and the third one shows two nematodes undulating in a swarming-like motion. In all cases, the analysis pipeline is composed of the three same steps: nematode segmentation using landmark snakes, construction of a shape-space dictionary, and extraction of the eigenworms. What differs in each example is the nature of the dictionary and, as a consequence, the kind of information that the eigenworms provide.

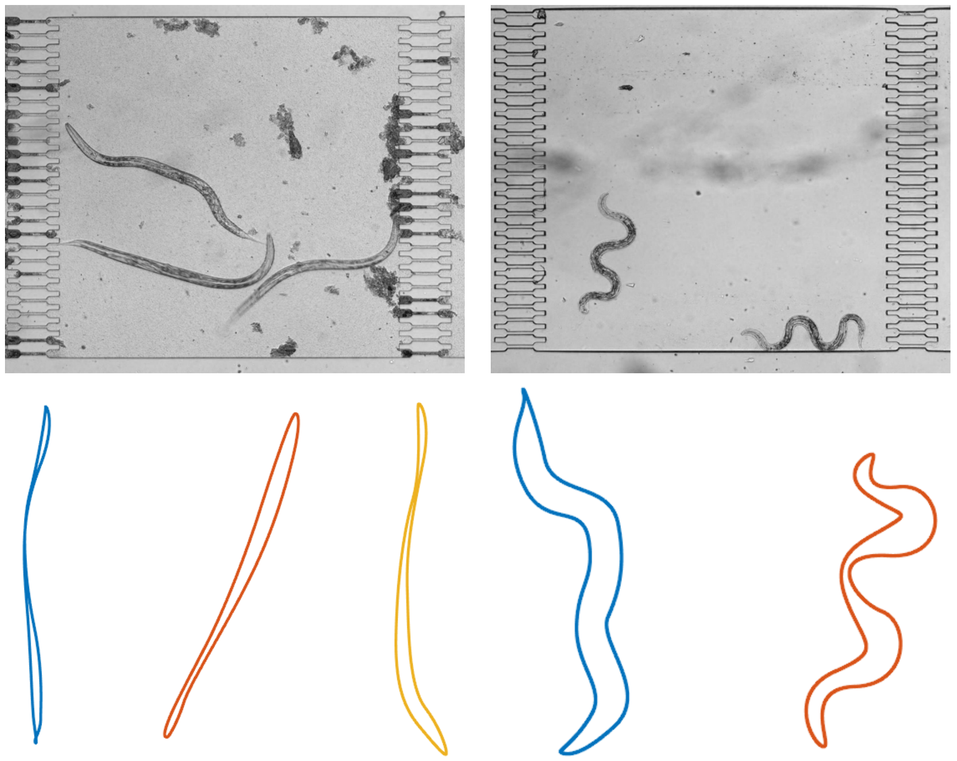
Analysis of Dominating Dynamics

In a first setting, we study different populations of *C. elegans* in different chambers of the microfluidic chip. This is meant to illustrate situations in which different experimental conditions are tested, for instance by delivering different type of drugs to each chamber. We consider the second and third sequences. They feature several nematodes (three and two, respectively) in each chamber, and one characteristic motion type per chamber (regular swimming and undulating, respectively). We build one dictionary for each nematode by gathering all its segmentation outlines over the sequence, and extract the first eigenworm of the resulting shape-space. We therefore have three dictionaries of 188 shapes for the second sequence and two dictionaries of 188 shapes for the third one. We show the first eigenworms of each dictionary in Figure 6.15. We also indicate the amount of variance they capture in their own dictionary, which is in all cases very high. This indicates that each of the five dictionaries most likely contains a single motion type.

The first eigenworms of individuals in the same chamber are similar. This indicates that, within each chamber, nematodes share the same kind of dynamics. Conversely, the eigenworms are strongly dissimilar between chambers, as expected from the different dynamic in these two environments. Interestingly, the eigenworms of the second sequence appear as very straight, which might seem to be contradictory for swimming *C. elegans.* This result however still makes sense since, in the swimming process, most of the nematode body remains steady while the extremities are swiftly wiggling. A straight shape is therefore best suited to get a first order approximation of the shape of the nematode while swimming. The small variations due to bending at the tips correspond to high frequency features and are captured by further eigenworms.

Analysis of Motion-Type Evolution

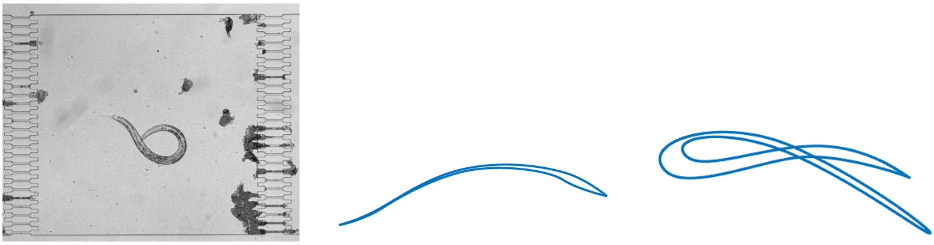
In a second setting, we focus on the first sequence, which displays a single nematode with changing motion dynamics. We first construct a shape dictionary by simply considering the segmentation outlines in all frames of the sequence. In Figure 6.16,



(a) $\lambda = 88\%$, $\lambda = 86\%$, and $\lambda = 93\%$

(b) $\lambda = 93\%$ and $\lambda = 95\%$

Figure 6.15: Qualitative estimation of types of motion relying on eigenworms analysis. Sample raw image frames (top row) and first eigenworm (bottom row) for each nematode in the (a) second and (b) third image sequence. The eigenworms are obtained by performing continuous principal component analysis on a dictionary composed of the collection of outlines of each individual nematode across the image sequence. The amount of variance that each eigenworm captures is indicated as λ.



(a) Raw image

(b) $\lambda = 86\%$

(c) $\lambda = 10\%$

Figure 6.16: Identification of multiple motion modes relying on eigenworms analysis. (a) Sample raw image frame, (b) first and (c) second eigenworm for the nematode in the first image sequence. The eigenworms are obtained by performing continuous principal component analysis on a dictionary composed of the collection of outlines at each frame of the image sequence. The amount of variance that each eigenworm captures is indicated as λ.

we display the two first eigenworms along with their associated variance. The shapes composing the dictionary can be mainly represented by these two eigenworms, indicating the presence of two dominating dynamics. The first one is representative of standard swimming behavior as discussed in the previous experiment, and the second one hints at coiling events.

To investigate more precisely the time evolution of the motion through the sequence, we add temporal resolution to our eigenworm analysis and build dictionaries using segmentation outlines from a sliding window around each frame. At frame t, the dictionary is thus composed of outlines from frames $t - \delta$ to $t + \delta$, where δ corresponds to half the size of the sliding window. In this way, a collection of eigenworms can be extracted from every frame of the sequence to describe the main modes of motion in a close temporal neighborhood. The first eigenworms for a collection of selected time points in the sequence are displayed in Figure 6.17. The transition from coiling in the first part of the sequence to standard swimming dynamics gets clearly noticeable.

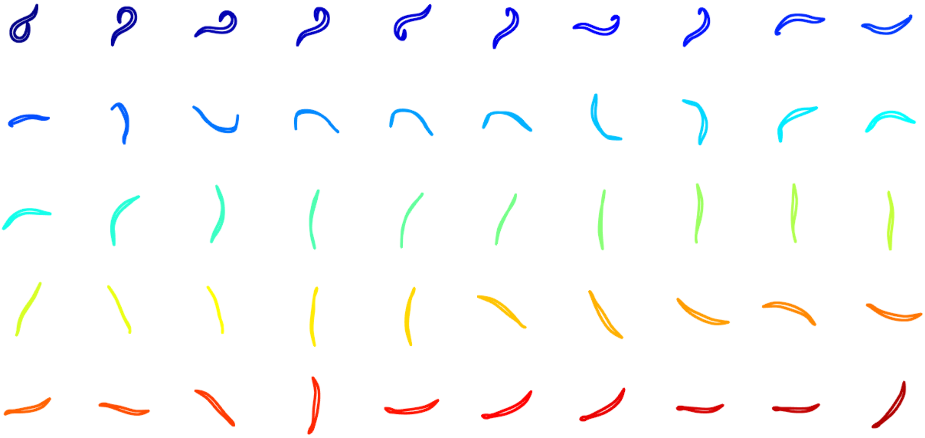


Figure 6.17: Eigenworms analysis with temporal resolution. For a collection of time points, we show the first eigenworm obtained from the dictionary composed of outlines in frames surrounding the considered time point. It offers a compact view of the dynamic of the nematode through the sequence. The color code indicates the frame value (blue, top left: beginning of the sequence, red, bottom right: end of the sequence).

6.3.3 Discussion

The study of *C. elegans* dynamics is a good illustration of the potential held by landmark active contours for image analysis problems involving the outlining of objects with clear landmarks. Our approach is flexible enough to provide precise outlines of swimming nematodes in time-lapse microscopy images in a mostly automated fashion. Segmentation errors do arise but can easily be corrected manually. In order to further enhance automation, segmentation robustness could be reinforced by incorporating learning-based energy terms. Another approach could be to monitor segmentation quality and automatically flag frames where problems are detected, for instance when the value of the energy at convergence is too high, in order to efficiently guide user input and feedback. The eigenworms analysis, although remaining only qualitative, provides an interesting decomposition of the nematode motion and allows identifying different types of dynamics. Combining landmark snakes with such an analysis is a first step towards the design of automated pipelines for high-throughput microfluidic experiments involving *C. elegans*. Obvious required improvements include the introduction of more refined tracking strategies, alleviating the need for overlap in successive frames. In addition, more quantitative measurements of motion could be extracted from the eigenworms. Clustering algorithms could for instance be used to get quantitative estimations on how distant different eingenworms are. In our last example involving dictionaries at different time steps, eigenworm analysis also provides an automated way of identifying transitions in motion types. Such events could be automatically detected by monitoring the distance between eigenshapes extracted from successive time points.

This application is a typical example where landmark snakes appear to be a convenient solution, mainly because of the salient features (here, the nematode extremities) that characterize the objects of interest. Outlining nematodes could in fact be achieved by other spline snake models, but would involve a lot of control points and imply an increased risk of entanglement at the extremities. Similarly, continuous shape space analysis analogous to the one we propose could be derived for other spline models. However, shape alignment requires that all objects have the same parameterization origin. If tips were to be represented by a collection of aggregated control points, imposing a rule stating which control point should be considered as the origin of the parameterized curve would not be an easy task. In our case, however, since landmark snakes offer a one-to-one correspondence between control points and landmarks on the curve, a standardized parameterization is straightforwardly obtained.

Chapter 7

Concluding Remarks

We close this thesis with a few concluding remarks. First, we propose to look down the road and recap our technical contributions. Then, we briefly discuss future research possibilities.

7.1 Taking a Look Back

Our work focused on the problem of segmentation, or outlining, in the context of bioimage analysis. We could identify two different angles to approach this problem, namely identifying contours or detecting features in the image. Bringing these two ideas together, we designed landmark active contours, which consist in curves that deform according to both contour and feature information. Various technical contributions came up along the way. We summarize them in the following, grouped by field.

In the Fields of Spline and Approximation Theory

- We explicitly made the link between the well-known cubic Hermite spline interpolation scheme and more classical ones such as Bézier curves and cubic B-splines.

- We studied the approximation error of the cubic Hermite interpolation scheme and compared it with other methods using multigenerators in a generalized sampling context.

- We demonstrated several optimality results of cubic Hermite splines, both in terms of smoothness properties and of MMSE estimation.

- We briefly illustrated further generalization possibilities in the spirit of L-splines. In particular, we introduced the exponential Hermite splines, a novel Hermite spline family generating piecewise-exponential curves with C^1-continuity at the knots.

In the Fields of Wavelet Theory and Steerable Filters Design

- We introduced a procedure to design angular profile for the detection of symmetric patterns. We showed proof-on-concept application examples in bioimages featuring K-fold symmetries.

- We proposed a scheme to design radial wavelet profiles that optimize different measures of localization. We illustrated how they improve performances in experiments involving steerable filters for local orientation estimation and image reconstruction from a sparse subset of image features.

In the Field of Bioimage Analysis

- We introduced the landmark snake model based on cubic Hermite splines, which differs from classical spline snake models by offering direct control over the tangents of the curve at the control point locations. It allows interpreting control points as landmarks, and adapting their tangents to the nature of local image features.

- We proposed novel local feature-based energies designed for the landmark snake and showed proof-of-concept examples of their usefulness in practice.

- We illustrated the use of landmark snakes by segmenting objects in a collection of bioimages from various microscopy modalities. We also compared results with those obtained with classical spline snakes.

- We described three collaborative projects where our steerable feature detectors, cubic Hermite splines and landmark snakes were used individually or together to tackle a particular bioimage analysis problem.

More aspects are left to be investigated in all the fields we explored. About splines, a complete theoretical framework characterizing Hermite L-splines in all generality remains to be developed in a mathematically rigorous way. Similarly, regarding the design of steerable filters, preliminary results obtained with filters learned from templates call for further investigation and extensive tests in practical applications. Our main novelty, the landmark snake, was observed to bring a true added value in practical applications. However, a virtually infinite number of variations on this theme could still be explored. Many other approaches can be considered to extract landmarks locations and descriptors, and specific energies can be constructed to exploit the information that they provide. Also, we only focused on first-order Hermite interpolation, but the Hermite framework is actually much richer. Exploring higher orders of Hermite interpolation and higher order Hermite splines built from a variety of differential operators opens the way to an endless number of snake model variants. Our landmark snake model is based on cubic polynomial Hermite splines, but another model using exponential Hermite generators could directly be constructed from (5.3) and (5.7) by simply changing the generators ϕ_1 and ϕ_2. Similarly, more refined landmark snakes could be designed by considering Hermite interpolation of higher order. One additional order would result in a snake variant that relies on three basis functions and has direct control over the first and second derivative at each control point, making it possible to constrain the local curvature at landmark locations.

Taking a look back at the road traveled through this thesis, several technical contributions can be pulled out, both theoretical (in the fields of steerable filters and spline) and practical (in the domain of segmentation and, more precisely, of bioimage analysis). This excursion is however far from being self-sufficient. All studied elements can serve as starting points for further research directions. More than proposing answers to the questions presented at the beginning of this work, our results open a multitude of additional paths remaining to be carved.

7.2 Future Prospects

In the introduction, we described the segmentation problem and highlighted how the challenges it poses differ in the fields of computer vision and bioimage analysis. An interesting aspect of the landmark snake is that it bring these domains closer together in its ability to take into account both the specificity of the objects being searched for (as in computer vision settings) and the flexibility of their nature (as in bioimage informatics applications). Landmark energies allow introducing potentially very specific information on the appearance of the objects of interest, which stands out of the classical snake energy paradigm. Individual snake energy terms are usually more generic, based on global intensities or textures considerations, or on the overall shape the curve is expected to adopt. The specificity then emerges from the unique combination of such general terms. The use of landmarks and their association with control points introduces targeted specificity in the segmentation process, without affecting the overall flexibility and generality of snake algorithms. Designing energies for classical keypoint detectors such as SIFT and exploring the use of landmark snakes in computer vision-type segmentation problems is a natural future direction for our work.

From a broader perspective, the landmark snake also provides a primer to reconcile learning and more classical image processing approaches to the segmentation problem. The snake model is built in a classical image analysis spirit, relying on well-established results from spline and approximation theory. The nature of the detected landmark and the incorporation of landmark-related information remains however open to learning-based methods. Although not explored here, it is very well imaginable to rely on the output of appropriately trained deep neural networks in order to identify landmark locations and their local properties, which would allow to directly reuse the local feature-based energy terms we introduced. The next step consists in coming up with more refined energies, possibly being themselves shaped by a learning process, thus making full use of the deep learning potential. This combination would yield mathematically sound and well-characterized segmentation outlines obtained from a learned combination of available image information. In the age of glory of deep neural networks, such a construction could offer a pleasant compromise using the best of both worlds.

Bibliography

[1] C. Vonesch, F. Aguet, J.-L. Vonesch, and M. Unser, "The colored revolution of bioimaging," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 20–31, May 2006.

[2] A. Miyawaki, "Visualization of the spatial and temporal dynamics of intracellular signaling," *Developmental Cell*, vol. 4, no. 3, pp. 295–305, March 2003.

[3] D. Muzzey and A. van Oudenaarden, "Quantitative time-lapse fluorescence microscopy in single cells," *Annual Review of Cell and Developmental*, vol. 25, pp. 301–327, November 2009.

[4] 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, no. 3, pp. 54–62, May 2006.

[5] M. Oheim, "Advances and challenges in high-throughput microscopy for live-cell subcellular imaging," *Expert Opinion on Drug Discovery*, vol. 6, no. 12, pp. 1299–1315, December 2011.

[6] A. Cardona and P. Tomancak, "Current challenges in open-source bioimage informatics," *Nature Methods*, vol. 9, no. 7, pp. 661–665, June 2012.

[7] K.W. Eliceiri, M.R. Berthold, I.G. Goldberg, L. Ibáñez, B.S. Manjunath, M.E. Martone, R.F. Murphy, H. Peng, A.L. Plant, B. Roysam, N. Stuurman, J.R. Swedlow, P. Tomancak, and A.E. Carpenter, "Biological imaging software tools," *Nature Methods*, vol. 9, no. 7, pp. 697–710, June 2012.

[8] G. Myers, "Why bioimage informatics matters," *Nature Methods*, vol. 9, no. 7, pp. 659, June 2012.

[9] M.D. Abràmoff, P.J. Magalhães, and S.J. Ram, "Image processing with ImageJ," *Biophotonics International*, vol. 11, no. 7, pp. 36–42, July 2004.

[10] 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.J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, June 2012.

[11] 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 research," *Nature Methods*, vol. 9, no. 7, pp. 690–696, July 2012.

[12] A.E. Carpenter, T.R. Jones, M.R. Lamprecht, C. Clarke, I.H. Kang, O. Friman, D.A. Guertin, J.H. Chang, R.A. Lindquist, J. Moffat, P. Golland, and D.M. Sabatini, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biology*, vol. 7, no. 10, pp. R100, October 2006.

[13] C. Sommer, C. Straehle, U. Koethe, and F.A. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *Proceedings of the Eighth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'11)*, Chicago, IL, USA, March 30-April 2, 2011, pp. 1917–1920.

[14] D. Marr, *Vision A Computational Investigation into the Human Representation and Processing of Visual Information*, MIT Press, Cambridge, MA, USA, 2010.

[15] N.R. Pal and S.K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, September 1993.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the Eighteenth International Conference on Medical Image Computing and Computer-Assisted*

Intervention (MICCAI'15), Munich, Germany, October 5-9, 2015, pp. 234–241.

[17] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, June 2008.

[18] D.G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer vision (ICCV'09)*, Kerkyra, Greece, September 20-27, 1999, vol. 2, pp. 1150–1157.

[19] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings of the Ninth European Conference on Computer Vision (ECCV'06)*, Graz, Austria, May 7-13, 2006, vol. 1, pp. 404–417.

[21] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, October 2005.

[22] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Proceedings of the Ninth European Conference on Computer Vision (ECCV'06)*, vol. 1, pp. 430–443, May 7-13, 2006.

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," *Proceedings of the Eleventh European Conference on Computer Vision (ECCV'10)*, vol. 4, pp. 778–792, September 5-11, 2010.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV'11)*, Barcelona, Spain, November 6-13, 2011, pp. 2564–2571.

[25] S. Leutenegger, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceedings of the Thirteenth IEEE International*

Conference on Computer Vision (ICCV'11), Barcelona, Spain, November 6-13, 2011, pp. 2548–2555.

[26] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proceedings of the Twenty-Fifth IEEE conference on Computer Vision and Pattern Recognition (CVPR'12)*, Providence, RI, USA, June 16-21, 2012, pp. 510–517.

[27] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, Manchester, United Kingdom, August 31-September 2, 1988, pp. 147–151.

[28] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, no. 1–2, pp. 87–119, October 1995.

[29] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, May 1997.

[30] P.R. Beaudet, "Rotationally invariant image operators," in *Proceedings of the Fourth International Joint Conference on Pattern Recognition (IJCPR'78)*, Tokyo, Japan, November 7-10, 1978, pp. 579–583.

[31] M. Jacob and M. Unser, "Design of steerable filters for feature detection using Canny-like criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1007–1019, August 2004.

[32] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41–48, January 2015.

[33] A.P. Britto and G. Ravindran, "Review of deformable curves—A retro analysis," *Journal of Information Technology*, vol. 6, no. 1, pp. 26–36, January 2007.

[34] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988.

[35] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, February 1997.

[36] T. Heimann and H.-P. Meinzer, “Statistical shape models for 3D medical image segmentation: A review,” *Medical Image Analysis*, vol. 13, no. 4, pp. 543–563, August 2009.

[37] R. Malladi, J.A. Sethian, and B.C. Vemuri, “Shape modeling with front propagation: A level set approach,” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, February 1995.

[38] H. Zhang, Z. Bian, Y. Guo, B. Fei, and M. Ye, “An efficient multiscale approach to level set evolution,” in *Proceedings of the Twenty-Fifth Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’03)*, Cancun, Mexico, September 17-21, 2003, pp. 694–697.

[39] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, “Image segmentation using active contours: Calculus of variations or shape gradients?,” *SIAM Journal on Applied Mathematics*, vol. 63, no. 6, pp. 2128–2154, September 2003.

[40] B. Appleton and H. Talbot, “Globally optimal geodesic active contours,” *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 67–86, July 2005.

[41] X. Bresson, P. Vandergheynst, and J.-P. Thiran, “Multiscale active contours,” *International Journal of Computer Vision*, vol. 70, no. 3, pp. 197–211, December 2006.

[42] 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, no. 4, pp. 668–676, April 2010.

[43] S. Osher and J.A. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, November 1988.

[44] P. Brigger, J. Hoeg, and M. Unser, "B-Spline snakes: A flexible tool for parametric contour detection," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1484–1496, September 2000.

[45] M.A.T. Figueiredo, J.M.N. Leitão, and A.K. Jain, "Unsupervised contour representation and estimation using B-splines and a minimum description length criterion," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1075–1087, June 2000.

[46] L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, November 1992.

[47] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, June 2000.

[48] A.K. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Processing*, vol. 71, no. 2, pp. 109–129, December 1998.

[49] S. Menet, P. Saint-Marc, and G. Medioni, "Active contour models: Overview, implementation and applications," in *Proceedings of the 1990 IEEE International Conference on Systems, Man and Cybernetics (SMC'90)*, Los Angeles, CA, USA, November 4-7, 1990, pp. 194–199.

[50] N. Ray, B. Chanda, and J. Das, "A fast and flexible multiresolution snake with a definite termination criterion," *Pattern Recognition*, vol. 34, no. 7, pp. 1483–1490, 2001.

[51] V.V. Kindratenko, "On using functions to describe the shape," *Journal of Mathematical Imaging and Vision*, vol. 18, no. 3, pp. 225–245, May 2003.

[52] L.D. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, August 1997.

[53] B. Li and S.T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2096–2106, August 2007.

[54] N. Ray and S.T. Acton, "Motion gradient vector flow: An external force for tracking rolling leukocytes with shape and size constrained active contours," *IEEE Transactions on Medical Imaging*, vol. 23, no. 12, pp. 1466–1478, December 2004.

[55] N. Dénervaud, J. Becker, R. Delgado-Gonzalo, P. Damay, A.S. Rajkumar, M. Unser, D. Shore, F. Naef, and S.J. Maerki, "A chemostat array enables the spatio-temporal analysis of the yeast proteome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 39, pp. 15842–15847, September 24, 2013.

[56] D. Schmitter, P. Wachowicz, D. Sage, A. Chasapi, I. Xenarios, V. Simanis, and M. Unser, "A 2D/3D image analysis system to track fluorescently labeled structures in rod-shaped cells: Application to measure spindle pole asymmetry during mitosis," *Cell Division*, vol. 8, no. 6, pp. 1–13, April 27, 2013.

[57] M. Unser, "Sampling—50 years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.

[58] R. Delgado-Gonzalo and M. Unser, "Spline-based framework for interactive segmentation in biomedical imaging," *IRBM—Ingénierie et Recherche Biomédicale / BioMedical Engineering and Research*, vol. 34, no. 3, pp. 235–243, June 2013.

[59] R. Kimmel and A.M. Bruckstein, "Regularized Laplacian zero crossings as optimal edge integrators," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 225–243, July 2003.

[60] W.M. Neuenschwander, P. Fua, L. Iverson, G. Székely, and O. Kübler, "Ziplock snakes," *International Journal of Computer Vision*, vol. 25, no. 3, pp. 191–201, December 1997.

[61] Y.Y. Wong, P.C. Yuen, and C.S. Tong, "Segmented snake for contour detection," *Pattern Recognition*, vol. 31, no. 11, pp. 1669–1679, November 1998.

[62] M.B. Smith, H. Li, T. Shen, X. Huang, E. Yusuf, and D. Vavylonis, "Segmentation and tracking of cytoskeletal filaments using open active contours," *Cytoskeleton*, vol. 67, no. 11, pp. 693–705, November 2010.

[63] J. Melonakos, E. Pichon, S. Angenent, and A. Tannenbaum, "Finsler active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 412–423, March 2008.

[64] R. Delgado-Gonzalo, P. Thévenaz, and M. Unser, "Exponential splines and minimal-support bases for curve representation," *Computer Aided Geometric Design*, vol. 29, no. 2, pp. 109–128, February 2012.

[65] M. Jacob, T. Blu, and M. Unser, "Efficient energies and algorithms for parametric snakes," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1231–1244, September 2004.

[66] L.D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, March 1991.

[67] C. Xu and J.L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, March 1998.

[68] J. Tang and S.T. Acton, "Vessel boundary tracking for intravital microscopy via multiscale gradient vector flow snakes," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 2, pp. 316–324, February 2004.

[69] P. Thévenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, February 2011.

[70] R. Delgado-Gonzalo, P. Thévenaz, C.S. Seelamantula, and M. Unser, "Snakes with an ellipse-reproducing property," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1258–1271, March 2012.

[71] T.F. Chan and L.A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001.

[72] C.M. Hoffmann, *Geometric and Solid Modeling*, Morgan Kaufmann, Burlington, MA, USA, first edition, 1989.

[73] C. Zimmer and J.-C. Olivo-Marin, "Coupled parametric active contours," *IEEE Transactions on Pattern Anaysis and Machine Intelligence*, vol. 27, no. 11, pp. 1838–1842, November 2005.

[74] P.R. Lipow and I.J. Schoenberg, "Cardinal interpolation and spline functions. III. Cardinal Hermite interpolation," *Linear Algebra and its Applications*, vol. 6, pp. 273–304, 1973.

[75] I.J. Schoenberg and A. Sharma, "Cardinal interpolation and spline functions. V. The B-splines for cardinal Hermite interpolation," *Linear Algebra and its Applications*, vol. 7, no. 1, pp. 1–42, January 1973.

[76] V. Uhlmann, J. Fageot, and M. Unser, "Hermite snakes with control of tangents," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2803–2816, June 2016.

[77] V. Uhlmann, J. Fageot, H. Gupta, and M. Unser, "Statistical optimality of Hermite splines," in *Proceedings of the Eleventh International Workshop on Sampling Theory and Applications (SampTA'15)*, Washington DC, USA, May 25-29, 2015, pp. 226–230.

[78] M.H. Schultz and R.S. Varga, "L-splines," *Numerische Mathematik*, vol. 10, no. 4, pp. 345–369, November 1967.

[79] L. Schumaker, *Spline Functions: Basic Theory*, Cambridge University Press, Cambridge, United Kingdom, third edition, 2007.

[80] M. Unser and T. Blu, "Self-similarity: Part I—Splines and operators," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1352–1363, April 2007.

[81] W. Dahmen, B. Han, R.-Q. Jia, and A. Kunoth, "Biorthogonal multiwavelets on the interval: Cubic Hermite splines," *Constructive Approximation*, vol. 16, no. 2, pp. 221–259, February 2000.

[82] M. Unser, A. Aldroubi, and M. Eden, "B-Spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.

[83] M. Unser, A. Aldroubi, and M. Eden, "B-Spline signal processing: Part II—Efficient design and applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, February 1993.

[84] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, November 1999.

[85] M. Unser and P. Tafti, “Sparse stochastic processes,” in *An Introduction to Sparse Stochastic Processes*, pp. 149–189. Cambridge University Press, Cambridge, United Kingdom, first edition, August 2014.

[86] 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.

[87] A. Aldroubi and K. Gröchenig, “Nonuniform sampling and reconstruction in shift-invariant spaces,” *SIAM Review*, vol. 43, no. 4, pp. 585–620, December 2001.

[88] A. Aldroubi, “Oblique projections in atomic spaces,” *Proceedings of the American Mathematical Society*, vol. 124, no. 7, pp. 2051–2060, July 1996.

[89] A. Papoulis, “Generalized sampling expansion,” *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, November 1977.

[90] D. Slepian, “On bandwidth,” *Proceedings of the IEEE*, vol. 64, no. 3, pp. 292–300, March 1976.

[91] M. Unser and J. Zerubia, “A generalized sampling theory without band-limiting constraints,” *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 45, no. 8, pp. 959–969, August 1998.

[92] T. Blu and M. Unser, “Approximation error for quasi-interpolators and (multi-) wavelet expansions,” *Applied and Computational Harmonic Analysis*, vol. 6, no. 2, pp. 219–251, March 1999.

[93] M. Unser and J. Zerubia, “Generalized sampling: Stability and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 2941–2950, December 1997.

[94] T. Blu and M. Unser, “Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors,” *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2783–2795, October 1999.

[95] G. Strang and G. Fix, “A Fourier analysis of the finite element variational method,” in *Constructive Aspect of Functional Analysis*, pp. 796–830. Cremonese, Rome, Italy, 1971.

[96] J. Kybic, T. Blu, and M. Unser, "Generalized sampling: A variational approach—Part I: Theory," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1965–1976, August 2002.

[97] C. de Boor, *A Practical Guide to Splines*, Springer, New York, NY, USA, first edition, 1978.

[98] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*, Springer, Berlin, Germany, first edition, 1988.

[99] J.H. Ahlberg, E.N. Nilson, and J.L. Walsh, "Intrinsic properties of cubic splines," in *The Theory of Splines and their Applications*, vol. 38 of *Mathematics in Science and Engineering*, pp. 75–108. Academic Press, New York, NY, USA, 1967.

[100] M. Unser and T. Blu, "Generalized smoothing splines and the optimal discretization of the Wiener filter," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2146–2159, June 2005.

[101] A. Amini, P. Thévenaz, J.P. Ward, and M. Unser, "On the linearity of Bayesian interpolators for non-Gaussian continuous-time AR(1) processes," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5063–5074, August 2013.

[102] A. Amini and M. Unser, "Optimal interpolation laws for stable AR(1) processes," in *Proceedings of the Tenth International Workshop on Sampling Theory and Applications (SampTA'13)*, Bremen, Germany, July 1-5, 2013, pp. 380–383.

[103] K. Sato, *Lévy Processes and Infinitely Divisible Distributions*, vol. 68 of *Cambridge Studies in Advanced Mathematics*, Cambridge University Press, Cambridge, United Kingdom, November 1999.

[104] M. Unser, P.D. Tafti, A. Amini, and H. Kirshner, "A unified formulation of Gaussian *versus* sparse stochastic processes—Part II: Discrete-domain theory," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 3036–3051, May 2014.

[105] T. Blu and M. Unser, “Self-similarity: Part II—Optimal estimation of fractal processes,” *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1364–1378, April 2007.

[106] M. Unser, P.D. Tafti, and Q. Sun, “A unified formulation of Gaussian versus sparse stochastic processes—Part I: Continuous-domain theory,” *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1945–1962, March 2014.

[107] T.K. Moon and W.C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, Upper Saddle River, NJ, USA, first edition, August 1999.

[108] M. Unser and T. Blu, “Cardinal exponential splines: Part I—Theory and filtering algorithms,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1425–1438, April 2005.

[109] V. Uhlmann, R. Delgado-Gonzalo, C. Conti, L. Romani, and M. Unser, “Exponential Hermite splines for the analysis of biomedical images,” in *Proceedings of the Thirty-Ninth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’14)*, Firenze, Italy, May 4-9, 2014, pp. 1650–1653.

[110] C. Conti, L. Romani, and M. Unser, “Ellipse-preserving Hermite interpolation and subdivision,” *Journal of Mathematical Analysis and Applications*, vol. 426, no. 1, pp. 211–227, June 2015.

[111] Z. Püspöki, V. Uhlmann, C. Vonesch, and M. Unser, “Design of steerable wavelets to detect multifold junctions,” *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 643–657, February 2016.

[112] P. Pad, V. Uhlmann, and M. Unser, “VOW: Variance-optimal wavelets for the steerable pyramid,” in *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP’14)*, Paris, France, October 27-30, 2014, pp. 2973–2977.

[113] P. Pad, V. Uhlmann, and M. Unser, “Maximally localized radial profiles for tight steerable wavelet frames,” *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2275–2287, May 2016.

[114] W.T. Freeman and E.H. Adelson, “Steerable filters for early vision, image analysis, and wavelet decomposition,” in *Proceedings of the Third International Conference on Computer Vision (ICCV’90)*, Osaka, Japan, December 4-7, 1990, pp. 406–415.

[115] E.P. Simoncelli and W.T. Freeman, “The steerable pyramid: A flexible architecture for multi-scale derivative computation,” in *Proceedings of the 1995 IEEE International Conference on Image Processing (ICIP’95)*, Washington D.C., USA, October 23-26, 1995, vol. 3, pp. 3444–3444.

[116] M. Unser, N. Chenouard, and D. Van De Ville, “Steerable pyramids and tight wavelet frames in $L_2(\mathbb{R}^d)$,” *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2705–2721, October 2011.

[117] N. Chenouard and M. Unser, “3D steerable wavelets in practice,” *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4522–4533, November 2012.

[118] D. Slepian, “Prolate spheroidal wave functions, Fourier analysis, and uncertainty-V: The discrete case,” *The Bell System Technical Journal*, vol. 57, no. 5, pp. 1371–1430, May–June 1978.

[119] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger, “Shiftable multiscale transforms,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, March 1992.

[120] M. Unser and N. Chenouard, “A unifying parametric framework for 2D steerable wavelet transforms,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 102–135, January 2013.

[121] M. Papadakis, G. Gogoshin, I.A. Kakadiaris, D.J. Kouri, and D.K. Hoffman, “Nonseparable radial frame multiresolution analysis in multidimensions,” *Numerical Functional Analysis and Optimization*, vol. 24, no. 7-8, pp. 907–928, January 2003.

[122] S. Held, M. Storath, P. Massopust, and B. Forster, “Steerable wavelet frames based on the Riesz transform,” *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 653–667, March 2010.

[123] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[124] J.R. Romero, S.K. Alexander, S. Baid, S. Jain, and M. Papadakis, "The geometry and the analytic properties of isotropic multiresolution analysis," *Advances in Computational Mathematics*, vol. 31, no. 1-3, pp. 283–328, October 2009.

[125] P. Pad and M. Unser, "Optimality of operator-like wavelets for representing sparse AR(1) processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4827–4837, September 2015.

[126] J.P. Ward, P. Pad, and M. Unser, "Optimal isotropic wavelets for localized tight frame representations," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1918–1921, November 2015.

[127] M.D. Buhmann, *Radial Basis Functions*, Cambridge University Press, Cambridge, United Kingdom, 2003.

[128] H. Sagan, *Introduction to the Calculus of Variations*, Dover Publications, New York, NY, USA, 2012.

[129] Y.I. Alber, A.N. Iusem, and M.V. Solodov, "On the projected subgradient method for nonsmooth convex optimization in a Hilbert space," *Mathematical Programming*, vol. 81, no. 1, pp. 23–35, March 1998.

[130] G.B. Folland and A. Sitaram, "The uncertainty principle: A mathematical survey," *Journal of Fourier Analysis and Applications*, , no. 3, pp. 207–238, May 1997.

[131] I. Bayram and I.W. Selesnick, "Frequency-domain design of overcomplete rational-dilation wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 2957–2972, August 2009.

[132] J. Portilla and E.P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, October 2000.

[133] R. Rezakhaniha, A. Agianniotis, J.T.C. Schrauwen, A. Griffa, D. Sage, C.V.C. Bouten, F.N. van de Vosse, M. Unser, and N. Stergiopulos, "Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal laser scanning microscopy," *Biomechanics and Modeling in Mechanobiology*, vol. 11, no. 3–4, pp. 461–473, March 2012.

[134] D. Van De Ville and M. Unser, "Complex wavelet bases, steerability, and the Marr-like pyramid," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2063–2080, November 2008.

[135] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, January 2011.

[136] B. Beck, "Design of optimal 2D steerable filters: Theory and implementation," M.S. thesis, École polytechnique fédérale de Lausanne, Lausanne, VD, Switzerland, 2017.

[137] V. Uhlmann and M. Unser, "Tip-seeking active contours for bioimage segmentation," in *Proceedings of the Twelfth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'15)*, Brooklyn NY, USA, April 16-19, 2015, pp. 544–547.

[138] V. Uhlmann and M. Unser, "Snakes with tangent-based control and energies for bioimage analysis," in *Proceedings of the Eleventh IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'14)*, Beijing, China, April 29-May 2, 2014, pp. 806–809.

[139] G.B. Arfken, *Mathematical Methods for Physicists*, Academic Press, Orlando, FL, USA, third edition, 1985.

[140] R. Ronfard, "Region-based strategies for active contour models," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 229–251, October 1994.

[141] O. Henricsson and W. Neuenschwander, "Controlling growing snakes by using key-points," in *Proceedings of the Twelfth IEEE International Conference on Pattern Recognition (IAPR'94)*, Jerusalem, Israel, October 9-13, 1994, vol. 1, pp. 68–73.

[142] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, United Kingdom, third edition, 1986.

[143] E. Bostan, E. Froustey, B. Rappaz, É. Shaffer, D. Sage, and M. Unser, "Phase retrieval by using transport-of-intensity equation and differential interference contrast microscopy," in *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP'14)*, Paris, France, October 27-30, 2014, pp. 3939–3943.

[144] A. Amsterdam and J.D. Jamieson, "Structural and functional characterization of isolated pancreatic exocrine cells," *Proceedings of the National Academy of Sciences*, vol. 69, no. 10, pp. 3028–3032, October 1972.

[145] W.H. Fahrenbach and D.D. Knutson, "Surface adaptations of the vertebrate epidermis to friction," *Journal of Investigative Dermatology*, vol. 65, no. 1, pp. 39–44, June 1975.

[146] V. Ljosa, K.L. Sokolnicki, and A.E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nature Methods*, vol. 9, no. 7, pp. 637, June 2012.

[147] M.J. Galko and M.A. Krasnow, "Cellular and genetic analysis of wound healing in *Drosophila* larvae," *PLoS Biology*, vol. 2, no. 8, pp. e239, July 2004.

[148] S.R. Olsen and R.I. Wilson, "Cracking neural circuits in a tiny brain: new approaches for understanding the neural circuitry of *Drosophila*," *Trends in Neurosciences*, vol. 31, no. 10, pp. 512–520, October 2008.

[149] L.P.J.J. Noldus, A.J. Spink, and R.A.J. Tegelenbosch, "Computerised video tracking, movement analysis and behaviour recognition in insects," *Computers and Electronics in Agriculture*, vol. 35, no. 2, pp. 201–227, August 2002.

[150] H. Dankert, L. Wang, E.D. Hoopfer, D.J. Anderson, and P. Perona, "Automated monitoring and analysis of social behavior in *Drosophila*," *Nature Methods*, vol. 6, no. 4, pp. 297–303, April 2009.

[151] K. Branson, A.A. Robie, J. Bender, P. Perona, and M.H. Dickinson, "High-throughput ethomics in large groups of *Drosophila*," *Nature Methods*, vol. 6, no. 6, pp. 451–457, June 2009.

[152] N. Donelson, E.Z. Kim, J.B. Slawson, C.G. Vecsey, R. Huber, and L.C. Griffith, "High-resolution positional tracking for long-term analysis of *Drosophila* sleep and locomotion using the "tracker" program," *PLoS One*, vol. 7, no. 5, pp. e37250, May 2012.

[153] A. Pérez-Escudero, J. Vicente-Page, R.C. Hinz, S. Arganda, and G.G. De Polavieja, "idTracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature Methods*, vol. 11, no. 7, pp. 743–748, July 2014.

[154] Y. Deng, P. Coen, M. Sun, and J.W. Shaevitz, "Efficient multiple object tracking using mutually repulsive active membranes," *PLoS One*, vol. 8, no. 6, pp. e65769, June 2013.

[155] J. Kain, C. Stokes, Q. Gaudry, X. Song, J. Foley, R. Wilson, and B. de Bivort, "Leg-tracking and automated behavioural classification in *Drosophila*," *Nature Communications*, vol. 4, pp. 1910, May 2013.

[156] J.A. Bender, E.M. Simpson, and R.E. Ritzmann, "Computer-assisted 3D kinematic analysis of all leg joints in walking insects," *PLoS One*, vol. 5, no. 10, pp. e13617, October 2010.

[157] C.S. Mendes, I. Bartos, T. Akay, S. Márka, and R.S. Mann, "Quantification of gait parameters in freely walking wild type and sensory deprived *Drosophila melanogaster*," *eLife*, vol. 2, pp. e00231, January 2013.

[158] A. Isakov, S.M. Buchanan, B. Sullivan, A. Ramachandran, J.K.S. Chapman, E.S. Lu, L. Mahadevan, and B. de Bivort, "Recovery of locomotion after injury in *Drosophila melanogaster* depends on proprioception," *Journal of Experimental Biology*, vol. 219, no. 11, pp. 1760–1771, June 2016.

[159] G.J. Berman, D.M. Choi, W. Bialek, and J.W. Shaevitz, "Mapping the stereotyped behaviour of freely moving fruit flies," *Journal of The Royal Society Interface*, vol. 11, no. 99, pp. 20140672, October 2014.

[160] G.J. Berman, W. Bialek, and J.W. Shaevitz, "Predictability and hierarchy in *Drosophila* behavior," *Proceedings of the National Academy of Sciences*, vol. 113, no. 42, pp. 11943–11948, October 2016.

[161] S. Pick and R. Strauss, "Goal-driven behavioral adaptations in gap-climbing *Drosophila*," *Current Biology*, vol. 15, no. 16, pp. 1473–1478, August 2005.

[162] A.M. Seeds, P. Ravbar, P. Chung, S. Hampel, F.M. Midgley, B.D. Mensh, and J.H. Simpson, "A suppression hierarchy among competing motor programs drives sequential grooming in *Drosophila*," *eLife*, vol. 3, pp. e02951, August 2014.

[163] V. Uhlmann, P. Ramdya, R. Delgado-Gonzalo, R. Benton, and M. Unser, "FlyLimbTracker: An active contour based approach for leg segment tracking in unmarked, freely behaving *drosophila*," *PLoS One*, vol. 12, no. 4, pp. e0173433, April 2017.

[164] F. de Chaumont, R. Dos-Santos Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J.-C. Olivo-Marin, "Computerized video analysis of social interactions in mice," *Nature Methods*, vol. 9, no. 4, pp. 410–417, April 2012.

[165] N. Chenouard, J. Buisson, I. Bloch, P. Bastin, and J.-C. Olivo-Marin, "Curvelet analysis of kymograph for tracking bi-directional particles in fluorescence microscopy images," in *Proceedings of the Seventh IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'10)*, Rotterdam, Netherlands, April 14-17, 2010, pp. 3657–3660.

[166] R. Delgado-Gonzalo, D. Schmitter, V. Uhlmann, and M. Unser, "Efficient shape priors for spline-based snakes," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3915–3926, November 2015.

[167] E.W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959.

[168] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[169] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, vol. 8, no. 19, pp. 415–428, August 2012.

[170] M. Raviglione and G. Sulis, "Tuberculosis 2015: burden, challenges and strategy for control and elimination," *Infectious Disease Reports*, vol. 8, no. 2, June 2016.

[171] J.D. McKinney, "In vivo veritas: the search for TB drug targets goes live," *Nature Medicine*, vol. 6, no. 12, pp. 1330–1333, April 2000.

[172] M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain, "Stochastic gene expression in a single cell," *Science*, vol. 297, no. 5584, pp. 1183–1186, August 2002.

[173] Y. Wakamoto, N. Dhar, R. Chait, K. Schneider, F. Signorino-Gelo, S. Leibler, and J.D. McKinney, "Dynamic persistence of antibiotic-stressed mycobacteria," *Science*, vol. 339, no. 6115, pp. 91–95, January 2013.

[174] Q. Wu, F. Merchant, and K. Castleman, *Microscope Image Processing*, Academic Press, New York, NY, USA, 2010.

[175] A. Ducret, E.M. Quardokus, and Y.V. Brun, "MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis," *Nature Microbiology*, vol. 1, pp. 16077, July 2016.

[176] O. Sliusarenko, J. Heinritz, T. Emonet, and C. Jacobs-Wagner, "High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics," *Molecular Microbiology*, vol. 80, no. 3, pp. 612–627, May 2011.

[177] A. Paintdakhi, B. Parry, M. Campos, I. Irnov, J. Elf, I. Surovtsev, and C. Jacobs-Wagner, "Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis," *Molecular Microbiology*, vol. 99, no. 4, pp. 767–777, February 2016.

[178] N.O.E. Vischer, J. Verheul, M. Postma, B. van den Berg van Saparoea, E. Galli, P. Natale, K. Gerdes, J. Luirink, W. Vollmer, M. Vicente, and T. den Blaauwen, "Cell age dependent concentration of *Escherichia coli* divisome proteins analyzed with ImageJ and ObjectJ," *Frontiers in Microbiology*, vol. 6, pp. 586, June 2015.

[179] J.W. Young, J.C.W. Locke, A. Altinok, N. Rosenfeld, T. Bacarian, P.S. Swain, E. Mjolsness, and M.B. Elowitz, "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy," *Nature Protocols*, vol. 7, no. 1, pp. 80–88, December 2011.

[180] S. Chowdhury, M. Kandhavelu, O. Yli-Harja, and A.S. Ribeiro, "Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE)," *BMC Bioinformatics*, vol. 14, no. 10, pp. S8, August 2013.

[181] I. Santi, N. Dhar, D. Bousbaine, Y. Wakamoto, and J.D. McKinney, "Single-cell dynamics of the chromosome replication and cell division cycles in mycobacteria," *Nature Communications*, vol. 4, pp. 2470, September 2013.

[182] I. Mekterović, D. Mekterović, and Z. Maglica, "BactImAS: a platform for processing and analysis of bacterial time-lapse microscopy movies," *BMC Bioinformatics*, vol. 15, no. 1, pp. 251, July 2014.

[183] F. Jug, T. Pietzsch, D. Kainmüller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, and G. Myers, "Optimal joint segmentation and tracking of *Escherichia coli* in the mother machine," in *Proceedings of the First International Workshop on Bayesian and grAphical Models for Biomedical Imaging (BAMBI'14)*, Cambridge, MA, USA, September 18 2014, pp. 25–36.

[184] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F.A. Hamprecht, "Graphical model for joint segmentation and tracking of multiple dividing cells," *Bioinformatics*, vol. 31, no. 6, pp. 948–956, March 2015.

[185] E. Turetken, X. Wang, C.J. Becker, C. Haubold, and P. Fua, "Network flow integer programming to track elliptical cells in time-lapse sequences," *IEEE Transactions on Medical Imaging*, April 2016.

[186] S.K. Sadanandan, Ö. Baltekin, K.E.G. Magnusson, A. Boucharin, P. Ranefall, J. Jaldén, J. Elf, and C. Wählby, "Segmentation and track-analysis in time-lapse imaging of bacteria," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 1, pp. 174–184, February 2016.

[187] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, NY, USA, 1998.

[188] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[189] Z. Tu, “Auto-context and its application to high-level vision tasks,” in *Proceedings of the Twenty-First IEEE Conference on Computer Vision and Pattern Recognition (CVPR’08)*, Anchorage, AK, USA, June 23-28, 2008, pp. 1–8.

[190] C. Haubold, V. Uhlmann, M. Unser, and F. Hamprecht, “Diverse M-best solutions by dynamic programming,” in *Proceedings of the Thirty-Ninth German Conference on Pattern Recognition (GCPR’17)*, Basel, Switzerland, September 13-15, 2017.

[191] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, February 2001.

[192] P. Matula, M. Maška, D.V. Sorokin, P. Matula, C. Ortiz de Solórzano, and M. Kozubek, “Cell tracking accuracy measurement based on comparison of acyclic oriented graphs,” *PLoS One*, vol. 10, no. 12, pp. e0144959, December 2015.

[193] L. Shamir, “Assessing the efficacy of low-level image content descriptors for computer-based fluorescence microscopy image analysis,” *Journal of Microscopy*, vol. 243, no. 3, pp. 284–292, September 2011.

[194] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, “Support vector machine learning for interdependent and structured output spaces,” in *Proceedings of the Twenty-First International Conference on Machine Learning (ICML’04)*, Banff, AB, Canada, July 4-8, 2004, pp. 104–111.

[195] P. Yadollahpour, D. Batra, and G. Shakhnarovich, “Discriminative re-ranking of diverse segmentations,” in *Proceedings of the Twenty-Sixth IEEE Conference on Computer Vision and Pattern Recognition (CVPR’13)*, Portland, OR, USA, June 25-27, 2013, pp. 1923–1930.

[196] J. Funke, F.A. Hamprecht, and C. Zhang, “Learning to segment: Training hierarchical segmentation under a topological loss,” in *Proceedings of the Eighteenth International Conference on Medical Image Computing and*

Computer-Assisted Intervention (MICCAI'15), Munich, Germany, October 5-9, 2015, pp. 268–275.

[197] S.D. Buckingham and D.B. Sattelle, "Strategies for automated analysis of *C. elegans* locomotion," *Invertebrate Neuroscience*, vol. 8, no. 3, pp. 121–131, August 2008.

[198] A.K. Corsi, B. Wightman, and M. Chalfie, "A transparent window into biology: A primer on *Caenorhabditis elegans*," *Genetics*, vol. 200, no. 2, pp. 387–407, June 2015.

[199] J.C. Gray and A.D. Cutter, "Mainstreaming *Caenorhabditis elegans* in experimental evolution," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 281, no. 1778, pp. 20133055, March 2014.

[200] G. Tsechpenakis, L. Bianchi, D.N. Metaxas, and M. Driscoll, "A novel computational approach for simultaneous tracking and feature extraction of *C. elegans* populations in fluid environments," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1539–1549, May 2008.

[201] E. Yemini, T. Jucikas, L.J. Grundy, A.E. Brown, and W.R. Schafer, "A database of *Caenorhabditis elegans* behavioral phenotypes," *Nature Methods*, vol. 10, no. 9, pp. 877–879, July 2013.

[202] Y. Koren, R. Sznitman, P.E. Arratia, C. Carls, P. Krajacic, A.E. Brown, and J. Sznitman, "Model-independent phenotyping of *C. elegans* locomotion using scale-invariant feature transform," *PLoS One*, vol. 10, no. 3, pp. e0122326, 2015.

[203] G.J. Stephens, B. Johnson-Kerner, W. Bialek, and W.S. Ryu, "Dimensionality and dynamics in the behavior of *C. elegans*," *PLoS Computational Biology*, vol. 4, no. 4, pp. e1000028, April 2008.

[204] L.P. O'Reilly, C.J. Luke, D.H. Perlmutter, G.A. Silverman, and S.C. Pak, "*C. elegans* in high-throughput drug discovery," *Advanced Drug Delivery Reviews*, vol. 69–70, pp. 247–253, April 2014.

[205] S.J. Husson, W.S. Costa, C. Schmitt, and A. Gottschalk, "Keeping track of worm trackers," in *WormBook*, WormBook, chapter 5. The *C. elegans* Research Community, http://www.wormbook.org, September 2013.

[206] A. San-Miguel and H. Lu, “Microfluidics as a tool for *C. elegans* research,” in *WormBook*, WormBook, chapter 5. The *C. elegans* Research Community, http://www.wormbook.org, September 2013.

[207] D. Ramot, B.E. Johnson, T.L. Berry, L. Carnell, and M.B. Goodman, “The parallel worm tracker: a platform for measuring average speed and drug-induced paralysis in nematodes,” *PLoS One*, vol. 3, no. 5, pp. e2208, 2008.

[208] S.D. Buckingham and D.B. Sattelle, “Fast, automated measurement of nematode swimming (thrashing) without morphometry,” *BMC Neuroscience*, vol. 10, no. 1, pp. 84, July 2009.

[209] Z. Feng, C.J. Cronin, J.H. Wittig, P.W. Sternberg, and W.R. Schafer, “An imaging system for standardized quantitative analysis of *C. elegans* behavior,” *BMC Bioinformatics*, vol. 5, no. 1, pp. 115, August 2004.

[210] W. Geng, P. Cosman, C.C. Berry, Z. Feng, and W.R. Schafer, “Automatic tracking, feature extraction and classification of *C. elegans* phenotypes,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1811–1820, October 2004.

[211] E. Fontaine, J. Burdick, and A. Barr, “Automated tracking of multiple *C. elegans*,” in *Proceedings of the Twenty-Eighth International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'06)*, New York, NY, USA, August 30-September 3, 2006, pp. 3716–3719.

[212] N.A. Swierczek, A.C. Giles, C.H. Rankin, and R.A. Kerr, “High-throughput behavioral analysis in *C. elegans*,” *Nature Methods*, vol. 8, no. 7, pp. 592–598, July 2011.

[213] D.G. Kendall, “Shape manifolds, procrustean metrics, and complex projective spaces,” *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984.

[214] T. Cootes, C. Taylor, D. Cooper, and J. Graham, “Active shape models-their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, January 1995.

[215] I. Dryden and K. Mardia, *Statistical Shape Analysis*, John Wiley & Sons, New York, NY, USA, 1998.

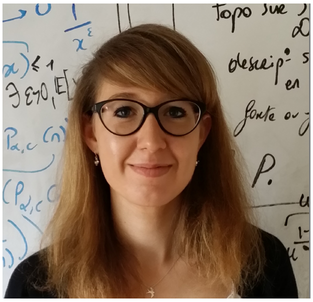
[216] M. Unser, B.L. Trus, and A.C. Steven, "Normalization procedures and factorial representations for classification of correlation-aligned images: A comparative study," *Ultramicroscopy*, vol. 30, no. 3, pp. 299–310, July-August 1989.

[217] D. Schmitter and M. Unser, "Similarity-based shape priors for 2D spline snakes," in *Proceedings of the Twelfth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'15)*, Brooklyn NY, USA, April 16-19, 2015, pp. 1216–1219.

[218] A. Badoual, D. Schmitter, and M. Unser, "An inner-product calculus for periodic functions and curves," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 878–882, June 2016.

Curriculum Vitæ

Virginie Uhlmann
Ch. Du Trabandan 43
CH-1006 Lausanne
Switzerland
uhlmann.virginie@gmail.com
(+41) 79 726 83 58



Swiss citizen
Born on 23.03.1989

EDUCATION

2012 - 2017 École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland

PhD in Electrical Engineering

Project: Landmark Active Contours for Bioimage Analysis: A Tale of Points and Curves

Advisor: Prof. Michael Unser

2016 Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

Collaboration project at the Heidelberg Collaboratory for Image Processing (HCI) in Prof. Fred Hamprecht's group

2011 - 2012 Broad Institute of MIT and Harvard, Cambridge, MA, USA

Master project at the Imaging Platform

Project: A segmentation-free image classifier for biological applications

Advisor: Dr. Anne Carpenter and Prof. John McKinney

2010 - 2012 École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland

MSc in Bioengineering

2007 - 2009 École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland

BSc in Life Sciences and Technology

HONOR AND AWARDS

2014	**Best student paper** (ICIP award)
2013	**Best teaching team** (EPFL School of Life Sciences award)
2012	**Best grade point average** for complete Master studies in EPFL School of Engineering ("SIA Vaudoise – Ingénieurs" award)
2012	**Best Master project** in Biomedical Technologies (EPFL award)
2012	**Best grade point average** in Bioengineering Master (EPFL award)
2010-2012	**Excellence fellowship** for Master studies (EPFL fellowship)

RESEARCH INTERESTS

Approximation Theory	Interpolation Methods Spline Theory
Image Processing	Image Segmentation *Deformable models* Object Recognition *Filtering, statistical and learning-based approaches* Feature Detection *Filter design, machine learning* Object Tracking *Shortest path approaches, graphical models*
Applications to Bioimage Analysis	Translation of Mathematical Approaches into Softwares User-friendly Implementation of Image Analysis Methods Development of Custom Image Analysis Pipelines

PUBLICATIONS

Journal:

1. **V. Uhlmann**, C. Haubold, F.A. Hamprecht, M. Unser, "DiversePathsJ: Diverse Shortest Paths for Bioimage Analysis," *Bioinformatics*, in press.

2. **V. Uhlmann**, P. Ramdya, R. Delgado-Gonzalo, R. Benton, M. Unser, "FlyLimbTracker: An Active Contour Based Approach for Leg Segment Tracking in Unmarked, Freely Behaving Drosophila," *PLoS ONE*, vol. 12, no. 4, pp. 1-21, April 28, 2017.

3. J. Fageot, **V. Uhlmann**, M. Unser, "Gaussian and Sparse Processes Are Limits of Generalized Poisson Processes," under review [*preprint available at arXiv:1702.05003v1 [math.PR]*].

4. **V. Uhlmann**, S. Singh, A.E. Carpenter, "CP-CHARM: Segmentation-Free Image Classification Made Accessible," *BMC Bioinformatics*, vol. 17, no. 1, paper no. 51, December 2016.

5. **V. Uhlmann**, J. Fageot, M. Unser, "Hermite Snakes with Control of Tangents," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2803-2816, June 2016.

6. A. Badoual, D. Schmitter, **V. Uhlmann**, M. Unser, "Multiresolution Subdivision Snakes," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1188-1201, March 2017.

7. P. Pad, **V. Uhlmann**, M. Unser, "Maximally Localized Radial Profiles for Tight Steerable Wavelet Frames," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp 2275-2287, May 2016 [*joint first authorship*].

8. Z. Püspöki, V. Uhlmann, C. Vonesch, M. Unser, "Design of Steerable Wavelets to Detect Multifold Junctions," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 643-657, February 2016.

9. R. Delgado-Gonzalo, D. Schmitter, V. Uhlmann, M. Unser, "Efficient Shape Priors for Spline-Based Snakes," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3915-3926, November 2015.

10. R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, M. Unser, "Snakes on a Plane: A Perfect Snap for Bioimage Analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41-48, January 2015.

Conference, symposium and workshops:

1. C. Haubold, V. Uhlmann, M. Unser, F. Hamprecht, "Diverse *M*-best solutions by Dynamic Programming," *to appear in* Proceedings of the Thirty-ninth German Conference on Pattern Recognition (GCPR'17), Basel, Switzerland, September 13-15, 2017 [*nominated for best paper award*].

2. V. Uhlmann, M. Unser, "Bioimage Segmentation with Active Contours," Fourth Swiss Imaging Core Facility Day, Geneva, Switzerland, September 16, 2016.

3. V. Uhlmann, M. Unser, "Spline-based models for image segmentation," First European Machine Vision Forum, Heidelberg, Germany, September 8-9, 2016.

4. V. Uhlmann, D. Schmitter, M. Unser, "Shape-constrained Tracking with Active Contours," SIAM Conference on Imaging Science (IS'16), Albuquerque, NM, USA, May 23-26, 2016.

5. V. Uhlmann, R. Delgado-Gonzalo, M. Unser, P.O. Michel, L. Baldi, F.M. Wurm, "User-Friendly Image-Based Segmentation and Analysis of Chromosomes," Proceedings of the Thirteenth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'16), Prague, Czech Republic, April 13-16, 2016, pp. 395-398.

6. V. Uhlmann, J. Fageot, H. Gupta, M. Unser, "Statistical Optimality of Hermite Splines," Proceedings of the Eleventh International Workshop on Sampling Theory and Applications (SampTA'15), Washington DC, USA, May 25-29, 2015, pp. 226-230.

7. V. Uhlmann, D. Sage, M. Unser, "Directional Image Analysis in Icy," Second Icy Coding Party, Paris, France, May 4-6, 2015.

8. V. Uhlmann, M. Unser, "Tip-Seeking Active Contours for Bioimage Segmentation," Proceedings of the Twelfth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'15), Brooklyn NY, USA, April 16-19, 2015, pp. 544-547 [*nominated for best paper award*].

9. V. Uhlmann, M. Unser, "Open Active Contours with Tangent-Based Controls and Energies, " First International Conference on Subdivision, Geometric and Algebraic Methods, Isogeometric Analysis and Refinability in Tuscany (SMART'14), Pontignano, Italy, September 28-October 1, 2014.

10. P. Pad, V. Uhlmann, M. Unser, "VOW: Variance-Optimal Wavelets for the Steerable Pyramid," Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP'14), Paris, France, October 27-30, 2014, pp. 2973-2977.

11. C. Conti, L. Romani, **V. Uhlmann**, M. Unser, "Cardinal Hermite Exponential Splines: Theoretical Insights and Applications to Active Contours," Proceedings of the Eighth International Conference Curves and Surfaces (ICCS'14), Paris, France, June 12-18, 2014, pp. 74.

12. **V. Uhlmann**, R. Delgado-Gonzalo, C. Conti, L. Romani, M. Unser, "Exponential Hermite Splines for the Analysis of Biomedical Images," Proceedings of the Thirty-Ninth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14), Firenze, Italy, May 4-9, 2014, pp. 1650-1653.

13. **V. Uhlmann**, R. Delgado-Gonzalo, M. Unser, "Snakes with Tangent-Based Control and Energies for Bioimage Analysis," Proceedings of the Eleventh IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'14), Beijing, China, April 29-May 2, 2014, pp. 806-809

14. O. Mariani, **V. Uhlmann**, D. Sage, M. Unser, "BisQuit: Open-Source Interactive Tool for Image Sequence Analysis," Second Swiss Image-Based Screening Conference (SIBSC'13), Lausanne, Switzerland, June 10-11, 2013

15. **V. Uhlmann**, S. Singh, C. Wählby, J. McKinney, A. Carpenter, "Whole Image-Based Classification Using CellProfiler," Proceedings of BioImage Informatics 2012 (BI'12), Dresden, Germany, September 16-19, 2012, pp. 59-60.

TEACHING

Teaching Assistant:

Signals and Systems I & II (Autumn 2009 and Autumns 2012 to 2016)

Students Supervised:

2017 Julien Rüegg (co-advised with J. Fageot and Prof. M. Unser)
Semester Project: A study of Hermite spline interpolation

2015 Gaëlle Thurre (co-advised with Prof. John McKinney)
Semester Project: Modeling Single-Cell Behavior of Mycobacteria

2015 Bertrand Vermot (co-advised with J. Fageot and Prof. M. Unser) *Semester Project:* Corner Detection based on PDE Methods

2015 Miryam Chaabouni (co-advised with J. Fageot, D. Schmitter and Prof. M. Unser)
Semester Project: Design of active contour models using NURBS

2015 Arik Girsault (co-advised with E. Bostan and Prof. M. Unser)
Semester Project: A nonlinear model of Phase Contrast Microscopy

2014 Lucas Amoudruz (co-advised with Prof. M. Unser)
Semester Project: Refining leg-movement extraction in high-speed videos of insect locomotion

2014 Lucas Vandroux (co-advised with P. Pad and Prof. M. Unser)
Semester Project: Detection and enhancement of blood flow in a video

2014 Amicie de Pierrefeu (co-advised with D. Sage, and Prof. M. Unser) *Semester Project:* Tracking flagella undulations in microscopy images

2014 Laurène Donati (co-advised with Prof. M. Unser)
Semester Project: Modeling Mycobacteria behavior in time-lapse microscopy images

2014 Pritish Chakravarty (co-advised with Prof. M. Unser)
Semester Project: Designing Tip-Detectors

2013 Loïc Perruchoud (co-advised with Dr. C. Vonesch and Prof. M. Unser) *Semester Project:* Tracking leg movements in high-speed videos of insect locomotion

2013 Olivia Mariani (co-advised with Prof. M. Unser)
Master Project: Segmentation and tracking of growing colony of mycobacteria

PROFESSIONAL EXPERIENCE

2012 - ongoing **Researcher and teaching assistant** at the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL)

Developed theory and software solutions for segmentation using active contours in biomedical imaging. Defined and supervised research projects, all of which involved the design and implementation of specific tools for image processing applications. Some projects were conducted in collaboration with other EPFL labs.

2010 - 2011 **Research intern** at the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL)

In collaboration with Prof. J. McKinney's lab at EPFL, initiated the development of a semi-automated image analysis pipeline for segmenting and tracking Mycobacteria in phase contrast images. This project gave birth to three student projects and a large-scale collaborative project with Prof. F. Hamprecht's group at the University of Heidelberg.

2009 **Summer intern** at the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL)

Designed a library of keynote presentations to illustrate practical examples of applications of signal processing in life sciences for Prof. M. Unser's Signals and Systems class.

PROFESSIONAL ACTIVITIES

Reviewer for:

IEEE Transactions on Image Processing
IEEE Signal Processing Magazine
IEEE Signal Processing Letters
Journal of Computational and Applied Mathematics
Signal Processing: Image Communication
IEEE/ACM Transactions on Computational Biology and Bioinformatics
The Visual Computer (International Journal of Computer Graphics)
Journal of Mathematical Imaging and Vision
IEEE ISBI 2013, 2014, 2015, 2016, 2017
SampTA 2015
IEEE GlobalSIP 2015
MICCAI 2016 (Brainles Workshop)

Journal of Histochemistry & Cytochemistry

Microscopy Research and Technique

TECHNICAL SKILLS

Software Development:	Java, Python, C/C++, Perl
Scientific Computing:	Matlab, Mathematica, Maple
Image Processing:	ImageJ, Fiji, Icy, CellProfiler
Graphical Design:	Photoshop, Illustrator

LANGUAGES

French: Native

English: Fluent

German: B2 level

PERSONAL

Since 2007, active member of Valdésia, hundred-year-old student society in Lausanne

Consumer of French, German and English contemporary and classic literature

Amateur cartoonist