
Subdivide and Conquer: Active Contours and Surfaces for Biomedical Image Segmentation

Anais Badoual

Thèse N° 9326 (avril 2019)

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

Prof. Pierre Vandergheynst, *président*
Prof. Michael Unser, *directeur de thèse*
Dr. Daniel Sage, *co-directeur de thèse*
Prof. Lucia Romani, *rapporteur*
Prof. Mario Figueiredo, *rapporteur*
Prof. Dimitri Van De Ville, *rapporteur*

École polytechnique fédérale de Lausanne—2019

Cover design by Annette Unser
Printing and binding by Repro-EPFL
Typeset with L^AT_EX
Copyright © 2019 by Anaïs Badoual
Available at <http://bigwww.epfl.ch/>

Abstract

Ongoing advances in imaging techniques create new demands regarding the analysis of images in medicine and biology. Image segmentation is a key step of many image analysis pipelines and its proper execution is a particularly challenging task. This thesis is dedicated to the development of segmentation algorithms for biomedical structures in 2D and 3D images. In this work, we aim to improve upon classical parametric active contours/surfaces, whose limitations we address.

This thesis is organized in three parts. First, we introduce two representation models. They adapt their resolution to the level of detail of the object to be segmented. We then focus on the formulation of cost functions, called energies, that guide the curve/surface toward the boundary of the target in the image. Among others, we present novel energies based on ridge and texture information. Finally, with these two ingredients in hand, we design new semi-automated active contours/surfaces, also called snakes, for various applications. Our methods are generic enough to be used with a broad variety of data. In particular, we illustrate their performance in segmenting real biomedical images. In addition to those three parts, we provide mathematical tools for signal processing that we designed to efficiently process periodic functions.

To find the optimal curve/surface that best fits a given target, we adopt throughout the thesis a *subdivide and conquer* strategy. First, we look at several smoothed versions of the original image and, for each, adapt the resolution of the snake curve/surface to the level of detail of the target (*i.e.*, subdivide). Then, we segment the object of interest at each resolution recursively, from the coarsest to the finest image (*i.e.*, conquer). This robust subdivide and conquer strategy exploits the multiresolution property of our curve/surface representations, as well as characteristics of smoothed images (few details and low noise).

Finally, we give a special attention to the conversion of our algorithms into usable software that respect the open-source, user-friendly and reproducibility criteria.

Keywords: Biomedical image analysis, segmentation, active contours, active surfaces, geometric representation, parametrization, local refinement, subdivision, multiresolution, energy, ridge, texture, usable software, splines, inner product.

Résumé

Les nouvelles techniques d'imagerie ont donné naissance à de nouvelles exigences concernant l'analyse d'images biomédicales. La segmentation constitue une étape clé de l'analyse d'images dont la bonne exécution est une tâche particulièrement ardue. Cette thèse se consacre au développement d'algorithmes pour la segmentation de structures biomédicales dans des images 2D et 3D. A travers ce travail, nous aspirons notamment à améliorer les méthodes dites classiques de modèles déformables paramétriques, en identifiant leurs limitations.

Cette thèse est organisée en trois parties. Dans un premier temps, nous introduisons deux modèles de représentation qui adaptent leur résolution au niveau de détail de l'objet à segmenter. Ensuite, nous formulons des fonctions de coût, appelées *énergies*, qui permettent de guider nos courbes/surfaces vers le pourtour des objets d'intérêt dans des images. Entre autres, nous présentons de nouvelles énergies basées sur la détection de texture et de lignes de crêtes. Enfin, avec ces deux éléments en main, nous concevons de nouveaux modèles déformables semi-automatisés, également appelés *snakes*. Nos approches sont suffisamment générales pour être utilisées avec de nombreux types de données. En particulier, nous illustrons leurs performances pour la segmentation d'images biomédicales réelles. En plus de ces trois parties, nous fournissons des outils mathématiques de traitement du signal, conçus pour traiter efficacement des fonctions périodiques.

Pour trouver la courbe/surface qui délimite au mieux le pourtour de l'objet d'intérêt, nous adoptons tout au long de cette thèse une stratégie qui consiste à *subdiviser pour régner*. Tout d'abord, nous réalisons plusieurs versions lissées de l'image originale. Pour chacune d'elles, nous adaptons la résolution du snake au niveau de détail de l'objet à segmenter (*i.e.*, subdiviser). Ensuite, nous segmentons récursivement l'objet d'intérêt à chaque niveau de résolution, en allant de l'image

la plus grossière à la plus fine (*i.e.*, régner). Cette stratégie est robuste et exploite la propriété de multirésolution de nos représentations, ainsi que les caractéristiques des images lissées qui contiennent peu de détails et sont peu bruitées.

Enfin, nous portons une attention toute particulière à convertir nos algorithmes en logiciels qui soient libres d'accès, faciles d'utilisation et reproductibles.

Mots clefs : Analyse d'images biomédicales, segmentation, contours actifs, surfaces actives, représentation géométrique, paramétrisation, raffinement local, subdivision, multirésolution, énergie, ligne de crête, texture, logiciels utilisateur, splines, produit scalaire.

Nothing great is achieved without chimeras.
– Ernest Renan

Acknowledgement

This thesis is the fruit of four years of research and of a true personal life experience. This thesis would not have been possible without the help and support of many people. I take this opportunity to express my sincere gratitude to all of them.

First, I would like to thank the president of the thesis jury, Prof. Pierre Vandergheynst, and the committee members, Prof. Lucia Romani, Prof. Dimitri Van De Ville, and Prof. Mario Figueiredo, for their time and for reviewing my thesis. I also address my gratitude to the Swiss National Science Foundation for funding my research.

I would like to thank two persons who have made this thesis a reality. I owe a lot to Christophe Lallement for believing in me and encouraging me to apply for a PhD at EPFL. I would then like to sincerely thank my thesis advisor, Prof. Michael Unser, for his guidance throughout these four years. Michael is a great advisor and a dedicated teacher. I particularly appreciated his enthusiasm for research and that his door was always wide open for discussion. Working with him was a great opportunity for me and allowed me to grow as a scientist.

I was very fortunate to do my PhD within the Biomedical Imaging Group (BIG). It is a truly pleasant workplace promoting scientific exchanges and collaborations. Overall, BIG is a place of sharing and friendships. The meeting room welcomed in turn pictictionary games, movie nights, apéros and raclette parties (and meetings occasionally). Not to mention, the lab outings that conducted the entire team to share ski, hiking, rafting or via ferrata experiences. I want to warmly thank all BIG members (too many for me to list here), past and present, for all these joyful

moments.

Beyond Michael, the BIG is carefully worned by Philippe Thévenaz and Daniel Sage. Thank you, Philippe, for your careful reading of my scientific papers. This greatly improved the quality of my work and allows me to improve my English skills as well. Daniel Sage (a.k.a. Lucky Luke a.k.a. Columbo) perfectly fulfilled his role of co-director of my thesis. He was always available for advices, help and collaborations especially for student projects. His door too was always open for a coffee break or to work on Caffe. But more important for me, he lent a listening ear every time I needed to talk about the joys and difficulties that I encountered in my work as well as in my personal life. In a nutshell, kindness, benevolent and young minded are the best words to describe him. Thank you, Daniel, for all of this. I am also very grateful for all the laughter that you contributed to (ski with one stick, laser game, Vallée Blanche, pétanque parties). I thank as well Manuelle, Beatrice, Nadia, Monika and Claudia for their help in all the administrative tasks as well as in the organization of the lab outings. A big thank goes to my office mates, Thomas D. and Shayan, for creating such a good working atmosphere; including, but not limited to, supporting my jokes.

I would like to thank all my collaborators, who contributed a lot to this thesis. Inside the BIG, I had the chance to work with Virginie, Daniel Schmitter, Daniel Sage, and Adrien to bring new assets into snakes. Julien guided me in the discovery of the periodic world of Gaussian processes. I have been also very fortunate to work with Lucia Romani from the University of Bologna and Paola Novara. They helped me to discover with enthusiasm the mathematics of subdivisions. I also want to thank Raoul Schorer from Geneva University Hospital for bringing medical applications into my thesis.

During this thesis, I took part in several teaching activities. I had the opportunity to supervised enthusiastic students. It was truly rewarding to guide them to conduct their own research. I would like to particularly thank Axel, Brune, Yann and Aymeric for their investment which led to publications. I particularly enjoyed being involved into the teaching organization for the Signaux & Systèmes class attended by Bachelor students. With the so called “dream team” of teaching assistants we invested ourselves week after week. In particular, I would like to deeply thank my BIG colleagues Virginie, Laurène, Julien, Thomas D., Toulouse, as well as the LTS2 members Johan, Yann, Michael D., Lionel, for making the assistantship such

a great time, but also for all the moments and memories we shared in Brussels, seminars or sitting around a game of Esquissé.

A special brand of thanks goes to Daniel Schmitter, Julien and Laurène, who were truly precious these last four years. When I first arrived in Lausanne, Daniel Schmitter took really good care of me, so I could make a place for myself both professionally and personally. I deeply thank you Daniel for this. My biggest debt of gratitude goes to Laurène and Julien. This thesis would have been so much different without you, your support and your friendship. Thank you Laurène for letting me discover what a beautiful country Switzerland is, and more specially your Valais. The three of us have shared a lot together, whether it be an apartment for 3 years with Laurène or a deep passion for biathlon with Julien. The travels, discussions, “vins chaud”, sporting activities with you two are just a few examples of the memories I will keep with me and treasure. Thank you for all of this and all the rest. I am sure that our path will cross again soon.

There are people in life that are truly invaluable for a work like this. I am deeply grateful to my family, for surrounding me with love, and encouraging me in every adventure I undertook or went through. Their support was unconditional and was the most solid source of energy throughout my thesis. Finally, I would have never been able to achieve these four years of research without the affection, support and patience of my partner Thomas. Thank you so much for being by my side and believing in me, sometimes for the both of us. There are no words that could accurately express my gratitude and love for you.

Before closing this chapter, I would like to have a thought for Victor Muñoz, whose passing has profoundly changed me, my way of seeing things, and undeniably the course of my thesis.

Of course, I thank you, reader, who takes the time to read this thesis. I really hope you will enjoy it as much as I did!

Remerciement

Cette thèse est le fruit de quatre années de recherche et d'une véritable expérience de vie. Elle n'aurait pas été possible sans l'aide et le soutien de nombreuses personnes. Je saisis cette occasion pour leur exprimer ma sincère gratitude.

Tout d'abord, j'aimerais remercier Prof. Pierre Vandergheynst, président du jury, ainsi que les membres du comité, Prof. Lucia Romani, Prof. Dimitri Van De Ville et Prof. Mario Figueiredo, pour leur temps et leur appréciation favorable de mon travail. Je remercie également le Fonds National Suisse d'avoir financé mes recherches.

Je voudrais ensuite remercier deux personnes qui ont fait de cette thèse une réalité. Je suis extrêmement reconnaissante envers Christophe Lallement d'avoir cru en moi et de m'avoir encouragé à postuler pour un doctorat à l'EPFL. Je remercie aussi tout particulièrement mon directeur de thèse, Prof. Michael Unser, pour ses conseils tout au long de ces quatre années. Michael est un excellent superviseur et un enseignant dévoué. J'ai particulièrement apprécié son enthousiasme pour la recherche et le fait que sa porte soit toujours ouverte. Travailler avec lui a été une extraordinaire opportunité pour moi et cela m'a permis de grandir en tant que scientifique.

Ce fut un immense privilège de faire mon doctorat au sein du laboratoire d'imagerie biomédicale (LIB). C'est un lieu de travail extrêmement agréable, propice aux échanges et aux collaborations scientifiques. Par-dessus tout, le LIB est un lieu de partage et d'amitié. La salle de réunion a ainsi accueilli tour à tour des parties de Pictionary, des soirées cinéma, des apéros ainsi que des soirées raclette (et parfois des réunions). Sans oublier les sorties qui ont conduit toute l'équipe à partager de nombreuses expériences : ski, randonnée, rafting, via ferrata... Pour tous ces

moments de partage et de bonne humeur, je tiens à remercier chaleureusement tous les membres du LIB (trop nombreux pour les énumérer ici), passés et présents.

En plus de Michael, le LIB repose également sur les épaules de Philippe Thévenaz et Daniel Sage. Merci, Philippe, pour tes relectures scrupuleuses de mes papiers. Cela a considérablement amélioré la qualité de mon travail et m'a également permis d'améliorer ma communication scientifique écrite. Daniel Sage (alias Lucky Luke, alias Columbo) a rempli son rôle de co-directeur de thèse à la perfection. Il a toujours été de bon conseil et prêt à aider et à collaborer. En particulier lorsqu'il s'agissait d'encadrer des projets d'étudiant. Sa porte était toujours ouverte pour une pause-café ou pour travailler sur Caffè. Je me suis sentie écoutée à chaque fois que j'ai eu besoin de parler des hauts et des bas que je pouvais rencontrer dans mon travail ou dans ma vie personnelle. Gentil, bienveillant et enfantin sont les trois mots qui le décrivent le mieux. Merci pour tout Daniel. Tous ces fous rires que tu nous as procurés à Laurène et moi (ski avec un bâton, laser game, Vallée Blanche, parties de pétanque) resteront gravés dans ma mémoire. Je remercie également Manuelle, Beatrice, Nadia, Monika et Claudia pour leur précieuse aide dans toutes les tâches administratives, ainsi que dans l'organisation des sorties du laboratoire. Un grand merci à mes collègues de bureau, Thomas D. et Shayan, pour avoir créé une si bonne ambiance de travail ; notamment, mais pas uniquement, en riant à mes blagues plus ou moins drôles.

J'aimerais remercier les personnes avec lesquelles j'ai collaboré et qui ont beaucoup contribué à cette thèse. Au sein du LIB, j'ai eu la chance de travailler avec Virginie, Daniel Schmitter, Daniel Sage et Adrien. Ensemble nous avons apporté de précieuses pierres au grand édifice des *snakes*. Julien, quant à lui, m'a fait découvrir le monde périodique et merveilleux des processus gaussiens. J'ai également eu la chance de travailler avec Lucia Romani de l'Université de Bologne et Paola Novara. Toutes deux m'ont aidé à découvrir avec enthousiasme la mathématique des *subdivisions*. Je souhaite également remercier Raoul Schorer de l'Hôpital Universitaire de Genève pour avoir apporté un brin d'application médicale à ma thèse.

Tout au long de ma thèse, j'ai eu le plaisir de participer à plusieurs activités d'enseignement. J'ai notamment eu l'occasion de superviser des étudiants au cours de leur cursus. Les guider dans leurs propres recherches a été pour moi très enrichissant. Je voudrais particulièrement remercier Axel, Brune, Yann et Aymeric pour leur investissement qui a conduit à de précieuses publications. J'ai partic-

ulièrement apprécié endosser le rôle d’assistante pour le cours de Signaux & Systèmes donné à des étudiants de Bachelor. Nous nous sommes, avec la “dream team” des assistants, investis semaine après semaine pour que le cours soit une réussite. En particulier, je voudrais remercier profondément mes collègues du LIB (Virginie, Laurène, Julien, Thomas D., Toulouse) ainsi que les membres du LTS2 (Johan, Yann, Michael D., Lionel) pour avoir fait de l’assistanat un si bon moment, mais aussi pour tous les moments et souvenirs que nous avons partagés à Bruxelles, pendant nos séminaires ou à travers nos parties d’Esquissé.

Je tiens à remercier tout particulièrement Daniel Schmitter, Julien et Laurène, qui ont été extrêmement précieux ces quatre dernières années. A mon arrivée à Lausanne, Daniel Schmitter m’a prise sous son aile afin que je puisse me faire une place à la fois professionnellement et personnellement. Je te remercie sincèrement Daniel pour cela. Ma plus grande gratitude va à Laurène et Julien. Cette thèse aurait été tellement différente sans vous, votre soutien et votre amitié. Merci Laurène de m’avoir fait découvrir la beauté de la Suisse, et plus particulièrement ton canton du Valais. Nous avons partagé beaucoup de choses tous les trois, qu’il s’agisse d’un appartement pendant trois ans avec Laurène ou d’une profonde passion pour le biathlon avec Julien. Les voyages, les discussions, les vins chauds, les activités sportives avec vous deux ne sont que quelques exemples des souvenirs que je garde avec moi et que je chéris déjà. Merci pour cela et tout le reste. A tout bientôt chers amis !

Il y a des personnes dans la vie qui sont tout simplement inestimables dans un projet comme celui-ci. Je suis profondément reconnaissante envers ma famille qui m’a entourée d’amour et encouragée dans toutes mes aventures. Leur soutien inconditionnel aura été la source d’énergie la plus précieuse tout au long de ma thèse. Enfin, je n’aurais jamais pu aller au bout de ces quatre années de recherche sans l’affection, le soutien et la patience de mon compagnon Thomas. Merci infiniment d’avoir été à mes côtés et d’avoir cru en moi, parfois pour nous deux. Les mots me manquent pour exprimer avec précision ma gratitude et mon amour pour toi.

Avant de terminer ce chapitre, je voudrais avoir une pensée toute particulière pour Victor Muñoz. Son décès a profondément changé la personne que je suis, ma vision des choses et, indéniablement, le cours de ma thèse.

Bien entendu, je te remercie, lecteur. Toi qui prends le temps de lire cette thèse. J'espère que tu l'apprécieras autant que moi !

Contents

Abstract	i
Résumé	iii
Acknowledgement	vii
Remerciement	xi
Notation	xxi
1 Introduction	1
1.1 Context	1
1.1.1 Biomedical Image Analysis	2
1.1.2 Segmentation	2
1.1.3 Usable Bioimaging Software	3
1.2 A Survey on Active Contours/Surfaces	4
1.2.1 Curve/Surface Representation	5
1.2.2 Snake Energy	6
1.2.3 Challenges	8
1.3 Contributions	9
1.4 Roadmap of The Thesis	11
2 Review of Parametric Snakes	13
2.1 Geometric Representation	13
2.1.1 Parametric Closed Curves	13

2.1.2	Parametric Representation of Tensor-Product Surfaces	14
2.1.3	Desirable Properties of the Basis Functions	16
2.2	Image Energies	18
2.2.1	Contour-based energy	18
2.2.2	Region-based energy	19
2.3	Splines	22
2.3.1	Exponential B-Splines as Basis Functions	22
2.3.2	Exponential Splines and Their Periodic Counterpart	23
2.3.3	Periodic Exponential Splines as Optimal Interpolators for Closed Curves	24
3	Parametrization with Local Refinement	29
3.1	Notations	30
3.2	Scaling and Refinable Functions	30
3.3	Representation with Local Refinement	32
3.3.1	Closed Curves	32
3.3.2	Tensor-Product Surfaces	33
3.4	Appendices	37
3.4.1	Proof of Proposition 3.3.1	37
3.4.2	Proof of Proposition 3.3.2	38
4	Subdivision-Based Representation	39
4.1	Notations	40
4.2	Subdivision Schemes	41
4.2.1	Convergent Subdivision Schemes	42
4.2.2	Interpolating Subdivision Schemes	43
4.2.3	Affine Invariance	44
4.3	Design of Subdivision Schemes	44
4.3.1	Generation of Polynomials	44
4.3.2	Deslauriers-Dubuc Subdivision Scheme	45
4.3.3	Minimum-Support Subdivision Scheme	46
4.4	Design of Non-Stationary Subdivision Schemes	46
4.4.1	Generation of Exponential Polynomials	47
4.4.2	Generation of Trigonometric Functions	47
4.4.3	Non-Stationary Deslauriers-Dubuc Subdivision Scheme	47
4.4.4	Non-Stationary Minimum-Support Subdivision Scheme	49

4.5	Connection with Scaling Functions and Parametric Curves	49
4.6	Subdivision Surfaces	51
4.6.1	Notation and Terminology	51
4.6.2	Subdivision Schemes	51
4.6.3	Properties	52
4.6.4	Examples of Subdivision Schemes	53
4.7	Appendices	58
4.7.1	Derivation of Equation (4.2.3)	58
4.7.2	Derivation of Equation (4.2.7)	58
4.7.3	Proof of Proposition 4.2.2	59
5	Image Energies	61
5.1	Edge-based Energy	62
5.1.1	Derivation for the Locally Refinable Parametrization	62
5.1.2	Discretization for the Subdivision-based Representation	63
5.2	Region-based Energy using Intensity Distribution	65
5.2.1	Derivation for the Locally Refinable Parametrization	66
5.2.2	Discretization for the Subdivision-based Representation	67
5.3	Ridge-based Energy	69
5.4	Texture-based Energy	72
5.4.1	Texture Analysis with Filters	72
5.4.2	Energy Term	74
5.4.3	Fisher's Linear Discriminant Analysis	75
5.5	Conclusion	76
5.6	Appendices	77
5.6.1	Proof of Proposition 5.1.1	77
5.6.2	Proof of Proposition 5.1.2	77
5.6.3	Notation and Properties of Regular Meshes	78
5.6.4	Proof of Proposition 5.1.3	79
5.6.5	Proof of Proposition 5.2.1	81
5.6.6	Proof of Proposition 5.2.2	82
5.6.7	Proof of Proposition 5.2.3	83

6	Design of Active Contours	85
6.1	Locally Refinable Parametric Snakes	87
6.1.1	Framework	87
6.1.2	Implementation Details	88
6.1.3	Experiments and Validation	88
6.1.4	Conclusions	90
6.2	Texture-Driven Parametric Snakes	92
6.2.1	Framework	93
6.2.2	Implementation Details	94
6.2.3	Experiments and Validation	96
6.2.4	Conclusions	110
6.3	Multiresolution Subdivision Snakes	113
6.3.1	Framework	114
6.3.2	Optimization: a Multiresolution Approach	114
6.3.3	Experiments and Validation	115
6.3.4	Discussion	122
6.3.5	Conclusions	127
6.4	Active Tessellations	129
6.4.1	Framework	130
6.4.2	Implementation Details	133
6.4.3	Experiments and validation	133
6.4.4	Conclusions	136
7	Active Subdivision Surfaces	139
7.1	Framework	140
7.2	A Coarse-to-Fine Optimization Strategy	141
7.3	Experiments and Validation	143
7.3.1	Robustness to Noise	144
7.3.2	Robustness to the Initialization	144
7.3.3	Segmentation Accuracy	147
7.3.4	Segmentation on Real Biomedical Images	152
7.4	Conclusions	152
8	Conclusion	155
8.1	Contributions	155
8.2	Research Outlook	158

A	An Inner-Product Calculus for Periodic Functions and Curves	161
A.1	Introduction	162
A.1.1	Notations	162
A.1.2	Inner Products	162
A.2	Inner-Product Calculus	163
A.2.1	General Calculation	163
A.2.2	Specific Cases of a_{12} in Practice	165
A.3	Applications	166
A.3.1	Resampling of a Spline Curve	166
A.3.2	Area Enclosed by a Parametric Curve	167
A.4	Conclusion	169
A.5	Appendices	170
A.5.1	Proof of Proposition A.2.1	170
A.5.2	Proof of Proposition A.2.3	170
A.5.3	Proof of Proposition A.2.4	171
B	Periodic Splines and Gaussian Processes	173
B.1	Introduction	174
B.1.1	Inverse Problems in the Continuous Domain	174
B.1.2	Variational vs. Statistical Methods	175
B.1.3	Periodic and General Setting	176
B.1.4	Related Works	176
B.1.5	Outline and Main Contributions	178
B.2	Mathematical Background for Periodic Signals	178
B.2.1	Linear and Shift-Invariant Operators	179
B.2.2	Periodic L-Splines	180
B.3	Periodic Representer Theorem	181
B.3.1	Search Space	182
B.3.2	Periodic Reproducing-Kernel Hilbert Space	183
B.3.3	Periodic Representer Theorem	184
B.4	Periodic Processes and MMSE	185
B.4.1	Non-Periodic Setting	185
B.4.2	Gaussian Bridges	186
B.4.3	Measurement Model and MMSE Estimator	188
B.4.4	MMSE Estimation as a Representer Theorem	189
B.5	Quality of the Estimators on Simulations	190

B.5.1	Generation of Gaussian Bridges	191
B.5.2	Influence of λ	191
B.5.3	Influence of γ^2	194
B.6	Discussion	200
B.6.1	Comparison with Inverse Problems on the Real Line	200
B.6.2	Comparison with TV Regularization	201
B.7	Conclusion	202
B.8	Appendices	203
B.8.1	Proof of Proposition B.2.1	203
B.8.2	Proof of Proposition B.3.1	203
B.8.3	Proof of Proposition B.3.3	204
B.8.4	Proof of Theorem B.3.4	205
B.8.5	Proof of Proposition B.3.5	208
B.8.6	Proof of Proposition B.3.6	209
B.8.7	Proof of Proposition B.4.3	210
B.8.8	Proof of Theorem B.4.4	210
B.8.9	Proof of Proposition B.4.5	212
B.8.10	Proof of Proposition B.5.1	212
Bibliography		212
Curriculum Vitæ		241

Notation

OBJECT REPRESENTATION

\mathbf{p}	Sequence of 2D/3D points
\mathcal{M}	Triangular mesh
\mathbb{P}	Set of points describing \mathcal{M}
\mathbf{r}	Continuously defined curve
\mathcal{C}	Contour described by \mathbf{r}
$d\mathbf{r}$	Infinitesimal vector element of \mathcal{C}
Ω	Surface enclosed by \mathbf{r}
$ \Sigma $	Area of Ω
σ	Continuously defined surface
\mathcal{S}	Surface described by σ
$d\sigma$	Vector differential area
\mathcal{V}	Volume enclosed by σ
$ V $	Volume of \mathcal{V}
\mathbf{n}	Unit normal vector
φ	Basis function
$\dot{\varphi}$	Derivative of φ

SNAKE QUANTITIES

\mathbf{c}	Sequence of control points
Θ	Snake parameters
E_{snake}	Snake energy
ρ	Refinement filter
k	Subdivision step
f	Image data
∇f	Gradient vector of f
Δf	Laplacian of f

MISCELLANEOUS

$\langle \cdot, \cdot \rangle$	Scalar product
\otimes	Tensor product
$*$	Convolution
$\lfloor \cdot \rfloor$	Floor function
$\lceil \cdot \rceil$	Ceil function
$\delta[\cdot]$	Kronecker delta
δ	Dirac impulse
III	Dirac comb
j	Unit imaginary number

Chapter 1

Introduction

1.1 Context

Image analysis is usually understood as the extraction of meaningful information or measurements from images. It typically consists of three main steps: Feature extraction, segmentation to locate object boundaries, and meaningful description through statistical metrics and interpretation (*e.g.*, classification or recognition) (Figure 1.1).

In this thesis, the focus is on the development and use of novel segmentation algorithms for biomedical image analysis. Moreover, we give a special attention to convert these algorithms to friendly and usable software.

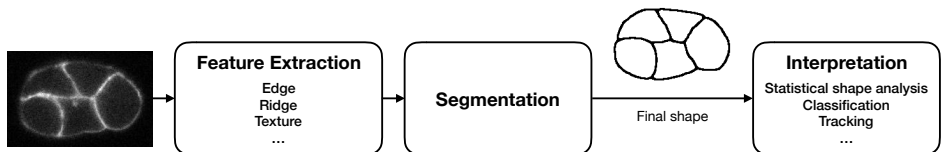


Figure 1.1: Image analysis workflow.

1.1.1 Biomedical Image Analysis

In the biomedical field, the characterization of the spatial and temporal organization of a structure is an ever growing need [1–4]. Various advances in imaging techniques have made this possible by improving spatial resolution and speed of acquisition, which has led to improved temporal resolution as well. Such progresses do not come without additional difficulties though: the size, number and complexity of recorded data are constantly increasing. Typically, to study an *in vivo* development of cells (time-lapse), one needs to process and analyze hundreds of images containing hundreds of cells each.

The human analysis and processing of such large amounts of multidimensional data is time consuming. Moreover, there is important variability among experimenters, as well as within the realizations of a single experimenter. This has promoted the development of computerized image analysis tools that 1) considerably decrease processing time; 2) bring robustness and reproducibility to an extent that is not achievable by humans.

Coined *bioimage informatics* [5,6], this emerging field has worked towards developing new algorithms for the image analysis workflow, and making them accessible to the whole community. Many open source bioimage analysis tools have thus been developed [7,8], such as ImageJ [9], Fiji [10], Icy [11], CellProfiler [12,13] and Ilastik [14] for the biologists, and 3DSlicer [15], MIA [16] and GIMIAS [17] for the medical world.

Recently, the emergence of deep learning techniques [18] (a.k.a. neural networks) has drastically transformed the field of bioimage informatics and its quest for more accurate and faster diagnostics. Deep learning-based approaches have for example been used in medical imaging to detect skin [19] and brain [20] cancers. Successful use of these learning-based techniques in various biomedical imaging problems are found on a regular basis [21,22] and the trend is likely to continue.

1.1.2 Segmentation

Image segmentation is a key step of many image analysis pipelines (Figure 1.1). The goal of segmentation is to partition an image into different regions, such that pixels in each region have similar properties. In other words, segmentation separates a desired object of interest from its background. This usually consists in finding the boundaries or the homogeneous region of the target. In biomedical imaging,

the objects of interest are typically cells or anatomical structures (*e.g.*, vessels and organs).

The conceptual interest of segmentation is that it aims for a high-level description of an object in an image (*e.g.*, a shape) rather than focusing on individual pixels. Segmentation is thus tasked with changing the representation of an image into something that is more meaningful to analyze. The best apparatus to detect and extract high level information from an image is the visual cortex [23]. For this reason, many segmentation methods such as edge or ridge detectors and neural networks are inspired by human visual perception models. If properly executed, segmentation makes the subsequent extraction of information much easier. The whole difficulty lies in the fact that the proper execution of segmentation is a particularly challenging task. Segmentation is highly application dependent and it is an ill-defined problem.

A large variety of segmentation algorithms exists [24]. Unavoidably, no existing segmentation method can be considered generic enough for all applications. They differ depending on the imaging modality (*e.g.*, x-ray tomography, fluorescence microscopy), the application domain, and the level of automation. Among classical methods, we can cite intensity thresholding [25–27], edge detection, and watershed approaches [28–32]. These methods perform well in good imaging conditions, but they are known to be sensitive to noise, which often results in over-segmentation, and poor image contrast. Popular alternative segmentation methods are deformable models, also known as active contours/surfaces. Active contours are more flexible compared to other approaches as they combine efficient and well-controlled image segmentation with extensive and eased user interaction. Deformable models allow for manual edition of the segmentation outcome, which is a significant advantage as automatic segmentation methods rarely reach 100% accuracy. Active contours/surfaces are further described in Section 1.2 and constitute the main topic of this thesis.

1.1.3 Usable Bioimaging Software

Eventually, the reason bioimage analysis algorithms are developed is to help biologists and physicians analyze their data. As those practitioners may have only basic programming training and image processing knowledge, it is of fundamental importance to convert the developed algorithms to friendly and usable software if one aims to have a meaningful impact in the community.

To ensure the success of a bioimage segmentation software, some requirements have been established [33,34]. Among them we focus our attention on three specific ones.

- *User-friendly*: Bioimaging software has to be usable by non-programming experts. The interface should be intuitive and easy-to-use, and documentations such as clear instructions or video tutorials have to be provided.
- *Open-source*: The openness of a software provides the necessary transparency. It is fundamental for a biologist or a scientist to be able to understand how the algorithms work and to be able to adapt it for research purpose if needed. Another advantage of the openness is that it is publicly accessible.
- *Reproducibility*: One should be able to replicate the experiments carried out by the developer. Hence, data and parameters used for the software validation should always be provided.

1.2 A Survey on Active Contours/Surfaces

Active contours/surfaces, also called “snakes”, are among the most popular tools for image segmentation. They were first proposed in 2D by Kass *et al* [35] in 1988 and generalized to the 3D case by Terzopoulos *et al.* [36]. They have become popular models to segment structures in biomedical images as they provide an excellent trade-off between flexibility and efficiency [37–43]. They consist in a deformable curve/surface that is deformed from an initial-user provided position towards the boundary of an object of interest in a 2D/3D image. The deformation of the snake contour is driven by the minimization of a suitable objective function, often called *energy* in this context [44]. They allow for user interactions, either to specify the initial position or for manual corrections if needed. Two components thus play an important role in the construction of an active contour: The *geometric representation* of the snake, which describes the nature of its contour and determines some geometric properties (*e.g.*, smoothness, shape reproduction); and a suitable *energy* functional that drives the fitting of the curve/surface to the image data. The choice of this energy term is crucial because it determines the quality of the segmentation outcome [44, 45]. The choice and design of both the geometric model and energy of the snake depend on the application and imaging modality.

Many snake models have been proposed [37, 46]. Currently, active contours/surfaces are described either implicitly (*e.g.*, level sets [47, 48]), or explicitly, with point/mesh-based [43] and parametric snakes [49–52]. For the energy term, the most common approaches are based on edge and intensity information aggregated from either inside or on the curve [44].

1.2.1 Curve/Surface Representation

Snakes are usually categorized based on their curve/surface representation.

Geodesic snakes (or Level-set methods) are based on the idea developed by Osher and Sethian to model propagating solid/liquid interfaces with curvature-dependent speeds [53]. In geodesic approaches [47, 54–56], the curve/surface of the snake has an implicit representation described as the zero level-set of a higher-dimensional manifold (Figure 1.2 (a)). Formally, the continuous curve/surface is given by $\Phi^{-1}(0) = \{\mathbf{p} \in \mathbb{R}^n \mid \Phi(\mathbf{p}) = 0\}$, where $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is a scalar function defined all over the image domain. These snakes can be extended to any number of dimension and they are particularly flexible in terms of topology. Indeed, under a suitable energy functional, they have the ability to automatically handle topology changes. However, they tend to be computationally expensive since they evolve a manifold with a higher number of dimensions than the actual contour/surface to segment. They have many degrees of freedom, which can lead to overfitting in practice. In summary, geodesic snakes based on level-sets are well suited to segment shapes with high variability, otherwise they are suboptimal.

Point-snakes/active meshes are historically the first snakes [35, 49]. They have a simple discrete representation where the shape is described by a set of ordered points (2D or 3D) (Figure 1.2 (b)). Point-snakes can handle topology changes and their discrete nature allows for an easy implementation. Moreover, in 3D they are compatible with open source libraries for optimization or visualization. However, they have two main drawbacks. First, the discrete nature of the representation does not ensure smoothness of the curve/surface, which requires an internal regularization term. Second, a lot of parameters are required to encode shapes, even the simple ones. In fact, there are two (or three in 3D) degrees of freedom for each snake point. This large number

of parameters makes the segmentation algorithm none robust, and results in a high computational complexity.

Parametric snakes have a continuous spatial representation via the use of basis functions. They are encoded by a set of *control points* and a continuous parameter [50, 57] (Figure 1.2 (c)). Contrary to point-snakes, they are built to ensure continuity and smoothness. They require fewer parameters, which leads to a faster optimization and better robustness. One can also encode more complex shapes by increasing the number of control points. Since the curve/surface of parametric snakes is represented explicitly, it is easy to introduce prior knowledge such as shape constraints [58, 59]. Moreover, they allow the user to suitably modify results in a user-friendly way by moving some control points. However, well-known drawbacks of parametric approaches are the restricted nature of the shape that they can generate, and their inability to deal with topology changes such as contour/surface merging and splitting, although solutions have been proposed for specific cases [60]. Parametric snakes are at the heart of this thesis and are further described in Chapter 2.

1.2.2 Snake Energy

The energy functional of the snake drives the evolution of the curve/surface to fit object boundaries. Kass *et al.* [34] originally formulated the snake energy as a linear combination of three terms:

- the *image energy* E_{image} , which is purely data driven and is responsible for guiding the curve/surface of the snake towards the boundary of interest;
- the *internal energy* E_{int} , which ensures smooth boundaries of the segmented object. In the formulation of parametric snakes, the smoothness of the representation is often ensured by the choice of the basis functions, thus eliminating the need for an explicit internal energy term;
- the *constraint energy* E_c , which gives rise to external constraints to put the snake near the desired local minimum. This can be done through an interface to let the user interact with the snake.

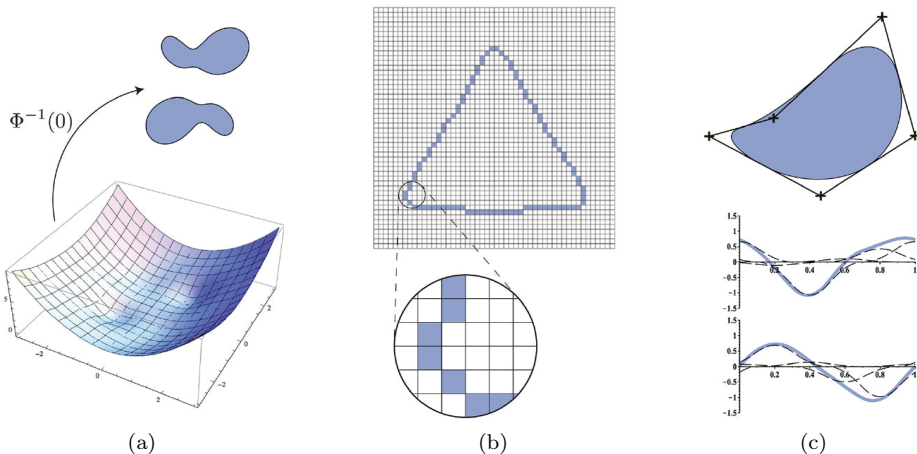


Figure 1.2: The three main curve representations of active contours. (a) A geodesic snake, defined as a continuous curve corresponding to the zero level-set $\Phi^{-1}(0)$ of a scalar function Φ ; (b) a point-snake over the grid associated to a discrete image model. The discrete curve is displayed as shaded pixels and here satisfies an 8-neighbor connectivity; (c) a parametric snake, where the coordinate functions are depicted in solid lines, and the dashed lines indicate the weighted basis functions. The symbols “+” are the control points. Source: These illustrations are taken from [37].

The total energy of the snake is expressed as

$$E_{\text{snake}}(\Theta) = E_{\text{image}}(\Theta) + E_{\text{int}}(\Theta) + E_c(\Theta), \quad (1.2.1)$$

where Θ stands for the curve/surface representation (manifolds, snake points or control points). The deformation of the contour consists in an optimization procedure in which the snake is iteratively updated from an initial position until the minimum of $E_{\text{snake}}(\Theta)$ is obtained. The optimal Θ_{opt} ¹ is thus obtained as

$$\Theta_{\text{opt}} = \arg \min_{\Theta} E_{\text{snake}}(\Theta). \quad (1.2.2)$$

¹Note that Θ_{opt} is not necessarily unique.

Many methods exist to minimize the energy functional (*e.g.*, gradient descent, partial differential equations approaches, dynamic programming, Powell-like line-search method [61]), and each optimization scheme is usually linked to a particular snake representation.

The image energy is the most important of the three terms in (1.2.1) since it incorporates image information (features) to guide the snake towards the boundary of the object of interest. Different image energy terms are used in practice. The most commonly approaches can be categorized into two broadly defined categories:

- *contour-based methods* E_{contour} , which use local image information and are purely based on edge or ridge maps obtained by computing the gradient or Hessian of an image [35, 44, 50, 62]. They provide a good localization of the contour of the object to segment. However, they have a narrow basin of attraction making a good initialization critical;
- *region-based methods* E_{region} , which use statistical information (*e.g.*, intensity distribution, texture) to distinguish different homogeneous regions [44, 48, 57, 63, 64]. The region-based energies have a larger basin of attraction and can converge even if explicit edges are not present. However, they do not provide a good localization of the boundaries of the object to segment.

In order to benefit from the advantages of both methods, a unified image energy was proposed in [44] and in [64]. Typically, E_{image} can be expressed by the following combination

$$E_{\text{image}}(\Theta) = bE_{\text{contour}}(\Theta) + (1 - b)E_{\text{region}}(\Theta), \quad (1.2.3)$$

where $b \in [0, 1]$ is a trade-off parameter that balances the contribution of the two energies. It is generally agreed that the choice of the features, *i.e.*, the visual attributes that differentiate the target from its surroundings, to detect through E_{image} depends on the modality and application. In Figure 1.3, we highlight some features to detect according to the structure to segment.

1.2.3 Challenges

In the context of bioimage segmentation, there are ongoing challenges for snake-model algorithms [37].

- *Robustness*: The snake has to robustly perform in real-life imaging conditions (*e.g.*, images with heavy noise and low contrast);

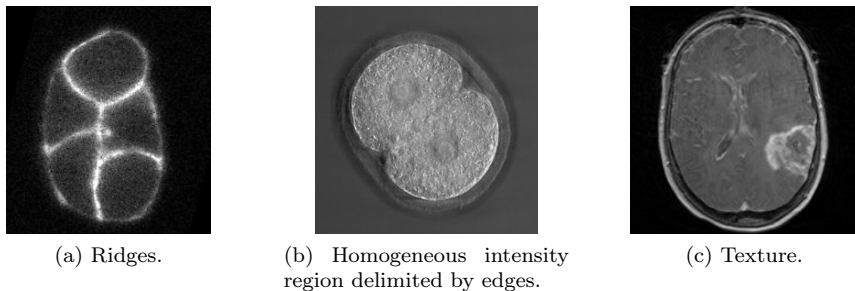


Figure 1.3: Features in biomedical images. (a) The cell membranes of a *C. elegans* embryo are separated by ridges; (b) two dividing cells that form an homogeneous intensity region delimited by edges; (c) a tumor whose intensity distribution is not homogeneous. Here, the best feature to distinguish the tumor from the background is its texture. Sources: (a) R. Jankele and P. Gönczy, EPFL; (b) <http://www.cellimagelibrary.org/images/35450/>; (c) image taken from [65].

- *Flexibility and prior knowledge*: The snake has to be versatile enough to accommodate a wide range of shapes. At the same time, one should have the possibility to integrate prior knowledge (*e.g.*, shape constraints) into the segmentation procedure;
- *Computational efficiency*: It is crucial to provide reasonably fast implementations that run on standard computers, especially when dealing with 3D data.

1.3 Contributions

This thesis is dedicated to the study and development of segmentation algorithms for biomedical structures in 2D and 3D images. We identify limitations of parametric snakes and address them providing new representation models as well as novel energies. We then merge these theories to construct new active contours/surfaces for various applications.

Throughout this thesis, we focus on closed (*i.e.*, periodic) curves/surfaces. Those

models indeed have a wide range of applications in bioimaging [66,67]. In addition, we are interested in models that accurately represent shape with few control points as possible, since those have better robustness and lower computational cost (see Section 1.2.3).

For the optimization process, we adopt a *subdivide and conquer* strategy. First, we look at several smoothed versions of the original image and, for each, adapt the resolution of the snake curve/surface to the level of detail of the target (*i.e.*, subdivide). Then, we segment the object of interest at each resolution recursively, from the coarsest to the finest image (*i.e.*, conquer). This robust strategy exploits the multiresolution property of our new representation models, as well as characteristics of smoothed images (few details and low noise).

Finally, we give special attention to the conversion of our algorithms into usable software, as described in Section 1.1.3.

Hereafter we provide a short summary of our five main scientific contributions and their related publications.

1. **Locally refinable parametric snakes** [68,69]: We introduce the possibility to locally increase the number of control points of 2D/3D parametric snakes by inserting basis functions at specific locations. Our approach relies on scaling and refinable functions that are related to wavelets.
2. **Subdivision snakes** [70–73]: Subdivision snakes are our most significant contribution. We introduce subdivision schemes, traditionally used in computer graphics for modeling, into a new framework for 2D/3D multiresolution snakes. We also derive the energy terms associated to this subdivision representation. Subdivision snakes have the ability to adapt the resolution of their curve/surface to the level of detail of their target. It allows us to also propose and adopt a coarse-to fine optimization strategy. Those snakes are more robust to noise and initialization than parametric snakes, and their geometric representation is also easier to extend to higher dimensions. Moreover, subdivision snakes can handle topology changes such as curve/surface merging and splitting.
3. **Ridge and texture-based energies** [73,74]: We propose two new energies, which allow for segmentation methods that are valid for a wider range of applications. The first energy attracts the snake towards ridges in the image and takes into account the direction of the normal to the curve/surface. The

second energy incorporates texture information and can be used with any filter-based texture features.

4. **Bioimaging software:** We implement our algorithms as 2D and 3D software that respect user-friendliness, open access and reproducibility criteria.
5. **Mathematical tools for periodic signal processing** [75, 76]: We introduce a calculus of the inner-product between two compactly supported and periodized basis functions. This tool is often needed in signal processing, especially in the construction of snakes. In addition, we present two approaches, variational and statistical, for the reconstruction of periodic continuous-domain signals from their corrupted discrete measurements.

In Figure 1.4, we give an outlook of these contributions, separated in modules, and their interconnections.

1.4 Roadmap of The Thesis

The thesis is organized as follows.

In Chapter 2, we review parametric snakes, which are fundamental theoretical tools for this thesis. We also recall the mathematical concepts that are extensively used throughout our work.

We continue by introducing two new representation models. We first propose the generic formulations of locally refinable parametric curves and surfaces in Chapter 3. Then, we present in Chapter 4 subdivision schemes that generate curve and surfaces satisfying the property of multiresolution.

In Chapter 5, we adapt standard energies initially defined for parametric snakes to our new representation models. In addition, we propose two novel energies that detect ridges and incorporate texture information.

The theories presented until now are merged in Chapters 6 and 7 to design new active contours and surfaces, respectively.

Finally, conclusions are drawn in Chapter 8.

In Appendices A and B, we introduce two signal processing theories to efficiently process periodic functions.

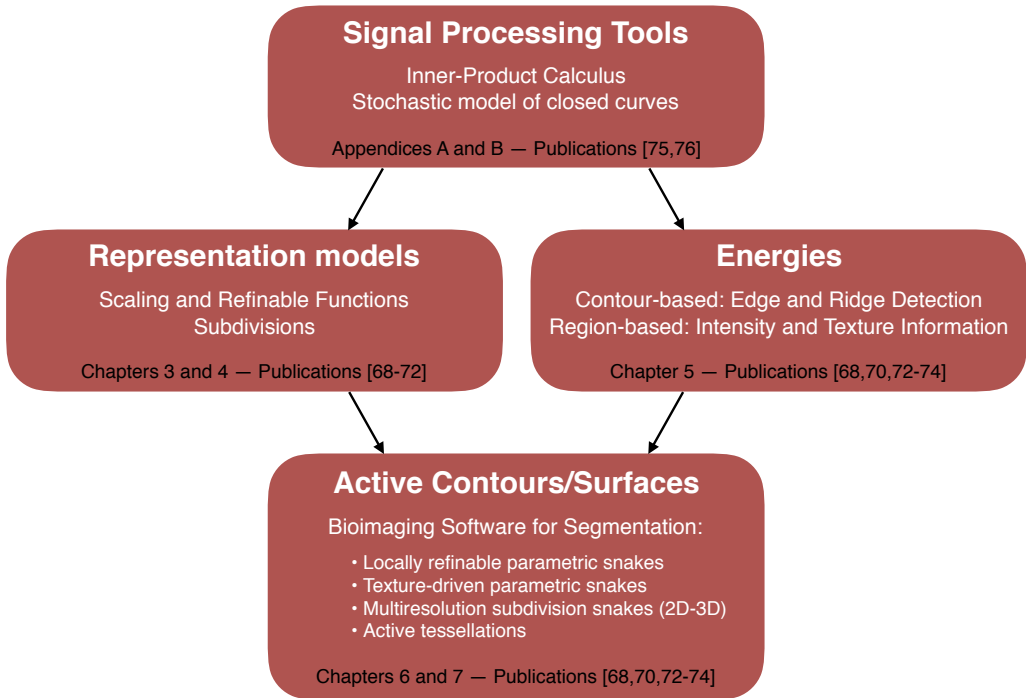


Figure 1.4: Content and roadmap of the thesis.

Chapter 2

Review of Parametric Snakes

In this chapter, we review parametric snakes that are the foundation for our research. We also introduce notations and notions that are relevant for this thesis. Finally, we briefly revisit exponential B-splines that constitute basis functions and introduce periodic exponential splines that naturally appear in the reconstruction of closed curves.

2.1 Geometric Representation

2.1.1 Parametric Closed Curves

A 2D planar curve $\mathbf{r} : \mathbb{R} \rightarrow \mathbb{R}^2$ is described by a pair of one dimensional coordinate functions $(r_1(t), r_2(t))$, where $t \in \mathbb{R}$ is a continuous parameter. Each of these functions is parametrized by a suitable linear combination of shifted basis functions $\{\varphi(\cdot - m)\}_{m \in \mathbb{Z}}$, where $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, specified by a sequence of control points $\{\mathbf{c}[m] = (c_1[m], c_2[m])\}_{m \in \mathbb{Z}}$, such that

$$\mathbf{r}(t) = \begin{pmatrix} r_1(t) \\ r_2(t) \end{pmatrix} = \sum_{m \in \mathbb{Z}} \mathbf{c}[m] \varphi(t - m), \quad t \in \mathbb{R}. \quad (2.1.1)$$

We are interested in closed curves in order to be able to segment blob-like or elliptical structures as it is often the case in bioimages. In this case, the two

coordinate functions r_1 and r_2 are periodic with the same period. The parametric snake is thus characterized by an M -periodic sequence of control points $\{\mathbf{c}[m]\}_{m \in \mathbb{Z}}$ with $\mathbf{c}[m] = \mathbf{c}[m + M]$. We re-express (2.1.1) as the finite summation

$$\begin{aligned} \mathbf{r}(t) &= \sum_{m=0}^{M-1} \sum_{n \in \mathbb{Z}} \mathbf{c}[m + Mn] \varphi(t - m - Mn) \\ &= \sum_{m=0}^{M-1} \mathbf{c}[m] \sum_{n \in \mathbb{Z}} \varphi(t - m - Mn) \\ &= \sum_{m=0}^{M-1} \mathbf{c}[m] \varphi_M(t - m), \end{aligned} \quad (2.1.2)$$

where $t \in [0, M[$ and φ_M is the M -periodization of the basis function φ defined by

$$\varphi_M(t) = \sum_{n \in \mathbb{Z}} \varphi(t - Mn). \quad (2.1.3)$$

Without loss of generality, the period can also be normalized to one so that $\mathbf{r}(t) = \mathbf{r}(t + 1)$ for all $t \in \mathbb{R}$ and hence, we only consider $t \in [0, 1[$, such that

$$\mathbf{r}(t) = \sum_{m=0}^{M-1} \mathbf{c}[m] \varphi_M(Mt - m). \quad (2.1.4)$$

The number M of control points determines the degree of freedom of the model. A small M leads to smooth and constrained shapes, while increasing M brings additional flexibility to approximate intricate shapes. We show in Figure 2.1 a parametric curve and its coordinate functions where the period was normalized to unity.

2.1.2 Parametric Representation of Tensor-Product Surfaces

A 3D surface $\sigma : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is described by a triplet of coordinate functions $(\sigma_1(u, v), \sigma_2(u, v), \sigma_3(u, v))$, where $u, v \in \mathbb{R}$ are continuous parameters. Each coordinate function is parametrized by a suitable linear combination of integer-shifted separable basis functions $\{\varphi_1(u - m)\varphi_2(v - n)\}_{m, n \in \mathbb{Z}}$ weighted by a sequence of control points $\{\mathbf{c}[m, n]\}_{m, n \in \mathbb{Z}}$. The functions φ_1 and φ_2 determine the shapes that

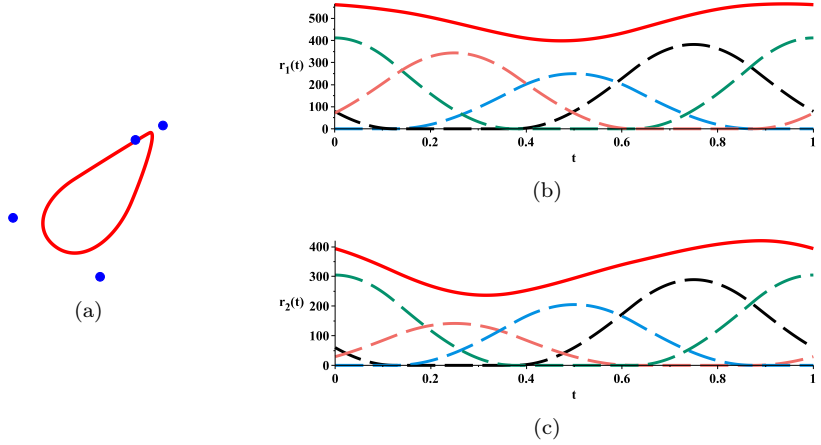


Figure 2.1: A parametric curve (a) and its coordinate functions (b) and (c). We used the exponential B-spline presented in [64, (8)] as basis function φ and $M = 4$. The blue dots are the control points and the dashed lines are the basis functions.

the parametric surface can adopt. Then, the parametric representation of the surface is given by the equation

$$\boldsymbol{\sigma}(u, v) = \begin{pmatrix} \sigma_1(u, v) \\ \sigma_2(u, v) \\ \sigma_3(u, v) \end{pmatrix} = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \mathbf{c}[m, n] \varphi_1(u - m) \varphi_2(v - n), \quad (2.1.5)$$

where $\{\mathbf{c}[m, n] = (c_1[m, n], c_2[m, n], c_3[m, n])\}_{m, n \in \mathbb{Z}}$ are the 3D control points describing the shape. To be a closed surface, $\boldsymbol{\sigma}(u, v_0)$ must be periodic in u for all v_0 . To satisfy this condition, it is necessary to apply periodic boundary conditions along the first index of the sequence of control points. Therefore, the sequence of control points becomes M_1 -periodic and satisfies $\mathbf{c}[m, n] = \mathbf{c}[m + M_1, n]$. In (2.1.5), we normalized this period to unity and the new expression is given by

$$\boldsymbol{\sigma}(u, v) = \sum_{m=0}^{M_1-1} \sum_{n \in \mathbb{Z}} \mathbf{c}[m, n] \varphi_{1, M_1}(M_1 u - m) \varphi_2(M_2 v - n), \quad (2.1.6)$$

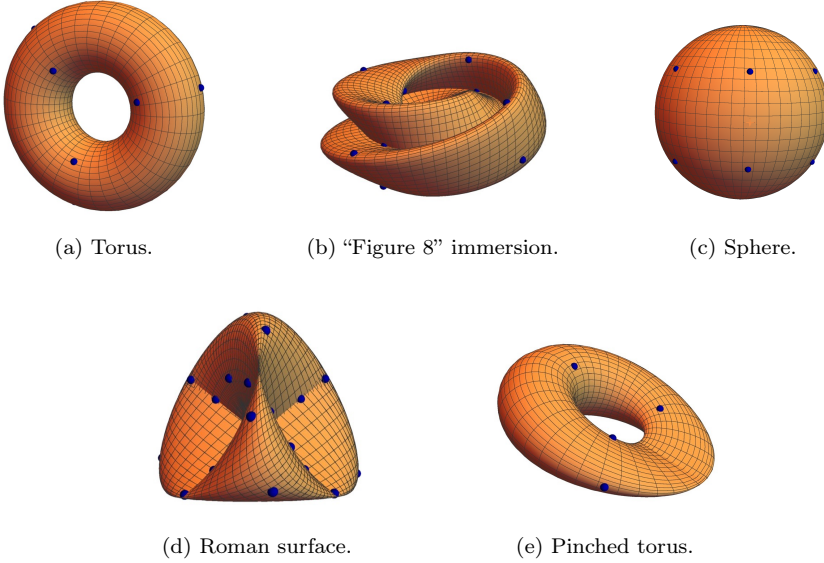


Figure 2.2: Parametric surfaces constructed with the family of interpolatory basis functions proposed in our paper [77]. Blue dots: control points.

where φ_{1,M_1} is the M_1 -periodization of φ_1 given by (2.1.3). The basis functions φ_1 and φ_2 are usually chosen to be compactly supported. The infinite sums in (2.1.5) and (2.1.6) can thus be reduced to a finite one, where the limits depend on the size of the support of the basis functions. In Figure 2.2, we illustrate some parametric shapes designed from (2.1.5) and (2.1.6).

2.1.3 Desirable Properties of the Basis Functions

The basis functions in (2.1.4) and (2.1.6) are responsible for the smoothness of the curve/surface as well as the shape that the snake can reproduce. Moreover, important considerations have to be taken into account to properly select the generator φ . Hereafter, we describe in detail these requirements for the 2D case. The extension to surfaces is straightforward by taking their bivariate analogous.

Riesz basis: Uniqueness and stability of the representation are guaranteed by the so-called Riesz-basis condition for the basis function φ [78]: There must exist two positive constants $0 < A, B < \infty$ such that,

$$A\|\mathbf{c}\|_{\ell_2(\mathbb{Z})}^2 \leq \left\| \sum_{m \in \mathbb{Z}} \mathbf{c}[m] \varphi(\cdot - m) \right\|_{L_2(\mathbb{R})}^2 \leq B\|\mathbf{c}\|_{\ell_2(\mathbb{Z})}^2. \quad (2.1.7)$$

Approximation power: A fundamental requirement is that the closed curve model given by (2.1.4) should have the capability of approximating any closed curve as closely as desired as the number M of control points tends to infinity. A necessary (and sufficient) condition [78] is that φ should be able to reproduce constants, which we formalize by

$$\forall t \in \mathbb{R}, \sum_{m \in \mathbb{Z}} \varphi(t - m) = 1. \quad (2.1.8)$$

In the literature, this constraint is often named the partition-of-unity condition [78].

Affine invariance: We want to represent shapes independently from their location and orientation. The parametric form of the model must be preserved at least through scaling and rigid-body transformations. This is guaranteed if the model (2.1.4) is affine invariant, which means that

$$\mathbf{A} \mathbf{r}(t) + \mathbf{b} = \sum_{m=0}^{M-1} (\mathbf{A} \mathbf{c}[m] + \mathbf{b}) \varphi_M(Mt - m), \quad (2.1.9)$$

where $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ and $\mathbf{b} \in \mathbb{R}^2$. It is easy to show that the constraint (2.1.9) is ensured if and only if the partition-of-unity condition (2.1.8) is satisfied.

Compact and small support: For practical and computational efficiency reasons, φ is often chosen to be of compact support [52, 78]. In fact, in this case the change of position of a control point modifies the shape only locally. This allows for a local control by the user.

A broad family of basis functions that fulfills the above properties are the exponential B-splines. Moreover, they have relevant reproduction properties for the

segmentation of biomedical structures. For these two reasons, exponential B-splines are often used to represent parametric snakes. We briefly describe these functions in Section 2.3.1.

2.2 Image Energies

The evolution of the curve/surface described in Section 2.1 is driven by the optimization of a cost functional referred to as snake energy. Using the notations introduced in Section 1.2.2, the optimization process for parametric snakes is performed by iteratively updating the collection of control points $\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}}$ from a starting position.

We here provide examples of standard image energies (see Section 1.2.2) for parametric snakes that are relevant to our work. The most common approaches are based on edge and intensity information aggregated from either inside or on the curve/surface. Their expressions are set with a minus sign since the snake optimization is defined as a minimization process (see (1.2.2)). In the following, we denote by \mathcal{C} the contour parametrized by \mathbf{r} and we recall that the relation between \mathbf{r} and the control points is given by (2.1.2).

2.2.1 Contour-based energy

One traditional approach to detect edges is to use the magnitude of the gradient [50, 62]. The energy functional is then given by

$$E_{\text{edge}}(\Theta) = - \oint_{\mathcal{C}} |\nabla f(\mathbf{r})| dr, \quad (2.2.1)$$

where dr is an infinitesimal vector element of \mathcal{C} , $\nabla f(\mathbf{r})$ is the gradient vector of the input image f and \mathbf{r} is the snake curve. A drawback of this approach is that it does not take into account the direction of the gradient. An improvement was proposed in [44], using the fact that, at the boundary of the object of interest, the image gradient should be perpendicular to the contour. In this case, the energy is expressed by

$$E_{\text{edge}}(\Theta) = - \oint_{\mathcal{C}} \langle \nabla f(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle dr, \quad (2.2.2)$$

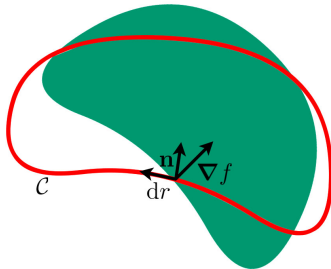


Figure 2.3: Schematic representation of a parametric snake (red contour) in interaction with a target (green region).

where $\langle \cdot, \cdot \rangle$ is the scalar product, $\mathbf{n}(\mathbf{r})$ is the unit vector normal to the curve \mathbf{r} , and $\nabla f(\mathbf{r})$ is the within-plane gradient of the image f at location (r_1, r_2) . In Figure 2.3, we present the configuration of the various quantities involved.

The 3D extension of (2.2.2) is proposed in [52] and is expressed by

$$E_{\text{grad}}(\Theta) = - \iint_{\mathcal{S}} \langle \nabla f(\boldsymbol{\sigma}), d\boldsymbol{\sigma} \rangle, \quad (2.2.3)$$

where \mathcal{S} is the surface parametrized by $\boldsymbol{\sigma}$ (see (2.1.6)) and $d\boldsymbol{\sigma}$ represents the vector differential area.

These approaches give a good localization of the edges. However, they are sensitive to noise and have a poor basin of attraction (Figure 2.5), which implies that the snake has to be initialized near to the object to segment [37].

2.2.2 Region-based energy

Region-based energies discriminate a homogeneous region from its background [64, 79, 80]. They are usually based on intensity information. The idea is to build a curve \mathbf{r}_λ around the snake \mathbf{r} , obtained by dilating it by a factor $\sqrt{2}$ with respect to its center of gravity. The surfaces enclosed by \mathbf{r} and \mathbf{r}_λ , denoted by Ω and Ω_λ , respectively, are such that $\Omega \subset \Omega_\lambda$ and the surfaces Ω and $\Omega_\lambda \setminus \Omega$ have the same area. We illustrate these quantities in Figure 2.4. The corresponding energy

functional is given by

$$E_{\text{intensity}}(\Theta) = -\frac{1}{|\Sigma|} \left| \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_{\lambda} \setminus \Omega} f(\mathbf{x}) dx_1 dx_2 \right|, \quad (2.2.4)$$

where $dx_1 dx_2$ is an infinitesimal vector element of Ω , f is the input image, and $|\Sigma|$ is the area of Ω with $\Sigma := \Sigma(\Theta)$ defined by

$$\Sigma(\Theta) = \iint_{\Omega} dx_1 dx_2. \quad (2.2.5)$$

The term in the absolute value in (2.2.4) can be positive or negative depending on whether we segment a bright object over a dark background or inversely. The minus sign and the absolute value thus ensure that $E_{\text{intensity}}$ is always minimized. Equations (2.2.4) and (2.2.5) require the computation of surface integrals, which are computationally expensive. To decrease the computational cost, an approach is to apply Green's theorem to convert them into line integrals [44]. This can only be achieved if the curve/surface is defined continuously and it does not self-intersect. For instance, the expanded formula of Σ is obtained by Green's theorem and (2.1.4) as [44, 64]

$$\begin{aligned} \Sigma(\Theta) &= -\oint_{\mathcal{C}} r_2 dr_1 \\ &= -\sum_{m=0}^{M-1} \sum_{n=0}^{M-1} c_1[m] c_2[n] \int_0^M \dot{\varphi}_M(t-m) \varphi_M(t-n) dt. \end{aligned} \quad (2.2.6)$$

Note that the area obtained by (2.2.6) is signed due to the clockwise or anti-clockwise orientation of the curve. We therefore take the absolute value of Σ in (2.2.4).

The 3D extension of (2.2.4) is proposed in [52] and adopt a strategy similar to the one we followed in the 2D case. We build a surface σ_{λ} around the snake σ , obtained by dilating it by a factor $\sqrt[3]{2}$ with respect to its center of gravity. The volumes enclosed by σ and σ_{λ} , denoted by \mathcal{V} and \mathcal{V}_{λ} , respectively, are such that $\mathcal{V} \subset \mathcal{V}_{\lambda}$ and \mathcal{V} and $\mathcal{V}_{\lambda} \setminus \mathcal{V}$ have the same volume. The corresponding energy functional is expressed by

$$E_{\text{intensity}}(\Theta) = -\frac{1}{|\mathcal{V}|} \left| \iiint_{\mathcal{V}} f(\mathbf{x}) dx_1 dx_2 dx_3 - \iiint_{\mathcal{V}_{\lambda} \setminus \mathcal{V}} f(\mathbf{x}) dx_1 dx_2 dx_3 \right|, \quad (2.2.7)$$

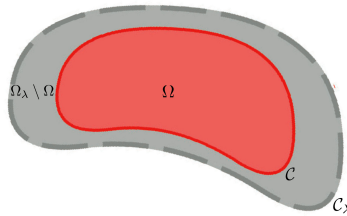


Figure 2.4: Illustration of the contours \mathcal{C} and \mathcal{C}_λ , and the shell $\Omega_\lambda \setminus \Omega$.

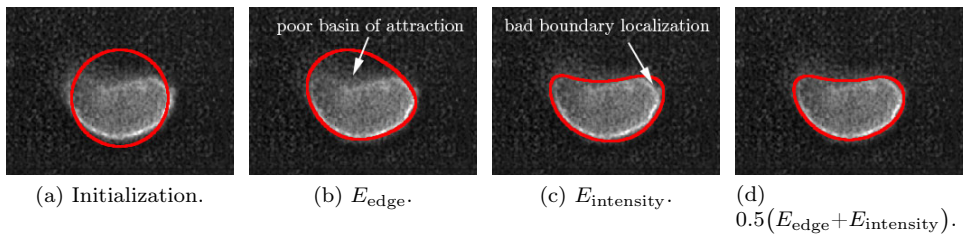


Figure 2.5: Illustration of the edge- and region-based energies. In order to benefit from the advantages of both energies, we use the combination (1.2.3) for $b = 0.5$.

where $dx_1 dx_2 dx_3$ is an infinitesimal vector element of \mathcal{V} , V is the volume of \mathcal{V} , and the relation between σ and its control points is given by (2.1.6).

Region-based energies are robust to noise and have a larger basin of attraction than contour-based energies [37] (Figure 2.5). However, the functionals (2.2.4) and (2.2.7) are well suited only when the intensity distributions of the object and its background have different variances and means [44, 74].

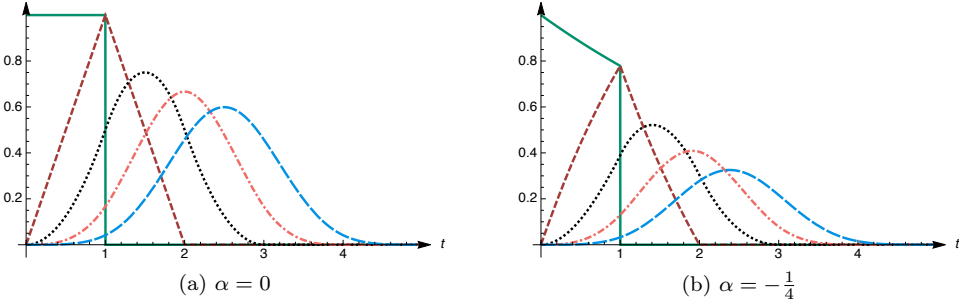


Figure 2.6: L -th order exponential B-splines $\beta_{(\alpha, \dots, \alpha)}$ for $L \in \{1, \dots, 5\}$ (from left to right).

2.3 Splines

2.3.1 Exponential B-Splines as Basis Functions

Exponential B-splines are popular not only in sampling and approximation theory but also to represent parametric curves and surfaces. They are the exponential counterpart of the well-known polynomial B-splines [81–83]. They are compactly supported [81] and have relevant reproduction properties for the segmentation of biomedical structures. An exponential B-spline of order L is fully characterized by its unordered list of (complex-valued) poles $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_L)$, where the α_n can be non-distinct. It is supported in $[0, L]$ and its causal form is characterized in the frequency domain as

$$\hat{\beta}_{\boldsymbol{\alpha}}(\omega) = \prod_{n=1}^L \frac{1 - e^{\alpha_n - j\omega}}{j\omega - \alpha_n}. \quad (2.3.1)$$

We illustrate in Figure 2.6 several exponential B-splines, where we see that a wide range of behaviors can be obtained by varying L and $\boldsymbol{\alpha}$. An exponential B-spline has the property of reproducing exponential polynomials, *i.e.*, it generates the whole family $\{e^{\alpha_n t} t^p\}_{p \in \{0, \dots, L_{\alpha_n} - 1\}}$, where L_{α_n} is the multiplicity of the element $\alpha_n \in \boldsymbol{\alpha}$. If all the poles are equal to zero (*i.e.*, $\boldsymbol{\alpha} = \mathbf{0}_L$), we obtain the classical polynomial B-splines of degree $(L - 1)$.

The most relevant properties of exponential B-splines for our purposes are [81]

- The exponential B-splines are always well-defined (*i.e.*, bounded and compactly supported), and form a Riesz basis if and only if $(\alpha_{n_1} - \alpha_{n_2}) \notin 2\pi j\mathbb{Z}$ for all purely imaginary pairs such that $n_1 \neq n_2$ [81];
- An exponential $\lambda\beta_\alpha$ satisfies the partition-of-unity condition (2.1.8) if it contains at least one vanishing pole (*i.e.*, if 0 is an element of α), with λ a normalization constant;
- The convolution of two exponential B-splines yields another B-spline of augmented order

$$\beta_{\alpha_1} * \beta_{\alpha_2} = \beta_{\alpha_1 \cup \alpha_2}, \quad (2.3.2)$$

where $(\alpha_1 \cup \alpha_2)$ denotes the concatenation of the two lists of poles α_1 and α_2 .

2.3.2 Exponential Splines and Their Periodic Counterpart

Exponential Splines

Exponential splines are naturally associated to a differential operator of a given order [84] and are defined by the following definition [81].

Definition 2.3.1. Let L_α , $\alpha = (\alpha_1, \dots, \alpha_L)$, be a differential operator of order L given by

$$L_\alpha = (D - \alpha_1 I) \cdots (D - \alpha_L I), \quad (2.3.3)$$

where D and I are the derivative and identity operators, respectively. We say that a function $r : \mathbb{R} \rightarrow \mathbb{R}$ is an exponential spline with parameter α if

$$L_\alpha r(t) = \sum_{n=1}^N a_n \delta(t - t_n), \quad (2.3.4)$$

for some integer $N \geq 1$, weights $a_n \in \mathbb{R}$, knot locations $t_n \in \mathbb{R}$, and δ is the Dirac distribution.

The exponential B-splines β_α described in Section 2.3.1 are examples of exponential splines where the knots are at the integer, *i.e.* $t_n = n$. Every cardinal exponential spline of parameter α can be expressed as a linear combination of integer shifted exponential B-splines β_α of the same order [81].

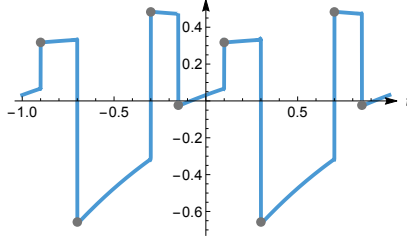


Figure 2.7: Illustration of the periodic exponential spline associated to $L_\alpha = D + I$. Dots: $N = 4$ nodes $(t_n, r(t_n))$.

Periodic Exponential Splines

Here we adapt Definition 2.3.1 to the periodic setting, where the Dirac impulse δ is replaced by the Dirac comb $\mathbb{III} = \sum_{n \in \mathbb{Z}} \delta(\cdot - n)$.

Definition 2.3.2. Consider the differential operator L_α given in (2.3.3). We say that a function $r : \mathbb{T} = [0, 1) \rightarrow \mathbb{R}$ is a periodic exponential spline with parameter α if

$$L_\alpha r(t) = \sum_{n=1}^N a_n \mathbb{III}(t - t_n) \quad (2.3.5)$$

for some integer $N \geq 1$, weights $a_n \in \mathbb{R}$, and knot locations $t_n \in \mathbb{T}$.

We illustrate the periodic exponential spline associated to $L_\alpha = D + I$ in Figure 2.7.

2.3.3 Periodic Exponential Splines as Optimal Interpolators for Closed Curves

The reconstruction of continuous signals from a sequence of samples (also known as interpolation problem) plays an essential role in signal and image processing as it constitutes a bridge between the discrete and continuous worlds. Two reconstruction paradigms over the real line have been widely developed in the literature: variational and statistical. In the variational approach, the reconstructed signal is solution of an optimization problem that establishes a tradeoff between fidelity to

the data and smoothness conditions via a regularization term [85]. In the statistical approach, the signal is modeled as a random process defined from a Gaussian white noise and is optimally reconstructed using estimation theory [86]. Two fundamental results are that 1) these two frameworks are deeply connected [87]; and 2) the solution of either problem can be expressed as a spline function in relation with a differential operator involved in regularization (variational approach) or whitening (statistical approach) [88, 89].

In the context of this thesis, we are interesting in the reconstruction of a continuous closed curve from its samples. This implies to reconstruct periodic continuous-domain functions that are the coordinate functions of the closed curve. This motivated us to develop the theory of the variational and statistical approaches in a periodic setting and in a very general context, *i.e.*, for a broad class of differential operators and for general measurements that include sampling. This theory is fully presented in Appendix B and is subject to our publication [76]. In this section, we summarize some of our results showing that periodic exponential splines naturally appear when we optimally reconstruct a closed curve from its samples.

We consider the following problem. Let $\{\mathbf{r}(t_n)\}_{n \in \{1, \dots, N\}}$, $t_n \in \mathbb{T} = [0, 1)$, be N samples of a continuous closed curve $\mathbf{r}(t) = (r_1(t), r_2(t))$, $t \in \mathbb{T}$, where the coordinate functions r_1 and r_2 are 1-periodic. We look for the optimal closed curve \mathbf{r}_{opt} that best connects its N (possibly noisy) observed data $\mathbf{y}_n = (y_{1,n}, y_{2,n}) \approx \mathbf{r}(t_n)$, for $n = 1, \dots, N$.

Variational Approach

We consider the variational problem

$$\mathbf{r}_{\text{opt}} = \begin{pmatrix} r_{1,\text{opt}} \\ r_{2,\text{opt}} \end{pmatrix} = \arg \min_{(r_1, r_2)} \left(\sum_{n=1}^N \left((y_{1,n} - r_1(t_n))^2 + (y_{2,n} - r_2(t_n))^2 \right) + \lambda \left(\|\mathbf{L}_\alpha r_1\|_{L_2}^2 + \|\mathbf{L}_\alpha r_2\|_{L_2}^2 \right) \right), \quad (2.3.6)$$

where \mathbf{L}_α is the differential operator given by 2.3.3 and the parameter λ quantifies the tradeoff between the fidelity to the data and the regularization constraint. The solution \mathbf{r}_{opt} of (2.3.6) is unique and its coordinate functions $r_{i,\text{opt}}$, $i = 1, 2$, are periodic exponential splines associated to the operator $(\mathbf{L}_\alpha^* \mathbf{L}_\alpha)$ (see Appendix B.3).

Statistical Approach

We change perspective and consider that the coordinates functions r_i , $i = 1, 2$, of \mathbf{r} are real periodic Gaussian processes with zero-mean and are related to L_α . We are looking for the optimal estimator $\mathbf{r}_{\text{opt}} = \tilde{\mathbf{r}}_{\text{MMSE}}$ over \mathbb{T} of \mathbf{r} , in the sense that each of its coordinate functions $\tilde{r}_{i,\text{MMSE}}$, $i = 1, 2$, satisfies

$$\tilde{r}_{i,\text{MMSE}} = \arg \min_{\tilde{r}_i} \mathbb{E} [\|r_i - \tilde{r}_i(\cdot|\{y_{i,n}\}_{n \in \{1, \dots, N\}})\|_{L_2}^2] \quad (2.3.7)$$

among the estimators $\tilde{r}_i(\cdot|\{y_{i,n}\}_{n \in \{1, \dots, N\}})$ of r_i such that $\tilde{r}_i(t_n|\{y_{i,n}\}_{n \in \{1, \dots, N\}})$ for $n = 1, \dots, N$. The solution of (2.3.7) is unique and is a periodic exponential spline associated to $(L_\alpha^* L_\alpha)$ (see Appendix B.4). It means that the unique optimal closed curve $\tilde{\mathbf{r}}_{\text{MMSE}}$ has coordinate functions that are periodic splines. In Figure 2.8, we optimally reconstruct (in the sense of (2.3.7)) stochastic closed curves from their observed data $\{\mathbf{y}_n\}_{n \in \{1, \dots, N\}}$ for different operators L_α .

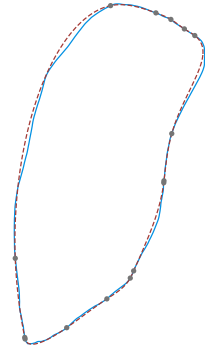
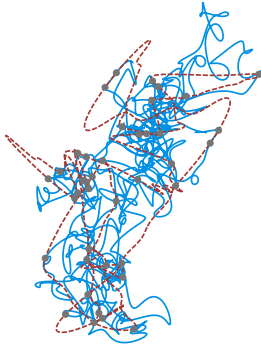
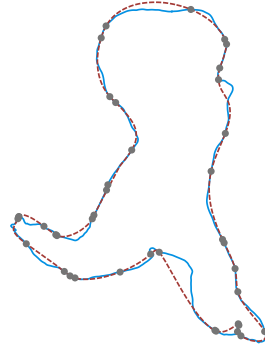
(a) $L_{\alpha} = D + I$ and $N = 40$.(b) $L_{\alpha} = D^2 + 4\pi^2 I$ and $N = 15$.(c) $L_{\alpha} = D$ and $N = 40$.(d) $L_{\alpha} = D^2$ and $N = 40$.

Figure 2.8: Reconstruction of stochastic closed curves for different operators L_{α} . Solid blue line: unknown stochastic curve \mathbf{r} ; dashed red line: estimator $\hat{\mathbf{r}}_{\text{MMSE}}$ of \mathbf{r} ; Dots: samples.

Chapter 3

Parametrization with Local Refinement

The geometric representation of active contours/surfaces determines their ability to approximate the shape of interest as well as the speed of convergence of related optimization algorithms. It is thus of great interest that one can allocate additional degrees of freedom to the curve/surface only where an increase in local detail is required.

A crucial aspect in the development of local refinement algorithms is to refine specific regions while keeping the rest of the curve/surface unchanged. This local refinement is not inherent to standard methods as Non-Uniform Rational Basis Splines (NURBS) or classical parametrizations of curves/surfaces. Existing methods to insert points at specific locations were developed in [90–94].

In this chapter, we present a new parametrization for curves and tensor-product surfaces where the degrees of freedom (*i.e.*, control points) can be locally increased without altering the shape of the curve/surface. In a segmentation context, these additional control points then allow to locally deform the shape with better accuracy. We locally improve the level of detail of the parametric model by inserting basis functions at specific locations. Our approach is generic and relies on refinable and scaling functions that are related to wavelets [95,96].

Among all scaling and refinable functions, throughout this chapter we focus on

the one of compact support, as it is a desirable property in practice.

This chapter is based on our publications [68, 69], in collaboration with D. Schmitter and M. Unser. The chapter is organized as follows: In Section 3.1 we fix the notations. We review refinable functions in Section 3.2. Finally, the main contribution is described in Section 3.3, where we propose a novel and generic parametrization of closed curves, as well as tensor-product surfaces, that are locally refinable.

3.1 Notations

We denote by t a continuous parameter in \mathbb{R} . We define $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_L)$ where the α_n can be non-distinct, and denote by L_{α_n} the multiplicity of the element $\alpha_n \in \boldsymbol{\alpha}$, for $n = 1, \dots, L$. We denote by $\varphi_{\boldsymbol{\alpha}}$ a function that reproduces exponential polynomials in span $\{e^{\alpha_n t}, \dots, t^{L_{\alpha_n}-1} e^{\alpha_n t}\}_{n \in \{1, \dots, L\}}$, *i.e.*, for all $i \in \{0, \dots, L_{\alpha_n} - 1\}$ there exists a sequence $\{c[m]\}_{m \in \mathbb{Z}}$ such as

$$t^i e^{\alpha_n t} = \sum_{m \in \mathbb{Z}} c[m] \varphi_{\boldsymbol{\alpha}}(t - m). \quad (3.1.1)$$

Exponential B-splines (see Section 2.3.1) are examples of such functions [81].

3.2 Scaling and Refinable Functions

In this section, we define the notions of scaling and refinable functions.

Definition 3.2.1. *A basis function φ is called a scaling function for a refinement factor $\rho > 1$ if it verifies the refinement relation given by*

$$\varphi(t) = \sum_{m \in \mathbb{Z}} h[m] \varphi(\rho t - m), \quad (3.2.1)$$

where h is the discrete refinement filter [97, 98].

If φ is a scaling function, it can thus be expressed as a linear combination of its contracted version shifted by integers. Examples of such functions are polynomial

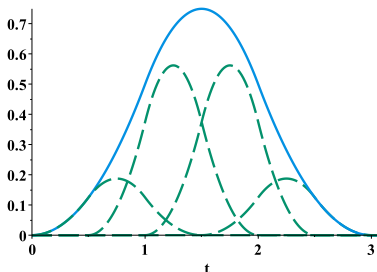


Figure 3.1: Refinement of a quadratic B-spline with a refinement factor ρ equal to 2 (solid blue line). It can be expressed as a linear combination of four contracted versions (green dashed lines) of integer-shifted quadratic B-splines.

B-splines [81]. In Figure 3.1, we illustrate the refinement of a quadratic B-spline for $\rho = 2$, whose corresponding refinement filter is defined by its z -transform as

$$H(z) = \frac{1}{4} (1 + z^{-1})^3. \quad (3.2.2)$$

Definition 3.2.2. A basis function φ_{α} is called a refinable function¹ for a refinement filter $\rho > 1$, if it verifies the refinement relation given by

$$\varphi_{\alpha}(t) = \sum_{m \in \mathbb{Z}} h_{\alpha, \rho}[m] \varphi_{\frac{\alpha}{\rho}}(\rho t - m), \quad (3.2.3)$$

where $\{h_{\alpha, \rho}[m]\}_{m \in \mathbb{Z}}$ are the coefficients of the discrete refinement filter [101, 102].

The non-standard aspect here is the fact that the scheme is non-stationary, meaning that the basis functions on both sides of (3.2.3) involve different parameters, *i.e.*, α and $\frac{\alpha}{\rho}$. If φ_{α} is a refinable function, its dilatation by ρ can be expressed as a linear combination of the integer shifts of the generator $\varphi_{\frac{\alpha}{\rho}}$. Examples of such functions are exponential B-splines [81]. When $\alpha = \mathbf{0}$, we obtain a scaling function as defined in Definition 3.2.1.

¹In relation with the classical definition of a scaling function used in multiresolution models [99], refinable functions defined in Definition 3.2.2 are sometimes also called scaling functions [99, 100].

When the functions φ and φ_α in (3.2.1) and (3.2.3) are compactly supported, then the support of h and $h_{\alpha,\rho}$ are compact as well. We denote by $\{n_0, \dots, n_0 + N - 1\}$ the support of the filters where N is the size.

3.3 Representation with Local Refinement

In the following, we first describe the local refinement of parametric closed curves. We use scaling functions in order to present the theory in a clear manner, but the extension to refinable functions is straightforward. We then extend the theory in 3D using the general case of refinable functions.

3.3.1 Closed Curves

We consider a closed curve \mathbf{r} parametrized by (2.1.2), where we choose (compactly supported) scaling functions (see Definition (3.2.1)) as basis functions. The shape is encoded by M control points and we briefly recall the parametrization as

$$\mathbf{r}(t) = \sum_{m=0}^{M-1} \mathbf{c}[m] \varphi_M(t - m), \quad (3.3.1)$$

where $t \in [0, M[$. In a non-periodic formulation (see (2.1.1)), the local refinement of \mathbf{r} with respect to one particular control point $\mathbf{c}[p]$ consists in replacing the shifted basis function $\varphi(\cdot - p)$ by the linear combination of its N contracted version given by (3.2.1). To ensure the periodicity in (3.3.1), we do it for each scaling function $\varphi(\cdot - p - nM)$, $n \in \mathbb{Z}$. In Proposition 3.3.1, we derive the new formulation of \mathbf{r} in which we locally increased its number of control points.

Proposition 3.3.1. *A parametric closed curve that has been locally refined with respect to $\mathbf{c}[p]$ can be expressed as*

$$\mathbf{r}(t) = \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \mathbf{c}[m] \varphi_M(t - m) + \sum_{n=n_0}^{n_0+N-1} \tilde{\mathbf{c}}_p[n] \varphi_{\rho M}(\rho t - \rho p - n), \quad (3.3.2)$$

where ρ is the refinement factor and N is the size of the discrete filter h , whose support is $\{n_0, \dots, n_0 + N\}$. The functions φ_M and $\varphi_{\rho M}$ are the M - and ρM -

periodization of φ as defined by (2.1.3), respectively, and

$$\tilde{\mathbf{c}}_p[n] = h[n]\mathbf{c}[p]. \quad (3.3.3)$$

The proof is given in Appendix 3.4.1. We give Proposition 3.3.1 for closed curves as we focus on those models in this thesis. However, the extension of this Proposition to open curves is straightforward. The local refinement described by Proposition 3.3.1 allows the part of the curve initially controlled by $\mathbf{c}[p]$ to be described by N new control points $\{\tilde{\mathbf{c}}_p[n]\}_{n \in \{n_0, \dots, n_0 + N - 1\}}$. We thus increase the approximation power of the curve at this specific region. By approximation power we mean the ability of the model to approximate a shape with accuracy. The error of approximation decreases when the number of control points increases [103]. The local refinement of a parametric curve for $\rho = 2$ is illustrated in Figure 3.2.

3.3.2 Tensor-Product Surfaces

The idea to locally refine tensor-product surfaces is similar than in 2D. We use (compactly supported) refinable functions as generators φ_{α_1} and φ_{α_2} to construct surfaces described by (2.1.5). We apply the refinement relation (3.2.3) locally, *i.e.*, only with respect to the particular control point $\mathbf{c}[p, q]$ associated to the basis functions $\varphi_{\alpha_1}(u - p)$ and $\varphi_{\alpha_2}(v - q)$. The refinement factors are equal to ρ_1 and ρ_2 in the directions u and v , respectively.

Proposition 3.3.2. *A locally refined parametric tensor-product surface is given by*

$$\begin{aligned} \sigma(u, v) = & \sum_{\substack{(m,n) \in \mathbb{Z}^2 \\ (m,n) \neq (p,q)}} \mathbf{c}[m, n] \varphi_{\alpha_1}(u - m) \varphi_{\alpha_2}(v - n) \\ & + \sum_{i=n_1}^{n_1 + N_1 - 1} \sum_{j=n_2}^{n_2 + N_2 - 1} \tilde{\mathbf{c}}_{p,q}[i, j] \varphi_{\frac{\alpha_1}{\rho_1}}(\rho_1 u - \rho_1 p - i) \varphi_{\frac{\alpha_2}{\rho_2}}(\rho_2 v - \rho_2 q - j), \end{aligned} \quad (3.3.4)$$

where N_1 and N_2 are the sizes of the discrete filters $h_{\frac{\alpha_1}{\rho_1}, \rho_1}$ and $h_{\frac{\alpha_2}{\rho_2}, \rho_2}$, respectively, whose supports are $\{n_1, \dots, n_1 + N_1 - 1\}$ and $\{n_2, \dots, n_2 + N_2 - 1\}$, and

$$\tilde{\mathbf{c}}_{p,q}[i, j] = \mathbf{c}[p, q] h_{\frac{\alpha_1}{\rho_1}, \rho_1}[i] h_{\frac{\alpha_2}{\rho_2}, \rho_2}[j]. \quad (3.3.5)$$

Thereby, p and q are freely chosen.

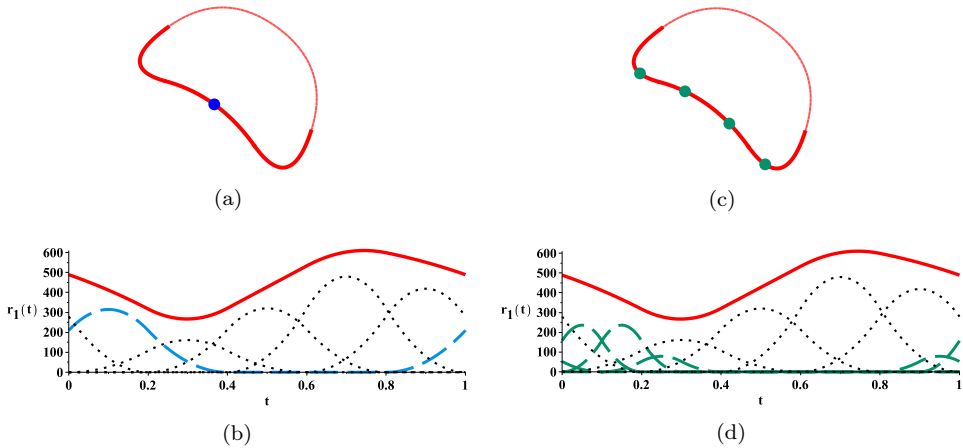


Figure 3.2: Locally refinable closed curve. (a) A parametric closed curve \mathbf{r} represented with the quadratic B-spline and $M = 5$. The bold line corresponds to the part of the curve that is controlled by $\mathbf{c}[p]$ (blue dot); (b) the first coordinate function of \mathbf{r} with its shifted basis functions (dotted lines). We highlight $\varphi(\cdot - p)$ in a blue dashed line; (c) the curve \mathbf{r} after local refinement with respect to $\mathbf{c}[p]$. We used the refinement filter defined by (3.2.2). The bold line of the curve initially encoded by $\mathbf{c}[p]$ is now controlled by four new control points $\{\tilde{\mathbf{c}}_p[n]\}_{n \in \{0, \dots, 3\}}$ (green dots.). The shape of \mathbf{r} remains unchanged; (d) the first coordinate function of \mathbf{r} , where the blue dashed line in (b) was replaced by four new basis functions (green dashed lines), such as in Figure 3.1. As the quadratic B-spline is not an interpolating function, we projected the control points on the curve for better clarity of these plots.

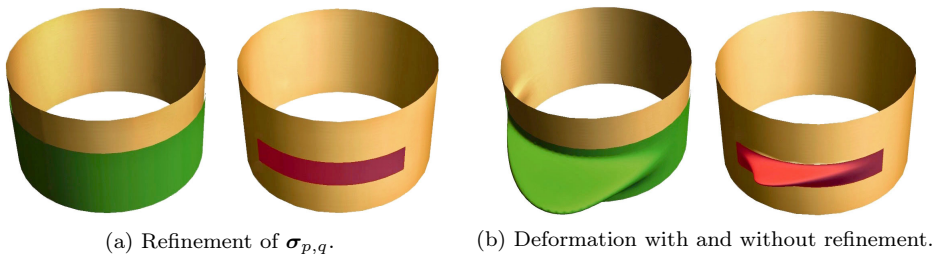


Figure 3.3: Local refinement of a cylindrical surface. (a) One of the small areas (red pattern) obtained by the local refinement of $\sigma_{p,q}$ (green pattern); (b) the displacement of the control point $\mathbf{c}[p, q]$ deforms the entire region $\sigma_{p,q}$ (left), while the local refinement with respect to $\mathbf{c}[p, q]$ allows to accurately control a specific part of $\sigma_{p,q}$ (right).

The proof is given in Appendix 3.4.2. The infinite sums in (3.3.4) are in practice reduced to finite ones, as we consider compactly supported basis functions. One can refine the surface at several specific locations by applying Proposition 3.3.2 with respect to each corresponding control point.

We denote by $\sigma_{p,q}$ the region of the surface σ that is initially controlled by $\mathbf{c}[p, q]$, *i.e.*, $\sigma_{p,q}(u, v) = \mathbf{c}[p, q]\varphi_{\alpha_1}(u-p)\varphi_{\alpha_2}(v-q)$. The local refinement described by Proposition 3.3.2 leaves the shape of the surface unchanged while dividing $\sigma_{p,q}$ in $(N_1 \times N_2)$ smaller areas; each being controlled by one of the new control points $\{\tilde{\mathbf{c}}_{p,q}[i, j]\}_{i \in \{n_1, \dots, n_1+N_1-1\}, j \in \{n_2, \dots, n_2+N_2-1\}}$. We thus increase the approximation power of the surface at the specific region $\sigma_{p,q}$. In Figure 3.3, we illustrate a local refinement on a cylindrical surface. We show the influence of the refinement for each direction u and v in Figure 3.4. Finally, in Figure 3.5, we compare our local approach to the global one used with the classical parametrization (2.1.2).

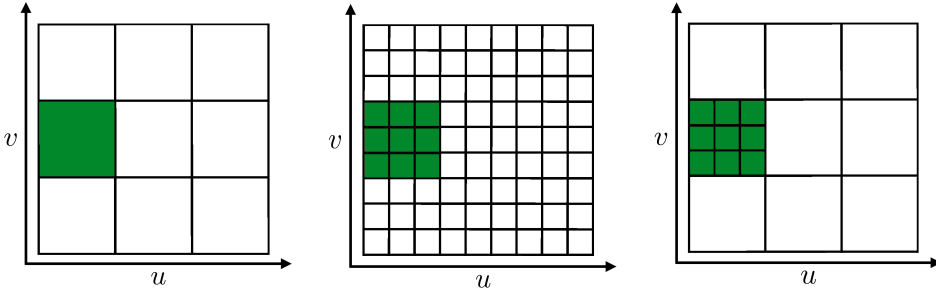


(a) Refinement in the direction u with $\rho_1 = 2$.

(b) Refinement in the direction v with $\rho_2 = 2$.

(c) Refinement in both directions u and v with $\rho_1 = 2$ and $\rho_2 = 2$.

Figure 3.4: Refinement of $\sigma_{p,q}$ (green pattern) with respect to $\mathbf{c}[p,q]$ performed in direction u in (a), v in (b) as well as in both directions in (c). Red pattern: surface controlled by one of the new points $\tilde{\mathbf{c}}_{p,q}$.



(a) Initial parameter domain.

(b) Global approach.

(c) Local refinement.

Figure 3.5: Increase of the approximation power of the region $\sigma_{p,q}$ (green pattern). We compare our method to the global approach used with classical parametrizations. (a) Initial configuration; (b) configuration obtained after globally increasing by a factor 3 the number of control points in each direction. The entire parameter domain is affected; (c) configuration obtained by applying a local refinement with respect to the control point $\mathbf{c}[p,q]$ with $\rho_1 = \rho_2 = 3$. The region $\sigma_{p,q}$ is divided into 9 areas, whereas the rest of the parameter domain remains unchanged.

3.4 Appendices

3.4.1 Proof of Proposition 3.3.1

Using (2.1.2) we write

$$\mathbf{r}(t) = \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \mathbf{c}[m] \varphi_M(t-m) + \mathbf{c}[p] \varphi_M(t-p). \quad (3.4.1)$$

By combining (2.1.3) and (3.2.1), $\varphi_M(t-p)$ can be expressed as

$$\begin{aligned} \varphi_M(t-p) &= \sum_{n \in \mathbb{Z}} \varphi(t-p-Mn) \\ &= \sum_{n \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} h[m] \varphi(\rho t - \rho p - \rho Mn - m) \\ &= \sum_{m \in \mathbb{Z}} h[m] \sum_{n \in \mathbb{Z}} \varphi(\rho t - \rho p - \rho Mn - m) \\ &= \sum_{m \in \mathbb{Z}} h[m] \varphi_{\rho M}(\rho t - \rho p - m). \end{aligned} \quad (3.4.2)$$

Therefore,

$$\mathbf{c}[p] \varphi_M(t-p) = \sum_{m \in \mathbb{Z}} \underbrace{\mathbf{c}[p] h[m]}_{\tilde{\mathbf{c}}_p[m]} \varphi_{\rho M}(\rho t - \rho p - m). \quad (3.4.3)$$

By taking into account the size of the filter h , which is equal to N , as well as its localization on $\{n_0, \dots, n_0 + N\}$ we can simplify the infinite sum in (3.4.3) to obtain

$$\mathbf{c}[p] \varphi_M(t-p) = \sum_{m=n_0}^{n_0+N-1} \tilde{\mathbf{c}}_p[m] \varphi_{\rho M}(\rho t - \rho p - m). \quad (3.4.4)$$

By combining (3.4.1) and (3.4.4) we obtain (3.3.2). ■

3.4.2 Proof of Proposition 3.3.2

Using (2.1.5) we write

$$\sigma(u, v) = \sum_{\substack{(m,n) \in \mathbb{Z}^2 \\ (m,n) \neq (p,q)}} \mathbf{c}[m, n] \varphi_{\alpha_1}(u - m) \varphi_{\alpha_2}(v - n) + \underbrace{\mathbf{c}[p, q] \varphi_{\alpha_1}(u - p) \varphi_{\alpha_2}(v - q)}_{\sigma_{p,q}(u,v)}. \quad (3.4.5)$$

Using the refinement property (3.2.3) we obtain

$$\begin{aligned} \sigma_{p,q}(u, v) &= \mathbf{c}[p, q] \left(\sum_{i \in \mathbb{Z}} h_{\frac{\alpha_1}{\rho_1}, \rho_1} [i] \varphi_{\frac{\alpha_1}{\rho_1}}(\rho_1 u - \rho_1 p - i) \right) \left(\sum_{j \in \mathbb{Z}} h_{\frac{\alpha_2}{\rho_2}, \rho_2} [j] \varphi_{\frac{\alpha_2}{\rho_2}}(\rho_2 v - \rho_2 q - j) \right) \\ &= \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \underbrace{\mathbf{c}[p, q] h_{\frac{\alpha_1}{\rho_1}, \rho_1} [i] h_{\frac{\alpha_2}{\rho_2}, \rho_2} [j]}_{\tilde{\mathbf{c}}_{p,q}[i,j]} \varphi_{\frac{\alpha_1}{\rho_1}}(\rho_1 u - \rho_1 p - i) \varphi_{\frac{\alpha_2}{\rho_2}}(\rho_2 v - \rho_2 q - j). \end{aligned} \quad (3.4.6)$$

To simplify the infinite sums in (3.4.6) we take into account the size of the filters $h_{\frac{\alpha_1}{\rho_1}, \rho_1}$ and $h_{\frac{\alpha_2}{\rho_2}, \rho_2}$, which is equal to N_1 and N_2 , respectively, as well as their localization on $\{n_1, \dots, n_1 + N_1 - 1\}$ and $\{n_2, \dots, n_2 + N_2 - 1\}$. We obtain

$$\sigma_{p,q}(u, v) = \sum_{i=n_1}^{n_1+N_1-1} \sum_{j=n_2}^{n_2+N_2-1} \tilde{\mathbf{c}}_{p,q}[i, j] \varphi_{\frac{\alpha_1}{\rho_1}}(\rho_1 u - \rho_1 p - i) \varphi_{\frac{\alpha_2}{\rho_2}}(\rho_2 v - \rho_2 q - j). \quad (3.4.7)$$

By combining (3.4.5) and (3.4.7), we obtain (3.3.4). ■

Chapter 4

Subdivision-Based Representation

As exposed in Chapter 2, parametric snakes have a continuous representation via the use of basis functions. They are parametrized by only a few control points, which results in a faster optimization and better robustness. They are usually built in a way that ensures continuity and smoothness, and it is easy to introduce shape constraints. However, two well-known drawbacks of parametric approaches are (i) the restricted nature of the shape that they can generate; and (ii) their inability to deal with topology changes such as contour merging and splitting. On the contrary, point-snakes/active meshes can handle topology changes. In addition, their discrete nature allows for an easy implementation and is compatible with open-source libraries for optimization or visualization. However, they rely on a large number of parameters (*i.e.*, snake points/mesh vertices), which requires an internal regularization term and makes the optimization more challenging.

In this chapter, we present a geometric representation that combines the advantages of point-snakes and parametric snakes. In our representation, the curve/surface is driven by a set of a few master points, the control points, that are the parameters of the model. Then, slave points describing the curve/surface are generated by specific iterative procedures. The property that makes it possible is called *subdivision* [101, 104–106], which is one of the basic geometric tools in computer

graphics for representation and modeling [107, 108]. It is tightly linked to the theory of wavelets [109] and allows describing a contour/surface of arbitrary topology [71, 110–112] by an initial set of a few control points which, by the iterative application of refinement rules, becomes continuous in the limit. The discrete nature of the representation is convenient in practical applications. At the same time, it implicitly yields a continuously defined model whose smoothness depends on the particular choice of the subdivision mask. The main benefits of subdivision schemes are their simplicity of implementation, the possibility to control their order of approximation, and their multiresolution property, which allows for the contour of a shape to be represented at varying resolutions.

This chapter is based on our publications [70, 71] and work [72], in collaboration with P. Novara, L. Romani, D. Schmitter, V. Uhlmann and M. Unser. It is organized as follows: We first fix the notations in Section 4.1. In Section 4.2, we introduce and describe the theory of subdivision that is relevant to the construction of closed curves. In Section 4.3, we present several subdivision schemes that possess various properties such as being interpolatory (a useful property for user-interactive applications), having different sizes of support, and reproducing polynomials. In Section 4.4, we show how subdivision schemes can be used to reproduce trigonometric functions for the construction of elliptic and circular curves. In Section 4.5, we emphasize the connection between the subdivision-based and parametric representations. Finally, in Section 4.6 we extend the theory to the construction of subdivision surfaces.

4.1 Notations

We represent by $\mathbf{p}[\cdot]$ a discrete sequence of points $\mathbf{p}[m] = (p_1[m], p_2[m])$, indexed by $m \in \mathbb{Z}$, where p_1 and p_2 are the corresponding coordinates. We write $\mathbf{p}_{(k)}[\cdot] = (p_{1(k)}[\cdot], p_{2(k)}[\cdot])$ to describe a $(2^k M)$ -periodic sequence, $k \geq 0$, with the property that $\mathbf{p}_{(k)}[m + n2^k M] = \mathbf{p}_{(k)}[m]$, $\forall n \in \mathbb{Z}$. The discrete convolution of $\mathbf{p}_{(k)}[\cdot]$ with a scalar mask $h[\cdot]$ is defined as

$$(h * \mathbf{p}_{(k)})[m] = \sum_{n \in \mathbb{Z}} h[m - n] \mathbf{p}_{(k)}[n]. \quad (4.1.1)$$

4.2 Subdivision Schemes

A subdivision scheme generates a continuously defined function as the limit of an iterative algorithm that is applied to an initial set of M control points. A *refinement rule* is applied repeatedly k times to double the number of points at each iteration, ultimately yielding a set of $2^k M$ points. Note that, at each iteration, the new set of points does not necessarily contain the previous ones. The subdivision scheme is said to be *convergent* when the set of points converges to the continuous curve $\mathbf{r} = (r_1, r_2)$ with $r_1, r_2 \in \mathcal{C}^0$ as $k \rightarrow \infty$.

A closed curve at resolution k is represented by a $(2^k M)$ -periodized coordinate sequence $\mathbf{p}_{(k)}[\cdot]$. The refinement rule from $(k-1)$ to k is defined by

$$\mathbf{p}_{(k)}[m] = h * \mathbf{p}_{(k-1)\uparrow_2}[m], \quad (4.2.1)$$

where h is the *subdivision mask* of the subdivision scheme [113] and \uparrow_2 denotes an upsampling by a factor of 2, given by

$$\mathbf{p}_{(k)\uparrow_2}[m] = \begin{cases} \mathbf{p}_{(k)}[n], & m = 2n \\ 0, & \text{otherwise.} \end{cases} \quad (4.2.2)$$

In practice, the mask h has a finite number of non-zero elements so that the infinite sum in (4.2.1) is often reduced to a finite one. Applying (4.2.1) iteratively, we can express the refinement rule as a function of the initial set of control points $\mathbf{p}_{(0)}$. The *subdivision points* at the k th iteration ($k \geq 1$) are thereby described by

$$\mathbf{p}_{(k)} = h_{0 \rightarrow k} * \mathbf{p}_{(0)\uparrow_{2^k}}, \quad (4.2.3)$$

where

$$h_{0 \rightarrow k} = h_{\uparrow_{2^{k-1}}} * h_{\uparrow_{2^{k-2}}} * \cdots * h_{\uparrow_2} * h. \quad (4.2.4)$$

The derivation of (4.2.3) is given in Appendix 4.7.1. Note that each set of points $\mathbf{p}_{(k)}$ is encoded with the M control points $\{\mathbf{p}_{(0)}[m]\}_{m \in \{0, \dots, M-1\}}$. The subdivision scheme is illustrated in Figures 4.1 and 4.2.

In the following, the term *control points* designates the M initial points $\{\mathbf{p}_{(0)}[m]\}_{m \in \{0, \dots, M-1\}}$ and the term *subdivision points* describes the $2^k M$ points $\{\mathbf{p}_{(k)}[m]\}_{m \in \{0, \dots, 2^k M-1\}}$ at the k th iteration ($k \geq 1$).

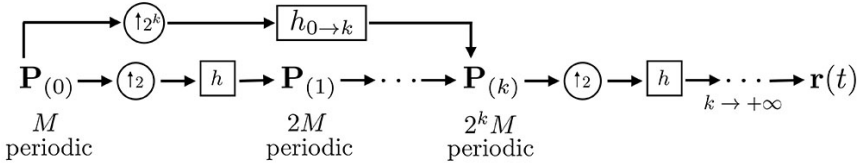


Figure 4.1: Flowchart of a subdivision scheme. The periodic sequence $\mathbf{p}_{(k)}$, associated to the subdivision points at iteration k , converges to the continuous curve \mathbf{r} ; h is the subdivision mask and the sequence $h_{0 \rightarrow k}$, defined by (4.2.4), allows obtaining $\mathbf{p}_{(k)}$ directly from the initial set of control points $\mathbf{p}_{(0)}$.

4.2.1 Convergent Subdivision Schemes

Let h be a subdivision mask with z -transform¹ $H(z) = \sum_{n \in \mathbb{Z}} h[n]z^n$, $z \in \mathbb{C} \setminus \{0\}$. A necessary condition for the corresponding (stationary) subdivision scheme to be convergent [114] is that

$$\sum_{n \in \mathbb{Z}} h[2n] = \sum_{n \in \mathbb{Z}} h[2n+1] = 1. \quad (4.2.5)$$

This condition is similar to the partition-of-unity condition (2.1.8) presented for basis functions in Section 2.1.3. The subdivision scheme thus reproduces constants and $H(z) = (1+z)B(z)$, where $B(z)$ is a Laurent polynomial and $B(1) = 1$ [115].

For any convergent subdivision scheme, the points of the sequence $\mathbf{p}_{(k)}$, as $k \rightarrow \infty$, sample the limit curve \mathbf{r} , in the sense that [115–117]

$$\mathbf{r}(t) \Big|_{t=\frac{m}{2^k}} = \mathbf{p}_{(k)}[m]. \quad (4.2.6)$$

When the coordinates function of the curve satisfy $r_1, r_2 \in \mathcal{C}^1$, the derivative $\dot{\mathbf{r}} = \frac{d\mathbf{r}}{dt}$ is also sampled by

$$\dot{\mathbf{r}}(t) \Big|_{t=\frac{m}{2^k}} = 2^k (\mathbf{p}_{(k)}[m+1] - \mathbf{p}_{(k)}[m]) \quad (4.2.7)$$

in the limit case $k \rightarrow \infty$ [116, 118]. The derivation of (4.2.7) is given in Appendix 4.7.2. A necessary and sufficient condition for a subdivision scheme to

¹This is the conventional definition of the z -transform used in subdivision theory.

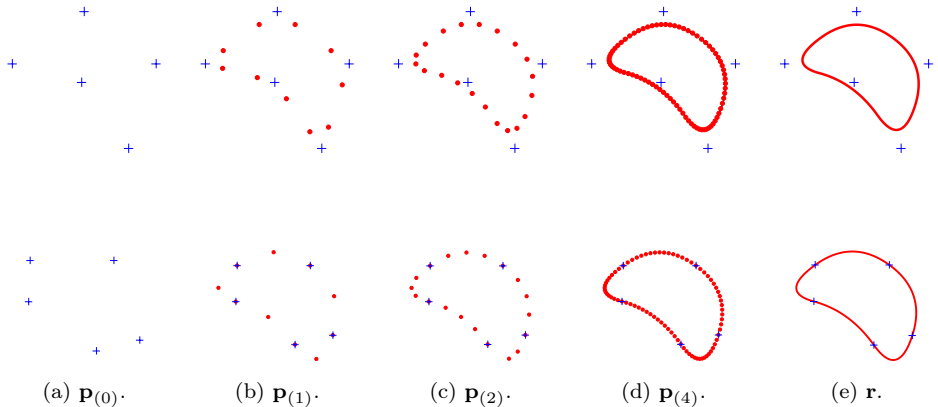


Figure 4.2: Illustration of non-interpolating (first row) and interpolating (second row) subdivision schemes. (a) Control points. Dots in (b)-(d): Subdivision points of the first, second and fourth iterations. As the points become denser with each iteration, they converge to the continuous curve \mathbf{r} (e), which is still encoded by the five control points (blue crosses). In the case of the interpolating scheme, the subdivision points interpolate the limit curve at each iteration of the process.

converge uniformly to a continuous limit function is [114, 118]

$$\begin{cases} H(1) = 2 \\ H(-1) = 0 \\ \max_m |h_{0 \rightarrow k}[m+1] - h_{0 \rightarrow k-1}[m]| \xrightarrow{k \rightarrow +\infty} 0. \end{cases} \quad (4.2.8)$$

In practice, few iterations are enough for the contour points to be sufficiently dense.

4.2.2 Interpolating Subdivision Schemes

A subdivision scheme is said to be interpolating if $h[2m] = \delta[m]$, where δ denotes the Kronecker delta. It means that, at each step k , the subdivision points interpolate the limit curve \mathbf{r} and $\{\mathbf{p}_{(k-1)}[m]\}_{m \in \mathbb{Z}} \subset \{\mathbf{p}_{(k)}[m]\}_{m \in \mathbb{Z}}$. We illustrate an interpolating subdivision scheme in Figure 4.2 (second row).

4.2.3 Affine Invariance

As mentioned in Section 2.1.3, an important requirement for the construction of snakes, as it is our final motivation, is that the representation model be affine invariant to ensure to describe the curve independently from its location and orientation.

Definition 4.2.1. *A subdivision scheme is said to be affine invariant if, for any (2×2) matrix \mathbf{A} and translation vector $\mathbf{b} \in \mathbb{R}^2$, the following relation holds:*

$$\begin{aligned} \lim_{k \rightarrow \infty} h_{0 \rightarrow k} * \left(\mathbf{A} \mathbf{p}_{(0)} + \mathbf{b} \right) \uparrow_{2^k} &= \mathbf{A} \left(\lim_{k \rightarrow \infty} h_{0 \rightarrow k} * \mathbf{p}_{(0)} \right) + \mathbf{b} \\ &= \mathbf{A} \mathbf{r} + \mathbf{b}. \end{aligned} \tag{4.2.9}$$

Proposition 4.2.2. *Every convergent subdivision scheme is affine invariant.*

The derivation of Proposition 4.2.2 is given in Appendix 4.7.3.

4.3 Design of Subdivision Schemes

In the context of segmentation, the continuously defined limit curve of a convergent subdivision scheme can be used to describe an active contour. In this case, the properties of the snake are determined by the choice of the subdivision mask h . There are thus three important properties to consider for h . The first defines its capability to perfectly reproduce specific shapes, such as polynomial or trigonometric curves. The second is whether the control points interpolate the curve or not. The third is the support of the mask. This can affect the optimization of the snake and, generally, a short mask is preferred over a large one in a context of user interaction. In practice, a tradeoff between the advantages and limitations regarding these properties has to be made. The purpose of this section is to offer guidance on the choice of the subdivision mask. We discuss the two most interesting families: the Deslauriers-Dubuc and the minimum-support subdivision schemes.

4.3.1 Generation of Polynomials

Proposition 4.3.1 gives a criterion that a subdivision scheme must verify to generate polynomials.

Proposition 4.3.1. (*Conti and Hormann [115, Equation (7)]*) A subdivision scheme generates polynomials up to degree $(L-1)$ if the z -transform of the subdivision mask takes the form

$$H(z) = (1+z)^L B(z), \quad (4.3.1)$$

where $B(z)$ is a Laurent polynomial with $B(1) = \frac{1}{2^{L-1}}$ and $B(-1) \neq 0$.

4.3.2 Deslauriers-Dubuc Subdivision Scheme

The Deslauriers-Dubuc subdivision scheme is convergent and interpolating [119], [120]. It reproduces polynomials up to degree $(L-1)$ [109, 121, 122]. The mask has a support of size $2(L-1) + 1$ and is computed by solving the system [100, 123]

$$\begin{cases} H(z) + H(-z) = 2 \\ H(z) = R(z)Q(z), \end{cases} \quad (4.3.2)$$

where $R(z) = (1+z)^L$ and $Q(z)$ is the shortest-possible polynomial. We solve (4.3.2) using *Bézout's* theorem and we obtain

$$H(z) = (-1)^{\frac{L}{2}} (1-z^2)^L z^{-L} \left(\sum_{q=1}^L \frac{(-1)^q a_q}{(z-1)^q} \right), \quad (4.3.3)$$

where $\{a_q\}_{q \in \{1, \dots, L\}}$ are the coefficients of the simple-fraction decomposition

$$\frac{2(-1)^{\frac{L}{2}} z^L}{(z^2-1)^L} = \sum_{q=1}^L a_q \left(\frac{1}{(z+1)^q} + \frac{(-1)^q}{(z-1)^q} \right). \quad (4.3.4)$$

Example-Reproduction of Third-Degree Polynomials: We now focus on the particular case when $L = 4$. It corresponds to the well-known subdivision scheme introduced by Deslauriers and Dubuc in [119] that reproduces polynomials up to degree 3. The corresponding subdivision mask h has a support of size 7 and its z -transform is defined by

$$H(z) = -\frac{1}{16}z^{-3} + \frac{9}{16}z^{-1} + 1 + \frac{9}{16}z - \frac{1}{16}z^3. \quad (4.3.5)$$

4.3.3 Minimum-Support Subdivision Scheme

The minimum-support subdivision scheme has the property to generate polynomials with the shortest mask. However, it is not interpolating, meaning that the control points do not lie on the limit curve. In a segmentation context, it results that it is less intuitive for the user to interact with the curve. The mask associated to the scheme that generates polynomials up to degree $(L - 1)$ is defined as

$$H(z) = \frac{1}{2^{L-1}} (1 + z)^L \quad (4.3.6)$$

and has a support of size $L + 1$ [124].

Example-Shortest Generation of Third-Degree Polynomials: In this example, we construct a minimum-support subdivision scheme that generates polynomials up to degree 3. The corresponding mask is of size 5 and is defined by

$$H(z) = \frac{1}{8} + \frac{1}{2}z + \frac{3}{4}z^2 + \frac{1}{2}z^3 + \frac{1}{8}z^4. \quad (4.3.7)$$

4.4 Design of Non-Stationary Subdivision Schemes

The subdivision schemes that we have described so far are called *stationary*, meaning that the subdivision mask h is the same at each iteration k . A subdivision scheme is called *non-stationary* if the subdivision mask h_k is different at each iteration k , with the rest of the procedure being the same as in Section 4.2. Non-stationary subdivision schemes are required to reproduce exponential polynomials, which allows to construct trigonometric functions. The refinement rule is now

$$\mathbf{p}^{(k)} = h_k * \mathbf{p}^{(k-1)}_{\uparrow 2}, \quad (4.4.1)$$

where h_k is the subdivision mask at the k th iteration. The relation between the periodic sequence $\mathbf{p}^{(k)}$ at the k th iteration and the control points $\mathbf{p}_{(0)}$ is still defined by (4.2.3) but $h_{0 \rightarrow k}$ is now computed by

$$h_{0 \rightarrow k} = h_{1 \uparrow_{2^{k-1}}} * h_{2 \uparrow_{2^{k-2}}} * \cdots * h_{k-1 \uparrow_2} * h_k. \quad (4.4.2)$$

If we set $h = h_k$, we recover all the formulas of the stationary scheme. Furthermore, every convergent stationary subdivision scheme verifies the property of affine invariance stated in Definition 4.2.1 (see Proposition 4.2.2). In the non-stationary setting, however, it must be verified case by case [115].

4.4.1 Generation of Exponential Polynomials

We define $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_L)$ and denote by L_{α_n} the multiplicity of the element $\alpha_n \in \alpha$, for $n = 1, \dots, L$. A non-stationary subdivision scheme is said to generate exponential polynomials if it generates the whole family $\{e^{\alpha_m t} t^n\}_{n \in \{0, \dots, L_{\alpha_m} - 1\}}$. In this case, the subdivision mask at the k th iteration is characterized by $\alpha_k = \frac{\alpha}{2^k}$ and its z -transform is denoted by H_k^α .

4.4.2 Generation of Trigonometric Functions

The generation of trigonometric functions allows one to efficiently construct a scheme that is capable of generating circles and ellipses, which are useful structures in the context of segmentation in bioimaging. We now present a criterion that a (non-stationary) subdivision scheme must verify to generate trigonometric functions.

Proposition 4.4.1. *(Romani [102, Proposition 2]) A non-stationary subdivision scheme perfectly generates ellipses if the z -transform of the subdivision mask at the k th iteration verifies*

$$H_k(z) = (1+z) \left(1 + e^{\frac{j2\pi}{2^k M}} z\right) \left(1 + e^{\frac{-j2\pi}{2^k M}} z\right) Q_k(z), \quad (4.4.3)$$

where $Q_k(z)$ is a polynomial in z .

That means that the subdivision scheme has to generate exponential polynomials and that $\left(0, \frac{j2\pi}{M}, \frac{-j2\pi}{M}\right) \subset \alpha$. In the following we provide two examples of ellipse-generating subdivision schemes: the non-stationary Deslauriers-Dubuc and the non-stationary minimum-support subdivision schemes.

4.4.3 Non-Stationary Deslauriers-Dubuc Subdivision Scheme

The non-stationary Deslauriers-Dubuc subdivision scheme is interpolating and capable of reproducing the exponential polynomials defined in Section 4.4.1 [100, 123], [125]. As for the stationary case, the mask at the k th iteration has a support of size $2(L-1)+1$ and is obtained by solving

$$\begin{cases} H_k^\alpha(z) + H_k^\alpha(-z) = 2 \\ H_k^\alpha(z) = R^{\alpha_k}(z)Q_k(z), \end{cases} \quad (4.4.4)$$

where $R^\alpha(z) = \prod_{n=1}^L (1 + e^{\alpha_n} z)$, $\alpha_k = \frac{\alpha}{2^k}$, and $Q_k(z)$ is a polynomial in z . Vonesch *et al.* [100] extensively studied this scheme and proposed simplified solutions to solve (4.4.4) by applying *Bézout's identity*

$$C_k(Z)D_k(Z) + C_k(-Z)D_k(-z) = 2, \quad (4.4.5)$$

where $Z = \frac{z+z^{-1}}{2}$, $C_k(Z) = z^{-\frac{k}{2}} R^{\alpha_k}(z)$, and $D_k(Z) = z^{\frac{k}{2}} Q_k(z)$. The shortest polynomial $D_k(Z)$ is given by

$$D_k(Z) = \left(\sum_{q=1}^K \sum_{s=1}^{L_q} \frac{(-1)^s a_{q,s}}{(Z + Z_q)^s} \right) C_k(-Z), \quad (4.4.6)$$

where $K < L$ is the number of different elements of α , $\{Z_q\}_{q \in \{1, \dots, K\}}$ are the roots of $C_k(Z)$ with multiplicity L_q , and $\{a_{q,s}\}_{q \in \{1, \dots, K\}, s \in \{1, \dots, L_q\}}$ are the coefficients of the simple-fraction decomposition

$$\frac{2}{C_k(-Z)C_k(Z)} = \sum_{q=1}^K \sum_{s=1}^{L_q} a_{q,s} \left(\frac{1}{(Z - Z_q)^s} + \frac{(-1)^s}{(Z + Z_q)^s} \right). \quad (4.4.7)$$

Example-Ellipse-Reproducing Scheme: We construct a non-stationary Deslauriers-Dubuc subdivision scheme that is capable of reproducing ellipses. Therefore, we want to be able to construct trigonometric functions. According to Proposition 4.4.1, $\left(0, \frac{j2\pi}{M}, \frac{-j2\pi}{M}\right) \subset \alpha$. Moreover, it was shown in [100] that the elements of α must come in complex-conjugate pairs and that, if 0 is an element of α , then it must have even multiplicity. Hence, $\alpha = \left(0, 0, \frac{2j\pi}{M}, -\frac{2j\pi}{M}\right)$. The mask at iteration k is of size 7. By solving (4.4.4), for $M = 4$, we obtain the scheme

$$h_k[m] = \begin{cases} -\frac{2^k \sqrt{-1}}{2(1 + 2^{k+1} \sqrt{-1})^2 (1 + 2^k \sqrt{-1})}, & |m| = 3 \\ \frac{(1 + 2^{k+1} \sqrt{-1} + 2^k \sqrt{-1})^2}{2(1 + 2^{k+1} \sqrt{-1})^2 (1 + 2^k \sqrt{-1})}, & |m| = 1 \\ 1, & m = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (4.4.8)$$

Note that, when $k \rightarrow \infty$, the mask h_k converges towards the stationary Deslauriers-Dubuc scheme defined by (4.3.5) which reproduces polynomials of degree up to 3.

4.4.4 Non-Stationary Minimum-Support Subdivision Scheme

The non-stationary minimum-support subdivision scheme generates exponential polynomials defined in Section 4.4.1 with the shortest mask [81]. It has a support of size $L + 1$ and is given by

$$H_k^\alpha(z) = \frac{1}{2^{L-1}} \prod_{n=1}^L \left(1 + e^{\frac{\alpha_n}{2^k}} z\right). \quad (4.4.9)$$

Example-Shortest Ellipse-Generating Scheme: We construct a non-stationary minimum-support subdivision scheme that is capable of generate ellipses. Therefore, we choose $\alpha = \left(0, \frac{2j\pi}{M}, -\frac{2j\pi}{M}\right)$. By imposing the affine invariance of Definition 4.2.1, the subdivision mask at iteration k is of size 4 and is given by $\text{sinc}^{-2}\left(\frac{1}{M}\right)H_k^\alpha(z)$, where

$$H_k^\alpha(z) = \frac{1}{4} \left(1 + \left(1 + e^{\frac{-2j\pi}{2^k M}} + e^{\frac{2j\pi}{2^k M}}\right) z + \left(1 + e^{\frac{-2j\pi}{2^k M}} + e^{\frac{2j\pi}{2^k M}}\right) z^2 + z^3\right). \quad (4.4.10)$$

4.5 Connection with Scaling Functions and Parametric Curves

For any convergent stationary subdivision scheme, there is a continuously-defined function φ associated to the mask $h_{0 \rightarrow k}$, given by (4.2.4), and applied to initial data δ [116, 117], which is the Kronecker delta. The function φ is called the basic limit function of the subdivision scheme [106] (a.k.a. fundamental function [119]) and is defined as

$$\varphi(t) \Big|_{t=\frac{m}{2^k}} = \lim_{k \rightarrow +\infty} h_{0 \rightarrow k}[m]. \quad (4.5.1)$$

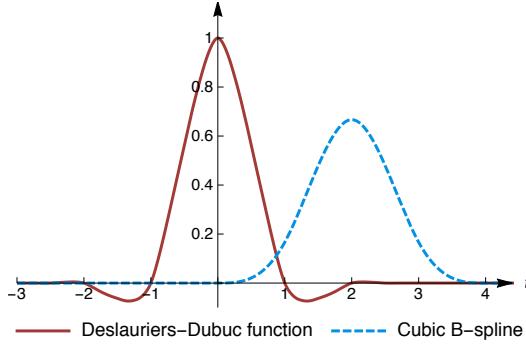


Figure 4.3: Illustration of the basic limit functions of the stationary Deslauriers-Dubuc (red solid line) and minimum-support (blue dashed line) subdivision schemes that reproduce polynomials up to degree 3.

The function φ is actually a scaling function (see Definition 3.2.1) that verifies the two-scale refinement relation [126]

$$\varphi(t) = \sum_{n \in \mathbb{Z}} h[n] \varphi(2t - n), \quad (4.5.2)$$

and recursively, we have that

$$\varphi(t) = \sum_{n \in \mathbb{Z}} h_{0 \rightarrow k}[n] \varphi(2^k t - n). \quad (4.5.3)$$

In Figure 4.3, we illustrate the basic limit functions of the Deslauriers-Dubuc and minimum-support subdivision schemes that reproduce polynomials up to degree 3 (see (4.3.5) and (4.3.6), respectively). They correspond to the Deslaurier-Dubuc interpolation function [119], given by the auto-correlation of the Daubechies scaling function [99, 109, 122], and the cubic B-spline $\beta_{(0,0,0,0)}$ (see Section 2.3.1). By combining (4.2.3), (4.2.6) and (4.5.1), for any convergent stationary subdivision scheme we write

$$\begin{aligned}
\mathbf{r}(t)\Big|_{t=\frac{m}{2^k}} &= \lim_{k \rightarrow +\infty} \mathbf{p}_{(k)}[m] \\
&= \lim_{k \rightarrow +\infty} \left(h_{0 \rightarrow k} * \mathbf{p}_{(0)} \Big|_{\uparrow_{2^k}} \right) [m] \\
&= \sum_{n \in \mathbb{Z}} \mathbf{p}_{(0)}[n] \varphi \left(\frac{m - 2^k n}{2^k} \right) \\
&= \sum_{n \in \mathbb{Z}} \mathbf{p}_{(0)}[n] \varphi(t - n),
\end{aligned} \tag{4.5.4}$$

where $m \in \mathbb{Z}$ and $t = \frac{m}{2^k} \in \mathbb{R}$ is a continuous parameter. We recover the formulation of a parametric curve. It means that in the particular case where the basis functions of parametric curves are scaling functions, there is a connection with the proposed work: the discrete filters of the scaling functions can be used as subdivision masks for stationary schemes.

4.6 Subdivision Surfaces

4.6.1 Notation and Terminology

A triangular mesh $\mathcal{M}_{(k)}$, at resolution k , is defined by the set $\mathbb{P}_{(k)} = \{\mathbf{p}_{(k)}[m] \in \mathbb{R}^3, m \in \{0, \dots, N_k - 1\}\}$ of N_k points. These points are implicitly connected by triangles. The valence ξ of the vertex $\mathbf{p}_{(k)}[m]$ denotes the number of its adjacent vertices in the mesh $\mathcal{M}_{(k)}$. A vertex is extraordinary if its valence is different from six, otherwise it is regular. A mesh $\mathcal{M}_{(k)}$ is called regular if all its vertices have valence six.

4.6.2 Subdivision Schemes

Given the coarse mesh $\mathcal{M}_{(0)}$, we apply repeatedly k times a subdivision rule to obtain a finer mesh $\mathcal{M}_{(k)}$, which does not necessarily contain the coarser mesh $\mathcal{M}_{(k-1)}$. At the limit of the process, we obtain the continuously defined surface $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$ with $\sigma_1, \sigma_2, \sigma_3 \in \mathcal{C}^0$. The subdivision rule from $(k-1)$ to k is defined by

$$\mathbb{P}_{(k)} = S_{k-1} \mathbb{P}_{(k-1)}, \tag{4.6.1}$$

where S_k is the subdivision operator at the k th iteration of the subdivision scheme. As in 2D, we say that a subdivision scheme is stationary if the subdivision rules in S_k are the same at each iteration; otherwise, it is non-stationary. The vertices of the mesh at the k th iteration ($k \geq 1$) can be directly obtained from the initial set of control points $\mathbb{P}_{(0)}$ by

$$\mathbb{P}_{(k)} = S_{k-1}S_{k-2} \cdots S_0\mathbb{P}_{(0)}. \quad (4.6.2)$$

Formulation (4.6.2) makes it obvious that $\mathbb{P}_{(k)}$ depends exclusively on the N_0 vertices of $\mathbb{P}_{(0)}$, which we call control points. The N_k vertices of the mesh $\mathcal{M}_{(k)}$ at the k th iteration ($k \geq 1$) are called subdivision points.

4.6.3 Properties

In the literature there is a wide range of subdivision schemes to produce surfaces. They differ in the properties that they confer to the limit surface.

Reproduction of specific shapes: The ability of a subdivision scheme to reproduce ellipsoids or spheres [71, 101, 105, 127].

Interpolation: At each iteration, the control points lie on the limit surface [111, 128–131].

Smoothness: The subdivision rule given in [132] leads to \mathcal{C}^1 limit surfaces, while the ones developed in [110, 133] produce \mathcal{C}^2 continuous limit surfaces everywhere except at extraordinary vertices where they are only \mathcal{C}^1 continuous. Non-stationary schemes generalizing the Doo-Sabin and Catmull-Clark schemes were proposed in [134]. Theorems to analyze the smoothness of a non-stationary scheme in regions with regular and extraordinary vertices can be found in [135] and [136], respectively.

Affine invariance: The geometry of the limit surface changes in synchrony with any affine transformation that would be applied to the initial mesh. Conditions on the subdivision operators S_k to ensure this property are given in [105, 137].

4.6.4 Examples of Subdivision Schemes

Stationary Loop's scheme

Loop's scheme is a widely used subdivision scheme for triangular meshes. It has been developed by Charles Loop in 1987 [110]. It generates a limit surface which is C^2 -continuous everywhere, except at extraordinary vertices where the regularity is C^1 . This scheme refines each triangle of a coarse mesh into four subtriangles (Figure 4.5 (a) and (b)). A subdivision iteration from k to $(k + 1)$ consists in two steps.

Vertex-Point Rule: The location of every former vertex $\mathbf{p}_{(k)}[m]$, $m \in \{0, \dots, N_k - 1\}$, is updated. A new vertex $\mathbf{p}_{(k+1)}[q]$, $q \in \{0, \dots, N_{k+1} - 1\}$, is obtained by the convex combination

$$\mathbf{p}_{(k+1)}[q] = \alpha \mathbf{p}_{(k)}[m] + \beta \sum_{\mathbf{p}_{(k)}[u] \in \mathcal{V}_m} \mathbf{p}_{(k)}[u], \quad (4.6.3)$$

where \mathcal{V}_m is the set of the ξ adjacent vertices of $\mathbf{p}_{(k)}[m]$ in $\mathcal{M}_{(k)}$, $\alpha = \frac{3}{8} + \left(\frac{3}{8} + \frac{1}{4} \cos\left(\frac{2\pi}{\xi}\right)\right)^2$ and $\beta = \frac{1-\alpha}{\xi}$. In Figure 4.4 (a) we illustrate the vertex-point stencil where the local linear combination (4.6.3) is suggested graphically.

Edge-Point Rule: For every edge in the coarser mesh, a new vertex is inserted.

Let us consider the two old adjacent triangles made of the vertices $\mathbf{p}_{(k)}[m_0]$, $\mathbf{p}_{(k)}[m_1]$, $\mathbf{p}_{(k)}[m_2]$ and $\mathbf{p}_{(k)}[m_3]$, with $m_0, m_1, m_2, m_3 \in \{0, \dots, N_k - 1\}$, such that $\mathbf{p}_{(k)}[m_0]\mathbf{p}_{(k)}[m_1]$ is the common edge (Figure 4.4 (b)). The subdivision rule yields

$$\mathbf{p}_{(k+1)}[q] = \frac{3}{8} (\mathbf{p}_{(k)}[m_0] + \mathbf{p}_{(k)}[m_1]) + \frac{1}{8} (\mathbf{p}_{(k)}[m_2] + \mathbf{p}_{(k)}[m_3]), \quad (4.6.4)$$

where $q \in \{0, \dots, N_{k+1} - 1\}$.

This scheme is stationary, easy to implement, and has the property to be affine invariant. In Figure 4.5, we illustrate the sphere approximated by Loop's scheme using an octahedron as initial mesh. Extensions of Loop's scheme to get additional properties like optimal shrinkage and exponential polynomial reproduction are considered in [71].

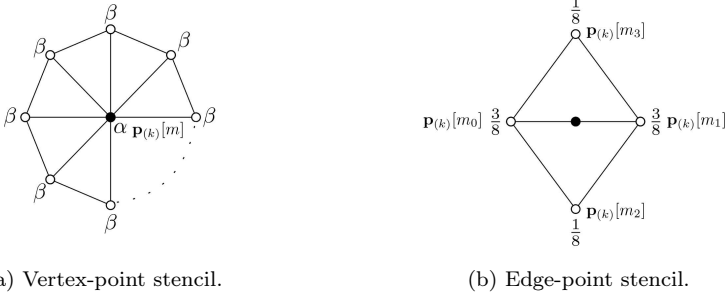


Figure 4.4: Stencils for vertex-point (a) and edge-point (b) rules of Loop's scheme.

Non Stationary BLOB scheme

The Butterfly-Loop optimal blending (BLOB) scheme is a new subdivision scheme that we introduced in [71]. It is an affine-invariant non-stationary subdivision scheme for the recursive refinement of any triangular mesh that is regular or has extraordinary vertices of valence 4. In particular, when applied to an arbitrary convex octahedron, it produces a G^1 -continuous surface with a blob-like shape as the limit of the recursive subdivision process. In case of a regular octahedron, the subdivision process provides an accurate representation of ellipsoids. Like Loop's scheme, it belongs to the class of *primal* subdivision schemes [127, 137] since, at each step of the refinement process, a finer mesh is created by splitting the faces of the coarse mesh. However, we use a level-dependent vertex-point stencil and a much larger (and also level-dependent) edge-point stencil. A subdivision iteration from k to $(k + 1)$ consists in two steps.

Vertex-Point Rules: The location of every former vertex $\mathbf{p}^{(k)}[m]$, $m \in \{0, \dots, N_k - 1\}$, is updated. A new vertex $\mathbf{p}^{(k+1)}[q]$, $q \in \{0, \dots, N_{k+1} - 1\}$, is obtained by the convex combination

$$\mathbf{p}^{(k+1)}[q] = \begin{cases} a^{(k)} \mathbf{p}^{(k)}[m] + b^{(k)} \sum_{\mathbf{p}^{(k)}[u] \in \mathcal{V}_m} \mathbf{p}^{(k)}[u], & \mathbf{p}^{(k)}[m] \text{ is regular,} \\ \tilde{a}^{(k)} \mathbf{p}^{(k)}[m] + \tilde{b}^{(k)} \sum_{\mathbf{p}^{(k)}[u] \in \mathcal{V}_m} \mathbf{p}^{(k)}[u], & \mathbf{p}^{(k)}[m] \text{ has valence four,} \end{cases} \quad (4.6.5)$$

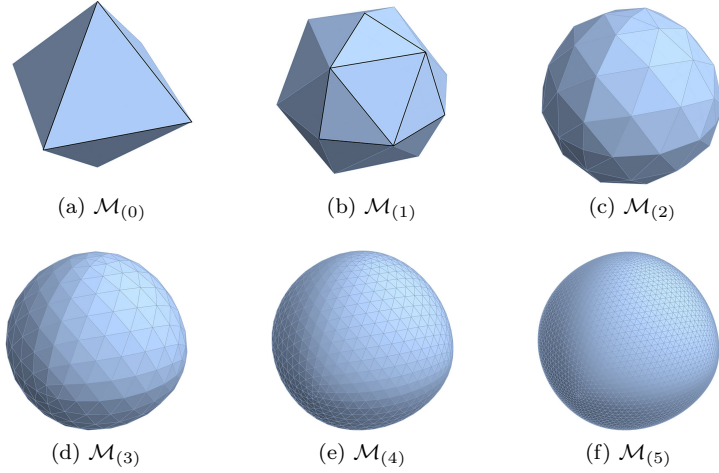


Figure 4.5: Sphere approximated by Loop's scheme, starting from an octahedron.

where \mathcal{V}_m is the set of the ξ adjacent vertices of $\mathbf{p}^{(k)}[m]$ in $\mathcal{M}^{(k)}$,

$$\begin{aligned} a^{(k)} &= \frac{4(v^{(k)})^2 + 2v^{(k)} + 1}{4(v^{(k)} + 1)^2}, & b^{(k)} &= \frac{2v^{(k)} + 1}{8(v^{(k)} + 1)^2}, \\ \tilde{a}^{(k)} &= \frac{45(v^{(k)})^2 + 18v^{(k)} + 1}{48(v^{(k)} + 1)^2}, & \tilde{b}^{(k)} &= \frac{3(v^{(k)})^2 + 78v^{(k)} + 47}{192(v^{(k)} + 1)^2}, \end{aligned} \quad (4.6.6)$$

and

$$v^{(k)} = \frac{1}{2} \left(e^{j \frac{\lambda}{2^{(k+1)}}} + e^{-j \frac{\lambda}{2^{(k+1)}}} \right), \quad \lambda \in [0, \pi) \cup j(0, 2 \operatorname{acosh}(500)). \quad (4.6.7)$$

The parameter λ influences the final shape of the limit surface. These rules are illustrated in Figures 4.6 (a) and 4.7 (a).

Edge-Point Rules: For every edge in the coarser mesh, a new vertex is inserted. We distinguish three cases: the old edge has 1) two regular endpoints; 2) two extraordinary endpoints with valence four; 3) one regular endpoint and one extraordinary endpoint with valence four. We graphically provide the

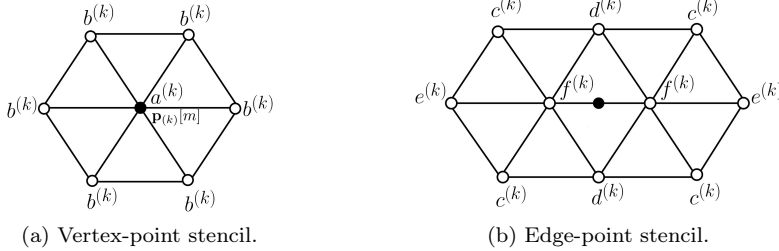


Figure 4.6: Stencils for vertex-point (a) and edge-point (b) rules of the BLOB scheme in the regular regions of the mesh.

corresponding linear combinations in Figure 4.6 (b) and Figure 4.7 (b) and (c), respectively. The involved coefficients are defined by

$$\begin{aligned}
 c^{(k)} &= \frac{2v^{(k)}+1}{16(v^{(k)}+1)^3}, & d^{(k)} &= \frac{(2v^{(k)}+1)^2}{8(v^{(k)}+1)^3}, & e^{(k)} &= \frac{1}{16(v^{(k)}+1)^3}, \\
 f^{(k)} &= \frac{(2v^{(k)}+1)(4(v^{(k)})^2+6v^{(k)}+3)}{16(v^{(k)}+1)^3}, & \tilde{d}^{(k)} &= \frac{16(v^{(k)})^2+18v^{(k)}+5}{32(v^{(k)}+1)^3}, & \tilde{e}^{(k)} &= \frac{2v^{(k)}+5}{64(v^{(k)}+1)^3}, \\
 \tilde{f}^{(k)} &= \frac{32(v^{(k)})^3+64(v^{(k)})^2+54v^{(k)}+15}{64(v^{(k)}+1)^3}, & g^{(k)} &= \frac{(2v^{(k)}+3)(2v^{(k)}+1)}{8(v^{(k)}+1)^2}, & h^{(k)} &= \frac{1}{8(v^{(k)}+1)^2},
 \end{aligned}
 \tag{4.6.8}$$

where $v^{(k)}$ is defined in (4.6.7).

The vertex-point rules have the same size and structure as the one of Loop's subdivision scheme [110], and the size and structure of the edge-point rules are the same as those of the modified Butterfly scheme [128]. This is why we named this scheme the Butterfly-Loop optimal blending subdivision scheme. In Figure 4.8, we illustrate the sphere approximated by the BLOB scheme using an octahedron as initial mesh. This scheme is non-stationary, so more challenging to implement than Loop's scheme. However, it provides a better approximation of the sphere than Loop's scheme while starting from a coarse mesh with few control points [71]. In a segmentation context, this suggests that the BLOB scheme will be more accurate when delineating spherical shapes with a small number of control points. As more control points are included, the two schemes will produce outcomes of equivalent quality [71].

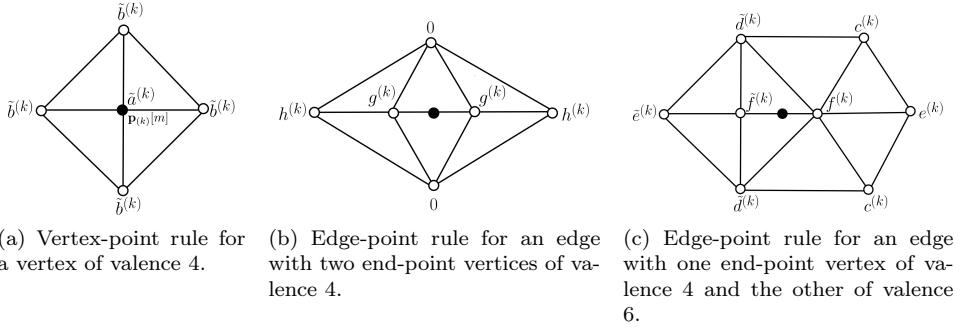


Figure 4.7: Stencils for vertex- and edge-point rules of the BLOB scheme involving extraordinary vertices of valence four.

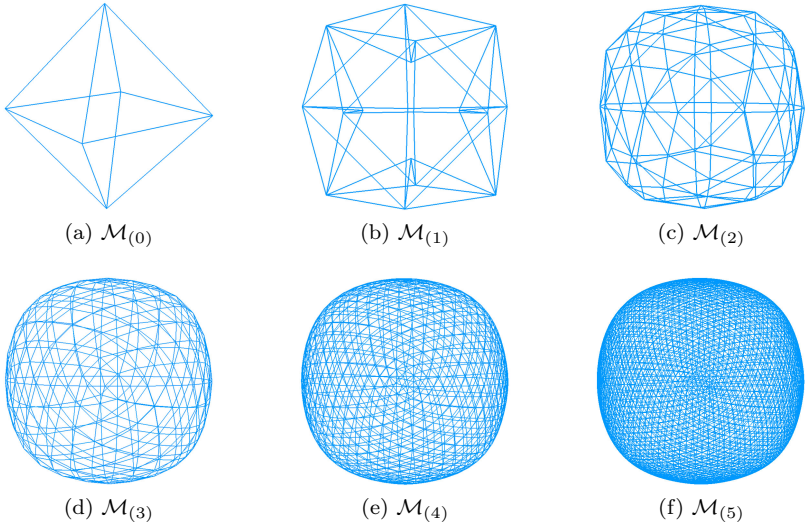


Figure 4.8: Sphere approximated by the BLOB scheme, starting from an octahedron.

4.7 Appendices

4.7.1 Derivation of Equation (4.2.3)

Using (4.2.1), we have that

$$\mathbf{p}_{(1)}[m] = \sum_{n \in \mathbb{Z}} h[m - 2n] \mathbf{p}_{(0)}[n] \quad (4.7.1)$$

and

$$\begin{aligned} \mathbf{p}_{(2)}[m] &= \sum_{n \in \mathbb{Z}} h[m - 2n] \mathbf{p}_{(1)}[n] \\ &= \sum_{n \in \mathbb{Z}} h[m - 2n] \sum_{q \in \mathbb{Z}} h[n - 2q] \mathbf{p}_{(0)}[q] \\ &= \sum_{q \in \mathbb{Z}} \left(\sum_{l \in \mathbb{Z}} h[m - 4q - 2l] h_{\uparrow_2}[2l] \right) \mathbf{p}_{(0)}[q] \\ &= \sum_{q \in \mathbb{Z}} \underbrace{(h * h_{\uparrow_2})}_{h_{0 \rightarrow 2}}[m - 4q] \mathbf{p}_{(0)}[q]. \end{aligned} \quad (4.7.2)$$

Combining (4.2.1), (4.7.1), and (4.7.2), we recursively obtain

$$\mathbf{p}_{(k)}[m] = \sum_{n \in \mathbb{Z}} h_{0 \rightarrow k}[m - 2^k n] \mathbf{p}_{(0)}[n], \quad (4.7.3)$$

where $h_{0 \rightarrow k}$ is given by (4.2.4). ■

4.7.2 Derivation of Equation (4.2.7)

$$\begin{aligned} \dot{\mathbf{r}}(t) \Big|_{t=\frac{m}{2^k}} &= \lim_{\epsilon \rightarrow 0} \frac{\mathbf{r}(t + \epsilon) - \mathbf{r}(t)}{\epsilon} \Big|_{t=\frac{m}{2^k}} \\ &= \lim_{k \rightarrow \infty} 2^k \left(\mathbf{r}\left(t + \frac{1}{2^k}\right) - \mathbf{r}(t) \right) \Big|_{t=\frac{m}{2^k}}, \end{aligned} \quad (4.7.4)$$

where we used $\epsilon = \frac{1}{2^k}$. Combining this result with (4.2.6), we obtain

$$\dot{\mathbf{r}}(t)\Big|_{t=\frac{m}{2^k}} = \lim_{k \rightarrow +\infty} 2^k (\mathbf{p}_{(k)}[m+1] - \mathbf{p}_{(k)}[m]), \quad (4.7.5)$$

which ends the proof. ■

4.7.3 Proof of Proposition 4.2.2

For any convergent subdivision scheme, we have the necessary condition (4.2.5). Under this condition, it follows directly from (4.2.1) that constant functions are reproduced [115]. It means that if $\mathbf{p}_{(0)}[m] = 1, \forall m$, then we have that, for a fixed k , $\mathbf{p}_{(k)}[m] = 1, \forall m$. We rewrite (4.2.3) for this particular case and we obtain that

$$\sum_{n \in \mathbb{Z}} h_{0 \rightarrow k}[m - 2^k n] = 1, \quad \forall m \in \mathbb{Z}. \quad (4.7.6)$$

Let \mathbf{A} be a (2×2) matrix and $\mathbf{b} \in \mathbb{R}^2$ be a translation vector. We calculate

$$(h_{0 \rightarrow k} * (\mathbf{A}\mathbf{p}_{(0)} + \mathbf{b})_{\uparrow_{2^k}})[m] = \left(\mathbf{A} \left(h_{0 \rightarrow k} * \mathbf{p}_{(0)\uparrow_{2^k}} \right) \right) [m] + \mathbf{b} \sum_{n \in \mathbb{Z}} h_{0 \rightarrow k}[m - 2^k n]. \quad (4.7.7)$$

We use (4.2.3) and (4.7.6) in (4.7.7) to obtain

$$(h_{0 \rightarrow k} * (\mathbf{A}\mathbf{p}_{(0)} + \mathbf{b})_{\uparrow_{2^k}})[m] = \mathbf{A}\mathbf{p}_{(k)}[m] + \mathbf{b}. \quad (4.7.8)$$

For $k \rightarrow +\infty$ in (4.7.8), we obtain

$$\lim_{k \rightarrow \infty} (h_{0 \rightarrow k} * (\mathbf{A}\mathbf{p}_{(0)} + \mathbf{b})_{\uparrow_{2^k}})[m] = \mathbf{A}\mathbf{r}(t)\Big|_{t=\frac{m}{2^k}} + \mathbf{b}, \quad (4.7.9)$$

which corresponds to the condition of affine invariance. ■

Chapter 5

Image Energies

In the previous chapters, we introduced two new representation models for curves/surfaces. In Table 5.1, we summarize the geometric representations and their parameters Θ that we have seen so far. We now focus on the second main aspect of active contours/surfaces, namely the snake energy. We recall that the energy is a function of Θ that is minimized during the optimization process by adapting the values of Θ , *i.e.*,

$$\Theta_{\text{opt}} = \arg \min_{\Theta} E_{\text{snake}}(\Theta). \quad (5.0.1)$$

In particular, we focus on the image energy term (see (1.2.1) and Section 1.2.2). The choice of this term is crucial as it drives the evolution of the snake contour to fit image data, and thus determines the quality of the segmentation. It is generally tuned according to the features of a given segmentation task.

In this chapter, we collect and unify all the image energies that were proposed in our publications [68, 70, 73, 74] and work [72]. These have been carried out in collaboration with A. Depeursinge, A. Galan, L. Romani, D. Sage, D. Schmitter, V. Uhlmann and M. Unser. The chapter is organized as follows: In Sections 5.1 and 5.2, we first adapt the standard edge- and region-based energies, initially defined for parametric snakes in Section 2.2, to our new representation models (see Chapters 3 and 4). Then, in Sections 5.3 and 5.4, we identify some limitations of these energies and propose new ones, namely ridge- and texture-based energies, that overcomes

Table 5.1: Summary of the representation models and their parameters Θ .

Model	Parameters	Relation
Parametrization	$\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}}$	$\mathbf{r} := \mathbf{r}(\Theta)$ (2.1.2) $\boldsymbol{\sigma} := \boldsymbol{\sigma}(\Theta)$ (2.1.6)
Locally refinable parametrization	$\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}, m \neq p}$ $\cup \{\tilde{\mathbf{c}}_p[n]\}_{n \in \{n_0, \dots, n_0+N-1\}}$	$\mathbf{r} := \mathbf{r}(\Theta)$ (3.3.2) $\boldsymbol{\sigma} := \boldsymbol{\sigma}(\Theta)$ (3.3.4)
Subdivision	$\Theta = \{\mathbf{p}_{(0)}[m]\}_{m \in \{0, \dots, M-1\}}$ <small>with $M = N_0$ in 3D</small>	$\mathbf{p}_{(k)} := \mathbf{p}_{(k)}(\Theta)$ (4.2.3) $\mathbb{P}_{(k)} := \mathbb{P}_{(k)}(\Theta)$ (4.6.2)

these problems.

We strongly recommend the reader to refer to Table 5.1 throughout this chapter.

5.1 Edge-based Energy

By definition an edge is a border between areas of high and low gray value. The energy term (2.2.2) is an efficient edge-based energy proposed by [44] that we briefly recall as

$$E_{\text{edge}}(\Theta) = - \oint_{\mathcal{C}} \langle \nabla f(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle d\mathbf{r}. \quad (5.1.1)$$

It relies on edge maps derived from the image and incorporates information about the directionality of the snake curve. This allows the snake to discriminate on which side of an object it is located (*e.g.*, inside or outside an object).

5.1.1 Derivation for the Locally Refinable Parametrization

The energy term $E_{\text{edge}}(\Theta)$ has the advantage of being independent of the parametrization. It can thus be applied to our new parametrization (3.3.2), that is to a parametric closed curve that has been locally refined with respect to the control point

$\mathbf{c}[p]$. In Proposition 5.1.1, we derive¹ the explicit formula of $E_{\text{edge}}(\Theta)$ associated to locally refinable curves in which the control points $\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}, m \neq p}$ $\cup \{\tilde{\mathbf{c}}_p[n]\}_{n \in \{n_0, \dots, n_0+N-1\}}$ are highlighted.

Proposition 5.1.1. *Let \mathbf{r} be a locally refinable curve as described in Proposition 3.3.1. Then, the edge-based energy term $E_{\text{edge}}(\Theta)$ can be expressed as*

$$E_{\text{edgeLR}}(\Theta) = \sum_{\substack{m=0 \\ m \neq p}}^{M-1} c_2[m] \int_0^M G(\mathbf{r}(t)) \dot{\varphi}_M(t-m) dt \\ + \rho \sum_{n=n_0}^{n_0+N-1} \tilde{c}_{p,2}[n] \int_0^M G(\mathbf{r}(t)) \dot{\varphi}_{\rho M}(\rho t - \rho p - n) dt, \quad (5.1.2)$$

where ρ is the refinement factor, $\{n_0, \dots, n_0+N\}$ is the support of the refinement filter, $\mathbf{r} := \mathbf{r}(\Theta)$ and $\tilde{\mathbf{c}}_p = (\tilde{c}_{p,1}, \tilde{c}_{p,2})$ are given by (3.3.2) and (3.3.3), respectively, and $\dot{\varphi}(t) = \frac{d\varphi(t)}{dt}$. The image G is given by

$$G(x_1, x_2) = - \int_{-\infty}^{x_1} \Delta f(\tau, x_2) d\tau, \quad (5.1.3)$$

where Δf is the Laplacian of the image f .

The proof of Proposition 5.1.1 is given in Appendix 5.6.1.

5.1.2 Discretization for the Subdivision-based Representation

The subdivision-based representation presented in Chapter 4 is discrete by nature. We derive² the corresponding discrete edge-based energy term as

$$E_{\text{edgeSD}}(\mathbf{p}_{(k)}(\Theta)) = - \frac{1}{2^k} \sum_{m=0}^{2^k M-1} \langle \nabla f(\mathbf{p}_{(k)}[m]), \mathbf{n}(\mathbf{p}_{(k)}[m]) \rangle, \quad (5.1.4)$$

¹This work is based on our publication [68], in collaboration with D. Schmitter and M. Unser.

²This work is based on our publication [70], in collaboration with D. Schmitter, V. Uhlmann and M. Unser.

where $\mathbf{p}_{(k)}[m]$ is the location of the m -th subdivision point defined by (4.2.3), and $\nabla f(\mathbf{p}_{(k)}[m])$ and $\mathbf{n}(\mathbf{p}_{(k)}[m])$ are the within-plane gradient of the image f and the approximation of the unit normal vector, respectively, at $\mathbf{p}_{(k)}[m]$. The vector $\mathbf{n}(\mathbf{p}_{(k)}[m])$ is defined by

$$\mathbf{n}(\mathbf{p}_{(k)}[m]) = \begin{pmatrix} n_1(\mathbf{p}_{(k)}[m]) \\ n_2(\mathbf{p}_{(k)}[m]) \end{pmatrix} = \frac{1}{b_1} \begin{pmatrix} 2^k(p_{2(k)}[m+1] - p_{2(k)}[m]) \\ -2^k(p_{1(k)}[m+1] - p_{1(k)}[m]) \end{pmatrix} \quad (5.1.5)$$

and converges to

$$\mathbf{n}(\mathbf{p}_{(k)}[m]) \xrightarrow[k \rightarrow \infty]{} \mathbf{n}(\mathbf{r}(t)|_{t=\frac{m}{2^k}}) = \frac{1}{b_2} \begin{pmatrix} \dot{r}_2(t)|_{t=\frac{m}{2^k}} \\ -\dot{r}_1(t)|_{t=\frac{m}{2^k}} \end{pmatrix}, \quad (5.1.6)$$

where $\mathbf{n}(\mathbf{r})$ is the unit vector normal to the curve \mathbf{r} and, b_1 and b_2 are constants such that $\|\mathbf{n}(\mathbf{p}_{(k)})\| = 1$ and $\|\mathbf{n}(\mathbf{r})\| = 1$.

Proposition 5.1.2. *The energy defined by (5.1.4) converges to the standard energy $E_{\text{edge}}(\Theta)$, i.e.,*

$$E_{\text{edgeSD}}(\mathbf{p}_{(k)}(\Theta)) \xrightarrow[k \rightarrow \infty]{} E_{\text{edge}}(\Theta) = - \oint_{\mathcal{C}} \langle \nabla f(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle d\mathbf{r}. \quad (5.1.7)$$

The proof of Proposition 5.1.2 is given in Appendix 5.6.2. This proposition guarantees the efficiency of our discrete formulation (5.1.4) as it has the proper limit commonly used in the continuous case.

Extension to the surface energy

Following the notations of Section 4.6, we derive³ the discrete formulation for subdivision surfaces of the gradient energy (2.2.3). The term is expressed as

$$E_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta)) = -\frac{1}{2^{2k}} \sum_{m=0}^{N_k-1} \langle \nabla f(\mathbf{p}_{(k)}[m]), \mathbf{n}(\mathbf{p}_{(k)}[m]) \rangle, \quad (5.1.8)$$

where $\nabla f(\mathbf{p}_{(k)}[m])$ and $\mathbf{n}(\mathbf{p}_{(k)}[m])$ are the gradient of f and the approximation of the unit normal vector, respectively, at the vertex $\mathbf{p}_{(k)}[m]$. The expression of

³This work is based on our work [72], in collaboration with L. Romani and M. Unser.

$\mathbf{n}(\mathbf{p}_{(k)}[m])$ is given by

$$\mathbf{n}(\mathbf{p}_{(k)}[m]) = \frac{\sum_{T \in \mathcal{O}_m} \mathbf{n}_t(T)}{\|\sum_{T \in \mathcal{O}_m} \mathbf{n}_t(T)\|}, \quad (5.1.9)$$

where \mathcal{O}_m is the set of all the triangles to which $\mathbf{p}_{(k)}[m]$ belongs and $\mathbf{n}_t(T)$ is the normal of the triangle T .

Proposition 5.1.3. *The energy given by (5.1.8) converges to the standard energy $E_{\text{grad}}(\Theta)$, i.e.,*

$$E_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta)) \xrightarrow[k \rightarrow \infty]{} E_{\text{grad}}(\Theta) = - \iint_{\mathcal{S}} \langle \nabla f(\boldsymbol{\sigma}), d\boldsymbol{\sigma} \rangle, \quad (5.1.10)$$

where \mathcal{S} is the surface described by $\boldsymbol{\sigma}$ and $d\boldsymbol{\sigma}$ represents the vector differential area.

The proof of Proposition 5.1.3 is given in Appendix 5.6.4. This proposition justifies our discretization (5.1.8) for the gradient energy.

5.2 Region-based Energy using Intensity Distribution

The region-based energy (2.2.4) allows to distinguish between homogeneous regions in the image using intensity information. We recall its definition as

$$E_{\text{intensity}}(\Theta) = - \frac{1}{|\Sigma|} \left| \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_{\lambda} \setminus \Omega} f(\mathbf{x}) dx_1 dx_2 \right|. \quad (5.2.1)$$

It maximizes the contrast between the mean intensity over the surface Ω enclosed by \mathbf{r} and the mean intensity over the shell $\Omega_{\lambda} \setminus \Omega$. The surface Ω_{λ} is built from $\mathbf{r}_{\lambda} = (r_{\lambda,1}, r_{\lambda,2})$, a dilated version of \mathbf{r} , such that $\Omega \subset \Omega_{\lambda}$, and $\Omega_{\lambda} \setminus \Omega$ and Ω have the same area. This last criteria is to enforce $E_{\text{intensity}}(\Theta) = 0$ when f takes a constant value, for instance in flat regions of the image. Note that the energy (5.2.1) implicitly supposes that \mathbf{r} does not self-intersect.

5.2.1 Derivation for the Locally Refinable Parametrization

The energy term $E_{\text{intensity}}(\Theta)$ is independent of the parametrization. We explicit⁴ the formula in case of locally refinable closed curves, where the refinement has been done with respect to the control point $\mathbf{c}[p]$.

Proposition 5.2.1. *If \mathbf{r} is locally refinable in the sense of Proposition 3.3.1, then the region-based energy $E_{\text{intensity}}(\Theta)$ can be expressed as*

$$\begin{aligned}
 E_{\text{intensityLR}}(\Theta) = -\frac{1}{|\Sigma|} & \left| 2 \sum_{\substack{m=0 \\ m \neq p}}^{M-1} c_2[m] \int_0^M F(\mathbf{r}(t)) \dot{\varphi}_M(t-m) dt \right. \\
 & + 2\rho \sum_{n=n_0}^{n_0+N-1} \tilde{c}_{p,2}[n] \int_0^M F(\mathbf{r}(t)) \dot{\varphi}_{\rho M}(\rho t - \rho p - n) dt \\
 & - \sum_{\substack{m=0 \\ m \neq p}}^{M-1} c_{\lambda_2}[m] \int_0^M F(\mathbf{r}_\lambda(t)) \dot{\varphi}_M(t-m) dt \\
 & \left. - \rho \sum_{n=n_0}^{n_0+N-1} \tilde{c}_{\lambda p,2}[n] \int_0^M F(\mathbf{r}_\lambda(t)) \dot{\varphi}_{\rho M}(\rho t - \rho p - n) dt \right|, \tag{5.2.2}
 \end{aligned}$$

where ρ is the refinement factor, $\tilde{\mathbf{c}}_p = (\tilde{c}_{p,1}, \tilde{c}_{p,2})$ is given by (3.3.3), $\{n_0, \dots, n_0 + N\}$ is the support of the refinement filter and F is the pre-integrated image along the first dimension defined by

$$F(x_1, x_2) = \int_{-\infty}^{x_1} f(\tau, x_2) d\tau. \tag{5.2.3}$$

The signed area $\Sigma := \Sigma(\Theta)$ enclosed by \mathbf{r} is given by

⁴This work is based on our publication [68], in collaboration with D. Schmitter and M. Unser.

$$\begin{aligned}
\Sigma(\Theta) = & - \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \sum_{\substack{n=0 \\ n \neq p}}^{M-1} c_2[m]c_1[n] \int_0^M \varphi_M(t-m)\dot{\varphi}_M(t-n)dt \\
& - \rho \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \sum_{n=n_0}^{n_0+N-1} c_2[m]\tilde{c}_{p,1}[n] \int_0^M \varphi_M(t-m)\dot{\varphi}_{\rho M}(\rho t - \rho p - n)dt \\
& - \sum_{m=n_0}^{n_0+N-1} \sum_{\substack{n=0 \\ n \neq p}}^{M-1} \tilde{c}_{p,2}[m]c_1[n] \int_0^M \varphi_{\rho M}(\rho t - \rho p - m)\dot{\varphi}_M(t-n)dt \\
& - \rho \sum_{m=n_0}^{n_0+N-1} \sum_{n=n_0}^{n_0+N-1} \tilde{c}_{p,2}[m]\tilde{c}_{p,1}[n] \int_0^M \varphi_{\rho M}(\rho t - \rho p - m)\dot{\varphi}_{\rho M}(\rho t - \rho p - n)dt.
\end{aligned} \tag{5.2.4}$$

The proof is given in Appendix 5.6.5.

5.2.2 Discretization for the Subdivision-based Representation

The discrete region-based energy that we propose⁵ for subdivision curves is expressed as

$$\begin{aligned}
E_{\text{intensitySD}}(\mathbf{p}_{(k)}(\Theta)) = & - \frac{1}{2^k |\Sigma(\mathbf{p}_{(k)})|} \left| 2 \sum_{m=0}^{2^k M-1} F(\mathbf{p}_{(k)}[m])n_1(\mathbf{p}_{(k)}[m]) \right. \\
& \left. - \sum_{m=0}^{2^k M-1} F(\mathbf{p}_{\lambda(k)}[m])n_1(\mathbf{p}_{\lambda(k)}[m]) \right|, \tag{5.2.5}
\end{aligned}$$

where $\mathbf{p}_{\lambda(k)}$ is the sequence of subdivision points that describes the curve \mathbf{r}_λ , n_1 is the first coordinate of the approximation of the unit normal vector given by (5.1.5),

⁵This work is based on our publication [70], in collaboration with D. Schmitter, V. Uhlmann and M. Unser.

and F is defined by (5.2.3). We define the signed area $\Sigma(\mathbf{p}_{(k)}) := \Sigma(\mathbf{p}_{(k)}(\Theta))$ as

$$\Sigma(\mathbf{p}_{(k)}(\Theta)) = \frac{1}{2^k} \sum_{m=0}^{2^k M-1} p_{1(k)}[m] n_1(\mathbf{p}_{(k)}[m]). \quad (5.2.6)$$

Proposition 5.2.2. *The area (5.2.6) and the energy defined by (5.2.5) converges to*

$$\Sigma(\mathbf{p}_{(k)}(\Theta)) \xrightarrow[k \rightarrow \infty]{} \Sigma(\Theta) = \iint_{\Omega} dx_1 dx_2, \quad (5.2.7)$$

and

$$E_{\text{intensitySD}}(\mathbf{p}_{(k)}(\Theta)) \xrightarrow[k \rightarrow \infty]{} E_{\text{intensity}}(\Theta) = -\frac{1}{|\Sigma|} \left| \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_{\lambda} \setminus \Omega} f(\mathbf{x}) dx_1 dx_2 \right|, \quad (5.2.8)$$

where $\Sigma := \Sigma(\Theta)$ is the signed area enclosed by the contour \mathbf{r} and, Ω and Ω_{λ} are the surfaces enclosed by the curve \mathbf{r} and \mathbf{r}_{λ} , respectively.

The proof of Proposition 5.2.2 is given in Appendix 5.6.6. This proposition justifies our discrete formulation (5.2.5) of $E_{\text{intensity}}(\Theta)$.

Extension to volume energy

We consider a continuously defined, orientable, closed limit surface σ of a subdivision scheme that does not self-intersect. Following the notations of Section 4.6, for the 3D extension of (5.2.5) we propose⁶

$$E_{\text{intensitySD}}(\mathbb{P}_{(k)}(\Theta)) = -\frac{1}{|\mathbb{V}(\mathbb{P}_{(k)})| 2^{2k}} \left| 2 \sum_{m=0}^{N_k-1} F(\mathbf{p}_{(k)}[m]) n_1(\mathbf{p}_{(k)}[m]) - \sum_{m=0}^{N_k-1} F(\mathbf{p}_{\lambda(k)}[m]) n_1(\mathbf{p}_{\lambda(k)}[m]) \right|, \quad (5.2.9)$$

⁶This work is based on our work [72], in collaboration with L. Romani and M. Unser.

where $\mathbb{P}_{\lambda(k)}$ is the set of subdivision points that defines the surface σ_λ , n_1 is the first coordinate of the approximation of the unit normal vector given by (5.1.9), and F is the pre-integrated volumetric image along the first dimension defined by

$$F(x_1, x_2, x_3) = \int_{-\infty}^{x_1} f(\tau, x_2, x_3) d\tau. \quad (5.2.10)$$

The quantity $V(\mathbb{P}_{(k)})$ is the signed volume of the triangular mesh $\mathcal{M}_{(k)}$ that defines σ . To obtain this volume, we decompose the mesh in tetrahedrons, each one being composed of the center of gravity and a triangle of the mesh. We then sum the volume of each tetrahedron to obtain $V(\mathbb{P}_{(k)})$.

Proposition 5.2.3. *The volume $V(\mathbb{P}_{(k)})$ and the energy (5.2.9) converges to*

$$V(\mathbb{P}_{(k)})(\Theta) \xrightarrow[k \rightarrow \infty]{} V(\Theta) = \iiint_{\mathcal{V}} dx_1 dx_2 dx_3, \quad (5.2.11)$$

and

$$E_{\text{intensitySD}}(\mathbb{P}_{(k)})(\Theta) \xrightarrow[k \rightarrow \infty]{} E_{\text{intensity}}(\Theta) = -\frac{1}{|V|} \left| \iiint_{\mathcal{V}} f(\mathbf{x}) dx_1 dx_2 dx_3 - \iiint_{\mathcal{V}_\lambda \setminus \mathcal{V}} f(\mathbf{x}) dx_1 dx_2 dx_3 \right|, \quad (5.2.12)$$

where \mathcal{V} and \mathcal{V}_λ are the volume enclosed by σ and its dilated version σ_λ , respectively, $V := V(\Theta)$ is the signed volume of \mathcal{V} , and $dx_1 dx_2 dx_3$ is an infinitesimal vector element of \mathcal{V} .

The proof of Proposition 5.2.3 is given in Appendix 5.6.7. This proposition justifies our discrete formulation (5.2.9) as it converges to the standard energy (2.2.7).

5.3 Ridge-based Energy

In some application, the structures of interest are delimited by ridges, *i.e.*, thin lines darker or brighter than their neighborhood. If we perform an edge detection on ridge areas we will obtain a double line, one from each side of the ridge (Figure 5.1).

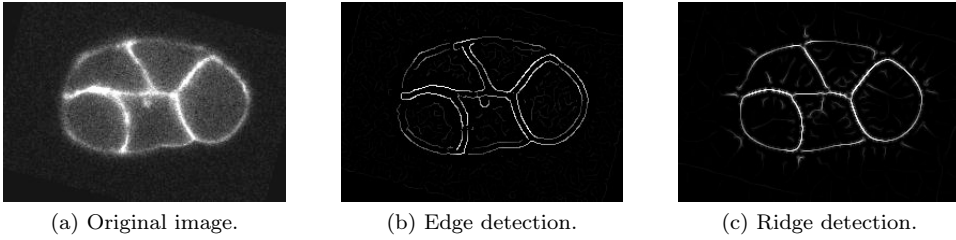


Figure 5.1: Illustration of edge and ridge detection. Source: R. Jankele and P. Gönczy, EPFL.

Edge-based energies, such as (2.2.2), are thus not well suited for these kind of applications. In this section, we propose⁷ an oriented ridge-based energy. We provide its continuous and discrete formulations.

To detect ridges, a common approach is to compute the Hessian matrix \mathbf{U} at location $\mathbf{x} = (x_1, x_2)$ as

$$\mathbf{U}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 u(x_1, x_2)}{\partial x_1^2} & \frac{\partial^2 u(x_1, x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 u(x_1, x_2)}{\partial x_2 \partial x_1} & \frac{\partial^2 u(x_1, x_2)}{\partial x_2^2} \end{pmatrix}, \quad (5.3.1)$$

where $u(x_1, x_2) = f(x_1, x_2) * (g(x_1)g(x_2))$ with f the input image and $g(x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i)^2}{2\sigma^2}}$ for $i = 1, 2$. High values of the standard deviation $\sigma \in \mathbb{R}$ increase the basin of attraction of the active contour. We define the ridge strength at location \mathbf{x} by

$$\xi(\mathbf{x}) = \sqrt{|\lambda_{\min}(\mathbf{x})|} \sqrt{|\lambda_{\min}(\mathbf{x}) - \lambda_{\max}(\mathbf{x})|}, \quad (5.3.2)$$

where $\lambda_{\min}(\mathbf{x})$ and $\lambda_{\max}(\mathbf{x})$ are the minimum and maximum eigenvalues of $\mathbf{U}(\mathbf{x})$, respectively. On the ridge, the ridge strength is maximum and the eigenvector $\mathbf{v}_{\min}(\mathbf{x})$ is normal to the ridge.

⁷This work is based on our publication [73], in collaboration with D. Sage, A. Galan and M. Unser.

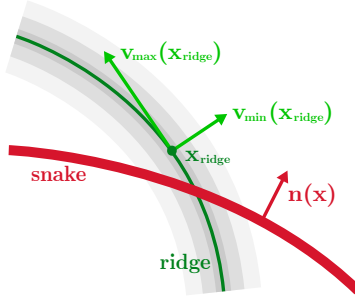


Figure 5.2: Ridge feature. When minimizing the ridge-based energy (5.3.4), the snake (red line) tends to align with the center of the ridge (green line).

The oriented ridge-based energy that we propose is

$$E_{\text{ridge}}(\Theta) = - \oint_{\mathcal{C}} \xi(\mathbf{r}) \frac{|\langle \mathbf{v}_{\min}(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle|}{\|\mathbf{v}_{\min}(\mathbf{r})\|} dr, \quad (5.3.3)$$

where $\mathbf{n}(\mathbf{r})$ is the unit vector normal to the curve \mathbf{r} and dr is an infinitesimal vector element of \mathcal{C} . The energy (5.3.3) is minimal when the vectors \mathbf{v}_{\min} and \mathbf{n} are aligned and when the snake lies on the ridge (Figure 5.2). This energy term is independent from the parametrization so it can be applied either to parametric or locally refinable parametric curves defined by (2.1.2) and (3.3.2), respectively. The discretization of (5.3.3) for subdivision curves is given by

$$E_{\text{ridgeSD}}(\mathbf{p}_{(k)}(\Theta)) = - \frac{1}{2^k} \sum_{m=0}^{2^k M - 1} \xi(\mathbf{p}_{(k)}[m]) \frac{|\langle \mathbf{v}_{\min}(\mathbf{p}_{(k)}[m]), \mathbf{n}(\mathbf{p}_{(k)}[m]) \rangle|}{\|\mathbf{v}_{\min}(\mathbf{p}_{(k)}[m])\|}, \quad (5.3.4)$$

where $\mathbf{p}_{(k)}$ is defined by (4.2.3), and $\mathbf{n}(\mathbf{p}_{(k)}[m])$, $\xi(\mathbf{p}_{(k)}[m])$ and $\mathbf{v}_{\min}(\mathbf{p}_{(k)}[m])$ are the approximations of the unit normal vector, the ridge strength and the eigenvector, respectively, at the m th subdivision point. The vector $\mathbf{n}(\mathbf{p}_{(k)}[m])$ is given by (5.1.5) and we have that

$$\mathbf{n}(\mathbf{p}_{(k)}[m]) \xrightarrow[k \rightarrow \infty]{} \mathbf{n}(\mathbf{r}(t)) \Big|_{t=\frac{m}{2^k}}, \quad (5.3.5)$$

$$\xi(\mathbf{p}_{(k)}[m]) \xrightarrow[k \rightarrow \infty]{} \xi(\mathbf{r}(t)) \Big|_{t=\frac{m}{2^k}}, \quad (5.3.6)$$

and

$$\mathbf{v}_{\min}(\mathbf{p}^{(k)}[m]) \xrightarrow[k \rightarrow \infty]{} \mathbf{v}_{\min}(\mathbf{r}(t)) \Big|_{t=\frac{m}{2^k}}. \quad (5.3.7)$$

Our discretization(5.3.4) is justified by the following proposition.

Proposition 5.3.1. *The discrete energy $E_{\text{ridgeSD}}(\mathbf{p}^{(k)}(\Theta))$ converges to the continuous one $E_{\text{ridge}}(\Theta)$, i.e.,*

$$E_{\text{ridgeSD}}(\mathbf{p}^{(k)}(\Theta)) \xrightarrow[k \rightarrow \infty]{} E_{\text{ridge}}(\Theta) = - \oint_{\mathcal{C}} \xi(\mathbf{r}) \frac{|\langle \mathbf{v}_{\min}(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle|}{\|\mathbf{v}_{\min}(\mathbf{r})\|} d\mathbf{r}. \quad (5.3.8)$$

Given (4.2.6), (5.3.5), (5.3.6) and (5.3.7), the proof of Proposition 5.3.1 is straightforward.

5.4 Texture-based Energy

Most often, structures cannot be fully characterized from their internal distribution of pixel values. Therefore, segmentation methods based on image intensity alone, such as the energy term (2.2.4), do not perform well on images where the contrast between the object of interest and the background is low [44, 74]. The incorporation of texture information is one complementary way to account for the spatial organization of the pixels inside the desired object [138–140]. It allows one to capture the morphological structure of a tissue [141].

We propose⁸ a new energy term that combines image intensity and texture information. The method is developed for 2-dimensional images and is valid for any 2D filter-based texture feature extraction method, including Gabor filters [142] or circular harmonic wavelets (CHW) [143]. The optimal balance between intensity and texture is learned using Fisher’s linear discriminant analysis (LDA).

5.4.1 Texture Analysis with Filters

We use a set of N filters $\phi_n : \mathbb{R}^2 \rightarrow \mathbb{R}$ to extract texture properties at a given position of the input 2-dimensional image f . We create a sequence $\{f_n\}_{n \in \{0, \dots, N\}}$

⁸This work is based on our publication [74], in collaboration with A. Depeursinge and M. Unser.

of $1 + N$ intensity and texture channels defined by

$$f_n(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & n = 0 \\ |(f * \phi_n)(\mathbf{x})|, & n \neq 0, \end{cases} \quad (5.4.1)$$

where $\mathbf{x} = (x_1, x_2)$ is a coordinate position and ϕ_n is a filter. For better clarity, we present the formulation for gray images but the extension to color images is straightforward: For a color image in red-green-blue (RGB) representation, we instead compute the response maps of the red, green, and blue image components. In this case, we have a sequence of $3(1 + N)$ channels $\{f_n\}_{n \in \{0, \dots, 3(1+N)-1\}}$.

The proposed method is valid for any collection of filters $\{\phi_n\}_{n \in \{1, \dots, N\}}$ extracting texture information. Here after we describe two state-of-the-art filters: CHWs and Gabor filters.

Circular Harmonic Wavelets

CHWs provide an estimation of the local organization of image directions (LOID) in a rotation-invariant fashion and at a low computational price. The LOID was found to be a fundamental property of structures found in *e.g.* biomedical tissue [144]. It allows one to linearly characterize the local circular frequencies, which are at the origin of the success of texture approaches based on local binary patterns [145].

In (5.4.1) let $\phi_n = \phi_{(p,q)}$ be the CHWs of harmonic index $p = 0, \dots, P - 1$ and scale $q = 1, \dots, Q$ for $n = 1, \dots, N$. The $N = P \cdot Q$ positive response maps $|f * \phi_n|$ characterize local circular frequencies in f up to a maximum harmonic order $(P - 1)$ and scale Q [143]. They are also locally rotation invariant [146]. The CHWs are defined in the Fourier domain indexed with polar coordinates (ω, θ) as

$$\hat{\phi}_{(p,q)}(\omega, \theta) = 2^q \hat{h}(2^q \omega) \cdot e^{jp\theta}. \quad (5.4.2)$$

There, \hat{h} is a purely radial function that controls the scale profile of the wavelet. We use Simoncelli's radial wavelet for \hat{h} , which is expressed by

$$\hat{h}(\omega) = \begin{cases} \cos\left(\frac{\pi}{2} \log_2\left(\frac{2\omega}{\pi}\right)\right), & \frac{\pi}{4} < \omega \leq \pi \\ 0, & \text{otherwise.} \end{cases} \quad (5.4.3)$$

Gabor Filters

Gabor filter banks allow extracting multi-directional and multi-scale texture information via a systematic parcellation of the Fourier domain with elliptic Gaussian windows [142]. They are not rotation-invariant and are therefore best suited for application where the absolute feature orientation is meaningful. In the spatial domain, Gabor kernels are complex Gaussian-windowed oscillatory functions defined as

$$\phi_{(u,s)}(\mathbf{x}) = \frac{\gamma_s^2}{\pi\sigma_1\sigma_2} e^{-\gamma_s^2\left(\left(\frac{\tilde{x}_{u,1}}{\sigma_1}\right)^2 + \left(\frac{\tilde{x}_{u,2}}{\sigma_2}\right)^2\right)} e^{j2\pi\gamma_s\tilde{x}_{u,1}}, \quad (5.4.4)$$

where $(\tilde{x}_{u,1}, \tilde{x}_{u,2}) = \mathbf{R}_{\theta_u}\mathbf{x}$ defines the radial and orthoradial elliptic Gaussian axes at the orientation θ_u via the rotation matrix \mathbf{R}_{θ_u} . In polar Fourier, σ_1 and σ_2 are the radial and orthoradial standard deviations of the Gaussian window, respectively, and γ_s is the radial position of its center.

We follow the procedure described in [142] to extract response maps at multiple orientations $\{\theta_u\}_{u \in \{1, \dots, U\}}$ and frequencies $\{\gamma_s\}_{s \in \{1, \dots, S\}}$, where σ_1 and σ_2 are defined to cover all directions and scales up to the maximum frequency γ_S .

5.4.2 Energy Term

We propose an energy that combines image intensity and texture information. It is expressed by

$$E_{\text{texture}}(\Theta) = \sum_{n=0}^N w_n E_n(\Theta), \quad (5.4.5)$$

where the w_n are weights balancing the importance of the region-based energies E_n , that allow for the distinction between homogeneous regions in the channel f_n . The weights are of great importance for the efficiency of (5.4.5). In Section 5.4.3 we will present an adequate method to choose them. For E_n , we adopt a strategy similar to the term (5.2.1) *i.e.*, the energy functional is given by

$$E_n(\Theta) = -\frac{1}{|\Sigma|} \left| \iint_{\Omega} f_n(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_\lambda \setminus \Omega} f_n(\mathbf{x}) dx_1 dx_2 \right|, \quad (5.4.6)$$

where $\{f_n\}_{n \in \{0, \dots, N\}}$ is the sequence of images described in Section 5.4.1, Ω and Ω_λ are the surfaces enclosed by the curves \mathbf{r} and its dilated version \mathbf{r}_λ , respectively,

and $\Sigma := \Sigma(\Theta) = \iint_{\Omega} d\mathbf{x}$ is the area of Ω . The discrete counterpart of (5.4.6) for subdivision curves is given by

$$E_n(\mathbf{p}_{(k)}(\Theta)) = -\frac{1}{2^k |\Sigma(\mathbf{p}_{(k)})|} \left| 2 \sum_{m=0}^{2^k M-1} F_n(\mathbf{p}_{(k)}[m]) n_1(\mathbf{p}_{(k)}[m]) - \sum_{m=0}^{2^k M-1} F_n(\mathbf{p}_{\lambda(k)}[m]) n_1(\mathbf{p}_{\lambda(k)}[m]) \right| \quad (5.4.7)$$

(see Proposition 5.2.2), where $\mathbf{p}_{\lambda(k)}$ is the sequence of subdivision points that describes the curve \mathbf{r}_{λ} , n_1 is the first coordinate of the approximation of the unit normal vector given by (5.1.5), and the signed area $\Sigma(\mathbf{p}_{(k)})$ is defined by (5.2.6). The image F_n is given by

$$F_n(x_1, x_2) = \int_{-\infty}^{x_1} f_n(\tau, x_2) d\tau. \quad (5.4.8)$$

5.4.3 Fisher's Linear Discriminant Analysis

The $N + 1$ channels $\{f_n\}_{n \in \{0, \dots, N\}}$ contain information about the object to be segmented and the background of the image f . While some of these information allows to well discriminate the target from its background, others can be redundant. It is thus important to well balance these information. This is the purpose of the weights in (5.4.5).

To adequately set the weights $\{w_n\}_{n \in \{0, \dots, N\}}$ in (5.4.5), we use Fisher's LDA [147], which is a supervised technique for dimensionality reduction and classification. Given two classes C and B , Fisher's LDA seeks the most discriminant hyperplane, characterized by the normal vector \mathbf{w} , that maximizes the between-class variance while minimizing the within-class variance. In our segmentation context, the two classes are the core of the target (C) and the background (B) of f . The vector $\mathbf{w} \in \mathbb{R}^{N+1}$ then contains the optimal weights w_n for the energy term (5.4.5).

Let Ω_C and Ω_B be two regions of interest of C and B , respectively, in the image (see Figure 6.4). We consider $\mathbf{f}(\mathbf{x}) = (f_0(\mathbf{x}), \dots, f_N(\mathbf{x}))$, where f_n is given by (5.4.1), for $\mathbf{x} = (x_1, x_2)$ belonging to Ω_C or Ω_B . The optimal solution is given by [148]

$$\mathbf{w} \propto (S_C + S_B)^{-1}(\boldsymbol{\mu}_C - \boldsymbol{\mu}_B), \quad (5.4.9)$$

Table 5.2: Summary of the Different Types of Image Energies.

Type	Feature	Continuous Formulation	Discrete Formulation
Contour-based	Edge	(2.2.2) in 2D (2.2.3) in 3D	(5.1.4) in 2D (5.1.8) in 3D
	Ridge	(5.3.3)	(5.3.4)
Region-based	Intensity	(2.2.4) in 2D (2.2.7) in 3D	(5.2.5) in 2D (5.2.9) in 3D
	Texture	(5.4.5) and (5.4.6)	(5.4.5) and (5.4.7)

where $S_C, S_B \in \mathbb{R}^{(N+1) \times (N+1)}$ are covariance matrices. Their expressions are given by

$$\boldsymbol{\mu}_I = \frac{1}{|\Sigma_I|} \iint_{\Omega_I} \mathbf{f}(\mathbf{x}) dx_1 dx_2 \quad (5.4.10)$$

and

$$S_I = \frac{1}{|\Sigma_I|} \iint_{\Omega_I} \langle \mathbf{f}(\mathbf{x}) - \boldsymbol{\mu}_I, \mathbf{f}(\mathbf{x}) - \boldsymbol{\mu}_I \rangle dx_1 dx_2, \quad (5.4.11)$$

where $I = \{B, C\}$, $\boldsymbol{\mu}_I$ is the mean vector of size $N + 1$ of the class C or B, and $|\Sigma_I|$ is the area of Ω_I . In practice, we discretize f , and the integrals in (5.4.10) and (5.4.11) are sums over the pixels of Ω_I .

5.5 Conclusion

This chapter is a unification of the image energies that were developed in our publications. Those energies shall be validated in the two next chapters, in which implementation tools to speed up their computation are also presented. We summarized in Table 5.2 the standard and novel energies that we have seen so far. One can refer to this table throughout its reading.

5.6 Appendices

5.6.1 Proof of Proposition 5.1.1

We first recall the following theorem.

Green's Theorem: Let \mathcal{C} be a positively oriented, piecewise-smooth, simple closed curve in a plane and let Ω be the region bounded by \mathcal{C} . If Q and A are functions of (x_1, x_2) defined on an open region containing Ω and have continuous partial derivatives there, then

$$\oint_{\mathcal{C}} (Q(r_1, r_2)dr_1 + A(r_1, r_2)dr_2) = \iint_{\Omega} \left(\frac{\partial A(x_1, x_2)}{\partial x_1} - \frac{\partial Q(x_1, x_2)}{\partial x_2} \right) dx_1 dx_2, \quad (5.6.1)$$

where the path of integration along \mathcal{C} is counterclockwise.

We rewrite (2.2.2) as

$$E_{\text{edgeLR}}(\Theta) = - \oint_{\mathcal{C}} \left(\frac{\partial f(r_1, r_2)}{\partial r_1} dr_2 - \frac{\partial f(r_1, r_2)}{\partial r_2} dr_1 \right). \quad (5.6.2)$$

Using Green's theorem with $Q(r_1, r_2) = -\frac{\partial f(r_1, r_2)}{\partial r_2}$ and $A(r_1, r_2) = \frac{\partial f(r_1, r_2)}{\partial r_1}$, (5.6.2) is expressed as the surface integral

$$\begin{aligned} E_{\text{edgeLR}}(\Theta) &= - \int \int_{\Omega} \Delta f(\mathbf{x}) dx_1 dx_2 = \oint_{\mathcal{C}} G(\mathbf{r}) dr_2 \\ &= \int_0^M G(\mathbf{r}(t)) \frac{dr_2(t)}{dt} dt. \end{aligned} \quad (5.6.3)$$

Taking the derivative of the second component r_2 of \mathbf{r} in (3.3.2) and combining it with (5.6.3), we obtain (5.1.2). ■

5.6.2 Proof of Proposition 5.1.2

We first recall the following theorem.

Theorem of the Riemann Sum: Let $g : [a, b] \rightarrow \mathbb{R}$ be a real function that is Riemann-integrable on $[a, b]$. The Riemann sum R_n is defined by

$$R_n = \frac{b-a}{n} \sum_{m=0}^{n-1} g \left(m \frac{b-a}{n} \right) \quad (5.6.4)$$

and converges to $\lim_{n \rightarrow +\infty} R_n = \int_a^b g(t) dt$.

By combining (4.2.6), (5.1.4), and (5.1.6), we obtain that

$$\lim_{k \rightarrow \infty} E_{\text{edgeSD}}(\mathbf{P}_{(k)}(\Theta)) = - \underbrace{\lim_{k \rightarrow \infty} \frac{1}{2^k} \sum_{m=0}^{2^k M-1} g\left(\frac{m}{2^k}\right)}_E, \quad (5.6.5)$$

where $g(t) = \langle \nabla f(\mathbf{r}(t)), \mathbf{n}(\mathbf{r}(t)) \rangle$ is Riemann-integrable on $[0, M]$ because f, r_1, r_2 are differentiable. We use the theorem of the Riemann sum with $a = 0$, $b = M$, and $n = 2^k M$ to obtain

$$\begin{aligned} E &= - \int_0^M \langle \nabla f(\mathbf{r}(t)), \mathbf{n}(\mathbf{r}(t)) \rangle dt \\ &= - \oint_{\mathcal{C}} \langle \nabla f(\mathbf{r}), \mathbf{n}(\mathbf{r}) \rangle dr, \end{aligned} \quad (5.6.6)$$

which concludes the proof. ■

5.6.3 Notation and Properties of Regular Meshes

In this section, we present properties of regular meshes, *i.e.*, without extraordinary vertices, that will be used in Appendices 5.6.4 and 5.6.7. For subdivision schemes applied to regular meshes, the vertices of a mesh $\mathcal{M}_{(k)}$, $k \geq 0$, are on a rectangular grid (u, v) and

$$N_k = 2^{2k} N_0. \quad (5.6.7)$$

Let the N_0 control points be on a rectangular grid of size $(N_{0,u} \times N_{0,v})$, such that $N_{0,u}, N_{0,v} \in \mathbb{N}$ and $N_{0,u} N_{0,v} = N_0$. We re-express the set $\mathbb{P}_{(k)}$, initially defined in Section 4.6.1, as $\mathbb{P}_{(k)} = \{\mathbf{p}_{(k)}[p, q] \in \mathbb{R}^3, p \in \{0, \dots, 2^k N_{0,u} - 1\}, q \in \{0, \dots, 2^k N_{0,v} - 1\}\}$ (Figure 5.3).

For convergent subdivision schemes applied to regular meshes, we have that

$$\lim_{k \rightarrow \infty} \mathbf{p}_{(k)}[p, q] = \boldsymbol{\sigma}(u, v) \Big|_{(u,v) = \left(\frac{p}{2^k}, \frac{q}{2^k}\right)}. \quad (5.6.8)$$

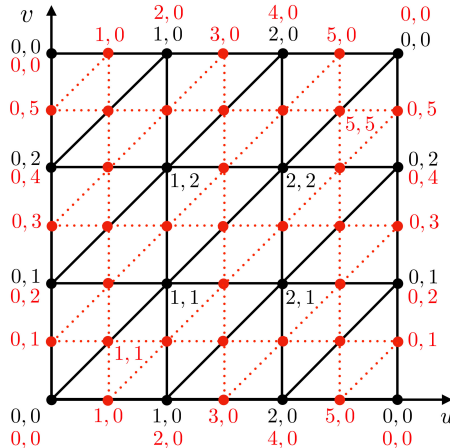


Figure 5.3: Flat image of a regular mesh that forms a closed surface. We show the mesh for two subdivision steps of an interpolating subdivision scheme. The initial coarse mesh $\mathcal{M}_{(0)}$ (black) is made of $N_0 = 9$ control points that are on a rectangular grid of size $(N_{0,u} \times N_{0,v})$, with $N_{0,u} = N_{0,v} = 3$. At the first subdivision step, we obtain a regular mesh $\mathcal{M}_{(1)}$ (red) made of $2^2 N_0 = 36$ vertices (black and red circles) that are on a rectangular grid of size $(2N_{0,u} \times 2N_{0,v})$. The indices correspond to the pair (p, q) of $\mathbf{p}_{(k)}[p, q]$, $k = 0, 1$.

5.6.4 Proof of Proposition 5.1.3

We first recall a classical result.

Theorem of the double Riemann sum: *Let $g : [a, b] \times [c, d] \rightarrow \mathbb{R}^2$ be a real continuous function that is Riemann integrable on $[a, b] \times [c, d]$. The Riemann sum $R_{n,m}$ defined by*

$$R_{n,m} = \frac{b-a}{n} \frac{d-c}{m} \sum_{p=0}^{n-1} \sum_{q=0}^{m-1} g\left(p \frac{b-a}{n}, q \frac{d-c}{m}\right) \tag{5.6.9}$$

satisfies

$$\lim_{n \rightarrow +\infty} \lim_{m \rightarrow +\infty} R_{n,m} = \int_a^b \int_c^d g(u, v) du dv. \tag{5.6.10}$$

We first prove Proposition 5.1.3 for regular meshes that are topologically equivalent to a torus. Surfaces that are topologically equivalent to a torus are periodic along u, v and are parametrized by

$$\sigma(u, v) = \sum_{p=0}^{N_{0,u}-1} \sum_{q=0}^{N_{0,v}-1} \mathbf{c}[p, q] \varphi_{N_{0,u}}(u-p) \varphi_{N_{0,v}}(v-q), \quad (5.6.11)$$

where $(u, v) \in [0, N_{0,u}] \times [0, N_{0,v}]$, the $\mathbf{c}[p, q]$ are the control points, and $\varphi_{N_{0,u}}$ and $\varphi_{N_{0,v}}$ are the $N_{0,u}$ - and $N_{0,v}$ -periodizations (Equation (2.1.3)), respectively, of a suitable basis function φ .

Using the notation for regular meshes described in Appendix 5.6.3, we rewrite $E_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta))$ as

$$E_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta)) = -\frac{1}{2^{2k}} \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} \langle \nabla f(\mathbf{p}_{(k)}[p, q]), \mathbf{n}(\mathbf{p}_{(k)}[p, q]) \rangle. \quad (5.6.12)$$

Combining (5.6.8) and (5.6.12), we have that

$$\lim_{k \rightarrow \infty} E_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta)) = - \underbrace{\lim_{k \rightarrow \infty} \frac{1}{2^{2k}} \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} g\left(\frac{p}{2^k}, \frac{q}{2^k}\right)}_E, \quad (5.6.13)$$

where $g(u, v) = \langle \nabla f(\boldsymbol{\sigma}(u, v)), \mathbf{n}(\boldsymbol{\sigma}(u, v)) \rangle$ is Riemann integrable on $[0, N_{0,u}] \times [0, N_{0,v}]$ because $f, \sigma_1, \sigma_2, \sigma_3 \in \mathcal{C}^1$. We use the theorem of the double Riemann sum with $a = c = 0$, $b = N_{0,u}$, $d = N_{0,v}$, $n = 2^k N_{0,u}$, and $m = 2^k N_{0,v}$ to obtain that

$$E = - \int_0^{N_{0,u}} \int_0^{N_{0,v}} \langle \nabla f(\boldsymbol{\sigma}(u, v)), \mathbf{n}(\boldsymbol{\sigma}(u, v)) \rangle \, dudv. \quad (5.6.14)$$

We have that

$$\begin{aligned} d\boldsymbol{\sigma} &= (\boldsymbol{\sigma}_u \wedge \boldsymbol{\sigma}_v) \, dudv \\ &= \mathbf{n}(\boldsymbol{\sigma}(u, v)) \, dudv. \end{aligned} \quad (5.6.15)$$

So, using (5.6.14) and (5.6.15), we finally obtain that

$$E = - \iint_S \langle \nabla f(\boldsymbol{\sigma}), d\boldsymbol{\sigma} \rangle. \quad (5.6.16)$$

This concludes the proof for regular meshes with a topology equivalent to a torus. For regular meshes that form closed surfaces with a different topology, the proof is the same. The parametrization (5.6.11) still holds for $(u, v) \in [0, N_{0,u}] \times [0, N_{0,v}]$ but the bounds of the sum over q depend on the support of φ and additional conditions over the control points may be necessary.

For meshes with extraordinary vertices, we give a gist of the proof, as a detailed one may imply the introduction of too many notations and notions. The gist of the proof is based on the following mathematical pipeline:

- we consider a portion of the initial mesh that contains one extraordinary vertex only, and we locally express the limit surface $\boldsymbol{\sigma}$ as the union between the limit point of the extraordinary vertex and the sequence of rings $\{\boldsymbol{\epsilon}_{(k)}\}_{k \geq 0}$ defined by the regular vertices around it [136, 149];
- for each ring $\boldsymbol{\epsilon}_{(k)}$, we apply the reasoning that we used for regular meshes where the Riemann integral is instead defined on a local part of the surface $\boldsymbol{\sigma}$;
- finally, doing this for each ring and taking the union over k we conclude the proof.

■

5.6.5 Proof of Proposition 5.2.1

We rewrite (2.2.4) as

$$\begin{aligned} E_{\text{intensityLR}}(\Theta) &= -\frac{1}{|\Sigma|} \left| 2 \int \int_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \int \int_{\Omega_\lambda} f(\mathbf{x}_\lambda) dx_1 dx_2 \right| \\ &= -\frac{1}{|\Sigma|} \left| 2 \oint_{\mathcal{C}} F(\mathbf{r}) dr_2 - \oint_{\mathcal{C}_\lambda} F(\mathbf{r}_\lambda) dr_{2,\lambda} \right|, \end{aligned} \quad (5.6.17)$$

where \mathcal{C} and \mathcal{C}_λ are the positive oriented contours described by \mathbf{r} and \mathbf{r}_λ , respectively, and the image F is given by (5.2.3). The area can be expressed as

$\Sigma := \Sigma(\Theta) = \iint_{\Omega} dx_1 dx_2$, while Green's theorem (recalled in Appendix 5.6.1) yields the signed area

$$\Sigma = - \oint_{\mathcal{C}} r_2 dr_1. \quad (5.6.18)$$

Using equation (3.3.2) and its derivative, and combining with (5.6.17) and (5.6.18), we obtain (5.2.2) and (5.2.4). ■

5.6.6 Proof of Proposition 5.2.2

We use (4.2.6), (5.1.6), (5.2.5), and the theorem of the Riemann sum (recalled in Appendix 5.6.2) with $a = 0$, $b = M$, $n = 2^k M$, and $g(t) = F(\mathbf{r}(t))\dot{r}_2(t)$ with F defined in (5.2.3). The function g is Riemann-integrable on $[0, M]$ because $f \in \mathcal{C}^1$. We obtain that

$$\begin{aligned} \lim_{k \rightarrow \infty} E_{\text{intensitySD}}(\mathbf{p}(k)(\Theta)) &= \frac{1}{|\Sigma|} \left| 2 \int_0^M F(\mathbf{r}(t))\dot{r}_2(t) dt - \int_0^M F(\mathbf{r}_{\lambda}(t))\dot{r}_{\lambda,2}(t) dt \right| \\ &= \frac{1}{|\Sigma|} \left| 2 \oint_{\mathcal{C}} F(\mathbf{r}) dr_2 - \oint_{\mathcal{C}_{\lambda}} F(\mathbf{r}_{\lambda}) dr_{\lambda,2} \right|, \end{aligned} \quad (5.6.19)$$

where \mathcal{C} and \mathcal{C}_{λ} are the positive oriented contours that describe \mathbf{r} and \mathbf{r}_{λ} , respectively. We use Green's theorem with $A = F$ and $Q = 0$. We finally obtain that

$$\begin{aligned} \lim_{k \rightarrow \infty} E_{\text{intensitySD}}(\mathbf{p}(k)(\Theta)) &= \frac{1}{|\Sigma|} \left| 2 \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_{\lambda}} f(\mathbf{x}_{\lambda}) dx_{\lambda_1} dx_{\lambda_2} \right| \\ &= \frac{1}{|\Sigma|} \left| \iint_{\Omega} f(\mathbf{x}) dx_1 dx_2 - \iint_{\Omega_{\lambda} \setminus \Omega} f(\mathbf{x}) dx_1 dx_2 \right|. \end{aligned} \quad (5.6.20)$$

For $\Sigma(\mathbf{p}(k))$, we apply the same reasoning as previously, using first the theorem of the Riemann sum and then Green's theorem, to obtain that

$$\Sigma(\mathbf{p}(k)(\Theta)) \xrightarrow{k \rightarrow \infty} \Sigma(\Theta) = \iint_{\Omega} dx_1 dx_2. \quad (5.6.21)$$

■

5.6.7 Proof of Proposition 5.2.3

We first recall the following theorem.

Gauss' theorem: *Let \mathcal{V} be a subset of \mathbb{R}^n that is compact and has a piecewise-smooth boundary \mathcal{S} . If \mathbf{G} is a continuously differentiable vector field defined on a neighborhood of \mathcal{V} , then we have that*

$$\oint_{\mathcal{S}} \langle \mathbf{G}, d\mathbf{S} \rangle = \iiint_{\mathcal{V}} \operatorname{div}(\mathbf{G}) dV. \quad (5.6.22)$$

For this proof, we follow the same procedure as described in Appendix 5.6.4, with the same gist of proof for meshes with extraordinary vertices. Hereafter, we thus only detail the proof for regular meshes with a topology equivalent to a torus.

Using the notation for regular meshes introduced in Appendix 5.6.3, we can rewrite $E_{\text{intensitySD}}(\mathbb{P}_{(k)}(\Theta))$ as

$$\begin{aligned} E_{\text{intensitySD}}(\mathbb{P}_{(k)}(\Theta)) = & - \frac{1}{|V(\mathbb{P}_{(k)})| 2^{2k}} \left| 2 \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} F(\mathbf{p}_{(k)}[p, q]) n_1(\mathbf{p}_{(k)}[p, q]) \right. \\ & \left. - \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} F(\mathbf{p}_{\lambda(k)}[p, q]) n_1(\mathbf{p}_{\lambda(k)}[p, q]) \right|. \end{aligned} \quad (5.6.23)$$

As σ is the limit surface of the subdivision scheme, *i.e.*, $\sigma = \lim_{k \rightarrow \infty} \mathcal{M}_{(k)}$, we have

$$\lim_{k \rightarrow \infty} |V(\mathbb{P}_{(k)}(\Theta))| = |V|, \quad (5.6.24)$$

where $V(\mathbb{P}_{(k)})$ and V are the volume enclosed by $\mathcal{M}_{(k)}$ and σ , respectively. Combining (5.6.8), (5.6.23) and (5.6.24) we obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} E_{\text{intensitySD}}(\mathbb{P}_{(k)}(\Theta)) = & - \lim_{k \rightarrow \infty} \frac{1}{2^{2k} |V|} \left| 2 \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} g\left(\frac{p}{2^k}, \frac{q}{2^k}\right) \right. \\ & \left. - \sum_{p=0}^{2^k N_{0,u}-1} \sum_{q=0}^{2^k N_{0,v}-1} g_{\lambda}\left(\frac{p}{2^k}, \frac{q}{2^k}\right) \right| \\ = & E, \end{aligned} \quad (5.6.25)$$

where $g(u, v) = F(\boldsymbol{\sigma}(u, v))n_1(\boldsymbol{\sigma}(u, v))$ and $g_\lambda(u, v) = F(\boldsymbol{\sigma}_\lambda(u, v))n_1(\boldsymbol{\sigma}_\lambda(u, v))$ are Riemann integrable on $[0, N_{0,u}] \times [0, N_{0,v}]$ because $f, \sigma_1, \sigma_2, \sigma_3, \sigma_{\lambda,1}, \sigma_{\lambda,2}, \sigma_{\lambda,3} \in \mathcal{C}^1$. We use the theorem of the double Riemann sum (recalled in Appendix 5.6.4) with $a = c = 0$, $b = N_{0,u}$, $d = N_{0,v}$, $n = 2^k N_{0,u}$ and $m = 2^k N_{0,v}$ to obtain that

$$\begin{aligned}
 E &= -\frac{1}{|V|} \left| 2 \int_0^{N_{0,u}} \int_0^{N_{0,v}} F(\boldsymbol{\sigma}(u, v))n_1(\boldsymbol{\sigma}(u, v))dudv \right. \\
 &\quad \left. - \int_0^{N_{0,u}} \int_0^{N_{0,v}} F(\boldsymbol{\sigma}_\lambda(u, v))n_1(\boldsymbol{\sigma}_\lambda(u, v))dudv \right| \\
 &= -\frac{1}{|V|} \left| 2 \int_0^{N_{0,u}} \int_0^{N_{0,v}} \langle \mathbf{G}(\boldsymbol{\sigma}(u, v)), \mathbf{n}(\boldsymbol{\sigma}(u, v)) \rangle dudv \right. \\
 &\quad \left. - \int_0^{N_{0,u}} \int_0^{N_{0,v}} \langle \mathbf{G}(\boldsymbol{\sigma}_\lambda(u, v)), \mathbf{n}(\boldsymbol{\sigma}_\lambda(u, v)) \rangle dudv \right|, \quad (5.6.26)
 \end{aligned}$$

where $\mathbf{G}(x_1, x_2, x_3) = (F(x_1, x_2, x_3), 0, 0)$. Thus, we have that

$$E = -\frac{1}{|V|} \left| 2 \iint_S \langle \mathbf{G}, d\boldsymbol{\sigma} \rangle - \iint_{S_\lambda} \langle \mathbf{G}, d\boldsymbol{\sigma} \rangle \right|. \quad (5.6.27)$$

The use of Gauss' theorem then yields

$$\begin{aligned}
 E &= -\frac{1}{|V|} \left| 2 \iiint_V \operatorname{div}(\mathbf{G})dV - \iiint_{V_\lambda} \operatorname{div}(\mathbf{G})dV \right| \\
 &= -\frac{1}{|V|} \left| 2 \iiint_V f dV - \iiint_{V_\lambda} f dV \right| \\
 &= -\frac{1}{|V|} \left| \iiint_V f(\mathbf{x})dx_1dx_2dx_3 - \iiint_{V_\lambda \setminus V} f(\mathbf{x})dx_1dx_2dx_3 \right|, \quad (5.6.28)
 \end{aligned}$$

which concludes the proof in case of regular meshes. ■

Chapter 6

Design of Active Contours

In this chapter, we merge the theories on curve representation and image energy presented so far, and summarized in Tables 5.1 and 5.2, to design four new active contours. Each one has a specific purpose that addresses a limitation of standard parametric snakes [64].

We developed our active contours with the following characteristics:

- The concern to fill in the requirements mentioned in Section 1.2.3, that is robustness, reasonable computational time, flexibility, and friendly user interaction.
- The energy consists in an image energy term only (see Section 1.2.2 and (1.2.1)). In fact, we use representation models that ensure the smoothness of the curve, which eliminates the need for an explicit internal energy term. We also obviate the constraint energy and we instead provide an interface allowing the user to interact with the snake.
- The optimization process is efficiently carried out by a Powell-like line search method [61] that can be summarize as follows: For each control point, a direction is chosen depending on the partial derivatives of the energy. Then, the control point is displaced along the selected direction to minimize the energy. The process is repeated until convergence.

- A validation is performed on both synthetic and real data. We use the Jaccard J index to measure the overlap between a segmentation result Ω and the corresponding ground truth Ω_{GT} . It is defined as

$$J = \frac{|\Omega \cap \Omega_{\text{GT}}|}{|\Omega \cup \Omega_{\text{GT}}|}. \quad (6.0.1)$$

Clearly, $0 \leq J \leq 1$, and perfect overlap is described by $J = 1$.

- An implementation as a user-friendly plugin¹ for the bioimaging platform Icy [11]. In order for the software to be usable (see Section 1.1.3), we provide a website with the source code, some documentations (abstract, link to the related scientific paper and demo), and the necessary data and parameters needed to replicate some of our experiments.

We present the four new active contours in independent sections that are structured as follows: We first describe the framework. Then, we provide implementation details or algorithms that speed up the computation. Finally, we perform an extensive validation of the snake. The main contributions related to the design of the active contour are summarized in a concluding part.

This chapter is organized as follows: In Sections 6.1 and 6.2, we present new parametric snakes that are either locally refinable or that incorporate texture information. In Sections 6.3 and 6.4, we propose a new family of active contours by taking advantage of subdivisions, and we illustrate their use on different applications.

¹All plugins can be found at <http://bigwww.epfl.ch/demo/deformable-models-segmentation.html>.

6.1 Locally Refinable Parametric Snakes

The shapes of biological structures often exhibit different levels of detail [45, 66]. For conventional parametric snakes, this is dealt with in a global fashion by simply increasing the number of control points resulting in an increase of the degrees of freedom of the model [50, 150]. While this improves the approximation power of the snake model [151], it renders the optimization more challenging and increases the computation substantially. Moreover, introducing more degrees of freedom in the part of the curve where they are not required does not necessarily improve the segmentation outcome. On the other hand, few parameters allow for faster optimization [152–154] but with less accurate approximation. A precise location of the insertion of the additional control points is thus preferable.

In this section, we propose² a new parametric snake that has the ability to locally increase its approximation power. This allows for a more efficient allocation of the degrees of freedom of the snake by concentrating them on segments of higher complexity. This is controlled by a user-interface that permits the refinement of an initial segmentation around an anchor position selected by a user. For this purpose we exploit the refinability property of scaling functions [95]. We demonstrate the accuracy of our snake and its robustness under noisy conditions on phantom data. We also present segmentation results on real cell images, which are our main target.

6.1.1 Framework

We represent our active contour by a parametric closed curve that is locally refinable with respect to the control point $\mathbf{c}[p]$, as described by (3.3.2). Its shape is parametrized by the control points $\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}} \cup \{\tilde{\mathbf{c}}_p[n]\}_{n \in \{n_0, \dots, n_0+N-1\}}$, where $\tilde{\mathbf{c}}_p$ is given by (3.3.3), M is the number of control point before refinement and $\{n_0, \dots, n_0+N-1\}$ is the support of the refinement filter. Note that the index p is freely chosen and that one can refine the curve at several specific locations by applying Proposition 3.3.1 with respect to each corresponding control point.

For the snake energy, we use a combination of the edge and region-based ener-

²This section is based on our publication [68], in collaboration with D. Schmitter and M. Unser. A demo of the corresponding plugin and related documentation are available at <http://bigwww.epfl.ch/demo/locally-refinable-snake/>.

gies (5.1.2) and (5.2.2) such as

$$E_{\text{snake}}(\Theta) = bE_{\text{EdgeLR}}(\Theta) + (1 - b)E_{\text{intensityLR}}(\Theta), \quad (6.1.1)$$

where $b \in [0, 1]$ is a trade-off parameter that balances the contribution of the two energies. The curve of the snake is thus deformed based on gradient and intensity information from image data.

6.1.2 Implementation Details

The local refinement is handled by the user through an interface by clicking on the control point $\mathbf{c}[p]$. He can also adjust the parameter b and the refinement factor $\rho > 1$.

From a computational point of view, the few number of control points allowed by the locally refinable parametric curve speeds up the optimization. Moreover, the use of the explicit equations (5.1.2) and (5.2.2) allows for the exact analytical computation of the energy gradient with respect to each control point. To further speed up the computation of E_{snake} , that is evaluated at each iteration of the optimization process, we 1) precompute and store in look-up tables the images (5.1.3) and (5.2.3) that only depend on the input image f ; and 2) use the inner-product calculus that we propose in Appendix A to efficiently compute the exact calculation of the signed area Σ in (5.2.2).

6.1.3 Experiments and Validation

We performed experiments on phantom and real data to test the accuracy and the robustness of our proposed method. For each experiment, in (3.3.2) we chose as refinement factor $\rho = 2$ and we used quadratic B-splines as scaling functions, *i.e.*, $\varphi = \beta_{(0,0,0)}$ (see Section 2.3.1). The corresponding refinement filter h is defined by its z-transform by

$$H(z) = \frac{1}{4} (1 + z^{-1})^3. \quad (6.1.2)$$

Phantom data

We created a test image simulating fluorescence microscopy showing a cell. We then corrupted this image by different levels of additive Gaussian white noise. For each

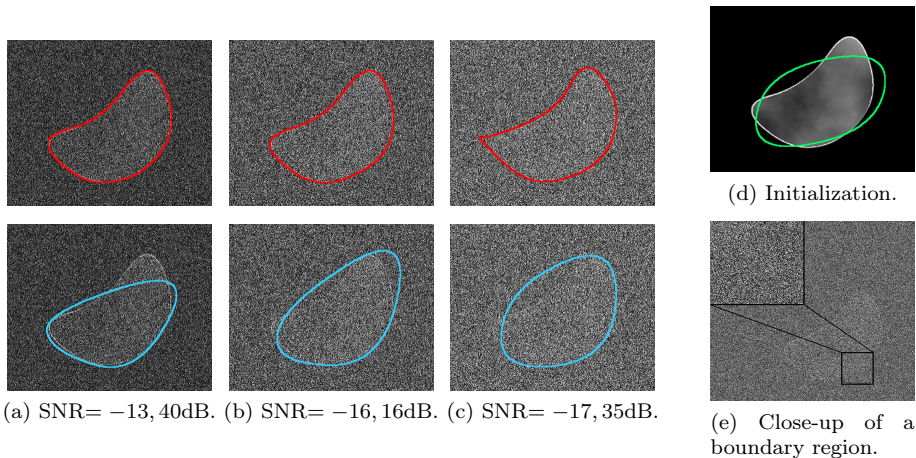


Figure 6.1: Robustness with respect to noise of the locally refinable parametric snake. (a)-(c) Comparison for different SNR between two quadratic B-spline snakes with $M = 4$: locally refinable (top row) and traditional (bottom row) snakes; (d) initialization for both snakes; (e) close-up of a boundary region between the test cell and its background, SNR = $-17, 35\text{dB}$.

image, we segmented the structure of interest using our locally refinable parametric snake as well as a traditional parametric snake [64]. For both snakes we used the same initialization with $M = 4$ control points (Figure 6.1 (d)).

For our snake, we first performed an optimization without local refinement. In this way, our initial segmentation is rough at first but the upside is a fast segmentation. Then, the user clicks on a desired control point through the interface and the corresponding basis function is refined. Finally, we optimized again to refine local details.

Signal-to-noise ratios (SNRs) corresponding to the noise level and Jaccard indices are shown in Table 6.1 and illustrated in Figure 6.1. Both Table 6.1 and Figure 6.1 show the improved accuracy induced by the local refinement and the robustness of our method.

Table 6.1: Jaccard indices for segmentation of noisy data.

SNR [dB]	With local refinement	Without local refinement
-7, 44	0.95	0.79 (fail)
-10, 95	0.95	0.73 (fail)
-13, 40	0.95	0.79 (fail)
-15, 07	0.94	0.81 (fail)
-16, 16	0.94	0.83 (fail)
-16, 88	0.92	0.84 (fail)
-17, 35	0.91	0.78 (fail)

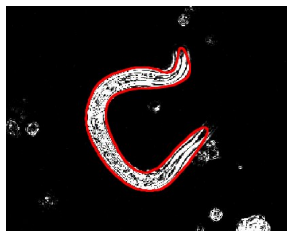
Real data

We have applied our snake on two real fluorescence microscopy images, where the ground truth is unknown. They are challenging because of the presence of noise, and because the structures of interest have different level of detail. The segmentation outcomes are satisfactory (Figure 6.2).

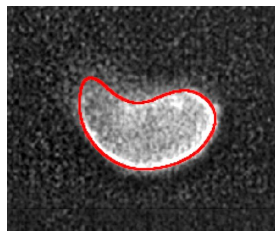
6.1.4 Conclusions

We have presented a new and complete formulation of locally refinable parametric snakes for image segmentation. Through a user interface, we introduced the possibility of inserting additional basis functions at a specific location. Our method is generic and can be used with any valid scaling function. It allows for a fast and efficient energy computation with few control points. We have demonstrated its ability of improving segmentation results as well as its robustness under noisy conditions. The primary contributions related to this work are:

- A novel parametrization of closed curves that are refinable locally (see (3.3.2));
- The derivation of corresponding edge and region-based energies (5.1.2) and (5.2.2)



(a) *Heligmosomoides polygyrus bakeri*.



(b) Sickle cell.

Figure 6.2: Segmentation of an *Heligmosomoides polygyrus bakeri* (a) and a sickle cell (b) on fluorescence microscopy images using locally refinable parametric snakes.

- The demonstration of the benefits of refinability in the context of semi-interactive segmentation.

6.2 Texture-Driven Parametric Snakes

In this section, our motivation is to develop a general and versatile framework for interactive segmentation of a single structure of interest in an image, possibly under low-contrast conditions. We want the user to be able to easily specify the desired structure and to modify the outcome when needed.

As it was mentioned in Section 2.2, the most common energies of active contours are based on edge or intensity information aggregated from either inside or on the curve [44]. However, these intensity-only schemes do not perform well on images where the contrast between the object of interest and the background is low. In this context, an efficient approach to account for the spatial organization of the pixels inside the desired object is to incorporate texture information. It allows one to capture the morphological structure of a tissue [141].

Recent approaches were proposed to incorporate texture information into active contours [155–158]. Among them, common characterizations of texture properties were gray-level co-occurrence matrices [139], Gabor filters [140, 159, 160], sparse texture dictionaries [161], variational image decompositions [162], or deep learning based on convolutional neural networks (CNN) [138, 163–167]. Those methods can be categorized into supervised and unsupervised methods. A limitation of unsupervised approaches, such as in [162], is that the incorporation of prior knowledge is difficult. Meanwhile, a limitation of supervised approaches such as CNNs is that they cannot be trained on-the-fly with only a few labeled pixels, as required for natural interactions with snakes. For instance, in the interactive methods of [139] and [161], texture is learned from the pixels inside the manual initialization of the snake or by providing region boxes for the foreground and the background, respectively.

In this section, we design³ a new texture-driven parametric snake for the supervised and interactive segmentation of single structures of interest in images. The framework is based on the theory presented in Section 5.4, that is an energy term that combines image intensity and texture information, and the Fisher’s linear discriminant analysis (LDA) that finds the optimal balance between the two type of information. A very small number of samples provided by the user is sufficient to perform adequate on-the-fly training. The framework is valid for any filter-based

³This section is based on our publication [74], in collaboration with A. Depeursinge and M. Unser. A demo of the corresponding plugin and related documentation are available at <http://bigwww.epfl.ch/demo/texture-snake/>.

texture feature extraction method, including Gabor filters [142] or circular harmonic wavelets (CHW) [143]. Here, we mainly focus on CHWs as they provide a powerful tool to model local circular frequencies at multiple scales with invariance to local image rotations. We perform a comprehensive performance evaluation of the texture-driven parametric snake on both synthetic and natural images. We measure its robustness and accuracy with respect to noise and initialization, as well as to parameter sensitivity. In addition, we compare our model to supervised and interactive segmentation methods. Regarding the advantages of our method, it is worth noting that a comparison to fully automatic approaches as CNN would not be relevant as they cannot be trained on-the-fly. Overall, our approach allows one to efficiently segment subtle structures in low-contrast images with only a few clicks while allowing a high level of interaction with the user.

6.2.1 Framework

We describe our snake by the parametric closed curve (2.1.2) encoded by M control points $\Theta = \{\mathbf{c}[m]\}_{m \in \{0, \dots, M-1\}}$. We use as basis function φ the exponential B-spline defined in [64] (Equation (8)) by

$$\varphi(t) = \text{sinc}\left(\frac{1}{M}\right)^{-2} \beta_{(0, -\frac{2i\pi}{M}, \frac{2j\pi}{M})} = \begin{cases} \frac{\cos\left(\frac{2\pi|t|}{M}\right)\cos\left(\frac{\pi}{M}\right) - \cos\left(\frac{2\pi}{M}\right)}{1 - \cos\left(\frac{2\pi}{M}\right)} & 0 \leq |t| \leq 0.5, \\ \frac{1 - \cos\left(\frac{2\pi(1.5 - |t|)}{M}\right)}{2\left(1 - \cos\left(\frac{2\pi}{M}\right)\right)} & 0.5 \leq |t| \leq 1.5, \\ 0 & 1.5 \leq |t|, \end{cases} \quad (6.2.1)$$

with $M \geq 3$. This basis function ensures that the snake can perfectly reproduce elliptical shapes using few control points, which is relevant to delineate blob-like objects. In addition, the snake is versatile enough to provide good approximations of any closed curves. The exponential B-spline has a small support, which is advantageous for both computational aspects and the user interaction (moving one control point affects the structure of the snake locally only). Moreover, it verifies the partition-of-unity condition (2.1.8), which ensures that our model is invariant under affine transformations.

To drive the deformation of the curve, we use the energy functional $E_{\text{snake}}(\Theta) = E_{\text{texture}}(\Theta)$ given by (5.4.5) that combines image intensity and texture information. It is obtained as follows: we first perform a texture analysis of the image using a bank of N filters $\{\phi_n\}_{n \in \{1, \dots, N\}}$ (see Section 5.4.1); then, we perform

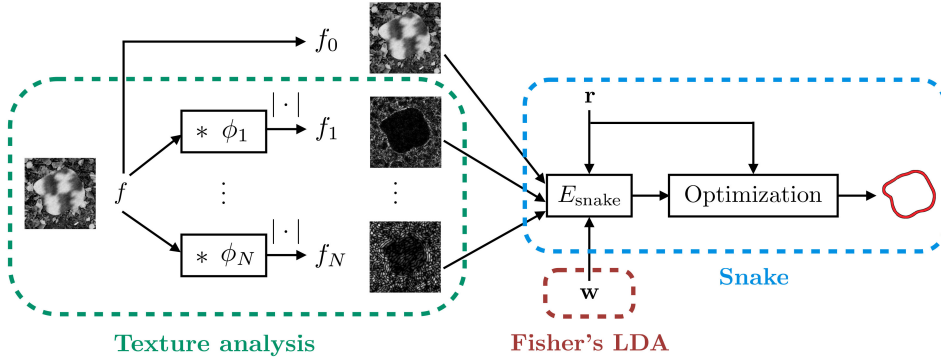


Figure 6.3: Flowchart of the proposed framework: a texture analysis is first performed with a bank of filters $\{\phi_n\}_{n \in \{1, \dots, N\}}$. Note that the symbol $*$ denotes a convolution. Then, the original image $f_0 = f$ and the resulting positive response maps $\{f_n = |f * \phi_n|\}_{n \in \{1, \dots, N\}}$ are balanced using Fisher's linear discriminant analysis. We thus obtain the vector of weights $\mathbf{w} \in \mathbb{R}^{(N+1)}$. Finally, the curve \mathbf{r} of the snake is deformed through the minimization of the region-based energy E_{texture} given (5.4.5). This term allows for the distinction between homogeneous regions in each channel f_n weighted by w_n .

Fisher's LDA (see Section 5.4.3) to obtain a vector of weights that balances the original image $f_0 = f$ and the positive response maps $\{f_n = |f * \phi_n|\}_{n \in \{1, \dots, N\}}$. Finally, the term (5.4.5) allows for the distinction between homogeneous regions in each channel f_n weighted by w_n . The flowchart of the proposed framework is depicted in Figure 6.3.

6.2.2 Implementation Details

Fast Implementation

The main computational bottleneck of our framework is the evaluation of the surface integrals in (5.4.6), which needs to be performed $(N + 1)$ times at each iteration of the optimization process. We use Green's theorem to efficiently implement (5.4.6)

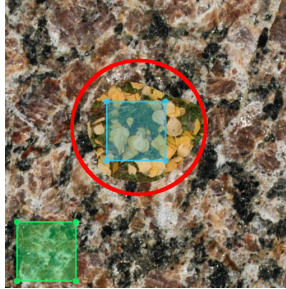


Figure 6.4: Extraction of Ω_C (blue ROI) and Ω_B (green ROI) for training Fisher's LDA.

with line integrals as

$$E_n(\Theta) = -\frac{1}{|\Sigma|} \left| 2 \oint_{\mathcal{C}} F_n(\mathbf{r}) dr_2 - \oint_{\mathcal{C}_\lambda} F_n(\mathbf{r}) dr_2 \right|, \quad (6.2.2)$$

where \mathcal{C} and \mathcal{C}_λ are the positive oriented contours described by \mathbf{r} and \mathbf{r}_λ , respectively, and the image F_n is given by (5.4.8). Similarly, we have that $\Sigma := \Sigma(\Theta) = \oint_{\mathcal{C}} r_1 dr_2$. The use of Green's theorem dramatically reduces the computational cost. To further accelerate the computation, we 1) precompute and store in lookup tables the images F_n and the weights w_n , for $n \in \{0, \dots, N\}$; and 2) use the inner-product calculus that we propose in Appendix A to efficiently compute the signed area Σ in (5.4.6).

Supervision of Fisher's LDA

Two rectangular ROIs Ω_C and Ω_B , necessary to train Fisher's LDA (see Section 5.4.3), are automatically extracted from the initialization of the snake. The first one is localized at the center of gravity of the initialization and the second one outside of the snake (Figure 6.4). A manual mode is also provided to adjust either one of the ROIs when needed. In the global framework, Fisher's LDA is trained on-the-fly once during the initialization of the snake. The resulting weights remain then unchanged during the entire optimization process.

6.2.3 Experiments and Validation

We proceed in three steps to evaluate the performance of the proposed texture-driven parametric snake. First, we test the effect of the parameters on the accuracy of the outcome and study the robustness with respect to initialization and noise. Second, we compare the proposed snake in term of accuracy against other segmentation methods. Third, we illustrate applications on real data. In all the following experiments we use CHWs in our method to extract texture information.

Databases

To validate our model, we created three databases drawn from the real textures of the Prague Texture Segmentation Benchmark⁴ [168]. Each database was created based on the following pipeline. First, we selected a set of texture classes; then, each texture was combined in pairs using a binary mask of blob-like shape to create an image of (512×512) pixels. The mask was obtained by thresholding a mixture of several Gaussians with random parameters. To vary the shape to be segmented, we used the five different masks of Figure 6.5 for each combination. For the first database, called Database 1, we used a set of ten textures of different classes (*e.g.*, wood, stone, flowers). The textures are shown in Figure 6.6 (a). Database 1, made of 450 images, allows us to test the snake on a diverse set of texture patterns. An image of this database is illustrated in Figure 6.16. The two other databases were constructed using five textures of the same class. Database 2 is made of the class "wood" and Database 3 of the class "flower". The corresponding textures are shown in Figure 6.6 (b) and (c). We use those two databases, made of 100 images each, to study the efficiency of the snake when segmenting similar textures that differ only in subtle ways.

Databases 1, 2, and 3 are made of color images in the RGB representation. One advantage of the proposed snake is that it can handle several channels. In the case of RGB images, it uses both texture and intensity information in every color channel. However, the multichannel information can be predominant over the texture. Typically, textures of Database 3 are very similar (*i.e.*, flowers) but often the color differs. Hence, in order to evaluate the ability of our snake to discriminate textures, as opposed to colors, the validation is performed on both

⁴The textures were taken from <http://mosaic.utia.cas.cz/index.php?act=intro>.



Figure 6.5: Masks used for the evaluation.



(a) Database 1.



(b) Database 2.



(c) Database 3.

Figure 6.6: Textures used for the evaluation.

RGB and grayscale versions of the three databases. The grayscale images are obtained by averaging the red, green, and blue channels of the RGB images.

For all experiments, Fisher’s LDA is trained using two fixed ROIs that contain the foreground and background in each mask of Figure 6.5.

Parameters and Validation of the Model

Degrees of Freedom of the Curve: The number M of control points is an important parameter of the proposed snake. The choice of M depends on the application. A large value of M increases the ability of the snake to approximate intricate shapes but makes the optimization process more complex and penalizes robustness. To illustrate this, we segmented Database 1 for different values of M , for $P = 5$ harmonics and $Q = 3$ scales. The corresponding Jaccard indices are reported in Figure 6.7. The default box spans from the 0.25 quantile to the 0.75 quantile. The dark (grey, respectively) dots are the outliers defined as points beyond 1.5 (3, respectively) times the interquartile range from the edge of the box. We observe that the median increases as M increases. However, the segmentation becomes less robust as the number of outliers increases. The best tradeoff between accuracy and robustness was found to be $M = 6$ for Database 1. In fact, keeping M small acts as a regularizer for the curve.

Influence of P and Q for the CHW Decomposition: We study the impact of the number of harmonics and scales on the accuracy of the segmentation outcome. For fixed P and Q , we can reconstruct the image f_{snake} to generate a two-dimensional projection that estimates what the snake “sees” using

$$f_{\text{snake}} = \sum_{n=0}^N w_n f_n, \quad (6.2.3)$$

where $\{w_n\}_{n \in \{0, \dots, N\}}$ are the weights in (5.4.5) estimated with Fisher’s LDA. In Figure 6.8, f_{snake} is shown for different values of P and Q , along with their Jaccard index. The original image is a grayscale image of Database 2. The initialization of the snake and the original image are depicted in Figure 6.9. We observe that the wavelet scale acts as regularizer. It smooths the textures on f_{snake} . At high P and Q , the image is less detailed and the snake is less

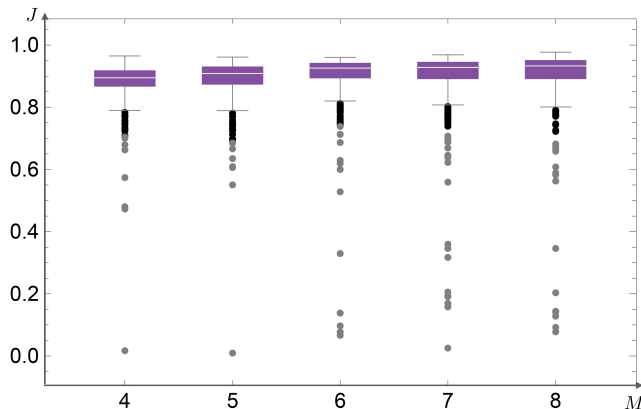


Figure 6.7: Segmentation performance on Database 1 (RGB images) according to the number M of control points. We used 3 scales and 5 harmonics.

likely to be trapped in local minima but, when Q is too large, the boundary of the object is not well-defined, which results in an inaccurate segmentation. Increasing the number of harmonics leads to a better discrimination of the two textures. However, more than 5 harmonics yields no more relevant information, resulting in decreased segmentation performance because Fisher's LDA fails to find adequate separating hyperplanes in spaces with too many dimensions.

In a second experiment, Database 2 was segmented using various values of P and a fixed Q equal to 3. The results are shown in Figure 6.10. It can be observed that the accuracy improves from $P = 1$ to $P = 5$, which is even more remarkable on grayscale images. Then, the accuracy plateaus and decreases. To conclude, the combination of 5 harmonics with 3 scales provides enough information to discriminate the textures while preserving an accurate segmentation. Hence, $P = 5$ and $Q = 3$ were fixed in all following experiments.

Dependence on Initialization: An important aspect is the initial position from which the snake is optimized. Circular shapes for closed snakes are common initial

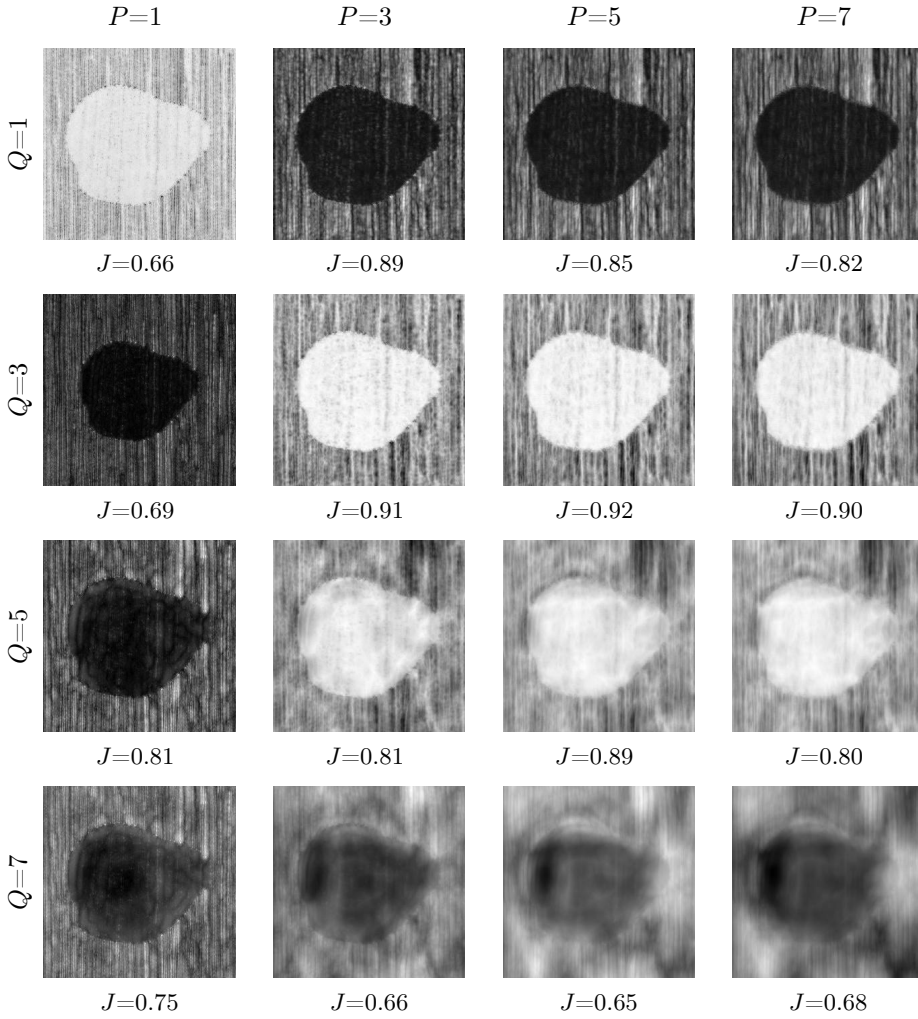


Figure 6.8: Illustrations of f_{snake} for $P = 1, 3, 5, 7$, and $Q = 1, 3, 5, 7$.

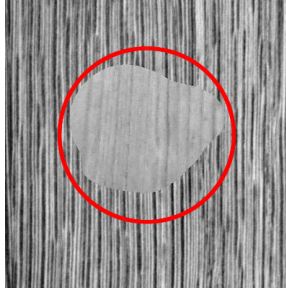

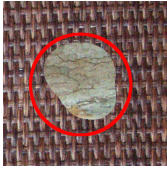

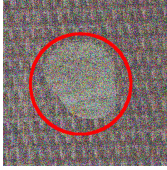
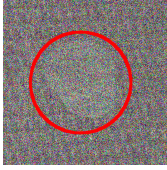
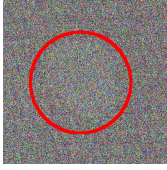
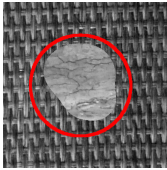
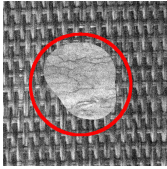
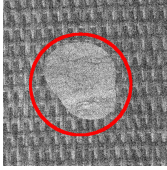
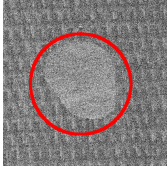
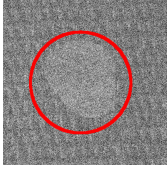
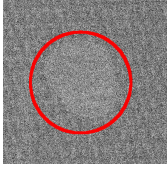


Figure 6.9: Initialization on the original image, $J = 0.62$.

contours. We segmented Database 1 using the five initializations shown in Figure 6.11. The corresponding Jaccard indices are reported in Figure 6.12. The best accuracy is obtained for the first two initializations. In fact, the energies E_n , $n \in \{0, \dots, N\}$, given in (5.4.6), are sensitive to the image contrast between the core and the shell of the snake. Hence, the snake should be initialized such that the core intersects the object of interest and the shell intersects the background.

Robustness with Respect to Noise: We investigated the robustness of the texture-driven snake to noise in the image as a function of the number M of control points. We generated 100 realizations of noisy data for each one of five levels of additive white Gaussian noise. We ran the optimization process until convergence using the proposed texture-driven snake. SNR corresponding to a given noise level and median Jaccard index were computed. We used a pixelwise SNR that compares the noisy image and the ground-truth image. The results are summarized in Table 6.2. The initialization of the snake is overlaid in the thumbnails which depict the noise-corrupted images. Its overlap with the ground truth corresponds to $J = 0.55$. From the results, we observe that the texture-driven snake is robust with respect to noise since it is able to give a proper segmentation outcome even for low SNRs. This can be explained by the fact that each energy E_n in (5.4.5), for $n = 0, \dots, N$, estimates the mean intensity over regions, while Gaussian noise has zero mean. The performance of the snake decreases faster for numerous control points, where higher noise levels induce many local minima.

Table 6.2: Jaccard indices for the segmentation of noisy data on RGB (top) and grayscale (bottom) images.

							SNR = ∞ dB	SNR = 14.11dB	SNR = 6.15dB	SNR = 0.13dB	SNR = -3.39dB	SNR = -5.89dB
M												
4	0.90	0.88	0.88	0.81	0.68	0.62						
5	0.90	0.91	0.87	0.80	0.71	0.65						
6	0.96	0.92	0.88	0.78	0.69	0.64						
7	0.95	0.92	0.87	0.78	0.69	0.64						
8	0.95	0.90	0.87	0.78	0.68	0.65						
							SNR = ∞ dB	SNR = 14dB	SNR = 6.04dB	SNR = 0.02dB	SNR = -3.5dB	SNR = -6dB
M												
4	0.76	0.87	0.84	0.69	0.64	0.57						
5	0.84	0.90	0.81	0.70	0.63	0.59						
6	0.96	0.90	0.82	0.68	0.61	0.56						
7	0.95	0.90	0.81	0.70	0.63	0.58						
8	0.96	0.89	0.84	0.69	0.59	0.57						

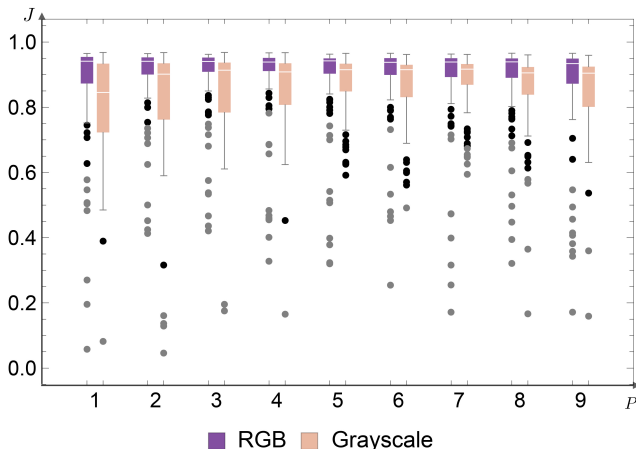


Figure 6.10: Segmentation performance on Database 2 for various numbers P of harmonics. We used 3 scales and 6 control points.

Comparisons with Existing Approaches

We carry out two experiments in which we compare the proposed texture-driven snake in term of accuracy against two segmentation methods: 1) the exponential B-spline parametric snake described in [64]. This snake has the same reproduction properties and smoothness as the proposed snake but relies on a different region-based energy (intensity information only). The implementation of this method was taken from the free open-source image-processing package Icy [11]; 2) the texture-based discrete parametric snake described in [139]. This algorithm generates texture feature maps from gray-level co-occurrence matrices (GLCM) and selects the features that are best suited using a relative standard deviation criteria. We used the implementation given in the platform MESA [169]. In the following, we refer to those methods as “intensity-based snake” and “GLCM-based snake”, respectively. Similarly to our framework, those two snakes allow for user interaction and can be trained on-the-fly. We recall that a comparison to fully automatic approaches would not be appropriate since we focus on methods that can be trained on-the-fly with one image. It is worth noting that the two competing methods assume grayscale images.

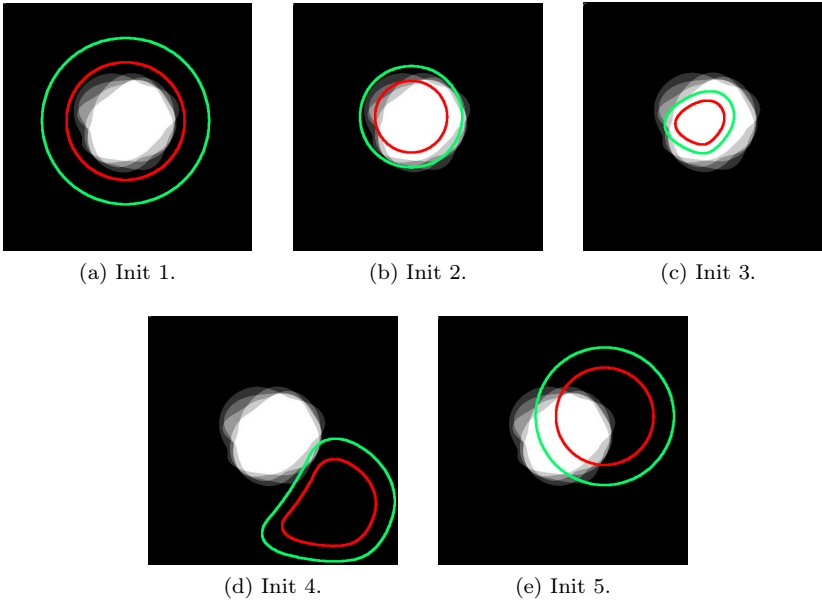


Figure 6.11: Various initializations with 6 control points. Inner red circle: snake; Outer green circle: shell. The initializations are superimposed on the image of the sum of the five masks given in Figure 6.5.

Mean and Variance Equalization: The goal of this experiment is to emphasize the importance of textural information by illustrating the limitations of the intensity-based snake, and to justify our choice to use CHWs to extract texture information. To only have texture information in the grayscale databases 1, 2, and 3, we equalized mean and variance inside and outside the mask. We optimized the intensity-based, GLCM-based and proposed snakes on each resulting database. Our snake and the intensity-based snake were initialized using 6 control points. For the proposed method we used 3 scales and 5 harmonics. For the GLCM-based snake we set the sensitivity parameter to 3 and enable the option "All angles". The corresponding Jaccard indices are

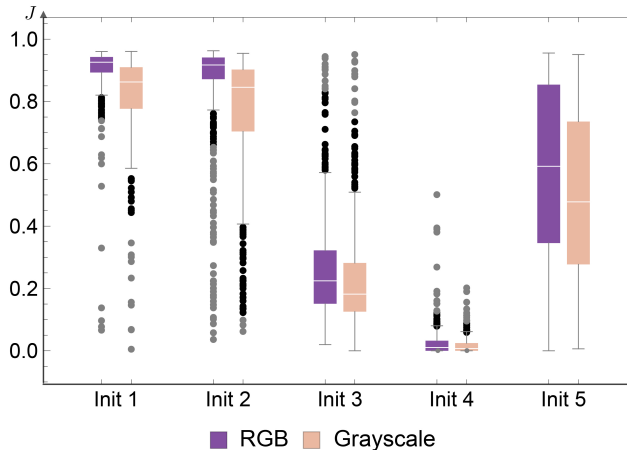


Figure 6.12: Segmentation performance on Database 1 for the initializations depicted in Figure 6.11. We used 6 control points, 3 scales, and 5 harmonics.

reported in Figure 6.13. Note that the energy of the GLCM-based snake is based on a sensitivity parameter. Therefore, if the algorithm does not sufficiently discriminate the texture of interest, the snake will spread over the entire image yielding low Jaccard indices. This explains the high standard deviations for this method in Figure 6.13. We observe that, in each case, the proposed texture-driven snake achieved an adequate segmentation of the object of interest, whereas the intensity-based snake got trapped in local energy minima due to the presence of inhomogeneous regions. Thus, the additional value of texture information is clearly observed. This is reinforced by the GLCM-based snake that yields to a higher maximum Jaccard index than the intensity-based snake on Databases 1 and 2. However, in each database, the GLCM-based snake is less accurate and robust than the proposed snake. The bad result on Database 3 could be explained by the fact that the feature selection algorithm in [169] penalizes feature maps with high relative standard deviation, which is not a true discriminative criteria when compared to Fisher's LDA.

We want to compare the effectiveness of our method when using CHWs or

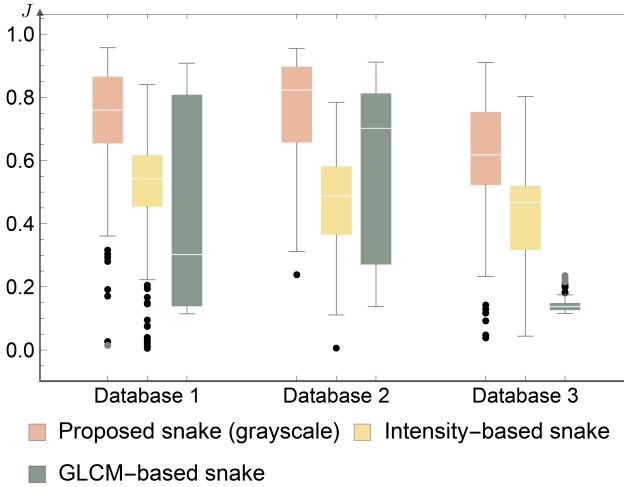


Figure 6.13: Segmentation performance for the three databases when mean and variance were equalized inside and outside the mask. For the proposed snake we used 6 control points, 3 scales, and 5 harmonics.

Gabor filters in the texture analysis. We thus repeated the experiment using the Gabor filters, described in Section 5.4.1, with 3 scales and 5 orientations. The results are given in Figure 6.14. The performances of Gabor filters and CHWs are similar on Database 1. On Database 2, Gabor filters are more efficient. This is due to the strong and constant directionality of the textures in this database (Figure 6.6 (b)), which is efficiently captured by Gabor filters because they are not invariant to image rotations. However, this lack of rotation-invariance explains that their efficiency significantly decreases on Database 3 where the flower petals have different orientations within the same texture class (Figure 6.6 (c)). This justifies our choice to use CHWs in our experiments in order to be more robust. Moreover, an advantage of our framework is that one can choose the filters that are best suited to his application.

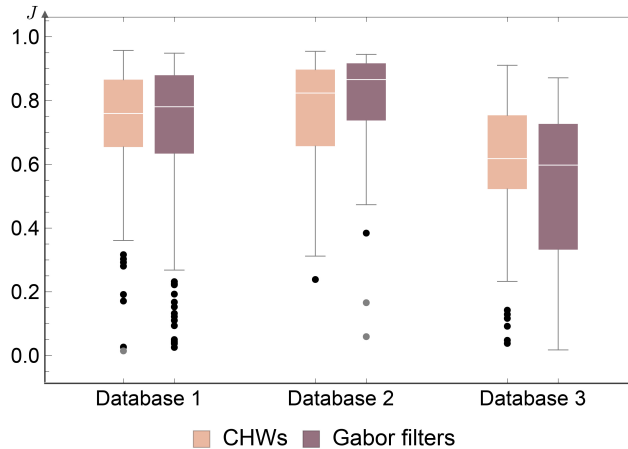


Figure 6.14: Segmentation performance obtained with the proposed snake for the three databases when mean and variance were equalized inside and outside the mask. We used 3 scales and 5 harmonics for the CHW decomposition, and 3 scales and 5 orientations for the Gabor filters.

Original Data: In this experiment, we evaluate the segmentation performance on the original databases described in Section 6.2.3. For comparison purposes, we also provide results obtained with the proposed snake when $\mathbf{w} = \mathbf{1}$ to investigate the influence of Fisher's LDA. The GLCM-based snake was initialized with a circle inside the texture of interest and we set the sensitivity parameter to 3 and enable the option "All angles". We initialized the other methods with a circle centered on the image and let them evolve automatically until convergence using 6 control points. For our snake we used 3 scales and 5 harmonics. We compared the final segmentation result to the corresponding ground truth of the synthetic data. The associated Jaccard indices are reported in Figure 6.15. Illustrations of the segmentation results are shown in Figure 6.16. We observe that, for each database, we obtain more accurate segmentation outcomes with the proposed texture-driven snake, either on RGB or grayscale images. We also remark that removing Fisher's LDA from the proposed method (*i.e.*, when $\mathbf{w} = \mathbf{1}$) significantly decreases the per-

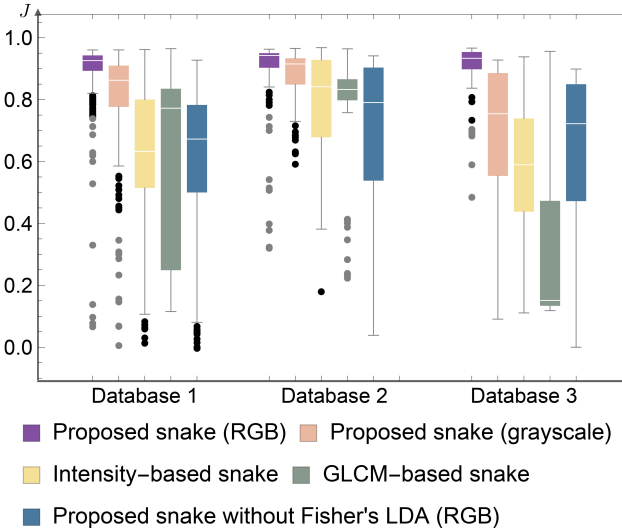


Figure 6.15: Segmentation performance for the three databases. Database 1: 450 images; Database 2 and Database 3: 100 images. For the proposed snake we used 6 control points, 3 scales, and 5 harmonics.

formances. This shows the importance of the weights \mathbf{w} and that Fisher's LDA is an adequate method to choose them. Finally, the proposed method gets better results and robustness when it is applied on RGB images rather than on grayscale images. This is striking for Database 3 and highlights the advantage of our method to be able to deal with different channels.

Real Data Scenarios

We illustrate the behavior of the proposed snake on real data scenarios. For each experiment we manually initialized the snake and let the optimization evolve until convergence for $P = 5$ and $Q = 3$. As user interaction is one of the main assets of our framework, we locally refined some segmentation outcomes by manually moving one or several control points.

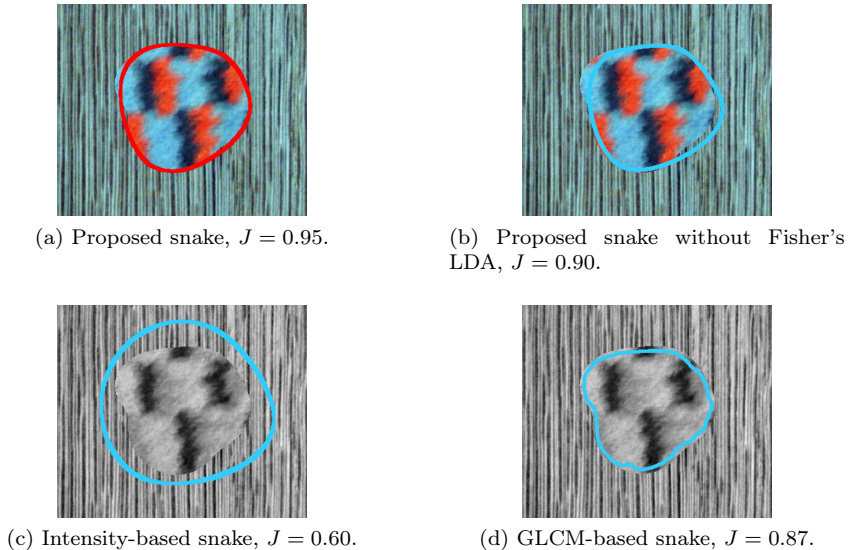


Figure 6.16: Segmentation of an image of Database 1. The intensity-based snake and GLCM-based snake are optimized on the grayscale version of the image as they can not handle several channels.

Photographic Images: We applied our snake on 4 natural photographs taken from Unsplash⁵, a website dedicated to sharing copyright-free photography. Those images are challenging as the background and the object of interest have similar color. Moreover, the illumination is not uniform which makes the texture more difficult to extract. The initializations, segmentation outcomes and manual edits are shown in Figures 6.17, 6.18, 6.19, and 6.20.

Biological Images: Texture information is also widely used to characterize biological tissues. We applied our snake to 3 microscopy images from the Cell Image Library⁶. Those images are challenging as the color inside and outside the structure to segment are similar, and they contain several textures. The initializations and segmentation outcomes are shown in Figures 6.21, 6.22,

⁵The images were taken from <https://unsplash.com/>, as of September 2018.

⁶The images were taken from <http://www.cellimagelibrary.org/>, as of September 2018.

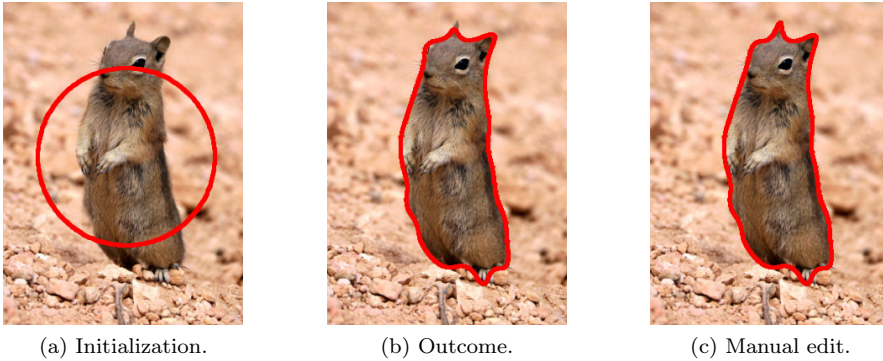


Figure 6.17: Segmentation of a squirrel. Source: J de Gier.

and 6.23. The qualitative assessment of the segmentation yields satisfactory results.

6.2.4 Conclusions

We have presented a new parametric snake that efficiently allows one to segment structures with similar intensity distribution and low contrast with the background. Our main contribution related to this work is the derivation of a new energy that combines intensity and texture information (see (5.4.5) and Section 5.4). The contribution of the two types of information is balanced using Fisher's LDA (Section 5.4.3). The method is general and any suited filter banks can be used to extract texture features. This framework is trained on-the-fly from small collections of pixels provided by the user. One main advantage of this method is that one can easily interact with the snake to edit the segmentation outcome when required. We have compared the performance of our snake to existing ones. In particular, we have observed that the texture-driven snake always performs better than classical parametric snakes that rely on intensity information only. This improvement was even more substantial when the intensity distributions are similar over the background and the object of interest. We have studied the parameter sensitivity of our proposed method as well as its robustness to noise. Finally, we have shown its practical usefulness on real images.



Figure 6.18: Segmentation of a leaf. Source: Joshua Newton.

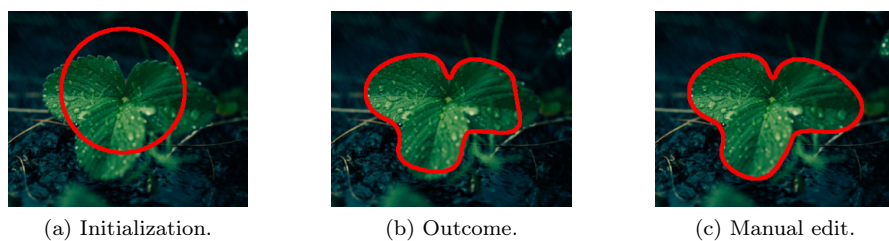


Figure 6.19: Segmentation of leaves. Source: Mikael Kristenson.



Figure 6.20: Segmentation of a mushroom. Source: Nancy Newton.

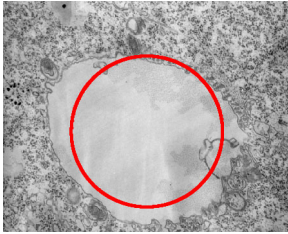


(a) Initialization.

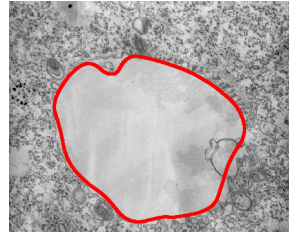


(b) Outcome.

Figure 6.21: Segmentation of a hair follicle on a light micrograph. Source: Ivor Mason, 2012, CIL:39094.

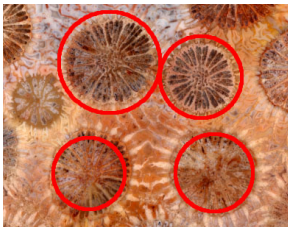


(a) Initialization.



(b) Outcome.

Figure 6.22: Segmentation of a nascent digestive vacuole on an electron microscopy image. Source: Richard Allen, University of Hawaii, 2012, CIL:39720.



(a) Initialization.



(b) Outcome.

Figure 6.23: Segmentation of a fossil of red sponge coral on a microscopy image. Source: Norm Barker, 2009, CIL:41842.

6.3 Multiresolution Subdivision Snakes

Subdivision is a powerful scheme to generate a continuously defined curve starting from an initial set of points. As it was motivated in Chapter 4, this geometric representation combines the advantages of parametric and point snakes. Moreover, its discrete nature and multiresolution property are particularly relevant for the design of active contours.

The use of subdivisions for the construction of segmentation models was pioneered by [170] and [171] for Doo-Sabin surfaces [132] and the DLG-scheme [172], respectively. In the first case, left ventricles are modeled whereas, in the second case, they improved editing semantics of traditional snakes.

In this section, we propose⁷ a general approach that remains valid for any subdivision scheme as we derive the construction of a 2D subdivision snake in a generic way. The main contributions related to this work are: 1) a new geometrical representation based on subdivision. A crucial aspect is the choice of the subdivision mask that determines important properties of the model such as its approximation properties, the capability of reproducing circular, elliptical, or polynomial shapes [123], as well as the possibility of being interpolatory [173, 174] or not; 2) the derivation of associated energy functions; 3) the presentation of an integrated strategy where the snake is optimized in a coarse-to-fine fashion. This multiscale approach is algorithmic and inherently recursive: We increase the number of points describing the curve as the algorithm progresses to the solution; at each step, the scale of the image feature (on which the optimization is performed) is matched to the density of the point cloud. This speeds up the computation and increases the robustness.

We give several examples of explicit constructions of subdivision snakes. We illustrate their use on real images as well as on test data simulating real biological conditions. We compare our proposed model to existing parametric snakes and measure its robustness and accuracy with respect to noise and initialization. Specifically, we show that the proposed coarse-to-fine approach allows the optimizer to 1) have a larger basin of attraction which makes it robust to initial conditions; 2) escape some local optima; 3) be efficient by progressively increasing the snake resolution; 4) delineate structures of different sizes contained within an image without

⁷This section is based on our publication [70], in collaboration with D. Schmitter, V. Uhlmann and M. Unser. A demo of the corresponding plugin and related documentation are available at <http://bigwww.epfl.ch/demo/subdivision-snake/>.

having to adapt the initialization.

6.3.1 Framework

We implicitly describe the contour of our snake by the continuously defined limit curve \mathbf{r} of a convergent subdivision scheme given by (4.2.6) and (4.2.3). Its shape is encoded by the M control points $\Theta = \{\mathbf{p}_{(0)}[m]\}_{m \in \{0, \dots, M-1\}}$. This representation implies that the properties of the snake (*e.g.*, smoothness, shape reproducibility) are determined by the choice of the subdivision mask h . We shown in Section 4.2.3 that affine invariance is ensured for every convergent subdivision scheme.

For the snake energy, we use a combination of the edge- and region-based terms (5.1.4) and (5.2.5) such as

$$E_{\text{snake}}(\mathbf{p}_{(k)}(\Theta)) = b E_{\text{edgeSD}}(\mathbf{p}_{(k)}(\Theta)) + (1 - b) E_{\text{intensitySD}}(\mathbf{p}_{(k)}(\Theta)), \quad (6.3.1)$$

where $b \in [0, 1]$, is a tradeoff parameter that balances the contribution of the two energies, and $\mathbf{p}_{(k)}$ describes the contour of the snake and is given by (4.2.3).

6.3.2 Optimization: a Multiresolution Approach

The segmentation outcome, when using active-contour models, depends on the initialization of the snake. A larger basin of attraction allows for a rougher initialization. With common singleresolution segmentation algorithms, a tradeoff has to be made between the desired accuracy and the amount of blurring one applies to an image. Blurring enlarges the basin of attraction but also decreases the resolution of an object, which in turn affects the quality of the delineation. Multiresolution approaches are powerful methods to speed up the optimization process and improve robustness. Existing methods mainly rely on the construction of an image pyramid, where the active contour is upsampled from a coarse scale to a finer scale of the image [175–177]. One limitation of those methods is that the object to segment may not have the same topology on the coarsest and finest images. In this section, we present a multiresolution approach which is inherent to the iterative process of subdivisions. The subdivision snake has the advantage that the resolution of the representation can be adapted to the resolution of the object to be segmented. The number of subdivision points used to describe the snake and to determine its energies (5.1.4) and (5.2.5) is controlled by the number k of subdivisions. If fewer

points are used, the optimization is faster. We exploit this multiresolution property both to enlarge the basin of attraction and to accelerate the optimization.

Algorithm: We apply K successive lowpass filters G_k to the original image to obtain K smoothed images f_k . The snake is first optimized on the coarsest image f_1 that corresponds to the lowest resolution and, hence, the structure of interest only contains few details. The initialization on f_1 can be very rough because the blurring enlarges the basin of attraction. The snake is optimized on f_1 and is then used as initialization at the next resolution level on f_2 . The process continues until the optimization reaches the finest resolution level that corresponds to the original image f . Because the smoothed images contain fewer details and less noise than the original one the snake is more robust to initial conditions. The subdivision scheme allows us to adapt the number of subdivision points describing the curve \mathbf{r} to the level of detail in the image. Thus, we start with few subdivision points (*i.e.*, one subdivision step), which allows for fast optimization. At each subsequent iteration of the multiresolution algorithm, we keep constant the number of control points and increase the density of the subdivision points. The pseudo-code in Algorithm 6.3 describes this algorithm. Note that the position of the control points $\mathbf{p}_{(0)}$ changes after each optimization. We denote by $\mathbf{p}_{(0)_{\text{opt}}}$ the sequence describing the optimized control points. From now on, we denote by $E_{\text{snake}}(f, \mathbf{p}_{(k)}(\Theta))$ the energy of the snake as it also depends on the image on which it is optimized. The images f_k and their pre-integrated versions (5.2.3) are pre-computed, which accelerates the segmentation process and decreases the memory requirements.

6.3.3 Experiments and Validation

We compare the proposed multiresolution snake to parametric singleresolution snakes [50]. We first test the robustness with respect to initial conditions and, in a second step, we measure its robustness with respect to noise as well as its ability to segment objects of varying sizes in an image. Finally, we illustrate applications on real data where the ground truth is not available.

ALGORITHM 6.3: Multiresolution algorithm.

Input: original image f , low-resolution snake encoded by $\mathbf{p}_{(0)}$

Initialization: $\mathbf{p}_{(1)} = \mathbf{p}_{(0)\uparrow_2} * h$

For K iterations ($K \geq 1$) over k :

compute: image $f_k = f * G_k$

optimize: $\mathbf{p}_{(0)\text{opt}} = \arg \min_{\mathbf{P}_{(0)\text{opt}}} E_{\text{snake}} \left(f_k, \mathbf{P}_{(k)}(\mathbf{P}_{(0)\text{opt}}) \right)$

$\mathbf{P}_{(k)\text{opt}} = \mathbf{P}_{(0)\text{opt}\uparrow_{2^k}} * h_{0 \rightarrow k}$

increase the resolution of the snake: $\mathbf{p}_{(k+1)} = \mathbf{p}_{(k)\text{opt}\uparrow_2} * h$

Until: high-resolution segmentation on the original image f

Accuracy and Robustness to Initial Conditions

We carry out two experiments in which we compare our multiresolution subdivision snake to a parametric singleresolution snake based on quadratic B-splines as described in [50]. In order to compare snakes with the same reproduction properties, we construct our subdivision snake with a minimum-support subdivision scheme that generates polynomials of degree up to 2 (see Section 4.3.3).

In the first experiment, we test the accuracy of the segmentation. We created a test image of (854×768) pixels that simulates realistic conditions in fluorescence microscopy (Figure 6.24 (second row)), including noise. It shows a rod-shaped cell representative of a *Schizosaccharomyces pombe* (*S. pombe*) [178]. We then blurred the test image with five Gaussian kernels having different standard deviations σ whose values are given in Table 6.4. Four resulting images are shown in Figure 6.24 (first row). The higher the standard deviation, the fewer details are present in the filtered image. The initialization of the snakes was drawn manually with $M = 8$ control points (Figure 6.24 (second row)). Its overlap with the actual structure corresponds to the Jaccard index $J = 0.544$. We first optimized our subdivision snake using the multiresolution algorithm described in Section 6.3.2. At each it-

eration we did one subdivision step corresponding to a multiplication by a factor of 2, starting with $2M = 16$ subdivision points. The curve evolves guided by the edge-based energy E_{edgeSD} , *i.e.*, we choose $b = 1$ in (6.3.1). The optimized contours at different levels of the multiresolution algorithm are shown in Figure 6.24 (first row). We compared the final segmentation to the ground truth of the synthetic data; the corresponding Jaccard index is given in Table 6.4. We consider that a snake succeeds in segmenting the structure of interest if $J \geq 0.95$. We then independently optimized the parametric singleresolution snake with the edge energy E_{edge} , given by (2.2.2), on the six images (the five blurred images and the original one) using the same initialization. Results are shown in Figure 6.24 (third row) and the corresponding Jaccard indices are given in Table 6.4. The segmentation succeeded only on the smoothed image corresponding to $\sigma = 8$. The singleresolution snake is able to segment the structure of interest only on a smoothed image because the basin of attraction is too narrow otherwise for the edge energy. The variance of the Gaussian filter has to be well-chosen according to the initialization. We conclude that the multiresolution approach improves the accuracy of the segmentation. This result is explained by the fact that the multiresolution is initialized on the coarsest image with reduced details and a large basin of attraction. By adapting the resolution of the subdivision snake to the image details, it is able to converge to the structure to segment on the original image.

In the second experiment, we evaluate the impact of the multiresolution approach on the robustness of the snake with respect to the initialization. For this experiment, we generated another test image of (854×768) pixels of a sickle cell [179] acquired through fluorescence microscopy (Figure 6.25 (d)). We compared the basin of attraction of both the singleresolution and the multiresolution subdivision snakes using $M = 6$ control points. Each basin of attraction was computed as follows: a rough approximation of the goldstandard was constructed. This shape was rescaled to construct several initial positions of the snake. We then optimized the active contour using an edge-based energy (*i.e.*, E_{edge} and E_{edgeSD}). For each segmentation result, we computed the Jaccard index and associated a grayscale value to J where white corresponds to $J = 0$ and black to $J = 1$. Finally, we generated an image where each initialization was drawn with the color corresponding to the Jaccard index of the corresponding segmentation result. For the singleresolution snake, we realized this experiments on two images: the original one and a smoothed version with $\sigma = 10$. The results are shown in Figure 6.25 (a) and (b), respectively. For our subdivision snake, we used the multiresolution approach on the original image.

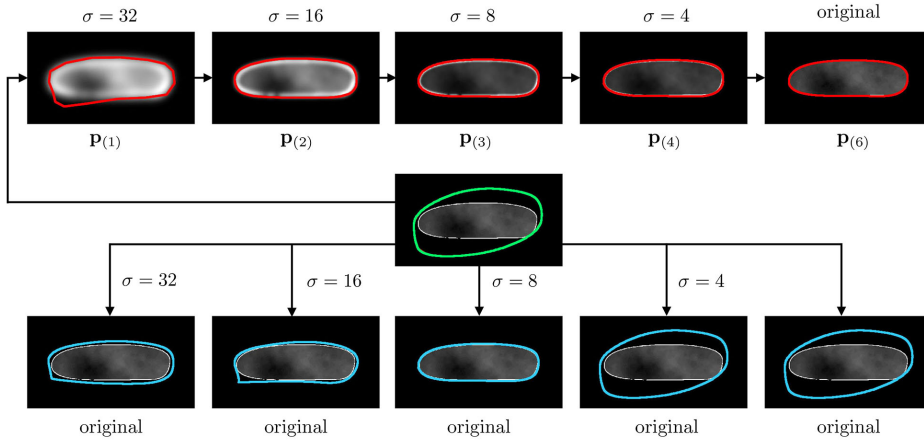


Figure 6.24: Comparison of the accuracy of the segmentation between our multiresolution subdivision snake and the parametric singleresolution snake. Both snakes generate polynomials of degree up to 2. First row: Evolution of the subdivision snake during the six-level multiresolution process. The last illustration shows the final segmentation on the original image. Second row: Initialization. Third row: Several segmentation results obtained with the parametric singleresolution snake for different blurred versions of the original test image.

The result is given in Figure 6.25 (c). The white regions in the images showing the basin of attraction correspond to positions that were not considered for initialization, including the boundary of the shape to segment. Note that the average Jaccard values inside the shape to segment appear to be less uniform than outside. This can be attributed to the two following reasons: First, as seen on the original image (Figure 6.25 (d)), the outside of the shape is completely uniform in intensity while the inside of the shape exhibits variations in pixel values. Snakes evolving from outside of the object therefore encounter no risk of getting diverted from their target due to variations of pixel intensities. Snakes which start to deform from the inside of the shape are, however, evolving on a nonuniform region and are more likely to get trapped into local energy minima. Second, for a given number of control points, smaller snakes tend to diverge more easily than larger ones. This effect is simply due to the fact that, if their number is fixed, control points are physically

Table 6.4: Jaccard indices for segmentation obtained with the singleresolution and subdivision snakes, both generating polynomials of degree up to 2.

	σ [pixel]	J	Result
Singularresolution snake	32	0.803	fail
	16	0.860	fail
	8	0.950	succeed
	4	0.544	fail
	2	0.544	fail
	0	0.544	fail
Subdivision snake	0	0.989	succeed

closer in smaller shapes. During the optimization process and as the control points are moved, it becomes therefore more likely for the snake to get entangled. In the present experiment, initial shapes inside the object to segment are smaller than the ones outside the object, and optimization results tend to get more unstable due to the enhanced risk of entanglement. We observe that the singleresolution snake is very sensitive to the initialization. On the contrary, the subdivision snake leads to accurate segmentation even for initializations far from the object to segment.

Robustness with Respect to Noise

As further test of robustness, we performed segmentation on the test image described in Figure 6.24 (second row) with different levels of additive white Gaussian noise. We still used the multiresolution subdivision snake constructed with the minimum-support subdivision scheme that generates polynomials of degree up to 2 (see Section 4.3.3) and $M = 8$ control points. The initial overlap of our snake with the ground truth corresponds to $J = 0.593$ (Figure 6.26 (e)). SNRs corresponding to a given noise level and associated Jaccard indices were computed. We used a pixelwise SNR that compares the noisy image and the ground truth image. The results are summarized in Table 6.5 and Figure 6.26. For all cases, we obtained $J > 0.95$.

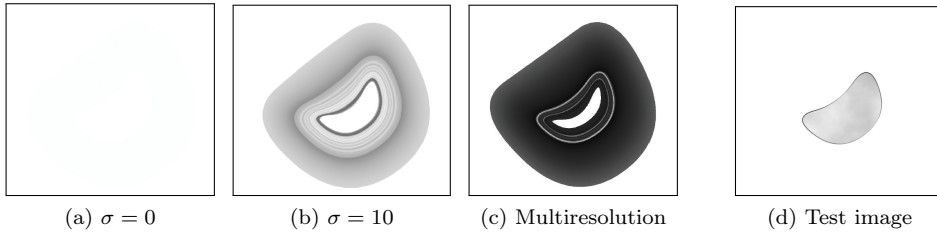


Figure 6.25: Comparison of the basin of attraction of the multiresolution subdivision snake versus the parametric singleresolution snake using an edge energy. (a)-(b) Basins of attraction of the singleresolution snake obtained for the original image and for a blurred version ($\sigma = 10$); (c) basin of attraction of the multiresolution subdivision snake obtained on the original image (d).

Table 6.5: Jaccard indices for the segmentation of noisy data.

SNR [dB]	J
-7.83	0.990
-13.80	0.987
-16.60	0.987
-17.82	0.984

Segmentation of Objects of Varying Sizes

The multiresolution algorithm for segmentation presented in Section 6.3.2 suggests that the approach is very robust to initialization. To verify this property, we created a test image of size $(5,500 \times 2,700)$ pixels composed of eight circular cells of different sizes (Figure 6.27). The initializations correspond to circles with a radius of 461 pixels centered in each cell (Figure 6.27 (a)). By adjusting the variance of the lowpass filters to the smallest structure present in the image, we were able to segment all the cells. We used a multiresolution subdivision snake based on the ellipse-reproducing Deslauriers-Dubuc scheme defined by (4.4.8). Results are shown

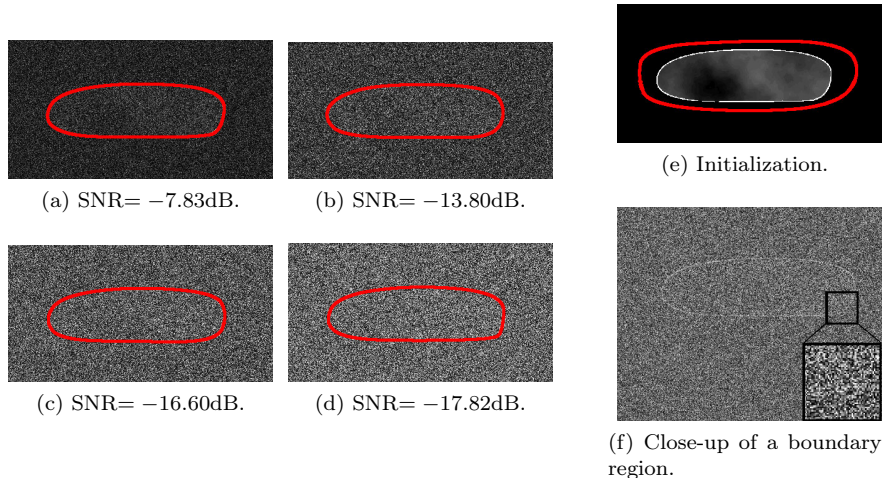


Figure 6.26: Robustness with respect to noise of the multiresolution subdivision snake. A close-up of a boundary region between the test rod-shape and its background for $\text{SNR} = -17.82\text{dB}$ is depicted in (f).

in Figure 6.27 (b) and the corresponding Jaccard indices are presented in Table 6.6 (first line). Each structure was accurately segmented with $J \geq 0.95$.

Real Data

We illustrate the behavior of the proposed snake on real data. In this context, the ground truth is unknown and we have to rely on qualitative assessments to validate the accuracy of the segmentation. We applied our multiresolution subdivision snake, constructed with the non-stationary minimum-support subdivision scheme that generates ellipses defined by (4.4.10), to four microscopic images (Figure 6.28). These images are challenging because of the presence of noise and of objects with different sizes. Moreover, shapes can be close to each other. They represent elliptic cells (Figure 6.28 (a), inverted contrast), rod-shaped cells of *S. pombe* (Figure 6.28 (b)), a sickle cell (Figure 6.28 (c)), and circular cells (Figure 6.28 (d)). The qualitative assessment of the segmentation yields satisfactory results. We used $b = 0.5$

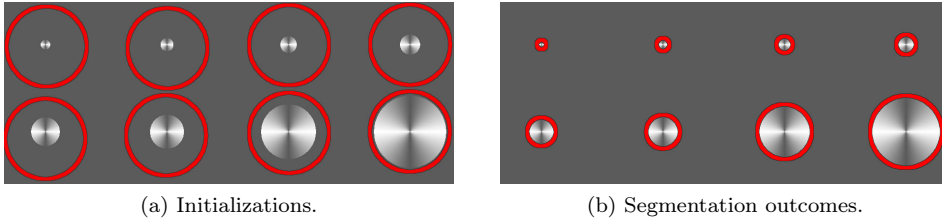


Figure 6.27: Segmentation of circles with different radii using the ellipse-reproducing Deslauriers-Dubuc subdivision scheme defined by (4.4.8).

Table 6.6: Jaccard indices for the segmentation of circles of various sizes obtained with the stationary and the non-stationary Deslauriers-Dubuc schemes.

Radius [pixels]	55	75	95	115	165	195	315	415
Reproducing scheme								
Ellipses	0.992	0.994	0.995	0.996	0.996	0.996	0.997	0.998
Polynomials of degree up to 3	0.810	0.778	0.764	0.772	0.756	0.777	0.771	0.765

in (6.3.1) and the average time to delineate one cell was less than 0.2 seconds on a 1.7 GHZ processor with 8 GB RAM.

Note that, as the principal motivation for our work is the segmentation of biological images, it was important through those experiments to show that our model can reproduce circular or elliptic shapes. However, the reproduction properties of the presented schemes are not restricted to those shapes. More complex shapes can be segmented by increasing the number of control points.

6.3.4 Discussion

Guidelines for the Choice of the Subdivision Scheme

In Chapter 4, we presented two main families of subdivision schemes: the Deslauriers-Dubuc and the minimum support subdivision schemes. We highlighted their prop-

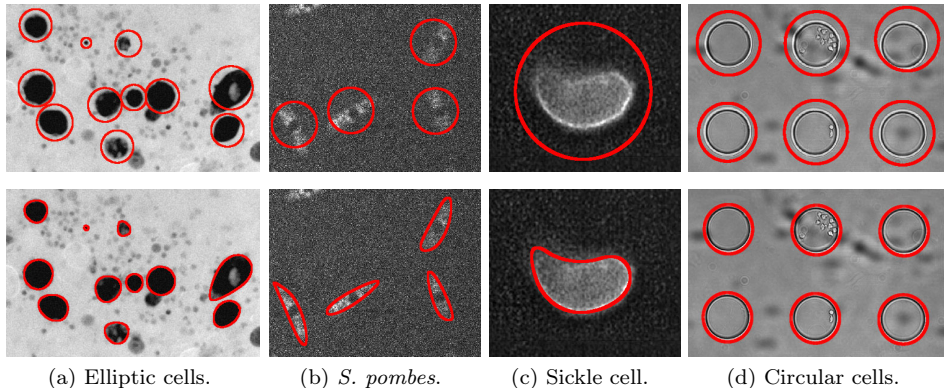
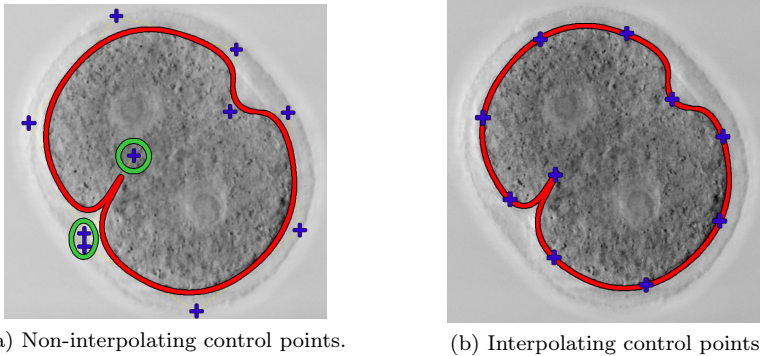


Figure 6.28: Segmentation of real data using multiresolution subdivision snakes constructed with the non-stationary minimum-support subdivision scheme that generates ellipses (see (4.4.10)). First row: snake initializations. Second row: segmentation outcomes.

erties in case of stationary or non-stationary approaches (see Sections 4.3 and 4.4, respectively). Here after, we discuss the choice of the subdivision mask to construct our snake according to the application.

Minimum-Support vs. Deslauriers-Dubuc Subdivision Schemes: The computation of the snake energy and the speed of the optimization algorithm is related to the length of the support N of the subdivision mask. More precisely, the complexity when calculating the subdivision points (4.2.3) is $\mathcal{O}((N-1)^k)$. Therefore, the fastest algorithm is obtained using minimum-support subdivision schemes. In return, the Deslauriers-Dubuc subdivision is interpolating. This can be an advantage if user interaction is involved, because it facilitates the editing of the curve. We present in Figure 6.29 an intermediate stage in the segmentation of a dividing cell. User interaction makes it possible to improve the result by moving the control points. However, the interaction is more intuitive when they lie on the curve (Figure 6.29 (b)). Otherwise, it is difficult to know which parameter controls the part of the curve that has to be modified (Figure 6.29 (a)).



(a) Non-interpolating control points.

(b) Interpolating control points.

Figure 6.29: User-friendly interaction according to the interpolation property of the subdivision scheme. Blue crosses: control points; red curve: snake; green circles: control points for which it is difficult to know which part of the curve they control. Source: <http://www.cellimagelibrary.org/images/35450/>.

The choice of the subdivision mask ultimately depends on the application: for an automatic method, we suggest to use a minimum-support subdivision scheme; whereas, when one would like to benefit from friendly user interactions, it is preferable to use a Deslauriers-Dubuc subdivision scheme.

Stationary vs. Non-Stationary Subdivision Schemes: Non-stationary subdivision schemes are somewhat more complicated than stationary ones because the subdivision mask is different at each iteration. Their main advantage lies in their capability to reproduce cosine and sinus, which allows for an efficient construction of ellipses and circles. In biomedical imaging, circular or elliptic structures are often encountered. It is therefore desirable for the snake to be able to reproduce these shapes. The non-stationary schemes presented in Sections 4.4.3 and 4.4.4 reproduce ellipses with the minimum number of control points $M = 3$, whereas the reproduction is only approximated with the stationary schemes for $M < +\infty$. As the speed of the algorithm scales with the number of control points, it is preferable to use a non-stationary subdivision scheme with few control points to segment elliptic structures. To illustrate

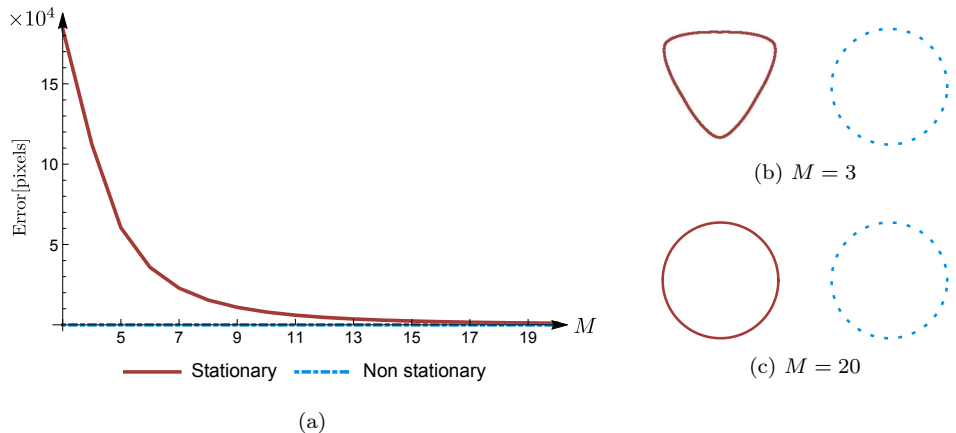


Figure 6.30: Approximation of trigonometric curves with the stationary (red solid line) and non-stationary (blue dashed line) Deslauriers-Dubuc subdivision schemes. (a) Evolution of the approximation error as a function of the control points. Approximated ellipses for $M = 3$ (b) and $M = 20$ (c) are given for each scheme.

this property, we computed the error when approximating a circle as a function of M with the stationary Deslauriers-Dubuc subdivision scheme defined by (4.3.5). In Figure 6.30, we see that the error decreases as M increases. However, a large number of control points is needed to obtain an acceptable error. Therefore, the segmentation of circular shapes with a small number of control points $\mathbf{p}_{(0)}$ is more accurate with a non-stationary scheme. To highlight this property, we performed the same experiment as the one presented in Figure 6.27, using the stationary Deslauriers-Dubuc scheme that reproduces polynomials of degree up to 3 (see (4.3.5)). We used the same initializations and $M = 4$ control points. The results are shown in Figure 6.31. We computed the Jaccard indices and compared them to the ones obtained previously with the non-stationary scheme (see Table 6.6). All the Jaccard indices are worse than 0.95, which is due to the fact that the stationary scheme does not approximate well circles for $M = 4$.

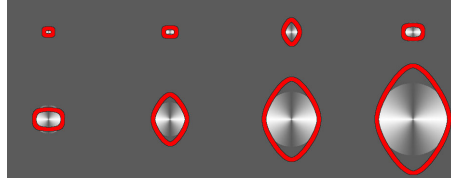


Figure 6.31: Segmentation of circles of different sizes obtained with the multiresolution subdivision snake based on the stationary Deslauriers-Dubuc scheme that reproduces polynomials of degree up to 3 (see (4.3.5)).

Table 6.7: Properties of the different subdivision schemes.

Subdivision scheme	Ellipse reproduction	Interpolant	Shortest mask
Minimum-support (MS)	No	No	Yes
Non-stationary MS	Yes	No	Yes
Deslauriers-Dubuc (DD)	No	Yes	No
Non-stationary DD	Yes	Yes	No

To conclude, if the structure of interest has many details, that requires a high number of control points, then we suggest the use of stationary schemes, thereby privileging computation simplicity while preserving the accuracy of the result; otherwise, one should use a non-stationary scheme.

Summary: The properties and advantages of each subdivision scheme presented in Sections 4.3 and 4.4 are summarized in Table 6.7. As in biomedical imaging we often deal with elliptic structures and that biologists may need to interact with the segmentation result, we recommend the use of non-stationary Deslauriers-Dubuc subdivision schemes.

Table 6.8: Parameters of the multiresolution algorithm.

Level Parameter	1	2	3	4	5	6
σ	32	16	8	4	2	0
$\mathbf{P}^{(k)}$	$\mathbf{P}^{(1)}$	$\mathbf{P}^{(2)}$	$\mathbf{P}^{(3)}$	$\mathbf{P}^{(4)}$	$\mathbf{P}^{(5)}$	$\mathbf{P}^{(6)}$

Choice of the Multiresolution Parameters in Practice

Regarding the variance and the number of subdivision steps at each level of the multiresolution algorithm, we found in practice that six subdivision iterations are enough to obtain satisfactory convergence. At each resolution level, we compute one subdivision step, so that the samples of the curve are upsampled by a factor 2. As smoothing is equivalent to a downsampling operation, we obtain the variance of the coarser lowpass filter by decreasing the resolution of the original image by a factor 2 at each iteration. Hence, we propose a multiresolution algorithm with 6 levels where the first one is characterized by $\sigma = 2^5$ and $2M$ subdivision points. At each iteration, the value of σ is divided by two and one subdivision step is performed. The value of the parameters at each step are summarized in Table 6.8. The choice of these parameters holds when the snake is initialized far from the object to segment. Otherwise, a smaller variance can be used for the coarsest lowpass filter but the convergence of the subdivision scheme is still required on the finest level.

6.3.5 Conclusions

We have presented the 2D generic construction of multiresolution snakes based on subdivision. The snakes approximate closed curves with arbitrary precision by iteratively refining a set of control points. We have provided several examples of explicit constructions of such snakes and discussed their properties. We have shown how they should be chosen according to desired properties that depend on the structures to be segmented. We have also proposed a multiresolution algorithm to adapt the resolution of the curve to the level of detail in the image. We have compared our framework to traditional parametric singleresolution snakes and shown that our snakes have a larger basin of attraction, which means that they are more robust

with respect to initial conditions. Furthermore, the multiresolution property accelerates the optimization. We have validated our snakes on test data as well as on real bioimages. The primary contributions related to this work are:

- A new geometrical representation based on subdivision;
- The derivation of associated energy functions such as the region- and edge-based terms (5.1.4) and (5.2.5), respectively;
- The presentation and integration of an algorithmic coarse-to-fine optimization strategy (see Section 6.3.2).

6.4 Active Tessellations

In this section, we address⁸ the segmentation of the cell aggregates that appear in images of several biological specimen such as *C. elegans* embryo or cornea endothelium (Figures 6.32 and 6.38). This task is challenging because of the proximity of the cells and the presence of gaps in the membranes.

Over the past decade, automated methods were proposed for the segmentation of cell membranes, including intensity thresholding, morphological operations, Voronoi-based methods [180], labeling procedures [181], or watershed transform [28–32]. Those methods have three main limitations. First, they suffer from leakage in case of dimmed membranes. Second, they are sensitive to noise and might result in over-segmentation. Third, it is not easy to introduce prior knowledge to improve the accuracy of the segmentation [182]. Recent approaches are based on the detection of membrane patterns coupled with graph-cut [183], or on deep learning [184]. If these methods are better suited for incorporating prior knowledge, they do not allow for easy and user-friendly interaction. Moreover, they may provide non-continuous cell boundaries, which complicates the extraction of quantitative measurements. Topology adaptive methods, such as level sets or T-snakes [60, 185], are not required in this context as the topological structure of the cell aggregate is generally known.

As it was exposed in Chapter 2, parametric snakes are built to ensure continuity and smoothness, which prevents leakage, and they are encoded by only a few control points, which results in fast optimization and provides robustness to noise. Moreover, the underlying shape has a continuous representation in terms of basis functions, which facilitates the incorporation of prior knowledge. However, parametric snakes are not well suited to segment objects that are close to each other: They might yield overlapping segmentations (Figure 6.32 (c)) as the snakes are optimized independently. In the literature, only few works regarding active contours address the segmentation of touching objects. An extension of traditional parametric snakes to track non-occluding objects that transiently touch each other is presented in [42, 186]. These methods could only segment cell aggregates with thick membranes or non-touching cells. Networks of active contours were introduced in [187] and [188]. However, they involve many parameters (nodal points).

⁸This section is based on our publication [73], in collaboration with D. Sage, A. Galan and M. Unser. A demo of the corresponding plugin and related documentation are available at <http://bigwww.epfl.ch/demo/active-tessellation/>.

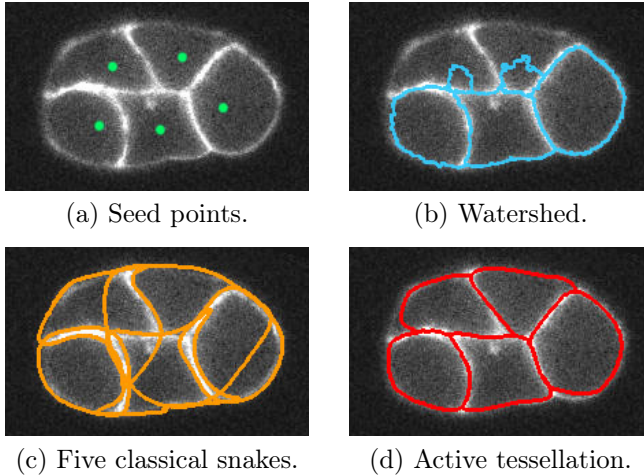


Figure 6.32: Segmentation of a *C. elegans* embryo; Source: R. Jankele and P. Gönczy, EPFL. (a) Seed points used for the initialization of each method; (b) watershed method; (c) five classical snakes [64]; (d) active tessellation.

In this section, we propose a new active contour with a geometric representation that keeps the advantages of parametric snakes while addressing globally the segmentation problem of cell aggregates. The model consists in a smooth tessellation, so-called *active tessellation*, that is globally deformed towards the cell membranes through the minimization of a suitable ridge-based energy. The smooth tessellation is encoded by a set of control points and generated through a subdivision scheme. By construction, the segmented tiles are non-overlapping and the tessellation structure bridges membrane gaps. After optimization, each cell of the aggregate can be individually extracted to compute statistical descriptors. We illustrate the benefits of the proposed active tessellation on real biological applications in the context of semi-interactive segmentation.

6.4.1 Framework

Our active contour model is a smooth tessellation, as described in Figure 6.33 (e). Its shape is parametrized by control points that are grouped in tiles (Figure 6.33

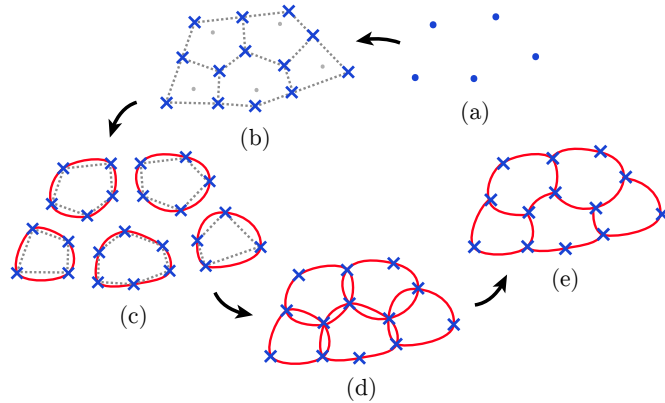


Figure 6.33: Construction of the smooth tessellation. Given seed points (a), we first generate control points (blue crosses) grouped in tiles (b). Then, we specify the smooth boundary of each tile with a continuous closed curve by applying a subdivision scheme to its control points (c). Finally, we keep one of the common edges (d) to obtain the final smooth tessellation (e).

(b)). Each tile is associated with a closed curve that specifies its smooth boundary (Figure 6.33 (c)). This smooth curve is generated from the control points via a subdivision scheme. The smooth tessellation is the union of the closed curve of each tile. As two control points that belong to two adjacent tiles are connected by two continuous edges (Figure 6.33 (d)), we keep only the edge of the largest tile (Figure 6.33 (e)). The remaining edges are then optimized by fitting them to the image data via energy optimization using the ridge energy presented in Section 5.3.

A Tiling Made of Closed Subdivision Curves

We use the subdivision process presented in Chapter 4 to generate the closed curve of each tile. In the present case, $\mathbf{p}_{(0)}$ is the M -periodic sequence that contains the control points of a tile. We apply the recursive refinement (4.2.3) to obtain the $(2^k M)$ -periodic sequence of subdivision points $\mathbf{p}_{(k)}$, such that the contour points are sufficiently dense. For the subdivision mask h we use the convergent Deslauriers-

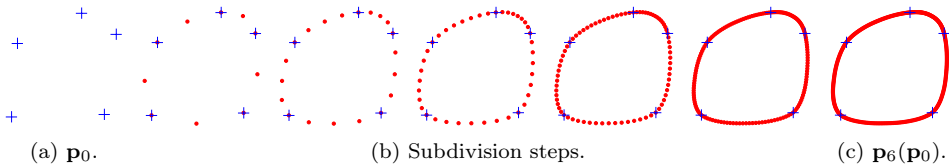


Figure 6.34: Interpolating Deslauriers-Dubuc subdivision scheme. (a) Control points \mathbf{p}_0 ; (b) subdivision steps that converge to the continuous curve (c) which is encoded by the five control points \mathbf{p}_0 (blue crosses).

Dubuc subdivision scheme defined by (4.3.5) which generates C^1 -continuous functions and reproduces polynomials up to degree 3. We choose this scheme as it has the advantages to be interpolating, which facilitates the editing of the curve, and affine invariant. Moreover, it ensures smoothness as it produces C^1 -continuous curve. In Figure 6.34, we illustrate the generation of a tile using the interpolating Delauriers-Dubuc subdivision scheme.

An Oriented Ridge-based Energy

To attract the smooth tessellation towards the cell membranes, we use the discrete formulation of the oriented ridge-based energy (5.3.4), that is $E_{\text{snake}} = E_{\text{ridgeSD}}$. Let Θ be the control points of the active tessellation. They generate a set of N points $\mathbf{p}_{\text{tess}} := \mathbf{p}_{\text{tess}}(\Theta)$ that delineates the smooth tessellation. Then, the ridge-based energy (5.3.4) can be re-expressed as

$$E_{\text{ridgeSD}}(\mathbf{p}_{\text{tess}}(\Theta)) = -\frac{1}{N} \sum_{m=0}^{N-1} \xi(\mathbf{p}_{\text{tess}}[m]) \frac{|\langle \mathbf{v}_{\min}(\mathbf{p}_{\text{tess}}[m]), \mathbf{n}(\mathbf{p}_{\text{tess}}[m]) \rangle|}{\|\mathbf{v}_{\min}(\mathbf{p}_{\text{tess}}[m])\|}, \quad (6.4.1)$$

where $\mathbf{n}(\mathbf{p}_{\text{tess}}[m])$, $\xi(\mathbf{p}_{\text{tess}}[m])$ and $\mathbf{v}_{\min}(\mathbf{p}_{\text{tess}}[m])$ are the approximation of the unit normal vector defined by (5.1.5), the ridge strength and the eigenvector of the Hessian matrix of the image, respectively, at the m th point of the tessellation. This energy is minimal when the active tessellation lies on the ridge.

6.4.2 Implementation Details

We initialize the active tessellation as follows: We construct a Voronoi diagram around seed points and then apply a mask to shrink the Voronoi tiles in the neighborhood of the cell aggregate. This mask is obtained by constructing the convex hull of the seed points. The vertices of the resulting tiles are the control points Θ of the active tessellation. Note that the seed points are manually specified through the interface or automatically detected from a provided image of the cells' nuclei.

We follow a coarse-to-fine optimization strategy. We first optimize the active tessellation made of the few control points of the Voronoi tiling. We obtain a rough segmentation that is less likely to be stuck in local minima. We then double the number of control points to increase the flexibility of the active tessellation and we optimize it once again. This strategy makes our active contour less sensitive to initialization.

6.4.3 Experiments and validation

We perform three experiments to evaluate the performance of our active tessellation. We first investigate its robustness with respect to noise and dim membranes on synthetic data. Then, we illustrate applications on real data.

Synthetic Data

We compare our approach in term of accuracy against the watershed method [189] implemented by I. Arganda-Carreras and D. Legland for the bioimage platform Fiji⁹. In the experiments, we compute the Jaccard index for each cell and take the average. It is this average value that we refer to as Jaccard index in the next sections.

We created a test image that simulates the fluorescence microscopy of a *C. elegans* embryo with 5 cells (Figure 6.35 (a)). We use the same seed points to initialize the two methods. The initial configuration of the active tessellation is illustrated in Figure 6.35 (a). Its initial similarity with the ground truth corresponds to $J = 0.64$.

Robustness with Respect to Noise: We corrupted the test image by different levels of additive Gaussian noise (20 realizations per level of noise, Figure 6.35

⁹The source code is available at https://imagej.net/Classic_Watershed.

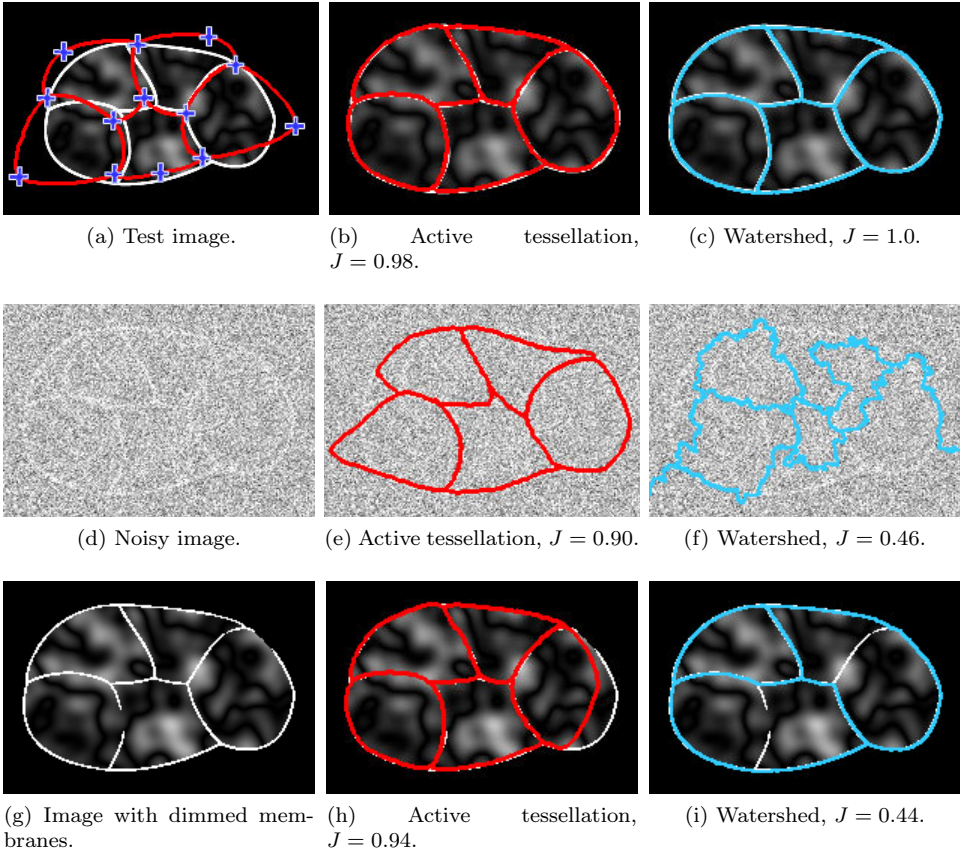


Figure 6.35: Segmentation outcomes. (a)-(c) Test images. (a) Initial configuration of the active tessellation; (d)-(f) noisy data with SNR= 0.81; (g)-(i) image with 23.95% of membrane information loss.

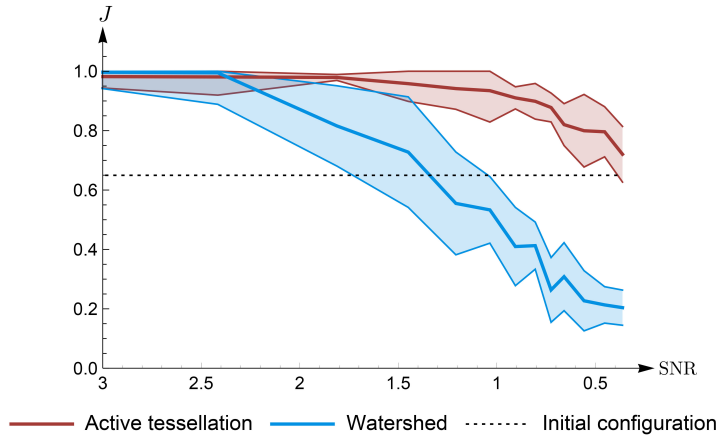


Figure 6.36: Segmentation of noisy data. Evolution of the Jaccard index as a function of the SNR. Filled area: standard deviation across the 20 realizations.

(d)). The SNR corresponding to a given noise level and Jaccard index were computed. The SNR that we use is the ratio of the mean value of the signal and the standard deviation of the noise. The results are given in Figure 6.36 and illustrated in Figures 6.35 (e) and (f). We observe that the active tessellation is robust with respect to noise since it is able to give a proper segmentation outcome even for low SNRs. On the contrary, the accuracy of the watershed method degrades significantly for a SNR below 2.4.

Robustness with Respect to Dim Membranes: We progressively dimmed the fluorescence signal on the membranes of the test image (Figure 6.35 (g)). We computed the Jaccard index as a function of the information-loss percentage. This dimming percentage corresponds to the ratio of the mean intensity on the membrane of the test image over the one of the corrupted image. The resulting plot is given in Figure 6.37 and we illustrate results in Figures 6.35 (h) and (i). The active tessellation accurately segments the cells until 49% of information loss while the watershed method can tolerate no more than 15% of information loss, then it quickly decreases. As this model is only based on intensity, it leaks through dim membranes. Due to the structure and smooth-

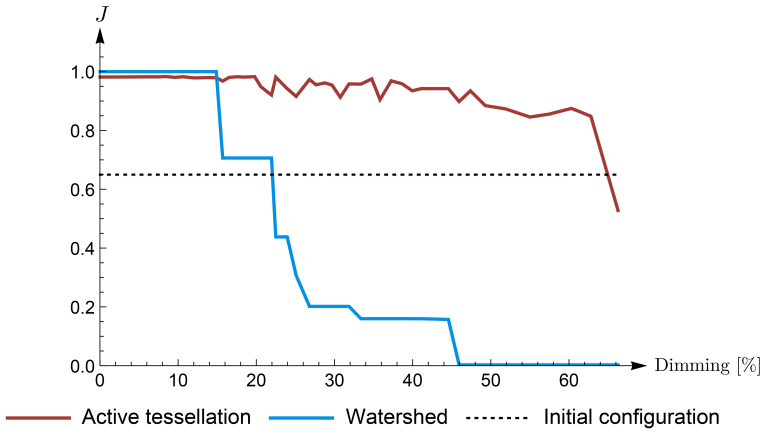


Figure 6.37: Evolution of the Jaccard index as a function of the dimming percentage.

ness of the active tessellation, the proposed framework does not suffer from leakage.

Real Data

We applied our active tessellation on real biomedical images. These images are challenging because of the presence of noise and gaps in the membranes. For each segmentation, the initial configuration of the active tessellation has 2.6 control points per cell in average. We compute the Jaccard index of each outcome considering a manual segmentation as ground truth. The results obtained are satisfactory in most cases (Figures 6.32 and 6.38).

6.4.4 Conclusions

We have presented a new subdivision-based active contour for the segmentation of cell aggregates. We have modeled the active contour by a smooth tessellation and used the oriented-ridge-based energy term designed in Section 5.3 to efficiently attract the curve toward the membranes. The tessellation structure prevents from

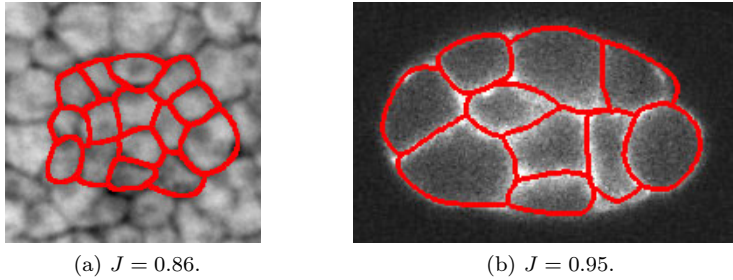


Figure 6.38: Cell segmentation of (a) cornea endothelium; (b) *C. elegans* embryo a in light-sheet fluorescence-microscopy image. Source: R. Jankele and P. Gönczy, EPFL.

overlapping segmentation of the cells and from leakage issues. Moreover, each cell of the segmentation outcome can be easily extracted as a continuous closed curve making possible the computation of cell metrics. We have demonstrated the robustness of our method under noisy conditions and to dim membranes. We have also illustrated its behavior on real bioimages. The main contributions related to this work are:

- The construction of a smooth tessellation to describe an active contour;
- The derivation of an oriented ridge-based energy functional (see (5.3.3) and (5.3.4));
- The implementation of the whole framework.

Chapter 7

Active Subdivision Surfaces

In this chapter, we extend the 2D multiresolution subdivision snake exposed in Section 6.3 to its 3D counterpart for the extraction of volumetric structures.

Subdivision is widely used in computer graphics for representation and modeling [107, 108]. As it was motivated in Chapter 4, this geometric representation combines the advantages of parametric and mesh-based approaches: The continuously defined limit surface is fully driven by the initial coarse mesh which consists of only few parameters.

The use of subdivision to construct segmentation models was pioneered in 2D by [171] for the DLG-scheme [172]. We then presented a generic framework that is valid for any convergent subdivision schemes in [70]. The extension to 3D models is more challenging. From a computational point of view, the geometry of the surface and the mesh connectivity increase the complexity of the implementation. Shapes are encoded with more control points, with three degrees of freedom for each one, which renders the optimization more complicated and slower. Moreover, it might be necessary to maintain evenly spaced control points to favor a representative sampling of the surface. In the literature, only few works used subdivision to segment volumes. The authors of [170] presented the modelization of left ventricles using Doo-Sabin surfaces [132], while the segmentation of branch vessel structures was performed in [190] using Loop's subdivision scheme [110].

In this chapter, we present¹ the generic construction of active subdivision surfaces in the context of any subdivision scheme that operates on triangular meshes. The main contributions related to this work are 1) a new 3D geometrical representation based on subdivision. The subdivision operator confers important properties to the surface such as smoothness, reproduction of desirable shapes and interpolation; 2) the derivation of region- and gradient-based energy functions that are guaranteed to have the proper limit proposed in [52]; 3) the presentation and integration of an algorithmic coarse-to-fine optimization strategy. This speeds up the computations and increases the robustness. We have implemented the method as a user-friendly open-source plugin² for the bioimaging platform Icy [11].

Throughout this chapter we use the notations described in Section 4.6. Moreover, we consider orientable closed surfaces, *i.e.*, compact and without boundary, since we want our active surface to segment blob-like objects within 3D images.

The chapter is organized as follows: In Section 7.1, we describe the generic construction of active subdivision surfaces on triangular meshes. Then, in Section 7.2, we provide a coarse-to-fine optimization strategy. In Section 7.3, we perform an extensive validation of active subdivision surfaces on both synthetic and real biological images. In particular, we show that the scheme is robust in the presence of noise and with respect to the initialization. Finally, conclusions are drawn in Section 7.4.

7.1 Framework

We implicitly represent the surface of the deformable model by the continuously defined, orientable, closed limit surface σ of a convergent subdivision scheme

$$\sigma = \lim_{k \rightarrow \infty} \mathcal{M}_{(k)}, \quad (7.1.1)$$

where $\mathcal{M}_{(k)}$ is the triangular mesh at the k -th subdivision step obtained by (4.6.2). Its shape is encoded by the $M = N_0$ control points $\Theta = \mathbb{P}_{(0)} = \{\mathbf{p}_{(0)}[m]\}_{m \in \{0, \dots, M-1\}}$. The number M of control points determines the number of degrees of freedom of the

¹This section is based on our work [72], in collaboration with L. Romani and M. Unser. A demo of the corresponding plugin and related documentation are available at <http://bigwww.epfl.ch/demo/subdivision-surfaces/>.

²See footnote 1.

model. A small M leads to simple and constrained shapes, while an increase in M brings additional flexibility to approximate arbitrary surfaces. This representation implies that the properties of the active surface depend on the choice of the subdivision scheme (see Section 4.6.3). A mandatory requirement is affine invariance. Moreover, the quality of the segmentation outcome can be influenced as follows: first, the regularity of the surface defines the smoothness of the segmentation result; second, the geometric reproduction properties have to match the shape of interest.

We reduce the energy of the active subdivision surface to an image energy term (see Section 1.2.2). The smoothness of the surface is ensured by choosing a subdivision scheme that produces at least C^1 surfaces. We use a combination of the gradient- and region-based terms (5.1.8) and (5.2.9) such as

$$E_{\text{snake}}(\mathbb{P}_{(k)}(\Theta)) = bE_{\text{gradSD}}(\mathbb{P}_{(k)}(\Theta)) + (1 - b)E_{\text{intensitySD}}(\mathbb{P}_{(k)}(\Theta)), \quad (7.1.2)$$

where $\mathbb{P}_{(k)} := \mathbb{P}_{(k)}(\Theta)$, given by (4.6.2), describes the surface and $b \in [0, 1]$ is a tradeoff parameter that balances the contribution of the two energies.

7.2 A Coarse-to-Fine Optimization Strategy

In Section 6.3, we have shown the advantage of a multiresolution strategy, for 2D subdivision models, to accelerate the segmentation and to make it more robust to the initialization. The use of such algorithms is even more relevant in 3D as the optimization is more difficult. There, we propose an integrated coarse-to-fine-optimization strategy that combines the refinement of 1) the mesh resolution, to make the computation faster and less sensitive to the initialization; 2) the number of control points, to maintain a favorable sampling of the mesh throughout the process.

Coarse-to-Fine Mesh Resolution: The energy terms (5.1.8) and (5.2.9) depend on the subdivision points $\mathbb{P}_{(k)}$, which is the source of the main computational bottleneck. The accuracy of the energy and, therefore, of the segmentation, increases with the resolution of the mesh. However, it also considerably slows down the computation. Moreover, active surfaces tend to be sensitive to the initialization, especially when using gradient-based energy. To address those issues, we optimize the active surface in a coarse-to-fine fashion that is inherent to the iterative process of subdivisions. Our algorithm exploits the

following properties: 1) a smoothed volume contains fewer details and less noise than the original one; 2) the resolution of the mesh (*i.e.*, the number of subdivision points) can be adapted to the resolution of the object to be segmented (*i.e.*, the level of detail in the volume).

Algorithm: We apply K successive lowpass filters G_k to the original volume f to obtain K smoothed volumes f_k , with the width of G_k being higher than that of G_{k+1} . The active subdivision surface is first optimized on the coarsest volume f_1 , where the object of interest only contains few details. The initial mesh on f_1 can be coarse as well since the shape of the underlying object tends to get simplified. The optimization on f_1 is fast and the outcome is then used as initialization at the next resolution level on f_2 . We refine the mesh and optimize it on f_2 . The process continues until the optimization reaches the finest resolution level that corresponds to the original volume f .

Coarse-to-Fine Density of the Control Points: The segmentation of intricate shapes requires a large number M of control points in order to catch all the details. However, the segmentation becomes less robust when M increases as the optimization iterative process is more likely to be stuck in local minima. Moreover, the deformation of the surface can lead to an undesirable distribution of the control points along the surface, which results in an unfavorable sampling of the mesh. To avoid it during the optimization, we start with few control points and then progressively increase M . This is made possible by the fact that the mesh $\mathcal{M}_{(k)}$ is entirely defined by any coarser mesh $\mathcal{M}_{(q)}$, $0 \leq q \leq k - 1$. In this way, our initial segmentation is rough at first, with a poor flexibility of the active subdivision surface. The upside is a good distribution of the control points and, thus, of the subdivision points over the surface. We then progressively refine local details by increasing M . The final number of control points depends on the intricacy of the shape to be segmented.

The pseudo-code in Algorithm 7.1 describes the entire coarse-to-fine optimization strategy. It combines the refinement of the resolution of the mesh with that of the density of the control points. In this code, the set \mathbb{C} contains the control points. Note that they change after each round of optimization. We thus denote by \mathbb{C}_{opt} the set of the optimized control points. From now on, we denote by $E_{\text{snake}}(f, \mathbb{P}_{(k)}(\Theta))$ the energy of the active surface as it also depends on the volume image on which it

ALGORITHM 7.1: Coarse-to-fine optimization strategy.

Input: original volume f , initial control points $\mathbb{P}_{(0)}$, level k_f for the final control points $\mathbb{P}_{(k_f)}$

Initialization: $\mathbb{P}_{(1)} = S_0\mathbb{P}_{(0)}$ (low-resolution mesh)
 $\mathbb{C} = \mathbb{P}_{(0)}$ (set containing the control points)
 $k_0 = 0$ (current level of the control points)

For K iterations over $k \geq k_0 + 1$:

compute: $f_k = f * G_k$

optimize: $\mathbb{C}_{\text{opt}} = \arg \min_{\mathbb{C}_{\text{opt}}} E_{\text{snake}}(f_k, \mathbb{P}_{(k)}(\mathbb{C}_{\text{opt}}))$

$\mathbb{P}_{(k)_{\text{opt}}} = S_{k-1}S_{k-2} \cdots S_{k_0}\mathbb{C}_{\text{opt}}$

refine the resolution of the mesh: $\mathbb{P}_{(k+1)} = S_k\mathbb{P}_{(k)_{\text{opt}}}$

refine the density of the control points:

If $(k_0 < k_f)$ $\mathbb{C}_{\text{opt}} = S_{k_0}\mathbb{C}_{\text{opt}}$ and $k_0 \leftarrow k_0 + 1$

Until: high-resolution segmentation on the original volume f

is optimized. The volumes f_k and their pre-integrated versions are precomputed, which accelerates the segmentation process.

7.3 Experiments and Validation

We proceed in four steps to evaluate the performance of the proposed active subdivision surface. We first test its robustness to noise and, in a second step, its sensitivity to the initialization. Then, we investigate its accuracy in the context of the segmentation of an intricate shape, when a lot of flexibility is required from the active surface. Finally, we illustrate applications on real biomedical data where the ground truth is not available.

For each experiment, we use Loop's scheme, presented in Section 4.6.4, to repre-

sent our active subdivision surface. We carry out the optimization by a Powell-like line-search method [61]. The experiments are performed on a 1.7 GHZ processor with 8 GB RAM.

We use the Jaccard index J to measure the overlap between a segmentation result \mathcal{V} and the corresponding ground truth \mathcal{V}_{GT} . It is defined as

$$0 \leq J = \frac{|\mathcal{V} \cap \mathcal{V}_{\text{GT}}|}{|\mathcal{V} \cup \mathcal{V}_{\text{GT}}|} \leq 1. \quad (7.3.1)$$


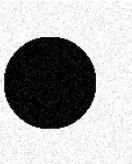
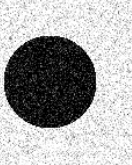
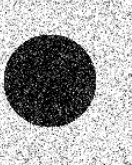
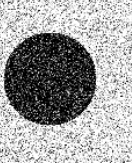
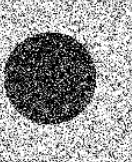
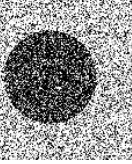
7.3.1 Robustness to Noise

We investigate the robustness to noise of the active subdivision surface as a function of the number M of control points. The test images consist in a binary sphere of radius 40 voxel units on a 3D array of size $(256 \times 256 \times 256)$ voxels. We corrupted the test image by six levels of additive white Gaussian noise (50 realizations per level of noise). We initialized the active subdivision surface with the roughly spherical surface described by the high-resolution mesh $\mathcal{M}_{(4)}$ given in Figure 4.5 (e). Its overlap with the ground truth corresponds to $J = 0.39$ (Figure 7.1). We ran the optimization process for 3,500 iterations using only the region-based energy $E_{\text{intensitySD}}$, *i.e.*, we choose $b = 0$ in (7.1.2), and the coarse-to-fine strategy described in Section 7.2. The signal-to-noise ratio corresponding to a given noise level and the median Jaccard index were computed. The SNR that we use is the ratio of the mean value of the signal and the standard deviation of the noise. The results are summarized in Table 7.2, where M is the final number of control points. We observe that the active subdivision surface is robust to noise since it is able to segment satisfactorily, even for low SNRs. This can be explained by the fact that the region energy $E_{\text{intensitySD}}$, given by (5.2.9), estimates the mean intensity over regions, while Gaussian noise has zero mean.

7.3.2 Robustness to the Initialization

To study the sensitivity of the active subdivision surface to its initialization, we compared our model in terms of accuracy and speed against other segmentation methods such as the active parametric surface described in [52] and the 3D active mesh of [43]. The implementation of the two methods was taken from the package Icy [11].

TABLE 7.2: Jaccard Indices for the Segmentation of Noisy Data.

		$\text{SNR} = \infty$						
		$\text{SNR} = 12.75$						
		$\text{SNR} = 5.1$						
		$\text{SNR} = 3.18$						
		$\text{SNR} = 2.55$						
		$\text{SNR} = 1.7$						
		$\text{SNR} = 1.275$						
M								
6		0.96	$0.96 \pm 7 \times 10^{-4}$	$0.95 \pm 2 \times 10^{-3}$	0.94 ± 0.06	0.94 ± 0.12	0.79 ± 0.16	0.59 ± 0.14
18		0.98	0.98 ± 10^{-3}	0.97 ± 0.11	0.95 ± 0.06	0.94 ± 0.089	0.80 ± 0.16	0.60 ± 0.23

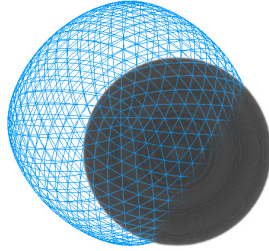


Figure 7.1: Initialization on the volumetric image, $J = 0.39$.

The test image is the binary sphere of Section 7.3.1. For each method, the initialization is (essentially) a sphere of radius r voxel units centered in the image. The goal is to segment the binary sphere from several initializations by varying the value of r .

We initialized our active subdivision surface with the low-resolution mesh $\mathcal{M}_{(2)}$ illustrated in Figure 4.5 (c), the control points being the 6 vertices that make up the octahedron (see Figure 4.5 (a)). We optimized the model using the gradient-based energy E_{gradSD} , *i.e.*, we choose $b = 1$ in (7.1.2), and the coarse-to-fine strategy with the parameters given in Table 7.3. In this table, γ denotes the standard deviation of a Gaussian filter. For the active mesh, we set the mesh resolution to 12, the time-evolution step to 0.1, the window size to 100, the contour smoothness to 0.04, and we evolved the mesh using a gradient-based energy with weight (-0.1) . For the active parametric surface, we set the number of control points to 12 as it favors ellipsoid-like shapes during the segmentation process. We deformed the parametric surface using E_{grad} , given by (2.2.3). For this method only, we ran the optimization on both the original image and on a smoothed version of the image filtered with a Gaussian kernel with $\gamma = 7$. We computed the Jaccard index of the segmentation outcome over all initializations and methods. The results as well as the segmentation time (without preprocessing) are given in Table 7.4. In this table, "failed" means that the active mesh did not detect the sphere and vanished, or that the parametric surface self-intersected. Active subdivision surfaces and active parametric surfaces have a similar performance in terms of accuracy and speed as soon as the initialization is close enough to the object to segment. Otherwise, the basin of attraction of

TABLE 7.3: Parameters of the Coarse-to-Fine Optimization Strategy.

Iteration	Control Points		Mesh		Filter Size γ
1	$\mathbb{P}_{(0)}$	6 pts	$\mathcal{M}_{(2)}$	66 pts	7
2	$\mathbb{P}_{(1)}$	18 pts	$\mathcal{M}_{(3)}$	258 pts	2
3	$\mathbb{P}_{(1)}$	18 pts	$\mathcal{M}_{(4)}$	1026 pts	none

the active parametric surface is too narrow for the gradient-based energy. We observe that the standard deviation $\gamma = 7$ is large enough to attract the active parametric surface for almost every initial configuration. However, in each case the segmentation is less accurate as the boundary of the sphere to segment is smoothed. The active mesh performs well, provided that the initialization includes the sphere to segment. Otherwise, it systematically fails. The active subdivision surface led to accurate segmentation even for initializations far from the object to segment. This result is explained by the use of the coarse-to-fine optimization strategy since the model is initialized on the coarsest image with reduced details and a large basin of attraction. In addition, the proposed method is also the fastest.

7.3.3 Segmentation Accuracy

In this section, we generated a binary synthetic blebbing cell [191] located at the center of a volumetric image of size $(256 \times 256 \times 256)$ voxels (Figure 7.2 (a)). The presence of blebs on the surface of the cell makes the segmentation challenging and requires a lot of flexibility from the segmentation model. We then compared our proposed method in terms of accuracy to the two segmentation methods mentioned in Section 7.3.2.

For each method, the initialization is (essentially) a centered sphere of radius 50 voxel units. Its overlap with the blebbing cell corresponds to $J = 0.56$. We initialized the active subdivision surface with the low-resolution mesh $\mathcal{M}_{(2)}$ illustrated in Figure 4.5 (c), encoded by the 6 control points that form the octahedron of Figure 4.5 (a). For the energy E_{snake} , we set the tradeoff parameter to $b = 0.8$ in (7.1.2). We evolved the active subdivision surface using the coarse-to-fine optimization strategy with the parameters given in Table 7.5. For the active mesh, we

TABLE 7.4: Accuracy and Efficiency with Respect to the Initialization.

Method Radius	Active Subdivision Surface		Active Parametric Surface		Active Parametric Surface, $\gamma = 7$		Active Mesh	
	J	Time [s]	J	Time [s]	J	Time [s]	J	Time [s]
20	0.99	5.74	0.98	17.56	0.91	5.40	failed	-
25	0.99	5.71	0.16	5.87	0.91	7.27	failed	-
30	0.99	5.73	0.33	6.36	0.91	5.35	failed	-
35	0.99	5.72	0.99	6.15	0.91	5.32	failed	-
45	0.99	5.82	0.99	6.01	0.91	5.40	0.96	7.61
50	0.99	5.85	0.49	5.84	0.91	5.73	0.95	12.38
55	0.99	5.81	0.37	5.63	0.91	5.83	0.95	18.66
60	0.99	5.83	0.30	5.82	0.91	5.42	0.95	32.58
65	0.99	5.84	0.23	6.10	0.91	5.81	0.95	41.92
70	0.99	5.80	0.17	6.40	failed	-	0.96	60.70

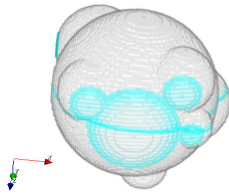
TABLE 7.5: Parameters of the coarse-to-fine optimization strategy.

Iteration	Control Points		Mesh		Filter Size γ
1	$\mathbb{P}_{(0)}$	6 pts	$\mathcal{M}_{(2)}$	66 pts	10
2	$\mathbb{P}_{(1)}$	18 pts	$\mathcal{M}_{(3)}$	258 pts	5
3	$\mathbb{P}_{(2)}$	66 pts	$\mathcal{M}_{(4)}$	1026 pts	2.5
4	$\mathbb{P}_{(3)}$	258 pts	$\mathcal{M}_{(5)}$	4098 pts	none

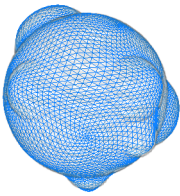
set the mesh resolution to 5, the time-evolution step to 0.1, the window size to 100, the contour smoothness to 0.05, and we optimized the mesh using a region-based energy with weight 1. For the active parametric surface, we used a mix of the gradient-based energy E_{grad} and the region-based energy $E_{\text{intensity}}$, given by (2.2.3) and (2.2.7), respectively, with a tradeoff parameter equal to 0.8.

For this last segmentation method, we repeated the experiment for different numbers of control points. They are listed in Table 7.6, where we also show a comparison of the final Jaccard index³. 3D views as well as 2D orthogonal views of the segmentation outcomes are illustrated in Figures 7.2 and 7.3, respectively. We observe that both our proposed method and the active mesh accurately segment the blebbing cell and give smooth meshes (Figure 7.2 (b) and (c)). The active mesh tends to extend outside of the boundary of the object compared to the active subdivision surface (Figure 7.3, YZ plane). However, it segments better two close blebs (Figure 7.3, XZ plane). Clearly, the active parametric surface has the worst performance. The segmentation with 12 control points leads to a very smooth surface. However, only the main sphere of the blebbing sphere is segmented, as the flexibility afforded by 12 control points is low. With the additional flexibility afforded by 36 control points it is able to segment some blebs. However, irregularities in the surface start to appear. When we further increase the flexibility of the active parametric surface, the distribution of the control points misbehaves; moreover, irregularities and twists of the surface grow too large.

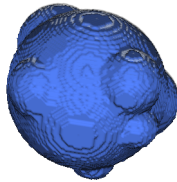
³The ideal segmentation of just the main sphere of the blebbing cell would correspond to $J = 0.88$.



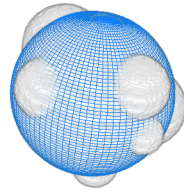
(a) Ground truth. Blue lines: planes used for the 2D views of Figure 7.3.



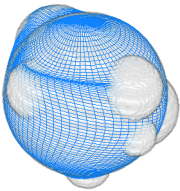
(b) Active subdivision surface.



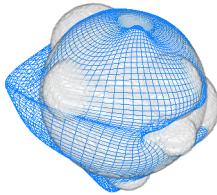
(c) Active mesh.



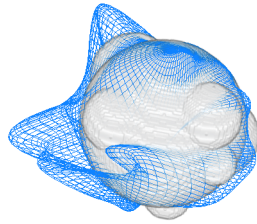
(d) Active parametric surface (12 control points).



(e) Active parametric surface (36 control points).



(f) Active parametric surface (62 control points).



(g) Active parametric surface (96 control points).

Figure 7.2: 3D views of the segmentation outcomes.

TABLE 7.6: Jaccard Indices for the Segmentation of a Blebbing Cell.

Method	J
Active Subdivision Surface	0.98
Active Mesh	0.96
Active Parametric Surface (12 control points)	0.87
Active Parametric Surface (36 control points)	0.93
Active Parametric Surface (62 control points)	0.87
Active Parametric Surface (96 control points)	0.69

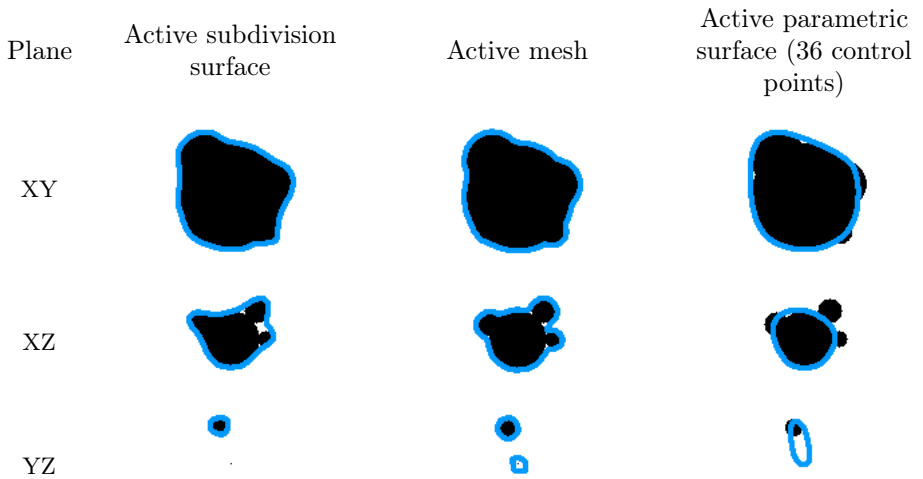


Figure 7.3: 2D orthogonal views of the segmentation outcomes.

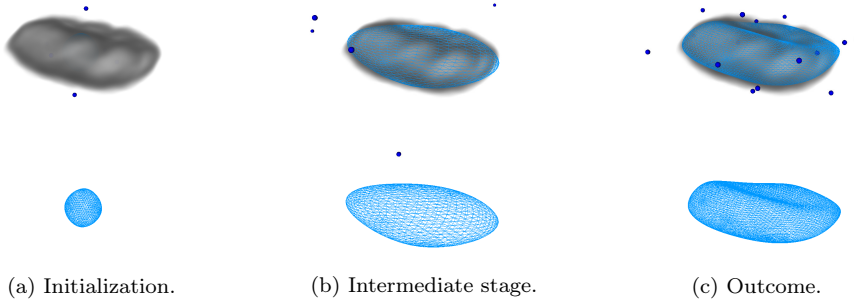


Figure 7.4: Segmentation of the nucleus of the neuron of a rat in a 3D microscopic volume. Dots: control points.

7.3.4 Segmentation on Real Biomedical Images

We illustrate the behavior of the proposed active subdivision surface on two real biomedical images, with unknown ground truth. We rely on qualitative assessments to validate the accuracy of the segmentation. The volumetric data of Figure 7.4 represent the optical density of the neuron of a rat in a 3D microscopic image [1]. The volumetric data of Figure 7.5 results from the MRI acquisition of a human brain, with the purpose of measuring its total intracranial volume (TIV). TIV is used in medicine to detect temporal morphological changes related to neurological diseases [4]. However, its segmentation is challenging because of the numerous irregularities that compose the brain, such as the convoluted areas formed around the temporal lobe and the cerebellum. The qualitative assessment of our segmentations yields satisfactory results.

7.4 Conclusions

We have presented the 3D generic construction of a new family of active surfaces on triangular meshes. The deformable model is characterized by few control points and a natural discrete implementation. It can approximate closed surfaces with arbitrary precision by iteratively refining a set of 3D control points. We have also

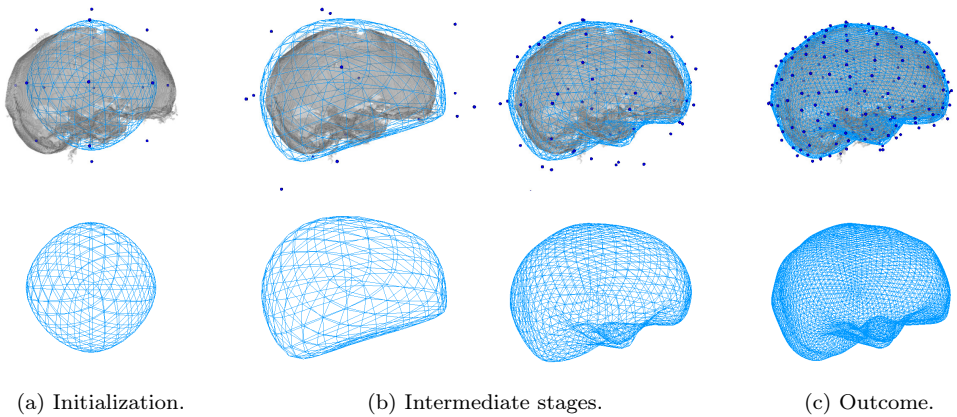


Figure 7.5: TIV segmentation of a 3D MRI scan. Dots: control points.

proposed an integrated coarse-to-fine optimization strategy to adapt the resolution of the mesh to the level of detail in the volume. It results in speed-up of the optimization and better robustness. Moreover, this multiresolution strategy allows us to maintain a favorable sampling of the mesh by gradually increasing the flexibility of the model during the optimization. We have applied our proposed method to a variety of problems that involve synthetic data and real biomedical images. We have compared our framework with two segmentation methods and shown that our model is robust with respect to noise and initial conditions. The primary contributions related to this work are:

- A new geometrical representation based on subdivision;
- The design of the gradient-based energy (5.1.8) as well as the robust region-based energy term (5.2.9);
- The presentation and integration of an algorithmic coarse-to-fine optimization strategy (see Section 7.2).

Chapter 8

Conclusion

We end this thesis with a concluding chapter. We first recap the novel scientific contributions and results. Then, we briefly comment on further directions of research.

8.1 Contributions

We focused on the problem of segmentation in the context of biomedical image analysis. We identified limitations of parametric snakes and addressed them providing new representation models as well as novel energies. With these two ingredients in hand, we constructed new active contours and surfaces. In the following, we summarize our contributions grouped by field.

Representation

- *Locally refinable parametrization:* We presented a new parametrization for curves and tensor-product surfaces where the number of control points can be locally increased without altering the shape. In a segmentation context, these additional control points then allow to locally deform the shape with better accuracy.

- *Subdivision-based representation:* We introduced subdivision schemes, traditionally used in computer graphics, into a segmentation framework by describing a snake as a subdivision curve/surface. The main benefits of this representation is its simplicity of implementation and its multiresolution property, which allows for the contour of a shape to be represented at varying resolutions.

Image Energy

- *Edge- and region-based energies:* We adapted standard edge- and region-based energies to our locally refinable parametric closed curves. In addition, we provided discrete edge- and region-based energies tailored for the subdivision-based representation.
- *Ridge-based energy:* As edge-based energies do not perform well on ridge areas, we proposed an oriented ridge-based energy. We provided the continuous and discrete formulations.
- *Texture-based energy:* We proposed a novel energy that combines image intensity and texture information. The two types of image information are balanced using Fisher's linear discriminant analysis and the framework can be used with any filter-based texture features. We provided the continuous and discrete formulations. This energy was motivated by the inefficiency of standard edge- and region-based energies to discriminate a target from its background in images with a low contrast between the two.

Optimization

- *Subdivide and conquer strategy:* We presented and integrated to our snakes an algorithmic coarse-to-fine optimization strategy. First, we successively apply lowpass filters to the original image to obtain several smoothed images. We initialize the snake on the coarsest image and adapt its resolution to the scale of the target feature (*i.e.*, subdivide). Then, we perform a first optimization on this image that is fast, and we use the outcome as initialization at the next resolution level. We successively increase the resolution of the snake and the image until the optimization reaches the finest resolution level that corresponds to the original image (*i.e.*, conquer). This subdivide and conquer

strategy was tailored for our representations that satisfy the multiresolution property. It improves the robustness in the presence of noise and enlarges the basin of attraction of our snakes compared to traditional parametric snakes.

Bioimaging Software for Segmentation

With the previous contributions in hand, we designed new active contours/surfaces that are generic enough to be applied in a wide range of applications. We implemented the corresponding framework as bioimaging software for segmentation.

- *2D locally refinable snakes:* We presented new parametric snakes that are locally refinable. We exploited our parametrization for locally refinable closed curves, as well as our corresponding edge- and region-based energies. Our method is generic and can be used with any valid scaling function. Locally refinable snakes segment intricate shapes with better accuracy using fewer control points than classical parametric snakes.
- *2D/3D subdivision snakes:* We proposed a generic framework to construct subdivision snakes. We exploited subdivision schemes to represent the snakes, and as energy terms we used our discrete formulations of the edge- and region-based energies. These snakes satisfy the property of multiresolution, which allowed us to well adopt the subdivide and conquer strategy. In addition, they are robust in the presence of noise and have a enlarged basin of attraction compared to classical parametric snakes.
- *2D active tessellations:* We presented a new active contour to segment cell aggregates. We described the snake by a smooth tessellation built from subdivision schemes, and we deformed it in a global manner using our oriented ridge-based energy. By construction, cells are segmented without overlap and the tessellation structure is maintained even on dim membranes. Leakage, which afflicts usual image-processing methods, is thus prevented. In addition, the proposed method is robust to membrane gaps and to high levels of noise.
- *2D texture-driven parametric snakes:* We proposed new parametric snakes that are efficient to segment structures with a similar intensity distribution as their background. We represented the snake with a classical parametric closed

curve and used our texture-based energy. This framework is interactive and is trained on-the-fly from small collections of pixels provided by the user. These snakes are robust to noise and improve the segmentation performance when compared to classical parametric snakes that rely on intensity information only.

Mathematical Tool for Signal Processing

- *Inner-product calculus:* We introduced an inner-product calculus to evaluate correlations and L_2 distances between closed curves represented by basis functions. In particular, we presented formulas for the direct and exact evaluation of correlation matrices in the case of closed (*i.e.*, periodic) parametric curves and periodic signals. Our first motivation for this work was to provide an efficient way to compute the exact calculation of the area enclosed by parametric snakes.
- *Stochastic model of closed curves:* We presented two approaches for the reconstruction of periodic continuous-domain signals, that are the coordinate functions of a closed curve, from their noisy measurements. We focused on two reconstruction paradigms: variational and statistical. We showed that the two approaches are connected, and that for each one the optimal solution is a periodic spline related to a differential operator.

8.2 Research Outlook

The research presented in this thesis opens several interesting avenues for future investigation. Some of them are listed below.

- *Extend the texture-driven parametric snake for the segmentation of 3D and multi-modal data:* A combination of medical image modalities (*e.g.* MRI, CT with various contrasts) can provide complementary information about the texture of a specific tissue (*e.g.* organ, tumor). Hence, as our texture-driven parametric snake handles several channels, it could be of interest to use images of different modalities as inputs of the framework.

-
- *Incorporate normal control in active subdivision surfaces:* The introduction of adjustable normals in a deformable model has several advantages: First, it provides additional control over the shape, which facilitates the reproduction of sharp corners or circonvolutions; second, it allows for the design of directional energy functionals. We would like to introduce normal control in active surfaces by using normal-based subdivision schemes for their representation. Our publication [192] is a first step in this direction.
 - *Further exploit subdivision schemes:* Subdivision is a powerful paradigm for the generation of surfaces of arbitrary topology. In this thesis, we focused on closed subdivision curves/surfaces of genus 0. We plan to account for other kinds of topological properties. For instance, to segment cell aggregates, we could use a subdivision scheme to generate a smooth tessellation composed of several tiles, instead of generating each tile independently, as our active tessellation does. This method would be more elegant, simpler to implement and much easier to extend to 3D.
 - *Incorporate deep neural networks into deformable models:* A snake energy is usually tailored to detect specific features in images. This approach has some limitations. If the features of the object to segment are also present in an object of the image background, the segmentation will fail as too many features would be detected. In this context, we could use the potential of neural networks. It is possible to train an appropriate deep neural network to detect exclusively the desired edges or features of the object of interest. Then, we could directly use our energies with the output of the network.

Appendix A

An Inner-Product Calculus for Periodic Functions and Curves

In this chapter, our motivation¹ is the design of efficient algorithms to process closed curves represented by basis functions or wavelets. To that end, we introduce an inner-product calculus to evaluate correlations and L_2 distances between such curves. In particular, we present formulas for the direct and exact evaluation of correlation matrices in the case of closed (*i.e.*, periodic) parametric curves and periodic signals. We give simplifications for practical cases that involve B-splines. To illustrate this approach, we also propose a least-squares approximation scheme that is able to resample curves while minimizing aliasing artifacts. Another application is the exact calculation of the enclosed area (see (2.2.6)).

¹This work is based on our publication [75], in collaboration with D. Schmitter and M. Unser.

A.1 Introduction

A.1.1 Notations

We consider parametric closed curves as given by (2.1.4), that we briefly recall as

$$\mathbf{r}(t) = \begin{pmatrix} r_1(t) \\ r_2(t) \end{pmatrix} = \sum_{m=0}^{M-1} \mathbf{c}[m] \varphi_M(Mt - m), \quad (\text{A.1.1})$$

where $t \in [0, 1]$. To a closed curve \mathbf{r}_i is assigned the couple (φ_i, M_i) , where φ_i is the basis function and M_i is the associated number of control points. We express the corresponding M_i -periodized basis function by

$$\varphi_{i, M_i}(t) = \sum_{n \in \mathbb{Z}} \varphi_i(t - nM_i). \quad (\text{A.1.2})$$

We define the vector $\boldsymbol{\varphi}_i$, of size M_i , that contains the basis $\{\varphi_{i, M_i}(M_i \cdot -m)\}_{m \in \{0, \dots, M_i-1\}}$, as

$$\boldsymbol{\varphi}_i(t) = \begin{pmatrix} \varphi_{i, M_i}(M_i t) \\ \vdots \\ \varphi_{i, M_i}(M_i t - M_i + 1) \end{pmatrix}. \quad (\text{A.1.3})$$

The condensed notation

$$\mathbf{r}_i(t) = \mathbf{C}_i^T \boldsymbol{\varphi}_i(t) \quad (\text{A.1.4})$$

is equivalent to (A.1.1). There, \mathbf{C}_i is the $(M_i \times 2)$ matrix defined as

$$\mathbf{C}_i = \begin{pmatrix} c_{i,1}[0] & c_{i,2}[0] \\ \vdots & \vdots \\ c_{i,1}[M_i - 1] & c_{i,2}[M_i - 1] \end{pmatrix}. \quad (\text{A.1.5})$$

In the case of 1D signals, the matrix \mathbf{C}_i collapses to a vector.

A.1.2 Inner Products

The exact computation of inner products is a frequent operation in signal and image processing such as for the evaluation of L_2 distances, orthogonal projections

or similarity measurements. Thus, our interest here is in the efficient calculation of the L_2 -distance between two curves that may be parametrized with a different number of control points [99, 193]. We express the L_2 -inner product between the two closed curves $\mathbf{r}_1, \mathbf{r}_2 \in L_2([0, 1])$ as

$$\begin{aligned} \langle \mathbf{r}_1, \mathbf{r}_2 \rangle_{L_2([0,1])} &= \int_0^1 \mathbf{r}_1^\top(t) \mathbf{r}_2(t) dt \\ &= \text{tr} \left(\mathbf{C}_1^\top \underbrace{\left(\int_0^1 \varphi_1(t) \otimes \varphi_2(t) dt \right)}_{\mathbf{A}_{12}} \mathbf{C}_2 \right), \end{aligned} \quad (\text{A.1.6})$$

where \mathbf{A}_{12} is the *correlation matrix* of size $(M_1 \times M_2)$ specified as $[\mathbf{A}_{12}]_{k,l} = \langle [\varphi_1]_k, [\varphi_2]_l \rangle_{L_2([0,1])}$ and \otimes denotes the tensor product. To evaluate (A.1.6), the entries of the correlation matrix require the evaluation of some integrals. We present in Section A.2 a calculus that facilitates these computations in the continuous domain.

A.2 Inner-Product Calculus

A.2.1 General Calculation

We start by providing a general formula for precomputing the matrix \mathbf{A}_{12} and then discuss a number of situations that can be resolved analytically.

Proposition A.2.1. *Let φ_1 and φ_2 be two compactly supported generators with $\text{supp}\{\varphi_1\} = [a_1, b_1]$, $\text{supp}\{\varphi_2\} = [a_2, b_2]$, $M_1 \geq \text{supp}\{\varphi_1\}$, and $M_2 \geq \text{supp}\{\varphi_2\}$. The entries of the $(M_1 \times M_2)$ cross-correlation matrix $\mathbf{A}_{12} = \int_0^1 \varphi_1(t) \otimes \varphi_2(t) dt$ are given by*

$$[\mathbf{A}_{12}]_{k,l} = \frac{1}{M_1} \sum_{m=m_1}^{m_2} a_{12}(-\tau_{k,l,m}), \quad (\text{A.2.1})$$

where

$$\begin{aligned}
 a_{12}(t) &= \int_{\mathbb{R}} \varphi_1(u) \varphi_2\left(\frac{M_2}{M_1}(u-t)\right) du \\
 &= \left(\varphi_1 * \varphi_2\left(-\frac{M_2}{M_1}\cdot\right) \right)(t), \tag{A.2.2}
 \end{aligned}$$

$\tau_{k,l,m} = M_1 \left(m + \frac{k}{M_1} - \frac{l}{M_2} \right)$, $m_1 = \lceil \min(p_1, p_2) \rceil$, $m_2 = \lfloor \max(1 + p_1, 1 + p_2) \rfloor$,
 $p_1 = \left(\frac{1}{M_2}(a_2 + l) - \frac{1}{M_1}(a_1 + k) \right)$, and $p_2 = \left(\frac{1}{M_2}(a_2 + l) - \frac{1}{M_1}(b_1 + k) \right)$. There, $\lfloor \cdot \rfloor$
 and $\lceil \cdot \rceil$ denote the floor and the ceil function, respectively.

The proof of Proposition A.2.1 is given in Appendix A.5.1. In the case where the generators are even or odd functions with respect to the same axis, Proposition A.2.1 is simplified as specified by Corollary A.2.2.

Corollary A.2.2. *Let φ_1 and φ_2 be two even or odd functions with respect to the same axis of symmetry.*

a) *The correlation between the one-periodic functions $[\varphi_1]_k$ and $[\varphi_2]_l$ is*

$$[\mathbf{A}_{12}]_{k,l} = \frac{1}{M_1} (a_{12}(-\tau_{k,l}) + a_{12}(M_1 - \tau_{k,l}) + a_{12}(-M_1 - \tau_{k,l})), \tag{A.2.3}$$

where $\tau_{k,l} = M_1 \left(\frac{k}{M_1} - \frac{l}{M_2} \right)$.

b) *If φ_1 and φ_2 have the same parity, then the correlation is expressed as*

$$[\mathbf{A}_{12}]_{k,l} = \frac{1}{M_1} (a_{12}(|\tau_{k,l}|) + a_{12}(|\tau_{k,l}| - M_1)), \tag{A.2.4}$$

with $\tau_{k,l} = M_1 \left(\frac{k}{M_1} - \frac{l}{M_2} \right)$.

Observe that, if $M_1 = M_2$, further simplifications of Proposition A.2.1 are obtained. For instance, the case when $\varphi_1 = \varphi_2$ or $\varphi_2 = \dot{\varphi}_1 = \frac{d\varphi_1}{dt}$ implies that $a_{12} = \dot{a}_{11}$. Also note that, due to the periodicity of the generators and to $M_1 = M_2$, the matrix \mathbf{A}_{12} is circulant and thus entirely specified by its M_1 entries $\{[\mathbf{A}_{12}]_{0,l}\}_{l \in \{0, \dots, M_1-1\}}$ [194]. This matrix is diagonalizable and hence, an explicit expression for its inverse is easy to obtain.

A.2.2 Specific Cases of a_{12} in Practice

(Exponential) B-splines, that we reviewed in Section 2.3.1, are basis functions that are widely used in signal processing and have interesting mathematical properties that can be exploited to simplify the proposed inner-product calculus. In this section, we illustrate how the expression of a_{12} is simplified for specific cases that frequently appear in practice and that involve B-splines. We use the notations and properties presented in Section 2.3.1.

Correlation Between Polynomial B-Splines

The function a_{12} for the case of polynomial B-splines of different orders is determined according to Proposition A.2.3.

Proposition A.2.3. *Let $\varphi_1 = \beta_{0_{L_1}}$ and $\varphi_2 = \beta_{0_{L_2}}$. Then,*

$$a_{12}(t) = \left(\frac{M_2}{M_1}\right)^{L_2-1} \sum_{l=0}^{L_1} \sum_{k=0}^{L_2} \binom{L_1}{l} \binom{L_2}{k} (-1)^{l+k+L_2} \zeta^{L_1+L_2-1} \left(t + \frac{kM_1}{M_2} - l\right), \tag{A.2.5}$$

where ζ^L is the polynomial simple element of degree L defined as $\zeta^L(t) = \frac{t^L \operatorname{sgn}(t)}{2^{(L!)}}$ for $L \in \mathbb{N}$.

The proof is given in Appendix A.5.2.

Correlation Between Exponential B-Splines

In the case where $\varphi_1 = \beta_{\alpha_1}$ and $\varphi_2 = \beta_{\alpha_2}$ are two exponential B-splines of order L_1 and L_2 , respectively, and $M_1 = M_2$, we obtain

$$a_{12}(t) = \left(\prod_{n=1}^{L_2} e^{\alpha_2^*, n}\right) \beta_{\alpha_1 \cup (-\alpha_2^*)}(t + L_2), \tag{A.2.6}$$

where α^* is the complex conjugate of α [81]. Equation (A.2.6) corresponds to the cross-correlation of two exponential B-splines which yields an exponential B-spline of augmented order. Proposition A.2.4 provides a simplified expression of (A.2.6) in the case where $\varphi_1 = \beta_{\alpha}$ and $\varphi_2 = \dot{\varphi}_1$.

Proposition A.2.4. *Let $\varphi_1 = \beta_\alpha$ be an exponential B-spline of order L that contains at least one vanishing pole (we suppose $\alpha_L = 0$), and $\varphi_2 = \beta_\alpha$. Then,*

$$a_{12}(t) = - \left(\prod_{n=1}^{L-1} e^{\alpha_n^*} \right) \Delta \beta_{\alpha \cup (-\alpha^* \setminus \{0\})}(t + L - 1), \quad (\text{A.2.7})$$

where $\Delta f(t) = (f(t) - f(t - 1))$ denotes the finite difference of f , and $\alpha \setminus \{\alpha_n\}$ describes a list from which the element α_n has been excluded.

The proof is given in Appendix A.5.3. Note that, the same kind of formula also applies for fractional B-splines [195].

A.3 Applications

A.3.1 Resampling of a Spline Curve

The general scheme to reduce the size of a polygonal or spline curve \mathbf{r}_1 is to decrease its number M_1 of control points [196]. The standard method is to simply resample the curve [78]. However, this does not take into account details localized between two samples, which alters the accuracy of the approximation while eventually introducing aliasing artifacts [151]. We propose a new method which consists in computing the L_2 approximation \mathbf{r}_2 of the curve \mathbf{r}_1 , with $M_1 > M_2$. This is equivalent to compute $\arg \min_{\mathbf{C}_2} \|\mathbf{r}_1 - \mathbf{r}_2\|_{L_2}^2$. It is not difficult to show that the general solution, in the context of our framework, is given by

$$\mathbf{C}_2 = \mathbf{A}_{22}^{-1} \mathbf{A}_{21} \mathbf{C}_1, \quad (\text{A.3.1})$$

where \mathbf{C}_1 and \mathbf{C}_2 are the coefficient matrices of size $(M_1 \times 2)$ and $(M_2 \times 2)$, respectively. The entries of the matrices \mathbf{A}_{21} and \mathbf{A}_{22} , of size $(M_2 \times M_1)$ and $(M_2 \times M_2)$, respectively, can be evaluated using Proposition A.2.1 and Proposition A.2.3.

To experimentally compare resampling and approximation, we propose to reduce the outline \mathbf{r}_1 of the map of Switzerland defined by $M_1 = 930$ control points interpolated with the linear spline $\varphi_1 = \beta_{(0,0)}$ (Figure A.1, black curve). We resample \mathbf{r}_1 with both the sampling and the L_2 approximation methods for different values of $M_2 < M_1$ control points and in the basis of the quadratic spline $\varphi_2 = \beta_{(0,0,0)}$.

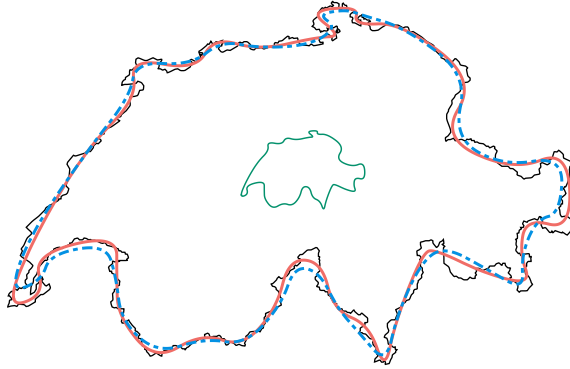


Figure A.1: Resampling of the outline of the map of Switzerland (black curve). Solid red curve and dashed blue curve: resampled versions obtained by the L_2 approximation and sampling methods, respectively, with $M_2 = 40$ samples. Green curve: reduced version of the map obtain with the L_2 approximation.

We illustrate the case $M_2 = 40$ in Figure A.1. We observe that the resampled curves act as smoothed versions of \mathbf{r}_1 with less details and increased regularity. We compute their approximation error for each value of M_2 . In Figure A.2, it is seen that the best approximation of the reduced version of the map, without aliasing artifacts, is obtained with our proposed method (Figure A.1, green curve).

A.3.2 Area Enclosed by a Parametric Curve

In this section, we consider a non-intersecting curve \mathbf{r}_1 and its derivative $\dot{\mathbf{r}}_1 = M_1 \mathbf{C}_1^\top \dot{\boldsymbol{\varphi}}_1$. The factor M_1 is due to the normalization in (A.1.1). The computation of the area enclosed by a parametric curve usually involves the evaluation of a surface integral. We propose instead to use Green's theorem [197] to express this surface integral as a contour integral, which results in a signed area expressed as

$$I = \oint_{\mathbf{r}_1} r_{1,2} dr_{1,1} = \langle r_{1,2}, \dot{r}_{1,1} \rangle_{L_2([0,1])} = M_1 \mathbf{c}_{1,2}^\top \mathbf{A}_{12} \mathbf{c}_{1,1}, \quad (\text{A.3.2})$$

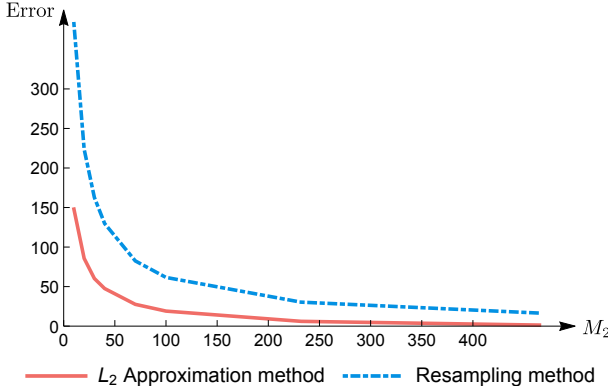


Figure A.2: Evolution of the approximation error as a function of the number M_2 of samples.

where $\varphi_2 = \dot{\varphi}_1$, $M_2 = M_1$, and $\mathbf{c}_{1,1}$ and $\mathbf{c}_{1,2}$ are the first and second column of the matrix (A.1.5), respectively. The sign of I depends on the direction in which the curve is traversed.

In the case of centered (exponential) B-splines, (A.3.2) is easily computed. For $\varphi_1 = \beta_\alpha$, we evaluate the entries of the matrix \mathbf{A}_{12} using Corollary A.2.2.a) and Proposition A.2.4. We obtain

$$[\mathbf{A}_{12}]_{k,l} = \frac{1}{M_1} \sum_{n=-1}^1 \Delta_c \beta_{\alpha \cup (\alpha \setminus \{0\})}(k - l + nM_1), \tag{A.3.3}$$

where $\Delta_c f(t) = f(t + \frac{1}{2}) - f(t - \frac{1}{2})$ denotes the centered finite difference of f . As the matrix is circulant, we only compute these values for $k = 0$ and $l \in \{0, \dots, M_1 - 1\}$. For instance, if the parametric curve (A.1.4) is constructed with the centered linear B-spline $\varphi_1 = \beta_{(0,0)}$ (Figure A.3, solid green line), we have $\Delta_c \beta_{\alpha \cup (\alpha \setminus \{0\})} = \Delta_c \beta_{(0,0,0)}$ (Figure A.3, dot-dashed blue line), where $\beta_{(0,0,0)}$ is the centered quadratic B-spline (Figure A.3, dashed red line). Then, each row of the correlation matrix is expressed as a periodic shift of the centered finite difference $[\frac{1}{2} \ 0 \ -\frac{1}{2}]$.

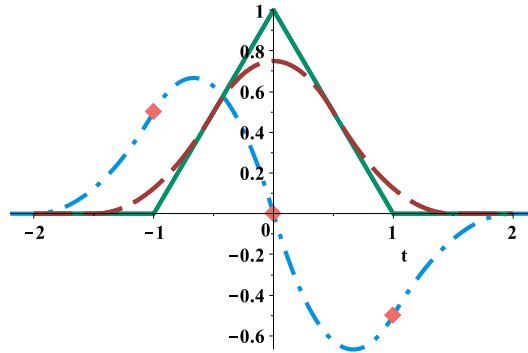


Figure A.3: Solid green line: centered linear B-spline; red dashed line: centered quadratic B-spline; blue dot-dashed line: $\Delta_c \beta_{(0,0,0)}$; pink diamonds: $\Delta_c \beta_{(0,0,0)}(k)$ for $k = -1, 0, 1$.

A.4 Conclusion

The computation of inner products between periodized basis functions requires the evaluation of a correlation matrix \mathbf{A}_{12} . This matrix frequently appears in periodic settings in classical L_2 -based signal processing as well as in image processing when dealing with parametric closed curves. We have presented exact formulas to evaluate its entries and gave simplified expressions for particular cases. As the correlation matrix itself does not depend on the weights (or control points) that specify the signal (or parametric curve), its values can be precomputed and stored in look up tables for a fast evaluation of L_2 distances. We also proposed an L_2 approximation method to resample a curve, which consists in describing the curve in a different basis using less control points. These new points are found by a least-squares minimization: The general solution requires the evaluation of two correlation matrices that can be precomputed using our proposed formulas. We compared our approach to the classical uniform resampling method and showed that the best approximation was obtained with our method. We also illustrated the use of the proposed formulas to evaluate the area enclosed by a parametric closed curve. Our inner-product calculus allows for a fast and exact evaluation of correlation integrals, which frequently appear in practice and are often only approximately computed up to date.

A.5 Appendices

A.5.1 Proof of Proposition A.2.1

$$\begin{aligned}
 \int_0^1 \varphi_{1,M_1}(M_1t - k)\varphi_{2,M_2}(M_2t - l)dt &= \int_{-\frac{l}{M_2}}^{1-\frac{l}{M_2}} \varphi_{1,M_1}(M_1(t' + \frac{l}{M_2}) - k)\varphi_{2,M_2}(M_2t')dt' \\
 &= \int_{\frac{a_2}{M_2}}^{1+\frac{a_2}{M_2}} \varphi_{1,M_1}(M_1(t + \frac{l}{M_2}) - k)\varphi_{2,M_2}(M_2t)dt \\
 &= \int_{\frac{a_2}{M_2}}^{1+\frac{a_2}{M_2}} \sum_{m=-\infty}^{+\infty} \varphi_1(M_1t - \tau_{k,l,m})\varphi_2(M_2t)dt.
 \end{aligned} \tag{A.5.1}$$

We set $\tau_{k,l,m} = M_1(m + \frac{k}{M_1} - \frac{l}{M_2})$, $m_1 = \lceil \min(p_1, p_2) \rceil$, $m_2 = 1 + \lfloor \max(p_1, p_2) \rfloor$, $p_1 = (\frac{1}{M_2}(a_2 + l) - \frac{1}{M_1}(a_1 + k))$ and $p_2 = (\frac{1}{M_2}(a_2 + l) - \frac{1}{M_1}(b_1 + k))$. Now, (A.5.1) is simplified as

$$\begin{aligned}
 \sum_{m=m_1}^{m_2} \int_{\frac{a_2}{M_2}}^{1+\frac{a_2}{M_2}} \varphi_1(M_1t - \tau_{k,l,m})\varphi_2(M_2t)dt &= \frac{1}{M_1} \sum_{m=m_1}^{m_2} \int_{\mathbb{R}} \varphi_1(t - \tau_{k,l,m})\varphi_2(\frac{M_2}{M_1}t)dt \\
 &= \frac{1}{M_1} \sum_{m=m_1}^{m_2} \int_{\mathbb{R}} \varphi_1(t)\varphi_2(\frac{M_2}{M_1}(t + \tau_{k,l,m}))dt \\
 &= \frac{1}{M_1} \sum_{m=m_1}^{m_2} a_{12}(-\tau_{k,l,m}),
 \end{aligned} \tag{A.5.2}$$

where $a_{12}(t) = (\varphi_1 * \varphi_2(-\frac{M_2}{M_1}\cdot))(t)$ and we have used the fact that $\varphi_2(\pm\frac{M_2}{2} - M_2n) = 0$ if $|n| \geq \frac{\text{supp}\{\varphi_2\} + M_2}{2M_2}$ and that $\varphi_1(\pm\frac{M_1}{2} - M_1p) = 0$ if $|p| \geq \frac{\text{supp}\{\varphi_1\} + M_1}{2M_1}$. ■

A.5.2 Proof of Proposition A.2.3

We define by Δ_b^L the L th-order causal finite-difference operator with $b \neq 0$, defined as

$$\Delta_b^L f(t) = \sum_{k=0}^L \binom{L}{k} (-1)^k f(t - \frac{k}{b}). \tag{A.5.3}$$

The Fourier transform \mathcal{F} of the causal polynomial B-spline $\beta_{\mathbf{0}_L}$ is given by

$$\begin{aligned}\mathcal{F}\{\beta_{\mathbf{0}_L}(t)\}(\omega) &= \hat{\beta}_{\mathbf{0}_L}(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega}\right)^L \\ &= \hat{\Delta}_1^L(\omega)\mathcal{F}\{\zeta^{L-1}(t)\}(\omega),\end{aligned}\tag{A.5.4}$$

where $\mathcal{F}\{\zeta^L(t)\}(\omega) = \frac{1}{(j\omega)^{L+1}}$. Let $\varphi_1 = \beta_{\mathbf{0}_{L_1}}$ and $\varphi_2 = \beta_{\mathbf{0}_{L_2}}$. We compute

$$\begin{aligned}a_{12}(t) &= \left(\beta_{\mathbf{0}_{L_1}} * \beta_{\mathbf{0}_{L_2}}\left(-\frac{M_2}{M_1}\cdot\right)\right)(t) \\ &= \mathcal{F}^{-1}\left\{\hat{\beta}_{\mathbf{0}_{L_1}}(\omega)\frac{M_1}{M_2}\hat{\beta}_{\mathbf{0}_{L_2}}\left(-\frac{M_1}{M_2}\omega\right)\right\}(t) \\ &= \mathcal{F}^{-1}\left\{\left(\frac{1 - e^{-j\omega}}{j\omega}\right)^{L_1}\frac{M_1}{M_2}\left(\frac{1 - e^{j\frac{M_1}{M_2}\omega}}{-j\frac{M_1}{M_2}\omega}\right)^{L_2}\right\}(t) \\ &= \mathcal{F}^{-1}\left\{(-1)^{L_2}\left(\frac{M_2}{M_1}\right)^{L_2-1}\frac{(1 - e^{-j\omega})^{L_1}\left(1 - e^{j\frac{M_1}{M_2}\omega}\right)^{L_2}}{(j\omega)^{L_1+L_2}}\right\}(t) \\ &= (-1)^{L_2}\left(\frac{M_2}{M_1}\right)^{L_2-1}\mathcal{F}^{-1}\left\{\hat{\Delta}_1^{L_1}(\omega)\hat{\Delta}_{-\frac{M_2}{M_1}}^{L_2}(\omega)\zeta^{L_1+L_2-1}(\omega)\right\}(t) \\ &= \left(\frac{M_2}{M_1}\right)^{L_2-1}\sum_{l=0}^{L_1}\sum_{k=0}^{L_2}\binom{L_1}{l}\binom{L_2}{k}(-1)^{l+k+L_2}\zeta^{L_1+L_2-1}\left(t + \frac{kM_1}{M_2} - l\right).\end{aligned}\tag{A.5.5}$$

■

A.5.3 Proof of Proposition A.2.4

The derivative of an exponential B-spline that contains a vanishing pole is given by $\dot{\beta}_{\boldsymbol{\alpha}\cup\{0\}} = \Delta\beta_{\boldsymbol{\alpha}}$. Let $\varphi_1 = \beta_{\boldsymbol{\alpha}}$ and $\varphi_2 = \dot{\varphi}_1$. Using (A.2.6), we compute

$$a_{12}(t) = \left(\beta_{\boldsymbol{\alpha}} * \dot{\beta}_{\boldsymbol{\alpha}}(\cdot)\right)(t)$$

$$\begin{aligned} &= -(\beta_{\alpha} * \Delta\beta_{\alpha \setminus \{0\}}(-\cdot))(t) \\ &= -\left(\prod_{n=1}^{L-1} e^{\alpha_n^*}\right) \Delta\beta_{\alpha \cup (-\alpha^* \setminus \{0\})}(t + L - 1). \end{aligned} \tag{A.5.6}$$

■

Appendix B

Periodic Splines and Gaussian Processes for the Resolution of Linear Inverse Problems

This chapter deals with the resolution of inverse problems in a periodic setting or, in other terms, the reconstruction of periodic continuous-domain signals from their noisy measurements. We focus¹ on two reconstruction paradigms: variational and statistical. In the variational approach, the reconstructed signal is solution to an optimization problem that establishes a tradeoff between fidelity to the data and smoothness conditions via a quadratic regularization associated to a linear operator. In the statistical approach, the signal is modeled as a stationary random process defined from a Gaussian white noise and a whitening operator; one then looks for the optimal estimator in the mean-square sense. We give a generic form of the reconstructed signals for both approaches, allowing for a rigorous comparison of the two. We fully characterize the conditions under which the two formulations yield the same solution, which is a periodic spline in the case of sampling measurements. We also show that this equivalence between the two approaches remains valid on simulations for a broad class of problems. This extends the practical range of

¹This work is based on our publication [76], in collaboration with J. Fageot and M. Unser.

applicability of the variational method.

B.1 Introduction

This chapter deals with inverse problems: one aims at recovering an unknown signal from its corrupted measurements. To be more specific, the motivation of this work is the reconstruction of an unknown *continuous-domain* and *periodic* signal f from its M noisy measurements $y_m \approx \langle \nu_m, f \rangle = \int_0^1 \nu_m(t) f(t) dt$ for $m = 1 \dots M$, where the ν_m are measurement functions. The goal is then to build an output signal f_{opt} that is as close as possible to f .

B.1.1 Inverse Problems in the Continuous Domain

Inverse problems are often formulated in the discrete domain [198–202]. This is motivated by the need of manipulating digital data on computers. Nevertheless, many naturally occurring signals depend on continuous variables (*e.g.*, time or position). This leads us to attempt recovering a signal $f_{\text{opt}}(t)$ that depends on the continuous variable $t \in [0, 1]$. In contrast with the classical discrete setting, our search space for this reconstructed signal is thus infinite-dimensional [203]. Moreover, we choose a regularization based on true derivatives (as opposed to finite differences) to impose some smoothness on the reconstructed signal, a concept that is absent in the discrete setting.

When considering continuous-domain reconstruction methods, a majority of works, typically in machine learning, deal with sampling measurements. The goal is then to recover f from its (possibly noisy) values $y_m \approx f(t_m)$ at fixed location t_m . In order to investigate a more general version of inverse problems, we shall consider generalized measurements [204, 205]. They largely exceed the sampling case and include Fourier sampling or convolution (*e.g.*, MRI, x-ray tomography [206, 207]). Our only requirement is that the measurements y_m depend linearly on, and evolve continuously with, the unknown signal f up to some additive noise, so that $y_m \approx \langle \nu_m, f \rangle$.

B.1.2 Variational vs. Statistical Methods

In the discrete domain, two standard strategies are used to reconstruct an input signal \mathbf{x} from its noisy measurements $\mathbf{y} \approx \mathbf{H}\mathbf{x}$, where \mathbf{H} models the acquisition process [202]. The first approach is deterministic and can be tracked back to the '60s with Tikhonov's seminal work [85]. The ill-posedness of the problem usually imposes the addition of a regularizer. By contrast, Wiener filtering is based on the stochastic modelization of the signals of interest and the optimal estimation of the targeted signal \mathbf{x} . This chapter generalizes these ideas for the reconstruction of *continuous* signals from their *discrete* measurements.

In the variational setting, the reconstructed signal is a solution to an optimization problem that imposes some smoothness conditions [208]. More precisely, the optimization problem may take the form

$$f_{\text{opt}} = \arg \min_f \left(\sum_{m=1}^M (y_m - \langle \nu_m, f \rangle)^2 + \lambda \|Lf\|_{L_2}^2 \right), \quad (\text{B.1.1})$$

where L is a linear operator. The first term in (B.1.1) controls the data fidelity. The regularization term $\|Lf\|_{L_2}^2$ constrains the function to satisfy certain smoothness properties (for this reason, the variational approach is sometimes called a smoothing approach). The parameter λ in (B.1.1) quantifies the tradeoff between the fidelity to the data and the regularization constraint.

In the statistical setting, the signal is modeled as a random process and is optimally reconstructed using estimation theory [86]. More precisely, one assumes that the continuous-domain signal is the realization of a stochastic process s and that the samples are given by $y_m = \langle \nu_m, s \rangle + \epsilon_m$, where ϵ_m is a random perturbation and ν_m a linear measurement function. In this case, one specifies the reconstructed signal as the optimal statistical estimator in the mean-square sense

$$f_{\text{opt}} = \arg \min_{\tilde{s}} \mathbb{E} [\|s - \tilde{s}(\cdot|\mathbf{y})\|_{L_2}^2], \quad (\text{B.1.2})$$

where the estimators $t \mapsto \tilde{s}(t|\mathbf{y})$ are computed from the generalized samples y_m . The solution depends on the measurement function ν_m and the stochastic models specified for s and ϵ_m . In our case, the random process s is characterized by a linear operator L that is assumed to have a whitening effect (it transforms s into a periodic Gaussian white noise, see Definition B.4.1), while the perturbation is i.i.d. Gaussian.

B.1.3 Periodic and General Setting

The variational and statistical approaches have been extensively studied for continuous-domain signals defined on the infinitely supported real line. However, it is often assumed in practice that the input signals are periodic. In fact, a standard computational approach to signal processing is to extend by periodization the signals of otherwise bounded support. Periodic signals arise also naturally in applications such as the parametric representation of closed curves [64, 70, 209]. This has motivated the development of signal-processing tools and techniques specialized to periodic signals in sampling theory, error analysis, wavelets, stochastic modelization, or curve representation [75, 151, 210–214].

In this chapter, we develop the theory of the variational and statistical approaches for periodic continuous-domain signals in a very general context, including the following aspects:

- We consider a broad class of measurement functions, with the only assumptions that they are linear and continuous.
- Both methods refer to an underlying linear operator L that affects the smoothness properties of the reconstruction. We deal with a very broad class of linear operators acting on periodic functions.
- We consider possibly non-quadratic data fidelity terms in the smoothing approach.

B.1.4 Related Works

The topics investigated in this chapter have already received some attention in the literature, mostly in the non-periodic setting.

Reconstruction over the Real Line: Optimization problems of the form (B.1.1) appear in many fields and receive different names, including inverse problems in image processing [202], representer theorems in machine learning [215], or sometimes interpolation elsewhere. Schoenberg was the first to show the connection between (B.1.1) and spline theory [216]. Since then, this has been extended to other operators [217], or to the interpolation of the derivative of the signal [51, 218]. Many recent methods deal with non-quadratic regularization, especially for the reconstruction of sparse discrete [219, 220] or

continuous signals [203, 221–223]. We discuss this aspect more extensively in Section B.6.2.

A statistical framework requires the specification of the noise and of the signal stochastic model. The signal is then estimated from its measurements. A classical measure of the quality of an estimator is the mean-square error. This criterion is minimized by the minimum mean-square error (MMSE) estimator [86, 224]. The theory has been developed mostly for Gaussian processes and in the context of sampling measurements [87]. We are especially interested in innovation models, for which one assumes that the signal can be whitened (*i.e.*, transformed into a white noise) by the application of a linear operator [225, 226]. Non-periodic models have been studied in many situations, including the random processes associated with differential [227, 228] or fractional operators [229]. Extensions to non-Gaussian models are extensively studied by Unser and Tafti [230].

The statistical and variational frameworks are deeply connected. It is remarkable that the solution of either problem can be expressed as spline functions in relation with the linear operator L involved in regularization (variational approach) or whitening (statistical approach). Wahba has shown that the two approaches are strictly equivalent in the case of stationary Gaussian models [88]. This equivalence has also been recognized by several authors since then, as shown by Berlinet and Thomas-Agnan [87], and Unser and Blu [89]. In the non-stationary case, this equivalence is not valid any more and the existence of connections has received less attention.

Reconstruction of Periodic Signals: Some strong practical concerns have motivated the need for an adaptation of the theory to the periodic setting. Important contributions in that direction have been proposed. Periodic splines are constructed and applied to sampling problems by Schoenberg [231] and Golomb [232]. The smoothing spline approach is studied in the periodic setting by Wahba [88] for derivative operators of any order. Although the periodic extension of the classical theory is briefly mentioned by several authors [87, 88, 233], we are not aware of a global treatment. Providing a general analysis in the periodic setting is precisely what we propose in this chapter.

B.1.5 Outline and Main Contributions

Section B.2 contains the main notations and tools for periodic functions and operators. In Section B.3, we state the periodic representer theorem (Theorem B.3.4). It fully specifies the form of the solution in the variational approach in a very general setting. For the specific case of sampling measurements, we show that this solution is a periodic spline (Proposition B.3.6). Section B.4 is dedicated to the statistical approach. We introduce a class of periodic stationary processes (the *Gaussian bridges*) for which we specify the MMSE estimator in the case of generalized linear measurements (Theorem B.4.4). We also provide a theoretical comparison between the variational and statistical approaches by reformulating the MMSE estimation as the solution of a new optimization problem (Proposition B.4.5). This highlights the strict equivalence of the two approaches for invertible operators and extends known results from sampling to generalized linear measurements. For non-invertible operators, we complete our analysis with simulations in Section B.5. In particular, we give empirical evidence of the practical relevance of the variational approach for the reconstruction of periodic stationary signals. We provide in Section B.6 a comparison between our results in the periodic setting and the known results over the real line. Finally, we conclude in Section B.7. All the proofs have been postponed to the Appendix sections.

B.2 Mathematical Background for Periodic Signals

Throughout the chapter, we consider periodic functions and random processes. Without loss of generality, the period can always be normalized to one. Moreover, we identify a periodic function over \mathbb{R} with its restriction to a single period, chosen to be $\mathbb{T} = [0, 1)$. We use the symbols f , s , and \tilde{s} to specify a function, a random process, and an estimator of s , respectively.

We call $\mathcal{S}(\mathbb{T})$ the space of 1-periodic functions that are infinitely differentiable, $\mathcal{S}'(\mathbb{T})$ the space of 1-periodic generalized functions (dual of $\mathcal{S}(\mathbb{T})$), and $L_2(\mathbb{T})$ the Hilbert space of square integrable 1-periodic functions associated with the norm $\|f\|_{L_2} = (\int_0^1 |f(t)|^2 dt)^{1/2}$. Working with $\mathcal{S}'(\mathbb{T})$ allows us to deal with functions with no pointwise interpretation, such as the Dirac comb defined by

$$\text{III} = \sum_{k \in \mathbb{Z}} \delta(\cdot - k), \quad (\text{B.2.1})$$

where δ is the Dirac impulse. The duality product between an element $f \in \mathcal{S}'(\mathbb{T})$ and a smooth function $g \in \mathcal{S}(\mathbb{T})$ is denoted by $\langle f, g \rangle$. For instance, $\langle \text{III}, g \rangle = g(0)$ for every g . When the two real functions are in $L_2(\mathbb{T})$, we simply have the usual scalar product $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$. All these concepts are extended to complex-valued functions in the usual manner with the convention that $\langle f, g \rangle = \int_0^1 f(t)\overline{g(t)}dt$ for square-integrable functions. The complex sinusoids are denoted by $e_k(t) = e^{j2\pi kt}$ for any $k \in \mathbb{Z}$ and $t \in \mathbb{T}$. Any periodic generalized function $f \in \mathcal{S}'(\mathbb{T})$ can be expanded as

$$f(t) = \sum_{k \in \mathbb{Z}} \widehat{f}[k]e^{j2\pi kt} = \sum_{k \in \mathbb{Z}} \widehat{f}[k]e_k(t), \tag{B.2.2}$$

where the $\widehat{f}[k]$ are the Fourier coefficients of f , given by $\widehat{f}[k] = \langle f, e_k \rangle$. Finally, the convolution between two periodic functions f and g is given by

$$(f * g)(t) = \langle f, g(t - \cdot) \rangle. \tag{B.2.3}$$

If $f, g \in L_2(\mathbb{T})$, we have that $(f * g)(t) = \int_0^1 f(\tau)g(t - \tau)d\tau$.

B.2.1 Linear and Shift-Invariant Operators

Let L be a linear, shift-invariant (LSI), and continuous operator from $\mathcal{S}(\mathbb{T})$ to $\mathcal{S}'(\mathbb{T})$. The shift invariance implies the existence of $\widehat{L}[k] \in \mathbb{C}$ such that

$$Le_k = \widehat{L}[k]e_k, \tag{B.2.4}$$

for any $k \in \mathbb{Z}$. We call $\widehat{L}[k]$ the frequency response of the operator L ; it is also given by

$$\widehat{L}[k] = \langle L\{\text{III}\}, e_k \rangle = \int_0^1 L\{\text{III}\}(t)e^{-j2\pi kt}dt. \tag{B.2.5}$$

The sequence $(\widehat{L}[k])$ is the Fourier series of the periodic generalized function $L\{\text{III}\}$, and is therefore of slow growth [234, Chapter VII]. This implies that L , a priori from $\mathcal{S}(\mathbb{T})$ to $\mathcal{S}'(\mathbb{T})$, actually continuously maps $\mathcal{S}(\mathbb{T})$ into itself. This is a significant difference with the non-periodic setting — we discuss this point in the conclusion in Section B.7. Therefore, one can extend it by duality from $\mathcal{S}'(\mathbb{T})$ to $\mathcal{S}'(\mathbb{T})$. Then, for every $f \in \mathcal{S}'(\mathbb{T})$, we easily obtain from (B.2.4) that

$$Lf(t) = \sum_{k \in \mathbb{Z}} (\widehat{Lf})[k]e_k(t), \text{ where } (\widehat{Lf})[k] = \widehat{f}[k]\widehat{L}[k]. \tag{B.2.6}$$

The null space of L is $\mathcal{N}_L = \{f \in \mathcal{S}'(\mathbb{T}) \mid Lf = 0\}$. We shall only consider operators whose null space is finite-dimensional, in which case \mathcal{N}_L can only be made of linear combinations of sinusoids at frequencies that are annihilated by L . We state this fact in Proposition B.2.1 and prove it in Appendix B.8.1.

Proposition B.2.1. *Let L be a continuous LSI operator. If L has a finite-dimensional null space \mathcal{N}_L of dimension N_0 , then the null space is of the form*

$$\mathcal{N}_L = \text{span}\{e_{k_n}\}_{n=1}^{N_0}, \quad (\text{B.2.7})$$

where the $k_n \in \mathbb{Z}$ are distinct.

From (B.2.4) and (B.2.7), we deduce that $\widehat{L}[k] = 0$ if and only if $k = k_n$ for some $n \in [1 \dots N_0]$. In the following, we consider real-valued operators. In that case, we have the Hermitian symmetry $\overline{\widehat{L}[-k]} = \widehat{L}[k]$. Moreover, $e_{k_n} \in \mathcal{N}_L$ if and only if $e_{-k_n} \in \mathcal{N}_L$. The orthogonal projection of f on the null space \mathcal{N}_L is given by

$$\text{Proj}_{\mathcal{N}_L}\{f\} = \sum_{n=1}^{N_0} \widehat{f}[k_n] e_{k_n}. \quad (\text{B.2.8})$$

Let $\mathcal{K}_L = \mathbb{Z} \setminus \{k_n\}_{n \in \{1 \dots N_0\}}$. Then, (B.2.2) can be re-expressed as $f = \text{Proj}_{\mathcal{N}_L}\{f\} + \sum_{k \in \mathcal{K}_L} \widehat{f}[k] e_k$ and we have that $Lf(t) = \sum_{k \in \mathcal{K}_L} \widehat{f}[k] \widehat{L}[k] e_k(t)$, which yields the Parseval relation

$$\int_0^1 |Lf(t)|^2 dt = \sum_{k \in \mathcal{K}_L} |\widehat{f}[k]|^2 |\widehat{L}[k]|^2. \quad (\text{B.2.9})$$

B.2.2 Periodic L-Splines

Historically, splines are functions defined to be piecewise polynomials [235]. A spline is hence naturally associated to the derivative operator of a given order [84] in the sense that, for a fixed $N \geq 1$, a spline function $f : \mathbb{R} \rightarrow \mathbb{R}$ satisfies $Lf(t) = \sum a_m \delta(t - t_m)$ with $L = D^N$ the N th derivative. Splines have been extended to differential [81, 236–238], fractional [217, 239] or, more generally, spline-admissible operators [230]. We adapt here this notion to the periodic setting, where the Dirac impulse δ is replaced by the Dirac comb III .

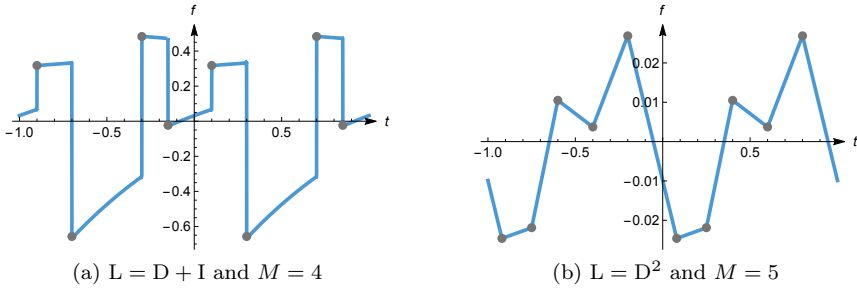


Figure B.1: Illustrations of periodic L-splines. Dots: nodes $(t_m, f(t_m))$. The spline in (a) corresponds to the periodization of an exponential B-spline (see Figure 1 in [81]).

Definition B.2.2. Consider an LSI operator L with finite-dimensional null space. We say that a function f is a periodic L-spline if

$$Lf(t) = \sum_{m=1}^M a_m \text{III}(t - t_m) \tag{B.2.10}$$

for some integer $M \geq 1$, weights $a_m \in \mathbb{R}$, and knot locations $t_m \in \mathbb{T}$.

Periodic L-splines play a crucial role in the variational and statistical approaches for the resolution of inverse problems in the periodic setting. We represent some periodic splines associated to different operators in Figure B.1.

B.3 Periodic Representer Theorem

We now consider a continuous LSI operator L with finite-dimensional null space \mathcal{N}_L . Let ν be the vector of the linear measurement functions ν_1, \dots, ν_M . They usually are of the form $\nu_m = \delta(\cdot - t_m)$ for time-domain sampling problems. Here, we consider general linear measurements to include any kind of inverse problems. In this section, our goal is to recover a function f from observed data $\mathbf{y} = (y_1, \dots, y_M)$

such that $y_m \simeq \langle \nu_m, f \rangle$. To do so, we consider the variational problem

$$\min_f \left(F(\mathbf{y}, \boldsymbol{\nu}(f)) + \lambda \|L f\|_{L_2}^2 \right), \quad (\text{B.3.1})$$

where $F : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^+$ is a strictly convex and continuous function called the *cost function*. This function controls the fidelity to data. A special attention will be given to the quadratic data fidelity of the form

$$F(\mathbf{y}, \boldsymbol{\nu}(f)) = \sum_{m=1}^M (y_m - \langle \nu_m, f \rangle)^2. \quad (\text{B.3.2})$$

We give the solution of (B.3.1) for the space of 1-periodic functions in Theorem B.3.4. To derive this solution, we first introduce and characterize the space of functions on which (B.3.1) is well-defined.

B.3.1 Search Space

The optimization problem (B.3.1) deals with functions such that $L f$ is square-integrable, which leads us to introduce $\mathcal{H}_L = \{f \in \mathcal{S}'(\mathbb{T}) \mid L f \in L_2(\mathbb{T})\}$. Due to (B.2.9), we have that

$$\mathcal{H}_L = \{f \in \mathcal{S}'(\mathbb{T}) \mid \sum_{k \in \mathcal{K}_L} |\hat{f}[k]|^2 |\hat{L}[k]|^2 < +\infty\}. \quad (\text{B.3.3})$$

Similar constructions have been developed for functions over \mathbb{R} or for sequences by Unser *et al.* [222, 240]. We now identify a natural Hilbertian structure on \mathcal{H}_L . If $L : \mathcal{H}_L \rightarrow L_2(\mathbb{T})$ is invertible, then \mathcal{H}_L inherits the Hilbert-space structure of L_2 via the norm $\|L f\|_{L_2}$. However, when L has a nontrivial null space, $\|L f\|_{L_2}$ is only a semi-norm, in which case there exists $f \neq 0$ (any element of the null space of L) such that $\|L f\|_{L_2} = 0$. To obtain a *bona fide* norm, we complete the semi-norm with a special treatment for the null-space components in Proposition B.3.1.

Proposition B.3.1. *Let L be a continuous LSI operator whose finite-dimensional null space is defined by $\mathcal{N}_L = \text{span}\{e_{k_n}\}_{n=1}^{N_0}$. We fix $\gamma^2 > 0$. Then, \mathcal{H}_L is a Hilbert space for the inner product*

$$\langle f, g \rangle_{\mathcal{H}_L} = \langle L f, L g \rangle + \gamma^2 \sum_{n=1}^{N_0} \hat{f}[k_n] \overline{\hat{g}[k_n]}. \quad (\text{B.3.4})$$

The proof is given in Appendix B.8.2. We have that $\|f\|_{\mathcal{H}_L}^2 = \|Lf\|_{L_2}^2 + \gamma^2 \|\text{Proj}_{\mathcal{N}_L}\{f\}\|_{L_2}^2$, where $\text{Proj}_{\mathcal{N}_L}\{f\}$ is given by (B.2.8). The coefficient γ^2 balances the contribution of both terms.

B.3.2 Periodic Reproducing-Kernel Hilbert Space

Reproducing-kernel Hilbert spaces (RKHS) are Hilbert spaces on which the evaluation maps $f \mapsto f(t)$ are well-defined, linear, and continuous. In this section, we answer the question of when the Hilbert space \mathcal{H}_L associated to an LSI operator L with finite-dimensional null space is a RKHS. This property is relevant to us because periodic function spaces that are RKHS are precisely the ones for which one can use measurement functions of the form $\nu_m = \text{III}(\cdot - t_m)$ in (B.3.1).

Definition B.3.2. *Let $\mathcal{H} \subseteq \mathcal{S}'(\mathbb{T})$ be a Hilbert space of 1-periodic functions and \mathcal{H}' be its dual. Then, we say that \mathcal{H} is a RKHS if the shifted Dirac comb $\text{III}(\cdot - t_0) \in \mathcal{H}'$ for any $t_0 \in \mathbb{T}$.*

This implies that any element f of a RKHS has a pointwise interpretation as a function $t \rightarrow f(t)$. As is well known, for any RKHS there exists a unique function $h : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{R}$ such that $h(\cdot, t_0) \in \mathcal{H}'$ and $\langle f, h(\cdot, t_0) \rangle = f(t_0)$, for every $t_0 \in \mathbb{T}$ and $f \in \mathcal{H}$. We call h the *reproducing kernel* of \mathcal{H} .

Proposition B.3.3. *Let L be a continuous LSI operator with finite-dimensional null space. The Hilbert space \mathcal{H}_L (see (B.3.3)) is a RKHS if and only if*

$$\sum_{k \in \mathcal{K}_L} \frac{1}{|\widehat{L}[k]|^2} < +\infty. \tag{B.3.5}$$

Then, the reproducing kernel for the scalar product (B.3.4) is given by $h(t, \tau) = h_\gamma(t - \tau)$, where $h_\gamma \in \mathcal{S}'(\mathbb{T})$ is

$$h_\gamma(t) = \sum_{n=1}^{N_0} \frac{e_{k_n}(t)}{\gamma^2} + \sum_{k \in \mathcal{K}_L} \frac{e_k(t)}{|\widehat{L}[k]|^2}. \tag{B.3.6}$$

The proof is given in Appendix B.8.3. Note that the reproducing kernel only depends on the difference $(t - \tau)$.

B.3.3 Periodic Representer Theorem

Now that we have defined the search space of the optimization problem (B.3.1), we derive the representer theorem that gives the explicit form of its unique periodic solution.

Theorem B.3.4. *We consider the optimization problem*

$$\min_{f \in \mathcal{H}_L} \left(F(\mathbf{y}, \nu(f)) + \lambda \|Lf\|_{L_2}^2 \right), \quad (\text{B.3.7})$$

where

- $F : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^+$ is strictly convex and continuous;
- L is an LSI operator with finite-dimensional null space;
- $\nu = (\nu_1, \dots, \nu_M) \in (\mathcal{H}'_L)^M$ such that $\mathcal{N}_L \cap \mathcal{N}_\nu = \{0\}$;
- $\mathbf{y} = (y_1, \dots, y_M) \in \mathbb{R}^M$ are the observed data; and
- $\lambda > 0$ is a tuning parameter.

Then, (B.3.7) has a unique solution of the form

$$f_{\text{RT}}(t) = \sum_{m=1}^M a_m \varphi_m(t) + \sum_{n=1}^{N_0} b_n e_{k_n}(t), \quad (\text{B.3.8})$$

where $a_m, b_n \in \mathbb{R}$, $\varphi_m = h_\gamma * \nu_m$, and h_γ is given by (B.3.6). Moreover, the vector $\mathbf{a} = (a_1, \dots, a_M)$ satisfies the relation $\mathbf{P}^\top \mathbf{a} = \mathbf{0}$, with \mathbf{P} the $(M \times N_0)$ matrix with entries $[\mathbf{P}]_{m,n} = \langle e_{k_n}, \nu_m \rangle$.

The proof of Theorem B.3.4 is given in Appendix B.8.4. The optimal solution depends on $(M + N_0)$ coefficients, but the condition $\mathbf{P}^\top \mathbf{a} = \mathbf{0}$ implies that there are only $(M + N_0 - N_0) = M$ degrees of freedom. In the case when F is quadratic of the form (B.3.2), the solution is made explicit in Proposition B.3.5.

Proposition B.3.5. *Under the conditions of Theorem B.3.4, if F is given by (B.3.2), then the vectors \mathbf{a} and \mathbf{b} satisfy the linear system*

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} \mathbf{G} + \lambda \mathbf{I} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}, \quad (\text{B.3.9})$$

where $\mathbf{P} \in \mathbb{C}^{M \times N_0}$ is defined by $[\mathbf{P}]_{m,n} = \langle e_{k_n}, \nu_m \rangle$ and $\mathbf{G} \in \mathbb{R}^{M \times M}$ is a Gram matrix such that

$$[\mathbf{G}]_{m_1, m_2} = \int_0^1 \int_0^1 \nu_{m_1}(t) h_\gamma(t - \tau) \nu_{m_2}(\tau) dt d\tau. \tag{B.3.10}$$

The proof is given in Appendix B.8.5. In the case of sampling measurements, we show moreover in Proposition B.3.6 that the optimal solution is a periodic spline in the sense of Definition B.2.2. We recall that such measurements are valid as soon as the search space \mathcal{H}_L is a RKHS, a situation that has been fully characterized in Proposition B.3.3.

Proposition B.3.6. *Under the conditions of Proposition B.3.5, if L satisfies (B.3.5) and if the measurements are of the form $\nu_m = \text{III}(\cdot - t_m)$, $t_m \in \mathbb{T}$, then the unique solution of (B.3.7) is a periodic (L^*L) -spline with weights a_m and knots t_m .*

The proof is given in Appendix B.8.6.

B.4 Periodic Processes and MMSE

In this section, we change perspective and consider the following statistical problem: given noisy measurements of a zero-mean and real periodic Gaussian process, we are looking for the optimal estimator (for the mean-square error) of the complete process over \mathbb{T} .

B.4.1 Non-Periodic Setting

In a non-periodic setting, it is usual to consider stochastic models where the random process s is a solution to the stochastic differential equation [230]

$$Ls = w, \tag{B.4.1}$$

where L is a linear differential operator and w a continuous domain (non-periodic) Gaussian white noise. When the null space of the operator is nontrivial, it is necessary to add boundary conditions such that the law of the process s is uniquely defined.

B.4.2 Gaussian Bridges

In the periodic setting, the construction of periodic Gaussian processes has to be adapted. We first introduce the notion of periodic Gaussian white noise, exploiting the fact that the law of a zero-mean periodic Gaussian process s is fully characterized by its covariance function $r_s(t, \tau)$ such that

$$\mathbb{E}[\langle s, f \rangle \langle s, g \rangle] = \int_0^1 \int_0^1 f(t) r_s(t, \tau) \overline{g(\tau)} dt d\tau. \quad (\text{B.4.2})$$

Definition B.4.1. A periodic Gaussian white noise² is a Gaussian random process w whose covariance is $r_w(t, \tau) = \text{III}(t - \tau)$.

For any periodic real function f , the random variable $\langle w, f \rangle$ is therefore Gaussian with mean 0 and variance $\|f\|_{L_2}^2$. Moreover, $\langle w, f \rangle$ and $\langle w, g \rangle$ are independent if and only if $\langle f, g \rangle = 0$. Hence, the Fourier coefficients $\widehat{w}[k] = \langle w, e_k \rangle$ of the periodic Gaussian white noise satisfy the following properties:

- $\widehat{w}[k] = \Re(\widehat{w}[k]) + j \Im(\widehat{w}[k]);$
- $\overline{\widehat{w}[-k]} = \widehat{w}[k];$
- $\Re(\widehat{w}[k]), \Im(\widehat{w}[k]) \sim \mathcal{N}(0, \frac{1}{2}), \forall k > 0;$
- $\widehat{w}[0] \in \mathbb{R}$ and $\widehat{w}[0] \sim \mathcal{N}(0, 1);$
- $\Re(\widehat{w}[k]), \Im(\widehat{w}[k]),$ and $\widehat{w}[0]$ are independent.

Put differently, for any nonzero frequency k , $\mathbb{E}[\widehat{w}[k]^2] = 0$ and $\mathbb{E}[\widehat{w}[k] \overline{\widehat{w}[k]}] = 1$. This means that $\widehat{w}[k]$, $k \neq 0$, follows a complex normal distribution with mean 0, covariance 1, and pseudo-covariance 0 [241]. When L has a nontrivial null space, there is no hope to construct a periodic process s solution of (B.4.1) with w a periodic Gaussian white noise. Indeed, the operator L kills the null-space frequencies, which contradicts that $\widehat{w}[k_n] \neq 0$ almost surely for $n = 1 \dots N_0$. One should adapt (B.4.1) accordingly by giving special treatment to the null-space frequencies. We propose

²Without loss of generality, we only consider Gaussian white noise with zero-mean and variance 1.

TABLE B.1: Gaussian bridges for several operators.

	D + I	D	D ² + 4π ² I	D ²
$\widehat{L}[k]$	$j2\pi k + 1$	$j2\pi k$	$4\pi^2(1 - k^2)$	$-4\pi^2 k^2$
\mathcal{N}_L	$\text{span}\{0\}$	$\text{span}\{e_0\}$	$\text{span}\{e_1, e_{-1}\}$	$\text{span}\{e_0\}$
Gaussian bridges $\gamma_0^2 = 1$				

here to consider a new class of periodic Gaussian processes: the *Gaussian bridges*. Given some operator L and $\gamma_0 > 0$, we set

$$L_{\gamma_0} = L + \gamma_0 \text{Proj}_{\mathcal{N}_L}, \tag{B.4.3}$$

where $\text{Proj}_{\mathcal{N}_L}$ is given by (B.2.8). Note that $L_{\gamma_0} = L$ for any γ_0 when the null space of L is trivial. Moreover, we remark that

$$\|L_{\gamma_0} f\|_{L_2}^2 = \|L f\|_{L_2}^2 + \gamma_0^2 \|\text{Proj}_{\mathcal{N}_L}\{f\}\|_{L_2}^2 = \|f\|_{\mathcal{H}_L}^2, \tag{B.4.4}$$

where $\|f\|_{\mathcal{H}_L}^2 = \langle f, f \rangle_{\mathcal{H}_L}$ is given in (B.3.4) (with $\gamma = \gamma_0$).

Definition B.4.2. A Gaussian bridge is a periodic Gaussian process s , solution to the stochastic differential equation

$$L_{\gamma_0} s = w, \tag{B.4.5}$$

with w a periodic Gaussian white noise and L_{γ_0} given by (B.4.3) for some LSI operator L with finite-dimensional null space and $\gamma_0 > 0$. We summarize this situation with the notation $s \sim \mathcal{GB}(L, \gamma_0^2)$. When the null space is trivial, in which case the parameter γ_0^2 is immaterial, we write $s \sim \mathcal{GB}(L)$.

The Gaussian-bridge terminology is inspired by the Brownian bridge, the periodic version of the Brownian motion³. Several realizations of our Gaussian bridges for

³Our definition differs from the classical one, in which the Brownian bridge is zero at the origin instead of being zero-mean [242].

various operators are shown in Table B.1 for $\gamma_0^2 = 1$. The influence of the parameter γ_0^2 is illustrated in Figure B.2.

Proposition B.4.3. *The covariance function of the Gaussian bridge $s \sim \mathcal{GB}(\mathbf{L}, \gamma_0^2)$ is*

$$r_s(t, \tau) = h_{\gamma_0}(t - \tau), \quad (\text{B.4.6})$$

where h_{γ_0} is defined in (B.3.6). It implies that

$$\mathbb{E}[\langle s, f \rangle \langle s, g \rangle] = \langle h_{\gamma_0} * f, g \rangle. \quad (\text{B.4.7})$$

In particular, we have that

$$\mathbb{E}[\widehat{s}[k]^2] = \widehat{h}_{\gamma_0}[k]. \quad (\text{B.4.8})$$

The proof of Proposition B.4.3 is given in Appendix B.8.7. An important consequence is that a Gaussian bridge is stationary since its covariance function only depends on the difference $(t - \tau)$.

B.4.3 Measurement Model and MMSE Estimator

For this section, we restrict ourselves to operators \mathbf{L} for which the native space $\mathcal{H}_{\mathbf{L}}$ is a RKHS. In that case, using (B.4.8) and (B.3.6), the Gaussian bridge s satisfies

$$\mathbb{E}[\|s\|_{L_2}^2] = \sum_{k \in \mathbb{Z}} \mathbb{E}[\widehat{s}[k]^2] = \sum_{k \in \mathcal{K}_{\mathbf{L}}} \frac{1}{|\widehat{L}[k]|^2} + \sum_{n=1}^{N_0} \frac{1}{\gamma_0^2}, \quad (\text{B.4.9})$$

which is finite according to (B.3.5). Therefore, the Gaussian bridge s is (almost surely) square-integrable.

The observed data \mathbf{y} are assumed to be generated as

$$\mathbf{y} = \langle \boldsymbol{\nu}, s \rangle + \boldsymbol{\epsilon}, \quad (\text{B.4.10})$$

where $s \sim \mathcal{GB}(\mathbf{L}, \gamma_0^2)$ is a Gaussian bridge (see Definition B.4.2), $\boldsymbol{\nu} = (\nu_1, \dots, \nu_M)$ is a vector of M linear measurement functions, and $\boldsymbol{\epsilon}$ are independent random perturbations such that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$. Given \mathbf{y} in (B.4.10), we want to find the estimator \widehat{s} of the Gaussian bridge s , imposing that it minimizes the quantity $\mathbb{E}[\|s - \widehat{s}\|_2^2]$.

Theorem B.4.4. Let $\mathbf{y} = (y_1, \dots, y_M)$ be the noisy measurement vector (B.4.10) of the Gaussian bridge $s \sim \mathcal{GB}(\mathbf{L}, \gamma_0^2)$, with measurement functions $\nu_m \in \mathcal{H}'_{\mathbf{L}}$, $m = 1 \dots M$. Then, the MMSE estimator of s given the samples $\{y_m\}_{m \in [1 \dots M]}$ is

$$\tilde{s}_{\text{MMSE}}(t) = \sum_{m=1}^M d_m \varphi_m(t), \tag{B.4.11}$$

where $\varphi_m = h_{\gamma_0} * \nu_m$ with $\nu_m \in \mathcal{H}'_{\mathbf{L}}$, $\mathbf{d} = (d_1, \dots, d_M) = (\mathbf{G} + \sigma_0^2 \mathbf{I})^{-1} \mathbf{y}$, and \mathbf{G} is the Gram matrix defined in (B.3.9).

The proof is given in Appendix B.8.8. Theorem B.4.4 can be seen as a generalization of the classical Wiener filtering, designed for discrete signals, to the hybrid case where the input signal is in a (periodic) continuous-domain and the (finite-dimensional) measurements are discrete. A leading theme of this chapter is that the form of the MMSE estimator \tilde{s}_{MMSE} is very close to the one of the solution of the representer theorem f_{RT} with $\lambda = \sigma_0^2$ and for a quadratic cost function. This connection is exploited in Section B.4.4.

B.4.4 MMSE Estimation as a Representer Theorem

The MMSE estimator given in Theorem B.4.4 can be interpreted as the solution of the optimization problem described in Proposition B.4.5.

Proposition B.4.5. Consider an LSI operator \mathbf{L} with finite-dimensional null space, $\gamma > 0$, and $\nu_m \in \mathcal{H}'_{\mathbf{L}}$ for $m = 1 \dots M$. We set \mathbf{L}_γ as in (B.4.3). Then, the solution of the optimization problem

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{m=1}^M (y_m - \langle f, \nu_m \rangle)^2 + \lambda \| \mathbf{L}_\gamma f \|_{L_2}^2 \right) \tag{B.4.12}$$

exists, is unique, and given by

$$f_{\text{opt}}(t) = \sum_{m=1}^M d_m \varphi_m(t), \tag{B.4.13}$$

where $\varphi_m = h_\gamma * \nu_m$ and $\mathbf{d} = (d_1, \dots, d_M) = (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{y}$. In particular, the unique minimizer of (B.4.12) is the MMSE estimator given in Theorem B.4.4 for $\lambda = \sigma_0^2$ and $\gamma = \gamma_0$.

The proof of Proposition B.4.5 follows the same steps as the ones of Theorem B.3.4 (form of the minimizer for the periodic representer theorem) and Proposition B.3.5 (explicit formulas in terms of system matrix for the vectors \mathbf{a} and \mathbf{b}), with significant simplifications that are detailed in Appendix B.8.9. Proposition B.4.5 has obvious similarities with Theorem B.3.4, but it also adds new elements.

- Proposition B.4.5 gives an interpretation of the MMSE estimator of a Gaussian bridge given its measurements as the solution to an optimization problem. This problem is very close to the periodic representer theorem (Theorem B.3.4) for a quadratic cost function. However, (B.4.12) differs from (B.3.7) because the regularization also penalizes null-space frequencies.
- If the null space $\mathcal{N}_{\mathbf{L}}$ is trivial, then

$$f_{\text{RT}} = \tilde{s}_{\text{MMSE}} \quad (\text{B.4.14})$$

for $\lambda = \sigma_0^2$. This means that Theorem B.3.4 (smoothing approach) and B.4.4 (statistical approach) correspond to the same reconstruction method. This equivalence is well-known for stationary processes on \mathbb{R} in the case of time-domain sampling measurements [88]. Our results extend this to the periodic setting and to the case of generalized linear measurements.

- If the null space is nontrivial, then Theorem B.3.4 and Proposition B.4.5 yield different reconstructions. In particular, this implies that one cannot interpret the optimizer f_{RT} in Theorem B.3.4 as the MMSE estimator of a Gaussian bridge. Yet, the solutions get closer and closer as $\gamma_0 \rightarrow 0$. In Section B.5, we investigate more deeply this situation.

B.5 Quality of the Estimators on Simulations

We consider $\tilde{s}_{\gamma,\lambda}(t|\mathbf{y}) = \sum_{m=1}^M d_m \varphi_m(t)$ as the linear estimator of s given \mathbf{y} , where $\varphi_m = h_\gamma * \nu_m$, $\mathbf{d} = (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{y}$, and \mathbf{G} is defined in Proposition B.3.5. To simplify notations, we shall omit \mathbf{y} when considering $\tilde{s}_{\gamma,\lambda}(\cdot|\mathbf{y}) = \tilde{s}_{\gamma,\lambda}$. Each pair (λ, γ) gives an estimator. In particular, if s is a Gaussian bridge, then $\tilde{s}_{\text{MMSE}} = \tilde{s}_{\gamma_0, \sigma_0^2}$, according to Theorem B.4.4. The mean-square error (MSE) of $\tilde{s}_{\gamma,\lambda}$ over N experiments is computed as $\text{MSE} = \frac{1}{N} \sum_{n=1}^N \|s_n - (\tilde{s}_{\gamma,\lambda})_n\|_{L_2}^2$, where the s_n are independent

realizations of s that yield a new noisy measurement \mathbf{y}_n and $(\tilde{s}_{\gamma,\lambda})_n = \tilde{s}_{\gamma,\lambda}(\cdot|\mathbf{y}_n)$ is the estimator based on \mathbf{y}_n . We define the normalized mean-square error (NMSE) by

$$\text{NMSE} = \frac{\text{MSE}}{\frac{1}{N} \sum_{n=1}^N \|s_n\|_{L_2}^2} \approx \frac{\mathbb{E}[\|s - \tilde{s}_{\gamma,\lambda}\|_{L_2}^2]}{\mathbb{E}[\|s\|_{L_2}^2]}. \quad (\text{B.5.1})$$

In this section, we first detail the generation of Gaussian bridges (Section B.5.1). We then investigate the role of the parameters λ (Section B.5.2) and γ^2 (Section B.5.3) on the quality of the estimator $\tilde{s}_{\gamma,\lambda}$. We primarily focus on time-domain sampling measurements with $\langle \boldsymbol{\nu}, s \rangle = (s(t_1), \dots, s(t_M))^T$, where the t_m are in \mathbb{T} .

B.5.1 Generation of Gaussian Bridges

We first fix the operator L with null space \mathcal{N}_L of dimension N_0 and $\gamma_0 > 0$. Then, we generate $(2N_{\text{coef}} + 1)$ Fourier coefficients $\{\hat{w}[k]\}_{k \in [-N_{\text{coef}} \dots N_{\text{coef}}]}$ of a Gaussian white noise according to Definition B.4.1. Finally, we compute the Gaussian bridge s as

$$s(t) = \sum_{\substack{k \in \mathcal{K}_L \\ |k| \leq N_{\text{coef}}}} \frac{\hat{w}[k]}{\hat{L}[k]} e_k(t) + \sum_{n=1}^{N_0} \frac{\hat{w}[k_n]}{\gamma_0} e_{k_n}(t). \quad (\text{B.5.2})$$

Since $N_0 < \infty$, (B.5.2) provides a mere approximation of the Gaussian bridge. However, the approximation error can be made arbitrarily small by taking N_{coef} large enough. In Figure B.2, we generate $s \sim \mathcal{GB}(\text{D} + 4\pi^2\text{I}, \gamma_0^2)$ for four values of γ_0^2 . For small values of γ_0^2 , the null-space component dominates, which corresponds in this case to the frequency $|k| = 1$. When γ_0^2 increases, the null-space component has a weaker influence.

B.5.2 Influence of λ

We evaluate the influence of the parameter λ for the case of the invertible operator $L = \text{D} + \text{I}$. In this case we have that $\text{Proj}_{\mathcal{N}_L} = 0$ (since $\mathcal{N}_L = \{0\}$), which simplifies (B.4.3). Hence, the parameter γ_0^2 is immaterial and we denote by \tilde{s}_λ the estimator associated to $\lambda > 0$. We consider $s \sim \mathcal{GB}(\text{D} + \text{I})$ and $\sigma_0^2 = 10^{-2}$.

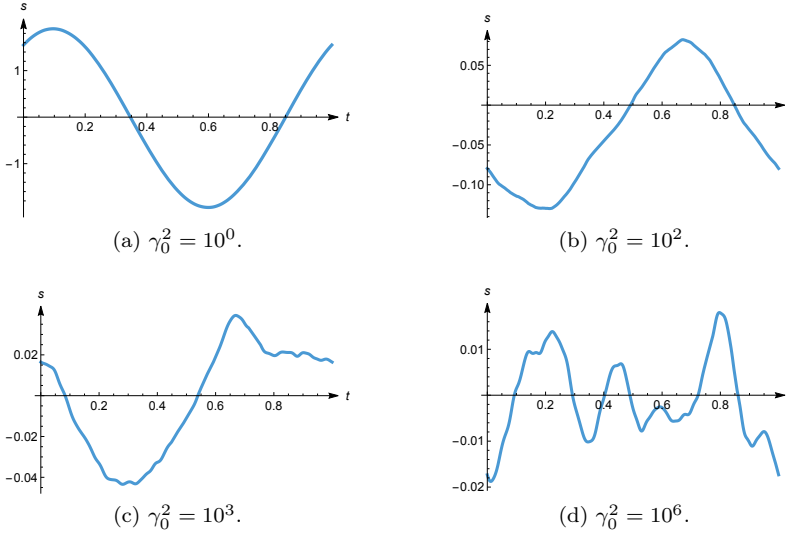


Figure B.2: Illustration of $s \sim \mathcal{GB}(D^2 + 4\pi^2 I, \gamma_0^2)$ for different values of γ_0^2 .

Time-Domain Sampling Measurements

We generated $N = 500$ realizations of s . From each one, we extracted $M = 30$ noisy measurements. We then computed 30 estimators $\{(\tilde{s}_\lambda)_n\}_{\lambda \in \mathcal{L}_1}$, where \mathcal{L}_1 is the set of values obtained by uniform sampling of the interval $[0.001, 0.03]$. The plot of the NMSE (approximated according to (B.5.1)) as a function of λ is given in Figure B.3 (a). The minimum error is obtained for $\lambda \simeq 0.01$, which corresponds to σ_0^2 . This result validates the theory presented in Theorem B.4.4. Actually, when λ is small, the estimator interpolates the noisy measurements while, for a large λ , the estimator tends to oversmooth the curve. The MMSE estimator makes an optimal tradeoff between fitting the data and smoothing the curve. These observations about λ retain their validity for other operators, including noninvertible ones.

Fourier-Domain Sampling Measurements

We consider complex exponential measurement functionals, inducing $\langle \boldsymbol{\nu}, s \rangle = (\widehat{s}[k_1], \dots, \widehat{s}[k_M])^\top$, where the k_m are in \mathbb{Z} . We define $\mathcal{N}_\boldsymbol{\nu} = \{k_m\}_{m=1 \dots M}$, such that $(-k_m) \in \mathcal{N}_\boldsymbol{\nu}$ for every $k_m \in \mathcal{N}_\boldsymbol{\nu}$. We consider the measurements $\boldsymbol{\nu} = (e_{k_1}, \dots, e_{k_M})$. Note that these measurement functionals are complex, which calls for a slight adaptation of the framework presented so far⁴. The noise $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_M)$ is then also complex and satisfies the properties:

- $\epsilon_m = \Re(\epsilon_m) + j \Im(\epsilon_m)$;
- $\epsilon_{m_1} = \overline{\epsilon_{m_2}}, k_{m_1} = -k_{m_2}$;
- $\Re(\epsilon_m), \Im(\epsilon_m) \sim \mathcal{N}(0, \frac{\sigma_0^2}{2}), \forall k_m \neq 0$;
- $\epsilon_m \in \mathbb{R}$ and $\epsilon_m \sim \mathcal{N}(0, \sigma_0^2), k_m = 0$;
- $\Re(\epsilon_m), \Im(\epsilon_m)$ and $\epsilon_{m_1}, k_{m_1} = 0$, are independent.

This means that $\mathbb{E}[|\epsilon_m|^2] = \sigma_0^2$ for every m .

We repeated the experiment done with the time-domain sampling using exactly the same procedure and parameters, and $\mathcal{N}_\boldsymbol{\nu} = \{-2, -1, 0, 1, 2\}$. The experimental curve of the evolution of the NMSE with λ is given in Figure B.3 (b). Again, the minimum is obtained for $\lambda \simeq 0.01 = \sigma_0^2$. We now want to compare this curve to the theoretical one.

For the Fourier-sampling case, we were also able to derive the corresponding closed-form formulas for the NMSE (B.5.1).

Proposition B.5.1. *Let s be a Gaussian bridge associated with an invertible operator L , and $y_m = \widehat{s}[k_m] + \epsilon_m, m = 1 \dots M$, with $k_m \in \mathcal{N}_\boldsymbol{\nu}$ the sampled frequencies and $\boldsymbol{\epsilon}$ a complex Gaussian noise with variance σ_0^2 as above. Then, the MSE of the estimator $\tilde{s}_\lambda = \tilde{s}_\lambda(\cdot | \mathbf{y})$ is given by*

$$\mathbb{E} [\|s - \tilde{s}_\lambda\|_{L_2}^2] = \sum_{m=1}^M \frac{\widehat{h}[k_m](\lambda^2 + \widehat{h}[k_m]\sigma_0^2)}{(\widehat{h}[k_m] + \lambda)^2} + \sum_{k \notin \mathcal{N}_\boldsymbol{\nu}} \widehat{h}[k], \quad (\text{B.5.3})$$

where h is the reproducing kernel of \mathcal{H}_L .

⁴One could equivalently consider cosine and sine measurements, to the cost of heavier formulas.

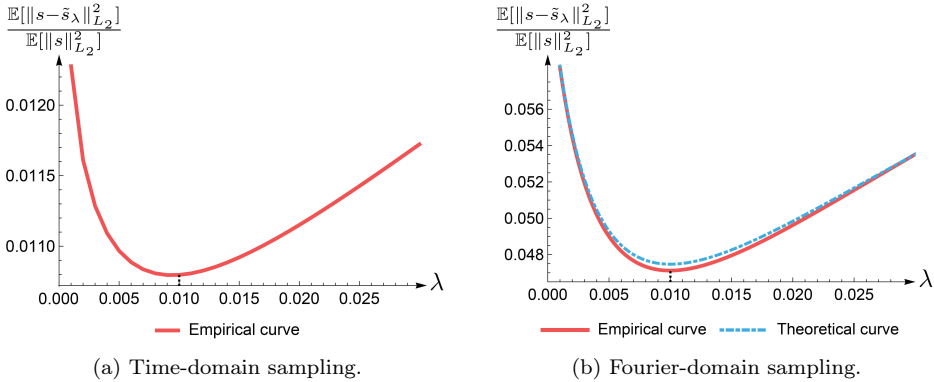


Figure B.3: Evolution of the NMSE in terms of λ for $s \sim \mathcal{GB}(D + I)$ for time and Fourier-domain sampling measurements.

The proof is given in Appendix B.8.10. Note that $\hat{h}[k] = 1/|\hat{L}[k]|^2$ is real-valued and strictly positive for every k . From (B.5.3), we also recover the property that the optimum is reached for $\lambda = \sigma_0^2$ since each of the M terms that appear in the first sum is minimized for this value of λ .

The theoretical curve for $\mathcal{N}_\nu = \{-2, -1, 0, 1, 2\}$ is given in Figure B.3 (b) and is in good agreement with the experimental curve. We explain the slight variation (0.15% for the L_2 -norm over $\lambda \in [0.001, 0.03]$) by the fact that (B.5.1) is only an estimation of the theoretical NMSE.

B.5.3 Influence of γ^2

In this section, we only consider noninvertible operators since invertibility has already been addressed in Section B.4.4 (see (B.4.14)). In order to evaluate the specific influence of γ , we set $\lambda = \sigma_0^2$. Hence, $\tilde{s}_{\gamma, \sigma_0^2} = \tilde{s}_\gamma$. We generated $N = 500$ realizations of a Gaussian bridge s , and from each one, we extracted $M = 30$ noisy measurements. We repeated this for several operators L and values of γ_0^2 and σ_0^2 . For each case, we compared \tilde{s}_{MMSE} to $\tilde{s}_{\gamma \rightarrow 0}$, $\tilde{s}_{\gamma \rightarrow \infty}$, and f_{RT} in (B.3.8), seen here as an additional estimator. The corresponding NMSEs (see (B.5.1)) are given in

Table B.2. We make four observations.

1. In each case, the best result is obtained with \tilde{s}_{MMSE} , as expected. We see, moreover, that

$$\lim_{\gamma \rightarrow 0} \mathbb{E}[\|s - \tilde{s}_\gamma\|_{L_2}^2] \simeq \mathbb{E}[\|s - f_{\text{RT}}\|_{L_2}^2].$$

This is in line with the fact that the functional (B.3.7) to minimize in Theorem B.3.4 corresponds to (B.4.12) with $\gamma = 0$.

2. For small values of γ_0^2 (*i.e.*, 10^{-3} or 10^0), we see that

$$\mathbb{E}[\|s - f_{\text{RT}}\|_{L_2}^2] \simeq \mathbb{E}[\|s - \tilde{s}_{\text{MMSE}}\|_{L_2}^2].$$

This means that the performances of \tilde{s}_{MMSE} and f_{RT} are very similar. This is illustrated in Figure B.4 (a), where \tilde{s}_{MMSE} and f_{RT} do coincide. Meanwhile, we see that

$$\lim_{\gamma \rightarrow \infty} \mathbb{E}[\|s - \tilde{s}_\gamma\|_{L_2}^2] \gg \mathbb{E}[\|s - \tilde{s}_{\text{MMSE}}\|_{L_2}^2].$$

This is also illustrated in Figure B.4 (a) for $L = D$. The reconstruction for $\gamma \rightarrow +\infty$ significantly fails to recover the original signal s , as the corresponding estimator tends to have zero-mean.

3. For intermediate values of γ_0^2 (*i.e.*, $\gamma_0^2 = 10^3$ or 10^6 according to σ_0 and the order of the operator), the minimal NMSE is obtained for \tilde{s}_{MMSE} only. We also observe that

$$\mathbb{E}[\|s - f_{\text{RT}}\|_{L_2}^2] < \lim_{\gamma \rightarrow \infty} \mathbb{E}[\|s - \tilde{s}_\gamma\|_{L_2}^2].$$

This is illustrated in Figure B.4 (b) for $L = D^2 + 4\pi^2\mathbf{I}$, $\gamma_0^2 = 10^6$ and $\sigma_0^2 = 10^{-4}$, where we can distinguish \tilde{s}_{MMSE} , $\tilde{s}_{\gamma \rightarrow \infty}$, and f_{RT} .

4. For large values of γ_0^2 (*i.e.*, $\gamma_0^2 = 10^9$), we observe that

$$\lim_{\gamma \rightarrow \infty} \mathbb{E}[\|s - \tilde{s}_\gamma\|_{L_2}^2] \simeq \mathbb{E}[\|s - \tilde{s}_{\text{MMSE}}\|_{L_2}^2]$$

and

$$\mathbb{E}[\|s - f_{\text{RT}}\|_{L_2}^2] > \mathbb{E}[\|s - \tilde{s}_{\text{MMSE}}\|_{L_2}^2].$$

In fact, for large γ_0^2 , the Gaussian bridge tends to have vanishing null-space frequencies (with (B.5.2), we have that $\widehat{s}[k_n] = \widehat{w}[k_n]/\gamma_0$ for $n = 1 \dots N_0$). Meanwhile, the reconstructed signal f_{RT} is not constrained to attenuate null-space frequencies. The null-space part in (B.3.8) is mainly responsible for a higher error compared to \tilde{s}_{MMSE} . This is highlighted in Figure B.4 (c).

Observations 2), 3), and 4) suggest the existence of three regimes. For further investigation, we present in Figure B.5 the evolution of NMSE as a function of $\log \gamma^2$ for $L = D$ and $\gamma_0^2 = 10^0, 10^3$, and 10^6 . The minimal error is always obtained for $\gamma^2 \simeq \gamma_0^2$, as predicted by the theory. For the three cases, we observe two plateaus: one for $\gamma^2 \in (0, v_1)$ and the other for $\gamma^2 \in (v_2, \infty)$, where $v_1, v_2 > 0$. It means that, for each value of γ_0^2 , the estimators \tilde{s}_γ with $\gamma^2 \in (0, v_1)$ ((v_2, ∞) , respectively) are very similar and the reconstruction algorithms are practically indistinguishable. The values of v_1 and v_2 depend on γ_0^2 . When $\gamma_0^2 = 10^0$ (10^6 , respectively), we have that $\gamma_0^2 \in (0, v_1)$ ((v_2, ∞) , respectively). However, $\gamma_0^2 = 10^3 \in [v_1, v_2]$ belongs to none of the plateaus.

Two main conclusions can be drawn from our experiments. First, we have strong empirical evidence that

$$\tilde{s}_\gamma \xrightarrow{\gamma \rightarrow 0} f_{\text{RT}}, \quad (\text{B.5.4})$$

which we conjecture to be true for any Gaussian-bridge model. This is remarkable because it presents the reconstruction based on the periodic representer theorem as a limit case of the statistical approach. Second, we empirically see that, for reasonably small values of γ_0^2 , the estimators corresponding to $\gamma^2 \leq \gamma_0^2$ are practically indistinguishable from the MMSE estimator. This is in particular valid for the representer-theorem reconstruction, for which we then have that

$$f_{\text{RT}} \approx \tilde{s}_{\text{MMSE}}. \quad (\text{B.5.5})$$

The variational method is *theoretically* suboptimal to reconstruct Gaussian bridges. However, based on our experiments, it is reasonable to consider this method as *practically* optimal for small values of γ_0^2 and $\lambda = \sigma_0^2$.

TABLE B.2: Comparison of NMSE for $\tilde{s}_{\gamma \rightarrow 0}$, f_{RT} , \tilde{s}_{MMSE} , and $\tilde{s}_{\gamma \rightarrow \infty}$ over $N = 500$ iterations. **Bold**: optimal result.

L	γ_0^2	$\sigma_0 = 10^{-1}$				$\sigma_0 = 10^{-2}$			
		$\tilde{s}_{\gamma \rightarrow 0}$	f_{RT}	\tilde{s}_{MMSE}	$\tilde{s}_{\gamma \rightarrow \infty}$	$\tilde{s}_{\gamma \rightarrow 0}$	f_{RT}	\tilde{s}_{MMSE}	$\tilde{s}_{\gamma \rightarrow \infty}$
D	10^{-3}	1.37×10^{-5}	1.37×10^{-5}	1.37×10^{-5}	1.78	8.40×10^{-6}	8.40×10^{-6}	8.40×10^{-6}	2.94
	10^0	1.17×10^{-2}	1.17×10^{-2}	1.17×10^{-2}	1.66	8.44×10^{-3}	8.44×10^{-3}	8.44×10^{-3}	2.72
	10^3	1.59×10^{-1}	1.56×10^{-1}	1.49×10^{-1}	1.58×10^{-1}	1.05×10^{-1}	1.05×10^{-1}	9.96×10^{-2}	1.21×10^{-1}
	10^6	1.61×10^{-1}	1.60×10^{-1}	1.43×10^{-1}	1.43×10^{-1}	1.07×10^{-1}	1.07×10^{-1}	9.11×10^{-2}	9.11×10^{-2}
	10^9	1.66×10^{-1}	1.66×10^{-1}	1.47×10^{-1}	1.47×10^{-1}	1.10×10^{-1}	1.10×10^{-1}	9.34×10^{-2}	9.34×10^{-2}
D ²	10^{-3}	8.43×10^{-7}	8.43×10^{-7}	8.43×10^{-7}	1.07	3.12×10^{-8}	3.11×10^{-8}	3.11×10^{-8}	1.34
	10^0	9.06×10^{-4}	9.06×10^{-4}	9.05×10^{-4}	1.07	3.34×10^{-5}	3.34×10^{-5}	3.34×10^{-5}	1.33
	10^3	4.04×10^{-1}	4.04×10^{-1}	3.61×10^{-1}	7.1×10^{-1}	1.46×10^{-2}	1.46×10^{-2}	1.46×10^{-2}	5.78×10^{-1}
	10^6	6.53×10^{-1}	6.53×10^{-1}	3.66×10^{-1}	3.66×10^{-1}	2.63×10^{-2}	2.63×10^{-2}	2.26×10^{-2}	2.29×10^{-2}
	10^9	6.62×10^{-1}	6.62×10^{-1}	3.86×10^{-1}	3.86×10^{-1}	2.65×10^{-2}	2.65×10^{-2}	2.16×10^{-2}	2.16×10^{-2}
D ² + 4I	10^{-3}	5.53×10^{-7}	5.53×10^{-7}	5.53×10^{-7}	1.03	1.71×10^{-8}	1.71×10^{-8}	1.71×10^{-8}	1.22
	10^0	5.56×10^{-4}	5.56×10^{-4}	5.55×10^{-4}	1.04	1.77×10^{-5}	1.77×10^{-5}	1.77×10^{-5}	1.24
	10^3	3.67×10^{-1}	3.67×10^{-1}	3.04×10^{-1}	8.79×10^{-1}	1.21×10^{-2}	1.21×10^{-2}	1.20×10^{-2}	8.71×10^{-1}
	10^6	1.52	1.52	4.63×10^{-1}	4.63×10^{-1}	3.94×10^{-2}	3.94×10^{-2}	2.98×10^{-2}	3.04×10^{-2}
	10^9	1.47	1.47	4.87×10^{-1}	4.87×10^{-1}	4.67×10^{-2}	4.67×10^{-2}	3.18×10^{-2}	3.18×10^{-2}

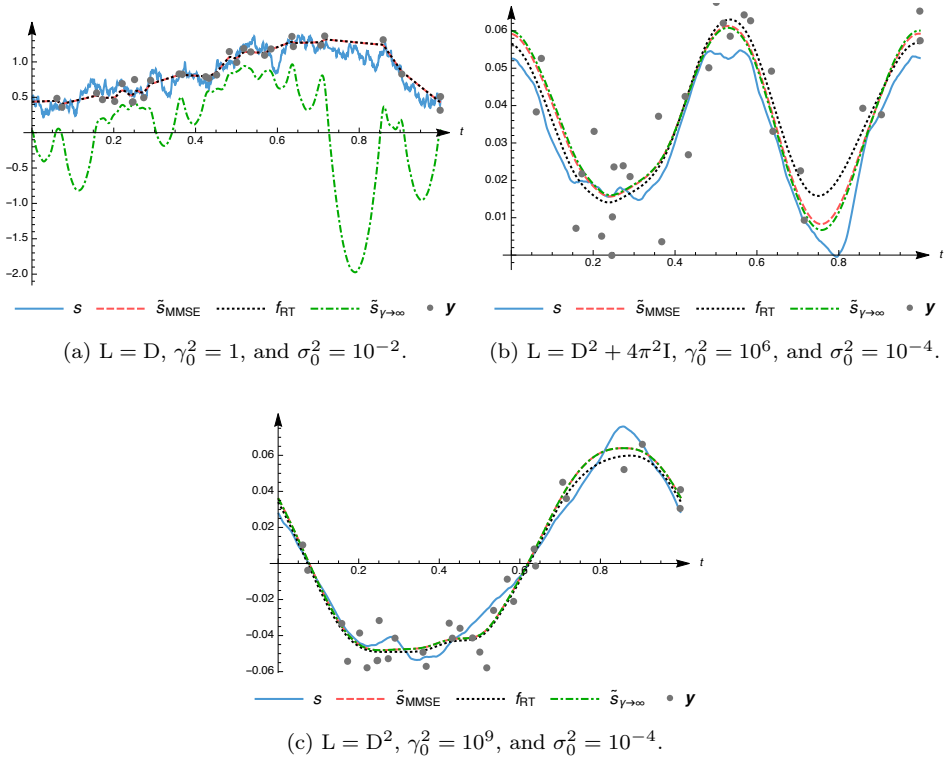


Figure B.4: Illustrations of $s \sim \mathcal{GB}(L, \gamma_0^2)$, \tilde{s}_{MMSE} , f_{RT} , and $\tilde{s}_{\gamma \rightarrow \infty}$ for several operators and values of γ_0^2 and σ_0^2 . We used $M = 30$ noisy measurements $\mathbf{y} = (y_1, \dots, y_M)$.

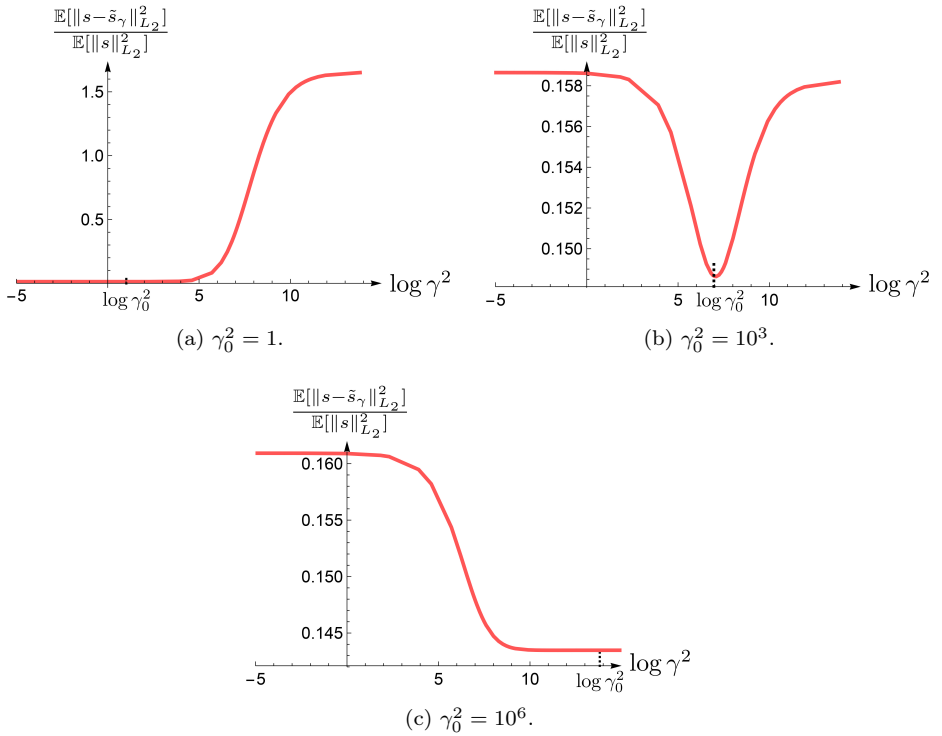


Figure B.5: Evolution of NMSE according to γ for $s \sim \mathcal{GB}(D, \gamma_0^2)$.

B.6 Discussion

B.6.1 Comparison with Inverse Problems on the Real Line

It is worth noting that the periodic setting has important differences as compared to reconstruction methods over the complete real line, which motivated and played an important role in this chapter.

- The role of the Dirac impulse δ is played by the Dirac comb III in the periodic setting. It is indeed the neutral element of the periodic convolution (B.2.3) and appears in the definition of the periodic L-splines (Definition B.2.2) and RKHS (Definition B.3.2).
- In the real-line setting, in addition to smoothness properties, functions are also characterized by their property of decay at infinity [243]. For periodic functions, we only consider the smoothness properties, which brings substantial simplifications.
- In general, a continuous LSI operator does not preserve the asymptotic behavior of the input function. For instance, a test function in the space $\mathcal{S}(\mathbb{R})$ of smooth and rapidly decaying functions is not necessarily mapped to a rapidly decaying function. In contrast, any continuous LSI operator maps the space of periodic test functions $\mathcal{S}(\mathbb{T})$ onto itself (see Section B.2.1). This greatly simplifies the study of operators that act on periodic functions.
- The null space of a continuous LSI operator can differ for the two cases. In particular, when acting on periodic functions, the null space of the n th derivative D^n is reduced to constant functions for every $n \geq 1$. This is crucial due to the role of the null space in Theorems B.3.4 and B.4.4.
- In Proposition B.3.3, we give a necessary and sufficient condition for a continuous LSI operator of finite-dimensional null space to specify a RKHS in the sense of Definition B.3.2. This is significantly more complicated over the real line, for which only partial results are known [222].
- We have seen that it is not always possible to find a periodic solution s to the equation $LS = w$, where w is a periodic Gaussian white noise. This lead us to modify the stochastic differential equation (see (B.4.5)) and to introduce the family of Gaussian bridges.

- In Theorem B.4.4, we give the MMSE estimator of the *complete* process s , not only for the estimation of $s(t_0)$ at a fixed time t_0 . In the non-periodic setting, however, solutions of stochastic differential equations are generally not square-integrable. For instance, if s is a nontrivial stationary Gaussian process, then

$$\begin{aligned} \mathbb{E}[\|s\|_{L_2(\mathbb{R})}^2] &= \sum_{k \in \mathbb{Z}} \mathbb{E}[\|1_{[k, k+1)} \cdot s\|_{L_2(\mathbb{R})}^2] \\ &\stackrel{(i)}{=} \sum_{k \in \mathbb{Z}} \mathbb{E}[\|1_{[0, 1)} \cdot s\|_{L_2(\mathbb{R})}^2] = \infty, \end{aligned} \quad (\text{B.6.1})$$

where $1_{[a, b)}$ is the indicator function on $[a, b)$ and (i) exploits stationarity. Another example is the Brownian motion, whose supremum over $[0, t]$ grows faster than t^p for any $p < 1/2$ (almost surely) when t goes to infinity [244], hence being of infinite energy. As a consequence, it is irrelevant to consider the MMSE estimator of the complete process and one ought to, for instance, restrict to MMSE estimators of local values $s(t_0)$ of the process.

B.6.2 Comparison with TV Regularization

A tendency in the field of signal reconstruction is to rely on sparsity-promoting regularization, motivated by the fact that many real-world signals are sparse in some adequate transform domain [230, 245, 246].

The vast majority of works focuses on the finite-dimensional setting via ℓ_1 -type regularization. However, some authors have recently promoted the reconstruction of infinite-dimensional sparse signals [203, 247]. The adaptation of discrete ℓ_1 methods to the continuous domain is based on the total-variation (TV) regularization norm, for which it is possible to derive representer theorems (see [222, Theorem 1]). A comparison between Tikhonov and TV variational techniques is proposed by Gupta *et al.* [223] for non-periodic signals. In brief, at identical measurements and regularization operator L , Tikhonov regularization favors smooth solutions restricted to a finite-dimensional space, while TV regularization allows for adaptive and more compressible solutions. In [223, Table I], it was shown on simulations that Tikhonov methods perform better on fractal-type signals, while TV methods are better suited to sparse signals. We expect similar behaviors for the periodic setting.

At the heart of the present chapter is the connection between L_2 -regularization and the statistical formalism of MMSE estimation of Gaussian processes. A theoretical link between deterministic and stochastic frameworks is much harder to provide for sparsity-inducing priors. There is strong empirical evidence that sparse stochastic models are intimately linked to TV-based methods [230], but the extent to which such estimators approach the MMSE solution is still unknown.

B.7 Conclusion

We have presented two approaches for the reconstruction of periodic continuous-domain signals from their corrupted discrete measurements. The first approach is based on optimization theory and culminates with the specification of a periodic representer theorem (Theorem B.3.4). In the second approach, a signal is modeled as a stationary periodic random process and the reconstruction problem is transformed into an estimation problem. Theorem B.4.4 then gives the optimal estimator (in the mean-square sense) for Gaussian bridges.

We have also provided theoretical and experimental comparisons of the two approaches and identified two main findings. First, for invertible operators, the statistical and variational approaches are equivalent and correspond to an identical reconstruction scheme. For noninvertible operators, however, this equivalence is not valid anymore, but the variational method corresponds to the statistical reconstruction when the parameter γ vanishes. More importantly, for small values of γ_0^2 , the variational method is practically equivalent to the optimal statistical reconstruction. This demonstrates the efficiency of the representer theorem for reconstructing Gaussian bridges, even for noninvertible operators.

B.8 Appendices

B.8.1 Proof of Proposition B.2.1

The main argument is very classical in the non-periodic setting. We detail it for the sake of completeness and adapt it to the periodic case.

Let p be a function of \mathcal{N}_L . As L is shift-invariant, $p(\cdot - t_0) \in \mathcal{N}_L$ for every $t_0 \in \mathbb{T}$. Moreover, \mathcal{N}_L is closed in $\mathcal{S}'(\mathbb{T})$ (as any finite-dimensional linear subspace), thus the first derivative $p' = p^{(1)}$ of p is in \mathcal{N}_L as the limit of the function $\frac{1}{t_0}(p(\cdot - t_0) - p) \in \mathcal{N}_L$ when $t_0 \rightarrow 0$. We propagate this property to all the derivatives of p .

We now have that \mathcal{N}_L is a finite-dimensional space of dimension N_0 and $p^{(k)} \in \mathcal{N}_L, \forall k \in [1 \dots N_0]$. Hence, the family of $(N_0 + 1)$ functions $p, p^{(1)}, \dots, p^{(N_0)}$ satisfies an equation of the form $a_{N_0}p^{(N_0)} + \dots + a_0p = 0$, where $a_k \in \mathbb{C}$ and $(a_0, \dots, a_{N_0}) \neq \mathbf{0}$. This implies that p , as solution of a differential equation with constant coefficients, is a sum of functions of the form $q(t)e^{\mu t}$ with q a polynomial and $\mu \in \mathbb{C}$.

Finally, since we deal with 1-periodic functions, this constrains q to be a constant function and $\mu = 2\pi jk$ with $k \in \mathbb{Z}$. This concludes the proof. ■

B.8.2 Proof of Proposition B.3.1

The linearity, Hermitian symmetry, and non-negativity are easily obtained. We only need to verify that $\|f\|_{\mathcal{H}_L} = \langle f, f \rangle_{\mathcal{H}_L}^{\frac{1}{2}} = 0 \Leftrightarrow f = 0$. For this, we observe that

$$\begin{aligned} \langle f, f \rangle_{\mathcal{H}_L} = 0 &\Leftrightarrow \int_0^1 |Lf(t)|^2 dt + \gamma^2 \sum_{n=1}^{N_0} |\widehat{f}[k_n]|^2 = 0 \\ &\Leftrightarrow \sum_{k \in \mathcal{K}_L} |\widehat{f}[k]|^2 \underbrace{|\widehat{L}[k]|^2}_{\neq 0} + \gamma^2 \sum_{n=1}^{N_0} |\widehat{f}[k_n]|^2 = 0, \end{aligned} \tag{B.8.1}$$

which implies that $\widehat{f}[k] = 0$ for all $k \in \mathbb{Z}$. Hence, $\langle f, f \rangle_{\mathcal{H}_L} = 0 \Leftrightarrow f = 0$. ■

B.8.3 Proof of Proposition B.3.3

For the proof, we set $A = \sum_{k \in \mathcal{K}_L} \frac{1}{|\widehat{L}[k]|^2}$. The Hilbert space \mathcal{H}_L is a RKHS if and only if $\mathbb{III} \in \mathcal{H}'_L$ or, equivalently, if there exists $C > 0$ such that

$$\forall f \in \mathcal{S}(\mathbb{T}), \quad |\langle \mathbb{III}, f \rangle| \leq C \|f\|_{\mathcal{H}_L}. \quad (\text{B.8.2})$$

Assume that $A < +\infty$. Let c be the sequence such that $c[k] = 1/\widehat{L}[k]$ if $k \in \mathcal{K}_L$ and $c[k] = 1/\gamma$ otherwise. Using the Cauchy-Schwarz inequality, we have, for every $f \in \mathcal{S}(\mathbb{T})$, that

$$\begin{aligned} \langle \mathbb{III}, f \rangle^2 &= \left(\sum \widehat{f}[k] \right)^2 \leq \left(\sum |c[k]|^2 \right) \left(\sum \left| \frac{\widehat{f}[k]}{c[k]} \right|^2 \right) \\ &= (N_0/\gamma^2 + A) \|f\|_{\mathcal{H}_L}^2. \end{aligned} \quad (\text{B.8.3})$$

Hence, (B.8.2) is satisfied for $C = (N_0/\gamma^2 + A)^{1/2} > 0$. For the converse, we define $f_m \in \mathcal{S}(\mathbb{T})$ such that

$$\widehat{f}_m[k] = \begin{cases} 0, & \text{if } |k| > m \text{ or } k = k_n, n \in [1 \dots N_0] \\ \frac{1}{|\widehat{L}[k]|^2}, & \text{otherwise.} \end{cases}$$

Then, we readily observe that $\lim_{m \rightarrow +\infty} \frac{|\langle \mathbb{III}, f_m \rangle|}{\|f_m\|_{\mathcal{H}_L}} = \sqrt{A}$. Therefore, as soon as $A = +\infty$, $\langle \mathbb{III}, f \rangle / \|f\|_{\mathcal{H}_L}$ is not bounded in $\mathcal{S}(\mathbb{T})$ and \mathcal{H}_L is not a RKHS.

The reproducing kernel is characterized by the relation $f(\tau) = \langle h(\cdot, \tau), f \rangle_{\mathcal{H}_L}$ for every $f \in \mathcal{H}_L$. Let \mathbf{R} be the operator, often called the Riesz map, such that $\langle \mathbf{R}g, f \rangle_{\mathcal{H}_L} = \langle g, f \rangle$ for any $f \in \mathcal{H}_L$ and $g \in \mathcal{H}'_L$. Then, $h(\cdot, \tau) = \mathbf{R}\{\mathbb{III}(\cdot - \tau)\}$. Moreover, we have that $\langle \mathbf{R}e_k, e_m \rangle_{\mathcal{H}_L} = \delta[k - m]$. In addition,

$$\begin{aligned} \langle \mathbf{R}e_k, e_m \rangle_{\mathcal{H}_L} &= \langle \mathbf{L}\mathbf{R}e_k, \mathbf{L}e_m \rangle + \gamma^2 \sum_{n=1}^{N_0} \widehat{\mathbf{R}e_k}[k_n] \overline{\widehat{e_m}[k_n]} \\ &= \langle \mathbf{R}e_k, \mathbf{L}^* \mathbf{L}e_m \rangle + \gamma^2 \sum_{n=1}^{N_0} \widehat{\mathbf{R}e_k}[k_n] \delta[m - k_n] \\ &= |\widehat{L}[m]|^2 \widehat{\mathbf{R}e_k}[m] + \gamma^2 \sum_{n=1}^{N_0} \widehat{\mathbf{R}e_k}[k_n] \delta[m - k_n]. \end{aligned} \quad (\text{B.8.4})$$

Hence, R is characterized for $k, m \in \mathbb{Z}$ by the relation

$$|\widehat{L}[m]|^2 \widehat{Re}_k[m] + \gamma^2 \sum_{n=1}^{N_0} \widehat{Re}_k[k_n] \delta[m - k_n] = \delta[k - m]. \quad (\text{B.8.5})$$

For $k \in \mathcal{K}_L$, we deduce from (B.8.5) that $\widehat{Re}_k[m] = 1/|\widehat{L}[k]|^2$ if $m = k$ and 0 otherwise. We also deduce that, for $k = k_n$, $\widehat{Re}_{k_n}[m] = 1/\gamma^2$ if $m = k_n$ and 0 otherwise. Thus, R is shift-invariant ($\widehat{Re}_k[m] = 0$ for every $m \neq k$), meaning that $h(t, \tau)$ depends only on $(t - \tau)$. Moreover, the Fourier multiplier of R , which is also the discrete Fourier transform of $h_\gamma(t) = h(t, 0)$, is $\widehat{R}[k] = 1/|\widehat{L}[k]|^2$ if $k \in \mathcal{K}_L$ and $1/\gamma^2$ if $k = k_n$. This is equivalent to (B.3.6) and concludes the proof. \blacksquare

B.8.4 Proof of Theorem B.3.4

To prove Theorem B.3.4, we first show that the optimization problem (B.3.7) has a unique solution by convex-optimization arguments. Then, we connect this solution to the abstract representer theorem (see for instance [248, Theorem 16.1]) to deduce the form of the solution. We start with some preliminary results for the first part.

Lemma B.8.1. *Under the condition of Theorem B.3.4, the functional $\phi : \mathcal{H}_L \rightarrow \mathbb{R}^+$ defined by $\phi(f) = F(\mathbf{y}, \langle \boldsymbol{\nu}, f \rangle) + \lambda \|L f\|_{L_2}^2$ is strictly convex and coercive, meaning that $\phi(f) \rightarrow \infty$ when $\|f\|_{\mathcal{H}_L} \rightarrow \infty$.*

Proof. Strict convexity: ϕ is convex as a sum of two convex functions. For the strict convexity, we fix $\mu \in (0, 1)$ and $f, g \in \mathcal{H}_L$. It is then sufficient to show that the equality

$$\phi(\mu f + (1 - \mu)g) = \mu\phi(f) + (1 - \mu)\phi(g) \quad (\text{B.8.6})$$

implies that $f = g$. The functions $F(\mathbf{y}, \boldsymbol{\nu}\{\cdot\})$ and $\|L \cdot\|_{L_2}$ are convex, therefore (B.8.6) together with the linearity of both $\boldsymbol{\nu}$ and L implies the two relations

$$\begin{aligned} F(\mathbf{y}, \mu \boldsymbol{\nu}(f) + (1 - \mu)\boldsymbol{\nu}(g)) &= \mu F(\mathbf{y}, \boldsymbol{\nu}(f)) + (1 - \mu)F(\mathbf{y}, \boldsymbol{\nu}(g)) \\ \|\mu L f + (1 - \mu)L g\|_{L_2}^2 &= \mu \|L f\|_{L_2}^2 + (1 - \mu)\|L g\|_{L_2}^2. \end{aligned} \quad (\text{B.8.7})$$

Now, taking advantage of the strict convexity of $F(\mathbf{y}, \cdot)$ and $\|\cdot\|_{L_2}^2$, we deduce that $\boldsymbol{\nu}(f) = \boldsymbol{\nu}(g)$ and $L f = L g$. This means, in particular, that $(f - g)$ is in the

intersection of the null spaces of ν and L , assumed to be trivial. Finally, $f = g$ as expected.

Coercivity: The measurement functional ν is linear and continuous, hence there exists $A > 0$ such that $|\langle \nu, f \rangle|^2 \leq A \|f\|_{\mathcal{H}_L}^2$ for any $f \in \mathcal{H}_L$. Moreover, since ν is injective and linear when restricted to the finite-dimensional null space \mathcal{N}_L , there exists $B > 0$ such that $\|\langle \nu, p \rangle\|^2 \geq B \|p\|_{\mathcal{H}_L}^2$ for any $p \in \mathcal{N}_L$. Any $f \in \mathcal{H}_L$ can be decomposed uniquely as

$$f = \sum_{k \in \mathcal{K}_L} \widehat{f}[k] e_k + \sum_{n=1}^{N_0} \widehat{f}[k_n] e_{k_n} = g + p. \tag{B.8.8}$$

In that case, we easily see that $\|g\|_{\mathcal{H}_L} = \|Lf\|_{L_2}$. In particular, we deduce that

$$\begin{aligned} \|f\|_{\mathcal{H}_L}^2 &= \|g\|_{\mathcal{H}_L}^2 + \|p\|_{\mathcal{H}_L}^2 \leq \|Lf\|_{L_2}^2 + \frac{1}{B} \|\langle \nu, p \rangle\|^2 \\ &\leq \|Lf\|_{L_2}^2 + \frac{1}{B} (\|\langle \nu, f \rangle\| + \|\langle \nu, g \rangle\|)^2 \\ &\leq \|Lf\|_{L_2}^2 + \frac{1}{B} \left(\|\langle \nu, f \rangle\| + A^{1/2} \|Lf\|_{L_2} \right)^2 \\ &\leq C (\|Lf\|_{L_2}^2 + \|\langle \nu, f \rangle\|^2) \end{aligned} \tag{B.8.9}$$

for $C > 0$ large enough. Now, consider a sequence of functions $f_m \in \mathcal{H}_L$ such that $\|f\|_{\mathcal{H}_L} \rightarrow \infty$. We want to show that, for m large enough, $\phi(f_m)$ is arbitrarily large. Due to (B.8.9), for m large enough, $\|Lf_m\|_{L_2}$ or $\|\langle \nu, f_m \rangle\|$ are arbitrarily large. The former implies obviously that $\phi(f_m)$ can be made as large as we want. It is also true for the latter because $\phi(f_m) \geq F(\mathbf{y}, \langle \nu, f_m \rangle)$ and F is coercive. This means that $\phi(f_m)$ goes to infinity when $m \rightarrow \infty$, hence ϕ is coercive. \square

As ϕ is a strictly convex and coercive functional (Lemma B.8.1), the optimization problem (B.3.7) has the unique solution f_{RT} . We denote $z_0 = \langle \nu, f_{RT} \rangle$. The function f_{RT} can be uniquely decomposed as

$$f_{RT} = \sum_{k \in \mathcal{K}_L} \widehat{f}_{RT}[k] e_k + \sum_{n=1}^{N_0} \widehat{f}_{RT}[k_n] e_{k_n} = g_{RT} + p_{RT}. \tag{B.8.10}$$

We recall the abstract representer theorem. This result can be found in [223, Theorem 8] with a formulation close to ours.

Proposition B.8.2. *Let \mathcal{H} be a Hilbert space, $\boldsymbol{\nu} = (\nu_1, \dots, \nu_M)$ be a vector of M linear and continuous measurement functionals over \mathcal{H} , and $\mathbf{y}_0 \in \mathbb{R}^M$. There exists a unique minimizer of the optimization problem*

$$\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}} \text{ s.t. } \boldsymbol{\nu} = \mathbf{y}_0, \quad (\text{B.8.11})$$

which is of the form $f_{\text{opt}} = \sum_{m=1}^M a_m R\nu_m$, where $a_m \in \mathbb{R}$ and $R : \mathcal{H}' \rightarrow \mathcal{H}$ is the Riesz map of \mathcal{H} .

We consider the Hilbert space $\tilde{\mathcal{H}}_L = \{f \in \mathcal{H}_L, \text{Proj}_{\mathcal{N}_L}\{f\} = 0\}$, on which $\|Lf\|_{L_2}$ is a Hilbertian norm. The linear measurements ν_m are in the dual space $\tilde{\mathcal{H}}'_L$, once restricted as linear functionals on $\tilde{\mathcal{H}}_L$. The interpolation constraint is chosen as $\mathbf{y}_0 = \mathbf{z}_0 - \boldsymbol{\nu}(p_{\text{RT}})$. Applying Proposition B.8.2 to this case, we deduce that there exists a unique minimizer

$$h_{\text{opt}} = \arg \min_{h \in \tilde{\mathcal{H}}_L, \boldsymbol{\nu}(h) = \mathbf{y}_0} \|Lh\|_{L_2} \quad (\text{B.8.12})$$

which is of the form $h_{\text{opt}} = \sum_{m=1}^M a_m R\nu_m$, R being the Riesz map between $\tilde{\mathcal{H}}'_L$ and $\tilde{\mathcal{H}}_L$. In our case, the function $R\nu_m$ is given by $R\nu_m = \sum_{k \in \mathcal{K}_L} \frac{\tilde{\nu}_m[k]}{|\tilde{L}[k]|^2} e_k$. In particular, one easily sees from the expression of φ_m that it satisfies

$$R\nu_m = \varphi_m - \gamma^2 \text{Proj}_{\mathcal{N}_L}\{\nu_m\}. \quad (\text{B.8.13})$$

Moreover, we have that $h_{\text{opt}} = g_{\text{RT}}$. Indeed, g_{RT} is clearly among the functions h over which one minimizes and one cannot have that $\|Lh_{\text{opt}}\|_{L_2} < \|Lg_{\text{RT}}\|_{L_2}$ (otherwise, the function $f = h_{\text{opt}} + p_{\text{RT}}$ would be a minimizer of (B.3.7) different from f_{RT} , which is impossible). Putting things together, we get that

$$\begin{aligned} f_{\text{RT}} &= g_{\text{RT}} + p_{\text{RT}} = \sum_{m=1}^M a_m R\nu_m + p_{\text{RT}} \\ &= \sum_{m=1}^M a_m \varphi_m - \gamma^2 \sum_{m=1}^M a_m \text{Proj}_{\mathcal{N}_L}\{\nu_m\} + p_{\text{RT}}. \end{aligned} \quad (\text{B.8.14})$$

Since $(-\gamma^2 \sum_{m=1}^M a_m \text{Proj}_{\mathcal{N}_L} \{\nu_m\} + p_{\text{RT}})$ is in the null space of L , it can be developed as $\sum_{n=1}^{N_0} b_n e_{k_n}$, giving (B.3.8).

The last ingredient is to remark that a_m satisfies $\mathbf{P}^\top \mathbf{a} = \mathbf{0}$. This comes from the fact that, by construction, $\sum a_m R\nu_m \in \tilde{\mathcal{H}}_L'$ and, by applying the Riesz map, $\sum a_m \nu_m \in \tilde{\mathcal{H}}_L$, meaning that the projection of this element into the null space is zero. This is precisely equivalent with the expected condition. ■

B.8.5 Proof of Proposition B.3.5

We compute (B.3.7) for F the quadratic cost function. We have that $f_{\text{RT}} = \sum_{m=1}^M a_m \varphi_m + \sum_{n=1}^{N_0} b_n e_{k_n}$, as given by (B.3.8). It then suffices to find the optimal vectors \mathbf{a} and \mathbf{b} . We therefore rewrite (B.3.7) in terms of these two vectors.

From simple computations, we have, with the notations of Proposition B.3.5, that $\langle \boldsymbol{\nu}, \sum_{n=1}^{N_0} b_n e_{k_n} \rangle = \mathbf{P}\mathbf{b}$ and $\langle \boldsymbol{\nu}, \sum_{m=1}^M a_m \varphi_m \rangle = \mathbf{G}\mathbf{a}$, where we used for the latter that $\mathbf{G}_{m_1, m_2} = \langle \nu_{m_1}, h_\gamma * \nu_{m_2} \rangle = \langle \nu_{m_1}, \varphi_{m_2} \rangle$. Hence,

$$\|\mathbf{y} - \langle \boldsymbol{\nu}, f \rangle\|^2 = \|\mathbf{y} - \mathbf{G}\mathbf{a} - \mathbf{P}\mathbf{b}\|^2. \quad (\text{B.8.15})$$

From the definition of h_γ in (B.3.6), we see that $(L^* L h_\gamma) * f = f$ for every f whose Fourier coefficients $\widehat{f}[k_n]$ do vanish for every $n = 1 \dots N_0$. Now, the relation $\overline{\mathbf{P}}^\top \mathbf{a} = \mathbf{0}$ in Theorem B.3.4 shows precisely that $\sum_{n=1}^M a_n \nu_n$ satisfies this property. In particular, we deduce that

$$L^* L \left\{ \sum_{m=1}^M a_m \varphi_m \right\} = (L^* L h_\gamma) * \sum_{m=1}^M a_m \nu_m = \sum_{m=1}^M a_m \nu_m. \quad (\text{B.8.16})$$

As a consequence, we have that

$$\begin{aligned} \|L f_{\text{RT}}\|_{L_2}^2 &= \langle L^* L \sum_{m_1=1}^M a_{m_1} \varphi_{m_1}, \sum_{m_2=1}^M a_{m_2} \varphi_{m_2} \rangle \\ &= \sum_{m_1=1}^M \sum_{m_2=1}^M a_{m_1} \mathbf{G}_{m_1, m_2} a_{m_2} = (\mathbf{G}\mathbf{a})^\top \mathbf{a}. \end{aligned} \quad (\text{B.8.17})$$

Finally, one has that

$$\|\mathbf{y} - \langle \boldsymbol{\nu}, f_{\text{RT}} \rangle\|^2 + \lambda \|L f_{\text{RT}}\|_{L_2}^2 = \|\mathbf{y} - \mathbf{Ga} - \mathbf{Pb}\|^2 + \lambda (\mathbf{Ga})^\top \mathbf{a}. \quad (\text{B.8.18})$$

By computing the partial derivatives, we find that the vectors \mathbf{a} and \mathbf{b} are given by (B.3.9). ■

B.8.6 Proof of Proposition B.3.6

Since $\nu_m = \text{III}(\cdot - t_m)$, the form of the solution (B.3.8) is $f_{\text{RT}}(t) = \sum_{m=1}^M a_m h_\gamma(t - t_m) + \sum_{n=1}^{N_0} b_n e_{k_n}(t)$. We have moreover that $\mathbf{P}^\top \mathbf{a} = \mathbf{0}$, where $[\mathbf{P}]_{m,n} = e^{j2\pi k_n t_m}$. From (B.3.6), we then deduce that $L^*L\{h_\gamma\}(t) = \sum_{k \in \mathcal{K}_L} |\widehat{L}[k]|^2 \frac{e_{k}(t)}{|\widehat{L}[k]|^2} = \left(\text{III}(t) - \text{Proj}_{\mathcal{N}_L} \{ \text{III} \}(t) \right)$. By linearity, we get that

$$\begin{aligned} L^*L\{f_{\text{RT}}\}(t) &= \sum_{m=1}^M a_m L^*L\{h_\gamma\}(t - t_m) \\ &= \sum_{m=1}^M a_m \text{III}(t - t_m) - \sum_{m=1}^M a_m \text{Proj}_{\mathcal{N}_L} \{ \text{III}(\cdot - t_m) \}(t) \\ &= \sum_{m=1}^M a_m \text{III}(t - t_m) - \sum_{n=1}^{N_0} \sum_{m=1}^M a_m e^{-j2\pi k_n t_m} e_{k_n} \\ &= \sum_{m=1}^M a_m \text{III}(t - t_m) - \sum_{n=1}^{N_0} [\overline{\mathbf{P}}^\top \mathbf{a}]_n e_{k_n} \end{aligned} \quad (\text{B.8.19})$$

$$= \sum_{m=1}^M a_m \text{III}(t - t_m), \quad (\text{B.8.20})$$

where we used that $[\overline{\mathbf{P}}]_{m,n} = e^{-j2\pi k_n t_m}$ in (B.8.19) and that $\overline{\mathbf{P}}^\top \mathbf{a} = \overline{\mathbf{P}^\top \mathbf{a}} = \mathbf{0}$ in (B.8.20). Finally, f_{RT} is a periodic (L^*L) -spline with weights a_m and knots t_m . ■

B.8.7 Proof of Proposition B.4.3

We start from

$$s = \sum_{k \in \mathcal{K}_L} \frac{\widehat{w}[k]}{\widehat{L}[k]} e_k + \sum_{n=1}^{N_0} \frac{\widehat{w}[k_n]}{\gamma_0} e_{k_n}. \quad (\text{B.8.21})$$

Our goal is to compute $r_s(t, \tau) = \mathbb{E}[s(t)s(\tau)]$. We do so by replacing $s(t)$ and $s(\tau)$ with (B.8.21). We develop the product and use the relations $\mathbb{E}[\widehat{w}[k]\widehat{w}[\ell]] = \mathbb{E}[\widehat{w}[k]^2] = 0$, $\mathbb{E}[\widehat{w}[k]^2] = 1$ for every $k, \ell \in \mathbb{Z}$, $k \neq \ell$ to deduce that

$$r_s(t, \tau) = \left(\sum_{k \in \mathcal{K}_L} \frac{e_k(t)e_{-k}(\tau)}{|\widehat{L}[k]|^2} + \frac{1}{\gamma_0^2} \sum_{n=1}^{N_0} e_{k_n}(t)e_{-k_n}(\tau) \right). \quad (\text{B.8.22})$$

Since $e_k(t)e_{-k}(\tau) = e_k(t - \tau)$, we have shown that $r_s(t, \tau) = h_\gamma(t - \tau)$, as expected. Then, we obtain (B.4.7) by injecting (B.4.6) into (B.4.2). Finally, we obtain (B.4.8) by particularizing (B.4.7) with $\nu_m = e_k$. ■

B.8.8 Proof of Theorem B.4.4

We fix a time $t_0 \in \mathbb{T}$. We first obtain the MMSE estimator for $s(t_0)$ (estimation of s at time t_0). (Note that $s(t_0) = \langle s, \text{III}(\cdot - t_0) \rangle$ is well defined because $\text{III}(\cdot - t_0) \in \mathcal{H}_L$ by assumption).

The linear MMSE estimator of $s(t_0)$ based on \mathbf{y} is of the form $\tilde{s}_{t_0} = \sum_{m=1}^M u_m y_m$. Because s and ϵ are Gaussian, the linear MMSE estimator coincides with the MMSE estimator [86]. The orthogonality principle [Section 3.2] [86] then implies that

$$\mathbb{E}[y_m(s(t_0) - \tilde{s}_{t_0})] = 0, \quad \forall m = 1 \dots M. \quad (\text{B.8.23})$$

We know from Proposition B.4.3 that $\mathbb{E}[\langle s, f \rangle \langle s, g \rangle] = \langle h_{\gamma_0} * f, g \rangle$. We use this relation to develop the different terms of (B.8.23). First, we have that

$$\begin{aligned} \mathbb{E}[y_m s(t_0)] &= \mathbb{E}[\langle \nu_m, s \rangle s(t_0)] + \mathbb{E}[\epsilon_m s(t_0)] \\ &= \mathbb{E}[\langle \nu_m, s \rangle \langle s, \text{III}(\cdot - t_0) \rangle] + \underbrace{\mathbb{E}[\epsilon_m]}_0 \mathbb{E}[s(t_0)] \\ &= (h_{\gamma_0} * \nu_m)(t_0). \end{aligned} \quad (\text{B.8.24})$$

As the estimator is of the form $\tilde{s}_{t_0} = \sum_{m=1}^M u_m y_m$ and exploiting that ϵ and s are independent, we have that

$$\begin{aligned}\mathbb{E}[\langle \nu_m, s \rangle y_k] &= \mathbb{E}[\langle \nu_m, s \rangle \langle \nu_k, s \rangle] + \mathbb{E}[\langle \nu_m, s \rangle \epsilon_k] = \langle h_{\gamma_0} * \nu_m, \nu_k \rangle \\ \mathbb{E}[\epsilon_m y_k] &= \mathbb{E}[\epsilon_m \langle \nu_k, s \rangle] + \mathbb{E}[\epsilon_m \epsilon_k] = \sigma^2 \delta[m - k]\end{aligned}\tag{B.8.25}$$

We have therefore that

$$\begin{aligned}\mathbb{E}[y_m \tilde{s}_{t_0}] &= \mathbb{E}[\langle \nu_m, s \rangle \tilde{s}_{t_0}] + \mathbb{E}[\epsilon_m \tilde{s}_{t_0}] \\ &= \sum_{k=1}^M u_k \mathbb{E}[\langle \nu_m, s \rangle y_k] + \sum_{k=1}^M u_k \mathbb{E}[\epsilon_m y_k] \\ &= \sum_{k=1}^M u_k \langle h_{\gamma_0} * \nu_m, \nu_k \rangle + u_m \sigma_0^2.\end{aligned}\tag{B.8.26}$$

We remark that $\langle h_{\gamma_0} * \nu_m, \nu_k \rangle = [\mathbf{G}]_{m_1, m_2}$ given in (B.3.10). Injecting (B.8.24) and (B.8.26) into (B.8.23), we have for $m = 1 \dots M$ that $(h_{\gamma_0} * \nu_m)(t_0) = \sum_{k=1}^M u_k [\mathbf{G}]_{m_1, m_2} + u_m \sigma_0^2$. Hence, $\mathbf{u} = (\mathbf{G} + \sigma_0^2 \mathbf{I})^{-1} \mathbf{c}$, where $\mathbf{c} = (h_{\gamma_0} * \nu)(t_0)$. As $\tilde{s}_{t_0} = \mathbf{u}^T \mathbf{y}$, we finally have that $\tilde{s}_{t_0} = \sum_{m=1}^M d_m (h_{\gamma_0} * \nu_m)(t_0)$, where $\mathbf{d} = (d_1, \dots, d_M) = (\mathbf{G} + \sigma_0^2 \mathbf{I})^{-1} \mathbf{y}$.

We have now obtained the form of the MMSE estimator \tilde{s}_{t_0} for $s(t_0)$ at a fixed time t_0 . We then deduce the MMSE estimator of the complete continuous random process $s : \mathbb{T} \rightarrow \mathbb{R}$ that minimizes $\mathbb{E}[\|s - \tilde{s}\|_{L_2}^2]$ among all the estimators \tilde{s} based on \mathbf{y} . We fix an estimator \tilde{s} . We have that

$$\begin{aligned}\mathbb{E}[\|s - \tilde{s}\|_{L_2}^2] &= \mathbb{E}\left[\int_0^1 (s(t) - \tilde{s}(t))^2 dt\right] = \int_0^1 \mathbb{E}[(s(t) - \tilde{s}(t))^2] dt \\ &\geq \int_0^1 \mathbb{E}[(s(t) - \tilde{s}_t)^2] dt = \mathbb{E}[\|s - \tilde{s}_{\text{MMSE}}\|_{L_2}^2].\end{aligned}\tag{B.8.27}$$

Hence, the function $\tilde{s}_{\text{MMSE}} : t \rightarrow \tilde{s}_t$ is the MMSE estimator of the complete process $s(t)$. ■

B.8.9 Proof of Proposition B.4.5

The proof is obtained by following the arguments of Theorem B.3.4 (for existence, unicity, and form of the solution) and Proposition B.3.5 (for the explicit formula of the coefficients d_m in (B.4.13)) with the following simplifications:

First, the existence and unicity of a solution is now direct. Indeed, the functional to minimize is $\|\mathbf{y} - \nu(f)\|_2^2 + \lambda \|f\|_{\mathcal{H}_L}^2$. It is clearly coercive and strictly convex because $\|\cdot\|_{\mathcal{H}_L}$ is. Second, the abstract representer theorem can now be applied directly to the Hilbert space \mathcal{H}_L . The form of the solution is then directly deduced. Third, the coefficients d_m are found with the arguments of Appendix B.8.5, except that there is no term for the null-space component (coefficients b_n) in that case, hence the system matrix is simpler. ■

B.8.10 Proof of Proposition B.5.1

We know the expression of \tilde{s}_λ from Proposition B.4.5. For Fourier sampling, the φ_m are complex exponential themselves, given by $\varphi_m = h * e_{k_m} = \widehat{h}[k_m]e_{k_m}$, while the Gram matrix \mathbf{G} is diagonal since $\mathbf{G}_{m_1, m_2} = \langle h * e_{k_{m_1}}, e_{k_{m_2}} \rangle = \widehat{h}[k_{m_1}]\delta[k_{m_1} - k_{m_2}]$. Hence, (B.4.13) gives that

$$\tilde{s}_\lambda = \sum_{m=1}^M \frac{(\widehat{s}[k_m] + \epsilon_m)\widehat{h}[k_m]}{\widehat{h}[k_m] + \lambda} e_{k_m}. \quad (\text{B.8.28})$$

After simplification, we have that

$$s - \tilde{s}_\lambda = \sum_{m=1}^M \left(\frac{\lambda \widehat{s}[k_m]}{\widehat{h}[k_m] + \lambda} - \frac{\widehat{h}[k_m]\epsilon_m}{\widehat{h}[k_m] + \lambda} \right) e_{k_m} + \sum_{k \notin \mathcal{N}_\nu} \widehat{s}[k]e_k. \quad (\text{B.8.29})$$

Exploiting the Fourier-domain independence, we deduce that

$$\begin{aligned} \mathbb{E} [\|s - \tilde{s}_\lambda\|_{L_2}^2] &= \sum_{m=1}^M \frac{\lambda^2}{(\widehat{h}[k_m] + \lambda)^2} \mathbb{E} [|\widehat{s}[k_m]|^2] + \frac{\widehat{h}[k_m]^2}{(\widehat{h}[k_m] + \lambda)^2} \mathbb{E} [|\epsilon_m|^2] \\ &\quad + \sum_{k \notin \mathcal{N}_\nu} \mathbb{E} [|\widehat{s}[k]|^2]. \end{aligned} \quad (\text{B.8.30})$$

From the relations $\mathbb{E} [|\widehat{s}[k]|^2] = \widehat{h}[k]$ (see (B.4.8)) and $\mathbb{E} [|\epsilon_m|^2] = \sigma_0^2$, we finally obtain (B.5.3). ■

Bibliography

- [1] S. A. da Silveira, B. L. Schneider, C. Cifuentes-Diaz, D. Sage, T. Abbas-Terki, T. Iwatsubo, M. Unser, and P. Aebischer, “Phosphorylation does not prompt, nor prevent, the formation of α -synuclein toxic species in a rat model of Parkinson’s disease,” *Human Molecular Genetics*, vol. 18, no. 5, pp. 872–887, March 2009.
- [2] Miguel A Luengo-Oroz, M J Ledesma-Carbayo, Nadine Peyri eras, and Andr es Santos, “Image analysis for understanding embryo development: a bridge from microscopy to biological insights,” *Current Opinion in Genetics & Development*, vol. 21, no. 5, pp. 630–637, October 2011.
- [3] Ann K. Corsi, Bruce Wightman, and Martin Chalfie, “A transparent window into biology: A primer on *Caenorhabditis elegans*,” *Genetics*, vol. 200, no. 2, pp. 387–407, June 2015.
- [4] Ian B Malone, Kelvin K Leung, Shona Clegg, Josephine Barnes, Jennifer L Whitwell, John Ashburner, Nick C Fox, and Gerard R Ridgway, “Accurate automatic estimation of total intracranial volume: A nuisance variable with less nuisance,” *NeuroImage*, vol. 104, no. 1, pp. 366–372, January 2015.
- [5] Hanchuan Peng, “Bioimage informatics: a new area of engineering biology,” *Bioinformatics*, vol. 24, no. 17, pp. 1827–1836, September 2008.
- [6] Gene Myers, “Why bioimage informatics matters,” *Nature Methods*, vol. 9, no. 7, pp. 659–660, June 2012.

- [7] Alexandre C. Dufour, Anneliene H. Jonker, and Jean-Christophe Olivo-Marin, “Deciphering tissue morphodynamics using bioimage informatics,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 372, no. 1720, May 2017.
- [8] Kevin W Eliceiri, Michael R Berthold, Ilya G Goldberg, Luis Ibáñez, Bangalore S Manjunath, Maryann E Martone, Robert F Murphy, Hanchuan Peng, Anne L Plant, Badrinath Roysam, Nico Stuurman, Jason R Swedlow, Pavel Tomancak, and Anne E Carpenter, “Biological imaging software tools,” *Nature Methods*, vol. 9, no. 7, pp. 697–710, June 2012.
- [9] Michael D Abràmoff, Paulo J Magalhães, and Sunanda 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] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, P. Golland, and David M Sabatini, “CellProfiler: image analysis software for identifying and quantifying cell phenotypes,” *Genome Biology*, vol. 7, no. 10, pp. R100, October 2006.
- [13] Michael R Lamprecht, David M Sabatini, and Anne E Carpenter, “CellProfilerTM: free, versatile software for automated biological image analysis,” *BioTechniques*, vol. 42, no. 1, pp. 71–75, January 2007.

-
- [14] Christoph Sommer, Christoph Straehle, Ullrich Köthe, and Fred 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.
- [15] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V Miller, Steve Pieper, and Ron Kikinis, “3D Slicer as an image computing platform for the Quantitative Imaging Network,” *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1323–1341, November 2012.
- [16] Gert Wollny, Peter Kellman, María-Jesus Ledesma-Carbayo, Matthew M Skinner, Jean-Jaques Hublin, and Thomas Hierl, “MIA-A free and open source software for gray scale medical image analysis,” *Source Code for Biology and Medicine*, vol. 8, no. 1, pp. 1–20, October 2013.
- [17] Ignacio Larrabide, Pedro Omedas, Yves Martelli, Xavier Planes, Maarten Nieber, Juan A Moya, Constantine Butakoff, Rafael Sebastián, Oscar Camara, Mathieu De Craene, Bart H Bijmens, and Alejandro F Frangi, “GIMIAS: An open source framework for efficient development of research tools and clinical prototypes,” in *International Conference on Functional Imaging and Modeling of the Heart (FIMH 2009)*, Nice, France, June 3–5, 2009, pp. 417–426.
- [18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [19] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, February 2017.
- [20] Michael Jermyn, Joannie Desroches, Jeanne Mercier, Marie-Andrée Tremblay, Karl St-Arnaud, Marie-Christine Guiot, Kevin Petrecca, and Frederic Leblond, “Neural networks improve brain cancer detection with Raman spectroscopy in the presence of operating room light artifacts,” *Journal of Biomedical Optics*, vol. 21, no. 9, pp. 094002, 2016.

- [21] Jian Zhou and Olga G Troyanskaya, “Predicting effects of noncoding variants with deep learning–based sequence model,” *Nature Methods*, vol. 12, no. 10, pp. 931–934, August 2015.
- [22] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey, “Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning,” *Nature Biotechnology*, vol. 33, no. 8, pp. 831–838, July 2015.
- [23] James J DiCarlo, Davide Zoccolan, and Nicole C Rust, “How does the brain solve visual object recognition?,” *Neuron*, vol. 73, no. 3, pp. 415–434, February 2012.
- [24] Alan C Bovik, *The essential guide to image processing*, Academic Press, 2009.
- [25] Richik N Ghosh and Watt W Webb, “Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules,” *Biophysical Journal*, vol. 66, no. 5, pp. 1301–1318, May 1994.
- [26] Nathalie Harder, Felipe Mora-Bermúdez, William J Godinez, Jan Ellenberg, Roland Eils, and Karl Rohr, “Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’06)*, Copenhagen, Denmark, October 1-6, 2006, pp. 840–848.
- [27] Meng Wang, Xiaobo Zhou, Fuhai Li, Jeremy Huckins, Randall W King, and Stephen T C Wong, “Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy,” *Bioinformatics*, vol. 24, no. 1, pp. 94–101, January 2008.
- [28] F. Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, no. 1, pp. 113–125, July 1994.
- [29] Y. Azuma and S. Onami, “Biologically constrained optimization based cell membrane segmentation in *C. elegans* embryos,” *BMC Bioinformatics*, vol. 18, no. 1, pp. 307, June 2017.

-
- [30] K. R. Mosaliganti, R. R. Noche, F. Xiong, I. A. Swinburne, and S. G. Megason, “ACME: Automated cell morphology extractor for comprehensive reconstruction of cell membranes,” *PLoS Computational Biology*, vol. 8, no. 12, pp. e1002780, December 2012.
- [31] A. Chakraborty, M. M. Perales, G. V. Reddy, and A. K. Roy-Chowdhury, “Adaptive geometric tessellation for 3D reconstruction of anisotropically developing cells in multilayer tissues from sparse volumetric microscopy images,” *PLoS One*, vol. 8, no. 8, pp. e67202, August 2013.
- [32] J. Stegmaier, F. Amat, W. C. Lemon, K. McDole, Y. Wan, G. Teodoro, R. Mikut, and P. J. Keller, “Real-time three-dimensional cell segmentation in large-scale microscopy data of developing embryos,” *Developmental Cell*, vol. 36, no. 2, pp. 225–240, January 2016.
- [33] Anne E Carpenter, Lee Kametsky, and Kevin W Eliceiri, “A call for bioimaging software usability,” *Nature methods*, vol. 9, no. 7, pp. 666–670, July 2012.
- [34] Albert Cardona and Pavel Tomancak, “Current challenges in open-source bioimage informatics,” *Nature Methods*, vol. 9, no. 7, pp. 661–665, June 2012.
- [35] 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.
- [36] Demetri Terzopoulos, Andrew Witkin, and Michael Kass, “Constraints on deformable models: Recovering 3D shape and nonrigid motion,” *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, August 1988.
- [37] 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.
- [38] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*, Springer-Verlag New York, Inc., 1st edition, 1998.
- [39] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, “Fast global minimization of the active contour/snake model,” *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, June 2007.

- [40] 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.
- [41] 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.
- [42] Christophe Zimmer and J-C Olivo-Marin, "Coupled parametric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1838–1842, November 2005.
- [43] A. Dufour, R. Thibeaux, E. Labruyère, N. Guillén, and J.-C. Olivo-Marin, "3-D Active meshes: Fast discrete deformable models for cell tracking in 3-D time-lapse microscopy," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1925–1937, July 2011.
- [44] 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.
- [45] 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.
- [46] Johan Montagnat, Hervé Delingette, and Nicholas Ayache, "A review of deformable surfaces: topology, geometry and deformation," *Image and Vision Computing*, vol. 19, no. 14, pp. 1023–1040, December 2001.
- [47] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, February 1997.
- [48] 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.
- [49] J. Cardinale, G. Paul, and I. F. Sbalzarini, "Discrete region competition for unknown numbers of connected regions," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3531–3545, August 2012.

-
- [50] P. Brigger, J. Hoeg, and M. Unser, “B-Spline snakes: A flexible tool for parametric contour detection,” *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1484–1496, September 2000.
- [51] 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.
- [52] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, “Spline-based deforming ellipsoids for interactive 3D bioimage segmentation,” *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3926–3940, October 2013.
- [53] Stanley Osher and James 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.
- [54] Ravi Malladi, James A Sethian, and Baba C Vemuri, “Shape modeling with front propagation: A level set approach,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, February 1995.
- [55] Nikos Paragios and Rachid Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, March 2000.
- [56] Kaihua Zhang, Huihui Song, and Lei Zhang, “Active contours driven by local image fitting energy,” *Pattern Recognition*, vol. 43, no. 4, pp. 1199–1206, April 2010.
- [57] 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.
- [58] 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.

- [59] 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.
- [60] Tim McInerney and Demetri Terzopoulos, “T-snakes: Topology adaptive snakes,” *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, June 2000.
- [61] 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.
- [62] 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.
- [63] M. Jacob, T. Blu, and M. Unser, “A unifying approach and interface for spline-based snakes,” in *Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI’01)*, San Diego CA, USA, February 17-22, 2001, vol. 4322, pp. 340–347.
- [64] 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.
- [65] Adrien Depeursinge, Tünde Szilágyi, Yan Liu, Kázmer Kovács, Reena P Thomas, Kristen W Yeom, Nancy Fischbein, Daniel L Rubin, Michael Iv, and Olivier Gevaert, “Locoregional radiogenomic models capture gene expression heterogeneity in glioblastoma,” *bioRxiv*, p. 304105, April 2018.
- [66] E. Meijering, “Cell segmentation: 50 years down the road,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, September 2012.
- [67] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, “Advanced level-set-based cell tracking in time-lapse fluorescence microscopy,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 852–867, March 2010.

-
- [68] A. Badoual, D. Schmitter, and M. Unser, “Locally refinable parametric snakes,” in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP’15)*, Québec QC, Canada, September 27–30, 2015, pp. TEC–P21.2.
- [69] A. Badoual, D. Schmitter, and M. Unser, “Local refinement for 3D deformable parametric surfaces,” in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP’16)*, Phoenix AZ, USA, September 25–28, 2016, pp. 1086–1090.
- [70] A. Badoual, D. Schmitter, V. Uhlmann, and M. Unser, “Multiresolution subdivision snakes,” *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1188–1201, March 2017.
- [71] A. Badoual, P. Novara, L. Romani, D. Schmitter, and M. Unser, “A non-stationary subdivision scheme for the construction of deformable models with sphere-like topology,” *Graphical Models*, vol. 94, pp. 38–51, November 2017.
- [72] A. Badoual, L. Romani, and M. Unser, “Active subdivision surfaces for the segmentation of biomedical volumes,” *in preparation*.
- [73] A. Badoual, A. Galan, D. Sage, and M. Unser, “Deforming tessellations for the segmentation of cell aggregates,” in *Proceedings of the Sixteenth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI’19)*, Venice, Italian Republic, April 8–11, 2019, in press.
- [74] A. Badoual, M. Unser, and A. Depeursinge, “Texture-driven parametric snakes for interactive image segmentation,” *submitted*.
- [75] 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.
- [76] A. Badoual, J. Fageot, and M. Unser, “Periodic splines and Gaussian processes for the resolution of linear inverse problems,” *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 6047–6061, November 15, 2018.
- [77] D. Schmitter, J. Fageot, A. Badoual, P. Garcia-Amorena, and M. Unser, “Compactly-supported smooth interpolators for shape modeling with varying resolution,” *Graphical Models*, vol. 94, pp. 52–64, November 2017.

- [78] M. Unser, “Sampling—50 Years after Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.
- [79] M. A. T. Figueiredo and J. M. N. Leitão, “Bayesian Estimation of Ventricular Contours in Angiographic Images,” *IEEE Transactions on Medical Imaging*, vol. 11, no. 3, pp. 416–429, September 1992.
- [80] P. Thévenaz, R. Delgado-Gonzalo, and M. Unser, “The ovuscul,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, February 2011.
- [81] 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.
- [82] Helmuth Späth, “Exponential spline interpolation,” *Computing*, vol. 4, no. 3, pp. 225–233, September 1969.
- [83] Carl de Boor and Amos Ron, “The exponentials in the span of the multiinteger translates of a compactly supported function; quasiinterpolation and approximation order,” *Journal of the London Mathematical Society*, vol. s2-45, no. 3, pp. 519–535, June 1992.
- [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] A. N. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” *Soviet Mathematics Doklady*, vol. 4, pp. 1035–1038, 1963.
- [86] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, vol. 1, Prentice Hall Upper Saddle River, NJ, 2000.
- [87] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer Science & Business Media, 2004.
- [88] G. Wahba, *Spline Models for Observational Data*, SIAM, 1990.

-
- [89] 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.
- [90] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, “T-spline simplification and local refinement,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 276–283, August 2004.
- [91] C. Giannelli, B. Jüttler, and H. Speleers, “Strongly stable bases for adaptively refined multilevel spline spaces,” *Advances in Computational Mathematics*, vol. 40, no. 2, pp. 459–490, April 2014.
- [92] T. Dokken, T. Lyche, and K. F. Pettersen, “Polynomial splines over locally refined box-partitions,” *Computer Aided Geometric Design*, vol. 30, no. 3, pp. 331–356, March 2013.
- [93] C. Giannelli, B. Jüttler, and H. Speleers, “THB-splines: The truncated basis for hierarchical splines,” *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 485–498, October 2012.
- [94] Carla Manni, Francesca Pelosi, and Hendrik Speleers, “Local hierarchical h-refinements in IgA based on generalized B-splines,” in *International Conference on Mathematical Methods for Curves and Surfaces (MMCS’12)*, Oslo, Norway, June 28 – July 3, 2012, pp. 341–363.
- [95] Martin Vetterli and Jelena Kovačević, *Wavelets and Subband Coding*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1995.
- [96] N. Leonardi and D. Van De Ville, “Tight Wavelet Frames on Multislice Graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3357–3367, July 2013.
- [97] S. G. Chang, Z. Cvetković, and M. Vetterli, “Locally adaptive wavelet-based image interpolation,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1471–1485, June 2006.
- [98] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, april 1992.

-
- [99] Stephane Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [100] C. Vonesch, T. Blu, and M. Unser, “Generalized Daubechies wavelet families,” *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4415–4429, September 2007.
- [101] Costanza Conti and Lucia Romani, “Algebraic conditions on non-stationary subdivision symbols for exponential polynomial reproduction,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 4, pp. 543 – 556, September 2011.
- [102] L. Romani, “From approximating subdivision schemes for exponential splines to high-performance interpolating algorithms,” *Journal of Computational and Applied Mathematics*, vol. 224, no. 1, pp. 383–396, February 2009.
- [103] M.-J. Lai and L. L. Schumaker, “On the approximation power of bivariate splines,” *Advances in Computational Mathematics*, vol. 9, no. 3-4, pp. 251–279, November 1998.
- [104] N. Dyn and E. Farkhi, “Spline subdivision schemes for compact sets. A survey,” *Serdica Mathematical Journal*, vol. 28, no. 4, pp. 349–360, 2002.
- [105] M. Charina, C. Conti, and L. Romani, “Reproduction of exponential polynomials by multivariate non-stationary subdivision schemes with a general dilation matrix,” *Numerische Mathematik*, vol. 127, no. 2, pp. 223–254, June 2014.
- [106] P. Novara and L. Romani, “Building blocks for designing arbitrarily smooth subdivision schemes with conic precision,” *Journal of Computational and Applied Mathematics*, vol. 279, pp. 67–79, May 2015.
- [107] Tony DeRose, Michael Kass, and Tien Truong, “Subdivision surfaces in character animation,” in *Proceedings of the Twenty-Fifth Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH’98)*, Orlando FL, USA, July 19-24, 1998, pp. 85–94.

-
- [108] Aaron Lee, Henry Moreton, and Hugues Hoppe, “Displaced subdivision surfaces,” in *Proceedings of the Twenty-Seventh Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, New Orleans LA, USA, July 23-28, 2000, pp. 85–94.
- [109] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia PA, USA, 1992.
- [110] C. Loop, “Smooth subdivision surfaces based on triangles,” *Master’s thesis, University of Utah, Department of Mathematics*, August 1987.
- [111] N. Dyn, D. Levin, and J. A. Gregory, “A butterfly subdivision scheme for surface interpolation with tension control,” *ACM Transactions on Graphics*, vol. 9, no. 2, pp. 160–169, April 1990.
- [112] A. Myles, K. Karčiauskas, and J. Peters, “Extending Catmull-Clark subdivision and PCCM with polar structures,” in *Proceedings of the fifteenth IEEE Pacific Conference on Computer Graphics and Applications (PG'07)*, Maui HI, USA, October 29-November 2, 2007, pp. 313–320.
- [113] Joe Warren and Henrik Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*, Morgan Kaufmann Publishers Inc., San Francisco CA, USA, 1st edition, 2002.
- [114] N. Dyn, “Subdivision schemes in CAGD,” *Advances in Numerical Analysis*, vol. 2, pp. 36–104, 1992.
- [115] C Conti and K Hormann, “Polynomial reproduction for univariate subdivision schemes of any arity,” *Journal of Approximation Theory*, vol. 163, no. 4, pp. 413–437, April 2011.
- [116] N. Dyn, J. Gregory, and D. Levin, “Analysis of uniform binary subdivision schemes for curve design,” *Constructive Approximation*, vol. 7, no. 1, pp. 127–147, December 1991.
- [117] Nira Dyn, David Levin, and Jungho Yoon, “Analysis of univariate nonstationary subdivision schemes with application to Gaussian-based interpolatory schemes,” *SIAM Journal on Mathematical Analysis*, vol. 39, no. 2, pp. 470–488, 2007.

-
- [118] O. Rioul, “Simple regularity criteria for subdivision schemes,” *SIAM Journal on Mathematical Analysis*, vol. 23, no. 6, pp. 1544–1576, 1992.
- [119] G. Deslauriers and S. Dubuc, “Symmetric iterative interpolation processes,” *Constructive Approximation*, vol. 5, no. 1, pp. 49–68, 1989.
- [120] C. Conti, L. Gemignani, and L. Romani, “From symmetric subdivision masks of Hurwitz type to interpolatory subdivision masks,” *Linear Algebra and its Applications*, vol. 431, no. 10, pp. 1971–1987, 2009.
- [121] Naoki Saito and Gregory Beylkin, “Multiresolution representations using the autocorrelation functions of compactly supported wavelets,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3584–3590, December 1993.
- [122] I. Daubechies, “Orthonormal bases of compactly supported wavelets,” *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, October 1988.
- [123] N. Dyn, D. Levin, and A. Luzzatto, “Exponentials reproducing subdivision schemes,” *Foundations of Computational Mathematics*, vol. 3, no. 2, pp. 187–206, May 2003.
- [124] J. M Lane and R. F Riesenfeld, “A theoretical development for the computer generation and display of piecewise polynomial surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 1, pp. 35–46, January 1980.
- [125] C. Conti, L. Gemignani, and L. Romani, “From approximating to interpolatory non-stationary subdivision schemes with the same generation properties,” *Advances in Computational Mathematics*, vol. 35, pp. 217–241, November 2011.
- [126] Albert Cohen and Nira Dyn, “Nonstationary subdivision schemes and multiresolution analysis,” *SIAM Journal on Mathematical Analysis*, vol. 27, no. 6, pp. 1745–1769, November 1996.
- [127] Lars-Erik Andersson and Neil F Stewart, *Introduction to the Mathematics of Subdivision Surfaces*, vol. 120, SIAM, 2010.

-
- [128] D. Zorin, P. Schröder, and W. Sweldens, “Interpolating subdivision for meshes with arbitrary topology,” in *Proceedings of the Twenty-Third Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH’96)*, New Orleans LA, USA, August 4 -9, 1996, pp. 189–192.
- [129] L. Kobbelt, “Interpolatory subdivision on open quadrilateral nets with arbitrary topology,” in *Computer Graphics Forum (EUROGRAPHICS’96)*, Poitiers, France, August 26–30, 1996, vol. 15, pp. 409–420.
- [130] C. Deng and W. Ma, “A unified interpolatory subdivision scheme for quadrilateral meshes,” *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 23, June 2013.
- [131] Paola Novara and Lucia Romani, “On extraordinary rules of quad-based interpolatory subdivision schemes,” *Computer Aided Geometric Design*, vol. 35–36, pp. 225–242, May 2015.
- [132] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, no. 6, pp. 356–360, November 1978.
- [133] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, November 1978.
- [134] Mei-e Fang, Weiyin Ma, and Guozhao Wang, “A generalized surface subdivision scheme of arbitrary order with a tension parameter,” *Computer-Aided Design*, vol. 49, pp. 8–17, April 2014.
- [135] M. Charina, C. Conti, N. Guglielmi, and V. Protasov, “Regularity of non-stationary subdivision: A matrix approach,” *Numerische Mathematik*, vol. 135, no. 3, pp. 639–678, March 2017.
- [136] C. Conti, M. Donatelli, L. Romani, and P. Novara, “Convergence and C^1 -regularity analysis of non-stationary subdivision schemes near extraordinary vertices and faces,” *arXiv preprint arXiv:1707.01954v3*, 2017.
- [137] J. Peters and U. Reif, *Subdivision Surfaces*, vol. 3, Springer Science & Business Media, 2008.

- [138] Assaf Hoogi, Arjun Subramaniam, Rishi Veerapaneni, and Daniel L. Rubin, “Adaptive Estimation of Active Contour Parameters Using Convolutional Neural Networks and Texture Analysis,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 3, pp. 781–791, March 2017.
- [139] D. Reska, C. Boldak, and M. Kretowski, “A texture-based energy for active contour image segmentation,” pp. 187–194, 2015.
- [140] Chen Sagiv, Nir A Sochen, and Yehoshua Y Zeevi, “Integrated active contours for texture segmentation,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1633–1646, June 2006.
- [141] A. Depeursinge and J. Fageot, “Biomedical texture operators and aggregation functions: A methodological review and user’s guide,” in *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, A. Depeursinge, O.S. Al-Kadi, and J.R. Mitchell, Eds., chapter 3, pp. 55–94. Academic Press, London, United Kingdom, 2017.
- [142] Francesco Bianconi and Antonio Fernández, “Evaluation of the effects of Gabor filter parameters on texture classification,” *Pattern Recognition*, vol. 40, no. 12, pp. 3325–3335, December 2007.
- [143] 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.
- [144] A. Depeursinge, “Multiscale and multidirectional biomedical texture analysis: Finding the needle in the haystack,” in *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, A. Depeursinge, O.S. Al-Kadi, and J.R. Mitchell, Eds., chapter 2, pp. 29–53. Academic Press, London, United Kingdom, 2017.
- [145] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.
- [146] Adrien Depeursinge, Zsuzsanna Püspöki, John-Paul Ward, and Michael Unser, “Steerable wavelet machines (SWM): Learning moving frames for

- texture classification,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1626–1636, April 2017.
- [147] Ronald A Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Human Genetics*, vol. 7, no. 2, pp. 179–188, September 1936.
- [148] Aleix M Martínez and Avinash C Kak, “PCA versus LDA,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, February 2001.
- [149] Georg Umlauf, “Analyzing the characteristic map of triangular subdivision schemes,” *Constructive Approximation*, vol. 16, no. 1, pp. 145–155, January 2000.
- [150] D. Schmitter, C. Gaudet-Blavignac, D. Piccini, and M. Unser, “New parametric 3D snake for medical segmentation of structures with cylindrical topology,” in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP’15)*, Québec QC, Canada, September 27–30, 2015, pp. TEC–P21.2.
- [151] M. Jacob, T. Blu, and M. Unser, “Sampling of periodic signals: A quantitative error analysis,” *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1153–1159, May 2002.
- [152] F. Precioso and M. Barlaud, “B-spline active contour with handling of topology changes for fast video segmentation,” *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 6, pp. 555–560, December 2002.
- [153] Y. Zhang, B. J. Matuszewski, A. Histace, and F. Precioso, “Statistical model of shape moments with active contour evolution for shape detection and segmentation,” *Journal of Mathematical Imaging and Vision*, vol. 47, no. 1–2, pp. 35–47, September 2013.
- [154] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, “Image segmentation using active contours: Calculus of variations or shape gradients?,” *SIAM Journal of Applied Mathematics*, vol. 63, no. 6, pp. 2128–2154, September 2003.
- [155] T. Boonnuk, S. Srisuk, and T. Sripramong, “Texture segmentation using active contour model with edge flow vector,” *International Journal of Information and Electronics Engineering*, vol. 5, no. 2, pp. 107–111, March 2015.

- [156] S. V. Pons, J. L. G. Rodríguez, and O. L. V. Pérez, “Active contour algorithm for texture segmentation using a texture feature set,” in *Proceedings of the Nineteenth IEEE International Conference on Pattern Recognition (ICPR 2008)*, Tampa FL, USA, December 8-11, 2008.
- [157] N. Paragios and R. Deriche, “Geodesic active regions and level set methods for supervised texture segmentation,” *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, February 2002.
- [158] J. Lu, G. Wang, and Z. Pan, “Nonlocal active contour model for texture segmentation,” *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10991–11001, April 2017.
- [159] B. Sandberg, T. Chan, and L. Vese, “A level-set and Gabor-based active contour algorithm for segmenting textured images,” in *UCLA Department of Mathematics CAM report*, 2002, pp. 1–10.
- [160] P. Moallem, H. Tahvilian, and S. A. Monadjemi, “Parametric active contour model using Gabor balloon energy for texture segmentation,” *Signal, Image and Video Processing*, vol. 10, no. 2, pp. 351–358, February 2016.
- [161] Y. Gao, S. Bouix, M. Shenton, and A. Tannenbaum, “Sparse texture active contour,” *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3866–3878, October 2013.
- [162] G. Wang, Z. Pan, Q. Dong, X. Zhao, Z. Zhang, and J. Duan, “Unsupervised texture segmentation using active contour model and oscillating information,” *Journal of Applied Mathematics*, vol. 2014, pp. 1–11, June 2014.
- [163] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’15)*, Munich, Germany, October 5–9, 2015, pp. 234–241.
- [164] Christian Rupprecht, Elizabeth Huaroc, Maximilian Baust, and Nassir Navab, “Deep active contours,” *arXiv preprint arXiv:1607.05074*, 2016.
- [165] Tuan Anh Ngo, Zhi Lu, and Gustavo Carneiro, “Combining deep learning and level set for the automated segmentation of the left ventricle of the heart

- from cardiac cine magnetic resonance,” *Medical Image Analysis*, vol. 35, pp. 159–171, January 2017.
- [166] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang, “Deep level sets for salient object detection,” in *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR 2017)*, July 21–26, 2017, pp. 540–549.
- [167] Yading Yuan, Ming Chao, and Yeh-Chi Lo, “Automatic skin lesion segmentation using deep fully convolutional networks with Jaccard distance,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 9, pp. 1876 – 1886, September 2017.
- [168] Michal Haindl and Stanislav Mikš, “Texture segmentation benchmark,” in *Proceedings of the Nineteenth IEEE International Conference on Pattern Recognition (ICPR 2008)*, Tampa FL, USA, December 8–11, 2008.
- [169] Daniel Reska, Krzysztof Jurczuk, Cezary Boldak, and Marek Kretowski, “MESA: Complete approach for design and evaluation of segmentation methods using real and simulated tomographic images,” *Biocybernetics and Biomedical Engineering*, vol. 34, no. 3, pp. 146–158, 2014.
- [170] F. Orderud and S. I. Rabben, “Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces,” in *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR’08)*, Anchorage AK, USA, June 23–28, 2008, pp. 1–8.
- [171] J. Hug, C. Brechbühler, and G. Székely, “Tamed snake: A particle system for robust semi-automatic segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’99)*, Cambridge, UK, September 19–22, 1999, vol. 1679, pp. 106–116.
- [172] N. Dyn, D. Levin, and J. A. Gregory, “A 4-point interpolatory subdivision scheme for curve design,” *Computer Aided Geometric Design*, vol. 4, no. 4, pp. 257–268, December 1987.
- [173] C. V. Beccari, G. Casciola, and L. Romani, “Construction and characterization of non-uniform local interpolating polynomial splines,” *Journal of Computational Applied Mathematics*, vol. 240, pp. 5–19, March 2013.

- [174] M. Antonelli, C. V. Beccari, and G. Casciola, "A general framework for the construction of piecewise-polynomial local interpolants of minimum degree," *Advances in Computational Mathematics*, vol. 40, no. 4, pp. 945–976, August 2014.
- [175] 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.
- [176] P. Brigger and M. Unser, "Multi-scale B-Spline snakes for general contour detection," in *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing VI*, San Diego CA, USA, July 19–24, 1998, vol. 3458, pp. 92–102.
- [177] B. Leroy, I. L. Herlin, and L. D. Cohen, "Multi-resolution algorithms for active contour models," in *Proceedings of the Twelfth International Conference on Analysis and Optimization of Systems (ICAOS'96)*, Paris, France, June 26–28, 1996, pp. 58–65.
- [178] P. G. Crichton, C. Affourtit, and A. L. Moore, "Identification of a mitochondrial alcohol dehydrogenase in *Schizosaccharomyces pombe*: New insights into energy metabolism," *Biochemical Journal*, vol. 401, no. 2, pp. 459–464, January 2007.
- [179] G. R. Serjeant and B. E. Serjeant, *Sickle Cell Disease*, Oxford University Press, third edition, 2001.
- [180] T. R. Jones, A. Carpenter, and P. Golland, "Voronoi-based segmentation of cells on image manifolds," in *International Workshop on Computer Vision for Biomedical Image Applications (CVBIA 2005)*, Beijing, China, October 21, 2005, pp. 535–543.
- [181] E. Ficarra, S. Di Cataldo, A. Acquaviva, and E. Macii, "Automated segmentation of cells with IHC membrane staining," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1421–1429, May 2011.
- [182] D. S. Cheng, V. Murino, and M. A. T. Figueiredo, "Clustering under prior knowledge with application to image segmentation," in *Advances in Neural*

- Information Processing Systems 19*, Vancouver, British Columbia, Canada, December 4–7, 2006, pp. 401–408.
- [183] S. Dimopoulos, C. E. Mayer, F. Rudolf, and J. Stelling, “Accurate cell segmentation in microscopy images using membrane patterns,” *Bioinformatics*, vol. 30, no. 18, pp. 2644–2651, September 2014.
- [184] A. Fabijańska, “Segmentation of corneal endothelium images using a U-Net-based convolutional neural network,” *Artificial Intelligence in Medicine*, vol. 88, pp. 1–13, June 2018.
- [185] J.-P. Bergeest and K. Rohr, “Efficient globally optimal segmentation of cells in fluorescence microscopy images using level sets and convex energy functionals,” *Medical Image Analysis*, vol. 16, no. 7, pp. 1436–1444, October 2012.
- [186] 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.
- [187] P. Jasiobedzki, “Adaptive adjacency graphs,” in *Proceedings of the SPIE Conference on Geometric Methods in Computer Vision II*, San Diego CA, USA, June 23, 1993, vol. 2031, pp. 294–304.
- [188] M. Butenuth, “Segmentation of imagery using network snakes,” in *Symposium of ISPRS Commission III, Photogrammetric Computer Vision (PCV’06)*, Bonn, Germany, September 20–22, 2006, vol. 36, pp. 1–6.
- [189] P. Soille and L. Vincent, “Determining watersheds in digital pictures via flooding simulations,” in *Proceedings of the SPIE Conference on Visual Communication and Image Processing*, Lausanne, Switzerland, September 1, 1990, vol. 1360, pp. 240–250.
- [190] Pieter H Kitslaar, Ronald van’t Klooster, Marius Staring, Boudewijn P F Lelieveldt, and Rob J van der Geest, “Segmentation of branching vascular structures using adaptive subdivision surface fitting,” in *Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI’15)*, Orlando, FL, USA, February 21–26, 2015, vol. 9413, p. 94133Z.

- [191] Mathew L Coleman, Erik A Sahai, Margaret Yeo, Marta Bosch, Ann Dewar, and Michael F Olson, “Membrane blebbing during apoptosis results from caspase-mediated activation of ROCK I,” *Nature Cell Biology*, vol. 3, pp. 339–345, April 2001.
- [192] L. Romani, A. Badoual, and M. Unser, “Normal-based interpolating subdivision for the geometric representation of deformable models,” in *Proceedings of the Sixteenth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'19)*, Venice, Italian Republic, April 8–11, 2019, in press.
- [193] L. Rebollo-Neira and D. Lowe, “Optimized orthogonal matching pursuit approach,” *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 137–140, April 2002.
- [194] P. J. Davis, *Circulant Matrices*, Pure and Applied Mathematics. Wiley, 1979.
- [195] M. Unser and T. Blu, “Fractional splines and wavelets,” *SIAM Review*, vol. 42, no. 1, pp. 43–67, March 2000.
- [196] J Anthony Parker, Robert V Kenyon, and Donald E Troxel, “Comparison of interpolating methods for image resampling,” *IEEE Transactions on Medical Imaging*, vol. 2, no. 1, pp. 31–39, March 1983.
- [197] M. Jacob, T. Blu, and M. Unser, “An exact method for computing the area moments of wavelet and spline curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 633–642, June 2001.
- [198] M. R. Banham and A. K. Katsaggelos, “Digital image restoration,” *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, March 1997.
- [199] N. B. Karayiannis and A. N. Venetsanopoulos, “Regularization theory in image restoration—The stabilizing functional approach,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1155–1179, July 1990.
- [200] M. A. T. Figueiredo and R. D. Nowak, “An EM algorithm for wavelet-based image restoration,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, August 2003.

-
- [201] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, March 2011.
- [202] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, CRC press, 1998.
- [203] B. Adcock and A. C. Hansen, “Generalized sampling and infinite-dimensional compressed sensing,” *Foundations of Computational Mathematics*, vol. 16, no. 5, pp. 1263–1323, October 2016.
- [204] Athanasios Papoulis, “Generalized sampling expansion,” *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, November 1977.
- [205] Y. C. Eldar and T. G. Dvorkind, “A minimum squared-error framework for generalized sampling,” *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2155–2167, June 2006.
- [206] E. L. Piccolomini, F. Zama, G. Zanghirati, and A. Formiconi, “Regularization methods in dynamic MRI,” *Applied Mathematics and Computation*, vol. 132, no. 2-3, pp. 325–339, November 2002.
- [207] E. Bostan, U. S. Kamilov, M. Nilchian, and M. Unser, “Sparse stochastic processes and discretization of linear inverse problems,” *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2699–2710, July 2013.
- [208] K. W. Cassel, *Variational Methods with Applications in Science and Engineering*, Cambridge University Press, 2013.
- [209] Fernand S. Cohen and Jin-Yinn Wang, “Part I: Modeling image curves using invariant 3-D object curve models—A path to 3-D recognition and shape estimation from image contours,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 1–12, January 1994.
- [210] M. Vetterli, P. Marziliano, and T. Blu, “Sampling signals with finite rate of innovation,” *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, June 2002.

- [211] I. Maravić and M. Vetterli, “Sampling and reconstruction of signals with finite rate of innovation in the presence of noise,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2788–2805, August 2005.
- [212] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, “Sparse sampling of signal innovations,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, March 2008.
- [213] H. Triebel, *Function Spaces and Wavelets on Domains*, vol. 7 of *EMS Tracts in Mathematics*, European Mathematical Society (EMS), Zürich, 2008.
- [214] J. Fageot, M. Unser, and J. P. Ward, “On the Besov regularity of periodic Lévy noises,” *Applied and Computational Harmonic Analysis*, vol. 42, no. 1, pp. 21–36, January 2017.
- [215] B. Schölkopf, R. Herbrich, and A. Smola, “A generalized representer theorem,” in *International Conference on Computational Learning Theory (COLT’01)*, Amsterdam, The Netherlands, July 16–19, 2001, pp. 416–426.
- [216] I. J. Schoenberg, “Spline functions and the problem of graduation,” *Proceedings of the National Academy of Sciences*, vol. 52, no. 4, pp. 947–950, August 1964.
- [217] 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.
- [218] L. Condat and T. Möller, “Quantitative error analysis for the reconstruction of derivatives,” *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2965–2969, June 2011.
- [219] Emmanuel J Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, February 2006.
- [220] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [221] Q. Denoyelle, V. Duval, and G. Peyré, “Support recovery for sparse deconvolution of positive measures,” *arXiv preprint arXiv:1506.08264*, 2015.

-
- [222] M. Unser, J. Fageot, and J. P. Ward, “Splines are universal solutions of linear inverse problems with generalized TV regularization,” *SIAM Review*, vol. 59, no. 4, pp. 769–793, December 2017.
- [223] Harshit Gupta, Julien Fageot, and Michael Unser, “Continuous-domain solutions of linear inverse problems with Tikhonov versus generalized TV regularization,” *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4670–4684, September 2018.
- [224] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, 2005.
- [225] T. Kailath, “An innovations approach to least-squares estimation—Part I: Linear filtering in additive white noise,” *IEEE Transactions on Automatic Control*, vol. 13, no. 6, pp. 646–655, December 1968.
- [226] 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.
- [227] G. S. Kimeldorf and G. Wahba, “Spline functions and stochastic processes,” *Sankhyā: The Indian Journal of Statistics, Series A*, vol. 32, no. 2, pp. 173–180, June 1970.
- [228] 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.
- [229] 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.
- [230] M. Unser and P.D. Tafti, *An Introduction to Sparse Stochastic Processes*, Cambridge University Press, Cambridge, United Kingdom, 2014, 367 p.
- [231] I. J. Schoenberg, “On trigonometric spline interpolation,” *Journal of Mathematics and Mechanics*, vol. 13, no. 5, pp. 795–825, 1964.

- [232] M. Golomb, “Approximation by periodic spline interpolants on uniform meshes,” *Journal of Approximation Theory*, vol. 1, no. 1, pp. 26–65, June 1968.
- [233] C. de Boor, *A Practical Guide to Splines*, vol. 27, Springer-Verlag New York, 1978.
- [234] L. Schwartz, *Théorie des distributions*, Hermann, 1966.
- [235] I. J. Schoenberg, *Cardinal Spline Interpolation*, Philadelphia, PA: SIAM, 1973.
- [236] L. Schumaker, *Spline Functions: Basic Theory*, Cambridge University Press, third edition, 2007.
- [237] M. H. Schultz and R. S. Varga, “L-splines,” *Numerische Mathematik*, vol. 10, no. 4, pp. 345–369, November 1967.
- [238] M. Unser, “Cardinal exponential splines: Part II—Think analog, act digital,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1439–1449, April 2005.
- [239] R. Panda and M. Dash, “Fractional generalized splines and signal processing,” *Signal Processing*, vol. 86, no. 9, pp. 2340–2350, September 2006.
- [240] M. Unser, J. Fageot, and H. Gupta, “Representer theorems for sparsity-promoting ℓ_1 regularization,” *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 5167–5180, September 2016.
- [241] Nathaniel R Goodman, “Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction),” *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 152–177, March 1963.
- [242] D. Revuz and M. Yor, *Continuous Martingales and Brownian Motion*, vol. 293, Springer Science & Business Media, 2013.
- [243] B. Simon, “Distributions and their Hermite expansions,” *Journal of Mathematical Physics*, vol. 12, no. 1, pp. 140–148, 1971.

-
- [244] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer Science & Business Media, 1998.
- [245] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Science & Business Media, 2010.
- [246] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic press, 2008.
- [247] E. J. Candès and C. Fernandez-Granda, “Towards a mathematical theory of super-resolution,” *Communications on Pure and Applied Mathematics*, vol. 67, no. 6, pp. 906–956, June 2014.
- [248] H. Wendland, *Scattered Data Approximation*, vol. 17, Cambridge university press, 2004.

Curriculum Vitæ



Anaïs Badoual

PhD Student

Experience

- 11/14 - 01/15 **Internship on “Segmentation Based on Subdivisions”**
[Biomedical Imaging Group - École Polytechnique Fédérale de Lausanne, Switzerland](#)
Advisor: Prof. Michael Unser
- 03/14 - 08/14 **Internship on “Development of a 3D Vessel Reconstruction Algorithm for Image-Guided Interventional Procedures”**
[Department of Computer Engineering - École Polytechnique Fédérale de Montréal, Canada](#)
Advisor: Prof. Samuel Kadoury
- 06/13 - 08/13 **Internship on “Dosimetric Effects of Daily Anatomic and Setup Variations for Rapid Arc Radiotherapy of Head and Neck Cancer”**
[Radiotherapy Dpt of Medical Physics - General Hospital of Montréal - McGill University, Canada](#)
Advisor: Assistant Prof. William Parker
- 2012 - 2013 **Engineering Project on “Development of Coupled Oculographic and Tactile Recordings for the Monitoring of Children with Eye Pathology”**
[Télécom Physique Strasbourg, France](#)
Partnership with the motive education institute “les Iris”

Education

- 2015 - Now **PhD in Biomedical Image Processing**
[École Polytechnique Fédérale de Lausanne, Switzerland](#)
Project: “Active Contours and Surfaces for Biomedical Image Segmentation”
Laboratory: Biomedical Imaging Group
Advisor: Prof. Michael Unser
- 2012 - 2014 **Master's Degree in Images, Robotics and Engineering for Life Science**
[Télécom Physique Strasbourg, France](#)
Specialization: Image and Vision
- 2011 - 2014 **Engineer's Degree in Information Technology and Communication for Health** - 2nd of the promotion
[Télécom Physique Strasbourg, France](#)
Specialization: Innovative Diagnostics and Medical Treatments
- 2009 - 2011 **Intensive Two-Year Preparation Course for the Competitive Entrance to French Engineering Schools**
[Lycée Charlemagne, Paris, France](#)
Specialization: Mathematics, Physics and Engineering Sciences
- 2007 - 2009 **Scientific High School Diploma** with highest honor
[Lycée Maurice Ravel, Paris, France](#)
Specialization: Mathematics

Birth

Paris, France
01 October 1991

Address

Av. de Floréal 42
1008 Prilly
Switzerland

Tel

+41 76 382 23 01

Mail

anaïs.badoual@epfl.ch

Web

bigwww.epfl.ch/demo/deformable-models-segmentation

Programing

Unix

Mathematica

Maple

Matlab

Caffe

Icy

ImageJ

C/C++

Java

(basic) Python

Languages

English fluent

Spanish basic

Skills

Image processing
Image segmentation
Signal processing

Deep learning
Development of plugins
Anatomy-Physiology

Physics imagers
Medical procedures

Publications

Journal Papers

A. Badoual, M. Unser, A. Depeursinge

Texture-Driven Parametric Snakes for Interactive Image Segmentation

Under Review

A. Badoual, J. Fageot, M. Unser

Periodic Splines and Gaussian Processes for the Resolution of Linear Inverse Problems

IEEE Transactions on Signal Processing, vol. 66, no. 22, pp. 6047-6061, November 2018

A. Badoual, P. Novara, L. Romani, D. Schmitter, M. Unser

A Non-Stationary Subdivision Scheme for the Construction of Deformable Models with Sphere-Like Topology

Graphical Models, vol. 94, pp. 38-51, November 2017

D. Schmitter, J. Fageot, A. Badoual, P. Garcia-Amorena, M. Unser

Compactly-Supported Smooth Interpolators for Shape Modeling with Varying Resolution

Graphical Models, vol. 94, pp. 52-64, November 2017

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

R. Schorer, A. Badoual, B. Bastide, A. Vandebrouck, M. Licker, D. Sage

A Feasibility Study of Color Flow Doppler Vectorization for Automated Blood Flow Monitoring

Journal of Clinical Monitoring and Computing, vol. 31, no. 6, pp. 1167-1175, Dec 2017

A. Badoual, D. Schmitter, M. Unser

An Inner-Product Calculus for Periodic Functions and Curves

IEEE Signal Processing Letters, vol. 23, no. 6, pp. 878-882, June 2016

Conference Papers

L. Romani, A. Badoual, M. Unser

Normal-Based Interpolating Subdivision for the Geometric Representation of Deformable Models

Proceedings of the Sixteenth IEEE International Symposium on Biomedical Imaging: From-Nano to Macro (ISBI'19), Venice, Italian Republic, April 8–11, 2019, in press

A. Badoual, A. Galan, D. Sage, M. Unser

Deforming Tessellations for the Segmentation of Cell Aggregates

Proceedings of the Sixteenth IEEE International Symposium on Biomedical Imaging: From-Nano to Macro (ISBI'19), Venice, Italian Republic, April 8–11, 2019, in press

A. Badoual, D. Schmitter, M. Unser

Local Refinement for 3D Deformable Parametric Surfaces

Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP'16), Phoenix AZ, USA, September 25-28, 2016, pp. 1086-1090

A. Badoual, M. Gérard, B. de Leener, N. Abi-Jaoudeh, S. Kadoury

3D Vascular Path Planning of Chemo-Embolizations Using Segmented Hepatic Arteries from MR Angiography

Proceedings of the Thirteenth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'16), Prague, Czech Republic, April 13-16, 2016, pp. 1346-1349

M. Gérard, A. Tang, A. Badoual, F. Michaud, A. Bigot, G. Soulez, S. Kadoury

Visualization of Hepatic Arteries with 3D Ultrasound During Intra-Arterial Therapies

Proceedings of the SPIE International Symposium on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling (SPIE-IGPRIM'16), San Diego CA, USA, February 27-March 3, 2016, vol. 9786, pp. 97862I-1–97862I-7

A. Badoual, D. Schmitter, M. Unser

Locally Refinable Parametric Snakes

Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP'15), Québec QC, Canada, September 27-30, 2015, paper no. TEC-P21.2

Conference Abstracts

A. Badoual, A. Galan, D. Sage, M. Unser

Segmentation of Adjacent Cells in a C. Elegans Embryo Using a Multi-Compartment Active Contour

2019 Quantitative Bioimaging Conference (QBI'19), Rennes, French Republic, January 9-11, 2019, in press

A. Badoual, M. Unser

Subdivision-Based Active Contours

SIAM Conference on Imaging Science (SIS'18), Bologna, Italian Republic, June 5-8, 2018, session MS47-2

Teaching

Teaching Assistant

2015 - 2019 **Signals and Systems I & II**

Supervision of Student Projects

- | | |
|------|--|
| 2017 | Active Contour for Jointly Segmentation of Multiple Cells |
| 2016 | Texture Snake for Tumor Grading |
| 2016 | Data Augmentation for Deep Learning |
| 2015 | Segmentation of Doppler Ultrasound Images for Monitoring the Blood Flow |
| 2015 | Physics-Based Active Contour Energy to Prevent Twisting of Parametric Spline Snakes |