Parametric Shape Processing in Biomedical Imaging

Mathews Jacob

Thèse N° 2857 (2003)

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

Prof. Theo Lasser, *président*
Prof. Michael Unser, *directeur de thèse*
Prof. Michel Barlaud, *co-rapporteur*
Prof. John Maddocks, *co-rapporteur*
Prof. Gabor Szekeley, *co-rapporteur*
Prof. Martin Vetterli, *co-rapporteur*

École polytechnique fédérale de Lausanne—2003

Cover design by Annette Unser
Cover printed by Copy-Tunnel
Printing and binding by Repro-EPFL
Typeset with LaTeX
Copyright © 2003 by Mathews Jacob
Available at `http://bigwww.epfl.ch/jacob/`

Abstract

In this thesis, we present a coherent and consistent approach for the estimation of shape and shape attributes from noisy images. As compared to the traditional sequential approach, our scheme is centered on a shape model which drives the feature extraction, shape optimization, and the attribute evaluation modules.

In the first section, we deal with the detection of image features that guide the shape-extraction process. We propose a general approach for the design of 2-D feature detectors from a class of steerable functions, based on the optimization of a Canny-like criterion. As compared to previous computational designs, our approach is truly 2-D and yields more orientation selective detectors.

We then address the estimation of the global shape from an image. Specifically, we propose to use cubic-spline-based parametric active contour models to solve two shape-extraction problems: (i) the segmentation of closed objects and (ii) the 3-D reconstruction of DNA filaments from their stereo cryo-electron micrographs. We present several enhancements of existing snake algorithms for segmentation. For the detection of 3-D DNA filaments from their orthogonal projections, we introduce the concept of projection-steerable matched filtering. We then use a 3-D snake algorithm to reconstruct the shape.

Next, we analyze the efficiency of curve representations using refinable basis functions for the description of shape boundaries. We derive an exact expression for the error when we approximate a periodic signal in a scaling-function basis. Finally, we present a method for the exact computation of the area moments of such shapes.

Résumé

Dans cette thèse, nous présentons une approche logique et cohérente pour l'évaluation de la forme et des attributs de forme à partir d'images bruitées. Par rapport à l'approche séquentielle traditionnelle, notre approche est basée sur un modèle de la forme qui pilote l'extraction de primitives, l'optimisation des formes et les modules d'évaluation d'attributs.

Dans la première section, nous nous intéréssons à la détection des primitives d'image qui guident le processus d'extraction des formes. Nous proposons une approche générale—basée sur l'optimisation d'un critère de type Canny—pour la conception de détecteurs de primitives 2–D à partir d'une classe de fonctions orientables. Comparée aux méthodes de calcul précédents, notre approche est véritablement 2–D et engendre des détecteurs dont l'orientation est plus sélective.

Nous abordons ensuite l'évaluation de la forme globale à partir d'une image en utilisant les primitives préalablement détectées. Spécifiquement, nous proposons d'employer les modèles actifs paramétriques basés sur les splines cubiques pour résoudre deux problèmes d'extraction de forme : (i) la segmentation d'objets fermés et (ii) la reconstruction 3-D de filaments d'ADN à partir de leurs micrographes stéréos obtenus par microscopie cryo-électronique. Nous proposons plusieurs améliorations aux algorithmes de snakes pour la segmentation déjà existants. Pour la détection des filaments d'ADN 3–D à partir de leurs projections orthogonales, nous introduisons le concept de filtrage adapté à projections orientables. Nous employons alors un algorithme de snake 3-D pour reconstruire la forme.

Ensuite, nous analysons l'efficacité des représentations de courbes en utilisant des fonctions de base raffinables pour la description des bords des formes. Nous dérivons une expression exacte pour l'erreur quand nous approximons un signal périodique dans une base de fonctions d'échelle. En conclusion, nous présentons une méthode pour le calcul exact des moments des zones définies

par de telles formes.

To my parents and grandparents.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to everyone who generously gave me their support.

On the first place, I would like to express my gratitude and appreciation to my thesis advisor Prof. Michael Unser. I found his creativity and enthusiasm extremely motivating. The discussions with him have not only contributed significantly to this work, but also have made lasting influences in me.

I was fortunate to share my office with Thierry Blu during my initial days, from whom I have learned quite a lot over the last 4 years. Although his mathematical skills often left me breathless, it was exciting to watch him perform. I also admire the selfless attitudes of Daniel Sage and Philippe Thévenaz. They were always ready to advice and help.

I really appreciate the friendly environment at BIG. Specifically, I would like to thank my office mate Michael Liebling for the support he gave me during all these years. It was wonderful to be with encircled by so many nice people like Slavica, Jan, Arrate Michael Suehling and Shai.

My special thanks to all my friends in Lausanne, specifically Arunan and Anil. They made my time here very lively. I will definitely miss the discussions and the fun.

I would like to dedicate this work to my parents and grandparents. Their love, constant support and encouragement made me what I am.

Contents

Abstract **i**

Résumé **iii**

Acknowledgements **vii**

Contents **ix**

1 Introduction **1**
1.1 Shape 2
1.2 Shape processing 4
1.3 Curve representation 6
1.3.1 Discrete approaches 7
1.3.2 Continuous representations 8
1.3.3 Level set curve description 8
1.3.4 Framework of our research work 9
1.4 Organization of this thesis 9

2 Parametric signal and curve representation **11**
2.1 Representation of continuous signals 11
2.1.1 Shift-invariant representation 12
2.1.2 Scaling function representation 13
2.2 Sampling of continuous signals 13
2.3 Examples of scaling function representations 16
2.3.1 B-spline basis 16
2.3.2 Bandlimited representation using sinc scaling functions . 21
2.4 Parametric description of closed curves 22

3 Optimal steerable filters for feature detection **25**
3.1 Introduction 25
3.2 Orientation independent matched filtering 27
3.2.1 Detection by rotating matched filtering 27
3.2.2 Steerable filters 27
3.2.3 Conventional detectors revisited 29
3.3 Design of steerable filters for feature detection 31
3.3.1 Optimality criterion 32
3.3.2 Derivation of the optimal detector 34
3.3.3 Feature detection by local optimization 36
3.4 2-D feature detectors 36
3.4.1 Edge detection 37
3.4.2 Ridge detection 42
3.5 Shape adaptable feature detection 45
3.5.1 Derivation of the wedge detector 47
3.5.2 Implementation 49
3.5.3 Results 49
3.6 Summary 50

4 Efficient energies and algorithms for parametric snakes **55**
4.1 Introduction 55
4.2 Mathematical Preliminaries 57
4.2.1 Parametric representation of closed curves 57
4.2.2 Active contour models: formulation 58
4.3 Image energy 58
4.3.1 Edge-based image energy 59
4.3.2 Region-based image energy 62
4.3.3 Unified image energy 65
4.4 Internal Energy 66
4.4.1 Curvilinear reparametrization energy 67
4.4.2 Choice of the scaling basis function 68
4.4.3 New internal energy term 68
4.5 External constraint energy 68
4.6 Evaluation of the partial derivatives 69
4.6.1 Partial derivatives of the internal energy 72
4.6.2 Partial derivatives of the constraint energy 73
4.6.3 Estimation of the probability distribution functions 74
4.6.4 Computation of the length and area 74
4.7 Evolving the curve 75

4.7.1 Optimization Algorithm 75
4.7.2 Loop detection and recovery 75
4.7.3 Shrinking/growing snakes 77
4.8 Discussion and Summary 77

5 3-D shape estimation of DNA molecules 85
5.1 Introduction 85
5.2 Mathematical Preliminaries 88
5.2.1 Parametric representation of 3-D curve 88
5.2.2 Orthogonal volume projection 90
5.3 Local filament detection 91
5.3.1 Projection based feature detection 91
5.3.2 Projection-steerable ridge detection 92
5.3.3 3-D ridge detection 94
5.4 Constrained reconstruction using the 3-D snake model 96
5.4.1 Active contour algorithm: Formulation 97
5.4.2 Image Energy 98
5.4.3 Internal Energy 100
5.4.4 External constraint energy. 101
5.5 Curve evolution: the optimization algorithm 102
5.5.1 Partial derivatives of the image energy 103
5.5.2 Partial derivatives of the internal energy 105
5.5.3 Partial derivatives of the constraint energy 106
5.6 Experiments 108
5.7 Synopsis 109

6 Sampling of Periodic Signals: A Quantitative Error Analysis 115
6.1 Introduction 115
6.2 Preliminaries 117
6.2.1 Notations 117
6.2.2 Sampling of Periodic Signals 117
6.3 Fourier Series Representation 119
6.4 Computation of the Square Error 119
6.5 Asymptotic Performance 121
6.6 Experimental verification of the error formula 123
6.7 Conclusion 124

7 An exact algorithm for computing the area moments of spline curves **133**
7.1 Introduction 133
7.2 Preliminaries 135
7.2.1 Computation of Moments using Green's Theorem 135
7.2.2 Parametric Representation of a curve 135
7.2.3 Differentiation of scaling functions 137
7.3 Moment computation 138
7.3.1 Computation of the Area 138
7.3.2 General Formula 140
7.3.3 Properties of the kernel - f 141
7.3.4 Examples with Splines 143
7.4 Implementation. 144
7.5 Computation of the Kernel 145
7.5.1 Exact Method 145
7.5.2 Approximate Method for Splines 147
7.6 Computation of the Area Moments using Riemann sums 148
7.6.1 Sinc Representation of the curve 148
7.6.2 Spline Representation of the curve 150
7.7 Experiments and results. 151
7.8 Synopsis 154

8 Conclusion **157**
8.1 Main Contributions 157
8.2 Future work 159

Bibliography **161**

Chapter 1

Introduction

The sense of vision is fundamental to our perception of the world. The shape, size, color, and texture of an object are the visual attributes that differentiate it from its surroundings. Of these, the shape and size are perhaps the most important attributes.

The concept of shape is crucial in many areas of science. For example, it plays an important role in biochemistry where it is the shape of a molecule that determines for the most part its ability to interact with other molecules. It is widely used in evolutionary biology where the distinctions within a particular species are often quantified in terms of shape; a systematic study of these shape changes is called morphometry[1, 2]. Over the years, biologists have investigated shape changes during the embryonic development of an organism, bacterial colonies when subjected to different stimuli, etc. [3]. The quantification of shape and shape changes (shape analysis) play an equally important role in understanding the dynamics of cellular organells, growth of cells, etc. It is also a key tool in medicine for understanding various body functions and for detecting abnormalities. Traditionally, shape analysis is performed by visual inspection or by direct measurement of some geometrical properties of the object such as its length, area, angles, etc.

The study of shape in the biomedical sciences is often performed using sophisticated imaging techniques. Microscopy, for example, enable us to see minute objects beyond the resolution limits of the eye, while x-ray imaging let us non-invasively observe the internal structure of objects. Recording techniques such as photography and video make the storage and transmission of visual information possible, thus breaking the spatial and temporal constraints. Many of

these modern imaging techniques require sophisticated signal processing tools. Thanks to the increasing power of digital computers, it is also possible to perform complex mathematical transformations on the visual information. There is a strong interest in computer-aided shape analysis from image data because it is more precise, faster and more reproducible than manual approaches.

In this thesis, we focus our attention on the processing of shapes in the broad area of biological and medical imaging. Computer-aided shape analysis is now an accepted tool in clinical diagnosis. Inspite of extensive research in this area, automatic shape processing still remains a challenging problem—based on our current knowledge, it seems unlikely that there will ever be a general solution that fits all applications. Thanks to the recent technological advances in high-resolution approaches, imaging is also emerging as a key tool for understanding various biomolecular processes. The biological constraints present several new challenges for shape extraction and processing. In biology, one is often forced to push the instrument to its limits, which results in a very low signal-to-noise ratio and reduced image contrast. The challenge is to create robust algorithms that can efficiently extract the valuable information from the available image data. In this context, we present a coherent model-based framework for the processing of shapes. Specifically, we focus on the robust extraction of image features, the estimation of shapes, and the derivation of shape features.

1.1 Shape

The word "shape" is widely used in common language to characterize the appearance of an object. Usually, our perception of the shape of an object is independent of its exact location, orientation and size. Hence an intuitive definition of shape may be given as follows [4, 5].

Definition 1 *The shape of an object is the geometrical information that remains when location, scale and rotational effects are filtered out from it.*

In mathematical terms, the shape information consists of the geometrical features of an object that are invariant to Euclidean similarity transformations. Two objects have the same shape if one can be mapped onto the other by scaling, rotation and translation.

The spatial description of an object may be provided in many equivalent ways. The most popular representations are:

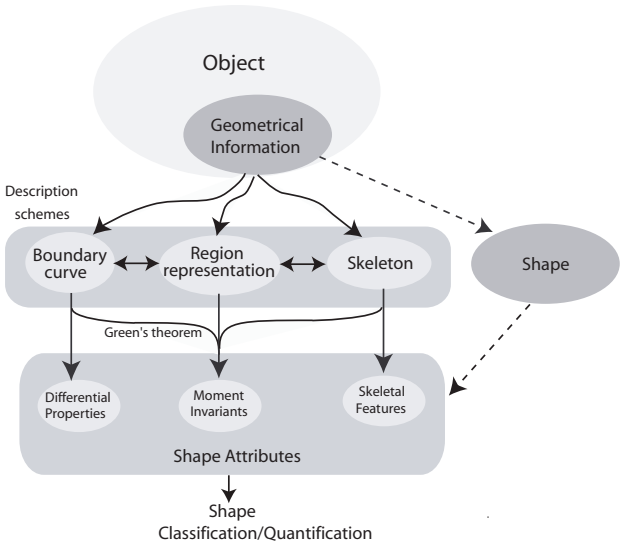


Figure 1.1: Overview

1. A region-based scheme where the shape is represented by the region Ω, as shown in Fig 1.2. Here, each pixel of the image has a binary label that indicates whether or not it belongs to the object. This approach has the advantage of being able to handle complex topologies naturally. Its downside is that the representation is not concise at all. Also, it is best adapted to the description of objects on a discrete grid as opposed to a continuum.

2. An explicit boundary representation [6, 7, 8, 9, 10] where the boundary $d\Omega$ (curve for planar object and surface for 3-D) defines the shape of the object. Note that the boundary representation can be obtained from the region description and vice versa; the two are duals.

3. A skeleton representation [11]. The skeleton of an object is the locus of all points in the shape that do not have a unique nearest boundary point upon the shape (c.f. Fig 1.2). The complete representation of the shape requires the skeleton as well as the width function. The width function at a specified point on the skeleton is defined as the distance to any of the set of equidistant boundary points. This representation is ideal for the

representation of wiggly shapes whose skeleton may be deformed without the width function; e.g a worm.

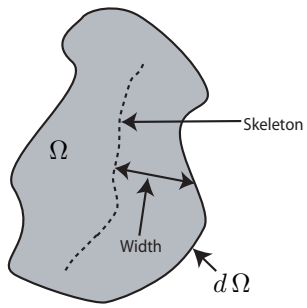


Figure 1.2: Equivalent Shape representations

Once the spatial description of an object is obtained, it may be used to extract some global shape attributes, with a preference being given to attributes that are invariant to similarity transforms. Examples of shape attributes include moment invariants[12], curvature scale-space of the bounding curve [13], etc. The shape attributes are then typically used for classification, for quantifying shape variations. See Fig. 1.1.

Here, we focus on explicit boundary representations due to their efficiency. In particular, they need much fewer degrees of freedom than region-based approaches and they are much simpler to implement and process than skeleton-based techniques. The various types of digital contour representations are discussed in Section 1.3. A clear advantage of having such an explicit description is that it is quite straightforward to derive curve-dependent shape attributes such as curvature or other differential geometric features. Thanks to Green's theorem, the area dependent shape features (e.g. area moments) can also be extracted efficiently from it.

1.2 Shape processing

The real-world objects are available to us as digital images; the shape attributes have to be estimated for classification, quantification or identification. The

standard approach for the parameter estimation is sequential (c.f. Fig 1.3). It consists of the following steps:

1. Feature extraction: features such as edges, ridges, corners are extracted from the image using appropriate operators. The algorithm gives a measure of the likeliness of the feature and its orientation at every pixel in the image; the derivation of the likeliness measure is performed independently for every pixel.

2. Shape estimation/segmentation: The geometric information of the object is estimated from the detected features using an appropriate segmentation algorithm; e.g. active contour models, edge-connecting algorithms, etc.

3. Evaluation of shape attributes: The detailed shape description is reduced to a more global representation in terms of a few shape attributes that may be derived from it.

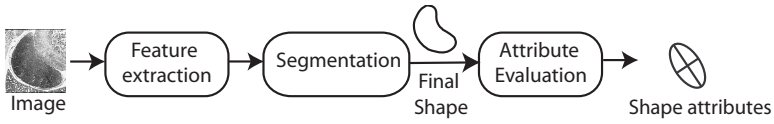


Figure 1.3: Estimation of shape attributes from images

This sequential approach is simple because it involves standard modules that can be combined very easily. It has been quite successful in a number of applications including computer vision and multi-media, where the images are relatively noise-free. In this work, however, we are dealing with biomedical images that are typically blurred and quite noisy. Hence, a more robust algorithm for shape recovery is highly desirable. This motivates us to investigate a more global and consistent model-based approach where the various modules are linked together as shown in Fig. 1.4. A shape model is central to all the steps of the algorithm; it drives both the feature extraction and the shape extraction algorithms. The shape attributes are also evaluated from the same model.

We introduce the concept of a steerable feature space, which provides an effective mean of computing a measure of likeliness of having a particular feature (e.g. edge, ridge, corner etc) at a specified location and orientation. The precomputation of a steerable map of elementary features makes it practically feasible to couple the feature extraction and the shape estimation algorithms.

The shape optimization algorithm starts with an initial guess that is refined iteratively based on the confidence measure provided by the feature estimation algorithm. The feature estimation stage takes in the current shape and computes a figure of merit; typically a weighted sum of likeliness of desired features along the boundary. We may use different features at different curve points depending on the local curve properties. For example, in contour regions with a high curvature, it may be more appropriate to use a corner detector rather than an edge detector. This concept is explained in detail in Chapter 5.

Note that, unlike classical schemes where the feature estimation at each pixel is performed independently, this approach performs a joint estimation of the features. Since the coefficients of the curve model are typically much fewer than the number of pixels through which the curve passes, we expect this approach to be more robust to noise. It is also more consistent than the traditional two-step approach: for instance, the local orientation that is provided by a classical edge detector (e.g. Canny's operator) may be different from the orientation of the tangent vector of the estimated contour line that is the result of the segmentation.

In the classical approaches, shape features such as moment-invariants are estimated using a discrete approximation. The estimated shape is rasterized to label the regions inside and outside of it, from which the moments are computed numerically. We use Green's theorem to compute these directly and from the contour model; note that this approach is mathematically exact and also numerically efficient.

In many biomedical problems the average shape of the object is known, which can be used to constrain the reconstruction process. Thanks to the efficient algorithms for the moment computation, this constraint can be added at relatively low cost. The user can also provide other constraints to manually aid the algorithm.

1.3 Curve representation

We have seen that the use of a curve model is central to our shape estimation algorithm. All the steps of the algorithm can profit from a good curve representation scheme. Ideally, we would like to have a curve model that can represent the shape well with the fewest coefficients as possible. Such a model will lead to a robust estimation algorithm at a low computational cost.

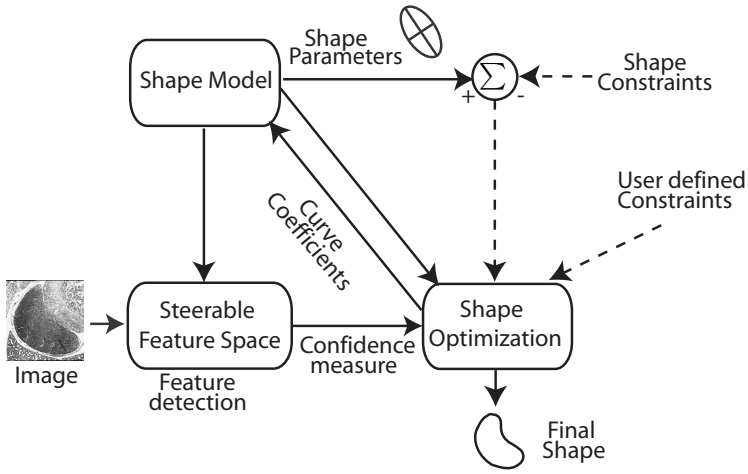


Figure 1.4: Our approach

1.3.1 Discrete approaches

The simplest representation of a discrete curve is an ordered collection of points. However, this approach does not ensure smoothness nor even continuity of the contour. In shape estimation algorithms that use this type of representation, the smoothness is often ensured by introducing extra constraints [8]. Another scheme that is popular is the Freeman code[7]. There, the discrete curve can only jump from the current position to one of its eight neighboring pixels. The curve is represented by a chain of integers from zero to seven, coding for the orientation of the current segment. The starting point is usually specified separately. Note that this approach results in a curve that is continuous which is not necessarily the case with the first approach.

In addition to their lack of implicit smoothness, discrete representations usually require many parameters to encode even a simple shape. The large number of parameters to be estimated can therefore impair the robustness of

the shape recovery algorithms. This may also result in a high computational complexity when sophisticated cost functions are being used.

1.3.2 Continuous representations

Most of the continuous approaches enjoy implicit continuity and smoothness. If the model is appropriate, they can yield sub-pixel accurate segmentation. Moreover, when compared with the discrete approaches, they usually require fewer coefficients. There are many different techniques for representing continuous curves. For a complete review, refer to [14].

The most popular among these approaches are the parametric curve representations. In this scheme, the curve is represented in terms of an arbitrary parameter t. The component functions $(x(t), y(t))$ are represented as a linear combination of some basis functions. In computer graphics, curves are often represented using non-uniform or uniform B-spline functions [15], and more recently NURBS[1]. NURBS, which are a generalization of Bezier curves, is the preferred approach in computer graphics since these functions are closed under perspective transformations (needed in computer graphics) and can represent conic sections exactly [16]. On the other hand, curve descriptions based on Fourier exponentials [6, 10] and uniform B-spline functions [9, 17] are popular in image processing and computer vision. The popularity of these approaches are due to the existence of efficient signal processing algorithms and their invariance to similarity transformations. Of these, the B-spline curves have the extra advantage of locality of control; a change in one of the knot points will only affect a small region of the curve. We discuss the parametric representation of curves in detail in the next chapter.

1.3.3 Level set curve description

A recent trend is to represent the curve as a level set of an an appropriate potential function [18, 19, 20, 21]. This implicit scheme tries to preserve the advantages of region-based representations; it can naturally handle shapes of complex topologies. In recent years, this research area has undergone an extensive development. One downside of the level-set approach is its computational complexity. This is because, during the segmentation process, one is evolving a surface rather than a curve. It is also not straightforward to introduce shape constraints.

[1] Non-Uniform Rational B-Spline.

1.3.4 Framework of our research work

In this thesis, we concentrate on parametric representations due to their simplicity and computational efficiency. In addition, we restrict ourself to simple shapes that are topologically equivalent to a circle. In other words, these are entirely specified by a single closed curve that defines their outer boundary.

1.4 Organization of this thesis

Following the global introduction that has just been made, the thesis proceeds with a review in Chapter 2 of mathematical concepts that are used extensively throughout the work. Special attention is given to the parametric representation of curves and signals in a basis composed of integer shifts of a generating function.

The subsequent part (Chapters 3-7) present the scientific contributions of the research. All the chapters correspond to work that has been published (or is currently under review) in peer-reviewed journals.

In Chapter 3, we concentrate on the detection of image features to guide the shape extraction process. We propose a general approach for the design of 2-D feature detectors from a class of steerable functions based on the optimization of a Canny-like criterion. In contrast with previous computational designs, our algorithm is truly 2-D and provides filters that have closed form expressions. It also gives operators that are more orientation selective than the classical gradient or Hessian-based detectors.

We then address the estimation of the global shape from an image using the detected features. Specifically, we use cubic spline-based parametric active contour models to address two shape extraction problems: (i) the segmentation of closed objects in Chapter 4, and, (ii) the 3-D reconstruction of DNA filaments from their stereo cryo-electron micrographs in Chapter 5. In both approaches, we profit from the optimality properties of cubic B-spline curves and efficient B-spline algorithms.

We present several enhancements over the classical parametric active contour algorithm for the segmentation of closed regions. We introduce a new edge-based energy that overcomes the shortcomings of the conventional one. We re-express this energy as a surface integral, thus unifying it naturally with the region-based schemes. We show that parametric snakes can guarantee low curvature curves, but only if they have a constant arc-length. Hence, we propose a new internal energy term to enforce this configuration.

For the detection of 3-D DNA filaments from their orthogonal projections, we introduce the concept of projection-steerable matched filtering. We design a 3-D template such that its orthogonal projections onto the image planes are steerable, i.e., the projections can be expressed as linear combinations of a few 2-D basis functions, for any orientation of the template. We use a 3-D active contour algorithm for the shape estimation. The feature detection algorithm returns a confidence measure by integrating the likeliness measures of the 3-D ridges along the contour; the likeliness measures are computed efficiently from the 2-D steerable feature space.

In Chapter 6, we analyze the efficiency of scaling function curves for the representaton of shapes. We derive an exact expression for the error when we approximate a periodic signal in a scaling function basis. The formula takes the simple form of a Parseval's like relation, where the Fourier coefficients of the signal are weighted against a frequency kernel that characterizes the approximation operator.

Finally in Chapter 7, we present a method for the exact computation of the moments of a region bounded by a curve represented by a scaling function or wavelet basis. Using Green's Theorem, we show that the computation of area moments is equivalent to applying a suitable multidimensional filter on the curve coefficients and thereafter computing a scalar product. The multidimensional filter coefficients are pre-computed exactly as the solution of a two-scale relation.

Chapter 2

Parametric signal and curve representation

In this chapter, we briefly review some mathematical concepts that will be used extensively in this thesis. We consider a number of popular signal representation schemes and show how these approaches can be used for the compact representation of parametric closed curves in the plane (e.g. shapes).

2.1 Representation of continuous signals

The description of a general signal $f(x) \in L_2(\mathbb{R})$, requires the knowledge of its values for every point x; it is not suitable for digital transmission or storage. Since most signals of practical interest have a finite rate of information, it is a general practice to assume them to be in some well defined vector space. A classical example is the representation of a signal in a bandlimited space in terms of sinc functions. In this case, only the coefficients of the basis functions at the sampling locations need to be stored. The choice of the vector space is crucial since it determines the quality of the representation and the time taken for the computations. We now review the signal models that are relevant for our purpose.

2.1.1 Shift-invariant representation

One popular approach is to describe the signal in a shift-invariant basis. Here, the bases are generated by the integer shifts of a single function. An arbitrary signal in this class is given by

$$f(x) = \sum_{k=-\infty}^{\infty} c(k)\,\varphi(x-k)\,. \tag{2.1}$$

φ is called the generating function [22, 23] and $c(k)$ are the coefficients. If $\varphi(k) = \delta_k$ (the Kroneker delta function), we have $c(k) = f(k)$; the coefficients are the signal samples themselves. Such a φ is called an interpolating function. An simple example is shown in Fig. 2.1.

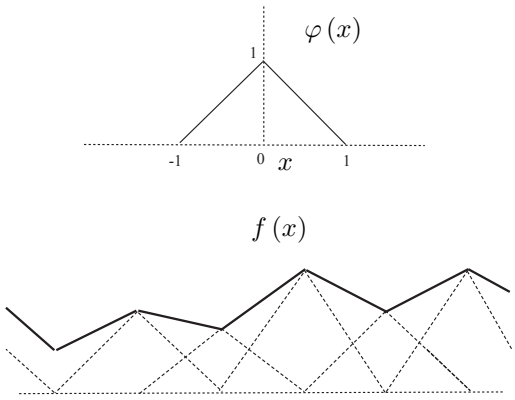


Figure 2.1: Shift invariant representation of a function $f(x)$. The dotted functions are the basis functions that are obtained by the integer shifts of φ. Since $\varphi(k) = \delta(k)$, the coefficients are the samples of $f(x)$ at $x = k$.

The representation is stable[1] and unambiguous if $\varphi(x)$ generates a Riesz basis of $V(\varphi) = \text{span}\{\varphi(x-k)\,; k \in \mathbb{Z}\}$; i.e., there must exist two strictly positive

[1] By stable, we mean that a small variation of the $c(k)$'s should result in a small variation of the function

constants A and B such that

$$\forall c(k) \in l_2,\ A \cdot \|c\|_{l_2}^2 \leq \left\| \sum_{k \in \mathbb{Z}} c(k)\, \varphi(x-k) \right\|_{L_2}^2 \leq B \cdot \|c\|_{l_2}^2, \tag{2.2}$$

where $\|c\|_{l_2}^2 = \sum_k |c(k)|^2$. In other words, we have an equivalence between the discrete and continuous norms. The L_2 norm $\|f\|_{L_2}^2 = \langle f, f \rangle_{L_2}$ is derived from the standard L_2 inner-product

$$\langle f, g \rangle_{L_2} = \int_{-\infty}^{\infty} f(x)\, g(x)\, dx \tag{2.3}$$

A sequence $c(k) \in l_2$ implies that it is square summable.

2.1.2 Scaling function representation

A generating function that satisfies the two-scale relation

$$\varphi\left(\frac{x}{2}\right) = \sum_{k=0}^{\infty} h(k)\, \varphi(x-k), \tag{2.4}$$

is called as a scaling function [22]. $h(k) \leftrightarrow \hat{h}(z)$ is the two-scale mask or refinement filter of φ. The above equation implies that the function $\varphi\left(\frac{x}{2}\right)$ as well as any of its integer shifs $\varphi\left(\frac{x}{2} - k\right); k \in \mathbb{Z}$ is contained in the space $V(\varphi) = V_1(\varphi)$, where

$$V_T(\varphi) = \operatorname{span}\left\{\varphi\left(\frac{x}{T} - k\right); k \in \mathbb{Z}\right\} \tag{2.5}$$

This, inturn, implies that $V_{2^{j+1}}(\varphi) \subseteq V_{2^j}(\varphi)$. The nested nature of the subspaces enables us to represent a signal at different resolutions (multi-resolution representation), which may be useful in various image processing algorithms[24]. This class of representations is sufficiently general to accommodate the widely used signal descriptions such as bandlimited (using the sinc basis function), B-spline and the wavelet representations.

2.2 Sampling of continuous signals

The continuous domain input signal $f(x)$ is not usually known directly. Rather, it is specified in terms of its uniform measurements

$$g_T(k) = \left\langle f(x), \tilde{\varphi}\left(\frac{x}{T} - k\right) \right\rangle_{L_2}, \tag{2.6}$$

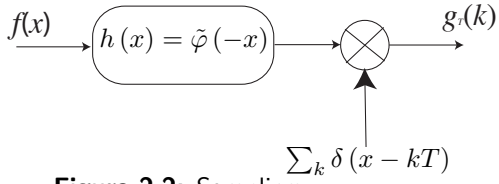


Figure 2.2: Sampling.

where $\tilde{\varphi}$ is an appropriate model for the measurement device (cf. Fig 2.2). For example, when $\varphi(x) = \delta(x)$—the Dirac's delta distribution— the measurements are uniform samples of f. In the context of the classical bandlimited sampling, $h(x) = \tilde{\varphi}(-x) = \operatorname{sinc}(x)$ is the ideal low-pass anti-aliasing filter.

Consistent sampling

Since the reconstruction of a general $f(x) \in L_2(\mathbb{R})$ from its uniform measurements is ill-posed, it is a general practice to reconstruct in a subspace of $L_2(\mathbb{R})$. The choice of the subspace dictates the quality of the reconstruction. One popular approach is to choose an $f_T(x) \in V_T(\varphi)$ which give the same measurements $g_T(k)$, if re-injected into the measurement system [25]. This approach is called consistent reconstruction. The measurement function $\tilde{\varphi}$ is called the analysis function and φ the synthesis function. The above scheme gives perfect reconstruction for signals in $V_T(\varphi)$ [25].

An arbitrary signal in $V_T(\varphi)$ is given by$f_T(x) = \sum c(k)\,\varphi\left(\frac{x}{T} - k\right)$. We have to choose $c(k)\,;\, k \in \mathbb{Z}$ such that:

$$\sum_k c(k) \underbrace{\left\langle \varphi\left(\frac{x}{T} - k\right), \tilde{\varphi}\left(\frac{x}{T} - l\right)\right\rangle}_{a_{\varphi,\tilde{\varphi}}(k-l)} = g_T(l)\,;\ \forall l \in \mathbb{Z} \tag{2.7}$$

This implies that the sequence $c(k)$ can be obtained from the measurements as (assuming that $\hat{a}_{\varphi,\tilde{\varphi}}(z)$ do not vanish on the unit circle)

$$\hat{c}(z) = \underbrace{\left(1/\hat{a}_{\varphi,\tilde{\varphi}}\left(z^{-1}\right)\right)}_{\hat{q}(z)} \hat{g}_T(z)\,, \tag{2.8}$$

where $\hat{c}(z)$ is the Z-transform of $c(k)$. The above expression implies that $g_T(k) = c(k)$ iff $a_{\varphi,\tilde{\varphi}}(k) = \delta_k$; if this condition holds then φ and $\tilde{\varphi}$ are biorthogonal to each other. The general reconstruction procedure is shown in

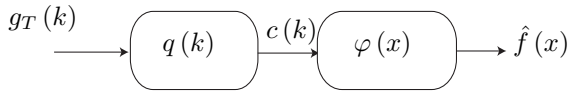


Figure 2.3: Reconstruction.

Fig. 2.3. In this case, there exists an equivalent generating function

$$\varphi_1(x) = \sum q(k)\,\varphi(x-k) \tag{2.9}$$

that is bi-orthogonal to $\tilde{\varphi}$. Thus $f_T(x)$ is an oblique projection of f onto $V_T(\varphi)$.

Projection error

The quality of the reconstruction is decided by the signal space, the measurement system, and, the reconstruction space. The exact expressions for the shift invariant error (the average error over all shifts of f) is derived in the Fourier domain as [26]

$$\epsilon_f^2(T) \triangleq \frac{1}{T}\int_0^T \|f(\cdot-\tau) - f_T(\cdot-\tau)\|^2\,d\tau \tag{2.10}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} E_{\varphi,\tilde{\varphi}}(T\omega)\left|\hat{f}(\omega)\right|^2 d\omega, \tag{2.11}$$

where the error kernel $E_{\varphi,\tilde{\varphi}}(\omega)$ is

$$E_{\varphi,\tilde{\varphi}}(\omega) = \left|1-\hat{\tilde{\varphi}}(\omega)\,\hat{\varphi}(\omega)\right|^2 + \left|\hat{\tilde{\varphi}}(\omega)\right|^2 \sum_{k\neq 0}\left|\hat{\varphi}(\omega+2k\pi)\right|^2 \tag{2.12}$$

$$= \underbrace{1-\frac{\left|\hat{\varphi}(\omega)\right|^2}{\hat{a}_{\varphi,\varphi}(\omega)}}_{E_{\min}(\omega)} + \underbrace{\hat{a}_{\varphi,\varphi}(\omega)\left|\hat{\tilde{\varphi}}(\omega)-\hat{\varphi}_d(\omega)\right|^2}_{E_{\mathrm{res}}(\omega)}, \tag{2.13}$$

where $\hat{\varphi}_d(\omega) = \hat{\varphi}(\omega)/\hat{a}_{\varphi,\varphi}(\omega)$ is the dual of φ. Note that we obtain the minimum possible error when $\tilde{\varphi} = \varphi_d$.

The approximation error tend to zero as the sampling step $T \to 0$, iff $E_{\varphi,\tilde{\varphi}}(0) = 0$; this implies the partition of unity constraint on the scaling function:

$$\sum_k \varphi(x-k) = 1 \tag{2.14}$$

This condition ensures that the constant is indeed in the space $V(\varphi)$. The asymptotic decay of the approximation error (behavior as the $T \to 0$) is often used to compare representation schemes. If $E_{\varphi,\tilde{\varphi}}(\omega) = C^2_{\varphi,\tilde{\varphi}}\omega^{2L} + \mathcal{O}\left(\omega^{2L+2}\right)$ (this implies that φ satisfies the Strang-Fix conditions of order L [22]), then

$$\|f - f_T\| \leq C_{\varphi,\tilde{\varphi}} \cdot T^L \|f^{(L)}\|_{L_2} \tag{2.15}$$

Here, $C_{\varphi,\tilde{\varphi}}$ is a known constant and $f^{(L)}$ is the Lth derivative of f. A φ that give such an error decay is called as an Lth order generating function. The constant $C_{\varphi,\tilde{\varphi}}$ is dependent on the analysis and the synthesis functions.

2.3 Examples of scaling function representations

2.3.1 B-spline basis

The B-spline basis functions possess several interesting properties that make them attractive for signal representation. A B-spline function of degree n is defined as the $(n+1)$-fold convolution of the rectangular function:

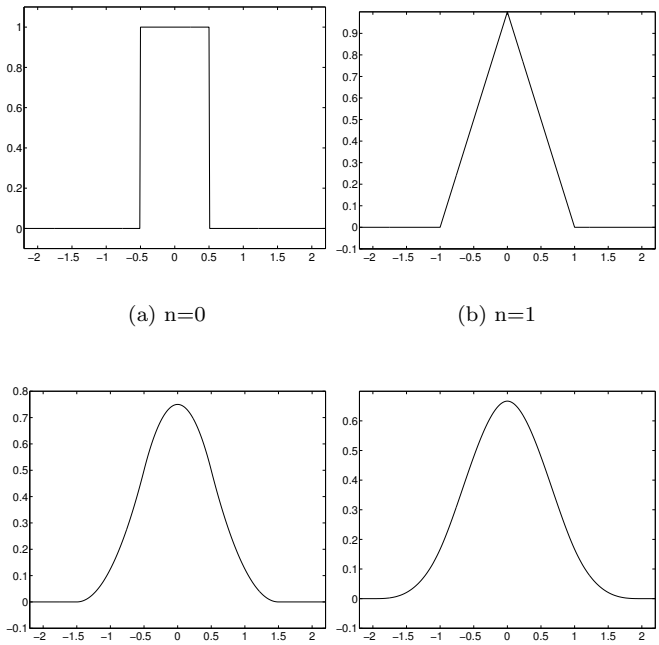
$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

$$\beta^m(x) = \underbrace{\beta^0 * \beta^0 \ldots * \beta^0}_{(m+1)\text{ times}} \tag{2.17}$$

The B-spline functions of degrees 0 to 3 are shown in Fig 2.4. Note that these functions are are all non-negative. They are interpolating only for $n = 0$ and $n = 1$. We now discuss various remarkable properties of B-spline functions that make them attractive for signal processing.

Approximation Properties

The B-splines of degree n have an approximation order $L = n + 1$. They are the smoothest and the shortest scaling functions of order L. This makes them



(a) n=0 (b) n=1

(c) n=2 (d) n=3

Figure 2.4: B-spline functions of different degrees.

optimal in the trade-off between computational cost [2] and performance [27].

Variational formulation

The B-spline representation is also optimal in a variational framework. Consider the interpolation problem where we have uniform samples of a signal denoted as $f_k; k \in \mathbb{Z}$. We have seen that this problem can be made well-defined by restricting the reconstructions to a subspace of $L_2(\mathbb{R})$. Another approach is to add a regularization term and solve it as a variational problem [28].

In this formulation, one tries to derive a function $f(x)$ that satisfies the interpolation constraints $f(k) = f_k;\ k \in \mathbb{Z}$ and minimizes a certain regularization

[2] Computational cost is proportional to the length of the generating function.

term which penalizes the norm of the derivative of f given by $\int_{-\infty}^{\infty} \left|\hat{f}^{(m)}(x)\right|^2 dx$. The standard approach to solve this problem is the Lagrange's multiplier's method where the criterion is given by

$$\mathcal{J}(f) = \int_{-\infty}^{\infty} \left|f^{(m)}(x)\right|^2 dx + \sum_{k\in\mathbb{Z}} \lambda_k \left(f(x_k) - f_k\right), \tag{2.18}$$

subject to the constraints $f(x_k) = f_k;\ k \in \mathbb{Z}$. Here, λ_k are the weights and $f^{(m)}(x)$ stands for the mth derivative of f. The x_k where$k \in \mathbb{Z}$ are the sampling locations. We consider a small perturbation of f as $f + \alpha g$ and observe the corresponding change in the criterion[3]

$$\begin{aligned}\mathcal{J}(f+\alpha g) - \mathcal{J}(f) &= \alpha\left(\sum_{k\in\mathbb{Z}} \lambda_k\, g(x_k) + 2\int_{-\infty}^{\infty}\left(f^{(m)}(x)\, g^{(m)}(x)\right) dx\right) \\ &= \alpha\int_{-\infty}^{\infty} g(x)\left(\sum_{k\in\mathbb{Z}} \lambda_k\, \delta(x-x_k) + 2f^{(2m)}(x)\right) dx\end{aligned} \tag{2.19}$$

In the last step, we used integration by parts to transfer the order of differentiation from the compactly supported g to f. If f^* is the solution to the variational problem, it will correspond to the minimum of the criterion. Hence, $\mathcal{J}(f^* + \alpha g) - \mathcal{J}(f^*)$ has to be zero for any g. This implies that

$$2f^{*(2m)}(x) = -\sum_{k\in\mathbb{Z}} \lambda_k\, \delta(x - x_k);\forall x \in \mathbb{R} \tag{2.20}$$

The solution to this differential equation is given by

$$f^*(x) = -\sum_{k\in\mathbb{Z}} \frac{\lambda_k}{2}\,(x-x_k)_+^{2m-1} + \underbrace{c_{2m-2}\, x^{2m-2} + \ldots + c_1\, x + c_0}_{\text{Kernel}}, \tag{2.21}$$

where

$$x_+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{2.22}$$

Now, the optimal weights $\lambda_k; k \in \mathbb{Z}$ have to be chosen so that the interpolation constraints are met.

[3] g is a finitely supported smooth test function

The kernel is a polynomial of order $2m-2$. Now, if we restrict ourselves to the uniform sampling case $x_k = k$, it can be shown that the kernel can be represented in the space

$$\operatorname{span}\left\{(x-k)_+^{2m-1}\,;k\in\mathbb{Z}\right\}.$$

Hence the solution to the variational problem in the uniform setting simplifies to

$$f^*(x) = -\sum_{k\in\mathbb{Z}} \lambda_k (x-k)_+^{2m-1}, \tag{2.23}$$

where the optimal weights are obtained from the interpolation constraints. In fact, the interpolant is a unique function in the space generated by the basis $\left\{(x-k)_+^{2m}\,;k\in\mathbb{Z}\right\}$ that satisfies the constraints.

Note that the basis functions $(x-k)_+^{2m-1}$ are not finitely supported; they grow with respect to increasing x. The derivation of the optimal weights λ_k from the interpolation constraints involves the solution of a linear system of equations of the form $\mathbf{AX} = \mathbf{B}$. The condition number of the matrix $\mathbf{A}$ will be very large since $(x-k)_+^{2m-1}$ are increasing functions. This means that the solution to the linear system will be numerically unstable. Fortunately for us, there exist other basis functions in this space that are finitely supported. Specifically, the function x_+^{2m-1} can be localized using the $(2m)^{\text{th}}$ order finite difference operator denoted by Δ^{2m}, where $\Delta f(x) = f(x+1/2) - f(x-1/2)$.

$$\Delta^{2m}\left(x_+^{2m-1}\right) = \beta^{2m-1}(x) \tag{2.24}$$

Thus we see that the reconstruction of f in a shift-invariant subspace of $L_2(\mathbb{R})$ is optimal in the variational setting, provided the basis functions are shifted B-splines of degree $2m-1$.

The fact that the solution can be expressed as a linear combination of B-splines also makes it obvious that the polynomial kernel in (2.21) is not needed, since it is well known that the uniform B-splines of degree $2m-1$ reproduce the polynomials of degree $2m-1$.

Explicit formula

Since the B-spline function is obtained as a linear combination of one sided power functions as in (2.24), it is piecewise polynomial in nature. For example,

a cubic B-spline function can be expressed as [29].

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^2}{2} & 0 \leq |x| \leq 1 \\ \frac{(2-|x|)^3}{6} & 1 \leq |x| \leq 2 \\ 0 & \text{else} \end{cases} \tag{2.25}$$

Efficient Algorithms

The existence of fast algorithms make them even more attractive.

- Interpolation by digital filtering: In many applications, the signal is known by its uniformly spaced samples. One has to derive the value of the function at non integer samples for many applications like zooming, rotations, image registration etc. In this case we have $\tilde{\varphi}(x) = \delta(x)$; $g_1(k) = f(k)$. Thus $a_{\varphi,\tilde{\varphi}}(k) = \beta^m(k)$. Hence, the B-spline coefficients are obtained as (2.8). Since $\beta^m(k)$ is a symmetric FIR filter, the B-spline filter given by $(\beta^m)^{-1}$ is an all-pole system that can be efficiently implemented using a cascade of causal and anti-causal recursive filters [29, 30].

- Fast zooming: In many applications, one needs a zoomed version of the signal onto a finer uniform grid. In this case the kernel values can be precomputed; zooming can be performed with as little as 3 multiplications/sample for cubic B-splines [29, 27]

- Arbitrary Resizing: In the context of signal resizing by arbitrary size factors (non-integer or non-rational), one can use the efficient least-squares resampling approach [31].

- Computation of derivatives: The derivatives of the B-spline functions have explicit expressions which are very useful in several algorithms [29, 27]. Since the B-spline of degree m is obtained as the $m+1$ fold convolution of the rectangular pulse, its Fourier transform is

$$\hat{\beta}^m(\omega) = \left(\frac{e^{j\omega/2} - e^{j\omega/2}}{j\omega}\right)^{m+1} \tag{2.26}$$

Thus, the Fourier transform of the differential of $\beta^m(x)$ is:

$$\begin{aligned}\mathcal{F}\left(\frac{\partial}{\partial x}\beta^m(x)\right) &= j\omega\left(\frac{e^{j\omega/2}-e^{j\omega/2}}{j\omega}\right)^{m+1} \\ &= \underbrace{\left(e^{j\omega/2}-e^{j\omega/2}\right)}_{\mathcal{F}(\delta(x+1/2)-\delta(x-1/2))}\underbrace{\left(\frac{e^{j\omega/2}-e^{j\omega/2}}{j\omega}\right)^{m}}_{\hat{\beta}^{m-1}(\omega)} \end{aligned} \tag{2.27}$$

From the above expression, we obtain

$$\frac{\partial}{\partial x}\left(\beta^m(x)\right) = \beta^{m-1}\left(x+\frac{1}{2}\right) - \beta^{m-1}\left(x-\frac{1}{2}\right) \tag{2.28}$$

This simple relation enables us to compute the derivative of spline functions at any point very efficiently.

2.3.2 Bandlimited representation using sinc scaling functions

The sinc function is a valid scaling function with a bandlimited two-scale mask. However, the function is not finitely supported in time; a direct implementation of the interpolation algorithm will be very expensive.

If the signal is finitely supported, one can extend it using periodic boundary conditions. A periodic signal $f(t) = f(t+kM)\,;\ k \in \mathbb{Z}$ can be expressed in the sinc basis as

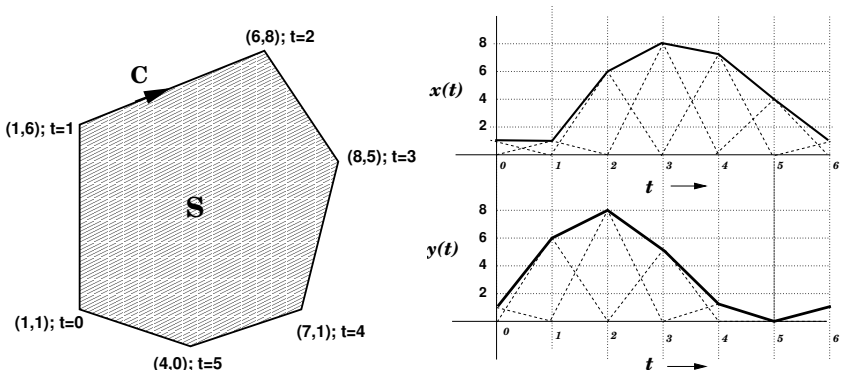
$$f(t) = \sum_{k=0}^{M} c(k)\,\mathrm{sinc}_p(t-k)\,, \tag{2.29}$$

where

$$\mathrm{sinc}_p(t) = \sum_{k=-\infty}^{\infty} \mathrm{sinc}(t-k\,M) \tag{2.30}$$

Due to the slow decay of the sinc function, sinc_p does not converge when M is even. However, when M is odd, it converges to a well defined function in $L_1([0,M])$. In this case, the signal representation can be reformulated as a Fourier series; one can draw upon FFT based techniques to speed up the computations [32]. It involves the computation of its FFT, padding the Fourier samples with zeros and computing the inverse FFT. This operation will give a complex signal if M is even and hence is not equivalent to the sinc interpolation of the real samples. However for odd M, the result is ensured to be real.

This model is not well suited to obtain the function sample for an arbitrary value of t, since it depends on all the coefficients $c_0 \dots c_{M-1}$. Also note that sinc corresponds to a spline interpolator with the degree $n \to \infty$ [33].



(a) Polygon

(b) Parametric representation

Figure 2.5: Scaling function representation of a polygon. The dotted lines in (b) indicate the corresponding linear B-spline basis functions. Note that in this special case the knots are the vertices of the polygon themselves.

2.4 Parametric description of closed curves

The shapes of objects are often represented by their bounding curves. Traditionally, the bounding curve is described as an ordered collection of points [8]. However, this scheme does not give a compact representation. Another approach involves the representation of the object boundary as a level set of a surface. Although this technique can handle complex topologies effectively, the computational complexity of the algorithms that use this representation is quite significant. Moreover, it is not very easy to introduce shape constraints into boundary extraction algorithms. For all these reasons, we prefer to use a parametric representation for describing the shape boundary.

A curve in the 3-D space can be described in terms of an arbitrary parameter t as $\mathbf{r}(t) = (x(t), y(t))$ [6, 9, 10, 17]. When the curve is closed, the function vector

$\mathbf{r}(t)$ is periodic. The component curve functions can be represented in a scaling function basis as

$$\mathbf{r}(t) = \begin{bmatrix} x\,(t) \\ y\,(t) \end{bmatrix} = \sum_{k=-\infty}^{\infty} \mathbf{c}\,(k)\,\varphi(t-k), \tag{2.31}$$

where $\mathbf{c}\,(k) = [c_x\,(k)\,, c_y\,(k)]$ is the coefficient vector; they are often called as knot points. If the period—M—is an integer, we have $\mathbf{c}\,(k) = \mathbf{c}\,(k+M)$. This reduces the infinite summation to

$$\mathbf{r}\,(t) \quad = \quad \sum_{k=0}^{M-1} \mathbf{c}\,(k)\,\varphi_p(t-k), \tag{2.32}$$

where φ_p is the M - periodization of φ:

$$\varphi_p(t) = \sum_{k=-\infty}^{\infty} \varphi(t-k\,M) \tag{2.33}$$

A representation of a planar closed curve $\mathbf{r}(t) = (x(t), y(t))$ in the linear B-spline basis is shown in Fig. 2.5. It also permits a multiresolution representation of the curve[34, 35]. Moreover, the scaling function representation is affine-invariant; an affine transformation of the curve is achieved simply by transforming the coefficient vector $\mathrm{b}_k,\ k = 0, 1, \ldots, M-1$. This is because of the linearity of the representation and the partition of unity condition:

$$\sum_{k=-\infty}^{\infty} \varphi(t-k) = 1, \tag{2.34}$$

which is satisfied by all valid scaling functions in wavelet theory.

We use B-splines for the representation of curves due to their advantages discussed before. This yields spline curves which are frequently used in computer graphics [15] and computer vision [36, 37, 38]. In addition to the advantages mentioned above, the compact support of the functions provide local control of the contour; by changing a coefficient, we change only a small section of the shape. In contrast, if we were dealing with a Fourier series representation, such a change would affect the whole shape. Moreover, the curve rendering can be performed efficiently using the approach used for zooming. Many shape estimation algorithms requires the computation of the curve tangents which can be easily be obtained by using (2.28).

Chapter 3

Optimal steerable filters for feature detection

The first step of our shape estimation algorithm (c.f Fig. 1.4) involves the generation of a steerable feature space from the image. In this chapter[1] we discuss the design of steerable filters for the detection of specific image features. We also deal with the local detection of features based on the steerable feature space pre-computed from the images.

3.1 Introduction

In his seminal paper on computational edge detection, Canny identified the desirable qualities of a feature detector and proposed an appropriate optimality criterion. Based on this criterion, he developed a general approach to derive the optimal detector for specific image features such as edges[39]. This work had a great impact on the field and stimulated further developments in this area, particularly on alternate optimality criteria and design strategies [40, 41].

All the above authors considered the derivation of optimal 1-D operators. For 2-D images, they applied the optimal 1-D operator orthogonal to the feature boundary while smoothing in the perpendicular direction (along the boundary). This extension is equivalent to computing inner-products between the image

[1] Based on the article "M.Jacob and M.Unser, *IEEE Transactions on Pattern Analysis and Machine Intelligence, in press*"

and a series of rotated versions of a 2-D reference template (tensor product of the optimal 1-D profile and the smoothing kernel). With this detector, the rotation angle of the template that yields the maximum inner product, gives the feature orientation. Since the optimal 1-D template did not have explicit formulae, they were typically approximated by simple first or second order differentials of a Gaussian. In practice, they were extended using Gaussian kernels of the same variance since the resulting 2-D template could be applied in a directional manner inexpensively via the computation of smoothed image gradients or Hessians.

An alternative to these differential approaches to rotation independent feature detection is provided by the elegant work of Freeman and Adelson on steerable filters [42]. The underlying principle is to generate the rotated version of a filter from a suitable linear combination of basis filters; this sets some angular bandlimiting constraints on the class of admissible filters. Perona et. al., Simoncelli and Teo et. al. used this framework to approximate and design orientation-selective feature detectors [43, 44, 45, 46]. The concept of steerablity was also applied successfully in other areas of image processing such as texture analysis [47, 48] and image denoising [49].

In this chapter, we propose to reconcile the two methodologies— computational approach and steerable filterbanks— by presenting a general strategy for the design of 2-D steerable feature detectors. We derive the filter directly in 2-D as opposed to the 1-D schemes (1-D optimization followed by an extension to 2-D) of Canny and others. Moreover, in contrast with the work of Perona [43], we do not approximate a given template within a steerable solution space, but search for the filter that gives the best response according to an optimality criterion. Our filter is specified so as to provide the best compromise in terms of signal-to-noise ratio, false detections and localization. We illustrate the method with the design of optimal edge and ridge templates. The detectors that we obtain analytically have better performance and improved orientation selectivity, yet they are still computationally quite attractive.

The chapter is organized as follows. In Section 3.2, we introduce the concept of steerable matched filtering and reinterpret some of the classical detectors within this framework. In Section 3.3, we propose an optimality criterion and show how to determine the best filter from a class of steerable functions. In Section 3.4, we concentrate on specific 2-D feature detectors and demonstrate their use in different applications. Though our algorithm is general, in this chapter, we focus only on the detection of edge and ridge features. In Section 3.5, we introduce the concept of shape adaptive feature extraction and illustrate it with an example.

3.2 Orientation independent matched filtering

3.2.1 Detection by rotating matched filtering

Suppose our task is to detect some feature in an image $f(x, y)$ at some unknown position and orientation. The detection procedure can be formulated as a rotated matched filtering. It involves the computation of inner-products with the shifted and rotated versions of a 2-D feature template $f_0(x, y) = h(-x, -y)$ at every point in the image. A high magnitude of the inner-product indicates the presence of the feature and the angle of the corresponding template gives the orientation. Some simple examples of templates are shown in Fig. 3.1. Mathematically, the estimation algorithm is

$$\theta^*(\mathbf{x}) = \arg\max_{\theta} \left(f(\mathbf{x}) * h(\mathbf{R}_\theta \mathbf{x}) \right) \tag{3.1}$$

$$r^*(\mathbf{x}) = f(\mathbf{x}) * h(\mathbf{R}_{\theta^*} \mathbf{x}), \tag{3.2}$$

where r^* is the magnitude of the feature and θ^* its orientation at the position $\mathbf{x} = (x, y)$; $\mathbf{R}_\theta$ is the rotation matrix

$$\mathbf{R}_\theta = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.3}$$

and $u * v$ stands for the convolution between u and v. Equations (3.1) and (3.2) correspond to the matched filter detection. They give the maximum likelihood estimation of the angle θ and weight r for the signal model

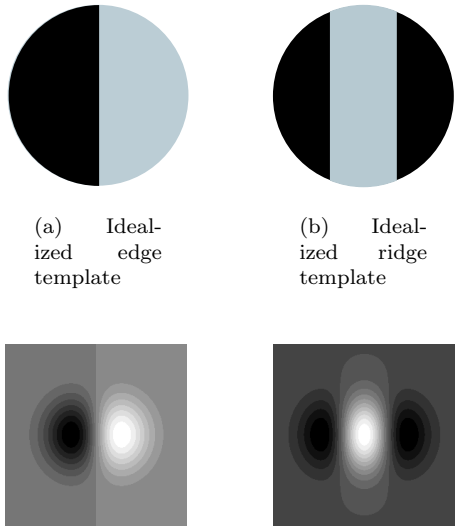
$$f(\mathbf{x}) = r \cdot f_0(\mathbf{R}_\theta(\mathbf{x} - \mathbf{x}_0) + \mathbf{x}_0) + n(\mathbf{x}_0),$$

where $n(\mathbf{x})$ denotes Gaussian white noise. However, this scheme of detection is not very practical, for it requires the implementation of a large number of filters (as many as the quantization levels of the angle).

3.2.2 Steerable filters

To cut down on the computational load, we select our detector within the class of steerable filters introduced by Freeman et. al [42]. These filters can be rotated very efficiently by taking a suitable linear combination of a small number of filters. Specifically, we consider templates of the form

$$h(x, y) = \sum_{k=1}^{M} \sum_{i=0}^{k} \alpha_{k,i} \frac{\partial^{k-i}}{\partial x^{k-i}} \frac{\partial^i}{\partial y^i} g(x, y), \tag{3.4}$$



(a) Idealized edge template

(b) Idealized ridge template

(c) Popular edge template

(d) Popular ridge template

Figure 3.1: Examples of feature templates. Feature detection is performed by convolution of the rotated versions of the template with the image

where $g(x, y)$ is an arbitrary isotropic window function. We call such a $h(x, y)$ an Mth order detector.

Proposition 1 *The filter $h(x, y)$ is steerable. In other words, the convolution of a signal $f(x, y)$ with any rotated version of $h(x, y)$ can be expressed as*

$$f(\mathbf{x}) * h(\mathbf{R}_\theta \mathbf{x}) = \sum_{k=1}^{M} \sum_{i=0}^{k} b_{k,i}(\theta) f_{k,i}(\mathbf{x}), \tag{3.5}$$

where the functions $f_{k,i}(x,y)$ are filtered versions of the signal $f(x,y)$

$$f_{k,i}(x,y) = f(x,y) * \underbrace{\left(\frac{\partial^{k-i}}{\partial x^{k-i}} \frac{\partial^{i}}{\partial y^{i}}\, g(x,y)\right)}_{g_{k,i}(x,y)}. \tag{3.6}$$

The orientation-dependent weights $b_{k,i}(\theta)$ are given by

$$b_{k,i}(\theta) = \left(\sum_{j=0}^{k} \alpha_{k,j} \sum_{l,m \in \mathcal{S}(k,j,i)} \binom{k-j}{l}\binom{j}{m} (-1)^{m} \cos(\theta)^{j+(l-m)} \sin(\theta)^{(k-j)-(l-m)}\right) \tag{3.7}$$

where, $\mathcal{S}(k,i,j)$ is the set

$$\mathcal{S}(k,i,j) = \{l,m \mid 0 < l < k-i; 0 < m < i; k-(l+m) = j\}.$$

The proof is given in the Appendix 3-A. A graphical representation of the implementation is given in Fig. 3.2. Once the $f_{k,i}(x,y)$ is available, $f(\mathbf{x}) * h(\mathbf{R}_\theta \mathbf{x})$ can be evaluated very efficiently via a weighted sum with its coefficients that are trigonometric polynomials of θ. Since the number of partial differentials in (3.5) for a general M^{th} order template is $M(M+3)/2$, $h(\mathbf{x})$ is steerable in terms of as many individual separable functions. Using some simplification, we can show that such a general $h(\mathbf{x})$ can also be rotated using $2M+1$ non-separable filters[2] (an example of such a simplification is given by (3.39)—(3.42)).

A case of special interest corresponds to $g(\mathbf{x})$ being the Gaussian; indeed the Gaussian is optimally localized in the sense of the uncertainty principle and the corresponding filters in (3.6) are all separable. Interestingly, the Gaussian family is equivalent to the class of moment filters (polynomials multiplied by Gaussian window) discussed in [42], but the filters are not identical. We will now show that the family described by (3.4) includes some popular feature detectors as particular cases.

3.2.3 Conventional detectors revisited

Canny's edge detector

As already observed by Freeman et. al., the widely-used Canny edge detection algorithm can be reinterpreted in terms of steerable filters [42]. This algorithm

[2]This is the minimum number of filters required to steer a general M^{th} order tempate.

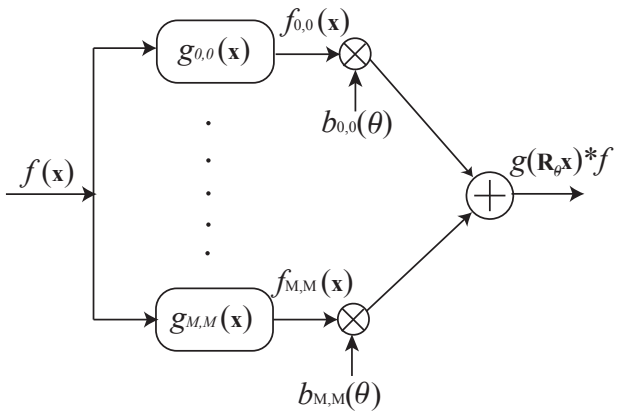


Figure 3.2: Implementation of steerable filtering (c.f (3.5))

involves the computation of the gradient-magnitude of the Gaussian-smoothed image. The direction of the gradient gives the orientation of the edge. Mathematically,

$$\theta^* = \arctan\left(\frac{(f*g)_y}{(f*g)_x}\right) \tag{3.8}$$

$$r^* = \sqrt{((f*g)_x)^2 + \left((f*g)_y\right)^2}, \tag{3.9}$$

where $g_x = \partial g/\partial x$ and $g_y = \partial g/\partial y$; g is a 2-D Gaussian of a specified variance. The above set of equations can be shown to be the solution of (3.1) and (3.2), with $h = g_x$. Substituting $M = 1$; $\alpha_{1,0} = 1, \alpha_{1,1} = 0$ in (3.7) we get $b_{1,0}(\theta) = \cos(\theta), b_{1,1}(\theta) = \sin(\theta)$. Thus,

$$\theta^*(\mathbf{x}) = \arg\max_\theta \left(f(\mathbf{x}) * g_x(\mathbf{R}_\theta\mathbf{x})\right) \tag{3.10}$$

$$= \arg\max_\theta \left(f * (g_x\cos(\theta) + g_y\sin(\theta))\right). \tag{3.11}$$

Here, we used the steerability of g_x from (3.5). To compute the maximum of the above expression, we set the differential of (3.11) with respect to θ to zero:

$$(f*g_x)\sin(\theta) - (f*g_y)\cos(\theta) = 0, \tag{3.12}$$

which results in (3.8) and (3.9). The corresponding feature template is shown in Fig.3.1-c.

Ridge detector

Less well known is the fact that a popular ridge estimator based on the eigen-decomposition of the Hessian matrix [50, 51, 52] can also be interpreted in terms of steerable filters. Assuming the template to be g_{xx} (the second derivative of a Gaussian), ridge detection can be formulated exactly as (3.1) and (3.2). The corresponding detector is shown in Fig.3.1-d. In this case, the steerability relation (3.5) can be expressed in a matrix form as

$$g_{xx}\left(\mathbf{R}_\theta \mathbf{x}\right) = \mathbf{u}_\theta^{\mathrm{T}} \underbrace{\begin{bmatrix} g_{xx}\left(\mathbf{x}\right) & g_{xy}\left(\mathbf{x}\right) \\ g_{xy}\left(\mathbf{x}\right) & g_{yy}\left(\mathbf{x}\right) \end{bmatrix}}_{\mathbf{H}_g} \mathbf{u}_\theta, \tag{3.13}$$

where $\mathbf{H}_g$ is the Hessian matrix and $\mathbf{u}_\theta = \left(\cos\left(\theta\right), \sin\left(\theta\right)\right)$. Using the linearity of convolution, $f\left(\mathbf{x}\right) * g_{xx}\left(\mathbf{R}_\theta \mathbf{x}\right) = \mathbf{u}_\theta^{\mathrm{T}}\ \mathbf{H}_{f*g}\ \mathbf{u}_\theta$. We would like to obtain the maximum of $\mathbf{u}_\theta^{\mathrm{T}}\ \mathbf{H}_{f*g}\ \mathbf{u}_\theta$, subject to the constraint $\mathbf{u}_\theta^{\mathrm{T}}\mathbf{u}_\theta = 1$. We solve this constrained optimization problem using Lagrange's multiplier method by setting the gradient of $\mathbf{u}_\theta^{\mathrm{T}}\mathbf{H}_{f*g}\mathbf{u}_\theta + \lambda\ \mathbf{u}_\theta^{\mathrm{T}}\mathbf{u}_\theta$ to zero:

$$\mathbf{H}_{f*g}\mathbf{u}_\theta = -\lambda\ \mathbf{u}_\theta. \tag{3.14}$$

This implies that $-\lambda$ is an eigen value of $\mathbf{H}_{f*g}$; the corresponding normalized eigenvectors are the possible solutions to the problem. Since we are looking for the maximum of $\mathbf{u}_\theta^{\mathrm{T}}\ \mathbf{H}_{f*g}\ \mathbf{u}_\theta$, the optimal response and the angle are given by

$$r^* = \lambda_{\max} \tag{3.15}$$

$$\mathbf{u}_{\theta^*} = \mathbf{v}_{\max}. \tag{3.16}$$

Here $\lambda_{\max}$ and $\mathbf{v}_{\max}$ are the maximum eigenvalue and the corresponding eigenvector respectively.

It can be seen from Fig.3.1-c and 3.1-d that these classical detectors do not have a good orientation selectivity. In the next section, we propose a new approach for the design of detectors that attempts to correct for this deficiency.

3.3 Design of steerable filters for feature detection

The widely-used contour extraction algorithm[39] has three steps: (a) feature detection, (b) non-maximum suppression, and, (c) thresholding. In this section,

we present a general strategy for the design of steerable filters for feature detection, while keeping in mind the subsequent steps. We propose a criterion similar to that of Canny and we analytically derive the optimal filter—or equivalently the optimal weights—within our particular class of steerable functions specified by (3.4).

3.3.1 Optimality criterion

We now review Canny's criterion and modify it slightly to enable analytical optimization. To derive the optimal 2-D operator, we assume that the feature (edge/ridge) is oriented in some direction[3] (say along the x axis) and derive an optimal operator for its detection. As the operator is rotation-steerable by construction, its optimality properties will be independent of the feature orientation.

The 3 different terms in Canny's criterion are as follows:

Signal-to-Noise Ratio

The key term in the criterion is the signal-to-noise ratio. The response of a filter $h(\mathbf{x})$ to a particular signal $f_0(\mathbf{x})$ (e.g. an idealized edge) centered at the origin is given by

$$S = \int_{\mathcal{R}^2} f_0(x, y)\, h(-x, -y)\, dx\, dy \tag{3.17}$$

S is given by the height of the response at its maximum. If the input is corrupted by additive white noise of unit variance, then the variance of the noise at the output is given by the energy of the filter:

$$\text{Noise} = \int_{\mathcal{R}^2} |h(x, y)|^2\, dx\, dy \tag{3.18}$$

We desire to have a high value of S for a given value of Noise; $\frac{S^2}{\text{Noise}}$ is the amplification of the desired feature provided by the detector. The detection stage is preceded by non-maximum suppression. The estimated feature position corresponds to the location of the local maximum of the response in the direction orthogonal to the feature boundary (y axis in our case). The presence of noise can cause an undesirable shift in the estimated feature location. The direct

[3] In 2-D the features of interest have boundaries of dimension 1.

extension of Canny's expression for the shift-variance (due to white noise of unit variance) to 2-D gives

$$E\left[(\Delta y)^2\right] = \frac{\int_{\mathcal{R}^2} |h_y(x,y)|^2 \, dx \, dy}{\left|\int_{\mathcal{R}^2} f_0(x,y) \, h_{yy}(-x,-y) \, dx \, dy\right|^2} \tag{3.19}$$

Canny has proposed to maximize the reciprocal of this term. The numerator of (3.19) is a normalization term which will be small automatically if the impulse response of the filter is smooth along the y axis (low norm for the derivative). Since we are imposing this type of smoothness constraint elsewhere via an additional regularization term (see next subsection), it is not necessary to optimize this term here, which also keeps the effects well separated. Therefore, we propose to maximize the second derivative of the response, orthogonal to the boundary, at the origin

$$\begin{aligned} \text{Loc} &= -\frac{d^2}{dy^2}(f_0 * h) \\ &= -\int_{\mathcal{R}^2} f_0(x,y) \, h_{yy}(-x,-y) \, dx \, dy \end{aligned} \tag{3.20}$$

which is the square-root of the denominator in (3.20). The above expression is ensured to be positive because the second derivative of the response is negative at the maximum (assuming $S > 0$). Note that the new localization term is a measure of the width of the peak. The drift in position of the maximum due to noise will decrease as the response becomes sharper. In this work, we are neglecting the effect of neighboring signals on the localization.

Elimination of false oscillations

Canny observed that when the criterion is optimized only with the SNR and the localization constraint, the optimal operator has a high bandwidth; the response will be oscillatory and hence have many false maximas. In 2-D, we desire that the response be relatively free of oscillations orthogonal to the feature boundary. This can be achieved by penalizing the term:

$$R_o = \int_{\mathcal{R}^2} |h_{yy}(x,y)|^2 \, dx \, dy \tag{3.21}$$

Note that this term is the numerator of the expression for the mean distance between zero crossings proposed by Canny. It is a thin-plate spline like regularization which is a standard technique to constrain a solution to be smooth (low bandwidth).

The thresholding step is easier if the response is flat along the boundary. The oscillation of the response along the boundary (x axis) can be minimized by penalizing

$$R_p = \int_{\mathcal{R}^2} \left| h_{xx}\left(x,y\right)\right|^2 dx\, dy \tag{3.22}$$

These terms will force the filter to be smooth making the response less oscillatory, thus resulting in fewer false detections.

3.3.2 Derivation of the optimal detector

We combine the individual terms to obtain a single criterion

$$C = S \cdot \mathrm{Loc} - \mu \underbrace{\left(R_o + R_p\right)}_{R} \tag{3.23}$$

The filter in the family described by (3.4) that maximizes this criterion, subject to the constraint[4] Noise = 1, is our optimal detector. The free parameter $\mu > 0$ controls the smoothness of the filter; a high value makes the response less prone to false maxima and reduces oscillation along the ridge. However, these properties impose a tradeoff on the localization of the response.

In this work, we are also interested in performing a scale-independent design. In other words, if we dilate the window by a factor σ, using $g_\sigma\left(\mathbf{x}\right) = \sigma^{-\frac{1}{2}} g\left(\frac{\mathbf{x}}{\sigma}\right)$, we want our solution to retain the shape independently of σ. This requires that we weight each of the terms in (3.23) using an appropriate power of the dilation factor. This issue is discussed later for each feature model separately.

For the ease of notation, we collect the component functions of (3.4) into a function vector $\mathbf{g}$ of length $\left(\frac{M(M+3)}{2}\right)$, whose components are

$$\begin{aligned} \left[\mathbf{g}\right]_i\left(x,y\right) &= \frac{\partial^{k-n}}{\partial x^{k-n}} \frac{\partial^n}{\partial y^n}\, g\left(x,y\right) \quad \text{with } i = \frac{\left(k-1\right)\left(k+2\right)}{2} + n \\ & k = 0...M,\ n = 0..k. \end{aligned}$$

Hence, an arbitrary function in the family is represented in a compact form as

$$h\left(\mathbf{x}\right) = \mathbf{a}^{\mathrm{T}} \mathbf{g}\left(\mathbf{x}\right) \tag{3.24}$$

[4]This constraint is just a normalization factor. Setting Noise to another constant will give detectors of the same shape, but with a different energy.

where $\mathbf{a}$ is the vector containing the $\alpha_{i,k}$'s in (3.4); it has the same length as the function vector. Now we express the terms of the criterion in a matrix form as $S = \mathbf{a}^{\mathrm{T}}\mathbf{s}$, Loc $= \mathbf{a}^{\mathrm{T}}\mathbf{q}$, Noise $= \mathbf{a}^{\mathrm{T}}\mathbf{P}\,\mathbf{a}$ and $R = \mathbf{a}^{\mathrm{T}}\mathbf{R}\,\mathbf{a}$, where

$$[\mathbf{s}]_i = \langle f_0(\mathbf{x}), [\mathbf{g}(-\mathbf{x})]_i \rangle \tag{3.25}$$

$$[\mathbf{q}]_i = \left\langle f_0(\mathbf{x}), ([\mathbf{g}(-\mathbf{x})]_i)_{yy} \right\rangle \tag{3.26}$$

$$[\mathbf{P}]_{i,j} = \left\langle [\mathbf{g}]_i, [\mathbf{g}]_j \right\rangle \tag{3.27}$$

$$[\mathbf{R}]_{i,j} = \left\langle ([\mathbf{g}]_i)_{yy}, \left([\mathbf{g}]_j\right)_{yy} \right\rangle + \left\langle ([\mathbf{g}]_i)_{xx}, \left([\mathbf{g}]_j\right)_{xx} \right\rangle. \tag{3.28}$$

$g_{yy}(x,y)$ and $g_{xx}(x,y)$ denote $\partial^2 g(x,y)/\partial y^2$ and $\partial^2 g(x,y)/\partial x^2$ respectively. $\mathbf{P}$ and $\mathbf{R}$ are matrices of size $\frac{M(M+3)}{2} \times \frac{M(M+3)}{2}$, while the vectors $\mathbf{q}$ and $\mathbf{s}$ are of length $\frac{M(M+3)}{2}$. Here $\mathbf{P}$ is ensured to be nonsingular. In the above expressions, the inner product of two functions is defined as

$$\langle f_1, f_2 \rangle = \int_{\mathcal{R}^2} f_1(x,y)\, f_2(x,y)\, dx\, dy.$$

Thus, the criterion (3.23) can be expressed in the matrix form as

$$C = \mathbf{a}^{\mathrm{T}} [\mathbf{Q} - \mu\, \mathbf{R}]\, \mathbf{a}, \tag{3.29}$$

where

$$\mathbf{Q} = \mathbf{s}\, \mathbf{q}^{\mathrm{T}} \tag{3.30}$$

Since all the terms in the criterion are quadratic, the solution for the optimal parameters can be found analytically by using Lagrange's multiplier method. To maximize the criterion subject to the constraint, we set the gradient of $C + \lambda$ Noise to zero:

$$2\,[\mathbf{Q} - \mu\mathbf{R} + \lambda\, \mathbf{P}]\, \mathbf{a} = 0 \tag{3.31}$$

Rearranging the terms, we get

$$\mathbf{P}^{-1}\,[\mathbf{Q} - \mu\mathbf{R}]\;\, \mathbf{a} = -\lambda\, \mathbf{a} \tag{3.32}$$

which implies that λ is an eigenvalue of the matrix $\left(-\mathbf{P}^{-1}[\mathbf{Q} - \mu\mathbf{R}]\right)$. The total number of eigenvalues is given by the dimension of $\mathbf{a}$. The corresponding eigenvectors $\mathbf{a}_{\lambda_i}$ need to be scaled so that the constraint $\mathbf{a}_{\lambda_i}^{\mathrm{T}}\mathbf{P}\mathbf{a}_{\lambda_i} = 1$ is satisfied. The optimal solution is therefore given by

$$\bar{\mathbf{a}} = \max\left\{\mathbf{a}_{\lambda_i}^{\mathrm{T}}\,[\mathbf{Q} - \mu\mathbf{R}]\,\mathbf{a}_{\lambda_i};\; i = 0...M\,(M+3)/2\right\} \tag{3.33}$$

Thus the design of the optimal feature detector boils down to an eigen-decomposition followed by an appropriate weighting of the eigen-vectors so as to satisfy the constraint.

3.3.3 Feature detection by local optimization

Due to (3.5), the optimal angle θ^* in (3.1) is obtained as the solution of

$$\begin{aligned} \frac{\partial}{\partial\theta}\left(f\left(\mathbf{x}\right) * h\left(\mathbf{R}_{\theta^*}\mathbf{x}\right)\right) &= \sum_{k=1}^{M}\sum_{i=0}^{k} f_{k,i}\left(x,y\right) \underbrace{\frac{\partial}{\partial\theta}\left(b_{k,i}\left(\theta\right)\right)|_{\theta=\theta^*}}_{c_{k,i}(\theta^*)} \\ &= 0 \end{aligned} \tag{3.34}$$

It is easy to see from (3.7) that each of the terms in $b_{k,i}\left(\theta\right)$ are of degree k in $\cos\left(\theta\right)$ and $\sin\left(\theta\right)$; $c_{k,i}\left(\theta\right)$ is of degree k as well. Hence, (3.34) is a polynomial of order M (in $\cos\left(\theta\right)$ and $\sin\left(\theta\right)$) and thus the estimation of the optimal angle involves the solution of an Mth order polynomial in two variables.

If $h\left(x,y\right)$ has only odd/even order partial derivatives (this is the case for many detectors), then $b_{k,i}\left(\theta\right)$ will be a polynomial with only odd/even degree terms (of $\cos\left(\theta\right)$ and $\sin\left(\theta\right)$) present. Consequently, (3.34) can be reduced[5] to a form where only terms of degree M are present. In this case, (3.34) can be further simplified (by dividing both the sides by $\left(\cos\left(\theta\right)\right)^M$) to a polynomial in only one variable—$\tan\left(\theta\right)$. We then have an analytic solution if $M <= 3$ [53]. This case is illustrated in Section 3.4.1. When $M = 2$, the solution can also be computed as an eigen-decomposition of the Hessian matrix, which is better known (but also boils down to the above mentioned solution). This case is described in Section 3.4.2. When the solution of (3.34) is not trackable analytically, it can be solved numerically using an iterative root finder such as the Newton-Raphson method.

3.4 2-D feature detectors

We now design operators optimized for the detection of different 2-D features. We chose the window function to be a Gaussian[6] $g\left(\mathbf{x};\sigma\right)$, where σ is the standard

[5] if there is a term of degree $M-2n$, we can multiply it by $\left(\cos\left(\theta\right)^2+\sin\left(\theta\right)^2\right)^n$ to make it of degree M

[6] it is the only function that is isotropic and separable.

deviation. When it is clear from the context, we will suppress the dependence on σ to simplify the notation.

3.4.1 Edge detection

As model for the edge, we choose the ideal step function

$$f_0(x,y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{else} \end{cases} \tag{3.35}$$

Since it is an odd function of y, the even order derivatives do not contribute to the signal energy; we therefore ignore[7] them in (3.4).

Case 1: $M = 1$

To illustrate the derivation of the optimal filter, we explain all the steps in detail in this simple case. Substituting the function vector $\mathbf{g} = [g_x, g_y]$ in the corresponding expressions, we get

$$\begin{aligned} \mathbf{s} &= -\sigma\sqrt{\pi}\ [0,1] \\ \mathbf{q} &= -\frac{2\sqrt{\pi}}{\sigma}\ [0,1] \\ \mathbf{P} &= \frac{\pi}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \mathbf{R} &= \frac{9\pi}{\sigma^4}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Thus,

$$\mathbf{Q} = \mathbf{q}^{\mathrm{T}}\mathbf{s} = 2\pi \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The matrices $\mathbf{Q}$ and $\mathbf{P}$ are independent of σ while $\mathbf{R}$ is inversely proportional to σ^4. So we weigh $\mathbf{R}$ by σ^4 to have a scale-invariant solution. Hence

$$\mathbf{P}^{-1}\left[\mathbf{Q} - \mu\sigma^4\mathbf{R}\right] = \begin{bmatrix} -18\mu & 0 \\ 0 & 4-18\mu \end{bmatrix} \tag{3.36}$$

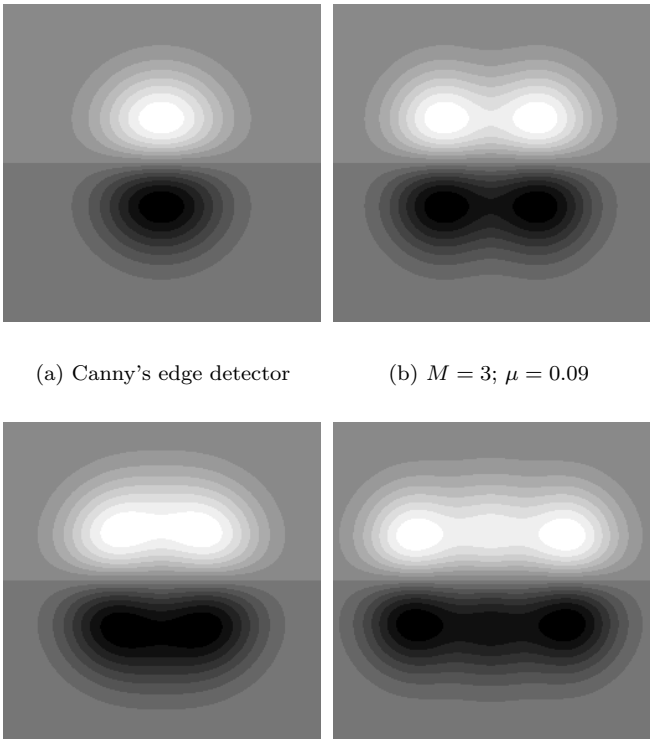
The eigenvalues of $\mathbf{P}^{-1}\left[\mathbf{Q} - \mu\sigma^4\mathbf{R}\right]$ are $\lambda_1 = -18\mu$ and $\lambda_2 = 4 - 18\mu$, respectively. The corresponding scaled eigenvectors (so as to satisfy the constraint)

[7] If we were to include them in the solution, their optimal coefficients would turn out to be zero anyway.

are $\left[0, -\sqrt{\frac{2}{\pi}}\right]$ and $\left[-\sqrt{\frac{2}{\pi}}, 0\right]$, respectively. When substituted in the criterion, they yield $4 - 18\mu$ and -18μ, respectively. Thus, the optimal solution is

$$\overline{\mathbf{a}} = \left[0, -\sqrt{\frac{2}{\pi}}\right]$$

(as $\mu > 0$), which corresponds to Canny's edge detector (c.f. Fig. 3.1-c).



(a) Canny's edge detector (b) $M = 3$; $\mu = 0.09$

(c) $M = 3$; $\mu = 0.2$ (d) M=5; $\mu = 0.15$

Figure 3.3: Edge Detectors for different parameters. The detectors become more orientation selective as M increases.

Higher order cases

For higher M, we obtain a family of solutions that are increasingly smooth when μ goes up. A few examples of higher order templates are given in Table 3.1 with the filter impulse responses shown in Fig. 3.3. By comparing Fig. 3.3-b and Fig. 3.3-c we observe that, as μ increases, the filter becomes smoother at the cost of directionality. The higher order templates are more elongated thus having higher SNR and localization (c.f. Table 3.1); they should therefore result in better detections, at-least for idealized edges. The dependence of SNR on σ^2 implies that this figure can also be improved by increasing the variance of the Gaussian. However, the ability to resolve two adjacent parallel edges decreases as σ increases.

Implementation

Here, we develop the implementation procedure mentioned in Section 3.3.3 for the special case of 3rd order edge detection. A general 3rd order edge template (for different values of μ) is given by

$$h(\mathbf{x}) = \alpha_{1,0}\, g_x + \alpha_{3,0}\, g_{xxx} + \alpha_{3,2}\, g_{xyy} \tag{3.37}$$

The rotated version[8] of this template h_θ is given by

$$\begin{aligned} h_\theta \;=\; & \alpha_{1,0}\, (g_x \cos(\theta) + g_y \sin(\theta)) + \\ & \alpha_{3,0} \Big(g_{xxx} \cos^3(\theta) + 3\, g_{xxy} \cos^2(\theta) \sin(\theta) + \\ & \quad 3\, g_{xyy} \cos(\theta) \sin^2(\theta) + g_{yyy} \sin^3(\theta) \Big) + \\ & \alpha_{3,2} \Big(g_{xyy} \cos^3(\theta) + (-2 g_{xxy} + g_{yyy}) \cos^2(\theta) \sin(\theta) + \\ & \quad (-2 g_{xyy} + g_{xxx}) \cos(\theta) \sin^2(\theta) + g_{xxy} \sin^3(\theta) \Big) \end{aligned}$$

Convolving the rotated template by f and simplifying, we get

$$\begin{aligned} (f * h_\theta)(\mathbf{r}) \;=\; & q_1(\mathbf{r}) \cos(\theta)^3 + q_2(\mathbf{r}) \cos(\theta)^2 \sin(\theta) + \\ & q_3(\mathbf{r}) \cos(\theta) \sin(\theta)^2 + q_4(\mathbf{r}) \sin(\theta)^3 , \end{aligned} \tag{3.38}$$

[8] The expression for a general rotated template is given by (3.5) and (3.7). However, for simple templates, it may be easier to derive it directly in the Fourier domain as shown in the Appendix, (3.60).

	S^2/Noise	Loc	# Basis filters	Expression	Implementation
$M = 1$	$2\ \sigma^2$	1.63	2 separable	$-\sqrt{\frac{2}{\pi}}\, g_y$	Analytic
$M = 3$			4 non-separable		
$\mu = 0.09$	$2.93\ \sigma^2$	1.98	6 separable	$-0.966\ g_y - 0.256\ \sigma^2\ g_{xxy}$	Analytic
$M = 3$			4 non-separable	$-1.0655\ g_y - 0.20\ \sigma^2\ g_{xxy}$	
$\mu = 0.2$	$3.01\ \sigma^2$	1.83	6 separable	$-0.042\ \sigma^2\ g_{yyy}$	Analytic
				$-1.1215\ g_y$	
				$-0.5576\ \sigma^2\ g_{xxy}$	
$M = 5$	$3.69\ \sigma^2$	2.15	6 non-separable	$-0.018\ \sigma^2\ g_{yyy}$	Sampling/
$\mu = 0.15$			12 separable	$-0.0415\ \sigma^4\ g_{xxxxy}$	Iterative
				$-0.0038\ \sigma^4\ g_{xxyyy}$	

Table 3.1: Edge detectors for different parameters.

where

$$q_1(\mathbf{r}) = \alpha_{3,0}\, f_{3,0}(\mathbf{r}) + \alpha_{3,2}\, f_{3,2}(\mathbf{r}) + \alpha_{1,0}\, f_{1,0}(\mathbf{r}) \tag{3.39}$$

$$q_2(\mathbf{r}) = (3\alpha_{3,0} - 2\alpha_{3,2})\, f_{3,1}(\mathbf{r}) + \alpha_{3,2}\, f_{3,3}(\mathbf{r}) + \alpha_{1,0}\, f_{1,1}(\mathbf{r}) \tag{3.40}$$

$$q_3(\mathbf{r}) = (3\alpha_{3,0} - 2\alpha_{3,2})\, f_{3,2}(\mathbf{r}) + \alpha_{3,2}\, f_{3,0}(\mathbf{r}) + \alpha_{1,0}\, f_{1,0}(\mathbf{r}) \tag{3.41}$$

$$q_4(\mathbf{r}) = \alpha_{3,0}\, f_{3,3}(\mathbf{r}) + \alpha_{3,2}\, f_{3,1}(\mathbf{r}) + \alpha_{1,0}\, f_{1,1}(\mathbf{r}) \tag{3.42}$$

We multiplied the single degree terms in $\cos(\theta)$ and $\sin(\theta)$ with $\left(\cos^2(\theta) + \sin^2(\theta)\right)$ so that we get a polynomial with only third degree terms. Note that the six functions $f_{k,i};\ k = \{1,3\}, i = 0 \ldots k$, obtained by separable filtering, are combined to derive $q_i;\ i = 1 \ldots 4$. They can also be obtained by non-separable filtering:

$$q_1(\mathbf{r}) = f * \left(\alpha_{3,0}\, g_{3,0} + \alpha_{3,2}\, g_{3,2} + \alpha_{1,0}\, g_{1,0}\right)(\mathbf{r}) \tag{3.43}$$

$$q_2(\mathbf{r}) = f * \left((3\alpha_{3,0} - 2\alpha_{3,2})\, g_{3,1} + \alpha_{3,2}\, g_{3,3} + \alpha_{1,0}\, g_{1,1}\right)(\mathbf{r}) \tag{3.44}$$

$$q_3(\mathbf{r}) = f * \left((3\alpha_{3,0} - 2\alpha_{3,2})\, g_{3,2} + \alpha_{3,2}\, g_{3,0} + \alpha_{1,0}\, g_{1,0}\right)(\mathbf{r}) \tag{3.45}$$

$$q_4(\mathbf{r}) = f * \left(\alpha_{3,0}\, g_{3,3} + \alpha_{3,2}\, g_{3,1} + \alpha_{1,0}\, g_{1,1}\right)(\mathbf{r}) \tag{3.46}$$

We use the separable approach due to its computational efficiency. The non-separable approach may be profitable for large values of M.

For a particular value of $\mathbf{r}$, $(f * h_\theta(\mathbf{r}))$ is a function of only one variable—θ. At the local maxima and the minima of $(f * h_\theta(\mathbf{r}))$, we have $\frac{\partial}{\partial\theta}(f * h_\theta(\mathbf{r})) = 0$. Substituting for $f * h_\theta$ from (3.38), we get

$$\begin{aligned} q_2\, \cos(\theta)^3 + (2q_3 - 3q_1)\cos^2(\theta)\sin(\theta) + \\ (3q_4 - 2q_2)\cos(\theta)\sin^2(\theta) - q_3\, \sin^3(\theta) \quad = \quad 0, \end{aligned}$$

(a) Noisy Image

(b) Canny (Time taken 141 ms)

(c) $M = 3$; $\mu = 0.09$ (Time taken 414 ms)

(d) $M = 5$; $\mu = 0.15$ (Time taken 1995 ms)

Figure 3.4: Edge detection on a 256 x 256 noisy image (Gaussian white noise of variance 85). The thresholding is performed such that there are 2000 detected pixels in each image. The variance of the Gaussian window is chosen as 1.7. Note that the higher order detectors give less wiggly contours with fewer breaks. The algorithm was implemented in Java as a plugin for ImageJ. The experiments were performed on a Intel Pentium processor at 2.66 GHz.

We divide both sides of this equation by $\cos(\theta)^3$ to get a cubic polynomial in one variable—$\tan\theta$:

$$q_2 + (2q_3 - 3q_1)\tan(\theta) + (3q_4 - 2q_2)\tan^2(\theta) - q_3\tan(\theta)^3 = 0, \tag{3.47}$$

The roots of this equation can be obtained analytically [53]. Since $\tan(\theta) = \tan((\theta + \pi) \mod 2\pi)$, there are six possible values of θ in the range $[0, 2\pi]$ that satisfy (3.47). One of these values of θ correspond to the global maximum; it can be found out by substituting all them into (3.38) and picking the one which gives the maximum value. We briefly describe the steps of the local optimization algorithm in Appendix 3-B.

For $M > 3$, the θ^* estimated for $M = 3$ can act as an approximate solution. This initial guess is further refined by performing a golden search [53] around the approximate solution.

Results

Because the scheme is optimized for noisy data, we perform edge detection on the cameraman image corrupted with additive white noise (c.f. Fig. 3.4-a). The size of the Gaussian window is the same in all the experiments. The detected edges after non-maximum suppression and thresholding are presented in Fig. 3.4. It is seen that Canny's edge detector has a lot of false detections. Moreover, the detected edges are wiggly due to poor localization. The new detectors have significantly lower false detections and better localization, thus confirming the theoretical improvement.

Note the time taken for the various edge detection schemes from Fig. 3.4 b-d. The 3rd order scheme only takes around 2.5 times the time as the Canny's detector. We believe that, for the performance improvement achieved, it is a quite reasonable price to pay. Since we resorted to a naive optimization algorithm using dichotomy, the 5th order method took more time. We believe that a better optimization scheme could drastically improve the computational efficiency.

3.4.2 Ridge detection

For simplicity, we choose the idealized line model as:

$$f_0(x, y) = \delta(y), \tag{3.48}$$

where δ denotes the Dirac delta function. A more realistic model can be assumed without any change in the computational strategy. Here $\mathbf{Q}$, $\mathbf{P}$ and $\mathbf{R}$

	S^2/Noise	Loc	# Basis filters	Expression	Implementation
$M = 2$ $\mu = 2$	2.67	$4.38/\sigma$	3 separable	$-\sqrt{\frac{2}{3\pi}}\sigma\ g_{yy}$	Analytic
$M = 2$ $\mu = 0$	3	$4.64/\sigma$	3 separable	$-\sqrt{\frac{3}{4\pi}}\sigma\ (g_{yy} - g_{xx}/3)$	Analytic
$M = 4$ $\mu = 0.1$	4.302	$6.41/\sigma$	5 non-separable 8 separable	$-0.204\sigma\ g_{yy} + 0.059\sigma\ g_{xx}$ $+0.063\sigma^3\ g_{yyyy} - 0.194\sigma^3\ g_{xxyy}$ $+0.024\sigma^3\ g_{xxxx}$	Analytic
$M = 4$ $\mu = 1/4$	4.47	$6.14/\sigma$	5 non-separable 8 separable	$-0.392\sigma\ g_{yy} + 0.113\sigma\ g_{xx}$ $+0.034\sigma^3\ g_{yyyy} - 0.184\sigma^3\ g_{xxyy}$ $+0.025\sigma^3\ g_{xxxx}$	Analytic

Table 3.2: Ridge detectors for different parameters

are inversely proportional to σ^4, σ^2 and σ^6, respectively. Hence, we scale $\mathbf{Q}$ by σ^2 and $\mathbf{R}$ by σ^4.

Optimized detectors

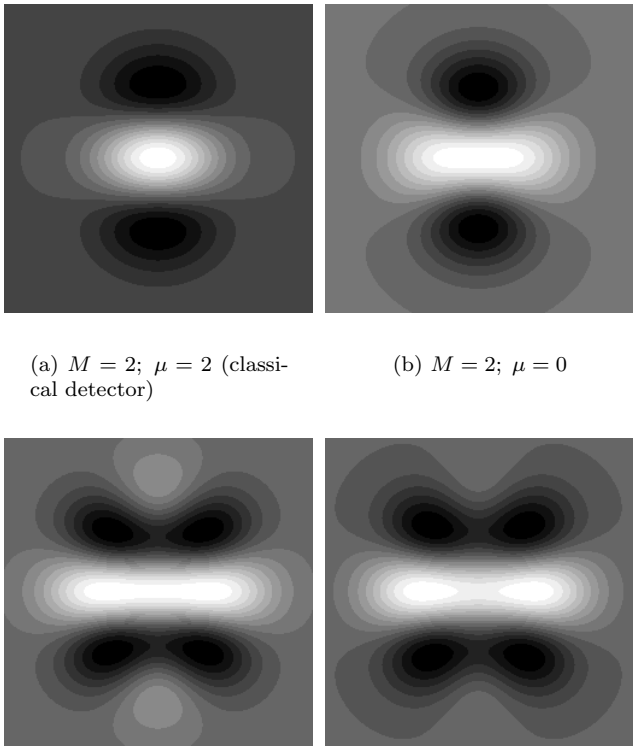
Some examples of optimal templates are shown in Table 3.2 and Fig. 3.5. Interestingly, we see from the table that the optimal detector for $M = 2$ and $\mu = 0$ is better than the classical detector, both in terms of SNR and localization, at no additional cost. Also note that the template in Fig. 3.5-b is more directional than the classical one in Fig. 3.5-a. The high value of $\mu = 2$ (adjusted to get the equivalence) overconstrains the optimization, resulting in a lower performance.

Two cases for $M = 4$ are also shown. It is seen that for small μ, the template oscillates along y producing undesirable sidelobes. However, it has a better localization at the expense of a lower SNR and R.

In general, we found that it is better to have a low value of μ for lower order templates; the model have few degrees of freedom and hence a high value of μ will overconstrain the system. On the other hand, for higher order templates, we need a higher value of μ to make them less oscillatory.

Implementation

Any second order detector can be implemented as an eigen-decomposition, similar to the classical Hessian (described in Section 3.2.3). For example, the de-



(a) $M = 2;\ \mu = 2$ (classical detector)

(b) $M = 2;\ \mu = 0$

(c) $M = 4;\ \mu = 0.1$

(d) $M = 4;\ \mu = 0.25$

Figure 3.5: Ridge Detectors corresponding to different orders and parameters.

tector with $\mu = 0$ can be implemented as

$$\begin{aligned} \theta^* &= \arg\max_{\theta} f * \left(g_{\mathbf{u}_\theta, \mathbf{u}_\theta} - \frac{g_{\mathbf{u}_{\theta+\frac{\pi}{2}}, \mathbf{u}_{\theta+\frac{\pi}{2}}}}{3} \right) \\ &= \arg\max_{\theta} \left(\mathbf{u}_\theta^{\mathrm{T}}\ \mathbf{H}_{\mathrm{mod}}\ \mathbf{u}_\theta \right), \end{aligned}$$

where $\mathbf{H}_{\mathrm{mod}} = \mathbf{H}_{f*g} - \frac{1}{3}\left(\mathbf{P}^{\mathrm{T}}\ \mathbf{H}_{f*g}\ \mathbf{P}\right)$; here $\mathbf{P}$ is the rotation matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{3.49}$$

such that $\mathbf{P}\mathbf{u}_\theta = \mathbf{u}_{(\theta+\frac{\pi}{2})}$. Thus the optimal direction and ridge magnitude can be computed with the eigen-decomposition of $\mathbf{H}_{\text{mod}}$; the computational complexity is the same as with the classical scheme.

For the 4th order detector, we proceed exactly as in the case of the third order edge template. The computation of the optimal angle involves the solution of a quartic polynomial, which is also performed analytically [53].

Results

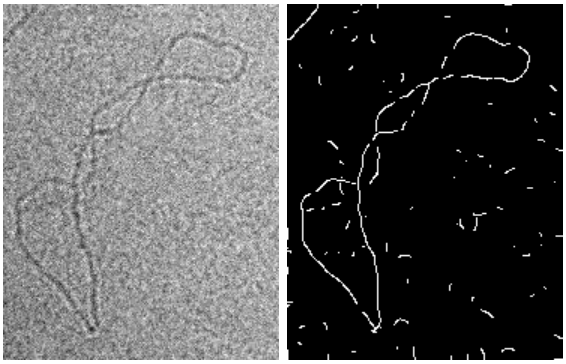
An interesting application, which motivated this whole development, is the detection of DNA filaments (cf Fig. 3.6-a) from their stereo cryo-electron micrographs [51]. The difficulty with these data is that the micrographs are extremely noisy because they are exposed to a low electron dose to avoid the degradation of the specimen. The results (Fig. 3.6-b - 3.6-d) correspond to the output of ridge detection algorithm followed by non-maximum suppression and thresholding. Overall, the $M = 4$ detector gives the best qualitative results: there are few breaks in the filament and the detection is less wiggly. Note that the performance improvement costed only 2 times the time taken for the classical approach. The optimal second order detector gave better results for the same computational complexity as the classical approach.

3.5 Shape adaptable feature detection

Steerability in rotation involves the representation of a template as a weighted linear combination of a few filters; the weights are nonlinear functions of a single parameter—the angle. This leaves us with extra degrees of freedom which can be utilized effectively. Perona used it to make the template steerable in scale [43]. We propose to utilize this freedom for the design of a shape-adaptable filter, thus making the system respond to different shapes depending on the parameters.

In Section 3.4.1, we designed templates for the detection of ideal step edges. However, as mentioned in [54], the edges are sometimes wedge shaped (close to image corners). Since this contradicts our assumption, we have low SNR at the corners. A bias in the position of the corner is also reported in the context of conventional corner detectors [55].

Corners are image regions with high surface curvature. They convey a lot of information about the image shape [56, 57, 58, 59]. Hence, we propose a new shape-adaptable, steerable corner detector that addresses these issues.



(a) DNA micrograph

(b) Classical detector (Time: 260 ms)

(c) $M = 2; \mu = 0$ (Time: 260 ms)

(d) $M = 4; \mu = 0.25$ (Time: 590 ms)

Figure 3.6: Detection of DNA filament from its noisy cryo-electron micrograph. The features were ridges that were roughly 2-3 pixels wide. We chose the standard deviation of the Gaussian window to be 3. The images were thresholded such that there are 1000 detected pixels.

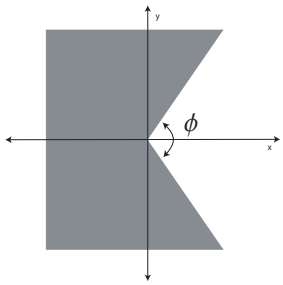


Figure 3.7: Model of an ideal wedge.

3.5.1 Derivation of the wedge detector

We model a corner as a wedge shown in Fig. 3.7, where the wedge angle ϕ is a variable. Analytically, we have

$$f_0(x,y) = \begin{cases} 1 & \text{if } -x\sin\left(\frac{\phi}{2}\right) \leq y\cos\left(\frac{\phi}{2}\right) \leq x\sin\left(\frac{\phi}{2}\right) \\ 0 & \text{otherwise} \end{cases} \tag{3.50}$$

We focus on the derivation of a third order corner detector. Since the 3rd order detectors cannot oscillate much, we set $\mu = 0$. We also get rid of the localization term—to obtain a simple expression, we optimize the detector only with respect to the SNR.

Setting the gradient of $S + \lambda$ Noise to zero (to maximize S subject to Noise $=$ 1), we get

$$2\lambda\,\mathbf{P}\,\mathbf{a} = -\mathbf{s}, \tag{3.51}$$

from which we obtain the optimal solution as

$$\overline{\mathbf{a}} = -\frac{\mathbf{P}^{-1}\mathbf{s}}{\sqrt{\mathbf{s}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{s}}}. \tag{3.52}$$

For a 3rd order detector ($\mathbf{g} = [g_x, g_y, g_{xx}, g_{yy}, g_{xy}]$) and the idealized wedge

model, $\mathbf{P}$ and $\mathbf{s}$ defined by (3.27) and (3.25) are given by

$$\mathbf{P} = \int_{\mathcal{R}^2} \mathbf{g}^T(x,y)\,\mathbf{g}(x,y)\,dxdy \tag{3.53}$$

$$= \begin{bmatrix} \frac{\pi}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{\pi}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{3\pi}{2\sigma^2} & \frac{\pi}{2\sigma^2} & 0 \\ 0 & 0 & \frac{\pi}{2\sigma^2} & \frac{3\pi}{2\sigma^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{\pi}{2\sigma^2} \end{bmatrix} \tag{3.54}$$

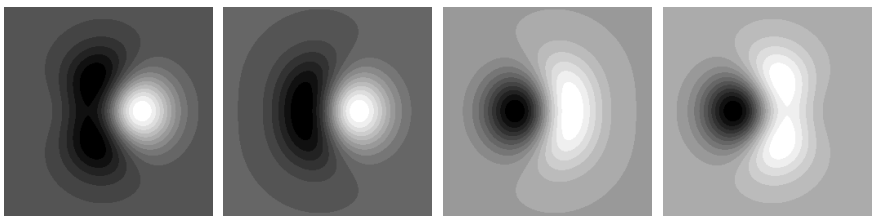
$$\mathbf{s} = \int_{\mathcal{R}^2} f_0(x,y)\,\mathbf{g}(x,y)\,dxdy \tag{3.55}$$

$$= \begin{bmatrix} -\sigma\sqrt{\pi}\sin\left(\frac{\phi}{2}\right) & 0 & \sin(\phi) & -\sin(\phi) & 0 \end{bmatrix}^{\mathrm{T}}. \tag{3.56}$$

Substituting the above in (3.52), we obtain the SNR-optimized 3rd order template as

$$h(\mathbf{x}) = -\sqrt{\frac{2}{2+\pi+2\cos\phi}}\left(g_x + \frac{\sigma\cos\frac{\phi}{2}}{\sqrt{\pi}}\left(g_{xx} - g_{yy}\right)\right) \tag{3.57}$$

It is interesting to note that the optimal corner detector is Canny's edge detector when $\phi = \pi$. Some examples of detectors for different values of ϕ are shown in Fig.3.8.



(a) $\phi = 0.6\pi$ (108°) (b) $\phi = 0.8\pi$ (144°) (c) $\phi = 1.2\pi$ (216°) (d) $\phi = 1.4\pi$ (252°)

Figure 3.8: Wedge Detectors for different wedge angles.

3.5.2 Implementation

We have a template $h_{\theta,\phi}$ which is now parametrized by two variables: θ—the orientation—and ϕ—the wedge angle. Hence the detection procedure involves a two variable optimization. For our experiments, we resort to a slightly suboptimal solution where θ is estimated from the $\phi = \pi$ solution and the optimal ϕ is estimated by sampling. This approach is justifiable as the optimal angle does not change much with respect to ϕ.

3.5.3 Results

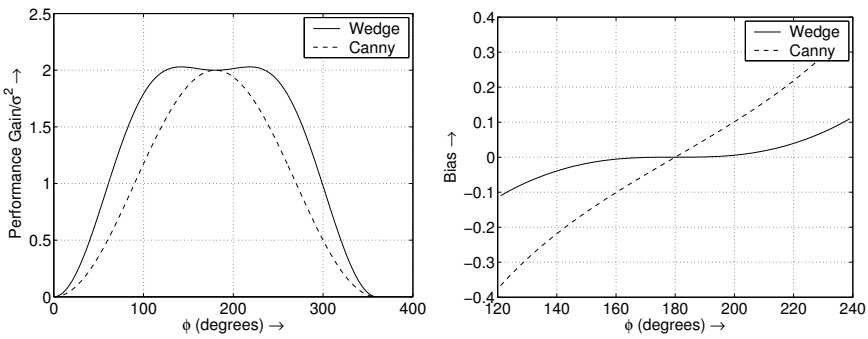


Figure 3.9: (a) S^2/Noise vs wedge angle; (b) measure of $\frac{\overline{y}}{\sigma}$ (ratio of the bias and the standard deviation of the Gaussian window)

We now study the theoretical performance improvement of the wedge detector over Canny's edge detection scheme. We consider the responses of Canny's edge detector and the optimal wedge detector (designed for a specific ϕ) to the wedge. In Fig. 3.9 we show the variation of the SNR with respect to the wedge angle. Note that for Canny's edge detector, the SNR falls off much more rapidly as compared to the wedge detector. The SNR of the wedge detector has a flat zone around $\phi = \pi$ for roughly a span of 140 degrees.

To analyze the bias in the position, we consider the response $r(x, y)$ of the wedge $f_0(x, y)$ (shown in Fig.3.7) to a template $h(x, y)$. The position of the maximum will be displaced from the origin, along the y axis. A first order approximation of the displacement can be obtained by using the Taylor series

expansion of the response $r(x,y) = f_0(x,y) * h(x,y)$ along the y axis.

$$r(0,y) = r(0,0) + r_y(0,0)\,y + \frac{r_{yy}(0,0)}{2}y^2 + \mathcal{O}(y)^3 \tag{3.58}$$

We look for the point $\overline{y}$ such that $r_y(0,\overline{y}) = 0$. From the above expression, we obtain the first order expression of $\overline{y}$ as $r_y(x_0,y_0)/r_{yy}(x_0,y_0)$. Substituting $r = f * h$ and by using the commutativity of convolution and differentiation, we get

$$\overline{y} = -\frac{f * h_y \mid_{0,0}}{f * h_{yy} \mid_{0,0}}$$

The plot of the bias ($\overline{y}$) for different wedge angles is shown in Fig. 3.9-b. It is seen that, for Canny's edge detector, the wedge is displaced from the actual location much more than for the wedge detector tuned to the corresponding angle.

In short, the wedge detector performs better than the edge detector for non-ideal step edges (wedges) for a range of angles; this range can be increased by considering higher order detectors.

To demonstrate the practical utility of the algorithm, we consider the synthetic pattern shown in Fig. 3.10-a and the real image shown in Fig. 3.10-c. We estimate the optimal parameters (θ and ϕ) and the response. We perform non-maximum suppression of the response and keep only the values above a certain threshold. The estimated value of ϕ where the response is greater than the threshold are shown in Fig. 3.10-b and Fig. 3.10-d. Note that the detector can distinguish between convex and concave wedges based on the difference in the estimated angles. The estimated position of the wedge is also a reasonable fit to their true positions. Since Canny's detector is also in the family of wedge detectors, this scheme works well for straight edges as well.

3.6 Summary

We have proposed a general approach to derive optimal 2-D operators for the detection of image features. We chose the optimal template from a family of steerable functions using an analytical optimization scheme based on a slight modification of Canny's criterion. In contrast to classical approaches, where the optimization is performed in 1-D, we specified the filter directly in 2-D. We derived optimal operators for a variety of image features and demonstrated their utility in various applications. We also introduced the notion of shape-adaptable feature detection and used it for the detection of image corners.

We now discuss a few issues that were not dealt with in this chapter and are still open for further investigation.

1. Class of steerable functions: Although we have concentrated on the space of Gaussian derivatives as the steerable family, the design methodology is applicable to other classes as well. Interesting variations may be obtained by changing the window function or using by other known families of steerable functions [60, 45].

2. Discretization: We have derived the optimal operators in continuous space, neglecting discretization issues. It could be interesting to address the discretization effects as in [61] to be closer to practical situations.

Even though further research is required to address these issues, the results presented here are promising enough to justify the use of the proposed detectors in a variety of practical applications. The methodology is also general enough to allow for the design of application-specific templates.

The implementation of the algorithm is available as a Java plugin for ImageJ [62] at http://bigwww.epfl.ch/demo/steerable/.

Appendix 3-A

Proof: Using the linearity of the Fourier transform and the property that differentiation corresponds to a multiplication with $\mathrm{j}\omega$ in the Fourier domain, it is easy to derive the transfer function of the filter h:

$$\hat{h}\left(\omega_x, \omega_y\right) = \sum_{k=1}^{M} \sum_{i=0}^{k} \alpha_{k,i} \left(\mathrm{j}\omega_x\right)^{k-i} \left(\mathrm{j}\omega_y\right)^{i} \ \hat{g}\left(\omega_x, \omega_y\right), \tag{3.59}$$

where $\mathrm{j} = \sqrt{-1}$. Since the rotation of a filter in space corresponds to a rotation of its Fourier transform, we get

$$\begin{aligned}
\Im\left(h\left(\mathbf{R}_\theta\,\mathbf{x}\right)\right) &= \sum_{k=1}^{M}\sum_{i=0}^{k}\alpha_{k,i}\left(\mathrm{j}\omega_x\cos\left(\theta\right)+\mathrm{j}\omega_y\sin\left(\theta\right)\right)^{k-i} \\
&\qquad \left(-\mathrm{j}\omega_x\sin\left(\theta\right)+\mathrm{j}\omega_y\cos\left(\theta\right)\right)^{i}\ \hat{g}\left(\omega_x,\omega_y\right) \\
&= \sum_{k=1}^{M}\sum_{i=0}^{k}\alpha_{k,i}\sum_{l=0}^{k-i}\sum_{m=0}^{i}\binom{k-i}{l}\binom{i}{m}\left(-1\right)^{m}\cos\left(\theta\right)^{i+(l-m)} \\
&\qquad \sin\left(\theta\right)^{(k-l)-(i-m)}\left(\mathrm{j}\omega_x\right)^{l+m}\left(\mathrm{j}\omega_y\right)^{k-(l+m)}\hat{g}\left(\omega_x,\omega_y\right)
\end{aligned} \tag{3.60}$$

Note that the window function is left unchanged because we are assuming that it is isotropic. Now multiplying both sides by $\hat{f}$ and computing the inverse Fourier transform, we get

$$\begin{aligned}
f\left(\mathbf{x}\right)*h\left(\mathbf{R}_\theta\mathbf{x}\right) &= \sum_{k=1}^{M}\sum_{i=0}^{k}\alpha_{k,i}\sum_{l=0}^{k-i}\sum_{m=0}^{i}\binom{k-i}{l}\binom{i}{m}\left(-1\right)^{m}\cos\left(\theta\right)^{i+(l-m)} \\
&\qquad \sin\left(\theta\right)^{(k-i)-(l-m)}f_{k,k-(l+m)}\left(\mathbf{x}\right),
\end{aligned} \tag{3.61}$$

where

$$f_{k,i}\left(\mathbf{x}\right) = f\left(\mathbf{x}\right)*\left(\frac{\partial^{k-i}}{\partial x^{k-i}}\frac{\partial^{i}}{\partial y^{i}}g\left(\mathbf{x}\right)\right)$$

Note that the component indices of f are dependent only on k and $l+m$. We collect the terms with the same values of $k-(l+m)$ and we define $\mathcal{S}\left(k,i,j\right)$ as

$$S\left(k,i,j\right) = \left\{l,m \mid 0<l<k-i; 0<m<i; k-(l+m)=j\right\} \tag{3.62}$$

Using this definition, we rewrite the right hand side of (3.61) as

$$\sum_{k=1}^{M}\sum_{i=0}^{k}\alpha_{k,i}\left(\sum_{j=0}^{k}\sum_{l,m\in\mathcal{S}(k,i,j)}\binom{k-i}{l}\binom{i}{m}\left(-1\right)^{m}\cos\left(\theta\right)^{i+(l-m)}\sin\left(\theta\right)^{(k-i)-(l-m)}f_{k,j}\left(\mathbf{x}\right)\right)$$

$$\sum_{k=1}^{M}\sum_{j=0}^{k}f_{k,j}\left(\mathbf{x}\right)\underbrace{\left(\sum_{i=0}^{k}\alpha_{k,i}\sum_{l,m\in\mathcal{S}(k,i,j)}\binom{k-i}{l}\binom{i}{m}\left(-1\right)^{m}\cos\left(\theta\right)^{i+(l-m)}\sin\left(\theta\right)^{(k-i)-(l-m)}\right)}_{b_{k,j}(\theta)}$$

■

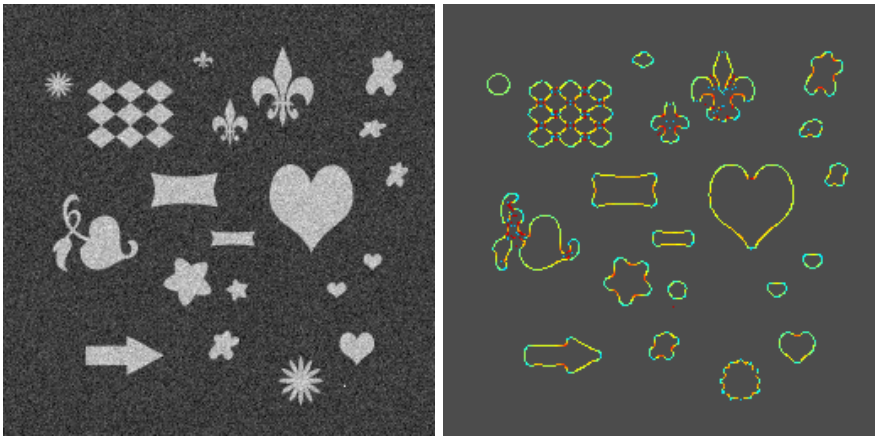
Appendix 3-B

In this section, we briefly outline the steps involved in the 3rd order edge detection algorithm. We denote the 1-D Gaussian of a specified variance, its first, second and third derivatives sampled on a certain grid by g, g', g'' and g''' respectively.

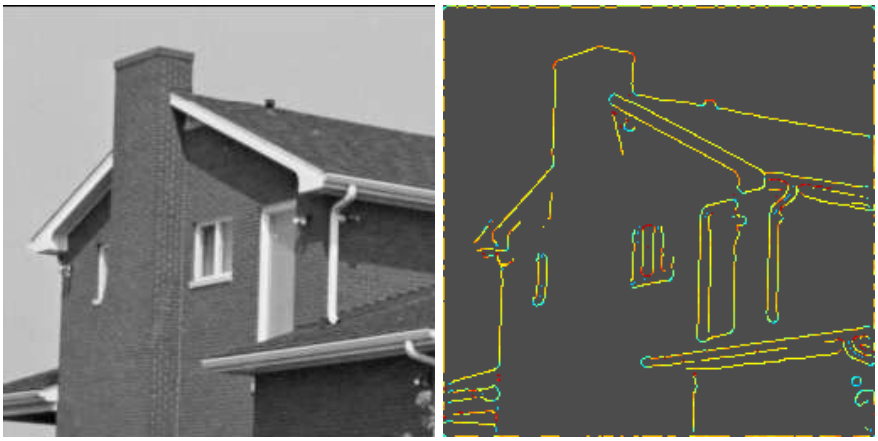
Algorithm

f_{10}= `filterSeparable(image,`g'`,`g`);`
f_{11} = `filterSeparable(image,`g`,`g'`);`
f_{30} = `filterSeparable(image,`g'''`,`g`);`
f_{31} = `filterSeparable(image,`g''`,`g'`);`
f_{32} = `filterSeparable(image,`g'`,`g''`);`
f_{33} = `filterSeparable(image,`g`,`g'''`);`
for `i=0 to Nrows-1` **do**
  for `j=0 to Ncols-1` **do**
    $q_1 = \alpha_{3,0}\, f_{3,0}\,(\mathtt{i},\mathtt{j}) + \alpha_{3,2}\, f_{3,2}\,(\mathtt{i},\mathtt{j}) + \alpha_{1,0}\, f_{1,0}(\mathtt{i},\mathtt{j})$;
    $q_2 = (3\alpha_{3,0} - 2\alpha_{3,2})\, f_{3,1}\,(\mathtt{i},\mathtt{j}) + \alpha_{3,2}\, f_{3,3}\,(\mathtt{i},\mathtt{j}) + \alpha_{1,0}\, f_{1,1}\,(\mathtt{i},\mathtt{j})$;
    $q_3 = (3\alpha_{3,0} - 2\alpha_{3,2})\, f_{3,2}\,(\mathtt{i},\mathtt{j}) + \alpha_{3,2}\, f_{3,0}\,(\mathtt{i},\mathtt{j}) + \alpha_{1,0}\, f_{1,0}\,(\mathtt{i},\mathtt{j})$;
    $q_4 = \alpha_{3,2}\, f_{3,1}\,(\mathtt{i},\mathtt{j}) + \alpha_{3,0}\, f_{3,3}\,(\mathtt{i},\mathtt{j}) + \alpha_{1,0}\, f_{1,1}\,(\mathtt{i},\mathtt{j})$;
    `solset = solveCubic(`q_2, $2q_3 - 3q_1$, $3q_4 - 2q_2$, $-q_3$`);`
    `thetaset = {atan(solset),atan(solset)+`π `};`
    $[\mathtt{optmag}(\mathtt{i},\mathtt{j}), \mathtt{optangle}(\mathtt{i},\mathtt{j})]$ = `giveMaximumRoot(thetaset,` q_1, q_2, q_3, q_4`);`
  end for
end for

The routine `giveMaximumRoot` substitutes the θ values into (3.38); it returns the maximum value and the corresponding angle.



(a) Noisy Image (b) Angle output of Wedge detector



(c) House image (d) Angle output of Wedge detector

Figure 3.10: Detected wedge angle. Here red stands for $\phi = \frac{3\pi}{2}$, yellow for $\phi = \pi$ and cyan for $\phi = \frac{\pi}{2}$. Here the corners are the points which are either in red or in cyan. Note that at the straight edges, the optimal wedge angle is π; the optimal detector is equivalent to the Canny's edge detector. In this experiment, we have chosen $\sigma = 3$.

Chapter 4

Efficient energies and algorithms for parametric snakes

In this chapter[1] we address the first application of the shape estimation algorithm discussed in the introductory chapter (cf. Fig 1.4).

4.1 Introduction

Snakes or active contour models have proven to be very effective tools for image segmentation. An active contour model is essentially a curve that evolves from an initial position towards the boundary of an object in such a way as to minimize some energy functional. The popularity of this semi-automatic approach may be attributed to its ability to aid the segmentation process with a-priori knowledge and user interaction.

Extensive research in this area have resulted in many snake variants [63, 64]; these are distinguished mainly by the type of curve representation used and the choice of the image energy term. The popular curve representation schemes in the snake literature are

[1]Based on the article *M.Jacob, T.Blu, M.Unser, *submitted to IEEE Transactions on Image Processing.*"

- Point-based snakes, where the curve is an ordered collection of discrete points (also termed as snaxels) [8, 65, 66].
- parametric snakes, where the curve is described continuously in a parametric form using basis functions such as B-splines [9, 67, 68, 17], Fourier exponentials [69, 10] etc.
- Geometric snakes, where the planar curve is represented as a level set of an appropriate 2-D surface [20, 70, 18, 71, 72].

The point-based approach can be viewed as a special case of parametric curve representation where the basis functions are uniform translates of a B-spline of degree zero[2]; likewise, parametric approaches using smooth basis functions will tend to the point-based scheme as the number of basis functions increases. In general, however, representations using smooth basis functions require fewer parameters than point-based approaches and thus result in faster optimization algorithms [9, 69, 73]. Moreover, such curve models have inherent regularity and hence do not require extra constraints to ensure smoothness [17, 73].

Since both the above mentioned schemes represent the curve explicitly, it is easy to introduce a-priori shape constraints into the snake framework [69, 74, 75, 76]. It is also straightforward to accommodate user-interaction; this is often done by allowing the user to specify points through which the curve should go through [8]. However, these models offer less flexibility in accounting for topological changes during the curve evolution. One will have to perform some extra bookkeeping to accommodate changes in topology.

Geometric approaches offer great flexibility as far as the curve topology is considered; they presently constitute a very promising research area [18, 71, 72]. However, they tend to be computationally more complex since they evolve a surface rather than a curve. Also, since the curve representation is implicit, it is much more challenging to introduce shape priors into this framework [77].

In this chapter, we focus on general parametric snakes due to its computational advantages and simplicity. We will start by taking a critical look at them, identifying some of their limitations, and propose some improvements to make them more attractive.

There are many different image energy terms that are used in practice. Most of the commonly used approaches fall into two broadly defined categories: (i) edge-based schemes which use local image information (typically gradient information) [9, 69, 73, 78, 17], and, (ii) region-based methods which uses global

[2] A B-spline of degree zero is defined as $\beta^0(x) = \begin{cases} 1 & \text{if } |x| < 0.5 \\ 0 & \text{else} \end{cases}$

image features (eg. statistical formulation) [10, 75, 79, 80, 81]. Since the best choice of the energy depends on the specific application at hand, we try to unify these approaches into a single framework; we obtain a general algorithm which can be tuned easily to the problem.

We propose a new edge energy term which is independent of the parametrization, unlike most of the commonly-used energies. The use of this energy will preserve the parametrization and consequently the curve stiffness. This energy is also more robust than the traditional gradient magnitude-based energy because it accounts for the direction of the gradient as well. We re-express this energy term as a region integral, thus unifying it with the region-based energies in a natural way. Thanks to the new approach, the choice of image energy is reduced to appropriately choosing the preprocessing.

We also clarify some earlier statements about splines by showing that parametric snakes can implicitly ensure smooth curves, but only if they are described in the curvilinear abscissa. Since general curve evolution approaches do not guarantee this configuration, we introduce a new internal energy term which forces the snake to the constant arc-length parametrization. We also propose efficient computational schemes for evaluating the partial differentials of the energy terms; thanks to the parametric curve representation in terms of finitely supported scaling functions, we can compute the differentials exactly and efficiently.

The chapter is organized as follows. In the next section, we provide some mathematical preliminaries and formulate the parametric active contour problem. We deal with the image energy, internal energy and the external constraint energy in Sections 4.3, 4.4 and 4.5 respectively. In Section 4.6, we derive efficient expressions for the partial derivatives of the energy terms. In Section 4.7, we propose a practical solution for the detection and suppression of loops.

4.2 Mathematical Preliminaries

4.2.1 Parametric representation of closed curves

We represent the boundary contour in the $x - y$ plane as a closed parametric curve in a scaling function basis as shown in Section 2.4.

4.2.2 Active contour models: formulation

An active contour, as introduced by Kass et. al. [8], is a curve described as an ordered collection of points which evolves from its initial position to some boundary within the image. The curve evolution is formulated as an energy minimization; the snake energy is typically a linear combination of three terms:

1. the image energy, which is responsible for guiding the snake towards the boundary of interest.

2. the internal energy, which ensures that the segmented region has smooth boundaries.

3. the constraint energy, which provides a means for the user to interact with the snake.

The total energy of the snake is written as

$$E_{\text{snake}}(\boldsymbol{\Theta}) = E_{\text{image}}(\boldsymbol{\Theta}) + E_{\text{int}}(\boldsymbol{\Theta}) + E_{\text{c}}(\boldsymbol{\Theta}), \tag{4.1}$$

where $\boldsymbol{\Theta}$ is the collection of curve coefficients $\boldsymbol{\Theta} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{M-1})$. The optimal curve parameters are obtained as

$$\overline{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} E_{\text{snake}}(\boldsymbol{\Theta}) \tag{4.2}$$

It is obvious that the quality of segmentation is dependent on the choice of the energy terms. We deal with them in detail in the following sections and are listed in Table. 4.1 for easy reference. The energy minimization process is essentially an optimization procedure, where we iteratively update the snake coefficients so as to reach the minimum of the cost/energy function.

4.3 Image energy

The image energy is the most important of the three energy terms. In this section, we identify some limitations of the widely-used gradient magnitude energy and propose a new cost function that overcomes these problems. We also present a unified framework which includes the edge-based and region-based approaches as particular cases.

4.3.1 Edge-based image energy

Traditional snakes rely on edge maps derived from the image to be guided to the actual contour. The most popular approach is based on the magnitude of the gradient.

Gradient magnitude energy

Many of the parametric snakes described in the literature use the integral of the square of the gradient magnitude along the curve as the image energy [17, 9, 69, 73]. Mathematically, we have

$$E_{\text{mag}} = -\int_0^M |\nabla f(t)|^2 \, dt, \tag{4.3}$$

where $\nabla f(t)$ denotes the gradient of f at the point $\mathbf{r}(t)$. As pointed out in [78], one disadvantage of this measure is that it does not use the direction of the gradient. At the boundary, the image gradient is perpendicular to the contour. This extra information can be incorporated into the external energy to make it more robust.

We have seen in the previous chapter that computing the magnitude of the gradient is equivalent to solving for the optimal orientation and then computing the optimal response. Hence using the gradient magnitude in the image energy is equivalent to performing the edge detection independently at each pixel and using the detected edges to drive the snake. This two-step approach may be less consistent; for instance, the direction of the gradient at a particular pixel on the contour need not be the same as tangent to the curve at that point.

Another problem is the dependence of E_{mag} on the parametrization; we obtain a different value of E_{mag} if the curve is represented in terms of a parameter $t' = w(t)$, where w is a monotonically increasing one to one warping function. The use of such an energy may therefore result in the curve re-adjusting its parametrization in trying to maximize E_{mag} (e.g. with B-spline curves, the knots will move to regions of the contour where the gradient magnitude is relatively high). This problem is demonstrated in Fig. 4.2-b.

New gradient-based image energy

The gradient magnitude energy is the integral of a scalar field derived from the gradient vector field. We propose a new energy that uses the vector field

directly:

$$E_{\text{grad}} = -\oint_{\mathcal{C}} \mathbf{k} \cdot (\nabla f(\mathbf{r}) \times d\mathbf{r}) \tag{4.4}$$

$$= -\oint_{\mathcal{C}} \nabla f(\mathbf{r}) \cdot \underbrace{(d\mathbf{r} \times \mathbf{k})}_{||d\mathbf{r}||\,\hat{\mathbf{n}}(\mathbf{r})}, \tag{4.5}$$

where $\mathbf{k}$ is the unit vector orthogonal to the image plane. Here $\hat{\mathbf{n}}(\mathbf{r})$ denotes the unit normal to the curve at $\mathbf{r}$. Since we are evaluating the likeliness of an edge oriented along the tangents at the curve points as compared to the two step strategy used in conventional schemes, we expect our algorithm to be more robust and consistent to the image data. Note that this approach of accounting for the gradient direction is similar in philosophy to [78], eventhough the expression used by these authors is different and parameter dependent.

This integration process is illustrated in Fig. 4.1; with our convention, the vector $\hat{\mathbf{n}}(\mathbf{r})$ is the inward unit normal to the curve[3] meaning that we are integrating the component of the gradient orthogonal to the curve. Note that (4.4) is independent of the parameter t, and hence does not depend on the parametrization. The improvement obtained by using the new energy instead of the parameter dependent magnitude-based energy is shown in Fig. 4.2-c.

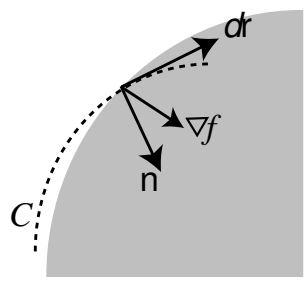


Figure 4.1: Gradient and normal to the curve

General edge-based image energy

We consider a generalized form of (4.4) by substituting ∇f with other feature-enhancing vector fields. A promising approach is the use of optimal steerable

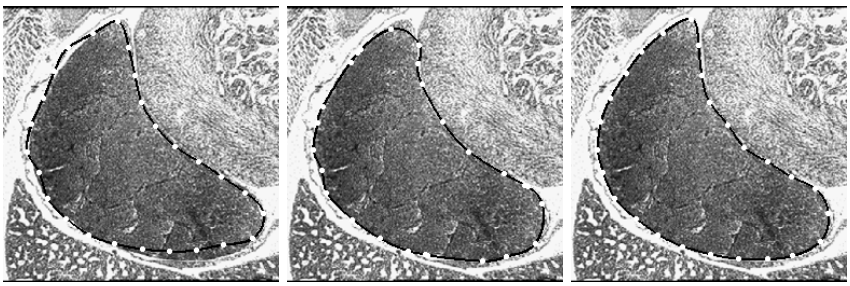
[3] $\mathbf{k}$ is chosen, depending on the direction in which the curve is described, so that $\hat{\mathbf{n}}(\mathbf{r}) = \frac{d\mathbf{r}\times\mathbf{k}}{||d\mathbf{r}||}$ is the inward unit normal.

filters to derive an appropriate edge enhancing vector field[82]. This method uses filters that are more directional than the x and y components of the conventional the gradient operator to derive a noise-resilent field.

The general form of edge-based image energy can be expressed mathematically as

$$E_{\text{edge}} = -\oint_{\mathcal{C}} \mathbf{k} \cdot \left(\mathbf{e}_f\left(\mathbf{r}\right) \times d\mathbf{r}\right), \tag{4.6}$$

where $\mathbf{e}_f$ is an appropriate vector field derived from f. The magnitude of $\mathbf{e}_f\left(\mathbf{r}\right)$ gives a measure of the edge strength at $\mathbf{r}$, while its direction specifies the edge orientation. We now show that the computation of this edge-based energy is equivalent to evaluating a region integral.

[h!tb] 

(a) Initialization (b) Magnitude-based energy (c) New edge-based energy

Figure 4.2: Segmentation of a mouse organ using edge-based energy (a) The knots (denoted by the white dots) are initialized so that the curve is approximately in the curvilinear abscissa. (b) Curve evolution based on the gradient magnitude-based energy. Note that the knots accumulate at some points along the curve in the final curve, thus restricting the flexibility of the curve. (c) Curve evolution based on our new edge-based energy; by better preserving the parametrization, it often result in a better segmentation.

Proposition 2 *The general edge-based image energy (4.6) can also be ex-*

pressed as

$$E_{\text{edge}} = \int_{\mathcal{S}} \underbrace{\nabla \cdot \mathbf{e}_f(\mathbf{s})}_{T_e(f)} d\mathbf{s}, \tag{4.7}$$

where $\nabla \cdot \mathbf{e}_f$ *denotes the divergence of the vector field* $\mathbf{e}_f$.

Proof: Green's theorem relates the volume integral of the divergence of a 3-D vector field $\mathbf{F}$ over a closed volume $\mathcal{V}$ bounded by the surface $\mathcal{S}$ to its integral over $\mathcal{S}$:

$$\int_{\mathcal{V}} (\nabla \cdot \mathbf{F})\, d\mathbf{v} = \int_{\mathcal{S}} \mathbf{F} \cdot d\mathbf{s}. \tag{4.8}$$

The restriction of Green's theorem to two dimensional space yields

$$\int_{\mathcal{S}} \left(\frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} \right) dxdy = \oint_{\mathcal{C}} (\mathbf{F}_y dx - \mathbf{F}_x dy) \tag{4.9}$$

The integral on the left is computed over the area $\mathcal{S}$ bounded by the curve $\mathcal{C}$ while the one on the right is over $\mathcal{C}$. Using the vector notation, we rewrite (4.9) as

$$\int_{\mathcal{S}} (\nabla \cdot \mathbf{F})\, d\mathbf{s} = -\oint_{\mathcal{C}} \mathbf{k} \cdot (\mathbf{F} \times d\mathbf{r}), \tag{4.10}$$

where $\mathbf{k}$ is the unit normal to the two dimensional space. Using this identity, we simplify (4.6) to the form (4.7). ■

Note that in the special case when $\mathbf{e}_f = \nabla f$, we get $T_e(f) = \nabla^2 f$. This means that our new gradient-based energy (4.4) is equivalent to integrating the Laplacian of the image in the region bounded by the curve.

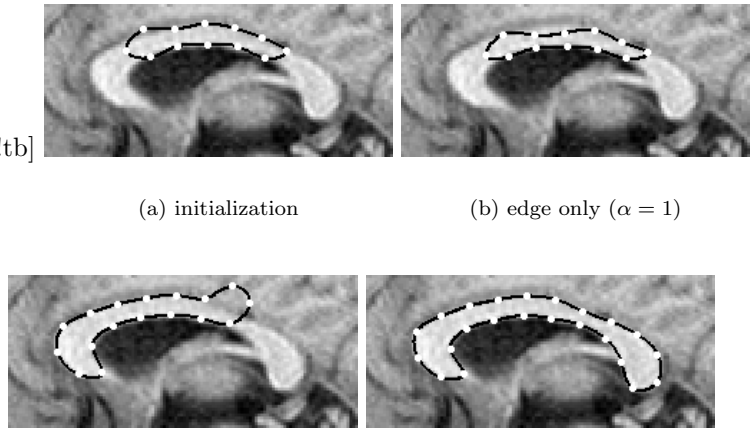
4.3.2 Region-based image energy

Recent research in active contours is increasingly focusing on the use of statistical region-based image energy [10, 75, 79, 80]. This type of energy can provide the snake with vital boundary information, especially while it is far away from the real contour, thus resulting in a larger basin of attraction.

The use of this energy assumes two main regions in image[4], with different probability distributions. A simple example is the case where we have to segment a white object from a dark background; the regions will have different means

[4]This can be generalized to $n > 2$ regions

[h!tb]



(a) initialization (b) edge only ($\alpha = 1$)

(c) region only ($\alpha = 0$) (d) unified ($\alpha = 0.5$)

Figure 4.3: Illustration of the use of the unified image energy in the segmentation of a corpus-callossum image (b) The use of the gradient based energy fails to converge in regions where the gradient information is absent (c) The region-based energy is misled by the lack of image contrast (d) The unified energy leads to a good segmentation.

and possibly different variances. We use the statistical formulation of Staib et. al. [69] to specify the region likelihood function:

$$\begin{aligned} E_{\text{region}} &= -\int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)\right) d\mathbf{s} \\ &\quad -\int_{\mathcal{S}'} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)\right) d\mathbf{s}, \end{aligned} \tag{4.11}$$

where $\mathcal{R}$ and $\mathcal{R}'$ denote the different image regions. We denote the regions in the curve and outside by $\mathcal{S}$ and $\mathcal{S}'$ respectively. It is easy to see that (4.11)

attains a maximum when $\mathcal{R} = \mathcal{S}$ and $\mathcal{R}' = \mathcal{S}'$. We rewrite the above integral as

$$\begin{aligned} E_{\text{region}} &= -\int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)\right) d\mathbf{s} \\ &\quad -C + \int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)\right) d\mathbf{s}, \end{aligned} \tag{4.12}$$

where $C = \int_{\mathcal{S}' \bigcup \mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)\right) d\mathbf{s}$. Since C does not depend on the position of the curve and hence we remove it from the cost function. Thus, the region-based cost function is simplified to

$$E_{\text{region}} = \int_{\mathcal{S}} \underbrace{-\log\left(\frac{\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)}{\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)}\right)}_{T_r(f)} d\mathbf{s} \tag{4.13}$$

We now give a few examples to illustrate (4.13).

1. The regions $\mathcal{R}$ and $\mathcal{R}'$ have Gaussian distributions with the same variance. In this case, we obtain

$$T_r\left(f\right) = -\frac{2\left(\mu_{\mathcal{R}} - \mu_{\mathcal{R}'}\right)}{\sigma^2}\left(f - \underbrace{\frac{\mu_{\mathcal{R}} + \mu_{\mathcal{R}'}}{2}}_{\mu_{\mathcal{R},\mathcal{R}'}}\right), \tag{4.14}$$

   where $\mu_{\mathcal{R}} > \mu_{\mathcal{R}'}$ are the means of the regions $\mathcal{R}$ and $\mathcal{R}'$ and and σ the standard deviation. The regions of f with values above $\mu_{\mathcal{R},\mathcal{R}'}$ are mapped to negative values while the ones below are assigned positive values. Hence, evolving the contour using (4.13) will result in the curve adjusting itself to have regions of f above $\mu_{\mathcal{R},\mathcal{R}'}$ inside while excluding the ones below $\mu_{\mathcal{R},\mathcal{R}'}$. The assumption of the variances of the regions being the same is appropriate if we have piecewise constant images corrupted by additive Gaussian noise.

2. The regions inside and outside the contour have Gaussian distributions with different variances. In this case, we obtain

$$T_r\left(f\right) = a\,f^2 + b\,f + c, \tag{4.15}$$

   where $a = \left(\frac{1}{\sigma^2_{\mathcal{R}'}} - \frac{1}{\sigma^2_{\mathcal{R}}}\right)$, $b = -2\left(\frac{\mu_{\mathcal{R}'}}{\sigma^2_{\mathcal{R}'}} - \frac{\mu_{\mathcal{R}}}{\sigma^2_{\mathcal{R}}}\right)$ and $c = \left(\frac{\mu^2_{\mathcal{R}'}}{\sigma^2_{\mathcal{R}'}} - \frac{\mu^2_{\mathcal{R}}}{\sigma^2_{\mathcal{R}}}\right) + \log\left(\frac{\sigma_{\mathcal{R}'}}{\sigma_{\mathcal{R}}}\right)$. Here, $\sigma_{\mathcal{R}}$ and $\sigma_{\mathcal{R}'}$ are the standard deviations of the regions

inside and outside the curve respectively. Since the snake uses the information from the variances as well, it can resolve the boundaries even when both regions have identical means but different variances [68].

In the absence of prior knowledge of the probability distributions $\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}\right)$ and
$\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)$, the statistical parameters are estimated from the image f themselves as the snake evolves; we assume the current position of the contour to define the regions (i.e. $\mathcal{S}=\mathcal{R}$ and $\mathcal{S}'=\mathcal{R}'$) and estimate the parameters as discussed in Section 4.6.3.

The extension of this method for the segmentation of multi-component images (e.g color images) is straightforward. For a n-D vector image $\mathbf{f}:\mathbb{R}^2\rightarrow\mathbb{R}^n$, we have

$$E_{\text{region}}=\int_{\mathcal{S}}\underbrace{-\log\left(\frac{\mathrm{P}\left(\mathbf{f}\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}\right)}{\mathrm{P}\left(\mathbf{f}\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)}\right)}_{T_r(\mathbf{f})}\,d\mathbf{s} \tag{4.16}$$

Note that the region information from the vector data is efficiently concatenated into the scalar image $T_r\left(\mathbf{f}\right)$. This framework is used for the segmentation of textures in [83]. They obtain an appropriate vector image from the gray level image using a Gabor filterbank.

4.3.3 Unified image energy

Both of the above mentioned energies (edge-based and region-based) have their own advantages and disadvantages. The edge-based energy can give a good localization of the contour near the boundaries. Unfortunately, it has a small basin of attraction, thus requiring a good initialization or a baloon force [84]. On the other hand, the region-based energy have a large basin of attraction and can converge even if explicit edges are not present [80]. However, it does not give as good a localization as the edge-based energy at the image boundaries. Motivated by the complementary features of these schemes and the similarity of the expressions (4.7) and (4.11), we propose a unified form of image energy. We choose a convex combination of the two energies to obtain an extended class, which inherits the advantages of both. The new image energy is given by

$$E_{\text{image}}=\int_{\mathcal{S}}\underbrace{T_{\mathrm{u}}\left(f\right)}_{f_{\mathrm{u}}}\left(\mathbf{s}\right)d\mathbf{s}, \tag{4.17}$$

where $f_{\mathrm{u}} = T_u(f) = \alpha T_e(f) + (1-\alpha) T_r(f)$. This unification is similar is philosophy to the approaches in [10, 71]. However, our scheme is more natural and yields a simpler expression since it combines the two energies into a single region integral. The simplicity of the expression will lead to computational advantages as will be discussed later on. We demonstrate the use of the unified energy in Fig. 4.3.

4.4 Internal Energy

The internal energy is responsible for ensuring the smoothness of the contour. Kass proposed an internal energy the linear combination of the length of the contour and the integral of the square of the curvature along the contour. This smoothness term is the most widely used one in applications [8, 63, 66]. Its direct extension to parametric curves gives

$$\begin{aligned} E_{\mathrm{int}} &= \lambda_1 \underbrace{\int_0^M \left(x'(t)^2 + y'(t)^2\right)^{\frac{1}{2}} dt}_{\text{Length}} + \\ & \lambda_2 \int_0^M \underbrace{\left(\frac{x''(t)\, y'(t) - y''(t)\, x'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{\frac{3}{2}}}\right)^2}_{|\kappa(\mathbf{r})|^2} dt, \end{aligned} \tag{4.18}$$

where $\kappa(\mathbf{r})$ is the curvature of the curve at the point $\mathbf{r}(t)$. The first integral in (4.18) can be computed, while the second one is more complicated. We show in the Appendix 4-A that the second term reduces to

$$\int_0^M |\kappa(\mathbf{r})|^2 dt = \frac{1}{c^2} \int_0^M \underbrace{\left(|x''(t)|^2 + |y''(t)|^2\right)}_{|\mathbf{r}''(t)|^2} dt \tag{4.19}$$

provided that

$$|x'(t)|^2 + |y'(t)|^2 = c; \quad \forall t. \tag{4.20}$$

that is, when the curve is parametrized by its curvilinear abscissa. Here

$$c = \frac{1}{M}\left(\underbrace{\int_0^M \left(x'(t)^2 + y'(t)^2\right)^{\frac{1}{2}} dt}_{\text{Length}}\right)^2 \tag{4.21}$$

is the total length/unit value of the parameter. It is justified to use $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ as the curvature term in point-based snakes since the snake points (snaxels) are almost equally spaced. For parametric snakes described in the curvilinear abscissa, the curvature term is inversely proportional to the fourth power of the distance between the knots along the curve (c.f. (4.19) and (4.20)). We will have a smooth curve if its knots are well separated.

Most parametric schemes rely on the smoothness of the representation, thus eliminating the need for an explicit internal energy term [10, 69, 9, 17, 73]. However, these approaches can only ensure a low value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$; they can guarantee low curvature curves only when (4.20) hold. For example, a spline curve may be rough even with a small value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ if some of the spline knots accumulate at one section of the curve. Similar problems exist with Fourier and other parametric representations. To counter this problem, we propose to add a new term to the criterion that will force the snake to satisfy (4.20).

4.4.1 Curvilinear reparametrization energy

Our new energy term that penalizes the curve for not being in the curvilinear abscissa is given by

$$E_{\text{curv}} = \int_0^M \left| |\mathbf{r}'(t)|^2 - c \right|^2 dt, \tag{4.22}$$

where c is given by (4.21). Evolving the curve with such a term will cause the curve knots to move tangential to the curve, thus bringing it to the curvilinear abscissa. An example of the type of improvement that can be obtained in this way is shown in Fig. 4.4.

Precioso et. al [85] proposed to reparametrize the curve to the constant arc-length representation after each step of the optimization algorithm to avoid the curves from looping. This scheme would yield the same results as our approach, but is computationally much more expensive.

4.4.2 Choice of the scaling basis function

As mentioned before, the parametric representations can guarantee a small value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$. Using the well-known variational properties of splines [28], we can show that the minimization of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ subject to interpolation constraints yields a cubic spline curve with knots at the integers. Thus, the cubic B-spline model appears to be the most natural choice for representing parametric curves; it will yield a minimum curvature curve provided the parametrization is the curvilinear abscissa (i.e. the knots are uniformly spaced on the curve). The use of spline curves also brings in additional gains due to the existence of efficient algorithms [27], the local control of the contour due to the finite support of the B-spline basis function and their good approximation properties [26].

Due to these nice properties, we choose cubic spline curves in our implementation. However, the theory we present in this chapter is general enough to accommodate for any other representation in terms of scaling function or wavelets.

4.4.3 New internal energy term

If we choose $c = \gamma \frac{\text{Length}}{M}$ in (4.22) instead of (4.21), we get

$$E_{\text{curv}}(\gamma) = E_{\text{curv}} + \frac{\left(1-\gamma^2\right)\text{Length}^2}{M} \tag{4.23}$$

This equation implies that we can also account for the Length term in (4.18) by choosing $\gamma < 1$. We thus simplify the internal energy to

$$E_{\text{int}} = E_{\text{curv}}(\gamma) \tag{4.24}$$

In practice, we found it better not to minimize the length of the curve under normal circumstances; in other words, we usually set $\gamma = 1$. However, when the curve is detected to be looping, we decrease the length of the curve by choosing $\gamma = 0$. We discuss this issue in Section 4.7.2.

4.5 External constraint energy

As mentioned before, external constraint energy provides a means for the user to interact with the snake; he can guide the snake to the boundary when image information is too weak or ambiguous.

We introduce a point constraint mode, where the user has the option to specify a few points that should lie on the contour to be detected. We constrain the snake by adding an energy term which is the distance between these points and the corresponding closest points on the curve. The constraint energy is given by

$$E_{\mathrm{c}} = \sum_{i=0}^{N_{\mathrm{c}}-1} \min_{t \in [0,M]} \left| \mathbf{r}(t) - \mathbf{r}_{\mathrm{c},i} \right|^2 , \tag{4.25}$$

where $\mathbf{r}_{\mathrm{c},i}; i = 0..N_{\mathrm{c}} - 1$ are the constraints. This approach can be thought of as introducing virtual springs that pulls the curve towards the desired points: One end of the spring is fixed to the constraint point while the other end slides on the curve.

4.6 Evaluation of the partial derivatives

In this section, we express the partial derivatives of the component energies of the snake. These are used by the optimization algorithm to converge to the minimum of the energy function.

The theory mentioned so far is valid for general scaling function representations ranging from band-limited curves (Fourier series representation) to polygons. In order to derive efficient numerical schemes, we now make the additional assumption that the basis function is finitely supported in the interval $[0, N]$. Note that this class is still very rich as it includes most of the known scaling function families. The interesting cases for our purpose are the cubic B-spline function, which is finitely supported in the interval $[0, 4]$, and the linear B-spline function with the support $[0, 2]$.

Partial derivatives of the magnitude-based image energy

Following the work of Flickner et. al. [73], we locally optimize the snake during the initialization process (when the user is in the process of entering the initial curve), thus providing the user with a visual feedback. For this optimization we use the simple gradient magnitude-based energy mainly because it is applicable even when the curve is not yet closed and also because it is simple and computationally efficient. However, we only perform few iterations with this energy as it tends to bring the curve knots closer as mentioned before (c.f. Fig. 4.2).

We consider the integral in (4.3) and differentiate it with respect to the

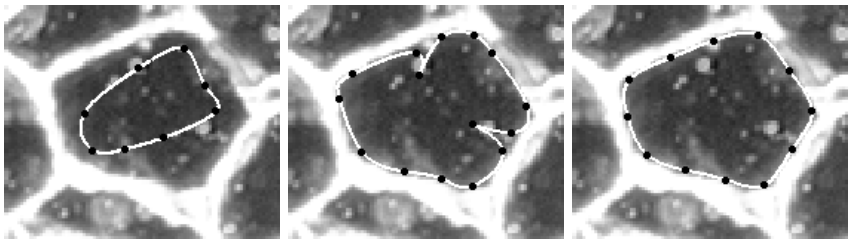
coefficients using the chain rule (using (2.31)):

$$\begin{bmatrix} \partial E_{\text{mag}}/\partial c_{x,k} \\ \partial E_{\text{mag}}/\partial c_{y,k} \end{bmatrix} = \int_0^M \nabla g\,(t)\;\varphi_p\,(t-k)\,dt \tag{4.26}$$

where $g = |\nabla f|^2$. We approximate the inner-product as a discrete sum:

$$\begin{bmatrix} \partial E_{\text{mag}}/\partial c_{x,k} \\ \partial E_{\text{mag}}/\partial c_{y,k} \end{bmatrix} \approx \frac{1}{R}\sum_{i=0}^{NR} \nabla g\left(\frac{[k\,R+i]_{MR}}{R}\right)\varphi\left(\frac{i}{R}\right), \tag{4.27}$$

where R is the sampling rate and $[k]_M$ stands for k mod M. In the above expression, we used the finite support of the scaling function to limit the range of the summation. Also note that we have transferred the periodicity from the kernel to ∇g; this means that the summation is evaluated assuming periodic boundary conditions on ∇g. Thus, if ∇g and $\varphi\left(\frac{i}{s}-k\right)$ are precomputed, the evaluation of the partial derivatives just involves a weighted sum. The computational complexity is therefore proportional to sMN.



(a) Initialization (b) No curvilinear energy (c) With curvilinear energy

Figure 4.4: Without the curvilinear energy, the parametric representation cannot guarantee low curvature curves. Note that for the same initialization, the curve with the curvilinear reparametrization energy leads to smoother curves. Without the energy, the curve knots accumulate at some regions of the curve, thus leading to sharp edges; low energy curves are ensured only if the arc length is constant on the curve.

Partial derivatives of the unified image energy

For closed curves, we preferentially use the unified energy to optimize the curve. In line with the work of [69, 10, 21], we now use Green's theorem (4.9) to convert region integrals (over the region bounded by a closed curve) to integrals over the curve; our main motivation is computational efficiency. (4.17) can be efficiently computed as the curve integral

$$\int_{\mathcal{S}} f_{\mathrm{u}}(x,y)\,dxdy = \oint_{\mathcal{C}} f_{\mathrm{u}}^{y}(x,y)\,dx \tag{4.28}$$

$$= -\oint_{\mathcal{C}} f_{\mathrm{u}}^{x}(x,y)\,dy, \tag{4.29}$$

where

$$f_{\mathrm{u}}^{y}(x,y) = \int_{-\infty}^{y} f_{\mathrm{u}}(x,\tau)\,d\tau \tag{4.30}$$

$$f_{\mathrm{u}}^{x}(x,y) = \int_{-\infty}^{x} f_{\mathrm{u}}(\tau,y)\,d\tau \tag{4.31}$$

Applying the chain rule of differentiation on (4.29), we obtain $\partial E_{\mathrm{image}}/\partial c_{x,k}$ as

$$\begin{aligned}
\frac{\partial}{\partial c_{x,k}}(E_{\mathrm{image}}) &= \frac{\partial}{\partial x}(E_{\mathrm{image}}) \cdot \frac{\partial}{\partial c_{x,k}}(x(t)) \\
&= -\int_{0}^{M} \underbrace{\frac{\partial f_{\mathrm{u}}^{x}}{\partial x}}_{f_{\mathrm{u}}} \varphi_p(t-k) \underbrace{\sum_{l=0}^{M} c_{y,l}\,\varphi_p{}'(t-l)}_{y'(t)} \\
&= -\sum_{l=0}^{M} c_{y,l} \int_{0}^{M} f_{\mathrm{u}}(t)\,\varphi_p(t-k)\,\varphi_p{}'(t-l)\,dt \\
&= -\sum_{l=-\infty}^{\infty} c_{y,l}^{p} \underbrace{\int_{-\infty}^{\infty} f_{\mathrm{u}}(t)\,\varphi(t-k)\,\varphi'(t-l)\,dt}_{Q_{f_{\mathrm{u}}}(k,l)}
\end{aligned} \tag{4.32}$$

In the last step we expanded $\varphi_p(t-k)$ using (2.33) and made a change of variable, thus extending the integral from $-\infty$ to ∞. We also transferred the periodicity of $Q_{f_{\mathrm{u}}}$ to the coefficient sequence. Since $Q_{f_{\mathrm{u}}}(k,l)$ is a finite sequence, the evaluation of (4.32) amounts to an appropriate finite sum. In a similar

manner, using (4.28) we obtain

$$\frac{\partial E_{\text{image}}}{\partial c_{y,k}} = \sum_{l=-\infty}^{\infty} c_{x,l}^{p} \underbrace{\int_{-\infty}^{\infty} f_{\text{u}}(t)\,\varphi(t-k)\,\varphi'(t-l)\,dt}_{Q_{f_{\text{u}}}(k,l)}$$

The main steps in the computation of the partial derivatives are:

1. The evaluation of the sequence $Q_{f_{\text{u}}}(k,l)\,;\ \ |k-l|<N$. (With a change of variables we obtain $Q_{f_{\text{u}}}(k,l)=\int_{-\infty}^{\infty} f_{\text{u}}(t+k)\,\varphi(t)\,\varphi'(t+k-l)\,dt$. Since $\varphi(t)$ is finitely supported in the interval $[0,N]$, $Q_{f_{\text{u}}}(k,l)$ is zero if $|k-l|\geq N$). Approximating the integral as a discrete sum, we obtain

$$Q_{f_{\text{u}}}(k,l)=\frac{1}{R}\sum_{i=0}^{NR} f_{\text{u}}\left(\frac{[kR+i]_M}{R}\right)\underbrace{\varphi\left(\frac{i}{R}\right)\varphi'\left(\frac{i}{R}+k-l\right)}_{b_{k-l}(i)} \tag{4.33}$$

   Provided we precompute[5] the sequence $b_m(i)\,;m=\{-N+1\ldots N-1\}$, the computation of $Q_{f_{\text{u}}}(k,l)\,;0<k,l<M$ involves an weighted sum of length Ns.

2. The evaluation of the partial derivatives, which are obtained as

$$\begin{bmatrix} \partial E_{\text{u}}/\partial c_{x,k} \\ \partial E_{\text{u}}/\partial c_{y,k} \end{bmatrix} = \sum_{l=0}^{M-1} \begin{bmatrix} -c_{y,l}^{p} \\ c_{x,l}^{p} \end{bmatrix} Q_{f_{\text{u}}}(k,l) \tag{4.34}$$

Here, the computational complexity of the order of $s\,M^2N^2$. Note that there is a factor of 2 advantage in implementing the partial derivatives of E_{grad} as in (4.34) rather than its direct evaluation from (4.4). The performance improvement in the implementation of the unified energy is even better as compared to the one in [10], where the energy is the sum of two integrals.

4.6.1 Partial derivatives of the internal energy

Differentiating the expression of $E_{\text{int}}=E_{\text{curv}}$ and simplifying further, we obtain the partial derivatives as simple multidimensional filtering of the scaling function

[5] The samples of φ can be computed by solving for its values at the integers as shown in [22] and using the two-scale relation to refine it .

coefficients. We show in the appendix that the partial derivatives of the the term E_{int} can be computed as

$$\begin{aligned}
\frac{\partial}{\partial c_{x,k}}\left(E_{\text{int}}\right) &= \sum_{|l|,|m|,|n|<N} c^p_{x,k-l}\, c^p_{x,k-m}\, c^p_{x,k-n}\, h_1\left(l,m,n\right) + \\
&\quad \sum_{|l|,|m|,|n|<N} c^p_{x,k-l}\, c^p_{y,k-m}\, c^p_{y,k-n}\, h_1\left(l,m,n\right) + \\
&\quad c \sum_{|l|<N} c^p_{x,k-l}\, h_2\left(l\right) \qquad (4.35) \\
\frac{\partial}{\partial c_{y,k}}\left(E_{\text{int}}\right) &= \sum_{|l|,|m|,|n|<N} c^p_{y,k-l}\, c^p_{y,k-m}\, c^p_{y,k-n}\, h_1\left(l,m,n\right) + \\
&\quad \sum_{|l|,|m|,|n|<N} c^p_{y,k-l}\, c^p_{x,k-m}\, c^p_{x,k-n}\, h_1\left(l,m,n\right) + \\
&\quad c \sum_{|l|<N} c^p_{y,k-l}\, h_2\left(l\right)
\end{aligned}$$

where

$$h_1\left(l,m,n\right) = \int_{-\infty}^{\infty} \varphi'\left(t\right)\varphi'\left(t+l\right)\varphi'\left(t+m\right)\varphi'\left(t+n\right)dt \qquad (4.36)$$

$$h_2\left(l\right) = \int_{-\infty}^{\infty} \varphi'\left(t\right)\varphi'\left(t+l\right)dt \qquad (4.37)$$

Note that the multidimensional filtering is performed assuming periodic boundary conditions. The computational complexity is small, since the sum depends only on the coefficient sequence whose number is typically much lesser than the number of curve samples. The computational complexity in evaluating the above sum is N^3M. The filter coefficients (4.36) and (4.37) are precomputed as shown in Appendix 4-C.

4.6.2 Partial derivatives of the constraint energy

Computing the partial derivatives of (4.25), in all its generality, would give a very complicated expression. To make the problem more tractable and to reduce its computational complexity, we make the assumption that the optimal

parameters $t_i;\ i = 0 \ldots N_\mathrm{c} - 1$ are known. In this case, (4.25) gets simplified to

$$E_\mathrm{c} = \sum_{i=0}^{N_\mathrm{c}-1} \left|\mathbf{r}\left(t_i\right) - \mathbf{r}_{\mathrm{c},i}\right|^2 \tag{4.38}$$

and its partial derivatives are given by:

$$\begin{bmatrix} \partial E_\mathrm{c}/\partial c_{x,k} \\ \partial E_\mathrm{c}/\partial c_{y,k} \end{bmatrix} = \sum_{i=0}^{N_\mathrm{c}-1} \left(\begin{bmatrix} x_{\mathrm{c},i} \\ y_{\mathrm{c},i} \end{bmatrix} - \begin{bmatrix} x\left(t_i\right) \\ y\left(t_i\right) \end{bmatrix} \right) \varphi\left(t_i - k\right) \tag{4.39}$$

Using the finite support of the scaling functions, we limit the sum to the relevant indices (we need to evaluate it only for $\{i|\ 0 < (t_i - k) < N_\mathrm{c}\}$). In practice, we resort to a two-step optimization where the snake is first evolved using the above formulas for the derivatives with the current set of t_i's. The optimal parameters t_i are then re-estimated within the loop as:

$$t_i = \arg \min_{t \in [0,M]} \left|\mathbf{r}\left(t\right) - \mathbf{r}_{\mathrm{c},i}\right|;\ \ i = 0 \ldots N_\mathrm{c} - 1 \tag{4.40}$$

4.6.3 Estimation of the probability distribution functions

The evaluation of (4.11) requires the specification of the probability distribution functions $P\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)$ and $P\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)$. If we do not have any a-priori knowledge of these distributions, these are estimated iteratively from the image data itself assuming $\mathcal{R} = \mathcal{S}$ and $\mathcal{R}' = \mathcal{S}'$. Note that these assumptions are valid if the snake is close to the real boundary. We use densities such as the Gaussian distribution which are represented by few parameters (mean and variance). The estimation of these parameters require integrating the image and its square in the region bounded by $\mathcal{S}$. We compute the integrals efficiently using (4.28) with the corresponding integrated functions (similar to (4.30)) precomputed.

The estimation of the distributions are followed a non-linear transformation which maps f into $f_\mathbf{u}$. Since this transformation is time consuming, the estimation of the distributions and the updating of $f_\mathbf{u}$ is only performed periodically, typically once every 10 iterations.

4.6.4 Computation of the length and area

The computation of the internal energy requires the estimation of the current length of the curve. We compute the length as a discrete approximation of the

integral

$$\int_0^M \left(x'(t)^2 + y'(t)^2\right)^{\frac{1}{2}} dt$$

as

$$\text{Length} = \frac{1}{R} \sum_{i=0}^{MR-1} \sqrt{x'\left(\frac{i}{R}\right)^2 + y'\left(\frac{i}{R}\right)^2}. \tag{4.41}$$

The area of the curve is obtained by Green's theorem as $\oint_C y dx$, which when expanded gives

$$\text{Area} = \sum_{k=0}^{M-1} \sum_{l=-N+1}^{N-1} c_{y,k}\, c_{x,l}^{p}\, q(k-l), \tag{4.42}$$

where $q(m) = \int_{-\infty}^{\infty} \varphi(t)\, \varphi'(t-m)\, dt$ is obtained as in [86]. Note that the area obtained by the above expression is signed; its sign is utilized to determine the sense (clockwise or anti-clockwise) of the curve.

4.7 Evolving the curve

4.7.1 Optimization Algorithm

As mentioned before, the active contour algorithm extracts the final contour by finding the minimum of the energy function. Having obtained the partial derivatives, we can use an efficient optimization algorithm to evolve the contour. Here, we implemented the conjugate gradient and steepest descend algorithms. The conjugate gradient algorithm resulted in slightly faster convergence, but was less flexible for loop recovery and knot addition/deletion discussed later. Hence in our final implementation we reverted to the simpler steepest descend algorithm, which was found to be entirely satisfactory for our purpose.

4.7.2 Loop detection and recovery

The optimization process can sometimes lead to looping curves. The probability of loops is greatly reduced by the introduction of the curvilinear reparameterization energy; without this term, the knots tend to bunch together, eventually resulting in loops (c.f. Fig. 4.4).

Despite the use of the new internal energy, looping may still arise occasionally when the image energy forces some knots to move faster than the others. This compromises our approach since we use Green's theorem which assumes simply

connected regions. In the case of polygonal representation (linear spline curve), Chesnaud et. al. proposed to perform crossing tests to detect the presence of loops [79]. Unfortunately, this method is time consuming and not directly applicable to general scaling function curves. Hence, we devised a fast method for loop detection. We compute the total tangential angle[6]:

$$\theta_{\text{total}} = \int_0^M d\theta\,(t)\,dt, \tag{4.43}$$

where

$$d\theta\,(t) = \frac{x'\,(t)\,y''\,(t) - y'\,(t)\,x''\,(t)}{x'\,(t)^2 + y'\,(t)^2}. \tag{4.44}$$

We show in the Appendix 4-D that the value of the integral (4.43) is $2\,(n-m)\,\pi$, where m and n are the number of loops in the clockwise and anti-clockwise sense, respectively. Hence, for a simply connected curve, we expect $\pm 2\pi$ (depending on the sense in which the curve is described). We approximate (4.43) by a discrete sum over the parameter t.

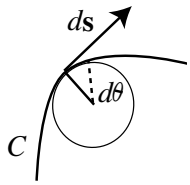


Figure 4.5: Computation of the elemental angle

Note that our criterion can give a value 2π even if the curve is looping (when $n+1=m$), which implies that it is not completely foolproof. In principle, it is possible to detect these cases by splitting the integral (4.43) over a series of smaller intervals and checking if there is a loop in each of the subintervals. However, such cases are unlikely to occur in practice and it was not necessary to implement such a finer level of detection.

Once we detect a loop, we evolve the curve with only the E_{int} term with $\gamma = 0$, thus decreasing its length. In practice, the curvature of the curve at the loops are high. Since minimizing the length corresponds to evolving the curve at every point depending on its curvature [18], the loops tend to disappear very rapidly.

[6] for a plane curve, the tangential angle θ is defined by $d\theta = \kappa\,|d\mathbf{r}|$ [87]

4.7.3 Shrinking/growing snakes

If the snake is initialized away from the actual boundary, it has to shrink/grow to reach the boundary. This changes the average spacing of the knots, which in turn controls the average curvature of the curve (cf. (4.19)).

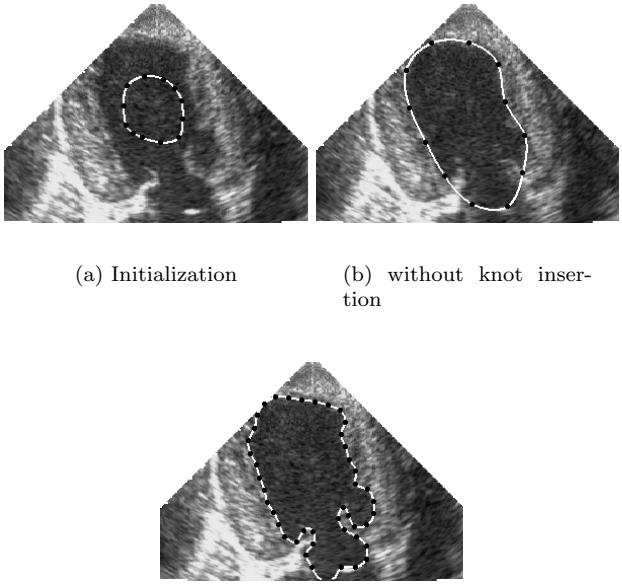
We monitor the length of the curve as it evolves in order to eventually add/delete knots as required. As length can be computed very efficiently by (4.41), this does not introduce a computational overhead. We may then add/delete knot points as required to control or maintain the elasticity of the curve during the evolution process. The addition/deletion of a knot temporarily destroys the uniform spacing of knots. But, thanks to the reparametrization energy term, it returns to the curvilinear abcsissa in a few iterations (without the reparametrization energy, knot insertion is a tricky issue as close knots may eventually lead to looping curves). The performance improvement in adopting this strategy is illustrated in Fig. 4.6.

4.8 Discussion and Summary

We have successfully applied the snake algorithm to a variety of cases including the segmentation of corpus-callossum from MR images and segmentation of the inner heart wall from ultrasound data. Some examples of the segmented corpus-callossum images are shown in Fig. 4.7. Thanks to the unified image energy, the snake gives a good segmentation even if it is not initialized very close to the actual boundary. This approach also makes the algorithm less sensitive to the initial shape of the snake.

The curvilinear reparametrization energy ensures that the curves are smooth. Without this term, the segmentation of the heart data (see from Fig 4.6) is impossible; the curves often resulted in loops. The knot insertion/deletion procedure ensures that the evolving curve has the same stiffness as the initialization.

To conclude, we have presented several enhancements over classical parametric snakes. We have identified some limitations of the conventional gradient magnitude image energy and proposed a new energy that eliminates these problems. We have shown that a general form of this energy can be expressed as a region integral, thus unifying it naturally with the region-based approaches. The unification yields a powerful class of image energies that combines the advantages of edge and region-based approaches. We have shown that the spline representation can guarantee smooth curves if these are described in the curvilinear abscissa. Since the curve evolution process can negatively affect the



(a) Initialization

(b) without knot insertion

(c) with knot insertion

Figure 4.6: Segmentation of the inner wall of the heart of a dog from its ultrasound image. Only the region-based energy is used in this case ($\alpha = 0$). Note that knot (knots are denoted by black dots) insertion and loopcheck is indispensable in this case.

reparametrisation of the curve, we proposed a new internal energy which forces the knot points to remain equally spaced. The various energy terms that we have proposed are summarized in Table 4.1.

The evolution of the curve may lead to looping curves that violate our assumption of the region to be simply connected. Hence, we introduced a simple loop detection test. We also proposed an efficient curve evolution-based algorithm for recovery from the loops. We introduced efficient computational schemes for the evaluation of the partial differentials used in the optimization; we converted all the quantities as curve integrals and simplified the expressions making use of the properties of scaling function curve representation.

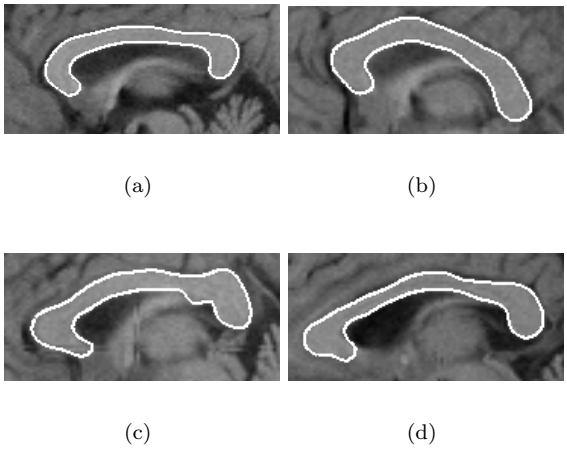


Figure 4.7: Segmentation of corpus-callossum of 4 different subjects from their MR images. The initialization was a small curve at the center similar to Fig. 4.3. We gave equal weight to the region and gradient terms ($\alpha = 0.5$).

The implementation of this algorithm is available as a java plugin for ImageJ [62] at http://bigwww.epfl.ch/jacob/SplineSnake.

Appendix 4-A: Simplification of the curvature term in the internal energy

The square of the curvature of the curve at a point $\mathbf{r}(t)$ can be expressed in the vector form as

$$|\kappa(\mathbf{r})|^2 = \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'')}{|\mathbf{r}'|^6} \tag{4.45}$$

Assuming the parameter t to be in the curvilinear abscissa, we have $|\mathbf{r}'(t)| = c$, $\forall t$. Making use of the vector identity $\mathbf{a}.(\mathbf{b} \times \mathbf{c}) = \mathbf{c}.(\mathbf{a} \times \mathbf{b})$, the numerator

Energy Type	General Expression	Special Cases
Image Energy	$\int_{\mathcal{S}} T_{\mathrm{u}}\left(f\right) d\mathbf{s}$	Gradient-based energy: Eq. (4.4) General edge-based energy: Eq. (4.6) $(\alpha = 1)$
		Region-based energy: Eq. (4.13) $(\alpha = 0)$
		Unified energy: Eq. (4.17) $(0 \leq \alpha \leq 1)$
Internal Energy	$\int_0^M \left\lvert \left\lvert \mathbf{r}'\left(t\right)\right\rvert^2 - \gamma \, \frac{\mathrm{Length}}{M} \right\rvert^2$	Curvilinear reparametrizaton energy: Eq. (4.22) $(\gamma = 1)$
		Length energy: Eq. (4.23) $(\gamma = 0)$
Constraint Energy	$\sum_{i=0}^{N_{\mathrm{c}}-1} \min_{t\in[0,M]} \left\lvert \mathbf{r}\left(t\right) - \mathbf{r}_{\mathrm{c,i}} \right\rvert$	Point constraint: Eq. (4.25)

Table 4.1: Different energy terms used in the snake optimization

of (4.45) can be rewritten as

$$\begin{aligned}
(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'') &= \mathbf{r}'' \cdot (\mathbf{r}' \times \mathbf{r}'' \times \mathbf{r}'') \\
&= \mathbf{r}'' \cdot (\mathbf{r}''(\mathbf{r}' \cdot \mathbf{r}') - \mathbf{r}'(\mathbf{r}' \cdot \mathbf{r}'')) \\
&= |\mathbf{r}''|^2 |\mathbf{r}'|^2 - |\underbrace{\mathbf{r}'' \cdot \mathbf{r}'}_{d(\mathbf{r}'^2)=0}|^2
\end{aligned}$$

In the second step, we make use of the identity $\mathbf{a} \times \mathbf{b} \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{b} \cdot \mathbf{c})\mathbf{a}$. So the expression for the curvature can be written as

$$|\kappa(\mathbf{r})|^2 = \frac{\left|\mathbf{r}''^2\right|}{\left|\mathbf{r}'\right|^4} = \frac{\left|\mathbf{r}''^2\right|}{c^2} \tag{4.46}$$

Appendix 4-B: Partial derivatives of the curvilinear reparametrization term

Expanding (4.22) we obtain

$$\begin{aligned} E_{\text{curv}} &= \int_0^M \left(x'(t)^4 + y'(t)^4 + 2x'(t)^2 y'(t)^2 \right) dt + \\ & \quad c^2 - 4c \int_0^M \left(x'(t)^2 + y'(t)^2 \right) dt \end{aligned} \tag{4.47}$$

Differentiating E_{curv} with respect to $c_{x,k}$, we get

$$\begin{aligned} \frac{\partial E_{\text{curv}}}{\partial c_{x,k}} &= \int_0^M \left(4x'(t)^3 + 4x'(t) y'(t)^2 \right) \frac{\partial}{\partial c_{x,k}} (x'(t))\, dt - \\ & \quad 4c \int_0^M x'(t) \frac{\partial}{\partial c_{x,k}} (x'(t))\, dt \end{aligned} \tag{4.48}$$

Now substituting for $x(t)$ and $y(t)$ from (2.31), yields

$$\begin{aligned} \frac{\partial E_{\text{curv}}}{\partial c_{x,k}} &= \sum_{l,m,n \in \mathbb{Z}} c^p_{x,l}\ c^p_{x,m}\ c^p_{x,n}\ h_1(k-l, k-m, k-n) + \\ & \quad \sum_{l,m,n \in \mathbb{Z}} c^p_{x,l}\ c^p_{y,m}\ c^p_{y,n}\ h_1(k-l, k-m, k-n) + \\ & \quad \sum_{l,m,n \in \mathbb{Z}} c^p_{x,l}\ h_2(k-l) \end{aligned} \tag{4.49}$$

The filters h_1 and h_2 are given by

$$h_1(l,m,n) = \int_{-\infty}^{\infty} \varphi'(t)\, \varphi'(t+l)\, \varphi'(t+m)\, \varphi'(t+n)\, dt \tag{4.50}$$

$$h_2(l) = \int_{-\infty}^{\infty} \varphi'(t)\, \varphi'(t+l)\, dt \tag{4.51}$$

With a change of variables and using the finite support of h_1 and h_2, we can simplify (4.49) to (4.35).

Appendix 4-C: Precomputation of the kernel

We use the property that the derivative of a scaling function φ can be written as $\varphi'(t) = \varphi^{\{1\}}(t) - \varphi^{\{1\}}(t-1)$, where $\varphi^{\{1\}}$ is the scaling function whose mask (scaling filter) is $A^{\{1\}}(z) = \left(\frac{2}{1+z^{-1}}\right) A(z)$; $A(z) \leftrightarrow a_k$ is the mask of φ. Using this relation, we rewrite the filter coefficients (4.36) and (4.37) as

$$h_1(l,m,n) = -\triangle^f \triangle_1^b \triangle_2^b \triangle_3^b g_1(l,m,n) \tag{4.52}$$

$$h_2(l) = -\triangle^f \triangle_1^b g_2(l), \tag{4.53}$$

where

$$\begin{aligned} \triangle_i^b g(l_1,..,l_i,..,l_n) &= g(l_1,..,l_i,..,l_n) - g(l_1,..,l_i-1,..,l_n) \\ \triangle^f g(l_1,l_2..,l_n) &= g(l_1+1,l_2+1,..,l_n+1) - g(l_1,l_2,..,l_n) \end{aligned}$$

and

$$g_1(l,m,n) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\,\varphi^{\{1\}}(t+l)\,\varphi^{\{1\}}(t+m)\,\varphi^{\{1\}}(t+n)\,dt \tag{4.54}$$

$$g_2(l) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\,\varphi^{\{1\}}(t+l)\,dt \tag{4.55}$$

The scaling function $\varphi^{\{1\}}$ satisfies the two-scale relation

$$\varphi^{\{1\}}(t) = \sum_{k=0}^{N} a_k^{\{1\}}\,\varphi^{\{1\}}(2t-k), \tag{4.56}$$

Consequently, the kernels g_1 and g_2 satisfy the two-scale relations

$$g_1(\mathbf{k}) = \sum_{\mathbf{l}} h_1(\mathbf{l})\,g_1(2\mathbf{k}-\mathbf{l}) \tag{4.57}$$

$$g_2(k) = \sum_{\mathbf{l}} h_2(l)\,g_2(2k-l), \tag{4.58}$$

where the two-scale masks H_1 and H_2 are given by

$$h_1(l,m,n) = \frac{1}{2}\left(\sum_k a^{\{1\}}(k)\,a^{\{1\}}(k-l)\,a^{\{1\}}(k-m)\,a^{\{1\}}(k-n)\right) \tag{4.59}$$

$$h_2(l) = \frac{1}{2}\left(\sum_k a^{\{1\}}(k)\,a^{\{1\}}(k-l)\right) \tag{4.60}$$

Using the two-scale relation, the sequences g_1 and g_2 are exactly computed as in [86].

Appendix 4-D: Integral of the tangential angle.

We start by observing that the integral (4.43) can be expressed as

$$\theta_{\text{total}} = \text{Im}\left(\int_0^M \frac{d\left(x'(t) + \mathbf{j}\, y'(t)\right)}{x'(t) + \mathbf{j} y'(t)} dt\right), \tag{4.61}$$

where $\text{Im}(z)$ gives the imaginary part of z and $\mathbf{j} = \sqrt{-1}$. This can be rewritten as the curve integral

$$\theta_{\text{total}} = \text{Im}\left(\oint_{\mathcal{C}'} \frac{dz}{z}\right), \tag{4.62}$$

where $\mathcal{C}'$ is the curve described $(x'(t), y'(t))$ and $z = x' + iy'$. Using Cauchy's integral formula, we obtain the value of this integral as 2π times the winding number[7] of the contour $\mathcal{C}'$ about the origin. Since each loop in $\mathcal{C}$ corresponds to one in $\mathcal{C}'$ in the same sense, but around the origin, the winding number of $\mathcal{C}'$ is $(m - n)$, where m and n are the number of times $\mathcal{C}$ loops in the anticlockwise and clockwise sense respectively.

[7]The winding number of a contour about a point z_0 is the number of times the contour passes around z_0 in the counterclockwise sense [87].

Chapter 5

3-D shape estimation of DNA molecules

In this chapter[1], we discuss the second application of the shape estimation algorithm shown in Fig. 1.4.

5.1 Introduction

Cryo-electron microscopy is an approach used to image bio-molecules such as DNA filaments [88, 89, 90]. The molecules are suspended in a thin layer of liquid, which is then cooled to a very low temperature. Thanks to the rapid cooling (of the order of 10^6 K/s), the resulting specimen can be considered to be a snapshot of its thermal oscillations. As compared to other approaches such as classical electron microscopy and atomic force microscopy, where the molecules are adsorbed onto supporting films, this method does not cause shape deformation. In this paper, we address the 3-D reconstruction of the shape of a DNA molecule from its stereo-micrographs (a typical pair of such images is shown in Fig. 5.1). This data is useful in probing the physical properties of the filament (such as its shape, stiffness, modes of oscillations, shape variation due to protein-bindings etc.), which play important roles in various bio-molecular processes.

[1]Based on the article "M.Jacob, T.Blu, C.Vaillant, J.Maddocks, M.Unser, *submitted to IEEE Transactions on Image Processing*".

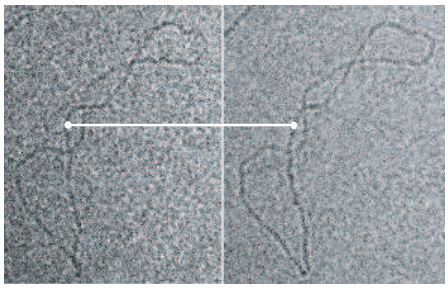


Figure 5.1: Stereo views separated by 30° of a super-coiled DNA filament (1800 base pairs) with a pair of corresponding points marked. Courtesy E. Larquet, Pasteur Institute.

Since exposure to electron beams causes degradation of the specimen, one usually restricts the number of views to two. Due to physical constraints, the angular separation between the views is limited to a maximum of 30 degrees. The micrographs also suffer from poor image contrast and low SNR due to the low electron dose. All these aspects make the reconstruction problem difficult.

The early approaches to this problem included manual reconstruction [91] and a semi-automatic search algorithm called the flying cylinder [92, 93]. In the manual scheme, the user clicks on the images to introduce pairs of corresponding points that define the curve; this is time consuming and not necessarily reproducible. The flying cylinder algorithm detects the filaments by matching the projections (onto the image planes) of a 3-D cylindrical template with the stereo images. Since deriving the 2-D projections of a 3-D cylinder (with arbitrary orientation) was difficult, the authors approximated them with oriented rectangles. To reduce the number of matchings required, they discretized the orientation space and used a sequential search algorithm. The detected fragments were then sorted and interpolated to obtain a continuous curve. The performance of this algorithm is limited by the approximations, angular discretization, and, the multi-step strategy; in particular, the interpolation of the curve is only based on the detected fragments and is not necessarily consistent with the image data, nor the global optimum.

We address these shortcomings and propose a new algorithm that solve the 3-D reconstruction problem in a more exact and consistent manner, using projection-steerable templates and a 3-D active contour model. An outline of the full procedure is given in Fig. 5.2. In the active contour framework, the

shape estimation problem is formulated as an energy minimization. The snake energy is a linear combination of the image energy, the internal energy and the constraint energy terms (we discuss the details of the snake algorithm in Section 5.4). At each iteration of the optimization algorithm, the curve model is evaluated from its coefficients and the energy terms are computed based on the model and image information. The curve coefficients are then updated so that the system converges towards the energy minimum.

The image energy term, which the crucial part of the snake energy, is a measure of the fit of the curve model with the image data. We consider a global model for the DNA filament, whose skeleton is a 3-D parametric Bspline curve, and has a certain radial profile. Ideally, we would project the global model onto the projection planes and match the projections with the images (we compute the sum of the inner-products between the projections and the images) to obtain the fitness measure. Thanks to the linearity of the B-spline representation, the skeletons of the 2-D projections will be 2-D B-spline curves. However, its profile will be different at different curve points, depending of the orientation of the filament at the corresponding 3-D point. Thus the evaluation of the exact projections and performing the matching operation is computationally very expensive. Note that since we use an iterative optimization algorithm, the projections and matching procedure have to performed in a loop.

To reduce the computational complexity, we propose to approximate the global 3-D model locally as an elongated blob-like template. We introduce the concept of projection-steerablity, which is inspired by the work on 2-D orientation steerablity by Freeman et. al. [42, 45, 46]. We derive an elongated template in 3-D that is projection-steerable, i.e., the 2-D projections of this elongated template can be expressed as a linear combination of a few basis functions. With this framework, the matching of the projection of such a 3-D template can be performed inexpensively as a weighted sum of the inner-products between the basis functions and the images. The weights are simple functions of the orientation of the 3-D template and the inner-products are evaluated efficiently by separable filtering. We discuss the projection-steerable ridge detection in detail in Section 5.3.

We show that cubic B-spline representation is optimal for the description of smooth 3-D curves, if described in the constant arc-length parametrization. We also use the constant arc-length assumption to derive a simple expression for the internal energy. For this assumption to hold, we reparametrize the initial curve (derived from user inputs) such that the curve knots are uniformly spaced. Since the length of the DNA molecules are known a-priori, we use an additional constraint term that penalizes the curve for not having the specified length.

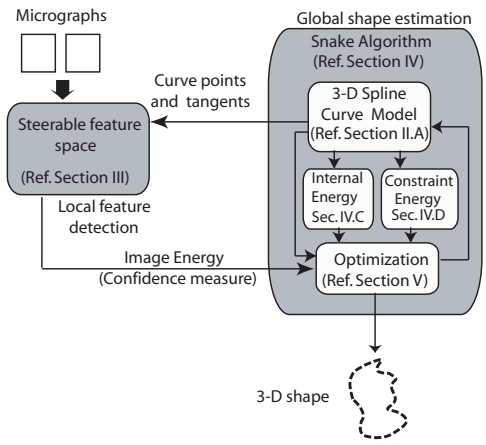


Figure 5.2: Outline of the global 3-D shape estimation algorithm.

We use conjugate gradient algorithm for snake optimization. This scheme requires the efficient evaluation of the partial derivatives of the energy terms. Thanks to the projection-steerable templates and the curve representation using finitely supported B-spline functions, they are computed exactly and efficiently as shown in Section 5.5.

5.2 Mathematical Preliminaries

5.2.1 Parametric representation of 3-D curve

A 3-D curve, (denoted as $\mathcal{C}$) can be described in terms of an arbitrary parameter t as $\mathbf{r}(t) = (x(t), y(t), z(t))$. When the curve is closed, the function vector, $\mathbf{r}(t)$, is periodic.

$\mathbf{r}(t)$ can be represented efficiently as a linear combination of some basis functions. Here, we focus on the B-spline curve representation [9, 17] due to numerous advantages discussed in Section 5.4.3. Specifically, we represent the component functions of a 3-D curve in a uniform B-spline basis as

$$\mathbf{r}(t) = \begin{bmatrix} x\,(t) \\ y\,(t) \\ z\,(t) \end{bmatrix} = \sum_{k=-\infty}^{\infty} \mathbf{c}\,(k)\,\beta^n(t-k), \tag{5.1}$$

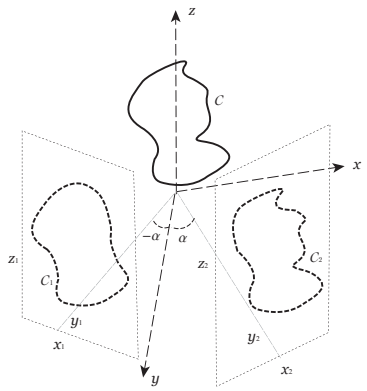


Figure 5.3: 3-D curve and its 2-D projections.

where $\mathbf{c}(k) = [c_x(k), c_y(k), c_z(k)]$ is a sequence of coefficient vectors; in computer graphics, these are often called the control points [15]. The basis function β^n is the B-spline of degree n [27]. If the period, M, is an integer, we have $\mathbf{c}(k) = \mathbf{c}(k+M)$. This reduces the infinite summation to

$$\mathbf{r}(t) = \sum_{k=0}^{M-1} \mathbf{c}(k)\, \beta_p^n(t-k), \tag{5.2}$$

where β_p^n is the M - periodization of β^n:

$$\beta_p^n(t) = \sum_{k=-\infty}^{\infty} \beta^n(t - k\,M) \tag{5.3}$$

Note that the special case of $n = 1$ (linear splines) yields a curve that is composed of line segments connecting the control points.

We denote the orthonormal basis vectors of the volume as $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. The basis vectors of the ith projection plane is $(\mathbf{e}_{x_i}, \mathbf{e}_{z_i})$, while the vector orthonormal to the plane is denoted by $\mathbf{e}_{y_i}$. An arbitrary vector $\mathbf{r}$, can be denoted as

$$\mathbf{r} = x\,\mathbf{e}_x + y\,\mathbf{e}_y + x\,\mathbf{e}_z \tag{5.4}$$

$$= x_i\,\mathbf{e}_{x_i} + y_i\,\mathbf{e}_{y_i} + z_i\,\mathbf{e}_{z_i} \tag{5.5}$$

The projection of the vector $\mathbf{r}$ to the plane is given by

$$\mathbf{r}_i = \mathbf{P}_i\,\mathbf{r} = x_i\,\mathbf{e}_{x_i} + z_i\,\mathbf{e}_{z_i}, \tag{5.6}$$

where $\mathbf{P}_i$ are the orthogonal projection matrices. The reconstruction algorithm requires projecting the curve model onto the image planes. Thanks to the linearity of the representation, the 2-D curve projections are also B-spline curves. Thus, the 2-D curve coefficients are:

$$\mathbf{c}_i\,(k) \quad = \quad \mathbf{P}_i\;\mathbf{c}\,(k)\,, \tag{5.7}$$

Thus, the 2-D curve projections (we denote them by $\mathcal{C}_i$) are given by

$$\mathbf{r}_i\,(t) \quad = \begin{bmatrix} x_i\,(t) \\ z_i\,(t) \end{bmatrix} = \quad \sum_{k=0}^{M-1} \mathbf{c}_i\,(k)\;\beta_p^n(t-k), \tag{5.8}$$

where $\mathbf{r}_i\,(t) = \mathbf{P}_i\mathbf{r}\,(t)$. The projection matrix $\mathbf{P}_i$ can be thought off as the composition of a rotation matrix and a simple projection operator $\mathbf{P}$:

$$\mathbf{P}_i = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{P}} \mathbf{R}_i \tag{5.9}$$

Note that $\mathbf{P}$ is the same for all $\mathbf{P}_i$, while the projection geometry is specified by the unitary matrix $\mathbf{R}_i$; The rotation matrix performs the coordinate transformation from (x, y, z) to (x_i, y_i, z_i).

For the projection geometry shown in Fig. 5.3 ($\alpha_0 = -\alpha$, $\alpha_1 = \alpha$), the rotation matrices are

$$\mathbf{R}_1 \quad = \quad \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

$$\mathbf{R}_2 \quad = \quad \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

respectively.

5.2.2 Orthogonal volume projection

We now model the measurement process by a line integral and obtain the expressions for the projection images, given the 3-D volume data. We denote the volume by $f\,(\mathbf{r})$, where $\mathbf{r} = (x, y, z)$. The projected images $f_i = \mathcal{P}_i f$ are represented in the 2-D coordinate system where $\mathcal{P}_i$ denote the orthogonal volume

projection operator. Thus,

$$f_i\left(\mathbf{r}_i\right) = \int_{-\infty}^{\infty} f\left(\mathbf{P}_i^t\,\mathbf{r}_i + \gamma\,\mathbf{e}_{y_i}\right) d\gamma. \tag{5.12}$$

The above equation is easier to understand in the Fourier domain. The Fourier transform of f is given by

$$\hat{f}\left(\boldsymbol{\omega}\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(\mathbf{r}\right) e^{j\langle\boldsymbol{\omega},\mathbf{r}\rangle} dxdydz, \tag{5.13}$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$. The Fourier transform of the function in the coordinate system (x_i, y_i, z_i) can be shown to be $\hat{f}\left(\mathbf{R}_i^t\,\boldsymbol{\omega}\right)$. The Fourier transform of the projection can be obtained by setting $\omega_{y_i} = 0$ (or by substituting $\boldsymbol{\omega} = \mathbf{P}^t\boldsymbol{\omega}_i;\ \ \boldsymbol{\omega}_i = (\omega_{x_i}, \omega_{z_i})$) in $\hat{f}\left(\mathbf{R}_i^t\,\boldsymbol{\omega}\right)$. Thus, the two-variable Fourier transform of the image is given by

$$\begin{aligned} \hat{f}_i\left(\boldsymbol{\omega}_i\right) &= \hat{f}\left(\mathbf{R}_i^t\mathbf{P}^t\boldsymbol{\omega}_i\right) \\ &= \hat{f}\left(\mathbf{P}_i^t\boldsymbol{\omega}_i\right) \end{aligned} \tag{5.14}$$

This expression can also be obtained using the Fourier-slice theorem [94].

5.3 Local filament detection

We have seen that the direct use of a global DNA model in the optimization algorithm can lead to a high computational complexity. Hence, we approximate the global model locally as an elongated blob. In this section, we address the detection of elongated blob-like structures in 3-D from their 2-D orthogonal projections (see Fig. 5.1). We addressed a similar problem in [95], where we derived the optimal rotation-steerable filters for 2-D feature detection. This method gave promising results for the detection of 2-D line-like structures. In this section, we generalize the concept of rotation steerablity to projection-steerablity for 3-D filaments detection. This approach is well-suited for both the local scheme (where the detection is performed independently at each point) as well as the global approach (where the optimal orientation is specified by a model whose parameters are estimated).

5.3.1 Projection based feature detection

Suppose our task is to check for the presence of an elongated 3-D blob—denoted by $f_{\mathrm{c}}\left(\mathbf{r}\right); \mathbf{r} \in \mathbb{R}^3$—with an unknown orientation, at a particular position $\mathbf{r}_{\mathrm{c}}$, in a

3-D volume f. The volume is known only through its orthogonal 2-D projections $f_i = \mathcal{P}_i f$. We formulate the detection procedure as a matched filtering; we consider a 3-D detector and match its orthogonal projections onto the image planes with the micrographs.

We choose the 3-D template to be $h(\mathbf{r}) = f_c(-\mathbf{r})$ and denote its rotated versions by $h_{\mathbf{v}}(\mathbf{r}) = h(\mathbf{R}_{\mathbf{v}}\mathbf{r})$, where $\mathbf{R}_{\mathbf{v}}$ is a 3-D rotation matrix. We use the sum of the inner-products between the 2-D template projections and the micrographs as the performance criterion:

$$C_{\mathbf{v}}(\mathbf{r}_{\mathbf{c}}) = \sum_{i=0}^{N-1} \left(f_i * \mathcal{P}_i(h_{\mathbf{v}})\right)(\mathbf{r}_{\mathbf{c},i}) \tag{5.15}$$

where $\mathcal{P}_i;\ \ i = 0 \ldots N-1$ are the orthogonal projection operators[2] and $\mathbf{r}_{\mathbf{c},i} = \mathbf{P}_i\,\mathbf{r}_{\mathbf{c}}$. Note that this criterion is a function of the orientation vector $\mathbf{v}$. If we perform the filament detection independently at each point, the optimal orientation vector and the likeliness measure are given by

$$\mathbf{v}^*(\mathbf{r}) = \arg\max_{|\mathbf{v}|=1}\left(C_{\mathbf{v}}(\mathbf{r})\right) \tag{5.16}$$

$$r^*(\mathbf{r}) = C_{\mathbf{v}^*}(\mathbf{r}), \tag{5.17}$$

For an arbitrary 3-D template, the computation of the projections $\mathcal{P}_i(h_{\mathbf{v}})$ are expensive. To obtain the optimal orientation by numerical optimization, the template projections and their inner-products with the micrographs have to computed for each iteration; a direct implementation of the algorithm is not very practical, unless simplifying assumptions are made.

5.3.2 Projection-steerable ridge detection

To reduce the complexity in performing the projection matched filter detection, we use an approach similar to rotation steerablity [42, 45, 46]. We would like to have a good 3-D filament detector whose projections (for any spatial orientation) are contained in a space spanned by a few basis functions. For such a detector, the evaluation of the performance criterion for each curve point simplifies to a weighted sum of the inner-products of the basis functions with the micrographs. The inner-products themselves can be efficiently pre-computed using 2-D filtering.

[2] In our case $N = 2$, but the scheme is applicable for the general case as well.

We now consider the family

$$V_{3\mathrm{D}} = \mathrm{span}\left\{\partial_{xx}\, g_3\left(\mathbf{r};\sigma\right), \partial_{yy}\, g_3\left(\mathbf{r};\sigma\right), \partial_{zz}\, g_3\left(\mathbf{r};\sigma\right), \partial_{xy}\, g_3\left(\mathbf{r};\sigma\right), \partial_{xz}\, g_3\left(\mathbf{r};\sigma\right), \partial_{yz}\, g_3\left(\mathbf{r};\sigma\right)\right\}, \tag{5.18}$$

where $g_D\left(\mathbf{r};\sigma\right) = \frac{1}{(2\pi\sigma)^{\frac{D}{2}}} \exp\left(-\frac{|\mathbf{r}|^2}{2\sigma^2}\right)$ is a D-dimensonal Gaussian and $\partial_{xy}\ f\left(\mathbf{r}\right) = \frac{\partial^2}{\partial x \partial y}\left(f\left(\mathbf{r}\right)\right)$. We now show that any 3-D filter in this family is ideally suited for projection matched filter detection.

Proposition 3 *The space $V_{3\mathrm{D}}$ is closed with respect to 3-D rotations.*

Proof: The Fourier transforms of the basis functions are

$$\begin{aligned}
\partial_{xx}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_x^2\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{yy}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_y^2\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{zz}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_z^2\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{xy}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_x\omega_y\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{xz}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_x\omega_z\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{yz}\, g_3\left(\mathbf{r};\sigma\right) &\overset{\mathcal{F}}{\leftrightarrow} -\left(2\pi\right)^{\frac{3}{2}}\ \omega_y\omega_z\ g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)
\end{aligned}$$

where $\boldsymbol{\omega} = \left[\omega_x, \omega_y, \omega_z\right]^t$. Since the basis functions are the products of second degree monomials with a Gaussian in the Fourier domain, an arbitrary function in $V_{3\mathrm{D}}$ is a second degree polynomial multiplied by a Gaussian. It can be written in a compact form as

$$\hat{h}\left(\boldsymbol{\omega}\right) = \left(2\pi\right)^{\frac{3}{2}}\ \left(\boldsymbol{\omega}^t \mathbf{A}\, \boldsymbol{\omega}\right) g_3\left(\boldsymbol{\omega};\sigma^{-1}\right) \tag{5.19}$$

Here, $\mathbf{A}$ is a symmetric 3x3 coefficient matrix that characterizes the shape of h. The Fourier transform of a $\mathbf{R}$-rotated version[3] of h is given by

$$\hat{h}\left(\mathbf{R}\boldsymbol{\omega}\right) = \left(2\pi\right)^{\frac{3}{2}} \left(\boldsymbol{\omega}^t \underbrace{\mathbf{R}^t \mathbf{A} \mathbf{R}}_{\mathbf{A_R}} \boldsymbol{\omega}\right) g_3\left(\boldsymbol{\omega};\sigma^{-1}\right), \tag{5.20}$$

[3]we use the property that the rotation of the filter is equivalent to rotating its Fourier transform

where $\mathbf{R}$ is the 3x3 rotation matrix. Note that in the above step, we have used the isotropy of the function $\hat{g}$. Since the new filter $h(\mathbf{Rr})$ has the same form as (5.19) for any rotation matrix $\mathbf{R}$, it is still in $V_{3\mathrm{D}}$. ■

Proposition 4 *The orthogonal projection $\mathcal{P}_i$ of the space $V_{3\mathrm{D}}$ onto a plane is the function space $V_{2\mathrm{D},i}$:*

$$V_{2\mathrm{D},i} = \mathrm{span}\left\{\partial_{x_i x_i}\, g_2(\mathbf{r}_i;\sigma), \partial_{z_i z_i}\, g_2(\mathbf{r}_i;\sigma), \partial_{x_i z_i}\, g_2(\mathbf{r}_i;\sigma)\right\}. \tag{5.21}$$

Proof: The Fourier transform of the projection of an arbitrary function in $V_{3\mathrm{D}}$ is obtained by substituting (5.19) in (5.14):

$$\hat{h}_i(\boldsymbol{\omega}_i) = \sqrt{2\pi}\left(\boldsymbol{\omega}_i^t \underbrace{\mathbf{P}_i \mathbf{A} \mathbf{P}_i^t}_{\mathbf{B}_i} \boldsymbol{\omega}_i\right) \underbrace{2\pi\, g_3\left(\mathbf{P}_i^t \boldsymbol{\omega}_i;\sigma^{-1}\right)}_{g_2(\boldsymbol{\omega}_i;\sigma^{-1})}, \tag{5.22}$$

where g_2 is a 2-D Gaussian. Since the 2x2 matrix $\mathbf{B}_i$ is symmetric, $\hat{h}_i(\boldsymbol{\omega}_i)$ is second degree polynomial in $\boldsymbol{\omega}_i$. This implies that h_i is a linear combination of the functions $\partial_{xx}\, g_2(\mathbf{r}_i,\sigma)$, $\partial_{xy}\, g_2(\mathbf{r}_i,\sigma)$ and $\partial_{yy}\, g_2(\mathbf{r}_i;\sigma)$. Thus $\mathcal{P}_i(V_{3\mathrm{D}}) \subseteq V_{2\mathrm{D,i}}$.

We also have the relations $\mathcal{P}_i(\partial_{xx}\, g_3) = \partial_{x_i x_i}\, g_2$, $\mathcal{P}_i(\partial_{zz}\, g_3) = \partial_{z_i z_i}\, g_2$ and $\mathcal{P}_i(\partial_{xz}\, g_3) = \partial_{x_i z_i}\, g_2$. They imply that $V_{2\mathrm{D,i}} \subseteq \mathcal{P}_i(V_{3\mathrm{D}})$. Thus we have $V_{2\mathrm{D,i}} = \mathcal{P}_i(V_{3\mathrm{D}})$

■

We have seen that $V_{3\mathrm{D}}$ is closed under 3-D rotations. Hence, if we choose a 3-D detector in this space, its rotated versions are guaranteed to be in the same space. We have also seen that $\mathcal{P}_i(V_{3\mathrm{D}}) = V_{2\mathrm{D,i}}$, which implies that the projection of any rotated version of the detector is in $V_{2\mathrm{D,i}}$. Moreover, since the functions in $V_{2\mathrm{D,i}}$ are band-pass, the detection scheme will not be sensitive to smooth intensity variations that are common with micrographs[4].

In this paper, we have restricted ourselves to second order detectors for simplicity. However the concept of projection-steerablity is more general; any 3-D function that can be represented as a linear combination of the differentials (up to a certain order) of an isotropic function is projection-steerable.

5.3.3 3-D ridge detection

We have seen that $V_{3\mathrm{D}}$ is ideally suited for projection-steerable matching. Hence, we would like to choose the most elongated blob-like structure in this space as

[4] In traditional schemes, these variations are removed by a high-pass preprocessing filter[92].

our local 3-D template. We derive the optimally elongated local template in Appendix 5-A as

$$h(\mathbf{r}) = \sqrt{\frac{3}{20}} \left(\partial_{yy}\, g_3(\mathbf{r},\sigma) + \partial_{zz}\, g_3(\mathbf{r},\sigma) - \frac{2}{3}\, \partial_{xx}\, g_3(\mathbf{r},\sigma) \right) \tag{5.23}$$

See 3-D plots of this detector in Fig. 5.4 and Fig. 5.5.

Neglecting the normalization constant, we rewrite the expression for the optimal filter (5.23) oriented along the unit vector $\mathbf{v}$ as

$$h_{\mathbf{v}}(\mathbf{r}) = \underbrace{\left(\partial_{xx} + \partial_{yy} + \partial_{zz}\right)\, g_3(\mathbf{r};\sigma)}_{\text{Laplacian of } g(\mathbf{r};\sigma)} - \frac{5}{3}\, \partial_{\mathbf{vv}}\, g_3(\mathbf{r};\sigma)\,, \tag{5.24}$$

where $\partial_{\mathbf{vv}} f(\mathbf{r}) = \frac{\partial^2}{\partial\gamma^2} f(\mathbf{r} + \gamma\,\mathbf{v})$. Note that the Fourier transform of the filter is given by (5.19) with $\mathbf{A} = \mathbf{I}_3 - \frac{5}{3}\,\mathbf{v}\mathbf{v}^t$, where $\mathbf{I}_3$ is the 3x3 identity matrix. By substituting (5.24) in (5.22) and by performing the manipulations shown in Appendix 5-B, we get

$$\mathcal{P}_i\left(h_{\mathbf{v}}(\mathbf{r})\right) = \mathbf{v}^t \left[\mathbf{R}_i^t\, \mathbf{G}_i(\mathbf{r}_i;\sigma)\, \mathbf{R}_i\right] \mathbf{v}, \tag{5.25}$$

where

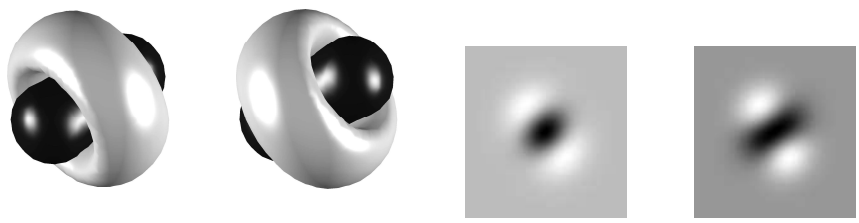
$$\mathbf{G}_i(\mathbf{r}_i;\sigma) = \begin{bmatrix} \left(\partial_{z_i z_i} - \frac{2}{3}\,\partial_{x_i x_i}\right) & 0 & -\left(\frac{5}{3}\,\partial_{x_i z_i}\right) \\ 0 & \left(\partial_{x_i x_i} + \partial_{z_i z_i}\right) & 0 \\ -\left(\frac{5}{3}\,\partial_{x_i z_i}\right) & 0 & \left(\partial_{z_i z_i} - \frac{2}{3}\,\partial_{x_i x_i}\right) \end{bmatrix} g_2(\mathbf{r}_i;\sigma) \tag{5.26}$$

and $\mathbf{R}_i$ is the rotation matrix given by (5.9) and (5.11).

Note that, when $\mathbf{R}_i\mathbf{v} = (1,0,0)$ (horizontal filament parallel to the image plane), we get $\mathcal{P}_i(h_{\mathbf{v}}) = g_{z_i z_i}(\mathbf{r}_i;\sigma) - \frac{2}{3} g_{x_i x_i}(\mathbf{r}_i;\sigma)$—an elongated detector. On the other hand, if $\mathbf{R}_i\mathbf{v} = (0,1,0)$ (i.e., the filament is orthogonal to the image plane so that its projection is an isotropic blob rather than a filament), we get $\mathcal{P}_i(h_{\mathbf{v}}) = g_{x_i x_i}(\mathbf{r}_i;\sigma) + g_{z_i z_i}(\mathbf{r}_i;\sigma)$—the isotropic Laplacian detector. In other words, we use different 2-D detectors on the image planes, depending on the spatial orientation of the 3-D template. We give some examples with the 3-D template and their projections in Fig. 5.4 and Fig. 5.5.

Using (5.25), we simplify (5.15) to

$$C_{\mathbf{v}}(\mathbf{r}) = \mathbf{v}^t \underbrace{\left[\sum_{i=0}^{N-1} \mathbf{R}_i^t\, \mathbf{H}_{f_i}(\mathbf{r}_i)\, \mathbf{R}_i\right]}_{\mathbf{H}_{3\mathrm{D}}(\mathbf{r})} \mathbf{v} \tag{5.27}$$



(a) front view (b) top view (c) left (d) right

Figure 5.4: (a) and (b) Isosurface plots of the front (viewed from the x-z plane) and top (viewed from the x-y plane) views of a 3-D detector oriented at 30° to the x- axis and 30° to the x-y plane ($\theta = 30°, \phi = 30°$). (c) and (d) Projections of the 3-D filter onto the image planes oriented at $-15°$ and $-15°$ to the y axis

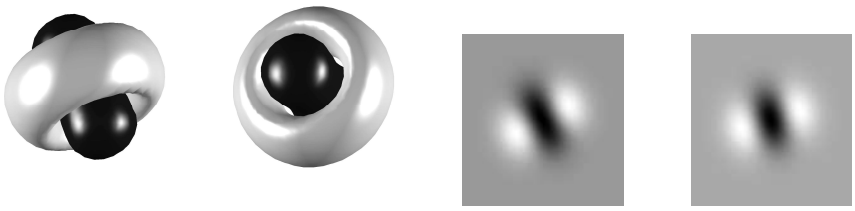
where

$$\mathbf{H}_{f_i}(\mathbf{r}_i) = f_i * \mathbf{G}_i(\mathbf{r}_i; \sigma) \tag{5.28}$$

Thus, the evaluation of likeliness of a filament at a specified 3-D oriented along $\mathbf{v}$ is given by (5.27); the evaluation of $\mathbf{H}_{f_i}(\mathbf{r}_i)$ requires the evaluation of the inner-products with the micrographs and the 5 non-zero entries of (5.26). Since these entries are linear combinations of the three functions $\partial_{x_i x_i} g_2(\mathbf{r}_i; \sigma), \partial_{z_i z_i} g_2(\mathbf{r}_i; \sigma)$ and $\partial_{x_i z_i} g_2(\mathbf{r}_i; \sigma)$, we need to only evaluate the inner-products with them; the terms of (5.28) can be derived as linear combinations of these inner-products. The inner-products themselves can be obtained efficiently as by performing separable filtering of the micrographs with $\partial_{x_i x_i} g_2(\mathbf{r}_i; \sigma), \partial_{z_i z_i} g_2(\mathbf{r}_i; \sigma)$ and $\partial_{x_i z_i} g_2(\mathbf{r}_i; \sigma)$.

5.4 Constrained reconstruction using the 3-D snake model

In the previous section, we have addressed the local detection of elongated 3-D blobs. In this section, we combine the local likeliness measures into the global model and estimate its parameters. Since the skeleton of the DNA molecule is



(a) front view (b) top view (c) left (d) right

Figure 5.5: (a) and (b) Isosurface plots of the front (viewed from the x-z plane) and top (viewed from the x-y plane) views of a 3-D detector oriented at $-45°$ to the x- axis and $-60°$ to the x-y plane ($\theta = -45°, \phi = -60°$). (c) and (d) Projections of the 3-D filter onto the image planes oriented at $-15°$ and $-15°$ to the y axis

represented by a curve, the well established framework of active contour models [8, 63] is very appropriate for this purpose.

5.4.1 Active contour algorithm: Formulation

Traditionally, snakes or active contour models were introduced for the segmentation of closed objects in images. The popularity of these schemes may be attributed to their ability to aid the segmentation process with a-priori knowledge and user interaction. Snakes, as introduced in the seminal work of Kass et. al., are smooth curve models that evolve from an initial guess towards some boundary in the image such that some energy functional is minimized [8, 63].

These models were extended to 3-D for the estimation of coronary vessel centerlines from X-ray angiographic projections [96, 97]. This approach considers the evolution of a 3-D curve so that its 2-D projections onto the respective planes match the images. The matching is performed using distance maps or gradient vector flow fields. We have also used a similar approach previously for the estimation of DNA shape[51].

Here, we propose a refined approach, which mainly differs in the matching

procedure and the curve representation used. Our criterion is obtained by projecting the optimal templates, oriented along the curve tangents, onto the image planes and matching them with the micrographs. Note that the shapes of the template projections depend on the 3-D curve tangent directions. This suggests that this scheme should give better results than the previous approaches where the 2-D curve projections are matched.

Before going into the details of the algorithm, we briefly review the fundamentals of the snake algorithm. Since the final shape is determined by the minimum of the snake energy, its choice deserves proper attention. Similar to conventional snakes, we choose the energy functional as a linear combination of three separate terms.

1. The image energy, which is responsible for guiding the snake towards the filament.

2. The internal energy, which ensures that the extracted shape of the filament is smooth.

3. The constraint energy, which enables the user to enforce extra constraints such as the curve length.

The total energy of the snake is written as

$$E_{\text{snake}}\left(\mathbf{\Theta}\right) = E_{\text{image}}\left(\mathbf{\Theta}\right) + E_{\text{int}}\left(\mathbf{\Theta}\right) + E_{\text{const}}\left(\mathbf{\Theta}\right), \tag{5.29}$$

where $\mathbf{\Theta}$ is the collection of curve coefficients $\mathbf{\Theta} = \{\mathbf{c}\left(k\right);\ k = 0, \ldots M-1\}$. The optimal curve parameters are obtained as

$$\overline{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} E_{\text{snake}}\left(\mathbf{\Theta}\right) \tag{5.30}$$

We describe the different energy terms in detail in the following subsections.

5.4.2 Image Energy

The image energy term is a measure of the fit of the model to the image data. Consider a point $\mathbf{r}\left(t\right)$ on the planar curve $\mathcal{C}$; the tangent vector of the curve at $\mathbf{r}\left(t\right)$, given by $d\mathbf{r}\left(t\right) = \left(dx\left(t\right), dy\left(t\right), dz\left(t\right)\right)$, defines the direction of the elongated blob at that point. We define the likeliness of a blob at the curve point $\mathbf{r}\left(t\right)$, oriented along $d\mathbf{r}\left(t\right)$, as

$$E_{\text{goodness}}\left(\mathbf{r}\right) = d\mathbf{r}\left(t\right)^{t}\ \mathbf{H}_{\text{3D}}\left(\mathbf{r}\right)\ d\mathbf{r}\left(t\right). \tag{5.31}$$

Energy Type	Expression
Image Energy	$E_{\text{image}} = -\oint_{\mathcal{C}} d\mathbf{r}^t \, \mathbf{H}_{\text{3D}}(\mathbf{r}) \, d\mathbf{r}$ (ref Eq. (5.32))
Internal Energy	$E_{\text{int}} = \frac{1}{k^2} \int_0^M \lvert \mathbf{r}''(t) \rvert^2 \, dt$ (ref Eq. (5.34))
Length Constraint	$E_{\text{const}} \left(\int_0^M \lvert \mathbf{r}'(t) \rvert \, dt - \text{Length} \right)^2$ (ref Eq. (5.38))
Point Constraint	$E_{\text{const}} = \sum_{i=0}^{N_c - 1} \min_{t \in [0,M]} \lvert \mathbf{r}(t) - \mathbf{r}_{c,i} \rvert^2$ (ref Eq. (5.39))

Table 5.1: Different energy terms used in the snake optimization

Recall from subsection 5.3.2 that this quantity is equivalent to projecting the optimal 3-D detector, oriented along $d\mathbf{r}$, onto the projection planes and then computing the sum of the square errors between the template projections and the micrographs. Note that if $d\mathbf{r} = \mathbf{v}^*$ as in (5.16), we get $E_{\text{goodness}}(\mathbf{r}) = r^*(\mathbf{r})$ which is the maximum possible value.

We obtain the likeliness of the entire curve by integrating the goodness measures along the curve.

$$E_{\text{image}}(\Theta) = -\int_0^M \left(d\mathbf{r}(t)^t \, \mathbf{H}_{\text{3D}}(\mathbf{r}) \, d\mathbf{r}(t) \right) dt, \tag{5.32}$$

The negative sign is introduced since the curve evolution is posed as an energy minimization problem. By using (5.30) to obtain the optimal coefficients, we are jointly estimating the optimal orientations and magnitudes at the voxels through which the curve passes. Note that the optimal orientation at each curve point is dependent on the optimal coefficients indirectly through the curve model (i.e. the tangent to the curve). Since the number of curve coefficients is typically much less than the number of voxels through which the curve passes, this scheme is more robust than a local approach. Note that (5.32) is independent of the curve parameter t. Evolving the curve using such a measure will not cause the parametrization to change during the optimization process, thus preserving the curve stiffness[5].

[5]Many snake energies are parameter dependent, causing the curve knots to accumulate at points of high edge strength.

5.4.3 Internal Energy

The internal energy term is responsible for ensuring the smoothness of the reconstructed shape. It is essentially a regularization term that penalizes non-smooth shapes, thus making the reconstruction problem better conditioned. The smoothness of the curve can be quantified by its total curvature magnitude; a stiff curve will have a low value of mean curvature magnitude. The curvature of the curve at a point $\mathbf{r}(t)$ is defined as

$$|\kappa(\mathbf{r})|^2 = \left| \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{|\mathbf{r}'(t)|^3} \right|^2, \tag{5.33}$$

where

$$\mathbf{r}'(t) = (x'(t), y'(t), z'(t))$$

is the derivative vector and

$$\mathbf{r}''(t) = (x''(t), y''(t), z''(t))$$

is the vector of second differentials. Using the expression of the average curvature magnitude—$\int_0^M |\kappa(\mathbf{r})|^2 dt$—directly as the internal energy leads to complicated expressions for the partial derivatives. Using standard results from differential geometry [98], we show in Appendix 5-C that this term can be simplified to

$$\int_0^M |\kappa(\mathbf{r})|^2 dt = \frac{1}{k^2} \int_0^M |\mathbf{r}''(t)|^2 dt \tag{5.34}$$

provided

$$|\mathbf{r}'(t)|^2 = k, \quad \forall t; \tag{5.35}$$

that is, when the curve is parametrized by its curvilinear abscissa. Here

$$k = \frac{1}{M} \left(\underbrace{\int_0^M |\mathbf{r}'(t)| dt}_{\text{Length}} \right)^2 \tag{5.36}$$

is the total length per unit value of the parameter.

Since the uniform B-spline curve has its knots at the integer parameter values, (5.35) requires that the knots be uniformly spaced on the curve. Thus the smoothness term (5.34) is inversely proportional to the fourth power of

the distance between the knots; the curve will be smooth if its knots are well separated. We will see in Section 5.5.2 that the partial derivatives of the r.h.s of (5.34) are much easier to compute than those of its l.h.s. To ensure that (5.35) hold, we resample the initial curve (obtained by user initialization) such that we have constant arc-length.

Choice of the basis function

Using the well-known variational properties of B-splines [28], we can show that the minimization of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ subject to interpolation constraints give a cubic spline curve. Thus, the cubic B-spline representation appears to be the natural choice for parametric curves, for it gives minimum curvature curves when the knots are uniformly spaced. The use of spline curves also brings in additional gains due to the existence of efficient algorithms [27], the local control of the contour due to the finite support of the B-spline basis function, and their good approximation properties [26].

Internal energy term

We reparametrize the initial curve (derived from the interpolation of the user input points) so that the knot points are uniformly spaced. Thanks to the parameter independent image energy term, we can safely assume that the curve will remain approximately in the constant arc-length parametrization. Hence, we choose the internal energy term as

$$E_{\text{int}} = \frac{1}{k^2} \int_0^M |\mathbf{r}''(t)|^2 \, dt. \tag{5.37}$$

Recall from (5.34) that using this term as the internal energy is equivalent to minimizing the average square magnitude of the curvature.

5.4.4 External constraint energy.

As mentioned before, the external constraint energy is a means for the user to enforce extra constraints on the reconstruction. We use two constraint terms in our implementation.

Length constraint

The length of the DNA filaments are known a-priori; this information can be imposed on the reconstruction process to make it more robust. We introduce

this constraint into the framework by penalizing the term

$$E_{\text{const}} = \left(\int_0^M |\mathbf{r}'(t)|\, dt - \text{Length} \right)^2, \tag{5.38}$$

where Length is the expected length of the molecule.

Point constraint

We use a point constraint to enable the user to aid the reconstruction process; he can specify a few 3-D points that should lie on the final shape. This constraint is basically the sum of the distances between these points and the closest points on the curve. The constraint energy is given by

$$E_{\text{const}} = \sum_{i=0}^{N_{\text{c}}-1} \min_{t \in [0,M]} |\mathbf{r}(t) - \mathbf{r}_{\text{c},i}|^2, \tag{5.39}$$

where $\mathbf{r}_{\text{c},i}; i = 0, \ldots, N_{\text{c}} - 1$ are the constraints. This approach can be thought off as introducing virtual springs that pull the curve towards the desired points. One end of the spring is fixed to the constraint point, while the other end slides on the curve.

5.5 Curve evolution: the optimization algorithm

As mentioned before, the snake algorithm evolves the curve from its initial position to the final shape using energy minimization. Since the individual energy terms are non-linear functions of the curve coefficients, we require a numerical optimization algorithm. We use the conjugate gradient algorithm to refine the initial guess derived from the user inputs. The user specifies pairs of corresponding points on the stereo images that are then interpolated to derive the initial 3-D curve. This curve is later resampled to a specified number of knot points (since the approximation ability is decided by the number of knot points) such that (5.35) is satisfied. A summary of the whole algorithm is given in Fig 5.6.

The optimization scheme requires the evaluation of the partial derivatives of the snake energy. Since these quantities have to be repeatedly evaluated in the iteration loop, their computational complexity will determine the time taken by the snake algorithm. In this section, we derive efficient expressions for the derivatives of the individual energy terms.

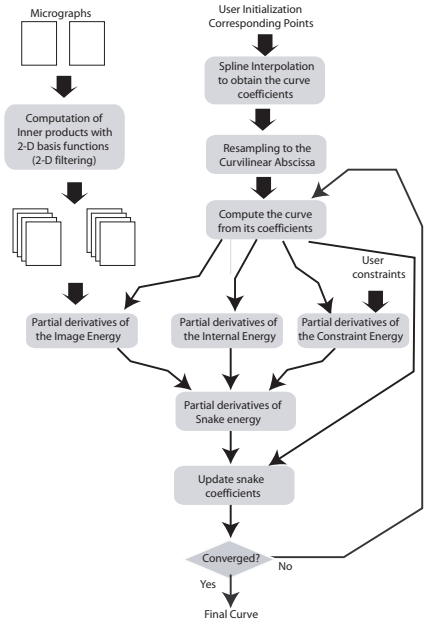


Figure 5.6: Block diagram of the snake optimization algorithm.

5.5.1 Partial derivatives of the image energy

Differentiating (5.32) with respect to the coefficient $c_x(k)$ and applying the chain rule, we get

$$\frac{\partial}{\partial c_x(k)} E_{\text{image}} = \underbrace{-2\int_0^M \left(d\mathbf{r}(t)^t \, \mathbf{H}_{3\text{D}}(\mathbf{r}(t)) \begin{bmatrix} \beta_p^{n\prime}(t-k) \\ 0 \\ 0 \end{bmatrix} \right) dt}_{I_1} \underbrace{-2\int_0^M \left(d\mathbf{r}^t(t) \, \mathbf{H}_{3\text{D,x}}(\mathbf{r}(t)) \, d\mathbf{r}(t) \, \beta_p^n(t-k) \right) dt}_{I_2}, \tag{5.40}$$

where $\mathbf{H}_{3\text{D,x}}$ is a 3x3 matrix whose entries are the partial derivatives of the corresponding entries of $\mathbf{H}_{3\text{D}}$ with respect to x. We now focus on obtaining the expression of $\mathbf{H}_{3\text{D,x}}$

	Along x	Along y	Along z	Total Error
Manual Tracing	1.923 ± 0.9747	1.9097 ± 0.3785	1.71 ± 0.7414	3.279 ± 1.076
Snake Output	1.4024 ± 0.6654	1.3998 ± 0.3429	1.2322 ± 0.5896	2.39 ± 0.795
Performance Improvement	37.08%	36.42%	38.7%	37.2%

Table 5.2: Comparison with the reference curve for real micrographs: average value of the absolute error (in pixels)

$$\begin{aligned}
\mathbf{H}_{\mathrm{3D,x}}(\mathbf{r}) &= \sum_{i=0}^{N-1} \mathbf{R}_i^t \left(\frac{\partial}{\partial x} \mathbf{H}_{f_i}(\mathbf{r}_i) \right) \mathbf{R}_i \\
&= \sum_{i=0}^{N-1} \mathbf{R}_i^t \left(\underbrace{\frac{\partial}{\partial x_i} \mathbf{H}_{f_i}}_{\mathbf{H}_{f_i,x_i}}(\mathbf{r}_i) \ \frac{\partial x_i}{\partial x} + \underbrace{\frac{\partial}{\partial z_i} \mathbf{H}_{f_i}}_{\mathbf{H}_{f_i,z_i}}(\mathbf{r}_i) \ \frac{\partial z_i}{\partial x} \right) \mathbf{R}_i \qquad (5.41) \\
&= \sum_{i=0}^{N-1} \left(\mathbf{R}_i^t \mathbf{H}_{f_i,x_i}(\mathbf{r}_i) \mathbf{R}_i \right) \underbrace{\frac{\partial x_i}{\partial x}}_{\mathbf{P}_i(0,0)} + \left(\mathbf{R}_i^t \mathbf{H}_{f_i,z_i}(\mathbf{r}_i) \mathbf{R}_i \right) \underbrace{\frac{\partial z_i}{\partial x}}_{\mathbf{P}_i(1,0)} \qquad (5.42)
\end{aligned}$$

Here, the matrices $\mathbf{H}_{f_i,x_i}$ and $\mathbf{H}_{f_i,z_i}$ are

$$\begin{aligned}
\mathbf{H}_{f_i,x_i}(\mathbf{r}_i) &= f * \left(\frac{\partial}{\partial x_i} \mathbf{G}_i(\mathbf{r}_i;\sigma) \right) \\
\mathbf{H}_{f_i,z_i}(\mathbf{r}_i) &= f * \left(\frac{\partial}{\partial z_i} \mathbf{G}_i(\mathbf{r}_i;\sigma) \right)
\end{aligned}$$

Plugging (5.42) into the integral I_1 in (5.43) , we get

$$I_2 = \sum_{i=0}^{N-1} \left[\mathbf{P}_i(0,0), \mathbf{P}_i(1,0) \right] \begin{bmatrix} \int_0^M \left(d\mathbf{r}^t(t) \mathbf{R}_i^t \mathbf{H}_{f_i,x_i}(\mathbf{r}_i(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \\ \int_0^M \left(d\mathbf{r}^t(t) \mathbf{R}_i^t \mathbf{H}_{f_i,z_i}(\mathbf{r}_i(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \end{bmatrix}$$

Thus, we get

$$\begin{bmatrix} \frac{\partial}{\partial c_x(k)} \\ \frac{\partial}{\partial c_y(k)} \\ \frac{\partial}{\partial c_z(k)} \end{bmatrix} E_{\text{image}} = -2\int_0^M \left(d\mathbf{r}(t)^t \mathbf{H}_{\text{3D}}(\mathbf{r}(t))\beta_p^{n\prime}(t-k)\right)dt$$
$$-2\sum_{i=0}^{N-1} \mathbf{P}_i^t \begin{bmatrix} \int_0^M \left(d\mathbf{r}^t(t)\mathbf{R}_i^t \mathbf{H}_{f_i,x_i}(\mathbf{r}_i(t))\mathbf{R}_i d\mathbf{r}(t)\beta_p^n(t-k)\right)dt \\ \int_0^M \left(d\mathbf{r}^t(t)\mathbf{R}_i^t \mathbf{H}_{f_i,z_i}(\mathbf{r}(t))\mathbf{R}_i d\mathbf{r}(t)\beta_p^n(t-k)\right)dt \end{bmatrix} \tag{5.43}$$

The evaluation of the matrices $\mathbf{H}_{f_i,x_i}$ and $\mathbf{H}_{f_i,y_i}$ necessitates the computation of the quantities $(f_i * g_{x_i x_i x_i})$, $(f_i * g_{x_i x_i y_i})$, $(f_i * g_{x_i y_i y_i})$ and $(f_i * g_{y_i y_i y_i})$ for each micrograph. For the first term in (5.43), we require the matrix $\mathbf{H}_{f_i}$, which in-turn needs the quantities $(f_i * g_{x_i x_i})$, $(f_i * g_{x_i y_i})$ and $(f_i * g_{y_i y_i})$. Note that all these quantities involve the convolution of f with the second and third order partial derivatives of the 2-D Gaussian; they can be pre-computed efficiently using separable linear filtering. We discretize the integrals for their evaluation. Thanks to the steerable implementation, the criterion and its partial derivatives can be computed exactly and efficiently.

	Along x	Along y	Along z	Total Error
Manual Tracing	2.336 ± 1.103	3.068 ± 0.81	2.116 ± 0.8799	4.481 ± 1.385
Snake Output	1.811 ± 0.943	2.480 ± 0.848	1.454 ± 0.830	3.48 ± 1.315
Performance Improvement	28.9%	23.7%	45.52%	28.7%

Table 5.3: Comparisons between trials: Repeatability (in pixels)

5.5.2 Partial derivatives of the internal energy

The internal energy term can be re-written as

$$\int_0^M |\mathbf{r}''(t)|^2 dt = \int_0^M \left(|x''(t)|^2 + |y''(t)|^2 + |z''(t)|^2\right)dt \tag{5.44}$$

We now consider the term $\int_0^M |x''(t)|^2 \, dt$ and simplify it as follows:

$$\begin{aligned}
\int_0^M |x''(t)|^2 \, dt &= \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} c_x(k) \, c_x(l) \int_0^M \beta_p^{n''}(t-k) \, \beta_p^{n''}(t-l) \, dt \\
&= \sum_{l=0}^{M-1} \sum_{k=-\infty}^{\infty} c_x(k) \, c_x(l) \underbrace{\int_{-\infty}^{\infty} \beta_p^{n''}(t-k) \, \beta_p^{n''}(t-l) \, dt}_{q(k-l)}
\end{aligned} \tag{5.45}$$

In the last step, we have used the periodicity of $\beta_p^{n''}(t-l)$ to extend the integral from $-\infty$ to ∞ and have transferred the periodicity of $\beta_p^{n''}(t-k)$ to the coefficient sequence c_k. Thanks to the curve representation using cubic B-spline functions, the sequence $q(k)$ is finitely supported and can be exactly computed. Thus, we obtain the partial derivatives of the internal energy term as

$$\begin{bmatrix} \frac{\partial}{\partial c_x(k)} \\ \frac{\partial}{\partial c_y(k)} \\ \frac{\partial}{\partial c_z(k)} \end{bmatrix} E_{\text{int}} = \sum_{k=-\infty}^{\infty} \begin{bmatrix} c_x(k) \\ c_y(k) \\ c_z(k) \end{bmatrix} q(k-l) \, .$$

The above equation amounts to a simple filtering of the coefficient sequence by the filter $q(n)$, assuming periodic boundary conditions.

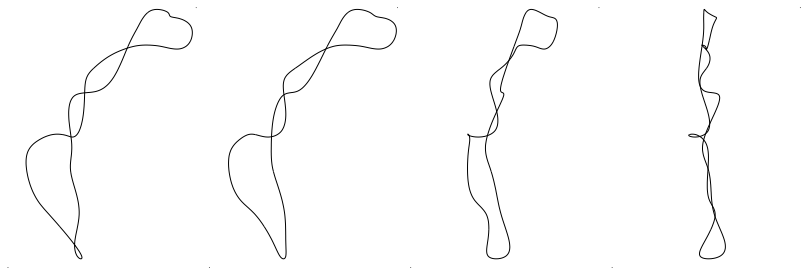
5.5.3 Partial derivatives of the constraint energy

Length constraint

Differentiating (5.38) with respect to $c_x(k)$:

$$\begin{aligned}
\frac{\partial}{\partial c_x(k)} E_{\text{const}} &= 2 \underbrace{\left(\int_0^M |\mathbf{r}'(t)| \, dt - \text{Length} \right)}_{\text{Error}} \int_0^M \frac{x'(t) \, \beta_p^n(t-k)}{|\mathbf{r}'(t)|} dt \\
&= 2 \, \text{Error} \int_{-\infty}^{\infty} \frac{x'(t+k) \, \beta^n(t)}{|\mathbf{r}'(t+k)|} dt
\end{aligned} \tag{5.46}$$

Here, Error is the difference between the current length of the curve and the expected one; the partial derivatives of the constraint energy is zero when the current length is the same as the expected one. The partial derivatives $\frac{\partial}{\partial c_y(k)} E_{\text{const}}$ and $\frac{\partial}{\partial c_z(k)} E_{\text{const}}$ is computed in a similar fashion.



(a) $-15°$ (b) $15°$ (c) $60°$ (d) $90°$

Figure 5.7: Reconstructed filaments for the micrograph pair in Fig. 5.1 at different viewing angles around the vertical axis. (a) and (b) correspond to the left and the right micrographs in Fig. 5.1

Evolving the curve with this term alone will cause its length to decrease or increase, (depending on the sign of Error) until Error = 0. Note that the integral (5.46) is limited over the support of the spline function. We discretize the integral for its evaluation.

Point constraint

Computing the partial derivatives of (5.39) in all generality would give a very complicated expression. To make the problem more tractable and to reduce its computational complexity, we make the assumption that the optimal constraint locations, $(t_i;\ i = 0 \ldots N_\mathrm{c} - 1)$, are known. In this case, (5.39) gets simplified to

$$E_\mathrm{c} = \sum_{i=0}^{N_\mathrm{c}-1} \left|\mathbf{r}\left(t_i\right) - \mathbf{r}_{\mathrm{c},i}\right|^2, \tag{5.47}$$

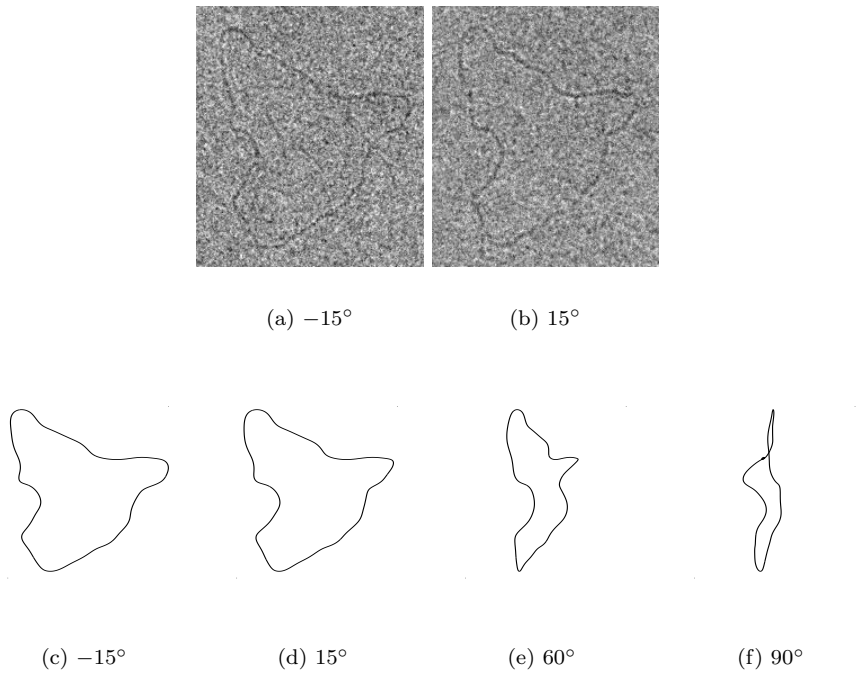
and its partial derivatives are given by

$$\begin{bmatrix} \partial E_\mathrm{c}/\partial c_{x,k} \\ \partial E_\mathrm{c}/\partial c_{y,k} \end{bmatrix} = \sum_{i=0}^{N_\mathrm{c}-1} \left(\begin{bmatrix} x_{\mathrm{c},i} \\ y_{\mathrm{c},i} \end{bmatrix} - \begin{bmatrix} x\left(t_i\right) \\ y\left(t_i\right) \end{bmatrix} \right) \beta^n\left(t_i - k\right) \tag{5.48}$$

Using the finite support of the scaling functions, we limit the sum to the relevant indices (we need to evaluate it only for $\{i|\ 0 < (t_i - k) < N_\mathrm{c}\}$). We resort to

a two-step strategy, where the snake is evolved using the above formulas for the derivatives, for a given t_i. The optimal parameters t_i are then re-estimated within the loop as:

$$t_i = \arg \min_{t \in [0,M]} \left| \mathbf{r}(t) - \mathbf{r}_{\mathrm{c},i} \right|; \quad i = 0 \ldots N_{\mathrm{c}} - 1 \tag{5.49}$$



(a) $-15°$ (b) $15°$

(c) $-15°$ (d) $15°$ (e) $60°$ (f) $90°$

Figure 5.8: Reconstructed filaments at different viewing angles around the vertical axis. (c) and (d) correspond to the left and the right micrographs in (a) and (b).

5.6 Experiments

In this section, we tested the performance of the algorithm on real data. Since the ground truth was not available, a reference curve was generated by magni-

fying the images 4 times and having the user carefully specifying a 3-D curve (by clicking on the stereo images).

To compare two 3-D curves $\mathcal{C}_a$ and $\mathcal{C}_b$, we choose the error metric

$$\mathcal{D}\left(\mathcal{C}_a, \mathcal{C}_b\right) = \frac{1}{2}\left(\frac{1}{M_a}\int_0^{M_a} D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right) dt + \frac{1}{M_b}\int_0^{M_b} D\left(\mathbf{r}_b\left(t\right), \mathcal{C}_a\right) dt\right), \tag{5.50}$$

where $\mathcal{C}_a \equiv \mathbf{r}_a\left(t\right);\ t \in [0, M_a]$ and $\mathcal{C}_b \equiv \mathbf{r}_b\left(t\right);\ t \in [0, M_b]$. The distance between a point $\mathbf{r}_a\left(t\right)$ and a curve $\mathcal{C}_b$ (denoted in the above equation as $D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right)$) is defined as the distance between $\mathbf{r}_a\left(t\right)$ and the closest point on $\mathcal{C}_b$:

$$D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right) = \min_{t\in[0,M]} \left\|\mathbf{r}_a\left(t\right) - \mathbf{r}_b\left(t\right)\right\| \tag{5.51}$$

We evaluate the distance metric by discretizing both curves.

We compared the performance of the snake algorithm with the manual tracing[6]. In our study, we used 5 stereo pairs and 2 independent users. For each stereo-pair, we performed 5 manual tracings each. These tracings were used as the initialization for the snake algorithm. These tracings and the snake output were compared with the corresponding reference curves to obtain the absolute errors. The average errors in the manual tracings and the snake-fitted curves are given in Table 5.5.1.

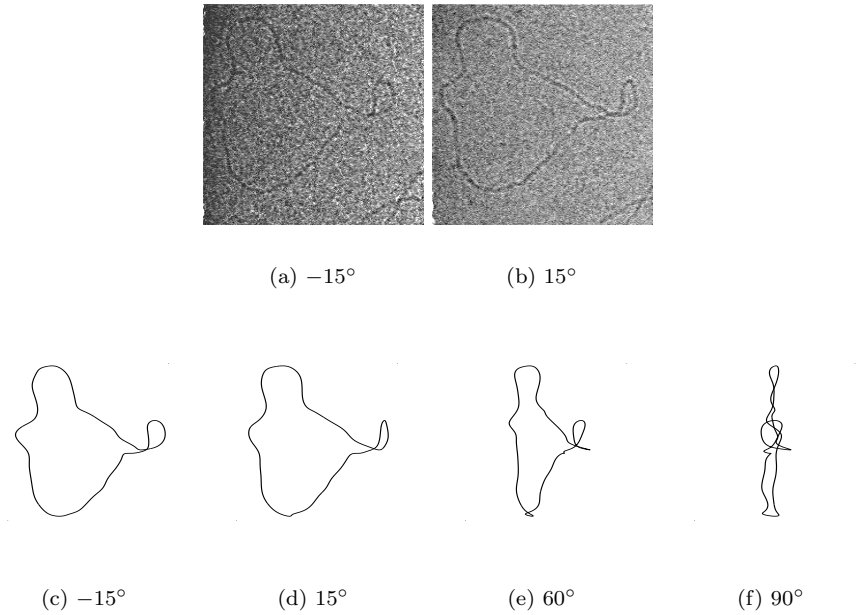
To study the inter-user variability, we compared the manual tracing on a pair-wise basis (there are 45 possible comparisons for each stereo pair). The same comparison was performed for the snake outputs. The experiment is performed only with real data since the ground truth is not required in this case. The results are shown in Table. 5.5.1.

Some examples of the 3-D reconstructions using our algorithm are shown in Fig. 5.7—Fig. 5.9. This illustrate the wide range of DNA configurations that may occur in nature as well as the difficulty of the problem.

5.7 Synopsis

We have presented a carefully engineered solution for the 3-D shape estimation of DNA molecules from stereo cryo-electron micrographs. We used a global 3-D model for the DNA filament and optimized its parameters such that its

[6]Unfortunately, the implementation of the flying cylinder algorithm was not available to us for comparison.



(a) $-15°$ (b) $15°$

(c) $-15°$ (d) $15°$ (e) $60°$ (f) $90°$

Figure 5.9: Reconstructed filaments at different viewing angles around the vertical axis. (c) and (d) correspond to the left and the right micrographs in (a) and (b).

2-D orthogonal projections matched with the micrographs. Since a direct implementation of this algorithm is computationally intensive, we approximated the model locally as an elongated blob. We derived an efficient algorithm to perform the local detection of such blobs.

To solve the local detection problem, we introduced the concept of projection-steerablity. Specifically, we derived a projection-steerable blob template whose 2-D projections can be represented as a linear combination of few basis functions. We derive an efficient algorithm for obtaining the likeliness of such a blob with a specific orientation at a certain point in 3-D space.

We used a 3-D B-spline curve model for the representation of the skeleton of the global DNA model. We show that the B-spline representation is optimal for the representation of smooth 3-D curves, if described in the constant arc-length

parameterization. We obtained simple expression for the an internal energy term that penalizes the average curvature magnitude by assuming a constant arc-length parametrization.

We used a conjugate gradients algorithm for the optimization of the curve parameters. Thanks to the projection-steerable blob detection algorithm and curve representation using compactly supported B-spline functions, all the directional derivatives of the snake energies are computed exactly and efficiently.

Appendix 5-A:Design of the 3-D projection-steerable elongated blob

We now derive a 3-D detector in $V_{3\mathrm{D}}$ that is good for the detection of a filament with a specific orientation (say along the x axis). Since the template is rotation-steerable by construction, we can steer it shape to any orientation exactly. An arbitrary function in $V_{3\mathrm{D}}$ is given by

$$h = a_0\, \partial_{xx}\, g_3 + a_1\, \partial_{yy}\, g_3 + a_2\, \partial_{zz}\, g_3 + a_3\, \partial_{xy}\, g_3 + a_4\, \partial_{xz}\, g_3 + a_5\, \partial_{yz}\, g_3 \quad (5.52)$$

Since a 1-D ridge oriented along the x axis is an even function along the axes, the terms $\partial_{xy}\, g_3$, $\partial_{xz}\, g_3$ and $\partial_{yz}\, g_3$ (they are odd functions) will not contribute to the ridge signal. Hence, we set a_3, a_4 and a_5 to zero.

We like to have a detector that is elongated along the x axis and narrow along the y and the z axis. The elongation along the axes can be measured by the magnitude of the second derivatives of h at the origin. For the detector to be maximally elongated along the x axis, we set $\partial_{xx} h|_{0,0,0}$ to zero:

$$\partial_{xx} h\left(\mathbf{r}\right)|_{\mathbf{r}=(0,0,0)} = \frac{1}{\sigma^4}\left(3a_0 + a_1 + a_2\right) = 0 \quad (5.53)$$

For the filter to be narrow along y and the z axes, we have to maximize the quantities

$$\partial_{yy} h\left(\mathbf{r}\right)|_{\mathbf{r}=(0,0,0)} = \left(a_0 + 3a_1 + a_2\right) \quad (5.54)$$

$$\partial_{zz} h\left(\mathbf{r}\right)|_{\mathbf{r}=(0,0,0)} = \left(a_0 + a_1 + 3a_2\right) \quad (5.55)$$

We maximize $\partial_{yy} h|_{0,0,0} + \partial_{zz} h|_{0,0,0}$ subject to (5.53) and the unit energy constraint:

$$\|h\left(\mathbf{r}\right)\|^2 = 3a_0^2 + 3a_1^2 + 3a_2^2 + 2a_0a_1 + 2a_0a_2 + 2a_1a_2 = 1 \quad (5.56)$$

We have removed the constants in all the above equations so as to simplify the formulas. The constants will not affect the shape of the detector. We solve this constrained optimization problem using Lagrange's multipliers by choosing the criterion as

$$\begin{aligned}\Lambda &= (2\,a_0 + 4\,a_1 + 4\,a_2) + \lambda_1\,(3a_0 + a_1 + a_2) + \\ &\qquad \lambda_2\left(3a_0^2 + 3a_1^2 + 3a_2^2 + 2a_0a_1 + 2a_0a_2 + 2a_1a_2 - 1\right) \end{aligned} \tag{5.57}$$

Setting the derivatives of Λ with respect to a_0, a_1 and a_2 to zero, we get:

$$\begin{bmatrix} 6 & 2 & 2 \\ 2 & 6 & 2 \\ 2 & 2 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \frac{1}{\lambda_2} \begin{bmatrix} 2 + 3\lambda_1 \\ 4 + \lambda_1 \\ 4 + \lambda_1 \end{bmatrix} \tag{5.58}$$

Solving this system of equations, we get

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \frac{1}{2\lambda_2} \begin{bmatrix} \lambda_1 \\ 1 \\ 1 \end{bmatrix} \tag{5.59}$$

Setting the above solution into the constraint (5.53), we get $\lambda_1 = -\frac{2}{3}$. Again by (5.56), we get $\lambda_2 = \sqrt{\frac{3}{5}}$. Thus the expression for the detector is given by

$$h\left(\mathbf{r}\right) = \sqrt{\frac{3}{20}}\left(\partial_{yy}\, g_3\left(\mathbf{r},\sigma\right) + \partial_{zz}\, g_3\left(\mathbf{r},\sigma\right) - \frac{2}{3}\,\partial_{xx}\, g_3\left(\mathbf{r},\sigma\right)\right) \tag{5.60}$$

Appendix 5-B:Projection of the optimal 3-D detector

The expression of the optimal filament detector, oriented along along $\mathbf{v}$, is given by

$$h_{\mathbf{v}}\left(\mathbf{r}\right) = \underbrace{g_{xx}\left(\mathbf{r};\sigma\right) + g_{yy}\left(\mathbf{r};\sigma\right) + g_{zz}\left(\mathbf{r}\right);\sigma}_{\text{Laplacian of } g\left(\mathbf{r}\right)} - \frac{5}{3}\, g_{\mathbf{vv}}\left(\mathbf{r};\sigma\right). \tag{5.61}$$

Note that in this case $\mathbf{A} = \mathbf{I}_3 - \frac{5}{3}\,\mathbf{v}\mathbf{v}^t$, where $\mathbf{I}_3$ is the 3x3 identity matrix.

Now, using (5.22), we obtain the corresponding $\mathbf{B}_i$ matrix as

$$\mathbf{B}_i = \underbrace{\mathbf{P}_i^t\mathbf{P}_i}_{\mathbf{I}_2} - \frac{5}{3}\,\underbrace{\left(\mathbf{P}_i\mathbf{v}\right)\left(\mathbf{P}_i\mathbf{v}\right)^t}_{\mathbf{v}_i\,\mathbf{v}_i^t}, \tag{5.62}$$

which implies that the Fourier transform of the 2-D projection of the template is given by

$$\hat{h}_i\left(\boldsymbol{\omega}_i\right) = -\left(\|\boldsymbol{\omega}_i\|^2 - \frac{5}{3}\left\langle \boldsymbol{\omega}_i, \mathbf{v}_i\right\rangle^2\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right). \tag{5.63}$$

The above expression can also be written as

$$\hat{h}_i\left(\boldsymbol{\omega}_i\right) = -\left(\|\boldsymbol{\omega}_i\|^2 - \frac{5}{3}\,\mathbf{v}_i^t \underbrace{\begin{bmatrix} \boldsymbol{\omega}_{x_i}^2 & \boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} \\ \boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} & \boldsymbol{\omega}_{z_i}^2 \end{bmatrix}}_{\boldsymbol{\omega}_i\,\boldsymbol{\omega}_i^t} \mathbf{v}_i\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right). \tag{5.64}$$

Since $\|\mathbf{v}_i\|^2 + v_{y_i}^2 = \|v\|^2 = 1$, we rewrite this expression as

$$\begin{aligned} \hat{h}_i\left(\boldsymbol{\omega}_i\right) &= -\Bigg(\|\boldsymbol{\omega}_i\|^2 v_{y_i}^2 + \mathbf{v}_i^t \begin{bmatrix} \boldsymbol{\omega}_{x_i}^2 + \boldsymbol{\omega}_{z_i}^2 & 0 \\ 0 & \boldsymbol{\omega}_{x_i}^2 + \boldsymbol{\omega}_{z_i}^2 \end{bmatrix} \mathbf{v}_i \\ &\qquad - \frac{5}{3}\mathbf{v}_i^t \begin{bmatrix} \boldsymbol{\omega}_{x_i}^2 & \boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} \\ \boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} & \boldsymbol{\omega}_{z_i}^2 \end{bmatrix} \mathbf{v}_i\Bigg) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right) \\ &= -\left(\|\boldsymbol{\omega}_i\|^2 v_{y_i}^2 - \mathbf{v}_i^t \begin{bmatrix} \boldsymbol{\omega}_{z_i}^2 - \frac{2}{3}\boldsymbol{\omega}_{x_i}^2 & -\frac{5}{3}\boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} \\ -\frac{5}{3}\boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} & \boldsymbol{\omega}_{x_i}^2 - \frac{2}{3}\boldsymbol{\omega}_{z_i}^2 \end{bmatrix} \mathbf{v}_i\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right) \end{aligned} \tag{5.65}$$

which is then modified to

$$\hat{h}_i\left(\boldsymbol{\omega}_i\right) = -\left(\mathbf{v}^t \mathbf{R}_i \begin{bmatrix} \boldsymbol{\omega}_{z_i}^2 - \frac{2}{3}\boldsymbol{\omega}_{x_i}^2 & 0 & -\frac{5}{3}\boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} \\ 0 & \boldsymbol{\omega}_{x_i}^2 + \boldsymbol{\omega}_{z_i}^2 & 0 \\ -\frac{4}{3}\boldsymbol{\omega}_{x_i}\boldsymbol{\omega}_{z_i} & 0 & \boldsymbol{\omega}_{x_i}^2 - \frac{2}{3}\boldsymbol{\omega}_{z_i}^2 \end{bmatrix} \mathbf{R}_i \mathbf{v}\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right), \tag{5.66}$$

where $\mathbf{R}_i$ is the rotation matrix given by (5.9) and (5.11). Finally, computing the inverse Fourier transform, we get

$$\mathcal{P}_i\left(h_{\mathbf{v}}\left(\mathbf{r}\right)\right) = \mathbf{v}^t\,\mathbf{R}_i\,G\left(\mathbf{r}_i; \sigma\right)\mathbf{R}_i\,\mathbf{v}, \tag{5.67}$$

where

$$G\left(\mathbf{r}_i; \sigma\right) = \begin{bmatrix} \left(\partial_{z_i z_i} - \frac{2}{3}\partial_{x_i x_i}\right) & 0 & -\left(\frac{5}{3}\partial_{x_i z_i}\right) \\ 0 & \left(\partial_{x_i x_i} + \partial_{z_i z_i}\right) & 0 \\ -\left(\frac{5}{3}\partial_{x_i z_i}\right) & 0 & \left(\partial_{z_i z_i} - \frac{2}{3}\partial_{x_i x_i}\right) \end{bmatrix} g_3\left(\mathbf{r}_i; \sigma\right) \tag{5.68}$$

Appendix 5-C: Simplification of the curvature term in the internal energy

The square of the curvature of the curve at a point $\mathbf{r}(t)$ is expressed in the vector form as

$$|\kappa(\mathbf{r})|^2 = \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'')}{|\mathbf{r}'|^6} \tag{5.69}$$

Assuming the parameter t to be the curvilinear abscissa, we have $|\mathbf{r}'(t)| = c,\ \forall t$. Making use of the vector identity $\mathbf{a}.(\mathbf{b} \times \mathbf{c}) = \mathbf{c}.(\mathbf{a} \times \mathbf{b})$, the numerator of (5.69) is rewritten as

$$\begin{aligned}
(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'') &= \mathbf{r}'' \cdot (\mathbf{r}' \times \mathbf{r}'' \times \mathbf{r}'') \\
&= \mathbf{r}'' \cdot (\mathbf{r}''(\mathbf{r}' \cdot \mathbf{r}') - \mathbf{r}'(\mathbf{r}' \cdot \mathbf{r}'')) \\
&= |\mathbf{r}''|^2 |\mathbf{r}'|^2 - |\underbrace{\mathbf{r}'' \cdot \mathbf{r}'}_{d(\mathbf{r}'^2)=0}|^2,
\end{aligned}$$

where we have used the identity $\mathbf{a} \times \mathbf{b} \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{b} \cdot \mathbf{c})\mathbf{a}$. So the expression for the curvature therefore simplifies to

$$|\kappa(\mathbf{r})|^2 = \frac{|\mathbf{r}''^2|}{|\mathbf{r}'|^4} = \frac{|\mathbf{r}''^2|}{k^2} \tag{5.70}$$

Chapter 6

Sampling of Periodic Signals: A Quantitative Error Analysis

We introduced a new scheme for shape extraction and processing in Chapter 1. Since a contour model is central to our algorithm (c.f. Fig. 1.4) the optimality of this model is crucial to the whole algorithm.

In biomedical imaging, it is common to have segmentation problems where we have a-priori knowledge of the shape. A typical scenario is where the average shape is known. We can use this information to tune the contour model to the shape. Specifically, we deal with the the choice of the number of knots and basis functions. To this end, we perform a theoretical analysis of approximation error in parametric curve representation and derive accurate performance bounds in this chapter[1].

6.1 Introduction

Classical sampling theory deals with the problem of reconstructing or approximating a signal $s(t)$ from a set of uniform samples or measurements. In its generalized version, the reconstructed approximation [24] is

$$s_h(t) = \sum_{k=-\infty}^{\infty} c_k \varphi\left(\frac{t}{h} - k\right), \tag{6.1}$$

[1]Based on the article "M.Jacob, T.Blu, M.Unser, *IEEE Transactions on Signal Processing, vol. 50, pp. 1153-1159, May 2002*".

where the underlying basis functions are rescaled translates of the generating[2] function φ; h is the sampling step. The generator can be selected so as to yield bandlimited (e.g., $\varphi = \text{sinc}$), spline, or wavelet representations of signals. The expansion coefficients c_k are either determined from the uniform samples of the input signal $s(kh)$ (interpolation or quasi-interpolation) or from a sequence of inner products with a suitable sequence of analysis functions [24]. This theory is well developed for the case in which the input signal is in $L_2(\mathbb{R})$, which also implies that it is defined over the whole real line. The approximation quality depends on the sampling step h, the type of algorithm used (e.g., interpolation vs. projection), and most importantly, on the choice of the generating function φ. This can be quantified rather precisely, thanks to the availability of sharp mean square error estimates in the $L_2(\mathbb{R})$ setting [26, 99]. Bounds are also available for the L_∞ approximation error (worst case scenario) [100].

In this chapter, we are interested in the case where the input signal $s(t)$ is periodic, which is an assumption that is commonly made in practice. One example, where the periodic representation is especially relevant, is the parametric representation of closed curves in terms of splines [17, 38, 15, 37] or Fourier basis functions [101]. Assuming the period T to be an integer multiple of the sampling step ($T = Nh$)[3], it is straightforward to adapt most of the L_2 techniques to the periodic case by simply considering periodized basis functions and by redefining the inner product accordingly [34] (see section 6.2). However, the error analysis for signals in $L_2(\mathbb{R})$ is not directly applicable because the square modulus of the Fourier transform is not defined for periodic signals.

The quantitative error analysis of periodic signals is the main focus of this chapter. In particular, we will derive a general predictive error formula that depends on the Fourier coefficients of $s(t)$. Interestingly, the formula bears a strong resemblance to the error expression of signals in $L_2(\mathbb{R})$. However, the recipe is different although the ingredients are more or less the same as in [26]; the average least squares error is obtained as a discrete sum of the Fourier series coefficients as opposed to a continuous integral in [26]. We also study the behavior of the approximation as the sampling step goes to zero.

[2]When the function satisfies a two-scale relation [22], it is called a scaling function. (e.g., splines, Daubechies functions or sinc)

[3]If we choose $T = Nh$, the resulting representation is assured to be T-periodic. Otherwise, this property is not satisfied in general.

6.2 Preliminaries

6.2.1 Notations

We denote the Fourier transform of a continuous signal $s(t)$ as

$$\hat{s}(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \tag{6.2}$$

6.2.2 Sampling of Periodic Signals

The general formula for determining the expansion coefficients in (6.1) is

$$c_k = \int_{-\infty}^{\infty} s(\xi)\, \tilde{\varphi}\left(\frac{\xi}{h} - k\right) d\frac{\xi}{h}, \tag{6.3}$$

where $\tilde{\varphi}$ is an appropriate analysis function. The usual setting for this formula is $s \in L_2(\mathbb{R})$ (finite energy signals). In particular, one can show that $c_k \in \ell_2$ when $\hat{\tilde{\varphi}}$ is bounded and when s has at least $r > \frac{1}{2}$ derivatives in the L_2 sense [26]. However (6.3) also works for more general cases. For instance, if $s(t)$ is bounded, then the c_k's will be bounded as well, provided that $\tilde{\varphi}$ is a distribution[4] of order 0.

We assume that $s(t)$ is T-periodic and that $T = Nh$, where N is a positive integer. Under those conditions, the sequence c_k defined by (6.3) is periodic as well, with period N. Furthermore, we can rewrite the synthesis and analysis equations (6.1) and (6.3) using N-periodized functions as

$$s_N(t) = \sum_{k=0}^{N-1} c_k \varphi_p\left(\frac{t}{h} - k\right) \tag{6.4}$$

$$c_k = \int_0^T s(\xi)\, \tilde{\varphi}_p\left(\frac{\xi}{h} - k\right) d\frac{\xi}{h}, \tag{6.5}$$

where

$$\varphi_p(t) = \sum_{l=-\infty}^{\infty} \varphi(t - lN) \tag{6.6}$$

[4] $\tilde{\varphi}$ is a distribution of order n iff $|\langle \tilde{\varphi}, s\rangle| \leq C \max_{k\leq n} \sup_x \left|s^{(k)}(x)\right|$, where C is a constant [102, pp. 24–25], [103, def. 1.3.1]; e.g., the Dirac delta distribution $\delta(x)$ is of order 0. An absolutely integrable function $\tilde{\varphi}$ can also be identified as a distribution of order 0.

Equation (6.5) calls for the definition of an inner product in $L_2([0,T])$: we denote the $L_2([0,T])$ inner product between two functions $s_1(t), s_2(t) \in L_2([0,T])$ as

$$\langle s_1(t), s_2(t)\rangle_{L_2([0,T])} = \frac{1}{T}\int_{-\infty}^{\infty} s_1(t)s_2(t)dt. \tag{6.7}$$

The corresponding norm is written as $||.||_{L_2([0,T])}$. We show in Appendix 6-A that a sufficient condition for φ_p to be in $L_2([0,T])$ is that φ be absolutely integrable ($\varphi \in L_1(\mathbb{R})$) and the discrete Fourier transform of the autocorrelation sequence

$$\hat{a}_\varphi(\omega) = \sum_{k=-\infty}^{\infty} |\varphi(\omega + 2k\pi)|^2 \tag{6.8}$$

is bounded. Under those assumptions, $s_N(t) \in L_2([0,T])$ provided of course that the c_k's are bounded. While these relatively mild conditions are satisfied by most generating functions used in practice, they are not applicable to the classical case $\varphi = \text{sinc}$, which present some difficulties i.e., $\text{sinc} \notin L_1(\mathbb{R})$. This case is dealt with in the next section.

Combining (6.4) and (6.5), we get

$$\begin{aligned} s_N(t) &= \mathcal{Q}_N s(t) \\ &= \sum_{k=0}^{N-1}\left[\int_0^T s(\xi)\,\tilde{\varphi}_p\left(\frac{\xi}{h}-k\right)d\frac{\xi}{h}\right]\varphi_p\left(\frac{t}{h}-k\right), \end{aligned} \tag{6.9}$$

where $\mathcal{Q}_N$ is the approximation operator. This linear operator is a projector if and only if the functions φ and $\tilde{\varphi}$ are bi-orthogonal; i.e., $\langle \varphi(t-k), \tilde{\varphi}(t-l)\rangle = \delta_{k-l}$ [25]. In this case $s_N(t)$ is a consistent reconstruction of the measurements c_k.

As we frequently use Parseval's relation, we now recall it. It relates the $L_2([0,T])$ inner product between two functions $s_1(t), s_2(t) \in L_2([0,T])$ to their Fourier series coefficients as

$$\langle s_1(t), s_2(t)\rangle_{L_2([0,T])} = \frac{1}{T}\int_0^T s_1(t)s_2(t)dt = \sum_{k=-\infty}^{\infty} S_1(k)S_2(k)^* \tag{6.10}$$

Using this expression, the $L_2([0,T])$ norm of $s(t) \in L_2([0,T])$ can be written as

$$\|s\|^2_{L_2([0,T])} = \frac{1}{T}\int_0^T |s(t)|^2\,dt = \sum_{k=-\infty}^{\infty} |S(k)|^2 \tag{6.11}$$

6.3 Fourier Series Representation

Bandlimited periodic signals can be represented as (6.4) by choosing $\varphi = \text{sinc}$. However, due to the slow decay of sinc, φ_p does not converge when N is even. However, when N is odd φ_p converges to a well defined function in $L_1([0,T])$. In this case, the signal representation can be reformulated as a Fourier series. Hence, we briefly review the Fourier series description of a periodic signal, when the period is odd.

A T-periodic signal ($s(t) \in L_2([0,T])$) can be expanded as

$$s(t) = \sum_{k=-\infty}^{\infty} S(k) e^{j\frac{2\pi k t}{T}}, \tag{6.12}$$

where the Fourier series coefficients $S(k)$ are obtained as

$$S(k) = \frac{1}{T}\int_0^T s(t) e^{-j\frac{2\pi k t}{T}} dt \tag{6.13}$$

In most practical applications, the function $s(t)$ is not directly available. Usually, it is only known through its samples $\{s(lh)\}_{l=0,\ldots N-1}$. In such cases, one often assumes that $s(t)$ is bandlimited and hence approximates the coefficients $S(k)$ with the N point DFT of $\{s(lh)\}$ for $k = -\lfloor \frac{N}{2} \rfloor .. \lfloor \frac{N}{2} \rfloor$ and 0 otherwise.

The corresponding continuous signal $s_N(t)$ is nothing but the periodized sinc interpolation of the samples[32, 104]. The corresponding sinc interpolation with a zooming factor M is implemented efficiently by computing the FFT of the input sequence and performing a larger size IFFT with zero padding the transform upto size NM. This representation turns out to be a special case of (6.9) with $\varphi = \text{sinc}$ and $\tilde{\varphi} = \delta$ — the Dirac's delta distribution.

6.4 Computation of the Square Error

The space spanned by the generating functions is not shift-invariant in general. Hence, the approximation error at a scale h is dependent on a time shift of the function $s(t)$. The shifted function is denoted by $s_\tau(t) = s(t-\tau)$.

The mean square approximation error for a shifted function s_τ is given by

$$\begin{aligned} \gamma_s(\tau, N) &= \frac{1}{T}\int_0^T |s_\tau(t) - \mathcal{Q}_N s_\tau(t)|^2 dt \\ &= ||s_\tau - \mathcal{Q}_N s_\tau(t)||^2_{L_2([0,T])} \end{aligned} \tag{6.14}$$

As the period of the signal is an integer multiple of the sampling step, $\gamma_s(\tau, N)$ is also h periodic in τ. In most applications, the exact phase of the signal is not known. Hence, we are interested in obtaining a measure of the error that is averaged over τ. This average error is given by

$$\eta_s(N) = \sqrt{\frac{1}{h}\int_0^h \gamma_s(\tau, N)d\tau} \tag{6.15}$$

The following theorem, which is the main result of this chapter, gives an explicit expression for the mean error $\eta_s(N)$.

Theorem 1 *Let $s(t)$ be a T-periodic signal with the Fourier-series coefficients $S(k)$. The mean square approximation error incurred in approximating $s(t)$ as in (6.9) is given by*

$$\eta_s(N) = \sqrt{\sum_{k=-\infty}^{\infty} |S(k)|^2 E\left(\frac{2\pi k}{N}\right)}, \tag{6.16}$$

where the approximation kernel $E(\omega)$ depends only on φ and $\tilde{\varphi}$ and assumes the expression

$$\begin{aligned} E(\omega) &= \left|1 - \hat{\tilde{\varphi}}(\omega)^* \hat{\varphi}(\omega)\right|^2 \\ &\quad + |\hat{\tilde{\varphi}}(\omega)|^2 \sum_{n\neq 0} |\hat{\varphi}(\omega + 2n\pi)|^2 \end{aligned} \tag{6.17}$$

$$= \underbrace{1 - \frac{|\hat{\varphi}(\omega)|^2}{\hat{a}_\varphi(\omega)}}_{E_{min}(\omega)} + \underbrace{\hat{a}_\varphi(\omega)\left|\hat{\tilde{\varphi}}(\omega) - \hat{\varphi}_d(\omega)\right|^2}_{E_{res}(\omega)}, \tag{6.18}$$

where $\hat{\varphi}_d(\omega) = \frac{\hat{\varphi}(\omega)}{\hat{a}_\varphi(\omega)}$.

The proof is given in Appendix 6-B.

Note that this kernel is identical to the one obtained in the case of signals in $L_2(\mathbb{R})$ [26]. The main difference with the $L_2(\mathbb{R})$ case is that the expression of the error (6.16) is a discrete sum as opposed to a continuous integral [26]

$$\eta_s(T) = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{s}(\omega)|^2 E(\omega T)\, d\omega}. \tag{6.19}$$

Here $\hat{s}(\omega)$ is the Fourier transform of the signal $s(t) \in L_2(\mathbb{R})$ and T is the sampling step.

Given a reconstruction space, the error kernel attains its minimum possible value $E_{\min}(\omega)$ for all ω when $\tilde{\varphi}$ is the dual of φ. It is obvious from (6.18) as $E_{\text{res}}(\omega) \geq 0$ and $E_{\min}(\omega)$ depends only on φ. This case corresponds to the minimum error approximation (orthogonal projection), as in the case of signals in $L_2(\mathbb{R})$ [105]. The second part E_{res} accounts for the additional error encountered for not choosing the optimal analysis function $\tilde{\varphi} = \varphi_d$. When $\tilde{\varphi}$ is bi-orthogonal to $\tilde{\varphi}$ but $\tilde{\varphi} \neq \varphi_d$, then the corresponding operator $\mathcal{Q}_N$ is called an oblique projection.

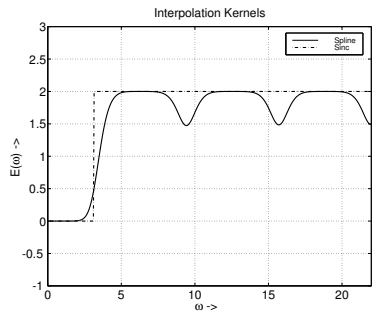


Figure 6.1: Error kernels for cubic B-Spline and Sinc representation.

6.5 Asymptotic Performance

The asymptotic performance of the representation is determined by the behavior of the kernel close to the origin. Using the Taylor-series expression of the kernel, we show that, for the minimum approximation error to decay as $\mathcal{O}\left(\frac{1}{N^L}\right)$ as the number of sampling points $N \to \infty$, we need $\hat{\varphi}(0) \neq 0$ and $\hat{\varphi}^{(n)}(2k\pi) = 0$, $\forall k \in \mathbb{Z} \setminus \{0\}$ for $n = 0, 1 \ldots L-1$. These are precisely the Strang-Fix conditions of order L [22]; a φ that satisfy these conditions is called as an Lth order generating function.

In the following theorem, we give the asymptotic bound for the projection error. Note that the projection need not be orthogonal [25].

Theorem 2 *Let φ and $\tilde{\varphi}$ be two mutually bi-orthogonal generating functions. Then the oblique projection error in approximating an L-times differentiable*

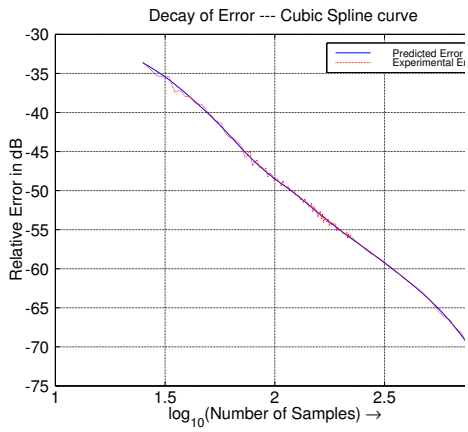


Figure 6.2: Decay of the cubic spline interpolation error for the map of Switzerland as a function of the number of samples.

function $s(t)$ as in (6.9) decays as $\mathcal{O}\left(\frac{1}{N^L}\right)$ as $N \to \infty$ iff φ is an Lth order generating function. If φ satisfies the Lth order Strang-Fix conditions, the error in approximation as $N \to \infty$ is asymptotically given as

$$\begin{aligned} \eta_s(N) &= C_{\varphi,\tilde{\varphi}} \, || \, (2\pi k)^L \, S(k) ||_{\ell_2} \left(\frac{1}{N}\right)^L + \mathcal{O}\left(\frac{1}{N^{L+1}}\right) \\ &= C_{\varphi,\tilde{\varphi}} T^L ||s^{(L)}||_{L_2[0,T)} \left(\frac{1}{N}\right)^L + \mathcal{O}\left(\frac{1}{N^{L+1}}\right), \end{aligned} \tag{6.20}$$

where, $s^{(L)}$ is the Lth derivative of s and the constant is given by the expression

$$C_{\varphi,\tilde{\varphi}} = \frac{1}{L!} \sqrt{\sum_{k \neq 0} |\hat{\varphi}^{(L)}(2\pi k)|^2 + |m^L_{\varphi_d} - m^L_{\tilde{\varphi}}|^2}. \tag{6.21}$$

Here, $\hat{\varphi}^{(L)}$ denotes the Lth derivative of φ and $m^L_u = \int x^L u(x) dx$; u is either $\tilde{\varphi}$ or φ_d.

The proof is given in Appendix 6-C.

Note that this result is almost the same as the bound derived in [106], except that the present norm is defined for $L_2([0,T])$ as opposed to $L_2(\mathbb{R})$ as in [106]. The minimum value attainable by this constant $C_\varphi^- = \frac{1}{L!}\sqrt{\sum_{k\neq 0}|\hat{\varphi}^{(L)}(2\pi k)|^2}$ is independent of the analysis function. This value is achieved when we have $m_{\varphi_d}^L = m_{\tilde{\varphi}}^L$.

6.6 Experimental verification of the error formula

In this section, we validate the expression for the error given by Theorem 1 experimentally. We compare the measured errors to the ones predicted by the theory for the approximation of a reference shape as a function of the sampling step h, or equivalently, the number of the samples N.

Our reference shape (Switzerland) is polygonal with 807 edges and is represented using two periodic functions $x(t)$ and $y(t)$. For each experiment, the initial model $(x(t), y(t))$ was resampled to a specified number of points.

We considered two types of approximations: (1) a cubic spline interpolation with $\varphi = \beta^3$ (cubic spline) and (2) a bandlimited one with $\varphi = \text{sinc}$. Note that the second approach is equivalent to a truncated Fourier approximation. In fact, we used an IFFT padded with zeros to generate the bandlimited interpolation functions at the required scale.

The comparisons between the experimental errors and the ones predicted by the theory are given in Fig. 6.2 and Fig. 6.3, respectively. It can be seen for both the graphs (Fig. 6.2 and Fig. 6.3) that the experimental error (for $\tau = 0.5$) is in good agreement with the theoretical prediction. The experimentally obtained curve of $\gamma_s(\tau, N)$ for $\tau = 0.5$ oscillates around the theoretically predicted curve of $\eta_s(N)$. This is because the theoretical prediction is an average of $\gamma_s(\tau, N)$ over all τ's.

From Fig. 6.4, it can be seen that the spline interpolation of curves perform slightly better (around 1 dB) than the sinc interpolation. This behavior can be explained with the aid of the error kernel we have just derived. We can see from Fig. 6.1 that the spline kernel has lower values as compared to the sinc interpolation kernel when $\omega > \pi$. Hence, at low sampling rates (when the signal has some non-negligible frequency components above π), spline interpolation will usually outperform the sinc one. The differences tend to vanish as the sampling step decreases.

The map of Switzerland interpolated from 45 samples using the spline and sinc functions are shown in Fig. 6.5. It can be seen that at some places, the

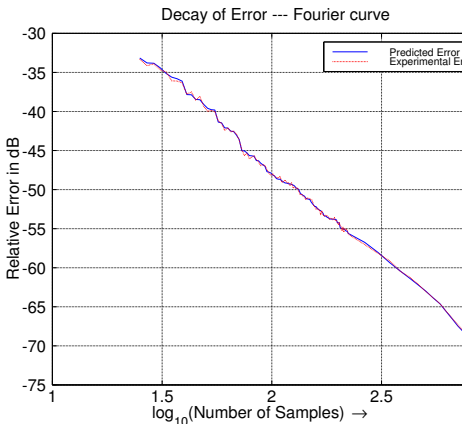


Figure 6.3: Decay of the sinc interpolation error for the map of Switzerland as a function of the number of samples.

sinc representation results in looping curves. This effect is less likely with the spline representation due to the more local behavior of spline interpolation.

6.7 Conclusion

We have derived an exact expression of the mean error in representing a periodic signal in a generating function basis. This expression may be useful for comparing different generating functions and for choosing the right one for an application. We have experimentally verified the expression; the experimental curves are in excellent agreement with the theoretical predictions. Using the expression for the error, we also analyzed the behavior of the approximation scheme as the sampling step approaches zero.

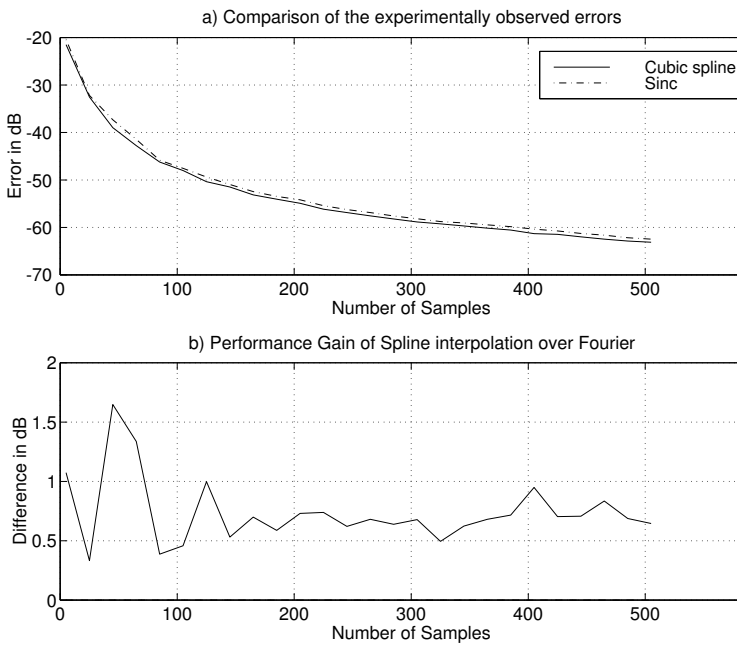


Figure 6.4: Comparison of spline and sinc interpolation.

Appendix 6-A: Sufficient condition for $\varphi_p \in L_2([0,T])$

$\varphi \in L_1(\mathbb{R})$ implies that $\varphi_p \in L_1([0,T])$ and that

$$\varphi_p(x) = \sum_{l\in\mathbb{Z}} \varphi(x - lN) = \frac{1}{N}\sum_{k\in\mathbb{Z}} \hat{\varphi}\left(\frac{2k\pi}{N}\right) e^{j\frac{2\pi kt}{N}} \tag{6.22}$$

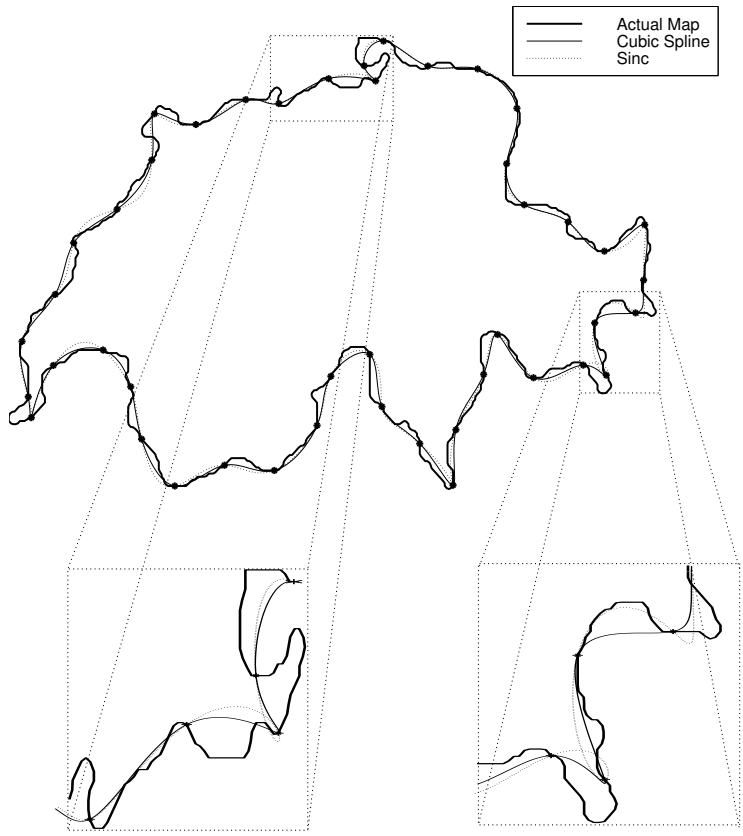


Figure 6.5: Actual Map of Switzerland represented using 807 edges is resampled to 45 points (indicated by dots). These points are then interpolated using cubic spline and sinc functions. The graphs below are the zoomed portions of the corresponding positions of the main graph which illustrates the looping nature of sinc interpolation.

in the sense of distributions [107]. Now the r.h.s of (6.22) is in $L_2([0,T])$ iff

$$\sum_{k\in\mathbb{Z}} \left|\hat{\varphi}\left(\frac{2k\pi}{N}\right)\right|^2 < \infty, \tag{6.23}$$

which is ensured if the the Fourier transform of the autocorrelation

$$\hat{a}_{\varphi}(\omega) = \sum_{k\in\mathbb{Z}} \left|\hat{\varphi}\left(\omega + 2k\pi\right)\right|^2 \tag{6.24}$$

is bounded for all ω. Thus $\varphi_p \in L_2\left([0,T]\right)$.

Appendix 6-B: Computation of the Square Error

Expanding (6.14), we get

$$\begin{aligned}\gamma_s(\tau, N) &= \frac{1}{T}\int_0^T \left[s_\tau(t)\right]^2 dt + \frac{1}{T}\int_0^T \left[\mathcal{Q}_N s_\tau(t)\right]^2 dt \\ &\quad -\frac{2}{T}\int_0^T s_\tau(t)\mathcal{Q}_N s_\tau(t) dt \end{aligned} \tag{6.25}$$

1. Using Parseval's theorem, the first term of (6.25) reduces to

$$\frac{1}{T}\int_0^T \left[s_\tau(t)\right]^2 dt = \sum_{k=-\infty}^{\infty} \left|S_\tau(k)\right|^2 = \sum_{k=-\infty}^{\infty} \left|S(k)\right|^2$$

2. To compute the second term of (6.25), we first compute the Fourier coefficients of $\mathcal{Q}_N s_\tau(t)$. From (6.4), they are obtained as

$$R_N(m) = \sum_{k=0}^{N-1} c_k \left[\frac{1}{T}\int_0^T \varphi_p\left(\frac{t}{h} - k\right) e^{-\frac{2\pi m t}{T}} dt\right] \tag{6.26}$$

We make a change of variables as $t = \frac{t}{h} - k$ and rearrange the terms to get

$$R_N(m) = \underbrace{\left[\int_{-\infty}^{\infty} \varphi\left(t\right) e^{-\frac{2\pi m t}{N}} dt\right]}_{\hat{\varphi}\left(\frac{2\pi m}{N}\right)} \frac{1}{N}\sum_{k=0}^{N-1} c_k e^{\frac{2\pi m k}{N}} \tag{6.27}$$

We now consider the expression of c_k from (6.5); the $L_2([0,T])$ inner product can be expressed in terms of the corresponding Fourier coefficients

using Parseval's theorem. Hence,

$$\begin{aligned}
\frac{1}{N}\sum_{k=0}^{N-1} c_k e^{\frac{2\pi m k}{N}} &= \frac{1}{N}\sum_{k=0}^{N-1} \underbrace{\sum_{l=-\infty}^{\infty} S_\tau(l)\, \hat{\hat{\varphi}}\left(\frac{2\pi l}{N}\right)^* e^{-\frac{j2\pi k l}{N}}}_{c_k} e^{\frac{2\pi m k}{N}} \\
&= \sum_{l=-\infty}^{\infty} S_\tau(l)\, \hat{\hat{\varphi}}\left(\frac{2\pi l}{N}\right)^* \underbrace{\frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{j2\pi k(l-m)}{N}}}_{\sum_{k=-\infty}^{\infty}\delta_{(l-m-kN)}} \\
&= \sum_{k=-\infty}^{\infty} S_\tau\left(m+kN\right) \hat{\hat{\varphi}}\left(\frac{2\pi\left(m+kN\right)}{N}\right)^* \qquad (6.28)
\end{aligned}$$

Combining (6.27) and (6.28), we get

$$R_N(m) = \hat{\varphi}\left(\frac{2\pi m}{N}\right)\left[\sum_{k=-\infty}^{\infty} S_\tau(m+kN)\, \hat{\hat{\varphi}}\left(\frac{2\pi(m+kN)}{N}\right)^*\right]$$

We now use Parseval's theorem to get

$$\frac{1}{T}\int_0^T \left[\mathcal{Q}_N s_\tau(t)\right]^2 dt = \sum_{m=-\infty}^{\infty} |R_N(m)|^2 \qquad (6.29)$$

Making use of the relation between the Fourier coefficients of the shifted function and the actual one $\left(S_\tau(k) = e^{-j\frac{2\pi k\tau}{T}} S(k)\right)$, we rewrite (6.29) as

$$\begin{aligned}
\frac{1}{T}\int_0^T \left[\mathcal{Q}_N s_\tau(t)\right]^2 &= \sum_{m=-\infty}^{\infty} \left|e^{-j\frac{2\pi m\tau}{T}} x_m(\tau)\right|^2 \left|\hat{\varphi}\left(\frac{2\pi m}{N}\right)\right|^2 \\
&= \sum_{m=-\infty}^{\infty} |x_m(\tau)|^2 \left|\hat{\varphi}\left(\frac{2\pi m}{N}\right)\right|^2
\end{aligned}$$

Here, $x_m(\tau)$ is the $h = \frac{T}{N}$ periodic function with the expression

$$x_m(\tau) = \sum_{k=-\infty}^{\infty} \underbrace{S(m+kN)\, \hat{\hat{\varphi}}\left(\frac{2\pi(m+kN)}{N}\right)^*}_{X_m(k)} e^{-j\frac{2\pi k\tau}{h}}$$

Averaging this expression over τ, $\frac{1}{h}\int_0^h \frac{d\tau}{T}\int_0^T |\mathcal{Q}_N s_\tau(t)|^2 dt$ becomes

$$\begin{aligned} & \sum_{m=-\infty}^{\infty} \left|\hat{\varphi}\left(\frac{2\pi m}{N}\right)\right|^2 \cdot \left[\frac{1}{h}\int_0^h |x_m(\tau)|^2 \, d\tau\right] \\ = & \sum_{m=-\infty}^{\infty} \left|\hat{\varphi}\left(\frac{2\pi m}{N}\right)\right|^2 \sum_{k=-\infty}^{\infty} |X_m(k)|^2 \end{aligned}$$

Here, we again made use of Parseval's theorem. Substituting for $X_m(k)$ and making a change of variable, the above summation can be rewritten as

$$\frac{1}{h}\int_0^h \frac{1}{T} d\tau \int_0^T |\mathcal{Q}_N s_\tau(t)|^2 dt \quad = \quad \sum_k |S(k)|^2 \, \hat{a}_\varphi\left(\frac{2\pi k}{N}\right) \left|\hat{\hat{\varphi}}\left(\frac{2\pi k}{N}\right)\right|^2$$

3. Making use of (6.29) and the Parseval's relation we rewrite the third integral $\frac{1}{T}\int_0^T s_\tau(t)^* \mathcal{Q}_N s_\tau(t) dt$ as

$$\sum_m \underbrace{S_\tau(m)^* \, \hat{\varphi}\left(\frac{2\pi m}{N}\right)^*}_{\left[S^*(m)\,\hat{\varphi}\left(\frac{2\pi m}{N}\right)^*\right] e^{j\frac{2\pi m \tau}{T}}} \underbrace{\sum_k S_\tau(m+kN) \, \hat{\hat{\varphi}}\left(\frac{2\pi\,(m+kN)}{N}\right)^*}_{e^{-j\frac{2\pi m \tau}{T}} x_m(\tau)}$$

Rearranging the terms, we get

$$\frac{1}{T}\int_0^T s_\tau(t)^* \mathcal{Q}_N s_\tau(t) dt = \sum_m \left[S(m)^* \, \hat{\varphi}\left(\frac{2\pi m}{N}\right)^*\right] x_m(\tau)$$

As before, $x_m(\tau)$ are a sequence of h periodic functions. Now averaging over τ as before, the term $\frac{1}{h}\int_0^h \frac{d\tau}{T}\int_0^T s_\tau(t)^* \mathcal{Q}_N s_\tau(t) dt$ becomes

$$\sum_m \left[S(m)^* \hat{\varphi}\left(\frac{2\pi m}{N}\right)^*\right] \cdot \underbrace{\frac{1}{h}\int_0^h x_m(\tau) d\tau}_{X_m(0) = S(m)\hat{\hat{\varphi}}\left(\frac{2\pi m}{N}\right)^*}$$

Substituting for the expression of $X_m(0)$ the expression above reduces to

$$\sum_m |S(m)|^2 \, \hat{\hat{\varphi}}\left(\frac{2\pi m}{N}\right)^* \, \hat{\varphi}\left(\frac{2\pi m}{N}\right),$$

which is equivalent to

$$\sum_m |S(m)|^2 \, \Re\left(\hat{\tilde{\varphi}}\left(\frac{2\pi m}{N}\right)^* \hat{\varphi}\left(\frac{2\pi m}{N}\right)\right) \tag{6.30}$$

Combining the three integrals, we get

$$\eta_s(N) = \sqrt{\sum_{k=-\infty}^{\infty} |S(k)|^2 E\left(\frac{2\pi k}{N}\right)}, \tag{6.31}$$

where

$$\begin{aligned} E(\omega) &= 1 + a_\varphi(\omega)\left|\hat{\tilde{\varphi}}(\omega)\right|^2 \\ &\quad -2\ \Re\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)\right) \\ &= |1-\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)|^2 \\ &\quad + |\hat{\tilde{\varphi}}(\omega)|^2 \sum_{k\neq 0} |\hat{\varphi}(\omega+2n\pi)|^2 \end{aligned}$$

Appendix 6-C: Asymptotic performance

In this proof, we assume that the kernel is L times continuously differentiable. Initially, we derive the conditions for which $\lim_{N\to\infty}(\eta_s(N))^2 = 0$. As $E(\omega)$ is bounded and $s(t) \in L_2([0,T])$, we use Lebesgue's dominated convergence theorem to interchange the limit and the summation in (6.16) to obtain

$$\begin{aligned} \lim_{N\to\infty}(\eta_s(N))^2 &= \sum_{k\in\mathbb{Z}} |S(k)|^2 \lim_{N\to\infty} E\left(\frac{2\pi k}{N}\right) \\ &= \sum_{k\in\mathbb{Z}} |S(k)|^2 E(0) = 0. \end{aligned}$$

Here, we used the continuity of the kernel. The above expression is true for any $s(t) \in L_2([0,T])$ if $E(0) = 0$. We have

$$E(0) = \frac{1}{a_\varphi(0)} \sum_{l\neq 0} |\hat{\varphi}(2l\pi)|^2 + a_\varphi(0)\left|\hat{\tilde{\varphi}}(0) - \hat{\varphi}_d(0)\right|^2 = 0$$

As the expression is a sum of positive quantities, it is equal to zero only if each of them is zero independently. In particular, we need $\hat{\varphi}(2l\pi) = 0,\ l \in \mathbb{Z} \setminus \{0\}$ and $\hat{\tilde{\varphi}}(0) = \hat{\varphi}_d(0)$. We also need $a_\varphi(0) \neq 0$ which is true iff $\hat{\varphi}(0) \neq 0$. These are precisely the Strang-Fix conditions of order 1.

Now, we look at the conditions for $\lim_{N\to\infty} (N\ \eta_s(N))^2 = 0$. This will imply that $\eta_s(N)$ decays faster than $\mathcal{O}\left(\frac{1}{N}\right)$ as $N \to \infty$. To derive the conditions, we rewrite the expression for $N\ \eta_s(N)$ as

$$(N\ \eta_s(N))^2 = \sum_{k\in\mathbb{Z}} |S(k)\,(2k\pi)|^2\, \frac{N^2}{(2k\pi)^2} E\left(\frac{2\pi k}{N}\right)$$

Now computing the limits by interchanging the sum and limit as $\frac{E(\omega)}{\omega^2}$ is bounded, we get

$$\lim_{N\to\infty} (N\ \eta_s(N))^2 = \underbrace{||S(k)\,(2k\pi)\,||^2_{\ell_2}}_{=\,T^2||s^{(1)}||^2_{L_2[0,T)}} \lim_{\omega\to 0}\left(\frac{E(\omega)}{\omega^2}\right)$$

Here, we made use of the fact that $E(\omega)$ is an even function of ω (its Taylor series has only even powers of ω).

$$\begin{aligned}
\lim_{\omega\to 0}\left(\frac{E(\omega)}{\omega^2}\right) &= \frac{1}{a_\varphi(0)} \sum_{l\neq 0} \left|\lim_{\omega\to 0} \frac{\varphi(\omega+2l\pi)}{\omega}\right|^2 + \\
&\qquad a_\varphi(0) \left|\lim_{\omega\to 0} \frac{\hat{\tilde{\varphi}}(\omega)}{\omega} - \lim_{\omega\to 0} \frac{\hat{\varphi}_d(\omega)}{\omega}\right|^2 \\
&= \sum_{l\neq 0} \left|\frac{\varphi^1(2l\pi)}{1!}\right|^2 + \left|\frac{\hat{\tilde{\varphi}}^{(1)}(0) - \hat{\varphi}_d^{(1)}(0)}{1!}\right|^2
\end{aligned} \tag{6.32}$$

With the same argument as before, in addition to Strang-Fix conditions of order 1, we need $\varphi^{(1)}(2l\pi) = 0,\ l \in \mathbb{Z} \setminus \{0\}$ and $\hat{\tilde{\varphi}}^{(1)}(0) = \hat{\varphi}_d^{(1)}(0)$. Continuing in the same fashion, we can see that $\eta_s(N)$ will decay as $\mathcal{O}\left(\frac{1}{N^L}\right)$ iff φ is an Lth order generating function and $\hat{\tilde{\varphi}}^{m)}(0) = \hat{\varphi}_d^{(m)}(0)$ for $m = 0 \ldots L-1$.

The function $\hat{\varphi}_d(\omega) = \frac{\hat{\varphi}(\omega)}{\sum_k |\hat{\varphi}(\omega+2k\pi)|^2}$ behaves as $\hat{\varphi}_d(\omega) = \frac{1}{\hat{\varphi}^*(\omega)} + \mathcal{O}(\omega)^L$ as $\omega \to 0$. Since $\hat{\tilde{\varphi}}$ is bi-orthogonal to $\hat{\varphi}$, it behaves as $\hat{\tilde{\varphi}}(\omega) = \frac{1}{\hat{\varphi}^*(\omega)} + \mathcal{O}(\omega)^L$ as $\omega \to 0$ (This follows from the bi-orthogonality relation $\sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega+2k\pi)\hat{\tilde{\varphi}}(\omega +$

$2k\pi) = 1)$. Hence, $\tilde{\varphi}$ being bi-orthogonal to φ ensures that $\hat{\tilde{\varphi}}^{m)}(0) = \hat{\varphi}_d^{(m)}(0)$ for $m = 0 \ldots L-1$. Thus the bi-orthogonality and the Strang-Fix conditions of order L are sufficient for the error $\eta_s(N)$ to decay as $\mathcal{O}\left(\frac{1}{N^L}\right)$.

L is the first positive integer for which

$$\begin{aligned} \lim_{N\to\infty} \left(N^L\ \eta_s(N)\right)^2 &= \underbrace{||S(k)\,(2k\pi)\,||_{\ell_2}^2}_{=\,T^{2L}||s^{(L)}||^2_{L_2[0,T)}}\ \underbrace{\lim_{\omega\to 0}\left(\frac{E(\omega)}{\omega^{2L}}\right)}_{=\,(C_{\varphi,\tilde{\varphi}})^2} \\ &\neq \quad 0 \end{aligned} \tag{6.33}$$

Proceeding as in (6.32), the expression of $C_{\varphi,\tilde{\varphi}}$ is

$$C_{\varphi,\tilde{\varphi}} = \sqrt{\sum_{k\neq 0}\left|\frac{\hat{\varphi}^{(L)}(2k\pi)}{L!}\right|^2 + \left|\frac{m^L_{\hat{\tilde{\varphi}}} - m^L_{\hat{\varphi}_d}}{L!}\right|^2} \tag{6.34}$$

In the above equation, we substituted for $\hat{\tilde{\varphi}}^{(L)}(0)$ and $\hat{\varphi}_d^{(L)}(0)$ with $(-j)^L m^L_{\tilde{\varphi}}$ and $(-j)^L m^L_{\varphi_d}$ respectively, where $m^L_u = \int x^L u(x)dx$.

Chapter 7

An exact algorithm for computing the area moments of spline curves

The last step of the shape estimation algorithm described in Fig. 1.4 involves the evaluation of shape parameters from the shape model. In this chapter[1], we introduce an efficient algorithm for the exact computation of area moments.

7.1 Introduction

Moments are standard descriptors of the shape of an object [108],[109],[110]; they easily yield features that are invariant to translation and rotation [12] or more generally to affine transformations, which makes them useful tools for pattern recognition. In the standard formulation, they are computed as surface integrals which requires raster scanning through the image. However, there are many instances where the boundaries of objects are described by parametric curves. This is the case, for example, when the objects are detected using parametric snakes which are represented using B-spline [17, 36, 73, 68] or wavelet basis functions [34, 111]. Another simple case is when the region is described as a polygon [112].

[1]Based on the article "M.Jacob, T.Blu, M.Unser, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 633-642, June 2001.*

In this chapter, we address the problem of computing the area moments of objects described by such parametric curves when the basis functions are scaling functions. The popular wavelet curve descriptors also fall into this class. The originality of our approach is that the computation is exact, and also more direct than the conventional pixel-based method which requires an explicit labelling of the inner region of the curve prior to computation. Moreover, the pixel-based schemes suffer a low accuracy due to the loss of subpixel details in the rasterizing process. Also, the error in the area-based computation of moments is dependent on the orientation of the shape.

Since a polygon can be represented in terms of linear splines, the computation of moments by approximating the shape as a polygon [112, 113, 114] is a particular case of our approach. While the polygon method can be made as accurate as desired by increasing the number of segments, the convergence is slow because of the low approximation order of linear splines. Moreover, it is not suitable for computing the curvature, which is an interesting shape feature as it is invariant to rotation and translations and can be easily normalized to scale changes. This motivates us to investigate higher order schemes where the curve is represented by smoother basis functions such as B-splines and other scaling functions that appear in wavelet theory [22, 23]. These type of basis functions also occur naturally when one seeks multiresolution representation of curves which are well suited for pattern recognition and shape simplification [115, 111].

The chapter is organized as follows. In section 7.2, we show how Green's Theorem can be used for the computation of the area moments of a parametric curve. In section 7.3, we consider the computation of the moments of such a curve represented in spline or wavelet bases. Here, we also discuss the properties of the multidimensional kernel used in the computation of moments. In section 7.4, we give the implementation details of the moment computation. In the following section, we deal with the precomputation of the kernel. In section 7.6, we present an alternate implementation that works for any order moments, but it is rigorously exact only when the scaling function is $\mathrm{sinc}(x)$. This is especially interesting because it makes our method applicable to the Fourier representation of curves as well. In the last section, we compare the new method with the existing schemes such as approximation using polygons and rasterizing.

7.2 Preliminaries

7.2.1 Computation of Moments using Green's Theorem

Green's Theorem relates the volume integral of the divergence of a vector field in a closed region to the integral of the field over the surface enclosing it. In this section, we show how it can be used to compute the moments of an area enclosed by a curve.

Consider a closed region $\mathcal{V}$, bounded by a surface $\mathcal{S}$. Green's Theorem states that, for any vector field F,

$$\int_{\mathcal{V}} (\nabla.\mathrm{F})\, dV = \int_{\mathcal{S}} \mathrm{F.dS}, \tag{7.1}$$

where dS is the unit vector pointing out of the surface $\mathcal{S}$. Assuming the volume to have a constant cross-section bounded by the curve $\mathcal{C}$, and that the variation of the field along the z-direction is zero, we can restrict the theorem to two dimensions as,

$$\int_{\mathcal{S}} (\frac{\partial \mathrm{F}_x}{\partial x} + \frac{\partial \mathrm{F}_y}{\partial y}) dxdy = \oint_{\mathcal{C}} (\mathrm{F}_y dx - \mathrm{F}_x dy) \tag{7.2}$$

The first integral is evaluated over the area $\mathcal{S}$ enclosed by the curve and the second one along the curve $\mathcal{C}$ in the clockwise direction. The computation of the moments involves the evaluation of the integral $\int_{\mathcal{S}} x^m.y^n.dxdy$ on the surface bounded by the curve. This, by (7.2), is equivalent to

$$I_{m,n} = \oint_{\mathcal{C}} \frac{x^m y^{n+1}}{n+1} dx, \tag{7.3}$$

with $\mathrm{F} = \mathrm{e_y}(\frac{x^m y^{n+1}}{n+1})$; $\mathrm{e_y}$ denotes the unit vector along the y direction. Note that the choice of F is not unique. We choose the vector field F that makes the computation simple. Another possible choice that has the same computational complexity is $\mathrm{F} = -\mathrm{e_x}(\frac{x^{m+1} y^n}{m+1})$.

7.2.2 Parametric Representation of a curve

We represent a closed curve in the x — y plane as discussed in Section 2.4. We repeat it here since the notations used in this chapter are slightly different. A curve in the x — y plane can be represented in terms of an arbitrary parameter

t as $\mathrm{r}(t) = (x(t), y(t))$. If the curve is closed, as discussed in the paper, the functions $x(t)$ and $y(t)$ are periodic.

When the curve $\mathcal{C}$ is represented as above, $\mathrm{r}(t)$ can be approximated efficiently as linear combinations of some basis functions, which makes the representation compact and easy to handle. In this paper, we mainly focus on the representation of the function vector $\mathrm{r}(t)$ in a scaling function basis as

$$\mathrm{r}(t) = \sum_{k=-\infty}^{\infty} \mathrm{b}_k \, \varphi(t-k) \tag{7.4}$$

Here b_k denotes the sequence of vector coefficients given by $\mathrm{b}_k = (c_k, d_k)$. If the period, M, is an integer, we have $\mathrm{b}_k = \mathrm{b}_{k+M}$. This reduces the infinite summations to

$$\mathrm{r}(t) = \sum_{k=0}^{M-1} \mathrm{b}_k \, \varphi_p(t-k) \tag{7.5}$$

where

$$\varphi_p(t) = \sum_{k=-\infty}^{\infty} \varphi(t - k\,M) \tag{7.6}$$

In the context of wavelets, φ is called the *scaling function*; it satisfies the two-scale difference equation

$$\varphi(t) = \sum_{k} h(k)\varphi(2t-k), \tag{7.7}$$

where $h(k)$ is the mask of the corresponding refinement filter [22]. The scaling function representation enables us to have local control of the contour, which is desirable in many applications. It also permits a multiresolution representation of the curve [34, 35]. Moreover, the scaling function representation is affine-invariant; an affine transformation of the curve is achieved simply by transforming the coefficient vector b_k, $k = 0, 1, \ldots, M-1$. This is because of the linearity of the representation and the partition of unity condition:

$$\sum_{k=-\infty}^{\infty} \varphi(t-k) = 1, \tag{7.8}$$

which is satisfied by all valid scaling functions in wavelet theory. Among the scaling functions, a case of special interest is $\varphi = \beta^n$, where β^n is the causal

B-spline of degree n [27] defined by its Fourier transform

$$\hat{\beta}^s(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega}\right)^{s+1}. \tag{7.9}$$

This yields spline curves which are frequently used in computer graphics [15] and computer vision [36, 37, 38].

The description of $\mathcal{C}$ in the scaling function basis is equivalent to a periodized wavelet representation[34]. This implies that, if we have a wavelet description of the curve, the scaling function coefficients at any scale can be obtained from the wavelet coefficients using the fast reconstruction equation described in [116]. Hence, the theory is sufficiently general to include the wavelet curve descriptors as well.

The representation of the curves in a sinc basis also falls in this class, as the sinc function is a valid scaling function. The description of the curve in the sinc basis as (7.5) is not efficient, as sinc has an infinite mask unlike most of the widely used scaling functions. It is well known (c.f. [32]) that the sinc interpolation of a periodic signal can be formulated into a numerically stable and efficient expression as

$$\mathrm{r}(t) = \sum_{k=-L}^{L} \mathrm{b}_k \, \exp\left(\frac{j2\pi kt}{M}\right) \tag{7.10}$$

where $2L + 1 = M$, assuming M to be odd. A similar expression is obtained for even M as well. Here b_k is the discrete Fourier transform of the vector sequence $\mathrm{r}(k)$. Note that (7.10) provides the Fourier series description of the curve, which is frequently used for the representation of closed curves [10, 69].

7.2.3 Differentiation of scaling functions

We will use the property that the kth derivative of a scaling function φ can be expressed as [117]

$$\varphi^{(k)}(x) = \Delta^k \, \varphi^{\{k\}}(x), \tag{7.11}$$

where $\varphi^{\{k\}}(x)$ denotes the scaling function whose mask is given by $H^{\{k\}}(z) = \left(\frac{2}{1+z^{-1}}\right)^k H(z)$; $H(z)$ is the mask of φ. Δ denotes the backward difference operator, defined as $\Delta \, \eta(x) = \eta(x) - \eta(x-1)$.

The relation (7.11) follows from the fact that any m^{th} order scaling function can be written as

$$\hat{\varphi}(\omega) = \underbrace{\left(\frac{1-e^{j\omega}}{j\omega}\right)^m}_{\hat{\beta}^{m-1}(\omega)} \hat{\gamma}(\omega),$$

where γ is a refinable distribution which does not satisfy the partition of unity. The mask of φ is $H(z) = \left(\frac{1+z^{-1}}{2}\right)^m H_\gamma(z)$. Note that $\left(\frac{1+z^{-1}}{2}\right)^m$ is the mask of β^{m-1}, and H_γ the mask of γ. Differentiating φ with respect to x, k number of times ($k \leq m$) yields

$$\varphi^{(k)}(x) \xrightarrow{\mathcal{F}} (j\omega)^k \hat{\varphi}(\omega) = \left(1-e^{j\omega}\right)^k \underbrace{\left(\frac{1-e^{j\omega}}{j\omega}\right)^{m-k} \hat{\gamma}(\omega)}_{\hat{\varphi}^{\{k\}}(\omega)} \xrightarrow{\mathcal{F}^{-1}} \Delta^k \varphi^{\{k\}}(x) \tag{7.12}$$

Thus the mask of $\varphi^{\{k\}}(x)$ is $H^{\{k\}}(z) = \left(\frac{1+z^{-1}}{2}\right)^{m-k} H_\gamma(z) = \left(\frac{2}{1+z^{-1}}\right)^k H(z)$.

7.3 Moment computation

To facilitate the understanding of our method, we first give a detailed derivation of the formula for the area of the region bounded by the curve. We then extend our formulation to the general case.

7.3.1 Computation of the Area

For the parametric representation of the curve, the area of the region is given by

$$I_{0,0} = \int_0^M y(t) \frac{dx(t)}{dt} dt \tag{7.13}$$

When the curve is described in a scaling function basis as in (7.5), we have

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \int_0^M \varphi_p(t-i)\varphi_p'(t-j)dt, \tag{7.14}$$

where

$$\varphi_p'(t) = \frac{d\,\varphi_p(t)}{dt}$$

Substituting for $\varphi_p'(t)$ from (7.44), we get

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \int_{-\infty}^{\infty} \varphi_p(t-i)\varphi'(t-j)dt, \tag{7.15}$$

which is equivalent to

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \underbrace{\int_{-\infty}^{\infty} \varphi_p(t-i+j)\varphi'(t)dt}_{g_0^p(i-j)}. \tag{7.16}$$

Again, substituting for φ_p from (7.44), we get the kernel $g_0^p(l)$ as the M periodized version of

$$g_0(l) = \int_{-\infty}^{\infty} \varphi'(t)\varphi(t-l)dt \tag{7.17}$$

as $g_0^p(l) = \sum_{k=-\infty}^{\infty} g_0(l + k\,M)$. With the simplification (7.11), the above equation becomes

$$g_0(l) = \Delta f_0(l), \tag{7.18}$$

and

$$f_0(x) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\varphi(t-x)dt. \tag{7.19}$$

Note that, if $\varphi(t) = \varphi(\tau - t)$, then $f_0(x)$ can be written as the convolution $\left(\varphi^{\{1\}} * \varphi\right)(\tau + x)$. We prefer to represent the kernel g^p in terms of f due to its nice properties, discussed later.

For the example given in Fig.2.5, we have

$$\begin{aligned} g_0(l) &= \Delta\left(\beta^0 * \beta^1\right)(l+2) = \Delta(\beta^2)(l+2) \\ g_0 &: (0.5, 0, -0.5); \quad l \in \{-1, 0, 1\}, \end{aligned}$$

where β^n is the causal B-spline function of degree n. Now for the polygon, $c(k) : (1, 1, 6, 8, 7, 4)$ and $d(k) : (1, 6, 8, 5, 1, 0)$. Hence, by (7.16), we have

$$\begin{aligned} I_{0,0} &= \frac{1}{2}\langle (6, 8, 5, 1, 0, 1), (5, 7, 1, -4, -6, -3)\rangle \\ &= 42 \ \text{ units} \end{aligned}$$

Here $\langle x_1, x_2\rangle$ stands for the ℓ_2 inner product given by $\sum_k x_1(k)x_2(k)$.

7.3.2 General Formula

Having shown how to compute the area, we proceed on to the general case. The formula for the computation of the general moments are given by the following theorem

Theorem 3 *Let $\mathcal{C}$ be a closed curve in the x-y plane represented in the parametric form in a periodized scaling function basis as (7.5). Then the $(m,n)^{\text{th}}$ order area moment of the region $\mathcal{S}$, bounded by the curve $\mathcal{C}$, given by*

$$I_{m,n} = \int_{\mathcal{S}} x^m y^n \, dxdy \quad \text{for } m,n \geq 0 \tag{7.20}$$

can be computed as

$$I_{m,n} = \frac{1}{n+1} \sum_{k \in \mathcal{R}} \sum_{\substack{\mathrm{i} \in \mathcal{R}^{m+1} \\ \mathrm{j} \in \mathcal{R}^{n}}} c_k \, \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} \, g^p_{m+n}(\mathrm{i}-k, \mathrm{j}-k), \tag{7.21}$$

where $\mathcal{R}$ is the integer range $[0 \ldots M-1]$. The kernel g^p_{m+n} in (7.21) is

$$g^p_{m+n}(\mathrm{k}) = \int_{-\infty}^{\infty} \varphi'(t) \, \varphi_p(t-k_1) \ldots \varphi_p(t-k_{m+n+1}) \, dt$$

Here $\mathrm{c}^{[m]}$ stands for the m-times tensor product[2] $c \otimes c \ldots \otimes c$ and $\mathrm{i}-k$ denotes the sequence $(i_1-k, i_2-k, \ldots i_{m+1}-k)$.

Proof: For a parametric curve, the evaluation of the $(m,n)^{\text{th}}$ order moment given by (7.20) can be reduced to

$$I_{m,n} = \frac{1}{n+1} \int_0^M x^m(t) \, y^{n+1}(t) \, \frac{dx(t)}{dt} \, dt \tag{7.22}$$

by (7.3). When the curve is described in a scaling function basis, we have

$$\begin{aligned} I_{m,n} \quad = \quad & \frac{1}{n+1} \sum_{k \in \mathcal{R}} \sum_{\substack{\mathrm{i} \in \mathcal{R}^{m+1} \\ \mathrm{j} \in \mathcal{R}^{n}}} c_k \, \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} \int_0^M \varphi_p(t-\mathrm{i}_1) \ldots \varphi_p(t-\mathrm{i}_m) \varphi_p(t-\mathrm{j}_1) \ldots \\ & \varphi_p(t-\mathrm{j}_{n+1}) \varphi'_p(t-k) dt. \end{aligned}$$

[2] $c^{[0]}$ is defined as the neutral element; $c^{[0]} \otimes c^{[m]} = c^{[m]}$.

Substituting for $\varphi_p'(t)$ from (7.44), we get

$$I_{m,n} = \frac{1}{n+1}\sum_{k\in\mathcal{R}}\sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1}\\ \mathrm{j}\in\mathcal{R}^{n}}} c_k\, \mathrm{c_i}^{[m]}\mathrm{d_j}^{[n+1]} \int_{-\infty}^{\infty} \varphi_p(t-\mathrm{i}_1)\ldots\varphi_p(t-\mathrm{i}_m)\varphi_p(t-\mathrm{j}_1)\ldots \varphi_p(t-\mathrm{j}_{n+1})\varphi'(t-k)dt.$$

The integral in the above equation is equivalent to

$$\underbrace{\int_{-\infty}^{\infty} \varphi'(t)\varphi_p(t+k-\mathrm{i}_1)\ldots\varphi_p(t+k-\mathrm{i}_m)\varphi_p(t+k-\mathrm{j}_1)\ldots\varphi_p(t+k-\mathrm{j}_{n+1})dt}_{g^p_{m+n}(\mathrm{i}-k,\mathrm{j}-k)}.$$

Hence the $(m,n)^{\text{th}}$ order moment is

$$I_{m,n} = \frac{1}{n+1}\sum_{k\in\mathcal{R}}\sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1}\\ \mathrm{j}\in\mathcal{R}^{n}}} c_k\, \mathrm{c_i}^{[m]}\mathrm{d_j}^{[n+1]}\, g^p_{m+n}(\mathrm{i}-k,\mathrm{j}-k).$$

■

As in the case of the area, the kernel g_p is obtained by the M-periodization of

$$g_{m+n}(\mathrm{k}) = \int_{-\infty}^{\infty} \varphi'(t)\varphi(t-k_1)..\varphi(t-k_{m+n+1}).dt,$$

where $\mathrm{k}\in\mathbb{Z}^{m+n+1}$. Expressing φ' in terms of $\varphi^{\{1\}}$, we get

$$g_{m+n}(\mathrm{k}) = f_{m+n}(\mathrm{k}) - f_{m+n}(\mathrm{k}-1), \tag{7.19}$$

where

$$f_{m+n}(\mathrm{x}) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\varphi(t-x_1)..\varphi(t-x_{m+n+1})dt, \tag{7.20}$$

where $\mathrm{x} = (x_1, x_2\ldots, x_{m+n+1}) \in \mathbb{R}^{m+n+1}$. The kernel f has many interesting properties, which are discussed next.

7.3.3 Properties of the kernel - f

1. **Finite support:** As the kernel is an integral of products of the translates of finitely supported functions, it has a finite support as well. If the scaling function is continuous and has a support $[0,N]$, then the kernel will be supported on the integer points in the interval

$$I = [-N+1, N-2]\times\ldots[-N+1,N-2]\times[-N+1,N-2] \tag{7.21}$$

2. **Symmetry:** The fact that the kernel is obtained from the integration of similar translated scaling functions introduces a lot of symmetry. As (7.20) is symmetric with respect to the parameters $k_1, k_2, ..$, interchanging them will not affect the value of the kernel. This implies

$$f(\mathrm{k}) = f(\sigma_i(\mathrm{k})) \tag{7.22}$$

where σ_i indicates all possible $(m+n+1)!$ permutation operators. In addition, if the scaling functions are symmetric as in the case of splines, we have

$$f(\mathrm{k}) = f(-\mathrm{k}) \tag{7.23}$$

Both these properties together imply $2((m+n+1)!)$ relations, which are used to accelerate the computation of the kernel as well as the moments.

3. **Two-scale relation:** We now show that the kernel satisfies a two-scale relation, which is the key to our computational approach. This property follows from the fact that the scaling functions $\varphi(t)$ and $\varphi^{\{1\}}(t)$, from which the kernel is derived, satisfy two-scale relations. If we consider (7.20) and rewrite the φ and $\varphi^{\{1\}}$ in terms of the corresponding two-scale relations (cf. (7.7)), we get

$$f_{m+n}(\mathrm{k}) = \sum_{\mathrm{l} \in \mathbb{Z}^{\mathrm{m+1}}} H_{m+n}(\mathrm{l}).f_{m+n}(2\mathrm{k} - \mathrm{l}) \tag{7.24}$$

where $\mathrm{k} \in \mathbb{Z}^{m+1}$.The mask H in the above equation is

$$H_m(l_1, l_2, .., l_m) = \frac{1}{2} \sum_k h_1(k).h(k - l_1)..h(k - l_m), \tag{7.25}$$

The z-transform of the mask is given by

$$H_m(z_1, z_2, .., z_m) = \frac{1}{2} H_1(\prod_{k=1}^{m} z_k) \prod_{k=1}^{m} H(z_k^{-1}) \tag{7.26}$$

It is this property that enables us to compute the kernels exactly, by solving a linear system of equations.This technique, which is discussed later, is analogous to the computation of the integer (or dyadic rational) samples of a scaling function from the transition operator [22].

Note that a scaling relation similar to (7.24) was also considered by mathematicians in the context of the wavelet-Galerkin method for the computation of integrals involving products of scaling functions and their

derivatives [118, 119]. The work of Dahmen and Miccheli is essentially theoritical; Restrepo and Leaf concentrated on numerical issues and proposed a solution which is equivalent to the computation of our kernel g_m instead of f_m. This slightly complicates the approach and also increases the dimensionality of the problem; this issue is discussed further in Section 7.5.1.

The above mentioned properties imply that the kernel can be computed exactly for any finitely supported scaling function, as discussed in section 5. In the next subsection, we will give some examples for the kernels when the scaling functions are B-splines.

7.3.4 Examples with Splines

Splines possess nice approximation properties. The B-splines have the maximum approximation order among the class of functions that satisfy a two-scale relation with a given support. Hence they give better local control of the contour. Moreover, they are symmetric, which facilitates the computation of the kernel and moments as discussed before. So it is worthwhile to analyze the properties of the kernels for a spline representation of the curve. For the results used in this section, refer to [27].

We consider causal B-splines, as they satisfy a two-scale relation for all orders. The refinement filter for a B-spline of degree n is the binomial filter

$$h(k) = \frac{1}{2^n}\binom{n+1}{k} \tag{7.27}$$

If we choose β^s, a B-spline of degree s, as φ, then $\varphi^{\{1\}} = \beta^{s-1}$; that is a spline of degree $s-1$. Hence the kernel f as given by (7.20) is a **box spline** [120] sampled at the integers. In particular,

$$f_0(k) = \beta^{2s}(k+s+1). \tag{7.28}$$

The spline functions have a closed-form representation in the Fourier domain, which the kernels also inherit. By taking the continuous Fourier transform of (7.20), when the scaling function is a B-spline, we get

$$\hat{f}_n^s(\omega);\ \ \omega \in \mathbb{Z}^n = \hat{\beta}^{s-1}(|\omega|)\prod_{i=1}^{n}\hat{\beta}^s(\omega_i), \tag{7.29}$$

where $|\omega|$ stands for $\sum_{j=1}^{n} \omega_j$. By using Poisson's formula

$$\sum_k \hat{f}_n^s(\omega + 2k\pi) = \frac{1}{2\pi} \sum_k f_n^s(k) e^{-2j\pi\omega k}, \tag{7.30}$$

we get the discrete Fourier transform of the kernel as the 2π-periodized version of (7.29).

We give some examples of kernels for the computation of the first three moments when we have a linear spline representation. For linear splines, the kernel $f_{m-1}(k_1, k_2, \ldots, k_m)$ is supported in the interval $[-1, 0] \times [-1, 0] \ldots [-1, 0]$. The kernels are

$$f_0(k_1); k_1 \in \{-1, 0\} \quad : \quad \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{7.31}$$

$$f_1(k_1, k_2); k_1, k_2 \in \{-1, 0\} \quad : \quad \frac{1}{6} \cdot \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \tag{7.32}$$

$$\begin{cases} f_2(-1, k_2, k_3); k_2, k_3 \in \{-1, 0\} & : \ \frac{1}{12} \cdot \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix} \\ f_2(0, k_2, k_3); k_2, k_3 \in \{-1, 0\} & : \ \frac{1}{12} \cdot \begin{bmatrix} 1 & 3 \\ 1 & 1 \end{bmatrix} \end{cases} \tag{7.33}$$

It is interesting to see that the computation of the moments using the linear spline kernel is the same as when the polygon is triangulated in a specified way and the moments of individual triangles added up as in [112].

We also give the kernel f_0 for the cubic spline representation.

$$f_0(k_1); k_1 = -3, \ldots 2 \quad : \quad \frac{1}{720} \cdot \begin{bmatrix} 1, & 57, & 302, & 302, & 57, & 1 \end{bmatrix} \tag{7.34}$$

The higher order kernels are omitted due to space constraints. They can be downloaded from http://bigwww.epfl.ch/jacob.

7.4 Implementation.

In this section, we analyze equation (7.21) and simplify it for faster computation. We start with the simplest case: the area of the region.

The area bounded by the curve (cf. (7.15)) is computed as

$$I_{0,0} = \sum_{k=0}^{M-1} c_k^p \sum_{l=-N+1}^{N-2} d_{k+l}^p g_0(l), \tag{7.35}$$

where g_0 is given by (7.17). The sequences c_k^p and d_k^p are M-periodized versions of the coefficients c_k and d_k with respect to the period M. This is simply because convolving a non-periodized sequence with a periodized kernel is equivalent to convolving a periodized sequence with a non-periodized kernel. We have also reduced the range of summation of the inner sum to $-N+1$ to $N-2$, which is typically much less than the range 0 to $M-1$. Similarly, for the higher order moments all the summations, except the outer one, are in the range $-N+1$ to $N-2$.

From (7.35), we see that the computation of the area involves just a filtering operation by $g(-l) = g^T(l)$, followed by an inner product. This can be written as,

$$I_{0,0} = \langle c^p, g_0^T * d^p \rangle, \tag{7.36}$$

where $\langle ., . \rangle$ stands for the inner product $\langle c, d \rangle = \sum_{k=0}^{M-1} c(k)d(k)$. With a similar notation, the computation of the other moments are given as

$$I_{m,n} = \frac{\langle c^p, g_{m+n}^T * (c^{p[m]} \otimes d^{p[n+1]}) \rangle}{n+1} \tag{7.37}$$

$$= -\frac{\langle d^p, g_{m+n}^T * (c^{p[m+1]} \otimes d^{p[n]}) \rangle}{m+1} \tag{7.38}$$

As the $(m+n+1)$ – D sequence is separable, the filtering operation is much simpler than the usual $(m+n+1)$ – dimensional filtering.

The complexity in the computation of the moment $I_{m,n}$ is $M.(2N-2)^{(m+n+2)}$, without taking the symmetries into account. Thus, for basis functions with small support and reasonable m and n, the complexity is quite managable.

7.5 Computation of the Kernel

In this section, we propose two schemes for computing the kernel. An exact space domain scheme and an approximate one in the Fourier domain.

7.5.1 Exact Method

In this scheme, we compute the kernels in space domain making use of the properties of kernels discussed before. We start with the computation of f_0, and later extend it to the general case. Making use of the finite support property,

the two-scale relation (7.24) can be rewritten in the matrix form as,

$$\mathbf{A}_0.\mathrm{f}_0 = \mathrm{f}_0, \tag{7.39}$$

where $\mathbf{A_0}$ is the square matrix with coefficients $[\mathbf{A}_0]_{k,l} = H_0(2k-l)$ and f_0 is the vector whose elements are $f_0(n)$. As the support of f_0 is $[-N+1, N-2]$, the indices of $\mathbf{A_0}$ run from $-N+1$ to $N-2$.

It can be seen from the equation (7.39) that f_0 is an eigen-vector of the matrix $\mathbf{A_0}$, with eigen-value 1. Solving for f_0 is equivalent to solving for a vector which falls in the nullspace of $(\mathbf{A_0} - \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Since $\mathrm{f}_0 \neq 0$, $\mathbf{A_0}$ must have the eigen-value 1, which is in general single. This provides f_0 up to a constant which is further set by the normalization identity

$$\sum_k f_0(k) = 1, \tag{7.40}$$

which can be seen from (7.19). This is because the function $\varphi(x)$ has at least an approximation order of one [22], which implies $\sum_k \varphi(x+k) = 1$. One of the equations in $(A_0 - I).\mathrm{f}_0 = 0$ can be substituted for by the equation (7.40) to yield the system of equations given by

$$\mathbf{B}.\mathrm{f}_0 = \mathrm{y}; \tag{7.41}$$

$\mathbf{B}$ is the matrix obtained by substituting one of the rows of $(\mathbf{A_0} - \mathbf{I})$ with the row vector $[1, 1, \ldots, 1]$ and y is given by $[0, 0, 0 \ldots, 0, 0, 1]^T$ c.f [99]. Now $\mathbf{B}$ is a full rank matrix, and hence the eigen-vector f_0 can be solved by matrix inversion.

To represent the two-scale relations of the higher order kernels in the matrix form, we introduce a one-to-one function $\rho : [-N+1, N-2]^m \mapsto [0, (2N-2)^m - 1]$. Using this function, (7.24) can be rewritten as

$$f_m(\rho^{-1}(k)) = \sum_{l=0}^{(2N-2)^{m+1}-1} H_m\left(2\rho^{-1}(k) - \rho^{-1}(l)\right) f_m\left(\rho^{-1}(l)\right)$$

which is a linear system of equations. This can be written in the matrix form as

$$\mathbf{A}_m \mathrm{f}_m = \mathrm{f}_m, \tag{7.42}$$

where $[\mathbf{A}_m]_{i,j} = H_m\left(2\rho^{-1}(i) - \rho^{-1}(j)\right)$ and $\mathrm{f}_m(i) = f\left(\rho^{-1}(i)\right)$. This equation is of the same form as (7.39) and can be solved in the same way, with the normalization constraint $\sum_i \mathrm{f}_m(i) = 1$.

Let us now compare our computational solution with the method developed for computing g_m in the context of wavelet-Galerkin approach [118]. For a scaling function of support N, the kernel g_m is zero outside the interval

$$I' = [-N+1, N-1] \times \ldots [-N+1, N-1] \times [-N+1, N-1]. \tag{7.43}$$

as compared to f_m whose support is given by (7.21). Thus the direct computation of g_m involves a linear system with $(2N-1)^m$ variables as compared to $(2N-2)^m$ for f_m in our case. For 3 dimensional kernels involving cubic splines, we achieve a 40% reduction in the number of equations. As the computational complexity in inverting a linear system is proportional to the third power of the number of equations, this implies a performance improvement of around 5 times. The approach becomes even more rewarding for higher order kernels. Moreover, the normalization constraint (7.40) that we use to make the system full rank is much more straight forward than the corresponding relation for the derivative functions.

Note that this simplification is covered by Dahmen and Michelli's general theory for integrals of multidimensional scaling functions [119]. This is because the mask of any m^{th} order $1-D$ scaling function can be always factored as proposed in [119, Corollary 3.3]. In the case of wavelet-Galerikin integrals, the performance improvement can even more substantial depending on the number of derivatives.

7.5.2 Approximate Method for Splines

Because the spline kernel has a closed-form expression in the frequency domain, the kernel can be obtained by taking the inverse DFT of the above mentioned Fourier transform (7.30) sampled at an appropriate rate; we make use of the finite support property of the kernel. As sinc is a decaying function, the periodization of the Fourier transform may be approximated with an appropriately truncated sum to achieve any desired accuracy. This is because we can have an upper bound for the error that is a decreasing function of the summation range. Moreover, the symmetries of the kernel discussed before may be used for the efficient computation of the box spline kernels as in [121].

However, this technique, besides being approximate, can be used only for scaling functions that have a closed form expression in the frequency domain, i.e splines in practice. This scheme may be useful to precompute the spline kernels for very high order moments, where the exact scheme can be computationally expensive.

7.6 Computation of the Area Moments using Riemann sums

An alternate approach to compute the moments is to approximate the integral (7.3) by a Riemann sum:

$$I_{m,n} = \frac{1}{(n+1)P} \cdot \sum_{l=0}^{MP-1} [x_{\text{int}}(l/P)]^m . [y_{\text{int}}(l/P)]^{n+1} . [x'_{\text{int}}(l/P)], \tag{7.44}$$

where P is an appropriate oversampling factor. We show in this section that this quadrature formula is exact when the curves are described in a sinc basis. For other representations, it can be used for the approximate computation of higher order moments.

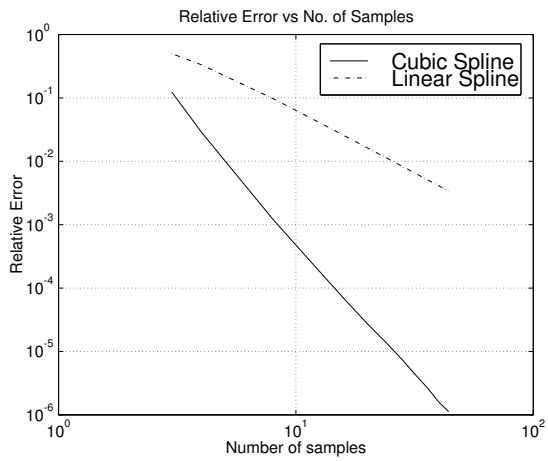


Figure 7.1: Comparison of moment estimators

7.6.1 Sinc Representation of the curve

A curve represented in a sinc basis also falls into the framework of Theorem 3 because $\text{sinc}(x)$ is a valid scaling function. However, computing the moments as described in Section 7.4 is expensive as the mask of the sinc function is not finitely supported. We remind the reader that the representation of a periodic

signal in the sinc basis is equivalent to the Fourier representation as seen in (7.10).

In this particular case, the moments can be computed exactly and more efficiently using (7.44), where the oversampling factor, P, is any integer greater than $\frac{m+n+2}{2}$.

Proposition 5 *The quadrature formula (7.44) is exact for the sinc representation provided that* $P \geq \lceil \frac{m+n+2}{2} \rceil$

The continuously defined functions $x_{\text{int}}(t)$ and $y_{\text{int}}(t)$ are obtained by interpolating the sample values of the curve at the integers, using the periodized sinc function. The computation is exact because we implicitly assume that the functions $x(t)$ and $y(t)$ are bandlimited functions, with bandwidth $B = 2\pi$.

Proof: The integral (7.3) can be considered as an $L_{2\ (0,M)}$ inner product of two functions, which are $\lceil \frac{m+n+2}{2} \rceil$ and $\lfloor \frac{m+n+2}{2} \rfloor$ fold[3] products of the corresponding band-limited functions. Hence they are bandlimited by $B' = B\lceil \frac{m+n+2}{2} \rceil$ and $B'' = B\lfloor \frac{m+n+2}{2} \rfloor$ respectively. So these functions are exactly represented in the basis $\{\text{sinc}(Px - k), \forall k \in \mathbb{Z}\}$, where $2\pi P \geq B'$. Because the sinc basis is orthogonal, the $L_{2\ (0,M)}$ inner product is equivalent to the $\ell_{2\ (0,MP-1)}$ inner product. Hence it is sufficient to compute the discrete summation instead of the integral. Finally, the sinc function is interpolating, so that the coefficients of the basis functions are the resampled curve values, and hence the result (7.44). ■

Using the equivalence of the sinc and the Fourier representations, we can compute the interpolated samples efficiently with a MP point inverse FFT of the Fourier coefficients c_k and d_k.

We will compare the sinc moment estimator with the scaling-function- based moment estimator in the next section. One disadvantage of the Fourier(sinc) representation of curves is the loss of local control property that we were having with the finitely supported scaling functions.

The complexity in the computation of the moments in this scheme is $MP(3\log(MP) + (m + n + 2))$. Here $3MP\log(MP)$ is the cost of the inverse FFT of the sequences c_k, d_k and $k.c_k$, and $(m + n + 2)MP$ corresponds to the multiplications.

[3] $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and the ceiling operators, operating on a fraction x to yield the lower and upper integers that bound x.

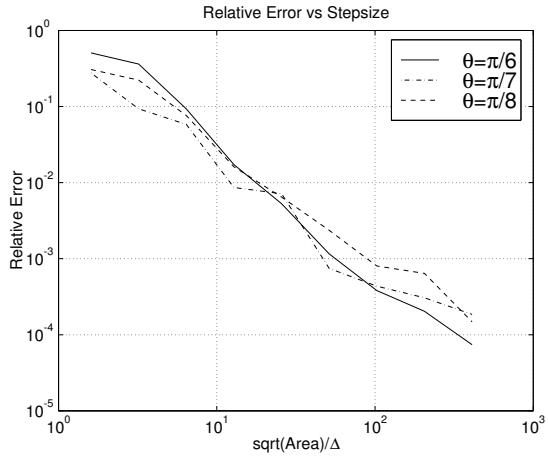


Figure 7.2: Variation of error vs $\frac{1}{\Delta}$ in a raster scan moment estimator

7.6.2 Spline Representation of the curve

The quadrature formula (7.44) is also applicable to the spline representation, provided that the functions $x_{\text{int}}(t)$ and $y_{\text{int}}(t)$ are obtained by interpolating the integer sample values, using the corresponding B-spline functions. This scheme is no longer exact, but it may be a viable alternative for computing the higher order moments. The necessary condition for the computation to be reliable is that the Fourier transform of the B-spline function is essentially bandlimited to $2\pi P$, where P is the oversampling factor. The error in the moments computed with the approximate method is thus proportional to the residual energy of the B-spline function in the corresponding outband. As the Fourier transform of the B-spline is a decaying function of the frequency, the error will be a decaying function of P as well. Thus, any desirable accuracy may be achieved by choosing P sufficiently large.

The complexity of the spline quadrature formula is

$$\mathcal{O}\left(M(m+n+2)(m+n+2+3N)P\right),$$

where $M(m+n+2)P$ is the total number of resampled points. The evaluation of the spline representation requires N multiplications to obtain one resampled point from the corresponding B-spline representation. Then the computation of

the discrete sum costs $m + n + 2$ multiplications per resampled point. Interestingly, the approximate scheme will give better results for higher order splines as these functions will become bandlimited as the order tends to infinity [33].

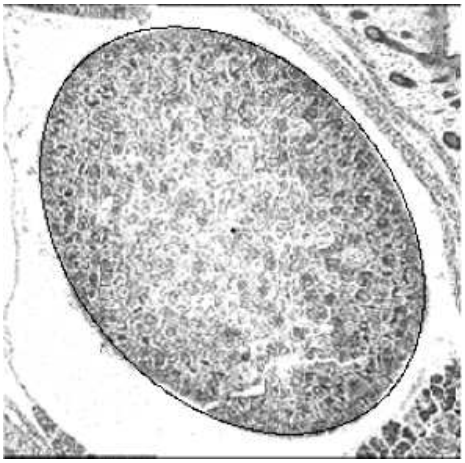


Figure 7.3: Estimated ellipse for a real image

7.7 Experiments and results.

In this section we compare the new technique with the existing ones: approximation using polygons and rasterizing. We first consider the exact scheme proposed in section 7.4. We try to estimate the parameters of a known ellipse and choose the relative error in the parameters as the criterion of comparison.

Our preferred choice is to represent the curve in a cubic B-spline basis due to its nice approximation properties and minimum curvature properties. To compare it with the approximation of the region as a polygon, the ellipse is sampled uniformly and the samples are interpolated using the two techniques (linear and cubic splines). The average relative error in the three centered 2nd order moments vs the number or samples are plotted in Fig 7.1. It can be seen that the relative error is much smaller for the cubic spline interpolation even at low sampling rates and that it exhibits a faster decay. In the traditional scanning approach, the ellipse is scanned along the x and y axes with a step size Δ and the monomials are computed at the grid points assigned to the interior of the curve. Fig 7.2 shows the decay of the average relative error for an ellipse vs

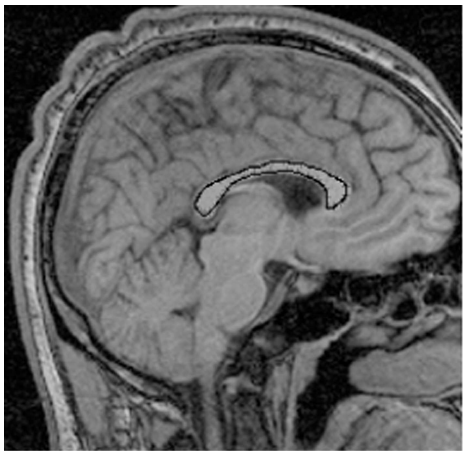


Figure 7.4: Shape of corpus callosum represented using a cubic B-spline curve with 20 knot points.

$\frac{\sqrt{\text{Area}}}{\Delta}$ for three different orientations. The plot clearly shows the dependence of the accuracy on the orientation of the ellipse.

It can be seen that to achieve a relative error of 0.1% the interior of the ellipse has to be sampled at about 3600 points, whereas to achieve the same error using the cubic spline interpolation we need only around 9 points on the curve. In comparison, the polygon method (linear spline) requires more than 40 samples to have a similar error. More interesting is the case when the interior of the ellipse has to be sampled at about 2.5×10^5 points to achieve an error of 0.002% while the cubic splines require only 25 samples to achieve the same accuracy.

In Fig 7.3, we show the ellipse corresponding to the 2nd order moments of the central structure in the image. The contour of the object was estimated using a snake where the curve was represented parametrically in terms of cubic B-splines; the moments are computed using our algorithm. Note that the fit is astonishingly good.

Having observed that the cubic spline estimator performs better than the polygon method, we now compare it with the Fourier(sinc) technique proposed in section 7.6. It is not fair to use the ellipse as we did before, because it can be represented exactly in a Fourier series representation with $L = 2$. So we choose

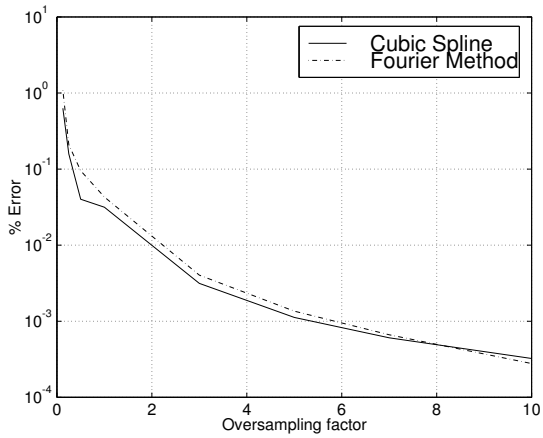


Figure 7.5: Comparison of Fourier Estimator with Cubic spline estimator.

the real shape of corpus callosum shown in Fig 7.4, represented in a linear spline basis with 39 knot points as the reference shape. This shape was resampled at different rates and these points were interpolated using cubic B-spline and Fourier representations respectively. The moments of the corresponding curves were calculated using the respective algorithms discussed before. Fig (7.5) shows the decay of the relative error with the resampling rate for both representations. We observe that the spline estimator is better than the Fourier estimator for small sampling rates, while the Fourier estimator performs better at very high sampling rates(typically more than 8 times the number of points used for the description of $\mathcal{C}$). In the example considered, the Fourier method performs better when the shape of corpus callosum is represented with around 312 samples.

To evaluate the performance of the approximate scheme introduced in 7.6.2, we now consider the case where the corpus callosum is represented by a cubic B-spline curve with 20 knot points. The relative error in the computation of the 2nd order moments by the quadrature formula as a function of its relative computational complexity(proportional to P) is shown in Fig. 7.6; here, the reference method is the kernel based computation, which is exact. Our results indicate that, for the 2nd order moments, the error of the quadrature formula is quite substantial (*eg.* 9.4%). Thus, it is not advantageous for computing the

lower order moments. However, the quadrature formula will eventually start to pay off for higher order moments, because its cost increases only quadratically with the degree as compared to exponentially for the kernel based method.

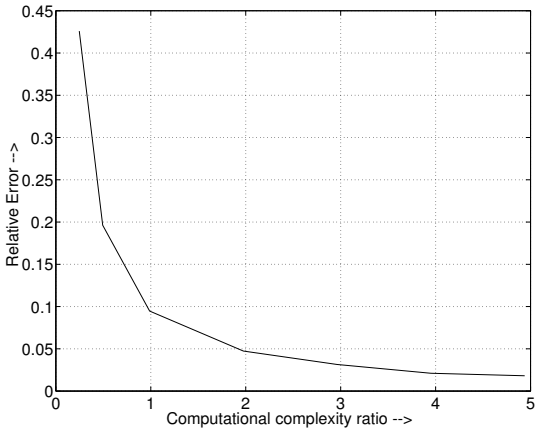


Figure 7.6: Relative Error vs Relative Computational complexity.

7.8 Synopsis

In this chapter we have presented a new approach for the computation of the moments of a curve described in a wavelet or scaling function basis. It is especially useful for objects detected using parametric snakes. The main advantages of the proposed scheme over the conventional methods are:

- the exactness of the computation;
- its independence of the orientation of the shape;
- the consistency with the snake model and the fact that it is the most direct method available.

In addition, the method is reasonably fast and easy to implement.

We recommend using our exact kernel-based approach for computing the lower order moments (typically $m + n \leq 2$) for which the kernels are available. For higher order moments, we have proposed a quadrature formula that

approximates the continuous integrals with Riemann sums. The latter method is exact for the sinc basis functions; otherwise it can be made as accurate as desirable by resampling the model at a finer rate (P sufficiently large).

Chapter 8

Conclusion

We have presented a coherent and consistent approach (cf. Fig. 1.4) for the estimation of shape and shape attributes from images. In contrast with the traditional sequential approaches, our scheme is centered on a shape model that drives the feature extraction, shape optimization and the attribute evaluation modules. Since it is more constrained that the traditional method, it is more robust to noise.

We have addressed several problems associated with the separate modules of this framework, keeping in mind the overall structure. Specifically, we have addressed the extraction of features, the representation of the shape and the evaluation of attributes from shapes. We applied the framework to solve two practical shape estimation problems. The main contributions of this thesis are listed below.

8.1 Main Contributions

- A general approach for the design of 2-D features from a class of steerable functions based on a Canny-like criterion: As compared to previous computational designs, our approach is truly 2-D. It provides filters with closed form expressions and better orientation selectivity than the conventional detectors.

- Several improvements for parametric snakes: Since the widely-used gradient magnitude-based energy is parameter dependent, we proposed a parameter independent term based on a steerable feature space. This term

accounts for the direction of the gradient and hence is more robust. Using Green's theorem, we re-expressed it as a surface integral, thus unifying it naturally with the region-based schemes.

We clarify some earlier statements about splines by showing that parametric snakes can guarantee low curvature curves, but only if they are described in the curvilinear abscissa. Since normal curve evolution do not ensure constant arc-length, we proposed a new internal energy term that will force this configuration.

We introduced an efficient scheme to check for the presence of loops in the curve. We also presented several practical enhancements to make the parametric framework even more attractive for the segmentation of biomedical images.

- A carefully engineered algorithm for the shape estimation of 3-D DNA molecules from its stereo cryo-micrographs: We used a global 3-D model and optimized its coefficients such that its projections matched with the micrographs. We approximated the global model locally by a projection-steerable elongated blob-like template; its projections onto the plane are contained in a finite dimensional space. We also derived an efficient algorithm to compute the likeliness of a 3-D filament with a specific orientation in 3-D space; the image energy of the 3-D snake is obtained by integrating the likeliness measures along the 3-D curve. Since we knew the final length of the DNA molecule, we used it to constrain the reconstruction.

- Quantitative analysis of error of the parametric representation of closed curves: We derived an exact expression for the L_2 error in approximating a periodic signal in a basis of shifted versions of a generating function. The formula takes the simple form of a Parseval's like relation where the Fourier coefficients of the signal are weighted against a frequency kernel that characterizes the approximation operator. This expression can be used to calculate the optimal number of coefficients and basis functions for a specific family of shapes.

- Exact computation of area moments: Using Green's Theorem, we showed that the computation of the area moments of a scaling function curve is equivalent to applying a suitable multidimensional filter on the coefficients of the curve and thereafter computing a scalar product. The multidimensional filter coefficients are pre-computed exactly as the solution of a two-scale relation. This algorithm can be used to evaluate the moments to constraint the reconstruction as shown in Fig. 1.4.

8.2 Future work

We now discuss a few directions along which we plan to continue our work.

- 3-D steerable feature detectors: The framework introduced in Chapter 3 can be extended to design feature detectors in 3-D. We plan to generate a steerable feature space from the volume data and use it to detect specific features like lines, surfaces, edges, blobs etc.

- Multi-scale detection of image features and denoising: One can decompose a given image using a multi-scale steerable pyramid to detect a a variety of multi-scale features (edges, ridges, corners etc.). By selectively preserving specific image features and reconstructing the image, we can obtain a denoised version of the image.

- Model-based reconstruction in limited angle tomography: The concept of projection steerablity can be used to reconstruct specific 3-D features from their 2-D projections. A higher order projection-steerable detector may be useful in tomography problems where the projections are noisy or when more views are difficult or impossible to generate.

- Account for the exact derivatives of the unified image energy: At the moment, the parameters of the probability distribution functions are assumed to be constant. However, in practice these parameters are estimated from the images depending of the current position of the contour. Thus these parameters are dependent on the curve coefficients. We would like to compute the exact directional derivatives in the optimization scheme and study the improvement.

- Application of the snake to practical problems: We would like to customize the snake models to practical biomedical problems. This will require several enhancements ranging from the choice of the image energy and type of shape constraint.

- 3-D active contour model: We plan to to extend the model-based consistent segmentation using a steerable feature space to 3-D. The parametric representation of general 3-D surfaces is difficult, unless the shapes assume simple forms like a tube; in this case the surface can be represented as using a spline model. Another promising approach may be to represent the shape using spherical harmonics as in [122, 123].

- Performance bounds on DNA shape estimation: The estimate of the position and orientation of the filament at a specified 3-D point will depend on the orientation of the filament and the projection geometry. We plan to compute the theoretical (Cramer Rao) bounds on the estimation error. These bounds will enable us to understand the problem better.

Bibliography

[1] D'Arcy Thompson, *On Growth and Form*, Cambridge University Press, 1942.

[2] F. L. Bookstein, *The measurement of biological shape and shape change*, Springer-Verlag, 1978.

[3] I. Stewart, *Life's other Secret*, John Wiley and sons, 1997.

[4] D. G. Kendall, *Shape and shape theory*, Wiley, Chichester, 1999.

[5] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*, John Wiley and Sons, 1998.

[6] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Transactions on Computers*, pp. 21–269, 1972.

[7] H. Freeman, "Computer processing of line-drawing images," *ACM Comput. Surveys*, vol. 6, pp. 57–97, 1974.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–332, 1988.

[9] S. Menet, P. Saint-Mark, and G. Medioni, "B-snakes: implementation and application to stereo," in *Image Understanding workshop*, 1990, pp. 720–726.

[10] A. Chakraborthy, L. H. Staib, and J. S. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging*, vol. 15, no. 6, pp. 859–870, 1996.

[11] H. Blum and R. N. Nagel, "Shape description using weighted symmetric axis features," in *IEEE Computer Society Conference on Pattern Recognition and Image processing*, 1977.

[12] L. X. Shen and Y. L. Sheng, "Noncentral image moments for invariant pattern recognition," *Optical Engineering*, vol. 34(11), pp. 3181–3186, 1995.

[13] F. Mokhtarian and A. K. Mackworth, "A theory of multi-scale, curvature-based shape representation for planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 789–805, 1992.

[14] V. V. Kindratenko, "On using functions to describe shape," *Journal of Mathematical Imaging and Vision*, vol. 18, pp. 225–245, 2003.

[15] R. H. Bartels, J. C. Beartty, and B. A. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, Morgan Kauffmann Publishers Inc, 1987.

[16] L. Piegl and W. Tiller, *The NURBS book*, Springer Verlag, 1997.

[17] 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, 2000.

[18] G. Sapiro, *Geometric partial differential equations and image analysis*, Cambridge University Press, 2001.

[19] J. A. Sethian, *Level Set Methods and Fast Marching methids*, Cambridge University press, 1998.

[20] V. Casselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *ICCV*, 1995.

[21] G. Aubert, M. Barlaud, O. Faugeras, and S.J. Besson, "Image segmentation using active contours: calculus of variations or shape gradients," *SIAM Journal of Applied Mathematics*, 2003.

[22] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[23] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, 1995.

[24] M. Unser, "Sampling–50 Years after Shannon," *Proceedings of the IEEE*, 2000.

[25] M. Unser and A. Aldroubi, "A general sampling theorem for nonideal acquistion devices," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2915–2925, 1994.

[26] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part I–Interpolators and projectors," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2796–2806, 1999.

[27] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[28] I. J. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.

[29] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I–Theory," *IEEE Transactions on Signal Processing*, pp. 821–833, 1993.

[30] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II–Efficient design and applications," *IEEE Transactions on Signal Processing*, pp. 834–848, 1993.

[31] A. Munoz, T. Blu, and M. Unser, "Least-squares image resizing using finite differences," *IEEE Transactions on Image Processing*, pp. 1365–1378, 2001.

[32] F. Candocia and J. C. Prince, "Comments on sinc interpolation of discrete periodic signals," *IEEE Transactions on Signal Processing*, vol. 11, pp. 2044–2047, 1998.

[33] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Processing*, vol. 28, pp. 127–138, 1992.

[34] G. C. H. Chuang and J. Kuo, "Wavelet descriptor of planar curves: theory and applications," *IEEE Transactions on Image Processing*, vol. 5, pp. 56–70, 1996.

[35] J. P. Antoine, D. Barache, R. M. Cesar, and L. F. Costa, "Shape characterisation with the wavelet transform," *Signal Processing*, vol. 62, pp. 265–290, 1997.

[36] J. Y. Wang and F.S. Cohen, "3-D object recognition and shape estimation from image contours using B-splines, unwarping techniques and neural network," in *IEEE International Joint Conference on Neural Networks, New York, NY, USA*, 1991.

[37] F.S. Cohen and Z. Huang, "Affine-invariant B-spline moments for curve matching," *IEEE Transactions on Image Processing*, vol. 5, pp. 1473–1480, 1996.

[38] F.S. Cohen and J. Y. Wang, "Modeling image curves using invariant 3-D object curve models, a path to 3-D recognition and shape estimation from image contours, Part 1,'," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1–12, 1994.

[39] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[40] S. Sarkar and K.L. Boyer, "On optimal infinite impulse response filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1154–1170, 1991.

[41] R. Deriche, "Fast algorithms for low-level vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 78–87, 1990.

[42] W.T. Freeman and E.H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[43] P. Perona, "Deformable kernels for early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 488–499, 1991.

[44] R. Manduchi, P. Perona, and D. Shy, "Efficient deformable filter banks," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1168 –1173, 1998.

[45] E.P. Simoncelli and H. Farid, "Steerable wedge filters for local orientation analysis," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1377–1382, 1996.

[46] P. C Teo and Y. Hel-Or, "Design of multi-parameter steerable functions using cascade-basis reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 552–556, 1999.

[47] P Campisi and G. Scarano, "A multiresolution approach for texture synthesis using the circular harmonic functions," *IEEE Transactions on Image Processing*, vol. 11, pp. 37–51, 2002.

[48] M.N. Do and M. Vetterli, "Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models," *IEEE Transactions on Multimedia*, pp. 146–158, 2002.

[49] E.P. Simoncelli and W.T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *International Conference on Image Processing*, 1995, pp. 444–447.

[50] A.F. Frangi, W.J. Niessen, K.L. Vincken, and M.A. Viergever, "Multiscale vessel enhancement filtering," in *Medical Image Computing and Computer-Assisted Intervention*. 1998, vol. 1496, pp. 130–137, Springer Verlag.

[51] M. Jacob, T. Blu, and M. Unser, "3-D reconstruction of dna filaments from stereo cryo-electron micrographs," in *IEEE International Symposium on Biomedical Imaging: Macro to Nano*, 2002, vol. 2, pp. 597–600.

[52] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C Scharlah, "Ridges for image analysis," *Journal of Mathematical Imaging and Vision*, pp. 353–373, 1994.

[53] W.H Press, S.A Teukolsky, W.T Vetterling, and B.P Flannery, *Numerical recipes in C*, Cambridge University Press, 1997.

[54] M.A. Ruzon and C. Tomasi, "Color edge detection with the compass operator," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, vol. 2, pp. 160–166.

[55] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," *The International Journal of Computer Vision*, vol. 10(2), pp. 101–124, 1993.

[56] L. Kitchen and A. Rosenfield, "Gray level corner detection," *Pattern Recognition Letters*, pp. 95–102, 1982.

[57] Z. Zheng, H. Wang, and E.K. Teoh, "Analysis of gray level corner detection," *Pattern Recognition Letters*, vol. 20, pp. 149–162, 1999.

[58] H. Wang and M. Brady, "Real time corner detection algorithm for motion estimation," *Image and Vision Computing*, vol. 13, no. 9, pp. 695–703, 1995.

[59] R. Deriche and O. Faugeras, "2-D curve matching using high curvature points: application to stereo vision," in *10th International Conference on Pattern Recognition*, 1990, vol. 1, pp. 240–242.

[60] A.A. Bharath, "Steerable filters from Erlang functions," in *The British Machine Vision Conference*, 1998, pp. 144–153.

[61] D. Demigny, "On optimal linear filtering for edge detection," *IEEE Transactions on Image Processing*, vol. 11, no. 7, pp. 728–737, 2002.

[62] W. Rasband, "ImageJ web site," *http://rsb.info.nih.giv/ij/*.

[63] A.K. Jain, Y. Zhong, and M.P.D. Jolly, "Deformable template models: A review," *Signal Processing*, vol. 76, pp. 109–129, 1998.

[64] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical Image Analysis*, vol. 1, pp. 91–108, 1996.

[65] C. Xu and J. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[66] J. Gao, A. Kosaka, and A. Kak, "A deformable model for human organ extraction," in *ICIP*, Chicago, 1998, pp. 323–327.

[67] M. Gebhard, J. Mattes, and R. Eils, "An active contour model for segmentation based on cubic B-splines and gradient vector flow," in *MICCAI*, 2001.

[68] M. A. Figueiredo and J. M. N. Leitao, "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, 2000.

[69] L. H. Staib and J. S. Duncan, "Boundary fitting with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, 1992.

[70] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–174, 1995.

[71] N. Paragios and R. Deriche, "Unifying boundary and region-based information for geodesic active tracking," in *IEEE Computer Vision and Pattern Recognition*, Forth Collins, Colarado, 1999.

[72] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert, "Simeltaneous inward and outward curve evolution," in *ICIP*, Kobe, Japan, 1999.

[73] M. Flickner, H. Sawhney, D. Pryor, and J. Lotspiech, "Intelligent interactive image outlining using spline snakes," in *28th Asilomar Conference on Signals, Systems and Computers*, 1994.

[74] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images," *Image and Vision computing*, vol. 12, pp. 355–365, 1994.

[75] M. Jacob, T. Blu, and M. Unser, "A unifying approach and interface for spline-based snakes," *SPIE International Symposium on Medical Imaging: Image Processing*, vol. 4322, pp. 340–347, 2001.

[76] C. Davatzikos, X. Tao, and D. Shen, "Hierarchical active shape models, using the wavelet transform," *IEEE Transactions on Medical Imaging*, 2003.

[77] M. Leventon, O. Faugeras, E. Grimson, and W. Wells, "Level set based segmentation with intensity and curvature priors," in *Mathematical Methods in Biomedical Image Analysis*. 2000, IEEE.

[78] H. Park, T. Schoepflin, and Y. Kim, "Active contour model with gradient directional information: Directional snake," *IEEE Transactions on Circuits and systems for video technology*, vol. 11, no. 2, pp. 252–256, 2001.

[79] C. Chesnaud, P. Refregier, and V. Boulet, "Statistical region snake-based segmentation adapted to different physical noise models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1145–1157, 1999.

[80] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266276, 2001.

[81] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 884–900, 1996.

[82] M. Jacob and M. Unser, "Design of steerable filters for feature detection using a Canny-like criterion," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[83] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *International Conference on Image Processing*, Corfou, Greece, 1999, pp. 688–694.

[84] L.D. Cohen and I. Cohen, "Finite-element methods for active contour models and baloons for 2-D and 3-D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131–1147, 1993.

[85] F. Precioso and M. Barlaud, "Regular B-spline active contours for fast video segmentation," in *IEEE International Conference on Image Processing*, Rochester, USA, 2003.

[86] 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, pp. 633–642, 2001.

[87] E. Weisstein, "Eric Weisstein's world of mathematics," *http://mathworld.wolfram.com.*

[88] J. Dubochet, M. Adrian, J. Chang, J. C. Homo, J. Lepault, A. W. McDowall, and P. Schultz, "Cryo-electron microscopy of vitrified specimens," *Quarterly review of Biophysics*, vol. 21(2), pp. 129–228, 1988.

[89] J. Dubochet, M. Adrian, I. Dustin, P. Furrer, and A. Stasiak, "Cryoelectron microscopy of DNA molecules in solution," *Methods Enzymol.*, vol. 211, pp. 507–518, 1992.

[90] M. Adrian, B. H. Bordier, W. Wahli, A. Stasiak, and J. Dubochet, "Direct visualization of supercoiled DNA molecules in solution," *The EMBO journal*, vol. 9(13), pp. 4551–4554, 1990.

[91] I. Dustin, P. Furrer, A. Stasiak, J. Dubochet, J. Langowski, and E. Egelman, "Spatial visualization of DNA in solution," *Journal of Structural Biology*, vol. 107, pp. 15–21, 1991.

[92] F. P. I. Margalef, P. Furrer, M. Kunt, and J. Dubochet, "The flying cylinder: A new algorithm for filament recognition in noisy stereo images," *Journal of Structural Biology*, vol. 116, pp. 25–29, 1996.

[93] J. Bednar, P. Furrer, V. Katritch, A. Stasiak, J. Dubochet, and A. Stasiak, "Determination of DNA persistence length by cryo-electron microscopy," *Journal of Molecular Biology*, vol. 254, pp. 579–594, 1995.

[94] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.

[95] M. Jacob and M. Unser, "Design of steerable filters for feature detection using Canny-like criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[96] C. Canero, P. Radeva, R. Toledo, J. J. Villanueva, and J. Mauri, "3-D curve reconstruction by biplane snakes," in *International Conference on Pattern Recognition*, 2000, pp. 4563–4567.

[97] C. Canero, F. Vilarino, J. Mauri, and P. Radeva, "Predictive undistortion model and 3-D reconstruction by biplane snakes," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1188–1201, 2002.

[98] D. J. Struik, *Lectures on classical differential geometry*, Dover Publications, 2nd edition, 1988.

[99] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part II–Wavelets," *IEEE Transactions on Signal Processing*, vol. 47, 1999.

[100] A. J. E. M Janssen, "The Zak transform and sampling theorems for wavelet subspaces," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3360–3364, 1992.

[101] G. H. Granlund, "Fourier preprocessing for hand print character recognition," *IEEE Transactions on Computers*, vol. 21, pp. 195–201, 1972.

[102] L. Schwartz, *Theorie des Distributions*, Hermann, 1988.

[103] G. Friedlander and M. Joshi, *Introduction to the Theory of Distributions*, Cambridge University Press, 1988.

[104] S. R. Doodey and A. K. Nandi, "Notes on interpolation of discrete periodic signals using sinc function related approaches," *IEEE Transactions on Signal Processing*, vol. 48, pp. 1201–1203, 2000.

[105] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory," *Numer. Funct. Anal. Opt.*, vol. 42, pp. 1–21, 1994.

[106] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Transactions on Signal Processing*, vol. 44, pp. 519–527, 1996.

[107] T. Blu and M. Unser, "Approximation error for quasi-interpolators and multi-wavelet expansions," *Appl. Comput. Harmon. Anal.*, vol. 6, pp. 219–251, 1999.

[108] S. Rad, K.C. Smith, and B. Benhabib, "Application of moment and Fourier descriptors to the accurate estimation of elliptical shape parameters," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1991.

[109] R. R. Desai and H. D. Cheng, "Pattern recognition by local radial moments," in *12th IAPR International Conference on Pattern Recognition*, 1994.

[110] K. Tsirikolias and B. G. Mertzios, "Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition," *Pattern-Recognition*, vol. 26, pp. 877–82, 1993.

[111] Y. Wang, S. L. Lee, and K. Toraichi, "Multiscale curvature based shape representation using Bspline Wavelets," *IEEE Transactions on Image Processing*, vol. 8, pp. 1586–1592, 1999.

[112] M. Singer, "A general approach to moment calculation for polygons and line segments," *Pattern Recognition*, vol. 26, pp. 1019–1028, 1993.

[113] S. F. Bockman, "Generalising the formula for areas of polygons to moments," *American Mathematical Monthly*, vol. 96, pp. 131–133, 1989.

[114] N. J. C. Strachan, P. Nesvadba, and A. R. Allan, "A method for working out the moments of a polygon using an integration technique," *Pattern Recognition Letters*, vol. 11, pp. 351–354, 1990.

[115] C. Fermuller and W. Kropatsch, "Hierarchical curve representation," in *11th IAPR International Conference on Pattern Recognition*, 1992.

[116] S. Mallat, "A theory for multiresolution signal decomposition : the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[117] T. Blu, "Iterated filter banks with rational rate changes connection with discrete wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3232–3244, 1993.

[118] J. M. Restrepo and G. K. Leaf, "Inner product computations using periodized daubechies wavelets," *International Journal of Numerical Methods in Engineering*, vol. 40, pp. 3557–3578, 1997.

[119] W. Dahmen and C. A. Micchelli, "Using the refinement equation for evaluating integrals of wavelets," *SIAM Journal of Numerical Analysis*, vol. 30, pp. 507–537, 1993.

[120] C. D. Boor, K. Hollog, and S. Riemenschneider, *Box Splines*, Springer-Verlag, 1998.

[121] M. D. McCool, "Optimised evaluation of box splines via the inverse FFT," *Graphics Interface*, 1995.

[122] M. Quicken, C. Brechbuhler, J. Hug, H. Blattman, and G. Szekely, "Parameterization of closed surfaces for parametric surface description," in *Computer Vision and Pattern Recognition*, 2000.

[123] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig, "Segmentation of 3D objects from MRI volume data using constrained elastic deformations of flexible fourier surface models," in *Computer Vision, Virtual Reality and Robotics in Medicine*, 1995.

Curriculum vitæ

ADDRESS

Professional:

Biomedical Imaging Group,
Swiss Federal Institute of Technology,
EPFL/STI/BIO-E, BM 4.141,
Switzerland
Tel: +41-21-693-5143
Fax: +41-21-693-3701

e-mail: Mathews.Jacob@ieee.org
URL: http://bigwww.epfl.ch/jacob

PERSONAL DETAILS

Date of Birth : 16/06/1975
Place of Birth : Kerala, India
Nationality : Indian

EDUCATION

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

Thesis: Parametric Shape Processing in Biomedical Imaging

Dept. of ECE, Indian Institute of Science, Bangalore, India

Master of Engineering in Signal Processing — April 1999

Dept. of ECE, Regional Engineering College (Now NIT), Calicut, India

Bachelor of Technology — July 1996

PROFESSIONAL EXPERIENCE

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

Research Assistant: Biomedical Imaging Group, with Prof. Michael Unser
. 1999 - 2003

- Design of optimal steerable filters for feature detection
- 3-D reconstruction of DNA filaments from stereo cryo-electron micrographs
- Efficient algorithms for spline-based snakes (active-contours)
- Exact algorithm for the computation of area moments of spline and wavelet curves
- Robust sampling of multi-channel data
- Quantitative error analysis in sampling periodic signals

Indian Institute of Science, Bangalore, India

M.E Thesis with Prof. G.V. Anand, Dept. of ECE 1998-1999

- Extension of conventional multi-resolution analysis
- New multi-wavelets for fractal image coding

Wipro Infotech R&D, Bangalore, India

Hardware Engineer 1996

- Design of traffic concentrator boards for ATM switches

Regional Engineering College, Calicut, India

Btech Thesis 1996

- Implementation of Fractal Image Coding

Summer Training 1995

- Design and development of an electronic rain gauge

TEACHING EXPERIENCE

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

Teaching Assistant for 2000 - 2003

- Wavelets and Applications (graduate course)
- Image Processing 1&2 (undergraduate course)

Project Supervison 2000 - 2003

- Francois Auget, "Feature detection using optimal steerable filters"
- Alwyn Fernandez, "Denoising of fluorescent microscopy images using edge/ridge preserving PDE "
- Geiser Pascal, "Background correction for the removal of auto-fluorescence in microscopy"
- Roland Michaely, "Measurement of the variation of vascular diameters by ultrasonic imaging"
- Anil Swaroop, "Mensuration of neuronal evolution in culture"
- Antoine Luisier, "Quantification of image similarity using invariants"
- Shai Tirosh, "Volume segmentation using level set snakes"

PROFESSIONAL ACTIVITIES

Regular reviewer for

- IEEE Transactions on Pattern Analysis and Machine Intelligence
- IEEE Transactions on Image Processing
- IEEE Transactions on Signal Processing
- IEEE Transactions on Medical Imaging
- Journal of Mathematical Imaging and Vision
- Advances in Computational Mathematics

Student Member of the IEEE

PUBLICATIONS

Journals

1. M. Jacob, T. Blu, 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.

2. M. Jacob, T. Blu, M. Unser, "An Error Analysis for the Sampling of Periodic Signals," Proceedings of the Fourth International Conference on Sampling Theory and Applications (SampTA'01), Orlando FL, USA, May 13-17, 2001, pp. 45-48.

3. M. Jacob, M. Unser, "Design of Steerable Filters for Feature Detection Using Canny-Like Criteria," IEEE Transactions on Pattern Analysis and Machine Intelligence, in press.

4. M. Jacob, T. Blu and M. Unser, "Efficient Algorithms for Spline Based Active Contours," submitted to IEEE Transactions on Image Processing.

5. E.Meijering, M. Jacob, J. C. F. Sarria, P. Steiner, H. Hirling and M. Unser, "Design and Validation of a Tool for Neurite Tracing and Analysis in Fluorescence Microscopy," submitted to Cytometry.

6. M. Jacob, T. Blu, C.Vaillant, J.Maddocks and M. Unser, "Reconstruction of DNA filaments from Stereo Cryo-electron Micrographs," submitted to IEEE Transactions on Image Processing.

7. M. Jacob, T. Blu and M. Unser, "Least Squares Multichannel Sampling of Noisy Data," to be submitted to IEEE Transactions on Signal Processing.

8. M. Jacob, T. Blu and M. Unser, "Regularized Multichannel Sampling of Noisy Data," to be submitted to IEEE Transactions on Signal Processing.

Referred Conference Proceedings

1. M. Jacob, T. Blu, M. Unser, "Exact Computation of Area Moments for Spline and Wavelet Curves," Proceedings of the Fifteenth International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, September 3-8, 2000, vol. III, pp. 131-134.

2. M. Jacob, T. Blu, M. Unser, "An Error Analysis for the Sampling of Periodic Signals," Proceedings of the Fourth International Conference on Sampling Theory and Applications (SampTA'01), Orlando FL, USA, May 13-17, 2001, pp. 45-48.

3. M. Jacob, T. Blu, M. Unser, "A Unifying Approach and Interface for Spline-Based Snakes," Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'01), San Diego CA, USA, February 17-22, 2001, vol. 4322, Part I, pp. 340-347.

4. M. Feilner, M. Jacob, M. Unser, "Orthogonal Quincunx Wavelets with Fractional Orders," Proceedings of the 2001 IEEE International Conference on Image Processing (ICIP'01), Thessaloniki, Greece, October 7-10, 2001, vol. I, pp. 606-609.

5. M. Jacob, T. Blu, M. Unser, "3-D Reconstruction of DNA Filaments from Stereo Cryo-Electron Micrographs," Proceedings of the First 2002 IEEE International Symposium on Biomedical Imaging: Macro to Nano (ISBI'02), Washington DC, USA, July 7-10, 2002, vol. II, pp. 597-600.

6. E. Meijering, M. Jacob, J.-C.F. Sarria, M. Unser, "A Novel Approach to Neurite Tracing in Fluorescence Microscopy Images," Proceedings of the Fifth IASTED International Conference on Signal and Image Processing (SIP'03), Honolulu HI, USA, August 13-15, 2003, M.H. Hamza Ed., ACTA Press, Calgary, pp. 491-495.

7. M. Jacob, M. Unser, "Optimal Steerable Filters for Feature Detection," Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP'03), Barcelona, Spain, September 14-17, 2003, pp. III.433-III.436.

Conference Abstracts

1. M. Unser, D. Sage, M. Jacob, E. Meijering, P. Thevenaz, "Image Processing and Analysis of Biological Images," Live Cell Imaging Workshop, Swiss Federal Institute of Technology, Lausanne, October 9-10, 2002

2. E. Meijering, M. Jacob, J-C. F. Sarria, P. Steiner, H. Hirling, M. Unser, "A Novel Tool for Neurite Tracing and Analysis in Fluorescence Microscopy," EMBO Workshop on Advanced Light Microscopy, Barcelona, 2003.

Reports

1. ME Thesis, Submitted at Dept of ECE, IISc, Bangalore on April 1999. (Downloadable from http://bigwww.epfl.ch/jacob/MEthesis.html)