INSTITUT FÜR INFORMATIK

UND PRAKTISCHE MATHEMATIK

Local Orientation Analysis in Images and Image Sequences Using Steerable Filters

Weichuan Yu

Bericht Nr. 2012

15. Dezember 2000

PAX OPTIMA RERUM

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik und Praktische Mathematik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Local Orientation Analysis in Images and Image Sequences Using Steerable Filters

Weichuan Yu

Bericht Nr. 2012
15. Dezember 2000

e-mail: wy@ks. informatik.uni-kiel.de

Dieser Bericht ist als persönliche Mitteilung aufzufassen.

Dieser Bericht gibt den Inhalt der Dissertation wieder, die der Verfasser im Juli 2000 bei der Technischen Fakultät der Christian-Albrechts-Universität zu Kiel eingereicht hat.

1. Gutachter:	Prof. Dr. G. Sommer (Universität Kiel)
2. Gutachter:	Prof. Dr. H. Niemann (Universität Erlangen)
3. Gutachter:	Prof. Dr. K. Daniilidis (University of Pennsylvania)

Datum der mündlichen Prüfung: 1. November 2000

Abstract

In this thesis, we address the issue of local orientation analysis using steerable filters. From the standpoint of the sampling theory, current orientation steerable filters sample the spectrum of the orientation space with Dirac functions. According to the well known uncertainty principle, we cannot simultaneously localize a signal both in the spatial domain and in the spectral domain exactly. This kind of uncertainty has a lower bound which can be achieved only using filters with Gaussian shape. With respect to this criterion, current steerable filters are not optimal because Dirac functions localize spectral components of the signal so exactly that the spatial counterparts of these Dirac functions almost lose their localization ability totally. As a result, we have to combine a large number of basis filters in order to achieve high resolution in orientation, which increase the computational complexity.

Our contribution is that we use angular Gaussian filters in constructing steerable filters to achieve the lower bound in the uncertainty principle. Theoretical analysis and experimental results show that this new steerable filter achieves higher orientation resolution with lower computation complexity. These advantages benefit many applications ranging from 2D/3D junction characterization, volume image processing, facial analysis to symmetry detection, and specially multiple motion analysis.

We analyze occlusion and transparency in detail both in the spatial domain and in the spectral domain and propose a unified multiple motion model in the spectral domain. Using the fact that multiple motions are equivalent to multiple planes in the derivative space or in the frequency space, we apply our 3D steerable filter for multiple motion estimation. We compare our approach with current motion algorithms like the 3D Hough transform, expectation maximization algorithm, and early 3D steerable filter approaches.

In occlusion analysis we introduce a multi-window strategy to detect and to eliminate *outliers*. This improves the *quality* of input data and therefore provides more exact results in motion estimation. We further apply the “shift-and-subtract” technique to localize occlusion boundaries and to track their movement in occlusion sequences. This technique can also be used to distinguish occlusion from transparency and to decompose transparency scenes into multi-layers.

Acknowledgments

It is my great pleasure to thank a lot of people, without whom I would never finish this thesis.

DAAD (German Academic Exchange Service) provided me a graduate research fellowship, which enabled me to come to Germany in 1996 for further study. I am grateful for its financial support all these years.

Unlimited thanks go to my supervisor Professor Sommer for giving me the opportunity to finish this work in Kiel cognitive systems group. His valuable advice, high criterion and detailed comments on my work are main motivation of my research during these years. Besides, he kindly supports me without reservation to attend many conferences, to take part in the international course for graduate students in Italy, to make scientific visits, and to cooperate with the Erlangen group. All these interesting experiences have given me fresh stimulations, new ideas, helpful suggestions, and encouraging feedback to improve the work.

Thank Professor Kostas Daniilidis for his intensive advice and sincere help. Kostas kindly shared his deep insights in computer vision with me when I began my study in Kiel. From those extensive discussions I have learned a lot from him. I really appreciate his continued advice even when he moved to Philadelphia later. I will always remember his warm reception in the GRASP lab at University of Pennsylvania.

I am thankful to Professor Heute for introducing me to Professor Sommer, for his interest in my work, for beneficial discussions, and for his wonderful lectures which gave me a deeper understanding of the signal theory. His lecture notes have become standard references for my work. Besides, he has never hesitated about writing recommendations to support my DAAD fellowship extension whenever I needed them.

My best German friend Dr. Gerhard Birkelbach got to know me when he visited our group. It is very interesting to talk with him due to his extensive knowledge. His insightful suggestions have improved the wedge filter significantly. He helped me to visit the Fraunhofer IGD in Darmstadt, which was my first scientific visit in Germany. All of his family members, his father, wife, two daughters, and the dog *Dunja* are so nice to me that it is always my pleasure to visit his family.

I appreciate very much Professor Hany Farid's help. It was very nice of Hany to meet a "freshman" during his conference visit in Kiel just because I wrote him an Email. He spent a whole day to discuss with me about the steerable wedge filter. I am very grateful to him for providing the source code of the steerable wedge filter and the test image, for his detailed comments on the first script of my first paper, and for providing extended literature.

Professor Jähne has kindly invited me to visit his group in Heidelberg, where I had valuable discussions with him, Hanno Scharr and Dr. H. Haußecker. His concise explanation helped me to understand the tensor approach and the SVD method more clearly.

Thank Professor Niemann, Dr. Dietrich Paulus, and Arnd Gebhard for their interest in my work, which results in a successful cooperation between the Erlangen group and our group. Their hospitable reception made my visits not only fruitful, but also comfortable. Arnd has also amicably provided the example of facial analysis in this thesis.

Thank Professor Burkhardt and Xiaofeng Zhang for their friendly reception when I visited their group in Freiburg.

The colleagues in the Kiel cognitive systems group, M. Hansen, J. Pauli, T. Bülow, M. Felsberg, M. Michaelis, and V. Krüger as well as my Chinese friends Professor H. Chen, Dr. T. Zhang, X. Zhang, and Dr. H. Li have given constructive comments which improved this work. Specially, G. Birkelbach, J. Pauli, and M. Hansen have warmheartedly corrected the manuscript of this thesis.

Finally, I thank my parents, my brother, and my wife for their full support and encouragement all these years.

Contents

Acknowledgments **v**

1 Introduction **1**

1.1 Local Orientation Analysis 1

1.2 Steerable Filters 2

1.3 Our Contributions 3

1.4 Outline of the Thesis 4

2 2D Orientation Steerability **7**

2.1 Introduction 8

2.2 2D Local Orientation Analysis 12

Definition of Our Steerable Filter 12

Difference to Exact Steerability 15

Complexity Analysis 17

2.3 Polar Pyramid 21

2.4 Experiments 24

Synthetic Junction Examples 24

Real Examples 29

2.5 Conclusion 35

3 3D Orientation Steerability 37

3.1 Introduction 37

The Work of Freeman and Adelson 38

Andersson's Work 39

Remark about Orientation Resolution 39

3.2 Local 3D Orientation Analysis 40

Definition of 3D Approximately Steerable Filter 40

Filter Responses of 3D Planes 43

3.3 Comparisons 45

3.4 Applications in Volume Image Processing 47

Compensation Issue 47

Application Examples 48

3.5 Related Issues 52

Extended Gaussian Image and 3D Orientation Histogram 53

Spherical Wavelets 55

3.6 Conclusion 57

4 Multiple Motion Analysis 59

4.1 Introduction 59

Spatial Approaches 60

Spectral Approaches 62

Resolution Issue of Principal Axis Analysis 63

Gabor Based Approaches 64

Skewness of Gabor Wavelets 66

Our Contribution 67

4.2 Understanding Multiple Motions 68

Spatial Observation of Multiple Motions 68
Spectral Analysis of Occlusion 68
Spectral Analysis of Transparency 71
Spectral Model of Multiple Motions 71
4.3 Spatial Model vs. Spectral Model 72
4.4 Multiple Motion Estimation Using 3D Steerable Filter 74
3D Hough Transform Based Algorithm 75
Spatial and Spectral Expectation Maximization Algorithm 78
4.5 Outlier Issue in Occlusion Estimation 80
Detection of Outliers 80
Precision Improvement after Eliminating Outliers 82
4.6 Scene Analysis 82
4.7 Experiments 85
Real Occlusion Analysis 91
4.8 Discussions 97

5 Summary **103**

List of Figures

1.1 The principle of steerable filter 3

2.1 The first derivative of 2D Gaussian 9

2.2 Gaussian wedge mask 13

2.3 Basis filter comparison 16

2.4 Relation between filter sizes 18

2.5 Complexity comparison between different steerability approaches . 20

2.6 Junction with orientation scale variation 22

2.7 Polar pyramid structure 23

2.8 Synthetic line junctions 25

2.9 Synthetic edge junctions 25

2.10 Robustness against noise 26

2.11 Keypoint deviation 26

2.12 Siemens star 27

2.13 Pyramid junction characterization 28

2.14 Parkbench 30

2.15 Nasa image 31

2.16 Lena 32

2.17 Face symmetry detection 33

2.18 Child face 34

3.1 Conic kernel 41

3.2 Distribution of conic kernels 42

3.3 Special planes 44

3.4 Normal planes 44

3.5 Filter support comparison 46

3.6 3D junction characterization 49

3.7 Cubic 50

3.8 Cup 51

3.9 Propeller 52

3.10 Extended Gaussian image 53

3.11 Spherical tessellation 54

3.12 The relation between (θ_n, ϕ_n) and (θ_m, ϕ_m) 58

4.1 Same orientation resolution principle 65

4.2 The skewness of the Gabor wavelet 67

4.3 Difference between occlusion and transparency 69

4.4 Orientation signatures of occlusion and transparency sequences . . 73

4.5 3D Filter equivalent to the Hough transform 76

4.6 Hough images of points and planes 76

4.7 Orientation signatures before and after eliminating outliers 83

4.8 “Shift-and-Subtract” technique 84

4.9 Real transparency sequence 89

4.10 Comparison of different models on real transparency sequence . . 90

4.11 Flower garden sequence 92

4.12 Real occlusion sequence 93

4.13 Real occlusion estimation 94

4.14 Block world sequence 95
4.15 Block world estimation 96
4.16 Aperture problem 100

Chapter 1

Introduction

1.1 Local Orientation Analysis

The question **"what is where?"** is so trivial for human that even a five-year-old child can provide the correct answer with no difficulty. However, it is a great challenge hitherto to enable a machine to answer this question. The reason lies in the fact that the vision principle of human is still elusive to us.

The goal of computer vision is to comprehend the principle of vision from a mathematical aspect. Computer vision, which is defined by Marr as "a computational investigation into the human representation and processing of visual information" [Mar82], has experienced rapid growth over the last three decades since its foundation in the 1960's. The themes in this area range from simple zero-crossing technique for edge detection to object tracking and facial recognition.

Roughly speaking, computer vision falls into low-level vision and high-level vision. In the low-level vision, we detect and extract features from sensor outputs (such as images and image sequences), while in the high-level vision, we further analyze and process these features. Feature [Koe93], taken literally, is an entity of distinctive characteristic. It may be brightness, color, texture, contour, shape, topological structure, and so on. Choosing an appropriate model of features is the foremost and the most difficult step in computer vision. A robust model should be capable of describing features under different kinds of conditions such as the change of illumination, the variation of view angle, and so on.

A feature is considered with respect to its geometric neighborhood in images or image sequences. Therefore, the geometric description of features is one of the most important models. Being the basic geometric transforms, translation, rotation, and scaling (dilation and shrinking) provide us the position, orientation, and scale information, respectively.

One of the basic features in images and image sequences is orientation. It is useful in pattern recognition, matching, coding, object tracking, facial analysis, and many other areas. Thus, it is very meaningful to extract orientation information accurately and efficiently. In this thesis, we address the problem of local orientation analysis and propose a new approach to improve current orientation analysis methods.

1.2 Steerable Filters

In order to analyze orientation, we need a set of filters pointing to different directions. Hence, the Laplace operator or tensor based approaches, providing only the gradient direction or main axis of the signal structure, are not sufficient. Moreover, elongation of the filter shapes is necessary to achieve fine resolution in the orientation space. But these requirements usually result in enormous complexity of computation.

Based on the inherent relation among differently deformed versions of the same filter with respect to the deformation parameters, Freeman and Adelson [FA91] and other people resorted to *steerable filters* to alleviate the computational complexity. The steerability of a filter means that all deformed versions of this filter can be expressed or approximated as a linear combination of a finite number of *basis filters*. Because we can exchange the order of linear combination and convolution, the responses of a whole set of deformed filters can be expressed as a linear combination of the responses of basis filters. The less number of the basis filters, the less complexity of the computation. This principle is shown in figure 1.1. Many areas such as image analysis, motion estimation, computer graphics, and pattern recognition [Teo98] benefit from this new filter.

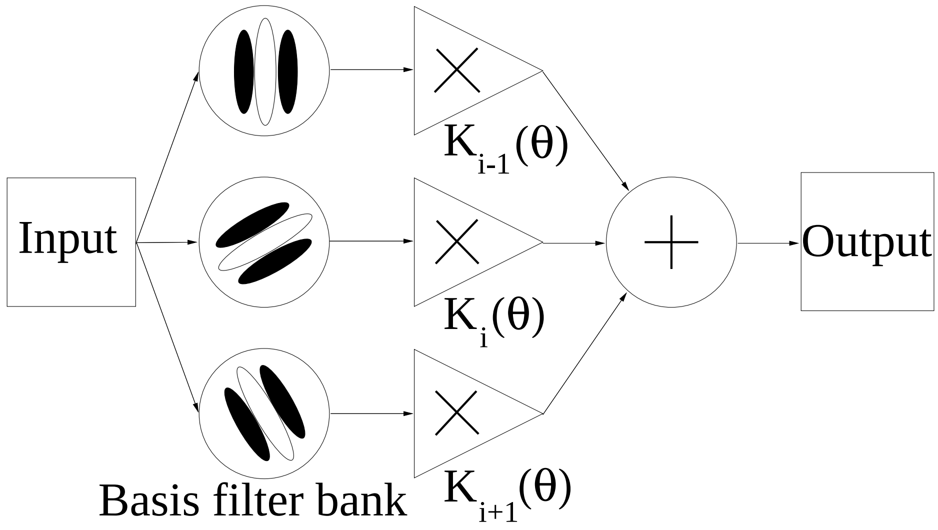


Figure 1.1: *The principle of steerable filter (redrawn from [FA91]). The response of the synthesized filter at an arbitrary orientation can be calculated as a linear combination of the basis filter outputs. The basis filter outputs are weighted by functions of orientation parameters.*

1.3 Our Contributions

The concept of steerability is actually another representation of signal decomposition theory if we consider the orientation parameters in steerable filters as variables. Besides, it is very much related to the sampling theory. In this thesis, we propose a new kind of steerable filter both in 2D space and in 3D space. This new filter substitutes the global decomposition principle used in current steerable filters with the local decomposition principle in order to reduce the redundancy among different basis filters. Specially, we use angular Gaussian functions to achieve the lower bound in the uncertainty principle [Gab46, Dau85]. Therefore, it is natural that the performance of our filter is superior to that of current steerable filters in achieving higher orientation resolution with lower complexity.

In this thesis,

- we apply the steerable filters in different orientation analysis tasks such as 2D junction characterization, facial analysis, 3D volume data processing, and multiple motion estimation.
- We show that a non-orthogonal basis may provide better performance than

an orthogonal basis in feature extraction.

- We compare our approach with other orientation analysis approaches and parameter estimation algorithms such as the tensor approach, Hough transform, expectation maximization (EM) algorithm, the 3D orientation histogram, and spherical wavelets.
- We provide a detailed analysis of multiple motions including occlusion and transparency and review current multiple motion estimation techniques. We specially compare the spatial motion model with the spectral motion model and point out their advantages as well as disadvantages. In occlusion analysis, we introduce a multi-window strategy to treat outliers near occlusion boundaries. We further use the "shift-and-subtract" technique to localize occlusion boundaries in one frame and to track their movement in occlusion sequences.

1.4 Outline of the Thesis

Chapter 2 begins with a review of current 2D orientation steerable filters. After pointing out their drawbacks from the point of view of uncertainty principle, we present a new 2D orientation steerable filter based on the angular Gaussian function. Then comes a theoretical analysis of the advantages of our filter. After that we further build a polar pyramid to treat the orientation scale variation. In the experiment section, we confirm the superior performance of our filter using both synthetic and real examples.

In chapter 3 we extend the new steerable filter from 2D space to 3D space, where only a few steerable filters are available hitherto. We confirm the advantage of our filter with the filter shape comparison and a simple example. For the sake of multiple motion estimation we discuss the filter responses of 3D planes in detail. After some application examples of 3D volume data processing, we analyze our filter in a broader background and develop a discussion of the relevance of our filter to 3D orientation histogram, surface tessellation, and spherical wavelets.

The main topic of chapter 4 is the application of 3D steerable filter in multiple motion analysis. At first, the state of the art of current research is provided in a review of multiple motion models and algorithms. We go on to analyze the spatial and spectral properties of both occlusion and transparency to obtain a deeper understanding of multiple motions. After the comparison between the spatial model and the spectral model we come to the conclusions that

- the spatial model can treat only occlusion, while the spectral model treats both occlusion and transparency;
- we prefer the spatial model for occlusion analysis because the spatial model has finer resolution and needs much less frames.

Then, we point out that our 3D filter outperforms existing 3D Hough transform and expectation maximization algorithm. Further, in occlusion analysis, a multi-window technique is introduced to improve the quality of input data for a better estimation. After obtaining motion parameters we use the "shift-and-subtract" technique instead of explicit boundary models to localize occlusion boundaries and to track their movement. In the experiment section we show that the "shift-and-subtract" technique has better performance than current boundary models and it can also be used to decompose transparency scenes into multi-layer representations.

The thesis is summarized in chapter 5.

Chapter 2

2D Orientation Steerability

Abstract

Junctions are significant features in images whose intensity variations exhibit multiple orientations. This makes the detection and characterization of junctions a challenging problem. In this chapter, we deal with the characterization of junctions which would ideally be the response of a filter at every orientation. This can be achieved by the principle of steerability that enables the decomposition of a filter into a linear combination of basis functions. However, current steerability approaches suffer from the consequences of the uncertainty principle: In order to achieve high resolution in orientation, they need a large number of basis filters, thus increasing the computational complexity. Furthermore, these functions usually have a wide support which accentuates the computational burden.

In this chapter, we propose a novel alternative to current steerability approaches. It is based on utilizing a set of polar separable filters with small support to sample orientation space locally. The orientation signature is then obtained by interpolating these samples using Gaussian functions. Compared with current steerability techniques, our approach achieves a higher orientation resolution with a lower complexity. In addition, we further build a polar pyramid to characterize junctions of arbitrary inherent orientation scales.

2.1 Introduction

Junctions of gray-value lines or edges carry important information for many image processing tasks like point matching in object recognition, point tracking in motion analysis, attentive coding, and line-drawing interpolation [Nal93]. In order to use junctions for such tasks, we must be able to localize their corresponding keypoints which are defined as intersection points of lines or edges. Then, we must characterize junctions by means of signatures and classify them in junction categories. Regarding keypoint detection and localization the reader is referred to [För94, Kov96, RO97, PGH98, FS00] and to the comparison of different operators by Rohr [Roh92, Roh97]. In this chapter, we address the problem of junction characterization. The resulting signature can be used for further junction classification.

Junctions are local structures with multiple intrinsic orientations and multiple spatial scales [BW93]. For the purpose of characterization we project them onto the orientation space and build a 1D signature function of the orientation parameter. Such signatures are often obtained by applying a set of filters at different orientations. This leads to an enormous computational load. For example, in order to extract orientation information of a junction, with conventional filter methods we have to rotate the same filter around the keypoint repeatedly. For an angular field of 360° and a sampling interval of 5° already 72 rotated copies of the original filter should be applied. The concept of steerability has been introduced in order to reduce this explosion of computational complexity. Steerable filters also provide an analytic model of deformations for further analysis of the grey-value structure [FA91, Bei94, GBG$^+$94, Per95, FP98, MPS98].

Denoting with $\alpha \in I\!R$ the deformation parameter, we define a filter $F(\mathbf{x})$ with $\mathbf{x} \in I\!R^n$ as a steerable filter if its deformed versions $F_\alpha(\mathbf{x})$ can be expressed as [MS95]

$$F_\alpha(\mathbf{x}) = \sum_{k=1}^{N} b_k(\alpha)\; A_k(\mathbf{x}), \tag{2.1}$$

where $A_k(\mathbf{x})$ and $b_k(\alpha)$ are referred to as basis filters and interpolation functions, respectively. For example, the deformed versions of the first derivative of 2D Gaussian function $g_1^\alpha(x,y)$ (figure 2.1) can be written as

$$g_1^\alpha(x,y) = cos(\alpha)\; g_1^{0^\circ}(x,y) + sin(\alpha)\; g_1^{90^\circ}(x,y), \tag{2.2}$$

with α denoting the angle between the axis through the filter lobes and x axis. Here $g_1^{0^\circ}(x, y)$ denotes the first derivative of Gaussian pointing to horizontal direction and $g_1^{90^\circ}(x, y)$ is a copy of $g_1^{0^\circ}(x, y)$ rotated by 90°.

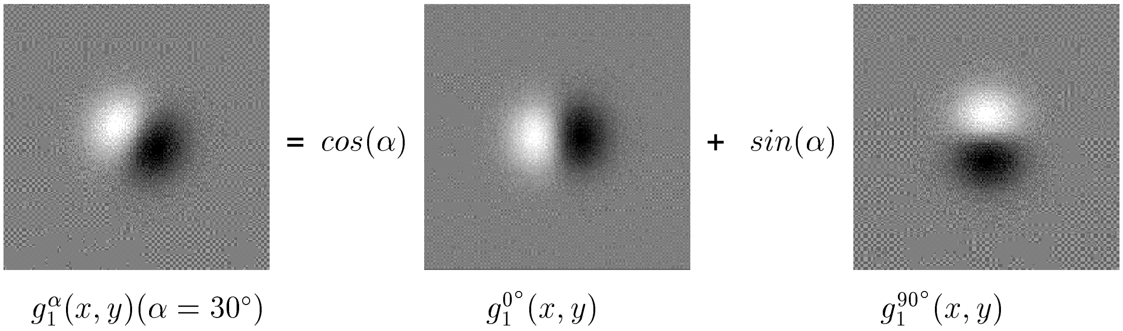


Figure 2.1: *One example of steerability using the first derivative of 2D Gaussian function. Here $g_1^{0^\circ}(x, y)$ denotes the first derivative of Gaussian pointing to horizontal direction and $g_1^{90^\circ}(x, y)$ is a copy of $g_1^{0^\circ}(x, y)$ rotated by 90°. (reproduced from [FA91]).*

The signature $s(\alpha)$ of a junction can be obtained by applying such a steerable filter on the image $I(\mathbf{x})$:

$$s(\alpha) := \langle F_\alpha(\mathbf{x}) | I(\mathbf{x}) \rangle = \sum_{k=1}^{N} \gamma_k b_k(\alpha) \tag{2.3}$$

with

$$\gamma_k := \langle A_k(\vec{x}) | I(\vec{x}) \rangle.$$

Here $\langle \cdot | \cdot \rangle$ denotes the usual inner product for two real functions $F(\mathbf{x})$ and $G(\mathbf{x})$:

$$\langle F(\mathbf{x}) | G(\mathbf{x}) \rangle \quad := \quad \int_{\mathbf{x}} F(\mathbf{x}) G(\mathbf{x}) \; d\mathbf{x}.$$

We see the motivation of steerability clearly in equation (2.3): The responses of the filters $F_\alpha(\mathbf{x})$ with $\alpha \in I\!R$ are expressed as a linear combination of N basis filter responses.

According to our opinion, steerability approaches may be classified in exact and approximate methods. Although steerability was implicitly used by Danielsson and Knutsson [Dan80, KG83], Freeman and Adelson [FA91] were the first who coined the concept and introduced an *exact* steerability approach. Freeman's and Adelson's approach to orientation steerability of a Gaussian derivative is an interpolation using harmonic functions in the orientation space given the fact that a filter is periodic with respect to orientation. The basis functions are rotated copies of the original filter. In [FA91] the derived kernels do not possess sufficient orientation resolution due to their large support in the orientation space (the orientation resolution of a filter can be measured by its angular support). Moreover, they are either symmetric or antisymmetric with respect to the mask center [SF96]. This results in a period of 180° in orientation and leads to an ambiguity in responses between terminating and non-terminating junctions [MS94].

Simoncelli et al. [SFAH92] extended this concept to include dilation and translation. They proposed some conditions upon which a filter is guaranteed to be exactly steerable, i.e. a filter can be synthesized with finite components from the Fourier basis. Recently, Simoncelli and Farid [SF96] designed a steerable wedge filter of which radial component looks like a wedge. Regarding the angular direction, they do not first choose one filter with expected shape and then project it onto the Fourier basis. Instead, they first choose finite components from the Fourier basis and then synthesize the filter *only* using these components. Therefore, their filter is guaranteed to be band limited and exactly steerable. Besides, the shape of the wedge filter can be adjusted to be arbitrarily narrow if they adopt adequate Fourier components. There is no more symmetric ambiguity applying a steerable wedge filter because the wedge kernel is asymmetric with respect to the mask center.

Michaelis and Sommer [MS95] and Teo and Hel Or [TO98, OT98, TO99] provided the formal justification of the exact steerability by applying Lie group theory. The basic concept of exact steerability is the shiftability [SFAH92]: Every periodic band-limited function can be approximated at every position (“shift”) with a finite linear combination of harmonic functions. In this sense, orientation and scaling become translations if we apply a logarithmic-polar transformation to a function with two arguments. Lie group theory gives us the theoretical framework for this transformation and for the exact interpolation. The complex harmonics $e^{j\omega k}$ are the generating operators of translation. On the other hand, all one-parameter Lie groups are locally isomorphic to the translation group if we change the coordinates to so called canonical coordinates, for example, Cartesian to polar coordinates for rotations. Thus, for every deformation we achieve exact steerability if we transform the coordinates to canonical ones and then apply Fourier analysis. Teo and Hel Or list a complete classification of functions steerable with respect to any Abelian group.

Perona [Per92] introduced the concept of deformable kernels which is based on the minimization of the discrepancy between the left-hand side and the right-hand side of equation (2.1) with respect to the basis functions $A_k(\mathbf{x})$. He showed that the basis functions are columns of the right orthonormal matrix of a continuous singular value decomposition (SVD). In case of rotation and periodic translation it is proved [Mic95] that the basis functions are the same in both the deformability and the exact Lie group based steerability. However, for other deformations the functions must be sampled with respect to the deformation parameter and a numerical SVD is applied [SMH98]. The advantage of the *deformability* approach is that it steers continuous as well as discrete filters and needs a minimal number of basis functions given a fixed error. However, orientation resolution is not addressed in this approach so that usually this approach suffers from the uncertainty constraint. Since we are interested only in orientation, both approaches yield the same set of basis functions. Therefore, from now on, we will use the term "exact steerability" for both approaches to orientation steerability.

The steerability problem may be also considered as a problem of signal reconstruction from samples $\gamma_k(k = 1, \cdots, N)$ (see equation (2.3)). This becomes evident if we consider the parameter α in equation (2.3) as the spatial variable. For clarity we change the notation to θ:

$$s(\theta) = \sum_{k=1}^{N} \gamma_k \; b_k(\theta). \tag{2.4}$$

Here the interpolation functions $b_k(\theta)$ may stem from many function classes, for example Laguerre functions or Legendre polynomials ([OS75], pp. 29-30). In current orientation steerability approaches $b_k(\theta)$ are usually the complex harmonics $e^{j\omega_k\theta}$, yielding

$$s(\theta) = \sum_{k=1}^{N} \gamma_k e^{j\omega_k\theta}. \tag{2.5}$$

Note that ω_k are not necessarily the first N frequencies as in the *standard* Fourier decompositions.

In this chapter, we will consider the following topics:

- We point out that the exact approach to orientation steerability has insufficient orientation resolution because it is based on the sampling of the angular frequency using Dirac functions. To achieve a high orientation resolution, a huge number of filters must be used. This computational burden is amplified by the large support of the basis functions.

- We will introduce a new approximation approach based on angular Gaussian functions. This approach might be non-optimal with respect to the approximation error, but substantially alleviates the above problems as shown in theory and experiments.

We extend this approach to an efficient hierarchical scheme and provide a complete analysis of computational complexity. Thus, by considering only the problem of orientation steerability, we sacrifice a coherent algebraic theory (as in Lie group and SVD-based deformability approaches) to achieve high orientation resolution as well as a dramatical decrease of the computational complexity.

This chapter is organized as follows: In section 2 we introduce the new steerability approach based on angular Gaussian functions and point out the theoretical difference between our approach and current steerability approaches. Besides, we analyze their computational complexity in detail. In section 3 we further introduce a polar pyramid scheme to treat orientation scale variations. Then we present experimental results on both synthetic and real data which vividly show the superior performance of our approach. This chapter is concluded with a short discussion.

2.2 2D Local Orientation Analysis

Definition of Our Steerable Filter

In the study of local orientation, we first conduct a local polar transformation of the image from Cartesian to the polar coordinate system and denote the new intensity function with $I(r, \theta)$, where r and θ are the radius and angle, respectively. Since we are merely interested in orientation, we eliminate the radial variable r by applying averaging along the radial direction. In order to obtain high orientation resolution, we are interested in filters with narrow angular support. We choose shifted Gaussian wedge functions as basis filters to sample the orientation space

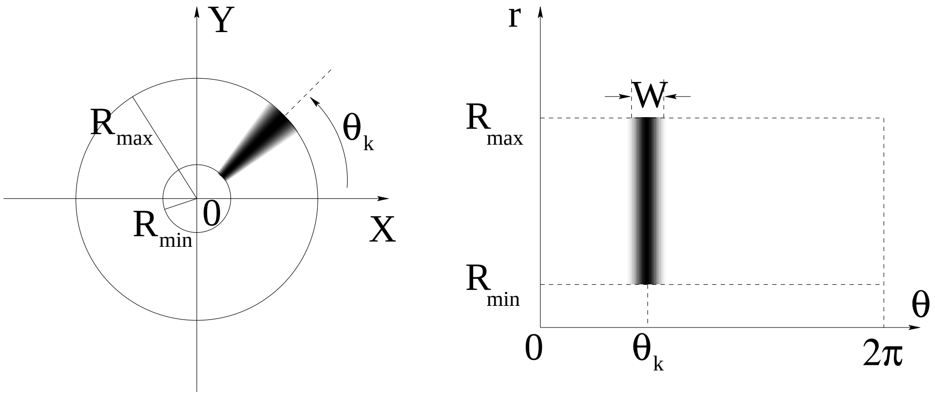


Figure 2.2: *A Gaussian wedge mask centered at angle θ_k. Darker pixels represent larger weights.* **Left:** *The mask in the Cartesian coordinate system. The keypoint is at the center of the circle.* **Right:** *The mask with r and θ as coordinates, $R_{\max}$ and $R_{\min}$ are radial boundaries of the mask, W is the angle width of the mask. We set $R_{\min} > 0$ to avoid the confusion close to the keypoint.*

locally

$$A_k := \frac{G_0(\mathcal{D}(\theta, \theta_k))}{\mathcal{N}(R_{\min}, R_{\max}, \theta_k)}, \tag{2.6}$$

with θ_k distributed evenly along the axis of the orientation variable θ. Here $\mathcal{N}(R_{\min}, R_{\max}, \theta_k)$ is an averaging factor along the radial direction which is the sum of discrete weights inside the basis filter mask centered at θ_k. We denote with $G_0(\mathcal{D}(\theta, \theta_k))$ the angular Gaussian function centered at θ_k

$$G_0(\mathcal{D}(\theta, \theta_k)) := \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mathcal{D}(\theta,\theta_k))^2}{2\sigma^2}}, \tag{2.7}$$

where σ denotes the scale of the Gaussian function. Since θ and θ_k are circular angles ($\theta, \theta_k \in [0, 2\pi]$), we define $\mathcal{D}(\cdot)$ to represent the minimal circular difference between θ and θ_k

$$\mathcal{D}(\theta, \theta_k) := min(|\theta - \theta_k|, |\theta - \theta_k - 2\pi|, |\theta - \theta_k + 2\pi|). \tag{2.8}$$

For example, $\mathcal{D}(2\pi, 0) = 0$; $\mathcal{D}((\frac{359\pi}{180}), \frac{\pi}{180}) = \frac{\pi}{90}$. Theoretically, a Gaussian function is not compactly supported. Thus, in implementation we only consider the part of $G_0(\mathcal{D}(\theta, \theta_k))$ whose variable varies from $\theta_k - \frac{W}{2}$ to $\theta_k + \frac{W}{2}$ (see equation (2.12)). Here W denotes the angular width of the basis filter. We will explain the choice of this parameter in section 3. In figure 2.2 we show a basis filter centered at θ_k, where $R_{\min}$ and $R_{\max}$ denote inner and outer boundary of the mask, respectively. We set $R_{\min} > 0$ to avoid the singularity close to the keypoint [MS94]. In order to choose $R_{\max}$ we must know the size of the significant neighborhood around the keypoint, which can be provided by the preceding keypoint detection step [För94, RO97]. In this chapter, we set $R_{\min} = 3$ pixels and $R_{\max}$ varies from 9 to 15 pixels.

After defining basis filters, we must interpolate them to build the whole set of steerable filter $g(\theta)$. Taking the local property of basis filters into account, we choose angular Gaussian function with narrow support instead of complex harmonics as the interpolation function

$$b_k(\theta) := G_0(\mathcal{D}(\theta, \theta_k)). \tag{2.9}$$

Thus, we construct a *continuous* steerable filter using Gaussian interpolation functions

$$g(\theta) := \sum_{k=1}^{M} A_k \; G_0(\mathcal{D}(\theta, \theta_k)). \tag{2.10}$$

In this new steerability approach, the neighboring basis filters as well as the interpolation functions are locally correlated and therefore non-orthogonal. To achieve *optimal* steerability we would have to apply a non-linear operator to estimate the coefficients of the interpolation function. Instead we use a linear operator and achieve an *approximate steerability* which approaches optimality with decreasing support and thus increasing orthogonality of the Gaussians. Approximating a function with a sum of Gaussians is a well-known method with properties extensively described in the Radial Basis Function approximation proposed by Poggio and Girosi [PG90]. If we ignore the penalty term in [PG90] enforcing smoothness our approach becomes the closer to the RBF approach when our Gaussians overlap minimally each other. To increase approximation optimality, the centers of the Gaussians can be estimated in [PG90], whereas in our approach they overlap with the pre-chosen samples of the unknown signature.

Applying this new steerable filter on the intensity function $I(r, \theta)$, we obtain the orientation signature $S(\theta)$

$$S(\theta) := \sum_{k=1}^{M} \gamma_k \; G_0(\mathcal{D}(\theta, \theta_k)), \tag{2.11}$$

where

$$\gamma_k := \langle A_k | I(r,\theta) \rangle = \sum_{\theta=\theta_k-\frac{W}{2}}^{\theta_k+\frac{W}{2}} G_0(\mathcal{D}(\theta,\theta_k)) \sum_{r=R_{\min}}^{R_{\max}} \frac{I(r,\theta)}{\mathcal{N}(R_{\min},R_{\max},\theta_k)}. \tag{2.12}$$

In the orientation signature $S(\theta)$ local extremes represent orientations of lines and the positions of steepest descent or ascent indicate orientations of edges. Correspondingly, the 2D orientation analysis reduces into 1D line/edge detection. In order to extract edge information, we estimate the derivative of $S(\theta)$ and consider the amplitude of the derivative as another signature

$$\begin{aligned} DS(\theta) &:= \mid \frac{d}{d\theta} S(\theta) \mid \\ &= \mid \sum_{k=1}^{M} \gamma_k \frac{d}{d\theta} G_0(\mathcal{D}(\theta,\theta_k)) \mid \\ &= \mid \sum_{k=1}^{M} \gamma_k \, G_1(\mathcal{D}(\theta,\theta_k)) \mid, \end{aligned} \tag{2.13}$$

where $G_1(\mathcal{D}(\theta,\theta_k))$ denotes the first derivative of Gaussian filter G_0 and $\mid \cdot \mid$ denotes the absolute value. Here, we are only interested in the amplitude of the derivative. We point out here that $S(\theta)$ and $DS(\theta)$ cannot constitute a quadrature pair because $DS(\theta)$ is not the Hilbert transform of $S(\theta)$. Such a pair would have significant energy on the negative frequencies and the estimated magnitude would be phase-dependent. To characterize junctions, we are not interested in a phase-independent magnitude response because we want to know which orientation responses are closer to an edge and which are closer to line. Therefore, we obtain two separate signatures from $S(\theta)$ and $DS(\theta)$, for lines and edges, respectively.

So far, we have defined the approximate steerability. In the following, we will compare this approach with current steerability approaches regarding the mathematical background and the implementation performance.

Difference to Exact Steerability

The main difference between our approach and the exact steerability is that we decompose one signal *locally* in the spatial domain, whereas exact approaches decompose the signal *globally*. This difference lies in both the basis filters and the interpolation functions: First, in order to calculate the response of a basis filter (see equation (2.5) and equation (2.12)), we need a Gaussian mask with local spatial support in the approximate steerability, while in the exact steerability every basis filter has the same wide spatial support as the steered filter (as shown in figure 2.3). Second, in order to obtain the signature values at an arbitrarily fixed orientation, we only need to interpolate several neighboring basis filter responses with Gaussian functions in the approximate steerability (equation (2.11)), while in the exact steerability we must interpolate *all* basis filter responses with the complex harmonics functions (equation (2.5)).

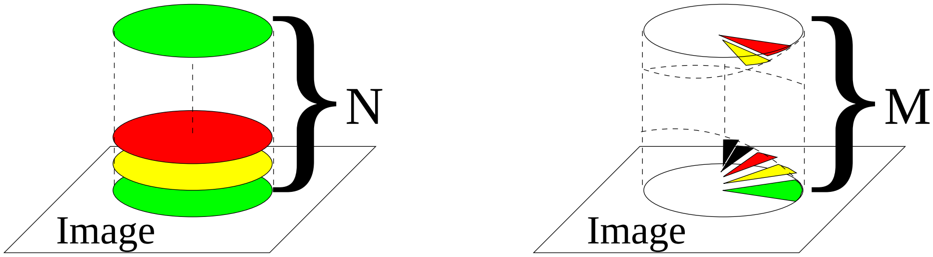


Figure 2.3: *Basis filter comparison.* **Left:** *N Basis filters of current steerable filters. Each basis filter has the same large spatial support as the steered filter.* **Right:** *Our basis filters. Their spatial support is much smaller.*

This difference can be described in the spectral domain as well. In the approximate steerability (equation (2.11)), the spectrum of a filter response is decomposed into a set of *spectral* Gabor functions (shifted Gaussian functions in the spatial domain) weighted by basis filter responses γ_k

$$\mathcal{F}[S(\theta)] := \sum_{k=1}^{M} \gamma_k \ \mathcal{F}[G_0(\theta - \theta_k)] = \sum_{k=1}^{M} \gamma_k \ \mathcal{F}[G_0(\theta)] e^{-j\omega\theta_k}, \tag{2.14}$$

where $\mathcal{F}$ denotes the Fourier transform. In contrast, in the exact steerability, the spectrum of a filter response (equation (2.5)) is decomposed into a series of Dirac sampling functions weighted by the corresponding basis filter responses

$$\mathcal{F}[s(\theta)] = \sum_{k=1}^{N} \gamma_k \ \delta(\omega - \omega_k). \tag{2.15}$$

This formula describes also the behavior of the exact approaches using rotated filter copies as basis functions. The proof of the steerability using rotated copies is based on the fact that a function is written as a Fourier series with respect to the angle (see equation (9) in [FA91]). Equation (2.15) is the expression of this fact in the frequency domain. Note that in equation (2.15) the term $a_n(r)$ in [FA91] is replaced by γ_k since we consider the filter response here.

Thus, in the case of approximate steerability we sample the signal in the spatial domain with Gaussian masks, whereas in the exact steerability the signal is sampled in the spectral domain by Dirac sampling functions. This difference makes our approach perform better with respect to orientation resolution. According to the well known uncertainty principle, we cannot simultaneously localize one signal both in the spatial domain and in the spectral domain exactly. If we use one Dirac sampling function to localize one spectral component of the signal exactly, as in the case of the exact orientation steerability, we will no more be able to localize this component in the spatial domain. Therefore, we need many Dirac impulses to increase the localization capability in space. This trade-off can be optimized by applying functions with Gaussian shape [Dau85]. Therefore, the approximate steerability has better properties with respect to the uncertainty principle.

To summarize, our main concern is high orientation resolution with low complexity. To achieve this goal, we directly built our filter in the spatial domain. The price we pay is that we do not achieve exact steerability but an approximation of the orientation response.

Complexity Analysis

Approximate steerability achieves a higher orientation resolution with a lower complexity due to the narrower angular support of the basis filters and the local interpolation functions. Our approach starts with a local polar mapping which can be done “off-line” since it is a transform between coordinates and is therefore valid for *all* different images. Online applying the resulting look-up-table (LUT) is of negligible complexity compared with calculating the filter responses.

In order to compare the implementation complexity of the approximate steerability and that of the exact approach we set the radial extensions of masks in both schemes to be the same. If we denote with P the 1D grid size of an exactly steerable filter, the following relation is satisfied: $P = 2\,R_{\max} + 1$. This relation is displayed in figure 2.4 as well.

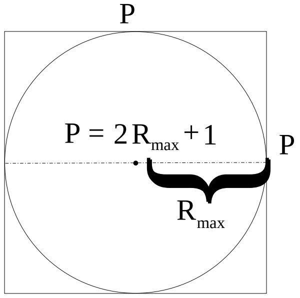


Figure 2.4: *In order to have the same radial extension of filter masks in exact steerability and in our approach, we have the relation $P = 2\ R_{\max} + 1$, where P is the 1D grid size of an exactly steerable filter and $R_{\max}$ is the outer boundary of our steerable filter.*

Here we choose the steerable wedge filter [SF96] for comparison since its shape is similar to the shape of our filter. The steerable wedge filter is a polar separable filter. Its radial component looks like a wedge and its angular component is synthesized by the Fourier series. Its computational complexity is proportional to the number and the spatial support of basis filters. In order to make a fair comparison we apply it after the local polar mapping, too.

It should be noticed that there is an essential difference between scalar product and convolution. In calculating scalar product we do not shift the filter mask, while in calculating convolution we shift the filter mask and calculate the corresponding scalar product repeatedly. Accordingly, in calculating convolution we can shift a separable filter mask *separately* and use two successive 1D convolutions to replace the 2D convolution. Then we need not to calculate the *local* 2D scalar products at every position in the image and can therefore benefit from the separability of the filter mask. Actually, our filter is also polar separable like the steerable wedge filter. Therefore, both filters have the advantage mentioned above in case of convolution. Of course we must also consider the extra computation cost of undertaking Cartesian to polar and polar to Cartesian coordinate transformation if we would like to use the polar separability.

In this paper, we only use the scalar product and the separability is no more an advantage. In calculating the scalar product between an image region and a filter mask, *every* pixel inside the image region should be weighted with the corresponding element in the filter mask *once and only once*. Thus, for a $P \times P$ image region and a $P \times P$ filter mask we need P^2 multiplications and additions whether or not this filter mask is separable.

Suppose we apply a steerable wedge filter composed of $2N$ basis filters (N odd basis filters and N even basis filters), as mentioned above, all basis filters have the same spatial support as the steered filter. Therefore, the 1D size of all basis filters is P, too. In order to apply one basis filter, we need P^2 multiplications and additions to calculate the corresponding scalar product. Straightforwardly, $2NP^2$ multiplications and additions are required to obtain $2N$ coefficients (see equation (2.3)). Assuming that we use signatures of length L, we need $2NL$ multiplications and additions to obtain outputs of odd and even filters. Thus, totally $2N(P^2 + L)$ multiplications and additions are computed to implement a steerable wedge filter composed of N odd basis filters and N even basis filters. Here, for the sake of a fair comparison, we do not consider the computational load to form the energy output because in our approach we do not use the energy output.

In our approach, the computational load is determined by the number and the angular width of Gaussian functions. In order to sample the whole orientation space with a sampling interval $\delta\theta$, we need totally $M = \frac{L}{\delta\theta}$ basis filters. Note L and $\delta\theta$ should have the same length unit. Here we use degree as the unit. According to the well known Shannon's Sampling Theorem, we determine the corresponding Nyquist frequency f_m with

$$f_m = \frac{1}{2\,\delta\theta}. \tag{2.16}$$

In our approach, we choose Gaussian masks instead of Dirac functions as sampling masks along the angular direction. This is equivalent to Dirac series (the shah function III(θ) [Bra86]) convolved with a Gaussian function. Correspondingly, the spectrum of the ideal sampling will be further multiplied by a low pass filter with Gaussian shape. The stop frequency of this low pass filter is determined by the term $\frac{1}{2\sigma}$ (here we define the turning frequency of the Gaussian function $\frac{1}{2\,\sigma}$ as the stop frequency). This stop frequency is preferred to be not below the Nyquist frequency f_m

$$\frac{1}{2\,\sigma} \geq f_m = \frac{1}{2\,\delta\theta}, \quad \text{hence} \quad \sigma \leq \delta\theta. \tag{2.17}$$

On one hand, we should set the sampling interval $\delta\theta$ small in order to have high Nyquist frequency. In this chapter we use $\delta\theta = 1^\circ$. On the other hand, the filter mask should contain adequate pixels to calculate averaging values robustly. Therefore, we set $\sigma = \delta\theta$ to achieve a compromise between these two contradictory requirements. Theoretically, a Gaussian function is not compactly supported.

Thus, we must cut off its support to build an FIR-filter. It is easy to show that, in order to keep the energy of the cut-off area below 1% of the total energy, the angular width of the sampling mask W must be at least $5\,\sigma$. In this chapter, we set $W = 6\,\sigma$.

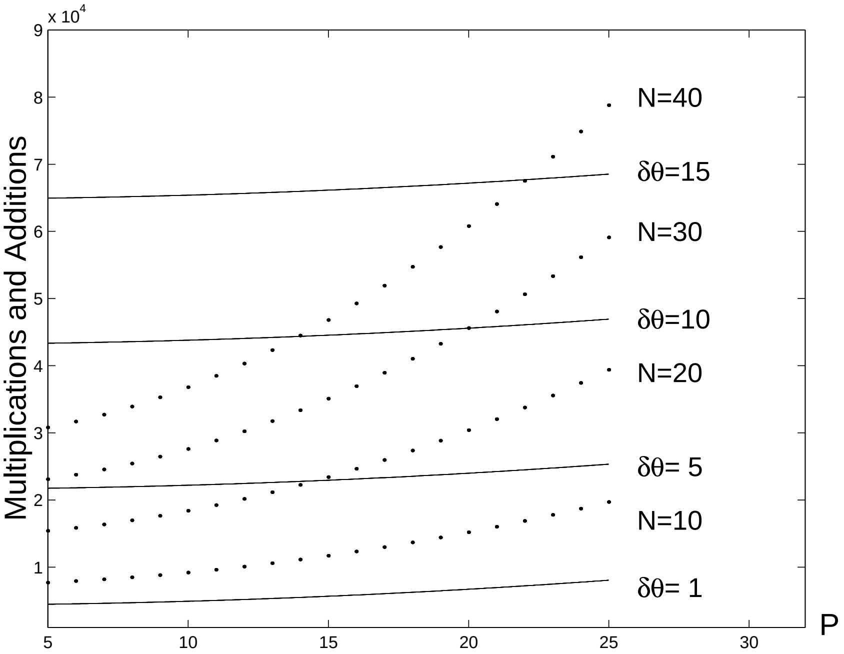


Figure 2.5: *Complexity comparison between the exact steerability and the approximate steerability. We set the length of the signature as $L = 360$. The dotted curves represent the complexity of exactly steerable filters composed of N odd basis filters and N even basis filters with P as 1D grid size. The solid lines show the relation between complexity and the sampling interval $\delta\theta$ and P in computation of $S(\theta)$ and $DS(\theta)$ in our approach. We see that the approximate steerability is more efficient than the exact steerability, especially in the case of high orientation resolution and large filter size.*

We use $M = \frac{L}{\delta\theta}$ sampling masks to obtain angular samples A_k $(k = 1, \cdots, M)$. On an average we have at most $\frac{W}{360}\,\pi\,(R^2_{\max} - R^2_{\min}) < \frac{W}{360}P^2$ pixels in one sampling mask. In order to analyze the upper limit of the complexity, we assume that in every sampling mask there are $\frac{W}{360}P^2$ pixels. Correspondingly, we need $\frac{W}{360}P^2$ multiplications and additions to calculate one angular sampling. Taking into account that $W = 6\,\delta\theta$, we need totally $M\frac{W}{360}P^2 = \frac{6L}{360}P^2$ multiplications and additions to obtain M samples. In constructing $S(\theta)$ or $DS(\theta)$, we apply a Gaussian function and its first derivative, respectively, with the angular width W as interpolation functions. In order to produce the signatures with the same length

L as in the above steerable wedge filter approach, we require LW multiplications and additions to build each one of both signatures $S(\theta)$ and $DS(\theta)$. Thus, totally we need $\frac{6L}{360}P^2 + 2\,L\,W = 6L\,(\frac{P^2}{360} + 2\,\delta\theta)$ multiplications and additions to obtain $S(\theta)$ and $DS(\theta)$ in the case of approximate steerability.

In figure 2.5 we plot an example of the complexity comparison of both approaches by setting $L = 360$. In the approximate steerability the parameter $\delta\theta$ plays a very important role. Here, $\delta\theta$ denotes not only the sampling interval, but also the angular support of basis filters because we set $\sigma = \delta\theta$ and $W = 6\,\sigma$. In order to achieve higher orientation resolution, we set $\delta\theta$ smaller so that the basis filters have narrower angular support. It is interesting to observe that the computational load using basis filters with narrower angular support is even lower than that using basis filters with broader angular support. This is precisely contrary to the tendency in the exact steerability, where the computational load increases when we use more basis filters to achieve higher orientation resolution. This sharp contrast demonstrates the merit of the approximate steerability vividly.

We also observe that, in the approximate steerability, the increase of the computational load with respect to the filter size P is not as strong as that in the exact steerability. By watching the corresponding computational load more closely, we find out that, though the computational load in both steerability approaches is proportional to P^2, the coefficient of P^2 in the approximate steerability is much less than the coefficient in the exact steerability. Thus, if the filter mask is large, the approximate steerability is even more efficient than the exact steerability.

2.3 Junction Characterization Using Polar Pyramid

The orientation scale problem is like every scale problem a trade-off between the intrinsic structure of a junction and the orientation scale of a filter. As shown in figure 2.6, if the orientation scale of a filter is too small, a blurred edge is not visible and a wide line will be recognized as two edges [MS94]. If the orientation scale of a filter is too large, two very close lines will be characterized as one line. In this section, we introduce a polar pyramid to obtain signatures of different scales efficiently. Here we do not treat the problem of steering spatial scale. Regarding the steerability of spatial scale the reader is referred to [Per95, Mic95].

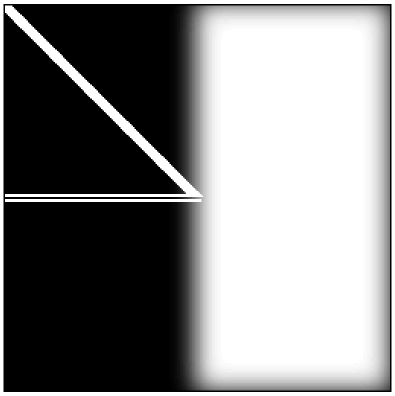


Figure 2.6: *A junction composed of two close lines with distance of 1 pixel, two edges blurred by a Gaussian function with $\sigma = 4$ and a wide line with a width of 6 pixels.*

It is known that one of the most appealing kernels for hierarchical approaches is the Gaussian function [BWBD86]. We know from spatial scale theory that the choice of scale necessitates a hierarchical treatment. Burt and Adelson [BA83] proved that a generating kernel of subsampling can be used as the interpolation function for reconstruction from coarser scales. Moreover, they argued that the interpolation functions can be (discrete approximations of) Gaussian functions with different scales (figure 2.7). Thus, the continuous orientation information can be reconstructed from all levels of the polar pyramid by interpolating the samples $\hat{S}^j(\theta_k)$ with Gaussian functions $G_0^j(\mathcal{D}(\theta, \theta_k))$ of different scales σ_j

$$S^j(\theta) = \sum_k \hat{S}^j(\theta_k)\; G_0^j(\mathcal{D}(\theta, \theta_k)), \qquad j \in [1, 2, \cdots] \tag{2.18}$$

with

$$G_0^j(\mathcal{D}(\theta, \theta_k)) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(\mathcal{D}(\theta,\theta_k))^2}{2\sigma_j^2}}, \qquad j \in [1, 2, \cdots], \tag{2.19}$$

where $\hat{S}^j(\theta_k)$ denote the samples on the j-th level of the polar pyramid and $S^j(\theta)$ represents the signature reconstructed from $\hat{S}^j(\theta_k)$. We can obtain edge signatures similarly with

$$DS^j(\theta) = \mid \sum_k \hat{S}^j(\theta_k)\; G_1^j(\mathcal{D}(\theta, \theta_k)) \mid, \qquad j \in [1, 2, \cdots], \tag{2.20}$$

where $G_1^j(\mathcal{D}(\theta, \theta_k))$ is the first derivative of $G_0^j(\mathcal{D}(\theta, \theta_k))$. The local maxima in $S^j(\theta)$ and $DS^j(\theta)$ denote orientations of lines and edges at different scales, respectively.

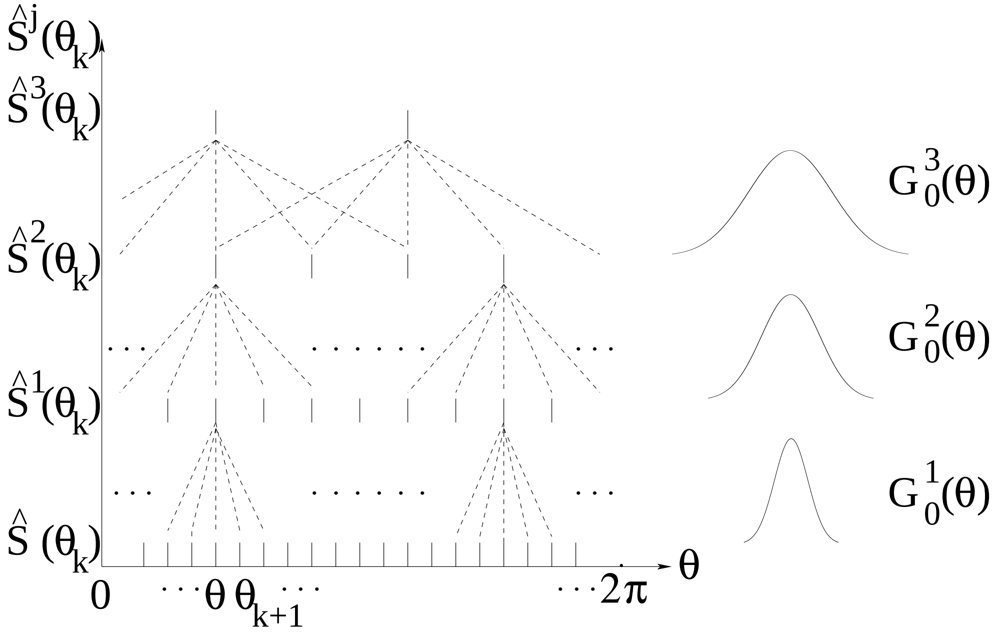


Figure 2.7: **Left:** *Polar pyramid structure, $\hat{S}(\theta_k)$ are samples on the lowest level of the polar pyramid, namely the outputs of Gaussian basis filters. $\hat{S}^j(\theta_k)(j = 1, 2, 3)$ are samples on the higher levels of the pyramid after subsampling with the generating kernel K_2 of equation (2.23).* **Right:** *Corresponding interpolation functions at different levels. They are Gaussian functions with different scales.*

If we want to build a pure 1D one-octave Gaussian pyramid of angles, according to [BA83] we should have $Q2^J + 1$ samples, where J is the number of levels and $Q + 1$ is the number of samples at the highest level. Taking the periodicity into account, we should have $Q2^J$ samples as the sampling outputs. However, since initially the orientation signal is defined as 360 discrete values, we cannot build a pure octave Gaussian pyramid. Alternatively, we apply a factor 2 subsampling of the first three levels, a factor 3 at the next two, and have 5 samples at the coarsest level $(360 = 2^3 \times 3^2 \times 5)$. Let us denote with $K_i(n)$ the n-th coefficient of the FIR filter in the i-th pyramid layer. According to [BA83] the generating kernels with subsampling factor i should be normalized

$$\sum_{n=1}^{2i+1} K_i(n) = 1, \qquad i \in [2, \cdots] \tag{2.21}$$

and symmetric

$$K_i(n) = K_i(2i + 1 - (n - 1)), \qquad n \in [1, \cdots, i + 1]. \tag{2.22}$$

Moreover, every sample at a given level should make equal contributions to construct the next higher level. The contributions of one sample are weighted by the corresponding coefficients of the generating kernels. Equal contributions imply that the sum of all its connected elements in the generating kernels should be a constant.

As mentioned in [BA83], the generating kernels satisfying the above constraints should have Gaussian shape. We use the following discrete approximations of Gaussian functions to serve as generating kernels with subsampling factor 2, 3 and 5

$$K_2 = \frac{1}{16} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \end{pmatrix}, \tag{2.23}$$

$$K_3 = \frac{1}{264} \begin{pmatrix} 3 & 22 & 66 & 82 & 66 & 22 & 3 \end{pmatrix}, \tag{2.24}$$

$$K_5 = \frac{1}{5120} \begin{pmatrix} 1 & 74 & 299 & 725 & 950 & 1022 & 950 & 725 & 299 & 74 & 1 \end{pmatrix}. \tag{2.25}$$

2.4 Experiments

Synthetic Junction Examples

In this section, we illustrate some examples of junction characterization using the new steerability and its hierarchical version. In figure 2.8 and figure 2.9 synthetic line junctions and edge junctions are shown. The corresponding signatures $S(\theta)$ and $DS(\theta)$ characterize them correctly. The small deviations in figure 2.9 come from the fact that an edge can only be represented by two pixels in the grid, while we cannot set the center of a filter mask between two pixels.

The robustness of our method against noise is shown in figure 2.10. The edge junction is disturbed with increasing random noise. Even in the very noisy case the junction is well characterized. The keypoints in figure 2.11 are deviated from the central positions of the masks. Though the signatures have some variations, we can still characterize the junctions.

In figure 2.12 we compare the performance of both steerability approaches. A complex junction called "Siemens star" with 16 edges spans the orientation space

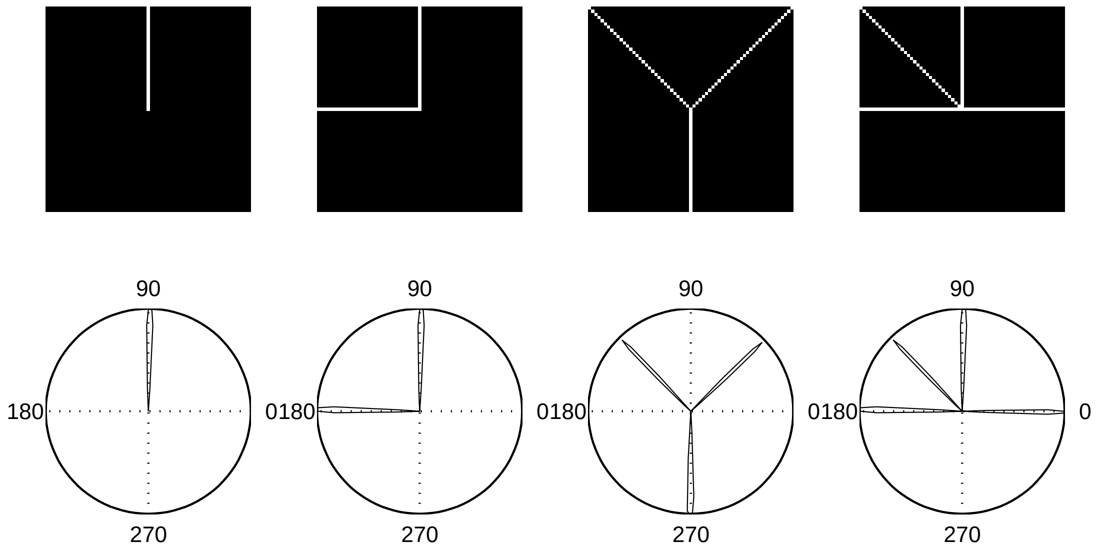


Figure 2.8: **Top:** *Different synthetic line junctions including a terminating line as well as 'L', 'Y', and 'K' junction.* **Bottom:** *Corresponding orientation signatures $S(\theta)$. We use the following parameters: $R_{\min} = 3$, $R_{\max} = 15$.*

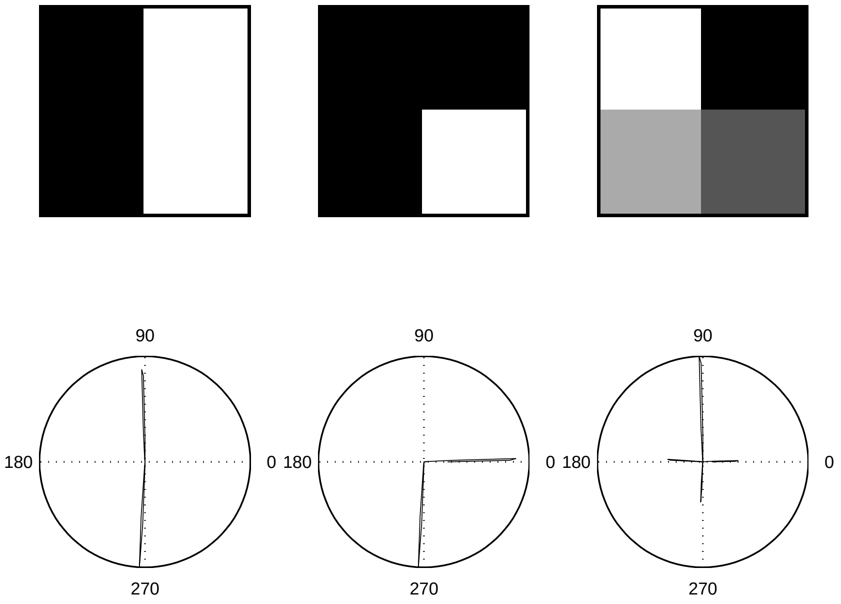


Figure 2.9: **Top:** *Synthetic edge junctions.* **Bottom:** *$DS(\theta)$. The local maxima show the orientation of edges. $R_{\min} = 3$, $R_{\max} = 15$.*

uniformly. Applying the steerable wedge filter, we even have to use 90 basis filters to achieve the same orientation resolution as applying our approximately steerable filter. Consequently, we need about eleven times as many multiplica-

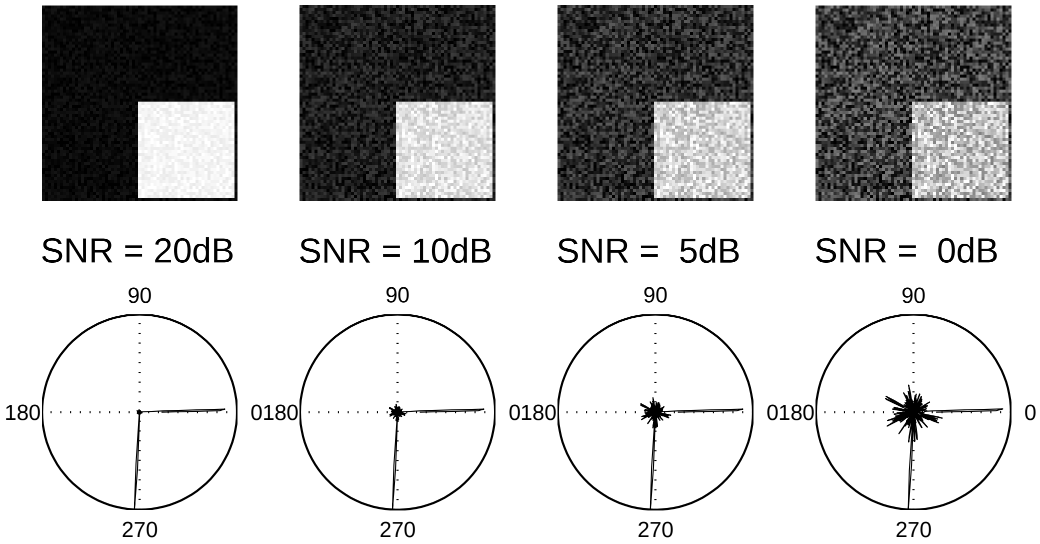


Figure 2.10: **Top:** *A synthetic edge junction disturbed by four increasing levels of random noise.* **Bottom:** *Corresponding $DS(\theta)$. Even in the very noisy case ($SNR = 0dB$) the signature can characterize the junction. $R_{\min} = 3$, $R_{\max} = 15$.*

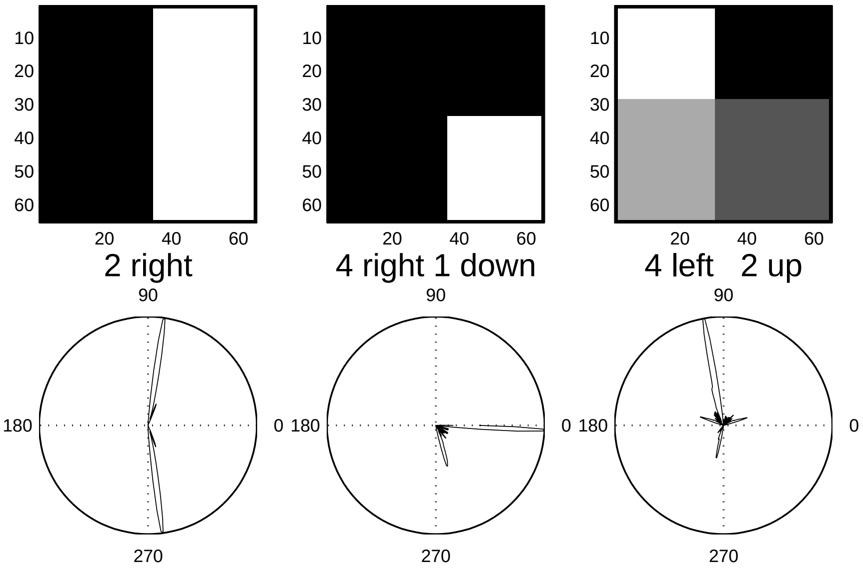


Figure 2.11: **Top:** *Deviation of the keypoint from the central position of the mask.* **Bottom:** $DS(\theta)$. $R_{\min} = 3, R_{\max} = 15$.

tions and additions using the steerable wedge filter as using our steerable filter. This demonstrates the advantages of the local decomposition scheme in the approximate steerability.

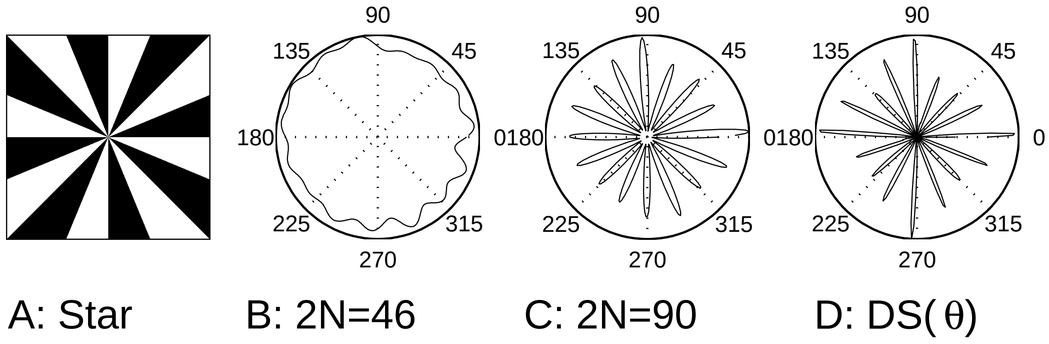


Figure 2.12: **A:** *A "Siemens star" with 16 edges spanning uniformly the orientation space.* **B:** *Polar plot of the result using the steerable wedge filter [SF96] composed of 46 basis filters with 31 grid size. Totally* 60766 *multiplications and additions are needed. The edges are hardly discernible.* **C:** *The same as in* **B** *but using 90 basis filters with* 118890 *multiplications and additions. The orientations of the edges are clearly presented.* **D:** $DS(\theta)$ *using the Gaussian averaging steerable filter. We compute only* 10086 *multiplications and additions to achieve the same resolution.*

Figure 2.13 is an example to solve the orientation scale problem applying a polar pyramid. The junction is composed of two blurred edges, a wide line and two close lines (see figure 2.6 for details). We use a polar pyramid with four levels to characterize it. With the increase of the pyramid level, the orientation of the wide line and blurred edges are characterized more and more distinctly. However, in the meantime, the responses of two close lines seem more and more like one line. This is exactly the demonstration of orientation scale problem.

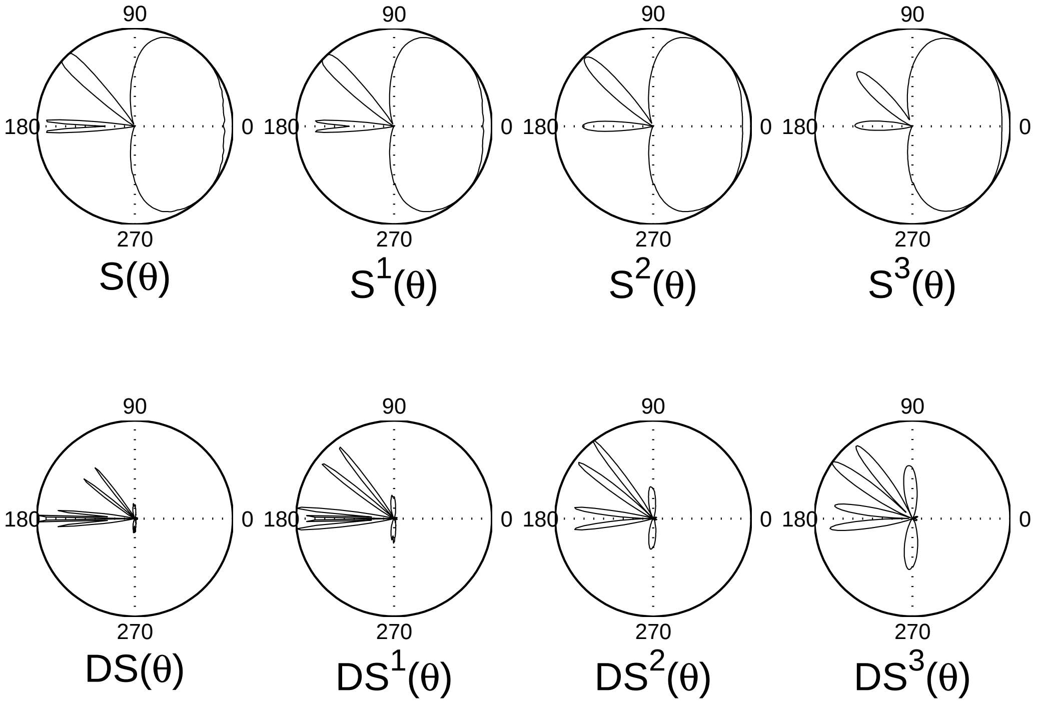


Figure 2.13: *Characterizing the junction shown in figure 2.6 using the polar pyramid.* **Top:** *The characterizing signatures reconstructed from the first four levels of the polar pyramid: $S(\theta)$, $S^1(\theta)$, $S^2(\theta)$, and $S^3(\theta)$. The wide line is distinctly characterized in $S^3(\theta)$. But two close lines are also recognized as one line.* **Bottom:** *Corresponding $DS(\theta)$, $DS^1(\theta)$, $DS^2(\theta)$, and $DS^3(\theta)$. The blurred edges are presented more and more clearly with the increase of pyramid level. The neighboring boundaries of two close lines near 180° can be seen only at the first two levels. At the third and the fourth scale levels, only outer boundaries of these two lines are recognized.*

Real Examples

We have observed the successful behavior of the approximate steerability on synthetic junction characterizations. Here we show the results of some real examples. One example is the parkbench picture used in [SF96] (figure 2.14). In comparison to the steerable wedge filter [SF96] with 30 basis filters, our filter characterizes the directions of junctions more distinctively. This is explicitly presented by the 'T' junction in **D**, where the blurred edge near 180° is better characterized with the approximately steerable filter. We also see that our approach is relatively more sensitive to high frequency components due to the differentiation.

Another real example is presented in figure 2.15. The kernel centers are displaced from the keypoints of the junctions. The results show that both steerable wedge filter and approximately steerable filter are stable with respect to the offsets of keypoints, while our approach achieves higher orientation resolution with lower cost.

In figure 2.16 we show the high orientation resolution of the approximate steerability. While the steerable wedge filter [SF96] with 90 basis filters only detects the dominant dark line between the lips of Lena, the approximately steerable filter characterizes edges of two lips distinctly. This may be very useful in facial feature analysis.

Our filter can be used for facial paresis diagnosis as well. It is known that most facial pareses occur on one side of faces. Therefore, we may use the orientation signatures on both sides of faces for symmetry analysis. In figure 2.17 we display such an example of facial feature symmetry analysis. [*]

A real example with varying scales is further presented in figure 2.18. The left corner of a child's right eye can be regarded as a combination of irregular wide lines and blurred edges disturbed by noise. The characterizing results from different pyramid levels form a complete set of signatures providing information at different orientation scales.

[*] A. Gebhard from University Erlangen-Nuremburg invoked and implemented this application and kindly provided these images.

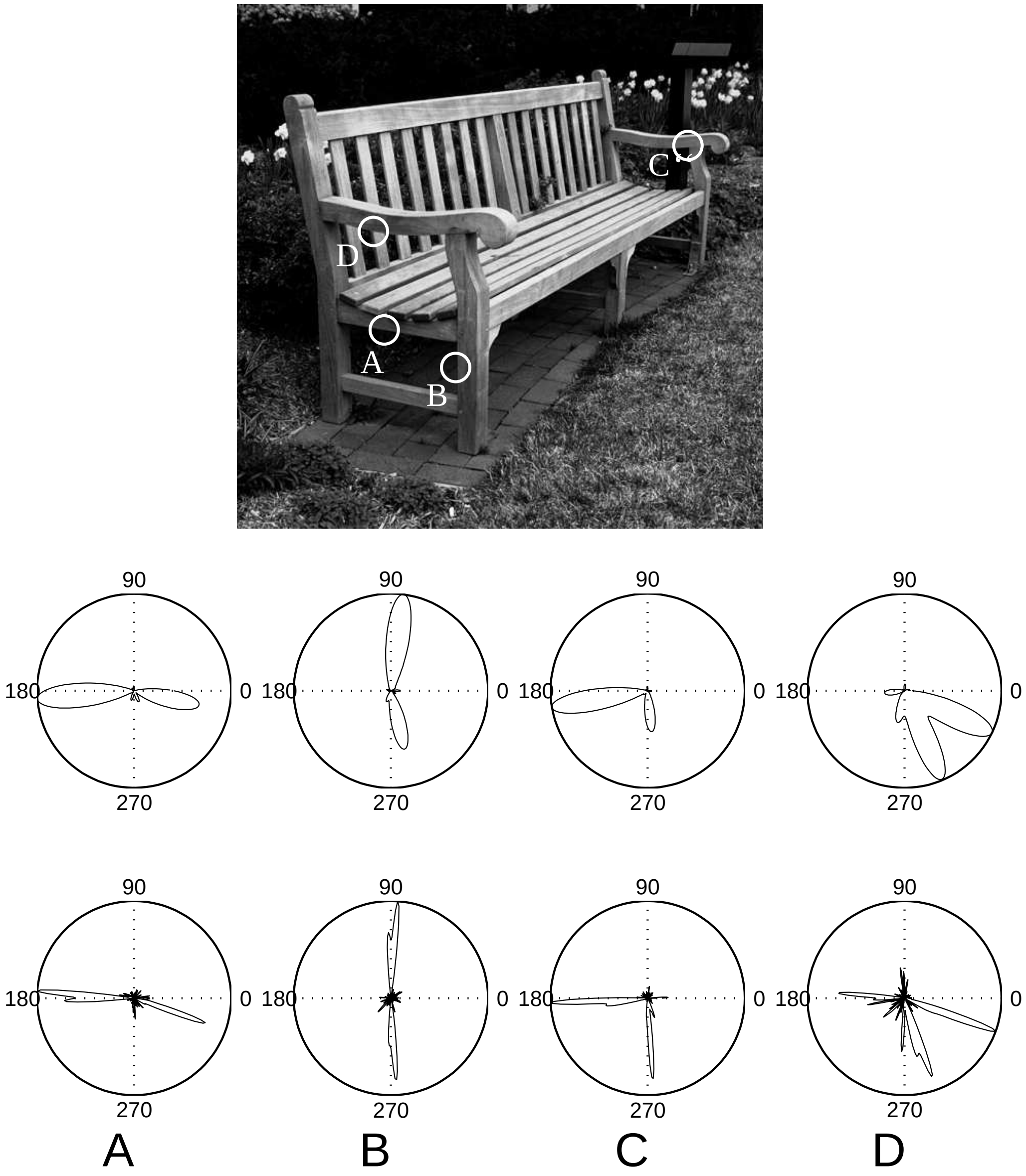


Figure 2.14: **Top:** *The parkbench with marked edge junctions.* **A***: horizontal edge;* **B***: vertical edge;* **C***: corner;* **D***: 'T'-junction.* **Middle:** *Steerable wedge filter results using* 30 *basis filters.* $P = 19$. **Bottom:** $DS(\theta)$ *of our approximately steerable filter. The edge near* 180° *in* **D** *is very blurred. But* $DS(\theta)$ *still can characterize it.* $R_{\min} = 3, R_{\max} = 9$.

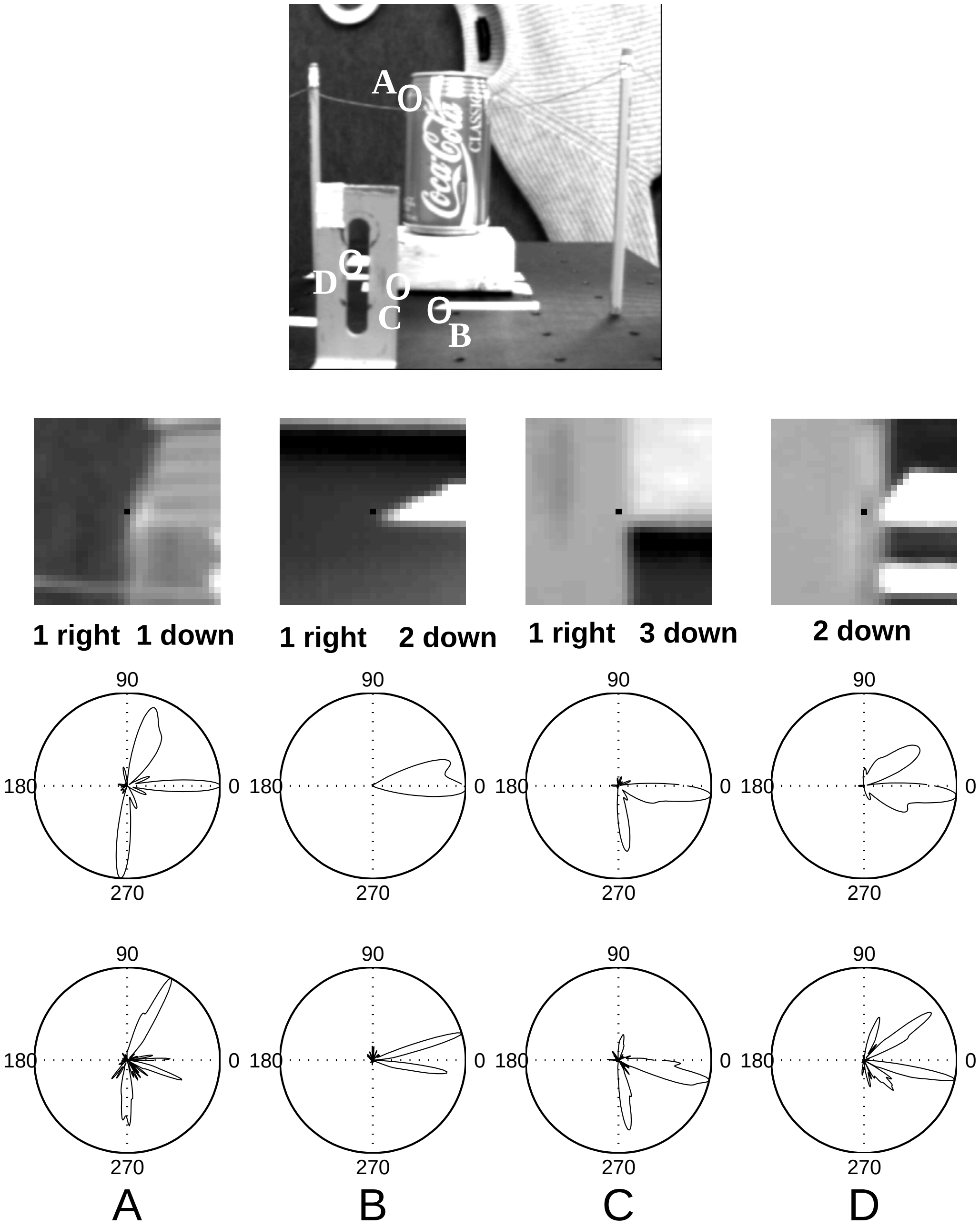


Figure 2.15: *Comparison between the steerable wedge filter [SF96] and approximately steerable filter.* **Row 1:** *An image of the NASA sequence with four kinds of marked junctions.* **A**: *'Y' junction;* **B**: *'V' junction;* **C**: *'T' junction;* **D**: *'K' junction.* **Row 2:** *Junctions in detail. We show centers of the masks with dark points. All keypoints deviate from centers of the masks.* **Row 3:** *Polar plots using the steerable wedge filter [SF96] composed of 46 basis filters with 31 as grid size.* **Row 4:** $DS(\theta)$ *using our approximately steerable filter (*$R_{\min} = 3, R_{\max} = 15$*). Both methods are stable with respect to the offsets of keypoints. Our method presents higher orientation resolution with lower cost.*

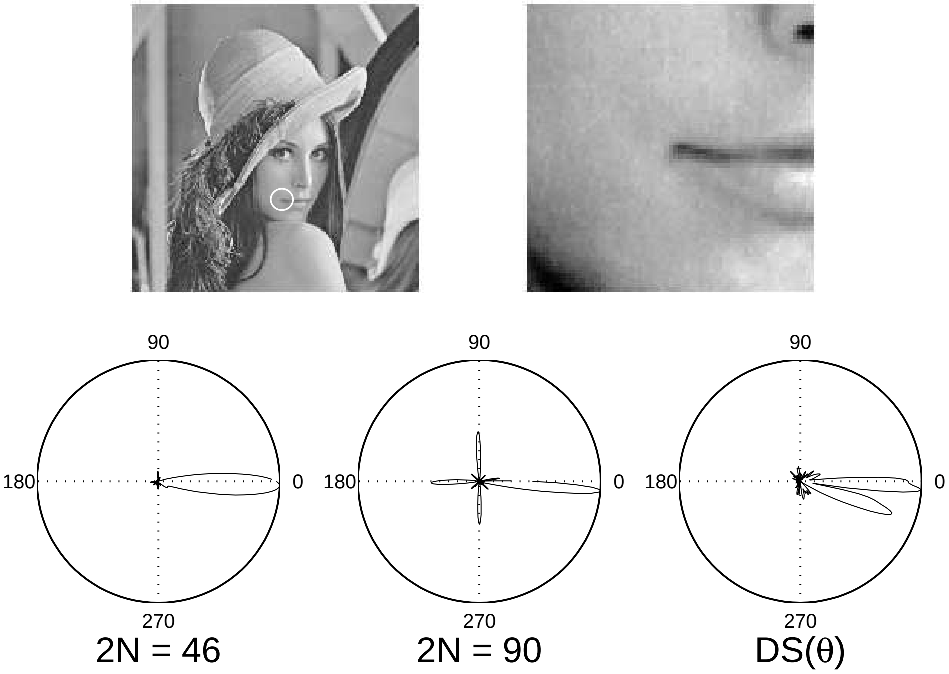


Figure 2.16: *Resolution comparison between the steerable wedge filter [SF96] and our approximately steerable filter.* **Top Left:** *The image "Lena" with her lips corner as a keypoint.* **Top Right:** *Lips corner in detail.* **Bottom Left:** *Polar plots using the steerable wedge filter [SF96] with* 46 *basis filters. Only the dominant dark line between the lips can be recognized.* **Bottom Middle:** *Even with* 90 *basis filters we cannot recognize two lips.* **Bottom Right:** $DS(\theta)$ *of our approximately steerable filter. The edges of two lips are characterized distinctly.*

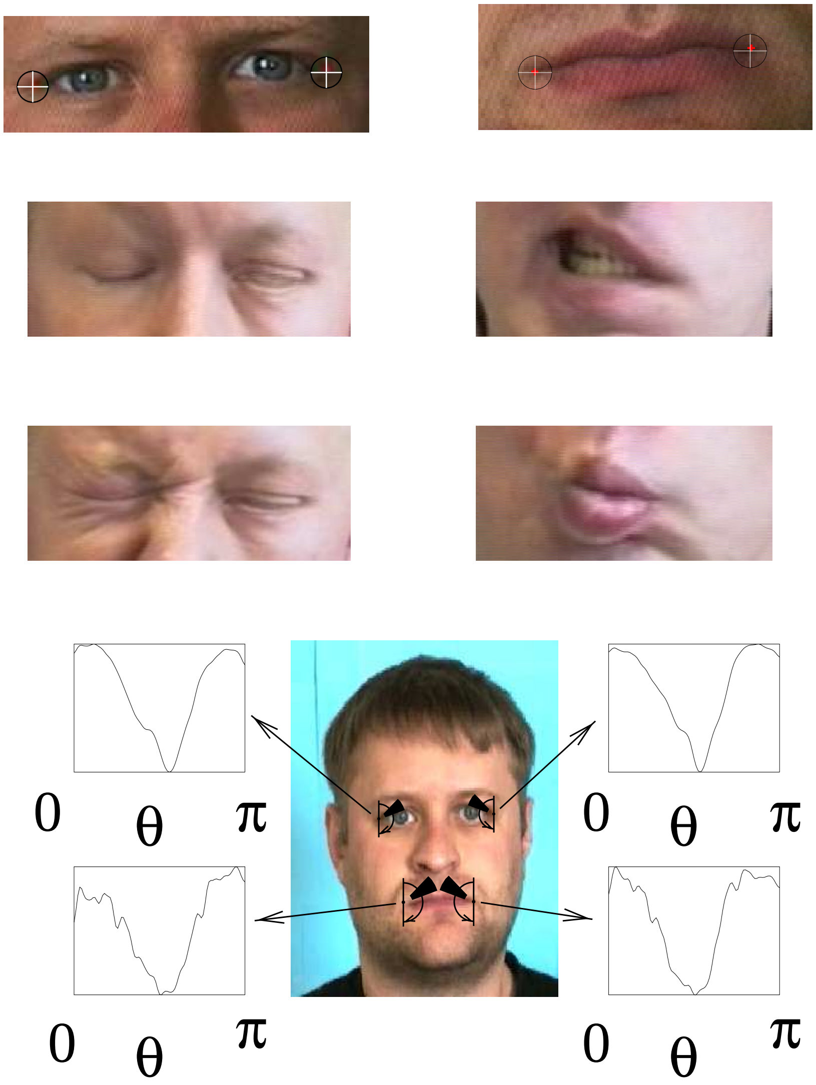


Figure 2.17: *The orientation signatures can be used for facial paresis detection.* **Row 1:** *The eye region and the mouth region of a healthy person with the keypoints for applying our filter.* **Row 2 and Row 3:** *The eye region and the mouth region of a paresis patient. We can observe that there is no more symmetry between both sides of his face.* **Row 4:** *After applying our filter we obtain orientation signatures for further symmetry analysis. (Images courtesy of A. Gebhard, University Erlangen-Nuremburg).*

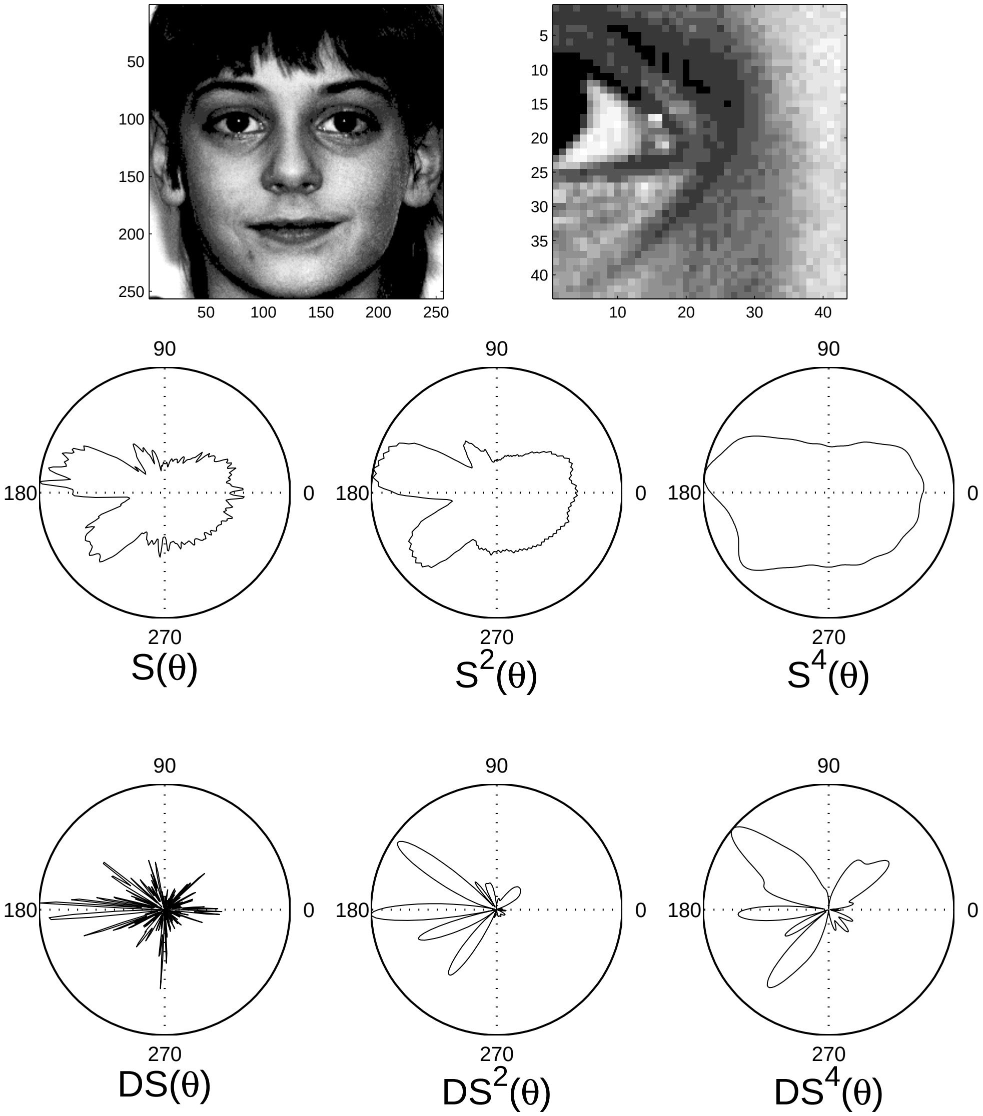


Figure 2.18: **Top Left:** *Face of a child.* **Top Right:** *The left corner of its right eye in detail. It can be regarded as a combination of irregular wide lines and blurred edges disturbed by noise.* **Middle:** *Orientation signatures reconstructed from the first, third and fifth level of the polar pyramid. The eyelids are shown clearly in $S^2(\theta)$ as two maxima at $135°$ and $225°$. The local maximum near $180°$ is due the white of the eye.* **Bottom:** *Corresponding $DS(\theta)$, $DS^2(\theta)$ and $DS^4(\theta)$. At small scales the signatures present more detail, but are also sensitive to noise. At large scales we obtain dominant structures but lose details. A valid characterization should combine all these signatures.*

2.5 Conclusion

Current orientation steerability approaches are based on the Fourier decomposition of the steered filter with respect to orientation. Although they are optimal with respect to approximation error, they suffer from the uncertainty principle: In order to achieve high orientation resolution, a huge number of basis filters must be applied.

We proposed a new approach to obtain the orientation signature for junction characterization. It enables the approximation of a continuous response with respect to orientation based on a number of basis filter responses. These filters are directly designed in the spatial domain. The narrow angular support of the basis functions enables a high orientation resolution with a moderate computational load. We showed this difference to earlier approaches both in theory as well as in real images of junctions.

We use Gaussian functions of 0-th and 1st order to characterize lines and edges, respectively. Though the corresponding filter responses $S(\theta)$ and $DS(\theta)$ seem similar to the even and the odd response of a quadrature filter, it is not suitable to combine them together as a single energy response since they are no Hilbert transformation of each other. Furthermore, in junction classification the magnitude signature of a quadrature filter would not suffice to discriminate lines from edges and we would further need odd/even or magnitude/phase signatures. In other words, the advantage of quadrature filter is its invariance with respect to both lines and edges. If we are not interested in this invariance, we need not to use a quadrature filter. Instead, we are interested in treating lines and edges separately. Thus, separate templates for lines and edges are more suitable.

In the next chapter, we will extend this approximate steerability to 3D filtering and display their applications in 3D image processing.

Chapter 3

3D Orientation Steerability

Abstract

This chapter presents a new Gaussian function based method for the extraction of local 3D orientation information. This new method is based on the decomposition of the sphere with a set of overlapping basis filters in the feature space. Compared with current 3D steerability approaches our method achieves higher orientation resolution with moderate complexity. This property enables us to solve challenging problems like 3D junction characterization, complex surface analysis and multiple motion estimation. We further study the problem of non-uniform distribution of the spherical coordinates and discuss the application of a weighting compensation function in the computation of a 3D orientation signature. In the analysis of our new method against a broader background, we compare our method with related themes such as 3D orientation histogram and spherical wavelets.

3.1 Introduction

In general filtering there is a conflict between performance and complexity. For example, in the orientation analysis we prefer filters with fine orientation resolution. But we have to face an enormous computational complexity while constructing or rotating such filters. In order to attenuate this conflict, the concept of steerability was introduced [FA91]. Having proposed a new kind of 2D approximately steerable filter in chapter 2, we further study the 3D orientation steerability in this chapter. While many 2D steerable filters have been applied in image processing and low level computer vision ([FA91, GBG$^+$94, MS94, Per95, SF96, FP98, MPS98]), there were only a few approaches dealing with 3D steerability [FA91, And92, HC95] which are mainly based on the global decomposition method. In the following, we would like to give a review of these related works.

The Work of Freeman and Adelson

Freeman and Adelson [FA91], being the first who introduced the concept of steerability into 3D filtering, interpolated derivatives of 3D Gaussian functions with a set of basis filters which are rotated copies of the original filter. The corresponding interpolation functions are trigonometric functions of the orientation parameters. As an example, we steer the first derivative of a 3D Gaussian function $G_1^{(\alpha,\beta,\gamma)}(x,y,z)$, whose orientation is represented with three directional angles (α,β,γ) between the axis through the filter lobes and x, y, and z axis, respectively

$$\begin{aligned} G_1^{(\alpha,\beta,\gamma)}(x,y,z) &= cos(\alpha)G_{1x}(x,y,z) \\ &+ cos(\beta)G_{1y}(x,y,z) \\ &+ cos(\gamma)G_{1z}(x,y,z), \end{aligned} \tag{3.1}$$

where three basis filters are rotated copies of $G_1^{(\alpha,\beta,\gamma)}$ along x, y, and z axis, respectively

$$\begin{aligned} G_{1x}(x,y,z) &= -xe^{-\frac{x^2+y^2+z^2}{2}}, \\ G_{1y}(x,y,z) &= -ye^{-\frac{x^2+y^2+z^2}{2}}, \\ G_{1z}(x,y,z) &= -ze^{-\frac{x^2+y^2+z^2}{2}}. \end{aligned}$$

For simplicity we set the coefficients of basis filters to one. Here the interpolation functions are $cos(\alpha)$, $cos(\beta)$, and $cos(\gamma)$.

According to equation (3.1), we need only three basis filters to synthesize $G_1^{(\alpha,\beta,\gamma)}$ in an arbitrary direction. Hence, the enormous computational complexity in rotating this filter to different directions is strongly attenuated.

Andersson's Work

Andersson attacked the problem of constructing 3D steerable filters with spherical harmonics [And92]. He pointed out that the filter shapes are not the same for spherical harmonics of order no less than two. This property consequently complicates the interpolation procedure.

To simplify the interpolation, Andersson defined an alternative set of basis filters of order l in the frequency domain [And92]

$$B_{li}(\overline{u}) = H(\rho)(\hat{n}_{li} \cdot \hat{u})^l, \tag{3.2}$$

where $\overline{u}$ and $\hat{u}$ are an arbitrary frequency coordinate vector and its corresponding normalized unit vector, respectively. The vector $\hat{n}_{li}$ denotes the orientation of the i-th basis filter of order l, and $H(\rho)$ represents the radial frequency response. According to this definition, the basis filters with the same order have the same shape. Furthermore, the orientation of basis filters is arranged in such a way that the basis filters with the same order also span evenly on the sphere. Correspondingly, the synthesized filters of order l after the interpolation procedure are rotated copies of one basis filter with the same order. After studying the regular polyhedra in detail, Andersson held that it is impossible to distribute more than ten basis filters evenly on the sphere [And92]. Consequently, basis filters with order $l \geq 4$ cannot span evenly on the sphere, as the number of basis filters is equal to $\frac{(l+1)(l+2)}{2}$.

Remark about Orientation Resolution

Here we would like to address the problem of orientation resolution in both approaches. As a matter of fact, we only need to analyze the performance of one basis filter since the synthesized filters are rotated copies of one basis filter. In the work of Freeman and Adelson, the derivatives of Gaussian functions have coarse orientation resolution due to their large spatial supports in the orientation space, as shown in figure 3.5. In Andersson's work, the basis filters are centered at the vertices of the corresponding regular polyhedron. In order to span the whole sphere with a set of basis filters, the angular support of each basis filter should not be smaller than a facet of the corresponding regular polyhedron. Correspondingly, the smaller the area of a facet is, the narrower support a filter will have. Since the orientation resolution of a filter is inversely proportional to its angular support, we may analyze the resolution property of a filter according to the corresponding angular support. It is confirmed that even for an icosahedron, which is a matching regular polyhedron with the smallest facet, the corresponding support of one facet is not yet small enough [Hor86]. Thus, the steerable filter proposed by Andersson does not provide sufficiently fine resolution, either.

The drawback to having insufficient orientation resolution limits the applications of current 3D steerable filters to solving challenging problems like 3D junction characterization, range image analysis, and multiple motion estimation.

In this chapter, we propose a new kind of 3D approximately steerable filter to extract local 3D orientation information using Gaussian functions in the orientation space. Compared with current 3D steerable filters, it achieves higher orientation resolution with moderate complexity.

This chapter is organized as follows: In section 2 we present the new filter and its responses of 3D planes in detail. Then we compare our filter with current 3D steerable filters in section 3. After that we display synthetic and real application examples of 3D junction characterization and range image processing in section 4. In section 5, we further analyze our approach in a broader background. Lastly, we conclude this chapter in section 6.

3.2 Local 3D Orientation Analysis

Definition of 3D Approximately Steerable Filter

To analyze 3D orientation naturally, we first compute a spherical mapping on the input data: $I(x, y, z) \to I(r, \theta, \phi)$, where $r = \sqrt{x^2 + y^2 + z^2}$, $\theta = \arctan(\frac{y}{x}), \phi = \arctan(\frac{z}{\sqrt{x^2+y^2}})$ (see figure 3.1). Here, the goal is to build an orientation signature $S(\theta, \phi)$ from $I(r, \theta, \phi)$. In order to have fine orientation resolution, we use *conic kernels* with small angular supports as basis filters to sample the orientation space locally. A *conic kernels* centered at (θ_i, ϕ_j) reads

$$A_{(\theta_i,\phi_j)}(r, \theta, \phi) := \frac{G_0^{(\theta_i,\phi_j)}(\theta, \phi)}{\mathcal{N}_{R_{min},R_{max}}^{(\theta_i,\phi_j)}(r)}, \tag{3.3}$$

where $\mathcal{N}^{(\theta_i,\phi_j)}_{R_{min},R_{max}}(r)$ is a weighting function along the radial direction and it is independent of the angular part of the filter. We will come back to the design of $\mathcal{N}(r)$ later. The angular part of the kernel is a 2D Gaussian function in the orientation space coordinated with (θ, ϕ)

$$G_0^{(\theta_i,\phi_j)}(\theta,\phi) := \frac{1}{2\pi\sigma^2}e^{-\frac{(\mathcal{D}(\theta,\theta_i))^2+(\phi-\phi_j)^2}{2\sigma^2}}, \tag{3.4}$$

with σ denotes the scale of the 2D Gaussian function. Since the angles along the θ direction are periodic, we define $\mathcal{D}(\cdot)$ like in chapter 2 to represent the minimal circular difference between θ and θ_i ($\theta, \theta_i \in [0, 2\pi]$)

$$\mathcal{D}(\theta,\theta_i) := \min(|\theta-\theta_i|, |\theta-\theta_i-2\pi|, |\theta-\theta_i+2\pi|). \tag{3.5}$$

One *conic kernels* centered at (θ_i, ϕ_j) is shown in figure 3.1. Theoretically, a Gaussian function is not compactly supported. Thus, we consider only the part of $G_0^{(\theta_i,\phi_j)}(\theta,\phi)$ inside the circular mask with a diameter D in implementation, as shown in figure 3.1 and equation (3.6).

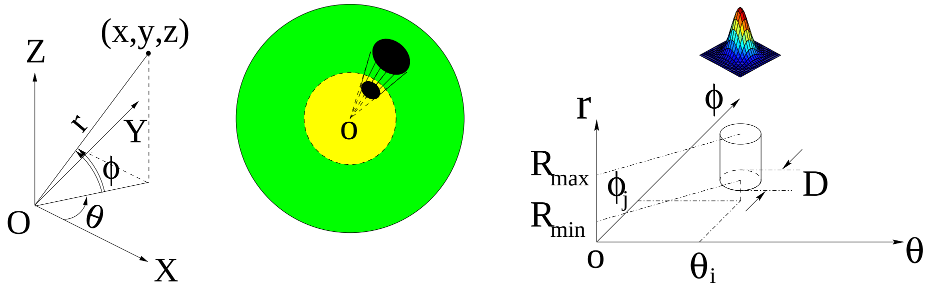


Figure 3.1: *A* conic kernels *centered at (θ_i, ϕ_j) with radial boundaries R_{min} and R_{max}.* **Left:** *The definition of the spherical coordinate system.* **Middle:** *The filter kernel in the 3D Cartesian coordinate system. The keypoint is at the center of the sphere.* **Right:** *The filter kernel with θ, ϕ and r as coordinates. A* conic kernels *in the Cartesian coordinate system turns into a cylinder. In the (θ, ϕ) plane the circular mask with a diameter D is weighted by a 2D Gaussian function, as shown above the cylinder.*

After applying such a *conic kernels* on $I(r, \theta, \phi)$, we get a basis filter response as a local sample located at (θ_i, ϕ_j)

$$\gamma_{(\theta_i,\phi_j)} := \sum_{\{(\theta,\phi)|\sqrt{(\theta-\theta_i)^2+(\phi-\phi_j)^2}\leq\frac{D}{2}\}}\sum G_0^{(\theta_i,\phi_j)}(\theta,\phi)\sum_{r=R_{min}}^{R_{max}}\frac{I(r,\theta,\phi)}{\mathcal{N}^{(\theta_i,\phi_j)}_{R_{min},R_{max}}(r)}. \tag{3.6}$$

Here the inner sum averages over the radius by normalizing with $\mathcal{N}^{(\theta_i,\phi_j)}_{R_{min},R_{max}}(r)$. The outer sums are the projection of the shifted 2D angular Gaussian on the angular signal.

Now let us consider the sampling of (θ, ϕ) plane using a set of basis filters. It is known that a sphere forms a rectangular region in the (θ, ϕ) plane. For this rectangular region it is impossible to have a tessellation with circular cells. Instead, we may overlap neighboring basis kernels to cover the whole rectangular region, as shown in figure 3.2. In this arrangement, we observe that this rectangular region is periodic along the θ direction and is mirror-symmetric about the boundary along the ϕ direction. These periodic and mirror-symmetric properties help to solve the boundary problem.

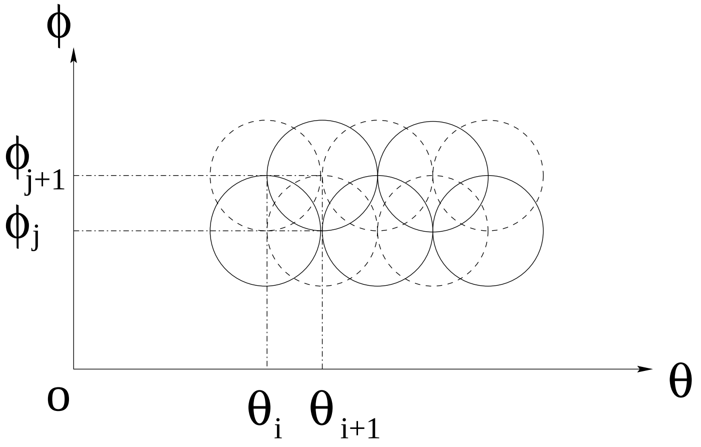


Figure 3.2: *The sampling of (θ, ϕ) plane using a set of* conic kernels. *The horizontal or vertical distance between two neighboring masks is equal to the radius of one mask.*

In order to obtain the orientation signature $S(\theta, \phi)$ from a set of samples $\gamma_{(\theta_i,\phi_j)}$, we use 2D Gaussian functions with local support $G_0^{(\theta_i,\phi_j)}(\theta, \phi)$ again as interpolation functions yielding

$$S(\theta, \phi) := \sum_{\theta_i} \sum_{\phi_j} \gamma_{(\theta_i,\phi_j)} G_0^{(\theta_i,\phi_j)}(\theta, \phi). \tag{3.7}$$

The legality of using 1D Gaussian functions as interpolation functions was already proved in [PG90]. Our approach can be viewed as an extension into 2D feature space. So far, we define an analytic model of 3D orientation analysis based on angular Gaussian functions.

Filter Responses of 3D Planes

For motion estimation we are interested in filter responses of 3D planes since it is proven that a single translational motion corresponds to a single plane [AB85, HC95] and multiple motions correspond to multiple planes [SM91, YDBS99] in the derivative space or in the frequency space. In the 3D Cartesian coordinate system, a plane passing through the origin $(0, 0, 0)$ with a unit normal vector $\mathbf{n} = (n_1, n_2, n_3)^T$ reads

$$xn_1 + yn_2 + zn_3 = 0. \tag{3.8}$$

In order to represent this plane with parameters θ and ϕ, we convert the Cartesian coordinates into spherical coordinates

$$\left\{ \begin{array}{lcl} x &=& r\cos(\phi)\cos(\theta) \\ y &=& r\cos(\phi)\sin(\theta) \\ z &=& r\sin(\phi), \end{array} \right.$$

and

$$\left\{ \begin{array}{lcl} n_1 &=& \cos(\phi_n)\cos(\theta_n) \\ n_2 &=& \cos(\phi_n)\sin(\theta_n) \\ n_3 &=& \sin(\phi_n). \end{array} \right.$$

After wiping out the radial variable r, we acquire an equation of the 3D plane with variables θ and ϕ

$$\cos(\phi)\cos(\phi_n)\cos(\theta - \theta_n) + \sin(\phi)\sin(\phi_n) = 0. \tag{3.9}$$

Different planes have different representations in the (θ, ϕ) space. For horizontal and vertical planes, whose normal vectors are parallel to the coordinate axes, their corresponding representations in the (θ, ϕ) space are straight lines, as shown in figure 3.3. In contrast, tilted planes in the Cartesian coordinates turn into periodic curves in the (θ, ϕ) space, as shown in figure 3.4. These curves look like trigonometric functions with different amplitudes and phases. If we know the extreme point of one curve with the maximal ϕ coordinate, ϕ_m, and the corresponding θ coordinate, θ_m, then we can find out the normal vector of the corresponding plane (see Appendix A for derivation)

$$\left\{ \begin{array}{lcl} \theta_n &=& \theta_m \pm \pi \\ \phi_n &=& \frac{\pi}{2} - \phi_m \end{array} \right. . \tag{3.10}$$

Here we use $+$ or $-$ sign to determine θ_n in equation (3.10) in such a way that the third component n_3 of the normal vector is positive. In practice, we obtain a set of points in the (θ, ϕ) space according to equation (3.9). It is not easy to find out the extreme point (θ_m, ϕ_m) directly due to noise or inadequate number of points. Therefore, we have to extract the parameter (θ_n, ϕ_n) from a set of points. This is a standard regression problem. For a single curve the least square estimation (LSE) algorithm is applicable; for multiple curves we may apply the expectation-maximization (EM) algorithm. We will give particulars of this point in chapter 4.

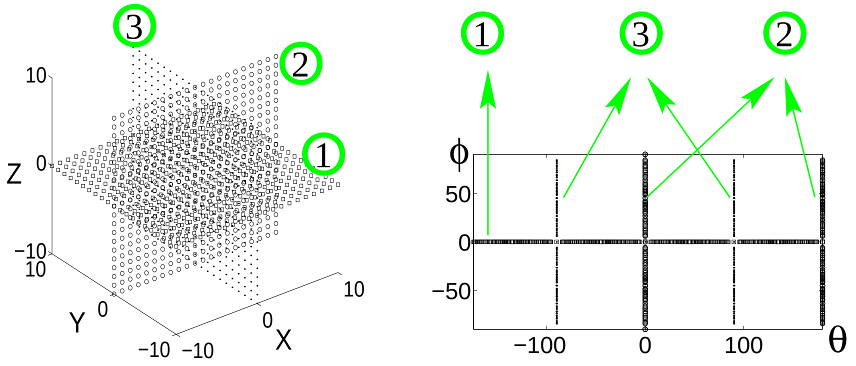


Figure 3.3: **Left:** *Three special planes in the Cartesian coordinates with normal vectors* $(0, 0, 1)$, $(1, 0, 0)$, *and* $(0, 1, 0)$, *respectively.* **Right:** *Special planes in the* (θ, ϕ) *space.*

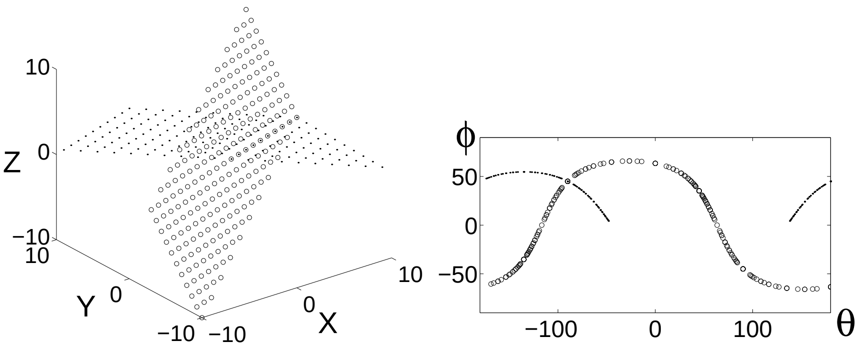


Figure 3.4: **Left:** *A plane with normal vector* $(-2, 1, 1)$ *and a plane with normal vector* $(1, 1, 1)$ *in the Cartesian coordinates.* **Right:** *The corresponding curves in the* (θ, ϕ) *space. Since the components on the plane with normal vector* $(1, 1, 1)$ *have only positive* z *coordinates, we observe only positive* ϕ *coordinates on the corresponding curve.*

3.3 Comparisons with Current 3D Steerable Filters

Current 3D steerability approaches are based on the global decomposition principle. In contrast, our 3D filter is based on the local decomposition principle. This difference leads our approach to have higher orientation resolution. In figure 3.5 we show the filter $G_1^{(\alpha,\beta,\gamma)}$ in the work of Freeman and Adelson [FA91], Andersson's third order filter [And92] (whose basis filters span evenly on the sphere with the finest angular support), and our filter, respectively. Since the orientation resolution of a filter can be measured with the angular support of this filter, we display also the angular supports of these filters in the (θ, ϕ) space with white regions. We obtain these regions by integrating the filter kernels over the radial variable. Note that the angular support of the filter in the spatial domain is the same as that in the frequency domain since the Fourier transform is an isometric mapping. The irregularity in the (θ, ϕ) space with $|\phi| > 40°$ is caused by the discrete representation of filter kernels. We notice that $G_1^{(\alpha,\beta,\gamma)}$ has a so large angular support that only the gap between its two lobes may be useful. Actually, Huang and Chen used this gap to fix the orientation of *one* plane in the single motion estimation [HC95]. Obviously, $G_1^{(\alpha,\beta,\gamma)}$ cannot detect multiple planes simultaneously. The orientation resolution of Andersson's filter is only a little bit better. Compared with these two steerable filters, our filter has a much higher orientation resolution.

The computational burden of applying a steerable filter is determined by the number of basis filters and the spatial support of each basis filter. Given the fact that current steerable filters and our filter are based on different decomposition principles, we can compare their complexity only by considering the computational burden per pixel in the input data. Concretely,

- The filter G_1 is composed of three basis filters with the *global* support, i.e. each basis filter covers the input data completely. Thus, each pixel in the input data is involved in the scalar product as well as in the interpolation procedure three times.

- The third order filter in Andersson's approach has ten basis filters. Thus, each pixel in the input data is involved in the scalar product and the interpolation procedure ten times.

- Our filter is based on the local decomposition principle. In figure 3.2 we ob-

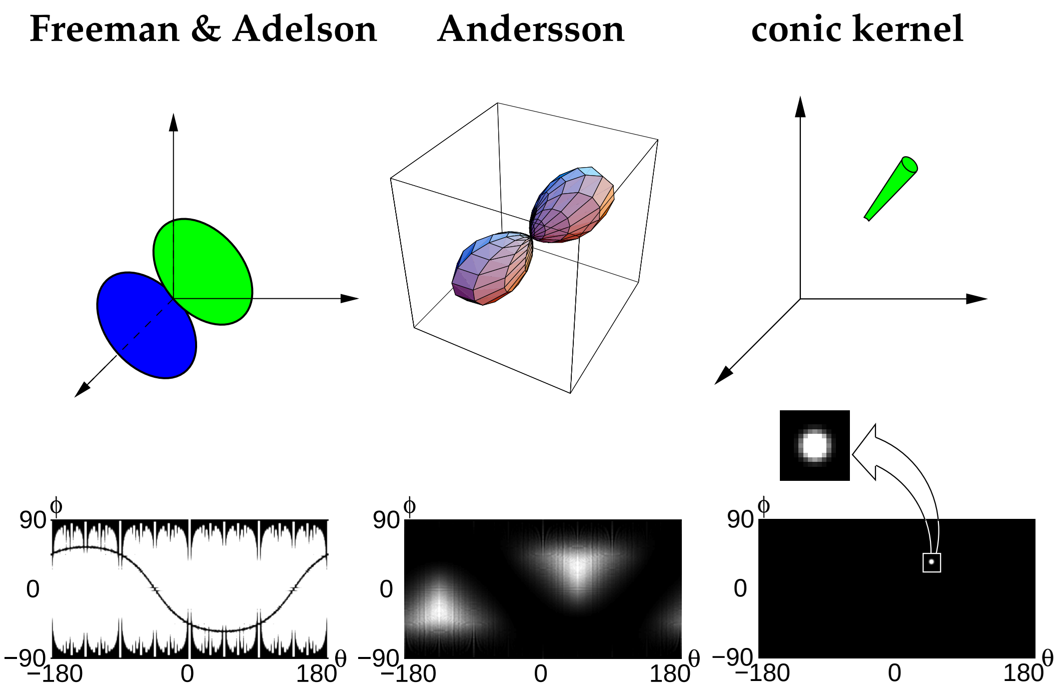


Figure 3.5: **Row 1:** *The rendering image of filter kernels.* **Left:** *The filter* $G_1^{(\alpha,\beta,\gamma)}$ *in the work of Freeman and Adelson (redrawn from [HC95]).* **Middle:** *The third order filter (i.e. with ten basis filters) in the frequency domain in Andersson's approach (redrawn from [And92]). Note that the angular support of the filter in the spatial domain is the same as that in the frequency domain since the Fourier transform is an isometric mapping.* **Right:** *Our filter.* **Row 2:** *The corresponding angular supports for above filters centered at* $\theta = 45.00°, \phi = 35.26°$ *are shown with white regions in the* (θ, ϕ) *space. These supports are actually measurements of the orientation resolution of the filters. The gap between two lobes of* $G_1^{(\alpha,\beta,\gamma)}$ *is clearly represented as a black curve in the angular support map. For clarity we enlarge the angular support of our filter in an extra image.*

serve that the quadratic area bounded by four lines $\theta = \theta_i$, $\theta = \theta_{i+1}$, $\phi = \phi_j$, and $\phi = \phi_{j+1}$ is covered by four quarter circular masks. Without studying the overlapping exactly, we may roughly say that a pixel in this quadratic area is involved in the scalar product four times. As the interpolation function has the same support as the basis filter we know that a pixel in this quadratic area is involved in the interpolation four times as well.

From above analysis, our filter needs a little bit more computation than the filter G_1 but much less computation than Andersson's filter.

It should be noticed that a complexity comparison is only fair, when the corresponding filters provide (about) the same orientation resolution. This is not the case for the three 3D steerable filters mentioned above. Actually, neither the filter G_1 nor Andersson's filter can achieve the same fine orientation resolution that our filter provides. One possibility to achieve such a fine orientation resolution using global decomposition method is to generalize the steerable wedge filter [SF96] from 2D space to 3D space with considerably higher effort [YDS01], which is not yet implemented according to current literature.

3.4 Applications in Volume Image Processing

Compensation Issue

Before presenting examples, let us discuss the design of the weighting function $\mathcal{N}(r)$ (see equation (3.3)). It is known that the horizontal angle θ and the vertical angle ϕ are defined differently in the spherical coordinates. All points with the same θ on a sphere surface lie on a great circle of this sphere, whereas all points with the same ϕ lie on a small circle. If we divide the whole (θ, ϕ) space with a homogeneous grid, it turns out that the higher the latitude value is, the denser the grid points are on the sphere surface. This kind of non-uniform distribution was addressed in [Hor86] in detail.

Actually, this non-uniform distribution exists already in 2D space. When we represent evenly distributed grid points in 2D Cartesian coordinates with the polar coordinates, these points are no more evenly distributed. In chapter 2, we normalized the filter outputs with the averaging factor $\mathcal{N}(R_{\min}, R_{\max}, \theta_k)$ (see equations (2.6) and (2.12)) to compensate the non-uniform distribution. Similarly, we may average this non-uniform distribution in 3D space by designing the weighting function $\mathcal{N}(r)$ as the sum of discrete weights in the basis kernels so that the filter response is relatively insensitive to the non-uniform distribution. This compensation "strengthens" the outputs of filter kernels with a few points and "suppresses" those outputs of filter kernels with many points. As a result, we are no more able to know the real distribution density of points in the (θ, ϕ) space. However, the density information is desirable in many motion estimation approaches. For example, in the EM algorithm we purely use statistics to extract parameters from a set of sample points with the belief that there are more *normal* points with similar statistic properties than noise and "incorrect" sample points with large deviation from the bulk of all data points [PTVF92]. The distribution density actually works as a weighting factor in the parameter regression procedure. If we lose the distribution density information, the estimation result will be much worse. For this reason, we would like to preserve the distribution density information by setting $\mathcal{N}(r)$ as a real positive constant.

In the next subsection, we show some examples in 3D orientation analysis with and without compensation and compare the corresponding filter responses. We will address the applications in multiple motion estimation in chapter 4.

Application Examples

This subsection is mainly about the applications of our method in volume image processing. Without exception we choose the parameters as follows: $D = \frac{\pi}{90}$, $R_{\min} = 0.2 \max(R)$, $R_{\max} = 0.8 \max(R)$.

We first apply the filter *with* averaging by setting $\mathcal{N}(r)$ as the sum of discrete weights inside the sampling kernels and obtain the orientation signature $S_a(\theta, \phi)$. Then we apply the filter *without* averaging by setting $\mathcal{N}(r)$ as a constant and denote the corresponding orientation signature with $S(\theta, \phi)$.

As an example of 3D junction characterization, we have in figure 3.6 a synthesized cubic with one of its vertices as keypoint. For comparison we apply the steerable filter $G_1^{(\alpha,\beta,\gamma)}$, Andersson's filter, and our filter (with and without averaging), respectively. In the response of $G_1^{(\alpha,\beta,\gamma)}$, the location of the maximal value is not geometrically meaningful since the angular support of $G_1^{(\alpha,\beta,\gamma)}$ is too large to interpret this 3D junction. Andersson's filter has higher orientation resolution than $G_1^{(\alpha,\beta,\gamma)}$. Though the edge information is blurred, the location of maximal value in the response corresponds to the location of weight center of the cubic. Compared with these two steerable filters, our filter provides evidently higher orientation resolution. Clearly, $S_a(\theta, \phi)$ is insensitive to the non-uniform distribution of grid points in the (θ, ϕ) space, but $S(\theta, \phi)$ is sensitive. It should be noticed that in $S(\theta, \phi)$ the edges are still clearly represented. It is possible to further extract edge information from the binary version of $S(\theta, \phi)$ by utilizing morphological operations. Taking into account that, by applying the filter, we diffuse one point of the (θ, ϕ) space into a circular region with a diameter of D, we perform "erosion"

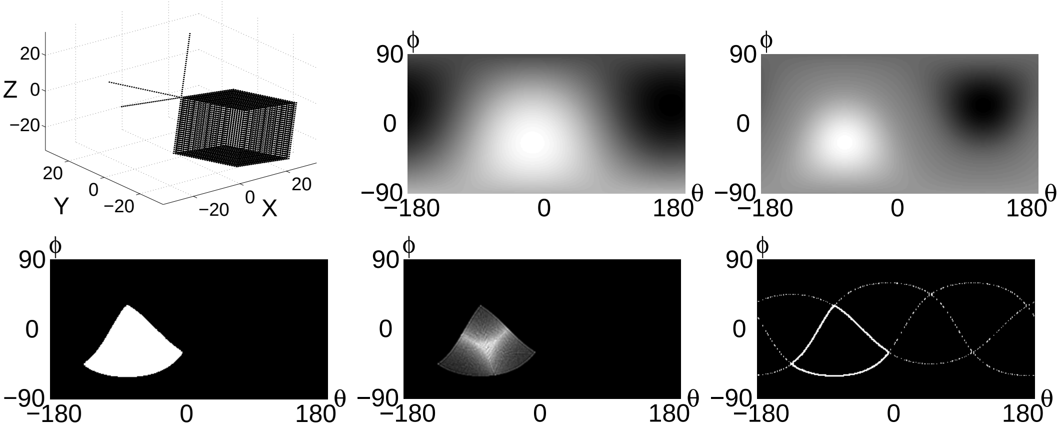


Figure 3.6: **Top Left:** *3D plot of a cubic with its three normal vectors at one vertex.* **Top Middle:** *The response of $G_1^{(\alpha,\beta,\gamma)}$ [FA91].* **Top Right:** *The response of Andersson's steerable filter [And92].* **Bottom Left:** *Orientation signature $S_a(\theta,\phi)$. It is insensitive to the non-uniform distribution of grid points in the (θ,ϕ) space. Three surfaces turn out to be three edges in the (θ,ϕ) space.* **Bottom Middle:** *Orientation signature $S(\theta,\phi)$. Compared with $S_a(\theta,\phi)$ it is more sensitive to the non-uniform distribution of grid points in the (θ,ϕ) space. But the edges are still clearly represented.* **Bottom Right:** *Extracted edges after morphological operations "erosion" and "remove". The junctions of these curves describe the orientation of ridges of neighboring edge planes. For comparison we also display those three surfaces connected with the keypoint using dotted curves. It is clear to see that they are consistent with the extracted edges.*

morphological operations to "shrink" edges by $\frac{D}{2}$. Then we use one "remove" morphological operation to extract edges. For comparison we display three surfaces of the cubic connected with the keypoint as well. They are consistent with the extracted edges.

In figure 3.7 we have a real range image of a cubic with one of its vertices as keypoint, as shown in the white window. For clarity we display the 3D plot of this range image in the window as well. In this plot, we convert the range variable into z coordinate and normalize the z coordinate with the maximal value of x (or y) coordinate. This facilitates the application of our filter. After applying the filters on this 3D plot, we compare the signatures $S_a(\theta,\phi)$ and $S(\theta,\phi)$. We can see that the amplitude of the regions with large ϕ is "suppressed" in $S(\theta,\phi)$, while the main structure information remains in both signatures.

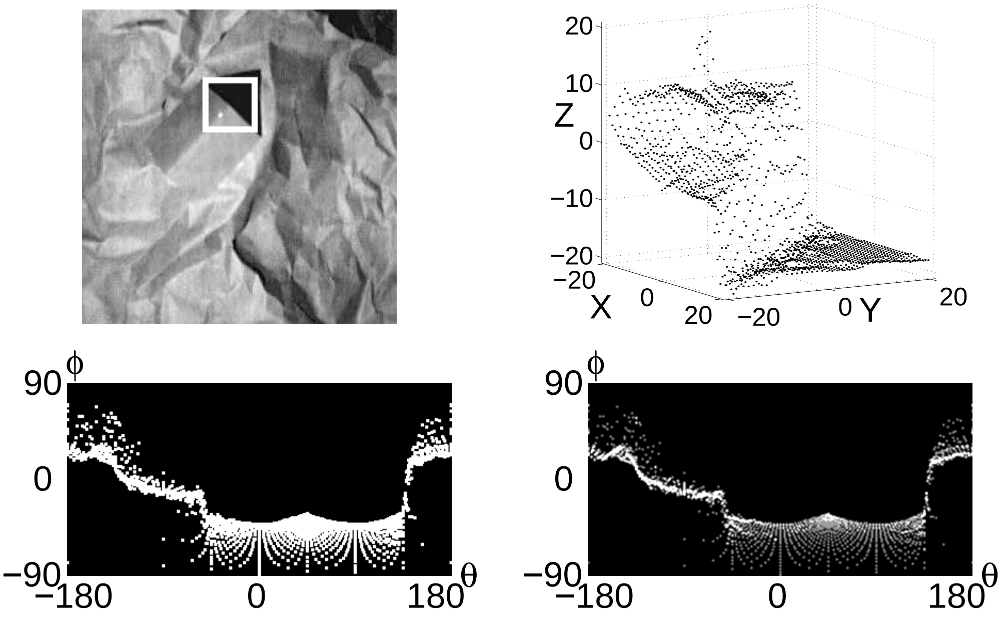


Figure 3.7: **Top Left:** *Range image of a real cubic. The white box shows us the window for 3D orientation analysis. The keypoint for applying 3D filter is at the center of the window.* **Top Right:** *3D plot of the range image inside the window. We set its center as* $(0, 0, 0)$. **Bottom Left:** *Orientation signature* $S_a(\theta, \phi)$. **Bottom Right:** *Orientation signature* $S(\theta, \phi)$. *We can see that the amplitude of the regions with large* ϕ *is "suppressed" in* $S(\theta, \phi)$, *while the main structure information remains.*

Figure 3.8 shows us a range image of a cup. From the 3D plot we observe that the bottom of the cup, the rim and the upper part of the handle are well represented in the range image. But there are just bare pixels corresponding to the side of the cup. We set the center of the cup bottom as keypoint and normalize the z coordinate according to the maximal value of x (or y) coordinate. In $S_a(\theta, \phi)$ we observe that the side of the cup is very much "strengthened". In contrast, $S(\theta, \phi)$ is really proportional to the distribution density. The filter responses of the bottom and the rim are much stronger than the filter response of the side of the cup. Particularly, the filter response of upper part of the handle is more clearly to see near $\theta = -120°, \phi \in [30°, 50°]$ in $S(\theta, \phi)$ than in $S_a(\theta, \phi)$. Notice that the white pixel near $(\theta, \phi) = (80°, -50°)$ in $S_a(\theta, \phi)$ is the response of the dark point outside the cup, while in $S(\theta, \phi)$ it is suppressed. The filter without averaging shows its advantage in this situation, as it preserves main structure information and suppresses small disturbance.

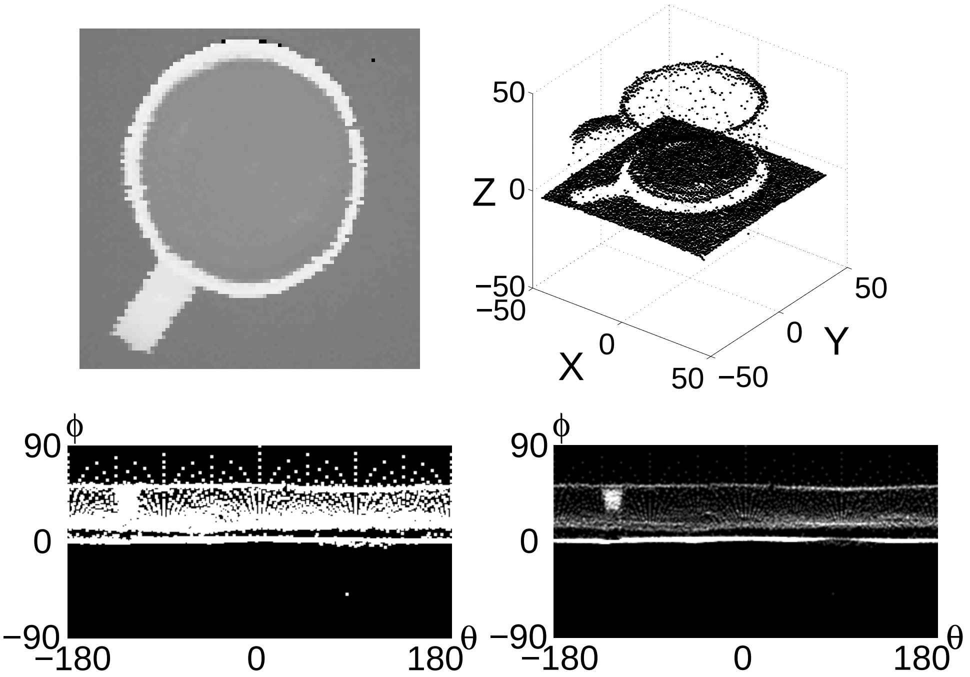


Figure 3.8: **Top Left:** *Range image of a cup.* **Top Right:** *3D plot of the range image. We normalize the z coordinate according to the maximal value of x (or y) coordinate. It shows clearly that there are bare pixels corresponding to the side of the cup and only the upper part of the handle is visible.* **Bottom Left:** *Orientation signature $S_a(\theta, \phi)$. We set the center of cup bottom as keypoint. The side of the cup is very much "strengthened", though there are bare pixels corresponding to the side of the cup in the range image. We can see the filter response of the handle near $\theta = -120°$, $\phi \in [30°, 50°]$. The white point near $(\theta, \phi) = (80°, -50°)$ is the response of the dark point outside the cup.* **Bottom Right:** *Orientation signature $S(\theta, \phi)$. This signature represents the real distribution density. We can observe the filter response of the upper part of the handle more clearly. The filter response of the dark point outside the cup is suppressed now.*

Our filter is also useful for symmetry analysis. In figure 3.9 we have a range image of a propeller with a very complex surface. Since our filter has high orientation resolution, we can describe such a complex surface exactly in the (θ, ϕ) space and further detect the symmetry by minimizing the angular correlation [SS97].

From above experiments we come to the following conclusions that

- Our filter provides higher orientation resolution than current steerability approaches with moderate complexity.

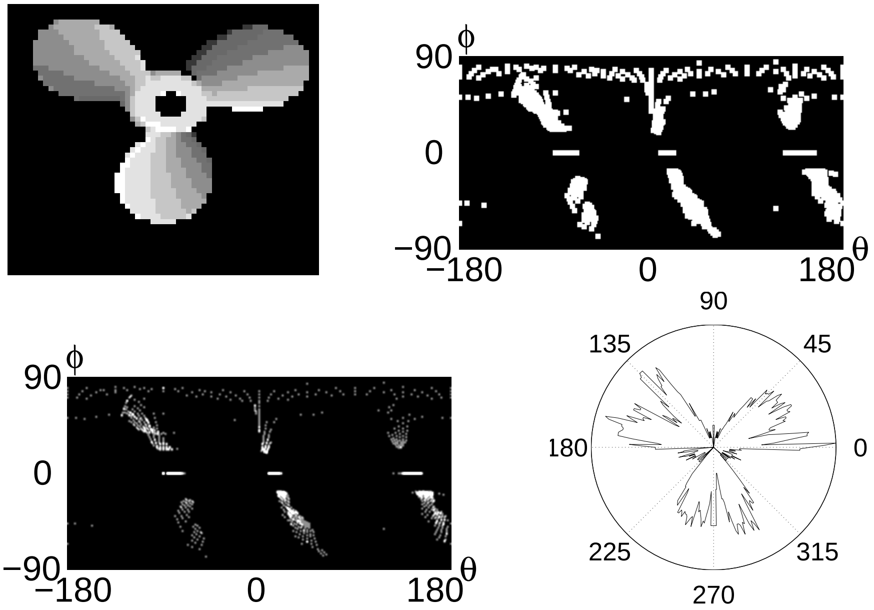


Figure 3.9: **Top Left:** *Range image of a propeller (Image courtesy of C. Sun, CSIRO Mathematical and Information Sciences, Australia). It has complex surfaces.* **Top Right:** *Orientation signature $S_a(\theta, \phi)$. We set the center of the propeller axis as keypoint. Three complex surfaces turn out to be three regions with similar shapes in the (θ, ϕ) space. The white region near $\theta \in [-180°, 180°], \phi = 90°$ in $S_a(\theta, \phi)$ is the filter response of the axis pixels. Though there are only a few pixels on the axis, $S_a(\theta, \phi)$ strengthens their existence.* **Bottom Left:** *Orientation signature $S(\theta, \phi)$. The amplitude of filter response in the regions near $\theta \in [-180°, 180°], \phi = 90°$ in $S_a(\theta, \phi)$ is suppressed.* **Bottom Right:** *Polar plot of $\sum_{\phi} S(\theta, \phi)$. Three symmetric axes can be detected at the minimal correlated positions near $\theta = 90°$, $215°$, and $330°$.*

- The response of our filter without averaging compensation is more sensitive to the non-uniform distribution than the response with averaging. But this susceptibility does not obstruct us from obtaining main structure information in the orientation signature. Moreover, the filter response without averaging shows its advantage in some applications.

3.5 Related Issues

The original motivation of our approach is to improve the orientation resolution of current steerability approaches. It is interesting to observe that our approach is related to some other themes in computer vision and signal processing areas. We believe it will be helpful to analyze our approach against a broader background. In the following, we first explain some related themes and then compare them with our approach.

Extended Gaussian Image and 3D Orientation Histogram

The extended Gaussian image (EGI) is used to represent the surface information of a 3D object. An EGI is a mapping of the object surface onto the unit sphere, in which a small facet on the object surface is transformed into a point whose orientation is the same as that of the small facet and whose weight is the area of the small facet. For convex objects it is proven that their corresponding EGIs are unique. In figure 3.10 we show a simple example of EGI.

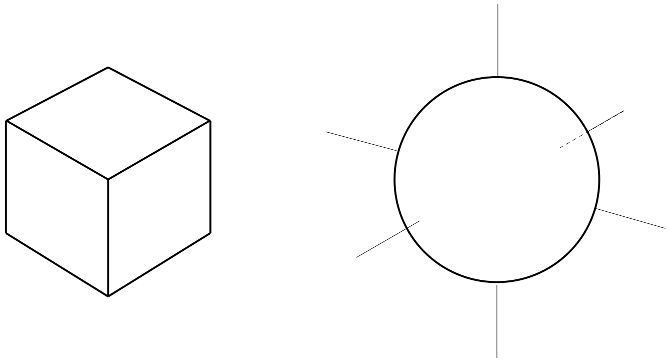


Figure 3.10: *An example of extended Gaussian image.* **Left:** *A square cubic with six faces.* **Right:** *Extended Gaussian image of the cubic. Each needle on the sphere represents the weight of one face of the cubic.*

In practice, we usually use the discrete approximation of EGI, which is referred to as 3D orientation histogram. In order to construct the 3D orientation histogram, we must at first tessellate the unit sphere. According to the ideal tessellation criteria, the sphere should be divided into cells with the same area and the same rounded shape. In addition, these cells should be located as a regular pattern and should provide fine angular resolution [Hor86]. Unfortunately, these criteria cannot be fulfilled at the same time. Among all possible candidates there are only five regular polyhedra, whose facets are located as a regular pattern with the same area and the same shape. They are tetrahedron, hexahedron, octahedron, dodecahedron, and icosahedron. In figure 3.11 we show an icosahedron whose facets have the finest support among all regular polyhedra mentioned above. However, it is clear to see that the facet of the icosahedron is still very large. Thus, none of these five regular polyhedra has an adequately fine orientation resolution.

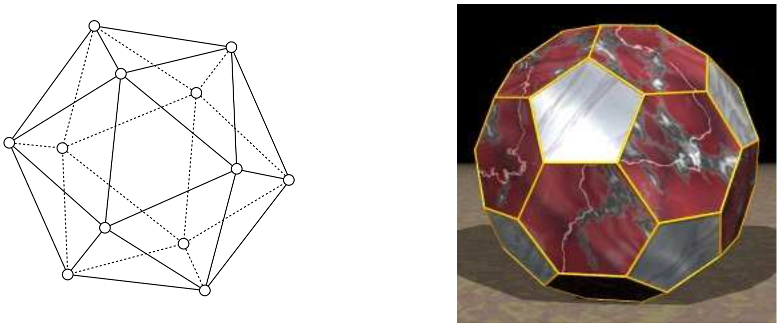


Figure 3.11: *Spherical tessellation using icosahedron and truncated icosahedron.* **Left:** *The icosahedron is composed of* 12 *vertices and* 20 *facets. The facet still has very large support. (redrawn from [And92]).* **Right:** *An example of spherical tessellation using truncated icosahedron composed of* 12 *pentagons and* 20 *hexagons. (This image is copied from http://www.physics.orst.edu/˜bulatov/polyhedra/uniform/u30.html).*

In order to improve the orientation resolution, we often consider the semi-regular polyhedra, whose regular facets are not the same. In figure 3.11 we show such an example using truncated icosahedron. We may start with an icosahedron and subdivide each triangular facet of the icosahedron equally into four smaller triangles. We further subdivide each smaller triangle in the same way repeatedly until we obtain a desired resolution [Hor86]. In order to make the cells round, we then combine six neighboring triangles into one hexagon, yet twelve pentagons composed of five neighboring triangles are always required to tessellate the sphere, just take a football as a typical example. It is clear that a hexagon does not have the same shape and area as a pentagon. Furthermore, though the hexagon has many rotation invariant axes, it is still not *isotropic*, i.e. it is still not rotation invariant for *every* direction.

With respect to the orientation resolution and decomposition principle our approach is very similar to the 3D orientation histogram. Both techniques achieve high orientation resolution and both methods decompose the sphere locally. However, there are still differences between them.

- The 3D orientation histogram is applied for 3D object surface analysis. If the object is convex, we do not consider its position and size. In other words, the 3D orientation histogram is shift- and scale-invariant. In contrast, our 3D filter is applied not only for surface analysis, but also for volume data analysis. It treats both convex and concave objects. But we must fix the keypoint and the radial boundaries at first.

- The 3D orientation histogram works on a unit sphere, while our approach projects the sphere onto the (θ, ϕ) space and works on this orientation space. Though after this non-isometric mapping we lose the rotational symmetry, we gain easier structure display and post-processing as compensation. For example, on the surface of this paper sheet, we cannot display all parts of a great circle of a sphere using the 3D orientation histogram. We have to imagine in our mind that there are parts hiding behind the paper sheet. In contrast, we can display the great circle completely on the (θ, ϕ) space, though with some deformation.
- The basis cells in the 3D orientation histogram cannot fulfill all ideal tessellation criteria simultaneously. In contrast, our approach provides *isotropic* cells in the (θ, ϕ) space (not on the sphere!) which satisfy these criteria.
- The 3D orientation histogram is based on the tessellation of the unit sphere. Each pixel on the sphere is involved *once and only once*. Our approach samples the (θ, ϕ) plane with Gaussian kernels. Since these basis kernels overlap, each pixel on the sphere is therefore involved *several times*. From the point of view of computation complexity, our method needs more computation than the 3D orientation histogram. Of course we can divide the (θ, ϕ) plane with a set of small rectangular cells which do not overlap. But these cells are then non-isotropic. Thus, overlapping is actually the price of having isotropic cells.

Spherical Wavelets

The main issue of this chapter is to find a suitable local basis for feature extraction. This is reminiscent of signal reconstruction theory and wavelet theory. It is known that, after applying a filter $\Phi(\mathbf{x})(\mathbf{x} \in I\!R^n)$, the signal structure inside the filter window is generally blurred [KRV99]. In order to reconstruct the original signal inside the filter window, we need to find out another filter $\Psi(\mathbf{x})(\mathbf{x} \in I\!R^n)$

such that

$$\int \Phi(\mathbf{x}')\Psi(\mathbf{x}+\mathbf{x}')\,d\mathbf{x}' \equiv \delta(\mathbf{x}), \tag{3.11}$$

with the variable $\mathbf{x}' \in I\!R^n$. This equation describes the property of basis in the wavelet theory. If $\Phi(\mathbf{x}) = \Psi(\mathbf{x})$, they are the same basis of the signal; if $\Phi(\mathbf{x}) \neq \Psi(\mathbf{x})$, $\Psi(\mathbf{x})$ is referred to as the dual basis of $\Phi(\mathbf{x})$.

This property was referred to as invertibility by Kalitzin et al. in their work [KRV99]. They proposed to use the derivatives of Gaussian functions as the invertible apertured orientation filters. Along the angular direction they use harmonic functions to serve as basis. Thus, their approach still belongs to the classic global decomposition approaches.

Recently, Schröder and Sweldens proposed a local decomposition approach to build bi-orthogonal wavelets based on the dual basis. They introduced a custom design technique called lifting scheme [SS95]. This lifting scheme combines two arbitrary wavelet functions on the same scale level to build a new wavelet so that the new wavelet is bi-orthogonal to the old one. Compared with the classic wavelet scheme, in which each basis function is a combination of several shifted copies of the same wavelet function on the finer scale level, the lifting scheme does not use scaling and dilation any more. As an example, Schröder and Sweldens built a spherical wavelet to represent functions defined on the sphere. This wavelet is based on the geodesic tessellation of the sphere, which tessellates the sphere with semi-regular polyhedra. In their approach, Schröder and Sweldens focused purely on the decomposition and synthesis properties of the basis and did not delve into the tessellation. Therefore, the limitation of EGI and 3D orientation histogram is still unsolved.

In this chapter, we presented an alternative to the spherical wavelet. In our approach, the local basis filters are non-orthogonal and their responses are locally correlated. Since our goal is not to reconstruct the original signal, but to extract features, this kind of correlation among neighboring basis filters does not bother us and we do not have to "decorrelate" the filter results.

It would be interesting to further study if there exists a dual basis of our basis filter. Though in [KRV99] Kalitzin et al. pointed out that the basis may be non-orthogonal, they did not provide a design technique. The lifting scheme provides a necessary condition which may help us to find out the desired dual basis of our filter so that our method can be used for signal reconstruction as well.

3.6 Conclusion

Early 3D steerability approaches are based on the global decomposition principle and thus suffer from the uncertainty principle strongly, as we pointed out in chapter 2. In this chapter, we presented a new 3D approximately steerable filter based on the local decomposition principle for orientation analysis. Since our filter has Gaussian shape in the (θ, ϕ) space, it achieves the lower bound in the uncertainty principle. This difference leads our filter to be more efficient in localizing one signal both in the spatial domain and in the spectral domain. Consequently, our filter provides higher orientation resolution with moderate complexity. This claim is confirmed by the filter support analysis as well as experiment results.

We implement our method in the feature space directly. Though projecting the sphere onto 2D feature space is not an isometric mapping and the rotation symmetry is lost after projection, this transform benefits structure display and post-processing.

The local basis filters in our method are non-orthogonal. Thus, the filter responses are locally correlated. This correlation does not bother us to extract features. But if we would like to reconstruct original signals, we need a dual basis of our basis filter.

Appendix A: Relation Between The Normal Vector and The Extreme Point

In this appendix, we derive the relationship between the orientation parameters (θ_n, ϕ_n) of the normal vector of a plane and the coordinate (θ_m, ϕ_m) of the extreme point on this plane. In figure 3.12, we represent all possible unit vectors on the 3D plane with a circle. The normal vector $\mathbf{n}$ is perpendicular to all vectors on this plane, including the unit vector pointing to the extreme coordinates (θ_m, ϕ_m) and the vector lying on the XY plane $(\theta_m - \frac{\pi}{2}, 0)$. As the vector $(\theta_m - \frac{\pi}{2}, 0)$ is perpendicular to the vector (θ_m, ϕ_m) as well, the vector $(\theta_m - \frac{\pi}{2}, 0)$ is then the normal vector of the plane containing vectors $\mathbf{n}$ and (θ_m, ϕ_m). Since the normal vector $(\theta_m - \frac{\pi}{2}, 0)$ also lies on the horizontal XY plane, we prove therefore that the plane containing vectors $\mathbf{n}$ and (θ_m, ϕ_m) is perpendicular to the XY plane. In this vertical plane we have

$$\phi_n + \frac{\pi}{2} + \phi_m = \pi. \tag{3.12}$$

This vertical plane always divides the circle equally as it passes through the origin. Taking into account that angles in the θ direction are periodic we have

$$\theta_n = \theta_m \pm \pi. \tag{3.13}$$

Then we obtain equation (3.10).

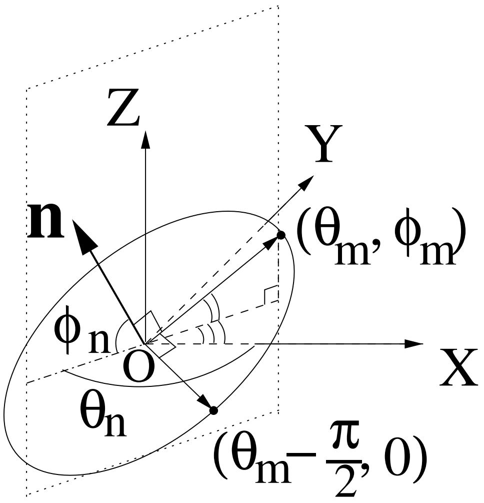


Figure 3.12: *The relation between the orientation parameters (θ_n, ϕ_n) of the normal vector of a plane and the coordinates (θ_m, ϕ_m) of the extreme point on this plane. The circle contains all possible unit vectors on the 3D plane. The plane containing the vectors $\mathbf{n}$ and (θ_m, ϕ_m) is a vertical plane perpendicular to the XY plane.*

Chapter 4

Multiple Motion Analysis Using 3D Steerable Filter

Abstract

In this chapter, we review the history of multiple motion analysis and study occlusion and transparency in the spatial domain as well as in the spectral domain. After comparing spatial and spectral motion model, we apply the 3D steerable filter to improve the performance of current estimation algorithms such as the 3D Hough transform and the expectation maximization algorithm. We use an eigenvalue-analysis based multi-window strategy to detect and to eliminate *outliers* in occlusion estimation. This technique purifies input data and improves therefore the precision of the estimation results. Based on the spatial coherence in image sequences, we further use the "shift-and-subtract" technique to localize occlusion boundaries and to track their movement in occlusion sequences. This technique can be also used to distinguish occlusion from transparency and to decompose transparency scenes into multi-layers.

4.1 Introduction

In the computation of optical flow, estimation of multiple motions as well as tracking of occlusion boundaries are challenging problems. Due to the aperture problem we use the behavior in the neighborhood of a considered point. Such assumptions are either explicit in area-based techniques or implicit in filter-based schemes where the addressed neighborhood is the filter support. Conventional flow estimation methods are based on the single motion assumption (there is only one single motion inside the neighborhood) and the smoothness assumption (the motions are piecewise-smooth). For example, the well known brightness change constraint equation (BCCE) [Hor86] is based on these assumptions

$$I_x u + I_y v + I_t = 0, \tag{4.1}$$

where I_x, I_y, and I_t denote the spatio-temporal partial derivatives of the image intensity and (u,v) is the optical flow vector. These assumptions are violated if the neighborhood contains motion boundaries or multiple transparent motions. If we still apply the BCCE or its equivalent constraints to estimate the optical flow, we will not be able to obtain correct estimation results.

The study of optical flow has a long history. With respect to different criteria the related flow estimation algorithms may be divided into global or local techniques, one frame or multiple frame methods, spatial or spectral domain based approaches, probabilistic or non-probabilistic models, and so on. The reader is referred to [BB95, HS99] for a general survey of flow estimation methods. Here we are interested in techniques studying multiple motions including occlusion and transparency. In the following, we would like to review some related works.

Spatial Approaches

Nagel and Enkelmann [NE86] first addressed the violation of single motion model at motion boundaries. They used a regularization term to penalize motion discontinuities and thus rejected motion boundaries in the computation of optical flow. Weickert and Schnörr further extended this regularization term into spatio-temporal space [WS99]. Black and Anandan [BA96] treated occlusion regions similarly. They referred to the pixels near occlusion boundaries as *outliers* of the motion constraint and set lower weights to these pixels in the estimation. The concept of *outlier* comes from statistics. It means "a small amount of data points with large deviation from the bulk of all data points" [PTVF92]. This concept represents exactly the relationship between the pixels near occlusion boundaries and the pixels with a single motion: The spatio-temporal partial derivatives of the pixels with a single motion form a plane in the derivative space and the derivatives of the pixels near occlusion boundaries deviate from this plane due to motion discontinuities. Based on the concept of *outliers*, many probabilistic methods were proposed to model occlusion boundaries [JB93] and to estimate multiple motions near occlusion boundaries [Bou89, BBHP92, WA96].

Schunck considered occlusion boundaries as noise in the constraint line clustering [Sch89]. By noticing that motions have more components than noise, he applied a statistic method to cluster the dominant intersection of constraint lines for motion estimation. This statistic method was used in the Hough transform based approaches as well [BK94, NCN97].

A second group of approaches is based on segmenting regions where parametric models of flow can be fitted [WK93]. But the segmentation is in turn dependent on the optical flow. This forms a "chicken-and-egg" dilemma: We need appropriate segmentation to estimate optical flow accurately, while we need accurate optical flow to segment images properly. In order to bypass this dilemma, iterative methods are proposed to refine the region segmentation and motion estimation gradually [Hor86]. Bergen *et al.* [BBHP92] proposed an iterative method based on the "shift-and-subtract" strategy to estimate two motions. They subtracted pixels connected to one motion during the refinement of the parameters of the other motion and *vice versa*. It is known that iterative methods are sensitive to the choice of initial parameter values. In order to obtain appropriately initial values, they applied the pyramid algorithm [BYX83] in the estimation. Since at the coarsest level of the pyramid all motions are small due to the subsampling, they can simply set the initial values to zero and still guarantee the convergence of iteration. Irani *et al.* [IRP94] applied this iterative method to track objects even with non-consistent speeds. They used temporal integration to blur out uninterested regions and let the tracked object remain sharp.

The iteration principle was used in the statistic approaches as well. Researchers have recently elaborated algorithms based on the expectation maximization (EM) principle [DLR77]. Whereas the maximization step is the usual maximum likelihood estimation given the assignment of points to groups, the expectation step is regrouping the points by updating membership weights. Several authors [JB96, WA96] applied the EM algorithm on the BCCE (equation (4.1)) or on already computed flow vectors. This process is equivalent to fitting planes through the origin in the (I_x, I_y, I_t)-space. In fitting models to data points, it is known that the more number of groups we have, the better the fitting will be. Therefore, an image may tend to be divided into many small separated regions which actually should be merged to estimate one common motion. This forms the "overfitting" problem. In order to avoid the "overfitting", Weiss [Wei97] introduced smoothness constraint into each model group to avoid fitting data with high order polynomials. For the same reason, Ayer and Sawhney [AS95] applied the minimum description length (MDL) principle of the information theory [Ris83] to obtain the minimal number of motions.

In presenting explicit multiple motion models, many researchers have made contributions. Wang and Adelson represented multiple motions with a multi-layer model based on the local motion estimation [WA93]. This model is consistent with the daily knowledge naturally. But the local motion estimation technique they used is still based on the BCCE. Fleet *et al.* [FBJ98] proposed an explicit model of occlusion in the spatial domain. Then, they applied the steerability theory to detect occlusion boundaries as step edges in both components of the optical flow field. Black and Fleet further proposed to use the Bayesian framework to determine which pixels belong to the motion boundary regions [BF99]. With this explicit model they wished to predict the location of occlusion boundaries in the next frame exactly and therefore to exclude the corresponding boundary regions in the next estimation. Moreover, by tracking the movement of the boundary, they solved the foreground/background ambiguity [TMB85, GD94, BF99].

Spectral Approaches

Motion estimation is also addressed from the point of view of orientation analysis. Adelson and Bergen [AB85] pointed out that motion is equivalent to spatio-temporal orientation. They introduced a spatio-temporal energy model for single motion representation and proposed to use quadrature filters for motion estimation. This was the first optical flow algorithm based on the spectral analysis.

Bigün *et al.* connected the orientation analysis with symmetry detection [BG87, BGW91]. They pointed out that a single motion can be described as a linear symmetric image, whose spectrum is a line passing through the origin in the frequency domain. By using the principal axis analysis, they introduced a moment measure in the frequency domain to fix the orientation of the spectral line. In order to avoid the cumbersome discrete Fourier transform, they converted this moment measure from the frequency domain back to the spatial domain using the Parseval theory.

Jähne *et al.* used a 3D structure tensor [Knu82] to detect symmetry and to estimate motion [Jäh93, HJ97, JHS^{+}98]. Besides, they introduced a coherence measure to distinguish different kinds of motions such as single constant motion and motion discontinuity. This coherence measure is a function of the largest and the smallest eigenvalue of the structure tensor. For details about this coherence measure the reader is referred to [Jäh93].

The principal axis analysis was used for multiple motion estimation as well. Shizawa and Mase proposed a superposition principle, which assumed blindly that multiple motions are additive superposition of two single motions in the frequency domain or in the derivative space [SM91]. Then they applied the eigenvalue analysis for multiple motion estimation.

Resolution Issue of Principal Axis Analysis

The principal axis analysis is also called principal component analysis (PCA), Karhunen-Loève transform (KLT), and Hotelling transform in different literature. It decomposes a signal into an orthogonal basis using eigenvector analysis or singular value decomposition (SVD). The problem of principal axis analysis is that it is only suitable to detect one dominant orientation. For a signal in a 3D space, after the principal axis analysis, we obtain three eigenvalues satisfying $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Their corresponding eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ should be orthogonal to each other. That is to say, after finding out the dominant orientation denoted with $\mathbf{u}_1$, the other two eigenvectors $\mathbf{u}_2$ and $\mathbf{u}_3$ should lie in the plane perpendicular to $\mathbf{u}_1$, no matter what kind of structure the signal has. Though from different combinations of σ_1, σ_2, and σ_3 we can determine if we have a plane, a line, or some other spatio-temporal structures in this 3D space, we cannot provide more information using principal axis analysis. For example, we are interested in determining the orientation of two intersected planes in the 3D space. Both planes pass through the origin and the angle between them is arbitrary. If these two planes are perpendicular to each other, $\mathbf{u}_1$ will lie on the intersection of these two planes and the other two eigenvectors $\mathbf{u}_2$ and $\mathbf{u}_3$ are parallel to the normal vectors of these two planes, separately. Thus, we still can determine the orientation of these two planes. But if these two planes are not orthogonal to each other, though $\mathbf{u}_1$ still lies on the intersection of these two planes, $\mathbf{u}_2$ and $\mathbf{u}_3$ will be no more parallel to the normal vector of either plane. Thus, the principal axis analysis is no more reliable to determine orientation of multiple planes. In other words, the orientation resolution of the principal axis analysis is not sufficient to solve a non-orthogonal multiple orientation problem.

Gabor Based Approaches

In order to have sufficient orientation resolution, Heeger sampled the spectrum of the image sequence with twelve Gabor filters to determine the orientation of the motion plane in the frequency domain [Hee87]. Grzywacz and Yuille [GY90] further pointed out that the spectral support of a Gabor filter is a measure of uncertainty and the angle between two tangential lines of the support, which pass through the spectral origin, represents the uncertainty of orientation estimation. For simplicity we use 2D Gabor filters for demonstration and represent the spectral support of a filter with a circle. The angle between two tangential lines of the circle is a measure of angular uncertainty. This angle is desired to be the same for filters at different frequencies (figure 4.1). If we apply filters with constant scale, we will have larger angles at low frequencies than at high frequencies. In order to have the same orientation resolution, the spectral support of filters should be directly proportional to the frequency. This is exactly the property of Gabor wavelets [Mac91, Lee96, WB97, WFKvdM97, Wür97].

Heitger *et al.* proposed a variant of 2D Gabor filter, whose odd and even part have zero mean values [HRdH$^+$92]. They called this variation "stretched Gabor" filter. Similarly, Xiong and Shafer used hypergeometric filter to sample the spectrum for motion estimation [XS97]. The hypergeometric filter is based on one kind of special function called *confluent hypergeometric function* in the mathematical physics. It is very similar to Gabor filter in shape, but its DC component is zero. For details about the *confluent hypergeometric function* the reader is referred to [NU88].

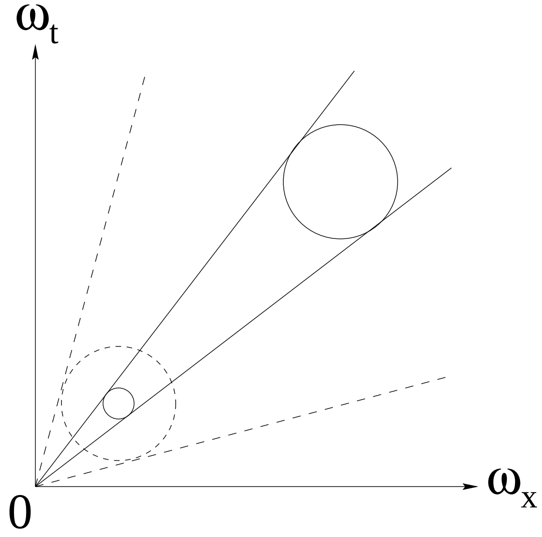


Figure 4.1: *The motivation of applying Gabor wavelets (redrawn from [GY90]). For simplicity we show a 2D example. In this image, we represent the spectral support of a Gabor filter with a circle. The angle between two tangential lines of the circle denotes the orientation resolution. Applying a set of filters with constant scale, we obtain coarser resolution at lower frequencies (as shown by the dashed circle and its tangential lines). It is desired that this resolution is the same when we apply a set of filters. Thus, all circles in one orientation should locate inside the space bounded by two tangential lines passing through the origin. Correspondingly, the spectral support of filters should be directly proportional to the frequency. This is exactly the property of Gabor wavelets.*

One main concern of using Gabor/hypergeometric filter based approaches is the enormous complexity of computation in sampling the spectral domain with fine resolution. Taking this concern into account, we understand clearly why Heeger used only twelve Gabor filters to sample the whole energy spectrum, though the corresponding orientation resolution is still very coarse.

In light of steerability approaches, we studied whether we can steer the Gabor filter to deal with the enormous complexity of computation [OT98]. In short words, using SVD we can steer a set of Gabor filters having isotropic shapes (i.e. they are not elongated) with respect to their positions (For details about the SVD the reader is referred to [PTVF92]). By combining Gabor filters with different scales we may maintain the same orientation resolution in sampling. However, Grzywacz and Yuille pointed out that, in the filter responses of Gabor wavelets, there is a positive skewness [GY90], which obstructs the application of Gabor wavelets in multiple orientation analysis. In the following, we explain the root of the skewness.

Skewness of Gabor Wavelets

For simplicity we use 2D Gabor filters. We define a 2D Gabor filter with variables x and t as follows

$$g_a(x,t;\omega_{x0},\omega_{t0},\sigma_x,\sigma_t) = \frac{1}{2\pi\sigma_x\sigma_t}\exp\{-(\frac{x^2}{2\sigma_x^2}+\frac{t^2}{2\sigma_t^2})+i(\omega_{x0}x+\omega_{t0}t)\}, \tag{4.2}$$

where $(\omega_{x0},\omega_{t0})$ denotes its central frequency components. Its scale parameters are denoted with σ_x and σ_t. The corresponding spectrum is described in the equation below with ω_x and ω_t as variables

$$G_a(\omega_x,\omega_t;\omega_{x0},\omega_{t0},\sigma_x,\sigma_t) = \exp\{-\frac{\sigma_x^2(\omega_x-\omega_{x0})^2+\sigma_t^2(\omega_t-\omega_{t0})^2}{2}\}. \tag{4.3}$$

It is worth mentioning that we observe the result of applications not in the frequency domain, but in the parameter space, i.e. in the space coordinated with ω_{x0} and ω_{t0}. Therefore, we should observe the spectrum with respect to the parameters ω_{x0} and ω_{t0}.

If σ_x and σ_t are constant, the spectrum is simply a Gaussian function of ω_{x0} and ω_{t0} and there is no skewness. But we prefer the wavelet property: The scale of a Gabor filter should be inversely proportional to its central frequency parameter so that its spectral support is directly proportional to the frequency. Thus, we have

$$\left\{\begin{array}{lcl}\sigma_x & = & \frac{C}{\omega_{x0}}\\ \sigma_t & = & \frac{C}{\omega_{t0}}\end{array}\right., \tag{4.4}$$

where C is a constant. Then the corresponding spectrum of Gabor wavelets yields

$$G_a(\omega_x,\omega_t;\omega_{x0},\omega_{t0},C) = \exp\{-\frac{C^2}{2}[(\frac{\omega_x-\omega_{x0}}{\omega_{x0}})^2+(\frac{\omega_t-\omega_{t0}}{\omega_{t0}})^2]\}. \tag{4.5}$$

When we interpret this expression with respect to parameters ω_{x0} and ω_{t0}, this expression is no more a Gaussian function because there are also ω_{x0} and ω_{t0} in the denominator of the exponential function. Thus, the skewness appears. This is shown in figure 4.2 in detail.

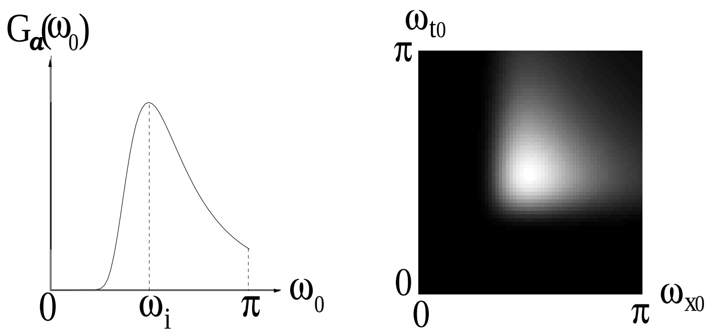


Figure 4.2: *The skewness of the Gabor wavelet. The spectrum is no more a Gaussian function of the filter parameters ω_{x0} and ω_{t0}.* **Left:** *1D section of the skewness (redrawn from [GY90]).* **Right:** *2D version of the skewness. $C = 3.5, \omega_x = \omega_t = \frac{\pi}{2}$.*

In figure 4.2 we observe that, if there is only one single motion, the maximum is still well localized at $(\omega_{x0}, \omega_{t0})$ in spite of the skewness since we do not have overlapping. But if we have multiple planes, the overlapping of different filter responses, specially the overlapping of skewness will disturb the locations of maximal values. As a result, the extracted orientation information will be no more convincible.

Our Contribution

Though the skewness obstructs the application of Gabor wavelets in multiple motion analysis, we find the idea of sampling the orientation locally is still attractive since it achieves higher orientation resolution than principal axis analysis and current 3D steerable filters (see chapter 3 for detail). According to our observation, current Gabor based approaches are still derived from the idea of complete signal representation and sample the whole spectrum directly instead of the orientation, while the orientation information is sufficient for multiple motion analysis. Based on this analysis and the consideration of maintaining the same orientation resolution, we proposed our 3D orientation steerable filter in chapter 3. In this chapter, we will apply this filter for multiple motion estimation.

This chapter is constructed as follows: The following section studies multiple motions in detail for a better understanding. Then, we compare the spatial motion model and the spectral motion model in section 3. After that, we use our orientation steerable filter for multiple motion estimation in section 4. Section 5 further discusses the multi-window strategy in occlusion estimation and the precision improvement of estimation results after eliminating *outliers*. Later on we use the “shift-and-subtract” technique to localize and to track motion boundaries in occlusion sequences in section 6. In section 7 we show application examples. Finally, this chapter is concluded in section 8 with some discussions.

4.2 Understanding Multiple Motions

Multiple motions may be divided into occlusion and transparency. In this section, we explain that occlusion is equivalent to multiple planes both in the (I_x, I_y, I_t) space and in the frequency space and transparency is equivalent to multiple planes *only* in the frequency space. Correspondingly, multiple motion analysis is equivalent to orientation analysis of multiple planes.

Spatial Observation of Multiple Motions

We start with an illustration of the difference between occlusion and transparency in figure 4.3. First, by isolating a spatial window, we observe that occlusion is more local than transparency. In case of transparency the entire window contains two motions. Second, the transparency is the addition of two models, whereas the occlusion involves a step-function [Niy95]. This difference makes it difficult to describe both kinds of multiple motions with a unified model in the spatial domain. Thus, we turn to the frequency domain to look for the solution.

Spectral Analysis of Occlusion

One motivation of studying multiple motions in the frequency domain is to provide an explicit model which describes both occlusion and transparency in a uniform manner. The underlying theoretical framework relies on spectral analysis and was first presented in [BB97] based on observations in [FL94]. Another framework for multiple motions formulated in the frequency domain is the superposition principle of Shizawa and Mase [SM91] which, however, does not separate occlusion from transparency.

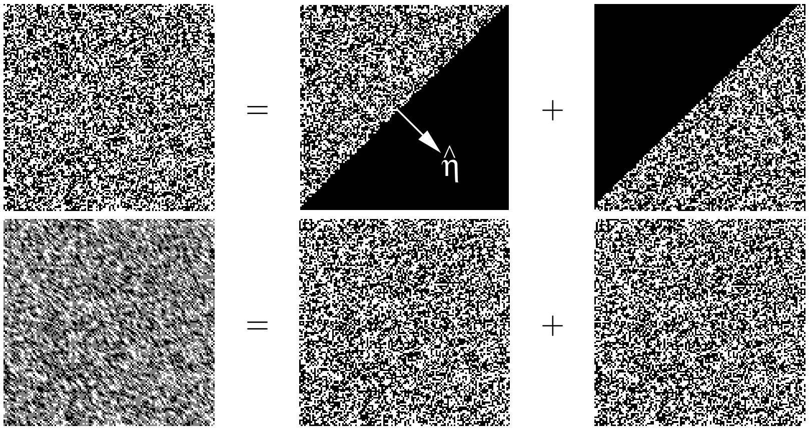


Figure 4.3: *Although occlusion and transparency can be decomposed into multiple layers, they are based on different decomposition principles. Here random dot regions represent motion and dark regions denote static status.* **Top:** *One frame of the occlusion sequence is decomposed into two layers by a Heaviside unit step function (equations (4.6) and (4.7)). There is motion discontinuity only at the boundary. The occluding signal is moving with the speed* $(1, 1)$ *[pixel/frame] and the occluded signal with* $(1, -1)$ *[pixel/frame]. We use* $\hat{\eta}$ *to denote the unit vector normal to the occluding boundary.* **Bottom:** *One frame of the transparency sequence is a simple superposition of two layers (equation (4.13)). Multiple motions exist in the entire window. The speeds of multiple motions are* $(1, 1)$ *[pixel/frame] and* $(1, -1)$ *[pixel/frame] as well.*

The spectrum of multiple motions was first analyzed by Fleet and Langley [FL94]. Assuming that an occlusion boundary is a characteristic function $\chi(\mathbf{x})$, they model the occlusion in the spatial domain as follows:

$$I(\mathbf{x}, t) = \chi(\mathbf{x} - \mathbf{v}_1 t) I_1(\mathbf{x} - \mathbf{v}_1 t) + [1 - \chi(\mathbf{x} - \mathbf{v}_1 t)] I_2(\mathbf{x} - \mathbf{v}_2 t), \qquad (4.6)$$

where $I_1(\mathbf{x})$ is a 2D **occluding** signal moving with velocity $\mathbf{v}_1 = (v_{1x}, v_{1y})^T$ and $I_2(\mathbf{x})$ is a 2D **occluded** signal moving with velocity $\mathbf{v}_2 = (v_{2x}, v_{2y})^T$.

Beauchemin and Barron [BB97, BB00] were the first who formulated an exact model in the frequency domain. They modeled the occlusion in the spatial domain with a Heaviside unit step function $U(\mathbf{x})$ for $\chi(\mathbf{x})$:

$$U(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}^T \hat{\eta} \geq 0 \\ 0 & \text{otherwise} \end{cases}, \qquad (4.7)$$

where x denotes 2D spatial Cartesian coordinates and $\hat{\eta}$ is a unit vector normal to the occluding boundary (figure 4.3).

We denote the spatial frequency vector as $\kappa = (\omega_x, \omega_y)^T$ and the temporal frequency as ω_t. Then, the Fourier transform of the image sequence reads

$$\begin{aligned}\tilde{I}(\kappa, \omega_t) \; &= \tilde{U}(\kappa)\delta(\kappa^T \mathbf{v}_1 + \omega_t) * \tilde{I}_1(\kappa)\delta(\kappa^T \mathbf{v}_1 + \omega_t) + \tilde{I}_2(\kappa)\delta(\kappa^T \mathbf{v}_2 + \omega_t) \\ &\quad -\tilde{U}(\kappa)\delta(\kappa^T \mathbf{v}_1 + \omega_t) * \tilde{I}_2(\kappa)\delta(\kappa^T \mathbf{v}_2 + \omega_t), \qquad (4.8)\end{aligned}$$

where $*$ means convolution and $\tilde{}$ denotes the Fourier transform of the corresponding signal. The spectrum of the Heaviside unit step function is given by

$$\tilde{U}(\kappa) = 2\pi[\pi\delta(|\kappa|) + \frac{\delta(\kappa^T \hat{\eta}_\perp)}{i\kappa^T \hat{\eta}}], \qquad (4.9)$$

where $\hat{\eta}_\perp$ denotes a unit vector perpendicular to $\hat{\eta}$. Taking the properties of the impulse function into account, we obtain (see Appendix B for detail)

$$\begin{aligned}\tilde{I}(\kappa, \omega_t) \; &= \; [2\pi^2 \tilde{I}_1(\kappa) + A(\kappa)]\delta(\kappa^T \mathbf{v}_1 + \omega_t) \\ &\quad +(1 - 2\pi^2)\tilde{I}_2(\kappa)\delta(\kappa^T \mathbf{v}_2 + \omega_t) + B(\kappa, \omega_t), \qquad (4.10)\end{aligned}$$

with

$$A(\kappa) = \frac{2\pi}{i\kappa^T \hat{\eta}}\delta(\kappa^T \hat{\eta}_\perp) * \tilde{I}_1(\kappa), \qquad (4.11)$$

$$B(\kappa, \omega_t) = \frac{2\pi}{i\kappa^T \hat{\eta}}\delta(\kappa^T \hat{\eta}_\perp)\delta(\kappa^T \mathbf{v}_1 + \omega_t) * \tilde{I}_2(\kappa)\delta(\kappa^T \mathbf{v}_2 + \omega_t). \qquad (4.12)$$

The first two terms of expression (4.10) are two oriented planes passing through the origin of the frequency space. Their normal vectors, namely $(u_1, v_1, 1)$ and $(u_2, v_2, 1)$, contain the velocities of the two signals. The second term is exactly the spectrum of the occluded signal, but the first term contains an additional distortion term $A(\kappa)$ on the plane of the occluding spectrum. However, here we are interested in the orientation of the plane and the term $A(\kappa)$ does not disturb the orientation. Actually, $A(\kappa)$ strengthens this spectral plane. Therefore, we do not consider it as distortion. The main discriminating term is the third one, $B(\kappa, \omega_t)$, which is a 3D volume filling the entire frequency space. We observe that this distortion term depends on the normal of the occluding boundary as well as *both* speeds of the occluding and the occluded signal.

If the energy of the distortion term $B(\kappa, \omega_t)$ is very high, we will not be able to recognize the two planes. Fortunately, the critical factor in the amplitude of

$B(\kappa, \omega_t)$ is the hyperbolic term $\frac{2\pi}{i\kappa^T\hat{\eta}}$, which reduces very quickly with the increase of $|\kappa|$. In most regions of the spectral domain, say for $|\kappa| \geq 1$, the amplitude of the distortion is much less than that of signals. Therefore, we may consider the spectrum only above a lower bound of the frequency and identify the two dominant planes.

Spectral Analysis of Transparency

Transparency is a special case of occlusion since we can simply substitute $\chi(\mathbf{x} - \mathbf{v}_1 t)$ with a real constant $a (a \in (0,\ 1))$ yielding

$$I(\mathbf{x}, t) = aI_1(\mathbf{x} - \mathbf{v}_1 t) + (1 - a)I_2(\mathbf{x} - \mathbf{v}_2 t). \tag{4.13}$$

The corresponding spectrum is then characterized by two oriented planes without distortion

$$\tilde{I}(\kappa, \omega_t) = a\tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1 + \omega_t) + (1 - a)\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2 + \omega_t). \tag{4.14}$$

Spectral Model of Multiple Motions

Though in the case of occlusion there exists a distortion term, the main energy proportion is on the two spectral planes due to the hyperbolic nature of the distortion term. In addition, we may low-stop the energy spectrum to abandon the strong disturbance at low frequencies. Thus, we propose a unified model for both occlusion and transparency in the spectral domain: Both occlusion and transparency are characterized as multiple spectral planes passing through the origin. The corresponding motion speeds are described by the normal vectors of these planes.

This model can be viewed as a generalization of the spatiotemporal energy model of single motion [AB85, Hee87]. At first sight it is very similar to the work of Shizawa and Mase [SM91]. But there are two different points in our work:

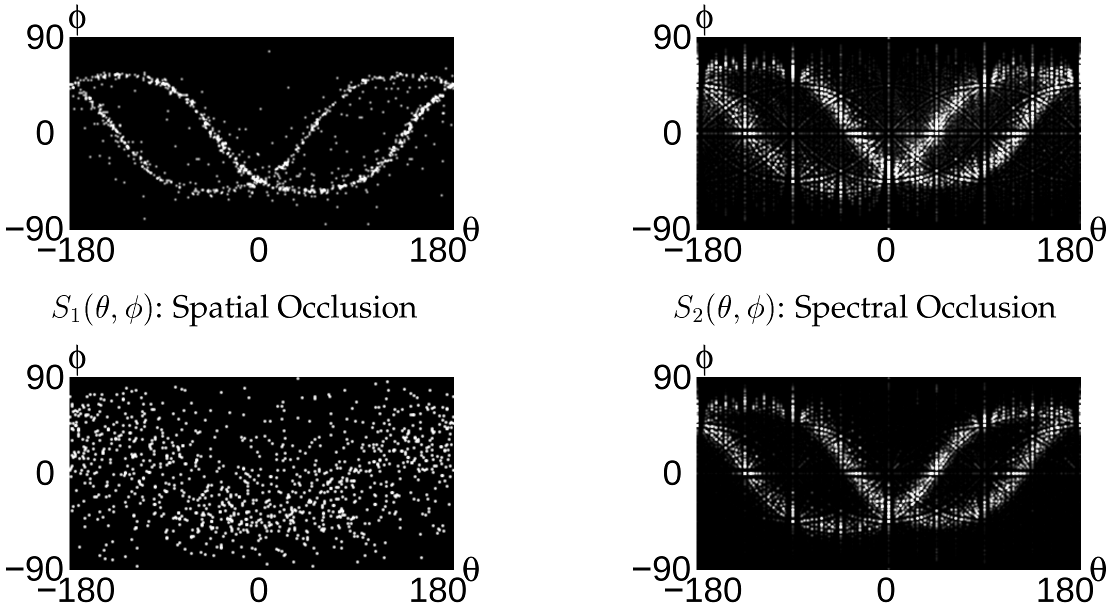
- Shizawa and Mase proposed that multiple motions are characterized as multiple planes both in the (I_x, I_y, I_t)-space and in the frequency domain. We argue that even the model is correct in the (I_x, I_y, I_t)-space, it is not feasible since we cannot estimate I_x, I_y, and I_t properly in case of transparency.

- At low frequencies, multiple planes are disturbed by the distortion term of the occlusion. We have to truncate low frequency components in order to fit multiple planes robustly.

4.3 Comparison between Spatial and Spectral Model

According to the analysis in the last section, the spectral model can describe both occlusion and transparency, while the spatial model can describe only occlusion. This is because the spatial model is very much dependent on the correct estimation of I_x, I_y, and I_t. In case of occlusion, most pixels in the image may have the correct I_x, I_y, and I_t due to the local property of motion boundaries (see figure 4.3). But in case of transparency, we can hardly assume that the intensity profile is differentiable. If we do not have *a priori* knowledge about motions, we cannot assume that all multiple motions are occlusion. In this situation, we should use the spectral model.

It should be noticed that, though we have a thorough analysis of the spectral motion model, there exists a severe problem in obtaining the energy spectrum of the image sequence due to the block effect of the discrete Fourier transform (DFT). This is actually one main problem in the frequency-based techniques. In order to avoid the block effect of DFT, we take a local Fourier transform (LFT), i.e. Gaussian windowed DFT. According to the convolution theorem in the Fourier analysis, Gaussian windowed DFT of the image sequence is equivalent to the convolution between the spectrum of the image sequence and the Gaussian function. Consequently, the spectrum is blurred after the LFT. Besides, the high frequency components leak from one side of the spectrum to the other side due to the periodic property of DFT. As a result, the spectral multiple motion model has worse resolution than the spatial multiple motion model. For compensation, we have to enlarge the window size so that we can improve the frequency resolution with finer interval. But this compensation has also limitations. First, using a larger window means a pixel has more neighbors in a frame. The constant motion assumption in this enlarged neighborhood is more fragile. Second, using a larger window means including more frames in the estimation. On one side, using multiple frames improves the robustness of the optical flow estimation algorithm (e.g. [Jäh93]). On the other side, if we include a very long sequence into estimation, we can hardly assume that the motion is constant over so large time interval.



$S_1(\theta, \phi)$: Spatial Occlusion

$S_2(\theta, \phi)$: Spectral Occlusion

$S_3(\theta, \phi)$: Spatial Transparency

$S_4(\theta, \phi)$: Spectral Transparency

Figure 4.4: *Orientation signatures of occlusion and transparency sequence in figure 4.3. Both occlusion and transparency have the same motion parameters.* **Top Left:** *The orientation signature $S_1(\theta, \phi)$ of the occlusion sequence based on the spatial multiple motion model. We use a $33 \times 33 \times 1$ window to obtain this signature. In the signature we observe* outliers *outside two main curves.* **Top Right:** *The signature $S_2(\theta, \phi)$ of the occlusion sequence based on the spectral multiple motion model. The structure is the same as that in $S_1(\theta, \phi)$ except that the curves are blurred. We apply band pass filter to reduce the high frequency aliasing as well as the low frequency disturbance. We use a $32 \times 32 \times 32$ window to obtain this signature.* **Bottom Left:** *The signature $S_3(\theta, \phi)$ of the transparency sequence based on the spatial model. The distribution of points is nearly random.* **Bottom Right:** *The signature $S_4(\theta, \phi)$ of the transparency sequence based on the spectral model. Comparing $S_4(\theta, \phi)$ with $S_2(\theta, \phi)$, we observe that the spectral model describes both occlusion and transparency in a uniform manner. Besides, the distortions outside the main curves in $S_2(\theta, \phi)$ disappear in $S_4(\theta, \phi)$.*

We confirm these claims with an example. In figure 4.4 we compare the orientation signatures of both occlusion and transparency using spatial and spectral motion model. These orientation signatures are obtained by applying our 3D steerable filter on the derivative or the energy spectrum of the image sequences in figure 4.3. Note that multiple planes in both derivative space and energy spectrum pass through the origin. Therefore, it is very easy to set the origin as the keypoint for the application of our 3D steerable filter. This is a great advantage compared to using steerable filter for 2D junction characterization, where we have to locate the keypoints at first. Since both occlusion and transparency sequences have the same motion parameters, it is desired that their orientation signatures have the same curves. A comparison between row 1 and row 2 shows that the spectral model can treat both occlusion and transparency, while the spatial model can treat only occlusion. This figure confirms our spectral analysis of occlusion and transparency as well. In $S_2(\theta, \phi)$ we observe distortions outside two main curves, while in $S_4(\theta, \phi)$ these distortions disappear. Besides, a comparison between $S_3(\theta, \phi)$ and $S_1(\theta, \phi)$ confirms that the spectral model has coarser resolution than the spatial model since the spectrum is blurred by LFT. Though we may avoid the use of LFT, we still need many frames by sampling the spectrum with Gabor filters and by using phase-based technique (e.g. [FJ90]) for an exact estimation. For example, here we have to use a $32 \times 32 \times 32$ window to obtain the orientation signature in the spectral domain, in which the resolution is still very coarse. Taking into account that the constant motion assumption will be very often violated in such a large window, we prefer to go back to the spatial model for occlusion analysis.

4.4 Multiple Motion Estimation Using 3D Steerable Filter

In this section, we apply our 3D steerable filter in the estimation of multiple motions and compare our 3D steerable filter with current estimation algorithms. After applying our 3D filter, we obtain an orientation signature. For parameter extraction or signal segmentation, we still need further processing like the EM algorithm. Since the 3D Hough transform based on equation (3.8) as well as the planar EM algorithm can extract the orientation parameters of planes **directly**, we face the following question in the application of 3D steerable filter: Why do we project the 3D data onto the 2D feature space before extracting parameters? In order to answer this question, we must analyze the 3D Hough transform and the EM algorithm in more detail.

3D Hough Transform Based Algorithm

The Hough transform is a sampling and searching method for parameter extraction. Concretely, in equation (3.8) we like to extract the normal vectors (n_1, n_2, n_3) from a set of points coordinated with $(I_{ix}, I_{iy}, I_{it})(i = 1, \cdots, N)$. For each point (I_{ix}, I_{iy}, I_{it}), we draw the corresponding vectors in the (n_1, n_2, n_3) space satisfying the equation

$$I_{ix}n_{1j} + I_{iy}n_{2j} + I_{it}n_{3j} = 0, \qquad (j = 1, \cdots) \tag{4.15}$$

where (n_{1j}, n_{2j}, n_{3j}) denotes the j-th vector normal to (I_{ix}, I_{iy}, I_{it}). After going through all points, we search in the (n_1, n_2, n_3) space the position with the maximal number of intersections. From the corresponding coordinates (n_{1m}, n_{2m}, n_{3m}) we obtain then the desired motion parameters

$$\left\{ \begin{array}{lcl} u_m & = & \frac{n_{1m}}{n_{3m}} \\ v_m & = & \frac{n_{2m}}{n_{3m}} \end{array} \right. . \tag{4.16}$$

In practice, we sample the parameter space with a finite interval and relax the orthogonal criterion with a positive threshold ε yielding replace the above equation (4.15) with

$$\mid I_{ix}n_{1j} + I_{iy}n_{2j} + I_{it}n_{3j} \mid \leq \varepsilon. \tag{4.17}$$

The equation (4.17) based 3D Hough transform is equivalent to a 3D filter with the concave disk shape centered at the origin of the 3D space (figure 4.5). This disk is actually a collection of *relaxed* normal vectors of all possible planes containing the vector (I_{ix}, I_{iy}, I_{it}). Comparing our filter shape (figure 3.1) with the shape of this disk filter we conclude that our filter samples the orientation space more efficiently than the 3D Hough transform. This conclusion is also confirmed by the comparison between our filter response of a plane (figure 3.4) and the Hough image of a point, which is actually the impulse response of the concave disk filter in 3D space (figure 4.6). It is interesting that the Hough image of a 3D point is very similar to our steerable filter response of a 3D plane except that the Hough image has no negative ϕ value (we only use normal vectors with $n_3 > 0$). Taking into account that our filter response of a 3D plane consists of filter responses of a lot of points we may confirm the above conclusion easily. This efficiency enables our filter to reduce the enormous memory requirement in Hough based approaches

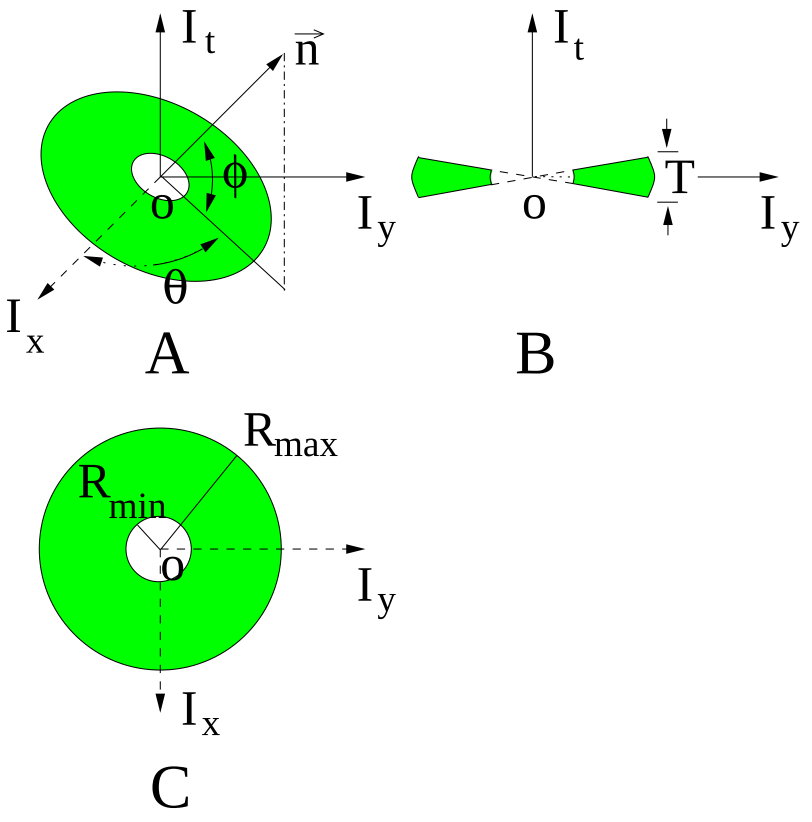


Figure 4.5: *The 3D Hough transform is equivalent to a filter with a concave disk shape.* **A:** *General projection drawing of the filter mask. The vector* **n** *is normal to the filter mask.* **B:** *Side view of the filter mask. The angular thickness T of the disk is determined by the clustering threshold ε in equation (4.17).* **C:** *Vertical view of the filter mask.*

[XOK90], especially the gigantic overlapping of the Hough curves (figure 4.6). As a result, we can extract the parameters of motion planes with much less complexity.

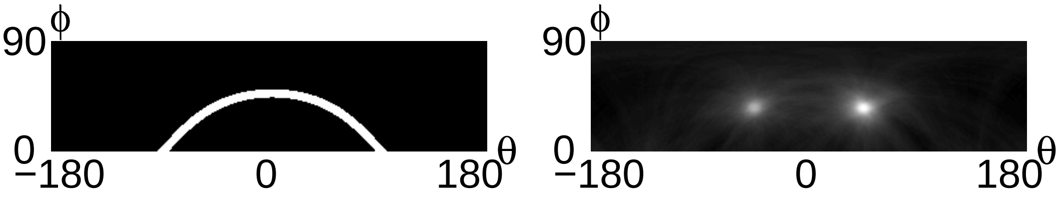


Figure 4.6: **Left:** *All unit vectors satisfying equation (4.17) form a curve similar to our filter response of a 3D plane in the (θ, ϕ) space. The width of the curve is determined by the clustering threshold ε in equation (4.17).* **Right:** *The Hough image of the random dot occlusion sequence in figure 4.3.*

Another point is that the intersections of different curves in the Hough image are blurred due to the introduction of ε, as shown in figure 4.6. Consequently, the global maximal position is no more a peak, but a smooth *mono-modal* distribution. Though the search of the global maximal position is still feasible, the search of the second maximal position is generally problematic because the properties of the *mono-modal* distribution are unknown and we do not know how to get rid of the neighbors of the global maximum automatically in searching the second maximum. We have such an example in figure 4.6, where we display the Hough image of the random dot occlusion sequence shown in figure 4.3. After finding out the global maximal position $(\theta_{n1}, \phi_{n1}) = (46°, 36°)$ as the first normal vector and extracting the corresponding speed $(u_1, v_1) = (0.956, 0.990)$, we do not know how to get rid of the neighbors of this global maximal position *automatically*. In order to test if the second maximal position is correct, we cut out the neighbors of the global maximum by setting the region with $\theta > 0$ in the Hough image to zero and further search the maximal position. This time we obtain two points with $(\theta_{n2}, \phi_{n2}) = (-46°, 36°)$ and $(\theta_{n3}, \phi_{n3}) = (-45°, 36°)$. Correspondingly, the second motion has two possible parameters, $(u_2, v_2) = (0.956, -0.990)$ or $(u_2, v_2) = (0.973, -0.973)$, and we do not know which one is the desired motion parameter. This problem is even worse, when these two maxima locate near each other.

This problem is easier to solve in our filter response. Though it is not easy to find out the extreme points *exactly* in the orientation signature due to noise or inadequate number of points, as we mentioned in chapter 3, we can get appropriately initial values using the following facts: First, the orientation signature of a plane is a periodic function. Second, the extreme point (θ_m, ϕ_m) (see equation (3.10) for reference) has its θ_m located in the middle of two zero-crossing points on the θ axis, while these two zero-crossing points should have a distance of π. Thus, we can simply detect the number of zero-crossing points on the θ axis in the orientation signature and analyze their relations to determine the number of motions. The θ value of the middle point between two corresponding zero-crossing points is the desired θ_m. Then we can search the ϕ_m along the ϕ direction starting from θ_m. After obtaining appropriately initial values of (θ_m, ϕ_m), we may use the EM algorithm for further refinement of motion parameters.

According to above descriptions we have the following remarks.

- Our 3D steerable filter is similar to the 3D Hough transform in the sense that both methods sample the parameter space in the implementation.
- Our filter samples the parameter space more efficiently than the Hough transform because the impulse response of our filter has much smaller support than the Hough image of a point. This advantage is reinforced in obtaining the responses of 3D planes, in which our approach works like mosaicing different impulse responses with overlapping only among neighboring responses, while the overlapping effect of the Hough images of points is much stronger.
- In Hough image the search of the second maximal position is in general a problem. In our approach we find the second maximal position or its appropriate approximation easily. Of course we still need further refinement of desired parameters.

Spatial and Spectral Expectation Maximization Algorithm

The EM algorithm consists of subsequent iterations of the expectation and maximization step until there is no significant difference in the parameter estimates. In the expectation step, the membership weights of points are updated by the new results of parameter estimation; in the maximization step, we use the usual maximum likelihood method to estimate parameters with the updated assignment of points to groups.

In the spatial EM algorithm based on equation (4.1), we choose arbitrarily initial values (u_{10}, v_{10}) and (u_{20}, v_{20}). In the expectation step, we assign the weights W_{i1} and W_{i2} to the i-th point as follows according to the corresponding squares of the residuals (see [JB93] for details)

$$\left\{ \begin{array}{lcl} W_{i1} & = & \frac{1}{1+e^{-\frac{R_{i2}-R_{i1}}{\sigma_r^2}}} \\ W_{i2} & = & \frac{1}{1+e^{-\frac{R_{i1}-R_{i2}}{\sigma_r^2}}} \end{array} \right. , \tag{4.18}$$

with

$$\left\{ \begin{array}{lcl} R_{i1} & = & (I_{ix}u_1 + I_{iy}v_1 + I_{it})^2 \\ R_{i2} & = & (I_{ix}u_2 + I_{iy}v_2 + I_{it})^2 \end{array} \right. , \tag{4.19}$$

where σ_r is a parameter to adjust the tolerance level of the residual. The weights are simple applications of the Bayes rule, which give the membership probability of every point.

In the maximization step, we solve the following two linear systems in order to update (u_1, v_1) and (u_2, v_2), where the indices $i1$ and $i2$ run over all points

$$\begin{pmatrix} \vdots & \vdots & \vdots \\ W_{i1}I_{ix} & W_{i1}I_{iy} & W_{i1}I_{it} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0, \tag{4.20}$$

$$\begin{pmatrix} \vdots & \vdots & \vdots \\ W_{i2}I_{ix} & W_{i2}I_{iy} & W_{i2}I_{it} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0. \tag{4.21}$$

In this spatial model we convolve the cube with the first spatiotemporal derivatives of a 3D Gaussian function to obtain I_{ix}, I_{iy}, and I_{it}.

In using the spectral motion model, we take a LFT to avoid the block effect of DFT and further band-pass the energy spectrum to truncate the distortion at low frequencies and the aliasing at high frequencies. Here we simply use an ideal band-pass filter with stop frequencies $|\omega_{low}| = \frac{\pi}{4}$ and $|\omega_{high}| = \frac{3\pi}{4}$. Then we obtain a set of data points in the spectral domain associated with the amplitude of the Fourier transform. Their amplitudes can be viewed as a mass density in fitting the plane to these points. We denote with A_i the amplitude of the i-th point in the data set. The spectral EM algorithm is similar to the spatial EM algorithm. We only need to build the new residuals

$$\begin{cases} R'_{i1} & = & A_i^2(\omega_{ix}u_1 + \omega_{iy}v_1 + \omega_{it})^2 \\ R'_{i2} & = & A_i^2(\omega_{ix}u_2 + \omega_{iy}v_2 + \omega_{it})^2 \end{cases}. \tag{4.22}$$

By replacing R_{i2} and R_{i2} in equations (4.19) with R'_{i1} and R'_{i2}, we build the corresponding weights W'_{i1} and W'_{i2} in the spectral domain. Correspondingly, the linear system turns out to be

$$\begin{pmatrix} \vdots & \vdots & \vdots \\ W'_{i1}\omega_{ix} & W'_{i1}\omega_{iy} & W'_{i1}\omega_{it} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0, \tag{4.23}$$

$$\begin{pmatrix} \vdots & \vdots & \vdots \\ W_{i2}^{'}\omega_{ix} & W_{i2}^{'}\omega_{iy} & W_{i2}^{'}\omega_{it} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0. \tag{4.24}$$

Since the EM algorithm is an iterative method, it has no closed-form solution. Generally, we do not know the number of motions exactly. In order to fix the number of motions, Weiss introduced the smoothness constraint into the motion model [Wei97] to avoid the "overfitting" problem. Ayer and Sawhney [AS95] and Gu *et al.* [GSA96] applied the minimum description length (MDL) principle [Ris83] to obtain the minimal number of motions. These approaches fix the number of motions implicitly. Actually, we may detect the number of motions directly in our filter responses by analyzing the zero-crossing points on the θ axis. Besides, the convergency and the robustness of the EM algorithm are very much dependent on the initial values. We also can improve the setting of initial values by analyzing curves in our filter responses, as we will show in the experiment section.

In summary, though the 3D steerable filter does not extract parameters or segment signals directly, it reduces the data dimension. This makes the access of parameters easier and results in the improvement of the performance of current algorithms.

4.5 Outlier Issue in Occlusion Estimation

Current probabilistic approaches include the *outliers* in the estimation. This makes the estimation fragile, especially if the number of *outliers* is comparable to the number of pixels with a single motion since probabilistic methods are purely based on statistics. Our motivation is to improve the quality of input data before extracting motion parameters. According to our observation this is possible by combining current techniques.

Detection of Outliers

We assume that the motions in image sequences are piecewise-smooth with possible occlusion. In the spatio-temporal derivative space we observe the following relations according to [Jäh93]:

- For a single constant translational motion we have a plane with a normal vector parallel to $(u, v, 1)$, where (u, v) denotes the optical flow vector. The eigenvalues of this plane satisfy
$$\sigma_1 \geq \sigma_2 > \sigma_3 = 0. \tag{4.25}$$

- For a single constant motion having the aperture problem, the plane above degenerates into a line whose corresponding eigenvalues satisfy
$$\sigma_1 > \sigma_2 = \sigma_3 = 0. \tag{4.26}$$

- For occlusion we observe multiple planes plus distortions [YDBS99] with three positive eigenvalues
$$\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0. \tag{4.27}$$

Thus, we can judge if there are multiple motions from different combinations of eigenvalues even without knowing motion parameters. In practice, the eigenvalues may deviate from their standard values due to noise or derivative approximation error. Thus, instead of checking if $\sigma_3 = 0$ we set a threshold λ_{31} for *outlier* detection. If $\sigma_3 > \lambda_{31}\sigma_1$, we conclude that there are multiple motions. We may also check the aperture problem by defining another threshold λ_{21} between σ_2 and σ_1. Here we set $\lambda_{31} = \lambda_{21} = 0.2$.

In case of occlusion, if we can purify multiple planes from distortions (i.e. *outliers*), we may improve the precision of estimation results. The remaining question is how to detect these *outliers*. We observe that if we have occlusion in a window, the occlusion boundaries should locate in this window as well, though we do not know their exact positions. Based on this observation, we use a multi-window strategy to eliminate *outliers* before estimation. We detect occlusion regions using eigenvalue analysis inside small windows and mark these regions as *outliers*. In a large window containing these small windows the pixels outside *outlier* regions are guaranteed to be *normal* pixels. Using only these *normal* pixels for estimation, we avoid the disturbance of *outliers* and improve therefore the precision of estimation results in the large window.

It should be noticed that we also abandon some *normal* pixels by marking *outliers* inside small windows. Therefore, we prefer to reduce the size of the small window so that this loss is as small as possible. On the other side, in order to provide robust eigenvalue analysis, we must have an adequate number of pixels in the small window. To solve this conflict, we limit the spatial size of the small window but extend its temporal size to include pixels from other frames as well (e.g. from frames $(t_0 - 1)$ and $(t_0 + 1)$, where t_0 denotes the current frame).

In order to verify that there are still adequate pixels remaining, we define a reliability measure which is a ratio between the number of pixels remaining and the total number of pixels in the window

$$r_m := \frac{\mathcal{N}_i}{\mathcal{N}_{all}}, \quad (i = 1, 2) \tag{4.28}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ denote the number of remaining pixels of the occluding and occluded signal. If either of these two ratios is below a threshold, we have to enlarge the window to include more pixels for estimation.

Precision Improvement after Eliminating Outliers

In figure 4.7 we show the result of *outlier* detection in the random dot occlusion sequence (figure 4.3), where the occluding signal moves with a speed of (1,1) [pixel/frame] and the occluded signal with a speed of (1,-1) [pixel/frame]. We show also the orientation signatures before and after eliminating *outliers*. We can see that after eliminating *outliers* the curves in the (θ, ϕ) space are more clearly. Consequently, we obtain better estimation results (see table 4.1). In order to analyze the effect of window size in the estimation, we reduce the window size from 33×33 to 17×17. In the 17×17 window, the number of *outliers* is liable to be comparable to the number of *normal* pixels. As a result, the disturbance of *outliers* increases strongly. In contrast, if we eliminate *outliers* before estimation, we can still obtain reasonable results.

4.6 Scene Analysis

After obtaining multiple motion parameters in the occlusion boundary regions, we would like to localize boundaries in one frame and further track their move-

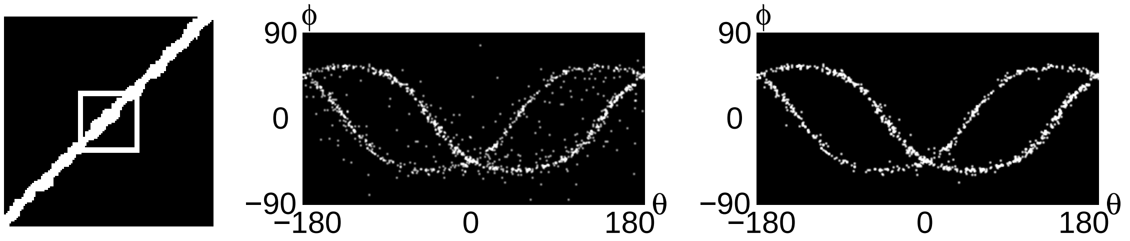


Figure 4.7: **Left:** *Marked* outliers *in the random dot occlusion sequence in figure 4.3 after eigenvalue analysis using a* $5 \times 5 \times 3$ *window. The white box here shows the estimation window across the occlusion boundary.* **Middle:** *Spherical representation of 3D data in the* (I_x, I_y, I_t) *space before eliminating* outliers. **Right:** *Spherical representation after eliminating* outliers. *Two curves are more clearly to see. See tables 4.1, 4.2 and 4.3 for estimation results.*

The Effect of Eliminating Outliers in Occlusion Estimation

window size	eliminating outliers	occluding speed	occluded speed
33×33	before	$(0.986, 0.999)$	$(0.986, -0.988)$
	after	$(0.998, 0.999)$	$(0.990, -0.995)$
17×17	before	$(0.880, 0.971)$	$(0.859, -0.869)$
	after	$(0.988, 1.013)$	$(0.993, -0.998)$

Table 4.1: *Estimation results before and after eliminating* outliers *with different window sizes. For comparison we apply the EM algorithm on the orientation signature with the same tolerance parameter and initial values before and after eliminating* outliers*:* $\sigma_r = 0, 1$, $(u_{10}, v_{10}) = (0.8, 0.3)$, $(u_{20}, v_{20}) = (1.2, -0.1)$*. The ratio between the number of* outliers *and the number of pixels with a single motion is larger in the* 17×17 *window than in the* 33×33 *window. Thus, the error of the results in the smaller window before eliminating* outliers *is larger than the error of other results. After eliminating* outliers*, we obtain reasonable results again. This example displays vividly that the EM algorithm is purely a statistic method.*

ment in the occlusion sequence. Fleet *et al.* [FBJ98] modelled an occlusion boundary explicitly in a local circular mask with six parameters, i.e. the orientation of the boundary, the distance between the boundary and the center of the circular mask, and four motion parameters of both occluding and occluded signal. But this model is only suitable for a straight-line boundary.

The spectral model of the occlusion boundary [YDBS99, BB00] assumes implicitly that the boundary is an edge (equation (4.7)). If the boundary has other contours, we have to change the $U(\mathbf{x})$ in equation (4.7). Consequently, the spectrum of $U(\mathbf{x})$ changes in equation (4.9) as well. Since the distribution of distortions is not yet studied under this circumstance, we cannot propose an explicit model in the spectral domain to describe all possible boundaries.

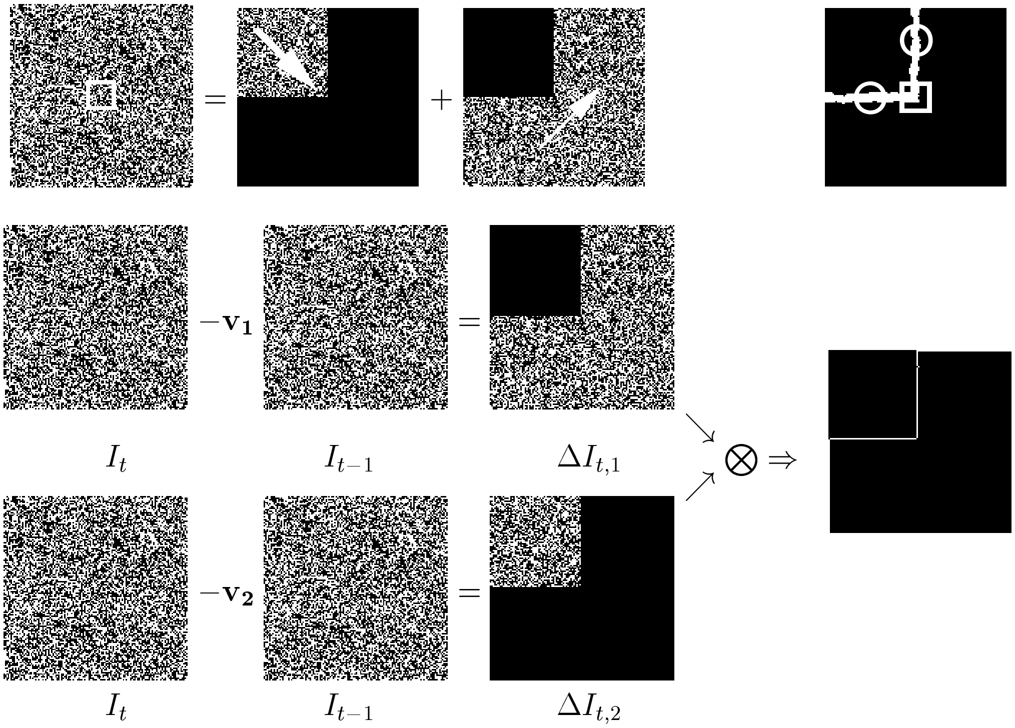


Figure 4.8: **Row 1 Left:** *One frame from a random dot occlusion sequence. It is composed of one occluding signal moving right-down and one occluded signal moving right-up.* **Row 1 Right:** *Marked occlusion regions after eigenvalue analysis. While the local explicit model [FBJ98] can describe the linear boundary parts marked with circles, it cannot describe the boundary corner marked with the square window.* **Row 2:** *"Shift-and-Subtract" with the occluding speed.* **Row 3:** *"Shift-and-Subtract" with the occluded speed.* **Between Row 2 and Row 3 Right:** *The localized occlusion boundaries. Here we do not consider the border problem.*

Instead of using the explicit boundary model, we apply the "shift-and-subtract" technique to localize motion boundaries. The "shift-and-subtract" technique is based on the spatial coherence of the image sequence [BBHP92, Cho95, WA96]. Concretely, we consider three successive frames I_{t-1}, I_t, and I_{t+1} in an image sequence. We first shift the frame I_{t-1} with two estimated speeds $\mathbf{v}_1$ and $\mathbf{v}_2$ to form the shifted frames $I_{t-1}(\mathbf{x}+\mathbf{v}_1)$ and $I_{t-1}(\mathbf{x}+\mathbf{v}_2)$. Then we calculate two difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$

$$\left\{ \begin{array}{lcl} \Delta I_{t,1}(\mathbf{x}) & = & I_t(\mathbf{x}) - I_{t-1}(\mathbf{x}+\mathbf{v}_1) \\ \Delta I_{t,2}(\mathbf{x}) & = & I_t(\mathbf{x}) - I_{t-1}(\mathbf{x}+\mathbf{v}_2) \end{array} \right. . \tag{4.29}$$

In case of occlusion we will observe one region with zero intensity in each one of $\Delta I_{t,1}$ and $\Delta I_{t,2}$. These two regions are complementary in coordinates (figure. 4.8). Their intersection localizes the boundary region B_t. Thus, we extract the boundary information in a simple way without using an explicit model.

By repeating the same process on frames I_t and I_{t+1}, we obtain the shifted boundary region B_{t+1} and track therefore the movement of occlusion boundaries. Since the occlusion boundaries move consistently with the occluding signal, we solve the foreground/background ambiguity [BF99]. This process is displayed in figure 4.8 as well.

The "shift-and-subtract" technique also distinguishes occlusion from transparency, as there is no zero region in either $\Delta I_{t,1}$ or $\Delta I_{t,2}$ in case of transparency. Furthermore, we can use this technique to decompose transparency scenes into their multi-layer representations [WA93] (figure 4.10).

4.7 Experiments

We begin with the occlusion sequence in figure 4.3, whose orientation signature is shown in figures 4.4 and 4.7. Since the EM algorithm based on the orientation signature (equation (3.9)) is similar to the planar EM algorithm based on the equation (3.8) except the advantage of setting initial values appropriately, we apply the EM algorithm based on the orientation signature in the following. We use our filter with and without radial averaging compensation to confirm the analysis in section 3.4. In the spatial model, we also compare the estimation results before and after eliminating *outliers*.

In order to compare the spatial and spectral model, we set the same tolerance parameter $\sigma_r = 0.1$ and the same initial values for both spatial and spectral EM algorithm. At first, we set the initial values of motion parameters arbitrarily as $(u_{10}, v_{10}) = (0.8, 0.3)$ and $(u_{20}, v_{20}) = (1.2, -0.1)$. Then, we use the extreme points

$(\theta_{m1}, \phi_{m1}) = (-130^\circ, 55^\circ)$ and $(\theta_{m2}, \phi_{m2}) = (130^\circ, 55^\circ)$ in the orientation signatures and calculate the initial values appropriately as $(u_{10}, v_{10}) = (0.9, 1.1)$ and $(u_{20}, v_{20}) = (0.9, -1.1)$. The results in tables 4.2 and 4.3 show that if we have properly initial values, the iteration number of the EM algorithm reduces.

Occlusion Estimation with Arbitrarily Initial Values

model	outlier	averaging	iteration	occluding	occluded
spatial model	before elimination	yes	3	$(0.927, 0.998)$	$(0.949, -0.971)$
		no	3	$(0.986, 0.999)$	$(0.986, -0.988)$
	after elimination	yes	3	$(0.985, 1.002)$	$(0.977, -0.991)$
		no	3	$(0.998, 0.999)$	$(0.990, -0.995)$
spectral model	not available	yes	7	$(1.187, 1.194)$	$(1.112, -1.147)$
		no	4	$(0.898, 0.948)$	$(1.106, -1.099)$

Table 4.2: *Estimation results of the occlusion sequence shown in figure 4.3, whose orientation signatures are shown in figures 4.4 and 4.7. In both spatial and spectral EM algorithm we use the same parameters:* $\sigma_r = 0, 1$, $(u_{10}, v_{10}) = (0.8, 0.3)$, $(u_{20}, v_{20}) = (1.2, -0.1)$. *We use a* 33×33 *window to obtain the orientation signature and a* $5 \times 5 \times 3$ *window to detect* outliers *in the spatial model. In the spectral model we use a* $32 \times 32 \times 32$ *window to obtain the orientation signature.*

Occlusion Estimation with Properly Initial Values

model	outlier	averaging	iteration	occluding	occluded
spatial model	before elimination	yes	1	$(0.938, 1.005)$	$(0.923, -0.960)$
		no	1	$(0.980, 0.997)$	$(0.963, -0.974)$
	after elimination	yes	1	$(0.987, 1.002)$	$(0.967, -0.986)$
		no	1	$(0.994, 0.997)$	$(0.978, -0.988)$
spectral model	not available	yes	2	$(1.182, 1.191)$	$(1.110, -1.145)$
		no	2	$(0.966, 1.002)$	$(1.007, -1.026)$

Table 4.3: *Occlusion estimation with properly initial values* $(u_{10}, v_{10}) = (0.9, 1.1)$ *and* $(u_{20}, v_{20}) = (0.9, -1.1)$. *These initial values are calculated using the extreme points in the orientation signatures in figure 4.4. The other conditions are the same as those in table 4.2. A comparison with table 4.2 shows that the iteration numbers of the EM algorithms reduce using these properly initial values. The estimation results in both tables are in the same precisions level since the input data does not change.*

In each table, we further observe that, due to the finer resolution in the spatial model, the spatial EM algorithm provides more accurate estimation results than the spectral EM algorithm and needs less iteration. We also confirm that the estimation results without averaging compensation are more precise than the results with averaging compensation. Moreover, if we detect and eliminate *outliers* before estimating occlusion with the spatial model, we can improve the precision of the estimation results.

For the transparency sequence in figure 4.3, we have similar comparisons in table 4.4. Since the spatial model cannot provide convincible results, we list only the results using the spectral model.

Spectral Estimation of Transparency

initial values	averaging	iteration	speed 1	speed 2
$(0.8, 0.3)$, $(1.2, -0.1)$	yes	6	$(1.173, 1.190)$	$(1.019, -1.050)$
	no	5	$(0.899, 0.963)$	$(1.026, -1.038)$
$(0.9, 1.1)$, $(0.9, -1.1)$	yes	3	$(1.187, 1.193)$	$(1.014, -1.048)$
	no	1	$(0.950, 0.999)$	$(0.956, -0.990)$

Table 4.4: *Estimation results of the transparency sequence using the spectral EM algorithm. Compared with the arbitrarily initial values, the properly initial values result in faster convergence. Similar to the occlusion estimation results, the results without averaging compensation are in general more accurate than the results with averaging compensation.*

In order to test the performance of the EM algorithm on determining the number of models automatically, we propose an example of one moving signal. We use a sequence with a single motion with velocity $(1, -1)$ [pixel/frame] for the test. Both the spatial and the spectral EM algorithm should converge to one speed even with the arbitrarily initial values, if they are able to determine the number of motions automatically. With the initial values $(1.2, -0.1)$ and $(0.8, 0.3)$ the spatial EM algorithm converges to $(0.995, -1.001)$ after 2 iterations and the spectral EM algorithm converges to $(1.057, -1.045)$ and $(0.951, -1.011)$ after 2 iterations. Taking into account that the spectrum of the image sequence is blurred, this result is not surprising. This fact indicates that in the EM algorithm we do not know the number of motions exactly. In order to confirm if the spectral EM algorithm converges with the properly initial values, we run the program again by setting both initial values as $(0.9, -1.1)$. This time the spectral EM algorithm converges to $(1.004, -1.029)$ after 2 iterations. By using properly initial values the iteration is guaranteed not to fall into some unexpected local minimum. From this result we also observe that the spatial model provides finer resolution than the spectral model.

In figures 4.9 and 4.10 we show a real transparency sequence to compare the spatial and spectral model. This transparency sequence contains a right moving portrait and a mirrored left moving muesli package. Note that the sequence has the aperture problem inside the multiple motion region.

For this sequence we apply the BCCE based algorithm to estimate single motion and use the eigenvalue analysis to determine the multiple motion candidates at first. In order to reduce the computational complexity, we apply the spatial and spectral EM algorithm only in these candidate regions to estimate multiple motions. From figure 4.10 we see that the spatial EM algorithm is not able to estimate transparent motions correctly, while the spectral EM algorithm works well. The optical flow in the spectral EM approach is sparse. This is due to the fact that in some regions of the package we do not have adequate texture information. For a robust performance we ignore these regions in estimation. Similarly, the multiple motion candidate regions are not in line with the package shape since some regions of the package have the aperture problem. This example confirms the above conclusion of model comparison.

After obtaining the motion parameters, we further decompose the transparency scene into multi-layers with the "shift-and-subtract" technique. The results are shown in difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$.

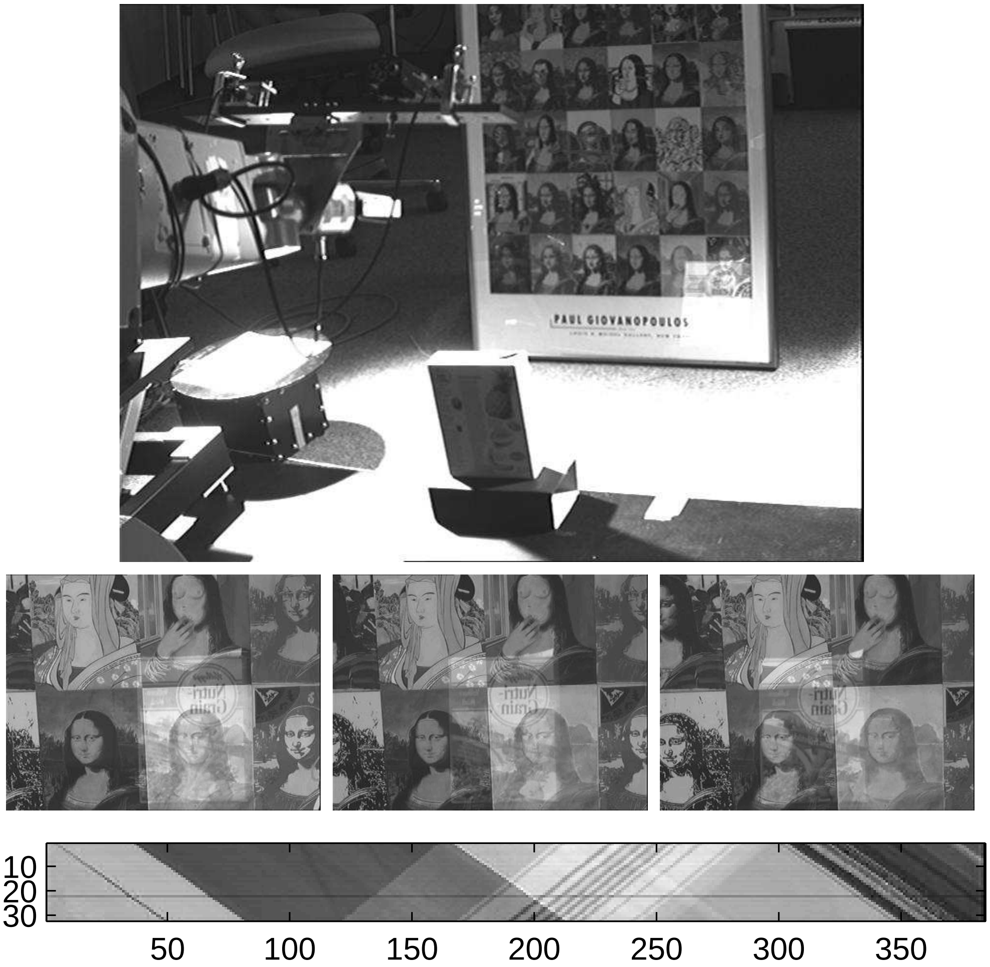


Figure 4.9: *Real transparency sequence.* **Row 1:** *This image displays how we produce the transparency sequence. We move the camera and the muesli package separately to produce two independent motions. The left moving muesli package is reflected on the glass of the portrait so that we obtain a transparency sequence.* **Row 2:** *The first,* 16*-th and* 32*-th frames of the image sequence. Each frame has* 288×384 *pixels. We use the* 16*-th frame for motion estimation and scene analysis.* **Row 3:** *Epipolar slide of the whole image sequence volume at the* 150*-th row. We can see the transparent motions clearly as the overlapping of two tilted structures.*

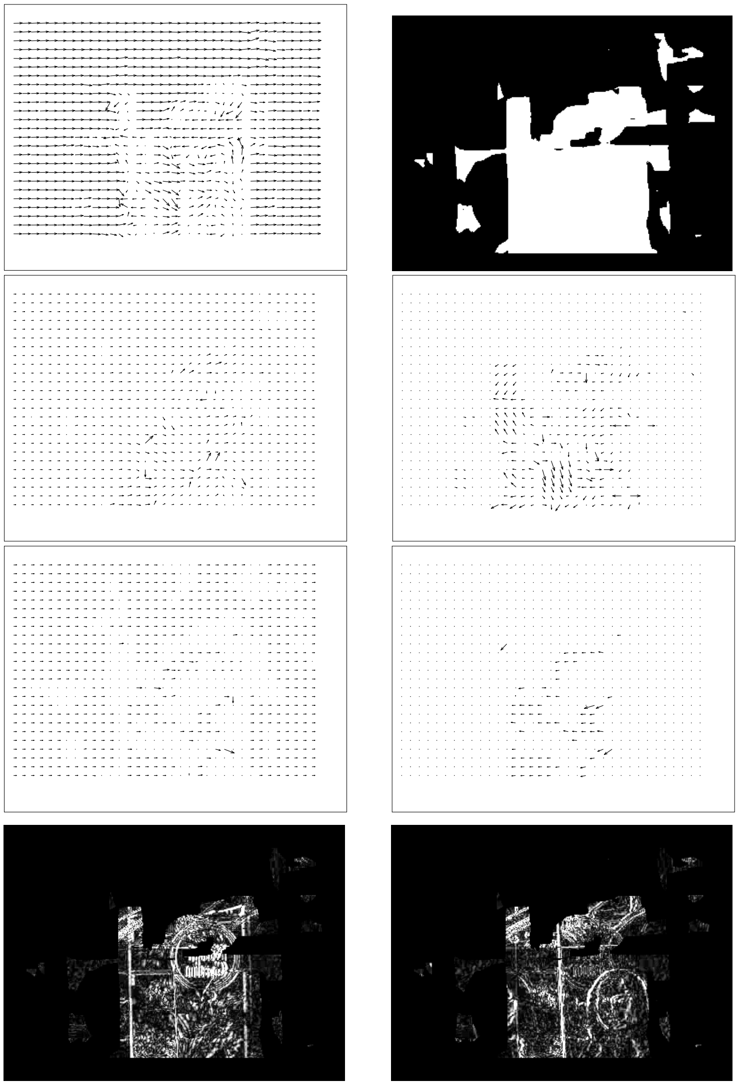


Figure 4.10: *Comparison of spatial and spectral EM algorithm on a real transparency sequence.* **Row 1 Left:** *Estimation results using a single motion model. Single motion model fails in case of transparent motions.* **Row 1 Right:** *Marked two motion candidate regions according to the eigenvalue analysis. This region is not in line with the package shape since on the package there are regions with the aperture problem.* **Row 2:** *Optical flow of the spatial EM approach. The results are not correct in the transparent regions.* **Row 3:** *Optical flow of the spectral EM approach. They are more accurate. The optical flow of the spectral EM approach is sparse since in some regions we do not have adequate texture information. For a robust performance we ignore the results in these regions.* **Row 4:** *Decomposition of the transparency scene into two layers using the spectral EM results and the "shift-and-subtract" technique.*

Real Occlusion Analysis

In this subsection we analyze real occlusion sequences. In figure 4.11 we show the well known "flower garden" occlusion sequence, in which a left moving trunk is covering left moving flower bed and houses. The speed of the trunk is about double so fast as the speed of the flower bed and the houses. We can see this clearly in the epipolar slide. We first estimate motions using the single motion model. At the occlusion boundaries the results are not correct, as shown in row 3 of figure 4.11. After the eigenvalue analysis we detect two motion candidate regions and the regions with the aperture problem, which are shown in row 3 as well. We observe that the regions with the aperture problem are very large in the sky and along the trunk. In the two motion candidate regions, we apply the spatial EM algorithm to estimate multiple motions. The results are displayed in row 4. In row 5 we apply the "shift-and-subtract" technique to localize occlusion boundaries. In fact, the difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$ can be viewed as the result of occlusion segmentation. Since there is no difference inside the regions with the aperture problem before and after the shifting, we observe only the boundaries of the trunk in $\Delta I_{t,1}$. We further localize occlusion boundaries from $\Delta I_{t,1}$ and $\Delta I_{t,2}$. The result is shown in row 5 as well. The boundaries are badly connected since the nonzero regions in $\Delta I_{t,1}$ are discrete due to the aperture problem. This indicates that the aperture problem obstructs not only motion estimation, but also scene analysis.

Figures 4.12 and 4.13 display an occlusion example, in which a right moving box is covering a left moving picture. The image is rich in texture so that we do not face the aperture problem. We first estimate motions using the single motion model. Since the box as well as the picture have purely translational motions and since there is no depth difference in the box region or in the picture region, there is no speed difference in either region. The incorrect results at the occlusion boundaries are clear to see. We confirm this observation using the eigenvalue analysis and mark all possible multiple motion regions in row 2. In this sequence we would like to test the effect of eliminating *outliers*. For performance comparison we apply the EM algorithm vertically along the occlusion boundary before and after eliminating *outliers*. It is a little bit difficult to compare the precision of estimation results since the ground truth is unknown. But we observe that on each side of the boundary there is almost no speed difference among pixels. There-

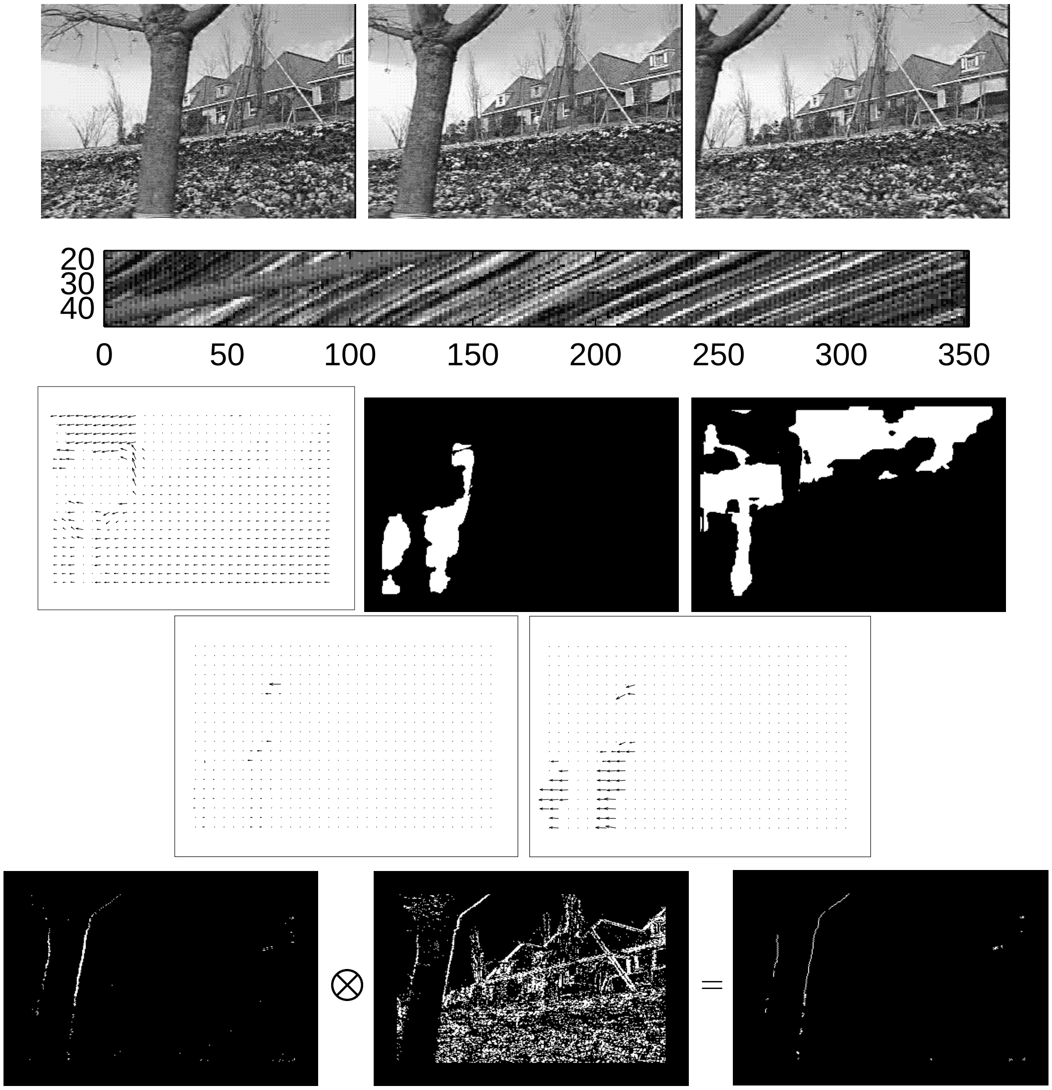


Figure 4.11: **Row 1:** *The* 17*-th,* 32*-th and* 48*-th frames of the image sequence. Each frame has* 240×352 *pixels. Here we use the* 32*-th frame for motion analysis.* **Row 2:** *Epipolar slide of the whole image sequence volume at the* 200*-th row.* **Row 3 Left:** *Estimation results using the single motion model. At motion boundaries the results are not correct.* **Row 3 Middle:** *Marked two motion candidate regions according to the eigenvalue analysis.* **Row 3 Right:** *Regions with the aperture problem.* **Row 4:** *Optical flow applying the spatial EM algorithm.* **Row 5 Left:** *Difference image* $\Delta I_{t,1}$*. The difference values are zero for any shifting inside the regions with the aperture problem. Thus, we observe only the boundaries of the trunk.* **Row 5 Middle:** *Difference image* ΔI_2*.* **Row 5 Right:** *Detected motion boundaries.*

fore, we may use the estimation results with a large window as ground truth since there are much more *normal* pixels than *outliers* in such a large window. In the results with a small window we observe the improvement after eliminating *outliers* clearly. In the window centered at $(160, 137)$ the results are not reasonable because there are only *four* pixels of the occluded signal remaining after eliminating *outliers*. This example demonstrates vividly the necessity of introducing reliability measure (equation (4.28)). By using the "shift-and-subtract" technique, we further localize the occluding boundary, which is displayed as intersection of zero regions in $\Delta I_{t,1}$ and $\Delta I_{t,2}$, as shown in row 4 of figure 4.13. This "shift-and-subtract" technique works also for boundaries with complex contours like the corners of the right moving box.

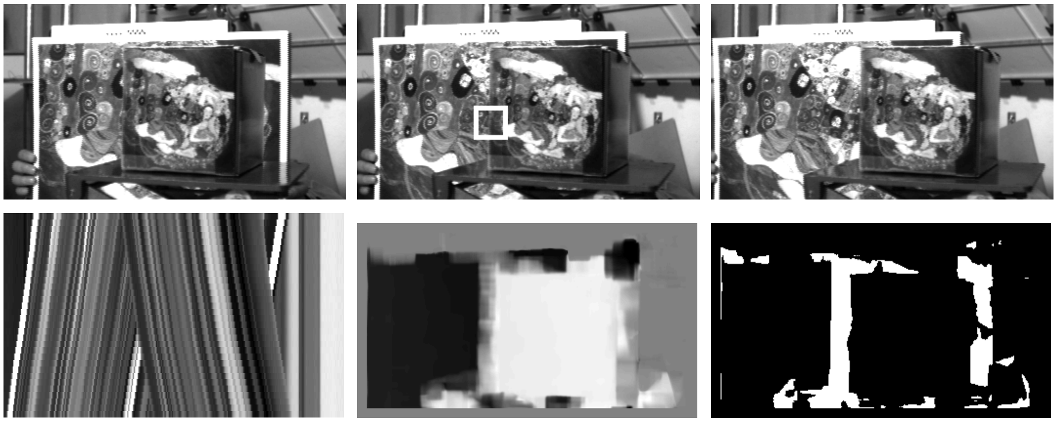


Figure 4.12: **Top:** *The first,* 16*-th, and* 32*-th frame of the occlusion sequence. Each frame has* 200×350 *pixels. The white window in the* 16*-th frame is centered at* $(122, 137)$*.* **Bottom Left:** *The epipolar slice of the sequence along row* 122*. The first frame is at the top of the slice. The occlusion is characterized as two overlapping structures.* **Bottom Middle:** *The result of the single motion estimation algorithm. Since the vertical speed components are almost zero in this sequence, we show only horizontal speed components, using black color for negative speed (moving to the left) and white color for positive speed (moving to the right).* **Bottom Right:** *Marked two motion candidate regions after the eigenvalue analysis. Notice the results in the still background are not correct.*

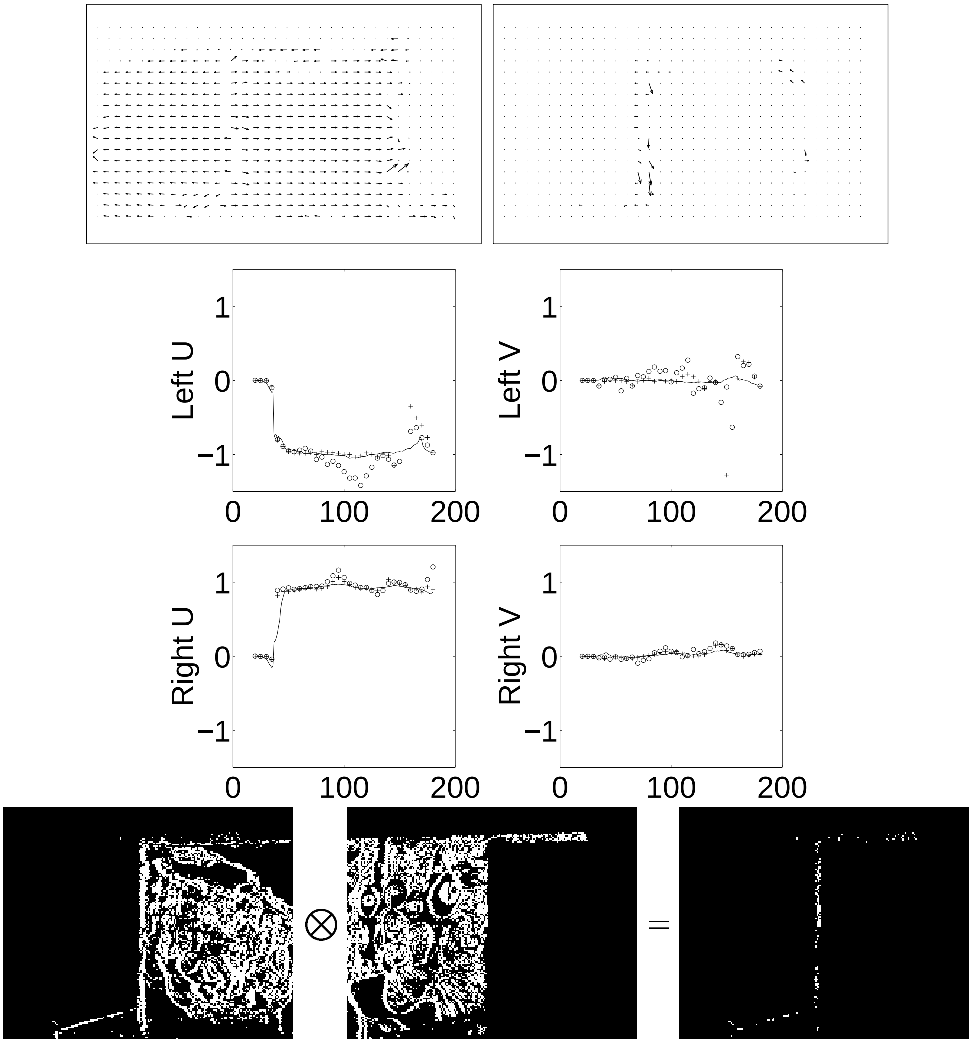


Figure 4.13: **Row 1:** *Motion estimation results in the* 16*-th frame.* **Row 2 and Row 3:** *Estimation results along column* 137 *using a* 15×15 *window. We use the results with a* 31×31 *window as the ground truth and draw them with solid lines. We draw the results before eliminating* outliers *with circles and the results after eliminating* outliers *with crosses. For comparison we draw different speed components separately. For clarity of drawing we sample the results with an interval of 5 pixels along column* 137*. In the window centered at* $(160, 137)$ *the results are not reasonable since there are only* four *pixels of the occluded signal remaining after eliminating* outliers*. This example demonstrates the necessity of introducing reliability measure clearly.* **Row 4:** *The result after "shift-and-subtract". For clarity we enlarge the occlusion boundary region. The "shift-and-subtract" technique works also for boundaries with complex contours like the corners of the right moving box.*

Figures 4.14 and 4.15 show us another real sequence with the ground truth. This "block world" sequence is very difficult for the constant motion model used here. But using the ground truth, we still can compare the performances of the EM algorithm before and after eliminating *outliers*. The results show that the error after elimination are less. We further test if the "shift-and-subtract" technique works for complex motions. The result is positive if we have correct motion parameters.

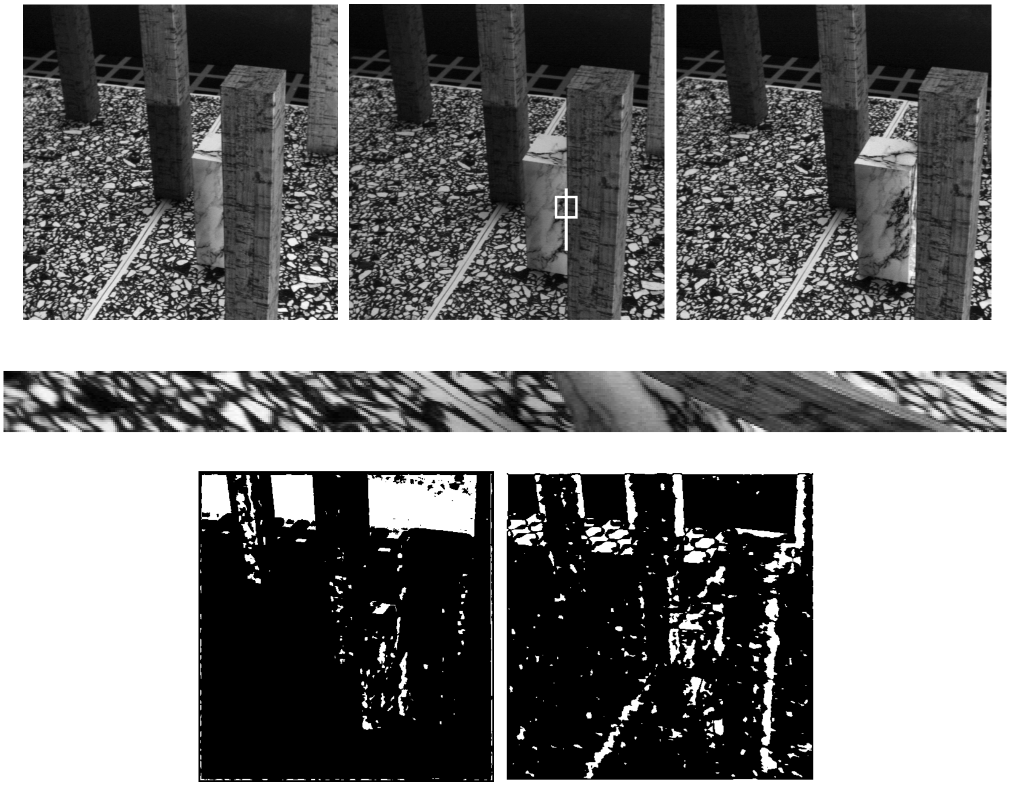


Figure 4.14: **Row 1:** *The first, the* 15*-th, and the* 31*-th frame of the block world sequence. Each frame has* 512×512 *pixels. The white window in the* 15*-th frame shows us the window across the boundary and the white line shows the column along which we apply the EM algorithm for comparison.* **Row 2:** *Epipolar slide along row 353. The first frame is at the top of the slice. This scene is very complicated due to depth difference, the aperture problem, and affine property of motions.* **Row 3 Left:** *Marked* outliers. **Row 3 Right:** *Regions with the aperture problem.*

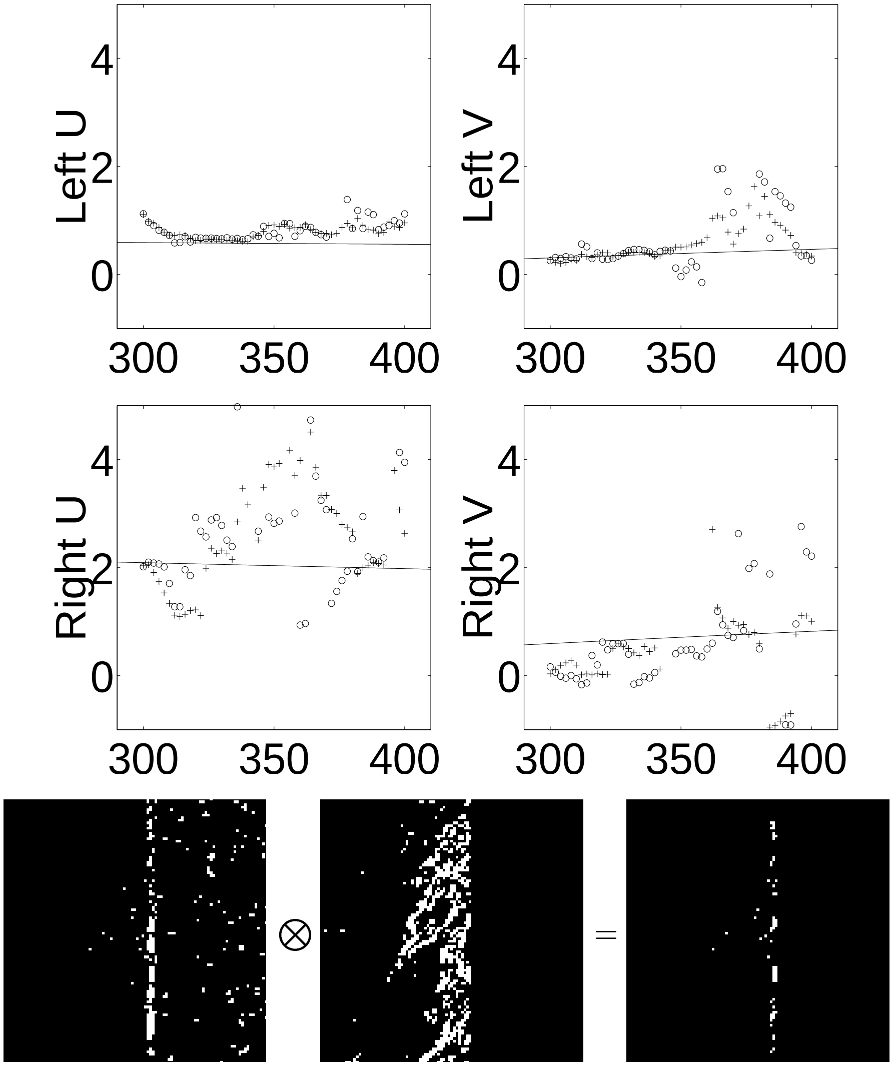


Figure 4.15: **Row 1 and Row 2:** *Estimation results before and after eliminating* outliers *vs. row index (from row 300 to row 400). We draw the ground truth with solid lines, the results before eliminating* outliers *with circles, and the results after eliminating* outliers *with crosses. In fact, the constant motion model cannot treat such a difficult sequence due to complicated motions and the aperture problem. But we still can see that the errors are less after eliminating* outliers. **Row 3:** *The "shift-and-subtract" technique works using the ground truth. For clarity we enlarge the boundary region.*

4.8 Discussions

In this chapter, we studied multiple motion analysis from the standpoint of orientation analysis. Thorough observation and theoretical analysis show that current spatial models cannot handle both occlusion and transparency in a unified manner, while the spectral model provides a unified framework. Based on this analysis, we used a new kind of 3D approximately steerable filter in motion estimation. This method is superior to principle axis analysis based approaches and current 3D steerability approaches in achieving high orientation resolution. Comparisons showed that our approach is more efficient and robust than the similar spatiotemporal Hough transform and outperforms the existing EM algorithm applied in the derivative space.

The good performance of a motion estimation algorithm relies on the correct model of motions. However, an ideal model does not guarantee that this model is feasible. In practice, the spatial models work in the derivative space. But the derivative of a discrete image sequence does not exist and we have to approximate the derivative with differentiation, while this approximation is still problematic [Sim94, FS97, SKJ97]. Similarly, in the frequency domain the ideal spectrum of the image sequence does not exist. We obtain only a blurred spectrum after applying the LFT. As a result, current motion estimation algorithms suffer more or less from the discrete property of input image sequences. According to the comparison, the problem in the spectral domain is severer than in the spatial domain. Therefore, the spectral model is less feasible.

Another related point is the choice of window size. Generally, we must have a small observing window around the considered pixel so that the simple motion model can approximate the actual motion in the neighborhood robustly. But in the spectral model we must extend the window to have a reasonable resolution of the spectrum. In this enlarged window a simple motion model will not approximate the actual motion convincingly. In fact, we provide *only* averaged velocities inside the neighborhood. In order to improve the approximation of motions, we must use motion models with higher order. In the spatial domain the affine model has been introduced. But in the spectral domain a multiple motion model of higher order still remains open [CGN96].

In occlusion estimation, we proposed to eliminate *outliers* in the derivative domain. Compared with current probabilistic approaches, which include the *outliers* in the estimation, our method improves the *quality* of input data and therefore provides more exact results.

In order to localize occlusion boundaries and track their movement, we utilized the spatial coherence inside the frame and applied the “shift-and-subtract” technique. We did not use an explicit local model of the boundary region. But we still obtained the desired information about the occlusion boundaries. Furthermore, multiple motions can be segmented very efficiently by combining estimation techniques and spatial coherence [WA96]: The region with the same motion parameters can be figured out by calculating the difference between two frames with estimated speeds.

The spatial coherence information is also a key cue to distinguish occlusion and transparency in the spatial domain. Actually, it is not difficult to distinguish occlusion from transparency in the frequency domain. For example, we can look at a set of estimation results by shifting the observing window and observe their variation. Since occlusion is more local than transparency, the number of motions changes from two to one after the observing window has crossed occlusion boundaries, while in case of transparency the number of motions remains the same. We may also observe the relative ratio between data points outside the motion planes and those on the motion planes [YDBS99]. This ratio is much larger in case of occlusion than in case of transparency since all energy of the transparency lies on the dominant planes. However, in the frequency domain we cannot localize motion boundaries due to the well known uncertainty principle: The spectrum of the observing window provides us no localization information inside the window. Therefore, we must go back to the spatial domain to detect and to localize motion boundaries. This advantage of the spatial motion model benefits from the fact that the spatial model maintains the spatial coherence information, while the spectral model does not. This fact indicates that the spatial coherence information plays a very important role in image segmentation and scene analysis. The “shift-and-subtract” technique and the recently introduced *normalized cut* approach [SM97, SM98] remind this point vividly.

Appendix B: The Spectrum of Occlusion

In this appendix, we derive the spectrum of occlusion and study if the occlusion may attenuate the aperture problem. Substituting equation (4.9) into (4.8) and utilizing the product property of the impulse function, we have

$$\begin{aligned}\tilde{I}(\kappa,\omega_t) &= 2\pi^2\tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t)+(1-2\pi^2)\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t)\\ &+\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t)*\tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t)\\ &-\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t)*\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t).\end{aligned} \tag{4.30}$$

For simplification, we define the third part as $A(\kappa,\omega_t)$ and the fourth one as $B(\kappa,\omega_t)$. Denoting with $\mathcal{F}$ and $\mathcal{F}^{-1}$ the forward and inverse Fourier transform, we have

$$\begin{aligned}A(\kappa,\omega_t) &= \tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t)*\tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t)\\ &= \mathcal{F}\{[\mathcal{F}^{-1}(\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t))]\cdot[\mathcal{F}^{-1}(\tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t))]\}.\end{aligned}$$

The first part of the convolution is a line in 3D frequency space and the second part is a plane containing the line in the first part. According to the relative symmetry of $\mathcal{F}$ and $\mathcal{F}^{-1}$ [GK95], the first part turns out to be an impulse plane in the spatial domain after inverse Fourier transform and the second part is then an impulse line on this plane. Their multiplication results in an impulse line in 3D spatial domain. We take the forward Fourier transform again and obtain an impulse plane with the orientation of $\kappa^T\mathbf{v}_1+\omega_t=0$

$$A(\kappa,\omega_t)=[\frac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)*\tilde{I}_1(\kappa)]\delta(\kappa^T\mathbf{v}_1+\omega_t)=A(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t). \tag{4.31}$$

Taking into account that $A(\kappa,\omega_t)$ has the same orientation as the occluding signal, we draw a conclusion that this part just does nothing else as strengthening the spectrum of the occluding signal.

The term $B(\kappa,\omega_t)$ can be computed similarly

$$\begin{aligned}B(\kappa,\omega_t) &= \tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t)*\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t)\\ &= \mathcal{F}\{[\mathcal{F}^{-1}(\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t))]\cdot[\mathcal{F}^{-1}(\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t))]\}.\end{aligned} \tag{4.32}$$

From equation (4.32) we see that $B(\kappa,\omega_t)$ is a convolution between a line and a plane in 3D frequency space. Only when $\tilde{I}_2(\kappa)$ is discrete, we may have some shifted *lines* formed by the spectral planes $\kappa^T \hat{\eta}_\perp = 0$ and $\kappa^T \mathbf{v}_1 + \omega_t = 0$. Otherwise, $B(\kappa, \omega_t)$ occupies a volume in the frequency space. Thus, we obtain the equation (4.10).

Now let us see if the occlusion may attenuate the aperture problem. The aperture problem is in fact a correspondence problem. In the spatial domain, if we do not have adequate information, we cannot find the corresponding points correctly (figure 4.16). Equivalently, in the frequency domain, if we do not have adequate components to form a spectral plane, we have too many possible normal vectors of the spectral plane to determine which vector is the desired one (figure 4.16).

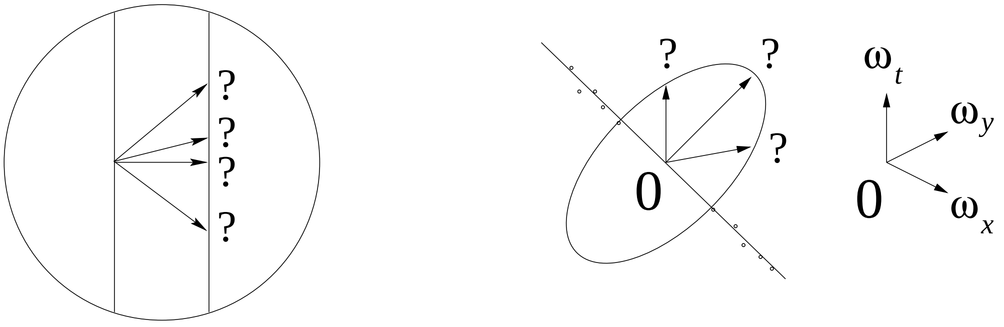


Figure 4.16: **Left:** *Aperture problem represented in the spatial domain: We cannot find the corresponding point correctly due to the lack of information in the neighborhood of the considered point (redrawn from [Jäh93]).* **Right:** *Aperture problem represented in the frequency domain: The energy spectrum degenerates into a line in the spectral domain. We cannot determine which normal vector of the line is normal to the motion plane. The spectral origin lies in the middle of the 3d-drawing.*

We assume that both the occluding and the occluded signal have an aperture problem. Correspondingly, in the $(\omega_x, \omega_y, \omega_t)$ space their spectra are characterized as vertical planes with constant components along the ω_t direction

$$\left\{ \begin{array}{lcl} \tilde{I}_1(\kappa) & = & a_1(\kappa)\delta(\kappa^T \xi_1) \\ \tilde{I}_2(\kappa) & = & a_2(\kappa)\delta(\kappa^T \xi_2) \end{array} \right. , \tag{4.33}$$

where ξ_i $(i = 1, 2)$ denotes the normal vector of the 2D spectral line in the (ω_x, ω_y) space and is orthogonal to the gradient of the signal. Thus, we may rewrite the spectrum of occlusion as

$$\begin{aligned}\tilde{I}(\kappa,\omega_t) &= 2\pi^2 a_1(\kappa)\delta(\kappa^T\xi_1)\delta(\kappa^T\mathbf{v}_1+\omega_t) + (1-2\pi^2)a_2(\kappa)\delta(\kappa^T\xi_2)\delta(\kappa^T\mathbf{v}_2+\omega_t)\\ &+\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t) * a_1(\kappa)\delta(\kappa^T\xi_1)\delta(\kappa^T\mathbf{v}_1+\omega_t)\\ &-\tfrac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t) * a_2(\kappa)\delta(\kappa^T\xi_2)\delta(\kappa^T\mathbf{v}_2+\omega_t).\end{aligned} \tag{4.34}$$

In the above expression, the spectrum of the occluding signal as well as that of the occluded signal are a 3D line in the frequency space and we cannot estimate their full velocities. The third term may help in the estimation of occluding velocity. If the gradient of the occluding signal is not parallel to the gradient of the motion boundary ($\xi_1 \nparallel \hat{\eta}_\perp$), we still can obtain the spectral plane from $A(\kappa,\omega_t)$ and estimate the full velocity of the occluding signal

$$A(\kappa,\omega_t) = [\frac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp) * a_1(\kappa)\delta(\kappa^T\xi_1)]\delta(\kappa^T\mathbf{v}_1+\omega_t). \tag{4.35}$$

If the gradients above are parallel ($\xi_1 \parallel \hat{\eta}_\perp$), then it is hopeless to determine the full velocity of the occluding signal because $A(\kappa,\omega_t)$ degenerates into a line

$$A(\kappa,\omega_t) = [\frac{2\pi}{i\kappa^T\hat{\eta}} * a_1(\kappa)]\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t). \tag{4.36}$$

The fourth term $B(\kappa,\omega_t)$ remains as distortion. If the gradient of the occluded signal is not parallel to the gradient of the motion boundary ($\xi_2 \nparallel \hat{\eta}_\perp$), $B(\kappa,\omega_t)$ is a convolution of two 3D lines. This results in a plane whose normal vector $\mathbf{n}$ is dependent on $\hat{\eta}_\perp$, ξ_2, and both speeds

$$\mathbf{n} = ((\hat{\eta}_\perp,0)\times(\mathbf{v}_1,1))\times((\xi_2,0)\times(\mathbf{v}_2,1)). \tag{4.37}$$

We cannot use it to estimate the full velocity of the occluded signal. If $\xi_2 \parallel \hat{\eta}_\perp$, we obtain then a vertical plane which cannot be used, either

$$B(\kappa,\omega_t) = [-\frac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\mathbf{v}_1+\omega_t) * a_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t)]\delta(\kappa^T\hat{\eta}_\perp). \tag{4.38}$$

In short, we may attenuate the aperture problem by utilizing the occlusion. If the gradient of the occluding signal is not parallel to the gradient of the occlusion boundary, we can estimate the full velocity of the occluding signal [BB00]. The full velocity of the occluded signal remains unsolvable.

In case of transparency the aperture problem remains unsolvable since both spectral planes degenerate into lines and we do not have the distortion term (see equation (4.14) for detail).

Chapter 5

Summary

In this thesis, we addressed the problem of local orientation analysis with steerable filters. From the standpoint of the sampling theory, as current 2D steerable filters sample the orientation space with Dirac functions in the frequency domain, they suffer from the consequences of the uncertainty principle: In order to achieve high resolution in orientation, they need a large number of basis filters. Furthermore, these basis filters unfortunately have a wide support which accentuates the computational burden. In order to achieve the lower bound in the uncertainty principle, we proposed a novel alternative to current steerability approaches by utilizing a set of polar separable filters with small angular support to sample orientation space locally. Then, the orientation signature is obtained by interpolating orientation samples using Gaussian functions. Compared with current steerability techniques, our approach achieves a higher orientation resolution with a lower complexity. In addition, we built a polar pyramid to characterize junctions of arbitrary inherent orientation scales.

Further, we extended the new steerable filter from 2D space to 3D space for volume image processing and multiple orientation analysis. The comparisons with current 3D steerable filters confirm once more the advantage of our filter with respect to the uncertainty principle. An analysis of our new filter in a broader background indicates the following points

- The tensor approach and the principal axis analysis detect only one dominant orientation. Consequently, they are not suitable for multiple orientation analysis.

- Our approach is relevant to the Gabor wavelets, which sample the spectrum using filters with Gaussian shape in the frequency domain and which have the same orientation resolution along the radial direction. The difference is that we use Gaussian functions only in the angular space and treat the radial direction differently since we are only interested in orientation analysis.
- Our 3D steerable filter is similar to the 3D Hough transform in the sense that both methods sample the parameter space in the implementation. But our filter is more efficient since it has much smaller support than the filter equivalent to the 3D Hough transform. Besides, in the Hough transform based methods the search of the second maximum is generally problematic, while in our approach this problem is solved.
- In multiple motion analysis, the expectation maximization (EM) algorithm does not detect the number of motions exactly. We may use our filter to determine the number of motions and then use the EM algorithm for estimation refinement. Moreover, as the EM algorithm is greatly dependent on the initial values, our filter helps to improve the performance of the EM algorithm by setting properly initial values of motion parameters.
- Compared with spherical surface tessellation based algorithms such as the 3D orientation histogram and the spherical wavelets, our filter decomposes the sphere alternatively in the feature space with a set of overlapped basis filters. Though projecting the spherical surface onto 2D feature space is not an isometric-mapping and the rotation symmetry is lost after projection, this transform benefits structure display and post-processing.

In motion analysis, we provided a detailed description of occlusion and transparency both in the spatial domain and in the spectral domain to comprehend multiple motions clearly. The comparison between the spatial motion model and the spectral motion model showed that the spatial model can treat only occlusion, while the spectral model treats both occlusion and transparency in a uniform manner. The comparison further indicates that the spatial model is more suitable for occlusion analysis: The spatial model provides finer resolution and needs much less frames. Besides, in the frequency domain we fail to localize motion boundaries due to the well known uncertainty principle: The spectrum of the observing window can provide us no localization information inside the window. The only cue to localize occlusion boundaries is the spatial coherence information.

In the occlusion estimation, we also dealt with the outlier problem and introduced a multi-window strategy to purify input data and to improve the precision of the estimation results. Having obtained the motion parameters, we further localized occlusion boundaries and tracked their movement in occlusion sequences with the “shift-and-subtract” technique. This technique, applying the spatial coherence in image sequences, can also be used to distinguish occlusion from transparency as well as to decompose transparency scenes into multi-layers. This fact reminds us again that the spatial coherence information is very useful for image segmentation and scene analysis.

In our approach, the local basis filters are non-orthogonal and their responses are locally correlated. Since our goal is not to reconstruct the original signal, but to extract features, this kind of correlation among neighboring basis filters does not bother us and we do not have to “decorrelate” the filter results. Surely, if we would like to represent the original signal with our steerable filter, we need to find out if there exists a dual basis which is bi-orthogonal to our basis. The lifting scheme may be useful to solve this open question [SS95]. We will be able to build a new kind of bi-orthogonal Gaussian wavelets after solving this problem successfully.

The preliminary results of this work are presented in [YDS98c, YDS98a, YDS98b, YDS01, YDBS99, YDS00b, YDS00a, YSD00]. The 2D orientation steerable filter has been used in the diagnosis support system of patients with facial paresis which is now installed at the Department of Otorhino-Laryngology of the University Erlangen-Nuremburg (HNO Klinik der Unversität Erlangen).

Bibliography

[AB85] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 1(2):284–299, 1985.

[And92] M. T. Andersson. *Controllable Multidimensional Filters and Models in Low Level Computer Vision*. PhD thesis, Department of Electrical Engineering, Linkoeping University, Linkoeping, Sweden, 1992.

[AS95] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proc. Int. Conf. on Computer Vision*, pages 777–784, Boston, MA, June 20-23, 1995.

[BA83] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[BA96] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[BB95] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27:433–467, 1995.

[BB97] S. S. Beauchemin and J. L. Barron. A Theory of Occlusion in the Context of Optical Flow. In F. Solina, W. Kropatsch, R. Klette, and R. Bajcsy, editors, *Advances in Computer Vision*, pages 191–200. Springer Wien New-York, November, 1997.

[BB00] S.S. Beauchemin and J.L. Barron. The frequency structure of 1d occluding image signals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:200–206, 2000.

[BBHP92] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9):886–895, 1992.

[Bei94] W. Beil. Steerable filters and invariance theory. *Pattern Recognition Letters*, 15:453–460, 1994.

[BF99] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *Proc. Int. Conf. on Computer Vision*, volume I, pages 551–558, Kerkyra, Greece, Sep. 20-27, 1999.

[BG87] J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *Proc. Int. Conf. on Computer Vision*, pages 433–438, London, UK, June 8-11, 1987.

[BGW91] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with application to texture analysis and optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[BK94] M. Bober and J. Kittler. Estimation of complex multimodal motion: an approach based on robust statistics and hough transform. *Image and Vision Computing*, 12(10):661–668, 1994.

[Bou89] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:499–511, 1989.

[Bra86] R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Book Company, 1986.

[BW93] D. H. Ballard and L. E. Wixson. Object recognition using steerable filters at multiple scales. In *IEEE Workshop on Qualitative Vision*, pages 2–10, 1993.

[BWBD86] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:26–33, 1986.

[BYX83] P.J. Burt, C. Yen, and X. Xu. Multi-resolution flow-through motion analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 246–252, Washington, DC, June, 1983.

[CGN96] W. Chen, G. B. Giannakis, and N. Nandhakumar. Spatiotemporal approach for time-varying global image motion estimation. *IEEE Trans. Image Processing*, 5(10):1448–1461, 1996.

[Cho95] G. T. Chou. A model of figure-ground segregation from kinetic occlusion. In *Proc. Int. Conf. on Computer Vision*, pages 1050–1057, Boston, MA, June 20-23, 1995.

[Dan80] P.-E. Danielsson. Rotation-invariant linear operators with directional response. In *Proc. Int. Conf. on Pattern Recognition*, Miami, Dec., 1980.

[Dau85] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1169, 1985.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[FA91] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[FBJ98] D.J. Fleet, M.J. Black, and A.D. Jepson. Motion feature detection using steerable flow fields. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 274–281, Santa Barbara, CA, June 23-25, 1998.

[FJ90] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[FL94] D.J. Fleet and K. Langley. Computational Analysis of non-Fourier Motion. *Vision Research*, 34:3057–3079, 1994.

[För94] W. Förstner. A framework for low level feature extraction. In *Proc. Third European Conference on Computer Vision*, volume II, pages 383–394, Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 801, 1994.

[FP98] T.C. Folsom and R.B. Pinter. Primitive features by steering, quadrature, and scale. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1161–1173, 1998.

[FS97] H. Farid and E. P. Simoncelli. Optimally rotation-equivariant directional derivative kernels. In *Int. Conf. on Computer Analysis of Images and Patterns*, Kiel, Germany, September 10-12, 1997.

[FS00] M. Felsberg and G. Sommer. Structure multivector for local analysis of images. Technical Report 2001, Institute of Computer Science, University Kiel, Germany, 2000.

[Gab46] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[GBG$^+$94] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C.H. Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 222–228, Seattle, WA, June 21-23, 1994.

[GD94] D. Geiger and K. I. Diamantaras. Occlusion ambiguities in motion. In *Proc. Third European Conference on Computer Vision*, volume I, pages 175–180, Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 800, 1994.

[GK95] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.

[GSA96] H. Gu, Y. Shirai, and M. Asada. MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1):58–64, 1996.

[GY90] N.M. Grzywacz and A.L. Yuille. A model for the estimate of local image velocity by cells in the visual cortex. *Proc. Royal Society of London.*, B 239:129–161, 1990.

[HC95] Chung-Lin Huang and Yng-Tsang Chen. Motion estimation method using a 3d steerable filter. *Image and Vision Computing*, 13:21–32, 1995.

[Hee87] D. J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1987.

[HJ97] H. Haußecker and B. Jähne. A tensor approach for precise computation of dense displacement vector fields. In *DAGM Symposium Mustererkennung*, pages 199–208, Braunschweig, Germany, Sep. 15-17, 1997.

[Hor86] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[HRdH$^+$92] F. Heitger, L. Rosenthaler, R. Von der Heydt, E. Peterhans, and O. Kuebler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, 32(5):963–981, 1992.

[HS99] H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißer, editors, *Handbook of Computer Vision and Applications*, volume 2, chapter 13, pages 309–396. Academic Press, 1999.

[IRP94] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16, 1994.

[Jäh93] B. Jähne. *Spatio-Temporal Image Processing*. Springer-Verlag, 1993.

[JB93] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 760–761, New York, NY, June 15-17, 1993.

[JB96] A. Jepson and M. Black. Mixture models for image representation. *Technical Report ARK96-PUB-54*, 1996.

[JHS$^+$98] B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor-based spatiotemporal image processing techniques. In *Proc. Fifth European Conference on Computer Vision*, volume II, pages 322–336, Freiburg, Germany, June 2-6, H. Burkhardt and B. Neumann (Ed.), Springer LNCS 1407, 1998.

[KG83] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Comput. Soc. Workshop Comp. Architecture Patt. Anal. Image Database Mgmt.*, pages 388–397, 1983.

[Knu82] Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Department of Electrical Engineering, Linkoeping University, Linkoeping, Sweden, 1982. Dissertation No. 88.

[Koe93] J. J. Koenderink. What is a “feature”? *Journal of Intelligent Systems*, 3(1):50–82, 1993.

[Kov96] P. Kovesi. *Invariant Measures of Image Features from Phase Information*. PhD thesis, Department of Psychology, University of Western Australia, 1996. http://www.cs.uwa.edu.au/pub/robvis/theses/PeterKovesi/.

[KRV99] S. N. Kalitzin, B. M. Ter Haar Romeny, and M. A. Viergever. Invertible apertured orientation filters in image analysis. *International Journal of Computer Vision*, 31(2/3):145–158, 1999.

[Lee96] T. S. Lee. Image Representation Using 2D Gabor Wavelets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[Mac91] B. MacLennan. Gabor Representations of Spatiotemporal Visual Images. Technical Report CS-91-144, Computer Science Department, University of Tennessee, USA, 1991.

[Mar82] D. Marr. *Vision*. W. H. Freeman and Company, 1982.

[Mic95] M. Michaelis. *Low Level Image Processing Using Steerable Filters*. PhD thesis, Institute of Computer Science, University Kiel, Germany, 1995. also available as Tech. Report No. 9716.

[MPS98] R. Manduchi, P. Perona, and D. Shy. Efficient deformable filter banks. *IEEE Trans. Signal Processing*, 46(4):1168–1173, 1998.

[MS94] M. Michaelis and G. Sommer. Junction classification by multiple orientation detection. In *Proc. Third European Conference on Com-*

puter Vision, volume I, pages 101–108, Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 800, 1994.

[MS95] M. Michaelis and G. Sommer. A Lie group approach to steerable filters. *Pattern Recognition Letters*, 16:1165–1174, 1995.

[Nal93] V. S. Nalwa. *A Guided Tour of Computer Vision*. Addison-Wesley Publishing Company, 1993.

[NCN97] J. M. Nash, J. N. Carter, and M. S. Nixon. Dynamic feature extraction via the velocity Hough transform. *Pattern Recognition Letters*, 18:1035–1047, 1997.

[NE86] H.H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[Niy95] S. A. Niyogi. Detecting kinetic occlusion. In *Proc. Int. Conf. on Computer Vision*, pages 1044–1049, Boston, MA, June 20-23, 1995.

[NU88] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special Functions of Mathematical Physics*. Birkhaeuser Verlag, 1988.

[OS75] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice-Hall, Inc., 1975.

[OT98] Y. Hel Or and P.C. Teo. Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9(1):83–95, 1998.

[Per92] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. *Image and Vision Computing*, 10(10):663–672, 1992.

[Per95] P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(5):488–499, 1995.

[PG90] T. Poggio and F. Girosi. Networks for approximatation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[PGH98] L. Parida, D. Geiger, and R. Hummel. Junctions: Detection, classification, and reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(7):687–698, 1998.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[Ris83] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[RO97] B. Robbins and R. Owens. 2d feature detection via local energy. *Image and Vision Computing*, 15:353–368, 1997.

[Roh92] K. Rohr. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3):213–230, 1992.

[Roh97] K. Rohr. On the precision in estimating the location of edges and corners. *Journal of Mathematical Imaging and Vision*, 7:7–22, 1997.

[Sch89] B. G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(10):1010–1027, 1989.

[SF96] E. P. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. *IEEE Trans. Image Processing*, 5(9):1377–1382, 1996.

[SFAH92] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multi-scale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, 1992.

[Sim94] E.P. Simoncelli. Design of multi-dimensional derivative filters. In *IEEE Int. Conf. on Image Processing*, volume I, pages 790–794, Austin, TX, November 13-16, 1994.

[SKJ97] H. Scharr, S. Körkel, and B. Jähne. Numerische Isotropieoptimierung von FIR-Filtern mittels Querglättung. In *DAGM Symposium Mustererkennung*, pages 367–374, Braunschweig, Germany, Sep. 15-17, 1997.

[SM91] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 289–295, Maui, Hawaii, June 3-6, 1991.

[SM97] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, Puerto Rico, June 17-19, 1997.

[SM98] J. Shi and J. Malik. Motion segmentation using normalized cuts. In *Proc. Int. Conf. on Computer Vision*, pages 1154–1160, Bombay, India, Jan. 4-7, 1998.

[SMH98] G. Sommer, M. Michaelis, and R. Herpers. The SVD approach for steerable filter design. In *IEEE International Symposium on Circuits and Systems*, volume V, pages 349–353, Monterey, CA, May 31-June 3, 1998.

[SS95] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings (SIGGRAPH 95)*, pages 161–172, 1995.

[SS97] C. Sun and J. Sherrah. 3D symmetry detection using the extended gaussian images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2):164–168, 1997.

[Teo98] P.C. Teo. *Theory and Applications of Steerable Functions*. PhD thesis, Department of Computer Science, Stanford University, 1998.

[TMB85] W. B. Thompson, K. M. Mutch, and V. A. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(4):374–383, 1985.

[TO98] P.C. Teo and Y. Hel Or. Lie generators for computing steerable functions. *Pattern Recognition Letters*, 19(1):7–17, 1998.

[TO99] P.C. Teo and Y. Hel Or. Design of multiparameter steerable functions using cascade basis reduction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(6):552–556, 1999.

[WA93] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 361–366, New York, NY, June 15-17, 1993.

[WA96] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 321–326, San Fransisco, CA, June 18-20, 1996.

[WB97] X. Wu and B. Bhanu. Gabor wavelet representation for 3-d object recognition. *IEEE Trans. Image Processing*, 6(1):47–64, 1997.

[Wei97] Y. Weiss. Smoothness in layers: Motion segmentation using non-parametric mixture estimation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, June 17-19, 1997.

[WFKvdM97] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[WK93] S.F. Wu and J. Kittler. A gradient-based method for general motion estimation and segmentation. *Journal of Visual Communication and Image Representation*, 4:25–38, 1993.

[WS99] J. Weickert and C. Schnörr. Räumlich-zeitliche Berechnung des optischen Flusses mit nichtlinearen flußabhängigen Glattheitstermen. In *DAGM Symposium Mustererkennung*, pages 317–324, Bonn, Germany, Sep. 15 - 17, 1999.

[Wür97] R. P. Würtz. Object recognition robust under translations, deformations, and changes in background. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):769–774, 1997.

[XOK90] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.

[XS97] Y. Xiong and S. A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. *International Journal of Computer Vision*, 24(1):25–59, 1997.

[YDBS99] W. Yu, K. Daniilidis, S. Beauchemin, and G. Sommer. Detection and characterization of multiple motion points. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 171–177, Fort Collins, CO, June 23-25, 1999.

[YDS98a] W. Yu, K. Daniilidis, and G. Sommer. Junction characterization using polar pyramid. In *DAGM Symposium Mustererkennung*, pages 69–76, Stuttgart, Germany, Sep. 29 - Oct. 1, 1998.

[YDS98b] W. Yu, K. Daniilidis, and G. Sommer. Low-cost junction characterization using polar averaging filters. In *IEEE Int. Conf. on Image Processing*, volume III, pages 41–44, Chicago, IL, October 4-7, 1998.

[YDS98c] W. Yu, K. Daniilidis, and G. Sommer. Rotated wedge averaging method for junction characterization. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 390–395, Santa Barbara, CA, June 23-25, 1998.

[YDS00a] W. Yu, K. Daniilidis, and G. Sommer. Eliminating outliers in motion occlusion analysis. In *DAGM Symposium Mustererkennung*, pages 373–380, Kiel, Germany, Sep. 13 - 15, 2000.

[YDS00b] W. Yu, K. Daniilidis, and G. Sommer. A new 3d orientation steerable filter. In *DAGM Symposium Mustererkennung*, pages 203–212, Kiel, Germany, Sep. 13 - 15, 2000.

[YDS01] W. Yu, K. Daniilidis, and G. Sommer. Approximate orientation steerability based on angular Gaussians. *IEEE Trans. Image Processing*, 2001. to appear.

[YSD00] W. Yu, G. Sommer, and K. Daniilidis. Multiple motion analysis using 3d orientation steerable filters. Technical Report 2008, Institute of Computer Science, University Kiel, Germany, 2000.