

Adaptive Multidimensional Filtering

Leif Haglund

To Christina, Lina and Maja

Abstract

This thesis contains a presentation and an analysis of adaptive filtering strategies for multidimensional data. The size, shape and orientation of the filter are signal controlled and thus adapted locally to each neighbourhood according to a predefined model. The filter is constructed as a linear weighting of fixed oriented bandpass filters having the same shape but different orientations. The adaptive filtering methods have been tested on both real data and synthesized test data in 2D, e.g. still images, 3D, e.g. image sequences or volumes, with good results. In 4D, e.g. volume sequences, the algorithm is given in its mathematical form. The weighting coefficients are given by the inner products of a tensor representing the local structure of the data and the tensors representing the orientation of the filters.

The procedure and filter design in estimating the representation tensor are described. In 2D, the tensor contains information about the local energy, the optimal orientation and a certainty of the orientation. In 3D, the information in the tensor is the energy, the normal to the best fitting local plane and the tangent to the best fitting line, and certainties of these orientations. In the case of time sequences, a quantitative comparison of the proposed method and other (optical flow) algorithms is presented.

The estimation of control information is made in different scales. There are two main reasons for this. A single filter has a particular limited pass band which may or may not be tuned to the different sized objects to describe. Second, size or scale is a descriptive feature in its own right. All of this requires the integration of measurements from different scales. The increasing interest in wavelet theory supports the idea that a multiresolution approach is necessary. Hence the resulting adaptive filter will adapt also in size and to different orientations in different scales.

Acknowledgements

I would like to express my gratitude to those who have helped me in pursuing this work:

To Gösta Granlund, Professor at Computer Vision at LiTH for introducing me to the exciting field of computer vision and for giving me the opportunity to work in his group. His ideas about vision has flavored most of the algorithms in this thesis.

To Dr. Hans Knutsson, for always finding time for discussions and for sharing his immense scientific knowledge. He has been a constant source of new ideas, explanations and comments, without which this thesis would have been considerably thinner. His comments regarding the presentation of the work in this thesis has also improved the final result.

To all the members of the Computer Vision Laboratory for the stimulating and friendly atmosphere in the group and for numerous comments and discussions on scientific and philosophical subjects.

To Carl-Fredrik Westin for all the time he has spent reading drafts of this thesis. His comments, regarding both scientific and editorial issues, have improved the final result considerably.

To Tomas Landelius for his implementations of the applications in Chapter 8.

To Catharina Holmgren, Professor Roland Wilson, Dr. Mats Andersson, and Dr Håkan Bårman for proof-reading different parts of the manuscript.

To the Swedish National Board for Technical Development and Prometheus-Sweden, for its financial support of this work.

Finally to somebody I haven't seen much during the last hectic months, but who means very much to me. Thank you, Christina, for your patience, love and support, and most of all for just being. By taking the utmost care of the daily life and of our two daughters Lina and Maja, you made it all possible.

Contents

1	Introduction	6
1.1	Organization of the Thesis	6
1.2	Previous Publications	8
1.3	Notations	9
I	Filter Design and Scale Analysis	11
2	Background	13
3	Filter Design and Wavelets	18
3.1	Quadrature Filters	18
3.2	Radial Filter Functions	20
3.2.1	General Design Principles	20
3.2.2	Wavelet Transform	20
3.2.3	The Lognormal Filter	25
3.2.4	Conclusion	31
3.A	Filter Optimization	31
4	On the Use of Phase in Scale Analysis	34
4.1	Phase Estimation in Images	34
4.1.1	Representation	37
4.1.2	Experimental Results	39
4.2	Feature Scale-Space Utilizing Phase	39
4.2.1	Scale Space Clustering	41
4.2.2	Spatial Frequency Estimation	47
4.2.3	Conclusions	54
4.A	Gaussian Resampling of Images	58
4.A.1	Subsampling	58
4.A.2	Interpolation to a Higher Sampling Rate	59

II	2D — Adaptive Filtering	61
5	Adaptive Filtering of 2D Images	63
5.1	Representation and Estimation of Orientation	63
5.1.1	Orientation Representation	65
5.1.2	Orientation Estimation	65
5.2	Orientation Adaptive Filtering	68
5.2.1	Implementation	71
6	Scale Adaptive Filtering	74
6.1	The Algorithm	74
6.1.1	Overview	74
6.1.2	Generation of the Bandpass Pyramid	76
6.1.3	Estimation of Orientation	78
6.1.4	Consistency of Orientation	78
6.1.5	Determination of Parameters	79
6.1.6	Adaptive Filtering	83
6.1.7	Reconstruction	84
6.2	Results	86
6.A	Calculations of the Energy in the Bandpass Pyramid	95
6.B	The Transfer Function	96
III	3D — Adaptive Filtering	97
7	Orientation Representation and Estimation in 3D	99
7.1	Orientation Representation	99
7.1.1	Implementation of the Orientation Algorithm	100
7.1.2	Evaluation of the Representation Tensor	102
7.1.3	Accuracy of the Orientation Estimate	102
7.2	Optical Flow Estimation	106
7.2.1	Velocity Estimation	108
7.2.2	Spatio-Temporal Channels	109
7.2.3	Quantitative Error Measurements	112
7.A	Filtering of Interlaced Video Signals	119
8	Applications in Structure from Motion	121
8.1	Extraction of Focus of Expansion	121

8.2	Motion Stereo Algorithm	124
8.2.1	Results	124
9	Adaptive Filtering of 3D Signals	128
9.1	The Algorithm	128
9.2	Implementation	134
9.3	Results	135
IV	4D — Adaptive Filtering	143
10	Filtering of 4D Signals	145
10.1	Orientation Representation and Estimation	145
10.2	Adaptive Filtering of 4D Signals	147

Chapter 1

Introduction

The main theme of this thesis is to present an adaptive filtering structure based on estimated parameters for local models. The models are based on the observation that complex high dimensional signals locally have very few degrees of freedom. For images it is believed that local one-dimensionality, i.e. a small neighbourhood varying in just one orientation, is a generally useful model for vision systems.

The chosen approach is inspired and guided by previous work at the Computer Vision Laboratory (CVL) at Linköping University, [39, 67, 40, 66, 97, 62, 14]. The main line of research at CVL is the design of robust operations which can work in a hierarchical, or modular, system. Each module of the system is designed to produce estimates that can be used directly by other modules. Larger system can then be designed using these modules as building blocks. In multi-level systems the information representation becomes crucial. A main guide-line for the work at CVL is that the behavior of the developed modules should be robust and that ordinary signal processing tools, e.g. differentiation and averaging, should be meaningful to apply. This implies that small changes in the input signal should not cause large changes in the output, which may seem a simple rule, but it has profound implications. One consequence of the robustness rule is the use of tensors as a general way of representing information, [59, 8, 94]. In some cases 1:st order tensors, i.e. vectors, may suffice. The norm of the tensor represents the strength of the represented event. The Eigenvectors and Eigenvalues of the tensor give a full representation of the type and the certainty of the event. An important feature of this representation is that the norm of the difference between two tensors will indicate how different the represented events are.

In this thesis the tensor representation is utilized for representing the control information in an adaptive filtering strategy. The robustness and accuracy of the estimated tensor in the case of 3D signal spaces is also evaluated.

1.1 Organization of the Thesis

The thesis is divided into four different parts. The first part presents the filters used and also describes a scale analyzing scheme for 2D images. The following three parts describe the adaptive filter algorithm for 2D, 3D and 4D signal spaces.

In each of these three parts the estimation of local orientation is described and examples of other applications where the estimated features have been used are given. The results are in a number of cases compared to results of other (widely) used algorithms. Problems associated with signals of different dimensionality are in many ways qualitatively different. It is, for this reason, natural to partitioning the thesis depending on the dimensionality of the signal space. However, some basic assumptions are the same for signals of any dimension, and some of the assumptions involved in the 3D and 4D parts are presented in the 2D part.

Part I – Filter Design

In the first part the quadrature filters used in the estimation throughout the thesis are presented in a wavelet theory context, [87, 25]. An extension of the one-dimensional wavelet transform to two dimensions is suggested giving a rotational unbiased output. The two-dimensional transform is used to perform scale analysis of images, [100, 18, 70, 99, 73]. The use of phase from the quadrature filters, [67, 32], is utilized in the scale analysis and a continuous representation of the phase for 2D signals is given. Finally an application in texture analysis is described.

Part II – 2D Adaptive Filtering

In the second part the estimation of local orientation on 2D images, and the representation as a tensor field is described, [59, 60]. The estimated orientation information is used as contextual control of an orientation adaptive filter, [66]. Although the adaptive filter has many degrees of freedom it can be constructed as a weighted sum of the output from fixed filters. Due to the tensor representation the weighting coefficients are calculated as a simple inner product of the tensor describing the neighborhood and the tensors corresponding to the fixed filters, [65]. The orientation adaptive filter is then extended to work in different scales. Working with different scales also enables a better decision of whether estimated local energy originates from noise or from the image signal. A scheme for estimating the noise level in images is presented and used for parameter setting in the adaptive filtering. This scheme has proven to work well on a wide variety of images.

Part III – 3D Adaptive Filtering

The third part contains the estimation of orientation for 3D signals, such as image sequences or image volumes. The estimated orientation is used for optical flow calculation and the results are compared to other algorithms. This information is also used for focus of expansion estimation in the case of a translating camera. The local orientation is represented as a 3 by 3 tensor, [59], which is used to adapt a 3D filter in this direction, [65]. The adaptive filtering algorithm is shown to perform very well, even in very noisy situations, on both image sequences and image volumes.

Part IV – 4D Adaptive Filtering

In the last part of the thesis a theoretical extension of the adaptive filtering algorithm to 4D signal spaces is presented. The representation and estimation of orientation of local structures in 4D are discussed. As in 3D, this information is used as control for orientation adaptive filtering.

1.2 Previous Publications

This thesis is a compilation and an extension of work previously documented in the following publications:

- L. Haglund, H.Knutsson and G.H. Granlund.
On Phase Representation of Image Information.
The 6th Scandinavian Conference on Image Analysis pp 1082–1089
Oulu, Finland, June 1989
- L. Haglund, H.Knutsson and G.H. Granlund.
Scale Analysis Using Phase Representation.
The 6th Scandinavian Conference on Image Analysis pp 1118–1125
Oulu, Finland, June 1989
- J. Wiklund, L. Haglund, H. Knutsson and G. H. Granlund
Time Sequence Analysis Using Multi-Resolution Spatio-Temporal Filters
The 3rd International Workshop on Time-Varying Image Processing and Moving Object Recognition pp 258–265
Florence, Italy, May 1989
- L. Haglund, H. Bårman and H. Knutsson
Estimation of Velocity and Acceleration in Time Sequences
Theory and Applications of Image Analysis pp 223–236
Publisher: World Scientific Publishing Co, May 1992
- H. Bårman, L. Haglund, H. Knutsson and G. H. Granlund
Estimation of Velocity, Acceleration and Disparity in Time Sequences
Proceedings of IEEE Workshop on Visual Motion pp 44–51
Princeton, NJ, USA, October 1991
- H. Knutsson, L. Haglund, H. Bårman and G. H. Granlund
A Framework for Anisotropic Adaptive Filtering and Analysis of Image Sequences and Volumes
Proceedings ICASSP-92 pp 469–472
San Francisco, CA, USA, March 1992

1.3 Notations

The following notations are used¹:

Lowercase letters are used to denote scalars, e.g. s .

Lowercase letters in boldface are used to denote vectors, e.g. \mathbf{v} , while lowercase letters with subscripts are used for individual vector elements, e.g. v_i . The norm of a vector \mathbf{v} is denoted v .

Uppercase letters in boldface are used for tensors of order 2 (and matrices), e.g. \mathbf{T} . The elements are denoted with the uppercase letter and subscripts, e.g. T_{ij} . The norm of a tensor \mathbf{T} is denoted by T .

\mathbf{e} Eigenvector.

λ Eigenvalue. λ_1 is the largest.

\mathbf{n} Filter direction in frequency domain.

q Filter output magnitude.

θ Phase angle.

\mathbf{q} Filter output (with phase angle).

ω Frequency coordinate vector.

\mathbf{x} Spatial coordinate vector.

φ Spatial orientation.

¹Some exceptions from the general notation guide can be found, in these cases the variables are defined explicitly to avoid misunderstandings.

Part I

Filter Design and Scale Analysis

Chapter 2

Background

In the last few years the wavelet theory and their transform have gained increasing attention, [56], in various signal processing fields. Wavelet theory could be said to be a unification of many “old” ideas, and the purpose of this part of the thesis is to reformulate some ideas in terms of this common framework.

Wavelet-like transforms, e.g. scale pyramids and scale-space, have been widely used in the computer vision community over the last decade. The original reason for using scale pyramids was to perform an efficient analysis, but in later years there has been an increasing support for the idea that analysis at multiple scales or in scale-space could provide important means of image description on which model-based image processing could be based. Many researchers have published work within the field. To mention some of them, without claiming to be comprehensive:

- Marr, [78], introduced the concept of scale-pyramids as early as in the mid-seventies. The frequency difference between two adjacent levels in the pyramid was an octave or, in other words, a subsampling of a factor 2 was used. The result of this is a logarithmic sampling in the scale dimension. The inspiration to this was that approximately the same performance had been detected in biological vision, in the so called frequency channel theory. He claimed that visually interesting features must exist in a rather broad band of scales, i.e. over many levels in the pyramid.
- Granlund, [39], suggested the use of a set of Cartesian separable Gabor-Gaussian filters in different scales for orientation estimation in images. This type of filters have been extensively used in a number of different applications, and have lately been termed Morlet wavelets.
- Knutsson, [67], introduced the a polar separable lognormal filter function, see also Section 3.2.3, obtaining a natural separation of scale and orientation. These filters were used on several scales with typical applications being orientation estimation, spatial frequency estimation and texture segmentation.
- Witkin, [100], extended the pyramid concept to a continuous so-called scale-space by using Gaussian filtering. The description he used was based on zero-crossings of the Laplacian of Gaussian filters. By varying the standard deviation of the Gaussian all scales will be covered by Laplacians. He stressed two things:

(1) Identity: Zero-crossings observed at different scales but lying on a common zero-contour, meaning that they correspond to a small scale change in scale-space, arise from the same underlying event.

(2) Localization: The true location of an event is its position in the finest available scale.

- Koenderink, [70, 69], has by using Gaussian filtering proved that the classical heat equation can correlate scale changes with changes in spatial dimensions. He embedded the original image in a one parameter family of derived images, where the parameter is resolution or scale. A study of this family makes it possible to use all scales simultaneously, which is a requirement if no a priori knowledge about the “right” resolution is available and it is desirable to retain all possible structures.
- Pizer, [29], has implemented Koenderink’s ideas in an impressive way, nearly in video-rate. By using the symmetry axis transform in multiple resolutions as description of images, good results have been achieved, mainly on medical images.
- Lindeberg, [73], has theoretically extended Koenderink’s ideas to discrete signals. He has also defined a representation, “the scale space primal sketch”, for the relations between different scale levels. The theory is applied to extracting image structures and detecting on what scale they appear.
- Crowley, [23], bases his analysis on differences of Gaussians, using the so called DOLP transform. Following what he calls peaks and ridges in the family of Laplacian images, which is the result of the DOLP transform, through many scales, gives a description of the original image. The final description is taken as the maximum over scale of each peak or ridge.
- Hoff and Olsen, [81], [51] are two examples of the use of scale analysis in stereo algorithms. The scale analysis is in this case used as a guidance to solve the stereo correspondence problem. By doing a tracking of events, i.e. zero-crossings of a Laplacian, from coarse to fine scale, the calculation and complexity of the correspondence problem is reduced significantly.
- Wilson and Calway, [96], has extended the use of the short-time Fourier transform, with several window sizes, to images in the so called Multiresolution Fourier Transform. The transform is a general tool for image analysis in multiple scales and has for example been applied to curve segmentation.

In order to give some understanding of how images behave when they go through a blurring process, a short description will be given. When resolution is decreased, the images become less articulated because the extremes, bright and dark “blobs”, disappear one after the other. This erosion of structure is a process that is similar in each case and there are three main reasons for this behavior:

1. Two extrema float together and annihilate.

scale of interest	events
under 5 cm	gravel and pebbles
5 - 50 cm	bricks
50 cm - 5 m	windows and doors
5 - 50 m	separate houses
50 - 500 m	blocks
over 500 m	the whole town

Table 2.1: *The correlation between scale and events in a picture of a town.*

2. Two extrema float together and one “swallows” the other. The result is that one large extremum continues in the blurring process.
3. An extremum and a saddle point float together and the extremum disappears.

Consequently each blob has a limited range in which it manifests itself. To understand this, think about a picture of a town. In this picture different events exist in different scales, see Table 2.1. The detection of, say, bricks shall then be in the scale interval between 5 cm and 50 cm. Finer or larger scales must not interfere with the detection procedure.

The phase from the wavelet transforms has during the last few years gained increasing interest in the computer vision community. The phase from Gabor-like filters has been applied in a number of different applications [44, 45, 34, 35, 93, 72, 95, 19, 98, 16]. The output from a quadrature filter is an analytic function which can be written in terms of an even real part and an odd imaginary part, as described in Chapter 3. The analytical function can also be written as an amplitude and an argument. The amplitude is a measure of the local energy (local both in spatial and frequency domains) and the argument is the phase.

The motivations to use phase are:

- The phase is insensitive to both mean luminance and contrast.
- The phase is a continuous variable, i.e. it can measure changes much smaller than the spatial quantization, giving subpixel accuracy, without subpixel representation of the image.
- The phase is also stable against scaling up to 20 %, [35].
- The phase is generally a very stable feature in scale space.

and the applications has for example been guidance for preattentive gaze control, [93], disparity estimation, [72, 35, 95, 98], image velocity estimation, [34], and feature modeling, [16, 19].

Among the motivations above the last one might be the most important motivation for the scale space clustering scheme in Section 4.2. In order to illustrate the phase behavior in scale space an 1D line from the well-known “Lenna” test image will be used, see Figure 2.1.

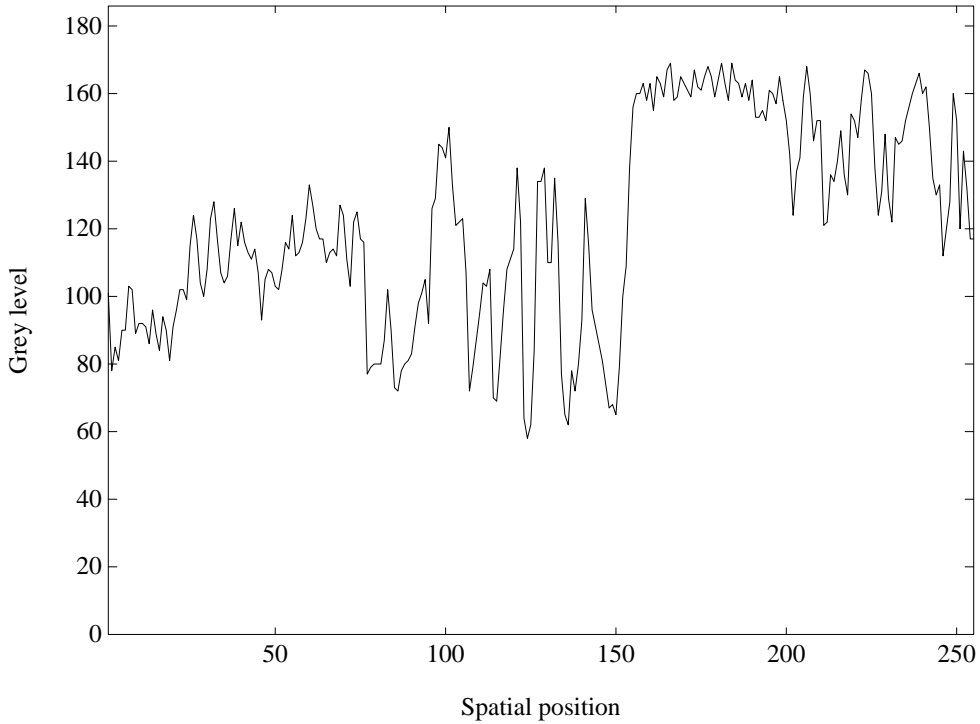


Figure 2.1: *The signal used to illustrate the scale behavior, Figure 2.2, of the lognormal filter.*

Figure 2.2 contains isophase contours from a lognormal filter, Eq. 3.16. It describes the phase behavior of the lognormal filter in scale-space. This type of figure will be named phaseogram. In the figure a continuous scale parameter has been used. The scale has the same meaning as in the wavelet community. The phase is stable over scale when the contours are vertical. It is clear that this is the case in most of the phaseogram. There are, however, some points in the phaseogram where the contours turn and become horizontal. These points are called singular points [32] because they are singular points of the analytical function.

The reason for their appearance can intuitively be explained in the following way. At low scales, i.e. high resolution, the phase cycle of the filter output has small spatial support; on the other hand at high scales, i.e. low resolution, the spatial support is much larger. If the signal has a limited support this means that the number of cycles must decrease with increasing scale. In the phaseogram, Figure 2.2, this can easily be seen by just looking at the variation of the density of the isophase curves when going from low to high scales. The decrease in the number of cycles occurs in these singular points.

In Chapter 4 a continuous representation of the local phase in images is presented and described. An extensions of the wavelet transform to 2D signal spaces, i.e. images, is suggested in Section 4.2 utilizing the phase of the transform. This 2D transform is used as a scale space clustering tool and applied as a spatial frequency estimator. In the application the filters are chosen to have constant relative bandwidth. The bandwidth $B = 1.5$ can be motivated from physiological studies.

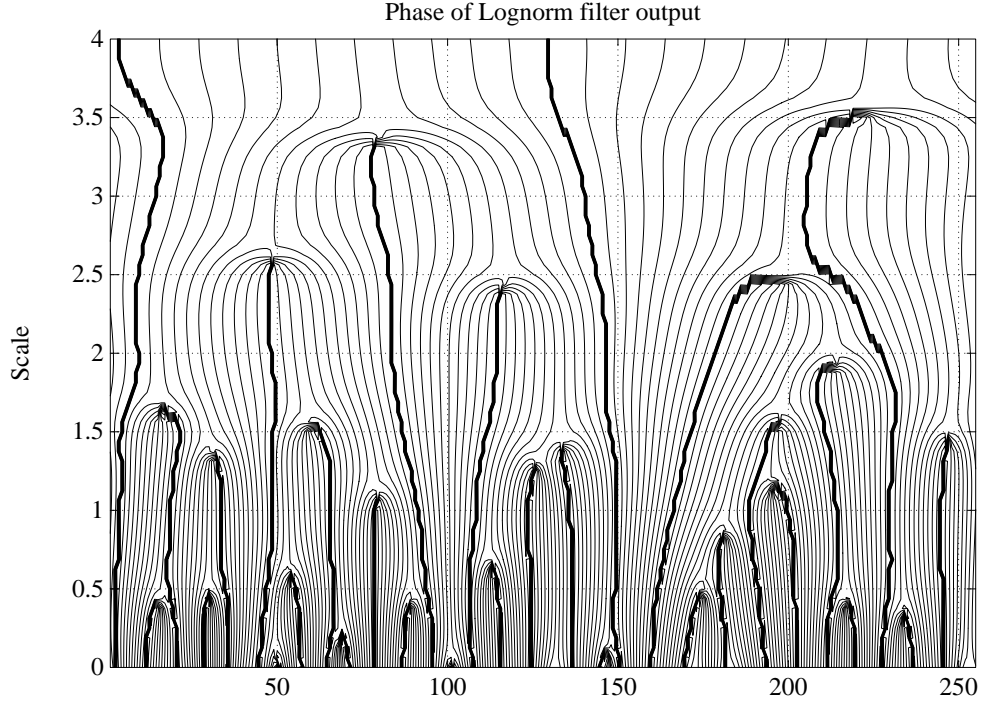


Figure 2.2: *Isophase contours from a lognormal filter. The broad, dark, lines are due to phase wrap around.*

Measurements both from adaption experiments on human beings and directly on striate cortex in macaque monkeys have in agreement with each other showed nearly constant relative bandwidth in biological visual systems, see [28] and [38]. The measured bandwidths range from 1.5 to 2 octaves. The calculations in Section 3.2.3 for an octave based wavelet transform also shows that this choice is mathematically acceptable.

Chapter 3

Filter Design and Wavelets

3.1 Quadrature Filters

It is well known that a real-valued image, $i(x, y)$, has a hermitian Fourier transform, [17], $I(\omega_x, \omega_y)$, i.e.

$$I(\omega) = I^*(-\omega) \quad (3.1)$$

where $*$ denotes the complex conjugate and ω is the frequency vector. It follows from this property that the energy contribution is even.

$$|I(\omega)|^2 = |I(-\omega)|^2 = (\text{Re}[I(\omega)])^2 + (\text{Im}[I(\omega)])^2 \quad (3.2)$$

From the evenness of the energy function, a measurement in a particular section of the Fourier domain can be obtained by designing filters that only measure the energy in one half plane. One way to perform the estimation is to use quadrature filters.

A typical frequency function, $Q_k(\omega)$, first suggested in [67], for a quadrature filter is

$$Q_k(\omega) = \begin{cases} F(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^{2A} & \text{if } (\hat{\omega} \cdot \hat{\mathbf{n}}_k) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

where

$$\omega = ||\omega||$$

A is a parameter which specifies the angular bandwidth and

\mathbf{n}_k is the main direction of the filter.

In more traditional terms of image processing the filter of Eq. 3.3 can be separated into line and edge detectors. Eq. 3.2 shows that it is possible to estimate the energy contribution separately from the real and the imaginary part of the local Fourier transform. The real part of the transform corresponds to even functions, such as lines, and the imaginary part corresponds to odd functions, such as edges. The equation for quadrature filters, Eq. 3.3, can be written in terms of an odd part, indexed o , and an even part, indexed e .

$$\begin{aligned} Q_k(\omega) &= H_{ke}(\omega) + H_{ko}(\omega) \\ H_{ke}(\omega) &= H_{ke}(-\omega) = \frac{1}{2}F(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^{2A} \\ H_{ko}(\omega) &= -H_{ko}(-\omega) = \frac{1}{2}F(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^{2A}\text{sign}[\hat{\omega} \cdot \hat{\mathbf{n}}_k] \end{aligned} \quad (3.4)$$

Using a quadrature approach to filter design has a number of advantages. The principal one is the well-defined behavior in the common frequency band of the line and edge detectors. This behavior may also be used to define a general signal phase of multidimensional signals, see Section 4 and [67]. Another important advantage is that “phase-independence” can be obtained in the output of different operations, e.g. orientation estimation. This means that the output will be equally strong for cosine (line)-like inputs as for sinus (edge)-like inputs. There is also physiological evidence for quadrature-like filters in biological vision systems, [76, 82, 83].

The reason for choosing polar separable filter functions is that an intuitive feeling for the filter parameters can be obtained. The angular function is closely related to the orientation of the filter and thus to the local orientation in the neighbourhood. The radial function, on the other hand, determines in what scale the filter has its sensitivity. By making the filter functions polar separable, it is easy to change these aspects of the estimation independently, with easy predictable effects on the filter sensitivity.

The choice of angular function in Eq. 3.3 and Eq. 3.4 is due to the interpolation and smoothness properties of trigonometric functions. The parameter A controls the tuning of the orientation sensitivity. A larger value of A gives a narrower angular bandwidth and thus the possibility to estimate the orientation more exactly, but this at the cost of more filters. The condition which increases the number of filters is the requirement of rotational invariance in the estimation. Rotational invariance means here that the exactness and the strength of the estimation should be invariant under rotation of the input. For example, with $A = 1$ in a two-dimensional image, at least three quadrature filters are required to get an unbiased output. The relationship between A and the number of filters, k , is $k \geq A + 1$, the proof is found in [67]. This requirement is in a similar context called rotational invariance [24].

The choice of the radial frequency $F(\omega)$ is based on the scale of interest. $F(\omega)$ has the character of a bandpass filter (see the next section for a more thorough investigation).

Applying these filters as a convolution between their spatial transforms, h_{ke} and h_{ko} , and the input image, $i(x, y)$, will form outputs q_{ko} and q_{ke}

$$\begin{aligned} q_{ke}(x, y) &= h_{ke}(x, y) * i(x, y) \\ q_{ko}(x, y) &= -j \cdot h_{ko}(x, y) * i(x, y) \end{aligned} \tag{3.5}$$

where $j = \sqrt{-1}$.

The energy contribution from the filter k , E_k , is given by

$$E_k = q_{ko}^2 + q_{ke}^2 \tag{3.6}$$

By combining the outputs q_k it is possible to design robust algorithms for orientation estimation [67].

3.2 Radial Filter Functions

3.2.1 General Design Principles

When designing filters for image processing the uncertainty principle must be kept in mind, [67, 97]

$$\Delta x \cdot \Delta \omega \geq \text{constant} \quad (3.7)$$

where x is the spatial variable and ω is the frequency variable.

The spreads $\Delta \omega$ and Δx are the standard deviations in the Fourier and the spatial domain respectively.

In order to minimize the simultaneous spread of the filters in both frequency and spatial domains a Gaussian approach is preferable. This is, however, not possible to combine with quadrature filters, [67]. The restrictions that must be fulfilled in order to design small spatial quadrature filters, which still behave properly in the Fourier domain, are

1. The DC-component the Fourier domain must be zero.
2. The value at the filter border, $\omega = \pi$, must be zero.

A consequence of the uncertainty principle is that a narrow filter in one domain will be wide in the corresponding dual domain. A wide filter in the Fourier space is thus appropriate to realize as a spatial convolution kernel, since only a small number of coefficients are needed. On the other hand, a wide filter has a long tail which may be hard to combine with the second demand above.

The reason for these restrictions is to avoid discontinuities in the odd part, or in the edge detector, of the filter. A discontinuity in the frequency domain gives rise to ringing in the spatial domain. An understanding of how these discontinuities can appear can be gained by looking at Figure 3.1, which shows a well-behaved quadrature filter plotted in the frequency domain. In Figure 3.2 a bad quadrature filter is plotted. Neither the first nor the second requirement stated above is fulfilled. Note the discontinuity in the odd part of the filter.

3.2.2 Wavelet Transform

Wavelet theory is a unification of similar ideas from different fields. In the mid-eighties, three French scientists, Morlet (geophysicist), Grossmann (theoretical physicist) and Meyer (mathematician), built a strong mathematical foundation based on the idea of looking at signals at different scales and various resolutions. In the field computer vision, Mallat and Daubechies have established connections to discrete signal processing, e.g. [25, 75].

The general reason for transforming or filtering a signal is to extract specific features from the signal. This means that the filter, in the ideal case, should be invariant to all other variations of the signal but the interesting feature, to which it should be equivariant. Hence, the transformed or filtered signal should vary only with the interesting feature and not with anything else.

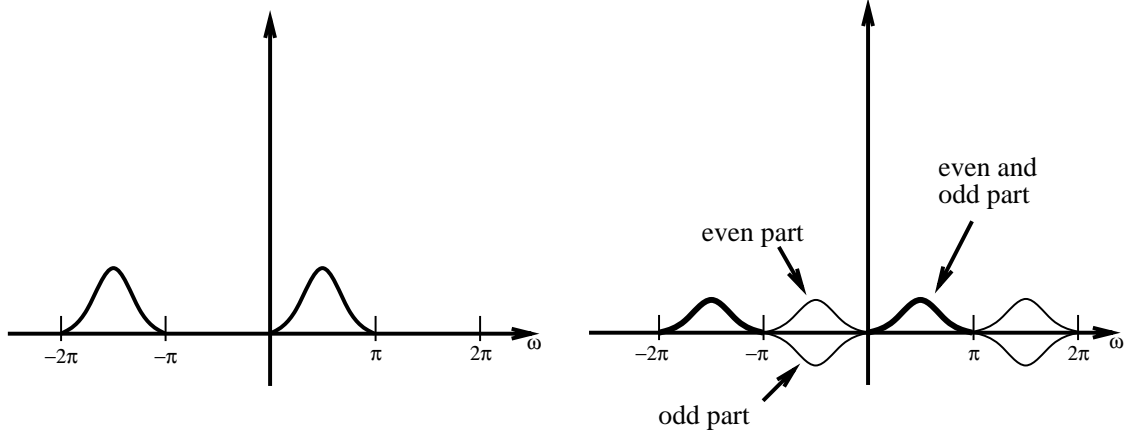


Figure 3.1: *Plot of a well-behaved quadrature filter both as one filter and split into odd and even parts. The periodicity of the Fourier domain corresponds to a discrete spatial domain.*

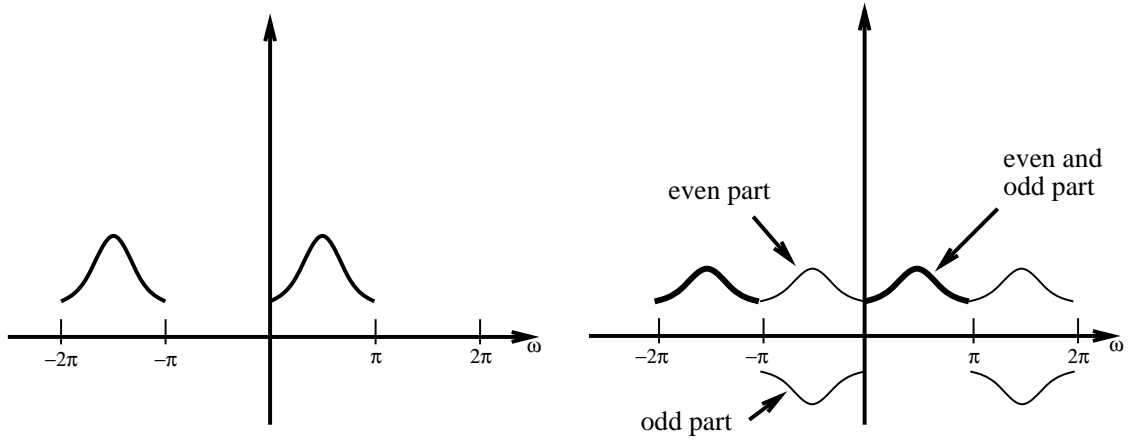


Figure 3.2: *Plot of a bad quadrature filter both as one filter and split in odd and even parts. Note the discontinuities in the odd part of the filter.*

The Fourier transform is an example of a transform where the interesting feature to extract is the frequency. The assumption is that the signal is stationary, e.g. sinewaves. If the signal is non-stationary, any abrupt change of the signal will be spread over the whole frequency axis and the spatial position of the discontinuity will be impossible to retain from the Fourier coefficients. The Fourier transform is apparently not sufficient for analyzing such signals.

The Short Time Fourier Transform, or windowed Fourier transform, is one way to modify the Fourier transform for better performance on non-stationary signals. There are, however, many possibilities to choose the windowing function. One widely used window function is the Gabor function, $G(\omega)$ or $g(x)$.

$$\begin{aligned} G(\omega) &= e^{-(\omega-\omega_0)^2/2\sigma_\omega^2} \\ g(x) &= e^{jx\omega_0} \frac{\sigma_\omega}{\sqrt{2\pi}} e^{-x^2\sigma_\omega^2/2} \end{aligned} \quad (3.8)$$

Here the window function in the frequency domain is a shifted Gaussian and spatially

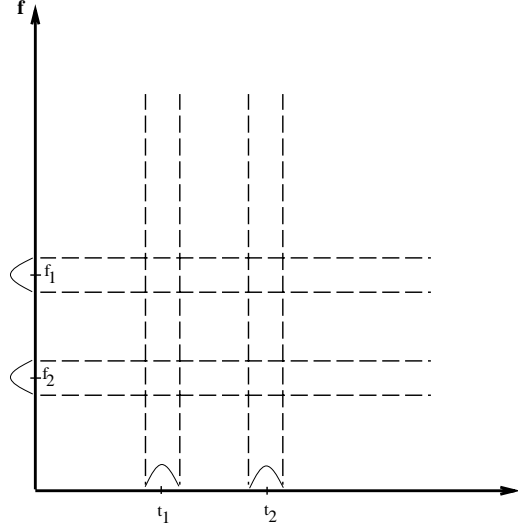


Figure 3.3: *Linear partitioning of the frequency domain, where the filter functions for two filters are indicated both spatially and in the frequency domain.*

a modulated Gaussian, giving a filter that is well concentrated in both domains, [37, 39].

To have equal sensitivity for all frequencies there is an obvious need for several filters. There are many ways of partitioning the frequency domain. Originally Gabor suggested a uniform splitting of the time-frequency domain, see Figure 3.3 and Figure 3.4.

In wavelet theory, a logarithmic partitioning is desirable, see Figure 3.4. This is achieved by scaling and translating a “mother wavelet”. This idea has been used in vision for several years, e.g. Laplacian pyramids, [18], and the spatial frequency channel theory of biological vision, [20].

The logarithmic partitioning is equivalent to using filters with constant relative bandwidth, or “constant-Q” analysis

$$\frac{\Delta\omega}{\omega_0} = \text{constant} \quad (3.9)$$

When the relation in Eq. 3.9 is fulfilled, the bandwidth, $\Delta\omega$, will vary with the center frequency of the filter. In spite of the uncertainty principle, it is now possible to get arbitrarily good time or **space** resolution at high frequencies, [99, 84]. The same holds for low frequencies, i.e the low frequency resolution can be made arbitrarily good. The hypothesis is that high frequency bursts are of short duration, while low frequency components have long duration. In images this hypothesis implies that “interesting” features have fixed shape but unknown size. This is a more realistic model than for example the global Fourier transform, where all frequencies are supposed to have infinite support in space.

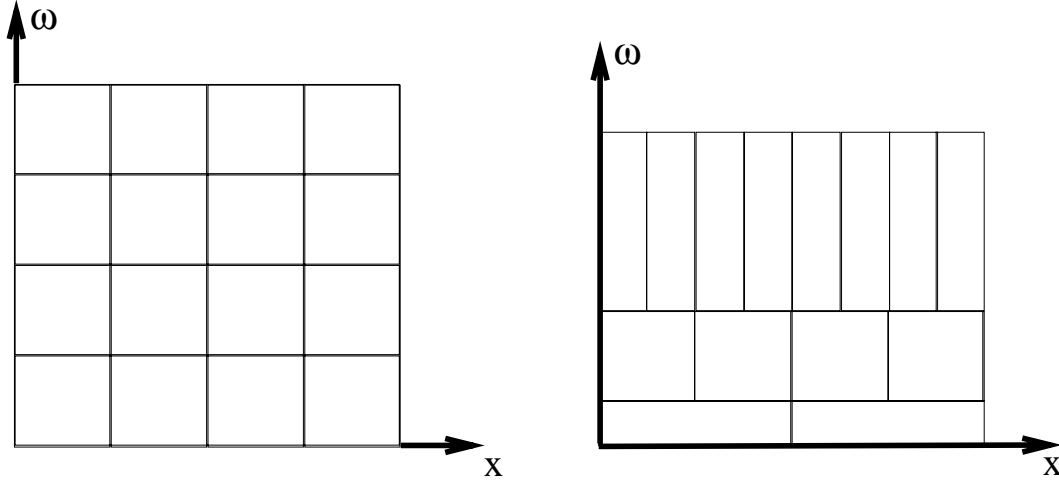


Figure 3.4: *Uniform (to the left) and logarithmic (to the right) splitting of the frequency domain.*

Continuous Wavelet Transforms

In the continuous case the wavelet transform is constructed from a single function $h(x)$ named the “mother wavelet”.

$$h_{a,\tau}(x) = \frac{1}{\sqrt{a}} h\left(\frac{x - \tau}{a}\right) \quad (3.10)$$

where

a is a scale, or dilation, factor,

τ is the translation,

$1/\sqrt{a}$ is a constant that is used for energy normalization.

The definition of the continuous wavelet transform (CWT) is then

$$CWT_h(\tau, a) = \frac{1}{\sqrt{a}} \int i(x) h^*\left(\frac{x - \tau}{a}\right) dx \quad (3.11)$$

Note that both the wavelet transform and the short time Fourier transform can be seen as Wigner-Ville distributions, [87].

The wavelet transform can be seen as projecting the signal onto a set of basis functions, $h_a(x)$. From a mathematical point of view it would be preferable to have an orthogonal basis to minimize redundancy and interference between different wavelets. This assures that it is possible to reconstruct the signal from its transform by summing up inner products.

$$i(x) = \int_{a>0} CWT(\tau, a) h_{a,\tau}(x) \frac{da d\tau}{a^2} \quad (3.12)$$

Surprisingly enough this holds even though the wavelets $h_{a,\tau}(x)$ are not orthogonal, [25]. The requirements on the wavelets are

- $h(x)$ should be of finite energy, i.e $\int h(x)h^*(x)dx < \infty$
- $h(x)$ should be of bandpass type, meaning that the reconstruction scheme in Eq. 3.12 is only for the signal energy $i(x)$, the DC-component cannot be reconstructed.

Discrete Wavelet Transforms

A natural question to ask is if it is possible to find an orthogonal wavelet basis by a careful sampling of a and τ . The answer is that it is possible but it depends critically on the choice of $h(x)$. There is a trade-off between the orthogonality and the restrictions on the wavelet. If the redundancy is high (oversampling), there are mild restrictions on $h(x)$. But if the redundancy is small (close to critical sampling), then the choice of wavelet is very constrained, [25].

The coefficients, c_{mn} , in the discrete wavelet transform (DWT) are given by

$$c_{mn} = \frac{1}{\sqrt{a_n}} \int i(x)h^*\left(\frac{x - b_{mn}}{a_n}\right)dx = \langle h_{mn}, i \rangle \quad (3.13)$$

where h_{mn} and the sampling points in the discrete wavelet transform are

$$\begin{aligned} h_{mn} &= h\left(\frac{x - b_{mn}}{a_n}\right) \\ a_n &= e^{\alpha n} \\ b_{mn} &= \beta m a_n \end{aligned} \quad (3.14)$$

where

a_n is the dilation sampling, or the sampling by the wavelet in the frequency domain.

b_{mn} is the spatial sampling.

The sampling density is now decided by the parameters α and β .

To analyze the behavior of the DWT, the notation **frame** from Daubechies [25] is an appropriate concept. The frame-bounds, A and B , of the wavelet transform are defined according to

$$A\|f\|^2 \leq \sum_{mn} \|\langle h_{mn}, f \rangle\|^2 = \sum_{mn} \|c_{mn}\|^2 \leq B\|f\|^2 \quad (3.15)$$

independently of the signal $f \in L^2$. Then, h_{mn} is a frame if $A > 0$ and $B < \infty$.

These frame-bounds balance the demand on the wavelet. If all the h_{mn} are normalized the following terminology is used.

$A = B = 1$ Orthonormal base.

$A = B > 1$ Tight frame, the number A is a measure of the redundancy.

$A \approx B$ Snug frame, the inverse is well-behaved.

The signal to noise ratio, SNR, of a synthesis of the original signal from its wavelet coefficients can be calculated from the frame-bounds, [25].

3.2.3 The Lognormal Filter

A function which is well concentrated in both domains, suggested in [67], is the so-called lognormal function, which in the Fourier domain is given by

$$F(\omega) = e^{-4 \ln^2(\omega/\omega_i)/B_{\log n}^2} \ln(2) \quad (3.16)$$

where

ω_i = center frequency,

$B_{\log n}$ = 6 dB sensitivity bandwidth in octaves.

This function is designed to meet several requirements:

- The DC-component should be zero.
- It should be possible to splice this function with zero sensitivity for negative frequencies.
Note: not only $F(0) = 0$ but also all its derivatives.
- The relative bandwidth, see Eq. 3.9, should be independent of the center frequency, ω_i .
- The function is self-similar which enables it to be used as a “mother wavelet”.

The drawback of the filter function is that the inverse Fourier transform is not known in a closed form.

Theoretical Frame-Bounds for the Lognormal Function

In Daubechies [26, 25], an extensive calculation scheme for different kinds of wavelet transforms is given. Following this scheme for the lognormal function, the frame-bounds A and B from Eq. 3.15 can be calculated.

The calculation of the frame-bounds involves an analysis in the Fourier domain of the suggested “mother wavelet” in the context of the sampling density chosen. In the case of the lognormal bandpass function this analysis is easiest and most comprehensible to evaluate when performed in the frequency domain.

The first demand of a wavelet is that it should be of finite energy, or in the terms of wavelets that the admissibility condition should be fulfilled, i.e.

$$c_h = 2\pi \int \frac{|H(\omega)|^2}{|\omega|} d\omega < \infty \quad (3.17)$$

For the lognormal frequency function $F(\omega)$, this is easily calculated giving the result

$$c_F = B_{\log n} \pi \sqrt{\frac{\pi \ln(2)}{2}} \quad (3.18)$$

In [26, 25], compact support of the Fourier transform of the “mother wavelet” in $[l, L]$ is assumed. Unfortunately, the lognormal function is not of compact support. In practical situations, however, the signals are discrete, i.e. they are sampled, with the Nyquist band limitation at π . This means that the Fourier domain is periodic and that the interval $[-\pi, 0[$ is the same as the interval $[\pi, 2\pi[$. Truncating the function at $\omega = \pi$, and choosing the interval $[l, L]$ as $[\varepsilon, 2\pi]$, where $\varepsilon \approx 0 > 0$, means that negative frequencies will not contribute with any energy. Hence the compact support interval will be $[\varepsilon, 2\pi]$. To give an impression of the approximation caused by the truncation of the filter function, say that the parameters $B_{\log n} = 1.5$ and $\omega_i = \pi/(2\sqrt{2})$ in Eq. 3.16 then the frequency function has decayed to $1/16$ of its top value at $\omega = \pi$. Thus the truncation will hardly introduce severe errors.

The calculation of the frame-bounds can now proceed by recalling Eq. 3.15, and rewrite it in terms of the sampling points, a_n and b_{mn}

$$A|f|^2 \leq \sum_m \sum_n |\langle a_n, b_{mn}; h|f \rangle|^2 = \sum_m \sum_n |\langle h_{mn}|f \rangle|^2 \leq B|f|^2 \quad (3.19)$$

where

$$\sum_m |\langle a_n, b_{mn}; h|f \rangle|^2 = \sum_m \left| \int_l^L e^{j\omega\beta m} H(n\omega) F(\omega) d\omega \right|^2 \quad (3.20)$$

By imposing

$$\beta = \frac{2\pi}{L-l} \approx 1$$

the spatial sampling density is locked to the scale sampling density, i.e.

$$b_{mn} = \frac{2\pi}{L-l} m a_n \approx m a_n$$

It is now possible to simplify Eq. 3.20 using the Parseval relation

$$\begin{aligned} \sum_m |\langle a_n, b_{mn}; h|f \rangle|^2 &= \sum_m \left| \int_l^L e^{j\omega 2\pi m/(L-l)} H(a_n \omega) F(\omega) d\omega \right|^2 \\ &= \int |H(a_n \omega)|^2 |F(\omega)|^2 d\omega \end{aligned} \quad (3.21)$$

It may be worth mentioning that in the above equations the Fourier domain is supposed to be continuous and the spatial domain discrete. The Parseval relation thus relates the $\sum_m |c_m|^2$ in the first line in Eq. 3.21 and the integral $\int |FH|^2 d\omega$ on the second line.

Remembering that only positive frequencies are involved in the calculations it is possible to define

$$\begin{aligned} FF(s) &= F(e^s), \\ HH(s) &= H(e^s). \end{aligned}$$

Substituting $\omega = e^s$ in Eq. 3.21, also remembering that $a_n = e^{n\alpha}$

$$\sum_n \sum_m |\langle a_n, b_{mn}; h|f \rangle|^2 = \int e^s |FF(s)|^2 \left(\sum_n |HH(s + n\alpha)|^2 \right) ds \quad (3.22)$$

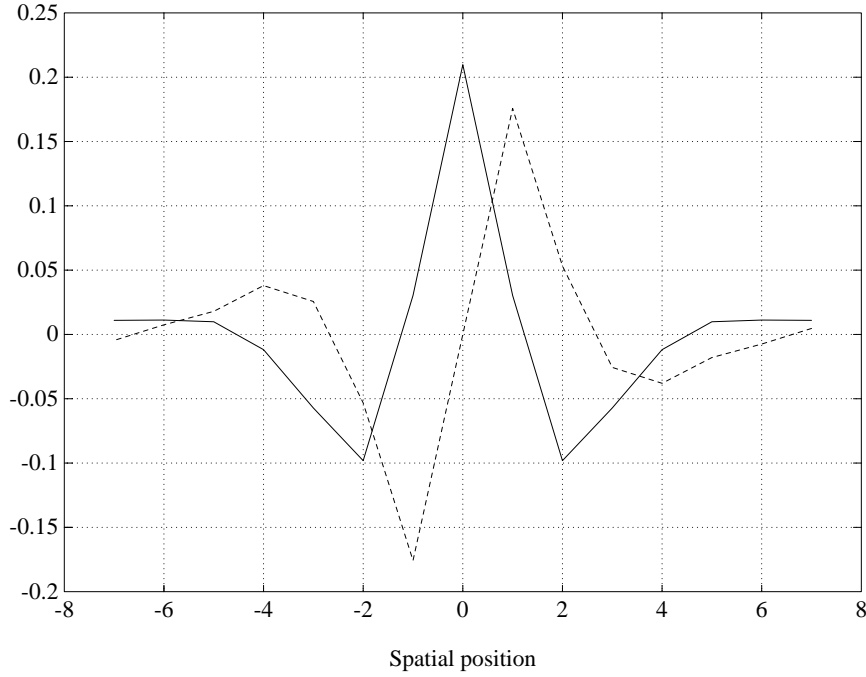


Figure 3.5: *The spatial lognormal filter realized as a 15 tap filter. The real part (solid) and the imaginary part (dashed) are plotted.*

This shows that

$$A = \inf_{s \in \mathbf{R}} \left[\sum_n |HH(s + n\alpha)|^2 \right]$$

$$B = \sup_{s \in \mathbf{R}} \left[\sum_n |HH(s + n\alpha)|^2 \right]$$

since

$$A \int e^s |FF(s)|^2 ds \leq \int e^s |FF(s)|^2 \left(\sum_n |HH(s + n\alpha)|^2 \right) ds \leq B \int e^s |FF(s)|^2 ds$$

From this short review of the calculations from [26, 25], it is now possible to proceed by investigating the frame-bounds for the lognormal frequency function.

Practical/Numerical Calculation of Frame-Bounds

As the analysis has been performed in a continuous Fourier domain, one must remember that the filters will be applied as spatial convolution kernels. These kernels must in practical situations be implemented with a limited number of coefficients. This is one reason for choosing functions which are smooth in both domains.

The lognormal filter has been realized as a spatial convolution with 15 coefficients, which is plotted in both domains in Figures 3.5 and 3.6. The parameters for the lognormal filter is set to $B_{\text{logn}} = 1.5$ and $\omega_i = \pi/(2\sqrt{2})$ in the Figures.

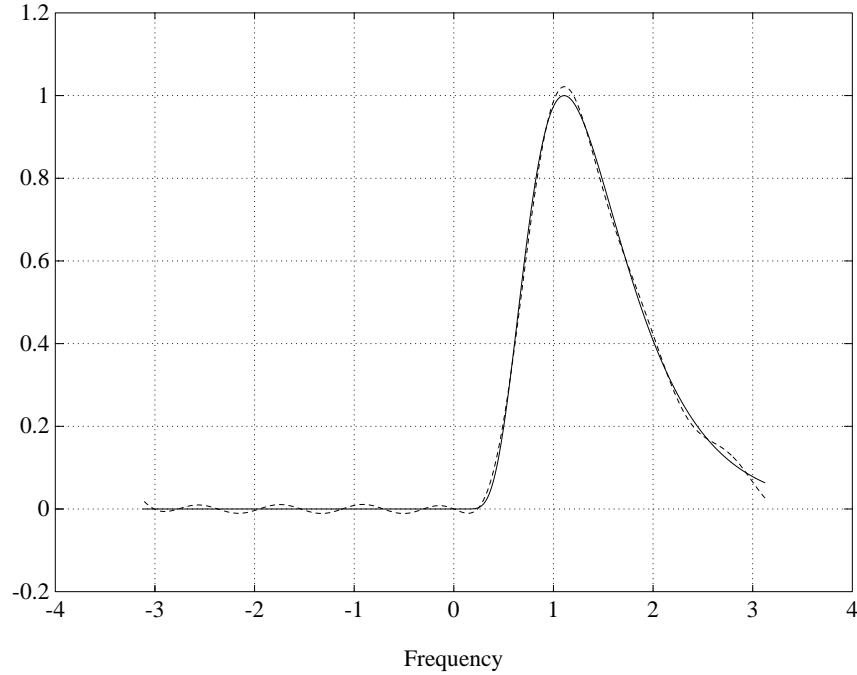


Figure 3.6: *The lognormal filter, both the ideal function (solid) and a transform of the realized convolution kernel from Figure 3.5 (dashed). The filters are plotted against a frequency axis and one period of the Fourier domain is shown.*

The frame-bounds for the lognormal functions have been calculated numerically following Eq. 3.22. Eq. 3.22 has been applied directly with the Fourier transform of the 15 tap filter as the “mother wavelet”. The frame-bounds with 8 discrete octave based scales used, i.e. 2^l where $l \in [-2, 5]$, are given as the maximum and minimum of the plot in Figure 3.7. The frame-bounds are for this specific case:

$$B = 1.33 \quad \text{and} \quad A = 1.12 \quad (3.23)$$

where the function is normalized with the energy.

This scheme has also been tested with another widely used filter function: the Gabor function. In the wavelet community this is called the Morlet wavelet, i.e. a Gabor function, Eq. 3.8, sampled with constant relative bandwidth. In image processing this sampling scheme was proposed by Granlund [39] and has been widely used, e.g. [84]. When implementing the Gabor functions as a quadrature filter, fulfilling the demands in section 3.1, the bandwidth, or standard deviation, of the filter must be rather small. There is a trade-off between the bandwidth of the filter and the demand that the filter should be zero, or at least small enough, at $\omega = 0$ and at $\omega = \pi$.

The 6 dB bandwidth B_g of a shifted Gaussian with standard deviation σ_ω and center frequency ω_0 is approximated with

$$B_g = \log_2 \frac{\omega_0 + \sigma_\omega}{\omega_0 - \sigma_\omega} \quad (3.24)$$

In practical situations B_g will be in the interval 0.8–1.2.

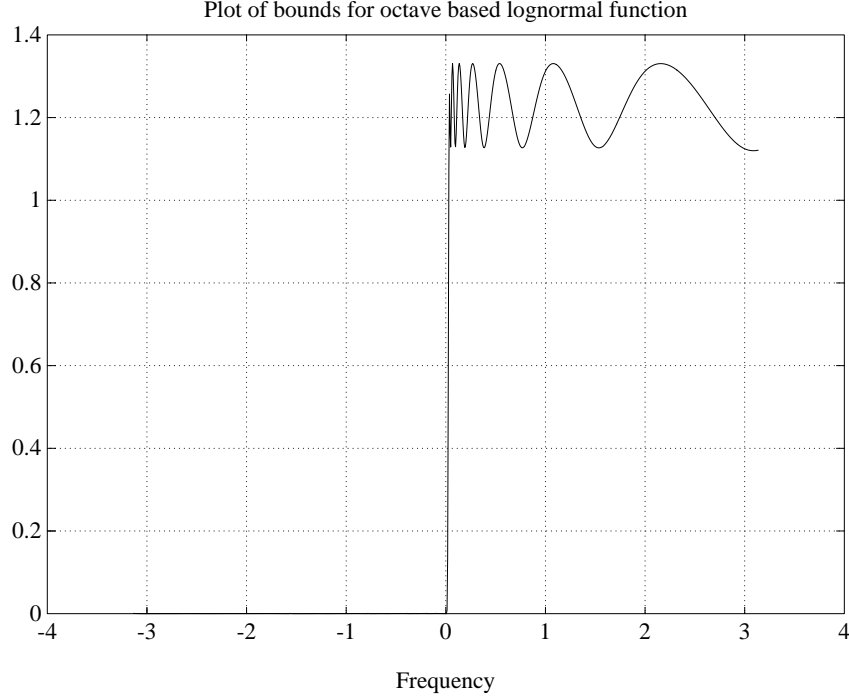


Figure 3.7: *Illustration of how the frequency sensitivity of the lognormal filter, used in an octave based DWT varies. The frame-bounds for lognormal filter in Figure 3.5 are the maximum and minimum of this plot.*

The experiment has been carried out with a Gabor filter with center frequency $\omega_0 = \pi/2\sqrt{2}$ and the relative bandwidth $B_g = 1.2$, realized as a spatial 15 tap filter. The used function is plotted in Figure 3.8 and its Fourier transform is given in Figure 3.9 together with the ideal function.

Applying Eq. 3.22, see Figure 3.10, for an octave based wavelet transform and, as before, 8 discrete scale levels give the frame-bounds for this Gabor function

$$B = 1.64 \quad \text{and} \quad A = 1.16 \quad (3.25)$$

The Scaling Function

So far nothing has been said about the scaling function, sometimes named the “wavelets’ father”, [31], that must be incorporated in the subsampling procedure. This function should, preferably, be orthogonal to the wavelets still remaining to apply, e.g. an ideal lowpass filter. A more realistic scaling function is a Gaussian, see Appendix 4.A, which has been applied and tested in the same way as above giving the following results. These results are relevant only if the wavelet is applied on a pyramid.

For the lognormal 15 tap filter in Figure 3.5 the result is that the frame-bounds over the total interval are

$$B = 1.30 \quad \text{and} \quad A = 0.68 \quad (3.26)$$

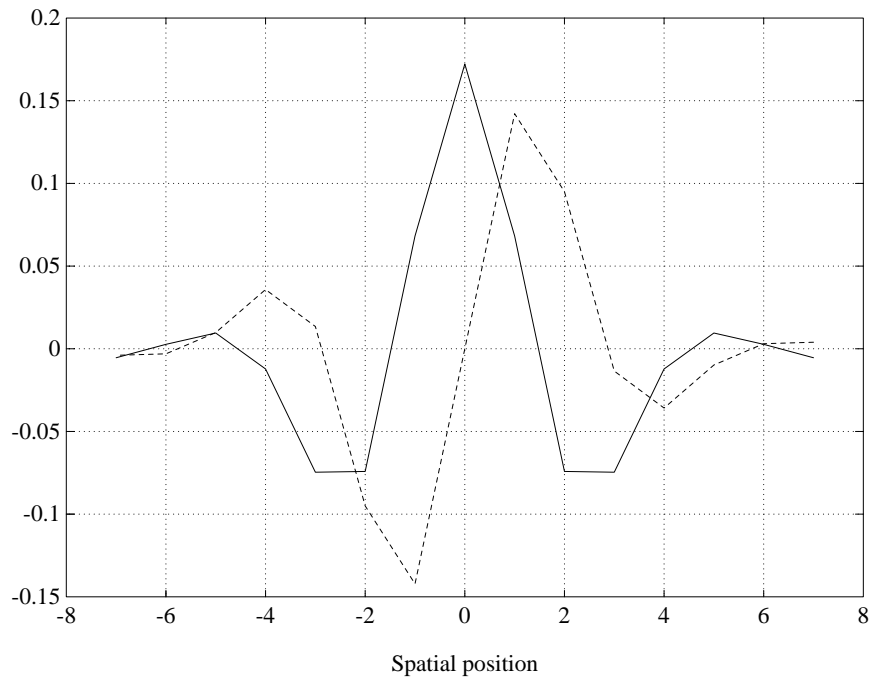


Figure 3.8: *The spatial Gabor filter realized as a 15 tap filter. The real part (solid) and the imaginary part (dashed) are plotted.*

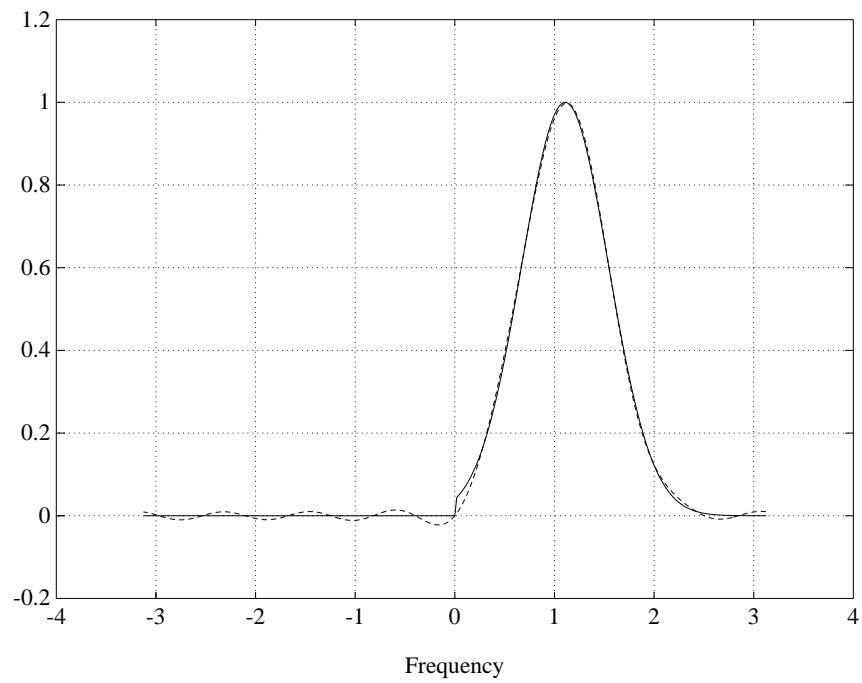


Figure 3.9: *The Gabor filter, both the ideal function (solid) and a transform of the realized convolution kernel from Figure 3.8 (dashed).*

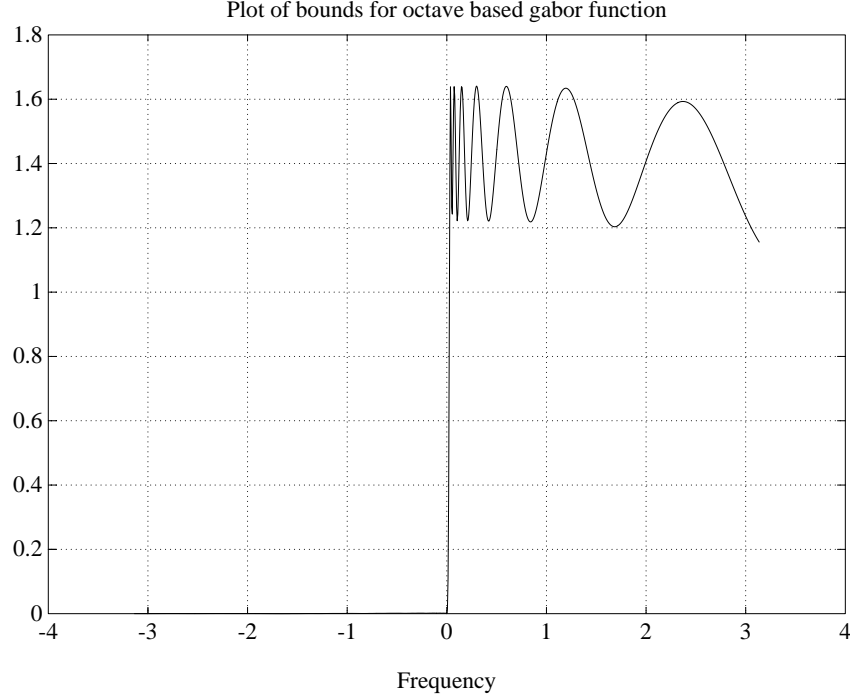


Figure 3.10: *Illustration of how the frequency sensitivity of the Gabor filter used in an octave based DWT. The frame-bounds for Gabor filter in Figure 3.8 are the maximum and minimum of this plot.*

but limiting the interval to be maximum $\pi/\sqrt{2}$ gives the frame-bounds

$$B = 1.30 \quad \text{and} \quad A = 1.02 \quad (3.27)$$

The result from Eq. 3.22 is given in Figure 3.11.

3.2.4 Conclusion

The calculations in this section has shown that an octave based bandpass pyramid based on lognormal filter functions is appropriate to apply. The maximum variation in sensitivity to different scales is approximately 20 %. This can be compared to an octave based bandpass pyramid based on Gabor functions, where the variation is approximately 40 %.

3.A Filter Optimization

This appendix is more or less a review of the methods reported about filter optimization in [67].

The reason for using an optimization procedure in filter design is that the filter must be realized with a limited number of coefficients. The optimization will then decide how to choose these coefficients so that a prescribed ideal function is approximated as well as possible, in some norm. Usually the ideal function is given in the Fourier domain and the

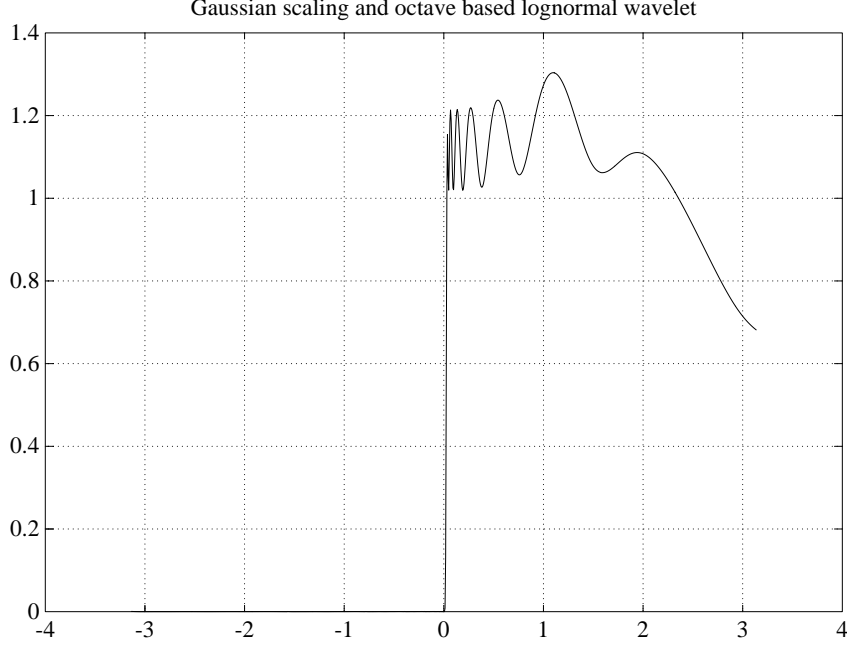


Figure 3.11: *Illustration of the frequency sensitivity for the lognormal filter, Figure 3.5, with a Gaussian scaling function. Maximum and minimum of this function are the frame-bounds for the total DWT.*

filter should be realized as a spatial convolution kernel. The Discrete Fourier Transform, DFT, could then be used to get the convolution kernel. The larger the spatial kernel, the better the approximation of the ideal function will be, but this is, of course, at the cost of higher complexity in the computation, and give also poorer spatial resolution.

Using a DFT imposes that it is equally important to follow the ideal function, F_{ide} , at all frequencies, i.e. the DFT performs the following minimization if the spatial grid is prescribed:

$$\min \sum_{\omega \in [-\pi.. \pi[} |F_{ide}(\omega) - \text{DFT}[f_{spat}(x)](\omega)|^2$$

meaning that f_{spat} is given by an inverse DFT, IDFT, of the ideal function. While f_{spat} is spatially limited the Fourier domain must be periodic, which is the reason for the interval $[-\pi.. \pi[$ in the optimization.

In many practical situations the assumption of equal importance for all frequencies will **not** hold, e.g. the DC-component has an extremely high importance. Thus a more realistic optimization would be

$$\min \sum_{\omega \in [-\pi.. \pi[} w(\omega) |F_{ide}(\omega) - \text{DFT}[f_{spat}(x)](\omega)|^2 \quad (3.28)$$

where $w(\omega)$ is a weighting function. In this minimization it is possible to put in a priori knowledge of both the signal spectrum and the noise spectrum.

Statistically, according to [67], images have a spectrum decaying somewhere between $1/\omega$ to $1/\omega^3$. This and the importance of the DC-component have been used when the weighting function, $w(\omega)$, has been designed.

It is also worth mentioning that a filter realized with a fixed number of spatial coefficients can be transformed to the Fourier domain and represented with a large number of coefficients. The trick is to use zero padding in the spatial domain to get a better resolution in the Fourier domain. This means that the ideal function, F_{ide} , can be discretized with a sufficient number of coefficients, even though the convolution kernel, f_{spat} , is supposed to be realized on a rather sparse grid.

Chapter 4

On the Use of Phase in Scale Analysis

4.1 Phase Estimation in Images

An extension of the phase concept from the well-known theory of one-dimensional signals to multidimensional signals is not trivial. The main reason is that the ordering problem does not have a unique solution for a dimensionality of two and higher. This implies that

- a) the neighbourhood should be one-dimensional, i.e. vary in only one direction,
- b) the direction in which the phase should be estimated must be predefined and
- c) the resulting phase representation must be invariant to rotation of the input signal.

The requirements **a)** and **b)** can be met by finding the one-dimensional structures and extracting a phase independent orientation, [67], giving a rather straightforward generalization. The estimation of orientation is further described in Section 5.1, here it is just assumed to be accessible in a “double angle” representation, [39], see Figure 4.1. The orientation measure is used to control in what direction the phase is estimated and the phase is given by the responses from the quadrature filters used. Further, it will be shown that a three-dimensional representation, involving phase and orientation, is necessary to meet the requirement **c)**.

Following this discussion by assuming a one-dimensional structure and that a filter, \mathbf{q}_k , is exactly aligned with the orientation of the one-dimensionality. It is then possible to define the phase of this filter, $\theta_{ph,k}$, as the quotient between the odd and the even filter results. The filter, \mathbf{q}_k , could for example be a lognormal quadrature filter, Eq. 3.16.

$$\begin{aligned}\mathbf{q}_k &= q_{ke} + j q_{ko} \\ \theta_{ph,k} &= \arg(\mathbf{q}_k)\end{aligned}$$

In order to give a better understanding of the phase concepts some particular instances of phase angles are presented below and in Figure 4.2.

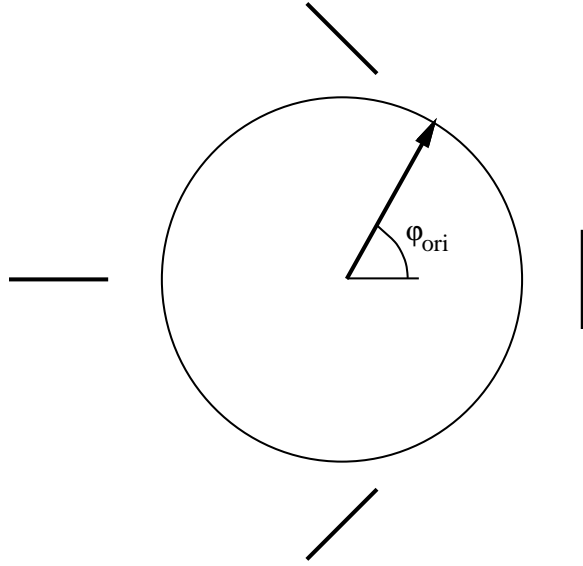


Figure 4.1: *The used orientation representation where a doubling of the argument produces a continuous mapping of the local orientation, see also Section 5.1.*

- $\theta_{ph} = 0$: corresponds to a symmetric input, i.e. a line, which is bright in comparison with its surroundings.
- $\theta_{ph} = \pi$: corresponds to a symmetric input, i.e. a line, which is dark compared to its surroundings.
- $\theta_{ph} = \pm\frac{\pi}{2}$: corresponds to an antisymmetric input, i.e. an edge. The sign cannot be defined without the orientation of the edge.

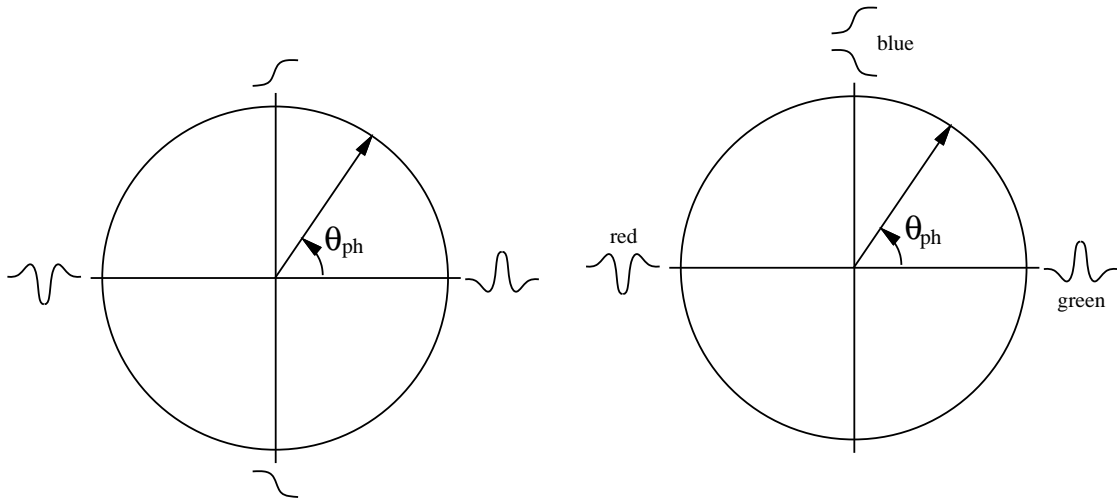


Figure 4.2: *Left: The phase as a line/edge detector. Right: Illustration of a rectified phase, where the sign of the edge is skipped, the colors corresponds to the colors in the result images.*

This is, of course, just the phase from one filter at a particular orientation. To get a general phase estimate that is equally sensitive to every orientation, all filters

must be involved in the overall estimate. One way of doing this is to sum the even filters separately from the odd ones and take the quotient between the sums.

$$\theta_{ph} = \arg(\sum_k q_{ke}, \sum_k q_{ko})$$

This is possible if the summation is carefully done, since the signs of the results from the odd filters are dependent on the main direction of the input. There is a fundamental discontinuity in the phase estimate, or more precisely in the sign of edges, on images. Consider a dark circle on a brighter background. The sign of the edge can for example be defined as positive, if white is to the right (**x** in Figure 4.3). Start to walk around the circle, and the sign of the edge changes from positive to negative when you have reached the opposite side (**o** in Figure 4.3). Somewhere between these two points the edge abruptly has changed its sign. This discontinuity is orientation dependent, so if the orientation is available simultaneously, the representation could be made continuous. The crucial factor in the summation of the filter responses is to ensure that all filters change signs at the same orientation.

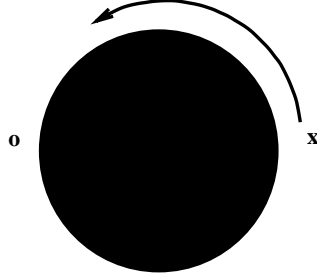


Figure 4.3: *Circle illustrating the edge discontinuity.*

Another way to define the phase is to use the orientation estimate to interpolate a quadrature filter in that direction. This will be the phase taken across a one-dimensional structure inside the neighbourhood. The interpolation can be analytically solved for the even part, but for the odd part of the filter an optimization criterion must be decided and solved numerically. Recalling Eq. 3.4 the even part of the filter has an angular variation $\cos^{2A}(\varphi - \varphi_k)$, where φ_k is the main direction of the filter. As an example, put $A = 1$, $\varphi_k = k\frac{\pi}{4}$ and use four filters corresponding to $k = 0, 1, 2, 3$. To interpolate in an arbitrary direction, φ_0 , the expected filter response looks like $\cos^2(\varphi - \varphi_0)$. This is readily obtained by simple trigonometric formulas.

$$\begin{aligned} \cos^2(\varphi - \varphi_0) &= \frac{1}{2}(1 + \cos(2(\varphi - \varphi_0))) = \\ &= \frac{1}{2}(1 + \cos 2\varphi \cos 2\varphi_0 - \sin 2\varphi \sin 2\varphi_0) = \\ &= \frac{1}{2}[1 + (\cos^2\varphi - \sin^2\varphi) \cos 2\varphi_0 - (\cos^2(\varphi - \frac{\pi}{4}) - \sin^2(\varphi - \frac{\pi}{4})) \sin 2\varphi_0] \end{aligned} \tag{4.1}$$

Since

$$\sin^2\varphi = \cos^2(\varphi - \frac{\pi}{2}) \quad \text{and} \quad 0.5 \sum_k \cos^2(\varphi - \varphi_k) = 1$$

all terms in Eq. 4.1 are among the four specified and fixed filters.

The odd part has an expected angular function $\cos^2(\varphi - \varphi_0)\text{sign}[\cos(\varphi - \varphi_0)]$, Eq. 3.4, which cannot be interpolated from the four fixed odd filters with the same main directions as the even filters. Thus a least square error interpolation table has been calculated, another way of doing this interpolation is presented in [6]. The normalized errors from this numerical solution are no worse than 0.08, and the maximum error is where the function value is smallest, for orientations right between the fixed orientations. No large effect from these errors has been observed in the calculation and processing of phase images.

The phase estimation after the interpolation is taken as the argument of the complex number, $q_{e\varphi_0} + jq_{o\varphi_0}$, where $q_{e\varphi_0}$ is the output from the interpolated even filter and $q_{o\varphi_0}$ is the output from the interpolated odd filter.

The second way of estimating phase has been chosen in the implementation, but the difference in appearance compared to the first method is negligible. In the case of a perfect one-dimensional signal there will be no difference at all between the methods.

4.1.1 Representation

To make the phase description invariant to rotation of the input signal, a three-dimensional representation is necessary. The parameters involved in the representation is the energy, the phase angle and the orientation. The chosen representation fulfills the three constraints, **a)**, **b)** and **c)**, from above. The representation builds on a spherical coordinate system, see Figure 4.4, and was originally proposed by H. Knutsson. In this system corresponds

$$\begin{aligned}\theta &= \theta_{ph} && \text{The phase} \\ \varphi &= \varphi_{ori}/2 && \text{Half the orientation estimate} \\ r &= E && \text{The energy}\end{aligned}\tag{4.2}$$

where the energy measure can be chosen as the sum of the energies from each filter, i.e. $\sqrt{\sum_k (q_{ko}^2 + q_{ke}^2)}$, or as the energy from the interpolated filter, i.e. $\sqrt{q_{e\varphi_0}^2 + q_{o\varphi_0}^2}$.

The slight modification here from the common definition of a spherical coordinate system is that $\theta \in [0, 2\pi[$ and $\varphi \in [0, \pi]$. Although it is possible to rewrite it on the standard form, this notation is kept in order to get an appreciation of what is happening in the chosen space when changing the input.

The representation fulfills the proposed requirements for a good representation, since it is

- invariant to rotation of the input,
- possible to average, where the mean vector equals the mean of the signal without any drift.

The last point is obvious, but the first one needs some contemplation. The

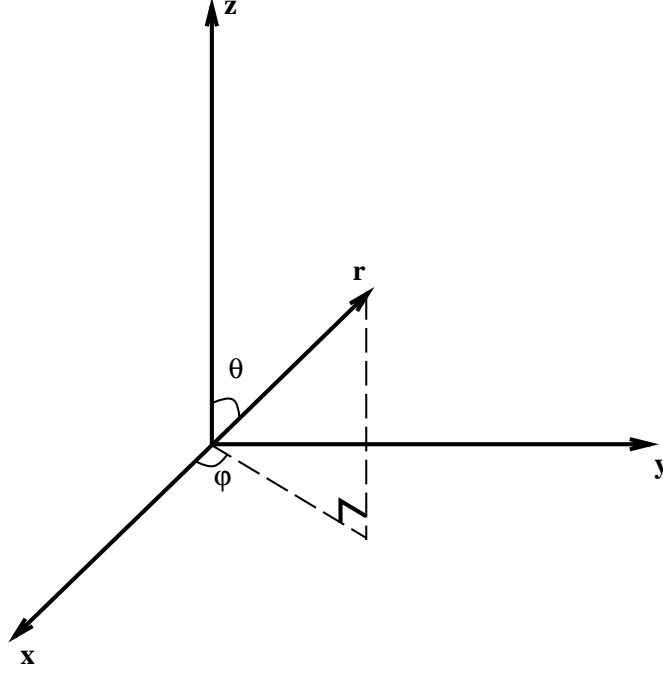


Figure 4.4: A spherical coordinate system.

invariance can be proven by introducing

$$\begin{cases} x &= r \sin\theta_{ph} \cos\varphi \\ y &= r \sin\theta_{ph} \sin\varphi \\ z &= r \cos\theta_{ph} \end{cases} \quad (4.3)$$

The easiest way to define phase angle and orientation is according to Figure 4.5. θ_{ph} changes signs abruptly in one specific orientation, φ_{ori} . This orientation can be chosen as $\varphi_{ori} = 0$ without loss of generality. By halving the “double angle” representation of orientation, the discontinuity will occur not between $\varphi_{ori} = 0$ and $\varphi_{ori} = 2\pi$, but between $\varphi = 0$ and $\varphi = \pi$. Let us then examine what happens in the representation space in these orientations.

What happens when $\varphi = 0$: Figure 4.5 defines $\theta_{ph} = +\frac{\pi}{2}$, then

$$\begin{cases} x &= r \sin(\frac{\pi}{2}) \cos(0) = r \\ y &= r \sin(\frac{\pi}{2}) \sin(0) = 0 \\ z &= r \cos(\frac{\pi}{2}) = 0 \end{cases} \quad (4.4)$$

Now let us look at $\varphi = \pi$ and $\theta_{ph} = -\frac{\pi}{2}$

$$\begin{cases} x &= r \sin(\frac{-\pi}{2}) \cos(\pi) = r \\ y &= r \sin(\frac{-\pi}{2}) \sin(\pi) = 0 \\ z &= r \cos(\frac{-\pi}{2}) = 0 \end{cases} \quad (4.5)$$

A popular way of describing the representation would be to say that the orientation changes its sign at the same place as the phase does, and that these changes compensate each other.

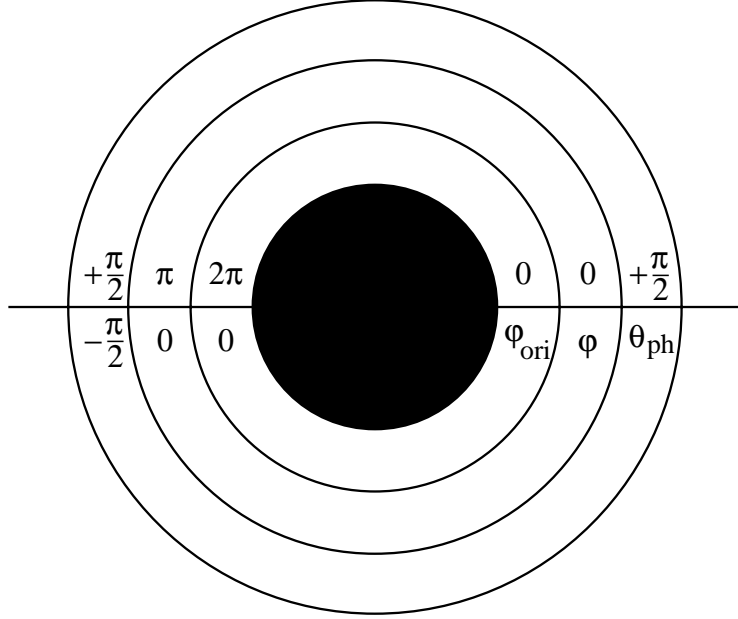


Figure 4.5: *Definition of phase, θ_{ph} , and orientation, φ_{ori} .*

By noting that the phase estimate is insensitive to the orientation when $\theta_{ph} = 0$ or $\theta_{ph} = \pi$, meaning centers of lines or blobs, the examination of the discontinuity points show that this space is invariant to rotation of an input edge or line.

4.1.2 Experimental Results

To complete the presentation of the three-dimensional phase description two examples of the representation on both a synthetic and a natural image will be presented in Figure 4.6 and Figure 4.7 respectively. The images are divided into four sections. The upper left part is the x-component, the upper right is the y-component, the lower left is the z-component and the lower right part is the original image. Before visualisation the components have been normalized to the interval between zero, meaning black, and one, meaning white. In these images it easy to see that the chosen three-dimensional phase representation is continuous in each of its Cartesian components.

The filter used for the radial part is the one described above. The angular function is

$$\cos^2(\varphi - \varphi_k) \quad \text{where} \\ \varphi_k = k \frac{\pi}{4} \quad \text{with} \quad k = 0, 1, 2, 3$$

4.2 Feature Scale-Space Utilizing Phase

This section describes a new algorithm which detects in what scale an event appears and also in what scale it disappears. In this way the scale-space is subdivided into a

Figure 4.6: *Example of the 3D-phase description of a natural image.*

number of intervals. Within each scale interval a consistency check is performed to get the certainty of the detection. High certainty means visually important features. The lowpass pyramid that is calculated as a first step in the algorithm is similar to the one Marr, [78], proposed. Hence the sampling in scale will be on octave basis. The pyramid will typically contain images from 512x512, 256x256, . . . , down to 16x16.

It is shown that by using the three-dimensional phase representation of image data from Section 4.1, it is possible to do both the splitting and the consistency check in a simple manner. The scale levels between different events are detected when a particular dot product becomes negative and the consistency will be a vector summation in the interval between these scales. The specific levels where a split occurs will, of course, be contextually dependent. There will also be different numbers of levels in different parts of the images.

To get scale invariance in the feature extraction, the chosen feature must be estimated in all, or at least many, scales. The sample density in resolution has been tested with both octave differences and half octave differences. Tests have shown that nothing is gained by doubling the sample density. This is not surprising, since the calculation of the frame-bounds in Section 3.2.3 indicates that almost all information is captured in an octave based pyramid using lognormal filters. The objective of the implementation is to estimate the phase of successively smaller images from the lowpass pyramid, suggested in Appendix 4.A. By using exactly the same set of quadrature filters at all size levels in the lowpass pyramid, an octave based bandpass pyramid is obtained. The filter function chosen is given in Eq. 3.16 with the center frequency $\omega_i = \pi/(2\sqrt{2})$ and the bandwidth $B = 1.5$, as in the frame calculations of Section 3.2.3.

Figure 4.7: *Example of the 3D-phase description of a synthesized image.*

4.2.1 Scale Space Clustering

The phase will always be dependent on the scale of the filters used to estimate it. Figure 4.8 shows the responses from filters with different frequency response applied to an edge. The phase response will be independent of the frequency responses of the filters if there is a scale invariant event, such as an edge or a line, and the filter is centered upon this event. This clearly indicates that scale analysis used for deciding signal phase and phase consistency over scale is an important tool in image processing. The proposed representation has been implemented in a scale pyramid and it has proved to be robust in rather noisy situations.

The description of scale space clustering is divided into two sections. The first section concerns isolated events, where no clustering is necessary. The second section is an extension of the first part in the sense that events not necessarily need to be isolated. In the second section the events are allowed to be objects within objects, what in the sequel will be referred to as nested events. In this case the scale space must be divided into different sections originating from different nesting levels.

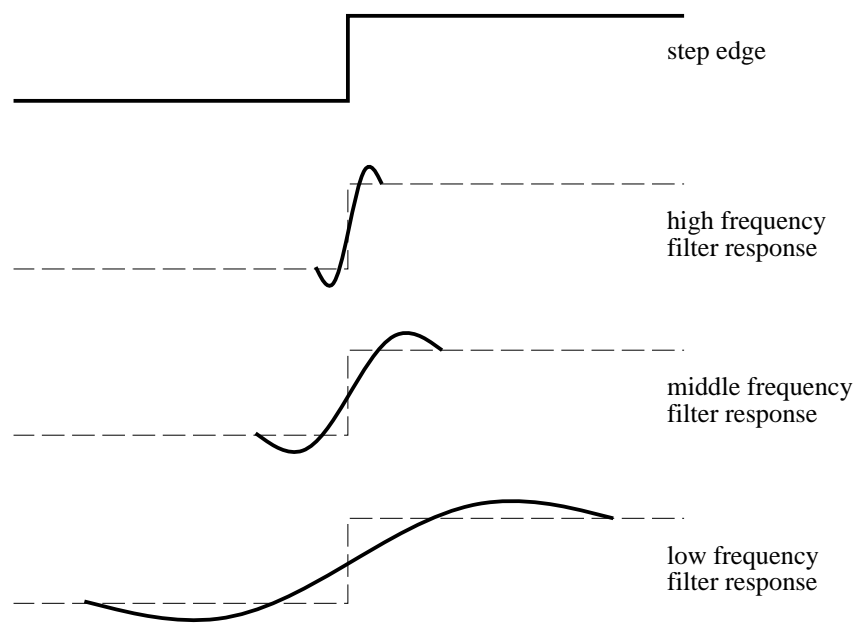


Figure 4.8: *Filter responses to a step edge from different filters.*

Scale Analysis for Isolated Events

One main idea in the proposed algorithm is to do scale analysis as pixelwise operations through all scales. To be able to do this between different scale levels in a simple manner the images must have the same size. The interpolation scheme proposed in Appendix 4.A can be applied, and afterwards the analysis will be restricted to pixelwise operations.

The scheme for interpolation to a higher sampling rate described for grey scale images can easily be extended to vector images, such as for example the result of the phase estimation. The only difference is that the averaging must be a vector averaging. Vector averaging can be performed as a convolution, in which the magnitude of the vectors are weighted with the proposed interpolation function and the summation is a vector summation. Applying a proper interpolation function at each level of the phase pyramid results in a high tower, in which images at all scale levels have the same size. An example of this is given in Figure 4.9, where three samples of scales are presented. The lowpass image and the corresponding phase image are displayed pairwise in Figure 4.9. The phase images are displayed in their rectified form, meaning that green corresponds to 0° (locally bright areas), blue to $\pm 90^\circ$ (unsigned edges) and red to 180° (locally dark areas), see also Figure 4.2.

Several combinations of the images from different scales have been tested. One combination is to find the scale level that has the highest energy. This is similar to the algorithm in [23], but using phase representation over the whole image. It turns out that this gives a discontinuous result image, since the chosen level can emanate from drastically different scales in adjacent pixels. The maximum criterion is thus no well-behaved combination over scale.

Another variation is to find the maximum pixelwise vector sum of phase images of adjacent scale levels according to Eq. 4.6.

The sum, \mathbf{C} , can be expressed in a Cartesian coordinate system as

$$\mathbf{C}(x, y, z) = \left(\sum_{l=i}^{l=j} x_l, \sum_{l=i}^{l=j} y_l, \sum_{l=i}^{l=j} z_l \right) \quad (4.6)$$

where x, y and z are defined in accordance with Eq. 4.3, and i and j are indices of what scale levels that contribute to the sum.

The vector sum can be regarded as a moving average of the samples in scale. Hence the maximum over the sums gives a smoother function than merely taking the maximum. The sum is not only taken for pairwise adjacent scale levels but triples and quadruples and so on are likewise regarded. The length of the resultants are normalized so that the resulting length is independent of the number of vectors forming the sum. The maximum of these vector resultants gives a good representation of the original image, at least as long as the event is isolated. It was then discovered that a vector sum involving all scale levels gives a slightly better representation and is much simpler to implement.

The vector summation over scale as a consistency measure is possible because of the chosen representation, where the length of the vector tells about energy, or consistency, in the specific frequency band, and the angles are the phase angle and

Figure 4.9: *Three samples of differently scaled images both as lowpass images and represented as phase.*

the orientation. If the operation makes the same “statement” in all scales, the output magnitude will be very large. If, on the other hand, the estimation is not the same for different scales the magnitude will go down, see Figure 4.10, which is an example of a two-dimensional representation.

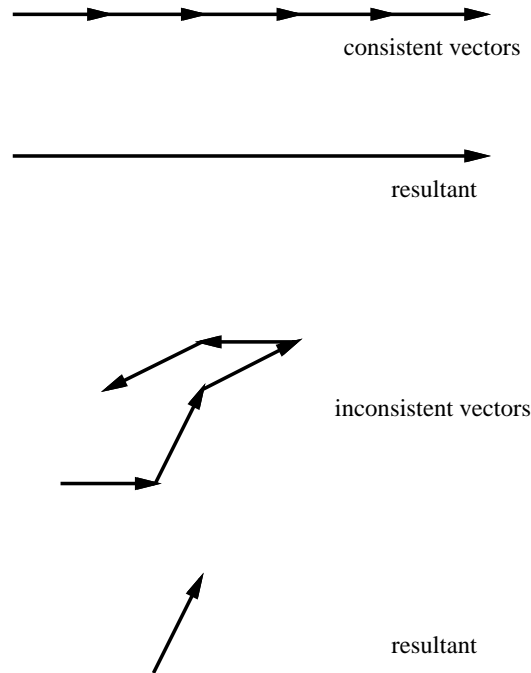


Figure 4.10: *Sums of consistent and of inconsistent vectors.*

To conclude, an example of the sum of phase vectors is given in Figure 4.11. The left image is the total phase summation image of the image in Figure 4.9 and the right one is a map of phase angles from color to grey scale. Note that almost all information from the original image is represented in the phase summation image.

The drawback of this solution is the restriction to isolated events. In a case where there are nested events, the result will be a combination of the phase vectors of all objects. The result will not correspond to either of the underlying events but to some kind of average of them. This average is, of course, not a good representation of a nested event. To take an everyday example, it is like averaging two roads that are situated to the left and to the right of a tree respectively. The average will be a non-existent road straight through the tree.

Scale Analysis for Nested Events

To be able to handle nested events in a consistent way, a subdivision of the scale pyramid, or actually of the scale high-tower, is necessary. This is based on how events behave in a blurring process. As stated earlier, each item manifests itself in a particular scale interval. If there is more than one object at a certain place, there is more than one maximum in the scale dimension. To distinguish between these events a subdivision is necessary.

To follow the reasoning consider the stylized image in Figure 4.12. The lines

Figure 4.11: *A phase summation image in its rectified form, left, and mapped to a grey scale image, right.*

are the grey level intensity at different scales and the vectors indicate the phase responses. Observe that the angles of the vectors normally change smoothly between adjacent scale levels. The exception is when the small minimum in the middle is “swallowed” by the larger maximum. Between level 2 and level 3 these particular vectors turn from 180° (\leftarrow) to 0° (\rightarrow). The implication of this is that a split should occur between these levels in the area of the small minimum. The small minimum should be analyzed only in the two scale levels with the best resolution to avoid interference with the maximum. The large maximum, on the other hand, should be analyzed in level 3 and in more blurred versions of the image.

The subdivision of the scale pyramid is a splitting process. The essence of this process is a dot product between phase vectors from two adjacent scale levels. If the dot product is positive, then the event is said to exist in both scales, and no split should occur. If it is negative, a split between these levels should take place. The negative sign of the dot product implies that the phase vectors of the regarded scale levels originate from different events. The proposed algorithm results in a nested description of the image. The number of levels in the description is contextually dependent and equals the number of nesting levels in the image. After the subdivision of the pyramid each interval only contains one isolated event at any given location of the image. This implies that the vector summation can be used as a consistency measure.

Some examples of the performance of the algorithm, are shown Figure 4.14 and Figure 4.15. The images are split into four sections, where the upper left part is the scale interval with the finest resolution. The upper right part is the next interval

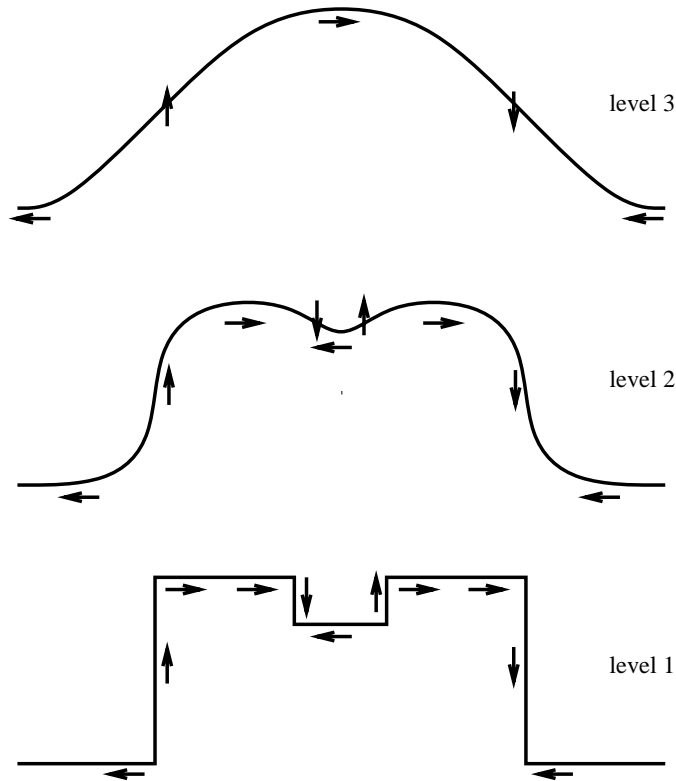


Figure 4.12: *A stylized example of a blurring process for nested events.*

with larger scale, the dark parts mean that there is only one scale interval in these areas. The lower left shows if there are any third intervals in the images. Finally, the lower right part indicates where in the scale the splitting levels occur. The intensity is where the first split level is, low intensity means that the first split occurs at a rather high frequency level. The hue reflects if there is any second split and on what level this takes place. The colors in the phase images correspond to phase angles without signs going from green to blue to red meaning, 0 , $\pi/2$ and π angles. The intensities in these images are taken directly from the vector sum their respective scale interval.

The subdivision of the pyramid is not only valid for phase estimations but can be used as contextual control for other features. Orientation and spatial frequency are two examples of such features. Hence it is possible to give a description of an image event containing many aspects that are estimated in the correct scale interval.

4.2.2 Spatial Frequency Estimation

An algorithm for spatial frequency estimation will be described and results on synthesized images as well as on natural images will be presented. The algorithm presented here uses the separate results from each level of the bandpass pyramid to estimate spatial frequency. Spatial frequency is known to be a good discriminator between different kinds of textures, [61, 16]. This property is further enhanced when it is combined with a phase-gating process. The purpose of the gating is to ensure that only neighborhoods in which spatial frequency is a good descriptor are utilized

Figure 4.13: *The original test image.*

in the classification.

A Spatial Frequency Algorithm and Representation

The chosen algorithm is intimately connected to the chosen representation. The belief is that spatial frequency can be represented in a modular, or circular, manner. The inspiration to this is the fact that one, and only one, color is perceived when humans observe light with an arbitrarily complex (time) frequency distribution. The color is mainly determined by the frequency component with the largest energy. If all different frequencies contained in the perceived light are represented as vectors pointing in a color circle, see Figure 4.16, and the lengths of the vectors reflect the energy of the represented frequency, then the perceived color would be determined by a vector sum of these vectors.

The key idea in the spatial frequency estimation is to decide the major frequency in a given neighbourhood and, if many frequencies are detected simultaneously, to lower the certainty. In the modular representation this is implicitly given. It will be possible to do spatial averaging directly upon the result image. A conventional linear representation of spatial frequency would not be easy to do averaging on, since a drift towards the middle of the represented interval should occur. The parallel to the decrease in confidence with increasing bandwidth in color spaces, is the change of color to white, in which all (time) frequencies are “contained” and the saturation equals zero.

To obtain the circular representation, a vector combination of the energies from the levels of the phase pyramid, which is a bandpass pyramid, with fixed axes is executed in accordance with Figure 4.17, where E_6 corresponds to the energy from

Figure 4.14: *A phase description of a synthesized image.*

the lowest resolution, and E_1 corresponds to the energy from the highest resolution. This example is from a pyramid with six levels, but the strategy can be used with any number of levels. The only condition is that the axes of summation are evenly spread over the unit circle. In Figure 4.17 there is also an intuitive interpretation of what different angles mean in the circular spatial frequency representation. The translation of these angles to corresponding colors is also indicated to make understanding of the result images easier.

The section between E_1 and E_6 in Figure 4.17 is due to the modular representation and is a parallel to the so-called purple line in color perception. In the case of spatial frequency the statements in this section mean mixed frequencies, both very high and very low ones, in the same area.

The representation will be a complex number in accordance with the hierarchical scheme proposed in [39]. In a general case with a pyramid with N levels, where level 1 is the one with the lowest resolution, the mapping to spatial frequency, $\mathbf{Fre}(m, \phi)$, looks like

$$\mathbf{Fre}(m, \phi) = \sum_{k=1}^N E_k \exp(j\pi(-1 + \frac{2k-1}{N})) \quad (4.7)$$

where m is the confidence of the estimate and ϕ is the estimate.

To analyze the estimate, a plot has been calculated for a single frequency as input. The plot can be found in Figure 4.18 with logarithmic frequency on the horizontal axis. The upper part is a plot of the confidence, m , in \mathbf{Fre} and the lower part is the estimate, ϕ .

Observe how linear the response is in an interval over 4 octaves. The possibil-

Figure 4.15: *A phase description of a natural image.*

ity to extend the interval by normalizing the estimate with the response curve in Figure 4.18 is apparent and it seems reasonable to get a linearized estimate over 6 octaves. The plot of the confidence is almost constant over the linear interval. The maximum of the confidence for high frequencies comes from the fact that in Figure 4.18 the band limitation of the original image is assumed to be caused by an ideal lowpass filter. The filter with the highest peak frequency operates directly upon the original image, while the other filter operates on images that have been filtered with a Gaussian in the subsampling.

If spatial frequency is supposed to be used as a size descriptor, then the estimate must be object-oriented. Thus a problem that has not been mentioned yet in connection with frequency estimation appears: the existence of nested events in the image. To get a proper size description of an object, the estimation must be done in the correct scale interval, and other scales must not interfere in the description. This problem was solved in the case of phase estimation by a subdivision of the scale pyramid. The solution can also be used in frequency estimation. The most natural way is to use the very same subdivision. This implies that the index in Eq. 4.7 should be restricted to the considered scale interval, and the restriction should be controlled by the subdivision of the phase pyramid. The description of the image will then be both a phase estimate and a frequency estimate, originating from the same scale interval.

The presentation will be concluded with some examples of results from the estimations, see Figure 4.19 and Figure 4.20. The images are, as in the case of phase estimation, divided into four sections. The partitioning is the same as in the phase images, Figure 4.14 and Figure 4.15. The difference is that the hue now means spatial frequency. The colors that reflect the frequency, in order from low to high

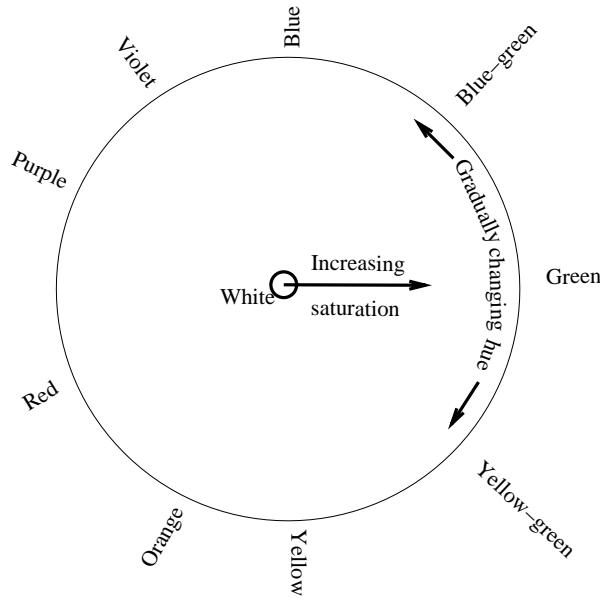


Figure 4.16: *An approximation of a psychological color space as a circle.*

frequencies, are orange, green and blue which also is indicated in Figure 4.17. The mixed frequencies are displayed in a reddish hue. The intensity is a combination of energy in the input and the bandwidth, high bandwidth lowers the intensity.

Phase-Gating

The phase-gating concept is a postulate that estimates from an arbitrary operator or algorithm are valid only in particular places, where the relevance of the estimate is high. To take an example in the context of frequency estimation, the size of an object should be decided independently of how sharp the edges are, thus only the even component of a quadrature filter should be involved in the frequency estimation. First a test was performed on a test image containing ten circles with different radii, Figure 4.21. The radii range from 4 pixels to 90 pixels, and the increase of the radii is half-octave based, i.e. the radius is increased by a factor of $\sqrt{2}$ from one circle to the next. The result from the phase gating is given in yellow in Figure 4.21, where the original image have been corrupted with white additive Gaussian noise¹. To illustrate the performance of the proposed spatial frequency algorithm the mean of the angles, ϕ from Eq. 4.7, inside each phase gated area is plotted in Figure 4.22 versus the radii of the circles.

The second experiment was to test the discrimination ability of the size descriptor for texture segmentation. The image to classify is the one in Figure 4.9 and the only feature used is spatial frequency. Figure 4.20 shows the spatial frequency estimate for the texture image to be classified. The scale interval with the highest consistency, i.e. in the interval with the highest vector sum resultant in the phase pyramid, is

¹The SDEV of the noise was 12.8 and the amplitude of the signal 153.

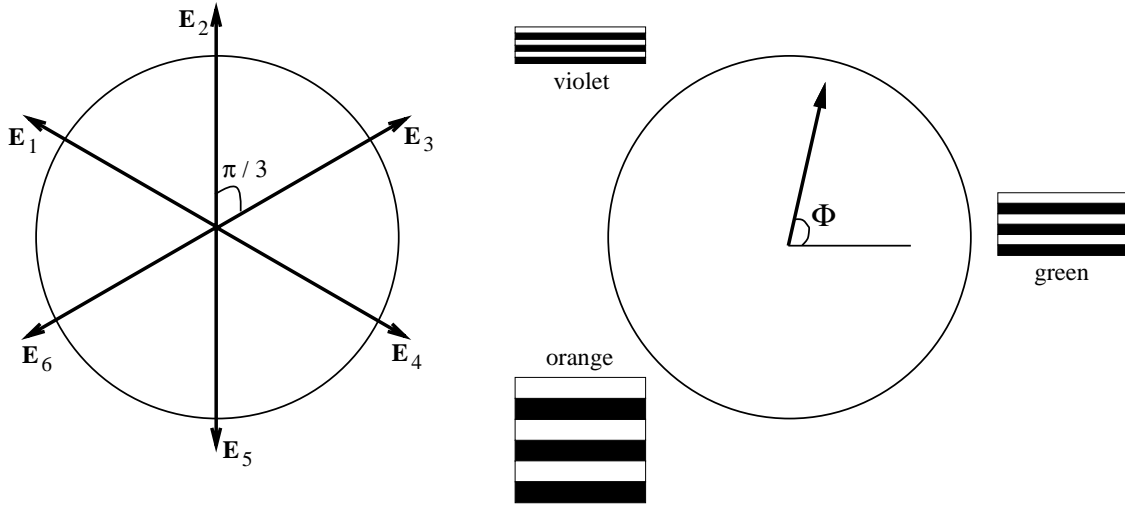


Figure 4.17: *Left: A six level example of how the frequency algorithm works. Right: An intuitive explanation of what the angles mean in the circular spatial frequency representation.*

used in the classification to simplify calculations.

By using phase-gating, the frequency estimate is decomposed into one "dark" and one "bright" frequency estimate, or one estimate for the background and one for the object. The bright frequency estimate is taken as a size descriptor of locally bright areas of the image. The phase from exactly the same scale interval as the frequency controls the division. Phase angles around 0° , meaning locally bright areas, are masked upon the frequency estimate and thus a "bright" frequency estimate is created and the "dark" frequency estimate is created in a similar way. The construction of these images are defined in such a way that edge frequencies, where the phase angle is $\pm 90^\circ$, are neglected.

These two images of size 256×256 have before being used as feature images in the classifier, been averaged with a 31×31 average filter to use larger areas in the calculation of the estimate. The average also spreads data from the center point of the objects outward, making the classification possible over the whole texture.

The classification scheme is as follows:

1. Train the classifier in some areas of textures. This is implemented as a painting in the original image, see Figure 4.23. Different textures are painted in different colors.
2. The classifier collects statistics under the training areas from the feature images.
3. A classification of the whole image is performed. The chosen classifier is a Maximum-Likelihood classifier and the result is found in Figure 4.24.

The results prove that spatial frequency is a good texture descriptor, also described in [61, 99, 16], at least if it is combined with phase-gating. The only area that has been misclassified is the one in the upper left texture. In the misclassified

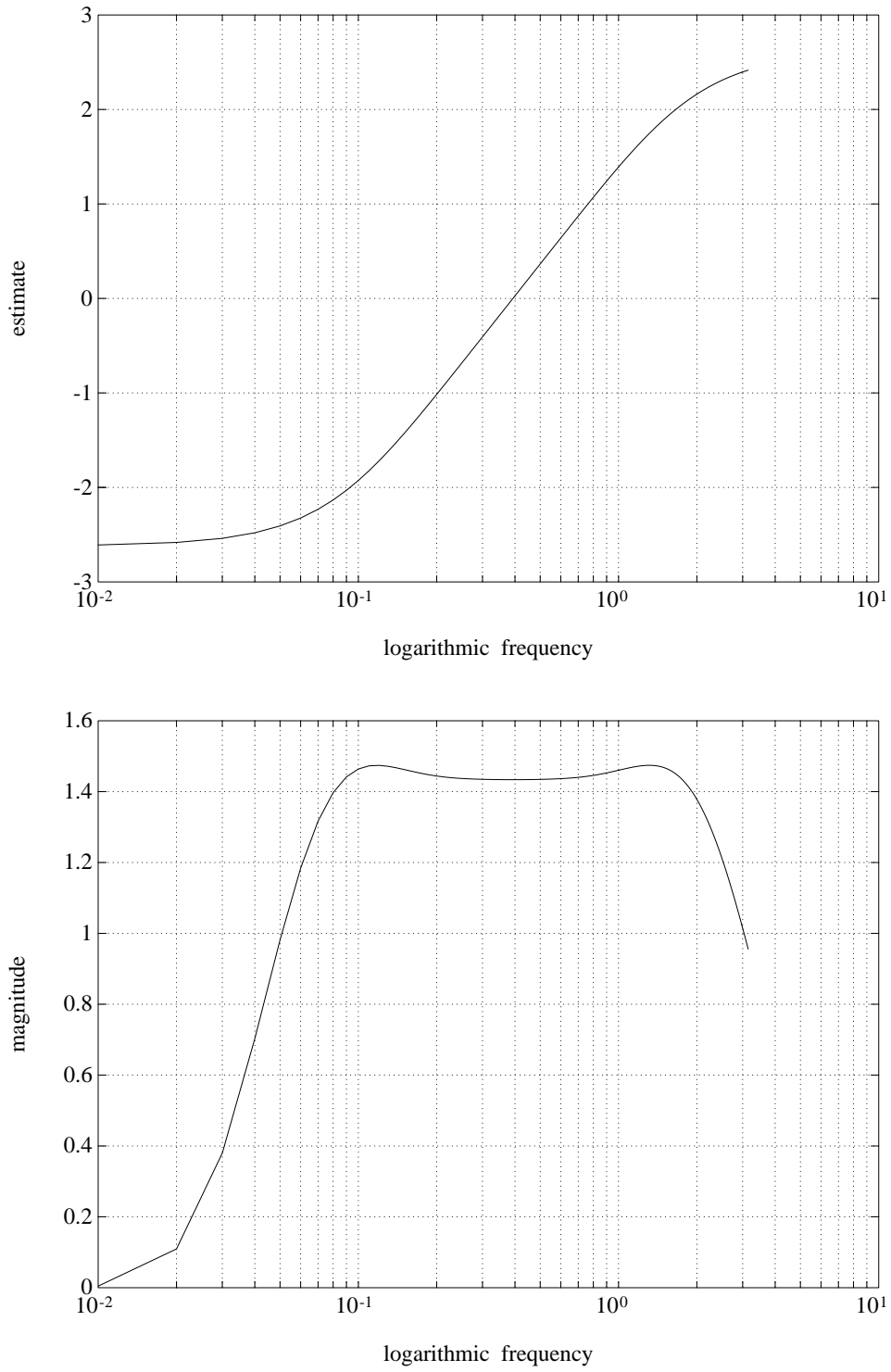


Figure 4.18: *The characteristics of the proposed frequency estimation method. The lower part is confidence and the upper part is the estimate of spatial frequency plotted against a logarithmic scale.*

Figure 4.19: *A spatial frequency description of a synthesized image.*

area, one can see that the size of the pellets is the same as the size of the holes in the sea-weed, and that the distance between the pellets is the same as the size of the sea-weed. In this area, size is not a good discriminator, and the result shows this as well.

4.2.3 Conclusions

This chapter has described a study of an extension of the wavelet transform to 2D signals. The phase of the transform is utilized and a continuous representation of the phase is presented. A new algorithm has been presented, which detects in what scale an event appears and also in what scale it disappears. The scale space is in this way subdivided into a number of intervals isolating individual events. Within each scale interval a consistency check is performed to get the certainty of the detection. It has been shown that by using a three-dimensional phase representation of image data, it is possible to do both the subdivision and the consistency check in a simple manner.

A classification example with a size descriptor from the proposed scale analysis scheme has shown some of the possibilities with this approach.

Figure 4.20: *A spatial frequency description of a natural image.*

Figure 4.21: *The original test image containing ten circles with some additive noise. The areas marked in the phase gating is given in yellow.*

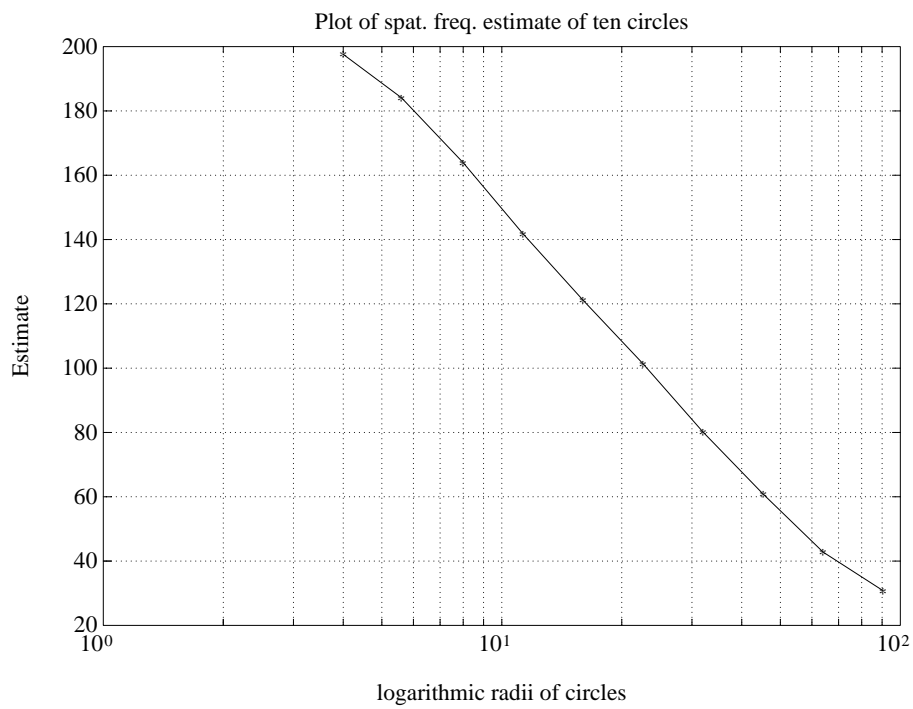


Figure 4.22: *The result of the spatial frequency estimation on Figure 4.21 plotted versus the logarithm of the radii of the circles.*

Figure 4.23: *The training areas for the classifier.*

Figure 4.24: *Classification result of the texture image.*

Appendix

4.A Gaussian Resampling of Images

To provide a simple interaction between different levels of the pyramid, it is preferable that all images have the same size. To preserve spatial position, a combination of odd- and even-sized lowpass filters is required, such that a blob in a fine resolution image keeps its center position if it goes through a subsampling followed by an interpolation to the original size. The lowpass pyramid, which is built with the proposed algorithm, is the traditional one, with one octave between each level. In other words, a subsampling from an image of size $2^N \times 2^N$ to $2^{N-1} \times 2^{N-1}$ will occur between each level. The resampling process is schematically illustrated in Figure 4.25.

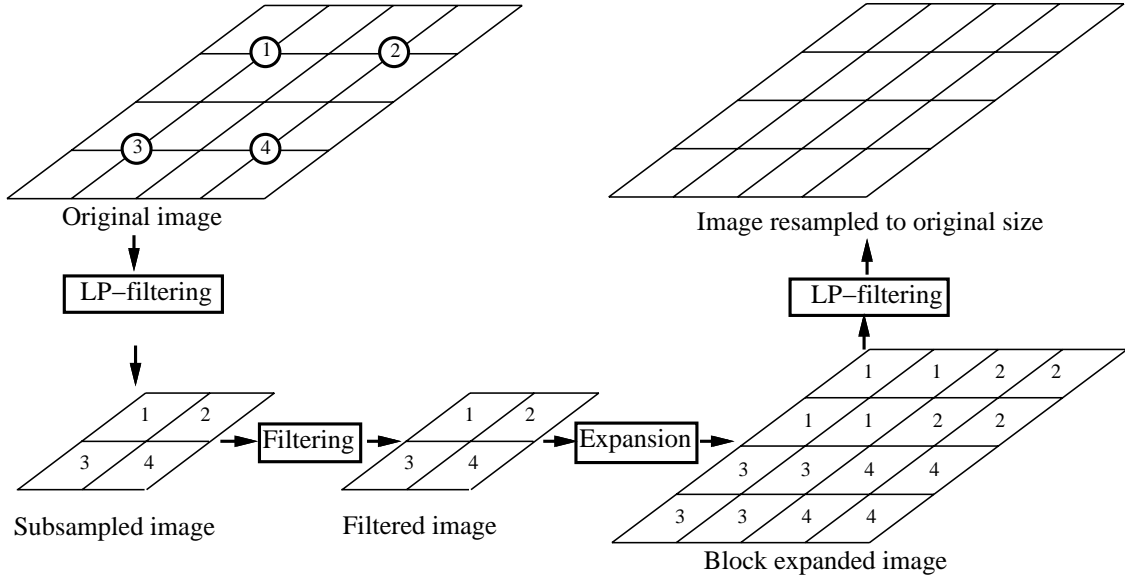


Figure 4.25: *Illustration of the subsampling and the resampling process. The circles indicates center position of Gaussian lowpass filters. The numbers refer to the outputs from these filters.*

4.A.1 Subsampling

The subsampling procedure is explained by a general example. Start with an original image of 512×512 pixels, which will be the bottom of the pyramid. To avoid aliasing effects in the subsampling this image is filtered before the subsampling. The filter proposed here has the following frequency function

$$G(\omega_x, \omega_y) = e^{-(\omega_x^2 + \omega_y^2)/(2\sigma^2)} \quad (4.8)$$

The corresponding spatial function is

$$g(x, y) = \sqrt{2\pi} \sigma e^{-(x^2 + y^2)\sigma^2/2} \quad (4.9)$$

and the parameter

$$\sigma = \pi/4$$

to minimize aliasing effects for a subsampling by a factor of two, and at the same time to preserve as much information as possible in the smaller image.

In the subsampling an even sized filter is used moving the sampling grid half a pixel from the usual grid, i.e. $(-\frac{1}{2}, -\frac{1}{2})$ is the origin for the filter. Filter position $[i, j]$, where i and j are integers, will have the value

$$g((i - \frac{1}{2}), (j - \frac{1}{2}))$$

The center of the filter is then defined in the middle of the four pixels that are connected to the same pixel in the smaller image.

To continue our subsampling example after the convolution with the filter, it is resampled by a factor of two to a 256x256 image. The resampling is done by retaining only every other column and every other row in the filtered image. Almost all aliasing effects have been removed because of the filtering. The effects from the fact that Gaussians are not strictly bandlimited, and the choice of σ from above, (the filter has a value of about 13.5 % of its top value at the frequency $\omega = \frac{\pi}{2}$), are so small that they have not been detected in any, of many, test images. This 256x256 image will be the next level in the lowpass pyramid.

The resampling from 512x512 down to 256x256 pixels moves the frequency spectrum one octave so that $\omega_{512} = \frac{\pi}{2} \longleftrightarrow \omega_{256} = \pi$. It is therefore possible to apply the same Gaussian lowpass filter on this image and resample it to a 128x128 image. The number of operations in the convolution of the 256x256 image will be a factor of four less than it was on the 512x512 image.

The result after the described procedure is a pyramid that contains successively smaller Gaussian images of sizes 512x512, 256x256, . . . , 16x16. The largest image is not filtered with a Gaussian, but it is assumed to be very near Gaussian, because it is more or less blurred before it is sampled and therefore bandlimited, (hopefully Mother Nature does band limitation in a Gaussian fashion).

4.A.2 Interpolation to a Higher Sampling Rate

The procedure for interpolation to a larger image can be divided into two steps:

1. Expand the smaller image, so that each pixel becomes four in the larger image.
2. To avoid block effects in the larger image, the spectrum has to be cut at $\omega_{larger} = \frac{\pi}{2}$, because higher frequencies cannot be represented in the smaller image. The filter described below is applied for this purpose.

The first step is an ordinary block-expansion, and the filter used for interpolation, in the second step, has the same frequency function as the filter used for subsampling, see Eq. 4.8 and Eq. 4.9, but is spatially differently sampled. The usual sampling grid is used in the interpolation scheme, i.e. $g(0,0)$ corresponds to the center of the filter and $g(i,j)$ to filter position $[i, j]$, where the size of the filter will be an odd number.

The combination of resampling and filtering is an approximation of a larger filter. If this larger filter could be applied it should have an odd size to preserve spatial position. Thus the restriction of preservation of position requires that the total filter, after a subsampling followed by an interpolation to the original size of the image, must have an odd size. The motivation for using even-sized filters is that an expansion from one image size to a larger

one can be performed as a convolution with an even-sized filter. An octave based block expansion is the same as to add a column and a row between all old rows and old columns followed by a convolution with a 2×2 filter with the value one in all four positions.

The filtering procedure for subsampling and interpolation to the original size is consequently a combination of a filter of even size (in the subsampling) with another filter of even size (in the expansion), and finally a convolution with a filter of an odd size (in the interpolation) which means that the total filter is an odd-sized filter as required.

Part II

2D — Adaptive Filtering

Chapter 5

Adaptive Filtering of 2D Images

For one-dimensional signals, adaptive filtering often means distinguishing between signal and noise. After an estimation, the filter adapts to the signal in an optimal way, in most cases according to a least squared error criterion. One example of this category is the Wiener filter, e.g. [85]. This filter is directly extendable to higher dimensions, e.g. images. However, the stationary 2D filter has a tendency to blur image details, such as lines and edges, due to its lowpass character.

The idea of the presented adaptive filtering strategy is to enhance detectability, both for a human observer and for algorithms, of features in an image and also to reduce the noise level in the signal. Image enhancement techniques, where the size of a smoothing filter is adjusted according to the signal and the filtering might be anisotropic, i.e. directed along some prescribed orientation, are typical applications. Examples of strategies, where the filter locally adapts to the signal are [66, 1, 5, 36, 21].

In this context a number of interesting parallels to biological systems becomes relevant. The human visual system is much more sensitive to noise in spatially broad, flat, low frequency regions than at edges. Thus an adaptive filtering scheme should preferably perform smoothing over large regions in low frequency areas and smoothing over small regions in high frequency areas, e.g. near edges. The effect of noise in the vicinity of a directed feature depends on its orientation relative to the feature. Noise in the same orientation may enhance detectability, and noise in the perpendicular orientation will reduce it, [71, 79, 58].

This chapter will describe an idea and an implementation of an anisotropic filtering strategy. The similarities between the enhancement algorithm in [66] and the presented tensor controlled adaptive filtering are discussed.

5.1 Representation and Estimation of Orientation

The fundamental idea behind this work is that local one-dimensionality, i.e. a neighbourhood with variations in just one orientation, is a generally useful model for vision systems. This means grey scale variations such as lines or edges, which are the

foundation of many image processing algorithms. These kinds of one-dimensional structures are defined by two different properties; first, by their position, and second, by their orientation. One way of estimating the first aspect is to use a local window in the scene, i.e. the image, to decide the position. Inside this window, or neighborhood, a number of directed filters are applied and the filter outputs combined to estimate the local oriented energy, [39, 67]. This oriented energy corresponds either to an edge, a line or a mixture between such events. The algorithm described below also gives a certainty measure that tells how well the neighbourhood is described by the one-dimensional model.

The estimation of orientation can be achieved by splitting up the Fourier transform and measuring the energy contribution in the different sections. If an orientational field is to be used as contextual control of an adaptive filter, a continuous orientation representation simplifies the procedure. This enables relaxation of orientation, i.e. to use larger areas to update the certainty of the estimate as simple as by local averaging in the representation space.

There is one representation constraint for orientation, which has not been mentioned. A line which is rotated π radian will have the same orientation as the original line. An appropriate representation should naturally map both these lines to the same statement. For edges there is a possibility to have a sign meaning brighter to the right or to the left. For more complex fields than grey scale images it is impossible to talk about the direction of border edges between areas. Considering color, it is not possible to define that red $<$ blue $<$ green or something similar. The only reasonable statement which can be made is that we have a borderline between differently colored areas.

To handle directional data in a proper way the directional aspect must be treated correctly. In the case of tangent directions without sign, this means that the data will be distributed over an angle interval $[\varphi_0, \pi + \varphi_0[$. If the problem then is to find the mean of a particular distribution, it is easy to show that the mean will depend on φ_0 . Consider for example a distribution that contains only two samples, the angles ε and $\pi - \varepsilon$, where ε is a small value. If the interval is defined as $[0, \pi[$ the mean will be $\pi/2$, but if the interval is defined as $[-\pi/2, \pi/2[$ the mean will be 0. This is of course not acceptable, and there are statistical distributions that can handle these problems, see [77] for an extensive overview.

A general representation of the orientation in image processing can be achieved using a “double angle” representation, of the tangent direction, [39]. This means that a line with the orientation φ will be mapped onto the same statement as a line with orientation $(\varphi + \pi)$ because $2\varphi = 2(\varphi + \pi)$, since the angular variation is modulo 2π . The same holds for edges, where it is the same as ignoring the sign. This vector representation can be used not only on grey scale images, but on every type of images. The magnitude of the vector has been used as a confidence measure of the actual estimate, which is represented by the argument of the vector, [40]. The main advantage with this representation is that orientation will be described in a continuous way, making ordinary signal processing concepts, e.g. averaging and differentiation, meaningful.

Presenting features in a general form independent of the particular feature has

many advantages. One benefit of this vector representation is data visualization using color. By letting the magnitude of a complex number control the intensity of the pixels and letting the argument control the hue, an image is constructed that creates an impression of the data information that is easy to interpret. High intensity means high confidence in the estimate, while the color states the class membership. In areas with high confidence the color will be clearly visible, while areas with low certainty will be dark, and the color will be harder to discern.

5.1.1 Orientation Representation

The orientation estimates can be represented in a continuous way by tensors, [59]. These tensors are, for edges, in essence the scaled outer product of the gradient direction, but on real 2D data there exist, of course, cases where there is more than one orientation, i.e. the scalar function is not locally one-dimensional. Neglecting this fact for a moment, the tensor in the ideal case of a strictly one-dimensional neighbourhood can be written

$$\mathbf{T} = \frac{1}{x} \mathbf{x} \mathbf{x}^t = \frac{1}{x} \begin{pmatrix} x_1^2 & x_1 x_2 \\ x_1 x_2 & x_2^2 \end{pmatrix} \quad (5.1)$$

where $\mathbf{x} = (x_1, x_2)^t$ is a vector pointing in the direction of the local gradient and $x = \sqrt{x_1^2 + x_2^2}$ is the norm of \mathbf{x} .

The tensor, \mathbf{T} , fulfills three basic requirements for a proper orientation representation, [68]:

1. **The uniqueness requirement**, i.e. there is just one representation point for each orientation. This is not trivial, considering that the orientation of a line is modulo π . This means that the orientation of a line with the gradient direction \mathbf{x} should be mapped onto the same point as a line with the orientation $-\mathbf{x}$. This ambiguity was resolved by Granlund [39] by doubling the angle of the gradient orientation. The tensor mapping, Eq. 5.1, also fulfills this requirement since the tensor is a quadratic form and the sign is thus neglected.
2. **The uniform stretch**, i.e. the representation must be equally sensitive to orientation changes in all orientations. The implication of this is that the angle metric of the orientation is preserved, i.e. similar events are represented “close to each other” and the representation does not have discontinuities.
3. **Polar separability**, i.e. the estimated orientation should be represented independently of the strength of the signal and that the norm of the tensor is invariant to rotation of the signal. The tensor norm is then possible to use as a certainty measure not influencing the actual estimate.

5.1.2 Orientation Estimation

Recall from Chapter 3 that a real valued 2D function, e.g. an image, has a hermitian Fourier spectrum, i.e. $F(\omega) = F^*(-\omega)$, where $*$ denotes complex conjugate. Locally

one-dimensional signals, such as lines and edges, will have an energy distribution on one single line in the Fourier domain.

The orientation algorithm then requires a number of precomputed quadrature filters evenly spread in one half of the Fourier space. The minimum number of quadrature filters required for orientation estimation in 2D is 3, [67], where the filters are evenly distributed, i.e. the angular difference of two filters is $\pi/3$. In the following these orientations are taken as 0, $\pi/3$ and $2\pi/3$, or in vector notation:

$$\begin{aligned}\hat{\mathbf{n}}_1 &= 0.5 \begin{pmatrix} 2, & 0 \end{pmatrix}^t \\ \hat{\mathbf{n}}_2 &= 0.5 \begin{pmatrix} 1, & \sqrt{3} \end{pmatrix}^t \\ \hat{\mathbf{n}}_3 &= 0.5 \begin{pmatrix} -1, & \sqrt{3} \end{pmatrix}^t\end{aligned}\tag{5.2}$$

A quadrature filter designed with a suitable bandpass function, $H_\rho(\omega)$, e.g. the lognormal function described in Chapter 3, is given by:

$$\begin{cases} H_k(\omega) = H_\rho(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 & \text{if } \omega \cdot \hat{\mathbf{n}}_k > 0 \\ H_k(\omega) = 0 & \text{otherwise} \end{cases}\tag{5.3}$$

where $\omega = (\omega_1, \omega_2)$ is the frequency coordinate, $\hat{\omega}$ is a unit vector directed along ω , and $\omega = \sqrt{\omega_1^2 + \omega_2^2}$. The spatial filter coefficients are found by a straightforward 2D-DFT or by use of an optimization technique, see Appendix 3.A. The resulting filter is complex-valued, with the real part being even and the imaginary part being odd. This procedure is used to obtain three complex-valued quadrature filters.

The tensor describing the neighbourhood is given by, [60]:

$$\mathbf{T}^e = \frac{4}{3} \sum_k q_k (\mathbf{N}_k - \frac{1}{4} \mathbf{I})\tag{5.4}$$

where q_k denotes the magnitude of the output from filter k and \mathbf{I} is the unity tensor. $\mathbf{N}_k = \hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^t$ denotes the direction of the filter expressed in the tensor representation, i.e.

$$\begin{aligned}\mathbf{N}_1 &= \hat{\mathbf{n}}_1 \hat{\mathbf{n}}_1^t = \frac{1}{4} \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} \\ \mathbf{N}_2 &= \hat{\mathbf{n}}_2 \hat{\mathbf{n}}_2^t = \frac{1}{4} \begin{pmatrix} 1 & \sqrt{3} \\ \sqrt{3} & 3 \end{pmatrix} \\ \mathbf{N}_3 &= \hat{\mathbf{n}}_3 \hat{\mathbf{n}}_3^t = \frac{1}{4} \begin{pmatrix} 1 & -\sqrt{3} \\ -\sqrt{3} & 3 \end{pmatrix}\end{aligned}\tag{5.5}$$

Rewriting Eq. 5.4 in terms of an algorithm:

1. Convolve the input data with the three complex-valued filters, i.e. perform six scalar convolutions.
2. Compute the magnitude of each complex-valued filter output by

$$q_k = \sqrt{q_{ke}^2 + q_{ko}^2}$$

where q_{ke} denotes the output of the real part of filter k and q_{ko} denotes the output of the imaginary part of filter k .

3. Compute the tensor \mathbf{T}^e by Eq. 5.4, i.e.

$$\mathbf{T}^e = \frac{4}{3} \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix}$$

where

$$\begin{aligned} T_{11} &= q_1 + A(q_2 + q_3) - S \\ T_{22} &= B(q_2 + q_3) - S \\ T_{12} &= C(q_2 - q_3) \end{aligned}$$

for

$$\begin{aligned} S &= \frac{1}{4} \sum_{k=1}^3 q_k \\ A &= \frac{1}{4} \\ B &= \frac{3}{4} \\ C &= \frac{\sqrt{3}}{4} \end{aligned}$$

By studying the distribution of the eigenvalues and eigenvectors of \mathbf{T}^e a description of the neighbourhood is obtained. Utilizing the fact that \mathbf{T} (and \mathbf{T}^e) has a regular form, i.e. it is symmetric and positive semidefinite, the eigenvectors can be written in a simpler form.

Let \mathbf{e}_1 and \mathbf{e}_2 be the eigenvectors to \mathbf{T}^e , where \mathbf{e}_1 has the largest eigenvalue and \mathbf{e}_2 has the least. The directions of \mathbf{e}_1 and \mathbf{e}_2 are denoted by φ_1 and φ_2 respectively. Then

$$\begin{aligned} 2\varphi_1 &= 2 \arg(e_{1x}, e_{1y}) = \arg(T_{11} - T_{22}, 2T_{12}) \\ 2\varphi_2 &= 2 \arg(e_{2x}, e_{2y}) = \arg(T_{22} - T_{11}, -2T_{12}) \end{aligned} \tag{5.6}$$

It can be noted, [94], that the two eigenvectors in the double angle space will be parallel, but with reversed signs. This follows from the fact that the eigenvectors of symmetric tensors, or matrices, are perpendicular. In 2D a doubling of the argument of perpendicular vectors necessarily means that they are oppositely directed.

In [39, 67] an orientation algorithm is suggested by combining filter outputs directly in a double angle fashion. By combining the filter outputs vectorially according to Eq. 5.7, giving an orientation estimate corresponding to Figure 4.1.

$$\mathbf{ori} = \frac{4}{3} \sum_{k=1}^3 q_k e^{2i(k-1)\pi/3} = m e^{i\varphi_{ori}} \tag{5.7}$$

Direct calculation shows that

$$\varphi_{ori} = \arg(\mathbf{ori}) = 2\varphi_1$$

The magnitude m in Eq. 5.7 will depend on

1. the energy in the considered neighbourhood, i.e. two neighborhoods that only differ in their local energies, will have different magnitudes and the difference in magnitude will be linear w.r.t. the RMS of the energy.
2. how well the neighbourhood fits into the one-dimensionality model. A perfectly one-dimensional structure will give rise to a higher magnitude than for example a cross.

The magnitude, m , is therefore referred to as a certainty measure, and the angle, φ_{ori} , is the orientation estimate expressed in “double angle” representation. The division of the result from operations in one part corresponding to the certainty and one part corresponding to the actual estimate is a fundamental issue, originally discussed in [39].

The reason to use a tensor as the representation is the richer description of the neighbourhood. The spectrum theorem, which is found in all standard textbooks of linear algebra, e.g. [43], allows us to write \mathbf{T}^e as

$$\mathbf{T}^e = \lambda_1 e_1 e_1^t + \lambda_2 e_2 e_2^t \quad (5.8)$$

where λ_1 is the largest eigenvalue to \mathbf{T}^e and λ_2 the least.

This means that not only the largest eigenvalue is given, describing the energy in the dominant orientation of the neighbourhood, but also the least eigenvalue describing the energy in the perpendicular direction.

The correspondence between the eigenvalues and the magnitude of \mathbf{ori} follows from the observation that \mathbf{e}_1 and \mathbf{e}_2 are oppositely directed in the double angle space.

$$m = \text{mag}(\mathbf{ori}) = (\lambda_1 - \lambda_2) \quad (5.9)$$

By taking the norm of the tensor \mathbf{T}^e as the Frobenius norm, i.e.

$$\|\mathbf{T}^e\|_{frob}^2 = \sum T_{ij}^2 = \sum \lambda_i^2 = \frac{2}{3} \left(\sum q_k^2 + \frac{1}{3} ((q_1 - q_2)^2 + (q_1 - q_3)^2 + (q_2 - q_3)^2) \right) \quad (5.10)$$

the energy in the tensor representation and in the vector representation will be equal for an ideal input, i.e. a one-dimensional signal. The same holds also if the norm is chosen as the maximum norm, i.e.

$$\|\mathbf{T}^e\|_{max}^2 = \lambda_1^2 \quad (5.11)$$

5.2 Orientation Adaptive Filtering

The purpose of this section is to describe how the estimated shape and orientation of \mathbf{T}^e can be used to adapt a filter behaving in the same way as the signal. One suitable function, F , is, according to Knutsson, Wilson and Granlund [66], a linear weighting between an isotropic bandpass filter and an anisotropic filter having the same bandpass function

$$F(\omega) = F(\omega, \varphi) = F(\omega) \left(g_1(\lambda_1, \lambda_2) (\hat{\omega} \cdot \mathbf{e}_1)^2 + g_2(\lambda_1, \lambda_2) \right) \quad (5.12)$$

where $F(\omega)$ is a bandpass function with high upper cut-off frequency, g_1 is the weighting function for the anisotropic part and g_2 is the weighting function for the isotropic part.

The first term in the sum will be a line filter in the direction of the eigenvector corresponding to the largest eigenvalue. The second part will be an isotropic filter. Typical choices of the weighting functions g_1 and g_2 are

$$\begin{aligned} g_1(\lambda_1, \lambda_2) &= \frac{\lambda_1 - \lambda_2}{\lambda_1} \\ g_2(\lambda_1, \lambda_2) &= \frac{\lambda_2}{\lambda_1} \end{aligned} \quad (5.13)$$

Choosing these functions will result in a linear weighting between the isotropic and the anisotropic filter. It also implies that the maximum norm, Eq. 5.11, of the tensor representation is the most suitable when the tensor should be used to adapt an anisotropic filter.

Is it then possible to adapt a filter according to Eq. 5.12, from the tensor \mathbf{T}^e ?

Proposition 1

Assume the use of three polar separable filters having the same radial filter function, $F(\omega)$, and angular $(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$, where the filter directions $\hat{\mathbf{n}}_{\mathbf{k}}$ is given in Eq. 5.2. The adaptive filter of Eq. 5.12 is then given by, [64]

$$F(\omega) = F(\omega) \frac{4}{3} \sum_{k=1}^3 \hat{\mathbf{T}}^e \bullet (\mathbf{N}_k - \frac{1}{4}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2 \quad (5.14)$$

$\hat{\mathbf{T}}^e$ is the normalized estimate, the choice of normalization will give different types of filter shapes, in other words it will affect the functions g_1 and g_2 . If $\hat{\mathbf{T}}^e$ is normalized with λ_1 , the resulting filter will be interpolated using the chosen g_1 and g_2 from Eq. 5.13. \mathbf{I} is the identity tensor and \bullet symbolizes the tensor inner product defined according to Eq. 5.15

$$\mathbf{A} \bullet \mathbf{B} = \sum_{i,j} A_{ij} B_{ij} \quad (5.15)$$

Proof of proposition 1

□

Using the spectrum theorem, Eq. 5.8, to divide the estimated structure of the neighbourhood in two parts, each one pointing in *one* direction, it is possible to write the projection as:

$$\hat{\mathbf{T}}^e = \frac{1}{\lambda_1} \mathbf{T}^e = \mathbf{e}_1 \mathbf{e}_1^t + \frac{\lambda_2}{\lambda_1} \mathbf{e}_2 \mathbf{e}_2^t \quad (5.16)$$

It is now possible to study each part of the tensor independently, starting with the first one. Setting $\mathbf{e}_1 = (a, b)^t$, where $\sqrt{a^2 + b^2} = 1$ leads to the control tensor

$$\mathbf{T}' = \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} \quad (5.17)$$

The weighting coefficients in the interpolation is calculated according to Eq. 5.15

$$\begin{aligned}
(\mathbf{N}_1 - \frac{1}{4}\mathbf{I}) \bullet \mathbf{T}' &= \begin{pmatrix} 1 - \frac{1}{4} & 0 \\ 0 & -\frac{1}{4} \end{pmatrix} \bullet \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} = \frac{1}{4}(3a^2 - b^2) \\
(\mathbf{N}_2 - \frac{1}{4}\mathbf{I}) \bullet \mathbf{T}' &= \frac{1}{4} \begin{pmatrix} 1 - 1 & \sqrt{3} \\ \sqrt{3} & 3 - 1 \end{pmatrix} \bullet \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} = \frac{1}{4}(2\sqrt{3}ab + 2b^2) \\
(\mathbf{N}_3 - \frac{1}{4}\mathbf{I}) \bullet \mathbf{T}' &= \frac{1}{4} \begin{pmatrix} 1 - 1 & -\sqrt{3} \\ -\sqrt{3} & 3 - 1 \end{pmatrix} \bullet \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} = \frac{1}{4}(-2\sqrt{3}ab + 2b^2)
\end{aligned}$$

Denoting the filter functions by $\cos^2 \varphi$, $\cos^2(\varphi - \pi/3)$ and $\cos^2(\varphi - 2\pi/3)$ for filter number 1, 2 and 3 respectively leads to the following interpolated filter:

$$\begin{aligned}
F'(\omega) &= F(\omega) \frac{4}{3} \frac{1}{4} \left((3a^2 - b^2) \cos^2 \varphi + \right. \\
&\quad \left. (2\sqrt{3}ab + 2b^2) \cos^2(\varphi - \pi/3) + \right. \\
&\quad \left. (-2\sqrt{3}ab + 2b^2) \cos^2(\varphi - 2\pi/3) \right) = \\
&= F(\omega) \frac{1}{2} (a^2 + b^2 + (a^2 - b^2) \cos 2\varphi + 2ab \sin 2\varphi)
\end{aligned} \tag{5.18}$$

Remembering that $(a, b)^t$ is an eigenvector to \mathbf{T}^e allows us to put $(a, b)^t = (\cos \varphi_0, \sin \varphi_0)^t$, which gives

$$\begin{aligned}
F'(\omega) &= F(\omega) \frac{1}{2} \left(\cos^2 \varphi_0 + \sin^2 \varphi_0 + \right. \\
&\quad \left. (\cos^2 \varphi_0 - \sin^2 \varphi_0) \cos 2\varphi + \right. \\
&\quad \left. 2 \cos \varphi_0 \sin \varphi_0 \sin 2\varphi \right) = \\
&= F(\omega) \cos^2(\varphi - \varphi_0)
\end{aligned} \tag{5.19}$$

Thus the first part in Eq. 5.16 will adapt a filter to the main orientation of the structure found in the neighbourhood.

The second part of Eq. 5.16 will in the same way as above give rise to a filter perpendicular to the filter in Eq 5.19. The angular function of this filter can then be denoted $(b, -a)^t = (\sin \varphi_0, -\cos \varphi_0)^t$ and the resulting filter will be

$$F''(\omega) = F(\omega) \sin^2(\varphi - \varphi_0) \tag{5.20}$$

The total filter can now be written as

$$\begin{aligned}
F(\omega) &= F'(\omega) + \frac{\lambda_2}{\lambda_1} F''(\omega) = \\
&= F(\omega) \left(\cos^2(\varphi - \varphi_0) + \frac{\lambda_2}{\lambda_1} \sin^2(\varphi - \varphi_0) \right) = \\
&= F(\omega) \left(\frac{\lambda_1 - \lambda_2}{\lambda_1} \cos^2(\varphi - \varphi_0) + \frac{\lambda_2}{\lambda_1} \right)
\end{aligned} \tag{5.21}$$

which is the filter proposed in Eq. 5.12.

□

In some applications the angular function $\cos^2 \varphi$ might be too broadly tuned. However, by iterations of the same filter it is possible to get more narrowly tuned filters. The filter will get the angular variation $\cos^{2i} \varphi$ after i iterations. Iterations of the filter put great demands on the radial function since it will be iterated as well, see Eq. 5.24

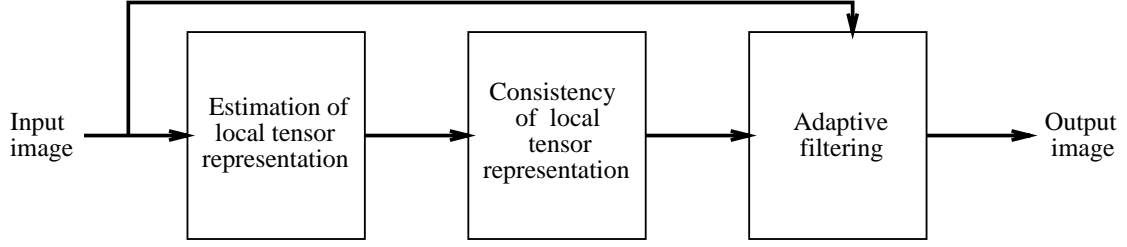


Figure 5.1: *Overview of the orientation adaptive filtering algorithm.*

To use the presented filtering strategy for noise reduction, a suitable lowpass filter must be applied to keep the low frequency components, including DC, of the original signal. The decision of where to use the bandpass filter is controlled by the energy of the tensor according to Frobenius $\|\mathbf{T}^e\|_{frob}$. The adaptive filter, F_{ada} could now be written as

$$F_{ada}(\omega) = F_{LP}(\omega) + \sqrt{\|\mathbf{T}^e\|_{frob}} F_{HP}(\omega) \frac{4}{3} \sum_{k=1}^3 \hat{\mathbf{T}}^e \bullet (\mathbf{N}_k - \frac{1}{4}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 \quad (5.22)$$

where F_{LP} is the lowpass filter, see Eq. 5.23, and F_{HP} is a bandpass filter with rather high upper cut-off frequency, see Eq. 5.24.

Note that no eigenvalue decomposition is necessary in the interpolation strategy. The largest eigenvalue must be available to calculate the maximum norm, Eq. 5.11, but this is the only component needed. This is not critical in the case of 2 by 2 tensor, but for higher dimensional signals an eigenvalue decomposition is computationally demanding.

5.2.1 Implementation

Implementing these ideas is straightforward. The implementation is a three-stage operation which is summarized in the block scheme in Figure 5.1.

The estimation of the local information, represented in the tensor, is accomplished with quadrature filters to get a smooth phase independent tensor field. Typically rather broadband lognormal filters, Eq. 3.16, are used in the estimation. In the result section the applied estimation filter has the bandwidth $B_{logn} = 4$ and the center frequency $\omega_i = \pi/4$. These filter results are combined to a tensor according to Eq. 5.4.

The continuity of the tensor representation is utilized when a tensor averaging operation is applied to the estimates. This type of local averaging in the representation space originates from [67]. The averaging affects both the eigenvectors and the relation of the eigenvalues. To give a few examples:

1. In a truly one-dimensional neighborhood, with some noise added, the averaging will work as relaxation of the orientation with respect to a larger spatial region. This means that rapid changes in the representation space are suppressed.
2. In the proximity of a corner the averaging mainly will change the shape of the tensor. In this kind of neighborhoods there will exist two perpendicular

“one dimensional” tensors. Averaging of these gives rise to a tensor whose eigenvalues are approximately equal, meaning that the shape of the tensor will be isotropic.

In the first of these two cases, the implication is that the adaptive filter will have one dominant orientation, sharpening the one-dimensional signal. In the second case the adaptive filter will be isotropic and the corner, which has no unique orientation, will be unaffected, i.e. the filter will be of allpass type.

In the adaptive filtering four separate filters are applied. The filters used are one isotropic lowpass and three anisotropic bandpass filters with high upper cut-off frequency. The filter function in the Fourier domain for the lowpass filter is given according to Eq. 5.23, [66].

$$F_{LP}(\omega) = \begin{cases} \cos^2(\pi\omega/1.8) & \omega \leq 0.9 \\ 0 & \omega > 0.9 \end{cases} \quad (5.23)$$

The radial part of the three bandpass filters is given of Eq. 5.24.

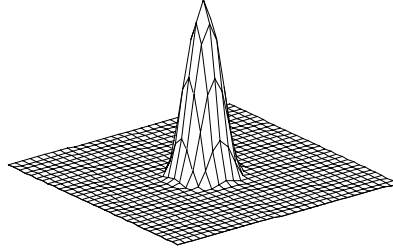
$$F_{HP}(\omega) = \begin{cases} 1 - \cos^2(\pi\omega/1.8) & \omega \leq 0.9 \\ 1 & 0.9 < \omega \leq \pi - 0.9 \\ \cos^2(\pi(\omega - \pi + 0.9)/1.8) & \pi - 0.9 < \omega \leq \pi \end{cases} \quad (5.24)$$

while the angular function is, as said before, $(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$.

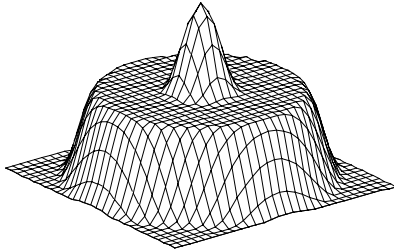
The values, 0.9 and 1.8, of the constants in Eq. 5.23 and 5.24 are chosen under the demand that the filters must be possible to realize on a 15x15 grid.

The sum of the isotropic bandpass filter, i.e. $\frac{2}{3} \sum_{k=1}^3 F_{HP}(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$, and the lowpass filters is flat in the frequency domain. This is a natural requirement in the design phase of the filters. The reason is that with no a priori information about the image to be filtered, all spatial frequencies must be regarded equally important. Another reason is that the flatness is necessary if the filters are supposed to be used iteratively. If there is a deviation in the spectrum, which otherwise is flat, the deviation will grow larger if the filter is used iteratively.

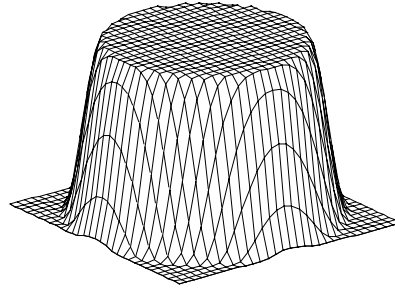
To give an impression of what the filters look like for different control tensors, five different interpolations in the Fourier domain have been plotted in Figure 5.2.



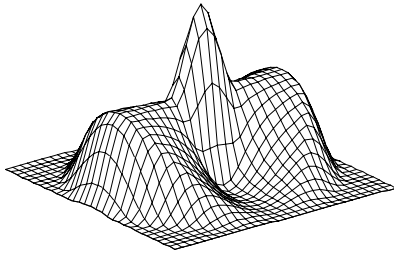
a)



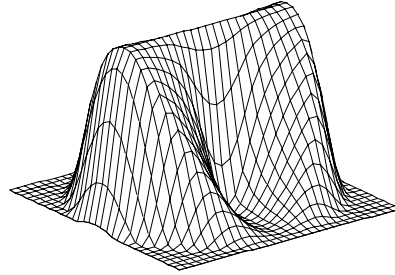
b)



c)



d)



e)

Figure 5.2: *The filter shape for different relations between the eigenvalues λ_1 and λ_2 of the control tensor.*

- a) $\lambda_1 = 0, \lambda_2 = 0$ b) $\lambda_1 = 1/2\sqrt{2}, \lambda_2 = 1/2\sqrt{2}$ c) $\lambda_1 = 1/\sqrt{2}, \lambda_2 = 1/\sqrt{2}$
d) $\lambda_1 = 1/2, \lambda_2 = 0$ e) $\lambda_1 = 1, \lambda_2 = 0$

Chapter 6

Scale Adaptive Filtering

An extension of the orientation adaptive filtering strategy was proposed in [46] where a bandpass or Laplacian pyramid [18] is generated and each level is filtered according to the anisotropic filtering strategy. In this chapter will this strategy be further described. The adaptive filter is constructed as a linear combination of three fixed filters on each scale level. The three weighting coefficients are obtained from the local energy, the local orientation, and the consistency of local orientation, all represented by 2x2 tensors. The orientation adaptive filtering is applied on each level of a bandpass pyramid. The resulting filter is interpolated from all bandpass levels, and spans over more than 6 octaves. It is possible to reconstruct an enhanced original image from the filtered images. The performance of the reconstruction algorithm displays two desirable but normally contradictory features, namely edge enhancement and an improvement of the signal-to-noise ratio. The results are satisfactory on a wide variety of images from moderate signal-to-noise ratios to low, even lower than 0 dB, SNR.

6.1 The Algorithm

Initially a brief overview of the algorithm following the implementation will be given. Each part of the algorithm will then be explained in detail.

6.1.1 Overview

The overview follows the block scheme in Figure 6.1.

The **first** box is a more or less standard implementation of an octave based difference of lowpass pyramid (DOLP), [18], meaning that the images in the bandpass, or Laplacian, pyramid fit together such that the sum of them reconstructs the original image. The octave based pyramid has bandpass images with typical sizes 512^2 , 256^2 , ... 32^2 and an image of size 16^2 containing the low frequency components including DC.

In the **second** box the local energy and the local orientation are estimated on each level of the pyramid. The feature extraction is performed by a set quadrature

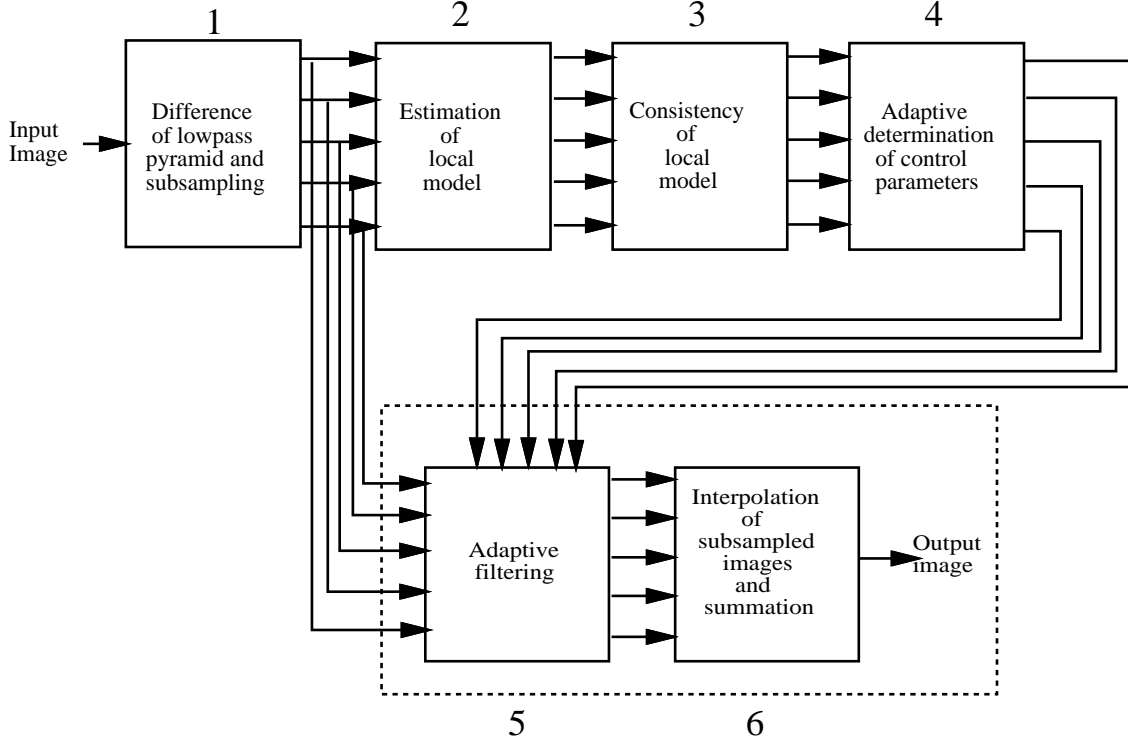


Figure 6.1: *Overview of the scale and orientation adaptive algorithm.*

filters, see Section 3. The use of quadrature filters ensure that the local energy is independent of the phase of the signal, e.g. independent of whether it is a line or an edge, which is important for the smoothness of the output. The tensor representation of orientation, Section 5.1, is used because it provides a continuous representation space.

The continuity of the representation space is utilized in the **third** box where the consistency, or relaxation, of orientation information is computed as tensor averaging of the orientation information. The relaxation provides larger spatial regions to control the certainty of the orientation estimation. A novel feature of this algorithm is that the relaxation is not only made within a single scale level, but incorporating several levels.

In the **fourth** box the parameters to a nonlinear transfer function are determined, interpreting the estimated information. The parameter setting is based on global statistics of the local energy.

The first four boxes taken together are used to estimate a local model for each neighborhood in the image.

The **fifth** and **sixth** boxes can be seen as a model based reconstruction of the original signal increasing the signal-to-noise ratio, [66]. The proposed bandpass environment enables a efficient implementation of edge enhancement in edge areas, simply by putting more weight on the higher frequency levels in the reconstruction.

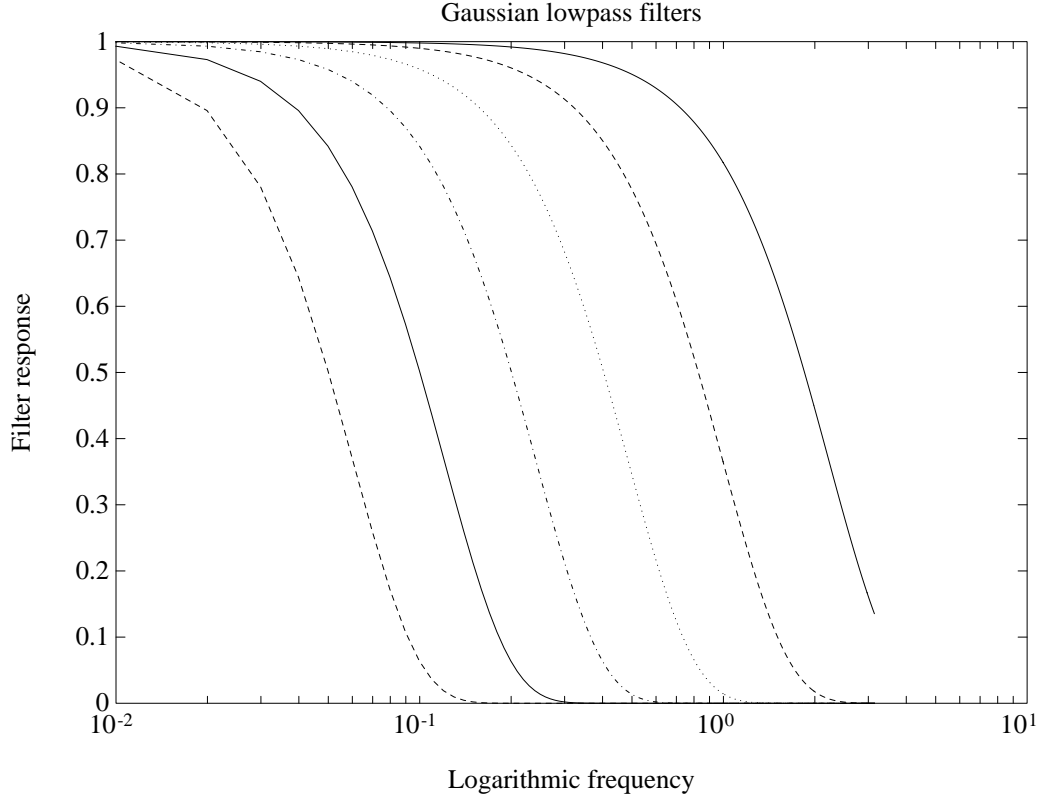


Figure 6.2: *The frequency response of the Gaussian lowpass filters.*

6.1.2 Generation of the Bandpass Pyramid

Computational efficiency is one reason for using subsampled images, and as pointed out in Section 3 it also provides means to implement a wavelet transform. The simplicity of reconstructing the original image from the DOLP pyramid is maybe the major reason to use a bandpass pyramid instead of the lowpass pyramid used in the Chapter 4. This type of scale pyramid is widely used for a number of computer vision and image processing tasks [18, 23].

In the current implementation a Gaussian approach is chosen. The filter used has a spatial standard deviation of $4/\pi$, with one exception. The first applied filter has the standard deviation of $2/\pi$. The latter filter with standard deviation $2/\pi$ can be used to generate a half-octave based pyramid. The first filter is better suited for an octave-based approach. The reason to use the Gaussian with the smaller spatial standard deviation is that otherwise the bandpass level with the highest cut-off frequency, which actually is a highpass filter, collects a too large proportion of the signal energy, see Appendix 6.A. The frequency of the lowpass filters are plotted in Figure 6.2.

The bandpass pyramid is obtained by taking subsequent differences between the adjacent lowpass filters. The frequency response of the bandpass filters are plotted in Figure 6.3. The sum of the bandpass filters is also plotted in Figure 6.3, where it is easy to see why the reconstruction is so simple.

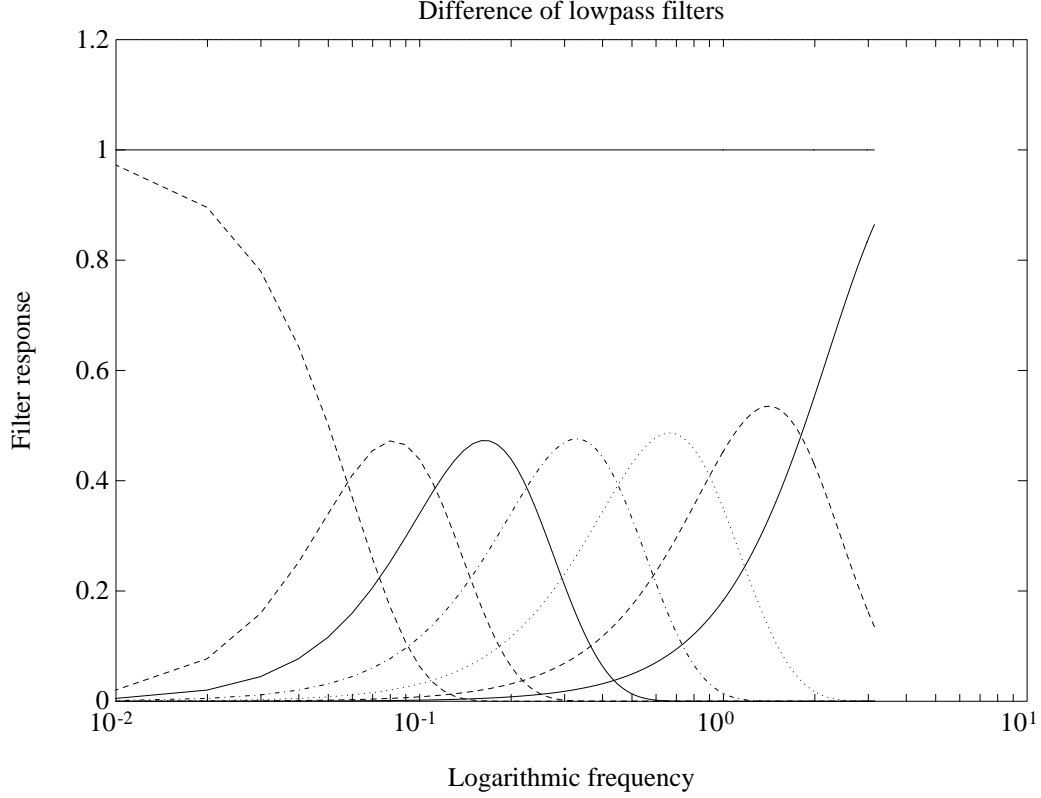


Figure 6.3: *The difference of lowpass filters and the sum of the bandpass filters.*

The Implementation

There are some aspects that must be taken into account when implementing sub-sampled, e.g. bandpass, pyramids. One will be treated below and some others are mentioned in Appendix 4.A. It is desirable that the reconstruction could be made using pixelwise operations. The pyramid is implemented as follows.

Define the Gaussian filters, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$, as

$$\begin{aligned} g_1(\mathbf{x}) &= \frac{\pi}{8} e^{-\pi^2 x^2/8} \\ g_2(\mathbf{x}) &= \frac{\pi}{32} e^{-\pi^2 x^2/32} \end{aligned} \quad (6.1)$$

where $x = \|\mathbf{x}\|$. The first level in the bandpass pyramid is easy to obtain since the sampling grid is the same as for the original image, i.e.

$$\begin{aligned} lp_1 &= g_1 * org \\ bp_1 &= org - lp_1 \end{aligned} \quad (6.2)$$

where org is the original image and $*$ denotes convolution.

The second level should in the same way be calculated as

$$lp_2 = g_2 * lp_1$$

$$bp_2 = lp_1 - lp_2 \quad (6.3)$$

and the original image is to be reconstructed as $org = bp_1 + bp_2 + lp_2$. But the lowpassed version, lp_2 , of the original is supposed to be subsampled in order to obtain a pyramid. However, the original image can not generally be reconstructed using the subsampled image, ${}^{2\downarrow}lp_2$, of lp_2 . The superscript ${}^{2\downarrow}$ indicates that the image has been subsampled by a factor 2 in each dimension, according to the scheme in Appendix 4.A. To ensure that the original can be exactly reconstructed bp_2 is given by

$$bp_2 = lp_1 - {}^{2\uparrow}({}^{2\downarrow}lp_2) \quad (6.4)$$

This means that before computing the difference, the image that is supposed to be subsampled should be subsampled and interpolated to the original grid size. The superscript ${}^{2\uparrow}$ indicates an enlargement of the image with a factor 2, using the interpolation process described in Appendix 4.A. The image is now partitioned in three images, bp_1 , bp_2 and ${}^{2\downarrow}lp_2$, whose sum exactly reconstructs the original. Iterating this procedure gives the subsequent bandpass images.

In the resulting bandpass pyramid the two levels with highest frequency content are represented on the same spatial sampling grid as the original image. The subsequent levels are subsampled by a factor of two in each dimension, for each level. The result after the transform is that the image is represented using about $\frac{7}{3}$ of the number of pixels in the original¹ image. Typically one highpass level, five bandpass levels, and one DC level have been used in the implementation.

6.1.3 Estimation of Orientation

The local orientation is estimated on each level in the bandpass pyramid. The estimation is performed by applying the same filter set on each level. Using an octave based pyramid means that the effective center frequency of the filter set will be reduced by a factor of two for each level. The resulting pyramid is termed an orientation pyramid, where the representation for each pixel is a 2x2 tensor according to Eq. 5.4.

6.1.4 Consistency of Orientation

There are two common ways of separating the signal from the noise: spectral methods, e.g. edge focusing [12, 21], and spatial correlation methods. A method of the latter type was utilized in Chapter 5 where, on a single scale, the orientation estimate was relaxed with respect to its neighbors. In a multi-scale environment it is natural not only to do spatial consistency of orientation, but also scale consistency. The reason is that it is believed that visually interesting features must exist over several scales. It is straightforward to implement the relaxation as tensor averaging in the same sense as described in Chapter 5.

¹One of the attractions of bandpass, and Laplacian, pyramids has been the coding potential inherent in this type of structure, [90]. However, the coding aspect is not the subject of the proposed algorithm.

Implementation

The extension of the averaging procedure compared to the procedure in Chapter 5 is that the average filter is three-dimensional, with scale as the third dimension. In the filter design a lot of heuristics have been used. The exact shape of the filter has proven to be non-critical, but some general design criteria can be given:

1. The filter should be relatively narrow in the scale direction, and only the current and lower resolution images should be used.
2. The filter should have roughly the same region of spatial support for each scale level involved.
3. A smooth function should be used.

The first point can be expounded by considering how hard it is to distinguish an image signal from noise, especially in the high frequency range. The support for visually interesting features should therefore come from lower resolution, where the separation between signal and noise is likely to be better. The second and third point is built on heuristics and it has proven to work satisfactorily in all experiments.

The implemented filter has a depth of 3, i.e. three different scale levels are regarded in the relaxation. The filter planes, f_p , are defined according to

$$\begin{aligned} f_1(\mathbf{x}) &= \frac{1}{2} \frac{\pi}{16} e^{-\pi^2 x^2 / 16} = \frac{1}{2} \text{Gaussian}(\sigma = \frac{2\sqrt{2}}{\pi}) \quad \text{on a 9x9 grid} \\ f_2(\mathbf{x}) &= \frac{1}{3} \frac{\pi}{32} e^{-\pi^2 x^2 / 32} = \frac{1}{3} \text{Gaussian}(\sigma = \frac{4}{\pi}) \quad \text{on a 7x7 grid} \\ f_3(\mathbf{x}) &= \frac{1}{6} \frac{\pi}{64} e^{-\pi^2 x^2 / 64} = \frac{1}{6} \text{Gaussian}(\sigma = \frac{4\sqrt{2}}{\pi}) \quad \text{on a 5x5 grid} \end{aligned} \quad (6.5)$$

This filter is applied after the estimate images have been resampled to the finest resolution image size involved in the filtering.

Note that this filtering does not change the sensitivity to high frequency components in the original data. It only means that rapid changes in the representation of local structure are suppressed. (The original estimate was made by a filter set of size 15x15.)

6.1.5 Determination of Parameters

So far, the estimation of the local models have only incorporated linear operations. But in order to interpret the data a non-linear function is useful. The main reason is that different Signal-to-Noise Ratios (SNR) require different interpretations of the orientation estimates.

The most important parameter to be estimated is the value of the parameter called **a** in Figure 6.4 and Appendix 6.B. The estimation of **a** in the transfer function, $s(E)$, for the local energy is based on global statistics of the relaxed energy. On each scale level the pixels with lowest energy will control the position of **a**.

The strategy is based on the hypothesis of a uniform noise level over the image, and that at least 1% of the signal area is assumed to be flat, i.e. have a signal energy

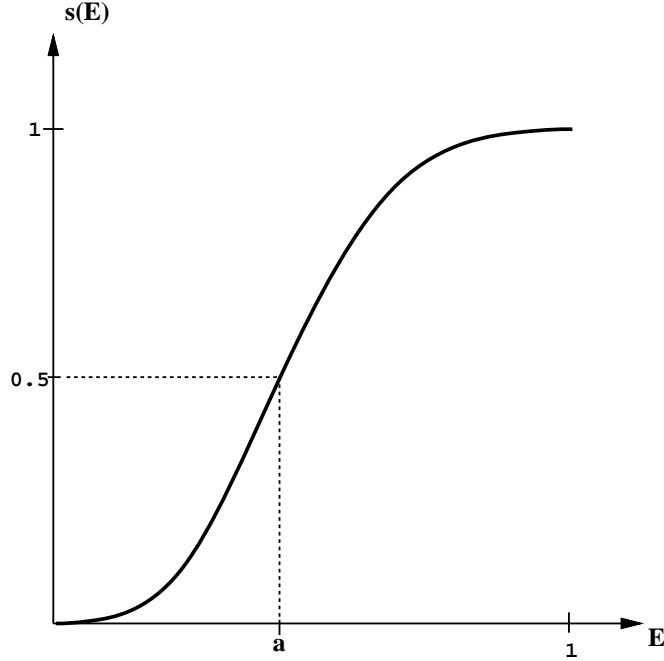


Figure 6.4: *Illustration of the transfer function $s(E)$, see also Appendix 6.B.*

which is less than the noise energy. An estimation of the noise energy is achieved by investigating the energy limit for these pixels. This assumption is supported by the examination of a test on the “boat” image, Figure 6.8, with additive white noise. The energy histograms from one single scale level are displayed in Figure 6.5. The behavior is, as expected, that the pixels with the lowest energy will have higher energy relative to the maximum in the images with low SNR.

To determine \mathbf{a} , a contemplation of the behavior of the noise energy compared with the signal energy in the bandpass transform is useful. Assuming that the noise is white, it follows that the noise energy in the bandpass pyramid increases approximately as ω_l^2 , where ω_l is the center frequency for bandpass level l . As white noise has a flat energy spectrum, the noise energy in the bandpass levels will only depend on their area in the frequency domain. Using the proposed bandpass pyramid leads to a transform with constant relative bandwidth, B , and the area of each bandpass level, Figure 6.6, is proportional to

$$\text{Area} \propto \pi((\omega_l + B\omega_l)^2 - (\omega_l - B\omega_l)^2) = 4B\pi\omega_l^2$$

A more thorough calculation of the energy is given in Appendix 6.A.

On the other hand, stochastically distributed lines have a energy spectrum decreasing as $1/\omega$ and stochastically distributed edges have a spectrum decreasing as $1/\omega^3$, [67]. Taking $1/\omega^2$ as the mean for the sum of these stochastic distributions will then lead to an approximately flat distribution in the bandpass transform.

These considerations taken together makes the use of a quadratic weighting of the scale level, l , from the estimated noise energy, based on the histogram, to the setting of \mathbf{a} adequate. Typically the 1% value from the histogram will be weighted

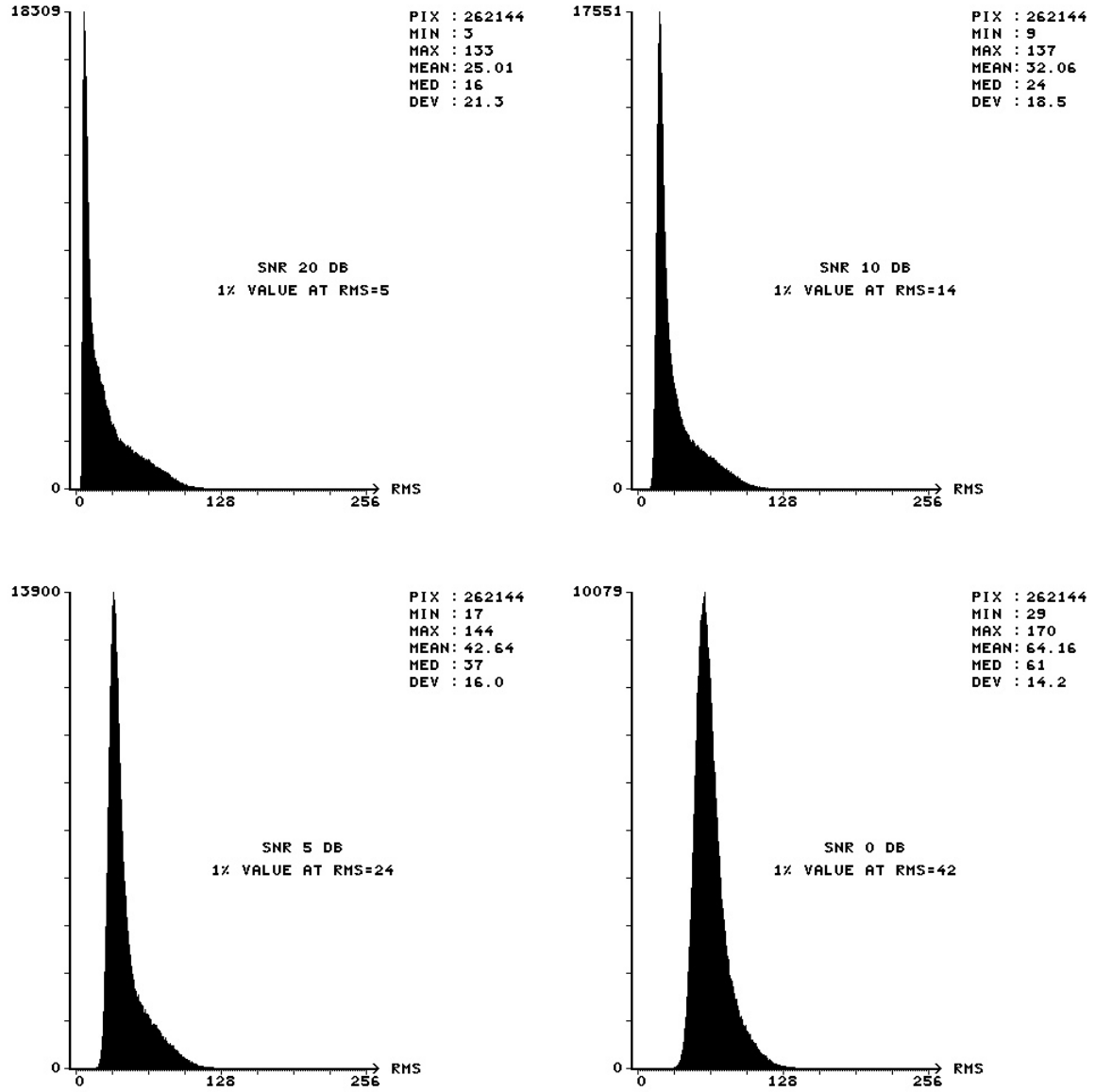


Figure 6.5: Four histograms of the estimated RMS value of energy for the "boat" image with different signal-to-noise ratios on a single scale. The value of $RMS_{1\%}$ is given in each histogram. The histograms are as expected shifted to the right, meaning toward higher energy, if the SNR decreases, while the signal energy is constant.

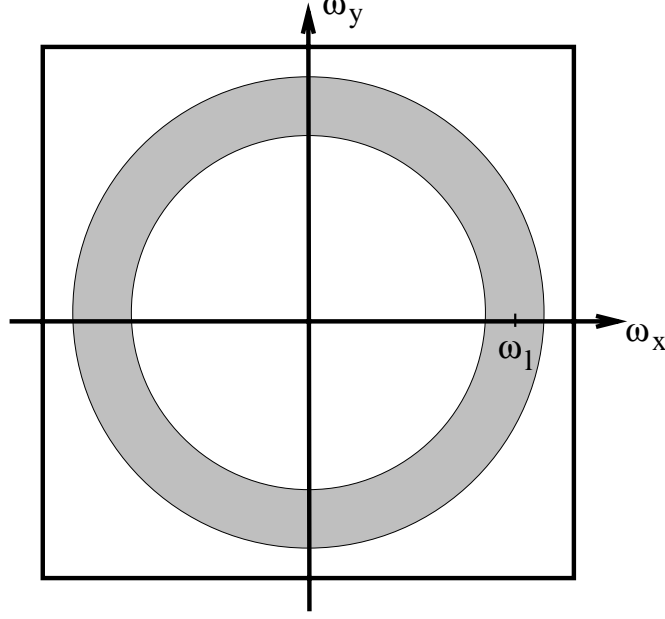


Figure 6.6: *The area of a bandpass filter with center frequency ω_l . The borders illustrate the bandwidth of the filter.*

dependent on the bandpass level l as

$$(c_1 l + c_2)^2 \quad (6.6)$$

where c_1 and c_2 are constants that should be set depending on the application, see for example Eq. 6.7.

In the discussion above the noise is assumed to be white, but if there is another a priori knowledge of the noise distribution or the signal spectrum this is easy to incorporate in the bandpass structure.

Implementation

In this implementation the RMS value is used instead of the energy to keep the numerical bounds in a manageable range. This means that the function suggested in Eq. 6.6 will not be quadratic but linear and it has been implemented as:

$$\mathbf{a} = \left(-\frac{3}{5}l + 4\frac{3}{5}\right)\text{RMS}_{1\%} \quad (6.7)$$

where $\text{RMS}_{1\%}$ is the estimate of the RMS value of the noise from the histogram.

The transfer function, $s(E)$, for outputs in the interval $[0, 1]$ has mainly two degrees of freedom, the half-value, \mathbf{a} , and the slope at \mathbf{a} . The function is symmetric around \mathbf{a} in the sense that it will go to zero as fast as it goes to 1. Now when \mathbf{a} is determined the slope also has to be settled. In this implementation, it has been done by fixating $s(0) = 0$ and using the symmetry of the function.

The drawback of the automatic parameter setting is that the noise level is supposed to be more or less constant over the image. If this is not the case, the

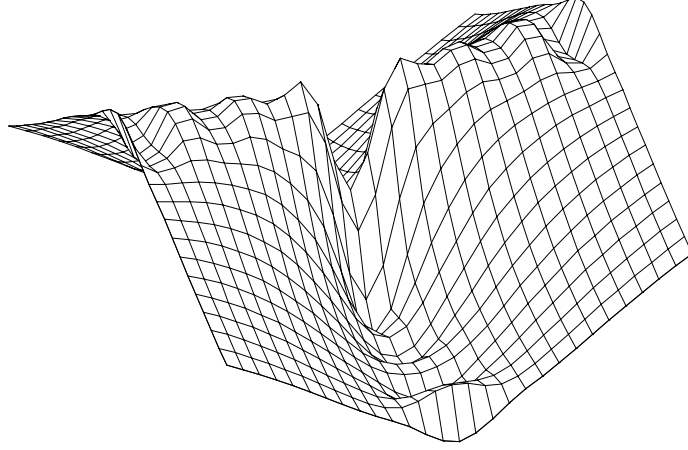


Figure 6.7: A plot in the Fourier domain of one of the used allpass filters with an angular variation of $(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$.

estimation of $\text{RMS}_{1\%}$ must occur separately in different parts of the image. In each of these parts different parameters will control $s(E)$. However, in most situations a uniform noise distribution is a useful model.

6.1.6 Adaptive Filtering

The adaptive filtering from Section 5 is applied on each level of the bandpass pyramid. The contextual control parameters are obtained by passing the tensor averaged data through the transfer function, $s(E)$.

When the adaptive filtering is applied on a bandpass image there is no need to incorporate a lowpass filter as described in Section 5.2, since the DC-component is zero anyway. Another set of filters is then more appropriate, from a complexity as well as a theoretical point of view. The filters used are as before, see Section 5, three polar separable filters with angular function $(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$, where $\hat{\mathbf{n}}_{\mathbf{k}}$ is the main direction of each filter. The bandpass function of the filters can be adapted to the bandpass environment where it is applied.

The ideal bandpass function is an allpass function, but this is not possible to combine with the proposed angular function. However, it is possible to optimize a filter from these specifications. The optimization procedure, [67], is the same as described in Appendix 3.A, i.e. the weighted error is minimized in a least square sense. The weighting function is in this case the bandpass function from the pyramid generation. The result is a filter that optimally approximates an allpass function in the window of the current level, see Figure 6.7. The filter is realized on a 15x15 spatial grid.

The weighting functions of how the anisotropic part of the adaptive filter should

Figure 6.8: *The used image, a circle in the lower right part is indicating where the filter in Figure 6.9 is taken.*

depend of the estimates are chosen in accordance with Eq. 5.13. The smooth decision on how to apply the adaptive filter is based on the estimated local energy according to the transfer function $s(E)$ and setting of the parameter \mathbf{a} .

6.1.7 Reconstruction

As mentioned above the reconstruction of the original image from the bandpass pyramid is simple due to the construction of the pyramid. The sum of the filtered bandpass images will produce a filtered original image.

To show the degrees of freedom in this filtering strategy, the filter around one pixel in the original image, indicated by a circle in the lower right part of Figure 6.8, is shown in Figure 6.9. The filter is given in the Fourier domain. In the center of the image the DC-component is located, and in the low frequency area it is possible to see that a vertical structure is detected, originating from the tree-trunk, and the filter will adapt its orientation to this structure. At higher frequencies the filter adapts to a horizontal structure originating from the tree-branch.

To do edge (or line) enhancement is easy in this adaptive filter environment. On each scale level, the local model provides information of the reliability of the interesting features. High frequency amplification in edge areas can then simply be done by putting higher weights to these bandpass levels in the reconstruction.

The amplification of high frequency bands is inspired by the human visual system where the same behavior is detected, e.g. [30]. The amplification function has been

Figure 6.9: *An example in the Fourier domain of an adapted filter.*

chosen to be exponential with respect to a logarithmic frequency axis, i.e. linear on a log-log plot of spatial frequency versus amplification.

Implementation

The reconstruction of the original image can be seen as the reverse of the generation of the pyramid. The filtered original is constructed by successive enlargements of the filtered bandpass images followed by a summation. The original is then:

$$\begin{aligned} org &= bp_1 + bp_2 + {}^{2\uparrow}BP_{l>2} \\ \text{where} & \\ BP_{l>k} &= bp_k + {}^{2\uparrow}BP_{l>k-1} \end{aligned} \tag{6.8}$$

Amplification of high frequencies on the used logarithmic scale was done according to the following function in the experiments

$$bp_l = bp_l e^{\alpha(6-l)} \tag{6.9}$$

where the bandpass pyramid has six levels and α is a constant controlling the amount of high frequency amplification. Typical values for α is in the interval $[0, 0.2]$. This means that the lowest frequency band will be unaffected and the highest frequency band will be amplified by a factor of $e^{6\alpha}$. Note, this amplification takes place after the filtering, i.e it will occur in areas where the signal energy has high frequency content and noise in flat areas will consequently not be amplified.

6.2 Results

Extensive testing of the algorithm has been carried out on both real and synthetic images. Some of these results are given below. The images used in this section include a synthetic test image, see Figure 6.10, and the “boat” image from the USC image database. The images have been corrupted with white additive Gaussian noise to SNR down to -10dB. In the case of the “boat” image the performance is compared to the algorithm reported in [66], see Figure 6.16. Other images are a mammogram, i.e. a X-ray image of a female breast in Figures 6.20–6.21, and a satellite image over Iceland, Figures 6.22–6.23. It is believed that the result images show the possibilities of the proposed scale and orientation adaptive filtering strategy without further comments.

The synthetic test image is designed to include all possible orientations and a wide frequency range in order to test the sensitivity of the algorithm for these different aspects. The image contains a frequency range that spans approximately 6 octaves, including the Nyquist frequency. As the pattern is circular the orientations are equally distributed in each frequency band. The function generating the test pattern is given by:

$$\begin{cases} f(r) & 0 \leq r < 0.992 \\ W(r)f(r) & 0.992 \leq r < 1 \\ 0.5(1 - W(r)) & r > 1 \end{cases} \quad (6.10)$$

where

$$\begin{aligned} f(r) &= 0.5 * [1 + \cos(0.64\pi(e^{6r} - 1))] \\ W(r) &= \begin{cases} 1 & \text{if } r < 0.992 \\ \cos^2(64\pi(r - 0.992)) & \text{if } 0.992 \leq r < 1 \\ 0 & \text{if } r \geq 1 \end{cases} \end{aligned}$$

and r is the distance from the center in normalized pixel units, e.g. $r = \sqrt{2}$ in the corners of the image.

Measuring the quality of an enhancement or reconstruction algorithm is an extremely complex task. The reason is that there exists no unifying theory of measuring image quality. The problem is aggravated due to the fact that the quality of an image is determined by the not so well understood human visual system (HVS). The most commonly used measure to compare the quality of images is the SNR, i.e. calculating the mean of the squared difference between the original image and the reconstructed image. However, SNR is sensitive to all of the transforms that HVS is insensitive to, that is translation, rotation, contrast stretch and scaling. As a consequence an image that is perceived as a “good quality” image can have very poor SNR due to any of the above transforms.²

For historical reasons SNR tables have been calculated for the synthetic image in Figure 6.10 and for the “boat” image. These are presented in Table 6.1 and Table 6.2.

²An extreme case is when SNR is negative: Then a constant output with a value equal to the correct mean of the signal will have an output SNR that is zero.

Input SNR	Output SNR
∞ dB	26.1 dB
20 dB	22.7 dB
10 dB	12.5 dB
5 dB	8.2 dB
0 dB	5.5 dB
-5 dB	4.0 dB
-10 dB	2.3 dB

Table 6.1: *The result of the scale adaptive enhancement algorithm for the synthetic image in Figure 6.10.*

Input SNR	Output SNR
∞ dB	24.1 dB
20 dB	20.0 dB
10 dB	15.8 dB
5 dB	13.5 dB
0 dB	11.2 dB
-5 dB	8.9 dB
-10 dB	6.0 dB

Table 6.2: *The result of the scale adaptive enhancement algorithm for the “boat” image in Figure 6.14.*

Figure 6.10: *The synthetic test image with white Gaussian noise added to a SNR of 20dB.*

Figure 6.11: *Adaptive filtering of the test image in Figure 6.10, the parameter $\alpha = 0$ in Eq. 6.9.*

Figure 6.12: *Adaptive filtering of the test image Figure 6.10, the parameter $\alpha = 0.1$ in Eq. 6.9.*

Figure 6.13: *Adaptive filtering of the test image Figure 6.10, the parameter $\alpha = 0.2$ in Eq. 6.9.*

Figure 6.14: *The original “boat” image.*

Figure 6.15: *The original “boat” image with white Gaussian noise added, $SNR=0dB$.*

Figure 6.16: *Orientation adaptive filtering of Figure 6.15. This is the result from the algorithm reported in [66] using one single scale.*

Figure 6.17: *Scale and orientation adaptive filtering of Figure 6.15, the parameter $\alpha = 0.0$ in Eq. 6.9.*

Figure 6.18: *Scale and orientation adaptive filtering of Figure 6.15, the parameter $\alpha = 0.1$ in Eq. 6.9.*

Figure 6.19: *Iteration of the scale and orientation adaptive filtering of Figure 6.15, the parameter $\alpha = 0.1$ in Eq. 6.9, for both of the filterings.*

Figure 6.20: *The original mammogram, left: The complete image, right: a part of the image where there are micro calcifications.*

Figure 6.21: *Scale and orientation adaptive filtering of Figure 6.20. The micro calcifications, the small bright dots, are now easier to detect.*

Figure 6.22: *The original satellite image of Iceland.*

Figure 6.23: *Scale and orientation adaptive filtering of Figure 6.22, the parameter $\alpha = 0.2$ in Eq. 6.9.*

Appendices

6.A Calculations of the Energy in the Bandpass Pyramid

To verify that the noise energy in the bandpass filters increases approximately as ω_l^2 , the integral under each filter is calculated. The RMS of the energy in the bandpass pyramid is given by:

$$\begin{aligned} \text{RMS} &= \sqrt{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left(e^{(\omega_x^2 + \omega_y^2)/2\sigma_1^2} - e^{(\omega_x^2 + \omega_y^2)/2\sigma_2^2} \right)^2 d\omega_x d\omega_y} \\ &= \sqrt{\pi \left(\sigma_1^2 (\text{erf}(\pi/\sigma_1))^2 + \sigma_2^2 (\text{erf}(\pi/\sigma_2))^2 - 4 \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \left(\text{erf} \left(\pi \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1^2 \sigma_2^2}} \right) \right)^2 \right)} \end{aligned}$$

where erf is the error function defined as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

The first bandpass level, bp_1 , must be calculated in a somewhat different manner since it is constructed as a difference between a flat filter and a Gaussian.

$$\begin{aligned} \text{RMS} &= \sqrt{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left(1 - e^{(\omega_x^2 + \omega_y^2)/2\sigma_1^2} \right)^2 d\omega_x d\omega_y} \\ &= \sqrt{4\pi^2 + \pi\sigma_1^2 (\text{erf}(\pi/\sigma_1))^2 - 4\pi\sigma_1^2 \left(\text{erf} \left(\frac{\pi}{\sqrt{2}\sigma_1} \right) \right)^2} \end{aligned}$$

The RMS of the energy, if the signal is white, is for each bandpass level given in Table 6.3. The subsampling and resampling procedure, Eq 6.4, has been considered as a filtering with a Gaussian, see Appendix 4.A. The model of the process is specified by the used standard deviations, SDEV, for the Gaussians

Bandpass level	SDEV for the large Gaussian	SDEV for the small Gaussian
bp_1	0	$\frac{\pi}{2}$
bp_2	$\frac{\pi}{2}$	$\frac{\pi}{6}$
bp_3	$\frac{\pi}{\sqrt{20}}$	$\frac{\pi}{\sqrt{148}}$
bp_4	$\frac{\pi}{\sqrt{84}}$	$\frac{\pi}{\sqrt{596}}$
bp_5	$\frac{\pi}{\sqrt{340}}$	$\frac{\pi}{\sqrt{2388}}$
bp_6	$\frac{\pi}{\sqrt{1364}}$	$\frac{\pi}{\sqrt{9556}}$

As the bandpass levels are approximately octave based, this shows that the two upper levels fit into the scheme. The subsampling has been considered as a Gaussian lowpass filtering in accordance with the description in Appendix 4.A.

Bandpass level	RMS	Quotient
BP_1	4.35	1.86
BP_2	2.33	2.31
BP_3	1.01	2.07
BP_4	0.49	2.02
BP_5	0.24	2.00
BP_6	0.12	

Table 6.3: *The RMS of the energy for each bandpass level, and the quotient of the RMS between adjacent levels.*

6.B The Transfer Function

The implemented transfer function, $s(E)$, is defined according to the following equation, where \mathbf{a} is the center of the interval according to Figure 6.4

$$s(E) = \begin{cases} \frac{1}{2} \left(\frac{E}{\mathbf{a}} \right)^2 & 0 \leq E < \mathbf{a} \\ 1 - \frac{1}{2} \left(2 - \frac{E}{\mathbf{a}} \right)^2 & \mathbf{a} \leq E < 2\mathbf{a} \\ 1 & E \geq 2\mathbf{a} \end{cases}$$

Part III

3D — Adaptive Filtering

Chapter 7

Orientation Representation and Estimation in 3D

7.1 Orientation Representation

The fast progress in the data acquisition area has led to a situation where the use of 3D data is commonplace. Typical examples here are video sequences, ultrasonic sequences, MR and CT data. There is therefore a strong need for methods that can handle this type of data efficiently.

The local orientation estimates are in 3D, as well as in 2D, represented with tensors, [59]. This representation fulfills the three requirements from [68] also were given in Section 5.1. Here the requirements are just repeated:

- **Uniqueness**, meaning a one-to-one mapping of orientations reflecting the fact that planes are modulo π .
- **Uniform stretch**, meaning equal sensitivity to changes in orientation for all orientations.
- **Polar separability**, meaning that the estimate should be independent of the strength of the signal.

A representation fulfilling these requirements will make ordinary operations, e.g. averaging and differentiation, in the representation space meaningful. This is often not the case for other representations, see [60, 59] for a more complete discussion. This representation supplies a richer description of the local neighbourhood compared to representing local orientation with the gradient vector. The tensor representation of the local orientation of a neighbourhood with one single orientation is given by

$$\mathbf{T} = \frac{1}{x} \begin{pmatrix} x_1^2 & x_1x_2 & x_1x_3 \\ x_1x_2 & x_2^2 & x_2x_3 \\ x_1x_3 & x_2x_3 & x_3^2 \end{pmatrix} \quad (7.1)$$

where $\mathbf{x} = (x_1, x_2, x_3)$ is directed as the normal vector to the plane of the neighbourhood and $x = \sqrt{x_1^2 + x_2^2 + x_3^2}$. The magnitude of \mathbf{x} is determined by the local energy distribution.

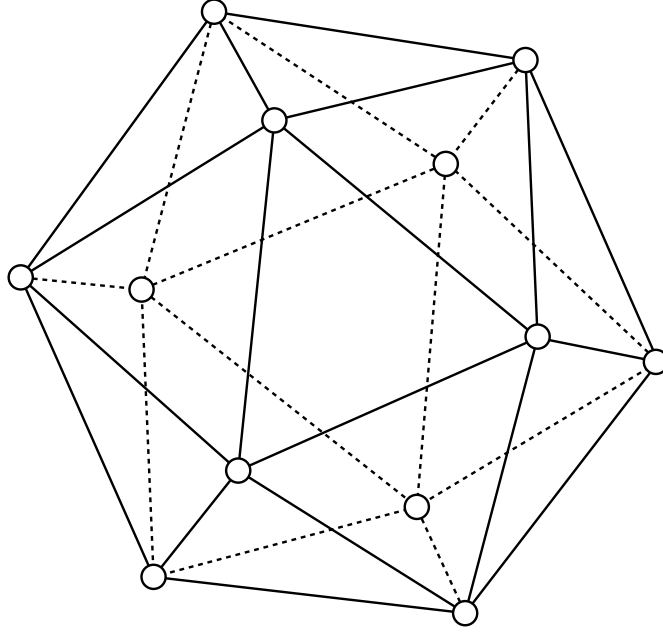


Figure 7.1: *An icosahedron (one of the 5 Platonic polyhedra).*

7.1.1 Implementation of the Orientation Algorithm

The orientation estimation algorithm requires a number of precomputed quadrature filters evenly spread in one half of the Fourier space. The number of filters in 3D must be greater than four, [59, 60]. Consider for example the case of four evenly spread filters, whose axes pass through the vertices of a cube centered around the origin in the Fourier domain. Then let the input to these filters have an energy contribution on a line in the Fourier domain and let this line passes through the center of two opposing faces of the cube. In this case all the filters will have the same output and the filter set cannot discern which pair of faces the line passes through. Thus the minimum number of filter must be greater than four. Spreading the filters evenly puts additional demands on the number of filters, in 3D the only possible numbers are 3, 4, 6 and 10. Hence the minimum number of quadrature filters required for orientation estimation in 3D is 6, where the filters are directed as the vertices of a hemiicosahedron, see Figure 7.1:

$$\begin{aligned}
 \hat{\mathbf{n}}_1 &= c \begin{pmatrix} a, & 0, & b \end{pmatrix}^t \\
 \hat{\mathbf{n}}_2 &= c \begin{pmatrix} -a, & 0, & b \end{pmatrix}^t \\
 \hat{\mathbf{n}}_3 &= c \begin{pmatrix} b, & a, & 0 \end{pmatrix}^t \\
 \hat{\mathbf{n}}_4 &= c \begin{pmatrix} b, & -a, & 0 \end{pmatrix}^t \\
 \hat{\mathbf{n}}_5 &= c \begin{pmatrix} 0, & b, & a \end{pmatrix}^t \\
 \hat{\mathbf{n}}_6 &= c \begin{pmatrix} 0, & b, & -a \end{pmatrix}^t
 \end{aligned} \tag{7.2}$$

with

$$\begin{aligned}
 a &= 2 \\
 b &= 1 + \sqrt{5} \\
 c &= (10 + 2\sqrt{5})^{-1/2}
 \end{aligned} \tag{7.3}$$

A quadrature filter designed with your favorite bandpass function, e.g. the log-

normal function, Eq 3.16, $F_\omega(\omega)$ is given in the frequency domain by:

$$\begin{cases} F_k(\omega) = F_\omega(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 & \text{if } \omega \cdot \hat{\mathbf{n}}_k > 0 \\ F_k(\omega) = 0 & \text{otherwise} \end{cases} \quad (7.4)$$

The spatial filter coefficients are found by a straightforward 3D-DFT or by use of an optimization technique, Appendix 3.A. The resulting spatial filter is complex-valued. This procedure is used to obtain the six quadrature filters.

It is easy to implement the orientation algorithm with these precomputed filters [59]. The tensor describing the neighbourhood is given by:

$$\mathbf{T}^e = \sum_k q_k (\mathbf{N}_k - \frac{1}{5} \mathbf{I}) \quad (7.5)$$

where q_k again denotes the magnitude of the output from filter k . $\mathbf{N}_k = \hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^t$ denotes the direction of the filter expressed in the tensor representation and \mathbf{I} is the unity tensor. A less compact description of Eq. 7.5 is:

1. Convolve the input data with the six complex-valued filters, i.e. perform twelve scalar convolutions.
2. Compute the magnitude of each complex-valued filter by

$$q_k = \sqrt{q_{ke}^2 + q_{ko}^2}$$

where q_{ke} denotes the filter output of the real part of filter k and q_{ko} denotes the filter output of the imaginary part of filter k .

3. Compute the tensor \mathbf{T}^e by Eq. 7.5, i.e.

$$\mathbf{T}^e = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{12} & T_{22} & T_{23} \\ T_{13} & T_{23} & T_{33} \end{pmatrix}$$

where

$$\begin{aligned} T_{11} &= A(q_1 + q_2) + B(q_3 + q_4) - S \\ T_{22} &= A(q_3 + q_4) + B(q_5 + q_6) - S \\ T_{33} &= A(q_5 + q_6) + B(q_1 + q_2) - S \\ T_{12} &= C(q_3 - q_4) \\ T_{13} &= C(q_1 - q_2) \\ T_{23} &= C(q_5 - q_6) \end{aligned}$$

for

$$\begin{aligned} S &= \frac{1}{5} \sum_{k=1}^6 q_k \\ A &= \frac{4}{10 + 2\sqrt{5}} \\ B &= \frac{6 + 2\sqrt{5}}{10 + 2\sqrt{5}} \\ C &= \frac{2 + 2\sqrt{5}}{10 + 2\sqrt{5}} \end{aligned}$$

7.1.2 Evaluation of the Representation Tensor

It is shown in [59] that the eigenvector corresponding to the largest eigenvalue of \mathbf{T}^e is the normal vector of the plane best describing the neighbourhood. This implies that an eigenvalue analysis is appropriate for evaluating the tensor. Below the eigenvalue distribution and the corresponding tensor representation are given for three particular cases of \mathbf{T}^e , where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are the eigenvalues in decreasing order, and $\hat{\mathbf{e}}_i$ is the eigenvector corresponding to λ_i .

1. $\lambda_1 > 0; \lambda_2 = \lambda_3 = 0;$
 $\mathbf{T}^e = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^t$

This case corresponds to a neighbourhood that is perfectly *planar*, i.e. is constant on planes in a given orientation. The orientation of the normal vectors to the planes is given by $\hat{\mathbf{e}}_1$.

2. $\lambda_1 = \lambda_2 > 0; \lambda_3 = 0;$
 $\mathbf{T}^e = \lambda_1 (\mathbf{I} - \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^t)$

This case corresponds to a neighbourhood that is constant on *lines*. The orientation of the lines is given by the eigenvector corresponding to the least eigenvalue, $\hat{\mathbf{e}}_3$.

3. $\lambda_1 = \lambda_2 = \lambda_3 > 0;$
 $\mathbf{T}^e = \lambda_1 \mathbf{I}$

This case corresponds to an *isotropic* neighbourhood, meaning that there exists energy in the neighbourhood but no orientation, e.g. in the case of noise.

The eigenvalues and eigenvectors are easily computed with standard methods such as the Jacobi method, e.g. [86]. Note that the spectrum theorem states that all neighborhoods can be expressed as a linear combination of these three cases.

7.1.3 Accuracy of the Orientation Estimate

The performance of the algorithm is evaluated on two synthetic test patterns. The first test pattern is generated to be locally planar with all directions equally represented. The second test pattern is designed to be locally constant on lines evenly distributed in the volume. The patterns reflect case 1 and case 2 from above.

Accuracy of Estimation of Locally Planar Structures

The test pattern is a 64x64x64 volume that contains all possible 3D plane orientations for a wide frequency range. A traveler heading outward from the center and moving in a straight line would experience a sine wave with decreasing frequency.

Four instances of the test pattern were used for the evaluation, one without noise and two with added Gaussian distributed noise, see also Figure 7.2. The volumes with noise have a SNR of 20 dB, 10 dB and 0 dB respectively, with SNR defined as:

$$\text{SNR} = 20 \log \frac{\text{SDEV}(\text{pattern})}{\text{SDEV}(\text{noise})} \quad (7.6)$$

Figure 7.2: From left to right: without noise, 10 dB and 0 dB. From top to bottom: slice 4, 18 and 32.

The estimated orientation tensor \mathbf{T}^e (with tensor elements T_{ij}^e) was compared with the ideal tensor, \mathbf{T}^f (with tensor elements T_{ij}^f), for all points in the volume. The comparison was done with the error estimate:

$$\text{ERR} = \sqrt{\frac{1}{N} \sum \sum_{ij} (\hat{T}_{ij}^e - \hat{T}_{ij}^f)^2} \quad (7.7)$$

where the $\hat{\cdot}$ -notation indicates that the tensor has been normalized in the Frobenius norm, i.e. $\sum_{ij} \hat{T}_{ij}^2 = 1$, and N is the number of points involved in the error calculation. Since this is a synthetic test pattern it is possible to generate the correct tensor field, $\hat{\mathbf{T}}^f$, as

$$\hat{\mathbf{T}}^f = \hat{\mathbf{x}}\hat{\mathbf{x}}^t \quad (7.8)$$

where \mathbf{x} is the coordinate vector. The results obtained with quadrature filters are compared with the results produced by gradient filters. Both quadrature filters and gradient filters are of size 7x7x7. The gradient estimate, $(x_1, x_2, x_3)^t$, is transformed into the tensor shape by scaling the outer product with $1/\sqrt{x_1^2 + x_2^2 + x_3^2}$. This gives a ‘tensor’ with a magnitude comparable with the orientation algorithm and enables the use of Eq. 7.7.

The results are given in Table 7.1. The clear advantage of the algorithm using quadrature filters is what one could expect, since one should be able to acquire a more robust estimate with twelve filters compared to three. The frequency responses of the filters influence the noise suppression, and a more thorough comparison with a variety of frequency functions is needed. An extensive examination is, however, not

SNR	Quadrature	Gradient
∞ dB	0.04	0.10
20 dB	0.06	0.30
10 dB	0.11	0.49
0 dB	0.32	0.61

Table 7.1: *Comparison of quadrature and gradient filters using Eq. 7.7.*

the issue here. The intention is to demonstrate that quadrature filters give accurate estimates of the 3D orientation.

Note that it is possible to average the orientation tensor to get even more accurate estimates. This is a virtue of the representation and does not depend upon the filters used to achieve the initial estimates. Gradient filters in combination with this representation and averaging are for example used in [15, 57].

Accuracy of Estimation of Locally Linear Structures

The maximum number of evenly distributed lines in 3D is 10, corresponding to the space diagonals of a dodecahedron, [22]. These have been generated from Eq. 7.9

$$V = \sum_{k=1}^{10} \left(\frac{\mathbf{x} \cdot \hat{\mathbf{d}}_{\mathbf{k}}}{x} \right)^{2x^2} \quad (7.9)$$

where $V(x_1, x_2, x_3)$ is the intensity in the voxel, \mathbf{x} is the spatial coordinate, $x = \|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$ and $\hat{\mathbf{d}}_{\mathbf{k}}$ is the normalized direction to one vertex of the dodecahedron. In essence this is a squared scalar product between the direction of the line and the direction of the actual position. Directly applying the scalar product will, however, produce a cone instead of the expected line. To compensate for this broadening of the line, the exponent x^2 is necessary. The result from Eq. 7.9 is rendered in Figure 7.3.

The used test pattern has the size 64x64x64. When doing the actual comparison the center part is omitted. The reason is that there is no unique orientation in the center where all lines interfere with each other.

The line volume has been corrupted with white Gaussian distributed noise to different SNR. SNR is in this case given by to Eq. 7.6. In order to get meaningful noise estimates the standard deviations of the pattern and the noise are calculated only in the voxels where the signal exists.

The accuracy of the orientation estimate is calculated as suggested in Eq. 7.7. The correct tensors for each line are derived from the directions, $\hat{\mathbf{d}}_{\mathbf{k}}$ of the lines by the use of the equation in Section 7.1.2, paragraph 2

$$\hat{\mathbf{T}}^f = \frac{1}{\sqrt{2}}(\mathbf{I} - \hat{\mathbf{d}}_{\mathbf{k}}\hat{\mathbf{d}}_{\mathbf{k}}^t) \quad (7.10)$$

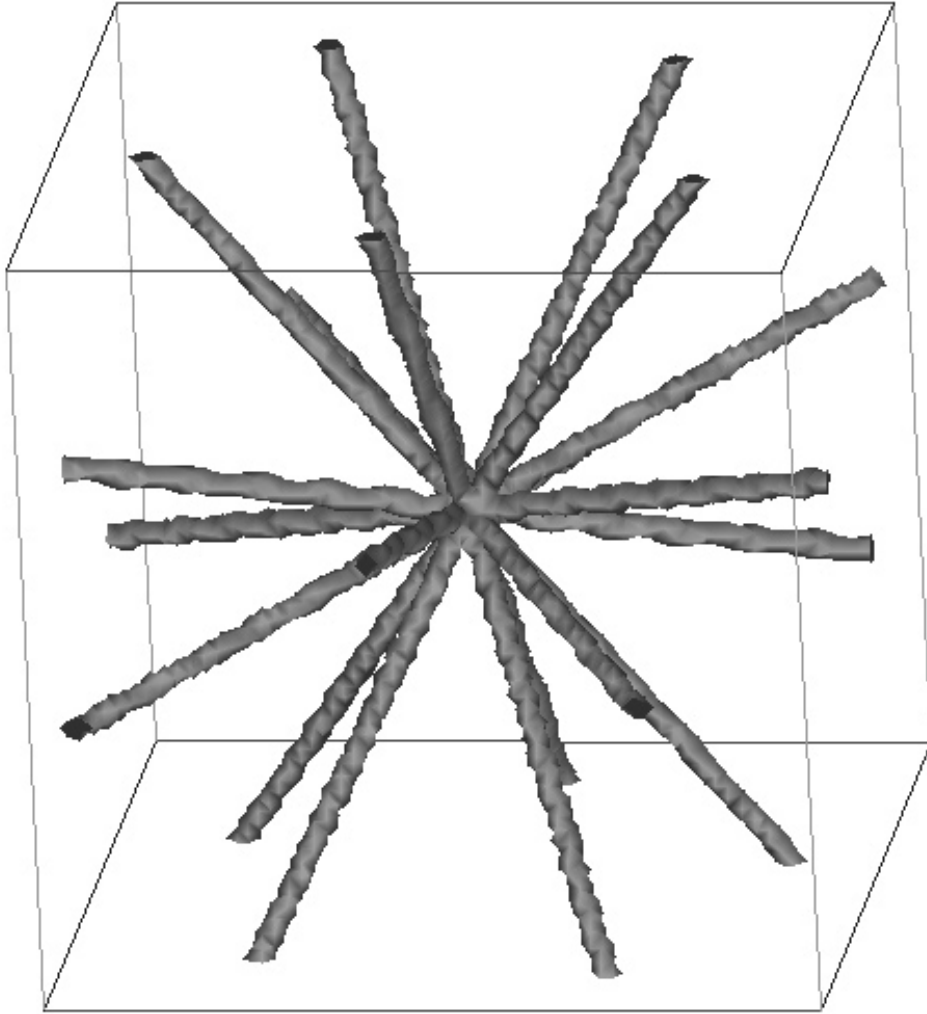


Figure 7.3: *The test volume for the line case. The lines corresponds to the space diagonals of a dodecahedron.*

SNR	Error
∞ dB	0.11
20 dB	0.12
10 dB	0.13
5 dB	0.15
0 dB	0.19

Table 7.2: *The result of the noise sensitivity for line cases according to Eq. 7.7*

and the directions of the lines can be written as

$$\begin{aligned}
\hat{\mathbf{d}}_1 &= k (d, 0, -b)^t \\
\hat{\mathbf{d}}_2 &= k (d, 0, b)^t \\
\hat{\mathbf{d}}_3 &= k (b, d, 0)^t \\
\hat{\mathbf{d}}_4 &= k (-b, d, 0)^t \\
\hat{\mathbf{d}}_5 &= k (0, b, d)^t \\
\hat{\mathbf{d}}_6 &= k (0, -b, d)^t \\
\hat{\mathbf{d}}_7 &= k (f, -f, f)^t \\
\hat{\mathbf{d}}_8 &= k (f, f, f)^t \\
\hat{\mathbf{d}}_9 &= k (-f, f, f)^t \\
\hat{\mathbf{d}}_{10} &= k (-f, -f, f)^t
\end{aligned} \tag{7.11}$$

where

$$\begin{aligned}
d &= a + 2b \\
f &= a + b \\
k &= \frac{1}{\sqrt{3(a+b)}}
\end{aligned}$$

The constants a, b and c are given in Eq. 7.3. This is a symmetric formulation of the vertices of a dodecahedron. Since the structures in the test pattern coincide with the frame slicing it is better if the test volume is tilted with respect to the coordinate system of the filters. Then there will be a larger number of different test cases. Thus the test pattern has been rotated 30° around the x-axis followed by a rotation of 30° around the new y-axis.

The result from the error calculation, Eq. 7.7, for the quadrature filter is given in Table 7.2. One possible explanation for the rather large error for the original signal is the construction of the ideal tensor, $\hat{\mathbf{T}}^{\mathbf{f}}$. Centered on the line $\hat{\mathbf{T}}^{\mathbf{f}}$ from Eq. 7.10 holds, but a few voxels away the shape of tensor should also reflect the variation across the lines, which $\hat{\mathbf{T}}^{\mathbf{f}}$ does not do.

7.2 Optical Flow Estimation

A natural way of estimating optical flow in an image sequence is to estimate 3D-orientation in the sequence, as described above. Orientation in three dimensions (two spatial and one time-dimension) contains information of both the spatial orientation

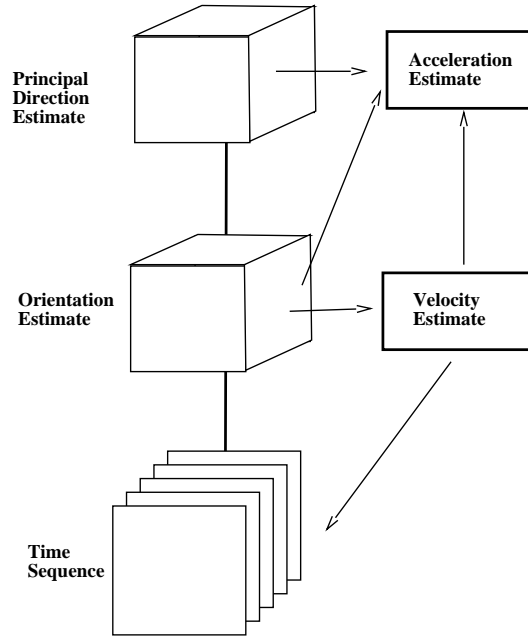


Figure 7.4: *Overview of a hierarchical algorithm.*

and the optical flow. Over the years several other algorithms have been proposed for time sequence analysis. Three different approaches can be distinguished in this field.

- **Matching** 2D image processing techniques are used on one frame at a time to extract feature points, curves, etc. Matching of the feature extraction result is used to find the corresponding parts in the neighbor frames [53, 54, 4].
- **Derivatives** The solution to the optical flow equation is approximated by the use of gradient filters. The filters are 3D-filters (two spatial dimensions + time) [80, 91, 57, 15].
- **Signal Processing** The 3D data set is analyzed with tools such as the Fourier Transform to design algorithms for the estimation of the planes in the spatio-temporal data set originating from moving lines and curves originating from moving points [34, 2, 48].

The borderlines between these three methods are not always clear and it is possible to use algorithms which incorporate aspects of all of the three approaches. This section will not elaborate on the virtues and drawbacks of the different approaches. The methods presented here can be considered to belong to the signal processing approach.

The algorithm introduced here supplies estimates of velocity and can be extended with acceleration estimates [9, 8]. Fig. 7.4 shows possible algorithm components to indicate the hierarchical structure where the optical flow estimates are supposed to function.

Figure 7.4 presents a general framework for the analysis of time sequences. Recent developments in the human visual system models suggest the existence of frequency

and orientation channels representing the local spatial, as well as the local spatio-temporal image spectrum, [55, 3]. In the latter case the energy concentration in the local spectrum to a particular orientation and frequency channel means a velocity vector with a particular direction and coarseness range. Adelson and Bergen [2] as well as Watson and Ahumada [92] and others have proposed human visual motion sensing models based on the local spectra by means of the spatio-temporal filter responses using separable filters of Gabor type.

7.2.1 Velocity Estimation

If the signal to analyze is a time sequence, a plane means a moving line and the. The optical flow will be obtained by an eigenvalue analysis of the estimated representation tensor. The projection of the eigenvector corresponding to the largest eigenvalue onto the image plane will give the flow field. However, the so-called aperture problem will give rise to an unspecified velocity component, the component moving along the line. The aperture problem is a problem in all optical flow algorithms which rely on local operators. On the other hand, the aperture problem does not exist for moving points in the sequence. In this case of velocity estimation the correspondence between the energy in the spatial dimensions and the time dimension is established to get correct velocity estimation.

By examining the relations between the eigenvalues in the orientation tensor it is possible to divide the optical flow estimation into different categories, [9, 47]. Depending on the category different strategies can be chosen, see Section 7.1.2. Case number two in Section 7.1.2, i.e. the line case, gives a correct estimation of the velocity in the image plane and is thus very important in the understanding of the motion. To give an illustration of this the synthetic test sequence, Figure 7.5¹ is used as an example. In Figure 7.6 the correct velocity field is given with arrows, white arrows correspond to the plane case and black arrows to the line case.

To do this division of different shapes of the tensor the following functions are chosen

$$p_{plane} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (7.12)$$

$$p_{line} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (7.13)$$

$$p_{iso} = \frac{\lambda_3}{\lambda_1} \quad (7.14)$$

These expressions can be seen as the probability for each case. In Figure 7.7 the division is made by selecting the case having the highest probability.

The calculation of the optical flow is done using Eq. 7.15 for the plane case and Eq. 7.16 for the line case. In neighborhoods classified as ‘isotropic’ no optical flow is computed. The ‘true’ optical flow in neighborhoods of the ‘plane’ type, such

¹A rotating and translating star together with a fair amount of Gaussian noise. The star is rotated 1.8° counter-clockwise around its center, and translates 0.5 pixel up and 1 pixel to the right between each frame.

Figure 7.5: *One frame from the original sequence of the translating and rotating star, with white Gaussian noise added.*

as moving lines, cannot be computed by optical flow algorithms using only local neighbourhood operations as mentioned earlier. The optical flow is computed by

$$\begin{aligned}\mathbf{x} &= \hat{\mathbf{e}}_1 \\ \mathbf{v}_{line} &= (-x_1x_3\hat{\mathbf{x}}_1 - x_2x_3\hat{\mathbf{x}}_2)/(x_1^2 + x_2^2)\end{aligned}\tag{7.15}$$

where $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are the orthogonal unit vectors defining the image plane.

The aperture problem does not exist for neighborhoods of the ‘line’ type, such as moving points. This makes them, as mentioned, very important for motion analysis. The optical flow is computed by

$$\begin{aligned}\mathbf{x} &= \hat{\mathbf{e}}_3 \\ \mathbf{v}_{point} &= (x_1\hat{\mathbf{x}}_1 + x_2\hat{\mathbf{x}}_2)/x_3\end{aligned}\tag{7.16}$$

In Figure 7.8 the result from Eq. 7.15, white arrows, and Eq. 7.16, black arrows, is given.

7.2.2 Spatio-Temporal Channels

The human visual system has difficulties handling high spatial frequencies simultaneously with high temporal frequencies [7, 27]. This means that objects with high velocity cannot be seen sharply without tracking. A possible explanation to this is that the visual system performs an effective data reduction. The data reduction is made in such a way that high spatial frequencies can be handled if the temporal frequency is low, and vice versa. This strategy is possible to use in a computer vision model for time sequences.

The input image sequence is subsampled both spatially and temporally into different channels. In Table 7.3 the data content in the different channels relatively to

Figure 7.6: *The correct velocity vectors from the star sequence. Black vectors correspond to the moving point case and white ones to the moving line case.*

Figure 7.7: *The subdivision in different cases following Eq. 7.12 for the test sequence.*

Figure 7.8: *The result of the optical flow algorithm. Black vectors correspond to the moving point case and white ones to the moving line case.*

a reference sequence, *ch00*, is shown. The reference sequence has maximum resolution in all dimensions; typically this means a video signal of 50 Hz², height 576 and width 720 pixels. The frequency difference between adjacent channels is one octave, i.e. a subsampling factor of 2 is used.

The numbers in Table 7.3 indicate that large data reduction can be made by not using the channels with high resolution in both spatial and temporal domains. For instance, the channels on the diagonal together contain approximately 1/4 of the data in the reference sequence (*ch00*).

As pointed out in Chapter 3 there is also a signal theoretical reason to use a pyramid representation of the image. A single filter has a particular limited pass band, both temporally and spatially, which may or may not be tuned to the different features to describe. In Figure 7.9a the upper cut-off frequency for a spatio-temporal quadrature filter set is indicated. The lower cut-off frequency is not plotted for the sake of clarity. Only the first quadrant in the ω_s, ω_t plane is plotted. The use of this filter set on different subsampled channels corresponds to using filters with different center frequencies and constant relative bandwidth. Figure 7.9b indicates the upper cut-off frequency when convolving the channels on the diagonal in Table 7.3 with this filter set.

To avoid aliasing in the subsampling the sequence must be prefiltered with a lowpass filter. As the resulting channel shall be further processed the design of the lowpass filter is critical. The estimation of optical flow from Eq. 7.15 and Eq. 7.16 utilize the relationship of energies originating from spatial variations and from temporal variations. The lowpass filter used for anti-aliasing should then *not* influence this relationship.

Take *ch10* as an example. This channel is subsampled a factor of two in the

²A strategy to filtering interlaced sequences is described in Appendix 7.A.

		Spatial subsampling				
		1/8	1/4	1/2	1	
Relative data content		1/64	1/16	1/4	1	1
		1/128	1/32	1/8	1/2	1/2
		1/256	1/64	1/16	1/4	1/4
		1/512	1/128	1/32	1/8	1/8

		Spatial subsampling				
		1/8	1/4	1/2	1	
Notation for sequence		ch30	ch20	ch10	ch00	1
		ch31	ch21	ch11	ch01	1/2
		ch32	ch22	ch12	ch02	1/4
		ch33	ch23	ch13	ch03	1/8

Table 7.3: *Data content and name convention for the different spatio-temporal channels.*

spatial dimension and not subsampled in the time dimension. A natural way of obtaining this channel is to apply a spatial Gaussian, as described in Appendix 4.A, prior to the subsampling. However, by such a filtering the relationship between spatio-temporal domain is altered. It is obvious from Figure 7.10 where a stylized Gaussian is plotted, that the energy in the subsampled spatial domain is reduced due to the filtering.

The way to design the anti-aliasing filter to obtaining *ch10* is as an elliptic spatio-temporal function, e.g. an elliptic Gaussian

$$G(\omega_s, \omega_t) = e^{-0.5((\omega_s/\sigma_s)^2 + (\omega_t/\sigma_t)^2)} \quad (7.17)$$

where $\sigma_s = \pi/4$ and $\sigma_t = \pi/2$. Then is not the relations in the spatio-temporal affected. The reasoning is the same for designing the anti-aliasing filters for other channels. The design is reduced to set the values of σ_s and σ_t .

7.2.3 Quantitative Error Measurements

The estimation error is computed in two different ways; first following an error measure from [34] given in Eq. 7.18, second by a histogram, where the correct magnitude of the image flow has been quantized in 100 different velocities. In each of these velocities the relative mean error is calculated. The result is that the sensitivity with respect to the speed of the used filter set can be determined. The latter method will be used to examine the performance of the normal flow calculation in different spatio-temporal channels.

In order to make comparisons with other algorithms, the same angular measure of error is used as in [11, 34]. In [11] an extensive testing of several optical flow algorithms is performed using the following error measure between the correct vector $\hat{\mathbf{o}}^f$ and an estimate $\hat{\mathbf{o}}^e$

$$\psi_E = \arccos(\hat{\mathbf{o}}^e \cdot \hat{\mathbf{o}}^f) \quad (7.18)$$

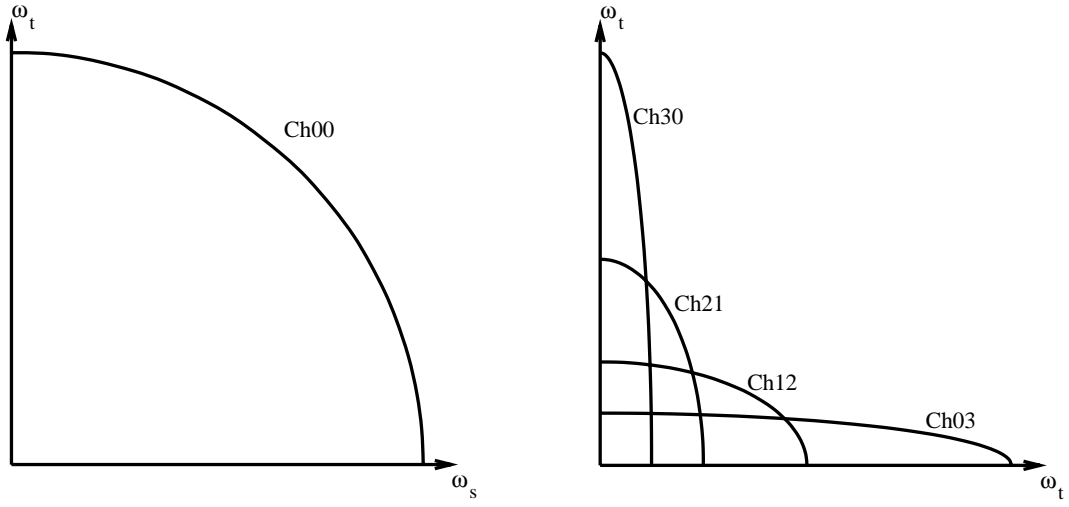


Figure 7.9: *Cut-off frequency for a spatio-temporal filter.*

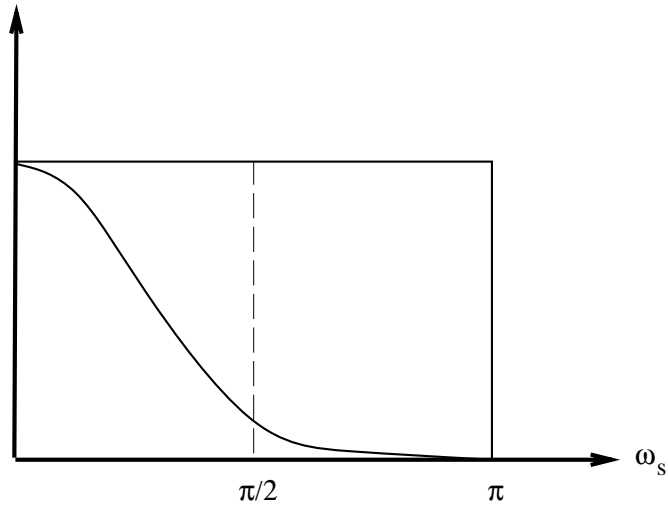


Figure 7.10: *Illustration of a Gaussian filtering, a subsampling will move the frequency $\pi/2$ to the maximum frequency π .*

where the optical flow vector $\mathbf{v} = (v_1, v_2)$ is expressed as a 3D-orientation vector

$$\hat{\mathbf{o}} \equiv \frac{(v_1, v_2, st)^t}{\sqrt{v_1^2 + v_2^2 + st^2}} \quad (7.19)$$

where $st = 1$ in [34, 11].

This error measure can be further expounded by considering the behavior if the absolute error, Δv , is small, see Figure 7.11. Differentiating the angle α in

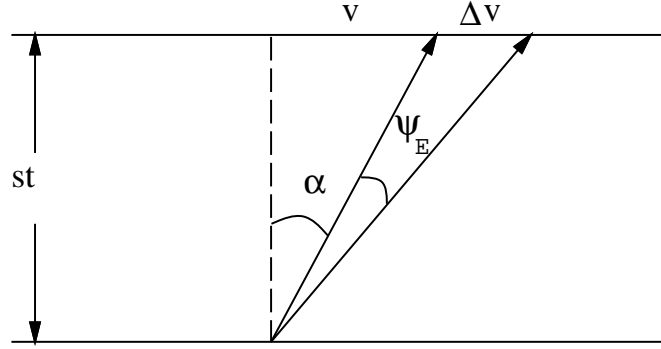


Figure 7.11: *Illustration of the angular error suggested in [34].*

Figure 7.11 w.r.t. the absolute error will give the correspondence to ψ_E .

$$\Delta v = \sqrt{(v_1^e - v_1^f)^2 + (v_2^e - v_2^f)^2} \quad (7.20)$$

$$\alpha = \arctan\left(\frac{v}{st}\right) \quad (7.21)$$

$$\psi_E = \Delta\alpha = \frac{st}{st^2 + v^2} \Delta v \quad (7.22)$$

This angular measure handles small velocities without the amplification inherent in the commonly used relative measure of vector differences.

Results

The error estimates by applying Eq. 7.18 is calculated for two of the sequences in the examination in [10], i.e. the translating tree and the divergent tree. One frame from each of the original sequence with the correct velocity fields superimposed can be found in Figure 7.12.

The results of the proposed method, based on eigenvalue decomposition according to Eq. 7.16, depend highly on the way the estimate is averaged in the representation space. The smooth motion fields in the test sequences enable the use of large averaging filters without destroying the information. The values from the calculation of Eq. 7.18 is given in Table 7.4, where three different sizes of a Gaussian low pass filter was used. As can be seen in the density column in Table 7.4 not all pixels in the sequence contributes in the error calculation. This is due to the aperture problem, the pixels that has been used fulfill the following demands

$$||\mathbf{T}||_{frob} \geq 0.1 \quad \text{and} \quad p_{line} \geq 0.5$$

Figure 7.12: Two frames from the tree sequences. To the left the motion field from the translating sequence is superimposed and to the right the motion field from the divergence sequence is superimposed (further information about these sequences can be found in [33]).

Size of averaging filter	Average error	Standard Deviation	Density
7x7x7	1.61°	0.94°	11.3 %
15x15x15	0.98°	0.51°	17.7 %
21x21x21	0.85°	0.42°	21.1 %

Size of averaging filter	Average error	Standard Deviation	Density
7x7x7	2.93°	2.10°	8.5 %
15x15x15	1.93°	1.28°	13.7 %
21x21x21	1.77°	1.14°	17.7 %

Table 7.4: The result of of the error calculation from the velocity estimation algorithm based on eigenvalue decomposition of the representation tensor. In the upper table are the results from the translating tree sequence and in the lower are the results from the diverging tree sequence.

where p_{line} is defined in Eq. 7.12. Other values of these thresholds give different values in the table. In general one can state that the higher density of the estimate the higher the error will be. This indicates that some kind of regularisation, see e.g.[13], using the pixels with highest certainty, might be a useful postprocessing step to get a motion field over the whole image.

To compare the values in Table 7.4 in other algorithms two tables from [11, 10] is given in Tables 7.5-7.6. The names in these tables refer to the authors of the articles from which Barron et al have implemented the algorithms, see [52, 74, 91, 4, 49, 34]. The decrease of the error value in Table 7.4 depending on the size of the averaging filters shows how extremely difficult it is to do a impartial sensitivity examination.

The error estimate for the histogram method is calculated for the test pattern in Figure 7.2, that also has been used in the performance examination of the quadrature filters, see Table 7.1. The correct image velocities, has been calculated for this

Technique	Average error	Standard Deviation	Density
Horn and Schunck	38.99°	27.84°	100 %
Lucas and Kanade ($\lambda_2 \geq 1.0$)	1.75°	1.43°	40.8 %
Lucas and Kanade ($\lambda_2 \geq 5.0$)	1.12°	0.82°	13.6 %
Uras et al. (unthresholded)	0.71°	0.81°	100 %
Uras et al. ($\det(\mathbf{H}) \geq 1.0$)	0.47°	0.29°	41.7 %
Anadan	4.54°	2.98°	100 %
Heeger	4.79°	2.39°	13.8 %
Fleet and Jepson ($\tau = 2.5$)	0.36°	0.41°	76.0 %
Fleet and Jepson ($\tau = 1.25$)	0.23°	0.20°	51.5 %

Table 7.5: *The results from [11, 10] of different algorithms from the translating tree. (The table is reprinted by courtesy of David Fleet.)*

Technique	Average error	Standard Deviation	Density
Horn and Schunck	12.77°	12.00°	100 %
Lucas and Kanade ($\lambda_2 \geq 1.0$)	3.04°	2.53°	49.4 %
Lucas and Kanade ($\lambda_2 \geq 5.0$)	2.32°	1.84°	24.8 %
Uras et al. (unthresholded)	5.11°	3.96°	100 %
Uras et al. ($\det(\mathbf{H}) \geq 1.0$)	4.05°	2.26°	56.5 %
Anadan	8.23°	6.17°	100 %
Heeger	4.95°	3.09°	73.8 %
Fleet and Jepson ($\tau = 2.5$)	1.24°	0.72°	64.3 %
Fleet and Jepson ($\tau = 1.25$)	1.10°	0.53°	49.7 %

Table 7.6: *The results from [11, 10] of different algorithms from the diverging tree sequence. (The table is reprinted by courtesy of David Fleet.)*

pattern. Then has the optical flow, according to Eq. 7.15 been estimated for four different spatio-temporal channels, *ch00*, *ch01*, *ch10* and *ch11* of the test volume. Prior to the estimation the pattern has been corrupted with white Gaussian additive noise to a SNR of 10dB. The correct image velocities are quantized into 100 'boxes' in the interval $[0, 5.5]$ pixels/frame. In each of these 'boxes' is the sum of the absolute velocity errors and the number of voxels involved in the error sum calculated. In Figure 7.13 is the relative mean error for each 'box' plotted.

The behavior for the spatio-temporal channels is that *ch01* has a better performance for low velocities, while *ch10* has better performance for higher velocities. *ch00* and *ch11* has compared to each other approximately the same performance for different velocities. This is no surprise since maximum velocity sensitivity for these channels are

<i>ch00</i>	1 pixel/frame
<i>ch01</i>	0.5 pixel/frame
<i>ch10</i>	2 pixel/frame
<i>ch11</i>	1 pixel/frame

which also is indicated in Figure 7.9.

The histograms in Figure 7.13 show also that a combination of different spatio-temporal channels allows a relative error well below 10 % for a wide range of image velocities. It should then be considered that the input pattern were corrupted with a signal-to-noise ratio of 10dB.

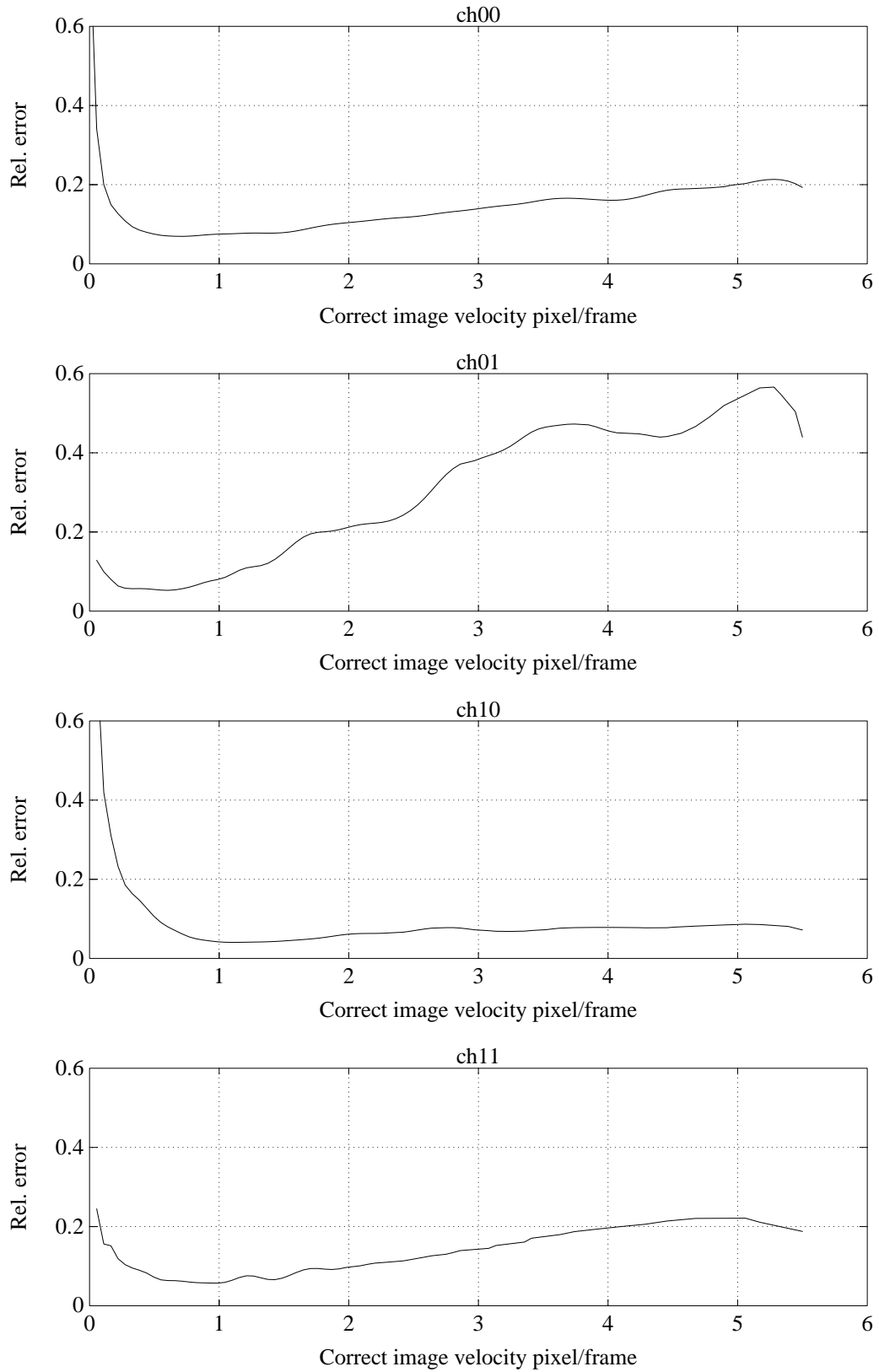


Figure 7.13: *The relative mean error for different image velocities and four different spatio-temporal channels. The results are compiled from estimations of the test pattern in Figure 7.2, the SNR=10dB.*

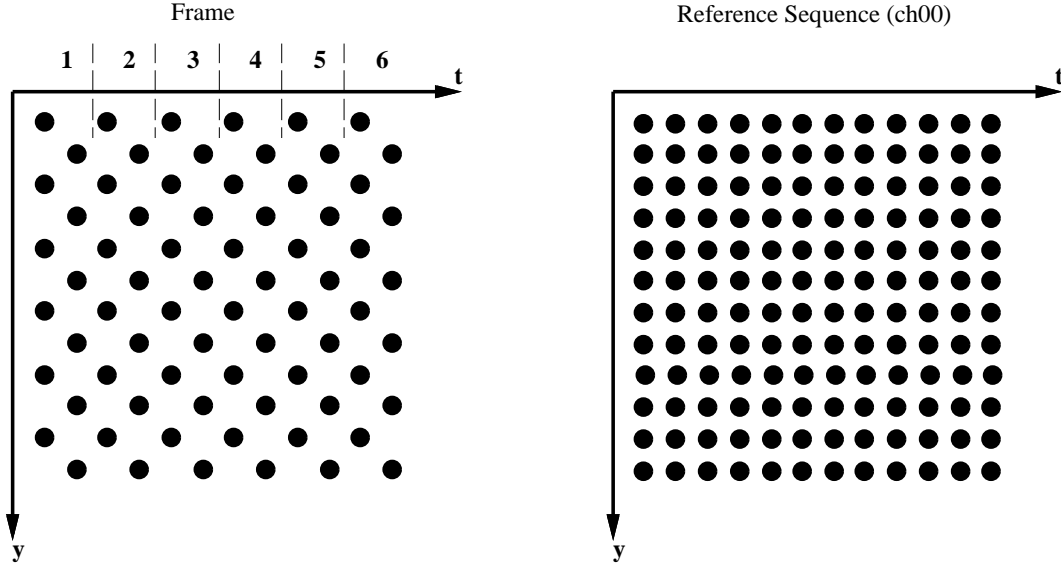


Figure 7.14: *Sampling grid for interlaced and reference sequences respectively.*

Appendix

7.A Filtering of Interlaced Video Signals

Ordinary video equipment produces images where the sampling grid is different from the commonly used grid for digital images. This is due to the interlacing defined in the video standard. In Figure 7.14, the sampling grid for the different lines are plotted against time. In the horizontal direction (x-direction), the sampling density is the same as the density for the t- and y-directions in the reference sequence. It can be seen that the interlaced signal contains half the number of sampling points compared to the reference sequence, *ch00*.

To be able to perform feature extraction by using standard filters, the sequence should consist of non-interlaced images. The easiest way to convert the sequence is to put the odd and the even field from each frame in a separate image. The resulting sequence then gets maximum resolution in time, but the resolution in the y-direction is only half of the maximum resolution. Because of the full resolution in the x-direction, the pixels are not square shaped any longer. Furthermore, the same line in two consecutive fields does not cover the same area in the scene. This gives aliasing if for instance a one pixel wide horizontal line appears in the scene.

A more accurate way is to resample the interlaced sequence so that the different resolution channels can be created. Figure 7.15 shows the frequency domain corresponding to the interlaced and the ordinary sampling grids. The rhomb indicates the Nyquist frequency for the interlaced signal, and the square for the reference sequence. Because the rhomb covers only a subarea of the square, it follows that it is impossible to achieve the resolution of *ch00* from the video signal. Without any aliasing it is possible to get *ch11*, i.e. half the resolution both spatially and temporally. This corresponds to the shaded square inside the rhomb in Figure 7.15b. See also the results in Chapter 9, where a complete reconstruction of *ch00* from the interlaced original is given in Figures 9.10-9.11.

From Figure 7.15a it is clear that the interlaced signal has maximum resolution in the

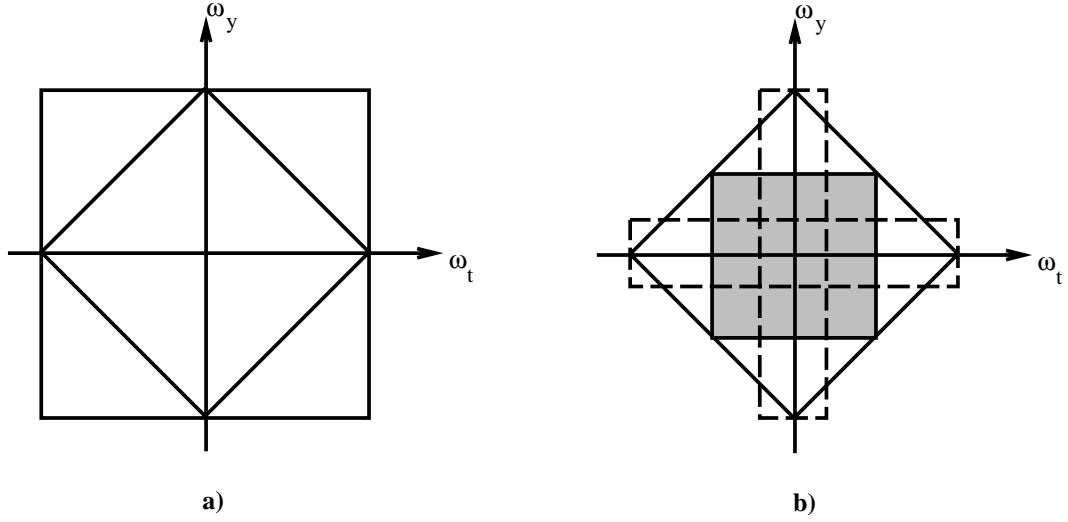


Figure 7.15: *Nyquist frequencies for the two grids.*

y-direction when the temporal frequency is zero, i.e. the scene is static. Furthermore the video signal has maximum temporal resolution when the spatial frequency is zero, that is, when the scene is uniform.

These two cases are quite degenerated, but similar degeneration has been found in the mammal visual system (x- and y-cells) [27], i.e. high spatial frequency can only be detected if the temporal frequency is low and vice versa.

To be able to extract more of the information inside the rhomb in Figure 7.15 and simultaneously avoiding aliasing, two more filters have been implemented. These filters correspond to *ch02* and *ch20*. They are indicated with dashed rectangles in Figure 7.15b.

The frequency function of the filters is a Gaussian, G :

$$G(\omega_x, \omega_y, \omega_t) = e^{(-\frac{1}{2}(\omega_x^2/\sigma_x^2 + \omega_y^2/\sigma_y^2 + \omega_t^2/\sigma_t^2))}$$

The choice of $\sigma_x, \sigma_y, \sigma_t$ and the size of the filters for the different channels are:

$$\begin{array}{llll} \text{ch11 : } & \sigma_x = \pi/4 & \sigma_y = \pi/4 & \sigma_t = \pi/4 \quad 7 \times 7 \times 7 \\ \text{ch20 : } & \sigma_x = \pi/8 & \sigma_y = \pi/8 & \sigma_t = \pi/2 \quad 13 \times 13 \times 4 \\ \text{ch02 : } & \sigma_x = \pi/2 & \sigma_y = \pi/2 & \sigma_t = \pi/8 \quad 4 \times 4 \times 13 \end{array}$$

The filters are designed on an ordinary sampling grid, as in Figure 7.14b, but applied on an interlaced grid, Figure 7.14a. Because of the fact that the grid is sparse, only about one half of the points in the filter are necessary.

Further subsampling is performed on an ordinary grid, and therefore standard filters can be used.

Chapter 8

Applications in Structure from Motion

The purpose of this chapter is to exemplify how the information captured in the 3D orientation tensor is applicable for further processing. Therefore, all the details of the used algorithms will not be presented. Two applications relevant for estimation of structure from motion will be described. First a technique for estimation of the focus of expansion for a translating camera will be presented and second an application of motion stereo is given.

8.1 Extraction of Focus of Expansion

The problem of finding the Focus of Expansion (FOE) for a moving camera is one step in the extraction of the scene geometry from the image flow,[50, 41]. Figure 8.1 presents the geometrical structure of the problem and the notation.

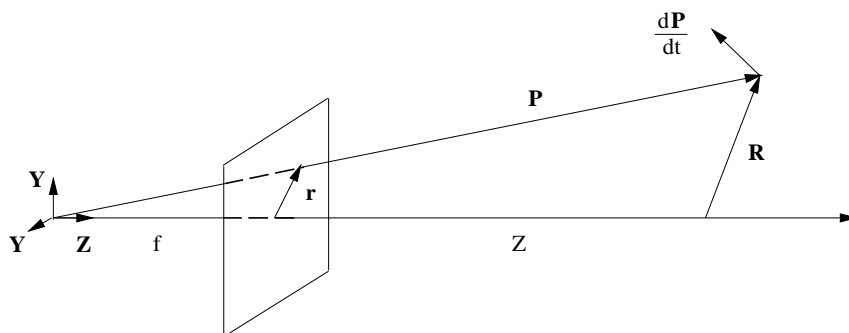


Figure 8.1: *Geometry of the projected motion field for a translating camera.*

Using simple geometry and the notations from Figure 8.1, where f is the focal length and Z is the depth, the following relationship between R , the projection of a translating point, P , in the scene onto the (X, Y) -plane and, r , the projection onto the image plane (x, y) of the same point, P , can be established:

$$\frac{\mathbf{r}}{f} = \frac{\mathbf{R}}{Z} \quad (8.1)$$

Taking the derivative of Eq. 8.1 yields:

$$f \frac{d\mathbf{R}}{dt} = Z \frac{d\mathbf{r}}{dt} + \frac{dZ}{dt} \mathbf{r} \quad (8.2)$$

Denoting the velocity of a point \mathbf{P} in the scene by:

$$\frac{d\mathbf{P}}{dt} = \left(\frac{d\mathbf{R}(X, Y)}{dt}, \frac{dZ}{dt} \right) = (V_x, V_y, V_z)$$

and the velocity of a point \mathbf{r} in the image by:

$$\frac{d\mathbf{r}(x, y)}{dt} = (u, v)$$

Inserting the derivatives of \mathbf{R} , \mathbf{r} and Z in Eq. 8.2 results in

$$f(U, V) = Z(v_x, v_y) + V_z(x, y) \Rightarrow \begin{cases} v_x = \frac{1}{Z}(fV_x - V_zx) \\ v_y = \frac{1}{Z}(fV_y - V_zy) \end{cases} \quad (8.3)$$

For a pure translation of the camera, features in the image move towards or away from a single point in the image, the FOE. This means that the velocity at the FOE is zero. Inserting this condition into Eq. 8.3 gives the image coordinates of the FOE, $\mathbf{r}_0 = (x_0, y_0)^T$:

$$\begin{cases} x_0 = \frac{f V_x}{V_z} \\ y_0 = \frac{f V_y}{V_z} \end{cases} \Rightarrow \frac{Z}{W} = \frac{x_0 - x}{v_x} = \frac{y_0 - y}{v_y} \Rightarrow$$

which implies

$$(x - x_0)v_y - (y - y_0)v_x = 0 \quad (8.4)$$

Using velocity estimates from more than two points not suffering from the aperture problem, i.e. the line case from Section 7.2.1, an over determined linear equation system in x_0 and y_0 formed by N equations on the form of Eq. 8.4 is to be solved.

$$\mathbf{A}\mathbf{r}_0 = \mathbf{b} \quad , \quad \mathbf{A} = \begin{pmatrix} -v_{y1} & v_{x1} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ -v_{yN} & v_{xN} \end{pmatrix} \quad , \quad \mathbf{b} = \begin{pmatrix} -x_1v_{y1} + y_1v_{x1} \\ \cdot \\ \cdot \\ \cdot \\ -x_Nv_{yN} + y_Nv_{xN} \end{pmatrix} \quad (8.5)$$

A least square solution to the Eq. 8.5 can be obtained by finding the pseudo inverse \mathbf{A}^+ to the equation system matrix \mathbf{A} . It can be noted that it is simple to

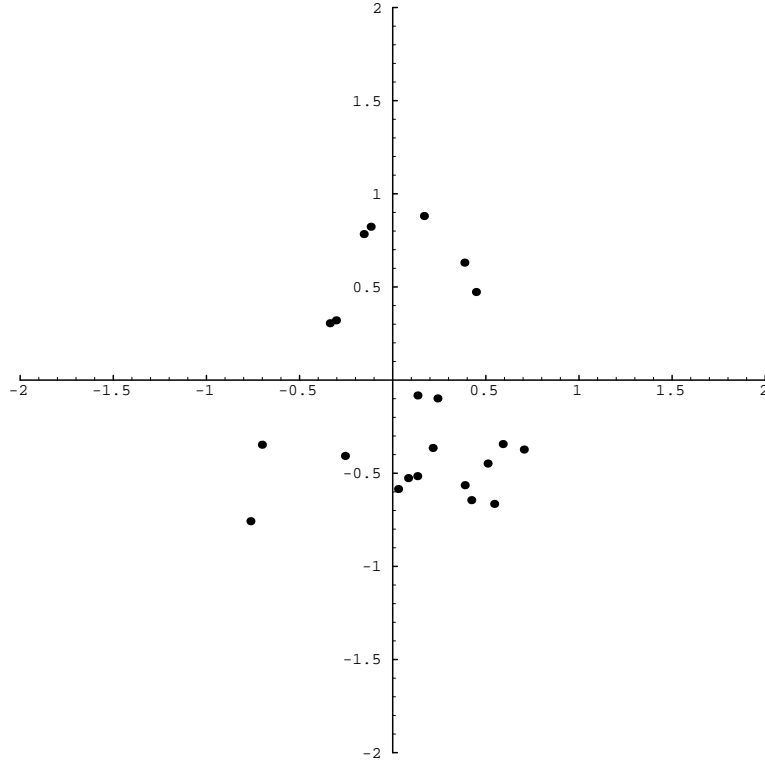


Figure 8.2: *Error in the calculation of FOE, scale in pixels. The results are for 22 frames of the diverging tree sequence, where the original image size is 150x150 pixels.*

incorporate certainties in the solution of the equation system by just multiplying each equation by a certainty factor, [86].

$$\mathbf{A}^+ = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$$

The solution can now be written:

$$\mathbf{r}_0 = \mathbf{A}^+ \mathbf{b} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

An experiment using 22 frames from an image sequence containing a camera translation towards a tilted picture, see Figure 7.12, and about 86 velocity estimates from each frame gave the result shown in Figure 8.2 indicating an error of less than 1%. The correct answer is $\mathbf{r}_0 = (0, 0)^T$.

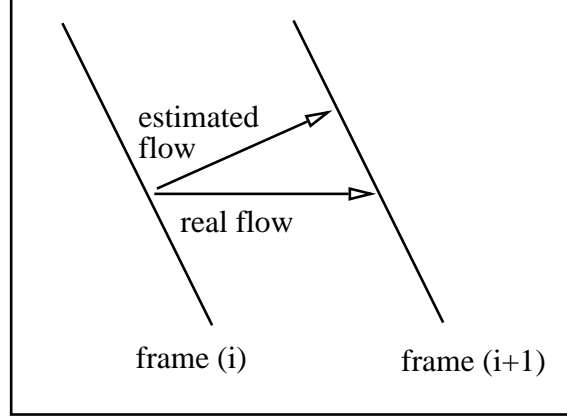


Figure 8.3: *A line in two different frames.*

8.2 Motion Stereo Algorithm

Motion stereo is a natural application of the optical flow algorithm described in Section 7.2.1. Under the assumption that the scene is static during the translation of the camera, the image velocity, at least in principle, is inversely proportional to the depth in the scene. If the focus of expansion is known or estimated, the induced image flow from the camera translation is known. This magnitude of this known image flow will, however, depend on the depth to the feature point.

To estimate the depth from motion the following ideas are utilized. Figure 8.3 illustrates that the optical flow algorithm will, due to the aperture problem, underestimate the flow, i.e. overestimate the disparity, in the case of a moving line. The knowledge, from the position of FOE, of the direction of the image flow, \mathbf{v}_{foe} enables a compensation of this overestimation. The orientation of the line is used to compute the depth, d , which projects onto the normal vector of the line:

$$d_{line} = \frac{\mathbf{v}_{foe} \cdot \mathbf{v}_{line}}{|\mathbf{v}_{line}|^2} \quad (8.6)$$

Moving points, e.g. the end points of the line in Figure 8.3, do not have the aperture problem and the disparity is computed from the optical flow by:

$$d_{point} = 1/v_{point} \quad (8.7)$$

The partitioning to the 'point-case' and to the 'line-case' is performed according to Eq. 7.12.

8.2.1 Results

Two sequences, the diverging tree and the translating tree, have been used for quantitative evaluation of the proposed method. The correct depth in the scene is given in Figure 8.4, where the intensity is proportional to the closeness to the camera. The result from the method, in the points where it is meaningful to calculate the

	ERR _{depth}	σ_{ERR}
Diverging tree	-0.03	0.07
Translating tree	-0.02	0.06

Table 8.1: *The relative depth error for the translating and diverging tree sequences.*

Figure 8.4: *The correct depth for the translating tree (left) and for the diverging tree (right).*

image velocity is given in Figure 8.5. The error is calculated as the mean of the relative depth error, i.e.

$$\Delta d = (d_{\text{est}} - d_{\text{correct}}) / d_{\text{correct}}$$

$$\text{ERR}_{\text{depth}} = \frac{1}{N} \sum_{i=1}^N \Delta d_i$$

and the standard deviation, σ_{ERR} of the errors as

$$\sigma_{\text{ERR}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\Delta d_i - \text{ERR}_{\text{depth}})^2}$$

The performance of the depth estimation for one frame in the translating tree sequence and for one frame in the diverging tree sequence are given in Table 8.1. The correct depths, in these particular frames, are in the interval [11.3, 16.4] for the diverging tree, and in the interval [11.3, 14.7] for the translating tree.

A qualitative result from this method is given in Figure 8.7. The original sequence originates from the image sequence database at Sarnoff Labs, Princeton, USA and the camera is translated perpendicular to the line of sight. One frame from the original sequence is shown in Figure 8.6.

Figure 8.5: *The estimated depth for the translating tree (left) and for the diverging tree (right).*

Figure 8.6: *One frame from the Sarnoff tree sequence.*

Figure 8.7: *The result from the motion stereo algorithm, bright means close to the camera and dark far away.*

Chapter 9

Adaptive Filtering of 3D Signals

In this chapter the extension of the 2D orientation adaptive filtering scheme from Section 5.2 to 3D signal spaces is described. The 3D scheme utilizes the tensor representation from Section 7.1 which ensures a simple control of an adaptive filter for image sequences or other data volumes. The adaptivity of the filter consists in a signal controlled variation of shape and orientation. The shape of the filter has 3 degrees of freedom and can be said to span the range *line - plane - sphere*. The shape of the filter is rotation invariant and the filter can have any orientation. Algorithms for adaptive, or steerable, 3D filters have not been the subject for many publications, but there are a few examples, [6, 36, 42].

The basic idea in the 3D case is the same as in 2D, Section 5.2, i.e. to take advantage of the spatial relations in signal. By for example doing lowpass filtering within an estimated signal the detectability, both for a human observer and for further processing, can be considerably enhanced. The location, orientation and velocity of lines and edges in the sequence or volume are estimated according to the method in Section 7.1. This information is used as contextual control to construct an adaptive filter. The adaptive filter is constructed as a linear combination of 6 fixed filters having the same shape but different orientations.

The method has been tested both on synthetic patterns and on actual recorded sequences. The results are indeed satisfying, which can be seen in the result Section 9.3, on a wide variety of volumes.

9.1 The Algorithm

The estimated local structure and orientation from \mathbf{T}^e , Eq. 7.5, can be used to adapt a filter to the signal. The desired filter containing the degrees of freedom for an orientation adaptive filter can be written as:

$$F(\omega) = F_\omega(\omega) \left(g_1(\lambda_1, \lambda_2, \lambda_3) (\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + g_2(\lambda_1, \lambda_2, \lambda_3) (1 - (\hat{\omega} \cdot \hat{\mathbf{e}}_3)^2) + g_3(\lambda_1, \lambda_2, \lambda_3) \right) \quad (9.1)$$

where $F_\omega(\omega)$ is a bandpass function and

g_1 is a weighting function for a spatial plane filter in the direction of the eigenvector corresponding to the largest eigenvalue, $\hat{\mathbf{e}}_1$, of \mathbf{T}^e .

g_2 is a weighting function for a spatial line filter in the direction of the eigenvector corresponding to the least eigenvalue, $\hat{\mathbf{e}}_3$.

g_3 is a weighting function for an isotropic filter.

A direct extension of the 2D filter, see Eq. 5.12-5.13, yields the following weighting functions, g_i .

$$\begin{aligned} g_1(\lambda_1, \lambda_2, \lambda_3) &= (\lambda_1 - \lambda_2)/\lambda_1 \\ g_2(\lambda_1, \lambda_2, \lambda_3) &= (\lambda_2 - \lambda_3)/\lambda_1 \\ g_3(\lambda_1, \lambda_2, \lambda_3) &= \lambda_3/\lambda_1 \end{aligned} \quad (9.2)$$

These choices give a filter following the signal structure as it is estimated by the representation tensor. The above choice also follows the probabilities from Eq. 7.12 of how the optical flow should be interpreted.

Is it now possible to extend the 2D scheme from Proposition 1 in Section 5.2, using the weighting functions from Eq. 9.2 ?

Proposition 2

Assume six polar separable filters having the same radial frequency function, $F_\omega(\omega)$, and angular functions $(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2$. The directions, $\hat{\mathbf{n}}_k$, are the same as in the estimation procedure, Eq. 7.2. Then the interpolation can be written, [63]:

$$F(\omega) = \frac{5}{4} F_\omega(\omega) \sum_{k=1}^6 \hat{\mathbf{T}}^e \bullet (\mathbf{N}_k - \frac{1}{5} \mathbf{I}) (\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 \quad (9.3)$$

where $\hat{\mathbf{T}}^e$ is the normalized estimate and \bullet symbolizes the tensor inner product defined in Eq. 5.15.

Proof of proposition 2

□

From the spectrum theorem it is possible to split the estimated tensor in three parts

$$\hat{\mathbf{T}}^e = \frac{1}{\lambda_1} \mathbf{T}^e = \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^t + \frac{\lambda_2}{\lambda_1} \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^t + \frac{\lambda_3}{\lambda_1} \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^t \quad (9.4)$$

each one corresponding to *one* orientation. The spectrum theorem shows that it is enough to investigate the interpolation in Eq. 9.3 for a tensor corresponding to *one* orientation.

An representation tensor, \mathbf{T}' , originating a one-dimensional input, $\hat{\mathbf{e}}_1 = (u_1, u_2, u_3)^t$ is

$$\mathbf{T}' = \begin{pmatrix} u_1^2 & u_1 u_2 & u_1 u_3 \\ u_1 u_2 & u_2^2 & u_2 u_3 \\ u_1 u_3 & u_2 u_3 & u_3^2 \end{pmatrix} \quad (9.5)$$

where $\sqrt{u_1^2 + u_2^2 + u_3^2} = 1$. Expanding the filter orientation tensors, $\mathbf{N}_k = \mathbf{n}_k \mathbf{n}_k^t$, yields

$$\begin{aligned}
\mathbf{N}_1 &= c^2 \begin{pmatrix} a^2 & 0 & ab \\ 0 & 0 & 0 \\ ab & 0 & b^2 \end{pmatrix} \\
\mathbf{N}_2 &= c^2 \begin{pmatrix} a^2 & 0 & -ab \\ 0 & 0 & 0 \\ -ab & 0 & b^2 \end{pmatrix} \\
\mathbf{N}_3 &= c^2 \begin{pmatrix} b^2 & ab & 0 \\ ab & a^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\mathbf{N}_4 &= c^2 \begin{pmatrix} b^2 & -ab & 0 \\ -ab & a^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\mathbf{N}_5 &= c^2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & b^2 & ab \\ 0 & ab & a^2 \end{pmatrix} \\
\mathbf{N}_6 &= c^2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & b^2 & -ab \\ 0 & -ab & a^2 \end{pmatrix}
\end{aligned} \tag{9.6}$$

where the constants a, b and c are given in Eq. 7.3.

Next the tensor inner products, $\mathbf{T}' \bullet (\mathbf{N}_k - \frac{1}{5} \mathbf{I})$, are calculated.

$$\begin{aligned}
\mathbf{T}' \bullet (\mathbf{N}_1 - \frac{1}{5} \mathbf{I}) &= c^2 (u_1^2 a^2 + u_3^2 b^2 + 2u_1 u_3 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2) \\
\mathbf{T}' \bullet (\mathbf{N}_2 - \frac{1}{5} \mathbf{I}) &= c^2 (u_1^2 a^2 + u_3^2 b^2 - 2u_1 u_3 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2) \\
\mathbf{T}' \bullet (\mathbf{N}_3 - \frac{1}{5} \mathbf{I}) &= c^2 (u_1^2 b^2 + u_2^2 a^2 + 2u_1 u_2 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2) \\
\mathbf{T}' \bullet (\mathbf{N}_4 - \frac{1}{5} \mathbf{I}) &= c^2 (u_1^2 b^2 + u_2^2 a^2 - 2u_1 u_2 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2) \\
\mathbf{T}' \bullet (\mathbf{N}_5 - \frac{1}{5} \mathbf{I}) &= c^2 (u_2^2 b^2 + u_3^2 a^2 + 2u_2 u_3 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2) \\
\mathbf{T}' \bullet (\mathbf{N}_6 - \frac{1}{5} \mathbf{I}) &= c^2 (u_2^2 b^2 + u_3^2 a^2 - 2u_2 u_3 ab) - \frac{1}{5} (u_1^2 + u_2^2 + u_3^2)
\end{aligned} \tag{9.7}$$

To further investigate the adaptive filter, the filter outputs $(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2$ are expressed in Cartesian coordinates $\hat{\omega} = (\omega_1, \omega_2, \omega_3)^t / \omega$

$$\begin{aligned}
(\hat{\omega} \cdot \hat{\mathbf{n}}_1)^2 &= c^2 \omega^{-2} (a^2 \omega_1^2 + 2ab \omega_1 \omega_3 + b^2 \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_2)^2 &= c^2 \omega^{-2} (a^2 \omega_1^2 - 2ab \omega_1 \omega_3 + b^2 \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_3)^2 &= c^2 \omega^{-2} (b^2 \omega_1^2 + 2ab \omega_1 \omega_2 + a^2 \omega_2^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_4)^2 &= c^2 \omega^{-2} (b^2 \omega_1^2 - 2ab \omega_1 \omega_2 + a^2 \omega_2^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_5)^2 &= c^2 \omega^{-2} (b^2 \omega_2^2 + 2ab \omega_2 \omega_3 + a^2 \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_6)^2 &= c^2 \omega^{-2} (b^2 \omega_2^2 - 2ab \omega_2 \omega_3 + a^2 \omega_3^2)
\end{aligned} \tag{9.8}$$

Now it is possible to express the interpolation from Eq. 9.3

$$\begin{aligned}
\sum_{k=1}^6 \mathbf{T}' \bullet (\mathbf{N}_k - \frac{1}{5}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 = \\
2c^4\omega^{-2} \left[u_1^2 \left((a^4 + b^4)\omega_1^2 + a^2b^2\omega_2^2 + a^2b^2\omega_3^2 \right) + \right. \\
u_2^2 \left(a^2b^2\omega_1^2 + (a^4 + b^4)\omega_2^2 + a^2b^2\omega_3^2 \right) + \\
u_3^2 \left(a^2b^2\omega_1^2 + a^2b^2\omega_2^2 + (a^4 + b^4)\omega_3^2 \right) + \\
\left. 4a^2b^2 (u_1u_2\omega_1\omega_2 + u_1u_3\omega_1\omega_3 + u_2u_3\omega_2\omega_3) \right] - \\
\frac{2c^2\omega^{-2}}{5} \left((a^2 + b^2)(\omega_1^2 + \omega_2^2 + \omega_3^2)(u_1^2 + u_2^2 + u_3^2) \right)
\end{aligned} \tag{9.9}$$

In the simplification of this expression some relations of the constants a, b and c are helpful

$$\begin{aligned}
c^{-2} &= a^2 + b^2 \\
b^2 - a^2 &= ab \\
b^4 + a^4 &= 3a^2b^2 \\
(b^2 + a^2)^2 &= 5a^2b^2
\end{aligned} \tag{9.10}$$

Rearranging terms in Eq. 9.9 and utilizing the relations in Eq. 9.10 gives

$$\begin{aligned}
\sum_{k=1}^6 \mathbf{T}' \bullet (\mathbf{N}_k - \frac{1}{5}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 = \\
2c^4\omega^{-2} \left[2a^2b^2(\omega_1^2u_1^2 + \omega_2^2u_2^2 + \omega_3^2u_3^2) + \right. \\
4a^2b^2 (u_1u_2\omega_1\omega_2 + u_1u_3\omega_1\omega_3 + u_2u_3\omega_2\omega_3) + \\
\left. a^2b^2 \left((\omega_1^2 + \omega_2^2 + \omega_3^2)(u_1^2 + u_2^2 + u_3^2) \right) \right] - \\
\frac{2c^2\omega^{-2}}{5} \left((a^2 + b^2)(\omega_1^2 + \omega_2^2 + \omega_3^2)(u_1^2 + u_2^2 + u_3^2) \right) = \\
\frac{4}{5}\omega^{-2} (\omega \cdot \hat{\mathbf{e}}_1)^2 = \frac{4}{5} (\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2
\end{aligned} \tag{9.11}$$

which is, except from a constant, the filter proposed for a one-dimensional tensor in the direction of $\hat{\mathbf{e}}_1$.

By combining the proof from above and the spectrum theorem, Eq. 9.4, the total filter shape can be calculated.

$$\begin{aligned}
F(\omega) = \\
\frac{5}{4}F_\omega(\omega) \sum_{k=1}^6 \mathbf{T}^e \bullet (\mathbf{N}_k - \frac{1}{5}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 = \\
\frac{5}{4}F_\omega(\omega) \left(\lambda_1(\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + \lambda_2(\hat{\omega} \cdot \hat{\mathbf{e}}_2)^2 + \lambda_3(\hat{\omega} \cdot \hat{\mathbf{e}}_3)^2 \right) = \\
\frac{5}{4}F_\omega(\omega) \left((\lambda_1 - \lambda_2)(\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + (\lambda_2 - \lambda_3) \left(1 - (\hat{\omega} \cdot \hat{\mathbf{e}}_3)^2 \right) + \lambda_3 \right)
\end{aligned} \tag{9.12}$$

By normalizing the tensor in the maximum norm, i.e. with the largest eigenvalue λ_1 , the adaptive filter behaves as proposed in Proposition 2.

□

If the adaptive filter is supposed to be used for enhancement, or reconstruction, of degraded volumes or sequences an appropriate lowpass filter, $F_{LP}(\omega)$ see Eq. 9.18, must be added. The reason is that the low frequencies, including DC, should not be lost in the process since these are important in such applications. The final filter, $F_{ada}(\omega)$, can now be written as

$$F_{ada}(\omega) = F_{LP}(\omega) + \sqrt{\|\mathbf{T}^e\|_{frob}} F_{\omega}(\omega) \frac{5}{4} \sum_{k=1}^6 \mathbf{T}^e \bullet (\mathbf{N}_k - \frac{1}{5} \mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 \quad (9.13)$$

where the weighting if the adaptive bandpass filter should be a part of the output is controlled by the energy, according to Frobenius. This energy measure can also be mapped by a function, e.g. a sigmoid such as $s(E)$ from Appendix 6.B, in order for the adaptive filter to function in different noise situations.

Note that a complete eigenvalue decomposition of the representation tensor, \mathbf{T}^e , is *not* necessary for adapting the filter. When normalizing the tensor in the maximum norm, only the largest eigenvalue, λ_1 , is needed. It is even possible to approximate the adaptive filter of Eq.9.3 only using the Frobenius norm and the trace of \mathbf{T}^e . By normalizing \mathbf{T}^e with the trace, i.e. with

$$Trace(\mathbf{T}) = \sum T_{ii} = T_{11} + T_{22} + T_{33} = \sum \lambda_i$$

none of the eigenvalues need to be calculated. Using the trace for normalizing the tensor is the same as changing the weighting functions, g_i , in the adaptive filtering. The functions will in this case be

$$\begin{aligned} g_1(\lambda_1, \lambda_2, \lambda_3) &= \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \\ g_2(\lambda_1, \lambda_2, \lambda_3) &= \frac{\lambda_2 - \lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \\ g_3(\lambda_1, \lambda_2, \lambda_3) &= \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \end{aligned} \quad (9.14)$$

Compared to the weighting functions in Eq. 9.2 the line filter, weighted by g_2 , will be given smaller weights. However, the difference in appearance of filtered sequences is hard to discover. The results in Section 9.3 are filtered with an adaptive filter, where the representation tensor is normalized in maximum norm, while "line-like" and "plane-like" structures then are given equal importance. If there is a need to minimize the computational complexity in the adaption, the normalization by the trace can be chosen.

To illustrate the possibilities and degrees of freedom for the adaptive filter, Eq. 9.13 some specific cases are plotted as isosurfaces in Figure 9.1.

TOP:	$\lambda_1 = 0$	$\lambda_2 = 0$	$\lambda_3 = 0$	i.e. a lowpass filter
MIDDLE LEFT:	$\lambda_1 = 0.5$	$\lambda_2 = 0$	$\lambda_3 = 0$	i.e. a spatial plane filter
MIDDLE RIGHT:	$\lambda_1 = 1$	$\lambda_2 = 0$	$\lambda_3 = 0$	i.e. a spatial plane filter
BOTTOM LEFT:	$\lambda_1 = 1/2\sqrt{2}$	$\lambda_2 = 1/2\sqrt{2}$	$\lambda_3 = 0$	i.e. a spatial line filter
BOTTOM RIGHT:	$\lambda_1 = 1/\sqrt{2}$	$\lambda_2 = 1/\sqrt{2}$	$\lambda_3 = 0$	i.e. a spatial line filter

Table 9.1: *The eigenvalues and the interpretation of the isosurfaces in Figure 9.1.*

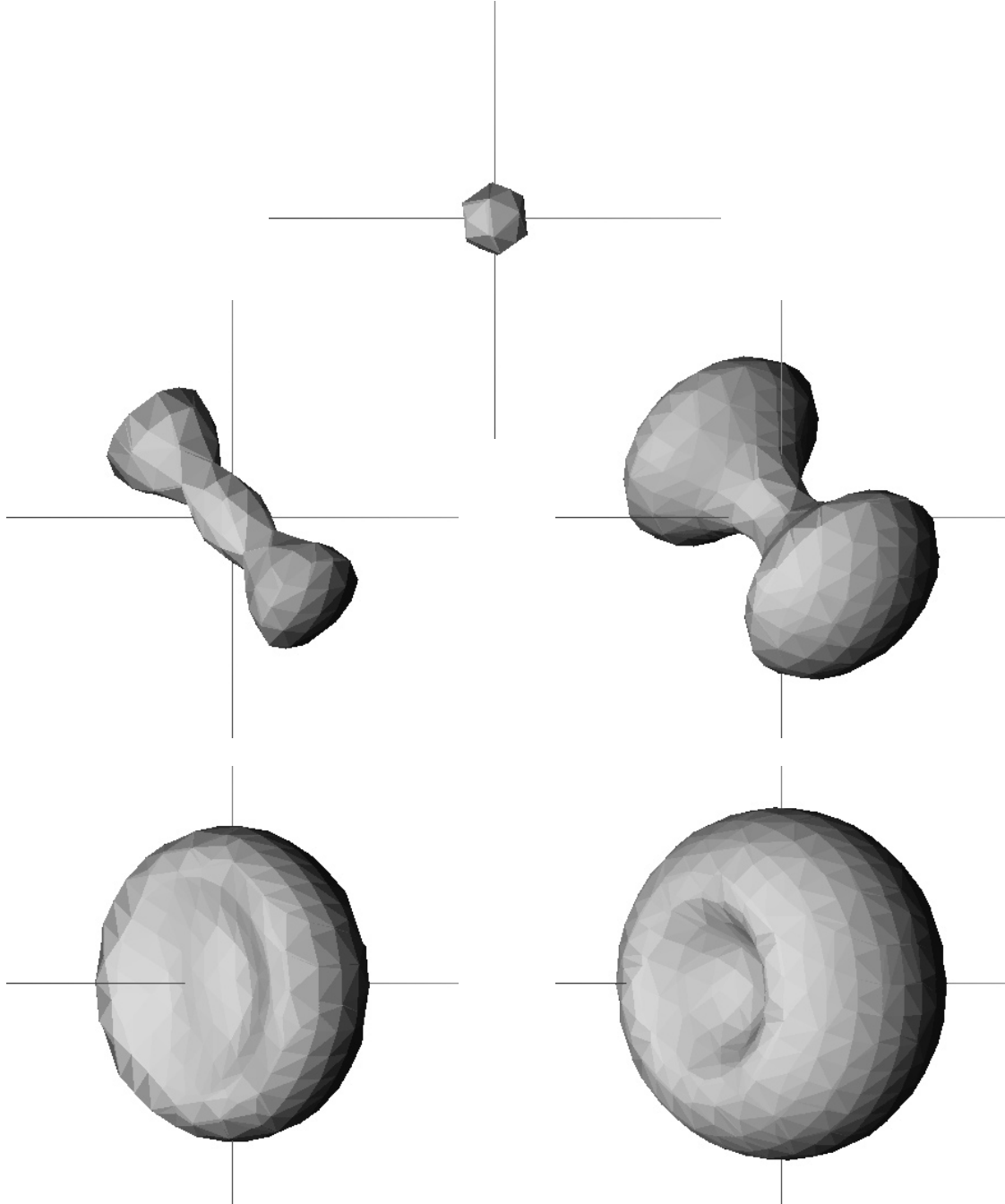


Figure 9.1: *Isosurfaces of adaptive filters in the Fourier domain illustrating the filter shapes for different relations of the eigenvalues of \mathbf{T}^e , see Table 9.1. Compare this figure with Figure 5.2 to get the filter functions.*

9.2 Implementation

The implementation of the 3D algorithm falls naturally into the same three parts as in the 2D case:

1. Estimation of context information.
2. Relaxation of the context information.
3. Adaptive filtering of the original image.

Estimation of Context Information

Considering the estimation of the contextual information described in Section 7.1, the quadrature filters used are defined according to

$$Q_k(\omega) = \begin{cases} f(\omega) (\hat{\omega} \cdot \hat{n}_k)^2 & \text{if } \omega \cdot \hat{n}_k \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (9.15)$$

where $f(\omega)$ is a lognormal function from Eq. 3.16. All parameters used in the filter design, as the center frequency and the bandwidth of the filter above, are chosen so that the corresponding spatial convolution kernel will be possible to realize typically as a 9x9x9 filter, with negligible deviation from the specified frequency response. The center frequency is $\omega_i = \pi/(2\sqrt{2})$ and the bandwidth $B = 2$.

Relaxation of Context Information

An important property of the tensor representation described above is that averaging of such tensors is a meaningful operation [59]. It is thus possible to increase robustness by convolving the tensor field with an averaging filter. The filter used in the experiments has been a truncated isotropic Gaussian filter, $g_{LP}(\mathbf{x})$, of size 7x7x7 having a standard deviation of 1.3 (inter voxel distances). Note that this filtering does not change the sensitivity to high frequency components in the original data, it only means that rapid changes in the representation of local structure are suppressed. This relaxed tensor,

$$\mathbf{T}_{rel} = g_{LP}(\mathbf{x}) * \mathbf{T}^e$$

where $*$ denotes convolution, is then used for a more robust control of the adaptive filter.

Adaptive Filtering

The filters are defined with the angular function $(\hat{\omega} \cdot \hat{n}_k)^2$ and the frequency function $F_\omega(\omega)$ defined as

$$F_\omega(\omega) = \begin{cases} \sin^2(\pi\omega/2) & 0 < \omega < 1 \\ 1 & 1 < \omega < \pi - 1 \\ \sin^2(\pi(\pi - \omega)/2) & \pi - 1 < \omega < \pi \\ 0 & \pi < \omega \end{cases} \quad (9.16)$$

Input SNR	Output SNR
∞ dB	21.0 dB
20 dB	21.0 dB
10 dB	18.5 dB
0 dB	9.8 dB

Table 9.2: *The result of the 3D enhancement algorithm for the volume in Figure 9.2*

This function is chosen to give a flat frequency response, so that the filter will be possible to use iteratively. The proposed contextual control, eq. 9.3, will be a combination of \mathbf{T}_{rel} and the convolutions of the filters specified by eq. 9.16. To apply the adaptive filter to reduce noise, the bandpass filter must be combined with a corresponding lowpass filter defined by:

$$F_{LP}(\omega) = \begin{cases} \cos^2(\pi\omega/2) & 0 < \omega < 1 \\ 0 & 1 < \omega \end{cases} \quad (9.17)$$

The filters $F_{LP}(\omega)$ and $F_{\omega}(\omega)(\hat{\omega} \cdot \hat{\mathbf{n}}_{\mathbf{k}})^2$ are applied as 9x9x9 convolutions kernels.

By denoting the output from the lowpass filter, h_{LP} , and the output from the directed bandpass filters, h_k , the final filter is given as:

$$h_{LP} + \sqrt{\|\mathbf{T}^e\|_{frob}} \sum_k a_k h_k \quad (9.18)$$

where a_k is

$$a_k = \frac{5}{4} \sum_{k=1}^6 \hat{\mathbf{T}}^e \bullet (\mathbf{N}_k - \frac{1}{5}\mathbf{I}) \quad (9.19)$$

9.3 Results

The algorithm has been tested extensively on both synthetic and real image volumes and sequences. In this section some results are presented.

First the test volume used in the performance examination in Section 7.1.3 is used in the test. Sixteen slices of the original volume is shown in Figure 9.2. The original pattern has been corrupted with white additive Gaussian noise to a signal-to-noise ratio, calculated from SDEV of the signal and the noise according to Eq. 7.6, of 0 dB, see Figure 9.3. The same slices from the enhanced volume are shown in Figure 9.4. An improvement of about 10 dB of the SNR has been achieved. To illustrate the noise performance of the algorithm a SNR table has been calculated¹ in Table 9.2.

¹See also the comments about SNR as quality measure in Section 6.2.

Figure 9.2: *Sixteen slices from the original test pattern.*

The second test pattern has also been used in the accuracy estimation and it is rendered in Figure 7.3. The volume has been corrupted with white Gaussian noise to a SNR according to Eq. 7.6 of -4 dB. Sixteen slices from the original sequences and sixteen slices from filtered is shown in Figures 9.5- 9.7. In this case the strength of the spatio-temporal approach is undoubtedly shown.

Figure 9.3: *Sixteen slices from the test pattern corrupted with white additive Gaussian noise, SNR= 0dB.*

Figure 9.4: *Result from 3D enhancement of the test pattern in Figure 9.3.*

Figure 9.5: *Sixteen slices from the original test pattern.*

The third example is an ultrasonic sequence of a beating heart.² Figure 9.8 contains four frames from the original sequence. As can be seen, the sequence is corrupted by a lot of measurement noise which is significantly reduced in the enhanced images, Figure 9.9.

The fourth example is a reconstruction of an interlaced video sequence, Figures 9.10-9.11. The estimation of the orientation tensor is obtained according to the scheme described in Appendix 7.A. Two different adaptive filters are then applied, one designed with its center on the existing sampling points in the interlaced sequence and one designed with its center on the new sampling points, see Figure 7.14.

²Tomas Gustavsson, CTH, Sweden is acknowledged for this sequence.

Figure 9.6: *Sixteen slices from the test pattern corrupted with white additive Gaussian noise, SNR= -4dB.*

Figure 9.7: *Result from 3D enhancement of the test pattern in Figure 9.6.*

Figure 9.8: *Four frames from an ultrasound image sequence of a beating heart, original.*

Figure 9.9: *Result from 3D enhancement of the ultrasound sequence, Figure 9.8.*

Figure 9.10: *A zoomed frame from an interlace sequence.*

Figure 9.11: *The reconstruction using 3D adaptive filtering of the interlaced sequence in Figure 9.10.*

Part IV

4D — Adaptive Filtering

Chapter 10

Filtering of 4D Signals

Volume sequences, i.e. 4D signals, have been accessible from special devices during last years, [88, 89]. However, the progress in image producing devices such as Magnetic Resonance (MR)cameras and positron cameras will in the near future lead to a situation where 4D signals will be common. Today it is for example possible to produce volume sequences in ordinary MR devices. In such situations the human eye has problems to interpret the data and machine vision will have a great potential to guide and interpret the signals. Handling 4D data put hard demands on the computers to use. Today's peak performance is maybe a bit to slow for everyday use, but the challenge will probably be met with the hardware development.

Extending the orientation algorithm and the adaptive filtering strategy from 2D and 3D signals to 4D is therefore natural.

10.1 Orientation Representation and Estimation

Extending the proposed tensor representation from 2D and 3D to 4D is straightforward, [60]. The representation tensor originating from a one dimensional input $\mathbf{x} = (x_1, x_2, x_3, x_4)^t$ is given by:

$$\mathbf{T} = \mathbf{x}\mathbf{x}^t = \frac{1}{x} \begin{pmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_1x_4 \\ x_1x_2 & x_2^2 & x_2x_3 & x_2x_4 \\ x_1x_3 & x_2x_3 & x_3^2 & x_3x_4 \\ x_1x_4 & x_2x_4 & x_3x_4 & x_4^2 \end{pmatrix} \quad (10.1)$$

To estimate the tensor the number of filters should exceed the number of faces of a 4D cube, i.e. it should be greater than 8. The number of filters is also restricted by the demand that they should be distributed in accordance with the vertices of a regular polytope. This leaves the 24-cell [22] as the only alternative. (Computational complexity makes the 120-cell and the 600-cell unrealistic alternatives.)

The 12 filter directions are given in cartesian coordinates by

$$\begin{aligned}
\hat{\mathbf{n}}_1 &= c \begin{pmatrix} 1, & 1, & 0, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_2 &= c \begin{pmatrix} 1, & -1, & 0, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_3 &= c \begin{pmatrix} 1, & 0, & 1, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_4 &= c \begin{pmatrix} 1, & 0, & -1, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_5 &= c \begin{pmatrix} 1, & 0, & 0, & 1 \end{pmatrix}^t \\
\hat{\mathbf{n}}_6 &= c \begin{pmatrix} 1, & 0, & 0, & -1 \end{pmatrix}^t \\
\hat{\mathbf{n}}_7 &= c \begin{pmatrix} 0, & 1, & 1, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_8 &= c \begin{pmatrix} 0, & 1, & -1, & 0 \end{pmatrix}^t \\
\hat{\mathbf{n}}_9 &= c \begin{pmatrix} 0, & 1, & 0, & 1 \end{pmatrix}^t \\
\hat{\mathbf{n}}_{10} &= c \begin{pmatrix} 0, & 1, & 0, & -1 \end{pmatrix}^t \\
\hat{\mathbf{n}}_{11} &= c \begin{pmatrix} 0, & 0, & 1, & 1 \end{pmatrix}^t \\
\hat{\mathbf{n}}_{12} &= c \begin{pmatrix} 0, & 0, & 1, & -1 \end{pmatrix}^t
\end{aligned} \tag{10.2}$$

where $c = \frac{1}{\sqrt{2}}$.

The filters to use are quadrature filters defined by

$$\begin{cases} F_k(\boldsymbol{\omega}) = F_\omega(\omega)(\hat{\boldsymbol{\omega}} \cdot \hat{\mathbf{n}}_k)^2 & \text{if } \boldsymbol{\omega} \cdot \hat{\mathbf{n}}_k > 0 \\ F_k(\boldsymbol{\omega}) = 0 & \text{otherwise} \end{cases} \tag{10.3}$$

where $F_\omega(\omega)$ is a bandpass filter.

Implementing the algorithm is as easy in a 4D signal space as in the lower dimension signal spaces, i.e the tensor is estimated as, [60, 8]

$$\mathbf{T}^e = \sum_{k=1}^{12} q_k (\mathbf{N}_k - \frac{1}{6} \mathbf{I}) \tag{10.4}$$

where q_k is the filter output from filter number k .

When going to implementation of Eq. 10.4 the following scheme is used:

1. Convolve the input data with the twelve complex-valued filters, i.e. perform twenty four scalar convolutions.
2. Compute the magnitude of each complex-valued filter by

$$q_k = \sqrt{q_{ke}^2 + q_{ko}^2}$$

where q_{ke} denotes the filter output of the real part of filter k and q_{ko} denotes the filter output of the imaginary part of filter k .

3. Compute the tensor \mathbf{T}^e by Eq. 7.5, i.e.

$$\mathbf{T}^e = \begin{pmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{12} & T_{22} & T_{23} & T_{24} \\ T_{13} & T_{23} & T_{33} & T_{34} \\ T_{14} & T_{24} & T_{34} & T_{44} \end{pmatrix}$$

where

$$\begin{aligned}
T_{11} &= c^2(q_1 + q_2 + q_3 + q_4 + q_5 + q_6) - S \\
T_{22} &= c^2(q_1 + q_2 + q_7 + q_8 + q_9 + q_{10}) - S \\
T_{33} &= c^2(q_3 + q_4 + q_7 + q_8 + q_{11} + q_{12}) - S \\
T_{44} &= c^2(q_5 + q_6 + q_9 + q_{10} + q_{11} + q_{12}) - S \\
T_{12} &= c^2(q_1 - q_2) \\
T_{13} &= c^2(q_3 - q_4) \\
T_{14} &= c^2(q_5 - q_6) \\
T_{23} &= c^2(q_7 - q_8) \\
T_{24} &= c^2(q_9 - q_{10}) \\
T_{34} &= c^2(q_{11} - q_{12})
\end{aligned}$$

where

$$c^2 = \frac{1}{2} \quad \text{and} \quad S = \frac{1}{6} \sum_{k=1}^{12} q_k$$

This algorithm has been implemented and tested on synthetic data, [8], but the process is quit painful, due to limitation in the hardware. The test implies, though, that the algorithm gives robust and correct results.

10.2 Adaptive Filtering of 4D Signals

In this section the extension of the proposed orientation adaptive algorithm from 2D, Section 5.2, and 3D, Chapter 9, to 4D signal spaces is given. It is not obvious that the scheme holds for 4D signal spaces, since in both 2D and 3D the filter directions have the same scalar product with all the others, i.e. they are equally far away from any of the other. However, in 4D there are two different scalar products, i.e. the angular distance from one filter direction to another can have two different values. The interpolation scheme will be shown to behave in the same way in 4D as in lower dimensional signal spaces, and can be extended in a straightforward manner.

The estimated local structure and orientation from \mathbf{T}^e , Eq. 7.5, can be used to adapt a filter to the signal. desired filter is:

$$\begin{aligned}
F(\omega) = & F_\omega(\omega) \left(g_1(\lambda_1, \lambda_2, \lambda_3, \lambda_4) (\omega \cdot \hat{\mathbf{e}}_1)^2 + \right. \\
& g_2(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \left((\omega \cdot \hat{\mathbf{e}}_1)^2 + (\omega \cdot \hat{\mathbf{e}}_2)^2 \right) + \\
& g_3(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \left(1 - (\omega \cdot \hat{\mathbf{e}}_4)^2 \right) + \\
& \left. g_4(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \right)
\end{aligned} \tag{10.5}$$

where $F_\omega(\omega)$ is a bandpass function with rather high cut-off frequency or designed using a priori knowledge of the signal spectrum and noise spectrum and

g_1 is a weighting function for a spatial volume filter in the direction of the the eigenvector corresponding to the largest eigenvalue, $\hat{\mathbf{e}}_1$, of \mathbf{T}^e .

g_2 is a weighting function for a spatial plane filter in the direction of the eigenvectors corresponding to the largest eigenvalues, $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, of \mathbf{T}^e .

g_3 is a weighting function for a spatial line filter in the direction of the eigenvector corresponding to the least eigenvalue, $\hat{\mathbf{e}}_4$.

g_4 is a weighting function for an isotropic filter.

A direct extension of the 2D and 3D schemes is obtained by choosing the weighting functions, g_i , as:

$$\begin{aligned} g_1(\lambda_1, \lambda_2, \lambda_3, \lambda_4) &= (\lambda_1 - \lambda_2)/\lambda_1 \\ g_2(\lambda_1, \lambda_2, \lambda_3, \lambda_4) &= (\lambda_2 - \lambda_3)/\lambda_1 \\ g_3(\lambda_1, \lambda_2, \lambda_3, \lambda_4) &= (\lambda_3 - \lambda_4)/\lambda_1 \\ g_4(\lambda_1, \lambda_2, \lambda_3, \lambda_4) &= \lambda_4/\lambda_1 \end{aligned} \quad (10.6)$$

Proposition 3

Assume twelve polar separable filters having the same radial frequency function, $F_\omega(\omega)$, and angular functions $(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2$. The directions, $\hat{\mathbf{n}}_k$, are the same as in the estimation procedure, Eq. 10.2. Then the desired interpolation is given by:

$$F(\omega) = F_\omega(\omega) \sum_{k=1}^{12} \hat{\mathbf{T}}^e \bullet (\mathbf{N}_k - \frac{1}{6}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 \quad (10.7)$$

$\hat{\mathbf{T}}^e$ is the normalized estimate and \bullet symbolizes the tensor inner product defined in Eq. 5.15.

Proof of proposition 3

□

Using the spectrum theorem for the positive semi-definite tensor \mathbf{T}^e yields

$$\hat{\mathbf{T}}^e = \frac{1}{\lambda_1} \mathbf{T}^e = \hat{\mathbf{e}}_1 \mathbf{e}_1^t + \frac{\lambda_2}{\lambda_1} \hat{\mathbf{e}}_2 \mathbf{e}_2^t + \frac{\lambda_3}{\lambda_1} \hat{\mathbf{e}}_3 \mathbf{e}_3^t + \frac{\lambda_4}{\lambda_1} \hat{\mathbf{e}}_4 \mathbf{e}_4^t \quad (10.8)$$

showing that it is enough to investigate the orientation interpolation in Eq. 10.7 for a tensor corresponding to *one* orientation. An estimation tensor, \mathbf{T}' , from a one dimensional input, $\hat{\mathbf{e}}_1 = (u_1, u_2, u_3, u_4)^t$ is

$$\mathbf{T}' = \begin{pmatrix} u_1^2 & u_1 u_2 & u_1 u_3 & u_1 u_4 \\ u_1 u_2 & u_2^2 & u_2 u_3 & u_2 u_4 \\ u_1 u_3 & u_2 u_3 & u_3^2 & u_3 u_4 \\ u_1 u_4 & u_2 u_4 & u_3 u_4 & u_4^2 \end{pmatrix} \quad (10.9)$$

where $u = \sqrt{u_1^2 + u_2^2 + u_3^2 + u_4^2} = 1$.

The filter orientation tensors $\mathbf{N}_k = \hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^t$'s are given by

$$\begin{aligned}
\mathbf{N}_1 &= c^2 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \mathbf{N}_2 &= c^2 \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{N}_3 &= c^2 \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \mathbf{N}_4 &= c^2 \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{N}_5 &= c^2 \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} & \mathbf{N}_6 &= c^2 \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \\
\mathbf{N}_7 &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \mathbf{N}_8 &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\mathbf{N}_9 &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} & \mathbf{N}_{10} &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \\
\mathbf{N}_{11} &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \mathbf{N}_{12} &= c^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}
\end{aligned} \tag{10.10}$$

Calculating the tensor inner products $\mathbf{T}' \bullet (\mathbf{N}_k - \frac{1}{6}\mathbf{I})$ gives

$$\begin{aligned}
\mathbf{T}' \bullet (\mathbf{N}_1 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 + 2u_1u_2 + u_2^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_2 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 - 2u_1u_2 + u_2^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_3 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 + 2u_1u_3 + u_3^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_4 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 - 2u_1u_3 + u_3^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_5 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 + 2u_1u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_6 - \frac{1}{6}\mathbf{I}) &= c^2 (u_1^2 - 2u_1u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_7 - \frac{1}{6}\mathbf{I}) &= c^2 (u_2^2 + 2u_2u_3 + u_3^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_8 - \frac{1}{6}\mathbf{I}) &= c^2 (u_2^2 - 2u_2u_3 + u_3^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_9 - \frac{1}{6}\mathbf{I}) &= c^2 (u_2^2 + 2u_2u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_{10} - \frac{1}{6}\mathbf{I}) &= c^2 (u_2^2 - 2u_2u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_{11} - \frac{1}{6}\mathbf{I}) &= c^2 (u_3^2 + 2u_3u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2 \\
\mathbf{T}' \bullet (\mathbf{N}_{12} - \frac{1}{6}\mathbf{I}) &= c^2 (u_3^2 - 2u_3u_4 + u_4^2) - \frac{1}{6} \sum_{k=1}^6 u_k^2
\end{aligned} \tag{10.11}$$

The filter outputs:

$$\begin{aligned}
(\hat{\omega} \cdot \hat{\mathbf{n}}_1)^2 &= c^2 \omega^{-2} (\omega_1^2 + 2\omega_1\omega_2 + \omega_2^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_2)^2 &= c^2 \omega^{-2} (\omega_1^2 - 2\omega_1\omega_2 + \omega_2^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_3)^2 &= c^2 \omega^{-2} (\omega_1^2 + 2\omega_1\omega_3 + \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_4)^2 &= c^2 \omega^{-2} (\omega_1^2 - 2\omega_1\omega_3 + \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_5)^2 &= c^2 \omega^{-2} (\omega_1^2 + 2\omega_1\omega_4 + \omega_4^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_6)^2 &= c^2 \omega^{-2} (\omega_1^2 - 2\omega_1\omega_4 + \omega_4^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_7)^2 &= c^2 \omega^{-2} (\omega_2^2 + 2\omega_2\omega_3 + \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_8)^2 &= c^2 \omega^{-2} (\omega_2^2 - 2\omega_2\omega_3 + \omega_3^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_9)^2 &= c^2 \omega^{-2} (\omega_2^2 + 2\omega_2\omega_4 + \omega_4^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_{10})^2 &= c^2 \omega^{-2} (\omega_2^2 - 2\omega_2\omega_4 + \omega_4^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_{11})^2 &= c^2 \omega^{-2} (\omega_3^2 + 2\omega_3\omega_4 + \omega_4^2) \\
(\hat{\omega} \cdot \hat{\mathbf{n}}_{12})^2 &= c^2 \omega^{-2} (\omega_3^2 - 2\omega_3\omega_4 + \omega_4^2)
\end{aligned} \tag{10.12}$$

Summing the filter outputs weighted by the tensor inner products now gives

$$\begin{aligned}
& \sum_{k=1}^{12} \mathbf{T}' \bullet (\mathbf{N}_k - \frac{1}{6}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 = \\
& 2c^4\omega^{-2} [\\
& \quad u_1^2 (3\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) + u_2^2 (\omega_1^2 + 3\omega_2^2 + \omega_3^2 + \omega_4^2) + \\
& \quad u_3^2 (\omega_1^2 + \omega_2^2 + 3\omega_3^2 + \omega_4^2) + u_4^2 (\omega_1^2 + \omega_2^2 + \omega_3^2 + 3\omega_4^2) + \\
& \quad 4(u_1u_2\omega_1\omega_2 + u_1u_3\omega_1\omega_3 + u_1u_4\omega_1\omega_4 + \\
& \quad u_2u_3\omega_2\omega_3 + u_2u_4\omega_2\omega_4 + u_3u_4\omega_3\omega_4)] - \\
& \frac{1}{6}c^2\omega^{-2} \left((u_1^2 + u_2^2 + u_3^2 + u_4^2) (6\omega_1^2 + 6\omega_2^2 + 6\omega_3^2 + 6\omega_4^2) \right) = \\
& \omega^{-2} (u_1\omega_1 + u_2\omega_2 + u_3\omega_3 + u_4\omega_4)^2 = (\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2
\end{aligned} \tag{10.13}$$

proving that the orientation adaptivity algorithm works for 4D signal spaces in the same way as in 2D and 3D signal spaces.

By applying the spectrum theorem in same way as in the proofs for the 2D and 3D the final filter is given

It remains to show the interpolation from all the filters from the spectrum theorem

$$\begin{aligned}
F(\omega) &= \\
F_\omega(\omega) \sum_{k=1}^{12} \mathbf{T}^e \bullet (\mathbf{N}_k - \frac{1}{6}\mathbf{I})(\hat{\omega} \cdot \hat{\mathbf{n}}_k)^2 &= \\
F_\omega(\omega) \left(\lambda_1(\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + \lambda_2(\hat{\omega} \cdot \hat{\mathbf{e}}_2)^2 + \lambda_3(\hat{\omega} \cdot \hat{\mathbf{e}}_3)^2 + \lambda_4(\hat{\omega} \cdot \hat{\mathbf{e}}_4)^2 \right) &= \tag{10.14} \\
F_\omega(\omega) \left((\lambda_1 - \lambda_2)(\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + (\lambda_2 - \lambda_3) \left((\hat{\omega} \cdot \hat{\mathbf{e}}_1)^2 + (\hat{\omega} \cdot \hat{\mathbf{e}}_2)^2 \right) + \right. \\
&\quad \left. (\lambda_3 - \lambda_4) \left(1 - (\hat{\omega} \cdot \hat{\mathbf{e}}_4)^2 \right) + \lambda_4 \right)
\end{aligned}$$

Normalizing the control tensor, \mathbf{T}^e in maximum norm gives the desired filter from Eq. 10.5 and Eq. 10.6.

□

It would have been truly interesting to test this strategy on real data, but that will be a question for the future.

Bibliography

- [1] J. F. Abramatic and L. M. Silverman. Non-stationary linear restoration of noisy images. In *Proceedings IEEE Conf. on Decision and Control*, pages 92–99, Fort Lauderdale, FL., 1979.
- [2] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Jour. of the Opt. Soc. of America*, 2:284–299, 1985.
- [3] E. H. Adelson and J. A. Movshon. Phenomenal coherence of moving gratings. *Nature*, 200:523–525, 1982.
- [4] P. Anadan. A computational framework and an algorithm for the measurement of visual motion. *Int. J. of Computer Vision*, 2:283–310, 1989.
- [5] M. Andersson and H. Knutsson. Orientation estimation in ambiguous neighbourhoods. In *Proceedings of SCIA91*, Aalborg, Denmark, 1991.
- [6] M. T. Andersson. *Controllable Multidimensional Filters in Low Level Computer Vision*. PhD thesis, Linköping University, Sweden, S-581 83 Linköping, Sweden, September 1992. Dissertation No 282, ISBN 91-7870-981-4.
- [7] M. A. Arbib and A. Hanson, editors. *Vision, Brain and Cooperative Computation*, pages 187–207. MIT Press, 1987.
- [8] H. Bårman. *Hierachical Curvature Estimation in Computer Vision*. PhD thesis, Linköping University, Sweden, S-581 83 Linköping, Sweden, September 1991. Dissertation No 253, ISBN 91-7870-797-8.
- [9] H. Bårman, L. Haglund, H. Knutsson, and G. H. Granlund. Estimation of velocity, acceleration and disparity in time sequences. In *Proceedings of IEEE Workshop on Visual Motion*, pages 44–51, Princeton, NJ, USA, October 1991.
- [10] J.L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. Technical Report RPL-TR-9107, Robotics and Perception Laboratory, Queen’s University, Canada, 1992.
- [11] J.L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. In *Proc. of the CVPR*, pages 236–242, Champaign, Illinois, USA, 1992. IEEE.
- [12] F. Bergholm. Edge focusing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(6):726–741, 1987.

- [13] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proc. of the IEEE*, 76(8):869–889, 1988.
- [14] J. Bigün. *Local Symmetry Features in Image Processing*. PhD thesis, Linköping University, Sweden, 1988. Dissertation No 179, ISBN 91-7870-334-4.
- [15] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation: texture analysis and optical flow. In *Proceedings of the SSAB Symposium on Image Analysis*. SSAB, March 1991.
- [16] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.
- [17] R. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 2nd edition, 1986.
- [18] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comm.*, 31:532–540, 1983.
- [19] A. Calway. *The Multiresolution Fourier Transform*. PhD thesis, University of Warwick, 1989.
- [20] F. W. Campbell and J. G. Robson. Application of Fourier analysis to the visibility of gratings. *J. Physiol.*, 197:551–566, 1968.
- [21] S. Clippingdale and R. Wilson. Quad-tree image estimation: a new image model and its application to minimum mean square error image restoration. In *Proc. 5th Scand. Conf. on Image Anal.*, 1987.
- [22] H. S. M. Coxeter. *Introduction to Geometry*. John Wiley & Sons, Inc., 1961.
- [23] J. Crowley and A. Parker. A representation for shape based on peaks and ridges in difference of low-pass transform. *IEEE Trans. on PAMI*, (6):156–169, 1989.
- [24] P.E. Danielsson and O. Seger. Rotation invariance in gradient and higher derivative detectors. *Computer Vision, Graphics, and Image Processing*, 49(2), February 1990.
- [25] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. on Information Theory*, 36(5):961–1005, September 1990.
- [26] I. Daubechies, A. Grossmann, and Y. Meyer. Painless nonorthogonal expansions. *J. Math. Phys*, 26(5):1271 – 1283, May 1986.
- [27] Hugh Davson, editor. *The Eye*, volume 2A. Academic Press, New York, 2nd edition, 1976.
- [28] R. L. DeValois, D. G. Albrecht, and L. G. Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:549–559, 1982.

- [29] S. M. Pizer et al. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Trans. on PAMI*, 31:156–177, 1985.
- [30] F. L. Van Nes and M. A. Bouman. Variation of contrast with luminance. *J. Opt. Soc. of America*, 57:401–406, 1967.
- [31] Marie Farge. Wavelet transforms and their applications to turbulence. *Annu. Rev. Fluid Mech.*, 24:395–457, 1992.
- [32] D. J. Fleet and A. D. Jepson. Stability of phase information. In *Proceedings of IEEE Workshop on Visual Motion*, pages 52–60, Princeton, USA, October 1991. IEEE, IEEE Society Press.
- [33] David J. Fleet. *Measurement of image velocity*. Kluwer Academic Publishers, 1992. ISBN 0-7923-9198-5.
- [34] David J. Fleet and Allan D. Jepson. Computation of Component Image Velocity from Local Phase Information. *Int. Journal of Computer Vision*, 5(1):77–104, 1990.
- [35] David J. Fleet, Allan D. Jepson, and Michael R. M. Jenkin. Phase-based disparity measurement. *CVGIP Image Understanding*, 53(2):198–210, March 1991.
- [36] W. T. Freeman and E. H. Adelson. The design and use of steerable filters for image analysis, enhancement, and wavelet representation. Technical report, Vision and Modeling Group, Media Lab., MIT, Cambridge, September 1990.
- [37] D. Gabor. Theory of communication. *Proc. Inst. Elec. Eng.*, 93(26):429–441, 1946.
- [38] M. A. Georgeson and M. G. Harris. Spatial selectivity of contrast adaptation: Models and data. *Vision Research*, 24:729–741, 1984.
- [39] G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–178, 1978.
- [40] G. H. Granlund and H. Knutsson. Contrast of structured and homogenous representations. In O. J. Braddick and A. C. Sleight, editors, *Physical and Biological Processing of Images*, pages 282–303. Springer Verlag, Berlin, 1983.
- [41] R. Guissin and S. Ullman. Direct Computation of the Focus of Expansion from Velocity Field Measurements. In *Proceedings of IEEE Workshop on Visual Motion*, pages 146–155, Princeton, USA, October 1991. IEEE, IEEE Society Press.
- [42] T Gustavsson and S Nivall. Adaptive spatio-temporal filtering of ultrasound image sequences. In *Proceedings of the SSAB Conf. on Image Analysis*, pages 104–108, Gothenburg, Sweden, March 1989. SSAB.
- [43] P. Hackman. *Boken med kossan på, Läroplamflett i Linjär Algebra*. LiTH, Linköpings Universitet, 1983.

- [44] L. Haglund, H. Knutsson, and G. H. Granlund. On phase representation of image information. In *The 6th Scandinavian Conference on Image Analysis*, pages 1082–1089, Oulu, Finland, June 1989.
- [45] L. Haglund, H. Knutsson, and G. H. Granlund. Scale analysis using phase representation. In *The 6th Scandinavian Conference on Image Analysis*, pages 1118–1125, Oulu, Finland, June 1989.
- [46] L. Haglund, H. Knutsson, and G.H. Granlund. On scale and orientation adaptive filtering. In *Proceedings of the SSAB Symposium on Image Analysis*, Uppsala, March 1992. SSAB.
- [47] O. Hansen. *On the use of Local Symmetries in Image Analysis and Computer Vision*. PhD thesis, Aalborg University, March 1992.
- [48] D. J. Heeger. Optical flow from spatiotemporal filters. In *First Int. Conf. on Computer Vision*, pages 181–190, London, June 1987.
- [49] D. J. Heeger. Optical Flow Using Spatio-Temporal Filters. *Int. Journal of Computer Vision*, 2(1):279–302, 1988.
- [50] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *Int. Journal of Computer Vision*, 7(2):95–117, Januari 1992.
- [51] W. Hoff and N. Ahuja. Depth from stereo. In *Proc. of the fourth Scandinavian Conf. on Image Analysis*, 1985.
- [52] B. K. P. Horn. *Robot vision*. The MIT Press, 1986.
- [53] T. S. Huang, editor. *Image Sequence Analysis*. Information Sciences. Springer Verlag, Berlin, 1981.
- [54] T. S. Huang, editor. *Image Sequence Processing and Dynamic Scene Analysis*, Berlin, June 1982. NATO Advanced Study Institute, Springer Verlag.
- [55] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.*, 148:574–591, 1959.
- [56] *IEEE Trans. on Information Theory*, March 1992. Special Issue on Wavelet Transforms and Multiresolution Signal Analysis.
- [57] Bernd Jähne. Motion determination in space-time images. In O. Faugeras, editor, *Computer Vision-ECCV90*, pages 161–173. Springer-Verlag, 1990.
- [58] P. E. King-Smith and J. J. Kulikowski. The detection and recognition of two lines. *Vision Research*, 21:235–250, 1981.
- [59] H. Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH-ISY-I-1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.

- [60] H. Knutsson, H. Bårman, and L. Haglund. Robust orientation estimation in 2d, 3d and 4d using tensors. In *Proceedings of International Conference on Automation, Robotics and Computer Vision*, September 1992.
- [61] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, Pasadena, October 1983.
- [62] H. Knutsson and G. H. Granlund. Apparatus for determining the degree of variation of a feature in a region of an image that is divided into discrete picture elements. Swedish patent 8502569-0 (US-Patent 4.747.151, 1988), 1986.
- [63] H. Knutsson, L. Haglund, H. Bårman, and G. H. Granlund. A framework for anisotropic adaptive filtering and analysis of image sequences and volumes. In *Proceedings ICASSP-92*, San Fransisco, CA, USA, March 1992. IEEE.
- [64] H. Knutsson, L. Haglund, and G. Granlund. Adaptive filtering of image sequences and volumes. In *Proceedings of International Conference on Automation, Robotics and Computer Vision*, September 1992.
- [65] H. Knutsson, L. Haglund, and G. H. Granlund. Tensor field controlled image sequence enhancement. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 163–167, Linköping, Sweden, March 1990. SSAB. Report LiTH-ISY-I-1087, Linköping University, Sweden, 1990.
- [66] H. Knutsson, R. Wilson, and G. H. Granlund. Anisotropic non-stationary image estimation and its applications — part I: Restoration of noisy images. *IEEE Trans on Communications*, COM-31(3):388–397, March 1983. Report LiTH-ISY-I-0462, Linköping University, Sweden, 1981.
- [67] Hans Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.
- [68] Hans Knutsson. Producing a continuous and distance preserving 5-D vector representation of 3-D orientation. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, pages 175–182, Miami Beach, Florida, November 1985. IEEE. Report LiTH-ISY-I-0843, Linköping University, Sweden, 1986.
- [69] J. J. Koenderink and A. J. van Doorn. Dynamic shape. *Biological Cybernetics*, 1986.
- [70] J.J. Koenderink and A.J. van Doorn. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [71] J. J. Kulikowski and P. E. King-Smith. Spatial arrangement of line, edge and grating detectors revealed by subthreshold summation. *Vision Research*, 13:1455–1478, 1973.

- [72] K. Langley, T.J. Atherton, R.G. Wilson, and M.H.E Larcombe. Vertical and horizontal disparities from phase. In O. Faugeras, editor, *Computer Vision-ECCV90*, pages 315–325. Springer-Verlag, April 1990.
- [73] Tony Lindeberg. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*. PhD thesis, Royal Institute of Technology, 1991.
- [74] B. Lucas and T. Kanade. An Iterative Image Registration Technique with Applications to Stereo Vision. In *Proc. Darpa IU Workshop*, pages 121–130, 1981.
- [75] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. PaMI*, 11:674–693, July 1989.
- [76] S. Marcelja. Mathematical description of the responses of simple cortical cells. *Journal of Opt. American Society*, 70, 1297–1300 1980.
- [77] K. V. Mardia. *Statistics of Directional Data*. Academic Press, 1972.
- [78] D. Marr, T. Poggio, and S. Ullman. Bandpass channels, zero-crossings, and early visual information processing. *Journal of Opt. American Society*, 69, 914–916 1977.
- [79] H. Mostafavi and D. J. Sakrison. Structure and properties of a single channel in the human visual system. *Vision Research*, 16:957–968, 1976.
- [80] H. H. Nagel. Constraints for the estimation of displacement vector fields from an image sequence. In *Proceedings of Int. Joint. Conf. on Artificial Intelligence in West Germany*, 1983.
- [81] S. I. Olsen. Concurrent solution of the stereo correspondence problem and the surface reconstruction problem. In *Proc. of the Eighth Int. Conf. on Pattern Recognition, Paris*, pages 1038–1040, 1986.
- [82] D. A. Pollen and S. F. Ronner. Spatial computation performed by simple and complex cells in the visual cortex of the cat. *Vision Research*, 22:101–118, 1982.
- [83] D. A. Pollen and S. F. Ronner. Visual cortical neurons as localized spatial frequency filters. *IEEE Trans. on Syst. Man Cybern.*, 13(5):907–915, 1983.
- [84] M. Porat and Y. Y. Zeevi. The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision. *IEEE Trans. on PAMI*, 10(4):452–467, 1988.
- [85] W. K. Pratt. Generalized Wiener filtering computation techniques. *IEEE Trans. Comput.*, C-21:297–303, July 1972.
- [86] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes*. Cambridge University Press, 1986.

- [87] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–38, October 1991.
- [88] Richard A. Robb. The dynamic spatial reconstructor: An x-ray video-fluoroscopic ct scanner for dynamic volume imaging of moving organs. *IEEE Trans.s on Medical Imaging*, MI-1(1):22–33, July 1982.
- [89] Richard A. Robb, Eric A. Hoffman, Lawrence J. Sinak, Lowell D. Harris, and Erik L. Ritman. High-speed three-dimensional x-ray computed tomography: The dynamic spatial reconstructor. *Proceedings of the IEEE*, 71(3):308–319, March 1983.
- [90] E. Simoncelli and E. H. Adelson. Subband transforms. In J. Woods, editor, *Subband Image Coding*, chapter 4, pages 143–192. Kluwer, 1991.
- [91] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, pages 79–97, 1988.
- [92] A. B. Watson and Jr. A. J. Ahumada. Model of human visual-motion sensing. *Jour. of the opt. soc. of America A*, 1(2):322–342, 1985.
- [93] C-J Westelius. Preattentive gaze control for robot vision, June 1992. Thesis No. 322, ISBN 91–7870–961–X.
- [94] C-F Westin. Feature extraction based on a tensor image description, September 1991. Thesis No. 288, ISBN 91–7870–815–X.
- [95] J. Wiklund, C-J Westelius, and H. Knutsson. Hierarchical phase based disparity estimation. Report LiTH-ISY-I-1327, Computer Vision Laboratory, Linköping University, Sweden, 1992.
- [96] R. Wilson, A. D. Calway, and E. R. S. Pearson. A Generalized Wavelet Transform for Fourier Analysis: The Multiresolution Fourier Transform and Its Application to Image and Audio Signal Analysis. *IEEE Trans. on Information Theory*, 38(2):674–690, 1992.
- [97] R. Wilson and G. H. Granlund. The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6), November 1984. Report LiTH-ISY-I-0576, Computer Vision Laboratory, Linköping University, Sweden, 1983.
- [98] R. Wilson and H. Knutsson. A multiresolution stereopsis algorithm based on the Gabor representation. In *3rd International Conference on Image Processing and Its Applications*, pages 19–22, Warwick, Great Britain, July 1989. IEE. ISBN 0 85296382 3 ISSN 0537–9989.
- [99] R. Wilson and M. Spann. *Image segmentation and uncertainty*. Research Studies Press, 1988.
- [100] A. Witkin. Scale-space filtering. In *8th Int. Joint Conf. Artificial Intelligence*, pages 1019–1022, Karlsruhe, 1983.