# A Tensor Framework for Multidimensional Signal Processing

Carl-Fredrik Westin

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden

About the cover

The figure on the cover shows a visualization of a symmetric tensor in three dimensions,

$$\mathbf{G} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T$$

The object in the figure is the sum of a spear, a plate and a sphere. The spear describes the principal direction of the tensor $\lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T$, where the length is proportional to the largest eigenvalue, $\lambda_1$. The plate describes the plane spanned by the eigenvectors corresponding to the two largest eigenvalues, $\lambda_2(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T)$. The sphere, with a radius proportional to the smallest eigenvalue, shows how isotropic the tensor is, $\lambda_3(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T)$. The visualization is done using AVS [WWW94]. I am very grateful to Johan Wiklund for implementing the tensor viewer module used.

# Abstract

This thesis deals with filtering of multidimensional signals. A large part of the thesis is devoted to a novel filtering method termed "Normalized convolution". The method performs local expansion of a signal in a chosen filter basis which not necessarily has to be orthonormal. A key feature of the method is that it can deal with uncertain data when additional certainty statements are available for the data and/or the filters. It is shown how false operator responses due to missing or uncertain data can be significantly reduced or eliminated using this technique. Perhaps the most well-known of such effects are the various 'edge effects' which invariably occur at the edges of the input data set. The method is an example of the *signal/certainty* - philosophy, i.e. the separation of both data and operator into a signal part and a certainty part. An estimate of the certainty must accompany the data. Missing data are simply handled by setting the certainty to zero. Localization or windowing of operators is done using an applicability function, the operator equivalent to certainty, not by changing the actual operator coefficients. Spatially or temporally limited operators are handled by setting the applicability function to zero outside the window.

The use of tensors in estimation of local structure and orientation using spatio-temporal quadrature filters is reviewed and related to dual tensor bases. The tensor representation conveys the degree and type of local anisotropy. For image sequences, the shape of the tensors describe the local structure of the spatio-temporal neighbourhood and provides information about local velocity. The tensor representation also conveys information for deciding if true flow or only normal flow is present. It is shown how normal flow estimates can be combined into a true flow using averaging of this tensor field description.

Important aspects of representation and techniques for grouping local orientation estimates into global line information are discussed. The uniformity of some standard parameter spaces for line segmentation is investigated. The analysis shows that, to avoid discontinuities, great care should be taken when choosing the parameter space for a particular problem. A new parameter mapping well suited for line extraction, the Möbius strip parameterization, is defined. The method has similarities to the Hough Transform.

Estimation of local frequency and bandwidth is also discussed. Local frequency is an important concept which provides an indication of the appropriate range of scales for subsequent analysis. One-dimensional and two-dimensional examples of local frequency estimation are given. The local bandwidth estimate is used for defining a certainty measure. The certainty measure enables the use of a normalized averaging process increasing robustness and accuracy of the frequency statements.

# PREFACE

The thesis consists of eight chapters. Chapter 1 gives a background to the work and presents the goals and philosophy behind the approaches in this thesis. Chapter 2 presents some basic notions from tensor analysis and functional analysis which gives a useful background to the material presented later.

Although a number of details have been added while finalizing this thesis, the basic ideas presented in the chapters 3 - 8 have been published earlier. A list of references on which the chapters are based on is found below.

C-F Westin and H. Knutsson. Processing Incomplete and Uncertain Data using Subspace Methods. In *Proceedings of 12th International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994. IAPR.

C-F Westin, K. Nordberg, and H. Knutsson. On the equivalence of normalized convolution and normalized differential convolution. In *Proceedings of IEEE International Conference on Acoustics, Speech, & Signal Processing*, Adelaide, Australia, April 1994. IEEE.

H. Knutsson, C-F Westin, and G.H. Granlund. Local Multiscale Frequency and Bandwidth Estimation. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, November 1994. IEEE.

C-F Westin and H. Knutsson. Estimation of Motion Vector Fields using Tensor Field Filtering. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, November 1994. IEEE.

C-F Westin. Line extraction using tensors. In H. I. Christensen and J.L. Crowley, editors, *Vision as Process*, Kluwer Academic Publishers, 1994.

C-F Westin. Using non-orthogonal filter bases. In *2nd EC-Israel-US workshop*, The Hebrew University of Jerusalem, October 1994. ESPRIT "SAM".

C-F Westin. Vector and tensor field filtering. In G.H. Granlund and H. Knutsson, principal authors, *Signal Processing for Computer Vision*, Kluwer Academic Publishers, 1994.

H. Knutsson and C-F Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York City, USA, June 1993. IEEE.

C-F Westin and H. Knutsson. The Möbius strip parameterization for line extraction. In *Proceedings of ECCV–92, LNCS–Series Vol. 588*. Springer–Verlag, 1992. LiTH–ISY–R–1514, Linköping University, Sweden.

Related material to this work but not explicitly reviewed in the thesis are:

C-J Westelius, J. Wiklund, and C-F Westin. Prototyping, visualization and simulation using the application visualization system. In H. I. Christensen and J.L. Crowley, editors, *Experimental Environments for Computer Vision and Image Processing*, volume 11 of *Series on Machine Perception and Artificial Intelligence*, pages 33–62. World Scientific Publishers, 1994. ISBN 981-02-1510-X.

C-F Westin. Representation and Averaging. In G.H. Granlund and H. Knutsson, principal authors, *Signal Processing for Computer Vision*, Kluwer Academic Publishers, 1994.

C-F Westin and C-J Westelius. ESPRIT Basic Research Action 7108, Vision as Process, Integration of Low-level FOA & Control Mechanisms. Report, Computer Vision Laboratory, S–581 83 Linköping, Sweden, 1993.

C-F Westin. ESPRIT Basic Research Action 3038, Vision as Process, Model Support and Local FOA Control. Report, Computer Vision Laboratory, S–581 83 Linköping, Sweden, 1992.

C-F Westin and H. Knutsson. Extraction of local symmetries using tensor field filtering. In *Proceedings of 2nd Singapore International Conference on Image Processing*. IEEE Singapore Section, September 1992. LiTH–ISY–R–1515, Linköping University, Sweden.

C-F Westin. Feature extraction based on a tensor image description, September 1991. Thesis No. 288, ISBN 91–7870–815–X.

C-F Westin and C-J Westelius. Brain chaos. A feature or a bug? Report LiTH–ISY–I–0990, Computer Vision Laboratory, Linköping University, Sweden, 1989.

C-J Westelius and C-F Westin. A colour representation for scale-spaces. In *The 6th Scandinavian Conference on Image Analysis*, pages 890–893, Oulu, Finland, June 1989.

C-F Westin and C-J Westelius. A colour model for hierarchical image process-
ing. Master's thesis, Linköping University, Sweden, August v1988. LiTH–
ISY–EX–0857.

# Acknowledgements

I wish to express my sincere gratitude to the many people, who have in different ways contributed to the work presented in this thesis.

First of all I would like to thank my supervisor, Prof. Gösta Granlund for introducing me to the exciting field of Computer Vision and revealing many of its hidden secrets based on his broad experience, and for providing an stimulating and cheerful research environment to work in.

I would like to thank Prof. Hans Knutsson, with whom I have been working closely throughout this work, for being a constant source of inspiration. His ability to get working ideas never stops astonishing me. His comments regarding the presentation of this work have been most valuable.

A special thank to Klas Nordberg for always taking his time to discuss my mathematical problems and for all the time spent reading drafts of this thesis. His comments regarding both scientific and editorial issues have improved the final result considerably.

I would like to thank to thank Carl-Johan Westelius, my friend and colleague, for all useful "tools" he could not resist making and for helping me debugging my programs. Without his help this work would have taken a considerably longer time. His comments regarding this manuscript and the editorial help is very appreciated.

I would like to thank Dr. Leif Haglund for many inspiring discussions and for cheering me up via internet the year he spent in Canada. His comments on this manuscript first via mail and then on his return, has been most useful.

I would like to thank Johan Wiklund and Mikael Wedlin for providing an excellent computer environment to work in.

I would like to thank Dr. Magnus Herbertsson for always taking time for discussions and sharing his great knowledge in tensor theory.

I would like to thank Prof. Roland Wilson for many valuable comments on this manuscript.

I would like to thank Mats Gökstorp for valuable comments on this manuscript.

I would like to thank Catharina Holmgren for her skillful proof reading.

I would like to express my gratitude for the constant encouragement I have received from Sonja and Alex, my parents.

Finally, there is someone whom I haven't seen very much of those last months. Thank you, Eva, for all your patience, love and support.

# CONTENTS

# 1

# INTRODUCTION AND OVERVIEW

## 1.1 INTRODUCTION

A common problem in signal processing is erroneous operator responses because of missing or uncertain data. Perhaps the best known of such effects are the various 'edge effects' which invariably occur at the edges of the input data set. In the standard scalar representation for grey level images, the value zero is commonly used to represent both 'black' and 'outside the image'. This is called black border extension of the signal. This causes most operators to produce undesired edge effects. Common approaches to reduce such effects are *border extraction* where the edge values are used for extrapolation and *wrap around* where a strictly compact signal is viewed as cyclic. Figure 1.1 shows three examples of extending a signal outside its border. Extending the signal by mirroring the signal in different ways can also be used to reduce edge effects. In all these cases, however, new structures are added to the signal and most operators give unpredictable and unwanted responses to these structures.

A representation having similar ambiguity to the standard scalar representation for grey level images is the vector representation for velocity. Consider the following simple example. In an image motion field, velocity and its direction are commonly represented by vectors. Only in regions of the image containing structure is it possible to measure the local image velocity. In a region having constant grey-level value, it is impossible to find motion and thus impossible to measure it. If the velocity vectors are set to zero in these regions, this will cause artifacts such as discontinuities in the motion field. Thus, if velocity is estimated from an image sequence, it cannot be fully described by vectors as a sole description. In an example in chapter 5, it is shown that, using a vector as the sole representation for local velocity, borders between regions of missing data and good data can induce strong erratic responses.
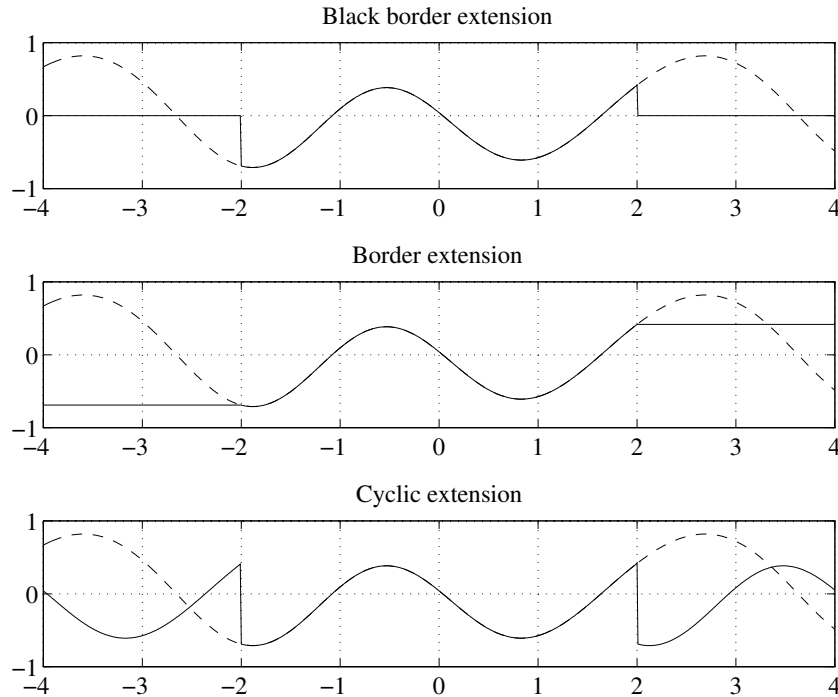
**Figure 1.1**   Examples of commonly used methods for defining the signal border when filtering strictly compact signals.

In this thesis it is shown how unwanted effects due to missing or uncertain data can be significantly reduced or eliminated. The theory is based on linear operations and is general in that it allows for both data and operators to be scalars, vectors or tensors of higher order.

The information representation issue is complex and what is regarded as a good information representation varies with the application. Nevertheless, we stress that an important feature of a good representation is that it keeps "statement" and "certainty of statement" separate. From a philosophical point of view, there should be no argument that "knowing" and "not knowing" are different situations regardless of what is to be known. Such thoughts are by no means new [Mac69, Gra78, GK83, WK88, Knu89] and can, depending on point of view, be said to be related to probability theory, fuzzy set theory, quantum mechanics and evidence calculus. However, it is felt that the vision community would benefit from an increased awareness of the importance of these ideas. The present thesis is intended to be a contribution towards this end.

A data representation which has certainty and statement separated avoids the need to decide the grey-level value outside the image. In practice, estimation or generation of additional certainty information is not unusual. As an example, range data normally consists of two parts; a scalar value defining the distance and an energy measure used to identify points, so called drop-outs, where the range camera has failed to estimate the distance [LRS89].

## 1.2 OVERVIEW

In the next chapter, chapter 2, notations used in this thesis are described. A short introduction to tensor theory and basic notions in functional analysis is given is this chapter.

In chapter 3, non-orthogonal filter bases is discussed using the concepts of contravariant and covariant coordinates introduced in chapter 2. The concept of *frames* is described in this context.

In chapter 4, we extend the discussion of chapter 3 and define *normalized convolution*. It is a general method for filtering of missing or uncertain data. It is shown how such data can be modelled using an additional scalar certainty field. In the missing parts of the signal, the certainty parts are set to zero. A similar certainty field for the operator is denoted the *applicability* function. This field is used for spatial localization of the operator. It is shown that this approach differs from classical windowing where the operator itself is changed to a smoother and more localized version. The chapter is concluded with examples illustrating these theories.

In chapter 5, it is shown how a part of the parameter vector can be calculated. Using this technique we define *normalized differential convolution*. Differentiating sparse fields and model based generation of certainty fields for robot vision are examples of application which are discussed.

Chapter 6 discusses the use of tensors in estimation of local structure and orientation. The tensor representation is shown to be crucial to unambiguous and continuous representation of local orientation in multiple dimensions. In addition to orientation, the tensor representation also conveys the degree and type of local anisotropy. In section 6.4 presents a method for computation of two-dimensional motion vector fields from an image sequence. The magnitudes from a set of spatio-temporal quadrature filters are combined into a tensor description as described in chapter 6. The shape of the tensors describes locally the structure of the spatio-temporal neighbourhood and provides information about local velocity and if true flow or only normal flow is present . It is shown how normal

flow estimates are combined into a true flow using averaging on this tensor field description.

Chapter 7 focuses on various aspects of representation and grouping of information. We begin with an investigation of the uniformity of some standard parameter spaces (also termed parameter mappings). The analysis shows that, to avoid discontinuities, great care should be taken when choosing the parameter space for a particular problem. It is shown that the local strucure tensor introduced in chapter 6 can be decomposed into projection operators which can be used for grouping collinear/coplanar estimates . Based on these results, a new parameter mapping well suited for line extraction, the Möbius strip parameterization , is defined. The method has similarities to the Hough Transform.

Chapter 8 deals with estimation of local spatial frequency and bandwidth. Local frequency is an important concept that provides an indication of the appropriate range of scales for subsequent analysis. One-dimensional and two-dimensional examples of local frequency estimation are given.

# 2
# NOTATIONS AND PRELIMINARIES

Most people are familiar with *orthonormal coordinate systems*. How signals are described in coordinates of such systems and how to recover a signal from the coordinates is common knowledge. The procedures for handling *non-orthonormal coordinate systems* are less familiar and it is the purpose of this chapter to describe the basic tools for non-orthogonal theory.

Well-known concepts from tensor algebra and functional analysis are reviewed: the Einstein summation convention, dual spaces, the metric tensor, Hilbert spaces, Riez representation theorem, etc. Readers familiar with all these basic concepts are recommended to, at least, glance through this chapter since most notations used in this thesis are defined here as the concepts are described. The discussion of non-orthogonal systems is continued in chapter 3, where it is extended to redundant non-orthonormal discrete systems, so-called frames.

Functional and tensor analysis are branches of mathematics that use intuition and the language of geometry in the study of functions. It can be useful to represent a point in space by a triple of numbers, but it can also be advantageously, when dealing with a triplets of numbers, to think of them as coordinates of a point in space. This is called *geometrization of algebra*.

This chapter contains a brief review of the definitions, concepts and conventions used in this thesis. The notations defined are based on notations commonly used in functional analysis, tensor theory and signal analysis [You88, Sto89, Dau92, Mey92]. In particular, this chapter contains a short introduction to tensor theory. More complete introductions can be found in [Sto89, Kay88, You78, Ken77]. Tensor algebra is a *multi-linear* extension of traditional linear algebra, and tensor analysis is a generalization of the notions from vector analysis. The need for such a theory is motivated by the fact that there are many physical quantities of complicated nature that cannot naturally be described or represented by scalars or vectors. Examples are the stress at a point of a solid body due to internal

forces, the deformation of an arbitrary element of volume of an elastic body, and the moments of inertia. These quantities can be described and represented adequately only by the more sophisticated mathematical entities called tensors. As we will see, scalars and vectors also belong to this family of elements. Thus, scalars and vectors are special cases of tensors. The name "tensor" originates from the french word "tension" which happens to be the English word as well.

## 2.1 OBJECTS AND SCALARS

Scalars, typically vector or tensor coordinates, will be denoted using italics,

$$A = B + c$$

Vectors and tensors will be denoted using boldface. Generally, lower case letters will be used for vectors and upper case letters used for tensors of order higher than one,

$$\mathbf{A} = \mathbf{b} \otimes \mathbf{c}$$

When working with matrices and vectors in standard linear algebra notations the difference between vectors, such as objects in a vector space, and its coordinates is not very accentuated

$$\mathbf{u}^T \mathbf{A} \ \mathbf{v} = \mathbf{u}^T \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mathbf{v}$$

When using linear algebra notation, vectors and matrices will be denoted in boldface letters indicating that they are regarded as objects although they effectively are arrays of scalar numbers.

## 2.2 INDEX CONVENTIONS

### 2.2.1 Bases

Any set of $n$ linearly independent vectors in an $n$-dimensional vector space is called a *basis*. Let $\mathcal{V}$ denote a vector space of finite dimension and let $\{\mathbf{v}_i\}$ denote a basis for $\mathcal{V}$:

$$\mathcal{V} \ : \ \{\mathbf{v}_1, \mathbf{v}_2, ........, \mathbf{v}_n\} \tag{2.1}$$

Any vector $\mathbf{x}$ in $\mathcal{V}$ can be expressed as a unique linear combination of the basis vectors

$$\mathbf{x} \; = \; \sum_i x^i \mathbf{v}_i \tag{2.2}$$

where $x^i$ is the coordinates of $\mathbf{x}$ in the basis $\{\mathbf{v}_i\}$. In cases where the vector may be described in more than one basis, an additional subscript inside a parenthesis will be used for indicating which basis the coordinates correspond to:

$$\mathbf{x} \; = \; \sum_i (x_v)^i \mathbf{v}_i \; = \; \sum_i (x_b)^i \mathbf{b}_i \tag{2.3}$$

## 2.2.2 Einstein summation convention

The set over which an index ranges is a subset of integers and is specified when not obvious. For tensors it is convenient to use both subscripts and superscripts for indexing the elements or coordinates. The summation convention introduced by Einstien will be used:

$$x^1 \mathbf{v}_1 \; + \; x^2 \mathbf{v}_2 \; + \; ..... \; + x^n \mathbf{v}_n \; = \; \sum_{i=1}^{n} x^i \mathbf{v}_i \; \equiv \; x^i \mathbf{v}_i \tag{2.4}$$

The summation rule implies that a summation is performed in any expression involving *diagonally* repeated indices.

**Example 1** *A double sum expressed using the Einstien summation convention is written*

$$\sum_{i=1}^{2} \sum_{j=1}^{3} A_{ij} x^i y^j \; \equiv \; A_{ij} x^i y^j \tag{2.5}$$

*where the indicies $i$ and $j$ range over the index sets $\{1, 2\}$ and $\{1, 2, 3\}$ respectively. Note that the summation is applied over diagonally repeated indices.*
□

## 2.3   SCALAR PRODUCT

An *inner product* or *scalar product* on a complex vector space, $\mathcal{V}$, is denoted $\langle \cdot , \cdot \rangle$ where the dots indicate the entries for two vectors $\in \mathcal{V}$. A scalar product on a complex vector space $\mathcal{V}$ is thus a mapping

$$\langle \cdot , \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \qquad \qquad (2.6)$$

### 2.3.1   Scalar product axioms

The usual requirements for a scalar product are that it should be Hermitian and a linear. In the real case, the scalar product is linear in both its arguments, it is a *bi-linear operator*. A common requirement of the scalar product operator is that it should be positive. In the finite dimensional complex case, this corresponds to a conjugate symmetric operator having all eigenvalues positive. Let us summarize the four scalar product axioms:

1. $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$

2. $\langle \lambda \mathbf{u}, \mathbf{v} \rangle = \lambda \langle \mathbf{u}, \mathbf{v} \rangle$

3. $\langle \mathbf{u} + \mathbf{w}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{w}, \mathbf{v} \rangle$

4. $\langle \mathbf{u}, \mathbf{u} \rangle > 0$ when $\mathbf{u} \neq 0$

where $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{V}$ and $\lambda \in$ . $\overline{a}$ denotes complex conjugate of a complex number $a$. The fourth axiom is sometimes written "If $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ for arbitrary $\mathbf{u}$, then $\mathbf{v} = 0$". This definition incorporates so-called semidefinite scalar products allowing for vectors to have negative length and for vectors to have zero length without being the zero vector, as for example in the *Riemann spaces* commonly used in theoretical physics. The requirement (4) above, however, makes the scalar product positive definite and all equidistance surfaces form elipsoides.

### 2.3.2   Scalar product in terms of coordinates

Expanding the vectors $\mathbf{u}$ and $\mathbf{v}$ in the basis $\{\mathbf{b}_i\}$ gives

$$\mathbf{u} = u^i \mathbf{b}_i \quad \text{and} \quad \mathbf{v} = v^i \mathbf{b}_i \qquad \qquad (2.7)$$

where the components $u^i$ and $v^i$ are coordinates of the vector $\mathbf{u}$ and $\mathbf{v}$ in the basis $\mathbf{b}_i$. By the scalar product axioms (1) and (2)

$$\langle \mathbf{u}, \mathbf{v} \rangle = u^i \overline{v}^j \langle \mathbf{b}_i, \mathbf{b}_j \rangle \qquad \qquad (2.8)$$

If the basis vectors, $\mathbf{b}_i$, are *orthonormal*, then

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = \delta_{ij} \tag{2.9}$$

where *Kronecker's symbol* $\delta$ is used with the usual meaning

$$\delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

that is

$$\delta_{ij} = \begin{pmatrix} 1 & 0 & . & 0 \\ 0 & 1 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & 1 \end{pmatrix} \tag{2.10}$$

Inserting this in equation (2.8) gives the well-known formula

$$\langle \mathbf{u}, \mathbf{v} \rangle = u^i \overline{v}^j \delta_{ij} = u^i \overline{v}_i \tag{2.11}$$

stating that the scalar product is the product sum of the coordinates (and a conjugation). As shown here, this formula implicitly requires that the involved basis vectors are *orthonormal*.

## 2.4 HILBERT SPACES

A complex vector space with an inner product is called an *inner product space* or a *pre-Hilbert space*. A *Hilbert space* is an inner product space which is also complete.

### 2.4.1 The norm of a vector

A common way of defining the *norm* of a vector is via the scalar product. The reason for this is that calulations become easy with such a norm. We define the norm of a vector by

$$\|\mathbf{f}\|^2 = \langle \mathbf{f}, \mathbf{f} \rangle \tag{2.12}$$

## 2.4.2 The norm of an operator

A common *operator norm* is the *max norm*. The max norm of an operator $\mathbf{A}$ is defined as the maximum amplification of a vector that the operator can achieve:

$$\|\mathbf{A}\| \equiv \sup_{\mathbf{v} \neq 0} \frac{\|\mathbf{A}\mathbf{v}\|}{\|\mathbf{v}\|} = \beta \tag{2.13}$$

In the finite-dimensional case, the value $\beta$ is equal to the largest eigenvalue of the operator. Similarly a lower bound of $\mathbf{A}$ is defined by

$$\inf_{\mathbf{v} \neq 0} \frac{\|\mathbf{A}\mathbf{v}\|}{\|\mathbf{v}\|} = \alpha \tag{2.14}$$

where $\alpha$ in the finite-dimensional case corresponds to the magnitude of smallest eigenvalue.

## 2.4.3 The Frobenius norm

In the finite dimensional case, operators can be represented by arrays of numbers related to a chosen basis. Unfolding the array into a one-dimensional array of numbers allows the operator to be associated with a vector and the use of equation (2.12) for defining a norm of the operator. This matrix norm is called the *Frobenius norm* and is denoted $\| \cdot \|_F$. An alternative way of calculating the Frobenius norm is to use the *trace operator*,

$$\|\mathbf{A}\|_F = \text{trace}(\mathbf{A}\mathbf{A}^*) \tag{2.15}$$

## 2.4.4 A bounded positive operator

An operator $\mathbf{A}$ is said to be *positive definite* if there exists a positive constant $k$ such that

$$\langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle \geq k\langle \mathbf{x}, \mathbf{x} \rangle \quad \text{for all } \mathbf{x} \in \mathcal{V} \tag{2.16}$$

which means that the constant $\alpha$ in equation (2.14) is strictly positive. If the max norm is bounded, $\beta < \infty$, the operator is said to be a *bounded operator*. If a positive definite operator is bounded it is denoted a *bounded positive operator*. In

the finite-dimensional case this corresponds to all eigenvalues $\lambda_i$ being bounded and greater than zero,

$$0 < \alpha \leq \lambda_i \leq \beta < \infty \tag{2.17}$$

A bounded positive operator $\mathbf{A}$ is invertible. If $\mathbf{A}$ has lower bound $\alpha$, its inverse $\mathbf{A}^{-1}$ is bounded by $\frac{1}{\alpha}$.

## 2.4.5    The adjoint operator

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces and let $\mathbf{A}$ denote a bounded operator $\mathcal{H}_1 \to \mathcal{H}_2$ (which may be equal to the first one). The adjoint of $\mathbf{A}$, denoted $\mathbf{A}^*$, is then uniquely defined by

$$\langle \mathbf{v}_1, \mathbf{A}\mathbf{v}_2 \rangle \;=\; \langle \mathbf{A}^\star \mathbf{v}_1, \mathbf{v}_2 \rangle \tag{2.18}$$

which should hold for all $\mathbf{v}_1 \in \mathcal{H}_1$ and $\mathbf{v}_2 \in \mathcal{H}_2$. The norms of these operators are equal.

$$\|\mathbf{A}\| = \|\mathbf{A}^\star\| \tag{2.19}$$

In a finite-dimensional Hilbert space, a linear operator may be represented using a $n \times m$ matrix and the adjoint operator is equal to the conjugate transpose of this matrix. If $\mathbf{A}^\star = \mathbf{A}$, then $\mathbf{A}$ is called *self-adjoint* or *Hermitian*.

## 2.4.6    Common Hilbert spaces

Commonly used Hilbert spaces in literature have special notations. The set of all square integrable functions on $^N$ or $^N$ respectively form Hilbert spaces and are denoted

$$\mathcal{L}^2 \;=\; \{f : {}^N \to \; : \|f\| < \infty\} \tag{2.21}$$

$$\ell^2 \;=\; \{f : {}^N \to \; : \|f\| < \infty\} \tag{2.22}$$

in the continuous and the discrete case respectively.

## 2.5   THE DUAL SPACE AND COVECTORS

For a complex vector space $\mathcal{V}$, the linear operators that map vectors to complex numbers are important. These operators form a new vector space of the same dimension as $\mathcal{V}$, called the *dual* or the *reciprocal* space. The dual space corresponding to $\mathcal{V}$ is denoted $\mathcal{V}^*$.

In $\mathcal{V}$, the elements are called *contravariant vectors*, and the elements in $\mathcal{V}^*$ are called *covariant vectors* or shorter, *covectors*.

$$\mathbf{v}^i : \mathcal{V} \;\rightarrow\; \tag{2.22}$$

In functional analysis these elements are called *linear functionals*. Covectors are indexed by *superscripts* and contravariant vectors are indexed by *subscripts*. The dual of the dual space is in the finite dimensional equal to the space we started with, $(\mathcal{V}^*)^* = \mathcal{V}$, i.e. every $\mathbf{x} \in \mathcal{V}$ is a functional on $\mathcal{V}^*$.

Given a basis, its *dual basis* is uniquely defined via the following relation:

$$\mathbf{v}^i \mathbf{v}_j \;=\; \delta^i_j \tag{2.23}$$

In Hilbert spaces, Riesz representation theorem [You88] states that all elements in $\mathcal{V}^*$ are *uniquely* associated with elements in $\mathcal{V}$.

**Theorem 1 (Riesz)** *Let $\mathcal{H}$ be a Hilbert space and let $\mathbf{b}^i \in \mathcal{H}^*$ be a continuous linear functional on $\mathcal{H}$. There exists a unique $\mathbf{b}_i \in \mathcal{H}$ such that*

$$\mathbf{b}^i(\mathbf{x}) = \langle \mathbf{b}_i, \mathbf{x} \rangle$$

The conclusion is that every covector $\mathbf{b}^i$ can be represented with a contravariant vector which here also is denoted $\mathbf{b}^i$. Given a basis, its dual basis is uniquely defined via the scalar product.

$$\langle \mathbf{v}^i, \mathbf{v}_j \rangle \;=\; \delta^i_j \tag{2.24}$$

**Example 2** *If $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is a basis for $\mathcal{V}$, its dual basis is denoted by*

$$\{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3\} \tag{2.25}$$

$\square$

Riesz proved this theorem for the infinite-dimensional case. In the simpler finite-dimensional case, the theorem implies that if we have a representation of a vector $\mathbf{v} = v^i \mathbf{b}_i$, there exists another closely related representation given by

$$\mathbf{v} \;=\; v^i \mathbf{b}_i \;=\; v_i \mathbf{b}^i \tag{2.26}$$

where the components $v^i$ are called the *contravariant coordinates* of $\mathbf{v}$, and $v_i$ are the *covariant coordinates*.

An example in two dimensions of a basis and its dual basis is shown in figure 2.1.
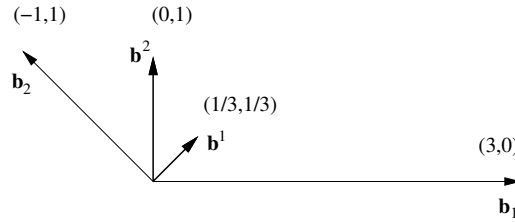


**Figure 2.1** A basis $\{\mathbf{b}_1, \mathbf{b}_2\}$ and its dual basis $\{\mathbf{b}^1, \mathbf{b}^2\}$. Note that the vector $\mathbf{b}_1$ is orthogonal to $\mathbf{b}^2$ and that $\mathbf{b}_2$ is orthogonal to $\mathbf{b}^1$ (equation (2.24)).

## 2.5.1 Contravariant and covariant coordinates

The *contravariant* coordinates of a vector $\mathbf{x}$ related to a basis $\mathbf{b}_i$ are denoted $(x_b)^i$ and may be expressed as the scalar product between $\mathbf{x}$ and the dual *covariant basis* vectors $\mathbf{b}^i$,

$$\langle \mathbf{x}, \mathbf{b}^i \rangle \;=\; \langle (x_b)^j \mathbf{b}_j, \mathbf{b}^i \rangle \tag{2.28}$$

$$=\; (x_b)^j \langle \mathbf{b}_j, \mathbf{b}^i \rangle \tag{2.29}$$

$$=\; (x_b)^j \delta_j{}^i \tag{2.30}$$

$$=\; (x_b)^i \tag{2.31}$$

And, similarly, for the *covariant* coordinates of a vector $\mathbf{x}$, $(x_b)^i$, which may be expressed as the scalar product between $\mathbf{x}$ and the *contravariant basis* vectors

$$\langle \mathbf{x}, \mathbf{b}_i \rangle \;=\; (x_b)_i \tag{2.31}$$

This important observation can be used for expressing covariant coordinates in terms of contravariant coordinates and vice versa:

$$(x_b)_i \;=\; \langle \mathbf{x}, \mathbf{b}_i \rangle \;=\; \langle (x_b)^j \mathbf{b}_j, \mathbf{b}_i \rangle \;=\; (x_b)^j \langle \mathbf{b}_j, \mathbf{b}_i \rangle \tag{2.33}$$

$$(x_b)^j \;=\; \langle \mathbf{x}, \mathbf{b}^j \rangle \;=\; \langle (x_b)_i \mathbf{b}^i, \mathbf{b}^j \rangle \;=\; (x_b)_j \langle \mathbf{b}^i, \mathbf{b}^j \rangle \tag{2.34}$$

An important observation here is that there exist matrices $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$ and $\langle \mathbf{b}^i, \mathbf{b}^j \rangle$ that transform coordinates between the contravariant and the covariant versions thereof.

## 2.6   TENSORS

Scalars and vectors are only simple examples from the class of quantities in the field of applied mathematics. There are quantities of more complicated structure than scalars or vectors, called *tensors*. The concept of a tensor, like that of a vector, is an entity that does not depend on any frame of reference or coordinate system. However, just as a vector can be represented by its components when referred to a particular coordinate system, so can a tensor.

There are two avenues to tensors, and there is a general disagreement over which is the better approach; the *component approach* or the *object approach*. In the component approach the underlying coordinate system is only implicit. This has the disadvantage that all the components describing the tensor change under transformation, although the tensor still is the "same". The other approach, the object approach, is favoured by the mathematical community. Although this way of treating tensors is necessary for many modern applications and may give a more complete understanding for tensor theory, the component approach is easier to begin with. The component approach also has the advantage of facilitating interpretation of the calculations in terms of linear algebra.

Tensors are based on two forms. Depending on the transformation properties of a tensor, it will be categorized as being a *covariant tensor*, a *contravariant tensor* or a mixture of these two, i.e. a *mixed tensor*. In section 2.5 the distinction between vectors and covectors was discussed; a vector is an example of a contravariant tensor and a covector of a covariant tensor. An electrical field or the gradient of a scalar field are examples of covectors. A vector is expressed in units of the coordinate grid, while an electrical field is expressed in voltage *per* unit of the coordinate grid.

Consider a change of coordinate system resulting in doubling the value of each vector component, i.e. a rescaling of the axes. What does this bring about for

the electrical field description? The electrical field expressed in its components will in the new basis have *half* of the original component values. The general rule is: contravariant vectors have the transformation property that a change to shorter basis vectors gives larger coordinates, while the same change gives smaller coordinates for covariant vectors. Another example of this is presented in figure 2.2.
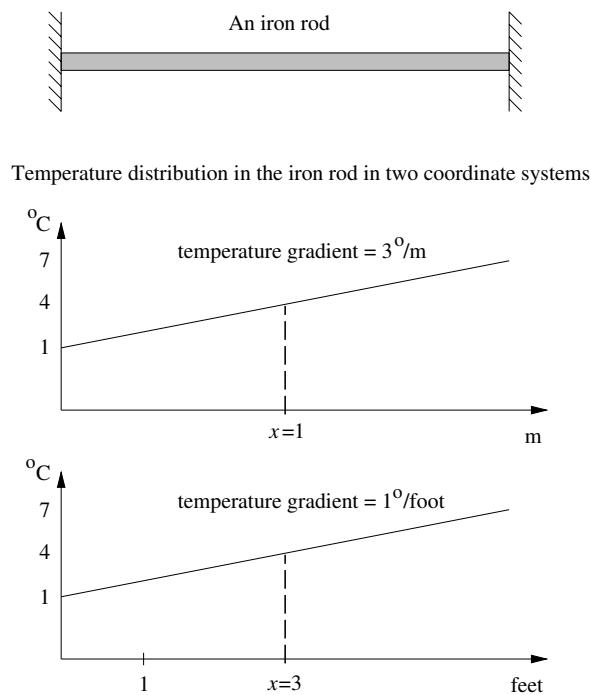


**Figure 2.2**   The position vector is a *contravariant* vector: Changing to a shorter basis vector, from 1m to 1foot, increases the position coordinate from 1 to 3 (for simplicity 1m=3feet). The gradient is a *covariant* vector: Changing to a shorter basis vector, from 1m to 1foot, decreases the coordinate of the gradient from 3 to 1.

## 2.6.1   Definition of finite-dimensional tensors

The definition of higher order tensors proceeds by making use of the spaces $\mathcal{V}$ and $\mathcal{V}^*$ and defines *multi-linear functions* $\mathbf{F}(\mathbf{u}, \mathbf{v}, ..., \mathbf{w})$ in which the elements $\mathbf{u}, \mathbf{v}, ..., \mathbf{w}$ are vectors that range independently over $\mathcal{V}$ or $\mathcal{V}^*$. The term multi-

linear means that a function is linear in each of its arguments. A tensor of order $(p, q)$ is a multi-linear mapping:

$$\underbrace{\mathcal{V}^* \times \mathcal{V}^* ..... \times \mathcal{V}^*}_{p} \underbrace{\times \mathcal{V} \times \mathcal{V} ..... \times \mathcal{V}}_{q} \; \rightarrow \qquad\qquad (2.34)$$

The *order* of a tensor defines the number of subscripts and superscripts needed in an element description. A tensor

$$F_{i_1..i_p}{}^{j_1..j_q} \qquad\qquad (2.35)$$

is of order $(p, q)$. A tensor having both covariant and contravariant components is said to be a mixed tensor.

From this definition we see that a vector is a tensor of order $(1, 0)$ and a covector is $(0, 1)$. Tensors of order $(2, 0)$, $(0, 2)$ and $(1, 1)$ can in the finite-dimensional case be represented with matrices. Note that the order of a tensor has nothing to do with the size of the array, i.e. if it is $2 \times 2$ or $3 \times 3$, etc. Third order tensors, for example tensors of order $(3, 0)$, have one more index and can therefore be interpreted as a three-dimensional array of numbers (in the finite-dimensional case).

**Example 3** *A second order tensor of type (2,0)* $\in \mathcal{V} \otimes \mathcal{V}$*. This space,* $\mathcal{V} \otimes \mathcal{V}$*, consists of all multi-linear mappings:* $\mathcal{V}^* \times \mathcal{V}^* \; \rightarrow$ $\qquad\qquad\qquad$ $\square$

**Example 4** *A second order tensor of type (1,1)* $\in \mathcal{V} \otimes \mathcal{V}^*$*. This space,* $\mathcal{V} \otimes \mathcal{V}^*$*, consists of all multi-linear mappings:* $\mathcal{V}^* \times \mathcal{V} \; \rightarrow$ $\qquad\qquad\qquad$ $\square$

## 2.6.2   The metric tensor

We have earlier defined the scalar product. It takes two vectors and produces a real number, i.e. it is a multilinear mapping $\mathcal{V} \times \mathcal{V}$. This means that the scalar product corresponds to a $(0, 2)$-tensor. This tensor is called the *metric tensor* or the *first fundamental tensor* and is denoted $\mathbf{G}$. Depending on in which basis it is descibed, it has different coordinates:

$$\mathbf{G} \;=\; (G_b)_{ij} \mathbf{b}^i \otimes \mathbf{b}^j \qquad \mathbf{G} \;=\; (G_b)^{ij} \mathbf{b}_i \otimes \mathbf{b}_j$$

$$\mathbf{G} \;=\; (G_b)^i{}_j \mathbf{b}_i \otimes \mathbf{b}^j \qquad \mathbf{G} \;=\; (G_b)_i{}^j \mathbf{b}^i \otimes \mathbf{b}_j$$

The coordinates are defined by

$$(G_b)_{ij} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle \tag{2.37}$$

$$(G_b)^{ij} = \langle \mathbf{b}^i, \mathbf{b}^j \rangle \tag{2.38}$$

$$(G_b)_i{}^j = \langle \mathbf{b}_i, \mathbf{b}^j \rangle = \delta_i^j \tag{2.39}$$

$$(G_b)^j{}_i = \langle \mathbf{b}^j, \mathbf{b}_i \rangle = \delta_j^i \tag{2.40}$$

Inserting these notations in equations (2.32) and (2.33) gives the two first equations below. The two last equations are given by symmetry.

$$(x_b)_i = (G_b)_{ij}(x_b)^i \tag{2.41}$$

$$(x_b)^i = (G_b)^{ij}(x_b)_i \tag{2.42}$$

$$(x_b)^j = (G_b)^i{}_j(x_b)^i \tag{2.43}$$

$$(x_b)_i = (G_b)_j{}^i(x_b)_j \tag{2.44}$$

The coordinates $(G_b)_{ij}, (G_b)^{ij}, (G_b)^i{}_j = (G_b)_j{}^i = \delta_j^i$ are covariant, contravariant and mixed coordinates respectively of the metric tensor.

As in the vector case, when multiple bases are used to describe a tensor, a subscript and a parenthesis are used for indicating in which basis the metric tensor is described:

$$\mathbf{G} = (G_v)_{ij}\mathbf{v}^i \otimes \mathbf{v}^j = (G_b)_{ij}\mathbf{b}^i \otimes \mathbf{b}^j \tag{2.44}$$

The *bracket notation* and the metric tensor closely related.

$$\langle \cdot, \cdot \rangle \quad \leftrightarrow \quad \mathbf{G}(\cdot, \cdot) \tag{2.45}$$

The bracket notation will sometimes be used in parallel since it may simplify the interpretation for readers not familiar with tensor algebra.

The metric tensor is a tensor of order (0,2) which means that it is a *bilinear mapping* from $V \times V \to$ . As indicated in section 2.5.1, this tensor can be used to transform vectors to covectors. Inserting one basis vector gives

$$\mathbf{G}(\mathbf{v}_i, \cdot) = ((G_v)_{kj}\mathbf{v}^k \otimes \mathbf{v}^j)(\mathbf{v}_i, \cdot)$$

$$= (G_v)_{kj} \underbrace{\mathbf{v}^k(\mathbf{v}_i)}_{=\delta^k{}_i} \mathbf{v}^j \tag{2.47}$$

$$= (G_v)_{ij}\mathbf{v}^j \tag{2.48}$$

and if the basis $\{v_i\}$ is orthonormal, then $(G_v)_{ij} = \delta_{ij}$ giving

$$\mathbf{G}(\mathbf{v}_i, \cdot) = \delta_{ij}\mathbf{v}^j = \mathbf{v}^i \tag{2.48}$$

The coordinates of a metric tensor, corresponding to the covariant basis, are obtained by having the metric act on all corresponding pairs of contravariant basis vectors.

$$\begin{aligned}
\mathbf{G}(\mathbf{v}_i, \mathbf{v}_j) &= ((G_v)_{kl}\mathbf{v}^k \otimes \mathbf{v}^l)(\mathbf{v}_i, \mathbf{v}_j) \\
&= (G_v)_{kl}\mathbf{v}^k(\mathbf{v}_i)\mathbf{v}^l(\mathbf{v}_j) \\
&= (G_v)_{kl}\, \delta_i^k\, \delta_j^l \\
&= (G_v)_{ij}
\end{aligned} \tag{2.50}$$

where $\mathbf{v}^i \otimes \mathbf{v}^j$ are the covariant basis elements in which the metric is described. More explicitly,

$$(G_v)_{ij} = \begin{pmatrix}
\mathbf{G}(\mathbf{v}_1, \mathbf{v}_1) & \mathbf{G}(\mathbf{v}_1, \mathbf{v}_2) & . & \mathbf{G}(\mathbf{v}_1, \mathbf{v}_i) \\
\mathbf{G}(\mathbf{v}_2, \mathbf{v}_1) & \mathbf{G}(\mathbf{v}_2, \mathbf{v}_2) & . & \mathbf{G}(\mathbf{v}_2, \mathbf{v}_i) \\
. & . & . & . \\
\mathbf{G}(\mathbf{v}_i, \mathbf{v}_1) & \mathbf{G}(\mathbf{v}_i, \mathbf{v}_2) & . & \mathbf{G}(\mathbf{v}_i, \mathbf{v}_i)
\end{pmatrix} \tag{2.50}$$

If the basis vectors, $\mathbf{v}_i$, are orthogonal, the coordinates of the metric is defined by

$$(G_v)_{ij} = \begin{pmatrix}
\mathbf{G}(\mathbf{v}_1, \mathbf{v}_1) & 0 & . & 0 \\
0 & \mathbf{G}(\mathbf{v}_2, \mathbf{v}_2) & . & 0 \\
. & . & . & . \\
0 & 0 & . & \mathbf{G}(\mathbf{v}_i, \mathbf{v}_i)
\end{pmatrix} \tag{2.51}$$

and if the system is orthonormal, the coordinates of the metric are reduced to

$$(G_v)_{ij} = \begin{pmatrix}
1 & 0 & . & 0 \\
0 & 1 & . & 0 \\
. & . & . & . \\
0 & 0 & . & 1
\end{pmatrix} = \delta_{ij} \tag{2.52}$$

**Example 5** *The standard metric in* $\mathbb{R}^3$ *when using a Cartesian orthonormal basis has the following components:*

$$G_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.53}$$

*and the scalar product between two vectors in* $\mathbb{R}^3$ *in matrix notation is denoted:*

$$G_{ij}a^i b^i = \begin{pmatrix} a^1 & a^2 & a^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b^1 \\ b^2 \\ b^3 \end{pmatrix} = a^1 b^1 + a^2 b^2 + a^3 b^3 \tag{2.54}$$

$\square$

## 2.7 CHANGE OF BASIS

### 2.7.1 Transformation of vectors

Let $\{\mathbf{v}_i\}$ be a basis for $\mathcal{V}$. Any vector $\in \mathcal{V}$ can be described as a sum of coordinates times the corresponding basis vectors, $\mathbf{b} = (b_v)^i \mathbf{v}_i$. A set of vectors, $\mathbf{b}_n$, may be described by adding a subscript $n$,

$$\mathbf{b}_n = (b_v)^i{}_n \mathbf{v}_i \tag{2.55}$$

If the vectors in the set $\mathbf{b}_n$ are linearly independent and span the same space as $\{\mathbf{e}_n\}$, they, too, constitute a basis for $\mathcal{V}$. This illustrates that the *coordinates* of the new basis functions expressed in the old basis define a linear transformation matrix $(b_v)^i_n$. Expressing a vector in both systems

$$\mathbf{x} = (x_v)^i \mathbf{v}_i \tag{2.57}$$

$$= (x_b)^n \mathbf{b}_n \tag{2.58}$$

$$= (x_b)^n (b_v)^i{}_n \mathbf{v}_i \tag{2.59}$$

gives $((x_v)^i - (x_b)^n (b_v)^i{}_n)\mathbf{v}_i = 0$. Since all the basis vectors are linearly independent, the expression inside the parenthesis vanishes and we get

$$(x_v)^i = (b_v)^i{}_n (x_b)^n \tag{2.59}$$

## 2.7.2   Transformation of the metric tensor

Equation (2.49) gives an expression for the coordinates of the metric tensor in the new basis $\mathbf{b}^i \otimes \mathbf{b}^j$:

$$
\begin{aligned}
(G_b)_{nm} &= \mathbf{G}(\mathbf{b}_n, \mathbf{b}_m) & (2.61)\\
&= ((G_b)_{kl}\mathbf{b}^k \otimes \mathbf{b}^l)(\mathbf{b}_n, \mathbf{b}_m)\\
&= ((G_v)_{kl}\mathbf{v}^k \otimes \mathbf{v}^l)((b_v)^i_n \mathbf{v}_i, (b_v)^j_m \mathbf{v}_j)\\
&= (G_v)_{kl}\ (b_v)^i_n (b_v)^j_m\ \mathbf{v}^k(\mathbf{v}_i)\mathbf{v}^l(\mathbf{v}_j)\\
&= (G_v)_{kl}\ (b_v)^i_n (b_v)^j_m\ \delta^k_i\ \delta^l_j\\
&= (G_v)_{ij}\ (b_v)^i_n (b_v)^j_m & (2.62)
\end{aligned}
$$

which gives the new coordinates of metric tensor expressed in the coordinates of the old basis:

$$(G_b)_{nm} = (b_v)^i_n (b_v)^j_m (G_v)_{ij} \tag{2.62}$$

## 2.8   PROJECTION OPERATORS

A Hermitian operator $\mathbf{A}$ with eigenvalues $\lambda_i$ can be decomposed using the spectral decomposition theorem [Str80],

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^* \tag{2.63}$$

where the vectors $\hat{\mathbf{e}}_i$ are normalized eigenvectors of $\mathbf{A}$.

**Definition 1** *A linear operator $\mathbf{A}$ is said to be a projection operator if is is both Hermitian and* idempotent, *where the last property is defined by* $\mathbf{A}\mathbf{A} = \mathbf{A}$.   □

From the idempotent requirement it follows that a projection operator has eigenvalues $\{0, 1\}$. It is easy to verify that all the outer products of the normalized eigenvectors are projection operators since

$$(\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T)^T \;=\; \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \tag{2.64}$$

and the operator is idempotent:

$$\mathbf{P} \;=\; \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \qquad \text{and} \qquad \mathbf{P}\mathbf{P} \;=\; \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \;=\; \hat{\mathbf{e}}_i \underbrace{(\hat{\mathbf{e}}_i^T \hat{\mathbf{e}}_i)}_{=1} \hat{\mathbf{e}}_i^T \;=\; \mathbf{P} \tag{2.65}$$

it follows from definition 2.1 that $\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T$ is a projection operator. This class of operators is commonly used in quantum mechanics [Jau68, Hug89].

# 3

# NON-ORTHOGONAL BASES AND FRAMES

*Non-orthonormal basis* systems are rarely discussed in literature. Lately, however, the increasing interest in *wavelet theory* has brought some light to the theory needed for non-orthogonal systems. However, wavelet discussions are often reduced to concern only orthonormal systems. Rather than reviewing wavelet theory in detail, this chapter describes filtering in terms of tensor theory. The reason is that there are many attributes associated with wavelets. For example, low order moments of the wavelet is zero, a *wavelet basis* is constructed as a shifted and dilated version of a "mother-wavelet" and wavelet descriptions are often considered to be redundant. In the following discussion, the basis functions are neither required to be dilated and shifted versions of each other, nor having zero low order moments. Further, no orthogonality requirements of the functions are imposed. Note that non-orthogonality does not imply redundancy.

Without loss of generality, this discussion will be restricted to functions in $\mathcal{L}^2(\;^N)$. Most functions of interest belong to this class. In principle we are interested in all functions in this Hilbert space, and any basis spanning it is infinite. However, in practise the basis set is always finite.

## 3.1   THE BASIS OPERATOR

A set of basis filters can compactly be defined using an operator formalism.

**Definition 2** *Let $\{\mathbf{b}_i\}$ be a basis for $\mathcal{H}$, then the operator*

$$\mathbf{B} = \begin{pmatrix} | & | & ... & | \\ \mathbf{b}_1 & \mathbf{b}_2 & ... & \mathbf{b}_m \\ | & | & ... & | \end{pmatrix} \tag{3.1}$$

*is denoted the* basis operator. □

The chosen basis functions span a subspace of $\mathcal{L}^2(\ ^N)$ and defines which part, denoted $\mathcal{B}$, that is of interest. Even if the basis is carefully chosen, functions outside $\mathcal{B}$ will be present. A typical example when this occurs is in the presence of noise. When dealing with a function outside the subspace, $\mathcal{B}$, it is preferable to describe it with the closest function inside $\mathcal{B}$. Defining what close means is done by choosing a suitable norm. One natural norm in $\mathcal{L}^2(\ ^N)$ is the one related to the scalar product $\| \cdot \|_2^2 = \langle \cdot, \cdot \rangle$. For the sake of simplicity we omit the subscript below.

Let $\theta$ denote the coordinate vector describing the function in the basis $\{\mathbf{b}_i\}$. Finding the best approximation by minimizing the difference between the function $\mathbf{f}$ and $\mathbf{B}\,\theta$ in this norm is done by differentiating the squared error norm,

$$\epsilon \;=\; \|\mathbf{f} - \mathbf{B}\,\theta\|^2 \tag{3.3}$$

$$\;=\; \|\mathbf{f}\|^2 - \mathbf{f}^*\mathbf{B}\,\theta - \theta^*\mathbf{B}^*\mathbf{f} + \theta^*\mathbf{B}^*\mathbf{B}\theta \tag{3.4}$$

The minimum value is given by solving

$$\frac{\partial \epsilon}{\partial \theta} \;=\; -2\mathbf{B}^*\mathbf{f} + 2\mathbf{B}^*\mathbf{B}\theta \;=\; 0 \tag{3.4}$$

giving

$$\mathbf{B}^*\mathbf{f} \;=\; \mathbf{B}^*\mathbf{B}\theta \tag{3.5}$$

If $\mathbf{B}$ is a basis operator with $m$ columns, the operator $(\mathbf{B}^*\mathbf{B})$ has the size $m \times m$ and has been constructed using $m$ linearly independent basis functions. This ensures that the matrix $(\mathbf{B}^*\mathbf{B})$ is invertible and a closed expression of the coordinate vector can be calculated,

$$(\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*\mathbf{f} \;=\; \theta \tag{3.6}$$

The expression $(\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^* = \mathbf{B}^+$ is known as *More-Penrose inverse* of $\mathbf{B}$. Acting with the basis operator on the coordinate vector gives the input signal, or more correct, the part of the signal that can be described by the chosen basis.

$$\mathbf{f}' \;=\; \mathbf{B}\theta \;=\; \mathbf{B}(\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*\mathbf{f} \;=\; \mathbf{B}\mathbf{B}^+\mathbf{f} \tag{3.7}$$

where $\mathbf{P}_B = \mathbf{B}(\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^* \;=\; \mathbf{B}\mathbf{B}^+$ is said to be a projection operator,

$$\mathbf{f}' \;=\; \mathbf{P}_B\mathbf{f} \tag{3.8}$$

This means that the orthogonal projection of a signal $\mathbf{f}$ onto the subspace $\mathcal{B}$ gives an approximation $\mathbf{f}'$ which is as close as possible in least squares sense. Expressing a signal in basis functions spanning a subspace of the total signal space is visualized in figure 3.1.
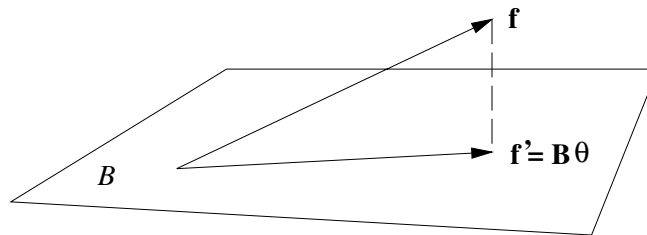


**Figure 3.1** Expressing a signal using a basis spanning a subspace if the total signal space.

## 3.2 SAMPLED SYSTEMS

In the finite-dimensional case, where $\{\mathbf{b}_i\} \in \ell^2(\ ^N)$, the filter functions can be described by $n$-dimensional vectors, where $n$ is the number of samples defining a filter. Then, the basis operator can be represented by a $n \times m$ matrix having these $n$-dimensional basis vectors as column vectors:

$$\mathbf{B} = \begin{pmatrix} . & . & ... & . \\ . & . & ... & . \\ \mathbf{b}_1 & \mathbf{b}_2 & ... & \mathbf{b}_m \\ . & . & ... & . \\ . & . & ... & . \end{pmatrix} \tag{3.9}$$

In the case of finite-dimensional basis operators, the number of rows have to be equal or greater than the number of columns (= number of basis functions). In other words, $n \geq m$. If this requirement is not fulfilled, the number of basis functions, $m$, is greater than the dimension of the space in which the filters are described. This means that the basis set is linearly dependent and thus not a basis. Such *redundant discrete systems* are called *frames*. In the frame case $(\mathbf{B}^*\mathbf{B})$ is not invertible. This issue will be touched upon in section 3.4.

The original pixel description can be interpreted as defining a multidimensional signal space. We denote this space $E$, and a basis spanning this space is denoted $\{\mathbf{e}_i\}$ This space is visualized in figure 3.2 (top). A coordinate set times the basis functions corresponding to a linear grey-level ramp is also shown in this figure (bottom).
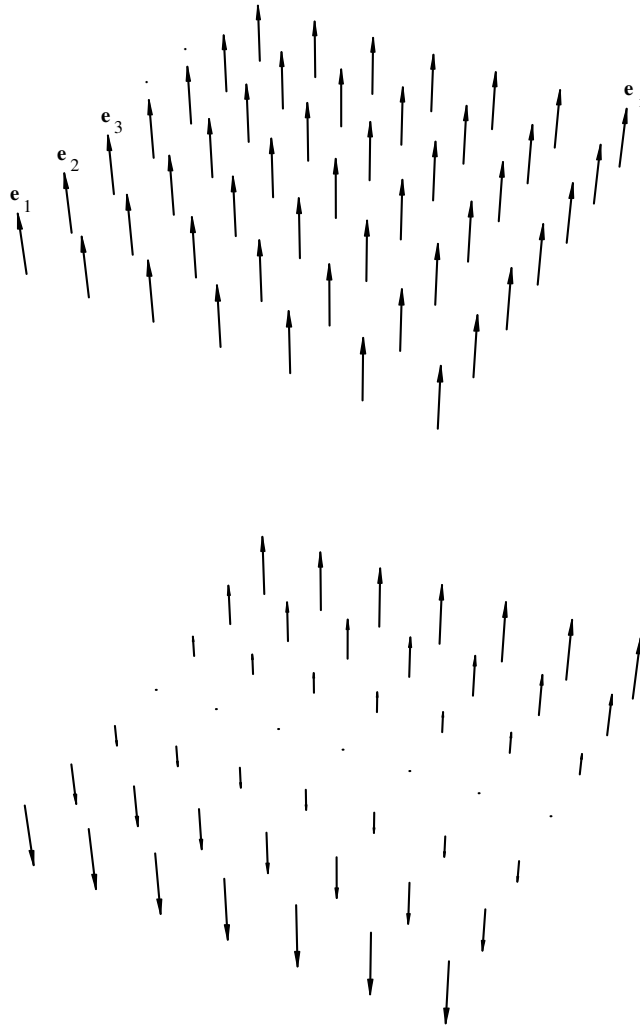
**Figure 3.2** **Top**: The standard pixel base can be seen as a set of Dirac basis vectors on a Cartesian grid. **Bottom**: The pixel base times coordinates describing a linear grey-level ramp.

## 3.2.1   Subspace of orthogonal basis vectors

In this section an explicit example is given of how transformations between orthogonal bases affect the coordinates of the vectors and the metric tensor. The purpose of this rather simple exercise is not only a way to make the reader familiar to the notation, but the results from this section will be compared to a similar derivation in the next section dealing with non-orthogonal basis vectors. In this section a two-dimensional ramp function similar to the on in figure 3.2 will be expressed in two different bases: one nine-dimensional ($3 \times 3$) orthonormal dirac bases denoted $\{\mathbf{e}_i\}$ and a two-dimensional orthonormal basis denoted $\{\mathbf{b}_i\}$.

Let $\mathcal{V}$ be a vector space and let $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_9\}$ be a basis for $\mathcal{V}$. Suppose a vector $\mathbf{f}$ is defined by

$$\mathbf{f} = (f_e)^i \mathbf{e}_i \qquad \leftrightarrow \qquad (f_e)^i = \begin{pmatrix} -3 & 0 & 3 \\ -3 & 0 & 3 \\ -3 & 0 & 3 \end{pmatrix} \tag{3.10}$$

In a product sum, the location and the ordering of the elements is not of importance and viewing the certainty function as a vector reduces the number of indices.

$$(f_e)^i = \begin{pmatrix} -3 & 0 & 3 & -3 & 0 & 3 & -3 & 0 & 3 \end{pmatrix} \tag{3.11}$$

The nine signal coordinates $(f_e)^i$ are spatially arranged in a two-dimensional array indicating that it represents a ramp to the left. A subspace of $\mathcal{V}$ containing this ramp function is the one spanned by the two-dimensional orthonormal basis $\mathbf{b}_n = (b_e)^i_n \mathbf{e}_i$,

$$(\mathbf{b}_1, \mathbf{b}_2) \quad \leftrightarrow \quad (b_e)^i_1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad (b_e)^i_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{3.12}$$

where $\mathbf{b}_1$ is a ramp function to the left and $\mathbf{b}_2$ is a ramp orthogonal to the first one. Let the subspace spanned by $\{\mathbf{b}_i\}$ be denoted $\mathcal{B}$. It is easy to see that $\mathcal{B}$ contains the signal vector $\mathbf{f}$ and thus can be described in this new basis since $\mathbf{f} = 3\mathbf{b}_1$ (figure 3.3).

What is the coordinates of $\mathbf{f}$ in the new basis $\mathbf{b}_i$? Expressing the vector in both the $\mathbf{b}_n$ and the $\mathbf{e}_i$ bases gives

$$\mathbf{f} = (f_b)^n \mathbf{b}_n = (f_e)^i \mathbf{e}_i \tag{3.13}$$
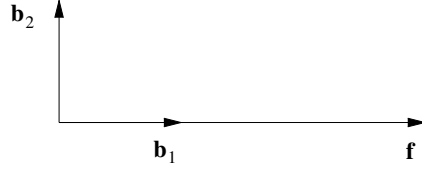
**Figure 3.3** Visualization of the vector **f** (equation (3.10)) and the two basis vectors, $\mathbf{b}_i$ (equation (3.12)). The basis $\mathbf{b}_i$ spans a two-dimensional subspace of $\mathcal{V}$. The signal can exactly be described in $\mathcal{B}$.

From equation (2.59) we get how the old coordinates are expressed in the new ones.

$$(f_e)^i = (b_e)^i_n (f_b)^n \tag{3.14}$$

In order to describe the signal as coordinates in the new basis, the reverse coordinate relation is needed. Unfortunately the coordinate matrix $(b_e)^i_n$ is not a square matrix, and thus not invertible. However, the matrix can be made square by acting with coordinates of the dual basis vectors, $(G_e)_{ij}(b_e)^j_m$, on the left and the right hand of the expression:

$$(G_e)_{ij}(b_e)^j_m (f_e)^i = (G_e)_{ij}(b_e)^j_m (b_e)^i_n (f_b)^n \tag{3.15}$$

Identifying $(G_e)_{ij}(b_e)^i_m (b_e)^i_n$ as the coordinates of the metric in the new basis $\mathbf{b}_n \otimes \mathbf{b}_m$ gives

$$(G_e)_{ij}(b_e)^j_m (f_e)^i = (G_b)_{nm} (f_b)^n \tag{3.16}$$

where $(G_b)^{nm}$ are the coordinates of the inverse metric $\mathbf{G}^{-1}$ where $(G_b)_{nm}(G_b)^{nm} = \delta^n_k$. Thus, the coordinates of the signal in the new basis are

$$(f_b)^n = (G_b)^{nm} \underbrace{(G_e)_{ij}(b_e)^j_m (f_e)^i}_{\text{dual coordinates}} = (G_b)^{nm}(f_b)_m \tag{3.17}$$

Now the coordinates of the metric tensor is needed explicitly. Choosing to view the basis $\mathbf{e}_i$ as orthonormal gives the following coordinates of the metric.

$$(G_e)_{ij} = \mathbf{G}(\mathbf{e}_i, \mathbf{e}_j) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.18}$$

Equation (2.62) gives the coordinates of this metric expressed in the new basis, $\mathbf{b}^i \otimes \mathbf{b}^j$.

$$(G_b)_{nm} = \mathbf{G}(\mathbf{b}_n, \mathbf{b}_m) = (b_v)^i_n (b_v)^j_m (G_v)_{ij} = \tag{3.19}$$

$$= \begin{pmatrix} -1 & -1 \\ 0 & -1 \\ 1 & -1 \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 \\ 0 & -1 \\ 1 & -1 \\ -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \tag{3.20}$$

$$= \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix} \tag{3.21}$$

which gives the coordinates of the corresponding inverse metric (the *conjugate metric tensor*)

$$(G_b)^{nm} = \begin{pmatrix} \frac{1}{6} & 0 \\ 0 & \frac{1}{6} \end{pmatrix} \tag{3.22}$$

Inserting numerical values in $(G_e)_{ij}(b_e)^j_m (f_e)^i$ (equation (3.17)) gives the dual coordinates of the signal. (This part corresponds to "standard filtering".)

$$(f_b)_m = (G_e)_{ij}(b_e)^j_m (f_e)^i = \begin{pmatrix} 18 & 0 \end{pmatrix} \tag{3.23}$$

By acting with the inverse metric the coordinates of the vector in the filter basis

$$(f_b)^n \;=\; (G_b)^{nm}(f_b)_m \;=\; \begin{pmatrix} 18 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{6} & 0 \\ 0 & \frac{1}{6} \end{pmatrix} \;=\; \begin{pmatrix} 3 \\ 0 \end{pmatrix} \tag{3.24}$$

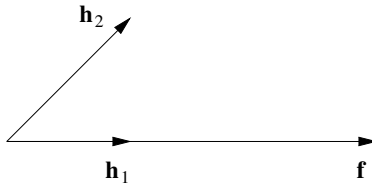which are the coordinates of the signal in the signal space defined by the basis $\mathbf{b}_i$ (figure 3.3).

## 3.2.2   Transformation to a subspace spanned by non-orthogonal basis vectors

Consider the same signal vector  as in the previous section. In this section we will derive its coordinates corresponding to a non-orthogonal basis. Let the new basis vectors be

$$\mathbf{h}_1 \;=\; \mathbf{b}_1 \;=\; (b_e)_1^i \mathbf{e}_i \tag{3.26}$$

$$\mathbf{h}_2 \;=\; \mathbf{b}_1 + \mathbf{b}_2 \;=\; (b_e)_1^i \mathbf{e}_i + (b_e)_2^i \mathbf{e}_i \tag{3.27}$$

The new basis $\{\mathbf{h}_1, \mathbf{h}_2\}$ spans the same space as $\{\mathbf{b}_1, \mathbf{b}_2\}$ but is non-orthogonal. In this section it is shown that the coordinates of the metric tensor compensate for this. A visualization of this basis is shown in figure 3.4



**Figure 3.4**   Visualization of the signal vector  in the $\mathbf{h}_i$ basis. Note that the signal is fully described by a scaling of the $\mathbf{h}_1$ vector although it is not orthogonal to $\mathbf{h}_2$.

The coordinates of the new basis functions are:

$$(h_e)_1^i \;=\; (b_e)_1^i \;=\; \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{3.28}$$

$$(h_e)_2^i = (b_e)_1^i + (b_e)_2^i = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \tag{3.29}$$

As in the previous section, the coordinates of the metric tensor are needed explicitly. The coordinates corresponding to the new basis, $\mathbf{h}^i \otimes \mathbf{h}^j$, are

$$(G_h)_{nm} = \mathbf{G}(\mathbf{h}_n, \mathbf{h}_m) = (h_e)_n^i (h_e)_m^j (G_e)_{ij} = \begin{pmatrix} 6 & 6 \\ 6 & 12 \end{pmatrix} \tag{3.29}$$

and the corresponding inverse metric (the conjugate metric tensor)

$$(G_h)^{nm} = \begin{pmatrix} \frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{6} \end{pmatrix} \tag{3.30}$$

Inserting numerical values gives the dual coordinates of the signal

$$(f_h)_m = (G_e)_{ij}(h_e)_m^j (f_e)^i = \begin{pmatrix} 18 & 18 \end{pmatrix} \tag{3.31}$$

This shows that the projection of the signal is equal on the two new basis functions. Transforming the dual coordinates to standard coordinates using the inverse metric can be seen as an orthogonalization of the basis vectors.

$$(f_h)^n = (G_h)^{nm}(f_h)_m = \begin{pmatrix} 18 & 18 \end{pmatrix} \begin{pmatrix} \frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{6} \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \tag{3.32}$$

which are the coordinates of the signal in the signal space defined by the basis $\mathbf{h}_i$ (figure 3.4).

## 3.3 CONVOLUTION

*Convolution* means filtering each neighbourhood of a signal with a sliding filter. In each point, however, the operation performed can be seen as a scalar product. Let a basis filter be denoted $\mathbf{b}_k$. Using scalar product notation, filtering a function $\mathbf{f}$ with this basis is written

$$f_k = \langle \mathbf{b}_k, \mathbf{f} \rangle \tag{3.33}$$

where $f_k$ are the coordinates of the function in the dual basis (section 2.5.1). For simplicity, this notation will in the following also be used for convolution.

When the basic operation is understood, adding indices for denoting the spatial variables serves no purpose.

In image analysis, the function $\mathbf{f}$ often is a two- or three-dimensional function sampled on a Cartesian grid. The sample values are here thought of as *coordinates* corresponding to Dirac basis functions that are positioned at the sample positions. Let this "sample" basis be denoted by $\mathbf{e}_i$, where $i$ indexes a specific spatial position.

### 3.3.1   Averaging

Lowpass filtering increases the *local correlation* of the signal (figure 3.5). This means that the notion of an orthonormal *Dirac basis* in, for example an image, is correct only at the "highest" resolution.
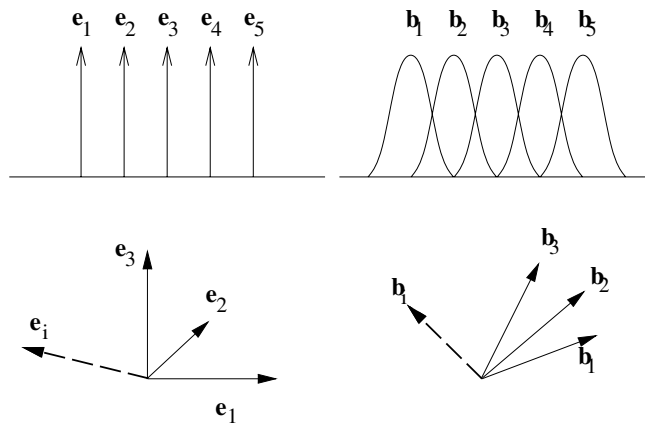


**Figure 3.5**   Averaging can be seen as changing to a non-orthogonal basis.

In many practical cases the "highest" resolution does not exist. Images acquired from an analogue video tape, for example, always have lower resolution than expected. Further, all optical systems have point spread functions blurring the images, in particular when the system is out of focus. In the cases where the samples of input signal are locally correlated, the basis functions that describe the signal should not be regarded as Dirac impulses but broader functions. (figure 3.5, right). If this correlation is known, the dual basis and metric tensors can be calculated, and the correlation can be compensated for using the inverse metric. This means that filtering can be performed as if the "original" unblurred signal was present. This procedure can be seen as an *inverse filtering* step followed by

a "standard" filtering step. However, as in all inverse filtering, this only works well if the noise level is low.

## 3.3.2   Gabor expansion

In 1946 Gabor [Gab46] proposed a combined representation of time and frequency. He expanded the signal in modulated *Gaussian basis functions*

$$f(x) \; = \; \sum_i \alpha_i \; g_{u_i,\sigma}(x) \tag{3.34}$$

where $u$ is the frequency variable and

$$g_{u_i,\sigma}(x) \; = \; \mathrm{e}^{iux} e^{-\frac{x^2}{\sigma^2}} \tag{3.35}$$

These complex modulated Gaussian basis functions are today commonly referred to as *Gabor functions*. Gabor's motivation for choosing these functions was that they are maximally compact in time and frequency simultaneously. The main criticism of these basis functions is that the Gabor functions are non-orthogonal, and although the functions are well localized they have infinite support, forcing truncation in all practical implementations.

Since the functions are non-orthogonal, the coefficients $\alpha_i$ in equation (3.34) can not be obtained by convolving the signal with the Gabor basis functions themselves. As mentioned earlier, the coefficients are obtained by convolving the signal with the dual Gabor basis functions. Gabor proposed an iterative method for finding the coefficients. Quite recently, methods for solving this problem analytically have been proposed. Bastiaans [Bas80] derived an analytical expression for a dual Gabor basis in one dimension. This dual basis is constructed using a so-called *auxiliary function* (figure 3.6). This result was extended to two dimensions by Porat and Zeevi [PZ88].

Figure 3.6 shows two centered Gabor functions, one with zero modulation, i.e. a Gaussian function, and one function modulated by $\Delta\omega$. A Gaussian function shifted $\Delta x$ is also shown in this figure. The product between the shift and the modulation is chosen so that $\Delta\omega\Delta x = 2\pi$. This corresponds to critical sampling when having infinitely many shifts and modulations. A shift-modulation product gives a redundant system, a frame. Multiplication of the centered Gaussian function with the auxiliary function produces a square wave with exponentially decreasing amplitude.
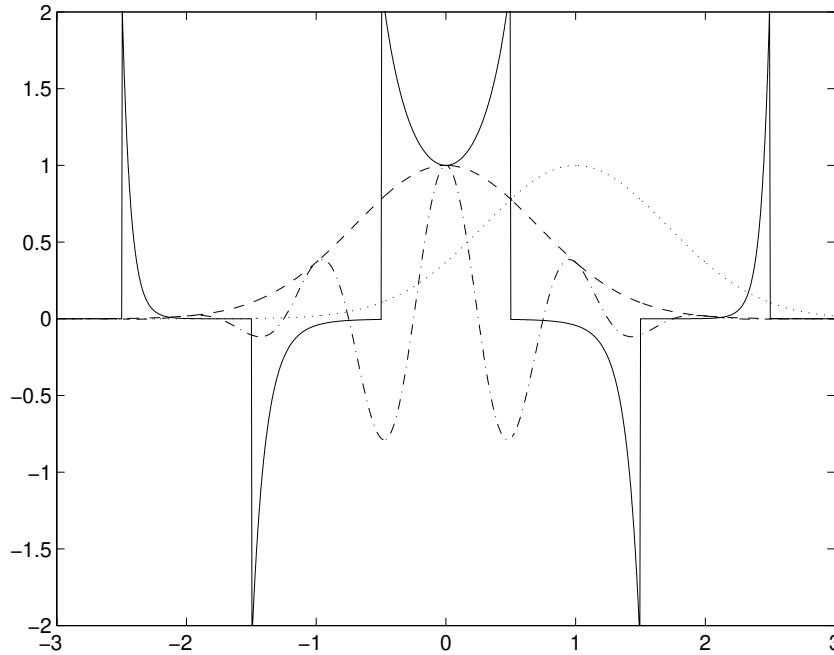
**Figure 3.6** Auxiliary function used for generation of the dual filter base corresponding to a critically sampled Gabor filter basis. Three different Gabor functions are shown (dashed).
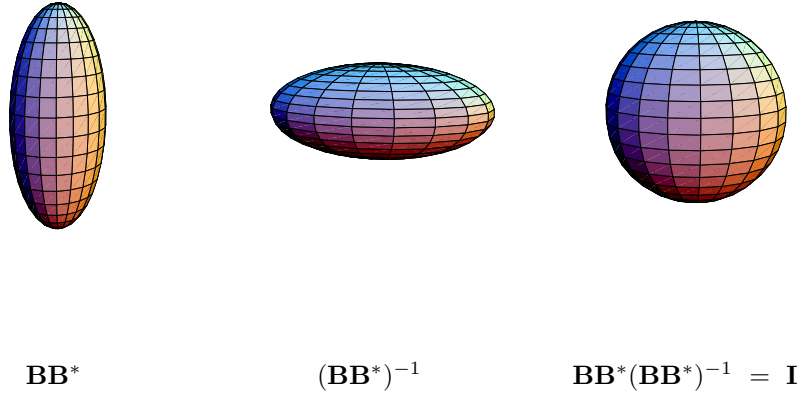
## 3.4 FRAMES

The frame concept refers to an extension of the basis concept allowing for redundant basis vectors. So, contrary to bases, these linearly dependent basis systems contain more "basis" vectors than dimensions of the space described.

The frame concept related to redundant basis systems was introduced by Duffin and Schaeffer in 1952 in the context of expanding a function in complex exponentials $e^{i\lambda_n x}$ where $\lambda_n \neq 2\pi n$. Note that frames do not necessarily need to be redundant or non-orthogonal. The concept includes the special cases of non-orthogonal and orthonormal bases. The following definition of a frame is adopted from Daubechies [Dau92]:

**Definition 3** *A family of functions* $\mathbf{b}_i$ *in a Hilbert space* $\mathcal{H}$ *is called a frame if there exist constants* $0 < \alpha,\ \beta < \infty$ *such that for all* $\mathbf{f} \in \mathcal{H}$,

$$\alpha\|\mathbf{f}\|^2 \leq \sum_i |\langle \mathbf{f}, \mathbf{b}_i \rangle|^2 \leq \beta\|\mathbf{f}\|^2 \tag{3.36}$$

$$\mathbf{B}\mathbf{B}^* \qquad\qquad (\mathbf{B}\mathbf{B}^*)^{-1} \qquad\qquad \mathbf{B}\mathbf{B}^*(\mathbf{B}\mathbf{B}^*)^{-1} = \mathbf{I}$$

**Figure 3.7** **Left**: Geometric interpretation of $\mathbf{B}\mathbf{B}^*$. The dimension of the hyper ellipsoid is equal to the dimension of the basis vectors, $m = \dim(\mathbf{b}_i)$. **Middle**: Geometric interpretation of $(\mathbf{B}\mathbf{B}^*)^{-1}$. **Right**: Geometric interpretation of $\mathbf{B}\mathbf{B}^*(\mathbf{B}\mathbf{B}^*)^{-1}$.

*The $\alpha$ and $\beta$ are called frame bounds. If $\alpha = \beta$ the frame is called a tight frame.*
□

The definition ensures that the basis functions $\mathbf{b}_i$ span the space since no signals $\mathbf{f}$ are allowed to have zero projection on every basis function. Note also that the definition does not require the basis vectors to be linearly independent.

## 3.4.1 The frame operator

Expanding the sum in equation (3.36)

$$\sum_i |\langle \mathbf{f}, \mathbf{b}_i \rangle|^2 = \sum_i \langle \mathbf{f}, \mathbf{b}_i \rangle \langle \mathbf{b}_i^*, \mathbf{f} \rangle \tag{3.37}$$

which in the finite-dimensional case can be written in matrix form. Let $\mathbf{b}_i \in {}^m$,

$$= \sum_i \mathbf{f}^* \mathbf{b}_i \mathbf{b}_i^* \mathbf{f} = \mathbf{f}^* \mathbf{B}\mathbf{B}^* \mathbf{f} \tag{3.38}$$

where

$$\mathbf{BB}^* \ = \ \mathbf{b}_1\mathbf{b}_1^* + \mathbf{b}_2\mathbf{b}_2^* + \ ... \ + \mathbf{b}_m\mathbf{b}_m^*$$

and $\mathbf{B}$ is a $n \times m$ matrix,

$$\mathbf{B} = \begin{pmatrix} | & | & ... & | \\ \mathbf{b}_1 & \mathbf{b}_2 & ... & \mathbf{b}_n \\ | & | & ... & | \end{pmatrix} \tag{3.39}$$

This operator is denoted the *frame operator*.

A visualization of the operator $\mathbf{BB}^*$ is shown in figure 3.7 (left). In general, the shown ellipsoid is an $n$-dimensional hyper-ellipsoid where $n$ is the dimension of the column vectors in $\mathbf{B}$. This mental imagery may help the understanding of the properties of a frame. The inverse of $\mathbf{BB}^*$ is obtained by inverting the radius of the principal axes. This is only possible if the minimum radius is greater than zero and the maximum radius bounded. The minimum value and the maximum value correspond to the frame bounds. If the radius of the hyper-ellipsoid collapses to zero in some of its dimensions, this sets the lower frame bound to zero and $\mathbf{B}$ is not a frame anymore. In mathematical terms this is equal to: since $\mathbf{B}$ is a frame, the symmetric operator $\mathbf{BB}^*$ is a positive bounded operator which makes it invertible (section 2.4.4).

## 3.4.2   The dual frame operator

Using the identity $\mathbf{BB}^*(\mathbf{BB}^*)^{-1} \ = \ \mathbf{I}$ we can write

$$\mathbf{B} \ = \ \mathbf{IB} \ = \ \mathbf{BB}^*(\mathbf{BB}^*)^{-1}\mathbf{B} \ = \ \mathbf{BI} \tag{3.40}$$

where the right identity operator is equal to $\mathbf{B}^*(\mathbf{BB}^*)^{-1}\mathbf{B}$. The dual frame is defined by the *dual frame operator* :

$$\tilde{\mathbf{b}}^i \ = \ (\mathbf{BB}^*)^{-1}\mathbf{b}_i \tag{3.41}$$

It can be shown that the dual basis vectors form a frame as well. We denote this frame the *dual frame*. It can be shown that the dual frame bounds are $\tilde{A} = \frac{1}{B}$ and $\tilde{B} = \frac{1}{A}$.

$$\mathbf{BB}^* \qquad\qquad (\mathbf{BB}^*)^{-1} \qquad\qquad \mathbf{BB}^*(\mathbf{BB}^*)^{-1} = \mathbf{I}$$
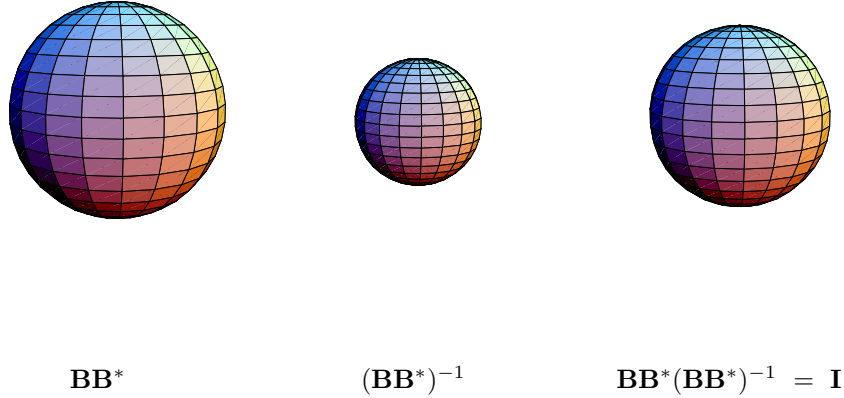
**Figure 3.8** **Left**: Geometric interpretation of $\mathbf{BB}^*$ in the case where $\mathbf{B}$ is a tight frame. **Middle**: Geometric interpretation of $(\mathbf{BB}^*)^{-1}$. **Right**: Geometric interpretation of $\mathbf{BB}^*(\mathbf{BB}^*)^{-1}$.

### 3.4.3   Describing a signal in a frame

In section 3.1, an expression for the coordinates of the signal corresponding to the filters basis was derived. This expression required that $\mathbf{B}^*\mathbf{B}$ was invertible. However, if $\mathbf{B}$ corresponds to the frame, $\mathbf{B}^*\mathbf{B}$ is not invertible and the solution in equation (3.6) does not exist. Rewriting equation (3.5) gives

$$\mathbf{BB}^*\mathbf{f} = \mathbf{BB}^*\mathbf{B}\theta \tag{3.42}$$

Although $\mathbf{B}^*\mathbf{B}$ is not invertible, the frame condition (equation (3.36)) gives that $\mathbf{BB}^*$ is invertible.

$$(\mathbf{BB}^*)^{-1}\mathbf{BB}^*\mathbf{f} = \mathbf{B}\theta \tag{3.43}$$

This gives the description of $\mathbf{f}' = \mathbf{B}\theta$ using the frame $\mathbf{B}$

$$\mathbf{f}' = (\mathbf{BB}^*)^{-1}\mathbf{BB}^*\mathbf{f} \tag{3.44}$$

Compare the expression with equation (3.7), where the signal $\mathbf{f}$ is expanded in a basis, $\mathbf{f}' = \mathbf{B}(\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*\mathbf{f}$

Using scalar product notation, equation (3.44) converts to

$$\mathbf{f}' = \tilde{\mathbf{b}}^i \langle \mathbf{f}, \mathbf{b}_i \rangle \tag{3.45}$$

where $\langle \mathbf{f}, \mathbf{b}_i \rangle$ are the coordinates of the signal corresponding to the dual frame. By symmetry, we also have

$$\mathbf{f} = \mathbf{b}_i \langle \mathbf{f}, \tilde{\mathbf{b}}^i \rangle \tag{3.46}$$

where $\langle \mathbf{f}, \tilde{\mathbf{b}}_i \rangle$ are the coordinates of the signal in the space spanned by the filter basis.

In a tight frame the relation between the frame and the dual frame is

$$\tilde{\mathbf{b}}^i = \frac{\mathbf{b}_i}{\alpha} \tag{3.47}$$

where $\alpha$ is the frame bound.

**Example 6 (Two orthogonal basis vectors in 2D)**
*The frame bounds for the orthonormal basis in figure 3.9 (upper left) are $A = B = 1$ since for any vector $\mathbf{f} = f^i \mathbf{b}_i \in \mathcal{H}$,*

$$\sum_{i=1}^{2} |\langle \mathbf{f}, \mathbf{b}_i \rangle|^2 = |f^1|^2 + |f^2|^2 = \|\mathbf{f}\|^2 \tag{3.48}$$

*According to Parseval: $\sum_i |\langle \mathbf{f}, \mathbf{b}_i \rangle|^2 = \|\mathbf{f}\|^2$, all orthonormal bases are tight frames with frame bounds $A = B = 1$.*                                                □

**Example 7 (Three equally spread basis vectors in 2D)**
*For a frame with three equally spread basis vectors (figure 3.9 (upper right)). The frame bounds are $A = B = \frac{3}{2}$ since*

$$\sum_{i=1}^{3} |\langle \mathbf{f}, \mathbf{b}_i \rangle|^2 = |f^1|^2 + \left| \frac{\sqrt{3}}{2} f^2 - \frac{1}{2} f^1 \right|^2 + \left| -\frac{\sqrt{3}}{2} f^2 - \frac{1}{2} f^1 \right|^2 \tag{3.50}$$

$$= \frac{3}{2} (|f^1|^2 + |f^2|^2) = \frac{3}{2} \|\mathbf{f}\|^2 \tag{3.51}$$

□

**Example 8 (Four equally spread basis vectors in 2D)**
*A frame consisting of four pairwise heavily dependent basis vectors is shown in figure 3.9 (lower left). It turns out that this frame is tight as well, with bounds $A = B = 2$*

$$\sum_{i=1}^{4} |\langle f, \mathbf{b}_i \rangle|^2 = |f^1|^2 + |f^2|^2 + |f^1|^2 + |f^2|^2 = 2\|f\|^2 \tag{3.51}$$

$\square$

Finally a frame with five equally spread basis vectors and the corresponding dual frame is shown in figure 3.9 (lower right).
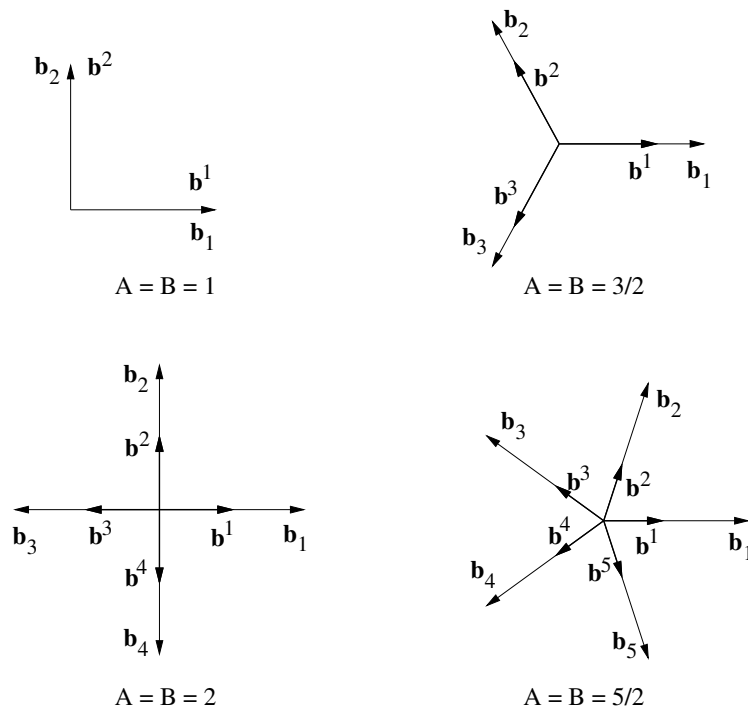


**Figure 3.9**  Examples of tight frames with corresponding dual frames and frame bounds, $\mathbf{b}^i = \frac{1}{A}\mathbf{b}_i$.

# 4

# NORMALIZED CONVOLUTION

One of the most important operations in image processing is convolution. The fundamental assumption underlying this fact is that the original representation of a particular neighbourhood, i.e. the values of the signal for each pixel, is not a good one and that the neighbourhood can be better understood when expressed in terms of a set of carefully chosen basis functions. The original basis, i.e. the set of impulses located at pixel positions, is considered to have a shift invariant metric, and thus the change of basis becomes trivial. However, the assumption of a shift invariant metric is often false, and neglecting this fact can introduce severe errors.

In this chapter a new method termed *Normalized convolution* is presented [KW93]. The method is an example of the power of the *signal/certainty* philosophy, i.e. the separation of data and operator into a signal part and a certainty part [Gra78, GK83, WK88, Knu89]. Missing data is simply handled by setting the certainty to zero. Localization or 'windowing' of operators is done using an applicability function, the operator equivalent to certainty, not by changing the actual operator coefficients. Spatially or temporally limited operators are handled by setting the applicability function to zero outside the window.

In this chapter many applications of normalized convolution are presented. Some of the examples can be comparably solved using standard techniques based on, for example, linear interpolation. The examples presented are not intended as a comparison with existing methods but are intended for increasing the understanding and to indicate the range of problems to which the normalized convolution technique can be applied. Although special purpose algorithms may outperform general purpose techniques, in the situations they are designed to handle, such algorithms are seldom extendable to solving new problems. In this chapter examples of normalized convolution, including interpolation of multidimensional signals and applications to spectrum analysis, are discussed. Further applications are found in chapter 5.

## 4.1   OPERATOR LOCALIZATION

In the previous chapter non-orthonormal bases were discussed. In this chapter the discussion is extended to deal uncertain data and operator locality. As described in chapter 2, filtering a signal, $\mathbf{f}$, with basis set, $\mathbf{b}_i$, gives the coordinates of the dual filter basis $f_i$. If the basis is orthonormal, the contravariant and the covariant coordinates are equal. Thus, the coordinates $f^i$ of the signal in this basis, $\mathbf{f} = f^i \mathbf{b}_i$ is given directly. If the basis is not orthonormal, we know from equation (2.30) how to compensate for this. Acting on the filter results with the coordinates of the metric tensor viewing the basis set as orthonormal, $\langle \mathbf{b}_i, \mathbf{b}_j \rangle^{-1}$, yields the coordinates $f^j$

$$f^j = \langle \mathbf{b}_i, \mathbf{b}_j \rangle^{-1} \langle \mathbf{b}_i, \mathbf{f} \rangle \tag{4.1}$$

If the used filters are not spatially localized as, for example, in the case of the spatially infinite basis functions of the discrete Fourier transform, the filters are not realizable. Classically, filters are made localized through multiplication by a *window function*. Such an operation changes the filter properties as well as the relation between filters in a filter basis set. For example, the Fourier basis just mentioned, is *orthogonal* on an infinite grid. Multiplication by a *Gaussian window* destroys that property and makes the system *non-orthogonal*.

### 4.1.1   Avoiding signal windowing

A much preferable way of obtaining local signal analysis is to let the basis functions be accompanied by a value stating how important the function, at each given point, is for the analysis. In this way, the operator can be left unchanged and locality is introduced by letting the importance of the signal decrease with the distance from the centre according to a window function. How this is done is shown in this section.

If the signal can be fully described by the basis functions used, there exist coordinates such that

$$\mathbf{f} \;=\; f^k \mathbf{b}_k \tag{4.2}$$

Inserting this in equation (4.1) gives

$$f^j \;=\; \langle \mathbf{b}_i, \mathbf{b}_j \rangle^{-1} \langle \mathbf{b}_i, f^k \mathbf{b}_k \rangle \tag{4.4}$$

$$\;=\; f^k \underbrace{\langle \mathbf{b}_i, \mathbf{b}_j \rangle^{-1} \langle \mathbf{b}_i, \mathbf{b}_k \rangle}_{\delta_k^j} \tag{4.5}$$

A sinusoidal basis function and a spatial localization window

Spatially localized basis function
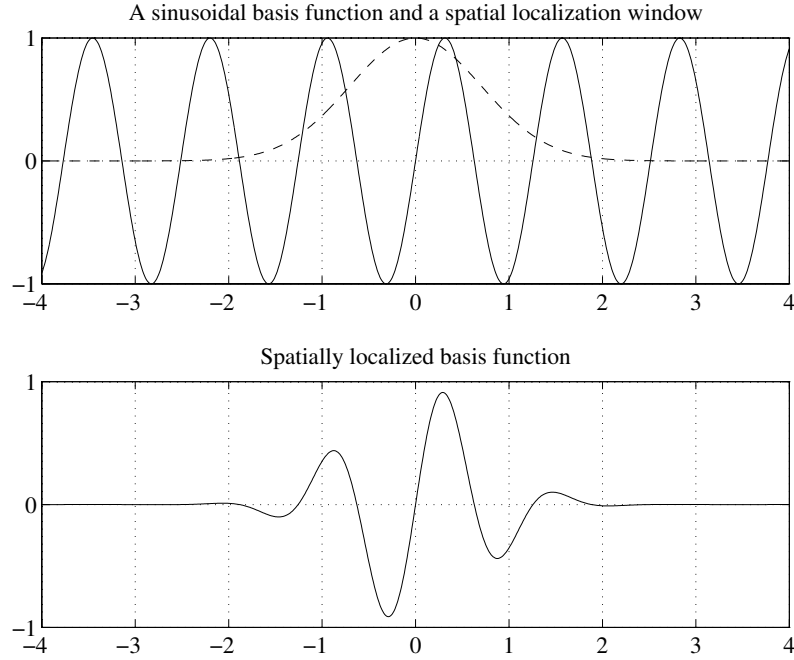
**Figure 4.1**  **Top**: A sinusoidal basis function, **b**, and spatial localization window (dashed), $a$. **Bottom**: The product of the basis filter and the localization window, $a\mathbf{b}$.

which shows that the estimated coordinate vector is equal to the signal coordinate vector $f^j$. The shape of any window not making the basis function linearly dependent can be compensated for since

$$\langle a\mathbf{b}_i, \mathbf{b}_j \rangle^{-1} \langle a\mathbf{b}_i, \mathbf{b}_k \rangle = \delta_k^j \tag{4.5}$$

where $a$ denotes a scalar localization window. This shows that for all windows where the matrix $\langle a\mathbf{b}_i, \mathbf{b}_j \rangle$ is invertible, the correct coordinate vector is estimated. The requirement on the window is not particularly restrictive and is not a problem in practise.

The formula in equation (4.5) can be interpreted as calculating the coordinates of metric tensor interpreting the *windowed basis function* set as orthonormal. The effect of this is that the window is used as a certainty mask making the filter localized without actually changing it. This certainty mask is denoted the *applicability function*. Note that if the coordinates of the metric tensor are calculated using the windowed filter functions $\langle a\mathbf{b}_i, a\mathbf{b}_j \rangle = \langle \mathbf{w}_i, \mathbf{w}_j \rangle$ the coordinates, $(f_w)^i$,

corresponding to the windowed $\mathbf{w}_i = a\mathbf{b}_i$ basis are obtained, not the coordinates, $(f_b)^i$, corresponding the original filter basis $\mathbf{b}_i$

The above discussion shows how to make the basis, in which the signal is expressed in, *invariant* to the localizing operator.

**Example 9** *Let the basis set of the discrete Fourier transform be denoted* $\{\mathbf{b}_i\}$. *This set is orthonormal,*

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = \delta_{ij} \tag{4.6}$$

*Assuming sufficient resolution in the Fourier domain, the transform of a sinusoidal input signal produces an impulse function. In other words, only one of the basis functions matches the signal and the signal is equal to a scaled version of that basis function,* $\mathbf{f} = \alpha\mathbf{b}_k$.

$$\langle \mathbf{b}_i, \mathbf{f} \rangle = \begin{cases} \alpha & i = k \\ 0 & otherwise \end{cases} \tag{4.7}$$

*In a practical implementation, a windowing function with compact support has to be used. This corresponds to a smearing in the Fourier domain making all basis functions correlated. In general this leads to all filters responding to the single frequency input signal, although normally not as strong as the response from the basis filter* $\mathbf{b}_k$.

$$\langle a\mathbf{b}_i, f \rangle \neq 0 \qquad \forall\, i \tag{4.8}$$

*As described in equation (4.5), a solution that removes the unwanted cross talk between the filters, is to use the metric tensor defined by the basis used:*

$$f^j \;=\; \langle a\mathbf{b}_i, \mathbf{b}_j \rangle^{-1} \langle a\mathbf{b}_i, \mathbf{f} \rangle \tag{4.9}$$

$\square$

It has been shown how non-orthogonality introduced by windowing can be compensated for by measuring using the appropriate metric. This is true also if the original basis is non-orthonormal. The metric tensor compensates for both the window and the intrinsic non-orthogonality.

## 4.2   SIGNALS AND CERTAINTIES

The applicability function, $a$, introduced in the previous section is a certainty function for the basis functions. By symmetry, we may define a *certainty function* for the data, too. This scalar function is denoted by $c$, for certainty.

$$\mathbf{f} \;=\; \langle \mathbf{b}_i, c\mathbf{b}_j \rangle^{-1} \langle \mathbf{b}_i, c\mathbf{f} \rangle \mathbf{b}_j \tag{4.10}$$

Note that $c$ is not a scalar value but a scalar function. This formulation allows for the data to be uncertain. The uncertain basis functions become weaker when the certainty becomes small. Dimensions that are uncertain are compressed and finally vanish when the certainty approaches zero.

### 4.2.1   Missing samples

An example of uncertain data is the *missing data* case. The certainty function $c$ is then a binary function being zero where data is missing. Data may be missing of various reasons. It may be because of *drop outs*, or more common, outside a signal border, as in figure 1.1 in the introductory chapter. The same signal is shown in figure 4.2 demonstrating how borders are handled. The certainty is set to zero outside the signal border.

A signal may be expressed in different bases. Spatially limited signals call for a description using spatially localized functions which in the extreme case take the form of Dirac functions. When working with sampled signals, the samples themselves can be seen as such Dirac basis functions. A missing sample means that the impulse basis function is missing. In a changing basis situation, proper signal analysis is possible only if the representation of the signal is complete, i.e. the representation includes not only the coordinates of the signal but also the basis in which the coordinates are given. For images, a general representation of one basis impulse function would consist of its spatial position and of its strength. In the following, spatial positions are assumed to be quantized in a regular fashion so that positions can be identified simply by enumeration. The impulse strength, however, will be explicitly represented and denoted by $c$. In the following $c$ will be referred to as being the *certainty* of the signal.

Missing data should be interpreted as the basis vector defining the data is missing and not that the coordinate of this vector is zero. In the eyes of the data, this distinction is only philosophical since the basis vector disappears when multiplying it with a coordinate having zero value. However, for a filter basis function this view is critical. Since the filter coordinate of the basis function in general is not zero, the filter has to know that the basis vector is missing and therefore not
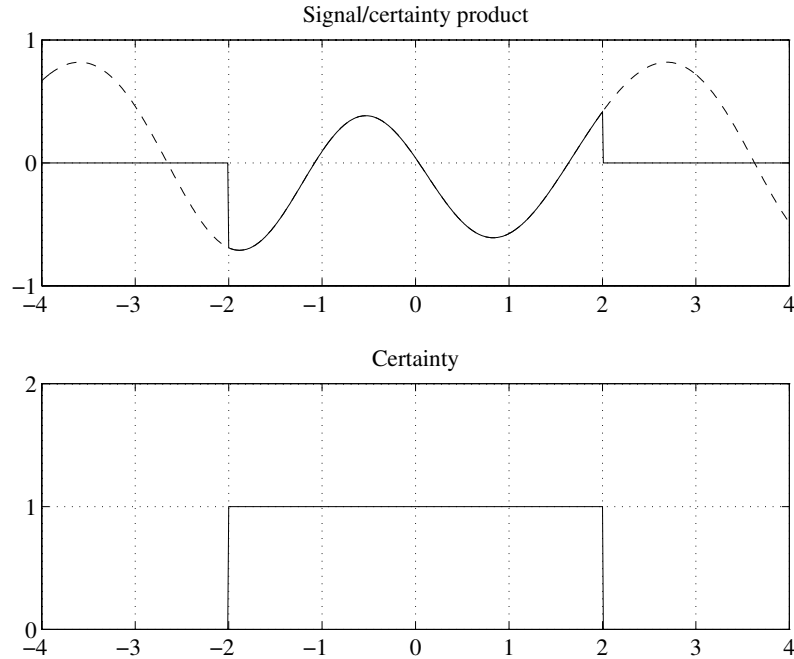
**Figure 4.2**  Unwanted border effects can be reduced by setting the certainty of the signal to zero outside the border.

use it for description. Otherwise, the filter function would be changed to a new *different* filter having zero value in this position. For this purpose a certainty value associated with each basis function has been inserted in the formulas.

## 4.2.2  Non-orthonormal basis and uncertain data

The calculations in section 3.2.2 will now be carried out for the missing data case. Consider the same signal vector  as before, but multiplied by a certainty field. In this example most of the data is lost. The certainty function is set to:

$$(c_e)^i = \begin{pmatrix} 0 & 0 & 1 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.11}$$

As mentioned earlier, in a product sum, the location and the ordering of the elements is not of importance and viewing the certainty function as a vector reduces the number of indices

$$(c_e)^i = (\,0 \quad 0 \quad 1 \quad 0.1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\,) \tag{4.12}$$

which generates the new metric coordinates

$$(G_c)_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.13}$$

The coordinates of this metric expressed in the new basis, $\mathbf{b}^i \otimes \mathbf{b}^j$ are

$$(G_c)_{nm} = \mathbf{G}(\mathbf{c}_n, \mathbf{c}_m) = \begin{pmatrix} 1 & -1 \\ -1 & 1.001 \end{pmatrix} \tag{4.14}$$

and the corresponding inverse metric (the conjugate metric tensor )

$$(G_c)^{nm} = \begin{pmatrix} 1001 & 1000 \\ 1000 & 1000 \end{pmatrix} \tag{4.15}$$

From equation (3.17) we get:

$$(f_b)^n = (G_b)^{nm} \underbrace{(G_c)_{ij}(b_c)^j_m (f_c)^i}_{\text{dual coordinates}} = (G_b)^{nm}(f_b)_m \tag{4.16}$$

Inserting numerical values gives the *dual coordinates* of the signal

$$(f_b)_m = (G_c)_{ij}(b_c)^j_m (f_c)^i = (\,3 \quad -3\,) \tag{4.17}$$

and

$$(f_b)^n = (G_b)^{nm}(f_b)_m = (\,3 \quad -3\,)\begin{pmatrix} 1001 & 1000 \\ 1000 & 1000 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \tag{4.18}$$

which are the coordinates of the signal in the signal space defined by the basis $\mathbf{b}_i$ (figure 3.4).

## 4.3 NORMALIZED CONVOLUTION

As mentioned in chapter 1, the philosophy of *normalized convolution* is that data as well as operators are accompanied by a scalar function field representing the appropriate 'weight' assigned to data or operator values. Combining the approach in sections 4.1 and 4.2 gives

$$\mathbf{f} \;=\; \langle a\mathbf{b}_i, c\mathbf{b}_j \rangle^{-1} \langle a\mathbf{b}_i, c\mathbf{f} \rangle \mathbf{b}_j \tag{4.19}$$

In a practical case, we need to work with localized basis functions, the term $c\mathbf{b}_j$ is inappropriate. One way of solving this problem was shown in example 4.1. Here, we have a data dependent weighting implying that the window needed varies spatially. It is not an appealing solution since it would require new filters at every spatial position. A solution is to rearrange the terms

$$\langle a\mathbf{b}_i, c\mathbf{b}_j \rangle \mathbf{b}_j \;=\; \langle a\mathbf{b}_i\mathbf{b}_j^*, c \rangle \mathbf{b}_j \tag{4.20}$$

which shows that position invariant outer product filters, $\mathbf{b}_i\mathbf{b}_j^*$, applied on the scalar certainty field give the same result. We are now ready to define a method for performing general convolution operations on data of signal/certainty type - normalized convolution:

**Definition 4** *Normalized convloution is defined by*

$$\mathbf{f} \;=\; \langle a\mathbf{b}_i\mathbf{b}_j^*, c \rangle^{-1} \langle a\mathbf{b}_i, c\mathbf{f} \rangle \mathbf{b}_j \tag{4.21}$$

$\square$

## 4.3.1 Applicability functions

The applicability function can be said to define the localization of the convolution operator. The appropriate choice of this function depends of course on the application. The family of applicability functions used in our experiments is given by:

$$a \;=\; \begin{cases} r^{-\alpha} \, cos^{\beta}(\frac{\pi \, r}{2 \, r_{max}}) & r < r_{max} \\ 0 & \text{otherwise} \end{cases} \tag{4.22}$$

where $r$ denotes the distance from the neighbourhood center, $\alpha$ and $\beta$ are positive. In figure 4.3, $\alpha = 0$, $\beta = 2$ and $r_{max} = 8$ yielding a fairly smooth applicability function.
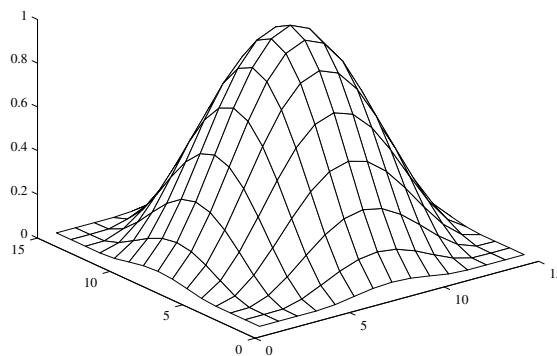


**Figure 4.3**   Example of an applicability function with $\alpha = 0$, $\beta = 2$ and $r_{max} = 8$.

A shaper filter is shown in figure 4.4, where the parameters are set to $\alpha = 3$, $\beta = 0$ and $r_{max} = 8$. This type of applicability function is useful when, for example, interpolating missing data. Convolving a signal with such a narrow filter will not change the existing data much since the filter can be considered all-pass (the filter is very broad in the Fourier domain). However, in areas where data is missing, neighbouring values will be used for recovering the signal.

## 4.4   APPLICATIONS OF NORMALIZED CONVOLUTION

### 4.4.1   Interpolation by normalized averaging

To illustrate the above method, we consider an example using normalized convolution to obtain an efficient interpolation algorithm in a missing sample situation. In the simplest possible case, the operator filter basis consists of only one position
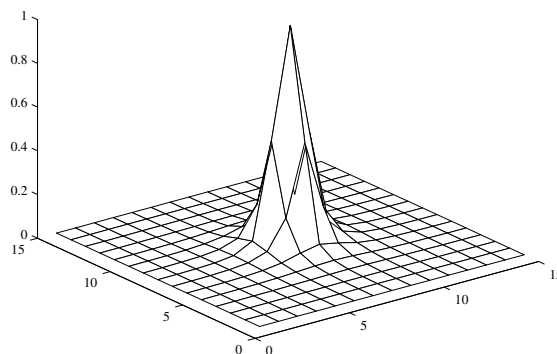
**Figure 4.4** An applicability function having $\alpha = 3$, $\beta = 0$ and $r_{max} = 8$.

invariant basis function, i.e. $\mathbf{b} = 1$. In this special case, equation (4.21) reduces to

$$\mathbf{f} = \langle a\mathbf{b}, c\rangle^{-1}\langle a\mathbf{b}, c\mathbf{f}\rangle\mathbf{b} = \langle a, c\rangle^{-1}\langle a, c\mathbf{f}\rangle\mathbf{b} \qquad (4.23)$$

Since $\mathbf{b}$ now is a constant, it is evident that all components of $\mathbf{f}$ will be subject to the same transformation, and it is consequently enough to study the case where $\mathbf{f}$ is a scalar.

Figure 4.5 (top left) shows a *sparsely* and *irregularly* sampled test-image, constructed by using gated white noise, with a threshold chosen so that only 10% of the data remains. Any attempt to reconstruct this image by simple smoothing will fail because of the variation of the sample density. Figure 4.5 (top right) shows the result of smoothing done using the filter of figure 4.3. Figure 4.5 (bottom left) shows the result using the same filter as applicability function in normalized averaging: the local normalization performed compensates effectively for the sample density variations. Although the result is satisfactory, an even better result can be obtained using a more localized applicability function. The parameter values for the applicability function used to produce the result shown in figure 4.5 (bottom right) are $\alpha = 3$ and $\beta = 2$. Using such a filter results in an adaptive smoothing of the reconstructed image. The more samples missing, the greater the smoothing performed.
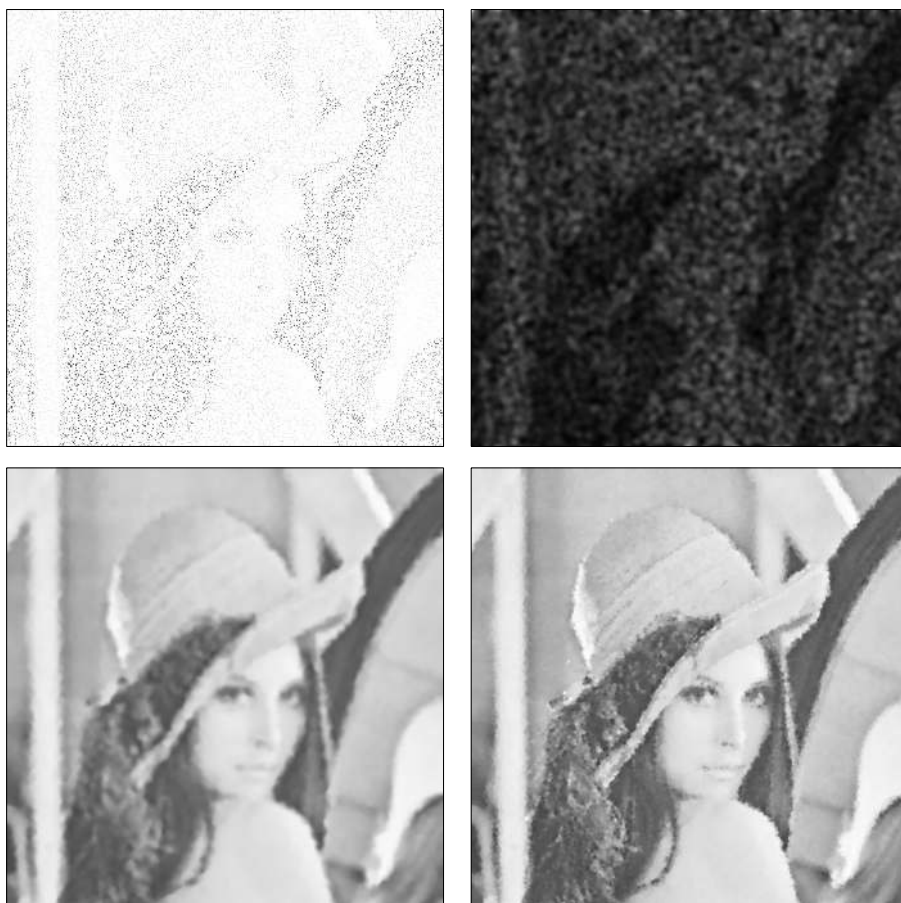
**Figure 4.5  Top left:** The famous Lena-image degraded to a grey-level test image containing only 10% of the original information. **Top right:** Interpolation using standard convolution with a normalized smoothing filter (see figure 4.3). **Bottom left:** Interpolation using normalized averaging with the same filter used as applicability function. **Bottom right:** Normalized averaging using a more local applicability function (see figure 4.4).

The second example is interpolation of a densely sampled image having large missing regions, see figure 4.6. The result shows that the algorithm is capable of filling in the 'holes' with 'plausible' data. The test also shows that the algorithm performs well close to the image border since this region of an image also belongs to the case densely samples/large missing regions.



**Figure 4.6**  **Left**: Test image. **Right**: Interpolation using normalized convolution.

## 4.4.2   Higher order interpolation

In section 4.4.1, interpolation using *one* basis function was discussed. In this section, it is shown that even better interpolation results can be achieved using a larger set of basis functions. *Signal reconstruction* always implies the use of a signal model. In the case of normalized convolution, the model is given by the chosen basis functions $\{\mathbf{b}_i\}$. The signal reconstruction, performed by normalized convolution, is a weighted least squares solution using the available basis functions, where the weights are adaptive and given by the data certainties and the applicability function [KW93]. Note, however, that a good reconstruction can be achieved only if the chosen basis functions are likely to be good descriptors for a large part of the signal.

In the two examples below, seven basis functions were used; one constant function and three sin/cos pairs having one, two and three cycles per graph width. Although the signal was chosen not to fit any of the individual basis functions, the reconstruction of the signal is good. Each part of the signal is reconstructed locally.

Figure 4.7 shows higher order interpolation of a "random walk" signal. The expectation value of the spectrum of such a signal decreasing as one over the frequency variable (as the spectrum in many images). Although more than half the samples were removed at random, the reconstruction is good.
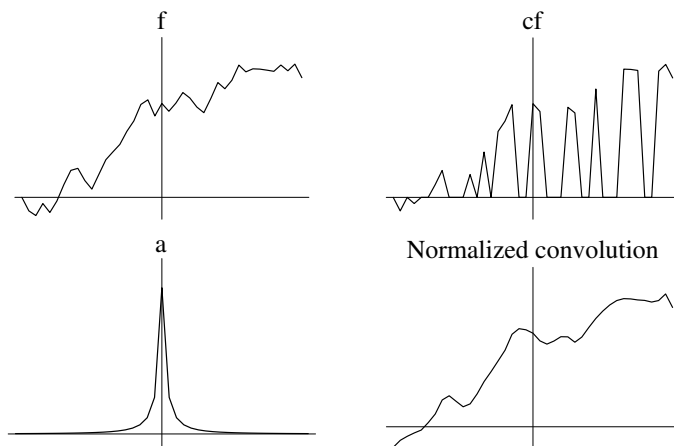


**Figure 4.7** Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).

Figure 4.8 shows higher order interpolation of a smoothly varying signal. In this case, where the used basis functions describe the signal well, the signal is almost completely restored although large parts are missing.

Figure 4.9 shows higher order interpolation of a signal containing the signal presented in figure 4.8 plus a term with fast variation. In areas where the certainty is high, the reconstruction is very good. In areas where large parts are missing, a "low-pass" version of the signal is reconstructed: the signal is adaptively smoothed.

## 4.4.3 Spectrum analysis

All *spectrum analysis* methods, explicitly or implicitly, involve some kind of windowing operation. In image processing the windows are typically small and the standard windowing operation is multiplication, i.e. the signal is simply multiplied by the window function prior to the analysis. The effect is that what is being analyzed is not the signal but a corrupted version thereof, the implications
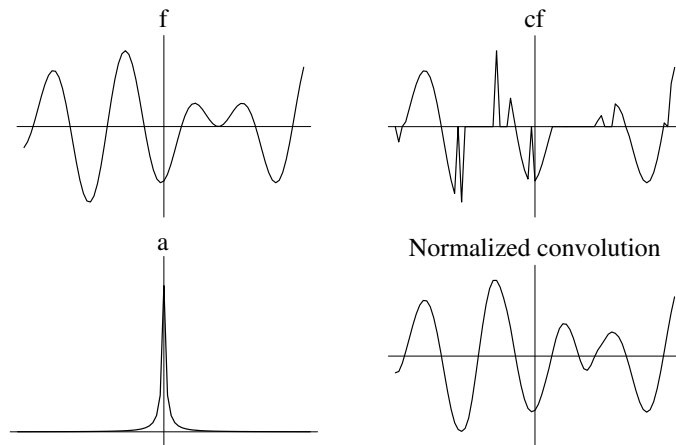
**Figure 4.8** Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).
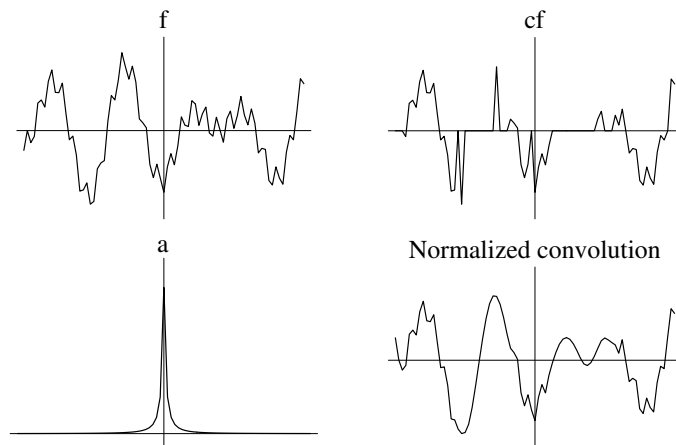


**Figure 4.9** Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).

of which are well-known and unwanted [Tho82]. This way of solving the locality problem is a clear violation of the *signal/certainty principle*.

A missing sample situation can equivalently be seen as the signal being irregularly sampled. Standard spectrum analysis has no way of coping with irregularly

sampled signals. The signal is given in an inappropriate basis and the coordinates of the appropriate metric tensor is not the identity matrix.

Figure 4.10 shows a spectrum analysis experiment. The top left shows the signal which is a sum of a constant term plus two different sinusoids. The top right shows the function that is used both as window and as applicability function. The bottom of figure 4.10 shows the result of standard windowed DFT analysis (left) and of spectrum analysis using normalized convolution (right). The advantage of using normalized convolution is evident - the frequency domain resolution is significantly improved and the two frequency components of the sinusoids are clearly separated.
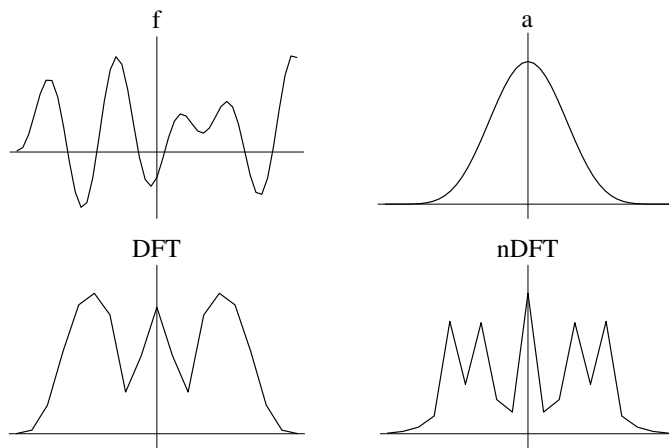


**Figure 4.10**  A spectrum analysis experiment. The top left shows the signal. The top right shows the function that was used both as window and as applicability function. At the bottom, the result of standard windowed DFT analysis (left) and of spectrum analysis using normalized convolution (right) is shown.

Figure 4.11 shows the result of an experiment using the same signal as in figure 4.10, the difference being that 50% of the samples were removed at random. Unsurprisingly the removal of the samples has a dramatic effect on the standard DFT of the signal. The spectrum analysis performed by normalized convolution is, however, remarkably robust and only minor changes can be seen.
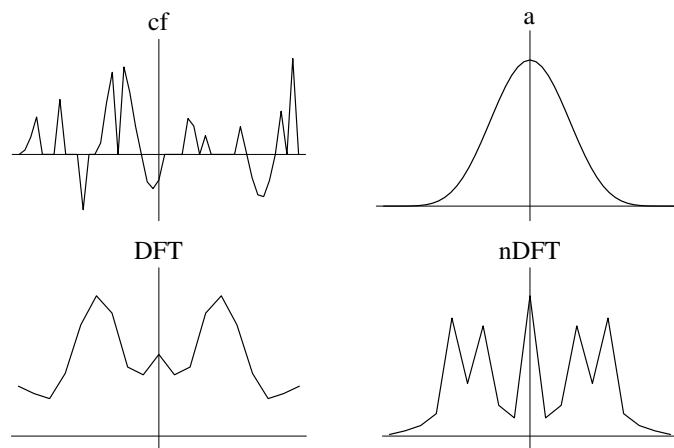
**Figure 4.11**   The result of an experiment using the same signal as in figure 4.10, the difference being that 50% of the samples were removed at random. The removal of the samples has a dramatic effect on the standard DFT of the signal, the spectrum analysis performed by normalized convolution is, however, robust.

# 5

# NORMALIZED DIFFERENTIAL CONVOLUTION

This chapter presents a particular version of normalized convolution. The chapter begins with presenting a consistency algorithm introduced in a patent [KG86]. This alghorithm is a predecessor to an algorithm here termed *normalized differential convolution* [KW93]. As the name indicates, the method is based on analysis of signal differences. In [WNK94] it was shown that normalized differential convolution is invariant to any constant term in the signal but otherwise produces the same estimates as normalized convolution. It is shown that a generalization of normalized differential convolution is possible and that operations can be made invariant to any subspace of the space spanned by the filter functions.

## 5.1   A CONSISTENCY ALGORITHM

In this section a consistency algorithm detecting *rotational symmetries* will be reviewed [KG86]. The vectors in this section are two-dimensional, i.e. they can be described by pairs of scalar numbers. It is sometimes convenient to regard these vectors as complex numbers. In this section, we shall limit ourselves to the study of polar separable functions whose angular variation is defined by the circular harmonic functions $e^{ik\varphi}$. The angular modulation speed, $k$, determines the family of symmetries to which the filter is sensitive. Two important families are first order symmetries ($k = 1$), also termed *parabolic symmetries*, and second order symmetries ($k = 2$), also termed *circular symmetries*. The algorithm is applied on vector data produced by the two-dimensional orientation algorithm described in chapter 6. The names of the symmetry families refer to the grey-level patterns corresponding to these symmetries.

### 5.1.1 Circular symmetries

The simplest circular symmetry objects are circles with centres at the origin. Figure 5.1 (top left) shows such a grey-level neighbourhood. The corresponding double angle orientation representation (section 6.2.3) is shown below in figure 5.1.

$$\mathbf{f} = f(r)e^{i\varphi_f} = f(r)e^{i2\varphi} \tag{5.1}$$

where $f(r)$ is a radial magnitude function.

A filter function tuned to the signal $\mathbf{f}$ is a filter having the same angular variation, with a suitable lowpass magnitude function limiting the spatial extent of the filter,

$$\mathbf{b}_2 = a(r)e^{i2\varphi} \tag{5.2}$$

where the subscript indicates the order of the symmetry. Filtering a signal $\mathbf{f} = f(r)e^{i\varphi_f}$ with this filter function can in each point be expressed in terms of the scalar product between the signal and the filter,

$$\mathbf{s} = \langle \mathbf{f}, \mathbf{b}_2 \rangle \tag{5.3}$$

Assuming some suitable discretization of the pattern and the filter, this scalar product may be expressed by a product sum,

$$\mathbf{s} = \langle \mathbf{f}, \mathbf{b}_2 \rangle = \sum_k \mathbf{f}_k \mathbf{b}_2^* = \sum_k f_k a_k e^{i((\varphi_f)_k - (\varphi_b)_k)} \tag{5.4}$$

where $\varphi_b$ denotes the phase of the filter $\mathbf{b}_2$. If the signal has the same angular variation speed as the filter function but differs by a phase shift of $\alpha$,

$$\varphi_f = 2\varphi + \alpha \tag{5.6}$$

$$\varphi_b = 2\varphi \tag{5.7}$$

then the filter output from equation (5.4) gives

$$\mathbf{s} = \sum f_k a_k e^{i(2\varphi_k + \alpha - 2\varphi_k)} = e^{i\alpha} \sum f_k a_k = se^{i\alpha} \tag{5.7}$$

This shows that the filter proposed, $\mathbf{b}_2$, can be used to discriminate between a whole class of different second order symmetries:

$$\mathbf{f} \;=\; f(r)e^{i2\varphi} \qquad\qquad \mathbf{f} \;=\; f(r)e^{i2(\varphi+\frac{\pi}{2})}$$
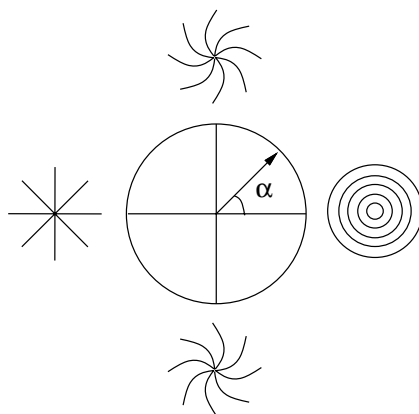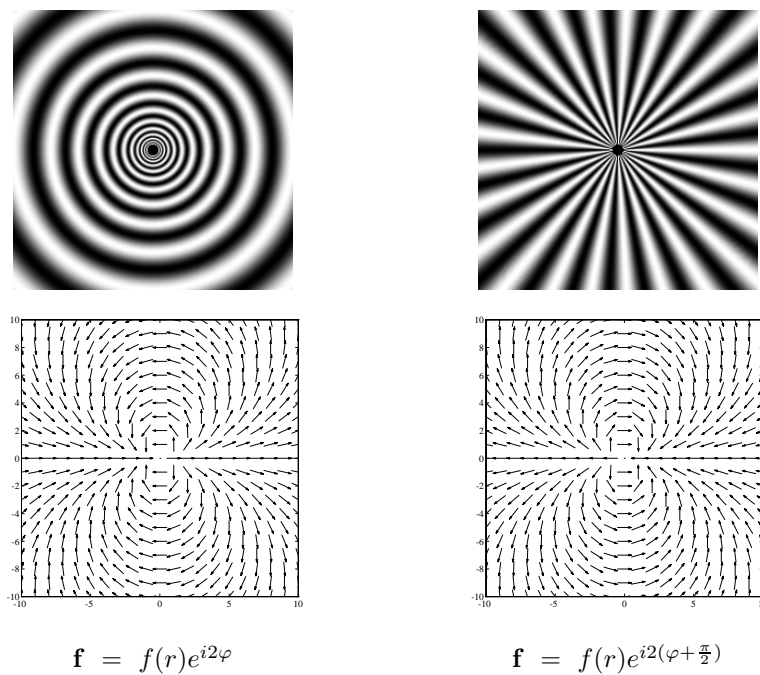
**Figure 5.1** **Top**: Two circular symmetries and the corresponding local orientation representation. Note that the right vector field is $-1$ times the left one, i.e. every vector differs maximally between the two fields. **Bottom**: Vector representation of circular symmetries.

$$\mathbf{f} = ce^{i(2\varphi+\alpha)}, \qquad \alpha \in [0, 2\pi[ \tag{5.8}$$

The argument of the complex filter output describes the type of pattern. $\alpha = \frac{\pi}{2}$ corresponds to a star pattern (figure 5.1, top right). In addition to circular and radial patterns, a discrimination between left-handedness and right-handedness can be done. The resulting vector representation describing circular symmetries is shown in figure 5.1 (bottom).

## 5.1.2   Parabolic symmetries

For first order symmetries or parabolic symmetries we will simply assume that we have patterns that, in the vectorial representation given earlier, can be written as

$$\mathbf{f} = f(r)e^{i\varphi+\alpha} \tag{5.9}$$

Grey-level patterns producing this type of local orientation description are found in figure 5.2 (top) with $\alpha = \pi$ (left) and $\alpha = \frac{3\pi}{2}$. A description of the local orientation algorithm is found in chapter 6.

The difference between equation (5.9) and the corresponding equation in the preceding section, equation (5.1), is the fact that the vector rotates with a rate of $\varphi$, rather than $2\varphi$. As in the preceding case, the detection of these and intermediary classes of patterns can be achieved by filtering with a complex filter that matches the signal in equation (5.9),

$$\mathbf{b}_1 = a(r)e^{i\varphi}, \tag{5.10}$$

where the subscript indicates the order of the symmetry. This filter is sensitive to *curvature*, more specifically *convexity* and *concavity* , and the direction outward from the centre of convexity. The resulting vector representation describing parabolic symmetries is shown in figure 5.2 (bottom).

## 5.1.3   Increasing operator selectivity

The issue addressed in this section is the possibility to increase operator feature selectivity. The standard windowing approach for convolution operators necessarily makes such operators sensitive to a broad spectrum of signals, often much broader than that for which the operator was originally intended.

Consider for example the following example based on figure 5.3. First, note that the star pattern and the line are non-orthogonal in the signal space, i.e. the line is

$$\mathbf{f} \;=\; f(r)e^{i\varphi+\pi} \qquad\qquad\qquad \mathbf{f} \;=\; f(r)e^{i\varphi+\frac{3\pi}{4}}$$
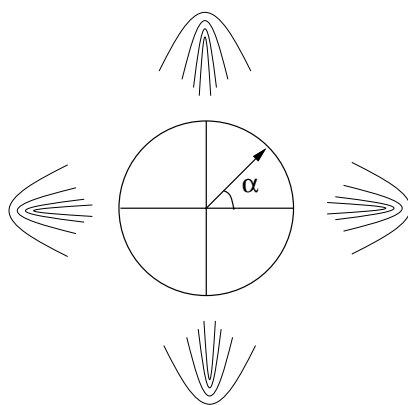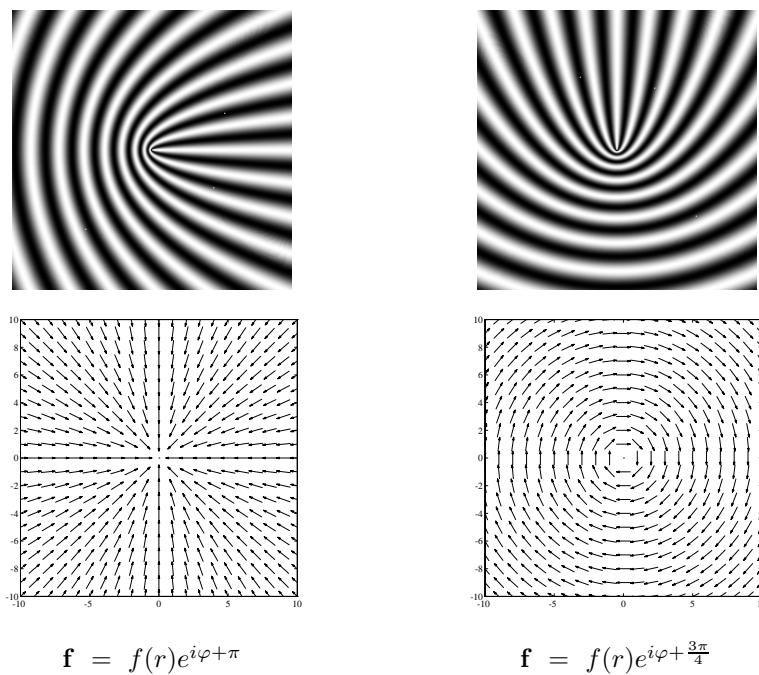


**Figure 5.2** **Top**: Two different parabolic symmetries and the corresponding local orientation representations. **Bottom**: Vector representation of parabolic symmetries.
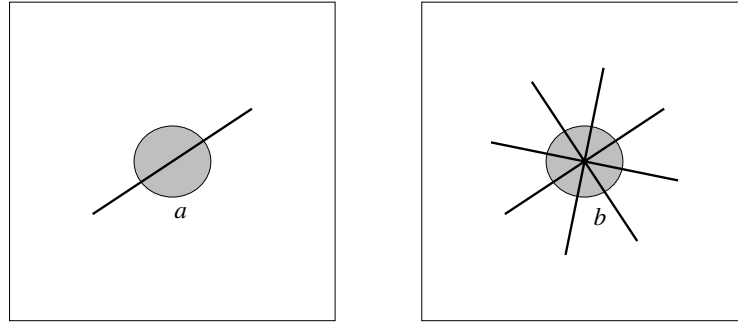
**Figure 5.3**   The left region, $a$, can be seen as a part of the more complex right region, $b$. The $b$-region can be seen as a linear combination of different oriented $a$-regions. This observation leads to the conclusion that, in general, any *linear* operation designed to detect patterns like $a$ will give some response on patterns like $b$ and vice versa!

a part of the star pattern. It follows that, in general, any *linear* operator designed to detect the line will have an expected response greater than zero for the star pattern and vice versa. In many situations a more specific operator would be preferable, e.g. when a single line is not considered to be the evidence of a star pattern.

The symmetry operators described in the previous section are sensitive to *changes* in vector fields. In the discussion above, however, only the change caused by angular variation is of interest. The magnitude of the vectors in the input field reflects the certainty of the local orientation estimated. If this variation is not compensated for when estimating local symmetries, the operators respond to many structures they were not designed for. A straightforward normalization of the input orientation field,

$$\hat{\mathbf{f}} \;=\; \frac{\mathbf{f}}{\|\mathbf{f}\|} \;=\; \frac{f(r)e^{i\varphi}}{f(r)} \tag{5.11}$$

gives a field that is undefined in positions where $f$ is zero. Inserting the value zero, or any other value in order to make the signal well-defined, introduces artifacts. Experimental results have shown that operators in general give strong erroneous responses to these new structures, unfortunately often much stronger than the responses to the structures of interest.

This problem was addressed by Knutsson et al in a patent [KHG86] describing an apparatus for detecting circular symmetries. The star pattern in figure 5.3 belongs to this class of symmetries. The patent describes a method termed a

consistency operation. The name refers to that the operator should only respond to signals that are consistent with the signal model. The unwanted responses on locally one-dimensional areas of a linear symmetry operator were removed by a combination of a set of convolutions. The following four filter results are needed:

$$\mathbf{s}_1 = \langle \mathbf{f}, \mathbf{b} \rangle \tag{5.13}$$

$$\mathbf{s}_2 = \langle \|\mathbf{f}\|, \mathbf{b} \rangle \tag{5.14}$$

$$\mathbf{s}_3 = \langle \mathbf{f}, \|\mathbf{b}\| \rangle \tag{5.15}$$

$$\mathbf{s}_4 = \langle \|\mathbf{f}\|, \|\mathbf{b}\| \rangle \tag{5.16}$$

where $\| \cdot \|$ denotes the magnitude of the filter or the data.

The four filter results are combined into

$$\mathbf{s} = \frac{\mathbf{s}_4 \mathbf{s}_1 - \mathbf{s}_2 \mathbf{s}_3}{\mathbf{s}_4^{\gamma}} \tag{5.16}$$

where $\gamma$ is a constant controlling the energy dependence of the algorithm. The value is normally set to one. The denominator is an energy normalization controlling the model versus energy dependence of the algorithm. With $\gamma = 1$, the output magnitude varies linearly with the magnitude of the input signal.

Figure 5.4 (left) shows a grey-level test image of size $128 \times 128$ and the corresponding estimates of local orientation using quadrature filters of size $15 \times 15$. The result of applying a filter corresponding to parabolic symmetries is shown in figure 5.5. The size of the parabolic filter is $15 \times 15$. Note that the filter is sensitive to the corners of the boxes where the orientation vector field is rotating, as well as to the edges, because of the variation in the magnitude of the vectors. The fact that the filters respond to lines and edges is not surprising. The filters above have been used for detecting linear structures in grey-level images [Dan80]. The other three images in figure 5.5 show that this can effectively be compensated for by applying the consistency scheme described above.
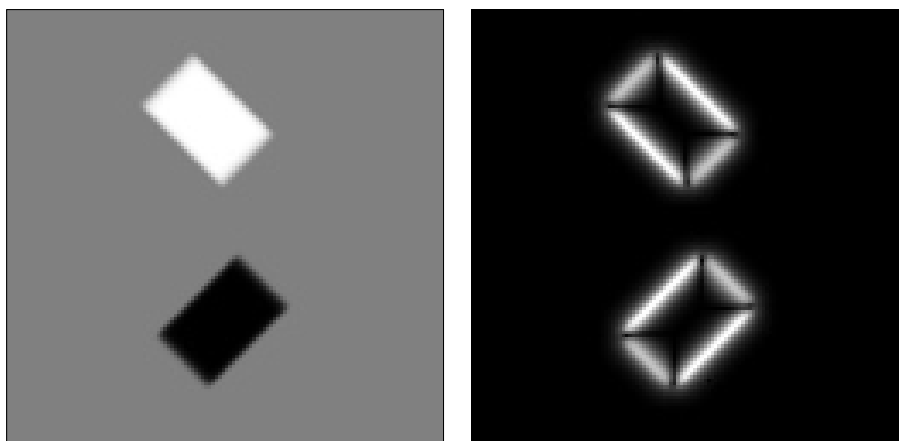
**Figure 5.4**   A grey-level test image containing two boxes is shown to the left. To the right, the local orientation is estimated. The brightness corresponds to the norm of the eigenvectors corresponding to the largest eigenvalue.
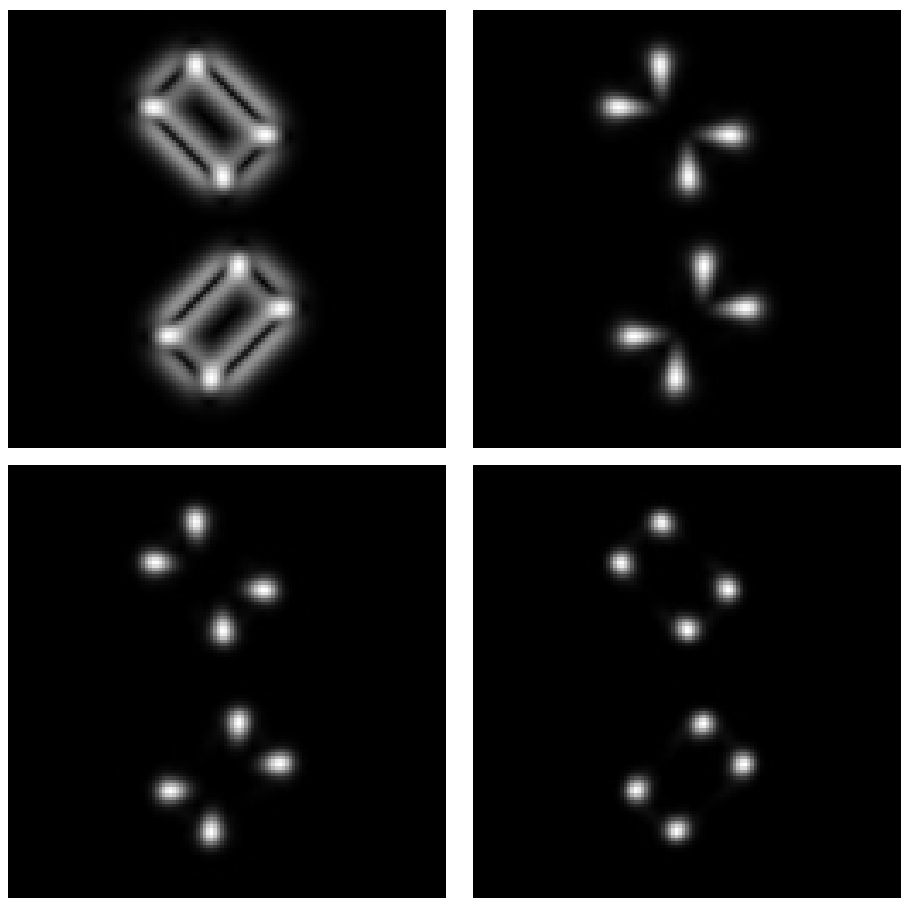
**Figure 5.5** **Top**: Result after filtering the orientation vector field above with a parabolic symmetry filter (left). Note that the filters not only respond to the vector patterns they were designed for, but to many other structures as well, $\gamma = 2$ (right). **Bottom**: Same as the top right figure but with $\gamma = 1$ (left) and $\gamma = 0$ (right).

## 5.2   NORMALIZED DIFFERENTIAL CONVOLUTION

In the previous section a consistency algorithm for detecting symmetries was reviewed. This algorithm is the predecessor to a procedure termed *normalized differential convolution*. The method is intimately related to normalized convolution which was presented in chapter 4 [WNK94].

We will first discuss an operation termed *differential convolution*. This operation can be shown to be equivalent to locally weighted sums over all operator differences acting on the corresponding data differences - hence the name. This description may, to begin with, not give a full understanding of the operation. The key words are, however, *operator differences* and *data differences*. Since differences is measured, the operation insensitive to any constant term in the input signal.

**Definition 5** *Let differential convolution between $a\mathbf{b}_i$ and $c\mathbf{f}$ be defined and denoted by:*

$$d_i \;=\; \langle a, c\rangle \langle a\mathbf{b}_i, c\mathbf{f}\rangle \quad - \quad \langle a\mathbf{b}_i, c\rangle \langle a, c\mathbf{f}\rangle \tag{5.17}$$

$\square$

Differential convolution is based on a nonlinear combination of different standard convolutions. The term, $\langle a\, c\rangle$, can be regarded as the local certainty energy and the second term, $\langle a\mathbf{b}_i, c\mathbf{f}\rangle$, is the term that corresponds to standard convolution. When it comes to the third and fourth term the interpretation is somewhat harder. Each operator in the filter is weighted locally by corresponding data producing a weighted average operator, where the weights are given by the data.

$\langle a\mathbf{b}_i, c\rangle \leftrightarrow$ data dependent mean-operator

For the fourth term it is vice versa. The mean-data is calculated using the operator applicability as weights.

$\langle a, c\mathbf{f}\rangle \leftrightarrow$ operator dependent mean-data

Differential convolution should consequently be interpreted as: a standard convolution weighted with the local "energy" minus the "mean" operator acting on the "mean" data.

The name *differential convolution* stems from a basic property that might not be evident from the definition. It can been shown that differential convolution performs summation over all operator differences acting on corresponding data differences, The meaning of this statement will hopefully be clearer after the following calculations. Writing the sums in equation (5.17) explicitly gives:

$$
\begin{aligned}
d_i &= (\sum_{k=1}^{N} a_k c_k)(\sum_{l=1}^{N} a_l \mathbf{b}_{il} c_l \mathbf{f}_l) - (\sum_{k=1}^{N} c_k a_k \mathbf{b}_{ik})(\sum_{l=1}^{N} a_l c_l \mathbf{f}_l) \\
&= \sum_{kl} a_l a_k c_l c_k \mathbf{b}_{il} \mathbf{f}_l - \sum_{kl} a_k a_l c_k c_l \mathbf{b}_{ik} \mathbf{f}_l \qquad (5.19)
\end{aligned}
$$

where $k$ and $l$ are spatial indices while $i$ corresponds to the basis function used. Since the sum is symmetric, it is invariant to interchange of $n$ and $m$, and the symmetric counter part of each term can be added if the result is divided by two

$$
\frac{1}{2}\left[\sum_{kl} a_k a_l c_k c_l (\mathbf{b}_{il}\mathbf{f}_l + \mathbf{b}_{ik}\mathbf{f}_k) - \sum_{kl} a_k a_l c_k c_l (\mathbf{b}_{ik}\mathbf{f}_l + \mathbf{b}_{il}\mathbf{f}_k)\right] =
$$

$$
\frac{1}{2}\sum_{kl} a_k a_l c_k c_l (\mathbf{b}_{il}\mathbf{f}_l + \mathbf{b}_{ik}\mathbf{f}_k - \mathbf{b}_{ik}\mathbf{f}_l - \mathbf{b}_{il}\mathbf{f}_k) =
$$

$$
\frac{1}{2}\sum_{klm} a_k a_l c_k c_l (\mathbf{b}_{il} - \mathbf{b}_{ik})(\mathbf{f}_l - \mathbf{f}_k) \qquad (5.20)
$$

which shows that differential convolution performs summation over all operator differences acting on corresponding data differences.

It may be worth repeating that the double sum in equation (5.19) is never carried out. The result is achieved by the combination of four simple sums. Note that only point pairs having non-zero $a_k, c_k, a_l$ and $c_l$ will contribute to the sum. If $\mathbf{b}_i$ or $\mathbf{f}$ is constant, the sum will vanish since either $\mathbf{b}_{il} - \mathbf{b}_{ik} = 0$ or $\mathbf{f}_k - \mathbf{f}_l = 0$.

Normalized differential convolution is, as the name indicates, a combination of the concepts presented in this chapter, normalized convolution and differential convolution. Normalized differential convolution is a method for performing general differential operations on data of signal/certainty type in cases where the constant term of the signal, i.e. the zero order moment, is zero or irrelevant.

**Definition 6** *Let normalized differential convolution between $a\mathbf{b}_i$ and $c\mathbf{f}$ be defined and denoted by:*

$$
f^j = G_{ij}^{-1}\left(\langle a, c \rangle \langle a\mathbf{b}_i, c\mathbf{f}\rangle - \langle a\mathbf{b}_i, c\rangle \langle a, c\mathbf{f}\rangle\right) \qquad (5.20)
$$

*where*

$$G_{ij} \;=\; \langle a,c\rangle\langle a\mathbf{b}_i\mathbf{b}_j^*,c\rangle - \langle a\mathbf{b}_i,c\rangle\langle a\mathbf{b}_j^*,c\rangle \tag{5.21}$$

$\square$

## 5.3   APPLICATIONS OF NORMALIZED DIFFERENTIAL CONVOLUTION

### 5.3.1   Gradients from planar basis functions

This example shows how the gradients of the signal can be estimated using normalized differential convolution. The minimum number of basis functions needed for estimating of the local gradient using normalized differential convolution is one for each dimension. The basis functions needed in $n$ dimensions are

$$\{1,\mathbf{x}_1,\mathbf{x}_2,\ \ldots\ ,\mathbf{x}_n\}$$

where $\mathbf{x}_k$ are planar functions. Figure 5.7 shows such filters in the two-dimensional case (second and third figure to the left).

Localization of these basis functions by an applicability function $a$ gives

$$\{a,a\mathbf{x}_1,a\mathbf{x}_2,\ \ldots\ ,a\mathbf{x}_n\}$$

Inserting the function in the relation derived in equation (5.17), writing all the indices explicitly, gives:

$$d_i \;=\; \langle a,c\rangle\langle a\mathbf{b}_i,c\mathbf{f}\rangle - \langle a\mathbf{b}_i,c\rangle\langle a,c\mathbf{f}\rangle \tag{5.23}$$

$$=\; \frac{1}{2}\sum_{kl}\underbrace{a_k a_l c_k c_l}_{d_{kl}}\underbrace{(\mathbf{x}_{ik}-\mathbf{x}_{il})}_{\Delta\mathbf{x}_{ikl}}\underbrace{(\mathbf{f}_k-\mathbf{f}_l)}_{\Delta\mathbf{f}_{kl}} \tag{5.24}$$

$$=\; \frac{1}{2}\sum_{kl} d_{kl}\Delta\mathbf{x}_{ikl}\Delta\mathbf{f}_{kl} \tag{5.25}$$

and

$$G_{ij} \;=\; \langle a,c\rangle\langle a\mathbf{b}_i\mathbf{b}_j^*,c\rangle - \langle a\mathbf{b}_i,c\rangle\langle a\mathbf{b}_j^*,c\rangle$$

$$= \frac{1}{2}\sum_{kl} d_{kl}\Delta\mathbf{x}_{ikl}\Delta\mathbf{x}_{jkl} \tag{5.26}$$

If the gradient, $\frac{\partial\mathbf{f}}{\partial x_i}$, is constant in the neighbourhood, then

$$\Delta\mathbf{f}_{kl} = \Delta\mathbf{x}_{kli}\frac{\partial\mathbf{f}}{\partial x_i} \quad \text{for all } k,l \tag{5.26}$$

Equation (5.25) then simplifies to:

$$d_i = \frac{\partial\mathbf{f}}{\partial x_i}\frac{1}{2}\sum_{kl} d_{kl}\Delta\mathbf{x}_{ikl}\Delta\mathbf{x}_{jkl} \tag{5.27}$$

Equations (5.25) and (5.27) inserted in equation (5.2), the definition of normalized differential convolution gives:

$$G_{ij}^{-1}d_i = \frac{\partial\mathbf{f}}{\partial x_i} \tag{5.28}$$

which shows that the true gradient is estimated for an ideal input signal.

## A two-dimensional example

This example shows estimation of the local gradient in a sparsely and irregularly sampled two-dimensional scalar field. The set of basis functions needed are:

$$\{1, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1^2, \mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2^2\} \tag{5.29}$$

where the last three filter functions are the three product filters, $\mathbf{b}_i\mathbf{b}_j^*$ in equation (5.21) needed for calculating the coordinates of the local metric. Localizing the functions with an applicability function $a$ gives

$$\{a, a\mathbf{x}_1, a\mathbf{x}_2, a\mathbf{x}_1^2, a\mathbf{x}_1\mathbf{x}_2, a\mathbf{x}_2^2\} \tag{5.30}$$

These filters are shown in figure 5.7 (right).

Inserting these functions in the definition of normalized convolution gives

$$d_i = (d_1, d_2) = \langle a, c\rangle \begin{pmatrix} \langle a\mathbf{x}_1, c\mathbf{f}\rangle \\ \langle a\mathbf{x}_2, c\mathbf{f}\rangle \end{pmatrix} - \begin{pmatrix} \langle a\mathbf{x}_1, c\rangle \\ \langle a\mathbf{x}_2, c\rangle \end{pmatrix} \langle a, c\mathbf{f}\rangle$$
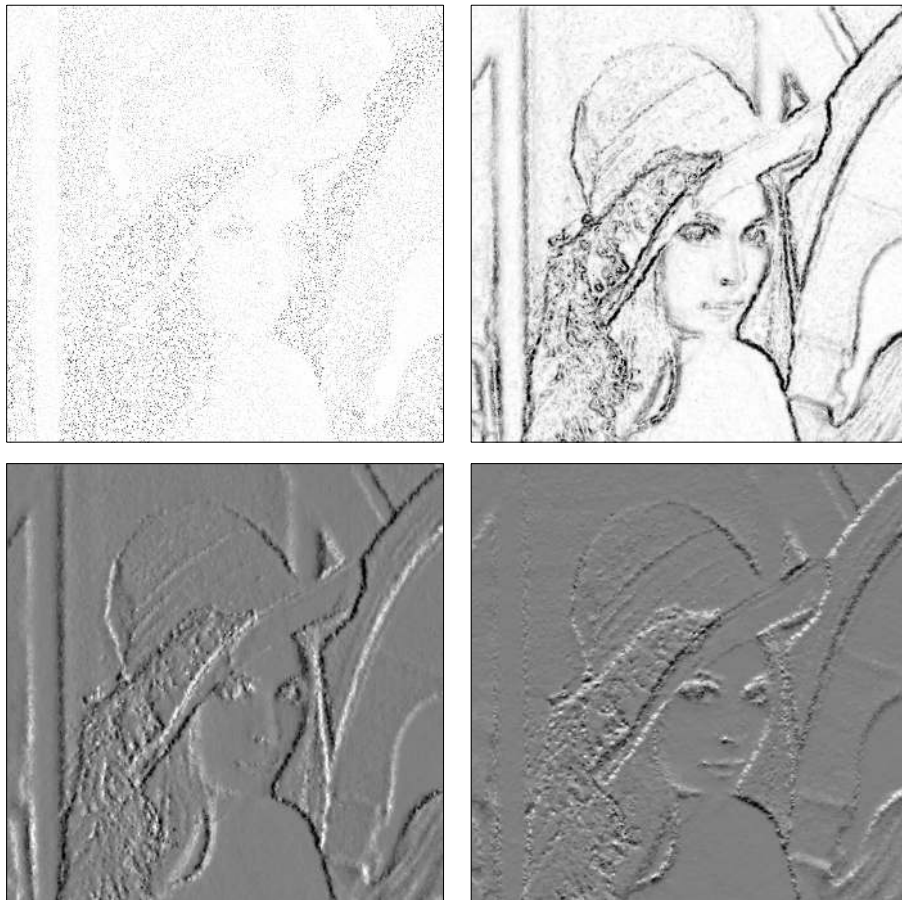
**Figure 5.6** **Top** Left: The Lena-image degraded to 10% of the original information. **Top** Right: Gradient magnitude estimation using normalized differential convolution. **Bottom**: Estimation of the x- and y-gradient in the top left image using normalized differential convolution with a filter having $\tau = -3, \beta = 0, r_{max} = 8$.
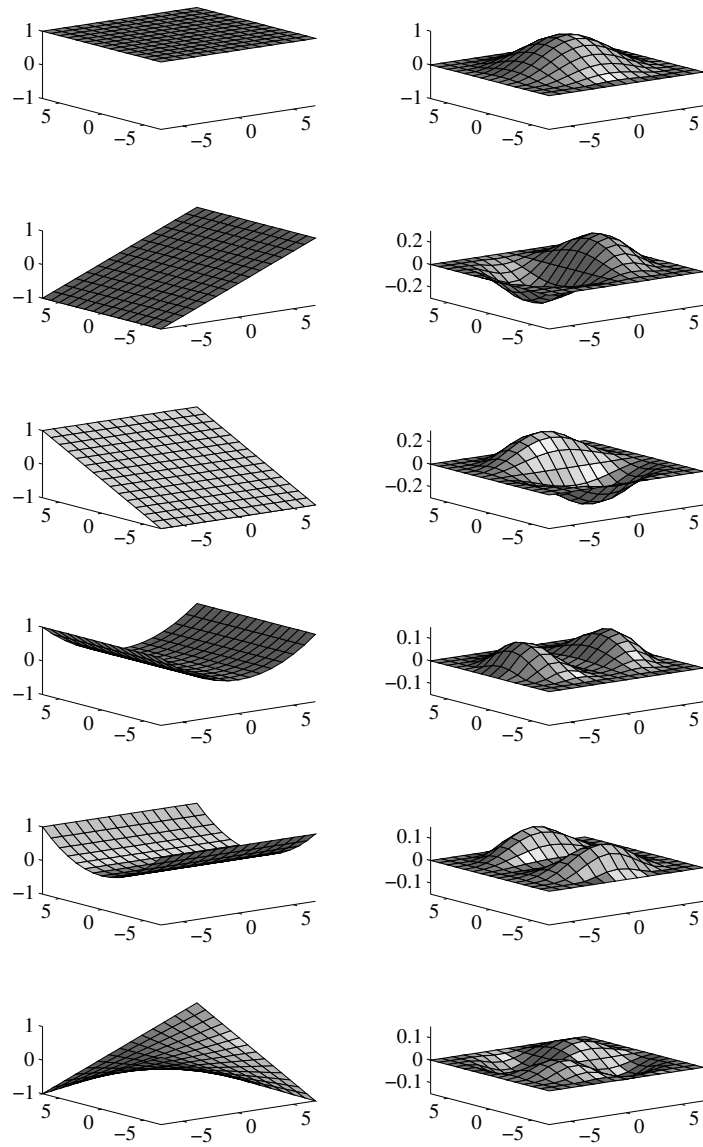
**Figure 5.7** **Left**: Filters required for gradient estimation in two dimensions. **Rigth**: Multiplying the left function with the applicability function produces filters with finite support. The top right function shows the shape of the applicability function, the analytical expression as defined in equation (4.22) have $\alpha = 0, \beta = 2, r_{max} = 8$. From the top the filters shown are $1, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1^2, \mathbf{x}_2^2, \mathbf{x}_1\mathbf{x}_2$.

$$
\begin{aligned}
G_{ij} \quad = \quad & \langle a,c \rangle \left(
\begin{array}{cc}
\langle a\mathbf{x}_1^2, c \rangle & \langle a\mathbf{x}_1\mathbf{x}_2, c \rangle \\
\langle a\mathbf{x}_1\mathbf{x}_2, c \rangle & \langle a\mathbf{x}_2^2, c \rangle
\end{array}
\right) - \\
& \left(
\begin{array}{cc}
\langle a\mathbf{x}_1, c \rangle \langle a\mathbf{x}_1, c \rangle & \langle a\mathbf{x}_1, c \rangle \langle a\mathbf{x}_2, c \rangle \\
\langle a\mathbf{x}_1, c \rangle \langle a\mathbf{x}_2, c \rangle & \langle a\mathbf{x}_2, c \rangle \langle a\mathbf{x}_2, c \rangle
\end{array}
\right)
\end{aligned}
$$

Note that many terms in this expression are identical. The actual number of convolutions needed is the number of basis functions in equation (5.30), i.e. six scalar convolutions, figure 5.7.

The sparsely sampled test image used in the interpolation example in section 4.3 is now used for testing the new gradient estimation method. The image is filtered with the six filters shown in figure 5.7. Combining these outputs according to the normalized convolution procedure gives the gradient estimate, see figure 5.6. As shown, the algorithm has no problem coping with the large variation in the sampling density

## 5.3.2   Divergence and curl in two dimensions

The gradient estimation method described in the previous section can be used for estimation of first order differential invariants of image velocity fields, i.e. curl, divergence and shear [KvD75]. Since these invariants have found considerable attention in Computer Vision [CB92, Kan86, NA88], it is interesting to test the signal/certainty philosophy on the problem of differentiating a velocity field. The motion field induced by a moving camera is often sparse since estimation of velocity requires moving texture or borders. The motion of flat surfaces is unknown (not zero).

The invariants divergence, curl and shear can all be defined as a combination of partial derivatives of the spatial coordinates Divergence in two dimensions is defined by:

$$
\nabla \cdot \mathbf{f} = \text{trace}(\nabla \mathbf{f}) = f_{11} + f_{22} = \frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} \tag{5.31}
$$

The curl in 2D is a scalar defined by:

$$
\nabla \times \mathbf{f} = f_{21} - f_{12} = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \tag{5.32}
$$

The use of complex numbers instead of two-dimensional vectors simplifies the formalism for the divergence and the curl operator. We therefore introduce a complex gradient operator and a complex data representation:

$$\nabla = \frac{\partial}{\partial x_1} + i \frac{\partial}{\partial x_2} \quad \text{and} \quad \mathbf{f} = f_1 + i f_2$$

This gives the following representation of divergence and rotation:

$$\text{Div} = x f_1 + y f_2 \quad \text{and} \quad \text{Curl} = x f_2 - y f_1$$

Let $\mathbf{b} = e^{i\beta_k}$ and $\mathbf{f} = e^{i\tau_k}$. Inserting this in the definition of normalized differential convolution gives:

$$
\begin{aligned}
d &= \langle a, c \rangle \langle a\mathbf{b}, c\mathbf{f} \rangle - \langle a\mathbf{b}, c \rangle \langle a, c\mathbf{f} \rangle \\
&= \frac{1}{2} \sum \underbrace{a_k a_l c_k c_l}_{r_{kl}} (e^{i\beta_k} - e^{i\beta_l})(e^{i\tau_k} - e^{i\tau_l}) \\
&= 2 \sum r_{kl} \sin(\frac{\Delta\beta_{kl}}{2}) \sin(\frac{\Delta\tau_{kl}}{2}) e^{i(\Delta\beta_{kl} + \Delta\tau_{kl})}
\end{aligned}
$$

where:

$$
\begin{aligned}
\Delta\tau &= \tau_k - \tau_l \\
\Delta\beta &= \beta_k - \beta_l
\end{aligned}
$$

and

$$
\begin{aligned}
G &= \langle a, c \rangle \langle a\mathbf{b}\mathbf{b}^*, c \rangle - \langle a\mathbf{b}, c \rangle \langle a\mathbf{b}^*, c \rangle \\
&= \frac{1}{2} \sum a_k a_l c_k c_l \, \|(e^{i\beta_k} - e^{i\beta_l})\|^2 \\
&= 2 \sum r_{kl} \sin^2(\frac{\Delta\beta_{kl}}{2})
\end{aligned}
$$

Thus

$$G^{-1}d = \frac{\sum r_{kl} \sin(\frac{\Delta\beta_{kl}}{2}) \sin(\frac{\Delta\tau_{kl}}{2}) e^{i(\Delta\beta_{kl} + \Delta\tau_{kl})}}{\sum r_{kl} \sin^2(\frac{\Delta\beta_{kl}}{2})} \tag{5.33}$$

In this example it is shown how local divergence and curl can be estimated in an image sequence [Fle92]. The camera is moving towards a slightly slanted photograph of a tree, figure 5.8 (left). The input data is the sparse estimated velocity field presented in figure 5.8 (right). The result of standard convolution using the same operator is shown in figure 5.9 (left). The result of normalized differential convolution using a divergence/curl operator is shown in figure 5.9 (right).
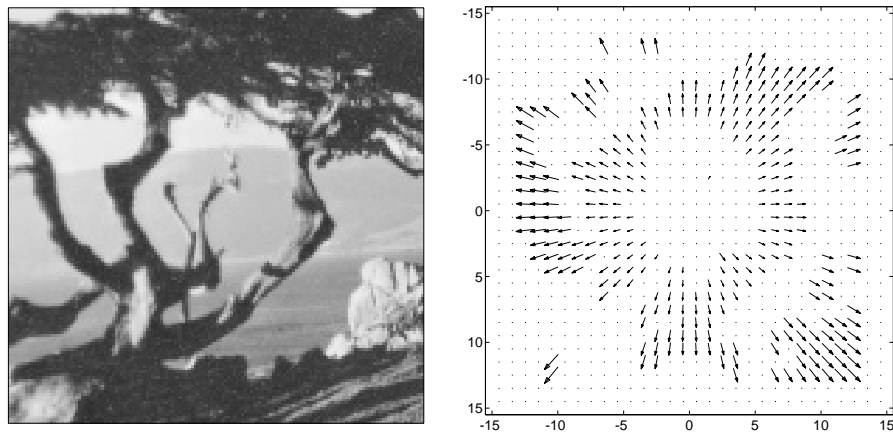


**Figure 5.8** **Left**: One frame from the tree-sequence. **Right**: Estimated velocities from the tree-sequence [LHK93]. In order to increase the visuability the original $150 \times 150$ image has been resampled to $30 \times 30$ pixels.

## 5.3.3   Robot vision applications

Another important area of applications for the normalized differential convolution technique is found in robot vision.

The experiment was carried out using a simulated environment used to study the performance of an active vision robot. The software package has been developed by Westelius within the ESPRIT Basic Research Action 7108 "Vision as Process" [Wes92].

### *Model based generation of certainty fields*

Extracting salient features to attract the focus of attention is a central problem in active vision. The opposite problem of neglecting salient features on already modelled structures is equally important. If the focus of attention gets stuck on the most salient feature it is not of very much help. The normalized differential
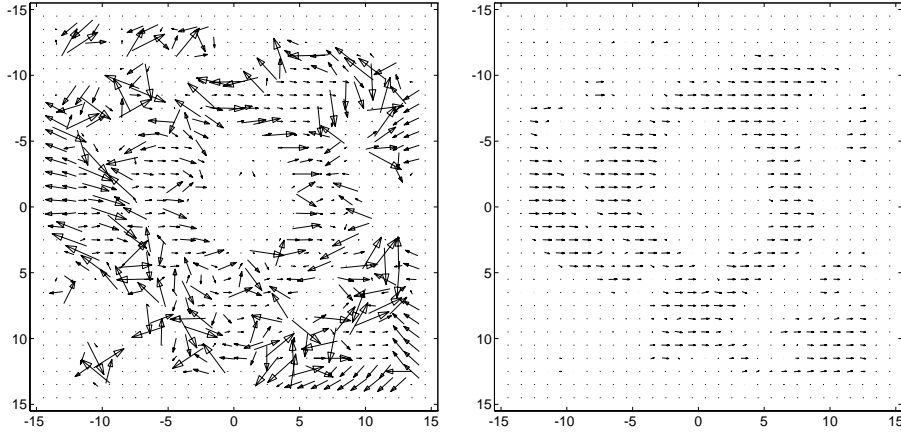
**Figure 5.9  Left**: Standard convolution. We can see that missing data completely destroy many of the estimates. **Right**: Normalized differential convolution on the velocity field in figure 5.8 produces a field pointing to the right. If the camera had had the reversed motion, i.e. a motion away from the scene, the output field would have pointed to the left. Pure rotation of the camera would have produced a vector field pointing up or down depending on the direction of rotation.

convolution scheme has the ability to disregard the information where the certainty has been set to zero. This is *not* the same thing as removing estimates after the operation has been run, which often creates problems for succeeding operations.

The approximate position of the robot arm can be derived from its pose control data. Information about the precise delineation between the arm and the background can then be estimated from the image. In the present implementation this is done and used for creating a certainty field matching the projections (stereo) of the arm. A certainty field can thus be used to mask out already modelled structures, allowing other structures to be detected. A series of experiments where the robot's own arm is masked has been carried out using a simulated robot working in a conveyor belt environment. Figure 5.10 shows that although the arm is continuously moving, only the moving boxes on the conveyor belt are detected.
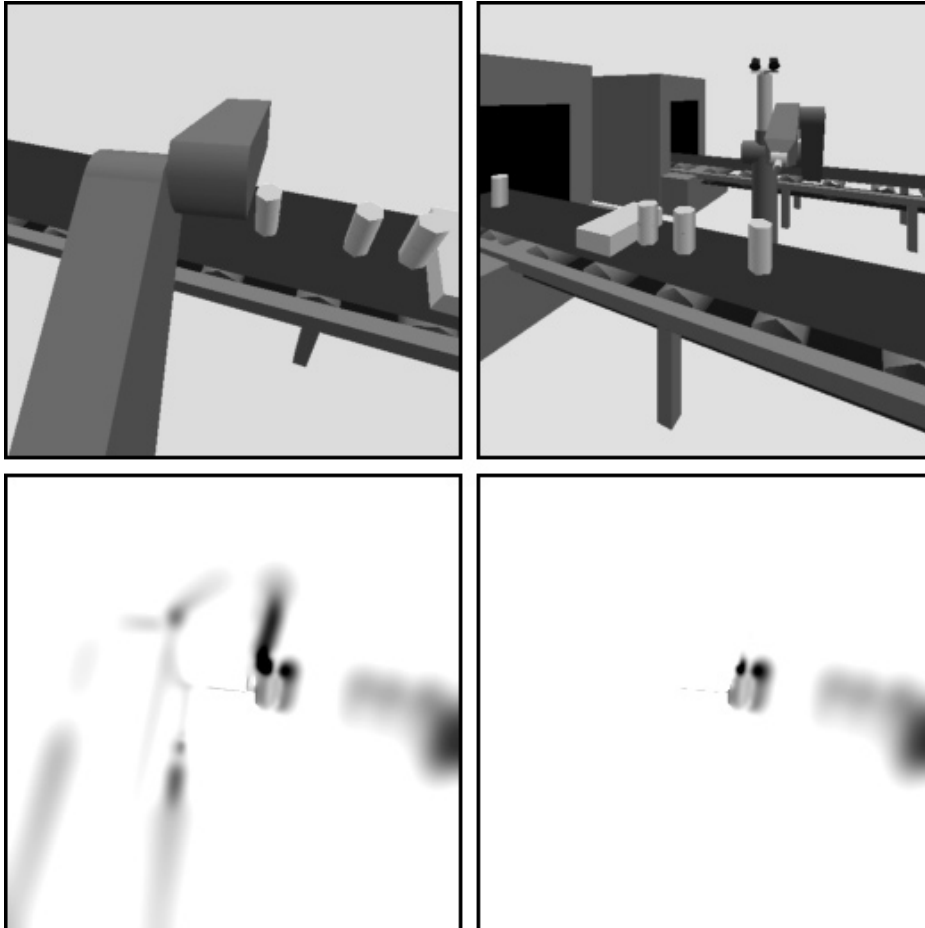
**Figure 5.10   Top left**: The robot is looking down at the conveyor belt, where the objects are moving from right to left, while the robot arm is moving back and forth. **Top right**: An overview showing the conveyor belt, the robot arm points towards the viewer and the stereo camera head is looking down on the conveyor belt. **Bottom left**: Traditional spatio-temporal filtering detecting all moving objects in the scene including the arm. **Bottom right**: Spatio-temporal filtering using normalized differential convolution and a zero certainty mask covering the robot arm, detecting only the unknown moving objects. This algorithm was implemented by Westelius.

## 5.4   NORMALIZED CONVOLUTION IN SUBSPACES

This section addresses the problem of how a *subset* of the coordintes $f^i$ of a signal described in a filter basis $\mathbf{f} = f^i\mathbf{b}_i$ can be estimated. Images are normally

described locally using simple basis functions. Low order local momentum such as order 0 (the local DC component), 1 and 2 are commonly used. Low order differentiations are also useful descriptors. In densely regularly sampled images, these descriptors are easily computed using standard convolution. However, when working with irregularly sampled data or incomplete data, the signal model has to be of higher order than the signal variations of interest. This is the case where only a part of the parameter vector is to be estimated. If possible, only this part of the parameter should be calculated explicitly as opposed to calculating the whole parameter vector. In this section a method based on partitioning the model subspace into two parts is described.

## 5.4.1   Signal spaces

A sampled system can be interpreted as defining a multidimensional signal space (figure 3.2). We denote this space $\mathcal{E}$ and let $\{\mathbf{e}_i\}$ be a basis spanning this space. The basis operator having these vectors as column vectors is denoted

$$\mathbf{E} = (\,\mathbf{e}_1 \quad \mathbf{e}_2 \quad ... \quad \mathbf{e}_n\,) \tag{5.34}$$

In general, information in images is locally low-dimensional. The tradition of extracting useful images information using gradient operators, implicitly incorporate the assumption that the images can locally be modelled as being one-dimensional. In this case it is obvious that the standard pixel description of this neighbourhood is an overkill. A signal only varying in one dimension does not need 7 by 7, i.e. 49 dimensions, to be described properly.

Expressing new vectors $\mathbf{b}_i$ in the basis $\mathbf{e}_i$ gives $n$-dimensional vectors. We define the basis operator, $\mathbf{B}$, having these $n$-dimensional basis vectors as column vectors:

$$\mathbf{B} = \begin{pmatrix} . & . & ... & . \\ . & . & ... & . \\ \mathbf{b}_1 & \mathbf{b}_2 & ... & \mathbf{b}_m \\ . & . & ... & . \\ . & . & ... & . \end{pmatrix} \tag{5.35}$$

It should be mentioned that the complexity of the calculation of a new transformation matrix is based on the number of new basis functions, and not, as in this particular case, on the number of dimensions in the huge sample basis $\{\mathbf{e}_n\}$.

In general the new basis $\{\mathbf{b}_i\}$ span a subspace of $\mathcal{E}$ Let this subspace be denoted $\mathcal{B}$. Let $O$ denote the oblivious space, the space not describable with the new basis. This space is the orthogonal complement to $\mathcal{B}$

$$\mathcal{E} = [\mathcal{B}, \mathcal{O}] \tag{5.36}$$

Naturally, the signal can only be hoped for being well described if the range space of $\mathcal{E}$ lies in $\mathcal{B}$. The part of the signal space which we are interested in is denoted $\mathcal{S}$, and the part of the signal space we want to be invariant to is denoted $\mathcal{L}$.

$$\mathcal{B} = [\mathcal{S}, \mathcal{L}] \tag{5.37}$$

A visualization of the spaces introduced is shown in figure 5.11.
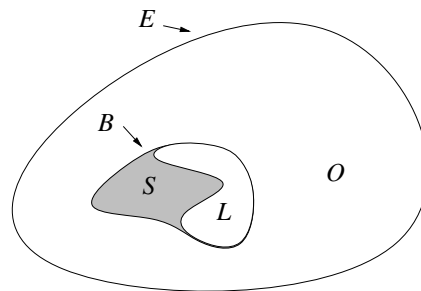


**Figure 5.11**  The signal of interest lies in the subspace $\mathcal{S}$.

## 5.4.2   Matrix form of normalized convolution

In [WNK94] the relation between normalized convolution and normalized differential convolution was described. It was shown that the estimated coordinate vectors is equal in all components except the additional parameter defining the local DC component for normalized convolution. The definition of normalized convolution may be written in matrix form [WK94]:

$$\mathbf{U} \;=\; \mathbf{G}^{-1}\mathbf{D} \;=\; (\mathbf{B}^*\mathbf{ACB})^{-1}\,\mathbf{B}^*\mathbf{ACf} \tag{5.38}$$

where $\mathbf{A}$ is an $n \times n$ diagonal matrix containing an applicability function, $\mathbf{C}$ is an $n \times n$ diagonal matrix containing the certainties of the signal, and $\mathbf{f}$ denotes

the signal. Note that the expression in equation (5.38) not is equal to the More-Penrose invers of $\mathbf{ACB}$ which would have squared matrices, $A$ and $C$, in the part corresponding to the metric in this equation.

The space $\mathcal{L}$ in $\mathcal{B}$ is the space we want to be invariant to and $\mathcal{S}$ is the part we want to describe. This can be achieved by dividing the basis operator $\mathbf{B}$ in two parts, $\mathbf{B} = (\mathbf{S} \quad \mathbf{L})$. Inserting this expression on equation (5.38) yields

$$\begin{pmatrix} \mathbf{U}_S \\ \mathbf{U}_L \end{pmatrix} = \begin{pmatrix} \mathbf{S}^*\mathbf{ACS} & \mathbf{S}^*\mathbf{ACL} \\ \mathbf{L}^*\mathbf{ACS} & \mathbf{L}^*\mathbf{ACL} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{S}^*\mathbf{A} \\ \mathbf{L}^*\mathbf{A} \end{pmatrix} \mathbf{Cf} \tag{5.39}$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_S \\ \mathbf{U}_L \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{S}^*\mathbf{ACS} & \mathbf{S}^*\mathbf{ACL} \\ \mathbf{L}^*\mathbf{ACS} & \mathbf{L}^*\mathbf{ACL} \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{S}^*\mathbf{A} \\ \mathbf{L}^*\mathbf{A} \end{pmatrix} \mathbf{Cf}$$

The inverse of a block matrix, $\mathbf{G}$, [Kai80] may be rewritten as:

$$\mathbf{G}^{-1} = \begin{pmatrix} \Delta_S^{-1} & -\Delta_S^{-1}\mathbf{X} \\ -\Delta_L^{-1}\mathbf{Y} & \Delta_L^{-1} \end{pmatrix} \tag{5.40}$$

where

$$\Delta_S = \mathbf{S}^*\mathbf{ACS} - \mathbf{S}^*\mathbf{ACL}(\mathbf{L}^*\mathbf{ACL})^{-1}\mathbf{L}^*\mathbf{ACS}$$

$$\Delta_L = \mathbf{L}^*\mathbf{ACL} - \mathbf{L}^*\mathbf{ACS}(\mathbf{S}^*\mathbf{ACS})^{-1}\mathbf{S}^*\mathbf{ACL}$$

$$\mathbf{X} = \mathbf{S}^*\mathbf{ACL}(\mathbf{L}^*\mathbf{ACL})^{-1}$$

$$\mathbf{Y} = \mathbf{L}^*\mathbf{ACS}(\mathbf{S}^*\mathbf{ACS})^{-1}$$

which gives an explicit expression of the parameter vector of interest,

$$\mathbf{U}_S = \Delta_S^{-1}(\mathbf{S}^*\mathbf{A} - \mathbf{XL}^*\mathbf{A})\mathbf{Cf} \tag{5.42}$$

$$= \Delta_S^{-1}(\mathbf{S}^*\mathbf{A} - \mathbf{S}^*\mathbf{ACL}(\mathbf{L}^*\mathbf{ACL})^{-1}\mathbf{L}^*\mathbf{A})\mathbf{Cf}$$

A special case of equation (5.41) is when the subspace $\mathcal{S}$ id invariant only to constant functions. Then $\mathbf{L}$ is a vector, which means that $(\mathbf{L}^*\mathbf{ACL})$ is a scalar that can extracted,

$$
\begin{aligned}
\mathbf{U}_S &= (\Delta_S\mathbf{L}^*\mathbf{ACL})^{-1}(\mathbf{L}^*\mathbf{ACLS}^*\mathbf{A} - \mathbf{S}^*\mathbf{ACLL}^*\mathbf{A})\mathbf{Cf} \\
&= \mathbf{G}_S^{-1}\mathbf{D}_S
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{D}_S &= \mathbf{L}^*\mathbf{ACLS}^*\mathbf{ACf} - \mathbf{S}^*\mathbf{ACLL}^*\mathbf{ACf} \\
\mathbf{G}_S &= \mathbf{L}^*\mathbf{ACLS}^*\mathbf{ACS} - \mathbf{S}^*\mathbf{ACLL}^*\mathbf{ACS}
\end{aligned}
$$

This case corresponds to normalized differential convolution [KW93].

In this section it has been described how a subset of the coordinate vector describing a signal can be estimated. The basis operator is divided into two parts: one part describing the space of interest, and one part not of interest. This is a generalization of normalized differential convolution [KW93] which is invariant to any constant term in the input signal. With the subspace formulation in, the algorithm can be made invariant to any subspace in $\mathcal{B}$, not only the one-dimensional space of constant functions. For example, operations invariant to everything *but* the constant function can be constructed. This is useful when the local DC component of the signal is to be estimated as in for example interpolation.

# 6

# LOCAL STRUCTURE AND MOTION

## 6.1 INTRODUCTION

The techniques presented in this chapter are based on spatial or spatio-temporal filtering and tensor representations [GK94]. For each neighbourhood a set of quadrature filters are employed. The filters are sensitive to signals having different orientations in the signal space. Each quadrature filter produces a magnitude and a phase as output. The phase can be used to describe local symmetry properties of the signal. The magnitudes of the different filters are combined to give a tensor description of local structure. In two dimensions, the structure is described in terms of dominant local orientation and isotropy, where isotropy means lack of dominant orientation. In three dimensions, in addition to isotropy, the tensor contains information about how planar/linear the neighbourhoods are. In higher dimensions, the basic shapes are more complicated than lines and planes, and the different possible anisotropies become complex.

In section 6.4 velocity estimation from image sequences will be discussed. The method is based on local spatio-temporal structure estimation [Knu85, Knu89, BHKG91]. Viewing a time-sequence of two-dimensional images as a three-dimensional volume makes it possible to estimate local velocity. In neighbourboods of this volume that are constant on lines, true velocity may be estimated. In neighbourhoods that are constant on planes, only the velocity component perpendicular to the local structure can be measured. This corresponds to the well-known aperture problem. It is shown how tensors describing planar structures can be combined into tensors containing information of the true local motion.

An underlying assumption for the approach taken in this chapter is the fact that local gradient directions contain important information. Equally important is the *local simplicity hypothesis* [Gra78]. The basis for this hypothesis is that the spatial variation of the gradient directions is in general much slower than the spatial

variation of the image itself. The importance of local orientation descriptions in the mammalian visual system has been demonstrated by the physiological findings of Hubel and Wiesel and others [HW78].

## 6.2 LOCAL ORIENTATION TENSOR

Knutsson [Knu89] has shown that a tensor representation of *local orientation* can be produced by combining the outputs from polar separable quadrature filters. Regardless of dimensionality, a representation tensor $\mathbf{T}$ for orientation is given by

$$\mathbf{T} \equiv A\hat{\mathbf{x}}\hat{\mathbf{x}}^T \tag{6.1}$$

where $A > 0$ is an arbitrary constant and $\hat{\mathbf{x}}$ is a vector pointing along the orientation of maximal signal variation. The size of the tensor will depend on the dimensionality of the signal, e.g. three-dimensional signals are described by a $3 \times 3$ tensor.

A *simple neighbourhood* of $n$ dimensions is expressed as:

$$\mathbf{S}(\boldsymbol{\xi}) = \mathbf{G}(\boldsymbol{\xi}^T\hat{\mathbf{x}}) \tag{6.2}$$

where $\mathbf{S}$ is an $n$-variable function and $\mathbf{G}$ is a single variable function. $\boldsymbol{\xi}$ is the spatial position vector and $\hat{\mathbf{x}}$ is a constant vector oriented along the axis of maximal signal variation. In three dimensions, simple functions are constant on parallel planes. The direction perpendicular to these planes is the direction of maximum signal variation, $\hat{\mathbf{x}}$, or "the direction" of the signal. Examples of three-dimensional simple signals are shown in figure 6.1.

### 6.2.1 Quadrature filters

A *quadrature filter* can, independently of the dimensionality of the signal space, be defined as a filter that is zero over one half of the Fourier space. In the spatial domain, the filter is complex: an even real part and an odd imaginary part. In two dimensions, the even part can be seen as a line filter, and the odd part as an edge filter [Knu82, KP94],.

The *lognormal filter* is a spherically separable quadrature pair filter with a radial frequency function that is Gaussian on a *logarithmic* scale.

$$Q(\mathbf{u}) = R(\rho)D_k(\hat{\mathbf{u}}) \tag{6.3}$$

$$\mathbf{S}_a(\boldsymbol{\xi}) \ = \ \mathbf{G}_1(\boldsymbol{\xi}^T \mathbf{x}_1) \qquad\qquad \mathbf{S}_d(\boldsymbol{\xi}) \ = \ \mathbf{G}_2(\boldsymbol{\xi}^T \mathbf{x}_2)$$
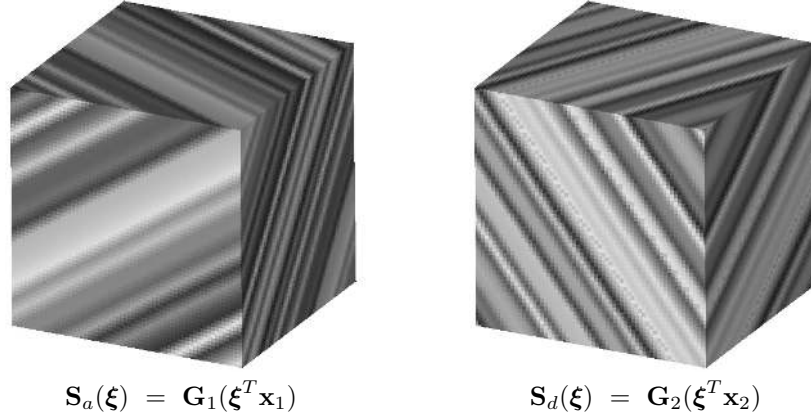
**Figure 6.1** Two different three-dimensional simple neighbourhoods. The neighbourhoods are constructed using two different signal functions ($\mathbf{G}_1$ and $\mathbf{G}_2$) and two different signal orienting vectors ($\mathbf{x}_1$ and $\mathbf{x}_2$).

where $\mathbf{u}$ is the multidimensional frequency variable, $R_{(\rho)}$ and $D_k(\hat{\mathbf{u}})$ are the radial and the directional function respectively.

*Directional functions* having the necessary properties for directional interpolation were first suggested by Knutsson in [Knu82] for the 2D case and in [Knu85] for the 3D case. These functions are written

$$\begin{cases} D_k(\hat{\mathbf{u}}) \ = \ (\hat{\mathbf{u}}^T \hat{\mathbf{n}}_k)^2 & \text{if } \mathbf{u}^T \hat{\mathbf{n}}_k \ > \ 0 \\ D_k(\hat{\mathbf{u}}) \ = \ 0 & \text{otherwise} \end{cases} \qquad (6.4)$$

where $\hat{\mathbf{n}}_k$ is the filter directing vector, i.e. $D(\hat{\mathbf{u}})$ varies as $\cos^2(\varphi)$, where $\varphi$ is the angle between $\mathbf{u}$ and the filter direction $\hat{\mathbf{n}}_k$.

If the input signal is simple with the direction $\hat{\mathbf{x}}$, the output magnitude from a quadrature filter in direction $k$ is

$$q_k \ = \ d(\hat{\mathbf{x}}^T \hat{\mathbf{n}}_k)^2 \ = \ d\langle \hat{\mathbf{x}}\hat{\mathbf{x}}^T, \hat{\mathbf{n}}_k\hat{\mathbf{n}}_k^T \rangle \qquad (6.5)$$

where $d$ is independent of the filter orientation and depends only on radial distribution of the signal spectrum and the *radial filter function.*

The radial function $R(\rho)$ can be chosen arbitrarily without violating the directional interpolation properties of the filter. This makes the choice of $R(\rho)$ subject

to considerations similar to those traditionally found in one-dimensional filter design. A radial filter function with useful properties [Knu89] is given by

$$R(\rho) \;=\; e^{-\frac{4}{B^2 ln2}ln^2(\rho/\rho_0)} \tag{6.6}$$

These functions are termed *lognormal functions*. $B$ is the relative bandwidth and $\rho_0$ is the centre frequency of the filter. This type of filter is also used in chapter 8 for the estimation of local frequency.

## 6.2.2 Construction of the orientation tensor

The minimum number of quadrature filters necessary for orientation estimation is three for the two-dimensional case. An example of filter orientations is given by vectors pointing to three adjacent vertices of a regular hexagon [Knu82].

In three dimensions, the minimum required number of quadrature filters is six [Knu89]. The orientations of the filters are given for example by vectors pointing to the vertices of a *hemi-icosahedron* [Knu85]. The orientation tensor $\mathbf{T}$ is obtained by a linear summation of the quadrature filter output magnitudes,

$$\mathbf{T} \;=\; \mathbf{N}^k q_k \tag{6.7}$$

were $\mathbf{N}^k$ is a *dual tensor basis* corresponding to a *tensor basis* $\mathbf{N}_k \;=\; \hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^T$, where $\hat{\mathbf{n}}_k$ are the filter directing vectors and $q_k$ is the output magnitude from the quadrature filter $k$. The expression of the dual basis is

$$\mathbf{N}^k \;=\; \langle \mathbf{N}_k, \mathbf{N}_l \rangle^{-1} \mathbf{N}_l \tag{6.8}$$

since

$$\langle \mathbf{N}^k, \mathbf{N}_j \rangle \;=\; \langle \mathbf{N}_k, \mathbf{N}_l \rangle^{-1} \langle \mathbf{N}_l, \mathbf{N}_j \rangle \;=\; \delta_j^k \tag{6.9}$$

which fulfills the dual basis requirement (equation (2.23)).

In the case of the filter directions pointing to the vertices of a hemi-icosahedron, the dual basis, corresponding to the tensor basis $\hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^T$, has the regular expression of [Knu89]

$$\mathbf{N}^k \;=\; \alpha \hat{\mathbf{n}}_k \hat{\mathbf{n}}_k^T \;-\; \beta \mathbf{I} \tag{6.10}$$

where $\mathbf{I}$ is the identity tensor and $\alpha$ is $\frac{4}{3}$ for 2D and $\frac{5}{4}$ for 3D. The parameter $\beta$ is $\frac{1}{3}$ for 2D and $\frac{1}{4}$ for 3D.

Note that the dual tensor basis is constant and can be pre-calculated. Thus, the orientation tensor can be estimated by weighted summation of the fixed tensors $\mathbf{N}^k$, the weights being the quadrature filter output magnitudes $q_k$.

Acquired data are seldom simple in the sense of equation (6.2). It is, however, still possible to find a best approximation to $\mathbf{T}$ corresponding to a simple neighbourhood, equation (6.2). This is done by finding the $\mathbf{T}_s$ that minimizes

$$\Delta \;=\; \|\mathbf{T} - \mathbf{T}_s\| \tag{6.11}$$

where $\mathbf{T}_s = A\hat{\mathbf{x}}\hat{\mathbf{x}}^T$. The tensor most similar to $\mathbf{T}$ corresponding to a simple neighbourhood is given by:

$$\mathbf{T}_s \;=\; \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \tag{6.12}$$

where $\lambda_1$ is the largest eigenvalue of $\mathbf{T}$ and $\hat{\mathbf{e}}_1$ is the corresponding eigenvector. The value of $\Delta$ indicates how well the one-dimensionality hypothesis fits the neighbourhood. The smaller the value, the better the fit.

For two-dimensional signals, it is possible to represent the *dominant orientation* of a neighbourhood using a two-component vector [Gra78]. This vector can be constructed using the same principles as for the tensor contruction, i.e. by multiplying the quadrature outputs with fixed elements and summing the results [Knu82].

$$\mathbf{z} \;=\; \hat{\mathbf{m}}^k q_k \tag{6.13}$$

where

$$\hat{\mathbf{m}}_k = \begin{pmatrix} \cos(2\varphi_k) \\ \sin(2\varphi_k) \end{pmatrix} \qquad \text{and} \qquad \varphi_k = \arg(\hat{\mathbf{n}}_k) \tag{6.14}$$

The minimum number of quadrature filters needed is the same as for the tensor case, i.e. three filters.

### 6.2.3   Geometric interpretation

In two dimensions, the *orientation tensor* can be visualized using an ellipse where the principal axes correspond to the directions of the eigenvector system (figure 6.2).

$$\mathbf{T} \;=\; \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T$$
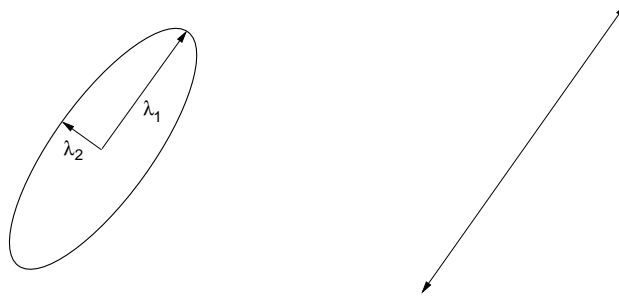


**Figure 6.2   Left**: Geometric representation of a rank 2 tensor as an ellipse. **Right**: Geometric representation of a rank 1 tensor: since the smallest eigenvalue is zero, the ellipse has collapsed to a line. Two arrows have been added indicating that it is an outer product of a vector which can be seen as an "unsigned" vector.

The relation between the *double angle vector* representation in equation (6.13) and the tensor representation in equation (6.7) is shown in figure 6.3.

A geometric interpretation of the relationship between the eigenvectors of a 2D tensor and a double angle description can be made in the following way. Suppose that we have a tensor:

$$\mathbf{T} \;=\; \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T \tag{6.15}$$

where $\lambda_n$, $\mathbf{e}_n$ denote the eigenvalues and eigenvectors of $\mathbf{T}$, respectively. When the tensor is of rank 1, only one of the eigenvalues is greater than 0,

$$\mathbf{T} \;=\; \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T \quad \leftrightarrow \quad \lambda_1 \begin{pmatrix} e_{1x}^2 & e_{1x} e_{1y} \\ e_{1x} e_{1y} & e_{1y}^2 \end{pmatrix}, \tag{6.16}$$

In this special case, proving the validity of the double angle method for extracting the direction of the eigenvectors is straightforward since

$$2\varphi_1 \;=\; 2\arg\,(e_{1x}, e_{1y}) \tag{6.18}$$
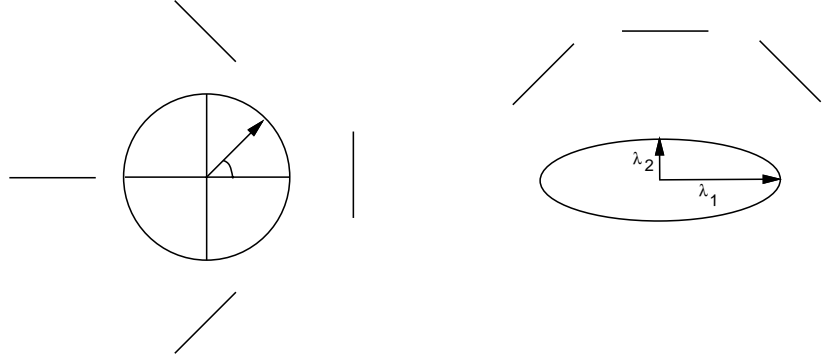
**Figure 6.3** Representation of orientation in two dimensions. **Left**: A double angle vector representation of local orientation. **Right**: A tensor representation of local orientation. The largest eigenvector of the tensor indicates the local orientation (in single angle), and the relation between the eigenvalues expresses the certainty of the statement. The larger $\lambda_1$ is compared to $\lambda_2$ the higher the certainty ($\lambda_1 \geq \lambda_2$).

$$= \quad \arg\,(e_{1x}^2 - e_{1y}^2, 2e_{1x}e_{1y}) \tag{6.19}$$

$$= \quad \arg\,(\cos(\varphi_1)^2 - \sin(\varphi_1)^2, 2\cos(\varphi_1)\sin(\varphi_1)) \tag{6.20}$$

$$= \quad \arg\,(\cos(2\varphi_1), \sin(2\varphi_1)). \tag{6.21}$$

This relationship applies separately for both the tensor $\lambda_1\mathbf{e}_1\mathbf{e}_1^T$ and the tensor $\lambda_2\mathbf{e}_2\mathbf{e}_2^T$ in equation (6.15). However, the reason that this simple formula holds in general, i.e. even if the tensor has more than one eigenvalue greater than zero, is that the two eigenvectors are actually anti-parallel in the $2\varphi$-domain. Figure 6.4 shows this relationship [Wes91]. The left side of this figure illustrates the direction of the two eigenvectors of a tensor; on the right the corresponding double angle representations of these vectors show that they are actually anti-parallel. This means that the resultant vector in the $2\varphi$-domain has the same direction for all $\lambda_2$. Only its length, $(\lambda_1 - \lambda_2)$, varies. The figure also illustrates that the method collapses when $\lambda_1 = \lambda_2$. However, in that case the tensor is completely circular, which means that *all* vectors are eigenvectors.

We have shown that, in two dimensions, the double angle vector descriptor and the tensor descriptor give the same result. The only difference is that the tensor descriptor has an additional statement of the *local energy*, the norm of the tensor. Both methods provide certainty estimates. In the tensor case, the relation between the eigenvalues gives an estimate of the certainty [GK94]. In the double angle method these estimates are mixed in the length of the produced vector. The similarities between the methods show, however, that the double angle con-
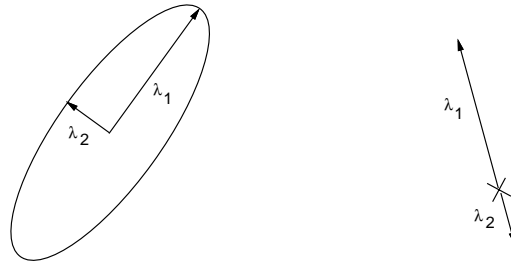
**Figure 6.4   Left**: Geometric representation of a tensor. **Right**: The two eigen-vectors expressed in double angle. Note that they are anti-parallel.

cept is a fundamental form for description of directional data in 2D. In higher dimensions, however, the tensor formulation must be used.

Discussions concerning filtering of four-dimensional signals using the orientation tensor framework can be found in [KGB90, Bår91, GK94].

In three dimensions, a visualization of a symmetric tensor can be made similarly using an ellipsoide. Suppose we have a tensor

$$\mathbf{T} \;=\; \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T$$

An ellipsoide having principal axes as the eigenvector system may be used for describing $\mathbf{T}$. However, the exact shape of an ellipsoide is difficult to interpret if only shading is used for depth cueing (figure 6.5, left). Usually motion is necessary for fully perceiving the shape of a two-dimensional projection of an ellipsoide.

An alternative is to use an object which can be seen as a sum of a spear, a plate and a sphere. The spear describes the principal direction of the tensor $\lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T$ where the length is proportional to the largest eigenvalue, $\lambda_1$. The plate describes the plane spanned by the eigenvectors corresponding to the two largest eigenvalues, $\lambda_2(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T)$. The sphere with a radius proportional to the smallest eigenvalue shows how isotropic the tensor is, $\lambda_3(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T)$. This type of object is shown in figure 6.5 (right). See also the cover of this book.
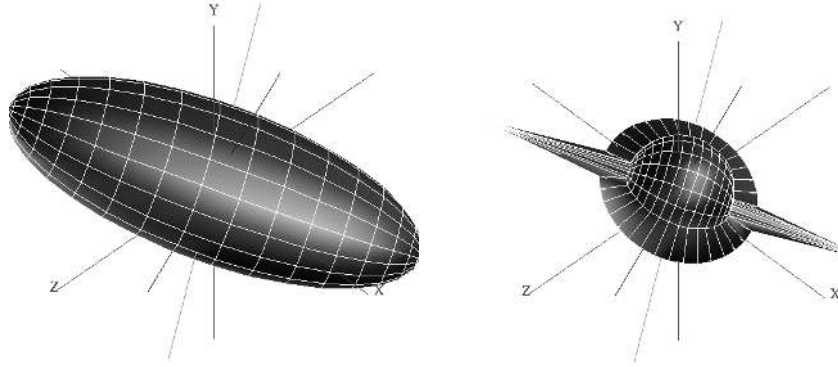
**Figure 6.5** Two ways of representing a symmetric tensor of rank 3 tensor. **Left**: An ellipsoide having principal axes as the eigenvector system of the tensor. **Right**: Alternative representation using a spear-plate-sphere object. Experience has shown that the ellipsoide representation is more difficult to interpret compared to the proposed one.

## 6.2.4 Higher rank neighbourhoods

Simple neighbourhoods are represented by tensors, $\mathbf{T}_s$, having rank 1. In higher-dimensional data there exist highly structured neighbourhoods that are not simple. The rank of the representation tensor reflects the complexity of the neighbourhood. The eigenvalue distributions and the corresponding tensor representations are given below for three particular cases of $\mathbf{T}$ for the three-dimensional case. The relations between the local spatial auto-correlation function and the corresponding energy distributions in the Fourier domain can be studied in figure 6.6.

Let $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ be the eigenvalues of a tensor $\mathbf{T}$ and let $\hat{\mathbf{e}}_i$ be an eigenvector corresponding to $\lambda_i$.

**Plane case**: A rank 1 or simple neighbourhood where $\lambda_1 \gg \lambda_2 \simeq \lambda_3$.

$$\mathbf{T} \simeq \lambda_1 \mathbf{T}_1 = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \tag{6.21}$$

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

**Line case**: A rank 2 neighbourhood where $\lambda_1 \simeq \lambda_2 \gg \lambda_3$.

$$\mathbf{T} \simeq \lambda_1 \mathbf{T}_2 = \lambda_1 \left( \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T \right) \tag{6.22}$$

**Spatial domain**                    **Fourier domain**
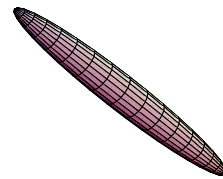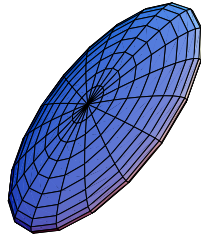


**Plane case:** A planar auto-correlation function in the spatial domain corresponds to energy being distributed on a line in the Fourier domain.
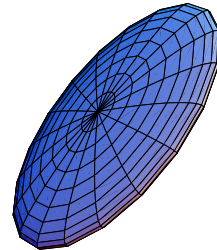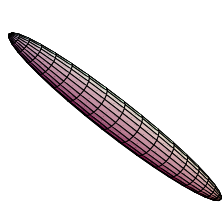


**Line case:** An auto-correlation function concentrated on a line in the spatial domain corresponds to a planar energy distribution in the Fourier domain.
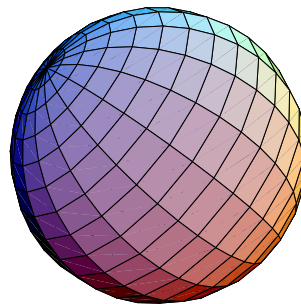


**Isotropic case:** A spherical auto-correlation function in the spatial domain corresponds to a spherical energy distribution in the Fourier domain.

**Figure 6.6**  Iso-surface plots of spatial auto-correlation functions and corresponding energy distributions in the Fourier domain.

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

**Isotropic case**: A rank 3 neighbourhood where $\lambda_1 \simeq \lambda_2 \simeq \lambda_3$.

$$\mathbf{T} \simeq \lambda_1 \mathbf{T}_3 = \lambda_1 \ (\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \ + \ \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T \ + \ \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T) \tag{6.23}$$

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

In general, $\mathbf{T}$ will be a linear combination of these cases,

$$
\begin{aligned}
\mathbf{T} &= \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T \\
&= (\lambda_1 - \lambda_2)\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T - \lambda_2 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + (\lambda_2 - \lambda_3)\hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T - \lambda_3 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T \\
&= (\lambda_1 - \lambda_2)\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + (\lambda_2 - \lambda_3)(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T) + \lambda_3(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T)
\end{aligned}
$$

which gives that $\mathbf{T}$ can be expressed as:

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_1 \ + \ (\lambda_2 - \lambda_3)\mathbf{T}_2 \ + \ \lambda_3 \mathbf{T}_3 \tag{6.24}$$

where $(\lambda_1 - \lambda_2)$, $(\lambda_2 - \lambda_3)$ and $\lambda_3$ may be viewed as coordinates of $\mathbf{T}$ in the tensor basis $\mathbf{T}_i$.

## 6.3 ORIENTATION ESTIMATION AND MISSING DATA

In this section local orientation and *phase* are estimated in a missing data example. In two dimensions it is convenient to use four complex quadrature filters [Knu82]. Each of the four quadrature filter has been compensated individually. The filters are regarded as two-dimensional bases containing one even and one odd basis filter (the real and imaginary part of the filters).

For quadrature filter $k \in \{1..4\}$, the expression of normalized differential convolution is:

$$f_k^j \ = \ G_{ijk}^{-1} \ (\langle a, c\rangle\langle a\mathbf{b}_{ik}, c\mathbf{f}\rangle - \langle a\mathbf{b}_{ik}, c\rangle\langle a, c\mathbf{f}\rangle) \tag{6.25}$$

where

$$G_{ijk} = \langle a, c \rangle \langle a\mathbf{b}_{ik}\mathbf{b}_{jk}, c \rangle - \langle a\mathbf{b}_{ik}, c \rangle \langle a\mathbf{b}_{jk}, c \rangle \qquad i, j \in \{1, 2\} \tag{6.26}$$

Note that $G_{ijk}^{-1}$ means inversion over indices $i, j$ while $k$ defines which filter it is.

The certainties associated with the estimated coordinates, $f_k^j$, are contained in $G_{ijk}$. In the examples below, it has shown to work well to use the determinant (over $i, j$) of $G_{ijk}$ as a certainty measure.

$$c_k = \det(G_{ij})_k \tag{6.27}$$

From equation (6.13) we have the expression of local orientation tensor expressed in the dual basis $\mathbf{M}^k$:

$$\mathbf{T} = \mathbf{M}^k q_k \tag{6.28}$$

where $q_k$ are the filter magnitude responses. Using the above certainty measure $c_k$, we define the compensated filter magnitude respons $q_k$ as

$$q_k = c_k \|f^j\|_k \tag{6.29}$$

Inserting this in equation (6.13) gives an expression of the local orientation

$$\mathbf{T} = \mathbf{M}^k c_k \|f^j\|_k \tag{6.30}$$

In figure 6.8 (top left), a grey level test image is shown. In the black areas of this image, data is missing. To the left of this figure, the corresponding certainty field is shown. Estimates of local orientation and local phase are shown in figures 6.9 and 6.10.

The interpretation of the colour coding is shown in figure 6.7 The vectors in correspond to a double angle representation of the eigenvector corresponding to the largest eigenvalue

$$\mathbf{v} \;=\; (\lambda_1 - \lambda_2)e^{i2\mathrm{arg}\mathbf{e}_1}$$

as described in figure 6.4. The phase is calculated by combining the phase outputs for the four complex filter responses $f_k^1 + if_k^2$, (with $i = \sqrt{-1}$) [Knu82].
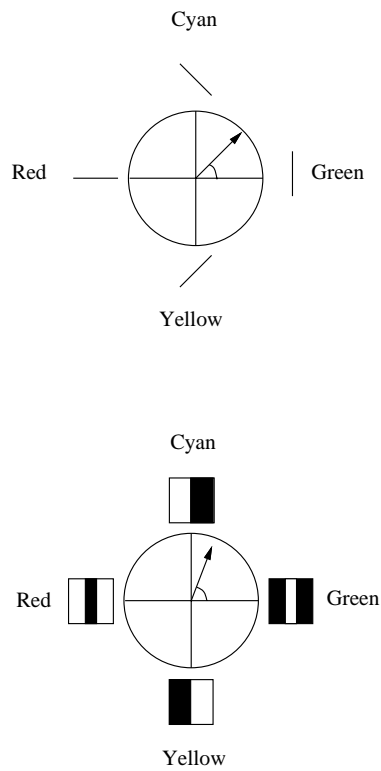
**Figure 6.7**  Colour coding of orientation (top) and phase (bottom).
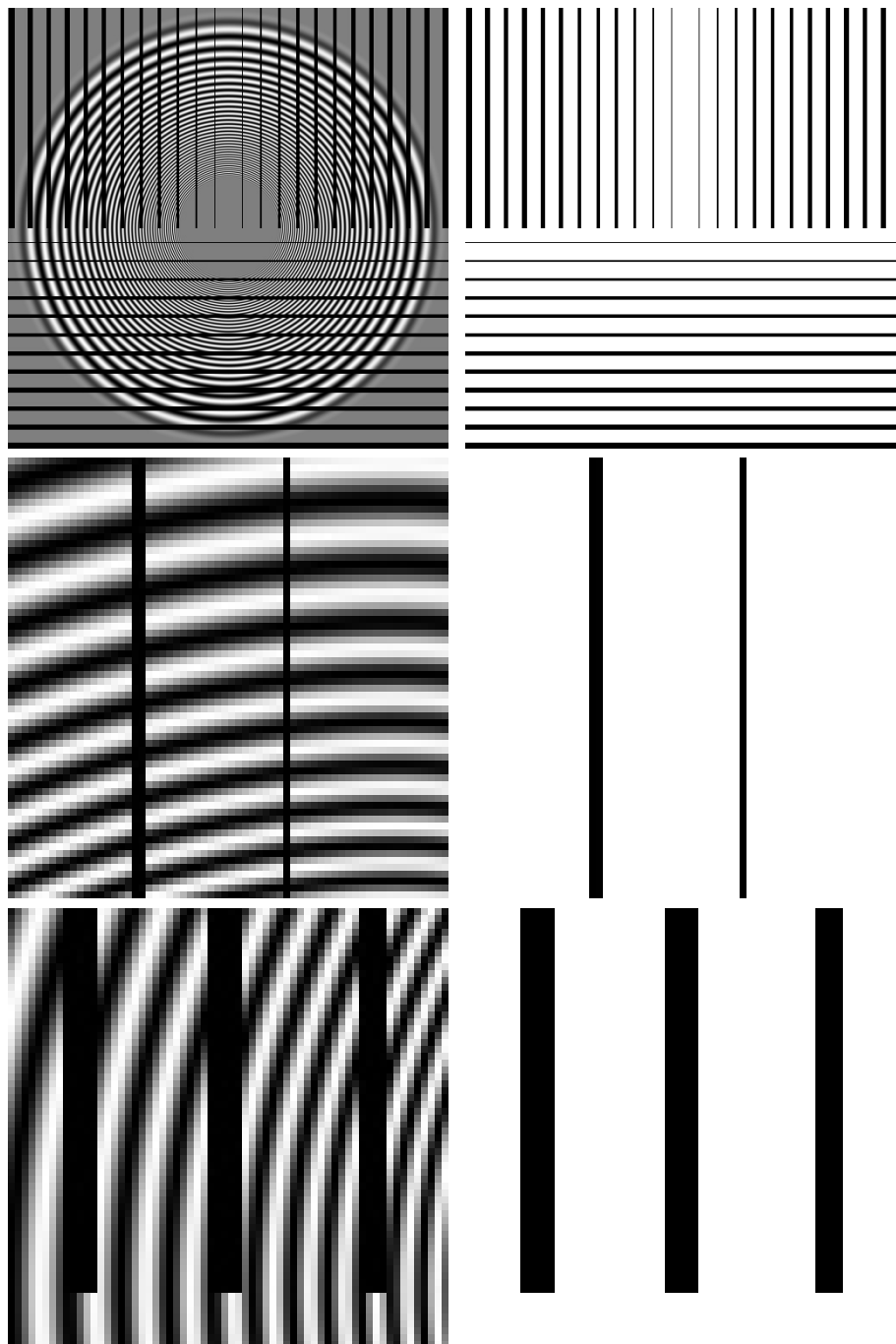
**Figure 6.8** **Top**: A grey level test image multiplied by the corresponding certainty field (left), and certainty field itself (right). **Middle** and **bottom** figures are zoomed parts from the original images.
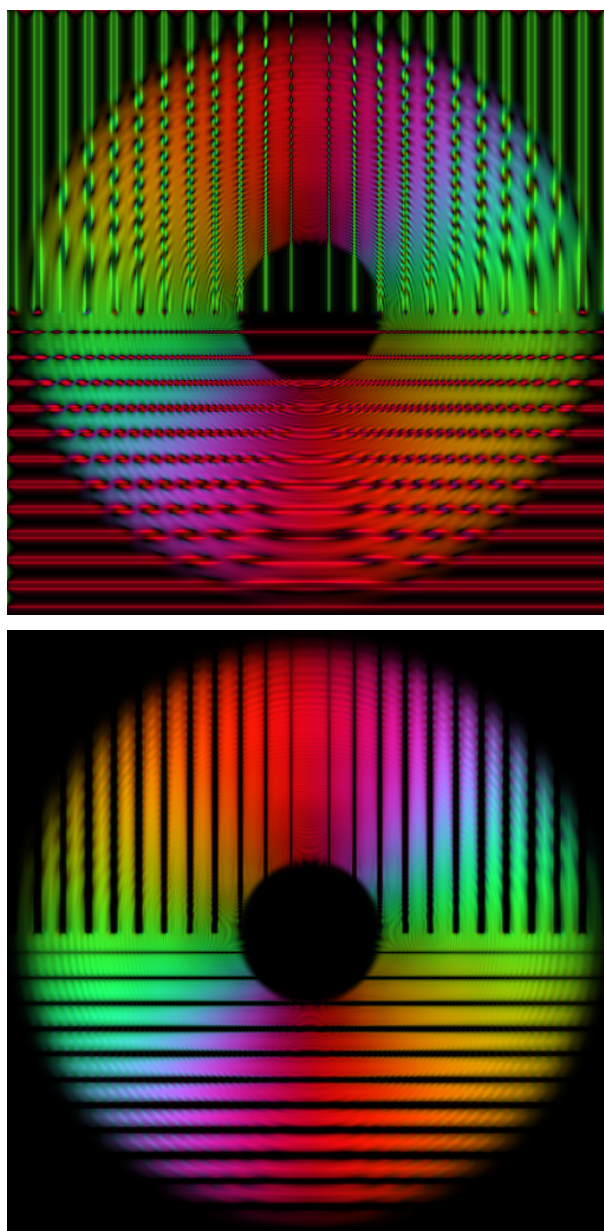
**Figure 6.9**   **Top**: Local orientation estimation of the top left image in figure 6.8 using standard convolution with $15 \times 15$ (complex) quadrature filters. **Bottom**: Local orientation estimation using normalized differential convolution.
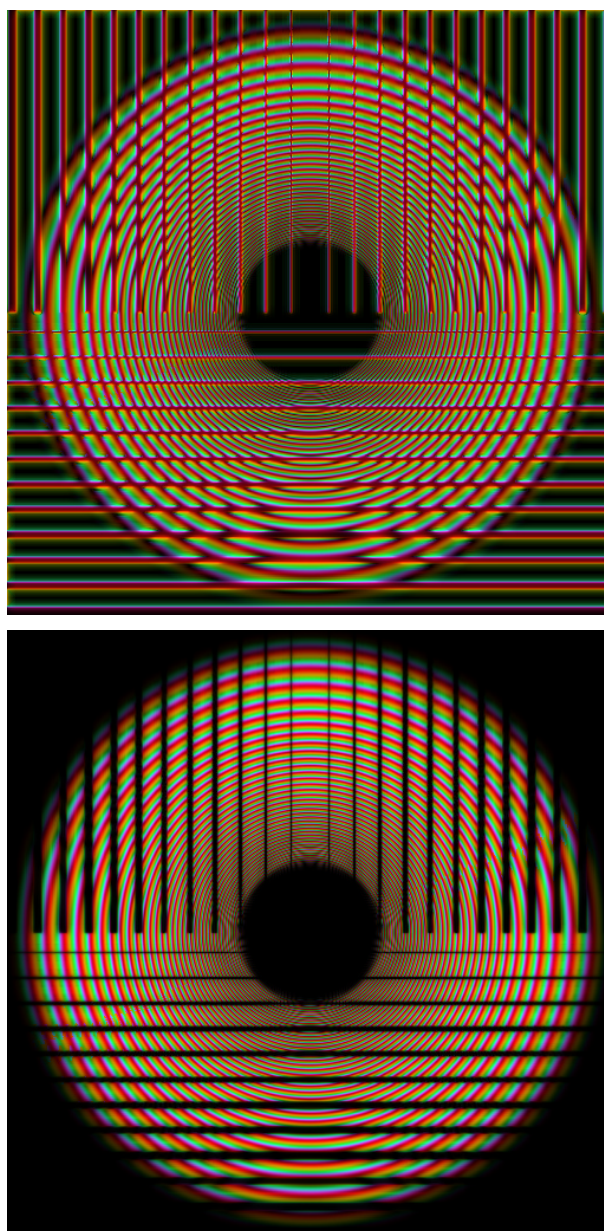
**Figure 6.10**   **Top**: Local phase estimation of the top left image in figure 6.8 using standard convolution with $15 \times 15$ (complex) quadrature filters. **Bottom**: Local phase estimation using normalized differential convolution.

Colour figure

Colour figure

Colour figure

Colour figure
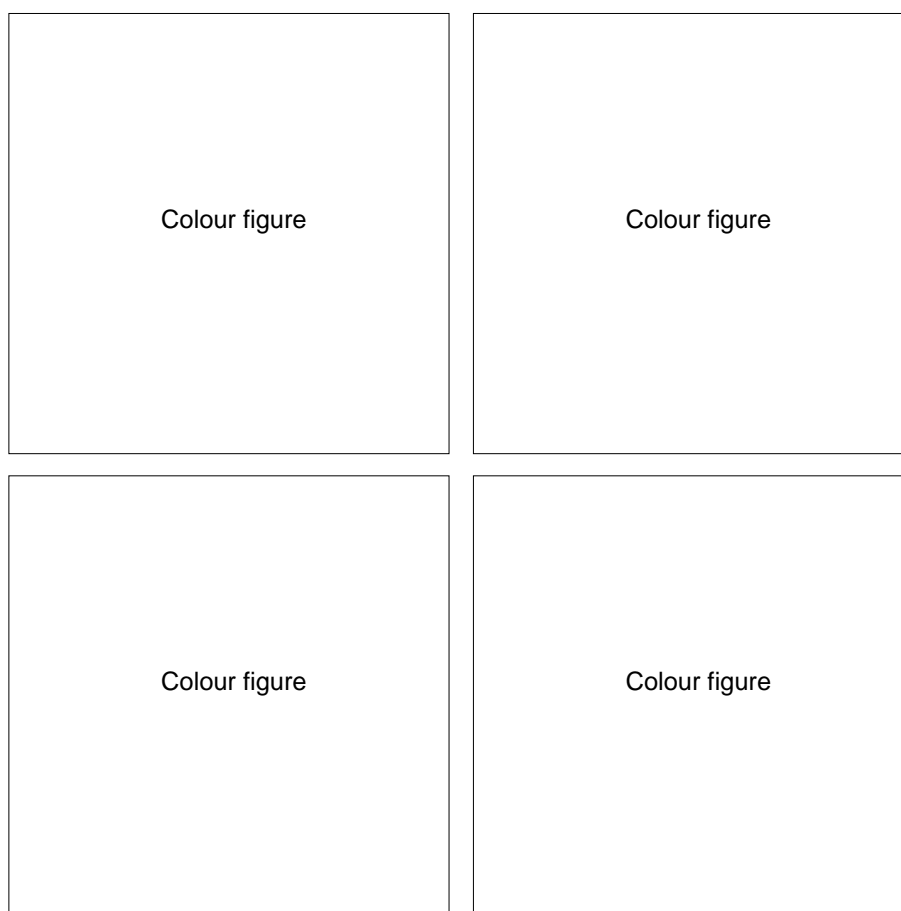
**Figure 6.11**   Parts from figure 6.9 and figure 6.10 where the structure of the missing data is close to perpendicular to the image structure. The top figures are the result from standard convolution. The bottom figures are the result obtained from using normalized convolution. We can see that the normalization compensates effectively for the missing data and makes the estimation insensitive to the structure of the certainty borders

Colour figure

Colour figure

Colour figure

Colour figure
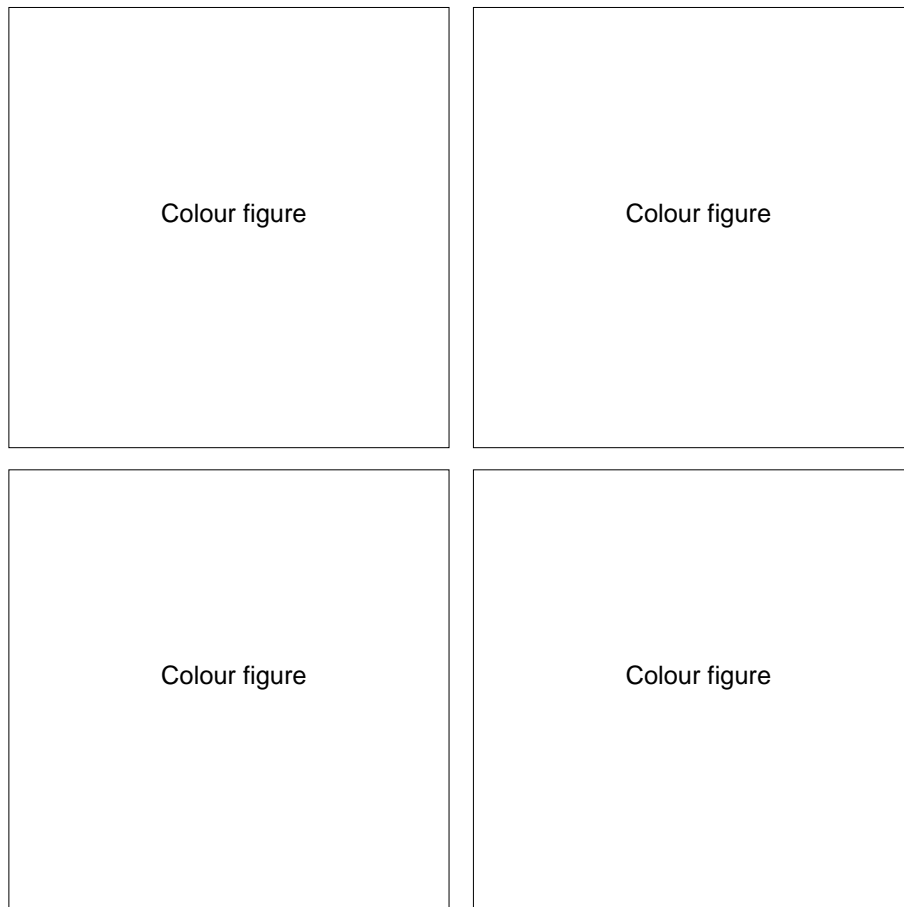
**Figure 6.12** Parts from figure 6.9 and figure 6.10 where the structure of the missing data is close to parallel to the image structure. The top figures are the result from standard convolution. The bottom figures are the result obtained from using normalized convolution. We can see that the normalization compensates effectively for the missing data, even though the orientation of the certainty borders are similar to that of the signal.

## 6.4   MOTION ESTIMATION

This section presents a method for computation of two-dimensional *motion vector fields* from an image sequence [WK94]. The magnitudes from a set of *spatio-temporal quadrature filters* are combined into a tensor description. The shape of the tensors describe, locally, the structure of the spatio-temporal neighbourhood and provides information about *local velocity* and if true flow or only normal flow is present [Knu89, Hag92]. Rather than presenting yet another motion algorithm, we focuses on representation issues of velocity. In chapter 1 it was argued that vectors as a sole description for velocity was not appropriate. The reason is that, since motion fields seldom are dense, artifacts are introduced if the missing data not is compensated for. As mentioned earlier, one way of compensating is to use additional certainty statements. For a discussion on other important aspects of information representation, see [Knu85, GK94]. When a representation of a feature is uniform and continuous, the concept of average feature becomes meaningful. In this section, the effects of averaging an orientation tensor field are discussed. It is shown how normal flow estimates are combined into a true flow using averaging of this tensor field description. An object tracker based on this concept has been developed [KWWK95].

Performance of the original algorithm can be found in [Hag92, GK94]. The quadrature filters used in the estimation were carefully optimized using the principle described in [Knu82].

*Optical flow* computation has been the subject of intensive research during the last decade and a large number of algorithms have been proposed. A recent performance study comparing a number of regularly cited optical flow techniques can be found in [BFB94]. One conclusion from this study is the need for appropriate confidence measures accompanying the estimates.

The data conservation constraint states that the image intensity is preserved locally in space and time in the direction of image motion [LM75, HS81].

$$\frac{df}{dt} \;=\; \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial t} \;=\; 0 \tag{6.31}$$

This constraint can be rewritten as

$$(f_x, f_y)\begin{pmatrix} u \\ v \end{pmatrix} + f_t \;=\; 0 \tag{6.32}$$

where $(f_x, f_y)$ are the components of spatial gradient, $f_t$ is the temporal gradient, and $(u, v)$ are the image velocity components. Lucas and Kanade [LK81] derived

a weighted least squares solution to equation (6.32) by minimizing the sum of *normal constraints* over a local window $a$, i.e.

$$\text{minimize} \quad \sum_i a_i^2 \left[ (f_x, f_y)_i \begin{pmatrix} u \\ v \end{pmatrix}_i + (f_t)_i \right]^2 \tag{6.33}$$

with the solution

$$\mathbf{v}' = (\mathbf{B}^T \mathbf{A}^2 \mathbf{B})^{-1} \mathbf{B}^T \mathbf{A}^2 \mathbf{b} \tag{6.34}$$

where

$$\mathbf{B}^T = \begin{pmatrix} f_{x1}, & \dots, f_{xn} \\ f_{y1}, & \dots, f_{yn} \end{pmatrix}$$

$$\mathbf{A} = \text{diag}(a_1, \dots, a_n)$$

$$\mathbf{b}^T = -(f_{t1}, \dots, f_{tn})$$

and where $\mathbf{v}'$ is the estimated velocity. Similar approaches can be found in [AB85, HJ92, SAH91, TS91].

The flow constraint in equation (6.32) can be rewritten more compactly as

$$(\nabla f)^T \mathbf{v} = 0 \tag{6.35}$$

where $(\nabla f)^T$ is the *spatio-temporal gradient* with components $(f_x, f_y, f_t)$ and $\mathbf{v}$ is a three-dimensional vector representation of the *image velocity* $\mathbf{v} = (u, v, 1)^T$.

Using this notation in equation (6.33) gives

$$\text{minimize} \quad \sum_i a_i^2 ((\nabla f)_i^T \mathbf{v})^2 \tag{6.36}$$

which is equal to

$$\text{minimize} \quad \sum_i a_i^2 \mathbf{v}^T \nabla f_i (\nabla f)_i^T \mathbf{v} \tag{6.37}$$

Let $\mathbf{F}_i$ denote the tensor $\nabla f_i (\nabla f_i)^T$ in spatial positions $\mathbf{x}_i$. Summing these tensors, over a spatial localization window, $a_i$, gives

$$\sum_i a_i^2 \mathbf{v}^T \mathbf{F}_i \mathbf{v} = \mathbf{v}^T \mathbf{F}' \mathbf{v} \tag{6.38}$$

The least squares estimate of the velocity is the eigenvector of $\mathbf{F}'$ corresponding to the *smallest* eigenvalue. Note that this estimate is equivalent to the estimate obtained in equation (6.34). The use of outer products of the gradient in a similar way can be found in [Big88] .

# 6.5 VELOCITY FROM THE ORIENTATION TENSOR

In the previous section the optical flow equation was rewritten in terms of spatio-temporal gradients. This shows that estimating the local spatio-temporal structure is a natural way to estimate velocity in an image sequence. Note that for time sequences, a spatial line in motion generates a spatio-temporal plane, and that a moving spatial point generates a spatio-temporal line [WAJA85, AB85].

Velocity estimates can be obtained directly from the orientation tensor. In *line-like* neighbourhoods, true velocity may be estimated. In *plane-like* neighbourhoods, only the velocity component perpendicular to the local structure can be measured. In this section it is shown how tensors describing plane-like structures, can be combined into tensors containing information of the true local motion.

For moving lines or moving linear structures only the velocity component perpendicular to the structure can be determined. The velocity component along the line is indeterminable since motion in this direction induces no change in the local signal, the so called *aperture problem* [Hor86]. In areas with enough structure, i.e. rank($\mathbf{T}$) $> 1$, the correct velocity can be estimated, see figure 6.13.
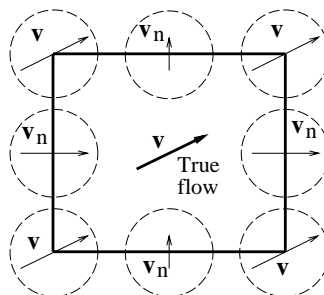


**Figure 6.13** Illustration of the "aperture problem". On moving lines only the normal flow (the true flow projected onto the normal vector of the line) only be estimated. The true flow can be estimated near the corners.

By examining the relations between the eigenvalues of the orientation tensor it is possible to determin to which of the above categories the neighbourhood belongs. The plane case corresponds to a moving line and gives an estimate only for the velocity component perpendicular to the line. The line case corresponds to a moving point and gives an estimate of the velocity in the image plane.

In order to categorize the estimated tensor into the plane case or the line case, the following certainty estimates were used:

$$c_1 = \frac{\lambda_1 - \alpha\lambda_2}{\lambda_1} \quad \text{and} \quad c_2 = \frac{\lambda_2 - \beta\lambda_3}{\lambda_2} \tag{6.39}$$

where $\alpha$ and $\beta$ are constants $> 1$. Negative certainty estimates, $c_i$, in the formulas above is set to zero. A large $\alpha$ sets the certainty to zero for all estimates not corresponding to the plane case. A large $\beta$ sets the certainty to zero for all estimates not corresponding to the line case.

In general, it is more robust to estimate normal flow as opposed to true flow [Alo90, Hag92]. We have experienced the same behaviour of the tensor estimates. Therefore, the certainty of the normal flow case, $c_1$, is used as a weight in the tensor averaging (equation (6.40)). This means that the presumably noisy original estimates, capable of estimating the true velocity, are removed by weighting the original tensor field with the scalar certainty field $c_1$.

The certainty measure, $c_1$, is high only in areas that can be described as simple. High certainty assures that the estimate is not describing a mixture of velocities on a motion boundary. Contrary to $\mathbf{T}$, the tensors $\mathbf{F} = \nabla f \nabla f^T$ (equation (6.38)) are always of rank 1 independent of the spatio-temporal structure (if $\mathbf{F} \neq \mathbf{0}$).

The $c_1$-weighted tensor field, $c_1\mathbf{T}$, contains tensors of rank 1 and describes the normal flow. In figure 6.14 these tensors are marked $\mathbf{T}_1$ (section 6.2.4). Averaging the $c_1$-weighted tensor field, with a spatial localization window, $a_i$, gives

$$\sum_i a_i c_{1i} \mathbf{T}_i = \mathbf{T}' \tag{6.40}$$

In figure 6.14 averaged tensors are marked $\mathbf{T}_2$. The average of two rank 1 tensors is a rank 2 tensor. The two eigenvectors having non-zero eigenvectors span a plane. Figure 6.14 shows that the eigenvalues of the $\mathbf{T}_2$ tensors vary because of summation over different structures. It also shows that in a rigid body motion, the plane the $\mathbf{T}_2$ tensors define is unique. The direction of the normal vector

to this plane defines the velocity. Similar to equation (6.38), this vector is the eigenvector $\mathbf{e}_3$ corresponding to the smallest eigenvalue of $\mathbf{T}'$.

$$\mathbf{v}' = \mathbf{e}_3 = k(v_x, v_y, 1) \tag{6.41}$$

Although the certainty measure removes estimates on motion boundaries, there may still exist multiple motions under the local window $W$. The eigenvalue distribution of the tensor $\mathbf{T}'$ reveals this: if the smallest eigenvalue is not close to zero, more than one motion may be present.
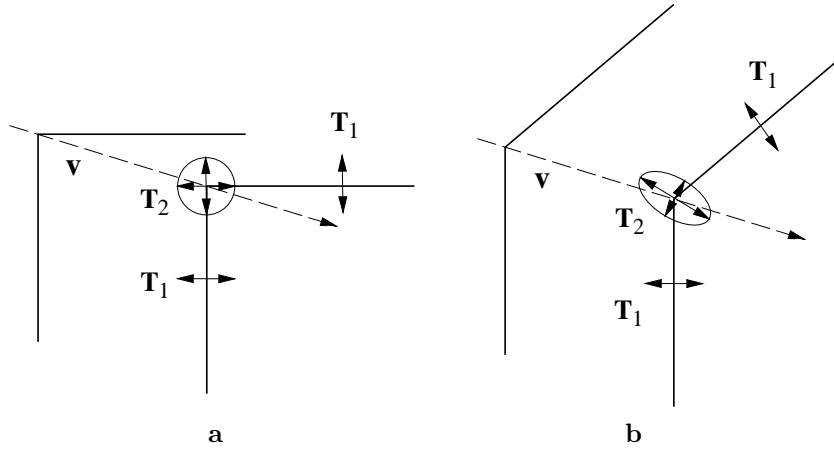


**Figure 6.14    a**) Example of a structure containing two perpendicular lines moving with a velocity **v**. **b**) Two moving lines which are not perpendicular.

Let the coordinate axes of the image plane be denoted $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and let the time axis be denoted $\hat{\mathbf{t}}$. Let $\mathbf{P}_x$ be a projection operator projecting onto the image plane

$$\mathbf{P}_x = \hat{\mathbf{x}}_1 \hat{\mathbf{x}}_1^T + \hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2^T$$

and let $\mathbf{P}_t$ be a projection operator projecting onto the time axis

$$\mathbf{P}_t = \hat{\mathbf{t}}\hat{\mathbf{t}}^T$$

The expression of the true velocity is then given by

$$\mathbf{v} = \frac{\mathbf{P}_x \mathbf{e}_3}{\|\mathbf{P}_t \mathbf{e}_3\|} = \frac{\mathbf{P}_x \mathbf{e}_3}{|\hat{\mathbf{t}}^T \mathbf{e}_3|} \tag{6.42}$$

Note that $\mathbf{e}_3$ does not necessarily have to be normalized.

In all practical cases, noise from various sources is present in the input signal. One way of removing noisy estimates is to remove all estimates below a certain threshold. Thresholding the output field based on the norm of the tensor removed some of the "good" estimates without removing all the noisy ones. Instead of doing this, it has shown useful to add a constant tensor, $\epsilon\mathbf{I}$, to the estimated ones and then remove tensors based only on their shape. The $\epsilon\mathbf{I}$ may be viewed as a *regularization* term. Small tensors will become proportionally "rounder" than larger ones when adding the tensor $\epsilon\mathbf{I}$, and will consequently be removed by a planar shape constraint. In the experiments below, $\epsilon\mathbf{I}$ has been added to the output field using an $\epsilon$ equal to 1 percent of the norm of the largest output tensor (globally).

The algorithm has been evaluated using a synthetic test sequence consisting of two moving boxes, see figure 6.15. The white box is moving to the left with a speed of 0.8 pixels per frame and the black box is moving 45° down and to the left with a speed of 1.6 pixels per frame. The size of the images in the sequence is $64 \times 64$ pixels. The filter $W_i$ used is a Gaussian filter of size $15 \times 15$ pixels. In figure 6.16, the information contained in the unfiltered tensor field is shown. The final result in figure 6.17 indicates that the algorithm produces correct estimates robustly.

We have introduced a method for estimating image flow based on lowpass filtering normal constraints represented by second order tensors. If the neighbourhood is not simple (equation (6.2)), rank 2 tensors containing information about true velocity and certainty measures thereof are produced by the algorithm. If the signal under the filter *is* simple, the algorithm will not break down. Instead, it produces output tensors of rank 1 indicating that a larger window has to be employed if true motion is to be estimated.

It is worth noting that, performing the smoothing in the tensor representation space is a crucial component of the algorithm and is *not* the same as smoothing normal velocities in the image plane. For example, the spatial average of the normal estimates of the white box in figure 6.16 would *not* produce a vector pointing in the direction of the true flow since the box is an elongated square and has more normal estimates pointing upwards than downwards.
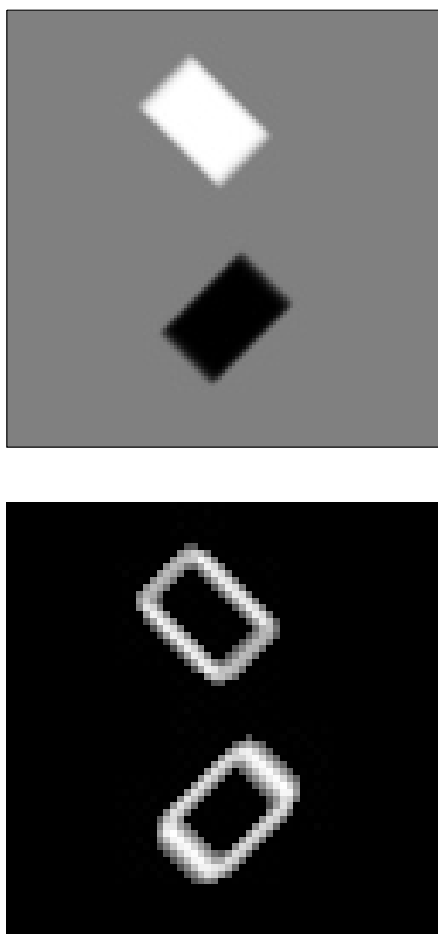
**Figure 6.15** **Top**: One frame from a synthetic test image sequence containing two moving boxes. The white box is moving to the left with a speed of 0.8 pixels per frame and black box is moving 45° down and left with a speed of 1.6 pixels per frame. **Bottom**: The norm of the estimated orientation tensor in the same frame.

$c_1$                                          $c_2$

estimated normal flow                    estimated true flow
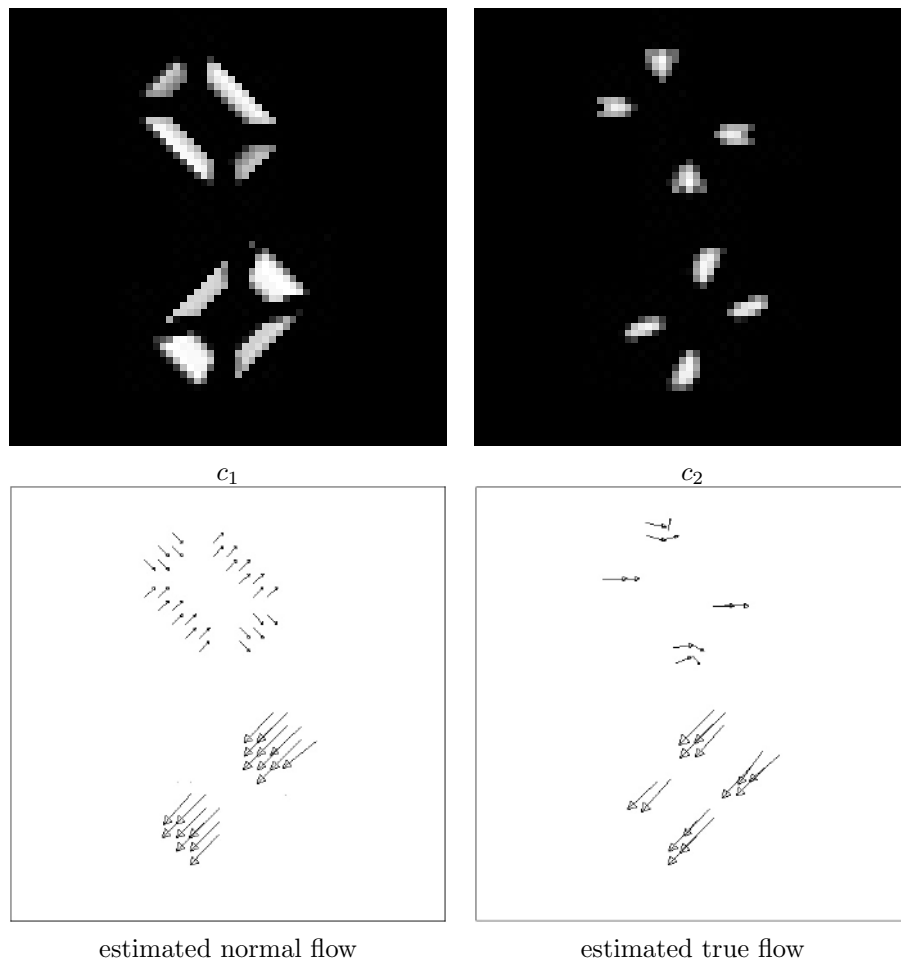
**Figure 6.16**  **Left** Top: Certainty of normal flow ($c_1$), $\alpha = 5$. Bottom: Estimated normal flow. **Right** Top: Certainty of true flow ($c_2$), $\beta = 1.4$. Bottom: Estimated true flow. Note that the estimates are less stable than in the normal flow case.
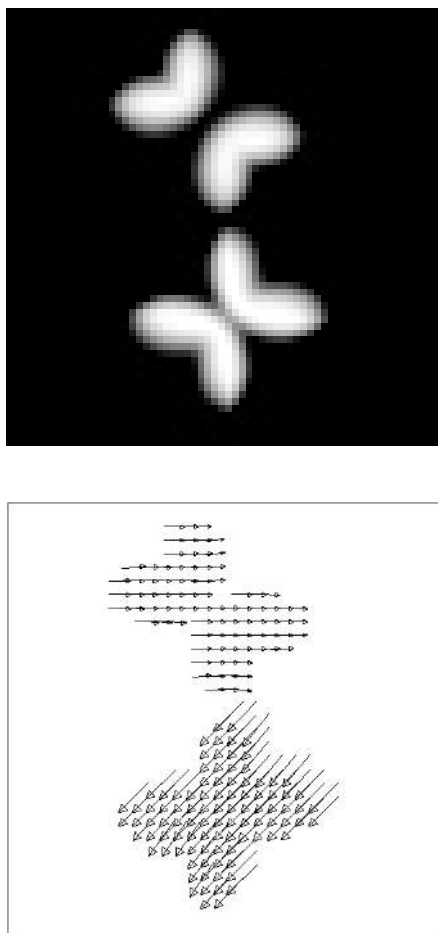
**Figure 6.17** **Top**: Certainty ($c_2$) of true flow after a spatial smoothing of the $c_1\mathbf{T}$ tensor field with a $15 \times 15$ Gaussian smoothing filter, $\beta = 1.4$. **Bottom**: Estimated true flow from the averaged tensor field.

## 6.6   SACCADE COMPENSATION

The experiment was carried out using a simulated environment used to study the performance of an active vision robot. The software package has been developed by Westelius within the ESPRIT Basic Research Action 7108 "Vision as Process" [Wes92].

*Saccades*, i.e. fast camera motions made in order to fixate a new point of interest, will normally introduce strong erroneous responses for a substantial period following the saccade. Such errors can almost completely be eliminated using normalized differential convolution [KWW93]. In the following example, a spatio-temporal Gabor basis function is used.

The *Gabor basis function* has the following form in the spatial domain:

$$\mathbf{b} \ = \ e^{i\omega_0 \mathbf{x}^T \hat{\mathbf{t}}} \tag{6.43}$$

where $\mathbf{x}$ is the spatio-temporal coordinate and $\hat{\mathbf{t}}$ is the normalized temporal direction vector. Inserting this basis function in equation (5.20) gives:

$$\theta \ = \ \frac{\langle a,c\rangle\langle ae^{i\omega_0 \mathbf{x}^T \mathbf{t}}, c\mathbf{f}\rangle - \langle ae^{i\omega_0 \mathbf{x}^T \mathbf{t}}, c\rangle\langle a, c\mathbf{f}\rangle}{\langle a,c\rangle^2 - \langle ae^{i\omega_0 \mathbf{x}^T \mathbf{t}}, c\rangle^2} \tag{6.44}$$

where $\mathbf{f}$ is the input signal.

Note that this equation contains only four different convolutions. The signal certainty, $c$, is set to zero during the saccade. Otherwise it is set to one. This is not the same thing as shutting off the camera during the saccade, since that would produce strong edge effects.

In figure 6.18 a sequence from a simulated environment has been used for showing the possibility of compensating for unwanted operator responses because of saccades. The scenario is an object continuously moving around on the floor. The initial viewpoint of the robot is on the wall at the rightmost painting. Then, the robot makes a saccade, shifting its attention to the leftmost painting. The middle sequence contains the output from standard convolution using a quadrature filter. The spatio-temporal filter responses caused by the saccade clutter the whole image. To the right in figure 6.18 the output from normalized differential convolution is shown. The unwanted responses caused by the saccade have been significantly reduced.
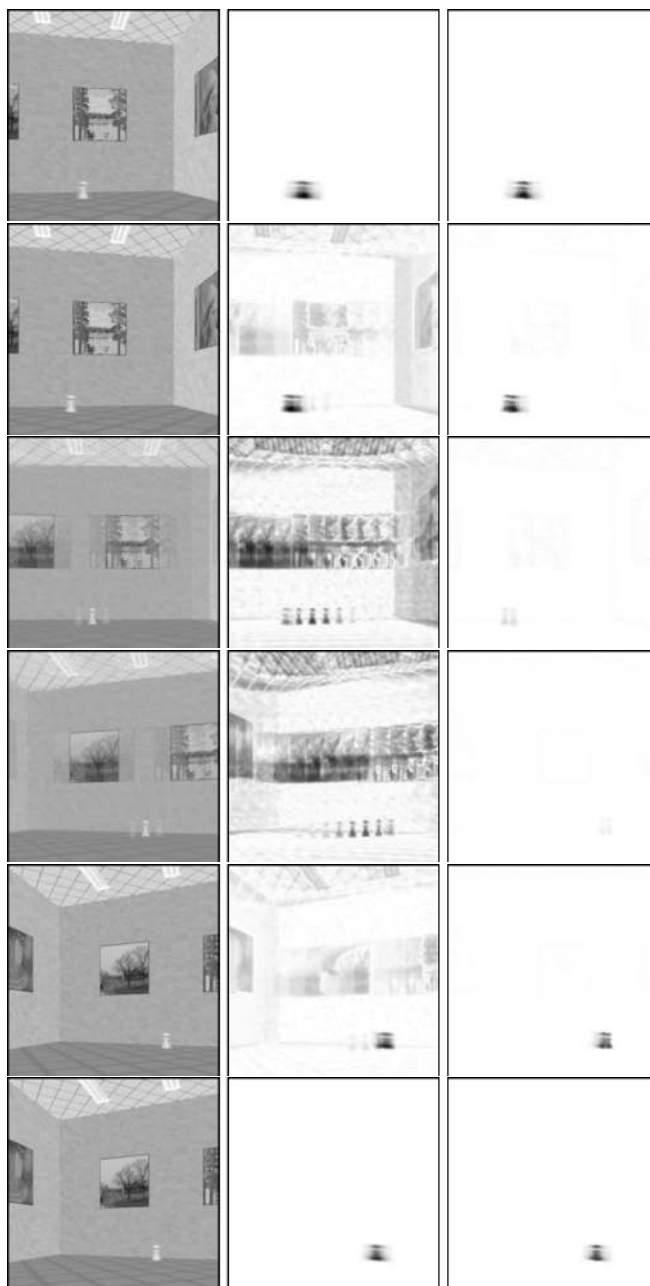
**Figure 6.18** **Left**: Original sequence, a simulation of an object moving around on the floor. Every fourth frame is shown here. **Middle**: Filter magnitude output from the used Gabor filter, $a\mathbf{b}$. The applicability function $a$ has the size $9 \times 9 \times 9$. **Right**: Output from Normalized differential convolution. The strong erroneous responses in the middle sequence are almost eliminated.

# 7

# LINE AND PLANE EXTRACTION USING TENSORS

## 7.1 INTRODUCTION

This chapter focuses on various aspects of representation and grouping of information. We begin with an investigation of the uniformity of some standard parameter spaces (also termed parameter mappings). The analysis shows that, to avoid discontinuities, great care should be taken when choosing the parameter space for a particular problem.

It is shown that the local strucure tensor introduced in chapter 6 can be decomposed into projection operators which can be used for grouping collinear/coplanar estimates [Wes91]. Based on these results, a new parameter mapping well suited for line extraction, the Möbius strip parameterization [WK92b, Wes91], is defined. The method has similarities to the *Hough Transform* [IK88].

## 7.2 PARAMETER MAPPINGS FOR LINE SEGMENTATION

The reason for using a *parameter mapping* is often to convert a difficult global detection problem in image space into a local one. Spatially extended patterns are transformed so that they produce spatially concentrated features in a space of parameter values. In the case of line segmentation, the idea is to transform the original image into a new domain so that collinear subsets, i.e. global lines, fall into clusters. The topology of the mapping must reflect closeness between wanted features, in this case features describing properties of a line. The metric describing closeness should also be uniform throughout the space with respect to the features. If the metric and topology do not meet these requirements, significant bias and ambiguities will be introduced into any subsequent classification

process. These aspects are especially important when working with dynamic scenes. If, for example, the position in the parameter space for a moving line does not change smoothly, it may be hard or impossible to keep track of them. In this section some problems with standard mappings for line segmentation will be illuminated. We start with discussing some intuitively natural mappings for line segmentation.

The Hough transform, HT, was introduced by P. V. C. Hough in 1962 as a method for detecting complex patterns [Hou62]. It has found considerable application due to its robustness when using noisy or incomplete data. A comprehensive review of the Hough transform covering the years 1962-1988 is found in [IK88].

## The km-parameter mapping

The key ideas of the HT can be illuminated by considering identifying sets of collinear points in an image. Points $(x, y)$ lying on a straight line satisfy the well-known relation

$$y = kx + m$$

where $k$ is the slope and $m$ is the intersection of the line with the $y$-axis. In the $x$-$y$ space, $(x, y)$ are variables and $(k, m)$ parameters. In the $k$-$m$ space it is vice versa. A mutual constraint between image points $(x, y)$ and parameter points $(k, m)$ is defined by

$$y - kx - m = 0 \tag{7.1}$$

The HT is a mapping from one to many. Each image point votes for *all* parameter combinations that could have produced it. This means that each image point $(x, y)$ votes for a complete line in the parameter space $(k, m)$. The histogram produced from all the points in the image contains a peak, $(k_{peak}, m_{peak})$, corresponding to the parameters defining the original line in the image, see figure 7.2 (left).

## The $\rho\varphi$-parameter mapping

Severe problems with standard Hough parameterization are that the space is unbounded and will contain singularities for large slopes, $k$. The difficulties of unlimited ranges of the values can be solved by using two plots, the second corresponding to interchanging the axes. This is of course not a satisfactory solution. Duda and Hart [DH72] suggested that straight lines might be most

usefully parameterized by the length, $\rho$, and orientation $\varphi$, of the normal vector to the line from the origin, the *normal parameterization*, see figure 7.1.

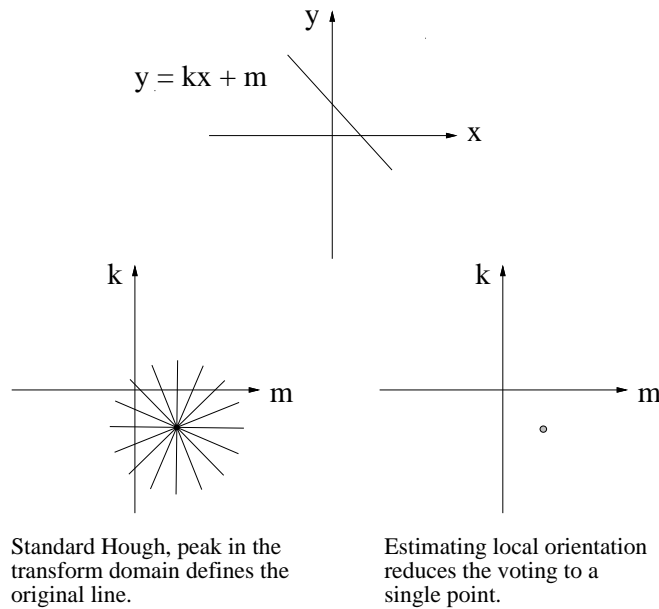$$\rho - x\cos(\varphi) - y\sin(\varphi) = 0 \qquad (7.2)$$

This mapping has the advantage of having no singularities.
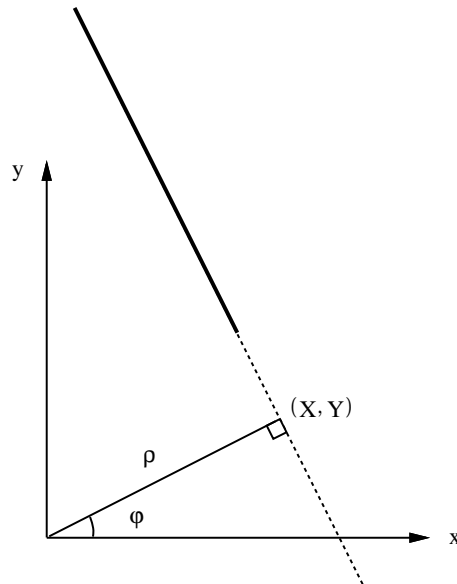
**Figure 7.1**   The *normal parameterization*, $(\rho, \varphi)$, of a line. $\rho$ is the magnitude of displacement vector from the origin and $\varphi$ is its argument.

Measuring local orientation provides additional information about the slope of the line or the angle $\varphi$ when using the normal parameterization. This reduces the standard HT to a one-to-one mapping. With one-to-one, we do not mean that the mapping is invertible, but that there is only one point in the parameter space that defines the parameters that could have produced it, see figure 7.2 (right).

Duda and Hart discussed this briefly in [DH73]. They suggested that this mapping could be useful when fitting lines to a collection of short line segments. Dudani and Luk [DL78] used this technique for grouping measured edge elements. Princen, Illingworth and Kittler performed line extraction using a pyramid structure [PIK90]. At the lowest level they used the ordinary $\rho\varphi$-HT on subimages for estimating small line segments. In the preceding levels they used the additional local orientation information for grouping the segments.

Standard Hough, peak in the
transform domain defines the
original line.

Estimating local orientation
reduces the voting to a
single point.

**Figure 7.2**   Measuring local orientation provides additional information about
the slope of the line or the angle $\varphi$ when using the normal parameterization.

Unfortunately, however, the normal parameterization has problems when $\rho$ is
small. Some lines may generate more than one cluster. Consider for example
a line going through the origin in a $xy$-coordinate system. When mapping the
coordinates according to the normal parameterization, *two* clusters will be pro-
duced separated in the $\varphi$-dimension by $\pi$, see figure 7.3. Note that this will
happen even if the orientation estimates are perfectly correct. A line will always
have at least an infinitesimal thickness and is therefore likely to be projected on
*both* sides of the origin. A final point to note is that a translation of the origin
outside the image plane will not remove this topological problem. It will only be
transferred to other lines.

Granlund introduced a double angle notation [Gra78] in order to achieve a suit-
able continuous representation for local orientation. However, using this double
angle notation for *global* lines, removes the ability to distinguish between lines
with the same orientation and distance, $\rho$, on opposite sides of the origin. How-
ever, the problem near $\rho = 0$ is removed. In figure 7.3 the two horizontal lines
(marked a and c), located at the same distance $\rho$ from the origin, are in the
double angle normal parameterization mixed into one cluster.
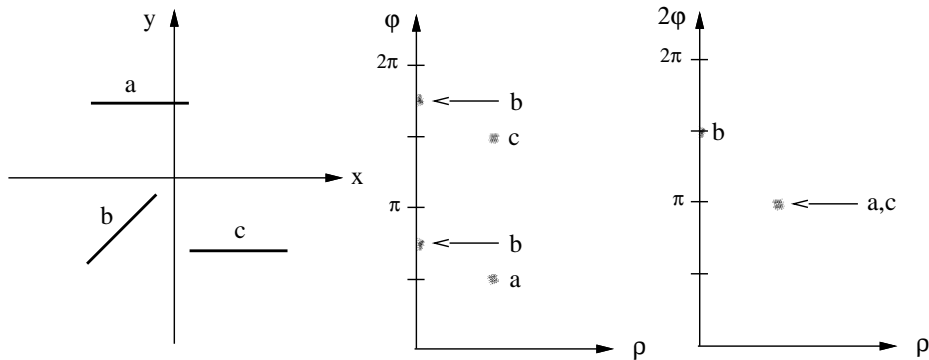
**Figure 7.3** A test image containing three lines and its transformation to the $\rho\varphi$-domain, the normal parameterization of a line. The cluster from the line at 45° is divided into two parts. This mapping has topological problems near $\rho = 0$. The $\rho 2\varphi$-domain, however, folds the space so the topology is good near $\rho = 0$, but unfortunately it is now bad elsewhere. The two horizontal lines, marked a and c, have in this parameter space been mixed in the same cluster.

It seems that we need a double angle representation around for small values of $\rho$, and a *single angle representation* elsewhere. This raises a fundamental dilemma: is it possible to achieve a mapping that fulfills both the single angle and the double angle requirements simultaneously?

We have been concerned with the problem of the normal parameterization spreading the coordinates around the origin unsatisfactorily although they are located very close in the Cartesian representation. Why do we not express the displacement vector, i.e. the normal vector to the line from the origin, in *Cartesian* coordinates, $(X, Y)$, since the topology is satisfactory? This parameterization is defined by

$$X = x\cos^2(\varphi) + y\cos(\varphi)\sin(\varphi) \tag{7.4}$$

$$Y = y\sin^2(\varphi) + x\cos(\varphi)\sin(\varphi) \tag{7.5}$$

where $\varphi$ as before, is the argument of the normal vector (the same as the displacement vector of the line).

Davis uses the $\rho\varphi$-parameterization in this way by storing the information in a Cartesian array [Dav86]. This gives the $(X, Y)$ parameterization. There are two reasons for not using this parameterization. First, the spatial resolution is very poor near the origin. Secondly, and worse, all lines having $\rho$ equal to 0 will be

mapped to the same cluster. The first problem, the poor resolution near the origin, can at least be solved by mapping the $XY$-plane onto a logarithmic cone. That would stretch the $XY$-plane so the points close to the origin get more space. However, the second problem still remains.

## 7.3   THE ORIENTATION TENSOR

In this section the orientation information contained in the orientation tensor presented in chapter 6 is related to the previous discussion concerning parameter mappings for lines. It is shown that this tensor can be decomposed into projection operators which can be used for *grouping*.

### 7.3.1   Decomposing the orientation tensor

As mentioned, the orientation tensor introduced in chapter 6 can be used for grouping. Since this tensor is symmetric, equation (2.63) shows that the tensor can be decomposed into a weighed sum of projection operators:

$$\mathbf{T} \;=\; \sum_{i=1}^{n} \lambda_i \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \tag{7.5}$$

where $\hat{\mathbf{e}}_i$ is a normalized eigenvector of $\mathbf{T}$.

If all the $\lambda_i$ are distinct, the decomposition of $\mathbf{T}$ is unique and the projection operators are the outer product of the eigenvectors:

$$\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T$$

If two or more eigenvalues are equal, there is an option in defining the eigenvectors in that subspace, and hence there is an option in the decomposition in eigenvectors. The subspace they span is, however, distinct. Say for example that $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ share the same eigenvalue and define an operator projecting onto this subspace:

$$\mathbf{T}_2 \;=\; \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T \tag{7.6}$$

Thus, decomposing the tensor in such operators gives a unique set of coefficients.

An interesting choice is to decompose into projection operators having different geometrical meaning. We define the following projection operators:

$$\mathbf{T}_1 = \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T \tag{7.8}$$

$$\mathbf{T}_2 = \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T \tag{7.9}$$

$$\mathbf{T}_i = \hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T + \ldots\ldots + \hat{\mathbf{e}}_i\hat{\mathbf{e}}_i^T \tag{7.10}$$

The first projection operator corresponds to a one-dimensional subspace, a line. The second corresponds to a two-dimensional subspace, a plane, etc. Expressing the orientation tensor in this new basis gives:

$$\mathbf{T} = \lambda_n\mathbf{T}_n + \sum_{i=1}^{n-1} (\lambda_i - \lambda_{i+1})\mathbf{T}_i \tag{7.10}$$

The coefficients are trivially derived by comparing this expression with the one produced be the spectral decomposition theorem in equation (2.63).

Note that the above tensor decomposition is the same as the one used in chapter 6 for classifying higher order neighbourhoods.

## 7.3.2   Covariant and contravariant tensors

In chapter 2 it was stated that tensors are based on two forms, namely covariant and/or contravariant tensors. Since the orientation tensor has close relations to the outer product of the gradient, it can be argued that this tensor is a covariant tensor of order two [Wes91]. Estimating the orientation in a simple neighbourhood (equation (6.2)) produces a tensor of rank 1 (chapter 6). In our context this corresponds to the projection operator $\mathbf{T}_1$ in equation (7.7). Letting this tensor act on its spatial position vector gives:

$$\boldsymbol{\rho} = \mathbf{T}_1(\mathbf{x}) \tag{7.11}$$

or written in component form (the index 1 is omitted):

$$\rho_j = T_{ij}x^i \tag{7.12}$$

where $T_{ij}$ denotes the components of the orientation tensor, $x^i$ denotes the components of the vector defining the position in image space and $\rho_j$ denotes the components of the produced tensor; a *covector*.

Thus, the projection operator $\mathbf{T}_1$ projects the spatial position (a contravariant vector) of the estimate (a covariant tensor of order two) to a covariant vector. The fact that the output is a covector is indicated by the *subscript* (chapter 2).

### 7.3.3 Lines in two dimensions

In two dimensions, equation (7.10) gives the decomposition of the orientation tensor into projection operators

$$\mathbf{T} = (\lambda_1 - \lambda_2)\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \lambda_2(\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^T) = (\lambda_1 - \lambda_2)\mathbf{T}_1 + \lambda_2\mathbf{T}_2 \qquad (7.13)$$

As mentioned, the operation in equation (7.11) produces a *covector*, $\boldsymbol{\rho}$.

It turns out that the orientation of this vector is *perpendicular* to the orientation of line which is to be estimated (figure 7.4). Thus, the vector is pointing at the line coordinate closest to the origin. To facilitate drawing of the orientation
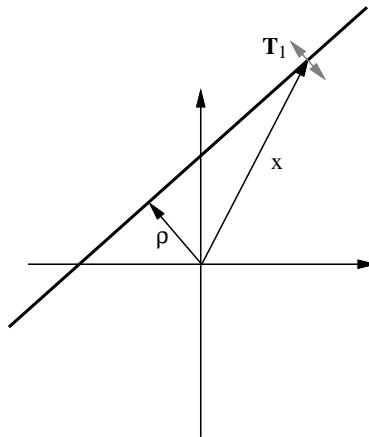


**Figure 7.4**   Grouping of information

tensor we have in this figure used the "double arrow" notation from figure 6.2. The double arrow corresponds to the direction of the largest eigenvector. The reason for having two directions is that if $\hat{\mathbf{e}}_1$ is the eigenvector of $\mathbf{T}_1$, so is $-\hat{\mathbf{e}}_1$. The double arrow indicates these two directions. A second order tensor of rank 1 can therefore be seen as an "unsigned" vector.

The grouping that the tensor operation performs can be illuminated by the following derivation. The vector **x** in figure 7.4 can be divided into one vector perpendicular to the line of collinearity and one parallel to it,

$$x^i = \rho^i + \alpha \rho^j_\perp \tag{7.14}$$

where $\alpha$ defines the position along the line. Using this expression in the operation from equation (7.12):

$$T_{ij} x^j = T_{ij}(\rho^j + \alpha \rho^j_\perp) = T_{ij}\rho^j + \alpha T_{ij}\rho^j_\perp \tag{7.15}$$

Recalling the definition of the ideal orientation tensor as the tensor product between the covector perpendicular to the line and itself,

$$T_{ij} = \rho_i \rho_j \tag{7.16}$$

gives:

$$T_{ij}\rho^j + \alpha T_{ij}\rho^j_\perp = \rho_i \rho_j \rho^j + \alpha \rho_i \rho_j \rho^j_\perp = \rho_i + 0 = \rho_i \tag{7.17}$$

which shows that all collinear tensors having equal orientation are projected onto the same covector.

The fact that both the strucure tensor and $\boldsymbol{\rho}$ are covariant can also be shown by studying their transformation properties geometrically. Suppose we change our coordinate system making the line in figure 7.4 look like the one in figure 7.5. In this figure the new transformed coordinate system is shown. If **T** and $\boldsymbol{\rho}$ were contravariant tensors, they would be transformed as in the left figure. It can be seen that they transform as the vector **x**, i.e. they become more vertical. Thus, they turn in the "wrong" direction. To the right in this figure, it is shown how covariant tensors would be transformed.

## 7.3.4 Lines and planes in three dimensions

In three dimensions, equation (7.10) shows that the decomposition of the orientation tensor into projection operators has the following form

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_1 + (\lambda_2 - \lambda_3)\mathbf{T}_2 + \lambda_3 \mathbf{T}_3 \tag{7.18}$$
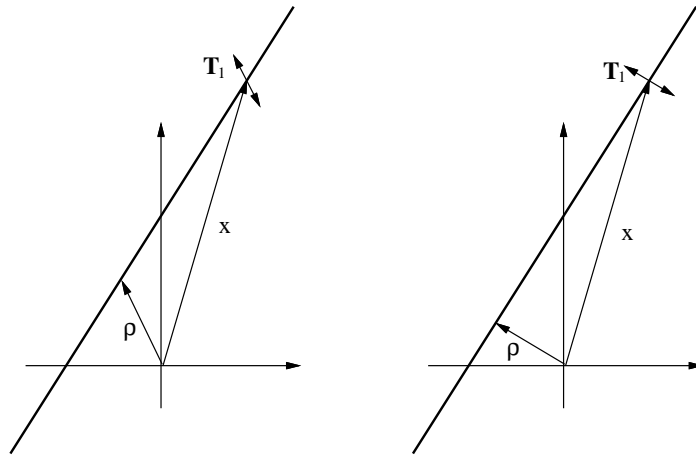
**Figure 7.5**   In this figure the new transformed coordinate system is shown. If
**T** and $\boldsymbol{\rho}$ were contravariant tensors, they would be transformed as in the left
figure. It can be seen that they transform as the vector **x**, i.e. they become more
vertical. To the right it is shown how covariant tensors would be transformed.

Simple and higher rank neighbourhoods were discussed in chapter 6 when in-
terpreting the orientation tensor in three dimensions. As mentioned there, the
rank of the tensor reflects the complexity of the neighbourhood. The plane case
corresponds to what was referred to as a simple neighbourhood. The line case
corresponds to a neighbourhood that is constant on *lines*. The orientation of the
lines is given by the eigenvector corresponding to the smallest eigenvalue, $\hat{\mathbf{e}}_3$.

The projection operator $\mathbf{T}_1$ groups coplanar points and $\mathbf{T}_2$ groups collinear
points. This grouping is visualized in figure 7.6, where the set of coplanar orien-
tation tensors will be projected onto $\boldsymbol{\rho}_1$ and the set of collinear tensors will be
projected onto $\boldsymbol{\rho}_2$. In both cases the elements are projected to the point on the
plane/line *closest* to the origin.

## 7.3.5   Parameter spaces defined by projection operators

*Two dimensions*

In two dimensions, the parameter mapping is two-dimensional: the components
of the produced covector. However, lines passing through the origin of the image
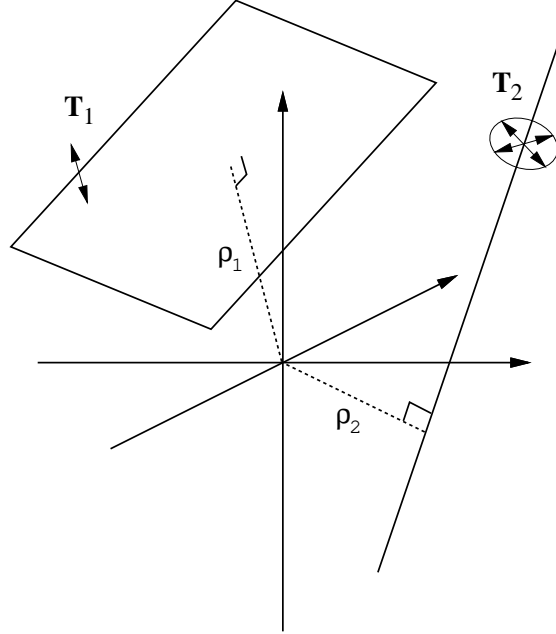
**Figure 7.6** Grouping of information in three dimensions.

coordinate system ($\mathbf{x} = 0$) are projected to the same cluster. One way of preventing the estimates from different lines from mixing is to incorporate the operator, $\mathbf{T}_1 = \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T$ in the mapping as well. The operator contains three different components and the produced covector contains two. This gives us a five-dimensional grouping space:

$$(\rho_x, \rho_y, e_{1_x}^2, e_{1_y}^2, e_{1_x} e_{1_y}) \tag{7.19}$$

## *Three dimensions*

In three dimensions, we similarly get two parameter mappings. One nine-dimensional grouping space for simple neighbourhoods:

$$(\rho_x, \rho_y, \rho_z, e_{1_x}^2, e_{1_y}^2, e_{1_z}^2, e_{1_x} e_{1_y}, e_{1_x} e_{1_z}, e_{1_y} e_{1_z}) \tag{7.20}$$

and another nine-dimensional grouping space for rank 2 neigbourhoods:

$$(\rho_x, \rho_y, \rho_z, e_{1_x}^2 + e_{2_x}^2, e_{1_y}^2 + e_{2_y}^2, e_{1_z}^2 + e_{2_z}^2, \tag{7.21}$$

$$e_{1_x} e_{1_y} + e_{2_x} e_{2_y}, e_{1_x} e_{1_z} + e_{2_x} e_{2_z}, e_{1_y} e_{1_z} + e_{2_y} e_{2_z})$$

Note, however, that the parameters in these mappings are dependent. This means that only a subspace of the parameter space contains data. Unfortunately these subspaces are folded in a complex manner.

This concludes the general discussion about projection operators and grouping data. In the next section we will focus on parameter mappings for two-dimensional lines and present a parameter mapping that unfolds the five-dimensional mapping presented in this section.

## 7.4  THE MÖBIUS STRIP PARAMETERIZATION

In this section we present a new parameter space and discuss its advantages with respect to the arguments of the previous section. The topology of the mapping introduces its name, the *"Möbius strip" parameterization*. This mapping has topological advantages over previously proposed mappings.

The Möbius strip mapping is based on a transformation to a four-dimensional space by taking the normal parameterization in figure 7.1, expressed in Cartesian coordinates $(X, Y)$ and adding a "double angle" dimension, (consider the $Z$-axis in a $XYZ$-coordinate system). The problem with the Cartesian normal parameterization is, as mentioned, that all clusters from lines going through the origin mix into one cluster. The additional dimension, $\phi = 2\varphi$, separates the clusters on the origin and close to the origin if the clusters originate from lines with different orientation. Moreover, the wrap-around requirement for $\phi$ is ensured by introducing a fourth dimension, $R$.

### The four-dimensional mapping

$$\begin{cases} X & = x\cos^2(\varphi) + y\cos(\varphi)\sin(\varphi) \\ Y & = y\sin^2(\varphi) + x\cos(\varphi)\sin(\varphi) \\ \phi & = 2\varphi \\ R & = R_0 \in \phantom{}^+ \end{cases} \tag{7.22}$$

The two first parameters, $X$ and $Y$, represent the normal vector in figure 7.1 expressed in Cartesian coordinates. The two following parameters, $\phi$ and $R$, define a circle with radius $R_0$ in the $R\phi$-subspace. Any $R_0 > 0$ is suitable. This gives a $XY\phi$-system with wrap-around in the $\phi$-dimension. This four-dimensional mapping is a subspace of the five-dimensional space presented in section 7.3.5. The relation between the mappings is: $X = \rho_x$, $Y = \rho_y$ and the

relation between the tensor components and the double angle representation is described in section 6.2.3.

In the mapping above, the parameters are dependent. As the argument of the vector in the $XY$-plane is $\varphi$ and the fourth dimension is constant, it follows that for a specific $(X, Y)$ all the parameters are given. Hence, the degrees of freedom are limited to two, the dimension of the $XY$-plane. Thus, all the mapped image points lie on a two-dimensional surface in the four-dimensional parameter space, see figure 7.7.
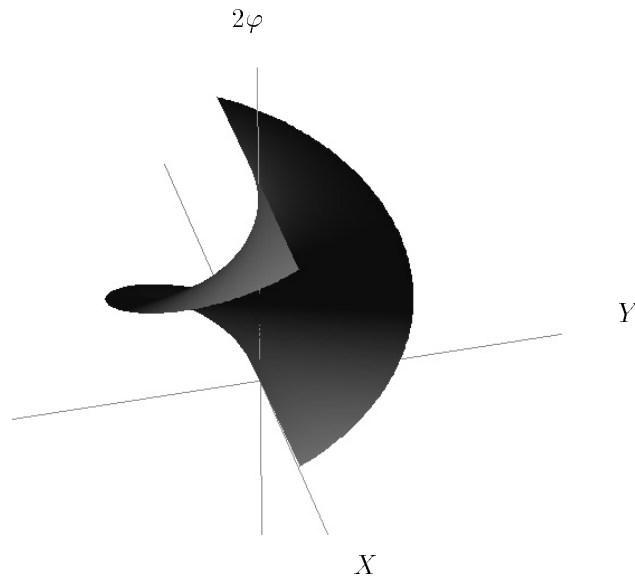


**Figure 7.7** The 2D subspace

## The two-dimensional surface

The regular form of the two-dimensional surface makes it possible to find a two-parameter form for the desired mapping. Let us consider a $\eta\phi$-plane corresponding to the flattened surface in figure 7.7. Let

$$\rho^2 = X^2 + Y^2 = (x\cos^2(\varphi) + y\cos(\varphi)\sin(\varphi))^2 + (y\sin^2(\varphi) + x\cos(\varphi)\sin(\varphi))^2 \quad (7.23)$$

$$\Leftrightarrow \qquad \rho = x\cos(\varphi) + y\sin(\varphi) \quad (7.24)$$

Then the $(\eta, \phi)$ mapping can be expressed as

$$\eta = \begin{cases} \rho & 0 \le \varphi < \pi \\ -\rho & \pi \le \varphi < 2\pi \end{cases} \tag{7.25}$$

$$\phi = 2\varphi \tag{7.26}$$

$\eta$ is the variable "across" the strip with 0 value meaning the position in the middle of the strip, i.e. *on* the $2\varphi$ axis. The wrap-around in the $\phi$ dimension makes it easy to interpret the surface as a Möbius strip.

Finally, using the same test image as before (figure 7.3), we see that we can distinguish between the two lines on opposite sides of the origin while not dividing the cluster corresponding to the line going through the origin (figure 7.8).
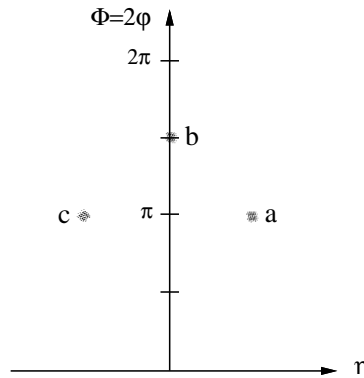


**Figure 7.8**   $(\eta, \phi)$ he The Möbius strip mapping with parameters $(\eta, \phi)$. We can see that we can distinguish between the two lines, a and c, at opposite sides of the origin at the same time as the cluster corresponding to the line going through the origin is not divided.

The name of the mapping, as mentioned above, reflects the topology of the $\eta\phi$-parameter surface, its *twisted* wrap-around in the $\phi$-dimension. The proposed mapping has the following properties:

1. The parameter space is bounded and has no singularities such as the standard HT parameterization for large slopes.

2. The metric reflects the underlying geometry of the features. The mapping does not share the topological problem of the $\rho\varphi$-mapping near $\rho = 0$. In

[WK92b] it was shown that any line passing through the origin produces *two* clusters, separated by $\pi$ in the $\varphi$-dimension.

Note that these properties have been achieved *without* increasing the dimension of the parameter space compared to for example the *normal parameterization* in figure 7.1.

## 7.5 LOCAL ESTIMATES WITH GLOBAL SUPPORT

In this section a simple segmentation example is shown. The method is based on segmenting estimates having local and global support simultaneously. The magnitude of the local orientation estimates are shown in figure 7.9 (top right). These estimates are projected and accumulated in a Möbius histogram (figure 7.9 top left).

The image is then transformed a second time. This time no new histogram is produced. The coordinate obtained via the mapping is only used as a pointer, checking the energy concentration in the earlier produced histogram. The model support for a global line, the energy at the location of the pointer in the histogram is combined with the certainty of the local orientation estimate is combined. This is shown in figure 7.9 (middle and bottom)
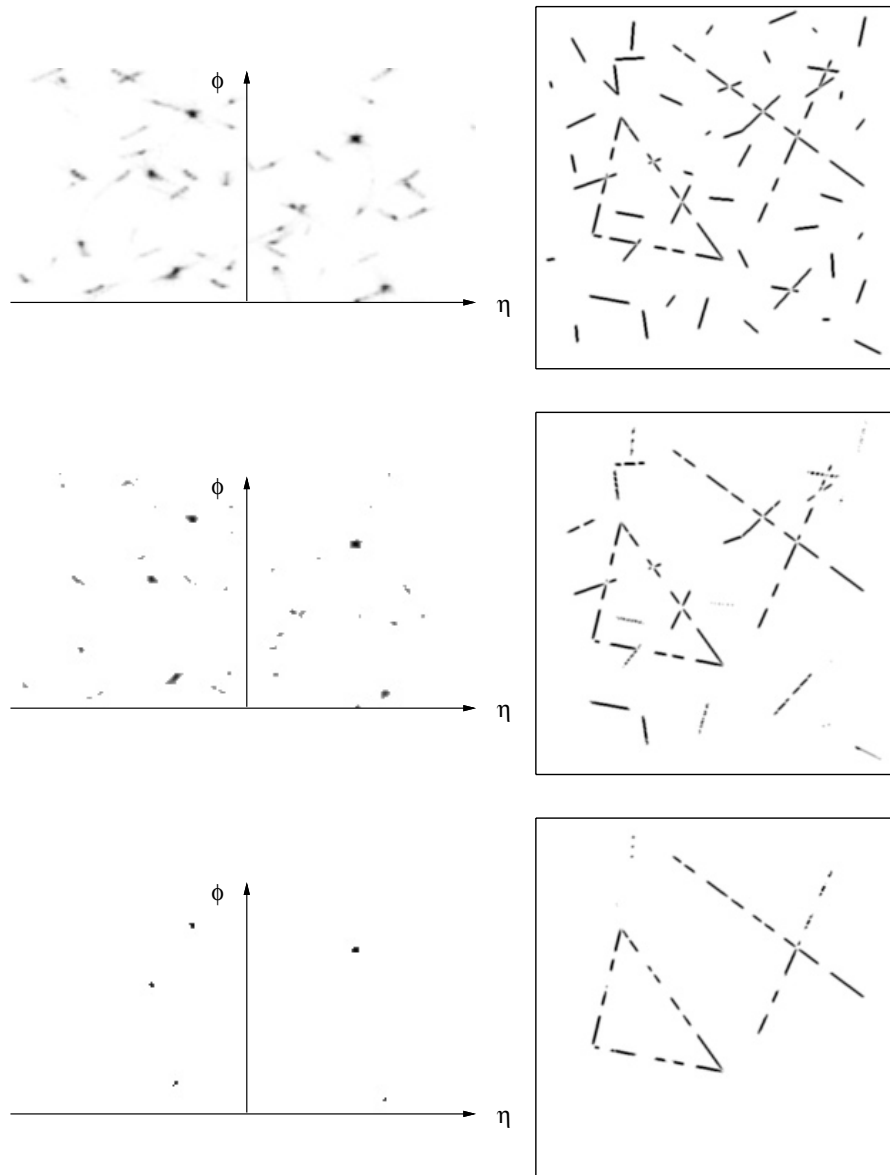
**Figure 7.9**  By demanding simultaneous support in the two domains, only local estimates having enough of neighbours along the global structure stating the same orientation survives. The required level of energy support in the histogram domain is denoted $\alpha$. The histogram maxvalue is scaled to 1. **Top**: $\alpha = 0$. **Middle**: $\alpha = 0.3$. **Bottom**: $\alpha = 0.7$.

# 8

# LOCAL FREQUENCY AND BANDWIDTH ESTIMATION

This chapter describes a technique for estimation of local signal frequency and bandwidth. Local frequency is an important concept useful for local structure analysis as well as for determining the appropriate range of scales for subsequent processing. The method is based on combining local estimates of instantaneous frequency over a large number of scales. The filters used are a set of lognormal quadrature filters. The bandwidth is used to produce a measure of certainty for the estimated frequency. The algorithm is applicable to multidimensional data and examples of the performance of the method are demonstrated for one-dimensional and two-dimensional signals. The material is based on [KWG94].

## 8.1 INTRODUCTION

The concept of frequency is a mathematically well-defined entity. Any stationary signal can be represented as a weighted sum of sine and cosine functions having particular amplitudes, phases and frequencies. For non-stationary signals, however, this is not the case, and a description in term of sines and cosines becomes meaningless. This fact has, since most real life signals are non-stationary, led to a number of attempts to find methods which would allow non-stationary signals to be analysed in a frequency-like manner.

### 8.1.1 Instantaneous frequency

One such attempt was the introduction of the notion of *instantaneous frequency*. Carson and Fry [CF37] and van der Pol [vdP46] defined instantaneous frequency and applied it to frequency modulated signals. The concept of instantaneous frequency is based on the so-called *analytic signal* which in turn is defined via

the Hilbert transform. We will shortly review these concepts, before defining instantaneous frequency.

## The Hilbert transform

The *Hilbert transform* corresponding to a signal $f(x)$ is obtained by convolving the signal with the function $\frac{-1}{\pi x}$,

$$f_{\mathcal{H}i}(x) = f(x) * \frac{-1}{\pi x} \tag{8.1}$$

Let $F_{\mathcal{H}i}(u)$ denote the Fourier transform corresponding to $f_{\mathcal{H}i}(x)$. Since convolution in the spatial domain corresponds to multiplication in the Fourier domain and the Fourier transform of $\frac{-1}{\pi x}$ is $i\,\text{sign}(u)$, it must be the case that

$$F_{\mathcal{H}i}(u) = F(u) \cdot i\,\text{sign}(u) \tag{8.2}$$

where $u$ is the frequency variable and $\text{sign}(u)$ equals one for $u > 0$ and minus one for $u < 0$. Hence, the Fourier transform of $f_{\mathcal{H}i}$ is obtained by multiplying $F$ by the imaginary unit $i$ and then change sign of the result for negative frequencies. Another way of describing this process is to turn the argument of the frequency components an angle $\frac{\pi}{2}$ in the positive direction for positive frequencies and in the negative direction for negative frequencies. From this follows immediately that two Hilbert transforms in succession will simply change the sign of a function.

## The analytic signal

Having defined the Hilbert transform, we can define the *analytic signal*, $f_A$, corresponding to a real signal $f$.

**Definition 7 (The analytic signal)** *The analytic signal is a complex signal and is uniquely defined as:*

$$f_A = f - i f_{\mathcal{H}i} \tag{8.3}$$

*where $f_{\mathcal{H}i}$ denotes the Hilbert transform of $f$.*                               $\square$

Since the Hilbert transform is well-defined only for one-dimensional signals, the analytic signal is well-defined only for these signals, too. The Hilbert transform can be used in higher dimensional spaces if a direction in this space is specified.

**Example 10** *The analytic signal corresponding to* $\cos x$ *is* $e^{iu}$.    □

**Example 11** *The analytic signal corresponding to* $\sin x$ *is* $-ie^{iu}$.    □

From the definition of the Hilbert transform it is clear that we can rewrite

$$f_A = f * [\delta(x) + \frac{i}{\pi x}], \tag{8.4}$$

which in the Fourier domain corresponds to

$$F_A = F \cdot [1 + \text{sign}(u)] = 2\, F \cdot \text{step}(u). \tag{8.5}$$

where $\text{step}(u)$ is the step function: $\text{step}(u) = 1$ for $u > 0$ and zero for $u < 0$. Hence, the analytic signal corresponding to $f$ is obtained by suppressing all its negative frequencies and multiplying by two. It should be noted that this implies that an analytic function can not be real and non-zero.

For a real signal, the corresponding analytic signal is complex, with the real part being the original signal itself and the imaginary part being its Hilbert transform. This is illustrated in figure 8.1.

## *The instantaneous frequency*

The instantaneous frequency is commonly defined as the rate of change in argument of the analytic function. In the following, the argument of $f_A$ is referred to as the *phase* of $f$.

$$\arg[f_A(x)] = \omega_0 + \omega x \ + \ \text{higher order terms}. \tag{8.6}$$

If $f$ is a single frequency signal, the phase of $f$ is a linear function of $x$ and $\arg[f_A(x)] = \omega_0 + \omega x$. Hence, one way of obtaining the angular frequency is to differentiate the phase with respect to $x$.

**Definition 8 (Instantaneous frequency)** *For any real function such that its phase has a well-defined derivative with respect to* $x$, *we define the instantaneous frequency of* $f(x)$ *as*

$$\omega = \frac{d}{dx} \arg[f_A(x)]. \tag{8.7}$$

□
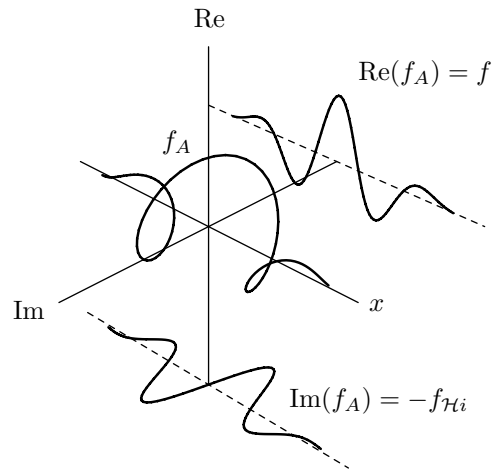
**Figure 8.1** The analytic signal $f_A$ corresponding to a real function $f$. The Hilbert transform $f_{\mathcal{H}i}$ of $f$ is shown with reversed sign in the imaginary domain.

In practice we are interested in filtered versions of the analytical signal. Note now, according to the definition of the analytic signal $f_A$, that all negative frequencies of $f$ are suppressed. Furthermore, we know that convolution in the spatial domain corresponds to multiplication in the Fourier domain. Hence, employing filters with zero response for negative frequencies, on the original signal, $f$, is equivalent to act on the analytical signal $f_A$. Such filters are called quadrature filters and can be thought of as acting on the analytic signal $f_A$, even though they in practice act on $f$.

Local behaviour and spatial frequency are since long recognized as important properties for signal processing. A historical review of the concept of instantaneous frequency, and methods for estimation instantaneous phase and frequency, can be found in [Boa92a, Boa92b]. It is noteworthy that - despite its name -, instantaneous frequency is a global entity, in that local alterations of the signal will effect the instantaneous frequency everywhere due to the infinite kernel associated with the Hilbert transform.

## 8.1.2    Local Fourier transforms

As mentioned in chapter 2, Gabor [Gab46] proposed a combined representation of time and frequency. He expanded the signal in modulated Gaussian basis functions

$$f(x) \ = \ \sum_i \alpha_i \ g_{u_i,\sigma}(x) \tag{8.8}$$

where $u$ is the frequency variable and

$$g_{u_i,\sigma}(x) \ = \ e^{iux} e^{\frac{x^2}{\sigma^2}} \tag{8.9}$$

where the coefficients $\alpha_i$ are obtained by convolving with the *dual* basis functions.

A similar expansion was demonstrated in chapter 4 where the the normalized Fourier transform was introduced. It was shown how a signal is expanded in infinite sinusoids and cosinusoids. The localization was achieved by using an applicability function as a certainty window instead of changing the basis function or the signal. In this case the dual basis is calculated implicitly via the metric tensor. This dual basis is not exactly the same as the one derived from the auxiliary function in figure 3.6 which is derived from a complete set of Gabor function which is infinit.

In many practical situations, however, orthogonality of the basis functions used is not a critical issue, and the Gabor functions have found extensive use as filter functions. Such localized filters can be defined using a general window function instead of the Gaussian. This leads to the definition of a *windowed Fourier transform* which can be expressed:

$$F(x,u) \ = \ \int_{-\infty}^{\infty} \mathrm{w}(x - x') f(x') e^{-iux'} \mathrm{d}x' \tag{8.10}$$

As above, this transform can be seen as a transform having different complex modulated windows as basis functions. Performing this local transform for all spatial positions results in a combined space-frequency representation, also termed a *spectrogram*. This representation has long been used for analyzing one-dimensional time varying signals (e.g. [AR77]).

Over the last few years, the wavelet concept has gained considerable attention [Dau92, Mey92, WCP92]. A wavelet basis is produced by translation and dilatations of a "mother wavelet". By using dilation to control the frequency

characteristic, a constant relative bandwidth on a logarithmic scale is achieved. The use of such filter sets, based on Gabor filters, was first suggested to the computer vision community by Granlund in 1978 [Gra78]. A feature of constant relative bandwidth filters is that the spatial localization becomes proportional to the local signal wavelength. It is also appropriate to note that filters with almost constant bandwidth are consistent with current multi-frequency channel theories in psychophysics [SK74, MS76, MPU77, Leg78]. A transform based on a set of scaled Gabor functions was introduced by Morlet in 1988.

## 8.2 A LOGNORMAL SPACE-FREQUENCY REPRESENTATION

In this section, local frequency can be interpreted in terms of a single frequency as well as in terms of local spectrum. The notion of instantaneous frequency is used for narrowband analysis over a number of scales. The estimates are weighted and summed to produce a wide band frequency estimate.

### 8.2.1 The lognormal quadrature filter

For convenience, the definition of the lognormal filter in chapter 6 is repeated. A new index $n$ is added to the radial function defining different centre frequencies

$$Q(\mathbf{u}) = R_n(\rho)D_k(\hat{\mathbf{u}}) \tag{8.11}$$

where $\mathbf{u}$ is the multidimensional frequency variable, $R_n(\rho)$ and $D_k(\hat{\mathbf{u}})$ are the radial and the directional function respectively. $\rho$ is the norm and $\hat{\mathbf{u}}$ is the direction of the frequency vector: $\mathbf{u} = \rho\hat{\mathbf{u}}$.

The radial function has the following form:

$$R_n(\rho) = e^{-C_B \ln^2\left(\frac{\rho}{\rho_n}\right)} \tag{8.12}$$

where $C_B = \frac{4}{B^2 \ln 2}$, $\rho_n$ is the centre frequency and $B$ is the 6 dB relative bandwidth in octaves, i.e. $B$ is given by

$$B = (\ln 2)^{-1} \ln\left(\frac{\rho_u}{\rho_l}\right) \tag{8.13}$$

where $\rho_l$ and $\rho_u$ are the $\rho$ values for which $R_n(\rho) = 0.5$.

As in chapter 6, but using bracket notation for the scalar product, the directional function of the quadrature filters has the following form:

$$\begin{cases} D_k(\hat{\mathbf{u}}) = \langle \hat{\mathbf{u}}, \hat{\mathbf{n}}_k \rangle^2 & \text{if} \quad \langle \mathbf{u}, \hat{\mathbf{n}}_k \rangle > 0 \\ D_k(\hat{\mathbf{u}}) = 0 & \text{otherwise} \end{cases} \tag{8.14}$$

where $\hat{\mathbf{n}}_k$ is the filter directing vector, i.e. $D(\hat{\mathbf{u}})$ varies as $\cos^2(\varphi)$, where $\varphi$ is the angle between $\mathbf{u}$ and the filter direction $\hat{\mathbf{n}}_k$ (figure 8.2). For simple signals (equation (6.2)) having an orientation given by $\hat{\mathbf{x}}$, the output magnitude from a quadrature filter in direction $k$ is given by

$$q_{kn} = d_n \langle \hat{\mathbf{x}}, \hat{\mathbf{n}}_k \rangle^2, \tag{8.15}$$

where $d_n$ is independent of the filter orientation and will be determined by the radial distribution of the signal spectrum $F(\rho)$ and the radial filter function $R_n(\rho)$.
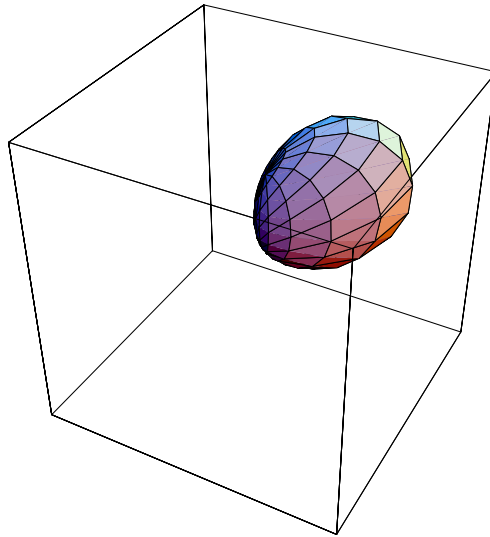


**Figure 8.2**    Iso-surface plot of the directional function, $D_k$, as a function of the signal orienting vector $\hat{\mathbf{x}}$.

## 8.2.2 Summation of directional components

An isotropic estimate of signal strength can be obtained by summing the magnitude of the outputs from a number of quadrature filters having the same frequency response, but different orientations.

$$q_n = \sum_k q_{kn} = \sum_k d_n \langle \hat{\mathbf{x}}, \hat{\mathbf{n}}_k \rangle^2 \tag{8.16}$$

If the filter orienting vectors $\hat{\mathbf{n}}_k$ constitute a tight frame (chapter 3),

$$\sum_k d_n \langle \hat{\mathbf{x}}, \hat{\mathbf{n}}_k \rangle^2 = d_n \beta \|\hat{\mathbf{x}}\|^2 = d_n \beta \tag{8.17}$$

where $\beta$ is the frame bound. In particular, if the filter orienting vectors $\hat{\mathbf{n}}_k$ constitute an orthonormal basis of the same dimensionality as $\hat{\mathbf{x}}$, equation (8.17) reduces to

$$q_n = \sum_k d_n \langle \hat{\mathbf{x}}_k, \hat{\mathbf{n}}_n \rangle^2 = d_n \tag{8.18}$$

since $\beta = 1$ for orthonormal bases. This shows that an isotropic estimate of signal strength, which is local both spatially and in the frequency domain, is obtained by summing the magnitudes of the outputs of oriented quadrature filters constituting a tight frame. The minimum number of filters required in each set is the same as the dimensionality of the signal: the orthogonal case.

## 8.2.3 The ratio of two lognormal filters

By combining the outputs from two or more sets of filters that differ only in centre frequency $\rho_n$, it is possible to produce a local frequency estimate.

For simple, single frequency neighbourhoods, the contribution in the Fourier domain will be concentrated at two points at the distance $\rho$ from the origin. The isotropic magnitude of a sum of filters is then given by equation (8.12),

$$q_n = d_n = A e^{-C_B \ln^2(\frac{\rho}{\rho_n})} \tag{8.19}$$

where $A$ is the local signal amplitude. The ratio between the outputs from two filters differing only in their centre frequencies $\rho_n$ and $\rho_m$ is

$$\frac{q_m}{q_n} = \frac{e^{-C_B \ln^2\left(\frac{\rho}{\rho_m}\right)}}{e^{-C_B \ln^2\left(\frac{\rho}{\rho_n}\right)}} \tag{8.20}$$

Simplification of this equation leads to a simple power relation [Knu82],

$$\frac{q_m}{q_n} = e^{C_B\left[\ln^2\left(\frac{\rho}{\rho_n}\right) - \ln^2\left(\frac{\rho}{\rho_m}\right)\right]}$$

$$= e^{C_B\left[\left(\ln\rho - \ln\rho_n\right)^2 - \left(\ln\rho - \ln\rho_m\right)^2\right]}$$

$$= e^{C_B\left[2\left(\ln\rho_m - \ln\rho_n\right)\ln\rho + \ln^2\rho_n - \ln^2\rho_m\right]}$$

$$= e^{C_B\left[\ln\rho_m - \ln\rho_n\right]\left[2\ln\rho - \ln\rho_m - \ln\rho_n\right]}$$

$$= e^{C_B\left[\ln\left(\frac{\rho_m}{\rho_n}\right)\right]\left[2\ln\left(\frac{\rho}{\sqrt{\rho_m\rho_n}}\right)\right]}$$

$$= \left(\frac{\rho}{\sqrt{\rho_n\rho_m}}\right)^{2C_B\ln\left(\frac{\rho_m}{\rho_n}\right)} \tag{8.22}$$

It is convenient at this point to introduce two constants. The first, $\alpha$, depends on the relative bandwidth and the ratio of the centre frequencies of the filters, and the second, $\rho_{(nm)}$, is the geometric mean of the two centre frequencies.

$$\alpha = \left[2C_B\ln\left(\frac{\rho_m}{\rho_n}\right)\right]^{-1} \tag{8.22}$$

$$\rho_{(nm)} = \sqrt{\rho_n\rho_m} \tag{8.23}$$

The ratio between two lognormal filter outputs can then be written

$$\frac{q_m}{q_n} = \left(\frac{\rho}{\rho_{(nm)}}\right)^{\alpha^{-1}} \quad . \tag{8.24}$$

Solving for $\rho$ yields:

$$\rho = \rho_{(nm)}\left(\frac{q_m}{q_n}\right)^{\alpha} \tag{8.25}$$

A particularly simple situation occurs if the filters are chosen such that $\alpha = 1$. Applying equation (8.22) then gives the following relation between the ratio of the filter frequencies and the relative bandwidth

$$\frac{\rho_m}{\rho_n} = 2^{\frac{B^2}{8}} \tag{8.26}$$

A centre frequency ratio of 2 is a natural choice and has been shown to work well; for $\alpha = 1$, this choice requires a relative filter bandwidth $B$ of $2\sqrt{2}$.
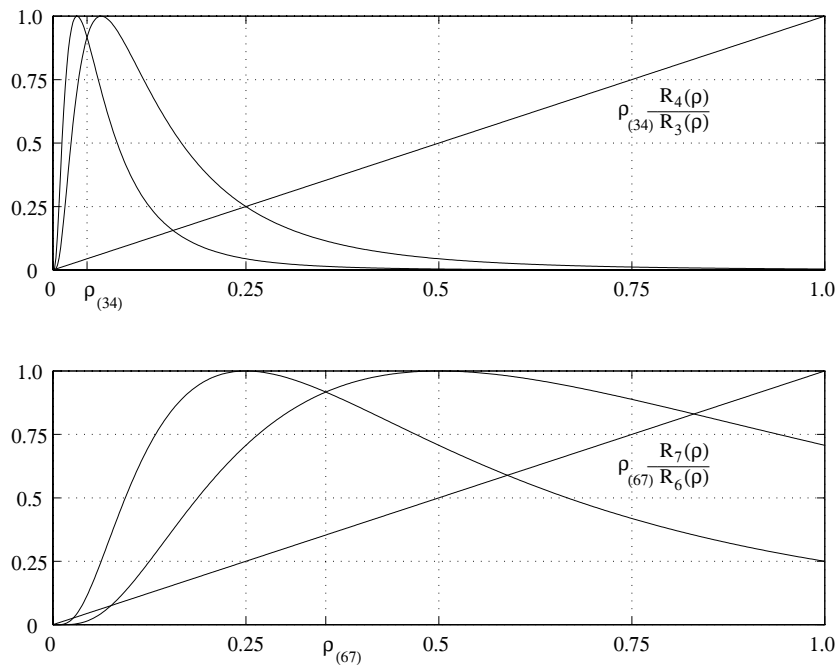


**Figure 8.3**   The local radial frequency $\rho$ can be estimated as the ratio between two lognormal functions times $\rho_{(n\ \ n+1)}$, the geometric mean of the two centre frequencies, if $\alpha = 1$ (equation (8.25)).

## 8.2.4   The relation between lognormal ratios and instantaneous frequency

In this section it is shown that the magnitude of the ratio between two lognormal quadrature filters can be interpreted as the instantaneous frequency of a filtered simple signal. To start, note that a quadrature filter suppresses all negative

frequencies and produces a filtered analytic signal $s_{An}$, where $n$ indicates the filter center frequency. Also note that for simple signals the problem can be treated as one-dimensional, since $\mathbf{u} = \rho\hat{\mathbf{x}}$ for all parts in the Fourier domain where the signal contribution is non-zero. For this reason $\rho$ can be thought of as equivalent to the one-dimensional frequency variable $u$. The instantaneous frequency $\omega_n$ for $s_{An}$ can then be written

$$\omega_i = \frac{\partial}{\partial x} \arg(s_{An}) \tag{8.27}$$

where $x$ is the projection of the spatial coordinate $\boldsymbol{\xi}$ on the signal directing vector $\hat{\mathbf{x}}$, i.e. $x = \boldsymbol{\xi}^T \hat{\mathbf{x}}$.

Equation (8.27) can be rewritten as

$$\omega_n = \text{Im}[\frac{\frac{\partial}{\partial x} s_{An}}{s_{An}}] \tag{8.28}$$

The equivalent to differentiation in the spatial domain is multiplication by the frequency variable and a phase shift of $\frac{\pi}{2}$ in the Fourier domain, i.e. for the derivative in equation (8.27):

$$\frac{\partial}{\partial x} s_{An} \;\leftrightarrow\; i\rho S_n \tag{8.29}$$

where $S_n$ is the Fourier transform of $s_{An}$.

$S_n$ is the product of the frequency response of the the quadrature filter $F_n$ and the Fourier transform of the signal $S$.

$$\frac{\partial}{\partial x} f_{An} \;\leftrightarrow\; i\,\rho F_n\, S \tag{8.30}$$

If the lognormal quadrature filter set is designed so that $\alpha = 1$, the following relation holds, (rewriting equation (8.25)).

$$\rho\, F_n = \rho_{(nm)} F_{n+1} \tag{8.31}$$

Combining equation (8.30) and equation (8.31) yields

$$\frac{\partial}{\partial x} f_{An} \;\leftrightarrow\; i\,\rho_{(nm)} F_{n+1}\, S \;=\; i\,\rho_{(nm)} S_{n+1} \tag{8.32}$$

Expressing the right hand side in spatial terms, equation (8.32) gives the interesting result

$$\frac{\partial}{\partial x} \, s_{An} \;=\; i \, \rho_{(nm)} \, s_{A(n+1)} \tag{8.33}$$

This implies that, given the output from one lognormal filter, a whole set of lognormal filters outputs can be generated by repeated spatial differentiation. The filters corresponding to the outputs will have the same relative bandwidth and geometrically increasing centre frequencies.

Finally, the instantaneous frequency can be written

$$\omega_n \;=\; \rho_{(nm)} \, \mathrm{Re}[\frac{s_{A(n+1)}}{s_{An}}] \tag{8.34}$$

The narrow-band instantaneous frequency estimates used in the present method differ somewhat from equation (8.34) in that the *magnitudes* of the filter ratios are used, i.e.

$$\omega_n \;=\; \rho_{(nm)} \, \|\frac{s_{A(n+1)}}{s_{An}}\| \;=\; \frac{q_{n+1}}{q_n} \tag{8.35}$$

For complex exponential signals equation (8.34) and equation (8.35) will be identical. However, for broadband signals, the estimates produced by equation (8.35) are less sensitive to phase interference effects and guarantee non-negativity. Negative estimates will frequently occur if equation (8.34) is used [GK94].

## 8.3   WIDE RANGE FREQUENCY ESTIMATION

Using two band-pass filters to estimate local frequency will in most situations only work well if the signal spectrum falls within the range of the filters. However, a wide range local frequency estimate can be obtained by weighted summation of a number of filter pairs covering an extended frequency range. Given a set of $N$ quadrature filters with a relative bandwidth of $2\sqrt{2}$ and the centre frequencies given by,

$$\rho_n \;=\; 2^n \, \rho_0 \tag{8.36}$$

Then a wide range frequency estimate can be obtained as

$$\rho = [\sum_{n=1}^{N-1} c_n]^{-1} \sum_{n=1}^{N-1} c_n \frac{q_{n+1}}{q_n} \sqrt{\rho_n \rho_{n+1}} \tag{8.37}$$

where $\sqrt{\rho_n \rho_{n+1}} = \rho_0 2^{n+0.5}$ and where $c_n$ are the weighting coefficients.

For single frequency neighbourhoods, all filter pairs will estimate the same frequency, see figure 8.3, and any choice of weighting coefficients $c_n$ will yield the same result. In practice, however, the choice of weighting coefficients is important. To obtain reliable estimates the weighting coefficients should reflect the amount of signal energy that a given filter pair encounters. A choice of $c_n$ that satisfies this requirement and at the same time leads to a particularly simple estimation algorithm is:

$$c_n = q_n \tag{8.38}$$

With this choice, equation (8.37) becomes:

$$\tilde{\rho} = \rho_0 [\sum_{n=1}^{N-1} q_n]^{-1} \sum_{n=1}^{N-1} 2^{n+0.5} q_{n+1} \tag{8.39}$$

i.e. , a wide range frequency estimate is obtained as the ratio between two sums of lognormal filter outputs. This estimation algorithm was used in the experiments below, with eight filters having a relative filter bandwidth $B$ of $2\sqrt{2}$, one octave apart.

## 8.3.1   Estimation of spectrum variance

The local frequency may be considered the average of the frequencies present at a given spatial position. It is useful to have a measure defining the deviation from this value. Such a measure may be interpreted as *instantaneous bandwidth* [Boa92a], and will be directly related to the reliability of the mean frequency estimate, $\tilde{\rho}$. The broader the bandwidth, the more uncertain the local frequency estimate. An estimation of the local signal spectrum variance can be obtained by summing the squared differences between the wide range estimation of the mean frequency, $\tilde{\rho}$, and the estimates obtained by the individual filter pairs, i.e.

$$\tilde{\sigma}^2 = [\sum_{n=1}^{N-1} q_n^2]^{-1} \sum_{n=1}^{N-1} q_n^2 [2^{n+0.5} \frac{q_{n+1}}{q_n} - \tilde{\rho}]^2 \tag{8.40}$$

A certainty measure, $c$, based on the local spectrum variance can then be defined as:

$$c = \frac{1}{1 + \sigma^2} \tag{8.41}$$

giving $c = 1$ as maximum certainty (when the variance i zero). This is the certainty measure used in the examples below.

## 8.4  EXPERIMENTAL RESULTS

The first example contains a non-stationary signal, i.e. a signal with a space-variant spectrum. It is constructed from three sinusoids covering four octaves having the frequencies $\frac{\pi}{64}$, $\frac{\pi}{16}$ and $\frac{\pi}{4}$ (figure 8.4). Note that the certainties of the estimates are high in the constant frequency regions and low in the regions where transitions between two frequencies occur.

The test signal in the second example (figure 8.5) is constructed as a sum of two components, a constant frequency component, $f = \pi/4$, and a chirp signal which has linearly increasing frequency from $f = \pi/8$ to $f = 3\pi/8$. This means that the single frequency model is not valid. In this case the algorithm estimates the mean of the two frequency components. Because of interference between the frequencies, the mean frequency is difficult to measure at some spatial positions. In these singular points the certainty of the estimate is close to zero. A smoother estimate can be obtained using normalized averaging. Using this technique an adaptive smoothing can be performed based on the accompanying certainty values. A new estimate of the certainty after this operation is also provided, as shown by the dashed line.

The third one-dimensional example contains a test signal with one constant mean frequency with exponentially increasing bandwidth from $B = 0.25$ to $B = 4$ octaves (figure 8.6). The broader the bandwidth, the weaker the notion of instantaneous frequency becomes. This is why the estimates become more and more noisy to the right in the middle right figure, and why the certainty decreases in the bottom right figure. The estimates after normalized averaging are also plotted in these figures.

In all two-dimensional examples below, five quadrature filters were used. The centre frequencies of the filters are $(\pi/\sqrt{2}, \pi/2, \pi/2\sqrt{2}, \pi/4, \pi/4\sqrt{2})$ The filters have a relative bandwidth of $B = 2\sqrt{2}$.

Figure 8.7 (top) shows a signal, constructed for two dimensions, consisting of a sum of two components and similar to the one-dimensional signal of figure 8.5.
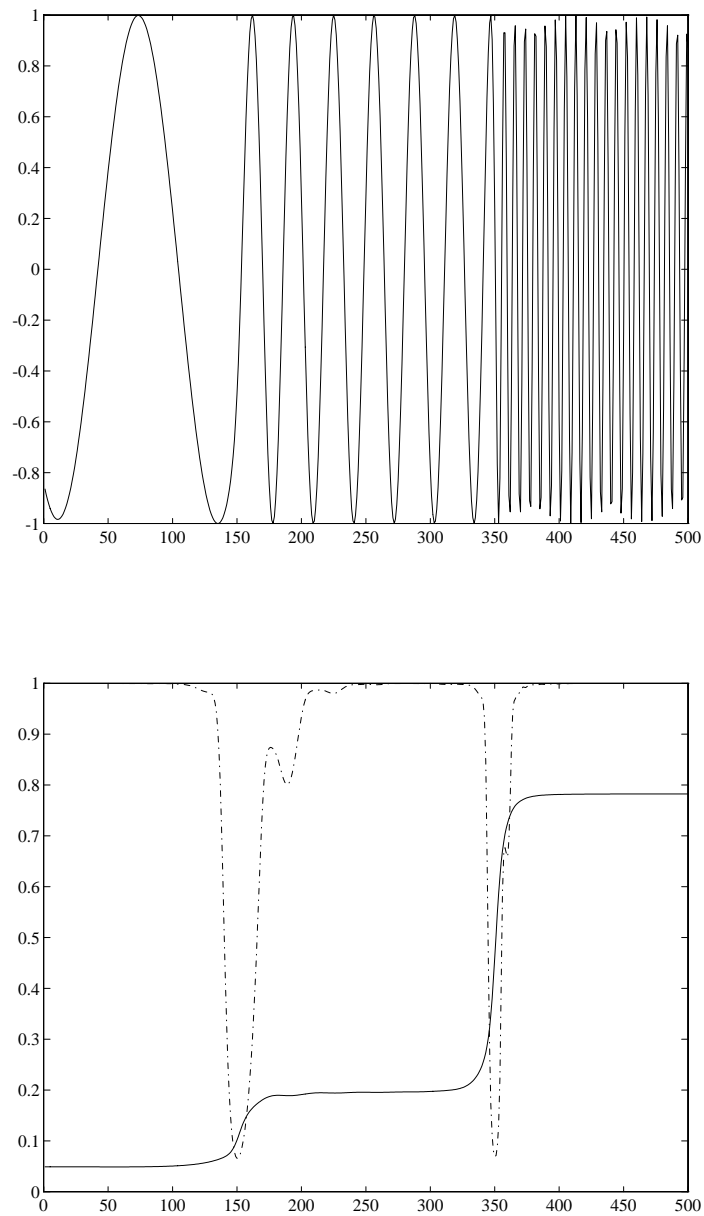
**Figure 8.4** **Top**: A test signal constructed by sinusoids of a stepwise increasing frequency ($\pi/64$, $\pi/16$, $\pi/4$). **Bottom**: Estimated local frequency (solid) and certainty (dashed).
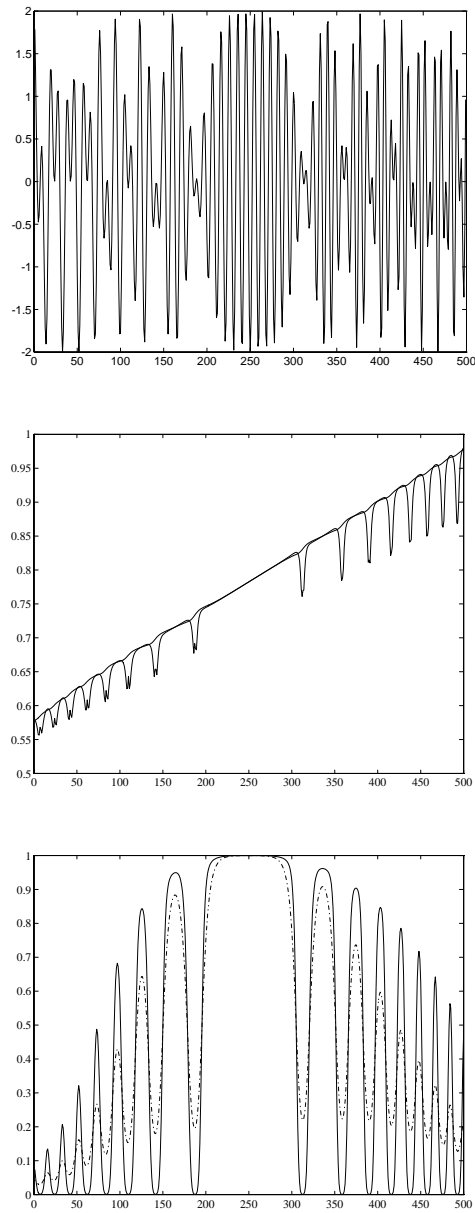
**Figure 8.5** **Top**: A test signal containing one constant and one linearly increasing frequency component. **Top**: Estimated local frequency before and after normalized averaging; the smoother trace is after normalized averaging of the estimate, using the below certainty estimate. **Bottom**: Estimated certainty before and after the normalized averaging step (dashed).
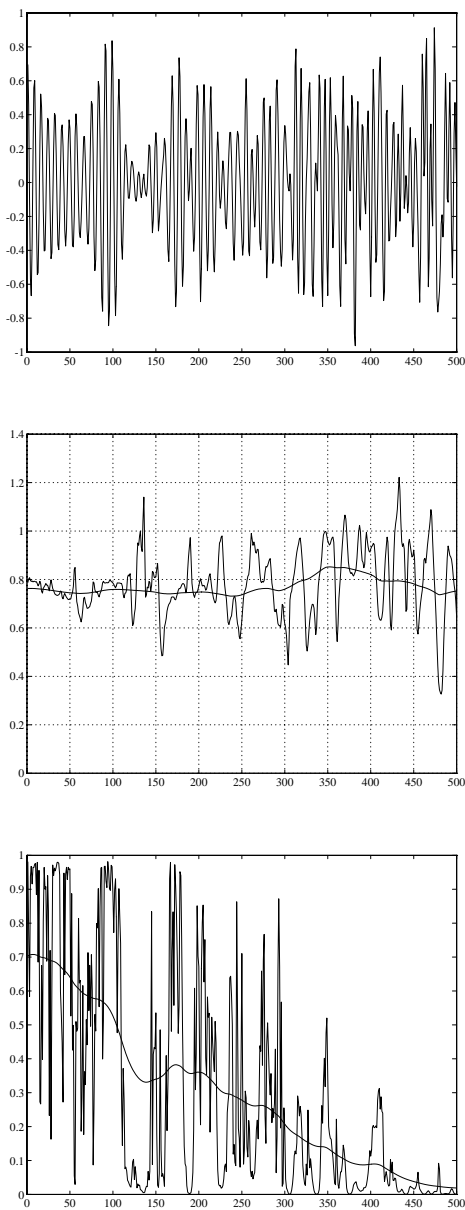
**Figure 8.6** **Top**: A test signal having constant mean frequency and a bandwidth exponentially increasing to the right. **Middle**: Estimated local frequency; the smoother trace is after normalized averaging the frequency estimate using the certainty estimate below. **Bottom**: Estimated certainty and before and after normalized averaging.

The first component has a constant frequency; the second component has a frequency that increases toward the centre. All the images have size $500 \times 384$ pixels. Estimated local frequency is shown in the middle figure. White corresponds to high frequency. Note the similar ringing of the estimates in the bottom image and those in figure 8.5. The one-dimensional signal in figure 8.5 corresponds to a line from the left top corner to the right bottom corner in bottom left figure. As in the one-dimensional case, a more robust estimate can be obtained using normalized averaging (figure 8.7 (bottom)). The filter used in the normalized averaging has the size $15 \times 15$ and is very sharp, with the magnitude decreasing as the reciprocal of the distance from the filter centre.

Another two-dimensional example is shown in figure 8.8. This image is constructed having regions from two different testpatterns that are overlapping in radial sections. One component is a narrow-band noise process and the other component has increasing frequency towards the centre covering 5 octaves. In the overlapping regions the algorithm estimates the mean frequency component. If the frequency components differ a lot, this is captured in the estimated certainty.

Figure 8.9 contains a third two-dimensional example. The test image contains a circular area having constant frequency surrounded by an area having a frequency gradient.
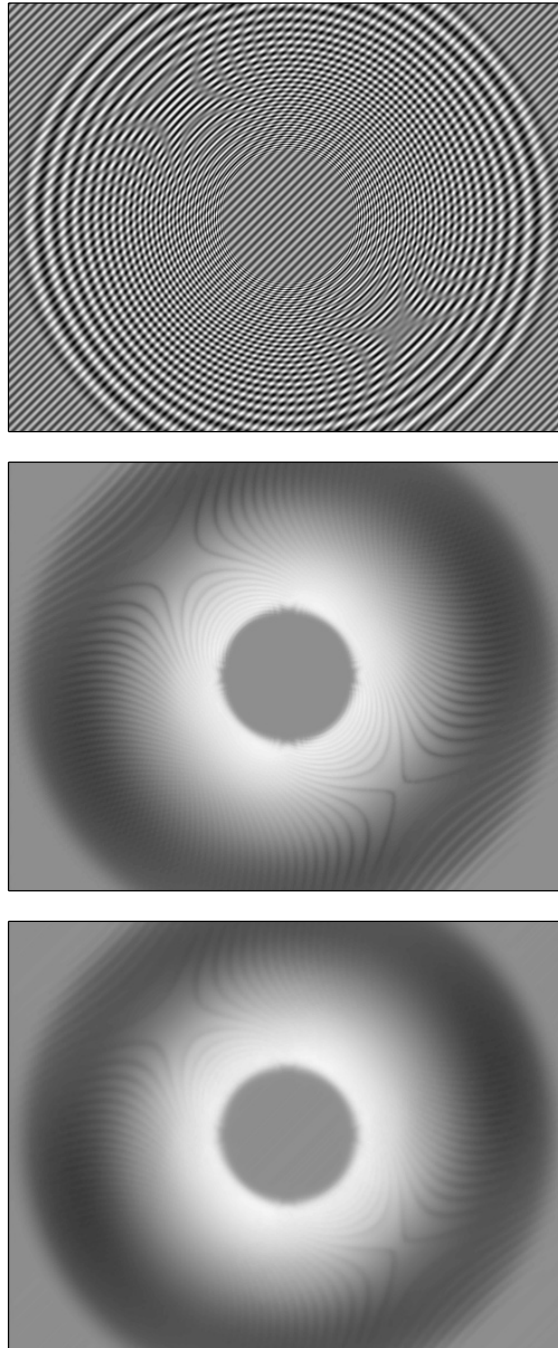
**Figure 8.7** **Top:** A test image consisting of a constant frequency plus a component with a frequency that increases toward the centre. **Middle**: Estimated local frequency. White corresponds to high frequency. **Bottom**: Updated frequency estimate using normalized averaging.
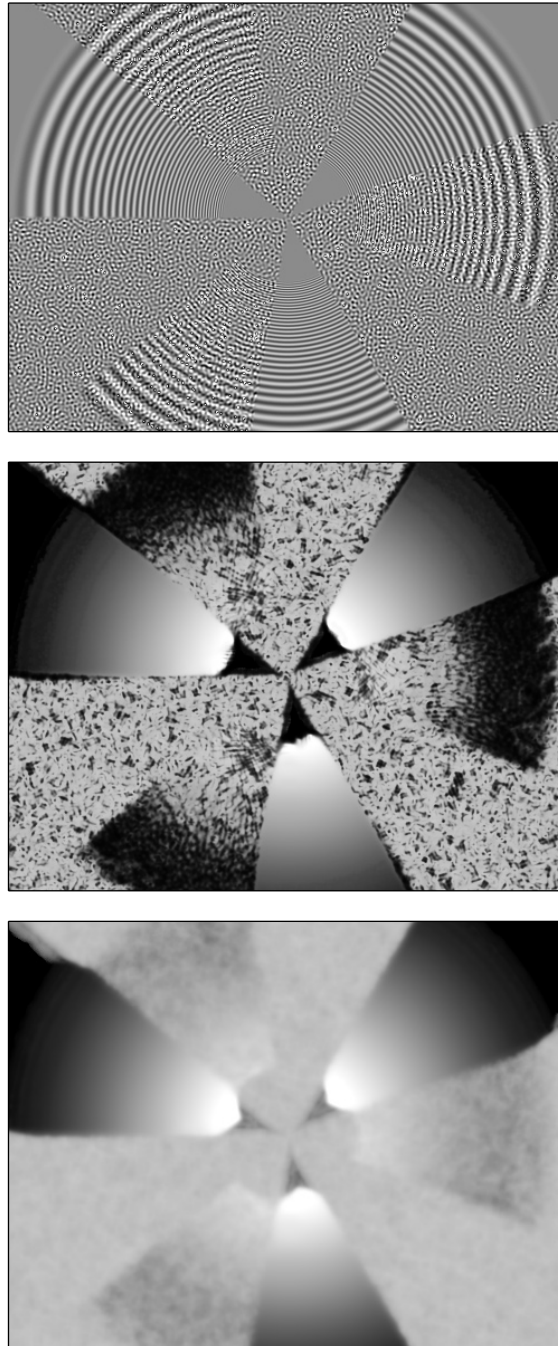
**Figure 8.8    Top:** The test image contains three different areas. One component has increasing frequency towards the the centre covering 5 octaves, another part contains a narrow-band noise process and the third part is a normalized sum of the two signals. **Middle**: Estimated local frequency. White corresponds to high frequency. In the overlapping regions the alghorithm estimates the mean frequency component. **Bottom**: Updated frequency estimate using normalized averaging.
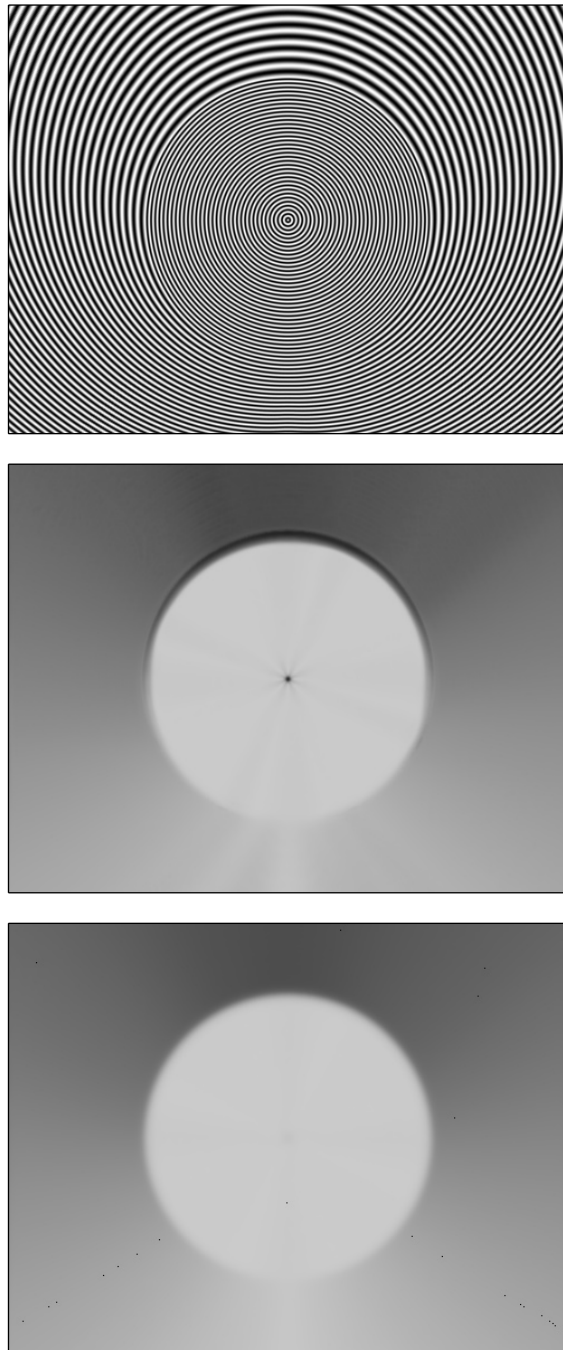
**Figure 8.9  Top**: A test image containing a circular area having constant frequency surrounded by an area having a frequency gradient. **Middle**: Estimated local frequency. White corresponds to high frequency. **Right**: Updated frequency estimate using normalized averaging.

# REFERENCES

[AB85] 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.

[Alo90] J. Aloimonos. Purposive and qualitative active vision. In *DARPA Image Understanding Workshop*, Philadelphia, Penn., USA, September 1990.

[AR77] J. B. Allen and L. R. Rabiner. A unified approach to short-time fourier analysis and synthesis. In *Proc. IEEE*, volume 65:11, pages 1558–1564, November 1977.

[Bår91] H. Bårman. *Hierarchical 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.

[Bas80] M. J. Bastiaans. Wiener distribution function and its application to 1st order optics. *J. Opt. Soc. Amer.*, 69:1710–1716, 1980.

[BFB94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[BGW91] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation: texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–13(8), August 1991.

[BHKG91] 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.

[Big88] J. Bigün. *Local Symmetry Features in Image Processing.* PhD thesis, Linköping University, Sweden, 1988. Dissertation No 179, ISBN 91–7870–334–4.

[Boa92a]    B. Boashash. Estimating and interpreting the instantaneous frequency of a signal - part 1: Fundamentals. *Proceedings of the IEEE*, 80(4), 1992.

[Boa92b]    B. Boashash. Estimating and interpreting the instantaneous frequency of a signal - part 2: Algorithms and applications. *Proceedings of the IEEE*, 80(4), 1992.

[Bur88]    P. J. Burt. Moment images, polynomial fit filters and the problem of surface interpolation. In *Proc. of Computer Vision and Pattern Recognition, Ann Arbor*. Computer Society Press, 1988.

[Cal89]    A. Calway. *The Multiresolution Fourier Transform*. PhD thesis, University of Warwick, 1989.

[CB92]    R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proceedings of ECCV–92, LNCS–Series Vol. 588*. Springer–Verlaag, 1992.

[CF37]    J. Carson and T. Fry. Variable frequency electric circuit theory with application to the theory of frequency modulation. *Bell Syste Tech. J.*, 16:513–540, 1937.

[Dan80]    P. E. Danielsson. Rotation invariant operators with directional response. In *Proceedings 5'th Int. Conf. on Pattern Recognition*, Miami Beach, Florida, 1980.

[Dau92]    I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[Dav86]    E. R. Davis. Image space transforms for detecting straight edges in industrial parts. *Pattern Recognition Letters*, Vol 4:447–456, 1986.

[DH72]    R. O. Duda and P. E. Hart. Use of the Hough transform to detect lines and cures in pictures. *Communications of the Association Computing Machinery*, 15, 1972.

[DH73]    R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. Wiley-Interscience, New York, 1973.

[DL78]    S. A. Dudani and A. L. Luk. Locating straight/lines edge segmentgs on outdoor scenes. *Pattern Recognition*, 10:145–157, 1978.

[Fle92]    D. J. Fleet. *Measurement of image velocity*. Kluwer Academic Publishers, 1992. ISBN 0–7923–9198–5.

[Gab46]    D. Gabor. Theory of communication. *Proc. Inst. Elec. Eng.*, 93(26):429–441, 1946.

[GK83]    G. H. Granlund and H. Knutsson. Contrast of structured and homogenous representations. In O. J. Braddick and A. C. Sleigh, editors, *Physical and Biological Processing of Images*, pages 282–303. Springer Verlag, Berlin, 1983.

[GK94]    G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1994.

[Gra78]   G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–178, 1978.

[Hag92]   L. Haglund.  *Adaptive Multidimensional Filtering*.  PhD thesis, Linköping University, Sweden, S–581 83 Linköping, Sweden, October 1992. Dissertation No 284, ISBN 91–7870–988–1.

[HJ92]    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.

[HM84]    A. C. Harvey and C. R. McKenzie. Missing observations in dynamic econometric models: A partial synthesis. In *Time series analysis of irregularly observed data, Proceedings*, 1984.

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

[Hou62]   P. V. C. Hough.  A method and means for recognizing complex patterns. U.S. Patent 3,069,654, 1962.

[HS81]    B. K. P. Horn and B. G. Schunk.  Determining optical flow.  *AI*, pages 185–204, 1981.

[Hug89]   R. I. G. Hughes. *The structure and interpretation of quantum mechanics*. Harvard University Press, 1989. ISBN: 0-674-84391-6.

[HW78]    D. H. Hubel and T. N. Wiesel. Brain mechanisms of vision. *Scientific American*, September 1978.

[IK88]    J. Illingworth and J. Kittler.  A survey of the Hough transform. *Computer Vision, Graphics and Image Processing*, 44, 1988.

[Jau68]   J. M. Jauch. *The Foundations of Quantum Mechanichs*. Reading, MA:Addison-Wesley, New York, 1968.

[KA94]    H. Knutsson and M. Andersson. N-dimensional orientation estimation using quadrature filters and tensor whitening. In *Proceedings of IEEE International Conference on Acoustics, Speech, & Signal Processing*, Adelaide, Australia, April 1994. IEEE.

[Kai80]    T. Kailath. *Linear Systems*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, N.J., 1980.

[Kan86]    K. Kanatani. Structure and motion from optical flow under orthographic projection. *Computer Vision, Graphics and Image Processing*, 35:181-199, 1986.

[Kay88]    D. C. Kay. *Schaum's outline of theory and problems of tensor calculus*. McGraw-Hill, 1988.

[Ken77]    D. E. Bourne P.C. Kendall. *Vector Analysis and Cartesian Tensors*. Van Nostrand Reinhold (UK), 1977. ISBN: 0 442 30743 8.

[KG86]     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 descrete picture elements. Swedish patent 8502569-0 1986(U.S. Patent 4.747.151, 1988), 1986.

[KGB90]    H. Knutsson, G. H. Granlund, and H. Bårman. A note on estimation of 4D orientation. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 192–195, Linköping, Sweden, March 1990. SSAB. Report LiTH–ISY–I–1089, Linköping University, Sweden, 1990.

[KHG86]    H. Knutsson, M. Hedlund, and G. H. Granlund. Apparatus for determining the degree of consistency of a feature in a region of an image that is divided into descrete picture elements. Swedish patent 8502570-8 1986 (U.S. Patent 4.747.152, 1988), 1986.

[Knu82]    H. Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.

[Knu85]    H. 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.

[Knu89]    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.

[KP94]     P. Kube and P. Perona. Scale-sspace properties of quadratic edge detectors. In J-O Eklundh, editor, *Computer Vision - ECCV'94*, number 800 in Lecture Notes in Computer Science, pages 115–122, Berlin, 1994. Springer-Verlag. ISBN 3-450-57957-7.

[KvD75]     J. J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Opt. Acta 22*, pages 773–791, 1975.

[KW93]      H. Knutsson and C-F Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York City, USA, June 1993. IEEE.

[KWG94]     H. Knutsson, C-F Westin, and G.H. Granlund. Local Multiscale Frequency and Bandwidth Estimation. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, November 1994. IEEE.

[KWW93]     H. Knutsson, C-F Westin, and C-J Westelius. Filtering of uncertain irregularity sampled multidimensional data. In *Twenty-seventh Asilomar Conf. on Signals, Systems & Computers*, Pacific Grove, California, USA, November 1993. IEEE.

[KWWK95]    J. Karlholm, C-J. Westelius, C-F. Westin, and H. Knutsson. Object tracking based on the orientation tensor concept. In *Proceedings of the 9th Scandinavian conference on Image Analysis*, Uppsala, Sweden, June 1995. SCIA. submitted Octobewr 1994.

[Leg78]     G. E. Leggs. Space domain properties of a spatial frequency channel in human vision. *Vision Research*, 18:959–969, 1978.

[LHK93]     T. Landelius, L. Haglund, and H. Knutsson. Depth and velocity from orientation tensor fields. In *Proceedings of the 8th Scandinavian conference on Image Analysis*, Tromsø, Norway, May 1993. NOBIM. Report LiTH-ISY-R-1529, Linköping University, Sweden, 1993.

[LK81]      B. Lucas and T. Kanade. An Iterative Image Registration Technique with Applications to Stereo Vision. In *Proc. Darpa IU Workshop*, pages 121–130, 1981.

[LM75]      J. O. Limb and J. A. Murphy. Estimating the velocity of moving images in television signals. *Computer Graphics and Images Processing*, pages 311–327, 1975.

[LRS89]     D. Letalick, I. Renhorn, and 0. Steinvall. Noise sources in laser radar systems. *Applied Optics*, 28:2657–2665, 1989.

[Mac69]     D. M. MacKay. *Information, Mechanism and Meaning*. M.I.T. Press, Cambridge, Massachusetts and London, England, 1969.

[Mey92]     Y. Meyer. *Wavelets and operators.* Cambridge studies in advances mathematics. Cambridge University Press, 1992. ISBN 0 521 42000 8 hardb.

[MI92]      F. A. Mussa-Ivaldi. From basis functions to basis fields:vector field approximation from sparse data. *Biological Cybernetics*, 67(6):479–489, October 1992.

[MPU77]     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.

[MS76]      H. Mostafavi and D. J. Sakrison. Structure and properties of a single channel in the human visual system. *Vision Research*, 16:957–968, 1976.

[NA88]      R. C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance: towards qualitative vision. In *Proc. 2nd Int. Conf. on Computer Vision*, pages pages 188–196, 1988.

[OABB85]    J. M. Ogden, E. H. Adelson, J. R Bergen, and P. J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30-5, Sept 1985.

[Pen92]     A. Pentland. Surface interpolation using wavelets. In *Proceedings of ECCV–92, LNCS–Series Vol. 588.* Springer–Verlaag, May 1992.

[Per92]     P. Perona. Steerable-scalable kernels for edge detection and junction analysis. In *Proceedings ECCV Conf. on Computer Vision*, pages 3–18, 1992.

[PIK90]     J. Princen, J. Illingworth, and J. Kittler. A hierarchical approach to line extraction based on the Hough transform. *Computer Vision, Graphics, and Image Processing*, 52, 1990.

[PZ88]      M. Porat and Y. Y. Zeevi. The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):452–467, 1988.

[SAH91]     E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 310–315, Maui, 1991. IEEE.

[SK74]      C. F. Stromeyer and S. Klein. Spatial frequency channels in human vision as asymmetric (edge) mechanisms. *Vision Research*, 14:1409–1420, 1974.

[Sto89]    J. J. Stoker. *Differential Geometry*. Wiley & Sons, New York, USA, 1989.

[Str80]    G. Strang. *Linear Algebra and Its Applications*. Academic Press, 1980.

[Tho82]    D. J. Thomson. Spectrum estimation and harmonic analysis. *Proc. IEEE*, 70:1055–1096, 1982.

[TS91]     M. Tistarelli and G. Sandini. Direct estimation of time–to–impact from optical flow. In *Proceedings of IEEE Workshop on Visual Motion*, pages 52–60, Princeton, USA, October 1991. IEEE, IEEE Society Press.

[vdP46]    B. van der Pol. The fundamental principles of frequency modulation. *Proceedings of the IEEE*, 93:153–158, 1946.

[WAJA85]   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.

[WB92]     R. Wilson and A. H. Bhalero. Kernel designs for efficient multiresolution edge detection and orientation estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:384–389, 1992.

[WCP92]    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.

[Wes91]    C-F Westin. Feature extraction based on a tensor image description, September 1991. Thesis No. 288, ISBN 91–7870–815–X.

[Wes92]    C-J Westelius. Preattentive gaze control for robot vision, June 1992. Thesis No. 322, ISBN 91–7870–961–X.

[WHKG89]   J. Wiklund, L. Haglund, H. Knutsson, and G. H. Granlund. Time sequence analysis using multi-resolution spatio-temporal filters. In *The 3rd International Workshop on Time-Varying Image Processing and Moving Object Recognition*, pages 258–265, Florence, Italy, May 1989. Invited Paper. Report LiTH–ISY–I–1014, Computer Vision Laboratory, Linköping University, Sweden, 1989.

[WK88]     R. Wilson and H. Knutsson. Uncertainty and inference in the visual system. *IEEE Transactions on Systems, Man and Cybernetics*, 18(2), March/April 1988.

[WK92a]    C-F Westin and H. Knutsson. Extraction of local symmetries using tensor field filtering. In *Proceedings of 2nd Singapore International Conference on Image Processing*. IEEE Singapore Section, September 1992. LiTH–ISY–R–1515, Linköping University, Sweden.

[WK92b]    C-F Westin and H. Knutsson. The Möbius strip parameterization for line extraction. In *Proceedings of ECCV–92, LNCS–Series Vol. 588*. Springer–Verlag, 1992. LiTH–ISY–R–1514, Linköping University, Sweden.

[WK94]    C-F Westin and H. Knutsson. Estimation of Motion Vector Fields using Tensor Field Filtering. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, November 1994. IEEE.

[WKG82]    R. Wilson, H. Knutsson, and G. H. Granlund. The operational definition of the position of line and edge. In *The 6th International Conference on Pattern Recognition*, Munich, Germany, October 1982.

[WNK94]    C-F Westin, K. Nordberg, and H. Knutsson. On the equivalence of normalized convolution and normalized differential convolution. In *Proceedings of IEEE International Conference on Acoustics, Speech, & Signal Processing*, Adelaide, Australia, April 1994. IEEE.

[WWW94]    C-J Westelius, J. Wiklund, and C-F Westin. Prototyping, visualization and simulation using the application visualization system. In H. I. Christensen and J.L. Crowley, editors, *Experimental Environments for Computer Vision and Image Processing*, volume 11 of *Series on Machine Perception and Artificial Intelligence*, pages 33–62. World Scientific Publisher, 1994. ISBN 981-02-1510-X.

[You78]    E. C. Young. *Vector and Tensor Analysis*. Dekker, 1978.

[You88]    N. Young. *An introduction to Hilbert space*. Cambride University Press, 1988. ISBN: 0 521 33071 8 hardb.

# INDEX