EDUPACK LiU.CVL.ORIENTATION

Version: November 30, 2009

By Hans Knutsson.

Revision, adaptation to the Edupack format and compilation of exercises by Klas Nordberg.

Summary

The material presented in this package is focused on representation and estimation of local orientation in multi-dimensional signals using tensors. The package covers the 2D, 3D, and 4D cases of signal dimensionality, but the theory can be extended to arbitrary finite dimensionality. Estimation of velocity is treated as a special case of 3D methods.

The package contains a theory part, an exercise part, and a computer exercise part based on Matlab.

Scientific Background

The material in this package is a compilation and adaptation of reports, compendiums, and books produced by the Computer Vision Laboratory of Linköping University for use in our computer vision undergraduate courses. The foundation of this material are the following publications

- Granlund [2], which describes the vector representation for local 2D orientation, and an estimation procedure for this representation based on Gabor filters.
- Knutsson [10], which describes a consistent approach for the estimation of the 2D vector representation using separable quadrature filters.
- Knutsson [6], which describes the tensor representation for local multi-dimensional structures, thereby extending the reprsentation domain from 2D signals to signals of arbitrary finite dimensionality.
- Knutsson, Bårman, and Haglund [8], which describes estimation procedures for the tensor representation in 2D, 3D, and 4D, using separable quadrature filters.
- Knutsson and Andersson [7], which describes the general mechanism for estimation of the local orientation tensor using in principle an arbitrary number of filters in arbitrary directions.

These results are summarized and further developed by Hans Knutsson in the book

- Granlund, Knutsson:
  ”Signal Processing for Computer Vision” [3].

Acknowledgement

As for the exercises, including the computer exercises, some of them have appeared in course material used for the undergraduate course, and result from efforts of many of the PhD students which have been working at CVL over the years. Other exercises are specifically designed for this package, and they, together with the compilation and adaptation of the entire package, have been brought together by Klas Nordberg.

The production of this package has been possible only through generous contributions from several sources, of which the following sponsors should be mentioned: Hewlett Packard as part of their Imaging Systems Engineering programme, the TEKIT board of Linköping University, and the Swedish research programme VISIT.

A special thanks to Björn Johansson and Todd Reed who have read early versions of the package and provided valuable feedback in terms of comments and a long list of typos and errors. Please send feedback to the authors if you have comments or find errors.

Prerequisites

In In order to comprehend this package you will have to be familiar with the following concepts:

- From linear algebra: vector spaces, scalar products, bases and dual bases, vectors and matrices, decomposition of symmetric matrices using eigenvalues and eigenvectors.
- From signal processing: the Fourier transforms of multidimensional functions. In particular the concepts of simple signals and the properties of their Fourier transforms. See chapter 4 of [3].

Feedback

This package is a dynamic document, and we invite you to send feedback to the authors in order to make it dynamic. If you find errors, or have a question, or think that some part of the text is unclear, it is likely that your comments will modify the package in some way or another.

Send comments to

klas@isy.liu.se

1 Introduction

This package presents a unified approach and a theory for estimation of local orientation and velocity in multi-dimensional spatial or spatio-temporal signals. The presentation is intended to serve as a basis for the design of efficient multi-dimensional signal analysis algorithms.

Multi-dimensional signals can be seen as functions of more than one variable and there are in principle no restrictions on what these variables may represent. However, signals as functions of space or space-time are by far the most frequently used. New techniques to produce and process multi-dimensional data are constantly being developed in a number of different fields.

The techniques presented in this package are based on spatial or spatio-temporal filtering and tensor representations. This means that each neighborhood of the signal is linearly combined with a set of filters. The filters employed here are quadrature filters, each sensitive to a particular orientation in space. The quadrature property implies that the filter outputs are complex numbers, representing a magnitude and a phase. 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 orientation.

In the 3D case, the resulting tensors contain information about how *plane-like* or *line-like* the neighborhoods are, in addition to the orientation description. If the 3D signal corresponds to a spatio-temporal sequence, velocity estimates can be directly obtained from the corresponding orientation tensors. In line-like, i.e., ”moving point” type regions, a true velocity estimate is produced. In plane-like, i.e., ”moving line” type regions, the velocity component perpendicular to the local structure is produced.

A few examples are given below of commonly available multi-dimensional data sets to which the techniques presented in this package could be applied.

- **Images (2D)**

  Photograph
  X-ray
  Satellite
  Infrared

- **Image sequences (2D + time)**

  Video
  Ultrasound
  Satellite

- **Volumes (3D)**

  Magnetic Resonance (MR)
  Computed Tomography (CT)
  Positron Emission Tomography (PET)

- **Volume sequences (3D + time)**

  Magnetic Resonance (MR)

1.1 Local orientation

A basis for the approach taken in this presentation is the fact that a major part of the information needed for image interpretation is contained in the directions of local gradients. Equally important is the local simplicity hypothesis [2, 9, 10]. 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. In summary, local orientation is an important and, for most neighborhoods, well defined local feature of multi-dimensional data. To be well defined means, in this case, that the corresponding local signal is linear in the 2D case, and it is planar in the 3D case, referred to as *simple* neighborhoods. Furthermore, local orientation plays an important role in the mammalian visual system, as has been demonstrated by the physiological findings of Hubel and Wiesel and others [4], as discussed in chapter 2 of [3].

1.2 Orientation and local Fourier transforms

To see how local orientation analysis can be attained it is useful to study the behavior of local Fourier transforms (see chapter 4 of [3]). If the image locally can be approximated as simple, i.e.constant along parallel lines or planes, this is indicated in the Fourier domain by the fact that most of the energy is concentrated in a narrow sector oriented at the same angle as the local image gradients. The less the variation of local orientation, the narrower is the sector.

The distribution of energy along the sector, in the radial direction, reflects the frequency properties of the neighborhood's variation in the gradient direction. This distribution thus becomes analogous to that of a one-dimensional Fourier transform.

From the above it can be concluded that a description of a neighborhood in terms of local orientation and local frequency content can be obtained by partitioning the local Fourier transform and studying the energy contribution in the different parts, see Figure 1. The following sections present a method for estimation of local orientation based on this observation.

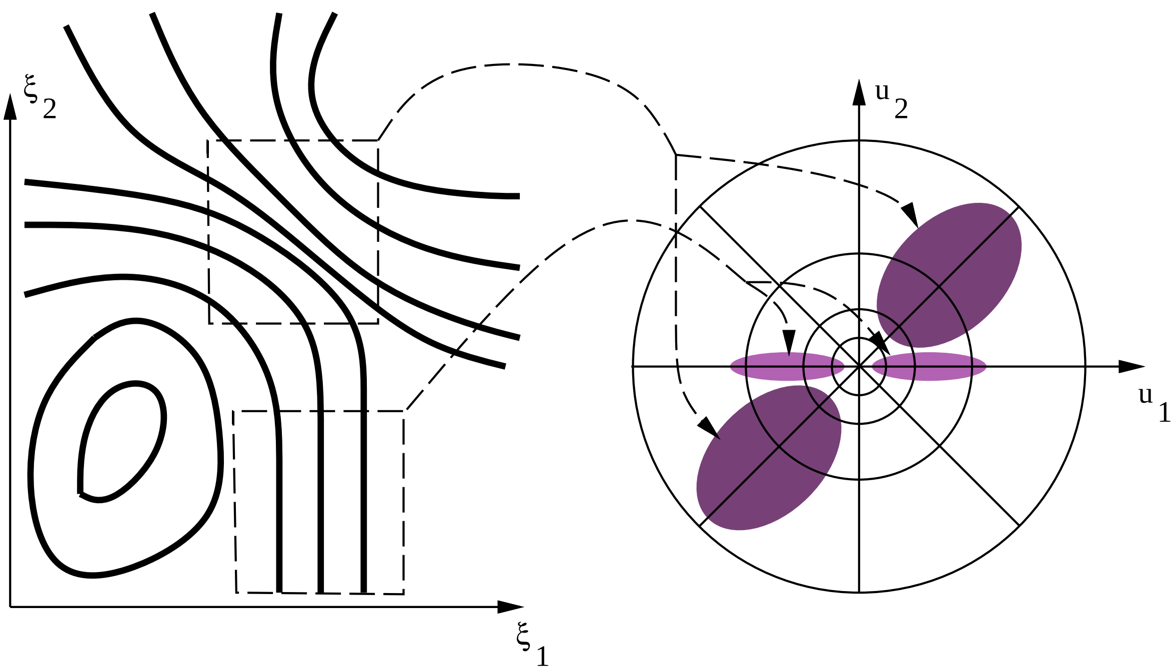


Figure 1: Two neighborhoods and the corresponding energy contributions in the Fourier domain.

2 Tensors - A short introduction

It is obvious that processing of higher-dimensional data sets puts high demands on computer power and storage capacity. Perhaps less obvious is that increasing the dimensionality of the data also has profound implications for its analysis. For example, in the analysis of data with a dimensionality higher than two it turns out that using scalars and vectors is no longer always convenient [6]. An important extension made here is the use of a generalization of the vector concept - the *tensor*.

The need for using tensors is motivated in the same way as for many physical quantities of complicated nature that cannot naturally be described or represented by scalars or vectors. Examples are the stress at a point in a solid body due to internal forces, the deformation of an arbitrary infinitesimal element of volume of an elastic body, and the moments of inertia and conductivity in anisotropic materials. These quantities can be described and represented adequately only by the more sophisticated mathematical entities called tensors. In fact, scalars and vectors belong to this family of elements. Thus, scalars and vectors are special cases of tensors.

Associated with a tensor is the order of that tensor. The order can be thought of as the complexity of the entity which the tensor represents, e.g., a zeroth order tensor is a scalar, a first order tensor is a vector and a second order tensor can be thought of as a linear mapping on a vector space. In this presentation, second order tensors are denoted by bold upper case letters, e.g., **T**, vectors by bold lower case letters, e.g., **x**, and scalars, both real and complex, by lower case italics, e.g., x.

The following sections contain a short introduction to tensors. However, the rather lengthy and theoretical presentation that is needed to give the full flavor of this rather general concept is outside the scope of this package. In particular, dual spaces and their relation to covariant and contra-variant tensors, as well as the metric tensor, are completely left out, and only one case of second order tensors is treated. Furthermore, only real vector spaces are considered in this presentation.

2.1 What are tensors?

Let us consider a distance in space, e.g., your height, and ask the question how long is this distance, i.e., what is your height? One answer could be 180 cm, another 72 inches, a third 6 feet. The point to be made is that the distance is the same but depending on the reference system, i.e., the unit of length, the answer is different. Your height is an example of a tensor, an object or phenomena which exists independent of any reference system that may be used for assigning a value to it. The tensor is the same, but since it is possible to choose the reference system quite arbitrary, the description of the tensor in terms of a value changes with the reference system.

First order tensors correspond to the basic concept of a vector, being an element of a vector space, and without relation to a reference system. If we wanted to use a reference system it would be a basis which defines a coordinate system of the vector space. Given such a basis, any vector can be described in terms of its coordinates relative to the basis. But, again, for one and the same vector this description changes if the basis is changed. Therefore, it is important to make the distinction between a vector and its coordinates, the vector is invariant relative to the choice of reference system, hence it is a first order tensor, whereas its coordinates varies with the reference system. The set of coordinates is a *description* of the vector, but it is not the same as the vector.

Second order tensors correspond to linear mappings on a vector space (of first order tensors), i.e., they map a vector of a vector space to a vector of the same vector space in a linear way. As an example, consider a line in the vector space and the linear mapping which rotates any vector around the line by some angle. This linear mapping is completely described by the previous sentence, and this is done without any reference to a coordinate system. If we want to know how the this particular mapping changes the coordinates of a vector relative to some particular basis, however, the mapping is conveniently described as a matrix whose action on the coordinates gives the coordinates of the resulting vector. A careful examination of the matrix reveals that the columns of the matrix contain the coordinates of the basis vectors *after* rotation. Hence, when choosing a different basis, the general effect is that the matrix which describes this particular mapping changes. To summarize, a matrix is a description of a second order tensor relative to a particular basis, whereas the tensor itself is invariant to any particular choice of basis.

2.1.1 Tensors and orientation

The same discussion applies to orientation. A linear structure in an image, e.g., a line or an edge, has an orientation regardless of which coordinate system is chosen to describe *which* orientation it is. For example, it is possible to consider two different edges and see if they have similar orientation or not, or to consider a curve and find that the local orientation along the curve changes in one way or another. All this can be done without actually assigning a value to the orientation. To find the value, however, a coordinate system is needed but the value assigned to the orientation of one and the same linear structure changes if the coordinate system changes.

As is shown in the following, orientation can be given a representation in terms of a second order tensor, which in addition is symmetric. Given a coordinate system, this second order tensor can be described in terms of a matrix, so the orientation tensor is in practice realized as a symmetric matrix. It is important to know, however, that this matrix is defined both by the orientation which it describes and the coordinate system being used. Since the latter is quite arbitrary chosen, this means that the matrix also has a flavor of arbitrariness in the sense that if we change to another coordinate system, the matrix changes even though the tensor is the same. This is the main motivation why second order tensors and matrices should not be confused, even though from a practical point of view, when a particular coordinate system is established, they are very much related.

To make the following presentation less abstract, the distinction between second order tensors and matrices is not emphasized in any particular way. This can be done by assuming that there exists some suitably chosen basis for the vector space, and, furthermore, that the vector space is equipped with a scalar product such that the basis is orthonormal. Most of the properties and results which are presented then follow more or less immediately from the corresponding statements about vectors and matrices made in linear algebra. In accordance to this, it is assumed that to each vector **x** there is a coordinate vector whose elements are denoted x_i, and to each tensor **T** there is a matrix whose elements are denoted T_{ij}, such that the result of applying the mapping **T** on **x** is the coordinate vector given by

$$\sum_{j=1}^{n} T_{ij}\, x_j \tag{1}$$

To make the presentation more compact, however, **x** is used also to denote the corresponding coordinate column vector, and **T** denotes the corresponding matrix. The resulting coordinate vector is then given by the matrix expression **T x**.

2.2 Tensors and vectors

Given the vectors (first order tensors) which constitute a vector space V, it is easy to see that the second order tensors also form a vector space. The sum of two linear mappings $\mathbf{T}$ and $\mathbf{U}$ is a new linear mapping $\mathbf{T} + \mathbf{U}$, which maps in the following way

$$[\mathbf{T} + \mathbf{U}]\,\mathbf{x} = \mathbf{T}\,\mathbf{x} + \mathbf{U}\,\mathbf{x} \tag{2}$$

for arbitrary $\mathbf{x} \in V$. Multiplying a linear mapping $\mathbf{T}$ by a real number a gives a linear mapping $a\,\mathbf{T}$, which maps in the following way

$$[a\,\mathbf{T}]\,\mathbf{x} = a\,(\mathbf{T}\,\mathbf{x}) \tag{3}$$

Furthermore, this vector addition and scalar multiplication comply with the usual requirements which are needed to make the set of second order tensors form a vector space. The space of second order tensors is here denoted V^2, the tensor product space of V with itself.

In the following, the word *vector* is used to represent a first order tensor, i.e., an element of the vector space V. The word *tensor* is used for second order tensors relative to V, i.e., elements of the vector space V^2. If the dimensionality of V is n, the dimensionality of V^2 is n^2.

2.3 Scalar products and norms

Given a vector space V (of first order tensors) a scalar product can be defined on that space. The scalar product between vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted $\langle \mathbf{x} \mid \mathbf{y} \rangle$, and it has the following properties

$$\begin{aligned} \langle \cdot \mid \cdot \rangle \; &: \; V \times V \rightarrow \mathbb{R} \\ \langle \mathbf{x} + \mathbf{y} \mid \mathbf{z} \rangle &= \langle \mathbf{x} \mid \mathbf{z} \rangle + \langle \mathbf{y} \mid \mathbf{z} \rangle \\ \langle a\,\mathbf{x} \mid \mathbf{y} \rangle &= a \langle \mathbf{x} \mid \mathbf{y} \rangle \\ \langle \mathbf{x} \mid \mathbf{y} \rangle &= \langle \mathbf{y} \mid \mathbf{x} \rangle \\ \langle \mathbf{x} \mid \mathbf{x} \rangle &\geq 0, \quad \text{with equality iff } \mathbf{x} = \mathbf{0} \end{aligned} \tag{4}$$

where $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z} \in V$ and a is a real number. These five properties together imply that the scalar product is a positive definite quadratic form on V.

In most cases, the basis used for V is orthonormal (ON) with respect to the scalar product being used. The consequence of this arrangement is that the scalar product between two vectors can be computed as the inner product of their corresponding coordinates, i.e.,

$$\langle \mathbf{x} \mid \mathbf{y} \rangle = \sum_{i=1}^{n} x_i \, y_i = \mathbf{x}^T \mathbf{y} \tag{5}$$

where x_i and y_i are the coordinates of $\mathbf{x}$ and $\mathbf{y}$, respectively, and $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$.

The scalar product induces a norm on V in the following way

$$\|\mathbf{x}\|^2 = \langle\, \mathbf{x} \mid \mathbf{x}\, \rangle \tag{6}$$

In terms of coordinates, relative to an ON-basis (where ON is relative to the scalar product), the norm can also be written as

$$\|\mathbf{x}\|^2 = \sum_{i=1}^{n} x_i^2 = \mathbf{x}^T\mathbf{x} \tag{7}$$

where x_i are the coordinates of **x**.

2.4 Eigenvalues and eigenvectors

Given a linear mapping **T**, any vector **e** which has the following property

$$\mathbf{T}\,\mathbf{e} = \lambda\,\mathbf{e} \tag{8}$$

where λ is a real number, is an *eigenvector* of **T**. The real number λ is the *eigenvalue* of **e**. The zero vector is normally not considered as an eigenvector, even though (or rather because) it satisfies Equation (8) for any λ. A thorough investigation on the existence of eigenvalues and, in particular, of eigenvectors for an arbitrary **T** is outside the scope of this presentation. The following discussion, therefore, relates only to the cases where there exist at least one eigenvector to each eigenvalue.

The first thing to note about eigenvalues and eigenvectors is the fact that if **e** satisfies Equation (8) for some λ, then so does $a\,\mathbf{e}$ for any real number a. Hence, to each eigenvalue there is a linear subspace, at least one-dimensional, of corresponding eigenvectors. Furthermore, if $\mathbf{e}_1$ and $\mathbf{e}_2$ are two linearly independent eigenvectors of **T**, both with eigenvalue λ, then any linear combination of the two is too an eigenvector with eigenvalue λ. Hence, each eigenvalue λ defines a subspace of V, which is at least one-dimensional, such that the subspace contains exactly those vectors which are eigenvectors of **T** and have eigenvalue λ.

An illustration of the above discussion is provided by **I**, the identity tensor. This tensor is such that

$$\mathbf{I}\,\mathbf{e} = \mathbf{e} \tag{9}$$

for any vector **e**. Thus, any vector is an eigenvector of **I**, and the corresponding eigenvalue is always unity. Therefore, the subspace defined by this single eigenvalue is the entirety of V.

It is noteworthy that the eigenvectors, and their corresponding eigenvalues, are invariant properties of the tensor itself. This means that if **e** is an eigenvector of **T**, with eigenvalue λ, this is the case regardless of which coordinate system is chosen. Of course, the coordinates of **e** are different for different bases, as mentioned above. The eigenvalue, however, does not depend on the choice of basis.

The eigenvalues of **T** are given by the roots of the *secular equation*

$$\det(\mathbf{T} - \lambda \mathbf{I}) = 0 \tag{10}$$

where **I** is the identity mapping. This is an n-th order polynomial in λ, and from that point of view it has n roots, i.e., **T** has n eigenvalues. On the other hand, it may be the case that two or more eigenvalues are equal. In fact, it can be shown that the multiplicity of a root λ, i.e., the number of roots which are equal to λ, is the same as the dimensionality of the corresponding subspace of eigenvectors. It should be noted that even though the eigenvalues of a particular **T** are not distinct, they are always treated as n individual scalar values.

2.5 Symmetric tensors

A tensor $\mathbf{T}$ is *symmetric* if

$$\langle \, \mathbf{T}\,\mathbf{x} \mid \mathbf{y} \, \rangle = \langle \, \mathbf{x} \mid \mathbf{T}\,\mathbf{y} \, \rangle \tag{11}$$

for all $\mathbf{x}, \mathbf{y} \in V$. From a practical point of view, this implies that the matrix representation of $\mathbf{T}$ has the following property

$$T_{ij} = T_{ji} \tag{12}$$

A well known result from linear algebra is the fact that for any symmetric tensor $\mathbf{T}$ it is possible to find an ON-basis of eigenvectors, $\{\hat{\mathbf{e}}_i\}$, and furthermore, it is possible to write $\mathbf{T}$ as a linear combination of outer products of these eigenvector with themselves, according to

$$\mathbf{T} = \sum_{i=1}^{n} \lambda_i \, \hat{\mathbf{e}}_i \, \hat{\mathbf{e}}_i^T \tag{13}$$

where λ_i is the eigenvalue corresponding to $\hat{\mathbf{e}}_i$. In fact, *any* ON-basis of eigenvectors $\{\hat{\mathbf{e}}_k\}$ validates Equation (13), when λ_k are the corresponding eigenvalues. Furthermore, the opposite is also true: if $\mathbf{T}$ can be decomposed in this way, where $\{\hat{\mathbf{e}}_k\}$ is an ON-basis, these vectors are eigenvectors of $\mathbf{T}$ and $\{\lambda_k\}$ are corresponding eigenvalues.

In the following, only symmetric tensors are considered. These tensors form a subspace of V^2 denoted $Sym(V^2)$, and any tensor of this space can thus be decomposed according to Equation (13). The dimensionality of $Sym(V^2)$ is $\frac{n(n+1)}{2}$. The symmetric property of **T** ensures that all its eigenvalues are real, which means that they can be ordered. The eigenvalues are labeled $\lambda_1, \lambda_2, \ldots, \lambda_n$, using the convention

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \tag{14}$$

In the following presentation, a terminology is used which implies that to each eigenvalue λ_K can be associated a normalized eigenvector $\hat{\mathbf{e}}_k$. This is practical, but according to the previous discussion, however, it is not true in a strict sense. The problem lies in the fact that for a given λ_k, being one of the *n* roots of the secular equation, the choice of $\hat{\mathbf{e}}_k$ is not unique. On the other hand, it turns out that all results presented are invariant to any particular choice that can be made, as long as the resulting set of eigenvectors $\{\hat{\mathbf{e}}_k\}$ is an ON-basis.

The rank of **T** is defined as the number of non-zero eigenvalues. This means that a rank one tensor **T** can always be written as

$$\mathbf{T} = \lambda\, \hat{\mathbf{e}}\, \hat{\mathbf{e}}^T \tag{15}$$

where λ is the single non-zero eigenvalue, and $\hat{\mathbf{e}}$ is the corresponding normalized eigenvector.

2.6 Trace

For any tensor $\mathbf{T} \in Sym(V^2)$, the trace of $\mathbf{T}$, denoted $\mathrm{tr}(\mathbf{T})$, is defined as

$$\mathrm{tr}(\mathbf{T}) = \sum_i \lambda_i \tag{16}$$

where $\{\lambda_i\}$ is the set of all eigenvalues of $\mathbf{T}$. According to the previous section, the eigenvalues of $\mathbf{T}$ are invariant to the choice of coordinate system. This must then be true also for the trace of $\mathbf{T}$, it is an intrinsic property of the tensor itself.

Given a matrix representation of $\mathbf{T}$, the trace is given by

$$\mathrm{tr}(\mathbf{T}) = \sum_i T_{ii} \tag{17}$$

which is a summation of the diagonal element of the corresponding matrix $\mathbf{T}$.

2.7 Scalar products and norms for tensors

Not only does V induce a vector space structure for V^2. If a scalar product is defined on V, it induces a scalar product, and eventually a norm, on $Sym(V^2)$. To see this, let $\langle \cdot \mid \cdot \rangle$ be a scalar product on V, and let **T** and **U** be symmetric tensors, i.e., they can be written as

$$\mathbf{T} = \sum_{i=1}^{n} \lambda_i \, \hat{\mathbf{e}}_i \, \hat{\mathbf{e}}_i^T \tag{18}$$

$$\mathbf{U} = \sum_{j=1}^{n} \sigma_j \, \hat{\mathbf{f}}_j \, \hat{\mathbf{f}}_j^T \tag{19}$$

The scalar product between **T** and **U**, $\langle \mathbf{T} \mid \mathbf{U} \rangle$, is then given by

$$\langle \mathbf{T} \mid \mathbf{U} \rangle = \sum_{ij} \lambda_i \, \sigma_j \, \langle \hat{\mathbf{e}}_i \mid \hat{\mathbf{f}}_j \rangle^2 \tag{20}$$

A proof that this is a scalar product, i.e., that it complies with the five properties stated in Equation (4), is demonstrated in Exercise 1. In terms of the matrix description of **T** and **U**, their scalar product can also be written

$$\langle \mathbf{T} \mid \mathbf{U} \rangle = \sum_{ij} T_{ij} \, U_{ij} \tag{21}$$

There is also third way of describing the scalar product between tensors. A careful examination of Equation (21) leads to the following expression

$$\langle \mathbf{T} \mid \mathbf{U} \rangle = \mathrm{tr}(\mathbf{T}^T \mathbf{U}) \tag{22}$$

Proofs of Equations (21) and (22) are presented in Exercise 2.

Note that the same notation for scalar products between vectors and between tensors are being used, i.e., $\langle \cdot \mid \cdot \rangle$. This is convenient, but not strictly correct, since these two scalar products are defined on two different vector spaces.

Since a scalar product between tensors is established, a norm can also be defined, according to

$$||\mathbf{T}||^2 = \langle \mathbf{T} \mid \mathbf{T} \rangle \tag{23}$$

In accordance with the above three descriptions of the tensor scalar product, this norm can also be expressed as

$$||\mathbf{T}||^2 = \sum_i \lambda_i^2 \tag{24}$$

$$||\mathbf{T}||^2 = \sum_{ij} T_{ij}^2 \tag{25}$$

$$||\mathbf{T}||^2 = \mathrm{tr}(\mathbf{T}^T\mathbf{T}) \tag{26}$$

This tensor norm is referred to as the *Frobenius norm*.

2.8 Bases and dual bases

Given a basis $\{\mathbf{e}_i\}$ for V, i.e., a set of linearly independent vectors which span V, any $\mathbf{x} \in V$ can be written as a linear combination of the basis vectors:

$$\mathbf{x} = \sum_{i=1}^{n} x_i \, \mathbf{e}_i \tag{27}$$

where x_i are the corresponding coordinates of $\mathbf{x}$ relative to this particular basis. Given the scalar product for V, there is a unique set of vectors $\{\tilde{\mathbf{e}}_j\}$, the *dual basis* of $\{\mathbf{e}_i\}$, which is characterized by

$$\langle \, \mathbf{e}_i \mid \tilde{\mathbf{e}}_j \, \rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{28}$$

In Exercise 3 it is shown that the coordinates of $\mathbf{x}$ relative to the basis $\{\mathbf{e}_k\}$ are given by

$$x_i = \langle \, \mathbf{x} \mid \tilde{\mathbf{e}}_i \, \rangle \tag{29}$$

Hence, the recipe for computing coordinates of a vector relative to a basis is first to find the corresponding dual basis, and then compute the scalar product between the vector and the dual basis vectors. Normally, the basis for V is an ON-basis which implies that it is its own dual basis.

The same discussion relates also to $Sym(V^2)$, the vector space of symmetric second order tensors. Any set of such tensors $\{\mathbf{B}_i\}$ which are linearly independent and which span

$Sym(V^2)$ is a basis for that space. An arbitrary $\mathbf{T} \in Sym(V^2)$ can then be written as

$$\mathbf{T} = \sum_{i}^{\frac{n(n+1)}{2}} t_i \, \mathbf{B}_i \tag{30}$$

where t_i are the corresponding coordinates of $\mathbf{T}$ relative to this basis. The scalar product defined for $Sym(V^2)$, Equations (20), (21) and (22), then establishes the dual basis of $\{\mathbf{B}_i\}$, denoted $\{\tilde{\mathbf{B}}_i\}$, according to

$$\langle\, \mathbf{B}_i \mid \tilde{\mathbf{B}}_j \,\rangle = \delta_{ij} \tag{31}$$

and the coordinates t_i are given by

$$t_i = \langle\, \mathbf{T} \mid \tilde{\mathbf{B}}_i \,\rangle \tag{32}$$

which implies that

$$\mathbf{T} = \sum_{i}^{\frac{n(n+1)}{2}} \langle\, \mathbf{T} \mid \tilde{\mathbf{B}}_i \,\rangle \, \mathbf{B}_i \tag{33}$$

Since the dual basis of $\{\tilde{\mathbf{B}}_i\}$ is the original basis, it is possible to express $\mathbf{T}$ also as

$$\mathbf{T} = \sum_{i}^{\frac{n(n+1)}{2}} \langle\, \mathbf{T} \mid \mathbf{B}_i \,\rangle \, \tilde{\mathbf{B}}_i \tag{34}$$

While Equation (31) gives a description of the relation between a basis and its dual basis, it does not explain how the dual basis is found. In this case, where the vector space is of finite dimensionality, the dual basis can always be computed in the following way. Define the symmetric matrix **P** as

$$P_{kl} = \langle \mathbf{B}_k \mid \mathbf{B}_l \rangle \tag{35}$$

Then,

$$\tilde{\mathbf{B}}_l = \sum_k \mathbf{B}_k \, Q_{kl} \tag{36}$$

where $\mathbf{Q}^T = \mathbf{P}^{-1}$. Hence, first compute the scalar product between all pairs of basis tensors, giving the matrix **P**. Then, the inverse matrix **Q** is used in a linear combination of the original basis tensors to give the sought for dual basis. A proof of this statement is presented in Exercise 4.

2.9 Frames and dual frames

In the following, we have reasons to consider also the case when a tensor $\mathbf{T}$ is written as a linear combination of a finite set of tensors, $\{\mathbf{B}_k\}$, which span the vector space of tensors, but are not linearly independent. Such a set is here referred to as a *frame*, even though the proper definition of a frame includes bases as a special case. In the same way as for a basis, we are interested in how the corresponding coefficients of the linear combination, Equation (30) (but with summation made over a larger set), can be found. In the case of a frame, however, it is easily realized that there is no unique set of coefficients (corresponding to coordinates in the case of a basis) which does the job. Similar to the case of a basis, though, a dual frame $\{\tilde{\mathbf{B}}_k\}$, can be defined such that the coefficients

$$t_k = \langle\, \mathbf{T} \mid \tilde{\mathbf{B}}_k \,\rangle \tag{37}$$

can be used in Equation (30). The derivation of the dual frame is outside the scope of this presentation, and instead a straightforward recipe is given.

Each tensor in the frame, represented as a symmetric matrix $\mathbf{B}_k$, must first be reshaped to a row vector. This has to be done so that each such row vector has the same norm as the tensor, and so that the resulting set of row vectors span a vector space of the same dimensionality as $Sym(V^2)$. This is done by placing the n diagonal elements $\mathbf{B}_k$, and $\frac{n(n-1)}{2}$ off-diagonal elements (either in the upper or lower off-diagonal), in some fixed order along the row vector. However, to preserve the norm, each off-diagonal element must by multiplied by $\sqrt{2}$. The reshaping of an $n \times n$ tensor thus results in a $1 \times \frac{n(n+1)}{2}$ row vector.

For each frame tensor $\mathbf{B}_k$, the above reshaping gives a row vector. Let these vectors define the rows of a matrix $\mathbf{F}$, referred to as the *frame operator*. The dual frame operator $\tilde{\mathbf{F}}$ is then given by

$$\tilde{\mathbf{F}} = \mathbf{F}\,(\mathbf{F}^T\mathbf{F})^{-1} \tag{38}$$

The rows of $\tilde{\mathbf{F}}$ then contains the corresponding dual tensors, reshaped according to the above procedure. To obtain the dual frame tensor $\tilde{\mathbf{B}}_k$, therefore, it is only needed to apply the inverse reshaping procedure on the k-th row of $\tilde{\mathbf{F}}$, including dividing each off-diagonal element by $\sqrt{2}$. Of course, the dual tensors also are symmetric, so each off-diagonal element taken from row k of $\tilde{\mathbf{F}}$ must be copied into both the upper and lower off-diagonals of $\tilde{\mathbf{B}}_k$.

3 Representing orientation

First of all, it should be emphasized that orientation is a local property of a signal. Almost no signal does in practice exhibit *one* single orientation, instead normal images are scattered with various structures, each having its own characteristics in terms of orientation and type, e.g., line or edge. Note, however, that these characteristics are not uncorrelated, two neighborhoods very close to each other normally have similar (but not necessary equal) characteristics.

In order to arrive at a useful representation of local orientation for a signal, it must first be clarified for which local neighborhoods the notion of orientation is well-defined. For example, a neighborhood that contains a single edge or line has a well-defined orientation, whereas neighborhoods that are constant, contains substantial amounts of noise, or several edges or lines, do not have a well-defined orientation in the sense used here.

For a signal of arbitrary dimensionality n, the model chosen for a local neighborhood which has well-defined orientation is that it is *simple*. A simple neighborhood is represented as a function of n variables, f, that can be expressed as

$$f(\xi) = g(\langle\, \xi \mid \mathbf{x} \,\rangle) = g(\xi^T \mathbf{x}) \qquad (39)$$

where ξ is the local spatial coordinate vector, g is a non-constant function of one variable, and $\mathbf{x}$ is a constant vector. According to this definition, f is constant along any vector that is perpendicular to $\mathbf{x}$, or, vice versa, f has its ”maximal variation” in the direction of $\mathbf{x}$. For example, in 2D f is constant along all lines that are perpendicular to $\mathbf{x}$, and all these lines are parallel. In 3D, f is constant on any plane that is perpendicular to $\mathbf{x}$, and all these planes are parallel. In general, the dimensionality of the hyper-planes on which f is constant is one less than the dimensionality of the signal. It should be noted that the concept of a simple neighborhood does not depend on the range of f and g. The range can be anything from real or complex numbers to vectors or tensors.

Obviously, the characteristic properties of a simple neighborhood f, defined by Equation (39), is given by the one-variable function g and the vector $\mathbf{x}$. However, the choice of g and $\mathbf{x}$ are not entirely unique. If the equation is true for a particular $\mathbf{x}$ and g, then it is true also for $a\,\mathbf{x}$, where $a \neq 0$, by choosing g' defined as

$$g'(x) = g(\frac{x}{a}) \tag{40}$$

instead of g. To make the situation less ambiguous, $\mathbf{x}$ can always be normalized, i.e., $a = ||\mathbf{x}||^{-1}$. In the following it is assumed that g and $\hat{\mathbf{x}}$ are chosen so that

$$f(\xi) = g(\langle\, \xi \mid \hat{\mathbf{x}}\, \rangle) = g(\xi^T \hat{\mathbf{x}}) \tag{41}$$

where $||\hat{\mathbf{x}}|| = 1$. Still, however, the sign of $\hat{\mathbf{x}}$ is undetermined.

Having restricted the signals for which orientation is well-defined to simple signals, it should be noted that there is a distinction between orientation and direction. A line has an unambiguous orientation, and no direction, whereas a (non-zero) vector has a unique direction and no orientation. The same goes for a plane in 3D, it has a unique orientation and no particular direction. Its orientation is given, e.g., by a normal vector, but the sign of this vector, i.e., its direction, is indeterminable.

With the above considerations in mind the approach taken here is to define the orientation of a simple neighborhood as being defined entirely by $\hat{\mathbf{x}}$, but independent of its sign, and it is also independent of g. Below, this latter property is discussed more formally, but the consequence is that the representation chosen is a true orientation representation, and not a representation of direction. As mentioned above, however, the direction of, e.g., a slope may be of importance, which means that the information on orientation sometimes has to be complemented with a description of direction. This can be done in terms of a local phase descriptor. In fact, the phase is a more general concept than direction, and can be used, for example, to determine whether a two-dimensional line is darker than the background or vice versa. It can even give an exact description of *where* an edge or a line is located. See chapter 7 of [3].

3.1 Requirements of the orientation representation

Assuming that the representation chosen for orientation, here denoted **T**, is an element of a normed vector space, it can be normalized according to

$$\hat{\mathbf{T}} = \frac{\mathbf{T}}{||\mathbf{T}||} \tag{42}$$

so that

$$\mathbf{T} = ||\mathbf{T}||\,\hat{\mathbf{T}} \tag{43}$$

Regardless of what type of object **T** really is, be it a tensor or something else, there are at least three requirements which it should meet.

3.1.1 The invariance requirement

It is evident that the orientation of the neighborhood is the same for all possible g in Equation (41), at least as long as g is not constant. In other words the entity which represents the orientation must be invariant to g, something which can be expressed as

$$\mathbf{T} \text{ must not depend on } g \tag{44}$$

In fact, this requirement is very strong and can not be met in practice. A somewhat weaker, but more practical requirement is to allow the norm of $\mathbf{T}$, i.e., $||\mathbf{T}||$, and *only* the norm, to change with g. This can be expressed as

$$\hat{\mathbf{T}} \text{ must not depend on } g \tag{45}$$

Note that the stronger requirement is a special case of the weaker. In the following section, it is illustrated how the invariance requirement is implemented for various types of variations in g.

3.1.2 The equivariance requirement

If the orientation of the local neighborhood changes by a small amount, this should always result in a small change in the representation $\mathbf{T}$. A more formal description of this situation is to say that there must be a continuous mapping from $\hat{\mathbf{x}}$ to $\mathbf{T}$. Furthermore, if making a small change in $\hat{\mathbf{x}}$, resulting in a small change also in $\mathbf{T}$, the norm of the latter should be invariant to $\hat{\mathbf{x}}$. This implies that the variation in $\mathbf{T}$ caused by a variation in $\hat{\mathbf{x}}$ is *isotropic*.

These two ideas can be formalized by describing any small change in $\hat{\mathbf{x}}$ as the result of adding to it a fraction of the vector $\hat{\mathbf{v}}$, where $\hat{\mathbf{v}}$ is perpendicular to $\hat{\mathbf{x}}$. The result is the vector $\mathbf{x}_1$,

$$\mathbf{x}_1 = \hat{\mathbf{x}} + \varepsilon\, \mathbf{v} \tag{46}$$

If $\mathbf{T}$ is the representation for the orientation given by $\hat{\mathbf{x}}$, and $\mathbf{T}(\varepsilon)$ is the representation given by $\mathbf{x}_1$, define

$$\frac{d\mathbf{T}}{d\varepsilon} = \lim_{\varepsilon \to 0} \frac{\mathbf{T}(\varepsilon) - \mathbf{T}}{\varepsilon} \tag{47}$$

The equivariance requirement can then be written as

$$\left\| \frac{d\mathbf{T}}{d\varepsilon} \right\| \text{ must not depend on } \hat{\mathbf{x}} \text{ or } \mathbf{y} \tag{48}$$

3.1.3 The uniqueness requirement

In order to be a representation of orientation, it is required that $\mathbf{T}$ is a one-to-one mapping from the orientation to the representation domain. Formally, such a mapping is referred to as *bijective*, and the consequence of this property is that to each orientation there is a unique representation $\mathbf{T}$, and vice versa, to each $\mathbf{T}$ in the representation domain there is a unique orientation.

Since the orientation of a simple neighborhood depends only on $\hat{\mathbf{x}}$, but not on the sign of this vector, the uniqueness requirement implies that

$$\mathbf{T}(\hat{\mathbf{x}}) = \mathbf{T}(\hat{\mathbf{y}}) \qquad \Leftrightarrow \qquad \hat{\mathbf{y}} = \pm\hat{\mathbf{x}} \tag{49}$$

3.1.4 Implications

The implications of the three requirements can easily be explained by studying the 3D signals in Figure 2. The invariance requirement implies that $\mathbf{T}$ should be the same for neighborhoods f_a and f_b, as well as the same for f_c and f_d, even though g is different in the two cases. Equivariance implies that in moving from f_a to f_c, or from f_b to f_d, the infinitesimal changes of **T** which integrate to the total change, should everywhere be proportionally to the change in local orientation. The uniqueness requirement implies that there can only be one unique representation for the orientation of, e.g., f_a and f_b, and this representation uniquely defines the orientation of the two neighborhoods.

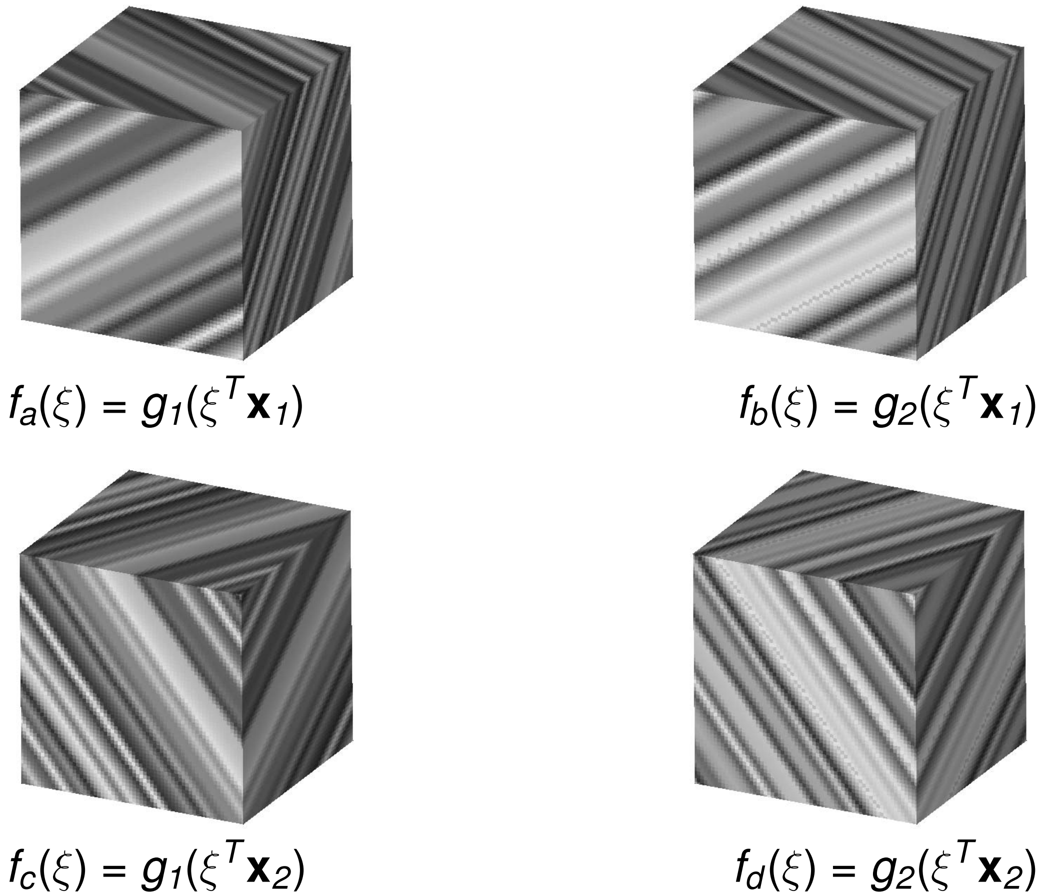


Figure 2: Four simple neighborhoods in 3D. The neighborhoods are constructed using two different signal functions (g_1 and g_2) and two different signal orienting vectors ($\mathbf{x}_1$ and $\mathbf{x}_2$).

3.2 The orientation tensor

A representation for orientation that meets the above criteria is a second order tensor given by:

$$\mathbf{T} \equiv A\,\hat{\mathbf{x}}\,\hat{\mathbf{x}}^T \tag{50}$$

where A is an arbitrary positive number which does not depend on $\hat{\mathbf{x}}$. Note that $\hat{\mathbf{x}}$ is an eigenvector of $\mathbf{T}$, with corresponding eigenvalue A, and that this is the only non-zero eigenvalue, so $\mathbf{T}$ is a tensor of rank one.

You may ask: why not use something simpler than a second order tensor? At least in 2D, a simple scalar which represents the angle of $\hat{\mathbf{x}}$ should do. Apart from not being useful in higher dimensions, where several angles would be needed, this representation does not meet the equivariance requirement. For some angle, there has to be a discontinuity so that a small change in $\hat{\mathbf{x}}$ corresponds to a very large change in the scalar. It is also possible to use just the first order tensor $\mathbf{x}$, or $\hat{\mathbf{x}}$, as a representation of the orientation. As already mentioned, however, $\mathbf{x}$ is not unique for a simple neighborhood, and the vector $-\mathbf{x}$ is then also a valid representation. The only way to overcome this ambiguity is to restrict the domain of $\mathbf{x}$ to some directional interval which, in turn, introduces discontinuities in the same way as for scalars.

A second order tensor avoids both the ambiguity problem, changing the sign of $\mathbf{x}$ does not change $\mathbf{T}$, and it gives a continuous representation. It also meets the invariance and equivariance requirements, as is shown below.

As a consequence of using a tensor for orientation representation, it is not necessary to make a distinction between the different dimensions of the signal when the tensor is formed. For example, local orientation of a 3D signal where two dimensions represent space and the third time is described in the same way as for a 3D signal where all three dimensions represent space. However, the interpretation of the information which the orientation tensor carries requires knowledge of which case it is, e.g., which dimension corresponds to time.

As mentioned in Section 2, given a suitable basis, $\hat{\mathbf{x}}$ is described as a column vector of n real numbers, and $\mathbf{T}$ is described as an $n \times n$ matrix. If the vector is

$$\hat{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{51}$$

then the tensor is

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1n} \\ T_{12} & T_{22} & \dots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{1n} & T_{2n} & \dots & T_{nn} \end{pmatrix} = A \begin{pmatrix} x_1^2 & x_1 x_2 & \dots & x_1 x_n \\ x_1 x_2 & x_2^2 & \dots & x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1 x_n & x_2 x_n & \dots & x_n^2 \end{pmatrix} \tag{52}$$

3.2.1 The norm

Calculating the norm of $\mathbf{T}$, Equation (24), gives:

$$||\mathbf{T}|| = A \tag{53}$$

Since $||\mathbf{T}||$ (or A) is independent of the orientation it represents, it can be used to represent another property. In Section 4 it is shown that A can be made to represent the local amplitude of the signal in a natural way.

3.2.2 Invariance

By its construction $\mathbf{T}$ is trivially invariant to the signal function g in the strong sense. However, as can be expected, making the actual orientation estimates invariant to g is by no means trivial. In practice, the weaker invariance requirement is used, which implies that $||\mathbf{T}||$, but not $\hat{\mathbf{T}}$, is dependent on g. A discussion of this topic appears in Section 4.

3.2.3 Equivariance

To show that the equivariance requirement is met by the suggested representation is fairly straightforward. Starting from

$$\mathbf{T} = A\,\hat{\mathbf{x}}\,\hat{\mathbf{x}}^T \tag{54}$$

Equation (46) gives

$$\mathbf{T}(\varepsilon) = A\,(\hat{\mathbf{x}} + \varepsilon\,\hat{\mathbf{v}})\,(\hat{\mathbf{x}} + \varepsilon\,\hat{\mathbf{v}})^T \tag{55}$$

and, finally, from Equation (47),

$$\frac{d\mathbf{T}}{d\varepsilon} = A\,(\hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T) \tag{56}$$

In Exercise 5 it is verified that $\hat{\mathbf{x}}+\hat{\mathbf{v}}$ and $\hat{\mathbf{x}}-\hat{\mathbf{v}}$ both are eigenvectors of the right hand side of Equation (56), with corresponding eigenvalues A and $-A$, respectively. Furthermore, these are the only non-zero eigenvalues. Hence, Equation (24) gives

$$\left\|\frac{d\mathbf{T}}{d\varepsilon}\right\|^2 = 2\,A^2 \tag{57}$$

which shows that the norm of the differential is independent of both $\hat{\mathbf{x}}$ and $\hat{\mathbf{v}}$, and therefore the equivariance requirement is met.

3.2.4 Exercises

Exercise 6 shows how the orientation tensor is described by numerical values in a matrix for a particular simple 2D neighborhood. Exercise 7 shows how to find the corresponding 2D local neighborhood given a particular orientation tensor in the form of a matrix.

3.3 The 2D orientation vector

In the special case of a 2D signal, e.g., an ordinary gray-scale image, it is in fact possible to use a vector for the representation of local orientation. This vector is neither $\mathbf{x}$ nor $\hat{\mathbf{x}}$, but rather something which rotates with twice the angular speed as either one of them. This representation of local 2D orientation was first introduced in [2].

In the 2D case, the vector $\hat{\mathbf{x}}$ has the form

$$\hat{\mathbf{x}} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \tag{58}$$

where α is measured relative to some suitably chosen coordinate system. This gives

$$\mathbf{T} = \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} = A \begin{pmatrix} \cos^2\alpha & \cos\alpha\sin\alpha \\ \cos\alpha\sin\alpha & \sin^2\alpha \end{pmatrix} \tag{59}$$

The orientation vector is then given by

$$\mathbf{z} = \begin{pmatrix} T_{11} - T_{22} \\ 2\,T_{12} \end{pmatrix} = A \begin{pmatrix} \cos^2\alpha - \sin^2\alpha \\ 2\,\cos\alpha\sin\alpha \end{pmatrix} = A \begin{pmatrix} \cos 2\alpha \\ \sin 2\alpha \end{pmatrix} \tag{60}$$

In some cases it is more convenient to describe $\mathbf{z}$ as a complex number instead of as a 2D vector. This complex number, z, is then given by

$$z = T_{11} - T_{22} + 2\,i\,T_{12} = A\,(\cos 2\alpha + i\sin 2\alpha) \tag{61}$$

Regardless of whether a 2D vector or a complex number is used, it is evident that it rotates with twice the angular speed relative to $\hat{\mathbf{x}}$. Hence, if $\hat{\mathbf{x}}$ rotates half a turn, corresponding to a change in the sign of the original $\hat{\mathbf{x}}$, $\mathbf{z}$ rotates a complete turn, returning to its original value. The following shows that the 2D orientation vector complies with the invariance and equivariance requirements.

3.3.1 The norm

The norm of $\mathbf{z}$ is given by

$$||\mathbf{z}||^2 = A^2(\cos^2\alpha + \sin^2\alpha) = A^2 \tag{62}$$

i.e., it is the same as the norm of $\mathbf{T}$.

3.3.2 Invariance

In the same manner as $\mathbf{T}$, $\mathbf{z}$ is trivially invariant to g.

3.3.3 Equivariance

Instead of constructing the rotation of $\hat{\mathbf{x}}$ by means of adding a perpendicular vector to it, the 2D case offers the angle α as a parameterization of the rotation. Hence,

$$\mathbf{z}(\alpha) = A \begin{pmatrix} \cos 2\alpha \\ \sin 2\alpha \end{pmatrix} \tag{63}$$

and

$$\frac{d\mathbf{z}}{d\alpha} = 2\,A \begin{pmatrix} -\sin 2\alpha \\ \cos 2\alpha \end{pmatrix} \tag{64}$$

which gives

$$\left\| \frac{d\mathbf{z}}{d\alpha} \right\| = 2\,A \tag{65}$$

Hence, the norm of the differential is a constant and the equivariance requirement is met.

4 Orientation estimation

Having found a suitable representation of orientation, it is then natural to ask the question:

Can the representation be implemented using measurements on actual image data, where lines or other structures are represented as local gray scale correlations?

It is shown here that by combining the outputs from polar separable quadrature filters, it is possible to produce a representation corresponding exactly to Equation (50). The exactness relies on the image data being locally *simple*, Equation (39), i.e., on the existence of a locally well-defined orientation. The case where the simplicity assumption does not hold is discussed in Section 6.

4.1 The signal and its Fourier transform

The analysis is restricted to *real* valued *simple* neighborhoods, i.e., neighborhoods f that can be expressed as

$$s(\xi) = g(\xi^T \hat{\mathbf{x}}) \tag{66}$$

where g is a real valued function of one variable, $\hat{\mathbf{x}}$ is a normalized vector.

The Fourier transform of s, denoted S, has the characteristic property of being concentrated to an impulse line which passes through the origin, with $\hat{\mathbf{x}}$ as a direction vector, and along that line S varies as G, the one-dimensional Fourier transform of g. See Exercise 8. Formally, this can be expressed as

$$S(\mathbf{u}) = (2\pi)^{n-1}\, G(\mathbf{u}^T \hat{\mathbf{x}})\, \delta_{\hat{\mathbf{x}}}^{\text{line}}(\mathbf{u}) \tag{67}$$

where $\mathbf{u}$ is the frequency coordinate, and $\delta_{\hat{\mathbf{x}}}^{\text{line}}(\mathbf{u})$ is the impulse line.

4.2 The quadrature filter concept

In order to realize the invariance requirement, the estimation procedure is designed using quadrature filters [10]. The quadrature filter concept forms a basis for minimizing the sensitivity to phase changes in the signal.

Independently of the dimensionality of the signal space, a quadrature filter can be defined as a filter that is zero over one half of the Fourier domain. More precisely, let F be the Fourier transform of the quadrature filter f, then

$$F(\mathbf{u}) = 0 \quad \text{if } \mathbf{u}^T\hat{\mathbf{n}} \leq 0 \tag{68}$$

where $\hat{\mathbf{n}}$ is the directing vector of the filter. Of course, this is only half of the story, the characteristics of F in the non-zero half of the Fourier domain are also important for this application.

An important property of a quadrature filter is the fact that both the filter itself and therefore also its output, in general, is complex. This is immediately realized by the fact that F is not Hermitian, and therefore f is not real. See chapter 4 of [3]. Hence, the filter output is in each point a complex number, here denoted q', and q is used to denote the magnitude of q', i.e., $q = |q'|$.

As an example of the phase invariant property of a quadrature filter, it can be mentioned that if g is a sinusoidal function, the argument of q' represents the local phase, whereas q is completely phase invariant. Consequently, if we consider a simple function $s(\xi)$, Equation (66), where

$$g(x) = A \sin(\omega x - \phi) \tag{69}$$

and apply a quadrature filter to this signal, then the filter output magnitude q is invariant to the position ξ. The following sections present an estimation procedure where the orientation representation is computed as a linear combination of fixed tensors and quadrature filter output magnitudes. Consequently, for a signal like this, the resulting orientation representation is indeed invariant to the local phase or position.

In practice, we have to limit the discussion to local regions of the signal which approximately can be modeled as simple functions. In some cases these can be assumed to have a sinusoidal characteristic, i.e.,

$$g(x) = e(x) \sin(\omega x - \phi) \tag{70}$$

corresponding to an envelop function e which describes the local amplitude times a modulating factor $\sin(\omega x - \phi)$. Provided that e varies sufficiently slow compared to the modulating frequency ω, we get $q(\xi) \approx e(\xi^T \hat{\mathbf{x}})$, and consequently that q has a relatively slow variation compated to the local frequency ω. This means also that the resulting representation varies relatively slow compared to the local frequency.

If, on the other hand, a locally simple signal does not has a sinusoidal characteristic, not even approximately, this implies that the representation may vary relatively fast over the corresponding local neighborhood. However, the estimation procedure which is presented here assures that the only variation in the representation over a locally simple neighborhood is in the norm of the representation. The estimated representation can always be approximated as a rank one symmetric tensor where the eigenvector corresponding to the largest eigenvalue is $\pm\hat{\mathbf{x}}$ provided that the corresponding local neighborhood is approximately a simple function.

4.3 Filter output for a simple neighborhood

If a filter is applied to the signal, the filter output at a specific point is the integral over the product of the Fourier transform of the filter and the Fourier transform of the local neighborhood around the point. Assuming that the neighborhood is simple, the latter is described by Equation (67). Let the filter output be denoted q', this gives

$$q' = \frac{1}{(2\pi)^n} \int S(\mathbf{u})\, F(\mathbf{u})\, d^n u \tag{71}$$

where integration is made over the entire Fourier domain. However, since S contains an impulse line, the integration can be restricted only to this line, i.e.,

$$q' = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(u)\, F(u\,\hat{\mathbf{x}})\, du \tag{72}$$

where the line is parameterized by $\mathbf{u} = u\,\hat{\mathbf{x}}$. Clearly, the integration can be partitioned into the two half lines which stretches out from the origin, i.e.,

$$q' = \frac{1}{2\pi} \left[\int_{0}^{\infty} G(-u)\, F(-u\,\hat{\mathbf{x}})\, du + \int_{0}^{\infty} G(u)\, F(u\,\hat{\mathbf{x}})\, du \right] \tag{73}$$

Since g is real valued, G is Hermitian, and q' can be written

$$q' = \frac{1}{2\pi}\left[\int_0^\infty G^\star(u)\, F(-u\,\hat{\mathbf{x}})\, du + \int_0^\infty G(u)\, F(u\,\hat{\mathbf{x}})\, du\right] \tag{74}$$

where * denotes complex conjugation.

4.4 Spherically separable filters

In the following, the filter *f* is assumed to be not only a quadrature filter, but also polar separable in the Fourier domain, i.e., its Fourier transform can be written as a product of one radial function *R* and one directional function *D*. Hence,

$$F(\mathbf{u}) = R(\rho)\, D(\hat{\mathbf{u}}) \tag{75}$$

where $\rho = ||\mathbf{u}||$, and $\hat{\mathbf{u}} = \mathbf{u}/\rho$. The quadrature property is then transferred to *D*, i.e., it is demanded that

$$D(\hat{\mathbf{u}}) = 0 \quad \text{if } \mathbf{u}^T\hat{\mathbf{n}} \leq 0 \tag{76}$$

Both *R* and *D* are here assumed to be real valued.

Inserting Equation (76) into Equation (74) yields

$$q' = \frac{1}{2\pi}\left[\int_0^\infty G^\star(u)R(u)D(-\hat{\mathbf{x}})du + \int_0^\infty G(u)R(u)D(\hat{\mathbf{x}})du\right] \tag{77}$$

and taking into account that q' is computed for a particular neighborhood where $\hat{\mathbf{x}}$ is fixed results in

$$q' = \frac{1}{2\pi}\left[D(-\hat{\mathbf{x}})\int_0^\infty G^\star(u)R(u)du + D(\hat{\mathbf{x}})\int_0^\infty G(u)R(u)du\right] \tag{78}$$

Setting

$$a = \frac{1}{2\pi} \int_{0}^{\infty} G(u)\, R(u)\, du \tag{79}$$

and remembering that R is real valued then implies that

$$q' = D(-\hat{\mathbf{x}})\, a^{\star} + D(\hat{\mathbf{x}})\, a \tag{80}$$

Taking the quadrature property into account it is clear that either $D(\hat{\mathbf{x}})$ or $D(-\hat{\mathbf{x}})$ is zero, implying that the two components do not interfere, and the magnitude of the quadrature filter output can be written

$$q = |q'| = |a|\, [D(-\hat{\mathbf{x}}) + D(\hat{\mathbf{x}})] \tag{81}$$

Thus, the magnitude of the filter output can be decomposed into two factors, $A = |a|$ which is orientation invariant and dependent on both G and R, and $[D(-\hat{\mathbf{x}}) + D(\hat{\mathbf{x}})]$ which is invariant to G and dependent on $\hat{\mathbf{x}}$. It is this latter dependency which is the topic of the next section.

It was mentioned that one prominent feature of a quadrature filter is the fact that q, together with the resulting orientation tensor, turn out to be phase-invariant for single-frequency signals. The proof of this assertion is presented in Exercise 19.

4.5 The directional function

To allow the resulting orientation tensor to meet the equivariance requirement it is necessary that the Fourier transforms of the filters have particular interpolation properties. Directional functions having the necessary properties were first suggested in [10] for the 2D case and in [5] for the 3D case. Regardless of dimension these functions can be written:

$$D(\hat{\mathbf{u}}) = \begin{cases} (\hat{\mathbf{u}}^T \hat{\mathbf{n}})^2 & \mathbf{u}^T \hat{\mathbf{n}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{82}$$

where $\hat{\mathbf{n}}_k$ is the filter directing vector. Thus, in the non-zero half of the Fourier domain, $D(\hat{\mathbf{u}})$ varies as $\cos^2 \Delta\varphi$, where $\Delta\varphi$ is the difference in angle between $\mathbf{u}$ and the filter direction $\hat{\mathbf{n}}$. Figure 3 is a plot of D as a function of $\Delta\varphi$, this plot also illustrates the directional function for the 2D case in a natural way. Figure 4 contains a visualization of the directional function in the 3D case for a particular choice of $\hat{\mathbf{n}}$.

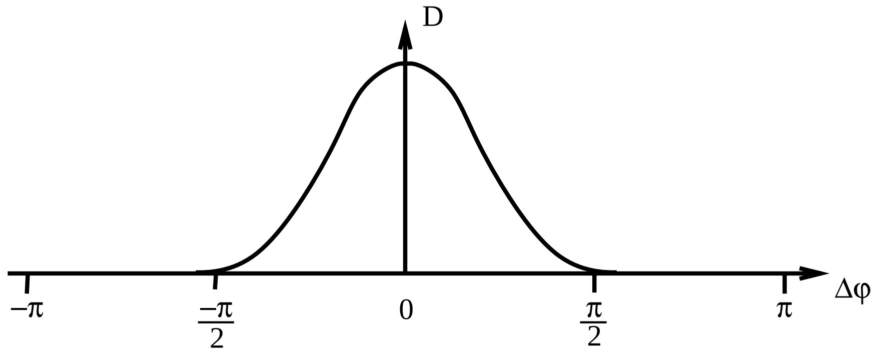


Figure 3: One-dimensional plot of the directional function with $\Delta\phi$, i.e., the angle between **u** and **n̂**, as parameter.

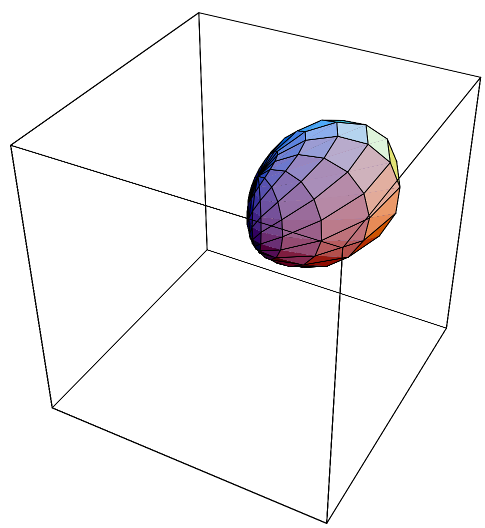


Figure 4: Angular plot,i.e., radius = $D(\hat{\mathbf{u}})$, of the directional function in 3-dimensional signal space for a particular choice of **n̂**. Origin of the Fourier domain is at the center of the cube.

4.6 The filter outputs

Finally, by combining Equations (81) and (82) the output magnitude from a quadrature filter in direction $\hat{\mathbf{n}}$ is found to be

$$q = A\,(\hat{\mathbf{x}}^T\hat{\mathbf{n}})^2 \tag{83}$$

where A is independent of the filter orientation and depends only on radial distribution of the signal spectrum $G(\rho)$ and the radial filter function $R(\rho)$. See Figure 5 for a visualization of the output magnitude q as a function of $\hat{\mathbf{x}}$ for a particular choice of $\hat{\mathbf{n}}$.

At this point it is finally possible to make the connection with the presented filtering and the orientation tensor discussed in Section 3. First, consider the two second order rank one tensors

$$\hat{\mathbf{T}} = \hat{\mathbf{x}}\,\hat{\mathbf{x}}^T \quad \hat{\mathbf{N}} = \hat{\mathbf{n}}\,\hat{\mathbf{n}}^T \tag{84}$$

which both have the single non-zero eigenvalue equal to one, with $\hat{\mathbf{x}}$ and $\hat{\mathbf{n}}$ as corresponding normalized eigenvectors. Using Equation (20), it is evidently the case that

$$q = A\,\langle\, \hat{\mathbf{T}} \mid \hat{\mathbf{N}} \,\rangle = \langle\, A\,\hat{\mathbf{T}} \mid \hat{\mathbf{N}} \,\rangle \tag{85}$$

which implies that the filter outputs can be interpreted as the scalar product between the tensor $\mathbf{T} = A\,\hat{\mathbf{T}}$, which represents the local orientation of the neighborhood, and $\hat{\mathbf{N}}$ which is a constant tensor related to the direction of the quadrature filter.

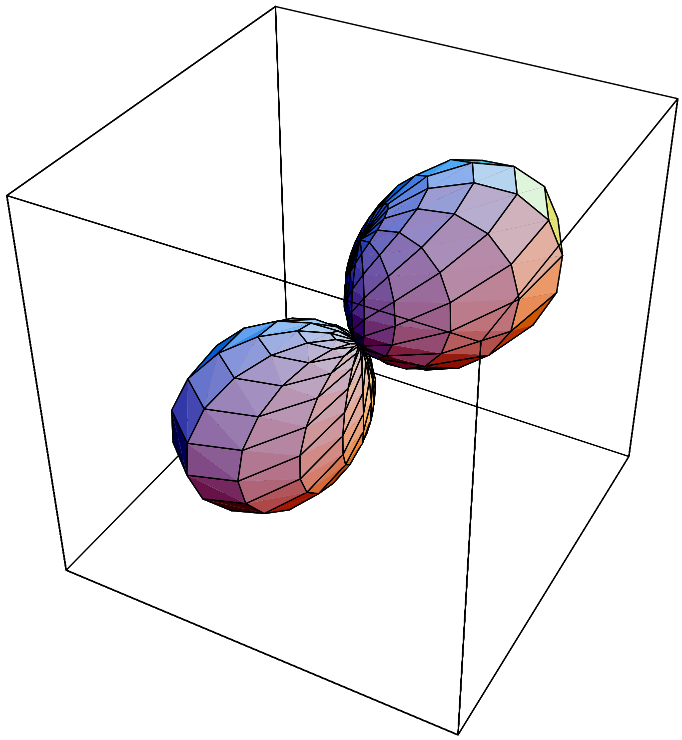


Figure 5: Angular plot of the quadrature output magnitude, q, as a function of the signal orienting vector $\hat{\mathbf{x}}$ for 3-dimensional simple signals and a particular choice of $\hat{\mathbf{n}}$.

4.7 Constructing the tensor

Having realized that filter outputs are described by Equation (85), it is possible to apply several filters of the prescribed type, all having the same radial frequency function R but different directing vectors $\hat{\mathbf{n}}_k$, and obtain a set of scalar products between the sought for tensor $\mathbf{T}$ and constant tensors

$$\hat{\mathbf{N}}_k = \hat{\mathbf{n}}_k \, \hat{\mathbf{n}}_k^T \tag{86}$$

According to Equation (34), if the latter set of tensors form a basis for $Sym(V^2)$, the $n(n+1)/2$-dimensional vector space of symmetric tensors, $\mathbf{T}$ can be written as

$$\mathbf{T} = \sum_{k}^{\frac{n(n+1)}{2}} \langle\, \mathbf{T} \mid \mathbf{N}_k \,\rangle\, \tilde{\mathbf{N}}_k = \sum_{k}^{\frac{n(n+1)}{2}} q_k\, \tilde{\mathbf{N}}_k \tag{87}$$

where $\{\tilde{\mathbf{N}}_k\}$ is the set of dual tensors relative to $\{\hat{\mathbf{N}}_k\}$, see Section 2.8. Hence, by using the magnitude of the filter outputs, q_k in a linear combination with a set of fixed and pre-calculated tensors $\{\tilde{\mathbf{N}}_k\}$, the sought for orientation tensor is obtained.

Evidently, at least $n(n+1)/2$ filters are needed to obtain a basis for $Sym(V^2)$. How to choose the corresponding filter directing vectors $\hat{\mathbf{n}}_k$ is an issue address in Section 5, but from a theoretical point of view it is sufficient that the corresponding tensors $\hat{\mathbf{N}}_k$ form a basis. The reason for choosing a particular set of filter directions is rather related to practical considerations, e.g., that the directions $\hat{\mathbf{n}}_k$ should be isotropically distributed over the space. Furthermore, in certain cases of dimensionality, e.g., the 4D case, it is desirable to use more than the smallest number of filter directions. In that case the constant tensors $\hat{\mathbf{N}}_k$ form a frame rather than a basis, and the tensors $\tilde{\mathbf{N}}_k$ is then the dual frame.

Regardless of how the filter directions are chosen, it should be noted that they cannot be choose such that $\{\hat{\mathbf{N}}_k\}$ is an ON-basis. To be sure, each $\hat{\mathbf{N}}_k$ is normalized, but since they are all of rank one, it follows that $\hat{\mathbf{N}}_k$ is orthogonal to $\hat{\mathbf{N}}_l$ if and only if $\hat{\mathbf{n}}_k$ is orthogonal to $\hat{\mathbf{n}}_l$. Since there are at most n orthogonal vectors $\hat{\mathbf{n}}_k$, it is not possible to find $n(n+1)/2$ orthogonal tensors $\hat{\mathbf{N}}_k$ if $n \geq 1$. As consequence, the dual basis $\{\tilde{\mathbf{N}}_k\}$ can not be the same as the original basis. However, each dual basis tensor is a linear combination of the original tensor basis, according to Section 2.8.

4.8 The radial function

It is clear from the preceding analysis that the radial function $R(\rho)$ can be chosen arbitrarily without violating the basic requirements. This makes the choice of $R(\rho)$ subject to considerations similar to those traditionally found in *one-dimensional* filter design. Typically $R(\rho)$ is a band-pass function having design parameters such as *center frequency* and *bandwidth*. Perhaps even more important than in traditional one-dimensional filter design are the concepts of locality and scale. Good radial functions are therefore found by studying the resulting filter simultaneously in the space-time and the Fourier domains. A radial filter function with useful properties, first suggested in [9], is given by:

$$R(\rho) = e^{-\frac{4}{B^2 \ln 2} \ln^2(\rho/\rho_i)} \tag{88}$$

This class of functions are termed *lognormal* functions. B is the relative bandwidth and ρ_i is the center frequency.

5 Working algorithms

This section addresses some of the practical issues which must be considered in order to develop implementations of the estimation procedure presented in Section 4. Particular choices of the filter directions for the cases of 2D, 3D, and 4D signals are presented, together with the corresponding dual bases $\{\tilde{\mathbf{N}}_k\}$. These results where originally presented in [10, 5, 6, 8].

As basic requirement for the choice of the filter directions is uniform distribution in space. Although not necessary for the tensor construction made in Section 4, there are a number of practical implications of this approach. First, since the filters are normally implemented as a number of optimized filter kernels on a discrete and finite grid, it is desirable to optimize only a few filters and obtain the rest through simple coordinate transformations like mirroring, e.g., in the coordinate axes, and rotations by multiples of 90°. A uniform distribution of the filter directions facilitates the process of generating the filter kernels. Second, as is shown in the following, the set of dual tensors $\{\tilde{\mathbf{N}}_k\}$ becomes particularly easy to realize when the filter directions are uniformly distributed.

The easiest way to accomplish a uniform distribution is to have the filter directions pass through the vertices of a regular polytope. Note, however, that for vertices located on opposite sides of the origin, only one of them can be used, since both $\hat{\mathbf{n}}$ and $-\hat{\mathbf{n}}$ gives the same basis tensor $\hat{\mathbf{N}} = \hat{\mathbf{n}}\hat{\mathbf{n}}^T$. This issue, and others, can make the choice of polytope a delicate matter.

5.1 The 2D case

In 2D, $Sym(V^2)$ is a three-dimensional space. Therefore, at least three filters are needed. In certain cases, however, four filters are used, and this is also discussed here. Finally, the consequences of using the vector representation of 2D orientation, Section 3.3, is also covered.

5.1.1 Three filters

First, the case of using three filters is considered. As mentioned above, their directing vectors $\{\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \hat{\mathbf{n}}_3\}$ should be uniformly distributed in the two-dimensional space V, and this is done by having them pointing to the vertices of a uniform triangle. The orientation of the triangle is of no importance, and just to make a particular choice, the following direction vectors are used.

$$\hat{\mathbf{n}}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \hat{\mathbf{n}}_2 = \begin{pmatrix} -1/2 \\ \sqrt{3}/2 \end{pmatrix} \quad \hat{\mathbf{n}}_3 = \begin{pmatrix} -1/2 \\ -\sqrt{3}/2 \end{pmatrix} \tag{89}$$

See Figure 6. It should be noted that the corresponding filters f_k, and therefore also the filter kernels, are such that the second and third filters are mirror images relative to the horizontal axis.

The tensor basis $\{\hat{\mathbf{N}}_k\}$ which corresponds to these direc-

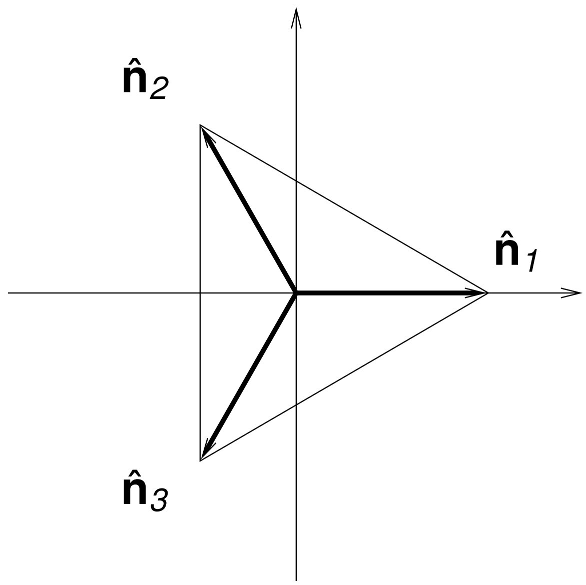


Figure 6: A triangle and the corresponding filter orienting vectors.

tions is given by $\hat{\mathbf{N}}_k = \hat{\mathbf{n}}_k\, \hat{\mathbf{n}}_k^T$, thus

$$
\begin{aligned}
\hat{\mathbf{N}}_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\
\hat{\mathbf{N}}_2 &= \begin{pmatrix} 1/4 & -\sqrt{3}/4 \\ -\sqrt{3}/4 & 3/4 \end{pmatrix} \\
\hat{\mathbf{N}}_3 &= \begin{pmatrix} 1/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 3/4 \end{pmatrix}
\end{aligned}
\tag{90}
$$

In Exercise 9 it is shown that the corresponding dual basis is given by

$$\tilde{\mathbf{N}}_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1/3 \end{pmatrix}$$
$$\tilde{\mathbf{N}}_2 = \begin{pmatrix} 0 & -1/\sqrt{3} \\ -1/\sqrt{3} & 2/3 \end{pmatrix} \tag{91}$$
$$\tilde{\mathbf{N}}_3 = \begin{pmatrix} 0 & 1/\sqrt{3} \\ 1/\sqrt{3} & 2/3 \end{pmatrix}$$

It should be noted that due to the uniform distribution of the filter directions there is a particularly simple relation between the basis and its dual basis according to

$$\tilde{\mathbf{N}}_k = \frac{4}{3}\,\hat{\mathbf{N}}_k - \frac{1}{3}\,\mathbf{I} \tag{92}$$

where **I** is the identity tensor.

5.1.2 Four filters

As is explained in Section 6, the orientation tensor can be used for representation of more than just the orientation of a single simple neighborhood. In order to facilitated this capability, however, the estimation procedure must carefully designed. One issue to consider then is to employ more than the minimal number of filters. For example, in 2D it proves profitable to use four filters, rather than three, provided that the extra computational effort caused by the additional filter is acceptable.

The polytope which gives a uniform distribution of the filter directions is not a square, which may be the first guess. The reason is that diagonally positioned vertices correspond to the same filter direction tensor $\hat{\mathbf{N}}$, which means that the resulting set of such tensors only span a two-dimensional space. Choosing four consecutive vertices of an octagon, however, gives both a uniform distribution of the filter directions and a set of direction tensors which span the tensor space. For example, it is possible to employ the following filter directions

$$\hat{\mathbf{n}}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \hat{\mathbf{n}}_2 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \hat{\mathbf{n}}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \hat{\mathbf{n}}_4 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad (93)$$

which are illustrated in Figure 7. The corresponding set of

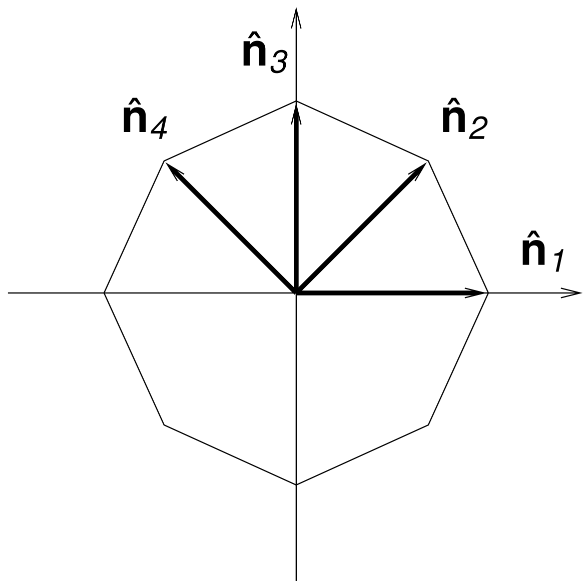


Figure 7: An octagon and the corresponding filter orienting vectors.

filter direction tensors is then

$$\begin{aligned} \hat{\mathbf{N}}_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \hat{\mathbf{N}}_2 &= \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \\ \hat{\mathbf{N}}_3 &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \hat{\mathbf{N}}_4 &= \begin{pmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix} \end{aligned} \tag{94}$$

It should be noted that this is a frame rather than a basis of the tensor space. Using the recipe in Section 2.9 for computing the dual frame results in (see Exercise 10)

$$\tilde{\mathbf{N}}_1 = \begin{pmatrix} 3/4 & 0 \\ 0 & -1/4 \end{pmatrix} \quad \tilde{\mathbf{N}}_2 = \begin{pmatrix} 1/4 & 1/2 \\ 1/2 & 1/4 \end{pmatrix}$$
$$\tilde{\mathbf{N}}_3 = \begin{pmatrix} -1/4 & 0 \\ 0 & 3/4 \end{pmatrix} \quad \tilde{\mathbf{N}}_4 = \begin{pmatrix} 1/4 & -1/2 \\ -1/2 & 1/4 \end{pmatrix} \tag{95}$$

Again, there is a simple relation between the frame tensors and their dual elements, according to

$$\tilde{\mathbf{N}}_k = \hat{\mathbf{N}}_k - \frac{1}{4}\mathbf{I} \tag{96}$$

Choosing the filter direction in this way implies that the corresponding four filter kernels are such that the first and third filters, as well as the second and fourth filters, are related through a 90° rotation. Hence, only two sets of filter coefficients have to be optimized.

A 22.5° rotation of the octagon in Figure 7 gives the filter directions illustrated in Figure 8. Evidently, filter $k+1$ is obtained by mirroring filter k in the axis k. Hence, by employing these filter directions, only one filter kernel needs to be optimized. Of course, the corresponding direction tensors $\hat{\mathbf{N}}_k$, and their dual tensors $\tilde{\mathbf{N}}_k$, can also be computed for this case, although the resulting matrices look a little bit messier than Equations (94) and (95). On the other hand, these alternative filter directions corresponds to a rotation of the original coordinate system by 22.5°, which simply means that the

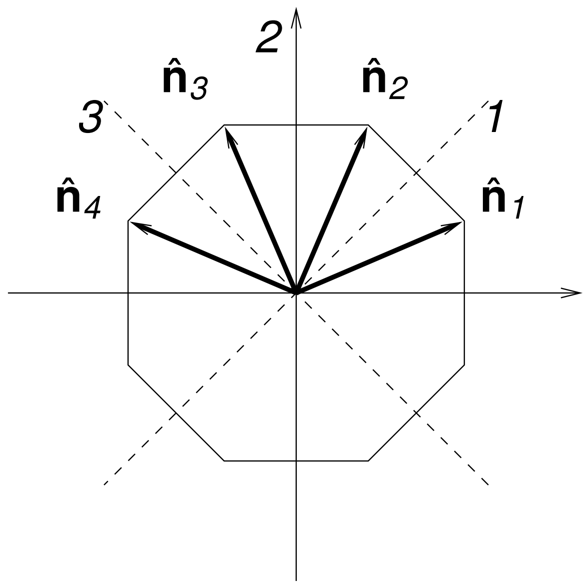


Figure 8: An alternative set of filter directing vectors.

resulting orientation tensor **T** gives the orientation relative to the rotated coordinate system. Therefore, it is possible to use the dual tensors in Equation (95) and defer the compensation due to the change of coordinate system until the tensor is being used.

To get a better feeling for how the orientation tensor **T** is constructed in the 2D case, see Exercise 18.

5.1.3 2D orientation vector

As mentioned in Section 3.3, the 2D case enables the use of a two-dimensional vector for representation of local orientation. Since the two components of this vector are simple linear combinations of the elements of the corresponding orientation tensor, and this tensor can be constructed as a linear combination of filter output magnitudes q_k and constant tensors $\tilde{\mathbf{N}}_k$, it must the case that the vector can be written as

$$\mathbf{z} = \sum_{k=1}^{3} q_k \, \mathbf{m}_k \tag{97}$$

where $\mathbf{m}_k$ are constant vectors given by

$$\mathbf{m}_k = \begin{pmatrix} m_{1,k} \\ m_{2,k} \end{pmatrix} = \begin{pmatrix} \tilde{N}_{11,k} - \tilde{N}_{22,k} \\ 2\,\tilde{N}_{12,k} \end{pmatrix} \tag{98}$$

and $\tilde{N}_{11,k}$,$\tilde{N}_{12,k}$,$\tilde{N}_{22.k}$ are the three elements of the tensor $\tilde{\mathbf{N}}_k$.

The two previous sections provides two particular choices of filter directions for three and four filter, respectively. For three filters, the combination of Equations (91) and (98) gives

$$\mathbf{z} = 4/3 \left[q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} q_2 \begin{pmatrix} -1/2 \\ -\sqrt{3}/2 \end{pmatrix} + q_3 \begin{pmatrix} -1/2 \\ \sqrt{3}/2 \end{pmatrix} \right] \tag{99}$$

For four filters, a combination of Equations (95) and (98) gives

$$\mathbf{z} = q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + q_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + q_3 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + q_4 \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tag{100}$$

In both cases it should be noted that the constant vectors $\hat{\mathbf{m}}_k$ used in the linear combination have *twice* the directional angle relative to the corresponding filter directions $\hat{\mathbf{n}}_k$.

5.2 The 3D case

In 3D, $Sym(V^2)$ is six-dimensional, so at least six filters are needed for estimation of the orientation tensor. Even though more than six filters can be considered, as was the case in 2D, it has proven practical to stick to the minimal number. Fortunately, there is a three-dimensional regular polytope which provides a uniform distribution of six filter directions; the icosahedron, see Figure 9. This polytope has twelve vertices, consisting of six pairs of points where the points of each pair are on opposite sides of the origin. Thus, it is possible to choose six points on one side of a plane through the origin as filter directions.

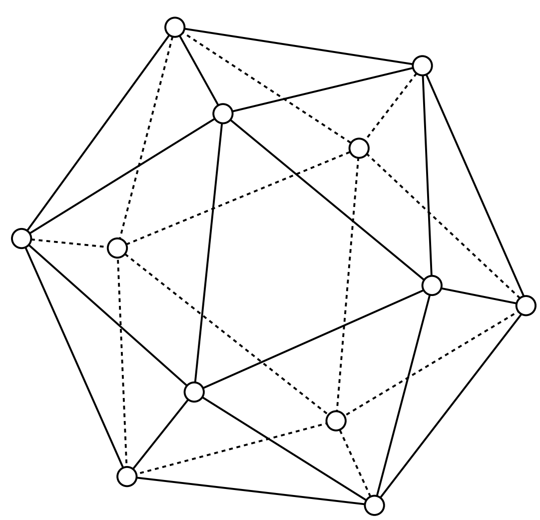


Figure 9: An icosahedron, one of the five Platonic polyhedra.

The following set of six filter directions, together with their negative counterparts, are the twelve vertices of an icosahedron.

$$
\begin{aligned}
\hat{\mathbf{n}}_1 &= \begin{pmatrix} a \\ 0 \\ b \end{pmatrix} & \hat{\mathbf{n}}_3 &= \begin{pmatrix} b \\ a \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_5 &= \begin{pmatrix} 0 \\ b \\ a \end{pmatrix} \\
\hat{\mathbf{n}}_2 &= \begin{pmatrix} -a \\ 0 \\ b \end{pmatrix} & \hat{\mathbf{n}}_4 &= \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_6 &= \begin{pmatrix} 0 \\ b \\ -a \end{pmatrix}
\end{aligned}
\tag{101}
$$

where

$$
a = \frac{2}{\sqrt{10 + 2\sqrt{5}}} \qquad b = \frac{1 + \sqrt{5}}{\sqrt{10 + 2\sqrt{5}}} \tag{102}
$$

The corresponding filter tensors $\hat{\mathbf{N}}_k$ are

$$
\begin{aligned}
\hat{\mathbf{N}}_1 &= \begin{pmatrix} a^2 & 0 & ab \\ 0 & 0 & 0 \\ ab & 0 & b^2 \end{pmatrix} & \hat{\mathbf{N}}_2 &= \begin{pmatrix} a^2 & 0 & -ab \\ 0 & 0 & 0 \\ -ab & 0 & b^2 \end{pmatrix} \\
\hat{\mathbf{N}}_3 &= \begin{pmatrix} b^2 & ab & 0 \\ ab & a^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \hat{\mathbf{N}}_4 &= \begin{pmatrix} b^2 & -ab & 0 \\ -ab & a^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\hat{\mathbf{N}}_5 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & b^2 & ab \\ 0 & ab & a^2 \end{pmatrix} & \hat{\mathbf{N}}_6 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & b^2 & -ab \\ 0 & -ab & a^2 \end{pmatrix}
\end{aligned}
\tag{103}
$$

where

$$a^2 = \frac{2}{5+\sqrt{5}} \qquad ab = \frac{1+\sqrt{5}}{5+\sqrt{5}} \qquad b^2 = \frac{3+\sqrt{5}}{5+\sqrt{5}} \tag{104}$$

In this case, the filter tensors form a basis, and in Exercise 11 it is shown that the corresponding dual tensors are

$$\begin{aligned}
\tilde{\mathbf{N}}_1 &= \begin{pmatrix} c & 0 & e \\ 0 & f & 0 \\ e & 0 & d \end{pmatrix} & \tilde{\mathbf{N}}_2 &= \begin{pmatrix} c & 0 & -e \\ 0 & f & 0 \\ -e & 0 & d \end{pmatrix} \\
\tilde{\mathbf{N}}_3 &= \begin{pmatrix} d & e & 0 \\ e & c & 0 \\ 0 & 0 & f \end{pmatrix} & \tilde{\mathbf{N}}_4 &= \begin{pmatrix} d & -e & 0 \\ -e & c & 0 \\ 0 & 0 & f \end{pmatrix} \\
\tilde{\mathbf{N}}_5 &= \begin{pmatrix} f & 0 & 0 \\ 0 & d & e \\ 0 & e & c \end{pmatrix} & \tilde{\mathbf{N}}_6 &= \begin{pmatrix} f & 0 & 0 \\ 0 & d & -e \\ 0 & -e & c \end{pmatrix}
\end{aligned} \tag{105}$$

where

$$c = \frac{5-\sqrt{5}}{4(5+\sqrt{5})}, \quad d = \frac{5+2\sqrt{5}}{2(5+\sqrt{5})}, \quad e = \frac{\sqrt{5}}{4}, \quad f = -\frac{1}{4} \tag{106}$$

A careful examination of the numerical values of the filter tensors and their corresponding dual tensors reveals the following simple relation

$$\tilde{\mathbf{N}}_k = \frac{5}{4}\hat{\mathbf{N}}_k - \frac{1}{4}\mathbf{I} \tag{107}$$

which is due to the uniform distribution of the filter directions.

As should be apparent from Equation (101), the six filter directions presented here are all related through simple mirroring operations. The practical implication of this property is that only one filter kernel needs to be optimized, and any of the other remaining five kernels can be obtained by the proper mirroring of the first one.

5.3 The 4D case

In 4D, $Sym(V^2)$ is ten-dimensional which implies that at least ten filters are needed for the tensor estimation. However, in 4D there is no regular polytope such that ten of its vertices can be used as uniformly distributed filter directions. Rather than using ten filter which are distributed in a non-uniform fashion, their number can be increased to twelve by considering the possibilities offered by the 24-cell [1]. This regular four-dimensional polytope has 24 vertices and, similar to the icosahedron, they consist of twelve pairs where the points of each pair are on opposite sides of the origin. Figure 10 shows an illustration of a projection of the 24-cell onto a two-dimensional space.

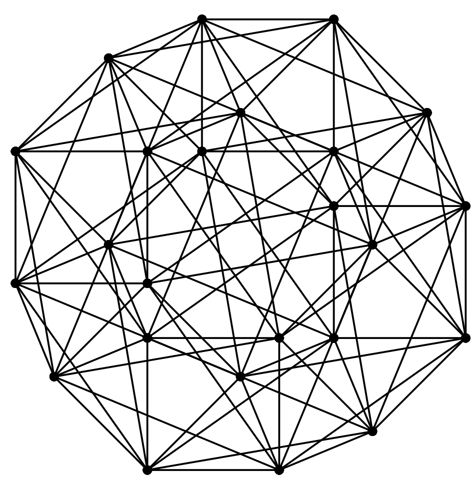


Figure 10: A projection of the 24–cell

The following set of ten filter directions, together with their negative counterparts, are the vertices of a 24-cell.

$$
\begin{array}{lll}
\hat{\mathbf{n}}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\
\hat{\mathbf{n}}_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_5 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \hat{\mathbf{n}}_6 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} \\
\hat{\mathbf{n}}_7 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_8 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} & \hat{\mathbf{n}}_9 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \\
\hat{\mathbf{n}}_{10} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} & \hat{\mathbf{n}}_{11} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} & \hat{\mathbf{n}}_{12} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}
\end{array}
\tag{108}
$$

The corresponding twelve filter tensors are

$$
\begin{aligned}
\hat{\mathbf{N}}_1 &= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \hat{\mathbf{N}}_2 &= \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\hat{\mathbf{N}}_3 &= \frac{1}{2}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \hat{\mathbf{N}}_4 &= \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\hat{\mathbf{N}}_5 &= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} & \hat{\mathbf{N}}_6 &= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \\
\hat{\mathbf{N}}_7 &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \hat{\mathbf{N}}_8 &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
\hat{\mathbf{N}}_9 &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} & \hat{\mathbf{N}}_{10} &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \\
\hat{\mathbf{N}}_{11} &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} & \hat{\mathbf{N}}_{12} &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}
\end{aligned}
\tag{109}
$$

Since two filter directions more than necessary are used, this set of tensors form a frame rather than a basis. Using the recipe presented in Section 2.9, the corresponding dual frame is given by (see Exercise 12)

$$
\begin{aligned}
\tilde{\mathbf{N}}_1 &= \frac{1}{6}\begin{pmatrix} 2 & 3 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} &
\tilde{\mathbf{N}}_2 &= \frac{1}{6}\begin{pmatrix} 2 & -3 & 0 & 0 \\ -3 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\
\tilde{\mathbf{N}}_3 &= \frac{1}{6}\begin{pmatrix} 2 & 0 & 3 & 0 \\ 0 & -1 & 0 & 0 \\ 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} &
\tilde{\mathbf{N}}_4 &= \frac{1}{6}\begin{pmatrix} 2 & 0 & -3 & 0 \\ 0 & -1 & 0 & 0 \\ -3 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\
\tilde{\mathbf{N}}_5 &= \frac{1}{6}\begin{pmatrix} 2 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 3 & 0 & 0 & 2 \end{pmatrix} &
\tilde{\mathbf{N}}_6 &= \frac{1}{6}\begin{pmatrix} 2 & 0 & 0 & -3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -3 & 0 & 0 & 2 \end{pmatrix} \\
\tilde{\mathbf{N}}_7 &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 3 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} &
\tilde{\mathbf{N}}_8 &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & -3 & 0 \\ 0 & -3 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\
\tilde{\mathbf{N}}_9 &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & -1 & 0 \\ 0 & 3 & 0 & 2 \end{pmatrix} &
\tilde{\mathbf{N}}_{10} &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -3 \\ 0 & 0 & -1 & 0 \\ 0 & -3 & 0 & 2 \end{pmatrix} \\
\tilde{\mathbf{N}}_{11} &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 3 & 2 \end{pmatrix} &
\tilde{\mathbf{N}}_{12} &= \frac{1}{6}\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & -3 \\ 0 & 0 & -3 & 2 \end{pmatrix}
\end{aligned}
\tag{110}
$$

It should be noted that also in the 4D case, when choosing the filter directions as the vertices of a hemi-24-cell, there is a simple relation between the filter direction tensors $\hat{\mathbf{N}}_k$ and their corresponding duals $\tilde{\mathbf{N}}_k$, according to

$$\tilde{\mathbf{N}}_k = \hat{\mathbf{N}}_k - \frac{1}{6}\mathbf{I} \tag{111}$$

Furthermore, the twelve filters kernels which must be realized, e.g., by means of an optimization process, are related through simple mirroring operations. In fact only one filter kernel has to be optimized, and the other eleven kernels can be obtained by suitable mirroring operations of the first one.

6 Interpretation of the orientation tensor

Applying algorithms for estimation of the local orientation tensor on data acquired from the real world, e.g., using the algorithms presented in the previous section, it will be apparent that not all neighborhoods are simple, Equation (39). There are many reasons why a neighborhood is not simple, e.g., the presence of noise or multiple linear structures of different orientations. Consequently, the resulting orientation tensor is not a rank one tensor, Equation (50). Therefore, some way of interpreting the estimated orientation tensor for the general case is needed. This section addresses this issue in two rather different ways. First, by considering the rank one tensor which is *closest* to the estimated one, thereby offering an interpretation of the tensor as representing a simple neighborhood. Second, by finding a consistent interpretation of tensors which have higher rank than one in terms of neighborhoods that are constant in other ways than just on hyperplanes. This approach is of particular interest in the 3D case. In both cases, the interpretation is achieved by considering the eigenvalues and eigenvectors of the tensor $\mathbf{T}$.

Given the estimation procedures of the previous section, the resulting orientation tensor is always symmetric, i.e., it can be decomposed as

$$\mathbf{T} = \sum_{k=1}^{n} \lambda_k \, \hat{\mathbf{e}}_k \, \hat{\mathbf{e}}_k^T \tag{112}$$

where each $\hat{\mathbf{e}}_k$ is an eigenvectors of $\mathbf{T}$ such that $\{\hat{\mathbf{e}}_k\}$ form an ON-basis, and $\{\lambda_k\}$ are the corresponding eigenvalues. For a simple neighborhood, the corresponding $\mathbf{T}$ is such that

$$\lambda_k = \begin{cases} A \geq 0 & k = 1 \\ 0 & k > 1 \end{cases} \tag{113}$$

Hence, the discussion in this section relates to the case when Equation (113) is not true. In order to make a useful interpretation of the orientation tensor it must meet at least one basic requirement: it should be positive semi-definite, i.e., $\lambda_k \geq 0$ for all k. The estimation procedures presented in Section 4 composes the estimated orientation tensor as a linear combination of filter output magnitudes q_k and dual filter tensors $\tilde{\mathbf{N}}_k$. A careful examination of each $\tilde{\mathbf{N}}_k$ shows that they are all indefinite, and it is only by assuming a simple neighborhood that the q_k are such that the resulting tensor is positive definite. However, in the general case, for non-simple neighborhoods, the resulting tensor may be indefinite. Rather than discarding such tensors, the convention of setting any negative eigenvalue to zero before making any further processing or interpretation of $\mathbf{T}$ is used.

6.1 Rank one approximation

Given an estimated orientation tensor **T**, not necessary of rank one, which rank one tensor $\mathbf{T}_s$ is *closest* to **T**? The rank one property of $\mathbf{T}_s$ implies that

$$\mathbf{T}_s = \lambda \, \hat{\mathbf{e}} \, \hat{\mathbf{e}}^T \tag{114}$$

and the question can be formulated as: which $\mathbf{T}_s$ minimizes

$$\varepsilon = ||\mathbf{T} - \mathbf{T}_s||^2 = ||\mathbf{T} - \lambda \, \hat{\mathbf{e}} \, \hat{\mathbf{e}}^T||^2 \tag{115}$$

The $\mathbf{T}_s$ which satisfies this criteria is referred to as a *rank one approximation* of **T**.

In Exercise 13 it is shown that any rank one approximation $\mathbf{T}_s$ of **T** is such that λ is the largest eigenvalue of **T**, with $\hat{\mathbf{e}}$ as the corresponding eigenvector, i.e., $\lambda = \lambda_1$ and $\hat{\mathbf{e}} = \hat{\mathbf{e}}_1$. Note that the choice of $\hat{\mathbf{e}}$ may ambiguous if $\lambda_1 = \lambda_2$, i.e., if the two largest eigenvalues of **T** are equal. Therefore, the rank one approximation of **T** is only of interest if $\lambda_1 > \lambda_2$.

By choosing $\mathbf{T}_s$ as the rank one approximation of **T**, the value of ε is

$$\varepsilon = ||\sum_{k=1}^{n} \lambda_k \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T - \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T||^2 = ||\sum_{k=2}^{n} \lambda_k \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T||^2 = \sum_{k=2}^{n} \lambda_k^2 \tag{116}$$

An upper bound of ε is then given by

$$\varepsilon \leq (n-1)\, \lambda_2^2 = (n-1)\, \lambda_1^2 \left(\frac{\lambda_2}{\lambda_1}\right)^2 \tag{117}$$

With the above considerations in mind, a measure is needed that indicates the successful of a rank one approximation of **T**. There are several candidates for such a measure, so the suggested one is not unique. Define

$$c_1 = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{118}$$

Since $\lambda_1 \geq \lambda_2$ it follows that $0 \leq c \leq 1$. Furthermore, it is the case that $c_1 = 1$ if and only if **T** is a rank one tensor, and $c_1 = 0$ if and only if $\lambda_1 = \lambda_2$. Hence, if $c_1 \approx 1$ this means that **T** is approximately a rank one tensor, corresponding to a simple neighborhood. Note, however, that it is only possible to assume that the corresponding neighborhood is simple, as there are non-simple neighborhoods for which **T** is of rank one when estimated according to Section 4.

Using the measure c_1 in Equation (117) gives

$$\varepsilon \leq (n - 1)\, \lambda_1^2\, (1 - c_1)^2 \tag{119}$$

which shows that the upper bound of ε vanishes when c_1 approaches unity and λ_1 is constant.

6.2 Estimation errors

The rank one approximation enables the estimated orientation tensor to be compared to ground truth, e.g., for evaluation of an estimation algorithm. Assume that the correct orientation of a neighborhood is known, represented by the vector $\hat{\mathbf{x}}$ which is a normal vector of the linear structure, and that the estimated orientation tensor is $\mathbf{T}$. It is then possible to make a rank one approximation of $\mathbf{T}$, resulting in $\mathbf{T}_s$, and compare the orientation represented by $\mathbf{T}_s$ with the true orientation, represented by $\hat{\mathbf{x}}$. A natural choice is to compute the angle between $\hat{\mathbf{x}}$ and the eigenvector $\hat{\mathbf{e}}_1$ of $\mathbf{T}_s$,

$$\alpha = \arccos\left(\left| \hat{\mathbf{x}}^T \hat{\mathbf{e}}_1 \right| \right) \tag{120}$$

Note that taking the absolute value of the scalar product implies that the signs of both $\hat{\mathbf{x}}$ and $\hat{\mathbf{e}}_1$ are irrelevant, thus $0 \leq \alpha \leq 90^\circ$. This angle can also be expressed as

$$\begin{aligned} \alpha &= \arccos\left(\sqrt{\left(\hat{\mathbf{x}}^T \hat{\mathbf{e}}_1\right)^2} \right) = \arccos\left(\sqrt{\langle\, \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1 \mid \hat{\mathbf{x}} \hat{\mathbf{x}}^T \,\rangle} \right) = \\ &= \arccos\left(\sqrt{\langle\, \hat{\mathbf{T}}_s \mid \hat{\mathbf{T}}_0 \,\rangle} \right) \end{aligned} \tag{121}$$

where $\hat{\mathbf{T}}_s$ and $\hat{\mathbf{T}}_0$ are the normalized versions of the rank one approximation of $\mathbf{T}_s$, and of the correct orientation tensor, respectively. Simple trigonometry shows that this angular error can also be written as

$$\alpha = \arcsin\left(\sqrt{\frac{1}{2}\left\|\hat{\mathbf{T}}_s - \hat{\mathbf{T}}_0\right\|^2}\right) \tag{122}$$

The above discussion relates to comparing one estimated orientation tensor $\mathbf{T}$ with the ground truth, represented by $\mathbf{T}_0$. What if the angular error is to be measured over a large set of orientation tensors, each corresponding to an angular error α_k? A straightforward approach would be to average α_k over the set. However, by doing so neither the wrap around effect of the angle nor the fact that α_k is always positive is taken into account. The first effect can be illustrated in 2D by comparing $\hat{\mathbf{T}}_s = \hat{\mathbf{e}}_1\,\hat{\mathbf{e}}_1^T$ and $\hat{\mathbf{T}}_0 = \hat{\mathbf{x}}\,\hat{\mathbf{x}}^T$ and let $\hat{\mathbf{x}}$ be subject to a continuous rotation. If the two vectors initially are equal, $\alpha = 0$, the rotation makes α increase monotonically to 90°. After this point, α decreases to zero, corresponding to the situation where $\hat{\mathbf{x}} = -\hat{\mathbf{e}}_1$. Hence, an increasing rotation angle does not necessary correspond to an increase in α. As a consequence, it is more reasonable to average $\sin\alpha$, since the difference of the sinus function for two angles is smaller the larger the angles are, e.g.,

$$\sin 90^\circ - \sin 80^\circ < \sin 15^\circ - \sin 5^\circ \tag{123}$$

The second effect, averaging of positive angles, implies that it is not possible to interpret the result as a true mean value of the angular error. Instead, it is normally reasonable to assume that the mean angular error is zero. In this case, the standard deviation $\mathbf{E}[\,\alpha_k^2\,]$ is an interesting measure.

Combining these two observations results in the following formulation of the angular error measure $\Delta\alpha$

$$\sin^2 \Delta\alpha = \mathbf{E}[\, \sin^2 \alpha_k \,] \tag{124}$$

Note that this formulation takes the wrap around effect into account, and that $\Delta\alpha$ amounts to a measure of standard deviation for small angles α_k. Each α_k is described by Equation (122), which finally gives

$$\sin^2 \Delta\alpha = \frac{1}{2}\, \mathbf{E}\left[\, \left\| \hat{\mathbf{T}}_s - \hat{\mathbf{T}}_0 \right\|^2 \right] \tag{125}$$

or

$$\Delta\alpha = \sin^{-1} \sqrt{\frac{1}{2}\, \mathbf{E}\left[\, \left\| \hat{\mathbf{T}}_s - \hat{\mathbf{T}}_0 \right\|^2 \right]} \tag{126}$$

6.3 Higher rank neighborhoods in 3D

Simple neighborhoods are represented by tensors having rank one. One the other hand, in higher dimensional data there exist highly structured neighborhoods that are not simple. In that case, the rank of **T** can be used to reflect the complexity of the neighborhood. The eigenvalue distributions and the corresponding tensor representations are given below for three particular cases of **T** in 3D. The interpretation of **T** is based on the energy distribution in the Fourier domain for different autocorrelation functions of the signal.

The case of a simple signal, and its representation in terms of a rank one tensor, has already be discussed, but the following facts are mentioned here in order to make generalizations. For the case of an approximately simple signal, where g, the variation of the signal across the linear structure, has a well localized autocorrelation function, the corresponding autocorrelation function is an impulse plane, with the same orientation as the signal. In the Fourier domain, the signal is approximately concentrated on a line, passing through the origin, constituting a one-dimensional subspace. The eigenvector $\hat{\mathbf{e}}_1$ of the corresponding rank one orientation tensor **T** spans the subspace, and it is also a normal vector to the parallel planes.

Consider now the case of a 3D signal which is approximately constant along parallel lines. Its autocorrelation function is then an impulse line, of the same orientation as the lines of constancy. In the Fourier domain, this signal is approximately concentrated on a plane passing through the origin. This plane is then a two-dimensional subspace and, similar to the planar case, a set of two vectors which span the subspace is used as a representation of this neighborhood. The corresponding orientation tensor is of rank two. See Section 6.3.2.

Finally, consider a signal which is isotropic, i.e., it is rotationally symmetric. This property is the inherited both by its autocorrelation function as well as by its Fourier transform. Since there is no concentration into subspaces in the Fourier domain, three vectors are needed to span the full Fourier domain. Hence, the corresponding orientation tensor has rank three. See Section 6.3.3.

It should be noted that the estimation procedures presented in Section 4, applied to the three different cases, gives approximately the intended orientation tensor. In the planar and isotropic cases, the orientation tensor is exactly the intended one. For the line case the resulting tensor can be shown to be the intended one only if the neighborhood is an impulse line. Note, however, that this is a shortcoming of the estimation procedure, rather than of the representation.

Figures 11, 12 and 13 shows the three cases discussed above, the left figure illustrates the autocorrelation function in the spatial domain, and the right figure illustrates the energy distribution in the Fourier domain. Note the reciprocity between the first two cases. The following summarizes this discussion and presents the eigensystem of the orientation tensor for the three cases.

6.3.1 The plane case

- The signal is constant on parallel planes.
- The Fourier transform of the signal is concentrated on a line.
- $\hat{\mathbf{e}}_1$ is a normal vector of the planes in the spatial domain, and a direction vector of the lines in the Fourier domain.
- The orientation tensor is

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \tag{127}$$

- In practice, any orientation tensor characterized by

$$\lambda_1 \gg \lambda_2 \tag{128}$$

  is classified as representing the plane case. See Figure 11.

6.3.2 The line case

- The signal is constant on parallel lines.
- The Fourier transform of the signal is concentrated on a plane.
- $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are perpendicular to the lines in the spatial domain, and span the plane in the Fourier domain.
- The orientation tensor is

$$\mathbf{T} = \lambda \left(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T \right) \tag{129}$$

- In practice, any orientation tensor characterized by

$$\lambda_1 \simeq \lambda_2 \gg \lambda_3 \tag{130}$$

  is classified as representing the line case. See Figure 12.

Note that in this case, when two eigenvalues are equal, the corresponding eigenvectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are ambiguous with respect to their individual orientations. However, given the plane in the Fourier domain, any pair of perpendicular vectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ which span the plane gives the same $\mathbf{T}$.

6.3.3 The isotropic case

- The signal is isotropic or rotationally symmetric.
- The orientation tensor is

$$\mathbf{T} = \lambda \left(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T \right) = \lambda \, \mathbf{I} \tag{131}$$

- In practice, any orientation tensor characterized by

$$\lambda_1 \simeq \lambda_2 \simeq \lambda_3 \tag{132}$$

  is classified as representing the isotropic case. See Figure 13.

The particular choice of eigenvectors is also ambiguous in this case.

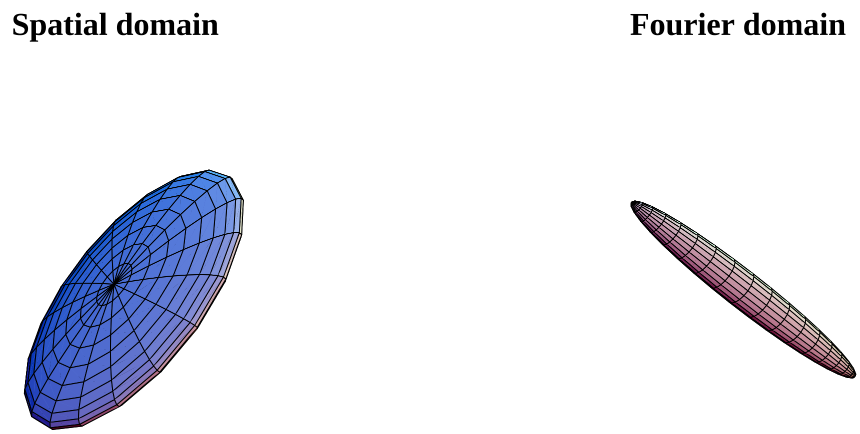


Figure 11: The plane case: A planar autocorrelation function in the spatial domain corresponds to energy being distributed on a line in the Fourier domain.

Figure 12: The line case: An autocorrelation function concentrated on a line in the spatial domain corresponds to a planar energy distribution in the Fourier domain.

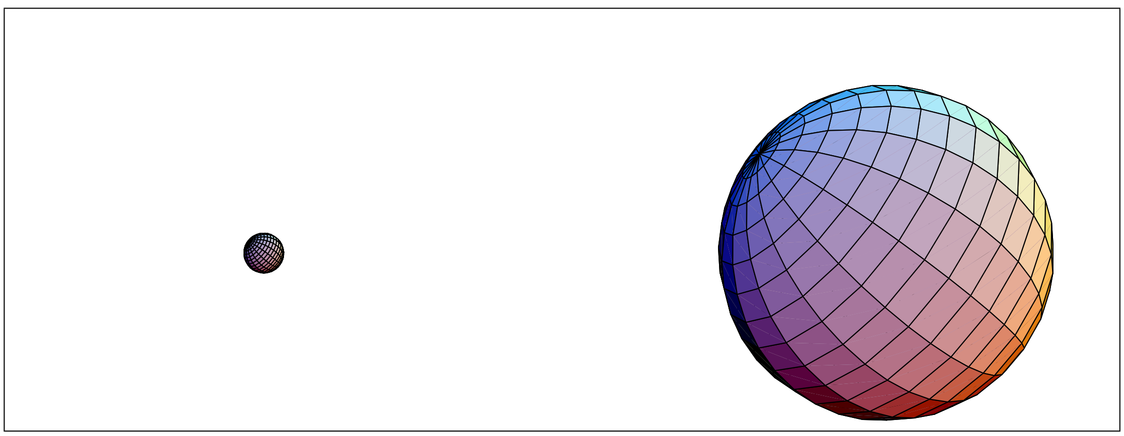


Figure 13: The isotropic case: A spherical autocorrelation function in the spatial domain corresponds to a spherical energy distribution in the Fourier domain.

6.3.4 The general 3D case

Obviously, not every 3D signal can be classified as belonging to one of the three cases presented above. However, any 3D orientation tensor $\mathbf{T}$,

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T \tag{133}$$

can be written as a linear combination of tensors of rank one, two, and three, according to

$$\mathbf{T} = (\lambda_1 - \lambda_2)\, \mathbf{T}_1 + (\lambda_2 - \lambda_3)\, \mathbf{T}_2 + \lambda_3\, \mathbf{T}_3 \tag{134}$$

where

$$\mathbf{T}_1 = \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T \tag{135}$$

$$\mathbf{T}_2 = \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T \tag{136}$$

$$\mathbf{T}_3 = \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T = \mathbf{I} \tag{137}$$

Note that $\mathbf{T}_1$ is a rank one tensor, $\mathbf{T}_2$ is a rank two tensor, and $\mathbf{T}_3$ is a rank three tensor.

Section 6.1 introduced a measure c_1, Equation (118), which can used for determining the closeness to the rank one case. Having decomposed a general tensor according to Equation (134), it is reasonable to use the following three measures as indicators for the three different cases

$$c_1 = \frac{\lambda_1 - \lambda_2}{\lambda_1} \qquad \text{The plane case} \tag{138}$$

$$c_2 = \frac{\lambda_2 - \lambda_3}{\lambda_1} \qquad \text{The line case} \tag{139}$$

$$c_3 = \frac{\lambda_3}{\lambda_1} \qquad \text{The isotropic case} \tag{140}$$

Note that

$$0 \leq c_k \leq 1 \tag{141}$$

and

$$c_1 + c_2 + c_3 = 1 \tag{142}$$

These three measures can be used to classify the 3D orientation tensor into either of the three classes, e.g., by choosing class k if c_k is the largest of the three. In most practical implementations, however, a null class is needed for those cases when the difference between the largest and second largest is too small. The particular values to be used as thresholds for each of the indicators in order to classify a tensor is entirely implementation specific. See Exercise 14.

7 Time sequences and velocity

Consider an image sequence, and some spatial neighborhood in this sequence which contains a linear structure, e.g., a line or an edge. The sequence can then be described as a 3D signal, with two spatial and one temporal dimensions. Regardless of whether the linear structure moves or not, it generates a 3D structure that corresponds to a simple neighborhood, i.e., the signal is locally constant on parallel planes. The orientation of these planes is determined by the motion of the linear structure. If the structure does not move, the planes of constancy are perpendicular to any spatial dimension, and the larger the velocity is, the less is the angle between the planes and the spatial dimensions.

The same discussion applies to a moving point. In that case, the corresponding 3D signal is constant on a line. If the point does not move, the line is perpendicular to the spatial dimensions, and the larger the velocity is, the less is the angle between the line and the spatial dimensions.

With this in mind, it is natural to design an estimation procedure for local 2D motion by means of estimation of local orientation in 3D. Using the orientation representation presented in Sections 3 and 6.3, and the estimation procedures presented in Section 5.2, the resulting orientation tensor is capable of determining which case it is (a moving linear structure or a moving point), as well as the velocity. The former is found by considering the eigenvalues of the tensor, and the latter by considering the appropriate eigenvector.

7.1 Moving linear structure

Before the properties of the orientation tensor for this case are discussed, the so called aperture problem must be addressed. If a moving linear structure is observed through a small aperture, the only velocity which can be determined is the component perpendicular to the structure. This corresponds to the situation where the 3D orientation is estimated by applying a set of filter kernels at each local neighborhood. Note that the terminology is somewhat misleading since it is the structure of the signal which causes the problem rather than the aperture. See Figure 14 for an illustration of the aperture problem. Neighborhoods labeled 'P' contain a simple local image structure, and only the normal velocity can be estimated. Neighborhoods labeled 'L' are not simple, and the true velocity can be estimated.

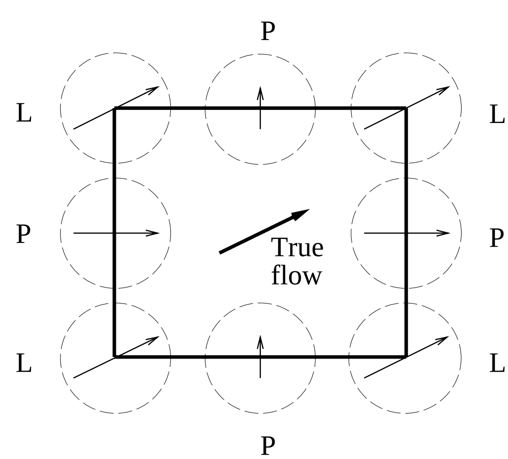


Figure 14: Illustration of the 'aperture problem'.

Consider a locally linear structure which moves with the velocity components

$$\tilde{\mathbf{v}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{143}$$

over the 2D image. If the unit of time is one per image frame in the sequence, this implies that any point on the linear structure moves a distance in 3D equal to

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ 1 \end{pmatrix} \tag{144}$$

between each frame. The vector $\mathbf{v}$ is referred to as the *spatio-temporal velocity vector*. From the above discussion is should be clear that $\mathbf{v}$ lies in the three-dimensional plane of constancy which is created when the linear structure moves. This implies that $\mathbf{v}$ is an eigenvector of the orientation tensor, and its eigenvalue is zero.

Since the signal in this case is locally simple, the corresponding orientation tensor is of rank one. If λ_1 is its largest eigenvalue, with

$$\hat{\mathbf{e}}_1 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{145}$$

as a corresponding normalized eigenvector, this means that $\hat{\mathbf{e}}_1$ and $\mathbf{v}$ are perpendicular, i.e.,

$$v_1\, x_1 + v_2\, x_2 + x_3 = 0 \tag{146}$$

Due to the aperture problem, this is the only statement that can be made about $\mathbf{v}$ (and $\tilde{\mathbf{v}}$). On the other hand, if $\hat{\mathbf{e}}_1$ is projected to the spatial dimensions, the resulting vector $\mathbf{n}$,

$$\mathbf{n} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{147}$$

is a normal vector of the linear 2D structure of the signal. Hence, the vector

$$\mathbf{m} = \begin{pmatrix} x_2 \\ -x_1 \end{pmatrix} \tag{148}$$

which is perpendicular to $\mathbf{n}$, points along the linear 2D structure. See Exercise 17. If $\tilde{\mathbf{v}}$ represents the normal velocity, this means that $\mathbf{m}$ and $\tilde{\mathbf{v}}$ are perpendicular, i.e.,

$$\tilde{\mathbf{v}}^T \mathbf{m} = v_1\, x_2 - v_2\, x_1 = 0 \tag{149}$$

Combining Equations (146) and (149) finally yields

$$\tilde{\mathbf{v}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = -\frac{x_3}{x_1^2 + x_2^2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{150}$$

Note that the velocity estimate derived here is not necessary the true velocity of the linear structure, it is the normal component of the true velocity.

Figure 15 shows an illustration of a local 3D neighborhood corresponding to a moving linear structure. The three horizontal lines are the image plane for three consecutive frames. The oblique thicker line is the plane generated by the moving linear structure. The spatio-temporal velocity vector $\mathbf{v}$ lies in this plane, and $\hat{\mathbf{e}}_1$ is perpendicular to it. The vector $\tilde{\mathbf{v}}$ is the projection of $\mathbf{v}$ onto the image plane, and $\mathbf{n}$ is the projection of $\hat{\mathbf{e}}_1$ onto the same plane. Note that in this figure, the temporal dimension goes from bottom to top, and that one of the spatial dimensions is horizontal.

Figure 16 shows the same situation, but in the image plane for a specific frame. The vectors $\tilde{\mathbf{v}}$ and $\mathbf{n}$ are perpendicular to the linear structure and $\mathbf{m}$ is parallel to it. Note that both dimensions in the figure are spatial.

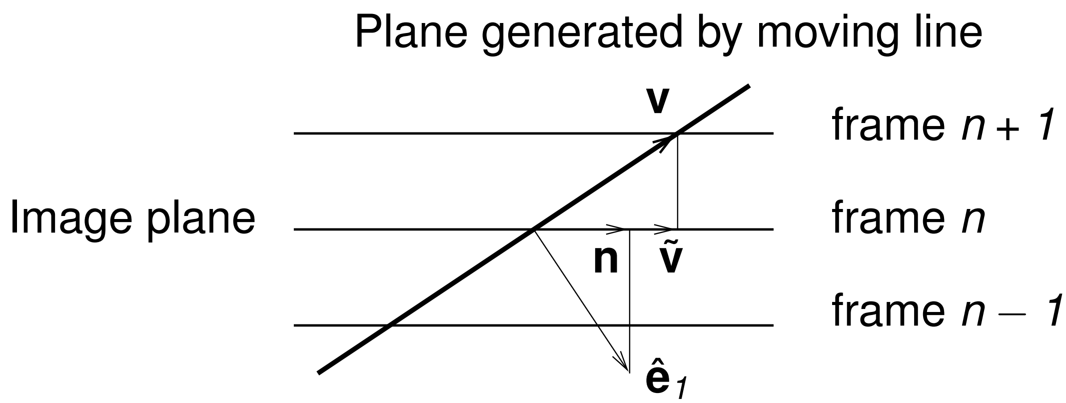


Figure 15: A spatio-temporal neighborhood for the case of moving linear structure.

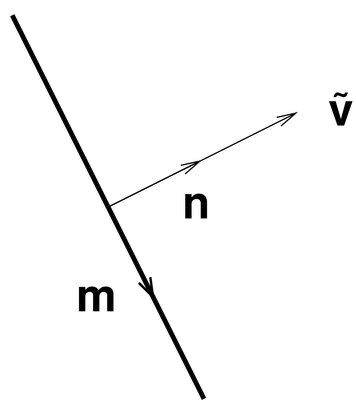


Figure 16: The image plane in the case of a moving linear structure.

7.2 Moving point

In the case of a moving point there is no aperture problem. Similarly to the case of a moving linear structure, however, a velocity vector $\tilde{\mathbf{v}}$ can be introduced together with the corresponding spatio-temporal velocity vector $\mathbf{v}$, Equations (143) and (144). In this case, the corresponding orientation tensor is of rank two, see Section 6.3.2, and there are two perpendicular eigenvectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, both with non-zero eigenvalues. Clearly, $\mathbf{v}$ is perpendicular to both $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, which means that

$$\mathbf{v} \propto \hat{\mathbf{e}}_3 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{151}$$

Consequently, the velocity vector $\tilde{\mathbf{v}}$ is given by

$$\tilde{\mathbf{v}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{152}$$

This velocity estimate is the true velocity of the point.

7.3 The general case

A general approach for estimating local velocity in a 2D image can now be formulated as follows.

- Consider the image sequence as a three-dimensional signal, and estimate, for each neighborhood, the corresponding 3D orientation tensor.
- The eigenvalues and eigenvectors of each orientation tensor are then computed.
- Using the measures discussed in Section 6.3.4, each tensor can be classified as being either approximately corresponding to the plane case, to the line case, or neither of these two cases (corresponding to the isotropic case and the null class).
- If the tensor is found to describe the plane case, corresponding to a moving linear structure, the normal velocity estimate derived in Section 7.1, Equation (150), is computed.
- If the tensor is found to describe the line case, corresponding to a moving point, the velocity estimate derived in Section 7.2, Equation (152), is computed.
- If the tensor describes neither the plane nor the line case no velocity is computed.

See Exercise 16.

It should be noted that the last case mentioned above, when the orientation tensor describes neither the plane nor the line case, has to be treated correctly. First of all, it is not possible to compute a velocity estimate in this case, but this does not imply that there is not motion at the corresponding point of the volume, i.e., that the velocity is zero. It is rather the case that the velocity is *unknown* or indeterminable. Therefore, it is not correct to set the velocity vector $\tilde{\mathbf{v}}$ to zero at all points corresponding to this case since these will be confused with zero velocity vectors which are estimated from plane or line type of tensors, corresponding to actual motion. Instead, a separate descriptor of certainty or confidence should be used in combination with the velocity vector as a result from the above velocity estimation.

Second, any type of local structure which is not a linear structure or a point, e.g., a corner or some other non-simple type of signal, is likely to result in a tensor that does not describe the plane or line cases, and this regardless of its motion. Therefore, even though the human eye sees a well-defined motion, the above motion estimation method may not be able to represent this fact. Solutions to this problem are discussed in the following section.

Third, the motion model used here assumes that the local motion is relatively constant over time. Only then does a moving linear structure or point correspond to a planar or linear structure in the image volume. If, for example, the line or point accelerate, the corresponding 3D structure is rather a parabolic surface or curve. If the deviation from linear motion is large enough, it is likely that the estimation procedure which generates the 3D orientation tensor produce tensors which are neither of rank one nor of rank two at these points. Therefore, tensors of this type can also correspond to moving linear structures or points which moves in a non-linear fashion. The acceptable deviation from linear motion is normally given by the temporal size of the filters being used for the estimation of the tensor.

7.4 Averaging of orientation tensors

The only case when the true velocity can be obtained is for the case of a moving point, corresponding to a rank two 3D orientation tensor. To be sure, it is enough for the moving structure to be non-linear or else the aperture problem implies that only the normal velocity can be estimated. For non-linear structures in general, however, the estimated orientation may not result in the proper rank two tensor. Consider the neighborhoods labeled 'L' in Figure 14. In these neighborhood it is possible to measure the true velocity, but since the local structure is a corner, it depends on the qualities of the the estimation algorithm being used whether or not the orientation tensor at these points is of rank two. In fact, it is only neighborhoods that contain a linear structure that gives a correct description of the 3D orientation in terms of a rank one tensor. At these points, however, only the normal velocity can be measured.

Consider again Figure 14, and in particular the neighborhoods labeled 'P' at the upper and left right borders. Evidently, the true velocity of these two borders are the same, but due to the aperture problem only the normal velocities can be determined there. To overcome this problem, recall that for the case of a moving linear structure, the spatio-temporal velocity vector **v** is an eigenvector of the orientation tensor **T**, with corresponding eigenvalue equal to zero. Hence, if $\mathbf{T}_1$ and $\mathbf{T}_2$ are the orientation tensors estimated at these two neighborhoods, **v** must be an eigenvector of zero eigenvalue relative to both tensors. In Exercise 15 it is shown that this property is inherited by the sum

$$\mathbf{T}_a = \mathbf{T}_1 + \mathbf{T}_2 \tag{153}$$

Thus, **v** is an eigenvector of $\mathbf{T}_a$ and the corresponding eigenvalue is zero. The point is that the rank one property of $\mathbf{T}_1$ and $\mathbf{T}_2$ *is not* inherited by $\mathbf{T}_a$. Instead $\mathbf{T}_a$ is a rank two tensor, provided that the local orientation of the two neighborhoods are different.

As a consequence of this observation, an estimate of the true velocity can be obtain by summing, or averaging, rank one tensors corresponding to linear structures which moves with the same velocity, provided that there is a sufficiently large difference in their orientation. The resulting integrated tensor $\mathbf{T}_a$ is then a rank two tensor from which the true velocity can be computed using the results derived in Section 7.2.

8 Exercises

The exercises part of this package contains the following exercises

- Exercise 1: Tensor scalar product
- Exercise 2: More tensor scalar products
- Exercise 3: Coordinates, bases and dual bases
- Exercise 4: Computing the dual basis
- Exercise 5: The eigenvectors of $\dfrac{d\mathbf{T}}{d\varepsilon}$
- Exercise 6: Establishing the orientation tensor in 2D
- Exercise 7: Establishing the local structure corresponding to a 2D orientation tensor.
- Exercise 8: The Fourier transform of simple functions.
- Exercise 9: Dual tensors for the 2D case with 3 filters.
- Exercise 10: Dual tensors for the 2D case with 4 filters.
- Exercise 11: Dual tensors for the 3D case with 6 filters.
- Exercise 12: Dual tensors for the 4D case with 12 filters.
- Exercise 13: Rank one approximation.
- Exercise 14: Classification of 3D orientation tensors.
- Exercise 15: Rank one preservation under tensor addition.

- Exercise 16: Image velocity in practice.
- Exercise 17: Image velocity in theory.
- Exercise 18: Computing a 2D orientation tensor.
- Exercise 19: Quadrature filters give phase invariance

Exercise 1

Consider two symmetric tensors **T** and **U** which are decomposed according to

$$\mathbf{T} = \sum_{i=1}^{n} \lambda_i \, \hat{\mathbf{e}}_i \, \hat{\mathbf{e}}_i^T, \qquad \mathbf{U} = \sum_{j=1}^{n} \sigma_j \, \hat{\mathbf{f}}_j \, \hat{\mathbf{f}}_j^T \tag{154}$$

Show that the following expression

$$\sum_{ij} \lambda_i \, \sigma_j \, \langle \, \hat{\mathbf{e}}_i \mid \hat{\mathbf{f}}_j \, \rangle^2 \tag{155}$$

is a scalar product between **T** and **U**.

Hint Solution

Hint to exercise 1

To be a proper scalar product it must satisfy the five conditions presented in Equation (4). You will probably find that all but the second condition are straightforward to demonstrate.

The second condition implies proving that

$$\langle\, \mathbf{T} + \mathbf{U} \mid \mathbf{W} \,\rangle = \langle\, \mathbf{T} \mid \mathbf{W} \,\rangle + \langle\, \mathbf{U} \mid \mathbf{W} \,\rangle \tag{156}$$

for all $\mathbf{W}$. To do so, assume that $\mathbf{T}$ and $\mathbf{U}$ are expanded according to Equation (154), and that

$$\mathbf{T} + \mathbf{U} = \sum_{k=1}^{n} \gamma_k \, \hat{\mathbf{g}}_k \, \hat{\mathbf{g}}_k^T, \qquad \mathbf{W} = \sum_{m=1}^{n} \alpha_m \, \hat{\mathbf{h}}_m \, \hat{\mathbf{h}}_m^T \tag{157}$$

where $\{\mathbf{g}_k\}$ and $\{\hat{\mathbf{h}}_m\}$ are two ON-bases of eigenvectors relative to $\mathbf{T} + \mathbf{U}$ and $\mathbf{W}$, respectively, and $\{\gamma_k\}$ and $\{\alpha_m\}$ are the corresponding eigenvalues. To prove Equation (156), two more relations are needed. First, each $\hat{\mathbf{g}}_k$ is an eigenvector of $\mathbf{T} + \mathbf{U}$. Using Equation (154), this implies that

$$\sum_i \lambda_i \,\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{g}}_k \,\rangle\, \hat{\mathbf{e}}_i + \sum_j \sigma_j \,\langle\, \hat{\mathbf{f}}_j \mid \hat{\mathbf{g}}_k \,\rangle\, \hat{\mathbf{f}}_j = \gamma_k \, \hat{\mathbf{g}}_k \tag{158}$$

Second, since any vector is an eigenvector of $\mathbf{I}$ with corresponding eigenvalue equal to unity, in particular this relates to the vectors $\hat{\mathbf{g}}_k$, and it follows that

$$\mathbf{I} = \sum_{k=1}^{n} \hat{\mathbf{g}}_k \, \hat{\mathbf{g}}_k^T \tag{159}$$

and therefore

$$\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{h}}_m \,\rangle = \hat{\mathbf{e}}_i^T \,\mathbf{I}\, \hat{\mathbf{h}}_m = \sum_k \langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{g}}_k \,\rangle \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle \tag{160}$$

and similarly

$$\langle\, \hat{\mathbf{f}}_j \mid \hat{\mathbf{h}}_m \,\rangle = \hat{\mathbf{f}}_j^T \,\mathbf{I}\, \hat{\mathbf{h}}_m = \sum_k \langle\, \hat{\mathbf{f}}_j \mid \hat{\mathbf{g}}_k \,\rangle \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle \tag{161}$$

By using these relations in the proper way, the second condition follows immediately.

Solution to exercise 1

All conditions but the second one follows immediately. Using the hints and the definition of the tensor scalar product, Equation (20), it follows that

$$\langle\, \mathbf{T} + \mathbf{U} \mid \mathbf{W} \,\rangle = \sum_{km} \gamma_k\, \alpha_m \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle^2 = \tag{162}$$

$$= \sum_{km} \langle\, \gamma_k\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle\, \alpha_m \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle \tag{163}$$

Now, insert the left hand side of Equation (158) instead of $\gamma_k\, \hat{\mathbf{g}}_k$ in the above equation. This gives

$$\langle\, \mathbf{T} + \mathbf{U} \mid \mathbf{W} \,\rangle = \sum_{ikm} \lambda_i\, \alpha_m \,\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{g}}_k \,\rangle \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle \,\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{h}}_m \,\rangle + \sum_{jkm} \sigma_j\, \alpha_m \,\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{g}}_k \,\rangle \,\langle\, \hat{\mathbf{g}}_k \mid \hat{\mathbf{h}}_m \,\rangle \,\langle\, \hat{\mathbf{f}}_j \mid \hat{\mathbf{h}}_m \,\rangle \tag{164}$$

Using Equations (160) and (161) this reduces to

$$\langle\, \mathbf{T} + \mathbf{U} \mid \mathbf{W} \,\rangle = \sum_{im} \lambda_i\, \alpha_m \,\langle\, \hat{\mathbf{e}}_i \mid \hat{\mathbf{h}}_m \,\rangle^2 + \sum_{jm} \sigma_j\, \alpha_m \,\langle\, \hat{\mathbf{f}}_j \mid \hat{\mathbf{h}}_m \,\rangle^2 = \langle\, \mathbf{T} \mid \mathbf{W} \,\rangle + \langle\, \mathbf{U} \mid \mathbf{W} \,\rangle \tag{165}$$

The last equality follows from the definition of the tensor scalar product. This concludes the proof of Equation (156).

Exercise 2

Show that the scalar product between **T** and **U** also can be expressed as

$$\langle \, \mathbf{T} \mid \mathbf{U} \, \rangle = \sum_{ij} T_{ij} \, U_{ij} \tag{166}$$

and

$$\langle \, \mathbf{T} \mid \mathbf{U} \, \rangle = \mathrm{tr}(\mathbf{T}^T \mathbf{U}) \tag{167}$$

Hint Solution

Hint to exercise 2

The elements of **T** and **U** can be expressed as

$$T_{ij} = \sum_{k=1}^{n} \lambda_k \, (e_i)_k \, (e_j)_k, \qquad U_{ij} = \sum_{l=1}^{n} \sigma_l \, (f_i)_l \, (f_j)_l \tag{168}$$

where $(e_i)_k$ and $(f_i)_k$ are the elements of eigenvectors $\hat{\mathbf{e}}_k$ and $\hat{\mathbf{f}}_k$, respectively. Using these two equations inserted in the right hand side of Equation (166), its equality with the left hand side follows immediately.

The formulation of the tensor scalar product according to Equation (167) follows immediately from the definition of the trace, Equation (17).

Solution to exercise 2

Using Equation (168), it follows that

$$\sum_{ij} T_{ij}\, U_{ij} = \sum_{ijkl} \lambda_k\, \sigma_l\, (e_i)_k\, (e_j)_k\, (f_i)_l\, (f_j)_l \tag{169}$$

The summation made over the indices i and j amounts to a scalar product between the vectors $\hat{\mathbf{e}}_k$ and $\hat{\mathbf{f}}_l$, which results in

$$\sum_{ij} T_{ij}\, U_{ij} = \sum_{kl} \lambda_k\, \sigma_l\, \langle\, \hat{\mathbf{e}}_k \mid \hat{\mathbf{f}}_l\, \rangle^2 = \langle\, \mathbf{T} \mid \mathbf{U}\, \rangle \tag{170}$$

where the last equality follows from the definition of the tensor scalar product, Equation (20).

Exercise 3

Consider a vector $\mathbf{x}$ and a basis $\{\mathbf{e}_k\}$. Then, $\mathbf{x}$ can be written as a linear combination of the basis vector, according to

$$\mathbf{x} = \sum_{i=1}^{n} x_i \, \mathbf{e}_i \tag{171}$$

where x_k are the corresponding coordinates. Show that

$$x_i = \langle \, \mathbf{x} \mid \tilde{\mathbf{e}}_i \, \rangle \tag{172}$$

where $\{\tilde{\mathbf{e}}_k\}$ is the dual basis relative to the original basis, characterized by the relation

$$\langle \, \mathbf{e}_i \mid \tilde{\mathbf{e}}_j \, \rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{173}$$

Hint Solution

Hint to exercise 3

Compute the right hand side of Equation (172) by substituting **x** for the right hand side of Equation (171), and apply Equation (173) on each term. The result is x_i

Solution to exercise 3

Substituting the right hand side of Equation (171) into the right hand side of Equation (172) results in

$$\langle \mathbf{x} \mid \tilde{\mathbf{e}}_i \rangle = \sum_j x_j \langle \mathbf{e}_j \mid \tilde{\mathbf{e}}_i \rangle \tag{174}$$

Using Equation (173) in each term in the sum gives

$$\langle \mathbf{x} \mid \tilde{\mathbf{e}}_i \rangle = \sum_j x_j \, \delta_{ij} \tag{175}$$

and since $\delta_{ij} = 1$ only when $i = j$, and zero otherwise, it follows immediately that

$$\langle \mathbf{x} \mid \tilde{\mathbf{e}}_i \rangle = x_i \tag{176}$$

which concludes the proof.

Exercise 4

Let $\{\mathbf{B}_k\}$ be a basis of tensors (or vectors). Show that the corresponding dual basis $\{\tilde{\mathbf{B}}_k\}$ is given by the expression

$$\tilde{\mathbf{B}}_l = \sum_k \mathbf{B}_k \, Q_{kl} \tag{177}$$

where $\mathbf{Q}^T$ is the matrix inverse of the matrix $\mathbf{P}$ defined as

$$P_{kl} = \langle \, \mathbf{B}_k \mid \mathbf{B}_l \, \rangle \tag{178}$$

Hint Solution

Hint to exercise 4

Show that the dual basis tensors $\mathbf{B}_l$, defined by Equation (177), satisfies the basic relation between a basis and its dual basis

$$\langle\, \mathbf{B}_j \mid \tilde{\mathbf{B}}_i \,\rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{179}$$

Solution to exercise 4

From Equation (177) it follows immediately that

$$\langle \mathbf{B}_i \mid \tilde{\mathbf{B}}_j \rangle = \sum_k \langle \mathbf{B}_i \mid \mathbf{B}_k \rangle \, Q_{jk} = \sum_k P_{ik} \, Q_{jk} \tag{180}$$

If Q_{jk} are the elements of the matrix $\mathbf{Q}$, and Q^T_{kj} are the elements of the matrix $\mathbf{Q}^T$, the transpose of $\mathbf{Q}$, then $Q_{jk} = Q^T_{kj}$. Hence,

$$\langle \mathbf{B}_i \mid \tilde{\mathbf{B}}_j \rangle = \sum_k P_{ik} \, Q^T_{kj} \tag{181}$$

But $\mathbf{Q}^T = \mathbf{P}^{-1}$, so

$$\langle \mathbf{B}_i \mid \tilde{\mathbf{B}}_j \rangle = \sum_k P_{ik} \, P^{-1}_{kj} \tag{182}$$

where P^{-1}_{kj} are the elements of the matrix $\mathbf{P}^{-1}$, the inverse of $\mathbf{P}$. A careful examination of the right hand side of the last equation reveals that it corresponds to a matrix multiplication between $\mathbf{P}$ and $\mathbf{P}^{-1}$, which by definition yields the identity matrix. Hence, the right hand side of this equation are the elements of the identity matrix, i.e., δ_{ij}. Therefore

$$\langle \mathbf{B}_j \mid \tilde{\mathbf{B}}_i \rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{183}$$

Exercise 5

Consider the tensor

$$\mathbf{D} = A\,(\hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T) \tag{184}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{v}}$ are normalized and perpendicular to each other. Show that $\hat{\mathbf{x}} + \hat{\mathbf{v}}$ and $\hat{\mathbf{x}} - \hat{\mathbf{v}}$ both are eigenvectors of $\mathbf{D}$, with eigenvalues A and $-A$, respectively. Show also that these are the only eigenvectors with non-zero eigenvalues (provided $A > 0$).

Hint Solution

Hint to exercise 5

Multiply $\mathbf{D}$ from left onto $\hat{\mathbf{x}} + \hat{\mathbf{v}}$ and $\hat{\mathbf{x}} - \hat{\mathbf{v}}$, respectively, and check that the result is a multiple ($\pm A$) of the original vector. Note that $\hat{\mathbf{x}}^T \hat{\mathbf{v}} = 0$, and that $\hat{\mathbf{x}}^T \hat{\mathbf{x}} = \hat{\mathbf{v}}^T \hat{\mathbf{v}} = 1$.

To show that $\pm A$ are the only non-zero eigenvalues, use the statement made in Section 2.5 regarding the fact that any symmetric tensor can be decomposed in terms of a linear combination of eigenvalues and eigenvectors. If it is true that $\pm A$ are the only non-zero eigenvalues, it it possible to decompose $\mathbf{D}$ in this way, using only the *normalized* eigenvectors

$$\frac{1}{\sqrt{2}} (\hat{\mathbf{x}} + \hat{\mathbf{v}}) \quad \text{with eigenvalue} \quad A \tag{185}$$

$$\frac{1}{\sqrt{2}} (\hat{\mathbf{x}} - \hat{\mathbf{v}}) \quad \text{with eigenvalue} \quad -A \tag{186}$$

Solution to exercise 5

Multiplying $\mathbf{D}$ from left onto $\hat{\mathbf{x}} + \hat{\mathbf{v}}$ and $\hat{\mathbf{x}} - \hat{\mathbf{v}}$, and observing the relations between the two vectors mentioned in the hints, gives

$$\begin{aligned} \mathbf{D}\,(\hat{\mathbf{x}} + \hat{\mathbf{v}}) &= A\,(\hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T)\,(\hat{\mathbf{x}} + \hat{\mathbf{v}}) = \\ &= A\,\left[\hat{\mathbf{x}}\,(\hat{\mathbf{v}}^T\hat{\mathbf{x}} + \hat{\mathbf{v}}^T\hat{\mathbf{v}}) + \hat{\mathbf{v}}(\hat{\mathbf{x}}^T\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\hat{\mathbf{v}})\right] = \\ &= A\,(\hat{\mathbf{x}} + \hat{\mathbf{v}}) \end{aligned} \tag{187}$$

and

$$\begin{aligned} \mathbf{D}\,(\hat{\mathbf{x}} - \hat{\mathbf{v}}) &= A\,(\hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T)\,(\hat{\mathbf{x}} - \hat{\mathbf{v}}) = \\ &= A\,\left[\hat{\mathbf{x}}\,(\hat{\mathbf{v}}^T\hat{\mathbf{x}} - \hat{\mathbf{v}}^T\hat{\mathbf{v}}) + \hat{\mathbf{v}}(\hat{\mathbf{x}}^T\hat{\mathbf{x}} - \hat{\mathbf{x}}^T\hat{\mathbf{v}})\right] = \\ &= A\,(-\hat{\mathbf{x}} + \hat{\mathbf{v}}) = -A\,(\hat{\mathbf{x}} - \hat{\mathbf{v}}) \end{aligned} \tag{188}$$

To show that $\pm A$ are the only non-zero eigenvalues, compute the linear combination of eigenvectors and eigenvalues, as suggested in the hints. This gives

$$A\,\frac{1}{\sqrt{2}}\,(\hat{\mathbf{x}} + \hat{\mathbf{v}})\,\frac{1}{\sqrt{2}}\,(\hat{\mathbf{x}} + \hat{\mathbf{v}})^T - A\,\frac{1}{\sqrt{2}}\,(\hat{\mathbf{x}} - \hat{\mathbf{v}})\,\frac{1}{\sqrt{2}}\,(\hat{\mathbf{x}} - \hat{\mathbf{v}})^T =$$

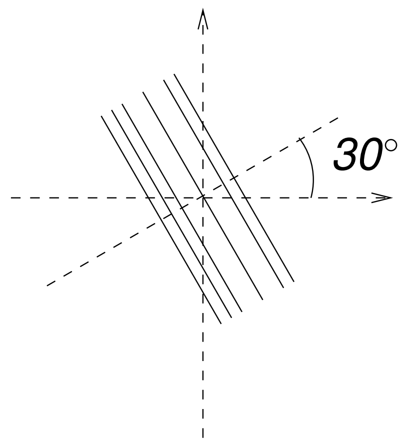
$$\frac{A}{2}(\hat{\mathbf{x}}\,\hat{\mathbf{x}}^T + \hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{v}}^T) - \frac{A}{2}(\hat{\mathbf{x}}\,\hat{\mathbf{x}}^T - \hat{\mathbf{x}}\,\hat{\mathbf{v}}^T - \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{v}}^T) =$$

$$A\,(\hat{\mathbf{x}}\,\hat{\mathbf{v}}^T + \hat{\mathbf{v}}\,\hat{\mathbf{x}}^T) = \mathbf{D}$$

Since $\mathbf{D}$ can be decomposed in this way it must be the case that $\pm A$ are the only non-zero eigenvalues of $\mathbf{D}$.

Exercise 6

Consider the two-dimensional linear structure in the figure below, corresponding to some local neighborhood in an image. The dashed lines indicate a local coordinate system of the neighborhood, and a line perpendicular to the linear structure. What is the orientation tensor of this neighborhood, described as a matrix containing numerical values?



Hint Solution

Hint to exercise 6

Clearly, the neighborhood is simple since it is constant on parallel lines. To find the orientation tensor, a (normalized) normal vector to the parallel lines must first be established. Use the coordinates defined by the coordinate system presented in the figure. Note that in this case, the normal vector is well-defined except for the sign. The orientation tensor is then defined as proportional to the outer product between the normal vector and itself. Therefore, which sign is chosen is irrelevant.

Solution to exercise 6

As a normal vector we can use

$$\hat{\mathbf{x}} = \begin{pmatrix} \cos 30^\circ \\ \sin 30^\circ \end{pmatrix} \approx \begin{pmatrix} 0.8660 \\ 0.5000 \end{pmatrix} \tag{189}$$

The orientation tensor **T** is then given by

$$\begin{aligned} \mathbf{T} &= A\,\hat{\mathbf{x}}\,\hat{\mathbf{x}}^T = \\ &= A \begin{pmatrix} \cos^2 30^\circ & \cos 30^\circ \sin 30^\circ \\ \cos 30^\circ \sin 30^\circ & \sin^2 30^\circ \end{pmatrix} \approx \\ &\approx A \begin{pmatrix} 0.7500 & 0.4330 \\ 0.4330 & 0.2500 \end{pmatrix} \end{aligned} \tag{190}$$

A is any positive real number.

Exercise 7

Consider the tensor

$$\mathbf{T} = \begin{pmatrix} 1.4510 & 0.6010 \\ 0.6010 & 0.2490 \end{pmatrix} \tag{191}$$

Which orientation does **T** represent? Make an illustration of the corresponding simple neighborhood.

Hint Solution

Hint to exercise 7

Compute the eigenvalues λ_1 and λ_2, and corresponding eigenvectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$. According to the conventions used here, the eigenvalues are labeled according to $\lambda_1 \geq \lambda_2$. Note that $\lambda_2 \approx 0$, so $\mathbf{T}$ is (approximately) a rank one tensor.

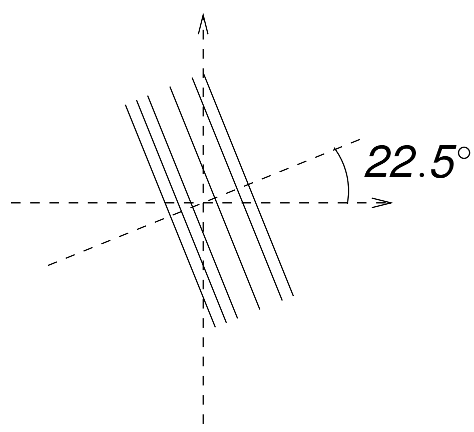
The orientation of the neighborhood is given by the eigenvector $\hat{\mathbf{e}}_1$. It is an normal vector to the corresponding linear structure.

Solution to exercise 7

The eigenvalues and eigenvectors of **T** are given by

$$\begin{aligned} \lambda_1 &\approx 1.6999 \qquad \hat{\mathbf{e}}_1 = \begin{pmatrix} 0.9239 \\ 0.3827 \end{pmatrix} \approx \begin{pmatrix} \cos 22.5^\circ \\ \sin 22.5^\circ \end{pmatrix} \\ \lambda_2 &\approx 0.0001 \qquad \hat{\mathbf{e}}_2 = \begin{pmatrix} 0.3827 \\ -0.9239 \end{pmatrix} \end{aligned} \tag{192}$$

The ratio between λ_1 and λ_2 suggests that **T** can be treated as a rank one tensor. The eigenvector $\hat{\mathbf{e}}_1$ is then a normal vector to the corresponding linear structure of the neighborhood. The figure below illustrates the characteristic properties regarding the orientation of the neighborhood. The variation *across* the parallel lines, however, is not represented by **T**, and therefore indeterminable.



Exercise 8

Show that the Fourier transform S of a simple function s,

$$s(\xi) = g(\xi^T \hat{\mathbf{x}}) \tag{193}$$

is concentrated to an impulse line in the Fourier domain. The line passes through the origin and it goes in the direction of $\hat{\mathbf{x}}$, and the variation along the impulse line is given by G, the one-dimensional Fourier transform of g.

To simplify the exercise, consider first the 2D case, and then extend the result to higher dimensions.

The Fourier transform of s is given by the expression

$$S(\mathbf{u}) = \int_{\mathbb{R}^n} s(\xi)\, e^{-i\mathbf{u}^T \xi}\, d\xi^n \tag{194}$$

The integration is made over the entirety of the spatial domain.

Hint Solution

Hint to exercise 8

Introduce a coordinate system corresponding to an ON-basis in the spatial domain, and express *s* in terms of the resulting coordinates, rather than in terms of ξ. The function *s* can then be rotated such that the first coordinate axis points in the direction of $\hat{\mathbf{x}}$, resulting in the function s'. Note that when *s* is rotated into s', the same rotation applies also to its Fourier transform.

Consider then the Fourier transform of s', and note that it is Cartesian separable into a product of *n* one-variable functions. One of these is *g*, and the remaining $n - 1$ are constant functions (= *1*). The Fourier transform of s' therefore decomposes into a product of *n* one-variable transforms, one is a transform of *g* and the remaining ones are of the constant function "1". Thus, the resulting product is indeed an impulse line, passing through the origin and in the direction of the first coordinate axis in the Fourier domain. The variation along the impulse line is described by *G*, the Fourier transform of *g*.

The above describes the Fourier transform of s', a rotated version of *s*. To get *S*, simply make the inverse rotation relative to the first one, and the result is that the impulse line now is oriented in the direction of $\hat{\mathbf{x}}$.

Solution to exercise 8

Using the hints, the function s' can be expressed as

$$s'(\xi_1,\xi_2) = g(\xi_1) \tag{195}$$

in the 2D case. Its Fourier transform is then given by

$$S'(\xi_1,\xi_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\xi_1)\, e^{-i(u_1\xi_1+u_2\xi_1)}\, d\xi_1\, d\xi_2 \tag{196}$$

Since s' is Cartesian separable, its transform can be decomposed into two factors, each corresponding to a one-dimensional transform.

$$\begin{aligned} S'(\xi_1,\xi_2) &= \int_{-\infty}^{\infty} g(\xi_1)\, e^{-iu_1\xi_1}\, d\xi_1 \int_{-\infty}^{\infty} 1\, e^{-iu_2\xi_2}\, d\xi_2 \\ &= G(u_1)\, 2\pi\, \delta(u_2) \end{aligned} \tag{197}$$

Thus, S' is an impulse line which lies on the u_1 axes in the Fourier domain. The variation along the impulse line is described by G. S' is related to S by the rotation that takes the $\hat{\mathbf{x}}$ vector to the ξ_1 axis, so S is obtained by the inverse rotation. This means that S is an impulse line oriented in the direction of $\hat{\mathbf{x}}$. The extension to higher dimensions is straightforward.

Exercise 9

Show that the dual basis relative to the three tensors in Equation (90) are the tensors presented in Equation (91).

Hint Solution

Hint to exercise 9

Refer to Section 2.8, in particular the last part which discusses how to compute the dual bases relative to a given basis, Equations (35) and (36). See also the related exercise. According to these results, the dual basis is found by first computing $\mathbf{P}$, the matrix which contains scalar product between all pairs of basis tensors, then invert $\mathbf{P}$ to get $\mathbf{Q}^T$, then use $\mathbf{Q}$ in a linear combination between the basis tensors.

Solution to exercise 9

Referring to the steps presented in the hints, first compute **P**. This results in

$$\mathbf{P} = \frac{1}{4}\begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \tag{198}$$

Hence, $\mathbf{Q}^T$ is given by

$$\mathbf{Q}^T = \mathbf{P}^{-1} = \frac{1}{9}\begin{pmatrix} 10 & -2 & -2 \\ -2 & 10 & -2 \\ -2 & -2 & 10 \end{pmatrix} \tag{199}$$

The elements of **Q** are now to be used in a linear combination of the basis tensors. For example,

$$\tilde{\mathbf{N}}_1 = Q_{11}\,\hat{\mathbf{N}}_1 + Q_{21}\,\hat{\mathbf{N}}_2 + Q_{31}\,\hat{\mathbf{N}}_3 \tag{200}$$

which results in

$$\tilde{\mathbf{N}}_1 = \begin{pmatrix} 1 & 0 \\ 0 & -\frac{3}{4} \end{pmatrix} \tag{201}$$

The other two dual tensors are computed in the corresponding way.

Exercise 10

Show that the dual frame relative to the tensors in Equation (94) are the tensors presented in Equation (95).

Hint Solution

Hint to exercise 10

Refer to Section 2.9, in particular the last part which discusses how to compute the dual frame relative to a given frame, Equation (38). According to these results, the dual frame is found by first constructing $\mathbf{F}$, the frame operator, which contains the frame tensors in its rows. The dual frame operator is then given by the expression

$$\tilde{\mathbf{F}} = \mathbf{F}\,(\mathbf{F}^T\mathbf{F})^{-1} \tag{202}$$

The rows of $\tilde{\mathbf{F}}$ contain the dual frame tensors.

Solution to exercise 10

Using the recipe presented in Section 2.9, we start by forming the frame operator. Each row of the frame operator corresponds to the elements of one frame tensor. For example, the first two elements of a row may correspond to the diagonal elements of the frame tensors, and the third element may correspond to the off-diagonal elements. Note, however, that the latter must be multiplied by $\sqrt{2}$. Hence, the following mapping from frame tensor $\hat{\mathbf{N}}$ to a row of $\mathbf{F}$ is being used

$$\begin{pmatrix} N_{11} & N_{22} & \sqrt{2}\,N_{12} \end{pmatrix} \tag{203}$$

If the k-th row of $\mathbf{F}$ represent $\hat{\mathbf{N}}_k$, this results in

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ 1/2 & 1/2 & -1/\sqrt{2} \end{pmatrix} \tag{204}$$

This gives

$$\mathbf{F}^T\mathbf{F} = \begin{pmatrix} 3/2 & 1/2 & 0 \\ 1/2 & 3/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{205}$$

and

$$(\mathbf{F}^T\mathbf{F})^{-1} = \begin{pmatrix} 3/4 & -1/4 & 0 \\ -1/4 & 3/4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{206}$$

Thus, the dual frame operator is

$$\tilde{\mathbf{F}} = \mathbf{F}\,(\mathbf{F}^T\mathbf{F})^{-1} =$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ 1/2 & 1/2 & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 3/4 & -1/4 & 0 \\ -1/4 & 3/4 & 0 \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 3/4 & -1/4 & 0 \\ 1/4 & 1/4 & 1/\sqrt{2} \\ -1/4 & 3/4 & 0 \\ 1/4 & 1/4 & -1/\sqrt{2} \end{pmatrix} \quad (207)$$

The rows of $\tilde{\mathbf{F}}$ describe the dual frame tensors in the same way as the rows of $\mathbf{F}$ describe the frame tensors. Hence, the dual frame tensors of Equation (95) follows immediately.

Exercise 11

Show that the dual basis relative to the three tensors in Equation (103) are the tensors presented in Equation (105).

Hint Solution

Hint to exercise 11

Refer to Section 2.8, in particular the last part which discusses how to compute the dual bases relative to a given basis, Equations (35) and (36). See also the related exercise. According to these results, the dual basis is found by first computing $\mathbf{P}$, the matrix which contains scalar product between all pairs of basis tensors, then invert $\mathbf{P}$ to get $\mathbf{Q}^T$, then use $\mathbf{Q}$ in a linear combination between the basis tensors.

Solution to exercise 11

Referring to the steps presented in the hints, first compute **P**. This results in

$$\mathbf{P} = \frac{1}{5}\begin{pmatrix} 5 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 5 & 1 & 1 & 1 \\ 1 & 1 & 1 & 5 & 1 & 1 \\ 1 & 1 & 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 1 & 1 & 5 \end{pmatrix} \tag{208}$$

Hence, $\mathbf{Q}^T$ is given by

$$\mathbf{Q}^T = \mathbf{P}^{-1} = \frac{1}{8}\begin{pmatrix} 9 & -1 & -1 & -1 & -1 & -1 \\ -1 & 9 & -1 & -1 & -1 & -1 \\ -1 & -1 & 9 & -1 & -1 & -1 \\ -1 & -1 & -1 & 9 & -1 & -1 \\ -1 & -1 & -1 & -1 & 9 & -1 \\ -1 & -1 & -1 & -1 & -1 & 9 \end{pmatrix} \tag{209}$$

The elements of **Q** are now to be used in a linear combination of the basis tensors. For example,

$$\tilde{\mathbf{N}}_1 = Q_{11}\hat{\mathbf{N}}_1 + Q_{21}\hat{\mathbf{N}}_2 + Q_{31}\hat{\mathbf{N}}_3 + Q_{41}\hat{\mathbf{N}}_4 + Q_{51}\hat{\mathbf{N}}_5 + Q_{61}\hat{\mathbf{N}}_6 \tag{210}$$

which results in

$$\tilde{\mathbf{N}}_1 = \begin{pmatrix} a^2 - b^2/4 & 0 & 5ab/4 \\ 0 & -(a^2 + b^2)/4 & 0 \\ 5ab/4 & 0 & b^2 - a^2/4 \end{pmatrix} \tag{211}$$

Setting

$$c = a^2 - b^2/4 \tag{212}$$

$$d = b^2 - a^2/4 \tag{213}$$

$$e = 5\,a\,b/4 \tag{214}$$

$$f = -(a^2 + b)/4 \tag{215}$$

and computing the other five dual tensors in the corresponding way gives the dual tensor presented in Equation (105).

Exercise 12

Show that the dual frame relative to the tensors in Equation (109) are the tensors presented in Equation (110).

Hint Solution

Hint to exercise 12

Refer to Section 2.9, in particular the last part which discusses how to compute the dual frame relative to a given frame, Equation (38). According to these results, the dual frame is found by first constructing $\mathbf{F}$, the frame operator, which contains the frame tensors in its rows. The dual frame operator is then given by the expression

$$\tilde{\mathbf{F}} = \mathbf{F}\,(\mathbf{F}^T\mathbf{F})^{-1} \tag{216}$$

The rows of $\tilde{\mathbf{F}}$ contain the dual frame tensors.

Solution to exercise 12

Using the recipe presented in Section 2.9, we start by forming the frame operator. Each row of the frame operator corresponds to the elements of one frame tensor. For example, the following mapping from elements of a frame tensor $\hat{\mathbf{N}}$ to a row of $\mathbf{F}$ can be used.

$$\begin{pmatrix} N_{11} & N_{22} & N_{33} & N_{44} & \sqrt{2}\,N_{12} & \sqrt{2}\,N_{23} & \sqrt{2}\,N_{34} & \sqrt{2}\,N_{13} & \sqrt{2}\,N_{24} & \sqrt{2}\,N_{14} \end{pmatrix} \tag{217}$$

Let the k-th row represent $\hat{\mathbf{N}}_k$. Hence,

$$\mathbf{F} = \frac{1}{2}\begin{pmatrix}
1 & 1 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & -\sqrt{2} & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & -\sqrt{2} & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\sqrt{2} \\
0 & 1 & 1 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -\sqrt{2} & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & -\sqrt{2} & 0 & 0 & 0
\end{pmatrix} \tag{218}$$

This gives

$$\mathbf{F}^T\mathbf{F} = \frac{1}{2}\begin{pmatrix} 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{219}$$

and

$$
(\mathbf{F}^T\mathbf{F})^{-1} = \frac{1}{6}\begin{pmatrix}
5 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 5 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 5 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6
\end{pmatrix} \qquad (220)
$$

Thus, the dual frame operator is

$$\tilde{\mathbf{F}} = \mathbf{F}\left(\mathbf{F}^T\mathbf{F}\right)^{-1} =$$

$$= \frac{1}{6}\begin{pmatrix} 2 & 2 & -1 & -1 & \sqrt{18} & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & -1 & -1 & -\sqrt{18} & 0 & 0 & 0 & 0 & 0 \\ 2 & -1 & 2 & -1 & 0 & 0 & 0 & \sqrt{18} & 0 & 0 \\ 2 & -1 & 2 & -1 & 0 & 0 & 0 & -\sqrt{18} & 0 & 0 \\ 2 & -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & \sqrt{18} \\ 2 & -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & -\sqrt{18} \\ -1 & 2 & 2 & -1 & 0 & \sqrt{18} & 0 & 0 & 0 & 0 \\ -1 & 2 & 2 & -1 & 0 & -\sqrt{18} & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 2 & 0 & 0 & 0 & 0 & \sqrt{18} & 0 \\ -1 & 2 & -1 & 2 & 0 & 0 & 0 & 0 & -\sqrt{18} & 0 \\ -1 & -1 & 2 & 2 & 0 & 0 & \sqrt{18} & 0 & 0 & 0 \\ -1 & -1 & 2 & 2 & 0 & 0 & -\sqrt{18} & 0 & 0 & 0 \end{pmatrix} \quad (221)$$

The rows of $\tilde{\mathbf{F}}$ describe the dual frame tensors in the same way as the rows of $\mathbf{F}$ describe the frame tensors, the k-th row of $\tilde{\mathbf{F}}$ represents $\tilde{\mathbf{N}}_k$. Hence, the dual frame tensors of Equation (95) follows immediately.

Exercise 13

Let $\mathbf{T}$ be an orientation tensor which has been estimated, e.g., using the procedures presented in Section 4. For a general neighborhood, then, $\mathbf{T}$ may not be a rank one tensor. Let $\mathbf{T}_s$ be a rank tensor

$$\mathbf{T}_s = \lambda\, \hat{\mathbf{e}}\, \hat{\mathbf{e}}^T \tag{222}$$

Show that any rank one tensor $\mathbf{T}_s$ which minimized the error

$$\varepsilon = \|\mathbf{T} - \mathbf{T}_s\|^2 \tag{223}$$

is constructed in such a way that λ_1 is the largest eigenvalue of $\mathbf{T}$, and $\hat{\mathbf{e}}_1$ is a corresponding normalized eigenvector, i.e., $\lambda = \lambda_1$ and $\hat{\mathbf{e}} = \hat{\mathbf{e}}_1$.

Hint Solution

Hint to exercise 13

The task is to minimize

$$\varepsilon = \|\mathbf{T} - \lambda\, \hat{\mathbf{e}}\, \hat{\mathbf{e}}^T\|^2 \tag{224}$$

over all λ and all normalized $\hat{\mathbf{e}}$. Using Equation (26), this expands to

$$\varepsilon = \|\mathbf{T}\|^2 - 2\lambda\, \hat{\mathbf{e}}^T \mathbf{T}\, \hat{\mathbf{e}} + \lambda^2 \tag{225}$$

This is a standard minimization problem with an additional constraint which is solved using multi-variable calculus, i.e., by considering gradients of ε with respect to λ and $\hat{\mathbf{e}}$. Start by minimizing ε with respect to $\hat{\mathbf{e}}$. The constraint defines a compact set and, therefore, ε indeed takes a minimal (as well as a maximal) value for at least one vector $\hat{\mathbf{e}}$. At any such point (corresponding to a local minimum or maximum), the gradient of ε with respect to $\hat{\mathbf{e}}$ is parallel to the gradient of the constraint. This gives a relation between $\mathbf{T}$ and $\hat{\mathbf{e}}$, with turns out to state that $\hat{\mathbf{e}}$ must be an eigenvector of $\mathbf{T}$. It is then straightforward to show that $\hat{\mathbf{e}}$ should be chosen to be $\hat{\mathbf{e}}_1$, an eigenvector of $\mathbf{T}$ with eigenvalue λ_1 which is the largest of all eigenvalues.

Solution to exercise 13

Using the hints, compute the gradient of ε with respect to $\hat{\mathbf{e}}$

$$\frac{d\varepsilon}{d\hat{\mathbf{e}}} = -4\lambda\, \mathbf{T}\, \hat{\mathbf{e}} \tag{226}$$

and of the constraint

$$c = \hat{\mathbf{e}}^T\, \hat{\mathbf{e}} = 1 \tag{227}$$

which gives

$$\frac{dc}{d\hat{\mathbf{e}}} = 2\, \hat{\mathbf{e}} \tag{228}$$

The two gradients are parallel at any local minimum, thus we seek $\hat{\mathbf{e}}$ which satisfies the relation

$$-4\lambda\, \mathbf{T}\, \hat{\mathbf{e}} = 2\gamma\, \hat{\mathbf{e}} \tag{229}$$

where γ is the corresponding Lagrange factor. According to this relation, it is either the case that $\lambda = \gamma = 0$, and then $\hat{\mathbf{e}}$ is any normalized vector, or the case that $\lambda \neq 0$, and then $\hat{\mathbf{e}}$ is an eigenvector of $\mathbf{T}$. Consider first the latter case. We can write

$$\mathbf{T} = \sum_i \lambda_i\, \hat{\mathbf{e}}_i\, \hat{\mathbf{e}}_i^T \tag{230}$$

where $\{\lambda_i\}$ are the eigenvalues of $\mathbf{T}$ and $\{\hat{\mathbf{e}}_i\}$ the corresponding eigenvalues. We know that $\hat{\mathbf{e}} = \hat{\mathbf{e}}_k$ for some k, but not which k. The difference between $\mathbf{T}$ and $\mathbf{T}_s$ can now the written

$$\mathbf{T} - \mathbf{T}_s = (\lambda_k - \lambda)\, \hat{\mathbf{e}}_k\, \hat{\mathbf{e}}_k^T + \sum_{i \neq k} \lambda_i\, \hat{\mathbf{e}}_i\, \hat{\mathbf{e}}_i^T \tag{231}$$

Evidently, this difference has the same eigenvectors as **T** has, the only difference is that eigenvalue λ_k of **T** corresponds to $\lambda_k - \lambda$ for $\mathbf{T} - \mathbf{T}_s$. Using Equation (24), this gives

$$\varepsilon = \|\mathbf{T} - \mathbf{T}_s\|^2 = (\lambda_k - \lambda)^2 + \sum_{i \neq k} \lambda_i^2 \tag{232}$$

Clearly, this sum is minimized if we choose $\lambda_k = \lambda_1 =$ the largest eigenvalue of **T**, and $\lambda = \lambda_1$. Consequently, it must be the case that $\hat{\mathbf{e}} = \hat{\mathbf{e}}_1 =$ the eigenvector corresponding to the largest eigenvalue of **T**. The minimal value of ε is then

$$\varepsilon_0 = \sum_{i \neq 1} \lambda_i^2 \tag{233}$$

Finally, observe that the case $\lambda = 0$, leads to $\varepsilon = \|\mathbf{T}\|^2 \geq \varepsilon_0$, so this case does not necessary correspond to a minimum.

It should be noted that, provided that $\lambda_1 > \lambda_2$, i.e., the largest eigenvalue of **T** is distinct, $\hat{\mathbf{e}}$ can be chosen as $\pm\hat{\mathbf{e}}_1$, but the resulting $\mathbf{T}_s$ does not depend on this choice. However, in the case that $\lambda_1 = \lambda_2$, i.e., there are two or more eigenvalues of **T** which are the largest ones, then the choice of $\hat{\mathbf{e}}$ is not just a matter of sign, since the subspace of corresponding eigenvectors is of dimensionality ≥ 2. One one hand, then, any normalized vector $\hat{\mathbf{e}}$ in this subspace minimizes ε. On the other hand, the problem of finding the rank one approximation of **T** becomes less interesting in that case.

Exercise 14

The following four orientation tensors have been estimated for different neighborhoods of a 3D volume. Classify each of the tensors as describing a planar structure, a linear structure, or an isotropic structure, or if it cannot reasonably be said to describe either of these three cases. In the planar and linear case, what is the orientation of the plane or lines?

$$\mathbf{T}_1 = \begin{pmatrix} 3.2098 & 1.4585 & 0.7406 \\ 1.4585 & 1.6950 & 0.3319 \\ 0.7406 & 0.3319 & 0.5453 \end{pmatrix} \tag{234}$$

$$\mathbf{T}_2 = \begin{pmatrix} 3.8629 & -0.0125 & -0.0681 \\ -0.0125 & 3.1576 & 1.0858 \\ -0.0681 & 1.0858 & 1.1095 \end{pmatrix} \tag{235}$$

$$\mathbf{T}_3 = \begin{pmatrix} 2.4460 & 0.0133 & -0.0252 \\ 0.0133 & 2.3255 & -0.0406 \\ -0.0252 & -0.0406 & 2.0985 \end{pmatrix} \tag{236}$$

$$\mathbf{T}_4 = \begin{pmatrix} 2.7913 & -0.4909 & 0.4134 \\ -0.4909 & 1.9154 & 0.2958 \\ 0.4134 & 0.2958 & 1.2133 \end{pmatrix} \tag{237}$$

Hint Solution

Hint to exercise 14

Compute eigenvalues and eigenvectors of each of the tensors. From the eigenvalues, compute the three measures c_1,c_2, and c_3, defined in Section 6.3.4. If a tensor is decided to belong to either the plane case or the line case, consider the appropriate eigenvector to get the corresponding orientation, see Sections 6.3.1 and 6.3.2.

Solution to exercise 14

The eigenvalues of $\mathbf{T}_1$ are

$$\lambda_1 \approx 4.27 \qquad \lambda_2 \approx 0.83 \qquad \lambda_3 \approx 0.35 \tag{238}$$

The three measures are

$$c_1 \approx 0.81 \qquad c_2 \approx 0.11 \qquad c_3 \approx 0.08 \tag{239}$$

Since c_1 is so much larger than the other two we can assume that this tensor represents a plane case, i.e., a neighborhood that is constant on parallel planes. The normal of the planes is given by $\hat{\mathbf{e}}_1$, a normalized eigenvector corresponding to λ_1 = the largest eigenvalue. In this case,

$$\hat{\mathbf{e}}_1 \approx \pm \begin{pmatrix} 0.8385 \\ 0.5022 \\ 0.2115 \end{pmatrix} \tag{240}$$

The eigenvalues of $\mathbf{T}_2$ are

$$\lambda_1 \approx 3.87 \qquad \lambda_2 \approx 3.62 \qquad \lambda_3 \approx 0.64 \tag{241}$$

The three measures are

$$c_1 \approx 0.06 \qquad c_2 \approx 0.77 \qquad c_3 \approx 0.17 \tag{242}$$

Here, it is reasonable to assume that the tensor represent a line case, i.e., a neighborhood that is constant on parallel lines. The orientation of the lines is given by $\hat{\mathbf{e}}_3$, a normalized eigenvector corresponding to λ_3 = the smallest eigenvalue.

In this case

$$\hat{\mathbf{e}}_3 \approx \pm \begin{pmatrix} 0.0179 \\ -0.3959 \\ 0.9181 \end{pmatrix} \tag{243}$$

The eigenvalues of $\mathbf{T}_3$ are

$$\lambda_1 \approx 2.45 \qquad \lambda_2 \approx 2.33 \qquad \lambda_3 \approx 2.09 \tag{244}$$

The three measures are

$$c_1 \approx 0.05 \qquad c_2 \approx 0.10 \qquad c_3 \approx 0.85 \tag{245}$$

In this case, the measure of isotropy (c_3) is much larger than the other two, so we can assume that the corresponding neighborhood (from any practical point of view) is isotropic. The isotropy implies that the neighborhood does not exhibit any particular orientation, so no orientation needs to be indicated.

The eigenvalues of $\mathbf{T}_4$ are

$$\lambda_1 \approx 3.05 \qquad \lambda_2 \approx 1.94 \qquad \lambda_3 \approx 0.93 \tag{246}$$

The three measures are

$$c_1 \approx 0.36 \qquad c_2 \approx 0.33 \qquad c_3 \approx 0.30 \tag{247}$$

The three measures are here more or less equal and, therefore, it is not reasonable to classify the tensor as belonging to either of the plane, line, or isotropy cases. Instead, we leave the tensor unclassified. The practical consequence of this choice is that this tensor (or others like it) will not take part in further analysis or measurements on the image data. It should be noted that these tensors are not anomalies, but are the result of applying estimation procedures which assume specific models of the local image data (e.g., that it is simple) where not every neighborhood fits the model.

Exercise 15

Let $\mathbf{T}_1$ and $\mathbf{T}$ be two orientation tensors such that $\mathbf{e}$ is an eigenvector with eigenvalue = 0 relative to both tensors. Show that the sum $\mathbf{T}_a = \mathbf{T}_1 + \mathbf{T}_2$ inherits this property, i.e., $\mathbf{e}$ is an eigenvector with eigenvalue = 0 also relative to $\mathbf{T}_a$.

Show that if both $\mathbf{T}_1$ and $\mathbf{T}_2$ are rank one tensors, then $\mathbf{T}_a = \mathbf{T}_1 + \mathbf{T}_2$ is of rank one if and only if $\mathbf{T}_1$ and $\mathbf{T}_2$ share the same eigenvector with non-zero eigenvalue.

Hint Solution

Hint to exercise 15

The first task is simple: apply $\mathbf{T}_a$ onto $\mathbf{e}$ and verify that the result is the zero vector.

Since $\mathbf{T}_1$ and $\mathbf{T}_2$ are rank one tensor they can be written

$$\mathbf{T}_1 = \lambda\, \hat{\mathbf{e}}\, \hat{\mathbf{e}}^T \tag{248}$$

and

$$\mathbf{T}_2 = \sigma\, \hat{\mathbf{f}}\, \hat{\mathbf{f}}^T \tag{249}$$

The second assertion to prove states that $\mathbf{T}_a = \mathbf{T}_1 + \mathbf{T}_2$ is too a rank one tensor if and only if $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are parallel, otherwise it is of rank two. Note that it is straightforward to show that $\mathbf{T}_a$ is of rank one if $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are parallel. To prove the second part of the assertion, assume that $\mathbf{T}_a$ is of rank one, i.e.,

$$\mathbf{T}_a = \gamma\, \hat{\mathbf{g}}\, \hat{\mathbf{g}}^T \tag{250}$$

and that $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are not parallel. From this follows that $\mathbf{T}_a\, \mathbf{h} = \alpha\mathbf{h}$, i.e., the result of applying the linear mapping $\mathbf{T}_a$ onto an arbitrary vector $\mathbf{h}$ is always a multiple of $\hat{\mathbf{g}}$. In particular this is true if $\mathbf{h}$ is either $\hat{\mathbf{e}}$ or $\hat{\mathbf{f}}$. On the other hand, $\mathbf{T}_a = \mathbf{T}_1 + \mathbf{T}_2$, so the result can also be written as a linear combination of $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$. Analyzing the relations that come from these considerations leads to the conclusion that $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ in fact are parallel, a contraction to the initial condition.

Solution to exercise 15

It follows immediately that

$$\mathbf{T}_a\,\mathbf{e} = (\mathbf{T}_1 + \mathbf{T}_2)\,\mathbf{e} = \mathbf{T}_1\,\mathbf{e} + \mathbf{T}_2\,\mathbf{e} = \mathbf{0} + \mathbf{0} = \mathbf{0} \tag{251}$$

Hence, $\mathbf{e}$ is an eigenvector of $\mathbf{T}_a$ with corresponding eigenvalue = 0.

Refer to the notation presented in the hints, and note that it is safe to assume that both λ and σ are non-zero. To show that $\mathbf{T}_a$ is of rank one only if $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are parallel, assume that $\mathbf{T}_a$ is of rank one and that $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are not parallel and show that this leads to a contradiction.

From the assumption it follows that

$$\mathbf{T}_a\,\hat{\mathbf{e}} = \left[\gamma\,\hat{\mathbf{g}}\,\hat{\mathbf{g}}^T\right]\,\hat{\mathbf{e}} = \gamma\,(\hat{\mathbf{g}}^T\hat{\mathbf{e}})\,\hat{\mathbf{g}} \tag{252}$$

$$\mathbf{T}_a\,\hat{\mathbf{f}} = \left[\gamma\,\hat{\mathbf{g}}\,\hat{\mathbf{g}}^T\right]\,\hat{\mathbf{f}} = \gamma\,(\hat{\mathbf{g}}^T\hat{\mathbf{f}})\,\hat{\mathbf{g}} \tag{253}$$

which shows that $\mathbf{T}_a\hat{\mathbf{e}}$ and $\mathbf{T}_a\hat{\mathbf{f}}$ are parallel. On the other hand, these two vectors can also be expressed as

$$\mathbf{T}_a\,\hat{\mathbf{e}} = \left[\lambda\,\hat{\mathbf{e}}\,\hat{\mathbf{e}}^T + \sigma\,\hat{\mathbf{f}}\,\hat{\mathbf{f}}^T\right]\,\hat{\mathbf{e}} = \lambda\,\hat{\mathbf{e}} + \sigma\,(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}})\,\hat{\mathbf{f}} \tag{254}$$

$$\mathbf{T}_a\,\hat{\mathbf{f}} = \left[\lambda\,\hat{\mathbf{e}}\,\hat{\mathbf{e}}^T + \sigma\,\hat{\mathbf{f}}\,\hat{\mathbf{f}}^T\right]\,\hat{\mathbf{f}} = \lambda\,(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}})\,\hat{\mathbf{e}} + \sigma\,\hat{\mathbf{f}} \tag{255}$$

They are parallel, i.e., $\mathbf{T}_a\,\hat{\mathbf{e}} = \kappa\,\mathbf{T}_a\,\hat{\mathbf{f}}$, whereas $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are not, which implies

$$\begin{cases} \lambda & = \kappa\,\lambda\,(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}}) \\ \sigma\,(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}}) & = \kappa\,\sigma \end{cases} \tag{256}$$

Since λ and σ are non-zero, these two equations can be rewritten as

$$\begin{cases} 1 & = \kappa\,(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}}) \\ (\hat{\mathbf{e}}^T\,\hat{\mathbf{f}}) & = \kappa \end{cases} \tag{257}$$

leading to

$$(\hat{\mathbf{e}}^T\,\hat{\mathbf{f}})^2 = 1 \tag{258}$$

Both $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are normalized, so the only way for this equation to be true is for $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ to be parallel. This a contradicts the initial conditions, and the only way to resolve the contradiction is to accept that $\mathbf{T}_a$ is of rank one only if $\hat{\mathbf{e}}$ and $\hat{\mathbf{f}}$ are parallel.

Exercise 16

An image sequence, represented as a 3D volume, has been processed to produce a 3D orientation tensor at each point of the volume. The numerical values of the orientation tensor is sampled at four different points in the volume, as presented in exercise 14. Using the assumption that the image volume represents an image sequence, what type of local motion is described by each of the four tensors, and what is the corresponding velocity.

Note that not all four tensors can give an estimate of the true local velocity, or not even an estimate of any type of velocity.

Hint Solution

Hint to exercise 16

Refer to Section 7.3. According to this ”recipe”, you need first the eigenvalues and eigenvectors for each of the tensors. Each tensor is then classified as describing either the plane case, the line case, or neither of the two. This has already been done in solution 14. Depending on which case, a normal velocity, a true velocity, or no velocity can be estimated.

Solution to exercise 16

Tensor $\mathbf{T}_1$ represents the plane case (rank one) which corresponds to a moving linear structure, e.g., a moving line or edge. In this case only the normal velocity relative to the linear structure can be determined. It is given by $\hat{\mathbf{e}}_1$, the eigenvector of $\mathbf{T}_1$ with largest corresponding eigenvalue. Here,

$$\hat{\mathbf{e}}_1 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \pm \begin{pmatrix} 0.8385 \\ 0.5022 \\ 0.2115 \end{pmatrix} \tag{259}$$

According to Equation (150), the normal velocity vector $\tilde{\mathbf{v}}$ then has the following value

$$\tilde{\mathbf{v}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = -\frac{x_3}{x_1^2 + x_2^2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \approx \begin{pmatrix} 0.19 \\ 0.11 \end{pmatrix} \tag{260}$$

Hence, the normal velocity of the linear structure is

$$\approx \; (0.19 \;\; 0.11) \tag{261}$$

spatial units (in the 2D image) per time unit. In practice, the spatial units are normally measured in pixels, and the corresponding time unit is the time between each consecutive image in the sequence.

Note that $\tilde{\mathbf{v}}$, apart from representing local motion, also describes local orientation since it is a normal vector to the local linear structure.

Tensor $\mathbf{T}_2$ represents the line case (rank two) which corresponds to a moving point. In this case, an estimate of the true velocity can be determined. It is given by $\hat{\mathbf{e}}$, the eigenvector corresponding to the eigenvector of $\mathbf{T}_2$ with smallest corresponding eigenvalue. Here,

$$\hat{\mathbf{e}}_3 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \pm \begin{pmatrix} 0.0179 \\ -0.3959 \\ 0.9181 \end{pmatrix} \tag{262}$$

According to Equation (152), the true velocity vector $\tilde{\mathbf{v}}$ has the following value.

$$\tilde{\mathbf{v}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \approx \begin{pmatrix} 0.02 \\ -0.43 \end{pmatrix} \tag{263}$$

The tensors $\mathbf{T}_3$ and $\mathbf{T}_4$ belongs to neither the plane nor the line case. Therefore, no estimate of velocity can be given for the corresponding two points in the image volume. However, this does not imply that there is no motion at these points, i.e., the velocity vector is zero. It is rather the case that the local structure does not comply with the models used here for motion estimation, which is either a linear structure or point moving with a velocity that is relatively constant over at least some few frames. Consequently, rather than setting the estimated velocity to zero at these points, it is taken to be unknown.

Exercise 17

Consider an image sequence represented as a 3D volume (2D spatial plus time), and a local neighborhood of this volume which describes a moving linear structure. Let $\mathbf{n}$ be a 2D normal vector of the linear structure, and let corresponding $\mathbf{v}$ be the spatio-temporal motion vector. The motion generates a planar 3D structure which can be represented by an orientation tensor $\mathbf{T}$ of rank one. Show that the projection of $\hat{\mathbf{e}}_1$, a normalized eigenvector of $\mathbf{T}$ with largest eigenvalue, onto the image plane is a normal vector to the linear structure.

Hint Solution

Hint to exercise 17

You need two 3D vectors that are linearly independent and both lie in the plane generated by the motion. From these you can easily construct a vector which is perpendicular to both of them, and therefore a normal vector to the plane. This vector is parallel to $\hat{\mathbf{e}}_1$. Project this vector onto the image plane and consider the result.

Solution to exercise 17

Let the non-zero 3D vector **m**,

$$\mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \\ 0 \end{pmatrix} \tag{264}$$

be parallel to the linear structure and lie in the image plane. This vector lies also in the plane that is generated by the motion of the linear structure. This is true also for **v**,

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ 1 \end{pmatrix} \tag{265}$$

the spatio-temporal velocity vector. They are linearly independent, and a vector perpendicular to both, i.e., a normal vector to the plane, is found by taking their cross product

$$\mathbf{m} \times \mathbf{v} = \begin{pmatrix} m_2 \\ -m_1 \\ m_1 v_2 - m_2 v_1 \end{pmatrix} \tag{266}$$

This vector is then parallel to $\hat{\mathbf{e}}_1$. The projection of $\hat{\mathbf{e}}_1$ onto the image plane gives the vector

$$\mathbf{n} = \pm \begin{pmatrix} m_2 \\ -m_1 \\ 0 \end{pmatrix} \tag{267}$$

which apparently is perpendicular to **m**. Hence, **n** is a normal vector to the linear structure.

Exercise 18

A 2D image neighborhood is described by the function f,

$$f(\xi) = g(\xi^T \hat{\mathbf{x}}) \qquad \xi = \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \qquad \hat{\mathbf{x}} = \begin{pmatrix} 0.8 \\ -0.6 \end{pmatrix} \tag{268}$$

where

$$g(y) = 0.3 \cos(0.7y) + 0.6 \sin(1.4y) \tag{269}$$

Hence, f complies with the model stated for estimation of local orientation according to Section 4. For this neighborhood, compute the corresponding orientation tensor, represented as a matrix with numerical values, according to one of the two procedures described in Section 5.1. Examine the resulting tensor to see if it is of rank one, and if the eigenvector corresponding to the largest eigenvalue is parallel to $\hat{\mathbf{x}}$. As a radial function of the quadrature filters, use the lognormal function described in Section 4.8, with

$$\rho_0 = 1.2 \qquad B = 2 \tag{270}$$

Hint Solution

Hint to exercise 18

You need to compute the filter outputs of all filters. This is done very easily by drawing a picture of the the signal in the Fourier domain. Since f is simple its Fourier transform is concentrated to a line which passes through the origin, and along that line it varies as G, the Fourier transform of g. In this picture, also the filters and their filter directing vectors can be drawn, and from this the filter outputs can be derived immediately. Note that the filter output is the integral of the product between signal and the filter taken in the Fourier domain (or in the spatial domain). The outputs are complex since the quadrature filters are complex filters. The absolute value of the filter outputs are then used in a linear combination with the dual tensors $\{\tilde{\mathbf{N}}_k\}$, according to Equation (87). The dual tensors are given by Equations (91) and (95), respectively.

To find out whether or not the resulting tensor in fact represents the correct orientation, compute its eigenvalues and eigenvectors. You will then find that one eigenvalue (λ_2) is very close to zero while the other (λ_1) is of significant value. Furthermore, the eigenvector $\hat{\mathbf{e}}_1$ (corresponding to λ_1) is almost parallel to $\hat{\mathbf{x}}$.

Solution to exercise 18

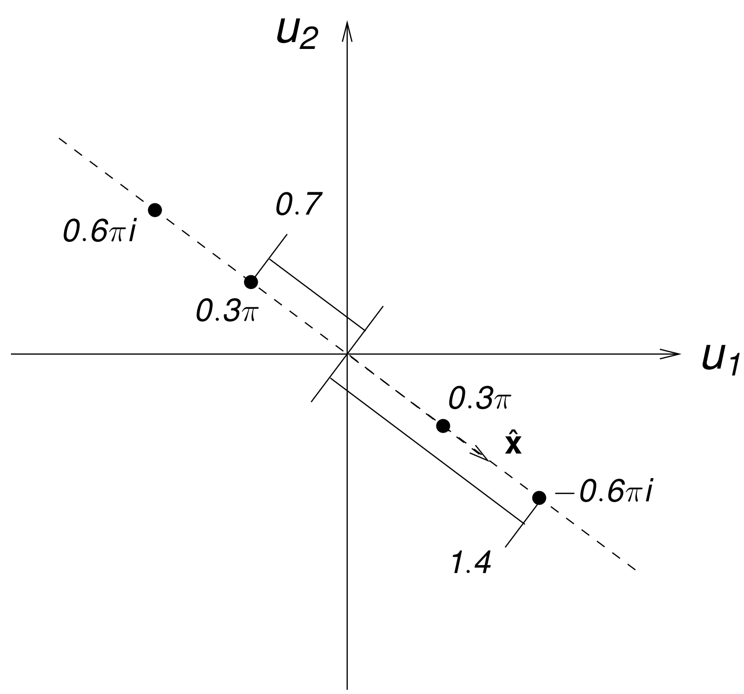
The function f is simple, and therefore its Fourier transform can be written

$$F(\mathbf{u}) = 2\pi\ G(\mathbf{u}^T\hat{\mathbf{x}})\ \delta_{\hat{\mathbf{x}}}^{\text{line}}(\mathbf{u}) \tag{271}$$

where $\mathbf{u}$ is the frequency coordinate, $\delta_{\hat{\mathbf{x}}}^{\text{line}}$ is an impulse line which passes through the origin in the direction of $\hat{\mathbf{x}}$, and G is the one-dimensional Fourier transform of g.

$$G(u)=0.3\pi[\delta(u+0.7)+\delta(u-0.7)]+0.6\pi i[\delta(u+1.4)-\delta(u-1.4)] \tag{272}$$

The Fourier transform of f is illustrated in the figure below



Note in particular the change in phase sign (sign of the argument) of the sinus term compared to the non-changing phase of the cosine term in the Fourier domain. This effect has consequences when determining the filter outputs.

In the following, we consider only the case of using three quadrature filters for the estimation of the orientation. The calculations are similar in the case of four filters. According to Section 5.1.1, the filters can have directing vectors according to Equation (89). This is illustrated in Figure 17, which shows the filter directing vector of the corresponding filter together with the Fourier transform of the signal. Furthermore, the zero half of each filter's Fourier transform is shaded. Note that different impulses show up in the non-zero part of the quadrature filters.

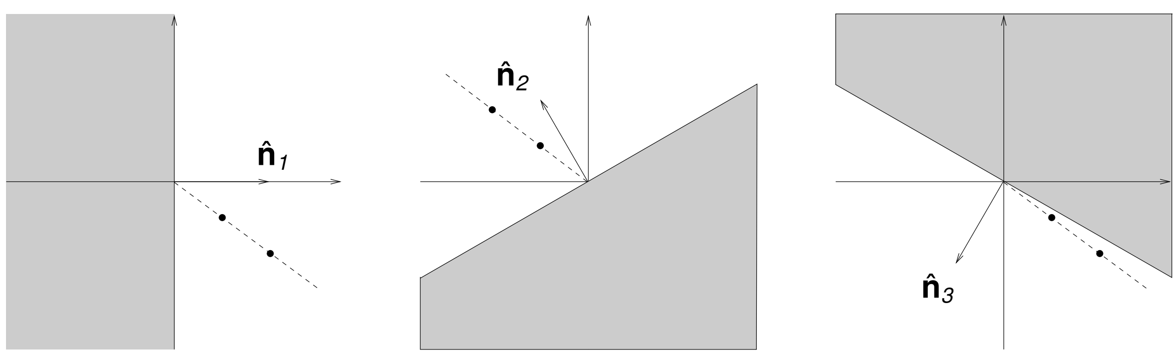


Figure 17: Three quadrature filters with directions given by the vectors $\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \hat{\mathbf{n}}_3$. The zero half of each filter's Fourier transform is shaded.

With these figures in mind, it is straightforward to calculate the three filter outputs. They are the integrals of signal's Fourier transform times the Fourier transform of the filter, the integral is taken over the entirety of the Fourier domain. However, since the Fourier transform of the signal is just a few impulses, the integral collapses into a sum over the impulses, where each term is the product of the corresponding impulse's "amplitude" and the value of the filter function at that point. Note that half of the impulses are in the zero-half of each filter, and can therefore be omitted in the sum. Furthermore, each filter is separable into a radial function, R, and an angular function, D. The radial function is presented in Equation (88), and the angular function is simply

$$D(\mathbf{u}) = (\hat{\mathbf{u}}^T \hat{\mathbf{n}}_k)^2 \tag{273}$$

This gives (with four significant digits)

$$q_1' = 0.64 \cdot [\, 0.3\, R(0.7) - 0.6i\, R(1.4)\,] \tag{274}$$

$$q_2' = 0.8457 \cdot [\, 0.3\, R(0.7) + 0.6i\, R(1.4)\,] \tag{275}$$

$$q_3' = 0.0143 \cdot [\, 0.3\, R(0.7) - 0.6i\, R(1.4)\,] \tag{276}$$

We also need

$$R(0.7) = 0.6576 \qquad R(1.4) = 0.9663 \tag{277}$$

which sets numerical values to all filter outputs

$$\begin{aligned} q'_1 &= 0.1263 - 0.3711i \\ q'_2 &= 0.1668 + 0.4903i \\ q'_3 &= 0.0028 - 0.0083i \end{aligned} \tag{278}$$

and, finally, to their absolute values

$$\begin{aligned} q_1 &= |q'_1| = 0.3920 \\ q_2 &= |q'_2| = 0.5179 \\ q_3 &= |q'_3| = 0.0088 \end{aligned} \tag{279}$$

With the absolute values of the filter outputs at hand, it is now possible to construct the orientation tensor as a linear combination of these absolute values and the dual tensors corresponding to the filter directions according to

$$\mathbf{T} = q_1\,\tilde{\mathbf{N}}_1 + q_2\,\tilde{\mathbf{N}}_2 + q_3\,\tilde{\mathbf{N}}_3 \tag{280}$$

The dual tensors are listed in Equation (91), which gives the orientation tensor

$$\mathbf{T} = \begin{pmatrix} 0.3920 & -0.2940 \\ -0.2940 & 0.2205 \end{pmatrix} \tag{281}$$

The eigenvalues of the tensor are

$$\lambda_1 = 0.6124 \qquad \lambda_2 = 0 \tag{282}$$

and a normalized eigenvector corresponding to λ_1 is

$$\hat{\mathbf{e}}_1 = \pm \begin{pmatrix} 0.8000 \\ -0.6000 \end{pmatrix} \tag{283}$$

Hence, $\mathbf{T}$ is indeed a rank one tensor, and $\hat{\mathbf{e}}_1$ represents the local orientation of the neighborhood as described in Section 3.

Try to make the same calculations also for four filters. In that case, both the filter output magnitudes q_k and the dual tensors $\tilde{\mathbf{N}}_k$ are different. The resulting tensor, however, should be the same.

Exercise 19

A signal neighborhood is described by the simple signal f,

$$f(\xi) = g(\xi^T \hat{\mathbf{x}}) \tag{284}$$

where

$$g(y) = \cos(\omega y + \varphi) = \cos(\omega[y + \frac{\varphi}{\omega}]) \tag{285}$$

Let this signal be subject to a filtering by a quadrature filter, e.g., a spherically separable filter as described in Section 4.4. Show that the filter output q' of this filter has an absolute value q which is independent of φ.

The consequence of this property is that q is *invariant* to φ, the local phase of the signal. From a more practical perspective, this mean that the filter output is invariant to position, as long a the local signal model is simple and g contains only one frequency component, or has narrow band-width. The orientation tensor $\mathbf{T}$ is constructed as the linear combination of the absolute value of several filter outputs at the same point, Equation (87). Consequently, also the tensor is phase-invariant or position-invariant in this case.

Hint Solution

Hint to exercise 19

Follow the steps for computing the filter output q presented in Sections 4.3 and 4.4, resulting in Equation (81). Using the g presented above, it is straightforward to obtain the scalar a, and it turns out that $|a|$ is independent of φ. As a consequence, so is q.

Solution to exercise 19

The scalar a is defined as

$$a = \frac{1}{2\pi} \int_0^\infty G(u)\, R(u)\, du \tag{286}$$

and from the definition of g follows that

$$\begin{aligned} G(u) &= \pi\, [\, \delta(u+\omega) + \delta(u-\omega)\,]\ e^{iu\varphi/\omega} = \\ &= \pi \left[\, \delta(u+\omega)\, e^{-i\varphi} + \delta(u-\omega)\, e^{i\varphi} \,\right] \end{aligned} \tag{287}$$

which gives

$$a = \frac{1}{2}\, R(\omega)\, e^{i\phi} \qquad \text{and} \qquad |a| = \frac{1}{2}\, R(\omega) \tag{288}$$

Furthermore, we have

$$q = |a|\, [D(-\hat{\mathbf{x}}) + D(\hat{\mathbf{x}})] \tag{289}$$

where the second factor only depends on $\hat{\mathbf{x}}$ (and not on g), which finally shows that q is invariant relative to φ.

9 Computer Exercises

The Computer Exercises part of this package contains the following exercises

- CE 1: Computing the 2D orientation tensor
  All exercises are executed in Matlab.

MATLAB AND IMAGES

There are some things that you need to know about when using (generating and displaying) images in Matlab.

1. All images are represented in terms of matrices. Hence, a typical gray-scale image is conveniently represented as a matrix of real numbers.
2. When the image is to be displayed this is no quite the truth. The standard mechanism for image display (the function `image`) uses the matrix elements as indices in a color table, which can be set by the user or by some program. Therefore, to display an image you need to set the color table (using the function `colormap`) and then render the image (by means of `image`). Along with this package there are some display functions which actually takes, e.g., an image (matrix) of gray-scale values and displays it properly (`normimage`). Other functions can display images of real numbers, both positive and negative (`grimage`), and even images of complex numbers (`gopimage`).
3. The indices of a Matlab matrix are such that the first index runs vertically and the second runs horizontally, starting at the top left corner with index ($1,1$) and increasing the coordinates downwards, and to the right, respectively. Images, on the other hand, have a coordinate system where the first coordinate runs horizontally and the second vertically, with the origin at the center of the image, and in-

creasing the coordinates to the left, and upwards, respectively. This image coordinate may not be universally accepted, but at least it is convenient in the following exercises. Consequently, the coordinate system of an image is not the same as the indices of the corresponding matrix, so this has to be taken care of. The easiest way to do this, is to construct images that contain the coordinates of the image coordinate system.

4. The the images used in the exercise can in principle be of arbitrary size. Neither the estimation procedures nor Matlab sets any hard limitations on the size. On the other hand, you computer has a limited amount of primary memory and processing power, which means that you have to limit the image size to something it can cope. The examples shown here uses the image size 512×512, which is a fairly commonly used image size in image processing. Depending on the memory and processing power offered by your computer this size may have to be decreased (or it can even by increased), which has to be taken into account when you do the exercise. It should be noted, though, that images sizes which are integer powers of two are to be preferred when Matlab's implementation of the DFT (`fft`) is used. The same goes for the resolution of the color table used for image display. This exercise assumes 24-bit color resolution, thereby allowing each figure to be displayed using its own color table. If you computer has a lower color resolution (e.g., 8-bit) also

this may imply that different images can not be displayed properly at the same time.

5. When an image is displayed, its size on the screen may not be determined by its natural size in pixels (number of row and columns of the corresponding image matrix), but rather by the size of the display window. Hence, a pixel in the image may not correspond to a pixel on the screen. If the size of the window changes, the size of the image change as well. This may seem like a nice feature, but the result can be confusing since the apparent change of image size is achieved by resampling the image. If the image is not band-limited or subject to any other restriction, displaying it in something else than its natural size can cause artifacts to appear or structures in the image to disappear. This is the case regardless of whether the displayed size is smaller or larger than the natural size. The display functions presented here (`normimage`, `grimage`, `gopimage`) always display an image in its natural size, and by using them you can avoid this problem. However, this assumes that the size of the display window is not changed after an image is rendered.

6. Images are not portable, not even on the same computer. Of course, there are a number of image formats (tiff, gif, jpeg, etc) which are portable in terms of storing and retrieving. However, when one and the same image is displayed on the screen, it may not look quite the same on

another computer or even on the same computer but with a different display software. Normally the differences lie in dynamic of the image intensity and in the color saturation. One part of the problem lies in the video monitor of you computer which typically is slightly different from that of another computer, but also in how various software interpret color information encoded in the images and how this is mapped into colors on the screen. Consequently, an image which looks perfectly nice in one situation may look much darker or brighter, or exhibit different color saturation, when displayed in a different environment. This effect does not cause that much of a problem in these exercises, unless you intend to use images produced elsewhere and run some processing on them, or you want to save some result images produced in Matlab for later rendering in other environments. In those cases, you should not be surprised if images imported into Matlab look slightly different when they are displayed there, or if images exported from Matlab look different elsewhere.

9.1 COMPUTER EXERCISE 1: 2D ORIENTATION ESTIMATION

The goal of this exercise is to get an understanding of how a representation of local orientation in terms of a tensor can be estimated for 2D images. The orientation tensor is here described as symmetric 2×2 matrix, according to

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} \tag{290}$$

The tensor is estimated at each point of the image, the result of the estimation procedure is therefore a new image which contains a 2×2 matrix at each point. Since there are three independent (at least in practice) elements in the matrix, the result can conveniently be represented as three images, one each for T_{11}, T_{12}, and T_{22}, respectively. At the end of the exercise you should be able to write a Matlab-function which take an image as input and computes the resulting tensor images.

The filtering operations that are defined for the estimation is here made by means of multiplication in the Fourier domain. This is usually not the most practical or even the fastest way of implementing a filtering process. Furthermore, certain edge effects will occur due to the corresponding cyclic convolution. However, with this approach, instead of using discrete a spatially truncated filters, no attention needs to be payed to choosing the filter coefficients in an optimal way.

9.1.1 COMPUTER EXERCISE 1: Preparations

First, generate two images which hold the first and second image coordinate.

```
x1=ones(512,1)*(-256:255);

x2=(255:-1:-256)'*ones(1,512);
```

You will also need to corresponding coordinates in the Fourier domain

```
u1=ones(512,1)*(-256:255)*pi/256;

u2=(255:-1:-256)'*ones(1,512)*pi/256;
```

It will prove convenient to have also the polar coordinates in the Fourier domain at hand

```
rho=sqrt(u1.*u1+u2.*u2);

psi=angle(u1+i*u2);
```

9.1.2 COMPUTER EXERCISE 1: The quadrature filters

We are now going to generate the three quadrature filters needed for the estimation procedure. These are described in the Fourier domain as spherically (polar) separable functions

$$F_k(\mathbf{u}) = R(\rho)\ D_k(\hat{\mathbf{u}}), \qquad \rho = \|\mathbf{u}\| \qquad \hat{\mathbf{u}} = \frac{\mathbf{u}}{\rho} \tag{291}$$

As was mentioned in Section 4.8, R can be chosen quite arbitrary, but considerations related to simultaneous localization in the spatial and Fourier domain lead to

$$R(\rho) \ = \ e^{-\frac{4}{B^2 \ln 2}\ln^2(\rho/\rho_i)} \tag{292}$$

where B is the relative bandwidth and ρ_0 is the center frequency of the filter. Just to set some values to these parameters, try $B = 3$ and $\rho_0 = \frac{\pi}{2\sqrt{2}}$ (radians per pixel). The radial function can now be computed as

```
B=3;

rho0=0.7;

R=exp(-4/(B*B*log(2))*(log((rho+eps)/rho0)).^2);
```

The term `eps` (which is the smallest positive number that Matlab can represent) is added to the radial variable to avoid taking the logarithm of zero.

The directional functions of the filters are defined as (see Section 4.5)

$$D_k(\hat{\mathbf{u}}) = \begin{cases} (\hat{\mathbf{u}}^T \hat{\mathbf{n}}_k)^2 & \hat{\mathbf{u}}^T \hat{\mathbf{n}}_k > 0 \\ 0 & \text{otherwise} \end{cases} \tag{293}$$

where $\hat{\mathbf{n}}_k$ is the directing vector of the corresponding filter. Note that the scalar product between $\hat{\mathbf{u}}$ and $\hat{\mathbf{n}}$ amounts to the cosine of the angular difference between the two vectors. The filter directing vectors are (see Section 5.1.1)

$$\hat{\mathbf{n}}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \hat{\mathbf{n}}_2 = \begin{pmatrix} -1/2 \\ \sqrt{3}/2 \end{pmatrix} \quad \hat{\mathbf{n}}_3 = \begin{pmatrix} -1/2 \\ -\sqrt{3}/2 \end{pmatrix} \tag{294}$$

which corresponds to the direction angles $0^\circ, \pm 120^\circ$ (or 0 and $\pm\frac{2\pi}{3}$ radians) relative to the u_1-axis of the coordinate system in the Fourier domain. The three angular functions can now be computed

```
an1=0;an2=2*pi/3;an3=-2*pi/3;

F1=R.*(cos(psi-an1).^2).*(cos(psi-an1)>0);

F2=R.*(cos(psi-an2).^2).*(cos(psi-ans2)>0);

F3=R.*(cos(psi-an3).^2).*(cos(psi-ans3)>0);
```

You can now display the three filter functions, e.g., each in its own figure window

```
figure(1);normimage(F1);
```

```
figure(2);normimage(F2);

figure(3);normimage(F3);
```

Note that the origin of the Fourier domain is at the center of each image. The quadrature property of the filters implies that they are all equal to zero at one half-plane, which is visualized in a better way using contour plots

```
figure(1);contour(F1);

figure(2);contour(F2);

figure(3);contour(F3);
```

9.1.3 COMPUTER EXERCISE 1: Filtering the image

At this point it is time to consider the image on which the orientation estimation should be made. For example, try the test image called `ploop`.

```
load ploop;

figure(4);normimage(ploop);
```

This test image contains a circular pattern consisting of a radial sine function, hence any possible orientation is present in the image, and also any phase. Furthermore, the spatial frequency increases when going toward the center of the image, and a large range of frequencies are represented (from $\approx$ *0.25* to π radians per pixel). This test image has been designed to allow performance evaluation of this or other estimation procedures.

To compute the filter output when each filter is applied to this image we can Fourier transform the test image, multiply by the Fourier transform of the filters (given by `F1, F2, F3`), and then do an inverse Fourier transform. Finally, the absolute value of the filter outputs is computed. It should be noted that this filtering approach has some disadvantages, e.g., it introduces edge effects and it may not be as fast compared to a straightforward convolution with localized and discrete filters. However, it is simple and we do not have to get involved into the filter design needed to create the discrete filters.

It should be noted that Matlab's standard implementation of the discrete Fourier transform implies that the origin of both the spatial and the Fourier domains are located at the top right corner. To shift the origin to the center of the domains, use the function `fftshift`.

```
Ploop=fftshift(fft2(fftshift(ploop)));
q1=abs(fftshift(ifft2(fftshift(F1.*Ploop))));
q2=abs(fftshift(ifft2(fftshift(F2.*Ploop))));
q3=abs(fftshift(ifft2(fftshift(F3.*Ploop))));
```

We now have the absolute value of the filter outputs in three images `q1, q2, q3`. Before we proceed, take a look at them, e.g., using the function `normimage`.

```
figure(1);normimage(q1);
figure(2);normimage(q2);
figure(3);normimage(q3);
```

As you can see, the output from each filter indicates that each filter is sensitive to a particular orientation range (not direction) of the signal, according to the directional function D_k. Furthermore, the absolute value of the filter outputs is more or less invariant to the phase of the signal.

9.1.4 COMPUTER EXERCISE 1: Generating the dual tensors

To obtain the final orientation tensor, the three scalars q_1,q_2, and q_3 are to be combined, at each point of the image, with the dual tensors $\tilde{\mathbf{N}}_1$,$\tilde{\mathbf{N}}_2$, and $\tilde{\mathbf{N}}_3$, respectively, see Equation (87). The dual tensor are listed in Equation (91), and their derivation is discussed in exercise 9. To get the full picture, however, the computation of these tensors are included here as well.

First, define the three direction vectors

```
n1=[cos(an1);sin(an1)];
n2=[cos(an2);sin(an2)];
n3=[cos(an3);sin(an3)];
```

Then the corresponding filter tensors are formed as the outer product of the filter directing vectors

```
N1=n1*n1'
N2=n2*n2'
N3=n3*n3'
```

By treating these tensors (or matrices) as vectors, the computation of the dual tensors is considerably simplified. Matlab's operator `(:)` reshapes a matrix to a column vector by concatenating the columns of matrix, one under the other, in the same order as they appear in the matrix.

```
N1=N1(:)
```

```
N2=N2(:)
```

```
N3=N3(:)
```

Form a basis matrix `B` which contains these vectors in its columns

```
B=[N1 N2 N3]
```

The dual basis matrix is the given by (see Section 2.8)

```
dualB=B*inv(B'*B)
```

This matrix contains the dual tensors in its column (and in the same way as `B` contains the original tensors in its columns), and they can be reshaped into matrices again

```
dualN1=reshape(B(:,1),2,2)
```

```
dualN2=reshape(B(:,2),2,2)
```

```
dualN3=reshape(B(:,3),2,2)
```

If you have followed the above steps correctly, the three matrices `dualN1`, `dualN2`, and `dualN3` have their values according to Equation (91).

9.1.5 COMPUTER EXERCISE 1: Generating the orientation tensor

With both filter outputs and dual tensors at hand, the only remaining step is their linear combination. This should be done point-wise, so the result can be considered as an image containing 2×2 symmetric matrices. Any such matrix contains three independent elements (at least linearly independent elements), hence the tensor image can be represented as three images of scalars, one for T_{11}, one for T_{12}, and one for T_{22}. Each element image is then a linear combination of the q_k images and the corresponding element of the dual matrices. For example, we get

$$T_{11} = q_1 \, \tilde{N}_{1,11} + q_2 \, \tilde{N}_{2,11} + q_3 \, \tilde{N}_{3,11} \tag{295}$$

where $N_{1,11}$ is the $(1,1)$ element of $\tilde{\mathbf{N}}_1$, etc. Hence,

```
T11=q1*dualN1(1,1)+q2*dualN2(1,1)+q3*dualN3(1,1);
T12=q1*dualN1(1,2)+q2*dualN2(1,2)+q3*dualN3(1,2);
T22=q1*dualN1(2,2)+q2*dualN2(2,2)+q3*dualN3(2,2);
```

We now have the orientation tensor at each point for the test image, and the three components of the tensor can now be displayed

```
figure(1);grimage(T11);
figure(2);grimage(T12);
figure(3);grimage(T22);
```

9.1.6 COMPUTER EXERCISE 1: Examining the orientation tensor

Having calculated the orientation tensor at each point of the image, it is now time to examine it an see if it has the expected properties. The three images previously displayed give some indications on this issue, although in a qualitative manner. Given this particular test image, and the coordinate system, it is evidently so that a neighborhood around point $\mathbf{x}$ is a simple function where $\hat{\mathbf{x}}$ is normal to the parallel lines of constancy. Hence, from a theoretical point of view, the orientation tensor should proportional to $\hat{\mathbf{x}}\hat{\mathbf{x}}^T$ at point $\mathbf{x}$. Therefore, if the angular coordinate of point $\mathbf{x}$ is α, the orientation tensor at that point has the following dependency with respect to α

$$\begin{aligned} T_{11} &= A \cos^2 \alpha \\ T_{12} &= A \cos \alpha \, \sin \alpha \\ T_{22} &= A \sin^2 \alpha \end{aligned} \tag{296}$$

(see Section 3.3). Returning to the three images that depict these three tensor components, you will find that the tensor has both a variation along the radial component, which is due to the radial frequency function of the filters, and an angular variation according to the above expressions. So far so good.

To get a more precise feeling for the qualities of the resulting tensor, use the Matlab function `tensorview2D`. It allows you to use the screen cursor as a probe in the tensor field; but clicking on a point in the test image it presents facts about the corresponding orientation tensor, e.g., the eigenvalues. Do

```
help tensorview2D
```

to get more information on what it does. Now, use it on the tensor fields derived from the test image

```
figure(1);normimage(ploop);

tensorview2D(1,2,T11,T12,T22);
```

You will find that each time you click in the area where the local orientation is well-defined, the eigenvalues are such that λ_2 is very small compared to λ_1, something that is reflected in c_1 which takes values very close to unity. Furthermore, the line at the bottom (representing $\pm\hat{\mathbf{e}}_1$) is perpendicular to the linear structure of each neighborhood.

To get a more global view of the tensor field, compute $\|\mathbf{T}\|$, the norm of the tensor, and its two eigenvalues λ_1 and λ_2.

```
normT=sqrt(T11.*T11+T22.*T22+2*T12.*T12);

l1=(T11+T22)/2+sqrt(((T11-T22)/2).^2+T12.^2);

l2=(T11+T22)/2-sqrt(((T11-T22)/2).^2+T12.^2);

figure(1);normimage(normT);

figure(2);grimage(l1);

figure(3);grimage(l2);
```

Note that the three images does not present absolute values of these three entities; the range of each variable is scaled differently so, for example, you can not compare the images of λ_1 and λ_2 directly. However, the first image show that $\|\mathbf{T}\|$ is practically constant with respect to orientation, i.e., if you choose a point and follow the corresponding circle centered at the image origin, the norm is more or less constant. There is a variation in the radial direction, but that effect is due to the radial frequency function of the filters combined with the variation of the local spatial frequency in the test image. These observations are valid also for λ_1. Finally, for λ_2 it should be noted that this eigenvalue in general is at approximately two orders of magnitude smaller than λ_1, this can be seen by computing the mean of the absolute value of each of the two eigenvalues

```
mean2(abs(l1))
```

```
mean2(abs(l2))
```

In the image which presents λ_2 (figure 3), you can also see that this eigenvalue is not exactly zero at all points. In fact, it even takes negative values (although small negative values) at some points, and the general properties of this eigenvalue seems to be quite dependent on orientation. To be sure, the first two images are not entirely invariant with respect to orientation. To see this, produce an image which indicates the pixels where $\|\mathbf{T}\|$ is larger than *99%* of the maximum value of the norm.

```
figure(1);normimage(normT>max2(normT)*0.99);
```

As this image shows, the set of points which match the condition are not aggregated along a perfect circle, but rather as some few blobs. Take some time to try and figure out one or two reasons why.

References

[1] H. S. M. Coxeter. *Introduction to Geometry*. John Wiley & Sons, Inc., 1961.

[2] G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–173, 1978.

[3] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.

[4] D. H. Hubel and T. N. Wiesel. Brain mechanisms of vision. *Scientific American*, September 1978.

[5] 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.

[6] 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.

[7] H. Knutsson and M. Andersson. Robust N-Dimensional Orientation Estimation using Quadrature Filters and Tensor Whitening. Adelaide, Australia, April 1994. IEEE. LiTH-ISY-R-1798.

[8] H. Knutsson, H. Bårman, and L. Haglund. Robust Orientation Estimation in 2D, 3D and 4D Using Tensors. In *Proceedings of Second International Conference on Automation, Robotics and Computer Vision, ICARCV'92*, Singapore, September 1992.

[9] H. Knutsson, B. von Post, and G. H. Granlund. Optimization of arithmetic neighborhood operations for image processing. In *Proceedings of the First Scandinavian Conference on Image Analysis*, Linköping, Sweden, January 1980.

[10] Hans Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.