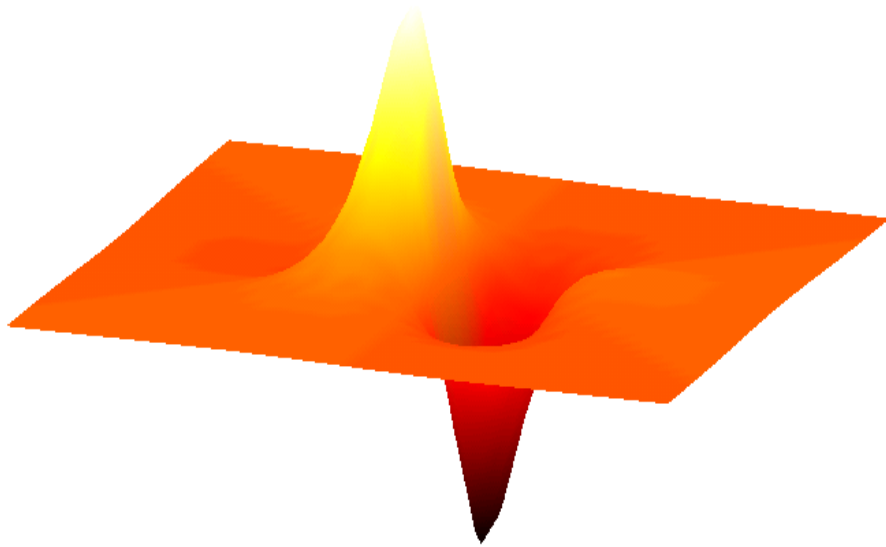


Diplomarbeit

# Local Feature Detection by Higher Order Riesz Transforms on Images



Anne Sedlazeck  
(Matrikelnummer: 598716)

Betreuer: Lennart Wietzke

Februar 2008

Kognitive Systeme  
Prof. Dr. Sommer  
Institut für Informatik  
Christian-Albrechts-Universität zu Kiel



## Abstract

An important issue in image analysis is the detection of features in images. Problems occur, when the images contain noise or are rotated. An example for a rotation invariant feature is the apex angle. However, apex angle detection is only expedient, on a special class of images, therefore in this thesis only images with up to two overlaid intrinsically one dimensional structures are considered. Images belonging to this class also allow the calculation of the local main orientation. Together, local main orientation and apex angle provide access to the computation of both orientations and vice versa. The estimation of the local main orientation and apex angle are done by applying Riesz transforms up to third order to a local neighborhood of the image. In order to accomplish that, a new interpretation of the Riesz transform is studied, which in turn provides a new interpretation of the monogenic signal and the monogenic curvature tensor. Another direct result of the new interpretation of the Riesz transform is the ability to detect the rotation angle between two otherwise identical images. In order to implement Riesz transforms up to the third order they need to be combined with a suitable bandpass filter. The Laplacian of Poisson allows the development of new filter kernels for the second and third order Riesz transforms. The implementation of this method is compared to an implementation of a well-known method using image gradients and computing the orientations of up to two overlaid structures that has been developed in Lübeck.

## Zusammenfassung

Eine wichtige Aufgabe in der Bildverarbeitung ist die Erkennung von Merkmalen in Bildern. Probleme treten auf, wenn die Bilder Rauschen enthalten oder gedreht sind. Ein Beispiel für ein rotationsinvariantes Merkmal ist der Öffnungswinkel. Allerdings ist die Berechnung des Öffnungswinkels nur auf einer bestimmten Klasse von Bildern sinnvoll, daher beschäftigt sich diese Arbeit nur mit Bildern, in denen zwei Signale mit intrinsischer Dimension eins überlagert sind. Bilder dieser Art erlauben außerdem die Berechnung der lokalen Hauptorientierung. Hauptorientierung und Öffnungswinkel zusammen bieten die Möglichkeit die Orientierungen der beiden Strukturen zu berechnen. Die lokale Hauptorientierung und der Öffnungswinkel werden durch Anwendung der Riesztransformation bis zur dritten Ordnung in einer lokalen Umgebung im Bild berechnet. Um das zu erreichen, wird eine neue Interpretation der Riesztransformation betrachtet, die wiederum eine neue Interpretation des monogenen Signals und des monogenen Krümmungstensor liefert. Ein weiteres direktes Ergebnis der neuen Interpretation der Riesztransformation ist die Fähigkeit, einen Drehwinkel zwischen zwei sonst gleichen Bildern zu berechnen. Um die Riesztransformationen bis zur dritten Ordnung zu implementieren, wird ein Bandpassfilter benötigt. Eine Kombination aus Laplaceoperator und Poissonfilter wird verwendet, um neue Filterkerne für die Riesztransformationen zweiter und dritter Ordnung zu berechnen. Eine Implementierung dieser Methode wird mit einer Implementierung einer weit verbreiteten Methode verglichen. Die zweite Methode wurde in Lübeck entwickelt und verwendet Gradienten in Bildern, um die Orientierungen von bis zu zwei überlagerten Strukturen zu bestimmen.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Orientation and Apex Angles . . . . .	3
1.1.1	Intrinsic Dimension of Images . . . . .	3
1.1.2	Orientation . . . . .	4
1.1.3	Apex Angles . . . . .	4
<b>2</b>	<b>Orientation Estimation with two Overlaid i1D Structures</b>	<b>7</b>
2.1	Using Riesz Transforms - the Kiel Method . . . . .	7
2.1.1	The Monogenic Signal as an Extension of the Analytic Signal . . . . .	7
2.1.2	The Monogenic Signal in Terms of the Radon Transform . . . . .	11
2.1.3	The Monogenic Curvature Tensor in Terms of the Radon Transform . . . . .	16
2.2	Using Gradients - the Lübeck Method . . . . .	23
2.2.1	Single Orientation Estimation . . . . .	24
2.2.2	Double Orientation Estimation . . . . .	25
<b>3</b>	<b>Filter Design and Implementation</b>	<b>31</b>
3.1	Implementation of Riesz Transforms . . . . .	31
3.1.1	Kernels Obtained by Repeated Convolution in the Spatial Domain . . . . .	31
3.1.2	Using the Discrete Radon Transform and its Inverse . . . . .	34
3.1.3	Kernels Developed in the Frequency Domain . . . . .	35
3.1.4	Kernels Using the LOP as a Bandpass Filter Approximation . . . . .	36
3.2	Implementation of the Lübeck Method . . . . .	43
<b>4</b>	<b>Comparison</b>	<b>45</b>
4.1	Local Analysis of Synthetic Images . . . . .	45
4.2	Global Tests on Images . . . . .	50
4.2.1	Synthetic Images . . . . .	51
4.2.2	Corners and Edges . . . . .	52
4.2.3	Textures or Fabrics . . . . .	54
4.2.4	Images Containing Noise . . . . .	54
<b>5</b>	<b>Application: Rotation Estimation</b>	<b>59</b>
5.1	Implementation and Results . . . . .	60
<b>6</b>	<b>Conclusion and Outlook</b>	<b>63</b>
<b>A</b>	<b>Notations</b>	<b>67</b>

<b>B Riesz Transform Kernel Derivation</b>	<b>69</b>
B.1 Derivation of the First Order Riesz Transform Kernels with a DOP Filter	69
B.2 LOP Kernel . . . . .	70
<b>C Implementation</b>	<b>75</b>

# 1 Introduction

Image analysis is a main task in Computer Vision. In order to analyze images, the detection of features in the image is a common technique. Typical features are for example edges and corners. In this thesis a special class of images is considered. It is assumed that images consist of up to two overlaid intrinsically one-dimensional signals, hence have intrinsic dimension two (see figure 1.1). Analyzing this kind of images allows the detection of the local main orientation and the apex angle between these signals. These features can be used to estimate the orientations of both signals. When making the additional assumption that the image is band limited - meaning it only consists of a narrow band of frequencies - Riesz transforms can be used to estimate these features. There are a lot of practical applications based on the detection of these features, or the use of Riesz transforms on images in general. E.g.:

- In medical imaging different semi-transparent tissue layers can be separated and analyzed.
- Layer separation is also used in texture analysis. If two textures are overlaid, they can be separated according to the different orientations.
- The paper of Barth et al. [4] describes a method for detecting optical flow of transparent overlaid structures.
- The first order Riesz transform allows matching between two images via rotation/scale detection of pixels and regions.
- Homogeneous regions in images can be segmented.

Up to now, the use of Riesz transforms of images has been studied in a few works, for example in [9], [8], [12], [6], [10], [11] by Sommer and Felsberg. In these works, the monogenic signal is defined - it consists of the image signal itself and the first-order Riesz transform of the image. Sommer and Felsberg showed that the monogenic signal allows the estimation of local phase, orientation, and amplitude. The definition of the monogenic signal (e.g. [9]) is motivated by viewing it as a 2D generalization of the analytic signal. This generalization is achieved by applying the Riesz transform to the signal instead of the Hilbert transform.

The monogenic curvature tensor consists of successively applied Riesz transforms up to third order and is motivated by the Hessian matrix used in differential geometry. It has been studied by Zang and Sommer in [24] and [25]. An ensuing work at the chair of cognitive systems consisted of deriving a theory about the Riesz transform interpreted using the Radon transform. Applying this new interpretation to the monogenic signal and monogenic curvature tensor allows computation of the apex angle and the local main

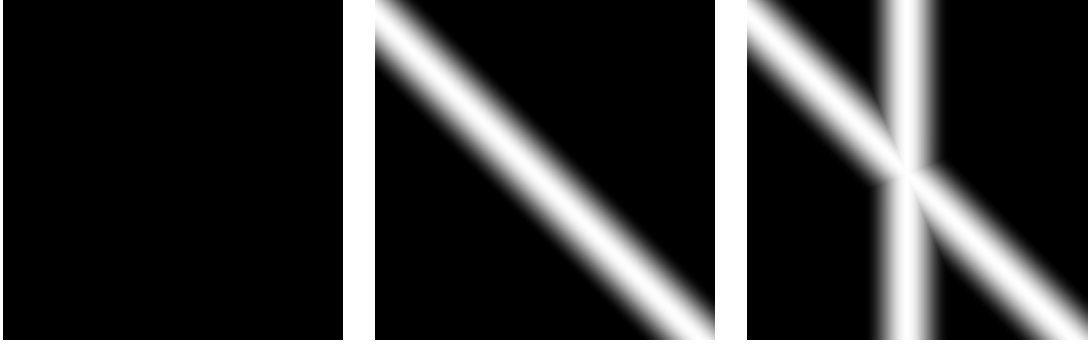


Figure 1.1: From left to right: a) intrinsic dimension zero (i0D), b) intrinsic dimension one (i1D), c) intrinsic dimension two (i2D).

orientation as well as the phase without the use of differential geometry. In addition to that, the new interpretation offers the possibility to detect the intrinsic dimension of local structures. More detailed explanations of this theory were published in [21], [23], and [22] by Wietzke et al.

The major assignment in this thesis is to implement the Riesz transforms up to third order and the new possibilities for feature detection resulting from the theory of the relation between Radon and Riesz transform. For simplicity this approach will from now on be called 'Kiel method'. As already stated, the Kiel method assumes that the images are band limited. This assumption is fulfilled, if a bandpass filter is applied to the image in before the Riesz transform. Felsberg showed in [6] that the 'difference of Poisson' filter is a suitable bandpass filter in combination with the first order Riesz transform. The major task in this thesis is to find convolution kernels for the second and third order Riesz transforms and a bandpass filter that can be combined with them. Amongst other things, this problem is solved by the use of the 'Laplacian of Poisson' as a bandpass filter.

The Kiel method is compared to another method using differential geometry based on image gradients. It has been developed at the university of Lübeck during the last years and will therefore be called 'Lübeck method'. It is a well-known method used for orientation estimation of up to two superimposed oriented patterns. Published works containing the Lübeck method include [19], [2], and [1]. As for the Kiel method, the images need to consist of either one or two intrinsically one-dimensional structures. If there are two structures, they need to be either overlaid or occluding.

In addition to the comparison of both methods, rotation estimation as a practical application is considered. In order to demonstrate the practical usability, the rotation estimation is implemented and tested. It allows the calculation of the rotation angle between two rotated images. The rotation can be derived directly from the Riesz transform theory based on the Radon transform in a very elegant way.

The remainder of this thesis is organized as follows: the next section introduces the concept of intrinsic dimension, local main orientation and local apex angle. Chapter 2 describes the orientation estimation of two overlaid intrinsically one-dimensional structures, first using the Kiel method, then using the Lübeck method. The description of



the underlying theory is followed by an outline of the implementation of both methods in chapter 3. Especially the implementation of the Riesz transforms posed some problems, so the design of suitable filter kernels for the first, second, and third order Riesz transforms is described in detail. Chapter 4 contains a comparison of both methods on synthetic and real images. Chapter 5 describes the rotation estimation between two images. The last chapter 6 gives a conclusion and an outlook.

## 1.1 Orientation and Apex Angles

In this section the features intrinsic dimension, local orientation and apex angle will be introduced. These features and their computation is a major concern of this thesis, therefore this chapter concludes with their introduction.

### 1.1.1 Intrinsic Dimension of Images

An image is defined as a two-dimensional function  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . In the following, the term intrinsic dimension is used to describe the images locally. Therefore the intrinsic dimension is defined within a neighborhood  $U \subset \Omega$ .

In contrast to considering the number of possible degrees of freedom in an n-dimensional structure, intrinsic dimension considers the number of actually used degrees of freedom within a local neighborhood  $U$ .

**Definition 1.1.1.** *Intrinsic dimension* classifies an image  $I$  into local neighborhoods  $U$  (see also [22])

$$I \in \begin{cases} i0D_U, & I(\mathbf{x}_i) = I(\mathbf{x}_j) \quad \forall \mathbf{x}_i, \mathbf{x}_j \in U \\ i1D_U, & I(x, y) = g(x \cos \theta + y \sin \theta) \quad \forall (x, y) \in U \text{ with } g \in \mathbb{R}^{\mathbb{R}} \text{ and } I \notin i0D_U \\ i2D_U, & \text{else.} \end{cases} \quad (1.1)$$

Basically, image areas with constant gray values have intrinsic dimension zero while all lines and edges are intrinsically one-dimensional. Areas that contain corners have intrinsic dimension two. An approach to intrinsic dimension is found in differential geometry (see [3]). Images are considered to be surfaces in a three-dimensional space. The x- and y-axis correspond to the pixel coordinates and the z-axis corresponds to the gray value at the pixel position  $(x, y) \in \mathbb{R}^2$ . A plane part of the surface forms an i0D neighborhood. An i1D neighborhood corresponds to a parabolic structure while elliptic or hyperbolic parts of the surface form neighborhoods with intrinsic dimension two. An analysis of intrinsic dimension can be done using gradients in images. This approach is employed by the Lübeck method described in section 2.2. The interpretation of the higher order Riesz transforms using the Radon transform yields a different access to intrinsic dimension that is independent of differential geometry.



Figure 1.2: Orientation of texture in a picture (from [14])

### 1.1.2 Orientation

The concept of intrinsic dimension is needed when considering local orientations in images. First the term 'direction' in an image has to be regarded. The direction is an angle with respect to the x-axis that defines a line through the origin of the image. Along that line, gray values are constant, or expressed differently, the image gradients are lowest. The term direction is not expedient though, when looking at local structures because direction is defined to have values in  $[0, 2\pi)$ . For further explanation, picture 1.2 can be examined. If the image is rotated by  $180^\circ$  around the origin, direction changes. However, if one just considers a small local vicinity and rotates the local vicinity around its origin, the direction does not change. Therefore the term local orientation is defined for regarding local neighborhoods of an image. Local orientation is denoted by  $\theta \in [-\pi/2, \pi/2)$  and is otherwise defined like direction.

The next step in analyzing orientation in images consists of examining intrinsic dimensions. In a neighborhood with intrinsic dimension one, exactly one orientation can be found. If however, several intrinsically one-dimensional structures are overlaid, several orientations occur. Estimating these or the main orientation of all of them is subject to this thesis.

### 1.1.3 Apex Angles

In case of several overlaid i1D structures, it is possible to estimate apex angles between these image structures. An intersection of two i1D structures yields two angles, the apex angle is defined to be the smaller angle. If the main orientation and the apex angle between two overlaid i1D structures can be estimated, a characterization of the image structures is gained. In addition to that, the local apex angle itself can be used as a rotation invariant feature of a local neighborhood of the image. This feature can be used in a variety of image analysis applications, e.g. texture classification or segmentation of homogeneous regions.

This concludes the introduction. In the next chapter the theory for estimating the local main orientation and apex angle using the Kiel method and using the Lübeck method is described.

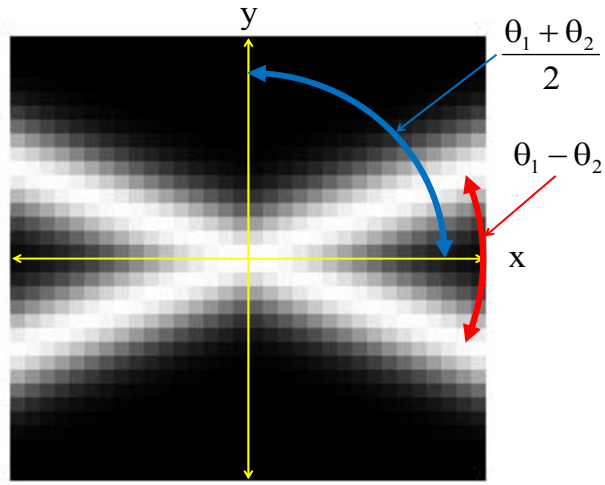


Figure 1.3: The apex angle between the two image structures is defined to be the smaller angle.  $\theta_1$  and  $\theta_2$  are the orientations of the each of the two structures with respect to the x-axis.  $\theta_{main} = \frac{\theta_1 + \theta_2}{2}$  is the average orientation of the two structures with respect to the x-axis (from [23]).



## 2 Orientation Estimation with two Overlaid 1D Structures

This chapter introduces the theory of both, the Kiel method and the Lübeck method.

### 2.1 Using Riesz Transforms - the Kiel Method

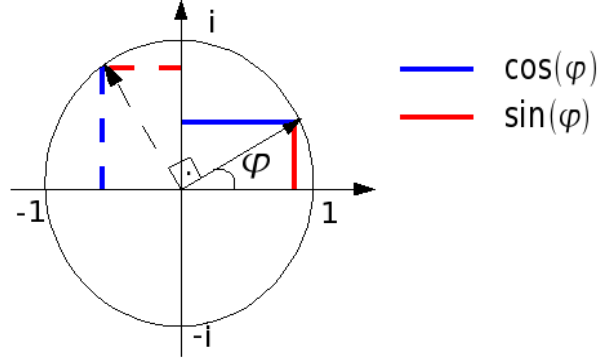
In this section an overview over two possibilities to derive the monogenic signal is given. First, the monogenic signal is considered to be a multi-dimensional extension of the analytic signal. This approach has been proposed by Felsberg in [6]. He also mentioned the possibility to derive the monogenic signal using the Radon transform. This second approach has been researched by Wietzke in [23] and is described in section 2.1.2.

#### 2.1.1 The Monogenic Signal as an Extension of the Analytic Signal

In this section the computation of the monogenic signal and the estimation of the local main orientation are described. There are different ways to derive the monogenic signal. It can be interpreted as a two-dimensional extension of the analytic signal. This approach has been known for a while now and it is introduced in this thesis for a better understanding of the underlying theory. Furthermore it is useful for the derivation of convolution kernels for the implementation of the Riesz transform, which is a major task in this work.

For easier calculation, it is assumed that the images consist of one frequency. An example is the signal  $g(x) = g_0 \cos(kx)$ ,  $k, x \in \mathbb{R}$ , for which the local phase can be calculated. The local phase of  $g(x)$  is the argument of the cosine. Shifting the phase of the signal by  $90^\circ$  transforms the cosine into a sine. By applying the arctangent, the phase of this signal can be determined. Unfortunately, in the real world a signal, especially an image, usually does not have just one frequency. In order to get a band limited signal representing an image, a bandpass filter needs to be applied. If all frequencies but a narrow band are filtered, the assumption that the signal consists of one frequency only is considered to be fulfilled. In addition to that, it is useful to work on the picture in different scale spaces, where the monogenic signal is calculated with pass bands at different scales.

In the following part the analytic signal is introduced. It provides a way for calculating the local phase of one-dimensional signals. The monogenic signal, which will be described after the analytic signal, is considered as the multi-dimensional extension of the analytic signal. In the two following sections, it is assumed that the signals to be processed only consist of a narrow band of frequencies. In section 3.1.1 the theory of the monogenic


 Figure 2.1: Rotating the phase by  $90^\circ$ 

signal and the required bandpass filter to obtain a working set of filters for practical applications are combined.

### The Analytic Signal

The analytic signal can only be computed for one-dimensional signals; the orientation estimation is therefore not expedient. The application of the analytic signal to a one-dimensional signal provides the capability to estimate the phase and the amplitude. Orientation estimation is possible using the extension to higher dimensions in the next section. For now the analytic signal is interpreted as an operator for phase estimation.

In order to calculate the phase of a one-dimensional signal, an operator is needed that shifts the global phase of the original signal by  $90^\circ$ . Since the global phase can be accessed easily in the frequency domain, the needed operator is derived after computing the Fourier transform of the signal  $g : \mathbb{R} \rightarrow \mathbb{R}$ :

$$\mathcal{F}\{g\}(u) = \int_{-\infty}^{\infty} g(x) \exp(-2\pi i u x) dx \text{ with } x, u \in \mathbb{R}. \quad (2.1)$$

Using Euler's formula, this becomes

$$\mathcal{F}\{g\}(u) = \int_{-\infty}^{\infty} g(x) \underbrace{\cos(-2\pi u x)}_{=:\varphi} + i g(x) \underbrace{\sin(-2\pi u x)}_{=:\varphi} dx \text{ with } x, u \in \mathbb{R}. \quad (2.2)$$

The sine and the cosine term have the same argument, the phase  $\varphi$  that is to be calculated. Figure 2.1 shows a signal in the frequency domain and the corresponding result after rotating the phase by  $90^\circ$ . A comparison of the sine and cosine terms of the original and the rotated phase in the image with equation 2.2 reveals that  $\mathcal{F}\{g\}(u)$  has to be multiplied with  $i$  in order to shift the phase by  $90^\circ$ .

The multiplication by  $i$  is done by the Hilbert operator. Its transfer function  $\mathcal{F}\{h\}$  is defined by

$$\mathcal{F}\{h\}(u) = i \operatorname{sign}(u) = i \frac{u}{|u|}, \quad u \in \mathbb{R}. \quad (2.3)$$

Its point spread function (the transform of the filter back into the spatial domain) is

$$h(x) = -\frac{1}{\pi x}, \quad x \in \mathbb{R}. \quad (2.4)$$

By applying the convolution theorem, the convolution of the original function  $g$  with  $h$  computes the function  $g_H$ :

$$\mathcal{F}\{g\}(u)\mathcal{F}\{h\}(u) \bullet \longrightarrow g(x) * h(x) =: g_H(x). \quad (2.5)$$

The analytic signal is then defined as the complex signal

$$g_A(x) := g(x) - i(h * g)(x) = g(x) - ig_H(x), \quad x \in \mathbb{R}. \quad (2.6)$$

The values of phase and amplitude are observed at a local spatial position  $x \in \mathbb{R}$  which is possible because of the assumption that the signal only consists of one frequency.

Using the two components of the analytic signal the local amplitude  $A$  and local phase  $\varphi$  can be computed:

Let  $x \in \mathbb{R}$ .

$$\begin{aligned} A(x) &= |g_A(x)| = \sqrt{g(x)^2 + g_H(x)^2} \\ \varphi(x) &= \arg(g(x) - ig_H(x)) = \text{atan2}\left(\frac{-g_H(x)}{g(x)}\right) = \text{sign}(-g_H(x)) \text{atan}\left(\frac{|g_H(x)|}{g(x)}\right). \end{aligned} \quad (2.7)$$

It is possible to apply the analytic signal to signals with a dimension higher than one. In order to do that, the local orientation has to be known in advance because the Hilbert filter can only be used in one direction. An easier way to process signals of higher dimension is given by the monogenic signal.

## The Monogenic Signal

The monogenic signal is a multi-dimensional extension of the analytic signal and has been introduced by Sommer and Felsberg in [9]. It uses the Riesz transform, a multi-dimensional Hilbert transform, to calculate a signal's amplitude, phase, and orientation. Alternatively, a phase vector combining phase and orientation can be obtained. As seen in the section above, an operator has to be found that shifts the signal's phase by  $90^\circ$ . To accomplish that, the Hilbert operator needs to be turned into a multi-dimensional operator. This is done in the frequency domain, so that equation 2.3 is transformed to

$$\mathcal{F}\{\mathbf{R}\}(\mathbf{u}) = i \frac{\mathbf{u}}{|\mathbf{u}|} = \left( i \frac{u_1}{|\mathbf{u}|}, i \frac{u_2}{|\mathbf{u}|} \right)^T = (\mathcal{F}\{R_x\}(\mathbf{u}), \mathcal{F}\{R_y\}(\mathbf{u}))^T, \quad \mathbf{u} \in \mathbb{R}^2. \quad (2.8)$$

the transfer function of the Riesz transform. It needs to be transformed back into the spatial domain. A closer look at this inverse Fourier transform, though already in combination with a bandpass filter, is given in appendix (B.1). Let  $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ . The result is

$$\mathbf{R}(\mathbf{x}) = -\frac{\mathbf{x}}{2\pi|\mathbf{x}|^3} \Leftrightarrow \begin{pmatrix} R_x(x_1, x_2) \\ R_y(x_1, x_2) \end{pmatrix} = \begin{pmatrix} -\frac{x_1}{2\pi(x_1^2+x_2^2)^{\frac{3}{2}}} \\ -\frac{x_2}{2\pi(x_1^2+x_2^2)^{\frac{3}{2}}} \end{pmatrix}. \quad (2.9)$$

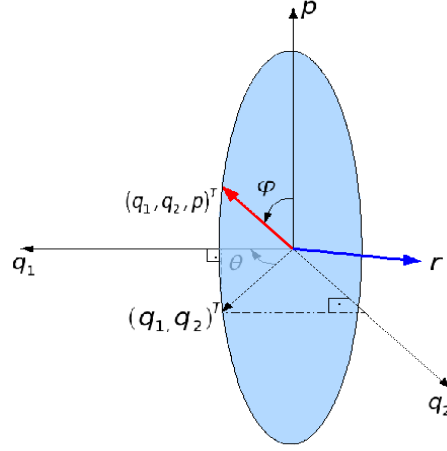


Figure 2.2: The coordinate system defined by the components of the monogenic signal.  $\theta$  and  $\varphi$ , the angles defined by the 3D vector, are the local orientation and local phase while  $\mathbf{r}$  represents the phase vector.

The resulting monogenic signal is a 3D signal because it represents three different features (local phase, local orientation and amplitude):

$$g_M(\mathbf{x}) := (g(x_1, x_2), (R_x * g)(x_1, x_2), (R_y * g)(x_1, x_2))^T, \quad \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2. \quad (2.10)$$

From now on the following abbreviations will be used:

$$\begin{aligned} p &:= g(x_1, x_2) \\ q_1 &:= (R_x * g)(x_1, x_2) \\ q_2 &:= (R_y * g)(x_1, x_2). \end{aligned} \quad (2.11)$$

In the case of the monogenic signal,  $p$  is representing the even part of the signal, while the odd part consists of two components,  $q_1$  and  $q_2$ .

Using this 3D vector, local amplitude, local phase, and local orientation of the monogenic signal have to be defined. In order to do that, consider the 3D coordinate system that is defined by  $p$ ,  $q_1$ , and  $q_2$  as can be seen in figure 2.2. The vector  $(p, q_1, q_2)^T$  is visualized as a red arrow.

**Amplitude:** The amplitude of the monogenic signal is defined similarly to the one of the analytic signal, namely as the norm of the vector  $(p, q_1, q_2)^T$ , or in image 2.2 the length of the red arrow.

$$A_{\mathbf{x}} := \sqrt{p^2 + q_1^2 + q_2^2} \quad (2.12)$$

**Orientation:** The local orientation is obtained by  $q_1$  and  $q_2$ . It is defined as the angle  $\theta$ . As can be seen in figure 2.2, the vectors  $\mathbf{q} = (q_1, q_2)^T$  and  $(p, q_1, q_2)^T$  span a plane,



the blue colored circle. It was already mentioned that the orientation  $\theta$  is in  $[-\pi/2, \pi/2]$ . Since both, the vector  $\mathbf{q}$  and  $-\mathbf{q}$  span the same plane, the values of the orientation  $\theta$  are also within that interval. The local orientation is

$$\theta_{\mathbf{x}} := \text{atan} \left( \frac{q_2(\mathbf{x})}{q_1(\mathbf{x})} \right) \quad \mathbf{x} \in \mathbb{R}^2. \quad (2.13)$$

**Phase:** Compared to the phase of the analytic signal, the phase of the monogenic signal cannot be derived as easily. Here the phase is defined by a 3D vector. Using figure 2.2,  $\varphi$  is the angle between the 3D vector  $(p, q_1, q_2)^T$  and the p-axis. The phase may also be interpreted as the argument of  $p + i\mathbf{q}$  and the remaining problem is the sign of  $\mathbf{q}$ . Therefore, a different method of obtaining the argument is needed:

$$\varphi_{\mathbf{x}} := \arg(p + i \text{sign}(q_1)|\mathbf{q}|) = -\text{sign}(q_1) \text{atan2} \left( \frac{|\mathbf{q}|}{p} \right). \quad (2.14)$$

**The Phase Vector:** In this paragraph, a 2D vector is defined that contains information on both phase and orientation. The benefit is that both components of the vector have values in  $[0, \pi]$ . This definition helps to avoid problems caused by the smaller range of values of the orientation. In figure 2.2 the phase vector is  $\mathbf{r}^T$ , the blue arrow. It is defined as:

$$\mathbf{r} := \frac{\mathbf{q}}{|\mathbf{q}|} \arg(p + i|\mathbf{q}|) = \begin{pmatrix} \frac{q_1}{|\mathbf{q}|} \text{atan2} \left( \frac{|\mathbf{q}|}{p} \right) \\ \frac{q_2}{|\mathbf{q}|} \text{atan2} \left( \frac{|\mathbf{q}|}{p} \right) \end{pmatrix}. \quad (2.15)$$

This vector is perpendicular to the plane spanned by  $\mathbf{q}$  and  $(p, q_1, q_2)^T$ . It is the normal of the blue colored plane in figure 2.2 multiplied by the phase.

This concludes the interpretation of the monogenic signal as a multi-dimensional extension of the analytic signal - it shifts the phase by  $90^\circ$ . Now a different interpretation is introduced. It gives access to more possibilities for feature detection and shows for example why the orientation can be derived as shown in 2.13.

### 2.1.2 The Monogenic Signal in Terms of the Radon Transform

This section is concerned with a new interpretation of the Riesz transform. The derivation via the Radon transform has been described by Wietzke in [23], [22], or [21]. Once again it is assumed that a signal consists of only one frequency (or at least only of a narrow band). Felsberg revealed in [6] that the monogenic signal can be expressed using the Radon transform and its inverse. The first step is to introduce the Radon transform and its inverse. After that, the Riesz transform is redefined.

**Definition 2.1.1.** The *Radon transform* of a continuous function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  is defined as

$$\mathcal{R}\{f\}(t, \theta) = \int_{(x,y) \in \Omega} f(x, y) \delta_0(x \cos \theta + y \sin \theta - t) d(x, y), \quad \theta \in [0, \dots, \pi), t \in \mathbb{R}. \quad (2.16)$$

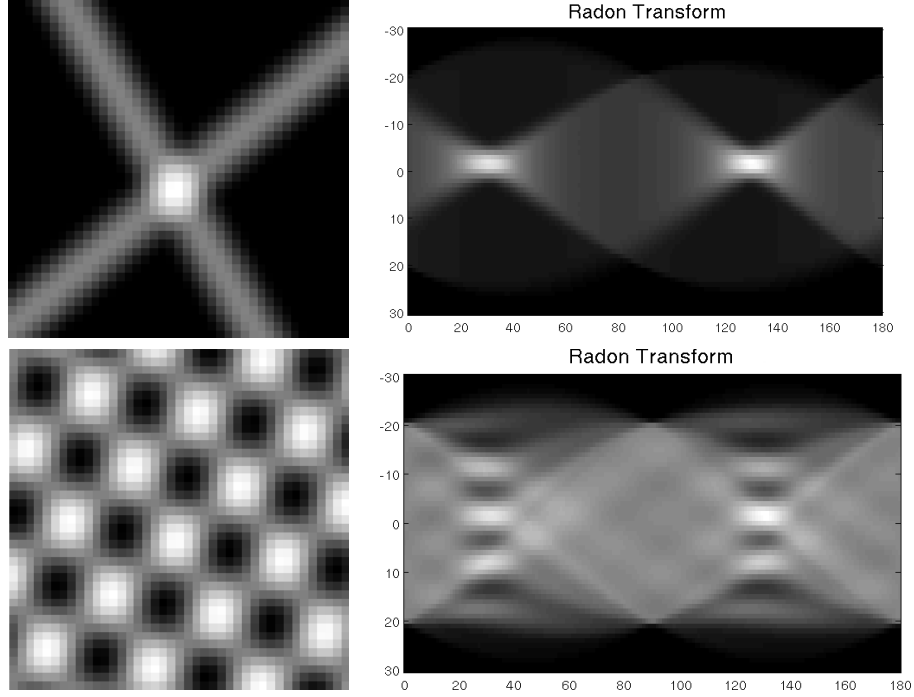


Figure 2.3: Left: Original images. Right: Radon transform of the corresponding images. The Radon transform in the top row shows two peaks which correspond to the lines at angles of  $30^\circ$  and  $130^\circ$  and distances somewhat below zero. In the bottom row the Radon transform has many peaks corresponding to the signal in the original image.

**Definition 2.1.2.** The *inverse Radon transform* of a continuous function  $r : \mathbb{R} \times [0, \dots, \pi) \rightarrow \mathbb{R}$  is given by

$$\mathcal{R}^{-1}\{r(t, \theta)\}(x, y) = \frac{1}{2\pi^2} \int_{\theta=0}^{\pi} \int_{t=-\infty}^{\infty} \frac{\frac{\partial}{\partial t} r(t, \theta)}{x \cos \theta + y \sin \theta - t} dt d\theta. \quad (2.17)$$

Two general properties of the Radon transform (see [20]) are:

- $\mathcal{R}\{\mathcal{R}^{-1}\{r\}\} = r$ , with function  $r : \mathbb{R} \times [0, \dots, \pi) \rightarrow \mathbb{R}$ .
- $\mathcal{R}\{\Sigma_{m \in M} f_m\} = \Sigma_{m \in M} \mathcal{R}\{f_m\}$ , with an index set  $M$  and a function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ .

According to definition 2.1.1 the Radon space, as applied in this thesis, is two-dimensional (see figure 2.3). A point in the Radon space consisting of an angle  $\theta \in [0, \dots, \pi)$  and a distance  $t \in \mathbb{R}$  represents a line in the original image defined by the angle  $\theta$  with respect to the x-axis and the distance  $t$  from the origin  $(0, 0) \in \mathbb{R}^2$ . The Radon transform at an angle  $\theta$  and a distance  $t$  calculates the integral (or in the discrete case, when working on images, the sum) of function values on the corresponding line. Therefore, estimating the Radon transform for intrinsically one-dimensional images yields an integral constant for all angles except for  $\theta_{main}$  which describes the orientation of the 1D structure; the points for different distances  $t \in \mathbb{R}$  for all angles  $\theta \in [0, \dots, \pi) \setminus \theta_{main}$  are constant. The only non-constant column is the one with  $\theta_{main}$ , the angle that equals the orientation

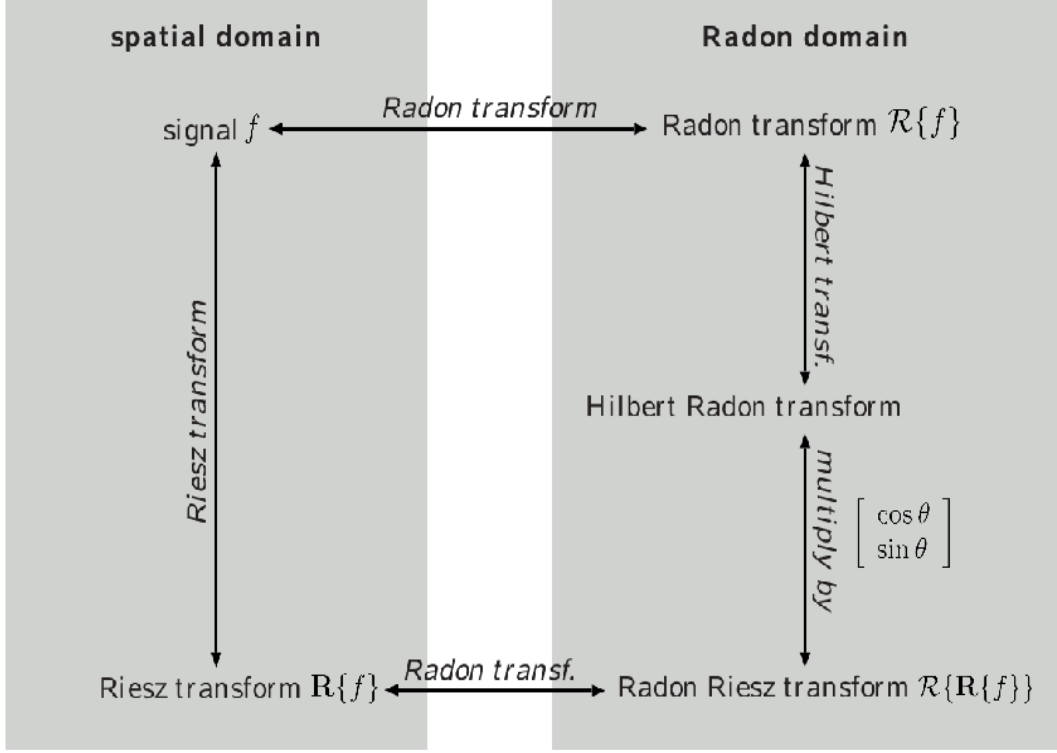


Figure 2.4: The Riesz transform defined through the Radon transform (from [6])

of the i1D structure. In figure 2.3 the Radon transform is calculated for two i2D images that contain two i1D signals with orientations  $30^\circ$  and  $130^\circ$ . The non-constant columns can be observed at those angles in both images. The other columns are not exactly constant, but there are some artifacts. These are due to the finite signals and the discretization. However, in theory this property of constant columns for all angles not being an orientation of the image, is used to describe the Riesz transform using the Radon transform.

**Definition 2.1.3.** Let  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  be an image. The *Riesz transform* at  $(x, y) \in \Omega$  can be defined using the Radon transform, its inverse, and the one-dimensional Hilbert transform  $h_1$ :

$$\begin{aligned} R_x\{I\}(x, y) &:= \mathcal{R}^{-1}\{\cos \theta h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y) \\ R_y\{I\}(x, y) &:= \mathcal{R}^{-1}\{\sin \theta h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y). \end{aligned} \quad (2.18)$$

Proof of the equivalence of this definition to the one given in 2.9 can be found in [6]; it is accomplished by using the central slice theorem. Figure 2.4 depicts the relation between Riesz and Radon transform.

### Interpretation of the Monogenic Signal on i1D Structures

In this paragraph it is assumed that the monogenic signal is used on local neighborhoods that are intrinsically one-dimensional. The orientation within the local neighborhood

can be estimated by  $\theta_{main} = \text{atan}(R_y/R_x)$  (see section 2.1.1). This derivation is done more intuitively when the Riesz transform is defined using the Radon transform.

When determining the orientation  $\theta_{main}$  for an i1D neighborhood,  $\theta_{main}$  equals the orientation of the i1D structure. The Riesz transform is applied to the neighborhood - a set of pixels grouped around the origin  $(0, 0)$  of the neighborhood.

It is sufficient to evaluate definition 2.1.3 at the origin - a fact that allows to simplify the inverse Radon transform to

$$\begin{aligned}
 & \begin{pmatrix} R_x\{I\}(0, 0) \\ R_y\{I\}(0, 0) \end{pmatrix} \\
 &= -\frac{1}{2\pi^2} \int_{\theta=0}^{\pi} \int_{t=-\infty}^{\infty} \frac{1}{0 \cos \theta + 0 \sin \theta - t} \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} h_1(t) * \frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta) dt d\theta \\
 &= \underbrace{\frac{1}{2\pi^2} \int_{t=-\infty}^{\infty} \frac{1}{t} h_1(t) * \frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta) dt}_{=: s_{\theta_{main}}} \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}. \tag{2.19}
 \end{aligned}$$

The outer integral can be eliminated because  $\mathcal{R}\{I\}(t_1, \theta) = \mathcal{R}\{I\}(t_2, \theta) \forall t_1, t_2 \in \mathbb{R}, \forall \theta \in [0, \dots, \pi) \setminus \theta_{main}$  and therefore  $\frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta) = 0 \forall t \in \mathbb{R}, \forall \theta \in [0, \dots, \pi) \setminus \theta_{main}$ .

After this simplification of the inverse Radon transform, it becomes obvious why  $\theta = \text{atan}(R_y/R_x)$  is the orientation: the term called  $s_{\theta_{main}}$  is canceled out.

The term  $s_{\theta_{main}}$  is a result of the new interpretation of the Riesz transform. Therefore the monogenic signal as defined in section 2.1.1 is now extended by a new component:  $s_{\theta_{main}} = \mathcal{R}^{-1}\{h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y) =: q_0$ .

**Definition 2.1.4.** The *monogenic signal* of an image  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  is defined as

$$\begin{pmatrix} p \\ q_0 \\ q_1 \\ q_2 \end{pmatrix} := \begin{pmatrix} I(x, y) \\ \mathcal{R}^{-1}\{h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y) \\ \mathcal{R}^{-1}\{\cos \theta h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y) \\ \mathcal{R}^{-1}\{\sin \theta h_1(t) * \mathcal{R}\{I\}(t, \theta)\}(x, y) \end{pmatrix}. \tag{2.20}$$

In order to calculate the monogenic signal,  $s_{\theta_{main}}$  needs to be computed as well as the Riesz transforms. Wietzke showed that it can be calculated either by a convolution with a Hilbert filter in the direction  $\theta_{main}$  or by using the two parts of the Riesz transform. The two parts of the Riesz transform can be used because of Euler's formula  $\cos^2 \theta_{main} + \sin^2 \theta_{main} = 1$ . The result is

$$\begin{aligned}
 s_{\theta_{main}} &= \sqrt{R_x^2\{I\} + R_y^2\{I\}} \\
 &= \sqrt{(\cos^2(\theta_{main}) + \sin^2(\theta_{main}))s_{\theta_{main}}^2} \\
 &\text{or} \\
 s_{\theta_{main}} &= (h_1 * I_{\theta_{main}})(0) \\
 &= -\frac{1}{\pi} \int_{\tau \in \mathbb{R}} \frac{I(\tau \cos(\theta_{main}), \tau \sin(\theta_{main}))}{\tau} d\tau. \tag{2.21}
 \end{aligned}$$

In section 2.1.1 amplitude and phase also are calculated using the components of the monogenic signal. The calculation of the amplitude remains the same. The introduction of the analytic signal showed that the phase of a one dimensional signal  $f_{1D}$  can be derived by calculating the signal's Hilbert transform and using it in  $\varphi = \text{atan} \left( \frac{h_1 * (f_{1D})}{f_{1D}} \right)$ . Equation (2.21) revealed that  $s_{\theta_{main}}$  equals the Hilbert transform of  $I$  in the direction of the main orientation. Therefore, the phase of  $I$  can be derived by  $\varphi = \text{atan} \left( \frac{s_{\theta_{main}}}{I} \right) = \text{atan} \left( \frac{h_1 * I_{\theta_{main}}}{I} \right)$ .

### Interpretation of the Monogenic Signal on i2D Structures

In the following, i2D structures are considered. After that, a possibility is presented to decide whether a signal is considered to be intrinsically one- or two-dimensional. In this paragraph it is assumed that a i2D signal consists of a finite set  $M := \mathbb{N}_{\leq n}, n \in \mathbb{N}$  of overlaid i1D structures such that  $I = \sum_{m \in M} I_m$ ; each signal  $I_m$ ,  $m \in M$  has its own orientation  $\theta_m$ .

In the i1D case the inner integral can be eliminated (see equation 2.19) since there is only one angle - the main orientation - for which the integral is non-zero. This simplification can now be done for the same reasons, except this time a sum is required in order to regard for each of the  $|M|$  orientations in

$$\begin{aligned} & \begin{pmatrix} R_x\{I\}(0,0) \\ R_y\{I\}(0,0) \end{pmatrix} \\ &= -\frac{1}{2\pi^2} \int_{\theta=0}^{\pi} \int_{t=-\infty}^{\infty} \frac{1}{0 \cos \theta + 0 \sin \theta - t} \begin{pmatrix} \cos \theta_m \\ \sin \theta_m \end{pmatrix} h_1(t) * \frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta) dt d\theta \\ &= \frac{1}{2\pi^2} \sum_{m \in M} \begin{pmatrix} \cos \theta_m \\ \sin \theta_m \end{pmatrix} \int_{t=-\infty}^{\infty} \frac{1}{t} h_1(t) * \frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta_m) dt. \end{aligned} \quad (2.22)$$

For each overlaid i1D signal  $I_m$ ,  $m \in M$ , the corresponding  $s_{\theta_m}$  term is defined as

$$\begin{aligned} s_{\theta_m} &:= \frac{1}{2\pi^2} \int_{t=-\infty}^{\infty} \frac{1}{t} h_1(t) * \frac{\partial}{\partial t} \mathcal{R}\{I\}(t, \theta_m) dt \\ &= \mathcal{R}^{-1}\{h_1(t) * \mathcal{R}\{I\}(t, \theta_m)\}. \end{aligned} \quad (2.23)$$

In order to estimate the main orientation for overlaid structures of i1D signals in the same way as described for the i1D case, a further assumption is required:  $s_{\theta_m} = s_{\theta_n} = s_{\theta}$ ,  $\forall m, n \in M$  - meaning that all the overlaid i1D signals have an arbitrary, but same phase. In this case the main orientation  $\theta_{main}$  is calculated by

$$\begin{aligned} \text{atan} \left( \frac{R_y\{I\}(0,0)}{R_x\{I\}(0,0)} \right) &= \text{atan} \left( \frac{\sum_{m \in M} \cos \theta_m}{\sum_{m \in M} \sin \theta_m} \right) \\ &= \frac{\sum_{m \in M} \theta_m}{|M|}. \end{aligned} \quad (2.24)$$

In case of only two overlaid i1D signals, it is of interest to estimate the apex angle between the two signals: it is a rotationally invariant feature and combined with the main orientation allows the calculation of the two orientations  $\theta_1$  and  $\theta_2$ .

The apex angle between two overlaid i1D signals with arbitrary, but same phase is derived by

$$\begin{aligned}
 & R_x^2\{I\} + R_y^2\{I\} \\
 &= s_{\theta_{main}}^2 (\cos \theta_1 + \cos \theta_2)^2 + s_{\theta_{main}}^2 (\sin^2 \theta_1 + \sin^2 \theta_2) \\
 &= s_{\theta_{main}}^2 (\cos^2 \theta_1 + 2 \cos \theta_1 \cos \theta_2 + \cos^2 \theta_2 + \sin^2 \theta_1 + 2 \sin \theta_1 \sin \theta_2 + \sin^2 \theta_2) \\
 &= s_{\theta_{main}}^2 (2 + 2 \cos(\theta_1 - \theta_2)) \\
 &\Leftrightarrow \\
 &\text{acos} \left( \frac{R_x^2\{I\} + R_y^2\{I\}}{2s_{\theta_{main}}^2} - 1 \right) = \theta_1 - \theta_2.
 \end{aligned} \tag{2.25}$$

This concludes the interpretation of the monogenic signal by means of the Radon transform. This theory results in a possibility to compute the phase of i1D signals and explains in more detail, why the orientation can be calculated using  $\text{atan} \left( \frac{R_y\{I\}}{R_x\{I\}} \right)$ . In addition to that, it provides a new way to analyze i2D structures even if additional assumptions are needed: the superposition of i1D signals in contrast to arbitrary i2D signals and the assumption that  $s_{\theta_m} = s_{\theta_n} = s_\theta$ ,  $\forall m, n \in M$ . The relation of  $s_\theta$  to the phase translates to all overlaid signals having the same, but arbitrary phase.

The next section uses higher order Riesz transforms in order to define the monogenic curvature tensor.

### 2.1.3 The Monogenic Curvature Tensor in Terms of the Radon Transform

The monogenic curvature tensor has been proposed by Zang in [24]. She has developed the monogenic curvature tensor in the Fourier domain which made interpretation difficult but showed a strong resemblance to differential geometry, especially the Gaussian curvature. In this work the monogenic curvature tensor is described in the spatial domain using the Radon transform. To achieve that, the monogenic curvature tensor needs to be understood as the composition of two or three Riesz transforms. A more detailed derivation of the monogenic curvature tensor has been done by Oliver Fleischmann in his student research project [13]. In order to define the monogenic curvature tensor the image needs to be embedded as a Monge patch:

$$\begin{aligned}
 & (x, y) \in \Omega \subset \mathbb{R}^2 \rightarrow (x\mathbf{e}_1, y\mathbf{e}_2, f(x, y)\mathbf{e}_3) \text{ with} \\
 & \{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{12}, \mathbf{e}_{13}, \mathbf{e}_{23}, \mathbf{e}_{123}\} \text{ being a set of basis vectors of Clifford algebra } \mathbb{R}_3.
 \end{aligned} \tag{2.26}$$

In this thesis, Clifford algebra is only used for the definition of the even and the odd tensor. For further calculations, Clifford algebra is not required and is therefore not explained here. For detailed introductions see [24] or [6].

The motivation for defining a monogenic curvature tensor comes from an examination of the Hessian matrix in the frequency domain. The Hessian matrix is used to get an analog to the Gaussian curvature and to the mean curvature (see [18]) and therefore provides access to surface theory. If the Hessian matrix of an image  $I$  is transformed into the frequency domain and viewed in polar coordinates, it reads

$$\begin{aligned} \mathcal{F}\{H_I\} &= \mathcal{F}\left\{\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix}\right\} \\ &= -k \begin{pmatrix} \cos^2(\theta)\mathcal{F}\{I\} & \cos(\theta)\sin(\theta)\mathcal{F}\{I\} \\ \cos(\theta)\sin(\theta)\mathcal{F}\{I\} & \sin^2(\theta)\mathcal{F}\{I\} \end{pmatrix}, \end{aligned} \quad (2.27)$$

with  $k$  being the radial part. Considering successive applications of Riesz transforms to an image in polar coordinates in the frequency domain, yields the following tensor:

**Definition 2.1.5.** The even part of the *monogenic curvature tensor* is defined as

$$\begin{aligned} \mathcal{F}\{T_{even}\} &= \begin{bmatrix} \mathcal{F}\{R_x\{R_x\{I\}\}\} & \mathcal{F}\{-R_x\{R_y\{I\}\}\}\mathbf{e}_{12} \\ \mathcal{F}\{R_x\{R_y\{I\}\}\}\mathbf{e}_{12} & \mathcal{F}\{R_y\{R_y\{I\}\}\} \end{bmatrix} \\ &= \begin{bmatrix} -\cos^2(\theta)\mathcal{F}\{I\} & -\cos(\theta)\sin(\theta)\mathcal{F}\{I\}\mathbf{e}_{12} \\ -\cos(\theta)\sin(\theta)\mathcal{F}\{I\}\mathbf{e}_{12} & -\sin^2(\theta)\mathcal{F}\{I\} \end{bmatrix}. \end{aligned} \quad (2.28)$$

When ignoring the radial part  $-k$ , this tensor resembles the Hessian matrix in the frequency domain. In the spatial domain using the Radon transform, the entries of the tensor can be simplified by using  $h_1(t) * h_1(t) * I = -I$ :

$$\begin{aligned} R_x\{R_x\{I\}\} &= \mathcal{R}^{-1}\{\cos\theta h_1(t) * \mathcal{R}\{\mathcal{R}^{-1}\{\cos\theta h_1(t) * \mathcal{R}\{I\}\}\}\} \\ &= \mathcal{R}^{-1}\{-\cos^2\theta \mathcal{R}\{I\}\} \\ R_x\{R_y\{I\}\} &= \mathcal{R}^{-1}\{-\cos\theta \sin\theta \mathcal{R}\{I\}\} \\ R_y\{R_y\{I\}\} &= \mathcal{R}^{-1}\{-\sin^2\theta \mathcal{R}\{I\}\}. \end{aligned} \quad (2.29)$$

In surface theory the determinant of the Hessian - as an analog to the Gaussian curvature - is used to distinguish between i1D and i2D signals. However, the determinant of the even tensor is not equivalent to the Gaussian curvature: it cannot be interpreted accordingly, though later in this section it will be derived that the tensor does still serve as a measure for intrinsic dimension when interpreted on i2D signals.

Applying a third Riesz transform to the even curvature tensor, yields the odd tensor.

**Definition 2.1.6.** The odd part of the *monogenic curvature tensor* is defined using  $T_{even}\{R_x\{I\} + R_y\{I\}\mathbf{e}_{12}\}$ :

$$T_{odd} = \begin{bmatrix} R_x\{R_x\{R_x\{I\} + R_y\{I\}\mathbf{e}_{12}\}\} & -R_x\{R_y\{R_x\{I\} + R_y\{I\}\mathbf{e}_{12}\}\}\mathbf{e}_{12} \\ R_x\{R_y\{R_x\{I\} + R_y\{I\}\mathbf{e}_{12}\}\}\mathbf{e}_{12} & R_y\{R_y\{R_x\{I\} + R_y\{I\}\mathbf{e}_{12}\}\} \end{bmatrix}. \quad (2.30)$$

This tensor contains third order Riesz transforms, which can again be simplified to a certain degree

$$\begin{aligned} R_x\{R_x\{R_x\{I\}\}\} &= \mathcal{R}^{-1}\{-\cos^3\theta h_1(t) * \mathcal{R}\{I\}\} \\ R_x\{R_x\{R_y\{I\}\}\} &= \mathcal{R}^{-1}\{-\cos^2\theta \sin\theta h_1(t) * \mathcal{R}\{I\}\} \\ R_x\{R_y\{R_y\{I\}\}\} &= \mathcal{R}^{-1}\{-\cos\theta \sin^2\theta h_1(t) * \mathcal{R}\{I\}\} \\ R_y\{R_y\{R_y\{I\}\}\} &= \mathcal{R}^{-1}\{-\sin^3\theta h_1(t) * \mathcal{R}\{I\}\}. \end{aligned} \quad (2.31)$$

This concludes the definition of the monogenic curvature tensor. The next two paragraphs contain an interpretation of the tensor on i1D and i2D structures. In each case, the higher order Riesz transforms are used to derive the local main orientation. The apex angle is estimated if two i1D structures are superimposed. In addition to that, the phase is calculated. However, in case of i2D signals, the constraint that the two overlaid signals have an arbitrary, but same phase is needed again.

### Interpretation of the Monogenic Curvature Tensor on i1D Structures

Section 2.1.2 shows that in case of intrinsically one dimensional signals, both parts of the first order Riesz transform can be simplified. This is accomplished by first eliminating the inner integral of the inverse Radon transform according to the number of orientations present in the signal. In a second step, the cosine and sine parts can be pulled out of the remaining integral of the inverse Radon transform which is possible because of the linearity of the integral. The application of these arguments to the already simplified second order Riesz transform in equation 2.29 results in an elimination of the Radon transform and the Hilbert transform because of the identity  $\mathcal{R}\{\mathcal{R}^{-1}\{I\}\} = I$  for an image  $I$  and the Hilbert transform property:  $h_1(t) * h_1(t) * I = -I$ . The result is

$$T_{even} = -I(0,0) \begin{bmatrix} \cos^2(\theta) & -\cos\theta \sin\theta \mathbf{e}_{12} \\ \cos\theta \sin\theta \mathbf{e}_{12} & \sin^2\theta \end{bmatrix}. \quad (2.32)$$

The third order Riesz transforms in the odd tensor can be simplified using the same arguments but in this case the term  $s_{\theta_{main}}$  cannot be eliminated:

$$\begin{aligned} T_{odd} &= -\underbrace{\mathcal{R}^{-1}\{h_1(t) * \mathcal{R}\{I\}\}}_{=s_{\theta_{main}}} \\ &\begin{bmatrix} \cos^2\theta(\cos\theta + \sin\theta \mathbf{e}_{12}) & -\cos\theta \sin\theta(\cos\theta + \sin\theta \mathbf{e}_{12})\mathbf{e}_{12} \\ \cos\theta \sin\theta(\cos\theta + \sin\theta \mathbf{e}_{12})\mathbf{e}_{12} & \sin^2\theta(\cos\theta + \sin\theta \mathbf{e}_{12}) \end{bmatrix}. \end{aligned} \quad (2.33)$$



From now on the single components of both tensors will be referred to as e.g.  $T_{even_{11}}$  or  $T_{odd_{12}}$ , the common way to refer to matrix entries.

In the i1D case, local (main) orientation derivation is possible in three different ways using the even tensor:

$$\left. \begin{aligned} \theta_{main} &= \text{atan} \left( \sqrt{\frac{T_{even_{22}}}{T_{even_{11}}}} \right) = \text{atan} \left( \sqrt{\frac{-I \sin^2 \theta}{-I \cos^2 \theta}} \right) \\ \theta_{main} &= \text{atan} \left( \frac{T_{even_{21}}}{T_{even_{11}}} \right) = \text{atan} \left( \frac{-I \cos \theta \sin \theta}{-I \cos^2 \theta} \right) \\ \theta_{main} &= \text{atan} \left( \frac{T_{even_{22}}}{T_{even_{21}}} \right) = \text{atan} \left( \frac{-I \sin^2 \theta}{-I \cos \theta \sin \theta} \right) \end{aligned} \right\} = \text{atan} \left( \frac{\sin \theta}{\cos \theta} \right). \quad (2.34)$$

The trace of the even tensor allows the reconstruction of the signal  $I$ ,

$$T_{even_{11}} + T_{even_{22}} = (-I \cos^2 \theta) + (-I \sin^2 \theta) = -I(\cos^2 \theta + \sin^2 \theta) = -I. \quad (2.35)$$

The monogenic curvature tensor and its higher order Riesz transforms yield an alternative for phase calculation using the traces of the even and the odd tensor and the relation between the phase and  $s_{\theta_{main}}$ :

$$\begin{aligned} \varphi &= \text{atan} \left( \frac{\text{trace}(T_{oddy})}{-\sin \theta_{main} \text{trace}(T_{even})} \right) \\ &= \text{atan} \left( \frac{\cos^2(\theta_{main}) \sin(\theta_{main}) s_{\theta_{main}} + \sin^3(\theta_{main}) s_{\theta_{main}}}{\sin(\theta_{main}) I} \right) \\ &= \text{atan} \left( \frac{s_{\theta_{main}}}{I} \right), \end{aligned} \quad (2.36)$$

with  $T_{oddy}$  denoting the tensor that is retrieved by ignoring the parts that do not have a  $\mathbf{e}_{12}$  component. By means of Clifford algebra the odd tensor can be split into two parts  $T_{odd} = T_{oddx} + T_{oddy} \mathbf{e}_{12}$ .

### Interpretation of the Monogenic Curvature Tensor on i2D Structures

In this paragraph i2D signals are examined. It is again assumed that an i2D signal is composed of two overlaid i1D signals and that both signals have arbitrary, but same phases. In this situation it is of interest to calculate both orientations  $\theta_1$  and  $\theta_2$  in addition to the local main orientation and the apex angle. It will later be seen that this task can not be accomplished unambiguously. When applying the monogenic curvature tensor to such signals, the determinants and traces of the tensor are useful. At first, the entries of both tensors need to be simplified under the assumption of two orientations. The steps of simplifying are basically the same ones that are taken in the i1D case.

The even tensor is reduced to

$$T_{even} = -I(0, 0) \begin{bmatrix} T_{even11} & T_{even12} \\ T_{even21} & T_{even22} \end{bmatrix}$$

with

$$\begin{aligned} T_{even11} &= \cos^2 \theta_1 + \cos^2 \theta_2 \\ T_{even12} &= -(\cos \theta_1 \sin \theta_1 + \cos \theta_2 \sin \theta_2) \mathbf{e}_{12} \\ T_{even21} &= (\cos \theta_1 \sin \theta_1 + \cos \theta_2 \sin \theta_2) \mathbf{e}_{12} \\ T_{even22} &= \sin^2 \theta_1 + \sin^2 \theta_2. \end{aligned} \quad (2.37)$$

While the odd tensor is simplified to

$$T_{odd} = - \underbrace{\mathcal{R}^{-1}\{h_1(t) * \mathcal{R}\{I\}\}}_{=s_{\theta_{main}}} \begin{bmatrix} T_{odd11} & T_{odd12} \\ T_{odd21} & T_{odd22} \end{bmatrix}$$

with

$$\begin{aligned} T_{odd11} &= (\cos^3 \theta_1 + \cos^3 \theta_2) + (\cos^2 \theta_1 \sin \theta_1 + \cos^2 \theta_2 \sin \theta_2) \mathbf{e}_{12} \\ T_{odd12} &= -(\cos^2 \theta_1 \sin \theta_1 + \cos^2 \theta_2 \sin \theta_2) \mathbf{e}_{12} + (\cos \theta_1 \sin^2 \theta_1 + \cos \theta_2 \sin^2 \theta_2) \\ T_{odd21} &= (\cos^2 \theta_1 \sin \theta_1 + \cos^2 \theta_2 \sin \theta_2) \mathbf{e}_{12} - (\cos \theta_1 \sin^2 \theta_1 + \cos \theta_2 \sin^2 \theta_2) \\ T_{odd22} &= (\cos \theta_1 \sin^2 \theta_1 + \cos \theta_2 \sin^2 \theta_2) + (\sin^3 \theta_1 + \sin^3 \theta_2) \mathbf{e}_{12}. \end{aligned} \quad (2.38)$$

In order to interpret these tensors, the determinant and trace of the even tensor are derived and simplified:

$$\begin{aligned} \det(T_{even}) &= I^2(0, 0)(\cos^2 \theta_1 \sin^2 \theta_2 + \cos^2 \theta_2 \sin^2 \theta_1 - 2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2) \\ &= I^2(0, 0)(\cos \theta_1 \sin \theta_2 - \cos \theta_2 \sin \theta_1)^2 \\ &= I^2(0, 0) \sin^2(\theta_1 - \theta_2) \\ \text{trace}(T_{even}) &= -I(0, 0)(\cos^2 \theta_1 + \cos^2 \theta_2 + \sin^2 \theta_1 + \sin^2 \theta_2) \\ &= -2 \cdot I(0, 0) \end{aligned} \quad (2.39)$$

This leads to the conclusion that the trace reconstructs the signal just like it does in the i1D case. The determinant on the other hand allows apex angle estimation, but it also becomes obvious that it can distinguish i1D and i2D structures. If  $\theta_1 = \theta_2$ , meaning the two overlaid signals have equal orientation, the determinant equals zero. However, signals with equal orientation are equivalent to the i1D case. The conclusion is

$$\det T_{even} \begin{cases} = 0 & I \text{ is an i1D signal} \\ > 0 & I \text{ is an i2D signal.} \end{cases} \quad (2.40)$$

The odd part of the tensor can be split into  $T_{odd} = T_{oddx} + T_{oddy} \mathbf{e}_{12}$ : it is also true for the determinant and trace of the odd part of the tensor. Both parts of the determinant and trace yield options for estimating features like local phase, orientation, and apex angle.

The application of transformations to equation 2.38 results in the following determinant for the odd tensor:

$$\begin{aligned}
 \mathbf{e}_1 \det(T_{odd}) &=: \mathbf{e}_1 s_{\theta_{main}}^2 B + \mathbf{e}_2 s_{\theta_{main}}^2 C \\
 B &= (\cos \theta_1 \sin \theta_2 - \cos \theta_2 \sin \theta_1) \\
 &\quad (\cos^2 \theta_1 \sin^2 \theta_2 + \cos^2 \theta_2 \sin^2 \theta_1 - 2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2) \\
 &= \cos(\theta_1 + \theta_2) \sin^2(\theta_1 - \theta_2) \\
 C &= (\cos \theta_1 \sin \theta_2 + \cos \theta_2 \sin \theta_1) \\
 &\quad (\cos^2 \theta_1 \sin^2 \theta_2 + \cos^2 \theta_2 \sin^2 \theta_1 - 2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2) \\
 &= \sin(\theta_1 + \theta_2) \sin^2(\theta_1 - \theta_2).
 \end{aligned} \tag{2.41}$$

Calculating the trace of the odd tensor is far more straight forward:

$$\begin{aligned}
 \mathbf{e}_1 \text{trace}(T_{odd}) &= \mathbf{e}_1 s_{\theta_{main}}^2 ((\cos^3 \theta_1 + \cos^3 \theta_2) + (\cos^2 \theta_1 \sin \theta_1 + \cos^2 \theta_2 \sin \theta_2 \mathbf{e}_{12}) \\
 &\quad + (\cos \theta_1 \sin^2 \theta_1 + \cos \theta_2 \sin^2 \theta_2) + (\sin^3 \theta_1 + \sin^3 \theta_2) \mathbf{e}_{12}) \\
 &= \mathbf{e}_1 s_{\theta_{main}}^2 (\cos^3 \theta_1 + \cos^3 \theta_2 + \cos \theta_1 \sin^2 \theta_1 + \cos \theta_2 \sin^2 \theta_2) \\
 &\quad + \mathbf{e}_2 s_{\theta_{main}}^2 (\cos^2 \theta_1 \sin \theta_1 + \cos^2 \theta_2 \sin \theta_2 + \cos^3 \theta_1 + \cos^3 \theta_2) \\
 &= \mathbf{e}_1 s_{\theta_{main}}^2 (\cos \theta_1 + \cos \theta_2) \\
 &\quad + \mathbf{e}_2 s_{\theta_{main}}^2 (\sin \theta_1 + \sin \theta_2) \\
 &= \mathbf{e}_1 s_{\theta_{main}}^2 2 \cos\left(\frac{\theta_1 + \theta_2}{2}\right) \cos\left(\frac{\theta_1 - \theta_2}{2}\right) \\
 &\quad + \mathbf{e}_2 s_{\theta_{main}}^2 2 \sin\left(\frac{\theta_1 + \theta_2}{2}\right) \cos\left(\frac{\theta_1 - \theta_2}{2}\right).
 \end{aligned} \tag{2.42}$$

Obviously, there are miscellaneous possibilities to use the trace and determinant of both, the even and the odd part of the tensor, for apex angle and local main orientation calculation.

The determinant and trace of the odd part of the tensor offer possibilities to estimate the main orientation  $\theta_{main}$  by

$$\begin{aligned}
 \theta_{main} &= \frac{1}{2} \text{atan} \left( \frac{s_{\theta_{main}}^2 C}{s_{\theta_{main}}^2 B} \right) \\
 &= \frac{1}{2} \text{atan} \left( \frac{\sin(\theta_1 + \theta_2) \sin^2(\theta_1 - \theta_2)}{\cos(\theta_1 + \theta_2) \sin^2(\theta_1 - \theta_2)} \right) \\
 &= \frac{1}{2} \text{atan} (\tan(\theta_1 + \theta_2)) \\
 \theta_{main} &= \text{atan} \left( \frac{s_{\theta_{main}} 2 \sin\left(\frac{\theta_1 + \theta_2}{2}\right) \cos\left(\frac{\theta_1 - \theta_2}{2}\right)}{s_{\theta_{main}} 2 \cos\left(\frac{\theta_1 + \theta_2}{2}\right) \cos\left(\frac{\theta_1 - \theta_2}{2}\right)} \right) \\
 &= \text{atan} \left( \tan \left( \frac{\theta_1 + \theta_2}{2} \right) \right).
 \end{aligned} \tag{2.43}$$

The second formula for orientation estimation is derived by using the two components of the trace of the odd part of the tensor. It is obvious that this formula also works in the i1D case when  $\theta_1 = \theta_2$ . This is not true for the determinant and a direct result of the new interpretation. Because of the invariance under intrinsic dimension, in this thesis the trace is used for orientation estimation.

The apex angle can also be estimated in several ways. The best choice is to use the determinant and trace of the even part of the monogenic curvature tensor in

$$\begin{aligned}
 |\theta_1 - \theta_2| &= 2 \operatorname{atan} \left( \sqrt{\left| \frac{1 - \sqrt{1 - \det(T_{\text{even}})}}{1 + \sqrt{1 - \det(T_{\text{even}})}} \right|} \right) \\
 |\theta_1 - \theta_2| &= 2 \operatorname{atan} \left( \sqrt{\left| \frac{\operatorname{trace}(T_{\text{even}}) - \sqrt{\operatorname{trace}(T_{\text{even}})^2 - \det(T_{\text{even}})}}{\operatorname{trace}(T_{\text{even}}) + \sqrt{\operatorname{trace}(T_{\text{even}})^2 - \det(T_{\text{even}})}} \right|} \right) \\
 \theta_1 - \theta_2 &= \operatorname{asin} \left( I(0, 0) \sqrt{\det(T_{\text{even}})} \right). \tag{2.44}
 \end{aligned}$$

The estimation of  $s_{\theta_{\text{main}}}$  or the related local phase in the i2D case is also possible by using the monogenic curvature tensor.

It can be shown that the phase of the signal is:  $\varphi = \operatorname{atan2} \left( \frac{|\det(T_{\text{odd}})|}{\det(T_{\text{even}})} \right)$ . The relation between  $s_{\theta_{\text{main}}}$  and the phase in the i2D case is

$$\varphi = \operatorname{atan2} \left( \frac{s_{\theta_{\text{main}}}}{I^2} \right) = \operatorname{atan2} \left( \frac{h_1 * I_{\theta_{\text{main}}}}{I^2} \right). \tag{2.45}$$

These equations for calculating the phase can be used to calculate  $s_{\theta_{\text{main}}}$  which in turn can be applied to equation (2.25) in order to estimate the apex angle.

This description of the theory of the monogenic signal and monogenic curvature tensor in terms of the Radon transform is not very detailed, but still quite long. That is why more extensive derivations are skipped. They can be found in Fleischmann's [13] work. His student research project's main task was the derivation and proof of parts of the theory, while this thesis' main concern is the implementation and comparison to another method next to some practical tests.

The following table summarizes the most important results that will later on be implemented and tested.

	i1D	i2D
$\theta_{main} = \text{atan}\left(\frac{R_x}{R_y}\right)$	x	x
$\theta_{main} = \text{atan}\left(\sqrt{\frac{R_y R_y}{R_x R_x}}\right)$	x	-
$\theta_{main} = \text{atan}\left(\frac{R_x R_y}{R_x R_x}\right)$	x	-
$\theta_{main} = \text{atan}\left(\frac{R_y R_y}{R_x R_x}\right)$	x	-
$\theta_{main} = \text{atan}\left(\frac{\det(T_{odd_x})}{\det(T_{odd_y})}\right)$	-	x
$\theta_{main} = \text{atan}\left(\frac{\text{trace}(T_{odd_y})}{\text{trace}(T_{odd_x})}\right)$	x	x
$\theta_1 - \theta_2 = \arccos\left(\frac{R_x^2 + R_y^2}{2s^2} - 1\right)$	-	x
$ \theta_1 - \theta_2  = 2 * \text{atan}\left(\sqrt{\left \frac{\text{trace}(T_{even}) - \sqrt{\text{trace}(T_{even})^2 - \det(T_{even})}}{\text{trace}(T_{even}) + \sqrt{\text{trace}(T_{even})^2 - \det(T_{even})}}\right }\right)$	-	x
$ \theta_1 - \theta_2  = 2 * \text{atan}\left(\sqrt{\left \frac{1 - \sqrt{1 - \det(T_{even})}}{1 + \sqrt{1 - \det(T_{even})}}\right }\right)$	-	x
$\theta_1 - \theta_2 = \arcsin\left(\sqrt{\frac{4 \det(T_{even})}{I(0,0)^2}}\right)$	-	x
$s = (h_1 * I_{\theta_{main}})(0)$	x	-
$\varphi = \text{atan}\left(\frac{s}{f}\right)$	x	-
$\varphi = \text{atan2}\left(\frac{ \det(T_{odd}) }{\det(T_{even})}\right)$	-	x
$-I = R_x R_x + R_y R_y$	x	-
$-2 \cdot I = R_x R_x + R_y R_y$	-	x

In order to implement this theory a suitable bandpass filter is needed to assure that the images are bandlimited. The derivation of such a bandpass filter and the implementation are depicted in chapter 3. In this more theoretical chapter the derivation of the Lübeck method will follow.

## 2.2 Using Gradients - the Lübeck Method

The Lübeck method has been described in several papers including [19] [1], and [2]. It uses image gradients calculated over a local neighborhood to estimate the local orientation. This method works in two steps. First the orientation is calculated assuming there is only one orientation - an i1D structure in the neighborhood. After that, a confidence measure is applied to check if the assumption holds. If it does not hold, the second step consists of estimating two orientations under the assumption, that the local neighborhood consists of two overlaid or occluding i1D structures. Figure 2.5 visualizes this approach.

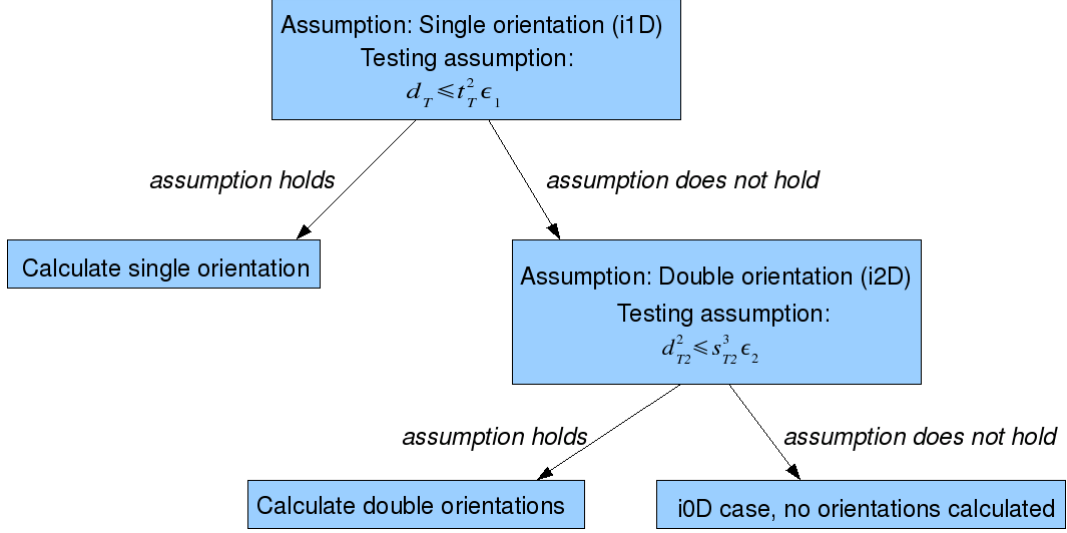


Figure 2.5: Testing for intrinsic dimension in the Lübeck method.

### 2.2.1 Single Orientation Estimation

Let  $I(\mathbf{x})$ ,  $\mathbf{x}^T = (x, y) \in \mathbb{R}^2$  be a two-dimensional function representing a gray-level image. The main orientation in a local neighborhood  $\Omega$  of this image can be described using  $\theta$ . A derivative operator in the direction of the main orientation can be defined as

$$\alpha(\theta) := \cos(\theta) \frac{\partial}{\partial x} + \sin(\theta) \frac{\partial}{\partial y}. \quad (2.46)$$

When considering a local neighborhood  $\Omega$  of an image  $I$  (with non-constant gray values) and  $(x, y) \in \mathbb{R}^2$  being the origin of the neighborhood, there exists an ideal orientation  $\theta$  if

$$\begin{aligned} \alpha(\theta)I(x, y) &= 0 \\ \iff \mathbf{v}^T \nabla I(x, y) &= 0 \end{aligned} \quad (x, y) \in \Omega, \quad (2.47)$$

with  $\mathbf{v}^T = (\cos(\theta), \sin(\theta))$  and  $\nabla := \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T$ . However, due to noise, discretization, etc., there usually is no ideal orientation. Therefore, orientation estimation is done by minimizing the following integral over the local neighborhood  $\Omega$  for which the orientation has to be calculated:

$$\begin{aligned} \mathbf{v} = \operatorname{argmin} Q(\mathbf{v}) &= \operatorname{argmin} \left( \int_{\Omega} (\mathbf{v}^T \nabla I)^2 d\Omega \right) \\ &= \operatorname{argmin} \left( \mathbf{v}^T \int_{\Omega} \nabla I (\nabla I)^T d\Omega \mathbf{v} \right) \\ &= \operatorname{argmin} \left( \mathbf{v}^T \mathbf{T} \mathbf{v} \right), \end{aligned} \quad (2.48)$$

under the additional criterion  $\mathbf{v}^T \mathbf{v} = 1$ . Using  $I_x, I_y$  as derivatives of  $I$  in x- and y-direction,  $\mathbf{T}$  can be written as

$$\mathbf{T} = \int_{\Omega} \nabla I (\nabla I)^T d\Omega = \int_{\Omega} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} d\Omega. \quad (2.49)$$

It is possible to use a weighting function, so that pixels further away from the origin of  $\Omega$  are weighted less than closer pixels. However, at the moment, it does not matter if such a weighting function is considered or not and is therefore omitted for now. Later in the implementation it will be used again.

The composition of both criteria leads to minimizing

$$\begin{aligned} \operatorname{argmin}(L(\mathbf{v})) &= \operatorname{argmin}(\mathbf{v}^T \mathbf{T} \mathbf{v} + \lambda_2(1 - \mathbf{v}^T \mathbf{v})) \\ \iff \mathbf{T} \mathbf{v} &= \lambda_2 \mathbf{v} \text{ with } \mathbf{v}^T \mathbf{v} = 1. \end{aligned} \quad (2.50)$$

This equation is solved for  $\mathbf{v}$  with  $\mathbf{v}$  being the corresponding eigenvector to the lowest eigenvalue  $\lambda_2$ . If there exists an ideal orientation in the local neighborhood  $\Omega$ , the matrix  $\mathbf{T}$  has rank one and the lower eigenvalue  $\lambda_2$  equals zero. When working on real images, it is sufficient for  $\lambda_2$  to be close to zero for assuming that a single orientation is found. The eigenvector  $\mathbf{v}$  points in the direction, where the gray values do not change. Therefore, a confidence measure is needed to determine if there is enough evidence for a single orientation. The confidence measure is constructed using the determinant and the trace of  $\mathbf{T}$ . Let  $\mathbf{T}_{11}$ ,  $\mathbf{T}_{12}$ , etc. be the components of  $\mathbf{T}$ , then

$$\begin{aligned} d_{\mathbf{T}} = \det(\mathbf{T}) &= \lambda_1 \lambda_2 = \mathbf{T}_{11} \mathbf{T}_{22} - \mathbf{T}_{12}^2 \\ t_{\mathbf{T}} = \operatorname{trace}(\mathbf{T}) &= \lambda_1 + \lambda_2 = \mathbf{T}_{11} + \mathbf{T}_{22}. \end{aligned} \quad (2.51)$$

These invariants can be interpreted as follows: if  $t_{\mathbf{T}} = 0$ , both eigenvalues are zero and the gray values in the neighborhood are constant.  $d_{\mathbf{T}} = 0$  and  $t_{\mathbf{T}} > 0$  are evidence for a single orientation. It can also be shown that  $d_{\mathbf{T}} \leq t_{\mathbf{T}}^2$  always holds, so using  $\epsilon_1$  as a confidence parameter, the confidence measure is

$$d_{\mathbf{T}} \leq t_{\mathbf{T}}^2 \epsilon_1, \quad (2.52)$$

with  $0 < \epsilon_1$ . If the confidence for a single orientation is high enough, the orientation still needs to be estimated which is done using

$$\begin{aligned} \mathbf{v} &= \frac{(-T_{12}, T_{11})^T}{\sqrt{T_{11}^2 + T_{12}^2}} \text{ or} \\ \mathbf{v} &= \frac{(T_{22}, -T_{12})^T}{\sqrt{T_{22}^2 + T_{12}^2}}. \end{aligned} \quad (2.53)$$

$\mathbf{v}$  can be derived this way because the lower eigenvalue is assumed to be zero. So  $\mathbf{T} \mathbf{v} = 0$ , the vector  $\mathbf{v}$  needs to be orthogonal to both rows of the matrix  $\mathbf{T}$ . This matrix has rank one - the rows are therefore linearly dependent. Consequently, an eigenvector  $\mathbf{v}$  needs to be orthogonal to either the first or the second row.

If the confidence for a single orientation is high, the algorithm terminates with the estimated single orientation. Otherwise, the image is tested for a double orientation.

### 2.2.2 Double Orientation Estimation

In case of more than one orientation in a neighborhood  $\Omega$ , the confidence measure 2.52 introduced in section 2.2.1 does not hold. Therefore it is necessary to estimate a

double orientation, if there are two orientations present in a local neighborhood which is introduced in this section. Again, one can define a confidence measure for the double orientation case and calculate this.

Two possibilities for double orientations in a local neighborhood are considered:

- Two overlaid orientations:  $I(\mathbf{x}) = I_1(\mathbf{x}) + I_2(\mathbf{x})$ ,  $\mathbf{x} \in \Omega$  where both components are assumed to be ideally oriented with orientations  $\theta_1$  and  $\theta_2$ .
- Two occluding orientations:  $I(\mathbf{x}) = \begin{cases} I_1(\mathbf{x}), & \forall \mathbf{x} \in \Omega_1 \\ I_2(\mathbf{x}), & \forall \mathbf{x} \in \Omega_2 \end{cases}$  with  $\Omega_1$  and  $\Omega_2$  being two regions of the image which are adjacent to each other or expressed in a more mathematical way:  $\Omega_1, \Omega_2 \subset \Omega$ ,  $\Omega_1 \cap \Omega_2 = \emptyset$ .

### Overlaid Case

First, the case of two overlaid orientations is considered. The operator defined for this case is a composition of the operator for the single orientation case, executed for both angles, denoted by  $\theta$  and  $\phi$ :

$$\begin{aligned}
 & \alpha(\theta)I_1(\mathbf{x}) = 0 \text{ and } \alpha(\phi)I_2(\mathbf{x}) = 0 & \forall \mathbf{x} \in \Omega \\
 \Rightarrow & \alpha(\theta)\alpha(\phi)I(\mathbf{x}) = (\cos(\theta)\frac{\partial}{\partial x} + \sin(\theta)\frac{\partial}{\partial y})(\cos(\phi)\frac{\partial}{\partial x} + \sin(\phi)\frac{\partial}{\partial y})I(\mathbf{x}) \\
 & = (\cos(\theta)\cos(\phi), \sin(\theta + \phi), \sin(\theta)\sin(\phi))^T (I_{xx}(\mathbf{x}), I_{xy}(\mathbf{x}), I_{yy}(\mathbf{x})) \\
 & = \mathbf{a}^T \mathbf{d}I \\
 & = 0.
 \end{aligned} \tag{2.54}$$

$\mathbf{a}$  is called the MOP-vector, which stands for Mixed Orientation Parameters, while  $\mathbf{d}$  is an operator for the second order derivatives,  $\mathbf{d} = \left( \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial x \partial y}, \frac{\partial^2}{\partial y^2} \right)^T$ . From now on let  $\mathbf{u} := (\cos(\theta), \sin(\theta))^T$  and  $\mathbf{v} := (\cos(\phi), \sin(\phi))^T$  be the demanded vectors. Constraint 2.54 can also be expressed by  $\alpha(\theta)\alpha(\phi)I(\mathbf{x}) = \frac{\partial^2 I(\mathbf{x})}{\partial \mathbf{u} \partial \mathbf{v}}$  (see also [1]). A closer look at  $\mathbf{a}$  reveals that the information needed for  $\mathbf{u}$  and  $\mathbf{v}$  is already contained in the MOP-vector. The next step is the computation of  $\mathbf{a}$ .

The arguments for computing the vector  $\mathbf{a}$  are basically the same as in the single orientation case. The lack of two ideal orientations results in 2.54 not equaling zero exactly, therefore the following integral is minimized.

$$\begin{aligned}
 \mathbf{a} = \operatorname{argmin} Q_1(\mathbf{a}) &= \operatorname{argmin} \left( \int_{\Omega} (\mathbf{a}^T \mathbf{d}I)^2 d\Omega \right) \\
 &= \operatorname{argmin} \left( \mathbf{a}^T \int_{\Omega} \mathbf{d}I (\mathbf{d}I)^T d\Omega \mathbf{a} \right) \\
 &= \operatorname{argmin} \left( \mathbf{a}^T \mathbf{T}_1 \mathbf{a} \right),
 \end{aligned} \tag{2.55}$$

where  $\mathbf{T}_1$  is again an integral of a matrix, except this time a  $3 \times 3$  matrix:

$$\begin{aligned}
 \mathbf{T}_1 &= \int_{\Omega} \mathbf{d}I (\mathbf{d}I)^T d\Omega \\
 &= \int_{\Omega} \begin{bmatrix} I_{xx}^2 & I_{xx}I_{xy} & I_{xx}I_{yy} \\ I_{xx}I_{xy} & I_{xy}^2 & I_{xy}I_{yy} \\ I_{xx}I_{yy} & I_{xy}I_{yy} & I_{yy}^2 \end{bmatrix} d\Omega.
 \end{aligned} \tag{2.56}$$



Minimizing  $Q_1$  under the criterion  $\mathbf{a}^T \mathbf{a} = 1$  is, like in the single orientation case, equivalent to finding  $\mathbf{a}$  such that  $\mathbf{T}_1 \mathbf{a} = \lambda_3 \mathbf{a}$ , where  $\lambda_3$  is the smallest eigenvalue of the matrix  $\mathbf{T}_1$  and  $\mathbf{a}$  the corresponding eigenvector. In order to have evidence for the double orientation case  $\lambda_3$  needs to be close to zero. This is sufficient because the single orientation case has already been ruled out. Like in the single orientation case, the estimation is done using a weighting function but is for now omitted in favor of better readability.

### Occluding Case

As already mentioned, instead of being overlaid, two i1D structures can also be occluding - the domain  $\Omega$  is separated into two non-intersecting subsets  $\Omega_1$  and  $\Omega_2$  with the corresponding parts of the image  $I_1$  and  $I_2$  respectively. In order to find the orientations  $\theta$  and  $\varphi$ , two equations need to be solved:

$$\frac{\partial I_1(\mathbf{x})}{\partial \mathbf{u}} = 0, \mathbf{x} \in \Omega_1 \text{ and } \frac{\partial I_2(\mathbf{x})}{\partial \mathbf{v}} = 0, \mathbf{x} \in \Omega_2. \quad (2.57)$$

In this case, the composite criterion is

$$\frac{\partial I(\mathbf{x})}{\partial \mathbf{u}} \frac{\partial I(\mathbf{x})}{\partial \mathbf{v}} = \mathbf{a}^T \mathbf{d}_0 I = 0, \quad (2.58)$$

where  $\mathbf{d}_0 = (I_x^2, I_x I_y, I_y^2)^T$ . This criterion holds everywhere except at the boundary between regions  $\Omega_1$  and  $\Omega_2$ . The arguments for finding the vector  $\mathbf{a}$  are exactly the same as in the overlaid case of two orientations above. They yield

$$\mathbf{a} = \operatorname{argmin} Q_2(\mathbf{a}) = \operatorname{argmin} \left( \mathbf{a}^T \int_{\Omega} \mathbf{d}_0 I (\mathbf{d}_0 I)^T d\Omega \mathbf{a} \right) = \operatorname{argmin}(\mathbf{a}^T \mathbf{T}_2 \mathbf{a}), \quad (2.59)$$

where  $\mathbf{T}_2$  is an integral of a matrix:

$$\mathbf{T}_2 = \int_{\Omega} \mathbf{d} I (\mathbf{d} I)^T d\Omega = \int_{\Omega} \begin{bmatrix} I_x^4 & I_x^3 I_y & I_x^2 I_y^2 \\ I_x^3 I_y & I_x^2 I_y^2 & I_x I_y^3 \\ I_x^2 I_y^2 & I_x I_y^3 & I_y^4 \end{bmatrix} d\Omega, \text{ with } \mathbf{a}^T \mathbf{a} = 1. \quad (2.60)$$

Minimizing  $Q_2$  is again equivalent to finding the eigenvector corresponding to the lowest eigenvalue. So in both, the overlaid and the occluding case, the same frame work is used, only the Matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$  differ. Hence both matrices are from now on referred to as  $\mathbf{T}$ .

### MOP-Vector Decomposition

In order to estimate the orientations in the overlaid or occluding case,  $\mathbf{T} \mathbf{a} = \lambda_3 \mathbf{a}$  needs to be solved for  $\mathbf{a}$ . After ruling out the single orientation case,  $\mathbf{T}$  has at most one eigenvalue that is close to zero. According to [1], it is possible to determine  $\mathbf{a}$  up to scale and sign, which is sufficient in this case.

The vector  $\mathbf{a}$  is not computed directly but by using the minors of  $\mathbf{T}$ . When estimating those, a confidence measure for the i2D case can also be derived. The minor  $M_{ij}$ ,  $i, j, \in \{1, 2, 3\}$  is obtained by calculating the determinant of the matrix  $\mathbf{T}$  where the  $4 - i$  row and the  $4 - j$  column are deleted. When  $\mathbf{T}_{ij}$  denotes the number of the  $i$ -th row and the  $j$ -th column of  $\mathbf{T}$ , the following invariants are defined:

$$\begin{aligned} d_{\mathbf{T}_2} &= \det(\mathbf{T}) = \lambda_1 \lambda_2 \lambda_3 = \mathbf{T}_{11} M_{11} - \mathbf{T}_{12} M_{12} + \mathbf{T}_{13} M_{13} \\ s_{\mathbf{T}_2} &= \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3 = M_{11} + M_{22} + M_{33}. \end{aligned} \quad (2.61)$$

The confidence for the i2D case is then measured using

$$d_{\mathbf{T}_2}^2 \leq s_{\mathbf{T}_2}^3 \epsilon_2, \quad \text{with } 0 < \epsilon_2. \quad (2.62)$$

The MOP-vector  $\mathbf{a}$  itself can be calculated in three different ways using the minors. Proof of this can be found in [16]:

$$\mathbf{a}_i = R_i(M_{i3}, -M_{i2}, M_{i1})^T = (a_i, b_i, c_i)^T, \quad R_i^2 = (a_i + c_i)^2 + b_i^2, \quad i \in \{1, 2, 3\}. \quad (2.63)$$

So far, the vector  $\mathbf{a} = (\cos(\phi) \cos(\phi), \sin(\theta + \phi), \sin(\theta) \sin(\phi))^T = (a, b, c)^T$  contains the orientations  $\theta$  and  $\varphi$  only implicitly. The decomposition of the MOP-vector is done by defining  $z_1$  and  $z_2$  by  $z_1 z_2 = ac$  and  $z_1 + z_2 = b$ .  $z_1$  and  $z_2$  can be computed by estimating the roots of the polynomial  $f(z) = (z - z_1)(z - z_2) = z^2 - bz + ac$ . These components are employed in the definition of a matrix  $M$  which translates back to the cosines and sines of the two sought-after orientations:

$$M := \begin{bmatrix} a & z_1 \\ z_2 & c \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cos(\phi) & \cos(\theta) \sin(\phi) \\ \sin(\theta) \cos(\phi) & \sin(\theta) \sin(\phi) \end{bmatrix}. \quad (2.64)$$

Each component of this matrix is known either from the MOP-vector or the roots of the polynomial. At the same time the rows and columns encode the two orientations. Considering both rows yields the vector  $\mathbf{v}$  scaled by a factor. By reducing the vector to unit length,  $\mathbf{v}$  is obtained:

$$\begin{aligned} &\cos(\theta) \mathbf{v}^T \quad \text{and} \quad \sin(\theta) \mathbf{v}^T \\ \Rightarrow & \\ \mathbf{v} &= \frac{(a, z_1)^T}{\sqrt{a^2 + z_1^2}} = \frac{(z_2, c)^T}{\sqrt{z_2^2 + c^2}}. \end{aligned} \quad (2.65)$$

The same is true for the columns which yield the vector  $\mathbf{u}$  encoding  $\theta$ :

$$\begin{aligned} &\cos(\phi) \mathbf{u} \quad \text{and} \quad \sin(\phi) \mathbf{u} \\ \Rightarrow & \\ \mathbf{u} &= \frac{(a, z_2)^T}{\sqrt{a^2 + z_2^2}} = \frac{(z_1, c)^T}{\sqrt{z_1^2 + c^2}}. \end{aligned} \quad (2.66)$$

Calculating the arctangent of the components of both vectors yields the sought orientations. According to [2] a more robust way of computing the orientations is comprised of combining both possibilities for computing  $\mathbf{u}$  and  $\mathbf{v}$ .

The result is

$$\mathbf{u} = \begin{cases} \frac{\sqrt{a^2+z_2^2}(a,z_2)^T + \sqrt{z_1^2+c^2}(z_1,c)^T}{\sqrt{a^2+z_2^2} + \sqrt{z_1^2+c^2}}, & \text{if } (a, z_2)^T(z_1, c) > 0 \\ \frac{\sqrt{a^2+z_2^2}(a,z_2)^T - \sqrt{z_1^2+c^2}(z_1,c)^T}{\sqrt{a^2+z_2^2} + \sqrt{z_1^2+c^2}}, & \text{otherwise} \end{cases}$$

$$\mathbf{v} = \begin{cases} \frac{\sqrt{a^2+z_1^2}(a,z_1)^T + \sqrt{z_2^2+c^2}(z_2,c)^T}{\sqrt{a^2+z_1^2} + \sqrt{z_2^2+c^2}} & \text{if } (a, z_1)^T(z_2, c) > 0 \\ \frac{\sqrt{a^2+z_1^2}(a,z_1)^T + \sqrt{z_2^2+c^2}(z_2,c)^T}{\sqrt{a^2+z_1^2} + \sqrt{z_2^2+c^2}} & \text{otherwise.} \end{cases}$$

This concludes the derivation of the Lübeck method. This chapter described the theory of both methods used for local main orientation and apex angle computation in this thesis. The Riesz transforms are derived in two ways, first as a multi-dimensional extension of the analytic signal, or more precisely, the Hilbert transform and second in terms of the Radon transform, an approach that led to new interpretations for i1D and i2D signals. The Lübeck method uses gradients instead of Riesz transforms in order to reach a similar goal. Instead of local main orientation and apex angle, the orientations of up to two signals are calculated. The next chapter is concerned with the implementation of both methods and the design of a suitable bandpass filter, especially for the second and third order Riesz transforms for the Kiel method.



## 3 Filter Design and Implementation

Both, the method using the higher order Riesz transforms and the method developed in Lübeck, are implemented in C++ and partly using Matlab. In addition to the implementation of the algorithms, a graphical user interface has been constructed. This GUI is implemented using Trolltech's Qt, an open source library for GUI development (see also <http://trolltech.com/developer>). Using the GUI, test images can be loaded and a variety of filters can be applied to them. Test images, filters and different implementations of the Riesz transform can be combined freely, allowing a wide range of experiments (for a more detailed description see C). The implementation was running on CPUs with about 3 GHz.

In this chapter the implementation of the Riesz transforms and later, of the Lübeck method will be explained. Both, implementation and search for filter kernels are closely related and are therefore both described in this chapter.

### 3.1 Implementation of Riesz Transforms

When implementing the monogenic signal or even higher order Riesz transforms, the complicated part is the usage of a suitable bandpass filter and finding the corresponding filter kernels. For single order Riesz transforms convolution kernels have been proposed by Felsberg and Sommer (see [6]). The derivation of the kernels for higher order Riesz transforms is complicated. To the best of my knowledge, no derivation of spatial kernels of higher order exists. However, there are several possibilities to circumvent this problem. First successive convolutions with the existing kernels can be done in order to obtain second and third order kernels. A second approach consists of using Matlab and its implementation of the discrete Radon transform and its inverse to implement the Radon based theory of the Riesz transforms directly. A third option is to compute the kernels in the frequency domain and apply the inverse Fourier transform. A fourth one involves the use of an approximation of the difference of Poisson (DOP), the bandpass filter that is usually used. All four possibilities are tried in this thesis and worked to a certain degree. In addition to that, all approaches contributed to the development of each other in some way and that is why all of them are important. The remainder of this chapter describes all four approaches.

#### 3.1.1 Kernels Obtained by Repeated Convolution in the Spatial Domain

In this section a DOP (Difference of Poisson) is used as a bandpass filter. When using a DOP, it is so far not possible to derive filter kernels for second and third order Riesz

transforms. However, as Felsberg is proposing in [6], it is possible to derive kernels for the single order Riesz transform (the classic monogenic signal). Using these, successive convolutions in the spatial domain can be applied to obtain higher order kernels. This is shown in the following section.

#### Bandpass Filter Implementation of the Monogenic Signal

Section 2.1.1 described the theory of the monogenic signal. Now, an implementation needs to be developed. The use of the monogenic signal assumes that the signal (or in this case the image to be processed) consists of few frequencies - that it is bandlimited. Computing the monogenic signal of real images results in a problem: real images usually consist of a wide range of frequencies. Therefore a set of bandpass filters needs to be combined with the monogenic signal. Those bandpass filters have to be two-dimensional, radial filters. The bandpass filter is build using a Poisson filter instead of a Gaussian filter. Like the Gaussian filter, the Poisson filter is a low pass filter for which the inverse Fourier transformation is known. Subtracting two Gaussian filters with different parameters from each other yields a bandpass filter. The same is true for the Poisson filter. The advantage of the Poisson filter is that the kernel of Poisson filter is closely related to the kernel of the monogenic signal.

Because the signal needs to be filtered, the even part that is so far simply the original signal, is now also convolved with a mask. This results in one even and two odd filters. They can be designed in the frequency domain more easily than in the spatial domain. Let  $\mathbf{u} = (u_1, u_2)^T$  be a 2D frequency vector and  $s \in (0, \infty)$  a scaling factor, the new even filter (simply a Poisson filter) in the frequency domain is

$$\mathcal{F}\{P\}(\mathbf{u}) := \exp(-2\pi|\mathbf{u}|s). \quad (3.1)$$

The already known filter kernels of the Riesz transform in the frequency domain  $\mathcal{F}\{R_x\}$  and  $\mathcal{F}\{R_y\}$  are now also multiplied by  $\exp(-2\pi|\mathbf{u}|s)$ , becoming the two odd filter parts  $\mathcal{F}\{R_x * P\}$  and  $\mathcal{F}\{R_y * P\}$ :

$$\begin{aligned} \mathcal{F}\{R_x * P\}(\mathbf{u}) &:= \mathcal{F}\{R_x\}(\mathbf{u}) \exp(-2\pi|\mathbf{u}|s) \\ \mathcal{F}\{R_y * P\}(\mathbf{u}) &:= \mathcal{F}\{R_y\}(\mathbf{u}) \exp(-2\pi|\mathbf{u}|s). \end{aligned} \quad (3.2)$$

These kernels can be transformed into the spatial domain. Proof has been provided by [6] and may also be found in the appendix of this thesis. Transformed into the spatial domain, the new kernels are

$$\begin{aligned} P(\mathbf{x}) &:= \frac{s}{2\pi(s^2+|\mathbf{x}|^2)^{\frac{3}{2}}} \\ (R_x * P)(\mathbf{x}) &:= \frac{-x_1}{2\pi(s^2+|\mathbf{x}|^2)^{\frac{3}{2}}} \\ (R_y * P)(\mathbf{x}) &:= \frac{-x_2}{2\pi(s^2+|\mathbf{x}|^2)^{\frac{3}{2}}}. \end{aligned} \quad (3.3)$$

In order to get a bandpass filter, the two Poisson kernels with different scales  $s$  are subtracted from each other.

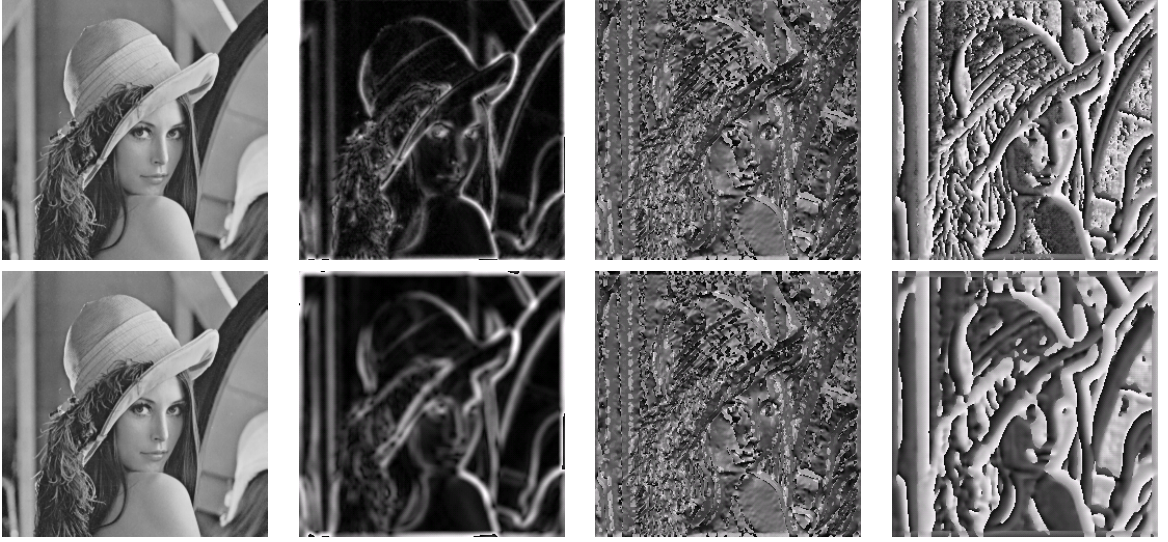


Figure 3.1: Monogenic signal of the famous Lena calculated at different scales: From left to right: Columns: Original image, amplitude, local orientation and local phase. Upper row: Monogenic signal with  $coarse = 2$  and  $fine = 1$  as parameters. Lower row: Monogenic signal with  $coarse = 4$  and  $fine = 5$  as parameters.

Using a coarse  $c \in ]0, \infty[$  and a fine  $f \in ]0, \infty[$  scale with  $c > f$ , the filter kernels resulting from the subtraction are

$$\begin{aligned}
 DOP(\mathbf{x}) &:= \frac{c}{2\pi(c^2+|\mathbf{x}|^2)^{\frac{3}{2}}} - \frac{f}{2\pi(f^2+|\mathbf{x}|^2)^{\frac{3}{2}}} \\
 (R_x * DOP)(\mathbf{x}) &:= \frac{-x_1}{2\pi(c^2+|\mathbf{x}|^2)^{\frac{3}{2}} - 2\pi(f^2+|\mathbf{x}|^2)^{\frac{3}{2}}} \\
 (R_y * DOP)(\mathbf{x}) &:= \frac{-x_2}{2\pi(c^2+|\mathbf{x}|^2)^{\frac{3}{2}} - 2\pi(f^2+|\mathbf{x}|^2)^{\frac{3}{2}}}.
 \end{aligned} \tag{3.4}$$

These three convolution kernels are used to compute the monogenic signal:

$$\begin{aligned}
 p &:= (DOP * g)(x_1, x_2) \\
 q_1 &:= ((R_x * DOP) * g)(x_1, x_2) \\
 q_2 &:= ((R_y * DOP) * g)(x_1, x_2).
 \end{aligned} \tag{3.5}$$

Phase vector, local phase, local orientation, and amplitude are computed as defined in equations (2.12) - (2.15) in section 2.1.1.

The monogenic signal is implemented in C++ as described above. In this thesis the orientation and the apex angle are of interest. However, the higher order Riesz transforms will be needed for robust apex angle computation. There are several possibilities for obtaining the higher order kernels, which will be described below. To conclude this section, some examples of the results of the monogenic signal computation are presented. Figure 3.1 shows the results of the monogenic signal computation for Lena, one of the standard test images in Computer Vision. The monogenic signal is computed at different scales. Displayed are the original image, the amplitudes, local orientation, and local phase.

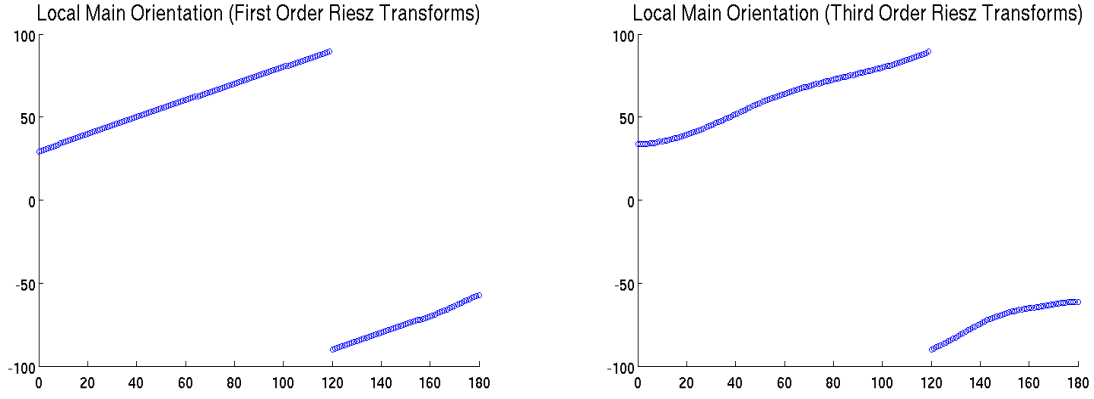


Figure 3.2: Results from a test run using the Radon transform implemented in Matlab. The first orientation  $\theta_1$  is kept constant at  $60^\circ$ , while the other one has taken all values from  $0^\circ - 180^\circ$ . The left image shows the results of the main orientation computation using the first order Riesz transform, while the main orientation in the right image is computed using the third order Riesz transform.

#### Riesz Transforms of Second and Third Order in the Spatial Domain

After calculating these kernels in the spatial domain, they can be convolved with each other in order to obtain kernels for Riesz transforms of second and third order. However, this approach does not work very accurately if the kernel sizes are kept constant. Because the kernel sizes grow with a factor of  $2 \times 2$  with each convolution, applying them to images becomes increasingly inefficient.

#### 3.1.2 Using the Discrete Radon Transform and its Inverse

For this thesis, a more direct implementation of the theory using the Radon transform and its inverse to interpret the Riesz transform is tested. This implementation is done using the Matlab Image Processing Toolbox where an implementation of the discrete Radon transform and its inverse are provided. The results of this approach are less accurate than the outcome after applying convolution kernels. Reasons are probably the finite nature of images and numerical problems. The images of the Radon transform (figure 2.3) in section 2.1.2 for example show artefacts in the columns for all orientation angles not present in the image. The implementation of the inverse Radon transform is even more complex. Another disadvantage is the high computational complexity of the implementation of the Radon transform.

However, it is possible to verify the theory because apart from some inaccuracies, the calculation of local main orientation and apex angle is successful. This can also be observed in figure 3.2. The local main orientation computation in both images is fairly accurate, especially in the left image, where  $\theta_{main} = \text{atan}\left(\frac{R_y}{R_x}\right)$ , thus only first order Riesz transforms are applied. The application of third order Riesz transform in the right image -  $\theta_{main} = \text{atan}\left(\frac{\text{trace}(T_{oddy})}{\text{trace}(T_{odd_x})}\right)$  - shows more inaccuracies. The apex angle



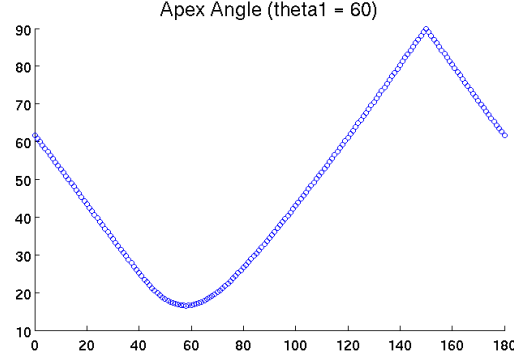


Figure 3.3: Result from a test run using the Radon transform implemented in Matlab. The first orientation  $\theta_1$  is kept constant at  $60^\circ$ , while the other one has taken all values from  $0^\circ - 180^\circ$ . Shown are the results of the apex angle computation.

computation poses the biggest problem (see figure 3.3), especially when very small apex angles need to be computed. The result in figure 3.3 shows the apex angle computed by  $|\theta_1 - \theta_2| = 2 * \text{atan} \left( \sqrt{\left| \frac{\text{trace}(T_{\text{even}}) - \sqrt{\text{trace}(T_{\text{even}})^2 - \det(T_{\text{even}})}}{\text{trace}(T_{\text{even}}) + \sqrt{\text{trace}(T_{\text{even}})^2 - \det(T_{\text{even}})}} \right|} \right)$ . The computation using the Radon transform estimated an apex angle between  $15^\circ$  and  $20^\circ$  at the non-linear part around  $60^\circ$ , where the true apex angle is zero, so the apex angle computation is not accurate for small apex angles.

A conclusion of these tests is that a direct implementation of the theory is possible, but inaccurate and time consuming because of the longer computation time of the Radon transform and Matlab itself. Therefore it is still important to find some suitable convolution kernels.

### 3.1.3 Kernels Developed in the Frequency Domain

It is difficult to derive useful kernels for the second and third order Riesz transforms in the spatial domain. Fortunately, this problem can also be solved in the frequency domain and the inverse Fourier transform can be applied to obtain the necessary kernels in the spatial domain. This procedure is implemented for this thesis and yields good results. A quick overview is given in this section. The theory is straight forward and fairly easy to program if a standard C++ Fourier transform is used. These kernels are programmed and tested in this thesis, so that they can be compared to the so called LOP kernels, which will be presented in the next section. In order to develop the kernels in the frequency domain, the convolution theorem is used. Let  $f : \mathbb{R}^2 \rightarrow \mathbb{R} \circ \bullet F : \mathbb{R}^2 \rightarrow \mathbb{R}$  and  $g : \mathbb{R}^2 \rightarrow \mathbb{R} \circ \bullet G : \mathbb{R}^2 \rightarrow \mathbb{R}$  be two functions to be convolved, then

$$f(x, y) * g(x, y) \circ \bullet F(u, v)G(u, v), \quad (x, y), (u, v) \in \mathbb{R}^2. \quad (3.6)$$

In order to develop the second and third order Riesz transform kernels in the frequency

domain, the first order kernels are considered first:

$$\begin{aligned}\mathcal{F}\{R_x\}(\mathbf{u}) &= \frac{u}{\mathbf{u}}i \\ \mathcal{F}\{R_y\}(\mathbf{u}) &= \frac{v}{\mathbf{u}}i.\end{aligned}\tag{3.7}$$

According to the convolution theorem, the second and third order Riesz transforms can now be obtained by multiplication of the first order kernels in the frequency domain. The filter kernel for the even part remains the same, the other kernels are the inverse Fourier transforms of

$$\begin{aligned}\mathcal{F}\{R_x\{R_x\}\}(\mathbf{u}) &= i\frac{u}{|\mathbf{u}|} i\frac{u}{|\mathbf{u}|} = -\frac{u^2}{|\mathbf{u}|^2} \\ \mathcal{F}\{R_x\{R_y\}\}(\mathbf{u}) &= i\frac{u}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} = -\frac{uv}{|\mathbf{u}|^2} \\ \mathcal{F}\{R_y\{R_y\}\}(\mathbf{u}) &= i\frac{v}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} = -\frac{v^2}{|\mathbf{u}|^2} \\ \mathcal{F}\{R_x\{R_x\{R_x\}\}\}(\mathbf{u}) &= i\frac{u}{|\mathbf{u}|} i\frac{u}{|\mathbf{u}|} i\frac{u}{|\mathbf{u}|} = -i\frac{u^3}{|\mathbf{u}|^3} \\ \mathcal{F}\{R_x\{R_x\{R_y\}\}\}(\mathbf{u}) &= i\frac{u}{|\mathbf{u}|} i\frac{u}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} = -i\frac{u^2v}{|\mathbf{u}|^3} \\ \mathcal{F}\{R_x\{R_y\{R_y\}\}\}(\mathbf{u}) &= i\frac{u}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} = -i\frac{uv^2}{|\mathbf{u}|^3} \\ \mathcal{F}\{R_y\{R_y\{R_y\}\}\}(\mathbf{u}) &= i\frac{v}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} i\frac{v}{|\mathbf{u}|} = -i\frac{v^3}{|\mathbf{u}|^3}.\end{aligned}$$

The needed bandpass filter - a difference of Poisson filter - is applied directly in the frequency domain.

The inverse Fourier transforms of these filter kernels are calculated in C++ by using the fftw library. This open source library is available on the Internet at <http://www.fftw.org> and computes the discrete Fourier transforms of one- or more-dimensional signals.

This approach is simple in theory but yields very good results, especially when the main orientation is calculated. However, it would be more elegant and probably faster to derive spatial kernels that can be computed directly in the necessary kernel size. The next section describes such an approach.

#### 3.1.4 Kernels Using the LOP as a Bandpass Filter Approximation

The calculation of the Riesz transform can only be done using a bandpass filter because of the already stated assumption that the signal is band limited. In his dissertation [8], Felsberg proposed the use of the difference of Poisson filter (DOP) instead of the difference of Gaussian filter (DOG), the standard band pass filter. Unfortunately, it is, to the best of my knowledge, not possible to derive spatial domain filter kernels for the second and third order Riesz transform in combination with DOP kernels. In this section an approach is described that allows the derivation of kernels for the second and third order Riesz transform. However, the kernels for the first order transform cannot be computed by this method.

It has been known for a while that the DOG (Difference of Gaussian) can be approximated with the LOG (Laplacian of Gaussian, also known as: Mexican Hat operator or Marr-Hildreth operator). This fact has been widely used in the opposite direction for

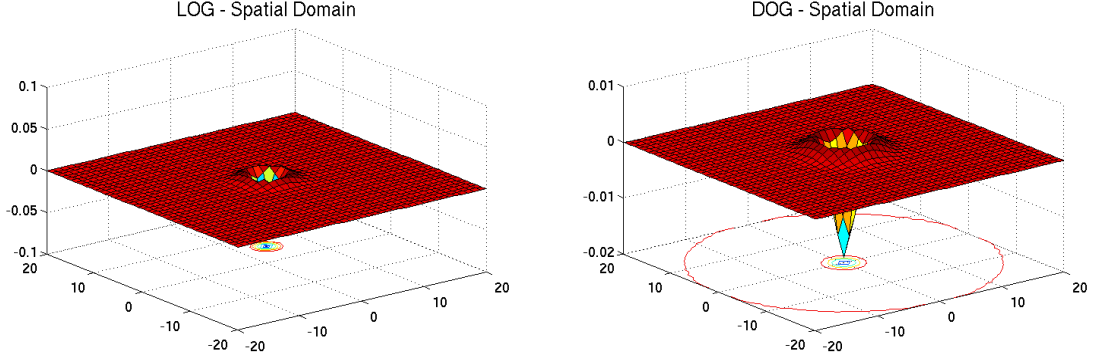


Figure 3.4: Left: LOG kernel. Right: DOG kernel

edge detection purposes: the LOG-filter is needed and the DOG-filter is used instead. The reason for this is the smaller kernel size that can be used with the DOG. The standard Gaussian filter is a low pass filter with parameter  $\sigma \in \mathbb{R} > 0$ . Using two different Gaussian filters with different parameters  $\sigma_1, \sigma_2 \in \mathbb{R} > 0$  and taking the difference of both yields a bandpass filter, the so called differences of Gaussian (DOP) filter. The LOG-filter is a combination of a Gaussian filter  $g(x, y)$ ,  $(x, y) \in \mathbb{R}^2$  and the Laplace operator denoted by  $\nabla$ . Both are applied to a signal  $f(x, y)$ ,  $(x, y) \in \mathbb{R}^2$ :

$$\begin{aligned}
 g(x, y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \\
 \nabla &= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \\
 (LOG * f)(x, y) &= \nabla(g * f)(x, y).
 \end{aligned} \tag{3.8}$$

For simplification, the  $\nabla$ -operator can also be combined directly with the Gaussian filter resulting in

$$\begin{aligned}
 LOG(x, y) &= \frac{\partial^2}{\partial x^2} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) + \frac{\partial^2}{\partial y^2} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) = \\
 &= -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).
 \end{aligned} \tag{3.9}$$

Figure 3.4 shows the similarity of both kernels.

In [15] and [7] Köthe and Felsberg examine the theoretical relation between Riesz transforms and gradients. For better comparison, they use the Laplacian of Gaussian (LOG) as a bandpass filter for the signals to be analyzed.

It seems that the same can be done using a Laplacian of Poisson (LOP) instead of the Difference of Poisson (DOP). This approach is chosen in this section and it turns out that it allows computation of filter kernels in the spatial domain for second and third order Riesz transforms, but unfortunately not for first order Riesz transforms. In the

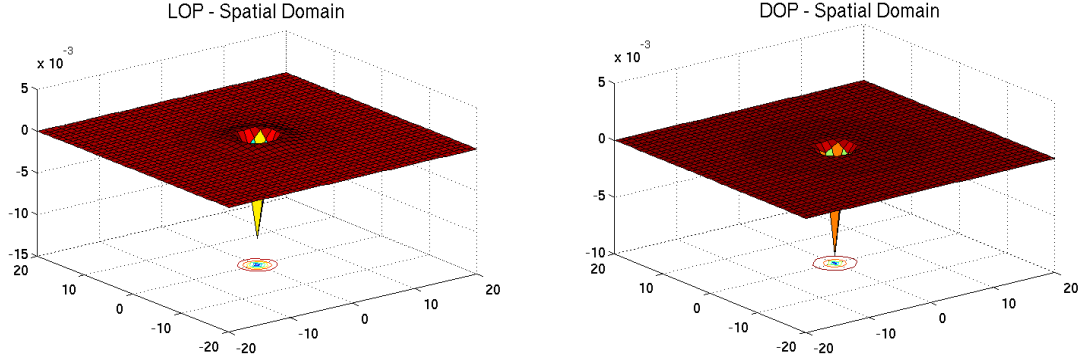


Figure 3.5: Left: LOP kernel. Right: DOP kernel.

following, the LOP is introduced and the kernels for the second and third order Riesz transforms are derived.

The standard Poisson filter for two dimensions is defined as

$$\mathcal{F}\{P\}(\mathbf{u}) = \exp(-2\pi|\mathbf{u}|s) \bullet \circ \frac{s}{2\pi|\mathbf{x} + s|^3}. \quad (3.10)$$

Analog to the Laplacian of Gaussian, the Laplacian of Poisson is derived by (see [5] for 2D Fourier transform theorems)

$$\begin{aligned} \mathcal{F}\{LOP\}(\mathbf{u}) &= -4\pi^2|\mathbf{u}|^2 \exp(-2\pi|\mathbf{u}|s) \bullet \circ \\ &\quad \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \frac{s}{2\pi|\mathbf{x} + s|^3} \\ &= -\frac{3s}{2\pi} \left[ 2(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5(x^2 + y^2 + s^2)^{-\frac{7}{2}}(x^2 + y^2) \right]. \end{aligned} \quad (3.11)$$

Figure 3.5 demonstrates the similarity between both kernels.

The LOP kernel is used as a bandpass filter instead of the DOP kernel. So far it has not been possible to derive the single order Riesz transforms, while using this filter, but the second and third order filter kernels can be derived directly. Again, it is done in the frequency domain, where the Riesz transform is multiplied with itself and with the new filter according to the convolution theorem. For reasons of readability not all the kernels will be described here. However, they may be found in the appendix B.2.

The second order Riesz transform has three components. For  $R_x\{R_x\}$  the derivation will be sketched here. A more detailed derivation may be found in the appendix. Again the convolution theorem is used and a multiplication of the Riesz transform and the LOP kernel is done in the frequency domain:

$$\begin{aligned} \mathcal{F}\{R_x\{R_x\} * LOP\}(\mathbf{u}) &= \left( -\frac{u^2}{|\mathbf{u}|^2} \right) (-4\pi^2|\mathbf{u}|^2) \exp(-2\pi|\mathbf{u}|s) \\ &= u^2 4\pi^2 \exp(-2\pi|\mathbf{u}|s). \end{aligned} \quad (3.12)$$

For this simplified function in the frequency domain, it is possible to derive filter kernels in the spatial domain. The idea is to employ the Poisson kernel in spatial and frequency

domain and apply the derivative theorem for two-dimensional Fourier transforms as for example described in [5]:

$$\begin{aligned}
 & \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} f(x, y) \right\} (\mathbf{u}) = -4\pi^2 u^2 F(u, v) \\
 \Rightarrow & \\
 & \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} \frac{s}{2\pi |\mathbf{x} + s|^3} \right\} (\mathbf{u}) = -4\pi^2 u^2 (\exp(-2\pi |\mathbf{u}|s)) \\
 \Rightarrow & \\
 & \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} \frac{-s}{2\pi |\mathbf{x} + s|^3} \right\} (\mathbf{u}) = 4\pi^2 u^2 (\exp(-2\pi |\mathbf{u}|s)) \\
 \Rightarrow & \\
 & \mathcal{F} \left\{ \frac{s}{2\pi} \left( 3(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 15x^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \right\} (\mathbf{u}) \\
 & = u^2 4\pi^2 \exp(-2\pi |\mathbf{u}|s). \tag{3.13}
 \end{aligned}$$

The derivation of the third order kernels is done in a analogous way. This time, four different filter kernels are derived which can be combined into the two parts of the odd curvature tensor. Like for the second order kernels, the derivation is done in the frequency domain using the convolution theorem. In favor of better readability, only the derivation of  $R_x\{R_x\{R_x\}\}$  will be sketched here. Again a more detailed version of the derivation with all different combinations of Riesz transforms may be found in the appendix. In the frequency domain the third order Riesz transform in x-direction with a LOP kernel reads

$$\begin{aligned}
 \mathcal{F}\{R_x\{R_x\{R_x\}\} * LOP\}(\mathbf{u}) &= \left( -i \frac{u^3}{|\mathbf{u}|^3} \right) (-4\pi^2 |\mathbf{u}|^2) \exp(-2\pi |\mathbf{u}|s) \\
 &= \frac{u^3}{|\mathbf{u}|} i 4\pi^2 \exp(-2\pi |\mathbf{u}|s). \tag{3.14}
 \end{aligned}$$

The idea for deriving the corresponding spatial domain kernel is similar to the concept for the second order kernels, except that a kernel of the combination of a single order Riesz transform and a single standard Poisson filter will be used. Afterwards the derivative theorem for 2D convolutions will be applied.

Felsberg [8] already showed how to derive the necessary kernel. Because it is used at this point again, the proof of the derivation may be found in the appendix B.1.

The third order kernel in the x-direction in the spatial domain reads

$$\begin{aligned}
& \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} f(x, y) \right\} (\mathbf{u}) = -4\pi^2 u^2 F(u, v) \\
& \Rightarrow \\
& \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} \frac{-x}{2\pi|\mathbf{x} + s|^3} \right\} (\mathbf{u}) = -4\pi^2 u^2 \left( \frac{u}{|\mathbf{u}|} i \exp(-2\pi|\mathbf{u}|s) \right) \\
& \Rightarrow \\
& \mathcal{F} \left\{ \frac{\partial^2}{\partial x^2} \frac{x}{2\pi|\mathbf{x} + s|^3} \right\} (\mathbf{u}) = 4\pi^2 u^2 \left( \frac{u}{|\mathbf{u}|} i \exp(-2\pi|\mathbf{u}|s) \right) \\
& \Rightarrow \\
& \mathcal{F} \left\{ -\frac{1}{2\pi} \left( 9x(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 15x^3(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \right\} (\mathbf{u}) \\
& = \frac{u^3}{|\mathbf{u}|} i 4\pi^2 \exp(-2\pi|\mathbf{u}|s). \tag{3.15}
\end{aligned}$$

The derivation of the other kernels is done in an analogous way. As already described in 2.1.2, the determinant of the even part of the monogenic curvature tensor serves as a measure for intrinsic dimension. If the determinant is close to or below zero, the structure in the image is considered intrinsically one-dimensional. If the determinant is greater, the structure is considered to consist of two overlaid intrinsically one-dimensional structures, hence the model for i2D structures is used. When implementing these kernels and applying them to images, there are always some numerical errors. Therefore, the determinant of the even tensor approximates zero in case of an intrinsically one-dimensional signal but does not equal zero. Experiments showed that instead of zero a threshold around 0.0000035 yields better results.

Unfortunately, there is so far no formal proof that the DOP can be approximated by the LOP. However, the performed experiments showed that the kernels work well; they are more accurate and faster than the kernels gained by the other implementations described above. Figures 3.6 and 3.7 show that the LOP kernels approximate those kernels that are developed in the frequency domain and afterwards transformed to the spatial domain.

At this point the convolution kernels have been computed in either of the ways described above. The images have been convolved with the kernels and the results have been used directly for the calculation of local main orientation and apex angle. Using these, both orientations can also be computed but unfortunately not unambiguously. This is due to the calculated apex angle which is always the smaller angle between two structures. A more detailed test can be found in chapter 4. Now the implementation of the Lübeck method will be described.

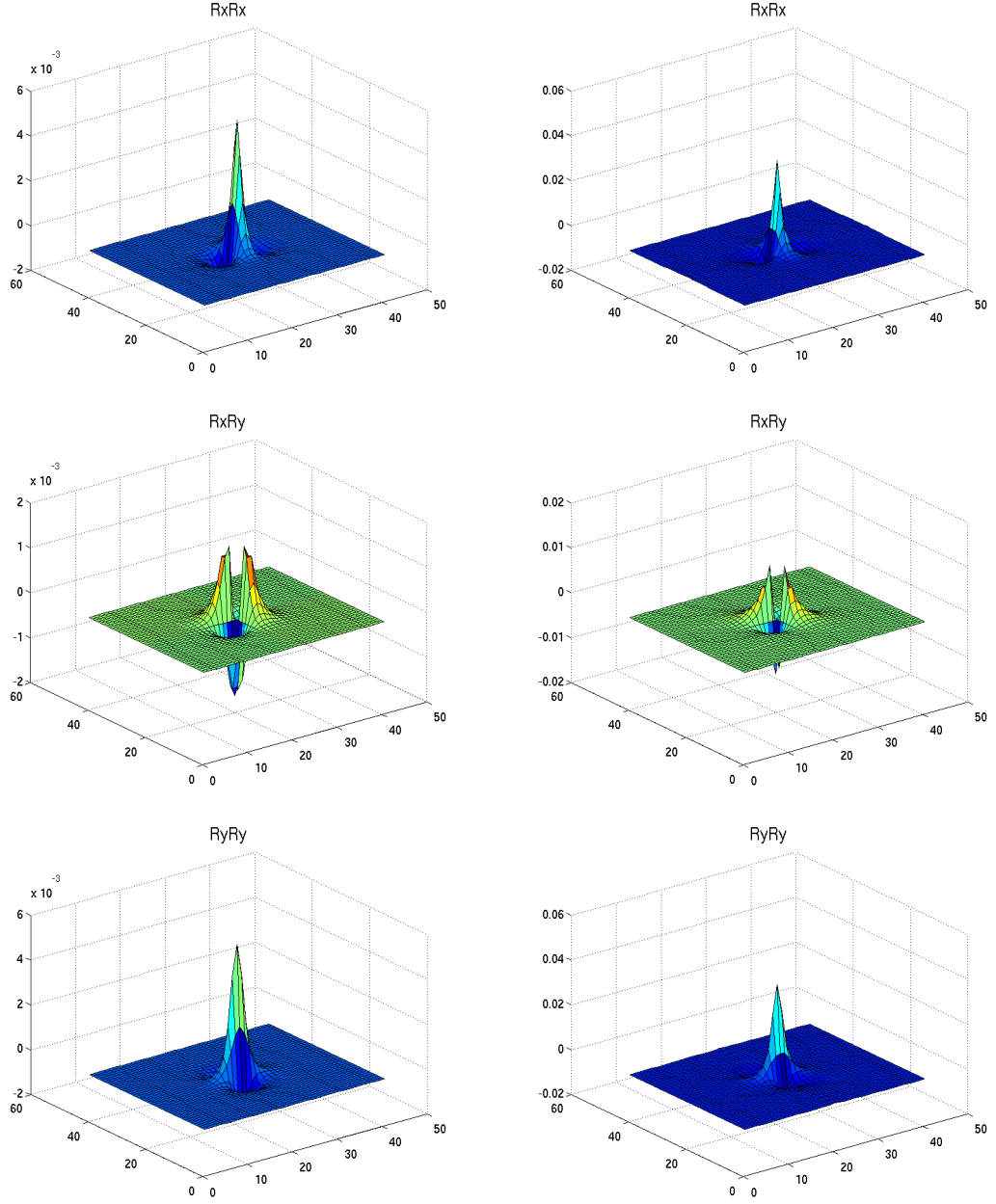


Figure 3.6: Left: LOP kernel. Right: Fourier kernel. From top to bottom:  $(R_x\{R_x\} * LOP)$ ,  $(R_x\{R_y\} * LOP)$ ,  $(R_y\{R_y\} * LOP)$

### 3 Filter Design and Implementation

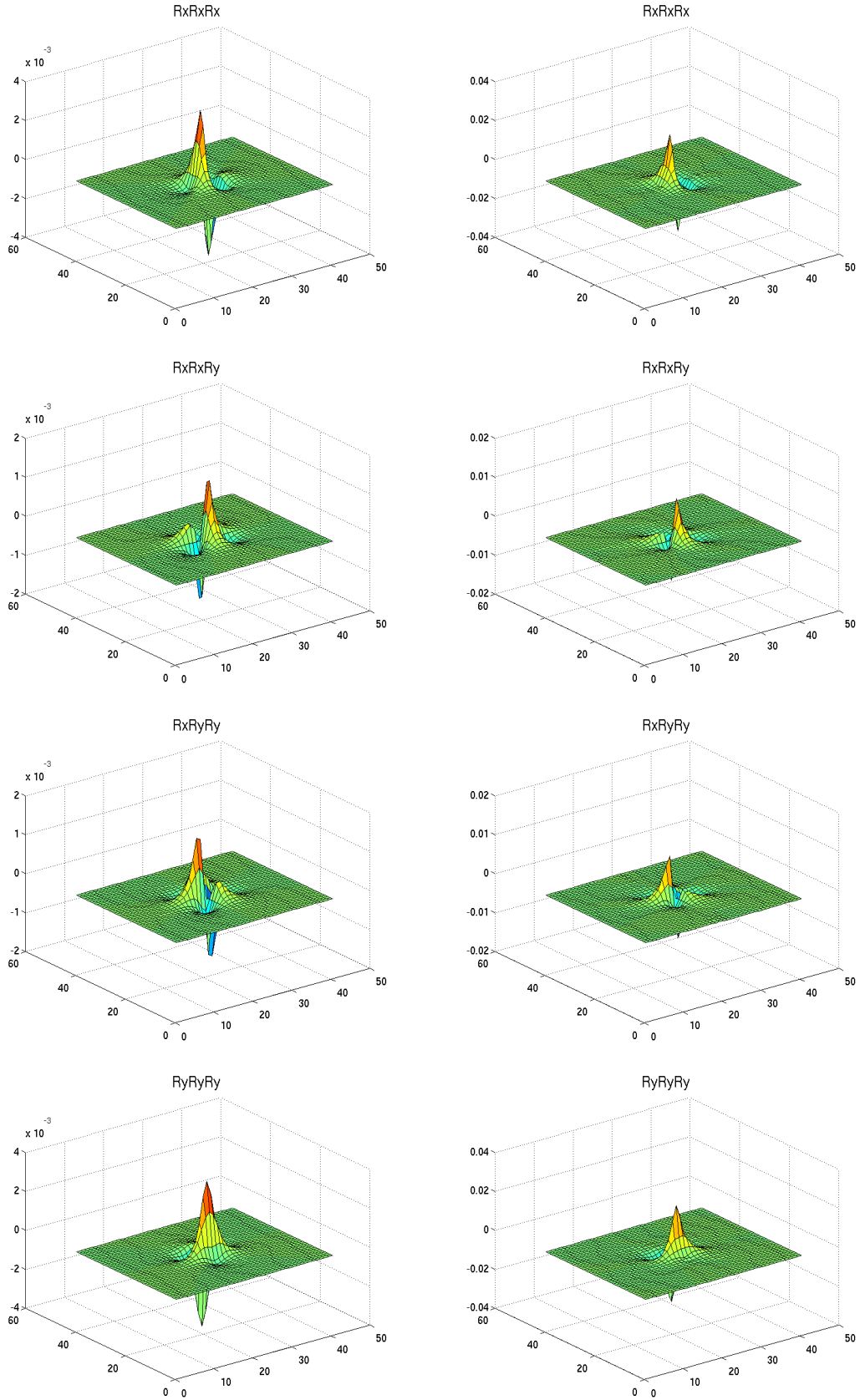


Figure 3.7: Left: LOP kernel. Right: Fourier kernel. From top to bottom:  $(R_x\{R_x\{R_x\}\} * LOP)$ ,  $(R_x\{R_x\{R_y\}\} * LOP)$ ,  $(R_x\{R_y\{R_y\}\} * LOP)$ ,  $(R_y\{R_y\{R_y\}\} * LOP)$



## 3.2 Implementation of the Lübeck Method

The Lübeck method is also implemented in C++. The implementation is more straightforward compared to the Kiel method. It has been mentioned in 2.2 that the entries of the matrices  $\mathbf{T}$ ,  $\mathbf{T}_1$ , and  $\mathbf{T}_2$  which are computed within a neighborhood, can be weighted with a weighting function. In this implementation a Gaussian filter (usually with  $\sigma = 1.3$ ) is applied. Another question that arises is, how the gradients are computed. Two variants are tested. First the common central differences (they were also used in [1]) are calculated, consider for example the kernel

$$\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad (3.16)$$

in x-direction. In some cases a so called 'advanced Sobel filter' is applied in order to handle noise. The filter mask in x-direction in this case is

$$\frac{1}{32} \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}. \quad (3.17)$$

The difference between these two methods for gradient computation are not visible most of the time. In some cases, especially on real images, a Gaussian filter is used to preprocess the image. However, it does not always help. The same is true for the Kiel method.

Another interesting problem in the implementation is posed by the i2D case. It is described in section 2.2 that there exist three possibilities for the computation of the vectors  $\mathbf{u}$  and  $\mathbf{v}$ . When using one of them, there are always some combinations of the two orientations where the estimation does not work properly and results are completely wrong or even infinite for one of the three possibilities. If, however, all three possibilities are used there are usually at least two results that are close to the correct one. In order to rule out a possible outlier, the two results with the smallest squared distance are selected and the average of them is set as the final result.

This concludes the description of the implementation including the search of suitable bandpass filters for the Kiel method. The next step is a comparison of both methods and is described in the next chapter.



## 4 Comparison

After programming the Kiel method and the Lübeck method, one main task in this thesis is the comparison of both methods. In this chapter both methods are tested on a series of images; some are generated synthetically in order to test the absolute accuracy, others are real images to see if the methods work on these as well.

### 4.1 Local Analysis of Synthetic Images

At first, both methods are tested on synthetic images. For these tests, two signals with different orientations are overlaid additively. When the signals are computed, orientations, phases, and wave lengths are known in advance. The size of these images is determined by the range or kernel size, meaning the image has the same size as the kernel. The local main orientation and apex angle are only computed for the pixel at the origin. Let  $\theta_1 \in [0^\circ, 180^\circ)$  and  $\theta_2 \in [0^\circ, 180^\circ)$  be the two orientations. With  $\lambda_1 \in \mathbb{R} > 0$  and  $\lambda_2 \in \mathbb{R} > 0$  being the two wave lengths and  $\varphi_1 \in [0^\circ, 360^\circ)$  and  $\varphi_2 \in [0^\circ, 360^\circ)$  being the two phases, the signal  $s \in \mathbb{R}$  for the coordinates  $(x, y) \in \mathbb{R}^2$  is computed in the following way:

$$\begin{aligned} s_1 &:= \frac{(\sin(\theta_1)x + \cos(\theta_1)y)2\pi}{\lambda_1} - \varphi_1 \\ s_2 &:= \frac{(\sin(\theta_2)x + \cos(\theta_2)y)2\pi}{\lambda_2} - \varphi_2 \\ s &:= \cos(s_1) + \cos(s_2). \end{aligned} \tag{4.1}$$

These signals are repeated over the whole image. Otherwise  $s_1$  and  $s_2$  are restricted to lie in  $[-\pi, \pi]$ , the resulting signal is therefore referred to as being single. See figure 4.1 for examples with 201 pixels,  $\varphi = 45^\circ$ ,  $\lambda = 0.5$  for both signals and orientations  $\theta_1 = 20^\circ$  and  $\theta_2 = 60^\circ$ .

The first test consists of calculating the two orientations for all combinations of angles and comparing the results with the known real angles. This is done with the single and the repeated forms of the images. The images are computed with 41 pixels, but otherwise they have the same parameters as in the example above. Since  $\varphi_1 = \varphi_2 = 45^\circ$ , the restriction of arbitrary, but same phases is fulfilled for all the images. Each method is tested on all of the  $180 \times 180$  images. The results are compared with the known correct main orientations and apex angles.

The absolute values of the differences are accumulated and yield a possibility to calculate

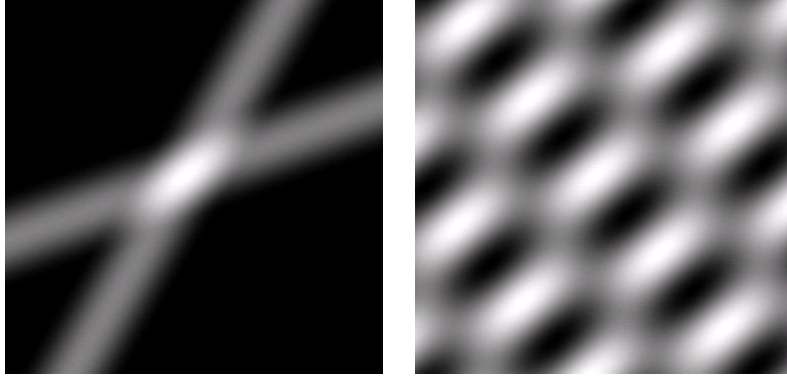


Figure 4.1: Left: Image with orientations  $20^\circ$  and  $60^\circ$ , composed of two single overlaid signals. Right: The same image composed of two repeated overlaid signals

the average error and compare both methods:

$$\begin{aligned} \text{Error}_{\text{apex}} &:= \frac{1}{180^2} \sum_{\theta_1 \in [0,180)} \sum_{\theta_2 \in [0,180)} |\text{result}_{\text{apex}} - \text{real}_{\text{apex}}| \\ \text{Error}_{\theta_{\text{main}}} &:= \frac{1}{180^2} \sum_{\theta_1 \in [0,180)} \sum_{\theta_2 \in [0,180)} |\text{result}_{\theta_{\text{main}}} - \text{real}_{\theta_{\text{main}}}| \end{aligned} \quad (4.2)$$

The following table shows the results of these tests:

	Parameters	Single Image	Repeated Image
Riesz Transform Method	LOP-parameter = 2.6, i1D/i2D = 0.0000035, kernel size = 41	Error <sub>apex</sub> : 0.1548° Error <sub>θ<sub>main</sub></sub> : 0.0096°	Error <sub>apex</sub> : 0.0926° Error <sub>θ<sub>main</sub></sub> : 0.0246°
Gradient Method	kernel size = 41, σ = 1.3, ε <sub>1</sub> = 0.000011, ε <sub>2</sub> = 1000.0	Error <sub>apex</sub> : 1.5727° Error <sub>θ<sub>main</sub></sub> : 1.1164°	Error <sub>apex</sub> : 1.7497° Error <sub>θ<sub>main</sub></sub> : 1.4304°

The Kiel method is run with parameters that were determined during the experiments. The LOP-parameter defines the size of the pass band of the bandpass filter. The i1D/i2D parameter is a boundary below which the determinant of the even tensor needs to be in order to consider the structure to be intrinsically one-dimensional at the given point. This parameter is set once and kept constant for all the following experiments, a fact that makes the method much more comfortable to use than the method developed in Lübeck where the parameters  $\epsilon_1$  and  $\epsilon_2$  are determined manually for almost every image. When experimenting with those parameters, it becomes obvious that the choice of  $\epsilon_1$  is a trade-off between a high Error<sub>apex</sub> and a high Error<sub>θ<sub>main</sub></sub> because a fairly large mistake is made around the nearly intrinsically one-dimensional structures, meaning when the apex angle is very small. The same is true to a lesser extent for the method using the monogenic signal. This can also be observed on the error surfaces in figure 4.2. This trade-off behavior is the reason why both errors are considered separably instead of creating one measure including both errors. By considering them separately, the parameter can be determined more easily.

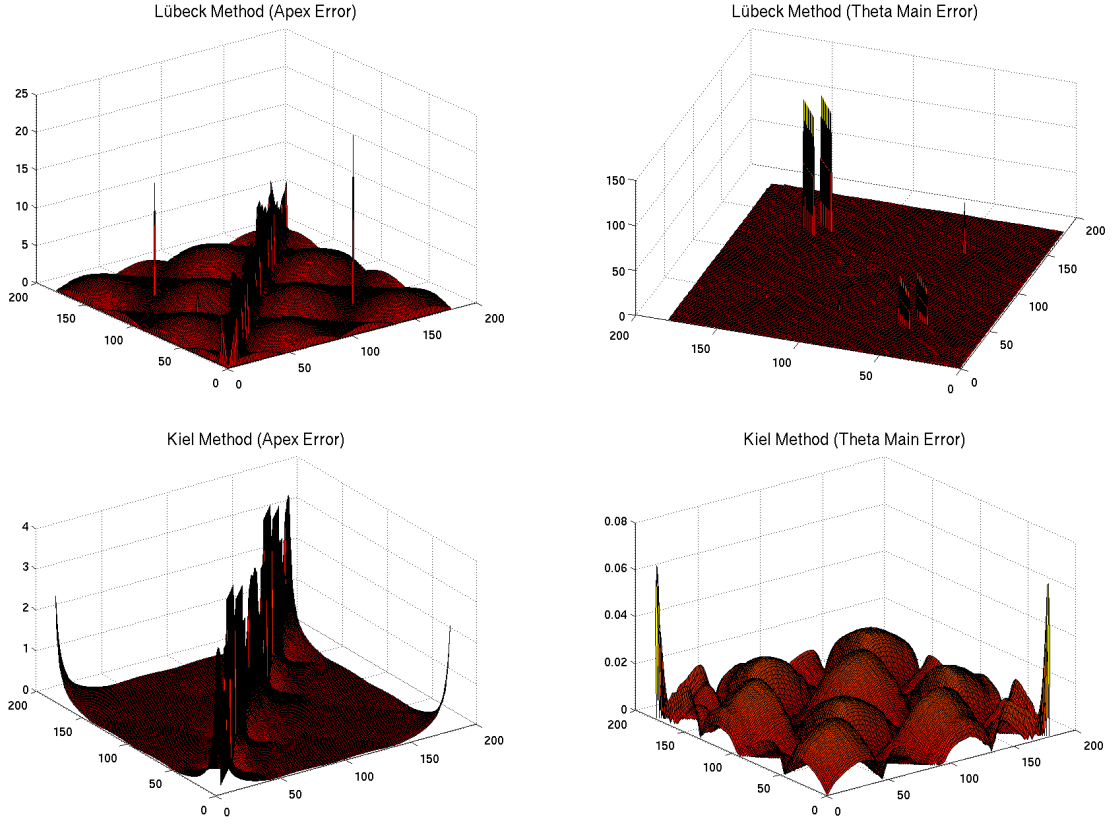


Figure 4.2: Error surfaces for the single overlaid signals. For each angle combination between  $0^\circ$  and  $180^\circ$  the absolute errors between real and calculated apex angle and real and calculated main orientation are determined. Top row: Lübeck method. Bottom row: Kiel method. Left: Apex angle error, Right:  $\theta_{main}$  error. Important to note are the actual heights of the error surfaces. The z-axis on the top left has a range from  $0^\circ$  to  $25^\circ$ , the z-axis on the top right from  $0^\circ$  to  $140^\circ$ , the bottom left ranges between  $0^\circ$  and  $4^\circ$  and the bottom right between  $0^\circ$  and  $0.07^\circ$ .

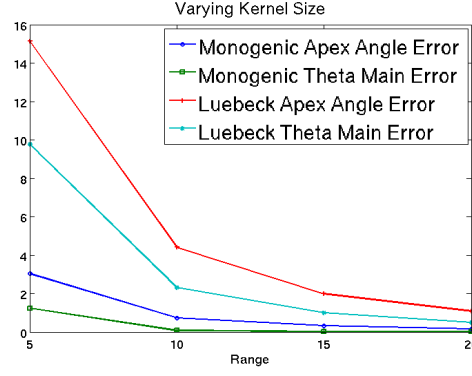


Figure 4.3: Development of the average error of apex angle and main orientation when the kernels size varies.

Other tests include the invariance against changes in wavelength, kernel size, and phases. Although the phases of the two signals need to be equal for the monogenic method to work, some tests with different phases are also conducted in order to see how the method behavior changes if this assumption is violated. This time, not all 32400 combinations of orientations are tested through. Instead, both orientations are raised by  $5^\circ$  in every step. This leads to a number of  $36^2 = 1296$  different tests. Accordingly, the average errors are calculated. The basic set of parameters is listed in the following table.

Parameter	Riesz Transforms	Lübeck Method
signal type	single	single
i1D/i2D parameter	$\det(\text{even}) \leq 0.00000035$	$\epsilon_1 = 0.00001$
LOP kernel	3.0	-
$\sigma$	-	1.3
kernel size	41	41
$\varphi_1$	$45.0^\circ$	$45.0^\circ$
$\varphi_2$	$45.0^\circ$	$45.0^\circ$
$\lambda_1$	0.5	0.5
$\lambda_2$	0.5	0.5
step size	5	5
$\text{Error}_{\text{apex}}$	$0.166^\circ$	$1.081^\circ$
$\text{Error}_{\theta_{\text{main}}}$	$0.011^\circ$	$0.492^\circ$

During the tests conducted, this was the basic set of parameters with the corresponding results in form of the error sums. In the following, single parameters are varied and the results are recorded and plotted.

**Kernel Size Modification** The first test consists of modifying the kernel size. Throughout the whole thesis kernels are computed using a certain range and the final kernel size is  $2 \cdot \text{range} + 1$  in order to ensure odd kernel sizes. The tested ranges are 5, 10, 15 and 20, the last one corresponding to the kernel size of 41 in the basic parameter set.

The results are shown in figure 4.3. Normally, the kernel size has to fit the frequency range of the image. However, since the same image is used for all tests, the stability of the

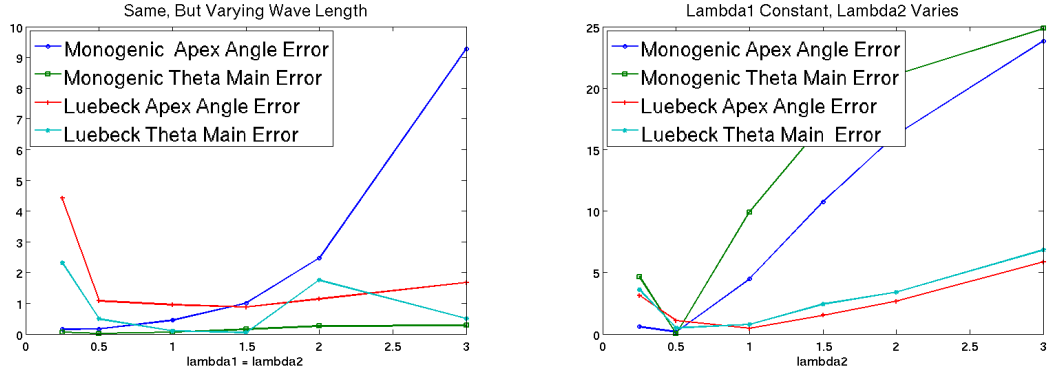


Figure 4.4: Left: Development of the average error of apex angle and main orientation when both wave lengths are the same but vary. Right: Development of the average error of apex angle and main orientation, when one wave length is kept constant but the other one varies.

algorithms with smaller kernel sizes and therefore greatly reduced cost of computation time is tested. One can see that the Kiel method is more stable when the kernel size is smaller. Even if the size is reduced to  $11 \times 11$ , the average error is below  $4^\circ$  for both, apex angle and main orientation.

**Wave Length Modification** This time the kernel size is kept constant but the wave lengths differ. First, both wave lengths are the same, but vary. The result is shown in figure 4.4 on the left. Changing the wave length corresponds to changing the size of the structure relative to the kernel size or - viewed in the frequency domain - to changing the frequency of a structure. Using smaller structures, i.e. smaller wave lengths leads to better results using the Riesz transforms. With growing wave lengths, however, the error when determining the apex angle with the monogenic signal is growing up to an average error of  $9^\circ$ . If only one wave length is changed while the other one is kept constant - meaning one structure is larger than the other - both methods make an increasing error in the computation of apex angle and main orientation, however, the method developed in Lübeck delivers more accurate results while the Kiel method is not usable anymore, since the average error is around  $20^\circ$ . Because a bandpass filter is used that only allows very few frequencies to pass, the change of one wave length - and therefore frequency - probably causes one structure to lie outside of the pass band. If this is the case, it is not surprising that the feature detection using the Kiel method fails.

**Phase Modification** In a third test concerning the change of parameters of the synthetic images, the phases of the images are changed. Like in the test above, at first both phases are the same, but vary. Images that are produced under these circumstances do not violate the assumption of equal phases that is made in the Kiel method. The results of the test are very good for both methods (see figure 4.5). The Kiel method is slightly better, except when both phases equal zero. This is not surprising because it can be proven that the orientation estimation is undefined when the phase equals zero. If the phases are different, the main orientation estimation using the Kiel method is

## 4 Comparison

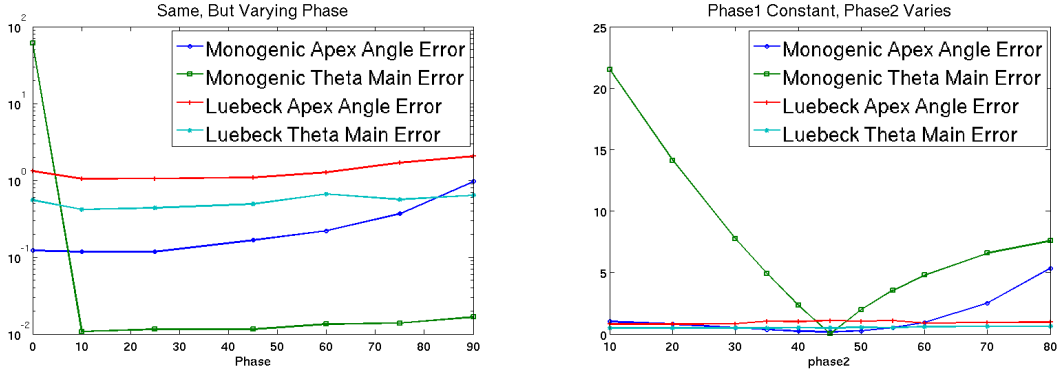


Figure 4.5: Left: Development of the average error of apex angle and main orientation when the phases of both signals are the same, but vary. Right: Development of the average error of apex angle and main orientation when one phase is kept constant, but the other one varies.

not possible. Already slight changes in the phase lead to large errors in the orientation estimation. The estimation of the apex angle, however, is surprisingly stable. This fact raises hope for the use of the apex angle in other practical applications like optical flow estimation and rotation estimation between two images. The Lübeck method is far more stable against phase differences, in fact, after further adjustments to the i1D/i2D factor  $\epsilon_1$ , very good results with error rates even below the ones for equal phases are achieved.

## 4.2 Global Tests on Images

In this section the orientations are computed over whole images. Hitherto, the apex angle and local main orientation are only calculated for the pixel in the origin of the synthetic images. In this section some results are presented where both orientations are indicated by arrows. These arrows are drawn by passing the matrices containing the orientations on to Matlab. Since it is impossible to draw the orientations for every single pixel, a rectangle filter is used to average the results for several neighboring pixels. The size of the rectangle filter corresponds to the number of orientations that are summarized in one arrow.

In order to get a detailed picture of the performance of both methods, a lot of test images are used. The following presentation of the results reveals some difficulties that each method has on different images. For all images the parameters  $\epsilon_1$  and  $\epsilon_2$  that are used by the Lübeck method for distinguishing between i1D and i2D structures are adapted. In addition to that, sometimes the kernel size is adapted for both methods. When using the Kiel method, there are some cases where the LOP-parameter characterizing the bandpass filter is adapted to the image. However, if possible, the same parameters are used. The test images are gained in different ways. Some are simply taken by a digital camera of the Canon Powershot series. Others are obtained by the CD-ROM accompanying [14]. There are also some images that are computed synthetically or are copied from the papers describing the Lübeck method.



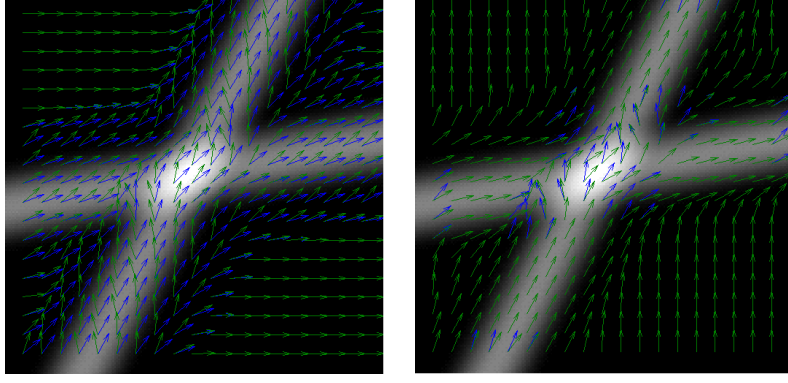


Figure 4.6: Left: Lübeck method results using  $\sigma = 1.3$  and kernel size = 21. Right: Kiel method's results using 3 as the LOP parameter and 21 as the kernel size.

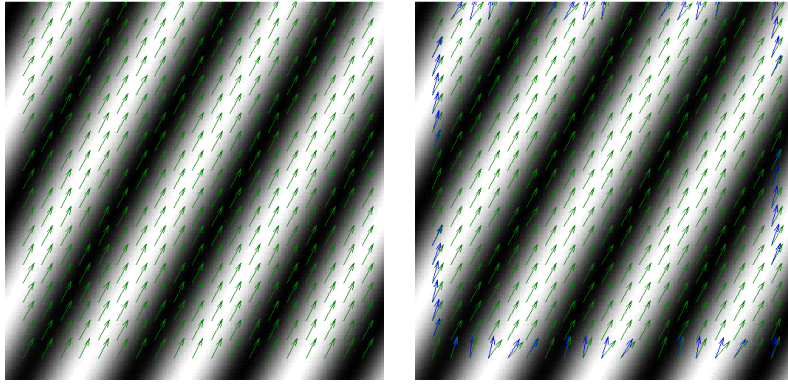


Figure 4.7: Left: Lübeck method using the same parameters as in figure 4.6. Right: Kiel method using 21 as kernel size and 5 as LOP parameter.

### 4.2.1 Synthetic Images

When the orientations are calculated over the whole synthetic image instead of solely for the pixel at the origin, there are some difficulties at the edges of the images but apart from that, the detection is accurate. Examples with a single i2D case and a repeated i1D case are shown in figures 4.6 and 4.7 respectively. 4.6 presents the results of the orientations estimation for two single signals with  $10^\circ$  and  $62^\circ$ . It becomes obvious that the detection of the local i1D parts works better using the Kiel method. When applying the Lübeck method, the parameters for distinguishing between i1D and i2D have to be adjusted for each image. In some cases it is very difficult to find parameters that estimate the intrinsic dimension correctly over the whole image.

The repeated i1D case shows an orientation of  $62^\circ$  degrees. Both methods deal well with this image, as can be seen in image 4.7.

The original image in the upper row of figure 4.8 is taken from [19]. It shows two overlaid signals. The results of the Lübeck method for this image are very good - as good as the ones in the original paper - which shows that the implementation works. The Kiel method has some difficulties with this synthetic image. In some pixels the orientations

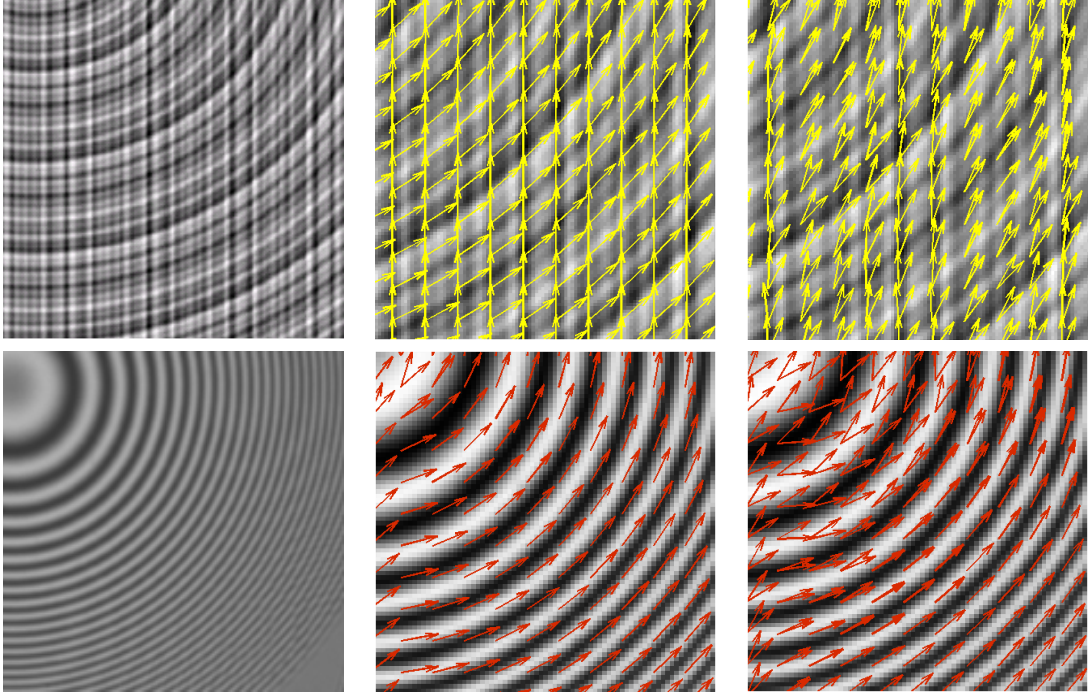


Figure 4.8: Top left: Synthetic Image from [19]. Top middle: Upper right corner of result image using Lübeck method, kernel size is 21,  $\sigma = 1.3$ . Top right: Same clipping using the Kiel method, same kernel size as in Lübeck method, LOP parameter is 5. Bottom left: Synthetic Image from [14]. Bottom middle: Upper left corner of Lübeck results. Bottom right: Same clipping using Kiel method. Both methods are run with the same parameters as in the top row.

are determined accurately, but in many points the method fails. Reasons for this can be found in the more detailed analysis, that is done in section 4.1. The tests with different phases and wave lengths reveal some weaknesses of the Kiel method that are probably the reason why it fails at so many points of this test image. The bottom row in figure 4.8 shows a synthetic image containing a round structure. The results are taken from the upper left corner. On this test image the Lübeck method yielded better results than the Kiel method. The estimated orientations are more closely aligned with the structure and a smaller part of the structure is considered to be intrinsically two-dimensional. This is also true for the clipping of the image containing the smaller part of the structure.

### 4.2.2 Corners and Edges

In this section, the detection of orientations at corners and edges of two different synthetic images is analyzed closer. Zooming in closely helps getting an idea of the problems around sharply defined structures. In a preprocessing step a Gaussian filter is applied to the image in order to improve the detection of sharply defined edges. Figure 4.9 shows the images of two simple boxes and the orientation estimation at the upper right corner. Both images are created using a software similar to Microsoft's Paint. Therefore, the edges are defined sharply as can be seen in the result images. Both methods have

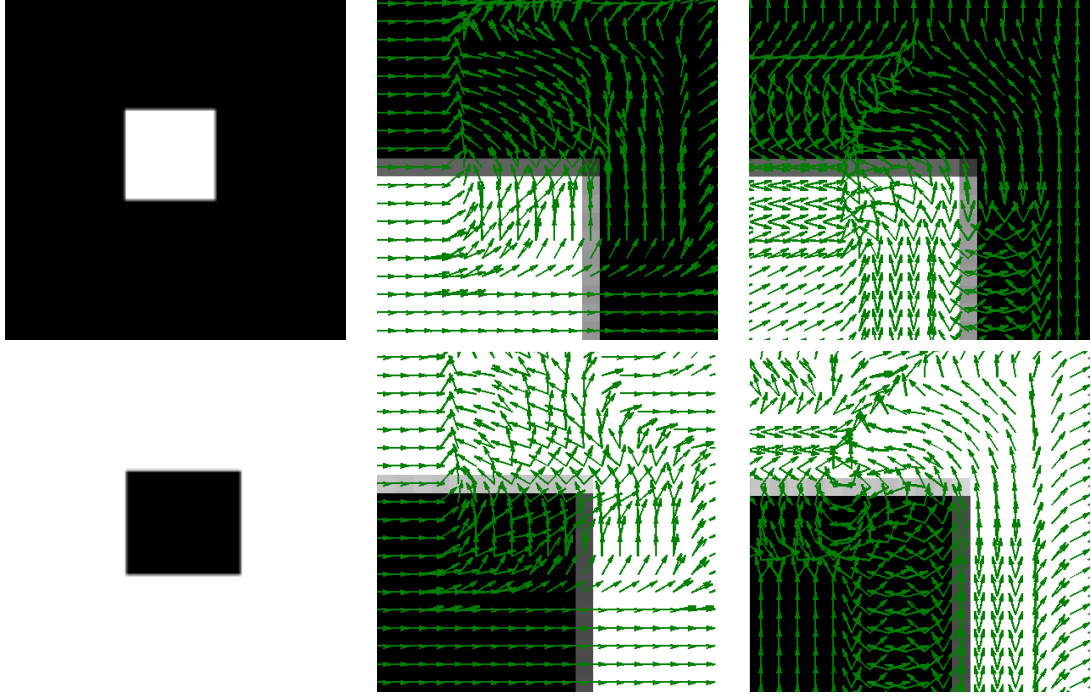


Figure 4.9: Left: Original images showing boxes drawn with Microsoft’s Paint. Middle: Results using the Lübeck method. Right: Results using the Kiel method. All result images show the orientations around the upper right corner.

difficulties with these images. The Lübeck method cannot detect sharp edges; a valid result is only delivered when a corner is present. This problem is examined closer in the next test image.

When using the Kiel method, the edges are determined to be intrinsically two-dimensional. In addition to that, some of the orientations are determined incorrectly. The problem is that the arrows are always averages over several pixels. For one of these pixels, a case occurs where the phase of the structure equals zero. In that case, the main orientation estimation is not possible - a large error arises. The averaging propagates the error that probably happened in one pixel only, over several pixels. This problem can only be solved through phase computation and discarding of the orientation estimation results at pixels where the phase equals zero. When using the LOP kernels, the phase needs to be computed using the second or third order Riesz transforms which is possible as Wietzke showed in [23]. However, due to a limited amount of time, it is not implemented for this thesis.

The original image in figure 4.10 shows several different structures with i1D and i2D areas. The upper end of the vertical edge is closely analyzed. The results of the Lübeck method in the upper row show that the vertical edge is ignored. It is only detected near the end of the line where the intrinsic dimension is two. When the image is preprocessed with a Gaussian filter, the vertical edge is detected correctly as seen in the bottom row. The problem which the Kiel method has at pixels where the phase equals zero also occurs in this image. A Gaussian filter does not help with this problem because there still is a position where the phase is zero.

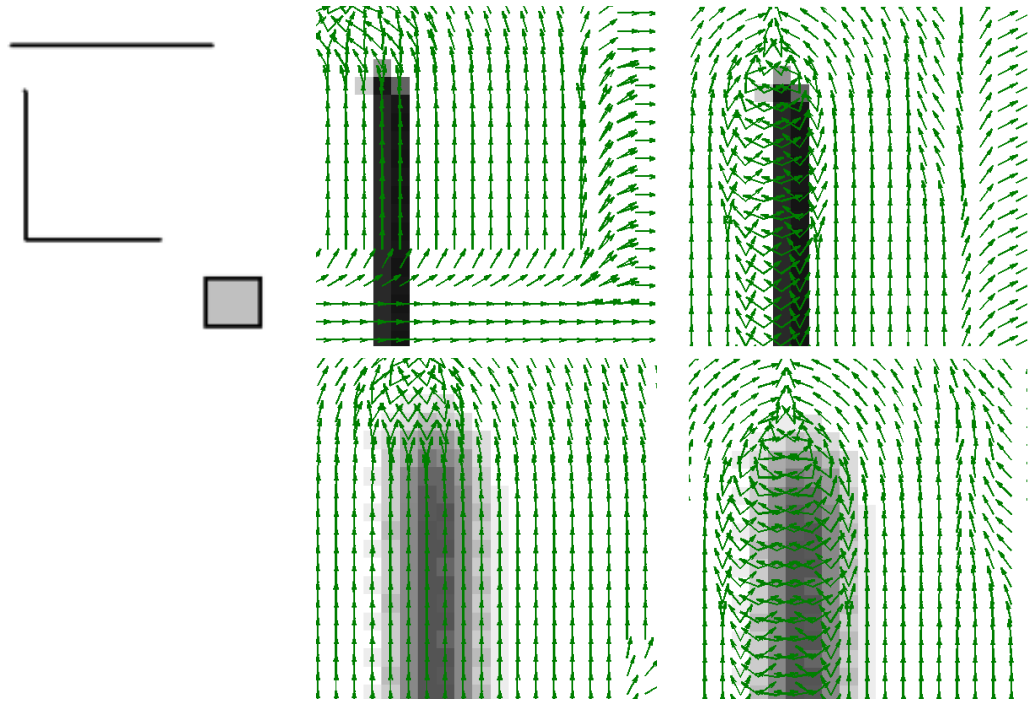


Figure 4.10: Left: Original image. Middle: Results at upper end of i1D signal after applying the Lübeck method. Right: Results using the Kiel method. Upper row: Original image is used. Bottom row: A Gaussian filter is applied as a preprocessing step.

### 4.2.3 Textures or Fabrics

One application of orientation estimation is texture analysis. If a texture has two main orientations and they are both known, it is possible to separate the different layers to a certain extent. This is already described by for example [1] and [19]. In this section some test pictures and the orientation estimation on textures using both methods are shown.

Figure 4.11 contains the analysis of two different patterns of fabric. For the first one (top), the Kiel method detects the local orientations remarkably well. The arrows seem to reflect the whole pattern. The Lübeck method has more trouble. The intrinsic dimension is detected correctly but the apex angle is usually too small. The second pattern in the bottom row shows a piece of fabric with a stitched pattern. This test image is a big challenge for both methods. The background part without stitches is not detected correctly by either method. In the foreground part, the Kiel method detects the two orientations, one vertical, one with an angle of about  $135^\circ$  better than the Lübeck method.

### 4.2.4 Images Containing Noise

The performance of both methods on images containing noise is not measured by adding artificial noise to the images but by using real images that contain noise. The test images consist of one i1D and two i2D examples. All three of them were taken with a digital



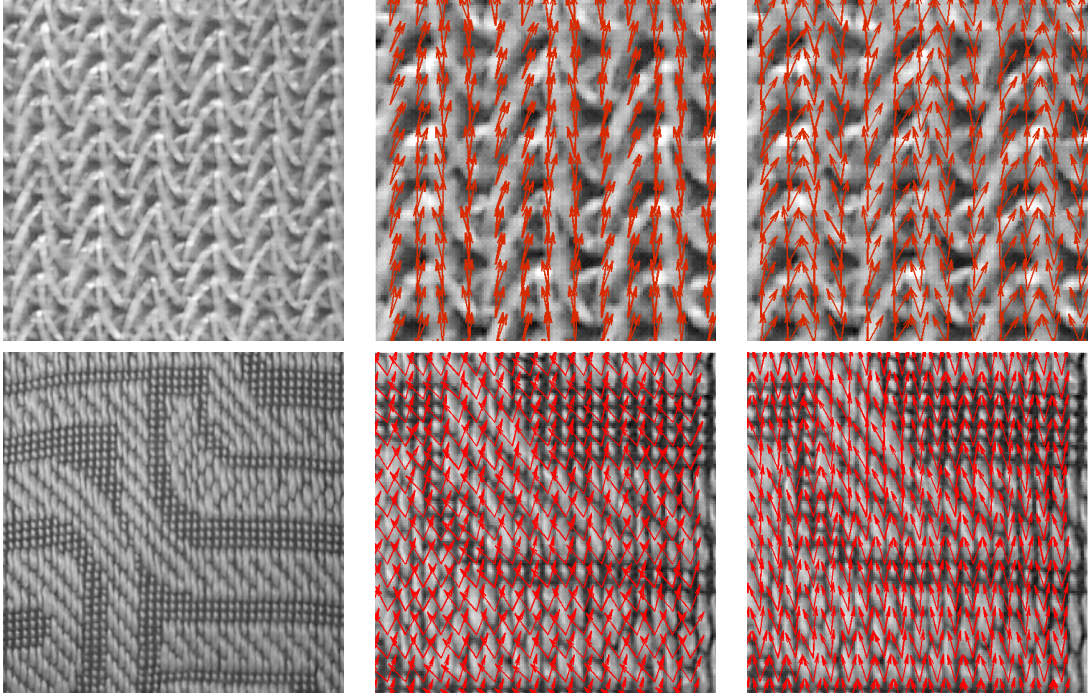


Figure 4.11: Upper row, left: Fabric photographed up close. Middle: Results from Lübeck method. Right: Results using Kiel method. The clippings of the results both stemmed from the upper middle of the original image. Bottom row, left: Fabric with stitches. Middle: Lübeck method results clipping of upper right corner. Right: Kiel method results, same clipping.

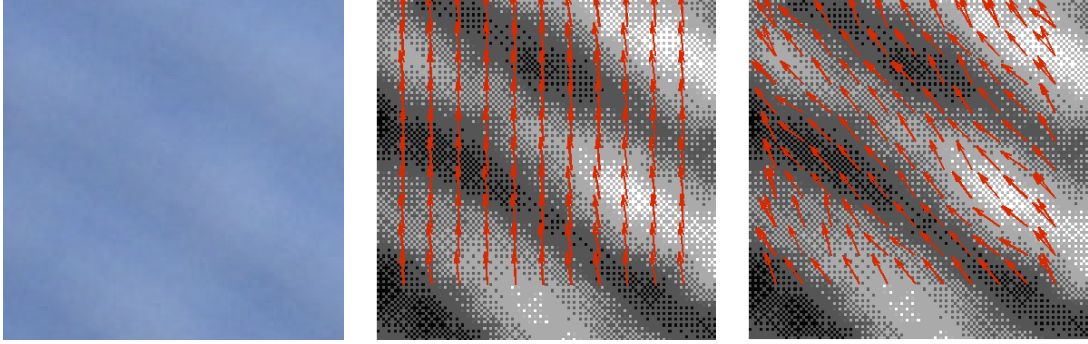


Figure 4.12: Left: Original image featuring a piece of fabric. Middle: Results using the Lübeck method after preprocessing with a Gaussian filter. Right: Results of the Kiel method without applying a Gaussian filter.

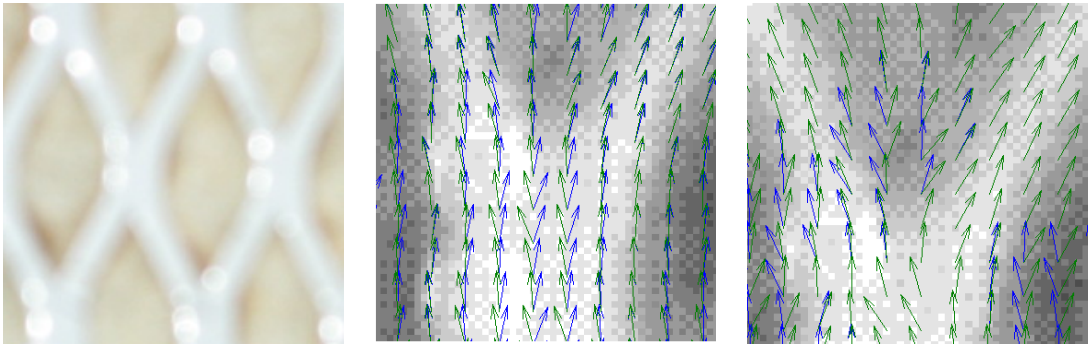


Figure 4.13: Left: Original image containing the bottom part of an ironing board. Middle: Results of the Lübeck method, shown is the left complete junction. Right: Same clipping with the results of the Kiel method.

camera. The noise occurs because all images were photographed from a short distance.

Figure 4.12 shows a i1D structure containing noise: a pattern on a piece of cloth. The original image is blurred and the human eye can hardly detect any orientation. It is obvious that the Kiel method works better on this test image. The arrows representing the resulting orientations of the Lübeck method only point upwards. The Kiel method detected the orientation of approximately  $135^\circ$  accurately for most pixels.

In figure 4.13 the bottom structure of an ironing board can be seen. The result images show noise in a clipping of an i2D structure. Despite the noise, both methods detected the orientations in this test image to a certain degree. They distinguish the local i1D and i2D areas correctly for the most part. The i2D structure is detected by both methods, however, the results differ. The Lübeck method detects it a little closer to the actual crossing than the Kiel method but both methods' orientations point in the right direction.

In the test image in figure 4.14 a piece of fabric with overlaid i1D signals can be observed. The black and white images show that the image contains noise but both methods are able to detect the orientations and intrinsic dimension accurately. Shown is the clipping

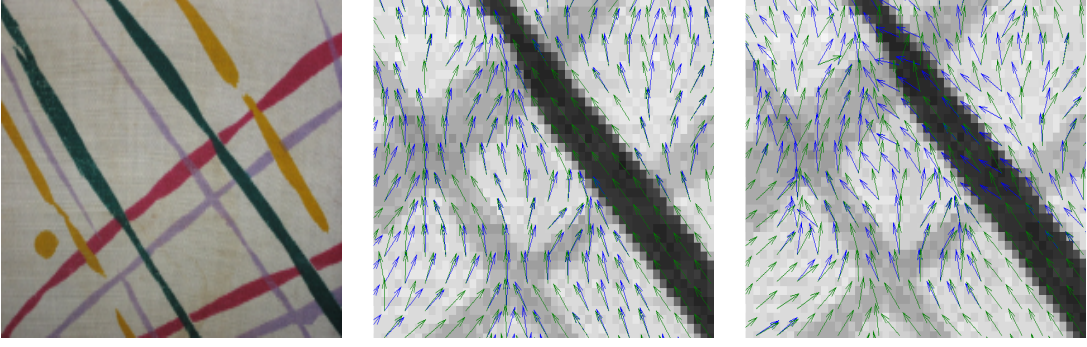


Figure 4.14: Left: Original image displaying the fabric of an ironing board. Middle: Clipping of results after applying the Lübeck method. Right: Clipping of results using the Kiel method.

containing several junctions of i1D structures. The Kiel method estimates the different orientations in these junctions more accurately and even detects the junctions where the dark structure crosses the lighter ones.

The tests on the synthetic and real images revealed shortcomings of both methods, but also demonstrated that both work in general. The most important shortcoming of the Lübeck method is the parameter  $\epsilon_1$  that has to be adjusted manually for each image. It ensures that the i1D and i2D areas of the image are distinguished properly. The greatest disadvantage of the Kiel method is the constraint concerning the phases of the two overlaid structures - they have to be the same - which greatly reduces the usability of the method. The parameter for the LOP kernel for the Kiel method improves the distinction between the intrinsic dimensions in an indirect way - by filtering convenient frequencies - but is by far not as delicately to adjust. Another advantage of the Lübeck method is that it can calculate both orientations unambiguously, something the Kiel method cannot always do due to the estimated apex angle being the smaller angle of the two angles. The Kiel method can deal with images containing noise far better, but the Lübeck method is more robust against changes in phase or wave length. This concludes the comparison of both methods. The next chapter describes an application of the single order Riesz transforms.





## 5 Application: Rotation Estimation

Riesz transforms can be used in different practical applications as for example the phase vector estimation in the calculation of the optical flow. Another application presented here, is the estimation of a rotation angle between two otherwise identical images. The theoretical background for this application is a direct conclusion from the definition of the single order Riesz transform and its interpretation using the Radon transform. The approach works independently of intrinsic dimension on synthetic and real images.

Input for this task are two images containing the same signal but are rotated by a certain angle (see in figure 5.1). Both input images contain a single signal and the second image is rotated by  $\beta = 60^\circ$ . The first step consists of computing the Radon transform of each image. According to the first and second input image  $img_1$  and  $img_2$  respectively, the Radon transforms are denoted by  $r_1(\theta, t)$  and  $r_2(\theta, t)$ . A rotation by a certain angle  $\beta$  between two images corresponds to a translation along the  $\theta$ -axis in Radon space (see figure 5.1). If the rotation angle is denoted by  $\beta$ , the following equation is true for the Radon transforms of the images:

$$r_1(\theta, t) = r_2(\theta + \beta, t), \quad \text{with } \theta \in [0^\circ, 360^\circ]. \quad (5.1)$$

The estimation of the rotation is achieved by calculating the first order Riesz transforms in x- and y-direction for both images. For the first image, the Riesz transforms read

$$\begin{aligned} R_{x_1} &= \mathcal{R}^{-1}\{\cos(\theta)h_1(t) * r_1(\theta, t)\}, \\ R_{y_1} &= \mathcal{R}^{-1}\{\sin(\theta)h_1(t) * r_1(\theta, t)\}. \end{aligned} \quad (5.2)$$

For the second image, the angle is denoted by  $\gamma = \theta + \beta$ . For  $R_{x_2}$  follows (using the linearity of the integral)

$$\begin{aligned} R_{x_2} &= \mathcal{R}^{-1}\{\cos(\gamma)h_1(t) * r_2(\gamma, t)\} \\ &= \mathcal{R}^{-1}\{\cos(\theta + \beta)h_1(t) * r_2(\theta + \beta, t)\} \\ &= \mathcal{R}^{-1}\{(\cos(\theta)\cos(\beta) - \sin(\theta)\sin(\beta))h_1(t) * r_1(\theta, t)\} \\ &= \cos(\beta)\mathcal{R}^{-1}\{\cos(\theta)h_1(t) * r_1(\theta, t)\} - \sin(\beta)\mathcal{R}^{-1}\{\sin(\theta)h_1(t) * r_1(\theta, t)\} \\ &= \cos(\beta)R_{x_1} - \sin(\beta)R_{y_1}. \end{aligned} \quad (5.3)$$

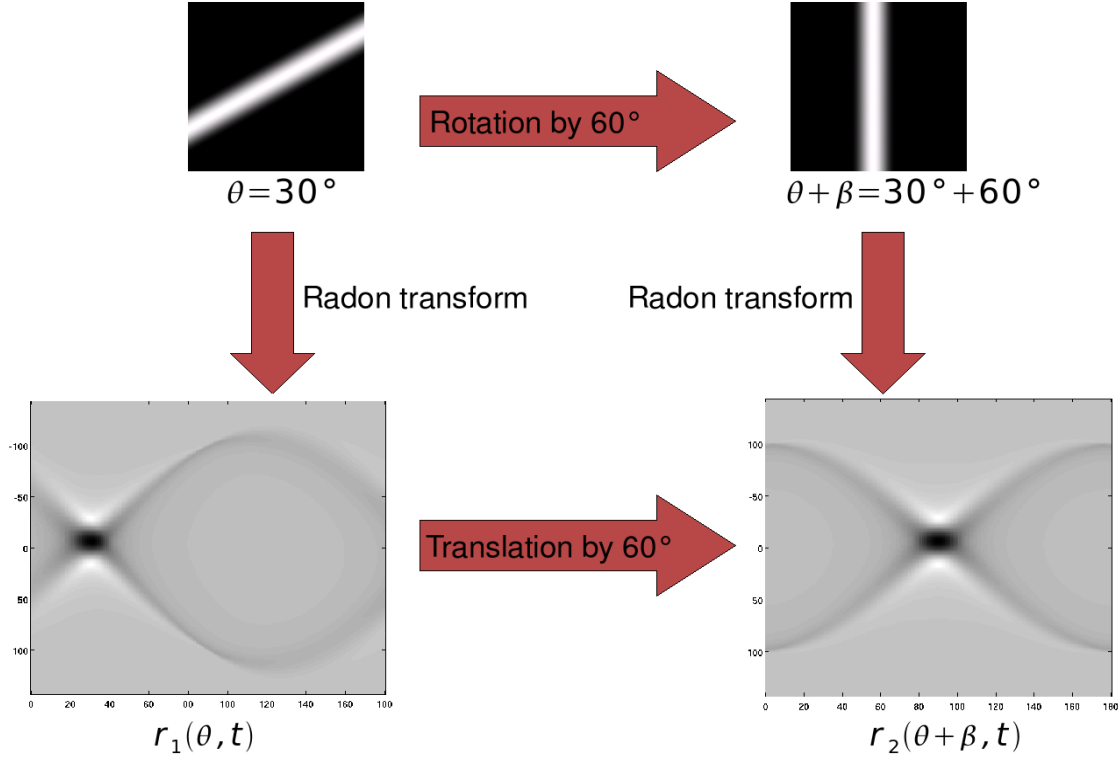


Figure 5.1: Rotation estimation is possible because the rotation corresponds to a translation in Radon space.

The Riesz transform in y-direction can be reduced to

$$\begin{aligned}
 R_{y_2} &= \mathcal{R}^{-1}\{\sin(\gamma)h_1(t) * r_2(\gamma, t)\} \\
 &= \mathcal{R}^{-1}\{\sin(\theta + \beta)h_1(t) * r_2(\theta + \beta, t)\} \\
 &= \mathcal{R}^{-1}\{(\sin(\theta)\cos(\beta) + \sin(\beta)\cos(\theta))h_1(t) * r_1(\theta, t)\} \\
 &= \cos(\beta)\mathcal{R}^{-1}\{\sin(\theta)h_1(t) * r_1(\theta, t)\} + \sin(\beta)\mathcal{R}^{-1}\{\cos(\theta)h_1(t) * r_1(\theta, t)\} \\
 &= \cos(\beta)R_{y_1} + \sin(\beta)R_{x_1}.
 \end{aligned} \tag{5.4}$$

Using equations 5.3 and 5.4, a system of linear equations needs to be solved in order to determine  $\cos(\beta)$  and  $\sin(\beta)$  which deliver the needed angle  $\beta$ .

## 5.1 Implementation and Results

The implementation of this method is tested. The first order Riesz transforms are already implemented and are thus applied on the two input images. After that, the system of equations is solved. These steps are applied to each pixel of the images. For more robustness, a  $3 \times 3$  rectangle filter is used to average the results. Only the region in the middle of the images can be used for rotation estimation because near the edges parts of the image might be invisible due to the rotation. The rotation routine comes

from the Qt class QPainter. The implementation only works for images with same edge length in x- and y-direction.

In order to test the rotation estimation implementation, three test images are rotated. As measure of error the average angular error (AAE) is computed. The image is rotated by each angle of  $[0^\circ, 360^\circ)$ . The average angular error is defined as  $AAE := \frac{|realRotation - estimatedRotation|}{\#tests}$ . The following table shows the test images, some parameters, and the resulting average angular error.

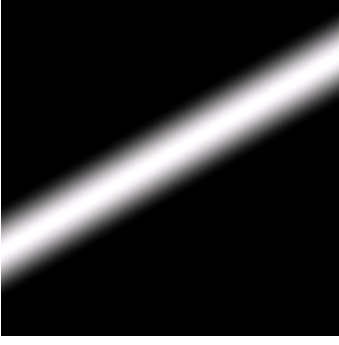


Image size:  $201 \times 201$ ,  
Coarse: 6, Fine: 3,  
Range: 10  
**AAE** =  $0.27144^\circ$



Image size:  $256 \times 256$ ,  
Coarse: 6, Fine: 3,  
Range: 10  
**AAE** =  $2.79329^\circ$



Image size:  $256 \times 256$ ,  
Coarse: 6, Fine: 3,  
Range: 10  
**AAE** =  $2.1518^\circ$

The resulting AAEs show that the method works well on the given test images. Overall, the conclusion is that rotation estimation is a working application resulting directly from the interpretation of the Riesz transform using the Radon transform.



## 6 Conclusion and Outlook

This chapter concludes this thesis and summarizes the results.

The Riesz transform is interpreted using the Radon transform which yields, especially for the higher order Riesz transforms, new possibilities for feature detection on images. The features include the local main orientation and apex angle. The derivation of a suitable bandpass filter - the Laplacian of Poisson - allows the direct derivation of convolution kernels for the second and third order Riesz transforms.

The Lübeck method uses gradients on images to compute the orientations of up to two overlaid i1D signals. By examining the structure tensor in the i1D case, a measure for intrinsic dimension is derived. In the i2D case, a tensor accounting for both orientations is considered and a confidence measure is derived in a similar way. Both tensors allow the calculation of one and two orientations respectively.

After implementing both methods, a detailed comparison of the performance of both algorithms is done by applying them to synthetic and real images. This comparison revealed weaknesses and strengths of both methods. The greatest shortcoming of the Lübeck method is the fact that the parameters need to be adjusted manually for each image and that it is often not easy to find a working set of parameters. On the other hand, the most serious disadvantage that the Kiel method is the constraint of the two overlaid signals needing to have the same phase. This constraint greatly reduces the range of images, the method can be applied to.

A direct insight of the interpretation of the Riesz transform by the Radon transform yields an interesting application, the estimation of a rotation angle between two otherwise identical images. This application is programmed and tested on real and synthetic images.

There are still a lot of things that can be done to extend this and Wietzke's work. One important improvement would be the disposal of the constraint that both i1D signals need to have the same phase. Apart from that, a formal proof that the LOP kernel approximates the DOP kernel needs to be found. An interesting application is using the Riesz transform for separating different layers of a texture according to their orientation. This approach was already proposed and tested using the Lübeck method and is for example described in [19]. A more direct improvement of the existing implementation could be achieved by finding a more robust measure for intrinsic dimension. So far the case of intrinsic dimension zero is ignored but it could probably be detected by using the energy of a signal. In addition to that, the i1D/i2D distinction could be computed more robustly, if additionally the determinant of the odd tensor is considered. Finally, a more robust implementation of the phase and the phase vector using the second and third order Riesz transform could help solving the problem of orientation estimation at pixels where the phase equals zero. An implementation of the phase vector designed

especially for i2D structures could be applied in optical flow estimation as can be seen in [17].

# Bibliography

- [1] T. Aach, C. Mota, I. Stuke, M. Muhlich, and E. Barth. Analysis of superimposed oriented patterns. *IP*, 15(12):3690–3700, December 2006.
- [2] T. Aach, I. Stuke, C. Mota, and E. Barth. Estimation of multiple local orientations in image signals. In *Proceedings ICASSP-2004*, pages III 553–556, Montreal, May 17–21 2004. IEEE.
- [3] E. Barth. Bewegung als intrinsische geometrie von bildfolgen. In *Mustererkennung 1999, 21. DAGM-Symposium*, pages 301–308, London, UK, 1999. Springer-Verlag.
- [4] E. Barth, I. Stuke, and C. Mota. Analysis of motion and curvature in image sequences. In *SSIAI '02: Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, page 206, Washington, DC, USA, 2002. IEEE Computer Society.
- [5] R. N. Bracewell. *Fourier Analysis and Imaging*. Kluwer Academic/Plenum Publishers, 2003.
- [6] M. Felsberg. *Low-Level Image Processing with the Structure Multivector*. PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, 2002.
- [7] M. Felsberg. Optical flow estimation from monogenic phase. In B. Jähne, R. Mester, E. Barth, and H. Scharr, editors, *1st International Workshop on Complex Motion*, volume 3417 of *LNCS*, pages 1–13, 2006.
- [8] M. Felsberg and G. Sommer. A new extension of linear signal processing for estimating local properties and detecting features. In *DAGM-Symposium*, pages 195–202, 2000.
- [9] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144, December 2001.
- [10] M. Felsberg and G. Sommer. Structure multivector for local analysis of images. In *Multi-image analysis*, volume 2032 of *LNCS*, pages 93–104. Proc. Dagstuhl Workshop on Theoretical Foundations of Computer Vision, Springer-Verlag, Berlin, 2001.
- [11] M. Felsberg and G. Sommer. The structure multivector. In L. Dorst, C. Doran, and J. Lasenby, editors, *Applications of Geometric Algebra in Computer Science and Engineering*, pages 437–448. Proc. AGACSE 2001, Cambridge, UK, Birkhäuser Boston, 2002.

- [12] M. Felsberg and G. Sommer. The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision*, 21:5–26, 2004.
- [13] O. Fleischmann. *Analyzing Riesz Transforms of Arbitrary Order in Radon Space*. Student Research Project, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, September 2007.
- [14] B. Jähne. *Digitale Bildverarbeitung*. Springer-Verlag Berlin Heidelberg, 2005.
- [15] U. Köthe and M. Felsberg. Riesz-transforms versus derivatives: On the relationship between the boundary tensor and the energy tensor. In R. Kimmel, N. A. Sochen, and J. Weickert, editors, *Scale-Space*, volume 3459 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2005.
- [16] C. Mota, I. Stuke, and E. Barth. Analytic solutions for multiple motions. In *ICIP (2)*, pages 917–920, 2001.
- [17] A. Sedlazeck. Student research project: Optical flow from local phase. February 2007.
- [18] G. Sommer. *Computer Vision I*. Lecture Notes, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, 2007.
- [19] I. Stuke, T. Aach, E. Barth, and C. Mota. Analysing superimposed oriented patterns. In *Southwest04*, pages 133–137, 2004.
- [20] P. Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.
- [21] L. Wietzke. *Determine the apex angle of superimposed 1D structures*. Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, 2007.
- [22] L. Wietzke, G. Sommer, C. Schmaltz, and J. Weickert. Analysis of the curvature tensor from the viewpoint of signal processing. In *Minisymposium on Clifford Algebras in Mathematics and Applied Sciences, ICNAAM 2007, Corfu, Greece, 16-20 September 2007*. Wolfgang Sprößig (Freiberg, Germany), Klaus Gürlebeck (Weimar, Germany), 2007.
- [23] L. Wietzke, G. Sommer, C. Schmaltz, and J. Weickert. Differential geometry of monogenic signal representations. In B. McDonald, U. Franke, R. Klette, and G. Sommer, editors, *2nd Workshop Robot Vision*. ROBVIS08, Springer-Verlag, 2008.
- [24] D. Zang. *Signal Modeling for Two-Dimensional Image Structures and Scale-Space Based Image Analysis*. PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, 2007.
- [25] D. Zang and G. Sommer. Signal modeling for two-dimensional image structures. *J. Vis. Comun. Image Represent.*, 18(1):81–99, 2007.



# A Notations

Symbol		Description
$DOG$		Difference of Gaussian kernel
$DOP$		Difference of Poisson kernel
$\mathcal{F}\{I\}(u, v)$		2D Fourier transform of $I$ at $(u, v)$
$\mathcal{F}^{-1}\{I\}(x, y)$		Inverse 2D Fourier transform of $I$ at $(x, y)$
$(f * g)(x, y)$		2D convolution of $f$ and $g$ at point $(x, y)$
$G$		Gaussian kernel
$g_M(x, y)$		Monogenic signal of 2D function $g$
$I(x, y)$	$\in \mathbb{R}$	Image, 2D function $I : \mathbb{R}^2 \rightarrow \mathbb{R}$
$I_x(x, y)$	$= \frac{\partial I}{\partial x}(x, y)$	First derivative in x-direction of $I$ at $(x, y) \in \mathbb{R}^2$
$I_{xy}(x, y)$	$= \frac{\partial^2 I}{\partial x \partial y}(x, y)$	Second derivative in x- and y-direction
$LOG$		Laplacian of Gaussian kernel
$LOP$		Laplacian of Poisson kernel
$P$		Poisson kernel
$\mathbf{R}(x, y)$	$= \frac{-\mathbf{x}}{2\pi \mathbf{x}+s ^3}$	Riesz transform vector (x- and y-direction)
$\mathcal{R}\{I\}(\theta, t)$		Radon transform of $I$ at angle $\theta$ and distance $t$
$\mathcal{R}^{-1}\{r(\theta, t)\}(x, y)$		Inverse Radon transform of $r$ at $(x, y)$
$R_x(x, y)$	$= \frac{-x}{2\pi \mathbf{x}+s ^3}$	Riesz transform in x-direction
$R_x\{R_y\}(x, y)$		Second order Riesz transform in x- and y-direction
$R_y(x, y)$	$= \frac{-y}{2\pi \mathbf{x}+s ^3}$	Riesz transform in y-direction
$\theta_{main}$	$\in [0^\circ, 180^\circ)$	Local main orientation
$ \mathbf{x} $	$= \sqrt{x^2 + y^2}$	Length of vector $\mathbf{x} \in \mathbb{R}^2$



# B Riesz Transform Kernel Derivation

## B.1 Derivation of the First Order Riesz Transform Kernels with a DOP Filter

The following derivation may be found in [8]. It is presented here because the same arguments are needed for the derivation of the third order LOP kernels in the next section. This derivation is done for 2D transform pairs, so that the kernels may be used on 2D image data. The coordinates in the spatial and frequency domain are written as  $\mathbf{x} = (x, y) \in \mathbb{R}^2$  and  $\mathbf{u} = (u, v) \in \mathbb{R}^2$  respectively. When written in polar coordinates, the length of the vector will be denoted by  $|\mathbf{x}| = r$  and  $|\mathbf{u}| = q$ . The following derivation is done using the Poisson kernel because the derivation without use of the scale space is not needed here, however, it may be found in [8] or [10]. The derivation of the second kernel in y-direction is the same as the one for the x-direction, so only the latter will be presented here. In the frequency domain the kernel of the single order Riesz transform in x-direction reads (see section 3.1.1)

$$\mathcal{F}\{R_x * P\}(\mathbf{u}) := \frac{u}{|\mathbf{u}|} i \exp(-2\pi|\mathbf{u}|s). \quad (\text{B.1})$$

In order to get the kernel in the spatial domain, the Hankel transform is applied. Bracewell states on page 358 in [5] that the Fourier transform of a function of  $r^2$  is

$$\mathcal{F}\left\{\frac{1}{(1+r^2)^{\frac{1}{2}}}\right\}(\mathbf{u}) = \frac{1}{|\mathbf{u}|} \exp(-2\pi q), \quad \mathbf{u} \in \mathbb{R}^2. \quad (\text{B.2})$$

Now a substitution takes place, with  $r' = rs$ ,  $s > 0$  and in direct accordance to [8] follows

$$\begin{aligned} \mathcal{F}\left\{s \frac{1}{(s^2 + r'^2)^{\frac{1}{2}}}\right\}(\mathbf{u}) &= s^2 (qs)^{-1} \exp(-2\pi qs) \\ \Rightarrow \mathcal{F}\left\{\frac{1}{(s^2 + |\mathbf{x}|^2)^{\frac{1}{2}}}\right\}(\mathbf{u}) &= |\mathbf{u}|^{-1} \exp(-2\pi|\mathbf{u}|s). \end{aligned} \quad (\text{B.3})$$

The next step consists of applying the derivative theorem for 2D Fourier transform pairs to this function. This is described in [5] for a function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . With

$$\mathcal{F}\left\{\frac{\partial}{\partial x} f(x, y)\right\}(\mathbf{u}) = i2\pi u F(u, v) \quad (\text{B.4})$$

follows (using the linearity of the integral)

$$\begin{aligned} \mathcal{F} \left\{ \frac{-x}{(s^2 + |\mathbf{x}|^2)^{\frac{3}{2}}} \right\} (\mathbf{u}) &= 2\pi i \frac{u}{|\mathbf{u}|} \exp(-2\pi|\mathbf{u}|s) \\ \implies \mathcal{F} \left\{ \frac{-x}{2\pi(s^2 + |\mathbf{x}|^2)^{\frac{3}{2}}} \right\} (\mathbf{u}) &= i \frac{u}{|\mathbf{u}|} \exp(-2\pi|\mathbf{u}|s). \end{aligned} \quad (\text{B.5})$$

The derivation of the Riesz transform in y-direction is practically the same and the result is

$$\mathcal{F} \left\{ \frac{-y}{2\pi(s^2 + |\mathbf{x}|^2)^{\frac{3}{2}}} \right\} (\mathbf{u}) = i \frac{v}{|\mathbf{u}|} \exp(-2\pi|\mathbf{u}|s). \quad (\text{B.6})$$

## B.2 LOP Kernel

In section 3.1.4 the idea of the derivation of the second and third order Riesz transform kernels is described. Now this derivation is described in more detail. Section 3.1.4 shows the kernels of the second and third order transforms in x-direction. Now all the kernels will be listed.

### LOP Kernel without Riesz Transform

Section 3.1.4 already shows that the Laplacian of Poisson (LOP) is derived by taking the Laplacian of the Poisson filter in the frequency and in the spatial domain yielding

$$\begin{aligned} \mathcal{F}\{LOP\}(\mathbf{u}) &= -4\pi^2|\mathbf{u}|^2 \exp(-2\pi|\mathbf{u}|s) \bullet \circ \\ &\quad \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \frac{s}{2\pi|\mathbf{x} + s|^3} \\ &= -\frac{3s}{2\pi} \left( 2(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5(x^2 + y^2 + s^2)^{-\frac{7}{2}}(x^2 + y^2) \right). \end{aligned} \quad (\text{B.7})$$

The derivation is found as follows:

The second derivative of the Poisson kernel in x-direction is

$$\begin{aligned} \frac{\partial^2}{\partial x^2} \frac{s}{2\pi|\mathbf{x} + s|^3} &= \frac{s}{2\pi} \frac{\partial^2}{\partial x^2} (x^2 + y^2 + s^2)^{-\frac{3}{2}} \\ &= \frac{s}{2\pi} \frac{\partial}{\partial x} \left( -\frac{3}{2} \right) (x^2 + y^2 + s^2)^{-\frac{5}{2}} 2x \\ &= -\frac{3s}{2\pi} \frac{\partial}{\partial x} (x^2 + y^2 + s^2)^{-\frac{5}{2}} x \\ &= -\frac{3s}{2\pi} \left( \left( -\frac{5}{2} \right) (x^2 + y^2 + s^2)^{-\frac{7}{2}} 2x^2 + (x^2 + y^2 + s^2)^{-\frac{5}{2}} \right) \\ &= -\frac{3s}{2\pi} \left( (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5x^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right). \end{aligned} \quad (\text{B.8})$$

And in an analog way follows for the y-direction

$$\frac{\partial^2}{\partial y^2} \frac{s}{2\pi|\mathbf{x} + s|^3} = -\frac{3s}{2\pi} \left( (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5y^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right). \quad (\text{B.9})$$

Combined:

$$\begin{aligned} & \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \frac{s}{2\pi(x^2 + y^2 + s^2)^3} \\ &= -\frac{3s}{2\pi} \left( 2(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5(x^2 + y^2 + s^2)^{-\frac{7}{2}}(x^2 + y^2) \right). \end{aligned} \quad (\text{B.10})$$

## Second Order Riesz Transform with LOP Kernel

So far, the first order Riesz Transform cannot be derived with the LOP as a bandpass filter. Therefore, this and the next paragraph are only concerned with the second and third order Riesz transform. Section 3.1.4 resulted in

$$\mathcal{F}\{R_x\{R_x\}\}(\mathbf{u}) = \mathcal{F}\left\{ \frac{\partial^2}{\partial x^2} \frac{-s}{2\pi|\mathbf{x} + s|^3} \right\}(\mathbf{u}) = 4\pi^2 u^2 (\exp(-2\pi|\mathbf{u}|s)) \quad (\text{B.11})$$

for the second order Riesz transform in x-direction. Therefore

$$\begin{aligned} \frac{\partial^2}{\partial x^2} \frac{-s}{2\pi|\mathbf{x} + s|^3} &= -\frac{s}{2\pi} \frac{\partial}{\partial x} \left( -\frac{3}{2} \right) 2x(x^2 + y^2 + s^2)^{-\frac{5}{2}} \\ &= \frac{3s}{2\pi} \frac{\partial}{\partial x} x(x^2 + y^2 + s^2)^{-\frac{5}{2}} \\ &= \frac{3s}{2\pi} \left( \left( -\frac{5}{2} \right) 2x^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} + (x^2 + y^2 + s^2)^{-\frac{5}{2}} \right) \\ &= \frac{3s}{2\pi} \left( (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5x^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right). \end{aligned} \quad (\text{B.12})$$

In order to derive the  $R_x\{R_y\}$  kernel, one derivation in has to be done in x- and one in y-direction. The idea is the same as in the  $R_x\{R_x\}$  case described in equation 3.13 with the derivative theorem for 2D Fourier transforms [5]:

$$\begin{aligned}
\mathcal{F}\{R_x\{R_y\} * LOP\}(\mathbf{u}) &= \left(-\frac{uv}{|\mathbf{u}|^2}\right) (-4\pi^2|\mathbf{u}|^2) \exp(-2\pi|\mathbf{u}|s) \\
&= uv4\pi^2 \exp(-2\pi|\mathbf{u}|s) \bullet \text{---} \circ \\
\frac{\partial^2}{\partial x \partial y} \frac{-s}{2\pi|\mathbf{x} + s|^3} &= \frac{-s}{2\pi} \frac{\partial^2}{\partial x \partial y} (x^2 + y^2 + s^2)^{-\frac{3}{2}} \\
&= \frac{-s}{2\pi} \frac{\partial}{\partial y} \left(-\frac{3}{2}\right) 2x(x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{3s}{2\pi} \frac{\partial}{\partial y} x(x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{3s}{2\pi} \left(-\frac{5}{2}\right) 2xy(x^2 + y^2 + s^2)^{-\frac{7}{2}} \\
&= -\frac{15s}{2\pi} xy(x^2 + y^2 + s^2)^{-\frac{7}{2}} \tag{B.13}
\end{aligned}$$

The derivation of the  $R_y\{R_y\}$  kernel is the same as the one for the  $R_x\{R_x\}$  kernel, except the second order derivative in the spatial domain, has to be done in y-direction:

$$\begin{aligned}
\mathcal{F}\{R_y\{R_y\} * LOP\}(\mathbf{u}) &= \left(-\frac{v^2}{|\mathbf{u}|^2}\right) (-4\pi^2|\mathbf{u}|^2) \exp(-2\pi|\mathbf{u}|s) \\
&= v^24\pi^2 \exp(-2\pi|\mathbf{u}|s) \bullet \text{---} \circ \\
\frac{\partial^2}{\partial y^2} \frac{-s}{2\pi|\mathbf{x} + s|^3} &= \frac{3s}{2\pi} \left((x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5y^2(x^2 + y^2 + s^2)^{-\frac{7}{2}}\right). \tag{B.14}
\end{aligned}$$

### Third Order Riesz Transform with LOP Kernel

The main idea for the derivation of the third order Riesz transform in x-direction is introduced in 3.1.4. The second derivative theorem for 2D Fourier transforms (see [5]) is applied to the first order Riesz transform in x-direction. All of the other components of the third order Riesz transform are derived analogous, only the directions of the first order Riesz transform and the derivations vary.

$$\begin{aligned}
\mathcal{F}\{R_x\{R_x\{R_x\}\} * LOP\}(\mathbf{u}) &= \left(-i \frac{u^3}{|\mathbf{u}|^3}\right) (-4\pi^2 |\mathbf{u}|^2) \exp(-2\pi |\mathbf{u}|s) \\
&= \frac{u^3}{|\mathbf{u}|} 4\pi^2 \exp(-2\pi |\mathbf{u}|s) \bullet \text{---} \circ \\
\frac{\partial^2}{\partial x^2} \frac{x}{2\pi |\mathbf{x} + s|^3} &= \frac{1}{2\pi} \frac{\partial^2}{\partial x^2} x(x^2 + y^2 + s^2)^{-\frac{3}{2}} \\
&= \frac{1}{2\pi} \frac{\partial}{\partial x} (x^2 + y^2 + s^2)^{-\frac{3}{2}} + \left(-\frac{3}{2}\right) 2x^2 (x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{1}{2\pi} \frac{\partial}{\partial x} (x^2 + y^2 + s^2)^{-\frac{3}{2}} - 3x^2 (x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{1}{2\pi} \left( \left(-\frac{3}{2}\right) 2x (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 6x (x^2 + y^2 + s^2)^{-\frac{5}{2}} \right. \\
&\quad \left. - 6x^3 \left(-\frac{5}{2}\right) (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \\
&= \frac{1}{2\pi} \left( -9x (x^2 + y^2 + s^2)^{-\frac{5}{2}} + 15x^3 (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \\
&= -\frac{3}{2\pi} \left( 3x (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5x^3 (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right). \tag{B.15}
\end{aligned}$$

The next kernel needed is  $R_x\{R_x\{R_y\}\}$ . In order to derive it, one of the derivatives is done in y-direction. The result is

$$\begin{aligned}
\mathcal{F}\{R_x\{R_x\{R_y\}\} * LOP\}(\mathbf{u}) &= \left(-i \frac{u^2 v}{|\mathbf{u}|^3}\right) (-4\pi^2 |\mathbf{u}|^2) \exp(-2\pi |\mathbf{u}|s) \\
&= \frac{u^2 v}{|\mathbf{u}|} 4\pi^2 \exp(-2\pi |\mathbf{u}|s) \bullet \text{---} \circ \\
\frac{\partial^2}{\partial x \partial y} \frac{x}{2\pi |\mathbf{x} + s|^3} &= \frac{1}{2\pi} \frac{\partial^2}{\partial x \partial y} x(x^2 + y^2 + s^2)^{-\frac{3}{2}} \\
&= \frac{1}{2\pi} \frac{\partial}{\partial y} (x^2 + y^2 + s^2)^{-\frac{3}{2}} + \left(-\frac{3}{2}\right) 2x^2 (x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{1}{2\pi} \frac{\partial}{\partial y} (x^2 + y^2 + s^2)^{-\frac{3}{2}} - 3x^2 (x^2 + y^2 + s^2)^{-\frac{5}{2}} \\
&= \frac{1}{2\pi} \left( \left(-\frac{3}{2}\right) 2y (x^2 + y^2 + s^2)^{-\frac{5}{2}} \right. \\
&\quad \left. - \left(-\frac{5}{2}\right) 3x^2 2y (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \\
&= -\frac{1}{2\pi} \left( 3y (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 15x^2 y (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right) \\
&= -\frac{3}{2\pi} \left( y (x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5x^2 y (x^2 + y^2 + s^2)^{-\frac{7}{2}} \right). \tag{B.16}
\end{aligned}$$

The remaining two kernels are not derived explicitly because their derivations are similar to the ones that were just described. The difference is that the single order Riesz

transform used, is the one in y-direction. The derivation of the  $R_x\{R_y\{R_y\}\}$  kernel yields

$$\begin{aligned}
 \mathcal{F}\{R_x\{R_y\{R_y\}\} * LOP\}(\mathbf{u}) &= \left(-i \frac{uv^2}{|\mathbf{u}|^3}\right) (-4\pi^2 |\mathbf{u}|^2) \exp(-2\pi |\mathbf{u}|s) \\
 &= \frac{uv^2}{|\mathbf{u}|} 4\pi^2 \exp(-2\pi |\mathbf{u}|s) \bullet \text{---} \circ \\
 \frac{\partial^2}{\partial x \partial y} \frac{y}{2\pi |\mathbf{x} + s|^3} &= \frac{3}{2\pi} \left( x(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5xy^2(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right).
 \end{aligned}
 \tag{B.17}$$

And finally the  $R_y\{R_y\{R_y\}\}$  is gained by

$$\begin{aligned}
 \mathcal{F}\{R_y\{R_y\{R_y\}\} * LOP\}(\mathbf{u}) &= \left(-i \frac{v^3}{|\mathbf{u}|^3}\right) (-4\pi^2 |\mathbf{u}|^2) \exp(-2\pi |\mathbf{u}|s) \\
 &= \frac{v^3}{|\mathbf{u}|} 4\pi^2 \exp(-2\pi |\mathbf{u}|s) \bullet \text{---} \circ \\
 \frac{\partial^2}{\partial y \partial y} \frac{y}{2\pi |\mathbf{x} + s|^3} &= -\frac{3}{2\pi} \left( 3y(x^2 + y^2 + s^2)^{-\frac{5}{2}} - 5y^3(x^2 + y^2 + s^2)^{-\frac{7}{2}} \right).
 \end{aligned}
 \tag{B.18}$$



# C Implementation

Chapter 3 describes the implementation of both, the Kiel method and the Lübeck method. In order to be able to freely experiment with both methods, a GUI is implemented. In Figure C.1 a screenshot of this GUI is shown.

The GUI allows to load images from file or to create images with one or two overlaid, single or repeated signals. The left part of the GUI features a toolbox that offers a variety of possibilities for image analysis. They are grouped into

- filters for preprocessing,
- computation of the monogenic signal, including local phase, orientation, amplitude, and phase vector estimation,
- computation of the monogenic curvature tensor, including the determinants and traces of the even and odd part,
- local orientation estimation on images with up to two overlaid signals,
- global orientation estimation on images with the possibility to visualize the result using Matlab, and
- rotation estimation.

Input images can be examined by various filters. Available filters for preprocessing the images are for example

- Gaussian filter,
- Laplacian filter,
- Rectangle filter,
- Sobel filter, or
- filters to determine intrinsic dimension using surface theory.

The implementation offers four possibilities to obtain the kernels for the Riesz transform:

- the LOP kernel,
- kernels developed in the frequency domain followed by an application of a inverse Fourier transform (fftw library),
- kernels obtained by successive convolution, and

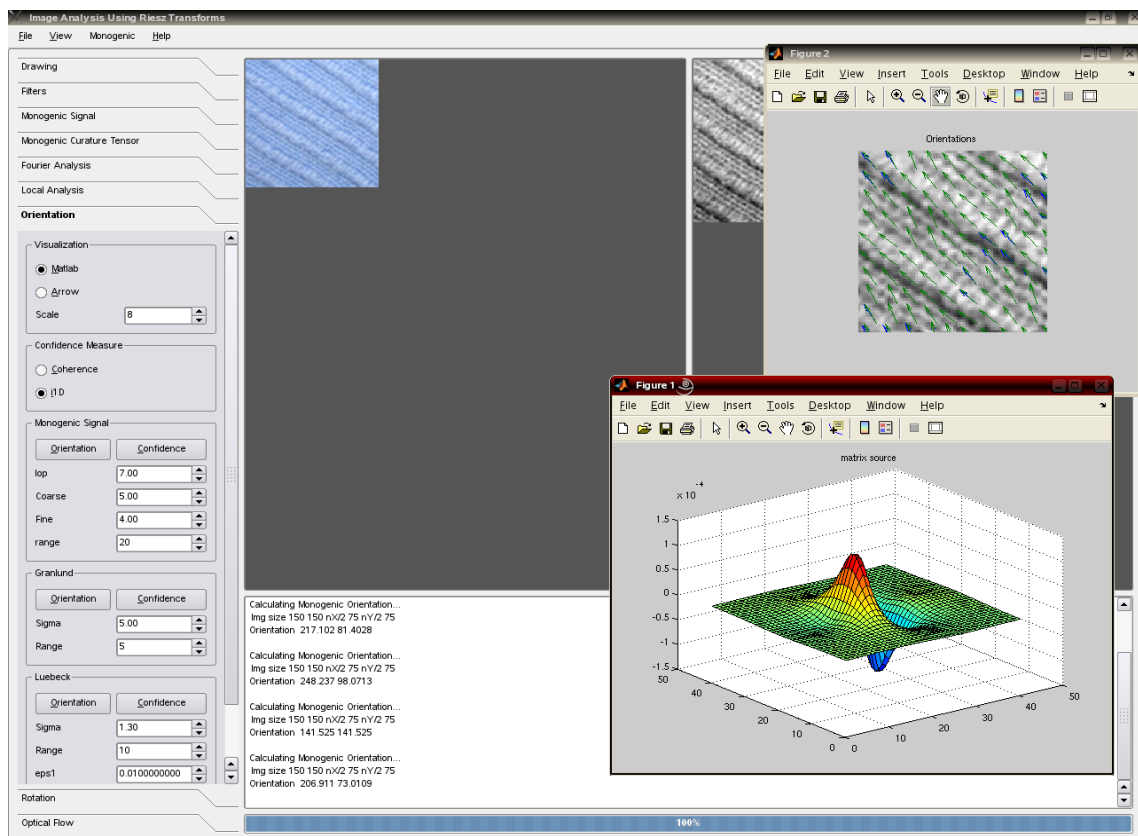


Figure C.1: A snapshot showing the GUI with the result of the orientation estimation and a LOP filter kernel.

- kernels developed in Matlab, which can either be read from files or computed via the Matlab access class.

The filter kernels can be chosen using the menu 'Monogenic' in the menu bar. They are computed using the parameters that are found in the corresponding tab of the toolbox where the parameters are adjusted to the application. If Matlab is available, the resulting kernels can be viewed. The matrices containing the kernel are sent to Matlab via a Matlab access class. They are plotted using the 'surf'-function (see also figure C.1).

For a closer examination the user of the GUI is able to zoom into the images or upsample or downsample them.



# Danksagung

Am Ende dieser Diplomarbeit, in meinem Fall auch am Ende meines Studiums, möchte ich die Gelegenheit nutzen, um mich bei einigen Leuten, die mich ganz besonders unterstützt haben, zu bedanken.

Zuerst möchte ich mich bei *Herrn Prof. Dr. Gerald Sommer* für die Möglichkeit an seinem Lehrstuhl eine Diplomarbeit zu schreiben, betreut zu werden und zudem noch einen Arbeitsraum komplett mit Computer zur Verfügung zu haben, bedanken.

Ein besonderer Dank gilt auch *Herrn Lennart Wietzke*, meinen Betreuer, der mich in vielen Diskussionen, aber auch mit viel Spaß durch diese Zeit begleitet hat.

Außerdem möchte ich mich bei *Carolin Town* und *Andreas Jordt* für das Korrekturlesen bedanken.

Ganz besonders bedanken möchte ich mich bei meinen Eltern *Dr. Klaus* und *Ulrike Sedlazeck*, die mir das Studium nicht nur in finanzieller Hinsicht überhaupt erst ermöglicht haben.

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.

Kiel, den 27. Februar 2008