1

Low-level Feature Detection Using the Boundary Tensor

Ullrich Köthe

Cognitive Systems Group, University of Hamburg,
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
koethe@informatik.uni-hamburg.de

Summary. Tensors are a useful tool for the detection of low-level features such as edges, lines, corners, and junctions because they can represent feature strength and orientation in a way that is easy to work with. However, traditional approaches to define feature tensors have a number of disadvantages. By means of the first and second order *Riesz transforms*, we propose a new approach called the *boundary tensor*. Using quadratic convolution equations, we show that the boundary tensor overcomes some problems of the older tensor definitions. When the Riesz transform is combined with the Laplacian of Gaussian, the boundary tensor can be efficiently computed in the spatial domain. The usefulness of the new method is demonstrated for a number of application examples.[1]

1.1 Introduction

Even when the raw image data are not tensor-valued, tensor-based methods have been found useful in image analysis because tensors describe local image properties in a way that is invariant under Euclidean transformations of the space. The two main applications so far are feature extraction and optical flow computation. Historically, the latter one has been investigated first. The optical flow problem can be formulated as the task of finding the main local orientation at every point of the 3-dimensional spatio-temporal domain that is formed by interpreting an image sequence as a 3-dimensional data set with two spatial and one temporal dimensions. One can then define the spatio-temporal gradient of the sequence f_3 as:

$$\nabla f_3 = \left(\frac{\partial f_3}{\partial x}, \frac{\partial f_3}{\partial y}, \frac{\partial f_3}{\partial t} \right)^T \tag{1.1}$$

[1] This work was performed during a visit at the Computer Vision Lab of the University of Linköping, Sweden. I'd like to thank G. Granlund, M. Felsberg and K. Nordberg for many valuable discussions, and the Informatics Department of the University of Hamburg for their generous support of this visit.

Under the assumption of constant optical flow in a neighborhood of the current point, the flow vector $\mathbf{v}$ can be determined from the null-space of the *structure tensor* $\mathbf{S}_3$ [2, 11], cf. chapters 2 by Brox et al. and 3 by Nagel in this volume:

$$\mathbf{S}_3\mathbf{v} = 0 \qquad \text{with} \qquad \mathbf{S}_3 = g_\sigma \star (\nabla f_3 \nabla f_3^T) \tag{1.2}$$

The structure tensor is the averaged outer product of the spatio-temporal gradient with itself, where the averaging filter g_σ (usually a Gaussian) is chosen according to the size of the neighborhood where the flow is assumed constant. The flow vector is only uniquely determined if the null space of the 3-dimensional structure tensor is 1-dimensional, i.e. if the structure tensor has rank 2. If it has lower rank, there is no unique flow vector, which is known as the *aperture problem*. This problem naturally leads to the definition of the 2D structure tensor as the averaged outer product of just the spatial gradient:

$$\mathbf{S}_2 = g_\sigma \star (\nabla f_2 \nabla f_2^T) \qquad \text{with} \qquad \nabla f_2 = \left(\frac{\partial f_2}{\partial x}, \frac{\partial f_2}{\partial y}\right)^T \tag{1.3}$$

This 2-dimensional tensor must have full rank for a unique flow vector to exist, which is the case if the local image structure is neither flat (as in homogeneous regions) nor 1-dimensional (as at edges), but has high variation in all directions. Points of maximal variation are called *spatial interest points* and correspond to important structural features such as gray level corners, junctions, and extrema. They can for example be found as the local maxima of the corner strength measures proposed by *Förstner* [6] and *Harris* [8]:

$$c_{\text{Förstner}} = \frac{\det(\mathbf{S}_2)}{\operatorname{tr}(\mathbf{S}_2)} \qquad c_{\text{Harris}} = \det(\mathbf{S}_2) - \kappa \operatorname{tr}^2(\mathbf{S}_2) \tag{1.4}$$

where κ is usually set to 0.04. In addition, Förstner [6] and Nagel [11] used the structure tensor to define a contrast independent measure of local isotropy:

$$c_{\text{roundness}} = \frac{4\det(\mathbf{S}_2)}{\left(\operatorname{tr}(\mathbf{S}_2)\right)^2} \tag{1.5}$$

A completely different approach to tensor-based feature detection was proposed by Granlund and Knutsson [9]. They were interested in the characterization of locally 1-dimensional image structures, i.e. edges and lines, which they call *simple structures*. Formally, simple structures are defined by the fact that the image is locally reduced to a 1-dimensional function that varies only along a certain direction $\mathbf{n}$ and is constant perpendicular to that direction:

$$f_2(\mathbf{x}) \approx f_1(\mathbf{x}^T\mathbf{n}) \tag{1.6}$$

Then the local signal energy and orientation can be represented by an *orientation tensor* as

$$\mathbf{T} = \lambda \mathbf{n}\mathbf{n}^T \tag{1.7}$$

Since in [9] the authors are interested in arbitrary 1-dimensional features, the estimation procedure for $\mathbf{T}$ must react uniformely to edges and lines. This property is called *phase invariance* because edges and lines can be understood as superpositions of trigonometric (complex exponential) basis functions at different phase (namely phase 0 or π for lines and $\pm\pi/2$ for edges). Phase invariance can be achieved by estimating the tensor with oriented quadrature filters [9] or with a local polynomial approximation [4]. *Quadrature filter pairs* were originally invented to estimate the instantaneous energy and phase of a 1-dimensional signal. A quadrature pair $(h_{\text{even}}, h_{\text{odd}})$ consists of an even and an odd symmetric filter, and the instantaneous (edge or line) energy can be calculated as the sum of squares of the filter responses:

$$E(x) = (h_{\text{even}} \star f_1)^2 + (h_{\text{odd}} \star f_1)^2 \tag{1.8}$$

To actually form a quadrature pair, the filters must be related by the Hilbert transform $\mathcal{H}$, which is defined in the Fourier domain by

$$H_{\text{odd}}(u) = \mathcal{H}[H_{\text{even}}(u)] = j\frac{u}{|u|}H_{\text{even}}(u) = j\,\text{sign}(u)H_{\text{even}}(u) \tag{1.9}$$

(slanted capitals denote the Fourier transforms of the corresponding lowercase functions). To apply these filters in 2D, it is conventional to rotate them into some orientation of interest. In order to estimate $\mathbf{T}$ on a 2D image, at least 3 orientations are necessary [9]. When the local image structure is indeed 1-dimensional and the orientations $\theta_i = [0, \pi/3, 2\pi/3]$ are used, we get

$$\mathbf{T} = \sum_i (\mathbf{m}_i\mathbf{m}_i^T - \mathbf{I}/4)\, E_i \tag{1.10}$$

where E_i is the energy computed for orientation i, $\mathbf{m}_i = (\cos\theta_i, \sin\theta_i)^T$ and $\mathbf{I}$ is the unit tensor. A second order *polynomial approximation* of the image structure around $\mathbf{x}_0$ is defined by the local model

$$f_{\text{model}}(\mathbf{x}_0 + \mathbf{x}) = c + \mathbf{x}^T\mathbf{b} + \mathbf{x}^T\mathbf{A}\mathbf{x} \tag{1.11}$$

An in-depth discussion of how to estimate $\mathbf{A}, \mathbf{b}, c$ can be found in [4]. Possibilities include local polynomial fits, facet models, moment filters, and Gaussian derivative filters. The orientation tensor is then defined as

$$\mathbf{T} = \mathbf{A}\mathbf{A}^T + \gamma\mathbf{b}\mathbf{b}^T \tag{1.12}$$

However, with the common estimation methods for $\mathbf{A}$ and $\mathbf{b}$ this tensor is only phase invariant for a single frequency determined by γ, which is therefore considered as an algorithm tuning parameter.

The existing methods have a number of shortcomings. The structure tensor approach is not phase invariant, because, being based on the image gradient, it reacts differently to edges and lines. Furthermore, due to averaging over a

neighborhood, nearby features (e.g. the corners of a small triangle) will blend into only a single response and cannot be resolved separately. In the quadrature filter approach, definite statements about the properties of $\mathbf{T}$ can only be made if the local image structure is indeed 1-dimensional. It is unclear exactly what happens at 2-dimensional configurations. Finally, when the tensor is based on a polynomial approximation, the choice of the parameter γ is problematic. Usually it is impossible to find a single γ that works well on the entire image, and a procedure to choose it locally is not known. Consequently, the response is not phase invariant at most locations, and multiple responses near a single line are common.

In this contribution I am discussing the *boundary tensor* introduced in [10] as a method designed to overcome these shortcomings. It will be based on a new generalization of quadrature filters to 2 dimensions using the *Riesz transform*. The boundary tensor will turn out to be structurally equivalent to the polymial-based tensor definition, but with a uniquely determined parameter $\gamma = 1$. It will exhibit phase invariance for all frequencies in the same way as the quadrature filter approach. By analysing the new method in the framework of quadratic convolution, we can also show that it reacts in a useful way to locally 2-dimensional configurations. An efficient spatial domain algorithm and a number of feature analysis examples conclude the paper.

1.2 The Boundary Tensor

Before we go on to define the boundary tensor, we recall that (Cartesian) tensors are in general characterized by the fact that the tensor elements in a rotated coordinate system can be calculated as linear combinations of the tensor elements in the original coordinate system (cf. chapter 1 in this book):

$$\tilde{T}_{i_1 \ldots i_p} = \sum_{l_1=1}^{N} \cdots \sum_{l_p=1}^{N} r_{i_1 l_1} \ldots r_{i_p l_p} T_{l_1 \ldots l_p} \tag{1.13}$$

where $T_{l_1 \ldots l_p}$ are the elements of a p^{th}-order tensor, and r_{il} are the elements of the N-dimensional rotation matrix. These transformation rules ensure that the properties represented by the tensor as a whole remain invariant under Euclidean transformations of the space, even when the individual tensor elements do not. New tensors can be created from existing ones by linear combinations, by means of the Cartesian (outer) product and by contraction. A tensor of order zero is a rotationally invariant scalar. Therefore, we can interpret every pixel of the original image or an image obtained by a rotationally symmetric filter as a 0^{th}-order tensor.

We can define a tensor-based generalization of quadrature filtering by replacing the 1-dimensional Hilbert transform with the N-dimensional *Riesz transform* [5] which is defined as:

$$\text{Fourier domain:} \quad \mathcal{H}_N[H(\mathbf{u})] = j\frac{\mathbf{u}}{|\mathbf{u}|}H(\mathbf{u}) \tag{1.14}$$

$$\text{spatial domain:} \quad \mathcal{H}_N[h(\mathbf{x})] = \frac{\Gamma((N+1)/2)}{\pi^{(N+1)/2}}\left(\frac{-\mathbf{x}}{|\mathbf{x}|^{N+1}} \star h(\mathbf{x})\right) \tag{1.15}$$

where Γ is the gamma function. In essence, the scalar-valued frequency coordinate u of the Hilbert transform is simply replaced by an N-dimensional frequency vector $\mathbf{u}$. The Riesz transform can be interpreted as a first-order tensor operator because it turns a scalar valued function into a first order tensor-valued one. This can be easily seen by observing that the ratio $\frac{\mathbf{u}}{|\mathbf{u}|}$ defines the first order spherical harmonics (i.e. $(\cos\theta, \sin\theta)^T$ in 2D, $(\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi)^T$ in 3D etc.), and polar separable functions with this angular behavior conform exactly to (1.13) with $p = 1$. Spherical harmonics are preserved by inverse Fourier transformation, so that the spatial domain version (1.15) of the Riesz transform has the same angular behavior and the tensor requirements are still satisfied. The Riesz transform is closely related to the gradient and acts in a qualitatively similar way, as can be seen by defining the latter in terms of the former:

$$\nabla_N h(\mathbf{x}) \quad \circ\!\!-\!\!\bullet \quad \mathcal{H}_N[\,|\mathbf{u}|\,H(\mathbf{u})] \tag{1.16}$$

where $\circ\!\!-\!\!\bullet$ denotes Fourier correspondence. Both operators have the same angular behavior, but the gradient in addition changes the radial part of the spectrum. This difference is of crucial importance for the definition of phase-invariant operators. Another important observation concerns the difference between the 1-dimensional Hilbert transform and the multi-dimensional Riesz transform: while applying the former transform twice just reproduces the original signal (with reversed sign), multiple applications of the Riesz transform create tensors of higher and higher orders. This is again similar to the gradient operator, where twofold application results in the Hessian matrix etc.

However, applying the Riesz transform to the original image makes little sense in practice, because its spatial domain kernel decreases only as $|\mathbf{x}|^{-N}$, so that feature localization would be bad. Instead, one combines it with a radially symmetric band-pass. In contrast to derivative filters, where the band-pass changes with the derivative order, the band-pass is kept the same for all orders of the Riesz transform. We define the first and second order band-pass Riesz transforms $\mathbf{b}$ and $\mathbf{A}$ of an image F in the Fourier domain as

$$\mathbf{b} \quad \circ\!\!-\!\!\bullet \quad \mathcal{H}_N[K(|\mathbf{u}|)F(\mathbf{u})] = j\frac{\mathbf{u}}{|\mathbf{u}|}K(|\mathbf{u}|)F(\mathbf{u}) \tag{1.17}$$

$$\mathbf{A} \quad \circ\!\!-\!\!\bullet \quad \mathcal{H}_N^2[K(|\mathbf{u}|)F(\mathbf{u})] = -\frac{\mathbf{u}\mathbf{u}^T}{|\mathbf{u}|^2}K(|\mathbf{u}|)F(\mathbf{u}) \tag{1.18}$$

where $K(|\mathbf{u}|)$ is the band-pass. It should be noted that these definitions are valid for all dimensions $N \geq 2$. The boundary tensor is now defined as

$$\mathbf{B} = \mathbf{b}\mathbf{b}^T + \mathbf{A}\mathbf{A}^T \tag{1.19}$$

This definition is structurally equivalent to (1.12), but the parameter γ is no longer needed, because the boundary tensor is phase invariant for all frequencies (see below). Since $\mathbf{A}$ and $\mathbf{b}$ are both real, it follows that $\mathbf{B}$ is always positive semi-definite. Therefore, the trace of the tensor can be interpreted as a measure of local signal energy, which will be called *boundary energy*. The choice of this name stems from the fact that the tensor indeed detects important boundary features, as is shown below.

1.3 Analysis of the Boundary Tensor as a Quadratic Filter

In order to analyse the properties of the boundary tensor, we follow the proposal of [12] and formulate the tensor as a quadratic filter [13]. Quadratic convolution is defined as

$$\tilde{f}(\mathbf{x}) = \iint h(\mathbf{x}-\mathbf{x_1}, \mathbf{x}-\mathbf{x_2}) f(\mathbf{x_1}) f(\mathbf{x_2})\, d\mathbf{x_1}\, d\mathbf{x_2} \tag{1.20}$$

where $h(.,.)$ is the kernel, and the method is termed "quadratic because the original image f appears twice in the integral. Let $g_i(\mathbf{x})$ denote the i^{th} component ($i = 1...N$) of the first order band-pass Riesz transform kernel. Then

$$\begin{aligned}(\mathbf{b}\mathbf{b}^T)_{il} = \mathbf{b}_i\mathbf{b}_l &= (g_i \star f)(g_l \star f) \\ &= \int g_i(\mathbf{x}-\mathbf{x_1}) f(\mathbf{x_1})\, d\mathbf{x_1} \int g_l(\mathbf{x}-\mathbf{x_2}) f(\mathbf{x_2})\, d\mathbf{x_2} \\ &= \iint g_i(\mathbf{x}-\mathbf{x_1}) g_l(\mathbf{x}-\mathbf{x_2}) f(\mathbf{x_1}) f(\mathbf{x_2})\, d\mathbf{x_1}\, d\mathbf{x_2}\end{aligned} \tag{1.21}$$

Similarly, let $g_{il}(\mathbf{x})$ represent component il ($i, l = 1...N$) of the kernel for the second order band-pass Riesz transform. This leads to

$$\begin{aligned}(\mathbf{A}\mathbf{A}^T)_{il} = \sum_k \mathbf{A}_{ik}\mathbf{A}_{kl} &= \sum_k (g_{ik} \star f)(g_{kl} \star f) \\ &= \iint \left(\sum_k g_{ik}(\mathbf{x}-\mathbf{x_1}) g_{kl}(\mathbf{x}-\mathbf{x_2}) \right) f(\mathbf{x_1}) f(\mathbf{x_2})\, d\mathbf{x_1}\, d\mathbf{x_2}\end{aligned} \tag{1.22}$$

We can combine both equations into a single quadratic convolution with

$$h_{il}(\mathbf{x_1}, \mathbf{x_2}) = g_i(\mathbf{x_1}) g_l(\mathbf{x_2}) + \sum_k g_{ik}(\mathbf{x_1}) g_{kl}(\mathbf{x_2}) \tag{1.23}$$

Then the components of the boundary tensor can be written as

$$\text{spatial domain: } \mathbf{B}_{il}(\mathbf{x}) = \iint h_{il}(\mathbf{x}-\mathbf{x_1},\mathbf{x}-\mathbf{x_2}) f(\mathbf{x_1}) f(\mathbf{x_2})\, d\mathbf{x_1}\, d\mathbf{x_2}$$

$$\text{Fourier domain: } \mathbf{B}_{il}(\mathbf{x}) \circ\!\!-\!\!\bullet \iint H_{il}(\mathbf{u},\mathbf{v}) F(\mathbf{u}) F(\mathbf{v}) e^{j(\mathbf{u}+\mathbf{v})^T\mathbf{x}}\, d\mathbf{u}\, d\mathbf{v} \quad (1.24)$$

where F is the N-dimensional Fourier transform of f, H_{il} is the $2N$-dimensional Fourier transform of h_{il}, and $e^{j(\mathbf{u}+\mathbf{v})^T\mathbf{x}}$ translates f so that the current point $\mathbf{x}$ becomes the origin. Inserting the Fourier representation of the band-pass Riesz transform, H_{il} gets a simple functional form:

$$\begin{aligned} H_{il}(\mathbf{u},\mathbf{v}) &= -\frac{\mathbf{u}_i}{|\mathbf{u}|}\frac{\mathbf{v}_l}{|\mathbf{v}|} K(|\mathbf{u}|)K(|\mathbf{v}|) + \sum_k \left(\frac{\mathbf{u}_i\mathbf{u}_k}{|\mathbf{u}|^2}\frac{\mathbf{v}_k\mathbf{v}_l}{|\mathbf{v}|^2}\right) K(|\mathbf{u}|)K(|\mathbf{v}|) \\ &= \frac{\mathbf{u}_i\mathbf{v}_l}{|\mathbf{u}||\mathbf{v}|}\left(-1+\frac{\mathbf{u}^T\mathbf{v}}{|\mathbf{u}||\mathbf{v}|}\right) K(|\mathbf{u}|)K(|\mathbf{v}|) \end{aligned} \quad (1.25)$$

In order for $\mathbf{B}_{il}$ to be real for real images f, it is required that $H_{il}(-\mathbf{u},-\mathbf{v}) = \overline{H_{il}(\mathbf{u},\mathbf{v})}$, which is easily verified. Conditions for an N-dimensional tensor operator to behave like a 1-D quadrature filter for simple images (i.e. images where $f(\mathbf{x}) = \hat{f}(\mathbf{x}^T\mathbf{n})$ for some unit vector $\mathbf{n}$ giving the signal orientation) are derived in [12]. If the signal is simple, the spectrum of $F(\mathbf{u})$ vanishes for all $\mathbf{u} \neq t\mathbf{n}$, and the restriction of the kernel to this line must reduce to

$$H_{il}(t\mathbf{n},\tau\mathbf{n}) = \mathbf{n}_i\mathbf{n}_l\hat{H}(t,\tau) \quad (1.26)$$

Furthermore, in order for the signal energy to be phase invariant $\hat{H}(t,t) = 0$ must hold for all t. Both conditions are fulfilled, because $\mathbf{u}_i = t\mathbf{n}_i$ and $\mathbf{v}_i = \tau\mathbf{n}_i$, and thus

$$\begin{aligned} H_{il}(t\mathbf{n},\tau\mathbf{n}) &= \frac{t\tau\mathbf{n}_i\mathbf{n}_l}{|t||\tau|}\left(-1+\frac{t\tau\mathbf{n}^T\mathbf{n}}{|t||\tau|})\right) K(|t|)K(|\tau|) \\ &= \mathbf{n}_i\mathbf{n}_l\,(-\operatorname{sign}(t\tau)+1)\,K(|t|)K(|\tau|) \end{aligned} \quad (1.27)$$

In fact, this is precisely 4 times the expression which [12] derived for the quadrature filter method according to [9], cf. 1.10), so that both approaches behave identically for simple signals. For simple signals the signal energy $\operatorname{tr}(\mathbf{B}) = \sum_k \mathbf{B}_{kk}$ reduces exactly to the 1-dimensional quadrature energy (1.8):

$$\begin{aligned} \operatorname{tr}(\mathbf{B}) &= \sum_k \iint \mathbf{n}_k^2(-\operatorname{sign}(t\tau)+1)K(|t|)K(|\tau|)\hat{F}(t\mathbf{n})\hat{F}(\tau\mathbf{n})\, e^{j(t+\tau)\mathbf{n}^T\mathbf{x}}\, dt\, d\tau \\ &= \left(\mathcal{H}_1[k_1] \star \hat{f}\right)^2 + \left(k_1 \star \hat{f}\right)^2 \end{aligned} \quad (1.28)$$

where k_1 is the 1-dimensional inverse Fourier transform of K and $\mathcal{H}_1[k_1]$ its Hilbert transform (derivation see appendix).

H_{il} is also a useful tool to analyse the behavior of the boundary tensor for intrinsically 2-dimensional features. To simplify matters, we consider points

$\mathbf{x}$ where the spectrum $F(\mathbf{u})$ computed with $\mathbf{x}$ as coordinate origin is polar separable within the pass-band of the tensor filter $K(|\mathbf{u}|)$. At many structures of interest this is at least approximately true. Then the product $K(|\mathbf{u}|)F(\mathbf{u})$ can be written as $K(|\mathbf{u}|)F_r(|\mathbf{u}|)F_a(\phi)$. After inserting this and (1.25) into the boundary tensor expression (1.24), the latter can be transformed into polar coordinates and simplifies into a product of two integrals:

$$\begin{aligned}
\mathbf{B}_{il} &= \iint \frac{\mathbf{u}_i \mathbf{v}_l}{|\mathbf{u}||\mathbf{v}|} \left(-1 + \frac{\mathbf{u}^T\mathbf{v}}{|\mathbf{u}||\mathbf{v}|}\right) K(|\mathbf{u}|)K(|\mathbf{v}|)F(\mathbf{u})F(\mathbf{v})\, d\mathbf{u}\, d\mathbf{v} \\
&= \left(\iint \mathbf{n}_i(\phi)\mathbf{n}_l(\psi)\left(-1 + \mathbf{n}^T(\phi)\,\mathbf{n}(\psi)\right) F_a(\phi)F_a(\psi)\, d\phi\, d\psi\right) \\
&\quad \left(\iint K(\rho_1)K(\rho_2)F_r(\rho_1)F_r(\rho_2)\,\rho_1\, d\rho_1\,\rho_2\, d\rho_2\right) \qquad (1.29) \\
&= \mathbf{B}_{a,il} tensor B_r
\end{aligned}$$

with $\mathbf{u} = \rho_1\mathbf{n}(\phi) = \rho_1(\cos(\phi), \sin(\phi))^T$ and $\mathbf{v} = \rho_2\mathbf{n}(\psi) = \rho_2(\cos(\psi), \sin(\psi))^T$. It should be noted that this is a major advantage of using the Riesz transform: Otherwise, the first and second order filter kernels would have had different radial parts, and the separation of angular and radial behavior were impossible. The angular integral $\mathbf{B}_{a,il}$ in (1.29) can be further simplified in terms of the Fourier coefficients of F_a:

$$\alpha_n = \int \cos(n\phi)F_a(\phi)\, d\phi \qquad \beta_n = \int \sin(n\phi)F_a(\phi)\, d\phi \qquad (1.30)$$

It turns out that only the Fourier coefficients up to second order are relevant (the others drop out due to orthogonality of trigonometric functions), and the boundary tensor components can be written as (see appendix):

$$\begin{aligned}
\mathbf{B}_{11} &= (\alpha_1^2 + \tfrac{1}{4}(\alpha_0 + \alpha_2)^2 + \tfrac{1}{4}\beta_2^2)\,\mathbf{B}_r \\
\mathbf{B}_{22} &= (\beta_1^2 + \tfrac{1}{4}(\alpha_0 - \alpha_2)^2 + \tfrac{1}{4}\beta_2^2)\,\mathbf{B}_r \qquad (1.31) \\
\mathbf{B}_{12} &= (\alpha_1\beta_1 + \tfrac{1}{2}\alpha_0\beta_2)\,\mathbf{B}_r
\end{aligned}$$

where $\mathbf{B}_r$ is the radial part of (1.29). These equations give us a qualitative understanding of how the boundary tensor reacts to 2D features: At (approximately) polar separable locations, its components are products of radial and angular expressions. The former measure the contrast of the local structure at the scale of the bandpass filter, and the latter determine how well the angular shape can be represented with circular harmonics up to order 2. Since many important structures (edges, lines, saddles, corners) are covered by this model, the boundary tensor reacts reasonably at many locations where for example the gradient (which solely relies on first-order circular harmonics) fails. We illustrate this with two examples: parameterized step and line edges. In the spatial domain the angular parts of these features can be written as

$$f_{a,\mathrm{edge}}(\phi) = \Theta(\phi+\phi_0) - \Theta(\phi-\phi_0) \qquad f_{a,\mathrm{line}}(\phi) = \delta(\phi+\phi_0) - \delta(\phi-\phi_0) \quad (1.32)$$

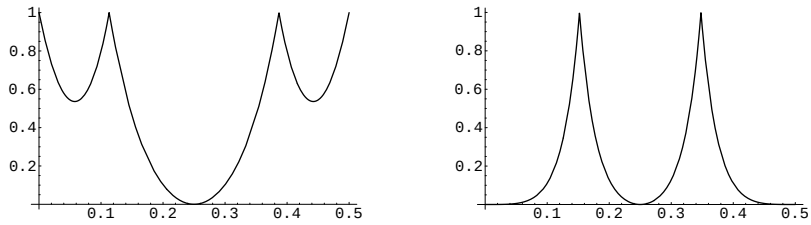

Fig. 1.1. Eigenvalue ratios $\mu = \min(\mathbf{B}_{11}, \mathbf{B}_{22}) / \max(\mathbf{B}_{11}, \mathbf{B}_{22})$ for a parameterized edge (left) and line (right) as a function of $l_0 = \phi_0/(2\pi)$.

where Θ is the step function and δ the impulse function. The parameter $\phi_0 \in [0, \pi]$ determines the angle of the corner, and $\phi_0 = \pi/2$ results in a straight edge or line. Due to symmetry, $\mathbf{B}_{12}$ is always zero at the center of these configurations, and $\mathbf{B}_{11}$, $\mathbf{B}_{22}$ are the tensor eigenvalues. The ratio of the eigenvalues is a measure that distinguishes locally 1-dimensional and 2-dimensional configurations – it is near 0 in the former and near 1 in the latter case. Fig. 1.1 shows these ratios as a function of ϕ_0. It can be seen that we get indeed $\mu = 0$ for $\phi_0 = \pi/2$ (straight edge/line) and $\mu = 1$ for $\phi_0 = \pi/4$ and $\phi_0 = 3\pi/4$ (90 degree corners). For $\phi_0 = 0$ the edge disappears, and $\mu = 1$ indicates that the remaining homogenuous region is interpreted as a 2D configuration, whereas in case of the line, $\phi_0 = 0$ results in a half-line which the boundary tensor cannot distinguish from a straight line, hence $\mu = 0$. Since more complex junction configurations can be expressed as combinations of multiple edge and/or line corners, they can be analysed in essentially the same way. All junctions that can be approximated well with an angular second-order Fourier series (e.g. saddle points, crossings of two straight lines) will be characterized correctly by the boundary tensor.

1.4 Efficient Computation of the Boundary Tensor

In order to compute the boundary tensor in practice we have to choose a suitable band-pass K. Filters based on Gaussian or exponential transfer functions and log-normal filters are obvious choices. If implemented in the Fourier domain, all these filters are equally easy to compute. However, we have only been able to find an efficient *spatial* domain implementation of the boundary tensor (or actually a close approximation of it) if the band-pass equals the magnitude of the Laplacian of Gaussian

$$K(|\mathbf{u}|, \sigma) = |\mathbf{u}|^2 e^{-\frac{|\mathbf{u}|^2 \sigma^2}{2}} \tag{1.33}$$

Moreover, we have experimentally found that this band-pass gives better feature resolution (less blending of nearby features into each other) than other choices, which is probably due to the Gaussian's optimal localization in both

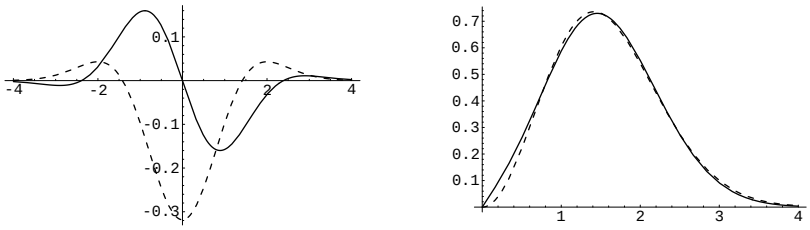

Fig. 1.2. left: g_1 (solid) and g_{11} (dashed) along the $\mathbf{x}_1$ axis when the band-pass is the Laplacian of Gaussian at $\sigma = 1$ (g_1 and its approximation $\tilde{g}_1$ according to (1.39) and (1.41) are almost indistinguishable in the depicted 4σ interval). Right: spectra of K (dashed) and its approximation $\tilde{G}_1 = \mathcal{F}[\tilde{g}_1]$ (solid) at $\sigma = 1$.

the spatial and frequency domains. The spectra of the filters are the product of the bandpass with first and second order Riesz transforms:

$$G_i(\mathbf{u}) = j\mathbf{u}_i|\mathbf{u}|e^{-\frac{|\mathbf{u}|^2\sigma^2}{2}} \tag{1.34}$$

$$G_{il}(\mathbf{x}) = -\mathbf{u}_i\mathbf{u}_l e^{-\frac{|\mathbf{u}|^2\sigma^2}{2}} \tag{1.35}$$

It can be seen that the second order spectra are exactly those of the second derivative of the Gaussian. Therefore, the spatial filter function is:

$$g_{il}(\mathbf{x}) = \frac{\mathbf{x}_i\mathbf{x}_l - 2\sigma^2\delta_{il}}{2\pi\sigma^6}\, e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}} \tag{1.36}$$

and the resulting tensor $\mathbf{A}$ is the Hessian of Gaussian, which can be efficiently computed by separable convolutions. Inverse Fourier transform of the first order filters is more complicated (see appendix). The result is

$$g_i(\mathbf{x}) = \frac{\mathbf{x}_i}{4\sqrt{2\pi}\sigma^7}\, e^{-\frac{|\mathbf{x}|^2}{4\sigma^2}} \left((|\mathbf{x}|^2 - 3\sigma^2) I_0\left(\frac{|\mathbf{x}|^2}{4\sigma^2}\right) - (|\mathbf{x}|^2 - \sigma^2) I_1\left(\frac{|\mathbf{x}|^2}{4\sigma^2}\right)\right) \tag{1.37}$$

where I_0 and I_1 are modified Bessel functions of the first kind. Fig. 1.2 left depicts the shape of g_1 and g_{11} along the $\mathbf{x}_1$ axis. Unfortunately, the first order kernels are unsuitable for practical applications because their asymptotic decay is only $\mathcal{O}(|\mathbf{x}|^{-4})$ and they are not Cartesian separable. This means that large 2-dimensional filter masks are needed, which makes computation of g_i very slow. Therefore, we apply a design technique similar to the one used for steerable quadrature filters [7] to approximate g_i with filters $\tilde{g}_i$ that can be computed separably and decay exponentially. The idea is to realize $\tilde{g}_i$ as sums of filters that are third order polynomials times a Gaussian. The polynomials-times-Gaussian are defined so that they together form a supersymmetric third order tensor filter $\tilde{g}_{ijk}$ (a supersymmetric tensor has the property that its components don't change under permutation of indices, i.e. $\tilde{g}_{112} = \tilde{g}_{121} = \tilde{g}_{211}$ etc.). Then the first order tensor filter can be obtained by contraction over any pair of indices, i.e. $\tilde{g}_i = \sum_k \tilde{g}_{ikk}$. We make the following ansatz:

$$\begin{aligned}
\tilde{g}_{iii}(\mathbf{x},\sigma') &= \left(\frac{a\mathbf{x}_i^3}{\sigma'^5} + \frac{b\mathbf{x}_i}{\sigma'^3}\right) g(\mathbf{x},\sigma') \\
\tilde{g}_{ill}(\mathbf{x},\sigma') &= \frac{\mathbf{x}_i}{\sigma'^2}\left(\frac{a\mathbf{x}_l^2}{\sigma'^3} + \frac{b}{3\sigma'}\right) g(\mathbf{x},\sigma') \qquad (i \neq l) \\
\tilde{g}_{ilk}(\mathbf{x},\sigma') &= \frac{a}{\sigma'^5}\,\mathbf{x}_i\mathbf{x}_l\mathbf{x}_k\; g(\mathbf{x},\sigma') \qquad (i \neq l \neq k)
\end{aligned} \tag{1.38}$$

where $g(\mathbf{x},\sigma')$ is an N-dimensional Gaussian, and the last function is only required if $N > 2$ (for $N = 2$, the condition $i \neq l \neq k$ is never satisfied). By expressing these functions in a rotated coordinate system, it is easy (if tedious) to verify that (1.13) is fulfilled with $p = 3$. The spectrum of $\tilde{g}_i$ is

$$\tilde{g}_i = \sum_k \tilde{g}_{ikk} \;\circ\!\!-\!\!\bullet\; \tilde{G}_i(\mathbf{u},\sigma') = \frac{\mathbf{u}_i}{\sigma'}\left(a(4 - |\mathbf{u}|^2\sigma'^2) + \frac{4b}{3}\right) e^{-|\mathbf{u}|^2\sigma'^2/2} \tag{1.39}$$

We now formulate a least squares problem to choose a, b, σ' so that the radial part of $\tilde{G}_i(\mathbf{u},\sigma')$ becomes as similar to $K(\mathbf{u},\sigma)$ as possible:

$$\text{minimize w.r.t. } a,b,\sigma': \quad \int \left(\tilde{G}(|\mathbf{u}|,\sigma') - K(|\mathbf{u}|,\sigma)\right)^2 d\mathbf{u} \tag{1.40}$$

where $\tilde{G}(|\mathbf{u}|,\sigma')$ is obtained from $\tilde{G}_i(\mathbf{u},\sigma')$ by replacing $\mathbf{u}_i$ with $|\mathbf{u}|$. The optimum depends on the dimension N of the space. For $N = 2, 3$ we get

$$\begin{aligned}
a_{2D} &= -0.5589, \qquad b_{2D} = 2.0425, \qquad \sigma'_{2D} = 1.0818\,\sigma \\
a_{3D} &= -0.5086, \qquad b_{3D} = 1.8562, \qquad \sigma'_{3D} = 1.0683\,\sigma
\end{aligned} \tag{1.41}$$

Fig. 1.2 right depicts $\tilde{G}$ and K for the 2D case. It should be noted that it is important to include the filter scale in the optimization because this significantly improves the fit. To conclude, we can compute the boundary tensor by using 7 separable, exponentially decaying filters in 2D, and 15 such filters in 3D. This can be compared with the structure tensor, where N filters are used to compute the gradients, but then $N(N+1)/2$ filters at a larger (typically doubled) scale are applied to integrate the gradient tensors over a neighborhood. Thus, the number of filters is lower, but larger windows are required, making the overall computational effort about equal.

1.5 Applications

The boundary tensor can be used much like the structure tensor, e.g. as an integrated detector for low-level image features such as edges, lines, corners, and junctions (in 3D additionally surfaces), and to estimate local orientation. For feature analysis it is advantageous to consider the eigensystem of the boundary tensor. Using the eigenvalues $\mu_1 \geq \mu_2 \geq 0$ and the eigenvector

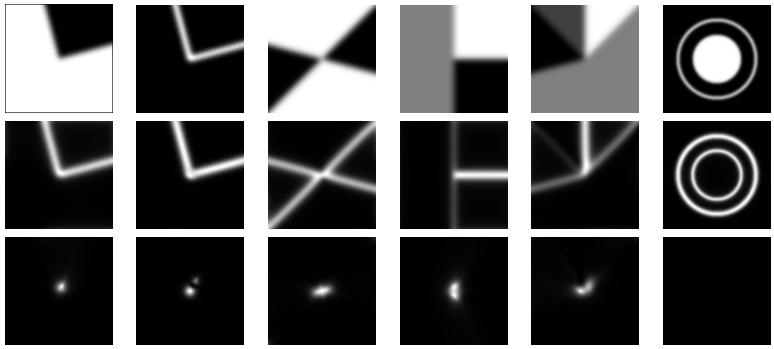

Fig. 1.3. Top: test patterns, center: boundary energy $\mathrm{tr}(\mathbf{B})$, bottom: junction energy $\mathrm{tr}(\mathbf{B}_{\mathrm{junction}})$

$\mathbf{n}$ corresponding to μ_1 (cf. chapter 1 by Hagen and Garth), one can decompose the boundary tensor into its 1D and 2D (edge/line and corner/junction) contributions:

$$\mathbf{B} = \mathbf{B}_{\mathrm{edge}} + \mathbf{B}_{\mathrm{junction}} = (\mu_1 - \mu_2)\mathbf{n}\mathbf{n}^T + \mu_2\mathbf{I} \tag{1.42}$$

Fig. 1.3 demonstrates this for a number of test patterns. The angle ψ between $\mathbf{n} = (\cos(\psi), \sin(\psi))^T$ and the x-axis is given as

$$\psi = \frac{1}{2}\arctan\left(\frac{2\mathbf{B}_{12}}{\mathbf{B}_{11} - \mathbf{B}_{22}}\right) \tag{1.43}$$

Local maxima of the 2D energy $\mathrm{tr}(\mathbf{B}_{\mathrm{junction}}) = 2\mu_2$ are a good corner and junction detector. Its localization error is only half as big as the errors of the Förstner and Harris detectors (1.4), fig. 1.4 left. Moreover, it does not give multiple responses at saddle-like junctions, fig. 1.4 right. An edge detector can be defined by reducing the edge part of the tensor to a vector

$$\mathbf{g} = \sqrt{\mu_1 - \mu_2}\ \mathbf{n} \tag{1.44}$$

which can be used instead of the gradient vector in Canny's algorithm [3]. This algorithm can then detect lines as well as edges, and sub-pixel accurate localization is still possible, although we have found the noise sensitivity of edge position to be somewhat higher than for the standard Canny algorithm. Edge/line detection can also be integrated with corner/junction detection, because both feature types are derived from the same tensor representation. In this way, a complicated integration step of edge and corner responses into a unified boundary representation is avoided. This is illustrated in fig. 1.5.

1.6 Conclusions

In this paper we discussed the *boundary tensor* as a new way to represent low-level feature strength and orientation. It combines many good properties

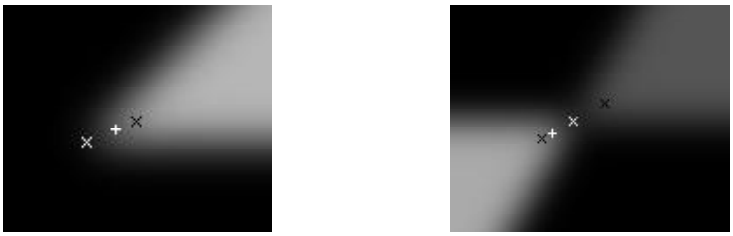

Fig. 1.4. Corner localization of the boundary tensor (white +) and Förstner detector (black ×) relative to exact corner location (white ×). Note the double response of the Förstner detector in the right image. Had the contrast of the two wings of the saddle been equal, the boundary tensor response would have been exact. Harris detector results are very close to Förstner's.

of existing tensor-based approaches and avoids a number of problems. The key insight is that the filters used to compute the boundary tensors should be defined in terms of the *Riesz transform* which determines the angular filter sensitivity, combined with a band-pass which controls scale sensitivity. In this way, the tensor components become products of an angular part that characterizes the feature type, and a radial part that determines feature strength at a given scale. Since the tensor definition does not depend on the dimension of the image, it can readily be used for 3D (volume or space-time) and 4D (volume-time) data sets.

We have shown that a tensor defined with Riesz transform filters reacts like a quadrature filter to locally 1-dimensional configurations. We used filters up to second order, so the boundary tensor also reacts in a predictable and useful way to 2-dimensional configurations that are well approximated by a second-order angular Fourier series, e.g. corners, saddle junctions and crossings of straight linesIf more complex junction configurations have to be analysed, it is possible to extend the boundary tensor definition towards third and higher order Riesz transforms by including terms of the form $\sum_{k,m} T_{ikm} T_{lkm}$ etc. However, higher order filters can no longer be used at small scales due to angular aliasing, so the best trade-off will be application dependent. By choosing the band-pass as the Laplacian of Gaussian, we were able to derive an efficient and accurate spatial domain implementation. In a number of examples we illustrated the good performance of the new method. Further illustrations can be found in [10].

Appendix

Derivation of (1.28): We want to show that in case of a simple signal the trace of the boundary tensor is exactly the 1-dimensional signal energy. Observe that $-\operatorname{sign}(t\tau) = j \operatorname{sign}(t) j \operatorname{sign}(\tau)$ and $\sum_k \mathbf{n}_k^2 = 1$:

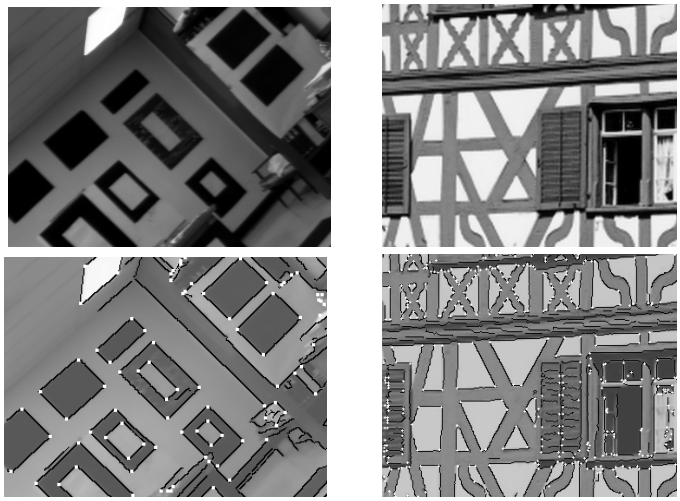

Fig. 1.5. Top: original image. bottom: integrated edge (black lines) and junction (white crosses) detection.

$$
\begin{aligned}
\mathrm{tr}(\mathbf{B}) &= \sum_k \iint \mathbf{n}_k^2(-\operatorname{sign}(t\tau)+1)K(|t|)K(|\tau|)\hat{F}(t\mathbf{n})\hat{F}(\tau\mathbf{n})\, e^{j(t+\tau)\mathbf{n}^T\mathbf{x}}\, dt\, d\tau \\
&= \sum_k \mathbf{n}_k^2 \left(\iint -\operatorname{sign}(t\tau)K(|t|)K(|\tau|)\hat{F}(t\mathbf{n})\hat{F}(\tau\mathbf{n})\, e^{j(t+\tau)\mathbf{n}^T\mathbf{x}}\, dt\, d\tau \right. \\
&\qquad \left. + \iint K(|t|)K(|\tau|)\hat{F}(t\mathbf{n})\hat{F}(\tau\mathbf{n})\, e^{j(t+\tau)\mathbf{n}^T\mathbf{x}}\, dt\, d\tau \right) \\
&= \left(\int j \operatorname{sign}(t)K(|t|)\hat{F}(t\mathbf{n})\, e^{jt\mathbf{n}^T\mathbf{x}}\, dt \right)^2 + \left(\int K(|t|)\hat{F}(t\mathbf{n})\, e^{jt\mathbf{n}^T\mathbf{x}}\, dt \right)^2 \\
&= \left(\mathcal{H}_1[k_1] \star \hat{f} \right)^2 + \left(k_1 \star \hat{f} \right)^2
\end{aligned}
$$

The last transition is simply based on recognizing the integrals as the Fourier domain correspondents of the respective spatial convolutions.

Derivation of (1.31): We show that the boundary tensor components can be expressed in terms of the Fourier coefficients of the angular function F_a when the spectrum $F(\mathbf{u})$ is polar separable. The Fourier series of F_a is:

$$
F_a(\phi) = \frac{\alpha_0}{2\pi} + \sum_{n=1}^{\infty} \frac{j^n}{\pi} \left(\alpha_n \cos(n\phi) + \beta_n \sin(n\phi) \right) \tag{1.45}
$$

where α_n and β_n are the Fourier coefficients according to (1.30). Note that the odd order terms are imaginary, because the spatial domain image is real. We perform the derivation for $\mathbf{B}_{a,11}$, the procedure for the other components

is analogous.

$$\begin{aligned}
\mathbf{B}_{a,11} &= \int_0^{2\pi}\int_0^{2\pi} \mathbf{n}_1(\phi)\mathbf{n}_1(\psi)\left(-1+\mathbf{n}^T(\phi)\,\mathbf{n}(\psi)\right) F_a(\phi)F_a(\psi)\,d\phi\,d\psi \\
&= \iint \cos(\phi)\cos(\psi)\left(-1+\cos(\phi)\cos(\psi)+\sin(\phi)\sin(\psi)\right) F_a(\phi)F_a(\psi)\,d\phi\,d\psi \\
&= -\left(\int \cos(\phi)F_a(\phi)\,d\phi\right)^2+\left(\int \cos^2(\phi)F_a(\phi)\,d\phi\right)^2+\left(\int \cos(\phi)\sin(\phi)F_a(\phi)\,d\phi\right)^2 \\
&= -\left(\int \cos(\phi)F_a(\phi)\,d\phi\right)^2+\left(\int \frac{1+\cos(2\phi)}{2}F_a(\phi)\,d\phi\right)^2+\left(\int \frac{\sin(2\phi)}{2}F_a(\phi)\,d\phi\right)^2
\end{aligned}$$

Now we insert the Fourier series for F_a. Due to orthogonality, all integrals involving a product of different trigonometric functions are zero. Only terms containing the square of a single trigonometric are nonzero, reproducing a Fourier coefficient, e.g.:

$$\int_0^{2\pi} \cos(\phi)F_a(\phi)\,d\phi = \frac{j}{\pi}\,\alpha_1 \int_0^{2\pi} \cos(\phi)\cos(\phi)\,d\phi = j\,\alpha_1$$

Collecting all "surviving" terms, we get the desired result:

$$\mathbf{B}_{a,11} = \alpha_1^2 + \frac{1}{4}(\alpha_0+\alpha_2)^2 + \frac{1}{4}\beta_2^2$$

Derivation of (1.37): We want to calculate the two spatial filter functions (i.e. inverse Fourier transforms) of the first order band-pass Riesz kernels $j\mathbf{u}_i|\mathbf{u}|e^{-\frac{|\mathbf{u}|^2\sigma^2}{2}}$. In this context, it is advantageous to interpret the pair of real valued filters as a complex valued function $g(\mathbf{x}) = g_1(\mathbf{x}) + j\,g_2(\mathbf{x})$. Then the inverse Fourier transform of both filters can be written as a single integral:

$$g(\mathbf{x}) = g_1(\mathbf{x}) + j\,g_2(\mathbf{x}) = \frac{1}{4\pi^2}\iint j\,(\mathbf{u}_1 + j\mathbf{u}_2)|\mathbf{u}|e^{-\frac{|\mathbf{u}|^2\sigma^2}{2}}e^{j\mathbf{u}^T\mathbf{x}}\,d\mathbf{u}$$

We turn to the polar representations $\mathbf{u} = \rho e^{j\phi}$ and $\mathbf{x} = re^{j\psi}$ and get:

$$g(re^{j\psi}) = \frac{1}{4\pi^2}\iint j\,e^{j\phi}\,\rho^2 e^{-\frac{\rho^2\sigma^2}{2}}e^{jr\rho\cos(\phi-\psi)}\,d\phi\,\rho\,d\rho$$

By means of the substitution $\phi' = \phi - \psi$, we can rearrange terms as follows:

$$g(re^{j\psi}) = \frac{1}{4\pi^2}\,e^{j\psi}\int_0^{\infty} j\rho^2 e^{-\frac{\rho^2\sigma^2}{2}}\left(\int_0^{2\pi} e^{j(r\rho\cos(\phi')+\phi')}\,d\phi'\right)\rho\,d\rho$$

The inner integral is a well-known representation of the first-order Bessel function of the first kind: $J_1(t) = \frac{1}{2\pi j}\int_0^{2\pi} e^{j(t\cos(\phi')+\phi')}\,d\phi'$. The outer integral is called the *first-order Hankel transform* of the kernel. It can be computed

by means of a symbolic mathematics program such as *Mathematica* or, more traditionally, by using [1], formulas 11.4.28, 13.4.2-5, and 13.6.3:

$$\begin{aligned}\mathcal{H}ankel[\rho^2 e^{-\frac{\rho^2\sigma^2}{2}}] &= 2\pi \int_0^\infty \rho^2 e^{-\frac{\rho^2\sigma^2}{2}} J_1(r\rho)\rho\, d\rho \\ &= \frac{\pi^{3/2}}{\sqrt{2}\sigma^7}\, r\, e^{-\frac{r^2}{4\sigma^2}} \left((3\sigma^2 - r^2) I_0\left(\frac{r^2}{4\sigma^2}\right) + (r^2 - \sigma^2) I_1\left(\frac{r^2}{4\sigma^2}\right)\right)\end{aligned}$$

where I_0 and I_1 are modified Bessel functions. Inserting this into the previous equation and going back to Cartesian coordinates, we arrive at the result:

$$g_i(\mathbf{x}) = \frac{\mathbf{x}_i}{4\sqrt{2\pi}\sigma^7}\, e^{-\frac{|\mathbf{x}|^2}{4\sigma^2}} \left((|\mathbf{x}|^2 - 3\sigma^2) I_0\left(\frac{|\mathbf{x}|^2}{4\sigma^2}\right) - (|\mathbf{x}|^2 - \sigma^2) I_1\left(\frac{|\mathbf{x}|^2}{4\sigma^2}\right)\right)$$

During the above calculations, the expression $g(re^{j\psi}) \sim rM(\frac{5}{2}, 2, \frac{-r^2}{4\sigma^2})$ occurs as an intermediate result, where M is a confluent hypergeometric function. Using this together with [1], 13.1.5, we obtain the filters' asymptotic behavior for large r as $\mathcal{O}(r^{-4})$.

References

1. M. Abramowitz, I. Stegun: *Handbook of Mathematical Functions*, Dover: 1972
2. J. Bigün, G. Granlund, J. Wiklund: *Multidimensional Orientation Estimation with Applications to Texture Analysis and Optic Flow*, IEEE Trans. Pattern Analysis and Machine Intelligence, 13(8):775-790, 1991
3. J. Canny: *A Computational Approach to Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986
4. G. Farnebäck: *Polynomial Expansion for Orientation and Motion Estimation*, PhD thesis, Linköping University, Dissertation No. 790, 2002
5. M. Felsberg, G. Sommer: *The Monogenic Signal*, IEEE Trans. Image Processing, 49(12):3136-3144, 2001
6. W. Förstner: *A Feature Based Correspondence Algorithm for Image Matching*, Intl. Arch. of Photogrammetry and Remote Sensing, vol. 26, pp. 150-166, 1986
7. W. Freeman, E. Adelson: *The design and use of steerable filters*, IEEE Trans. Pattern Analysis Machine Intelligence, 13(9):891-906, 1991
8. C.G. Harris, M.J. Stevens: *A Combined Corner and Edge Detector*, Proc. of 4th Alvey Vision Conference, pp. 147-151, 1988
9. G. Granlund, H. Knutsson: *Signal Processing for Computer Vision*, Kluwer Academic Publishers, 1995
10. U. Köthe: *Integrated Edge and Junction Detection with the Boundary Tensor*, in: ICCV 03, Proc. of 9th Intl. Conf. on Computer Vision, Nice 2003, vol. 1, pp. 424-431, Los Alamitos: IEEE Computer Society, 2003
11. H.H. Nagel: *Analyse und Interpretation von Bildfolgen II*, Informatik-Spektrum, 8(6):312-327, 1985
12. K. Nordberg, G. Farnebäck: *A Framework for Estimation of Orientation and Velocity*, Proc. IEEE Intl. Conf. on Image Processing, vol. 3:57-60, 2003
13. G. Sicuranza: *Quadratic Filters for Signal Processing*, Proc. of the IEEE, 80(8):1263-1285, 1992