

# Riesz-Transforms vs. Derivatives: On the Relationship Between the Boundary Tensor and the Energy Tensor

Ulrich Köthe<sup>1</sup> and Michael Felsberg<sup>2\*</sup>

<sup>1</sup> Cognitive Systems Group, University of Hamburg  
koethe@informatik.uni-hamburg.de

<sup>2</sup> Computer Vision Laboratory, Linköping University  
mfe@isy.liu.se

**Abstract.** Traditionally, quadrature filters and derivatives have been considered as alternative approaches to low-level image analysis. In this paper we show that there actually exist close connections: We define the quadrature-based boundary tensor and the derivative-based gradient energy tensor which exhibit very similar behavior. We analyse the reason for this and determine how to minimize the difference. These insights lead to a simple and very efficient integrated feature detection algorithm.

## 1 Introduction

Image features such as edges and corners can be detected by analysing the image in the neighborhood of every (candidate) point. A compact representation of the low-order characteristics of these neighborhoods is given by the first few derivatives at every point (the  $n$ -jet [9]). Numerous feature descriptors for edges, lines, blobs, corners and so on have been defined by various combinations of low-order derivatives (see e.g. [5,12]). However, these descriptors are usually only valid for a single feature type, and give no or wrong responses at points where the underlying feature model is violated. Improvements can be achieved by moving from scalar feature descriptors to tensor based ones. Second order tensors cannot only represent feature strength, but also allow to distinguish between intrinsically 1- and 2-dimensional features (edges vs. corners) and measure orientation.

The most common derivative-based tensor is the structure tensor [1,6] which is obtained by spatial averaging of the outer product of the gradient. It can represent step edges and their corners/junctions but is less suitable for the detection of lines and other second order features. All these feature types are covered by the energy tensor [3], which includes higher order derivatives (details below).

A different approach to feature detection is taken by the quadrature filter method [7,8] where derivatives are replaced with filters that are related to each other by the Hilbert transform. These operators react, by design, uniformly to both edges and lines. This property is called phase invariance because edges and lines can be interpreted as arising from the same magnitude spectrum, but at

---

\* This work has been supported by EC Grant IST-2002-002013 MATRIS and by EC Grant IST-2003-004176 COSPAL.

different (namely odd and even) phase. In 2D it is common to apply a number of 1D quadrature filters at different orientations. The filter responses can then be combined into an orientation tensor [8]. However, the orientation tensor reacts in a well-defined way only to 1-dimensional features. This problem is solved by means of the boundary tensor [10], which uses truly 2-dimensional quadrature filters to also model certain 2D feature types (details also given below).

When we experimented with both the energy and the boundary tensors, we observed a striking similarity of their behavior – qualitatively, their results are almost indistinguishable. This paper is devoted to a more detailed analysis of the relationship of the two approaches. We pursue this analysis on three levels: First, we establish a formal similarity between the derivative and quadrature filter methods by basing the latter on (first and second order) Riesz transform operators [2] which closely resemble derivatives. Second, we show that the spectral properties of the two tensors are very similar when optimal scales ratios are chosen. Third, we report on experiments illustrating the similarity in practice.

## 2 Tensor Definitions

The structure tensor is the most common derivative based tensor. It is defined as the spatial average of the outer product of the gradient  $\nabla f$  with itself [1,6]:

$$\mathbf{S} = g \star (\nabla f)(\nabla f)^T \quad (1)$$

where  $g$  is a smoothing filter (usually Gaussian),  $f$  the original image, and derivatives are always understood to operate at a particular scale. For the purpose of our analysis, it is advantageous to approximate the gradient with a Taylor series:

$$\nabla f(\mathbf{x}) \approx \nabla f|_{\mathbf{x}=\mathbf{x}_0} + \nabla \nabla^T f|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) \quad (2)$$

where  $\nabla \nabla^T f = \mathcal{H}f$  is the Hessian matrix. Inserting this into (1), we can execute the convolution analytically. If  $g$  is radially symmetric, the odd powers of  $\mathbf{x}$  cancel out, whereas the even ones give a constant. We get:

$$\mathbf{S} \approx (\nabla f)(\nabla f)^T + \lambda(\mathcal{H}f)(\mathcal{H}f)^T \quad (3)$$

where the parameter  $\lambda$  depends on  $g$ 's shape and scale. This operator is very good at detecting step edges and their corners, but often shows multi-modal or no responses at second-order features such as line edges and junctions. By adjusting  $\lambda$ , the behavior can be somewhat improved, but it is usually impossible to find a  $\lambda$  that works equally well on the entire image.

A richer signal model can be employed with the *energy tensor* [3]:

$$\mathbf{E} = (\nabla b)(\nabla b)^T - b(\mathcal{H}b) \quad (4)$$

where  $b$  is the signal to be analyzed. This is structurally similar to (3), but the square of the Hessian has been replaced with the negative product of the function  $b$  and its Hessian. We'll show later that the energy tensor achieves better feature

detection results when different scales are used for different derivative orders. In a strict sense, the name “energy tensor” may not be justified because  $\mathbf{E}$  is not in general positive semi-definite. But it has this property under certain conditions and degrades gracefully if these conditions are slightly violated (cf. figure 4, see [4] for an in-depth discussion of this issue). In image analysis, the energy tensor cannot be used in its pure form, because images are not DC free, so the energy would be dominated by the DC magnitude (average gray level) if  $b$  were the image  $f$  itself. Instead one computes  $\mathbf{E}$  from a bandpass filtered version of the image, for example one defines  $b = \nabla^T \nabla g \star f$  when the bandpass is the Laplacian or Gaussian. Since the Laplacian is a second order derivative,  $\mathbf{E}$  is effectively calculated from second, third and fourth order derivatives.

Unfortunately, this means that the important first order image structure is not considered at all. Therefore, we developed a variant called *gradient energy tensor*, or GET operator [4]. Here,  $b = \nabla g \star f$ , so  $b$  is the Gaussian gradient vector. The gradient of  $b$  is then the Hessian of  $f$ , whereas the Hessian of  $b$  gives a third order tensor. Since the energy tensor is only a second order tensor, two indices of this third order tensor are immediately contracted, giving:

$$(\mathcal{T}f)_i = \sum_{j=1}^N \frac{\partial^3 f}{\partial x_i \partial x_j^2} = \nabla(\nabla^T \nabla f) \quad \text{i.e. in 2D: } \mathcal{T}f = \begin{pmatrix} f_{xxx} + f_{xyy} \\ f_{xxy} + f_{yyy} \end{pmatrix} \quad (5)$$

( $N$  is the space dimension).  $\mathcal{T}f$  is equivalent to the gradient of the Laplacian of  $f$ , as can be seen by switching the order of differentiation and contraction. Since the outer product of two different tensors is not commutative, the gradient energy tensor must be symmetrized. This results in the following definition:

$$\mathbf{G} = \mathbf{G}^{\text{even}} + \mathbf{G}^{\text{odd}} = (\mathcal{H}f)(\mathcal{H}f)^T - \frac{1}{2} ((\nabla f)(\mathcal{T}f)^T + (\mathcal{T}f)(\nabla f)^T) \quad (6)$$

The *boundary tensor* was introduced in [10] using circular harmonics. Here we base its definition on the Riesz transform [2] to emphasize the formal similarity of quadrature filters to derivatives. The Riesz transform is the  $N$ -dimensional generalization of the Hilbert transform. It is defined in the Fourier domain as:

$$\mathcal{R} \circ \bullet - \mathbf{i} \frac{\mathbf{u}}{|\mathbf{u}|} \quad (7)$$

where  $\mathbf{u}$  is the frequency vector. Since the derivative is defined as  $\nabla \circ \bullet - \mathbf{i}\mathbf{u}$ , the two operators differ only by a factor of  $|\mathbf{u}|$  in the Fourier domain. The difference becomes clearer in polar coordinates where  $\mathcal{R} \circ \bullet - \mathbf{i}(\cos(\phi), \sin(\phi))^T$  and  $\nabla \circ \bullet - \mathbf{i}\rho(\cos(\phi), \sin(\phi))^T$ . Both operators have the same angular behavior. But the derivative operator also acts as a high-pass filter, whereas the Riesz transform leaves the radial shape of the spectrum unaltered. This property is responsible for the desirable phase invariance of the boundary tensor.

The spatial domain Riesz transform operator decays only as  $\mathcal{O}(|\mathbf{x}|^{-N})$ , where  $N$  is the space dimension. Therefore one applies Riesz transforms to a bandpass filtered version  $b$  of the image  $f$ . The boundary tensor is then defined as

$$\mathbf{B} = \mathbf{B}^{\text{even}} + \mathbf{B}^{\text{odd}} = (\mathcal{Q}b)(\mathcal{Q}b)^T + (\mathcal{R}b)(\mathcal{R}b)^T \quad (8)$$

where  $\mathcal{Q} = \mathcal{R}\mathcal{R}^T$  denotes the second order Riesz transform resulting in a matrix analogous to the Hessian (In contrast to the 1D Hilbert transform, which reproduces the negated original signal if applied twice, higher order Riesz transforms are useful because they create tensors when  $N \geq 2$ ). Eq. (8) is formally equivalent to (3) when we set  $\lambda = 1$  and replace derivatives with Riesz transforms. It should also be noted that the boundary tensor is always positive semi-definite by construction. Various bandpass filters can be used to obtain  $b$ . In [10], we used  $|\mathbf{u}| \exp(-|\mathbf{u}|^2 \sigma^2/2)$ , but in this paper we choose the Laplacian of Gaussian  $|\mathbf{u}|^2 \exp(-|\mathbf{u}|^2 \sigma^2/2)$  because this allows us to establish a very strong *functional* relationship between the gradient energy tensor (6) and the boundary tensor.

### 3 Analysis of the Tensors

In order to analyse the behavior of the tensors, it is beneficial to express the convolution operation explicitly with integrals. For simplicity, we assume that the coordinate origin is at the center of the current window. Due to Parseval's theorem we can then express the integral in either the spatial or Fourier domains. We must only take into account that the kernels are reflected in the spatial domain expressions, which has no effect for even kernels but requires a sign-change for odd kernels. Since we are always taking products of two odd filter responses, this sign also cancels out. Using the Laplacian of Gaussian bandpass, the boundary tensor components can be expressed in the Fourier domain as

$$\begin{aligned} \mathbf{B}_{ij} = & \int -\mathbf{u}_i \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma^2/2} F(\mathbf{u}) d\mathbf{u} \int -\mathbf{u}_j \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma^2/2} F(\mathbf{u}) d\mathbf{u} \\ & + \int -i \mathbf{u}_i |\mathbf{u}| e^{-|\mathbf{u}|^2 \sigma^2/2} F(\mathbf{u}) d\mathbf{u} \int -i \mathbf{u}_j |\mathbf{u}| e^{-|\mathbf{u}|^2 \sigma^2/2} F(\mathbf{u}) d\mathbf{u} \end{aligned} \quad (9)$$

where  $F(\mathbf{u})$  is the image spectrum, and we use Einstein's summation convention (for index  $k$ ). The components of the gradient energy tensor are

$$\begin{aligned} \mathbf{G}_{ij} = & \int -\mathbf{u}_i \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma_2^2/2} F(\mathbf{u}) d\mathbf{u} \int -\mathbf{u}_j \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma_2^2/2} F(\mathbf{u}) d\mathbf{u} \\ & - \frac{1}{2} \left( \int -i \mathbf{u}_i e^{-|\mathbf{u}|^2 \sigma_1^2/2} F(\mathbf{u}) d\mathbf{u} \int i \mathbf{u}_j \mathbf{u}_k \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma_3^2/2} F(\mathbf{u}) d\mathbf{u} \right. \\ & \left. + \int -i \mathbf{u}_j e^{-|\mathbf{u}|^2 \sigma_1^2/2} F(\mathbf{u}) d\mathbf{u} \int i \mathbf{u}_i \mathbf{u}_k \mathbf{u}_k e^{-|\mathbf{u}|^2 \sigma_3^2/2} F(\mathbf{u}) d\mathbf{u} \right) \end{aligned} \quad (10)$$

where we allow the derivatives of different order to be applied at different scales  $\sigma_1, \sigma_2, \sigma_3$ . If we equate  $\sigma$  and  $\sigma_2$ , the even parts of  $\mathbf{B}$  and  $\mathbf{G}$  become equal, so we will require this from now on. We analyse at first how the two tensors react to intrinsically 1-dimensional images, that is when  $F(\mathbf{u}) = F(t\mathbf{n})$  holds along a particular direction  $\mathbf{n}$ , and  $F(\mathbf{u}) = 0$  otherwise. Then the  $\mathbf{u}_i$  reduce to  $\mathbf{n}_i t$ , and the 2D integrals become 1D ones. The even part of both tensors is:

$$\mathbf{B}_{ij}^{\text{even}} = \mathbf{G}_{ij}^{\text{even}} = \mathbf{n}_i \mathbf{n}_j \left( \int t^2 e^{-t^2 \sigma_2^2/2} \Re(F(t)) dt \right)^2 \quad (11)$$

and the odd parts are:

$$\mathbf{B}_{ij}^{\text{odd}} = \mathbf{n}_i \mathbf{n}_j \left( \int -\mathbf{i} t |t| e^{-t^2 \sigma_2^2 / 2} \mathbf{i} \Im(F(t)) dt \right)^2 \quad (12)$$

$$\mathbf{G}_{ij}^{\text{odd}} = -\mathbf{n}_i \mathbf{n}_j \int -\mathbf{i} t e^{-t^2 \sigma_1^2 / 2} \mathbf{i} \Im(F(t)) dt \int \mathbf{i} t^3 e^{-t^2 \sigma_3^2 / 2} \mathbf{i} \Im(F(t)) dt \quad (13)$$

where we took advantage of the fact that the spectra of real signals have even real and odd imaginary components. It can be seen that  $\mathbf{B}$  is indeed a quadrature filter: The kernels of the even and odd tensor parts are related by the Hilbert transform  $-\mathbf{i} \text{sign}(t)$ . Thus, if we shift the signal phase by  $\pi/2$  (i.e. if we swap real and imaginary signal components, with the appropriate adjustment of spectrum symmetries), even and odd tensor parts are simply exchanged, but their sum remains unaltered. This is precisely the requirement of phase invariance. That requirement is not fulfilled by the GET operator: It has the same even part as the boundary tensor, but the odd parts differ. Detailed analysis of the odd parts reveals that the difference can actually be made very small. Consider at first a simple sine signal, i.e.  $F(t) = \mathbf{i} \frac{a}{2} (\delta(t - \omega_a) - \delta(t + \omega_a))$ . We get

$$\begin{aligned} \mathbf{B}_{ij}^{\text{odd}} &= \mathbf{n}_i \mathbf{n}_j a^2 \omega_a^4 e^{-\omega_a^2 \sigma_2^2} \\ \mathbf{G}_{ij}^{\text{odd}} &= \mathbf{n}_i \mathbf{n}_j a^2 \omega_a^4 e^{-\omega_a^2 (\sigma_1^2 + \sigma_3^2) / 2} \end{aligned}$$

These expressions are equal when  $\sigma_2^2 = (\sigma_1^2 + \sigma_3^2) / 2$  which we will require from now on. A more complicated case is the superposition of two sine waves  $F(t) = \mathbf{i} \frac{a}{2} (\delta(t - \omega_a) - \delta(t + \omega_a)) + \mathbf{i} \frac{b}{2} (\delta(t - \omega_b) - \delta(t + \omega_b))$ . Then we get

$$\begin{aligned} \mathbf{B}_{ij}^{\text{odd}} &= \mathbf{n}_i \mathbf{n}_j \left( a \omega_a^2 e^{-\omega_a^2 \sigma_2^2 / 2} + b \omega_b^2 e^{-\omega_b^2 \sigma_2^2 / 2} \right)^2 \\ \mathbf{G}_{ij}^{\text{odd}} &= \mathbf{B}_{ij}^{\text{odd}} + \mathbf{n}_i \mathbf{n}_j a b \omega_a \omega_b \left( \omega_a e^{-(\omega_a^2 \sigma_3^2 + \omega_b^2 \sigma_1^2) / 4} - \omega_b e^{-(\omega_a^2 \sigma_1^2 + \omega_b^2 \sigma_3^2) / 4} \right)^2 \end{aligned}$$

The eigenvalue of  $\mathbf{B}^{\text{odd}}$  (which we obtain by simply dropping  $\mathbf{n}_i \mathbf{n}_j$ ) is always positive, as required for a signal energy. However, the eigenvalue of  $\mathbf{G}^{\text{odd}}$  can become negative if  $a$  and  $b$  have opposite signs, i.e. if the two sines have opposite phase. This counters the intuition that the *energy* tensor  $\mathbf{G}$  indeed represents signal energy. However, due to the statistical properties of natural images the situation is not so bad in practice: High energy in the derivatives typically occurs at object boundaries (edges and corners/junctions). At these points the signal components have the *same* phase over many frequencies (phase congruency, [11]). Then the error term in  $\mathbf{G}$  is positive, and the measured energy becomes too large rather than too small. Negative energy typically occurs only in flat, but noisy areas, where it is safe to simply truncate negative eigenvalues to zero.

In addition, we can try to adjust the ratio  $\sigma_3 / \sigma_1$  so that the magnitude of the error term becomes as small as possible. It is necessary to use a scale-normalized error measure, because one could otherwise make the error arbitrarily small by taking  $\sigma_3 \rightarrow \infty$ . The natural scale normalization for the Laplacian of Gaussian

is  $\sigma_2^2 [= (\sigma_1^2 + \sigma_3^2)/2]$  [12], so that  $\mathbf{B}$  has to be multiplied with  $\sigma_2^4$ . To make the response of  $\mathbf{G}$  comparable, we normalize it with the same factor. Then we integrate over  $\omega_a$  and  $\omega_b$  to calculate the average error over all frequency pairs:

$$\begin{aligned}\epsilon &= \frac{(\sigma_1^2 + \sigma_3^2)^2}{4} \iint \omega_a \omega_b \left( \omega_a e^{-(\omega_a^2 \sigma_3^2 + \omega_b^2 \sigma_1^2)/4} - \omega_b e^{-(\omega_a^2 \sigma_1^2 + \omega_b^2 \sigma_3^2)/4} \right)^2 d\omega_a d\omega_b \\ &= \frac{1}{\sigma_1^2} + \frac{2}{\sigma_3^2} + \frac{\sigma_1^2}{\sigma_3^4} - \frac{2\pi}{\sigma_1^2 + \sigma_3^2}\end{aligned}\quad (14)$$

(we dropped the factor  $\mathbf{n}_i \mathbf{n}_j a b$  not depending on the ratio). The error is minimized for  $\sigma_3/\sigma_1 = 1/\sqrt{\pi^{1/3} - 1} \approx 1.47$ . It is interesting to compare the optimal error with the error obtained for other ratios: If  $\sigma_1 = \sigma_2 = \sigma_3$ , the error becomes more than 5 times as big! If  $\sigma_3/\sigma_1 = \sqrt{3}$  and  $\sigma_2/\sigma_1 = \sqrt{2}$  (which means that the same first derivative filter is applied repeatedly for the higher order derivatives, resulting in a very efficient algorithm), the error is only 36% bigger.

Another possibility to find an optimal scale ratio is to start directly from (12) and (13). We transform the products of integrals in these equations into 2-dimensional integrals over the product of the integrands. Then we interpret terms not depending on the signal spectrum as quadratic filter kernels [13]:

$$\begin{aligned}\left( \int -\mathbf{i} t |t| e^{-t^2 \sigma_2^2/2} F(t) dt \right)^2 &= - \iint B(t_1, t_2) F(t_1) F(t_2) dt_1 dt_2 \\ \int -\mathbf{i} t e^{-t^2 \sigma_1^2/2} F(t) dt \int \mathbf{i} t^3 e^{-t^2 \sigma_3^2/2} F(t) dt &= \iint G(t_1, t_2) F(t_1) F(t_2) dt_1 dt_2\end{aligned}$$

with (note that  $G$  is symmetric due to the symmetrization of  $\mathbf{G}$ )

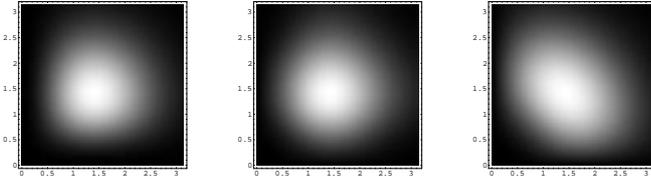
$$B(t_1, t_2) = t_1 t_2 |t_1 t_2| e^{-(t_1^2 + t_2^2) \sigma_2^2/2} \quad (15)$$

$$G(t_1, t_2) = \frac{1}{2} \left( t_1 t_2^3 e^{-(t_1^2 \sigma_1^2 + t_2^2 \sigma_3^2)/2} + t_1^3 t_2 e^{-(t_1^2 \sigma_3^2 + t_2^2 \sigma_1^2)/2} \right) \quad (16)$$

We choose the ratio  $\sigma_3/\sigma_1$  so that the scale-normalized mean squared difference between the two kernels is minimized:

$$\epsilon^2 = \sigma_2^8 \iint (B(t_1, t_2) - G(t_1, t_2))^2 dt_1 dt_2 \quad (17)$$

The minimum is achieved for  $\sigma_3/\sigma_1 \approx 1.55$ . The choice  $\sigma_1 = \sigma_2 = \sigma_3$  gives again a 5 times higher residual (see fig. 1), whereas it increases by only 23% for  $\sigma_3/\sigma_1 = \sqrt{3}$ . We also repeated the two optimizations while weighting the importance of the frequencies according to  $1/\omega$ , which better reflects the situation in real images. After modifying (14) and (17) accordingly, we got optimal  $\sigma_3/\sigma_1$  ratios of  $\sqrt{3}$  and 1.8 respectively, and the dependency of the residual on the ratio was reduced. Consequently, scale ratios between 1.5 and 1.8 give reasonable results, whereas it appears to be a bad idea to apply all derivatives at the same scale.



**Fig. 1.** Left:  $B(t_1, t_2)$  (for  $t_1, t_2 > 0$ ,  $\sigma_2 = 1$ ); center:  $G(t_1, t_2)$  with  $\sigma_3/\sigma_1 = 1.55$  and  $(\sigma_1^2 + \sigma_3^2)/2 = 1$ ; right:  $G(t_1, t_2)$  for  $\sigma_3/\sigma_1 = 1$ : the deviation from  $B$  is much higher.

Now we analyse the response of the tensors to intrinsically 2-dimensional structures. To simplify we consider points  $\mathbf{x}$  where the spectrum  $F(\mathbf{u})$  computed with  $\mathbf{x}$  as coordinate origin is (approximately) polar separable within the pass band of the tensor filters. In case of the boundary tensor, the pass band is determined by the Laplacian of Gaussian, and we require  $|\mathbf{u}|^2 \exp(-|\mathbf{u}|^2 \sigma^2/2) F(\mathbf{u}) \approx \rho^2 \exp(-\rho^2 \sigma^2/2) F_r(\rho) F_a(\phi)$ . Then the integrals over  $\mathbf{u}$  can be separated into products of two integrals over the radial and angular coordinates:

$$\mathbf{B}_{ij} = \int \mathbf{e}_i(\phi) F_a(\phi) d\phi \int \mathbf{e}_j(\phi) F_a(\phi) d\phi \left( \int \rho^2 e^{-\rho^2 \sigma^2/2} F_r(\rho) \rho d\rho \right)^2 \quad (18)$$

$$+ \int \mathbf{e}_i(\phi) \mathbf{e}_k(\phi) F_a(\phi) d\phi \int \mathbf{e}_j(\phi) \mathbf{e}_k(\phi) F_a(\phi) d\phi \left( \int \rho^2 e^{-\rho^2 \sigma^2/2} F_r(\rho) \rho d\rho \right)^2$$

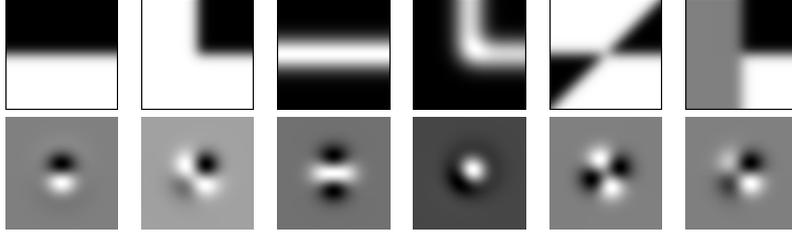
with  $\mathbf{e}(\phi) = (\cos(\phi), \sin(\phi))^T$ . When we define the Fourier coefficients of  $F_a(\phi)$  by  $c_m = \int \cos(m\phi) F_a(\phi) d\phi$  and  $s_m = \int \sin(m\phi) F_a(\phi) d\phi$ , the trace of the boundary tensor becomes:

$$\text{tr}(\mathbf{B}) = \frac{c_0^2 + 2c_1^2 + 2s_1^2 + c_2^2 + s_2^2}{2} \iint B(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \quad (19)$$

where the kernel  $B$  simplifies to  $B(\rho_1, \rho_2) = \rho_1^2 \rho_2^2 \exp(-(\rho_1^2 + \rho_2^2) \sigma_2^2/2)$  because  $\rho_1$  and  $\rho_2$  are non-negative. The trace is determined by two local image properties: by the local contrast (as given by the radial integrals), and by how well the angular shape variation is captured with low-order circular harmonics (as given by the magnitude of the first five Fourier coefficients). It is interesting to compare this with the gradient at a polar separable location:

$$(\nabla f)^2 = (c_1^2 + s_1^2) \iint S(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \quad (20)$$

where  $S(\rho_1, \rho_2) = \rho_1 \rho_2 \exp(-(\rho_1^2 + \rho_2^2) \sigma_1^2/2)$ . Again we obtain an expression of the form ‘‘contrast times Fourier coefficients’’. Since all Fourier coefficients in (19) and (20) are weighted by only one radial integral, the form of this integral is not crucial (gradients can be defined with many filters, the boundary tensor originally used the kernel  $S$  above, see [10]). Thus, the key difference between the boundary tensor and the gradient squared is that the former includes *three*



**Fig. 2.** Top: original images; bottom: reconstruction obtained by a weighted sum of the boundary tensor filter kernels, where the weights correspond to the normalized filter responses at the center pixel.

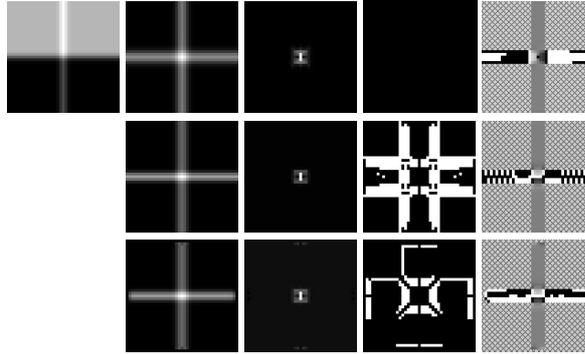
*additional Fourier coefficients:* The boundary tensor can be interpreted as a natural generalization of the gradient towards a more sophisticated local signal model. Fig. 2 illustrates this generalization by means of a local image reconstruction from the filter responses that constitute the boundary tensor. This reconstruction essentially shows how the boundary tensor “sees” certain shapes. Obviously large part of the shape information is already contained in five filter responses (only the first two patterns could be reconstructed from the gradient filters). A similar generalization to five Fourier coefficients is achieved by the structure tensor (3). At a polar separable point, its trace can be written as:

$$\begin{aligned} \text{tr}(\mathbf{S}) = & \lambda \frac{c_0^2 + c_2^2 + s_2^2}{2} \iint B(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \\ & + (c_1^2 + s_1^2) \iint S(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \end{aligned} \quad (21)$$

But here the even and odd Fourier coefficients are weighted by *different* radial integrals. One can try to optimize  $\lambda$  and  $\sigma_2/\sigma_1$  in order to minimize the difference between  $B$  and  $S$ , but it turns out that good agreement can only be achieved for a few frequencies at a time. This means in practice that at many image locations the contributions of even and odd tensor parts are not well balanced, which results in multiple responses for a single boundary or boundary gaps. Fortunately, the trace of the GET operator shows much better behavior:

$$\begin{aligned} \text{tr}(\mathbf{G}) = & \frac{c_0^2 + c_2^2 + s_2^2}{2} \iint B(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \\ & + (c_1^2 + s_1^2) \iint G(\rho_1, \rho_2) F_r(\rho_1) F_r(\rho_2) \rho_1 d\rho_1 \rho_2 d\rho_2 \end{aligned} \quad (22)$$

Although even and odd Fourier coefficients are still weighted differently, we have shown above (see fig. 1) that the kernels  $B$  and  $G$  can be made extremely similar, so that the GET operator  $\mathbf{G}$  can be considered a very good approximation of the boundary tensor  $\mathbf{B}$ . Strictly speaking this applies only at polar separable image locations, but we have found experimentally that this desirable behavior carries over to many interesting images features.

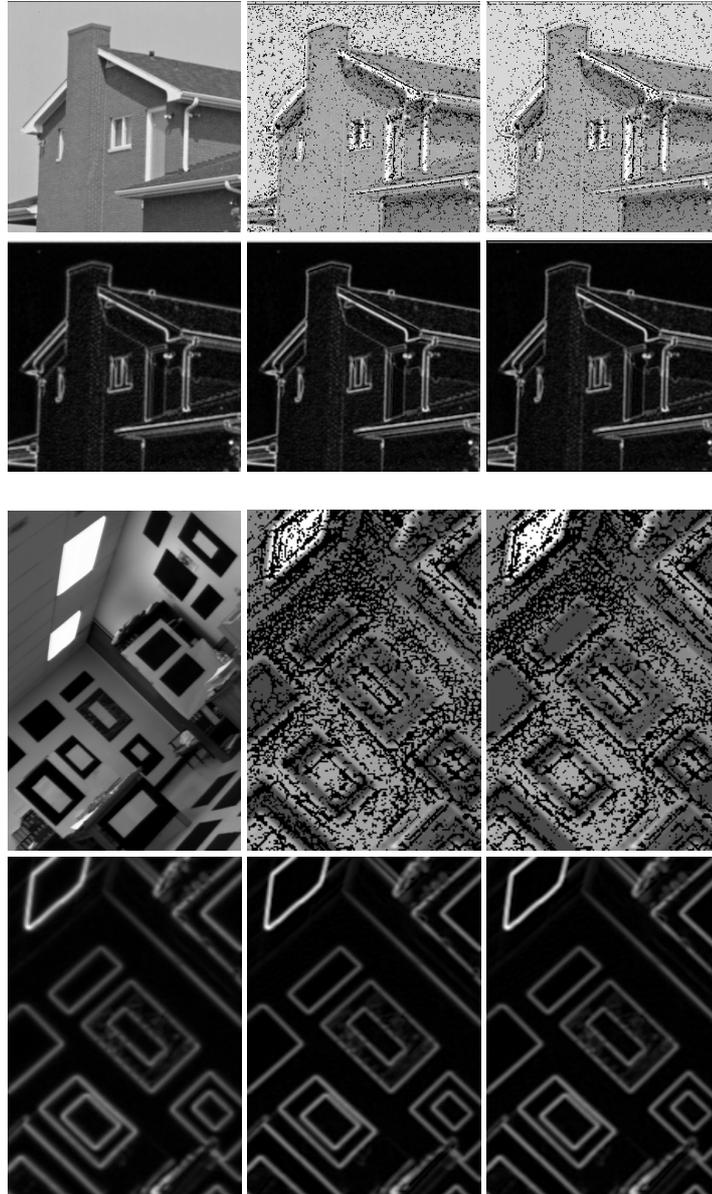


**Fig. 3.** Top left: original image; col. 2: tensor trace (row 1: boundary tensor, row 2: GET operator, Gaussian derivatives, row 3: GET operator,  $3 \times 3$  filter); col. 3: junction strength; col. 4: locations with negative junction strength; col. 5: edge orientation (hatched: not a 1D feature, black/white: horizontal edge, gray: vertical edge).

## 4 Experimental Comparison of Tensor Based Feature Detectors

The local shape information represented by the gradient energy and boundary tensors can be extracted in the usual way. The most important tensor characteristic in this context is the *tensor trace* which indicates the local contrast independently of feature type (edge, line, corner, or junction) and thus acts as a general boundary indicator. Intrinsically 1- and 2-dimensional parts of the boundary can be distinguished by the tensors' *eigenvalues*: The smaller eigenvalue indicates corner and junction strength, whereas the difference of the two eigenvalues represents edge and line strength. If the eigenvalues indicate a 1D structure, the *eigenvector* corresponding to the large eigenvalue points in the direction perpendicular to the edge or line. In all experiments we compare the following tensors: (i) the boundary tensor computed with the Laplacian of Gaussian at  $\sigma = 0.9$ , (ii) the gradient energy tensor computed from Gaussian derivatives with  $\sigma_2 = 0.9$  and various ratios  $\sigma_3/\sigma_1$  (images are shown for  $\sigma_3/\sigma_1 = 1.5$ ), and (iii) the gradient energy tensor computed by applying Scharr's optimal  $3 \times 3$  derivative filter  $(3, 10, 3)^T(1, 0, -1)/32$  one, two and three times [14].

In the first experiment, we computed the tensors for simple test images. Fig. 3 shows typical results. We found that all tensor variants have very similar trace (boundary strength) and small eigenvalue (corner strength). The trace is phase invariant (to very good approximation in case of the GET operator), i.e. responds uniformly to step edges and lines. The step edge response of the GET operator is slightly narrower than that of the boundary tensor, which may be desirable in practice as it reduces the likelihood that nearby edges blend into each other. On the other hand, there are several locations where the small eigenvalues of the GET operators are negative, but this only occurs away from junctions. The large eigenvalues are always positive.



**Fig. 4.** Rows 1 and 3: original image, negative small eigenvalues of GET operator with  $\sigma_3/\sigma_1 = 1.5$ , negative small eigenvalues of GET operator with Scharr filter; rows 2 and 4: square root of tensor trace for boundary tensor and the two GET operators.



**Fig. 5.** Integrated boundary detection from boundary tensor, GET operator ( $\sigma_3/\sigma_1 = 1.5$ ) and GET operator (Schar filter).

The second experiment illustrates the same properties on real images (fig. 4). Again the traces are almost indistinguishable. The small eigenvalue is negative at about 10...35% of the pixels, but never at corners or junctions (we checked this against the corner locations detected with the boundary tensor). Negative values in the trace occur much less frequently (about 1...10% of the pixels, and never on edges) because the large eigenvalue was never negative in the experiments (formal proof of this fact is subject to further research). Gaussian derivatives and the Schar filter perform similarly, with the exception of derivatives at  $\sigma_3/\sigma_1 = 1$ , where the number of negative pixels increases 1.5...3-fold.

In the last experiment we show that the three tensors can be used for integrated edge and junction detection as described in [10]. The tensor at each image location is decomposed into its corner/junction (small eigenvalue) and edge/line (difference of eigenvalues times main eigenvector) parts. Then local maxima above a certain threshold are detected in the corner/junction map, and oriented non-maxima suppression and thresholding is performed in the edge/line map. The resulting boundaries are overlaid over the original image, see fig. 5. Again, the results are extremely similar.

## 5 Conclusions

Traditionally, quadrature filters and derivatives have been used by what might be considered different schools of low-level image analysis. In this paper we demonstrated a very close relationship between two typical methods from both camps: the boundary tensor and the GET operator. It turned out that these operators behave almost identically in experiments. Theoretical analysis supports this finding: We established a close formal relationship by giving a new boundary tensor definition using Riesz transforms. And we showed for typical 1- and 2-dimensional image structures that the resulting integral expressions are very similar for the two methods, if suitable operator scales are chosen.

Boundary tensor and GET operator can be interpreted as natural generalizations of the gradient, which uses filters whose angular behaviour corresponds to the first two odd circular harmonics: they add filters representing the first

three even circular harmonics. It should be stressed that the feature detection capability depends mainly on this angular behavior – the radial filter shape can be altered considerably, as long as it remains approximately equal for all filters (in the Fourier domain): The boundary tensor can be defined with other band-pass filters, and slightly different radial shapes for even and odd filters can be tolerated in the GET operator. But the angular behavior has to be equal.

Some differences remain: The boundary tensor is always positive semi-definite by construction, whereas the GET operator sometimes measures negative corner strength. Since this does not occur at true corners, it is safe to truncate negative values at zero. On the other hand, the filters constituting the GET operator are simpler than the ones for the boundary tensor (in the spatial domain). The GET operator can already be computed accurately with a  $3 \times 3$  filter mask, and only seven convolutions with this mask are needed. This is roughly the same effort as needed for the structure tensor, but the underlying feature model is much richer, containing not only edges but also lines, corners, and junctions. Extension to 3D and to multiple scales will likely be easier for the GET operator due to the huge existing body of established analysis for derivative filters.

## References

1. J. Bigün, G. Granlund: *Optimal Orientation Detection of Linear Symmetry*, in: ICCV 87, Proc. 1st Intl. Conf. on Computer Vision, pp. 433-438, 1987
2. M. Felsberg, G. Sommer: *The Monogenic Signal*, IEEE Trans. Image Processing, 49(12):3136-3144, 2001
3. M. Felsberg, G. Granlund: *POI Detection Using Channel Clustering and the 2D Energy Tensor*, in: Rasmussen et al. (Eds.): Pattern Recognition, Proc. 26th DAGM Symposium, pp. 103-110, Springer LNCS 3175, 2004
4. M. Felsberg, U. Köthe: *GET: The Connection Between Monogenic Scale-Space and Gaussian Derivatives*, in: ScaleSpace 2005 (this volume)
5. L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, and M.A. Viergever: *Cartesian differential invariants in scale-space*, J. of Mathematical Imaging and Vision, 3(4), pp. 327-348, 1993
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.T. Freeman, E.H. Adelson, *The Design and Use of Steerable Filters*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 13(9), pp. 891-906, 1991
8. G. Granlund, H. Knutsson: *Signal Processing for Computer Vision*, Kluwer, 1995
9. J.J. Koenderink: *The Structure of Images*, Biological Cybernetics, 50:363-370, 1984
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. P. Kovesi: *Image Features From Phase Congruency*, Videre: A Journal of Computer Vision Research. MIT Press. Volume 1, Number 3, 1999
12. T. Lindeberg: *Scale-Space Theory in Computer Vision*, Kluwer, 1994
13. G. Sicuranza: *Quadratic Filters for Signal Processing*, Proc. of the IEEE, 80(8):1263-1285, 1992
14. J. Weickert, H. Schar: *A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance*, J. Visual Communication and Image Representation, 13(1/2):103-118, 2002