Boundary Characterization within the Wedge-Channel Representation

Ullrich Köthe

Cognitive Systems Group, University of Hamburg
koethe@informatik.uni-hamburg.de

Abstract. Junctions play an important role in motion analysis. Approaches based on the structure tensor have become the standard for junction detection. However, the structure tensor is not able to classify junctions into different types (L, T, Y, X etc.). We propose to solve this problem by the wedge channel representation. It is based on the same computational steps as used for the (anisotropic) structure tensor, but stores results into channel vectors rather than tensors. Due to one-sided channel smoothing, these channel vectors not only represent edge orientation (as existing channel approaches do) but edge direction. Thus junctions cannot only be detected, but also fully characterized.

1 Introduction

Feature-based algorithms constitute a large method class for various aspects of image analysis, including object recognition, motion estimation, stereo matching and shape from motion/stereo. The correct detection and characterization of image features such as edges and corners is crucial for these methods to produce accurate results or to succeed at all. In this paper we are interested in generic feature detection methods, i.e. methods that are not bound to a specific application and do not require prior (global) knowledge about the expected objects as provided by geometric shape models, eigenfaces and so on. Under the generic paradigm, features are detected in a bottom-up fashion, and the amount of information extracted from the original image data – without the help of the high-level system – should be maximized. Note that we do not question the usefulness of top-down image analysis. Our goal is rather the independent optimization of bottom-up processing so that the high-level system can start from intermediate data of the best possible quality.

In the context of motion analysis, corners and junctions are of utmost importance because they often arise from 3D features (object corners) that are stable under perspective projection and motion, or indicate important projection phenomena such as occlusion. Accurate junction characterization improves the robustness of feature tracking and correspondence estimation and aids in the correct interpretation of the measured flow fields. Over the years, the ability of local bottom-up operators to extract high-quality junction information has steadily improved. One early approach is to apply an edge operator to the image

and then detect corners and junctions as edge crossings in the resulting symbolic edge representation. However, this method is problematic because edge models break down near junctions, and the propagation of these errors leads to inaccurate, missed or hallucinated junctions which have to be repaired by high-level assumptions or heuristics.

The introduction of the structure tensor [1,5,6] extended the boundary model to include 2-dimensional features explicitly by integrating gradient information over a neighborhood. Recently, it was observed that the accuracy of the structure tensor can be improved by moving from linear to anisotropic integration [9], e.g. with an hourglass filter [7]: When the filter only smoothes along edges, nearby structures do not interfere with each other except at junctions where it is desired. Thus, the anisotropic structure tensor has effectively a higher resolution. However, it does not solve another problem: second order 2×2 tensors can only represent a single independent orientation (the other one is always at $90°$ of the first). Information at which angles the edges meet at a junction is therefore unavailable. One can distinguish intrinsically 1- and 2-dimensional locations, but classification of different junction types is impossible.

Independently of these "mainstream" methods a number of dedicated junction characterization algorithms have been proposed, see [8,12] for surveys. They build upon one-sided filters that determine whether there is an edge in a particular sector of the neighborhood of the given point. The complete junction characteristic can than be interpolated from the responses of a family of filters covering the entire neighborhood. However, these methods are problematic on two reasons: First, they use complicated filters that cannot be applied at fine scales due to aliasing in the sampled filter coefficients. Systematic investigations of their robustness don't seem to exist. Second, a difficult integration problem is posed when unrelated approaches are used for edge detection and junction characterization: The results don't fit exactly together, and inconsistencies in the integrated boundary representation are unavoidable.

In this paper, we propose a junction characterization method that directly generalizes the established structure tensor framework by means of the channel representation. The channel representation [10,4,2] is a carefully designed method for the discretization of continuous quantities (orientation in our case) with the goal that important properties of the original data distribution (e.g. the mean and mode) can be recovered accurately from the channel weights. We keep the idea of anisotropic integration of the gradient map, but instead of collecting the integrated data into a tensor, we store them in orientation channels. Depending on the number of channels, we can determine several independent orientations as long as they do not fall into a single or adjacent channels. Unlike previous work with orientation channels, we use one sided channel smoothing filters, so that we can keep track from which *direction* an edge enters the junction. A similar idea with slightly different filters was also proposed in [11]. This new *wedge channel representation* extends existing work by allowing us to precisely determine the degree of a junction, and distinguish various junction types even if they have the same degree (e.g. T and Y-junctions for degree 3).

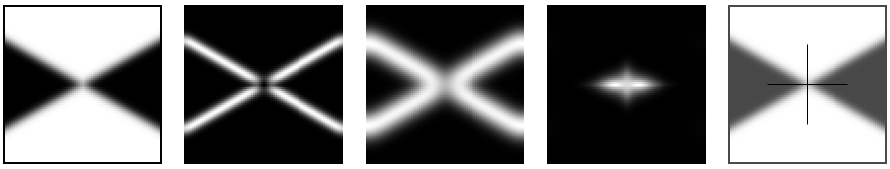

Fig. 1. From left to right: Original image; gradient squared magnitude; trace of structure tensor; small eigenvalue of structure tensor; eigenvector orientations.

2 Boundary Characterization with the Structure Tensor

Given an image $f(x, y)$, the structure tensor is based on the gradient of f, which is usually calculated by means of Gaussian derivative filters:

$$f_x = g_{x,\sigma} \star f, \qquad f_y = g_{y,\sigma} \star f \tag{1}$$

where $\star$ denotes convolution, and $g_{x,\sigma}, g_{y,\sigma}$ are the derivatives in x- and y-direction of a Gaussian with standard deviation σ:

$$g_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

The gradient tensor $\mathbf{Q}$ is obtained by calculating, at each point of the image, the Cartesian product of the gradient vector $(f_x, f_y)^T$ with itself.

$$\mathbf{Q} = \begin{pmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{pmatrix} \tag{3}$$

Spatial averaging of the entries of this tensor, usually with a Gaussian filter, then leads to the structure tensor [1,5,6]:

$$\mathbf{S}_{ij} = g_{\sigma'} \star \mathbf{Q}_{ij} \qquad (i, j \in \{1, 2\}) \tag{4}$$

The trace of the structure tensor (which is identical to the spatial average of the gradient squared magnitude) serves as a boundary indicator, whereas the gradient itself is only an edge indicator and gives no response at some junction types. Spatial maxima of the small eigenvalue of the structure tensor indicate junctions (see fig. 1). However, when two edges run close to each other, linear integration smears these edges into each other. This is desirable for edges that cross at a junction, but at other locations, e.g. when the edges run in parallel, it is not. The problem can be solved by replacing linear smoothing with an anisotropic filter such as the hourglass proposed in [7]. The hourglass kernel is defined as a polar separable function, where the radial part is a Gaussian, but the angular part modulates the Gaussian so that it becomes zero perpendicular to the local edge direction ϕ:

$$h_{\sigma',\rho}(r, \psi, \phi) = \begin{cases} \frac{1}{N} & \text{if } r = 0 \\ \frac{1}{N}\, e^{-\frac{r^2}{2\sigma'^2}}\, e^{-\frac{\tan(\psi-\phi)^2}{2\rho^2}} & \text{otherwise} \end{cases} \tag{5}$$

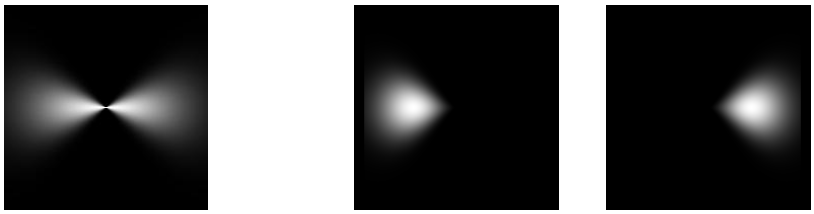

Fig. 2. Left: Hourglass like filter according to (5), with $\rho = 0.4$ and $\phi = 0$; Right: Hourglass filter multiplied with r^2 and split into two halves h_- and h_+.

where ρ determines the strength of orientedness (the hourglass opening angle), and N is a normalization constant that makes the kernel integrate to unity, see fig. 2 left. At every point in the image, this kernel is rotated according to the local edge orientation defined by $\phi(x, y)$, so that smoothing only occurs along the edge. The anisotropic structure tensor $\mathbf{T}$ is obtained by applying the hourglass to the gradient tensor $\mathbf{Q}$:

$$\mathbf{T}_{ij}(x,y) = \sum_{x',y'} h_{\sigma',\rho}(r, \psi, \phi(x',y'))\, \mathbf{Q}_{ij}(x',y') \tag{6}$$

$$\text{with } r = \sqrt{(x-x')^2 + (y-y')^2} \quad \text{and} \quad \psi = \tan^{-1}\left(\frac{y-y'}{x-x'}\right)$$

and $\phi(x', y')$ is the edge direction at the point (x', y'). At junctions, this tensor equals the linear structure tensor, but it removes the undesired behavior at other locations, see fig. 3. With $\rho = 0.4$, the hourglass has an opening angle of about $22.5°$ and can be applied at small scales σ' without significant angular aliasing.

While the tensors $\mathbf{S}$ and $\mathbf{T}$ are good junction detectors, they cannot be used for junction characterization. The information obtained from the eigenvectors only describes the orientation of isolated edges correctly. At corners and junctions one eigenvector typically points into the most salient region, and the other is at $90°$ of the first, fig. 1 right. This is a fundamental limitation of second order tensors. More detailed orientation information is in principle available in the anisotropic intergration framework, because the exact edge orientation $\phi(x', y')$ is fed into the hourglass filter. This information is lost when the gradient tensors are added to form the structure tensor. Therefore, we keep the idea of hourglass filtering, but change how the collected data are represented afterwards: we replace the tensor by a *channel representation*.

3 The Channel Representation

The channel representation was developed by [10,4,2] as a tool for estimating the local distribution of certain measurements. It can be considered as a generalization of histograms. Like the latter, a channel representation consists of bins

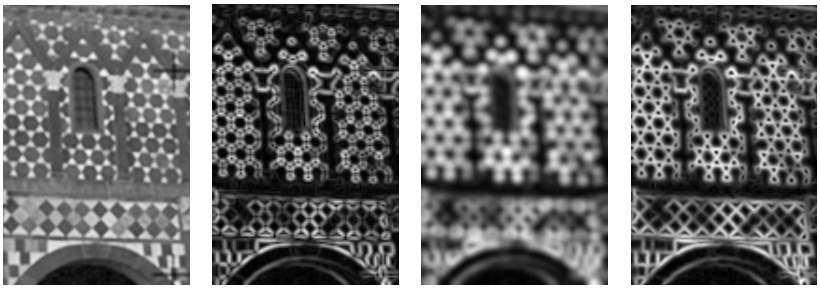

Fig. 3. From left to right: Original image; gradient squared magnitude; trace of structure tensor; trace of anisotropic structure tensor.

(here called channels, hence the name) whose weights encode the probability, confidence or frequency of a specific range of the measured quantity. But unlike histograms, where bins are separate, i.e. are only influenced by values that fall in between each bin's borders, channels *overlap*. The amount of overlap is determined by the channel encoding function Θ which can be understood as a smoothing filter that distributes every exact measurement over a certain range of channels. A channel representation can be obtained from either a continuous function or a finite set of samples:

$$c_k = \int \Theta(t-k) f(t)\, dt \tag{7}$$

$$c_k = \sum_i \Theta(t_i - k) f_i \tag{8}$$

where c_k is the weight of channel k, $f(t)$ is a continuous weighting function for value t, whereas f_i is the discrete weight of the i^{th} sample taken at point t_i. The definitions can be generalized to multiple dimensions in the obvious way.

An example for the continuous variety of the channel representation is the image itself. Here, $f(\mathbf{t})$ is the analog image that would be produced by an ideal pinhole camera, and c_{ij} is the discrete image we observe. The channel encoding function Θ is in this case formed by the combined effect of camera point spread function, defocus blur, and sensor response. In this paper we will be interested in discrete channel representations of local orientation: At every pixel, we store a 1D channel vector that encodes the orientation and strength of edges in a window around the pixel (more details below).

A representative value of the measured quantity can be reconstructed (decoded) from the channel representation in several ways. A global estimate is given by the mean over all channels. However, this is often not a very useful data description, as it smears all information together, regardless of whether the mean is *typical* or remote from any actual measurement. This is similar to linear smoothing of a checker board image, whose average intensity is gray, although gray did never occur in the original. More typical representatives can be obtained by switching to *robust* channel decoding [2]. This is best achieved by

looking at the *mode* (the global maximum) or the set of all maxima of the channel histogram. These values are always near measurements that frequently occured and thus tell more about what actually happened in a given set of samples.

Robust decoding requires an error norm that determines how many neighboring channels are considered in the estimation of each maximum. This can be understood as the reconstruction of a continuous weighting function from the discrete channel weights by means of a convolution with a continuous decoding function Ψ, followed by an analytic calculation of the global or local maxima of the reconstructed function. The functions Θ and Ψ must fulfill several criteria. Most importantly, it is required that both the global and robust reconstruction methods exactly reproduce the value t of a single encoded measurement. Second, the encoding function should be a partitioning of unity, i.e. the sum of a set of Θ-functions placed at the channel centers should be identically one in the entire domain of possible t-values. Likewise, the function Ψ should integrate to unity (this requirement was not posed in [2]). These two requirements ensure that the total confidence (weight) is preserved under channel encoding and decoding. Finally, both Θ and Ψ should be simple functions with small support so that computations can be efficiently executed.

In this paper we follow the *spline* channel model proposed by [2]. In this model Θ is the second order B-spline:

$$\Theta(t) = b_2(t) = \begin{cases} \frac{3}{4} - t^2 & |t| < \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - |t|)^2 & \frac{1}{2} \leq |t| < \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Robust decoding consists of a sequence of two filters. First, a first order recursive filter is applied with the transfer function

$$R(u) = \frac{4}{3 + \cos(u)} \tag{10}$$

where u is the frequency coordinate. Since the orientation domain is cyclic, the filter response should be computed in the Fourier domain to avoid border errors. Recursive filtering is followed by a convolution with Ψ defined by

$$\Psi(t) = \begin{cases} \frac{23}{48} - \frac{1}{4}\, t^2 & |t| < \frac{1}{2} \\ \frac{5-2|t|}{12}\, b_2(|t| - 1) + \frac{1}{2}\, b_2(t) & \frac{1}{2} \leq |t| < \frac{5}{2} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The local maxima of the resulting continuous function are the robust estimates of the dominant orientations. Due to our normalization requirement for Ψ this differs slightly from [2], but it gives the same maxima. In practice, the convolution with Ψ needs not be executed, because the local maxima can be found directly by solving a quadratic equation in each channel interval, and the height of each maximum is given by a cubic function. Using these definitions, we achieve a reasonable channel overlap while computations remain relatively simple.

Encoding and decoding alone would not make the channel representation a worthwhile concept. Its real significance stems from the fact that we can perform linear or anisotropic *channel smoothing* before decoding. If one channel histogram is attached to every pixel, channel smothing is done by interpreting corresponding channel values c_k across all pixels as one image that can be smoothed independently of all other channels. Channel smoothing has an important advantage over direct smoothing of the original measurements: Only values close to the channel center are coded in every channel image. Therefore, only values that likely result from the same distribution are averaged. Consider again the checker board, this time with added noise. Then linear averaging would still give us a useless gray, whereas in a channel representation noisy black and white samples would be averaged separately, resulting in two representative averages for black and white. Channel smoothing can be performed both in a linear and an anisotropic way [3].

4 Channel Coding of the Gradient Orientation

In every pixel we have a gradient squared magnitude $m(x,y) = |\nabla f(x,y)|^2$ and an edge direction $\phi(x,y)$ perpendicular to the gradient. The magnitude is interpreted as the confidence of the direction measurement. The angle must first be transformed to the channel domain by a linear mapping. When n channels are available, the mapping is

$$t(\phi) = n\frac{\phi}{\phi_{\max}} \qquad \phi(t) = \phi_{\max}\frac{t}{n} \tag{12}$$

where $\phi_{\max} = \pi$ or $\phi_{\max} = 2\pi$ depending on whether we work with orientation or direction. In this paper, we use orientation and choose $n = 8$, resulting in a channel spacing of $22.5°$. A single gradient measurement is encoded into channel c_k as

$$c_k(x,y) = m(x,y)\ \Theta\big(\Delta(t(\phi(x,y)),k)\big) \tag{13}$$

Since the angular domain is periodic, the difference between t and the channel center k must be taken modulo n: $\Delta(t,k) = \min(|t-k|, |t-k+n|, |t-k-n|)$ (note that the channel index equals the channel center here).

When channel encoding is followed by linear channel smoothing, the result is similar to the linearly integrated structure tensor: The sum of all channel weights at a given pixel is the local boundary strength analogous to the tensor trace. In order to distinguish 1D and 2D locations it is beneficial to transform the channel weights into an auxiliary tensor according to:

$$\mathbf{A} = \begin{pmatrix} \sum_k c_k \cos^2(\phi_k) & \sum_k c_k \cos(\phi_k)\sin(\phi_k) \\ \sum_k c_k \cos(\phi_k)\sin(\phi_k) & \sum_k c_k \sin^2(\phi_k) \end{pmatrix} \tag{14}$$

where $\phi_k = k\,\phi_{\max}/n$ is the center angle of channel k. As usual, the tensor's small eigenvalue is large at corners and junctions. At spatial maxima of the

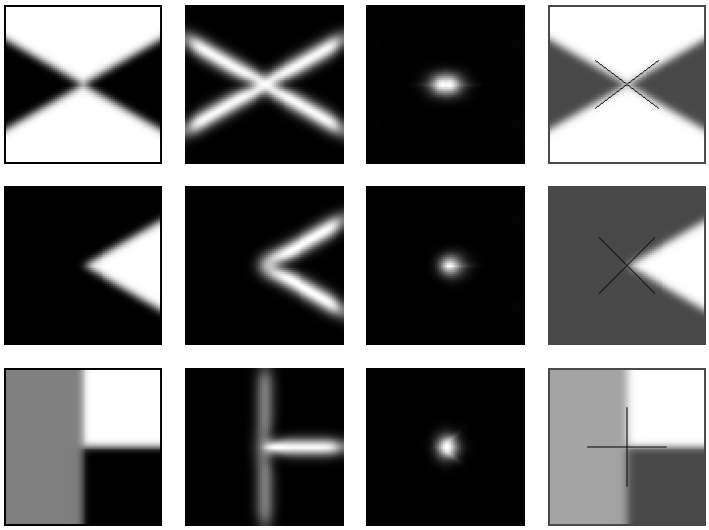

Fig. 4. From left to right: Original image; total channel weight; small eigenvalue of auxiliary tensor (14); edge orientation (all images with anisotropic channel smoothing).

junction strength, we can now do something that was not possible with the structure tensor: we can recover the orientation of the edges that contributed to the junction response by computing (with the function Ψ) at which angles the confidence becomes maximal (however, maxima with confidence below a certain threshold should be dropped as insignificant).

Like the structure tensor, channel encoding can be improved by switching to anisotropic channel smoothing. Here we have two possibilities. First we can apply the standard procedure: We define an anisotropic smoothing filter and apply it in every channel so that the main filter orientation equals the center angle ϕ_k of the channel [3]. However, since the center angle is only an approximation of the encoded edge orientation, smoothing occurs not always exactly along the edge. Therefore, we prefer a different approach here: Since we encoded only a single gradient before channel smoothing, the exact edge orientation is still known. We can thus apply the filter at exactly the correct angle. This brings us back to the hourglass formula (6). But instead of tensor entries, we now propagate magnitude/orientation pairs according to (13):

$$c_k(x,y) = \sum_{x',\,y'} h_{\sigma',\rho}(r,\psi,\phi(x',y'))\, m(x',y')\, \Theta\big(\Delta(t(\phi(x',y')),k)\big) \qquad (15)$$

(r, ψ, ϕ are as in (6)). Since we can apply this formula with an arbitrary number of channels, we have control over the angular resolution of our junction characterization. However, at small scales, channel spacing and hourglass opening angle should not drop below 22.5° in order to avoid angular aliasing. Fig. 4 shows the results of these computations for a number of example configurations.

5 Wedge Channel Coding

Fig. 4 also reveals a principal problem with the approach sketched so far: Since forward and backward propagation of edge information is performed in the same way, the information wether an edge entered the junction from left or right, from top or bottom, is lost. Consequently, we are unable to distinguish a corner (degree 2) from a T-junction (degree 3) or a saddle point (degree 4), because the channel representations have only two maxima in all cases.

We solve this problem by breaking the symmetry of forward and backward propagation. It turns out that a slightly modified hourglass kernel is ideal for this purpose. First, we multiply the hourglass with r^2 (the squared distance from he filter center). This is useful because gradients near a junction center do not contain valid orientation information and their exclusion leads to more accurate orientation estimates. Second, we split the kernel along the axis perpendicular to the edge into two halves (fig. 2 right). This does not introduce a discontinuity because the kernel is zero along this axis. Finally, we double the number of channels, and the first half of the channel vector receives edge contributions coming from angles between 0 and π, whereas the second half takes the contributions from π to 2π. In the kernel, the rule is reversed: we call h_+ the kernel that distributes information downwards (into the first half of the channel vector), and h_- the kernel that distributes upwards (into the second half). The channel smoothing formula (15) must be split accordingly:

$$c_{k<n}(x,y) = \sum_{x'} \sum_{y'>0} h_+(r,\psi,\phi(x',y'))\, m(x',y')\, \Theta\big(\Delta(t(\phi(x',y')),k)\big) \tag{16}$$

$$c_{k\geq n}(x,y) = \sum_{x'} \sum_{y'<0} h_-(r,\psi,\phi(x',y'))\, m(x',y')\, \Theta\big(\Delta(t(\phi(x',y')+\pi),k)\big) \tag{17}$$

Note that ϕ is still taken modulo π (i.e. is the edge orientation), but we use twice as many channels as before ($0 \leq k < 2n$) and set $\phi_{\max} = 2\pi$. The other algorithm steps are mostly uneffected by this change: The boundary strength can still be calculated as the sum of the channel weights, edges and corners can be distinguished by the small eigenvalue of the auxiliary tensor (14), and the orientation of the confidence maxima indicates the direction of the contributing edges. But the number of these maxima is now a true estimate of the junctions' degree. Corners have 2 maxima, whereas saddles have 4. It is even possible to distinguish different kinds of degree 3 junctions: a T-junction has two opposing maxima, but a Y-junction hasn't. A possible check for this classification is as follows: first calculate the number of maxima from the 2π channel representation. Then create an auxiliary channel vector ranging from 0 to π whose weights are the sum of the weights of opposite channel pairs from the original channels, and determine the number of its maxima. If this number is lower, one or more edges did not end at the junction, but crossed it.

Fig. 5 show some results obtained with the wedge channel representation. The discrepancy between the recovered orientations and the ground truth is below 1°. To obtain such a high accuracy, the hourglass kernel must be large

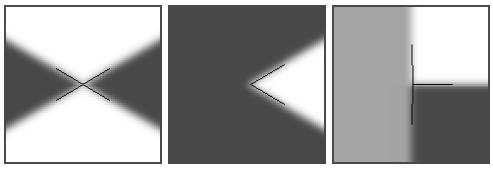

Fig. 5. Edge direction for the same images as in fig. 4, calculated from the wedge channel representation.

enough: When the gradient image has scale σ_g (as determined by the combined effect of the camera point spread function and the gradient filter), the scale of the hourglass should be $2\sigma_g ... 3\sigma_g$, depending on the junction configuration (T-junctions seem to require scales near $3\sigma_g$). On the other hand, smaller kernels may be necessary in order to prevent neighboring junctions from interfering. Then the method still works, albeit with reduced accuracy. Finally, it should be noted that there is no need to compute the rather expensive wedge channel representation at every point. It suffices to first detect corners and junctions using the eigenvalues of the anisotropic structure tensor, and perform the more expensive analysis only there. The high similarity between the kernels involved ensures that results remain consistent.

6 Results and Conclusions

We applied the wedge channel method to a number of real images (figs. 6 and 7). It can be seen that most corners are found correctly by the wedge channel representation, with few false positives. In a few cases, junctions give rise to a multi-modal response. The main advantage over the traditional structure tensor approach is the ability to perform detailed junction characterization. The estimated directions of the edges starting at each junctions have been marked. They are correct in most cases, although sometimes one edge is missing, or there is an extra response. Geometric accuracy is not always satisfying and needs further investigation. It should be noted, however, that the first and third images shown are not as easy to analyse as it may look at first sight: The tiled wall has very low resolution (diameter of the smallest tiles is 3 pixels), and the blocks image is relatively noisy.

Nevertheless, I believe that the wedge channel representation has a great potential because it directly generalizes well established edge and junction detection methods. It performs essentially the same computational steps as used in the anisotropic structure tensor calculations, only the final result is stored in a different way in order to keep as much information as possible: Depending on the number of channels, several independent edge directions can be recovered from the channels representation, in contrast to only one from a tensor. In contrast to existing orientation channel work, the wedge channel representation measures edge direction, so that the correct junction degree can be estimated. This is not possible with orientation channels, let alone the structure tensor. Due to

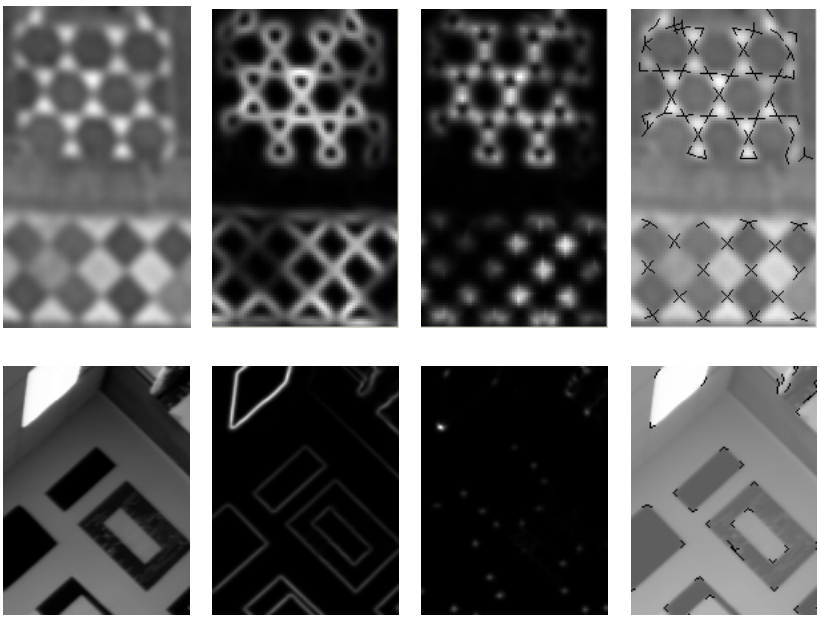

Fig. 6. From left to right: Original; boundary strength (total wedge channel weight); junction strength (small eigenvalue of auxiliary tensor); edge directions around the detected junctions: lines of length 3 pixels were drawn into the directions found.

the relatively simple filter shapes (Gaussian gradient and anisotropic hourglass masks), the new approach can be applied at fine scales. We expect that results can be further improved when the various parts of the algorithm are tuned to optimally fit together.

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, P.-E. Forssen, H. Scharr: *Efficient Robust Smoothing of Low-Level Signal Features*, Linköping University, Computer Vision Laboratory, Technical Report LiTH-ISY-R-2619, 2004
3. M. Felsberg, G. Granlund: *Unisotropic Channel Filtering*, in: Proc. 13th Scandinavian Conf. on Image Analysis, pp. 755-762, Springer LNCS 2749, 2003
4. G. Granlund: *An Associative Perception-Action Structure Using a Localized Space Invariant Information Representation*, in: Proc. Intl. Workshop on Algebraic Frames for the Perception-Action Cycle, Springer LNCS, 2000
5. W. Förstner: *A Feature Based Correspondence Algorithm for Image Matching*, Intl. Arch. of Photogrammetry and Remote Sensing, vol. 26, pp. 150-166, 1986
6. C.G. Harris, M.J. Stevens: *A Combined Corner and Edge Detector*, Proc. of 4th Alvey Vision Conference, 1988

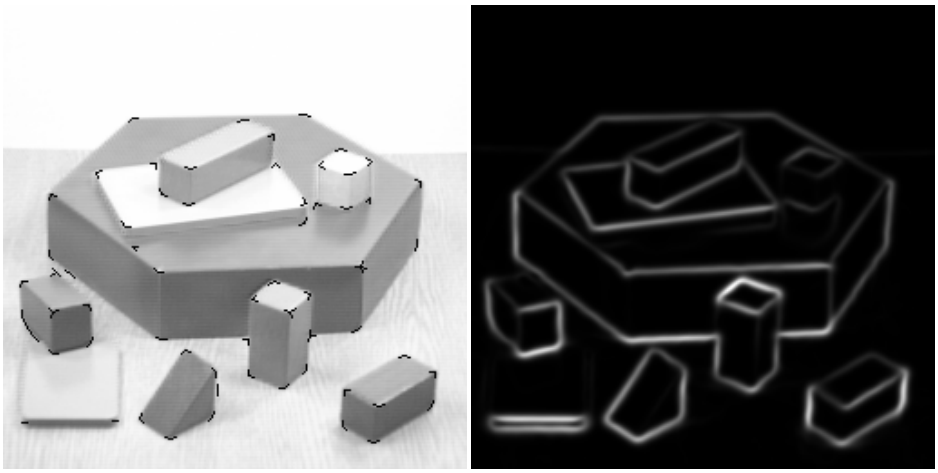

Fig. 7. Left: Original with junctions and edge directions; boundary strength.

7. U. Köthe: *Edge and Junction Detection with an Improved Structure Tensor*, in: Michaelis et al. (Eds.): Pattern Recognition, Proc. 25th DAGM Symposium, Springer, 2003
8. M. Michaelis: *Low Level Image Processing Using Steerable Filters*, PhD Thesis, GSF-BEricht 30/95, Christian-ALbrechts-Universität Kiel, 1995
9. H.-H. Nagel, A. Gehrke: *Spatiotemporal Adaptive Estimation and Segmentation of OF-fields*, in: ECCV 98, Proc. 5th European Conf. on Computer Vision, vol. II, LNCS 1407, pp. 86-102, Springer, 1998
10. H.P. Snippe, J.J. Koenderink: *Discrimination Thresholds for Channel-Coded Systems*, Biological Cybernetics, vol. 66, pp. 543-551, 1992
11. H. Spies, B. Johansson: *Directional Channel Representation for Multiple Line-Endings and Intensity Levels*, in: ICIP 03, Proc. IEEE Intl. Conf. on Image Processing, 2003
12. W. Yu: *Local Orientation Analysis in Images and Image Sequences Using Sterrable Filters*, PhD Thesis, Bericht Nr. 2012, Christian-ALbrechts-Universität Kiel, 2000