



High Accuracy Feature Detection for Camera Calibration: A Multi-Steerable Approach

Matthias Mühlich and Til Aach

Institute of Imaging and Computer Vision
RWTH Aachen University, 52056 Aachen, Germany
tel: +49 241 80 27860, fax: +49 241 80 22200
web: www.lfb.rwth-aachen.de

in: DAGM07: 29th Annual Symposium of the German Association for Pattern Recognition. See also
BIB_TE_X entry below.

BIB_TE_X:

```
@inproceedings{MUE07c,  
  author    = {Matthias M{"u}hlich and Til Aach},  
  title     = {High Accuracy Feature Detection for Camera Calibration: A Multi-Steerable Approach},  
  booktitle = {DAGM07: 29th Annual Symposium of the German Association for Pattern Recognition},  
  publisher = {Springer},  
  series    = {LNCS},  
  number    = {???},  
  pages     = {???--???},  
  month     = {Sept.\ 12--14},  
  year      = {2007},  
  note      = {to appear}  
}
```

© 2007 Springer-Verlag.

See also LNCS-Homepage: <http://www.springeronline.com/lncs>

High Accuracy Feature Detection for Camera Calibration: A Multi-Steerable Approach

Matthias Mühlich and Til Aach

Lehrstuhl für Bildverarbeitung, RWTH Aachen University,
52056 Aachen, Germany
{matthias.muehlich,til.aach}@lfb.rwth-aachen.de

Abstract. We describe a technique to detect and localize features on checkerboard calibration charts with high accuracy. Our approach is based on a model representing the sought features by a multiplicative combination of two edge functions, which, to allow for perspective distortions, can be arbitrarily oriented.

First, candidate regions are identified by an eigenvalue analysis of the structure tensor. Within these regions, the sought checkerboard features are then detected by matched filtering. To efficiently account for the double-oriented nature of the sought features, we develop an extended version of steerable filters, viz., multi-steerable filters. The design of our filters is carried out by a Fourier series approximation. Multi-steerable filtering provides both the unknown orientations and the positions of the checkerboard features, the latter with pixel accuracy. In the last step, the feature positions are refined to subpixel accuracy by fitting a paraboloid. Rigorous comparisons show that our approach outperforms existing feature localization algorithms by a factor of about three.

1 Introduction

Accurate camera calibration is a basic prerequisite for many image processing and computer vision algorithms. Jean-Yves Bouguet’s camera calibration toolbox (http://www.vision.caltech.edu/bouguetj/calib_doc/) [1, 2] has become a de-facto standard for this problem, mainly for three reasons: simple usage, high estimation quality and free availability as Matlab and C code. Additionally, its C version is part of the OpenCV library distributed by Intel.

Camera calibration from a set of M input images can be divided into two steps: first extract some feature points, for instance on a checkerboard grid, and then use these points to estimate internal and external camera parameters, see fig. 1. The second step is the actual calibration, where the camera model parameters are estimated. Recent papers focus on this part, e.g., by introducing advanced distortion models [3]. Here, however, we will improve the *first* step of the complete procedure. Evidently, the estimation quality of any calibration scheme can only be as accurate as the quality of the feature points which are used as input for the parameter estimation step.

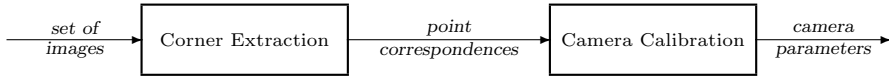


Fig. 1: Camera calibration from images is a two-step procedure: first, a set of point correspondences between world and image coordinates has to be extracted in the input images and then, these points are the input for a non-linear optimization of the sought camera parameters.

Any type of camera calibration requires some visual features on a calibration object which can be detected in its images as robustly and accurately as possible. Common choices for feature points are centers of gravity (of circles or squares), intersections of a line grid, corners, or patterns like the checkerboard pattern. However, centers of gravity are not invariant to perspective distortions, line-based approaches can lead to problems due to varying line thicknesses, and corner-based approaches suffer from biased estimates (see e.g. [4] for a discussion of fitting parametric models to corners). Checkerboard-based approaches, on the contrary, avoid localization bias due to their symmetry and have therefore become the most widely used choice for (2D) calibration patterns recently.

Bouguet’s toolbox uses a sub-pixel extension of the famous Harris corner detector [5] which finds prominent regions in the following way: Let $f(x, y)$ denote an image signal and let $\mathbf{g} = \nabla f$ denote its gradient. Then

$$\mathbf{S} = \int_{\Omega} \mathbf{g}\mathbf{g}^T, \quad (1)$$

where Ω is an area of local integration, defines the so-called *structure tensor* [6]. Its two eigenvalues λ_1 and λ_2 characterize the image region centered at $\mathbf{x} = (x, y)^T$: two small eigenvalues indicate a homogeneous regions, one small and one large eigenvalue indicate a linear feature and two large values finally denote features which usually allow exact localization. If the image is known to contain features such as corners or checkerboard crossings, one can safely assume that the corresponding regions can be found by looking for two large eigenvalues. Harris therefore introduced the following measurement for corner strength:

$$M_c = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2. \quad (2)$$

The tuning parameter κ penalizes regions where the sum but not the product is high, i.e., it penalizes lines or edges. Reasonable values are in the range 0.1 ± 0.05 .

Other advanced general feature detectors exist, for instance in the SIFT [7] algorithm, but for camera calibration, their use remains limited. In contrast to applications like motion detection, tracking, panorama stitching, 3D modelling or object recognition, we do not have to consider general objects – there is no need to extract every possible bit of information regardless of form or scale. Instead, we have a pretty good model of what the image of a *known* calibration pattern should look like. Our novel approach to corner detection is therefore based on designing filters specifically for images of the widely used checkerboard calibration patterns.

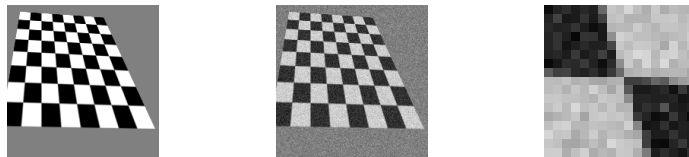


Fig. 2: A synthetic checkerboard image (left), the same image with added Gaussian noise (center; SNR = 10 dB), and a region of interest around a crossing (right). It can be seen that checkerboard crossings are characterized by *two* independent edges.

An existing signal model directly calls for a correlation-based feature detection approach, but what exactly is our signal model? Due to perspective distortions, images of checkerboard crossings are black-and-white patterns characterized by *two* independently varying local orientations, see fig. 2, which generate a large family of possible patterns. For any correlation-based approach, this is problematic: Assuming that each angle shall be sampled in 5 degree steps, we would need $72^2/2 = 2592$ checkerboard templates (the 2 in the denominator is due to the symmetry of the checkerboard pattern). The resulting computational load would be absolutely prohibitive.

As a solution, we present a novel feature detection approach which is based on an extension of the concept of steerable filters [8] to *multi-steerability*. Steerable filters have been used in [9] to detect edges and lines. Unfortunately, these linear feature never allow an exact feature localization; only the component orthogonal to the orientation direction can be determined (*aperture problem*, [10]). Therefore, steerable filters have not been used in exact feature localization yet. In this paper, we will show how to extend the steerable filter concept to a multi-steerable detector which allows high precision feature localization.

2 Design of Double-Steerable Filters for Modelling Checkerboard Patterns

Let $f(\mathbf{x})$ denote an image within which we seek a feature which can be modelled as a template $f_0(\mathbf{x})$. Then filtering the image with a filter $h(\mathbf{x}) = f_0(-\mathbf{x})$ yields an output image measuring how strong the feature is present at each location \mathbf{x} . This principle is known as *matched filter* [11]. Its application to the detection of a *family* of features is, in general, computationally inefficient, but for one special class of filters, namely rotated versions of some given template, the *steerable filter* approach introduced by Freeman and Adelson in [8] offers a convenient solution: by limiting the class of possible (unrotated) templates to those templates which can be represented in polar coordinates in the form

$$h(r, \phi) = \sum_{p=-P}^P a_p(r) \exp(jp\phi), \quad (3)$$

one can represent any rotated version of $h(r, \phi)$ as a linear combination of ν

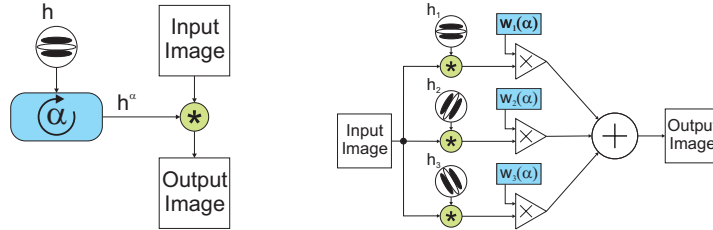


Fig. 3: The steerable filter concept: applying differently rotated filters for arbitrary rotation angles (left) is computationally expensive, while limiting oneself to a class of steerable filters allows a very fast implementation: compute some weights and sum up precomputed filter results.

base filters, where the minimum number for ν is given as the number of non-zero Fourier coefficients in (3). (Note that, in slight abuse of notation, we will always denote an image template as h , regardless whether it is represented in Cartesian or polar coordinates.) Following the notation of Freeman & Adelson and others, let us define a rotation operator: $h^\alpha(r, \phi) = h(r, \phi - \alpha)$. Different variants for designing steerable filters exist, but they all have in common that rotation can be expressed as linear combination of a set of base templates:

$$h^\alpha(r, \phi) = \sum_{a=1}^{\nu} w_a(\alpha) h_a(r, \phi). \quad (4)$$

Here, h_a denote the set of base filters. Evidently, the whole dependency on the steering angle is encapsulated in the weight coefficients w_a . The linearity of steerable filters allows to exchange the order of filtering and summation, see fig. 3, thus allowing to precompute a set of filtered images and obtain the correlation between image and template for any given position and angle by a weighted sum of filtered images. Hence, the computational load for correlation-based feature detection is reduced considerably.

In [9], Jacob and Unser applied this *rotated matched filter* approach to the detection of edges and lines in images. Unfortunately, steerable filters are limited to features which are characterized by a *single* steering parameter, viz., the orientation angle of the linear feature. Perspectively distorted checkerboard patterns, however, are characterized by *two* independently varying orientations.

The key idea of our approach now is the following: *can we combine two edges in such a way that they represent a checkerboard and, furthermore, the result is steerable again – but now with two steering angles?* In mathematical form, this can be expressed as

$$h_{\text{check}}^{\alpha, \beta} = h_{\text{edge}}^\alpha \circ h_{\text{edge}}^\beta \quad (5)$$

where \circ is some operator, and we now have to examine whether we can find a mathematical function that fulfills this requirement. Evidently, the sought operator must work for every point in the template individually, i.e.,

$$h_{\text{out}} = h_1 \circ h_2 \quad \Leftrightarrow \quad h_{\text{out}}(\mathbf{x}) = h_1(\mathbf{x}) \circ h_2(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega \quad (6)$$



Fig. 4: Creation of a checkerboard pattern as product of two individually rotated idealized edges. If black corresponds to -1 and white to 1 , this construction principle does not only hold for $\alpha = 0^\circ$ and $\beta = 90^\circ$ (left), but also for *arbitrary* angles like $\alpha = 20^\circ$ and $\beta = 130^\circ$ (right).

where Ω is the size of the templates. The graphical representation in fig. 4 visualizes that the desired steering properties automatically follow if we ‘only’ find a mathematical representation for four equations:

$$\begin{aligned} \text{white} \circ \text{white} &= \text{white} & \text{white} \circ \text{black} &= \text{black} \\ \text{black} \circ \text{white} &= \text{black} & \text{black} \circ \text{black} &= \text{white} . \end{aligned}$$

A solution is easily found by identifying white with 1 , black with -1 , and \circ with (point-by-point) multiplication. Note also that these four equations show that the sought operator *must* be non-linear. Having defined the scaling of black and white, we now multiply two steerable filters:

$$h^\alpha(r, \phi) \cdot h^\beta(r, \phi) = \sum_{a=1}^{\nu} \sum_{b=1}^{\nu} \underbrace{w_a(\alpha)w_b(\beta)}_{w_{a,b}^*(\alpha,\beta)} \underbrace{h_a(r, \phi) \cdot h_b(r, \phi)}_{h_{a,b}^*(r,\phi)} . \quad (7)$$

The result can again be represented as a linear combination of base functions $h_{a,b}^*$ which can be computed as point-by-point products of the base functions for the standard steerable filter. In a similar way, the new weight coefficients $w_{a,b}^*$ are found as products of the individual weights; therefore, they now depend on *two* angles, i.e., we have thus introduced a novel *double-steerable* filter. Extension to *multi-steerability* is straightforward.

Generating checkerboard patterns with two arbitrary orientations from (5) now implies replacing idealized edges with *approximated steerable edge functions*. Different approaches for this problem exist: Jacob and Unser [9] used a linear combination of derivatives of the Gaussian function; this has the big advantage of always yielding Cartesian-separable filters. Other authors [12] are interested in *phase-invariant* behavior [13] which means that the filter response should not depend on the signal orthogonal to some orientation; most importantly, lines and edges should lead to the same energy of the filter response. To comply with our needs of multi-steerable edge function approximation, we will propose a novel design concept. This concept is based on the observation that an edge is polar-separable which directly allows a Fourier series expansion. We set

$$h(r, \phi) = q(r) h_{\text{ang}}(\phi) \quad (8)$$

with radial function

$$q(r) = \begin{cases} 1 & r \leq r_{\text{max}} \\ 0 & \text{else} \end{cases} \quad (9)$$

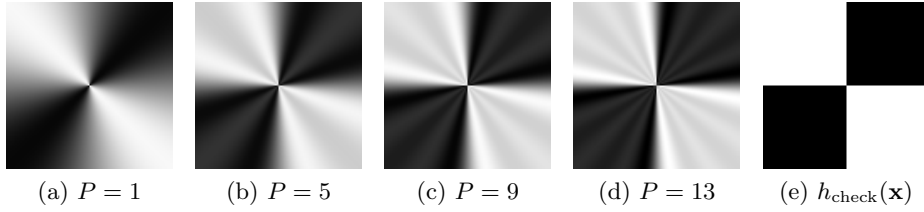


Fig. 5: Checkerboard patterns created by two steerable Fourier expansions of edge-functions for Fourier coefficients $p = 1, 3, \dots, P$. The higher P is chosen, the better the steerable filter approximates the idealized template shown in (e). A radial weighting can be added if desired.

and idealized angular edge function

$$h_{\text{ang}}^{\text{ideal}}(\phi) = \begin{cases} 1 & 0 \leq \phi < \pi \\ -1 & -\pi \leq \phi < 0 \end{cases} . \quad (10)$$

Note that the radial function $q(r)$ can be designed separately since it does not influence steerability; the disc chosen in (9) is mainly used because of its simplicity. The Fourier approximation of (odd) order P to the angular function $h_{\text{ang}}^{\text{ideal}}(\phi)$ rotated by α then is

$$h_{\text{ang}}^{\alpha}(\phi) = \frac{4}{\pi} \sum_{p=1,3,\dots}^P \frac{1}{p} \sin(p(\phi - \alpha)) . \quad (11)$$

Multiplying with radial function (9) yields the (single-)steerable template (8). This template has $P+1$ non-zero Fourier coefficients; therefore, we need $\nu = P+1$ base filters $h_a(r, \phi)$ which can be chosen exactly in the same way as in Freeman's paper: the approximated edge is rotated to $P+1$ equidistant angles in the interval $[0^\circ, 180^\circ)$. Consequently, the weights $w_a(\alpha)$ can be computed as in Freeman's work. One advantage of this Fourier approach (in comparison to other known steerable filter design concepts) is that increasing the approximation order P allows to increase the approximation quality at the price of higher computational complexity. Fig. 5 shows how the pattern converges to the idealized checkerboard when increasing P .

Having defined the checkerboard pattern as a double-steerable filter, we can now define the *principle of multiply rotated matched filtering*: A multi-oriented feature has M independent orientation angles ϕ_1, \dots, ϕ_M and is represented by a template $f_{\text{multi}}^{\phi_1, \dots, \phi_M}(\mathbf{x})$. A measure of how strong this feature is present in an image $f(\mathbf{x})$ at a fixed position \mathbf{x}_0 is $B_{\text{max}}(\mathbf{x}_0)$:

$$B_{\text{max}}(\mathbf{x}_0) = \max_{\phi_1, \dots, \phi_M} f \circledast f_{\text{check}}^{\phi_1, \dots, \phi_M} \Big|_{\mathbf{x}=\mathbf{x}_0} = \max_{\phi_1, \dots, \phi_M} f * h_{\text{check}}^{\phi_1, \dots, \phi_M} \Big|_{\mathbf{x}=\mathbf{x}_0} \quad (12)$$

where \circledast and $*$ denote correlation and convolution, respectively. In analogy to (single) rotated matched filtering, we define $h_{\text{multi}}^{\phi_1, \dots, \phi_M}(\mathbf{x}) = f_{\text{multi}}^{\phi_1, \dots, \phi_M}(-\mathbf{x})$ to be

the corresponding filter kernel and maximize the cross-correlation of the image patch centered at \mathbf{x}_0 and the multiply rotated template, thus assigning estimated orientation angles $\hat{\phi}_1(\mathbf{x}_0), \dots, \hat{\phi}_M(\mathbf{x}_0)$ to each image point \mathbf{x}_0 . In contrast to the detection of linear structures in [9], we are not limited by the aperture problem anymore; this generalized approach also allows *exact* localization of the sought features in addition to the estimation of orientation angles.

So far, this approach allows to detect checkerboard pattern at pixel accuracy. Next, we discuss how to extend the approach to sub-pixel accuracy, and how to increase computational efficiency.

3 Finding Checkerboard Crossings with Sub-Pixel Accuracy

Our algorithm consists of three steps: we first determine a list of candidate points, then apply our double-steerable filter (DSF) at these candidate positions to estimate the correlation strengths and, third, fit a paraboloid to each local maximum. The apex of this paraboloid is taken as feature location.

Our double-steerable filter makes correlation-based checkerboard crossing detection feasible, but nevertheless, the angle optimization in it remains the costly part of the procedure. To reduce the computational load, we therefore preprocess with the aim of applying the DSF only where crossings are likely to be found. Over the entire image, we compute the standard structure tensor \mathbf{S} , and subsequently only consider those points which are both sufficiently textured and do not represent linear structures, i.e., where the structure tensor exhibits *two* large eigenvalues. This can be tested using trace and determinant only, i.e., without computing eigenvalues:

$$\text{tr } \mathbf{S} > t_1 \quad \text{and} \quad \frac{\det \mathbf{S}}{\text{tr } \mathbf{S}} > t_2 . \quad (13)$$

Usually at most a few percent of all pixels qualify as candidate points, unless low resolution images or images with many checkerboard tiles are used. Only for the image points fulfilling these criteria, we compute the double-orientation structure tensor [14] (occluding model) and solve for two orientations. For every candidate point, these two orientations are then used as initial values for a Levenberg-Marquardt optimization of the two DSF angles α and β .

Having found the best fitting double steerable filter, we find the local maxima of the correlation and fit a paraboloid to the 9 correlation values in a 3×3 -neighborhood around each local maximum. Its apex is taken as the final feature location. If not all 8 neighbors of a maximum at pixel resolution were classified as candidates before (unlikely, but it can happen), then some values are missing in the paraboloid fitting step. In such rare cases, the DSF is applied to the missing pixels before carrying out the sub-pixel fitting step.

We do not optimize for angles and crossing position simultaneously because it would require an interpolation step to generate a pseudo-continuous image function. On the other hand, the correlation values around the true sub-pixel

maximum could be approximated extremely well with a second-order Taylor expansion, so fitting a paraboloid to the available correlation values at integer positions near maxima is mathematically justified – and also yields very good results.

4 Results

We tested our algorithm on both synthetic and real data. Experiments on synthetic data with known ground truth enable measuring the root mean square (RMS) error of the localization over varying signal-to-noise ratios (SNRs). This also allows a comparison to the corner finder from Bouguet’s calibration toolbox.

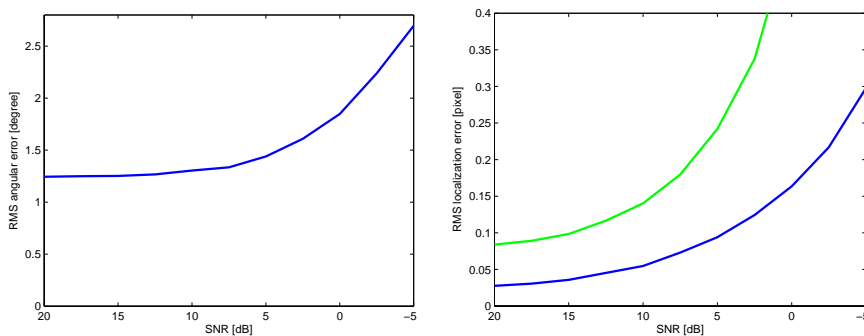


Fig. 6: Left: RMS angular error for our approach over varying SNR. Right: RMS localization error over SNR of our approach (dark) and Bouguet’s corner finder (light).

Our experimental setup was as follows: for SNRs from -5 dB to 20 dB in steps of 2.5 dB using additive white Gaussian noise, we calculated 10 noisy realizations for each of three different synthetic input images, resulting in 30 realizations for each noise level. For each realization, we estimated the locations of the crossings in an 8×8 tiles checkerboard, i.e., 49 inner crossings. Subsequently, the RMS error was computed. Then the average RMS error of the 30 estimation results was plotted against the noise level. The same was done for Bouguet’s corner finder. Here, we even gave Bouguet’s corner finder an unfair advantage: it needs an initial value, which we always initialized with the true optimum. The results of both algorithms are shown in fig. 6. The pixel error of our approach is roughly *one third* of Bouguet’s approach. For low noise levels, our algorithm achieves a localization accuracy of 0.028 pixels (Bouguet: 0.084). The accuracy of the angle estimates was approximately 1.25° for low and medium noise levels. This result was achieved with approximation order $P = 5$.

Apart from its increased accuracy and robustness, another advantage of our approach is that it needs neither initial values of approximate crossing positions nor assumptions such as small lens distortions. The design of our double-steerable

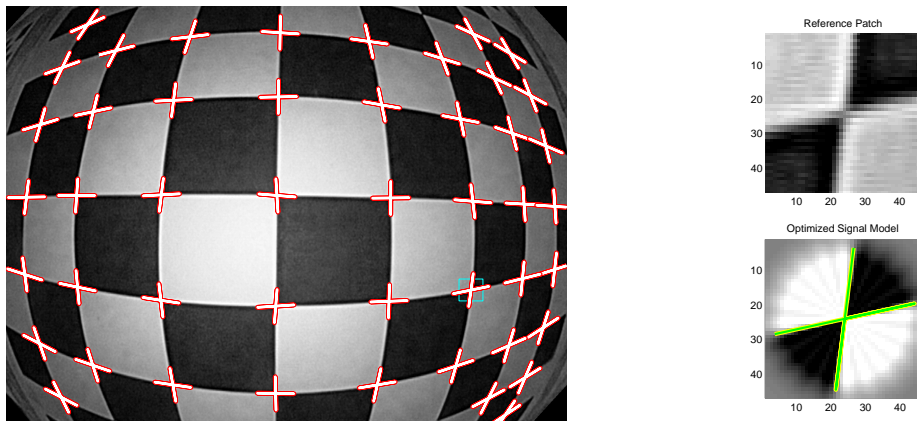


Fig. 7: Left: Estimated crossings in a calibration image taken with an Olympus “CF H-180 AL” endoscope. Right: Image patch and fitted signal model in the marked region of interest. Horizontal orientations are estimated with a slightly increased error because of interlacing artefacts (also visible in image patch).

filters makes searching the whole image for crossings feasible. One example, where the semi-automatic corner finder of Bouguet fails, is the calibration image shown in fig. 7, which was acquired through a wide-angle endoscope; this image exhibits extreme distortions which make (semi-)automatic detection of the crossings difficult. For this 1100×900 pixel image, our approach written in pure Matlab code (i.e., no precompiled C parts) needs approximately one minute on a 3 GHz Dual Pentium computer. This is acceptable for calibration (and definitely less tedious than clicking on all crossings by hand). Note that even the crossings in the strongly distorted regions near the image border were found.

A small bias in the angle estimation can appear if the transition from black to white is not symmetric around the true edge position (overexposure, underexposure, non-linearities). However, due to the symmetry of the checkerboard pattern, it only rotates the estimated edges, but the positions of the crossings, which we are primarily interested in, are *not* affected by this bias.

5 Conclusion and Summary

We have developed a new approach to detect and localize the crossings in checkerboard pattern charts for camera calibration. Its basis is a model characterizing the sought features by multiplicatively combining two edges which are scaled to the range $[-1, 1]$. To allow for perspective distortions, these edges may exhibit arbitrary orientations. The key ingredient of our approach is a multisteerable filter algorithm, which permits efficient matched filtering. The filters are designed using a Fourier series expansion, thus allowing to determine the approximation quality to an ideal edge function by a single parameter. Multisteerable matched filtering then provides not only feature location, but also the orientations, which

are determined by Levenberg-Marquard optimization. In our ongoing work, these angles will be used for, e.g.: (i) checking the plausibility of the detected crossings: the estimated orientations must be compatible to the orientations of neighboring crossings, (ii) exploiting additional information for the optimization of the camera parameters, (iii) speeding-up the detection by a sequential detection of crossings: one (or more) already detected crossings plus their orientations directly tell us where to look for the neighboring crossings.

Our technique exhibits two major advantages in comparison to existing approaches. Firstly, fully automatic corner extraction is possible – as we have shown, even in rather noisy conditions – because the whole image can be processed at low computational cost. Secondly, the availability of a signal model ensures much lower feature localization errors. In comparison to the corner finder in Bouguet’s camera calibration toolbox, the localization RMSE of our approach is lower by a factor of three.

Matlab demonstration code for double-steerable filters can be downloaded from www.lfb.rwth-aachen.de/en/highlights/multi_steerable_filters.html.

References

1. Zhang, Z.: A flexible new technique for camera calibration. In: *IEEE Trans. Pattern Analysis and Machine Intelligence*. **22** (2000) 1330–1334
2. Bouguet, J.Y.: *Visual Methods for Three-Dimensional Modeling*. PhD thesis (1999)
3. Kannala, J., Brandt, S.S.: A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Analysis and Machine Intelligence* **28** (2006) 1335–1340
4. Rohr, K.: Recognizing corners by fitting parametric models. *International Journal of Computer Vision* **9** (1992) 213–230
5. Harris, C., Stephens, M.: A combined corner and edge detector. In: *4th Alvey Vision Conference*. (1988) 147–151
6. Bigün, J., Granlund, G.H., J. Wiklund: Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 775–790
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60** (2004) 91–110
8. Freeman, W., Adelson, E.: The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 891–906
9. Jacob, M., Unser, M.: Design of steerable filters for feature detection using Canny-like criteria. *IEEE Trans. Pattern Analysis and Machine Intelligence* **26** (2004) 1007–1019
10. Jähne, B.: *Digital Image Processing*. 6 edn. Springer (2005)
11. Therrien, C.W.: *Decision, Estimation and Classification: Introduction to Pattern Recognition and Related Topics*. John Wiley and Sons (1989)
12. Simoncelli, E., Farid, H.: Steerable wedge filters. In: *Proc. Int. Conf. Computer Vision*. (1995)
13. Granlund, G., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer (1995)
14. Aach, T., Mota, C., Stuke, I., Mühlich, M., Barth, E.: Analysis of superimposed oriented patterns. *IEEE Trans. Image Processing* **15** (2006) 3690–3700