Accurate Detection and Localization of Checkerboard Corners for Calibration

Alexander Duda
aduda@krakenrobotik.de
Kraken Robotik GmbH
Bremen, Germany

Udo Frese
udo.frese@dfki.de
German Research Center for Artificial Intelligence, Cyber-Physical Systems
Bremen, Germany

Abstract

The calibration of cameras is a crucial step in machine vision and usually relies on an accurate detection and localization of calibration patterns in images. Therefore, checkerboards are often used, allowing precise subpixel estimation of their corners. However, noise in localization generates a proportional noise in the derived model parameters. Therefore, it is important that the localization has a certain robustness against image noise. This is even more important for deteriorated imaging conditions strongly affecting subpixel detectors. This paper presents a new checkerboard corner detector based on a localized Radon transform implemented by large box filters making it robust to low contrast, image noise, and blur while maintaining high subpixel accuracy.

1 Introduction

An optical camera projects light rays onto pixels according to their angle relative to the camera and measures the light's color or intensity. Hence, the image position of a point contains information on its spatial position, which is used in various applications from photogrammetry to object pose detection to image stitching. Beforehand, the mapping from pixel to ray needs to be determined, i.e. the camera must be calibrated [17]. This is usually done by detecting feature points on a physically known calibration object, mostly checkerboard corners, and fitting a camera model to that data. According to first order error propagation, the resulting calibration error is proportional to the error in the detection and hence an accurate checkerboard corner detector is needed.

This is even more relevant for in-field calibration (Fig. 1), where low contrast, image noise and motion blur challenge the detector's accuracy, and due to suboptimal checkerboard placement this error is amplified in the calibration result. To improve calibration in such imperfect conditions a new checkerboard corner detector is presented which is robust to low contrast, image, noise and motion blur, because it is based on box filters suppressing noise, instead of gradient filters amplifying it.

The article is structured as follows. In Section 2, a review over previous work related to checkerboard and corner detectors is given. In Section 3, the proposed algorithm together with its reference implementation is described. In Section 4, the results of synthetic tests and real case experiments are discussed. Following this, the article is concluded in Section 5.

(a) (b) (c)

Figure 1: Sample images drawn from two different datasets used for calibration: (a) calibration in controlled environments; (b-c) outdoor calibration.

2 Related Work

Checkerboard detection has mainly two aspects: The first is to detect the checkerboard as a whole and distinguish it from other image content. This determines, in particular in difficult conditions, whether the image can be used in the calibration or not. The second is finding the accurate corner locations. This determines the accuracy and precision of the calibration and is usually followed by a second step called sub-pixel refinement. Checkerboard corners are well suited for this task because they have strong gradients in all directions.

The Harris operator [10] is the most well known corner detector often used in calibration. Förstner proposed a sub-pixel corner localization method [6] frequently used as refinement and also implemented (`cornerSubPix`) and used in OpenCV's [1] camera calibration pipeline. It exploits, that the image gradient $\nabla f[x,y]$ at a point $\binom{x}{y}$ is perpendicular to the line from $\binom{x}{y}$ to the corner $\binom{c_x}{c_y}$. The corner is then least-squares estimated by

$$\begin{pmatrix}\hat{c}_x\\ \hat{c}_y\end{pmatrix} = \arg\min_{c_x,c_y} \sum_{x,y} \left(\nabla f[x,y]^T \begin{pmatrix} x-c_x \\ y-c_y \end{pmatrix} \right)^2 \tag{1}$$

Douskos et al. [4] use Harris corners with subpixel refinement to find the checkerboard corners, later discarding false positives. They report a precision of 0.1 pixels. To find the overall checkerboard, the median corner is set as origin and two neighbouring points with strong edges to the origin are primary axes. Following this, additional points are successively added.

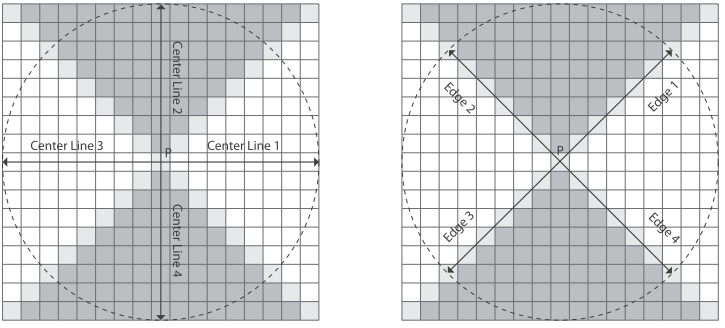
Chu et al. [2] suggest a template matching approach on a binarized image, with a threshold obtained from the image histogram.

OpenCV's [1] checkerboard detector (`findChessboardCorners`) uses an adaptive thresholding and erosion to binarize the image and separate the checkerboard squares into quadrilaterals by contour following [16]. Finally, the checkerboard is detected as a 2D grid of connected quadrilaterals. It follows the Förstner operator (1) for corner refinement.

Rufli *et al.* [14] improved the OpenCV approach for blurred and distorted images by adapting the erosion kernels and by introducing a new linking heuristic for the quadrilaterals.

Ha *et al.* [9] suggest triangles rather than quadrilaterals as primitives for calibration patterns for more precise localization in the fronto-parallel view, under blur and intensity noise. For refinement a third order polynomial is fitted to the intensity surface function at the corner.

Overall, following Sinzinger [15], there are two models for a checkerboard corner: Black and white sectors with centerlines (Fig. 2(a)) and a cross of edges (Fig. 3(b)). An edge



(a) Centerline based junction model (b) Edge based junction model

Figure 2: Junction models

based view leads to algorithms based on gradients which are susceptible to image noise. The proposed algorithm takes a sector based view to avoid this problem. In contrast to previous works, it exploits the point symmetry of checkerboard corners with a localized Radon transform approximated by box filters which is far less affected by image noise than gradient based methods.

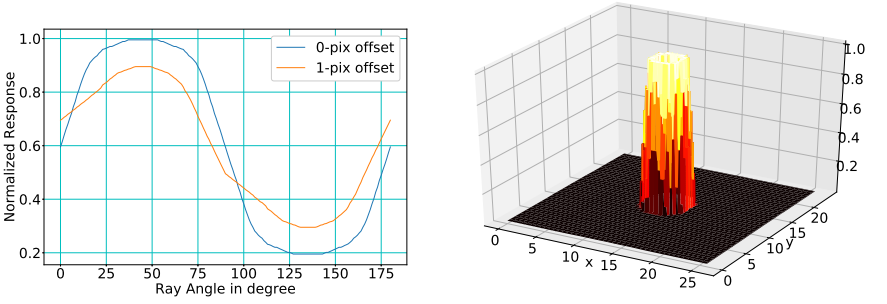
3 Corner Detection and Localization

The task of the corner detector is to identify points in the image which follow the corner model 2(a) and ideally also localize them up to a fraction of a pixel without additional refinement steps. In the case of checkerboard corners, all neighboring regions are point symmetric to the desired point, unlike in most "natural" corners. Therefore, centerlines of opposite regions are viewed as a common centerline.

The underlying concept of the detector is now that the sum of all pixel values along the centerline through the bright areas is higher than the sum of pixel values along the other centerline through the dark areas. In fact, if the two paths used for summation/integration belong to a checkerboard corner are its centerlines, the difference between both integrals will be maximized. In all other cases, the value will be reduced by the amount the paths are touching neighbouring regions and is zero for homogeneous image regions.

Assuming there is a function $f_c[x,y]$ which calculates the integral for all possible centerlines around a given image point $f[x,y]$ and returns the square difference of the maximal integral value and the minimal integral value, checkerboard corners can simply be identified with a min-max search by transforming the image into a response map using this function. This is similar to the Harris detector response map [10]. However, instead of shifting window patches in x and y direction of the image plane they are rotated around the selected point. After local maximums are identified in the response map subpixel position of corresponding corners can be estimated by subpixel peak algorithms. A comparison between different algorithms is given in [5] and [11] , here **Gaussian peak fit** is used.

A closer examination of the requested function $f_c[x,y]$ reveals that it is in fact similar to the Radon transform $Rf[r,\alpha]$ [8] localized to each image point $f[x,y]$ while setting r to zero. This localized Radon transform $Rf_{local}[x,y,\alpha]$ provides the integral for all possible centerlines or rays for a given point coordinate x,y. Here, only the square difference between



(a) Localized Radon Response (b) Derived Repsonse Map

Figure 3: Responses for a Checkerboard Corner

the strongest ray and the weakest ray is returned by $f_c[x,y]$ with m specifying how large the corner is expected to be in the image.

$$\begin{aligned} Rf_{local}[x,y,\alpha] &= \sum_{k=-m}^{k=m} f\left[x+k\cos(\alpha), y+k\sin(\alpha)\right] \\ f_c[x,y] &= \left[\max_{\alpha\in[0,\pi]}\left(Rf_{local}[x,y,\alpha]\right) - \min_{\alpha\in[0,\pi]}\left(Rf_{local}[x,y,\alpha]\right)\right]^2 \end{aligned} \tag{2}$$

However, calculating the localized Radon transform for each image point would be very time consuming. Therefore max and min are approximated by considering only $\alpha \in \{0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}\}$ as angles.

This is possible due to the point symmetry of checkerboard corners generating a function similar to a sine or cosine after the localized Radon transform is applied (Fig. 3(a)).

3.1 Implementation

Transforming an input image into a response map with this approximation can efficiently be achieved by convoluting the image with box filters using a kernel size matching the considered window patch around a corner. But instead of rotating the filter kernels by an angle α (steerable filter) the image is rotated counter wise, convoluted with two 1D kernels and rotated back. This is motivated by the work of Maire [13] using the same approach to speed up contour detection. In this case box filter kernels are used to calculate the integrals of a localized Radon transform for pre-defined discrete ray angles.

$$\begin{aligned} f_{blur}[x,y] &= \frac{1}{2k+1}\sum_{i=-m}^{i=m} f[x+i,y] \\ f_{rot}[x,y,\alpha] &= f[x\cos(\alpha)-y\sin(\alpha), x\sin(\alpha)+y\cos(\alpha)] \\ f_r[x,y,\alpha] &= f_{rot}\left(f_{blur}(f_{rot}(f[x,y],-\alpha)),\alpha\right) \propto Rf_{local}[x,y,\alpha] \\ f_c[x,y] &\sim \left[\max\left(f_r[x,y,0], f_r[x,y,\tfrac{\pi}{4}], f_r[x,y,\tfrac{2\pi}{4}], f_r[x,y,\tfrac{3\pi}{4}]\right) - \right. \\ &\quad \left. \min\left(f_r[x,y,0], f_r[x,y,\tfrac{\pi}{4}], f_r[x,y,\tfrac{2\pi}{4}], f_r[x,y,\tfrac{3\pi}{4}]\right)\right]^2 \end{aligned} \tag{3}$$

Each rotation combined with a convolution generates one new copy of the input image blurred in a discrete direction. Therefore, a blur kernel size of $n \times m$ is used corresponding to the windows size of the detector and is usually in the range between 1×3 and 1×9. It could also be generalized to more than 1 pixel vertically to detected corners of a specified

scale. In the next step, these blurred images are joined to a response map according to Eq. 3. A complete procedure of the detection is given below:

- Convert input image into a greyscale image.
- Supersample greyscale image by a factor of two to improve anti-aliasing.
- Rotate greyscale image around its center using the angles 0 and $\frac{\pi}{4}$.
- Blur rotated images each with an $n \times m$ and $m \times n$ box filter resulting in twice as many images as before and inflating the angle interval to $[0, \pi[$.
- Rotate blurred images back.
- Join images which are rotated back to their original orientation using Eq. 3.
- Blur resulting response map with a $k \times k$ box filter to account for the discretization errors.
- Locate corners by searching for local maxima in the blurred response map.
- Use thresholding and nonmaximal suppression to filter out weak corners.

4 Evaluation

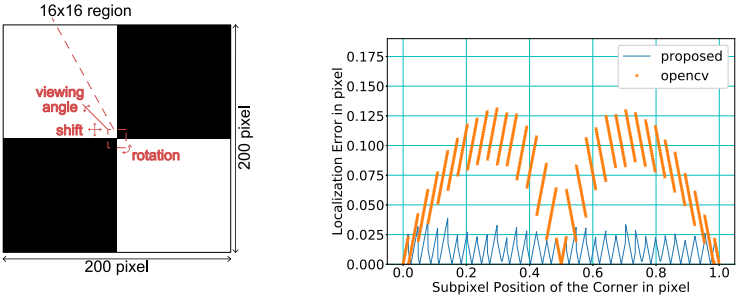
For evaluating the accuracy of the proposed checkerboard corner detector, it is tested on synthetic corners subject to rotation, perspective distortion, subpixel shifts, Gaussian blur, and Gaussian noise. As baseline, the sub-pixel corner localization method (1) proposed by Förstner [6] and implemented in OpenCV [1] is used which is also utilized by many checkerboard detectors as final corner refinement step such as [14] and [7]. In addition, a checkerboard detector is implemented on top of the proposed corner detector and tested on real images and compared against the checkerboard detector implemented in OpenCV [1] also using Förstner's sub-pixel corner localization.

4.1 Synthetic Datasets

The exact detector response to different corners is tested on a synthetic dataset. This allows one to precisely vary different parameters having influence on the imaged corner appearance. Here, each generated checkerboard corner is derived from the base checkerboard corner displayed in Fig 4(a) (200×200 pixel). For the evaluation, this base corner is perturbed by:

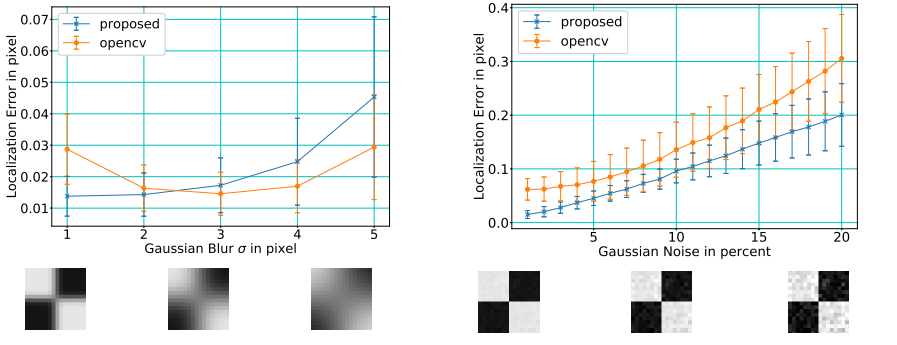
- Subpixel shift
- Angle of rotation around the image center
- Perspective viewing angle of the corner
- Standard deviation σ_b of a Gaussian blurring Filter
- Standard deviation σ_n of Gaussian image noise

All used synthetic datasets are derived from this base checkerboard by altering one or more of these parameters. However, after the base checkerboard corner is altered, only a region of $16x16$ pixel is used as input for testing the different detectors. Furthermore, the blur kernel size of the proposed method is fixed to $1x9$ pixel and the window size of the base line algorithm to $9x9$ pixel using at least 20 iterations. In addition, the proposed method super-samples the response map by a factor of two to decrease rounding errors.



(a) (b)

Figure 4: Localization errors for corners located at different subpixel locations. (a) base checkerboard corner with zero shift; (b) location error of the proposed method in comparison to OpenCV (dot plot) for corners shifted by a fraction of a pixel.



(a) (b)

Figure 5: Localization error of the proposed method for a checkerboard corner: (a) localization error over Gaussian blur; (b) localization error over Gaussian noise

For the first dataset the subpixel location of the checkerboard corner displayed in Fig. 4(a) is altered by small increments and used as input to calculate the localization error of the proposed method which is plotted in Fig. 4(b). Here, the baseline algorithm has difficulties to reliably estimate the subpixel location around one quarter away from a pixel boundary.

This different behaviour for various subpixel locations is reflected for the next datasets by generating 1000 runs using a uniform distribution for the subpixel location of the corner in addition to the altered parameter. For the second dataset the corner is blurred using different standard deviations for a Gaussian Filter. The location error for the proposed and the baseline method is plotted in Fig. 5(a) including the standard deviation of the localization error. Here, the baseline algorithm performs better for blurred images while the proposed method achieves its optimum already for crisp input data.

For the third dataset, the standard deviation of Gaussian noise and for the fourth dataset the rotation angle is altered and the localization errors are plotted in Fig. 5(b) and Fig. 6(a). The last dataset is generated by changing the viewing angle of the checkerboard corner in respect to the camera Fig 6(b).

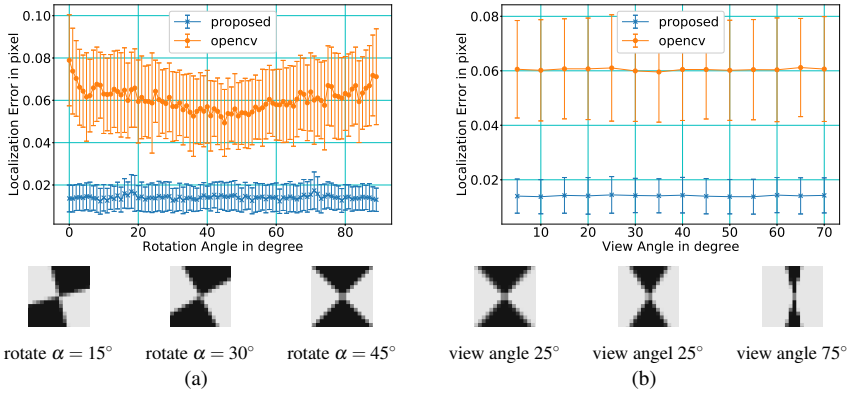


Figure 6: Localization error of the proposed method for a checkerboard corner: (a) localization error over rotation angle; (b) localization error over viewing angle

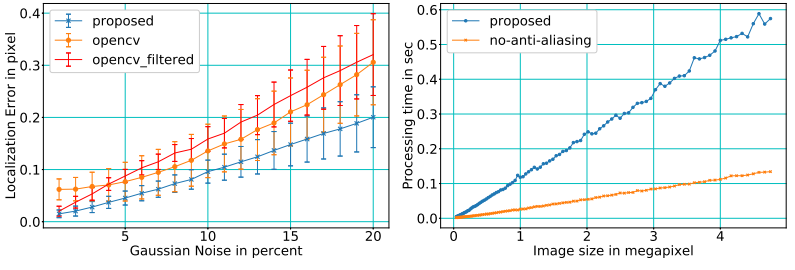


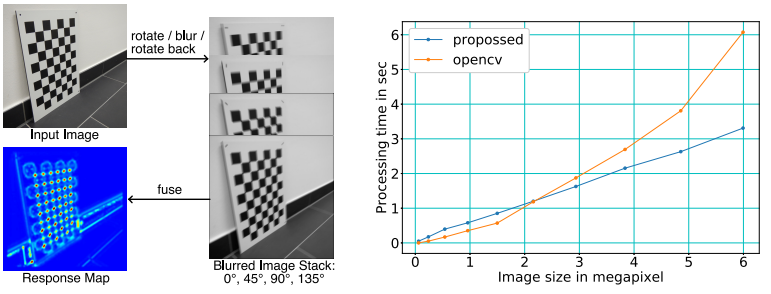
Figure 7: (a) Localization error over Gaussian noise of an modified baseline method using an additional filter stage of $\sigma = 3$; (b) processing time for corner detection and localization with and without anti-aliasing.

The baseline can be improved by a blur pre-filer to minimizing aliasing. However, this holds only for noise free images and the performance under Gaussian noise would degenerate as shown in Fig 7(a).

For online calibration it is essential to process incoming images fast enough to reduce latency between displayed images and recorded ones. Here, the processing time of the proposed method is mostly linear in respect to the input image size and plotted in Fig. 7(b). For measuring the processing time, the proposed method was executed on a 2.3 GHz Intel Core i7 for different image sizes utilizing one CPU core. The plotted data are the mean processing time for a certain image size running 10 times to reduce the effects of other processes running on the same hardware. The timing is given for the proposed method without an anti-aliasing step which has approximately two times worse subpixel accuracy but requires considerable less computing time.

4.2 Checkerboard Images

The proposed method has some additional advantages useful for a checkerboard detector, such as supporting scale space [3] [12] or providing the angle of each corner by fusing the



(a) (b)

Figure 8: Detection of Checkerboards using the proposed method: (a) response map calculation; (b) processing time for a checkerboard detection including subpixel estimation.

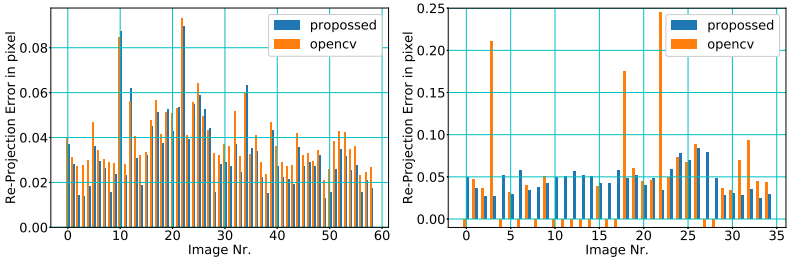
blurred input images using a subpixel peak detector estimating the angle of both centerlines. In addition, it also can be directly used to estimate the subpixel location of each corner. Based on this, a checkerboard detector is implemented which uses cross ratios to grow an initial $3x3$ checkerboard by adding more and more local maximums from the response map stored in a k-d tree. The initial board is generated by drawing a random point from an initial point set and by using the provided angles to search for neighbouring maximums in the response map. Here, the proposed method is also mostly linear in the number of processed pixels and usually outperforms the checkerboard detector implemented in OpenCV with respect to localization error and processing time. The processing time for the detection of the checkerboard visualized in Fig. 8(a) and scaled down to different image sizes is displayed in Fig. 8(b) for the proposed and the OpenCV method.

The checkerboard detector is also tested on two real image sets using two different camera setups. The first (Fig. 1(a)) was taken indoors in a controlled environment with a static checkerboard while the second (Fig. 1(b)) was taken outdoors under difficult illumination conditions and a quasi-static checkerboard setup introducing small amounts of motion blur.

For the checkerboard detection, the proposed method was compared to OpenCV's implementation ($9x9$ window size). The result was passed through the exact same code to estimate camera parameters and to calculate the remaining re-projection error of each checkerboard corner. These errors are plotted in Fig. 9 as mean error for each image of the set. In case, the checkerboard was not detected or the residual error was above 0.5 pixels the image was excluded and is marked with a negative re-projection error in the plots.

The camera model could absorb a localization bias in the principal point. All other parameters could only absorb an error with a rather complex systematic distribution in the image. The authors therefore believe that the residual constitutes a valid metric of evaluation.

Here, the proposed method is outperforming the OpenCV implementation under all tested conditions and results in considerable higher detection rates and a smaller re-projection error. The resulting intrinsic camera parameters are displayed in Tab. 1 and 2. In particular, the outdoor case shows a problematic increase of the re-projection errors for the baseline method while the proposed method can maintain a relative small re-projection error.



(a) (b)

Figure 9: Re-projection error of each detected checkerboard after calibration: (a) calibration in controlled environments; (b) outdoor calibration - negative values indicate that no checkerboard was detected.

	Proposed	OpenCV
Focal Length	990.52 / 990.94	990.31 / 990.66
Principle Point	642.94 / 552.31	642.37 / 551.81
Radial Distortion	-0.12172 / 0.12032	-0.11951 / 0.11917
Tangential Distortion	0.00316 / -0.00101	0.00316 / -0.00101
Mean Residual in pixel	0.0332	0.0391

Table 1: Camera Parameters - Indoors

	Proposed	OpenCV
Focal Length	746.49 / 745.86	746.79 / 745.48
Principle Point	514.66 / 512.73	514.12 / 514.12
Radial Distortion	0.01752 / -0.01585	0.02114 / -0.01993
Tangential Distortion	0.00062 / 0.002016	0.00147 / 0.000798
Mean Residual in pixel	0.0459	0.0738

Table 2: Camera Parameters - Outdoors

5 Conclusion

A new method for detecting and localizing checkerboard corners with subpixel accuracy was presented required for accurate camera calibration. It utilizes point symmetry of checkerboard corners in combination with a localized Radon transform approximated by box filters to achieve high performance even on large images. Here, tests have shown that the ability to localize checkerboard corners is close to the theoretical limit of $\frac{1}{100}$ of a pixel [11] while being considerably less sensitive to image noise than standard methods. Furthermore, due to the usage of simple image processing primitives such as blur and rotate, it can be easily implemented on GPUs and embedded systems to serve as fast, robust, and accurate detector for checkerboard calibration targets.

6 Acknowledgements

This work was supported by the ERA-NET Cofund MarTERA and the German Project Management Jülich (PtJ) as part of the joint research project RoboVaaS (grant 03SX463B).

References

[1] G Bradski. The OpenCV Library. *Dr Dobbs Journal of Software Tools*, 25:120–125, 2000. ISSN 1044-789X. doi: 10.1111/0023-8333.50.s1.10.

[2] Jun Chu, Anzheng GuoLu, and Lu Wang. Chessboard corner detection under image physical coordinate. *Optics & Laser Technology*, 48:599–605, jun 2013. ISSN 00303992. doi: 10.1016/j.optlastec.2012.11.010.

[3] J L Crowley and a C Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE transactions on pattern analysis and machine intelligence*, 6(2):156–70, feb 1984. ISSN 0162-8828.

[4] V Douskos, I Kalisperakis, G Karras, and E Petsa. Fully automatic camera calibration using regular planar patterns. *Int. Arch. Photogram. . . .*, 2008.

[5] RB Fisher and DK Naidu. A comparison of algorithms for subpixel peak detection. *Image Technology*, 1996.

[6] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305, 1987.

[7] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 3936–3943, 2012. ISBN 9781467314053. doi: 10.1109/ICRA.2012.6224570.

[8] M. Van Ginkel, C L Luengo Hendriks, L. J. Van Vliet, C. L. Luengo Hendriks, and L. J. Van Vliet. A short introduction to the Radon and Hough transforms and how they relate to each other. *Reading*, pages 1–11, 2004.

[9] Hyowon Ha, Michal Perdoch, Hatem Alismail, In So Kweon, and Yaser Sheikh. Deltille Grids for Geometric Camera Calibration. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 5354–5362, 2017. ISBN 9781538610329. doi: 10.1109/ICCV.2017.571.

[10] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Procedings of the Alvey Vision Conference 1988*, pages 23.1–23.6, 1988. doi: 10.5244/C.2.23.

[11] Ken Kiger. Introduction of Particle Image Velocimetry. *Burgers Program For Fluid Dynamics*, page 57, 2010.

[12] T Lindeberg. *Discrete Scale-space Theory and Scale-space Primal Sketch*. PhD thesis, Royal Institute of Technology,Sweden, 1991.

[13] Michael Randolph Maire. *Contour detection and image segmentation*. PhD thesis, University of California, Berkeley, 2009.

[14] Martin Rufli, Davide Scaramuzza, and Roland Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 3121–3126, 2008. ISBN 9781424420582. doi: 10.1109/IROS.2008.4650703.

[15] Eric D. Sinzinger. A model-based approach to junction detection using radial energy. *Pattern Recognition*, 41(2):494–505, feb 2008. ISSN 00313203. doi: 10.1016/j.patcog.2007.06.032.

[16] S Suzuki. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 46:32–46, 1985.

[17] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.