ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration

Simon Placht[1,2], Peter Fürsattel[1,2], Etienne Assoumou Mengue[2],
Hannes Hofmann[1], Christian Schaller[1], Michael Balda[1],
and Elli Angelopoulou[2]

[1] Metrilus GmbH, Erlangen, Germany
[2] Pattern Recognition Lab, University of Erlangen, Nuremberg, Germany

Abstract. We present a new checkerboard detection algorithm which is able to detect checkerboards at extreme poses, or checkerboards which are highly distorted due to lens distortion even on low-resolution images. On the detected pattern we apply a surface fitting based subpixel refinement specifically tailored for checkerboard X-junctions. Finally, we investigate how the accuracy of a checkerboard detector affects the overall calibration result in multi-camera setups. The proposed method is evaluated on real images captured with different camera models to show its wide applicability. Quantitative comparisons to OpenCV's checkerboard detector show that the proposed method detects up to 80% more checkerboards and detects corner points more accurately, even under strong perspective distortion as often present in wide baseline stereo setups.

Keywords: Checkerboard Detection, Saddle-Based Subpixel Refinement, Multi Camera Calibration, Low Resolution Sensors, Lens Distortion.

1 Introduction

The goal of camera calibration is the recovery of intrinsic parameters of cameras. In the case of a multi-camera setup it also includes the estimation of their spatial relation to each other. Most calibration methods involve the detection of an object with known geometric properties in the scene.

Multiple methods which put different constraints on the calibration pattern can be used to estimate the desired parameters. Some algorithms allow or require three dimensional objects [12], circular patterns [5] or planar checkerboard patterns. Three dimensional patterns are hard to build and typically more expensive than simple planar checkerboard patterns which can be generated by a regular printer. Circle patterns are not invariant with respect to projective and nonlinear transformations like lens distortion [5,7]. Complex patterns which include self-identifying patterns [4] are also not considered in this work as they typically have higher requirements on sensor resolution and noise levels.

Rather, we focus on planar patterns in combination with Zhang's calibration method [14]. Although this calibration approach is not limited to a particular

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 766–779, 2014.

planar calibration pattern we consider only checkerboard patterns due to their robustness with respect to different distortions, low price and simple construction. The regular pattern and high-contrast edges of checkerboards makes them particularly suitable for automatic detection with high accuracy. However, an extreme pose of the checkerboard, low sensor resolution, image noise or lens distortion may still lead to inaccurate corner point coordinates or may even make automatic detection impossible. Wide baseline stereo camera setups are particularly prone to extreme poses and a robust pattern detector becomes a necessity.

In this work we present a novel checkerboard detector which finds checkerboards at extreme poses and on distorted images with high accuracy. Our detection method involves two steps. First, we detect the checkerboard in the image and calculate initial pixel coordinates for the inner checkerboard corners. In the second step, the initial corners are refined to subpixel accuracy with a method based on the work of Lucchese and Mitra [6]. Their approach consists of fitting a polynomial to saddlepoints in the vicinity of the initial coordinates to calculate the corner coordinates with subpixel accuracy. Typically some sort of filtering, for example Gaussian smoothing [1], is applied before fitting the polynomial. Instead of using a Gaussian we apply a cone-shaped filter kernel escpecially tailored for surface fitting around checkerboard X-junctions.

There is considerable prior work on checkerboard pattern detection. A widely used approach is OpenCV's checkerboard detection algorithm which is based on the work of Vezhnevets[1]. After thresholding the images, the algorithm tries to separate black checkerboard quads from each other by applying erosion. Next quadrangles are fitted into the black quads. By merging the corners of the quadrangles the inner checkerboard corners can be calculated. Rufli et al. [9] present an extension to this algorithm which includes different erosion kernels and a heuristic for quadrangle linking in order to make the algorithm more robust to lens distortion. However, their algorithm does not include any subpixel refinement of corner points. Wang et al. [13] present a method which fits lines into initial corner points and calculates the final corner coordinates by intersecting these lines. While this method works well with images with only small distortions, the line fitting is expected to be less accurate if the image is significantly distorted. All the aforementioned methods have in common that they require prior knowledge about the number of squares of a calibration pattern. De la Escalera et al. [3] avoid this limitiation by automatically finding checkerboard-like patterns by combining corner and line detection. In their method the Hough transform is used to find straight lines, which however may fail for images under medium or strong lens distortions. Dao et al. present [2] a method which is able to detect checkerboards even if they are not planar or partly occluded. Such a technique is particularly useful if the checkerboard pattern is not used for camera calibration but for example for geometry reconstruction but is not useful for camera calibration.

[1] http://graphicon.ru/oldgr/en/research/calibration/opencv.html

We evaluate our proposed method with real images captured with different multi-camera setups consisting of both industrial and consumer cameras. This allows us to examine the influence of lens distortion, sensor resolution and checkerboard pose on detection accuracy. Furthermore, we compare our method to OpenCV's checkerboard detector[2]. We show that OpenCV's initial detection of checkerboards is less robust with respect to the aforementioned criteria and also that our subpixel refinement method achieves higher accuracy.

This paper is organized as follows. Section 2 describes our checkerboard detection algorithm as well as the subpixel refinement method. Our experiments and results are discussed in Section 3. In the last section we draw conclusions on our evaluation and present a short outlook on future work.

2 Robust Checkerboard Advanced Detection (ROCHADE)

The localization of a checkerboard pattern in an image can be divided into two subtasks: the detection of the checkerboard pattern, including corner initialization and the refinement of the inner checkerboard corners. An inner corner is the vertex shared by four adjacent checkerboard squares (two black, two white ones) which together form a larger square.

2.1 Detection of Checkerboard Pattern

A robust checkerboard pattern detector should satisfy the following properties:

- Detection even under the presence of strong lens distortions. Due to deviations from straight lines, the shape of the checkerboard squares in the acquired image will differ from the contours of a quadrangle.
- Invariance to camera-dependent parameters (range of values, resolution).
- Robustness with respect to pose of the acquired checkerboards.
- Detection, even when the checkerboard field lengths are not constant (important for projector to camera calibration).

In order to keep the calibration procedure user-friendly, it should also work in real-time, be fully automatic and without configuration parameters that need to be adjusted to camera types and setups.

The key idea behind our checkerboard detection algorithm is an edge graph generation for the whole image. Usually the graph is disconnected, since the checkerboard is not the only acquired object. Inner checkerboard corners correspond to saddle points in the edge graph, i. e. nodes with three or more neighbors. Hence, the search space is reduced to those connected components where the number of saddle points matches the number of inner corners of the checkerboard. An outline of the processing pipeline used for checkerboard detection is given in Figure 1. A detailed description of the 7-step processing pipeline is provided below:

[2] In our evaluation we use OpenCV 2.4.8

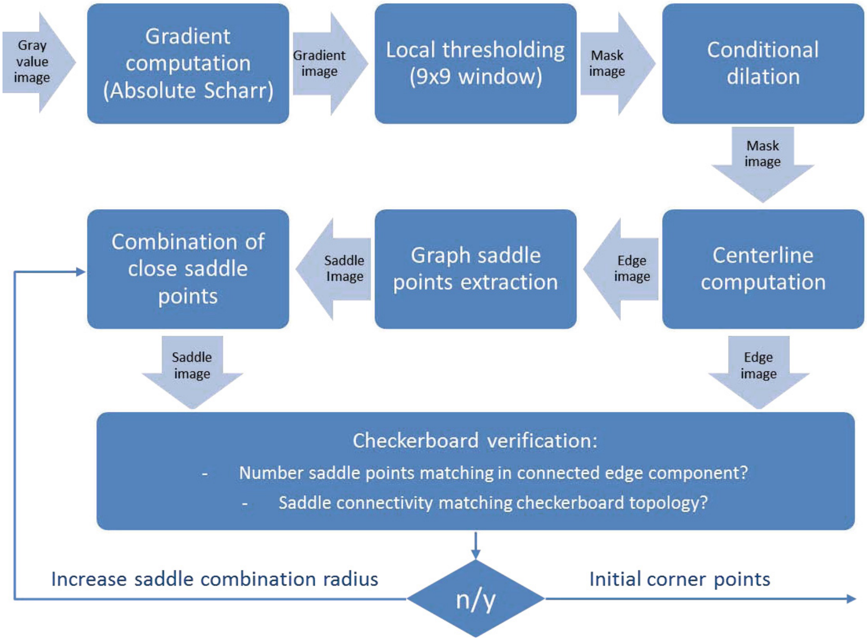

Gray value image
Gradient computation (Absolute Scharr)
Gradient image
Local thresholding (9x9 window)
Mask image
Conditional dilation
Mask image
Centerline computation
Edge image
Graph saddle points extraction
Saddle Image
Combination of close saddle points
Saddle image
Edge image
Checkerboard verification:
- Number saddle points matching in connected edge component?
- Saddle connectivity matching checkerboard topology?
Increase saddle combination radius
n/y
Initial corner points

Fig. 1. The processing pipeline of the checkerboard detection algorithm

0. Downsampling
   This is an optional stage for controlling the processing time of the checkerboard detection. If the input image is downsampled, the minimum field size of detectable checkerboards will increase. For the corner refinement stage described in Section 2.2 the original image size is used.
1. Gradient computation
   In order to highlight the edges/corners of the checkerboard pattern, a Scharr gradient computation with a kernel size of 3x3 is applied. We have chosen a Scharr kernel for filtering because of its better rotational symmetry compared to the more commonly used Sobel filter. Since the subsequent steps are not based on the gradient directions, we consider only the magnitude of the horizontal and vertical gradient.
2. Local thresholding
   Having computed the gradient, we generate a mask image (see Figure 2c) in which the high gradient values close to the checkerboard edges/corners are separated from the homogenous regions inside the checkerboard squares. This separation is carried out by thresholding in a $(2\tau+1)\times(2\tau+1)$ window around the current pixel. In all our experiments, we have set the thresholding window's radius to $\tau = 4$. If the gray value of the center-pixel is in the upper 60% of the windows intensity range, it is set to true in the resulting binary mask image, otherwise it is set to false. The threshold of 60% has

been determined heuristically using our test data. The effect of the local thresholding on the gradient image is threefold:

- Pixels very close to the checkerboard edges are set to true.
- Pixels, that are not very close to the edges (low gradient values) but within the thresholding window radius are set to false.
- Pixels of homogeneous regions which are not in the neighborhood of edges are set to true or false depending on image noise (see e.g. region above the topmost checkerboard square row in Figure 2c).

As can be seen in Figure 2c, homogenous checkerboard regions are clearly separated from the edge regions.

3. Conditional dilation
On the resulting mask image a conditional morphologic dilation is applied in order to close small holes and notches in the mask. These holes / notches can appear either due to sensor noise or image correction artifacts (especially for low-cost webcams). The applied conditional dilation only adds pixels with at least six "true" neighbors (out of eight) to the mask.
4. Centerline computation
Our implementation of the centerline extraction is based on the Distance Transform with thinning as post-processing step as proposed by Niblack et al. [8]. The result is a binary mask image g with a centerline thickness of 1 (see Figure 2e) which corresponds to the undirected edge image graph G. The edge image graph G with nodes V and edges E is defined as follows:

$$G = (V, E) \tag{1}$$

$$V = \{v \mid g(v) = \text{"true"}\} \tag{2}$$

$$E = \{e = \{v_i, v_j\} \mid \operatorname{dist}(v_i, v_j) \leq \sqrt{2}\} \text{ ("8-connected neighborhood")} \tag{3}$$

The set of saddle points in the edge image is defined as:

$$S = \{s \in V \mid |E_s| \geq 3\} \text{ with } E_s = \{e = \{v_i, v_j\} \in E \mid v_i = s \vee v_j = s\} \tag{4}$$

Since the checkerboard edges form a connected component in the input mask they are also connected in the resulting centerline mask. When the centerline mask is interpreted as a graph, every checkerboard edge pixel is part of a cycle.

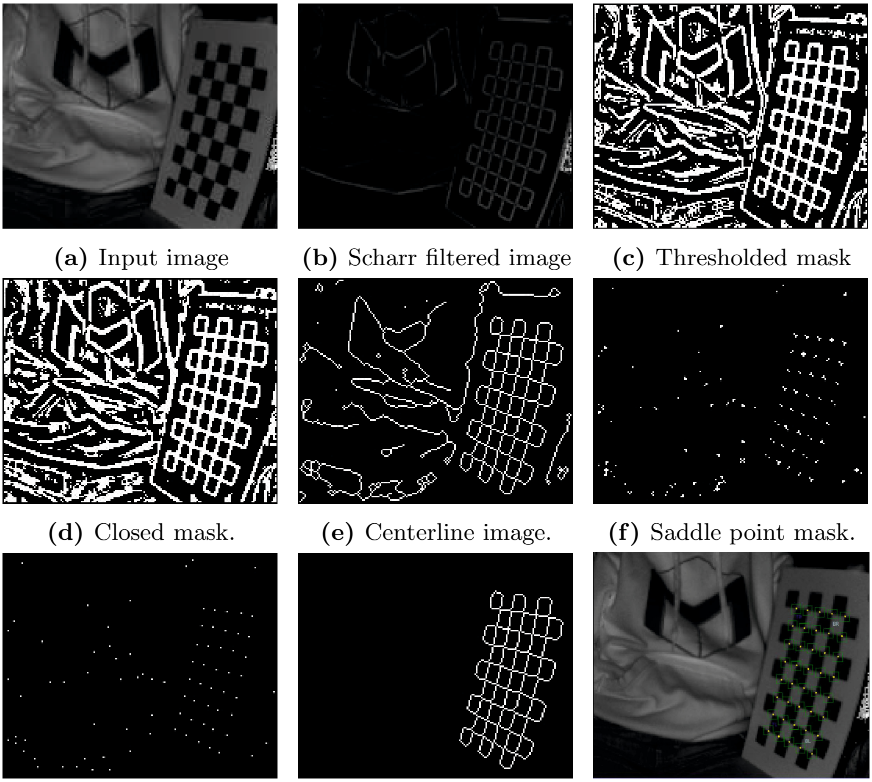
Due to noise, the connected centerline component related to the checkerboard can contain small branches, which are not part of a cycle and which would generate additional saddle points at the connection to the grid-shaped checkerboard edge structure. In order to remove these "dead ends", we search for pixels with only one neighbor and remove all "true" pixels along the path until we reach the next saddle point. The non-cycle removal is skipped for pixels at the image border in order to enable checkerboard detection if its margin squares are projected partially beyond the image borders.

After removal of acyclic paths there may remain mini-cycles at the branch region and the edge image does not have a centerline thickness of 1 anymore. Therefore, we remove these pixels by thinning to restore the single pixel thickness criterion (which has already been achieved after initial centerline computation). This thinned centerline image can again be interpreted as a graph $G^{'} = (V^{'}, E^{'})$ with saddle points $S^{'}$.

5. Graph saddle points extraction
   The extraction of saddle points in the thinned centerline image becomes equivalent to selecting all pixels with three or more "true" neighbors in the edge image graph $G^{'}$ (corresponds to set $S^{'}$). In the ideal case, the number of corners in the connected edge image graph now matches the number of inner corners in the checkerboard. However, the edge graph can contain multiple, nearby saddle points around a single checkerboard corner (see Figure 2f). This problem is adressed by combining closely clustered saddle points as described in step 6. The result of this step is a mask image with "true" values at the saddle locations and "false" values otherwise.
6. Combination of saddle points
   Since the inner checkerboard corners may generate multiple, nearby saddle points, we cluster saddle points which are within a saddle combination distance α from each other. We keep only one saddle point for each cluster. The combination of saddle points is based on the mask image representing all saddle points (Figure 2f). It is not using the graph respresentation. The challenge of the saddle point combination is the choice of the parameter α. If it is too high, saddle points corresponding to different checkerboard corners are combined for small checkerboard field sizes. Low values on the other hand may result in having multiple saddle points for one checkerboard corner. Therefore, we start with a low value of 2, proceed with the checkerboard verification step (7) and iteratively increase the parameter α as long as we either detect a checkerboard or the empirically chosen threshold of 5 is exceeded.
7. Checkerboard verification
   For the automatic detection and verification of the checkerboard pattern in the image we use both the centerline image (Figure 2e) and the mask image representing the combined saddle points (Figure 2g). We search for connected components in the centerline image and count the number of saddle points for every component. If the number of saddle points is equal to the number of checkerboard corners, we check the adjacency of the saddle points in the image graph $G^{'}$. Directly adjacent saddle points in $G^{'}$ (corresponding to adjacent checkerboard X-junctions) are the representatives of two different saddle clusters, but with no saddle point belonging to another cluster in the shortest path between the two points. If the adjacency structure of all saddle points equals a grid structure we return the mean values of the saddle point clusters as corner intializations for the subsequent refinement step.

2.2 Refinement of Checkerboard Corners

After initializing the checkerboard corners, the corners need to be refined with sub-pixel accuracy. For checkerboard patterns typically two methods are proposed: edge approximation techniques (e.g. used by OpenCV) and surface fitting



(a) Input image **(b)** Scharr filtered image **(c)** Thresholded mask

(d) Closed mask. **(e)** Centerline image. **(f)** Saddle point mask.

(g) Combined saddle point mask. **(h)** Detected checkerboard component. **(i)** Detection result in input image.

Fig. 2. Intermediate results of the checkerboard detection algorithm

around corner candidates [7]. Since edge approximation methods are potentially biased because of lens distortions, we apply a modified version of the corner point estimation strategy introduced by Lucchese and Mitra [6]. Our proposed method consists of two steps:

1. Image filtering around the initial corner point location to obtain an intensity surface amenable to fitting a bivariate quadratic polynomial.
2. Fitting a bivariate polynomial of degree 2 represented by the parameters $a_{1,2,\ldots,6}$ in the local neighborhood of the initial corner (x, y).

Similar to [6], we apply a lowpass filter as preprocessing step. Instead of using a normalized Gaussian filter as a preprocessing step we apply a 2-D cone filter to the original input image. Since a cone is sectionally linear, the convolution with a combination of step functions (checkerboard pattern) yield sectionally defined bivariate quadratic polynomials. The kernel c of the cone filter applied in our tests is defined as following:

$$c_{i,j} = \max\left(0, \gamma + 1 - \sqrt{(\gamma - i)^2 + (\gamma - j)^2}\right), \qquad (5)$$

where γ is the half size of the kernel and $i, j \in \{0...2\gamma\}$ are the kernel indices. The size of the cone filter should be at least chosen to be at least as large as the half window size used for surface fitting, otherwise the filtered surface will have constant regions which cannot be well approximated by a bivariate quadratic polynomial.

The polynomial fitting in the local neighborhood of the checkerboard corner is expressed by:

$$\underset{a_{1,2,\ldots,6}}{\arg\min} \sum_{i=-\kappa}^{\kappa} \sum_{j=-\kappa}^{\kappa} \left(a_1 x_i^2 + a_2 x_i y_j + a_3 y_j^2 + a_4 x_i + a_5 y_j + a_6 - f(x_i, y_j)\right)^2, \quad (6)$$

where $x_i = x + i$, $y_j = y + j$, f is the filtered input intensity image and κ is the half window size used for surface fitting. The saddle point of this polynomial is equal to the refined corner point.

3 Experiments

In this paper we propose a checkerboard detection technique which finds corner points of a checkerboard calibration pattern with subpixel accuracy. We evaluate this method with different sensors and different stereo camera setups with respect to detection accuracy and overall detection rate and compare it to OpenCV's checkerboard detector. All results which refer to OpenCV's checkerboard detector are computed with the detector's internal adaptive thresholding and image normalization enabled.

The evaluation section can be split into two parts. The first part covers stereo camera setups with a small baseline and cameras being verged only slightly towards each other. In these experiments we investigate the subpixel accuracy of the corner refinement stage and the influence of the number of calibration images on the accuracy of the calibration parameters. For these experiments we use only images in which checkerboards are detected both by ROCHADE and OpenCV. The subpixel accuracy is given by the error that occurs when measuring the size of known calibration patterns with a stereo camera setup. We perform these measurements with different sets of calibration parameters which are estimated with different numbers of calibration images.

In the second experiment we evaluate the robustness of the initial checkerboard detection in different setups. We compare the detection rate of the presented method with OpenCV's detection rate and additionally to the detection rate of the OCamCalib Toolbox [10] which implements an improved version of OpenCV's detector as decribed in [9].

3.1 Subpixel Accuracy

For our first experiment we use three different stereo camera setups with different sensor resolutions and lens distortions. Our first setup consists of two Mesa

SR4000 ToF cameras[3]. The Mesa SR4000 has a sensor resolution of 176 x 144 pixels, considerable lens distortion and a pixel resolution of 16 bits. In all our experiments only the 2D intensity images of the ToF sensor are used. We choose these cameras because of the low sensor resolution and the large amount of noise in the images.

The half size of the search window κ in the subpixel refinement step is set to 2 px in case of the Mesa cameras. The filtering parameter γ is set to κ (see Section 2). For better comparability the 16 bit valued images are converted to 8 bit valued images. This is required as OpenCV's checkerboard detection works only on 8 bit images. In the following experiments a simple min/max-windowing is performed, whenever necessary, on the Mesa camera images.

The second camera type we use is the Ensenso N10 stereo camera[4]. The sensor resolution of this camera is 752 x 480 pixels and its lenses introduce only a small amount of distortion. Images captured with this camera have a pixel resolution of 8 bits per pixel. For the Ensenso camera κ is set to 5 px. Again γ is set to the same value as κ.

Our third stereo setup consists of two IDS UI-1241LE[5] cameras. The distortion which is introduced by the lenses is insignficant. The images we capture have a size of 1280 x 1024 pixels with intensities represented with 8 bits per pixel. For this setup κ and γ are set to 5 px.

In all experiments we use a checkerboard with 7 x 10 squares as calibration pattern of which only the corners of the inner 5 x 8 squares are detected. Each square has an edge length of 50 mm.

The evaluation is performed with two image series per camera setup which we call the calibration images and the evaluation images. Both image series consist of image pairs which contain the checkerboard. For estimating the intrinsic and extrinsic parameters of these setups we use OpenCV's implementation of Zhang's method [14] with point correspondences derived from a subset of the calibration images. These point correspondences are either derived with OpenCV's checkerboard detector or ROCHADE.

We work with subsets of the calibration image set in order to investigate the influence of the number of used images on the overall calibration result. For each number of calibration images we estimate two sets of calibration parameters, the first estimated with point correspondences detected with ROCHADE, the second one with OpenCV's checkerboard detector. During calibration both radial and tangential lens distortion is considered.

After estimating the intrinsic and extrinsic parameters, the corners of the evaluation image series are detected with a detector $D = \{\text{OpenCV}, \text{ROCHADE}\}$. With the calibration parameters and the corner point correspondences we can then measure the length of the of the n-th checkerboard square edge $m_{j,n}$ on the j-th image by simple triangulation.

[3] Mesa Imaging AG, Zürich, `http://www.mesa-imaging.ch`

[4] Ensenso GmbH, Freiburg, `http://www.ensenso.de`

[5] IDS Imaging Development Systems GmbH, Obersulm, `http://www.ids-imaging.com`

The mean absolute error is calculated based on the true edge length l_t for all edge measurements of all images of the evaluation image series as given by Equation 7.

$$\mu_D = \frac{1}{N_i N_e} \sum_{j}^{N_i} \sum_{n}^{N_e} |m_{j,n} - l_t| \ , \tag{7}$$

where N_i denotes the number of evaluation images and N_e the number of edges per checkerboard.

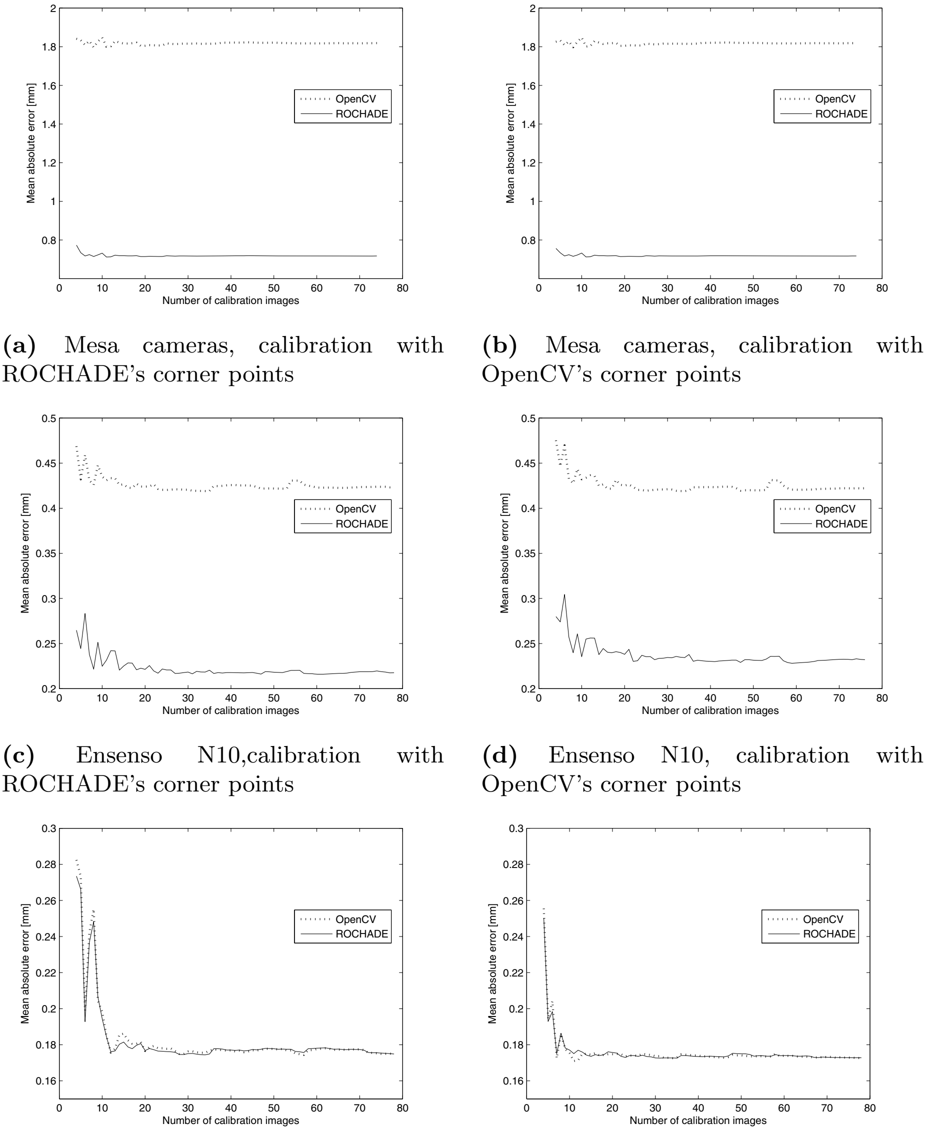
The results for the Mesa SR4000 stereo camera setup are shown in Figures 3a and 3b. One can see that the accuracy of the length measurements does not improve if more than 20 images are used during calibration. A larger number of images is not necessary because the low resolution of the images limits the accuracy for finding corners anyway. If a sufficient number of images is used during calibration the mean absolute measurement accuracy is approximately 0.71 mm with ROCHADE and 1.85 mm with OpenCV. It is worth mentioning that the accuracy of ROCHADE does not decrease noticably due to the 8 bit conversion performed before detecting the corners. One can also see that the measurement accuracy is independent of the pattern detector which is used during calibration of the stereo setup.

Figures 3c and 3d show the mean absolute measurement error for the checkerboards captured with the Ensenso N10 camera. Due to the higher resolution the corner points can be detected more accurately which leads to a significantly lower mean absolute error compared to the Mesa SR4000 camera setup. In this example the mean absolute error is approximately 0.21 mm for ROCHADE and 0.53 mm for OpenCV's detector. Again one can see that the measurement accuracy does not improve if more than approximately 30 images are used for calibration. The larger number of images required is caused by a higher resolution and less noisy images of the Ensenso camera.

The results for our third setup are given in Figures 3e and 3f. With this setup the mean absolute measurement error is almost identical for ROCHADE and OpenCV. Due to the higher resolution OpenCV's detector is able to refine checkerboard corner coordinates more accurately as gradients can be computed more reliably than in low resolution images. Furthermore, OpenCV benefits more from the camera's little lens distortion than ROCHADE as the squares and therefore gradients around intial corners are not distorted. Similar as in the previous setups the detection accuracy does not improve if more than approximately 30 images are used.

3.2 Overall Detection Rate

The overall detection rate is defined as the number of checkerboards which are detected compared to the total number of images of a series. When calibrating a new camera a certain number of point correspondences, which can be derived from detected checkerboards, is required. In this experiment, a high detection rate reflects high robustness with respect to lens distortion and extreme checkerboard poses.



(a) Mesa cameras, calibration with ROCHADE's corner points

(b) Mesa cameras, calibration with OpenCV's corner points

(c) Ensenso N10,calibration with ROCHADE's corner points

(d) Ensenso N10, calibration with OpenCV's corner points

(e) IDS uEye, calibration with ROCHADE's corner points

(f) IDS uEye, calibration with OpenCV's corner points

Fig. 3. Mean absolute measurement errors for different camera setups with respect to the number of used calibration images

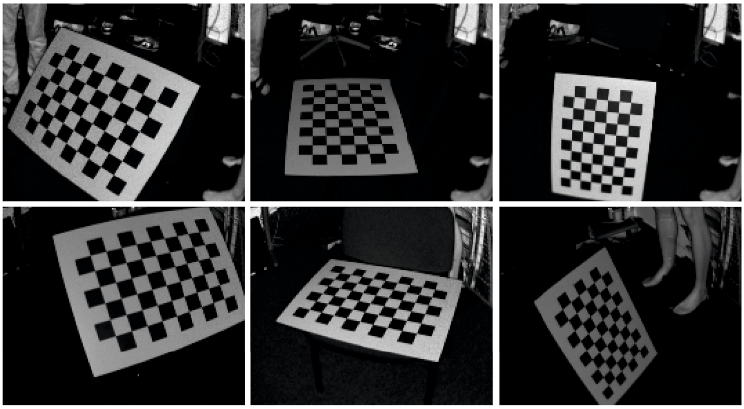


Fig. 4. Three sample image pairs of the Mesa SR4000 wide baseline image set. The top row shows the images of the first camera, the second row the corresponding images of the second camera. Images are enlarged with nearest neighbour interpolation.

Two image series are captured with a wide baseline stereo camera setup. In such a setup the detection of calibration pattern is difficult due to the extreme pose of the calibration pattern. The first setup consists of the two Mesa SR4000 cameras as described in Section 3.1 but with a baseline of approximately 103 cm. Figure 4 shows three image pairs of this series.

In our second setup we use the IDS uEye cameras we have already used in the first experiment but again with a wider baseline of approximately 224 cm.

The third image series was taken with a single GoPro Hero 3 camera. We use this camera to investigate the detection rate on high resolution images which suffer from strong lens distortion.

The results of this experiment are shown in Table 1.

Table 1. Successful checkerboard detections for ROCHADE, OpenCV and OCamCalib [10]

Camera Setup	Total images	ROCHADE	OpenCV	OCamCalib
Stereo Mesa SR4000 setup	103	91	8	50
Stereo IDS uEye setup	100	100	100	100
Single GoPro Hero 3	100	96	73	100

In the case of the Mesa SR4000 stereo camera setup 91 out of 103 checkerboard patterns are detected by ROCHADE whereas OCamCalib finds only 50 and OpenCV only 8 patterns. Note that the checkerboards are not only found but also detected highly accurately. The detected cornerpoints are suitable for estimating the intrinsic and extrinsic parameters of this stereo camera setup. When using 91 detected checkerboard patterns and performing the same evaluation as presented in Section 3.1 we achieve a mean abolute measurement error of 0.273 mm when measuring with ROCHADE. This error is smaller than in the previous experiment due to the wider baseline of the camera setup. However, due to extreme foreshortening ROCHADE fails to detect the checkerboard in a couple of images.

The GoPro Hero 3 comes with a wide angle lens which significantly distorts images and therefore makes the automatic detection of checkerboards more difficult. For evaluation purposes we use 100 images which are captured with a resolution of 3840 x 2160 pixels. Due to the higher resolution, OpenCV's checkerboard detector is able to detect a larger percentage of checkerboards than in the low resolution setup with the Mesa cameras. Similar to OpenCV's detector OCamCalib benefits greatly from the high resolution of the evaluation images. Due to the adaptions for the detection of checkerboards captured with omnidirectional cameras a high detection rate becomes possible.

All detectors are able to find all checkerboards in the IDS uEye wide baseline stereo setup, even though the image resolution is signifcantly lower than the resolution of the previously presented GoPro camera. However, with the IDS uEye camera almost no lens distoriton is present which simplifies the detection progress.

4 Conclusion

Recent developments in real time depth imaging (Microsoft Kinect, Creative Senz3D) make the fast and accurate (multi-) camera calibration of mid and low resolution sensors more and more important. Moreover, low cost lenses used in mobile devices often show a relatively high optical distortion. Our presented checkerboard detection method outperforms OpenCV's checkerboard detector in low resolution images or highly distorted images. Especially in the low resolution case ROCHADE is more robust with respect to extreme poses of the checkerboard. Furthermore our method performs at least as good as OpenCV's detector in medium and high resolution images if no significant amounts of distortion are present. Especially in multi camera setups with wide baselines the presented algorithm still allows a highly accurate estimation of intrinsic and extrinsic camera parameters whereas a calibration with OpenCV's checkerboard detection is impossible.

We also evaluate the influence of accurate corner detection on the overall calibration of a stereo setup. In our experiments we show that the accuracy of intrinsic and extrinsic parameters does not improve with a higher accuracy of the point correspondences, provided a certain subpixel accuracy has already been achieved. Future work will include the determination of this minimum accuracy.

By presenting quantitative results we also showed that the accuracy of multi-camera calibration typically cannot be increased if more than a certain number point correspondences, or calibration images, are used. These results comply with the results published by Sun et al. in [11]. However, it might be possible to reduce this number by chosing the right checkerboard poses. Furthermore an extension of the presented algorithm is planned which does not require that the whole checkerboard is visible in the image. A Matlab implementation of the method presented in this paper can be found on our website[6].

Acknowledgements. This publication was supported by the German Federal Ministry of Education and Research as part of the Spitzencluster Medical Valley program 13GW0029A.

References

1. Chen, D., Zhang, G.: A new sub-pixel detector for x-corners in camera calibration targets. WSCG (Short Papers) 5, 97–100 (2005)
2. Dao, V.N., Sugimoto, M.: A robust recognition technique for dense checkerboard patterns. In: Pattern Recognition (ICPR), 2010 20th International Conference on. pp. 3081–3084. IEEE (2010)
3. De la Escalera, A., Armingol, J.M.: Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. Sensors 10(3), 2027–2044 (2010)
4. Fiala, M., Shu, C.: Self-identifying patterns for plane-based camera calibration. Machine Vision and Applications 19(4), 209–216 (2008)
5. Heikkila, J.: Geometric camera calibration using circular control points. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1066–1077 (2000)
6. Lucchese, L., Mitra, S.K.: Using saddle points for subpixel feature detection in camera calibration targets. In: Asia-Pacific Conference on Circuits and Systems. vol. 2, pp. 191–195. IEEE (2002)
7. Mallon, J., Whelan, P.F.: Which pattern? biasing aspects of planar calibration patterns and detection methods. Pattern Recognition Letters 28(8), 921–930 (2007)
8. Niblack, C.W., Gibbons, P.B., Capson, D.W.: Generating skeletons and centerlines from the distance transform. CVGIP: Graph. Models Image Process. 54(5), 420–437 (Sep 1992)
9. Rufli, M., Scaramuzza, D., Siegwart, R.: Automatic detection of checkerboards on blurred and distorted images. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. pp. 3121–3126. IEEE (2008)
10. Scaramuzza, D.: Omnidirectional Vision: from Calibration to Root Motion Estimation. Ph.D. thesis, Swiss Federal Institute of Technology Zurich (ETHZ) (February 2008)
11. Sun, W., Cooperstock, J.R.: An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques. Machine Vision and Applications 17(1), 51–67 (2006)
12. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation 3(4), 323–344 (1987)
13. Wang, Z., Wu, W., Xu, X., Xue, D.: Recognition and location of the internal corners of planar checkerboard calibration pattern image. Applied mathematics and computation 185(2), 894–906 (2007)
14. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330–1334 (2000)

[6] `http://www.metrilus.de/rochade/`