https://doi.org/10.2352/EI.2022.34.7.ISS-231

Using Images of Partially Visible Chart for Multi-Camera System Calibration

Radka Tezaur, Gazi Ali, and Oscar Nestares; Intel Corp.; Santa Clara, California

Abstract

Making it possible to use images in which the calibration chart is only partially visible makes the geometric calibration of multi-camera system much more efficient as well as more accurate and robust. This is particularly true in the case of systems involving a larger number of cameras and wide field of view cameras with significant radial distortion. The calibration tool developed by us that uses a checkerboard chart with AprilTags is described, and the benefits of using such a tool compared to the traditional checkerboard chart calibration are demonstrated. The provided examples using both synthetic and real data sets illustrate the impact of the requirement that all chart points must be detected on the spatial distribution of the corner points used for calibration and the accuracy of the geometric calibration.

Introduction

Highly accurate geometric calibration of a camera system is crucial for a wide range of computer vision and computational imaging applications. The most common method uses a checkerboard calibration chart. While various other methods exist, e.g. [1], [2] and other methods using structured light, or methods employing different types of calibration targets, like [3], the calibration with a checkerboard chart is very popular because it does not require any expensive equipment, only a simple checkerboard chart that can be easily manufactured, and the calibration software is widely available. It is included in common software packages like OpenCV [4] or Matlab Computer Vision Toolbox [5] and it easy to use. It does not require any detailed camera specifications that may be hard to obtain or any educated guesses for initial parameter values, and it can be used with many different camera types, including fisheye cameras.

The main drawback of the geometric calibration with a checkerboard chart is that it is very labor intensive and slow. In order for the calibration process to converge and provide an accurate characterization of the camera, the chart needs to be captured in a number of different poses, including different orientations with respect to the camera, and the resulting set of calibration points must cover well the entire field of view (FOV) of the camera. Obtaining points that are very close to image borders, however, can be challenging. Since a captured image can be used only if it shows the entire chart and all the checkerboard corners can be successfully detected, the chart needs to be carefully aligned to the FOV boundaries. When a larger number of cameras need to be calibrated, calibration chart image capture may quickly become impractical. This is particularly true in the case of wide FOV cameras, since they not only require capturing more calibration images, but also require more careful positioning of the chart due to the large geometric distortion. Proper lighting also presents a bigger challenge in case of systems with very large FOV, as light reflections on the chart can render images of chart in certain positions and orientations unusable.

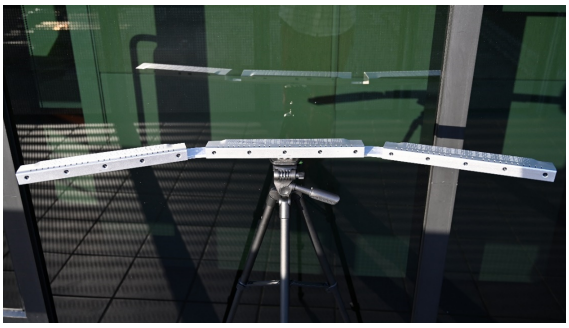

Figure 1. *Camera array comprising 15 miniature camera modules with the 120° field of view.*

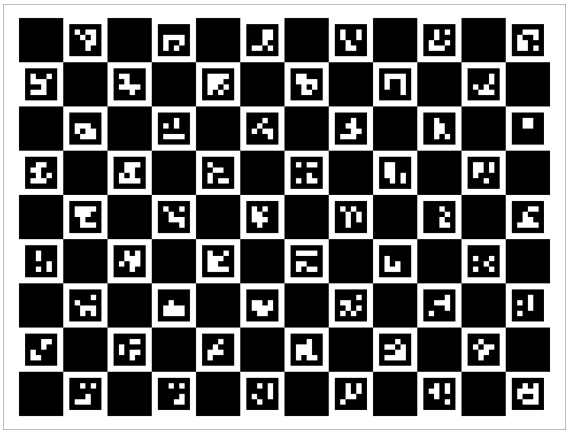

Figure 2. *ChArUco AprilTag chart for calibration employing the images of a partially visible chart.*

Besides making the calibration image capture more difficult, the requirement that all the checkerboard corner points must be successfully detected in an image leads to calibration points sets that are very center-heavy and contain only a small number of points close to image boundaries. This can have a negative impact on the accuracy of the estimated geometric distortion, as the outer regions of the image are under-represented and carry lower weight in the optimization producing the calibration. Moreover, in the regions where there are no or almost no points, the calibration can be completely off because highly flexible distortion models that are needed to accurately match strong distortion in modern high resolution cameras are very bad for extrapolation.

Creating calibration tools that can use also images in which

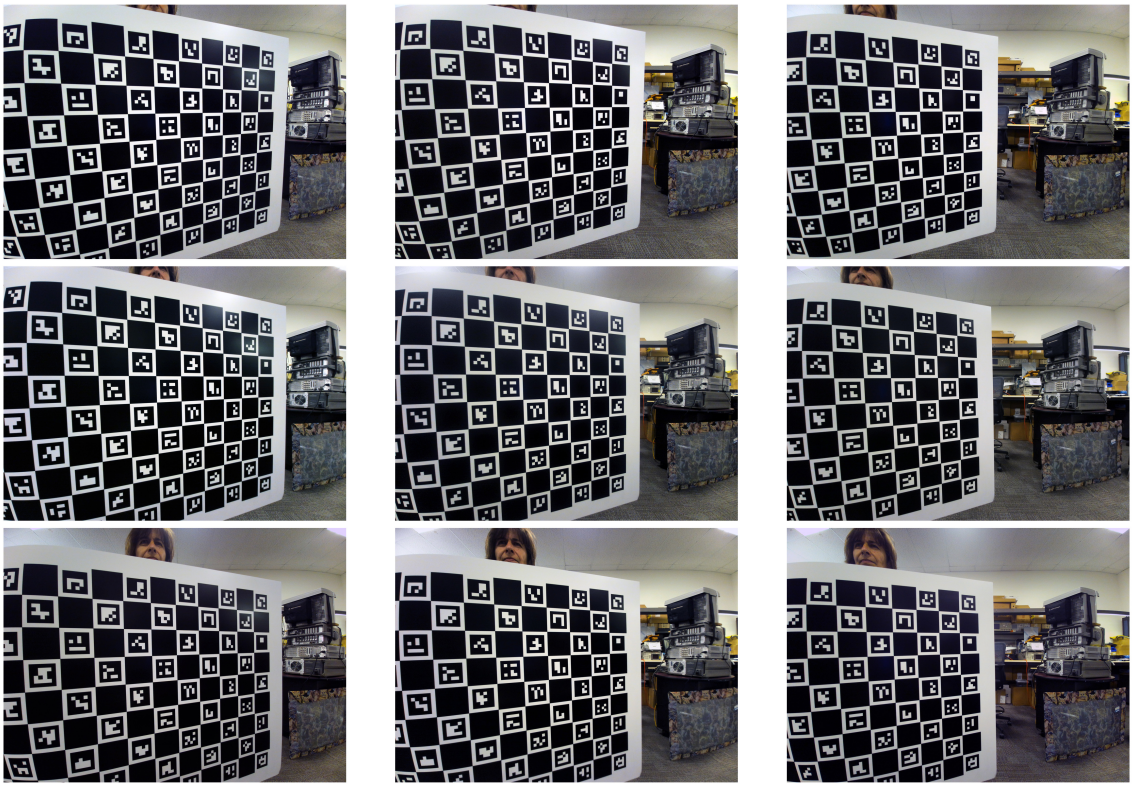

Figure 3. *When the calibration chart can be partially outside of the field of the view of a camera, a single chart pose can produce points close to the image boundaries for multiple cameras.*

some of the checkerboard corners are not visible or cannot be detected helps to make geometric calibration with a chart much more efficient as well as more accurate and reliable. It allows obtaining more points close to image borders without carefully positioning the chart, producing much better point sets with significantly less effort. While this is true also when a single camera is calibrated, when calibrating a multi-camera system that captures synchronized frames, this makes a particularly big difference. With such systems, when the chart is carefully aligned with respect to one of the cameras, for other cameras the chart position is less than ideal – the chart is either far from the border, or it is partly out of the field of view. When the images in which the chart is partly out the picture can be used, not only the number of captured images that get discarded drops to a small fraction, but a single chart position can produce points close to image borders for multiple cameras, as shown in Figure 3.

Calibration Chart Design and Point Detection

To be able to use images in which the chart is only partially visible, it is necessary to modify the test chart. The periodic checkerboard pattern needs to be replaced by a pattern making it possible to reliably identify what part of the chart has been detected in a captured image. The chart that we use has AprilTags [6] embedded in all white squares of the checkerboard, as shown in Figure 2. We use OpenCV ChArUco-ArUco library [7] for the chart design and the initial point detection. We found the OpenCV implementation better supported and suitable for our needs than other available alternatives, such as the Kalibr toolbox.

4x4 ArUco tag dictionary offers a sufficient number of tags to fill our chart and provides patterns that can be successfully detected even when the chart is farther from the camera, when it is distorted, and when the captured images are downsampled. In general, we found the tag detection to be very robust, except for a couple of the most simple tags which do not have distinct enough shape, especially the last one in the third row in Figure2. In spite of careful tuning, these tags can be occasionally mistaken for other features on the chart and elsewhere in the scene, producing gross outliers in the obtained point sets, which can break the calibration process. This tag and others that are similar to it either need to be avoided in the chart design, or the calibration process needs to be made robust to the outliers that they sometimes cause.

Although the new chart design contains more features that potentially could be detected and used for calibration, we choose to use only the checkerboard square corners. The main reason for this is that their position can be detected with higher accuracy than the position of the corners of the AprilTags and features within the tags. Also, the resulting point set is then a subset of the checkerboard chart point set, which makes adapting the existing algorithms and code easier.

The corner point detection requires subpixel refinement, which becomes computationally expensive for a large number of images. However, as each image can be processed completely independently, any software parallel processing scheme (e.g., OpenMP) with hardware optimized image processing li-

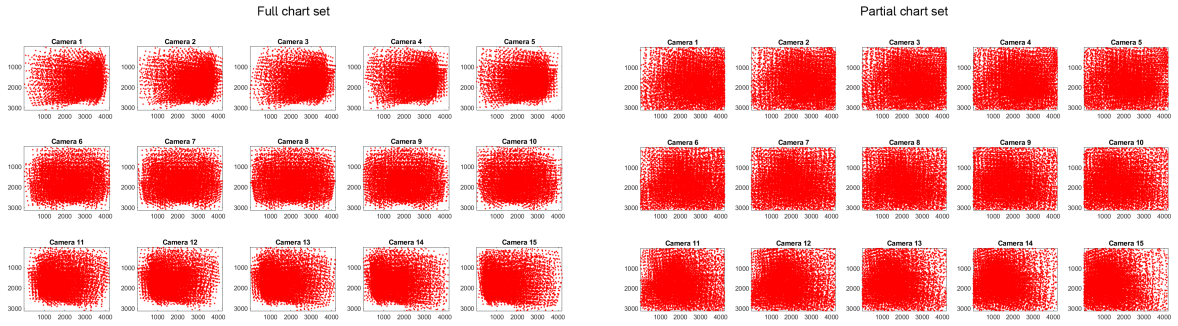

Figure 4. *Typical point distributions in calibration sets obtained with the traditional checkerboard chart, requiring that the entire chart is within the camera field of view (left), and when the chart that can be only partially visible (right). When the chart is allowed to be partially outside the field of view, it helps to produce more points in the outer regions of the camera field of view, all the way to image borders.*

braries (e.g., IPP, CUDA) can speed up the detection time significantly. The images also can be resized to achieve higher speed, but this tends to cause some loss of accuracy and we thus prefer to process full size images.

Geometric Calibration Algorithm

For the robust initialization of the calibration that can handle cameras with large distortion, including fisheye cameras, and does not require any camera specifications that may be hard to obtain, we use a modified version of the camera model proposed by Scaramuzza *et. al.* and their initialization algorithm [8]. Although it has been developed specifically for omnidirectional cameras, it works well also for other cameras, including those with normal FOV.

For a given point (x,y) in the ideal image plane, with the distance $\rho = \sqrt{x^2+y^2}$ from the principal point, the model provides a point $(x,y,f(\rho))$ on the ray that gets projected to this point. We use a slightly different model than [8] and set

$$f(\rho) = \sum_{m=0}^{M} f_{2m}\rho^{2m}, \tag{1}$$

as it tends to yield more accurate calibrations with the same number of parameters. If $\mathbf{R}_j$ and $\mathbf{t}_j$ is the rotation and translation of the chart in the j-th calibration image, then for the n-th chart point (X_n, Y_n) and the corresponding point $(x_{n,j}, y_{n,j})$ extracted from j-th calibration image we have

$$\mathbf{R}_j \begin{bmatrix} X_n \\ Y_n \\ 0 \end{bmatrix} + \mathbf{t}_j = d_{n,j} \begin{bmatrix} x_{n,j} \\ y_{n,j} \\ f(\rho_{n,j}) \end{bmatrix}, \tag{2}$$

with some unknown distance $d_{n,j} > 0$. This unknown distance can be eliminated by taking the cross product of the left and right hand side and setting it equal to zero. One of the three resulting equations,

$$(r_{11}^{(j)}X_n + r_{12}^{(j)}Y_n + t_1^{(j)})y_{n,j} - (r_{21}^{(j)}X_n + r_{22}^{(j)}Y_n + t_2^{(j)})x_{n,j} = 0, \tag{3}$$

does not involve the unknown distortion polynomial f and is linear in the unknowns that it involves. When these equations are collected for all chart points detected in an image, the resulting system of linear equations can be solved in the least squares sense, producing the estimates of rotation and translation coefficients $r_{11}^{(j)}$, $r_{12}^{(j)}$, $r_{21}^{(j)}$, $r_{22}^{(j)}$, $t_1^{(j)}$, and $t_2^{(j)}$ up to an unknown scale factor. However, this scale factor and the remaining rotation coefficients are uniquely determined because $\mathbf{R}_j$ is a rotation matrix satisfying $\mathbf{R}_j\mathbf{R}_j^T = \mathbf{I}$ and $\det(\mathbf{R}_j) = 1$. The obtained rotation and translation coefficients make it possible to calculate the unknown distances $d_{n,j}$ by using the top two rows in (2). Then, the last rows of (2) for all n and j can be collected, forming a single linear system that can be solved to estimate the coefficients of polynomial f and $t_3^{(j)}$ for all j.

This algorithm does not rely on any particular arrangement of the points on the chart, it only requires that the points are not collinear. It can be adapted to the case of partially visible charts in a straightforward manner, collecting only the equations corresponding to successfully detected chart points and allowing the systems of linear equations to be of different sizes. To avoid degenerate cases and cases where these initial estimates may not be accurate enough, we include only images in which at least 25% of the chart points have been successfully detected. Also, we exclude the corner points adjacent to AprilTags with the most simple shapes that are prone to misdetection, as these gross outliers cause the algorithm to fail.

To obtain the final calibration, the estimated initial distortion is first converted to the target model used for calibrating the particular camera. Many different models that can be used have being proposed in the literature. See, e.g., [9], [10], [11], [12], [13], or [14]. The parameters of the camera model and the extrinsic parameters, three rotation angle parameters and three translation parameters representing the chart position in each image, are refined by a non-linear optimization process. We use the most common approach, used by Zhang, [15], and many others, minimizing the total square reprojection error. The resulting non-linear least squares problem can be solved, for example, by using Levenberg-Marquardt algorithm (see, e.g., [16]). This process can also be easily adapted to the partially visible chart case. It only requires making it possible to use a variable number of points per image and including in the cost vector only the successfully detected points.

Other cost functions also can be used, similar to those proposed in literature for bundle adjustment. For example, when

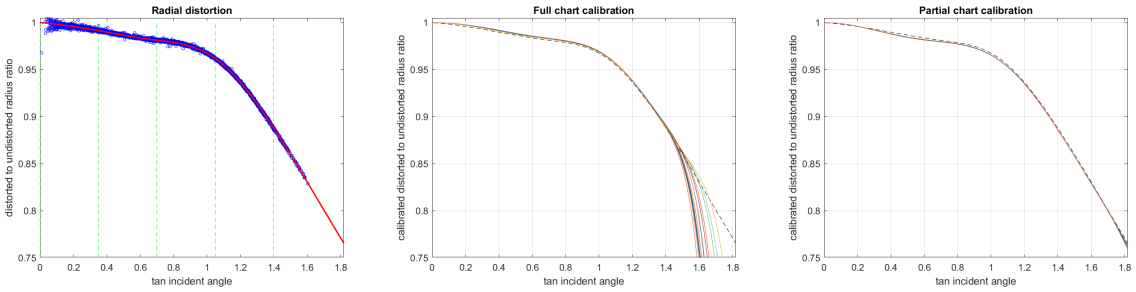

Figure 5. *Left to right: (a) Ground truth radial distortion spline curve modeling the distortion observed in our camera modules. (b) The result of radial distortion calibration using full chart images. (c) The result of radial distortion calibration using images in which the chart may be only partially visible.*

backprojection type models are used, such as the fisheye model of Scaramuzza *et al.* or our modified version of it (1), a cost function involving angular error [17] or other spatial domain error can be much more computationally efficient than the reprojection error. We have also experimented with robust costs that can reduce the impact of outliers in the data, similar to those described in [18] and [19]. However, we have found these robust costs unreliable for our purposes. They are unable to distinguish between true outliers in the data and the data points that are valid but are challenging for the selected camera model to match. In the case of difficult distortions, instead of improving the robustness of the calibration, they can undermine it by diminishing the contribution of certain points that are critical for its success.

Our tool can also perform joint calibration of the entire array, producing simultaneously estimates of intrinsic and extrinsic parameters of all the cameras, which are very robust thanks to incorporating multi-camera geometry constraints. After each individual camera is calibrated, to initialize the array calibration it is necessary to turn the pair-wise chart-to-camera rotation and translation estimates into a set of geometrically consistent rotations and translations describing each camera position and chart pose in the selected global coordinate system. This is a rotation averaging problem, common in SFM and SLAM, for which a number of solutions have been proposed (see, e.g., [20], [21]). We take the advantage of the fact that in our case the relative poses produced by calibrating individual cameras are always between a chart and a camera and improve the estimated global chart and camera poses iteratively, alternating between updating the camera poses and the chart poses. The variable number of calibration points per image plays no role here at all and no modifications are thus needed to accommodate partially visible charts.

The final joint non-linear refinement of all parameters that follows is then performed by a technique similar to the non-linear refinement for an individual camera – by minimizing the total square reprojection error or other cost that includes valid points only. The joint optimization of all parameters is costly, but it produces the most accurate results. We have observed that in situations when the accuracy of the camera model used for the calibration is limited, for example when a central projection model is used for calibrating fisheye cameras that exhibit some incident angle dependent pupil shift, when an individual camera is calibrated, errors in intrinsics and extrinsics may partly offset each other. Joint calibration of the array forces the extrinsics to be geometrically consistent and helps to prevent this. Furthermore, having an initial calibration for the array that is robust and already relatively accurate makes it possible to include in the final refinement also points that were previously excluded because they presented a risk in the earlier stages. We check the corner points adjacent to AprilTags that occasionally get misdetected and include those that appear to be correct in the final refinement. Also, the points from images in which only a small part of the chart is visible can be incorporated in the joint refinement, if another camera could see at least 25% of the points with the same chart pose. This helps to increase the number of points in outer regions of the camera field of view used for the calibration.

Experiment Results

1. Point Distribution

To demonstrate the impact of incorporating in the calibration images in which the chart is only partially visible on the accuracy and the reliability of the calibration, we are going to show here the results of experiments with both simulated and real-life data that analyze the performance for multi-camera devices that we built. Figure 1 shows a camera array consisting of three such devices that have been daisy-chained. Each of them includes five 13 Mpix miniature camera modules with diagonal field of view approximately 120°.

Figure 4 illustrates the difference in the typical spatial distribution of points in the calibration image sets captured with such an array. In this example, both sets were obtained by capturing the chart in 120 different positions, trying to cover well the entire field of view of all the cameras. In the first case, we have kept the chart within the FOV of the array and extracted chart points only from the images in which the full chart is visible, as it is done when using the commonly available calibration tools. In the second case, we have intentionally exceeded the FOV of the array and extracted all visible checkerboard corner points. As you can see, the first approach produces only a small number of points near the image boundaries and, without going to extreme measures that would make the image capture impractical, some areas may get missed completely. The second set, comprising images in which the chart may be only partially visible, not only provides significantly more points near the image boundaries, but also capturing it was much easier, as it did not require aligning the chart to image borders.

Table 1: Real-life data experiment, extrinsics re-calibration mean reprojection error

	Full chart calibration			Partial chart calibration		
	3 RD params	3 RD params	6 RD params joint	3 RD params	6 RD params	6 RD params joint
MRE (pixels)	3.76	1.73	1.38	3.08	0.84	0.83

Table 2: Real-life data experiment, point position prediction error (in pixels)

	Full chart calibration			Partial chart calibration		
	3 RD params	3 RD params	6 RD params joint	3 RD params	6 RD params	6 RD params joint
median	3.29	1.78	1.72	2.85	1.15	**1.01**
90%	11.77	5.52	5.14	9.61	3.32	**2.72**
99%	52.09	18.27	16.31	25.25	8.75	**7.43**
99.9%	812.27	515.19	382.89	57.34	15.63	**13.55**

2. Calibration with Synthetic Data

For the calibration experiments described in this paper, both with synthetic and real data, we have used the most common distortion model, supported by both OpenCV [4] and Matlab [5],

$$x_d = x_u(1 + K_1 r_u^2 + K_2 r_u^4 + \cdots) + 2P_1 x_u y_u + P_2(r_u^2 + 2x_u^2) \quad (4)$$

$$y_d = y_u(1 + K_1 r_u^2 + K_2 r_u^4 + \cdots) + P_1(r_u^2 + 2x_u^2) + 2P_2 x_u y_u.$$

Here, (x_u, y_u) are the coordinates of the scene point projected to plane $z = 1$ and (x_d, y_d) are the coordinates of the corresponding distorted point, which is then mapped to image pixel coordinates using the intrinsic matrix. Our experiments have shown, though, that three radial distortion coefficients K_m that are commonly used and supported by both OpenCV and Matlab, are not sufficient for accurate enough characterization of the distortion of our wide FOV camera modules (see our real-life data experiment results in the next section). We have thus increased their number to six.

Our synthetic data experiment simulates an array similar to the one shown in Figure 1, formed by three our five-camera devices. To make the camera parameters and the chart positions realistic, we have used for generating the synthetic image points the intrinsic and extrinsic parameters obtained when calibrating a similar real-life array, only for the radial distortion we have replaced the calibration output by a cubic spline curve approximately matching the distortion observed in the real camera modules, see Figure 5 (a). This spline curve provides a realistic distortion approximation that is not biased towards the model used for the calibration. For both the calibration with full chart images only, and for the one including images of partially visible chart, we simulated points corresponding to the chart being captured by all the 15 cameras forming the array in 120 different poses, which also come from our experiments with a real-life array. The resulting point sets are similar to shown in Figure 4. Gaussian noise with the standard deviation of 0.5 pixels has been added to the generated image point coordinates to simulate limited accuracy of point detection.

When these point sets are used for calibration, the mean reprojection error (MRE) is about 0.6 pixels in the case of full chart set, and 0.8 pixels in the case or partial chart set. This may give an impression that the full chart set provides more accurate calibration. However, the opposite is true. The MRE not only does not provide a complete picture, but also is not directly comparable in this case because it is measured in each case using a different set of points. The error is deceptively low in the case of the full chart image set because this set lacks points in the regions that are most challenging for the calibration model to fit. In reality, the calibrated geometric distortion is completely invalid in the areas near the corners of the frame, as shown in Figure 5 (b). This type of calibration error, although it impacts a relatively small part of the image frame, can cause serious problems in computational imaging and other applications, as it can cause bad artifacts or make algorithms to fail. When the partial chart set is used for calibration, the outer regions of the image frame are sufficiently covered with the detected points, and this problem is avoided. The calibrated distortion curves look as shown in Figure 5 (c).

3. Real-life Data Experiment

For the real-life data experiment we have used two our devices, forming an array of 10 cameras. In the absence of ground truth for the calibration we have used for the objective evaluation of the quality of the resulting calibration a simple practical 3D computer vision task, the accuracy of which we can measure. Our experiment simulates the way the array is used in practice. The array calibration obtained by the method described in this paper serves as the "factory" calibration for the array, providing the intrinsic parameters for each of the cameras that are stored within every device. Each time the array is used for capturing some content, the camera positions are re-calibrated using these intrinsic parameters and a faster calibration process that requires capturing only a small number of images.

The obtained calibrations have been evaluated using the same test image set consisting of 60 frames (i.e., 600 images in total). A small set of five frames captured at the same time as the test set was used for re-calibrating the array extrinsics. Similarly as the test image set, the set used for re-calibration was the same for all compared factory intrinsic calibrations. The intrinsic calibrations have been produced using calibration image sets de-

picting the calibration chart in 120 different positions. Similarly as in the previous experiments, we have produced two separate calibration sets, one for the traditional checkerboard chart calibration using only images showing the entire chart, and one taking the advantage of the chart being allowed to be partly outside of the field of view. We compare both types of calibration using the standard geometric distortion model (4) with 3 and 6 radial distortion coefficients, respectively, and in the case of 6 coefficients, we compare also the individual camera calibration and the result of joint optimization of the parameters for all the cameras.

The differences in the accuracy of the calibrated intrinsics start showing already during the re-calibration of camera positions. As shown in Table 1, the MRE during re-calibration is noticeably higher in the case of the calibration with full chart images. It exceeds one pixel, suggesting possible issues with the accuracy of calibration. While the calibration MRE may not provide the full picture, here it is more indicative than in our synthetic data experiment, as it is calculated using the same set of points. We also note that the MRE is high in both cases when only three radial distortion coefficients are used, which indicates that this common distortion model with three radial distortion coefficients only is not enough for these cameras.

The task that we have used for evaluating the accuracy of calibration involves using stereo pairs to estimate the positions of points in 3D space and then predicting their location in images captured by other cameras. In the captured test image set, we have detected 37,960 points, distributed fairly uniformly over the entire area of each image frame, which form 4139 trails of 3 or more matching image points. With this point set, we could make the total of 1,230,105 matching image point position estimates that we could compare to the real detected matching point positions. The statistics of the error of these predictions when different factory calibrations are used are shown in Table 2. Some errors exceeding one pixel are to be expected here, as in some cases the 3D position is triangulated using a pair of cameras that are close to each other, but this estimate is then used to predict the matching point location in the image captured by a camera that is much farther away. However, we see that in the case of full chart calibration between 0.1% and 1% of cases the position estimate is completely off, making it impossible to find the correct matching point by searching the neighborhood of the predicted location. The calibration with partial chart images both helps to resolve this issue, and improves the overall accuracy.The results in Table 2 also confirm that more than three radial distortion coefficients need to be used for these cameras and that the joint optimization for the entire array provides most accurate calibration.

Comments and Conclusions

Geometric calibration with a chart is inherently a slow process and is suitable only for off-line use. The main bottleneck is the image capture and the robustness and accuracy of the calibration are more important than the software execution time. Our method of calibration with images of partially visible chart makes the calibration image capture significantly easier and faster. It also helps to improve the robustness and accuracy of the calibration. The Matlab prototype of the calibration algorithm that we have implemented is already sufficient for the practical use in our conditions, even though it has not been optimized for speed. It does not require any human intervention to produce reliable results and even the joint optimization for the entire array is fast enough not to create a new bottleneck in the calibration process. As we have shown, the modifications that are required generally involve only keeping track of the points successfully detected in the calibration images and using arrays of variable size. With proper implementation optimized for speed the computation times should be comparable to existing standard calibration tools. Some additional computations are required in the point detection stage. With our existing code, on a laptop with Intel(R) Core(TM) i7-8850H CPU @2.60Ghz 6 Core processor, with 48GB or memory, running Windows 10, the chart point detection takes about 0.2 s per image with the 13 Mpix images captured by our array and the chart design shown in Figure 2. As we have mentioned in the section discussing the point detection, there is still room for improvement and higher speeds are achievable.

The calibration tool described in this paper is still under development. However, it has already helped us to make our multi-camera system calibration more efficient, reliable, and accurate, and it has provided many valuable insights for further improvements.

References

[1] Dietmar Wueller, A new dimension in geometric camera calibration, Electronic Imaging, Image Quality and System Performance XVII (2020).

[2] Ryusuke Sagawa, Masaya Takatsuji, Tomio Echigo, and Yasushi Yagi, Calibration of lens distortion by structured-light scanning, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 832-837 (2005).

[3] Hyowon Ha, Michal Perdoch, Hatem Alismail, In-So Kweon, and Yaser Sheikh, Deltille grids for geometric camera calibration, Proc. of IEEE International Conference on Computer Vision (ICCV), pp. 5354-5362 (2017).

[4] OpenCV, Camera Calibration and 3D Reconstruction, *https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html*.

[5] MathWorks, Matlab documentation, Camera Calibration, *https://www.mathworks.com/help/vision/camera-calibration.html*.

[6] Edwin Olson, AprilTag: A robust and flexible visual fiducial system, Proc. of IEEE International Conference on Robotics and Automation (ICRA), pp. 3400-3407 (2011).

[7] OpenCV, Detection of ChArUco Boards, *https://docs.opencv.org/3.4/df/d4a/tutorial_charuco_detection.html*.

[8] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart, A toolbox for easily calibrating omnidirectional cameras, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5695–5701 (2006).

[9] Duane C. Brown, Decentering distortion of lenses, Photogrammetric Engineering, 32(3):444–462 (1966).

[10] David Claus and Andrew W. Fitzgibbon, A rational function lens distortion model for general cameras, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. 213–219 (2005).

[11] Donald Gennery, Generalized camera calibration including fish-eye lenses, International Journal of Computer Vision, 68:239–266 (2006).

[12] Juho Kannala and Sami Brandt, A generic camera model and calibration method for conventional, wide-angle, and fisheye lenses, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(8):1335–1340 (2006).

[13] Srikumar Ramalingam and Peter Sturm. A unifying model for cam-

era calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(7):1309–1319 (2016).
[14] Johannes Beck and Christoph Stiller, Generalized B-spline camera model, Proc. of IEEE Intelligent Vehicles Symposium (IV), pp. 2137-2142 (2018).
[15] Zhengyou Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334 (2000).
[16] Richard Hartley and Andrew Zisserman, Multiple view geometry in computer vision, Cambridge University Press (2003).
[17] Juho Kannala, Sami Brandt, and Janne Heikkilä, Self-calibration of central cameras by minimizing angular error, Proc. of 3rd International Conference on Computer Vision Theory and Applications (VISAP), pp. 28-35 (2008).
[18] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon, Bundle adjustment – A modern synthesis, Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science, vol. 1883, pp. 198–372. Springer Verlag (2000).
[19] Christopher Zach, Robust bundle adjustment revisited, Proc. of European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, vol. 8693, pp. 772–787, (2014).
[20] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li, Rotation averaging, International Journal of Computer Vision, 103(3):267–305 (2013).
[21] Avishek Chatterjee and Venu Madhav Govindu, Robust relative rotation averaging, IEEE Trans. Pattern Anal. Mach. Intell., 40(4):958–972 (2018).

Author Biography

Radka Tezaur received her RNDr in Numerical Analysis from the Charles University (1990) and her PhD in Mathematics from the University of South Australia (1996). She was a Principal Engineer at the Nikon Research Corporation of America and currently she is a member of the Computational Imaging Laboratory at Intel. She served as a Vice President of the IS&T Board of Directors (2015-2019) and was the General Chair of IS&T Electronic Imaging Symposium in 2020.

Gazi Ali is a research scientist at Computational Imaging Lab which is a part of Intel Labs. Previously he was a member of Sony US research lab, Nikon research corporation of America, and Elo touch solutions. Gazi received his Ph.D. in Electrical and Computer Engineering from Purdue University (2007). His research interest includes camera hardware, computer vision, optics, and imaging algorithms.

Oscar Nestares is a Principal Engineer at Intel Labs working on computational imaging algorithms and lightfield processing. Before he was a tenured Research Scientist (Institute of Optics, CSIC) working on visual system models and biologically inspired image/video processing; a Fulbright Scholar at Stanford University; and consultant at Xerox PARC. He received his M.S. (1994) and Ph.D. (1997) in Telecommunication Engineering from Universidad Politécnica de Madrid. He has published 30+ international papers and 40+ patents.