Calibration and Auto-Refinement for Light Field Cameras

Yuriy Anisimov[1,2], Gerd Reis[1], Didier Stricker[1,2]

[1]German Research Center
for Artificial Intelligence,
Trippstadter Str. 122
67663 Kaiserslautern,
Germany

[2]University of Kaiserslautern,
Gottlieb-Daimler-Str. 47
67663 Kaiserslautern,
Germany

{firstname.lastname}@dfki.de

ABSTRACT

The ability to create an accurate three-dimensional reconstruction of a captured scene draws attention to the principles of light fields. This paper presents an approach for light field camera calibration and rectification, based on pairwise pattern-based parameters extraction. It is followed by a correspondence-based algorithm for camera parameters refinement from arbitrary scenes using the triangulation filter and nonlinear optimization. The effectiveness of our approach is validated on both real and synthetic data.

Keywords

light field, camera calibration, camera rectification, calibration refinement

1 INTRODUCTION

Light field cameras [Ng05a] utilize a multi-view principle, focusing incoming light on the image sensor over a grid of lenses. It creates a set of proportionally shifted images that can be used to reconstruct a three-dimensional representation of the captured scene.

These are two types of light field cameras. As proposed by Adelson and Wang in [Ade92a], a light field camera can be created from an array of micro-lenses placed in front of the camera sensor. An alternative way of constructing the light field camera was presented by Wilburn *et al.* in [Wil05a] and involves placing an array of ordinary cameras with predetermined proportional distances between them. A different principle, which can be described as intermediate between the previous two, is represented in [Ani19a]. There, the light field camera is built on a single camera sensor with an array of full-size lenses placed in front of it.

Among the various applications of light fields, the estimation of depth maps attracts particular attention, as the large number of viewpoints increases the quality of the reconstruction and the simple geometry of the light fields simplifies the calculations. Due to the inaccuracies in the placement of light field camera components

(micro-lens array or single-view cameras) the direct estimation of scene depth information will not be accurate. Natural lens distortion also affects the quality of the result. There are two ways to solve these problems. For the reconstruction algorithm, either the exact lens positions and geometry should be taken into consideration, or the images from such a camera can be rectified to assure constraints of images' common rotation and proportional placement.

One of the simplifications in reconstruction from multi-view systems has to do with limiting the search for matching correspondences to specific scan lines instead of searching the entire image. Therefore, the rectification-based way of lens placement compensation should be used in the multi-view systems if their view placement allows the rectification without vanishing of significant image plane parts.

Camera calibration, or camera resectioning [Har03a], is a process of retrieving camera intrinsic and extrinsic parameters. Usually, the intrinsic parameters include the focal length, coordinates of a principal point, axis skew, and distortion coefficients. Extrinsic parameters contain information about the pose of the view in the form of rotation and translation *w.r.t.* the scene origin.

These parameters may change slightly under the mechanical or thermal influences during the operation of the light field cameras. In general, the light field camera must be recalibrated to cope with such changes, but sometimes this is not possible due to camera operating limitations or conditions. In such cases, the calibration data can be checked and corrected using arbitrary scene information.

We present an approach for light field camera calibration and rectification, which is based on a well-known pattern-based method from Zhengyou Zhang [Zha00a]. The results of the calibration are used to generate look-up tables that are applied to the light field views for their rectification. The key feature of the method is its simplicity of implementation since this calibration requires a checkerboard pattern that can be produced on any conventional printer.

At the same time, designed a nonlinear optimization model for estimating the compensation of changes in the position of the light field camera lenses. This method uses features extracted from an arbitrary scene.

To improve their accuracy, we introduce a triangulation-based correspondences filtering method, as well as a chaining method for tracking them in different light field views.

Our algorithms were verified on a light field camera with single full-size lenses in an array. It can be potentially used in micro-lens-based cameras, however, the potential misplacement of lenses in a manufactured array is not as large as in cameras with full-size lenses.

The paper is structured as follows. An overview of previous work is given in Section 2, Sections 3 and 4 outline the proposed methods, and experimental results for them are presented in Section 5, followed by a conclusion in Section 6. To prevent the term "camera" from being unambiguous, we use it in reference to a light-field camera; and each image taken from a particular lens is called a "view".

2 RELATED WORK

The first paper describing the principles of light field calibration for camera arrays was proposed by Vaish *et al.* in [Vai04a]. Using planar parallax, the authors retrieve the relative positions of the light field views and use them for further processing.

Extension of stereo calibration and rectification principles for multi-view cameras is presented in the work of Kang *et al.* [Kan08a]. In our work similar principles were used for defining the common values of rectification parameters. Approach of Xu *et al.* [Xu15c] optimizes all parameters of light field camera simultaneously by constrained bundle adjustment model.

Most of the approaches for light field calibration consider usage of micro-lens-based cameras. They can't be applied directly to camera arrays due to the fact that the shift of the lenses relative to each other in such cameras is much higher than in the plenoptic cameras.

In the work of Bergamasco *et al.* [Ber15a] a parameter-free camera model is used for the light field rays representation for further triangulation-based calibration. An approach of Jin *et al.* [Jin16a] estimates the centers of sub-images, aligns all views in a grid, and performs
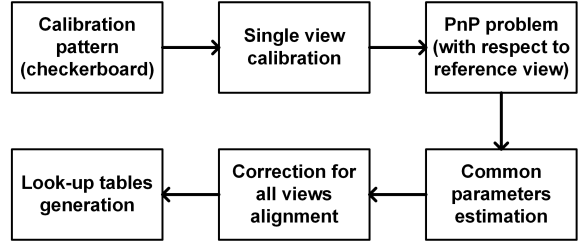

Figure 1: A pipeline of pattern-based calibration algorithm

the rotation of all views to a common plane. Bok *et al.* in [Bok16a] extracts lines from the raw-captured calibration pattern and attempts to compensate for lens distortion and misplacement inaccuracies using these features.

Likewise, Noury *et al.* in [Nou17a] employ the raw images; however, they use corners as the features for calibration procedure along with nonlinear optimization of the result. Method of O'Brien *et al.* [Obr18a] extracts disc features instead of conventional corners for retrieving the light field calibration data.

An approach by Sun *et al.* [Sun19a] works with three calibration targets at different distances together with gradient-based correspondences search. In the publication of Zhou *et al.* [Zho19b], the calibration problem is solved by estimating the original homography solution using a calibration scheme and its subsequent nonlinear optimization. Ji and Wu in [Ji19c] propose a calibration model for plenoptic cameras calibration with a conventional calibration target.

3 CALIBRATION ALGORITHM

A calibration algorithm estimates the intrinsic and extrinsic parameters of all light field views. This data is further used for the views rectification, which creates constrains for simplification of the depth estimation. The pipeline of this method is presented in Fig. 1.

3.1 Camera model

For a separated view of the light field a pinhole camera model is used [Har03a]. The general relation of two-dimensional (2D) image points and three-dimensional (3D) scene points can be described as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

$$P = K[R|t], \; m = PM \quad (2)$$

where m corresponds to homogeneous image point, K – camera calibration matrix, which consists of pixel focal

lengths f_x and f_y, principal point coordinates c_x and c_y, and axis skew s. R and t correspond to rotation matrix and translation vector respectively. These components are assembled into a projection matrix P. M stands for homogeneous world point.

This representation does not take the lens distortions into account. Therefore, in our model we compensate the radial distortions up to their second order by [Har03a]:

$$x = x_c + (x_n - x_c)(1 + k_1 r^2 + k_2 r^4)$$
$$y = y_c + (y_n - y_c)(1 + k_1 r^2 + k_2 r^4) \qquad (3)$$
$$r^2 = (x_n - x_c)^2 + (y_n - y_c)^2$$

where $\{x; y\}$ are the undistorted points, $\{x_n; y_n\}$ correspond to original image points, $(x_c; y_c)$ stands for radial distortion center, r is the distance from this center to the distorted image point, and k_1 and k_2 are the radial distortion coefficients.

3.2 Pattern-Based Calibration

A method, which uses a planar calibration pattern, was originally described by Zhengyou Zhang in [Zha00a]. The interest points of a calibration pattern should be such that they can be easily found by a corner detection algorithm. A pattern with a known structure, such as a checkerboard, is used. The measured physical size of the pattern elements is required for solving a task of view position determination with respect to the calibration target. It is used further to provide the correct values of the focal length, with which the pattern was captured.

First, the corners of the located checkerboard need to be found. Due to the high contrast of the black-and-white pattern, its elements can be easily segmented based on histogram analysis. Corners' position is determined by the algorithm from [Suz85a]. A sub-pixel refinement step, described in [Foe87a], is used for obtaining accurate correspondences. This information is used to find the optimal intrinsic parameters of each view.

Having a set of points $\{m\}$, which size corresponds to the number of corners detected in the checkerboard, and its 3-dimensional matches $\{M\}$, common for all images, the algorithm computes the intrinsic matrix K and the distortion coefficients vector d in a way that these values reduce the reprojection error between the original and projected points with K and d, assuming that images are taken with zero rotation and translation *w.r.t.* scene origin.

By using Eq. 2 with estimated intrinsic matrix K, and $R = I, t = [0\,0\,0]$, where I stands for identity matrix of size 3×3, the reprojections of 3D points can be found. They are used afterward for estimating the root-mean-square error between originally detected points and the reprojected ones.

The 3D coordinates of pattern points $\{M\}$ are assumed to be known. Their placement is defined by physical distances between real-world corners of the checkerboard, which can be expressed from the size of squares.

The algorithm results in the intrinsic values (focal length, camera center) of each light field view together with the distortion coefficient per lens. In order to reduce the reprojection error, the result is further optimized using the Levenberg-Marquardt algorithm [Mar63].

It has been empirically established that this method requires dozens of images with different pattern positions to accurately estimate the calibration parameters. The order of magnitude of this number can be explained by the need for the entire image space to be covered by the sum of the pattern positions among all captured frames.

Results of single-camera calibration are further used for estimation of the relative parameters of every view *w.r.t.* the reference. During this step we fix extrinsic parameters of the reference view, so that $R_{ref} = I, t_{ref} = [0\,0\,0]$, and compute relative rotation and translation of every other view $\{R_i; t_i\}, i = 1..N$, where N is a total number of views in the camera, based on their position in relation to scene center. It is possible since all views capture the same pattern position in every image.

By having an assumption of known 3D points the Perspective-n-Point (PnP) problem [Fis81a] is solved for obtaining the rotation and translation of every view *w.r.t.* reference. As before, the results are subject to Levenberg-Marquardt optimization.

In the end, we obtain a set of original rotation and translation vectors $\{r^O, t^O\}$, which are used for the processing in Section 4. Rotation vectors are obtained from rotations matrices R^O by applying the Rodrigues transformation [Rod40a]. It is used for the simplicity of calculations in the next steps.

3.3 Multi-View Rectification

The next step of the presented algorithm is the estimation of optimal values of the intrinsic and extrinsic parameters for all views. It is necessary for the alignment of all views on the same image plane with proportional distances between every view.

First, all light field views should be rectified with a common intrinsic matrix K_r, which is defined as the following:

$$f_r^x = \frac{\sum_{i=1}^{N} f_i^x}{N}; f_r^y = \frac{\sum_{i=1}^{N} f_i^y}{N};$$
$$c_r^x = \frac{w}{2}; c_r^y = \frac{h}{2};$$
$$K_r = \begin{bmatrix} f_r^x & 0 & c_r^x \\ 0 & f_r^y & c_r^y \\ 0 & 0 & 1 \end{bmatrix}, \qquad (4)$$

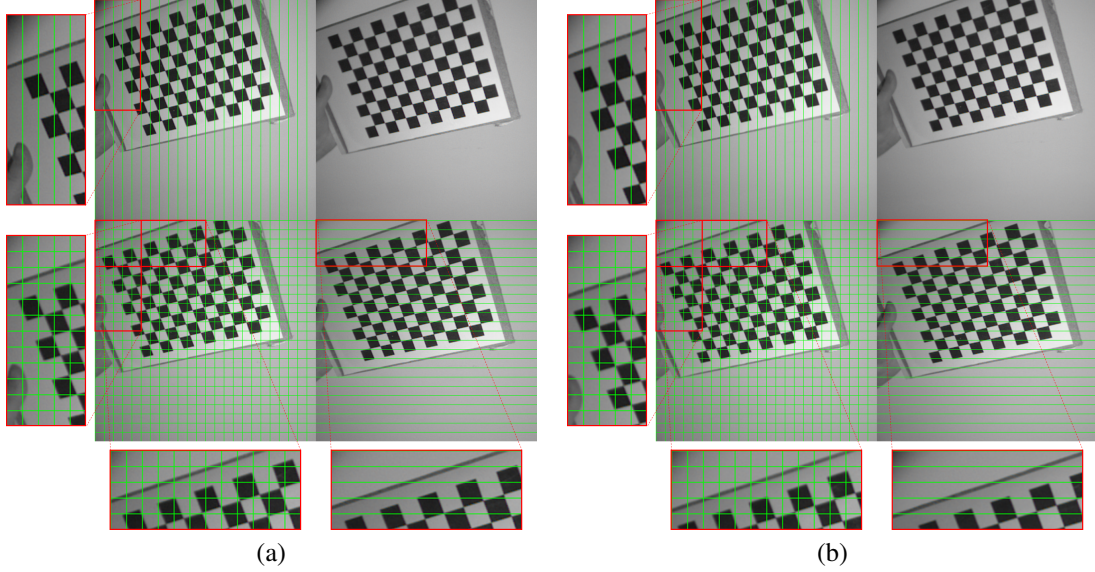<center>(a)</center> <center>(b)</center>

Figure 2: A subset of (a) original and (b) rectified light field after applying the algorithm from Section 3. The corners of the checkerboard are properly aligned, which confirms the rectification

where N is the number of light field views, and w, h correspond to width and height of every view.

Rotation and translation vectors $\{r^O, t^O\}$ are used for getting an optimal value of common rotation and translation vector, to which all views would be brought by the rectification.

Common rotation vector r_r is estimated as an average of all rotation vectors as:

$$r_r = \frac{\sum_{i=1}^{N} r_i^O}{N-1}, \tag{5}$$

and further converted to rotation matrix R_r by applying Rodrigues transform.

Computations of common translation vector involve the spatial information of every view. Since the translation vectors are defined *w.r.t.* reference view, we estimate the relative values by involving a and b as vertical and horizontal spatial dimensions of light field, where $a \times b = N$.

For a specific light field view $i = 1..N$, the translation vector t_i^V is formed from per-axis components as $t_i^V = [t_i^{V_x} \ t_i^{V_y} \ t_i^{V_z}]$ and estimated as a mean of all components not in the same row or column as:

$$\bar{a} = \lfloor i/a \rfloor; \bar{b} = i \bmod b;$$
$$t_i^{V_x} = \begin{cases} t_i^{O_x}/(\hat{b}-\bar{b}), \hat{b}-\bar{b} \neq 0 \\ 0, \hat{b}-\bar{b} = 0 \end{cases}$$
$$t_i^{V_y} = \begin{cases} t_i^{O_y}/(\hat{a}-\bar{a}), \hat{a}-\bar{a} \neq 0 \\ 0, \hat{a}-\bar{a} = 0 \end{cases} \tag{6}$$
$$t_i^{V_z} = 0$$

where $\hat{a}$ and $\hat{b}$ stand for spatial coordinates of reference view, defined in Section 3.2.

All translation vectors form a set t^V. Common translation vector $\bar{t}_r = [t_r^x \ t_r^y \ 0]$ is estimated as:

$$S_{t^x} = \sum_{i=1}^{N}(1-\delta_{t_i^{V_x},0}); S_{t^y} = \sum_{i=1}^{N}(1-\delta_{t_i^{V_y},0})$$
$$t_r^x = \frac{\sum_{i=1}^{N} t_i^{V_x}}{S_{t^x}}; t_r^y = \frac{\sum_{i=1}^{N} t_i^{V_y}}{S_{t^y}} \tag{7}$$

where S_{t_x} and S_{t_y} are numbers of particular non-zero components in t^V, found using Kronecker delta:

$$\delta_{x,y} = \begin{cases} 0, x \neq y \\ 1, x = y \end{cases}. \tag{8}$$

This vector is a basis for the set of per-view translation vectors t^p:

$$t_i^{p_x} = t_r^x(\hat{b}-\bar{b}); t_i^{p_y} = t_r^y(\hat{a}-\bar{a}); t_i^{p_z} = 0. \tag{9}$$

Using the results from Eq. 4–9 we estimate the rectified projection matrix P_r per every light field view $i = 1..N$ as [Har03a]:

$$\bar{R}_i = R_r R_i^{O^T}$$
$$\bar{t}_i = \bar{R}_i t_i^p \tag{10}$$
$$P_i^r = K_r[\bar{R}_i | \bar{t}_i]$$

The resulting camera projection matrix represents the ideal position of its corresponding light field view.

The further remapping of pixels to the proper position is done by applying a look-up table, which stores per-pixel coordinates of the rectified image and generated using P_i^r, distortion coefficients and original intrinsic and extrinsic values. Fig. 2 shows, how light field images changes after applying the described pipeline.
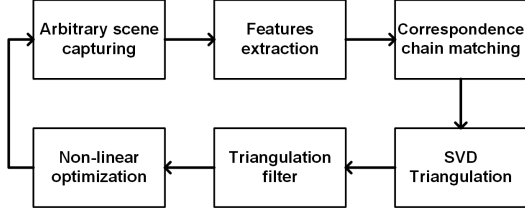
Figure 3: A pipeline of auto-refinement algorithm

4 CALIBRATION AUTO-REFINEMENT

An auto-refinement algorithm uses the previously found reference calibration (Section 3) and tries to estimate, how the calibration parameters need to be compensated for the current configuration of the multi-view camera. It is needed in cases when the placement of views was changed during the camera exploitation, *e.g.* camera is mounted on the car and it is a subject of shakes and other mechanical influences. A pipeline of this method is presented in Fig. 3.

4.1 Correspondence Chain Matching

The important criteria for the selection of correspondences detection algorithm were the robustness of features in real-world images and the running time of the algorithm. Among existing methods, the combination of "Good features to track" method for features extraction [Shi94a] with Kanade-Lucas-Tomasi (KLT) feature tracker [Tom91a] were chosen.

Exploiting the multi-view nature of the light field, in particular the fact that the projection of 3D points from the scene can be seen in all views, the determined features are tracked between different views in a chain manner. Correspondences from the reference view are verified in the neighboring view on the same axis, tracked ones are searched in the next view, and on the last view in one row features are matched in the upper one. This principle is demonstrated in Fig. 4.

Such a method is needed for reducing the number of possibly mismatched correspondences, which may occur especially in the small-baseline multi-view systems, like the camera used in Section 5. This method also helps for the verification of the correspondence inaccuracies, since the very strong matches have to be preserved in all views.

Additional filtration, based on the estimation of fundamental matrix between correspondences in adjacent views, and exclusion of non-matched features is applied. Having a fundamental matrix F, estimated by *e.g.* Random Sample Consensus method [Fis81a], the points from neighboring view x_1 and x_2 are checked by the value of $x_2^T F x_1$ being lower than a certain threshold.

There's a small number of features, which were extracted more than once from the image. To remove the possible influence of such correspondences, we preliminary check the result of the feature detection algorithm
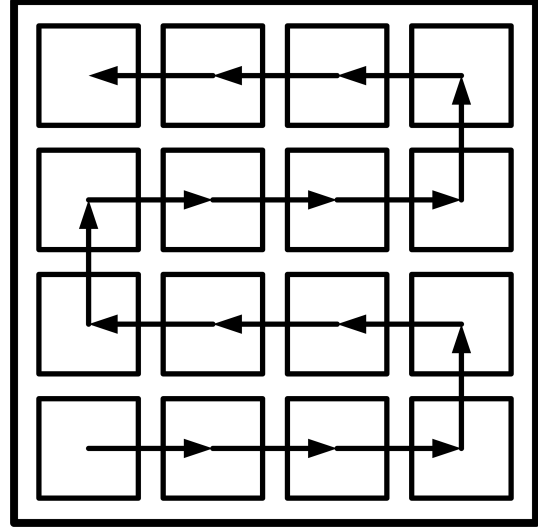

Figure 4: Visualization of views traversing in chain manner

by searching of the nearby correspondences using the Euclidean distance between points.

By empirically setting a certain threshold for these distances we can efficiently filter out the closely placed features. For our experiments it was set to $\sqrt{2}$.

4.2 Triangulation Filter

Previously described methods of filtration can eliminate a big amount of wrongly estimated correspondences. However, some false matches, especially the ones placed close or on the textureless areas, can survive these checks. To eliminate such mismatches we propose a filter, based on the re-projection of triangulated points.

The filtered points are triangulated based on original intrinsic values K_i and rotation and translation vectors $R_{Oi}, t_{Oi}, i = 1..N$. A projection matrix P_i^O is composed as $K_i[R_i^O|t_i^O]$. For every correspondence $m_i = [x_i, y_i]$, matched in every view out of N, by taking a vector P_{oi}^{rT} for every row of the corresponding projection matrix the matrix A is composed as follows [Har03a]:

$$
A = \begin{bmatrix} x_1 P_{O1}^{3T} - P_{O1}^{1T} \\ y_1 P_{O1}^{3T} - P_{O1}^{2T} \\ \vdots \\ x_N P_{ON}^{3T} - P_{ON}^{1T} \\ y_N P_{ON}^{3T} - P_{ON}^{2T} \end{bmatrix} \tag{11}
$$

Singular Value Decomposition is applied to this matrix, the triangulated points are extracted from the smallest singular value of A. These points are used afterward for the optimization result in Section 4.3.
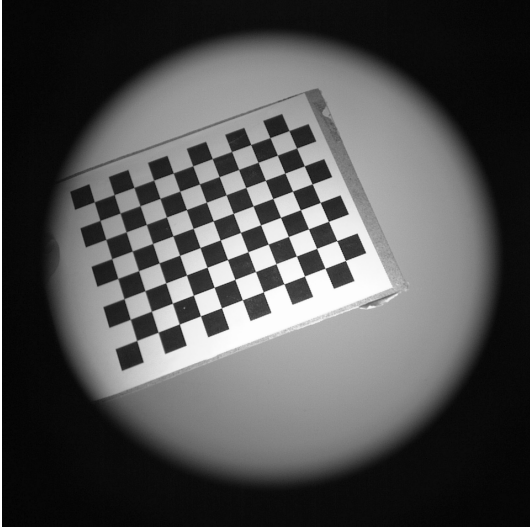
Figure 5: An example of checkerboard pattern, used for light field camera calibration

Using the P_O we project the triangulated points M_t to 2D space and estimate the Euclidean distance between original and projected points:

$$m_t = PM_t$$
$$dist_t = \|m_t - m_r\|, \tag{12}$$

where m_r stands for an original correspondence from reference view.

We find a median of all distances between each correspondence and its projection, and filter out all matches, for which the distance is bigger than the median value.

4.3 Bundle Adjustment

The bundle adjustment problem [Tri99a] is solved for estimating the compensation intrinsic and extrinsic values. For every initially found point x_{ik}, $k = 1..M$, where M is a total number of points, in all views $i = 1..N$ we try to minimize reprojection error as:

$$\sum_{i=1}^{N}\sum_{k=1}^{M} \|x_{ik} - Q(X_k, K_i, d_i, R_i, t_i)\| \tag{13}$$

, where $Q()$ projects the 3D points to 2D plane, involving Eq. $1-3$.

5 EXPERIMENTS

5.1 Calibration algorithm

For the tests of pattern-based calibration algorithm we used a light field camera from [Ani19a]. It composed of 4x4 lenses, providing a 960x960 pixels RGB image per view, which afterward converted to grayscale for the sake of calculations simplicity. A 12x9 checkerboard pattern with 20x20 mm squares was used for the calibration. An example of the calibration pattern is
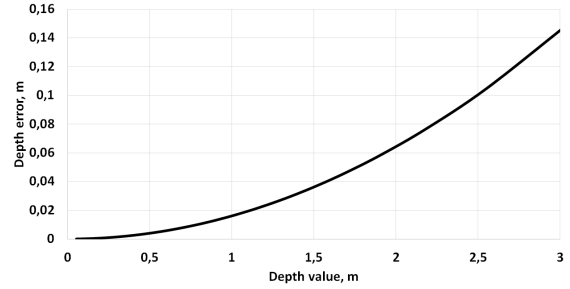

Figure 6: Depth error for $E_{PnP} = 0.246$ pixels

presented on Fig. 5. We captured 40 scenes with this pattern for testing purposes.

To control the accuracy of the estimated intrinsic matrix K and distortion coefficients d, the reprojection error is computed as in Eq. 13 for every light field view. Similar computations are done during the estimation of the extrinsic values.

For the test dataset the average reprojection errors for monocular calibration and PnP problem were $E_{mono} = 0.159$ and $E_{PnP} = 0.246$ pixels respectively.

A comparison of our method was done with the method of Xu *et al.* [Xu15c]. For the same test dataset we have obtained reprojection errors of $E_{mono} = 0.153$ and $E_{PnP} = 0.156$ pixels.

We can state that optimization of all parameters together for the precisely estimated points make sense in terms of accuracy. However, the potential drawback is related to higher running time of the joined optimization.

To verify influence of the reprojection error to the actual depth reconstruction we found a depth error with common focal length values $f = f_x = f_y = 850$ pix and baseline between the two most distant views on the same axis $b = 0.018$ m as:

$$\Delta Z = \frac{fb}{d - E_{pnp}} + \frac{fb}{d + E_{pnp}}, \tag{14}$$

, where d is the disparity value, *i.e.* displacement between pixels, which can be further converted to actual depth value. This error is visualized on Fig. 6.

On a target range of the test camera, which is 0.5-2.0 m [Ani19a], the estimated reprojection introduces an inaccuracy in the amount of 0.004-0.065 m, which is lower then the depth accuracy on this specific distance range.

5.2 Auto-refinement algorithm

A dataset of synthetic light fields was used for verification of auto-refinement stability. The rectified and undistorted images with known intrinsic parameters were generated using Blender and the
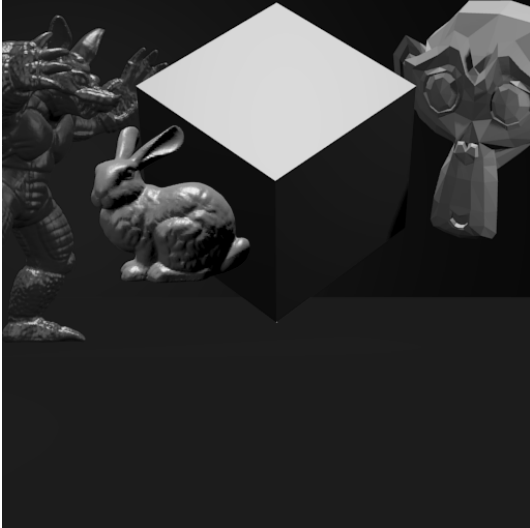
Figure 7: An example of synthetic scene for verifying the auto-refinement algorithm

Figure 8: Reprojection error of auto-refinement algorithm dependently of the applied noise for synthetic images

script from 4-dimensional Light Field Benchmark [4DLFB] [Hon16a]. 5x5 light fields of size 512x512 pixels per view with a baseline of 100 mm between adjacent views were generated from a simple scene with different overlapping objects, as demonstrated in Fig. 7. A sequence of ten images was used for the tests.

In average, 1/10 of originally detected points were filtered by correspondence chain matching, out of which half of the points survived the triangulation-based filtration. Processing of one frame takes around one second.

The difference between original and refined intrinsic and extrinsic parameters was measured and considered as non-informative, since no significant difference between original and refined values was found. It can be explained by the quality of correspondences, which is in general good for the synthetic data.

To check the accuracy of the auto-refinement algorithm, we applied rotation and translation noise to the captured frames and measured the average reprojection errors. Results are demonstrated on Fig. 8.

In total both types of noise create acceptable level of reprojection error. In a similar to pattern-based calibration manner we have evaluated the depth error for maximum reprojection error from the rotation noise, result of which is presented on Fig. 9.

For the noise simulation of rotation was applied only to Z-axis. Rotations on X and Y axes are the subject of tangential distortion. It occurs when the lens array is not parallel to the camera sensor plane. This type of distortion is assumed to be zero in the applied camera model.
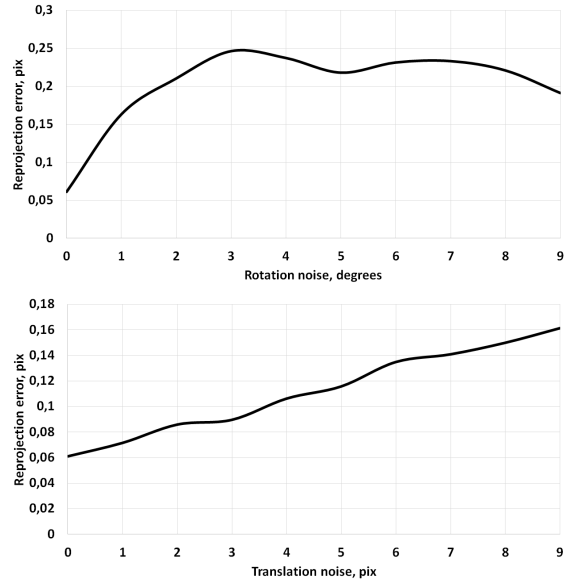
5.3 Discussion

The pattern-based calibration method used gives adequate results in terms of the resulting reprojection error. Empirically, we have found that the overall calibration quality depends largely on the quality of the calibration target, especially its flatness.

For the proper calibration we have come to the number of 25-30 pattern images in one sequence. All areas on the light field views should be covered with pattern images in various positions to ensure correct estimation of internal values.

One of the assumptions of the algorithms was the similarity of the lens parameters for each view. For cases with a significant shift (in terms of rotation or translation) of at least one of the views over the others, averaging over extrinsic values cannot be used; it should be replaced by nonlinear methods.

Experiments with the auto-refinement algorithm on synthetic images show that the reprojection error increases in proportion to the level of lens shift, while changes in their rotation affect only up to some point with a plateau thereafter.

During the auto-refinement experiments, we noticed that optimizing all the parameters together leads to incorrect results. This was the motivation for dividing the optimization procedure into three steps applied to the same model. First, only 3D points are optimized, while intrinsic and extrinsic camera values are fixed. This condition is relaxed in the second part, where the intrinsic values of all views are optimized. Finally, we optimize all camera parameters together. All of the above steps are repeated with new captured scenes. It can be
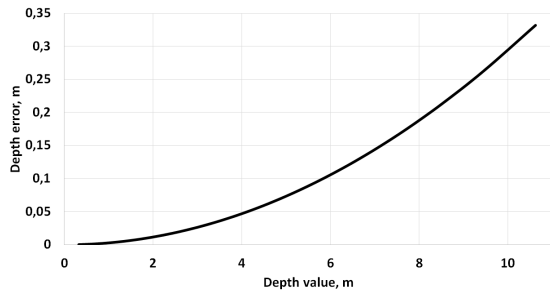
Figure 9: Auto-refinement depth error

stopped either by using a certain number of iterations or by reaching the desired reprojection error below a threshold value.

Several limitations of the auto-refinement algorithm were identified during the tests. It does not work well with repeating textures and with small distances between detected matches, *e.g.* on keyboards. This problem can be solved either by applying additional filtering measures or by changing the match detection algorithm. In addition, the coverage of the image area by the correspondence is important to obtain correct lens geometry, a similar requirement stands for pattern-based calibration. Additional correspondence distribution checks can be made to discard images without proper coverage. Because of this, the distortion coefficients cannot be correctly corrected by our auto-rectification method and remain locked during the optimization stage.

We have tested the auto-refinement algorithm with and without triangulation filter. Without filtering the results were totally incorrect, so they are not included in part of the experiments.

5.4 Implementation

The algorithm was implemented in C++, uses only a central processing unit (CPU), and requires additional libraries such as OpenCV [OpenCV] for image-related routines and Ceres Solver [Ceres] for solving nonlinear optimization problems. The algorithm was tested in Windows 7 with Intel Xeon CPU E3-1245 V2, additional tests were done on the NVIDIA Jetson TX2 and AGX Xavier platforms. Memory requirements directly depend on the size of the provided light field.

6 CONCLUSION

In this paper we presented the calibration and rectification pipeline for light field cameras together with the method for calibration parameters refinement from the arbitrary scenes. Additional filtration measures like triangulation filter and two-dimensional technics were outlined. The algorithms were evaluated using synthetic and real-world data, the results of the experiments support the idea of algorithms' utilization for real-world light field calibration and its refinement. The presented calibration and auto-refinement principles can be further extended to other multi-view systems with preserved similarity of views alignment. To further our research we plan to overcome the mentioned limitations and conduct additional experiments on other similar multi-view systems. For example, method from [Ddv08a] can be adapted to deal with distortions from arbitrary scene.

7 ACKNOWLEDGMENTS

8 REFERENCES

[Ade92a] E. H. Adelson, and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 99–106, IEEE, 1992.

[Ani19a] Y. Anisimov, O. Wasenmüller, and D. Stricker. A compact light field camera for real-time depth estimation *In International Conference on Computer Analysis of Images and Patterns*, pp. 52–63. Springer, 2019.

[Ber15a] F. Bergamasco, A. Albarelli, L. Cosmo, A. Torsello, E. Rodola, and D. Cremers, Adopting an unconstrained ray model in light-field cameras for 3D shape reconstruction. *2015 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3003–3012, IEEE, 2015.

[Bok16a] Y. Bok, H.-G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *2016 IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 39(2), pp. 287–300, 2016.

[Ddv08a] F. Ddvernay, and O. Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications*, vol. 5, pp. 14–24 Springer Verlag, 2013.

[Fis81a] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, vol. 24(6), pp. 381–395, 1981.

[Foe87a] W. Förstner, and E. C. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pp. 281–305, 1987.

[Har03a] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. *Cambridge University Press*, ISBN 0521540518, 2003.

[Hon16a] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. *2016 Asian Conference on Computer Vision*, pp. 19–34, Springer, 2016.

[Ji19c] Y. Ji, and J. Wu. Calibration method of light-field camera for photogrammetry application. *Measurement*, vol. 148, pp. 106943, Elsevier, 2019.

[Jin16a] J. Jin, Y. Cao, W. Cai, W. Zheng, and P. Zhou. An effective rectification method for lenselet-based plenoptic cameras. *Optoelectronic Imaging and Multimedia Technology IV*, vol. 10020, SPIE, 2016.

[Kan08a] Y.-S. Kang, C. Lee, and Y.-S. Ho. An efficient rectification algorithm for multi-view images in parallel camera array. *2008 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, pp. 61–64. IEEE, 2008.

[Mar63] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, vol. 11(2), pp. 431–441, 1963.

[Ng05a] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Doctoral dissertation*, Stanford University, 2005.

[Nou17a] C.-A. Noury, C. Teulière, and M. Dhome. Light-field camera calibration from raw images. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2017.

[Obr18a] S. O'brien, J. Trumpf, V. Ila, and R. Mahony. Calibrating light-field cameras using plenoptic disc features. *2018 International Conference on 3D Vision (3DV)*, pp. 286–294, IEEE, 2018.

[Rod40a] O. Rodrigues. On the geometrical laws that govern the displacements of a solid system in space, and on the change of coordinates resulting from these displacements considered independently of the causes that can produce them. *J Math Pures Appl*, vol. 5, pp. 380–440, 1840.

[Shi94a] J. Shi, and C. Tomasi. Good features to track. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, IEEE, 1994.

[Sun19a] X. Sun, X. Jin, P. Wang, Y. Chen, and Q. Dai. Blind calibration for focused plenoptic cameras. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 115–120. IEEE, 2019.

[Suz85a] S. Suzuki. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, vol. 30(1), pp. 32–46. Elsevier, 1985.

[Tom91a] C. Tomasi and T. Kanade. Detection and tracking of point features. *Technical Report CMU-CS-91-132, Carnegie, Mellon University*, 1991.

[Tri99a] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment-a modern synthesis. In *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[Vai04a] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane+parallax for calibrating dense camera arrays. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2004.

[Wil05a] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM SIGGRAPH 2005 Papers*, pp. 765-776, ACM, 2005.

[Xu15c] Y. Xu, K. Maeno, H. Nagahara, and R.I. Taniguchi. Camera array calibration for light field acquisition. *Frontiers of Computer Science*, vol. 9(5) pp. 691–702, Springer, 2015.

[Zha00a] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22(11) pp. 1330–1334, IEEE, 2000.

[Zho19a] P. Zhou, W. Cai, Y. Yu, Y. Zhang, and G. Zhou. A two-step calibration method of lenslet-based light field cameras. *Optics and Lasers in Engineering*, vol. 115, pp. 190–196, 2019.

[Zho19b] P. Zhou, Z. Yang, W. Cai, Y. Yu, and G. Zhou. Light field calibration and 3d shape measurement based on epipolar-space. *Optics express*, vol. 27(7), pp. 10171–10184, 2019.

[4DLFB] 4d Light Field Benchmark. `http://hci-lightfield.iwr.uni-heidelberg.de`. Accessed: 06.05.2021.

[Ceres] Google Ceres. `http://ceres-solver.org/`. Accessed: 06.05.2021.

[OpenCV] OpenCV. `https://opencv.org/`. Accessed: 06.05.2021.