A Multiple-Camera System Calibration Toolbox Using A Feature Descriptor-Based Calibration Pattern

Bo Li, Lionel Heng, Kevin Köser and Marc Pollefeys
Computer Vision and Geometry Group, ETH Zürich

Abstract— This paper presents a novel feature descriptor-based calibration pattern and a Matlab toolbox which uses the specially designed pattern to easily calibrate both the intrinsics and extrinsics of a multiple-camera system. In contrast to existing calibration patterns, in particular, the ubiquitous chessboard, the proposed pattern contains many more features of varying scales; such features can be easily and automatically detected. The proposed toolbox supports the calibration of a camera system which can comprise either normal pinhole cameras or catadioptric cameras. The calibration only requires that neighboring cameras observe parts of the calibration pattern at the same time; the observed parts may not overlap at all. No overlapping fields of view are assumed for the camera system. We show that the toolbox can easily be used to automatically calibrate camera systems.

I. INTRODUCTION

Multiple-camera systems have become increasingly prevalent in robotics and computer vision research. These systems include stereo cameras, Point Grey's Ladybug multi-camera systems, and a large variety of customized camera systems. Multiple-camera systems usually comprise normal pinhole cameras, fish-eye cameras and catadioptric cameras. To make such systems usable, both the intrinsics and extrinsics of the cameras have to be calibrated accurately.

A. Related work

Recently, many efficient methods have been developed for intrinsic calibration of many types of cameras. These methods can be divided into two categories: calibration with a special calibration object, and self-calibration. In this paper, we focus on the former category which is usually much more accurate than self-calibration. Many toolboxes are available for this category of methods. Seminal work on calibrating a pinhole camera can be found in [1]. Some popular calibration toolboxes [2], [3] are inspired by this method. For generic cameras, [4] proposes a toolbox to use a polynomial to approximate rays corresponding to each image point. This method generically applies to most camera models but does not provide a closed-form solution for undistorting raw images. In the toolbox proposed in [5], an unified projection model is proposed for calibrating a catadioptric system, fish-eye camera and camera-spherical mirror system. This model is similar to [4] but parameterizes rays instead of using an arbitrary polynomial, which makes undistortion much simpler.

Some toolboxes are also available to calibrate simple multiple-camera systems. [2] enables one to calibrate a stereo camera. For calibration with a system of more than
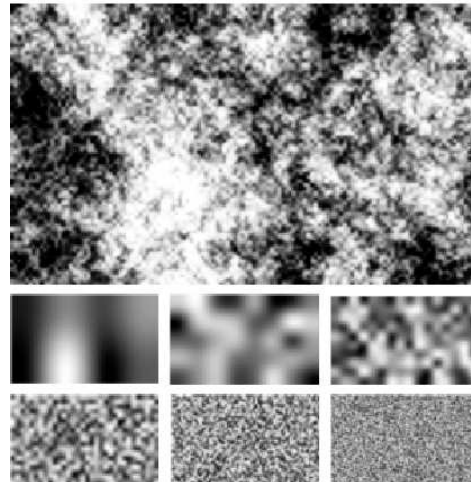

Fig. 1. Top: The proposed calibration pattern. Bottom: Image components with noise at different frequencies.

two cameras, these toolboxes [6], [7] can be used. These calibration toolboxes make use of the overlapping fields of view of the cameras; hence, these toolboxes can calibrate both a stereo camera and a circular camera rig with all cameras pointing inwards. However, these toolboxes are not suitable for calibrating a system of cameras with either no or minimal overlapping fields of view. Camera rigs with cameras pointing outwards are increasingly popular in both academia and industry; it is not easy to calibrate this system using existing calibration toolboxes due to the minimal overlapping fields of view. Hand-eye calibration algorithms [8], [9] can be used to calibrate this system but requires reconstructing visual odometry for each camera, and the calibration is often not accurate due to visual odometry drift.

In addition to camera models, research has also focused on development of easy-to-use calibration patterns. Early research made use of cubes with either a chessboard pattern or circular dots on their surfaces [2]. This pattern is not convenient to use as a perfect cube is not trivial to build. Current state-of-art calibration systems mainly make use of calibration boards which are often planes with a chessboard or circular dots printed on them. An automatic detector for such patterns is readily available [10]. A comparison of calibration accuracy between a chessboard and circular dots can be found in [11]. [7] uses a similar but even more simplified calibration object, a single point light, for calibrating multiple-camera systems. One disadvantage of

these patterns is that the entire pattern has to be visible in each calibration image; this requirement excludes cameras with minimal overlapping fields of view. In addition to the chessboard and circular dots, other patterns have been proposed. [12] uses a temporal coded pattern to calibrate cameras. This method uses the Gray code to match world points to image points, thus not requiring the entire pattern to be in the image. The drawback of this method is the limited flexibility; the calibration requires both a display projector and a tripod for mounting a camera.

In summary, the chessboard and circular-dot patterns are widely recognized as the state-of-art for calibrating single and stereo cameras, for their ease-of-use and high calibration accuracy. However, for a system with multiple cameras pointing in different directions, it is difficult to use these toolboxes to calibrate the extrinsics of the camera system. This is because current automatic and semi-automatic chessboard detectors require the chessboard to be entirely within the field of view of the cameras. Therefore, if two cameras have minimal overlapping fields of view, it is difficult to use the chessboard for extrinsic calibration in contrast to stereo camera calibration. To the best of our knowledge, there are no published calibration toolboxes for easily calibrating multiple-camera systems with minimal overlapping fields of view.

B. Contributions

In this paper, we solve the calibration problem for a multiple-camera system without assuming overlapping fields of view. The basic motivation is to relax the requirement that the calibration pattern be seen in its entirety by each camera; ideally, the calibration board is automatically detected even when the cameras observe different parts of the board. For most multiple-camera systems, it is fairly common for a camera pair to see different parts of a suitably-sized calibration board at the same time. Based on this motivation, we design a feature descriptor-based calibration pattern which is easy to detect even when seen partially by a camera, and an extrinsic calibration framework using this pattern.

This paper makes two novel contributions:

1) A new calibration pattern that encodes feature points using feature extraction techniques. Our pattern contains many detectable features on multiple scales as illustrated in section II. The pattern can be recognized and localized even if the pattern is partially seen in an image.
2) A toolbox based on the proposed calibration pattern. This toolbox can be used for both intrinsic and extrinsic calibration of a multiple-camera system, as illustrated in sections III and IV. Similarly to existing calibration toolboxes, our toolbox can also be used for intrinsic calibration of a single camera.

We note that our approach is similar to standard structure-from-motion approaches; the main difference is that we use points with known 3D coordinates.

II. CALIBRATION PATTERN

A. Feature Detection Revisited

Point-feature detection is a computer vision technique widely and successfully applied to many areas such as sparse reconstruction and object detection. A point feature typically contains two components: a keypoint and a descriptor. We look at the widely-used SIFT implementation [13] as a example. A Difference of Gaussian filter (DoG) is used to detect keypoints. This detection is executed on both the original image and downsampled images; in short, keypoint detection is done on different scales. For each keypoint, the image gradient in the keypoint's neighborhood is converted to a histogram which is then used as the descriptor for the corresponding feature point. SURF, a variant of SIFT, is also a widely-used technique for feature detection and descriptor extraction [14]. SURF replaces SIFT in many applications due to the its computational efficiency. In the proposed toolbox, we use SURF features.

B. Reverse Engineering

The basic idea behind the proposed calibration pattern is to find and design a pattern that yields a high number of detectable features. At the same time, the feature descriptors should be highly discriminative so that we can easily obtain unique feature point matches.

To facilitate feature detection, we use several noise images to compose a calibration pattern in accordance with the mechanism of SIFT/SURF. The DoG filter applied to a noise image can yield points with high response. However, a problem with high-frequency image noise is the blurring effect. For a grayscale image, if the camera is located far away, the noise image is perceived as a purely gray image. The solution to this problem is to compose images with noise from multiple scales. In our implementation, we generate noise images of different sizes, and resize them such that they have the same size. These images with noise on different scales are then added up together. This procedure can be interpreted as a reverse engineering of the scaling procedure in SIFT/SURF detection. Thus, the resulting image contains a high number of detectable features on different scales; such features can be detected by a camera at varying distances. The Matlab code in figure 2 generates a 600×800 calibration pattern. Figure 1 shows a calibration pattern at the top, and its components with noise on different scales at the bottom.

C. Feature Matching

Feature detection and feature matching between two images are two standard steps in a modern 3D vision pipeline. In our calibration approach, we employ a similar step to match features between each image and the known calibration pattern image. First, features detected from each image and the pattern image are matched according to the descriptor similarity. We use the well-known distance ratio check proposed in [13]. For a set of at least 2 candidate matches for each query feature, the best match is accepted only if its descriptor distance is smaller than 0.6 times the

```
N = 600; M = 800;
pattern = 0; count = 0;
m = 4;
while m < M
  n = round(N / M * m);
  noise = rand(n, m);
  noise = imresize(noise, [N, M]);
  pattern = pattern + noise;
  count = count + 1;
  m = m * 2;
end
pattern = pattern ./ count;
pattern = histeq(pattern);
imshow(pattern);
```

Fig. 2.   Matlab code for pattern generation.

descriptor distance between the query feature and its second best match.

Next, a *fundamental matrix for radial distortion* is estimated by RANSAC between the image points and pattern points to find inlier point correspondences. Note that this fundamental matrix is not the same matrix used in traditional epipolar geometry. The traditional matrix requires that point correspondences have image coordinates corresponding to rectified images. Details about this fundamental matrix can be found in [15]. We provide some simple explanation about the fundamental matrix here. Denote p^d as a distorted image point, p^u as its corresponding undistorted point and p^c as the corresponding point on the calibration pattern. If the image only has radial distortion and e is the distortion center, we can write:

$$p^d = e + \lambda(p^u - e) \qquad (1)$$

where λ is a distortion coefficient corresponding to p^u. In addition, since p^u can be obtained by a perspective transform from p^c, there exists a homography H such that $p^u = Hp^c$. Substituting this into the above equation, we obtain:

$$p^d = e + \lambda(Hp^c - e) \qquad (2)$$

Left-multiplying the equation by $[e]_\times$, we get:

$$[e]_\times p^d = \lambda[e]_\times Hp^c \qquad (3)$$

Left-multiplying the equation again by $p^{d\top}$, we get:

$$0 = p^{d\top}[e]_\times Hp^c \qquad (4)$$

$F \equiv [e]_\times H$ is the fundamental matrix for radial distortion, and e can be interpreted as the principal point. This fundamental matrix is well-suited to both the pinhole and unified projection models, and is used to remove false feature matches in our proposed calibration approach. In the case that the images perfectly agree with the pinhole camera model (zero lens distortion), it suffices to estimate a homography. This case could be detected by automatic model selection methods like the GRIC criterion[16]. However, since this is a very special case, for the current version, we leave it to the user to specify this mode.

III. SINGLE-CAMERA CALIBRATION

A single-camera calibration estimates the camera intrinsics and the poses of the calibration pattern with respect to the camera's coordinate system. This data from the single-camera calibration is used by the multiple-camera extrinsic calibration.

A. Camera Model

Consider a point $(u, v, \gamma)^\top$ on an image plane with focal length equal to γ and principal point $(0, 0)^\top$. It corresponds to a ray with direction $(u, v, f(u, v))^\top$ which goes through the origin of the camera's coordinate system. This ray definition unifies the various projection models via different definitions of $f(u, v)$. For the pinhole projection model,

$$f(u, v) \equiv \gamma \qquad (5)$$

For the unified projection model with its lens distortion parameter $\xi = 1$,

$$f(u, v) = \frac{\gamma}{2} - \frac{1}{2\gamma}\rho^2 \qquad (6)$$

where $\rho = \sqrt{u^2 + v^2}$. Details about equation 6 can be found in [5]. For the Taylor projection model, $f(u, v)$ is parameterized as a general polynomial with one variable ρ, as proposed in [4]:

$$f(u, v) = a_0 + \cdots + a_n\rho^n \qquad (7)$$

Equation 7 does not have a closed-form inverse transform. In the proposed toolbox, we use the unified projection model in [5] to model a varying range of cameras which include but are not limited to normal, wide-angle, fish-eye and catadioptric types. This model uses the same intrinsic parameters that the pinhole projection model uses: the focal length (γ_1, γ_2), aspect ratio s, principal point (u_0, v_0), and radial and tangent lens distortion parameters (k_1, k_2, k_3, k_4). The model uses one additional parameter ξ to model the omnidirectional camera effect.

B. Initialization

The unified projection model of [5] assumes the catadioptric coefficient $\xi = 1$ for initialization, and then, refines the estimated intrinsics and extrinsics. The initialization is very simple and can generate a good initial guess for all parameters. The limitation of the initialization is that it requires a known projection of a straight line on the pattern. Such a projection is easy to obtain for a chessboard but difficult for our proposed pattern. Fortunately, the initialization can be solved by substituting the unified projection model into the initialization of the Taylor projection model of [4].

In the initialization, we assume the two focal lengths (γ_1, γ_2) to be the same, the principal point to have the same coordinates as the image center, and zero lens distortion. For each detected feature point, we compute the corresponding u and v by subtracting the image center coordinates from the feature point's coordinates.

For a feature point with homogeneous image coordinates $p^c = [x, y, 1]^\top$, denote its corresponding 3D point as

$[x, y, 0, 1]^\top$. The projection equation relating this 3D point to its corresponding camera ray $[u, v, f(u,v)]^\top$ is:

$$\mu \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} p^c \tag{9}$$

Left-multiplying the equation by $[u, v, f(u,v)]^\top_\times$, we have:

$$0 = [u, v, f(u,v)]^\top_\times \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} p^c \tag{10}$$

Denote the rotation $r_i = [r_{1i}, r_{2i}, r_{3i}]^\top$ and the translation $t = [t_1, t_2, t_3]^\top$. The third row of equation 10 is independent of $f(u,v)$ and is a linear equation with respect to the unknowns $r_{11}, r_{21}, r_{12}, r_{22}, t_1, t_2$:

$$u(r_{21}x + r_{22}y + t_2) - v(r_{11}x + r_{12}y + t_1) = 0 \tag{11}$$

For each image pair, we can substitute the feature point correspondences into this equation and stack them as a linear system. Solving this system, we obtain $r_{11}, r_{21}, r_{12}, r_{22}, t_1, t_2$ up to a unknown scale. r_{31}, r_{32} and the scale can be determined by exploiting the unit and perpendicularity constraints of r_1 and r_2. Note that multiple solutions exist at this stage; we reject the incorrect solutions after estimating the remaining unknowns γ and t_3.

Solving γ and t_3 requires using the first and second row in equation 10:

$$v(r_{31}x + r_{32}y + t_3) - f(u,v)(r_{21}x + r_{22}y + t_2) = 0 \tag{12}$$
$$f(u,v)(r_{11}x + r_{12}y + t_1) - u(r_{31}x + r_{32}y + t_3) = 0 \tag{13}$$

For the pinhole projection model, $f(u,v)$ is replaced by γ. Substituting $r_{11}, r_{21}, r_{12}, r_{22}, t_1, t_2$, we obtain linear equations with respect to γ and t_3. The two unknowns are then solved by forming a linear system using detected multiple correspondences.

For the unified projection model, equation 6 is substituted into $f(u,v)$. We regard γ, $\frac{1}{\gamma}$ and t_3 as three unknowns and solve them by forming a linear system similar to the above. Since we ignore a constraint by treating γ and $\frac{1}{\gamma}$ as two unknowns, the estimates are less accurate. This is not an issue for the initialization since the estimate is further refined.

The initialization returns multiple solutions for the intrinsics and extrinsics; the correct solution can be selected by checking the reprojection error.

The toolbox initializes γ and the extrinsics for each input image, and selects the median of all values for γ as the initial estimate for γ.

C. Refinement

Based on the initial estimate, the toolbox then refines all intrinsic and extrinsic parameters using the Levenberg-Marquardt algorithm to minimize the sum of all reprojection errors. Further details about the optimization can be found in [4] and [5].
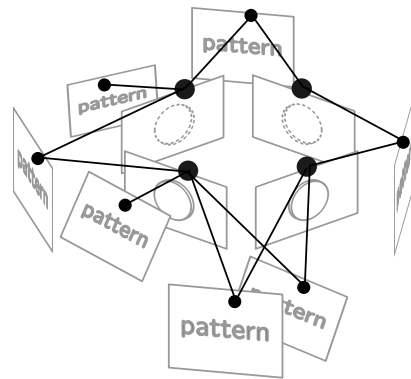

Fig. 3. A example of a pose graph for the calibration scenario of four cameras. Big dots denote camera vertices and small dots denote feature point vertices.

IV. EXTRINSIC CALIBRATION

In this section, we assume that the cameras are rigidly mounted to a rigid body. During the image capture process, we move the calibration pattern around the camera system. The single-camera calibration provides estimates of the poses of the calibration pattern. If the cameras are synchronized such that they capture images of the pattern at the same time, the relative poses of the pattern with respect to each camera are known. Thus, the initial camera poses can be extracted from these relative poses. The toolbox optimizes the initial camera poses using a bundle-adjustment-like method.

A. Initialization

We create a pose graph to denote the calibration scenario. Each camera is denoted by a camera vertex cam_i in the graph. Meanwhile, each pose of the pattern is also denoted by a pattern vertex pat_i. If cam_i takes a photo of the pattern at pose pat_j, then cam_i and pat_j are connected by an image edge denoted by $img_{i,j}$. Each edge uniquely maps to one image. Note that this graph is a bipartite graph and each edge links one camera vertex and one pattern vertex. Figure 3 provides a simple illustration of such a graph.

Vertices in the pose graph can be used to store the poses of the cameras and the pattern in a global coordinate system. Edges can be used to store the relative pose transform between the camera pose vertex and pattern pose vertex. For each image of the pattern at pose pat_j taken by cam_i, we have the relative pose of pat_j with respect to cam_i computed from the single-camera calibration.

Assuming that the global coordinate system is aligned with cam_1, the poses of all vertices connected to cam_1 can be obtained by following the image edges from cam_1. In practice, if two cameras see the pattern at the same time in their images, then the two cameras are connected via two image edges to one pattern vertex.

Our toolbox implementation first builds a pose graph based on the results of single-camera calibration performed for all cameras. Next, a spanning tree with cam_1 as its root is extracted using breadth-first search. Vertex poses are then computed by traversing the spanning tree from cam_1 and

following the image edges. In the end, we have initial pose estimates for all vertices in the global coordinate system.

B. Refinement

Denote the pose in the global frame of vertex i (either a camera vertex or a pattern vertex) in the graph as H_i. On an edge $img_{i,j}$ connecting cam_i and pat_j, the relative transform from pat_j to cam_i can be denoted as $H_{i,j} = H_i^{-1}H_j$. For a calibration pattern point p^c, its reprojection error in each image that the point is seen in is:

$$e_{\text{reproj}}(p^c, H_{i,j}, \mathcal{I}_i) = \|\pi(p^c, H_{i,j}, \mathcal{C}_i) - p^d\|^2 \tag{14}$$

where π is the image projection function corresponding to either the pinhole or unified projection model and $\mathcal{C}_i$ denotes the intrinsics of cam_i. p^d is the distorted image point corresponding with p^c.

The initial calibration estimate is refined to minimize the sum of all reprojection errors. The refinement can be over either all vertex poses or over both vertex poses and intrinsics. The optimization problem with only vertex poses is defined as:

$$\underset{H_i, i>1}{\arg\min} \sum_{img_{i,j}} \sum_{k} e_{\text{reproj}}(p_k^c, H_i^{-1}H_j, \mathcal{C}_i) \tag{15}$$

while the optimization problem with both vertex poses and intrinsics is:

$$\underset{H_i, \mathcal{C}_1, \mathcal{C}_i, i>1}{\arg\min} \sum_{img_{i,j}} \sum_{k} e_{\text{reproj}}(p_k^c, H_i^{-1}H_j, \mathcal{C}_i) \tag{16}$$

The optimization is over H_i with $i > 1$, since $H_1 \equiv I_{4 \times 4}$ is the reference frame. The toolbox executes the optimization using the Levenberg-Marquardt algorithm.

V. Experiments

We carry out two experiments with our proposed calibration pattern and toolbox. In the first experiment, we use a stereo camera and compare the calibration results from our toolbox and those from the OpenCV-based chessboard calibration. In the second experiment, we use our toolbox to calibrate a four-camera system. The latter camera system is a challenging case, especially for existing multiple-camera calibration methods.

A. Calibration of a stereo camera

For the case of a stereo camera, a chessboard is typically used for extrinsic calibration. The two cameras tend to have a large overlapping field of view, and the entire chessboard has to be in both cameras' fields of view. We test our toolbox with a custom-built stereo camera comprising two mvBlue-FOX cameras with hardware synchronization. Figure 4 shows a sample stereo image pair used for each calibration. Note that in contrast to the chessboard calibration, our proposed calibration pattern does not have to be entirely within the field of view. The calibration results are shown in tableI. The descriptor-based pattern provides many more features with significantly fewer images compared to the chessboard pattern. We observe that the two calibration results are very similar. The reprojection error for our proposed pattern is
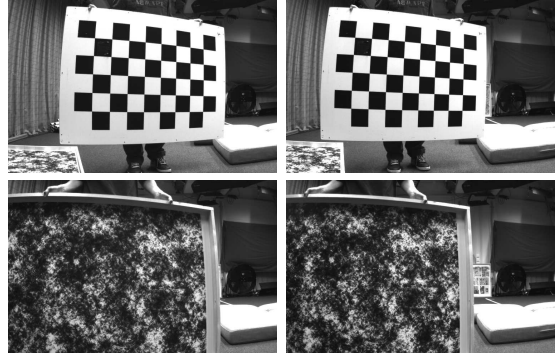

Fig. 4. Sample images used to calibrate the stereo camera. The top row shows a chessboard used by the chessboard calibration while the bottom row shows our calibration pattern used by our calibration toolbox.
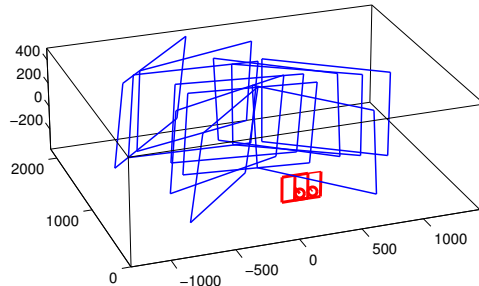

Fig. 5. 3D plot of stereo camera and calibration pattern poses generated by our toolbox.

TABLE I

Comparison of results from our method and OpenCV-based chessboard calibration for a stereo camera.

| Object	Descriptor-based pattern	5×8 chessboard
Image size	480×752	480×752
# images	10×2	30×2
# features (L/R)	3073 / 2942	1200 / 1200
Focal length (L)	$(720, 718)$	$(720, 722)$
Focal length (R)	$(709, 706)$	$(709, 710)$
Principal point (L)	$(383, 249)$	$(392, 250)$
Principal point (R)	$(387, 242)$	$(389, 249)$
Rotation vector	$[0.001, -0.010, 0.011]$	$[0.004, -0.009, 0.011]$
Translation vector (unit)	$[-1.00, -0.012, 0.005]$	$[-1.00, -0.011, 0.012]$
Reprojection error	0.4 px	0.2 px
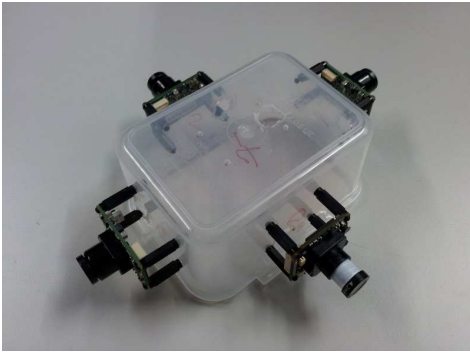
Fig. 6. A four-camera system with an approximate 90° relative rotation between each pair of neighboring cameras.

Fig. 7. Sample images used to calibrate the four-camera rig. Each row corresponds to an image pair from a different pair of neighboring cameras.

higher than that for the chessboard; SURF feature detection is slightly less accurate than sub-pixel chessboard corner detection in terms of feature location. For features on a coarser scale, the higher corresponding error of the estimated feature coordinates may increase the overall reprojection error. We visualize the results of our toolbox calibration in figure 5.

B. Calibration of a four-camera system

In the second experiment, we validate our toolbox on a four-camera system; neighboring cameras have minimal overlapping fields of view. One camera is a mvBlueFOX camera while the rest are Point Grey Firefly MV cameras. This system is shown in figure 6. Existing camera calibration toolboxes are difficult to use when it comes to calibrating such systems. 15 image pairs are taken for each pair of neighboring cameras; two examples are shown in figure 7. Due to the non-overlapping field of view, neighboring cameras only see a small part of the board close to the board's border. Thus, some images do not have sufficient features for matching, and are automatically discarded by the toolbox. In addition, to ensure accurate intrinsic calibration, we take 5 images for each camera with the pattern occupying a large part of each image. Figure 8 plots the 3D poses of both cameras and patterns. The average reprojection error over all images and corresponding to the estimated intrinsics and extrinsics is 0.7 pixels.

VI. CONCLUSIONS

We have proposed a calibration technique using a feature descriptor-based calibration pattern. This technique can be used for calibrating multiple-camera systems. We show our calibration to work successfully with two multiple-camera systems: a normal stereo camera with a large overlapping field of view, and a four-camera system with minimal overlapping fields of view. A toolbox based on our proposed method is available online at https://sites.google.com/site/prclibo/toolbox.

One limitation of our calibration technique is that large calibration patterns are required for certain multiple-camera systems. One example is a two-camera system in which one camera looks forward, and another camera looks to the right,
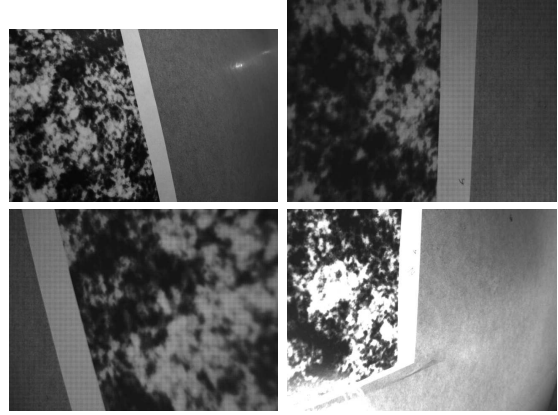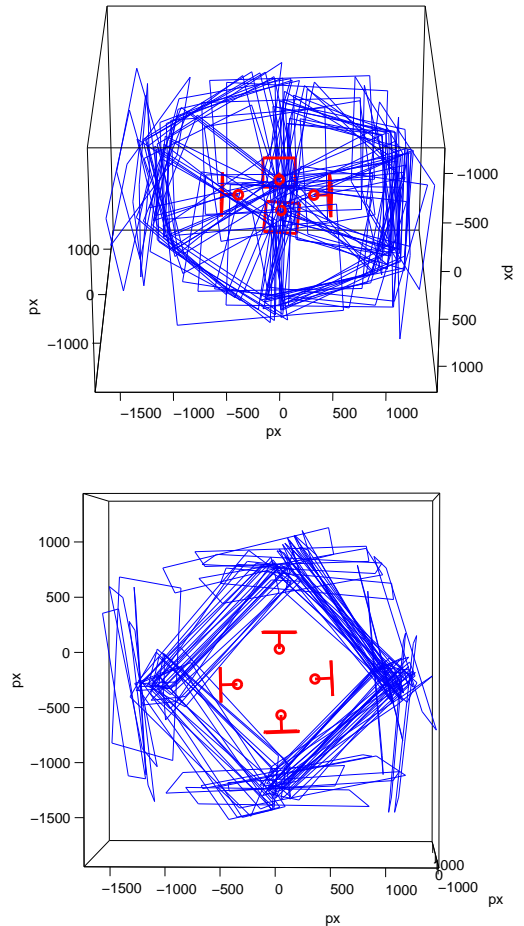

Fig. 8. Two viewpoints of a 3D plot of camera and calibration pattern poses generated by our toolbox for the 4-camera system.

and the baseline between the two cameras is a few meters. In this case, we require the width of the calibration pattern to be at least a few meters. This limitation is inherent in the fact that neighboring cameras must be able to observe some part of the calibration pattern.

VII. ACKNOWLEDGEMENT

REFERENCES

[1] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[3] D. Stoyanov, "Camera calibration tools," *Online Source, Updated Aug*, vol. 24, 2006.

[4] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 5695–5701.

[5] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3945–3950.

[6] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.

[7] J. Barreto, K. Daniilidis, N. Kelshikar, R. Molana, and X. Zabulis, "Easycal camera calibration toolbox."

[8] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 3, pp. 345–358, 1989.

[9] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax= xb," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 16–29, 1989.

[10] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3121–3126.

[11] J. Mallon and P. F. Whelan, "Which pattern? biasing aspects of planar calibration patterns and detection methods," *Pattern recognition letters*, vol. 28, no. 8, pp. 921–930, 2007.

[12] C. Schmalz, F. Forster, and E. Angelopoulou, "Camera calibration: active versus passive targets," *Optical Engineering*, vol. 50, no. 11, pp. 113 601–113 601, 2011.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[15] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1309–1321, 2007.

[16] P. Torr, "An assessment of information criteria for motion model selection," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 47–52.