CALIBRATION OF DENSE MULTI-CAMERA SETUPS FOR SPORTS ANALYTICS AND IMMERSIVE VIEWING EXPERIENCES

Chris Varekamp, Andy Willems

Philips Group Innovation, Research, The Netherlands

ABSTRACT

Dense multi-camera setups consisting of hundreds of low-cost cameras positioned around a sports arena could provide spectacular look-around effects and close-up views using computational photography techniques. In addition, the availability of the many viewing angles will allow for detailed computer vision and sports analytics. However, a practical difficulty is the calibration of these large-scale setups before a match and the challenge of keeping the system calibrated (in real-time) during a match. External factors such as a cheering crowd, wind, or a passing car can cause mechanical vibrations of the small cameras and even the smallest rotation will cause errors in rendering or analysis. In this paper, we investigate how to best initially calibrate a dense multi camera array and how to keep it calibrated while going live. We discuss the software that we developed for robust multi-camera calibration and calibration monitoring and present experimental results for both artificial and real captured data.

INTRODUCTION

Placing many low-cost cameras around a sports field brings new possibilities for 3D sports analysis and immersive viewing. For instance, being able to select the best perspective from more than 100 cameras will lead to a better analysis and judgement call for critical moments in sports. For the consumer the large number of cameras means that an immersive look-around effects can be produced using depth image-based rendering. The result can be viewed on a smartphone or even on a virtual reality headset for an immersive experience. Figure 1 shows a system diagram with algorithms (left) and an experimental outdoor six-camera setup that we built (right).

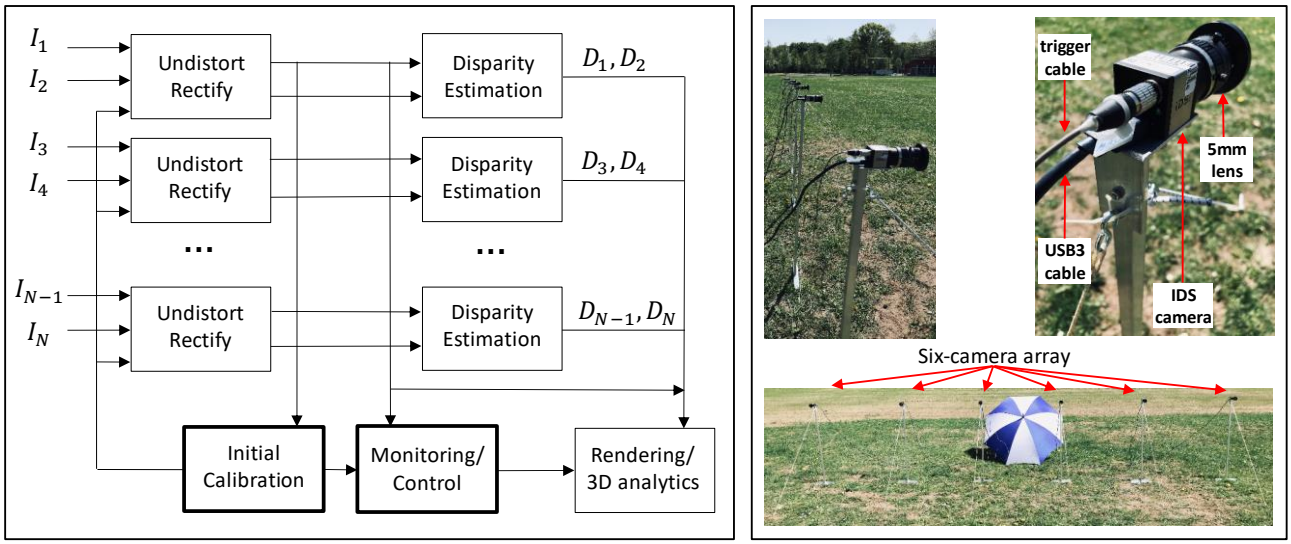

Figure 1 – Multi-camera system for 3D visualization and analytics (left of figure). Input to the system are N camera images $I_1 \cdots, I_N$. These are undistorted/rectified and used for pairwise disparity/depth estimation. Essential in this process are initial calibration and real-time calibration monitoring/control (topics of this paper). Our current experimental setup consists of a six-camera array (right side of figure).

The first step is undistortion and rectification. This step remaps camera images such that lens distortions are removed and adjacent pairs of cameras form a rectified stereo pair. Figure 2 illustrates the process. It is an important step since undistortion and stereo rectification allow for accurate real-time disparity estimation. The resulting disparity maps encode the depth information and can be used for view synthesis or 3D analysis. In previous publications we have shown that for certain camera configurations, disparity estimation and view synthesis can be done both accurately and in real-time. We have also presented an evaluation methodology for depth estimation and view synthesis [3][4]. However, in the past we, and other researchers, have not given sufficient attention to the topic of multi-camera calibration. We know that that even the slightest camera orientation changes (e.g. $0.01°$) can change the parameters of the rectification process such that stereo rectification, disparity estimation and view synthesis all fail. For small-scale camera setups where cameras are fix mounted close together (e.g. 5cm separation) on a metal bar the approach has often been calibrate once using a known calibration pattern and assume that after that nothing will ever move. This approach is clearly not sustainable when moving outdoors where cameras are no longer mounted close together and a known calibration pattern isn't practical due to the larger scale.

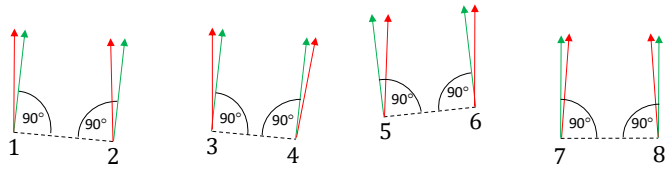

Figure 2 – Pairwise rectification of cameras for an 8-camera linear array. The rectification transforms each image such the optical axes (red arrows) become pairwise parallel (green arrows) and orthogonal to the line (dotted) that connects both cameras. The rectification allows for easier disparity estimation and depth calculation.

Accurate multi-camera calibration is therefore essential for the calculation of rectification transforms, disparity estimation, 3D view synthesis and the performance of semi-interactive 3D measurements. In general, one can identify the following calibration problems that influence a multi-camera setup:

1. Per camera intrinsic parameter calibration: This step concerns the estimation of lens focal length, lens distortion parameters and sensor principal point. The step can be done per camera in the factory or laboratory. The estimated parameters are expected to remain stable under intended use.
2. Per camera colour calibration: The colour properties of cameras can be set per individual camera.
3. Multi-camera photometric calibration: Individual analog to digital conversion gain per camera will result in image intensity differences that have a large effect on disparity estimation, image synthesis and blending/stitching. Automatic gain control is therefore preferably done for all cameras simultaneously in a single control loop. After initial calibration, no changes are needed.
4. Multi-camera extrinsic calibration: Both the synthesis of images for a new virtual viewpoint and 3D measurement rely on very accurate knowledge of the relative pose of the cameras in the multi-camera setup. The pose consists of three position and three orientation parameters.
5. Multi-camera extrinsic monitoring/control: This step involves the real-time measurement of changes in camera positions and orientations given the initially calibrated system. Since individual cameras do not change position so much this mostly concerns the real-time monitoring and estimation of orientation changes.

In this paper we focus on extrinsic calibration and monitoring/control (steps 4 and 5) since these steps have proven to be the most problematic especially for dense outdoor multi-camera setups.

The basic mathematical theory for multi-camera extrinsic calibration already exists for a long time [5]. However, less is known about the reliability and robustness in practice. Moreover, the real-time monitoring and control in the video case is almost never subject of research while this is very relevant for practical setups that capture live events. We have experimentally found it to be difficult to produce reliable software that runs in real-time for a large-scale (outdoor) setup. The reason for this is likely the overall system complexity since multiple image analysis and estimation algorithms are combined resulting in a large number of free parameters. Moreover, we notice that some processing steps are very sensitive to certain system parameters.

The contribution of this paper is to explain how we combine algorithms for feature detection, feature point correspondence estimation and bundle adjustment to arrive at a sufficient quality and robustness level. We identify key system parameters, their values and introduce (visual) monitoring procedures in order to diagnose errors. This makes it possible to quickly intervene and restart calibration when needed. In this paper, we present experimental results for both photorealistic images and for images that we captured outdoors at the High Tech Campus in Eindhoven using our multi-camera setup consisting of six machine vision cameras.

MULTI-CAMERA EXTRINSICS CALIBRATION

Feature detection and correspondence

It is often not practical to get access to the sports playground. Both the initial calibration and calibration monitoring therefore have to rely on detected scene features points. We detect feature points in all camera images separately and for each feature point find corresponding points in all other camera images. Only points that are present in at least three other views are kept. In this manner, the cameras are 'tied' together without the hard requirement that one feature point is visible in all cameras.

Bundle adjustment

Bundle adjustment [5] is the standard approach to simultaneously find camera poses and 3D positions for the feature points. Various algorithms exist for solving this non-linear optimization problem. For dense camera arrays, we noticed that the sensitivity to camera position is rather small. We therefore use the camera positions as measured during installation with a simple measurement tape. In contrast, the sensitivity to camera orientation is typically very high. During bundle adjustment, we therefore solve for the rotation matrices and 3D point positions using:

$$\left(R_1, \cdots, R_{N_{\text{camera}}}, \mathbf{x}_1, \cdots, \mathbf{x}_{N_{\text{point}}}\right) = \underset{R_i, \mathbf{x}_j}{\arg\min} \left(\sum_{i=1}^{N_{\text{camera}}} \sum_{j=1}^{N_{\text{point}}} v_{ij} \left\| \mathbf{f}\left(R_i, \mathbf{x}_j\right) - \mathbf{u}_{ij} \right\|_2 \right)$$

where $\mathbf{f}\left(R_i, \mathbf{x}_j\right)$ is the predicted image position of 3D world coordinate $\mathbf{x}_j$ of point j in camera i using rotation matrix R_i, $\mathbf{u}_{ij}$ is the image position of point j in camera i and $v_{ij} \in \{0,1\}$ is 1 if scene point $\mathbf{x}_j$ was either visible or detected in camera i and 0 otherwise. To decide whether the calibration has succeeded it is important to be able to check the iterative error updates of the fitting system. We therefore use the Powell minimization algorithm [6] to minimize the above error. While convergence is likely slower than the more optimal Levenberg-Marquardt algorithm, the Powell minimization algorithm allows for easier intermediate error understanding and visualization. We use Powell with multiple step sizes that vary from large to small rotation updates along all axes. The multi-scale rotation steps

are: $0.25°$,$0.1°$,$0.01°$ and $0.001°$. The fine rotation steps have the effect that synthesized views become sub-pixel aligned.

MULTI-CAMERA EXTRINSICS MONITORING/CONTROL

Monitoring the position of stable feature points

After the initial calibration, external factors such as a moving crowd of people, wind and temperature can cause significant orientation changes for one or more cameras. To monitor whether each camera is still calibrated we project the stationary 3D scene points $\mathbf{x}_1, \cdots \mathbf{x}_N$ into a reference image for each camera and store the resulting image points $\mathbf{u}_{i,j,\mathrm{ref}}$ as stationary reference points together with the reference image. Note that we make sure that all 3D scene points $\mathbf{x}_1, \cdots \mathbf{x}_N$ indeed correspond to stationary points that are either part of the fixed infrastructure in the stadium or visible markers mounted for this task. This initialization step is typically done semi-supervised (behind the computer) before players enter the playing field. During the game, points will become occluded or will not find a match in a new frame because of a correspondence estimation error (e.g. due to image noise). To exclude these causes for a possible error we estimate the motion of point $\mathbf{u}_{i,j,\mathrm{ref}}$ from frame $t = t_{\mathrm{ref}}$ to frame $t = t_k$ and back from the found point position $\mathbf{u}_{i,j,k}$ at frame $t = t_k$ to frame $t = t_{\mathrm{ref}}$. If the back estimation comes close enough (within 1 pixel) to the original point and both forward and back matches have a sufficiently low match error then we accept the estimated correspondence.

Even when taking the precaution above during the matching process, image noise and illumination differences tend to produce incorrect correspondences (Figure 3). To make the detection of a miscalibration robust, we ignore large position errors. We first sort all position errors for camera i in increasing order:

$$(e_i)_{m=1}^{M_i} \equiv \mathrm{sort}\left(\left\|\mathbf{u}_{i,j,\mathrm{ref}} - \mathbf{u}_{i,j,k}\right\|_2\right).$$

We then define the following criterion for detecting a miscalibration for camera i:

$$e_i\left[\mathrm{floor}\left(\frac{P}{100}M_i\right)\right] > T,$$

where P is the P-th percentile of all sorted errors [pixel] and T [pixel] is a threshold parameter for the error that corresponds with the P-th percentile. We use $P = 25\%$ smallest errors and set T to a tuneable parameter (typically order of a few pixels). Setting T to a small value will allow the detection of small calibration errors whereas choosing a larger value for T will ignore those small calibration errors (e.g. a small vibration caused by light wind).

The robust approach for detecting a miscalibration has large implications. It means that the multi-camera array can be monitored in a robust way. Moreover, the need for re-calibration (i.e. re-estimation of camera orientation) can be reduced to cases where there is really a problem with one or more of the cameras. In addition, in case we notice that many cameras

show problems at the same time we may conclude that, the mechanical mounting of the cameras is wrong and a different mounting approach will be needed in the future. Logging the detection results is therefore crucial in order to learn from field experiments.

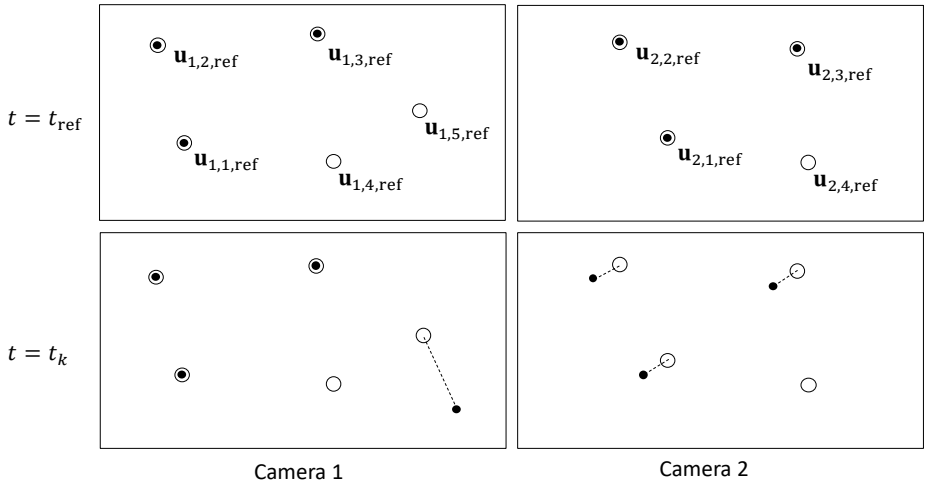

Figure 3 – Feature points in the reference frame ($t = t_{\mathrm{ref}}$) denoted with open circles and in a next frame ($t = t_k$) denoted with closed circles. Reference points are also shown at $t = t_k$. Correspondences are indicated with a dotted line. Correspondence can be absent due to occlusion (point 4). A false correspondence is still possible (point 5) but robustness to this is built in our detection algorithm. Camera rotation errors typically cause all points to change position (bottom right of figure).

Updating camera orientation estimates

Once we detect that a camera is no longer calibrated, we need to re-estimate the camera orientation parameters. Since the orientation estimation must be done in real-time for potentially all cameras, we use the efficient non-iterative Kabsch algorithm [7] to estimate the rotation matrix for the deviating cameras. Due to its efficiency for rotation estimation, The Kabsch algorithm is still used today in robotics [8]. The known scene points $\mathbf{x}_j$ that resulted from the bundle adjustment step can be used to calculate a 3D reference configuration of scene points. Under the assumption that a camera has only changed orientation and not position, the distances from the cameras to these scene points must not change. (Only the 2D position of projected image points change when a camera rotates). Using this assumption, we can compute and store reference ranges $r_{j,\mathrm{ref}}$ from all cameras to all scene points j at the initialization stage. Using these reference ranges, we calculate the 3D camera coordinates of scene points for all cameras i:

$$z_{\mathrm{c},j} = \frac{f r_{j,\mathrm{ref}}}{\sqrt{u_j^2 + v_j^2 + f^2}} \qquad x_{\mathrm{c},j} = \frac{u_j^2 z_{\mathrm{c},j}}{f} \qquad y_{\mathrm{c},j} = \frac{v_j^2 z_{\mathrm{c},j}}{f}$$

where f is the focal length [pixel] and u_j, v_j are the image coordinates of point j [pixel]. This gives calibration points in both world space and camera space. Using the Kabsch algorithm,

we now estimate the optical rotation matrix. Note that this algorithm has a complexity that is quadratic in the number of feature points per image. Since we already required high-quality feature point correspondences, the number of feature points will typically be rather small (e.g. maximally 50). The required number of computations is therefore limited. The Kabsch algorithm has the advantage that it is a direct method and can therefore efficiently solve for large rotation errors. In contrast, the stepwise, iterative, Powell minimization method would require many updates to converge. The downside of using Kabsch is that it does not minimize a re-projection error in image space but a 3D distance error instead and hence relies more heavily on the accuracy of the 3D point estimates. As a result, the rotation estimate will be less accurate. To remove the remaining bias we can use Kabsch only as first estimate that we can then further refine using the original re-projection error for a fixed number of small rotation updates.

EXPERIMENTAL RESULTS

Photorealistic image simulation is an indispensable tool when it comes to verification of algorithms and software. For instance, with artificial cameras it is possible to simulate an orientation disturbance in one of the cameras and investigate whether calibration monitoring and control is able to recover from this error.

Eight camera linear array using photorealistic simulations

A soccer stadium with players was acquired from the Unity Asset Store [1]. The 3D photorealistic rendering package Blender [2] was used to create images of an empty stadium and a stadium with players. Images were synthesized for a camera array consisting of eight cameras, spaced 1m apart. To test the initial calibration process, each camera was randomly rotated with a rotation between 0 and 2 degrees. Figure 4 shows bundle adjustment results for the first iteration (top row) and for the final iteration (bottom row) for cameras 1, 2 and 8. As can be seen, the projected 3D scene points (red dots) in general end up at the detected image position (green circle). Note that some outliers are still visible in the final iteration but excluded during the fitting process.

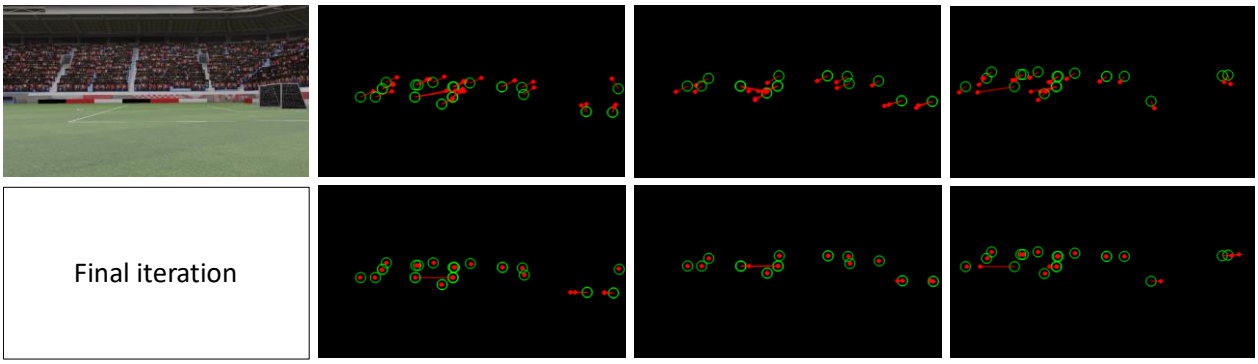

Figure 4 – Bundle adjustment visualizations for cameras 1, 2 and 8 for the first iteration (top row) and the final iteration (bottom row) for an empty soccer stadium. The red points are the re-projected 3D scene points into each image given the current estimates of the parameters: $R_1, \cdots, R_{N_{\text{camera}}}, \mathbf{x}_1, \cdots, \mathbf{x}_{N_{\text{point}}}$. Green circles are the original image feature points. The red lines connect the re-projected points with the original feature points.

To further validate estimated rotation parameters, we used the eight reference images to synthesize a perfectly aligned camera array. Results are shown in Figure 5. When comparing the image position of the soccer goal it can be seen that for the synthesized images (bottom row) its position is correctly aligned.

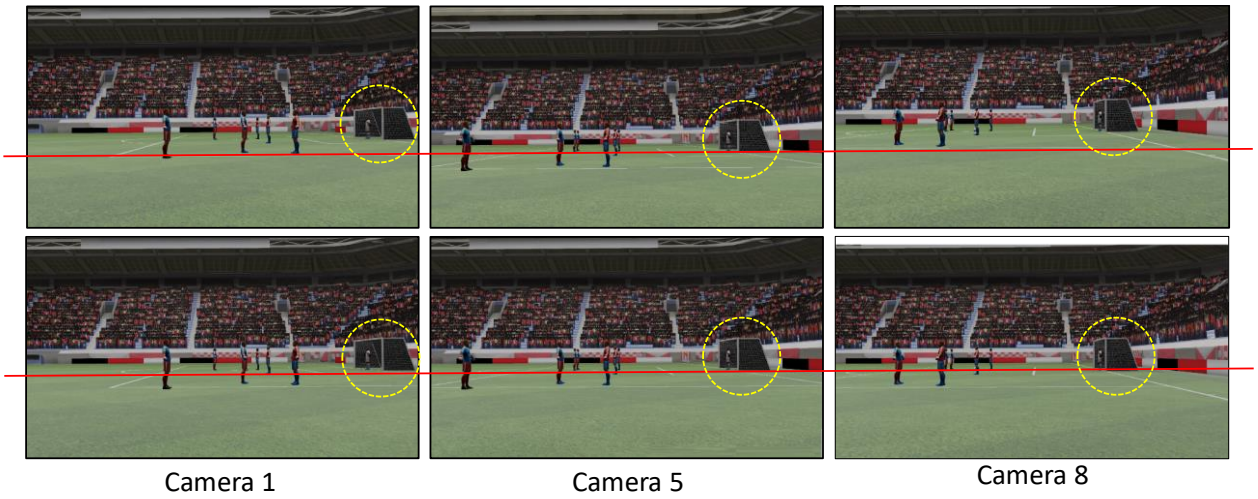

Figure 5 – Eight virtual cameras were synthesized (only three are shown) all pointing in the same direction. The top row shows the original non-rectified camera images. The bottom row shows the synthesis results for cameras 1, 5 and 8. The soccer goal (yellow circle) is now vertically aligned (red line) in all camera views.

To test calibration monitoring, an artificial rotation error of 0.5 degree magnitude was introduced in the second frame after the reference frame in the image sequence for one of

the cameras. The effect of calibration monitoring for this situation is shown in Figure 6. Images in the first and second row of Figure 6 show that most reference points are matched with a corresponding new point in a new frame and that if they do, the position remains stable. However, in the third row of Figure 6 it can be seen that for camera 2 that all points have shifted position. We can conclude that the error is (also visually in the plot) easily detected.

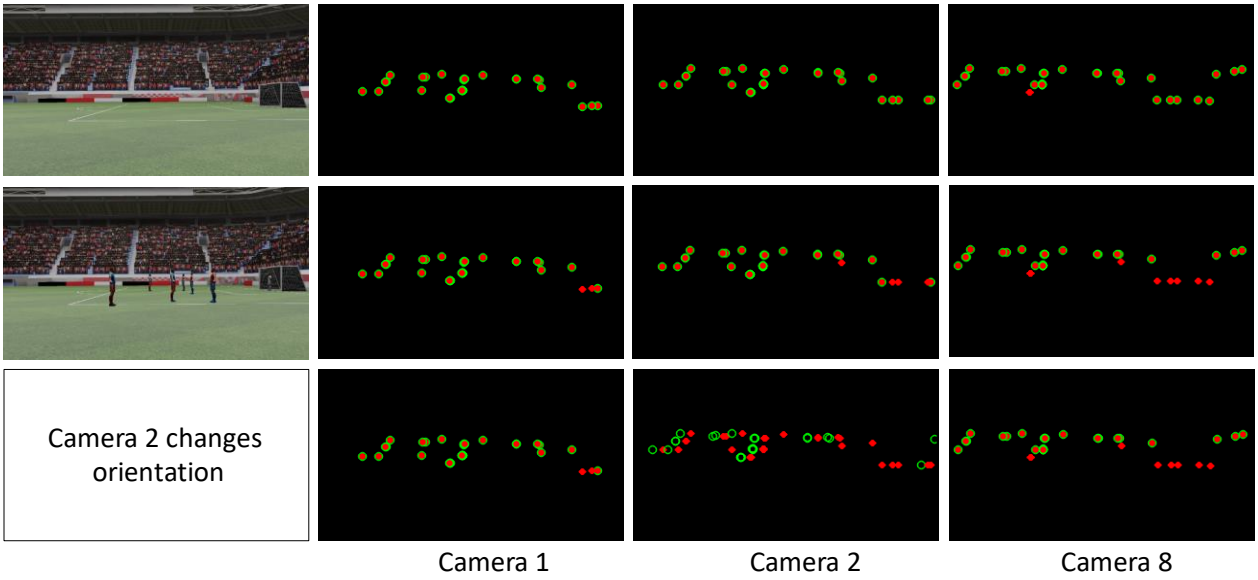

Figure 6 – Calibration monitoring results for simulated camera images. The 3D scene points are first projected in the reference frame (top row). Calibration status is judged correct when a given fraction of corresponding feature points (green circle) do not change position (middle row). When camera 2 is rotated on purpose 0.5 degree this is detected since all feature points now change position (bottom row).

Six camera linear array: outdoor capture

For the outdoor experimental setup, we validated the entire extrinsic calibration process: feature detection, feature correspondence estimation, bundle adjustment, orientation estimation, stereo rectification and view synthesis. Figure 7 compares raw input images with images after the final view synthesis where virtual views all point in the same direction.

Figure 7 – Comparison of original non-rectified camera images with synthesized images where the virtual views for the six cameras (only four are shown) all point in the same direction. The top row shows the original non-rectified camera images. The bottom row shows the synthesis results for cameras 1, 2, 5 and 6. After view synthesis, the light-blue object becomes vertically aligned (red line) in all camera views. This is a practical verification that all steps (feature detection, bundle adjustment, rectification, and view synthesis) were successful.

CONCLUSIONS

Moving from a laboratory-scale rigidly-mounted multi-camera array to a large-scale multi-camera outdoor setup with distances between cameras increasing by a factor of 20 (from 6cm baseline to over 1m baseline) has proven to be extremely difficult. We have found that these problems were mainly due to the intricate dependence of depth estimation and view synthesis on the success of multiple calibration steps (both photometric and geometric). We have identified robust approaches for the most important calibration steps and presented robust ways to calibrate a multi-camera array, monitor its status and re-calibrate when needed.

Not all problems will have been solved. While results for an experimental outdoor setup of six parallel cameras look encouraging, we expect new problems to arrive when we further scale up to more than 50 cameras and those camera will be spatially organised in new configurations (e.g. multiple sides of a sports field).

REFERENCES

1. Unity Asset Store: https://assetstore.unity.com/packages/3d/characters/soccer-players-stadiums-pack-105891
2. Blender 2.8: https://www.blender.org/.
3. C. Varekamp, B. Kroon, B. Sonneveldt, A. Willems. Depth based room-scale six degrees of freedom virtual reality capture and processing. *IBC 2018*.
4. C. Varekamp. Evaluating the quality of multi-camera video capture and view-point interpolation for 6DoF AR/VR applications. *IBC 2019*.
5. S. Agarwal et.al. Bundle Adjustment in the Large. *European Conference on Computer Vision (ECCV)*, pp 29-42. 2010.

6. M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*. 7 (2): 155–162. 1964.
7. W. Kabsch W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923. 1976. [https://onlinelibrary.wiley.com/doi/abs/10.1107/S0567739476001873?sentby=iucr]
8. X. Liu and R.D.Wiersma. Optimization based trajectory planning for real-time 6DoF robotic patient motion compensation systems. *PLOS ONE*, January 11, 2019. [https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0210385]