FORTH-ICS / TR-397 August 2009

A checkerboard detection utility for intrinsic and extrinsic camera cluster calibration†

Thomas Sarmis and Xenophon Zabulis
and Antonis A. Argyros

[0]† This work has been supported by the FORTH-ICS internal RTD Programme 'Ambient Intelligence and Smart Environments'.

A checkerboard detection utility for intrinsic and extrinsic camera cluster calibration

Thomas Sarmis and Xenophon Zabulis
and Antonis A. Argyros

Computational Vision and Robotics Laboratory
Institute of Computer Science
Foundation for Research and Technology — Hellas (FORTH)
N. Plastira 100, Vassilika Vouton
Heraklion, Crete, 700 13 Greece

Web: `http://www.ics.forth.gr/cvrl`
E-mail: {sarmis | zabulis | argyros}@ics.forth.gr
Tel: +30 2810 391600, Fax: +30 2810 391601

Technical Report FORTH-ICS / TR-397— August 2009

1 Introduction

Camera clusters are increasingly employed in a wide spectrum of applications ranging from surveillance, security and gaming applications, to applications that monitor an environment to infer social interaction for the anticipation of user needs. In addition, smaller clusters of cameras have been traditionally utilized for stereo reconstruction in a great diversity of applications with examples found in domains such as cultural heritage, medicine, three-dimensional photography and tele-immersion.

A crucial stage of setting up a camera cluster is the intrinsic and extrinsic calibration of its cameras, so that 3D information can be extracted by combining information acquired through the available views. Several different approaches to the calibration of cameras and camera clusters exist, employing a large diversity of calibration targets that provide reference points for the calibration process. After the seminal work of Tsai [10], the wide majority of these methods adopt the use of a checkerboard as a calibration pattern. The application of this method requires that a checkerboard and its corners are detected and precisely localized in the image. This typically requires user intervention and turns calibration into a tedious process. As reviewed in Sec. 2 of this report, techniques that reduce the amount of this intervention exist, but do not provide a fully automatic means of detecting the checkerboard in a wide variety of viewing conditions. In general, a complete, totally reliable and fully automatic procedure for achieving camera cluster calibration is not available and many of the existing techniques require hardware-specific configurations. In addition, although several approaches exist in the literature for the calibration of camera clusters, only a few approaches are publicly available.

In this report, we propose a method for the automatic detection of a checkerboard and its corners which are used for intrinsic and extrinsic camera calibration. The automation of this process greatly simplifies calibration. The data that are collected by the proposed method can be directly input to traditional, publicly available software tools for camera and camera cluster calibration. It is shown that besides automation, the proposed method increases both the quantity and the quality of the collected data, resulting in an increase of calibration accuracy.

The remainder of this report is organized as follows. In Sec. 2, related work is reviewed and, in Sec. 3, the camera cluster installation where the proposed method was tested is described. In Sec. 4, the proposed method for the detection of the checkerboard and the localization of its corners in the acquired images is presented. In Sec. 5, a method for the identification of the detected corners on the checkerboard calibration target is proposed. Finally, Sec. 6 summarizes this work.

2 Related work

One way to obtain input for the calibration of a camera cluster is by turning on a LED in a dark environment [9, 1], so that the stereo correspondences required for the calibration become directly available. Such approaches require control of the

environment illumination, as well as photometric adjustment of the cameras so that the light source is imaged in a way that guarantees that the spatial uncertainty of the LED's image projection is minimal. Furthermore, [1] is not fully automatic as manual calibration of one or more cameras is required to provide with metric information, while [9] requires Euclidean stratification and non-linear minimization.

On the other hand, employing a checkerboard as a calibration target is much more convenient, as the construction of the target is simple and no particular photometric adjustment is required for the cameras. Nevertheless, the detection of the checkerboard and its corners in the calibration images is required. The toolbox in [2] reduces the manual intervention required for this detection, by prompting the user to indicate the four reference corners of the checkerboard in each input calibration image. The rest of the corners are then automatically detected.

In greater relevance to the proposed approach are the methods presented in [11, 8], which automatically detect a checkerboard and its corners. However, as shown in the experiments of Sec. 4 the method in [11] is not able to detect corners in very oblique views of the checkerboard, constraining, this way, the collection of high-quality calibration data. Additionally, both methods in [11, 8] require that the checkerboard is fully imaged in order to be detected. This is often difficult to be achieved simultaneously for multiple views during the extrinsic calibration of a camera cluster. The method in [8] also requires that certain checkers are marked in order to associate detected corners to physical points on the checkerboard.

The proposed method detects the corners of a checkerboard even if it is partially visible and/or imaged in very oblique views. This greatly facilitates the calibration process. Moreover, the proposed method, as well as [11], utilizes a checkerboard which is comprised by an even number of checkers in the one dimension and an odd in the other, in order to establish this association automatically. It also offers the capability of combining detection results from multiple cameras acquired across time to extrinsically calibrate them and to refine their intrinsic calibration through bundle adjustment.

The approach in [6] utilizes scene reconstruction to improve an existing calibration, but still an initial calibration is required. Cameras are required to be located rather proximate to each other, in order for stereo correspondences to be accurately established. An advantage of such an approach, though, is that calibration can be continuously updated, as due to a number of factors (e.g. change of temperature, vibrations), the initial calibration can be cast invalid. In the proposed approach, we provide means for facilitating the bootstrapping of the calibration process.

3 Multicamera system setup

The multicamera system that has been employed as a testbed in the context of this work is installed in a $5 \times 5\, m^2$ room and consists of 8 FireWire cameras ($66^\circ \times 51^\circ$ field of view) that are mounted at the corners and at the in-between mid-wall points of the

room viewing it peripherally in steps of approximately 45°. Two additional cameras are mounted on the ceiling, with their optical axes approximately normal to the floor.

With respect to computational resources, 8 CPU cores are employed in total, equally distributed in two computers. A dedicated local area network link of $1\,GB$ bandwidth is reserved for their communication. A frame rate of $10\,Hz$ has been achieved when employing 8 views in a resolution of 960×1280 pixels. Latency was measured to be $\approx .4\,sec$, when all images had to be gathered in the same computer.

A crucial assumption in the majority of applications that employ a camera cluster is that images are synchronously acquired from all views. In the presented setup, synchronization is implemented across cameras through a dedicated FireWire bus and the use of timestamps [7]. The employed synchronization method guarantees that the time lag in the acquisition of any pair of images is below $125\mu sec$. The set of images acquired simultaneously from the camera cluster is, henceforth, referred to as a *multiframe*.

4 Checkerboard detection and intrinsic calibration

The proposed camera calibration approach focuses in collecting high quality calibration data in a fully automatic and convenient fashion. The developed method exhibits the capability of detecting checkerboard corners in challenging images that cannot be appropriately handled by conventional methods. This is beneficial to the calibration accuracy, as it increases both the quality and the quantity of the data utilized in the calibration process.

In a first step, the proposed method utilizes the method in [11] that detects checkerboard corners. This method detects checkerboard corners only in the case of moderate camera view obliqueness and lens distortion. To successfully employ the above method, the checkerboard is initially imaged in the center of the camera's field of view and in postures of moderate obliqueness. Checkerboard corners are detected by [11] and a coarse estimate of the camera's intrinsic and lens distortion parameters is obtained, by feeding the detected checkerboard corners to the method in [2]. This estimate is continuously refined as more calibration data become available.

At a second step, in order to collect more and better calibration data, corners are detected in images where the checkerboard may appear in the periphery of the view and under a very oblique perspective. The collection of such data contributes to the robustness and accuracy of the calibration, as the whole field of view of the camera is covered. Additionally, the great variability of checkerboard postures provides stronger constraints for the subsequent calibration parameter estimation. In the following subsections, these two steps are presented in more detail.

4.1 Conventional corner detection and intrinsic calibration

Initially, the method for corner detection in [11] is employed. Typically, this fails to detect the checkerboard in highly oblique checkerboard views, however it does detect

some of the checkerboard corners which are then organized in a list data structure. The detected corners are associated to checkers and organized on a spatial grid that corresponds to the structure of checkers on the calibration board. More specifically, for each corner in the input list, the 8 nearest corners in the image are selected. Each combination of 4 corners is evaluated to determine if these 4 corners occur on the same checker or not. Two criteria are used:

1. At least three out of the four corners are not collinear. This is tested by first estimating the line that passes through the four corners and then measuring the distance of each corner from that line. Four corners are considered to belong to a checkerboard square if the ratio of the sum of these distances to the perimeter of the box is above a threshold (1.0).

2. Almost no edgels occur inside the checkerboard square. This is tested by performing edge detection [3] on the input image and counting the edgels inside the candidate checker. Four corners are considered to belong to a checkerboard square if the ratio of these edgels to the total area of the square is below a threshold (0.01).

In order to determine that four corners belong to the same square both of the above criteria must be satisfied. The corners are assigned to the checker in clockwise order.

When all available corners are associated with checkers, a connected components labelling technique is employed. Two checkers are considered to be connected if they share one or two common corners. In order to determine and represent the spatial arrangement of the corners, we select the first checker of the largest connected component and we define its first corner as the top-left. Then, by examining which corners are common with the connected checkers we identify the type of connectivity (North, South, East, West, NorthWest, NortEast, SouthWest, SouthEast) as well as the top-left corner of the connected component. Then we select the connected checkers and repeat the same procedure recursively, until all checkers have been visited.

4.2 Rectification and intrinsic calibration refinement

By considering the fact that the checkerboard corners form a rectangular grid in the world, a homography is computed. This homography warps the image so that the checkerboard appears frontoparallel and in its true aspect ratio. The computed homography minimizes the backprojection error between pairs of corresponding points across the image and the output homography. In the warped view, corner locations are predicted given the known size of the checkers. The corners are then detected by applying corner detection [4] at the neighborhoods of these predictions. This may yield some spurious detections, as the limits of the checkerboard in the image are yet unknown. To discard spurious detections, their neighborhoods are compared with a synthetic template, using Normalized Cross Correlation applied with a conservative

threshold. Since the view of the checkerboard is frontoparallel, the ideal appearance of each corner is a priori known. More specifically, each corner is made up by four squares, two black and two white. Two templates are employed, one with a black top-left square and another with a white top-left square. The Normalized Cross Correlation of the synthetic templates with the corresponding image region is calculated at the location of each corner prediction. If the maximum of the two results indicates a high degree of similarity (> 0.85) and the absolute difference between the two results is high (> 0.4), this means that (a) the best scoring template constitutes a good match and (b) that the worst scoring template does indeed not match the corresponding image region[1]. Thus, it can be assumed that the underlying region images a checkerboard pattern and that it is not an accidental match. Therefore, the predicted location is considered to belong indeed to a checkerboard corner.

In Fig. 1 the result of the above procedure is illustrated, for two images. The first example (top row), shows the results of initial calibration. The inaccurate compensation of lens distortion indicates that the estimation of the camera parameters is still coarse. In the second example (bottom row), camera parameters have been more accurately estimated.

For each incoming calibration image, [2] is re-executed for all the corners detected up to that point. This way, the estimates of the intrinsic and lens distortion parameters are iteratively improved and corner detection becomes increasingly more accurate. In Fig. 2, the performance of the proposed corner detection method is compared to that of [11] and, in Fig. 3, the enumeration of corners organized on a grid is illustrated.

It should be noted that the proposed method does not require that the whole checkerboard is visible, but makes use of all the available input (detected corners). In contrast, [11] requires that the total checkerboard is visible in an image. During the extrinsic calibration of cameras (see Sec. 5) this proves to be a rather strong constraint, as it might be difficult or even impossible to acquire full views of the checkerboard from two or more cameras. Therefore, the proposed method not only provides with more detected corners but also, yields significantly more useful data to the calibration procedure, as more views of the checkerboard can be utilized.

In a calibration experiment (Fig. 2) that has been performed to assess the accuracy of the proposed approach, a checkerboard of 14×13 checkers, each of a size of $90x90\,mm$, was imaged 210 times in a wide variety of postures and distances from a 960×1280 camera with a wide ($66^\circ \times 51^\circ$) FOV and significant lens distortion. The proposed method utilized corners from 91 frames as opposed to 14 that [11] did, due to the failure of the latter method to detect the checkerboard in as many frames as the proposed method was able to. The corners that were input to the calibration algorithm were $12,293$ versus $2,184$ and the backprojection errors were 0.33 and 0.46 pixels, respectively.

[1] We have encountered cases that an accidental pattern in the environment, not belonging to the checkerboard, would provide a spuriously high correlation score. Requiring that the corresponding image region is also very dissimilar to the second template, eliminates such spurious matches.

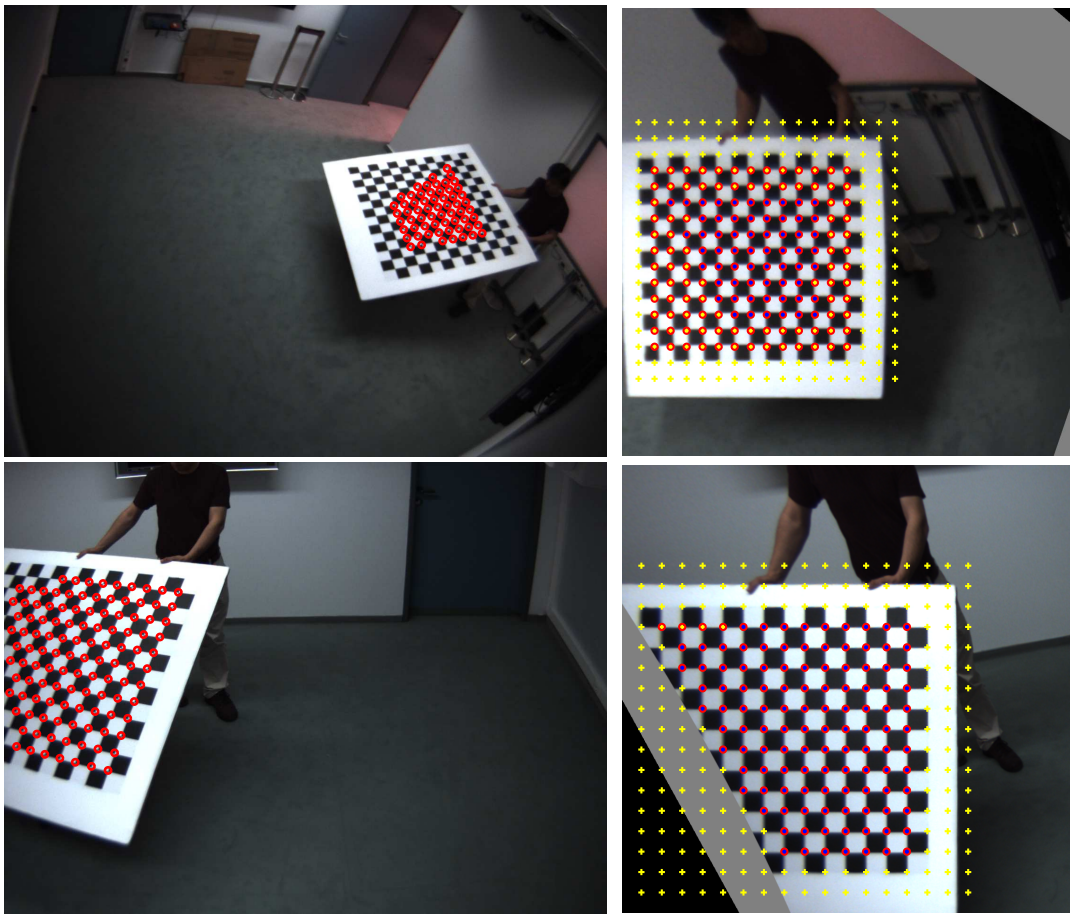

Figure 1: Two examples of checkerboard corner detection and organization (one example per row). Left column: the corner detection result of the method in [11]; some corners are not detected, while the spatial organization of the corners is undetermined. Right column: the obtained homography, with the generated seeds and detected corners superimposed (yellow crosses and red circles, respectively). In this homography, the four missing corners are detected.

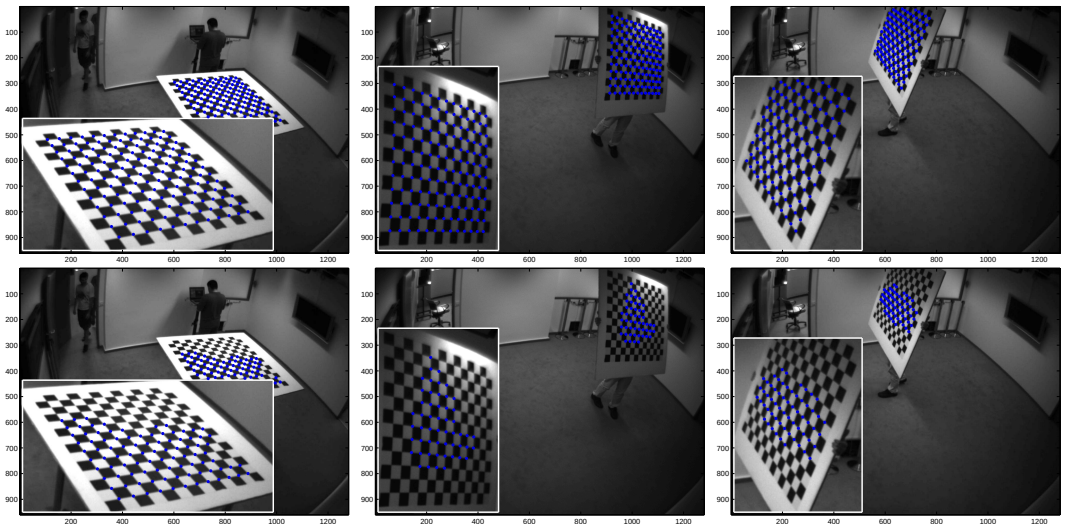

Figure 2: Checkerboard corner detection for camera calibration. Top row shows results from the proposed method and bottom row shows results obtained with the method in [11].

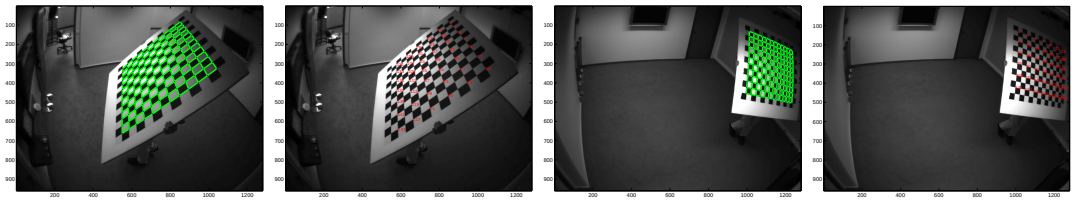

Figure 3: Grid organization of checkerboard corners. Detected corners from two views are organized and enumerated on a grid. For clarity, enumeration is shown for even checkers on each dimension. Left two images correspond to the 1st view and right two to the 2nd.

5 Extrinsic calibration

Extrinsic calibration of a camera cluster requires correspondences across images acquired simultaneously from multiple views, in order to estimate relative camera positions and orientations. A method to establish such correspondences, based on the proposed checkerboard detection method, is presented. Additionally, bundle adjustment is performed to refine the estimates of both the extrinsic and intrinsic camera parameters.

The developed calibration procedure is fully automatic, facilitates the collection of calibration data for multiple checkerboard poses and does not require that the checkerboard is visible in all images of a multiframe. In contrast, extrinsic calibration of a camera cluster with [11] is based on a single multiframe, where the checkerboard must be fully visible in all images. This requirement can be difficult or even impossible to fulfil for certain camera placements. Furthermore, the calibration obtained based on a single multiframe is, typically, less accurate than the result obtained by processing several of them in a bundle adjustment framework.

Extrinsic calibration is performed as follows. First, multiframes are acquired and a checker grid for each frame is generated, using the technique described in Sec. 4. Next, grid corners are enumerated (e.g., 3^{rd} horizontal, 2^{nd} vertical corner) and associated with physical points on the checkerboard. Given this association, correspondences across cameras can be directly established. This association is computed by identifying the top-left checker of the checkerboard. This is based upon two assumptions (a) one of the dimensions of the checkerboard consists of an even number of checkers and the other dimension consists of an odd number of checkers and (b) the color of the top-left square is known (black or white).

More specifically, let W be the number of corners per row, and H be the number of corners per column. In order to identify the top-left corner we need to identify the corners at nominal checkerboard coordinates $[1,1]$, $[W,1]$, $[1,H]$ and $[W,H]$. Let us assume that we have detected a partial checkerboard with dimensions W_d and H_d, in an input image. Let also the corners located at $[1,1]$, $[W_d,1]$, $[1,H_d]$, $[W_d,H_d]$ be defined as the "extremal corners" of the checkerboard. In the following, we test if these four corners are indeed the extremal corners of the checkerboard. For each candidate corner, we consider the three nearest seeds on the frontoparallel homography (see Sec. 4), which have *not* been verified to be checkerboard corners. If all of these three seeds occur within the image, we consider the candidate corner to be indeed an extremal one. The intuition behind the above reasoning is that since all three seeds are not considered as checkerboard corners, there are not any other checkerboard corners near the candidate corner; otherwise they would have been detected as regular checkerboard corners and there would not be unverified seeds available. In Fig. 4, the result of the above process is illustrated.

In order to proceed, we need to identify at least two verified extremal corners. From those, we can find the number of corners at one or two dimensions (two, if the extreme corners are diagonal) of the detected checkerboard. With this information

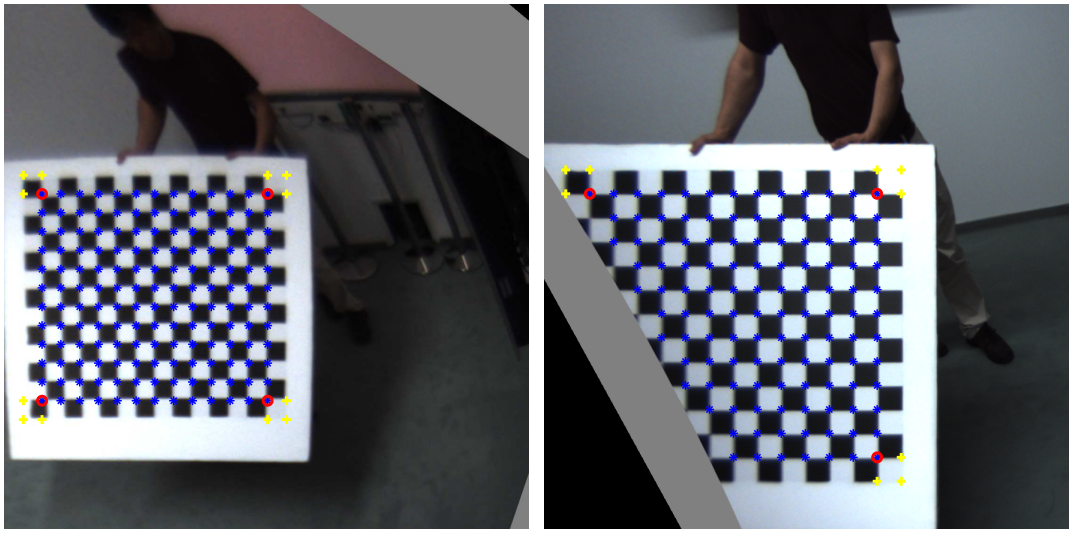

Figure 4: Two examples of recognition of checkerboard extremal corner detection. Superimposed blue markers indicate the detected and verified checkerboard corners. The red circles indicate the detected extremal corners of the checkerboard. For each extremal corner, the three yellow crosses mark the three nearest seeds, which are used to test whether the corner is indeed an extremal checkerboard corner. The images correspond to the two examples shown in Fig. 1.

we can decide whether the calculated "width" and "height" correspond to the actual width and height of the checkerboard. If they do not exhibit such a correspondence, then we transpose them.

Since we know that width and height are now properly assigned to the detected checkerboard, the only ambiguity is if the top-left corner is detected correctly or we need to rotate our labels by 180 degrees. But since the checkerboard has even width and odd height, the top-left and bottom-right checkers are of different colors (and top-right is different from bottom-left). By inspecting the color of one of the available verified extreme corners (whether it is black or white) we can decide if it is the correct color, or if we need to rotate the labels by 180 degrees.

The above corner identification approach is utilized in the detection of cases where the checkerboard has not been detected by the method. If these criteria are not fulfilled and if less than the 1/8 of the checkerboard corners have been detected, the method rejects the acquired calibration image from the input. In this way, images that might provide erroneous calibration input are automatically excluded.

The 3D coordinates of the recognized corners are required for bundle adjustment and are computed as follows. For each frame, the checkerboard is assumed to be coincident with the XY plane of a 3D coordinate system relative to which camera position and orientation is computed. Moreover, the 3D coordinates (0, 0, 0) are assigned to its top-left corner. Since the cameras are static, a transformation that

registers the camera centers is computed by estimating the rotation and translation that minimizes their squared distances. The inverse of this transformation applied to the 3D locations of the checkerboard corners (points on the XY plane), provides the estimates of their 3D coordinates. The 3D to 2D correspondences of the identified corners are passed on to bundle adjustment [5]. The final result of this process is an estimate of the extrinsic camera parameters as well as a refinement of the camera intrinsic parameters estimates.

As expected, it is observed that more accurate extrinsic calibration results are obtained when calibration data cover the entirety of the reconstruction volume, rather than when extrinsically calibrating from a single checkerboard pose. By utilizing more than one multiframe this becomes possible. At the same time, by using the proposed method, calibration is fully automated. As an indication of the obtained improvement, an experiment was performed utilizing the same dataset as in Sec. 4. The backprojection error using a single multiframe was 0.403 pixels. When all frames were utilized, the error dropped to 0.18 pixels.

The acquisition of data for the calibration process is facilitated by an on-line user-interface (see Fig. 5), which informs the user as to if the acquisition of more data is required for an accurate calibration of the camera cluster. Regarding intrinsic camera calibration, this interface visualizes the projection of the detected checkerboard corners on the field of view of each camera, indicating if there are areas of this field that have not been covered. Regarding extrinsic camera calibration and bundle adjustment, this user interface visualizes the 3D coordinates of the detected corners, indicating if the entirety of the reconstruction volume has been covered by the motion of the calibration target. In this way, the user is guided as where to target the acquisition of additional input for the calibration procedure, both in terms of field of view as well as in terms of reconstruction volume.

6 Summary

In this report a method for the automatic detection of a checkerboard and the identification of its corners has been proposed. In addition, techniques for the intrinsic and extrinsic calibration of a camera cluster which are based on this method, have been presented. These techniques are publicly available semi-automatic methods, which with the incorporation of the proposed checkerboard detection method have been fully automated.

The proposed checkerboard calibration technique detects and recognizes the corners of the checkerboard by predicting their image locations, based on initial estimates of the camera intrinsic and lens distortion parameters, which are continuously refined. The calibration process itself is based on publicly available and well established calibration and bundle adjustment methods. The proposed checkerboard detection method facilitates their application by automating the checkerboard detection operation. The proposed checkerboard detection technique is also experimentally compared against

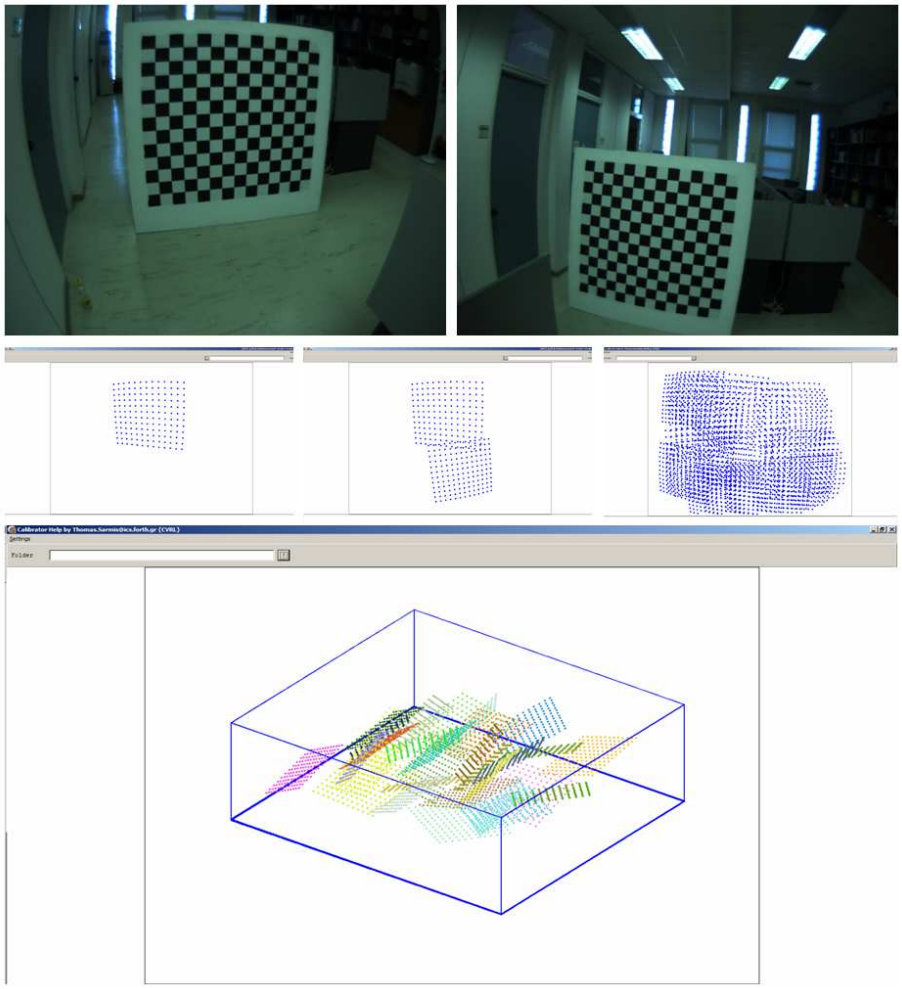

Figure 5: An online user interface indicates if the motion of the calibration checkerboard has covered a camera's field of view and the reconstruction volume. Top two rows show the acquired images and the corresponding field of view coverage for a camera. Bottom row shows estimated 3D points, estimated by triangulation, passed on to the bundle adjustment procedure.

the method in [11], demonstrating that it can detect the checkerboard calibration target in more challenging conditions (large obliqueness, significant radial distortion) than the former technique.

Employing the proposed utility software, numerous calibration images can be effortlessly utilized for the calibration of cameras providing, thus, higher calibration accuracy without the requirement of user intervention. This becomes important not only for intrinsic camera calibration but for extrinsic calibration as well, as the detected and recognized corners are employed in both processes. In the case of extrinsic calibration, the automatic detection of the checkerboard and its corners becomes of particular significance as, typically, a large number of images is required to cover a wide area, such as the one covered by the presented camera cluster. We have observed that the utilization of the reported software saves a considerable amount of time during camera cluster calibration, while at the same time, results in increased calibration accuracy.

References

[1] J. Barreto and K. Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In *International Work. on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, pages 37–44, 2004.

[2] J.Y. Bouguet. Camera calibration toolbox for Matlab. http//www.vision.caltech.edu/bouguetj/calib_doc.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[4] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[5] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[6] J. Ponce and Y. Furukawa. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.

[7] Point Grey Research. Multisync user manual. http://www.ptgrey.com/, 2008.

[8] W. Sepp and S. Fuchs. DLR CalDe and DLR callab. http://www.robotic.dlr.de/callab/. Institute of Robotics and Mechatronics, German Aerospace Center.

[9] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[10] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, FL, 1986.

[11] V. Vezhnevets. OpenCV calibration object detection. http://graphics.cs.msu.ru/en/research/calibration/opencv.html.