Easy–to–Use Calibration of Multiple–Camera Setups

Ferenc Kahlesz, Cornelius Lilge, and Reinhard Klein

University of Bonn,
Institute of Computer Science II,
Computer Graphics Group
Römerstrasse 164, D-53117 Bonn, Germany
{fecu,lilge,rk}@cs.uni-bonn.de

Abstract. Calibration of the pinhole camera model has a well–established theory, especially in the presence of a known calibration object. Unfortunately, in wide-base multi–camera setups, it is hard to create a calibration object, which is visible by all the cameras simultaneously. This results in the fact that conventional calibration methods do not scale well. Using well-known algorithms, we developed a streamlined calibration method, which is able to calibrate multi–camera setups only with the help of a planar calibration object. The object does not have to be observed by at the same time by all the cameras involved in the calibration. Our algorithm breaks down the calibration into four consecutive steps: feature extraction, distortion correction, intrinsic and finally extrinsic calibration. We also made the implementation of the presented method available from our website.

1 Introduction

Calibration of the pinhole camera model has a well-established theory [1]: given a number of 3D–2D correspondences the free parameters of the desired pinhole-model variant (having/not having skew, known aspect ratio, ...) can be estimated. Although a lot of research have been carried out to exploit properties of the observed scene for calibration, because of their higher accuracy, the correspondences still tend to stem from image(s) of calibration objects. 3D calibration objects have the advantage that one image taken by the camera suffices for calibration, 2D objects used by homography-based methods, however, are easier to manufacture and handle. Multiple cameras can be easily calibrated in a common reference frame as long as they are able to observe the calibration object simultaneously. This is not an issue if the cameras has approximately the same point of view (near-baseline case), but problems arise as the point of views get farther away from each other and conventional methods do not scale well (see Figure 1).

For systems with pre-known static camera setup appropriate calibration objects can be designed and calibration can be carried out automatically. In wide-base multiview settings the complexity of the calibration objects (and possible

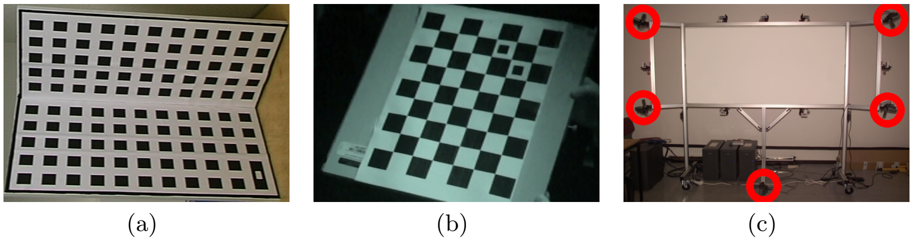

(a)
(b)
(c)

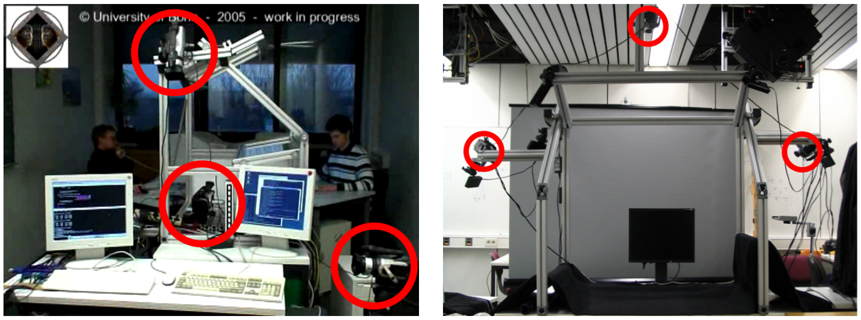
Fig. 1. Typical 3D (a) and 2D (b) calibration objects and the model of a 3D display (c) with 5 cameras (red circles) to track the users actions. Even for this small 5 camera setup the straightforward use of any of the calibration objects is hindered by the fact that they cannot be seen by all the cameras at the same time.

additional hardware, e.g. a LED-matrix controller) depends inherently on the camera setup. In multiview human tracking methods for medium to large volumes with three or more cameras (typical in ‘intelligent rooms’ or AR/VR) it is often not feasible or even possible to construct calibration objects which are observable by all cameras. Another drawback of such custom calibration objects is that change of the camera setup might render the object obsolete.

2 Motivation

Connected to our research in Computer Vision based Human-Computer Interaction (HCI) (more specifically ‘Hand-Tracking’, see Figure 2), we several times faced the problem that we had to be able to flexibly reposition our cameras. During the development of vision-based interfaces camera placement and image processing algorithm selection are intertwined. On the one hand, the available positions of the cameras are usually constrained by the environment itself (free instrumentation is not possible), on the other hand, success or failure of algorithms might depend on the point of view. In order to be able to freely experiment with camera-placement, processing algorithms and the tradeoffs between them, one has to be able to try alternative camera placements. Recalibrating the changed geometry of the cameras is not necessarily restricted only to the reestimation of the extrinsic parameters (position and orientation), as different distances to the observed scene might require changed zoomings, which modifies the intrinsic parameters (focal length, distortion coefficients, etc.). Another source of need for a new calibration can be the change of the camera(s) itself: in desktop oriented HCI webcams that are ubiquitous to the user are preferred to extra instrumentation, so it should be easy to experiment with different cameras.

Given these requirements, one faces the problem of frequent recalibration. Thus, we were interested in calibration methods that are robust and can be carried out with minimal human intervention and preferably in a small amount of time. In order to track the user’s movements we have to simultaneously grab from several cameras, so synchronized imaging during calibration does not pose a problem for us. This is not necessarily the case in other camera setups, for example one-shot digital camera networks.



(a) (b)

Fig. 2. Two hand-tracking systems developed at our department, cameras are marked with red circles. The first one (a) utilizes skin-color detection to segment the users hand, allowing for a relatively uncontrolled background and an orthogonal setup of the cameras. The second (b) uses a background subtraction for hand-segmentation in near infrared illumination. It possible to operate in darkened environments (often needed for 3D displays), but requires a homogenous dark background and the segmentation algorithm itself restricts the placement of the cameras above the tracked area.

3 Calibration Method

It was obvious that we cannot use calibration rigs similar to the one depicted in Figure. 1(a). Manufacturing of L-shaped rigs with exact angles is tedious and it is not possible to change the size of such objects.

The development in auto-calibration methods [1] makes the question valid: do we need a calibration object at all? Auto-calibration requires only a number of correspondences between the images. Utilizing RANSAC methods outliers are also tolerable. If the cameras have a near baseline setup and the background (at least during calibration) is static and has enough features, obtaining the required correspondences is a fairly straightforward task. The wide baseline matching problem is much harder and in the case of a homogenous observed scene (see e.g. Figure 2(b)) getting the matches automatically is impossible.

A very elegant way to generate the needed correspondences using a simple laser pointer as a minimalistic ”calibration object” was reported in [3] and a similar method is utilized by the VICON motion capture system for calibration [4], albeit with the use of highly reflective spheres instead of a laser pointer. A drawback of auto-calibration methods is that at least 3 cameras are needed for them to work. Even in this case all the 3 cameras must share the same intrinsic parameters, which clearly does not hold if different kinds of cameras are used.

To solve the calibration problem in general, at least 8 cameras are required. Another, smaller drawback of these methods is that collecting of the calibration points assumes at least some amount of control of the environment (e.g. to be able to darken the area), which is not always feasible (outdoor environments or exhibition booths).

Homography based calibration methods like [2],[5] utilize planar patterns to identify calibration points in the scene. As a consequnce, they offer a tradeoff between point-like reference features and 3D calibration rigs. Using a planar object has several advantages:

- even one camera can be fully calibrated
- the extrinsic parameters can be estimated in the pattern local coordinate system
- it is easy to modify the size of a pattern by printing one out and attaching it to planar surface
- the background behind the object is occluded, so the role played by the environment is reduced

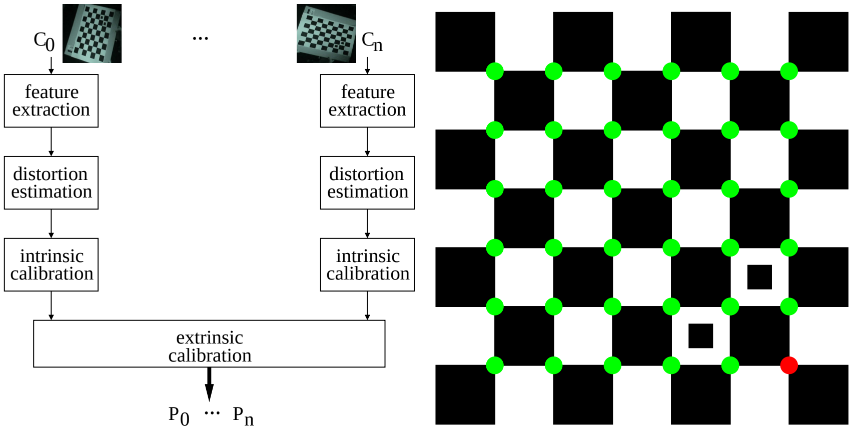
3.1 Algorithm Overview

The main steps of our algorithm are depicted in Figure 3(a). Similarly to [3] and [2], we make the assumption that we are able to grab from all of the cameras while we acquire calibration data. During data acquisition the user has to show a planar pattern (see Figure 3(b)) for the cameras; the pattern, however, does not have to be seen by all the cameras simultaneously. We put some restrictions on the dataset in order to be able to calibrate, these, however, are not limiting in practice for a large class of multicamera systems (see section 5).

In first stage of the algorithm we calibrate for the lens distortion parameters. After they are identified, the dataset is undistorted to be fit for the calibration of the linear pinhole model. The second stage is self-calibration, identifying intrinsic parameters for every camera independently. During the last stage, we compute the extrinsic calibration of the cameras. We explicitly take triangulation error into account, thus making the calibration results well suited for shape from silhouettes methods. Although the need for a planar object makes our method slightly more inconvenient for the user than [3], it allows us to decouple the full pinhole calibration into 3 subproblems with smaller dimensionality. This also has the benefit that the first 2 stages along with reference point extraction from calibration images can be parallelized.

3.2 Correcting Radial Distorion

In order to decouple the distortion estimation from some kind of structure estimation, like usually done during bundle adjustment, we use only the information that points in the calibration pattern should line on a line. We exploit the horizontal, vertical and 45^o lines of the pattern. From the possible suitable methods



(a) (b)

Fig. 3. (a) algorithm overview, (b) the calibration pattern: the green dots are the reference points; the two extra black squares in the lower right corner are used to identify the origin (red dot). A pattern like this (apart from finding the origin) can be easily detected by the Intel OpenCV library [6].

like [8], [9], [7], we choose to implement one of variants without homography estimation from [7]. The distortion in modeled in the usual way:

$$x_d = x_u(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4)$$

$$y_d = y_u(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4)$$

$$r_u = \|x_u - u, y_u - v\|$$

where x_d, y_d are the measurable image points, x_u, y_u are the linearly projected points, κ_1, κ_2 are the distortion coefficients and (u, v) is the distortion center. The parameters to estimate are $[u, v, \kappa_1, \kappa_2]$. Please note that (u, v) is not the principal point, as it is not known at this stage, but an independently determined distortion center.

The method estimates the true distortion of lenses, not a compensating distortion as it is usually done. Although computationally a bit more involved (there is no analitical inverse of the radial distortion function), it has the advantage that the undistorted images do not lose pixel information. A sample distorted and undistorted image made by a wide angle lense is shown in Figure 4.

4 Internal Calibration

The goal of the internal calibration is to estimate the parameters of the intrinsic matrix of the camera:

$$K = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

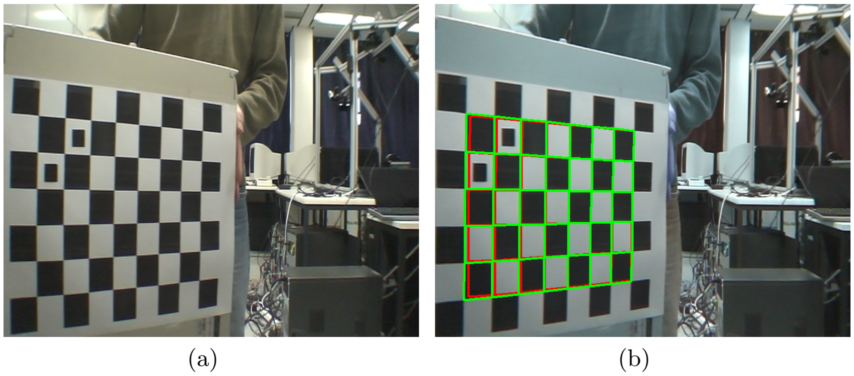


Fig. 4. Original (a) and undistorted image of made by a wide angle lense. Please note than the barrel distortion of the lense is estimated instead of a compensating pincushion distortion.

where α and β are the scaling factors of along the image main axes and (u_0, v_0) denotes the principal point. The method presented in [10] is used for self calibration. To carry out this algorithm, the fundamental matrices corresponding to the image pairs are needed, they were computed using the normalized 8-point algorithm [11]. Similarly to the radial distortion case, we tried to estimate only the parameters in question.

After computing the external parameters and reprojecting the points, however, we found that the reprojection error was too high. This indicated that the computed K matrix had to be refined. In order to get a better estimate of the intrinsic matrix, we iteratively reestimated its 4 parameters by minimizing the reprojection error, computing the also the external paramters in each iteration step. This indeed resulted in a great improvement, as can be seen in figure 5.

5 External Calibration

Planar objects has one disadvatage that point like features do not have: in wide baseline case, the planar pattern is most probably not visible in all cameras, while a point does not cause occlusions. The could be partially circumvented by using both sides of the plane, but these leads to manufacturing problems: the pattern on the two sides must have a known alignment with respect to each other. Fortunately, this problem can be solved by propagating the postion of the world points through the already computed camera calibrations.

Let G be the graph which describes the connectedness of the the cameras: two nodes representing cameras has an edge between them if they have at least one mutual view of the calibration object. As long as this graph is connected, the extrinsic parameters of the cameras can be computed in a common frame. The idea is depicted in 6.

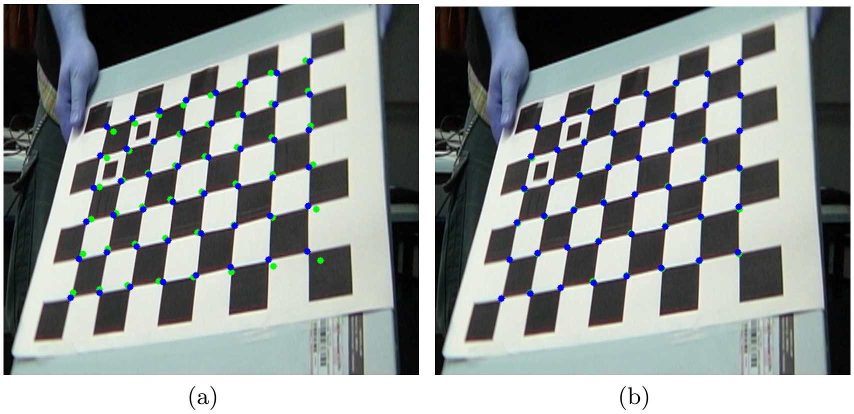


Fig. 5. Reprojection error before (a) and after (b) the reestimation of the intrinsic parameters. The blue dots are the undistorted calibration points, the green ones are the reprojected ones.

Another, more subtle problem connected to planar patterns and extrinsic calibration is depicted in Figure 7(a). Even if the planar pattern is visible by all the cameras in question, but occupies only a smaller portion of the image, its use for estimating the external parameters does take into account the rays which pass through the pattern and introduces triangulation errors along the non controlled rays. One solution would be to use a larger planar object, but this would make the use of the object inconvenient if not prohibitive.

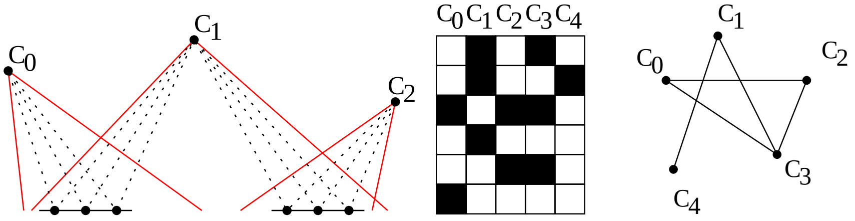
As we collect a large set of images of the planar pattern, we propose to alleviate this problem by using these extra images (not necessarily observed by all the cameras) and taking into account triangulation error of the reference points. As their position is unknown only relative information can be used (distances and angles in the plane) and the fact that the points must lie on a plane, see Figure 7(b).

6 Conclusions

We have presented an easy-to-use method for the complete calibration of a multi-camera network with minimal user interaction. Our method is able to automatically generate full calibration for all cameras from sparse information and does not require any special hardware.

The advantages of our method can be summarised as follows: enhanced stability and flexibility due to the independent calibration steps, full calibration for two up to any number of cameras and due to the use of a planar pattern, indifference to environmental conditions.

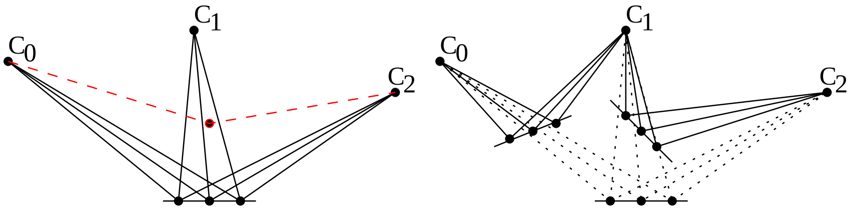
The implementation of the presented method along with its source code is available under the GPL license from our website (‘http://cg.cs.uni-bonn.de/project-pages/camcal/’) along with the most current version of this paper.



C0	C1	C2	C3	C4
□	■	□	■	□
□	■	□	□	■
■	□	■	■	□
□	■	□	□	□
□	□	■	■	□
■	□	□	□	□

(a) (b)

Fig. 6. The position of the reference points can be propagated from C_0 to C_2 through C_1 (a) (red indicates the view frustum of the cameras). In a larger camera network, this propagation works for all the cameras as long as graph G is connected. The concept of G is illustrated in (b), the rows of the matrix on the left represent the synchronously captured frames, the filled squares mean that the calibration pattern was found in the image of the camera.



(a) (b)

Fig. 7. The problem of estimating extrinsic parameters from a planar pattern, which does occupy only a part of the view-frustum of the cameras (a), proposed soloution (b): the known relative position of the planar points can be used to control the otherwise unoptimized viewing rays.

7 Acknowledgements

This research is partially supported by the COHERENT project (EU-FP6-510166), funded under the European FP6/IST program.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in computer vision (2001)
2. Baker, P., Aloimonos, Y.: Calibration of a Multicamera Network. IEEE Workshop on Omnidirectional Vision (2003)
3. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments, 14(4), (2005).
4. URL: www.vicon.com
5. Zhang, Z.: A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence (2000)
6. http://www.intel.com/technology/computing/opencv/

7. Pajdla, T. and Werner, T. and Hlavac, V.: Correcting Radial Lens Distortion without Knowledge of 3-D Structure Technical report TR (1997)
8. Brown, D. C.: Close-range camera calibration. Photogrammetric Engineering (37) (1971).
9. Devernay, F. and Faugeras, O. D.: Straight lines have to be straight. Machine Vision and Applications (2001).
10. Mendonca, P. and Cipolla, R.: A simple technique for self-calibration (1999).
11. Hartley, R.: In Defense of the Eight-Point Algorithm. IEEE Trans. Pattern Anal. Mach. Intell. (1997)