Automating Multi-Camera Self-Calibration

Kai Ide, Steffen Siering, Thomas Sikora
Communication Systems Group
Technische Universität Berlin
{ide,sikora}@nue.tu-berlin.de

Abstract

We demonstrate a convenient and accurate method for fully automatic camera calibration. The method needs at least two cameras and one projector to function, but the cameras need not to be synchronized. By projecting a predefined black and white sequence into the cameras' field of view a large number of individual points are tagged by a binary bit sequence over time. This solves the correspondence problem among the adjacent views and furthermore allows for Forward Error Correction (FEC) yielding a dense error free and subpixel accurate point cloud which is used for internal and external camera calibration. Experimental results and comparison with varying permutations of the projection sequence are given at the end of this paper. Finally, the method gives instant feedback to the user as the resulting calibration point cloud is in fact a 3D scan of the arbitrary calibration scene which can be easily visualized.

1. Introduction

Camera Calibration is a permanent and time consuming obstacle in computer vision. In most cases camera calibration simply distracts from the actual task that a researcher intends to focus on when working with his or her camera setup. For this reason camera calibration should ideally be a short and simple one click process.

The main contribution of this paper is the extension of existing calibration frameworks, which yields an algorithm for the fully automatic internal and external self-calibration of multiple cameras and projectors.

Camera calibration, also known as camera resectioning, is the process of finding a set of parameters in the form of a 3×4 matrix $\mathbf{P}$, that precisely map 3D points in a homogeneous world coordinate system $[x_w, y_w, z_w, 1]^T$ in front of the camera onto the image plane in homogeneous pixel coordinates $[u, v, 1]^T$. We can decompose the 3×4 camera matrix into internal and external parameters, so that the mapping can be written as

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{1}$$

given the scaling factor s, mapping between euclidean and metric space. The matrix $\mathbf{K}$ contains the intrinsic camera parameters such as focal length, principal point, and pixel aspect ratio. Matrix $\mathbf{R}$ and vector $\mathbf{t}$ denote the camera's external parameters and define the camera's position and its orientation in a world coordinate system.

In the absence of camera parameters existing computer vision techniques are already quite powerful. We can perform face recognition, objection segmentation or object tracking, compute disparity maps in a stereo setup, and much more. However, once a multi-camera system is available, the presence of precise camera parameters opens the door to a myriad of new possibilities. Knowing each camera's intrinsics and extrinsics, disparity maps can be converted to depth maps. Since we know the precise location and orientation of each camera, these depth maps can furthermore be merged into a 3D model of the scene [3]. Once a certain amount of the scene geometry is known classic image processing algorithms can be fed with more input resulting in more accurate face recognition, object classification or object segmentation. Given a 3D model of a scene in a world coordinate system we can accurately measure distances between any two points or augment the model with computer graphics which for instance allows for scene relighting.

Multi-camera systems are comprised of at least two cameras. Additionally, we need one projector which can be automatically calibrated along with the cameras. The method presented here places no upper limit to the amount of cameras and projectors to be calibrated simultaneously.

1.1. Related Work

There exist a variety of camera calibration methods such as the Direct Linear Transformation (DLT) [1], Tsai's approach [10], and the Zhang method [12] of which the lat-

ter two have remained the dominant algorithms used in the field. A good introduction into the field of camera calibration is given in [2]. In practice all of the above methods depend on some sort of calibration object which is in most cases a planar checkerboard pattern. To establish linearly independent correspondences among different camera views, several pictures of the calibration object under different orientations have to be taken for each camera pair. Knowing the number of black and white squares in horizontal and vertical direction as well as their exact size projection matrices for all cameras can then be computed automatically. Calibrating projectors is done similarly by precisely matching the checkerboard pattern with another checkerboard pattern that is projected by the projector. Thus, the projector can be calibrated by essentially treating it as a camera itself, such as in [11].

A different approach for establishing point correspondences has been presented by Svoboda *et al*. in [9]. They use a set of $N \geq 3$ synchronized cameras and a modified laser pointer to record an image sequence where the bright dot originating from the tip of the laser pointer can be seen in many views simultaneously. This results in a correspondence set when moving the laser pointer within the working volume over time. A modified approach is applied in [6] where two markers are attached to a stick of a predefined length. However, both processes still require manual labor and the ability to freely move around within the working volume, which may not always be the case. Additionally synchronizing cameras requires additional hardware which is usually expensive or simply not available.

2. The Working Principle

We propose a new technique that accurately identifies points in all of the cameras used – independent of their orientation towards one another. In order to overcome the problem of having to manually present some sort of calibration object to the cameras we simply *project* the calibration object into the scene with an off-the-shelf projector. When projecting a pre-defined black and white sequence into the working volume a large number of individual points are tagged by a binary bit sequence over time. Given a projector resolution of $M \times N$, we need at least $k = \text{ceil}\left(\log_2\left(M \cdot N\right)\right)$ bits to uniquely code each pixel in the projector matrix. A projector with a resolution of 1920×1080 can therefore code over 2 million points with a sequence length of not more than $k = 21$ frames. In practice two additional frames, one all white and one all black are used as threshold indicators. This technique is closely related to 3D measuring with structured light, a good overview of which is given in [8]. However, to our knowledge, structured light projection sequences have never before been used for automatic camera calibration.

The algorithm is comprised of eight functional blocks,

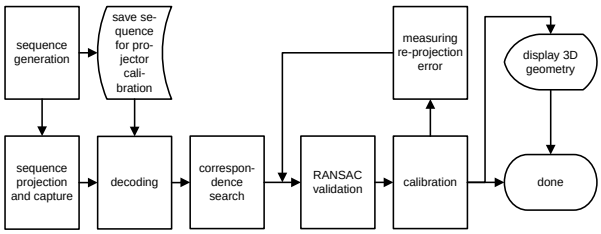


Figure 1. Schematic block diagram illustrating the functional overview of the presented system.

which are illustrated in figure 1. Sequence generation, sequence projection and its capture. Each camera's recorded image sequence is then decoded, yielding a bit pattern for every point in the camera's field of view. Next, point correspondences between all cameras - using the points' bit patterns - are found and verified by a fundamental matrix based RANSAC [4]. Once the correspondence set has accurately been established, the camera setup plus, if desired, the projectors can be calibrated utilizing a linear model and a non-linear model, allowing to estimate and compensate for radial lens distortion. The resulting calibration is iteratively refined until a certain accuracy is reached in terms of the mean re-projection error and its standard deviation. Finally, visualization of the scene geometry along with the position of all cameras and projectors provides a subjective quality measure.

3. Hardware Setup

The testing environment consists of a setup comprised of four Basler Scout 1.3 Megapixel GiGE Vision cameras with a resolution of 1294×964 pixel and an Epson TW3000 LCD projector with a native resolution of 1920×1080 pixel. In practice any projector and any number of $N \geq 2$ cameras can be used. The actual setup can be seen in fig. 2. The cameras need not to be synchronized as pattern projection for each of the k bits and pattern capture are performed consecutively.

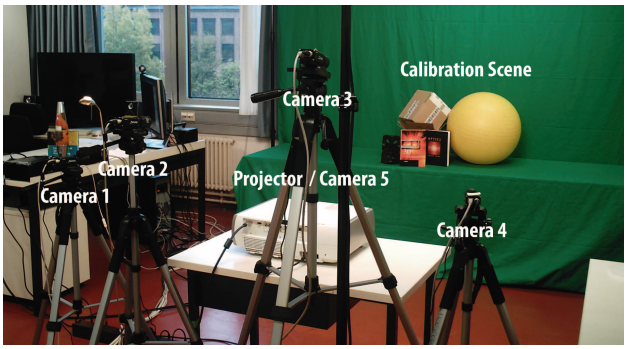


Figure 2. The image shows the hardware setup used for calibration, consisting of four cameras and one projector.

No predefined calibration object is needed. It is, however, important that the projected calibration pattern is able to cover as much of each camera's field of view as possible, resulting in an evenly spread point cloud and a more accurate calibration.

4. Gray Code Calibration Sequence

Ideally, as mentioned before, each image pixel in each camera receives a unique self-identifying bit sequence serving as an identifier. After thresholding the recorded calibration sequence we can decode the pixel identifying image $\mathbf{I}$ by multiplying the binary image set $\mathbf{IB}_{1,\dots,k}$ with different powers of 2, which can be written as:

$$\mathbf{I} = \sum_{i=1}^{k} \mathbf{IB}_i \cdot 2^{i-1} \tag{2}$$

High frequency black and white shifts in the pattern can easily cause aliasing errors in the camera views. So, in practice it is beneficial to code the sequence as a two-dimensional Gray Code [5] where higher frequency components resemble bits of lower significance. A thresholded sequence of length $k = 12$ is shown in fig. 3.

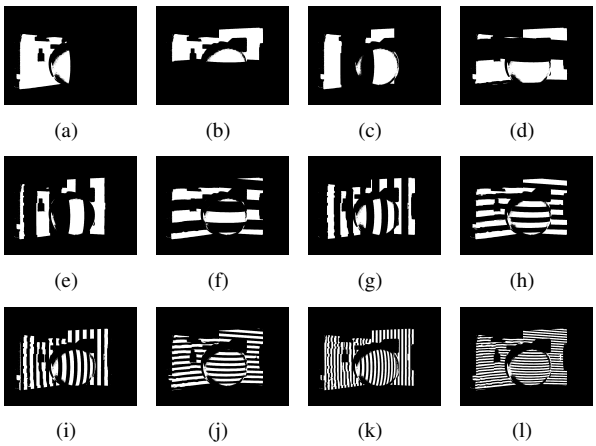


Figure 3. Captured and thresholded two-dimensional Gray Code. We notice that the images become aliased towards the least significant bit (LSB), making that particular bit useless for calibration.

Decoding the captured Gray Code calibration sequence in fig. 3 gives each pixel in each camera one of around two million unique identifiers. We visualize identifiers with a low number as blue, proceeding through the spectrum with increasing ID, passing green, yellow and orange, until a maximum number in the dark red domain is reached. Due to limitations in the perception of that many colors small individual regions with identical identifiers cannot be seen in the illustration given in fig. 4 but the overall concept should become clear. Notice the faceting of the colored regions resulting in a very fine identifier grid in each camera view.

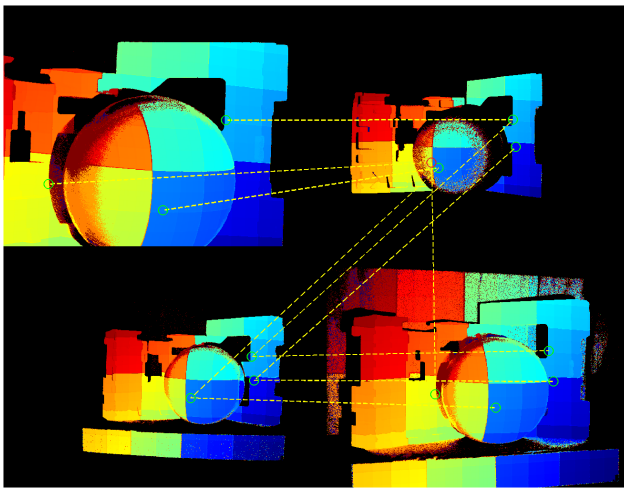


Figure 4. The image shows the decoded calibration sequence for four different camera views. Despite the large number of correct correspondences (three examples shown in green), without any kind of FEC scheme noise in the identifying image can lead to mismatches (one example shown in red). These mismatches are generally caused by bit errors towards the codeword's LSB.

Problems occur at code boundaries where the code is often inaccurately decoded. Additionally, if b bits of lower significance cannot be used the identifying grid essentially increases by a factor of $f = 2^b$. This makes calibration less accurate and can, if bits of higher order get inaccurately identified, lead to completely wrong correspondences.

The aspect of wrong point correspondences can to some extend be neglected as an erroneously detected Gray Code sequence results in relatively large errors, which can be easily detected by a later RANSAC runthrough but the aspect of losing b bits due to aliasing comes at a higher cost as the size of regions with a unique identifier increases. This makes the estimation of a given identifier's centroid difficult. A solution to this problem may be given in the form of utilizing a forward error correcting (FEC) scheme such as a BHC-Code or a Reed-Solomon Code which are both well described in [7]. Considering that roughly two million point correspondences are much more than enough for a later calibration step and can furthermore be problematic as they require a considerable amount of memory to be processed, we conclude that a modified calibration sequence with less but highly accurate points might be better.

5. The Modified Calibration Sequence

We postulated that less but highly accurate point correspondences give a significantly lower reprojection error in average than a large number of noisy correspondences. We therefore modify the calibration pattern to project dots of relatively low density. Each dot is circular in shape and has

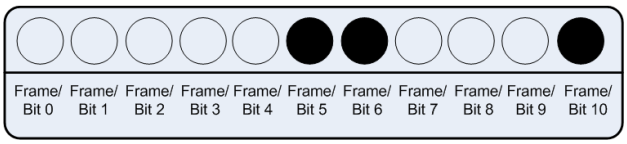


Figure 5. To illustrate the code we show the example for a dot with the identifier $\#999$. This identifier is converted to a binary representation which yields the sequence $1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0$. As the actual sequence consisted of $513 \leq j \leq 1024$ dots, 10 frames are needed to code this particular dot uniquely.

a radius r of only a few pixels, so that its center can be estimated with subpixel precision. The j dots again receive a unique ID $= 1, \ldots, j$ which is converted to a binary sequence, used to turn the dot on and off on a frame by frame basis. This is exemplified for the identifier $\#999$ in fig. 5.

6. Decoding the Calibration Sequence

Decoding the recorded modified calibration sequence $\mathbf{IB}$ is in accordance with the formulation in eq. 2. The problems of the standard Gray Code sequence which occur especially around identifier boundaries are overcome entirely. As the approximate shape of each dot is known in advance each dot's center can be estimated with a high degree of certainty. An example of the decoded identifier image $\mathbf{I}$ using a pattern with $M \times N$ dots is given in figure 6. As will be shown later, this approach yields far better results which is in agreement with the initial lemma of the previous section.

The first frame of the sequence contains an image with all dots turn on followed by an image where the projector is not projecting anything. Subtracting these two from one another we receive difference image $\mathbf{D}$ which serves as a threshold indicator. Due to varying reflectivity within the scene global thresholding is not an option because white dots projected on dark material are likely to have a lower intensity than dark regions on a highly reflective surface. We therefore subdivide $\mathbf{D}$ into a number of relatively large macroblocks $\mathbf{MB}$ and perform individual thresholding depending on the noise level in each block. As only noise can create negative values in $\mathbf{D}$, the absolute value of the lowest element in each $\mathbf{MB}$ serves as the threshold for that particular block. The resulting binary mask then undergoes a final erosion step to shrink dots in the mask by one pixel in radius. This compensates for noise at the boundaries and makes the detection of identifiers very robust.

7. Projector Calibration

The calibration sequence that is projected onto the scene can be interpreted as a binary image set $\mathbf{IB}_{1,\ldots,k}$ itself. Treating this set as an image sequence *captured* by the projector fully self-calibrates the projector along with the cameras. Thresholding and the computation of the individual dot's centroids needs, however, not to be performed as their precise locations are known beforehand. Notice that unlike previous projector calibration methods, where a calibration pattern print out is semi-automatically of even manually matched be the projector, the procedure presented here is very convenient as again no user interaction whatsoever is necessary. If $N \geq 3$ cameras are used projectors can also be excluded from the global calibration loop if this should become necessary due to the minimization of calibration time. Additionally there is neither an upper limit on the amount of projectors nor is there downside of increasing the number of projectors which are to be calibrated this way. On the contrary, each additional projector contributes more dots from different angles into the scene, potentially making the resulting calibration more robust.

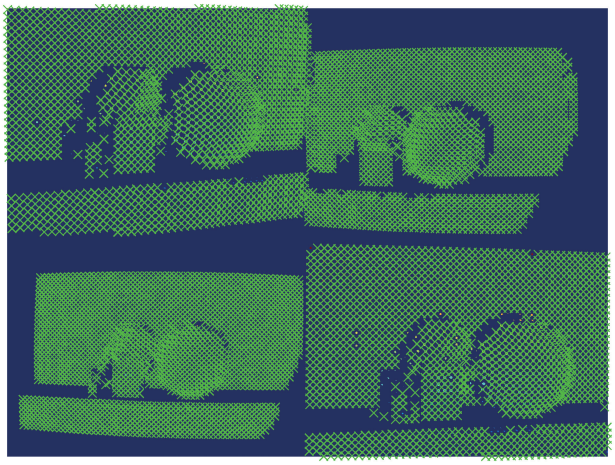


Figure 6. This is the decoded modified calibration sequence $\mathbf{I}$. Each dot has been uniquely identified in all four camera views. The center of each dot can be estimated by averaging over all pixels with the same identifier, thus automatically establishing a correspondence set with sub-pixel precision.

8. Results

In this section we will discuss the quality of the proposed calibration method with respect to different permutations of the calibration sequence. Fig. 7 illustrates the external parameter set for $N = 4$ cameras and an additional projector. The green camera pyramids point towards the captured scene. Their opening angle and their length serve as indicators for both their field of view and their focal length, respectively. The individual uv-axes, denoting origins of the individual image planes, are indicated in blue. The same goes for the projector pyramid, which is for clarification shown in blue. The 3D position of all dot centroids having survived the RANSAC verification are marked in red and give valuable feedback to the person doing the calibration

as they essentially represent of low resolution 3D scan of the captured scene. Notice that presently calibration lacks both a global size factor as well as the origin of a predefined world coordinate system. The later can be compensated for by simply defining the first camera's center as the center of that world coordinate system. The other aspect is something to be worked on in the future.

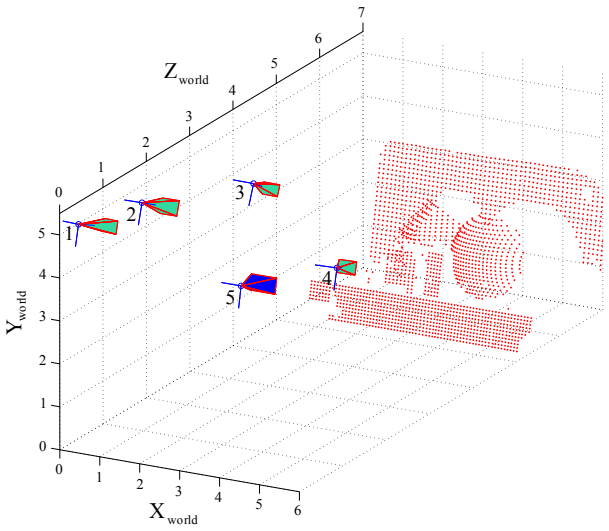
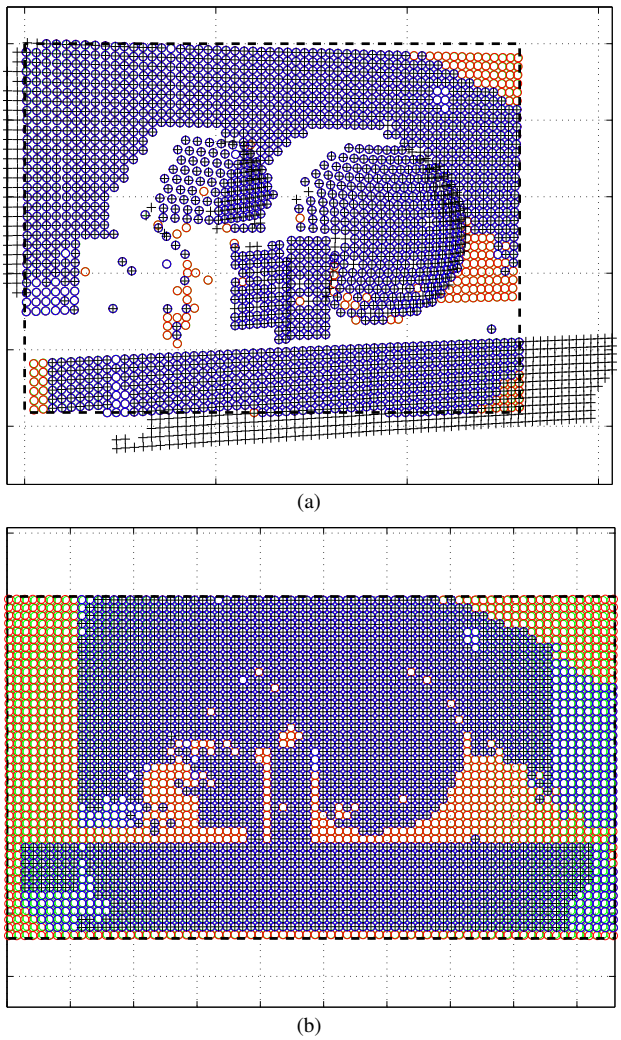


Figure 7. The image above illustrates the external camera- [1 to 4] (green) and projector [5] (blue) parameters with respect to the point cloud (red) in front of the camera setup.

8.1. Pixel Reprojection Error

The pixel reprojection error serves as a quantitative measure of camera calibration quality. Given the triangulated 3D point cloud, each point that has survived RANSAC verification is reprojected into each camera, using the calculated projection matrix. This is shown for camera 1 in fig. 8a and the projector in fig. 8b. Points that were recorded in that particular camera are marked as circles, their color being blue if the point was also detected in at least two other camera or projector views. If the point has no counterparts within the correspondence set it is shown as a red circle. Reprojected points are marked with a + and should ideally be located in the exact center of their corresponding circle. The difference between reprojected location and the actually measured location during the decoding step given in $\mathbf{I}$ is the reprojection error. In practice the reprojection error is usually larger at image boundaries as radial distortions due to the optical system used in the cameras become more apparent. Somewhat surprisingly this is also true for the optics of the projector even though its high quality object lens should show no signs of radial distortion when projecting rectangular images onto a screen.

Notice the variable image sizes in fig. 8. As mentioned before, camera resolution was 1294×964 pixel whereas projector resolution was 1920×1080 pixels as the method is flexible in terms of camera type and resolution.



(a)

(b)

Figure 8. Reprojection error within the image plane of camera 1 (a) and the projector (b). Measured points with counterparts within the correspondence set are shown in blue. Should they have no equivalent they are shown in red. Notice that this is especially true for the projector as its includes all possible dot IDs which need not necessarily have to be captured be any camera. Reprojected 3D points are marked with a +.

The reprojection error for our setup consisting of four cameras and one projector is shown shown in fig. 9. The total mean reprojection error is $0,29$ pixels with a standard deviation of $0,20$ pixels. It is important to mention that this result, despite being satisfactory, is not necessarily in itself meaningful as one has to compare this value with the distribution of points within the image space that have survived

the RANSAC verification. An even spread throughout the image plane is desirable, a property that has been demonstrated in fig. 8. As we decrease radius and spacing to $r = s = 1$ we get results similar to the unmodified calibration sequence, yielding a relatively large reprojection error of 0.8 pixels in average.

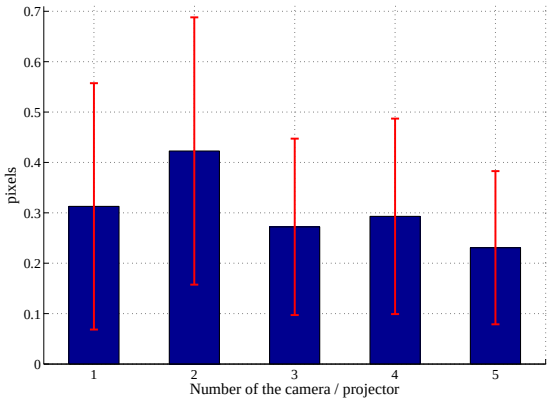


Figure 9. Reprojection error of cameras $[1$ to $4]$ and the projector $[5]$ (b). All points: Mean reprojection error (mean) is $0,29$ pixels, standard deviation (std) is $0,20$.

8.2. Time Consumption

Processing time depends both on the accuracy and number of established point correspondences. One could assume a O(n) dependency on the number of dots surviving the RANSAC iterations but in practice it seems to be somewhat more complicated. A more detailed analysis of the relation between actual camera placing, dot radius, dot clearance, and calibration time will be given in the future. Notice that the slowest computation time is below 5 minutes, whereas the fastest and most accurate calibration results were obtained in only 90 seconds due to rapid convergence of the correspondence set. The implementation contains C++ parts but has mainly been written in Matlab and is running on a Quad Core i7 3.2 GHz CPU with 6 GByte of system memory.

9. Summary

We have shown that fully automatic camera self-calibration is possible by extending existing frameworks with the method presented in this paper. In comparison to manual camera calibration the benefits are straightforward. Automatic camera self-calibration can be done without user interaction. This allows for great flexibility when moving cameras around or when changing the internal parameters such as focal length or the entire lens. Recalibration of a given multi-camera multi-projector system can now be performed in the loop during runtime if necessary. Time savings in camera calibration are significant especially considering the automatic projector calibration. Our results showed a mean reprojection error of under $0,29$ pixels with a standard deviation of $0,20$ pixels under optimal conditions. Visualization of the triangulated calibration point cloud along with all internal and external camera parameters such as shown in fig. 7 gives valuable feedback by instantly indicating plausibility of the calibration results.

10. Acknowledgments

This work has been supported by the Integrated Graduate Program on Human-Centric Communication at Technische Universität Berlin.

References

[1] Y. Abdel-Aziz and H. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range photogrammetry*, pages 1–18, 1971.

[2] T. Clarke and J. Fryer. The development of camera calibration methods and models. *Photogrammetric Record*, 16(91):51–66, 1998.

[3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, New York, NY, USA, 1996. ACM.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. volume 24, pages 381–395, New York, NY, USA, June 1981. ACM Press.

[5] F. Gray. Pulse code communication, March 1953.

[6] G. Kurillo, Z. Li, and R. Bajcsy. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–9, 2008.

[7] W. Peterson and E. Weldon. *Error-correcting codes*. The MIT Press, 1972.

[8] J. Salvi, J. Pags, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004.

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

[10] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of robotics and Automation*, 3(4):323–344, 1987.

[11] S. Zhang and P. S. Huang. Novel method for structured light system calibration. *Optical Engineering*, 45(8):083601, 2006.

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