Simple, Accurate, and Robust Projector-Camera Calibration

Daniel Moreno and Gabriel Taubin
School of Engineering
Brown University
Providence, RI, USA
Email: {daniel_moreno,gabriel_taubin}@brown.edu

***Abstract*—Structured-light systems are simple and effective tools to acquire 3D models. Built with off-the-shelf components, a data projector and a camera, they are easy to deploy and compare in precision with expensive laser scanners. But such a high precision is only possible if camera and projector are both accurately calibrated. Robust calibration methods are well established for cameras but, while cameras and projectors can both be described with the same mathematical model, it is not clear how to adapt these methods to projectors. In consequence, many of the proposed projector calibration techniques make use of a simplified model, neglecting lens distortion, resulting in loss of precision. In this paper, we present a novel method to estimate the image coordinates of 3D points in the projector image plane. The method relies on an uncalibrated camera and makes use of local homographies to reach sub-pixel precision. As a result, any camera model can be used to describe the projector, including the extended pinhole model with radial and tangential distortion coefficients, or even those with more complex lens distortion models.**

***Keywords*-structured-light; camera; projector; calibration; local homography;**

I. Introduction

Structured-light systems are the preferred choice for do-it-yourself 3D scanning applications. They are easy to deploy, only an off-the-shelf data projector and camera are required, and they are very accurate when implemented carefully. A projector-camera pair works as a stereo system, with the advantage that a properly chosen projected pattern simplifies the task of finding point correspondences. In such systems, projectors are modeled as inverse cameras and all considerations known for passive stereo systems may be applied with almost no change. However, the calibration procedure must be adapted to the fact that projectors cannot directly measure the pixel coordinates of 3D points projected onto the projector image plane as cameras do.

Viewpoint, zoom, focus, and other parameters ought to be adjusted, both in projector and camera, to match each target object size and scanning distance; invalidating any previous calibration. Therefore, structured-light systems must be calibrated before each use in order to guaranteed the best result, turning the calibration procedure simplicity as valuable as its precision. In this paper, we present a new calibration procedure for structured-light systems that is both very easy to perform and highly accurate.

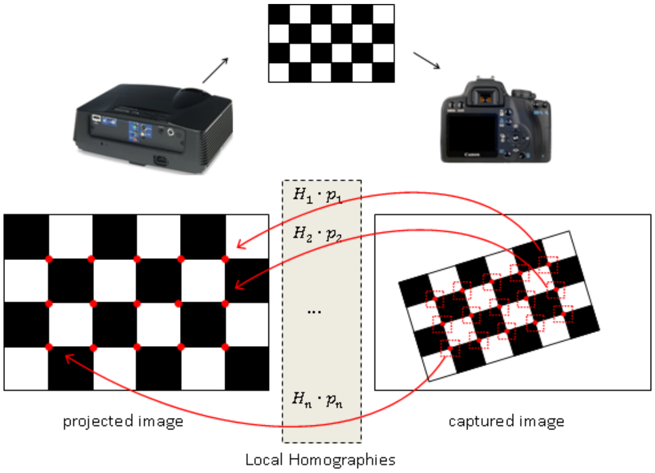

Figure 1. Structured-light system calibration

The key idea of our method is to estimate the coordinates of the calibration points in the projector image plane using local homographies. First, a dense set of correspondences between projector and camera pixels is found by projecting onto the calibration object identical pattern sequence as the one later projected to scan the target, reusing most of the software components written for the scanning application. Second, the set of correspondences is used to compute a group of local homographies that allow to find the projection of any of the points in the calibration object onto the projector image plane with sub-pixel precision. In the end, the data projector is calibrated as a normal camera.

Our main contribution is a method for finding correspondences between projector pixels and 3D world points. Once those correspondences are known any calibration technique available for passive stereo can be applied directly to the structured-light system. Our method does not rely on the camera calibration parameters to find the set of correspondences. As a result, the projector calibration is not affected in any way by the accuracy of the camera calibration.

We show, as a second contribution, that the proposed calibration method can be implemented in such a way that no user intervention is necessary after data acquisition, making the procedure effective even for unexperienced users. To this purpose, we have made a calibration software package, which we plan to make publicly available for anyone interested in structured-light applications to try. Concisely, our software requires two actions:

1) Project a sequence of gray code patterns onto a static planar checkerboard placed within the working volume. Capture one image for each pattern and store them all in the same directory. Repeat this step for several checkerboard poses until properly cover all the working volume. Use a separate directory for each sequence.
2) Execute the calibration software and select the directory containing all the sequences. Enter the checkerboard dimensions. Click on "Calibrate" button. The software will automatically decode all the sequences, find corner locations, and calibrate both projector and camera. The final calibration will be saved to a file for later use.

A. Related Work

Many projector calibration procedures exist, however, we have not found any satisfying the following two key properties: easy-to-perform for the common user and high precision to enable accurate 3D reconstructions. Several methods ([1], [2], [3], [4], [5], and [6]) use a pre-calibrated camera to find world coordinates in some calibration artifact, which in turn they use to assign projector correspondences. These methods might be simple to perform, but all of them lack of accuracy in the projector parameters due to their dependence on the camera calibration. The inaccuracies are a direct consequence of their approach: even small camera calibration errors could result into large world coordinate errors. Their failure point is to estimate the projector parameters from those, far from accurate, world coordinates decreasing the complete system precision.

A different approach is adopted in [7], [8], and [9] where, neither a calibrated camera, nor a printed pattern is required. Instead, they ask the user to move the projector to several locations so that the calibration pattern—projected onto a fix plane—changes its shape. We argue that moving the projector might be inconvenient, or impossible in general (e.g. a system mounted on a rig). Moreover, these methods are not applicable if a metric reconstruction is mandatory because their result is only up-to-scale.

Other authors have proposed algorithms ([10], [11], [12], and [13]) where a projected pattern is iteratively adjusted until they overlap a printed pattern. The overlap is measured with help of an uncalibrated camera. Since both patterns must be clearly identified, the classic black and white pattern is replaced by color versions of it—a color camera is also mandatory. In practice, switching to color patterns make color calibration unavoidable—printed and camera colors seldom match—imposing an extra requirement to the user. Besides, this calibration scheme demands continuous input from a camera to run, rendering impossible to separate the capture stage from the calibration algorithm, which is a common and useful practice in the field.

A common practice among projector calibration methods ([3], [7], [8], [10], and [12]) is to find one homography transformation between a calibration plane and the projector image plane. Despite the elegance of the concept, being homographies linear operators, they cannot model non-linear distortions as the ones introduced by projector lenses.

In [14], the authors claim to get very accurate results with their method that involves projecting patterns on a "flat aluminum board mounted on a high precision moving mechanism". Our complain is that such a special equipment is not available to the common user, limiting its general applicability. We disregard this method as non-practical.

Finally, Zhang and Huang [15], and others ([7], [16]) employ structured-light patterns similarly to us, however, instead of computing projector point correspondences directly from the images as captured by the camera, they create new synthetic images from the projector's viewpoint and feed them to standard camera calibration tools. The intermediate step of creating synthetic images at the projector's resolution, usually low, might discard important information being undesirable. On the contrary, the method we propose finds projector point correspondences from structured-light patterns directly at the camera resolution. No synthetic projector image is created.

The rest of the paper is organized as follows: Section II explains the calibration method, Section III expands the previous section with implementation details, Section IV discusses the experiments done to verify the precision of the method and presents a comparison with other calibration software, finally Section V concludes our work.

II. Method

Our setup comprises one projector and one camera behaving as a stereo pair. We describe them both using the pinhole model extended with radial and tangential distortion, an advantage over several methods ([3], [5], [6], [7], [8], [9], and [12]), which fail to compensate for distortions in the projected patterns. Moreover, we have seen in our experiments that most projectors have noticeable distortions outside their focus plane, distortions that affects the accuracy of the final 3D models.

We took Zhang's method [17] as inspiration in favor of its simplicity and well-known accuracy. It uses a planar checkerboard as calibration artifact, which is easy-to-make for anyone with access to a printer. In Zhang's camera calibration, the user captures images of a checkerboard of known dimensions at several orientations and the algorithm calculates the camera calibration parameters using the relation between the checkerboard corners in a camera coordinate system and a world coordinate system attached to the checkerboard plane.

A. Projector and camera models

The proposed calibration method allows to choose any parametric model to describe the projector and camera. Our implementation uses the pinhole model extended with radial and tangential distortion for both projector and camera. Let be $X \in \mathbb{R}^3$ a point in a world coordinate system with origin at the camera center, and let $u \in \mathbb{R}^2$ the pixel coordinates of the image of X in the camera plane, then X and u are related by the following equations:

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \end{bmatrix} \tag{1}$$

$$u = K_c \cdot L(\tilde{u}) \tag{2}$$

$$K_c = \begin{bmatrix} f_x & \gamma & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$L(\tilde{u}) = \begin{bmatrix} \tilde{u} \cdot (1 + k_1 r^2 + k_2 r^4) + \Delta_t(\tilde{u}) \\ 1 \end{bmatrix} \tag{4}$$

$$\Delta_t(\tilde{u}) = \begin{bmatrix} 2k_3\tilde{u}_x\tilde{u}_y + k_4(r^2 + 2\tilde{u}_x^2) \\ k_3(r^2 + 2\tilde{u}_y^2) + 2k_4\tilde{u}_x\tilde{u}_y \end{bmatrix} \tag{5}$$

$$r^2 = \tilde{u}_x^2 + \tilde{u}_y^2 \tag{6}$$

where K_c is known as camera intrinsic calibration, k_1 and k_2 as radial distortion coefficients, and k_3 and k_4 as tangential distortion coefficients. Similarly, if R and T are a rotation matrix and a translation vector that encode the pose of the projector's center of projection in the world coordinate system defined above, and let $v \in \mathbb{R}^2$ the pixel coordinates of the image of X in the projector plane, then

$$X' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \cdot X + T, \quad \tilde{v} = \begin{bmatrix} x'/z' \\ y'/z' \end{bmatrix} \tag{7}$$

$$v = K_p \cdot L(\tilde{v}) \tag{8}$$

where the projector is described by its intrinsic calibration K_p, and the pair (R, T) is known as the stereo system extrinsic calibration.

B. Data acquisition

Camera calibration involves collecting images of a planar checkerboard. We have modified this acquisition step to make possible to calibrate both camera and projector. The new data acquisition is: for each plane orientation, instead of capturing only one image, the user must project and capture a complete structured-light pattern sequence. Although any structured-light pattern sequence would work, we have used and recommend gray code sequences (Fig. 2) because they are robust to decoding errors—in a calibration routine avoiding all possible errors usually outweighs execution speed. Someone might argue that capturing many images for each checkerboard pose makes our method complex, but the whole data acquisition task is identical to the standard structured-light scanning task as would be executed later. Furthermore, the only actual requirement for the user is to keep the checkerboard static for a few seconds, time necessary to project and capture a complete sequence.

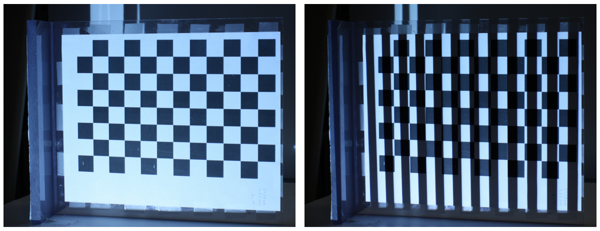

Figure 2. Example of the calibration images: completely illuminated image (left), projected gray code onto the checkerboard (right)

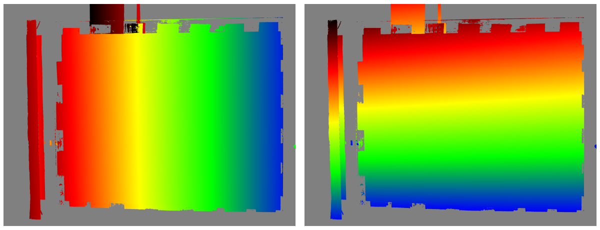

Figure 3. Decoded gray pattern example: pixels with the same color correspond either to the same projector column (left) or same projector row (right). Gray color means "uncertain". Note that there are no uncertain pixels in the checkerboard region.

C. Camera calibration

Intrinsic camera calibration refers to estimating the parameters in the chosen camera model. Following Zhang's method, we need to find the coordinates in the camera image plane of all the checkerboard corners, for each of the captured checkerboard orientations. Corner locations are sought in a completely illuminated image, of each checkerboard orientation, using a standard procedure. A completely illuminated image is an image captured while all data projector pixels are turned on—if no such image is available, it could be created as the maximum of every image in the sequence. The procedure continues as the usual camera calibration, please review [17] for more details.

Our software expects the first image in every gray code sequence to be a completely illuminated image that could be used directly for camera calibration. It uses OpenCV's *findChessboardCorners()* function [18] to automatically find checkerboard corner locations and, then, it refines them to reach sub-pixel precision. Finally, a call to the function *calibrateCamera()* returns the calibrated camera parameters.

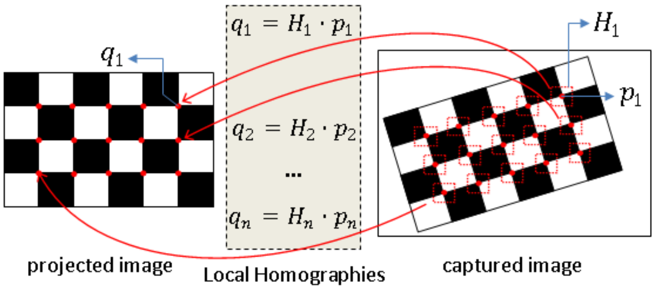

Figure 4. Projector corner locations are estimated with sub-pixel precision using local homographies to each corner in the camera image

D. *Projector calibration*

Our projector and camera are described with the same mathematical model, thus, we would like to follow identical procedure to calibrate them both. But the projector is not a camera. If the projector were a camera, it would be possible to capture images from its viewpoint, to search checkerboard corners in them, and to continue just as before. In reality no such images exist, but we know a relation between projector and camera pixels—extracted from structured-light sequences—and we will show how to use this relation to estimate checkerboard corner locations in projector pixel coordinates. Moreover, being all the computations carried on at the camera's original resolution, corner coordinates are localized with greater precision than if synthetic images at the projector's resolution were used.

The procedure to compute checkerboard corner coordinates in the projector coordinate system can be decompose in three steps: first, the structured-light sequence is decoded and every camera pixel is associated with a projector row and column, or set to "uncertain" (Fig. 3); second, a local homography is estimated for each checkerboard corner in the camera image; third and final, each of the corners is converted (Fig. 4) from camera coordinates to projector coordinates applying the local homography just found.

The structured-light decoding step depends on the projected pattern, in our case complementary gray codes for rows and columns. Here, our method differs from [15] where fringe patterns were proposed—our choice prioritize decoding precision over acquisition speed. As pointed out in [19], a subset of the gray code images—the ones where the stripes look "thin"—may be regarded as exhibiting a high frequency pattern. These high frequency patterns make possible to split the intensity measured at each pixel in a direct and a global component. Ideally, the amount of light perceived at each camera pixel is product of exactly one projector pixel being turned on or off, but in practice this is rarely true. The intensity value reported by the camera at one pixel is the sum of the amount of light emitted by a projector pixel, called direct component, plus some amount of light, known as global component, originated at other sources (including reflections from other projector pixels). Decoding errors in gray sequences are mostly caused by failure on identifying these components, or completely ignoring their existence. On the contrary, if each component is correctly identified, a simple set of rules permits to drastically reduce decoding errors (Fig. 3). The rules and additional information on the topic are given in [20] under the name of robust pixel classification.

The relation learned from structured-light patterns is not bijective—it cannot be used right away to translate from camera to projector coordinates. To overcome this issue we propose the concept of local homography: a homography that is valid only in a region of the plane. Instead of applying a single global homography to translate all the checkerboard corners into projector coordinates, we find one local homography for each of the checkerboard corners. Each local homography is estimated within a small neighborhood of the target corner and is valid only to translate that corner into projector coordinates and no other corner. Local homographies allow to model non-linear distortions because each corner is translated independently of the others. Additionally, they are robust to small decoding errors because they are overdetermined; they are estimated from a neighborhood with more points than the minimum required.

A local homography is found for each checkerboard corner considering all the correctly decoded points in a patch of the camera image centered at the corner location. Let be p the image pixel coordinates of a point in the patch under consideration, and let be q the decoded projector pixel for that point, then we find a homography $\hat{H}$ that minimizes:

$$\hat{H} = \operatorname*{argmin}_{H} \sum_{\forall p} ||q - Hp||^2 \tag{9}$$

$$H \in \mathbb{R}^{3\times 3}, \quad p = [x, y, 1]^T, \quad q = [col, row, 1]^T \tag{10}$$

The target corner $\bar{p}$, located at the center of the patch, is translated to $\bar{q}$, given in projector coordinates, applying the local homography $\hat{H}$:

$$\bar{q} = \hat{H} \cdot \bar{p} \tag{11}$$

The same strategy is repeated until all checkerboard corners have been translated. Now, knowing the location of all corners in the projector coordinate system, the projector intrinsic calibration is found with identical procedure as for the camera.

E. *Stereo system calibration*

Stereo calibration means finding the relative rotation and translation between projector and camera. At this point, the intrinsic parameters found before are kept fixed, the world coordinates are identified with camera coordinates, and we seek for the pose of the projector in world coordinates. The physical dimensions of the calibration checkerboard are known. The checkerboard corner projections onto both camera and projector image planes are also known—they were found in the previous steps. The calibration of the projector-camera stereo system, therefore, is identical to the calibration of any other camera-camera system.

Our software calls OpenCV's *stereoCalibrate()* function with the previously found checkerboard corner coordinates and their projections, the output is a rotation matrix R and a translation vector T relating the projector-camera pair.

F. Algorithm

The complete calibration procedure can be summarized in simple steps and implemented as a calibration algorithm:

1) Detect checkerboard corner locations for each plane orientation in the completely illuminated images.
2) Estimate global and direct light components for each set using gray code high frequency patterns.
3) Decode structured-light patterns into projector row and column correspondences by means of robust pixel classification, considering pixel global and direct components from step 2.
4) Take small image patches centered at the checkerboard corner coordinates from step 1 (e.g. a 47x47 pixels square) and use all the correctly decoded pixels in each patch to compute a local homography that converts from camera to projector coordinates. Correspondences were obtained in step 3.
5) Translate corner locations (step 1) from camera to projector coordinates using patch local homographies from step 4.
6) Fix a world coordinate system to the checkerboard plane and use Zhang's method to find camera intrinsics using camera corner locations from step 1.
7) Fix a world coordinate system to the checkerboard plane and use Zhang's method to find projector intrinsics using projector corner locations from step 5.
8) Fix camera and projector intrinsics (steps 6 and 7) and use world, camera, and projector corner locations (steps 1 and 5) to estimate stereo extrinsic parameters.
9) Optionally, all the parameters, intrinsic and extrinsic, can be bundle-adjusted together to minimize the total reprojection error.

III. Calibration software

We have implemented the algorithm in Section II-F into a complete structured-light system calibration software. The purpose is two-fold: first, to prove that our method can be executed fully automatic provided the calibration images are available; second, to facilitate the access to high quality 3D scans for a broad range of users—we think that structured-light systems are the key. Our experience says that calibrating structured-light systems accurately is a cumbersome and time consuming task. In hopes of ease the task, we have written a software (Fig. 5) with a Graphical User Interface (GUI) capable of calibrate such systems following a simple procedure. The software is completely written in C++, uses Qt [21] as a graphical interface library, and OpenCV [18] library for the vision related tasks. This library selection enables to build and run the software in common platforms such as Microsoft Windows and GNU/Linux.

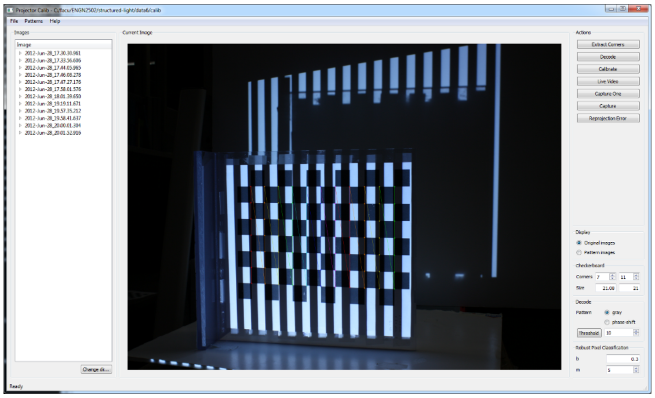

Image
2012-Jun-28_17.30.30.961
2012-Jun-28_17.33.56.606
2012-Jun-28_17.44.05.965
2012-Jun-28_17.46.08.278
2012-Jun-28_17.47.27.176
2012-Jun-28_17.58.01.576
2012-Jun-28_18.01.39.650
2012-Jun-28_19.19.11.671
2012-Jun-28_19.57.35.212
2012-Jun-28_19.58.41.637
2012-Jun-28_20.00.01.304
2012-Jun-28_20.01.52.916

Display
● Original images
○ Pattern images

Checkerboard
Corners 7 11
Size 21.08 21

Decode
Pattern ● gray ○ phase-shift
Threshold 10

Robust Pixel Classification
b 0.3
m 5

Figure 5. Calibration software main screen

Checkerboard corner detection is done with OpenCV's *findChessboardCorners()* function, however, as reported in [22], this function is very slow in combination with high-resolution images. We worked with 12Mpx images and we have observed this issue. Our solution is to downsample the input images in order to accelerate the corner search, and to consider the downsampled corner locations as an approximate solution to the high resolution search. This simple technique has proven to be fast yet effective: search speed is independent of the camera resolution and results as accurate as if no downsampling were performed—because the refinement is executed at the original resolution.

Theoretically, direct and global light components should be estimated from the highest frequency pattern projected. In practice, doing so results in decoded images with visible artifacts. Thus, we skip the highest frequency and we compute the direct and global components from the two second highest patterns. Combining more than one pattern gives better precision and skipping the last pattern removes the artifacts due to limited projector resolution.

Let be $S = \{I_1, \cdots, I_k\}$ the selected set of pattern images, and let be p a valid pixel location, then the direct and global components at p, $L_d(p)$ and $L_g(p)$, are found as follows:

$$L_p^+ = \max_{0<i\leq k} I_i(p), \quad L_p^- = \min_{0<i\leq k} I_i(p), \tag{12}$$

$$L_d(p) = \frac{L_p^+ - L_p^-}{1-b}, \quad L_g(p) = 2\,\frac{L_p^- - b\,L_p^+}{1-b^2}, \tag{13}$$

where $b \in [0,1)$ is a user-set value modeling the amount of light emitted by a turned-off projector pixel—we recommend the reader to study [19] for more details. We have set $b = 0.3$ in our setup.

Finally, local homographies are estimated from fix size image patches; we have, therefore, to select a proper patch size for them. If the chosen size is too small, the algorithm becomes very sensitive to decoding errors. On the contrary, if the patch is too large, the algorithm is robust to errors, but unable to cope with strong lens distortions. Experimentally, we have found a patch size of 47x47 pixels to perform well in our system; we have used this value in all our tests. Nevertheless, a more rigorous analysis is required to decide the optimum size given the system parameters.

IV. Results

We have developed this calibration method to enable high precision 3D scanning. In consequence, we think the best calibration quality evaluation is to scan objects of known geometry and to compare their 3D models with ground truth data. Additionally, we think that an evaluation would not be complete without a comparison with other available calibration methods. We have searched and found that Samuel Audet's ProCamCalib [10] and Projector-Camera Calibration Toolbox (also procamcalib) [1] are publicly available tools. We have tried both, but Audet's tool current version cannot be used with our camera, for that reason, we will compare our method with Projector-Camera Calibration Toolbox [1] only, from now on just "procamcalib".

A. Test setup

Our test setup comprises a Mitsubishi XD300U DLP data projector and a Canon EOS Rebel XSi camera. Projector's resolution is 1024x768 and camera's resolution is 4272x2848. They were placed one next to the other (Fig. 6). Their focus length, zoom, and direction were adjusted prior calibration accordingly to the scan target.

B. Reprojection error

Usually, the quality of camera calibration is evaluated considering only reprojection errors. But, a minimum reprojection error measured in the calibration images does not ensure the best reconstruction accuracy of arbitrary objects. In fact, in our experiments adding an additional minimization step of the intrinsic and extrinsic parameters altogether (Section II-F, step 9) overfitted the calibration data producing slightly less accurate 3D models. All in all, reprojection errors are indicators of calibration accuracy and we report ours as a reference for comparison with other methods. Table I shows the reprojection error of our method and procamcalib; for further comparison, we have also included a modified version of our method which uses one global homography instead of local ones. In result, using identical camera calibration, procamcalib reprojection error is much higher than ours as consequence of its dependency on the camera calibration to find world plane correspondences. The modified method is an improvement over procamcalib, however, given the linearity of its global homography, it fails to model projector lens distortion being suboptimal.

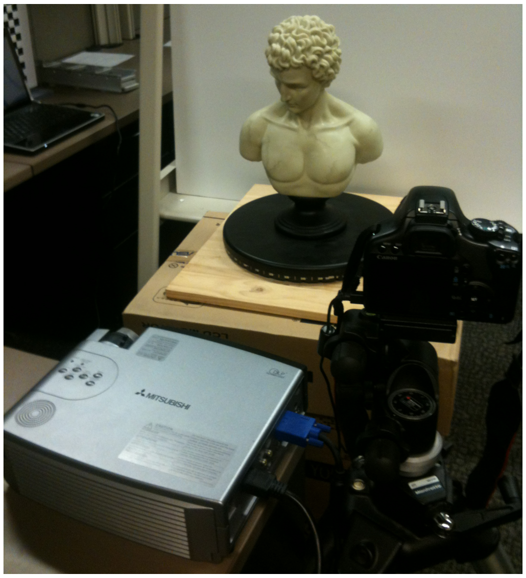

Figure 6. System setup

Method	Camera	Projector
Proposed	0.3288	0.1447
Proposed with global homography		0.2176
procamcalib		0.8671

Table I
Reprojection error

C. Projector lens distortion

One of the main advantages of our method is that it allows to model radial and tangential distortion in projector lenses the same as in cameras. Opposite to what is said in other papers (e.g. [9]), projector lenses have noticeable distortion, specially near the edges. Table II shows an example of the distortion coefficients estimated by our method; note that k_2 has a non-negligible value. The complete distortion model (Fig. 7) shows that points close to the top-left corner are displaced about 12 pixels from their ideal non-distorted coordinates; at the bottom-center of the projected image, where its principal point is located, there is no distortion as expected. In conclusion, data projectors have non-trivial lens distortions that cannot be ignored.

D. Ground truth data

To evaluate the quality of the calibration beyond its reprojection errors, we scanned real objects and created 3D models of them which could be compared with ground truth data. Our first model corresponds to a plane of 200x250 mm,

k_1	k_2	k_3	k_4
-0.0888	0.3365	-0.0126	-0.0023

Table II
PROJECTOR DISTORTION COEFFICIENTS: k_1 AND k_2 RADIAL DISTORTION, k_3 AND k_4 TANGENTIAL DISTORTION

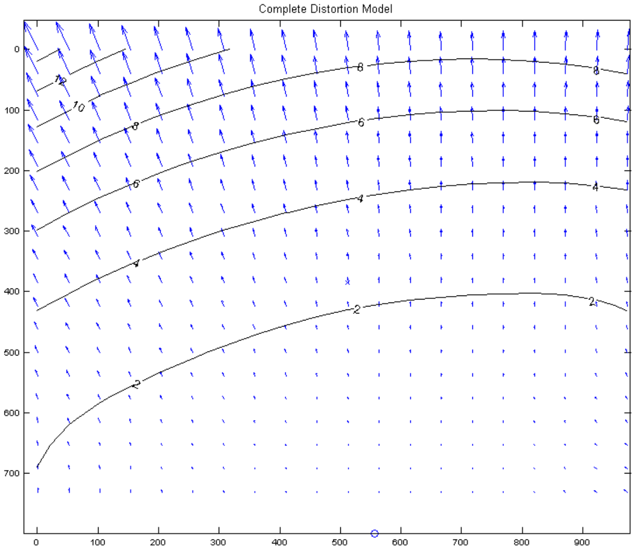

Figure 7. Projector distortion model: points are displaced about 12 pixels near the top-left corner

for which we created one 3D model with each of both calibrations in the previous section. The ground truth data for these models are points sampled from an ideal plane. The error distribution of the model reconstructed with our calibration (Fig. 8 top) resembles a Gaussian distribution where 95% of its samples are errors equal or less than 0.33 mm. On the other hand, the reconstruction made with procamcalib's calibration (Fig. 8 bottom) has an irregular error distribution denoting calibration inaccuracy. The results are summarized in Table III.

Our next model is a statue head scanned both with our structured-light system and with a commercial laser scanner. The laser scanner is a NextEngine Desktop Scanner 2020i. Both 3D models are compared with the Hausdorff distance and the result is shown as an image (Fig. 9). The color scale denotes the distance between both meshes ranging from 0 to 1 mm. The error reaches its maximum only in regions that were in shadow during scanning. In the face region the error is 0.5 mm at most.

Method	Max. Error	Mean Error	Std. Dev.
Proposed	0.8546	0.000042	0.1821
procamcalib	1.6352	0.000105	0.2909

Table III
IDEAL AND RECONSTRUCTED PLANE COMPARISON

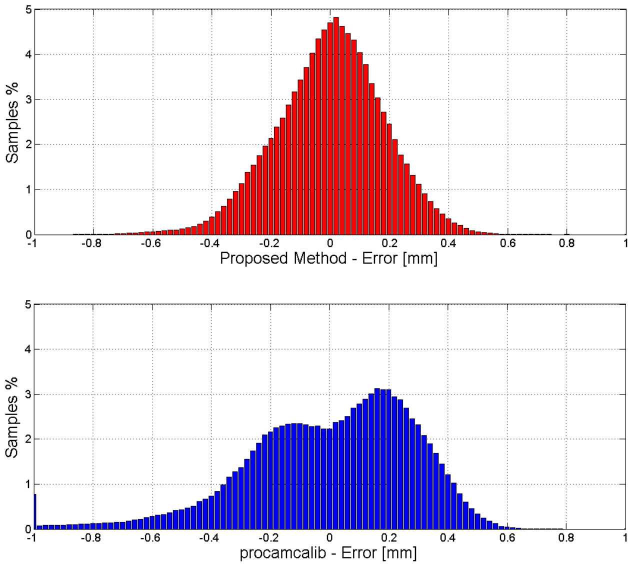

Figure 8. Plane error histogram: error between an ideal plane and a scanned plane recontructed using the proposed calibration (top) and procamcalib calibration (bottom)

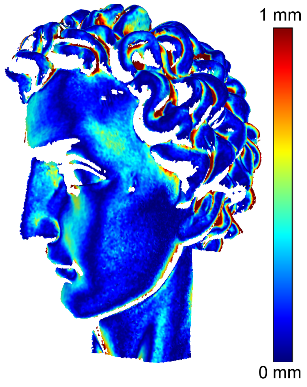

Figure 9. Structured-light versus laser scanner: Hausdorff distance between meshes from 0mm to 1mm

Finally, we scanned a statue from six different viewpoints and, after manual alignment and merging, we created a complete 3D model using Smooth Signed Distance (SSD) surface reconstruction ([23], [24]). The final mesh preserves (Fig. 10) even small details.

V. CONCLUSION

We have introduced a new method to calibrate projector-camera systems that is simple to implement and more accurate than previous methods because it uses a full pinhole model—including radial and tangential lens distortions—to describe both projector and camera behaviors and computes sub-pixel resolution 3D point projections from uncalibrated

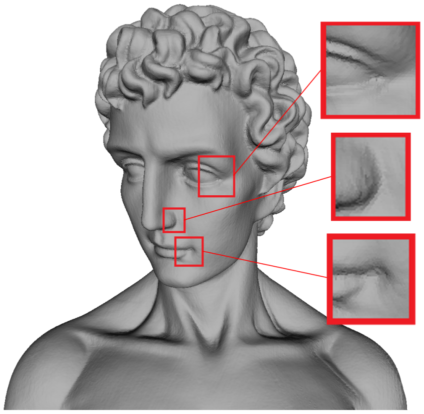

Figure 10. SSD 3D model from 6 structured-light scans

camera images. We have developed a simple-to-use calibration software that we will make freely available for people to experiment with.

Acknowledgment

The authors want to thank Fatih Calakli for proof-reading this paper and the useful suggestions he has made to improve this work. The material presented in this paper describes work supported by the National Science Foundation under Grants No. IIS-0808718, and CCF-0915661.

References

[1] G. Falcao, N. Hurtos, J. Massich, and D. Fofi, "Projector-Camera Calibration Toolbox," Tech. Rep., 2009, available at http://code.google.com/p/procamcalib.

[2] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross, "A practical structured light acquisition system for point-based geometry and texture," in *Point-Based Graphics. Eurographics/IEEE VGTC Symposium Proceedings*, june 2005, pp. 89–145.

[3] M. Kimura, M. Mochimaru, and T. Kanade, "Projector calibration using arbitrary planes and calibrated camera," in *Computer Vision and Pattern Recognition*, 2007, pp. 1–2.

[4] J. Liao and L. Cai, "A calibration method for uncoupling projector and camera of a structured light system," in *Advanced Intelligent Mechatronics*, july 2008, pp. 770–774.

[5] K. Yamauchi, H. Saito, and Y. Sato, "Calibration of a structured light system by observing planar object from unknown viewpoints," in *19th International Conference on Pattern Recognition*, dec. 2008, pp. 1–4.

[6] W. Gao, L. Wang, and Z.-Y. Hu, "Flexible calibration of a portable structured light system through surface plane," *Acta Automatica Sinica*, vol. 34, no. 11, pp. 1358–1362, 2008.

[7] H. Anwar, I. Din, and K. Park, "Projector calibration for 3D scanning using virtual target images," *International Journal of Precision Engineering and Manufacturing*, vol. 13, pp. 125–131, 2012.

[8] J. Draréni, S. Roy, and P. Sturm, "Geometric video projector auto-calibration," in *Computer Vision and Pattern Recognition Workshops*, 2009, pp. 39–46.

[9] J. Draréni, S. Roy, and P. Sturm, "Methods for geometrical video projector calibration," *Machine Vision and Applications*, vol. 23, pp. 79–89, 2012.

[10] S. Audet and M. Okutomi, "A user-friendly method to geometrically calibrate projector-camera systems," in *Computer Vision and Pattern Recognition Workshops*, 2009, pp. 47–54.

[11] I. Martynov, J.-K. Kamarainen, and L. Lensu, "Projector calibration by inverse camera calibration," in *Proceedings of the 17th Scandinavian conference on Image analysis*, Berlin, Heidelberg, 2011, pp. 536–544.

[12] J. Mosnier, F. Berry, and O. Ait-Aider, "A new method for projector calibration based on visual servoing," in *IAPR Conference on Machine Vision Applications*, 2009, pp. 25–29.

[13] S.-Y. Park and G. G. Park, "Active calibration of camera-projector systems based on planar homography," in *International Conference on Pattern Recognition*, 2010, pp. 320–323.

[14] X. Chen, J. Xi, Y. Jin, and J. Sun, "Accurate calibration for a cameraprojector measurement system based on structured light projection," *Optics and Lasers in Engineering*, vol. 47, no. 34, pp. 310–319, 2009.

[15] S. Zhang and P. S. Huang, "Novel method for structured light system calibration," *Optical Engineering*, vol. 45, no. 8, pp. 083 601–083 601–8, 2006.

[16] Z. Li, Y. Shi, C. Wang, and Y. Wang, "Accurate calibration method for a structured light system," *Optical Engineering*, vol. 47, no. 5, p. 053604, 2008.

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

[18] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. [Online]. Available: http://opencv.org/

[19] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, "Fast separation of direct and global components of a scene using high frequency illumination," in *SIGGRAPH*. New York, NY, USA: ACM, 2006, pp. 935–944.

[20] Y. Xu and D. G. Aliaga, "Robust pixel classification for 3d modeling with structured light," in *Proceedings of Graphics Interface*. New York, NY, USA: ACM, 2007, pp. 233–240.

[21] Qt Project, "Qt cross-platform application and UI framework." 2012. [Online]. Available: http://qt.nokia.com/

[22] J. Chen, K. Benzeroual, and R. Allison, "Calibration for high-definition camera rigs with marker chessboard," in *Computer Vision and Pattern Recognition Workshops*, 2012, pp. 29–36.

[23] F. Calakli and G. Taubin, "SSD: Smooth signed distance surface reconstruction," *Computer Graphics Forum*, vol. 30, no. 7, pp. 1993–2002, 2011.

[24] F. Calakli and G. Taubin, "SSD-C: Smooth signed distance colored surface reconstruction," in *Expanding the Frontiers of Visual Analytics and Visualization*, 2012, pp. 323–338.