Plane-based calibration of a projector-camera system

Gabriel Falcao, Natalia Hurtos, Joan Massich

VIBOT Master 2008

Accuracy and simplicity of system calibration is a key challenge in 3D computer vision tasks. A calibration method by a planar checkerboard to minimize calibration complexity and cost is proposed. This method is based in considering the projector as an inverse camera which maps 2D image intensities into 3D rays, thus making the calibration of a projector the same as that of a camera. In this way, having the 2D projected points and its 3D correspondences the system can be calibrated using a standard camera calibration method such as the one implemented in Bouguet's Calibration Toolbox. A projector-camera system has been calibrated by this method, and a good 3D reconstruction quality has been achieved by the calibrated system.

1. Introduction

This report addresses the problem of system calibration which is crucial for any 3D computer vision system and in particular for any 3D shape measurement system. Placed in this context, system calibration involves the calibration of a camera, which have been extensively studied over the years, and the calibration of a projector. Theoretically, a data projector can be seen as a dual of a camera. In practice, there are three main differences that make the calibration of a projector more complicated than that of a camera. The first one is obvious: projectors cannot image the surface that they illuminate so that the correspondence between the 2D projected points and the 3D illuminated points cannot be made without the use of a camera. The second one is that it is difficult to retrieve the co-ordinates of the 3D points because the calibrating pattern is projected and not attached to the world coordinate frame in general. This can explain that, to this date, there have been very few practical methods to compute the intrinsic parameters of a data projector.
In this work, we present a method of plane-based calibration for projector-camera systems which aims to be both accurate and easy-to-use. This method has been implemented as an extension to *Bouguet Camera Calibration Toolbox*.
In section 2 the principle of the proposed method is presented. In sections 3,4,5 the main steps of the method are described. In section 6 a tutorial on the use of the toolbox for calibrating a Camera-Projector system is provided. Finally in section 7 some experimental results using 3D reconstruction with structured light demonstrate the performance of the method and in section 8 the conclusions and future work is discussed.

2. Overview of the method

The goal of the presented method is to obtain the intrinsic and extrinsics parameters for both the camera and projector of the system. Both image capture and projection is generally described by a standard pinhole camera model with intrinsic parameters including focal length, principle point, pixel skew factor, and pixel size; and extrinsic parameters including rotation and translation from a world coordinate system to a camera or projector coordinate system.

The key point of the proposed calibration method is to consider the projector as an inverse camera (mapping intensities of a 2D image into 3D rays) thus making the calibration of a projector the same as that of a camera. In this way we can make use of any standard calibration procedure for cameras in order to calibrate the projector. So the main concern of our method is to find the 3D points of the projected pattern in order to use them together with the 2D points of the image we are projecting to finally obtain the intrinsics and extrinsics of the projector.

The proposed method to achieve the whole calibration of the system is divided in several steps:

- Calibrate the camera using Zhang's method
- Recover calibration plane in camera coordinate system
- Project a checkerboard on calibration board and detect corners
- Apply ray-plane intersection to recover 3D position for each projected corner

- Calibrate the projector using the correspondences between the 2D points of the image that is projected and the 3D projected points.

In the following sections the details of the main steps will be explained. However before going into detail it is convenient to describe the setup that the method requires. This setup is shown in figure 1.

Figure 1 Setup for the camera-projector system calibration

The calibration board is a planar surface with an attached checkerboard pattern and a blank area in which a projected pattern will be displayed. It is important to mention that the focus of the projector must be adapted to the correct distance in order to take the pictures of the calibration board inside the in-focus range.

3. Camera calibration using Zhang's method

Zhang's method [1], using a simple planar pattern has provided the research community with both easy-to-use and accurate algorithm for obtaining both intrinsic and extrinsic camera parameters. This algorithm was implemented in Matlab Camera Calibration Toolbox [2] by Jean-Yves Bouguet and C++ in Intel OpenCV library [3]. These two libraries are probably the most widely used tools for camera calibration nowadays.

Zhang's calibration method requires a planar checkerboard grid to be placed at different orientations (more than 2) in front of the camera. The developed algorithm uses the extracted corner points of the checkerboard pattern to compute a projective transformation between the image points of the *n* different images, up to a scale factor. Afterwards, the camera interior and exterior parameters are recovered using a closed-form solution, while the fourth-order radial distortion terms are recovered within a linear least-squares solution. A final nonlinear minimization of the reprojection error, solved using a Levenberg-Marquardt method, refines all the recovered parameters. For more details about Zhang's algorithm, see [1].

Therefore, in order to achieve our camera calibration we use Bouguet Toolbox and we obtain the intrinsics and extrinsics parameters of the camera related to the world frame placed on the upper-left corner of our calibration plane. Although our method is closely tight to Bouguet's camera calibration procedures, we are not going to provide here a detailed description of the toolbox. For a good description on the functions, parameters and procedures involved in the Bouguet toolbox, see [2].

4. Obtention of the 3D projected points

As it has been already mentioned the key point of our method is to find the 3D points of the projected pattern in order to use them with the correspondences in the 2D projected image and thus calibrate the projector with a standard camera calibration procedure.

To that end, we perform the following steps:

- Compute the equation of the calibration plane in camera coordinate system
- Detect the corners of the projected checkerboard
- Consider the rays that go from the camera trough the corners of the projected pattern
- Apply ray-plane intersection to recover 3D position for each projected corner

4.1 Equation of the calibration plane in camera coordinate system

Once the parameters of the camera are known, it is easy to compute the equation of the calibration plane in camera coordinate system.

In a three-dimensional space, a common way of defining a plane is by specifying a point and a normal vector to the plane. As at this point we are suppose to have the extrinsic camera parameters that relate the optical centre of the camera with the upper-left corner of the calibration plane we already have the ingredients to build the equation of the plane.

The translation vector of the extrinsic parameters (last column) gives actually the coordinates of one point of the plane (the origin, p) and the third column vector of the rotation matrix (n) is a the surface normal vector of the plane containing the planar grid in the camera reference frame.

$$K_{ext} = \begin{array}{cccc} & & \boldsymbol{n} & \boldsymbol{p} \\ \end{array} \begin{bmatrix} r_{11} & r_{12} & \boxed{r_{13}} & \boxed{t_x} \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

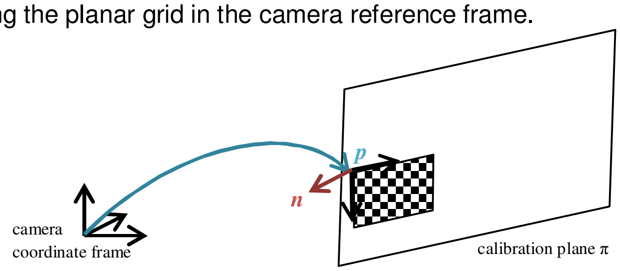


Figure 2 Use of the extrinsic parameters to compute plane equation

Then, being *p* the vector representing the position of a known point in the plane, and *n* a nonzero normal vector to the plane, the desired plane is the set of all points *r* such that :

$$n \cdot (r - p) = 0$$

If we write $n = a\hat{\mathbf{x}} + b\hat{\mathbf{y}} + c\hat{\mathbf{z}}$ $\quad r = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}}$

where $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$ are the Cartesian unit vectors and define *d* as the dot product:

$$d = -n \cdot p$$

then the plane π is determined by the condition

$$ax + by + cz + d = 0$$

where *a*, *b*, *c* and *d* are known real numbers and *a*, *b*, and *c* are not all zero.

4.2 Detection of the corners of the projected checkerboard

Next step consist in extracting the corners of the projected checkerboard pattern (which is projected over the same calibration plane where there is the attached pattern). This is done using the Bouguet functions for extracting the grid corners.

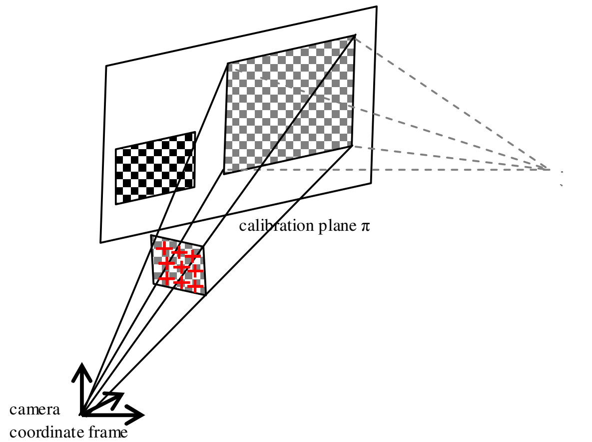


Figure 3 Corner extraction of the projected chekerboard

4.3 Consider the rays that go from the camera trough the corners of the projected pattern

Next step is to build the expression of the 3D rays that go from to optical centre of the camera trough the corners of the projected pattern that have been extracted in the previous step.
To obtain the 3D ray that goes through a corner (Cx,Cy) in the image plane, we apply the following projective transformation using the camera parameters:

where the vector of the ray (R_x,R_y,R_z) is up to a scale factor *s*.
In figure x it can be seen the graphical representation of the rays we are computing. Although they are depicted with a particular length, they are computed up to scale, and precisely this is the scale that has to be determined in order to intersect them with the calibration plane and obtain the 3D points.

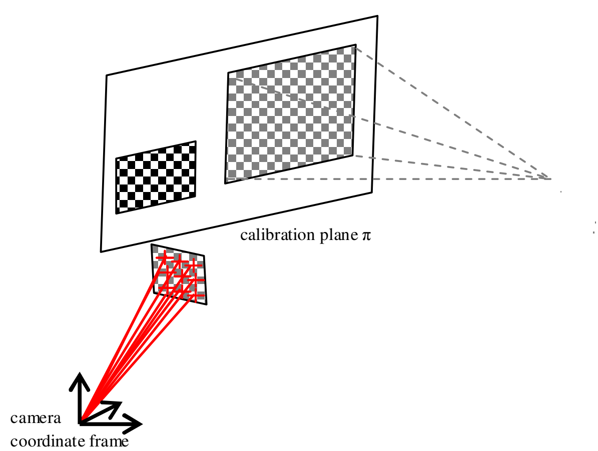


Figure 4 Rays coming from the camera and going through the corners of the projected pattern

4.3 Ray-plane intersection

Finally the intersection between the rays and the plane must be determined to obtain the 3D position of the projected corners. This means finding the *s* factor that makes the ray satisfy the equation of the plane. Therefore, from our equation of the calibration plane:

We substitute the values of the ray:

As *a,b,c* and *d* are known parameters of the plane, R_x, R_y and R_y are also known, the *s* satisfying the equation can be recovered.

Hence, once *s is* determined we have the 3D position of the projected corner. In our method this is done for all the corners in the image and for different images so as to obtain a big number of non-coplanar 3D points for the calibration.

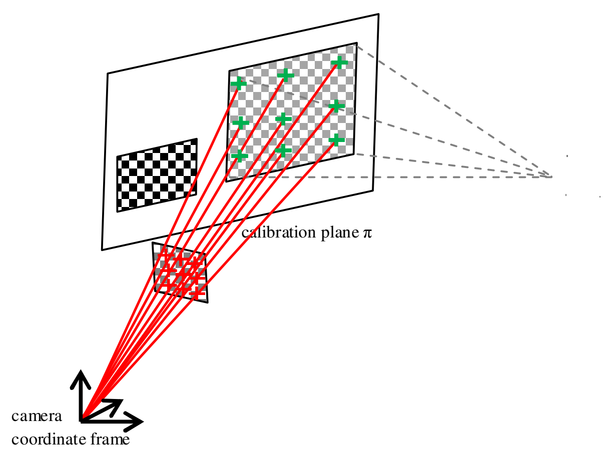


Figure 5 Ray-plane intersection gives 3D projected points

5. Projector calibration

A projector can be regarded as the inverse of a camera, because it projects images instead of capturing them. Thus if we have the 2D-3D of the projected points, the calibration of a projector is essentially the same as that of a camera, which is well established.

The projected 3D points were computed using the steps detailed in the previous section. The 2D points of the projected image are quite simple to obtain. It is only required to take the image of the pattern that is being projected (i.e a screenshoot) and extract the corners. It has to be taken into account that this image has to be at the same resolution in which the projector is displaying.

Once we have the 2D-3D correspondences we can proceed to the projector calibration using some standard camera calibration method.
In a first approach we implemented a *Faugeras* calibration algorithm with one level of distortion removal and to enter these correspondences. Although the results were relatively good, we finally decided to use the main routine of calibration used in *Bouguet* toolbox. In this way, we use the same procedure to calibrate the camera and the projector. As introduced in section 3, *Bouguet*'s calibration function follows Zhang's method and minimizes the pixel reprojection error in the least squares sense over the intrinsic camera parameters (including the distortion), and the extrinsic parameters (3D locations of the grids in space). Furthermore using *Bouguet* function for calibration we are able to easily take into account up to a 6^{th} level of radial distortion plus tangential distortion.

Therefore, after the calibration is done we obtain the optimized intrinsic and extrinsic parameters for the projector. In order to evaluate the results, the error of the corner reprojection over the images is given as well as an explanatory plot that shows the coordinate systems of both camera and projector together with all the calibration planes.

6. How to use the toolbox extension to calibrate a camera-projector system.

6.1 Installation Procedures

The Camera Projector toolbox requires that *Bouguet's Camera Calibration* toolbox be present and accessible by Matlab (see website). Unzip the file containing the code to a directory. The Camera calibration toolbox directory can be used in order to keep the files together. The next step is to add the created directory (if other than the root of the original toolbox) to the Matlab **path.**

6.2 Running the Camera-Projector Calibration Toolbox.

To run the camera-projector calibration toolbox simply type *cam_proj_gui* in the Matlab command prompt and press return (enter). This will bring the main user interface for the toolbox.

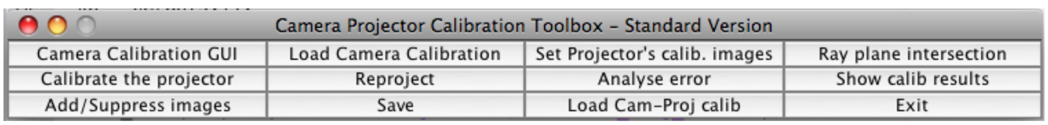


Figure 6 Camera Projector Calibration User Interface

6.3 How to proceed

The first step to calibrate the system is to have the camera calibrated. In order calibrate the camera the user can choose one of the two buttons of the user interface highlighted in red in figure 6. The first one "Camera Calibration GUI" calls the standard *Bouguet Toolbox* GUI for camera calibration. The second one allows the user to load a camera calibration result.

Once the camera is calibrated the next step (highlighted in yellow in the figure 6) is generate the data to perform the calibration. This data consists of the results of the intersection between the ray defined by the calibrated camera and the plane computed by extrinsic calibration. The first button allows the user to select different images from those ones used during the camera calibration. The second button computes the 3D points based on the intersection of rays defined by corners detected in the camera image plane of the perceived projected pattern and the plane obtained via extrinsic calibration (using the printed pattern).

The blue button performs the calibration.

The green zone is to analyze the calibration results. The GUI is equipped with a reprojection function, an error analyzer and a 3D representation of the system and acquired data.

The "Add/Suppress images" button allows the activation or deactivation of the images to perform the calibration.

Finally the white section permits save or load the system calibration and exit the GUI.

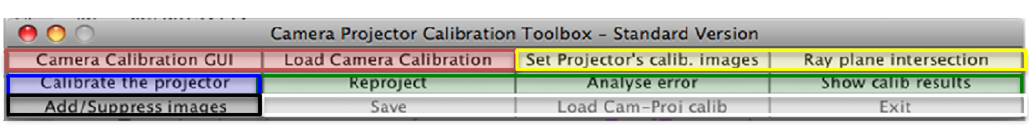


Figure 7 Different parts of the user interface

6.4 A step-by-step walk-through

For this example is assumed that the user is already familiar with the camera calibration process using the *Bouguet Toolbox*. Hence we will start from the point of loading the Camera Calibration results.

1. Click the Load Camera Calibration button. The files should be in the current directory.

2. The next step is to select set of images to calibrate the projector. Here the user can choose if he wants to use the same images used for the camera calibrations (they should contain the Projected Pattern as well) or select another set of images that contains both patterns. We will follow here the most general case of different images, because the other method is straightforward because the extrinsic of the plane are already known.

   When asked for the image set to use, selecting any character tells the toolbox to use different dataset. After this the user has to enter the base name of the images. For example if all our images begin with DSC we type it as a base name and hit enter.

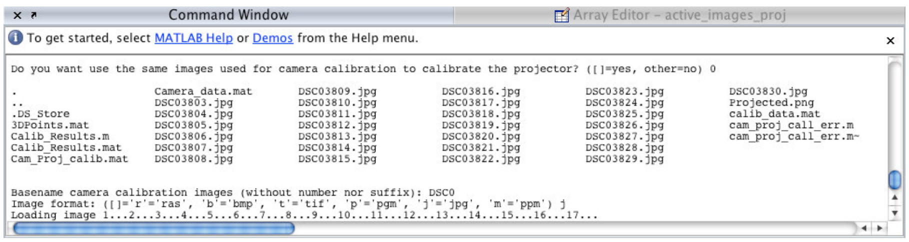


Figure 8 Console view for the images selection

   The user now has to click on the four corners of the Printed pattern in all the images in order to determine the position and orientation of the plane in space. Once selected the pattern its corners are computed and displayed.

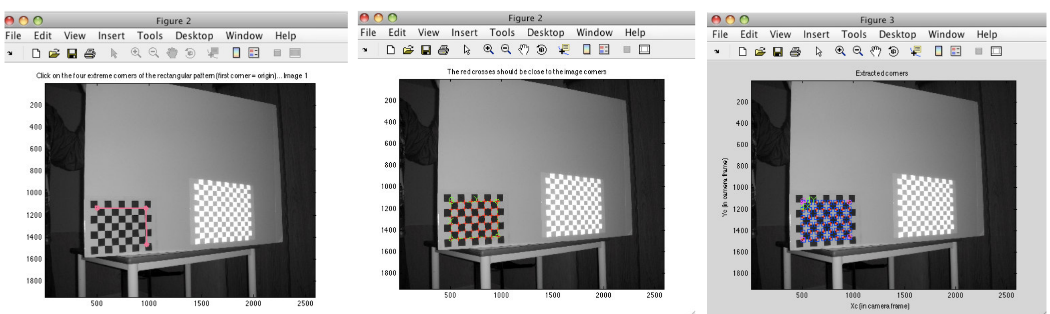


Figure 9 Printed pattern selection procedure

3. Now the intersection between the rays and the planes can be computed by pressing the Ray Plane Intersection button. Done this, the user is asked to click on the four extreme corners of the projected pattern in order to extract the interesting points for the ray tracing. This is repeated for all the active images. After this the toolbox will ask for the projected image used in order to build the matching's. This image should have the same resolution of the projector at the time of the calibration. The corners of this image have also to be extracted.

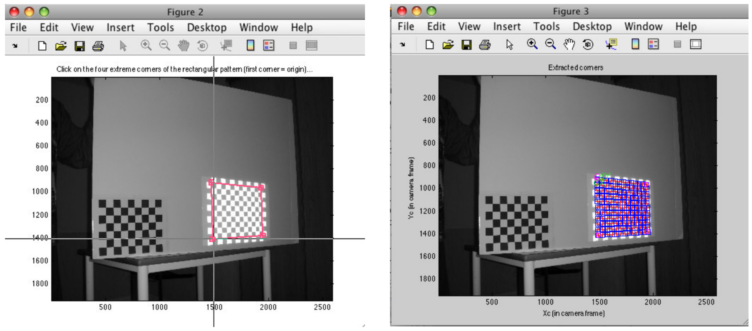


Figure 10 Projected pattern selection (camera frame)

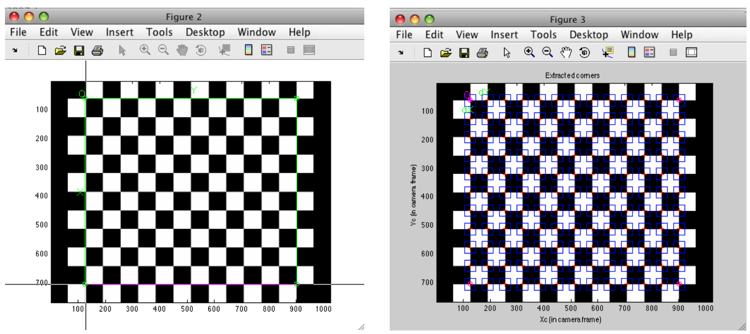


Figure 11 Projected pattern selection (projector frame)

4. Now all the data for calibration is available and the button Calibrate the Projector can be activated. This will fire the main calibration routine and optimization. At the end the computed parameters will be displayed along with their corresponding uncertainty.

5. Clicking in Reproject will reproject the 3D points into the projector image plane and display the corresponding errors. In the same fashion as in the original toolbox for camera calibration.

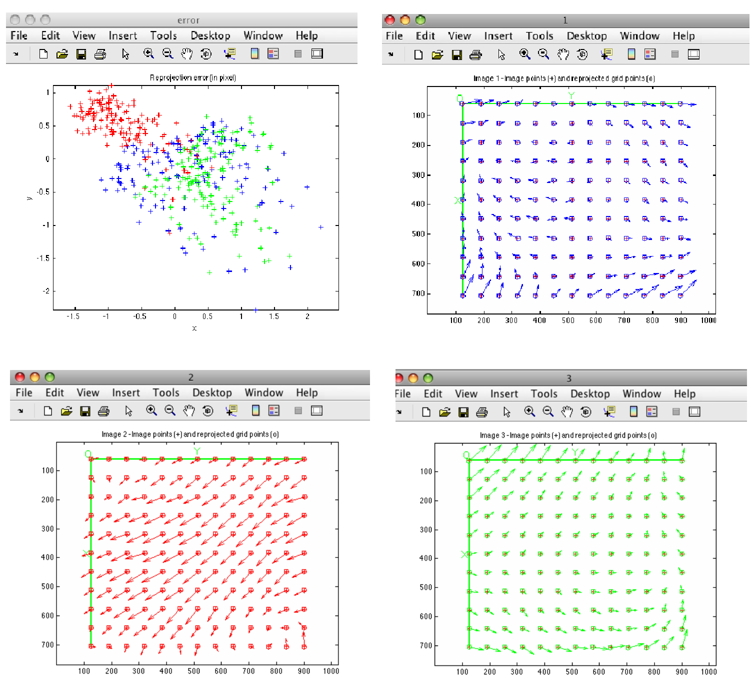


Figure 12 Reprojection error view

6. The user is invited to experiment the analyse error and the Show calib results buttons in order to have a better idea of the results obtained by the calibration.

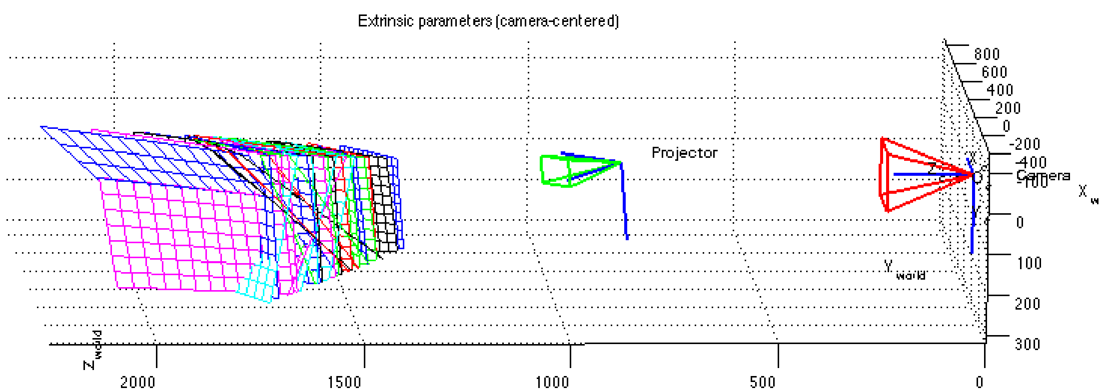
7. If not satisfied with some results it is possible to add/ suppress image from the dataset and re-run the calibration.

7. Performance evaluation

In order to evaluate the results, the error of the corner reprojection over the images is computed. Since the camera calibration is used as a preliminary step for the projector calibration, the error of the projector calibration will depend also in the amount of error of the camera calibration. The error also depends in the accuracy of the setup and of course of the corner extraction. Special care has to be taken in this step and exclude from the calibration the images that have suffered a bad corner extraction or present high errors for some other reason.

Several tests of camera-projector calibration have been performed. The experiments were realized in a not-so accurate setup (because of the available material and conditions) but being careful in the corner extraction. Under these variables, we have seen that our method has reprojection errors of around 1 pixel or less (being the camera calibration error already around 0.5 pixels), meaning that the overall system offers quite good results and have enough accuracy to deal with structured lighting applications for instance.

Other useful thing not exactly to evaluate the performance, but to proof the correctness of the calibration parameters is the plot that shows both the camera and projector coordinate systems together with all the calibration planes. In this plot the extrinsics of the calibration can be visually checked and compared to the actual setup. Next figure shows an example of this plot.



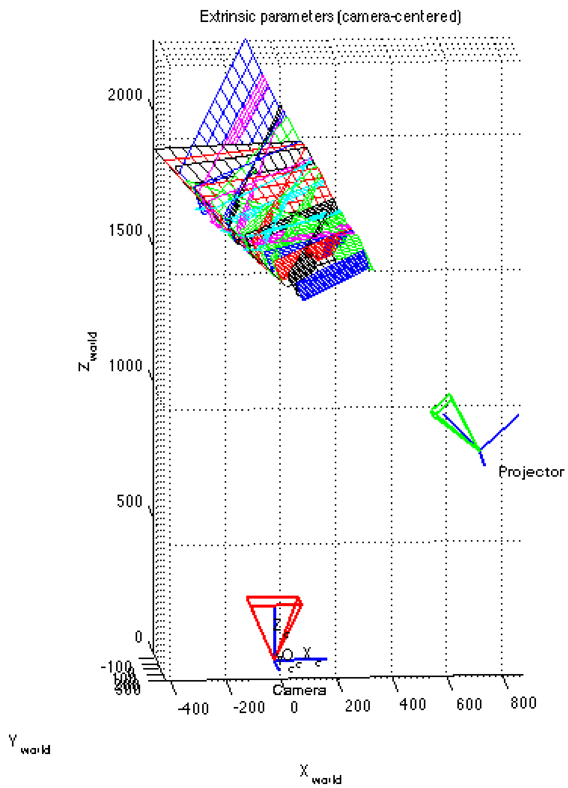


Figure 13 Plot of the system extrinsic parameters

The relation between the camera and projector coordinate systems can be clearly shown. In this case the plot corresponds to the setup presented on figure 1. Paying attention it can be seen than the principal point of the projector is not close to the image centre because of an off-axis projection. This is one particularity of most video projectors.

Finally, in order to show in another way the performance of our method, we have tried to do a 3D reconstruction using coded light. In order to try this we have used the Coded Structured Light Demo provided by Salvi and Pages [4]. The procedure is the following one:

- Given an image of an object illuminated with a coded pattern (we used Zhang's pattern [5]) the correspondences between the input image and the projected pattern are searched.
- Then a list of correspondences between camera pixels and projector columns is generated using the information of the camera-projector calibration. The points are reconstructed by intersecting a line coming from the camera (passing through the matched pixel) and a plane (matched column) coming from the projector.
- Finally with the list of 3D reconstructed points a .BYU file is generated where triangles have been defined between the cloud of points.

The figure on the left is an example of an input image illuminated with Zhang's pattern. After applying the above procedure a 3D reconstruction like the one of the right figure can be obtained.

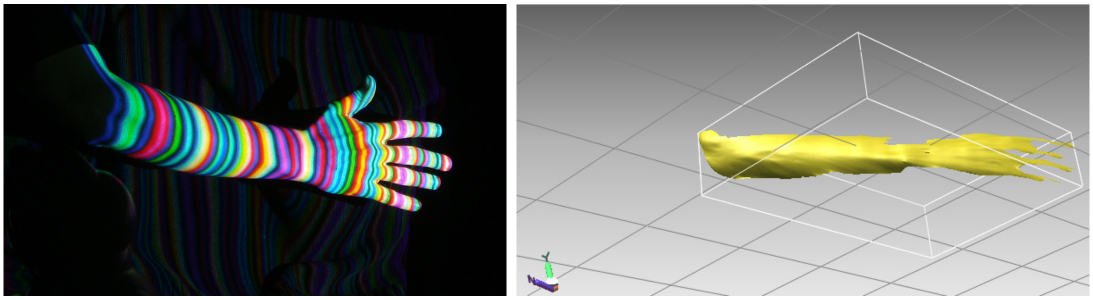


Figure 14 Projected code light pattern and 3D reconstruction

As it can be seen the reconstruction is not very accurate, but its accuracy level is more or less the same as that of other reconstructions obtained with this method (one-shoot pattern projection). In order to obtain a better reconstruction, images of different views could be taken and then apply a registration of the obtained 3D clouds of the different images. However it can be seen that the calibration of the camera-projector performed reasonably well in a real application.

8. Conclusion and future work

In this report we have described a plane-based calibration method for projector-camera systems. The proposed method has been shown to be accurate and easy-to-use.

The simplicity of the method rely in one hand in the fact that is a plane-based approach and in the other in considering the projector as the dual of a camera and thus making the calibration of a projector the same as that of a camera, for which there already exists well established methods. In that sense we have taken profit of the *Bouguet's Camera Calibraton Toolbox* which implements Zhang's calibration. Thus, this calibration method is used to obtain both camera and projector parameters through an iterative minimization procedure that already takes into account several levels of distortion. So in order to be coherent the method proposed in this paper has been implemented as an extension of Bouguet's toolbox and makes extensive use of its functions.

The acquisition step, consisting in taking pictures of a planar calibration board with a printed pattern and a projected one, should not be a problem for person who is doing computer vision research and the computation of the parameters is automatically solved by the toolbox. However the intermediate step of extraction of calibration pattern corners is the part that can become more tedious, specially as the number of the images grows. Thinking in possible improvements, some color markers or other kind of fidicials could be used to automatically extract corners.

Regarding performance issues, the method achieves relatively good results (around 1 pixel of error for the overall system). We have seen that this error depends on the camera calibration error as it is used as a preliminary step for the rest of the calibration method. Improving the setup conditions and a being carefully in the corner detection step could decrease this error. As a future extension, a final step of bundle adjustment over the calibration parameters could be implemented to reduce even more the error.

In addition, to further verify the correctness of our calibration method a simple reconstruction of a 3D object using coded lighting pattern has been performed. Reasonably good results were obtained taking into account the nature of the method.

9. REFERENCES

[1] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[2] Jean-Yves Bouguet, Camera Calibration Toolbox for Matlab®, http://www.vision.caltech.edu/bouguetj/calib_doc/

[3] Intel OpenCV Computer Vision Library (C++), http://www.intel.com/research/mrl/research/opencv/

[4] J. Pagès, J. Salvi, J. Forest. Optimized De Bruijn Patterns for One-Shot Shape Acquisition. Image and Vision Computing 23(8), pp 707-720, August 2005.

[5] L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Int. Symposium on 3D Data Processing Visualization and Transmission*, pages 24–36, Padova, Italy, June 2002.