

Simple, Accurate, and Robust Projector-Camera Calibration

Daniel Moreno Gabriel Taubin
Brown University
Providence RI, USA
{daniel.moreno,taubin}@brown.edu

Supplementary material

1. Introduction

In addition to the theoretical contribution developed in the main paper submission, we have implemented our method as a calibration software available as supplementary material. This document explains how to use the software to calibrate a structured-light system given a set of gray code images. Together with the calibration software, we have provided a subset of images we have used to calibrate our system. Given the supplementary material file size limit of 30MB, we could only upload a minimal set of three plane orientations and all the images had to be resized to one half their original size and their JPEG quality reduced to decrease their file size.

2. System requirements

We have uploaded a binary executable named `projector-calib.exe` for Microsoft Windows. The file is a 32 bits executable and can be used in both 32bits and 64bits OS's. The only prerequisite is to have installed the latest version of Microsoft Visual C++ runtime that can be downloaded from

- Microsoft Visual C++ 2010 Redistributable Package (x86):
<http://www.microsoft.com/en-us/download/details.aspx?id=5555>

We have tested the software on Microsoft Windows 7 64 bits.

3. Sample data

The sample data can be found inside a directory named `calib-small`. This directory contains three subdirectories with images named from `01.jpg` to `42.jpg` corresponding to three plane orientation gray code sequences. The directory tree is organized as follows:

The original images were taken with a Canon EOS Rebel XSi at 12Mpx, they were downsampled to 50% of their size and their JPEG compression increased to make smaller files, the resulting image set is for demonstration purpose only. All the results shown in the main paper were obtained using the original images without any modification.

4. Software operation

The calibration software processes the complete image set automatically, detecting the checkerboard, decoding the gray patterns, calibrating both camera and projector intrinsics, and finally it calibrates their relative rotation and translation as a stereo system. Now, we will explain how to use it to calibrate a structured-light system.

4.1. Select image set location

The first time the software runs it looks like Figure 2. In this screen click on "Change dir..." to open the directory selection dialog as shown in Figure 3. Once the directory selection dialog opens browse the disk to find the directory containing the calibration image subsets. In the sample data you must select the folder named "calib-small".

```
calib-small
|
+---2012-Jun-28_17.44.05.965
|   +---01.jpg
|   +---02.jpg
|   + ...
|   +---42.jpg
|
+---2012-Jun-28_18.01.39.650
|   +---01.jpg
|   + ...
|   +---42.jpg
|
+---2012-Jun-28_20.01.52.916
|   +---01.jpg
|   + ...
|   +---42.jpg
```

Figure 1. Sample calibration images directory tree

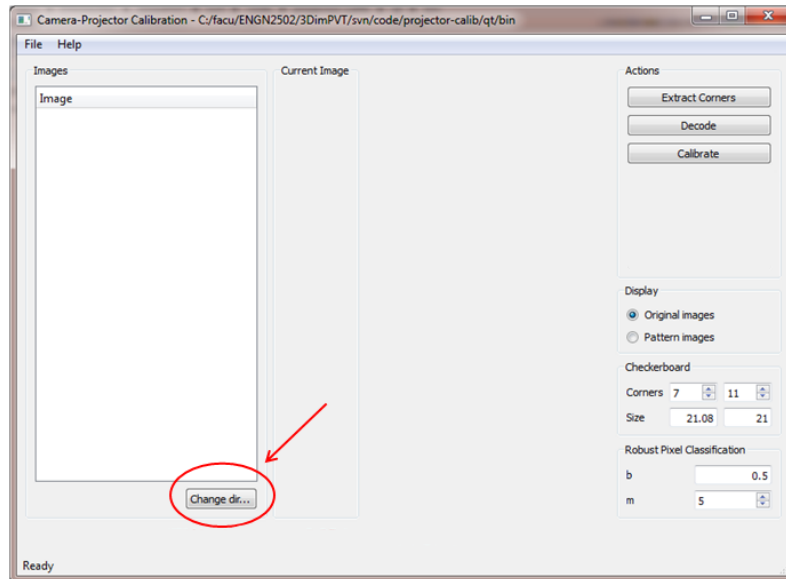


Figure 2. Main screen: click on “Change dir...” to select a directory

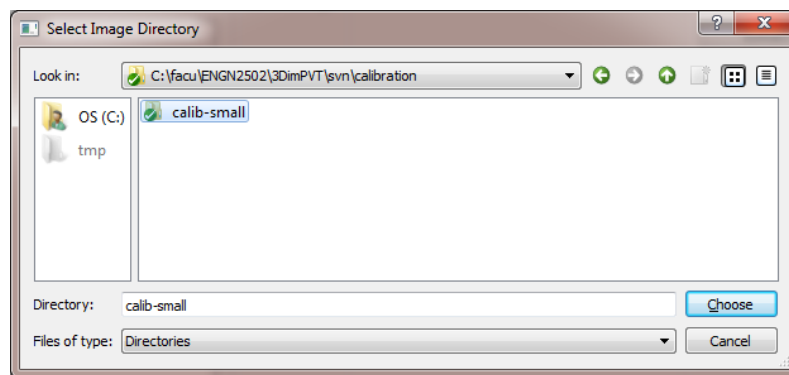


Figure 3. Select the directory containing all the plane orientation gray code sets

4.2. Checkerboard dimensions

Once a directory is selected the main screen will display the first image from the first subset as in Figure 4. Here, you must set the checkerboard corner count and size in the bottom-right boxes. The checkerboard from the sample set has 7x11 corners

and their size is 21.08mm x 21.00mm. Advanced users could modify the parameters from the “Robust pixel classification” box, although, the provided values of $b = 0.5$ and $m = 5$ work fine for a large range of images.

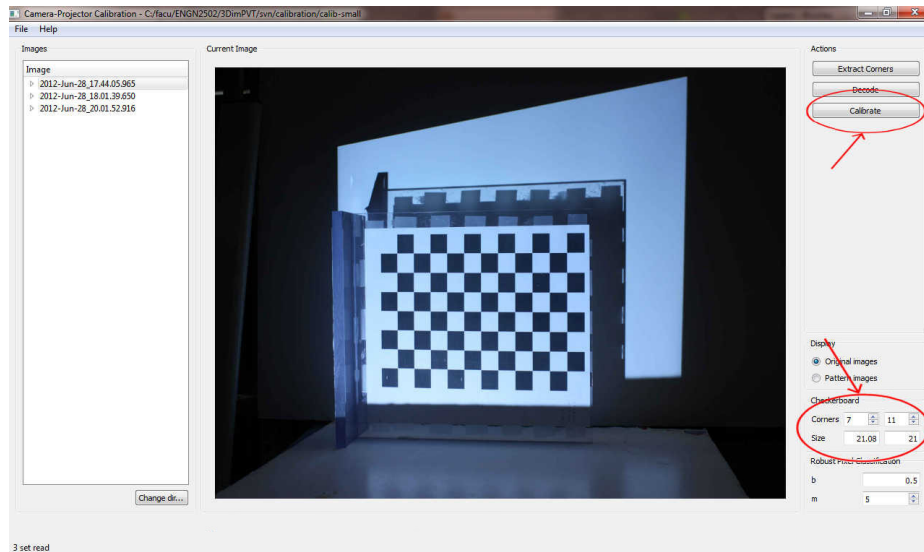


Figure 4. Main screen after directory selection: set checkerboard dimensions and press “Calibrate”

4.3. Calibration

To initiate the calibration just press “Calibrate” from the right side bar, Figure 4. A small dialog as Figure 5 will open displaying details about the calibration algorithm as it runs. Calibration with the sample data will take less than a minute and the dialog will remain open after finish to allow the user examine the results. The final calibration will be automatically saved in YAML format in the image set directory. The YAML file format is the standard OpenCV file storage and can be easily loaded by other tools.

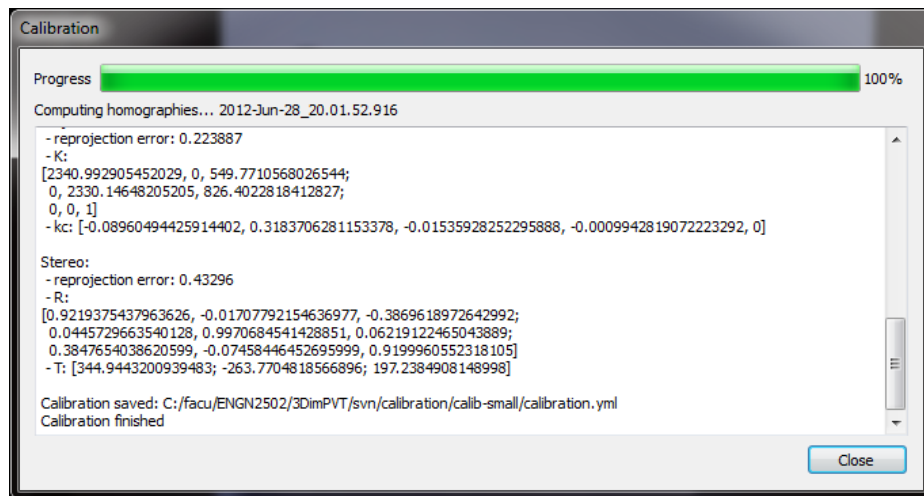


Figure 5. Calibration progress dialog

4.4. Decoded images and detected checkerboard visualization (optional)

After the calibration progress dialog (Figure 5) is closed, the main screen will overlap the detected checkerboard corners on top current image. Additionally, clicking in “Pattern images” radio button (Figure 6) will switch the view to display

the decoded projector column and row images. Usually, the image region containing the checkerboard plane will have no uncertain pixels (gray color) and, because it is a plane, colors will change smoothly on it. Other regions of the images might not be correctly decoded but they are not used by the calibration algorithm.

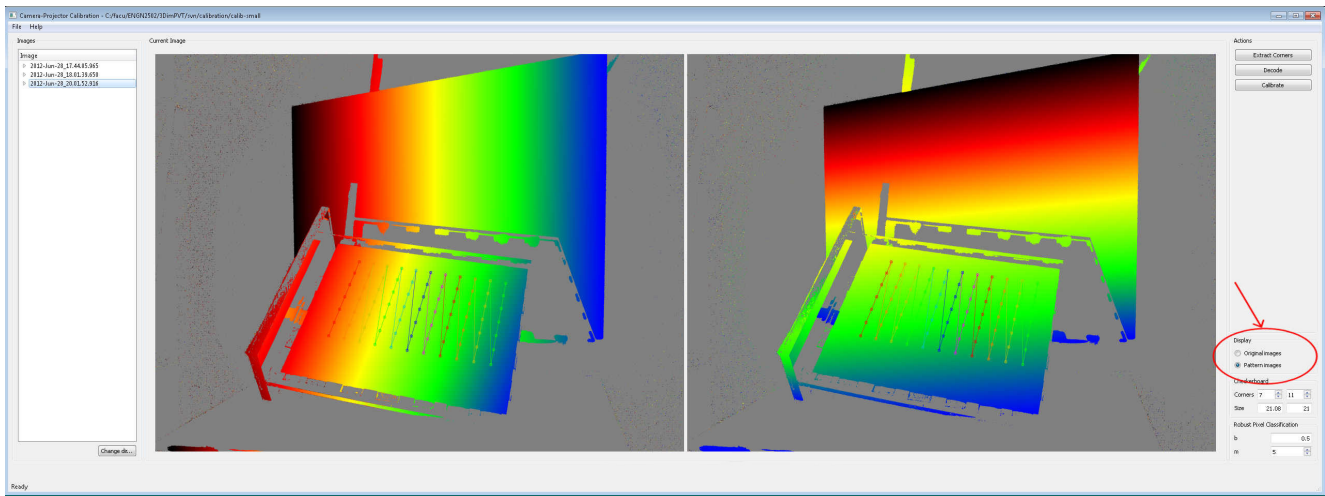


Figure 6. Decoded pattern images

5. Sample calibration results

Figure 7 shows the results we got by running the calibration software with the sample data. Note that the result is split in three sections each of them with its own reprojection error: camera intrinsics, projector intrinsics, and stereo calibration.

5.1. Camera intrinsics

$$K_{cam} = \begin{bmatrix} 2472.650464858401 & 0 & 1080.34400778601 \\ 0 & 2478.277708298557 & 592.8451712869836 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Camera distortion coefficients:

$$k_{cam} = [-0.07008445244701324, -0.4577870235611326, -0.01161241742799688, 0.001468694637194292, 0] \quad (2)$$

Camera reprojection error: 0.210507

5.2. Projector intrinsics

$$K_{proj} = \begin{bmatrix} 2472.650464858401 & 0 & 1080.34400778601 \\ 0 & 2478.277708298557 & 592.8451712869836 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Projector distortion coefficients:

$$k_{proj} = [-0.08960494425914402, 0.3183706281153378, -0.01535928252295888, -0.0009942819072223292, 0] \quad (4)$$

Projector reprojection error: 0.223887

5.3. Stereo calibration

Rotation matrix:

$$R = \begin{bmatrix} 0.9219375437963626 & -0.01707792154636977 & -0.3869618972642992 \\ 0.0445729663540128 & 0.9970684541428851 & 0.06219122465043889 \\ 0.3847654038620599 & -0.07458446452695999 & 0.9199960552318105 \end{bmatrix} \quad (5)$$

Translation vector:

$$kC_{proj} = \begin{bmatrix} 344.9443200939483 \\ -263.7704818566896 \\ 197.2384908148998 \end{bmatrix} \quad (6)$$

Stereo reprojection error: 0.43296

```
Extracting corners:
* 2012-Jun-28_17.44.05.965: found 77 corners
* 2012-Jun-28_18.01.39.650: found 77 corners
* 2012-Jun-28_20.01.52.916: found 77 corners

Estimate direct and global light components... done.
Decoding, please wait...
* 2012-Jun-28_17.44.05.965: finished
Estimate direct and global light components... done.
Decoding, please wait...
* 2012-Jun-28_18.01.39.650: finished
Estimate direct and global light components... done.
Decoding, please wait...
* 2012-Jun-28_20.01.52.916: finished

* Calibrate camera
* Calibrate projector
* Calibrate stereo

**** Calibration results ****

Camera:
- reprojection error: 0.210507
- K:
[2472.650464858401, 0, 1080.34400778601;
 0, 2478.277708298557, 592.8451712869836;
 0, 0, 1]
- kc: [-0.07008445244701324, -0.4577870235611326, -0.01161241742799688, 0.001468694637194292, 0]

Projector:
- reprojection error: 0.223887
- K:
[2340.992905452029, 0, 549.7710568026544;
 0, 2330.14648205205, 826.4022818412827;
 0, 0, 1]
- kc: [-0.08960494425914402, 0.3183706281153378, -0.01535928252295888, -0.0009942819072223292, 0]

Stereo:
- reprojection error: 0.43296
- R:
[0.9219375437963626, -0.01707792154636977, -0.3869618972642992;
 0.0445729663540128, 0.9970684541428851, 0.06219122465043889;
 0.3847654038620599, -0.07458446452695999, 0.9199960552318105]
- T: [344.9443200939483; -263.7704818566896; 197.2384908148998]

Calibration saved: C:/facu/ENGN2502/3DimPVT/svn/calibration/calib-small/calibration.yml
Calibration finished
```

Figure 7. Sample data calibration results