Kinect Calibration Toolbox

Carolina Raposo
Institute of Systems and Robotics
University of Coimbra, Portugal
http://arthronav.isr.uc.pt/~carolina/

This document is a guide on how to use the toolbox which accompanies the paper "Fast and accurate calibration of a Kinect sensor" (3DV 2013). It uses several functions from the publicly available Kinect Calibration Toolbox by Daniel Herrera (www.ee.oulu.fi/~dherrera/kinect/) [1], as well as some images from its dataset. Also, functions from Bouguet's toolbox (http://www.vision.caltech.edu/bouguetj/calib_doc/index.html) [2] are used in the color camera calibration step. Both these previous works have been referenced in our published paper.

Using the toolbox

A graphic user interface is provided, and the calibration can be achieved in 4 simple steps. The calibration result can be afterwards assessed by computing the reprojection errors for chosen images, or overlaying a depth map over the corresponding RGB image.

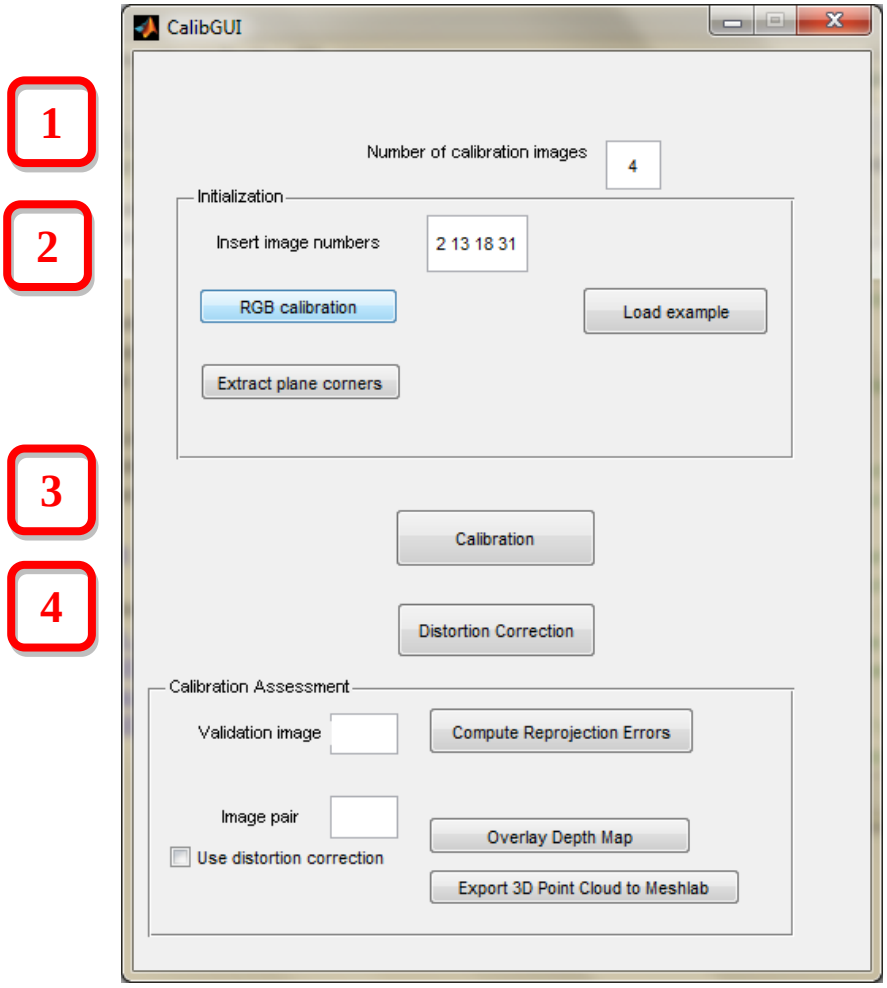

Figure 1 - Graphic User Interface for the Kinect Calibration Toolbox.

1 – Select the number of calibration images

Using only 3 calibration pairs, it is possible to achieve good calibration results. However, there has to exist enough variability between the images, or the calibration will be poor. Any number of calibration images larger than 2 can be chosen.

2 – Initialization

i) In this step, it is necessary to indicate the numbers of the calibration pairs, where the inserted numbers must be +1 than the number in the image name. As an example, the pair no. 1 corresponds to the pair 0000-c1.jpg (RGB image) – 0000-d.pgm (disparity map).

ii) Next, an RGB calibration is performed using Bouguet's toolbox [2]. Note that this step uses a .mat file with the extracted corners for all images in the dataset, which must be previously generated by the user. It has the format delivered by that toolbox and must have the name *Calib_Results*.

iii) Finally, plane corner extraction must be performed in the disparity images. This toolbox requires the user to input metric coordinates of known objects. If the plane size is known, it can be used as this metric requirement. Thus, after extracting the corners, the user can optionally assign metric coordinates (in millimetres) to the plane corners. The next images show how this can be done.

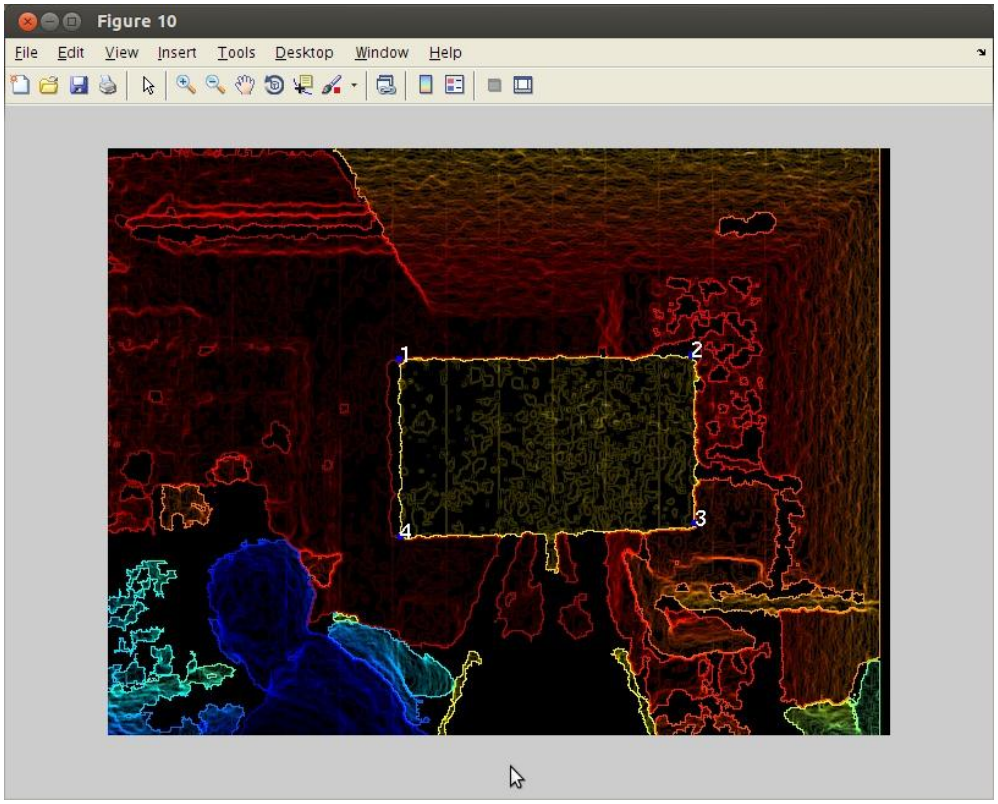

Figure 2 - Plane corner extraction in disparity maps.

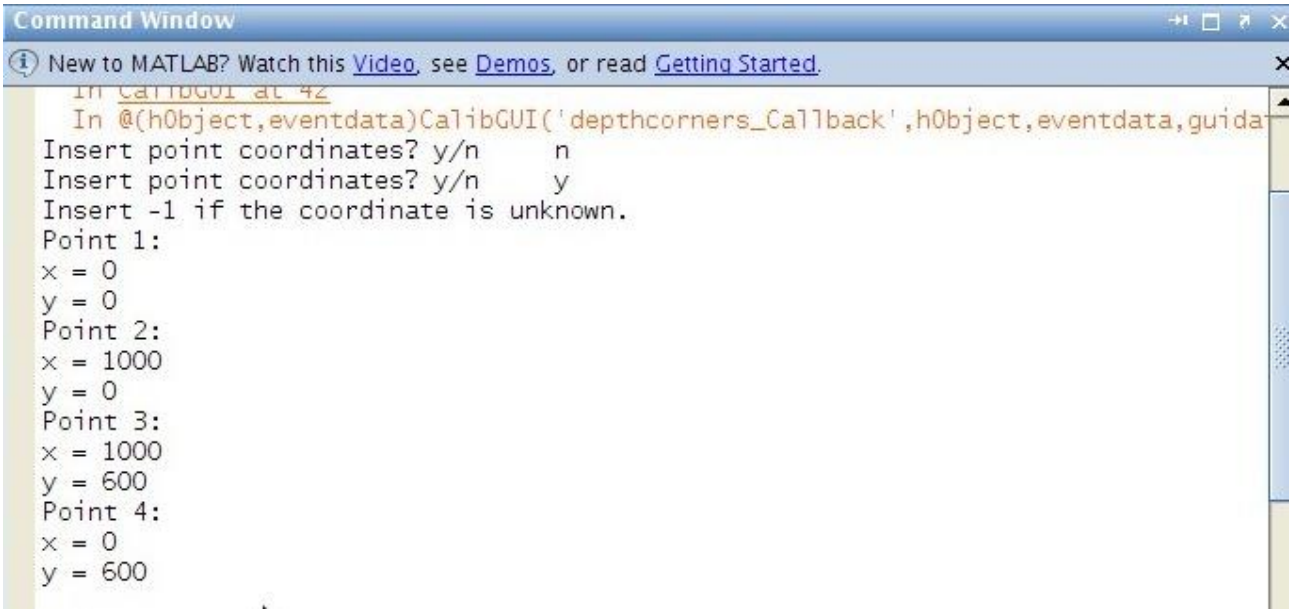

Figure 3 - Metric coordinates assigned to each plane corner in Figure 2.

Note: Example files are given for initializing the parameters, for 3, 4, 5 and 6 calibration pairs. The user can use these by simply clicking on "Load Example".

3 – Calibration

In this step, the initialization of the extrinsic calibration is computed, as well as the optimization of all the parameters (for more details refer to the original paper). The final calibration result is compared to the calibration provided in Herrera's toolbox [1], which was computed using 70 plane poses.

4 – Offline distortion correction

In this final step, the parameters of the model assumed by Herrera *et al.* [1] are determined in a single step using the calibration images.

Calibration assessment

The obtained calibration results can be assessed by computing the reprojection errors for chosen validation images, and by overlaying the RGB image with corresponding depth map, with the option to use the distortion parameters. Also, an option to export the 3D point cloud to a .ply file (opened with Meshlab) has been added. An example reconstruction is shown in Figure 4. Note that especially when using only 3 calibration images the distortion correction may be poorly estimated, in which case it is preferable to overlay the depth map without considering it.

Figure 4 - Example of the reconstruction of an image not used in the calibration.

Useful Functions

Some important functions have been developed, which may be useful for the user:

ComputeRMSerrorpix – computes per pixel reprojection errors in disparity units

Point3Dfromdisp – computes the 3D point cloud from pixel values and disparities

ProjectPoints – takes 3D points and projects them to another reference frame, delivering the corresponding pixel values

References

[1] D. Herrera C., J. Kannala, and J. Heikkila. *Joint depth and color camera calibration with distortion correction*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(10), 2012.

[2] J. Y. Bouguet. *Camera calibration toolbox*, 2010