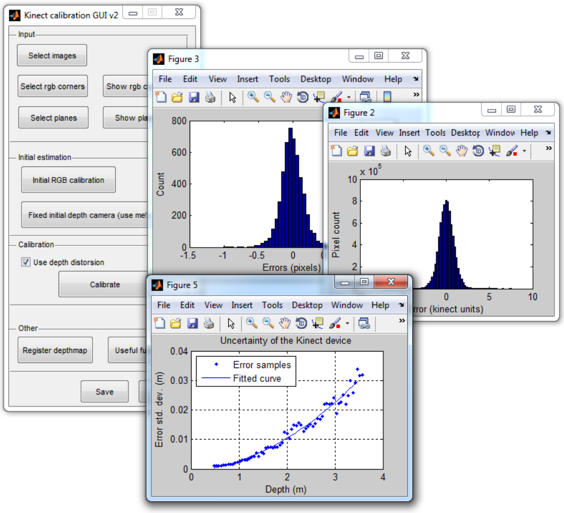
Kinect Calibration Toolbox

v2.1 (2012-06-15)
by Daniel Herrera C.
Center for Machine Vision Research
University of Oulu



The Kinect Calibration Toolbox offers a principled calibration approach for the Kinect sensor. It avoids the use of edges or corners in the disparity images because of edge noise in the Kinect data. This toolbox accompanies the paper "**Joint depth and color camera calibration with distortion correction**" (TPAMI 2012) and implements the algorithms described there. This is only a basic guide in how to use the toolbox. Please refer to the paper for further details. If you use this toolbox for your research, we appreciate that you cite our paper. Please visit the toolbox website for the complete reference: www.ee.oulu.fi/~dherrera/kinect/

Toolbox features

- Calibrates the Kinect RGB camera, the Kinect depth camera, and an external camera simultaneously.
- Inlcudes a disparity distortion model that improves the manufacturer's calibration.
- Portable Matlab implementation for calibration. Real time C++ implementation to transform disparity image to a point cloud using the obtained calibration.

Toolbox contents

`./toolbox`	matlab code for calibration. Run kinect_calib_gui.
`./capture`	a modified version of libfreenect glview to capture images.
`./doc`	this file
`./data`	sample images and calibration results

Calibration model

Both color and depth cameras follow a pinhole camera model with distortion. However, the distortion models differ slightly between the color and depth cameras. There is also a disparity distortion model for the values of the depth camera. This corrects the received disparity according to the formula:

$$d_k = d + D_\delta(u,v) \cdot \exp(\alpha_0 - \alpha_1 d)$$

Where d is the disparity returned by the Kinect, (u,v) are the pixel coordinates, and d_k is the corrected disparity. The other parameters are calibrated coefficients. This distortion correction has been shown to improve the accuracy of the reconstruction compared to the manufacturer's calibration, especially in the near range (<1.5m). An example of the obtained accuracy is shown below. Please refer to the paper for the full description and justification of the model.

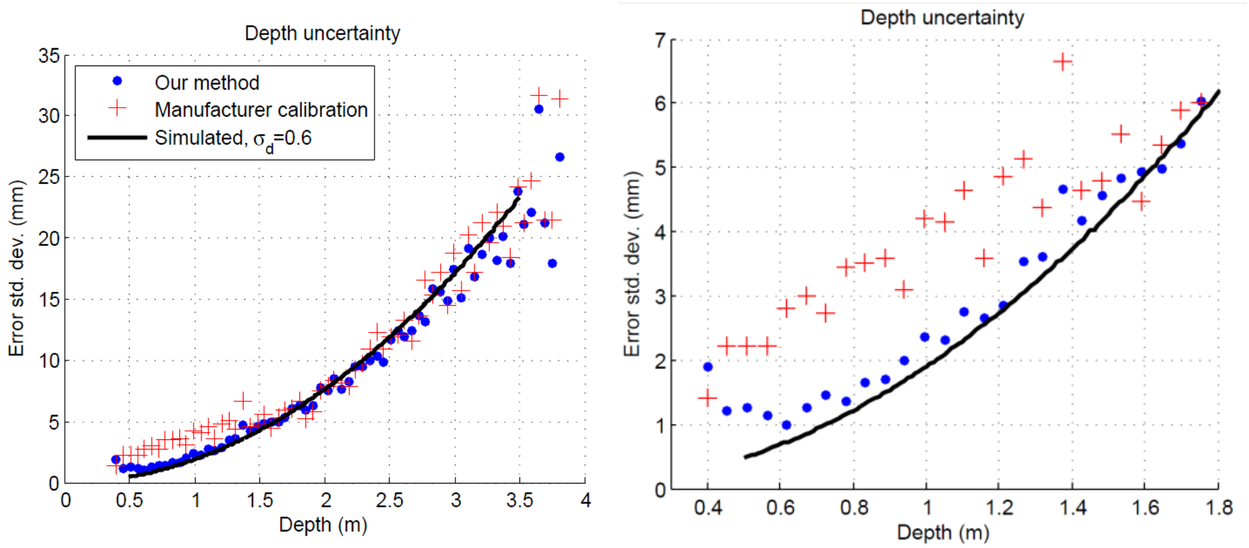


Figure 1 Comparison of reconstruction accuracy. **Left:** full calibrated range. **Right:** near range zoom in.

Calibrating an external color camera

The toolbox can calibrate an additional external color camera that is rigidly attached to the Kinect. If the external camera has a higher resolution than the Kinect color camera, it can improve the pose estimation of the plane during calibration. It is recommended to use an external camera to achieve the best possible calibration.

Preparing for calibration

Calibration object

A smooth plane with a checkerboard pattern is required for calibration. The pattern doesn't need to span the whole plane, just big enough so that the checkerboard corners can be detected in all the color images. An A4 checkerboard pattern glued to a table is a recommended option.

Capturing images

The algorithm requires several images (around 30) from different angles for proper calibration. For a successful calibration 4 types of images must be present: frontal plane, plane rotated around the X axis, plane rotated around the Y axis, and full image planar surface for distortion correction. Samples of the first three types are shown in Figure 2. It is recommended to capture at least 5 images of each type with the plane at different distances from the camera.

Figure 2 Sample images. **Left:** fronto-parallel. **Center:** rotated around Y axis. **Right:** rotated around X axis.

If you want to calibrate the disparity distortion model, you should also take some images of a bigger plane (e.g. a wall) that fills the entire image. The wall should be fronto-parallel to the image plane and it should be imaged at various distances. No checkerboard pattern is needed for this images. Do not include the RGB images of the wall, only the disparity images.

Image format

The color images may be stored in any format readable by Matlab. If you capture raw grayscale images with the Kinect, use Matlab's `demosaic()` function to convert them to RGB. The disparity images must be stored in PGM format (header should read "P5 640 480 2048", note that PGM files are stored in big-endian format regardless of your machine's byte-order).

Important: the disparity images MUST be in raw disparity units as returned by the libfreenect library. Other drivers return the data in millimiters. The toolbox could be adapted to calibrate with this data but it would require some modifications in the code. If you want to use this toolbox, it is highly recommended to use libfreenect.

File naming

The toolbox allows you to select the format of the filenames used. The default format is 0000-c1.jpg for the Kinect color camera, 0000-c2.jpg for the external camera, and 0000-d.pgm for the disparity map. Images corresponding to the same plane pose should have the same prefix number. Not all images are necessary for all plane poses. That is, a given plane pose may have only one color image, only the depth image, or any combination. The calibration uses the information from those images that are present.

Testing the toolbox

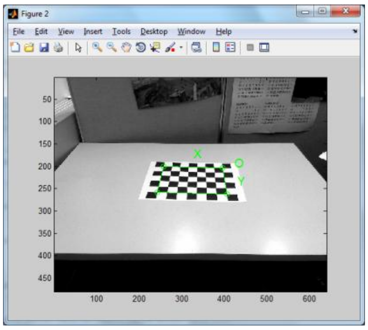
In order to test that the toolbox is running properly you can use the sample data provided. Follow the steps below to ensure that all is good with your installation:

- Copy the sample data to any folder in your HDD (e.g. `c:\kinect_toolbox\data`)
- Load a sample calibration: `do_load_calib('c:\kinect_toolbox\data\small_set.mat');`
- Add the global variables to your workspace: `global_vars();`
- Change the dataset path to your folder: `dataset_path = 'c:\kinect_toolbox\data\';`
- `Calibrate: do_calib(true);`

The toolbox should perform a quick calibration using only a small subset of the provided images. The images used for this subset are [1,4,9,16,17,20,25,27,32,38,39:2:54,60]. The reprojection errors should be around 0.2px, 0.5px, and 0.8kdu for the color, external, and depth cameras respectively.

Calibration procedure

The calibration procedure has 6 steps described below.



1. Select images

2. Select corners in RGB images

The corner extraction can be done manually or automatic. Note: The automatic corner is compiled for Windows. Linux users must compile it from source or use manual corner selection. The order of the selected corners does not matter as they are automatically reordered after selection.

3. Select planes in disparity images

A polygon is drawn to select the area of the image where the plane is visible. The order of the corners does not matter. If the image is a close up of the plane (Figure 3 left) the entire image should be selected.

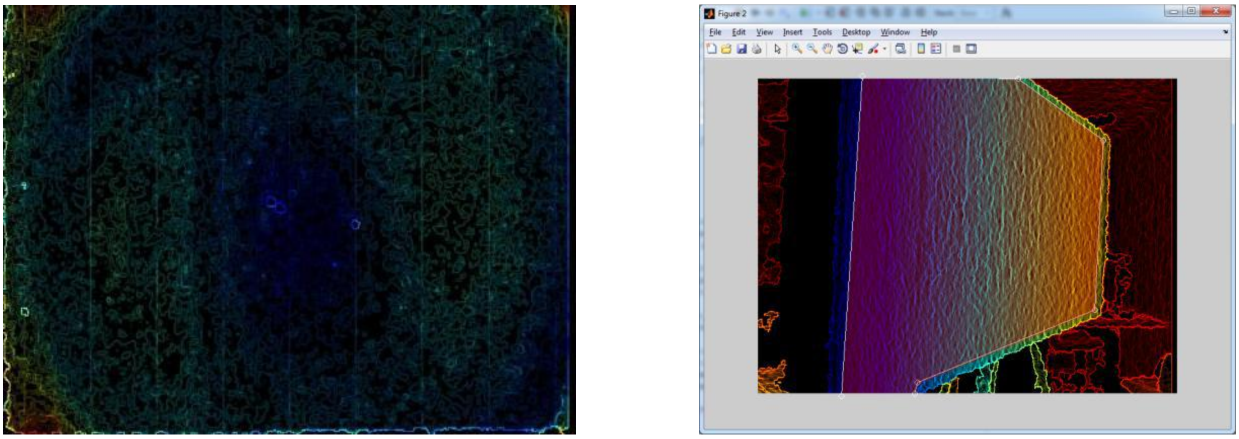


Figure 3 Sample plane selection images. **Left:** plane covers entire image. **Right:** Example of a selected polygon.

4. Perform initial estimation

The initial estimation is done by calibrating the RGB cameras individually and taking well-known calibration values for the depth camera. This gives an approximate calibration that serves as an initial guess for the non-linear minimization.

5. Calibrate

The final calibration is performed here. The disparity distortion can be enabled or disabled depending on your application. Calibration should not take too long.

6. Review results

The calibration results should not vary very much between Kinects. The values should make sense. The reprojection errors for the RGB cameras should be less than 1px and for the depth camera less than 1 kdu with distortion correction and less than 2 kdu without. The best way to check your calibration is to overlay the depth map over the color image. You can do this in Matlab using the compute_rgb_depthmap function or save the calibration in yml format (`save_calib_yml`), load it with the kinect_capture C++ application, and overlay the depth map on the color image in real time.

Important: There are many other useful functions for debugging and reviewing the calibration. Execute do_useful_fcn() to get a list of them.

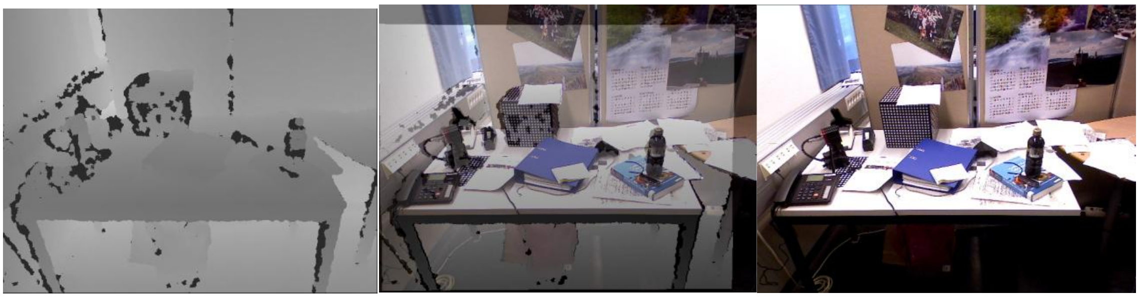


Figure 5 Overlayed depth map.

Results

With the sample data provided the obtained results are the following. Note that the calibration was performed using the high resolution (1280x1024) RGB inputs from the Kinect camera. The absolute value disparity distortion coefficients (dc_beta) are shown below.

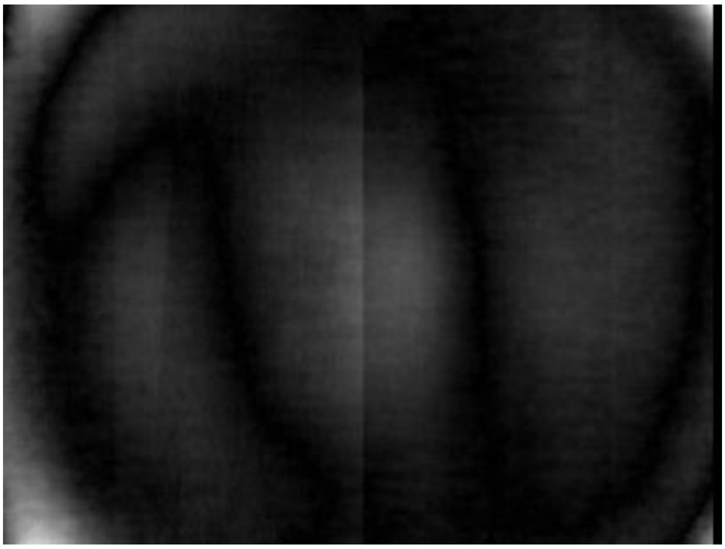


Figure 5 Estimated disparity distortion coefficients (dc_beta).

```
Calibration finished.
Color internals:
  Focal length:    [1060.27   1053.03]
                  ±[0.89   0.88]
  Principal point: [619.70   513.69]
                   ±[0.70   0.62]
  Distortion:      [0.2150   -0.6704   -0.0038   0.0012   0.6422]
                   ±[0.0029   0.0140   0.0002   0.0002   0.0187]
Relative pose:
  Rotation:    [1.00000   0.00000   0.00000;
                0.00000   1.00000   0.00000;
                0.00000   0.00000   1.00000]
  Translation: [0.00000   0.00000   0.00000]
              ±[0.00000   0.00000   0.00000]
Color internals:
  Focal length:    [4073.05   4047.75]
                  ±[3.54   3.52]
  Principal point: [1409.01   873.76]
                   ±[7.28   5.49]
  Distortion:      [-0.0680   -0.2043   -0.0028   0.0048   4.7656]
                   ±[0.0179   0.4464   0.0004   0.0005   3.1897]
Relative pose:
  Rotation:    [0.99933   0.00005   0.03657;
               -0.00023   0.99999   0.00488;
               -0.03657   -0.00488   0.99932]
  Translation: [-0.01107   -0.06884   -0.01690]
```

```
               ±[0.00002   0.00002   0.00023]
Depth internals:
  Focal length:    [579.83   586.73]
                  ±[0.51   0.50]
  Principal point: [321.55   235.01]
                  ±[0.42   0.42]
  Distortion:      [0.0000   0.0000   0.0000   0.0000   0.0000]
                  ±[0.0000   0.0000   0.0000   0.0000   0.0000]
  Depth params:    [3.12   -0.002855]
                  ±[0.003103   0.00]
  Depth distortion alpha:    [1.7907   0.0029]
                            ±[0.0295   0.0001]
Relative pose:
  Rotation:    [1.00000   0.00066   0.00260;
                -0.00063   0.99990   -0.01397;
                -0.00261   0.01397   0.99990]
  Translation: [-0.02308   0.00481   -0.00186]
              ±[0.00032   0.00034   0.00031]
Note: error is 3 times the standard deviation.
Color 1: mean=-0.000000, std=0.199017 [-0.009612,+0.010590] (pixels)
Color 2: mean=-0.000003, std=0.517421 [-0.026151,+0.028949] (pixels)
Depth  : mean=0.006969, std=0.797195 [-0.000702,+0.000703]
(disparity)
```

Using the calibrated model

To access to all the global variables used by the toolbox, run `global_vars()`. The final calibration parameters are stored in the global variable `final_calib`. The following functions are provided to easily use the calibrated model.

`[depthmap,points]=compute_rgb_depthmap(imd,calib,im_size,splat_size)`
Produces a depth map for the color camera given a disparity image (read by `read_disparity()`). The resulting depth map has resolution specified by im_size and each entry has the depth of the pixel in meters.

`Xw=disparity2world(u,v, disparity,calib)`
Takes a series of points in the disparity image and back projects them to the color camera reference frame in metric coordinates.

`p=project_points_k(X,K,kc,R,t)`
Takes points in metric coordinates and projects them onto the image plane.

Acknowledgments

Special thanks to the libfreenect community for their open source efforts. This toolbox includes some functions from Jean-Yves Bouguet's incredibly useful camera calibration toolbox and we are very thankful for his permission to do so. If you use this toolbox in your work please cite our paper, you know you want to.

Herrera C., D., Kannala, J., Heikkilä, J., " **Joint depth and color camera calibration with distortion correction**", TPAMI, 2012.

Questions, comments, and bugs to dherrera@ee.oulu.fi