# Lab Work 2: View + Depth 3D Capture

Elodie Charitat 291126
Jinhyeok Yoo 291234
Romain Husson 291135

| Tasks List | | |
|---|---|---|
| **Laboratory tasks** | L1. Collecting calibration data (Mandatory) | ✓ |
| | L2. Extracting the calibration parameters of the camera system (Mandatory) | ✓ |
| | L3. Capturing a set of still images (Mandatory) | ✓ |
| **Homework tasks** | H1. Estimating the 3D global coordinates from captured range data (Mandatory) | ✓ |
| | H2. Projecting 3D depth data to the 2D color camera plane (Mandatory) | ✓ |
| | H3. Re-sampling the projected data (Mandatory) | ✓ |
| | H4. Visualizing the combined depth/color data (Mandatory) | ✓ |
| | H5. Removing 3D model edge artifacts (+1) | |
| | H6. Mapping 2D camera color data to depth image plane and visualizing (+1) | |
| | H7. Z-buffering (+1) | |
| | H8. Occlusion handling (+1) | |

# Laboratory tasks

## L1. Collecting calibration data (Mandatory)

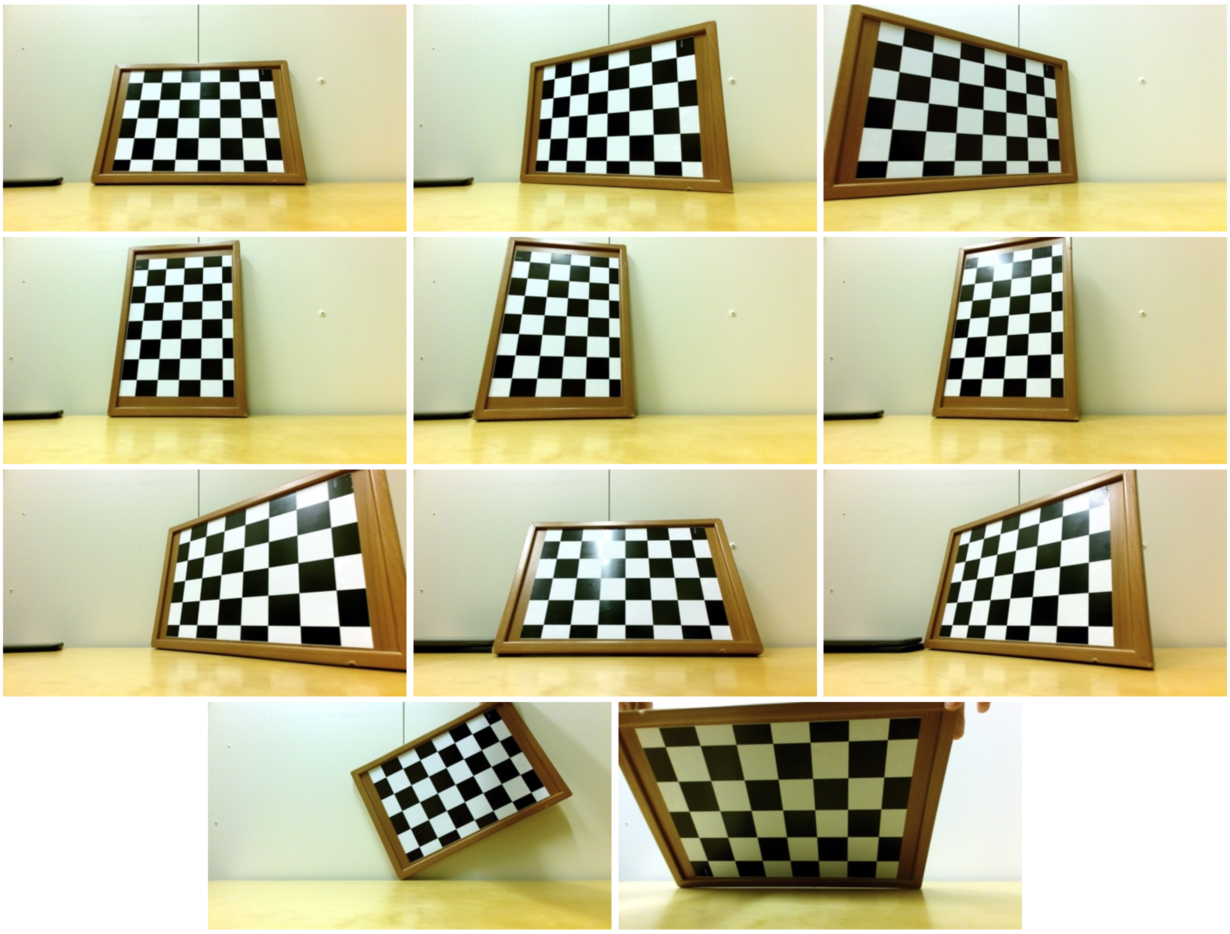
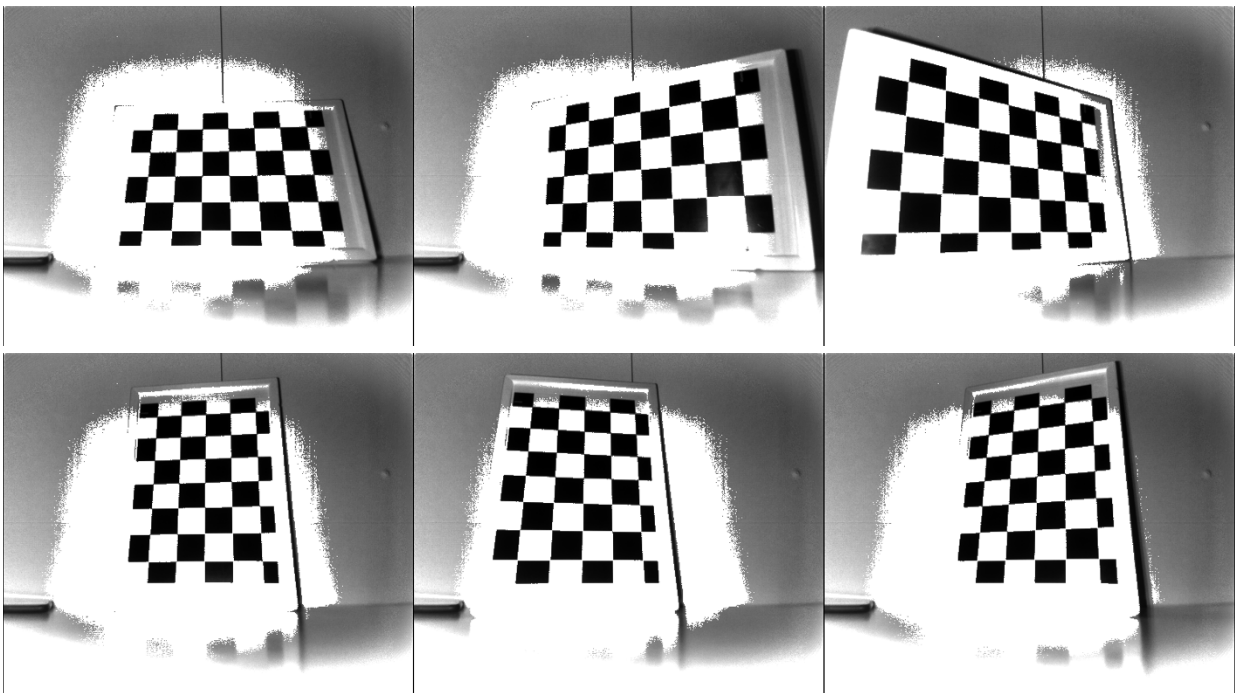


Figure 1 – 2D Color image of the checkerboard



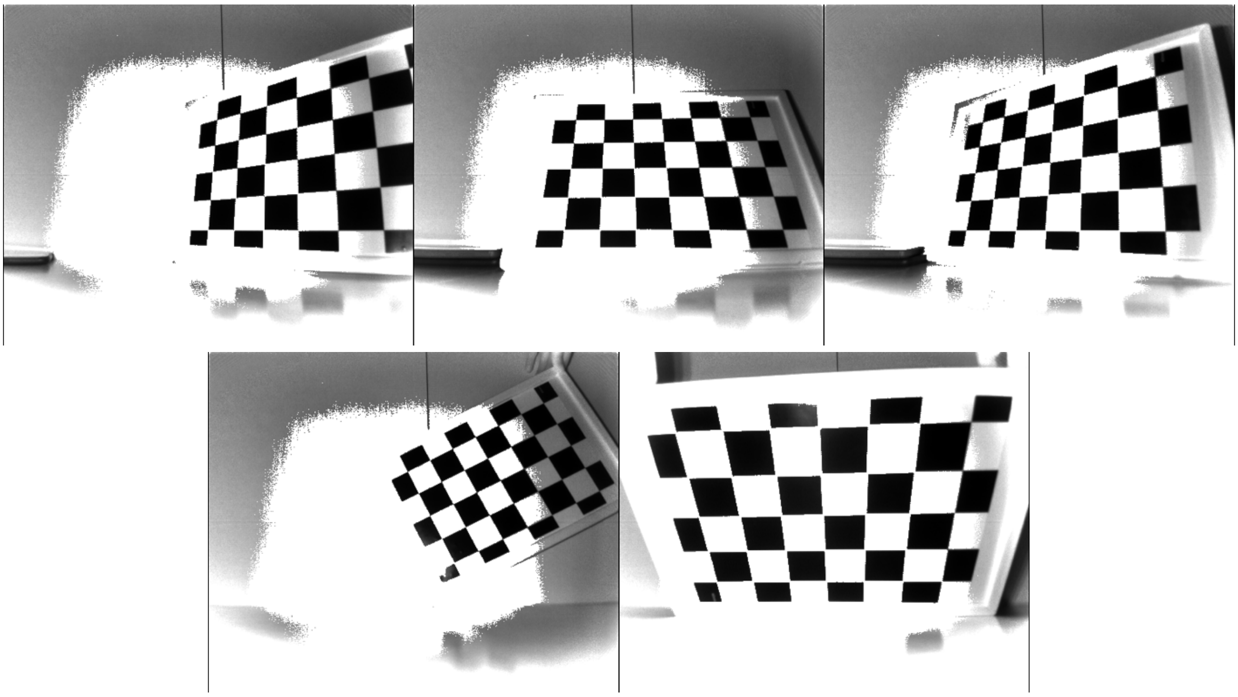


Figure 2 – IR image of the checkerboard

We captured 11 pictures (each 2D color and IR) of the different poses of the checkerboard. Some of the pictures were not suitable for use as data, so we removed them in next step.

## L2. Extracting the calibration parameters of the camera system (Mandatory)

We removed out-of-range and out-of-coordinate pictures before calibrating. And then we loaded the images and extracted the grid corners. We also tried to undistort the calibration images in order to verify that straight lines in real world remain straight in the undistorted image. Nevertheless, the overall error was not lower than the recommended value. We repeated the calibration process several times, but the error values did not go down. Inevitably, we performed the stereo calibration using the data from the lowest error. Finally, we got the file 'Calib_Results_stereo'.

## L3. Capturing a set of still images (Mandatory)

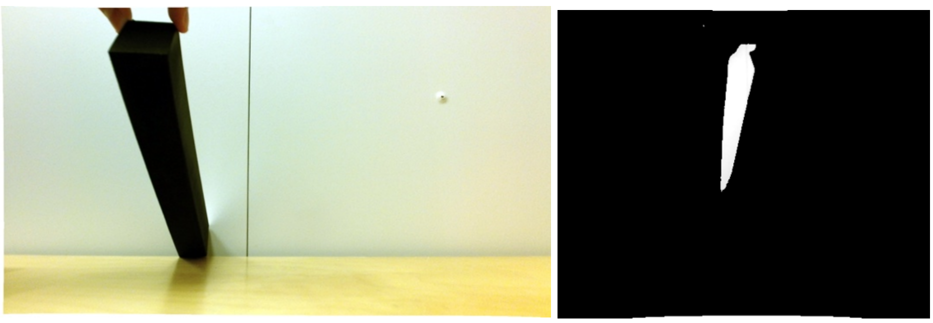


Figure 3 – a captured scene with Kinect 2.0: color image and depth image

We attempted to capture using a variety of objects in the lab, but few produced good depth images. We captured a black block, but it also did not produce a good image. We should have kept the proper distance and made sure the scene contains two different depth layers when capturing it. It was too late when we realized that this scene wasn't good for homework tasks. So, we just removed the distortion of the obtained image and got the Kinect data.

## H1. Estimating the 3D global coordinates from captured range data (Mandatory)

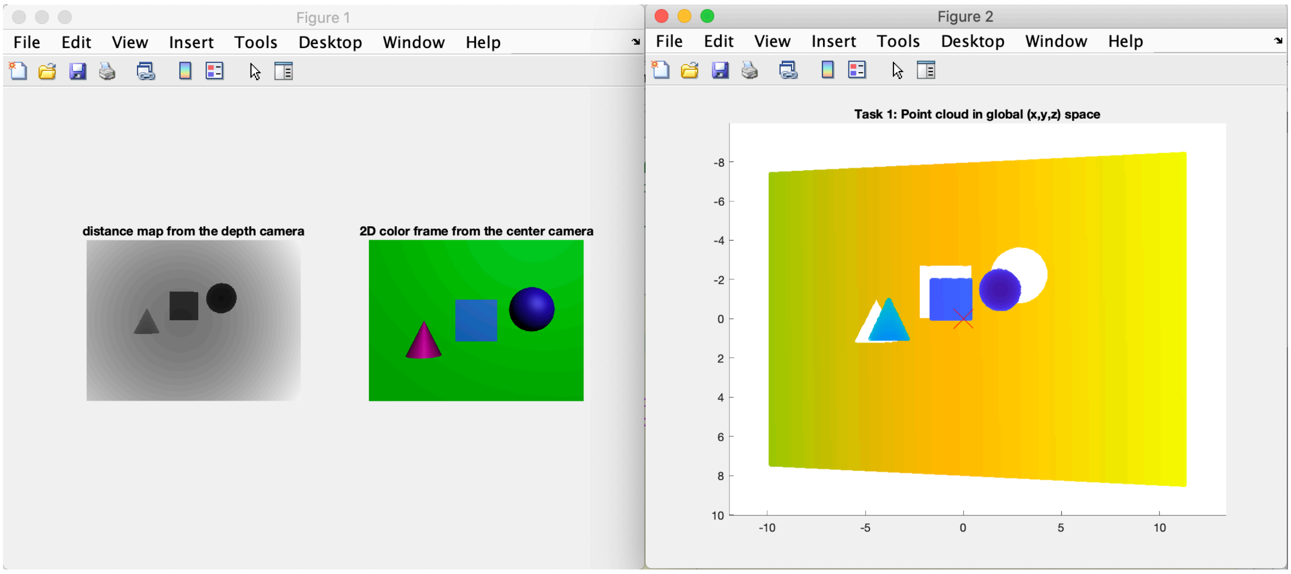



Figure 4 – a distance map and 2D color frame of the synthetic data, and its point cloud in global space

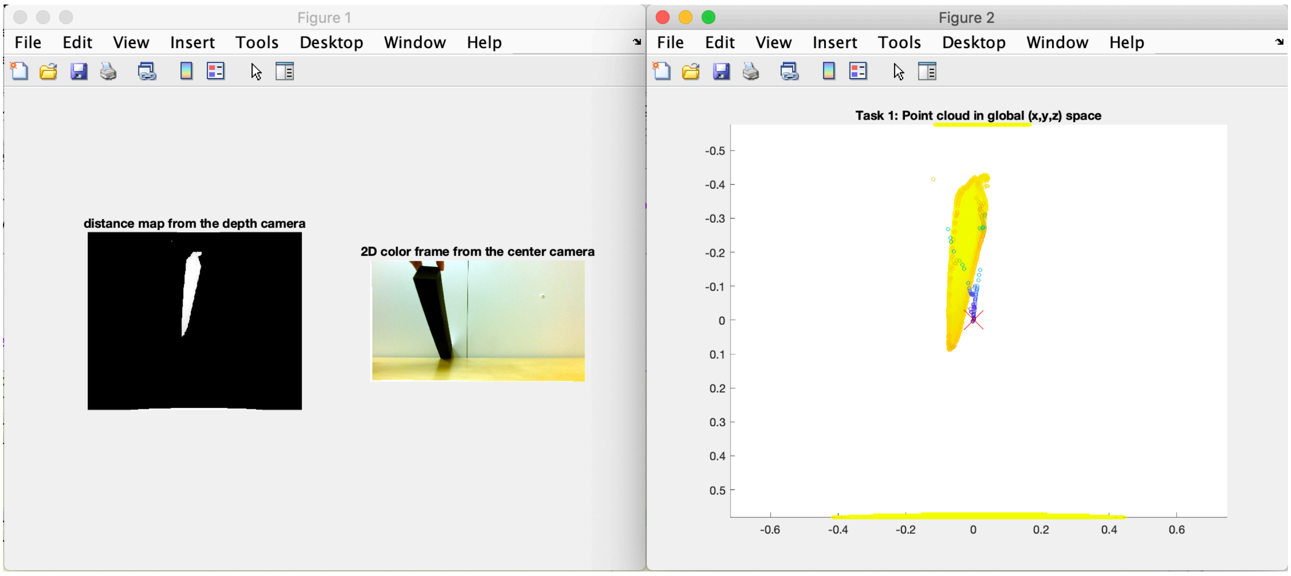



Figure 5 – a distance map and 2D color frame of the Kinect data, and its point cloud in global space

We listed all coordinates contained by the image without using the *meshgrid* function. We adjusted the center of the depth map to (0,0) and computed the global coordinates for each pixel. As Figure 5 shows, the result Kinect data is bad. As mentioned, we should made sure the scene contains two different depth layers when capturing it.

## H2. Projecting 3D depth data to the 2D color camera plane (Mandatory)

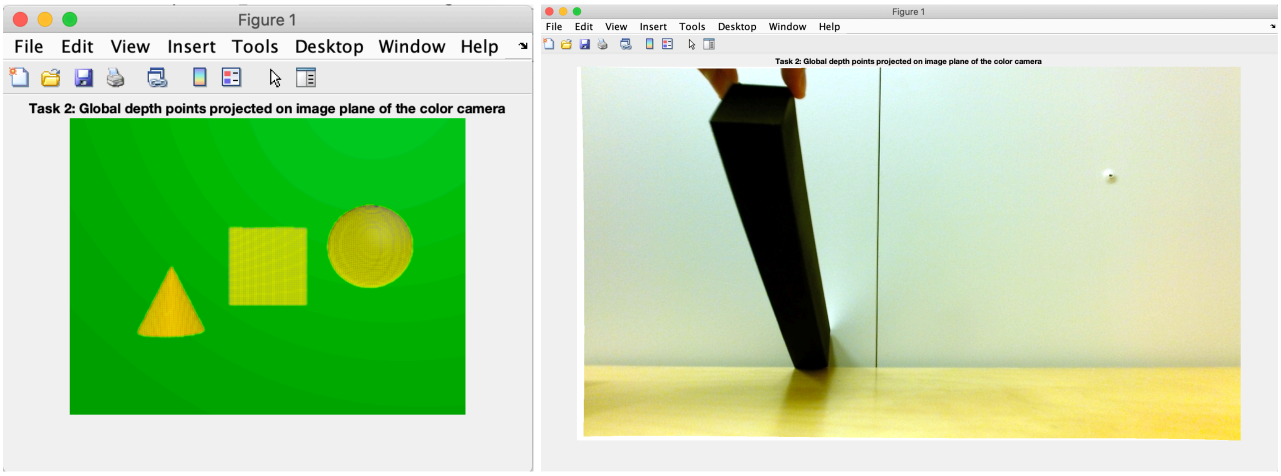



Figure 6 – global depth points projected on 2D color image of the synthetic data and the Kinect data

We applied the changes to the coordinates with R and T, moving them from the coordinate system centered on the depth camera to the color camera. And we projected the points from 3D space to the 2D image plane. Since the depth map value was not good at first, the global depth points were not displayed in the Kinect image plane.

## H3. Re-sampling the projected data (Mandatory)

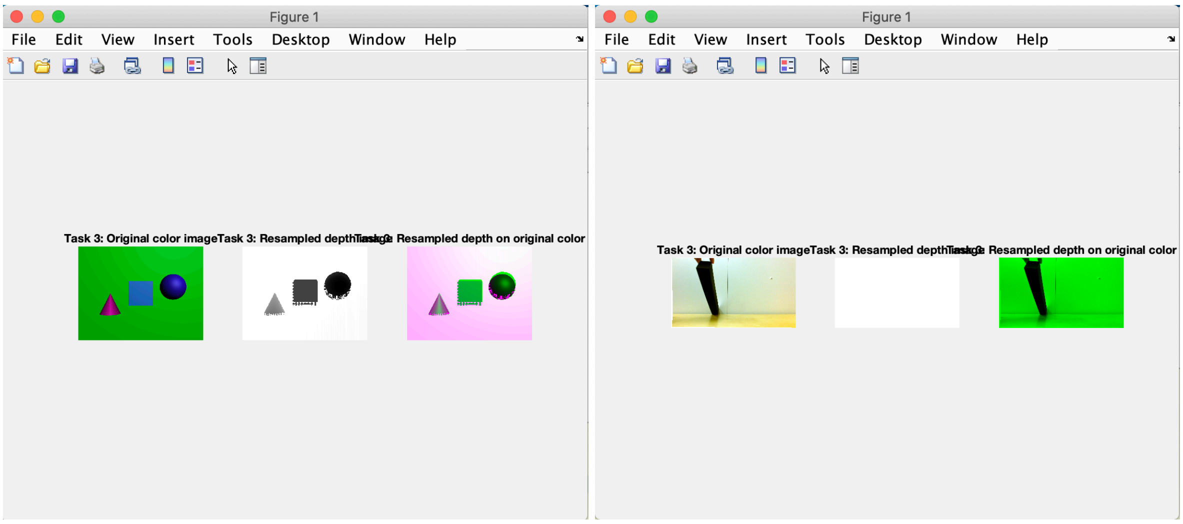



Figure 7 – resampling process of the depth images

Using nearest neighbor interpolation and the new z-coordinates, we re-sampled the projected data and plotted the resulting images. Since the depth map of Kinect data is not appropriate, resampling of Kinect data was not performed correctly.

## H4. Visualizing the combined depth/color data (Mandatory)

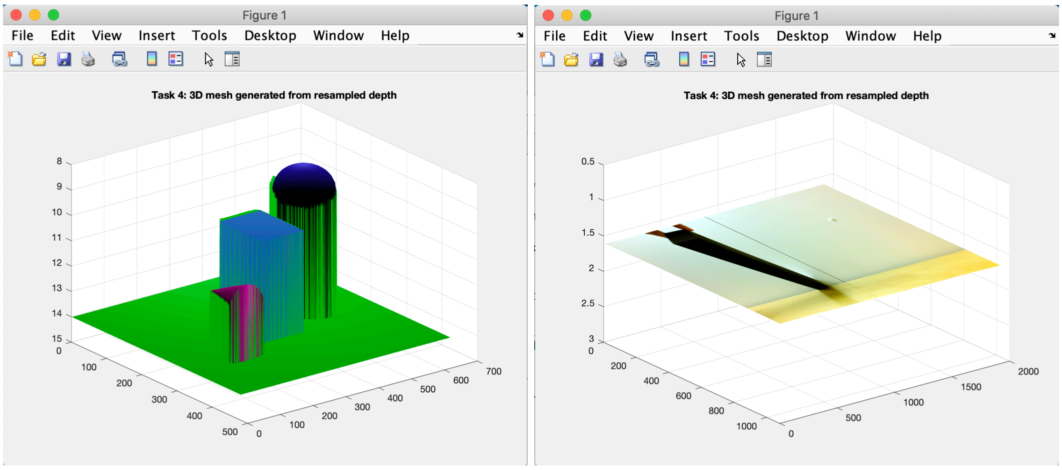



Figure 8 – 3D mesh generated from resampled depth

We visualized a textured surface using the projected, re-sampled data. The synthetic data was drawn correctly, but nothing was drawn on the 2D color image of the Kinect data because the resampled depth data was incorrect.

## Review

We implemented the algorithm correctly in all steps, but the result of Kinect data was not good because we captured a scene that was not suitable for the tasks with Kinect. The lab environment for good depth image capture was poor and we didn't have enough time to go back to the lab. If we had the correct depth map, we would have produced correct results of the Kinect data.