Sensors and Control for Mechatronics Systems
Tutorial 2

1. Camera Calibration

1. Follow the instructions on 'Getting Started' section of the follwing webpage (http://www.vision.caltech.edu/bouguetj/calib_doc/index.html) and download and setup the Camera Calibration Toolbox for Matlab.

2. Download the calibration image files provided in the learning materials.

3. Use the **calib_gui** command to start the calibration toolbox within the **cameraimages** directory. Pick the **Standard** option.

4. Click on **Image Names** and select the proper base name and image format for the downloaded images. If you have successfully imported images, you should now see a mosaic of all 10 images.

5. Click on **Extract grid corners** and select all images for corner extraction. Leave **wintx** and **winty** values as default. Use automatic square counting mechanism.

6. Select the four inner grid corners of the checker board. The first corner clicked will be used as the origin. Therefore, for all remaining images, click on the corresponding grid corner first. Set **dX** and **dY** values to 30. You will not need an initial guess for distortion in this dataset.

7. Continue to extract corners of the 9 remaining images using the same steps.

8. Click on **Calibration** to calibrate the camera. You should now see the instrinsic parameters of the camera on the Matlab interface.

9. Use **Reproject on images** to reproject grid corners. Observe the reprojection errors.

10. Use **Recomp. Corners** to recompute grid corners. Use the same parameters for **wintx** and **winty** and extract corners on all images.

11. Recalibrate the camera. Observe the reprojection errors. Comment on any changes with your previous observation.

12. Click on **Show Extrinsic** to view the extrinsic parameters estimated for each image.

13. **Save** calibration data. This will save calibration data as a .mat file. You can also **Export calib data** to export calibration data as text files.

14. Project 3D point (8,5,80) in camera coordinate frame to image coordinate frame (u,v) by writing a function.

2. Convolution

1. Write a Matlab function that takes a Greyscale image and a kernel as input and output the convolution between the image and the kernel.

2. Using the appropriate kernels introduced in the lecture and the Matlab function you implemented,
   i. Sharpen the *Sydney_Harbour_Bridge_from_Circular_Quay.jpg* image
   ii. Detect edges on *SydneyOperaHouse.jpg* image
   iii. Apply a Gaussian blur to *Ultrasound.png* image