Sensors and Control for Mechatronics Systems
Tutorial 2

Question 1 : Camera Calibration

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

1.2 : Download the calibration image files provided in the learning materials.

1.3 : Use the ***calib_gui*** command to start the calibration toolbox within the camera images directory. Pick the ***Standard*** option.

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

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

1.6 : Select the four inner grid corners of the checkerboard. 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.

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

1.8 : Click on ***Calibration*** to calibrate the camera. You should now see the intrinsic parameters of the camera on the Matlab interface.

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

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

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

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

1.13 : ***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.

1.14 : Transform 3D point (8,5,80) from camera coordinate frame to image coordinate frame (u,v) by writing a function.

Question 2: Convolution

2.1 : Write a Matlab function that takes a Grayscale image and a kernel as input, and outputs the convolution between the image and the kernel.

2.2 : Using the appropriate kernels introduced in the lecture,

i. Sharpen the ***Sydney_Harbour_Bridge_from_Circular_Quay.jpg*** image

ii. Detect edges on the ***SydneyOperaHouse.jpg*** image

iii. Apply a Gaussian blur to the ***Ultrasound.png*** image

Question 3: ROS USB_CAM Node

Setup the ROS ***usb_cam*** node. You may do this as a group activity, sharing the cameras.

3.1 : The computers in the lab should have ROS installed in the Ubuntu environment. Create a ROS workspace by following the 3rd step of the ROS tutorial "Installing and Configuring Your ROS Environment".
Link : http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment

3.2 : Checkout ROS usb_cam package to your Catkin Workspace from the source.
Link : https://github.com/ros-drivers/usb_cam
Use : ***git clone https://github.com/ros-drivers/usb_cam.git***

3.3 : Build the usb_cam package using ***catkin build***

3.4 : Plug in a USB camera and launch the usb_cam package using ***roslaunch usb_cam usb_cam-test.launch***

3.5 : List down the ROS topics available using ***rostopic list***

3.6 : Save images from the camera using ***rosrun rqt_image_view rqt_image_view***

3.7 : Explore the parameters listed down in the ROS wiki page of usb_cam. Change the brightness, contrast, saturation and sharpness parameters in the launch file and observe the effects. You can save images with different settings for comparison.
Link : http://wiki.ros.org/usb_cam