ECE592-064 Project 02

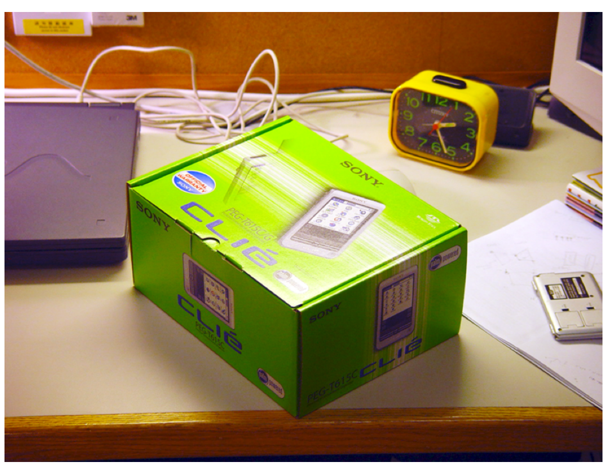
Due 04/09/2017

How to submit your solutions: put source code folder [your_unityid_code], your report (word or pdf) and results images (.png) in a folder named [your_unityid_project02] (e.g., twu19_project01), and then compress it as a zip file (e.g., twu19_project02.zip). Submit the zip file through **moodle**.

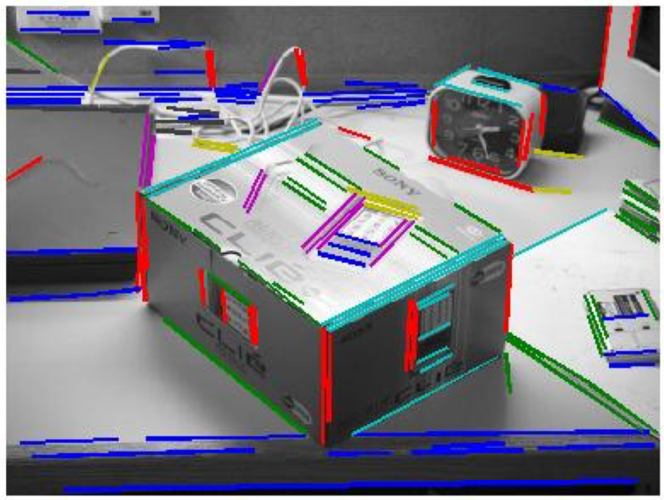
Submitted as a team: You can form a team with at most 3 members (including yourself). If you have a team, in the report, you need to clearly state the contribution of each member.

Project description: This project focuses on improving the implementation of the paper "single view metrology" (Criminisi, Reid and Zisserman, ICCV99) in Project01. It substitutes the manual annotation of vanishing points with a detection algorithm based on a line segment detector (http://www.ipol.im/pub/art/2012/gjmr-lsd/) and the RANSAC algorithm. You are encouraged to explore other methods of detecting vanishing points automatically. Basically, this project includes the following 4 subtasks:

- *Image acquisition (same as project01)*: Take a photo by yourself following the guide of 3-point perspective image (http://en.wikipedia.org/wiki/Perspective_(graphical)#Three-point_perspective) as input. E.g., you can take a photo of a box on your desk which looks like,

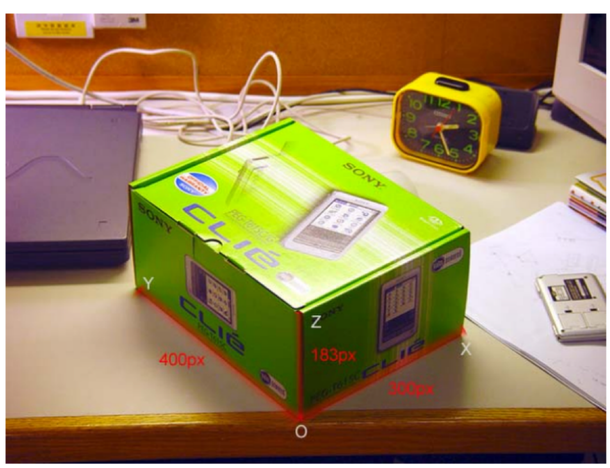


- *Computing vanishing points (new component to be developed in this project)*: You can run LSD to compute the line segment, and then compute the vanishing points based on RANSAC. Or, you can explore other methods.



For example, in your RANSAC, you can still utilize the method proposed by Prof. Robert. T Collins: https://kusemanohar.files.wordpress.com/2014/03/vanishingpt_bobcollins.pdf .

- *Computing the projection matrix and homograph matrix (same as project01)*: Based on the computed vanishing points, the projection matrix can be computed following the method discussed in lecture 06. However, the columns are up to a scaling constant. Thus you need reference points to evaluate the scaling constants. E.g., you can measure the reference points and distance in the reference direction as illustrated below,

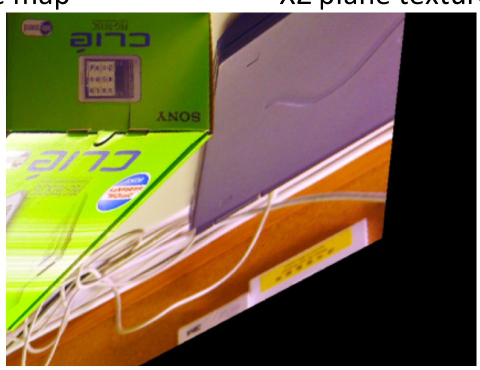


Once you computed the projection matrix you can obtain the homography matrix (H) corresponding to each of the normal planes (ie. XY, YZ, XZ). This can be done (for Hxy) by taking the 1st, 2nd and 4th column of the projection matrix. Similarly, Hyz, Hxz can be calculated.

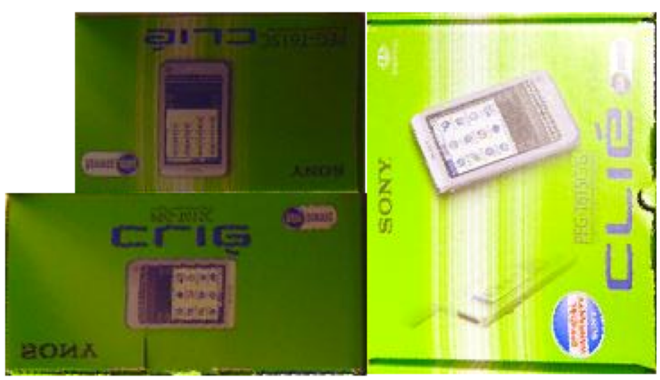
- *Computing the texture maps for XY, YZ and XZ planes respectively (same as project01)*: It is done by transforming the original image using the Homograph matrix computed above. E.g.,

XY plane texture map

XZ plane texture map



YZ plane texture map



Cropped portion of the object

Then, you can crop the required portion of the object in the texture maps.

- *Visualizing the reconstructed 3D model (same as project01)*:. After you have a set of texture maps and their corresponding 3D coordinates, you can create a .wrl file to build up the VRML file. The 3D model can be visualized on the browser. You can find many resources on VRML(e.g., [http://openvrml.org/)](http://openvrml.org/). E.g.

Or, you can use Blender to visualize the 3D model ([https://www.blender.org/)](https://www.blender.org/).

Requirement: Your main code should be self-contained: Given an input image and the annotated information (lines, reference points and distance, etc), it will generate the 3D model.