

ECE592-064 Project 01

Due 03/10/2017

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_project01] (e.g., twu19_project01), and then compress it as a zip file (e.g., twu19_project01.zip). Submit the zip file through **moodle**.

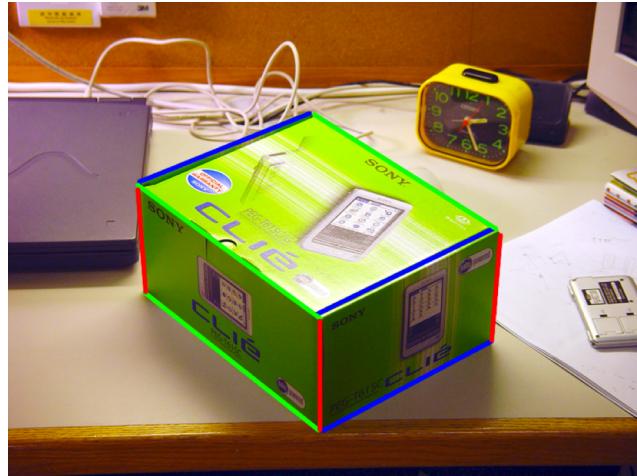
Submitted as a team: You can form a team with at most 3 members (including yourself). If you have a team, in the report, you need to clearly state the contribution of each member.

Project description: This project focuses on an implementation of the paper “single view metrology” (Criminisi, Reid and Zisserman, ICCV99). It describes how aspects of the affine 3D geometry of a scene can be computed from a single perspective image with some prior knowledge. Basically, this project includes the following 4 subtasks:

- *Image acquisition:* Take a photo by yourself following the guide of 3-point perspective image ([http://en.wikipedia.org/wiki/Perspective_\(graphical\)#Three-point_perspective](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 vanish points:* First, you need to write a simple annotation interface to draw several projected lines of real-world parallel lines in the image. You need at least 2 lines for each of the X, Y and Z directions, as illustrated by the lines in red, green and blue in the figure below.



Second, to compute the vanishing points, you can follow 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.* 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 H_{xy}) by taking the 1st, 2nd and 4th column of the projection matrix. Similarly, H_{yz} , H_{xz} can be calculated.

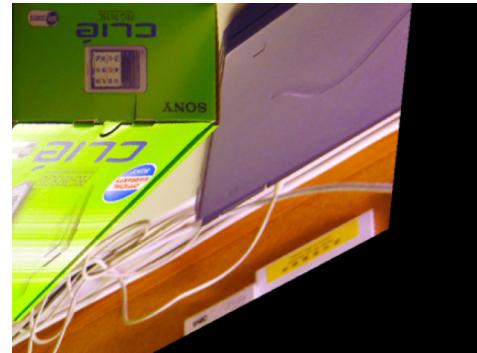
- *Computing the texture maps for XY, YZ and XZ planes respectively.* It is done by transforming the original image using the Homograph matrix computed above. E.g.,



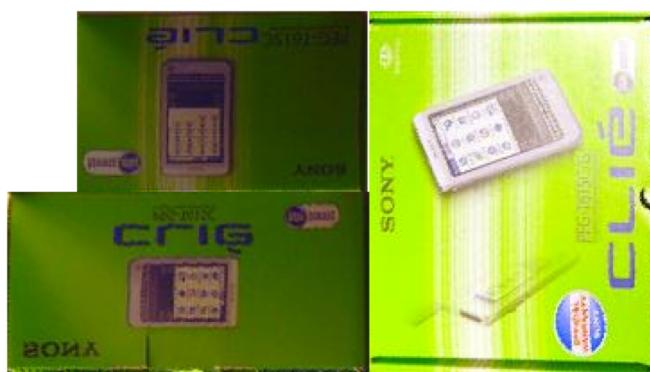
XY plane texture map



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.* 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/>). E.g.



Or, you can use Blender to visualize the 3D model (<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. The annotation information can be labeled by different interfaces which do not need to be integrated into your main code. The annotation tool can be shared between different teams if some are not very familiar with how to write a simple labeling tool. But, the main code MUST be implemented independently between different teams.