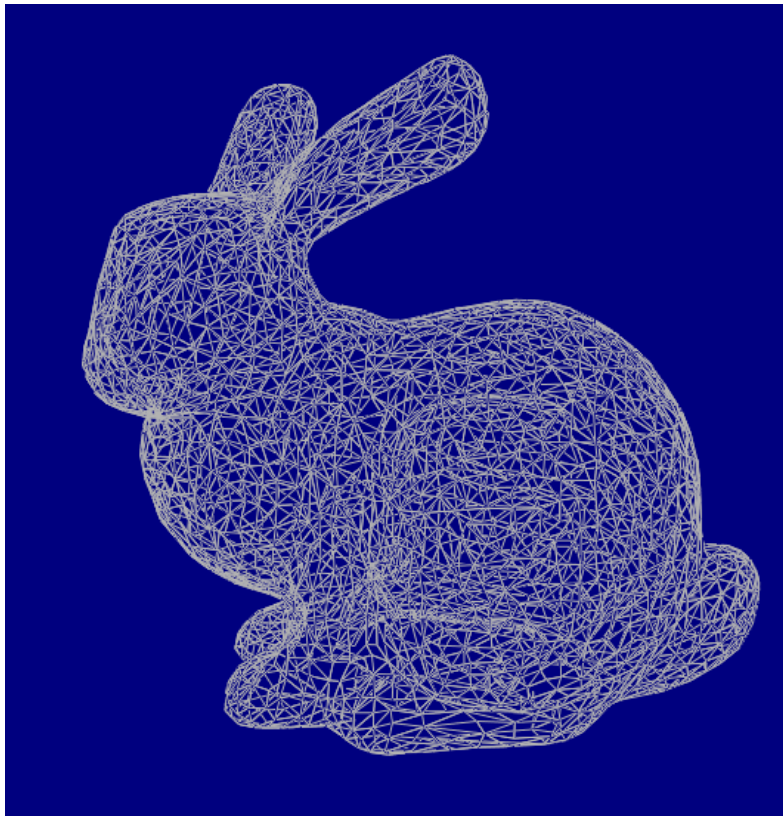


Visual Computing Programming Assignment

Due: 13 Nov



It is highly recommended to use Matlab for this assignment but if you still believe that it would be easier for you on Python, you can still do that but you are on your own for finding the equivalent functions in Python.

Goal: To calibrate the camera positions and extract the unknown geometry

Please download the zipped directory. This directory contains following folders:

1. **Data:** In this directory, you have been provided with images taken from multiple view points of a small object and the calibration pattern data. The file calibration.doc provides some of the essential 3D coordinates for blocks' corner points that you could see in the 2D images.
2. **vlfeat:** This directory contains the binary distribution of VLFEAT toolbox for Matlab. Using this toolbox you can extract and match SIFT features in the images.
3. **code:** In this directory we provide you the code for some required functions such as calculating the fundamental matrix.

Programming Assignment:

Choose 2 subsets of these images – each comprising of 3-6 images. Make sure that each of these sets is chosen such that every part of the object is viewed by at least 2 images. You can choose more images if you want – an efficient implementation could improve results with increasing number of images. The assignment is to write a program to (a) recover the camera calibration matrix and (b) extract the depth of the object and then test your program on the 2 subsets of images you have created.

Please consider the expecting time for each part and try to finish each part by that time.

Part 0: Installing vlfeat toolbox

Vlfeat must be added to Matlab search path by entering following command in Matlab prompt:

```
run('vlfeat/toolbox/vl_setup')
```

Run following command to make sure the toolbox has been installed:

```
vl_version verbose
```

You can also find the list of all Matlab commands in the following link:

<http://www.vlfeat.org/matlab/matlab.html>

Part 1: Extracting and matching features between images (20 points) (2 days)

Extract the SIFT features in all the images and find the matches between each pair of images. Sort the matches based on their scores and remove all the matches with very low scores. This will remove some of the wrong matches.

We also provided a mask image for each picture in the dataset. In the mask image the background is black and all the pixels which belongs to the object are white. Using the mask for each picture remove SIFT features which are not on the object.

Part 2: Finding the camera calibration matrix (30 points) (4 days)

Use the calibration pattern data to generate the 3D to 2D correspondences for each image. Use this data to find the camera calibration matrix from each image.

Part 3: Refining the matching features (20 points) (2 days)

At this step you can use the correspondences from calibration patterns in order to calculate the fundamental matrix between each pair of images. Using the fundamental matrix find the epipolar line for each SIFT feature in other images. Now for each pair of images you can remove all the matched features that are not on the epipolar line of corresponding feature. This will remove incorrect matched points in each pair of images.

Part 4: Finding Depth (30 points) (4 days)

Use a multi-view formulation for the depth reconstruction process. You may not find the same 3D point in all the images in your set of images, so you may have to select the views where the particular point is visible. This would provide a nice 3D point cloud of the object.

Deliverable

As deliverable prepare a document which provides the following for each of the 2 subsets you have used.

- All the pictures used in the subset.
- Show the top ten matching features for one pair of images.
- Show 3 examples of the epipolar line by showing the feature from one image and the corresponding line in the other images.
- Plot the position and orientation extracted from each of the camera calibration matrix to show the camera positions from which the pictures were taken. For showing orientation plot the three axes of the camera's local coordinate system.
- In the same plot, plot the 3D points recovered from this set of pictures by your depth estimation method. Extract around 150 point for the horse, this will give you a nice dense point cloud representation of the object.

Finally, create one plot where you plot the 3D points recovered from one set of images in red and the 3D points recovered from another set of images in blue. If you have done everything

accurately, you should see only one object even after merging the data from multiple image sets. Plot the camera positions and orientations from the two subsets using the distinguishing colors of red and blue in the same plot.

Create a pdf report with plots to show that you have implemented each part and got acceptable answers. Submit this pdf on Gradescope, **one pdf for each group**.