Lab 4 (Weeks 10,11): SIFT Matching and Stereo Reconstruction

This lab exercise is worth 7.5% of your final unit grade. *A task is only considered complete if you can demonstrate a working program and show understanding of the underlying concepts. Note that later tasks should reuse code from earlier tasks.*

In this laboratory exercise, you will create a program that matches SIFT keypoints by comparing their descriptors. The matches should be visualised on the corresponding input image. Images and their SIFT keypoints are provided as starting resources for this exercise.

References:
Euclidean distance: http://en.wikipedia.org/wiki/Euclidean_distance
General SIFT info: http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

Resources:
Input images: im1.jpg, im2.jpg,im3.jpg,im4.jpg
SIFT keypoints of images above (ASCII): im1.sift, im2.sift,im3.sift,im4.sift

Task 1 : Load and join images (0%)

Load the input images im1.jpg, im2.jpg and join them horizontally into a single output image ([image1, image2]). Show the resulting “double-wide” image in a new figure.

Tasks 2 and 3: Read and show SIFT keypoint locations (1.5%)

The SIFT keypoints are stored as ASCII text files (*.sift). Each line of text contains the following numbers:
`<x> <y> <scale> <orientation> <Descriptor vector values x128>`

Write a program that reads the data in the .sift files(im1.sift & im2.sift) and draws the key points onto the figure you created in Task 1. Draw the keypoint locations (x,y) on the output image as red crosses. There is no need to visualise the scale, orientation or descriptor vector values.

HINT: Start with a few keypoints and validate their correctness before trying to draw everything.

Tasks 4 and 5: Match SIFT keypoints and show matches(1.5%)

SIFT keypoints are matched by comparing their 128-value descriptor vector. The Euclidean distance between two descriptor vectors gives an indication of their similarity (shorter distance is more similar). To ensure only good matches are returned when matching keypoints between 2 images, the following algorithm is used for matching keypoints from im1 to im2

```
for each keypoint k in image1
      compare k against all keypoints in image2 using Euclidean distance
      Find the nearest and 2nd nearest match (distances d1 and d2)
      if d1/d2 < 0.5
            match is valid
      else no match found for k
```

Draw all valid matches (comparing image1 to image 2) in **green** with straight lines joining the matching keypoints.

Question: What happens if the nearest match is always used as a valid match instead of the ratio-of-distance metric used in the algorithm above?

Task 6: Stereo reconstruction (3.5%)

For Parts 1 & 2 assume a focal length & and a baseline of 1. Triangulate the key points based on the disparity of the two matching views.

Part 1

im2.jpg is obtained by translating the camera sideways from im1.jpg.Use the principles of parallel cameras you learnt in the lectures to triangulate the depth of each keypoint.Plot the reconstructed 3D points using a scatter plot.

Part 2

Repeat Tasks 1-6[Part 1] using the image pairs <im1,im3> and <im1,im4>. Plot all three reconstructions on the same scatter plot (Use different colours to denote the 3D points obtained from different image pairs).Explain the findings by quantifying the results.

Part 3

If the actual length of the box is 37 cm approximate how wide the baseline is for each image pair. The method you use should be non-trivial(i.e you should not manually select points on an image based on visual inspection)

Task 7: Reprojection of 3D points(1%)

Reproject the triangulated 3D points that you obtained using image pair <im1,im4> on to im3.jpg
HINT: Think about the different baselines of the image pairs.