

CSC420 Intro to Image Understanding Assignment 4 - Fall 2019

Post Date: November 19 Due Date: November 28 , 10:59 pm

Instructions for submission: Please write a document in a pdf or doc file with your solutions (include images where needed), and submit through MarkUS. Please include your code and specify the question to which it corresponds.

1. Stereo Matching Costs

- (a) **[0.5 point]** Imagine that two images for stereo matching are captured by the same camera but under different exposure times. Compare the two main cost functions for matching, i.e. SSD and NC, for estimating the correspondences in these two images. Which one performs better and why?

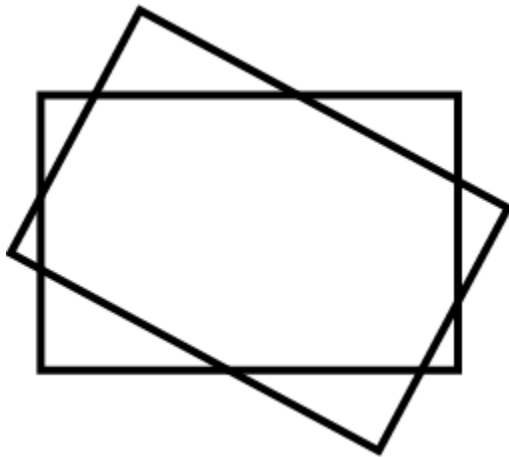
2. Stereo Matching Implementation - for this question, use the rectified image pair (000020_left.jpg and 000020_right.jpg), the given bounding box in file 000020.txt, and the given parameters in file 000020_allcalib.txt.

- (a) **[2 points]** Write a program to compute the depth for each pixel in the given bounding box of car. Use the algorithm given in class, where given a left patch, compare it with all the patches on the right image's scanline. To reduce computation complexity, you can try to use a small patch size, or sample patches (e.g. every other pixel) from scanline instead of comparing with all possible patches. Report what is the patch size, sampling method, and matching cost function you used. Use the parameters given and show how depth is computed for each pixel. Also visualize the depth information. Are there any outliers coming from incorrect point correspondences?
- (b) **[2 points]** After you compute depth using scanline (above), go to KITTI Stereo 2015 (http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo), and pick a machine learning model from the scoreboard. You can pick a model that comes with code and a pre-trained model so you don't need to implement yourself. Compute the depth for the whole image using their pretrained model. Compare your results from the previous questions with this results from this model. What is the difference, both in terms of quality and speed?
- (c) **[1 point]** Write a short summary of workflow for the model you pick, e.g. what layers do they use, what modules do they use, etc.

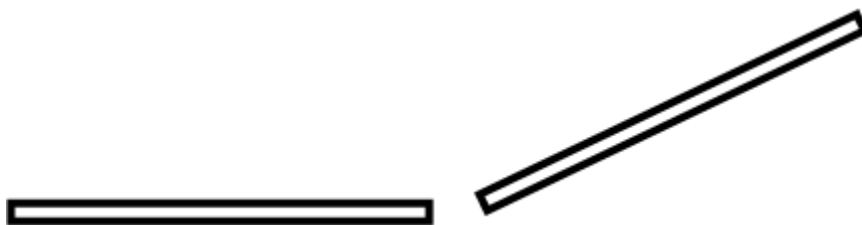
- (d) **[1.5 points]** Using the depth information from models, try to determine within the bounding box, which pixel belong to the car based on its distance to the box center pixel's 3D location (use a threshold). After that, try to determine a 3D bounding box for the car (min & max along X, Y, Z). Visualize the segmentation of pixels within the 2D box, and on another image visualize the 3D box. State the classification threshold for distance.

3. Fundamental Matrix - for this question, take 3 images (I_1 , I_2 , I_3) of an object or a stationary scene as follows: Images I_1 , I_2 from almost the same viewpoint, but with a (roll) rotation. That is, take an image (I_1), don't move, but rotate the camera in place around 30-45 degrees, and take another image (I_2). Take the third image (I_3) from a different viewpoint, e.g. move the camera ~20 cm to the right and rotate (out of plane) to point the camera towards the object again.

I_1 and I_2 :



I_1 and I_3 (top view):



- (a) **[1 points]** Use SIFT matching (or any other point matching technique) to find a number of point correspondences in the (I_1 , I_2) image pair and in the (I_1 , I_3) image pair. Visualize the results. If there are any outliers, either manually remove them or increase the matching threshold so no outliers remain. Pick 8 point correspondences from the remaining set for each image pair, i.e. (I_1 , I_2) and (I_1 , I_3). Visualize those 8 point matches. It helps in the

later steps if the 8 point matches are somewhat distributed over the images rather than being clustered in a small region.

- (b) **[1 point]** Using what we have learned in class (standard 8-point algorithm) calculate the fundamental matrix F_{12} for image pair (I_1, I_2) and Fundamental matrix F_{13} for image pair (I_1, I_3) . (for this question, implement your own standard 8-point algorithm)
- (c) **[1 point]** Using F_{12} , calculate the epipolar lines in the right image for each of the 8 points in the left image and plot them on the right image. (for this question you can use any OpenCV functions you want)
- (d) **[1 point]** Using F_{12} , rectify I_2 with I_1 and visualize the resulting image side by side with I_1 . Do the same for I_3 using F_{13} . (for this question you can use any OpenCV functions you want)
- (e) **[0.5 point]** Using OpenCV, compute F'_{12} and F'_{13} and compare with your results. I.e. are they same? Are they similar? Briefly discuss.
- (f) **[0.5 point]** Using OpenCV, rectify the images using F'_{12} and F'_{13} and compare with your rectifications (part d). Discuss any differences.