

University at Buffalo
Department of Computer Science and Engineering
CSE 473/573 - Computer Vision and Image Processing

Fall 2020

Project #2

Due Date: Nov. 09 2020, 11:59PM

1 Image Stitching (7 points)

The goal of this task is to stitch two images together to construct a panorama image. Please follow the following steps to complete the task. (1) find keypoints (points of interest) in both images, e.g., Harris detector or SIFT point detector. (2) use SIFT feature descriptors to extract features for these keypoints. (3) match the keypoints between two images by comparing their feature similarities. (4) compute the homography matrix using RANSAC algorithm. (5) use the homography matrix to stitch the two images. Figure 3 shows an exemplar result, while Figure 4 is the failed case. Note Figure 1 and Figure 2 are just examples, and the two images you need to stitch are provided in the .zip file.

Guidelines You can use APIs provided by numpy and opencv to complete the task, except for APIs that have “stitch” or “Stitch”, “match” or “Match” in their names, e.g., “cv2.BFMatcher()” and “cv2.Stitcher.create()”. You will be provided with a “task1.py” file and two original images. Please strictly follow the format in the “task1.py” and **do not** modify the code format provided to you. You need to provide the output panorama image “task1_result.jpg” after image stitching in the results folder.



Figure 1: Left Image



Figure 2: Right Image



Figure 3: Successful example



Figure 4: Failed example

2 Hough transform (8 points)

You are given an image ‘hough.jpg’ shown in figure 5, please design and implement an algorithm to detect lines and circles.

- Utilizing edge detection and Hough transformation to detect all vertical lines (red lines), save the result image name as ‘red_lines.jpg’ and the line parameters in ‘red_lines.txt’. (2 points)
- Detect all the diagonal lines (blue lines), save the result image as ‘blue_lines.jpg’ and the line parameters in ‘blue_lines.txt’ (3 points)
- Detect all the coins in the image, save the result image as ‘coins.jpg’ and the circle parameters in ‘coins.txt’. (3 points)

Guidelines

- For all the red and blue lines detected, you should report the θ and ρ parameters in ‘red_lines.txt’, ‘blue_lines.txt’ files respectively. For each of the detected line report θ and ρ considering line is represented by equation $x\cos\theta + y\sin\theta = \rho$. (Please refer slide 60 in Image Analysis slides in piazza). Each line in the .txt file will contain parameters of a single line in $[\theta, \rho]$ format. You can assume that the resolution of ρ is 1 pixel and resolution of θ is 1 degree.
- For all the detected circles, please report c_1, c_2, c_3 in the ‘coins.txt’ file, considering that a circle is represented by equation $(x - c_1)^2 + (y - c_2)^2 = c_3^2$. (Please refer slide 62 in Image Analysis slides in piazza). Each line in the txt file will contain parameters of a single circle in $[c_1, c_2, c_3]$ format.
- File naming conventions and the format of *.txt files needs to be strictly followed.

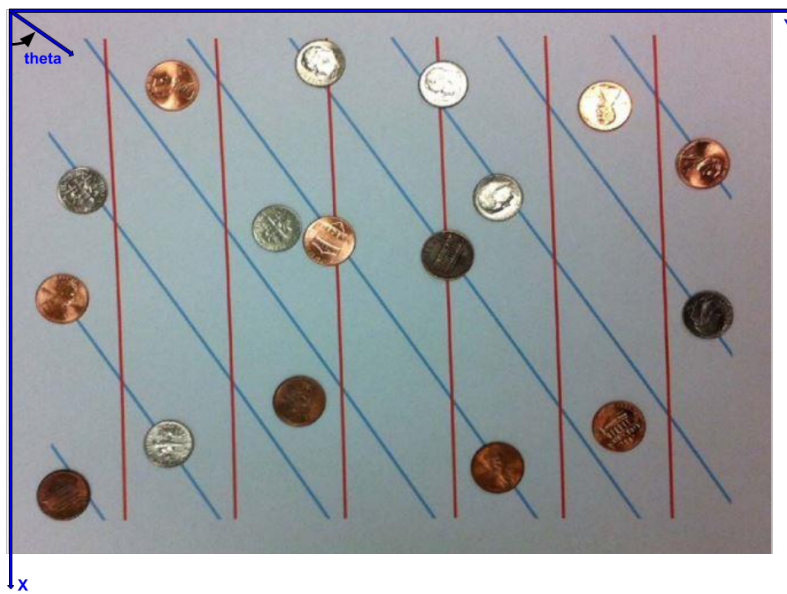


Figure 5: Hough.png

Submission Instructions:

- For the image stitching project, you need to submit “task1.py” or “task1.cpp” together with the stitched image “task1_result.jpg” placed in results folder provided. We recommend you to create “task1.cpp” based on “task1.py”.
- For the Hough transform project, you need to submit “task2.py” or task2.cpp, together with the detection results “red_lines.jpg”, “blue_lines.jpg”, “coin.jpg”, “red_lines.txt”, “blue_lines.txt”, “coin.txt” in the results folder provided, where the .jpg files should include the detection results while the .txt files should include the detected line/circle parameters.
- Compress the above mentioned files into a zip file and, name it as “UBID.zip” (replace “UBID” with your eight-digit UBID, e.g., 51399256) and upload it to UBLearns before the due time.
- Identical code will be treated as plagiarism. Please work it out independently.
- For codes raising “RuntimeError”, the grade will be 0 for this task.
- Late submissions within 24 hours are allowed and will result in a 40% penalty. After one day, submissions will not be accepted.