CmpE 537 Computer Vision

Assignment II - Image Stitching

uras.mutlu@boun.edu.tr

Due November 7, 2019 23:59

1 Problem

In this assignment, you are going to develop an image stitching system that can stitch multiple images in order to create a single panoramic image. You can see the example below. You will use python3 to implement this assignment.

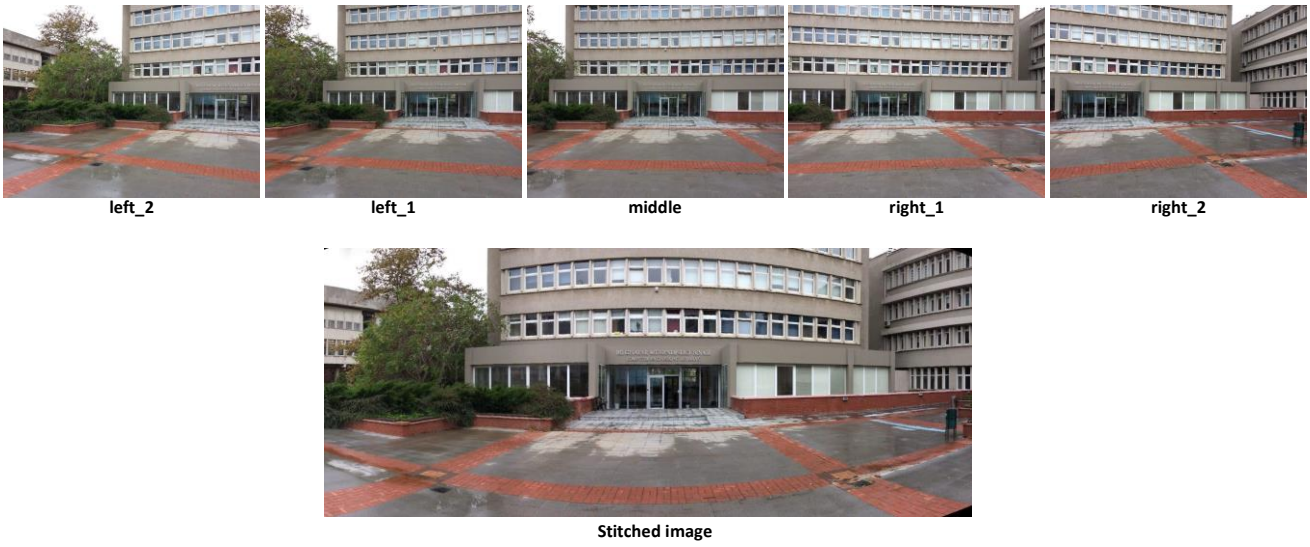

Figure 1: An example from CMPE building

2 Image Stitching Procedure

2.1 Select Common Points on Images

In order to stitch two images successfully, first you need a good selection of corresponding points between your input images. In your solution, you can use matplotlib.pyplot.ginput to select points by mouse clicks. Here is a link to ginput. You can use any other function if you find one. Note that automated interest point detector algorithms are forbidden such as SIFT, SURF or Harris.

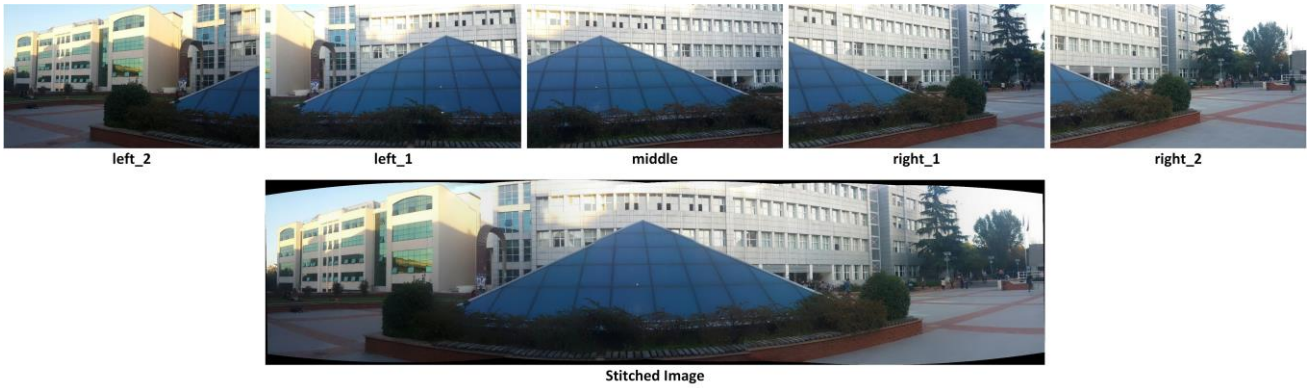

Figure 2: An example from Kuzey Kampus

2.2 Homography Estimation

Write a function that can compute homography between two images using the corresponding points that you have selected before. In your assignment, the signature of the function should be:

```
computeH(im1Points,  im2Points)
```

2.2.1 Normalized Points

Normalize points before calculating the homography. Move the average to (0, 0) and make the average length $\sqrt{2}$ for each image separately. Let us say x' is normalized points and T is normalizer transform. Hence, $x' = Tx$. So, you can derive H from $T_2x_2 = H'T_1'x_1$ and $x_2 = Hx_1$

In the signature, `im1Points` and `im2Points` are matrices where each column represent a corresponding point. In your assignment, you can use numpy svd method for solving equations.

2.3 Image Warping

After estimating the homography between two images, you should be able to warp the images using backward transform and the homography matrix that you have obtained in the previous step. In your assignment, the signature of the function should be in form:

```
warp(image, H)
```

You are supposed write your own warping function. You are not allowed to use any library that handles the task completely. However, you are allowed to use interpolation functions of numpy, scipy or any other library. If you want to write your own interpolation function, it will be rewarded. Note that it will be difficult, hence it is suggested you to try it after completing all the tasks.

2.4 Blending Images

Finally, you have the warped images that you can merge and show as a single mosaic/stitched image. If you directly put images on top of each other, every pixel you write will overwrite the previous image's pixels. For the areas that overlap use the image pixel that has the maximum intensity.

Finally, you have the warped images that you can merge and show as a single mosaic/stitched image.

If you directly put images on top of each other, every pixel you write will overwrite the previous image's pixels. For the areas that overlap, use the image pixel that has the maximum intensity. Clever blending implementations will be rewarded.

3 Experiments

Firstly, you will try to stitch three images left-1, middle, right-1 with the configurations below, except last one. In the last one, you will create a panorama with all images.

3.1 Number of Points

- Use only 5 correspondence points
- Use only 12 correspondence points

3.2 Point Selection

- Use All correspondence points with 3 wrong matches without normalization step
- Use All correspondence points with 3 wrong matches
- Use All correspondence points with 5 wrong matches

3.3 Noisy Points

Add Gaussian noise to your correspondence with different variances. Repeat it without normalization, mention which one is more tolerant to noise, and explain why in the report. Please keep variance in reasonable range. It is not meaningful to use 10^2 or 10^{-6} as standard deviation.

3.4 Stitching All

Use all of the given five images and create a panorama. Please write your code in a modular way. It will be easier to stitch all the images.

4 Report

Report is the most important part of assignment. Please add output images in the report and explain the results of the experiments. Examine the effects of different conditions above in detail. It is encouraged to write the report in Latex. If you support your results with derivations written in Latex, it will be rewarded. Note that if you cannot implement normalized points sections, please do your experiments without it. You will not lose points dramatically. Furthermore, clever ideas beyond the given tasks or good blending methods will be rewarded generously.

5 Deliverables

- Source Code: If you use PyCharm or any other IDE, please be sure that the code can be run from terminal. A main file is required. When the main is run, it will do all the experiments and show the resulting images with appropriate titles. You do not have to use comments in your code, but you must use the signatures of the functions as given to you above. Code must be easily understandable.
- Report: It must be a pdf file and contain all the images you produce. You do not have to write every detail, even if you can explain with equations, it is sufficient. Note that any other format will be ignored, only pdf.
- Correspondence Points: You can save your data as a numpy array or txt.
- Submit all files as one zip file. Please send files in correct format. Use zip for packaging, do not use rar, 7z etc. If you are not being able to upload your assignment files to Moodle, send compressed package via email with the title; ”[cmpe537-2][Your Student Id][Your Name]”.