CV 573 – Computer Vision

-Vaibhav Rao

Project 2

For task 1:

I have used SIFT. I have taken 500 descriptors. sift = cv2.xfeatures2d.SIFT_create(500) After that I take each descriptors and match each descriptor of 1st and match them with all the descriptors of the 2nd image. Here I calculate SSD to find the best matches and if the best two matches i.e. SSD[0] and SSD[1] are within a ratio of 0.7, we can take the indexes and save them as a GOOD Match, otherwise ignore this Descriptor.

SSD has actual values of SSD distance as the first parameter [0] and the indexes as the second parameter[1] (J)

In my good_matches list I save the Index(i) and index(j) at the 0 and 1 location respectively.

We can get these indexes as source and destination arrays from good_matches() and take the respective keypoints of these indexes and pass to our findHomography()

After this I used perspectivetransform() so that I can take the x,y min and max coordinates so that I ensure that my image does not get cropped and these values are taken into consideration in our matrix M we are using for warpPerspective().

FOREGROUND REMOVAL:

Once I have the result image warped and transformed, I just compare Pixel-by-pixel values of the other image(image2) and then apply the pixel with maximum Sum as the foreground and replace the pixels on the warped image1 with the image2 pixel(having a higher sum, I.e. lighter colour)

This removes the foreground and since the image is not a perfect match, there is slight visibility of the background person.

For task 2:

First I calculate the “One-Hot array”

My logic for calculation was simple, I just take each image and compare with the other by matching descriptors. For quicker calculation I just take 100 descriptors and if the images match more the ~20%, I say they have overlapping regions.

I do this for image (I,j) and also for image(j.i) to verify the matches are accurate since there were some reflections that were causing images to be marked as overlapping region.

However, I don’t calculate matches of images with itself (Since they would match and just apply 1 in the one-hot matrix)

After quick one-hot calculations, We reuse most of the code from Task1 to get a perspective transform and the stitch the images just ignoring the part for pixel-by-pixel comparison for identifying foreground/Backgrounds.

BONUS

For bonus, I took 5 images of the campus, However they do not blend well because they were taken at an angle where I was stationary and I rotated my camera in a circular angle. (As I did not have proper equipments) but we can see it stitches the images appropriately that were at a relative good angle.
I have hence submitted 3 images for this case.