

CmpE537 Assignment2

Image Stitching

Elif Çalışkan
201640013
Boğaziçi University

CMPE537 ASSIGNMENT2

Description:

We were asked to implement a program which generates a panorama from given images and corresponding points. For creating a panorama, the homography matrix should be calculated. But before calculating the homography matrix, the points should be normalized.

def normalize(points):

This function normalizes given points. It finds the distance of each point to the average pixel of these points. Then using this information, it finds the normalization matrix. For finding each of the points, the original points are multiplied with this normalization matrix.

def computeH(im1Points, im2Points):

This function computes the value of homography matrix. The im1Points are x and y values but im2Points are the prime values. I created a helper matrix for finding homography. With the use of single value decomposition, I found singular vector of helper matrix and reshaped it to create the homography matrix.

def warp(image, H):

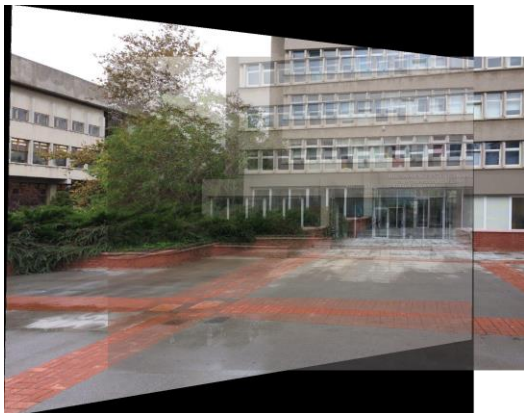
This function gets one image and one matrix for homography and returns the image np array. First, it finds the corners of the warped image by multiplying the corners of original image with homography matrix. Then, every pixel in the original image is transformed by multiplying with homography matrix. At this point, the image has many empty pixels because of integer casting. For the interpolation part, while iterating through the image, if one pixel is black, the rgb value of the corresponding point in original image is used. For this, inverse matrix is used.

def overlap(im1, im2):

This function blends the images by 50 percent. Im1 is the image that was multiplied with homography, im2 is the original image that will not move. First, shifting amount is calculated. For this, I used one selected point from im2 and got the corresponding point by multiplying with the inverse of homography matrix. Then, I subtracted the original from that point, by adding the size difference, shifting amount is calculated. I used paste function to paste the first image to the given place. By adding shift, I pasted the second one also. By using blend function I blended them with the alpha value 0.5.

Experiments: 5 points

Left2- left1



warped left2



Points: Left2 = [(526.6473354231974, 383.2527989252128), (773.8484102104791, 615.6755485893418), (902.822884012539, 375.19189431258405), (1092.2541424093147, 208.59986565158988), (1062.6974921630094, 471.922749664129)]

Points: Left1 = [(287.50716525, 373.84841021), (548.14308106, 615.67554859), (670.40013435, 371.16144201), (841.02261532, 216.66077026), (812.80944917, 459.83139275)]

Left1 – Middle



warped left1



Points: Left1 = [(287.50716524854465, 377.8788625167937), (544.1126287505598, 612.9885803851322), (670.4001343484102, 371.1614420062697), (843.7095835199283, 218.00425436632338), (1070.7583967756382, 363.10053739364093)]

Points: Middle = [(108.82377967, 371.16144201), (386.9249888, 615.67554859), (519.92991491, 367.1309897), (683.83497537, 220.69122257), (900.13591581, 367.1309897)]

Right1- Middle



warped right1



Points: Right1 = [(307.6594267801165, 363.10053739364093), (490.3732646663681, 223.3781907747425), (694.5828481862965, 367.1309896999553), (460.8166144200626, 455.8009404388715), (730.8569189431258, 524.318629646216)]

Points: Middle = [(518.58643081, 369.8179579), (693.23936408, 223.37819077), (900.13591581, 367.1309897), (665.02619794, 463.86184505), (945.81437528, 529.69256605)]

Right2 – right1



warped right2



Points: Right2 = [(48.36699507389176, 352.35266457680257), (256.6070309001344, 209.94334975369463), (475.59493954321545, 367.1309896999553), (228.3938647559338, 459.8313927451859), (515.8994626063592, 525.6621137483207)]

Points: Right1 = [(307.65942678, 361.75705329), (486.34281236, 220.69122257), (691.89587998, 367.1309897), (460.81661442, 457.14442454), (728.16995074, 522.97514554)]

How to run: : python main.py <path of image that moves> <path of image that doesn't move> <point num>

Conclusion: I conclusion, this project was very challenging to me since I didn't have enough time because of other projects. I wrote the normalization function but the homography matrix was calculated wrong with the given normalized points. So I didn't use that part in the solution. Since I didn't know some concepts, I searched some of them online: such as

normalization: <https://stackoverflow.com/questions/52940822/what-is-the-correct-way-to-normalize-corresponding-points-before-estimation-of-f>

warping image: <https://github.com/jmlipman/LAID/blob/master/IP/homography.py>

blending: <https://stackoverflow.com/questions/29106702/blend-overlapping-images-in-python>

Elif Çalışkan 2016400183