COL 780

Computer Vision

Report - Assignment 2

Submitted By :

*Arjun Bhardwaj*

*2016ME10754*

*Tanmay Goyal*

*2016ME20757*

1. **Initialising the code** :-

First of all the OpenCV, Numpy and array modules are imported. In the main function of the code, we call the createPanorama. The arguments of the function are the path to the folder where input images are.

1. **readImages** :-

readIMages function takes input the path to the folder where input images are and stores all the coloured images in color\_images array.



Gray image

1. **resizeImages** :-

It resizes all images to ¼ height and ¼ width and then converts it to gray image for further processing and return the array of images.

1. **getKeyPoints** :-

We have chosen SURF descriptor to find keypoints in the images because SURF features are robust like SIFT but also improve the speed in every step. Analysis shows it is 3 times faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change. In this way we written all the images in the form of key points. We also found out the descriptors of each of these key points.

1. **matchFeatures** :-

We have used flann matcher that takes 2 images and returns the matched keypoints which are present in both the images. It provides approximate nearest neighbours of a key points but it is much faster than the brute-force matching. The knn feature matching is used which provides k best matches for each keypoint. Here we have used k = 2. Next we apply the D’ Lowe ratio test to refine these matches.

1. **getUCHomography** :-

Now we need to find the homography between the images. If the 2 images are different, then we call the function getSingleHomography which creates arrays src\_pts and dst\_pts. These arrays have dtypes float32. Src\_pts is the array of all the keypoint1 and dst\_pts is the array of all the keypoint2 that have a match in the other image. Now we findHomography between the 2 sets of key points and try to fit model using RANSAC which is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. In this way we get the homography between any 2 images and know how to get the 2 images in the same coordinate frame.



1. **getCompleteHomography** :-

Once we have found out the homography between images that have sufficient matches, we can use this information to find out homography between each pair of images. This function returns a homography matrix for each pair of image i, j.

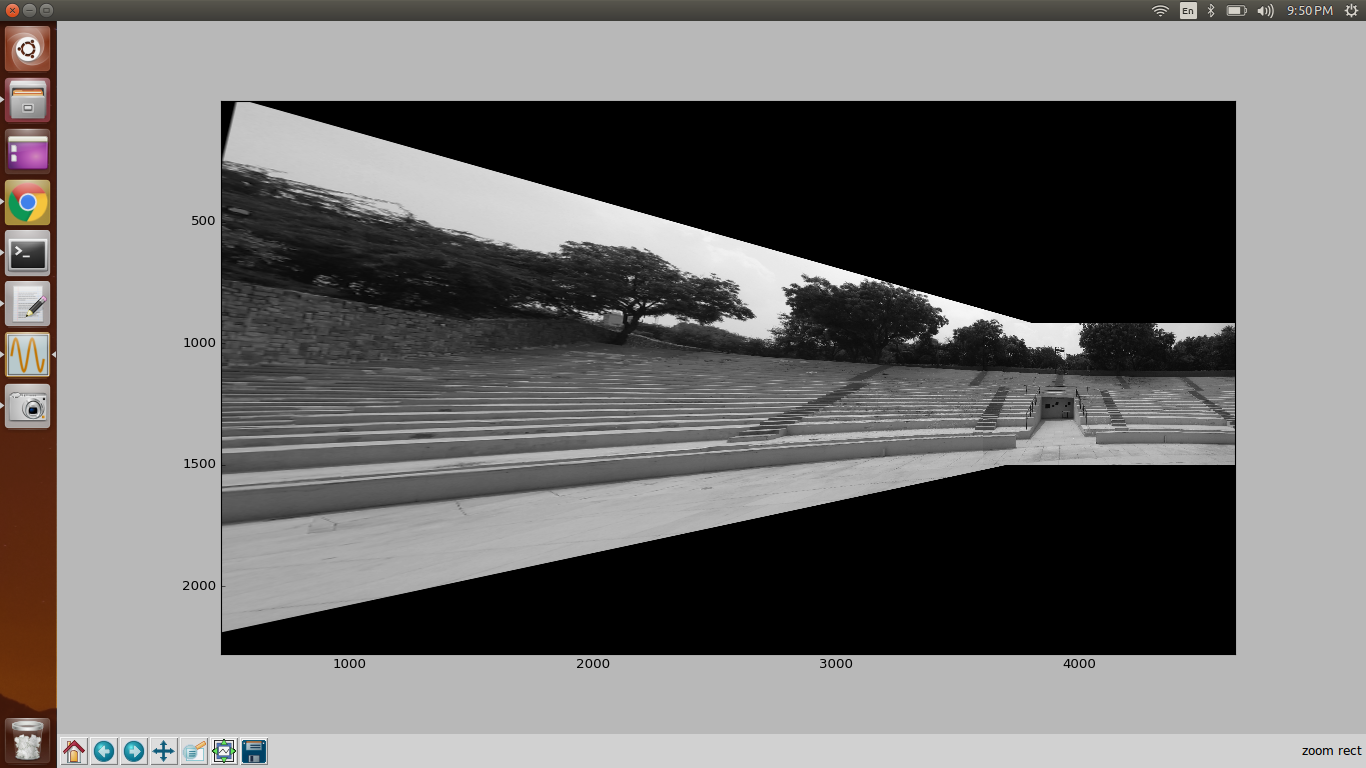
1. **findBestMatch** :-

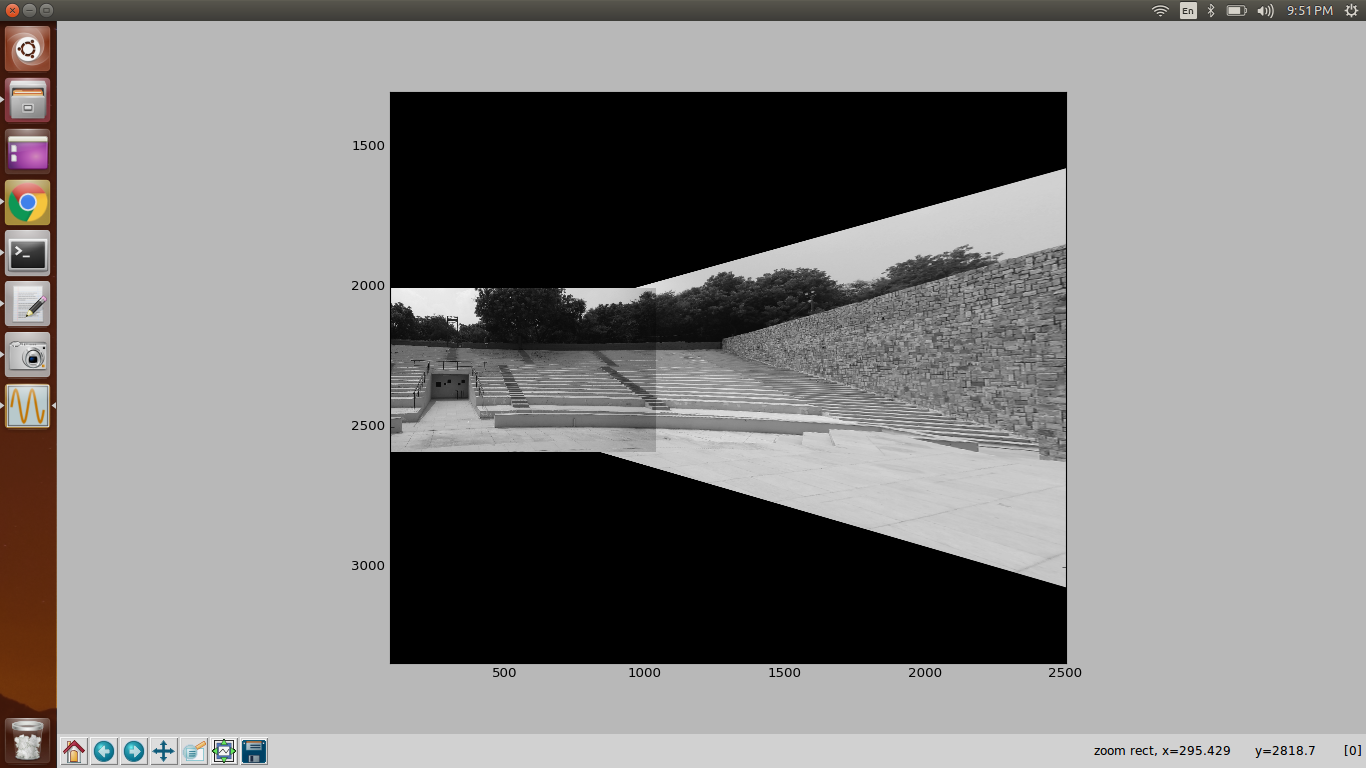
It iterates through all the matches between the 2 images and return the 2 images have maximum number of matches in keypoints. This is done to choose base picture which is one the image with maximum matches. Now as we go on iterating images, we keep on adding it to array **seq** and add their homography of all the images with respect to base in **app\_h** and remove the images from array **rem.**

1. **warpedImages** :-

In this function we calculate the total size of the image. First the calculated homography is applied to each of the image. Then the corner points of each of the image are found. If any image lies in the negative x or y direction, a suitable translation is done to bring all the images in a positive quadrant.







Calculated Homography of all images in a sample set

1. **mergeImages** :-

In this we take all the images having the same height and width as . We now traverse along height and width and for each pixel, we take the average value of pixel of all the images and get the final pixel value of that coordinate. In this way we blend all the images together to get the final joined image with each coordinate having a pixel value average of pixel values of all the images. In this we merge all the image and plot them.