Project 2: Panorama Stitching

Ravali Yerrapothu
CS Graduate Student
University of South Florida
ravaliy@mail.usf.edu

Pooja Vijayan
CS Graduate Student
University of South Florida
poojavijayan@mail.usf.edu

1. Introduction

A panorama is created by stitching a series of images into one single large image, thereby increasing the field of view of a camera. Thus it provides a complete view of an area or location that cannot fit in one single shot.

In this project, we are required to create a panorama of the scene based on the input images which are taken from multiple viewpoints. We have made use of the in-built functions in OpenCV which are needed to produce the single panoramic view. While considering the input images, we have assumed that the geometric variation between the images only involves translation, scaling and rotation.

2. Structure and Implementation

Firstly, we have read the input from user interface following the syntax, "panorama <number of input images (n)> <image1> <image2> <image3>……upto n". By checking the input, we make a call to the respective functions to implement the different operations to obtain the panorama output. The following are the major steps (functions) we implemented to perform panoramic stitching:

Feature Detection and Extraction: Feature points are the points of interest in an image and these interest points could be used to find matching regions in different (input) images. In our implementation, we used the SURF Feature Detector for detecting the feature points in the input images and then use the SURF Descriptor Extractor in order to compute descriptors or feature vectors. This is achieved by calling the "detectFeatures()" function.

SURF (Speeded Up Robust Features) is one of the most popular feature-detection algorithms. It is the speeded-up version of the SIFT (Scale Invariant Feature Transform) algorithm for feature detection. SURF rely on determinant of Hessian matrix for both scale and location. We have assigned the threshold value to 400.

Select Seed Image and Find Matching Pairs: These are performed by calling the functions "findSeedImage()" and "match()" respectively. We are randomly selecting an input image as the seed image and then trying to do feature matching by comparing it with each of the remaining input images (image pairs). We are making use of the FLANN (Fast Approximate Nearest Neighbor) based matcher in order to find the matching descriptor vectors of two images by checking the neighbors.

Of all the matching feature pairs formed, we introduce the concept of finding good matches by defining them as those matches who have their distance to be less than 2.5 times the minimum distance possible in the found matches. We can then proceed to decide the best matched pairs by considering good points and thus discard the mismatched couples, which are the irrelevant pairs to the formation of the final panoramic output and can thus be excluded from the stitching process.

Image Alignment and Stitching: These are performed by calling the function "computeAlignment()".For the best matched pairs , we are estimating the accurate Homography Matrix – the camera rotation and translation are extracted from an estimated homography matrix– by using RANSAC(Random Sample Consensus), and then we will proceed to warp the images. This will provide us a single large stitched output image which is the panorama which is to be obtained for the given input images. We can do image blending as the final step to enhance the stitched image.

3. Input images:

<u>Panorama stitching output:</u>

4. Interpretations of Results

We have performed panoramic stitching as a multi-image matching problem, and have used the best matched local features using SURF and FLANN matcher to find matches between all of the images. And hence our method is insensitive to the ordering, orientation and scale of the input images. Also it can recognize multiple panoramas in an unordered image dataset.

5. REFERENCES

[1] Richard Szeliski, Computer Vision : Algorithms and Applications. springer 2011.

[2] Coding and general queries: http://stackoverflow.com/