HACETTEPE UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

Image Processing

Assignment 3

Tunahan Pınar - 21727652

- **Explanation of Implementation**

In this assignment, our goal is to generate panoramic images from the given sequence of images. For this purpose, we need to implement stitching, blending, generating pyramids of images, and merging.

In my solution, I started with reading images and put the border to the middle image. Then, I extracted SIFT features of each image. These features are used for matching the same features and common parts on the image. For the matching stage, I used a Flann-based matcher which finds the best matches with the nearest search methods. In this step, I matched the features for the left-middle and right-middle pairs. Then I extracted the RANSAC features and find the homography on these pairs. After matching the extracted features, we need to warp the images and blend them. In this step, we need to use the warpPerspective method from open-cv. For this method, we have to pass the right parameters like RANSAC features, images, and shapes. After that, we can use warpPerspective outputs in the blending stage. In blending, we need to blend each image with the middle image. For that purpose, I firstly blend left and middle then blend its output with the right image. The blending part is a very important part of this assignment. In this part, I extracted image pyramids with the given level. After creating pyramids, I blend them. But after blending, we need the reconstruct the blended outputs. After that, the images are now in the panoramic type.

- **Explain the each step of the algorithm. Why can SIFT features be used for? What is the purpose of the RANSAC algorithm? Why are they used for?**

SIFT is an algorithm that can detects and match the features in images. It can be used in many different areas. When we check the algorithm, we can say there is a some main steps that SIFT performs.

One of them is scale-space peak selection. It perform blurring step, difference of Gaussian kernel step and finding key-point step. In this step, the image is convolved with Gaussian blur filter in different scales. After that, it calculates difference of Gaussian kernels. It applies DoG for different octaves in Gaussian pyramid. After this calculations, it calculates Laplacian of Gaussian approximations.

The next step is key-point localization. In this step, it clean-up some key-points from last step because it produces lots of key-points. Some of them are not useful points like features that lie along on the edge.

The next step is orientation assignment step. In this step, SIFT assign an orientation to each key-points that chosen in the previous step. In this way, each key-points become rotation in-variance.

The next step is key-point descriptor. In this step, each key-points already has a location, scale and orientation from previous steps. So, now, we need to compute a descriptors. In this way, it finds image regions that distinctive and invariant.

As a last step, there is only left the key-point matching. It matches the key-points with identifying their nearest neighbors.

The SIFT can be used for feature extraction for matching the features between images. Because, it is a very robust algorithm for feature extraction. It is a invariant to scaling, orientation, and illumination. Also, it is a very fast algorithm. It provides great accuracy, stability, and it handles very well with scale and rotational invariance.

The RANSAC algorithm basically does parameter estimation. It is a resampling technique. It uses minumum number of observations for estimate underlying model parameters. It uses very small set of data expecially when we compare it with its competitors.

It has very basic steps. It selects minumum number of points randomly for determining the model parameters, and solves them. Then, it finds how many points that fit with a predefined tolerance. At the end, if the number points that found exceeds the threshold, re-estimates the model, and terminates. If threshold not be exceeded, it will repeats the steps.

• Can you use different methods to merge images? What are the advantages of the blending compared to other methods?

For merging images, there are different algorithms. For example we can merge images via spatial domain of the images. It is very simple to use algorithm, I think. Firstly, it set the values for images, and aligns the images. After that, it normalizes the image values and it calculates the corresponding intensities with adding them that pixels in the image are not seen.

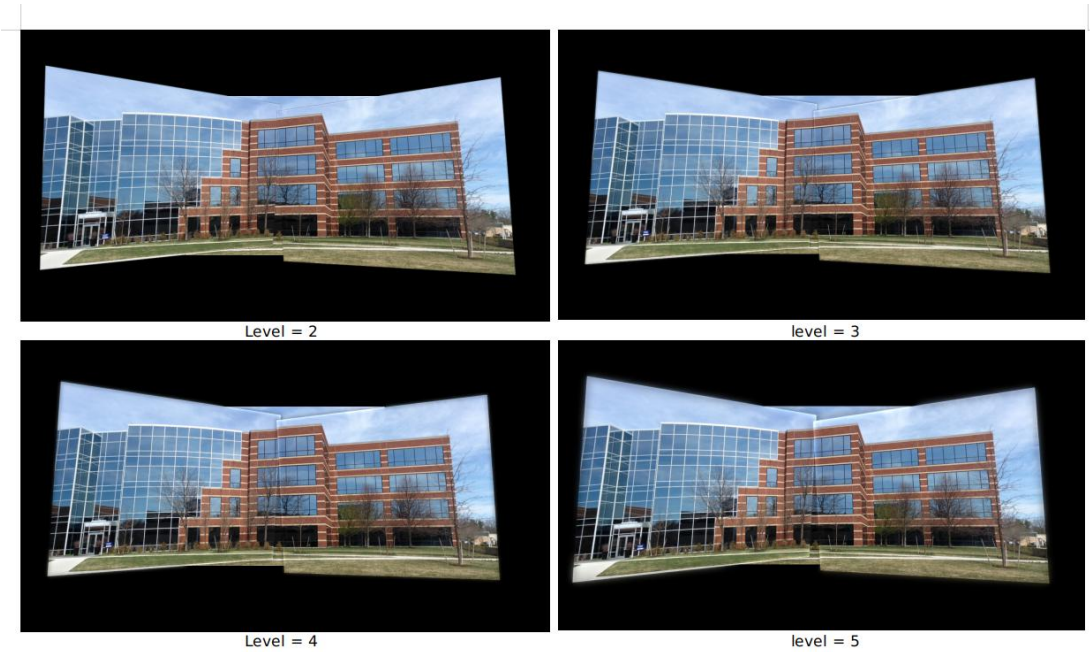
The other method to merging is doing it in frequency domain. This algorithm uses FTT for merging. So, it promises better accuracy. Its steps similar with previous example to the normalizing step. After normalizing, it identify highest frequencies from frequency spectrum. It determines the threshold value and then it adding the pixels that not seen in the image like in previous example. After that, it removes the all frequencies below threshold value. At the end, it takes the inverse of FFT for showing the merged image.

Blending has some advantages when it comes to merging images. First of all, it has great advantage in both computational and complexity costs. It has fewer steps from merging with standard FFT. It requires lower kernels for the same accuracy. It is faster than the

FFT. Also, the blending gives us the possibility the bring together different images into one. It can also provide smoothing or depth and insights into a picture.

- **How can you improve the results?**

In my implementation, we can only change the pyramid level parameter. For this improvement, I tried different levels as shown below, and three level pyramids produced the greatest output, I think. Also, we can try to smoothing edges that exist from side effect of the warping images.



When we look at the my implementation, we can see the there lots of thing that can be effect the outputs. We can use different algorithm for finding features, matching the features, or we can change the RANSAC with another algorithm. We can observe the results that provided from different combinations of the algorithms, and can find the best suitable for us.

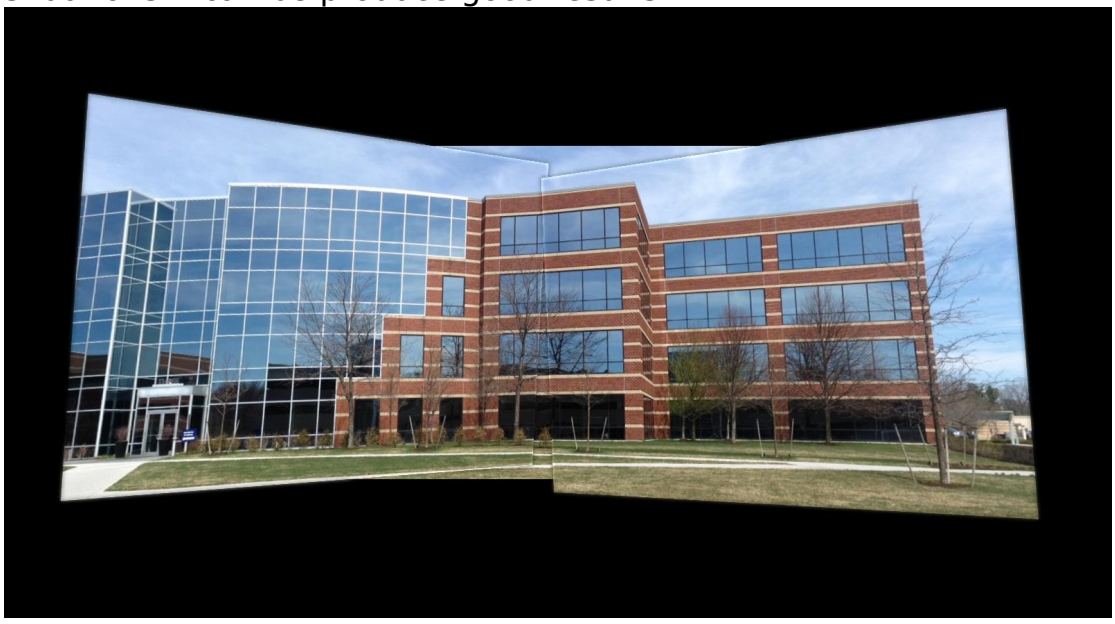
I tried to find some article about this topic, and generally I found that algorithms that we used for this implementation are generally good. But I found an article that they try to develop a method that improve the our algorithm. They changed the how to calculate transformation matrix. For the new input image through all adjacent images, uses the feature of transitivity of matrices acquired from the affine transformation matrix. For that, it does image registration for all adj. images. Then, from these matches, it calculates the affine matrix, and it gets the transformed affine matrix from any image due to transitivity of affine matrices. It uses the statistical registration of all the adj. images for calculating affine matrix. Another difference is

the how contrast the images. New method takes the middle first and than concatenate the other images dynamically. In the below image, I take it from the article. We can see the improved method has better result from the traditional one.

(b) The panorama obtained by the traditional stitching method

(c) The panorama obtained by the improved stitching method

An another thing I tried for improving our result, I changed the how warping the images. For that topic, I found the different method from mine which name is cylindrical warping. For our image it doesn't produce good result. But I write it here because maybe in some situations it can be produce good results.

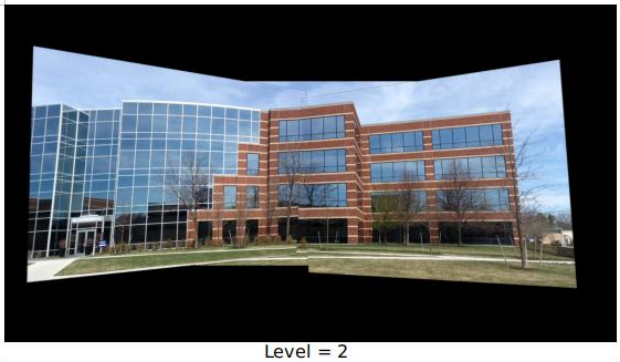


Perspective Warping (my implementation)

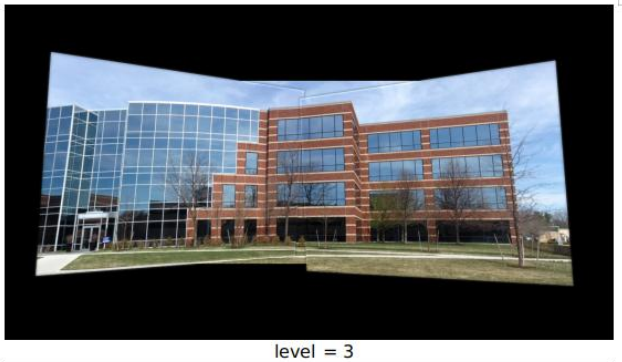
Cylindrical Warping

• Change the parameter for the image pyramid level and comment about the effects of it.

While creating image pyramids, it apply smoothing filter and then subsampling the smoothed image with factor of two along each coordinate. Then, the resulting image repeat these steps again for LEVEL times. At the end, we can see the pyramid that, at the bottom our original image, and from bottom to top, we can see the original image is getting smaller and smoother. In this steps, I can change the level of pyramid. I try to getting results with different level of image pyramids and I got an result that one of them is better. We can see the third one is better result from the other.



Level = 2



level = 3

Level = 4

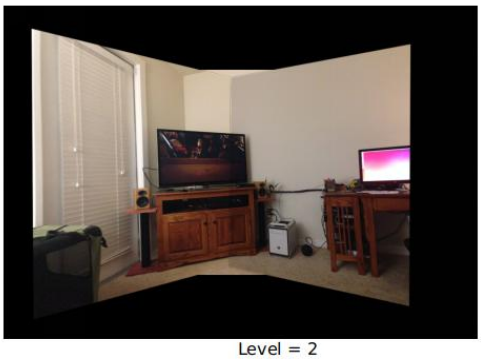
level = 5

While checking the images, I pay attention the how the algorithm matches the extracted features. For that, we can check the lines on the images. For example, when we look at the stone area on the ground at the bottom of the images, we can see the second and fifth images has really bad results. Also, when we check the brick patterns on the images, we can see the which images wrap the images better. We can see, especially in two and five levels, the angle is really wrong while wraping picture, and there is lots of mistakes while blending.

Levels are effects smoothing levels and angle of the images, I think. For example, when we look at the two and five level images, we can see the frames of the wrapped images more smooth in the five level image. Also, as the pyramid level increases, the combined pictures seem to be at an angle that leans more forward.

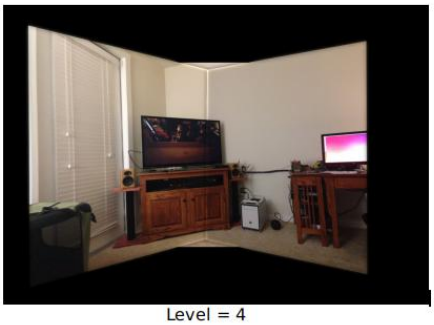
Level = 2

level = 3



Level = 2

level = 3



Level = 4