Final Project - Computer Vision

Name : Wishnuputra Dhanu
Student ID : 2013067

Project 1 - Panoramic Image

Implementation

The project is divided into 2 parts. The first part is using SIFT feature extraction and the second part is using the ORB feature extraction.

The program is structured into 5 steps as below. Each step is processed in a separate function.

Step 1 - Projecting the images into cylinder.

The whole process in this step is handled with the function:

```
projectImageIntoCylinder(cv::String imageFolder);
```

This function will take one argument which is the name of the image folder and return a list of the projected images in a vector. This function will project all the images in the specified folder into cylinder using the `cylindricalProj()` method.

Step 2 - Extracting the features using SIFT or ORB

The process in this step is handled with the function:

```
extractFeatures(std::vector<cv::Mat>& listOfProjectedImages,
                cv::String featuresType);
```

This function will take two arguments that is the list of all the projected images and the type of the feature (SIFT or ORB). Then it will return a list of SIFT or ORB features in a vector.

Step 3 - Find matching features

The process at this step is handled with function:

```
findMatchingFeatures(std::vector<cv::Mat>& listOfProjectedImages,
                     std::vector<Features>& listOfFeatures);
```

This function will take two arguments a list of all the projected images from Step 1 and a list of features extracted in Step 2. In this function we will find all the matching features using `cv::BFMatcher` class. Then we will refine the matches by selecting the matches with distance less than ratio_threshold * min_distance. Where the ratio_threshold = 3 when we are using SIFT and ratio_threshold = 6 when we are using ORB. The refined matches are then pushed into a vector. After that this function will return the vector which contains a list of matching features which has been refined.

Step 4 - Estimate translation between pair of adjacent images

The process in this step is handled with function:

```
estimateTranslation(std::vector<cv::Mat>& listOfProjectedImages,
            std::vector<Features>& listOfFeatures,
            std::vector<std::vector<cv::DMatch>>& listOfMatches);
```

This function will take three arguments a list of projected images from Step 1, a list of features extracted in Step 2, and a list of matches from Step 3. Then we estimate the translation matrix H of each pair of images using the `findHomography()` method with RANSAC algorithm. Next we will save all this matrices in a vector and return it at the end of the function.

Step 5 - Build Panoramic Image

The process in this step is handled with function:

```
buildPanoramicImage(std::vector<cv::Mat>& listOfProjectedImages,
                    std::vector<cv::Mat>& listOfHomography);
```

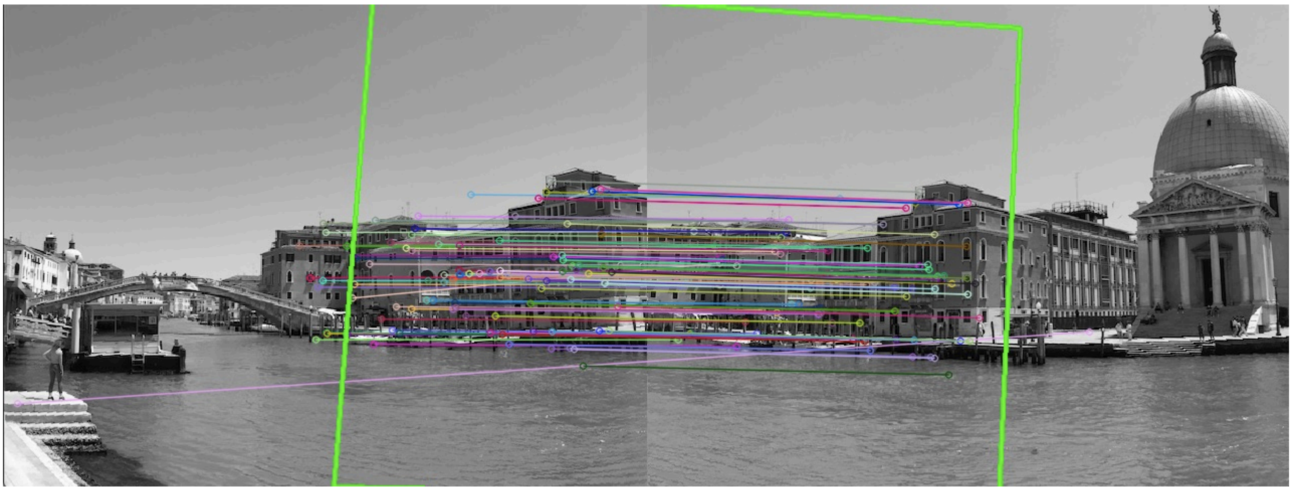
This function will take two arguments a list of projected images from Step 1 and a list of translation matrices from Step 4. Then we will merge the images using the translation matrix H and the `warpPerspective()` method. After we have merge the images, we will equalize the histogram of the merged image.

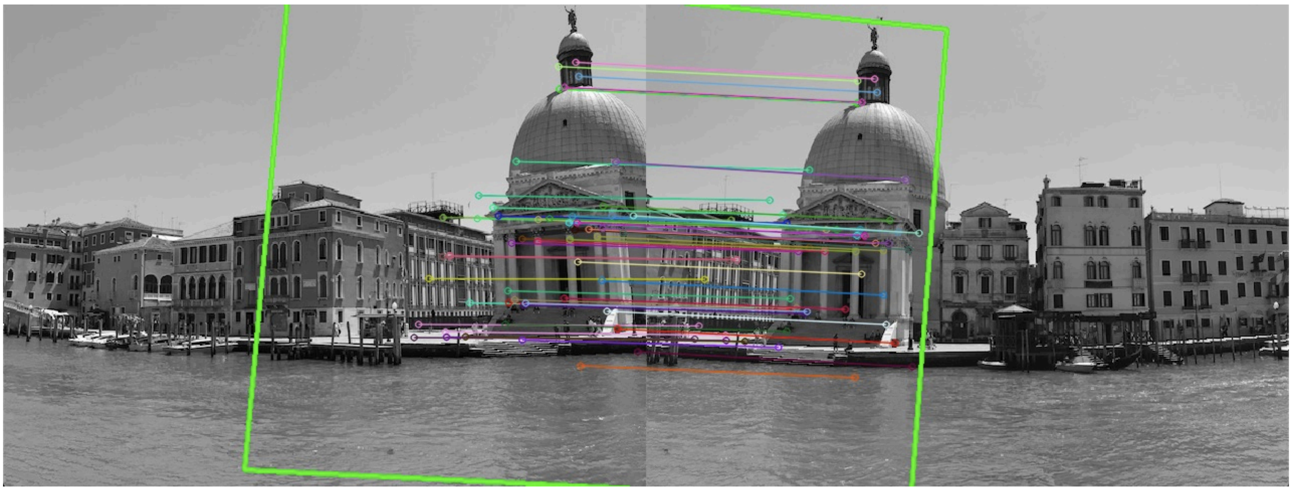
Result

PART 1 - Using SIFT Feature Extraction

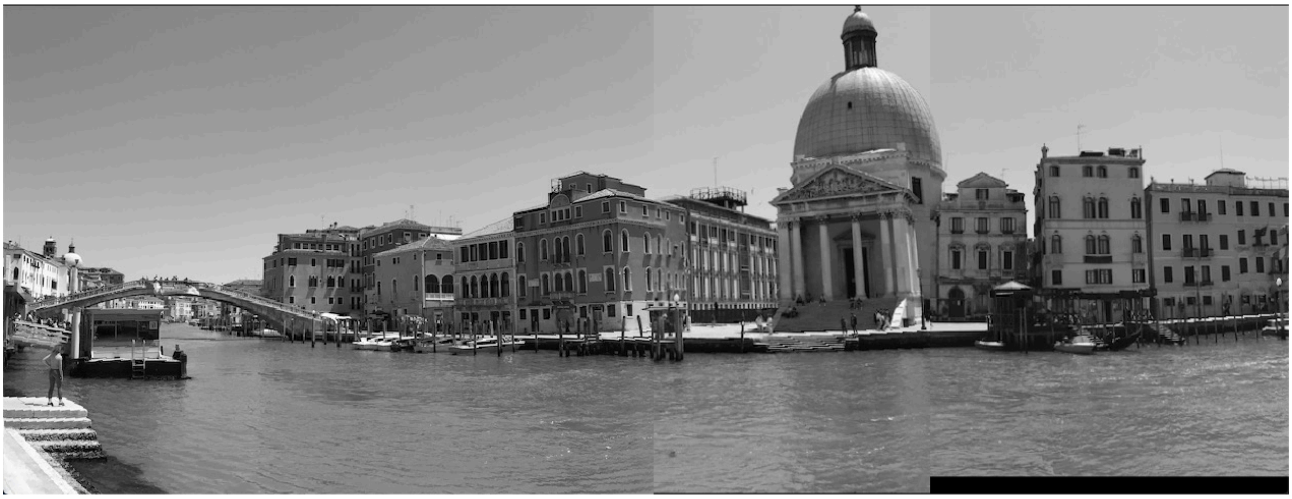
1. Venezia

This is the process of stitching images of Chiesa di San Simeon Piccolo which is taken in Venezia. The images consist of 3 separate images which are taken using a mobile phone. Below is the result after executing Step 4. In this step we have found the SIFT features, matching the features between the two images, and finally finding the translation matrix H.

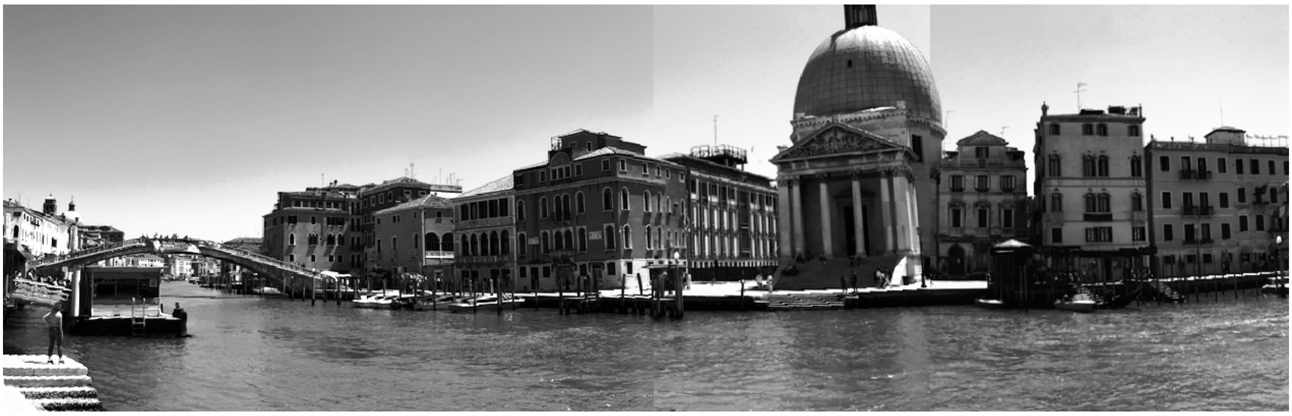




After merging the images using the translation matrix H, we can see that there is black background at the bottom right corner. This is because the image is also translated in y axis.

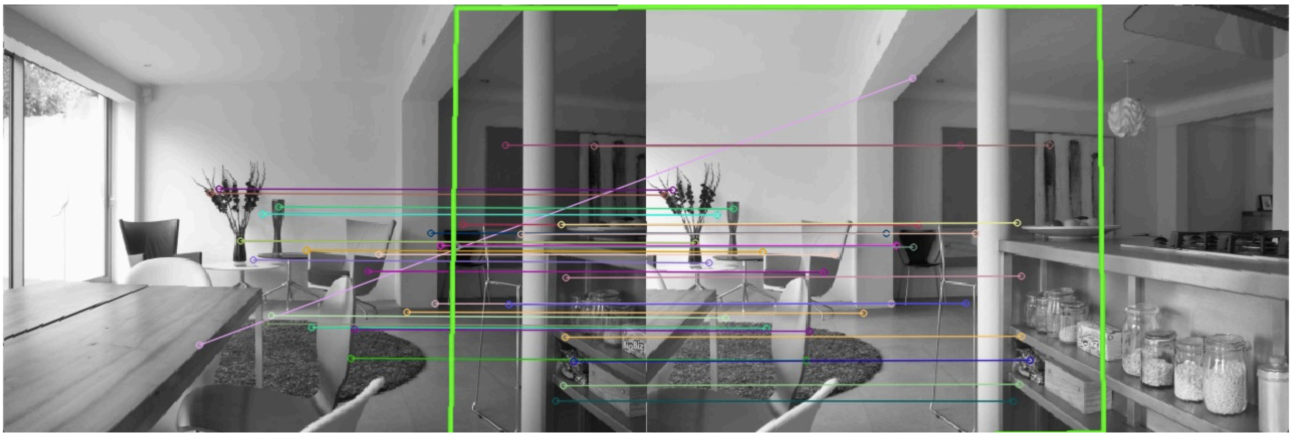


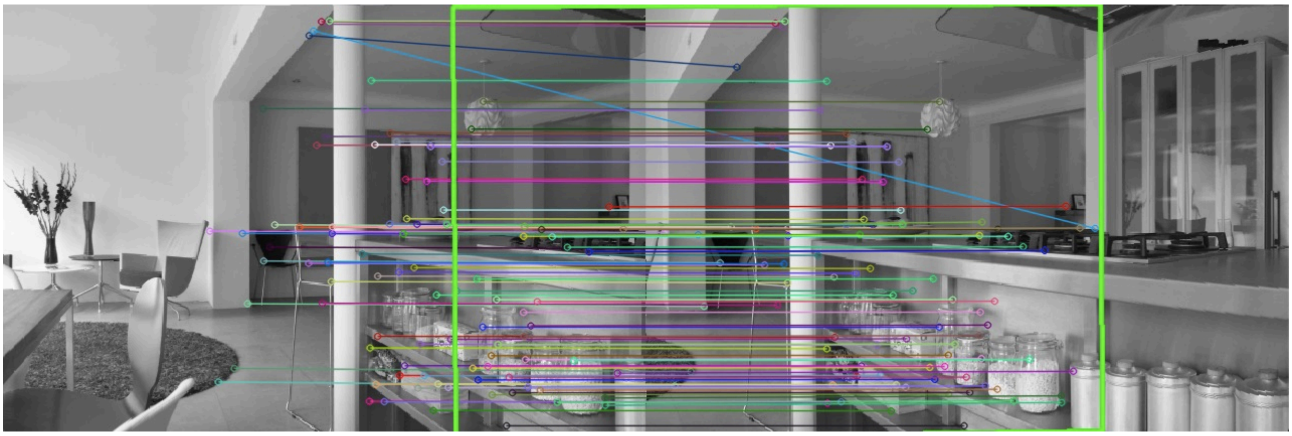
In order to eliminate this black background we will crop the image using the `operator()` function. After cropping the image, then we apply histogram equalization. Below is the final result of the panoramic image.

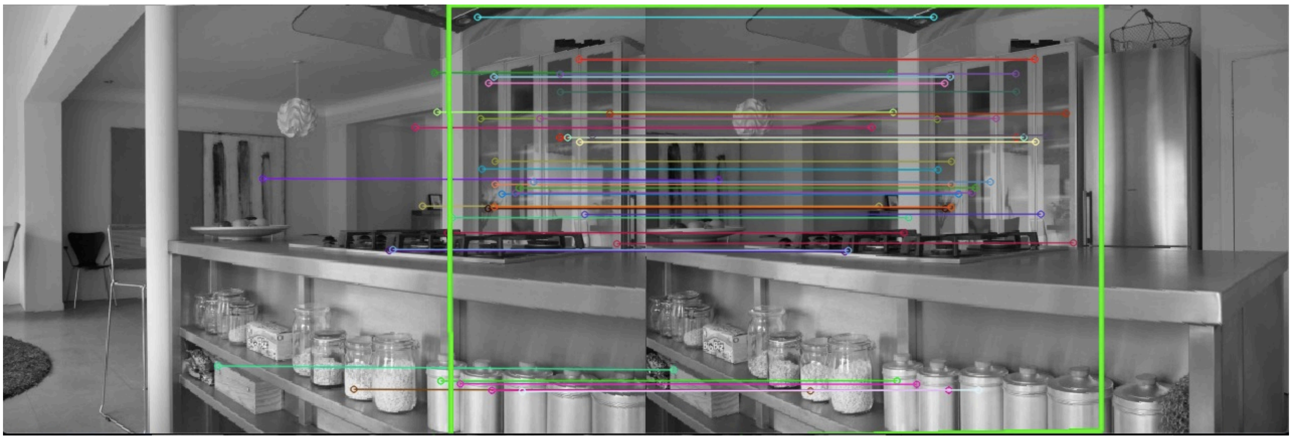


2. Kitchen

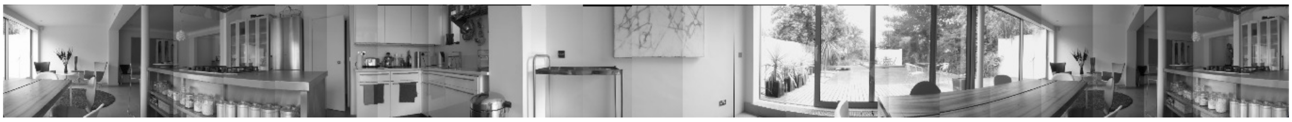
In this process we will use the set of kitchen images. It consists of 23 images. Below are the result of features matching and estimating the translation matrix H of each consecutive pair of images.



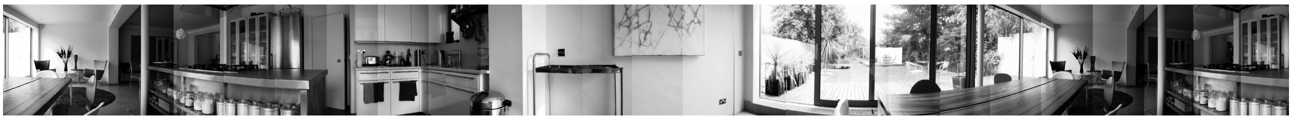




After we have found all the translation matrix H, we will then stitch the images to build the panoramic image. The picture below is the result of panoramic image of the kitchen which consist of 23 images.

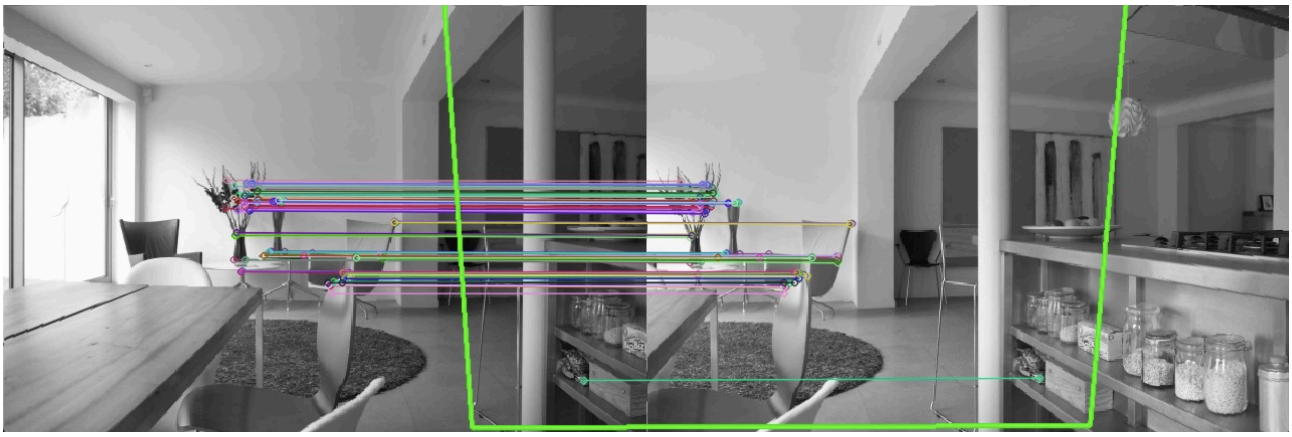


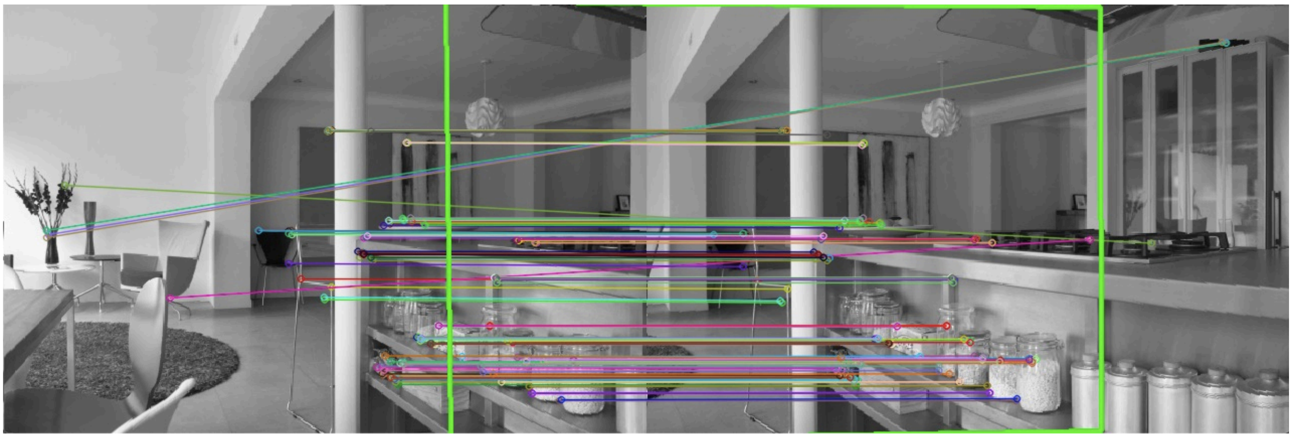
Below is the image result after cropping and applying the histogram equalization.

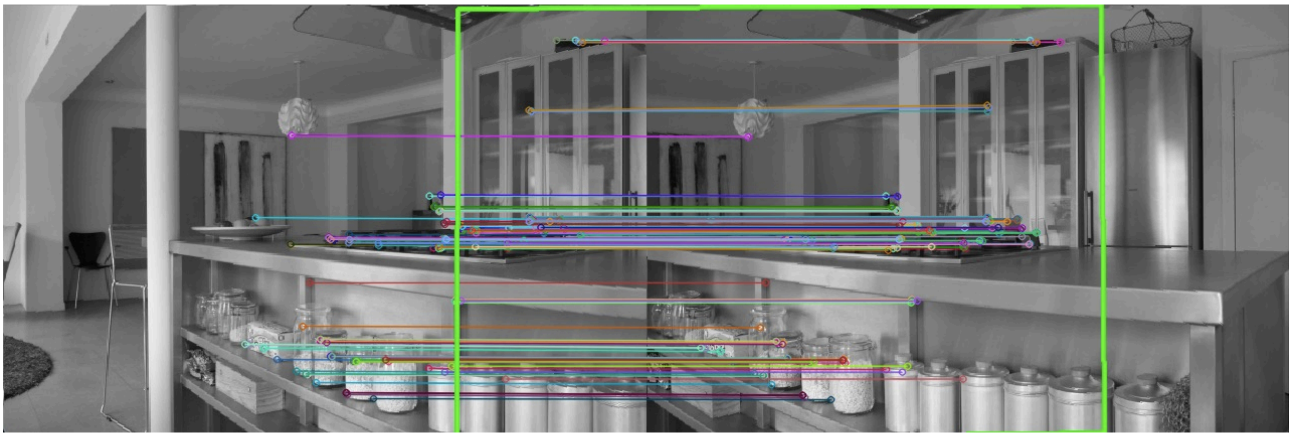


PART 2 - Using ORB Feature Extraction

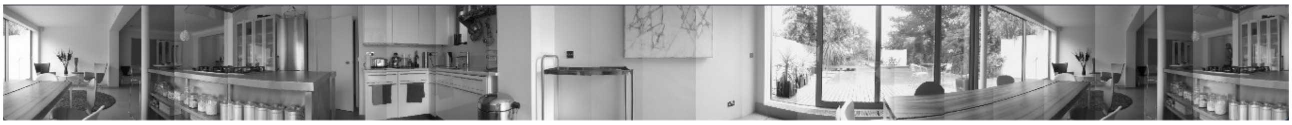
In this part we will use ORB Feature extraction with Hamming norm distance. Below are the result of matching the features and finding the translation matrix H.







After we have found all the translation matrices. Then we will merge the images based on the x translation. Below are the panoramic images after merging.



Next step is we apply histogram equalization onto the image to enhance the contrast of the image.