EEE 508: DIGITAL IMAGE AND VIDEO PROCESSING

PROJECT REPORT

IMAGE STITCHING FOR PANORAMA MODE

Kaushik Koneripalli : 1213175624

Sathish Kumar Katukuri : 1213347263

Arizona State University

Dept. of Electrical and Electronics Engineering

October 18, 2017

Contents

1 Introduction **3**

2 Feature Detectors and Descriptors **4**

2.1 Scale-Invariant Feature Transform (SIFT) 4

2.1.1 Feature Detection 4

2.1.2 Feature Description 6

2.2 Speeded-Up Robust Features (SURF) 7

2.2.1 Integral images 7

2.2.2 Box Filter 8

2.2.3 SURF Detection and Description 8

2.3 Oriented FAST and Rotated BRIEF (ORB) 10

3 Methodology **11**

4 Results and Analysis **12**

5 Challenges **16**

1 Introduction

Image stitching is a problem that has been around for a long time. The fundamental goal of image stitching is to make it as seamless as possible. The obvious method of concatenation of two images doesn't work well because of the overlap that might be there between the two images or because of the visibility of the seam of stitching. The very first approach that was aimed at seamless image stitching was proposed in the year 1983 through the paper "A Multi-resolution Spline With Application to Image Mosaics" [3]. This paper applied the method of Gaussian Pyramids and Laplacian pyramids to perform image stitching. However, this proposed method of seamless stitching works well provided there is no overlap between the two images. Naturally there arose a need to perform image stitching based on the percentage of overlap so that the final image is not just seamless but also doesn't contain any overlaps.

In 2004, a paper titled "Distinctive Image Features from Scale-Invariant Keypoints" [1] was introduced which proposed an algorithm for feature detection and description. It was later discovered that the method of feature detection can be extended to the image stitching application. Specifically, feature detection can be used to compare similarities between the images. Using these similarities, images can then be transformed so as to achieve non-overlapped stitching. The goal of this project is to perform image stitching using feature detectors and descriptors so as to create a panoramic view of a scene.

The organization of the report is as follows: The first section will discuss about the various feature detectors available and describe the ones we adopt, in detail. The second section will explain the methodology in detail. The third section will discuss, analyze and compare the obtained results between various feature descriptors. Finally, the fourth section will draw conclusions and mention a few challenges that are still unsolved.

2 Feature Detectors and Descriptors

Feature Detectors were a major breakthrough in the field of Computer Vision for they have been used to solve a wide range of problems since their inception. Scale Invariant Feature Transform (SIFT) was one such feature detector which was introduced in 2006. Later on, many such detectors like SURF, BRISK, ORB, FAST etc. were introduced which were computationally better than their previous versions. In this project we explore the SURF and ORB feature detectors for the purpose of image stitching.

The most valuable property of a feature detector is its repeatability. The repeatability expresses the reliability of a detector for finding the same physical interest points under different viewing conditions(e.g. different scale and rotation). The neighborhood of every interest point is represented by a feature vector called descriptor. This descriptor has to be distinctive and at the same time robust to noise, detection displacements and geometric and photometric deformations. Finally, the descriptor vectors are matched between different images. The matching is based on the distance (e.g. the Mahalanobis or Euclidean) between vectors. The dimension of the descriptor has a direct impact on the time this takes, and less dimensions are desirable for fast interest point matching. However, lower dimensional feature vectors are in general less distinctive than their high-dimensional counterparts. To understand the advantages of SURF and ORB over SIFT in terms of computations while not sacrificing performance, we have briefly described SIFT, then followed by SURF and ORB thus highlighting their need and advantage.

2.1 Scale-Invariant Feature Transform (SIFT)

2.1.1 Feature Detection

The corner detectors like Harris are rotation-invariant, i.e. the same set of corners can be found even if the images are rotated. This is because corners are retained in the rotated image. However, this is different in case of scaling wherein a corner may not be retained in the scaled image. This is illustrated in the image below. What appears as a corner in one scale may seem flat in another scale. This is where the Harris detector fails.

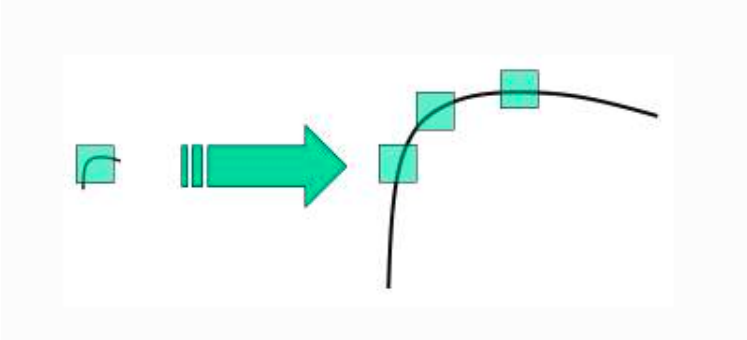


Figure 1: Problem with Harris detector

In 2004, D.Lowe, University of British Columbia, proposed a new algorithm called Scale Invariant Feature Transform (SIFT) through the paper [], which extract keypoints and compute the corresponding descriptors. The first part of SIFT is feature detection which is then followed by feature description. In order to perform detection, a scale space has to be generated in multiple octaves. Scale space is basically the image filtered by a Gaussian filter of varying standard deviations (sigma). After generating the scale space, the DoG(Difference of Gaussian) is computed between successive elements of the scale space to approximate the LoG(Laplacian of Gaussian) as illustrated below. This in-turn reduces computation complexity.

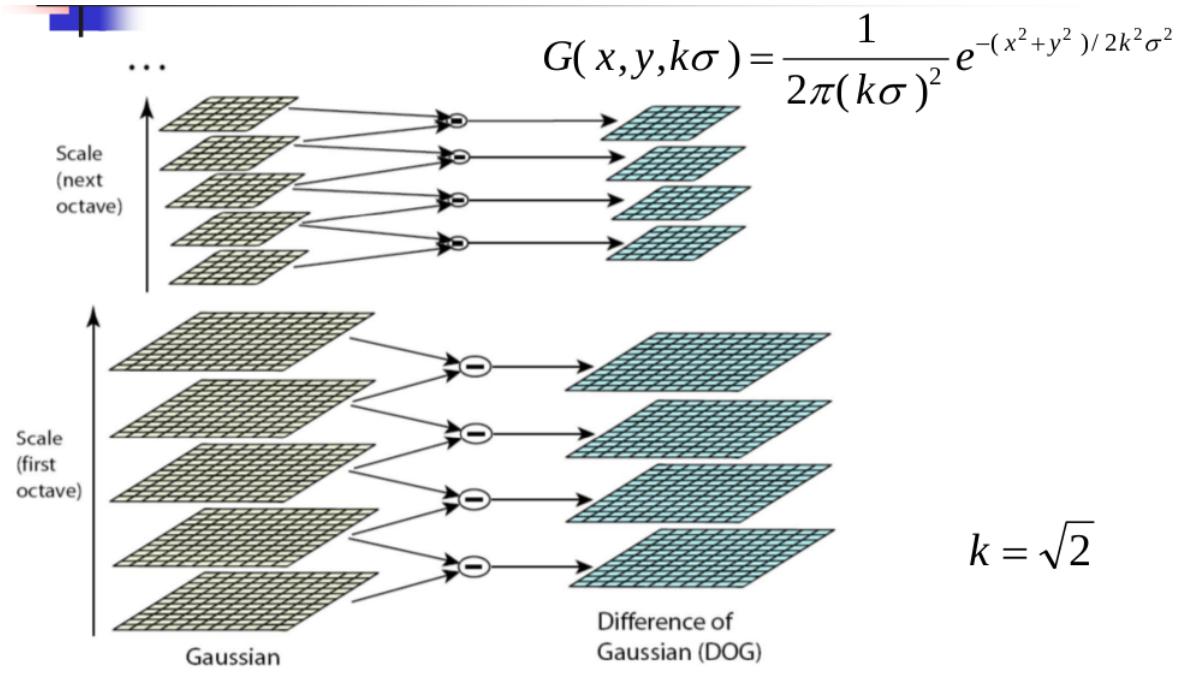


Figure 2: Difference of Gaussian

The next step is to compute the interest points/keypoints. This is performed as follows: Compare a pixel X with the 26 pixels in current and adjacent scales (Green Circles), select the pixel X as keypoint if it is larger/smaller than all 26 pixels. This will give rise to a large number of keypoints. In order to reduce the number of keypoints, the DoG can be approximated using the Taylor Series Expansion. Using this approximation, the extrema(keypoints) of the function is computed. If the function value at the extrema is smaller than some threshold value, then that particular extrema is discarded. In this way the keypoints are filtered out. Furthermore, even the method of taking the ratio of trace and determinant of the Hessian matrix of the keypoint, is used to eliminate those keypoints.

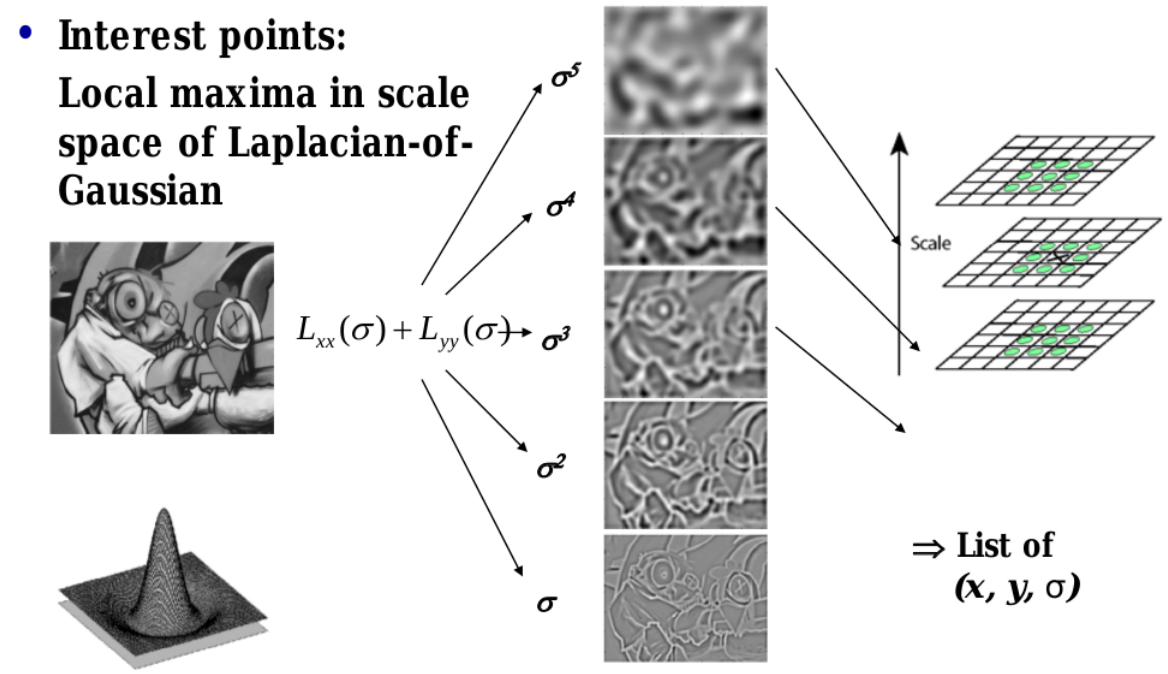


Figure 3: Computing Keypoints

2.1.2 Feature Description

A 16x16 neighborhood around the keypoint is considered. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is computed. So, a total of 128 bin values are now available. It is represented as a vector to form the keypoint descriptor. The figure below shows the weighted histogram(8 bin) for 4x4 regions.

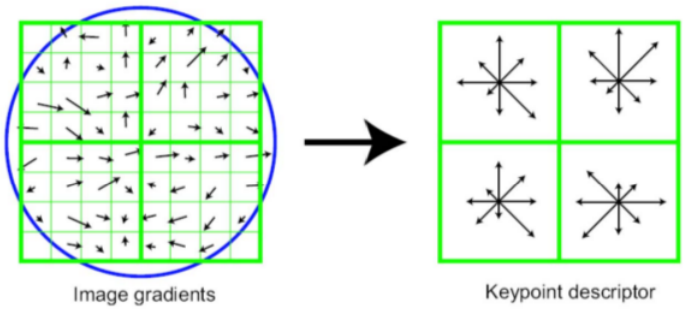


Figure 4: Computing Keypoint descriptors

2.2 Speeded-Up Robust Features (SURF)

As the name suggests, the goal of this algorithm is to make the feature detection and description computationally more efficient compared to SIFT. This algorithm was proposed in the paper [2]. In order to achieve this, it uses idea of integral images and box filters.

2.2.1 Integral images

Integral images allow for fast computation of box type convolution filters. The entry of an integral image $I(X)$ at a location $X = (x, y)$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and X. Once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any upright, rectangular area as shown in below figure. Hence, the calculation time is independent of its size.

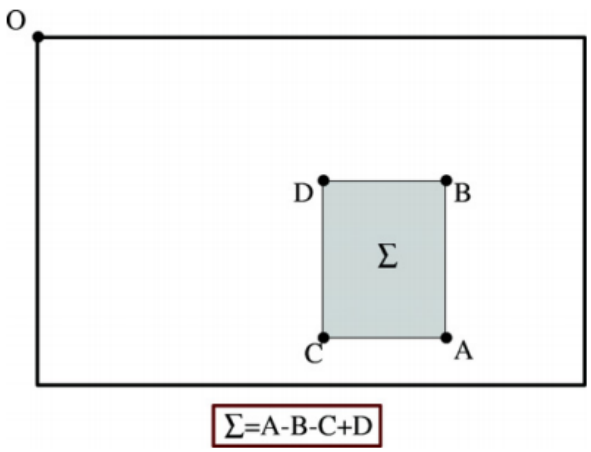


Figure 5: Integral Images

2.2.2 Box Filter

Gaussians are optimal for scale-space analysis, but in practice they have to be discretized and cropped. Discretization provides the fast convolution. Even the slight decrease in the performance does not outweigh the advantage of fast convolutions. As real filters are non-ideal in any case, SURF pushes the approximation for the Hessian matrix even further with box filters.These approximate second order Gaussian derivatives can be evaluated at a very low computational cost using integral images.

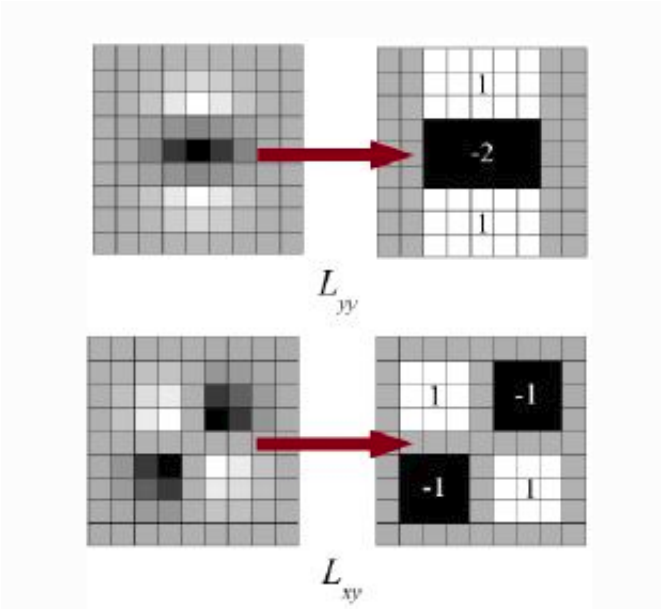


Figure 6: Advantage of Box filter

2.2.3 SURF Detection and Description

For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6. Adequate Gaussian weights are also applied to it. Then they are plotted in a space as given in below image. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. It is interesting to note that, wavelet response can be found out using integral images with ease at any scale. For many applications, rotation invariance is not required, which in-turn speeds up the process. SURF provides such a functionality called Upright-SURF (U-SURF) which boosts the speed of the algorithm.

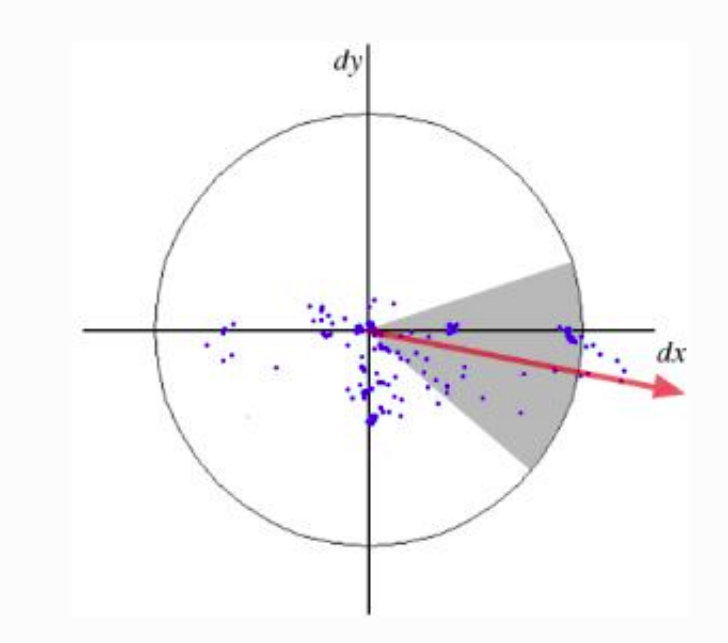


Figure 7: Advantage of Box filter

For feature description, SURF uses Wavelet responses in horizontal and vertical direction (use of integral images is helpful). A neighborhood of size 20s X20s is taken around the keypoint where 's' is the size. It is divided into 4x4 subregions. For each subregion, horizontal and vertical wavelet responses are taken and a vector is formed like:

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum \left| d_y \right| \right)$$

When this is represented as a vector, it gives the SURF feature descriptor with a total of 64 dimensions. Lower the dimension, higher the speed of computation and matching, but provide better distinctiveness of features. For more distinctiveness, SURF feature descriptor has an extended 128 dimension version. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y and $\left| d_y \right|$ are split up according to the sign of d_x , thereby doubling the number of features. It does not add much computation complexity.

Another important improvement is the use of sign of Laplacian (trace of Hessian Matrix) for underlying interest point. It adds no computation cost since it is already computed during detection. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. In the matching stage, we only compare features if they have the same type of contrast (as shown in image above). This minimal information allows for faster matching, without reducing the descriptor's performance.

In short, SURF adds a lot of features to improve the speed in every step. Analysis shows it is 3 times faster than SIFT while the 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.

2.3 Oriented FAST and Rotated BRIEF (ORB)

ORB is an an efficient alternative to SIFT or SURF in computation cost and matching performance proposed in [4]. ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it uses FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also uses the pyramid to produce multiscale-features. But one problem is that, FAST does not compute the orientation. In order to account for the rotation invariance, the following modification was proposed:

It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.

To compute descriptors, ORB uses BRIEF descriptors. But it is known that BRIEF performs poorly with rotation. Thus, ORB steers BRIEF according to the orientation of the keypoints. For any feature set of n binary tests at location (x_i, y_i), define a $2xn$ matrix, S which contains the coordinates of these pixels. Then using the orientation of patch, θ, its rotation matrix is found and rotates the S to get steered(rotated) version S_θ.

ORB discretizes the angle to increments of $\frac{2\pi}{30}$ (12 degrees), and constructs a lookup table of precomputed BRIEF patterns. As long as the keypoint orientation θ is consistent across views, the correct set of points S_θ will be used to compute its descriptor. BRIEF has an important property that each bit feature has a large variance and a mean of approximately 0.5. But once it is oriented along the keypoint direction, it loses this property and become more distributed. High variance makes a feature more discriminative, since it responds differentially to inputs. Another desirable property is to have the tests uncorrelated, since then each test will contribute to the result. To resolve all these, ORB runs a greedy search among all possible binary tests to find the ones that have both high variance and mean close to 0.5, as well as being uncorrelated. The result is called rBRIEF. For descriptor matching, multi-probe LSH which improves on the traditional LSH, is used. It is noted in literature that ORB is faster compared to SURF and SIFT and ORB descriptor works better than SURF. ORB

is a good choice in low-power devices for applications like panorama stitching.

3 Methodology

The process of image stitching happens in the stages described below:

1. Compute features and their descriptors - For this project, we consider the SURF and ORB descriptors available in OpenCV to calculate the features and their descriptors for the two images to be stitched.

2. Perform Matching - Once the features for both images are computed, the operation of matching is performed to match the features of the two images. For this purpose, we have made use of the Flann based matcher in OpenCV using the FlannBasedMatcher class .

3. Compute distance between matches - After matches are computed for the two images, the distance between each match is calculated using the distance method in OpenCV. Typically, the Euclidean/Mahalnobis distance is computed. These distances are then thresholded to discard bad (long distance) matches.

4. Compute Homography - Using the newly computed good matches, the homography matrix is now calculated using the RANSAC algorithm. We make use of the function findHomography in OpenCV.

5. Warp images - Using the homography matrix, a warp transform is computed for the first image using the command warpPerspective in OpenCV. This is done so that the overlap between the images is removed before stitching.

6. Stitching - Finally, the second image along with the warped first image are then stitched using a suitable stitching algorithm. We make use of the stitcher class of OpenCV to perform this. We use OpenCV 3.3 [5] for all our experiments.

A flowchart summarizing the procedure is given below:

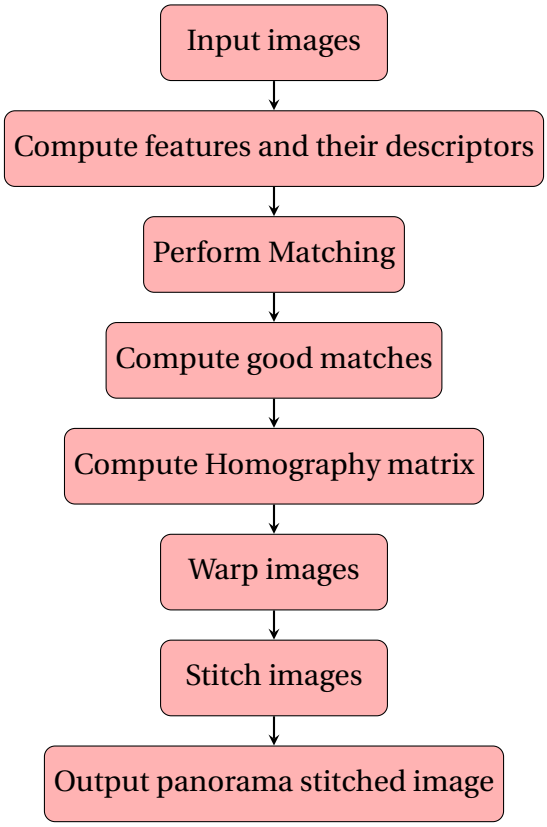


Figure 8: Methodology

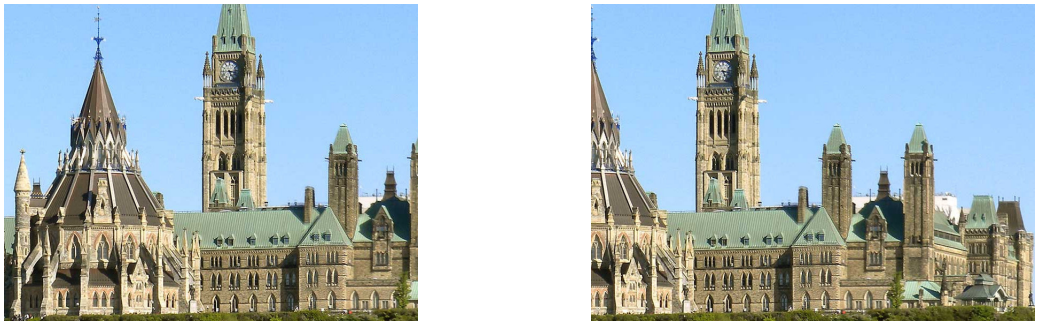
4 Results and Analysis

We have made use of the following image datasets to compute and compare the image stitching algorithm for different feature descriptors.

(a) Image 1

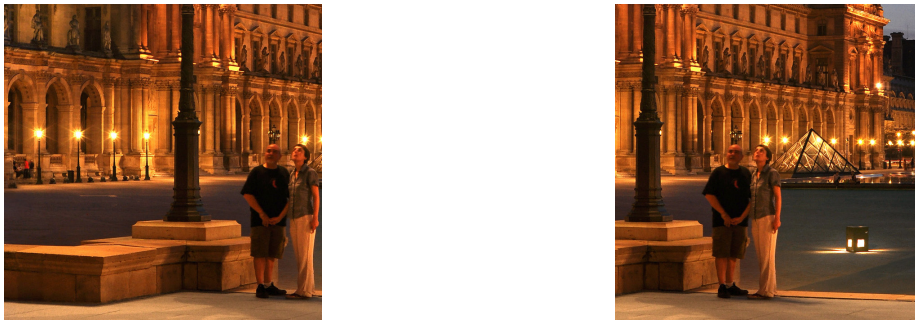
(b) Image 2

Figure 9: Image dataset 1



(a) Image 1 **(b)** Image 2

Figure 10: Image dataset 2



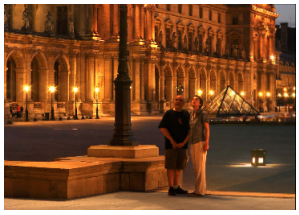
(a) Image 1 **(b)** Image 2

Figure 11: Image dataset 3

Panorama stitched image for the above image datasets using

1. SURF Descriptor:

(a) Image 1 **(b)** Image 2



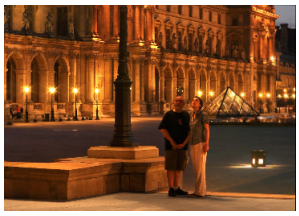
(c) Image 1

Figure 12: Panorama stitched image using SURF descriptor

2. ORB Descriptor:

(a) Image 1

(b) Image 2



(c) Image 1

Figure 13: Panorama stitched image using ORB descriptor

In all of the above cases, we consider two images for stitching. By visual inspection, we can observe that the stitching results obtained by both SURF and ORB are equally good. We can notice that SURF performs better for Image 1 compared to ORB. But, on an average, both descriptors perform equally good. We explore two methods of stitching: using the stitcher class and using normal concatenation. However, the latter doesnt include blending and therefore is not preferred. All our results are based on using the stitcher class except the grayscale image.

Now consider the case of multiple image stitching using the following dataset of six images as depicted in Fig. 14. As it can be seen, there is a significant amount of overlap between the input images. Consequently, the stitching output of the image is of very good quality. The output is shown below in Fig. 15 for both SURF and ORB descriptors.

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

(e) Image 5

(f) Image 6

Figure 14: Image dataset

(a) SURF output

(b) ORB output

Figure 15: Panorama stitched image using SURF and ORB descriptor

5 Challenges

There are several challenges that are still being faced in the area of image stitching. From our observation, the time taken for execution increases as the number of images to be stitched increase. Also, if there isn't significant overlap between multiple images to be stitched, the computed homography will be an empty matrix. Consequently, warping cannot be performed and hence stitching cannot be performed. As an outlier, consider the following case of grayscale images where the overlap is much less compared to the previous multi-image case.

(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Figure 16: Image dataset

We can notice from the stitched image that there are many visible black patches and seams. It can be reasoned as follows: Every time the warp transformation of the image is taken, the angle/perspective of the image changes. Also, in every iteration of loop the input image and the previously stitched image are being resized. Consequently, data is lost and hence there are not sufficient number of keypoints between two images to match. Thus, the warp transform is ruined and consequently the the stitched image is also not of good quality. It can observed from Fig. 17.

Figure 17: Stitched output of grayscale dataset of 4 images

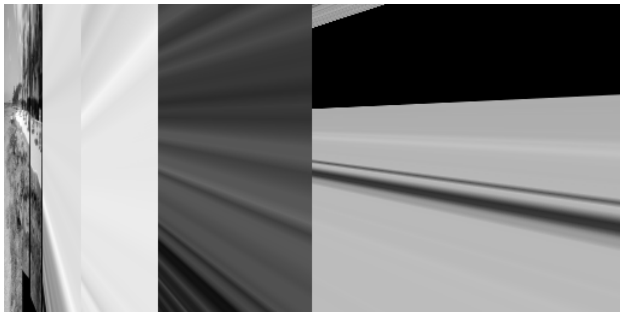


Figure 18: Stitched output of grayscale dataset of 8 images

We conclude by stating that SURF and ORB descriptors are similar in performance with respect to computation time and stitched image quality. There are however a few outlier cases that have been shown where the output of SURF is visually better that of ORB. As a general rule, we observed that as the number of images to be stitched goes on increasing, the quality of the stitched image deteriorates as depicted in Fig 18. This is one of the major challenges that was faced. Also, the images should be fed to the algorithm in sequential order i.e. it is not possible to automatically align the images based on the keypoint matching or any such method. These are a few challenges that were faced during the course of the project.

References

[1] Lowe, D. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision, 60(2), 91-110.

[2] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). "SURF: Speeded up robust features". Lecture Notes in Computer Science, 404-417.

[3] Burt, P., and Adelson, E. (1983). "A multiresolution spline with application to image mosaics." ACM Transactions on Graphics, 2, 217-236.

[4] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "ORB: An efficient alternative to SIFT or SURF". Computer Vision (ICCV), 2011 IEEE International Conference on, 2564-2571.

[5] OpenCV : https://opencv.org/opencv-3-3.html