# COSC 342 Assignment 1 – Image Mosaicing

## Report for Roland Locke, ID 5493858

## Introduction

This paper explores the image mosaic pipeline with a focus on understanding how the overlap between images influences the efficiency of stitching algorithms. Two Experiments will be performed in this paper, both involve the overlap of images on process of the image stitching pipeline. The first experiment tests the effects of different image overlaps on time taken to perform a Flann-based Feature Matcher. The second focuses on the effects of the same set of images average time to complete RANSAC homography estimations.

Experiments in this paper often reference image Overlap. Overlap is defined as follows: The amount of one image contained in another image (measured in pixels). That is, if there are three Images: Image1, Image2, and Image3. Image1 and Image2 both share %50 of the same scene. While Image1 and Image3 share %20 of the same scene. The number of pixels shared by Image2 and Image3 with Image1 differ. Each Image is a right-hand pan across the original larger image. That is, if Image4 contains 50% of image1, then the right-hand half of Image4 is the same as the left half of Image1. This is demonstrated clearly in Figure 2.

The Images selected for testing are 1200x900 pixel in jpeg format. To keep the colour, contrast, and sizing the same across all the images, each is a section cut from a single image. Image2 contains %90 of image1’s original pixels, while image3 contains %70 of image1's original pixels. For a full breakdown of the image overlap percentages see Figure 1.

|  |  |  |
| --- | --- | --- |
| **Percentage Of image1's Pixels Contained in the Image** | | |
| **Image** | **Overlap Percentage** | **# of Pixels** |
| Image1 | 100% | 1080000 |
| Image2 | 90% | 972000 |
| Image3 | 70% | 756000 |
| Image4 | 50% | 540000 |
| Image5 | 30% | 324000 |
| Image6 | 10% | 108000 |
| Image7 | 5% | 54000 |
| Image8 | 1% | 10800 |

Figure 1

Image1 Image4

Side Shared with Image4 Side Shared with Image1



Figure 2

As both experiments in this paper measure time it’s pertinent that the method of measurement is clarified. The timer used to measure the data in this experiment is C++’s std::chrono::high\_resolution\_clock.

## Experiment 1: Feature Matching:

The object of this experiment is to determine whether the overlap of two images being tested influences the time it takes to perform a Flann-based Feature Matcher. Or: Does the amount of overlap of the two images effect the approximate feature matching time?

### Experimental Design:

The feature detection method in this experiment implementation will be explained to ensure the experiment can be repeated. This experiment was performed in Visual Studio with C++ and used OpenCV’s methods to store data and implement many parts of the image stitching process. Firstly, OpenCV’s SIFT feature extractor was used to extract features from images which were stored as OpenCV’s Mat class. The detected features were stored as OpenCV Keypoints in a Standard Vector object. The Flann-based feature matcher is stored in in OpenCV’s Descriptor matchers and uses K-Nearest Neighbour’s searches to locate near features. K=2 in this experiment.

The Flann-based Feature Match implementation is provided by OpenCV and can be found in the Open CV documentation: <https://docs.opencv.org/3.4/dc/de2/classcv_1_1FlannBasedMatcher.html>

The implementation of this experiment loops through feature matching with image1 for each of the eight images in succession, this process is repeated ten times and occurs in the same order each time. The feature map of Image1 remains the same for all the tests, but the feature descriptors for all other images are re-detected every loop.

### Results:

Presented in Figure 3 are the average time in seconds results for each image after 10 feature matches as well as the standard deviation in seconds. Image1 is included in the results as it provides the time of two identical images. The time taken to match features grows with the percentage overlap of the two images. TheStandard Deviation column provides the standard deviation in time taken to match the features. Figure 4 and Figure 5 visually represents the data from Figure 3’s Average Time and Standard Deviation columns respectively.

|  |  |  |  |
| --- | --- | --- | --- |
| **Average Time Taken to Match Features** | | | |
| **Image** | **Average Time (seconds)** | **Overlap With Image1** | **Standard Deviation (seconds)** |
| image1 | 1.128 | 100% | 0.026 |
| image2 | 1.118 | 90% | 0.043 |
| image3 | 1.094 | 70% | 0.042 |
| image4 | 1.034 | 50% | 0.039 |
| image5 | 0.987 | 30% | 0.051 |
| image6 | 0.989 | 10% | 0.046 |
| image7 | 0.989 | 5% | 0.043 |
| image8 | 0.986 | 1% | 0.012 |

Figure 3

Figure 4

Figure 5

### Discussion/Conclusions:

The results of the experiment show that the more overlap of two images contain, the longer it takes a Flann-Based feature matcher to match features. There are however, diminishing returns as the image overlap approaches 0%. Tests with 30% to 1% image overlap hovered just under one second to compute the feature matching process. Feature matching two images that share 90% of the same pixels took 1.118 seconds. Down only 0.01 seconds from the longest result 1.128 seconds with two images that shared 100% of the same pixels (the same image). Images that overlap from 30% to 90% see a sharp uptake in time to feature match, while the 90%-100% range sees only a very small increase in time.

With negligible returns, between 30% and 1% image overlap, the data suggests that images should contain a 30% to 1% overlap for the best timing results.

The standard deviation shows that both identical images and images that share 1% overlap had the smallest standard deviations (0.026 and 0.012 respectively). While Image5 with 50% overlap had a standard deviation of 0.051.

## Experiment 2: RANSAC for Homography Estimation

The object of this experiment is to determine whether the overlap of two images effects the time taken to perform a RANSAC homography estimation. Or: How does the overlap Images effect the time it takes to perform a RANSAC homography estimation?

### Experimental Design:

The homography estimation method performed in this experiment uses OpenCV’s RANSAC implementation with a threshold of 0.3. The implementation of this experiment loops through a homography estimation with image1 for each of the eight images in succession, this process is repeated ten times and occurs in the same order each time. The feature map of Image1 remains the same for all the tests, but the feature descriptors for each image being matched with Image 1 are re-detected and re-matched every loop.

Information about OpenCV’s RANSAC Implementation can be found at the following link: <https://opencv.org/blog/evaluating-opencvs-new-ransacs/>

Bad matches from the feature matching process (whose implementation remains identical to the feature matching experiment above) are filtered out on every loop pass. Only feature matches with a distance less than 0.8 deviations from one another were accepted. This was done to improve an eventual homography warp.

### Results:

Presented in Figure 3 are the average time in seconds to complete a homography estimation for each image after 10 estimations as well as the standard deviation in seconds. Figure 7 visually represents the Average Time data from Figure 6 while Figure 8 visually represents the Standard Deviation column from Figure 6.

|  |  |  |  |
| --- | --- | --- | --- |
| **Average Time Taken to Perform RANSAC Homography Estimation** | | | |
| **Image** | **Average Time (seconds)** | **Overlap With Image1** | **Standard Deviation (seconds)** |
| image1 | 0.016 | 100% | 0.02 |
| image2 | 0.01 | 90% | 0.04 |
| image3 | 0.009 | 70% | 0.02 |
| image4 | 0.006 | 50% | 0.012 |
| image5 | 0.006 | 30% | 0.001 |
| image6 | 0.003 | 10% | 0 |
| image7 | 0.012 | 5% | 0.002 |
| image8 | 0.011 | 1% | 0.005 |

*Figure 6*

Figure 7

Figure 8

### Discussion/Conclusions:

The results of the experiment are not conclusive. It appears that a steady drop in Homography Estimation time occurs as the overlap of the two images decreases, but that assertion doesn’t hold true for image7 and image8. Instead, these two images see an increase in estimation time. This result isn’t skewed by outliers either as the standard deviation in estimation times for image7 and image8 are relatively low compared to other images. Due to this inconsistency, the data is inconclusive.

The data suggests there is a “sweet spot” for image overlap in RANSAC homography estimation at around 10% image overlap. However, the time improvement that this “sweet spot” provides is negligible in the image pipeline as a whole.

## Final Remarks

While the RANSAC Homography Estimation experiment provided no clear results on the overlap of images, the Feature Matching experiment provided clear results on improving time by overlapping the feature matched images. Each of the experiments suffered from some shortcomings.

Only a single group of images were used for the testing process in this experiment, further attempts should expand to using more images across a diverse range of conditions.

Note that quality of features matches was not measured in this experiment. For the feature matching experiment specifically, as the time results between 30% and 1% image overlap are almost identical, further research should explore the differences in quality from 30% to 1% image overlap. Quality of the homography estimation was not explored either but may explain the strange variations in estimation times.

Furthermore, both experiments standard deviation results should be viewed sceptically. Even small deviations in the computing time are heavily dependent on the background processes running on the machine. Further research on standard deviation should explore isolating the feature matching from background processes. Therefore, it’s hard to pull meaningful information from the standard deviation data.

Although the RANSAC Homography Estimation experiment didn’t provide clarity, the Feature Matching experiment showed advantage in time efficiency while feature matching when decreasing the overlap of the two images.