COSC342 Assignment 1 – Image Mosaicing

Report for Matthew Jennings, ID# 644046

In these experiments I intend to explore how varying the parameters and inputs of the image mosiacing pipeline affects the results. This will be done in two sections, in the first I will explore variances to the feature generation and feature matching between two given images and in the second I will explore variances to the homography estimation using RANSAC given these matches.

To do this I will be using the same set of images throughout the report, five sets of two images taken in a variety of settings. These images will be in jpg format at a resolution of 2048x1536 starting at 100% quality. These 5 sets of images were selected from a pool of 16 sets, all selected images could perform a visually good mosaic and had a variety of characteristics to create as wide a range of conditions as possible. Full images and experiment conducted for selection available on request.

Experiment 1: Feature Matching

In this section I intend to explore how using simplified images for the feature matching process affects the number of matches, the number of good matches and ultimately the time taken to complete the feature matching process.

For baseline I will be using both brute force and FLANN based matching on full quality versions of the images and for the matching on the variable information images I will be using brute force exclusively. The intent here is to see how reduction of the amount of information affects the number of features and the relative number of useful matches and how that affects time taken. What I expect to see is a reduction in features as the amount of information decreases and with a reduced number of features a significant reduction in time taken for feature matching. At some point there should be an overlap between time taken to compute the matches using brute force on simplified images and the time taken for FLANN based matching on the images with full information. At this point the most interesting data will appear; using the same amount of time how does reduced information brute force compare to full information FLANN in terms of number useful matches.

With respect to creating derivative images with reduced information I intend to use jpgs at two different compression levels: 50% and 10%. I will also be using scaled images at sizes of 0.5 and 0.25. In both cases I will be using the resulting homography to create a mosaic of the original images.

The goal of this experiment is to find features and their corresponding matches between the two images at full quality by using versions of these images that have been altered to contain less information to reduce the scope of data the algorithms have to work with which should in theory reduce their run time. The hope is that this can be done while maintaining adequate numbers of good matches to perform the rest of the mosaicing process and ideally yield a meaningful comparison between these matches and a FLANN based matching system.

In order to isolate the effect of the alterations made to the images themselves the internals of the process will remain static. For feature detection I will be using SIFT and for match filtering I will be using k nearest neighbors where k=2 and a good match will be defined by a match that has distance less than 0.8 times the second-best match. RANSAC with an inlier threshold of 3 will be used to generate a homography and this will be used to create a mosaic of the original quality images. The final mosaic will then be visually examined for changes and inaccuracies relative to the best-case mosaic.

Experiment 1 Data:

Below is data aggregated and simplified, full raw data available on request.

Average Data (mean)

Averages	Features1	Features2	F Time	Matches	G Matches	M Time	T Time
BF Full	33891.6	33874.6	0.848	33891.6	4066.6	3.5604	4.4084
FLANN	33891.6	33874.2	0.8532	33891.6	4072.6	0.677	1.5302
Scale 0.5	6220.6	6024.6	0.1964	6220.6	914.2	0.1054	0.3018
Scale 0.25	1710.6	1721.2	0.0506	1710.6	187.4	0.008	0.0586
JPG 50%	36470.8	36843.2	0.8528	36470.8	3490	4.0704	4.9232
JPG 10%	47199.2	47115.8	0.9194	47199.2	1724.6	6.8486	7.768

Percentage relative to brute force

	Matches	G Matches	F Time	M Time	T Time
BF Full	100.00%	100.00%	100.00%	100.00%	100.00%
FLANN	100.00%	100.15%	100.61%	19.01%	34.71%
Scale 0.5	18.35%	22.48%	23.16%	2.96%	6.85%
Scale 0.25	5.05%	4.61%	5.97%	0.22%	1.33%
JPG 50%	107.61%	85.82%	100.57%	114.32%	111.68%
JPG 10%	139.27%	42.41%	108.42%	192.35%	176.21%

G Matches = Good matches (0.8 knn)

F Time = Time to find features

M time = Time to find matches in these features

T time = Sum of F and M; total time.

Using percentages as a change from brute force values there are some very clear and meaningful differences here.

With respect to jpg quality changes there is a result that going into this I did not expect, but in hindsight makes complete sense. My expectation was that reducing the amount of information present in the image would reduce the number of features found and so the rest of the pipeline would have less data to deal with and therefore be faster. This is not what we see here, we see a significant increase in features found as the quality decreases as well as a significant reduction in the number of good matches found relative to both BF and FLANN. Ultimately this results in small but significant increase in time taken to find these features and a very large increase in the amount of time it takes to process them. There is a non-linear relationship between the number of matches found on lower quality images and the amount of time it takes to perform the matching process, an increase of 39% in number of matches found resulted in a 92% increase in time taken to find them. The reality is that jpg compression does reduce the amount of information present in an image, but it does not reduce the number of pixels, so therefore this result was predictable, the algorithms must process the same amount of information, but it is now less distinct. As these images contain a lot of the same “features” (in the more human vision defined sense) as they are taken in the same setting the distinction between them becomes less mathematically clear when compression is applied, that which made them distinct from each other inherently requires more data and removal of this data creates more ambiguity and so more matches that are not matches.

Scaling exceeded my expectations to the point where a performance comparison between them and FLANN is not as meaningful as I had hoped. What we are seeing here is a massive reduction in the amount of data in two stages, as by a scale reduces the number of pixels to process by pixels/((1/scale) ^2) meaning that a 0.5 scaling results in a division by 4 and 0.25 a division by 16. As the matching process is in essence a multiplication between the pixels of the two images in an O(n^2) operation this reduction is again raised to the power of 2. This means that a 0.5 scaling results in theoretical time reduction by a factor of 16 and 0.25 results in a factor of 256. These numbers are not exactly what we see in the data, at 0.5 we would expect 0.275 from the average brute force and we see 0.301 and at 0.25 we would expect 0.0171 and see 0.0586. There is no appreciable difference between expectation and reality at 0.5 but at 0.25 there is, it is taking more than 3 times as long as the math would suggest it should. In order to better analyze this, we need to look at what parts of the process make up this time in the real world; feature finding time of 0.0506 and a matching time of 0.008. Most of the processing time is dominated by feature finding time and not matching time.

Frobenius norm values compared to brute force on full images

Scale	A	B	C	D	E
0.5	18.94	3.07	3.3	3.71	0.67
0.25	17.14	0.45	11.72	8.56	2.62
0.1	15.5	2.96	22.21	15.69	20.28
0.05	25.382	FAIL	127.1	159.43	2368.27
Breakpoint	None	0.05	0.05	0.05	0.05

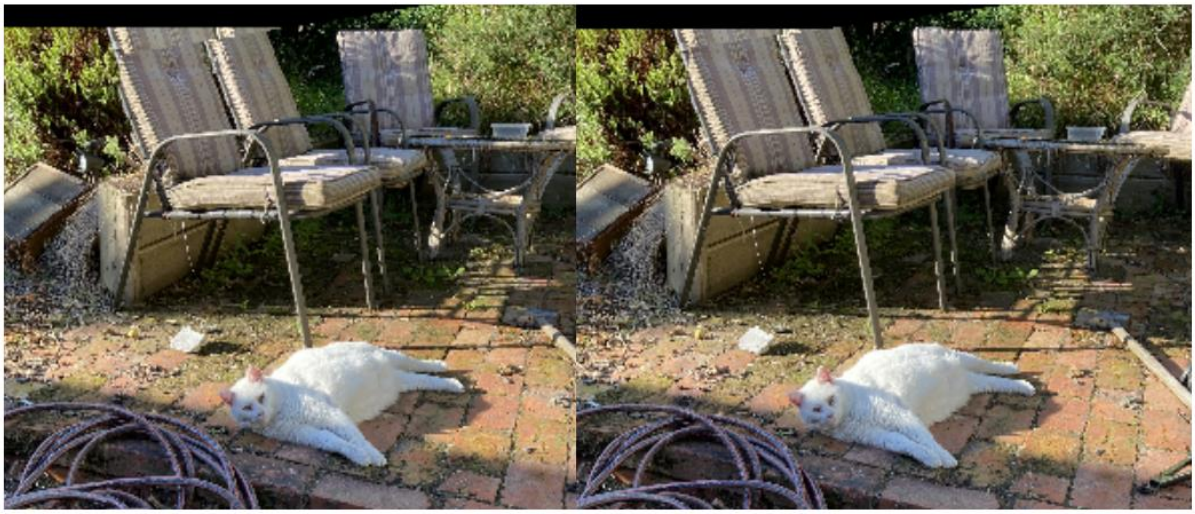
This table records how different the resulting homographies were from the baseline best case taken as a brute force approach on the image with full information. At the bottom is a record of when the resulting homography completely failed to perform a mosaic, in most cases by creating a warped meaningless image or in the case of B, not enough matches were generated to even attempt to create a homography.

I took this further than the previous section in terms of scale, measuring the time taken at these scales holds very limited usefulness as it is essentially instant, usually providing times of 0.001 seconds or just 0.

This value is not without flaw and is not useful as an objective measure. Because it is taken from the whole homography without normalization more weighting is placed on some aspects than others, specifically; far more weight is applied to translation than to rotation. However, these values do yield meaningful information when compared to each other, they all possess the same flaws so are useful as a rough estimate of variance.

A and B are unexpected to the point where I was questioning my methodology, it shows the homography getting more accurate as the scale reduces, this is the opposite of what all logic would say.

Once the Frobenius norm value reaches some threshold it is evident that the homography is so far from correct that it generates a meaningless mosaic, this was seen in C, D and E where the value exceeded 100 and the result was nonsense. This does enable a non-visual check to determine if a complete failure has occurred, if the value reaches some value greater than 25 and less than 127 the result was in all cases broken.



1 0.5

Above is a small section taken from image set A, where the resulting homography taken at both 0.5 and 0.25 scales was massively different from the homography at 1. Out of all the experiments this seems to be the most interesting, looking at the visual quality of the final mosaic result it appears that the homography taken at 1 is less accurate than at 0.5, this is evidenced by the greater amount of shearing visible in the top of the closest chair. So, what we are seeing here is not the high value in Frobenius norm representing an inaccuracy as it strays further from the best case homography but rather it is becoming more accurate. This is not the expected behavior and is not consistent with any of the other sets. To me this suggests; if the result of a mosaic is not satisfactory, trying to generate a homography from a scaled down image has the potential to improve it in some cases.

1 0.25 0.1

Above is small section taken from image set E. I have not included 0.5 as a scale in this example as it is completely non distinct from 1. As you can see looking at the 0.25 image, some small errors do start to appear at this scale. This is most obviously pronounced in the framing, there is minor shearing that can be noticed if examined closely. At 0.1, a 100x reduction in pixels using a 204x153 image for the homography we can see that while it did work, this shearing is now obvious.

Experiment 1 Conclusion:

From this I believe I can conclude that attempting to use compressed images in order to speed up the mosaicing process is a futile endeavor and this is ignoring the time taken to perform the compression itself. Using compressed images results only in more feature ambiguity and ultimately more time taken.

Image scaling however was much more interesting, this made the process faster by extreme factors and while there was a reduction in the quality of the homography this only became a visual issue at quite large factors in most cases. There are some outliers here specifically image set A at 0.5 generating an inordinately large Frobenius norm which seems to represent an inaccuracy not in the scaled images but in the "best case" homography itself. Based on the information gained from this experiment I believe it is safe to say that image scaling should be a staple part of any robust mosaicing pipeline. In some edge cases it can result in a better more accurate to reality homography but in general the main benefit is massively increased speed. This does usually come at the cost of accuracy but if there is need for real time mosaicing, for example taking images from a moving camera to create a scene, and perfection is less important than speed, scaling the images is likely essential.

Experiment 2: RANSAC Threshold

In this experiment I will be exploring how changes to the RANSAC threshold affect the accuracy of the resulting homography.

RANSAC works by performing some number of attempts to draw a conclusion about the homography from a small sample of the data and then determining how much of the rest of the data agrees with that estimation. The RANSAC threshold determines what the definition of agree is, we are dealing with inherently imperfect data, so some tolerance is generally introduced to allow for other data points to be considered to agree if they are close enough.

The default value is given as 3, and as it is virtually impossible to create an objectively perfect homography in terms of numerical data I will be measuring it as a distance from the homography generated at 3. In terms of more qualitative data, I will be performing a visual estimation of which homography was the best of the sampled thresholds, in this I will be primarily be looking for amount of shearing across the image seams as this is the most clear visual interpretation of accuracy.

The values I intend to test are 0,1,2,3,4,5 and 10. 0 and 10 are introduced as extreme cases and my expectation is these will yield messy results at best, especially 0 as with no inaccuracy allowed, I expect that no points will ever agree with the estimation, the consensus will be just the points that were randomly generated and it will just use the first attempt which is likely wildly inaccurate and unpredictable.

All other aspects will be controlled, feature detection will use SIFT, feature matching will be brute force, feature filtering will be done as knn with k=0 and a threshold of 0.8. As there is an element of randomness involved each experiment will be run 3 times and an average will be taken.

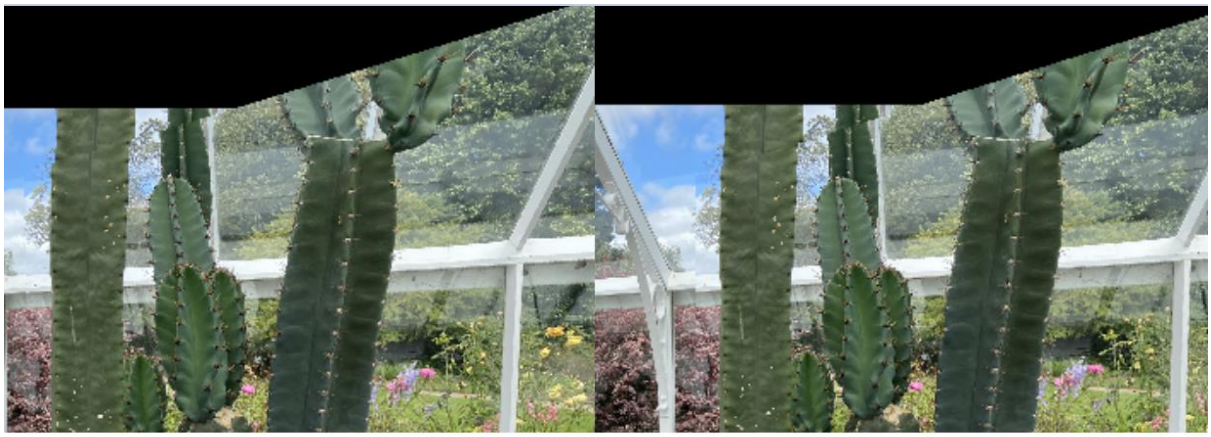
Experiment 2 Data:

Frobenius norm values compared to RANSAC 3

	A	B	C	D	E	Averages
0	0	0	0	0	0	0
1	5.25	5.04	9.96	6.92	1.27	5.688
2	1.12	4.82	15.36	5.07	2.59	5.792
4	0.32	1.21	7.07	7.77	0.33	3.34
5	0.39	1.55	13.98	8.12	1.04	5.016
10	12.26	4.55	8.19	7.65	3.16	7.162

This table shows the Frobenius norm values taken from a homography taken at RANSAC thresholds 0,1,2,4,5,10 and compared to a homography generated at a RANSAC threshold of 3 across each image set.

From this data it is clear that changing the RANSAC threshold does in fact change the resulting homography, although in most cases not by much and determining if this is an improvement or not is challenging.



3 2

This is a comparison between RANSAC 3 and RANSAC 2 on image set C, the most pronounced difference in terms of how much the homography changed. What is not clear is which of these is better, both are imperfect in different ways. The left most cactus has more shearing at 3 and the top of the cactus to it's right has more shearing at 2.

On average it appears that tightening the threshold has more of an impact than loosening it, but not in all cases, in image set D we saw the opposite behavior.

With respect to the introduced extreme cases, it seems that the RANSAC function agrees that a threshold of 0 would produce nonsense, so it doesn't allow it and just defaults to 3. Disappointing as it would be interesting, but this is probably sensible behavior. 10 did not end up creating as extreme of a result as I expected, it did not even result in the greatest change in absolute terms.

Experiment 2 Conclusion:

This experiment did not yield any particularly unexpected information. I can say with certainty that changing the RANSAC threshold does change the final homography but whether this change is positive or negative for a given image set with all else being the same is harder to draw a concrete conclusion about. Based on this I would say that if the goal is to generate the best quality mosaic possible and time is not a factor, trying different RANSAC thresholds and visually evaluating each to determine which to keep is probably a sensible step to take.

I found that running 3 tests and averaging was not necessary in this case, every result was identical with 0 variance. This is a limited sample size but I believe it is sufficient to say that it is likely that the RANSAC algorithm is performing adequate samples to reduce the effect of randomness.

Final Thoughts:

The mosaicing process is generally imperfect and challenging to measure levels of success or failure. This proved to be an obstacle in both of my experiments, without being able to determine a "perfect" homography it is hard to say with any amount of certainty whether a change made caused an improvement or not. It's easy to say it made a change, and even to say that the change was significant but beyond that generally devolves into a subjective visual inspection.

This clearly is not ideal, and a solution is not clear. Perhaps taking a large enough group of people and having them rank the resulting mosaics from best to worst could yield some more information than my own personal analysis.

A sample size of 5 sets of images was limited, more is obviously better, but time is a consideration. This number could be considered acceptable if ultimately, they all agreed with each other, but this is not the case. This is most evident in the scaling affects of homographic accuracy on image set A, this is a single data point that suggests it is possible to generate a more accurate homography from a scaled image. This is inadequate to even speculate about how often this occurs, and I am left wondering if I was extremely lucky to even find an example of this or if perhaps this is more common. Massively increasing the sample size should reveal more data on this phenomenon but once again evaluating improvement requires visual inspection.

With respect to the images themselves, they were all taken by hand using an iPhone, this creates an additional near certainty of errors resulting from deviation from perfect planar or rotational movement. This could be corrected by using a mechanical system such a tripod and a mechanism to enable perfect movement. With this we could potentially enable a mathematically correct homography to be generated but would no longer be representative of most types of real-world applications. In essence we would be trading one kind of error for an unrealistic scenario, both of which are likely to yield sub-optimal information about real world applications.

It would also be interesting to explore image set A in more detail, specifically regarding it's features and the relatively poor quality homography generated at full size. Is there some attribute of these features that makes them different from those at 0.5? Are they on average bigger, have some bias in direction, are they clustered? Answering these questions may yield a method of determining the quality of a homography before it is even computed and ultimately lead to an algorithm that can tell us something about feature quality. This would be extremely useful as it would allow the overall program to go back and try again with different settings, or it could allow a user to know what error is expected.