Assignment 1

Mosaicing

Due: 11:59pm, Friday 1st April

> Through the labs you will develop a simple mosaicing application in OpenCV. In this assignment you will need to conduct some experiments to see the effects of different choices in the mosaicing pipeline. You will need to submit a report on your experimental design, results, and analysis.

In this assignment you are asked to perform and report on experiments. This might be a bit different to what you've done before (in computing at least), so we'll give some general advice here. Designing and running good experiments can take a bit of practice, so feel free to discuss your plans with us ahead of time to get some more advice.

Selecting Images

Experiments in Visual Computing field are complicated by the fact that there is no such thing as a 'typical' or 'standard' image. This means that selecting images that give a fair comparison of different techniques can require a bit of thought. There are a number of factors that you might want to consider when choosing what images you want to use in your experiments:

- Scene type – different types of scene have different image characteristics – the textures you see in images of a forest are quite different to photos taken inside an office block.
- Number of image pairs – a single image pair won't tell you whether your results are general or specific to the example you've chosen.
- Image size – larger images are generally going to have many more features than smaller images. They will take longer to process, but also some methods might be better or worse when applied to smaller or larger images.
- Image format and compression – Image formats that use lossy compression mean that the image data may be altered. How (if at all) does this affect the mosaicing process?
- Image geometry – remember that mosaicing only works in two cases (near-planar scenes and rotation-only camera motion). Most of your test images should follow this, but it could be interesting to see the effect of small deviations from these assumptions.

You do, however, have to keep practicality in mind. If you have five different scene types, with 20 image pairs each at three resolutions you'll have $5 \times 20 \times 3 = 300$ image pairs to process. With a script to run the tests and collate the results, that's probably doable, but processing by hand would be tedious. If you add in four different image formats (say, uncompressed PNG and three different JPEG quality levels), and two different image capture situations (planar scenes and rotation only) you'll be up to $300 \times 4 \times 2 = 2400$ and there'll be even more processing and a lot of results to write up.

> The range of possible experiments can easily get out of hand. For this assignment it is perfectly OK to limit the factors that you want to experiment with. You should, however, use a range of input images so that you can draw general conclusions.
>
> Think about what you can re-use between the experiments. A single set of test images can probably be used for all of the tests, for example. Can you also use the same evaluation methods or measures?

Designing and Running Experiments

For each part of this assignment you will need to run an experiment. The starting point is a question or hypothesis – which are often the same thing expressed in different ways. A hypothesis is some theory or idea that you want to test, such as:

> SURF features are faster to compute than SIFT features.

This could be expressed more neutrally as a question:

> Are SURF or SIFT features faster to compute?

Or you could make a more general question:

> Which OpenCV feature detector is fastest?

Once you have your hypothesis or question, you can design an experiment to test or answer it. A well designed experiment will have different outcomes depending on whether the hypothesis is true or false (or what the answer to the question is). In this case you would try out different feature detectors and measure how long they took to compute.

You also need to make sure that the experiments are fair, and that they are going to measure what you are interested in. In this case you would have to use the same images for each detector, run in the same way, so that the only difference between the runs is which feature detector you are using. You should also decide on your data and approach before making any measurements so that you aren't trying to get a specific result.

You also need to decide exactly what you mean by "faster to compute" – is it the feature detection or the whole mosaicing process? A feature detector that takes longer to compute but produces fewer features that are more reliably matched may have performance gains elsewhere in the process.

Presenting Your Results

Finally you need to present your results in a report. For each experiment you need to explain clearly what your question or hypothesis is; what experiment you designed to test that hypothesis; what measurements that experiment produced; and what that tells you. This last bit is very important – you need to be able to interpret your results accurately and understand any limitations in your experiment.

Also think carefully about how you present your results. If you have an average time for each method then that's OK, but it doesn't tell you how much variation there is. Reporting a range of results (say, a standard deviation, or just a range of values from minimum to maximum) is better. Visual presentation can also be useful in some cases – even if it is just a table of numbers with the best values highlighted. Be careful to choose a method that clearly shows the data – your aim is to clearly present your findings, not to confuse us with flashy graphics.

In each of the following sections there are several suggestions given for questions that could be asked and different approaches. **You only need to ask one question and do one experiment per part.** A single clear question with a good experimental design and thoughtful discussion in each part is sufficient to achieve maximum marks. Extra credit will **not** be given for additional experiments beyond what is required.

1.1 Feature Matching Methods

The first step in the mosaicing pipeline is to detect features in the images, and we'll be using SIFT features as they are widely used and reliable. Once features have been found in both images, the next step is to match them on the basis of some descriptor. There are two main approaches implemented in OpenCV – Brute Force matching and fast approximate matching. Again, you should conduct an experiment to compare the two. Examples of questions you could ask here are:

- Under what circumstances is approximate matching faster than brute force matching?
- How does the use of the approximate method affect the feature matching accuracy?
- How does lossy image compression affect the quality of the feature matching?

Make sure that your experiment can answer the question you pose. For example, how can you measure how reliable matches are, or how accurate a homography is? Also remember that the choice of input images might affect the results – one feature matcher might be faster for large images but slower for small ones, or repeating textures might be more or less of a problem. It is important to consider a range of different images for this reason.

Task 1.1: Design, conduct, and report on an experiment to compare Brute-Force and FLANN-based matching for image mosaicing in OpenCV.

1.2 Homography Estimation Methods

Finally, we estimate a homography based on the matching features. Not all of the feature matches will be correct, so a RANSAC process is used to remove outliers based on some threshold, and we would like you to experiment with different values of this threshold. Questions you might ask at this stage are:

- How does the choice of RANSAC threshold affect the speed of homography estimation?
- How does the choice of RANSAC threshold affect the proportion of inlier matches?
- How does the choice of RANSAC threshold affect the accuracy of the homography?

In this part of the assignment you should just use SIFT features and a single matching method (whichever you think best).

> **Task 1.2:** Design, conduct, and report on an experiment to measure the effect of the RANSAC threshold on image mosaicing in OpenCV.

1.3 Deliverables

For this assignment we want you to produce a report describing your experiments. Your report should be 4-6 pages long and for each experiment should clearly tell us:

- What your question or hypothesis is.
- How you designed an experiment to test or answer your hypothesis or question.
- What the results of the experiment were.
- What that told you about your question/hypothesis.

> You are asked to discuss the results of the experiments you do. It is important to note that the results of the experiment aren't the important thing here – your ability to interpret them and explain what is going on is. For example, suppose you are trying to determine which of two methods is faster. The first takes 1.125 seconds to compute, and the second takes 1.127 seconds. It is much better to just say that your experiments don't show any practical difference between their speeds than to try and claim that the second is clearly superior.

A suggested report format is available as part of this assignment, but you don't have to follow that if you think there is a better way to clearly communicate your work.

1.4 Marking Criteria

You will be marked on the following criteria:

- The challenge posed by the questions that you ask in each experiment.
- The design of the experiments to answer these questions.
- The clarity of the results of those experiments.
- The insight provided in your discussion of those results.
- The overall quality of the presentation of your report.

These will be ranked on a scale of Poor (a failing or C- grade); Weak (C or C+); Average (B- or B); Good (B+ or A-); or Excellent (A or A+). The final mark will be determined from the overall balance of rankings across the criteria – an exact weighting is not given as Good or Excellent work in some regards may well outweigh minor deficiencies in others. The following sections explain what would constitute an Poor, Average, or Excellent grade in each category. Weak and Good criteria can be interpolated from these.

Challenging Questions

A Poor Report will fail to ask clear questions, or ask only very simple questions.

An Average Report will have a clear question for each experiment, and start to explore more complex ideas.

An Excellent Report will ask insightful questions for all experiments, and answering at those questions will be challenging.

Experimental Design

A Poor Report will have poorly conceived experiments that do not clearly address the questions posed and which have flawed designs or inappropriate data sets.

An Average Report will have experiments that address the questions raised, but which may have some limitations in their design, data sets, or evaluation metrics.

An Excellent Report have clearly designed experiments that directly address the questions posed, using a range of input data and appropriate evaluation metrics.

Clear Results

A Poor Report will have incomplete or disorganised results presented in a way that is difficult to understand or interpret.

An Average Report will have all of the results presented but may lack guidance for the reader, lack visual representations, or have some aspects that are unclear.

An Excellent Report will have results that are clearly described and presented with thoughtful use of figures, charts, and tables.

Insightful Discussion

A Poor Report will have a lack of discussion, or draw conclusions that are not supported by the evidence provided. Weaknesses in the approach taken will be ignored or dismissed.

An Average Report will draw conclusions from the evidence provided, but may lack insight as to the reasons for those results. Weaknesses in the approach taken will be clearly acknowledged.

An Excellent Report will relate the results to clear conclusions, providing sound explanations for observed results, and give insight into any unexpected results. Weaknesses in the approach taken will be acknowledged and ways to remedy them suggested.

Presentation Quality

A Poor Report will be difficult to read, poorly organised, and have many spelling and grammatical errors.

An Average Report will be well organised and generally readable with few spelling and grammatical errors.

An Excellent Report will be elegantly presented with strong visual elements. Use of clear simple language will make the report easy to read and understand.