COM31006: Salient Feature, Matching and Image Stitch: Assignment (50%)

Prof. Jungong Han

Deadline: 3pm, 20 May

Introduction

The assignment will involve using OpenCV and Python to implement salient feature extraction, feature matching and image stitch methods. Feature descriptors will be necessary for matching features, and a transformation model will be required for image stitching. Upon completion of this assignment, you will be able to: a) Utilize visual features to represent images, and b) Develop a program that integrates or fuses a pair of images.

Descriptions and Requirements

In this assignment, your task is to develop code for detecting discriminative features that exhibit reasonable invariance to translation, rotation, and illumination within an image. Subsequently, you will identify the best matching features in another image. These matched features, or correspondences, can then be utilized to seamlessly stitch two images together into a panorama.

To aid in visualising the results and debugging your program effectively, you will need to develop a user interface. This interface should display the detected features, their best matches in another image, and the generated panoramic image.

You must satisfy ALL the following requirements.

1. Image Dataset Construction

   You will be required to collect or capture a minimum of five pairs of images, including one provided by us in Figure 1. It's important to ensure that each pair of images has at least 70% overlapping content. All tests and comparisons will be conducted based on this image dataset.

2. Feature Detection

   In this step, your task is to identify points of interest in the two images using both the Harris corner detection method and the SIFT feature point detection method. You will

Figure 1: Example of a pair of images for image stitch.

then **display** these points, including their location and orientation, in a window on your user interface. In the report, you will be expected to not only describe the implementation details of both methods but also utilize experimental results to compare and contrast the differences between them.

3. Feature Description

   Once you have identified the points of interest, the next step is to create a descriptor for each feature centered at these interest points. This descriptor will serve as the representation used to compare features in different images to determine if they match. In this step, you will implement two descriptors: the SIFT descriptor and a low-cost descriptor, such as a low-dimensional descriptor or binary descriptor. Regarding the low-dimensional descriptor, you may need to conduct some research to determine the most suitable option for your specific needs.

4. Feature Matching

   Now that you have successfully detected and described your features, the next step involves coding to match them. This entails the process of identifying the best matching feature in another image given a feature in one image. One approach to achieve this is as follows: compare two features and compute a scalar distance between them. The feature with the smallest distance is considered the best match. For the SIFT descriptor, you will be required to implement two distance functions:

   (a) Sum of squared differences (SSD): $d(X, Y) = \|X - Y\|^2$, where X and Y are vectors of two descriptors, and $\|\cdot\|$ is the L_2 norm.

   (b) The ratio test: Find the closest and second closest features by SSD distance. The ratio test distance refers to the ratio between these distances, calculated as the SSD distance of the nearest feature match divided by the SSD distance of the second nearest feature match.

   In the report, you will be expected to i) provide detailed descriptions of the implementation for both descriptors, as well as the feature matching scheme; ii) display the matches between the two images. An illustrative example can be found in Figure 2; iii) conduct a comparative analysis of the two descriptors, focusing on matching accuracy and matching speed. iv) compare the two distance measurements in terms of the matching accuracy.

Figure 2: Example of feature correspondence.

Figure 3: Example of the image stitch.

5. Image Stitch

   Find the homography that relates the two images, i.e., the left and the right images. Warp the right image using the calculated homography, with an output width of left image width + right image width. Copy the content of the left image over the warped image; use slicing to indicate where in the warped image you want to put the left image. When displaying the final result, you should have something like Figure. 3.

   In the report, it is essential to provide a comprehensive implementation breakdown. Test all images within your constructed dataset.

UI and Visualisation

We expect a simple user interface (UI), where the primary functions are accessible via clickable buttons, and the generated results are displayed in a window. You are specifically required to show the results of Feature Detection, Feature Matching, and Image Stitching.

Deliverables and Marking

You should submit a zip file containing a copy of your program code, image files and a report that describes your implementations and results via Blackboard.
The allocation of marks between the program code and the report is 40% for the former and 60% for the latter. To validate the functionality of your program, your code will undergo comprehensive testing and execution on either the Windows or Linux platform, both of which are acceptable. The marking schemes for both the program code and the report are outlined as follows:

The program code

- Code structure, including comments and layout of the UI, use of data structures and use of functions (20%).
- How well the application works and the completeness of this, including quality and creativity of the work (80%).

The report

The descriptions of the following aspects will be evaluated: Image Dataset Construction (10%); Feature Detection (20%); Feature Description (20%); Feature Matching (30%); and Image Stitching (20%).

- Knowledge and understanding of relevant material – Do you demonstrate knowledge and understanding of what you are writing about? Is deeper understanding demonstrated by comparing and contrasting ideas? Is your information accurate, or is some of the material misleading or even incorrect, demonstrating that you don't understand it?
- Organisation, clarity of expression and illustrations – Is the report tidy and organised, with a good flow of information? Are all points in the writing expressed clearly and succinctly, and supported with the use of diagrams? All illustrations/diagrams should have a figure number and a caption and be referred to from the main text.

Practical considerations

- Unfair means

  The standard Department rules for the use of unfair means will be applied, as described in the undergraduate student handbook: https://sites.google.com/sheffield.ac.uk/comughandbook. We are aware that there are lots of tutorial sites on the Web. Do not copy them since that would be plagiarism. Instead, just learn from them.

  Do NOT use ChatGPT or any other AI tools for this assignment. This will be treated as plagiarism for the purposes of this assignment.

- Late handin

  Standard Department rules will be used for late handin – see the undergraduate student handbook: https://sites.google.com/sheffield.ac.uk/comughandbook/.