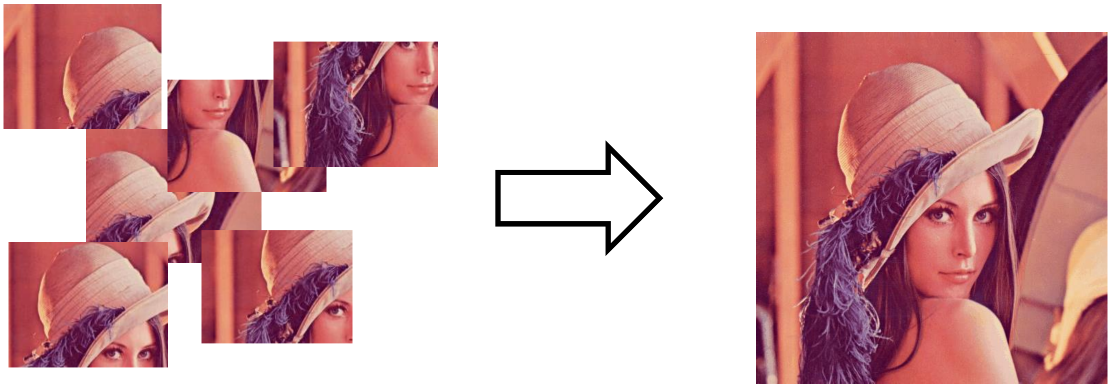
ELEC 474 Machine Vision Project

Autostitch:

Matching and Merging an Unordered Collection of Images



Due: Wednesday December 1st 2021, 11:59 pm

Introduction

The task is to create a panoramic image from a series of overlapping images of a scene. This task is divided into four steps:

> Step 1: The first step is to extract features from the images, and automatically establish feature correspondences between image pairs.
>
> Step 2: The second step is to use these correspondences to estimate transformations between each image, and establish the most likely transformations between the image pairs.
>
> Step 3: The third step is to apply these transformations to compose a single merged composite image from all images.
>
> Step 4: The final step is to document your solution and submit it to OnQ by the deadline.

Step 1: Match Features

The starting point is an unordered set of *N* images. The objective of this step is to extract features from each image, and establish which (if any) of the *N-1* other images contain a sufficient number of good matches so as to indicate that the two images partially overlap. You can use any feature that you prefer (e.g. ORB, SIFT, SURF) or some combination thereof.

You will need to develop a ***match metric*** that can be used to discriminate between good and bad image matches. For example, this match metric could take the number of matching features between images into account, as well as the quality of their match. Keep in mind that each image will only overlap with a small number of other images in the set, so ideally this match metric will score high for the images that do have overlap, and low otherwise.

Step 2: Estimate Transformation

For those image pairs that have a high enough match metric score from Step 1, calculate the transformation between them.

Step 3: Merge Images

Apply the transformations calculated in Step 2, and create and store a single merged image from the *N* images in the set. Apply both geometric and radiometric transformations, so that the resulting merged image appears relatively seamless.

Step 4: Documentation

Write a brief (no more than 5 page) report describing your solution and results. Include the following information:

- Your name and your student number.

- An inventory of the code that you developed (i.e. a list of all source modules). Indicate which of the submitted code modules are original, and which (if any) are not.

- A declaration of originality, indicating the extent that you are the author of the submitted solution.

- A description of the feature extraction method and matching metric used in Step 1. Describe any specialized data structures and algorithms that you developed and applied.

- Details of the transformation estimation method developed in Step 2.

- A description of the transformation application and image composition method developed in Step 3. Include details of any geometric or radiometric transformations applied.

- Any further relevant details of the developed software (optimization methods, data types of interest, etc.).

- A description of the tests that you executed and the results that you obtained. Indicate the time performance of the method (e.g. how long did it take to execute for each tested image set)? Also include a listing of the matching metric values between each image, for each data set.

- A discussion of the correctness and effectiveness of your solution. Include here a declaration of the success of your system, i.e. the extent that you considered your solution to solve the stated task. Also include any limitations that you observed.

- A proposal on how you would improve your solution, if you had more time.

Deliverables

Your deliverable should include the following material:

D1 – Complete development directory, including all source code and external resources (e.g. images), in the format described below (under Submission Format).

D2 - Documentation (as detailed in Step 4 above) in pdf format (called Project_Report.pdf). Include your name and student number on the first page.

D3 – Resulting merged images (as indicated in the directory structure below).

D4 – Completed Self Assessment Spreadsheet.

D5 (Optional) – Any additional material that you think is relevant. An example could be any extra image stitching samples that you tested.

Submission Format

Your code should run without any alteration/editing/debugging by the grading TA. Submissions will be run as-is from the main file. Make sure any resources (e.g. lena.jpg) are included and correctly referenced to run from the submitted zip archive. Marks will be deducted for non-conformity to the submission instructions on all submissions.

Please submit the deliverables as a single zip file named

Autostich-<yourLastname>.zip

with the following folder structure:

```
autostich-<yourLastname>.zip

|-- D1/Code/

|        |--- <yourCode>.ipynb (or .py)

|        |--- <etc..>

|-- D2/Autostich_Report-<yourLastname(s)>.pdf

|-- D3/Results/

|         |---<Resulting Composite Image 1>.jpg

|         |---<Resulting Composite Image 2>.jpg

|         |--- <etc..>

|-- D4/Autostich_SelfAssessment-<yourLastname(s)>.xlsx

|-- D5/Extra/
```

The **`Extra/`** directory can be used to contain any additional material that you feel is relevant to your assignment. For example, if you acquire new image sets for testing, you can include them here.

(On Windows, you should be able to achieve this by right-clicking the containing folder and selecting "Send To" -> "Compressed (zipped) folder".)

Marking Scheme

Step 1:	3 marks
Step 2:	3 marks
Step 3:	4 marks
Documentation:	5 marks
TOTAL:	15 marks

Late submission policy

Late submissions are accepted, at a deduction of 5 marks per week. The following table illustrated the late-submission deduction policy:

Submission Period	Mark Deduction
up to 11:59 pm, 1st December	0
12:00 am 1st December to 11:59 pm 8th December	5
12:00 am 9th December to 11:59 pm 15th December	10
After 12:00 am 16th December	15

Project Rules

1. Do not cheat.
2. Do not use any code from the **`cv::Stitcher`** class. The goal is essentially to create your own custom version of this functionality. You're free to read the documentation about this class when you're developing your own solution method, but don't use (or copy) any of the source code directly. You can use any other OpenCV classes and methods.
3. All work is to be done individually. You can discuss solution approaches with your friends, but do not share any code.
4. All code that you submit should be original. Do not mine the internet for solutions. The coding task is relatively straightforward, and it will be easier (and of greater benefit) to develop the solution from scratch (as is required) than to start from another solution (which is not allowed). You are allowed to use source code that you developed as part of a previous lab solution. The incorporation of any other source code is not allowed.
5. If you have any questions, then please ask them during the Tuesday evening labs. For fairness, neither the instructors nor the TAs will answer questions about the project by email, or outside of the labs.
6. Submit your complete solution (described above under *Deliverables*) through OnQ, by the deadline. The system will not accept submissions after the late submission deadline (16th December).

GOOD LUCK!