

Image Processing – 67829 – Exercise 4

Due Date: 29.01 at 23:59

Version 1.0 - Last Update 10.01.2024

1 Task

In this exercise, you will need to blend between a low-resolution image and a high-resolution part of it using the material taught in class up until now. In the Moodle, under "Exercise 4" -> "Exercise Inputs" (found here) you are given 2 low-resolution images and 2 high-resolution parts of the image. Your task is to design an algorithm that takes such a pair and blends them with each other.

Example: In Fig. 1 you can see the two inputs and a possible output.

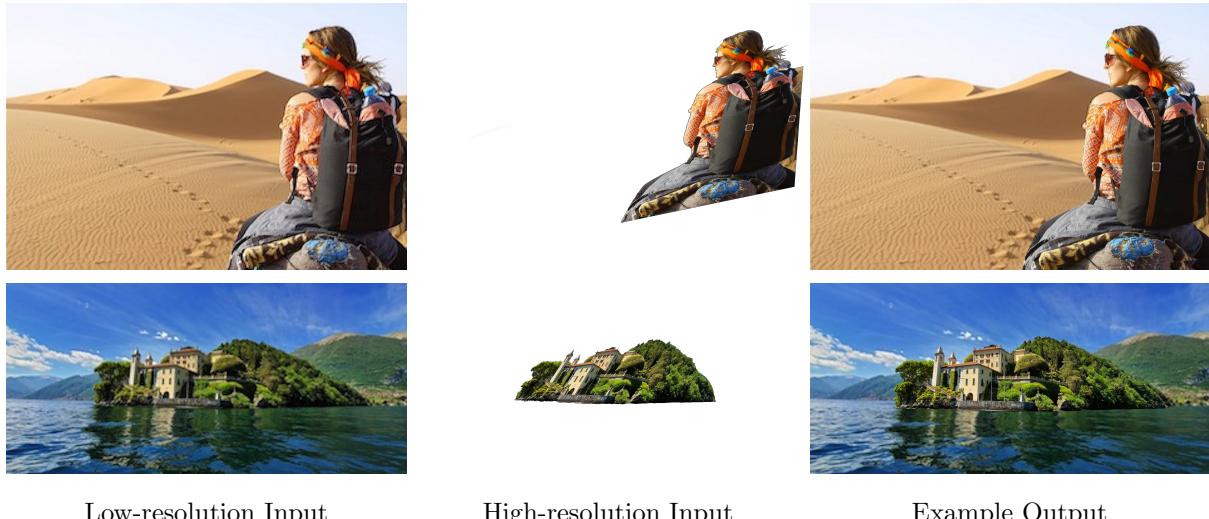


Figure 1: Zoom in to see the resolution differences

Note: You **must** use the following topics in your solution: feature extraction, feature matching, RANSAC, and image warping.

2 Submission

Submission instructions are given in the "Submissions Guidelines" document published on the course web page (here). Please read and follow them carefully. Any updates to those guidelines will be posted in the news forum, so be sure you follow the forum.

You must submit your code but there is no API you need to follow and there are no presubmission tests.

2.1 Report Guidelines

In addition to the code, you should submit a report describing your solution. The report must follow the following structure and address the topics below. We provide an English and Hebrew Google Docs template (you need to copy it to use it). In case you choose not to use it, please maintain a similar structure (font size, same sections, same number of figures, same number of pages, etc.), in particular, the report should be no longer than 5 pages and include the following sections and topics:

1. Introduction
 - (a) In your own words, state the goal of the exercise and what were the main techniques (i.e. an idea or concept you've learned in class, not a technical tool like numpy) you've used to solve it.
2. Algorithm
 - (a) Clearly describe the algorithm you've used to solve the exercise. If using more than one "sub algorithm", describe and elaborate on each one **and** on the entire final algorithm (i.e. describe the conceptual steps and building blocks, e.g., if the algorithm is how to make coffee, the main algorithm can be using a coffee maker to make coffee and foamed milk. The sub algorithms can be an algorithm that describes making the espresso and an algorithm that describes how to foam the milk).
 - (b) For each "sub algorithm", clearly describe its inputs and outputs.
3. Implementation Details
 - (a) Describe your implementation of the algorithm (i.e. describe the actual implementation of the steps from section 2.a, using the same coffee example, here you would describe the type of coffee machine, the type of coffee, how much sugar, water to milk ratio, etc.).
 - (b) Specify the parts that you implemented from scratch and those that you've used functionality from an existing library. What libraries did you use and why did you choose these?
 - (c) Describe and justify any necessary hyper-parameters, thresholds, or other choices used in your algorithm.

(d) Discuss any challenges faced during implementation and how they were addressed.

4. Visual Results

(a) For each “sub algorithm”, present in a visual manner the outputs of it (and if needed, also the inputs).

You should also explain in writing your visualization. If you have tried different hyper-parameters or algorithms, compare them,

(b) Present your final results, explain them and point out any flaws they may contain.

5. Conclusion

(a) Summarize your key findings and insights.

Your final submission should be a tar file containing a PDF named “ex4.pdf”, a python file named “ex4.py”, and a “requirements.txt” file with your dependencies. To create a tar file you can run the following command:

```
tar -cvf ex4.tar ex4.py ex4.pdf requirements.txt
```

3 Grading

Your exercise will be graded based on a manual inspection of your report (and code). As mentioned above, there will be no presubmission tests and no automatic tests.

Good luck and enjoy!