Image Processing – 67829 – Exercise 4

Due Date: 16.01 at 23:59

Version 1.0 - Last Update 24.12.2024

1 Task

In this exercise, you will implement the "Stereo Mosaicing" algorithm presented in class. The input of such an algorithm is a video (i.e., a sequence of images) scanning a scene from left to right due to camera rotation and/or translation. It is suggested to compute rigid transformations between frames, and we assume that there is a significant overlap between consecutive frames. The algorithm's output is a video of n different *panoramas*. When the video is not left-to-right, e.g. trees, pre- and post-processing rotating frames may be useful.

For your convenience, in Moodle, under "Exercise 4" -> "Exercise Inputs" (found here) you can find several example input videos. Under "Exercise 4" -> "Example Outputs" (found here) you can find a few example outputs.

Note: You **must** follow the steps described by Shmuel in the lecture (recording found here).

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.

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

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 7 pages** and include the following sections and topics. **Failing to adhere to these guidelines will result in point reduction.**:

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.
   (b) For each "sub algorithm", clearly describe its inputs and outputs and visually present them. You should also explain in writing your visualization.

   Note 1: For some steps, your figures should likely include some visualizations on top of the images. For example, if you want to show detected interest points, you should plot their locations over the images. Similarly, if you present matched features, you should plot the matchings. For an example, see Fig. 1.

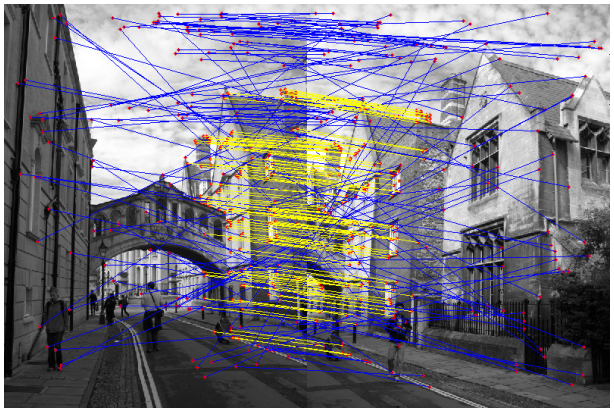

Figure 1: Example of a possible visualization of the outputs of a possible "sub algorithm".

   Note 2: Presenting the inputs and outputs for a small number of frames (sometimes as little as 2) is enough. You can also use the debug images to present these inputs and outputs.
3. Implementation Details
   (a) Describe your implementation of the algorithm and each of the "sub algorithms".
   (b) Specify the parts that you implemented from scratch and those that you've used functionality from an existing library.

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

(b) You should present results for 1 dynamic mosaic (e.g., trees or iguazu videos) and one changing viewpoint mosaic (e.g., boat or kessaria videos). **Note: present the results by showing the first, middle, and last frame of your result. You will submit the full videos separately.**

(c) Record your own videos and run your algorithm on these videos. Present the input video and the result of running your code on it (again, using the first, middle, and last frames).

(d) You should record 1 "good" video and 1 "bad" video. A good video is one where the algorithm works as expected, a bad video breaks one of the assumptions of the algorithm and therefore the result would not look as expected. Explain the difference between these videos and what made the bad video fail. **Note 1: Since most phones record high-resolution videos, you should probably reduce the resolution and possibly the frame rate of the videos you record. Note 2: You can record either a changing viewpoint video or a dynamic video, the choice is yours.**

5. Conclusion

(a) Summarize your key findings and insights.

Your final submission should be:

1. A PDF named "ex4.pdf".
2. A tar file containing all the input of output videos of your submission. This should include:
   - `dynamic_input.mp4` - the input video for the dynamic mosaic (out of those provided in the Moodle).
   - `dynamic_result.mp4` - the output video for the dynamic mosaic.
   - `viewpoint_input.mp4` - the input video for the change viewpoint mosaic (out of those provided in the Moodle).
   - `viewpoint_result.mp4` - the output video for the change viewpoint mosaic.
   - `good_input.mp4` - the "good" input video you filmed.
   - `good_result.mp4` - the output video for the "good" video you filmed.
   - `bad_input.mp4` - the "bad" input video you filmed.
   - `bad_result.mp4` - the output video for the "bad" video you filmed.

To create a tar file you can run the following command: `tar -cvf videos.tar dynamic_input.mp4 dynamic_result.mp4 viewpoint_input.mp4 viewpoint_result.mp4 good_input.mp4 good_result.mp4 bad_input.mp4 bad_result.mp4`

3. A tar file containing 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 requirements.txt`

Note: The PDF, videos, and tar should be submitted to the respective submission in the Moodle.

3 Bonus - up to 4 points in the final grade

As mentioned in class, we will provide a bonus of up to 4 points in the **final grade of the course** to students that implement the forward panoramas effect. To submit the bonus you should submit a **separate PDF** that is **up to** 2 **pages** long and describes your solution. The PDF should include the (relevant) topics described above, there is no need to repeat the details from your main report, simply describe your solution and how it differs from the original algorithm.

Note: The bonus should be submitted to the dedicated submission in the Moodle. It will be grades by Shmuel, and may include an interview where you will show the capabilities of your system.

4 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!