CS 523.V – Spring 2023

ÖZYEĞİN ÜNİVERSİTESİ

Problem Set 3

Assigned: 16 May 2023
Due: 26 May 2023

What to Submit:

Create a folder named ps<*problem set number*>_<*YourLastName*> (for example, ps2_Ates), with the following structure and contents:

ps<*problem set number*>_<*problem number*>.py → The main script to run.

/input/ → Directory containing input images, videos or other data used

/output/ → Directory containing output image, videos or other data generated

ps<*problem set number*>_report.pdf → A PDF report file. (Include figures, discussions, methods, etc., but do not include any code.)

Zip your folder and submit on **LMS.** Submissions that do not obey the guideline above will **NOT** be evaluated.

Problem 1:

Unzip the PS3_data.zip file. The folder contains a sample Python file, two images and a homography transformation between the images. When you run the Python code, you will see that it finds SIFT, SURF and ORB features, feature descriptors, and matches. Your task is to evaluate the performance of these feature extractors/descriptors.

(a) Find the best matches.
(b) Calculate and compare the repeatability rates of the feature detectors/descriptors. (Note 1: The included homography gives you the ground truth points. Note 2: You can define the repeatability as the number of correct matches divided by the average number of points extracted in the input images.)
(c) Write a function to calculate the homography matrix, incorporating the RANSAC algorithm. Compare this matrix with the given matrix.
(d) Write a function to warp one image onto the other one using the given (or the calculated) homography matrix. The warped image should have the same size as the other one. Display the residual image.
(e) Write a function that stitches two images given the homography matrix. Display the stitched image.

Problem 2:

(a) Calculate the optical flow between two images using the OpenCV calcOpticalFlowFarneback() method. Warp one image onto the other one; display the images, the warped image and the residual.
(b) Use OpenCV functions to obtain the Shi-Tomasi corner points and track them using Lucas-Kanade method calcOpticalFlowPyrLK().

You may utilize the OpenCV tutorials:
https://docs.opencv.org/4.7.0/d9/df8/tutorial_root.html
https://docs.opencv.org/4.7.0/d9/d97/tutorial_table_of_content_features2d.html
https://docs.opencv.org/4.7.0/d4/dee/tutorial_optical_flow.html