Mobile Robots

Summer Semester 2023
Assignment 7

Due: 18.06.2024, 10:15

Exercise 1 SIFT Detector (4 Points)

OpenCV provides an implementation of the SIFT Detector. Take a look at
https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html

(a) Copy the code and run it. Hand in your self generated key point images (from *blox.jpg*) and state how many key points you found.
You may encounter some error messages on your first try. Copy the error messages and search for a solution until you are able to get the code running. (2 Points)

(b) Now run the code on the different images provided on moodle (*morgenstelle.png* and *morgenstelle_blurry.png*).
Hand in the key point images as well as the number of key points found for each image. Explain your observations and provide some reasoning. (2 Points)

Exercise 2 Essential Matrix (6 Points)

(a) In a stereo setup, a calibration was made and from the calibration, the homogeneous transformation matrix of the relative pose of the right camera with respect to the left camera was found as:

$$^{L}T_R = \begin{pmatrix} \cos\left(\frac{\pi}{60}\right) & 0.0 & \sin\left(\frac{\pi}{60}\right) & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -\sin\left(\frac{\pi}{60}\right) & 0 & \cos\left(\frac{\pi}{60}\right) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Compute the corresponding essential matrix E. (2 Points)

(b) By matching image features of both images you found multiple candidate points in the right image

$$p_{R,0} = (11.0, 19.0, 1.0)^T$$
$$p_{R,1} = (11.0, 20.0, 1.0)^T$$
$$p_{R,2} = (12.0, 19.0, 1.0)^T$$
$$p_{R,3} = (12.0, 20.0, 1.0)^T$$

for one point in the left image $p_L = (30, 50, 1.0)^T$. Which is the best candidate to match with the point on the left image? (2 Points)

(c) You have found a feature correspondence in the right image at location $p_{R,4} = (9.0, 27.0, 1.0)^T$. Transform the point into the left camera's coordinate system. (2 Points)

Exercise 3 RANSAC (10 Points)

In this exercise we will implement the RANSAC algorithm from scratch to fit different types of functions to a set of 2D data points. Please use the attached python script *ransac.py*.

(a) In the first exercise, we will simply fit a model to data drawn from the function $f(x) = 2x+1$. Read the data points from the given csv-file. For this, understand what the format of the data is. Now, plot the data. (1 Point)

(b) Implement the basic steps of the RANSAC algorithm like in the script. Proceed in these basic steps:

- Randomly sample a subset of minimal length (what is that) of the data points.
- Fit a line to the sampled subset using `np.linalg.lstsq`.
- Determine the number of inliers for this model. An inlier is specified by the maximum distance given in the script.
- Repeat the process for the given number of iterations.
- Select the model with the highest number of inliers.

(4 Points)

(c) Plot the data points again, but this time, color the inliers and outliers according to the best model found by RANSAC. (1 Point)

(d) Now switch to the script *circle_ransac.py*. Here, we will be fitting a circle with center at the origin. Again, read the data points from the given csv-file. Plot the data. (1 Point)

(e) Reuse the logic for the RANSAC algorithm from before except for the circle fitting. Implement the function `fit_circle`.
Hint: Solve $Ac = b$ *with* $A = \begin{pmatrix} x & y & 1 \end{pmatrix}, b = x^2 + y^2$.
Then use $c = \begin{pmatrix} 2x_c & 2y_c & r^2 - x_c^2 - y_c^2 \end{pmatrix}^T$, *where* r *is the radius of the circle and* x_c, y_c *is its center.* (2 Points)

(f) Again, plot the data points, coloring the inliers and outliers according to the best model found by RANSAC and the best fitting circle. (1 Point)