376.054 Machine Vision and Cognitive Robotics (VU 4,0) 2018W

Exercise 4: Plane fitting with RANSAC

Download files for exercise 4

Starting with this exercise, you will work with *3D pointclouds* recorded with a depth camera instead of 2D images as input data. The goal of this exercise is to find the dominant plane (i.e. the floor) in the given pointclouds, as well as extracting multiple planes from more complex scenes. A common method for that is the RANSAC (Random Sample Consensus) algorithm, which has already been discussed in the lecture. You will only need to process the spatial information of the pointcloud, colour information is not used yet. An example pointcloud of the dataset is shown in Figure 1.

For this exercise the matlab Computer Vision System Toolbox is required.

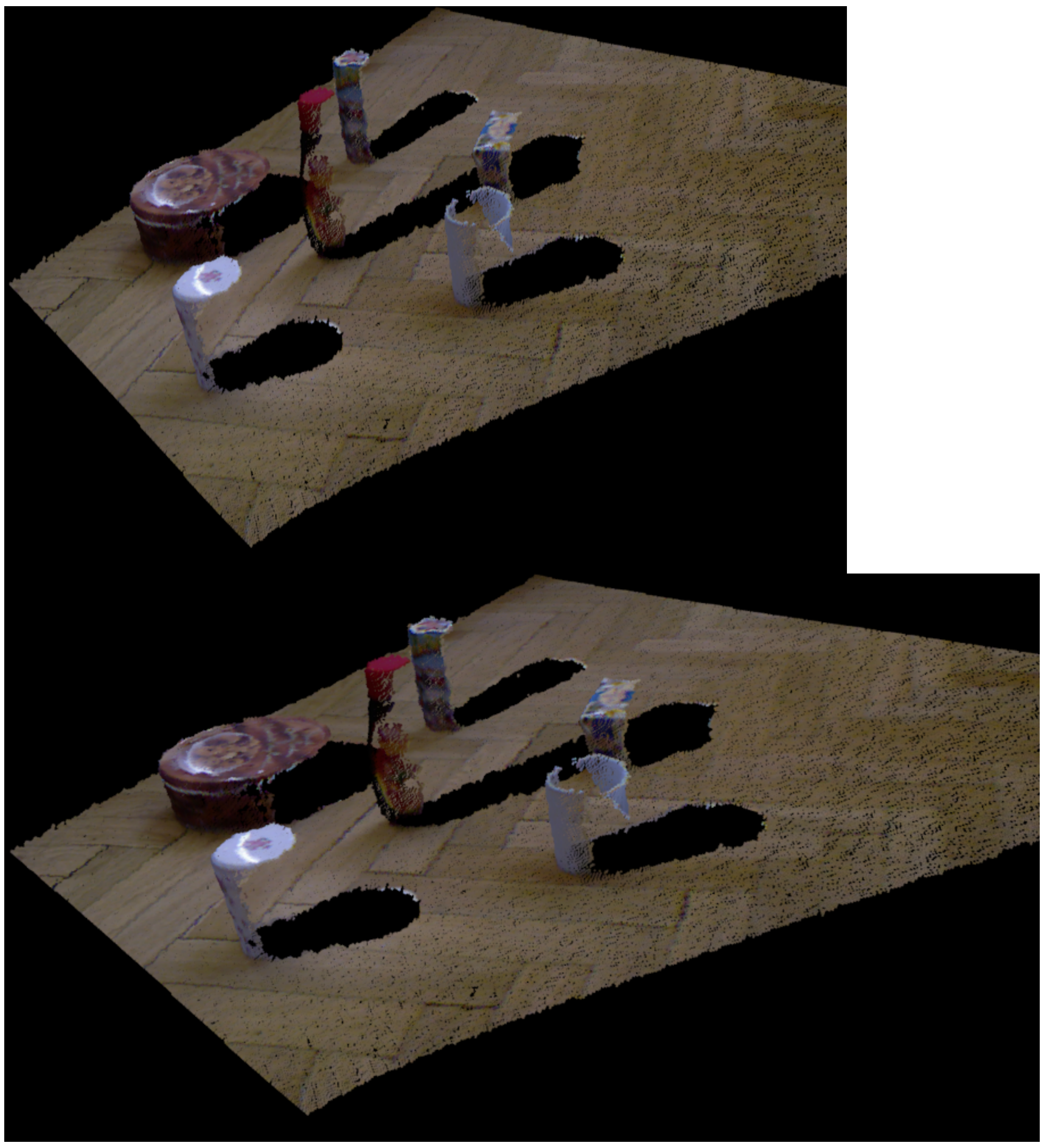

Figure 1: Example pointcloud from the dataset shown from two different viewpoints. Note the difficulties capturing the cup as one object.

Part 1: Implementation (10 points)

In this exercise, the RANSAC algorithm should be fully implemented as a MATLAB function.

1. Basic RANSAC (6 points)

Input parameters

- `p`: The pointcloud (Nx6 array, N = number of points), where each column is `[x y z]`. RGB values are in the range [0,1].
- `params.confidence`: Solution Confidence (in percent): Likelihood of all sampled points being inliers.
- `params.inlier_margin`: Max. distance of a point from the plane to be considered an inlier (in meters)
- `params.min_sample_dist`: Minimum distance of all sampled points to each other (in meters). For robustness.

The input parameters for the RANSAC function have to be altered in the main script main.m. The default values won't produce satisfying results!

Output parameters:

- `a, b, c, d`: Coefficients of the calculated plane, defined by the equation `a*x + b*y + c*z + d = 0`.
- `inliers`: Binary (1xN) array, containing a 1 for every inlier and a 0 for every outlier in p at the corresponding position. E.g. inliers = `[0, 1, 0, 0, 1, 0, 1, 1, ...]` means, the points p with column indices 2, 5, 7, 8,... are inliers and therefore part of the calculated plane.
- `k`: The number of necessary iterations to achieve the required solution confidence `confidence`.

General RANSAC sequence:

- See lecture slides
- Dynamic adaptation of the number of iterations depending on the number of found inliers
- After finding the largest set of inliers, the plane fit should be refined once more considering all inliers.

Refining the fit

After RANSAC has divided the set of points in inliers and outliers, the plane fit can be further improved by finally considering ALL inlier points to calculate the plane parameters a, b, c and d (`a*x + b*y + c*z + d = 0`). For that, an overdetermined system of equations has to be solved, which can be done in various ways, such as a least squares approximation, as follows.

$$\begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Parameter d is free to choose, such that for e.g. d=-1 we get the following simplified system of equations:

$$\begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

This system of equations is overdetermined (N_I equations for 3 unknowns, N_I = number of inliers). To get the least-squares solution of this system, MATLAB offers the backslash-operator `\\`:

System of equations: `A\*x = b`

Least-Squares-Solution: `x = A \\ b`

A plane can be defined such that points on one side of it are in a "positive" space, and others in a "negative" space, depending on the direction of the normal. For our purposes this does not matter. Ensure that all the normals have a positive z value, and make sure the rest of the vector components are kept consistent with this.

You should implement the cost function in `ransac_error` and call it from `fit_plane`. This allows the cost function to be changed easily using the input parameters, as will be necessary in part 3.

Complete **fit_plane.m** and **ransac_error.m**.

2. Multi-plane extraction (1 points)

Sometimes the pointcloud you are dealing with may have more than one plane present. Implement a function which will extract multiple planes from a single cloud, stopping when the number of points in the extracted plane is smaller than some proportion of the original cloud size.

Complete **filter_planes.m**.

3. MSAC and MLESAC (3 points)

Use section 5 of the [paper on MLESAC by Torr and Zisserman] to implement MSAC and MLESAC. Equation 10 shows the full negative log likelihood function. Note that there appears to be a misplaced bracket in the equation, the correct formulation should be:

$$-L = -\sum log(\gamma \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{e^2}{2\sigma^2}) + (1-\gamma)\frac{1}{v})$$

You may ignore the squaring of parts of the error term, and are not required to use 1.96 sigma as the threshold. The value of `v` should be set to the length of the maximum diagonal of the cloud. You should use `normpdf`. For MLESAC, assume that an inlier is a point which falls within one standard deviation of the normal distribution with the sigma value (i.e. the threshold) you passed in.

Complete **msac_error.m** and **mlesac_error.m**.

Part 2: Documentation (6 points)

- Answer each numbered question separately, with the answer labelled according to the experiment and question number. For example, experiment 1 question 2 should be numbered 1.2. You do not need to include the question text.
- Include images to support your answers. These should be referenced in the answers. The document should include information about what parameter settings were used to generate images so that they can be reproduced.
- Your answers should attempt an explanation of why an effect occurs, not just describe what you see in the supporting images.

- Be precise. "It is better" or similarly vague statements without any additional information are not sufficient.

1. Good result (screenshot) with pointcloud number corresponding to the last digit of your matriculation number (e.g. matriculation number 123456 -> pointcloud 6). (0.5 points)
2. For each of the 3 RANSAC parameters a short discussion of how the parameter influences the result (with supporting screenshots). (2 points)
3. Apply multi-plane extraction to the desk, door and kitchen pointclouds. Describe the results and explain issues that you observe. Discuss briefly ways you might solve the issues you describe. (1.5 points)
4. In `sac_comparison`, compare RANSAC, MSAC and MLESAC. Most of the code has been provided for you; you should choose at least 2 statistics to compute on the results (means and variances are fine), to give some idea of how the different methods perform. **Explain what each thing you are measuring is, and why you chose to measure the things you did**. Modify `repetition_vals` to cover a larger number of plane computations. Also look at different values of `gauss_prop` (e.g. a high value and a low value). Present a few graphs, and briefly describe interesting or surprising things that you see (if any). **You are not expected to be rigorous**. (2 points)

You must specify the parameters used in all of your screenshots.

Maximum 1500 words, arbitrary number of pages for result images (referenced in the text). Violating the text limit will lead to point deductions!

Assistance

Please keep to the following hierarchy whenever there are questions or problems:

1. Forum: Use the TUWEL forum to discuss your problems.
2. Email for questions or in-person help requests: machinevision@acin.tuwien.ac.at

Upload

Please upload the Matlab files `filter_planes.m`,`fit_plane.m`, `ransac_error.m`, `msac_error.m`, `mlesac_error.m`, and `sac_comparison.m`, as well as your documentation (pdf-format) **as one ZIP-file** via TUWEL.

Submission status

Submission status	No attempt
Grading status	Not marked
Due date	Monday, 3 December 2018, 12:00 PM
Time remaining	2 days 22 hours
Last modified	-
Submission comments	Comments (0)

Add submission

You have not made a submission yet

◄ MLESAC: A New Robust Estimator with A…

Jump to...

Files for Exercise 4: RANSAC ►

© Technische Universität Wien - Teaching Support Center - support@tuwel.tuwien.ac.at

AMC - Academic Moodle Cooperation

My courses
183.585 Computer Vision (VU 3,0) 2018W

[186.112 Heuristic Optimization Techniques (VU 2,0) 2018W]
[376.054 Machine Vision and Cognitive Robotics (VU 4,0) 2018W]
[All my courses]
[Course overview]
[Search courses]
[New course]
Help
[TUWEL Tutorials (Students)]
[TUWEL Tutorials (Teachers)]
[TUWEL scenarios]
[E-Mail Ticket Support]
[General Information]
[Terms of Use]
[Teaching Support Center]