Computer Vision Project
Winter 2024
Thomas Gorges

Individual exercise sheet 1, starting from October 14th, 2024, due February 17th, 2025, 02:00 p.m.

The following additional exercises allow you to extend the standard Project CV content (lectures and group exercises) from a 5 ECTS to a 10 ECTS option. In order to be eligible for 10 ECTS, you must complete all group and individual exercises. Compared to the group exercise, these additional ones must be worked on individually.

Similar to the group exercises, it is not sufficient to use existing implementations provided in common Computer Vision libraries (if they exist). Instead, you are expected to implement your own methods and strategies described below. Please also look at the related literature for further hints on how the respective methods work and some implementation hints.

Individual Exercise 1.1: Maximum Likelihood Estimation Sample Consensus (MLESAC)

In its default variant, RANSAC (random sample consensus) involves the calculation of several hypotheses/models evaluated with respect to the number of inliers (number of samples in the consensus set and thus are supported by the hypothesis). Ultimately, we select the hypothesis with the highest number of inliers, which corresponds to the following minimization problem:

$$\mathcal{C} = \sum_{i}^{N} p(d(s_i)), \quad p = \begin{cases} 0, & \text{if}\, d(s_i) < \varepsilon \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

, where p is the decision criterion, d marks some distance metric (e.g. Euclidean distance), s_i corresponds to the i-th data point, N denotes the maximum number of evaluated datapoints, and ε is the accepted error threshold.

As you might already observe from this expression, choosing an appropriate error threshold ε can be difficult. If ε is too large, most data points are considered inliers, leading to many hypotheses/models with low $\mathcal{C}$. In the extreme case, all hypotheses consider all data points as inliers ($\mathcal{C} = 0$).

In this exercise, we'll look at a generalizing variant of RANSAC called **MLESAC** (Maximum Likelihood Estimation Sample Consensus) [3], which partly deals with the aforementioned problem. MLESAC still adopts the same random sampling approach but is adjusted to maximize the likelihood of the solution instead of maximizing the number of inliers [2].

This strategy can be expressed as follows:

$$\mathcal{C} = \sum_{i}^{N} p(d(s_i)), \quad p = \begin{cases} d(s_i), & \text{if}\, d(s_i) < \varepsilon \\ \gamma, & \text{otherwise} \end{cases} \tag{2}$$

, where γ is a constant error larger than the error threshold ($\gamma > \varepsilon$). Please note that inlier points are not scored with zero cost but rather enter the formula with their corresponding distance metric error. Data points with distance errors equal to or beyond the error threshold $d_i \geq \varepsilon$ contribute to the overall cost $\mathcal{C}$ with γ.

Please extend your RANSAC implementation using the MLESAC strategy. You can add the option to your existing implementation or create a new variant. Try out different values for ε and verify whether

MLESAC helps to be less susceptible to those hyper-parameter choices. Please discuss any advantages and disadvantages of this approach.

Individual Exercise 1.2: Preemptive RANSAC

As you might have already experienced hands-on, RANSAC involves evaluating a substantial number of different model hypotheses until a certain confidence in the solution can be obtained. The more complex the data (e.g., having very low counts in the desired inlier set), the longer the required iterations until convergence to high confidence is reached. This property of RANSAC can be problematic in real-time and thus time-constrained scenarios where your application is expected to operate at a certain wall-clock time.

One option to deal with this issue is **Preemptive RANSAC** proposed by Nister [1]. This approach's general idea is to calculate a fixed number of hypotheses/models and then evaluate them in parallel on a subset of the data points. After evaluation, the hypotheses are ordered according to their respective cost $\mathcal{C}$. Only the top-scoring ones are retained and evaluated on the next subset of the data points. Thereby, the selection procedure is given by a preemption function $f(i), i = 1, ..., N$ that tells you the number of top-scoring hypotheses used for evaluating the i-th data point. The authors propose the following function:

$$f(i) = \lfloor M 2^{-\lfloor \frac{i}{B} \rfloor)} \rfloor \tag{3}$$

, where M is the initially defined number of hypotheses, and B is the number of data points a hypothesis is evaluated against before the preemption and reordering step occurs. $\lfloor \cdot \rfloor$ marks the flooring operator. Using the preemption function, you repeat the ordering and scoring steps until a single hypothesis is left or until all data points have been used [2].

Please implement the Preemptive RANSAC strategy and evaluate it on the exercise data. For that, please consider different choices for M and B and discuss how sensitive the model behaves across different parameterizations. Feel free to combine this approach with the MSAC variant. In addition to different parameterizations, please visualize the estimated plane model (either the top-box or floor plane) for three choices of M to showcase the effect of different time budgets. Please discuss any advantages and disadvantages of this approach.

Literatur

[1] D. Nister. Preemptive RANSAC for live structure and motion estimation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 199–206, 2003.

[2] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 500–513. Springer Berlin Heidelberg, 2008.

[3] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.