Lab 3 - Theoretical questions
Chalmers University of Technology
Spring 2024
Fanze Meng

Lab 3 - Theoretical questions

Fanze Meng - fanze@chalmers.se

1 Question1

1.1

To derive equations for the expected run-time of RANSAC for each solver, we need to consider several factors:

1. The number of iterations RANSAC will require to find a solution.
2. The time taken to evaluate a model on a single data point.
3. The time taken for the solver to run once.

which can be denoted:

- N as the total number of data points.
- m as the number of data points required by the solver.
- ϵ as the inlier ratio.
- η as the probability of not picking an all-inlier sample.

First, deriving the expected run-time for Solver 1:

1. Number of iterations required by RANSAC for Solver 1 (m = 2):

$$k_1 = \lceil \frac{\log(\eta)}{\log(1-\epsilon^m)} \rceil \tag{1}$$

2. Time taken to evaluate a model on a single data point: 0.005 time units.
3. Time taken for Solver 1 to run once: 1 time unit.

So, the expected run-time of RANSAC for Solver 1 can be calculated as:

$$T_1 = k_1 \times [(m_1 \times 0.005) + 3]$$

Next, deriving the expected run-time for Solver 2:

1. Number of iterations required by RANSAC for Solver 2 (m = 3):

$$k_2 = \lceil \frac{\log(\eta)}{\log(1-\epsilon^m)} \rceil \tag{2}$$

2. Time taken to evaluate a model on a single data point: 0.005 time units.

3. Time taken for Solver 2 to run once: 1 time unit.

So, the expected run-time of RANSAC for Solver 2 can be calculated as:

$$T_2 = k_2 \times [(m_2 \times 0.005) + 1] \times (1 - \eta)$$

In both equations, k_1 and k_2 represent the number of iterations required by RANSAC for Solver 1 and Solver 2, respectively.
These equations give us the expected run-time of RANSAC for each solver, taking into account the number of iterations, the time taken to evaluate a model, and the time taken for the solver to run once.

1.2

To determine which solver to prefer for different inlier ratios (ϵ) with given values of η and N, we need to calculate the expected run-time (T_1 and T_2) for each solver for each value of ϵ.
Given:

- $\eta = 0.01$
- $N = 100$

We use the equations derived earlier:
For Solver 1:

$$T_1 = k_1 \times [(m_1 \times 0.005) + 3] \tag{3}$$

For Solver 2:

$$T_2 = k_2 \times [(m_2 \times 0.005) + 1] \times (1 - \eta) \tag{4}$$

We calculate k_1 and k_2 using the formulas of equations (1) and (2), then substitute the values of k_1 and k_2 into the above equations for each value of ϵ.
Then, we compare the expected run-times for both solvers for each value of ϵ to determine which solver is preferable.
Given ϵ values of 0.1, 0.3, 0.5, 0.7, and 0.9, we calculate k_1 and k_2 using the provided formulas and then substitute them into the equations for T_1 and T_2.
We can get the result from the following two images:

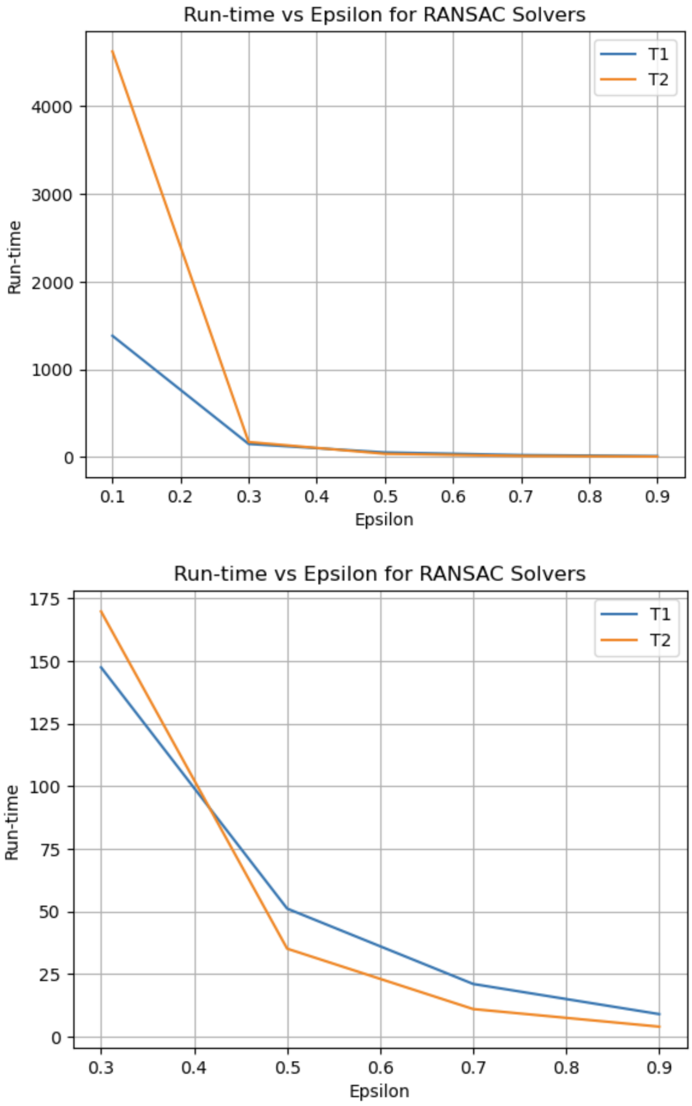

Figure 1: The run time with different ϵ values in the given condition

It's clear that we should choose solver 1 when $\epsilon = 0.1$ and 0.3. However, the run time of solver 2 becomes quicker as the increase of the ϵ, due to a different structure of solver 2, saving the time at big ϵ.

1.3

The equation

$$k_{\max} = \frac{\log \eta}{\log(1 - \epsilon^m)}$$

allows us to estimate the maximum number of iterations $k_{\max}$ required by RANSAC to achieve a desired probability η of missing an all-inlier sample, given an estimate ϵ of the inlier ratio and the number of data points m required by the minimal solver.

This equation assumes that it is sufficient to select one all-inlier sample to obtain a model that fits all inliers. However, this assumption may not always be fully valid in practice. In some cases, especially when dealing with noisy or complex datasets, selecting just one all-inlier sample may not be enough to obtain a model that fits all inliers accurately.

When this assumption is not fully valid, it can impact the number of iterations that RANSAC actually needs. If selecting just one all-inlier sample is not sufficient to obtain a robust model, RANSAC may require more iterations to achieve the desired level of confidence in the estimated model. This means that the estimated maximum number of iterations $k_{\max}$ provided by the equation may be underestimated, and RANSAC may need to run for more iterations to ensure a reliable estimation of the model parameters.

In such cases, practitioners may need to adjust the parameters of RANSAC, such as the termination threshold or the number of random samples, to account for the limitations of the assumption and improve the robustness of the estimation process. Additionally, using alternative techniques or modifications of RANSAC, such as PROSAC or MLESAC, which are designed to handle outliers and noise more effectively, may also help address the limitations of the assumption and improve the accuracy of the estimated model parameters.

2 Question2

2.1

To estimate the transformation between two sets of 3D points, we need to derive a linear system $M\theta = v$, where θ contains the parameters of the transformation. The transformation parameters in this case include a, b, tx, and ty.
Given a set of correspondences between the two sets of 3D points, we can represent each correspondence as (x, y, z) in the original set and (x', y', z') in the transformed set.
For each correspondence, the transformation equation can be written as:

$$\begin{cases} x' = ax - bz + tx \\ y' = y + ty \\ z' = bx + az \end{cases}$$

For n correspondences, we can rewrite the system of equations in matrix form:

$$\begin{bmatrix} x_1 & -z_1 & 1 & 0 \\ y_1 & 0 & 0 & 1 \\ x_2 & -z_2 & 1 & 0 \\ y_2 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -z_n & 1 & 0 \\ y_n & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ tx \\ ty \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix}$$

Let $M\theta = v$ denote this matrix equation, where $\theta = [a, b, tx, ty]^\top$ contains the parameters of the transformation.
The minimal number of correspondences needed by the solver to compute a transformation is determined by the number of unknown parameters in θ. In this case, there are 4 unknown parameters (a, b, tx, ty), so at least 4 correspondences are needed to form a system of linear equations that can be solved for the transformation parameters.

2.2

To significantly accelerate RANSAC using the additional equations provided by the minimal solver, we can follow the following strategy:

1. **Initial Transformation Estimation:**
   - Use a minimal set of correspondences (at least 4) to compute an initial estimate of the transformation parameters (a, b, tx, ty) using the minimal solver.
2. **Compute Residuals:**
   - For all correspondences, compute the residuals between the observed transformed points (x', y') and the points predicted by the estimated transformation $(\hat{x'}, \hat{y'})$.
3. **Thresholding:**
   - Use a predetermined threshold to classify correspondences as inliers or outliers based on their residuals. Correspondences with residuals below the threshold are considered inliers, and those above the threshold are considered outliers.
4. **Refinement:**
   - Refine the estimated transformation using a robust optimization technique (e.g., least squares) by only considering the inlier correspondences identified in the previous step.
5. **Iterative Refinement:**
   - Repeat steps 2-4 for a fixed number of iterations or until convergence to refine the estimated transformation further.

By utilizing this strategy, we can exploit the computational efficiency of the minimal solver while ensuring the accuracy of the estimated transformation by considering all available correspondences during refinement.

2.3

To determine which of the lines were not generated by a minimal solver, we need to analyze the properties of the residuals generated by each line.
Given the vectors of residuals r_1, r_2, r_3, and r_4, each corresponding to a line, we can analyze the residuals to determine if they were likely generated by a minimal solver.
If a line was generated by a minimal solver, we would expect:

1. The residuals to be small, indicating that the line fits the data well.
2. A consistent pattern in the residuals, indicating that the line captures the underlying structure of the data.

Analyze each vector of residuals to determine if they exhibit these properties:

1. $r_1 = (0.55, 0.18, 0.48, 0.45, 0.38, 0.77, 0.80, 0.00, 0.70, 0.00)$

- The residuals vary in magnitude, and there doesn't seem to be a clear pattern. It's possible that these residuals were not generated by a minimal solver.

2. $r_2 = (0.71, 0.05, 0.09, 0.82, 0.69, 0.03, 0.28, 0.32, 0.95, 0.03)$

   - Similar to r_1, the residuals vary in magnitude, and there doesn't seem to be a clear pattern. It's also possible that these residuals were not generated by a minimal solver.

3. $r_3 = (0.28, 0.34, 0.67, 0.66, 0.16, 0.11, 0.49, 0.94, 0.00, 0.22)$

   - The residuals are relatively small, and there is a consistent pattern where most residuals are below 0.7. This suggests that the line represented by r_3 fits the data well and may have been generated by a minimal solver.

4. $r_4 = (0.16, 0.97, 0.00, 0.49, 0.00, 0.14, 0.42, 0.92, 0.79, 0.96)$

   - The residuals vary in magnitude, and there doesn't seem to be a clear pattern. It's possible that these residuals were not generated by a minimal solver.

If we consider the sum of residuals for each line:
Sum of residuals for $r_1 = 4.31$
Sum of residuals for $r_2 = 3.97$
Sum of residuals for $r_3 = 3.87$
Sum of residuals for $r_4 = 4.85$
Based on this analysis, it appears that the line represented by r_3 is more likely to have been generated by a minimal solver, as its residuals are relatively small and exhibit a consistent pattern. The other lines represented by r_1, r_2, and r_4 do not exhibit clear patterns in their residuals and may not have been generated by a minimal solver.