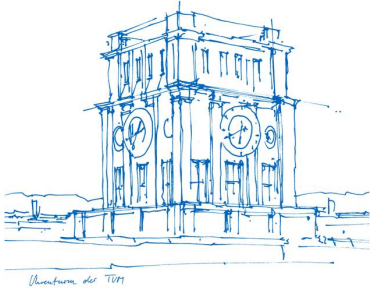
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 13 Robust Estimation

Dr. Haoang Li

12 July 2023 12:00-13:30



Announcement Before Class

➢ Updated Lecture Schedule

For updates, slides, and additional materials:
https://cvg.cit.tum.de/teaching/ss2023/cv2

90-minute course; 45-minute course

Wed 19.04.2023 Chapter 00: Introduction
Thu 20.04.2023 Chapter 01: Mathematical Backgrounds

Wed 26.04.2023 Chapter 02: Motion and Scene Representation (Part 1)
Thu 27.04.2023 Chapter 02: Motion and Scene Representation (Part 2)

Wed 03.05.2023 Chapter 03: Image Formation (Part 1)
Thu 04.05.2023 Chapter 03: Image Formation (Part 2)

Foundation

Wed 10.05.2023 Chapter 04: Camera Calibration
Thu 11.05.2023 Chapter 05: Correspondence Estimation (Part 1)

Wed 17.05.2023 Chapter 05: Correspondence Estimation (Part 2)
Thu 18.05.2023 No lecture (Public Holiday)

Wed 24.05.2023 No lecture (Conference)
Thu 25.05.2023 No lecture (Conference)

} Videos and reading materials about the combination of deep learning and multi-view geometry

Wed 31.05.2023 Chapter 05: Correspondence Estimation (Part 3)
Thu 01.06.2023 Chapter 06: 2D-2D Geometry (Part 1)

Wed 07.06.2023 Chapter 06: 2D-2D Geometry (Part 2)
Thu 08.06.2023 No lecture (Public Holiday)

Wed 14.06.2023 Chapter 06: 2D-2D Geometry (Part 3)
Thu 15.06.2023 Chapter 06: 2D-2D Geometry (Part 4)

Core part

Wed 21.06.2023 Chapter 07: 3D-2D Geometry
Thu 22.06.2023 Chapter 08: 3D-3D Geometry

Wed 28.06.2023 Chapter 09: Single-view Geometry
Thu 29.06.2023 Chapter 10: Combination of Different Configurations

Wed 05.07.2023 Chapter 11: Photometric Error and Direct Method
Thu 06.07.2023 Chapter 12: Bundle Adjustment

Wed 12.07.2023 Chapter 13: Robust Estimation
Thu 13.07.2023 Exam Information and Knowledge Review

Advanced topics and high-level tasks

Wed 19.07.2023 Chapter 14: SLAM and SFM
Thu 20.07.2023 No Onsite Lecture. Alternative: Online Meeting for Question Answering

Today's Outline

- Problem Formulation
- Expectation-maximization
- RANSAC
- Robust Kernel (M-estimator)

Problem Formulation

- Definition of Outliers

✓ Recall that correspondences are obtained based on descriptor similarity or other strategies.
✓ In practice, correspondences are usually contaminated by outliers (i.e., mis-matches).

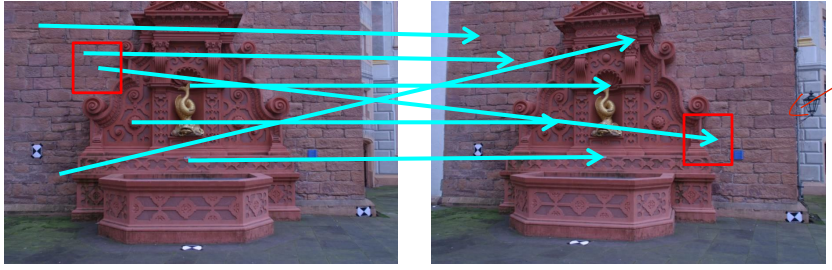


Image 1 Image 2

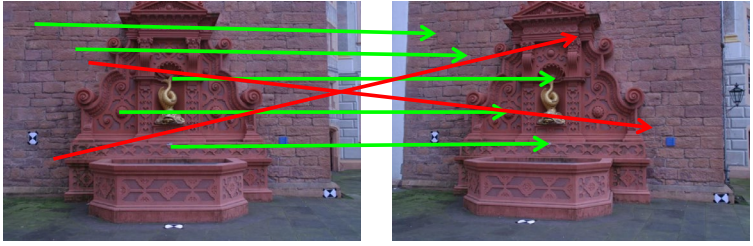
Problem Formulation

- 重复模式（即具有相同外观的特征）
- 几何和光度变化
- 图像噪音大
- 遮挡
- 移动物体
- 图像或运动模糊

➢ Definition of Outliers

✓ Reasons for outliers

- Repetitive patterns (i.e., features with the same appearance)
- Geometric and photometric changes
- Large image noise
- Occlusions
- Moving objects
- Image or motion blur

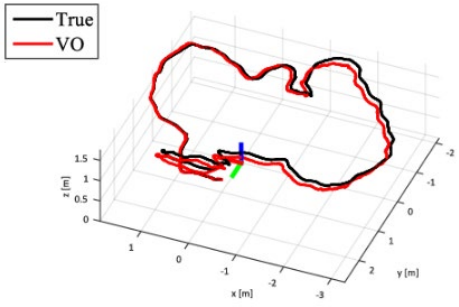


Problem Formulation

- Effects of Outliers for Visual Odometry

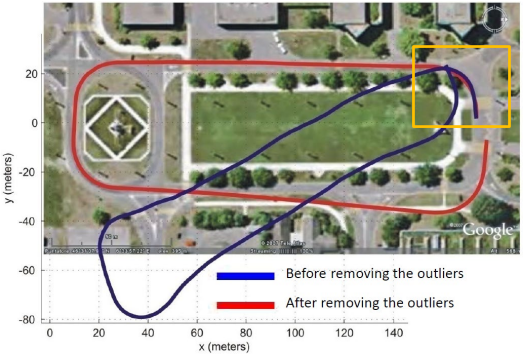
accept noise ⟷ remove outliers

Inaccurate trajectory caused by noise:



This case was introduced before

Wrong trajectory caused by outliers:



Problem Formulation

➢ Our Goal

✓ Robust Estimation is essential for reliable and accurate localization.
✓ We aim to remove outliers and also estimate the model parameters, e.g., camera pose.

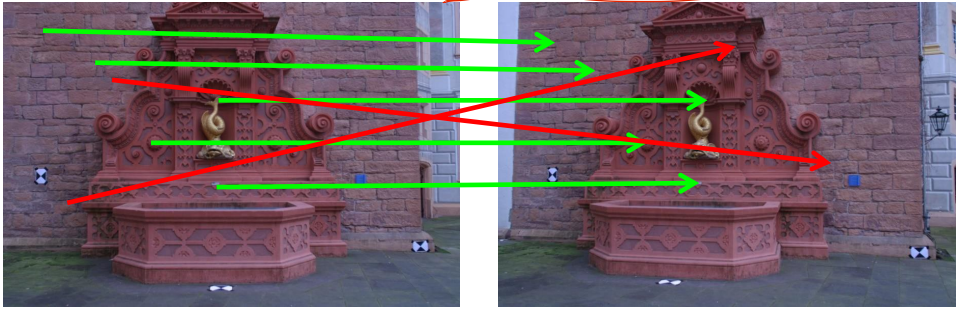


Image 1 Image 2

Problem Formulation

➢ Dominant Methods for Outlier Removal

✓ Expectation-maximization
✓ RANSAC
✓ Robust kernel (M-estimator)
…

remove outliers
estimate pose

✓ They can be applied to various geometric problems whose goal is to estimate the parameters of a model from the data, e.g.,

- Essential matrix estimation
- Fundamental matrix estimation
- PnP
- Homography estimation

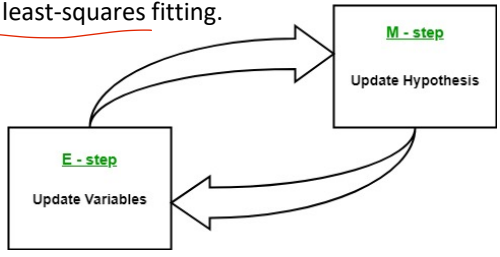
Expectation Maximization (EM) algorithm

➢ Overview

✓ EM is a general method for model fitting in the presence of outliers.

✓ At each iteration, it performs E-step and M-step.

- E-step is to assign each data a weight/outlier ratio ⇒ inlier = 1, outlier = 0
- M-step is to compute model parameters based on a weighted least-squares fitting.



✓ Let's take straight line fitting problem for example.

Expectation Maximization (EM) algorithm

- Pipeline and Example

✓ Problem definition

Given a set of point perturbed by outliers, we aim to simultaneously **remove outliers** and **estimate 2D line** expressed by (slope, intercept) or homogenous coordinates (a, b, c).

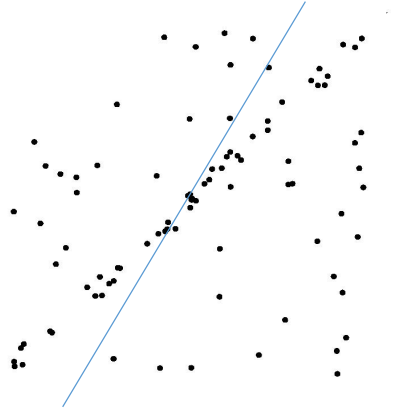
Expectation Maximization (EM) algorithm

➢ Pipeline and Example

✓ Step 1: Initialization based on least-squares fitting

treat all data equally

- Estimate line parameters that fit all data points, using least squares: $\min \sum r_i^2$, where $\boldsymbol{r}_i$ is the point-to-line distance.
- We do not have prior information about which points are inliers or outliers, so we have to treat them equally.
- The result is obviously affected by outliers.



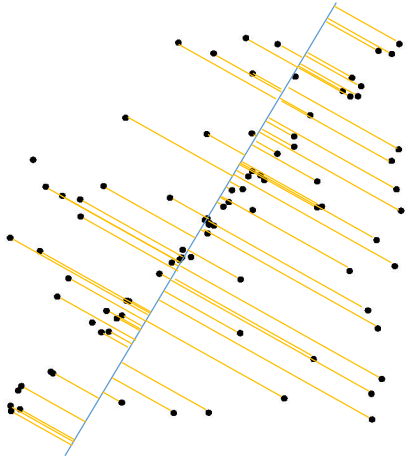
Expectation Maximization (EM) algorithm

➢ Pipeline and Example

✓ Step 2: E-step of an iteration: weight assignment

- Based on the estimated model (line), calculate residual error r_i for each data point.
- Use residual error to assign each data a weight. Weight should be inversely proportional to the error, e.g., $w_i = e^{-ri}$
- This weight from zero to 1 is the probability that a data is an inlier.

r large
error large
weight small

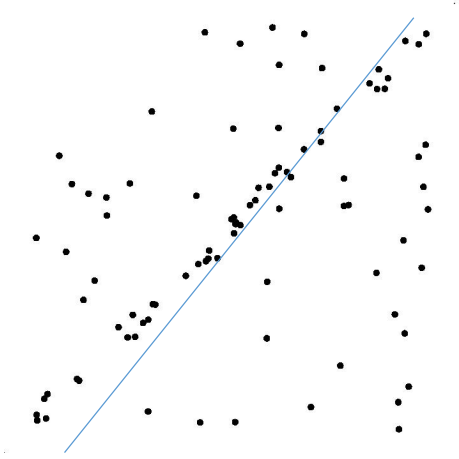


Expectation Maximization (EM) algorithm

➢ Pipeline and Example

✓ Step 3: M-step of an iteration: re-estimation of model parameters.

- Formulate a weighted least-squares loss: $\min \sum w_i r_i^2$
- The weighted least-squares solution is the updated parameter.
- Intuitively, it should be better than the initial least-squares solution.

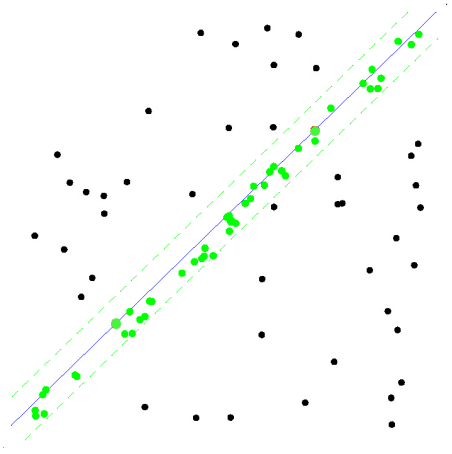


Expectation Maximization (EM) algorithm

➢ Pipeline and Example

✓ Step 4: Alternately perform step 2 and step 3 till convergence.
Accordingly, the optimal parameters are obtained.

✓ Step 5: Treat the data whose weights are higher than a threshold (e.g., 0.8) as inliers.



Expectation Maximization (EM) algorithm

➢ Limitation of EM algorithm

✓ It is highly sensitive to the initialization

Recall that EM selects the initial solution by minimizing the sum of squared residuals $\sum r_i^2$.

Initialization is strongly affected by a few large error values (i.e., outliers).

Thus, EM converges to the wrong solution if the initial solution is far from the ground-truth one.

RANSAC

➢ Overview

- ✓ Random sample consensus (RANSAC) is one of the most popular methods for model fitting in the presence of outliers.
- ✓ Main advantage: It is not sensitive to the initial condition.
- ✓ Another important property: It is non-deterministic. You get a different result every time you run it
- ✓ Similar to EM algorithm, RANSAC can be applied to all sorts of geometric problems where the goal is to estimate the parameters of a model from the data.

RANSAC

➢ Pipeline and Examples

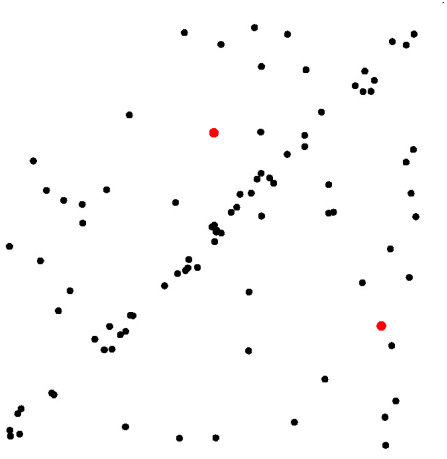
- ✓ Let's first introduce RANSAC for line fitting and then see how we can use it to do robust camera pose estimation.
- ✓ Given a set of points perturbed by outliers, we aim to simultaneously **remove outliers** and **estimate 2D line's parameters** (slope, intercept) or homogenous coordinates (a, b, c).

RANSAC

➢ Pipeline and Examples

✓ Step 1: Sample "minimal-case" data at random

- RANSAC typically considers the minimal case when conducting sampling.
- Two sampled points are the minimal case of 2D line fitting.
- It samples 5 point correspondences for relative pose estimation based on 5-point method.
- It samples 4 point correspondences for homography estimation.

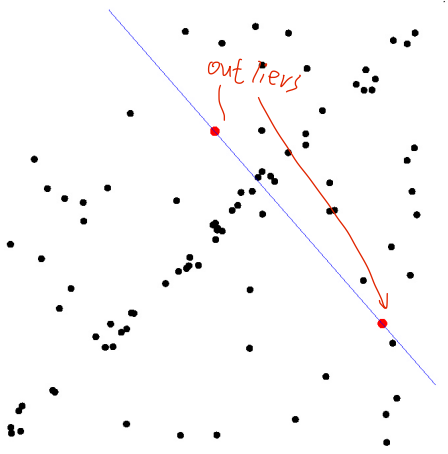


RANSAC

➢ Pipeline and Examples

✓ Step 2: Calculate model parameters using the sampled data (this step is also called “hypothesizing a model”)

- Given two points, we can easily compute a 2D line.
- For camera pose estimation, we can use five sampled point correspondences to estimate the camera pose based on five-point method.
- Note: We do not care about the correctness of the sampled data at this step. Regardless of a valid or invalid sampling, we directly use the sampled data to compute the model parameters.
  Mathematically, we just generate a system and solve this system.

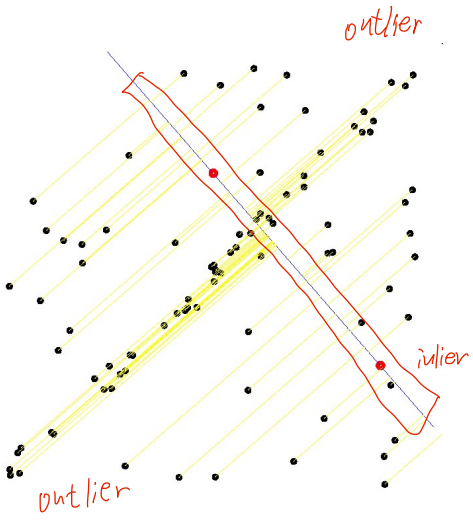


RANSAC

➢ Pipeline and Examples

✓ Step 3: For each data, calculate the residual error w.r.t. the computed/hypothesized model

- The sampled data must satisfy the fitted model, but the other data may not.
- For line fitting, the residual error is the point-to-line distance.
- For camera pose estimation, the residual error may be point-to-epipolar line distance.

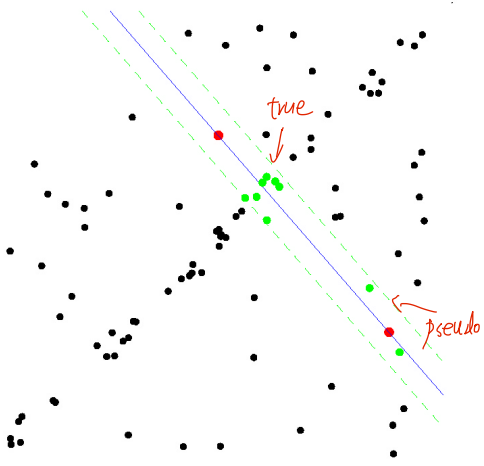


RANSAC

➢ Pipeline and Examples

✓ Step 4: Select data that support current hypothesis (this step is also called “inlier set generation”)

- If a data can be roughly fitted by the model estimated by the sampled data, we call it an “inlier” w.r.t this model.
- Note: an inlier may be a true inlier if the model is correct; it may be a pseudo inlier if the model is wrong.
- For each model, we can associate it with an inlier set.



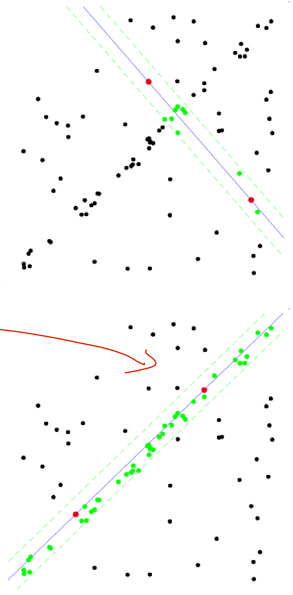
RANSAC

➢ Pipeline and Examples

✓ Step 4: Select data that support current hypothesis (this step is also called “inlier set generation”)

- Intuitively, the cardinalities of inlier sets associated with different models are different.
- If a model is computed by the sampled **true inliers**, its associated inlier set should lead to a high cardinality.

高卡数。

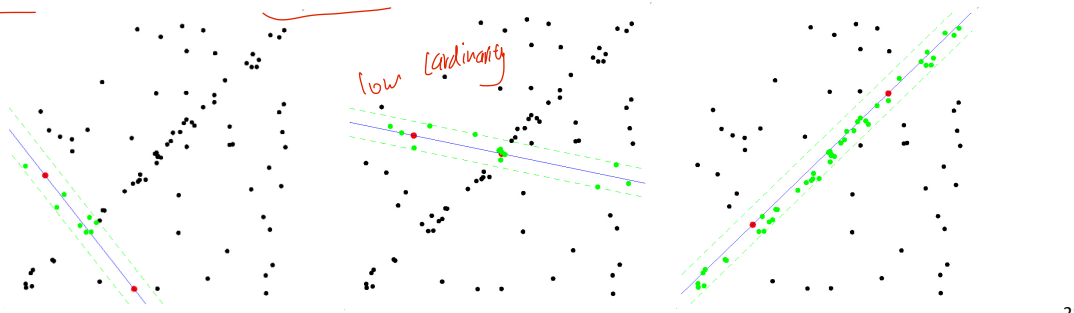


RANSAC

➢ Pipeline and Examples

✓ Step 5: Repeat from step 1 to step 4 k times, i.e., conduct k iterations

- We hypothesized/estimated k models after k samplings.
- k models are associated with k inlier sets.

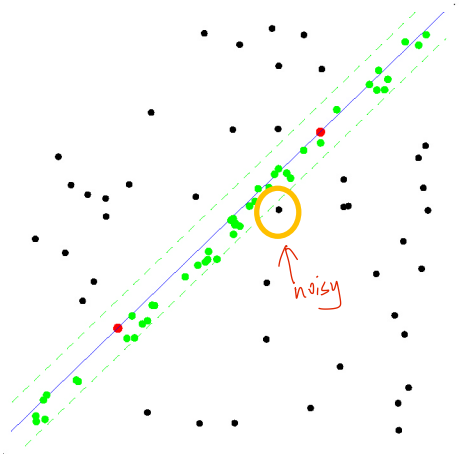


RANSAC

➢ Pipeline and Examples

✓ Step 6: Inlier set maximization

- Select the inlier set with the highest cardinality.
- The model associated with this inlier set (see blue line) should be estimated by two sampled true inliers (see red points). Most of other true inliers (see green points) can be fitted by this model.
- There is still a room for accuracy improvement due to the noise of sampled true inliers.

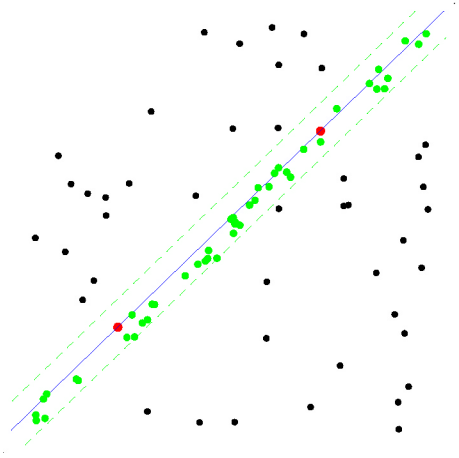


RANSAC

➢ Pipeline and Examples

✓ Step 7: Calculate the final model parameters using the optimal inlier set

- Use all the true inliers of the optimal inlier set to compute the least-squares solution.
- The result obtained at step 6 is already good.
- This step can further improve accuracy by noise compensation.



RANSAC

- Discussion about Minimal and Quasi-minimal Cases

- ✓ For relative pose estimation with outliers, RANSAC typically samples 5 points and estimates the essential matrix based on five-point method at each iteration. This is a typical minimal case.

- ✓ In practice, we can also let RANSAC sample 8 points and estimate the essential matrix based on eight-point method. This is a quasi-minimal case.

- ✓ The minimal case is preferred. Reason: In RANSAC, we should guarantee at least one valid sampling (i.e., we should sample a set of pure inliers at a certain iteration). Simultaneously sampling 5 true inliers is easier than sampling 8 true inliers.

RANSAC

➢ Discussion about Minimal and Non-minimal Cases

✓ Recap on DLT for 3D-2D geometry (perspective n points)

$$\mathbf{t}_1^T\mathbf{P} - \mathbf{t}_3^T\mathbf{P}u_1 = 0,$$

$$\mathbf{t}_2^T\mathbf{P} - \mathbf{t}_3^T\mathbf{P}v_1 = 0.$$

Constraint of one correspondence

$$\begin{pmatrix} \mathbf{P}_1^T & 0 & -u_1\mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N\mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N\mathbf{P}_N^T \end{pmatrix} \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix} = 0$$

Parameters of transformation

✓ 6 point correspondences lead to a well-determined linear system.

✓ > 6 point correspondences lead to an over-determined linear system. We can obtain a least-squares solution based on SVD. vs. noise

RANSAC

➢ Discussion about Minimal and Non-minimal Cases

How to choose the minimal case or redundant case?

✓ Minimal case

- If we cannot guarantee that all the correspondences are inliers, we consider the integration of **minimal case** and **RANSAC**.
- RANSAC samples a minimal number of points (e.g., 6 for DLT) ***N*** times. We use each set of points to generate a well-determined system. Accordingly, we have ***N*** well-determined systems.
- Based on inlier set maximization, we select the best system from ***N*** candidate systems.

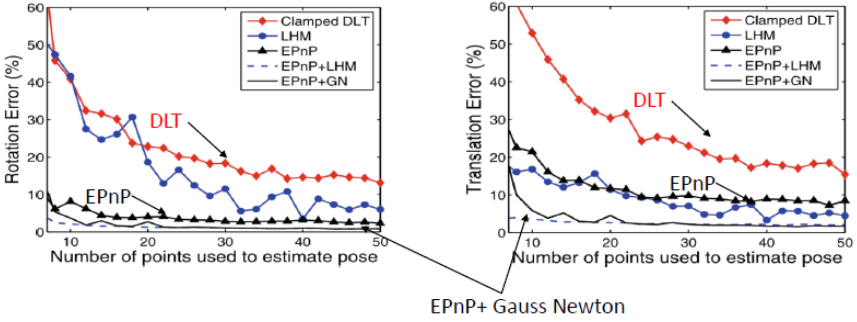
✓ Redundant Case

- If we have prior knowledge that all the correspondences are inliers, we can use all the correspondences to generate an **over-determined** linear system. The result is the least-squares solution (helpful for noise compensation). Computing least-squares solution is the final step of RANSAC.

RANSAC

- Discussion about Minimal and Non-minimal Cases

✓ The more inlier points we use, the higher the algorithm accuracy is.
✓ The following is an experimental Illustration of the advantage of non-minimal case.



An example of PnP in the outlier-free case.

RANSAC

- Comparison Between EM and RANSAC

sampling is random

- ✓ RANSAC is non-deterministic: every time you run it you may get a different result. The reason is that RANSAC selects minimal-case solution at random, leading to a random hypotheses' generation process. Conversely, EM is deterministic.

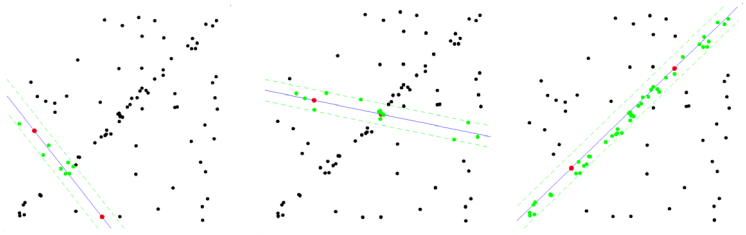
- ✓ RANSAC is not sensitive to initial condition, but EM is sensitive.

RANSAC

➢ Discussion about Number of Iterations

✓ Let us take 2D line fitting for example.

- Ideally: check all possible combinations of 2 points in a dataset of N points.
- Number of all pairwise combinations: $\frac{N(N-1)}{2}$
- However, it is computationally unfeasible if N is too large.



RANSAC

➢ Discussion about Number of Iterations

Do we really need to check all possibilities or can we stop RANSAC after some iterations?

- It is feasible to only validate a subset of all combinations, if we have a rough estimate of the inlier ratio in our dataset.
- This can be done in a probabilistic way. Derivation is relatively complicated. In our context, we only consider the following conclusion [1].
- At a certain confidence level *p* (such as 99%), we can guarantee that we have at least one valid sampling among *k* samplings.

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$

A parameter computed by inlier ratio

[1] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, 1981.

Robust Kernel (M-estimator)

- Overview

✓ Recap on photometric loss for direct SLAM

3D point back-projected by unknown depth

$$P^i, R, T = \arg\min_{P^i,R,T} \sum_{i=1}^{N} \rho\left(I_{k-1}(p^i_{k-1}) - I_k\left(\pi(P^i, K, R, T)\right)\right)$$

p^i_{k-1}: Known

P^i: Unknown depth

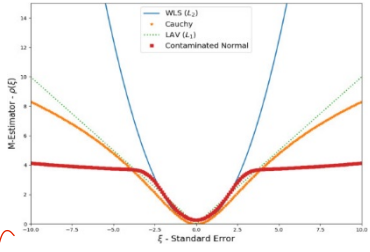
R, T: Unknown pose

M-estimator such as L_2, L_1, Huber ..

affected by outlier

robust function

M-estimator functions (e.g., L_2, L_1, Huber)



Error (e.g., photometric error or re-projection error of a pair of points)

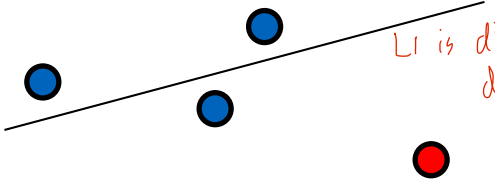
Robust Kernel (M-estimator)

Huber : combination of L_1, L_2

➢ An Intuitive Illustration

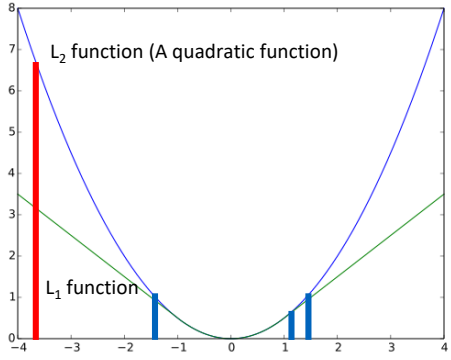
✓ Let's take straight line fitting problem for example.

- Recall that we aim to minimize an objective function.
- L_2 loss: Sum of squared errors is sensitive to outliers since it will amplify the effect of outliers.
- To solve this problem, Huber loss [1] was proposed. It is the combination of L_1 and L_2 losses.



L_1 is more robust to outlier

L_1 is difficult to derivate



[1] Huber, Peter J. (1964). "Robust Estimation of a Location Parameter". Annals of Statistics. 53 (1): 73–101

Summary

- Problem Formulation
- Expectation-maximization
- RANSAC
- Robust Kernel (M-estimator)

Thank you for your listening!
If you have any questions, please come to me :-)