Lecture 12 – M Estimator

Dev 04th, 2018

Danping Zou,

Associate Professor

Institute for Sensing and Navigation

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Outline

- Maximum Likelihood Estimator (MLE) vs Nonlinear Least Squares
- Robust estimation
- M-estimator
  - Loss function
  - Example - Pose estimation

Outline

- Maximum Likelihood Estimation (**MLE**)
  - Bayes theorem
  - Maximum A Posteriori (**MAP**)
  - Maximum Likelihood Estimation (**MLE**)
  - Parameter estimation
  - Logarithm of a likelihood with Gaussian distribution

Bayes theorem

- Given two random variables, X and Y, according to Bayes theorem

$$P(X,Y) = P(X|Y)P(Y)$$

Joint distribution = **Conditional distribution** x **Priori**

Bayes theorem

- Let $X = \Theta$ be some parameters we want to estimate, and Y be the observation data.

$$P(\Theta|Y)P(Y) = P(Y|\Theta)P(\Theta)$$

$P(\Theta|Y)$: **Posteriori**

$P(Y|\Theta)$: **Likelihood**

$P(\Theta)$: **Priori**

Bayes theorem

- Posteriori – The distribution of the state on the observation.

$$P(\Theta|Y) = \frac{P(Y|\Theta)P(\Theta)}{P(Y)}$$

$$= \frac{P(Y|X)P(\Theta)}{\sum_{\Theta'} P(Y|\Theta')P(\Theta')}$$

$$P(\Theta|Y) \propto P(Y|\Theta)P(\Theta)$$

Posteriori $\propto$ Likelihood x Priori

Maximum A Posteriori (MAP)

- Maximum a posteriori (MAP) estimation is to find an estimate that satisfies :

$$\Theta_{MAP} = \arg\max_{\Theta} P(\Theta|Y)$$

$$= \arg\max_{\Theta} P(Y|\Theta)P(\Theta)$$

Maximum Likelihood Estimation

- Maximum likelihood estimation (MLE) is to find

$$\Theta_{MLE} = \arg\max_{\Theta} P(Y|\Theta)$$

- If the priori distribution is a uniform distribution, we have

$$P(\Theta) = consant.$$

- Then MAP estimation and MLE estimation are equivalent.

$$\arg\max_{\Theta} P(Y|\Theta)P(\Theta) = \arg\max_{\Theta} P(Y|\Theta)$$

Parameter estimation

- Let the parameter variable belong to some parameter space

$$\Theta \in \mathcal{X}$$

- Let $h : \mathbb{R}^m \times \mathcal{X} \rightarrow \mathbb{R}^n$ be a mathematical model that describes the relationship between a input and an outcome, regulated by the model parameters Θ

$$\hat{\boldsymbol{y}}_i = h(\boldsymbol{x}_i, \Theta)$$

Parameter estimation

- Usually, the model is perturbed by some noise. Hence what we observe from the model is described by a noise $\boldsymbol{n}$

$$\boldsymbol{y}_i \sim h(\boldsymbol{x}_i, \Theta) + \boldsymbol{n}$$

- Parameter estimation tries to find an estimate Θ^* that best depicts the observation data Y .

$$Y = \{\boldsymbol{x}_i \leftrightarrow \boldsymbol{y}_i\}$$

Parameter estimation

- The likelihood is then computed as

$$P(Y|\Theta) = \Pi_i P(\boldsymbol{y}_i|\boldsymbol{x}_i, \Theta)$$

- Usually we prefer to get the logarithm of likelihood

$$L(Y, \Theta) = \log(P(Y|\Theta)) = \sum_i \log(P(\boldsymbol{y}_i|\boldsymbol{x}_i, \Theta))$$

$$= \sum_i \ell(\boldsymbol{y}_i|\boldsymbol{x}_i, \Theta)$$

- Each individual likelihood relies on how the observation noise $\boldsymbol{n}$ is defined.

$$\boldsymbol{y}_i \sim h(\boldsymbol{x}_i, \Theta) + \boldsymbol{n}$$

Examples

Linear regression problem

- The parameters we want to estimate is the two line parameters .

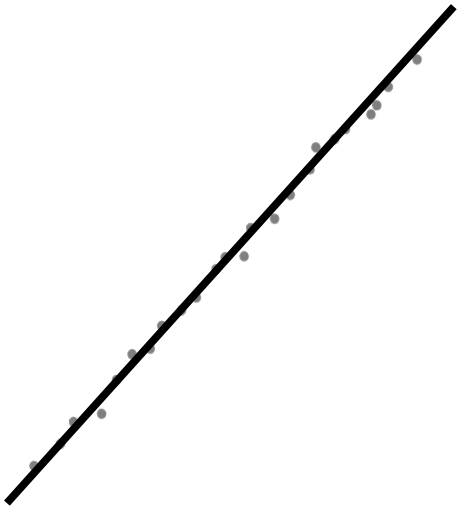
$$\Theta = (a, b)$$

- The model is the line equation

$$y_i = ax_i + b$$

- The observation is the set of coordinates:

$$Y = \{x_i \in \mathbb{R} \leftrightarrow y_i \in \mathbb{R}\}$$

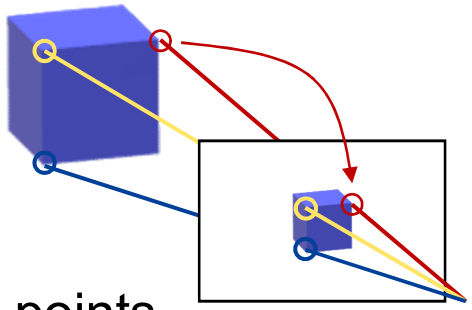


Examples

Pose estimation problem:

- The parameters we want to estimate is the camera pose

$$\Theta = (\mathbf{R}, \boldsymbol{t}) \in \mathcal{X} = SO(3) \times \mathbb{R}^{3\times 1}$$



- The model is perspective projection

$$\boldsymbol{x}_i = \mathcal{P}(\boldsymbol{X}_i, \mathbf{R}, \boldsymbol{t})$$

- The observation is the set of 3D-2D corresponding points

$$Y = \{\boldsymbol{X}_i \in \mathbb{R}^{3\times 1} \leftrightarrow \boldsymbol{x}_i \in \mathbb{R}^{2\times 1}\}$$

Parameter estimation

- Now lets consider the likelihood function

- What will the individual likelihood look like with different types noise n?

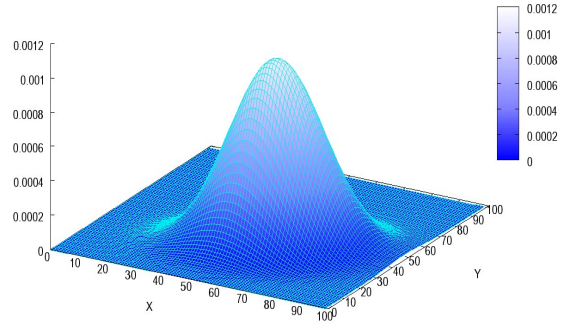
$$\boldsymbol{y}_i \sim h(\boldsymbol{x}_i, \Theta) + \boldsymbol{n}$$

- First, lets consider the noise be a Gaussian noise.

Gaussian distribution

- Gaussian Probability Density Function is defined as

$$P(\boldsymbol{x}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))$$



- A Gaussian random variable is

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Gaussian noise assumption

- For the model noise $\boldsymbol{n}$ of $\boldsymbol{y}_i \sim h(\boldsymbol{x}_i, \Theta) + \boldsymbol{n}$,now we have

$$\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

- Or equally, we have

$$\boldsymbol{y}_i \sim \mathcal{N}(h(\boldsymbol{x}_i, \Theta), \Sigma)$$

- Usually the covariance matrix is a diagonal matrix $\Sigma = \sigma^2\mathbf{I}$

Gaussian noise assumption

- Given K observations, the logarithm of the likelihood is computed as

$$L(Y,\Theta) = \sum_i \ell(\boldsymbol{y}_i|\boldsymbol{x}_i,\Theta) = \sum_i \log(P(\boldsymbol{y}_i|\boldsymbol{x}_i,\Theta))$$

- If $P(\cdot)$ is a Gaussian distribution, we have

$$P(\boldsymbol{y}_i|\boldsymbol{x}_i,\Theta) \propto \exp(-\tfrac{1}{2}(\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta))^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta))$$

$$\ell(\boldsymbol{y}_i|\boldsymbol{x}_i,\Theta) = -\tfrac{1}{2}(\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta))^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta) + const.$$

Gaussian noise assumption

- The logarithm of the likelihood is given by

$$L(Y,\Theta) = \sum_i \ell(\boldsymbol{y}_i|\boldsymbol{x}_i,\Theta)$$

$$= -\tfrac{1}{2}\sum_i (\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta))^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta)) + const.$$

$$= -\tfrac{1}{2\sigma^2}\sum_i \|\boldsymbol{y}_i - h(\boldsymbol{x}_i,\Theta)\|^2 + const. \ (\Sigma = \sigma^2\mathbf{I})$$

- Maximizing the likelihood is equivalent to minimize the **sum of squares**.

Summary

- Bayes theorem

$$P(\Theta|Y)P(Y) = P(Y|\Theta)P(\Theta)$$

- Maximum A Posteriori (MAP)

$$\Theta_{MAP} = \arg\max_{\Theta} P(\Theta|Y) = \arg\max_{\Theta} P(Y|\Theta)P(\Theta)$$

- Maximum Likelihood Estimation (MLE)

$$\Theta_{MLE} = \arg\max_{\Theta} P(Y|\Theta)$$

- Under Gaussian noise assumption, MLE = least squares

$$\arg\max_{\Theta} L(Y, \Theta) = \arg\max_{\Theta} \sum_i \ell(\boldsymbol{y}_i|\boldsymbol{x}_i, \Theta)$$

$$= \arg\min_{\Theta} \sum_i \|\boldsymbol{y}_i - h(\boldsymbol{x}_i, \Theta)\|^2$$

M-estimator

- **M-estimators** are a broad class of estimators in statistics, which are obtained as the minima of sums of functions of the data.

$$\min \sum_i \ell(\|\boldsymbol{r}_i\|)$$

- **M** represents for "**Maximum likelihood-type**"
- For the least squares estimator

$$\ell(\|\boldsymbol{r}_i\|) = \|\boldsymbol{r}_i\|^2$$

M-estimator

- How do we address outliers in non-linear least squares problem?

$$\min \frac{1}{2} \sum_i \|\boldsymbol{r}_i(\boldsymbol{x})\|^2$$

M-estimator

- We can use a loss function $\rho(s)$ to regulate the impact of outliers

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_i \rho(\|\boldsymbol{r}_i(\boldsymbol{x})\|^2)$$

$$\ell(\|\boldsymbol{r}_i\|) = \rho(\|\boldsymbol{r}_i(\boldsymbol{x})\|^2)$$

$\rho(s) = s$	$\ell(t) = t^2$	$\frac{1}{2}\sum_i \|\boldsymbol{r}_i(\boldsymbol{x})\|^2$
$\rho(s) = \sqrt{s}$	$\ell(t) = t$	$\frac{1}{2}\sum_i \|\boldsymbol{r}_i(\boldsymbol{x})\|$

M-estimator

- We can use a loss function $\rho(s)$ to regulate the impact of outliers

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_i \rho(\|\boldsymbol{r}_i(\boldsymbol{x})\|^2)$$

$$f(\boldsymbol{x}+\Delta\boldsymbol{x}) = \frac{1}{2}\sum_i \rho(\|\boldsymbol{r}_i(\boldsymbol{x}+\Delta\boldsymbol{x}))\|^2)$$

$$\frac{\partial f(\boldsymbol{x}+\Delta\boldsymbol{x})}{\partial \Delta\boldsymbol{x}} = \mathbf{0} \rightarrow \Delta\boldsymbol{x}$$

M-estimator

- We can use a loss function $\rho(s)$ to regulate the impact of outliers

$$f(\boldsymbol{x}+\Delta\boldsymbol{x}) = \frac{1}{2}\sum_i \rho(\|\boldsymbol{r}_i(\boldsymbol{x}+\Delta\boldsymbol{x})\|^2)$$

$$\frac{\partial f(\boldsymbol{x}+\Delta\boldsymbol{x})}{\partial \Delta\boldsymbol{x}} = \frac{1}{2}\sum_i \frac{\partial \rho(\|\boldsymbol{r}_i(\boldsymbol{x}+\Delta\boldsymbol{x})\|^2)}{\partial \Delta\boldsymbol{x}}$$

$$\approx \frac{1}{2}\sum_i \frac{\partial \rho((\boldsymbol{r}_i - \mathbf{H}_i\Delta\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{r}_i - \mathbf{H}_i\Delta\boldsymbol{x}))}{\partial \Delta\boldsymbol{x}}$$

M-estimator

- We can use a loss function $\rho(s)$ to regulate the impact of outliers

$$\frac{1}{2}\sum_i \frac{\partial \rho((\boldsymbol{r}_i - \mathbf{H}_i \Delta \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{r}_i - \mathbf{H}_i \Delta \boldsymbol{x}))}{\partial \Delta \boldsymbol{x}}$$

$$\frac{\partial \rho(\cdot)}{\partial \Delta \boldsymbol{x}} = \frac{\partial \rho(\cdot)}{\partial s_i}\frac{\partial}{\partial \Delta \boldsymbol{x}}(\boldsymbol{r}_i - \mathbf{H}_i \Delta \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{r}_i - \mathbf{H}_i \Delta \boldsymbol{x})$$

$$= 2\frac{\partial \rho(\cdot)}{\partial s_i}(\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i \Delta \boldsymbol{x} - \mathbf{H}_i^{\mathrm{T}}\boldsymbol{r}_i)$$

$$(s_i = \|\boldsymbol{r}_i\|^2)$$

M-estimator

- The loss function $\rho(s)$ can be designed to regulate the impact of outliers

$$\frac{\partial f(\boldsymbol{x}+\Delta\boldsymbol{x})}{\partial \Delta \boldsymbol{x}} \approx \sum_i \frac{\partial \rho(\cdot)}{\partial s_i}(\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\Delta\boldsymbol{x} - \mathbf{H}_i^{\mathrm{T}}\boldsymbol{r}_i)$$

- Let $\mathbf{w}_i = \frac{\partial \rho(\cdot)}{\partial s_i}\mathbf{I}_{M\times M}$, we have

$$\frac{\partial f(\boldsymbol{x}+\Delta\boldsymbol{x})}{\partial \Delta \boldsymbol{x}} \approx \sum_i \mathbf{H}_i^{\mathrm{T}}\mathbf{w}_i\mathbf{H}_i\Delta\boldsymbol{x} - \mathbf{H}_i^{\mathrm{T}}\mathbf{w}_i\boldsymbol{r}_i$$

M-estimator

- Let

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & 0 & \dots & \dots & \dots \\ 0 & \mathbf{w}_2 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \mathbf{w}_i & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_i \\ \vdots \end{bmatrix} \quad \boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_i \\ \vdots \end{bmatrix}$$

$$\frac{\partial}{\partial \Delta \boldsymbol{x}} f(\boldsymbol{x} + \Delta \boldsymbol{x}) = \mathbf{H}^{\mathrm{T}} \mathbf{W} \mathbf{H} \Delta \boldsymbol{x} - \mathbf{H}^{\mathrm{T}} \mathbf{W} \boldsymbol{r} = \mathbf{0}$$

$$\Delta \boldsymbol{x} = (\mathbf{H}^{\mathrm{T}} \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^{\mathrm{T}} \mathbf{W} \boldsymbol{r}$$

- It becomes a weighted least squares problem

M-estimator

- We can choose a loss function to make it give less weights on outliers and more weights on inliers.

$$\frac{\partial \rho(\cdot)}{\partial s_i} \Rightarrow \mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & 0 & \dots & \dots & \dots \\ 0 & \mathbf{w}_2 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \mathbf{w}_i & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

$$\frac{\partial \rho(\cdot)}{\partial s_i} = 1 \quad (s_i \leq 1) \qquad \text{Inliers}$$

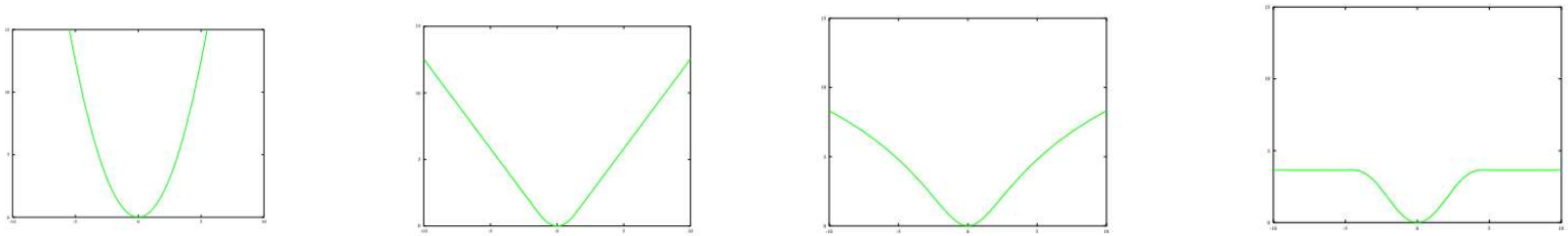
$$\frac{\partial \rho(\cdot)}{\partial s_i} < 1 \quad (s_i > 1) \qquad \text{Outliers}$$

$$s_i = \|\boldsymbol{r}_i\|^2 / (r_{max}^2)$$

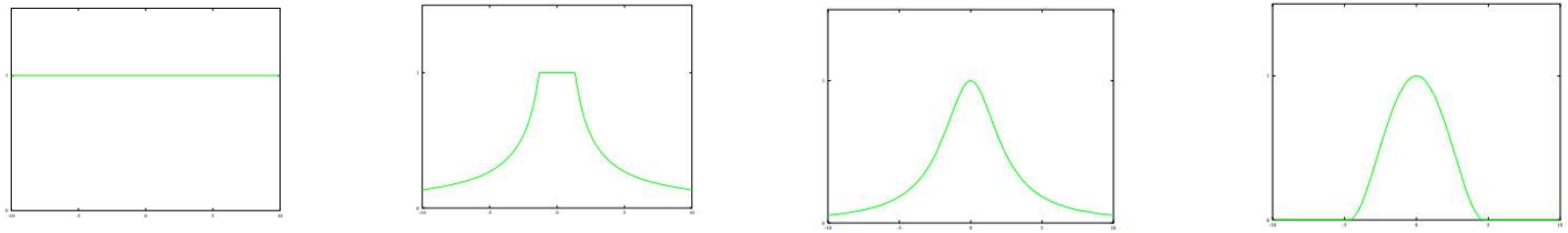
Robust estimator

- Least squares $\rightarrow \rho(s) = s \quad \rightarrow \frac{\partial \rho}{\partial s} = 1$
- We can choose a loss function to make it give less weights on outliers and more weights on inliers.

Null loss	Huber	Cauchy	Turkey
$\rho(s) = s$	$\rho(s) = \begin{cases} s & s \leq 1 \\ 2\sqrt{s} - 1 & s > 1 \end{cases}$	$\rho(s) = \log(1+s)$	$\rho(s) = \begin{cases} \frac{1}{6}(1-(1-s)^3) & s \leq 1 \\ 1 & s > 1 \end{cases}$



$\rho(x^2)$



$w \sim \frac{\partial \rho}{\partial s}$

Gauss-Newton method

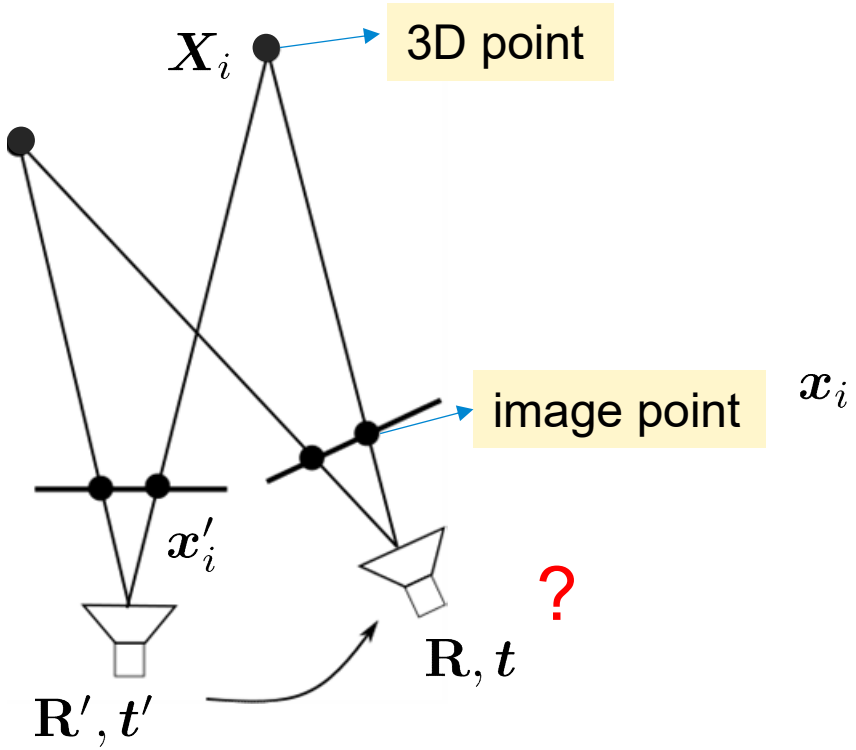
- Step 1 : Start from an initial point x_0
- Step 2 : solve an incremental step $\Delta \mathbf{x}$,

$$\Delta \boldsymbol{x} = (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\boldsymbol{r}$$

- Step 3 : Update the solution $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta \boldsymbol{x}$
- *Repeat Step 2~ Step 3 until convergence.*

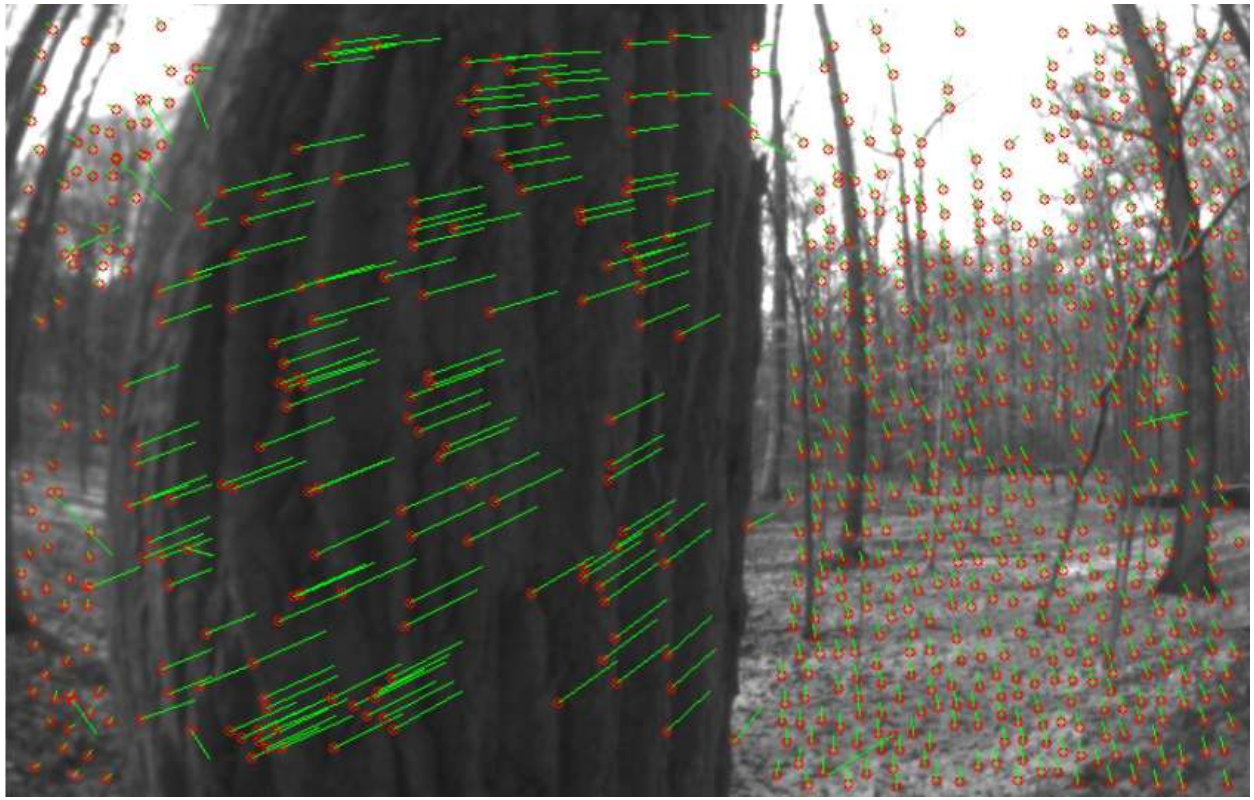
Robust pose estimation

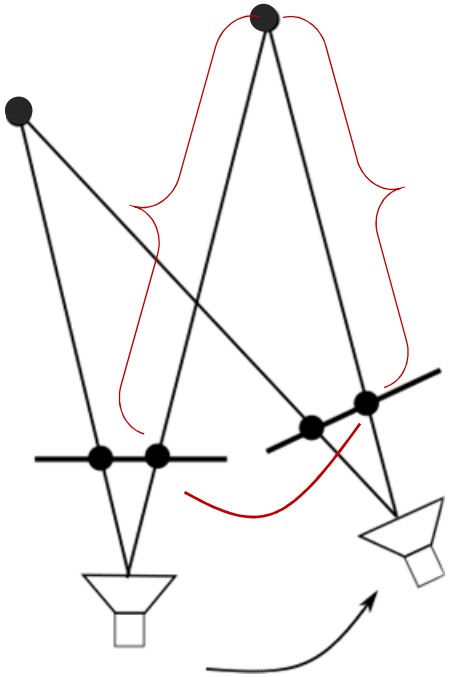
- Given 3D points and their corresponding images, how do we compute the camera pose ? (given that the camera is calibrated)
- We know the camera pose in the previous frame.



Robust pose estimation

- Step 1 – feature tracking or matching
  - Note that there are only a few outliers





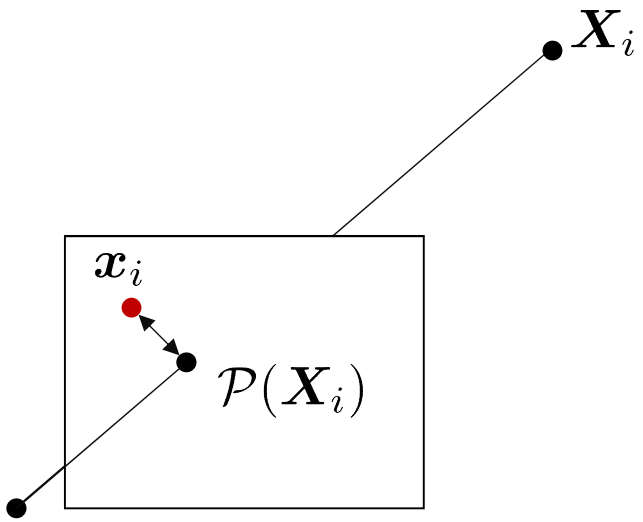
Robust pose estimation

- Step 2 – Minimize the re-projection error

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \mathbf{R}, \boldsymbol{t})\|^2$$

Image point

Projected point



Robust pose estimation

- Step 3 – add loss function to deal with outliers

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{i=1}^{n} \rho(\|\boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \mathbf{R}, \boldsymbol{t})\|^2)$$

Robust pose estimation

- Step 4 – Iterative optimization
  - Gauss-Newton or Levenberg-Marquardt algorithm can be applied.

$$\Theta \leftarrow \Theta \boxplus \Delta\Theta$$

$$\begin{pmatrix} \mathbf{R} \\ \boldsymbol{t} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{R}\exp(\Delta\theta^{\wedge}) \\ \boldsymbol{t} + \Delta\boldsymbol{t} \end{pmatrix}$$

- All we need is to solve the incremental step $\Delta\Theta = \begin{pmatrix} \Delta\theta \\ \Delta\boldsymbol{t} \end{pmatrix}$

Iterative optimization

- For one pair of corresponding points, $\boldsymbol{x}_i \leftrightarrow \boldsymbol{X}_i$, we rewrite the objective function as the following

$$\sum_i \rho(\|\boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \Theta)\|^2)$$

- By first-order approximation, we have

$$\begin{aligned}
&\approx \rho(\|\boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \Theta) - \mathbf{J}_i \Delta\Theta\|^2) \\
&= \rho(\|\boldsymbol{r}_i - \mathbf{J}_i \Delta\Theta\|^2) \\
&= \rho(\boldsymbol{r}_i^{\mathrm{T}} \boldsymbol{r}_i - 2\boldsymbol{r}_i^{\mathrm{T}} \mathbf{J}_i \Delta\Theta + \Delta\Theta^{\mathrm{T}} \mathbf{J}_i^{\mathrm{T}} \mathbf{J}_i \Delta\Theta)
\end{aligned}$$

$$\frac{\partial f(\Theta_0 + \Delta\Theta)}{\partial \Delta\Theta} = \sum_i \frac{\partial \rho}{\partial s_i} (\mathbf{J}_i^{\mathrm{T}} \mathbf{J}_i^{\mathrm{T}} \Delta\Theta - \mathbf{J}_i^{\mathrm{T}} \boldsymbol{r}_i) = \mathbf{0}$$

Iterative optimization

- Consider the following equation:

$$\sum_i \frac{\partial \rho}{\partial s_i}\left(\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i\Delta\Theta - \mathbf{J}_i^{\mathrm{T}}\boldsymbol{r}_i\right) = \mathbf{0}$$

$$\mathbf{J}_i \in \mathbb{R}^{2\times 6} = \frac{\partial \mathcal{P}(\boldsymbol{X}_i)}{\partial \Delta\Theta}$$

$$\boldsymbol{r}_i = \boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \Theta) \in \mathbb{R}^{2\times 1}$$

$$\mathbf{W} = \begin{bmatrix} \frac{\partial \rho}{\partial s_1} & 0 & 0 & \cdots & \cdots & \cdots \\ 0 & \frac{\partial \rho}{\partial s_1} & 0 & \cdots & \cdots & \cdots \\ 0 & 0 & \frac{\partial \rho}{\partial s_2} & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \frac{\partial \rho}{\partial s_2} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \end{bmatrix}$$

$$\Rightarrow \quad \mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\Delta\Theta = \mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{r}$$

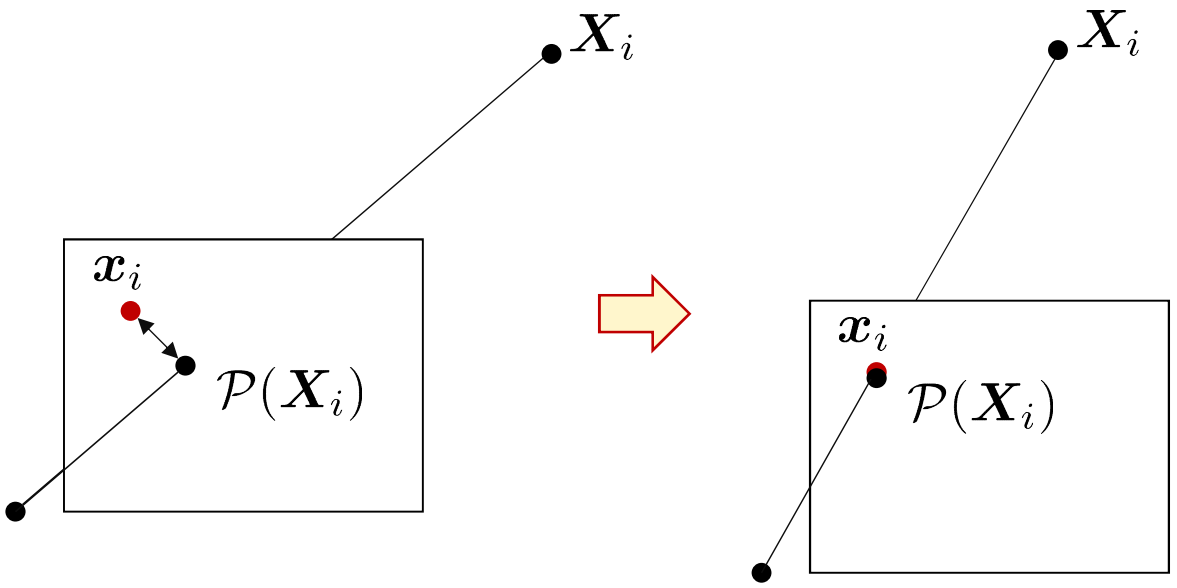
$$\Updownarrow$$

$$\mathbf{W}\mathbf{J}^{\mathrm{T}}\Delta\Theta = \mathbf{W}\boldsymbol{r}$$

Robust pose estimation

- Step 5 – get the final estimates

$$\mathbf{R}^*, \boldsymbol{t}^* = \min_{\mathbf{R},\boldsymbol{t}} \sum_{i=1}^{n} \rho(\|\boldsymbol{x}_i - \mathcal{P}(\boldsymbol{X}_i, \mathbf{R}, \boldsymbol{t})\|^2)$$



Summary

- Add an extra loss function to improve the robustness

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_i \rho(\|\boldsymbol{r}_i(\boldsymbol{x})\|^2)$$

- The loss function is related to the weights

$$\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\Delta\Theta = \mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{r}$$