Estimation of Camera Projection Matrix, Nonlinear

Computer Vision II

CSE 252B

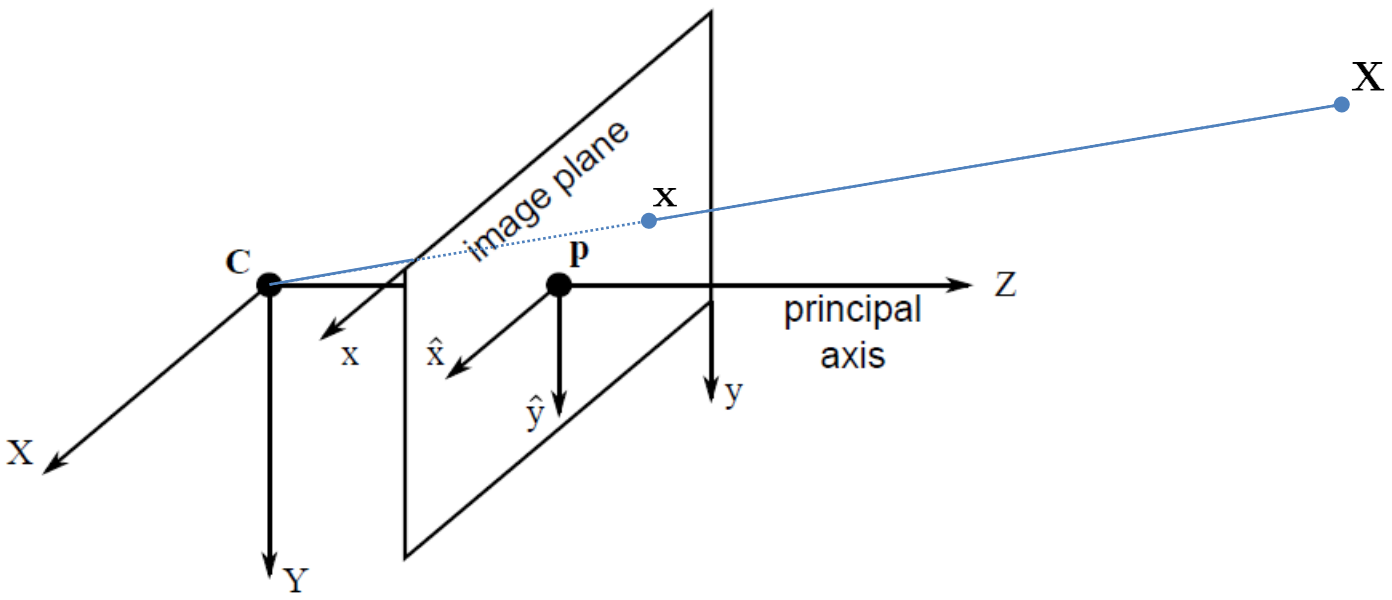
Lecture 6

Announcements

- Assignment 2 is due Feb 4, 11:59 PM
- Reading
  - Sections 7.2 and 4.5

Single view geometry

Camera projection



Camera projection matrix

- Projective camera

When camera projection matrix is known up to projective transformation
3×4 homogeneous (i.e., defined up to nonzero scale) matrix (11 degrees of freedom)

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$

When parameters of camera projection matrix are known or known up to nonzero scale (11 degrees of freedom)

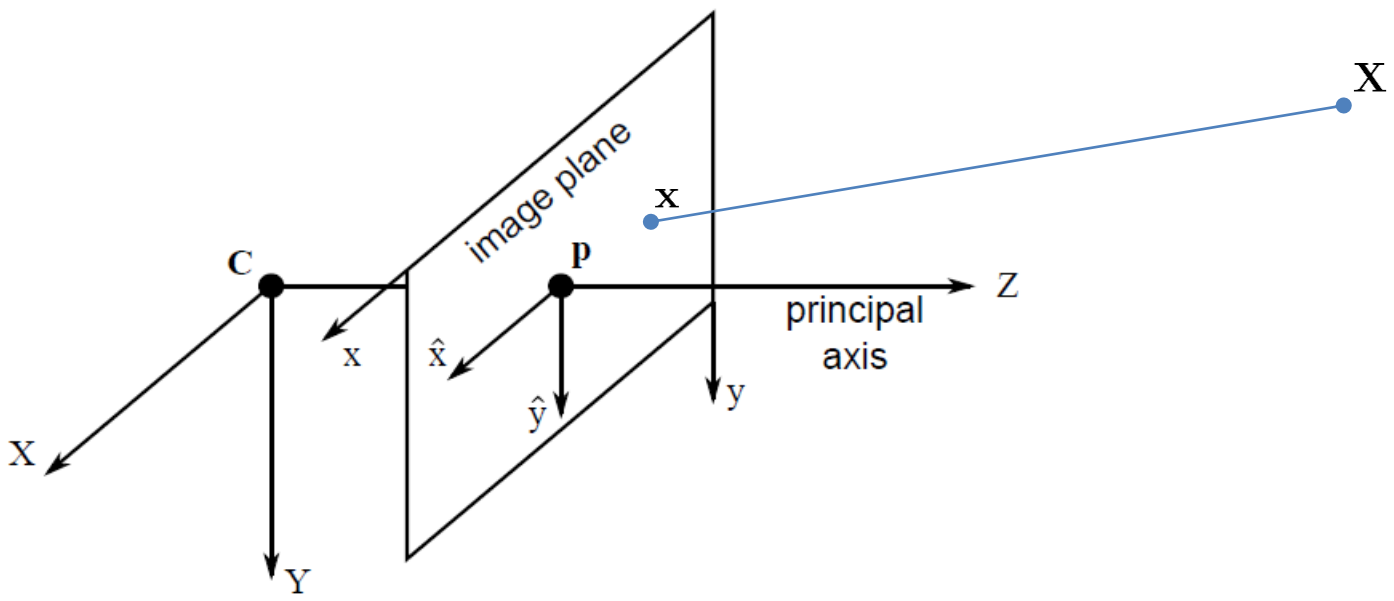
$$\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$$
$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \,|\, -\tilde{\mathbf{C}}]$$

where

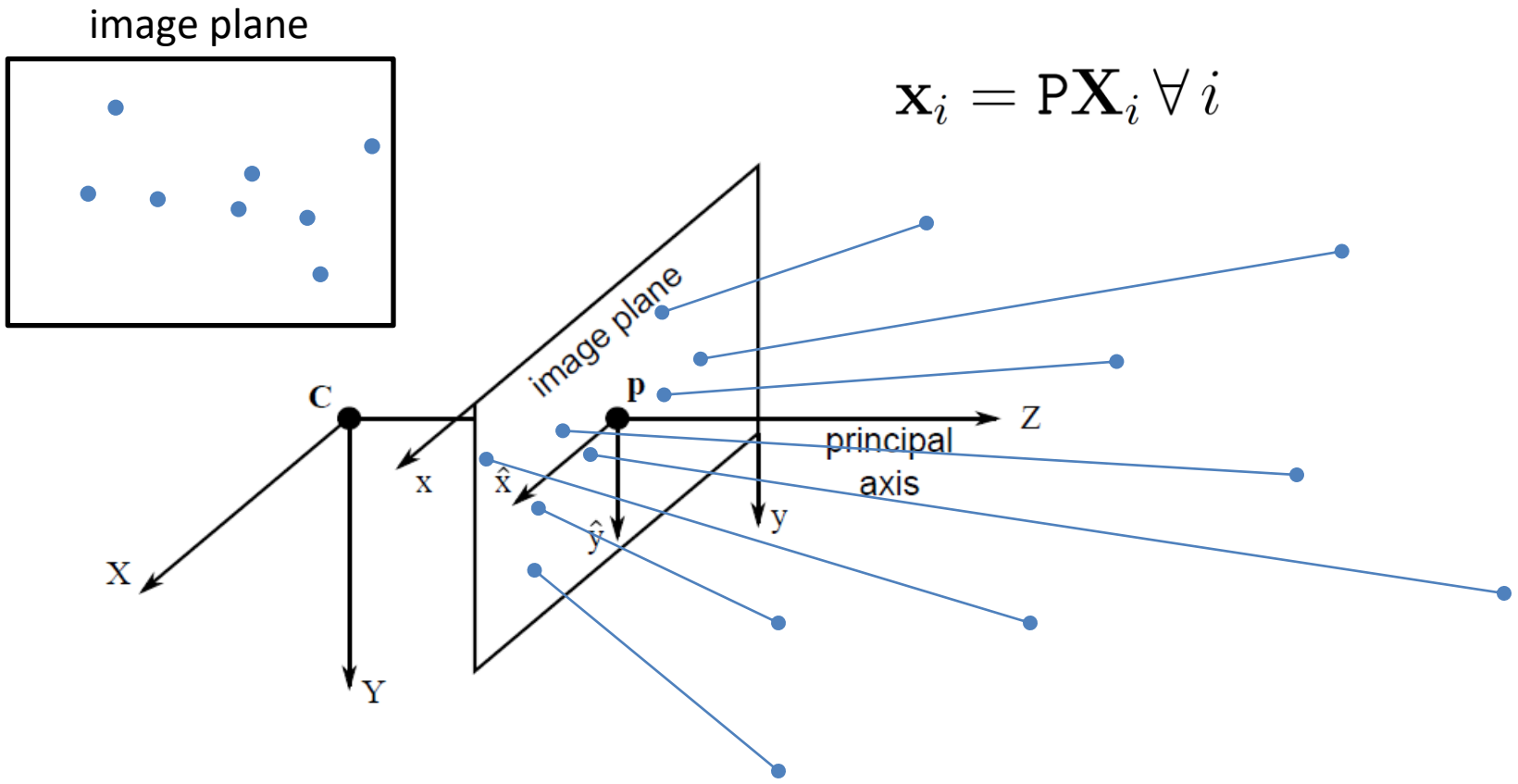
$\mathtt{K}$	Camera calibration matrix
$\mathtt{R}$	Camera rotation matrix
$\mathbf{t}$	Camera translation vector
$\mathbf{C}$	Camera center

Camera projection matrix

$$\mathbf{x} = \mathrm{P}\mathbf{X}$$

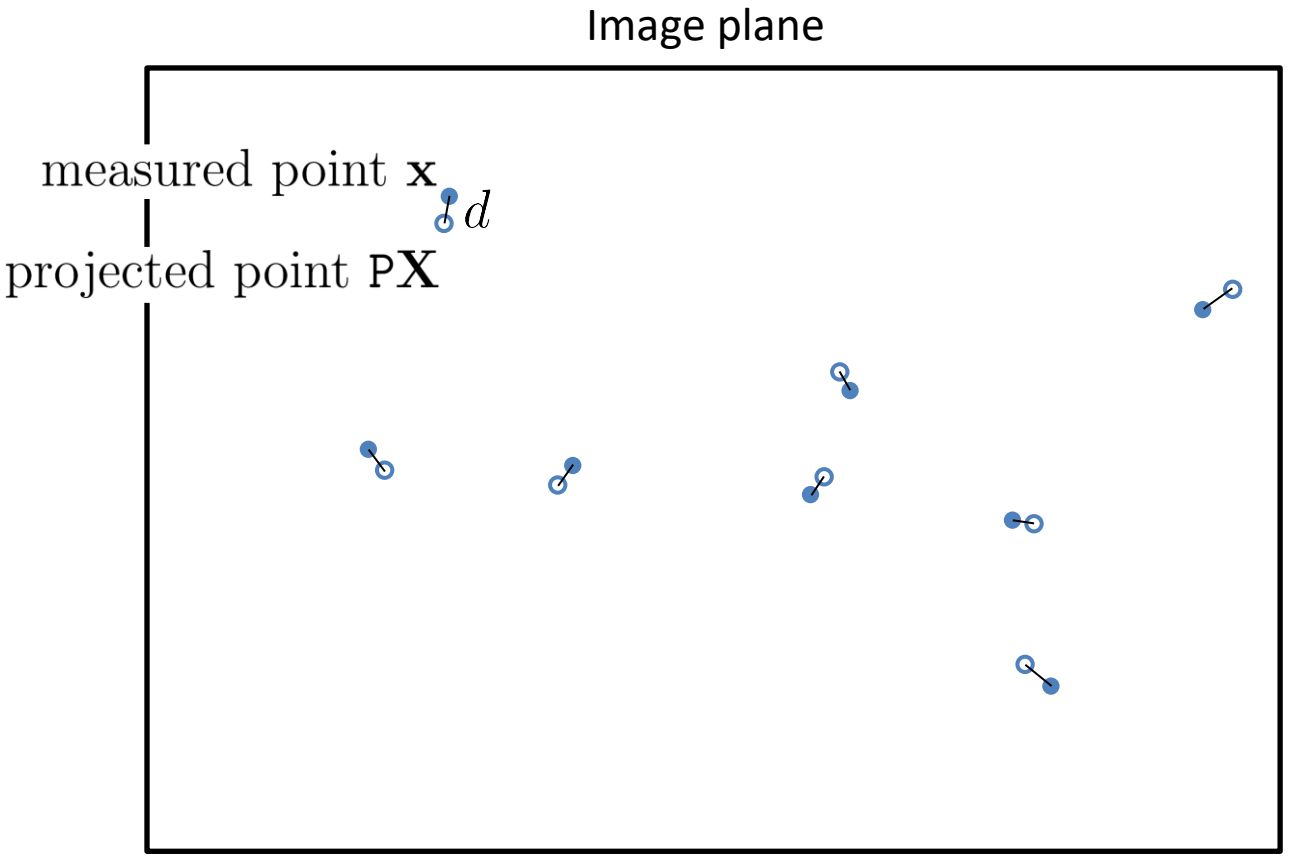


Camera projection matrix



Camera projection matrix

- Error in image point measurements



Camera projection matrix

Linear projection of points in homogeneous coordinates

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} Xp_{11} + Yp_{12} + Zp_{13} + Tp_{14} \\ Xp_{21} + Yp_{22} + Zp_{23} + Tp_{24} \\ Xp_{31} + Yp_{32} + Zp_{33} + Tp_{34} \end{bmatrix}$$

Nonlinear projection of points in inhomogeneous coordinates

$$\tilde{x} = \frac{x}{w} = \frac{Xp_{11} + Yp_{12} + Zp_{13} + Tp_{14}}{Xp_{31} + Yp_{32} + Zp_{33} + Tp_{34}} \text{ and}$$

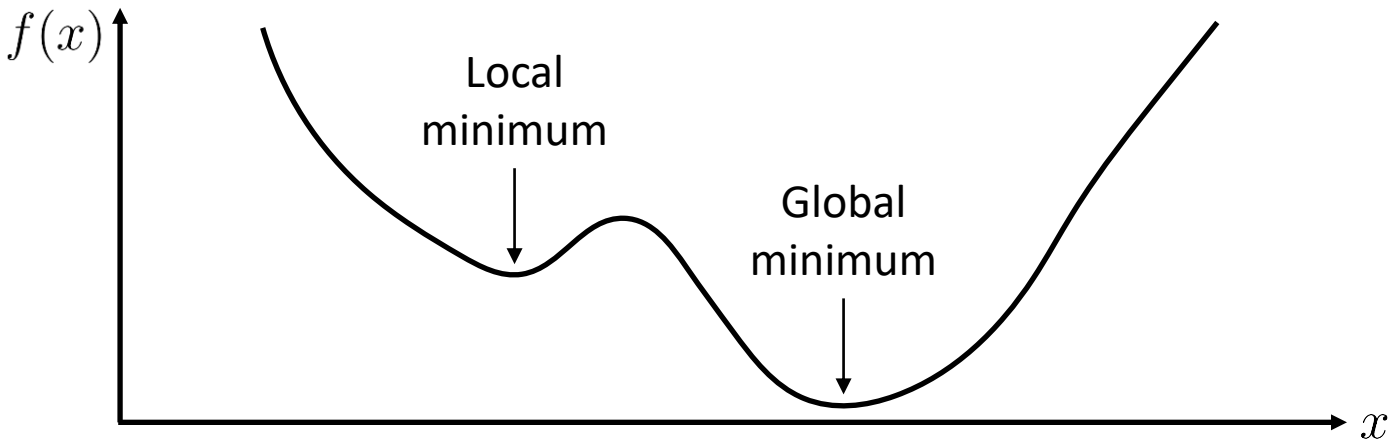
$$\tilde{y} = \frac{y}{w} = \frac{Xp_{21} + Yp_{22} + Zp_{23} + Tp_{24}}{Xp_{31} + Yp_{32} + Zp_{33} + Tp_{34}}$$

Do not forget this is a nonlinear projection

Estimation of camera projection matrix

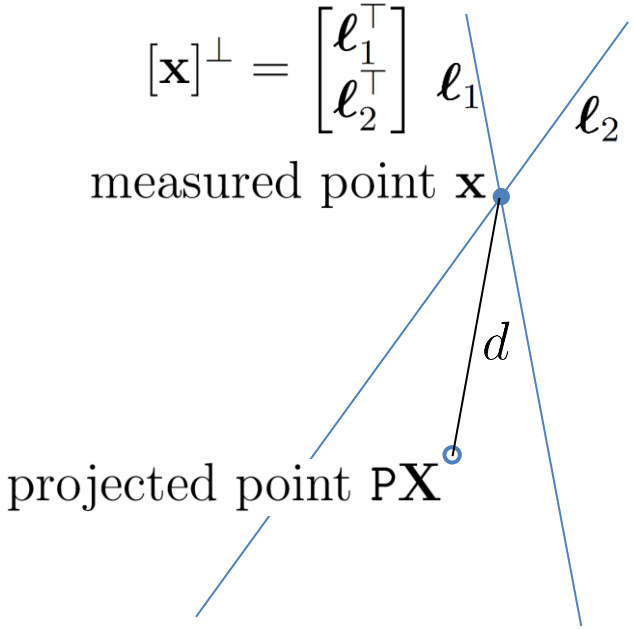
$$\mathbf{x}_i = \mathrm{P}\mathbf{X}_i \,\forall\, i, \text{ solve for } \mathrm{P}$$

- Given point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$
- Minimize geometric error $\sum_i d(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i)^2$
- A nonlinear mapping requires a nonlinear optimization problem solver
  - Use linear estimation for initial estimate (to mitigate converging to local optimum)
  - Iterative process to determine global optimum



Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

- Minimize algebraic error between the projected point and two lines intersecting at the measured point



$$[\mathbf{x}]^\perp \mathbf{x} = \mathbf{0} \qquad \mathbf{x} = \mathtt{P}\mathbf{X}$$
$$[\mathbf{x}]^\perp \mathtt{P}\mathbf{X} = \mathbf{0}$$

Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

Given point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$

$$\mathbf{x}_i = \mathtt{P}\mathbf{X}_i \,\forall\, i, \text{ solve for } \mathtt{P}$$

$$[\mathbf{x}]^{\perp}{}_i \mathtt{P}\mathbf{X}_i = \mathbf{0} \,\forall\, i, \text{ solve for } \mathtt{P}$$

$$\left([\mathbf{x}_i]^{\perp} \otimes \mathbf{X}_i^{\top}\right) \mathbf{p} = \mathbf{0} \,\forall\, i, \text{ solve for } \mathbf{p} = \text{vec}(\mathtt{P}^{\top})$$

$$\begin{bmatrix} [\mathbf{x}_1]^{\perp} \otimes \mathbf{X}_1^{\top} \\ [\mathbf{x}_2]^{\perp} \otimes \mathbf{X}_2^{\top} \\ \vdots \\ [\mathbf{x}_n]^{\perp} \otimes \mathbf{X}_n^{\top} \end{bmatrix} \mathbf{p} = 0$$

$$\mathtt{A}\mathbf{p} = \mathbf{0}, \text{ where } \mathtt{A} = \begin{bmatrix} [\mathbf{x}_1]^{\perp} \otimes \mathbf{X}_1^{\top} \\ [\mathbf{x}_2]^{\perp} \otimes \mathbf{X}_2^{\top} \\ \vdots \\ [\mathbf{x}_n]^{\perp} \otimes \mathbf{X}_n^{\top} \end{bmatrix}$$

But, data normalization must be used

Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

- Algorithm

1. Data normalize the points

$$\mathbf{x}_{\mathrm{DN},i} = \mathtt{T}\mathbf{x}_i \,\forall\, i$$
$$\mathbf{X}_{\mathrm{DN},i} = \mathtt{U}\mathbf{X}_i \,\forall\, i$$

2. Estimate the data normalized camera projection matrix $\mathtt{P}_{\mathrm{DN}}$ from the data normalized points

3. Data denormalize the data normalized camera projection matrix

$$\begin{aligned}
\mathbf{x}_{\mathrm{DN},i} &= \mathtt{P}_{\mathrm{DN}}\mathbf{X}_{\mathrm{DN},i} \,\forall\, i \\
\mathtt{T}\mathbf{x}_i &= \mathtt{P}_{\mathrm{DN}}\mathtt{U}\mathbf{X}_i \\
\mathbf{x}_i &= \mathtt{T}^{-1}\mathtt{P}_{\mathrm{DN}}\mathtt{U}\mathbf{X}_i \\
\mathbf{x}_i &= \mathtt{P}\mathbf{X}_i \,\forall\, i, \text{ where } \mathtt{P} = \mathtt{T}^{-1}\mathtt{P}_{\mathrm{DN}}\mathtt{U}
\end{aligned}$$

Nonlinear estimation using iterative estimation methods

Measurement vector $\mathbf{X}$ ← Not 3D point

Parameter vector $\mathbf{P}$ ← Not camera projection matrix

Nonlinear function f, where $\mathbf{X} = f(\mathbf{P})$

Given an estimate of the parameter vector $\hat{\mathbf{P}}$ and assuming f is approximately linear in the region about $\hat{\mathbf{P}}$

$$\mathbf{X} \approx f(\hat{\mathbf{P}}) + \frac{\partial \hat{\mathbf{X}}}{\partial \hat{\mathbf{P}}}(\mathbf{P} - \hat{\mathbf{P}})$$

$$\mathbf{X} \approx \hat{\mathbf{X}} + \mathtt{J}(\mathbf{P} - \hat{\mathbf{P}}), \text{ where } \hat{\mathbf{X}} = f(\hat{\mathbf{P}}) \text{ and } \mathtt{J} = \frac{\partial \hat{\mathbf{X}}}{\partial \hat{\mathbf{P}}}$$

$$\mathbf{X} - \hat{\mathbf{X}} \approx \mathtt{J}(\mathbf{P} - \hat{\mathbf{P}})$$

$$\boldsymbol{\epsilon} \approx \mathtt{J}\boldsymbol{\delta}, \text{ solve for } \boldsymbol{\delta} \text{ where } \boldsymbol{\epsilon} = \mathbf{X} - \hat{\mathbf{X}} \text{ and } \boldsymbol{\delta} = \mathbf{P} - \hat{\mathbf{P}}$$

Then

$$\boldsymbol{\delta} = \mathbf{P} - \hat{\mathbf{P}}$$

$$\hat{\mathbf{P}} + \boldsymbol{\delta} = \mathbf{P}$$

Adjustment

Perform again (i.e., iterate) using resulting $\mathbf{P}$ as $\hat{\mathbf{P}}$

Nonlinear estimation using the Levenberg-Marquardt algorithm

Given

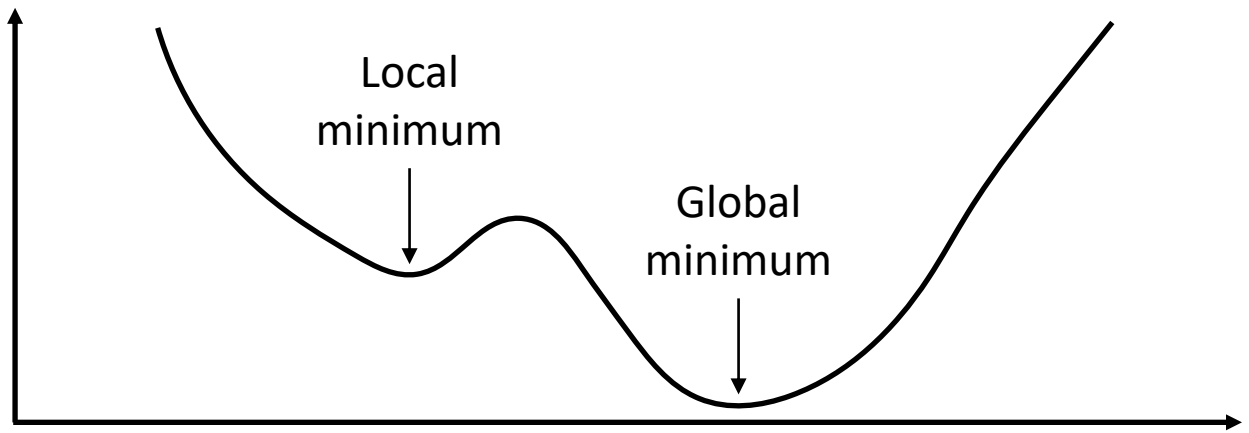
Measurement vector $\mathbf{X}$ with associated covariance matrix $\Sigma_{\mathbf{X}}$

Parameter vector $\hat{\mathbf{P}}$ (initial estimate)

Nonlinear function f, where $\hat{\mathbf{X}} = f(\hat{\mathbf{P}})$

Objective

Find $\hat{\mathbf{P}}$ that minimizes $\boldsymbol{\epsilon}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = \mathbf{X} - \hat{\mathbf{X}}$



Nonlinear estimation using the Levenberg-Marquardt algorithm

Algorithm

1. $\lambda = 0.001$ and $\boldsymbol{\epsilon} = \mathbf{X} - \hat{\mathbf{X}}$
2. Jacobian $\mathtt{J} = \dfrac{\partial \hat{\mathbf{X}}}{\partial \hat{\mathbf{P}}}$
3. Weighted least-squares normal equations $\left(\mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \mathtt{J}\right) \boldsymbol{\delta} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$
4. Augmented normal equations $\left(\mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \mathtt{J} + \lambda \mathtt{I}\right) \boldsymbol{\delta} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$, solve for $\boldsymbol{\delta}$
5. Candidate parameter vector $\hat{\mathbf{P}}_0 = \hat{\mathbf{P}} + \boldsymbol{\delta}$
6. $\hat{\mathbf{X}}_0 = f(\hat{\mathbf{P}}_0)$, then $\boldsymbol{\epsilon}_0 = \mathbf{X} - \hat{\mathbf{X}}_0$
7. If candidate cost $\boldsymbol{\epsilon}_0^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}_0 \geq \boldsymbol{\epsilon}^\top \Sigma_{\mathbf{X}}^{-1} \boldsymbol{\epsilon}$, then

   $\lambda = 10\lambda$

   Go to step 4 (this does not count as an iteration)

   else

   $\hat{\mathbf{P}} = \hat{\mathbf{P}}_0$ and $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_0$

   $\lambda = \lambda/10$

   Go to step 2 or terminate (this counts as an iteration)

Outer loop

Inner loop

Adjustment

Nonlinear estimation using the Levenberg-Marquardt algorithm

Termination criteria

- Using ratio tolerance τ_1, when

$$1 - \frac{\text{cost}_{\text{current}}}{\text{cost}_{\text{previous}}} \leq \tau_1$$

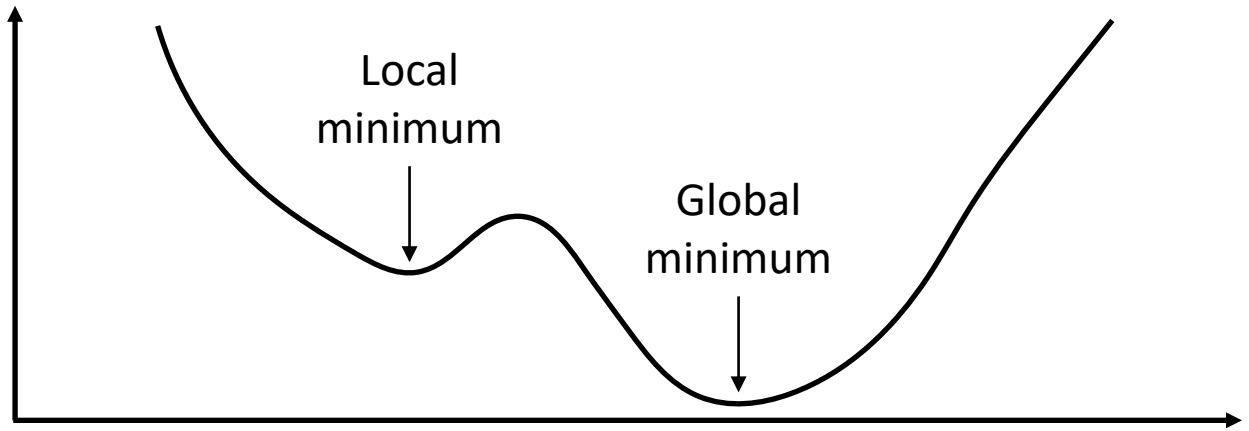
- Using difference tolerance τ_2, when

$$\text{cost}_{\text{previous}} - \text{cost}_{\text{current}} \leq \tau_2$$

- When the number of function evaluations (i.e., inner loop) has reached $100(\text{number of parameters} + 1)$

Parameter vector

- May get stuck in local minimum of high dimensional space
- **Use minimal parameterizations towards mitigating this**



Parameterization of a homogeneous vector

Let the homogeneous vector $\bar{\mathbf{v}} = (a, \mathbf{b}^\top)^\top \in \mathbb{R}^n$, where $\|\bar{\mathbf{v}}\| = 1$ (i.e., $\bar{\mathbf{v}}$ is a unit vector), be parameterized as

$$\mathbf{v} = \frac{2}{\operatorname{sinc}(\cos^{-1}(a))}\mathbf{b} \in \mathbb{R}^{n-1}$$

then, if $\|\mathbf{v}\| > \pi$, normalized by

$$\mathbf{v} = \left(1 - \frac{2\pi}{\|\mathbf{v}\|}\left\lceil \frac{\|\mathbf{v}\| - \pi}{2\pi} \right\rceil\right)\mathbf{v}$$

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix

Parameterization of a homogeneous vector

The parameterized homogeneous vector $\mathbf{v}$ is deparameterized as the homogeneous vector

$$\bar{\mathbf{v}} = \left(\cos\left(\frac{\|\mathbf{v}\|}{2}\right), \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathbf{v}^\top \right)^\top \in \mathbb{R}^n$$

Unit vector with nonnegative first element

$$\bar{\mathbf{v}} = \left(a, \mathbf{b}^\top\right)^\top, \text{ where } a = \cos\left(\frac{\|\mathbf{v}\|}{2}\right) \text{ and } \mathbf{b} = \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathbf{v}$$

where $\|\bar{\mathbf{v}}\| = 1$ and a is nonnegative. For the deparameterization,

$$\frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{v}} = \frac{\partial (a, \mathbf{b}^\top)}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\mathrm{d}a}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{b}}{\partial \mathbf{v}} \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$$

where

$$\frac{\mathrm{d}a}{\partial \mathbf{v}} = \begin{cases} \mathbf{0}^\top & \text{if } \|\mathbf{v}\| = 0 \\ -\frac{1}{2}\mathbf{b}^\top & \text{otherwise} \end{cases}$$

and

$$\frac{\partial \mathbf{b}}{\partial \mathbf{v}} = \begin{cases} \frac{1}{2}\mathtt{I} & \text{if } \|\mathbf{v}\| = 0 \\ \frac{\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{2}\mathtt{I} + \frac{1}{4\|\mathbf{v}\|}\frac{\mathrm{d}\operatorname{sinc}\left(\frac{\|\mathbf{v}\|}{2}\right)}{\mathrm{d}\frac{\|\mathbf{v}\|}{2}}\mathbf{v}\mathbf{v}^\top & \text{otherwise} \end{cases}$$

Parameterization of a homogeneous vector

- Sinc function

The sinc function

$$\operatorname{sinc}(x) = \begin{cases} 1 & \text{if } x = 0 \\ \frac{\sin(x)}{x} & \text{otherwise} \end{cases}$$

The derivative is given by

$$\frac{\mathrm{d}\operatorname{sinc}(x)}{\mathrm{d}x} = \begin{cases} 0 & \text{if } x = 0 \\ \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2} & \text{otherwise} \end{cases}$$

Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

Parameter vector

$$\underset{11 \text{ x } 1}{\hat{\mathbf{p}}}$$

Parameterization of homogeneous vector

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix

Mapping

$$\hat{\mathbf{x}}_i = \hat{\mathrm{P}}\mathbf{X}_i \,\forall\, i$$

Measurement vector

$$\underset{2n \text{ x } 1}{\left(\tilde{\mathbf{x}}_1^\top, \tilde{\mathbf{x}}_2^\top, \ldots, \tilde{\mathbf{x}}_n^\top\right)^\top} \text{ with associated covariance matrices } \Sigma_{\tilde{\mathbf{x}}_1}, \Sigma_{\tilde{\mathbf{x}}_2}, \ldots, \Sigma_{\tilde{\mathbf{x}}_n}$$

Cost

$$\sum_i \boldsymbol{\epsilon}_i^\top \Sigma_{\tilde{\mathbf{x}}_i}^{-1} \boldsymbol{\epsilon}_i, \text{ where } \boldsymbol{\epsilon}_i = \tilde{\mathbf{x}}_i - \hat{\tilde{\mathbf{x}}}_i$$

Jacobian

$$\mathrm{J} = \underset{2n \text{ x } 11}{\begin{bmatrix} \mathrm{A}_1 \\ \mathrm{A}_2 \\ \vdots \\ \mathrm{A}_n \end{bmatrix}}, \text{ where } \mathrm{A}_i = \frac{\partial \hat{\tilde{\mathbf{x}}}_i}{\partial \hat{\mathbf{p}}} = \underset{2 \text{ x } 12}{\frac{\partial \hat{\tilde{\mathbf{x}}}_i}{\partial \bar{\mathbf{p}}}} \underset{12 \text{ x } 11}{\frac{\partial \bar{\mathbf{p}}}{\partial \hat{\mathbf{p}}}}$$

But, use data normalization

Projection of a point under the camera projection matrix

The homogeneous 3D point $\mathbf{X}$ is projected to the homogeneous 2D point $\mathbf{x}$ under the (homogeneous) camera projection matrix $\mathtt{P}$ by

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$

Dehomogenizing the 2D point results in the mapping $\mathbf{X} \mapsto \tilde{\mathbf{x}}$. For this mapping

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}} = \frac{1}{w}\begin{bmatrix} \mathbf{X}^\top & \mathbf{0}^\top & -\tilde{x}\mathbf{X}^\top \\ \mathbf{0}^\top & \mathbf{X}^\top & -\tilde{y}\mathbf{X}^\top \end{bmatrix} \quad 2 \text{ x } 12$$

and

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{X}} = \frac{1}{w}\begin{bmatrix} \mathbf{p}^{1\top} - \tilde{x}\mathbf{p}^{3\top} \\ \mathbf{p}^{2\top} - \tilde{y}\mathbf{p}^{3\top} \end{bmatrix} \quad \text{(Used later in quarter)}$$

where $w = \mathbf{p}^{3\top}\mathbf{X}$ and $\mathbf{p}^{i\top}$ is the ith row of $\mathtt{P}$.

Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

Normal equations matrix

$$\underset{11 \times 11}{\mathtt{U}} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \mathtt{J}, \text{ where } \Sigma_{\mathbf{X}}^{-1} = \overset{2n \times 2n}{\begin{bmatrix} \Sigma_{\tilde{\mathbf{x}}_1}^{-1} & & & \\ & \Sigma_{\tilde{\mathbf{x}}_2}^{-1} & & \\ & & \ddots & \\ & & & \Sigma_{\tilde{\mathbf{x}}_n}^{-1} \end{bmatrix}}$$

Normal equations vector

$$\underset{11 \times 1}{\boldsymbol{\epsilon}_\mathbf{a}} = \mathtt{J}^\top \Sigma_{\mathbf{X}}^{-1} \underset{2n \times 1}{\boldsymbol{\epsilon}}, \text{ where } \boldsymbol{\epsilon} = \mathbf{X} - \hat{\mathbf{X}}$$

where $\mathbf{X} = \left(\tilde{\mathbf{x}}_1^\top, \tilde{\mathbf{x}}_2^\top, \ldots, \tilde{\mathbf{x}}_n^\top\right)^\top$ and $\hat{\mathbf{X}} = \left(\hat{\tilde{\mathbf{x}}}_1^\top, \hat{\tilde{\mathbf{x}}}_2^\top, \ldots, \hat{\tilde{\mathbf{x}}}_n^\top\right)^\top$

Augmented normal equations

$$\mathtt{S} = \mathtt{U}^* = \mathtt{U} + \lambda \mathtt{I} \qquad 11 \times 11$$

$$\mathbf{e} = \boldsymbol{\epsilon}_\mathbf{a} \qquad 11 \times 1$$

$$\mathtt{S}\boldsymbol{\delta}_\mathbf{a} = \mathbf{e}, \text{ solve for } \boldsymbol{\delta}_\mathbf{a} \qquad 11 \times 1$$

Candidate parameter vector

$$\hat{\mathbf{p}}_0 = \hat{\mathbf{p}} + \boldsymbol{\delta}_\mathbf{a}$$

Adjustment

Normalize parameterization of a homogeneous vector after adjustment

Data normalization with covariance propagation

Determine the similarity transformation $\mathtt{T}$ such that the mean (i.e., centroid) of the transformed points is at the origin and their standard deviation from the origin is $\sqrt{2}$.

$$\mathtt{T} = \begin{bmatrix} s & 0 & -s\mu_{\tilde{x}} \\ 0 & s & -s\mu_{\tilde{y}} \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } s = \sqrt{\frac{2}{\sigma_{\tilde{x}}^2 + \sigma_{\tilde{y}}^2}}$$

where $\mu_{\tilde{x}}$ and $\sigma_{\tilde{x}}^2$, and $\mu_{\tilde{y}}$ and $\sigma_{\tilde{y}}^2$ are the mean and variance of the $\tilde{x}$ and $\tilde{y}$ coordinates, respectively.

$$\mathbf{x}_{\text{DN}} = \mathtt{T}\mathbf{x}$$

$$\begin{bmatrix} \tilde{\mathbf{x}}_{\text{DN}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ 1 \end{bmatrix}, \text{ where } \mathtt{A} = s\mathtt{I} \text{ and } \mathbf{t} = \begin{bmatrix} -s\mu_{\tilde{x}} \\ -s\mu_{\tilde{y}} \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{x}}_{\text{DN}} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{A}\tilde{\mathbf{x}} + \mathbf{t} \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{x}}_{\text{DN}} = \mathtt{A}\tilde{\mathbf{x}} + \mathbf{t}$$

If unknown, then assume to be identity

Covariance propagation

$$\Sigma_{\tilde{\mathbf{x}}_{\text{DN}}} = \mathtt{J}\Sigma_{\tilde{\mathbf{x}}}\mathtt{J}^\top, \text{ where } \mathtt{J} = \frac{\partial \tilde{\mathbf{x}}_{\text{DN}}}{\partial \tilde{\mathbf{x}}} = \mathtt{A} = s\mathtt{I}$$

Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

- Algorithm

1. Data normalize the points

$$\tilde{\mathbf{x}}_{\mathrm{DN},i} = \mathtt{A}\tilde{\mathbf{x}}_i + \mathbf{t} \text{ and associated covariance matrices } \Sigma_{\tilde{\mathbf{x}}_{\mathrm{DN},i}} = s^2\Sigma_{\tilde{\mathbf{x}}_i}$$
$$\mathbf{X}_{\mathrm{DN},i} = \mathtt{U}\mathbf{X}_i \,\forall\, i$$

2. Estimate the data normalized camera projection matrix $\mathtt{P}_{\mathrm{DN}}$ from the data normalized points

3. Data denormalize the data normalized camera projection matrix

$$\begin{aligned}
\mathbf{x}_{\mathrm{DN},i} &= \mathtt{P}_{\mathrm{DN}}\mathbf{X}_{\mathrm{DN},i} \,\forall\, i \\
\mathtt{T}\mathbf{x}_i &= \mathtt{P}_{\mathrm{DN}}\mathtt{U}\mathbf{X}_i \\
\mathbf{x}_i &= \mathtt{T}^{-1}\mathtt{P}_{\mathrm{DN}}\mathtt{U}\mathbf{X}_i \\
\mathbf{x}_i &= \mathtt{P}\mathbf{X}_i \,\forall\, i, \text{ where } \mathtt{P} = \mathtt{T}^{-1}\mathtt{P}_{\mathrm{DN}}\mathtt{U}
\end{aligned}$$

Decomposition of camera projection matrix

If estimated from points in a metric coordinate frame
Solve for camera center $\mathbf{C}$

$$\mathtt{P}\mathbf{C} = \mathbf{0} \text{ (i.e., } \mathbf{C} \text{ is null space of } \mathtt{P})$$

Solve for camera calibration matrix $\mathtt{K}$ and camera rotation matrix $\mathtt{R}$

$$\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}] = [\mathtt{KR} \,|\, \mathtt{K}\mathbf{t}] = [\mathtt{M} \,|\, \mathtt{K}\mathbf{t}], \text{ where } \mathtt{M} = \mathtt{KR}$$

1. RQ decomposition of $\mathtt{M}$. $\mathtt{M} = \mathtt{KR}$, where $\mathtt{K}$ is upper triangular and $\mathtt{R}$ is orthogonal.
2. Scale $\mathtt{K}$ such that $k_{33} = 1$. $\mathtt{K} = \frac{1}{k_{33}}\mathtt{K}$.
3. Ensure k_{11} and k_{22} are positive

$$\mathtt{M} = \mathtt{KR} = \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1\top} \\ \mathbf{r}^{2\top} \\ \mathbf{r}^{3\top} \end{bmatrix}$$

If $k_{11} < 0$, then negate $\mathbf{k}_1$ and $\mathbf{r}^{1\top}$
If $k_{22} < 0$, then negate $\mathbf{k}_2$ and $\mathbf{r}^{2\top}$

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Ensure $\mathtt{R}$ is a special orthogonal. If $\det(\mathtt{R}) = -1$, then negate $\mathtt{R}$.

Next lecture

- Estimation of camera pose (calibrated camera), linear
- Reading
  - Lepetit, Moreno-Noguer, and Fua, “EPnP: An Accurate O(n) Solution to the PnP Problem”