香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY
電子及計算機工程學系
DEPARTMENT OF ELECTRONIC & COMPUTER ENGINEERING

Introduction to Aerial Robotics
Lecture 6

Shaojie Shen
Associate Professor
Dept. of ECE, HKUST

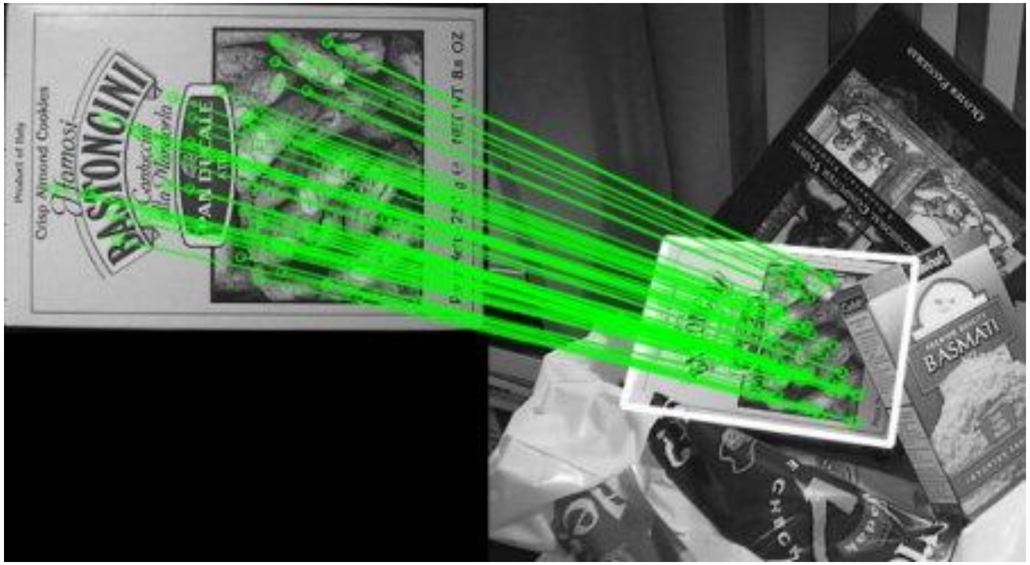
16 March 2021

Outline

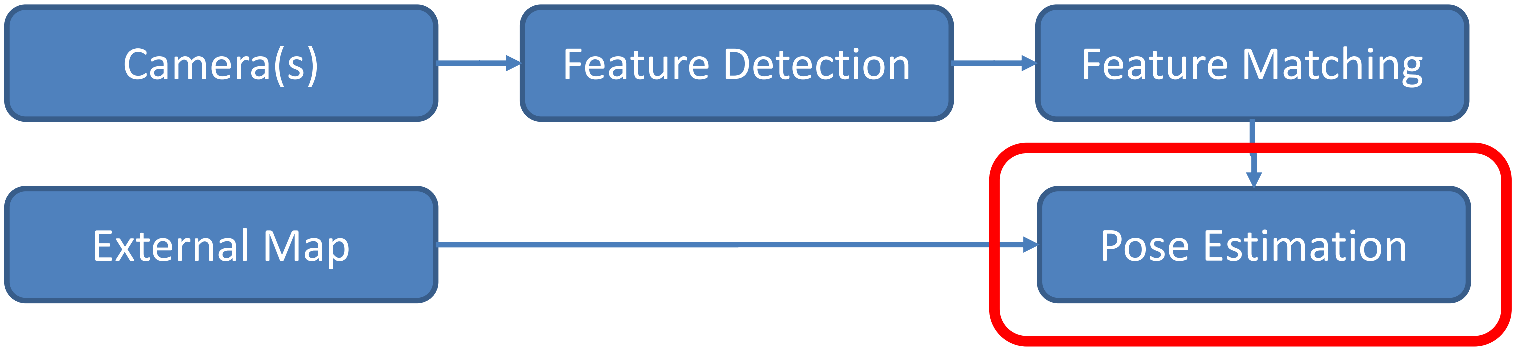
- 3D-3D Pose Estimation
- 3D-2D Pose Estimation
- Outlier Rejection and Robust Estimation

Review: Feature Detection and Matching

- Detect corners features in both images
- Use image patch as feature description
  - Could be extended to color, texture, SIFT/HOG descriptor
- Find correspondences using descriptor matching



Vision-based Pose Estimation Pipeline (aka. Map-based Localization)

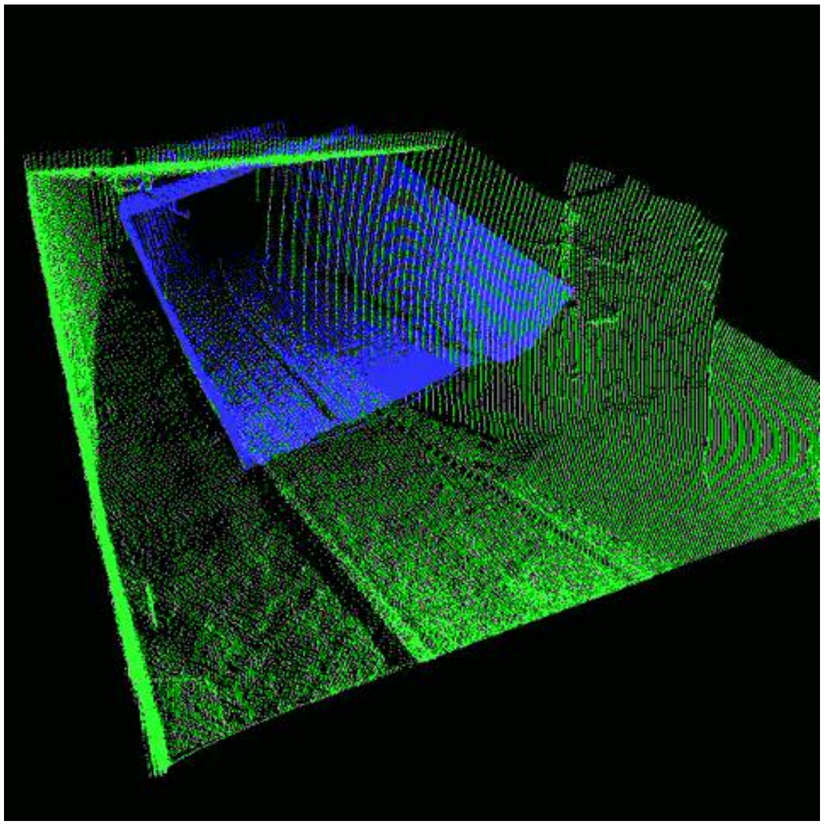


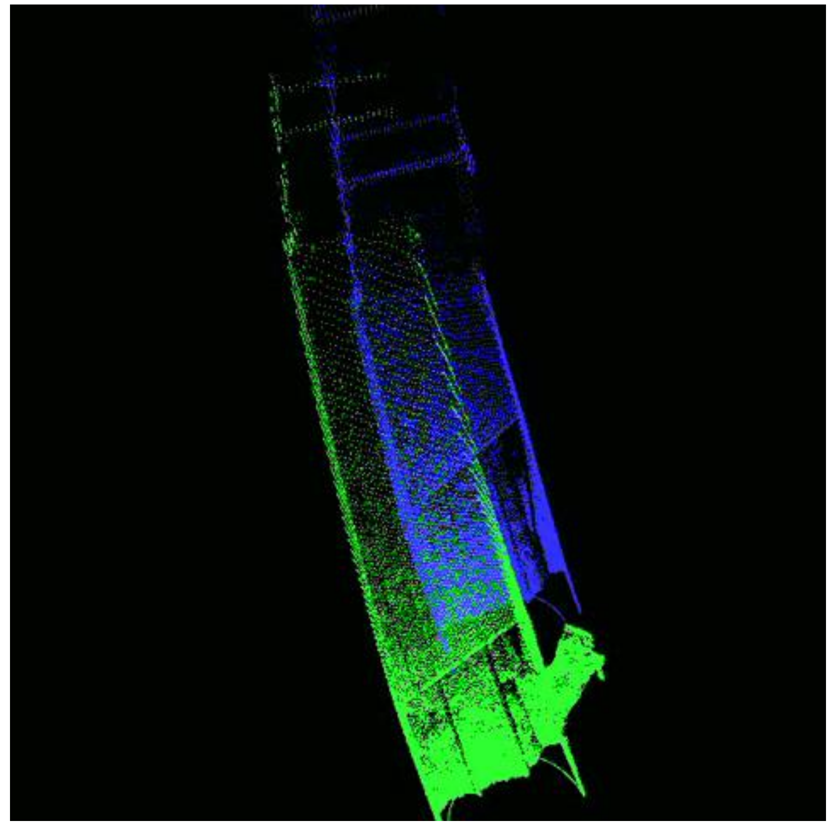
Vision-based Pose Estimation Pipeline (aka. Map-based Localization)

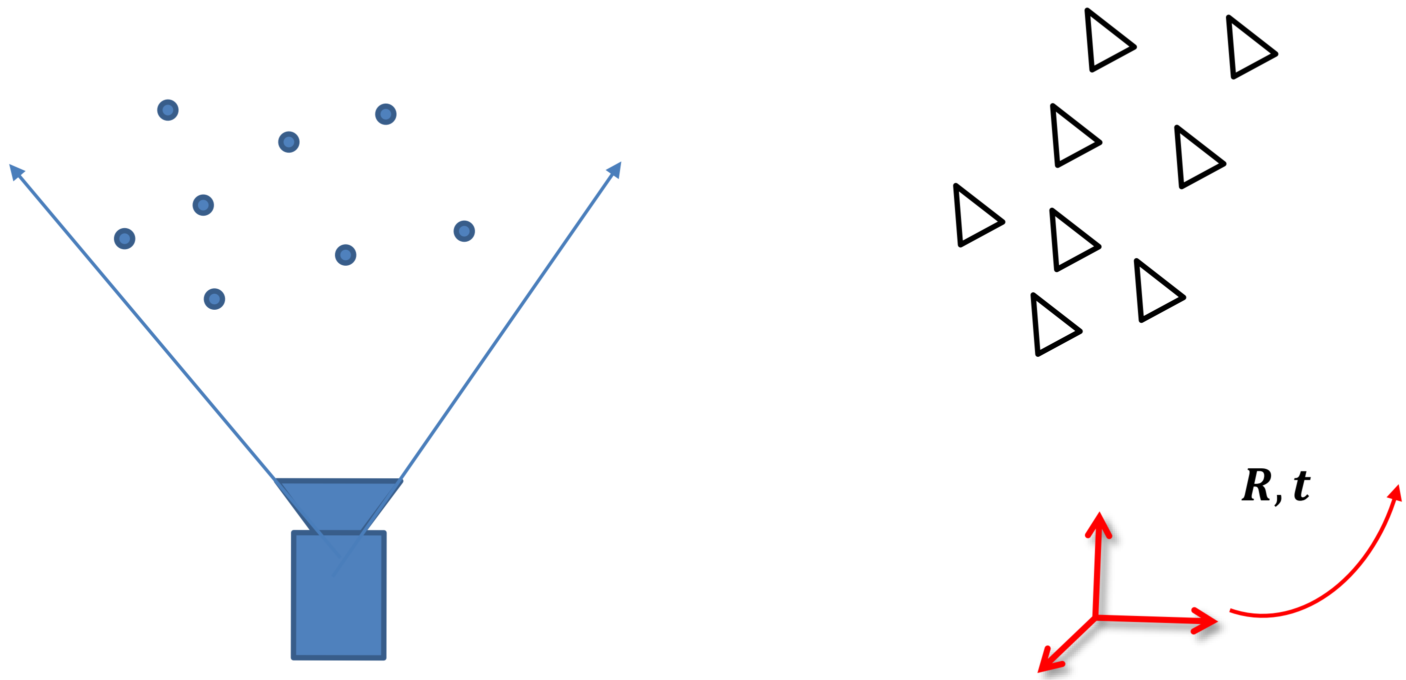
- Setup
  - Applicable to different camera configurations (monocular, stereo, RGB-D, etc.)
  - Sufficient illumination and texture
  - Dominance of static scene
  - Known map
  - Sufficient observation of map features
  - Focus on global consistency

3D-3D Pose Estimation (Point Cloud Registration)

3D-3D Registration of Point Cloud





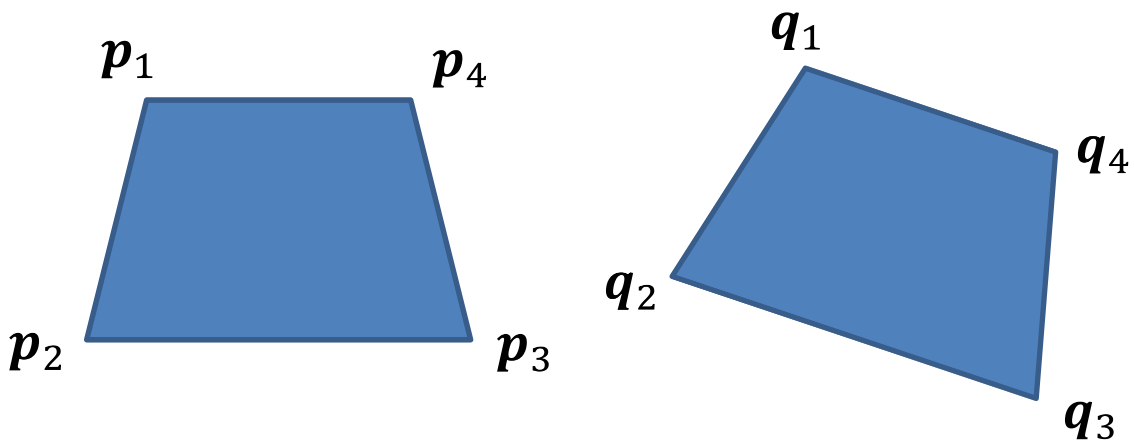


3D points in frame 1
(or known 3D points in world frame)

3D-3D

3D points in frame 2
(or 3D point observations in body frame)

3D-3D Pose Estimation



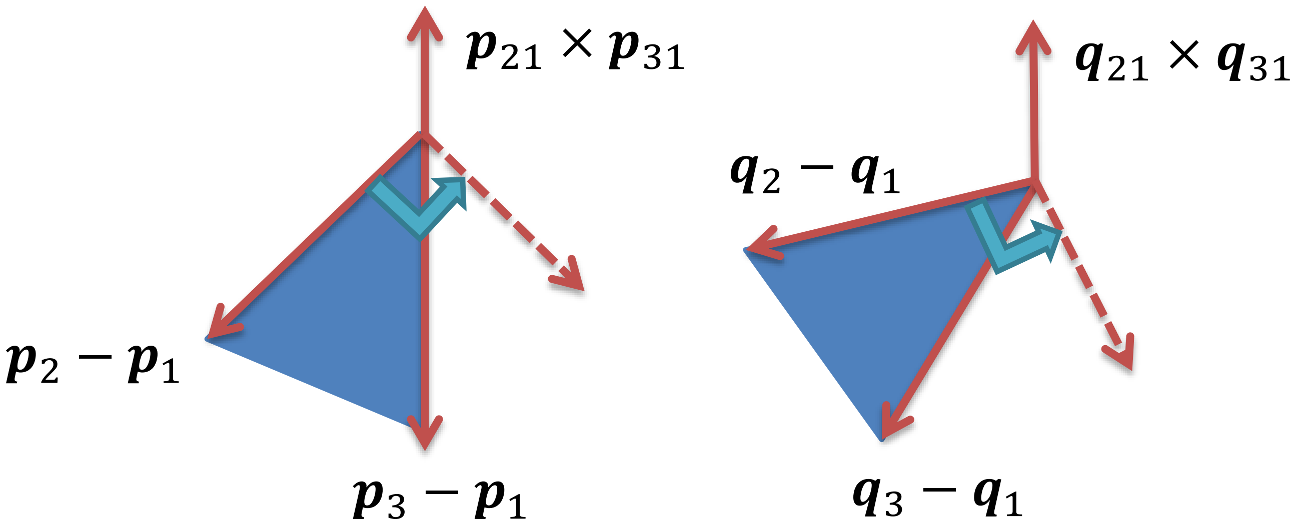
- How do we solve for $\boldsymbol{R}, \boldsymbol{t}$ from point correspondences?

$$\boldsymbol{p}_i = \boldsymbol{R}\boldsymbol{q}_i + \boldsymbol{t}$$

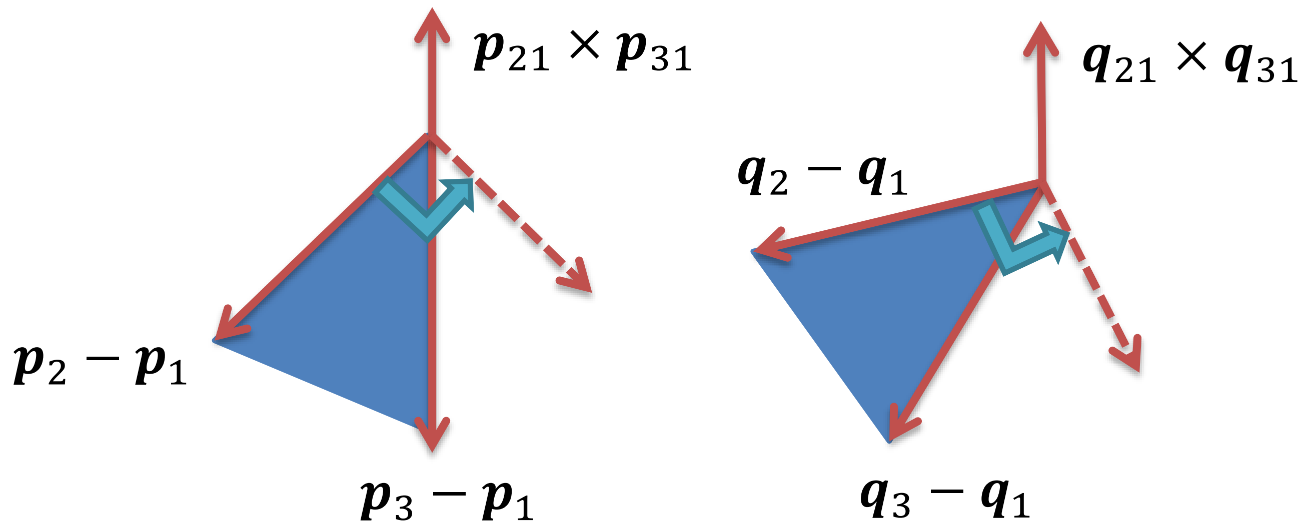
Slide adapted from Kostas Daniilidis

3D-3D Pose Estimation

- What is the minimal number of points needed?



Slide adapted from Kostas Daniilidis



- Three non-collinear points suffice: each triangle $\boldsymbol{p}_{i=1\ldots3}$ and $\boldsymbol{q}_{i=1\ldots3}$ make an orthogonal basis

$$\begin{pmatrix} \boldsymbol{p}_{21} & (\boldsymbol{p}_{21} \times \boldsymbol{p}_{31}) \times \boldsymbol{p}_{21} & \boldsymbol{p}_{21} \times \boldsymbol{p}_{31} \end{pmatrix}$$

and

$$\begin{pmatrix} \boldsymbol{q}_{21} & (\boldsymbol{q}_{21} \times \boldsymbol{q}_{31}) \times \boldsymbol{q}_{21} & \boldsymbol{q}_{21} \times \boldsymbol{q}_{31} \end{pmatrix}$$

- Rotation between two orthogonal bases is unique.

Slide adapted from Kostas Daniilidis

- We solve a minimization problem for N >= 3 point correspondences:

$$\min_{\boldsymbol{R},\boldsymbol{t}} \sum_{i}^{N} \|\boldsymbol{p}_i - (\boldsymbol{R}\boldsymbol{q}_i + \boldsymbol{t})\|^2$$

- After differentiating with respect to $\boldsymbol{t}$, we observe that the translation is the difference between the centroids:

$$\mathbf{t} = \frac{1}{N}\sum_{i}^{N} \boldsymbol{p}_i - \boldsymbol{R}\frac{1}{N}\sum_{i}^{N} \boldsymbol{q}_i = \overline{\boldsymbol{p}} - \boldsymbol{R}\overline{\boldsymbol{q}}$$

Slide adapted from Kostas Daniilidis

- We subtract the centroids $\overline{\boldsymbol{p}}$ and $\overline{\boldsymbol{q}}$ and rewrite the objective function as

$$\min_{R} \|\boldsymbol{P} - \boldsymbol{R}\boldsymbol{Q}\|_F^2$$

where

$$\boldsymbol{Q} = [\boldsymbol{p}_1 - \overline{\boldsymbol{p}}, \dots, \boldsymbol{p}_N - \overline{\boldsymbol{p}}]$$

and

$$\boldsymbol{P} = [\boldsymbol{q}_1 - \overline{\boldsymbol{q}}, \dots, \boldsymbol{q}_N - \overline{\boldsymbol{q}}]$$

- Some useful mathematics
  - Frobenius norm $\|\boldsymbol{A}\|_F = \sqrt{\sum\sum |a_{ij}|^2} \rightarrow \|\boldsymbol{A}\|_F = \sqrt{tr(\boldsymbol{A}\boldsymbol{A}^H)}$
  - $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = tr(\boldsymbol{B}\boldsymbol{A})$
  - $\mathrm{tr}(\boldsymbol{A}) = tr(\boldsymbol{A}^T)$
  - $\mathrm{tr}(\boldsymbol{A} + \boldsymbol{B}) = tr(\boldsymbol{A}) + tr(\boldsymbol{B})$

Slide adapted from Kostas Daniilidis

- We rewrite the Frobenius norm using the trace of the matrix
$$\|\boldsymbol{P}-\boldsymbol{R}\boldsymbol{Q}\|_F^2 = tr(\boldsymbol{P}^T\boldsymbol{P}) + tr(\boldsymbol{Q}^T\boldsymbol{Q}) - tr(\boldsymbol{P}^T\boldsymbol{R}\boldsymbol{Q}) - tr\left(\boldsymbol{Q}^T\boldsymbol{R}^T\boldsymbol{P}\right)$$

- And observe that only the two last terms depend on the unknown $\boldsymbol{R}$ yielding a maximization problem.

- Even without using the properties of the trace we can see that both last terms are equal to
$$\sum_i^N \boldsymbol{R}(\boldsymbol{q}_i - \overline{\boldsymbol{q}})(\boldsymbol{p}_i - \overline{\boldsymbol{p}})^T = tr(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{P}^T)$$

- The 3D-3D pose problem reduced to
$$\max_{\boldsymbol{R}} \; tr(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{P}^T)$$

Slide adapted from Kostas Daniilidis

- If the SVD of $\boldsymbol{QP}^T$ is $\boldsymbol{USV}^T$ and let $\boldsymbol{Z} = \boldsymbol{V}^T\boldsymbol{RU}$

$$tr\left(\boldsymbol{RQP}^T\right) = tr\left(\boldsymbol{RUSV}^T\right) = tr(\boldsymbol{ZS}) = \sum_1^3 z_{ii}\sigma_i \leq \sum_1^3 \sigma_i$$

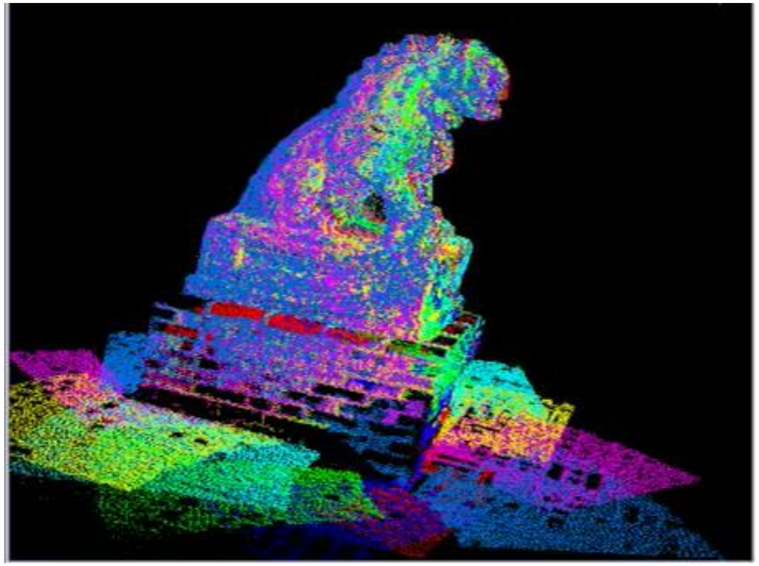
- The upper bound is obtained by setting

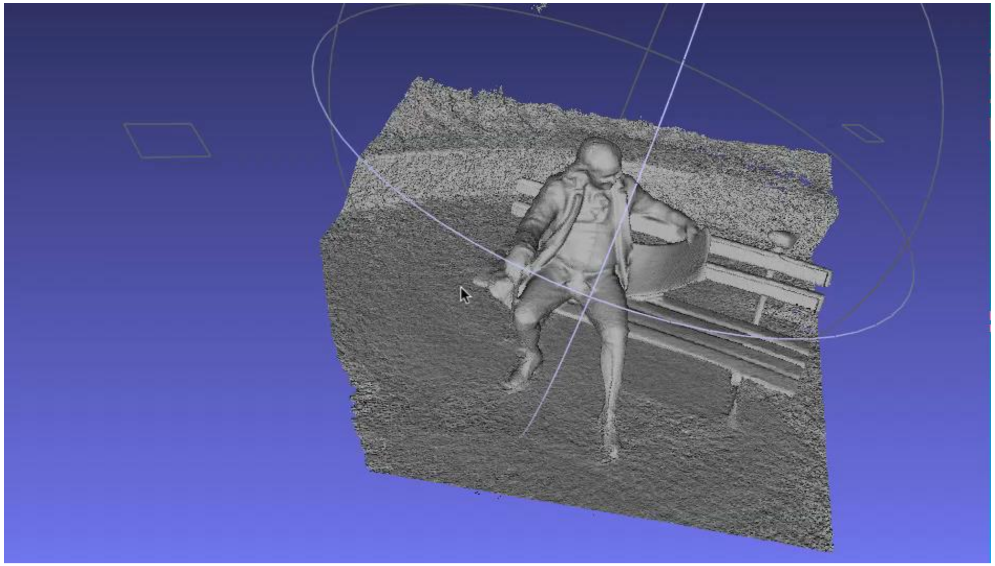
$$\boldsymbol{R} = \boldsymbol{VU}^T$$

Slide adapted from Kostas Daniilidis

3D-3D Pose Estimation

- 3D-3D Registration enables the creation of 3D models from multiple point clouds:





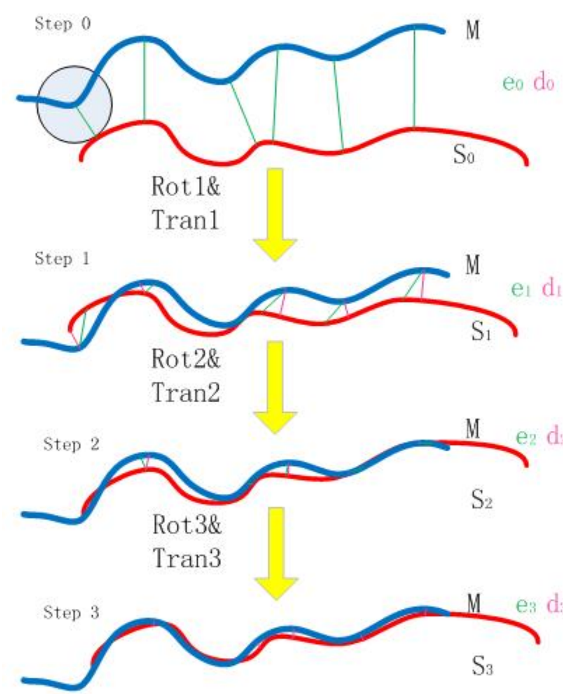
Slide adapted from Kostas Daniilidis

3D-3D Pose Estimation

- How to obtain 3D-3D data association?
  - "Soft" data association directly from point clouds
  - The Iterative Closest Point (ICP) algorithm
    1. Start with some initial guess of rotation and translation
    2. For each point in pointcloud1, find its nearest neighbor in pointcloud2 based on the current estimated rotation and translation
    3. Refine the rotation and translation based on the latest data association
    4. Iterate from step 2 until converge

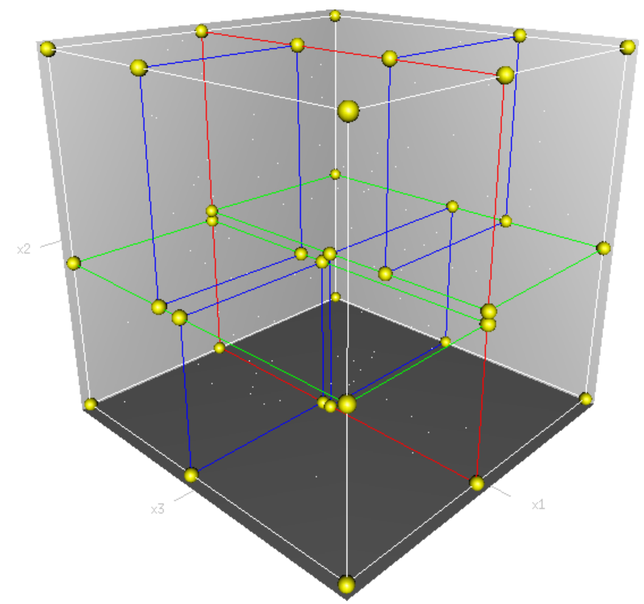
???

$$\min_{\boldsymbol{R},\boldsymbol{t}} \sum_{i}^{N} \|\boldsymbol{p}_i - (\boldsymbol{R}\boldsymbol{q}_i + \boldsymbol{t})\|^2$$

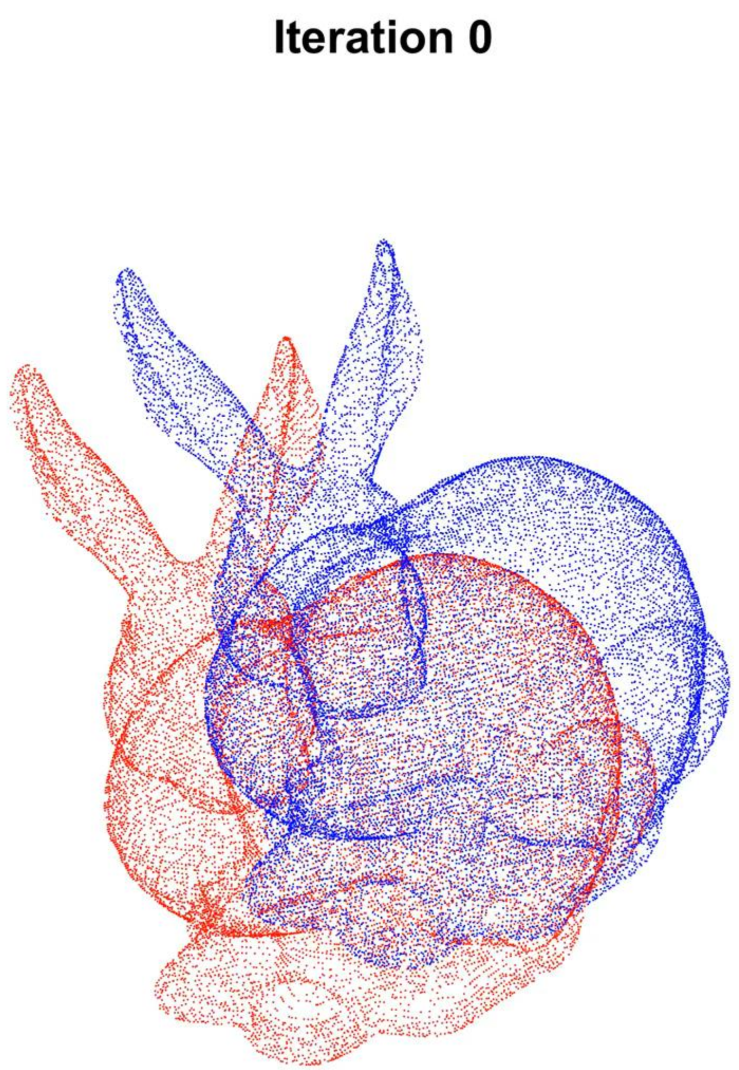


3D-3D Pose Estimation

- How to obtain 3D-3D data association?
  - "Soft" data association directly from point clouds
  - The Iterative Closest Point (ICP) algorithm
    - Need to speed up the search of nearest neighbors
    - Naive implementation: $O(N)$
    - K-d Tree: $O(logN)$



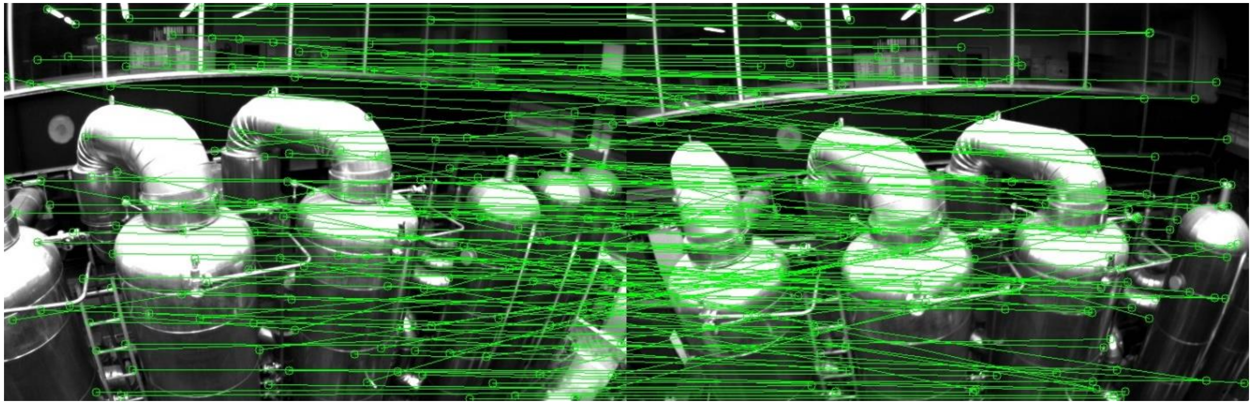
Iterative Closest Point Algorithm



3D Laser-Based Mapping

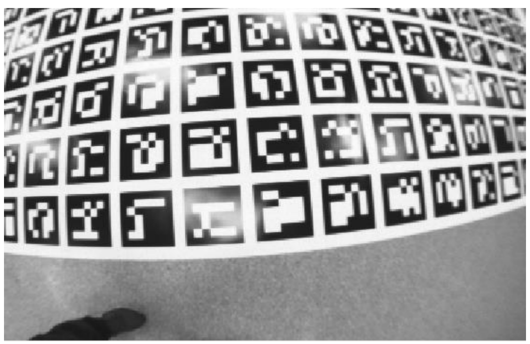
3D-3D Pose Estimation

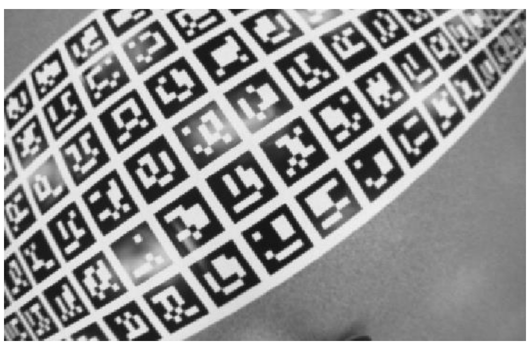
- How to obtain 3D-3D data association?
  - Feature matching using stereo images
    - Calibrated stereo image pairs as input – to be covered in the next lecture
    - Spatial matching – feature matching between stereo image pairs, for computation of 3D points
    - Temporal matching – feature matching between images captures at different times, for motion estimation
    - Need to address outlier removal – to be discussed soon
    - Usually poor performance due to increased ranging error at longer distance with stereo vision – use 3D-2D pose estimation instead

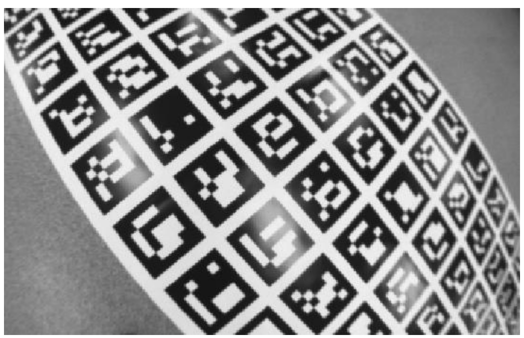


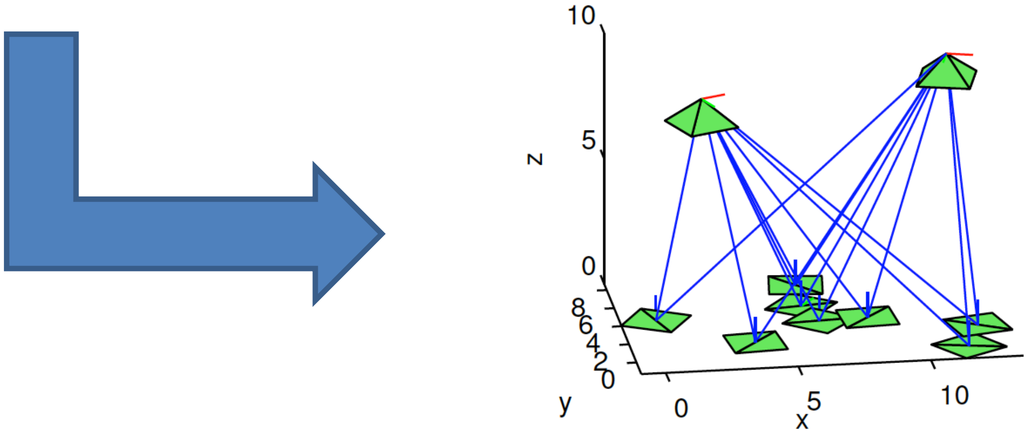
3D-2D Pose Estimation
(Pose from Projective Transform)
(The PnP Problem)

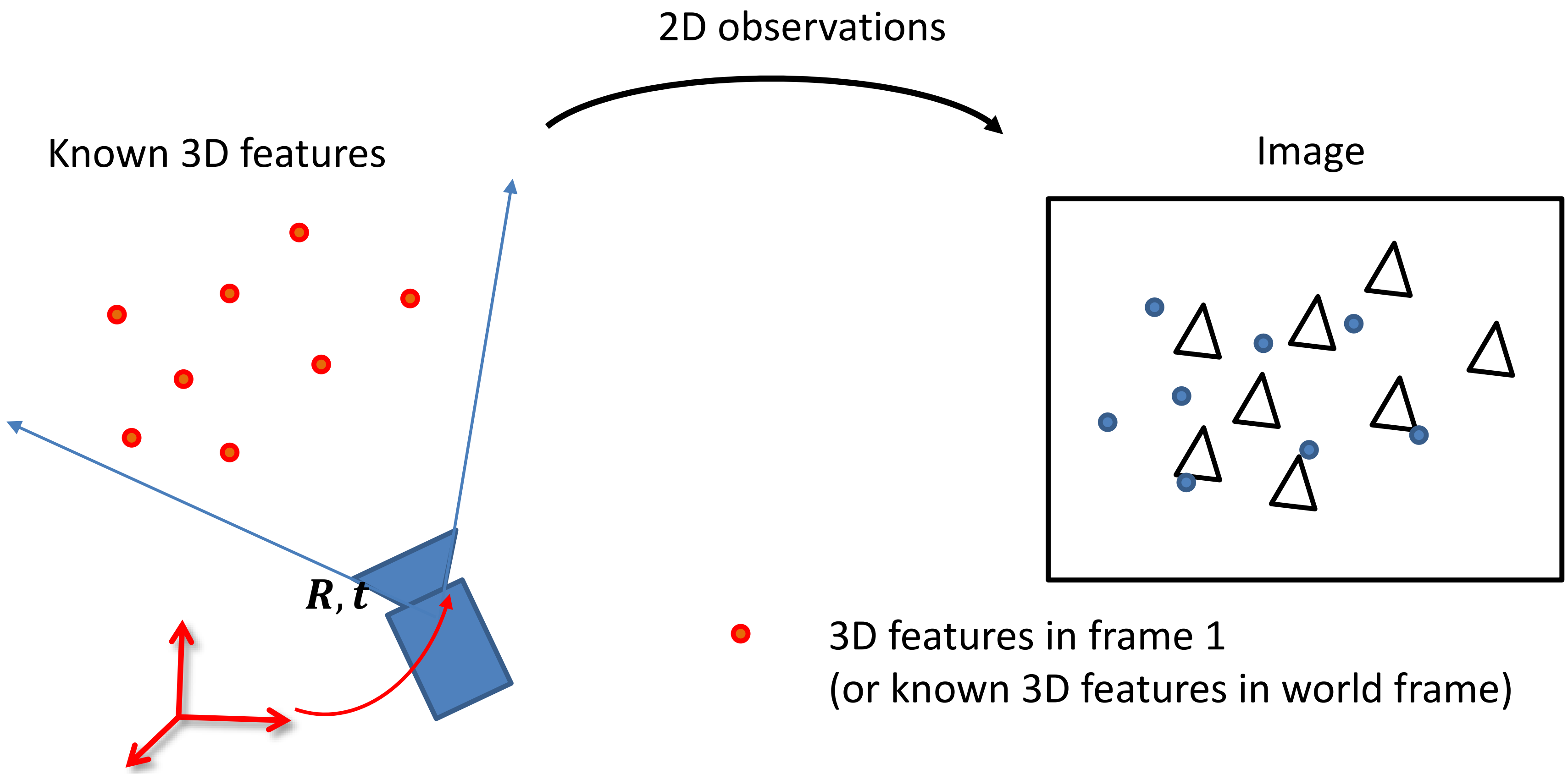
- Project2 phase1
  - Using the projective transformation to get the pose of a robot with respect to a planar pattern:











3D features in frame 1
(or known 3D features in world frame)

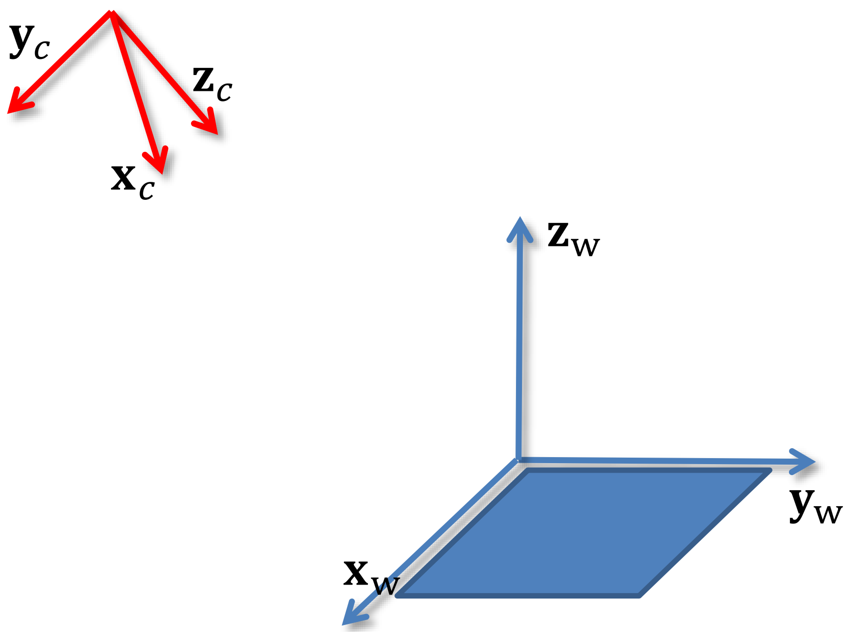
2D feature observations in frame 2
(or feature observations in body frame)

2D feature reprojections of given a estimated pose of frame 2 in frame 1 (or estimated transformation of body frame in world frame)

3D-2D

Linear 3D-2D Pose Estimation on Planar Scene

- Pose from reference points on plane $z_w = 0$



- Recall the projection from world to camera

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{K}(\boldsymbol{r_1} \quad \boldsymbol{r_2} \quad \boldsymbol{r_3} \quad \boldsymbol{t}) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- Assume all points in the world lie in the ground plane $z = 0$.

- Then the transformation reads

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{K}(\boldsymbol{r_1} \quad \boldsymbol{r_2} \quad \boldsymbol{t}) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Slide adapted from Kostas Daniilidis

- Suppose $\boldsymbol{H} = \boldsymbol{K}(\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t})$

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{aligned} \lambda u &= h_{11}x + h_{12}y + h_{13} \\ \lambda v &= h_{21}x + h_{22}y + h_{23} \\ \lambda &= h_{31}x + h_{32}y + h_{33} \end{aligned}$$

$$\begin{aligned} h_{31}xu + h_{32}yu + h_{33}u &= h_{11}x + h_{12}y + h_{13} \\ h_{31}xv + h_{32}yv + h_{33}v &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

Given a observation $(u, v) = (u_i, v_i)$ of feature (x_i, y_i)

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_iu_i & -y_iu_i & -u_i \\ 0 & 0 & 0 & x & y & 1 & -x_iv_i & -y_iv_i & -v_i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Solve this Ax = 0 problem using SVD

Stack pairs of observations

- Suppose we estimate an $\widehat{\boldsymbol{H}}$ from $N \geq 4$ correspondences.
- Let us assume that we know the intrinsic parameters $\boldsymbol{K}$.
- Pose estimation means finding $\boldsymbol{R}\ \boldsymbol{t}$ given $\boldsymbol{H}$ and intrinsic $\boldsymbol{K}$.
- We observe that $\boldsymbol{K}^{-1}\boldsymbol{H} = (\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t})$ has specific properties: its first two columns are orthogonal unit vectors.
- Nothing guarantees that the $\widehat{\boldsymbol{H}}$ we computed will satisfy this condition.

Slide adapted from Kostas Daniilidis

- Let us name the columns of $\boldsymbol{K}^{-1}\widehat{\boldsymbol{H}}$ be $[\widetilde{\boldsymbol{h}}_{\mathbf{1}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{2}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{3}}]$

- since $\boldsymbol{K}^{-1}\boldsymbol{H} = (\boldsymbol{r}_{\mathbf{1}} \quad \boldsymbol{r}_{\mathbf{2}} \quad \boldsymbol{t})$

- We seek orthogonal $\boldsymbol{r}_{\mathbf{1}}$ and $\boldsymbol{r}_{\mathbf{2}}$ that are the closest to $\widetilde{\boldsymbol{h}}_{\mathbf{1}}$ and $\widetilde{\boldsymbol{h}}_{\mathbf{2}}$. The solution to this position is given by the Singular Value Decomposition.

- We find the orthogonal matrix $\boldsymbol{R}$ that is the closest to $(\widetilde{\boldsymbol{h}}_{\mathbf{1}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{2}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{1}} \times \widetilde{\boldsymbol{h}}_{\mathbf{2}})$:
$$\underset{R \in SO(3)}{\arg\min} \|\boldsymbol{R} - (\widetilde{\boldsymbol{h}}_{\mathbf{1}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{2}} \quad \widetilde{\boldsymbol{h}}_{\mathbf{1}} \times \widetilde{\boldsymbol{h}}_{\mathbf{2}})\|_F^2$$

Slide adapted from Kostas Daniilidis

$$\arg\min_{R\in SO(3)} \|\boldsymbol{R} - (\tilde{\boldsymbol{h}}_1 \quad \tilde{\boldsymbol{h}}_2 \quad \tilde{\boldsymbol{h}}_1 \times \tilde{\boldsymbol{h}}_2)\|_F^2$$

- If the SVD of $(\tilde{\boldsymbol{h}}_1 \quad \tilde{\boldsymbol{h}}_2 \quad \tilde{\boldsymbol{h}}_1 \times \tilde{\boldsymbol{h}}_2) = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\boldsymbol{T}}$, then the solution is

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^{\boldsymbol{T}}$$

- To find the translation:

$$\boldsymbol{t} = \tilde{\boldsymbol{h}}_3 / \|\tilde{\boldsymbol{h}}_1\|$$

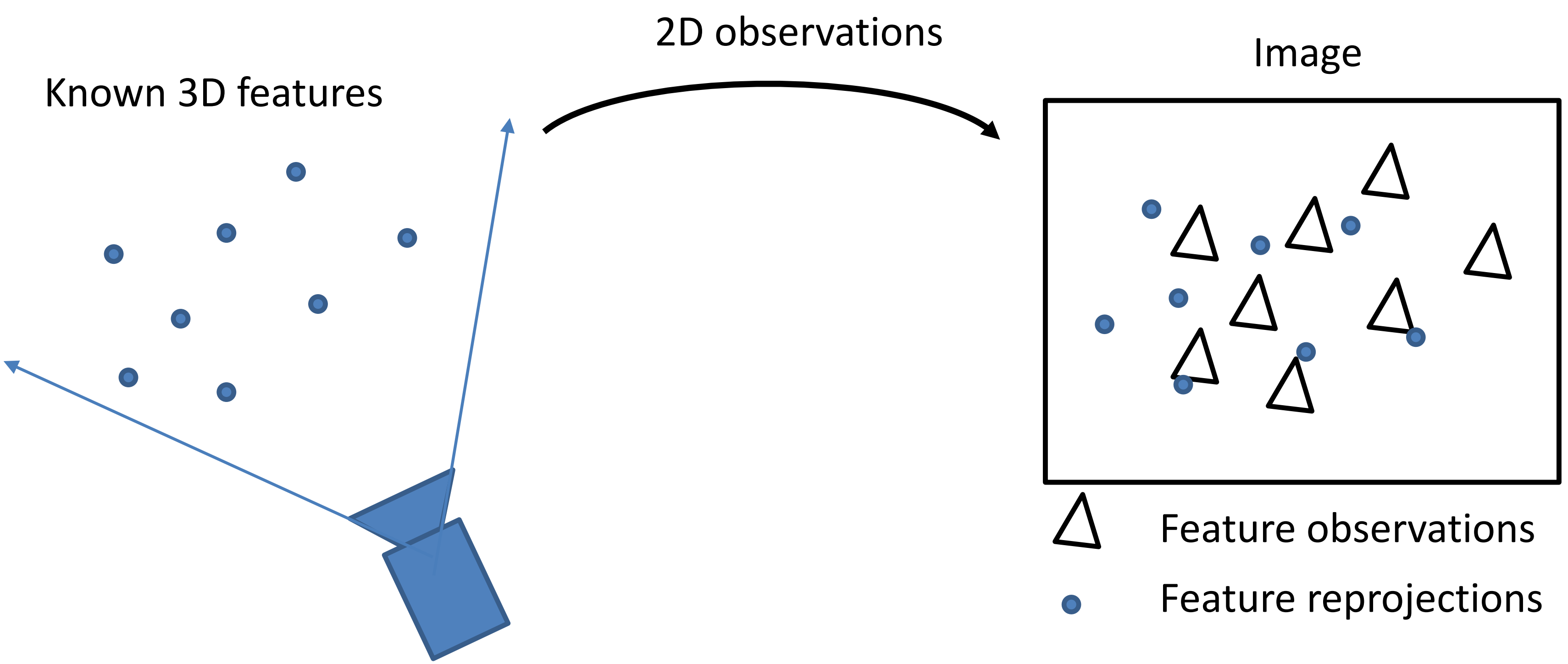
Slide adapted from Kostas Daniilidis

Nonlinear 3D-2D Pose Estimation

$\boldsymbol{\theta}$: $Euler\ Angles\ \epsilon\ R^3$ $\boldsymbol{t}$: $Translation\ \epsilon\ R^3$ $\pi(\cdot)$: $projection\ function$

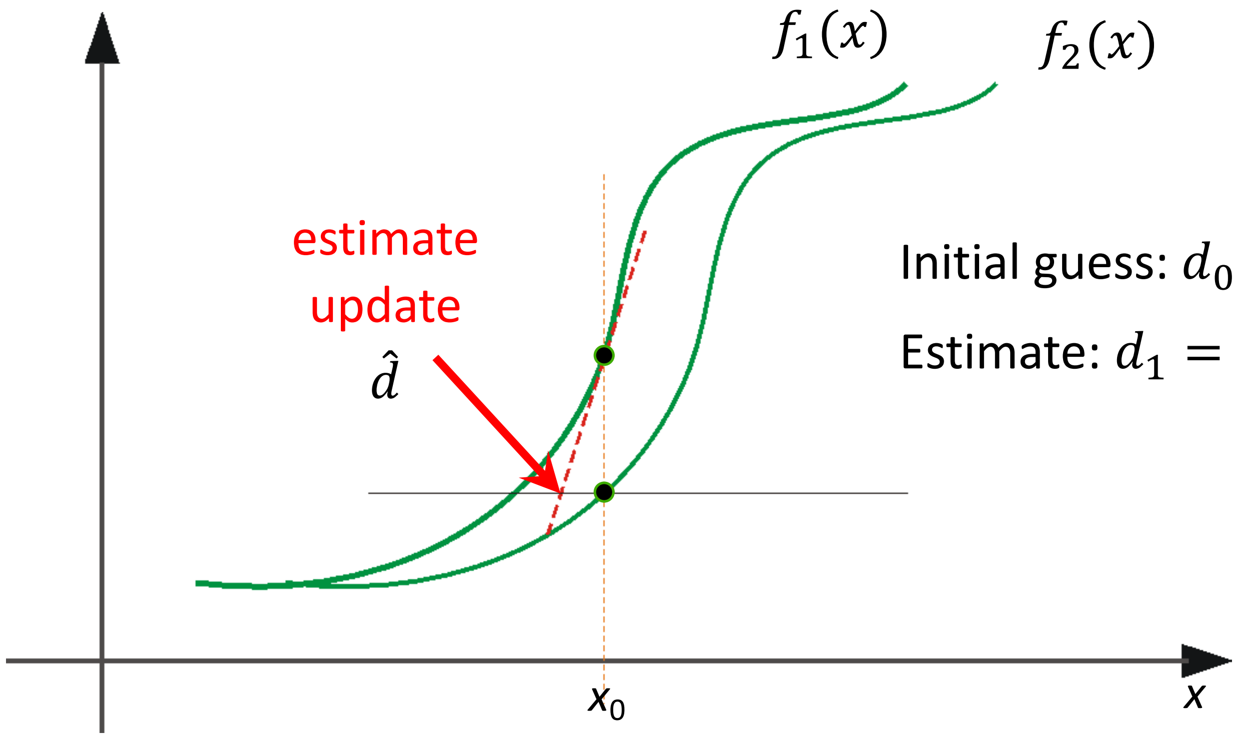
- Minimize the reprojection error w.r.t. camera pose
  - No need to assume planar scene

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K} \cdot \left(\boldsymbol{R}(\boldsymbol{\theta}) \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t} \right) \right) \right\|^2$$



Gauss Newton Method

Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



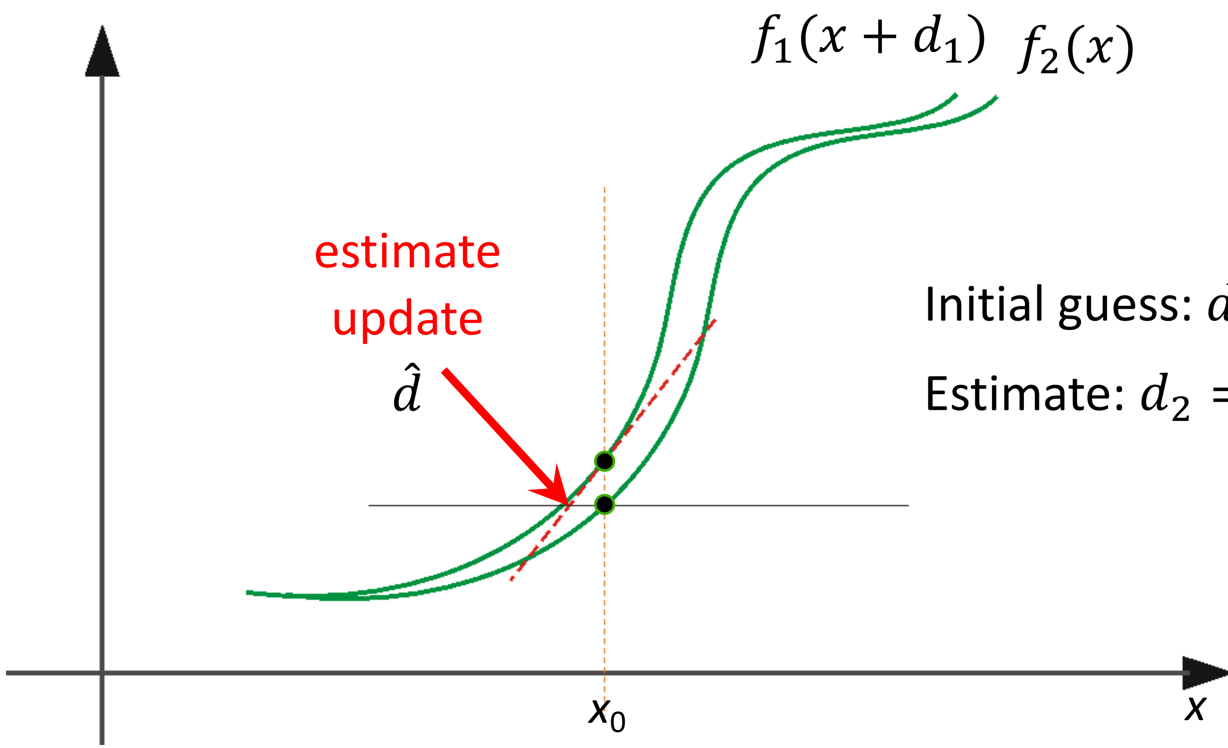
Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

Slide adapted from Richard Szeliski

Gauss Newton Method

Compute d to minimize $\| f_1(x+d) - f_2(x) \|^2$



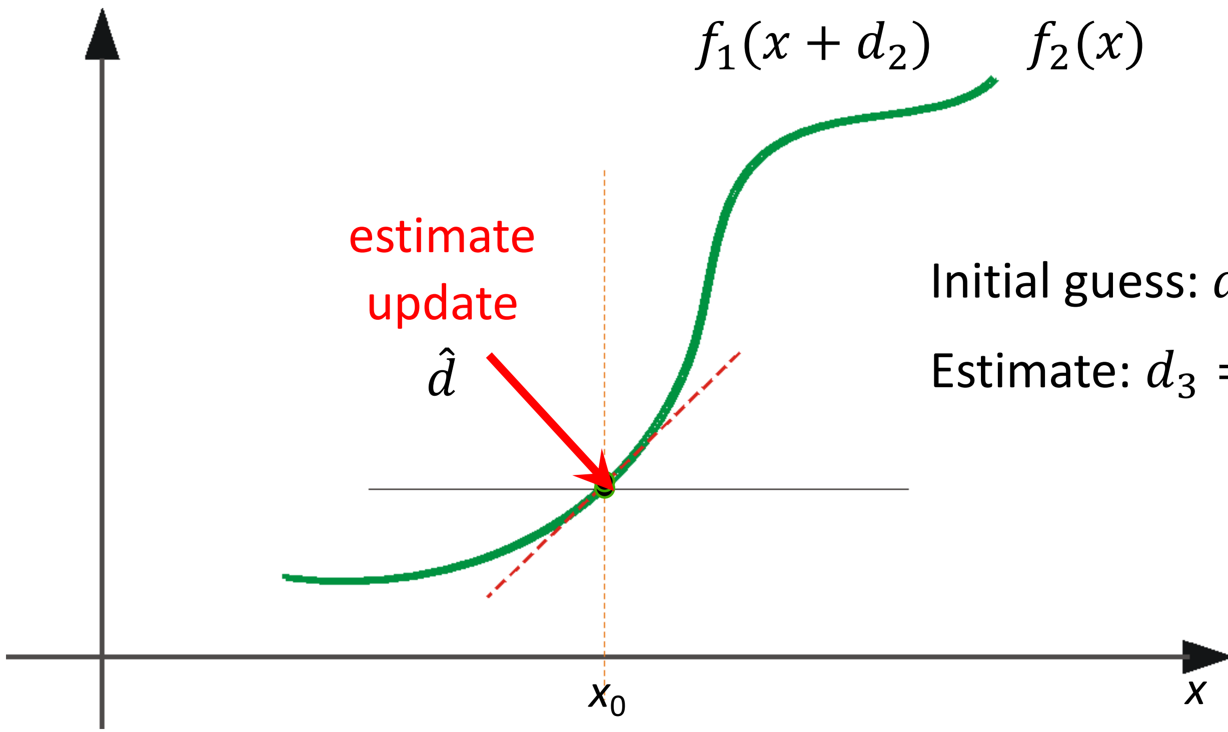
Initial guess: d_1

Estimate: $d_2 = d_1 + \hat{d}$

Slide adapted from Richard Szeliski

Gauss Newton Method

Compute d to minimize $\|f_1(x+d) - f_2(x)\|^2$



Initial guess: d_2

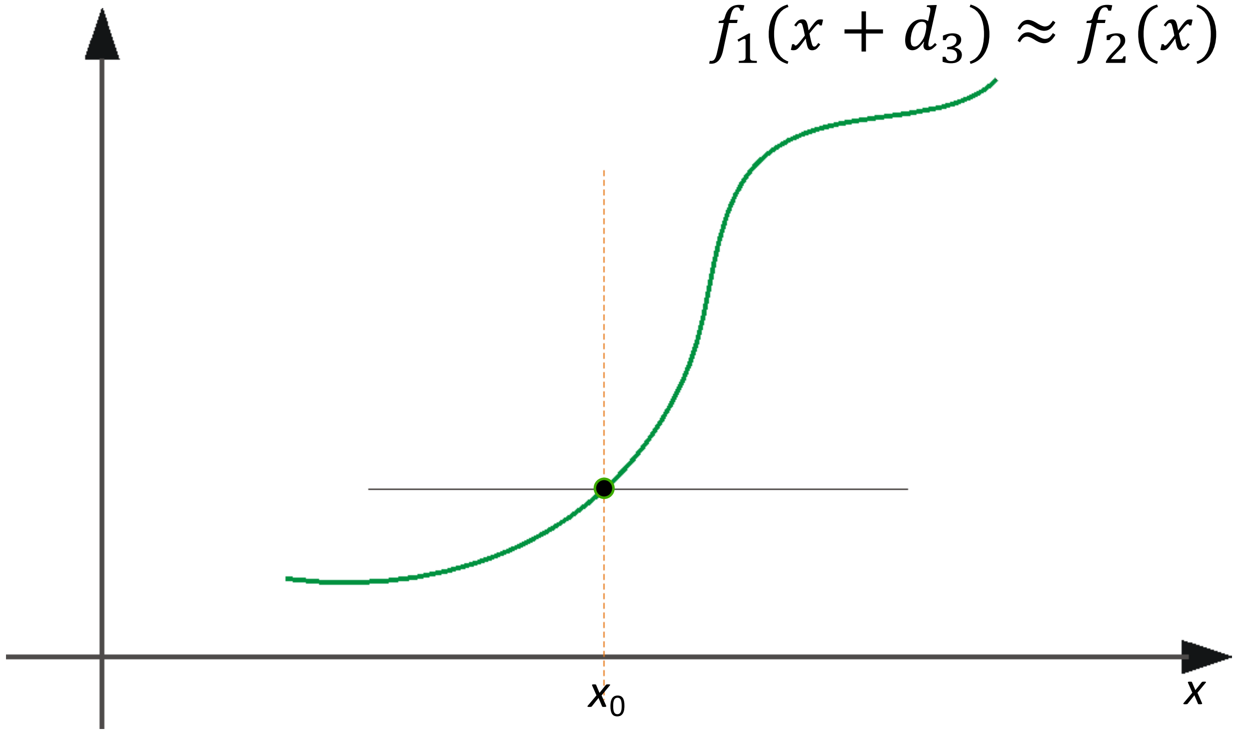
Estimate: $d_3 = d_2 + \hat{d}$

Slide adapted from Richard Szeliski

Gauss Newton Method

Compute d to minimize $\|f_1(x+d) - f_2(x)\|^2$

$$f_1(x + d_3) \approx f_2(x)$$



Slide adapted from Richard Szeliski

Nonlinear 3D-2D Pose Estimation

$\boldsymbol{\theta}$: $Euler\ Angles \in R^3$ $\quad$ $\boldsymbol{t}$: $Translation \in R^3$ $\quad$ $\pi(\cdot)$: $projection\ function$

1) Nonlinear least square

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K} \cdot \left(\boldsymbol{R}(\boldsymbol{\theta}) \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t} \right) \right) \right\|^2 = \min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \|\gamma_i(\boldsymbol{\theta}, \boldsymbol{t})\|^2$$

2) Linearization

$$\gamma_i(\boldsymbol{\theta}, \boldsymbol{t}) \approx \gamma_i\overbrace{(\boldsymbol{\theta_0}, \boldsymbol{t_0})}^{\text{Initial values}} + \underbrace{\frac{\partial \gamma_i}{\partial \boldsymbol{\theta_i}, \boldsymbol{t_i}}\Big|_{\boldsymbol{\theta_0},\boldsymbol{t_0}}}_{2\ \times 6\ matrix} \cdot \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \gamma_i(\boldsymbol{\theta_0}, \boldsymbol{t_0}) + \underbrace{\boldsymbol{J_i}}_{2\ \times 6\ matrix} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix}$$

The problem becomes:

$$\min_{\delta\boldsymbol{\theta},\delta\boldsymbol{t}} \sum_i \left\| \gamma_i(\boldsymbol{\theta_0}, \boldsymbol{t_0}) + \boldsymbol{J_i} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} \right\|^2$$

Nonlinear 3D-2D Pose Estimation

$\boldsymbol{\theta}$: $Euler\ Angles\ \epsilon\ R^3$ $\quad$ $\boldsymbol{t}$: $Translation\ \epsilon\ R^3$ $\quad$ $\pi(\cdot)$: $projection\ function$

3) Take derivative and set it to zero

$$\sum_i \boldsymbol{J}_i^T \boldsymbol{J}_i \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} + \sum_i \boldsymbol{J}_i^T \gamma_i(\boldsymbol{\theta}_0, \boldsymbol{t}_0) = 0 \Rightarrow \underbrace{\sum_i \boldsymbol{J}_i^T \boldsymbol{J}_i}_{\mathbf{A}} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \underbrace{- \sum_i \boldsymbol{J}_i^T \gamma_i(\boldsymbol{\theta}_0, \boldsymbol{t}_0)}_{\mathbf{b}}$$

4) Solve for the incremental states and update the optimization variables

$$\begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix} = \boldsymbol{A}^{-1}\boldsymbol{b} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_0 \\ \boldsymbol{t}_0 \end{bmatrix} + \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{t} \end{bmatrix}$$

5) Iterate from step 1)

Nonlinear State Estimation

- The nonlinear Gauss Newton method also works for 3D-3D pose estimation.

- In fact, it works for all nonlinear optimization problems, if provided with sufficiently good initial values.

- Simultaneous localization and mapping (SLAM) problems (with only cameras or with heterogeneous sensors) are solved in essentially the same way.

- Anything missing?

???

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K} \cdot \left(\boldsymbol{R}(\boldsymbol{\theta}) \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t} \right) \right) \right\|^2$$

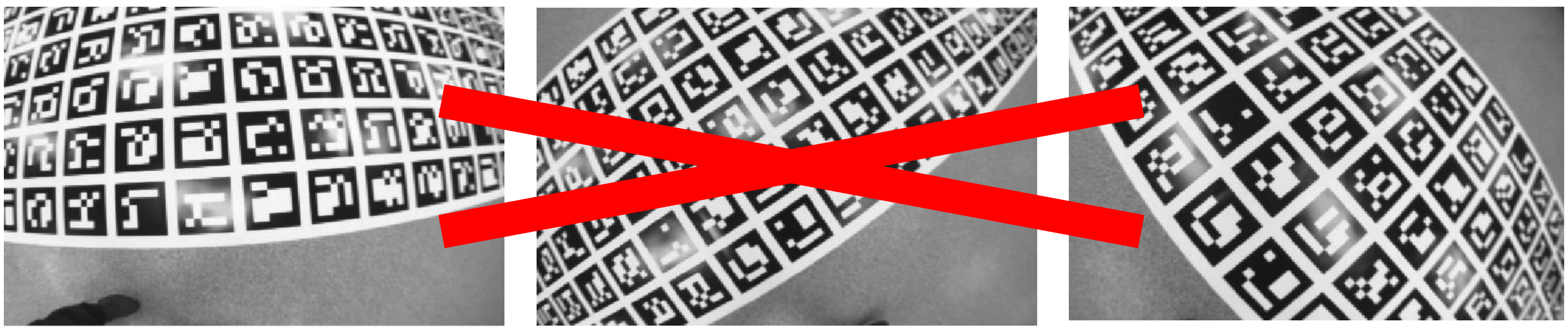
Outlier Rejection and Robust Estimation

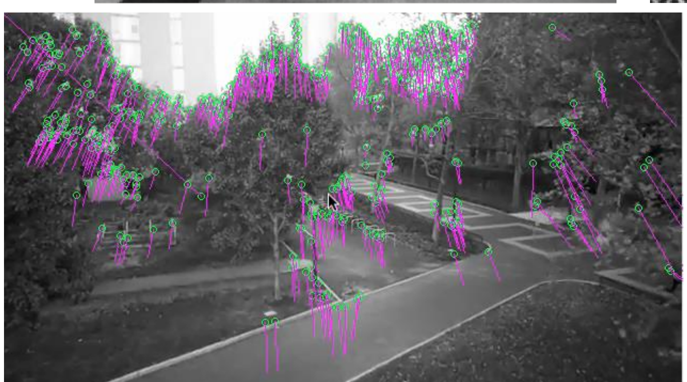
3D-2D Pose Estimation in Unstructured Environments

- Why do you need markers?
  - For data association
  - What if you do not have them?

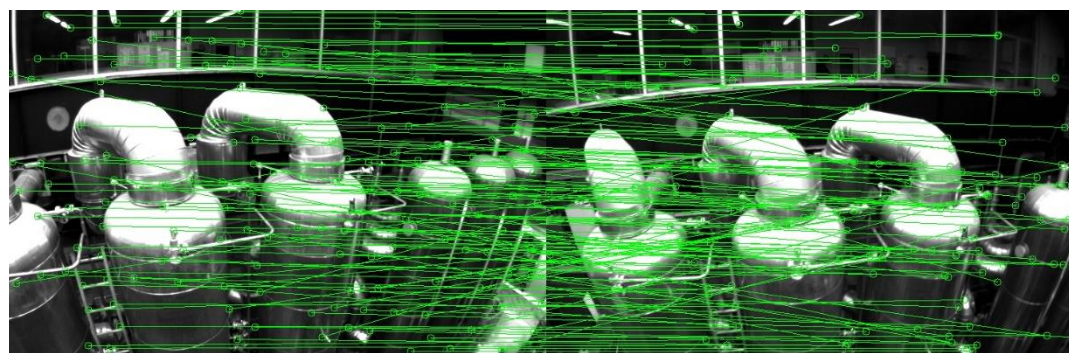
???

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K} \cdot \left(\boldsymbol{R}(\boldsymbol{\theta}) \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t} \right) \right) \right\|^2$$





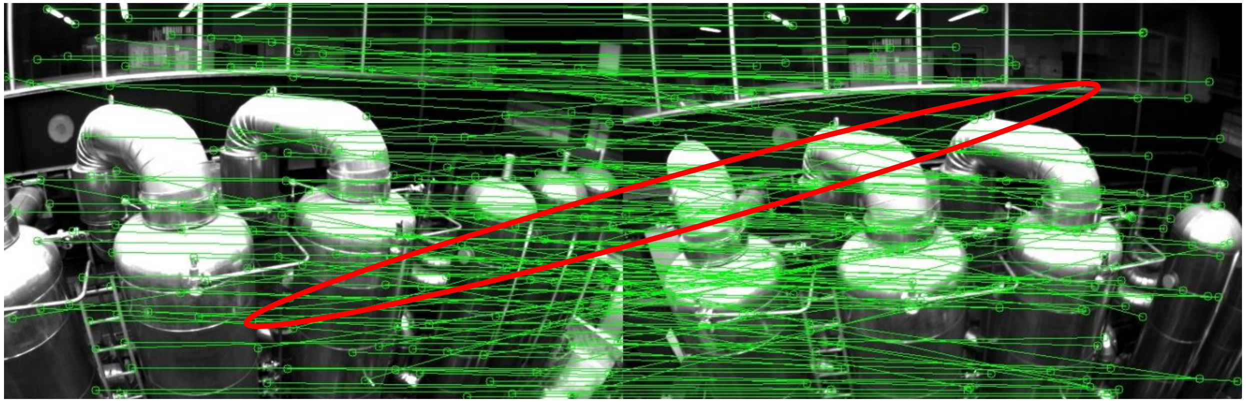
Optical flow (next lecture)

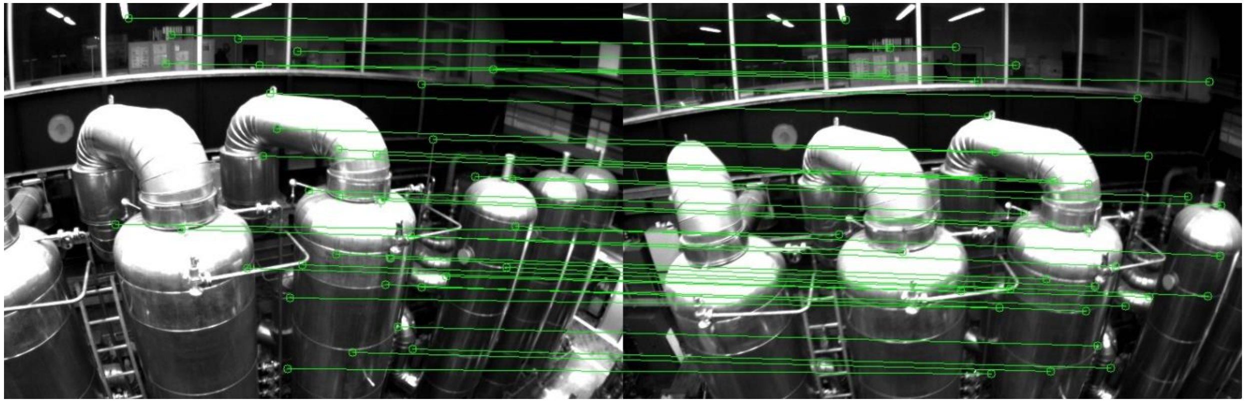


Feature matching

3D-2D Pose Estimation in Unstructured Environments

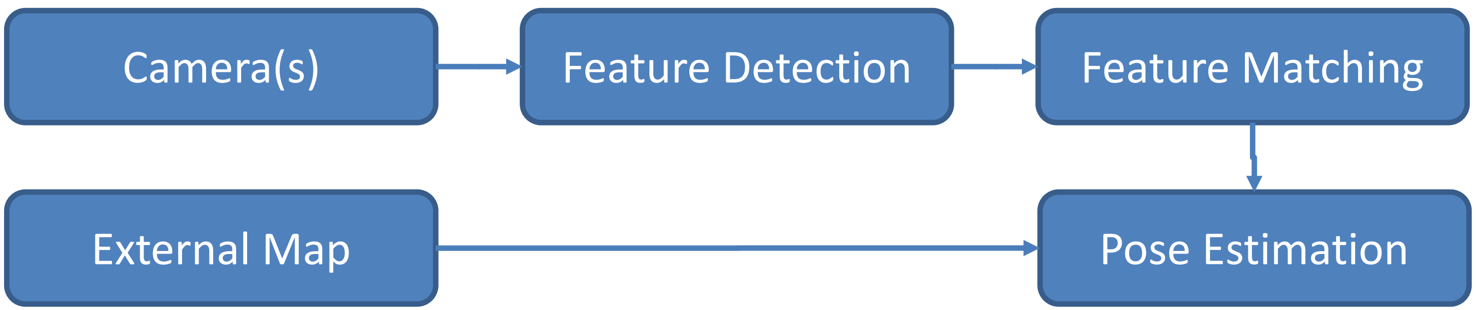
- What if you do not have markers?
  - Outlier rejection

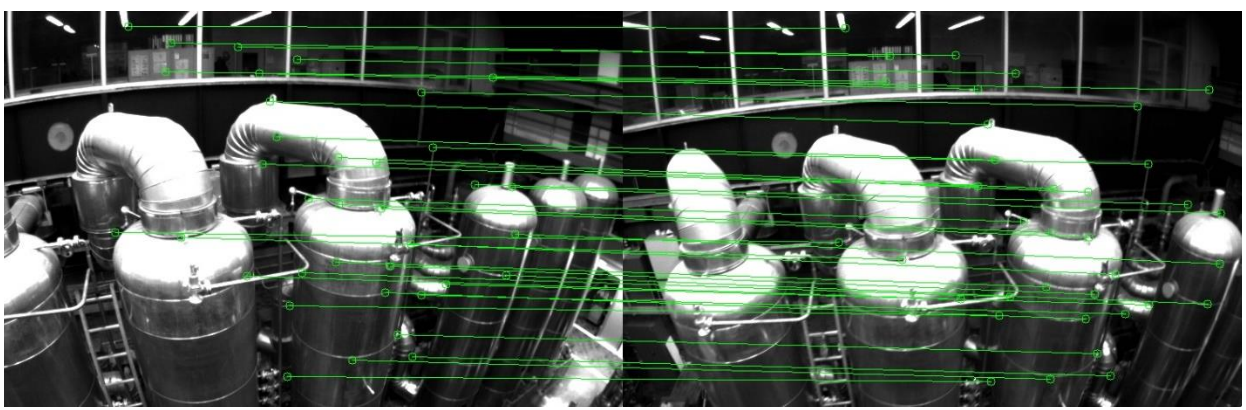




3D-2D Pose Estimation in Unstructured Environments

- What if you do not have markers?
  - Map representation – features stored in a database
  - Data association – descriptor matching
  - Outlier rejection – RANSAC





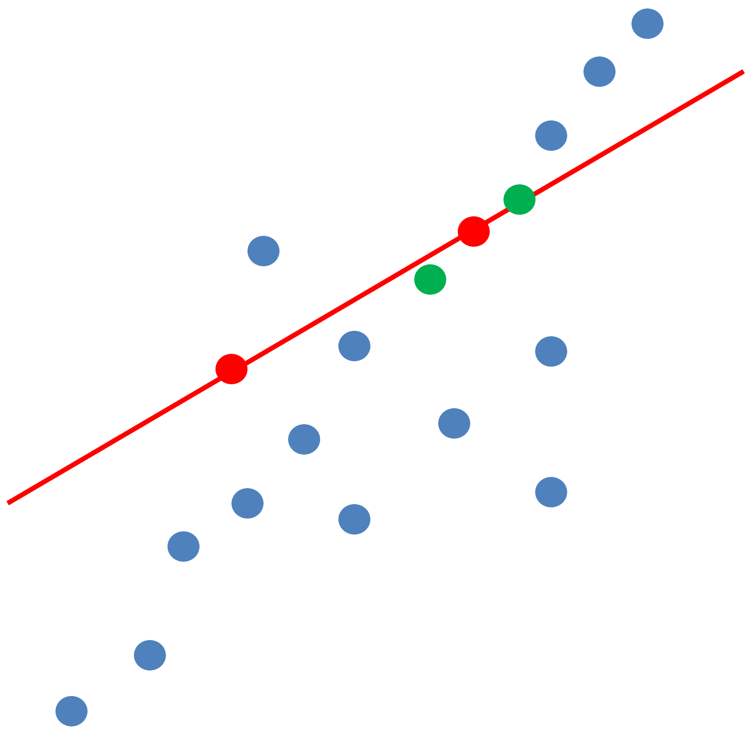
RANdom SAmple Consensus (RANSAC)

- Model fitting with outlier rejection
  - The 6-DOF pose you are trying to estimation is a model
- Algorithm:
  - Loop:
    - Randomly select a small amount of (or minimum) data points to find a model
    - See the error between the model and all other data points
    - Find the data points with error smaller than a threshold as inliers
    - If the current model has move inliers than all previous ones, record all inliers
    - Repeat
  - Use all inliers to find the best estimate of the model

RANdom SAmple Consensus (RANSAC)

- RANSAC for 2D line fitting
  - Minimum number of points to define a 2D line: 2
  - Error metric: point to line distance
  - How many iterations are required?

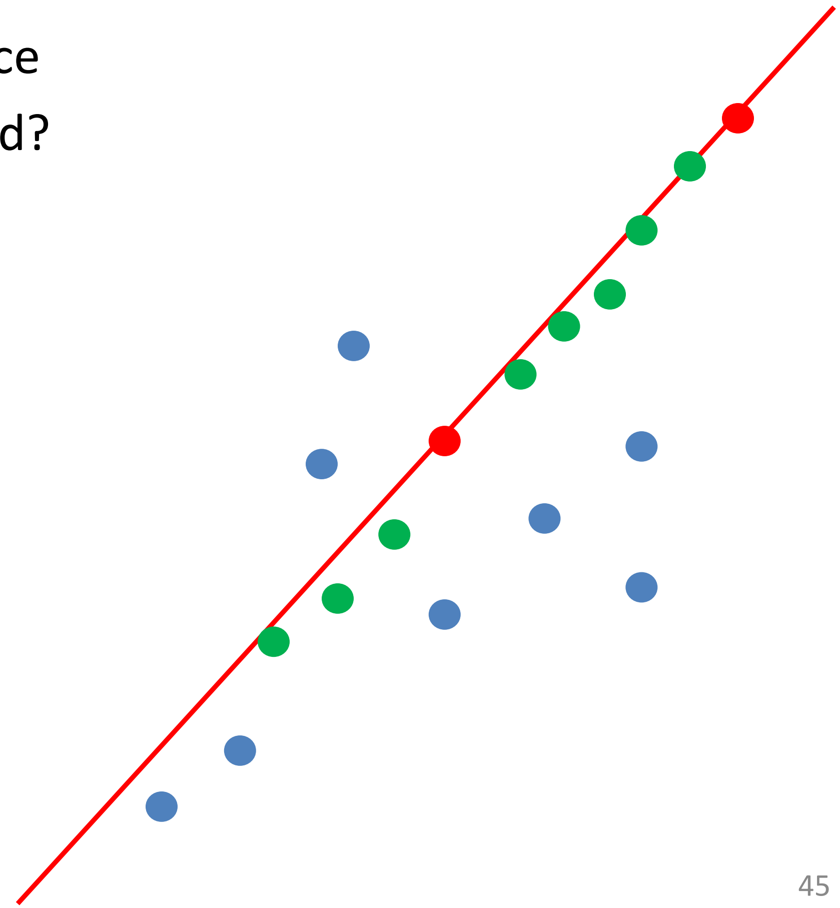
- Iteration 1: 4 Inliers



RANdom SAmple Consensus (RANSAC)

- RANSAC for 2D line fitting
  - Minimum number of points to define a 2D line: 2
  - Error metric: point to line distance
  - How many iterations are required?

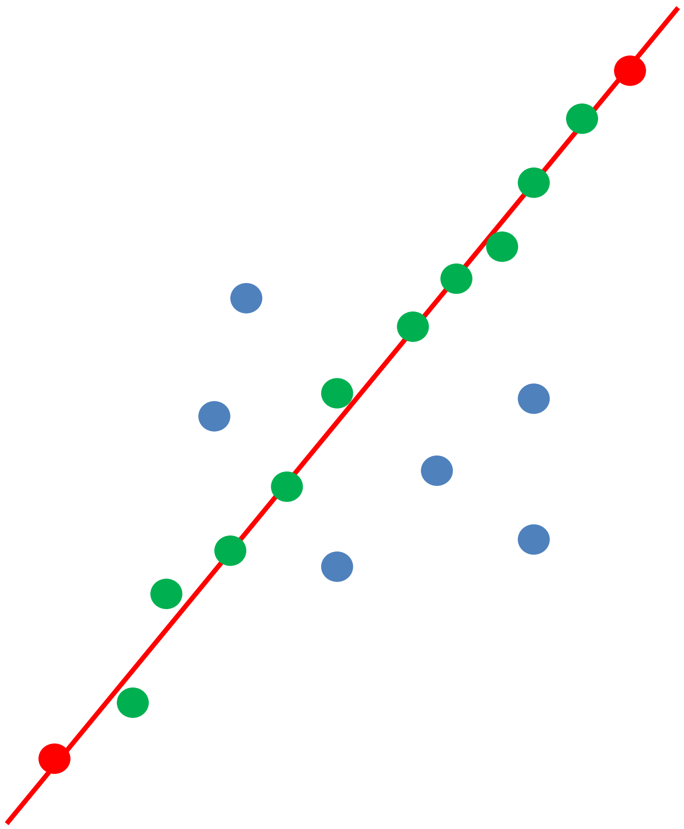
- Iteration 2: 10 Inliers



RANdom SAmple Consensus (RANSAC)

- RANSAC for 2D line fitting
  - Minimum number of points to define a 2D line: 2
  - Error metric: point to line distance
  - How many iterations are required?

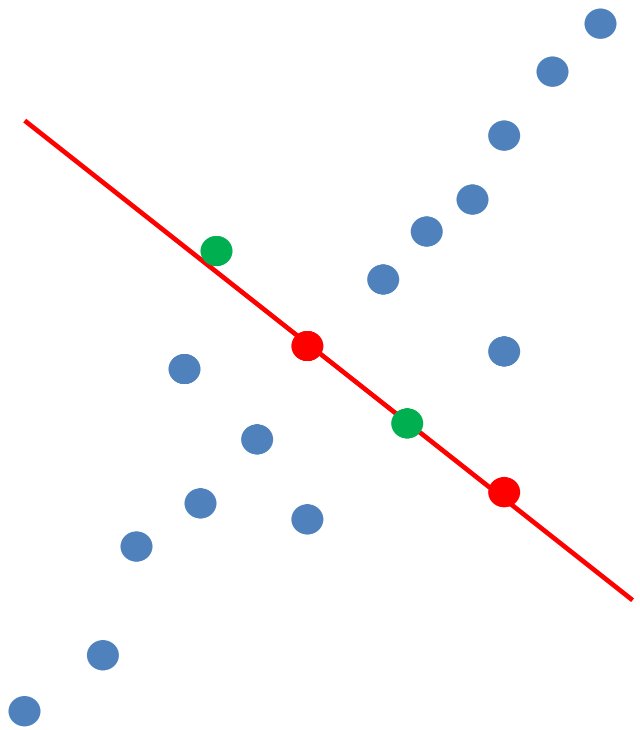
- Iteration 3: 12 Inliers



RANdom SAmple Consensus (RANSAC)

- RANSAC for 2D line fitting
  - Minimum number of points to define a 2D line: 2
  - Error metric: point to line distance
  - How many iterations are required?

- Iteration 4: 4 Inliers



Pose Estimation in Unstructured Environments

- How many feature correspondences are required to create a model?
  - 3D-3D: 3
  - 3D-2D: 3
  - It is OK to use more points to find the model (our 3D-2D requires 4)
  - But few number of points is better (Why?)
- How to find the model from feature correspondences?
- How to define the error metric?
  - 3D-3D: point distance
  - 3D-2D: reprojection error
- How many iterations are required?

RANdom SAmple Consensus (RANSAC)

- How many iterations are required - the probability
  - Probability of outlier: X (X < 1)
  - M number of data points to create a model
  - N iterations
- RANSAC failure: all random sample contains at least 1 outlier
  - Failure probability = $(1-(1-X)^{M})^{N}$
  - 2D line fitting example
    - 30% outliers
    - 2 data points to create a model
    - Failure probability for 5 iterations: 3.45%
    - Failure probability for 10 iterations: 0.12%

RANdom SAmple Consensus (RANSAC)

- How many iterations are required - the probability
  - Probability of outlier: X (X < 1)
  - M number of data points to create a model
  - N iterations
- RANSAC failure: all random sample contains at least 1 outlier
  - Failure probability = $(1-(1-X)^M)^N$
  - 3D-3D pose estimation
    - 30% outliers
    - 3 data points to create a model
    - Failure probability for 5 iterations: 12.24%
    - Failure probability for 10 iterations: 1.49%
    - Failure probability for 20 iterations: 0.02%

RANdom SAmple Consensus (RANSAC)

- How many iterations are required - the probability
  - Probability of outlier: X (X < 1)
  - M number of data points to create a model
  - N iterations
- RANSAC failure: all random sample contains at least 1 outlier
  - Failure probability = $(1-(1-X)^M)^N$
  - 3D-3D pose estimation
    - 30% outliers
    - 20 data points to create a model (bad example)
    - Failure probability for 5 iterations: 99.6%
    - Failure probability for 10 iterations: 99.2%
    - Failure probability for 20 iterations: 98.4%
    - Failure probability for 1000 iterations: 45%
    - Failure probability for 10000 iterations: 0.03%

Robust M-Estimator

$\boldsymbol{\theta}$: $Euler\ Angles \epsilon R^3$ $\boldsymbol{t}$: $Translation \epsilon R^3$ $\pi(\cdot)$: $projection\ function$

- Recall the nonlinear 3D-2D pose estimation problem

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi\left(\boldsymbol{K} \cdot \left(\boldsymbol{R}(\boldsymbol{\theta}) \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \boldsymbol{t} \right) \right) \right\|^2 = \min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \|\gamma_i(\boldsymbol{\theta}, \boldsymbol{t})\|^{\mathbf{2}}$$

- The original least square problem is very sensitive to outliers, due to the squared influence of data terms
- Even after RANSAC, there may still be outlier (or relatively bad "inliers") exists
- We want a method to reduce the impact of outliers

Robust M-Estimator

- Rewrite in a general form for minimizing w.r.t parameter p

$$\min_{\boldsymbol{\theta},\boldsymbol{t}} \sum_i \|\gamma_i(\boldsymbol{\theta},\boldsymbol{t})\|^2 \Rightarrow \min_p \sum_i \rho(\gamma_i(p))$$

  - ρ is a symmetric, positive-definite function with unique minimum at 0
- Take the derivative and set to 0

$$\sum_i \frac{\partial \rho}{\partial \gamma_i} \frac{\partial \gamma_i}{\partial p} = \sum_i \psi(\gamma_i) \frac{\partial \gamma_i}{\partial p} = 0$$

  - $\psi(x) = \partial\rho/\partial x$ is called the influence function
  - We can also define a weight function $\omega(x) = \psi(x)/x$
- We then have:

$$\sum_i \omega(\gamma_i)\gamma_i \frac{\partial \gamma_i}{\partial p} = 0$$

- This is exactly the system we obtain if we solve the following iterative re-weighted least square problem:

$$\min_p \sum_i \omega(\gamma_i^{k-1}) \|\gamma_i(p)\|^2$$

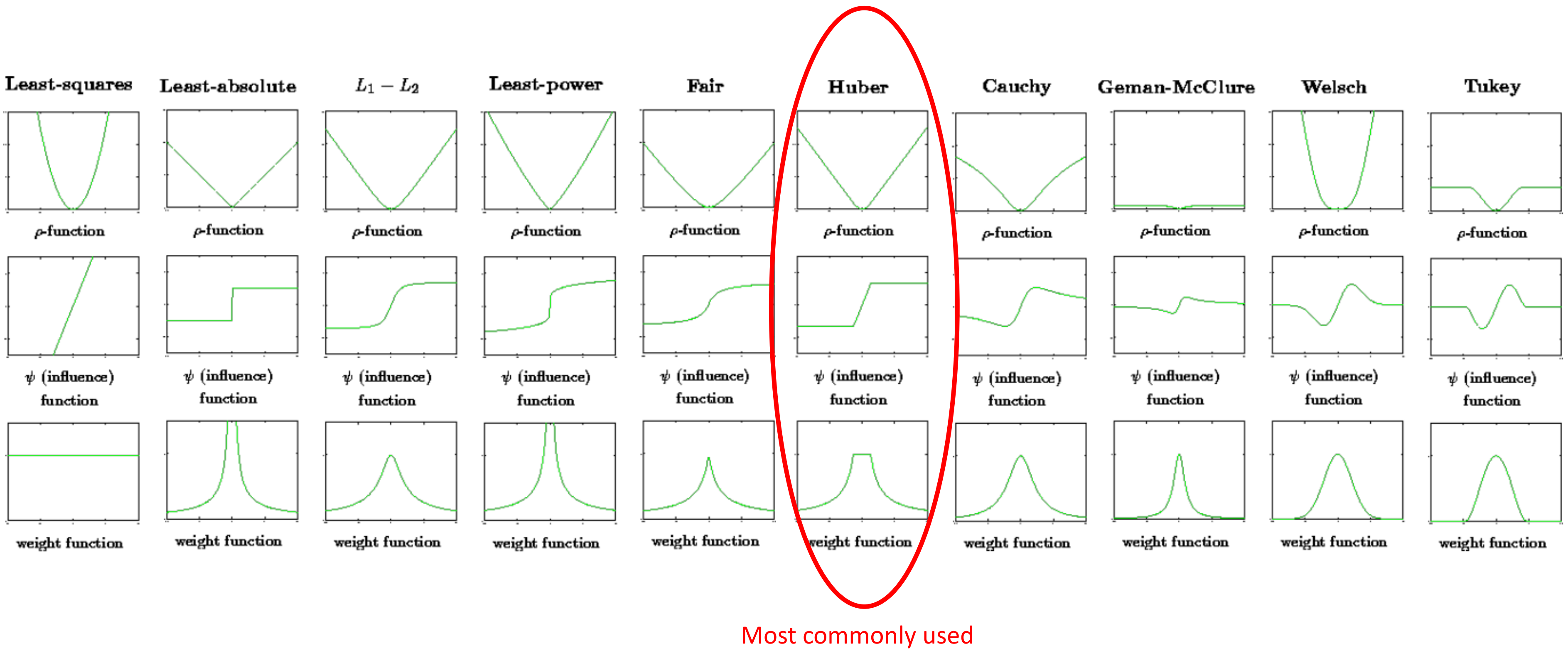
Robust M-Estimator

- The influence function $\psi(\cdot)$ measures the influence of a datum on the value of parameter estimation
- For least square with $\rho(x) = x^2/2$, the influence is $\psi(x) = x$. The influence of a datum increases w.r.t. the size of error
- When an estimator is robust, we want influence of any single datum is insufficient to yield any significant offsets, provided following constraints are met:
  - Bounded influence function
  - Individual $\rho(\cdot)$ function is convex in parameter p

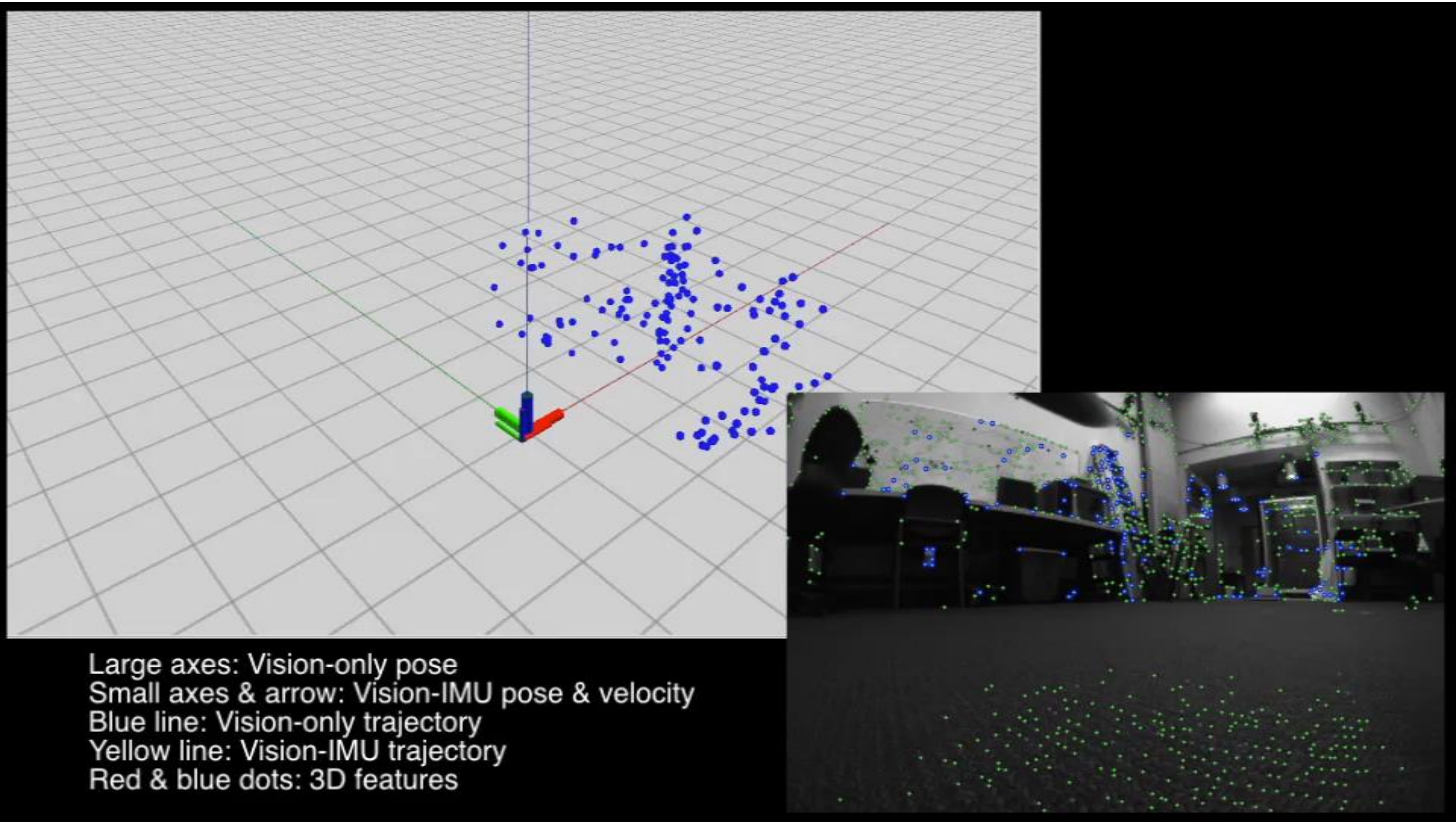
Robust M-Estimator

type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$x^2/2$	x	1
L_1	$\lvert x\rvert$	$\operatorname{sgn}(x)$	$\frac{1}{\lvert x\rvert}$
$L_1 - L_2$	$2(\sqrt{1+x^2/2}-1)$	$\frac{x}{\sqrt{1+x^2/2}}$	$\frac{1}{\sqrt{1+x^2/2}}$
L_p	$\frac{\lvert x\rvert^{\nu}}{\nu}$	$\operatorname{sgn}(x)\lvert x\rvert^{\nu-1}$	$\lvert x\rvert^{\nu-2}$
"Fair"	$c^2[\frac{\lvert x\rvert}{c} - \log(1+\frac{\lvert x\rvert}{c})]$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
Huber $\begin{cases} \text{if } \lvert x\rvert \le k \\ \text{if } \lvert x\rvert \ge k \end{cases}$	$\begin{cases} x^2/2 \\ k(\lvert x\rvert - k/2) \end{cases}$	$\begin{cases} x \\ k\operatorname{sgn}(x) \end{cases}$	$\begin{cases} 1 \\ k/\lvert x\rvert \end{cases}$
Cauchy	$\frac{c^2}{2}\log(1+(x/c)^2)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$
Welsch	$\frac{c^2}{2}[1-\exp(-(x/c)^2)]$	$x\exp(-(x/c)^2)$	$\exp(-(x/c)^2))$
Tukey $\begin{cases} \text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c \end{cases}$	$\begin{cases} \frac{c^2}{6}(1-[1-(x/c)^2]^3) \\ (c^2/6) \end{cases}$	$\begin{cases} x[1-(x/c)^2]^2 \\ 0 \end{cases}$	$\begin{cases} [1-(x/c)^2]^2 \\ 0 \end{cases}$

Robust M-Estimator



3D-2D Vision-Based Localization



Logistics

- Project 2, phase 1 is released (03/16)
  - Due in one week: 03/23
- Midterm grades are released, send your paper checking request by this Friday
- 16 students are enrolled in the optional physical lab, 2 per group, detail instruction will be available soon.
- Keep the mixed-mode teaching…