Camera Pose, Minimum Solution, and Outlier Rejection

Computer Vision II

CSE 252B

Lecture 9

Announcements

- Assignment 2 is due today, 11:59 PM
- Assignment 3 will be released today
  - Due Feb 18, 11:59 PM
- Reading
  - Haralick et al., “Review and Analysis of Solutions of the Three Point Perspective”
  - Section 4.7

Calibrated camera

Camera calibration matrix

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

where

$\alpha_x = f m_x$	Focal length in x direction in terms of pixel dimensions
$\alpha_y = f m_y$	Focal length in y direction in terms of pixel dimensions
s	Skew
$(x_0, y_0)^\top = (p_x m_x, p_y m_y)^\top$	Coordinates of principal point in terms of pixel dimensions

where m_x and m_y are number of pixels per unit disance in x and y directions, respectively

(and lens distortion parameters)

Single view geometry

Camera projection

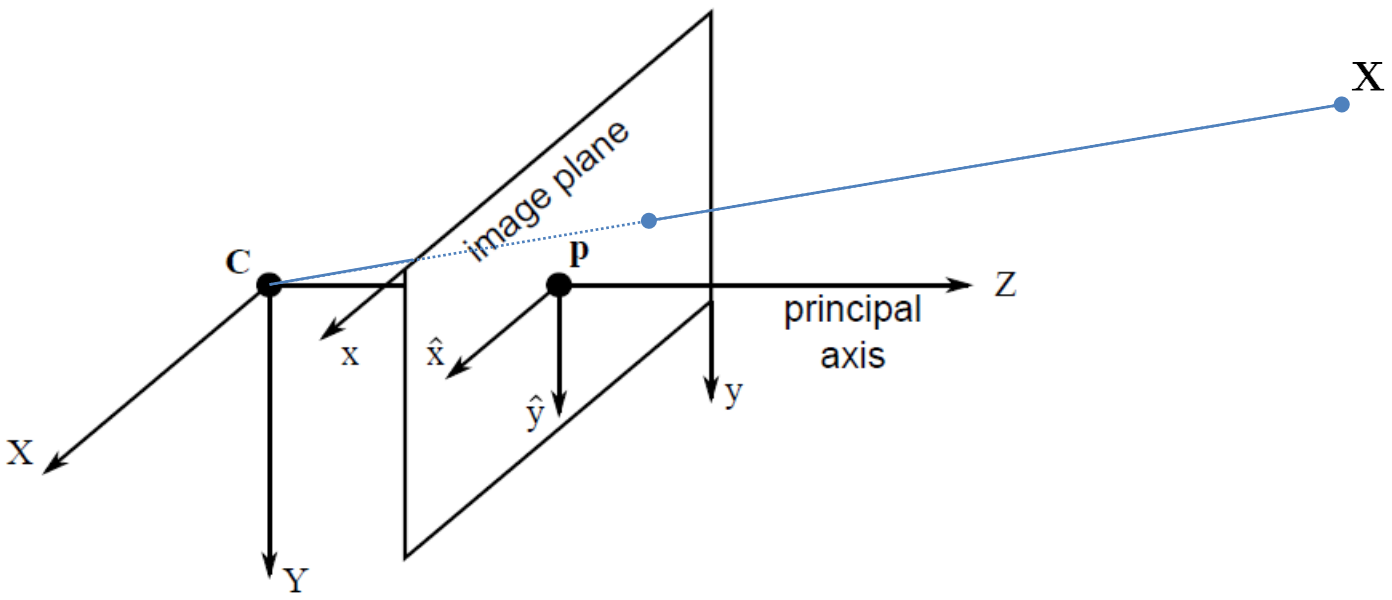
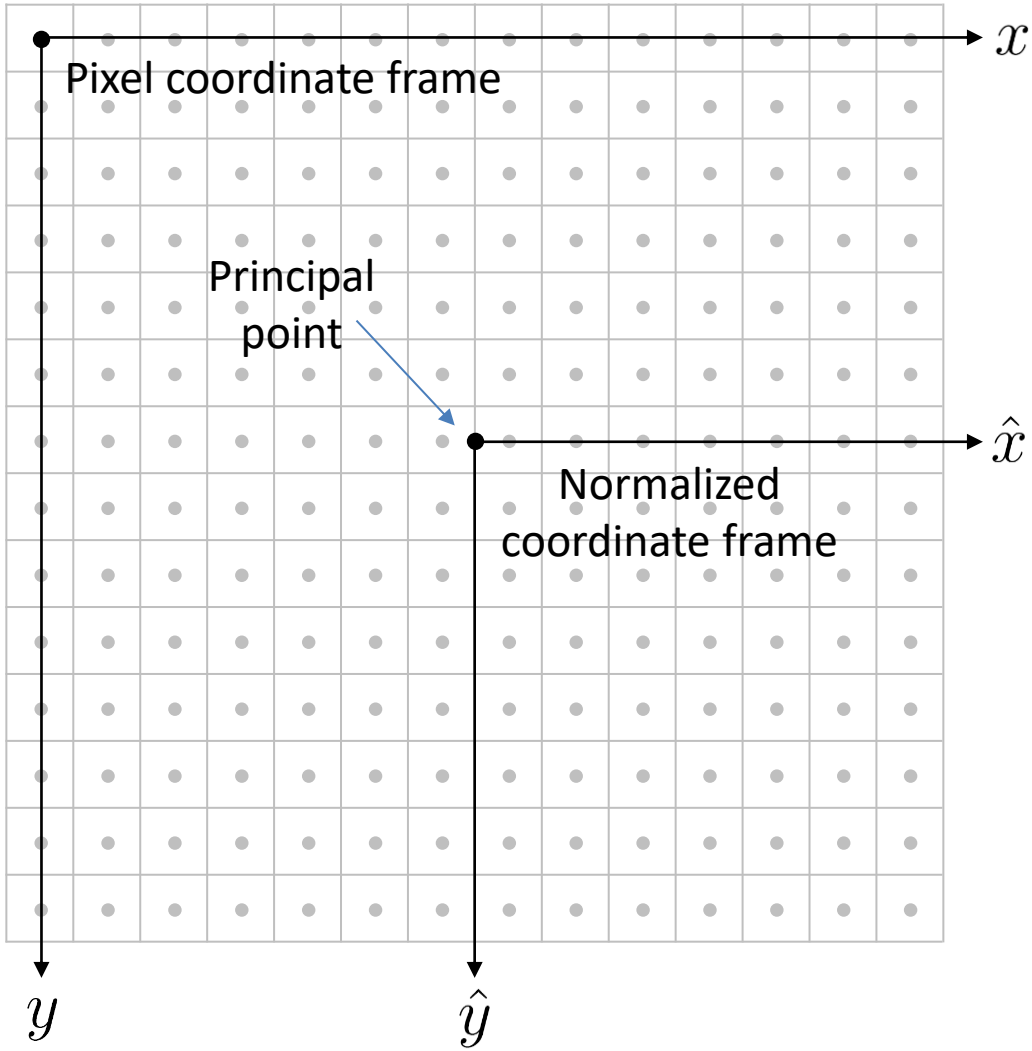


Image coordinate frames



Camera projection

Project 3D point in world coordinate frame $\mathbf{X}$ under camera projection matrix $\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$

$$\begin{aligned} \mathbf{x} &= \mathtt{P}\mathbf{X} \\ \mathbf{x} &= \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]\mathbf{X} \\ \mathtt{K}^{-1}\mathbf{x} &= [\mathtt{R} \,|\, \mathbf{t}]\mathbf{X} \\ \hat{\mathbf{x}} &= \hat{\mathtt{P}}\mathbf{X} \end{aligned}$$

Camera calibration matrix

where

Camera projection matrix	$\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$	$= \mathtt{K}\hat{\mathtt{P}}$
Normalized camera projection matrix	$\hat{\mathtt{P}} = [\mathtt{R} \,|\, \mathbf{t}]$	$= \mathtt{K}^{-1}\mathtt{P}$
Image point in pixel coordinates	$\mathbf{x} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]\mathbf{X}$	$= \mathtt{K}\hat{\mathbf{x}}$
Image point in normalized coordinates	$\hat{\mathbf{x}} = [\mathtt{R} \,|\, \mathbf{t}]\mathbf{X}$	$= \mathtt{K}^{-1}\mathbf{x}$

Together, the camera rotation and camera translation (or camera center $\tilde{\mathbf{C}} = -\mathtt{R}^{\top}\mathbf{t}$) are called the camera pose. The normalized camera projection matrix $\hat{\mathtt{P}} = [\mathtt{R} \,|\, \mathbf{t}]$ embodies the camera pose.

Calibrated camera

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

where

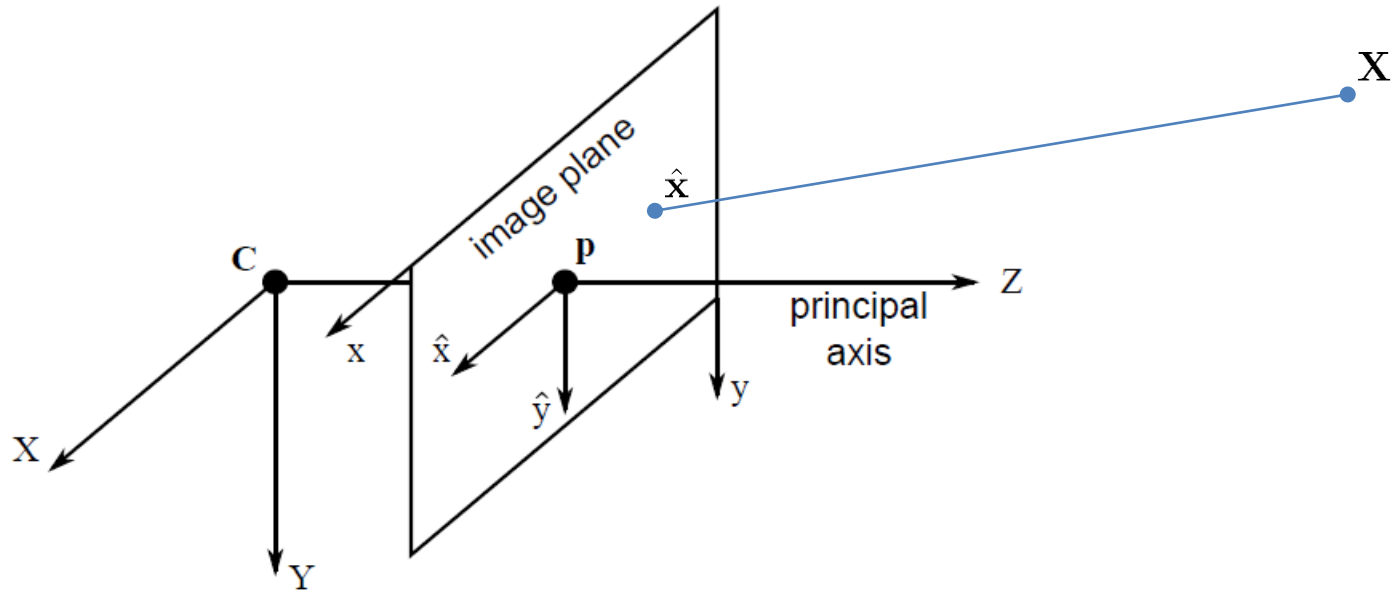
$\alpha_x = f m_x$	Focal length in x direction in terms of pixel dimensions
$\alpha_y = f m_y$	Focal length in y direction in terms of pixel dimensions
s	Skew
$(x_0, y_0)^\top = (p_x m_x, p_y m_y)^\top$	Coordinates of principal point in terms of pixel dimensions

where m_x and m_y are number of pixels per unit disance in x and y directions, respectively

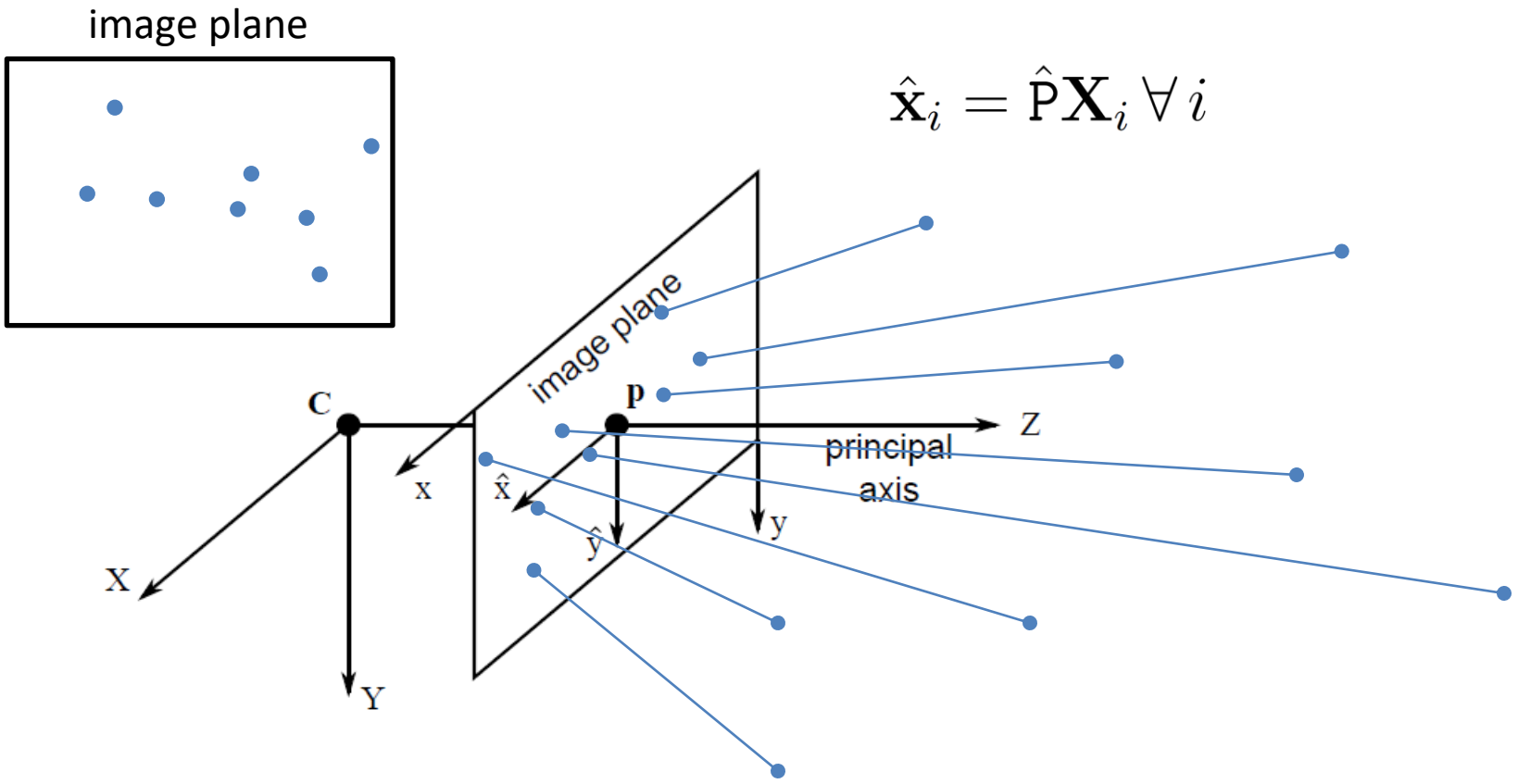
- If camera calibration parameters are known, then use normalized camera projection matrix and image points in normalized coordinates

Normalized camera projection matrix

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



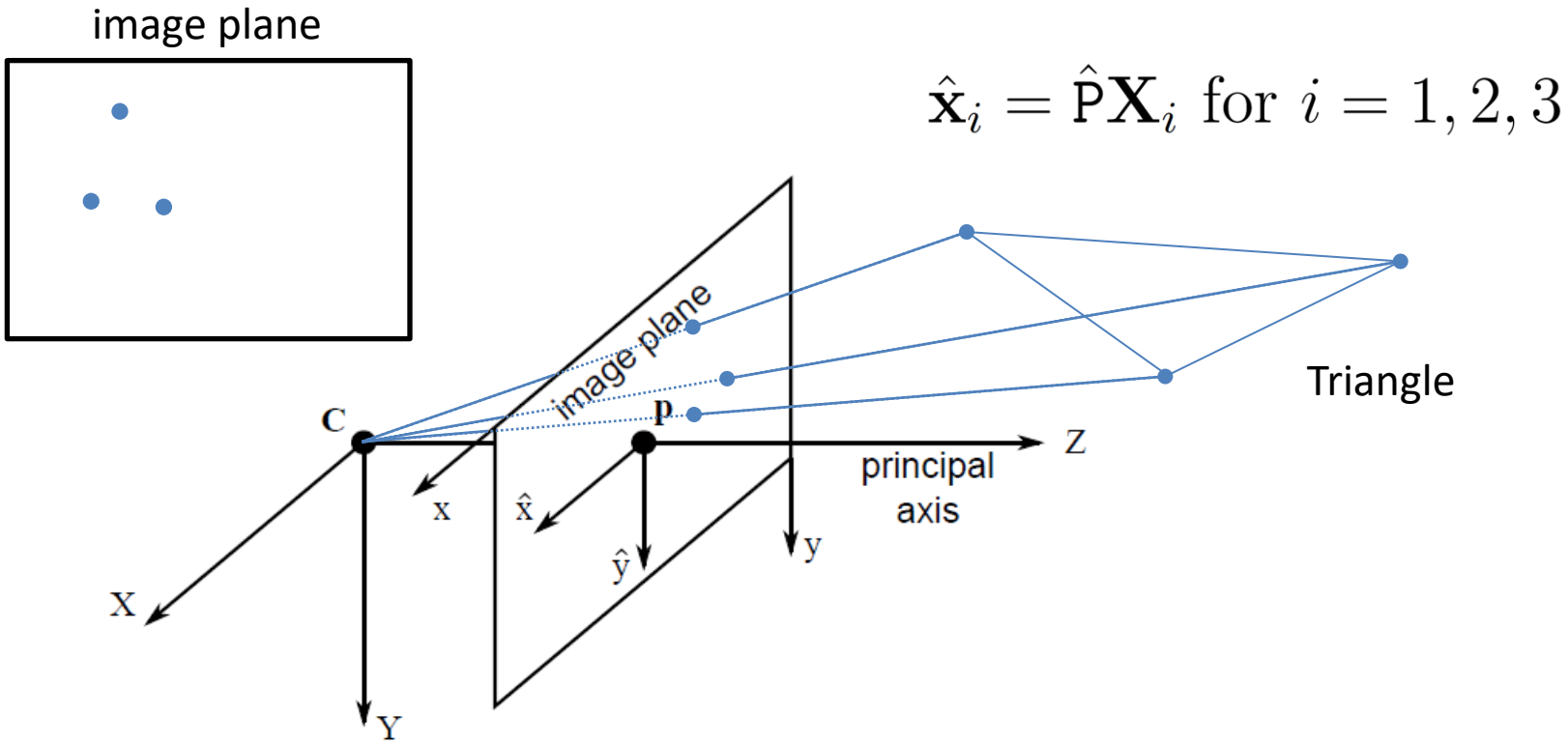
Normalized camera projection matrix



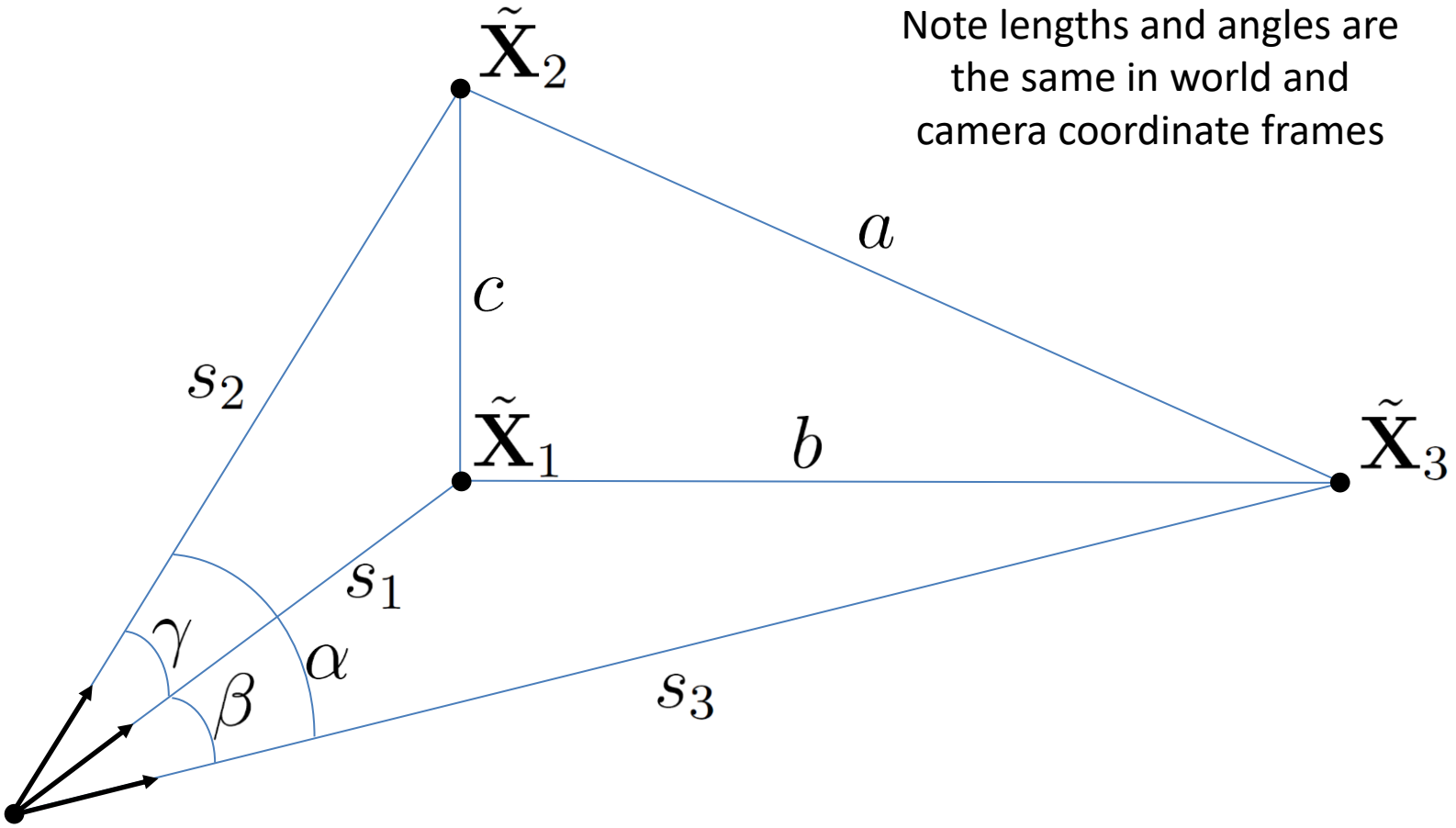
Estimation of normalized camera projection matrix

- Linear estimation
  - Requires at least 6 point correspondences, where the 3D points are in general position
  - Requires at least 4 point correspondences, where the 3D points are coplanar
- Nonlinear estimation requires at least 3 point correspondences
- The camera pose has 6 parameters, so needs 6 measurements for an exact solution
  - Two measurements for each image point (2 coordinates)

Normalized camera projection matrix



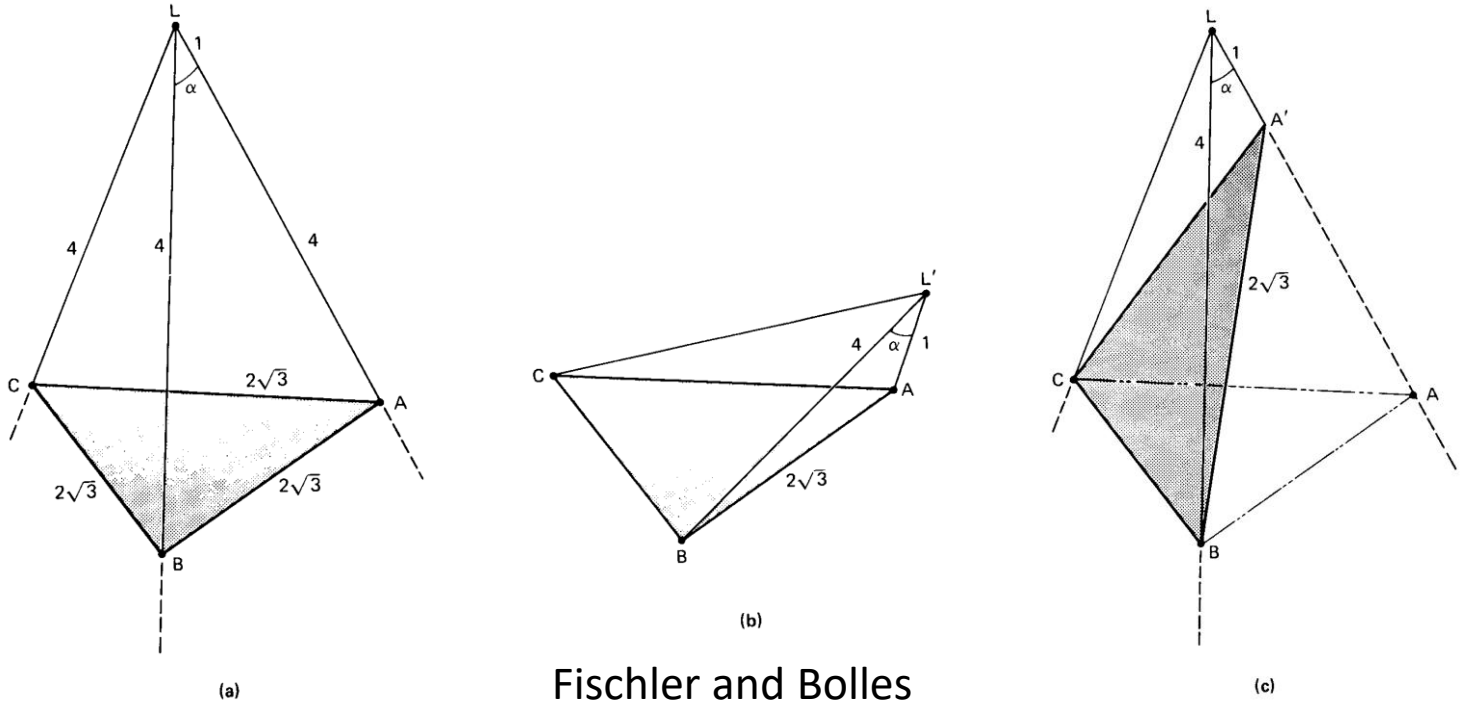
Perspective 3 Point (P3P)



Note lengths and angles are the same in world and camera coordinate frames

Camera pose, minimum solution

The minimum number of point correspondences $\hat{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{X}}_i$ needed to exactly solve for the camera pose (6 parameters) is 3 (i.e., 6 measurements). There are up to four exact solutions for the camera pose from 3 point correspondences. There are multiple approaches to solving this problem. Haralick et al. show Finsterwalder's solution is the most numerically stable method.

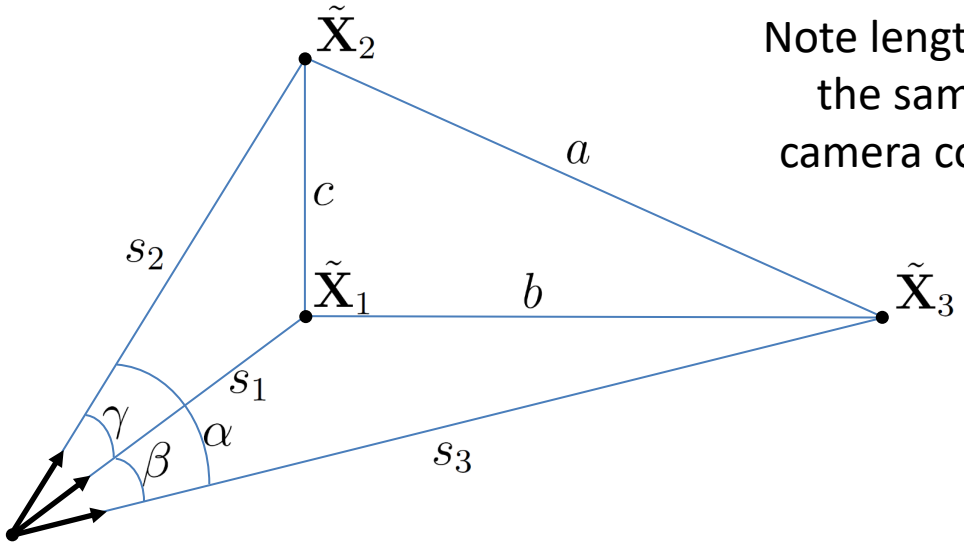


Fischler and Bolles

Camera pose, minimum solution

Given point correspondences $\hat{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{X}}_i$ for $i = 1, 2, 3$
Squared side lengths of triangle in 3D

$$a^2 = (\tilde{X}_2 - \tilde{X}_3)^2 + (\tilde{Y}_2 - \tilde{Y}_3)^2 + (\tilde{Z}_2 - \tilde{Z}_3)^2$$
$$b^2 = (\tilde{X}_1 - \tilde{X}_3)^2 + (\tilde{Y}_1 - \tilde{Y}_3)^2 + (\tilde{Z}_1 - \tilde{Z}_3)^2$$
$$c^2 = (\tilde{X}_1 - \tilde{X}_2)^2 + (\tilde{Y}_1 - \tilde{Y}_2)^2 + (\tilde{Z}_1 - \tilde{Z}_2)^2$$



Note lengths and angles are the same in world and camera coordinate frames

Backproject points to ray directions

Backprojection of image point in normalized coordinates to ray direction in camera coordinate frame

$$\hat{\mathbf{x}}_i = [\mathtt{R} \,|\, \mathbf{t}] \begin{bmatrix} \tilde{\mathbf{X}}_i \\ 1 \end{bmatrix}$$

$$\hat{\mathbf{x}}_i = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$$

$$\hat{\mathbf{x}}_i = \tilde{\mathbf{X}}_{\text{cam},i}, \text{ where } \tilde{\mathbf{X}}_{\text{cam},i} = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$$

$$\mathbf{d}_i = \tilde{\mathbf{X}}_{\text{cam},i} \textbf{ up to nonzero scale}, \text{ where } \mathbf{d}_i = \hat{\mathbf{x}}_i$$

Then, unitize $\mathbf{d}_i = (d_{i,1}, d_{i,2}, d_{i,3})^\top$ and ensure $d_{i,3}$ is positive (i.e., in front of camera)

$$\mathbf{d}_i = \frac{1}{\text{sign}(d_{i,3})\|\mathbf{d}_i\|}\mathbf{d}_i$$

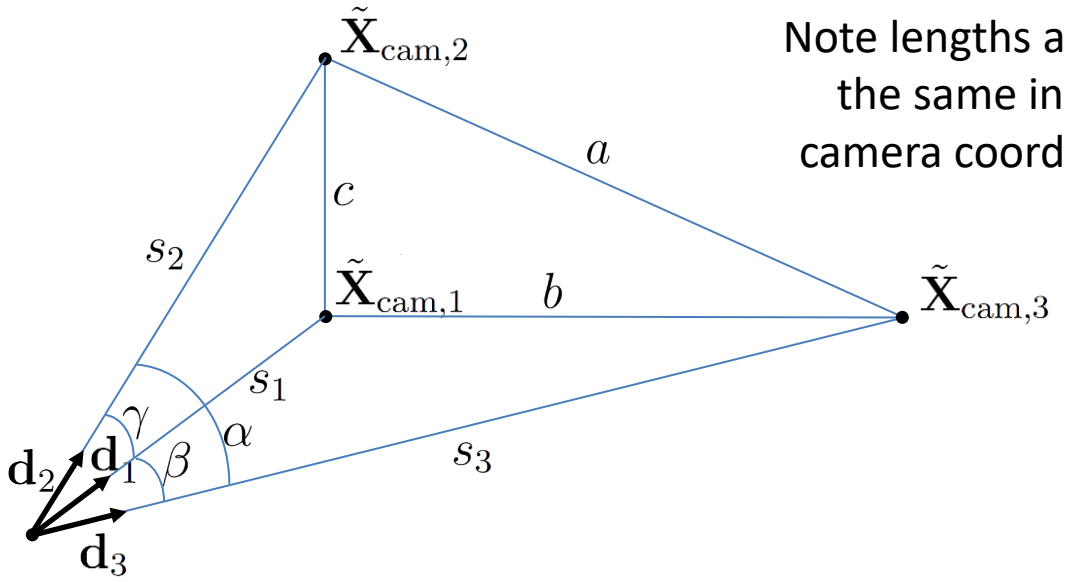
Finally

$$s_i\mathbf{d}_i = \tilde{\mathbf{X}}_{\text{cam},i} \textbf{ to scale}, \text{ where } s_i \text{ is distance from camera center to } \tilde{\mathbf{X}}_{\text{cam},i}$$

Camera pose, minimum solution

Calculate angles between ray directions

$$\cos(\alpha) = \mathbf{d}_2^\top \mathbf{d}_3 \qquad \sin^2(\alpha) = 1 - \cos^2(\alpha)$$
$$\cos(\beta) = \mathbf{d}_1^\top \mathbf{d}_3 \qquad \sin^2(\beta) = 1 - \cos^2(\beta)$$
$$\cos(\gamma) = \mathbf{d}_1^\top \mathbf{d}_2 \qquad \sin^2(\gamma) = 1 - \cos^2(\gamma)$$



Note lengths and angles are the same in world and camera coordinate frames

Camera pose, minimum solution

Solve for real roots of cubic

$$G\lambda^3 + H\lambda^2 + I\lambda + J = 0, \text{ solve for } \lambda$$

where

See Haralick et al. for derivation

$$\begin{aligned}
G &= c^2(c^2\sin^2(\beta) - b^2\sin^2(\gamma)) \\
H &= b^2(b^2 - a^2)\sin^2(\gamma) + c^2(c^2 + 2a^2)\sin^2(\beta) + 2b^2c^2(\cos(\alpha)\cos(\beta)\cos(\gamma) - 1) \\
I &= b^2(b^2 - c^2)\sin^2(\alpha) + a^2(a^2 + 2c^2)\sin^2(\beta) + 2a^2b^2(\cos(\alpha)\cos(\beta)\cos(\gamma) - 1) \\
J &= a^2(a^2\sin^2(\beta) - b^2\sin^2(\alpha))
\end{aligned}$$

Up to three real roots, pick any one. λ_0 is first real root.

Camera pose, minimum solution

Implicitly solve $Au^2 + 2Buv + Cv^2 + 2Du + 2Ev + F = 0$ for v, where

$$
\begin{aligned}
A &= b^2(\lambda_0 + 1) \\
B &= -b^2 \cos(\alpha) \\
C &= -a^2 + b^2 - \lambda_0 c^2 \\
D &= -b^2 \lambda_0 \cos(\gamma) \\
E &= \cos(\beta)(a^2 + \lambda_0 c^2) \\
F &= \lambda_0(b^2 - c^2) - a^2
\end{aligned}
$$

See Haralick et al. for derivation

by **explicitly** solving for real roots of quadratic

$$(b^2 - m_j^2 c^2)u^2 + 2(c^2(\cos(\beta) - n_j)m_j - b^2 \cos(\gamma))u - c^2 n_j^2 + 2c^2 n_j \cos(\beta) + b^2 - c^2 = 0$$

for $j = 1, 2$, solve for u

where

$$
\begin{aligned}
m_1 &= (-B + p)/C \qquad & m_2 &= (-B - p)/C \\
n_1 &= -(E - q)/C \qquad & n_2 &= -(E + q)/C
\end{aligned}
$$

where

$$
\begin{aligned}
p &= \sqrt{B^2 - AC} \\
q &= \text{sign}(BE - CD)\sqrt{E^2 - CF}
\end{aligned}
$$

Camera pose, minimum solution

The first solution for the quadratic using m_1 and n_1 results in up to two real roots. For each real root u, solve for the 3D points in the camera coordinate frame as follows.

$$v = um_1 + n_1$$

$$s_1 = \sqrt{b^2/(1 + v^2 - 2v\cos(\beta))}$$
$$s_2 = us_1$$
$$s_3 = vs_1$$

Only use solutions where $s_1 > 0, s_2 > 0$, and $s_3 > 0$ (i.e., points are in front of camera)

$\tilde{\mathbf{X}}_{\text{cam},i} = s_i \mathbf{d}_i$ for $i = 1, 2, 3$

Then, calculate the exact Euclidean transformation (i.e., the camera pose $\mathtt{R}$ and $\mathbf{t}$) $\tilde{\mathbf{X}}_{\text{cam},i} = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$ for $i = 1, 2, 3$.

Camera pose, minimum solution

Similarly, the second solution for the quadratic using m_2 and n_2 results in up to two real roots. For each real root u, solve for the 3D points in the camera coordinate frame as follows.

$$v = um_2 + n_2$$

$$s_1 = \sqrt{b^2/(1 + v^2 - 2v\cos(\beta))}$$
$$s_2 = us_1$$
$$s_3 = vs_1$$

Only use solutions where $s_1 > 0, s_2 > 0$, and $s_3 > 0$ (i.e., points are in front of camera)

$\tilde{\mathbf{X}}_{\text{cam},i} = s_i\mathbf{d}_i$ for $i = 1, 2, 3$

Then, calculate the exact Euclidean transformation (i.e., the camera pose $\mathtt{R}$ and $\mathbf{t}$) $\tilde{\mathbf{X}}_{\text{cam},i} = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$ for $i = 1, 2, 3$.

Since there are up to four real solutions for u, there are up to four real solutions for the camera pose.

Estimate 3D Euclidean transformation from points in the world coordinate frame to points in the camera coordinate frame

$\tilde{\mathbf{X}}_{\text{cam},i} = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t} \,\forall\, i$, solve for $\mathtt{R}$ and $\mathbf{t}$ using $\tilde{\mathbf{X}}_{\text{cam},i} \leftrightarrow \tilde{\mathbf{X}}_i$

1. Calculate the mean vectors $\boldsymbol{\mu}_{\tilde{\mathbf{X}}}$ and $\boldsymbol{\mu}_{\tilde{\mathbf{X}}_{\text{cam}}}$ (i.e., centroids) of the set of 3D points in world coordinate frame and camera coordinate frame, respectively

2. Compute the matrix

$$\mathtt{S} = \sum_i \left(\tilde{\mathbf{X}}_{\text{cam},i} - \boldsymbol{\mu}_{\tilde{\mathbf{X}}_{\text{cam}}}\right)\left(\tilde{\mathbf{X}}_i - \boldsymbol{\mu}_{\tilde{\mathbf{X}}}\right)^\top$$

   and its singular value decomposition $\mathtt{S} = \mathtt{U}\Sigma\mathtt{V}^\top$

3. The camera rotation matrix

$$\mathtt{R} = \begin{cases} \mathtt{U}\,\mathrm{diag}(1,1,-1)\mathtt{V}^\top & \text{if } \det(\mathtt{U})\det(\mathtt{V}) < 0 \\ \mathtt{U}\mathtt{V}^\top & \text{otherwise} \end{cases}$$

   such that $\det(\mathtt{R}) = +1$

4. The camera translation vector $\mathbf{t} = \boldsymbol{\mu}_{\tilde{\mathbf{X}}_{\text{cam}}} - \mathtt{R}\boldsymbol{\mu}_{\tilde{\mathbf{X}}}$

Outlier rejection

Outlier rejection

- Even the presence of a single outlier may result in an inaccurate estimate
- Linear estimate
  - Minimizes sum of squared error and, in general, outliers have large error relative to inliers
  - Resulting estimate may not be near global minimum
- Nonlinear estimate
  - Cost function minimizes sum of squared error, so may not converge to accurate solution
    - Alternatively, use a robust cost function

Random sample consensus (RANSAC) and M-estimator sample consensus (MSAC)

Objective is to determine the consensus set with the minimum cost

tol is the tolerance for establishing datum/model compatibility
τ_{cost} is the upper bound on the cost of an acceptable consensus set
maxTrials is the maximum number of attempts to find a consensus set

$\text{consensus}_{\text{cost}} = \infty$
for (trials $= 0$; trials $<$ maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)
    Select a random sample of unique data points
    Calculate the model using the random sample
    Calculate the error for each data point using model
    Calculate the cost
    if cost $< \text{consensus}_{\text{cost}}$
        $\text{consensus}_{\text{cost}} = \text{cost}$
        $\text{consensus}_{\text{model}} = \text{model}$
Calculate the error for each data point using $\text{consensus}_{\text{model}}$
Calculate the set of inliers (i.e., data points with error $\leq$ tol)

If multiple solutions, try each one, but it is still a single trial

Outlier rejection

- RANSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? 0 : 1
```

- MSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? error[n] : tol
```

RANSAC and MSAC

- Maximum number of trials
  - Adaptive maximum number of trials
- The tolerance for establishing datum/model compatibility

RANSAC and MSAC

- Maximum number of trials

Sample size s
Assumed probability p that at least one of the random samples does not contain any outliers (usually p is chosen to be 0.99)
Probability w that a data point is an inlier

$$w = \frac{\text{total number of inliers}}{\text{total number of data points}}$$

Probability $\epsilon = 1 - w$ that a data point is an outlier
Then, at least N trials are required, where

$$(1 - w^s)^N = 1 - p, \text{ solve for } N$$

$$N = \frac{\log(1-p)}{\log(1-w^s)}$$

The smaller the sample size, the smaller maximum number of trials. Use minimum solution!

RANSAC and MSAC

- Maximum number of trials

The smaller the sample size, the smaller maximum number of trials. Use minimum solution!

Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Table 4.3. *The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers, ϵ.*

Adaptive maximum number of trials

s is the sample size

p is the assumed probability that at least one of the random samples does not contain any outliers

tol is the tolerance for establishing datum/model compatibility

τ_{cost} is the upper bound on the cost of an acceptable consensus set

$\text{maxTrials} = \infty$

$\text{consensus}_{\text{cost}} = \infty$

for (trials = 0; trials < maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)

- Select a random sample of unique data points
- Calculate the model using the random sample
- Calculate the error for each data point using model
- Calculate the cost
- if cost < $\text{consensus}_{\text{cost}}$
  - $\text{consensus}_{\text{cost}} = \text{cost}$
  - $\text{consensus}_{\text{model}} = \text{model}$
  - Calculate the number of inliers
  - $w = \dfrac{\text{number of inliers}}{\text{total number of data points}}$
  - $\text{maxTrials} = \dfrac{\log(1-p)}{\log(1-w^s)}$

If multiple solutions, try each one, but it is still a single trial

Calculate the error for each data point using $\text{consensus}_{\text{model}}$

Calculate the set of inliers (i.e., data points with error $\leq$ tol)

RANSAC and MSAC

- The tolerance for establishing datum/model compatibility

Probability α that a data point is an inlier (usually α is chosen to be 0.95)
Variance σ^2 of the measurement error (if unkonwn, assumed to be 1)
Codimension m
Tolerance is square distance $t^2 = F_m^{-1}(\alpha)\sigma^2$, where $F_m^{-1}(\alpha)$ is the inverse chi-squared cumulative distribution function with m degress of freedom at the probability α

- Codimension

If w is a subspace of a finite-dimensional vector space v, then $\text{codim}(w) = \dim(v) - \dim(w)$
The normalized camera projection matrix is a variety of dimension 3 in $(\hat{\tilde{x}}, \hat{\tilde{y}}, \tilde{X}, \tilde{Y}, \tilde{Z}) \in \mathbb{R}^5$, so has codimension 2

Estimation of normalized camera projection matrix (i.e., camera pose)

1. Outlier rejection using MSAC
2. Linear estimate using inlier correspondences
3. Nonlinear estimate using inlier correspondences

Next lecture

- Camera rotation, two views
- Estimation of rotation matrix (calibrated camera)
- Imaging a plane, two views
- Reading
  - Section 8.4
  - Section 13.1