Today

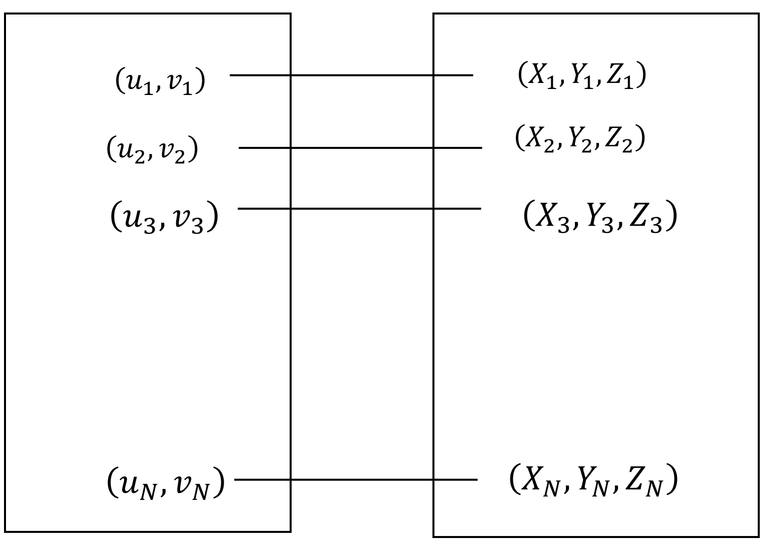
Camera calibration

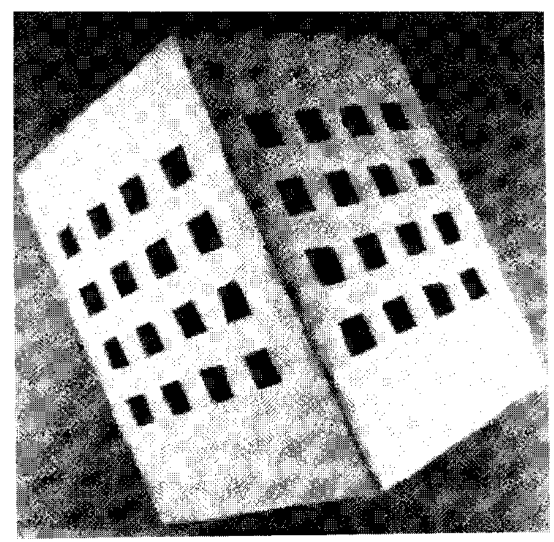
Conventional Camera Calibration

Input: pairs of 3D object points and 2D image points

General strategy:

- Take pictures from calibration patterns with known coordinates of 3D features (e.g., points, lines, or curves)
- Identify the corresponding image features
- obtain the projection matrix by minimizing projection error
- obtain intrinsic parameters from the projection matrix





Camera Calibration from Projection Matrix P: A Linear Method

For every pair of 3D-2D points, we have a pair of linear equations:

$$\mathbf{p}_1^t \mathbf{M}'^{(W)} + p_{14} - \mathbf{p}_3^t \mathbf{M}'^{(W)} u^{(I)} - p_{34} u^{(I)} = 0$$
$$\mathbf{p}_2^t \mathbf{M}'^{(W)} + p_{24} - \mathbf{p}_3^t \mathbf{M}'^{(W)} v^{(I)} - p_{34} v^{(I)} = 0$$

We can setup a linear equation system for N corresponding pairs with 2N equations:

$$\mathbf{A}\mathbf{v} = 0$$

A is a $2N \times 12$ matrix containing the known

v is a 12×1 vector containing the unknown

$$\mathbf{v} = (\mathbf{p}_1^t \quad p_{14} \quad \mathbf{p}_2^t \quad p_{24} \quad \mathbf{p}_3^t \quad p_{34})^t$$

When rank(**A**) = 11, **v** can be estimated by performing SVD on **A**

v is the last column of the **V** matrix with an unknown scalar

Issues in the Linear Method

- Numerical issue
- Outliers in observations

Numerical Stability: Data Normalization

The entries in **A** have large difference in unit and scale

$$\mathbf{A} = \begin{bmatrix} \left(\mathbf{M'}_1{}^{(W)}\right)^t & 1 & 0_{1\times3} & 0 & -\left(\mathbf{M'}_1{}^{(W)}\right)^t u_1{}^{(I)} & -u_1{}^{(I)} \\ 0_{1\times3} & 0 & \left(\mathbf{M'}_1{}^{(W)}\right)^t & 1 & -\left(\mathbf{M'}_1{}^{(W)}\right)^t v_1{}^{(I)} & -v_1{}^{(I)} \\ \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\ \left(\mathbf{M'}_N{}^{(W)}\right)^t & 1 & 0_{1\times3} & 0 & -\left(\mathbf{M'}_N{}^{(W)}\right)^t u_N{}^{(I)} & -u_N{}^{(I)} \\ 0_{1\times3} & 0 & \left(\mathbf{M'}_N{}^{(W)}\right)^t & 1 & -\left(\mathbf{M'}_N{}^{(W)}\right)^t v_N{}^{(I)} & -v_N{}^{(I)} \end{bmatrix}$$

Unit: mm
scale: vary

Unit: no
Scale: 0 or 1

Unit: mm*pixel
scale: vary

Unit: pixel
scale: the image size

- Numerical instable: resulting problem when data contains significant noise
- **Solution: data normalization**

Numerical Stability: Data Normalization

Replace **A** by **A**’ by centering and normalizing each point, respectively.

- **Centering**: translate 2D/3D points w.r.t. their centroid
- **Normalization**: scaling the coordinates by a scale factor
  - E.g., the scale factor can be computed as the inverse of average distance to the centroid

Numerical Stability: Data Normalization

Replace **A** by **A**' by centering and normalizing each point, respectively.

- **Centering**: translate 2D/3D points w.r.t. their centroid
- **Normalization**: scaling the coordinates by a scale factor
  - E.g., the scale factor can be computed as the inverse of average distance to the centroid

$$\mathbf{m}' = \mathbf{H}_{2D}\,\mathbf{m}$$

$$\mathbf{H}_{2D} = \begin{bmatrix} \frac{1}{s_{2D}} & 0 & -\frac{\overline{m_x}}{s_{2D}} \\ 0 & \frac{1}{s_{2D}} & -\frac{\overline{m_y}}{s_{2D}} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}' = \mathbf{H}_{3D}\,\mathbf{M}$$

$$\mathbf{H}_{3D} = \begin{bmatrix} \frac{1}{s_{3D}} & 0 & 0 & -\frac{\overline{M_X}}{s_{3D}} \\ 0 & \frac{1}{s_{3D}} & 0 & -\frac{\overline{M_Y}}{s_{3D}} \\ 0 & 0 & \frac{1}{s_{3D}} & -\frac{\overline{M_Z}}{s_{3D}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Numerical Stability: Data Normalization

- Build the linear equation system using the normalized 2D/3D points
- Solve the problem of $\mathbf{A}'\mathbf{v}' = 0$ and recover $\mathbf{P}'$ from $\mathbf{v}'$
- Recover the original projection matrix **P**

How to recover P?

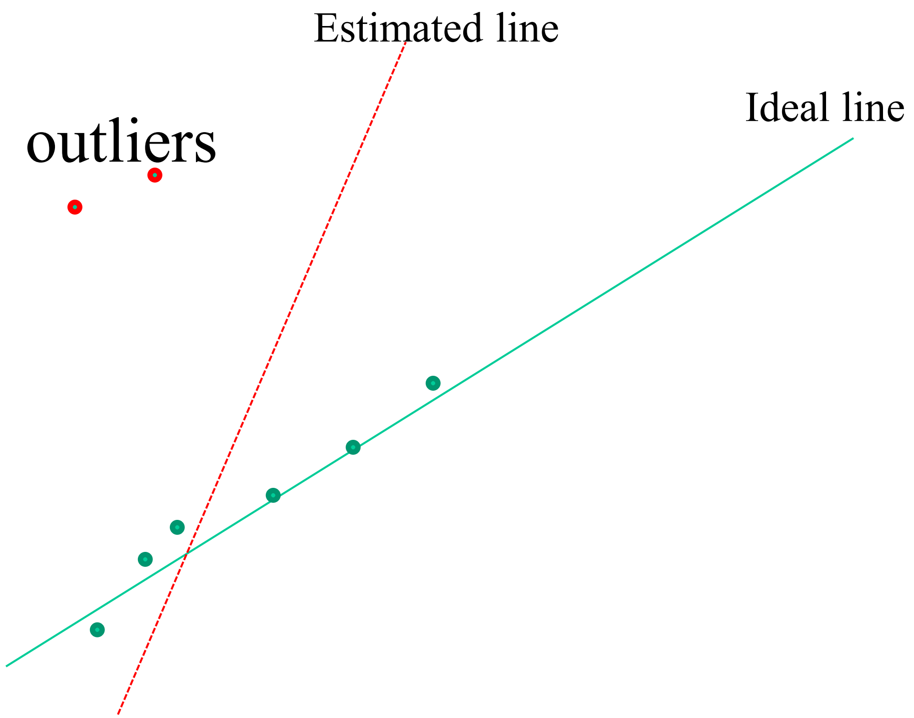
How to Recover P after Data Normalization?

$$P = {H_{2D}}^{-1} P' H_{3D}$$

Note: this preprocessing step is recommended for all methods involving image data

Robust Linear Method with RANSAC

The linear method is sensitive to image noise and localization errors.



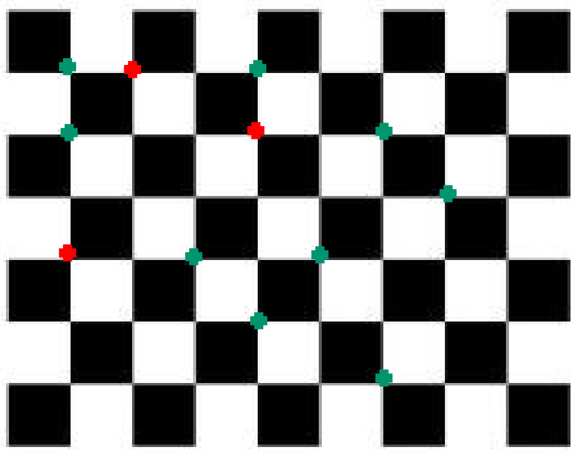
Robust Linear Method with RANSAC

A better solution can be obtained by using a robust method such as the RANSAC (Random Sample Consensus) method.

Assumptions:

- The model can be estimated from K data items
- There are N (N>K) data items in total
- The whole data set contains a fraction α of outliers
- A guess about the probability that the training set does not contain any outliers

Robust Linear Method with RANSAC



- Use green points to estimate the parameters
- Use red points to compute errors

Robust Linear Method with RANSAC

Repeat:

Step 1: randomly partition all N points into a training set (with K>6 points) and a testing set (with N-K points).

Step 2: compute the projection matrix **P** using the training set

Step 3: for each point in the testing set, compute its projection error $\varepsilon = \|\mathbf{A}_i \mathbf{v}\|$ if $\varepsilon < Threshold$, increment a counter for "inliers"

How to determine?

Until run sufficient times

Step 4: use the training set resulting the highest number of inliers plus all the inlier points to recompute **P**

Robust Linear Method with RANSAC

- Assume the probability that a data sample is an outlier is η
- The probability that all *K* data in a subset are good is

$$(1-\eta)^K$$

- The probability that all s different subsets/iterations will contain at least one or more outliers is

$$(1-(1-\eta)^K)^s$$

- The probability that at least one random subset has no outliers is given by $P = 1-(1-(1-\eta)^K)^s$

- The number of iterations/subsets needed is

$$\Rightarrow \quad s = \frac{\ln(1-P)}{\ln(1-(1-\eta)^K)}$$

Estimate Camera Parameters from P

sign(α)

$$\mathbf{P} = \begin{bmatrix} f_u \mathbf{r}_1^t + u_0 \mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v \mathbf{r}_2^t + v_0 \mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix} = \zeta \begin{bmatrix} \mathbf{p}_1^t & p_{14} \\ \mathbf{p}_2^t & p_{24} \\ \mathbf{p}_3^t & p_{34} \end{bmatrix}$$

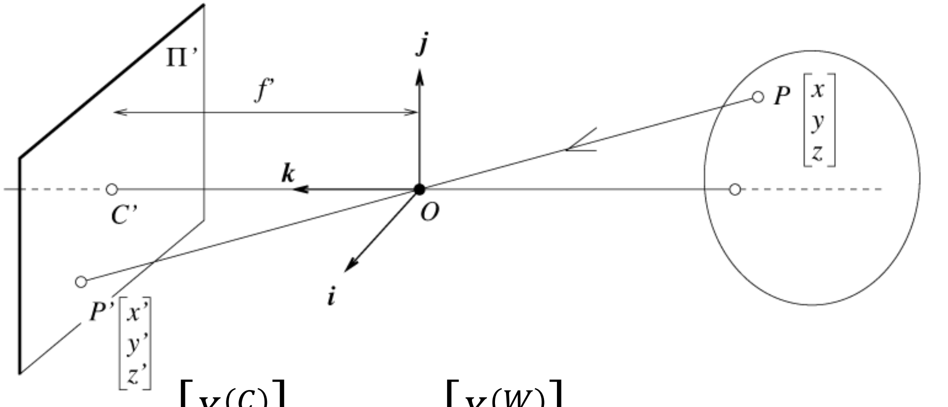
$$t_z = \zeta p_{34}$$

How to determine ζ

ζ is determined from t_Z

t_Z is negative if the object is before the camera

So if $p_{34} > 0, \zeta = -1$;
Otherwise $\zeta = 1$



$$\begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \end{bmatrix} = \mathbf{R}_{3\times3} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \end{bmatrix} + \mathbf{T}_{3\times1}$$

Estimate Camera Parameters from P

$$\mathbf{P} = \begin{bmatrix} f_u\mathbf{r}_1^t + u_0\mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v\mathbf{r}_2^t + v_0\mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix} = \zeta \begin{bmatrix} \mathbf{p}_1^t & p_{14} \\ \mathbf{p}_2^t & p_{24} \\ \mathbf{p}_3^t & p_{34} \end{bmatrix}$$

$$\mathbf{r}_3 = \mathbf{p}_3$$

$u_0 = \mathbf{p}_1^t\mathbf{p}_3$ $\qquad$ $v_0 = \mathbf{p}_2^t\mathbf{p}_3$ $\qquad$ Why?

Estimate Camera Parameters from P

$$\mathbf{P} = \begin{bmatrix} f_u \mathbf{r}_1^t + u_0 \mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v \mathbf{r}_2^t + v_0 \mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix} = \zeta \begin{bmatrix} \mathbf{p}_1^t & p_{14} \\ \mathbf{p}_2^t & p_{24} \\ \mathbf{p}_3^t & p_{34} \end{bmatrix}$$

$$f_u = \sqrt{\mathbf{p}_1^t \mathbf{p}_1 - u_0^2} \qquad f_v = \sqrt{\mathbf{p}_2^t \mathbf{p}_2 - v_0^2}$$

$$t_X = \zeta(p_{14} - u_0 p_{34})/f_u \qquad t_Y = \zeta(p_{24} - v_0 p_{34})/f_v$$

$$\mathbf{r}_1 = \zeta(\mathbf{p}_1 - u_0 \mathbf{p}_3)/f_u \qquad \mathbf{r}_2 = \zeta(\mathbf{p}_2 - v_0 \mathbf{p}_3)/f_v$$

Announcement

Project 1 has been posted in Blackboard and is due 11:59pm EST, Sunday, Feb 19th, 2023.

Requirement: you need to submit your project as **a SINGLE zipped file named as Lastname-Firstname-Proj1** through **Blackboard** including

- A written project report includes a brief introduction on the addressed problem, a succinct description on the methods you implemented with the major steps, the experimental results and analysis, conclusion, and reference.

- Code with appropriate comments.

Your report must be well organized and be easy to follow. There should be no spelling and grammar errors!

Written Report

The experimental results must include

- The experimental setting, e.g., the number of points used for calibration, the number of points used for calculating the projection error, and the number of iterations used in RANSAC
- The estimated camera parameters
- The calibration performance in terms of projection error

For analysis, you may consider what happen if you change the experimental settings.

Project 1

- 2Dpoints.txt
- 3Dpoints_part1.txt
- 3Dpoints_part2.txt