CS 534: Computer Vision
Camera Calibration

Ahmed Elgammal
Dept of Computer Science
Rutgers University

Outlines

- Camera Calibration
- Linear Least-Squares and related stuff

Camera Calibration, Why ?

- Camera calibration is a necessary step in 3D computer vision.
- A calibrated camera can be used as a quantitative sensor
- It is essential in many applications to recover 3D quantitative measures about the observed scene from 2D images. Such as 3D Euclidean structure
- From a calibrated camera we can measure how far an object is from the camera, or the height of the object, etc. e.g., object avoidance in robot navigation.

Camera Calibration

- Find the intrinsic and extrinsic parameters of a camera
  - Extrinsic parameters: the camera's location and orientation in the world.
  - Intrinsic parameters: the relationships between pixel coordinates and camera coordinates.
- VERY large literature on the subject
- Work of Roger Tsai is influential
- Good calibration is important when we need to:
  - Reconstruct a world model.
  - Interact with the world: Robot, hand-eye coordination
- Basic idea:
  - Given a set of world points P_i and their image coordinates (u_i, v_i)
  - find the projection matrix M
  - and then find intrinsic and extrinsic parameters.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} K_{3\times3} & 0_{3\times1} \end{bmatrix} \begin{pmatrix} {}^C_W R_{3\times3} & {}^C O_{W\,3\times1} \\ 0^T{}_{1\times3} & 1 \end{pmatrix}_{4\times4} \begin{pmatrix} {}^W P \\ 1 \end{pmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{K_{3\times3}}_{\textit{Intrinsic parameters}} \underbrace{\left(\underbrace{{}^C_W R_{3\times3}}_{\textit{Rotation}} \quad \underbrace{{}^C O_{W\,3\times1}}_{\textit{Translation}} \right)_{3\times4}}_{\textit{Extrinsic parameters}} \begin{pmatrix} {}^W P \\ 1 \end{pmatrix} \Rightarrow p = \underbrace{M}_{\textit{Projection matrix}} P$$

CS 534 – Calibration - 5

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{K_{3\times3}}_{\textit{Intrinsic parameters}} \underbrace{\left(\underbrace{{}^C_W R_{3\times3}}_{\textit{Rotation}} \quad \underbrace{{}^C O_{W\,3\times1}}_{\textit{Translation}} \right)_{3\times4}}_{\textit{Extrinsic parameters}} \begin{pmatrix} {}^W P \\ 1 \end{pmatrix} \Rightarrow p = \underbrace{M}_{\substack{\textit{Projection matrix} \\ \text{3x4}}} P$$

$$M = K_{3\times3} \left({}^C_W R_{3\times3} \quad {}^C O_{W\,3\times1} \right)_{3\times4} = K[r_1 \quad r_2 \quad r_3 \quad t]$$

Only 11 free parameters (not 12):
5 intrinsic, 3 for rotation, 3 for translation
Perspective Projection Matrix

CS 534 – Calibration - 6

Explicit Form of the Projection Matrix

$$\mathcal{M} = \begin{pmatrix} \boxed{\begin{matrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{matrix}} & \boxed{\begin{matrix} \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ t_z \end{matrix}} \end{pmatrix}$$

Note: If $\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b})$ then $|\boldsymbol{a}_3| = 1$.

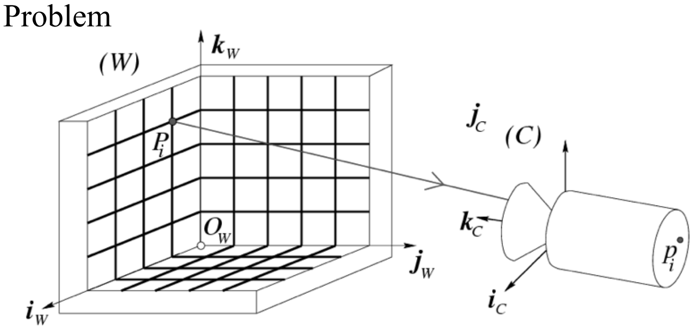
$$p = MP = \begin{pmatrix} m_1^T \\ m_2^T \\ m_3^T \end{pmatrix} P \quad \Rightarrow \quad u = \frac{\boldsymbol{m}_1^T \cdot \boldsymbol{P}}{\boldsymbol{m}_3^T \cdot \boldsymbol{P}} \qquad v = \frac{\boldsymbol{m}_2^T \cdot \boldsymbol{P}}{\boldsymbol{m}_3^T \cdot \boldsymbol{P}}$$

Replacing *M* by λM doesn't change *u* or *v*

M is only defined up to scale in this setting

CS 534 – Calibration - 7

Calibration Problem



Given n points $P_1, \dots, P_n$ with *known* positions and their images $p_1, \dots, p_n$

Find $\boldsymbol{i}$ and $\boldsymbol{e}$ such that

$$\begin{cases} u_i = \dfrac{\boldsymbol{m}_1(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \\ \\ v_i = \dfrac{\boldsymbol{m}_2(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \end{cases} \quad \text{for} \quad i = 1, \dots, n$$

i intrinsic parameters
e extrinsic parameters

$$\Rightarrow \sum_{i=1}^{n} \left[\left(u_i - \frac{\boldsymbol{m}_1(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 + \left(v_i - \frac{\boldsymbol{m}_2(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 \right] \quad \text{is minimized}$$

CS 534 – Calibration - 8

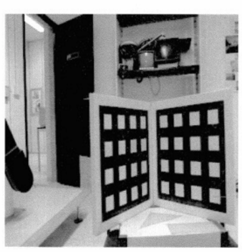
Calibration Techniques

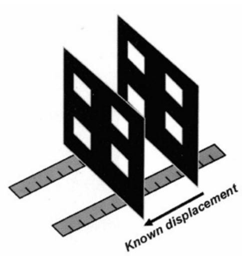
- Calibration using 3D calibration object
- Calibration using 2D planer pattern
- Calibration using 1D object (line-based calibration)
- Self Calibration: no calibration objects
- Vanishing points from for orthogonal direction
- Many other smart ideas

Calibration Techniques

Calibration using 3D calibration object:

- Calibration is performed by observing a calibration object whose geometry in 3D space is known with very good precision.
- Calibration object usually consists of two or three planes orthogonal to each other, e.g. calibration cube
- Calibration can also be done with a plane undergoing a precisely known translation (Tsai approach)
- (+) most accurate calibration, simple theory
- (-) more expensive, more elaborate setup

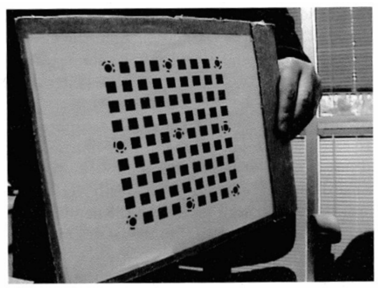




Calibration Techniques

2D plane-based calibration

- Require observation of a planar pattern shown at few different orientations
- No need to know the plane motion
- Set up is easy, most popular approach.



Calibration Techniques

1D line-based calibration:

- Relatively new technique.
- Calibration object is a set of collinear points, e.g., two points with known distance, three collinear points with known distances, four or more…
- Camera can be calibrated by observing a moving line around a fixed point, e.g. a string of balls hanging from the ceiling!
- Can be used to calibrate multiple cameras at once. Good for network of cameras.

Calibration Techniques

Self-calibration:

- Techniques in this category do not use any calibration object
- Only image point correspondences are required
- Just move the camera in a static scene and obtain multiple images
- Correspondences between three images are sufficient to recover both the internal and external parameters

Calibration Techniques

Which calibration technique to use ?

- Calibration with apparatus versus self-calibration:
  - Whenever possible, if you can pre-calibrate the camera, you should do it with a calibration object
  - Self-calibration cannot usually achieve the same accuracy as calibration with an object.
  - Sometimes pre-calibration is impossible (e.g., a scene reconstruction from an old movie), self-calibration is the only choice.

Calibration Techniques

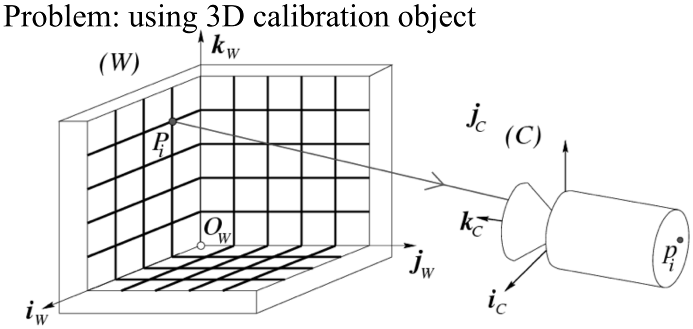
Which calibration technique to use ?

- using 3D calibration object
- using 2D planer pattern
- using 1D object (line-based calibration)
- Self calibration

More Accurate ↑ Less work ↓

- 2D planer pattern approaches seems to be a good compromise: good accuracy with simple setup
- 1D object is suitable for calibrating multiple cameras as once.

Calibration Problem: using 3D calibration object



Given n points $P_1, \dots, P_n$ with *known* positions and their images $p_1, \dots, p_n$

Find $\boldsymbol{i}$ and $\boldsymbol{e}$ such that

$$\begin{cases} u_i = \dfrac{\boldsymbol{m}_1(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \\ v_i = \dfrac{\boldsymbol{m}_2(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \end{cases} \quad \text{for} \quad i = 1, \dots, n$$

i intrinsic parameters
e extrinsic parameters

$$\sum_{i=1}^{n} \left[\left(u_i - \frac{\boldsymbol{m}_1(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 + \left(v_i - \frac{\boldsymbol{m}_2(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i},\boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 \right] \quad \text{is minimized}$$

Camera Calibration and least-squares

- Camera Calibration can be posed as least-squares parameter estimation problem.
- Estimate the intrinsic and extrinsic parameters that minimize the mean-squared deviation between predicted and observed image features.
- Least-squares parameter estimation is a fundamental technique that is used extensively in computer vision.
- You can formulate many problems as error minimization between observed and predicted values.
  - Linear Least-Squares
  - Nonlinear Least-Squares
- Typically, use linear least-squares to obtain a solution and then use nonlinear least square to refine the solution and estimate more parameters (e.g. radial distortion)

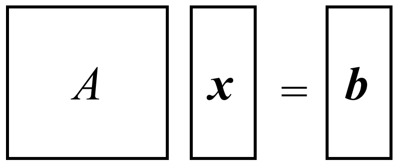
Linear Least-Squares

- Linear system of equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \quad \cdots \quad + a_{1q}x_q &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \quad \cdots \quad + a_{2q}x_q &= b_2 \\ \cdots & \\ a_{p1}x_1 + a_{p2}x_2 + \quad \cdots \quad + a_{pq}x_q &= b_p \end{aligned} \quad \Leftrightarrow \quad Ax = b$$

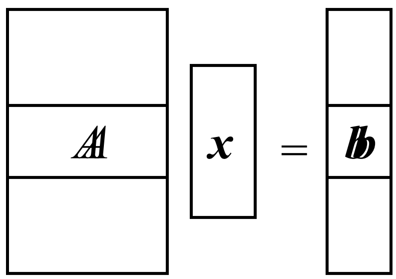
$$A_{p\times q} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & \cdots & a_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p1} & a_{p2} & \cdots & a_{pq} \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}$$

Linear Systems



Square system:

- unique solution
- Gaussian elimination



Rectangular system ??

- under-constrained: infinity of solutions
- over-constrained: no solution

Minimize $||Ax-b||^2$

CS 534 – Calibration - 19

How do you solve overconstrained linear equations ??

- Over-constrained system, obtain a least-squared solution:

Solution that minimize the squared deviations between *Ax* and *b*:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1q}x_q &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2q}x_q &= b_2 \\
\cdots & \\
a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{pq}x_q &= b_p
\end{aligned}
$$

$$E = \sum_{i=1}^{p} (a_{i1}x_1 + \cdots + a_{iq}x_q - b_i)^2 = |Ax - b|^2 = (Ax - b)^T \cdot (Ax - b) = e^T \cdot e$$

$$\frac{\partial E}{\partial x_i} = 0 \qquad i = 1, \cdots, q$$

CS 534 – Calibration - 20

How do you solve overconstrained linear equations ??

- Define $E = |\boldsymbol{e}|^2 = \boldsymbol{e} \cdot \boldsymbol{e}$ with

$$\boldsymbol{e} = A\boldsymbol{x} - \boldsymbol{b} = \left[\boldsymbol{c}_1 \middle| \boldsymbol{c}_2 \middle| \dots \middle| \boldsymbol{c}_n \right] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - \boldsymbol{b}$$

$$= x_1\boldsymbol{c}_1 + x_2\boldsymbol{c}_2 + \cdots x_n\boldsymbol{c}_n - \boldsymbol{b}$$

$$\frac{\partial e}{\partial x_i} = c_i$$

- At a minimum,

$$\frac{\partial E}{\partial x_i} = \frac{\partial \boldsymbol{e}}{\partial x_i} \cdot \boldsymbol{e} + \boldsymbol{e} \cdot \frac{\partial \boldsymbol{e}}{\partial x_i} = 2\frac{\partial \boldsymbol{e}}{\partial x_i} \cdot \boldsymbol{e}$$

$$= 2\frac{\partial}{\partial x_i}(x_1\boldsymbol{c}_1 + \cdots + x_n\boldsymbol{c}_n - \boldsymbol{b}) \cdot \boldsymbol{e} = 2\boldsymbol{c}_i \cdot \boldsymbol{e}$$

$$= 2\boldsymbol{c}_i^T(A\boldsymbol{x} - \boldsymbol{b}) = 0$$

- or

$$0 = \begin{bmatrix} \boldsymbol{c}_i^T \\ \vdots \\ \boldsymbol{c}_n^T \end{bmatrix} (A\boldsymbol{x} - \boldsymbol{b}) = A^T(A\boldsymbol{x} - \boldsymbol{b}) \Rightarrow A^T A\boldsymbol{x} = A^T\boldsymbol{b},$$

where $\boxed{\boldsymbol{x} = A^\dagger\boldsymbol{b} \text{ and } A^\dagger = (A^T A)^{-1}A^T \text{ is the } \textit{pseudoinverse} \text{ of } A\ !}$

Singular Value Decomposition

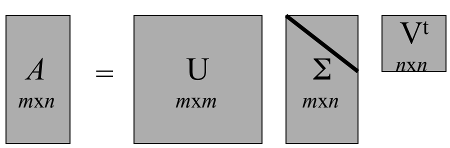
- Chief tool for dealing with *m* by *n* systems and singular systems.
- SVD: If **A** is a real *m* by *n* matrix then there exist orthogonal matrices

U ($m \cdot m$) and **V** ($n \cdot n$) such that

$\mathbf{U^t A V} = \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p)$ $\quad p = \min\{m,n\}$

$$\mathbf{U^t A V} = \Sigma \qquad\qquad \mathbf{A} = \mathbf{U}\Sigma\,\mathbf{V^t}$$

- U and V are orthonormal matercies



Singular Value Decomposition

- SVD: If **A** is a real *m* by *n* matrix then there exist orthogonal matrices

U ($m \cdot m$) and **V** ($n \cdot n$) such that $\mathbf{U}^t\mathbf{A}\mathbf{V} = \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$ $p = \min\{m,n\}$

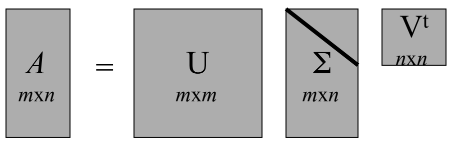
$\mathbf{U}^t\mathbf{A}\mathbf{V} = \Sigma$ $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^t$

- **Singular values**: Non negative square roots of the eigenvalues of $\mathbf{A}^t\mathbf{A}$. Denoted σ_i, $i=1,\ldots,n$
- $\mathbf{A}^t\mathbf{A}$ is symmetric ⇒ eigenvalues and singular values are real.
- Singular values arranged in decreasing order.
- Columns of V are the eigenvectors of A^tA

$$A^t A = (U \sum V^t)^t (U \sum V^t) = V \sum^t U^t U \sum V^t = V \sum^t \sum V^t = V \sum^2 V^{-1}$$

$$(A^t A)V = V \sum^2$$

$$(A^t A)v = v\lambda$$



- For over-constrained system $Ax=b$:
- Use SVD to decompose $A=U\Sigma V^t$

$$Ax = b$$

$$U \sum V^T x = b$$

$$(V \sum^{-1} U^T) U \sum V^T x = (V \sum^{-1} U^T) b$$

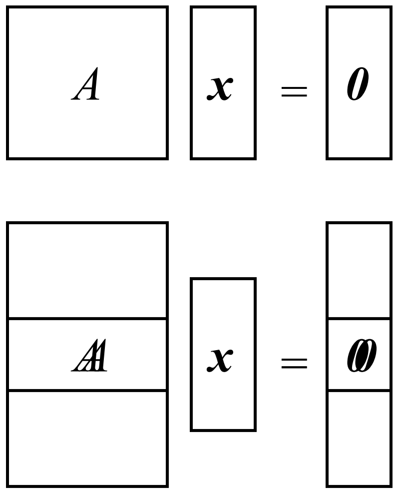
$$x = V \sum^{-1} U^T b$$

$$\sum^{-1} = diag(1/\sigma_1, 1/\sigma_2, \cdots, 1/\sigma_n)$$

For rank deficient matrices: we can obtain best rank-R approximation by sitting the inverse of the smallest n-R singular values to zeros

$$\sum^{-1} = diag(1/\sigma_1, 1/\sigma_2, \cdots, 1/\sigma_R, 0, \ldots, 0)$$

Homogeneous Linear Systems



Square system:

- unique solution: 0
- unless Det(A)=0

Rectangular system ??

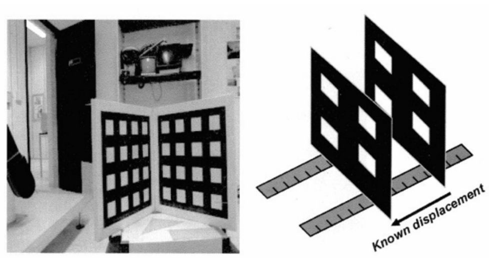
Minimize $|Ax|^2$
under the constraint $|x|^2 = 1$

x is the unit singular vector of $\boldsymbol{A}$ corresponding to the smallest singular value (the last column of $\boldsymbol{V}$, where $\boldsymbol{A} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$ is the SVD of $\boldsymbol{A}$)

CS 534 – Calibration - 25

Calibration using 3D calibration object:

- Calibration is performed by observing a calibration object whose geometry in 3D space is known with very good precision.
- Calibration object usually consists of two or three planes orthogonal to each other, e.g. calibration cube
- Calibration can also be done with a plane undergoing a precisely known translation
- (+) most accurate calibration, simple theory
- (-) more expensive, more elaborate setup



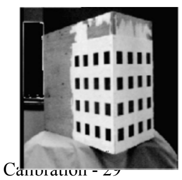
CS 534 – Calibration - 28

Calibration using 3D object

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target
- We position camera in front of target and find images of corners
- We obtain equations that describe point coordinates and contain intrinsic and extrinsic parameters of camera

Main Steps:

1. Detecting points of interest (e.g., corners of the checker pattern) in the 2D image and obtain their corresponding 3D measurement.
2. Find the best projection matrix M using linear least squares
3. Recover intrinsic and extrinsic parameters
4. Refine the parameters through nonlinear optimization



Linear Camera Calibration

Given n points $P_1, \dots, P_n$ with *known* positions and their images $p_1, \dots, p_n$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \end{pmatrix} \Longleftrightarrow \begin{pmatrix} \boldsymbol{m}_1 - u_i \boldsymbol{m}_3 \\ \boldsymbol{m}_2 - v_i \boldsymbol{m}_3 \end{pmatrix} \boldsymbol{P}_i = 0$$

$$m_{11}X_i + m_{12}Y_i + m_{13}Z_i + m_{14} \quad - m_{31}u_iX_i - m_{32}u_iY_i - m_{33}u_iZ_i - m_{34}u_i = 0$$

$$m_{21}X_i + m_{22}Y_i + m_{23}Z_i + m_{24} \quad - m_{31}v_iX_i - m_{32}v_iY_i - m_{33}v_iZ_i - m_{34}v_i = 0$$

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1\boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n\boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix}}_{2n\times 12} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix}}_{12\times 1}$$

Observations:

- This is a linear problem with respect to M but nonlinear with respect to the parameters (intrinsic and extrinsic)
- Each 3D-2D point match provides 2 equations
- We need at least 6 points in general configuration (12 equation for 11 parameters)

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1\boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n\boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix}}_{2n\text{x}12} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix}}_{12\text{x}1}$$

CS 534 – Calibration - 31

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1\boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n\boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix}}_{2n\text{x}12} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \underbrace{\begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix}}_{12\text{x}1}$$

Problem: $\min_m \|\mathcal{P}m\|^2 = 0$ Subject to: $\|m\| = 1$

- When $n \geq 6 \Rightarrow$ homogenous linear least-squares can be used to compute unite vector $m \Rightarrow M$
- *Solution:* The eigenvector of $\mathcal{P}^T\mathcal{P}$ corresponding to the smallest eigen value
- Degenerate case: All points lie on the same plane.
  Points should not be on the same plane
- When possible, have at least 5 times as many equations as unknowns (28 points)

CS 534 – Calibration - 32

Once *M* is known, you still got to recover the intrinsic and extrinsic parameters !!!

This is a decomposition problem, not an estimation problem.

$$\boxed{\rho}\ \mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}$$

- Intrinsic parameters
- Extrinsic parameters

CS 534 – Calibration - 33

- Once the projection matrix M is known, we can uniquely recover the intrinsic and extrinsic parameters of the camera

$$M = K[R \quad t]$$
$$B = KR$$
$$b = Kt$$

$$K = \begin{bmatrix} \alpha & \gamma & u_o \\ 0 & \beta & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

$$BB^T = KK^T = \begin{bmatrix} k_u & k_c & u_o \\ k_c & k_v & v_o \\ u_o & v_o & 1 \end{bmatrix} = \begin{bmatrix} \alpha^2 + \gamma^2 + u_o^2 & u_o v_o + \gamma\beta & u_o \\ u_o v_o + \gamma\beta & \beta^2 + v_o^2 & v_o \\ u_o & v_o & 1 \end{bmatrix}$$

$$\beta = \sqrt{k_v - v_o^2},$$
$$\gamma = \frac{k_c - u_o v_o}{\beta}$$
$$\alpha = \sqrt{k_u - u_o^2 - \gamma^2}$$

$$R = K^{-1}B,$$
$$t = K^{-1}b$$

CS 534 – Calibration - 34

Another approach to decomposition

$$ {}^{B}P = {}^{B}_{A}R \, {}^{A}P + {}^{B}O_{A} $$

$$ {}^{W}O_{W} = {}^{W}_{C}R \, {}^{C}O_{W} + {}^{W}O_{C} $$

$$ 0 = {}^{W}_{C}R \, {}^{C}O_{W} + {}^{W}O_{C} $$

$$ {}^{W}_{C}R \, {}^{C}O_{W} = -{}^{W}O_{C} $$

$$ {}^{C}O_{W} = -{}^{C}_{W}R \, {}^{W}O_{C} $$

$$ \begin{bmatrix} u \\ v \\ w \end{bmatrix} = K_{3\times3} \left({}^{C}_{W}R_{3\times3} \quad {}^{C}O_{W\,3\times1} \right)_{3\times4} \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix} $$

The translation vector can be written as:

$$ \begin{bmatrix} u \\ v \\ w \end{bmatrix} = K_{3\times3} \left({}^{C}_{W}R_{3\times3} \quad -{}^{C}_{W}R_{3\times3} \, {}^{W}O_{C\,3\times1} \right)_{3\times4} \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix} $$

$$ \begin{bmatrix} u \\ v \\ w \end{bmatrix} = K_{3\times3} \, {}^{C}_{W}R_{3\times3} \left(I_{3\times3} \quad -{}^{W}O_{C\,3\times1} \right)_{3\times4} \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix} $$

Getting Camera Translation

$$ M_{3\times4} = K_{3\times3} \, {}^{C}_{W}R_{3\times3} \left(I_{3\times3} \quad -{}^{W}O_{C\,3\times1} \right)_{3\times4} $$

Upper triangle matrix

Orthogonal matrix

Translation Vector: Where in the world is the camera center

Note that:
$$ \begin{pmatrix} 1 & 0 & 0 & -Xc \\ 0 & 1 & 0 & -Yc \\ 0 & 0 & 1 & -Zc \end{pmatrix} \begin{pmatrix} Xc \\ Yc \\ Zc \\ 1 \end{pmatrix} = 0 $$

If we solve the system $M \cdot t = 0$ we can get the translation vector! (*t* is a null vector for *M*)

Find null vector ***t*** of ***M*** using SVD

- ***t*** is the unit singular vector of ***M*** corresponding to the smallest singular value (the last column of ***V***, where $\boldsymbol{M} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{T}$ is the SVD of ***M***)

Getting Camera Orientation

$$M_{3\times4} = K_{3\times3}\,{}^{C}_{W}R_{3\times3}\left(I_{3\times3} \quad -{}^{W}O_{C\,3\times1}\right)_{3\times4}$$

Upper triangle matrix

Orthogonal matrix

Translation Vector: Where in the world is the camera center

- Left 3x3 submatrix *M'* of *M* is of form *M'=K R*
- K is an upper triangular matrix
- R is an orthogonal matrix
- Any non-singular square matrix *M'* can be decomposed into the product of an upper-triangular matrix K and an orthogonal matrix R using the RQ factorization
- Similar to QR factorization but order of 2 matrices is reversed A=Q.R

CS 534 – Calibration - 37

Rotation matrices

- 3-D rotation is result of three consecutive rotations around the coordinate axes $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$.
  - the angles of the rotations are the parameters of the rotation

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad R_2(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_3(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_1 R_2 R_3 = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma \_ \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\gamma \end{bmatrix}$$

- The order in which we perform the multiplications matters!
  - Six ways to represent a rotation matrix

CS 534 – Calibration - 38

Rotation matrices

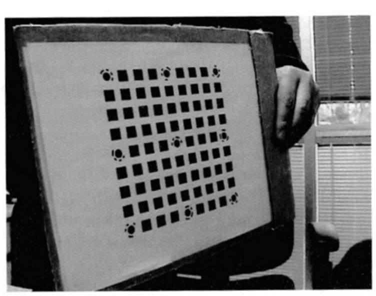
$$R = R_1 R_2 R_3 = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma - \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\gamma \end{bmatrix}$$

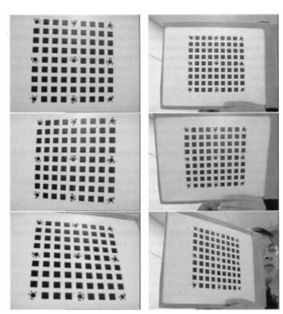
- The order in which we perform the multiplications matters!
  - Six ways to represent a rotation matrix
- It is easy to recover the rotation angles (α,β,γ) from R
  - $R_{1,3}$ gives β
  - Then γ can be recovered from $R_{1,1,}$ and α from $R_{2,3}$
- R is orthornormal $RR^T = R^T R = I$
- So, even though there are 9 entries in R, there are still only the 3 parameters
- This makes estimating R difficult in practice.

Camera Calibration with a 2D planer object

2D plane-based calibration

- Require observation of a planar pattern shown at few different orientations
- No need to know the plane motion
- Set up is easy, just print the calibration pattern an stick it to surface.
- most popular approach.





Camera Calibration with a 2D planer object

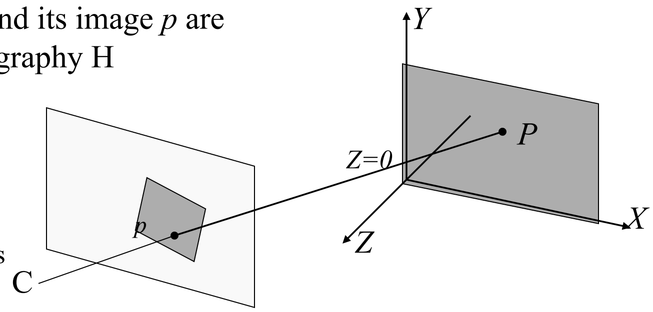
- Assume model plane is on Z=0 of the world coordinate system

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = K\begin{bmatrix}r_1 & r_2 & r_3 & t\end{bmatrix}\begin{bmatrix}X\\Y\\0\\1\end{bmatrix} = K\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}\begin{bmatrix}X\\Y\\1\end{bmatrix} \Rightarrow sp = \mathrm{H}_{3x3}P$$

A model point P and its image p are related by a homography H

$$\mathrm{H} = K\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}$$

H is a 3x3 matrix defined up to a scale factor (only 8 degrees of freedom)



- Given an image of the model plane, a homography can be estimated, Let's denoted by H=[h_1 h_2 h_3]

$$[\mathrm{h}_1 \quad \mathrm{h}_2 \quad \mathrm{h}_3] = \lambda K\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}$$
$$K^{-1}[\mathrm{h}_1 \quad \mathrm{h}_2 \quad \mathrm{h}_3] = \lambda\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}$$
$$K^{-1}\mathrm{h}_1 = \lambda r_1$$
$$K^{-1}\mathrm{h}_2 = \lambda r_2$$

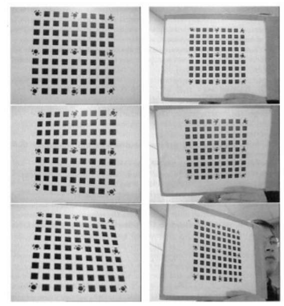
Since r_1 and r_2 are orthonormal, we can obtain the following two important constraints relating K and H

$$h_1^T K^{-T} K^{-1} h_2 = 0$$
$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$

- Given a set of images of the same plane we can establish a set of homographies H_i, one homography from each image.
- Given the constraints between H and K, we can solve for K

$$h_1^T K^{-T} K^{-1} h_2 = 0$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$



- Note $K^{-T}K^{-1}$ is a 3x3 symmetric matrix; we have six unknowns.
- Each homography gives two linear equations in 6 unknowns
- We need at least 3 images of the plane to estimate the parameters in K
- Once K is known, we can estimate the extrinsic parameters:

$$r_1 = \lambda K^{-1}\mathrm{h}_1, \qquad r_2 = \lambda K^{-1}\mathrm{h}_2, \qquad r_3 = r_1 \times r_2, \qquad t = \lambda K^{-1}\mathrm{h}_3$$

Sources

- Forsyth and Ponce, “Computer Vision a Modern approach” 3.1,3.2
- Medioni and Kang “Emerging Topics in Computer Vision” Prentice Hall 2004- Chapter 2 by Zhengyou Zhang.
- Fougeras, “Three-dimensional Computer Vision” MIT press.
- Slides by J. Ponce UIUC