Camera Calibration

CS 4391 Introduction Computer Vision

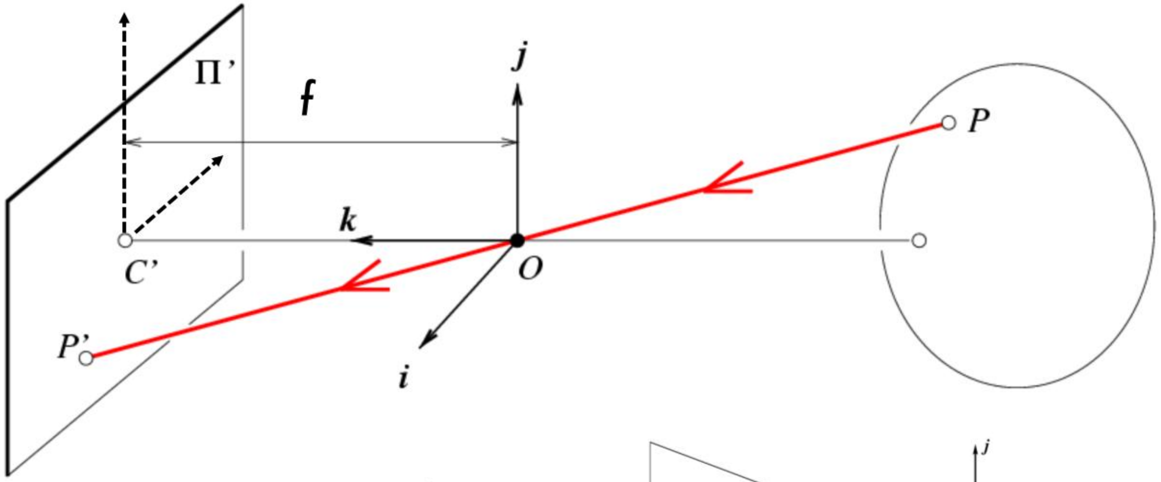
Professor Yu Xiang

The University of Texas at Dallas

Some slides of this lecture are courtesy Silvio Savarese

Recap Camera Models

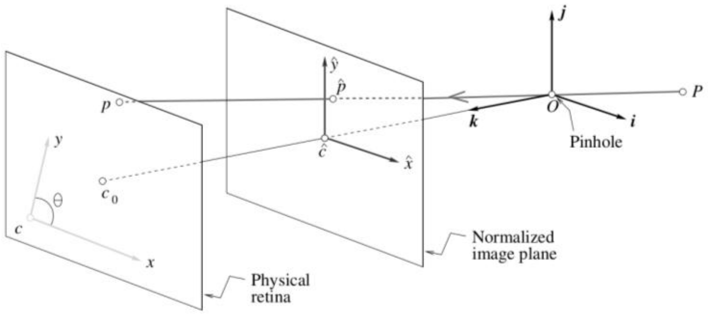
- Camera projection matrix



$$P = K[R|\mathbf{t}]$$

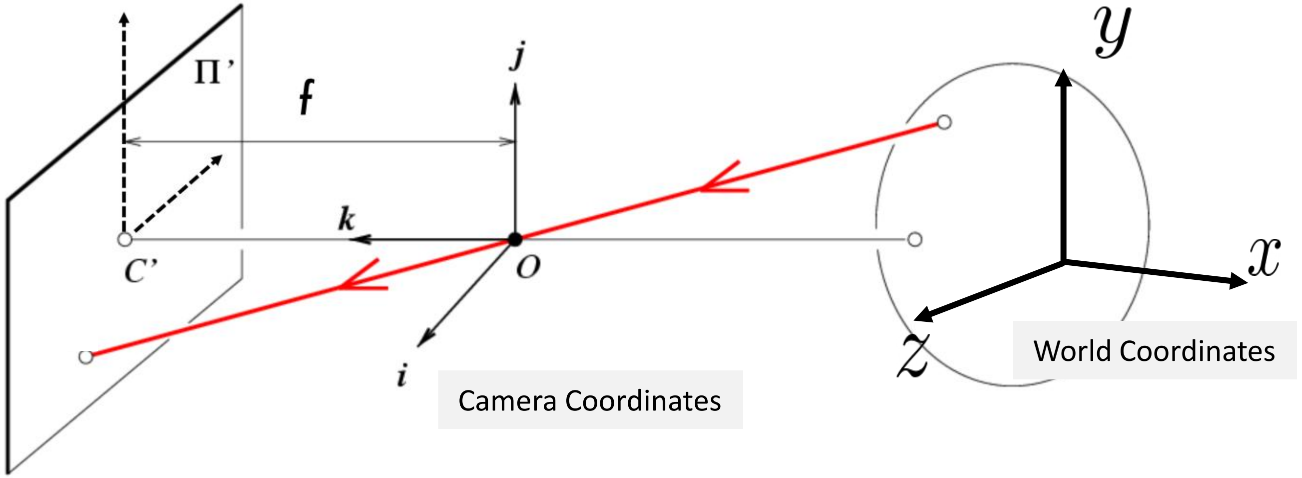
3x3 (Camera intrinsics)

3x4 (Camera extrinsics: rotation and translation)



$$K = \begin{bmatrix} \alpha & -\alpha \cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Back-projection to a Ray in the World Coordinate



$$P = K[R|\mathbf{t}]$$

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

- The camera center O is on the ray
- $P^{+}\mathbf{x}$ is on the ray

$$P^{+} = P^{T}(PP^{T})^{-1}$$

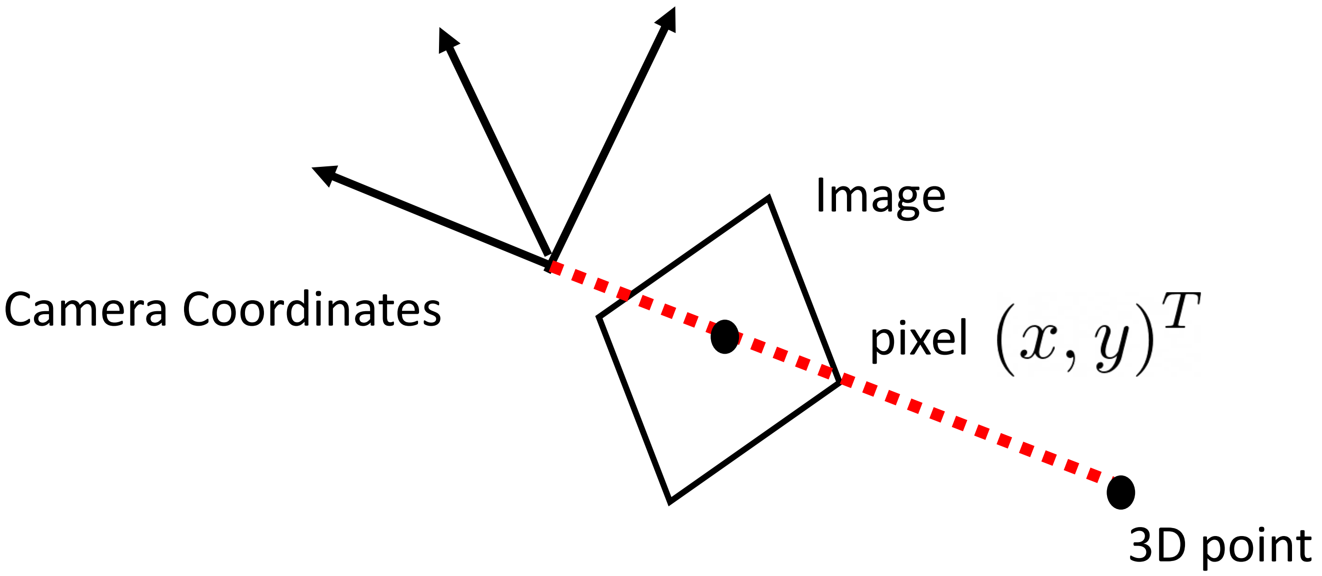
Pseudo-inverse

The ray can be written as

$$\mathbf{X}(\lambda) = (1-\lambda)P^{+}\mathbf{x} + \lambda O$$

- A pixel on the image backprojects to a ray in 3D

Back-projection to a 3D Point in Camera Coordinates



$$P = K[I|\mathbf{0}]$$

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_{\text{cam}}$$

$$\begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

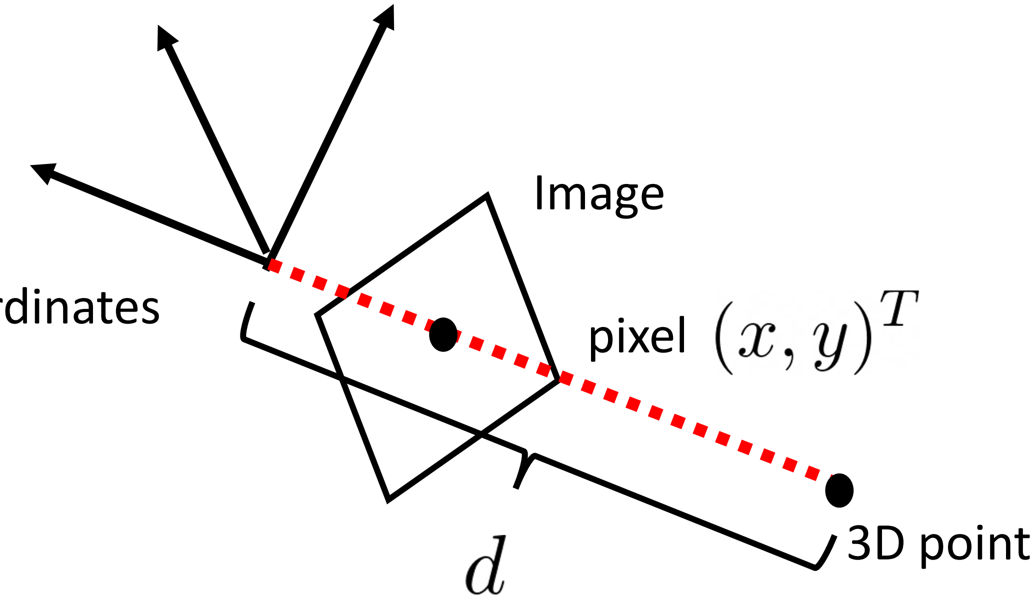
$$x = f\frac{X}{Z} + p_x \qquad \frac{X}{Z} = \frac{x - p_x}{f}$$

$$y = f\frac{Y}{Z} + p_y \qquad \frac{Y}{Z} = \frac{y - p_y}{f}$$

3D Point $\begin{bmatrix} \frac{x-p_x}{f} Z \\ \frac{y-p_y}{f} Z \\ Z \end{bmatrix}$

We need to know the depth of the pixel

Back-projection to a 3D Point in Camera Coordinates



3D camera coordinates $\begin{bmatrix} d\frac{x-p_x}{f_x} \\ d\frac{y-p_y}{f_y} \\ d \end{bmatrix}$

Equivalently

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X}_{\text{cam}}$$

$$K^{-1}\mathbf{x}$$

3D point with depth d : $dK^{-1}\mathbf{x}$

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

3D Point $\begin{bmatrix} \frac{x-p_x}{f}Z \\ \frac{y-p_y}{f}Z \\ Z \end{bmatrix}$

The Pinhole Camera Model

- Camera projection matrix: intrinsics and extrinsics

$$P = K[R|\mathbf{t}]$$

3x3 3x4

Camera intrinsics

Camera extrinsics: rotation and translation

Camera Calibration

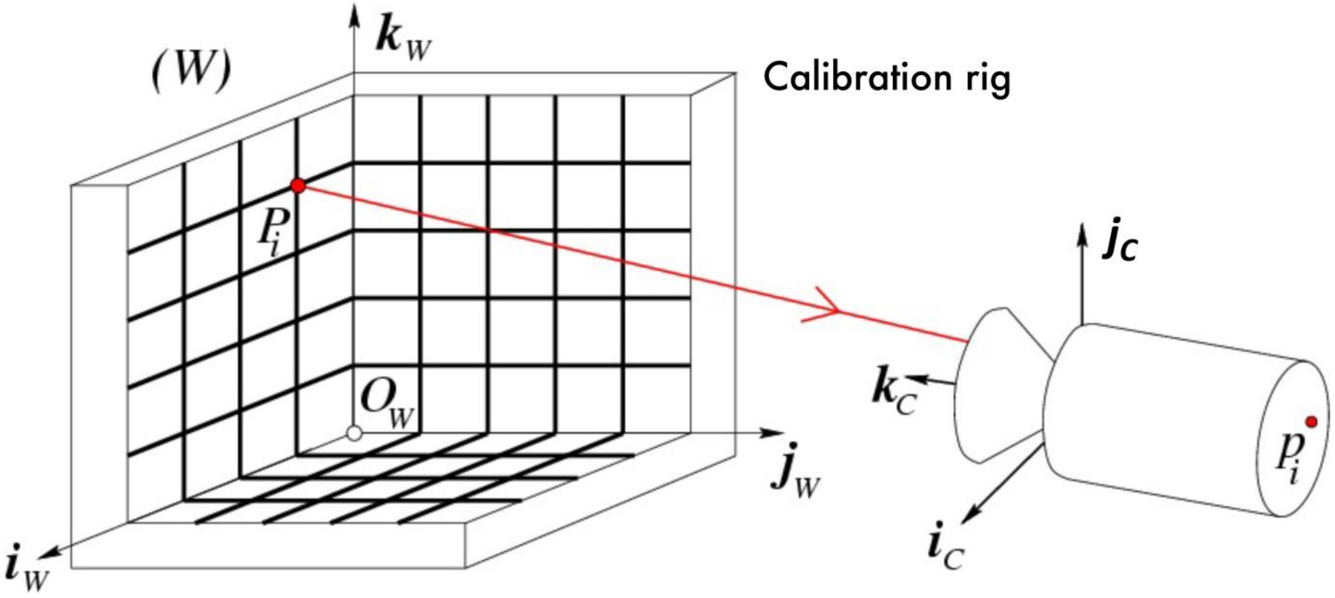
- Estimate the camera intrinsics and camera extrinsics $P = K[R|\mathbf{t}]$

- Why is this useful?
  - If we know K and depth, we can compute 3D points in camera frame
  - In stereo matching to compute depth, we need to know focal length
  - Camera pose tracking is critical in SLAM (Simultaneous Localization and Mapping)

Camera Calibration

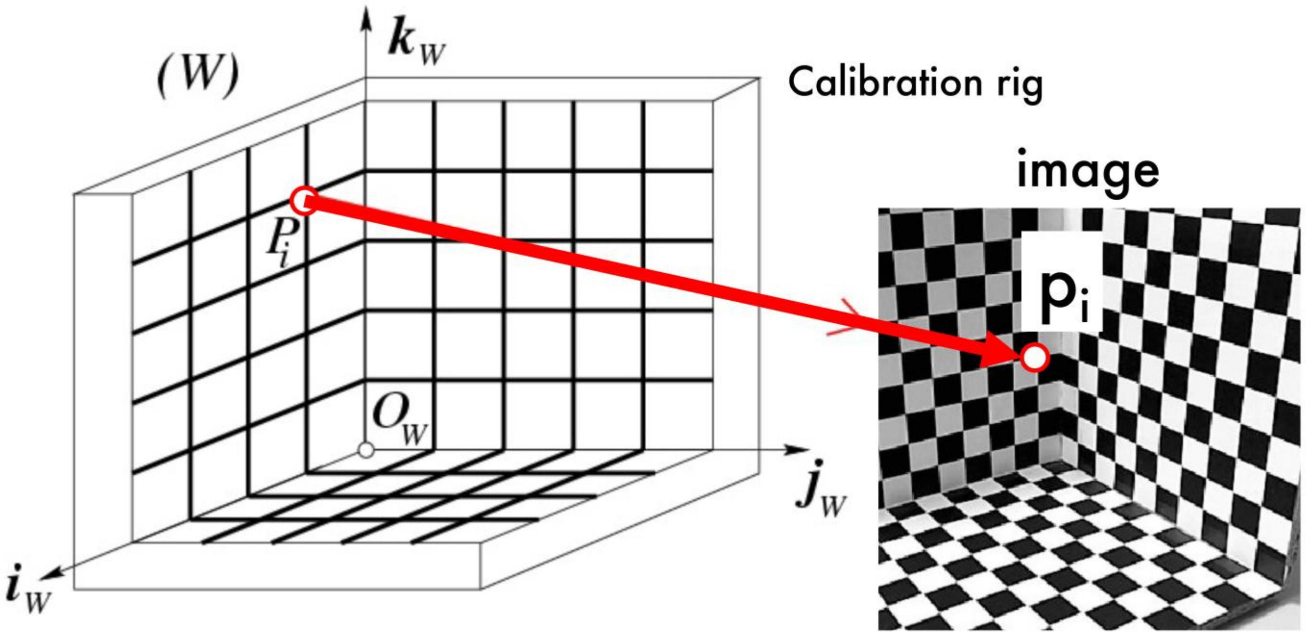
- Estimate the camera intrinsics and camera extrinsics $P = K[R|\mathbf{t}]$

- Idea: using images from the camera with a known world coordinate frame



checkerboard

Camera Calibration



- Unknowns

Camera intrinsics K

Camera extrinsics: rotation and translation R, T

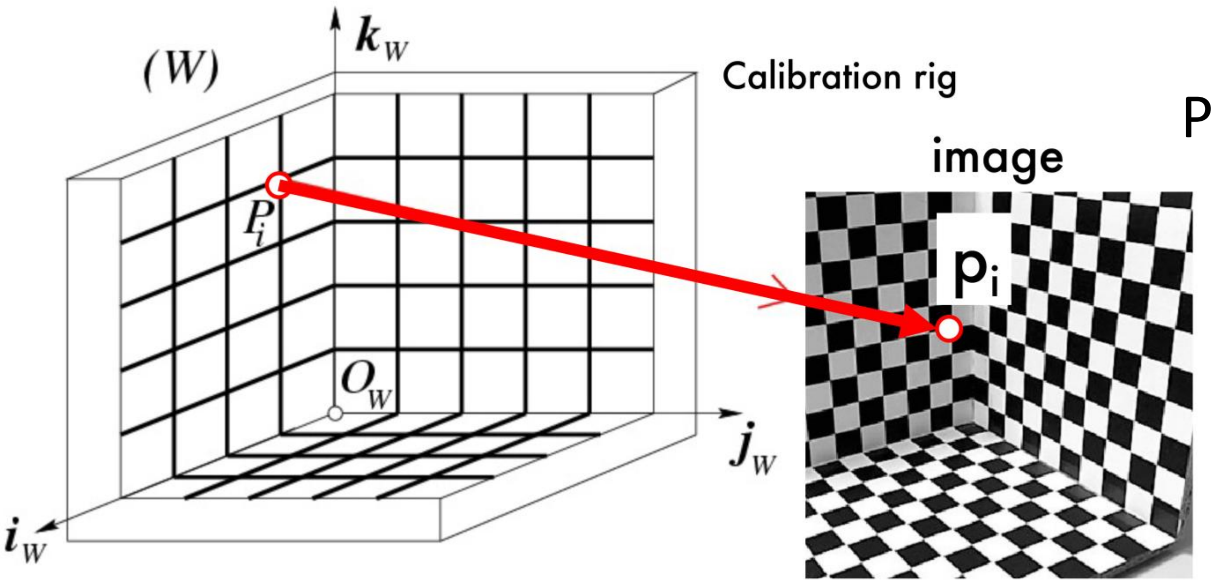
- Knowns

World coordinates $P_1, \ldots, P_n$

Pixel coordinates $p_1, \ldots, p_n$

Camera Calibration

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$



$$p_i = MP_i = K[R|T]P_i$$

Pixel coordinate → p_i; 3x4 → M; World coordinate → P_i

- How many unknowns in M?
  - 11
- How many correspondences do we need to estimate M?
  - We need 11 equations
  - 6 correspondences
- More correspondences are better

A Linear Approach to Camera Calibration

$$p_i = MP_i = K[R|T]P_i$$

$$M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \begin{matrix} 1 \times 4 \\ 1 \times 4 \\ 1 \times 4 \end{matrix} \qquad MP_i = \begin{bmatrix} \mathbf{m}_1 P_i \\ \mathbf{m}_2 P_i \\ \mathbf{m}_3 P_i \end{bmatrix} \qquad p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \frac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

Pixel

A pair of equations

$$u_i(m_3 P_i) - m_1 P_i = 0$$
$$v_i(m_3 P_i) - m_2 P_i = 0$$

A Linear Approach to Camera Calibration

- Given n correspondences $p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \leftrightarrow P_i$

2n equations

$$
\begin{aligned}
u_1(m_3 P_1) - m_1 P_1 &= 0 \\
v_1(m_3 P_1) - m_2 P_1 &= 0 \\
&\vdots \\
u_n(m_3 P_n) - m_1 P_n &= 0 \\
v_n(m_3 P_n) - m_2 P_n &= 0
\end{aligned}
$$

$$
\begin{bmatrix}
P_1^T & 0^T & -u_1 P_1^T \\
0^T & P_1^T & -v_1 P_1^T \\
 & \vdots & \\
P_n^T & 0^T & -u_n P_n^T \\
0^T & P_n^T & -v_n P_n^T
\end{bmatrix}
\begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} = \mathbf{P} m = 0
$$

$$2n \times 12 \qquad 12 \times 1$$

How to solve this linear system?

Linear System

$$\mathbf{P}m = 0$$

$$2n \times 12 \quad 12 \times 1$$

- Find non-zero solutions
- If m is a solution, k×m is also a solution for $k \in \mathcal{R}$
- We can seek a solution $||m|| = 1$

$$\min ||\mathbf{P}m||$$

Subject to $||m|| = 1$

Singular Value Decomposition

The SVD is a factorization of a $m \times n$ matrix into

$$\boldsymbol{A} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^T$$

where $\boldsymbol{U}$ is a $m \times m$ orthogonal matrix, $\boldsymbol{V}^T$ is a $n \times n$ orthogonal matrix and $\boldsymbol{\Sigma}$ is a $m \times n$ diagonal matrix.

For a square matrix ($\boldsymbol{m} = \boldsymbol{n}$):

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \dots$$

$$\boldsymbol{A} = \begin{pmatrix} \vdots & \dots & \vdots \\ \boldsymbol{u}_1 & \dots & \boldsymbol{u}_n \\ \vdots & \dots & \vdots \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} \begin{pmatrix} \dots & \mathbf{v}_1^T & \dots \\ \vdots & \vdots & \vdots \\ \dots & \mathbf{v}_n^T & \dots \end{pmatrix}$$

non-negative real numbers on the diagonal

Linear System

$$\min_{\|m\|=1} \|\mathbf{P}m\|$$

Singular value decomposition (SVD)

$$P = UDV^T \qquad \|Pm\| = \|UDV^T m\| = \|DV^T m\|$$

$$\|m\| = \|V^T m\| \qquad \min \|DV^T m\| \text{ s.t. } \|V^T m\| = 1$$

Let $y = V^T m \qquad \min \|Dy\|$ s.t. $\|y\| = 1 \qquad \mathbf{y} = (0, 0, \ldots, 0, 1)^T$

$$m = V\mathbf{y}$$

m is the last column of V

Linear System

$$\mathbf{P}m = 0$$

$2n \times 12 \quad 12 \times 1$

$$\min \|\mathbf{P}m\|$$

Subject to $\|m\| = 1$

Solution: $P = UDV^T$ SVD decomposition of P

$2n \times 2n \quad 2n \times 12 \quad 12 \times 12$

m is the last column of V

A5.3 in Multiview Geometry in Computer Vision

A Linear Approach to Camera Calibration

$$p_i = MP_i = K[R|T]P_i$$

$$\mathbf{P}m = 0$$

m is the last column of V

$$m \rightarrow M$$ Up to scale

How to extract K, R and T from M?

$$K = \begin{bmatrix} \alpha & -\alpha \cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{R} = \begin{bmatrix} \mathbf{r}_1^{\mathrm{T}} \\ \mathbf{r}_2^{\mathrm{T}} \\ \mathbf{r}_3^{\mathrm{T}} \end{bmatrix} \qquad \mathrm{T} = \begin{bmatrix} \mathrm{t}_x \\ \mathrm{t}_y \\ \mathrm{t}_z \end{bmatrix}$$

3 rows

$$\rho M = \begin{bmatrix} \alpha r_1^T - \alpha \cot\theta r_2^T + c_x r_3^T & \alpha t_x - \alpha \cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta} r_2^T + c_y r_3^T & \frac{\beta}{\sin\theta} t_y + c_y t_z \\ r_3^T & t_z \end{bmatrix}$$

Scale (→ ρ)

A Linear Approach to Camera Calibration

$$\frac{1}{\rho}\begin{bmatrix} \alpha r_1^T - \alpha \cot\theta r_2^T + c_x r_3^T & \alpha t_x - \alpha\cot\theta t_y + c_x t_z \\ \frac{\beta}{\sin\theta} r_2^T + c_y r_3^T & \frac{\beta}{\sin\theta} t_y + c_y t_z \\ r_3^T & t_z \end{bmatrix} = \begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

The rows of a rotation matrix are unit-length, perpendicular to each other

Intrinsics

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\rho = \pm\frac{1}{\|a_3\|}$$

$$c_x = \rho^2(a_1 \cdot a_3)$$

$$c_y = \rho^2(a_2 \cdot a_3)$$

$$\theta = \cos^{-1}\left(-\frac{(a_1 \times a_3)\cdot(a_2 \times a_3)}{\|a_1 \times a_3\| \cdot \|a_2 \times a_3\|}\right)$$

$$\alpha = \rho^2\|a_1 \times a_3\| \sin\theta$$

$$\beta = \rho^2\|a_2 \times a_3\| \sin\theta$$

Extrinsics

$$r_1 = \frac{a_2 \times a_3}{\|a_2 \times a_3\|}$$

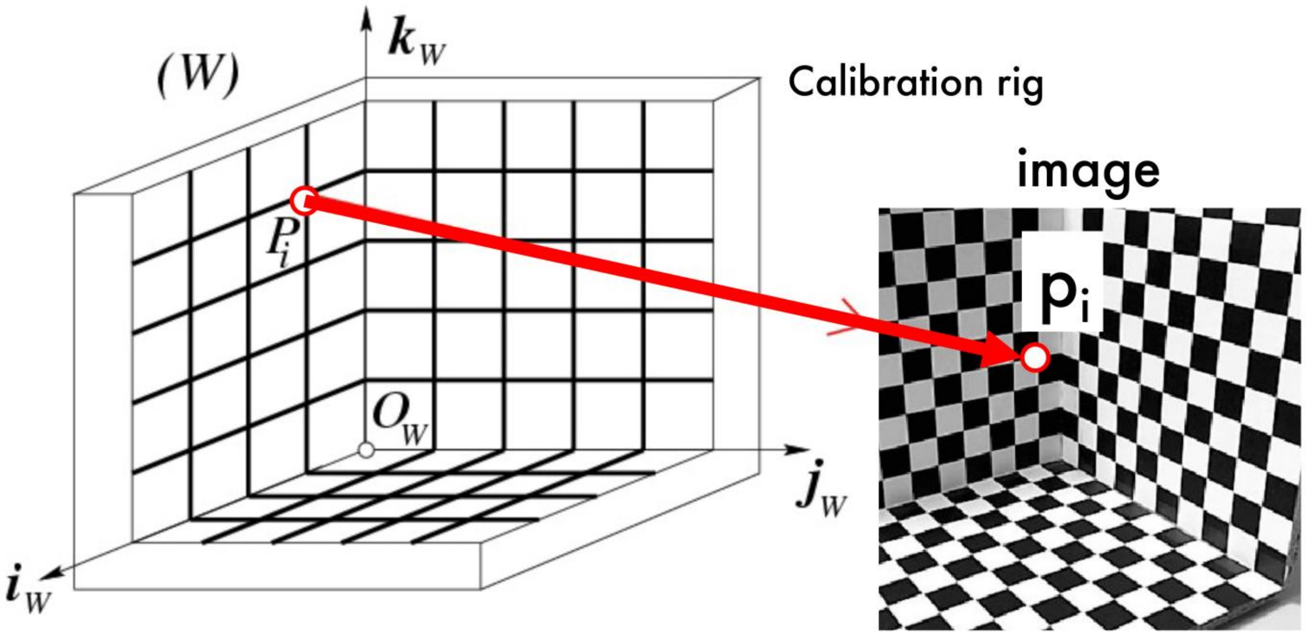
$$r_2 = r_3 \times r_1$$

$$r_3 = \rho a_3$$

$$T = \rho K^{-1} b$$

FP, Computer Vision: A Modern Approach, Sec. 3.2.2

A Linear Approach to Camera Calibration

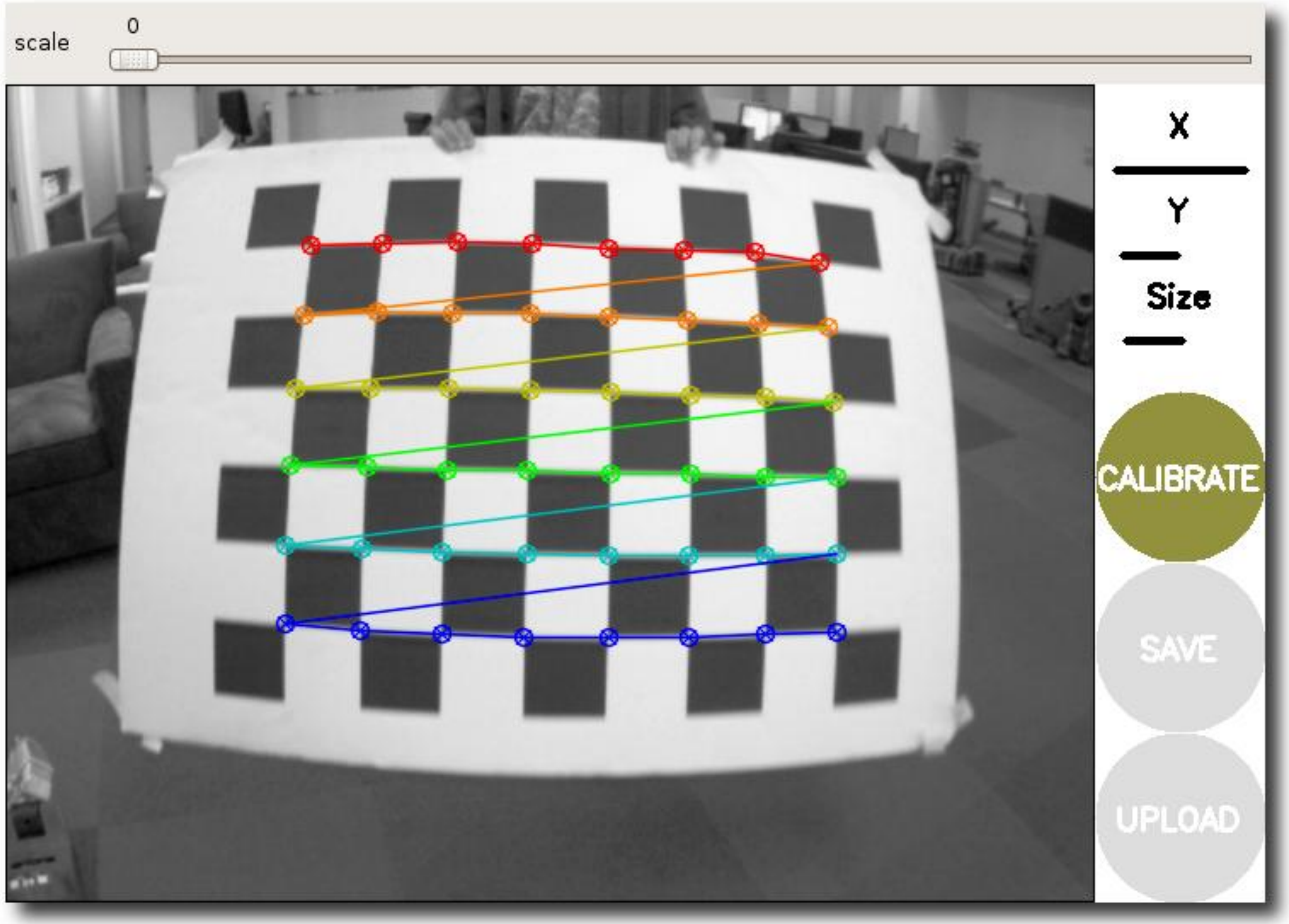


$$\mathbf{P}m = 0$$

All 3D points should NOT be on the same plane. Otherwise, no solution

FP, Computer Vision: A Modern Approach, Sec. 1.3

Camera Calibration with a 2D Plane

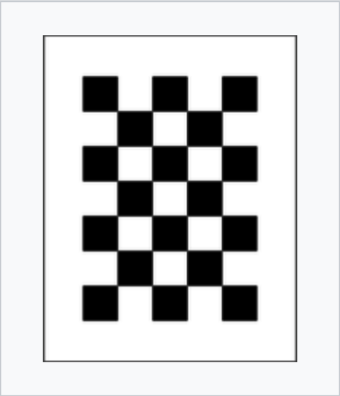


Harris Corner Detection

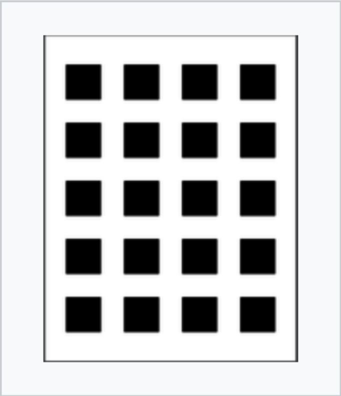
http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration

A Flexible New Technique for Camera Calibration. Zhengyou Zhang, TPAMI, 2000.

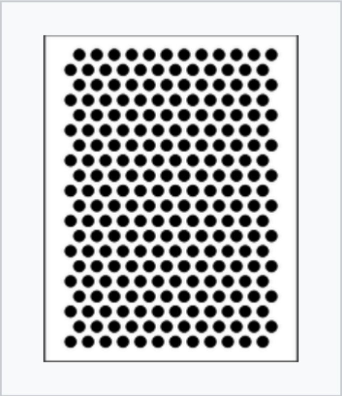
Calibration Patterns



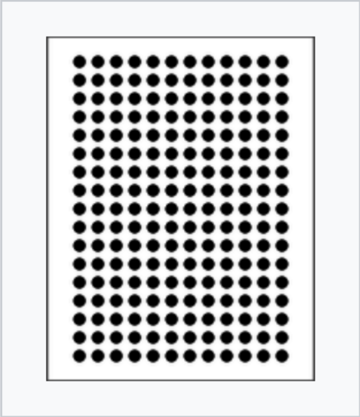
Chessboard



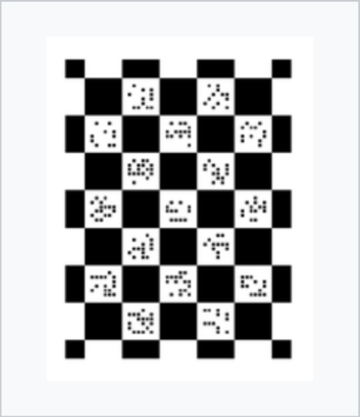
Square Grid



Circle Hexagonal Grid

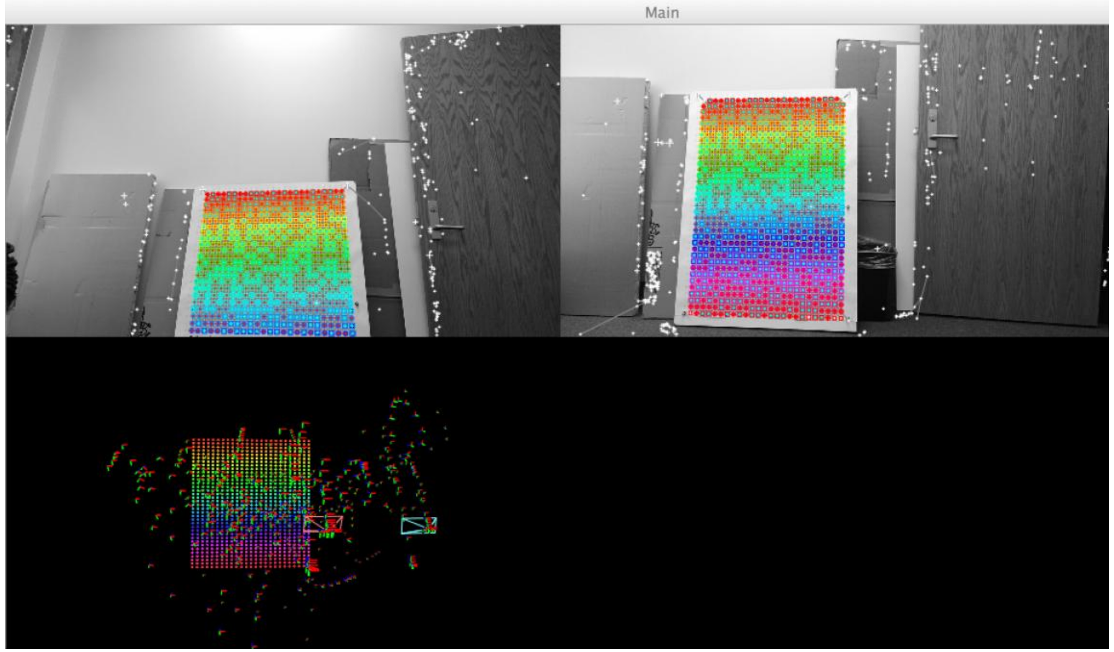


Circle Regular Grid



ECoCheck

https://boofcv.org/index.php?title=Tutorial_Camera_Calibration



https://github.com/arpg/Documentation/tree/master/Calibration

Further Reading

- Stanford CS231A: Computer Vision, From 3D Reconstruction to Recognition, Lecture 3 https://web.stanford.edu/class/cs231a/syllabus.html

- A Flexible New Technique for Camera Calibration. Zhengyou Zhang, TPAMI. 2000. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

- EP*n*P: An Accurate O(n) Solution to the P*n*P Problem. Lepetit et al., IJCV'09. https://www.tugraz.at/fileadmin/user_upload/Institute/ICG/Images/team_lepetit/publications/lepetit_ijcv08.pdf