Geometric Camera Calibration

EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

Readings: FP 1.3; SZ 6.3 (FL 4.6; extra notes)
Date: 9/17/14

Materials on these slides have come from many sources in addition to myself; I am infinitely grateful to these, especially Greg Hager, Silvio Savarese, and Steve Seitz.

Plan

- Review Perspective Projection
- Geometric Camera Calibration
  - Indirect camera calibration
    - Solve for projection matrix then the parameters
  - Direct camera calibration
  - Multi-planes method
    - Example with the Matlab Toolbox
- Catadioptric Sensing
  - *Different slide-deck. (See Chris Geyer's CVPR 2003 Tutorial)*

- Other calibration methods not covered
  - Vanishing points-based method (see SZ)
  - Self-calibration

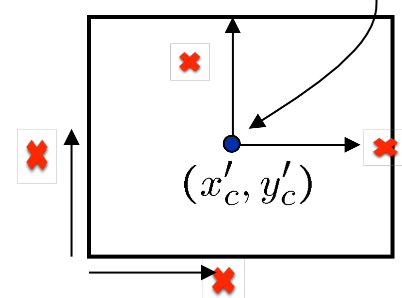
Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
- blue parameters are called "extrinsics," red are "intrinsics"

Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

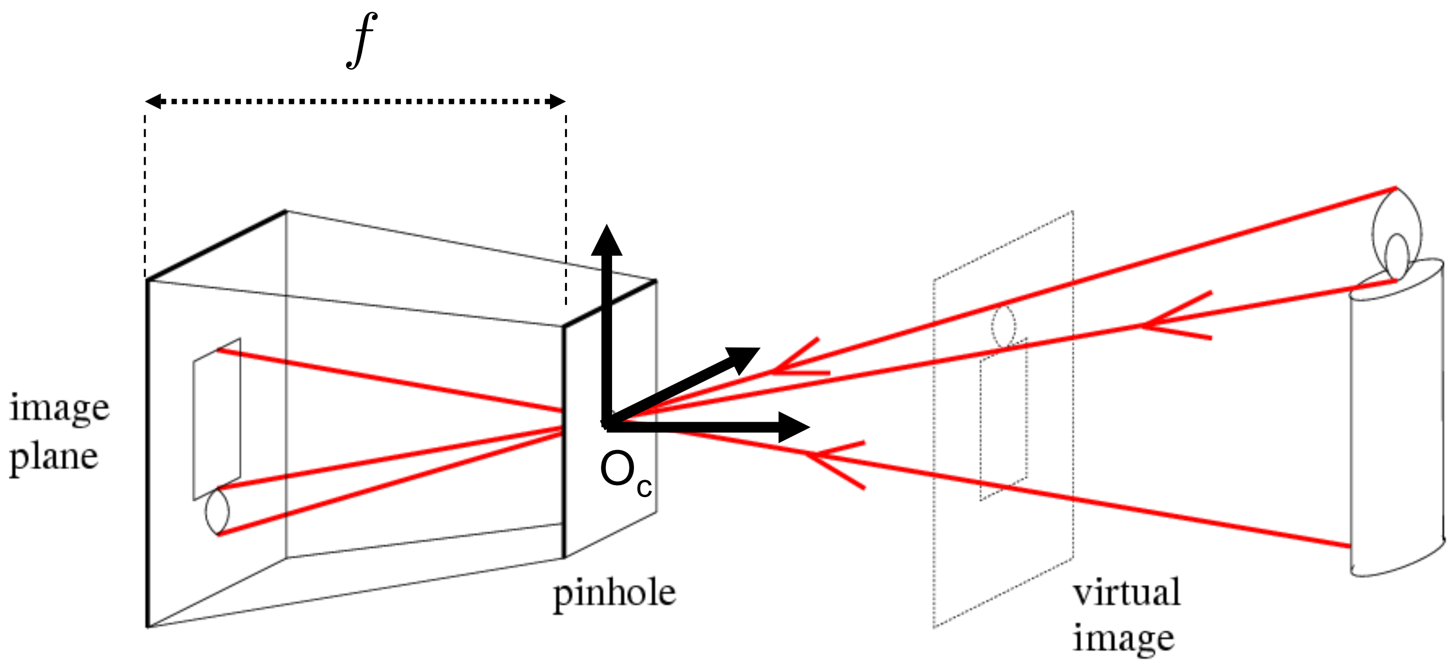
identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

- The definitions of these parameters are **not** completely standardized
  - especially intrinsics—varies from one book to another

Source: S Seitz slides.

Projective Camera



focal length: f

Source: S Savarese slides.

Projective Camera: The Normalized Image Plane

- The normalized image plane is parallel to the *physical retina* (e.g., ccd) but located at unit distance (f = 1) from the pinhole.

$$\hat{\mathbf{P}} = \frac{1}{Z}\begin{bmatrix}\mathcal{I} & \mathbf{0}\end{bmatrix}\mathbf{P} = \begin{bmatrix}\hat{X} \\ \hat{Y} \\ 1\end{bmatrix} = \begin{bmatrix}X/Z \\ Y/Z \\ 1\end{bmatrix}$$

$$\mathbf{P}' = \frac{1}{Z}\mathcal{M}\mathbf{P}$$

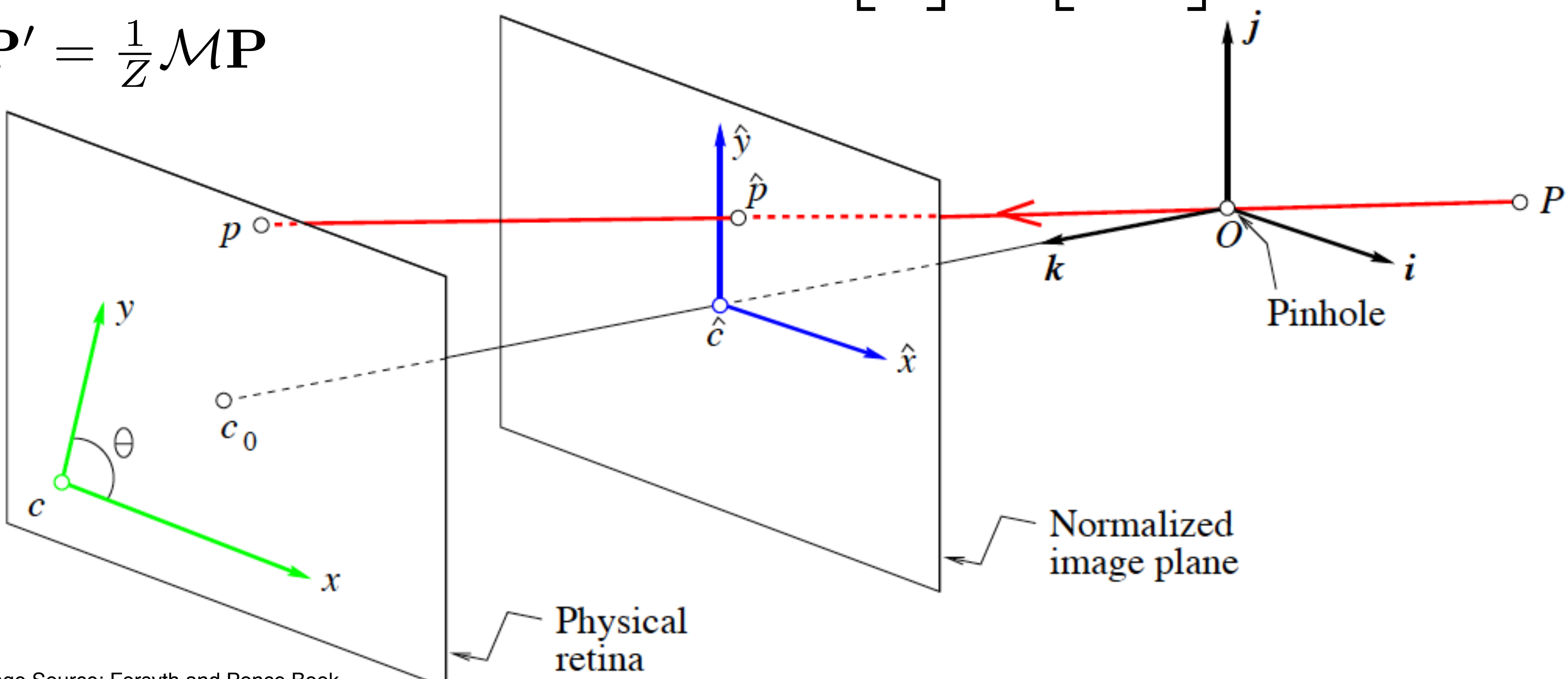


Image Source: Forsyth and Ponce Book.

Projective Camera: The Normalized Image Plane

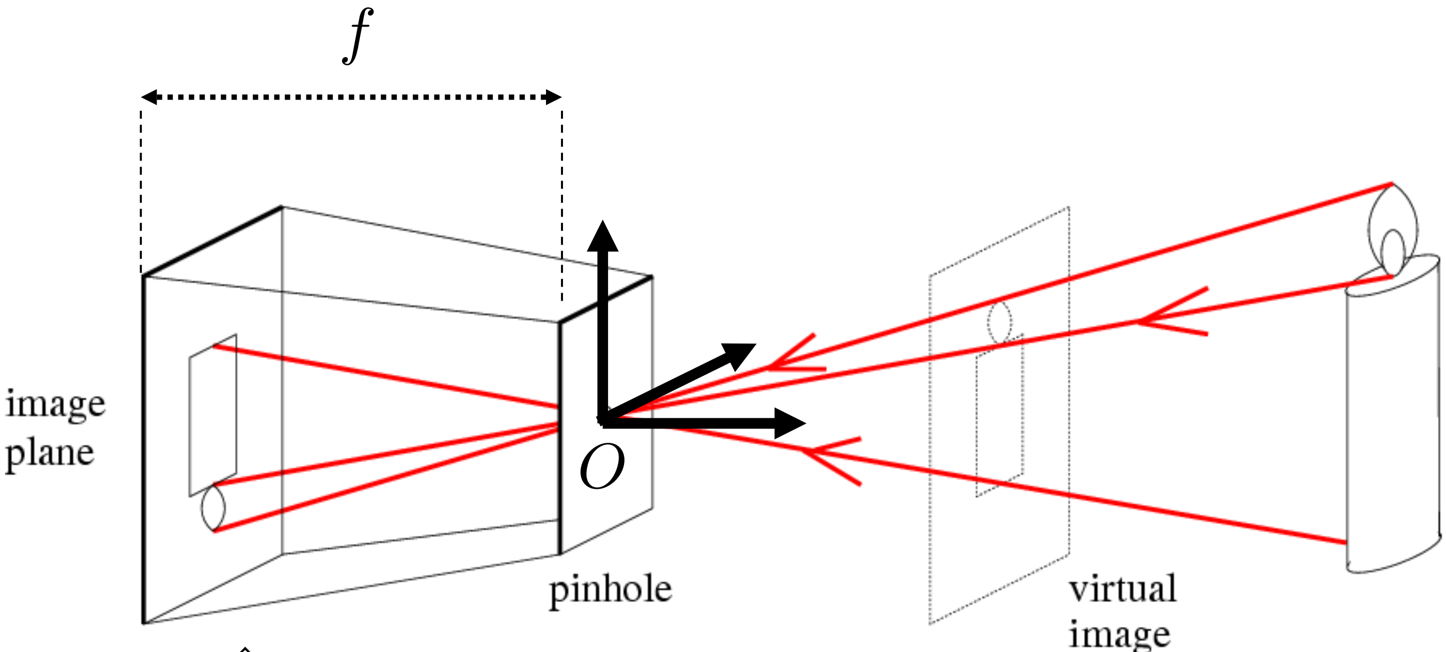
- Physical pixels in the *retina* (e.g. ccd) may not be square, so we have two additional scale parameters.

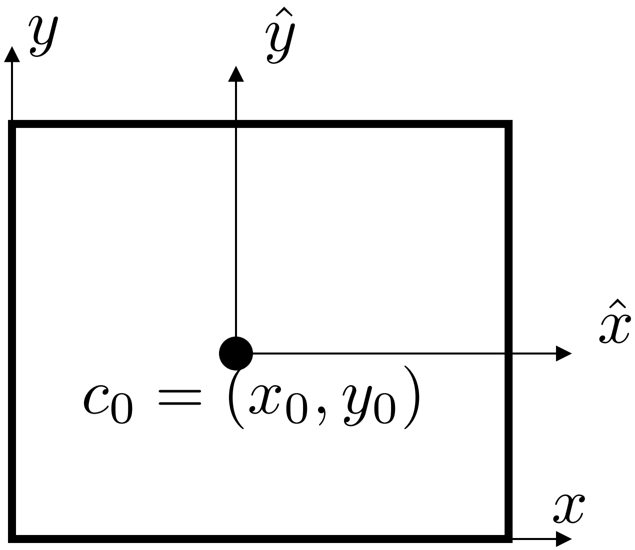
$$x = kf\hat{X} = kf\frac{X}{Z}$$
$$y = lf\hat{Y} = lf\frac{Y}{Z}$$

- Units:
  - f is a distance expressed in meters
  - A pixel will have dimensions $\frac{1}{k} \times \frac{1}{l}$ where k and l are in $\frac{\mathrm{px}}{m}$
- Can replace dependent pixel parameters

$$\alpha = kf$$
$$\beta = lf$$

Projective Camera

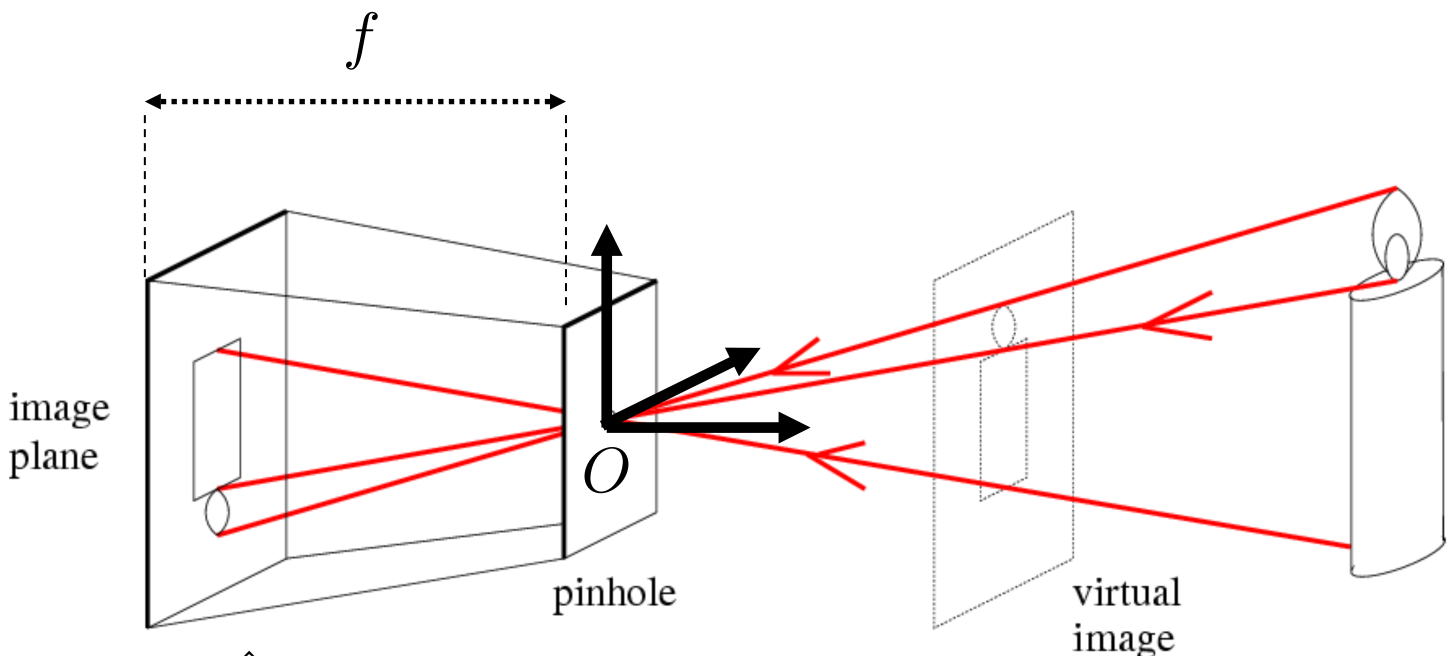


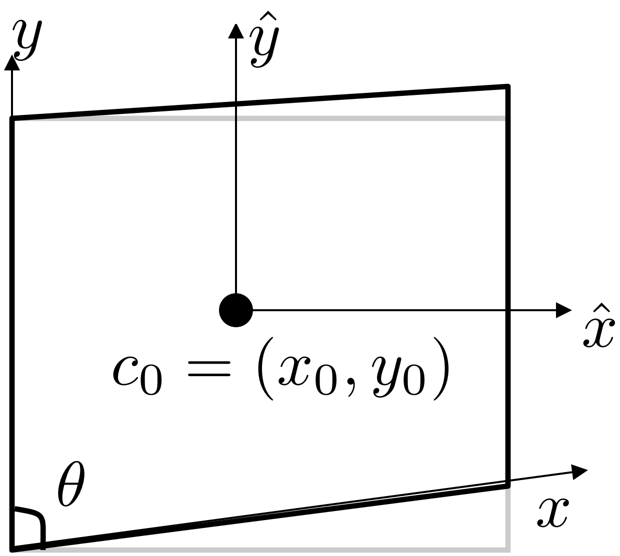


focal length: f
image center-point: c_0

Source: S Savarese slides.

Projective Camera

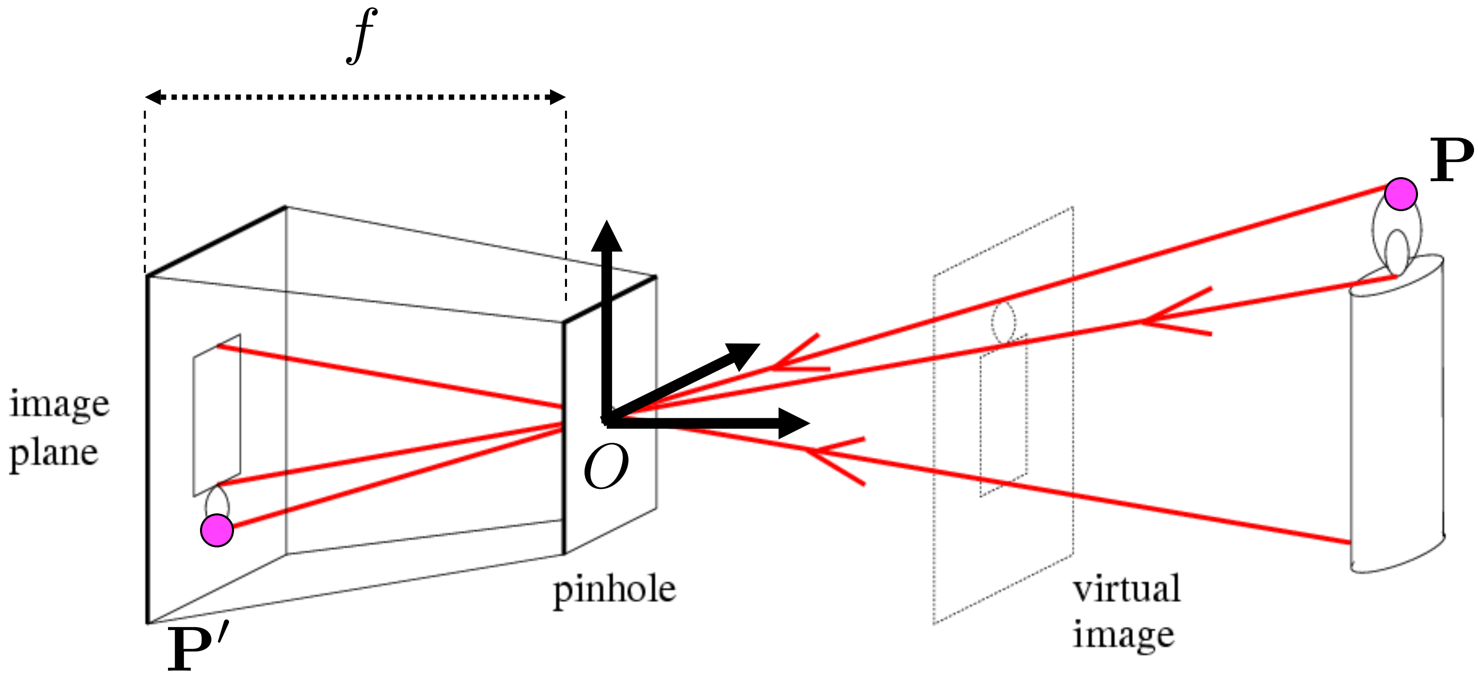




focal length: f
image center-point: c_0
non-square pixels: α, β
skew angle: θ

Source: S Savarese slides.

Projective Camera



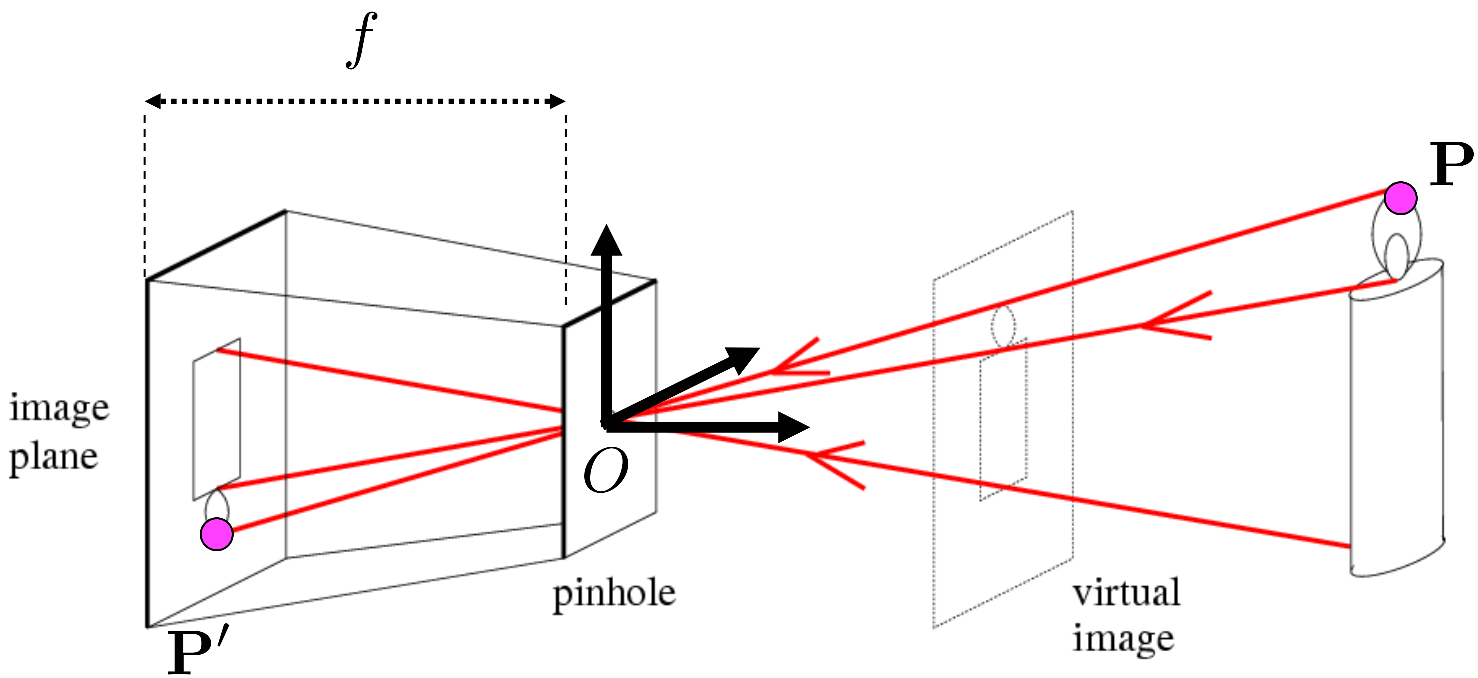
$$P' = \begin{bmatrix} \alpha & s & u_o & 0 \\ 0 & \beta & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

focal length: f
image center-point: c_0
non-square pixels: α, β
skew angle: θ

K has 5 degrees of freedom!

Source: S Savarese slides.

Projective Camera



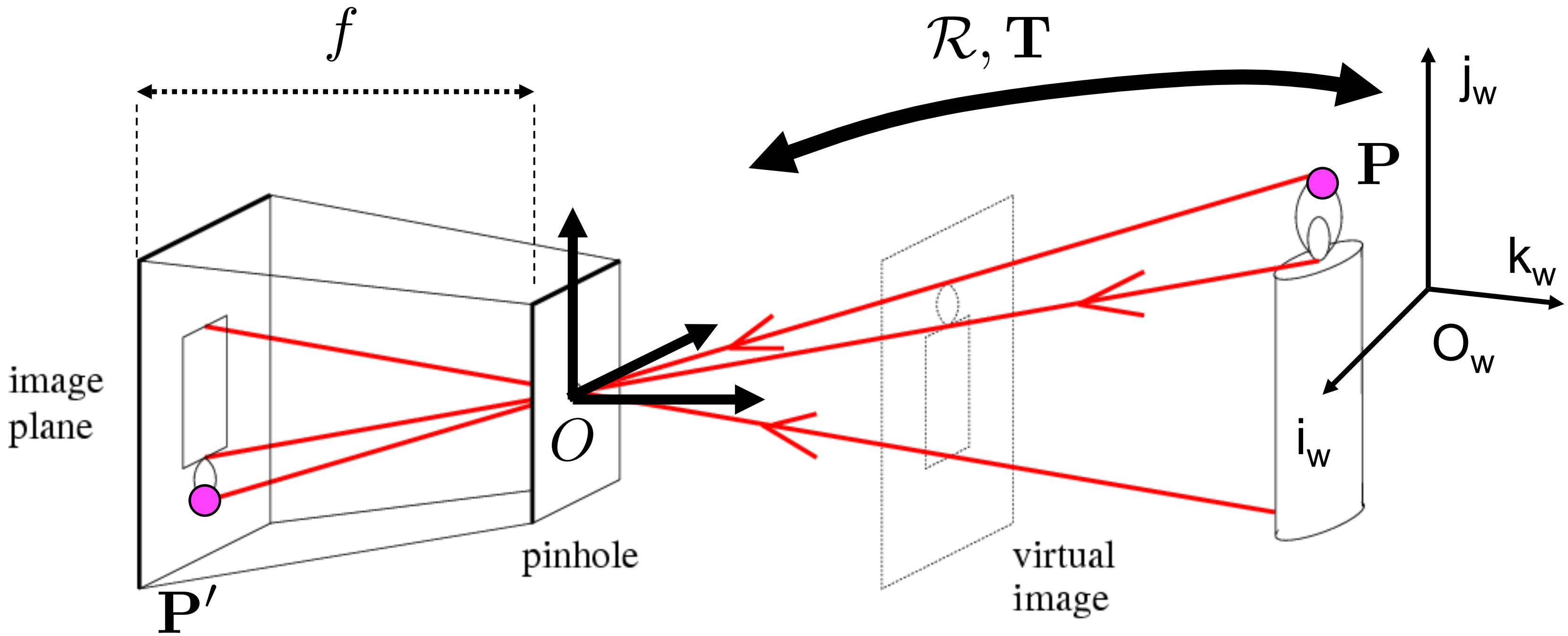
$$P' = \begin{bmatrix} \alpha & -\alpha \cot\theta & u_o & 0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

focal length: f
image center-point: c_0
non-square pixels: α, β
skew angle: θ

K has 5 degrees of freedom!

Source: S Savarese slides.

Projective Camera

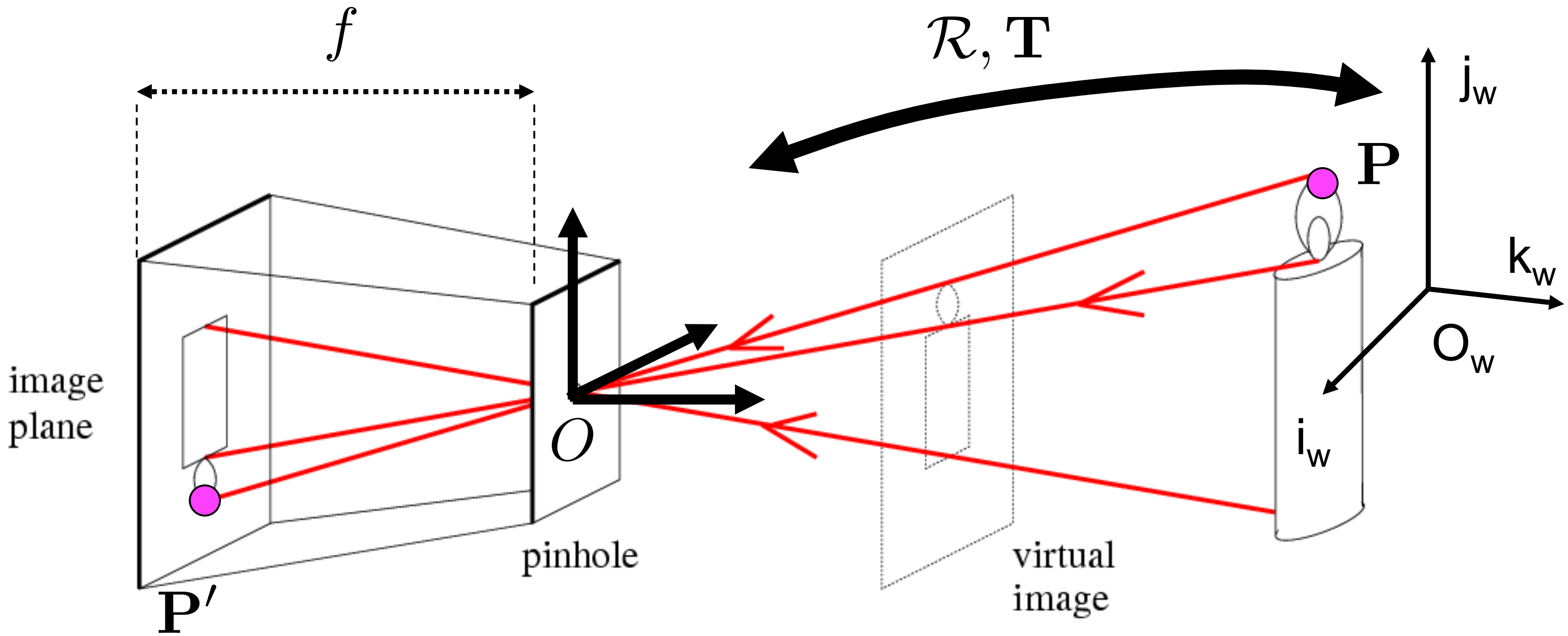


$$\mathbf{P} = \begin{bmatrix} \mathcal{R} & \mathbf{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}_{4\times 4} \mathbf{P}_w$$

focal length: f
image center-point: c_0
non-square pixels: α, β
skew angle: θ
rotation, translation: $\mathcal{R}, \mathbf{T}$

Source: S Savarese slides.

Projective Camera



$$\mathbf{P}' = \mathcal{M}\mathbf{P}_w$$

$$= \boxed{\mathcal{K}}\,\left[\boxed{\mathcal{R} \quad \mathbf{T}}\right]\mathbf{P}_w$$

Internal parameters

External parameters

focal length: f

image center-point: c_0

non-square pixels: α, β

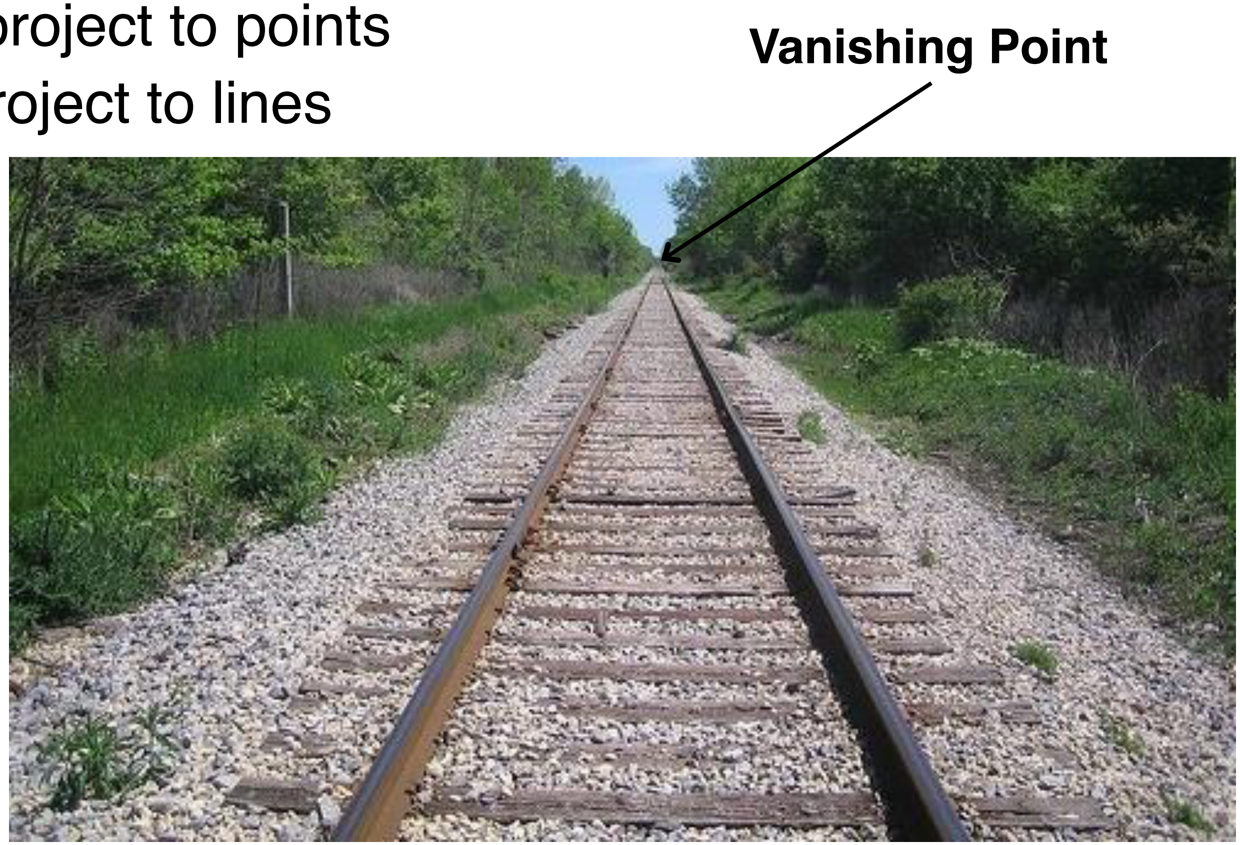
skew angle: θ

rotation, translation: $\mathcal{R}, \mathbf{T}$

Source: S Savarese slides.

Properties of Pinhole Perspective Projection

- Distant objects appear smaller
- Points project to points
- Lines project to lines



- Angles are not preserved.
- Parallel lines meet!

Source: S. Savarese slides.

Projective Camera

$$P' = M\, P_w \quad = \boxed{K}\big[\boxed{R \quad T}\big] P_w$$

Internal parameters

External parameters

Source: S Savarese slides.

Projective Camera

$$
\begin{aligned}
\mathbf{P}' &= \mathcal{M}\mathbf{P}_w \\
&= \boxed{\mathcal{K}}\ \left[\boxed{\mathcal{R} \quad \mathbf{T}}\right] \mathbf{P}_w
\end{aligned}
$$

Internal parameters

External parameters

Projective Camera

$$\mathbf{P}' = \mathcal{M}\mathbf{P}_w$$
$$= \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}\mathbf{P}_w$$

$$\boxed{\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}}$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \dfrac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Goal of Calibration

Estimate intrinsic and extrinsic parameters from 1 or multiple images

$$\mathbf{P}' = \mathcal{M}\mathbf{P}_w$$

$$= \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}\mathbf{P}_w$$

$$\boxed{\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}}$$

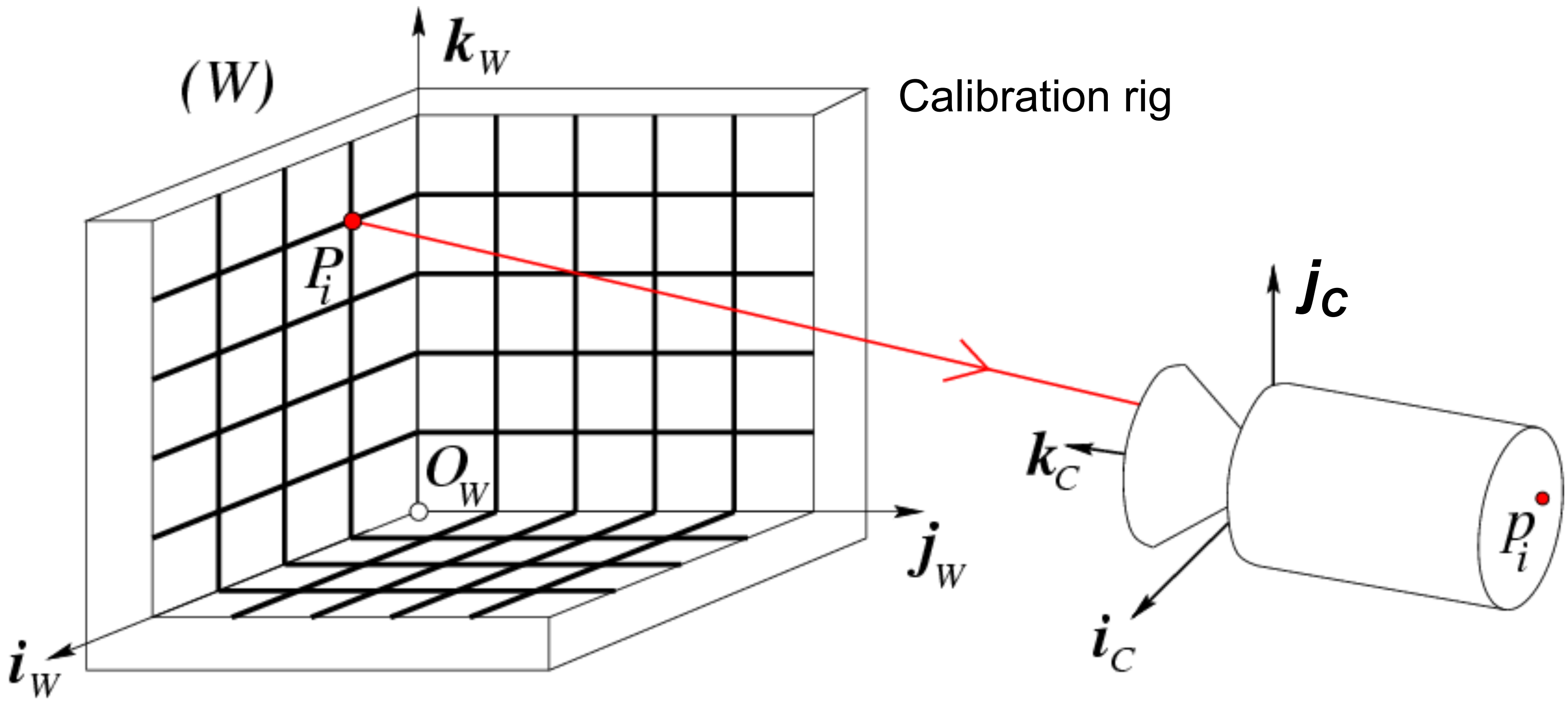
$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \dfrac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Change notation:
$P = P_w$

$p = P'$

Source: S Savarese slides.

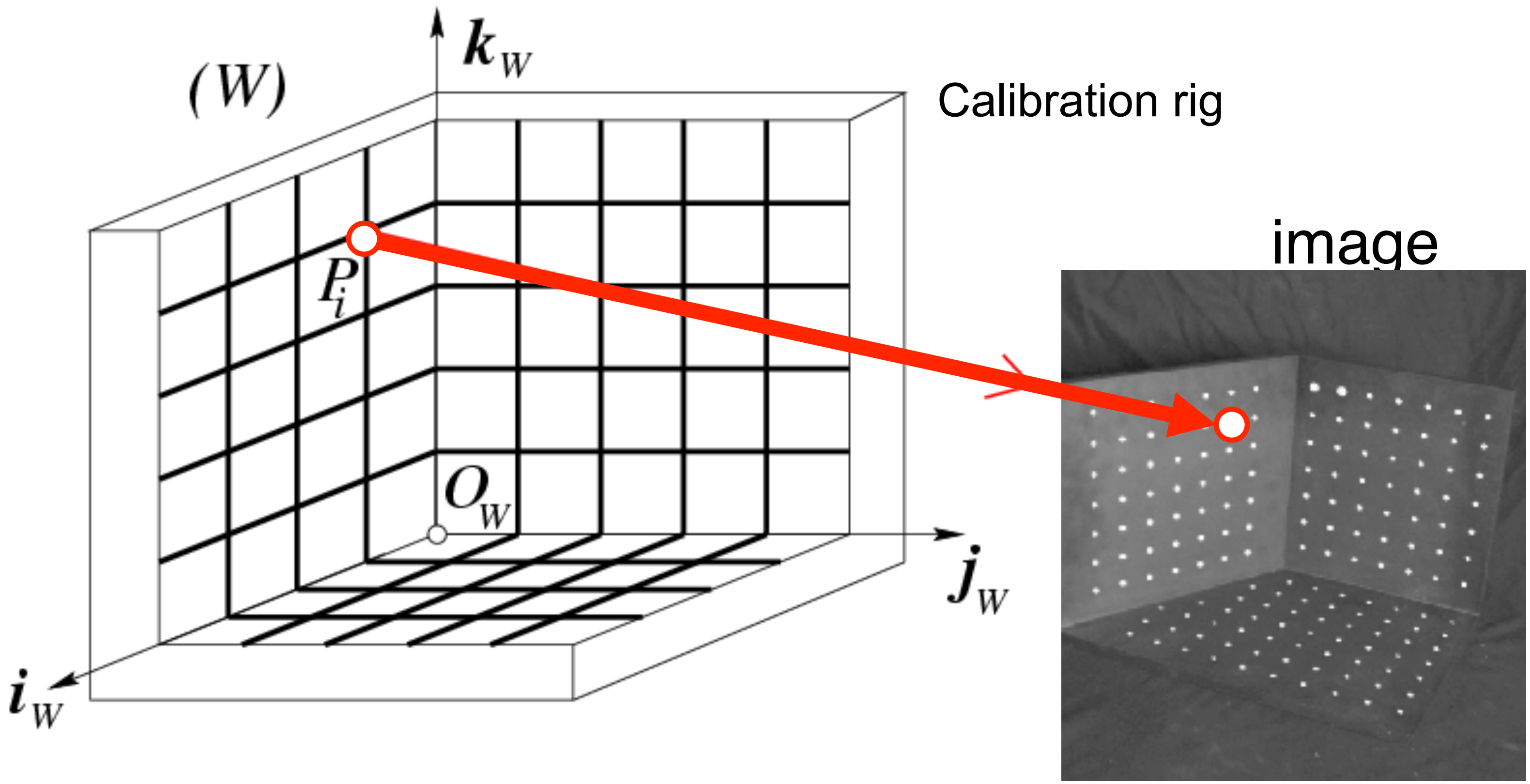
The Calibration Problem



•P_1… P_n with known positions in $[O_w,i_w,j_w,k_w]$
•p_1, … p_n known positions in the image
Goal: compute intrinsic and extrinsic parameters

Source: S Savarese slides.

The Calibration Problem

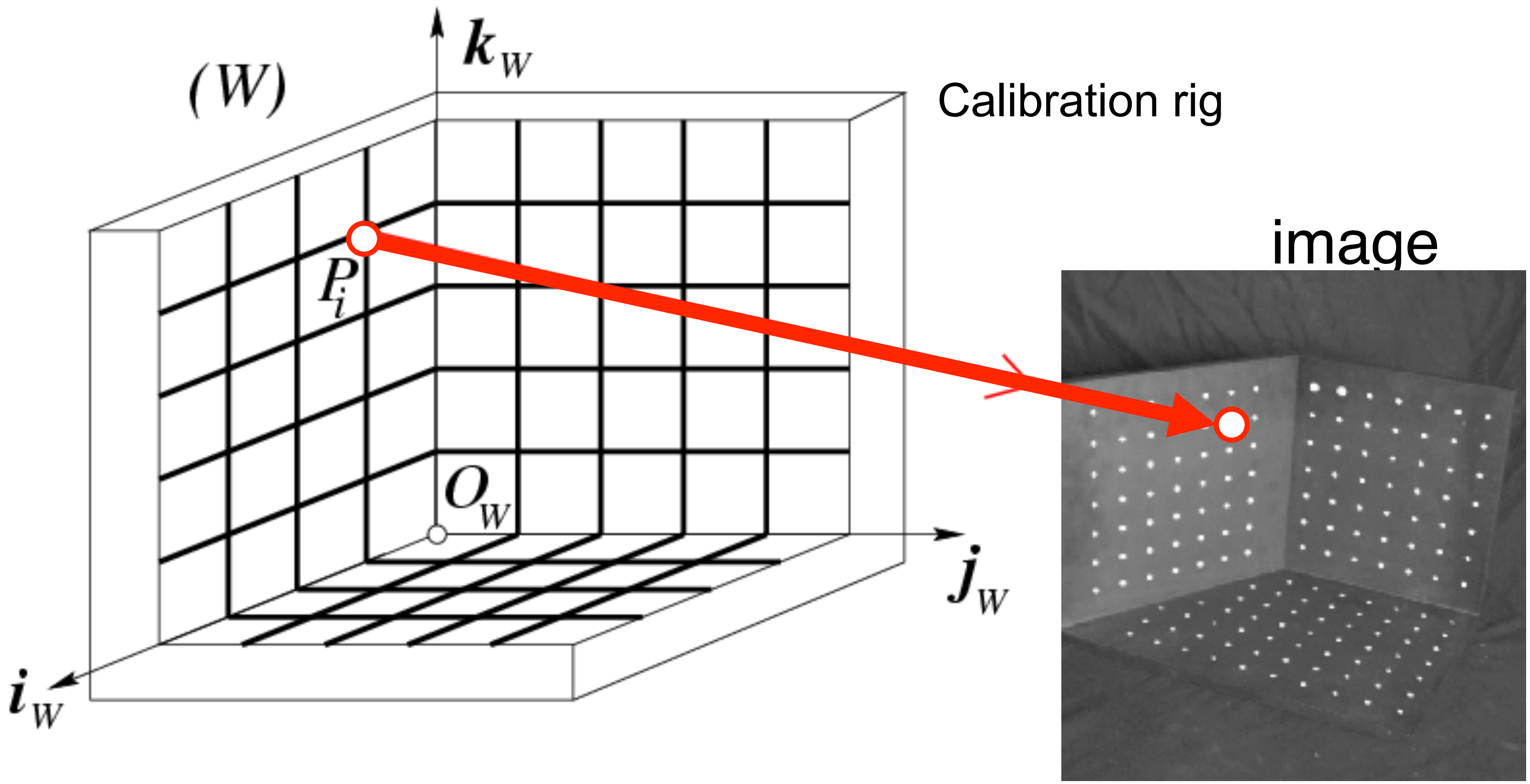


- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

Source: S Savarese slides.

The Calibration Problem

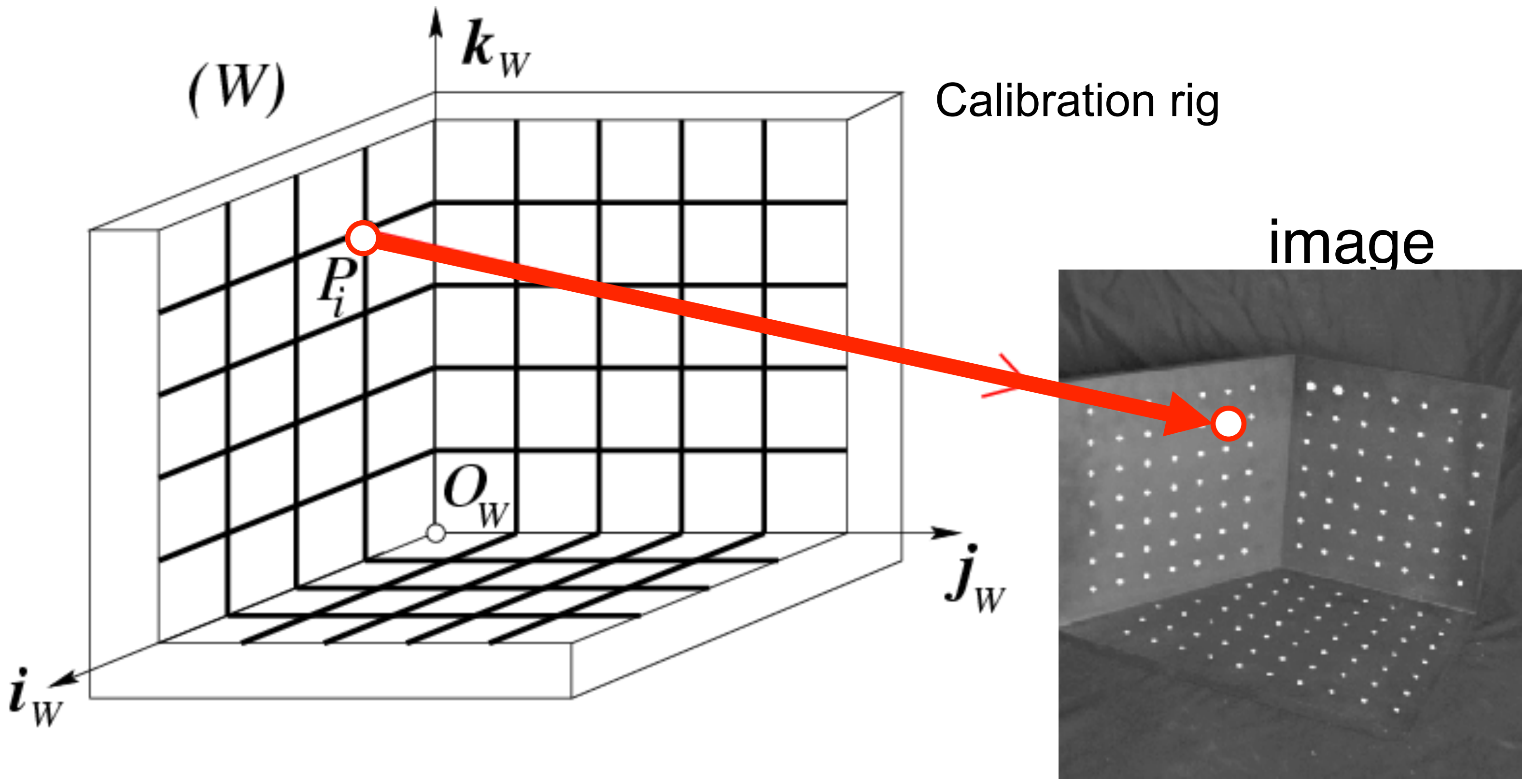


How many correspondences do we need?

- M has 11 unknown
- We need 11 equations
- 6 correspondences would do it

Source: S Savarese slides.

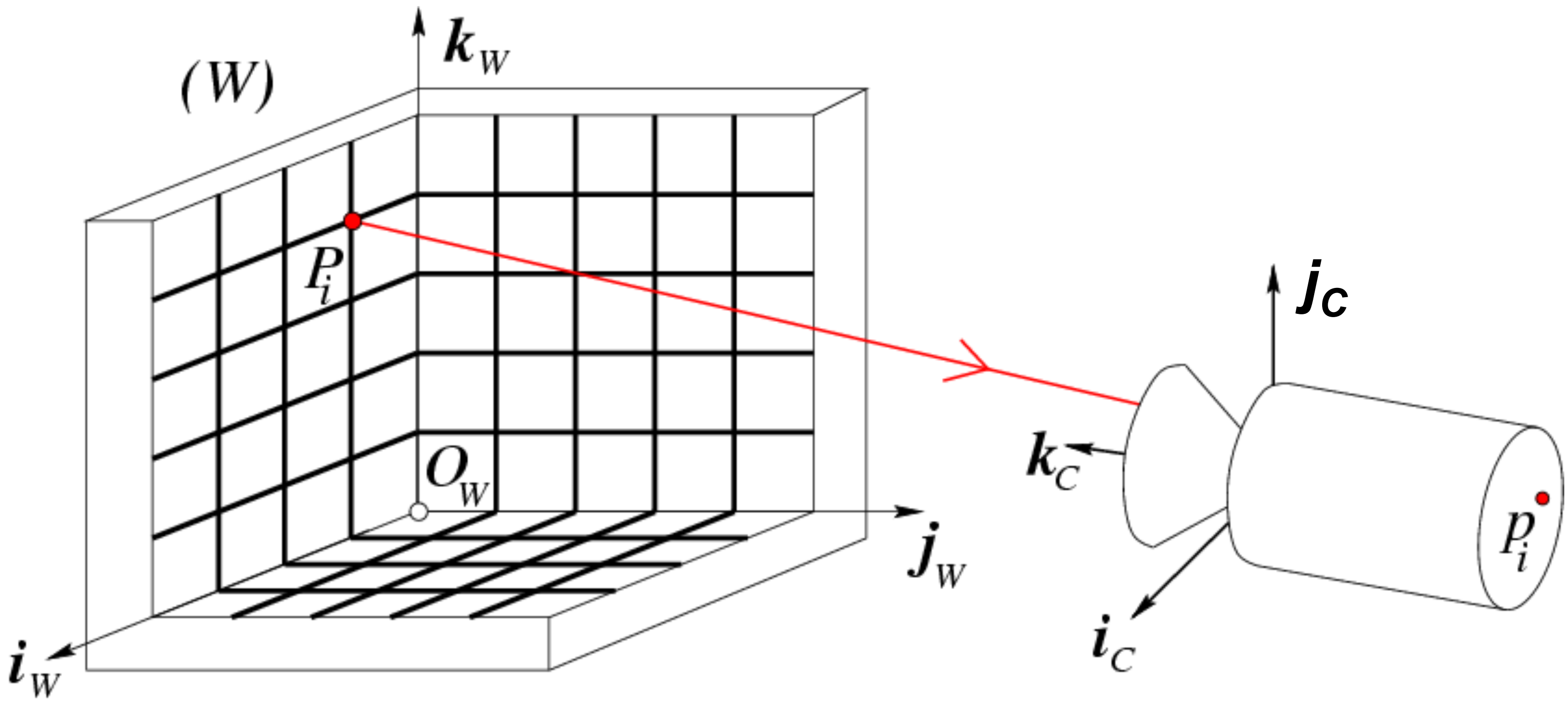
The Calibration Problem



In practice, using more than 6 correspondences enables more robust results

Source: S Savarese slides.

The Calibration Problem



$$P_i \rightarrow M\,P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1\, P_i}{\mathbf{m}_3\, P_i} \\ \dfrac{\mathbf{m}_2\, P_i}{\mathbf{m}_3\, P_i} \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

Source: S Savarese slides.

The Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 \, P_i}{\mathbf{m}_3 \, P_i} \\ \dfrac{\mathbf{m}_2 \, P_i}{\mathbf{m}_3 \, P_i} \end{bmatrix}$$

$$u_i = \frac{\mathbf{m}_1 \, P_i}{\mathbf{m}_3 \, P_i} \rightarrow u_i(\mathbf{m}_3 \, P_i) = \mathbf{m}_1 \, P_i \rightarrow u_i(\mathbf{m}_3 \, P_i) - \mathbf{m}_1 \, P_i = 0$$

$$v_i = \frac{\mathbf{m}_2 \, P_i}{\mathbf{m}_3 \, P_i} \rightarrow v_i(\mathbf{m}_3 \, P_i) = \mathbf{m}_2 \, P_i \rightarrow v_i(\mathbf{m}_3 \, P_i) - \mathbf{m}_2 \, P_i = 0$$

Source: S Savarese slides.

The Calibration Problem

$$
\begin{cases}
u_1(\mathbf{m}_3\ P_1) - \mathbf{m}_1\ P_1 = 0 \\
v_1(\mathbf{m}_3\ P_1) - \mathbf{m}_2\ P_1 = 0 \\
\vdots \\
u_i(\mathbf{m}_3\ P_i) - \mathbf{m}_1\ P_i = 0 \\
v_i(\mathbf{m}_3\ P_i) - \mathbf{m}_2\ P_i = 0 \\
\vdots \\
u_n(\mathbf{m}_3\ P_n) - \mathbf{m}_1\ P_n = 0 \\
v_n(\mathbf{m}_3\ P_n) - \mathbf{m}_2\ P_n = 0
\end{cases}
$$

Source: S Savarese slides.

Block Matrix Multiplication

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} \\ \mathrm{B}_{21} & \mathrm{B}_{22} \end{bmatrix}$$

What is AB ?

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

Source: S Savarese slides.

The Calibration Problem

$$\begin{cases} -u_1(\mathbf{m}_3\ P_1) + \mathbf{m}_1\ P_1 = 0 \\ -v_1(\mathbf{m}_3\ P_1) + \mathbf{m}_2\ P_1 = 0 \\ \vdots \\ -u_n(\mathbf{m}_3\ P_n) + \mathbf{m}_1\ P_n = 0 \\ -v_n(\mathbf{m}_3\ P_n) + \mathbf{m}_2\ P_n = 0 \end{cases} \longrightarrow \boxed{\mathcal{P}m = 0}$$

known

unknown

Homogenous linear system

1x4

$$\mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \mathbf{0}^T & -u_1\boldsymbol{P}_1^T \\ \mathbf{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \mathbf{0}^T & -u_n\boldsymbol{P}_n^T \\ \mathbf{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix}_{2n \times 12}$$

4x1

$$\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{m}_1^{\mathrm{T}} \\ \mathbf{m}_2^{\mathrm{T}} \\ \mathbf{m}_3^{\mathrm{T}} \end{pmatrix}_{12\times 1}$$

Source: S Savarese slides.

Homogeneous M x N Linear Systems

M=number of equations = 2n
N=number of unknown = 11

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

Rectangular system (M>N)

- 0 is always a solution
- To find non-zero solution

  Minimize $|\mathbf{P}\ \mathbf{m}|^2$

  under the constraint $|\mathbf{m}|^2 = 1$

The Calibration Problem

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

- How do we solve this homogenous linear system?
- Using DLT (Direct Linear Transformation) algorithm via SVD decomposition

Source: S Savarese slides.

Eigenvalues and Eigenvectors

Eigendecomposition

$$A = S\Lambda S^{-1} = S \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdot & \\ & & & \lambda_N \end{bmatrix} S^{-1}$$

Eigenvectors of A are columns of S

$$S = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_N \end{bmatrix}$$

Source: S Savarese slides.

Singular Value Decomposition

$$A = U \Sigma V^{-1} \qquad \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \cdot & \\ & & & \sigma_N \end{bmatrix}$$

U, V = orthogonal matrix

$$\sigma_i = \sqrt{\lambda_i}$$

σ = singular value

λ = eigenvalue of $A^t A$

Source: S Savarese slides.

Properties of SVD

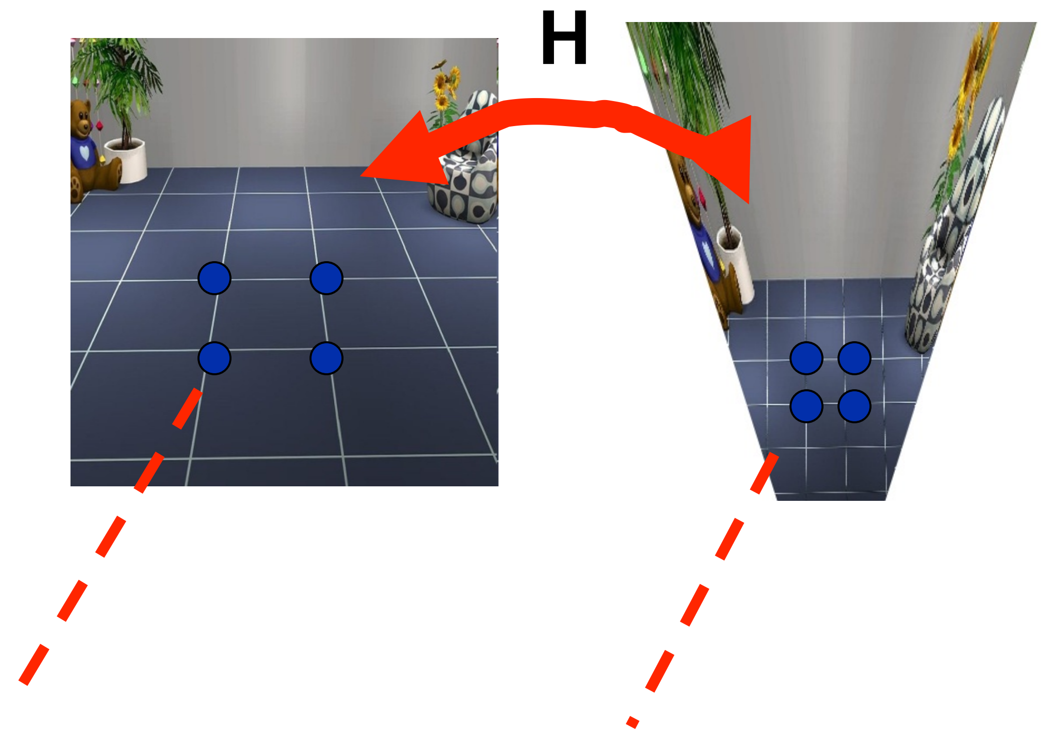
- Recall the singular values of a matrix are related to its rank.
- Recall that Ax = 0 can have a nonzero x as solution only if A is singular.
- Finally, note that the matrix V of the SVD is an orthogonal basis for the domain of A; in particular the zero singular values are the basis vectors for the null space.

- Putting all this together, we see that A must have rank 7 (in this particular case) and thus x must be a vector in this subspace.

- Clearly, x is defined only up to scale.

Source: G Hager slides.

DLT algorithm (Direct Linear Transformation)



$$\mathrm{x}'_i = \mathrm{H}\,\mathrm{x}_i \longrightarrow \underbrace{\mathrm{A}_i}_{\text{Function of measurements}}\overbrace{\mathbf{h}}^{\text{unknown}} = 0$$

Source: S Savarese slides.

The Calibration Problem

$$\boxed{\mathcal{P}}m = 0$$

SVD decomposition of P

$$\boxed{U_{2n\times 12}\ D_{12\times 12} V^{T}{}_{12\times 12}}$$

Last column of V gives $\boldsymbol{m}$

$$M$$

$$M\,P_i \rightarrow p_i$$

Why? See pag 593 of AZ

Source: S Savarese slides.

Clarification about SVD

•Thanks to Pat O' Keefe!

$$P_{m\times n} = \boxed{U}_{m\times n} \; D_{n\times n} \; \boxed{V}^{T}_{n\times n}$$

Has n orthogonal columns

Orthogonal matrix

- This is one of the possible SVD decompositions
- This is typically used for efficiency
- The classic SVD is actually:

$$P_{m\times n} = \boxed{U}_{m\times m} \; D_{\;m\times n} \; \boxed{V}^{T}_{n\times n}$$

orthogonal

Orthogonal

Source: S Savarese slides.

Extracting Camera Parameters

$$\rho\mathcal{M} = \left(\begin{array}{c|c} \alpha\mathbf{r}_1^\top - \alpha\cot\theta\mathbf{r}_2^\top + u_0\mathbf{r}_3^\top & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^\top + v_0\mathbf{r}_3^\top & \frac{\beta}{\sin\theta}t_y + v_0 t_z \\ \mathbf{r}_3^\top & t_z \end{array}\right) = \mathcal{K}\left[\mathcal{R} \quad \mathbf{T}\right]$$

$$\mathcal{A} \qquad \mathbf{b}$$

$$\mathcal{K} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \mathbf{a}_3^\top \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Estimated values

Intrinsic

$$\rho = \frac{\pm 1}{|\mathbf{a}_3|} \qquad u_0 = \rho^2\mathbf{a}_1^\top\mathbf{a}_3$$

$$v_0 = \rho^2\mathbf{a}_2^\top\mathbf{a}_3$$

$$\cos\theta = \frac{(\mathbf{a}_1 \times \mathbf{a}_3)^\top(\mathbf{a}_2 \times \mathbf{a}_3)}{|\mathbf{a}_1 \times \mathbf{a}_3| \cdot |\mathbf{a}_2 \times \mathbf{a}_3|}$$

Source: S Savarese slides.

Theorem (Faugeras, 1993)

Let $\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b})$ be a 3×4 matrix and let $\boldsymbol{a}_i^T$ $(i = 1, 2, 3)$ denote the rows of the matrix $\mathcal{A}$ formed by the three leftmost columns of $\mathcal{M}$.

- A necessary and sufficient condition for $\mathcal{M}$ to be a perspective projection matrix is that $\text{Det}(\mathcal{A}) \neq 0$.
- A necessary and sufficient condition for $\mathcal{M}$ to be a zero-skew perspective projection matrix is that $\text{Det}(\mathcal{A}) \neq 0$ and

$$(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0.$$

- A necessary and sufficient condition for $\mathcal{M}$ to be a perspective projection matrix with zero skew and unit aspect-ratio is that $\text{Det}(\mathcal{A}) \neq 0$ and

$$\begin{cases} (\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0, \\ (\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_1 \times \boldsymbol{a}_3) = (\boldsymbol{a}_2 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3). \end{cases}$$

Source: S Savarese slides.

Extracting Camera Parameters

$$\rho\mathcal{M} = \begin{pmatrix} \alpha\mathbf{r}_1^\top - \alpha\cot\theta\,\mathbf{r}_2^\top + u_0\mathbf{r}_3^\top & \alpha t_x - \alpha\cot\theta\, t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^\top + v_0\mathbf{r}_3^\top & \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ \mathbf{r}_3^\top & t_z \end{pmatrix} = \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}$$

$$\mathcal{A} \qquad\qquad \mathbf{b}$$

$$\mathcal{K} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \mathbf{a}_3^\top \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Estimated values

Intrinsic

$$\left.\begin{aligned} \alpha &= \rho^2|\mathbf{a}_1 \times \mathbf{a}_3|\sin\theta \\ \beta &= \rho^2|\mathbf{a}_2 \times \mathbf{a}_3|\sin\theta \end{aligned}\right\} \rightarrow f$$

Source: S Savarese slides.

Extracting Camera Parameters

$$\rho\mathcal{M} = \underbrace{\begin{pmatrix} \alpha\mathbf{r}_1^\top - \alpha\cot\theta\,\mathbf{r}_2^\top + u_0\mathbf{r}_3^\top \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^\top + v_0\mathbf{r}_3^\top \\ \mathbf{r}_3^\top \end{pmatrix}}_{\mathcal{A}} \underbrace{\begin{matrix} \alpha t_x - \alpha\cot\theta\, t_y + u_0 t_z \\ \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ t_z \end{matrix}}_{\mathbf{b}} = \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}$$

$$\mathcal{K} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \mathbf{a}_3^\top \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Estimated values

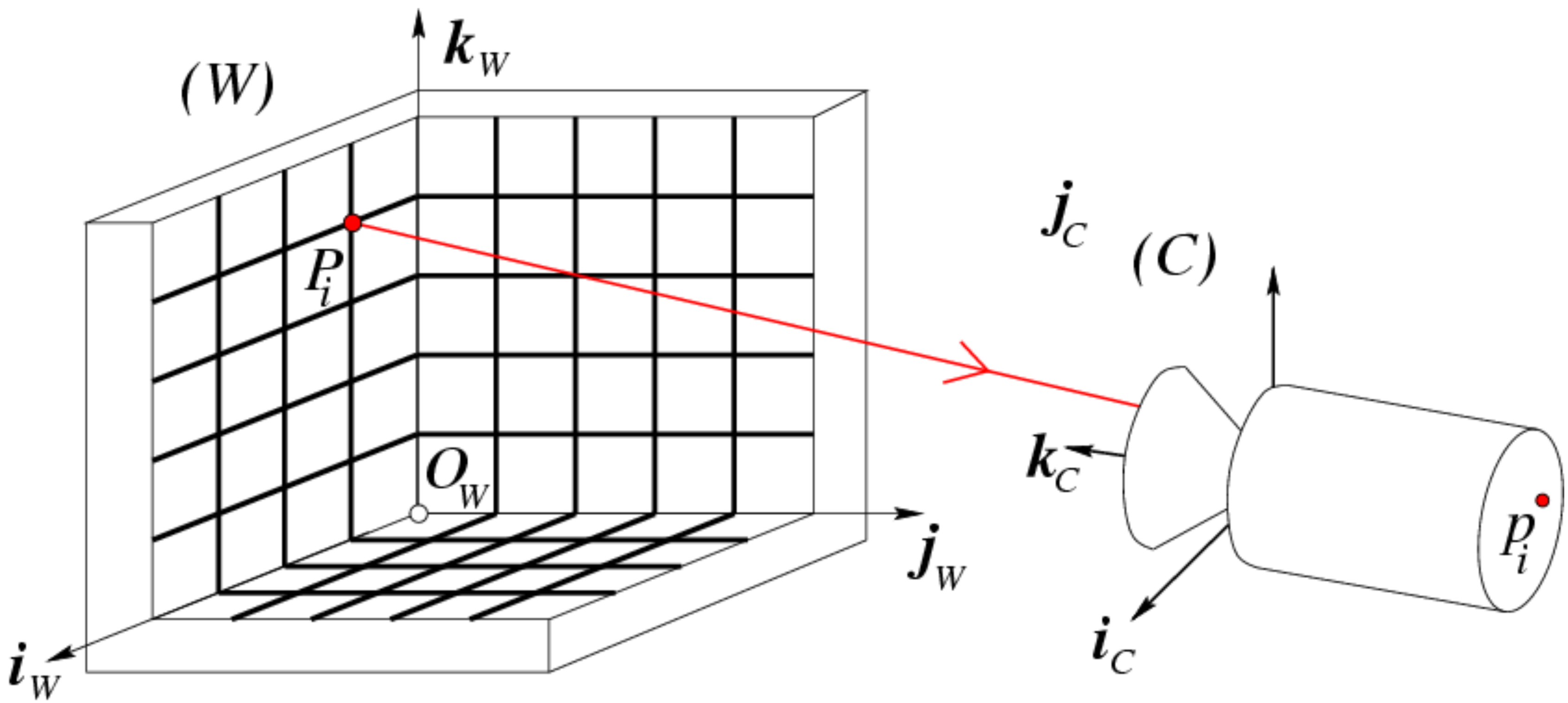
Extrinsic

$$\mathbf{r}_1 = \frac{\mathbf{a}_2 \times \mathbf{a}_3}{|\mathbf{a}_2 \times \mathbf{a}_3|} \qquad \mathbf{r}_3 = \frac{\pm\mathbf{a}_3}{|\mathbf{a}_3|}$$

$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1 \qquad \mathbf{T} = \rho\mathcal{K}^{-1}\mathbf{b}$$

Source: S Savarese slides.

Degenerate cases

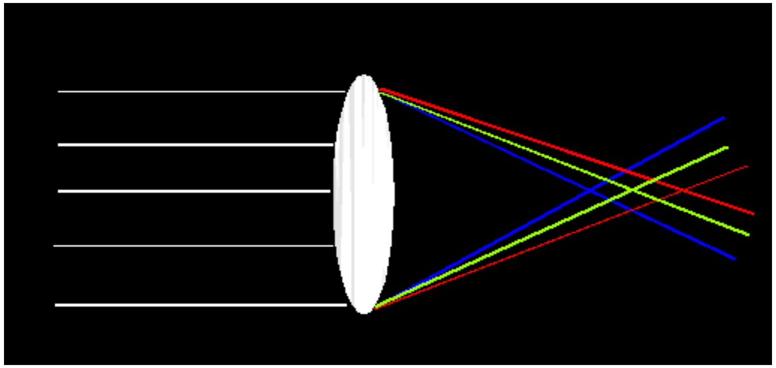


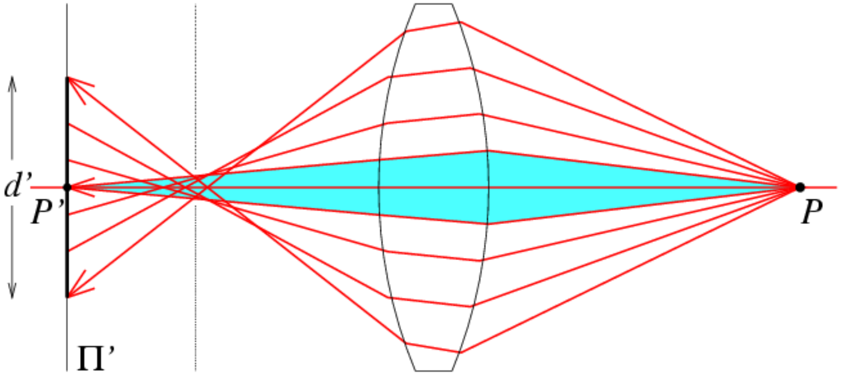
- P_i's cannot lie on the same plane!
- Points cannot lie on the intersection curve of two quadric surfaces

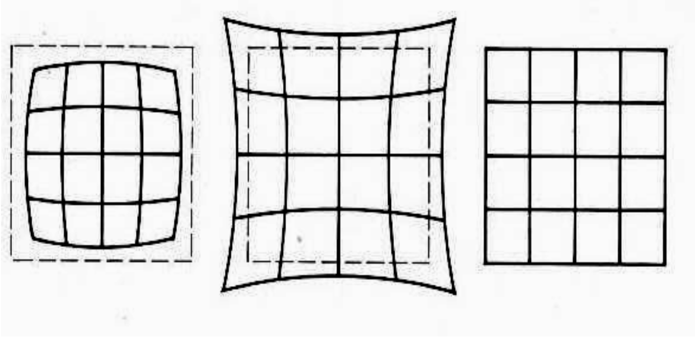
Source: S Savarese slides.

Taking lens distortions into account

- Chromatic Aberration
- Spherical aberration
- Radial Distortion



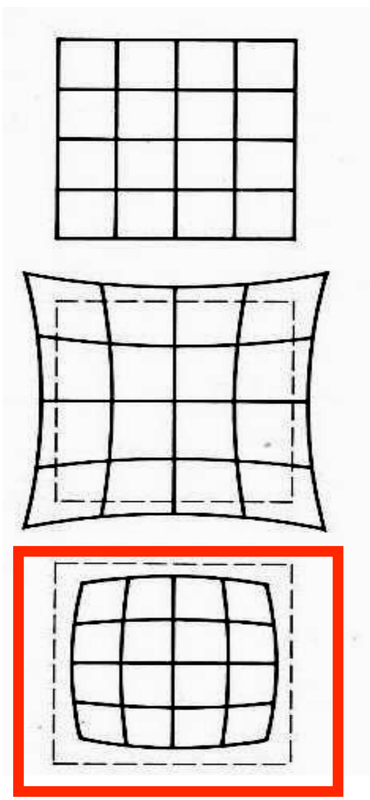




Source: S Savarese slides.

Dealing with Radial Distortion As Well

– Caused by imperfect lenses
– Deviations are most noticeable for rays that pass through the edge of the lens



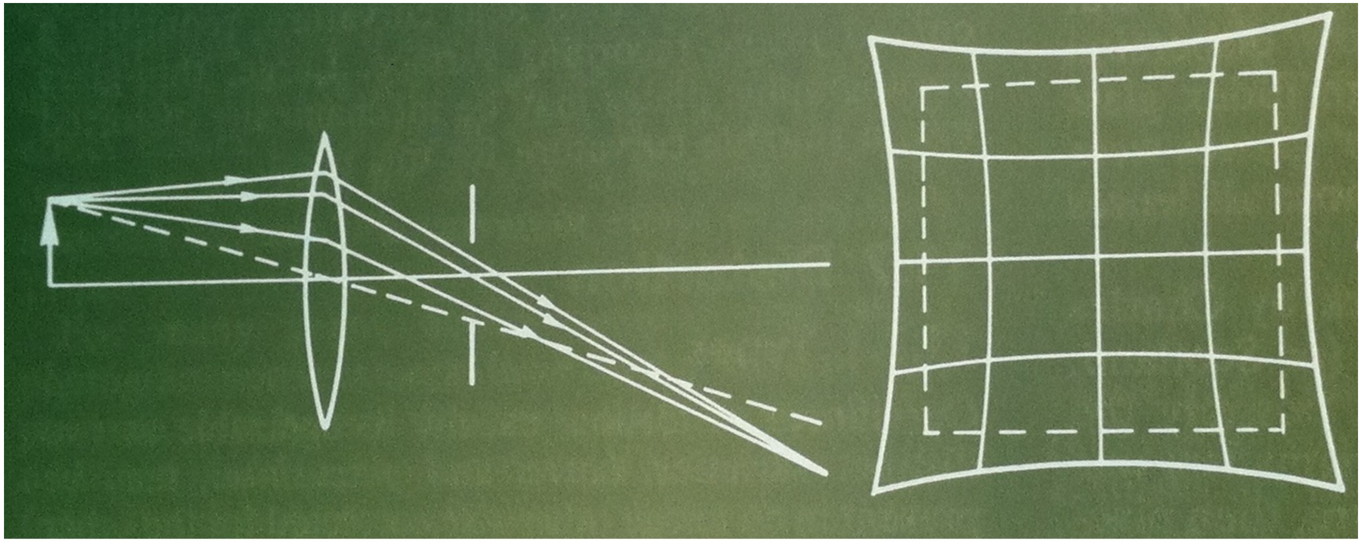
No distortion

Pin cushion

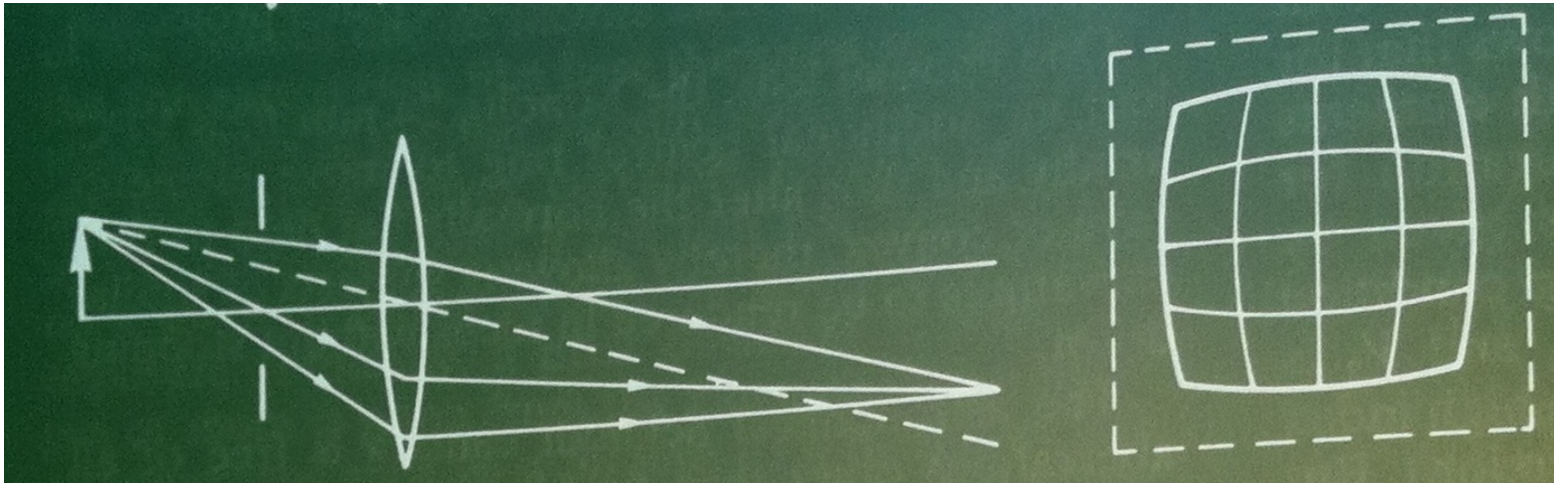
Barrel

Source: S Savarese slides.

Issues with lenses: Radial Distortion



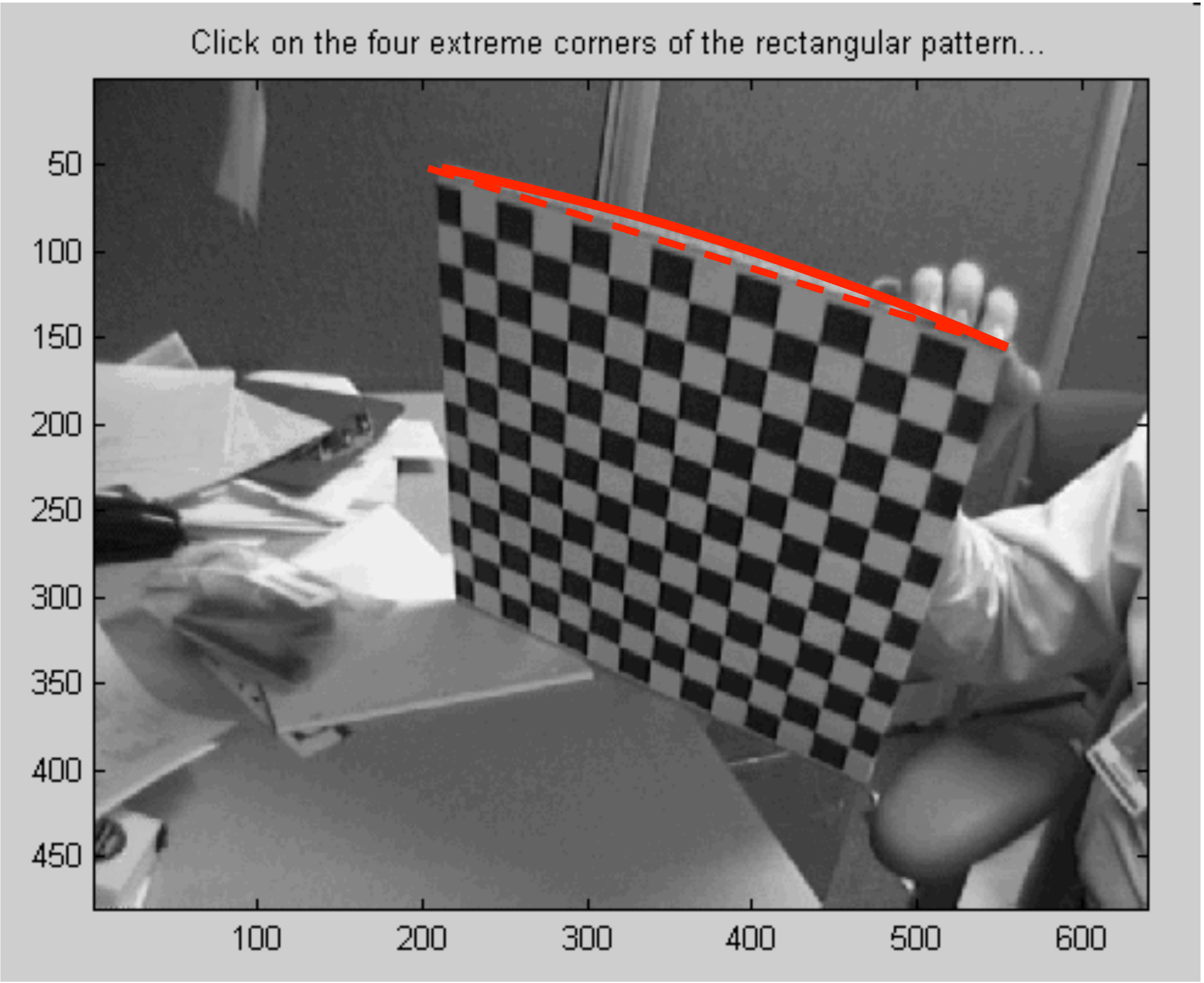
Pin cushion



Barrel (fisheye lens)

Source: S Savarese slides.

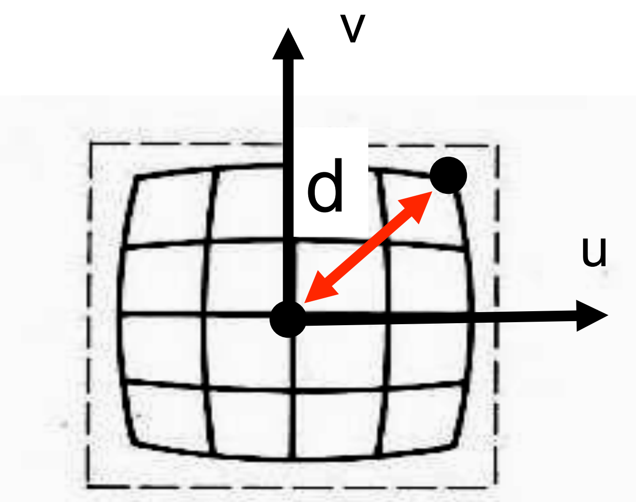
Radial Distortion



Source: S Savarese slides.

Radial Distortion

Image magnification in(de)creases with distance from the optical center



$$\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M\, P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i$$

$$d^2 = a\,u^2 + b\,v^2 + c\,u\,v$$

To model radial behavior

Distortion coefficient

$$\underbrace{\lambda = 1 \pm \sum_{p=1}^{3} \boxed{\kappa_p} d^{2p}}_{\text{Polynomial function}}$$

Source: S Savarese slides.

Radial Distortion

$$\underbrace{\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{Q} M P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i \qquad Q = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

Is this a linear system of equations?

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 P_i}{\mathbf{q}_3 P_i} \\ \dfrac{\mathbf{q}_2 P_i}{\mathbf{q}_3 P_i} \end{bmatrix} \rightarrow \begin{cases} u_i \mathbf{q}_3 P_i = \mathbf{q}_1 P_i \\ v_i \mathbf{q}_3 P_i = \mathbf{q}_2 P \end{cases}$$

No! why?

Source: S Savarese slides.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\[2ex] \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

-Newton Method

-Levenberg-Marquardt Algorithm

- Iterative, starts from initial solution
- May be slow if initial solution far from real solution
- Estimated solution may be function of the initial solution
- Newton requires the computation of J, H
- Levenberg-Marquardt doesn't require the computation of H

Source: S Savarese slides.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

A possible algorithm

1. Solve linear part of the system to find approximated solution
2. Use this solution as initial condition for the full system
3. Solve full system using Newton or L.M.

Source: S Savarese slides.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P}_\mathrm{i}}{\mathbf{q}_3 \, \mathrm{P}_\mathrm{i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P}_\mathrm{i}}{\mathbf{q}_3 \, \mathrm{P}_\mathrm{i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

Typical assumptions:

- zero-skew, square pixel
- u_o, v_o = known center of the image
- no distortion

$\longrightarrow$ Just estimate f and R, T

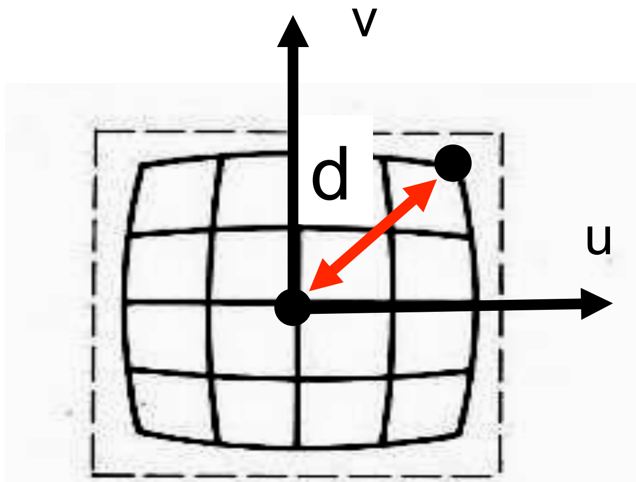
Source: S Savarese slides.

Radial Distortion

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \mathrm{P}_i}{\mathbf{q}_3 \mathrm{P}_i} \\ \dfrac{\mathbf{q}_2 \mathrm{P}_i}{\mathbf{q}_3 \mathrm{P}_i} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 \mathrm{P}_i}{\mathbf{m}_3 \mathrm{P}_i} \\ \dfrac{\mathbf{m}_2 \mathrm{P}_i}{\mathbf{m}_3 \mathrm{P}_i} \end{bmatrix}$$

Can estimate m_1 and m_2 and ignore the radial distortion?

Hint:



$$\frac{u_i}{v_i} = \text{slope}$$

Source: S Savarese slides.

Radial Distortion

Estimating m_1 and m_2…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix} \qquad \frac{u_i}{v_i} = \frac{\dfrac{(\mathbf{m}_1 P_i)}{(\mathbf{m}_3 P_i)}}{\dfrac{(\mathbf{m}_2 P_i)}{(\mathbf{m}_3 P_i)}} = \frac{\mathbf{m}_1 P_i}{\mathbf{m}_2 P_i}$$

$$\begin{cases} v_1(\mathbf{m}_1 P_1) - u_1(\mathbf{m}_2 P_1) = 0 \\ v_i(\mathbf{m}_1 P_i) - u_i(\mathbf{m}_2 P_i) = 0 \\ \vdots \\ v_n(\mathbf{m}_1 P_n) - u_n(\mathbf{m}_2 P_n) = 0 \end{cases} \qquad Q\,\mathbf{n} = 0 \qquad \mathbf{n} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}$$

Tsai technique [87]

Source: S Savarese slides.

Radial Distortion

Once that $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

$\mathbf{m}_3$ is non linear function of $\mathbf{m}_1, \mathbf{m}_2, \lambda$

There are some degenerate configurations for which m_1 and m_2 cannot be computed

Source: S Savarese slides.

Direct Calibration: The Algorithm

1. Compute image center from orthocenter
2. Compute the Intrinsic matrix (6.8)
3. Compute solution with SVD
4. Compute gamma and alpha
5. Compute R (and normalize)
6. Compute f_x and and T_z

Source: G Hager slides.

Basic Equations

$$^{c}T_{w} = (T_x, T_y, T_z)'$$

$$^{c}R_{w} = (R_x, R_y, R_z)'$$

$$^{c}p = {}^{c}R_{w}\,{}^{w}p + {}^{c}T_{w}$$

$$u = -f\frac{R_x p + T_x}{R_z p + T_z}$$

$$v = -f\frac{R_y p + T_y}{R_z p + T_z}$$

Source: G Hager slides.

Basic Equations

$$u_{pix} = \frac{1}{s_x} u + o_x$$

$$v_{pix} = \frac{1}{s_y} v + o_y$$

$$\bar{u} = u_{pix} - o_x = -f_x \frac{R_x p + T_x}{R_z p + T_z}$$

$$\bar{v} = v_{pix} - o_y = -f_y \frac{R_y p + T_y}{R_z p + T_z}$$

Source: G Hager slides.

Basic Equations

$$\bar{u}_i f_y (R_y p_i + T_y) = \bar{v}_i f_x (R_x p_i + T_x)$$
$$\bar{u}_i (R_y p_i - T_y) - \bar{v}_i \alpha (R_x p_i + T_x) = 0$$

$r = \alpha R_x$ and $w = \alpha T_x$

$t = R_y$ and $s = T_y$

one of these for each point

$A_i = (u_i p_i, u_i, -v_i p_i, -v_i)$ and $A[t, s, w, r]' = 0$

Source: G Hager slides.

Basic Equations

$A_i = (u_i p_i, u_i, -v_i p_i, -v_i)$ and
$A[t, s, w, r]' = Am = 0$

Note that m is defined up a scale factor!

$A = UDV'$ and choose m as column of V corresponding to the smallest singular value

Source: G Hager slides.

Properties of SVD Again

- Recall the singular values of a matrix are related to its rank.
- Recall that Ax = 0 can have a nonzero x as solution only if A is singular.
- Finally, note that the matrix V of the SVD is an orthogonal basis for the domain of A; in particular the zero singular values are the basis vectors for the null space.

- Putting all this together, we see that A must have rank 7 (in this particular case) and thus x must be a vector in this subspace.

- Clearly, x is defined only up to scale.

Source: G Hager slides.

Basic Equations

$$A_i = (u_i p_i, u_i, -v_i p_i, -v_i) \text{ and}$$
$$A[t, s, w, r]' = Am = 0$$

$$||t|| = |\gamma| \text{ gives scale factor for solution}$$
$$||w|| = |\gamma|\alpha$$

We now know R_x and R_y up to a sign and gamma.
$R_z = R_x \times R_y$

We will probably use another SVD to orthogonalize this system (R = U D V'; set D to I and multiply).

Source: G Hager slides.

Last Details about Direct Calibration

- We still need to compute the correct sign.
  - note that the denominator of the original equations must be positive (points must be in front of the cameras)
  - Thus, the numerator and the projection must disagree in sign.
  - We know everything in numerator and we know the projection, hence we can determine the sign.

- We still need to compute T_z and f_x
  - we can formulate this as a least squares problem on those two values using the first equation.

$$\bar{u} = -f_x \frac{R_x p + T_x}{R_z p + T_z} \rightarrow$$

$$\bar{u}(R_z p + T_z) = -f_x(R_x p + T_x)$$

$$f_x(R_x p + T_x) + \bar{u} T_z = -\bar{u} R_z p$$

$$A(f_x, T_z)' = b \rightarrow (f_x, T_z)' = (A'A)^{-1} A' b$$

Source: G Hager slides.

Self-Calibration

- Calculate the intrinsic parameters solely from point correspondences from multiple images.
- Static scene and intrinsics are assumed.
- No expensive apparatus.
- Highly flexible but not well-established.
- Projective Geometry – image of the absolute conic.

Source: G Hager slides.

Multi-Plane Calibration

- Hybrid method: Photogrammetric and Self-Calibration.
- Uses a planar pattern imaged multiple times (inexpensive).
- Used widely in practice and there are many implementations.
- Based on a group of projective transformations called homographies.

- m be a 2d point [u v 1]' and M be a 3d point [x y z 1]'.

- Projection is $$s\tilde{m} = A[R \quad T]\tilde{M}$$

Source: G Hager slides.

Planar Homographies

- First Fundamental Theorem of Projective Geometry:
  - There exists a unique homography that performs a change of basis between two projective spaces of the same dimension.

$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad Z \quad 1]^T$$
$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad 0 \quad 1]^T$$
$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad t][X \quad Y \quad 1]^T$$
$$s[u \quad v \quad 1]^T = H[X \quad Y \quad 1]^T$$

  - Projection Becomes

$$s\tilde{m} = H\tilde{M}$$

  - Notice that the homography is defined up to scale (s).

Source: G Hager slides.

Computing the Intrinsics

- We know that $$[h_1 \quad h_2 \quad h_3] = sA[r_1 \quad r_2 \quad t]$$

- From one homography, how many constraints on the intrinsic parameters can we obtain?
  - Extrinsics have 6 degrees of freedom.
  - The homography has 8 degrees of freedom.
  - Thus, we should be able to obtain 2 constraints per homography.

- Use the constraints on the rotation matrix columns…

Source: G Hager slides.

Computing Intrinsics

- Rotation Matrix is orthonormal:

$$r_i^T r_j = 0$$

$$r_i^T r_i = r_j^T r_j$$

- Write the homography in terms of its columns…

$$\begin{aligned} h_1 &= sAr_1 \\ h_2 &= sAr_2 \\ h_3 &= sAt \end{aligned}$$

Source: G Hager slides.

Computing Intrinsics

- Derive the two constraints:

$$
\begin{aligned}
h_1 &= sAr_1 \\
\frac{1}{s}A^{-1}h_1 &= r_1 \\
\frac{1}{s}A^{-1}h_2 &= r_2
\end{aligned}
$$

$$
\begin{aligned}
r_1^T r_2 &= 0 \\
\boxed{h_1^T A^{-T} A^{-1} h_2 = 0}
\end{aligned}
$$

$$
\begin{aligned}
r_1^T r_1 &= r_2^T r_2 \\
\boxed{h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2}
\end{aligned}
$$

Source: G Hager slides.

Closed-Form Solution

$$\text{Let } B = A^{-T}A^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta))}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta))}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

- Notice B is symmetric, 6 parameters can be written as a vector b.
- From the two constraints, we have $h_i^T B h_j = v_{ij}^T$

$$\begin{bmatrix} v_{ij}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0;$$

- Stack up n of these for n images and build a 2n*6 system.
- Solve with SVD (yet again).
- Extrinsics “fall-out” of the result easily.

Source: G Hager slides.

Non-linear Refinement

- Closed-form solution minimized algebraic distance.
- Since full-perspective is a non-linear model
  - Can include distortion parameters (radial, tangential)
  - Minimize squared distance with a non-linear method.

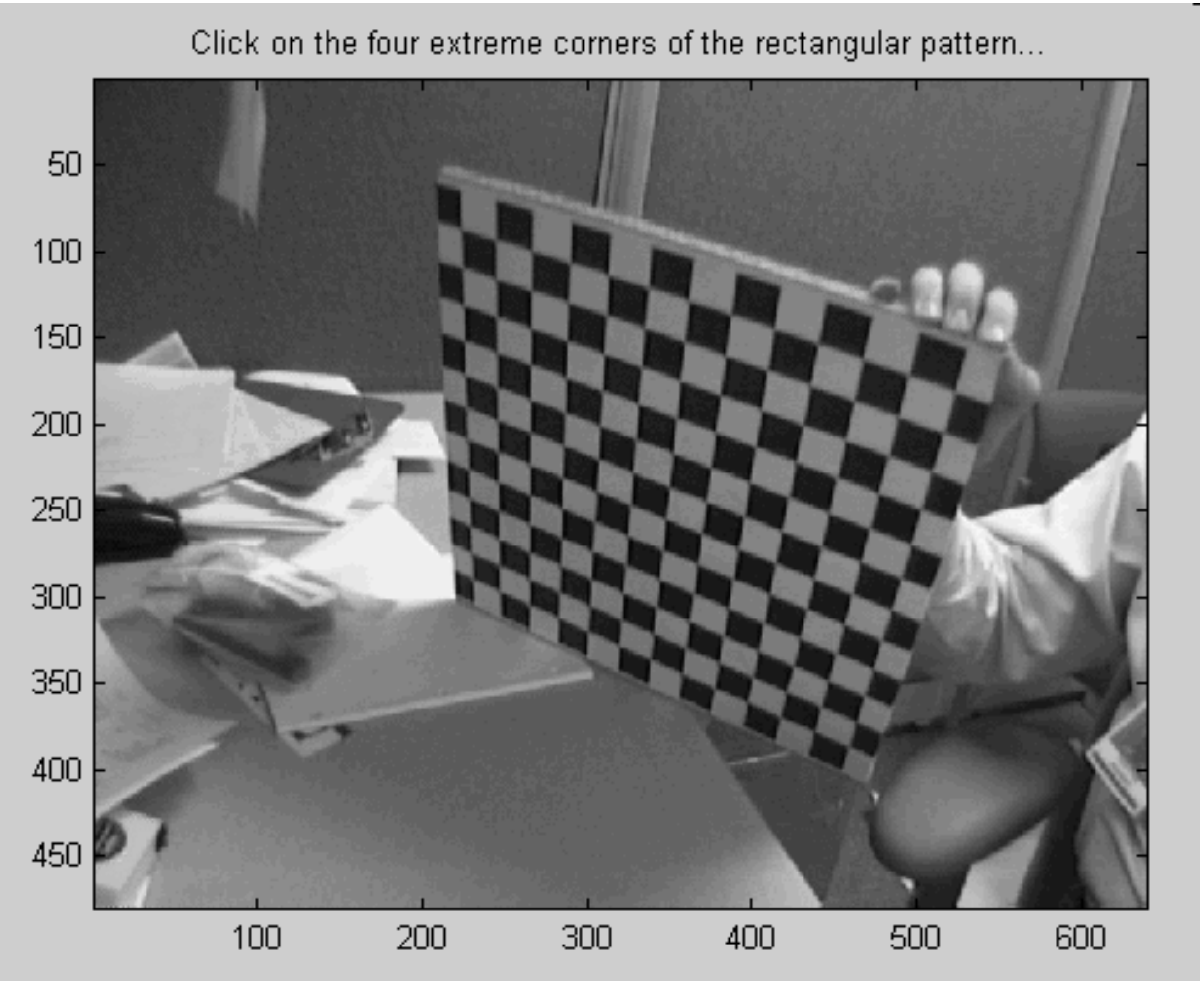
$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \hat{m}(A, R_k, T_k, M_j)||^2$$

Source: G Hager slides.

Example Calibration Procedure

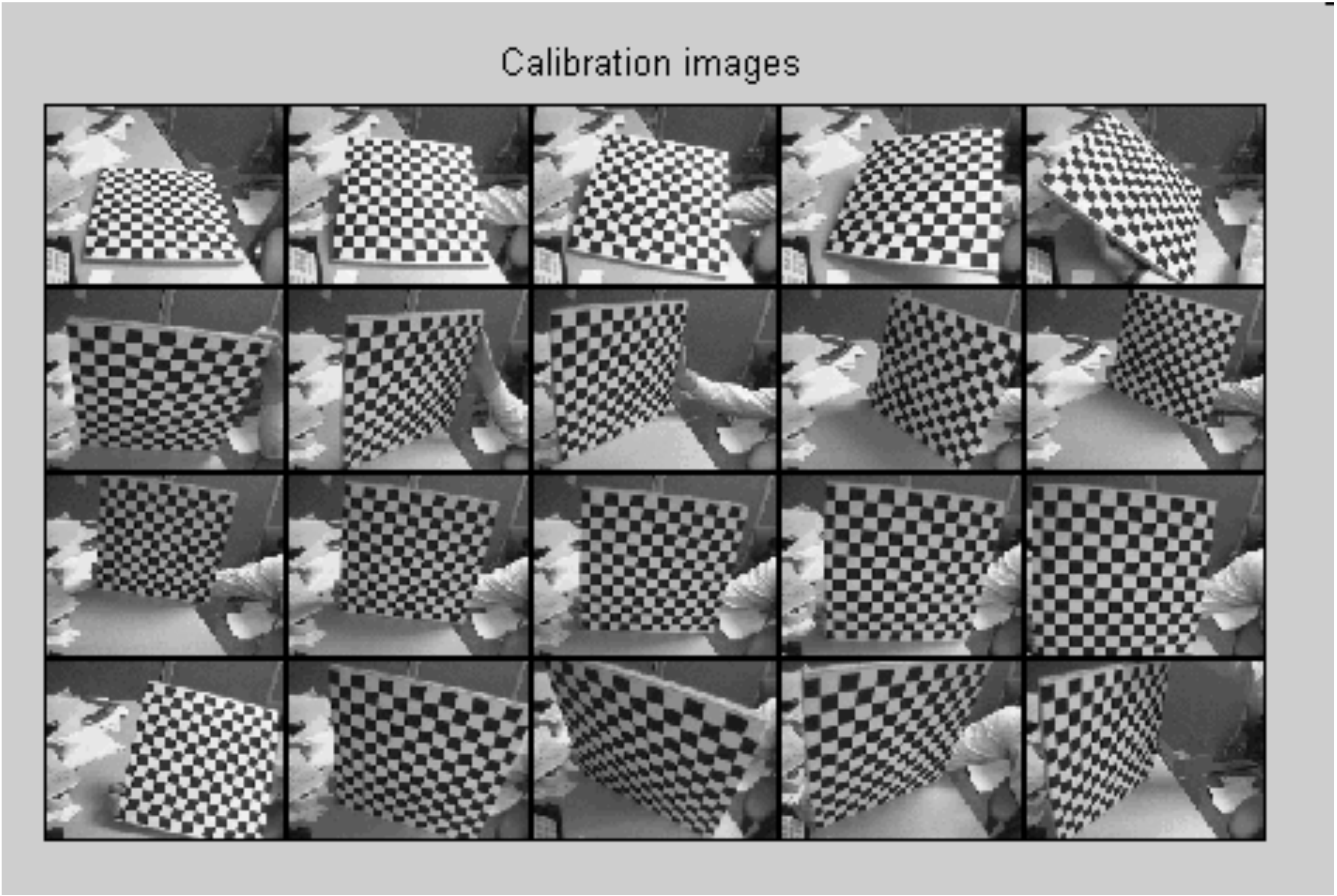
Camera Calibration Toolbox for Matlab
J. Bouguet – [1998-2000]

http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#examples



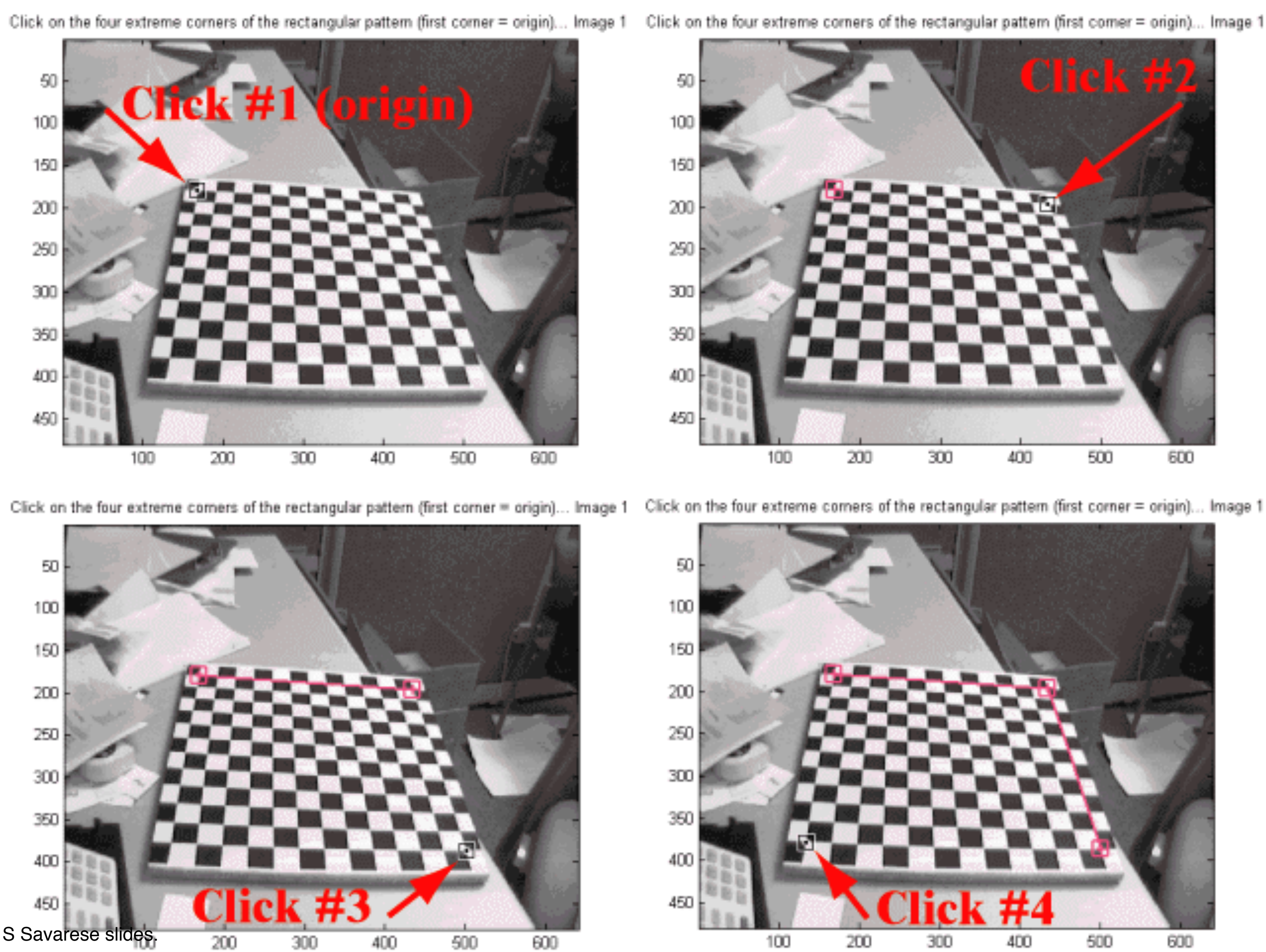
Source: S Savarese slides.

Example Calibration Procedure



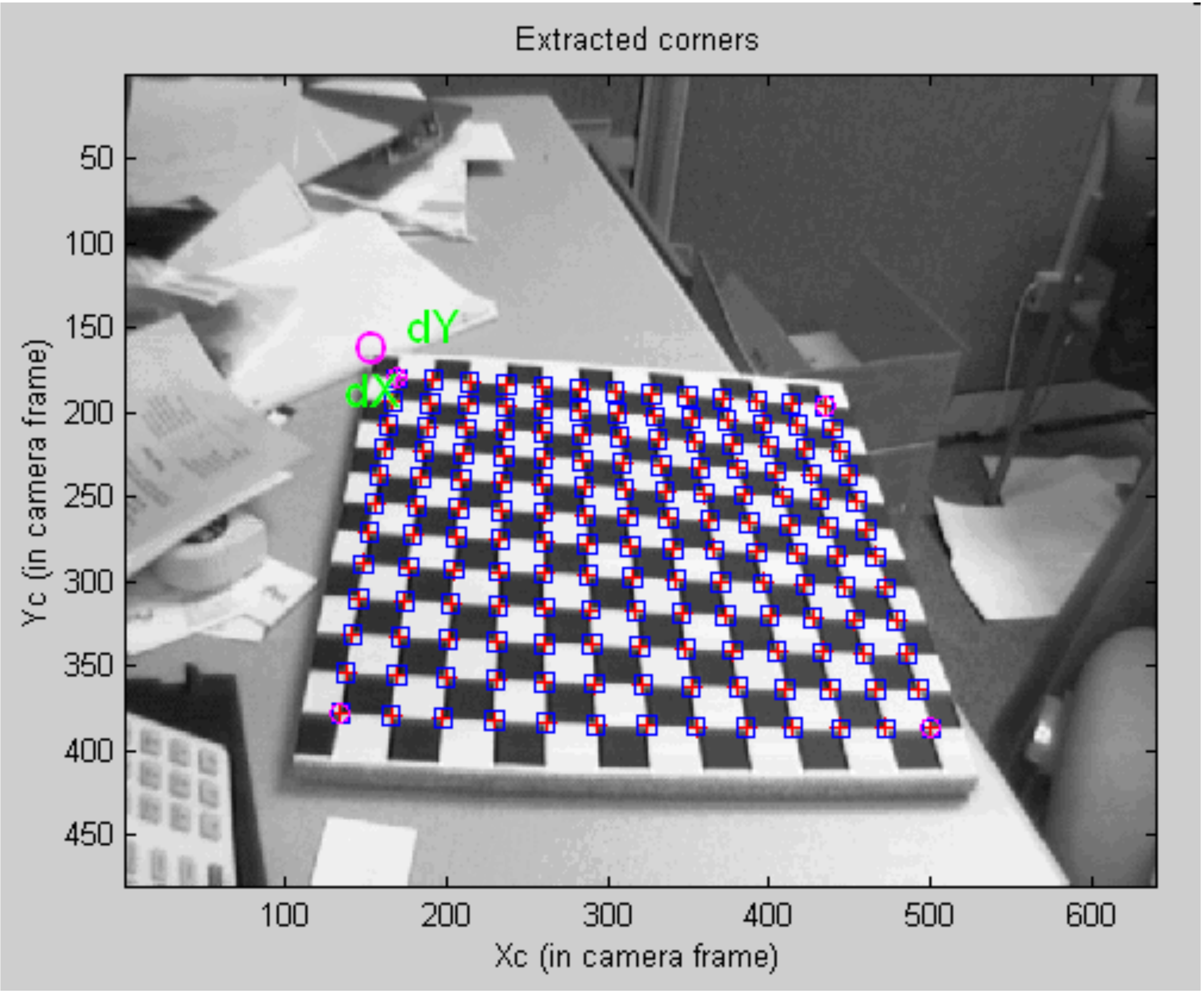
Source: S Savarese slides.

Example Calibration Procedure



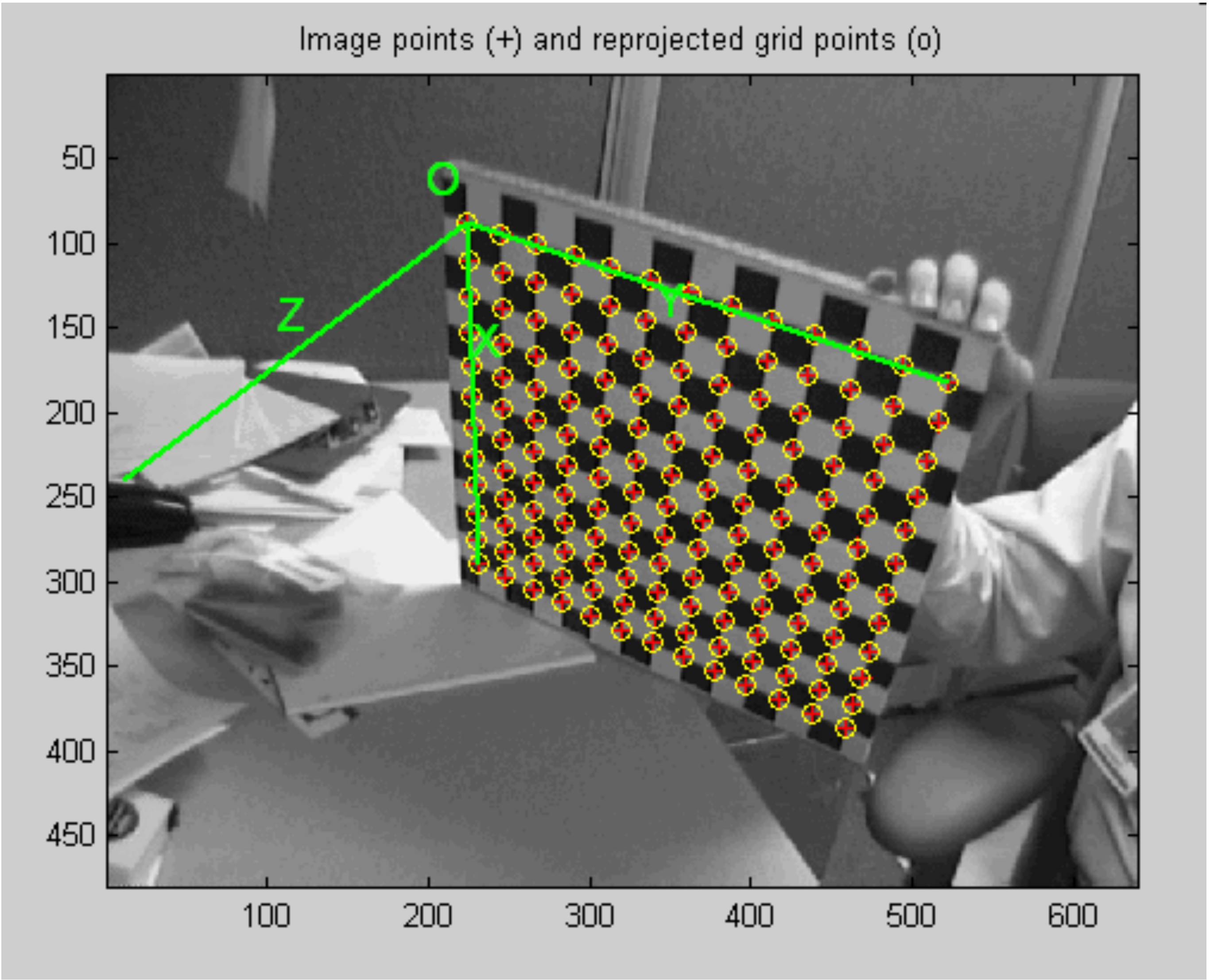
Source: S Savarese slides.

Example Calibration Procedure



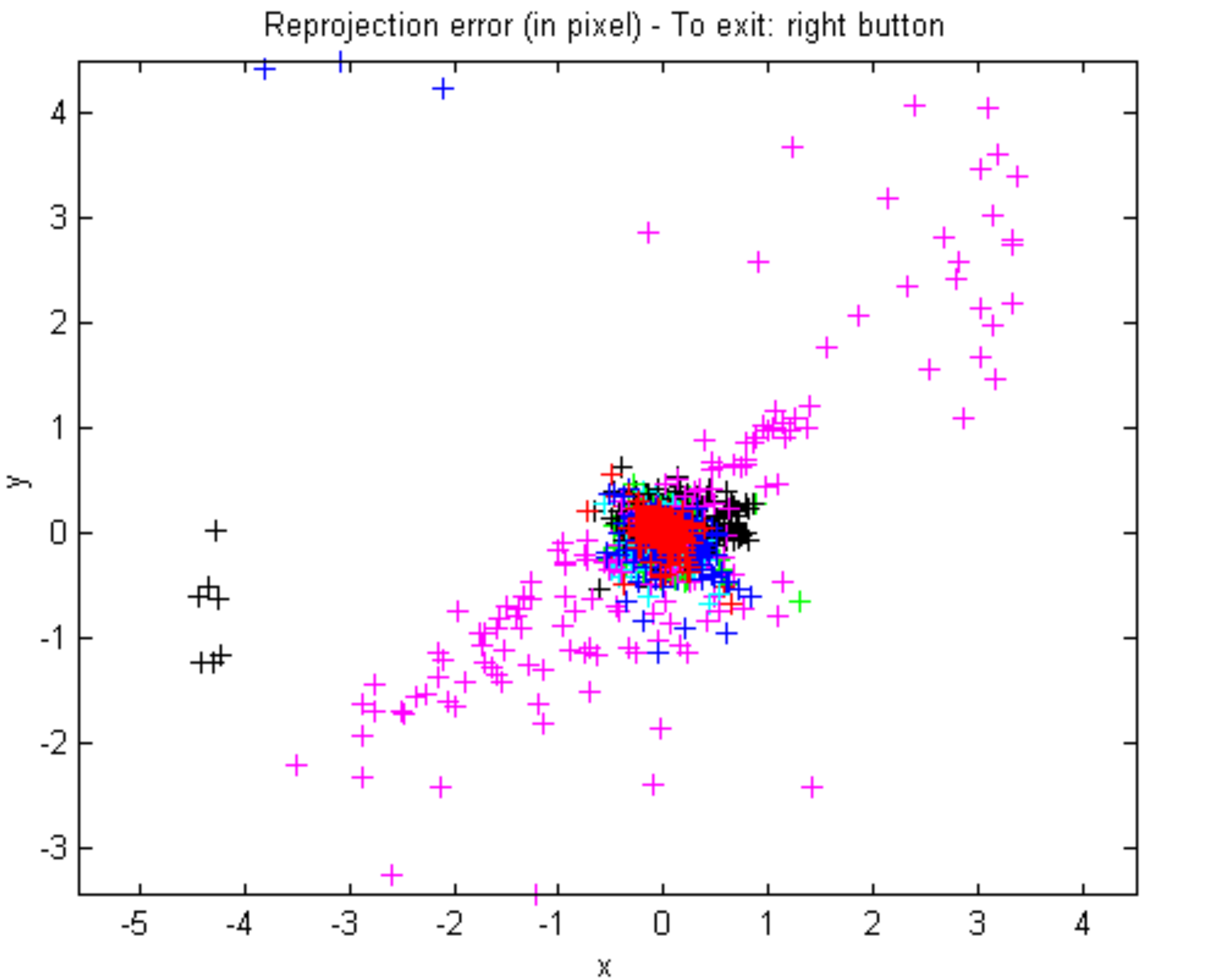
Source: S Savarese slides.

Example Calibration Procedure



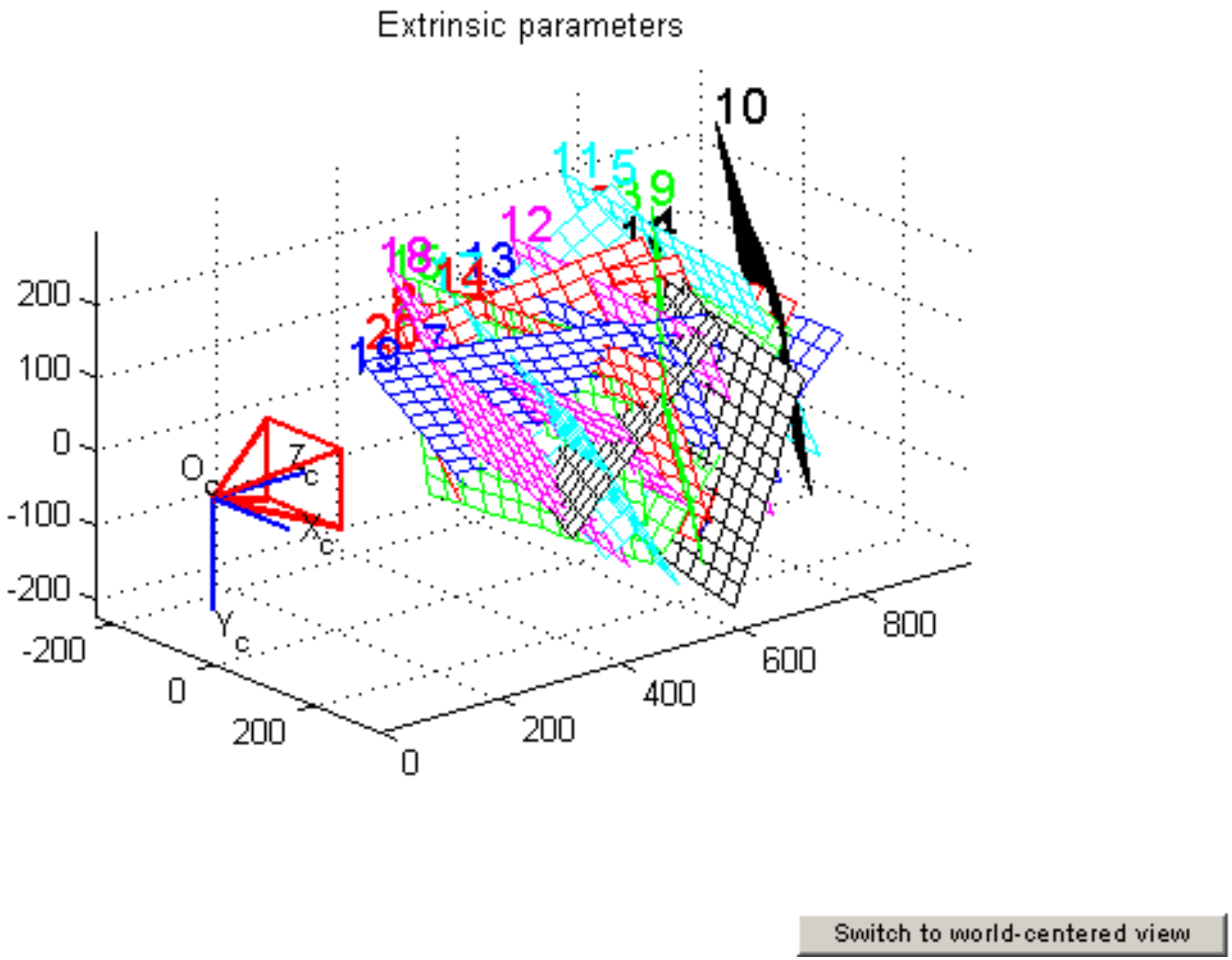
Source: S Savarese slides.

Example Calibration Procedure



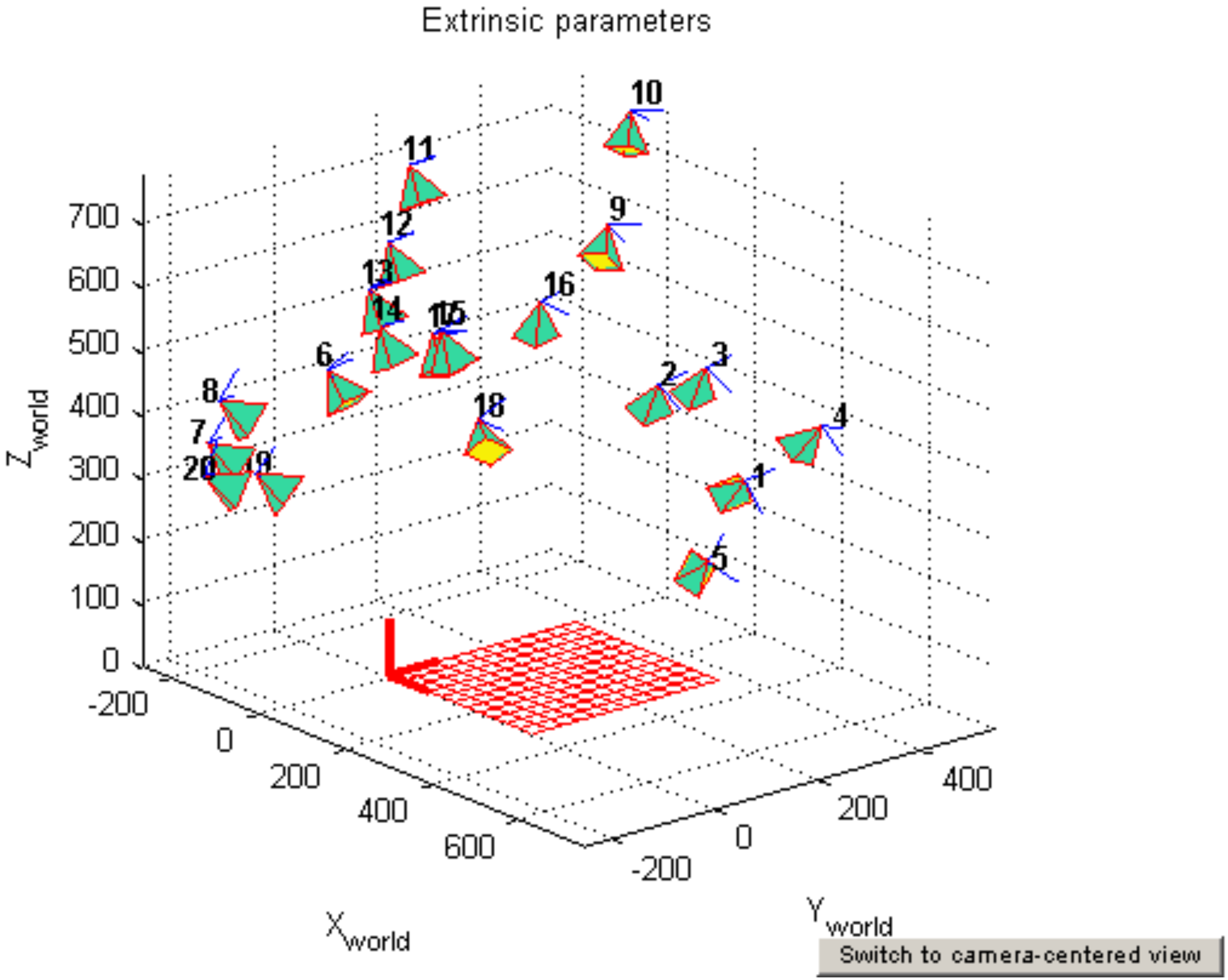
Source: S Savarese slides.

Example Calibration Procedure



Source: S Savarese slides.

Example Calibration Procedure



Source: S Savarese slides.

Next Lecture: Photometric and Radiometric Aspects

- Reading: FP 2, 3; SZ 2.2, 2.3