UNIVERSITÀ DI PARMA

Camera Models (2)

Summary

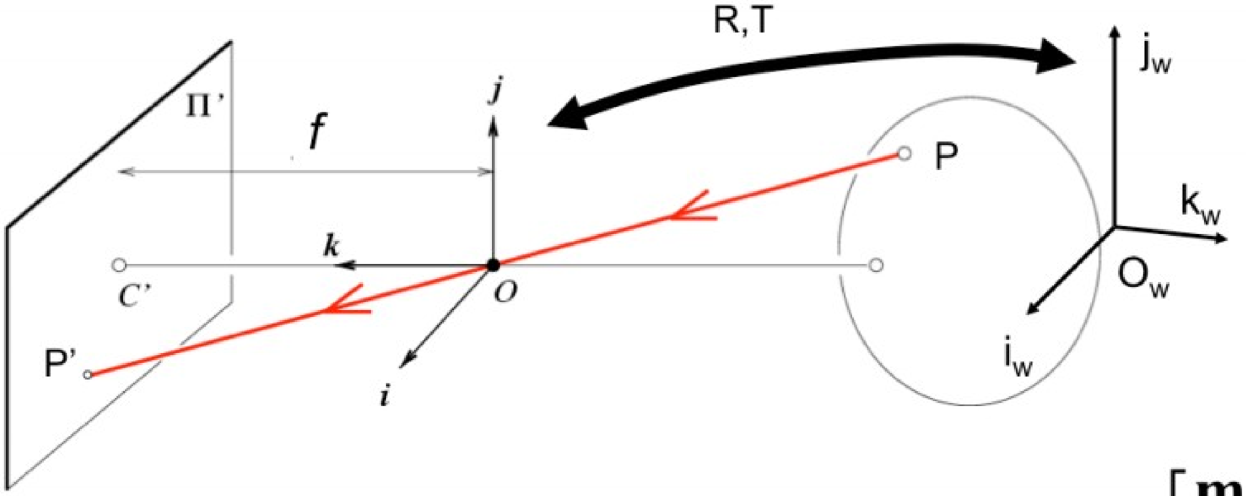
- Pin Hole Camera recap
- Calibration

UNIVERSITÀ DI PARMA

Pin Hole Camera recap

Perspective Transformation

UNIVERSITÀ DI PARMA



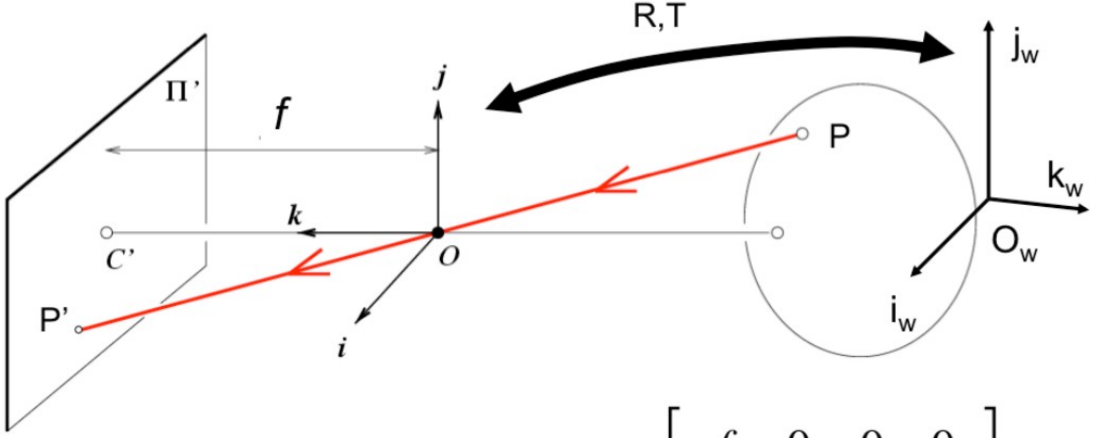
$$P'_{3\times1} = M\,P_w \quad = K_{3\times3}\begin{bmatrix} R & T \end{bmatrix}_{3\times4} P_{w\,4\times1} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ \mathbf{m}_3 P_W \end{bmatrix} \xrightarrow{\mathbf{E}} \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right) \quad \text{[Eq.12]}$$

Courtesy: Silvio Savarese

Perspective Transformation (ideal case)

- Simplified situation: no rotation, no translation, no skew, no offset, squared pixels



$$M = K\begin{bmatrix} R & T \end{bmatrix} = K\begin{bmatrix} I & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\rightarrow P'_E = \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right) = \left(f\frac{x_w}{z_w}, f\frac{y_w}{z_w}\right) \qquad P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Courtesy: Silvio Savarese

Perspective Transformation

- 11 degrees of freedom

$$P' = M\,P_w = \underbrace{K}_{\text{Internal parameters}}\underbrace{[R \quad T]}_{\text{External parameters}}P_w$$

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

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

Courtesy: Silvio Savarese

UNIVERSITÀ DI PARMA

Calibration
Tsai approach (1987)

Calibration

$$P' = M\,P_w = \underbrace{K}_{\text{Internal parameters}}\underbrace{\begin{bmatrix} R & T \end{bmatrix}}_{\text{External parameters}} P_w$$

- Calibration → **estimation of intrinsic/extrinsic parameters**
- We need 1 or, rather, more images

Change notation:
$P = P_w$

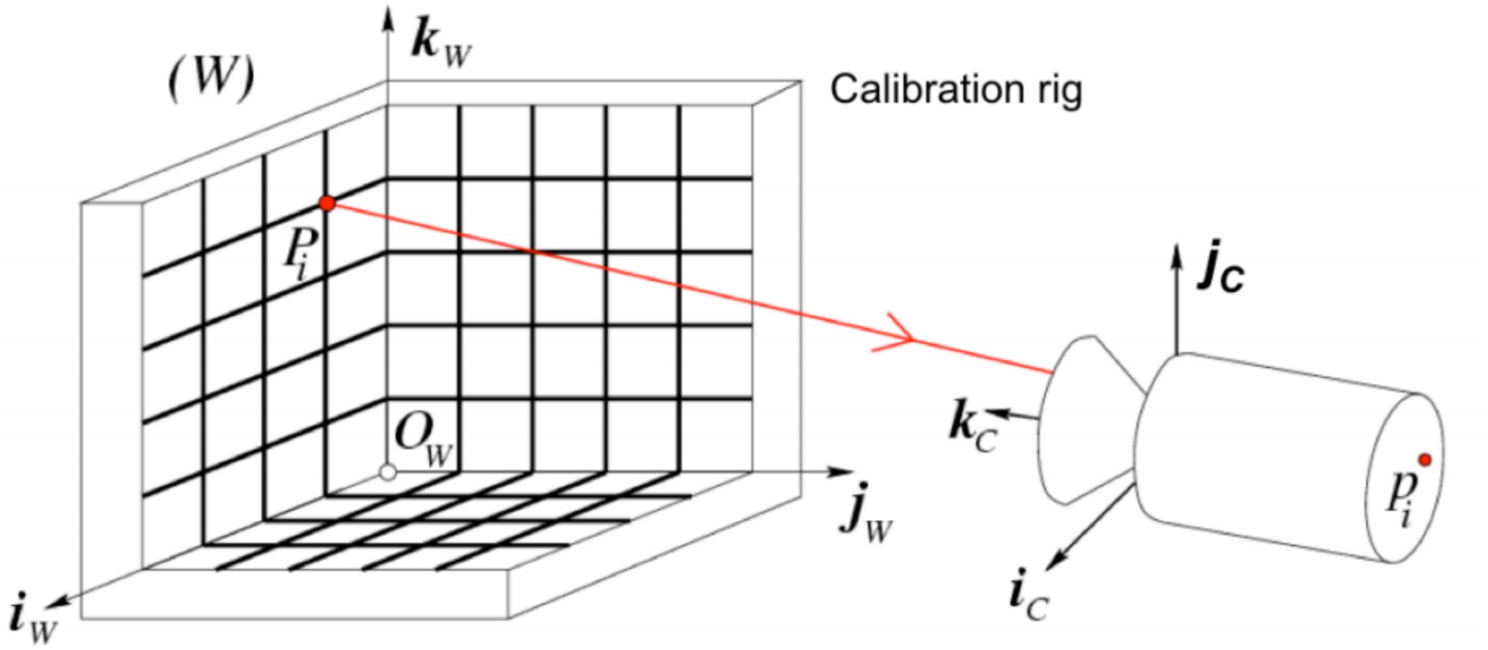
$p = P'$

Calibration

- Small notation change
  - $P_W \quad \rightarrow \quad P$
  - $P' \quad \rightarrow \quad p$

$$p = K\left[R\;T\right]P = MP$$

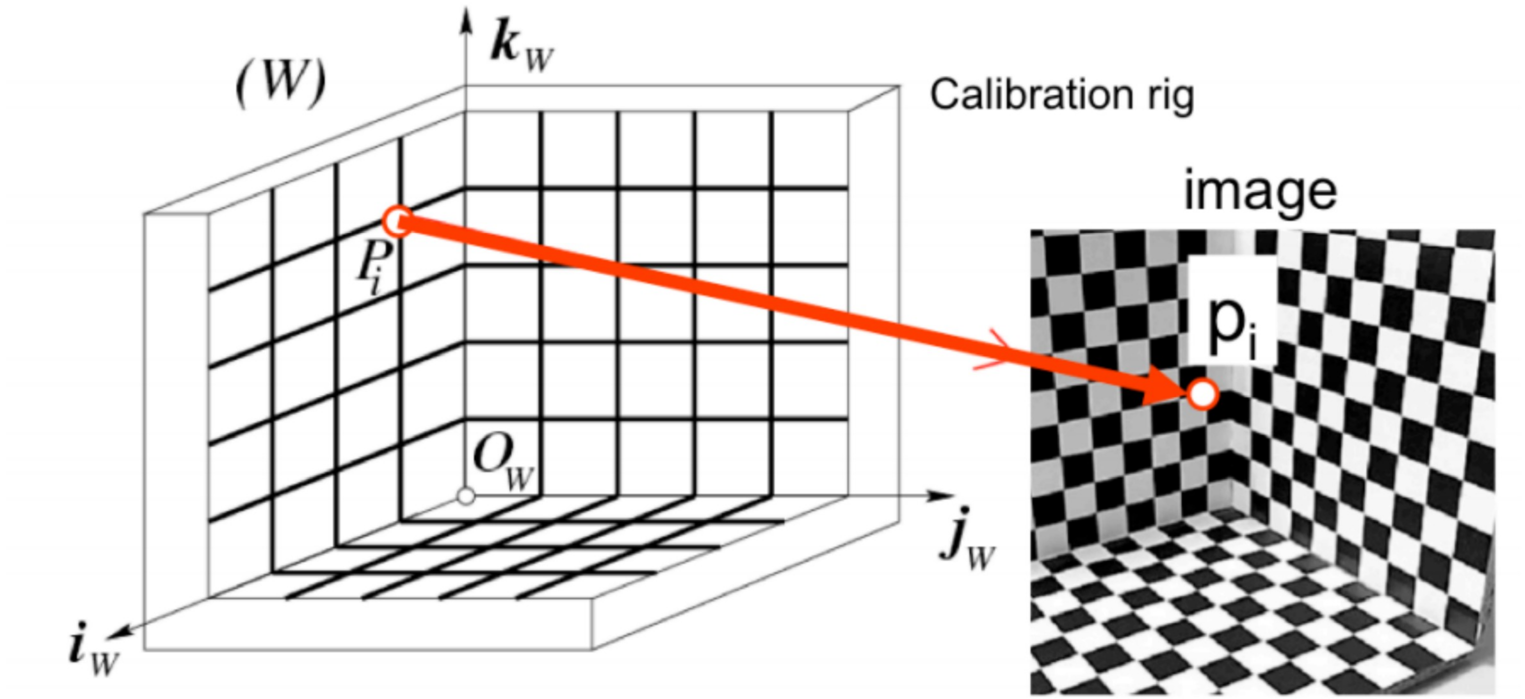
Calibration



- We need: n points (P_1, P_2,…,P_n) in known positions [O_w,i_w,j_w,k_w] in the world reference system

Courtesy: Silvio Savarese

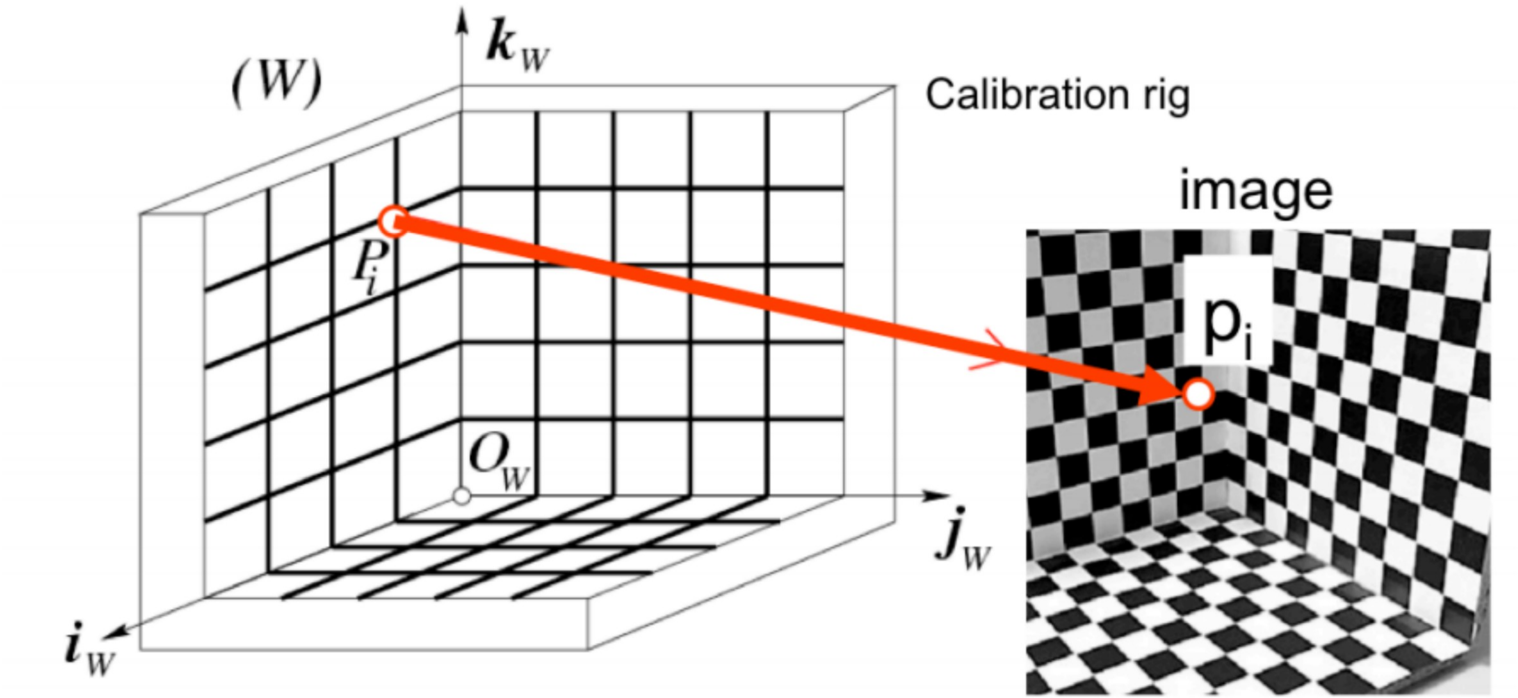
Calibration



- We also need: camera coordinates for the n points projections (p_1, p_2,…,p_n) as (i_c,j_c)

Courtesy: Silvio Savarese

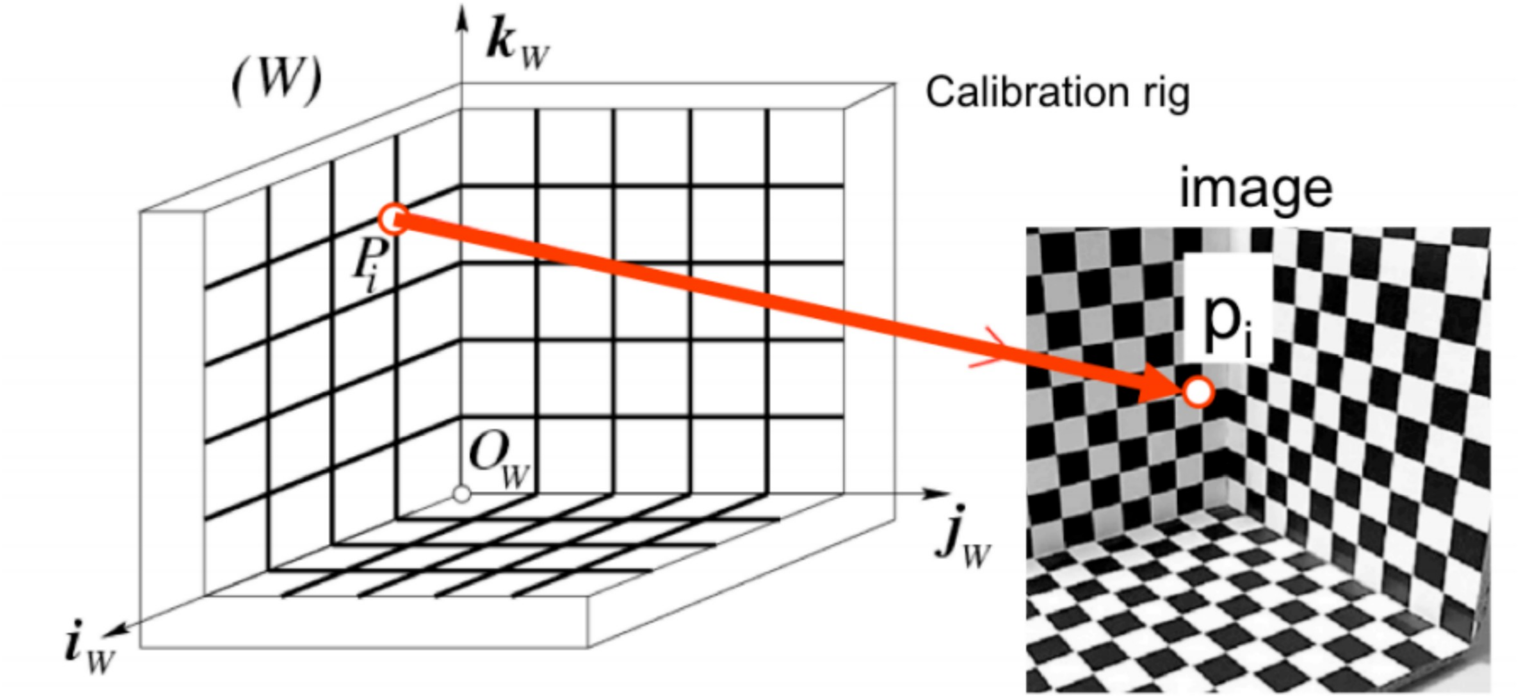
Calibration



- 11 degrees of freedom → how many matches we need?
- Each correspondence → 2 equations (world coordinates → row/column)
- We need, at least, 6 correspondences

Courtesy: Silvio Savarese

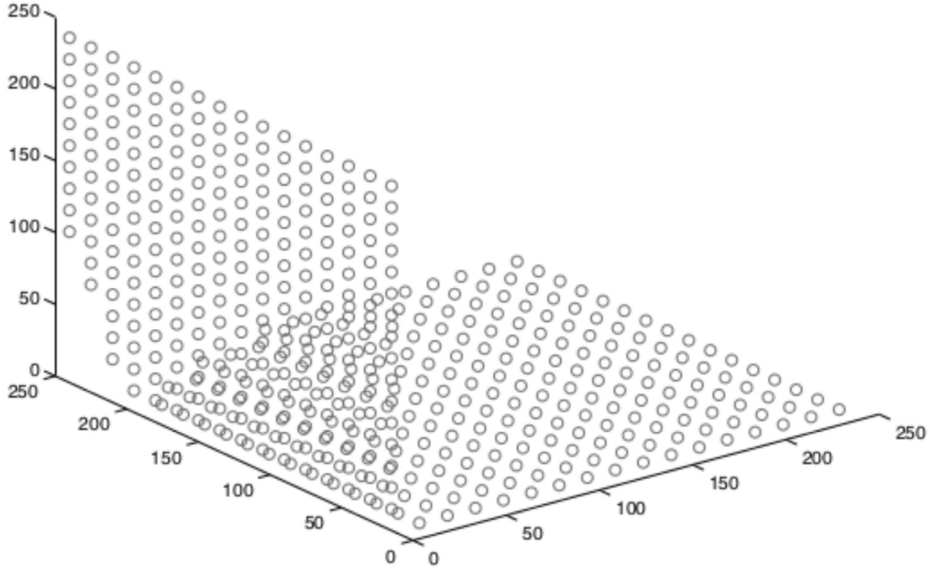
Calibration

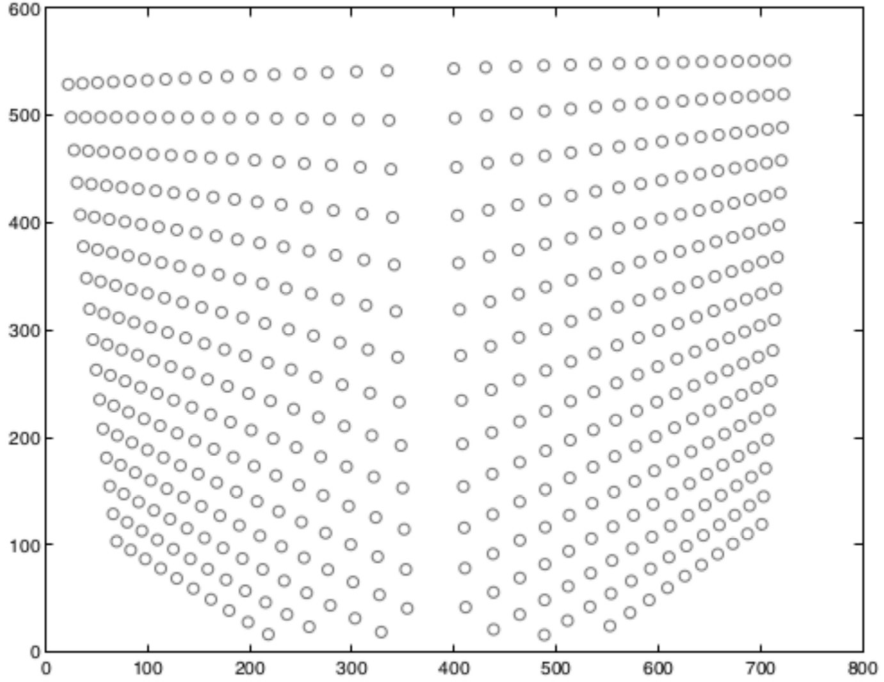


- Practically we need more than 6 correspondences!
- Consider error in measurements + digitalization error!

Courtesy: Silvio Savarese

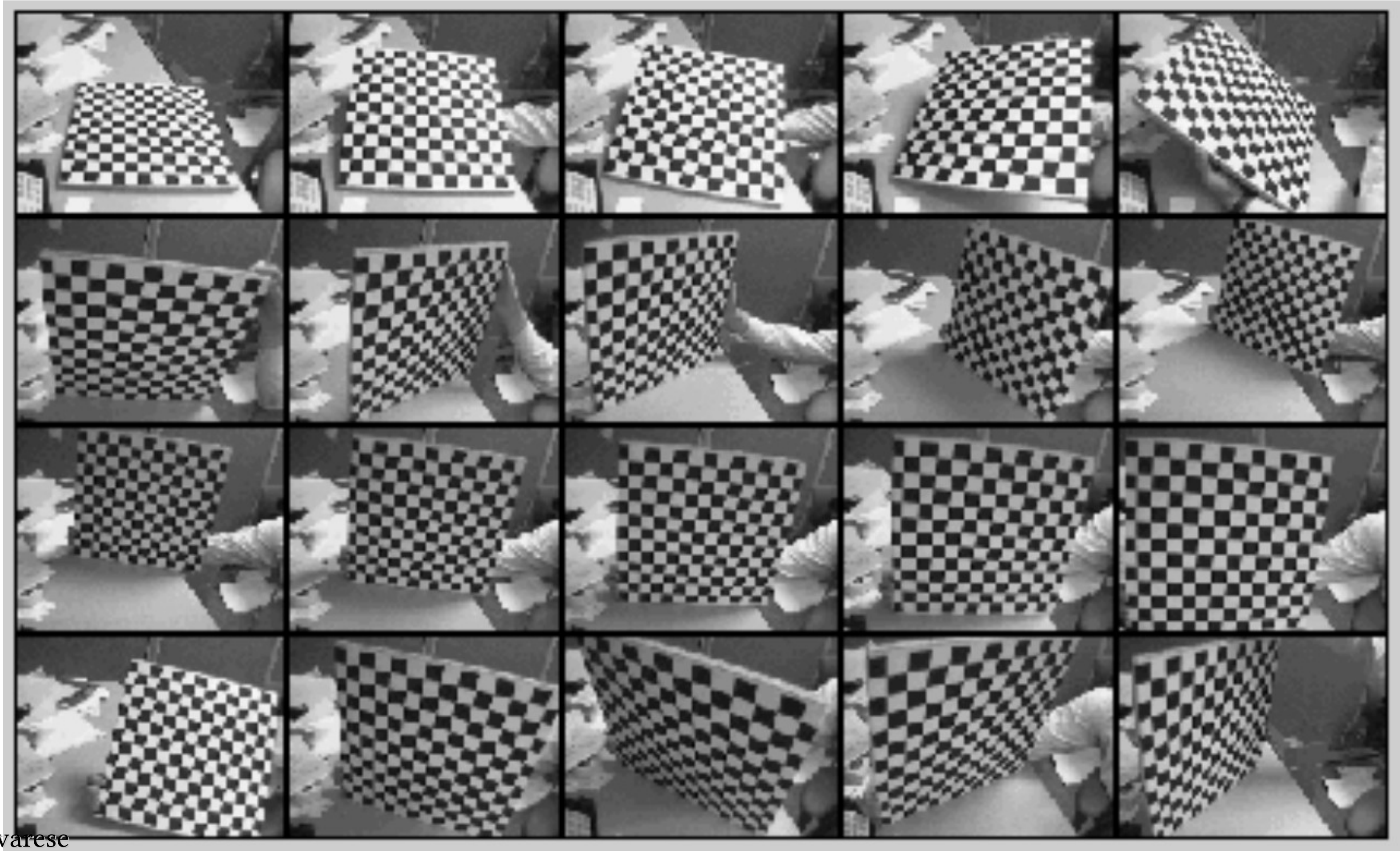
Example of calibration data





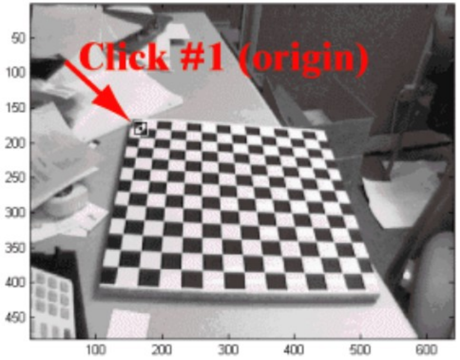
- Example of 491 3D points
  - From: Janne Heikkilä; data copyright 2000, University of Oulu.

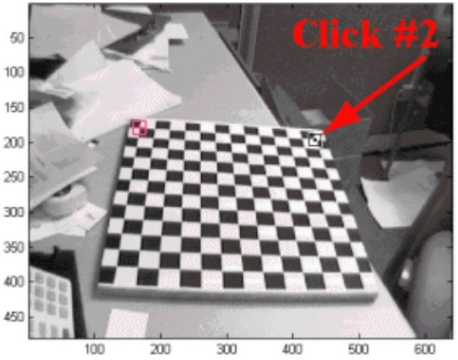
Example of calibration data

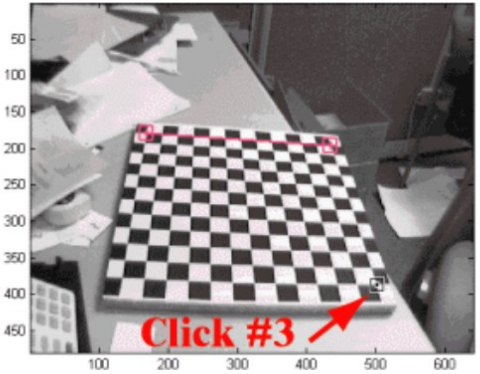


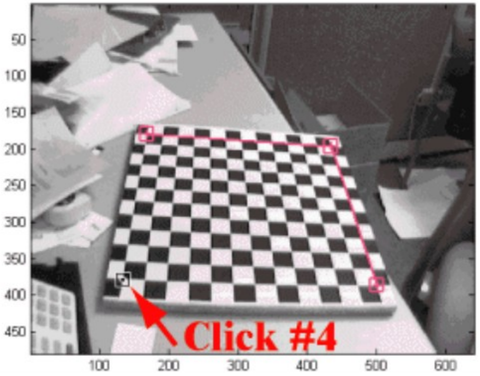
Courtesy: Silvio Savarese

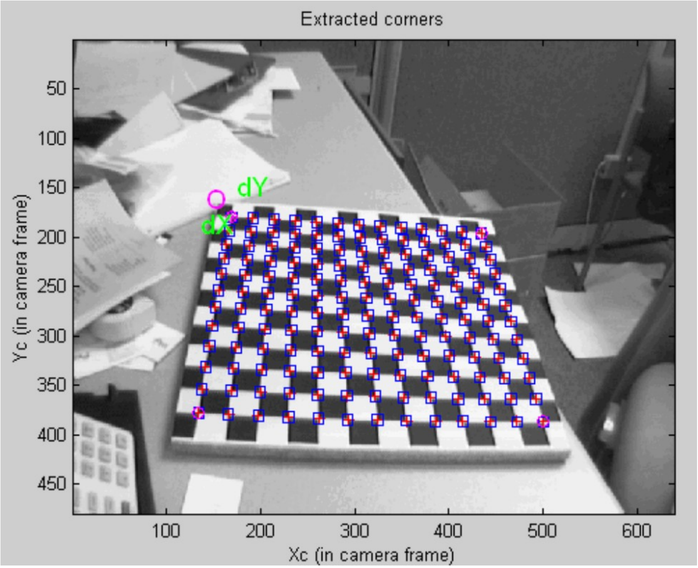
Example of calibration data











Courtesy: Silvio Savarese

- 12 elementi ma vi è dipendenza, di fatto sono 11 indipendenti

$$p = K\left[R\,T\right]P = MP$$

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Calibration

$$u_i=\frac{m_{00}X_i+m_{01}Y_i+m_{02}Z_i+m_{03}}{m_{20}X_i+m_{21}Y_i+m_{22}Z_i+m_{23}} \qquad v_i=\frac{m_{10}X_i+m_{11}Y_i+m_{12}Z_i+m_{13}}{m_{20}X_i+m_{21}Y_i+m_{22}Z_i+m_{23}}$$

$$m_{20}X_i+m_{21}Y_i+m_{22}Z_i+m_{23}-m_{00}u_iX_i-m_{01}u_iY_i-m_{02}u_iZ_i-m_{03}u_i=0$$

$$m_{10}X_i+m_{11}Y_i+m_{12}Z_i+m_{13}-m_{20}v_iX_i-m_{21}v_iY_i-m_{22}v_iZ_i-m_{23}v_i=0$$

2 equations for each correspondance

Calibration

UNIVERSITÀ DI PARMA

Homogeneous System

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\
0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n
\end{bmatrix}
\begin{bmatrix}
m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23}
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

Calibration

- Step back and write in a simpler form

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix} P_i$$

Calibration

UNIVERSITÀ DI PARMA

- Step back and write in a simpler form

$$u_i = \frac{m_{00} X_i + m_{01} Y_i + m_{02} Z_i + m_{03}}{m_{20} X_i + m_{21} Y_i + m_{22} Z_i + m_{23}} \qquad v_i = \frac{m_{10} X_i + m_{11} Y_i + m_{12} Z_i + m_{13}}{m_{20} X_i + m_{21} Y_i + m_{22} Z_i + m_{23}}$$

$$u_i = \frac{m_0 P_i}{m_2 P_i} \qquad v_i = \frac{m_1 P_i}{m_2 P_i}$$

$$-m_0 P_i + u_i (m_2 P_i) = 0 \qquad -m_1 P_i + v_i (m_2 P_i) = 0$$

Calibration

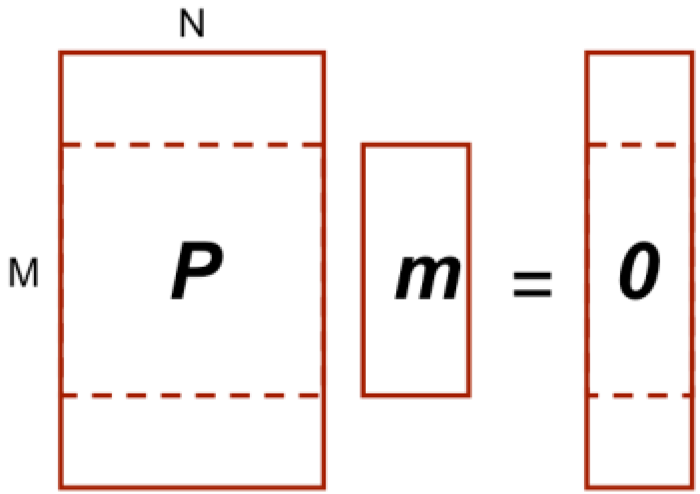
$$-m_0 P_i + u_i (m_2 P_i) = 0 \qquad -m_1 P_i + v_i (m_2 P_i) = 0$$

$$\begin{bmatrix} -P_0^T & 0^T & u_0 P_0^T \\ 0^T & -P_0^T & v_0 P_0^T \\ \dots & \dots & \dots \\ -P_n^T & 0^T & u_n P_n^T \\ 0^T & -P_n^T & v_n P_n^T \end{bmatrix}_{2n \times 12} \begin{bmatrix} m_0^T \\ m_1^T \\ m_2^T \end{bmatrix}_{12 \times 1} = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} \Rightarrow Pm = 0$$

Homogeneous System → 2n equations

Unknowns → m_i

Calibration



- M>N as we suggested
  - Rectangular system of equations
  - Overdetermined
    - "potentially" no solution other trivial one m=0
- The idea is to find the best fitting solution
- Minimize $|\mathrm{P}\,m|^2$
  - Assuming m != 0 $\longrightarrow$ $|m|^2 = 1$

Courtesy: Silvio Savarese

Inverting a non-square matrix

- A linear equation

$$Ax = y$$

- Can be solved finding a matrix B (ideally $B=A^{-1}$, namely $AB=I$)

$$x = By$$

- When A is taller than it is wide $\longrightarrow$ possibly no solutions
- When A is wider than it is tall $\longrightarrow$ multiple solutions

Singular Value Decomposition (SVD)

- Partially generalize matrix inversion for non square matrices
  - Do you remember something?
- Each matrix can be factorized into **singular vectors** and **singular values**
- More precisely each matrix A can be seen as product of three matrices:

$$A = UDV^T$$

Singular Value Decomposition (SVD)

- Supposing A as $m \times n$ matrix, U is a $m \times m$ matrix, D is a $m \times n$ one and V is a $n \times n$ one
- U and D orthogonal matrices
  - Namely $UU^T = U^T U = I$ and $VV^T = V^T V = I$
  - Columns are the left (U) or right (V) singular vectors
- D is a diagonal matrix
  - Elements along the diagonal are the singular values

Singular Value Decomposition (SVD)

UNIVERSITÀ DI PARMA

- A linear equation

$$Ax = y$$

- Can be solved finding a matrix B (ideally $B=A^{-1}$)

$$x = By$$

- When A is taller than it is wide $\longrightarrow$ possibly no solutions
- When A is wider than it is tall $\longrightarrow$ multiple solutions

Moore-Penrose pseudoinverse

- The pseudoinverse of a matrix A or A^+ is defined as

$$A^{+} = \lim_{\alpha \to 0} \left(A^{T} A + \alpha I \right)^{-1} A^{T}$$

- Practically A^+ can be approximated using SVD

$$A^{+} = V D^{+} U^{T}$$

- When A is taller than it is wide this allows to compute the best approximation

Calibration

- How we can solve such system?

$$Pm=0$$

- Via SVD decomposition!
  - This leads to the optimal solution assuming $|m|^2=1$

Calibration

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

SVD decomposition of P

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

- The last V column gives us *m*

$$\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{pmatrix} \longrightarrow M$$

Courtesy: Silvio Savarese

Parameters Extraction

Courtesy: Silvio Savarese

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

- We can recover parameters but with a scale factor ρ
- Actually not an unknown, we can impose |*m*|=1
  - or equivalent Froebius norm ||M||=1
- Extrinsic and Intrinsic parameters are separately extracted

Parameters Extraction

UNIVERSITÀ DI PARMA

$$\frac{M}{\rho} = \left(\underbrace{\begin{matrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{matrix}}_{A} \quad \underbrace{\begin{matrix} \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ t_z \end{matrix}}_{\mathbf{b}} \right) = \frac{K}{\rho}\begin{bmatrix} R & T \end{bmatrix}$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \dfrac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

Box 1

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

Estimated values

Intrinsic

$$\rho = \frac{\pm 1}{|\mathbf{a}_3|} \qquad u_o = \rho^2(\mathbf{a}_1 \cdot \mathbf{a}_3) \qquad v_o = \rho^2(\mathbf{a}_2 \cdot \mathbf{a}_3)$$

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

Courtesy: Silvio Savarese

[FP] Sec. 1.3.1

Parameters Extraction

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

A **b**

$$\mathrm{A} = \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \quad \mathrm{b} = \begin{bmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \mathrm{b}_3 \end{bmatrix}$$

Estimated values

Intrinsic

$$\alpha = \rho^2 \left|\mathbf{a}_1 \times \mathbf{a}_3\right| \sin\theta$$

$$\beta = \rho^2 \left|\mathbf{a}_2 \times \mathbf{a}_3\right| \sin\theta$$

Courtesy: Silvio Savarese

[FP] Sec. 1.3.1

Parameters Extraction

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

A **b**

$$\mathrm{A} = \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \quad \mathrm{b} = \begin{bmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \mathrm{b}_3 \end{bmatrix}$$

Estimated values

Extrinsic

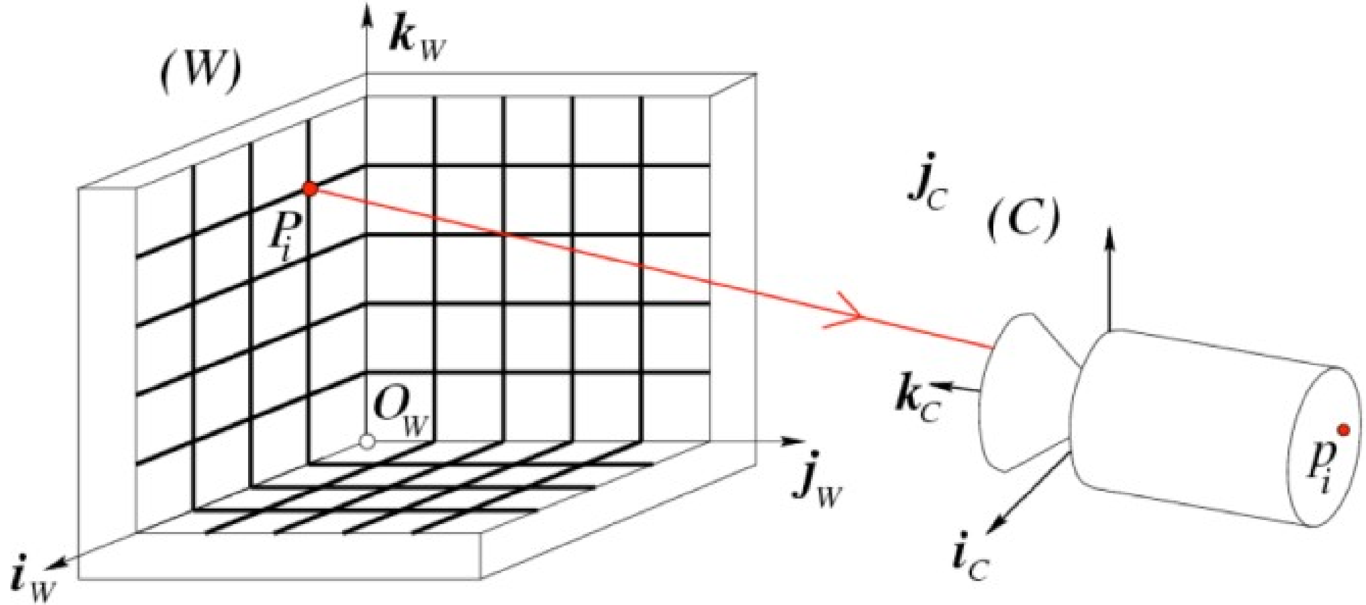
$$\mathbf{r}_1 = \frac{(\mathbf{a}_2 \times \mathbf{a}_3)}{|\mathbf{a}_2 \times \mathbf{a}_3|} \qquad \mathbf{r}_3 = \frac{\pm \mathbf{a}_3}{|\mathbf{a}_3|}$$

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

Courtesy: Silvio Savarese

[FP] Sec. 1.3.1

Degenerate Case



- Not all P_ws lead to a result!
  - P_w points must not lie on the same plane
  - P_w points must not lie on the intersection curve of 2 quadric surfaces

Courtesy: Silvio Savarese

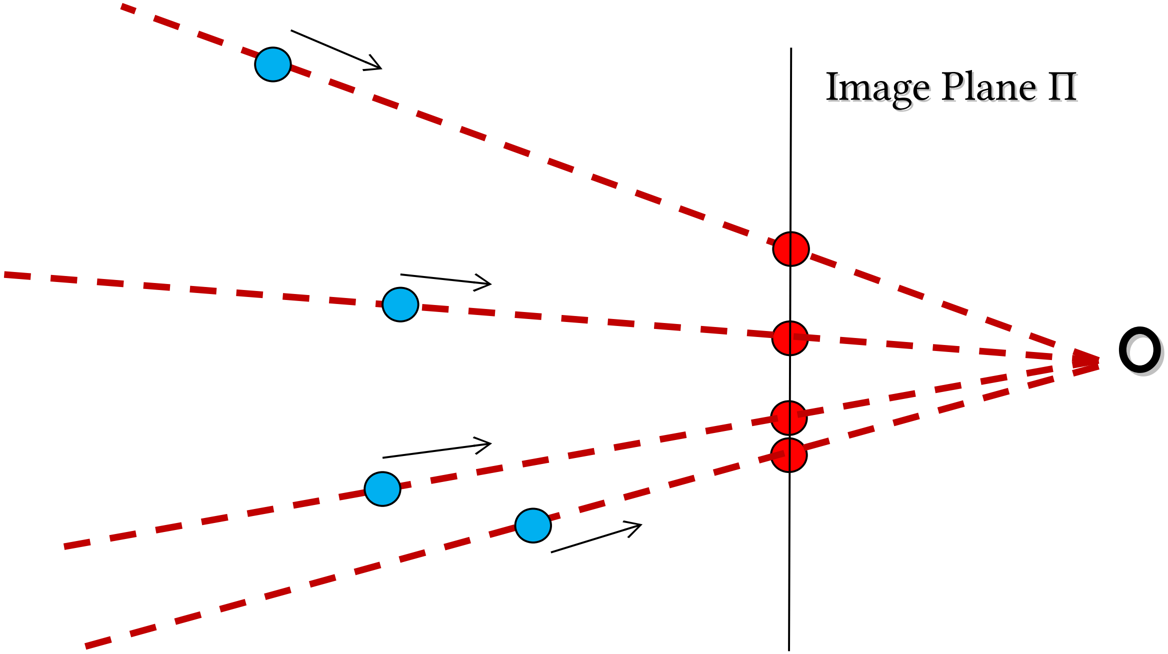
UNIVERSITÀ DI PARMA

Calibration
Zhang approach (1998)

Calibration

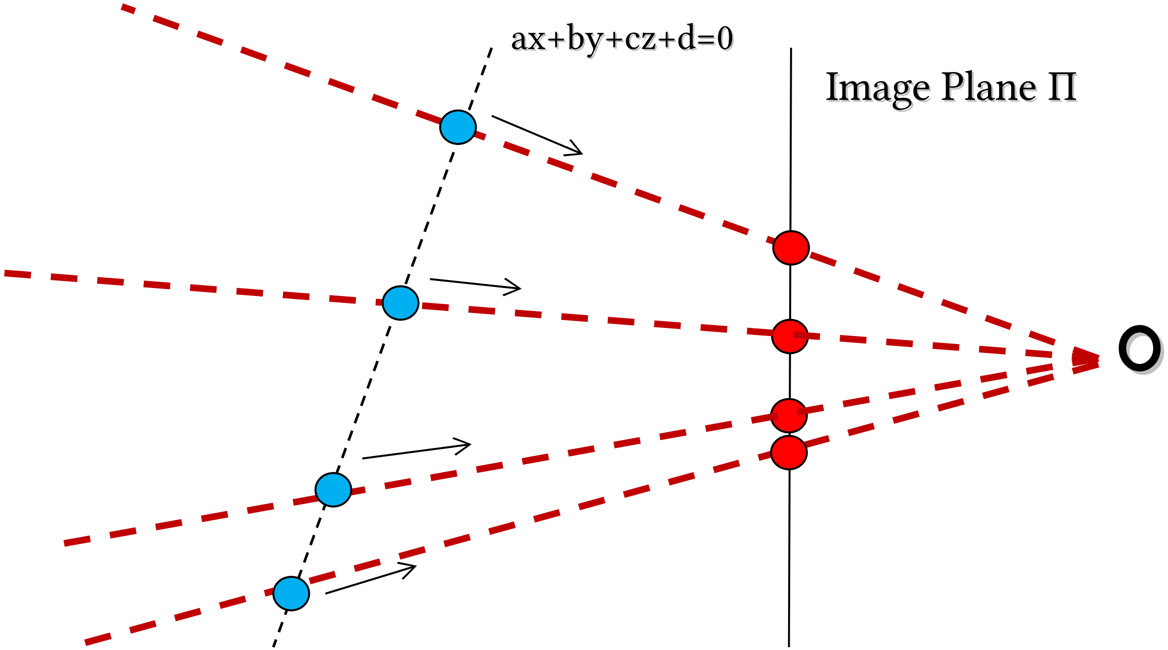
- Usually calibration grids are “flat”
- This does not fit well with Tsai’s requirements
  - No coplanar points
- Zhang’s approach, conversely, exploits this!

Ill posed problem



- Projective space → we lost one dimension
- 3D [x,y,z] are projected in [x/z,y/z,1]

Add a constraint



- Just image that the (red) image points are projection of points that lie on the same plane
- In such a case for each image point there is only one world point (cyan)

Add a constraint

- Given a plane aX + bY + cZ + d = 0

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underset{\boxed{3\times4}}{M} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ w \\ 0 \end{bmatrix} = \overset{\boxed{4\times4}}{\begin{bmatrix} M \\ a\,b\,c\,d \end{bmatrix}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} M \\ a\,b\,c\,d \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \\ 0 \end{bmatrix} \xrightarrow[\text{euclidean}]{} \begin{bmatrix} X/W \\ Y/W \\ Z/W \end{bmatrix}$$

- We can obtain euclidean world coordinates of pixel (u,v)

Specific case, plane Z=0

- Consider a very specific plane $\longrightarrow$ Z=0

$$\begin{bmatrix} u \\ v \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

- H is 3×3
- H is an **Homography**
- This actually works for every possible plane!

UNIVERSITÀ DI PARMA

- For each point on the planar surface of my grid I then obtain
- In the same image…

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = H \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \quad \text{with} \quad i=1,2,3,\ldots,n$$

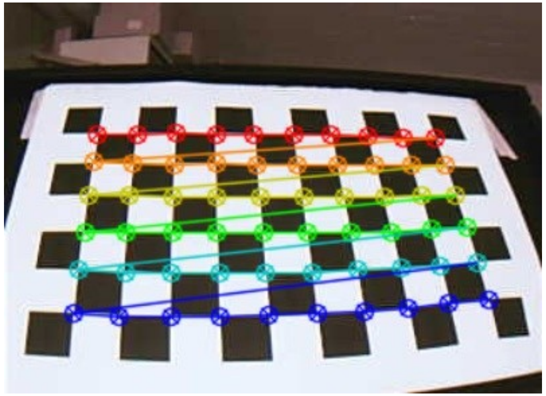
Calibration

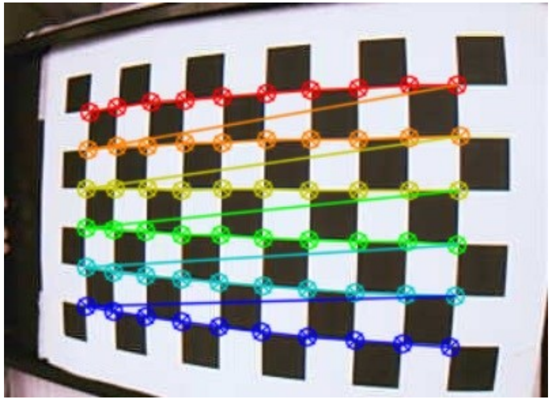
UNIVERSITÀ DI PARMA

- For solving this issue we can use the same approach used for "Tsai"
- We have a 3×3 homograpy instead of a 3×4 projection matrix
- We need to solve $QH = 0$ where Q is $2n \times 9$ and has rank 8
- We need at least 4 points
  - When $n > 4 \longrightarrow$ SVD

Calibration

- With a "single" H we can not anyway fully compute M
- 3 Hs are needed, at least, to compute M
  - The larger the number of views, the better the result
  - Usually 20-50 grid views

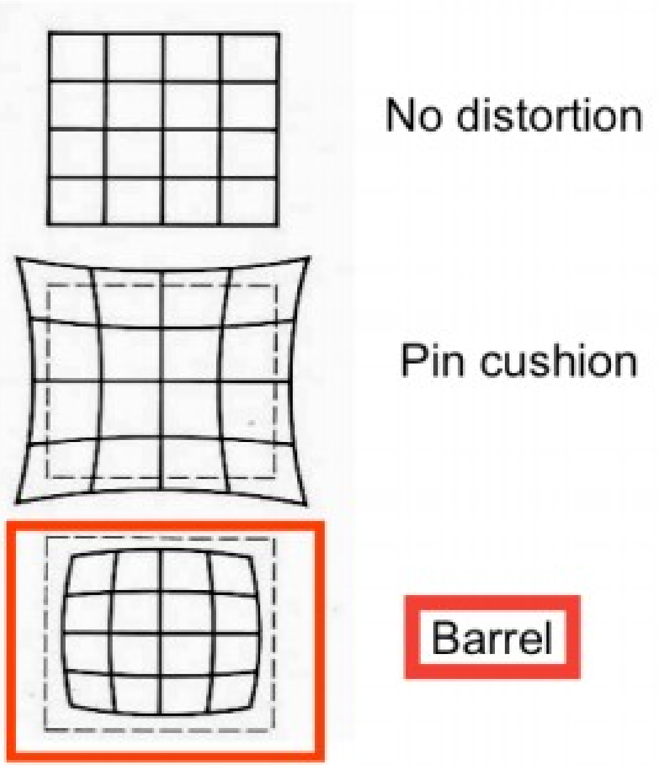




UNIVERSITÀ DI PARMA

Lens Distortion

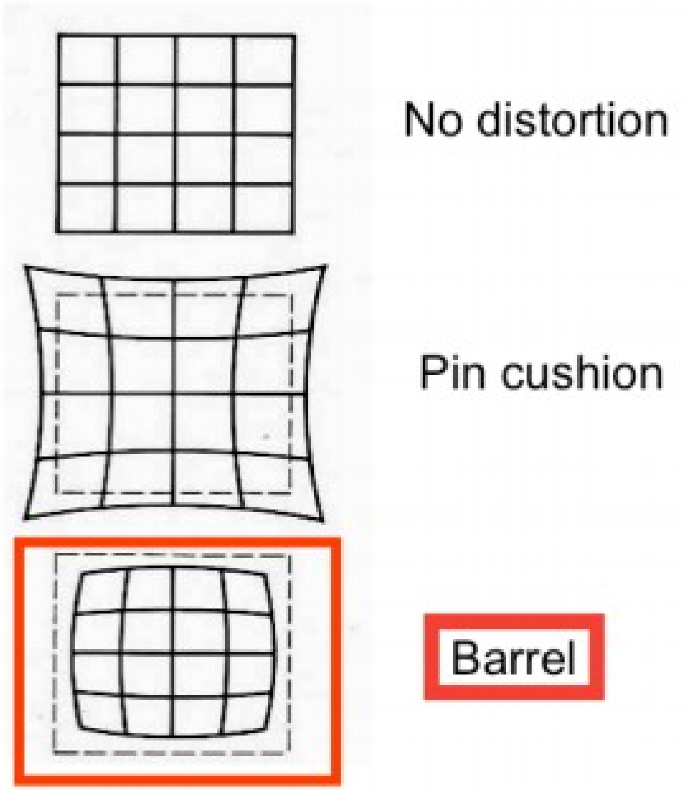
Lens Distortion



Courtesy: Silvio Savarese

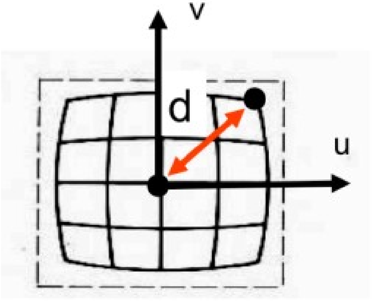
Radial Lens Distortion

- We assumed a “thin” lens
  - This basically means that lines are lines
  - Unfortunately this is not true!
- Radial Distortion
  - Image magnification/contraction depends on distance from optical axis
  - Deviations are more evident on lateral areas
  - Cheap optics



Courtesy: Silvio Savarese

Radial Lens Distortion



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

$$\lambda = 1 \pm \underbrace{\sum_{p=1}^{3} \overbrace{\boxed{\kappa_p}}^{\text{Distortion coefficient}} d^{2p}}_{\text{Polynomial function}} \qquad \text{[Eq. 5]}$$

$$\underbrace{d^2 = a\,u^2 + b\,v^2 + c\,u\,v}_{\text{To model radial behavior}} \qquad \text{[Eq. 6]}$$

- λ is the distortion factor
  - Actually it is a $\lambda_{(u,v)}$ since it depends on the distance from optical axis

Courtesy: Silvio Savarese

Radial Lens Distortion

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

- λ depends on u,v and this leads to a non linearity in the P → p mapping

Courtesy: Silvio Savarese

Radial Lens Distortion

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

- Tsai 87, initially estimate m_1 and m_2 by SVD and then $m_3=f(m_1, m_2, \lambda)$

Courtesy: Silvio Savarese

UNIVERSITÀ DI PARMA

OpenCV Calibration

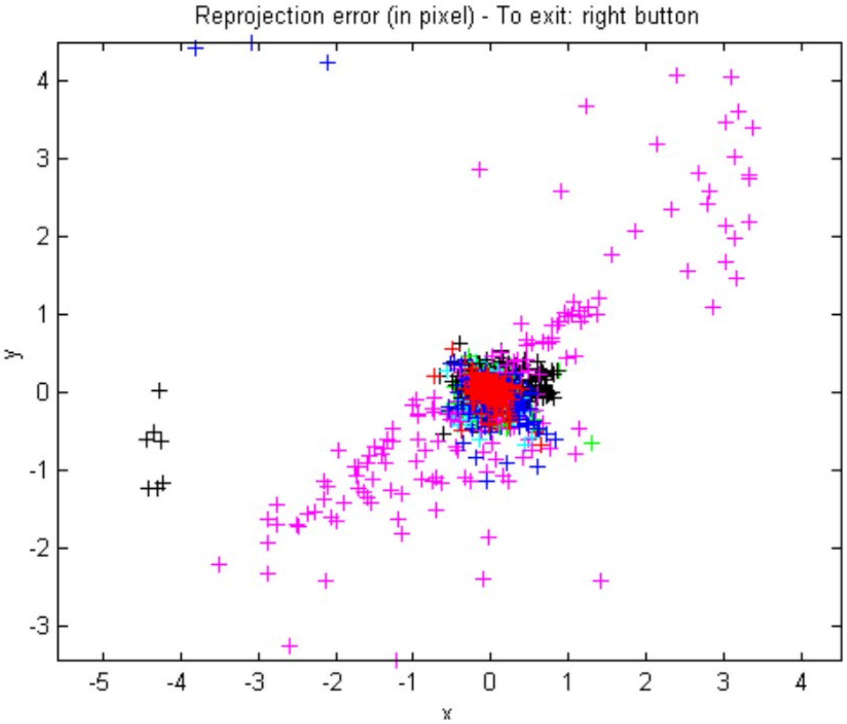
Multi plane calibration

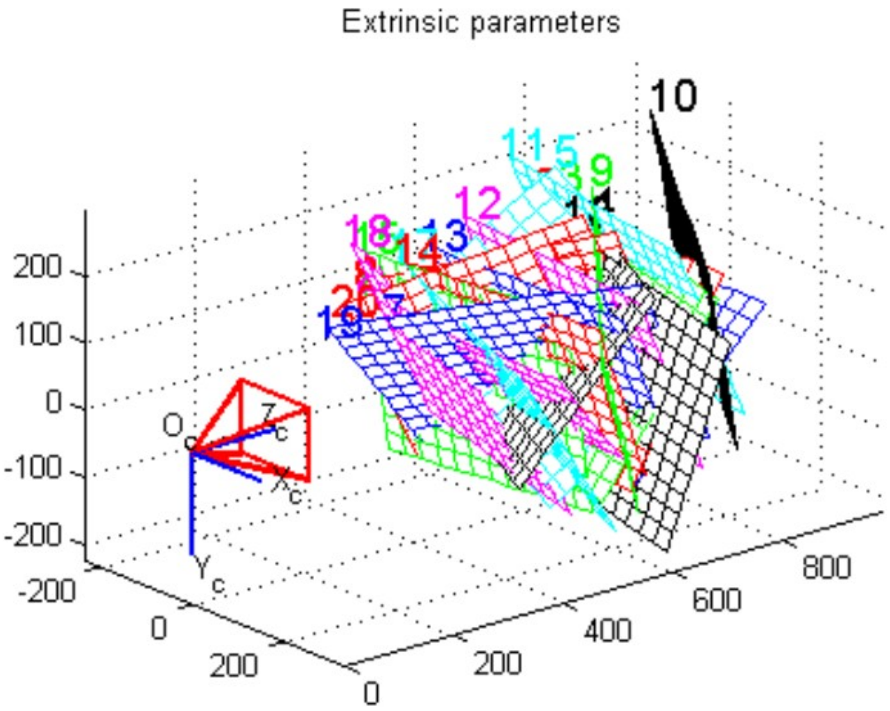
UNIVERSITÀ DI PARMA

- Requires a plane only!
- No known positions/orientations
- Strobl et al. approach (2011)

- Free code:
  - https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Example of calibration data





Courtesy: Silvio Savarese

UNIVERSITÀ DI PARMA

Camera Models (2)

Question time!