



UNIVERSITÀ DI PARMA

Camera Models (2)

Summary



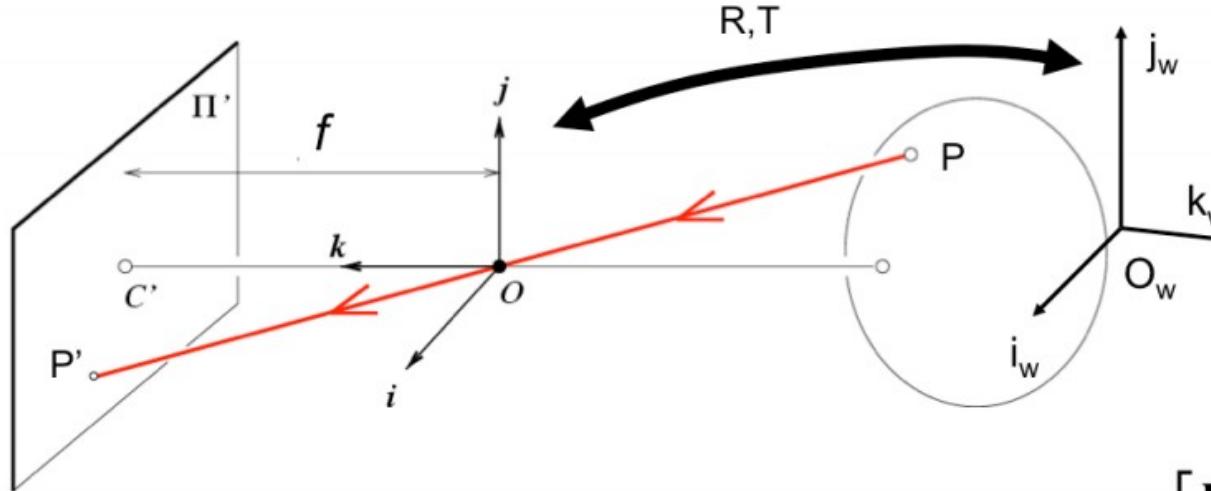
- Pin Hole Camera recap
- Calibration



UNIVERSITÀ DI PARMA

Pin Hole Camera recap

Perspective Transformation

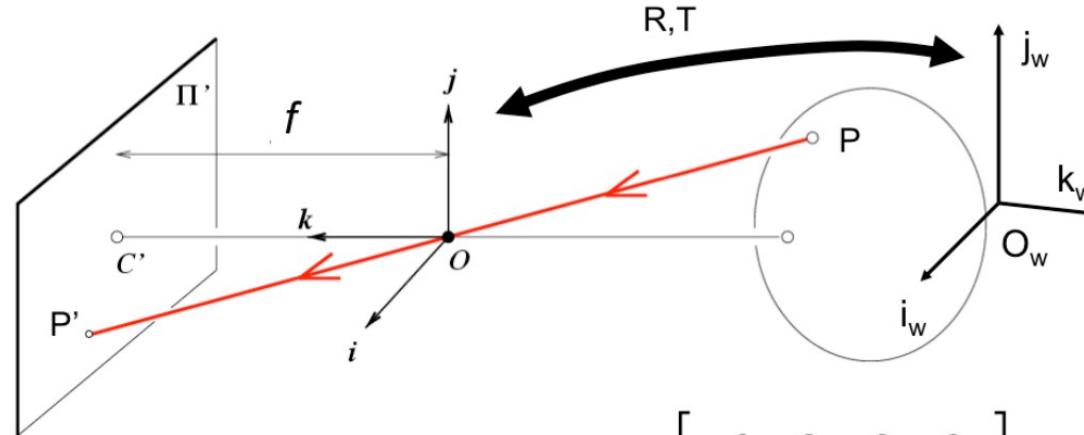


$$\begin{aligned}
 P'_{3 \times 1} &= M P_w = K_{3 \times 3} \begin{bmatrix} R & T \end{bmatrix}_{3 \times 4} P_w_{4 \times 1} & M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_w = \begin{bmatrix} \mathbf{m}_1 P_w \\ \mathbf{m}_2 P_w \\ \mathbf{m}_3 P_w \end{bmatrix} & \mathbf{E} \rightarrow \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w} \right) \quad [\text{Eq.12}]
 \end{aligned}$$

Perspective Transformation (ideal case)



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



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

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

$$P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Perspective Transformation

- 11 degrees of freedom

$$P' = M P_w = \boxed{K} \boxed{\begin{matrix} R & T \end{matrix}} P_w$$

Internal parameters External parameters

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

$$K = \begin{bmatrix} \alpha & -\alpha \cot \theta & u_o \\ 0 & \frac{\beta}{\sin \theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$



UNIVERSITÀ DI PARMA

Calibration

Calibration

$$P' = M P_w = \begin{bmatrix} K & R \\ T & I \end{bmatrix} P_w$$

Internal parameters External parameters

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

Change notation:
 $P = P_w$
 $p = P'$

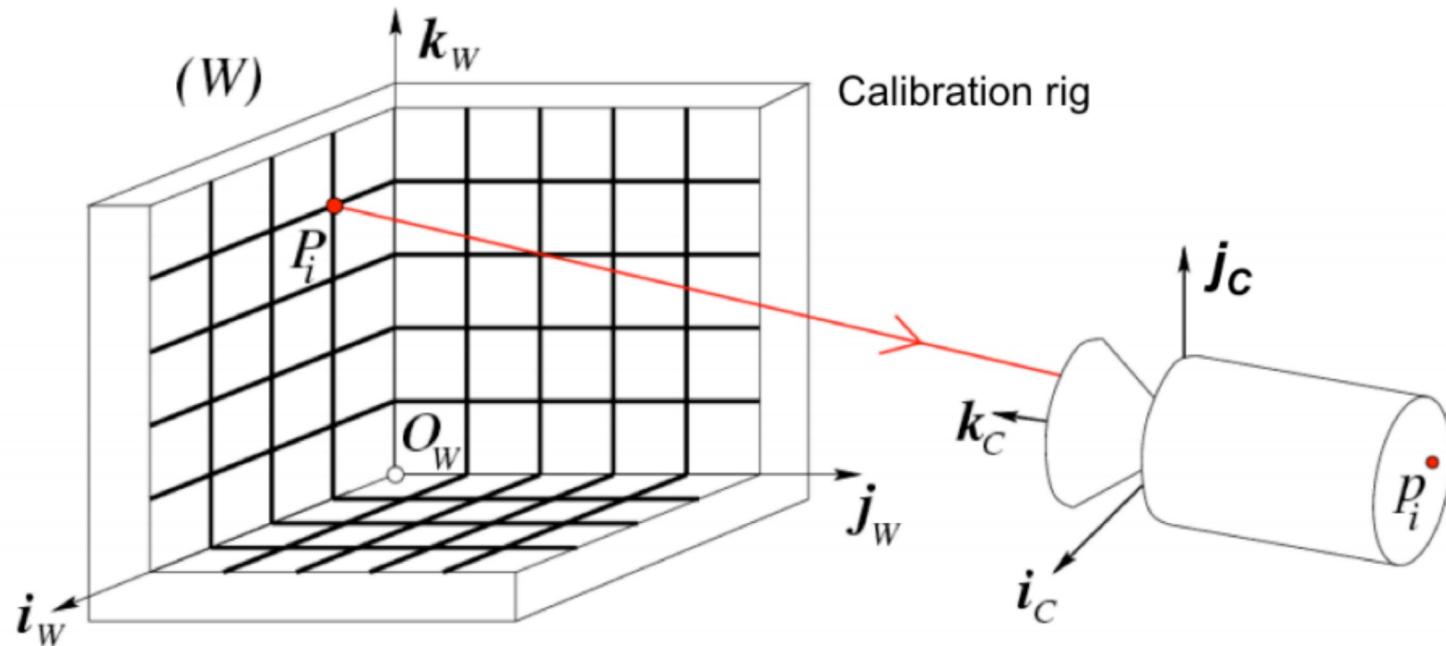
- Small notation change

$$-P_W \rightarrow P$$

$$-P' \rightarrow p$$

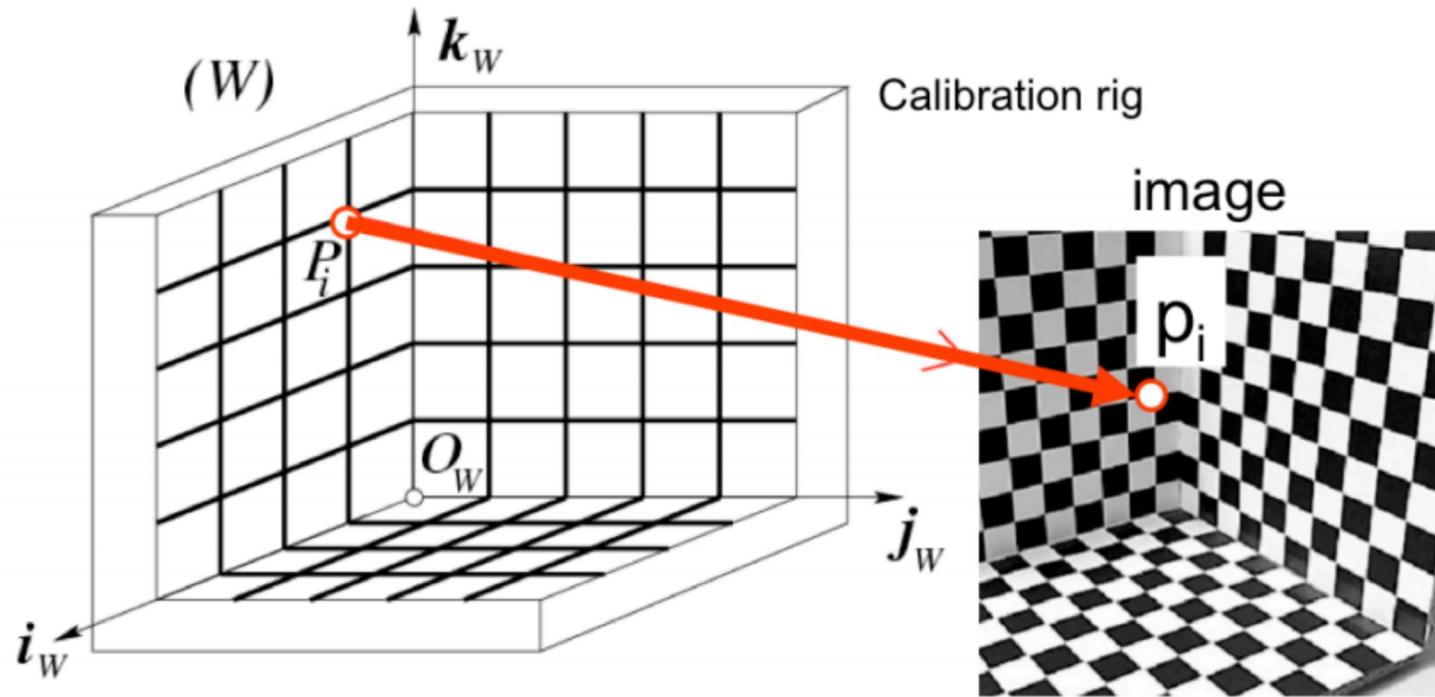
$$p = K[R\ T]\ P = MP$$

Calibration



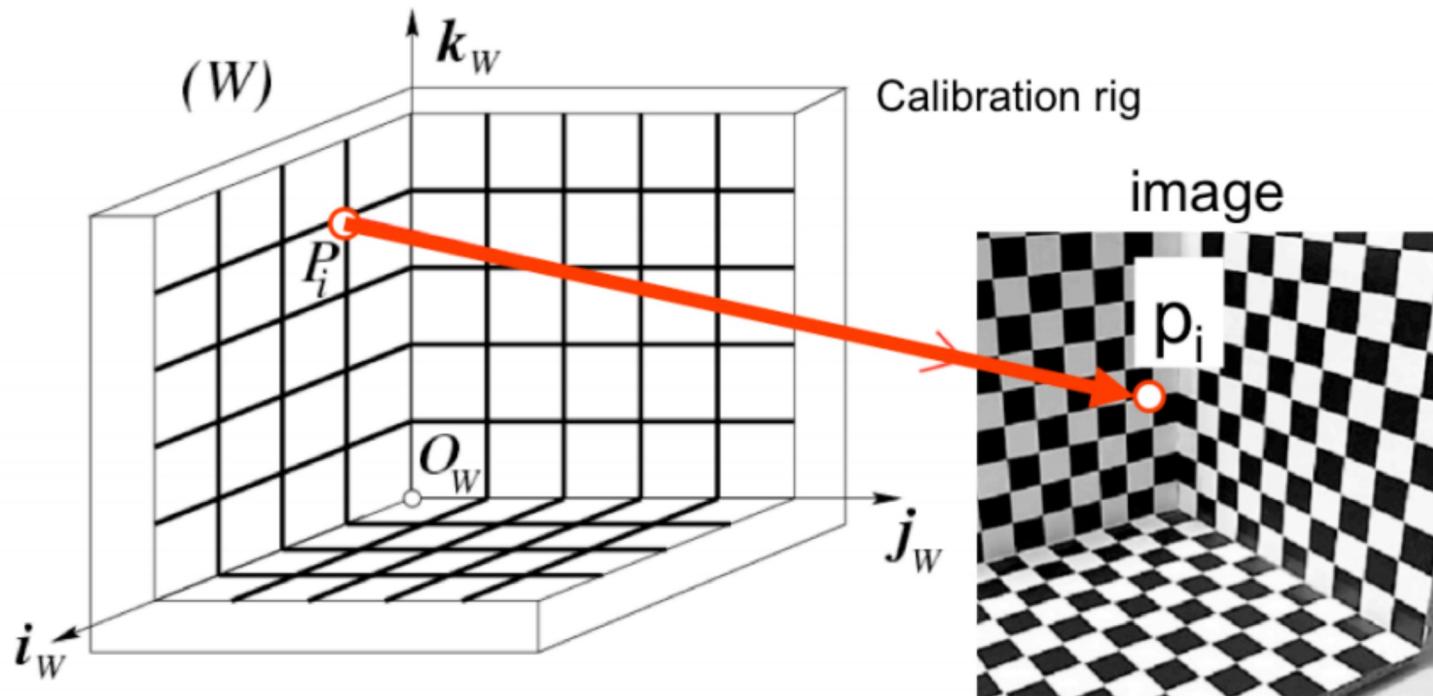
- We need: n points (P_1, P_2, \dots, P_n) in known positions $[O_w, i_w, j_w, k_w]$ in the world reference system

Calibration



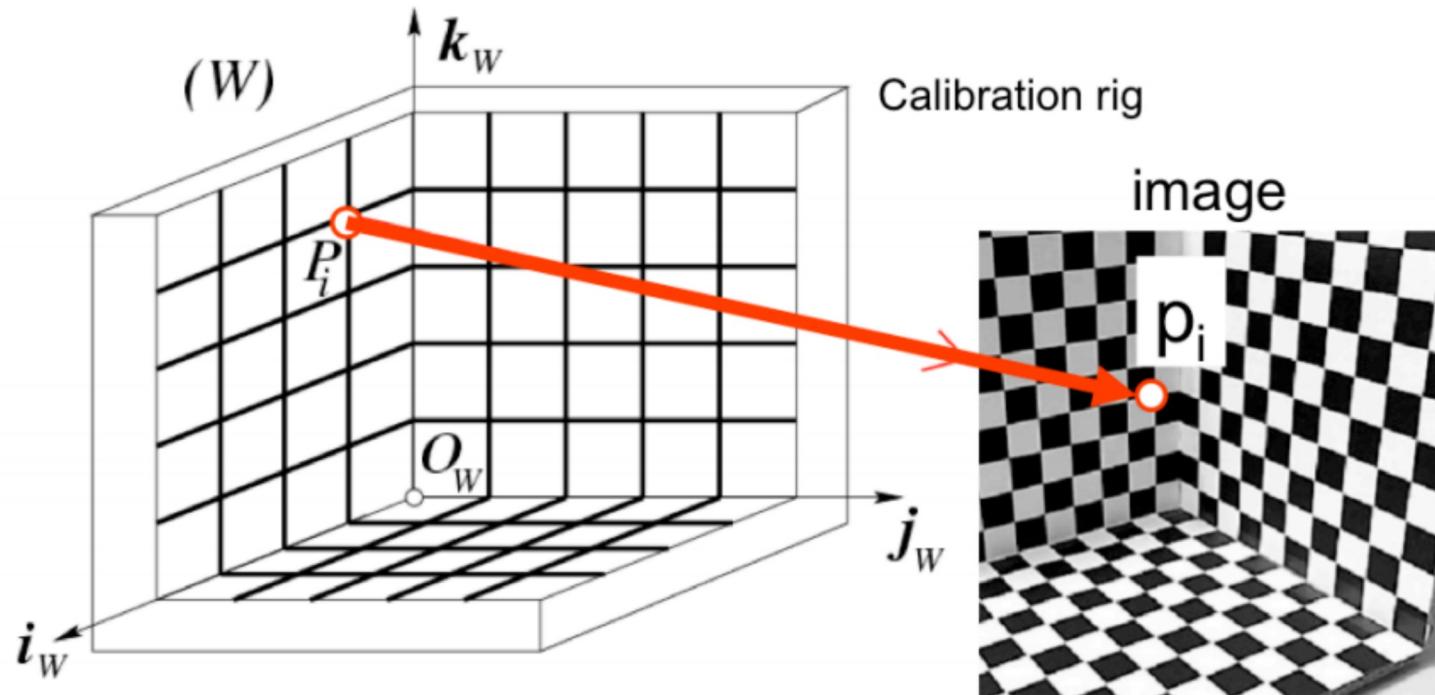
- We also need: camera coordinates for the n points projections (p_1, p_2, \dots, p_n) as (i_c, j_c)

Calibration



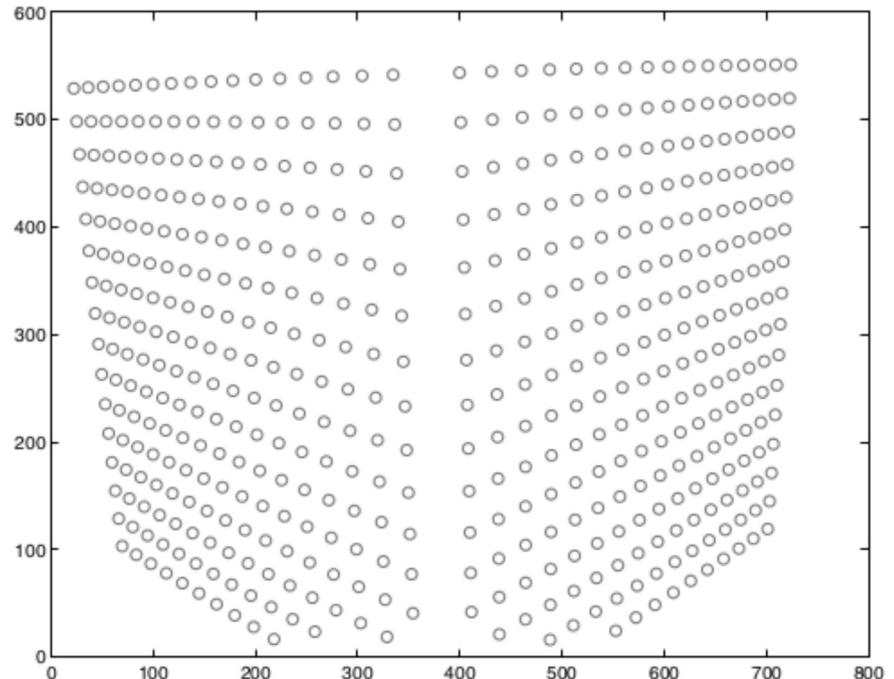
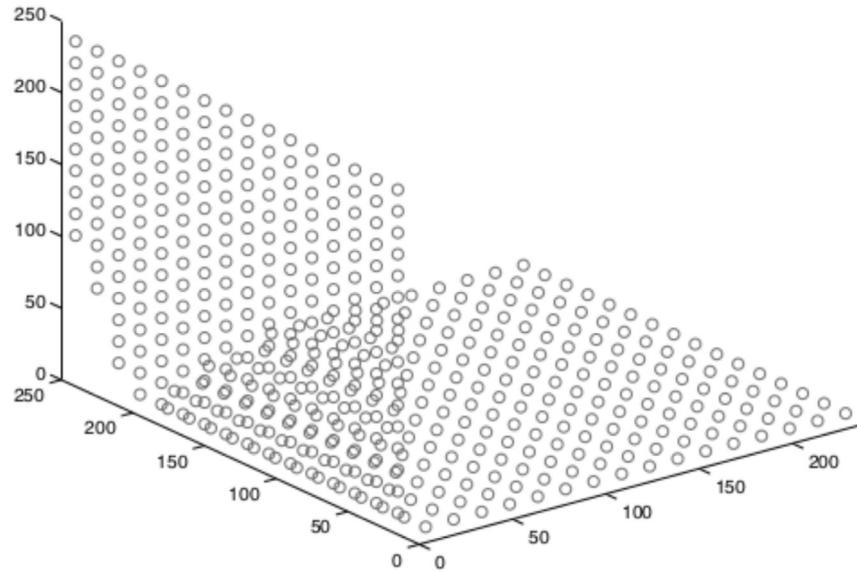
- 11 degrees of freedom → how many matches we need?
- Each correspondence → 2 equations (world coordinates → row/column)
- We need, at least, 6 correspondences

Calibration



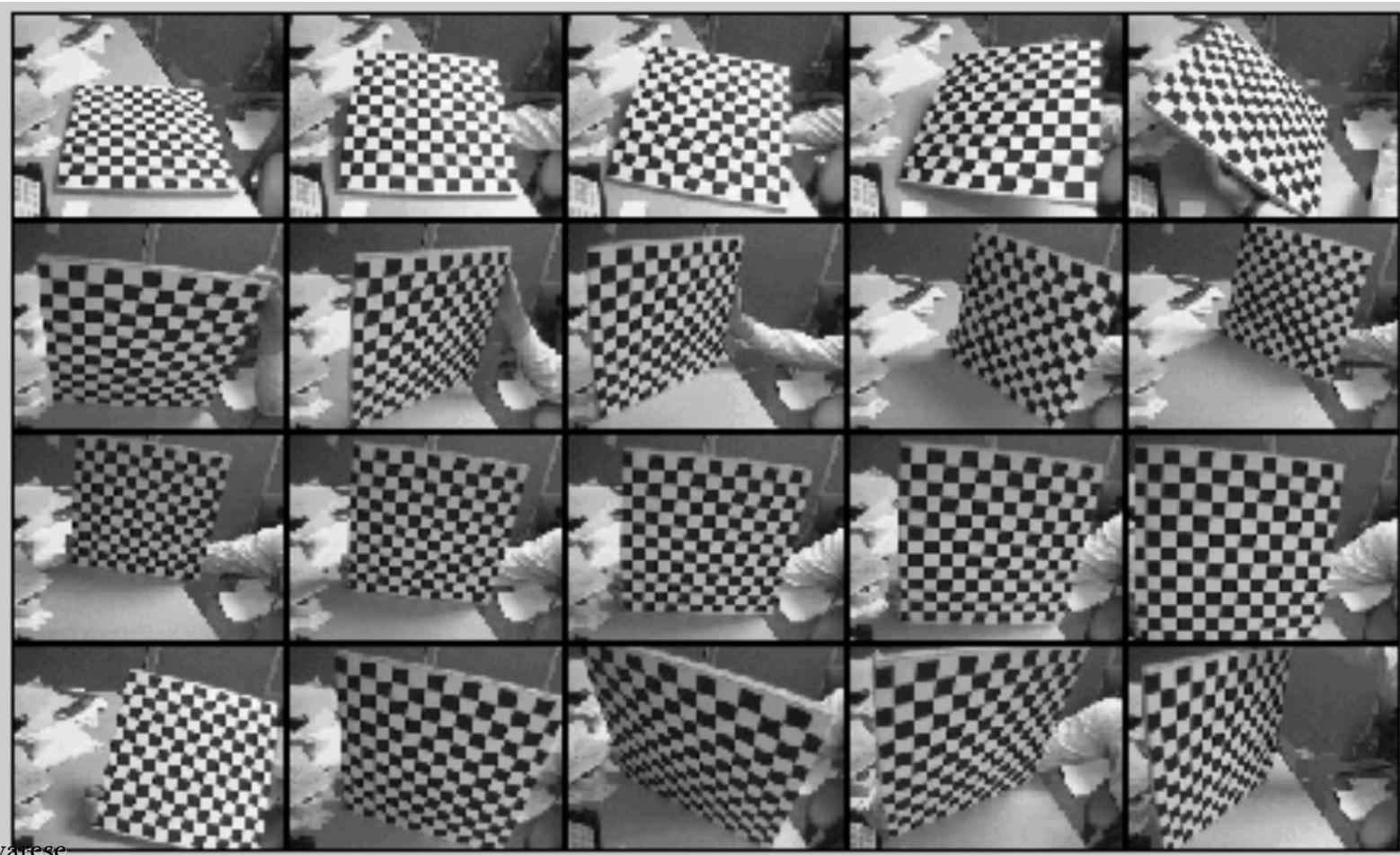
- Practically we need more than 6 correspondences!
- Consider error in measurements + digitalization error!

Example of calibration data

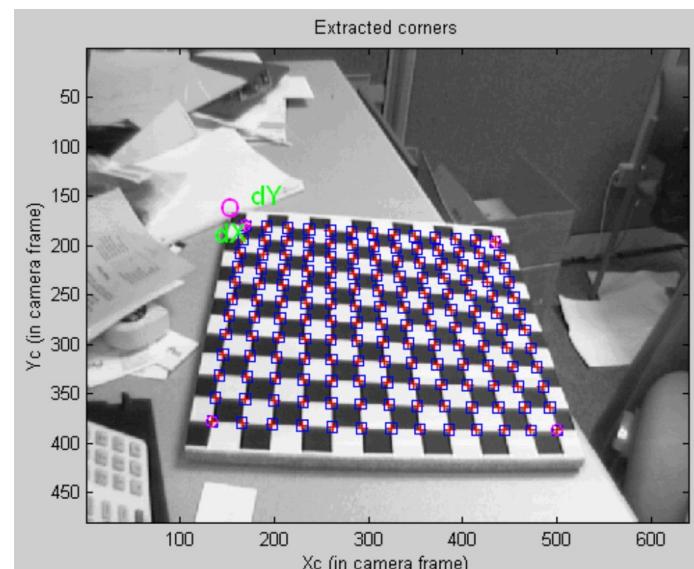
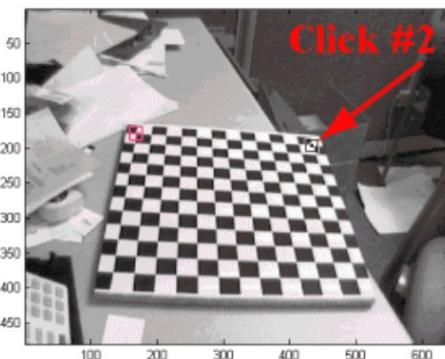
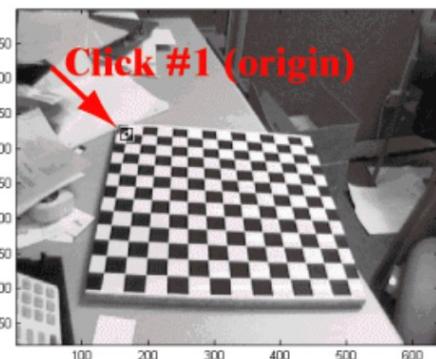


- Example of 491 3D points
 - From: Janne Heikkilä; data copyright 2000, University of Oulu.

Example of calibration data



Example of calibration data



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

$$p = K[R\ T]\ P = MP$$

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

Calibration

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

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

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

2 equations for each correspondance

Calibration



Homogeneous System

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

- Step back and write in a simpler form

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

- Step back and write in a simpler form

$$u_i = \frac{m_{00} X_i + m_{01} Y_i + m_{02} Z_i + m_{03}}{m_{20} X_i + m_{21} Y_i + m_{22} Z_i + m_{23}}$$

$$v_i = \frac{m_{10} X_i + m_{11} Y_i + m_{12} Z_i + m_{13}}{m_{20} X_i + m_{21} Y_i + m_{22} Z_i + m_{23}}$$

$$u_i = \frac{m_0 P_i}{m_2 P_i}$$

$$v_i = \frac{m_1 P_i}{m_2 P_i}$$

$$-m_0 P_i + u_i (m_2 P_i) = 0$$

$$-m_1 P_i + v_i (m_2 P_i) = 0$$

Calibration



$$-m_0 P_i + u_i (m_2 P_i) = 0$$

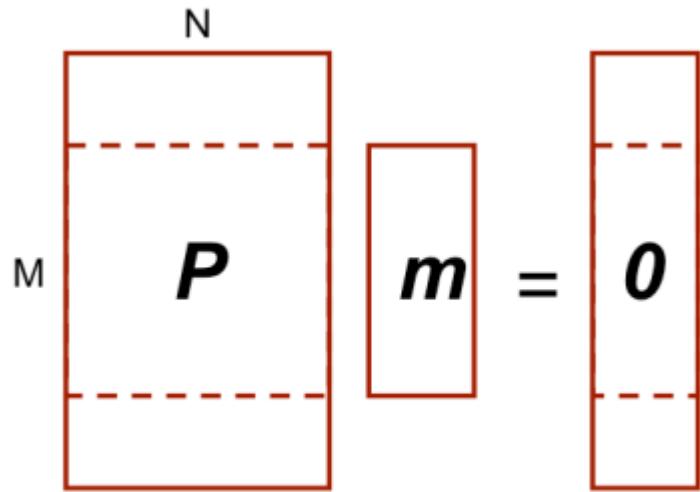
$$-m_1 P_i + v_i (m_2 P_i) = 0$$

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

Homogeneous System $\rightarrow 2n$ equations

Unknowns $\rightarrow m_i$

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 $|Pm|^2$
 - Assuming $m \neq 0 \rightarrow |m|^2 = 1$

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 \rightarrow possibly no solutions
- When A is wider than it is tall \rightarrow 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)

- 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 \rightarrow possibly no solutions
- When A is wider than it is tall \rightarrow multiple solutions

Moore-Penrose pseudoinverse

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

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

- 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

- How we can solve such system?

$$Pm=0$$

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

Calibration



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

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

$$\mathbf{m} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{pmatrix} \quad \xrightarrow{\hspace{1cm}} \quad M$$

Parameters Extraction

Courtesy: Silvio Savarese

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

- 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

$$\frac{M}{\rho} = \begin{pmatrix} \alpha \mathbf{r}_1^T - \alpha \cot \theta \mathbf{r}_2^T + u_0 \mathbf{r}_3^T & \alpha t_x - \alpha \cot \theta t_y + u_0 t_z \\ \frac{\beta}{\sin \theta} \mathbf{r}_2^T + v_0 \mathbf{r}_3^T & \frac{\beta}{\sin \theta} t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{pmatrix} = \frac{K}{\rho} \begin{bmatrix} R & T \end{bmatrix}$$
$$K = \begin{bmatrix} \alpha & -\alpha \cot \theta & u_0 \\ 0 & \frac{\beta}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Box 1

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

Estimated values

Intrinsic

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

Parameters Extraction

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

A **b**

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

Estimated values

Intrinsic

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

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

Parameters Extraction

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

A **b**

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

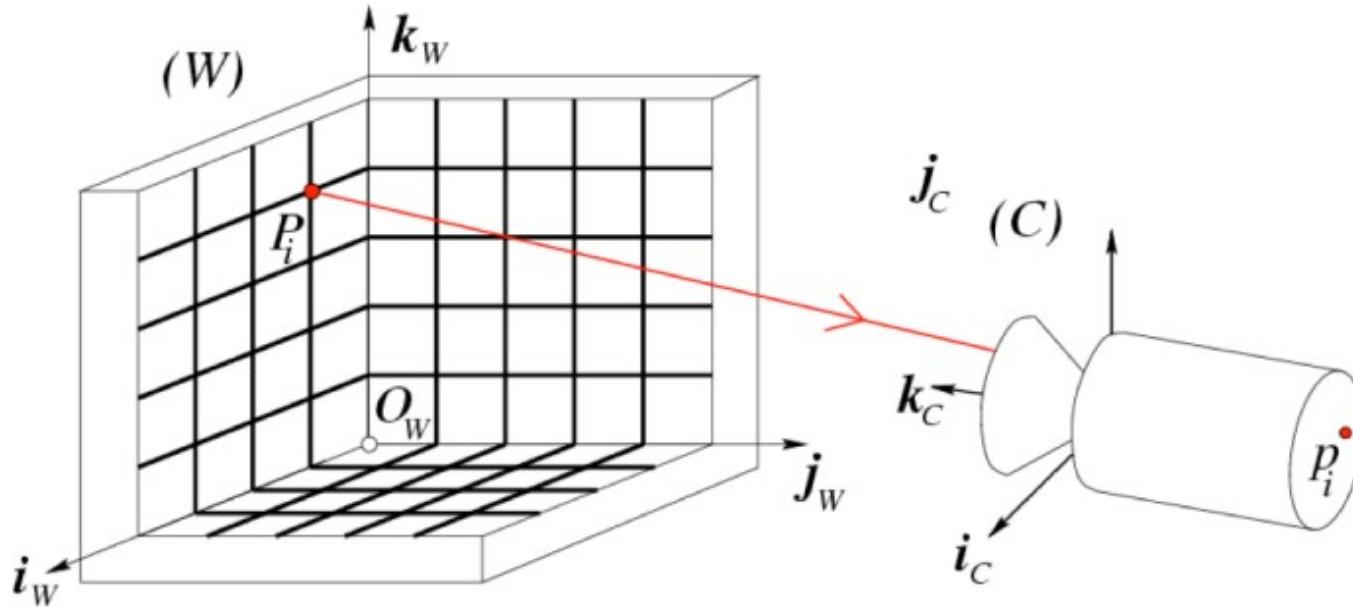
Estimated values

Extrinsic

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

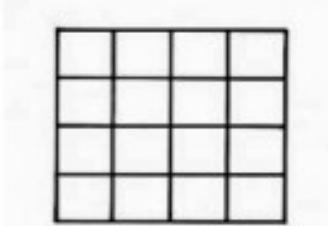
$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1 \quad \mathbf{T} = \rho \mathbf{K}^{-1} \mathbf{b}$$

Degenerate Case

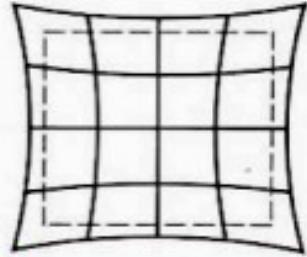


- Not all P_w s lead to a result!
 - P_w points must not lie on the same plane
 - P_w points must not lie on the intersection curve of 2 quadric surfaces

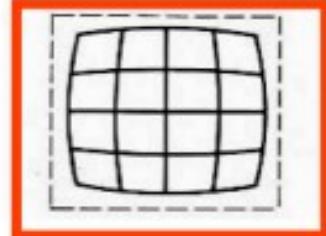
Lens Distortion



No distortion



Pin cushion

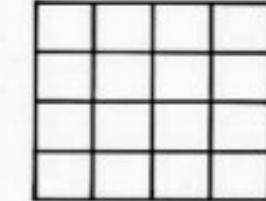


Barrel

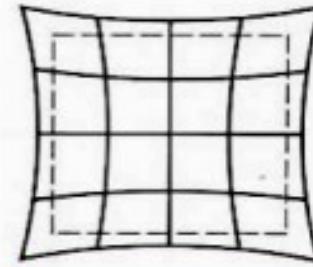


Radial Lens Distortion

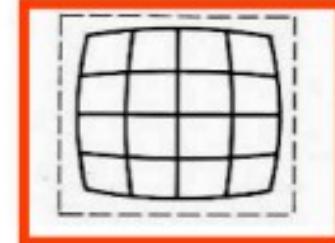
- We assumed a “thin” lens
 - This basically means that lines are lines
 - Unfortunately this is not true!
- Radial Distortion
 - Image magnification/contraction depends on distance from optical axis
 - Deviations are more evident on lateral areas
 - Cheap optics



No distortion

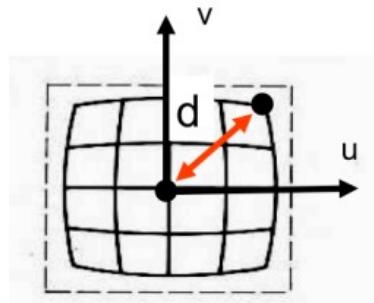


Pin cushion



Barrel

Radial Lens Distortion



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

$$\lambda = 1 \pm \sum_{p=1}^3 \kappa_p d^{2p}$$

[Eq. 5] [Eq. 6]

Distortion coefficient To model radial behavior

Polynomial function

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

Radial Lens Distortion



$$\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i \quad Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Q

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

Radial Lens Distortion



$$\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i \quad Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

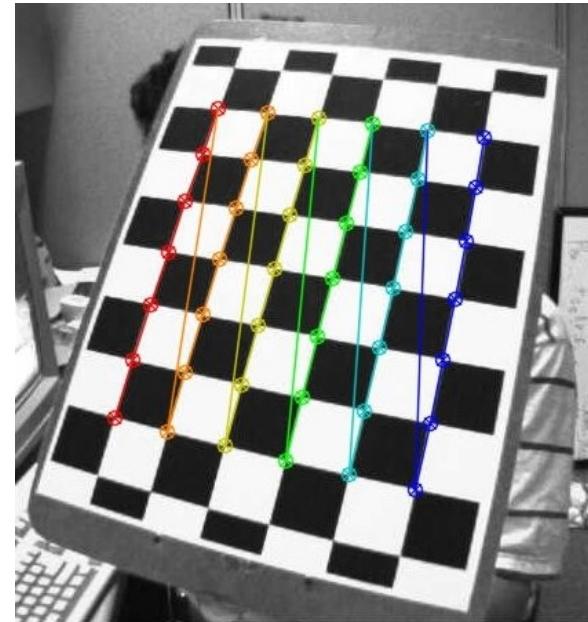
Q

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

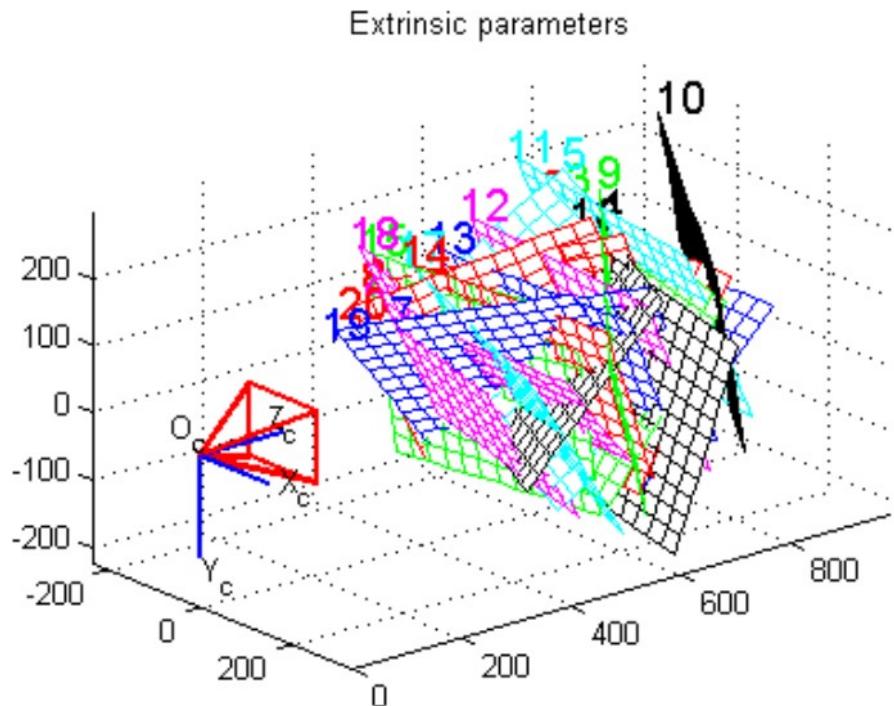
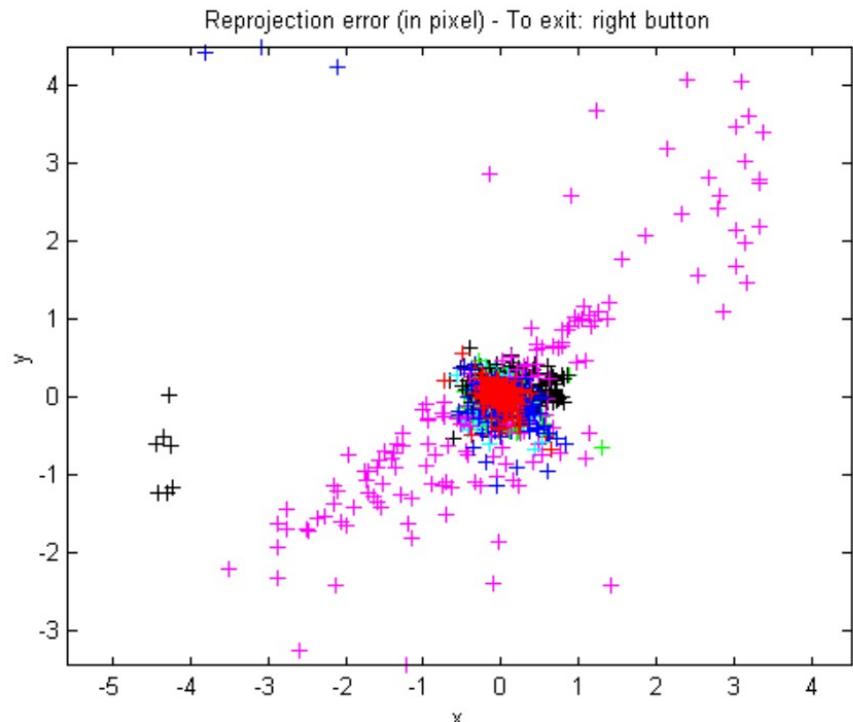
Multi plane calibration



- Requires a plane only!
- No known positions/orientations
- Free code:
 - https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html



Example of calibration data





UNIVERSITÀ DI PARMA

Camera Models (2)

Question time!

