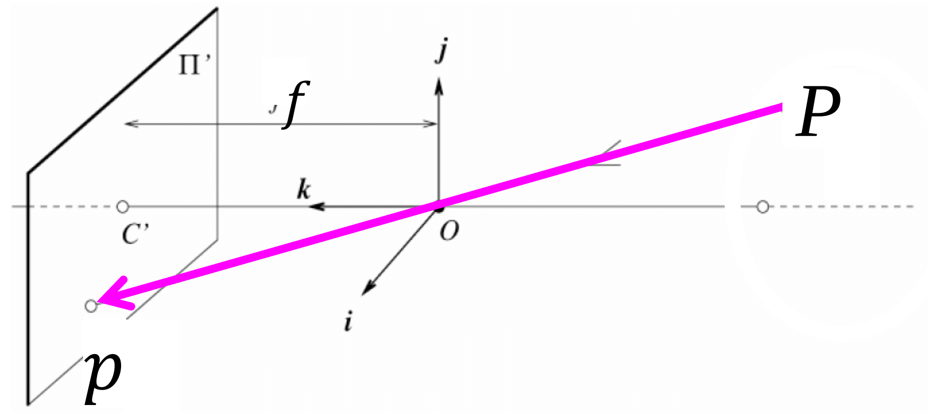
Camera calibration

Lihi Zelnik-Manor, Computer Vision

Goal of calibration

- Estimate the intrinsic and extrinsic parameters from one or multiple images

$$p = MP = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

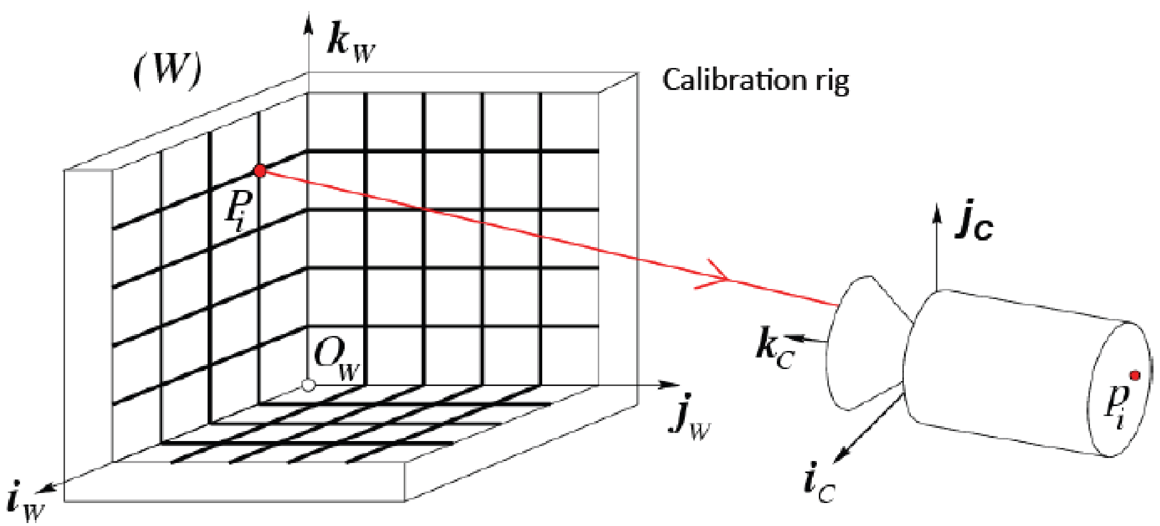


$$K = \begin{bmatrix} \alpha & s & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} - & r_1 & - \\ - & r_2 & - \\ - & r_3 & - \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

The calibration problem

- **Input**
  n points $P_1, \dots, P_n$ with known coordinates and known positions in the image $p_1, \dots, p_n$

- **Goal**
  Compute intrinsic and extrinsic parameters



The calibration problem

- **Input**
  n points $P_1, \dots, P_n$ with known coordinates and known positions in the image $p_1, \dots, p_n$

- **Goal**
  Compute intrinsic and extrinsic parameters

$$p = MP \quad \Rightarrow \quad \begin{bmatrix} xw \\ wy \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

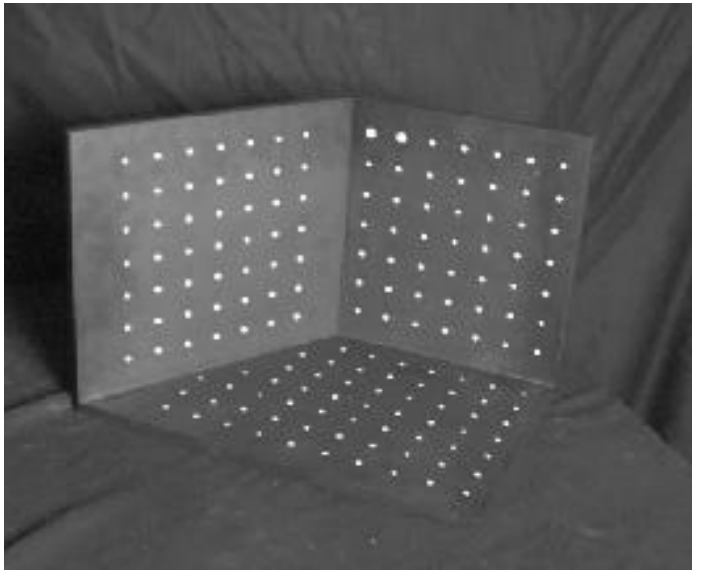
The calibration problem

- **Question**
  How many points $P_1, \dots, P_n$ we need?
- **Answer**
  11 unknowns ➔ 11 equations ➔ 6 points
- In practice we use more than 6

$$p = MP \quad \Rightarrow \quad \begin{bmatrix} xw \\ wy \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Method 1: Calibration using a reference object

- Place a known object in the scene
  - identify correspondence between image and scene
  - compute mapping from scene to image



Issues

- must know scene geometry very accurately
- must know 3D->2D correspondence

Equations from a single point

- A point in the world projects onto a point in the image

$$MP_i \rightarrow p_i$$

- The equations we get are

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \dfrac{m_1 P_i}{m_3 P_i} \\ \dfrac{m_2 P_i}{m_3 P_i} \end{bmatrix} \qquad M = \begin{bmatrix} - & m_1 & - \\ - & m_2 & - \\ - & m_3 & - \end{bmatrix}$$

Equations from a single point

- This can be re-written more conveniently

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \dfrac{m_1 P_i}{m_3 P_i} \\ \dfrac{m_2 P_i}{m_3 P_i} \end{bmatrix} \Rightarrow \begin{array}{c} x_i m_3 P_i = m_1 P_i \\ y_i m_3 P_i = m_2 P_i \end{array} \Rightarrow \begin{array}{c} x_i m_3 P_i - m_1 P_i = 0 \\ y_i m_3 P_i - m_2 P_i = 0 \end{array}$$

Equations from multiple points

$$
\begin{cases}
x_1 m_3 P_1 - m_1 P_1 = 0 \\
y_1 m_3 P_1 - m_2 P_1 = 0 \\
\vdots \\
x_n m_3 P_n - m_1 P_n = 0 \\
y_n m_3 P_n - m_2 P_n = 0
\end{cases}
\Rightarrow
\begin{pmatrix}
-P_1^T & 0^T & x_1 P_1^T \\
0^T & -P_1^T & y_1 P_1^T \\
\cdots & \cdots & \cdots \\
-P_n^T & 0^T & x_n P_n^T \\
0^T & -P_n^T & y_n P_n^T
\end{pmatrix}
\begin{pmatrix} m_1^T \\ m_2^T \\ m_3^T \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}
$$

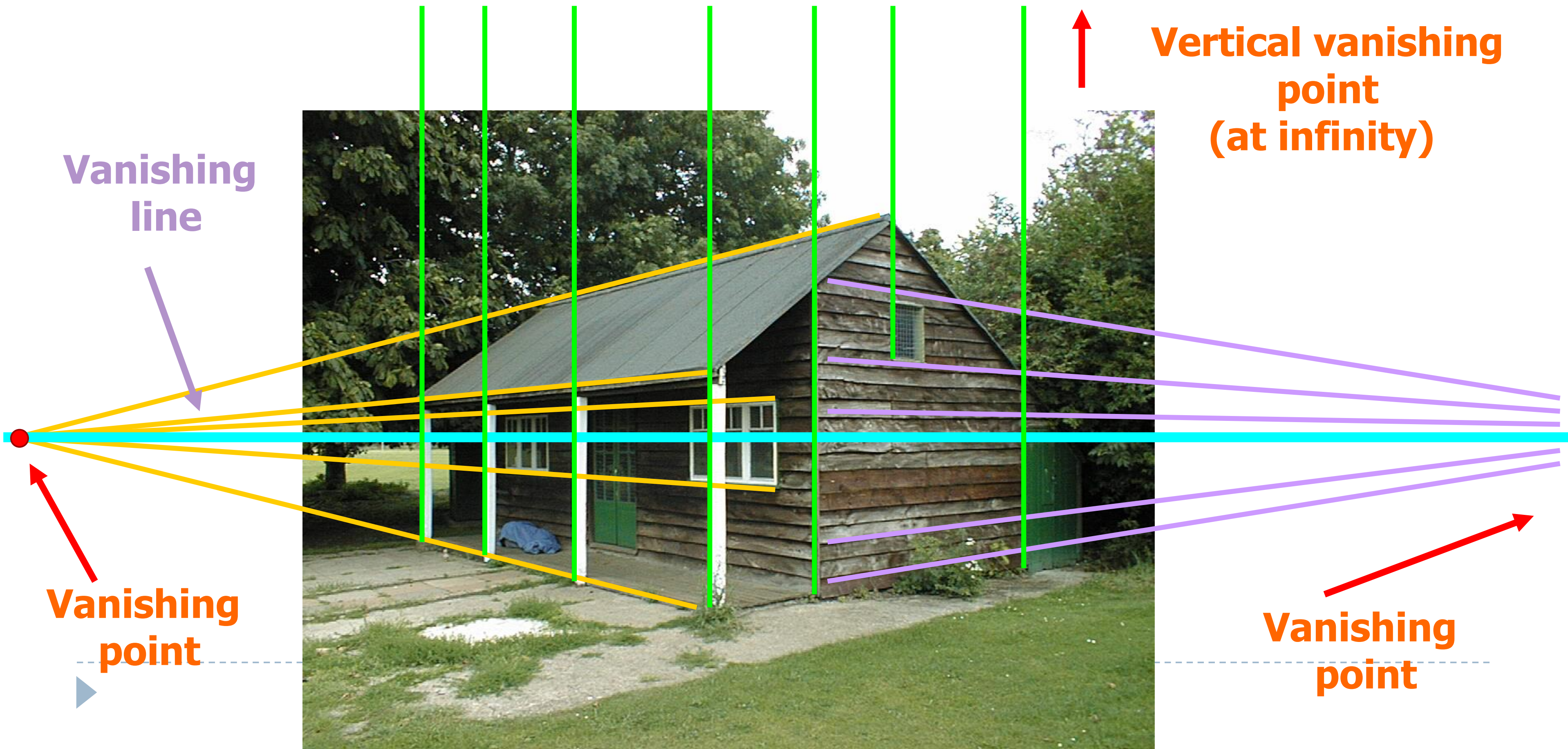
$$
\Downarrow
$$

$$
A_{2n\times 12} \begin{pmatrix} m_1^T \\ m_2^T \\ m_3^T \end{pmatrix}_{12\times 1} = 0
$$

Solve with SVD

Method 2:
Calibration using vanishing points

Find vanishing points corresponding to orthogonal directions



Calibration by orthogonal vanishing points

- Solve for K:

$$p = KRP$$

  - Use orthogonality as a constraint

For vanishing points

$$P^T P = 0$$

  - Model K with only f, c_x, c_y

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c \\ & & \end{bmatrix}$$

We will look at this more carefully next class

- What if you don't have three finite vanishing points?
  - Two finite VP: solve f, get valid c_x, c_y closest to image center
  - One finite VP: c_x, c_y is at vanishing point; can't solve for f

Calibration by orthogonal vanishing points

- Solve for R: $$p = KRP$$
  - Set directions of vanishing points
    - e.g., $\mathbf{X}_1 = [1, 0, 0]$
  - Each VP provides one column of **R**
  - Special properties of **R**
    - $\text{inv}(\mathbf{R}) = \mathbf{R}^T$
    - Each row an[illegible] ...gth

We will look at this more carefully next class

Extracting camera parameters

- Once we have found M up to scale, we can extract its intrinsic and extrinsic parameters
- When $s=0$ (no skew) we get

$$M = \begin{bmatrix} KR & KT \end{bmatrix} = \begin{bmatrix} \alpha r_1 + c_x r_3 & \alpha t_x + c_x t_z \\ \beta r_2 + c_y r_3 & \beta t_y + c_x t_z \\ r_3 & t_z \end{bmatrix}$$

Extracting intrinsic parameters

$$\frac{M}{\rho} = \left[\begin{array}{c|c} \alpha r_1 + c_x r_3 & \alpha t_x + c_x t_z \\ \beta r_2 + c_y r_3 & \beta t_y + c_x t_z \\ r_3 & t_z \end{array}\right]$$

$$\qquad\qquad A \qquad\qquad\qquad b$$

$$\rho = \frac{\pm 1}{|a_3|} \qquad c_x = \rho^2 (a_1 \bullet a_3)$$

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

Theorem (Faugeras, 1993)

$$M = K\begin{bmatrix} R & T \end{bmatrix} = \begin{bmatrix} KR & KT \end{bmatrix} = \begin{bmatrix} A & b \end{bmatrix}$$

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

Extracting intrinsic parameters

$$\frac{M}{\rho} = \left[\underbrace{\boxed{\begin{matrix} \alpha r_1 + c_x r_3 \\ \beta r_2 + c_y r_3 \\ r_3 \end{matrix}}}_{A} \quad \underbrace{\boxed{\begin{matrix} \alpha t_x + c_x t_z \\ \beta t_y + c_x t_z \\ t_z \end{matrix}}}_{b} \right]$$

$$\alpha = \rho^2 |a_1 \times a_3|$$
$$\beta = \rho^2 |a_2 \times a_3|$$

$\Rightarrow$ f

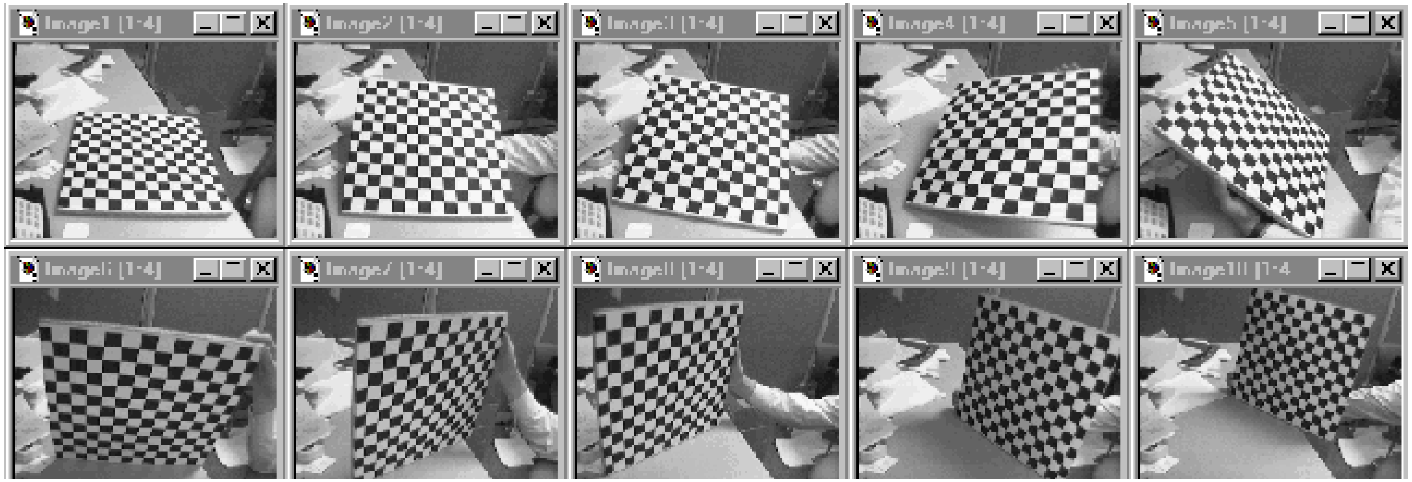
Extracting extrinsic parameters

$$\frac{M}{\rho} = \left[\underbrace{\boxed{\begin{matrix} \alpha r_1 + c_x r_3 \\ \beta r_2 + c_y r_3 \\ r_3 \end{matrix}}}_{A} \;\; \underbrace{\boxed{\begin{matrix} \alpha t_x + c_x t_z \\ \beta t_y + c_x t_z \\ t_z \end{matrix}}}_{b} \right]$$

$$r_1 = \frac{a_2 \times a_3}{|a_2 \times a_3|} \qquad r_3 = a_3$$

$$r_2 = r_3 \times r_1 \qquad T = \rho K^{-1} b$$

Alternative: multi-plane calibration

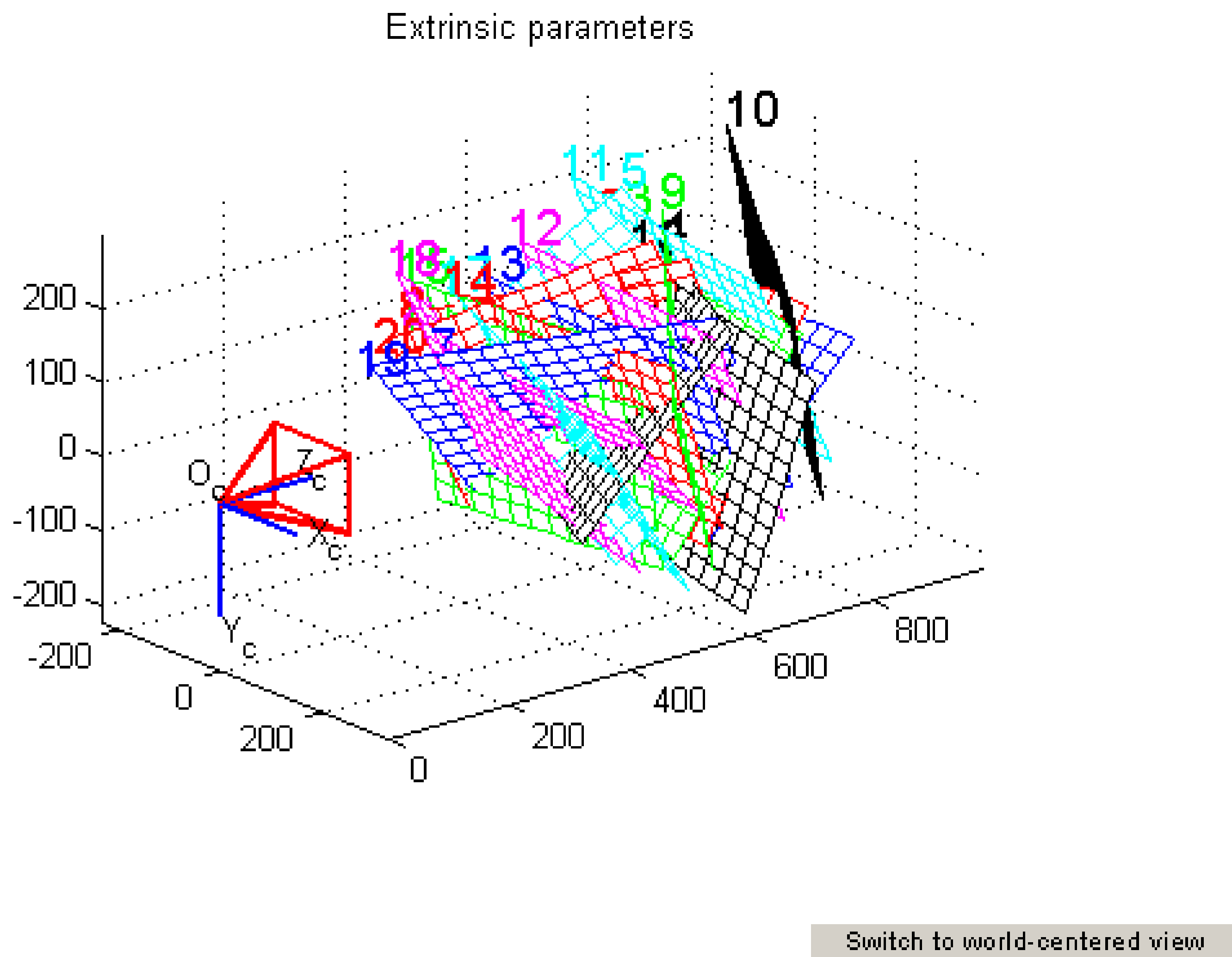


Images courtesy Jean-Yves Bouguet, Intel Corp.

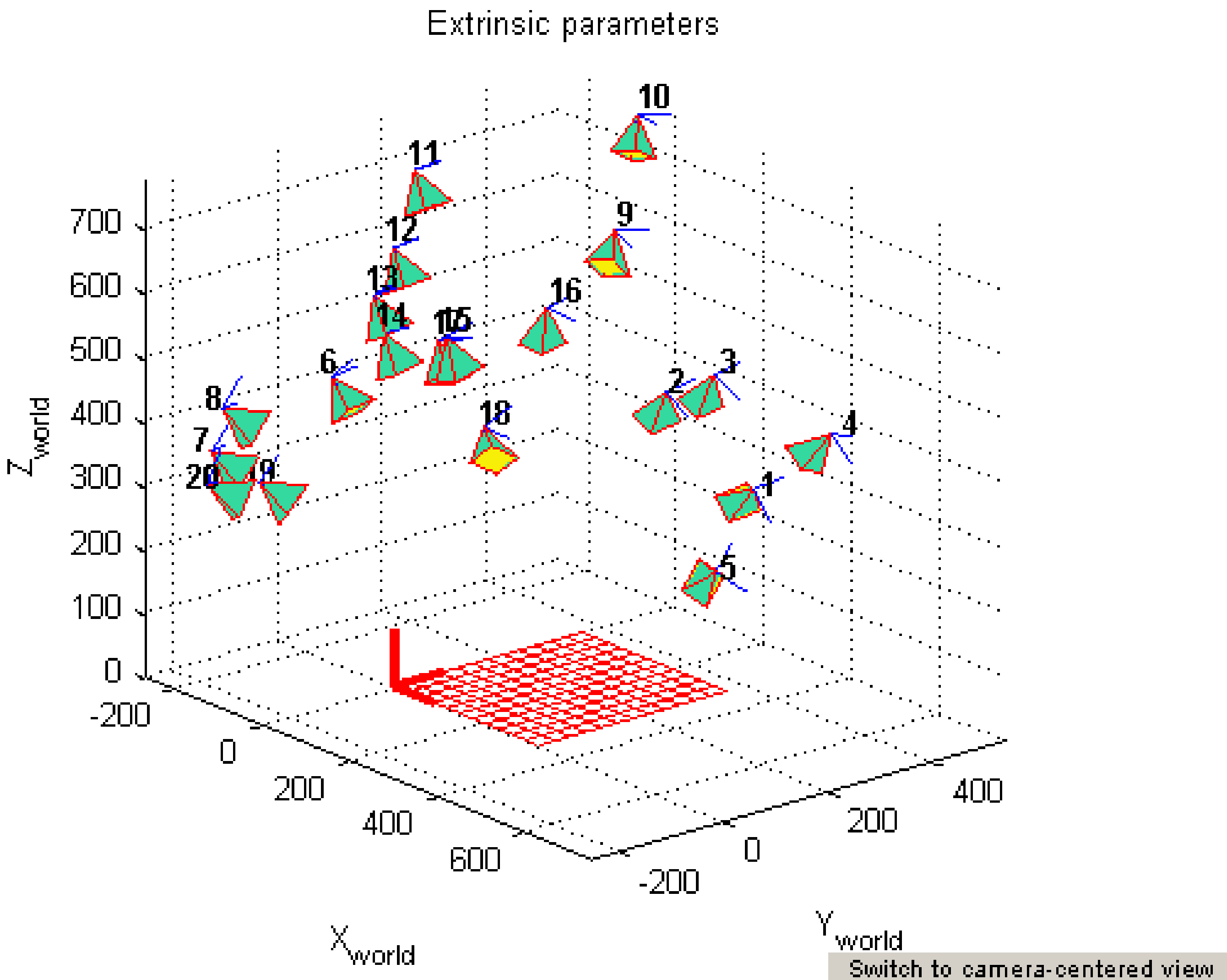
Advantage

- Only requires a plane
- Don't have to know positions/orientations
- Good code available online! (including in OpenCV)
  - Matlab version by Jean-Yves Bouget: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  - Zhengyou Zhang's web site: http://research.microsoft.com/~zhang/Calib/

Calibration demo



Calibration demo



End – Camera calibration

Now you know how it works