Image Formation I
Chapter 1 (Forsyth&Ponce)
Cameras

Guido Gerig

CS 6320 Spring 2013

Acknowledgements:

- Slides used from Prof. Trevor Darrell, (http://www.eecs.berkeley.edu/~trevor/CS280.html)
- Some slides modified from Marc Pollefeys, UNC Chapel Hill. Other slides and illustrations from J. Ponce, addendum to course book.

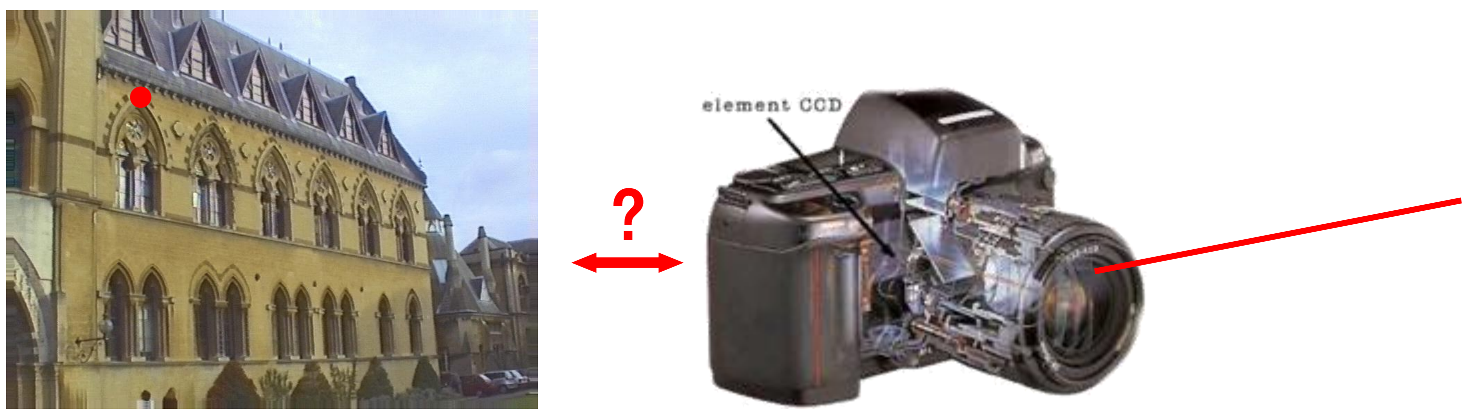
GEOMETRIC CAMERA MODELS

- The Intrinsic Parameters of a Camera
- The Extrinsic Parameters of a Camera
- The General Form of the Perspective Projection Equation
- Line Geometry

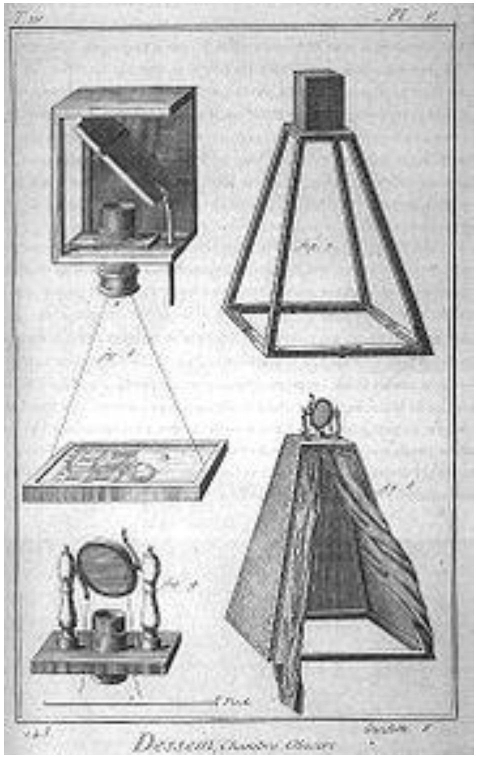
Reading: Chapter 1.

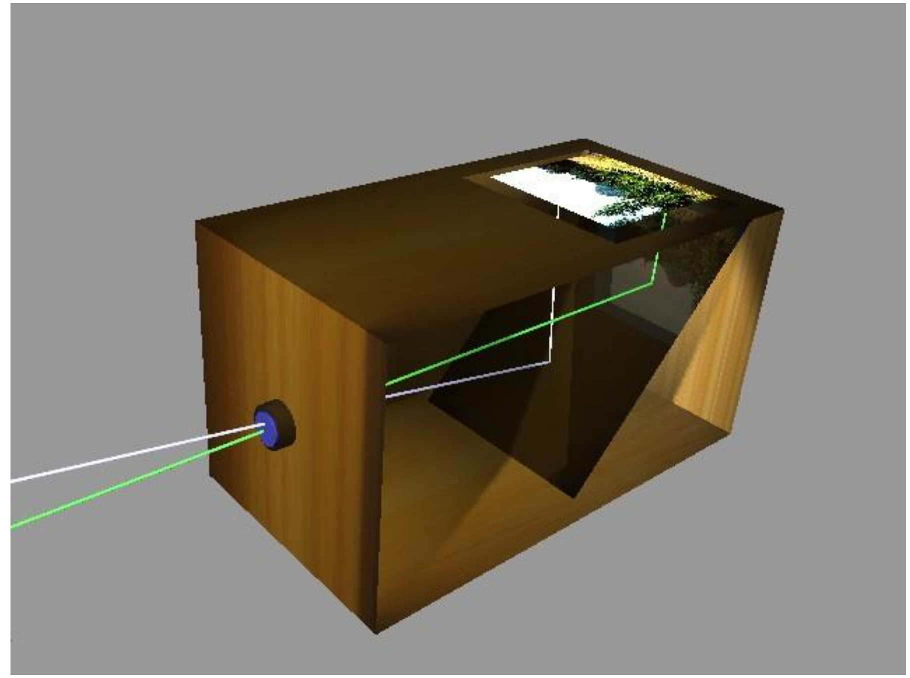
Camera model

Relation between pixels and rays in space



Camera obscura + lens





The **camera obscura** (Latin for 'dark room') is an optical device that projects an image of its surroundings on a screen (source Wikipedia).

Physical parameters of image formation

- Geometric
  - Type of projection
  - Camera pose
- Photometric
  - Type, direction, intensity of light reaching sensor
  - Surfaces' reflectance properties
- Optical
  - Sensor's lens type
  - focal length, field of view, aperture
- Sensor
  - sampling, etc.

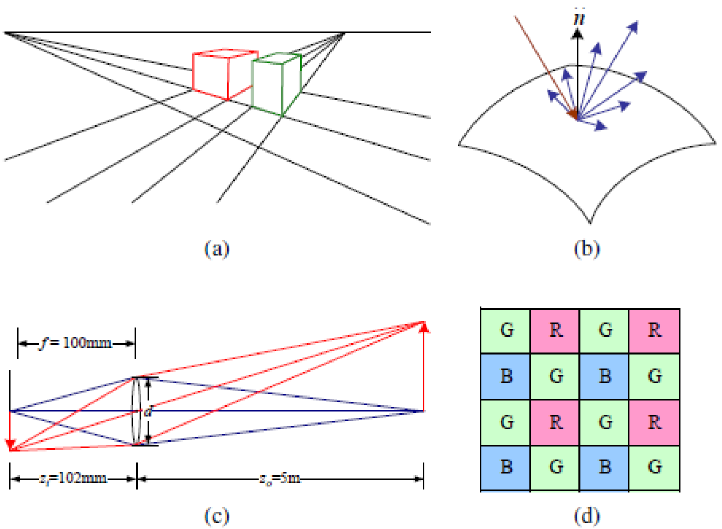


Figure 2.1 A few components of the image formation process: (a) perspective projection; (b) light scattering when hitting a surface; (c) lens optics; (d) Bayer color filter array.

Physical parameters of image formation

- Geometric
  - Type of projection
  - Camera pose
- Optical
  - Sensor's lens type
  - focal length, field of view, aperture
- Photometric
  - Type, direction, intensity of light reaching sensor
  - Surfaces' reflectance properties
- Sensor
  - sampling, etc.

Perspective and art

- Use of correct perspective projection indicated in 1^{st} century B.C. frescoes
- Skill resurfaces in Renaissance: artists develop systematic methods to determine perspective projection (around 1480-1515)

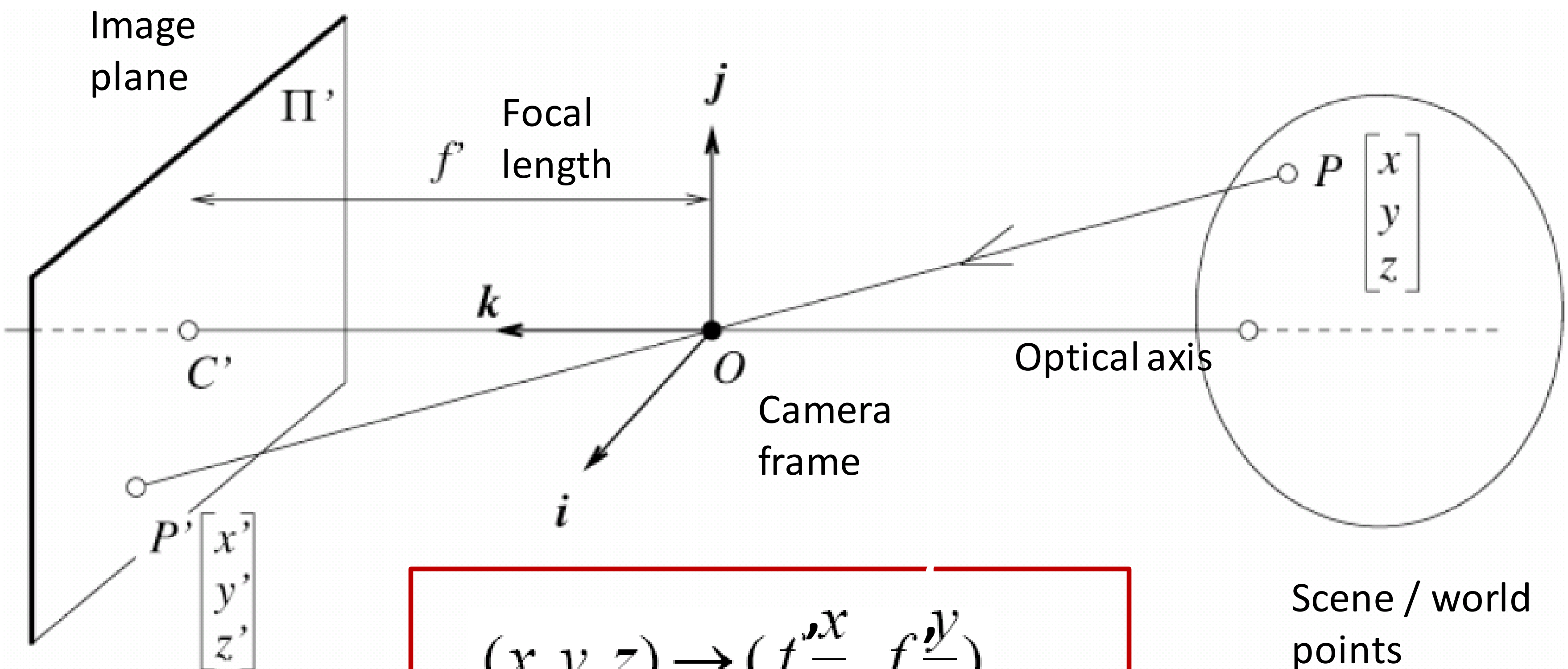
Raphael

Durer, 1525

K. Grauman

Perspective projection equations

- 3d world mapped to 2d projection in image plane

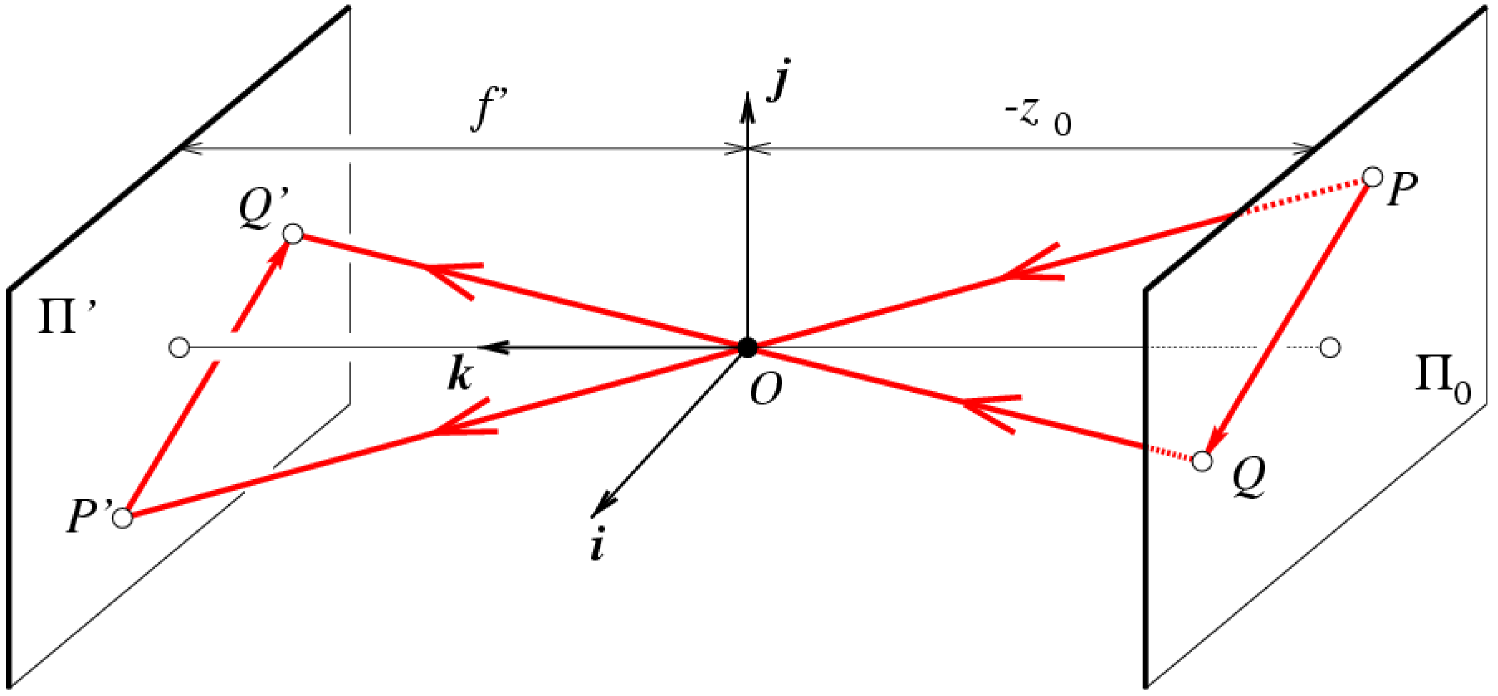


$$(x, y, z) \rightarrow (f' \frac{x}{z}, f' \frac{y}{z})$$

Scene point $\longrightarrow$ Image coordinates

Forsyth and Ponce

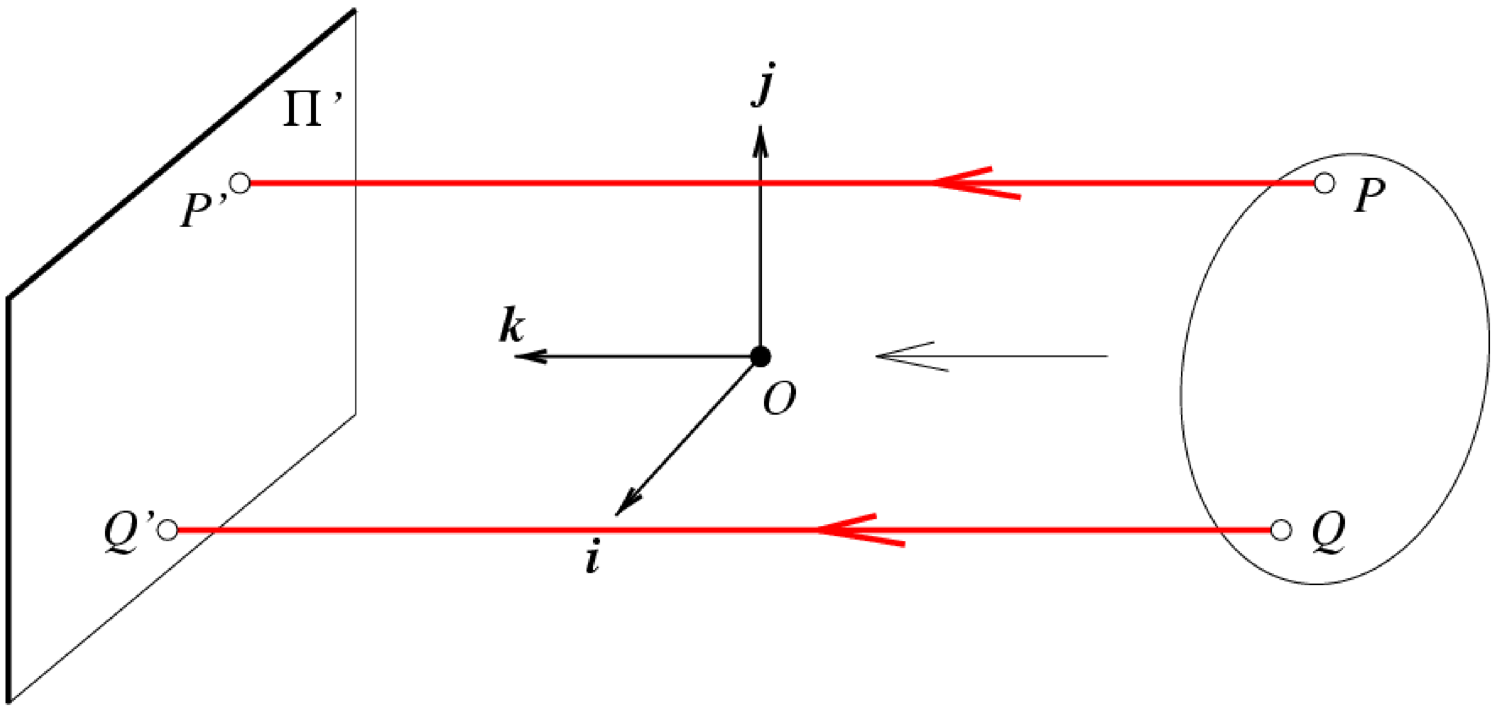
Affine projection models: Weak perspective projection



$$\begin{cases} x' = -mx \\ y' = -my \end{cases}$$ where $m = \frac{f'}{z_0}$ is the magnification.

When the scene relief is small compared to its distance from the Camera, m can be taken constant: weak perspective projection.

Affine projection models: Orthographic projection



$$\begin{cases} x' = x \\ y' = y \end{cases}$$

When the camera is at a (roughly constant) distance from the scene, take $m=1$.

Homogeneous coordinates

Is this a linear transformation?

- no—division by z is nonlinear

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Slide by Steve Seitz

Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates:

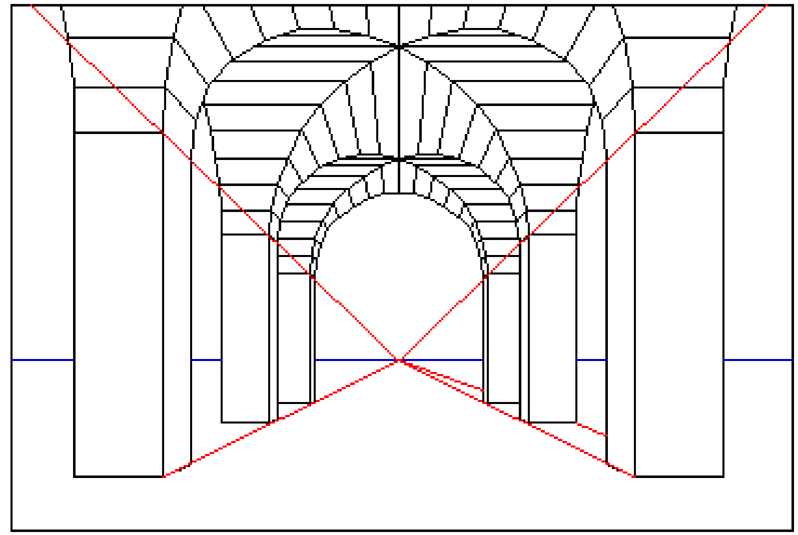
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f' \end{bmatrix} \Rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$$

divide by the third coordinate to convert back to non-homogeneous coordinates

Complete mapping from world points to image pixel positions?

Slide by Steve Seitz

Points at infinity, vanishing points



Points from infinity represent rays into camera which are close to the optical axis.

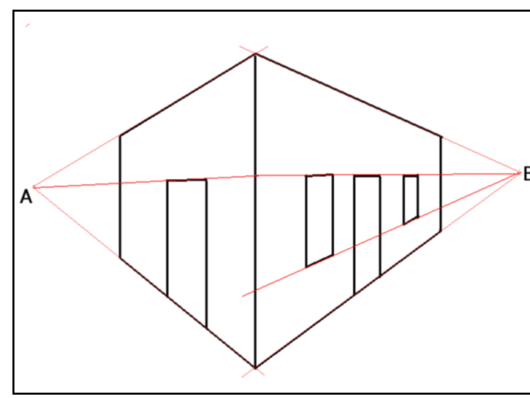
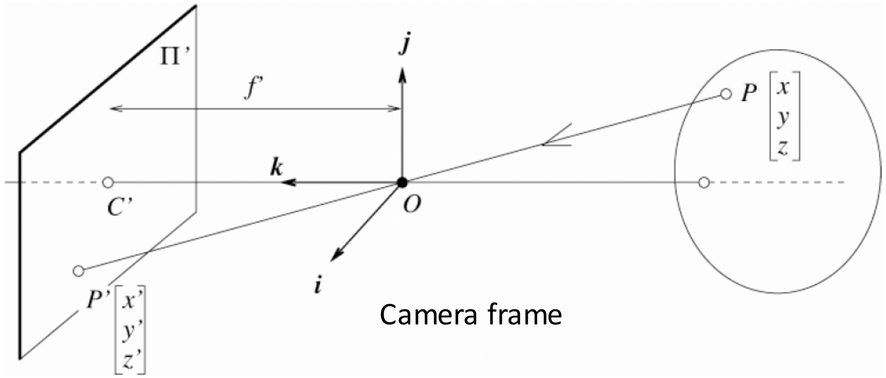


Image source: wikipedia

Perspective projection & calibration

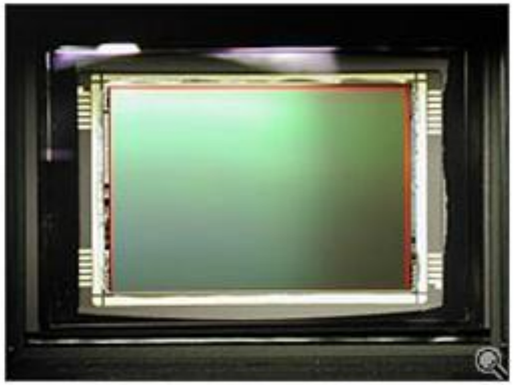
- Perspective equations so far in terms of *camera's* reference frame....
- Camera's *intrinsic* and *extrinsic* parameters needed to calibrate geometry.



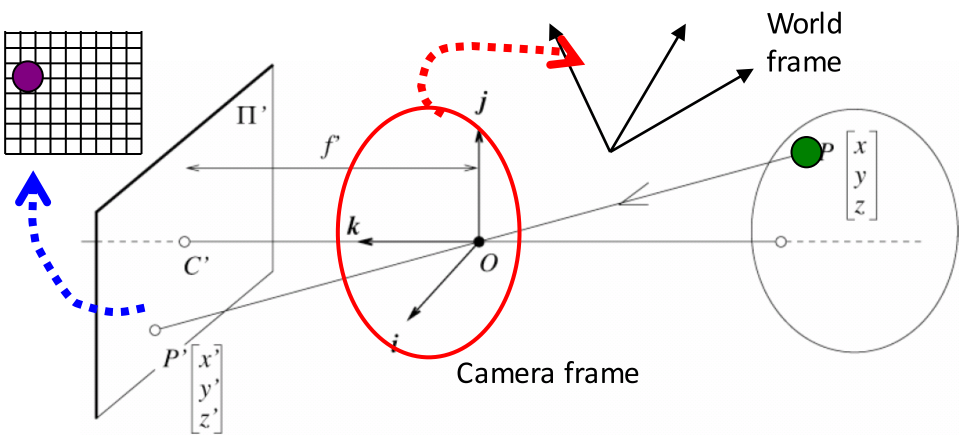
K. Grauman

The CCD camera

CCD camera



Perspective projection & calibration



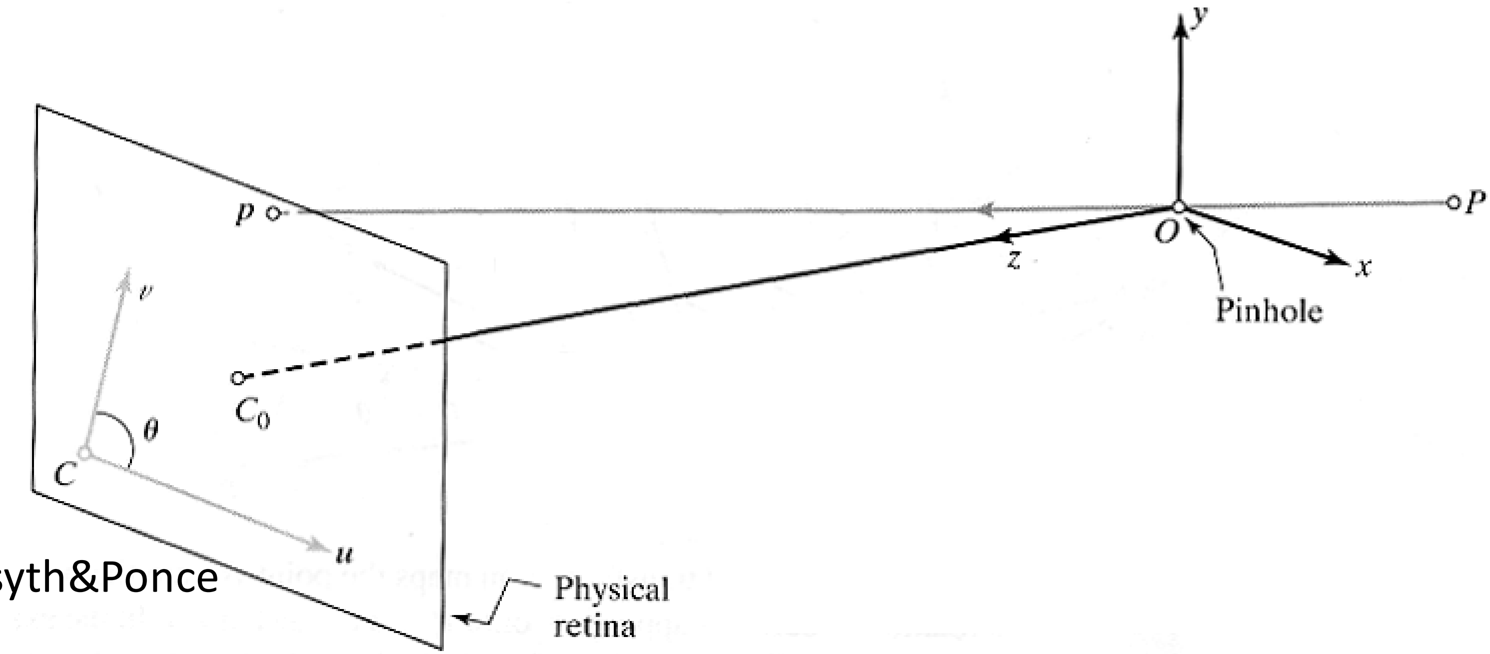
Extrinsic:
Camera frame ←→World frame

Intrinsic:
Image coordinates relative to camera ←→ Pixel coordinates

3D point (4x1)

K. Grauman

Intrinsic parameters: from idealized world coordinates to pixel values



Forsyth&Ponce

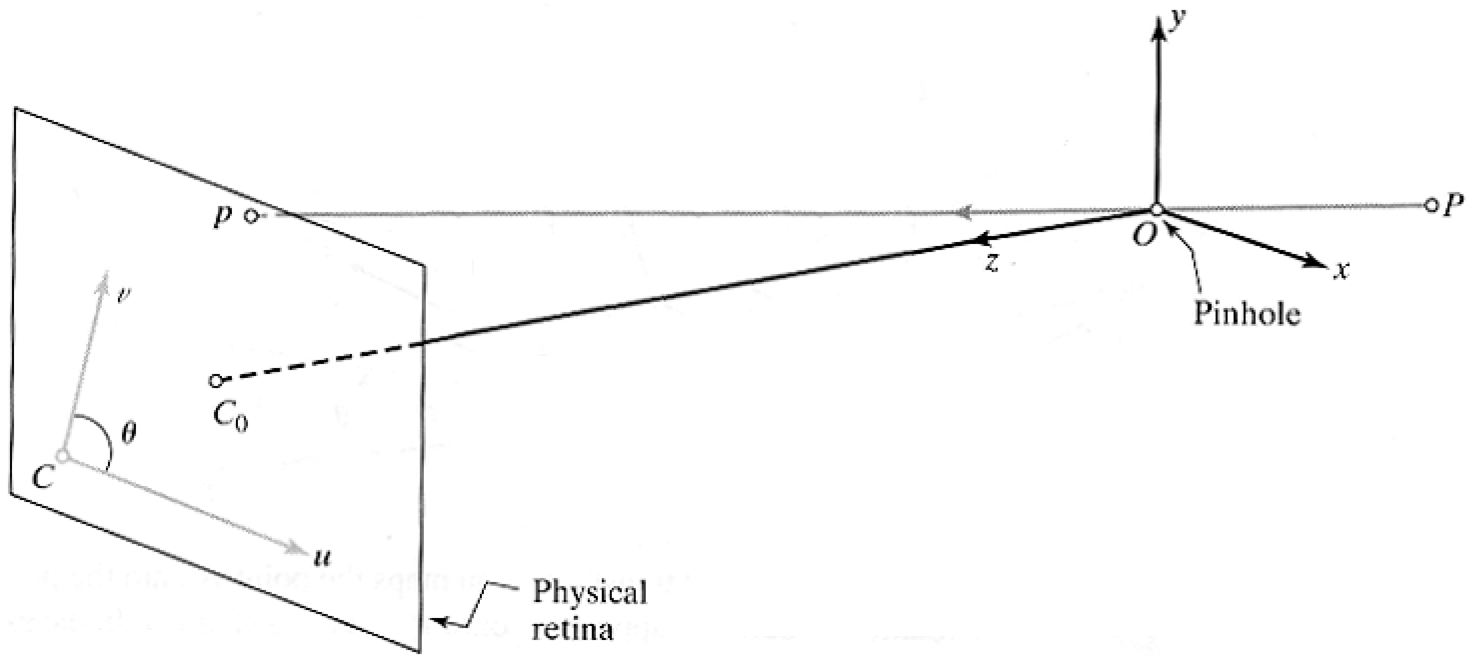
Perspective projection

$$u = f\frac{x}{z}$$

$$v = f\frac{y}{z}$$

W. Freeman

Intrinsic parameters



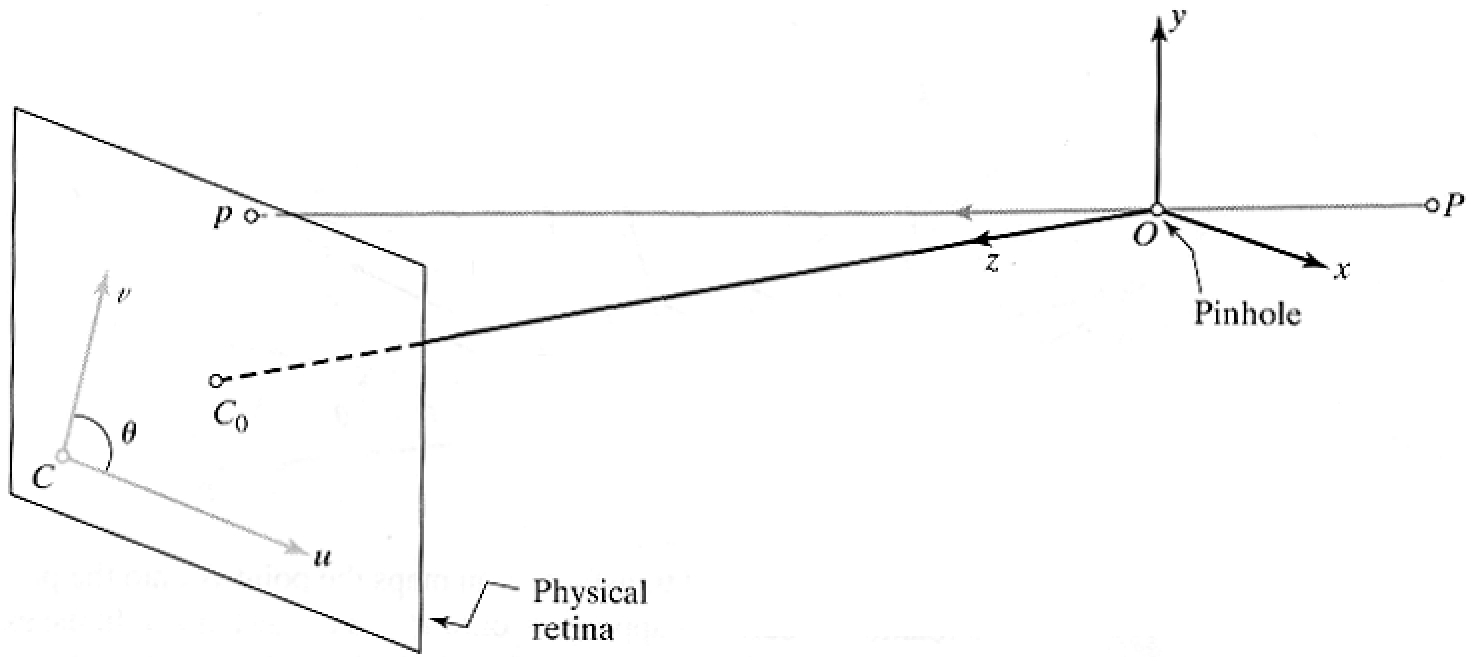
But "pixels" are in some arbitrary spatial units

$$u = \alpha \frac{x}{z}$$

$$v = \alpha \frac{y}{z}$$

W. Freeman

Intrinsic parameters



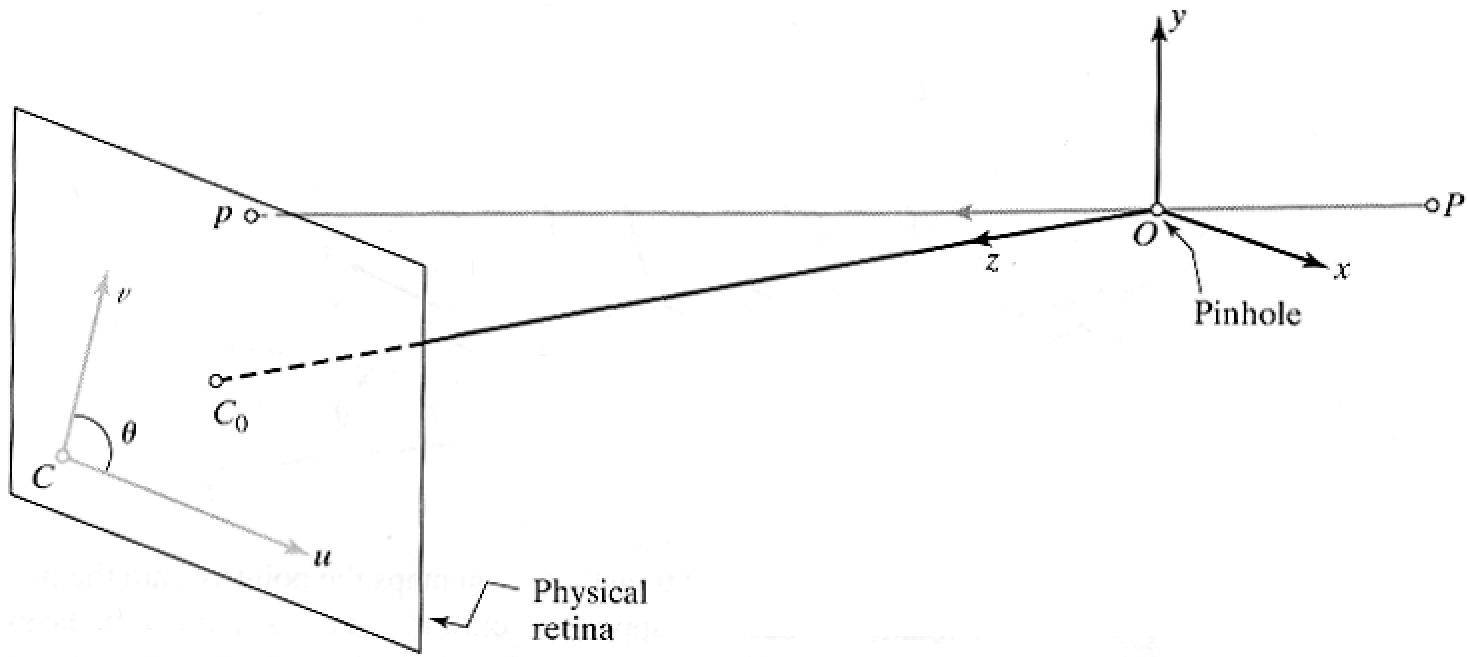
Maybe pixels are not square

$$u = \alpha \frac{x}{z}$$

$$v = \beta \frac{y}{z}$$

W. Freeman

Intrinsic parameters



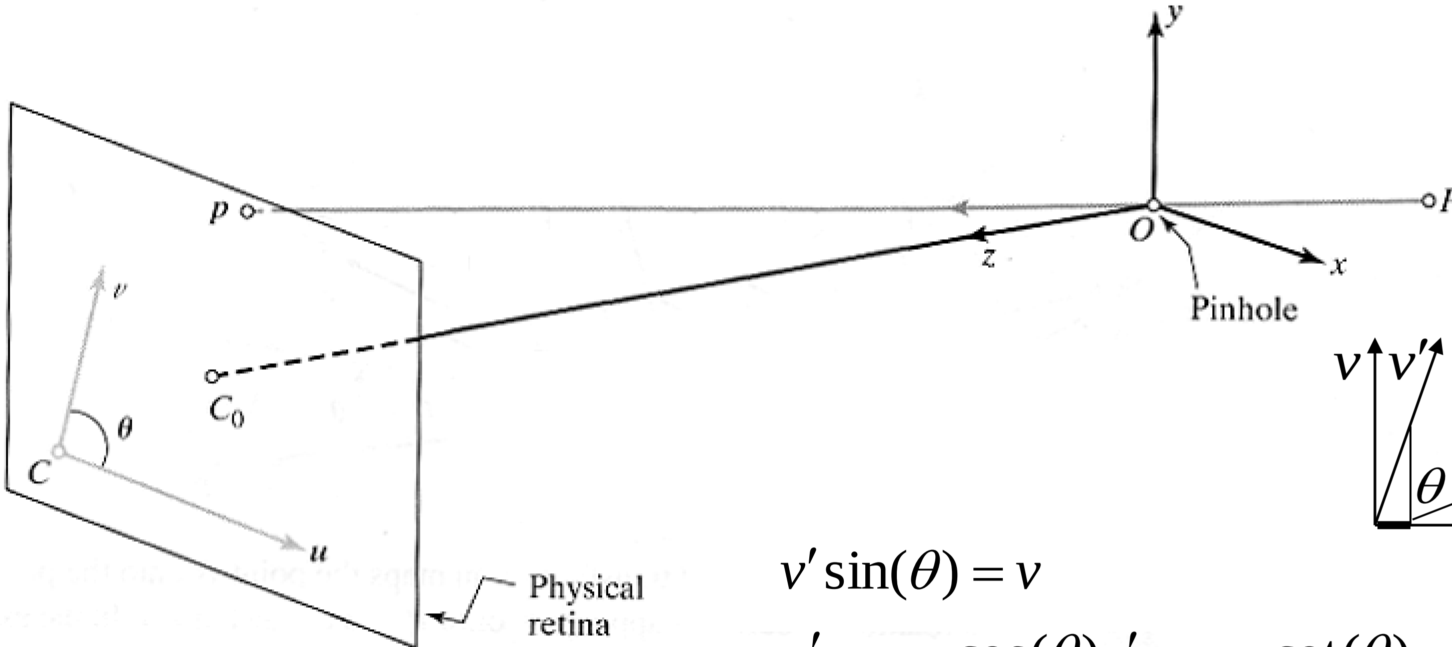
We don't know the origin of our camera pixel coordinates

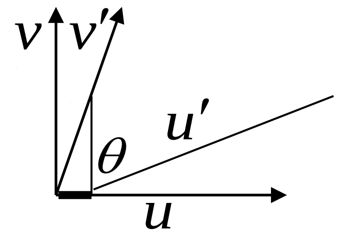
$$u = \alpha \frac{x}{z} + u_0$$

$$v = \beta \frac{y}{z} + v_0$$

W. Freeman

Intrinsic parameters





$$v' \sin(\theta) = v$$

$$u' = u - \cos(\theta)v' = u - \cot(\theta)v$$

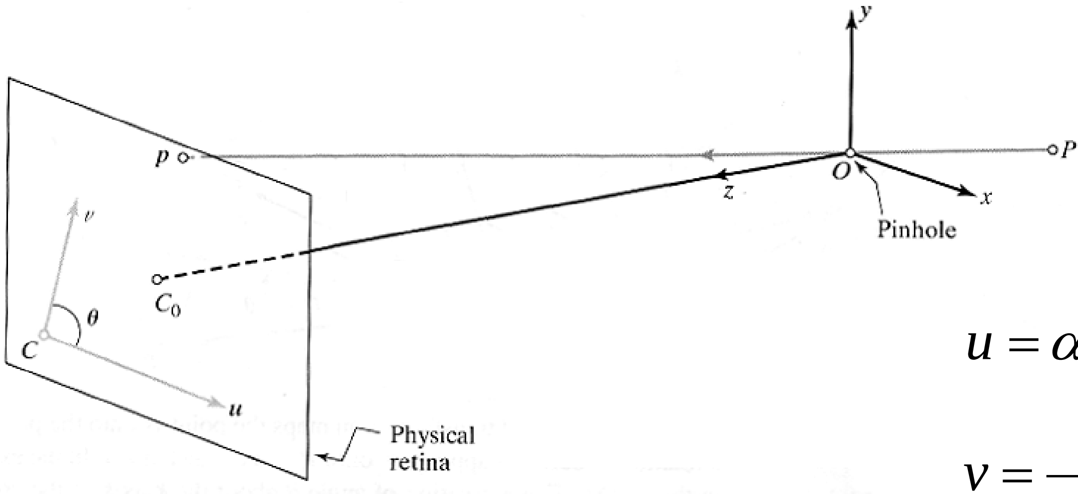
May be skew between camera pixel axes

$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

W. Freeman

Intrinsic parameters, homogeneous coordinates



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Using homogenous coordinates, we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} \alpha & -\alpha \cot(\theta) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

or:

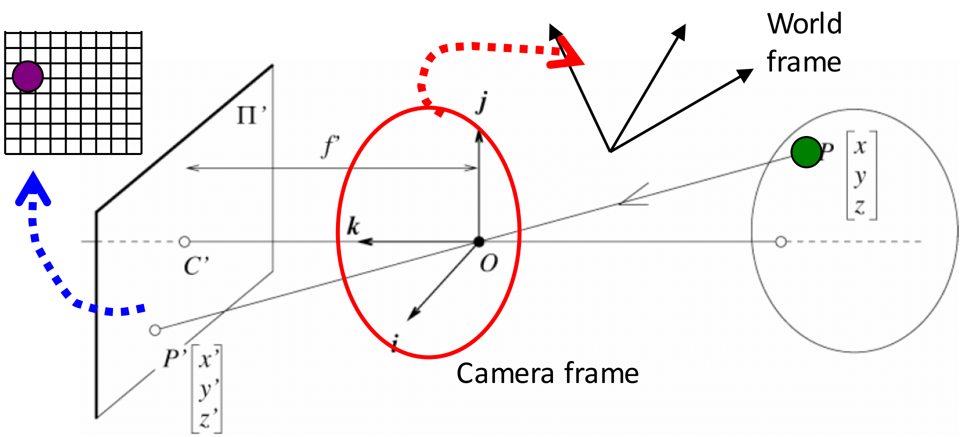
$$\vec{p} = \frac{1}{z} (\mathrm{K}) \; {}^{C}\vec{p}$$

In pixels

In camera-based coords

W. Freeman

Perspective projection & calibration



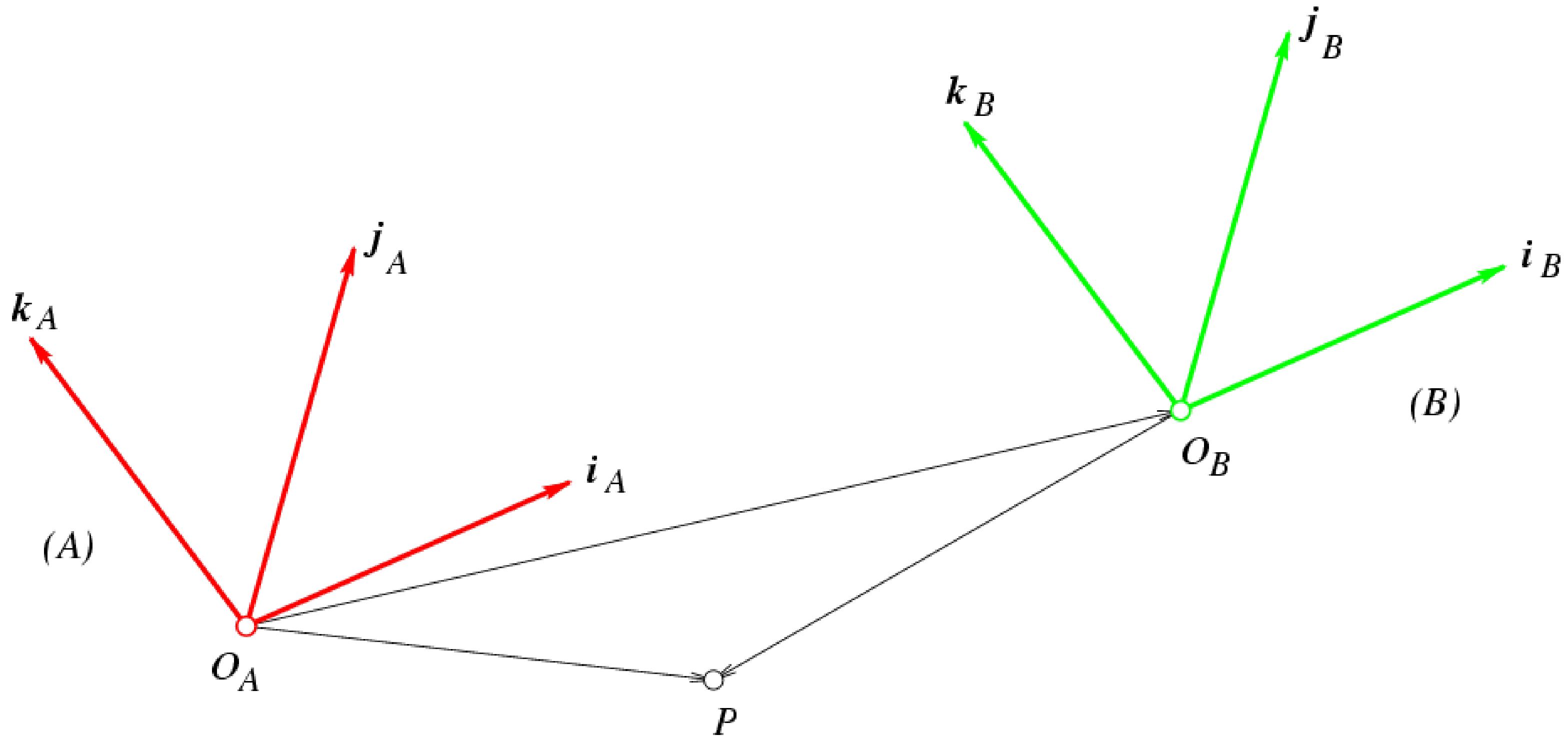
Extrinsic:
Camera frame ←→ World frame

Intrinsic:
Image coordinates relative to camera ←→ Pixel coordinates

3D point (4x1)

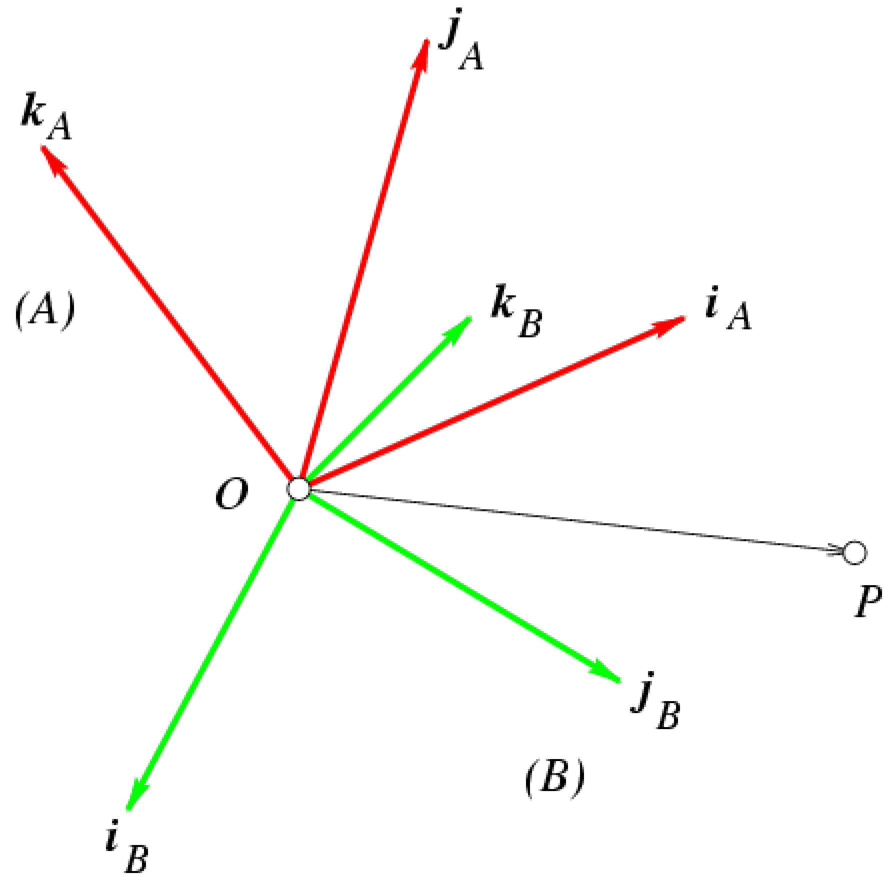
K. Grauman

Coordinate Changes: Pure Translations



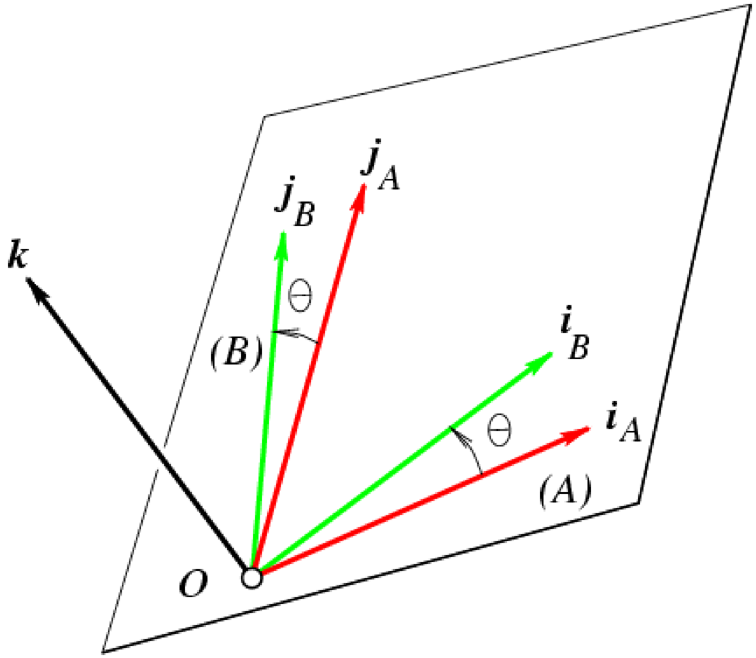
$$\overrightarrow{O_B P} = \overrightarrow{O_B O_A} + \overrightarrow{O_A P} \;,\quad {}^B P = {}^A P + {}^B O_A$$

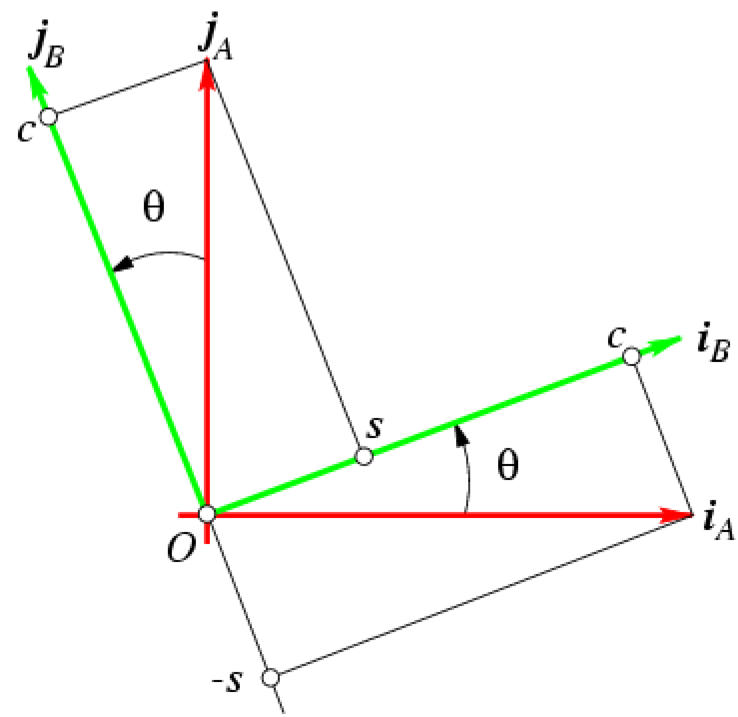
Coordinate Changes: Pure Rotations



$$
{}^B_A R = \begin{bmatrix} \mathbf{i}_A . \mathbf{i}_B & \mathbf{j}_A . \mathbf{i}_B & \mathbf{k}_A . \mathbf{i}_B \\ \mathbf{i}_A . \mathbf{j}_B & \mathbf{j}_A . \mathbf{j}_B & \mathbf{k}_A . \mathbf{j}_B \\ \mathbf{i}_A . \mathbf{k}_B & \mathbf{j}_A . \mathbf{k}_B & \mathbf{k}_A . \mathbf{k}_B \end{bmatrix} = ({}^B\mathbf{i}_A, {}^B\mathbf{j}_A, {}^B\mathbf{k}_A) = \begin{bmatrix} {}^A\mathbf{i}_B^T \\ {}^A\mathbf{j}_B^T \\ {}^A\mathbf{k}_B^T \end{bmatrix}
$$

Coordinate Changes: Rotations about the *k* Axis





$$
{}^{B}_{A}R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

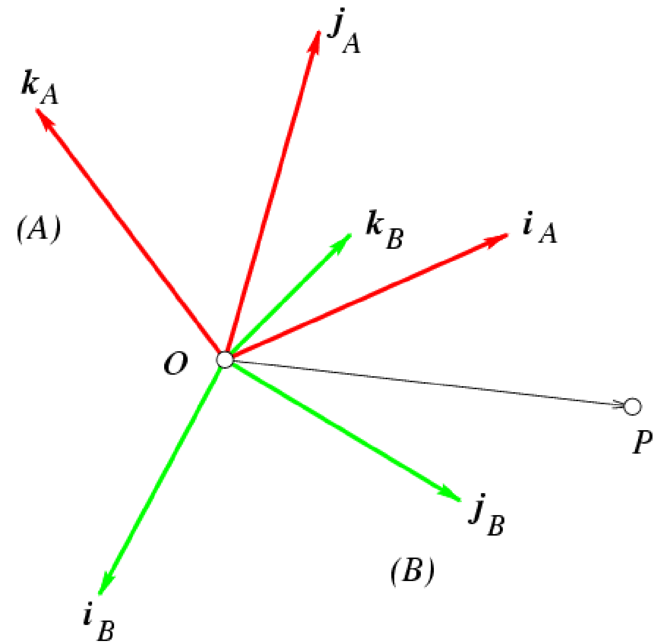
A rotation matrix is characterized by the following properties:

- Its inverse is equal to its transpose, and
- its determinant is equal to 1.

Or equivalently:

- Its rows (or columns) form a right-handed orthonormal coordinate system.

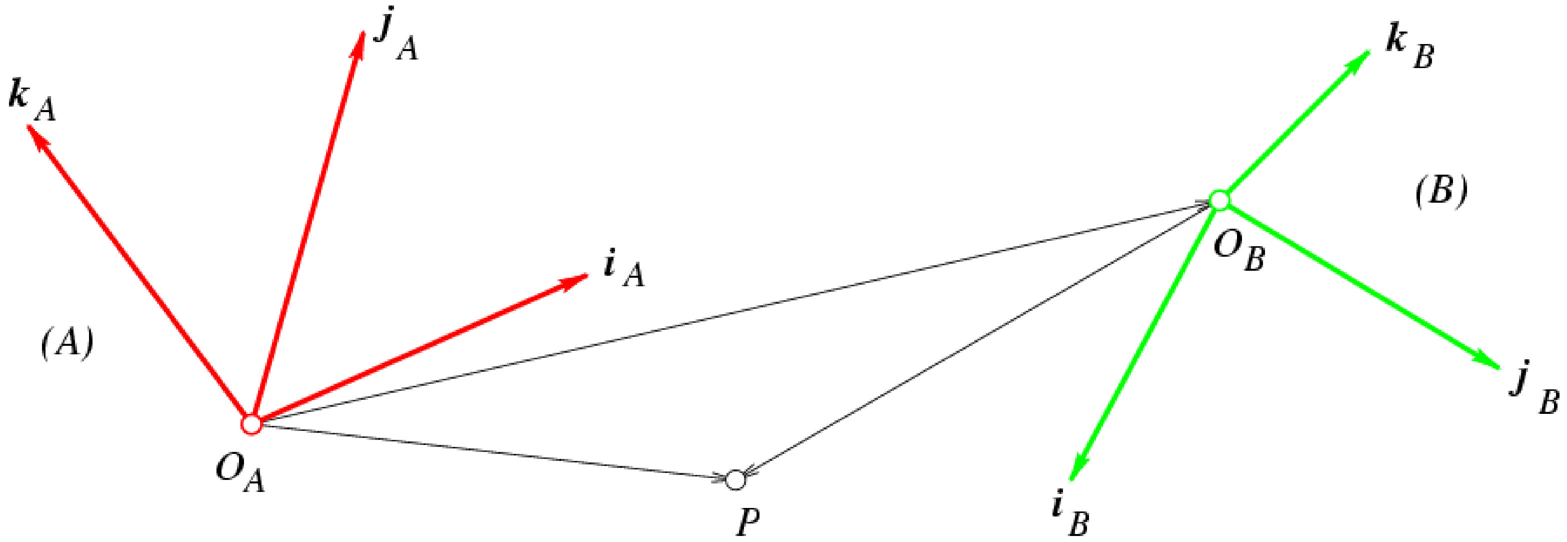
Coordinate Changes:
Pure Rotations



$$\overrightarrow{OP} = \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^A x \\ {}^A y \\ {}^A z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix} \begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix}$$

$$\Rightarrow \quad {}^B P = {}^B_A R \, {}^A P$$

Coordinate Changes: Rigid Transformations



$$ {}^{B}P = {}^{B}_{A}R\,{}^{A}P + {}^{B}O_A $$

Block Matrix Multiplication

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

What is AB ?

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

Homogeneous Representation of Rigid Transformations

$$\begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R & {}^{B}O_{A} \\ \mathbf{0}^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R\,{}^{A}P + {}^{B}O_{A} \\ 1 \end{bmatrix} = {}^{B}_{A}T \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix}$$

Extrinsic parameters: translation and rotation of camera frame

$$^{C}\vec{p} = {}^{C}_{W}R \; {}^{W}\vec{p} + {}^{C}_{W}\vec{t}$$

Non-homogeneous coordinates

$$\begin{pmatrix} {}^{C}\vec{p} \\ \\ \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{C}_{W}R & - & {}^{C}_{W}\vec{t} \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{W}\vec{p} \\ \\ \end{pmatrix}$$

Homogeneous coordinates

W. Freeman

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

pixels

$$\vec{p} = \frac{1}{z} K \; {}^{C}\vec{p}$$

Intrinsic

World coordinates

Camera coordinates

$$\begin{pmatrix} \\ {}^{C}\vec{p} \\ \\ \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{C}_{W}R & - & {}^{C}_{W}\vec{t} \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \\ {}^{W}\vec{p} \\ \\ \end{pmatrix}$$

Extrinsic

$$\vec{p} = \frac{1}{z} K \begin{pmatrix} {}^{C}_{W}R & {}^{C}_{W}\vec{t} \\ 0,0,0 & 1 \end{pmatrix} {}^{W}\vec{p}$$

$$\vec{p} = \frac{1}{z} M \; {}^{W}\vec{p}$$

Forsyth&Ponce

W. Freeman

Other ways to write the same equation

pixel coordinates

world coordinates

$$\vec{p} = \frac{1}{z} M \; {}^{W}\vec{p}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} . & m_1^T & . & . \\ . & m_2^T & . & . \\ . & m_3^T & . & . \end{pmatrix} \begin{pmatrix} {}^{W}p_x \\ {}^{W}p_y \\ {}^{W}p_z \\ 1 \end{pmatrix}$$

Conversion back from homogeneous coordinates leads to (note that $z = m^T{}_3 * P$) :

$$\begin{cases} u = \dfrac{m_1 \cdot \vec{P}}{m_3 \cdot \vec{P}} \\[2ex] v = \dfrac{m_2 \cdot \vec{P}}{m_3 \cdot \vec{P}} \end{cases}$$

W. Freeman

Extrinsic Parameters

- When the camera frame (C) is different from the world frame (W),

$$\begin{pmatrix} {}^C P \\ 1 \end{pmatrix} = \begin{pmatrix} {}^C_W\mathcal{R} & {}^C O_W \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} {}^W P \\ 1 \end{pmatrix}.$$

- Thus,

$$\boxed{\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P},} \quad \text{where} \quad \begin{cases} \mathcal{M} = \mathcal{K}\,(\,\mathcal{R} \quad \boldsymbol{t}\,)\,, \\ \mathcal{R} = {}^C_W\mathcal{R}, \\ \boldsymbol{t} = {}^C O_W, \\ \boldsymbol{P} = \begin{pmatrix} {}^W P \\ 1 \end{pmatrix}. \end{cases}$$

- Note: z is *not* independent of $\mathcal{M}$ and $\boldsymbol{P}$:

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \end{pmatrix} \Longrightarrow z = \boldsymbol{m}_3 \cdot \boldsymbol{P}, \quad \text{or} \quad \begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}, \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}. \end{cases}$$

Explicit Form of the Projection Matrix

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

Note: If $\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b})$ then $|\boldsymbol{a}_3| = 1$.

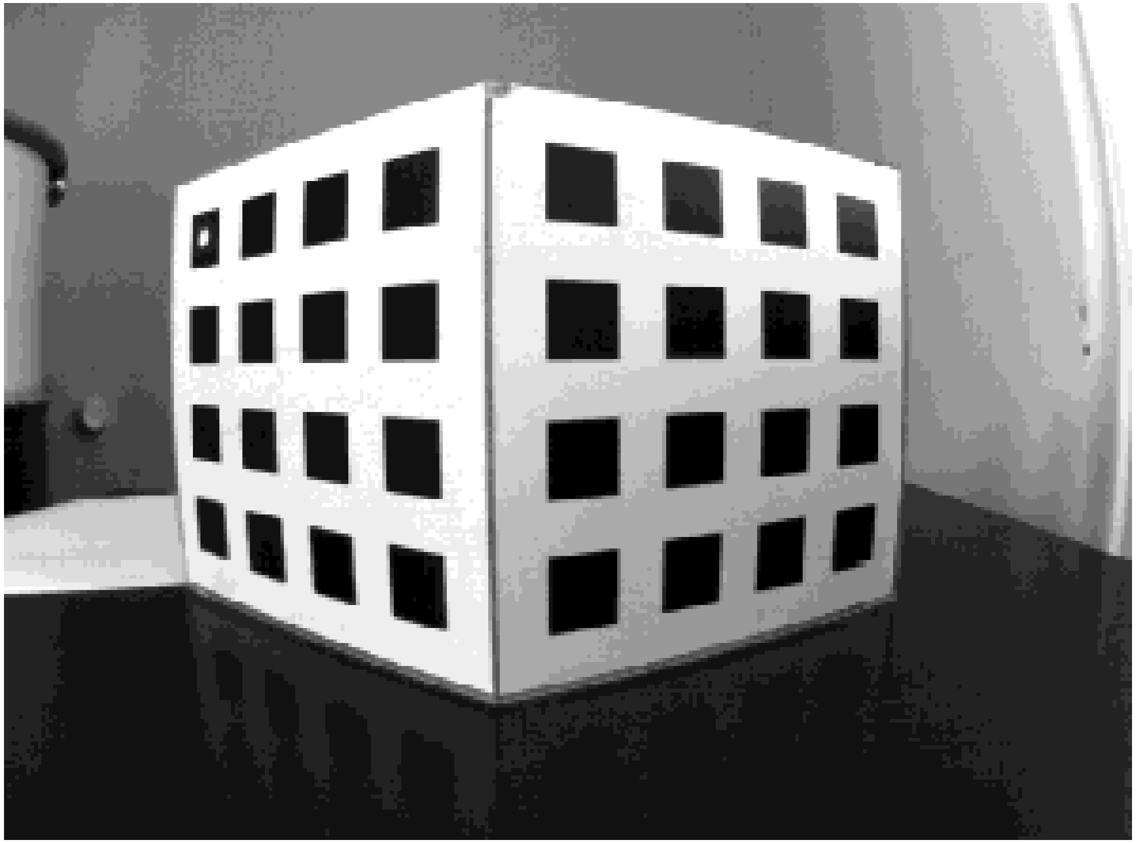
Replacing $\mathcal{M}$ by $\lambda\mathcal{M}$ in

$$\begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}} \end{cases}$$

does not change u and v.

M is only defined up to scale in this setting!!

Calibration target



The Opti-CAL Calibration Target Image

Find the position, u_i and v_i, in pixels, of each calibration object feature point.

http://www.kinetic.bc.ca/CompVision/opti-CAL.html