Introduction to computer vision 2

Jean Ponce and Elena Sizikova
jean.ponce@inria.fr, es5223@nyu.edu

Examples of vision tasks
Camera geometry and calibration

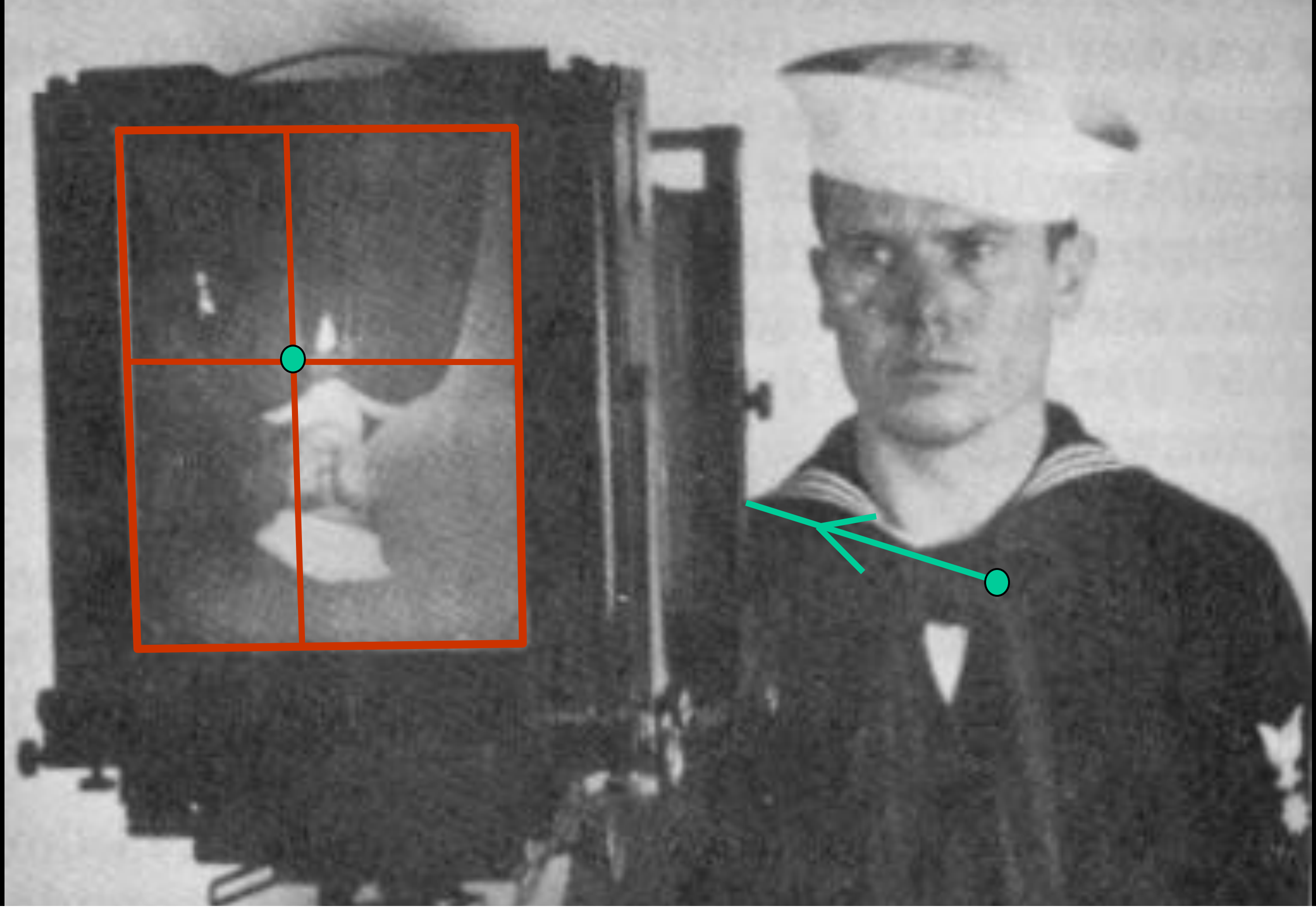
Course outline:

0. Introduction
1. Camera geometry and calibration
2. Filtering and feature detection
3. Radiometry and color
4. Texture and image segmentation
5. Stereopsis
6. Structure from motion and 3D models from images
7. Object recognition - historical perspective
8. CNNs for object classification and detection
9. 3D CNNs and applications to medical imaging
10. Weakly-supervised and unsupervised approaches to image and video interpretation

Important note:

- This class if about computer vision, not about
- deep learning, pytorch, etc.

They are formed by the projection of 3D objects.



They are also matrices of grey/color values
(Remember the green « Matrix » stuff..)

Images are two-dimensional patterns of brightness values.

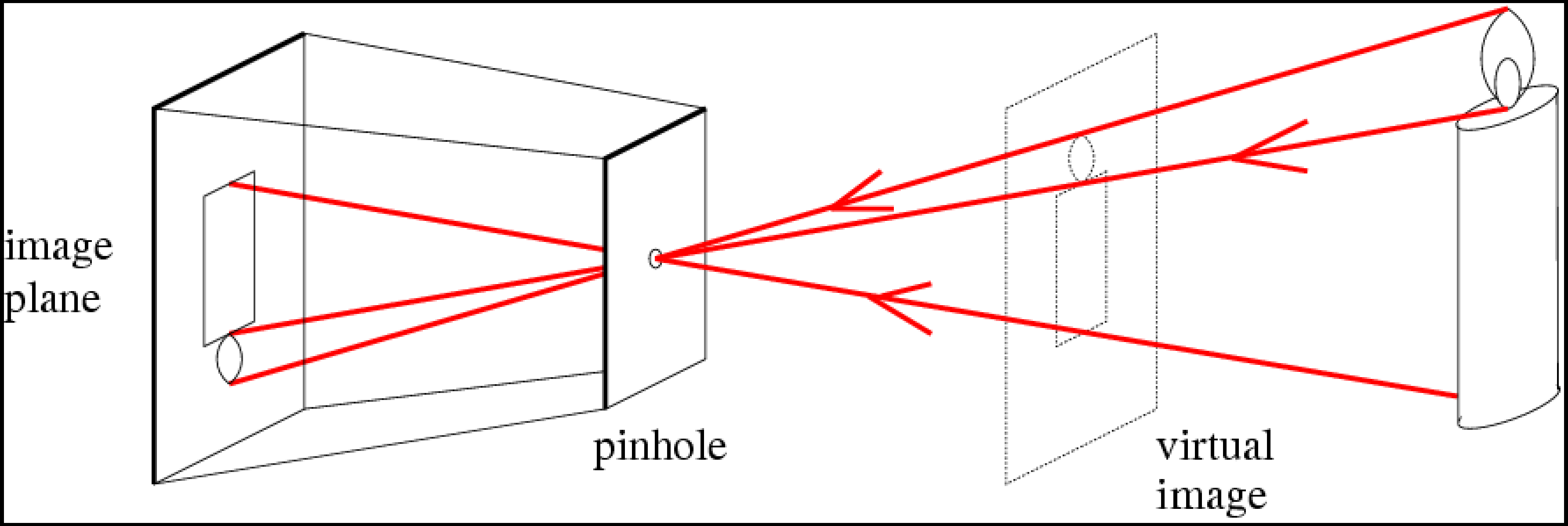


image
plane
pinhole
virtual
image


Canon
F-1
S.S.C. ASPHERICAL
CANON
LENS MADE IN JAPAN

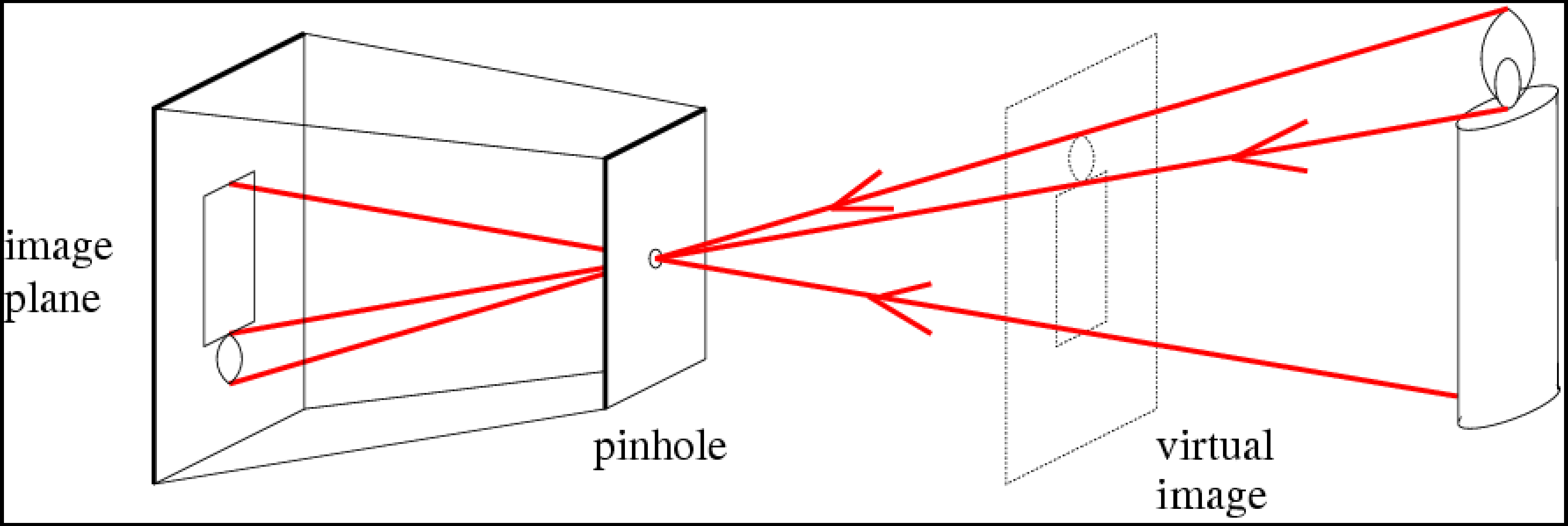
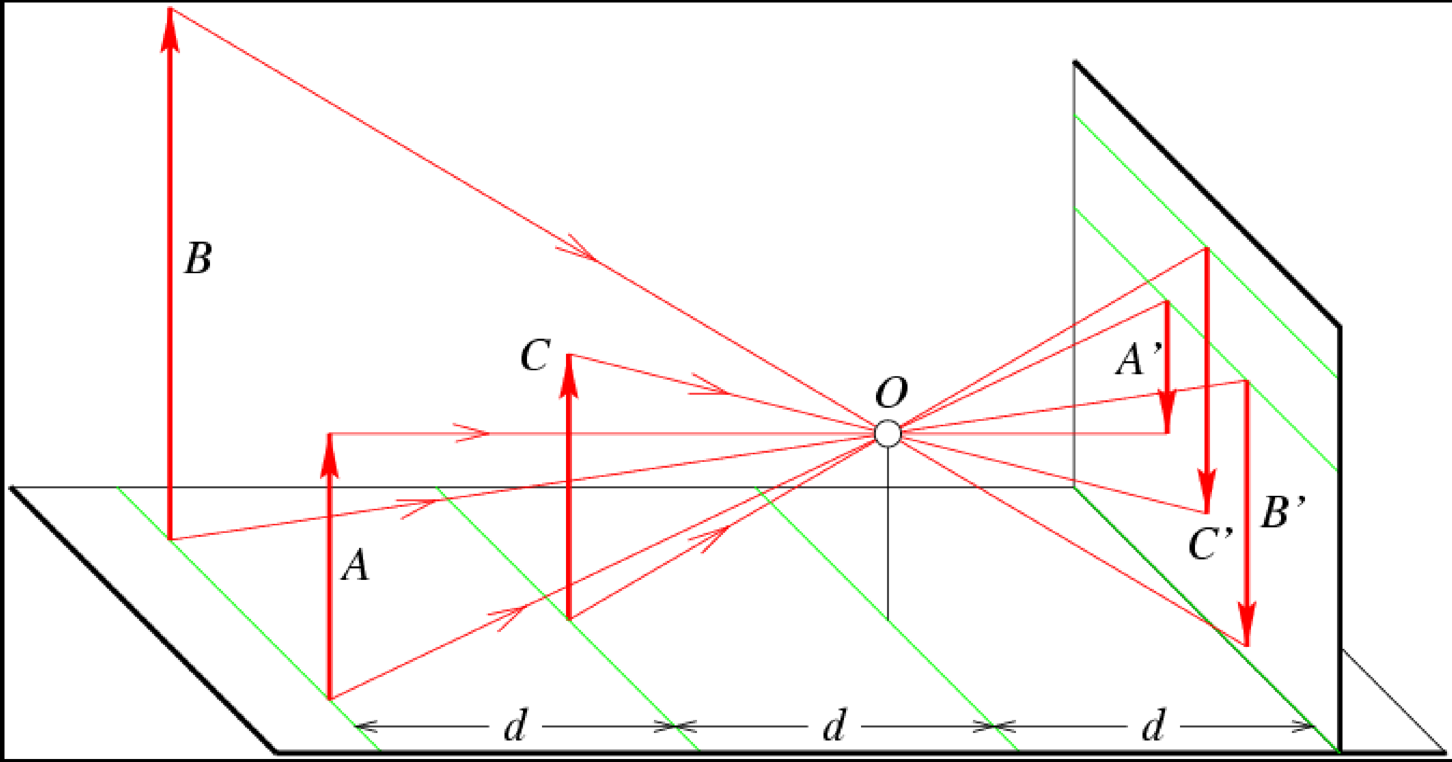
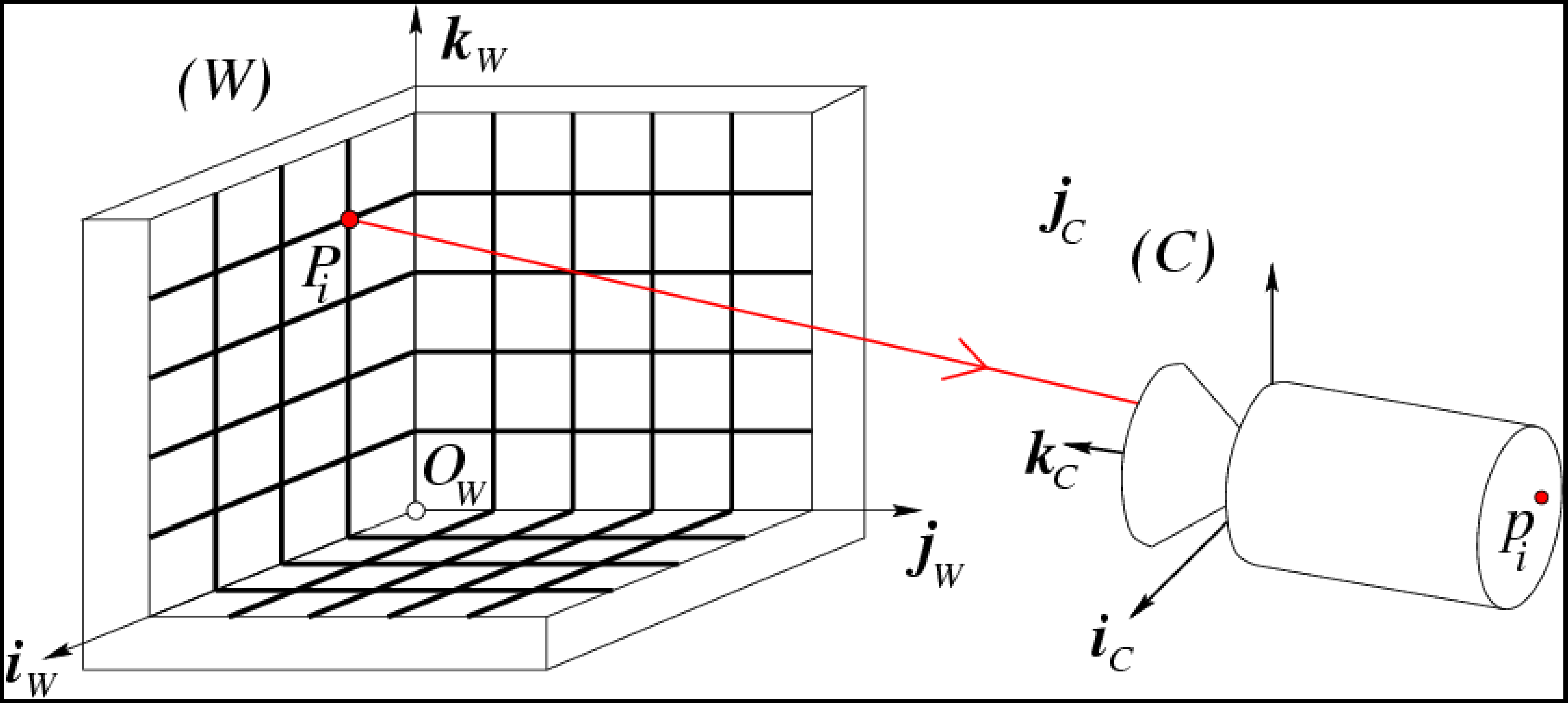


image
plane
pinhole
virtual
image

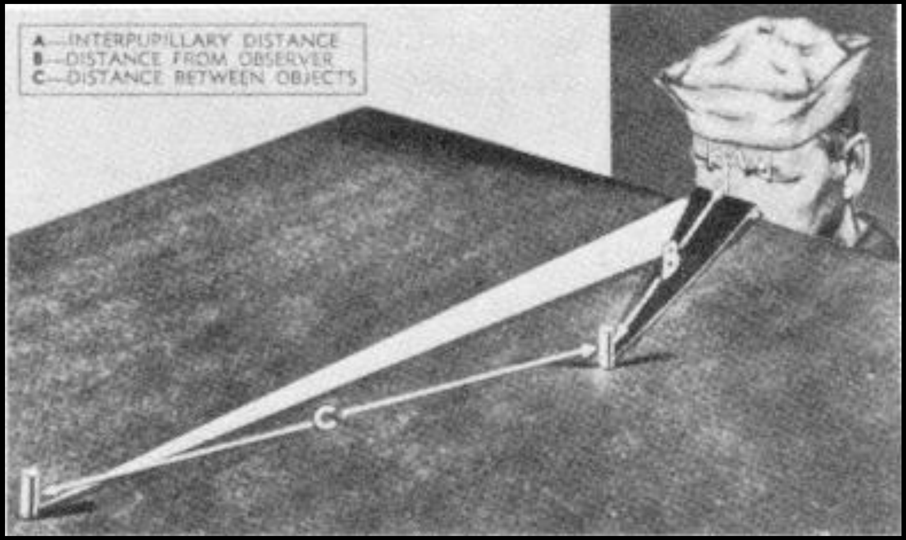

B
C
O
A'
A
B'
C'
d
d
d

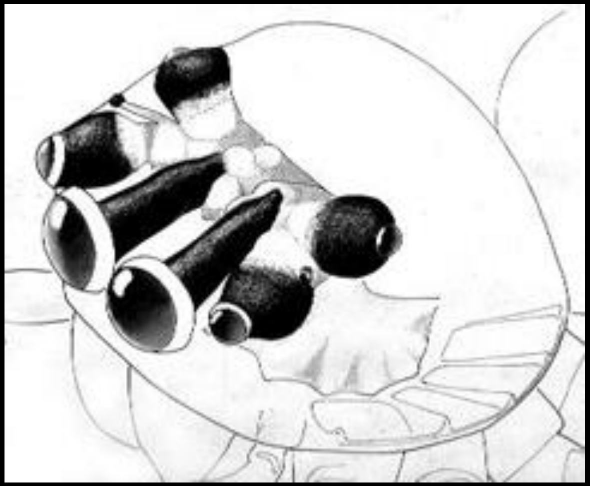
Quantitative Measurements and Calibration

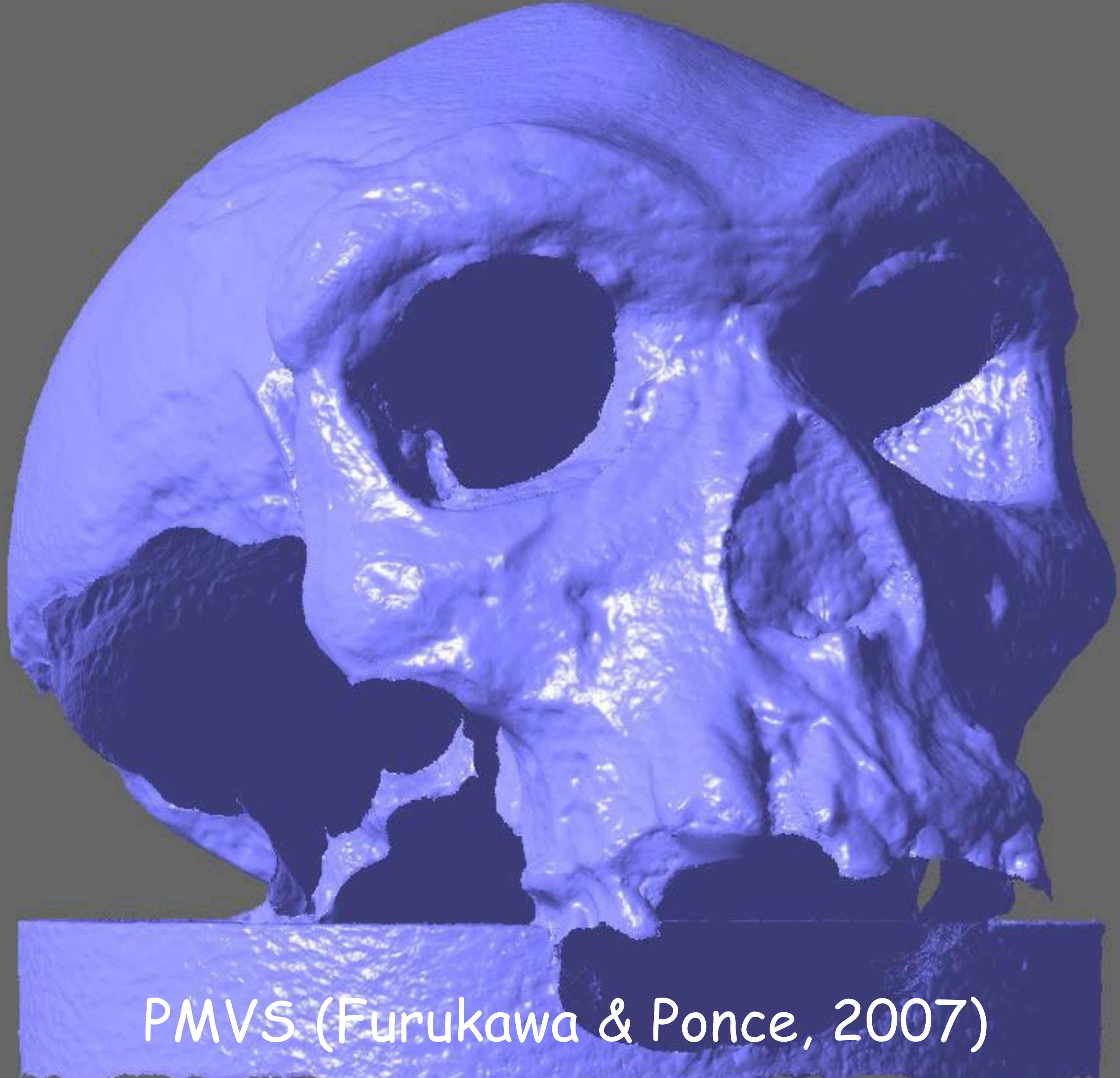


Euclidean Geometry

How do we perceive depth?





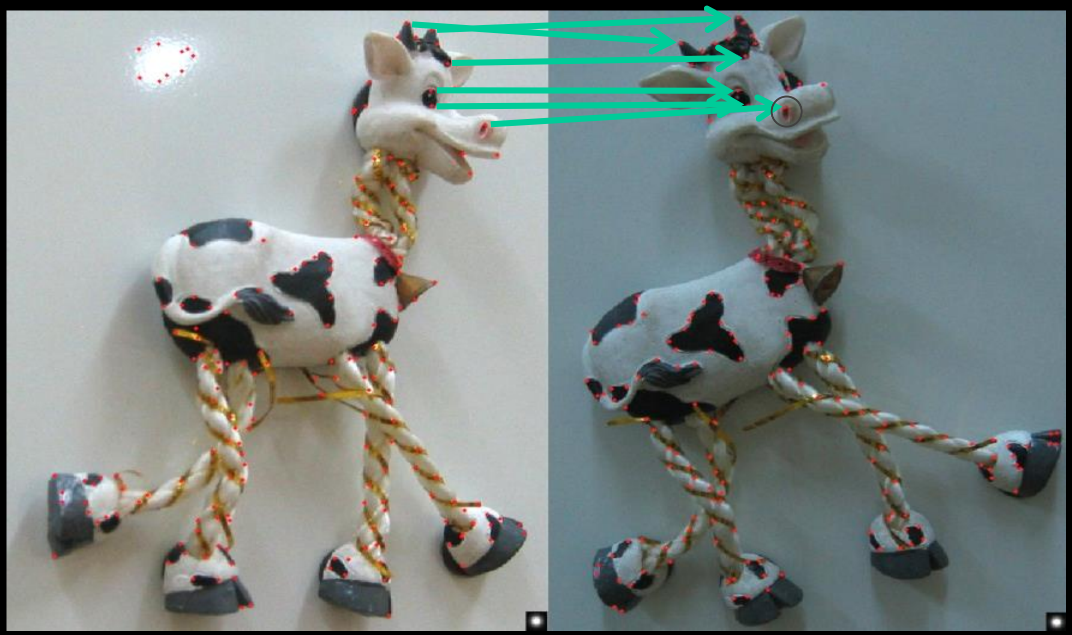


PMVS (Furukawa & Ponce, 2007)

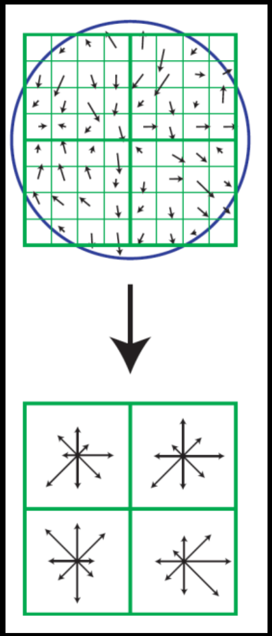
(Furukawa & Ponce, 2009)

Edge Detection

Interest points and local appearance models



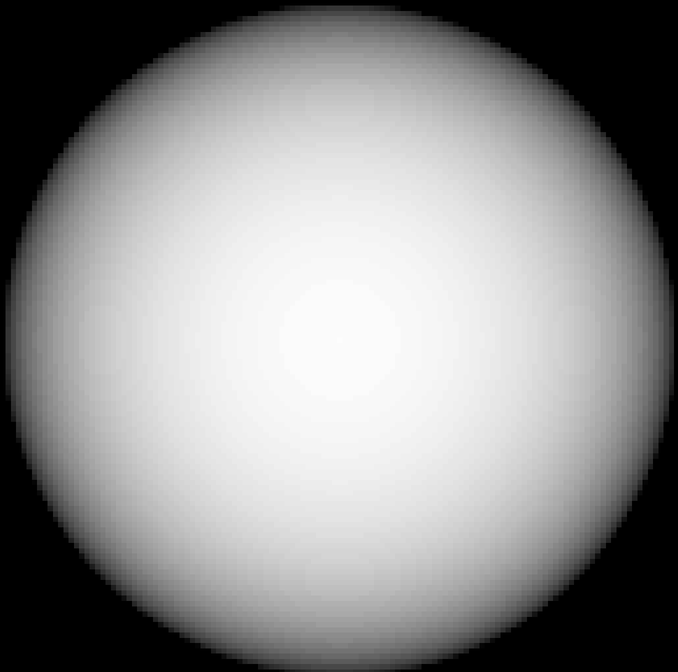
(Image courtesy of C. Schmid)

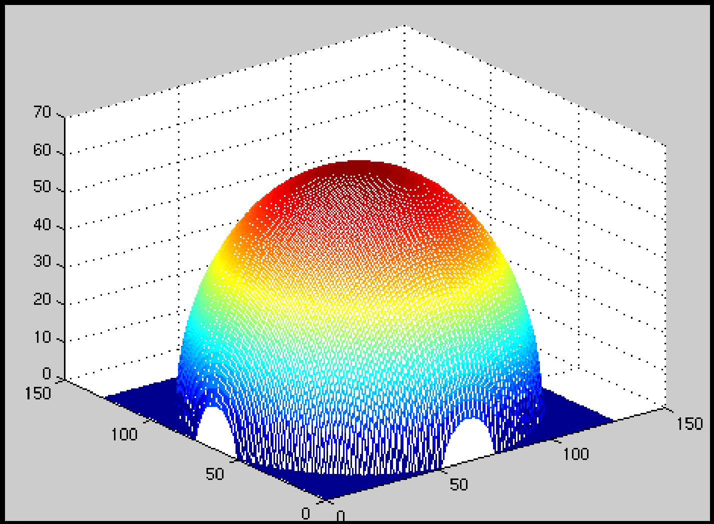


(Lowe 2004)

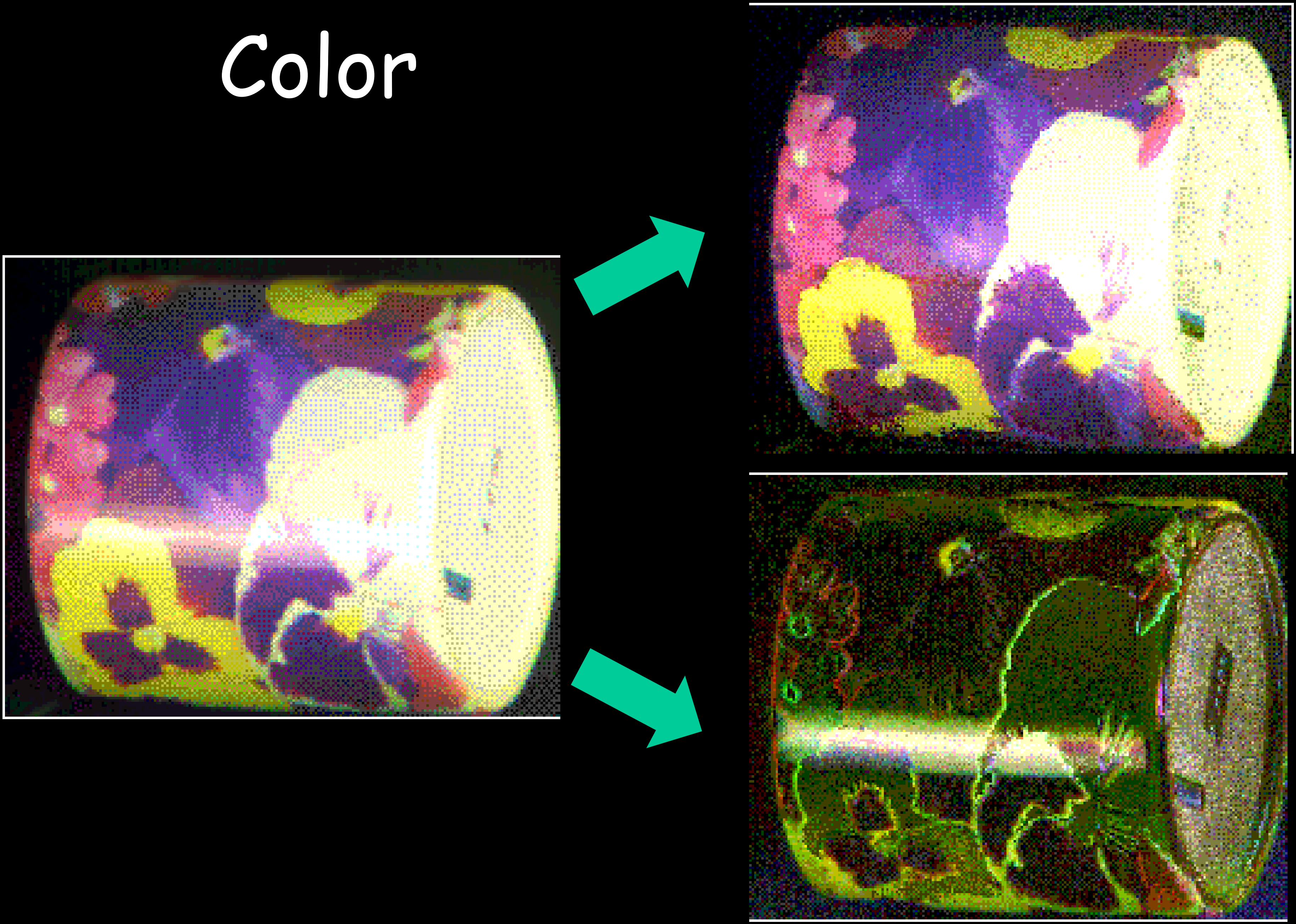
- Find features (interest points)
- Match them using local invariant descriptors (jets, SIFT)

Radiometry/Shading





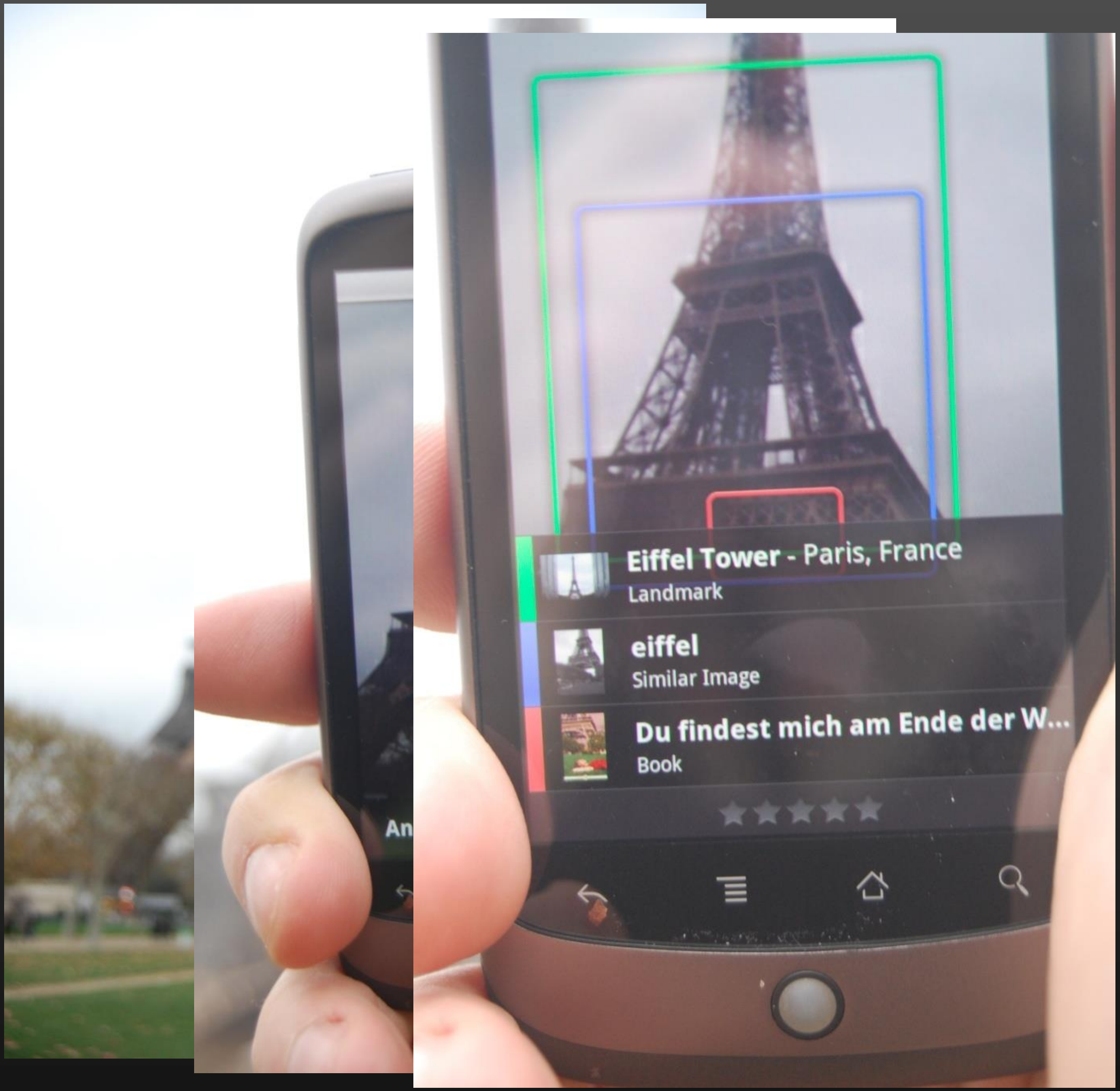
Color



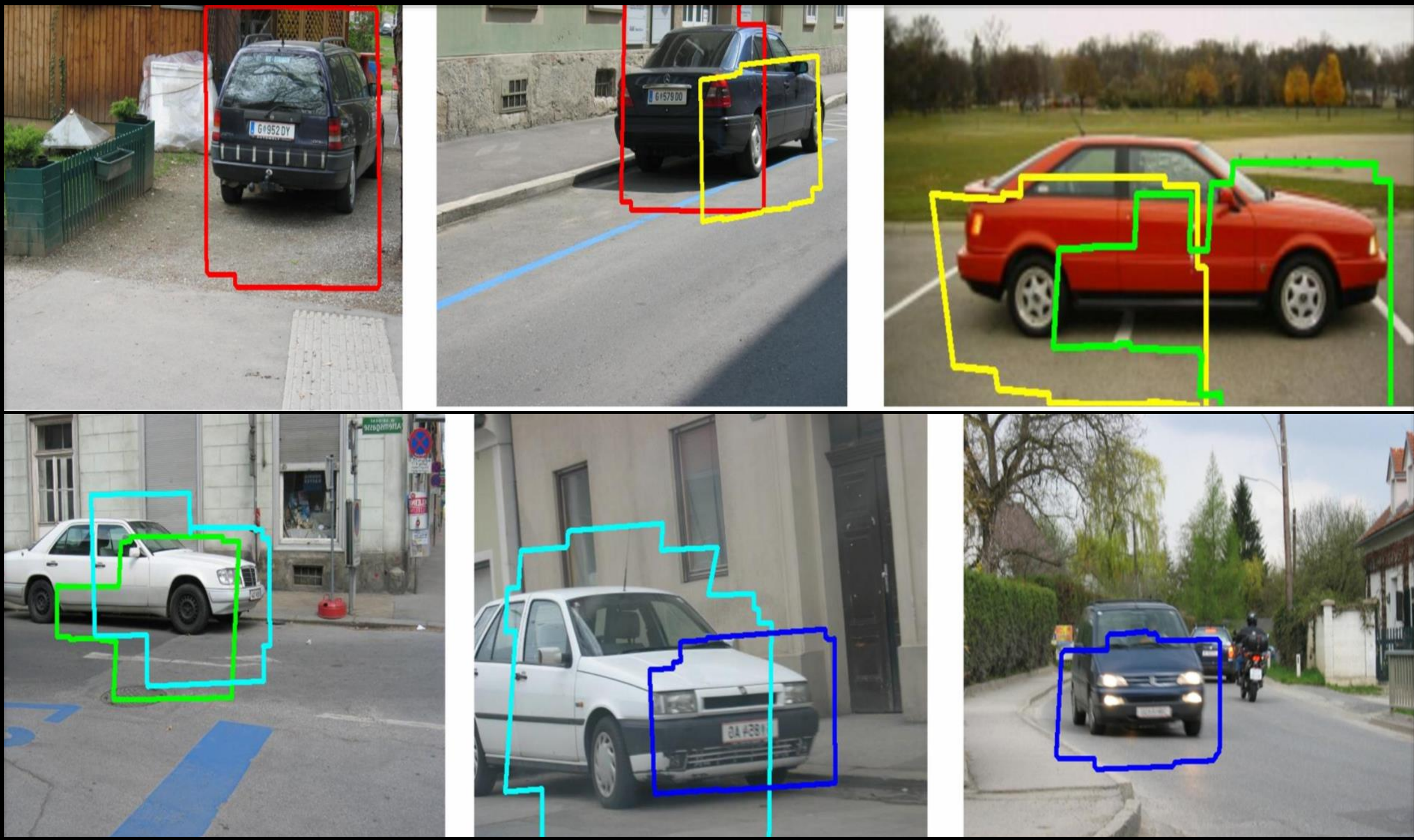
Segmentation

(Joulin, Bach, Ponce, CVPR'12)

Object instance recognition



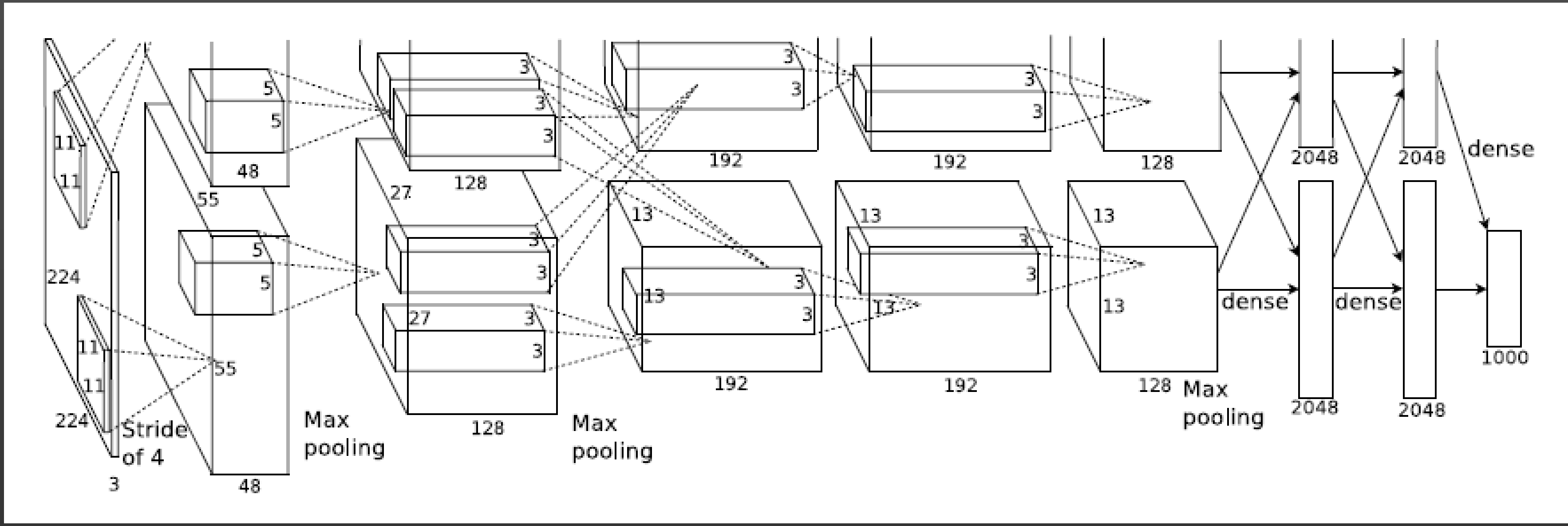
Category-level recognition



(Kushal et al., 2007)

Convolutional neural networks

[Krizhevsky et al. NIPS'12]



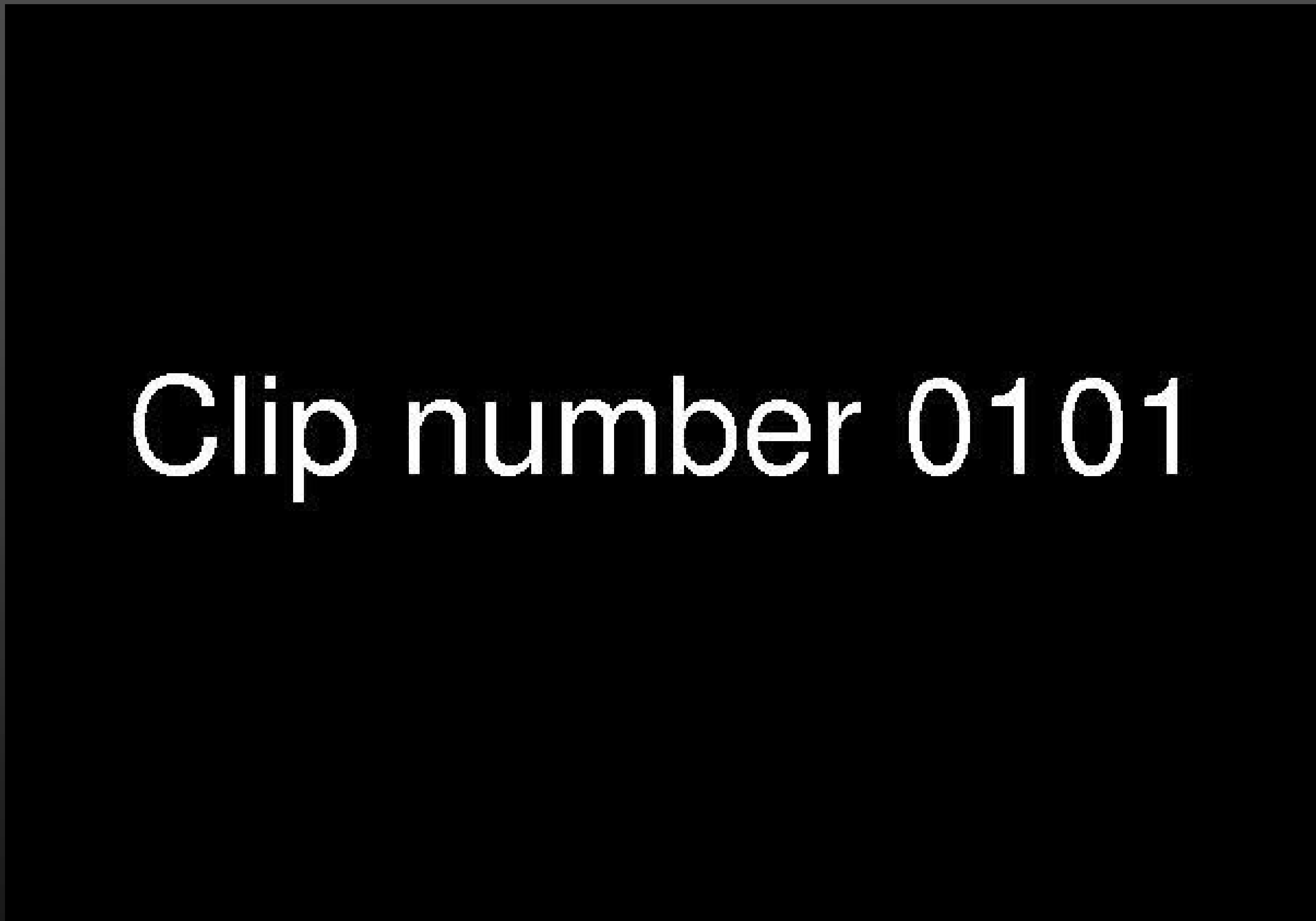
Convolutional Neural Networks:

- The main principles are known since LeCun'88
- Has 60M parameters and 650K neurons.
- Success is determined by (a) lots of labeled images and (b) fast GPU implementation. Neither (a) nor (b) have not been available until very recently.

Some results

[Oquab, Bottou, Laptev, Sivic, CVPR 2014]

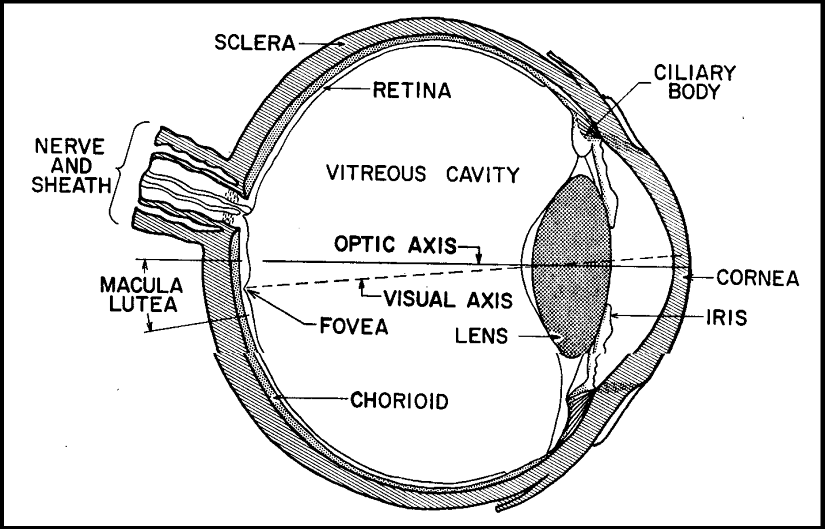
Weakly-supervised video interpretation



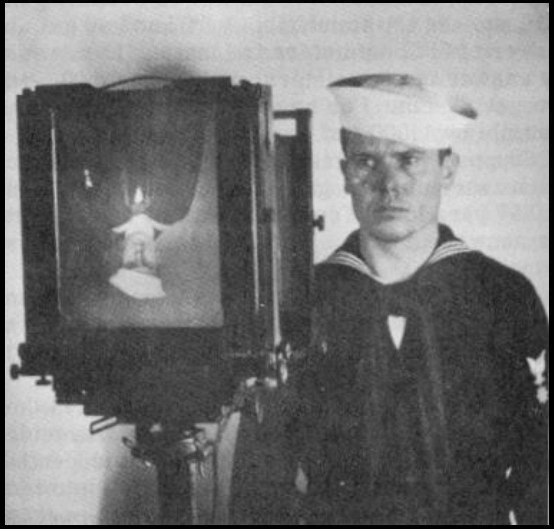
(Bojanowski et al., 2014)

Camera geometry and calibration

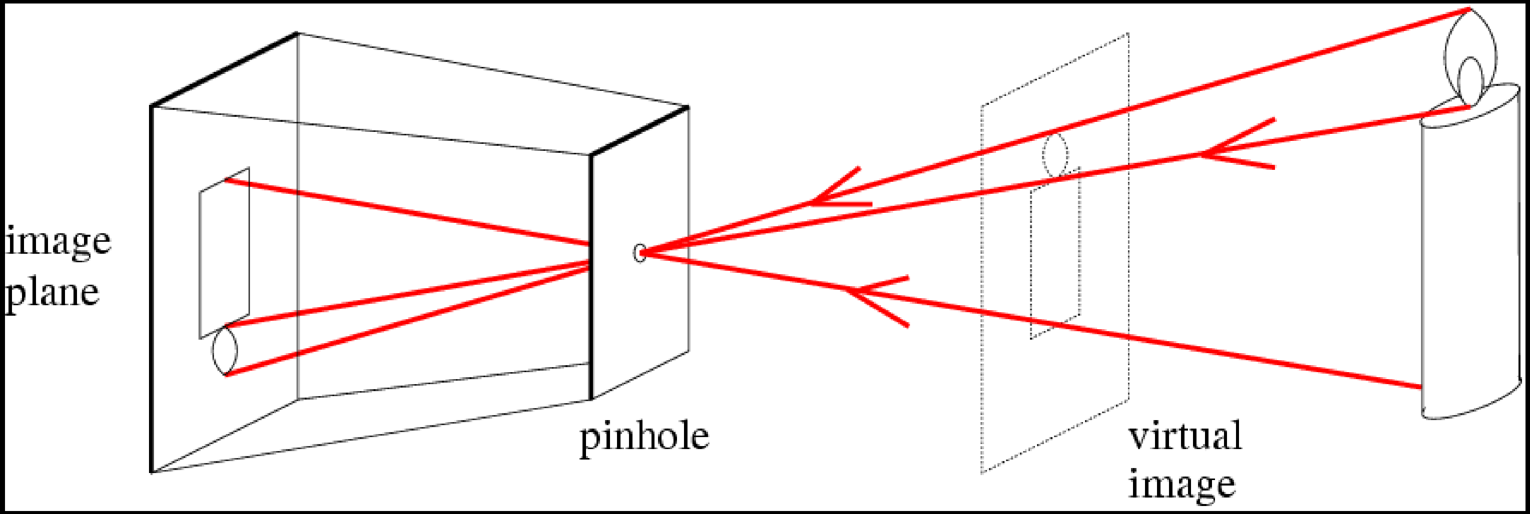
- Pinhole perspective projection
- Orthographic and weak-perspective models
- Non-standard models
- *A detour through sensing country*



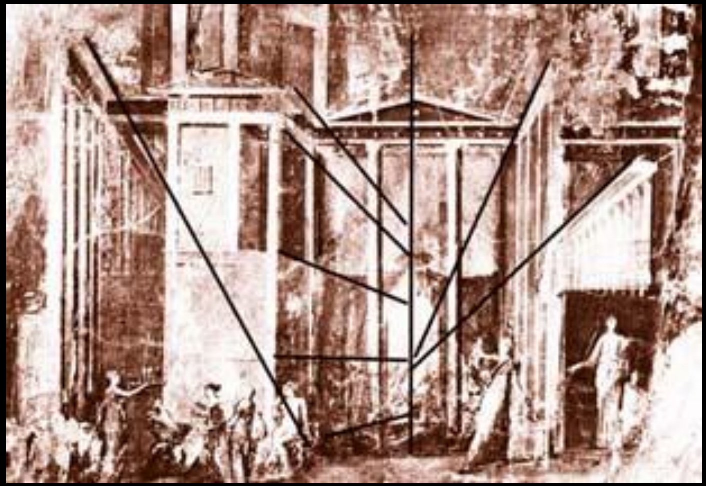
Animal eye: a looonnng time ago.



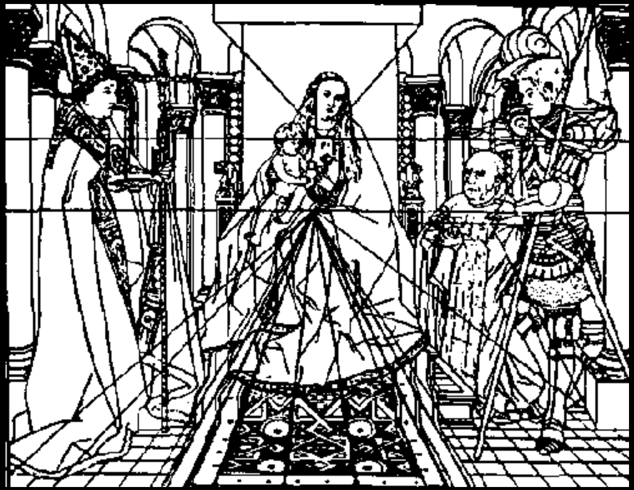
Photographic camera: Niepce, 1816.



Pinhole perspective projection: Brunelleschi, XVth Century.
Camera obscura: XVIth Century.



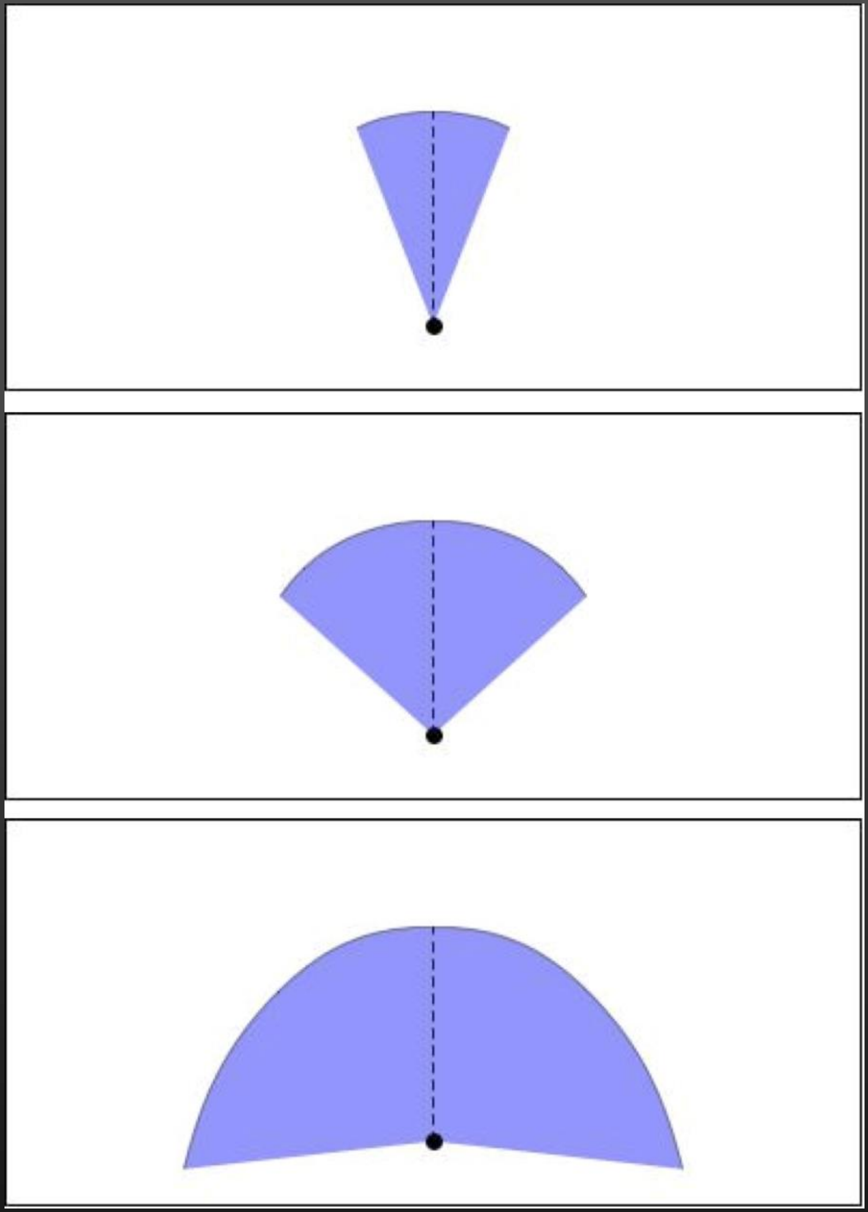
Pompei painting, 2000 years ago

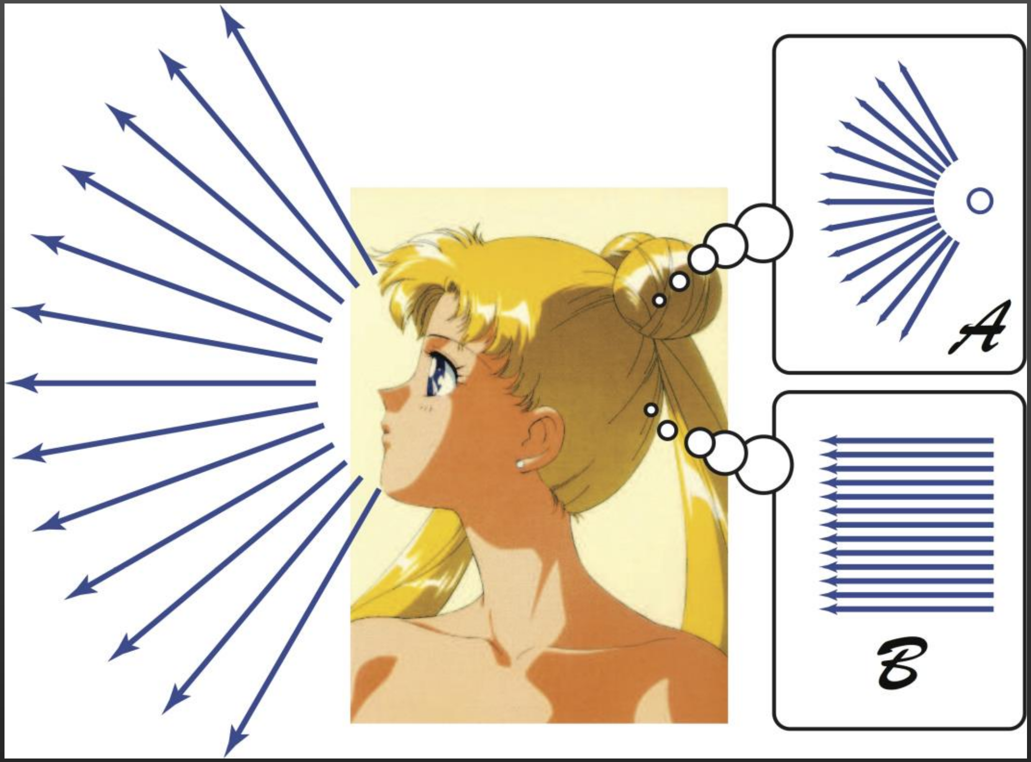


Van Eyk, XIVth Century

Brunelleschi, 1415

Massaccio's Trinity, 1425

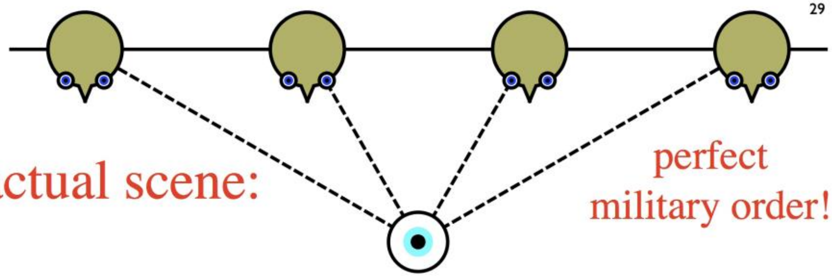




Most people don't experience the divergence of visual rays in a veridical manner. This is fine. [Koenderink]

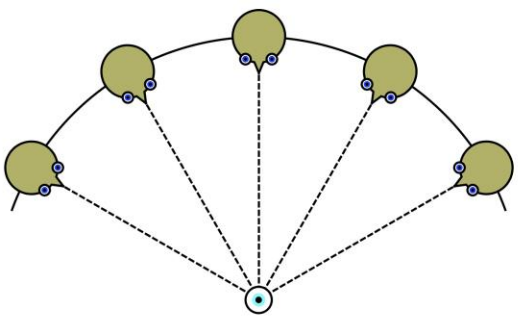
Ferdinand Hodler

actual scene:



perfect military order!

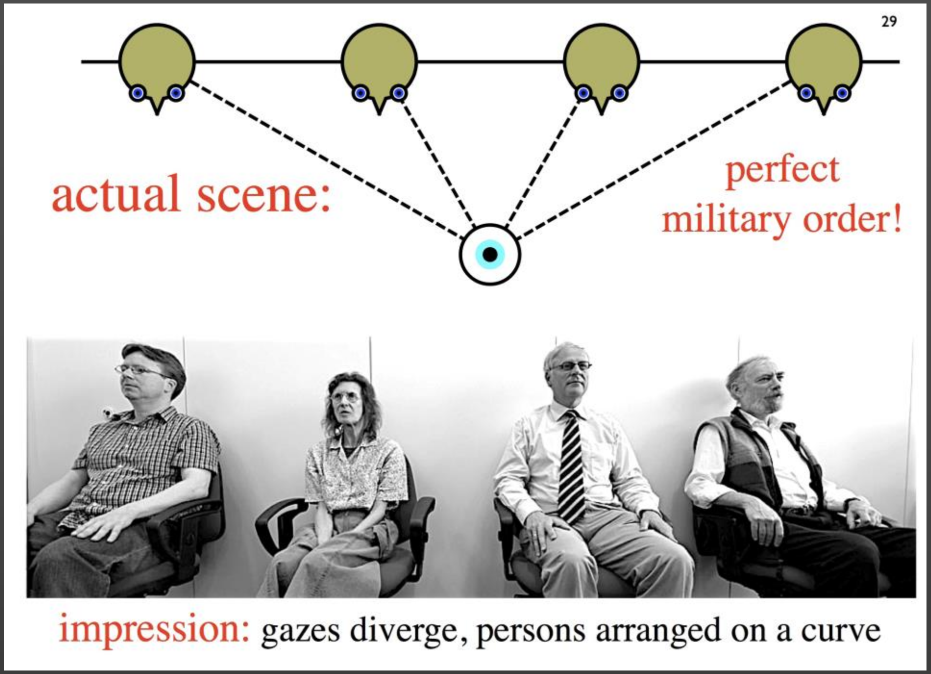
impression: gazes diverge, persons arranged on a curve



actual scene:

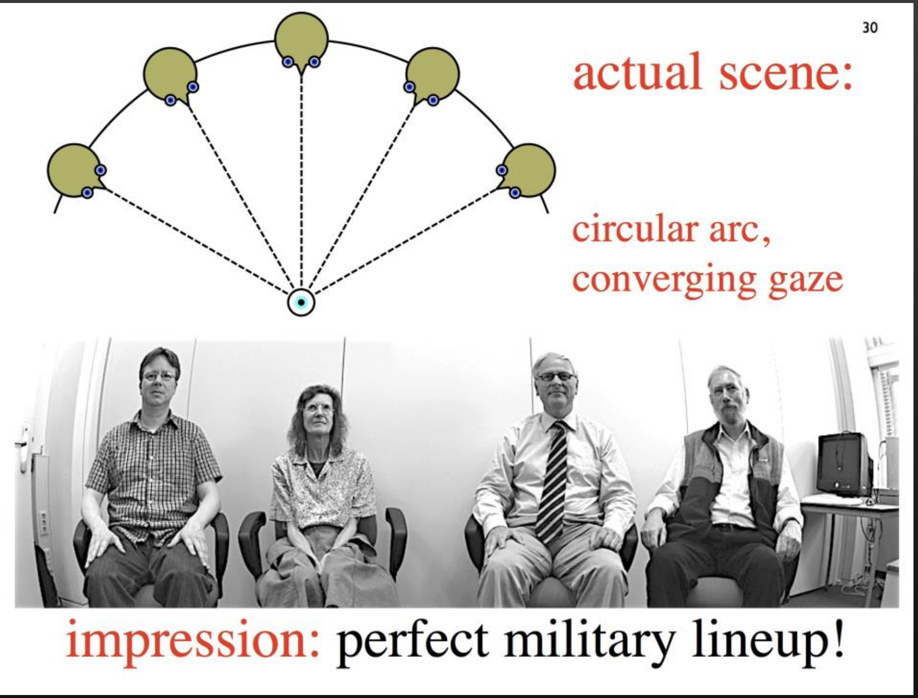
circular arc, converging gaze

impression: perfect military lineup!



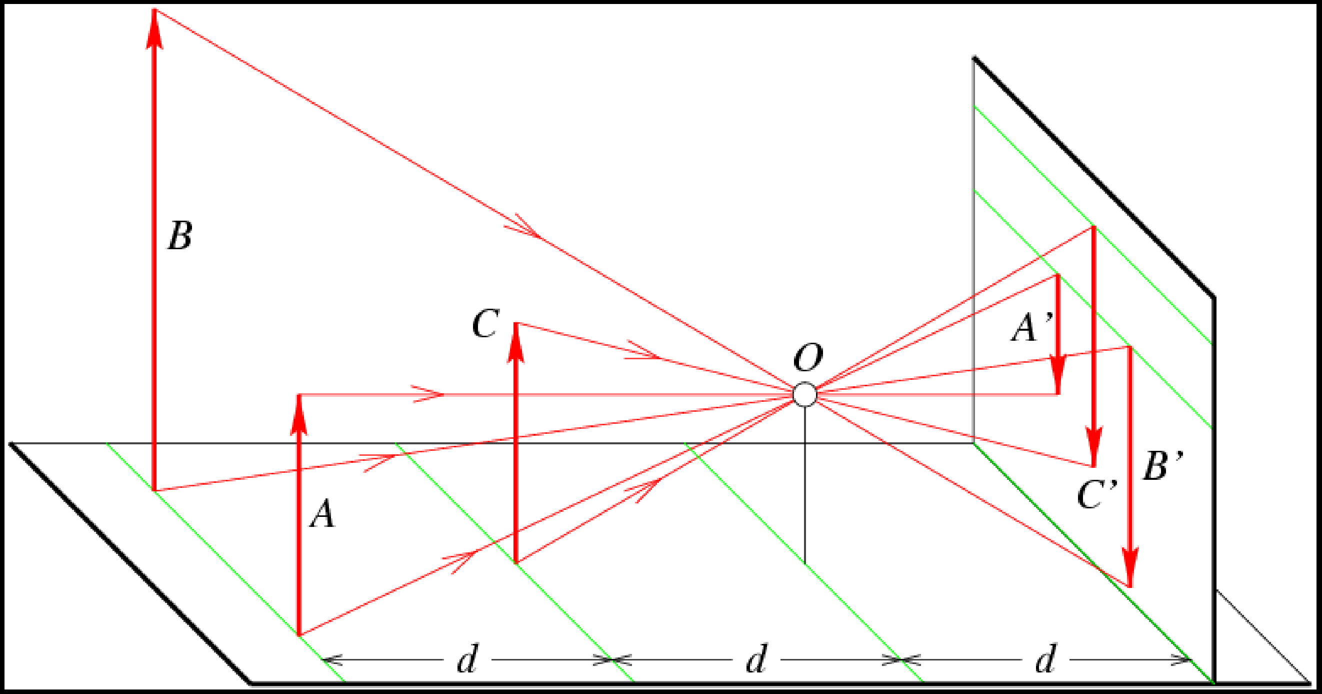
29
actual scene:
perfect
military order!
impression: gazes diverge, persons arranged on a curve

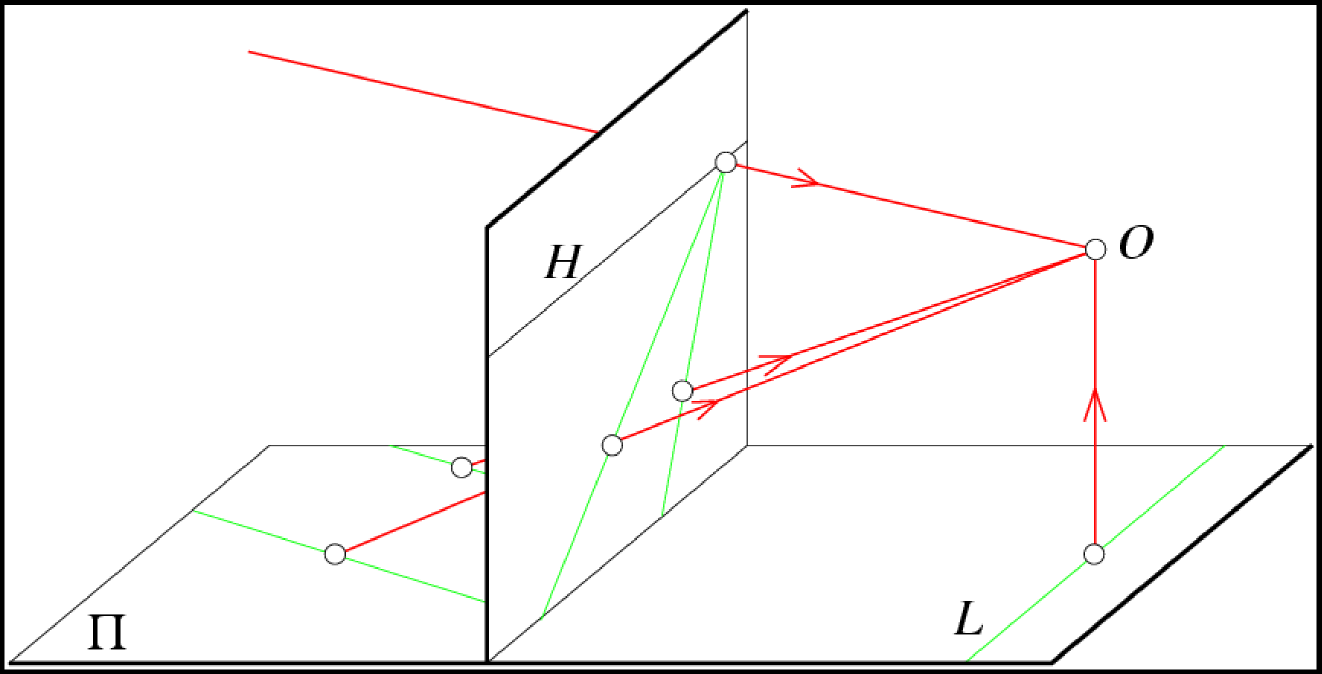
"YOUR COUNTRY NEEDS
YOU"



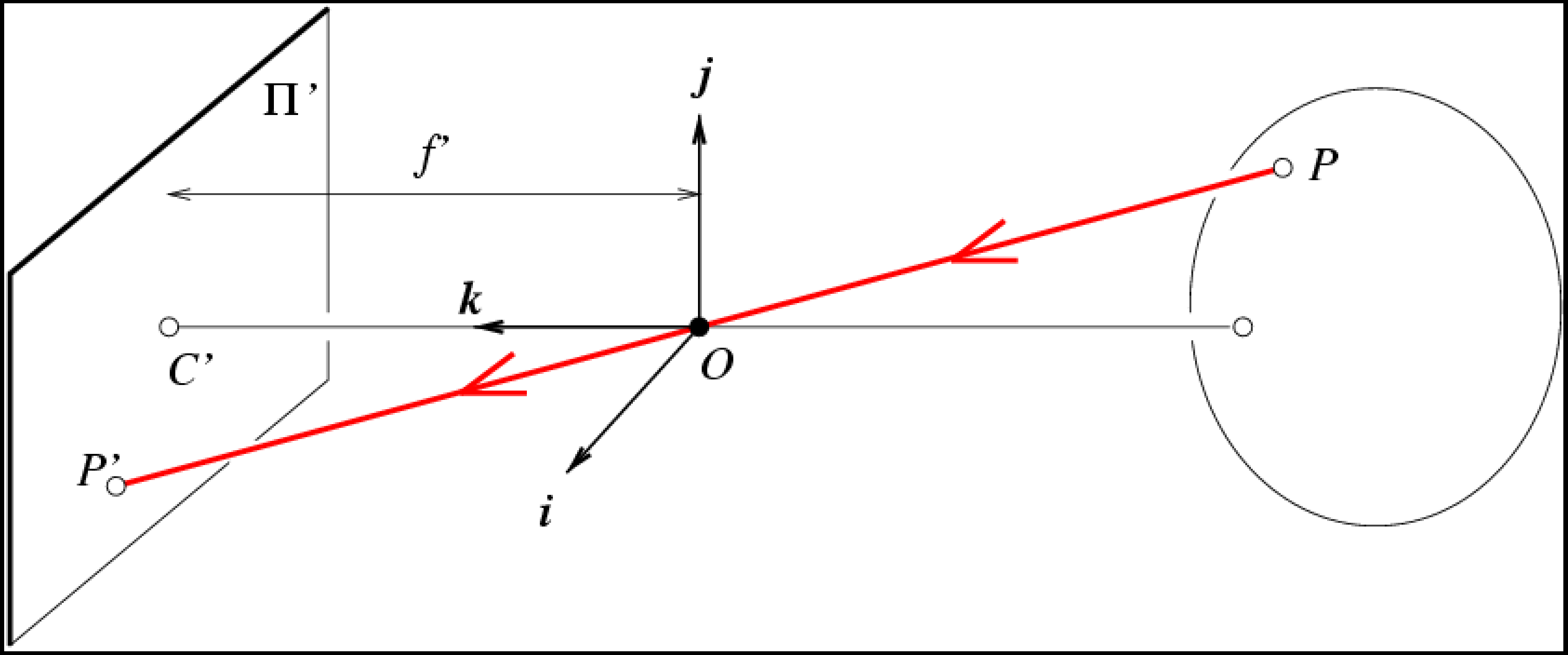
30
actual scene:
circular arc,
converging gaze
impression: perfect military lineup!

"YOUR COUNTRY NEEDS
YOU"


B
C
O
A'
A
B'
C'
d
d
d


H
O
Π
L

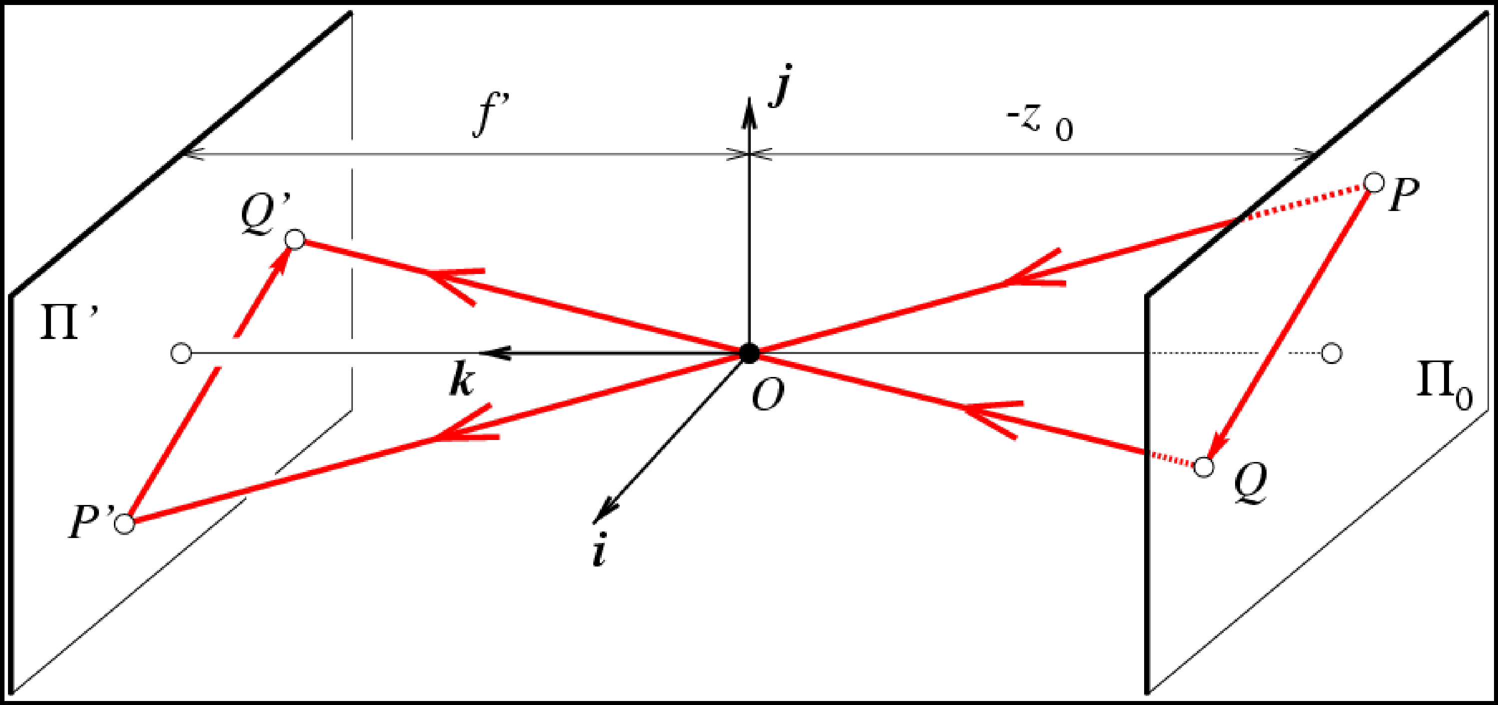
Pinhole Perspective Equation



$$\begin{cases} x' = f'\dfrac{x}{z} \\ y' = f'\dfrac{y}{z} \end{cases}$$

NOTE: z is always negative..

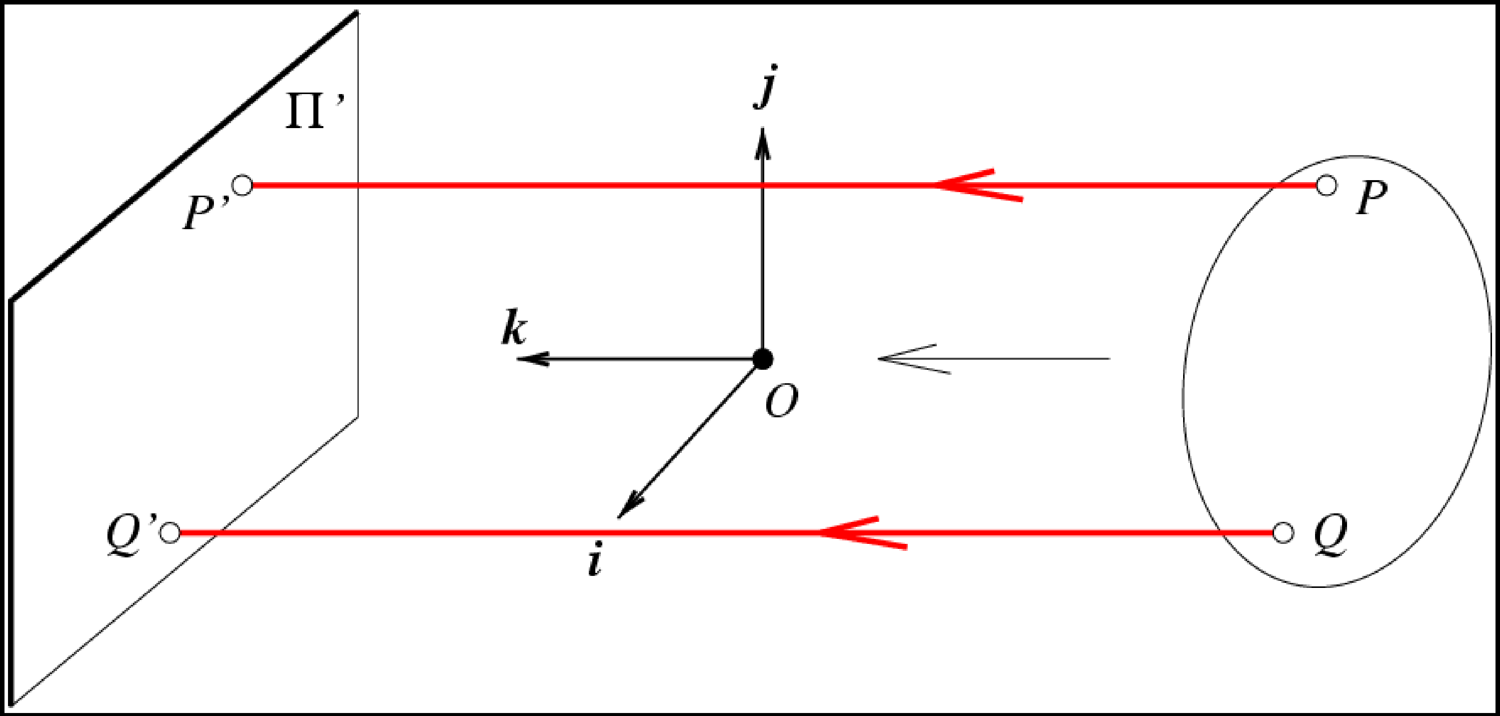
Affine projection models: Weak perspective projection



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

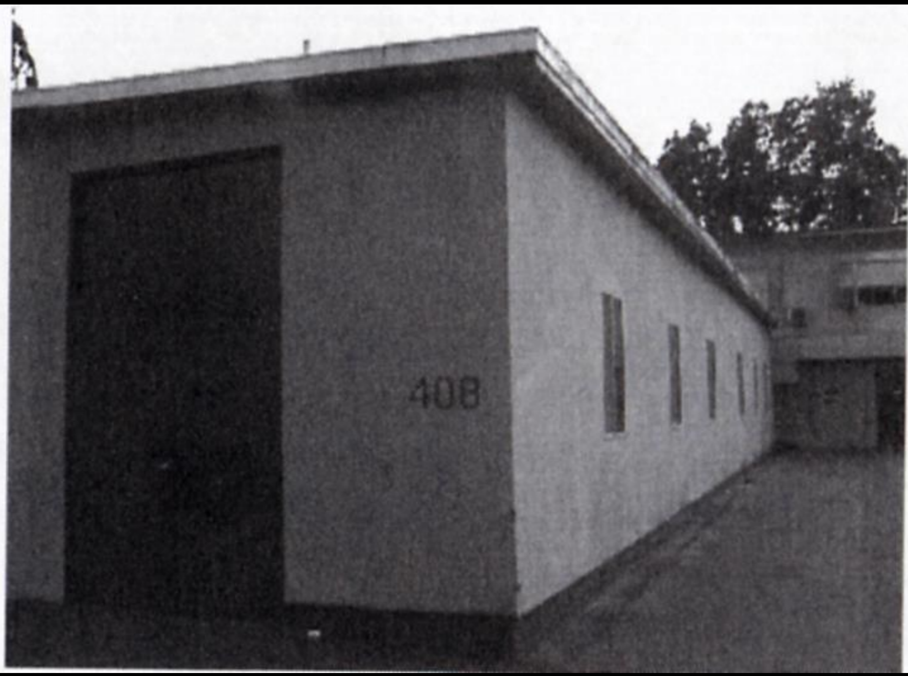
When the scene relief is small compared its distance from the Camera, *m* can be taken constant: weak perspective projection.

Affine projection models: Orthographic projection



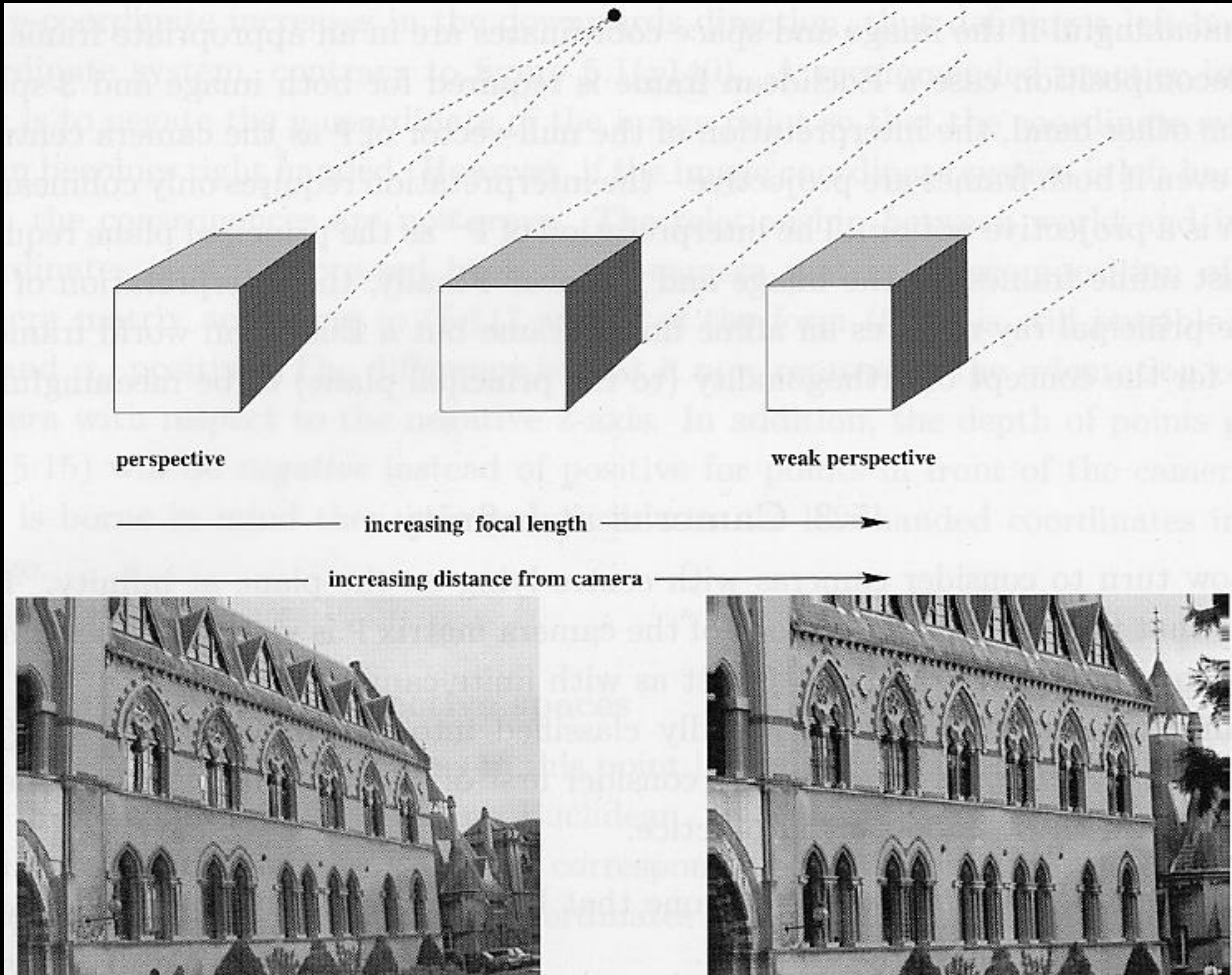
$$\begin{cases} x' = x \\ y' = y \end{cases}$$

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



Strong perspective:

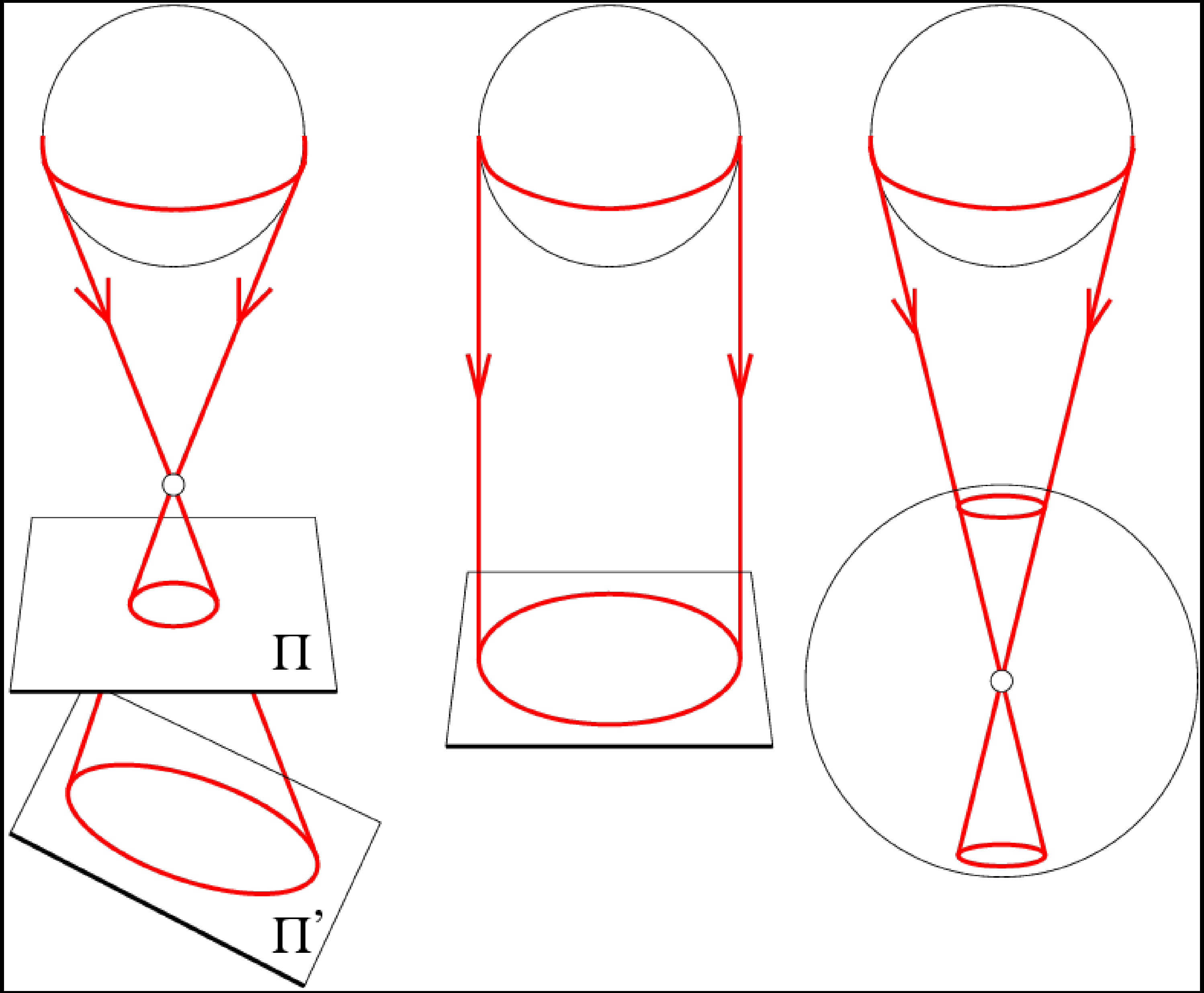
- Angles are not preserved
- The projections of parallel lines intersect at one point



From Zisserman & Hartley

Strong perspective:
Angles are not preserved
The projections of parallel lines intersect at one point

Weak perspective:
Angles are better preserved
The projections of parallel lines are (almost) parallel

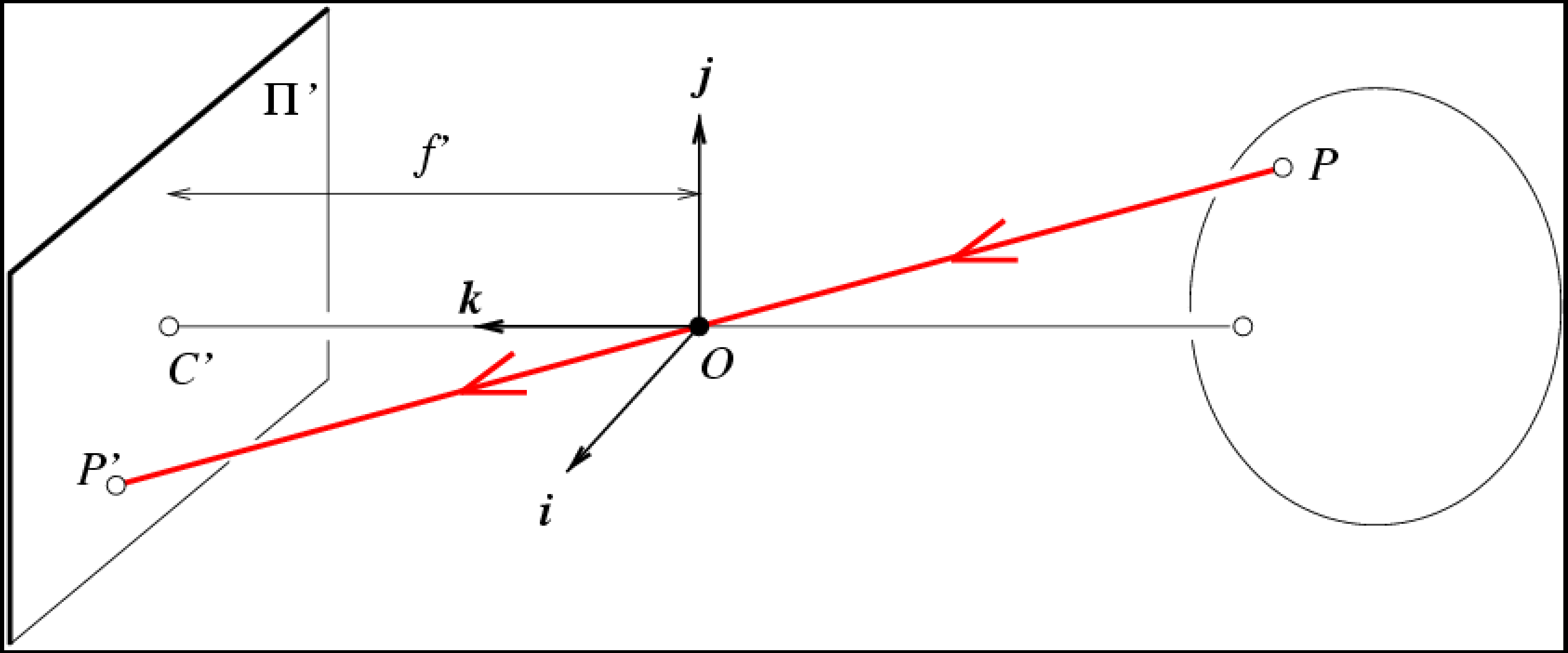


Planar pinhole perspective

Orthographic projection

Spherical pinhole perspective

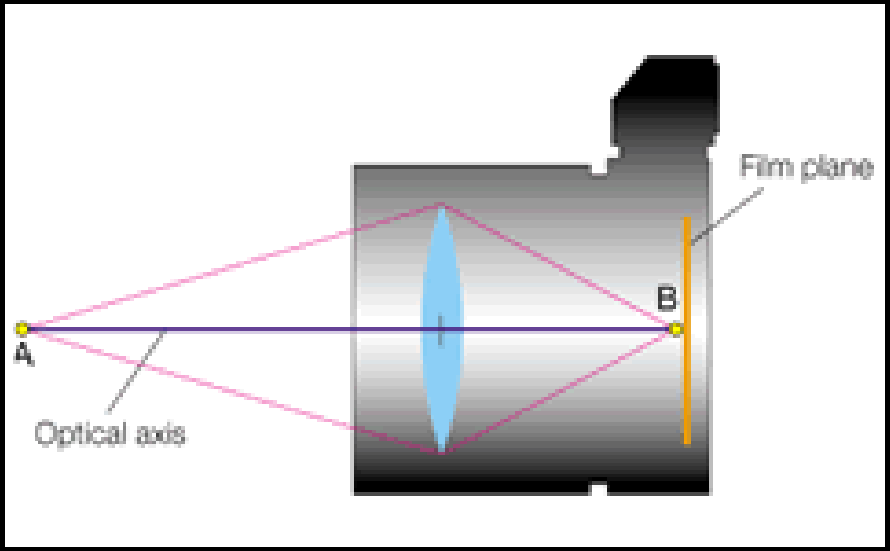
Pinhole Perspective Equation



$$\begin{cases} x' = f' \dfrac{x}{z} \\ y' = f' \dfrac{y}{z} \end{cases}$$

NOTE: z is always negative..

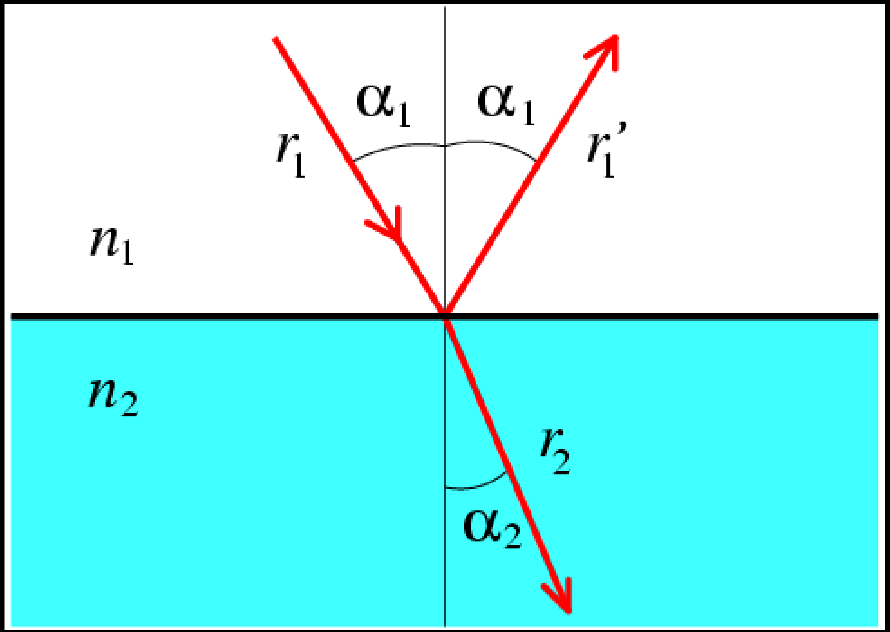
Lenses



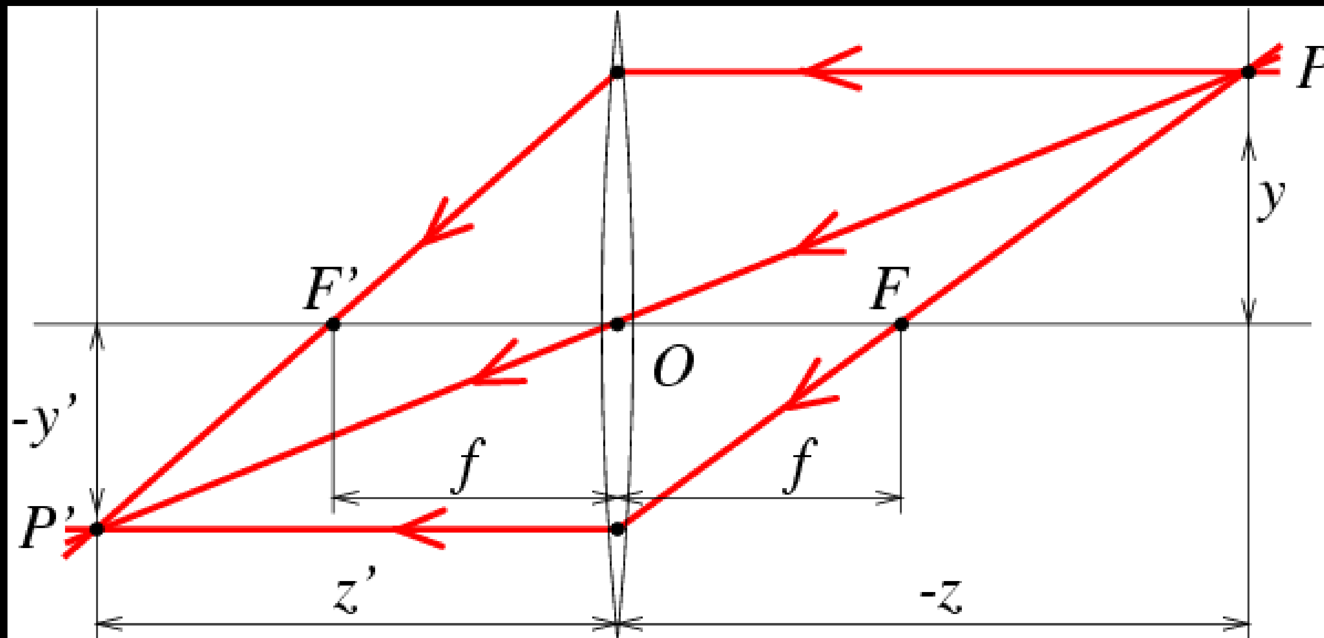
Snell's law

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$$

(Descartes' law
for Frenchies)

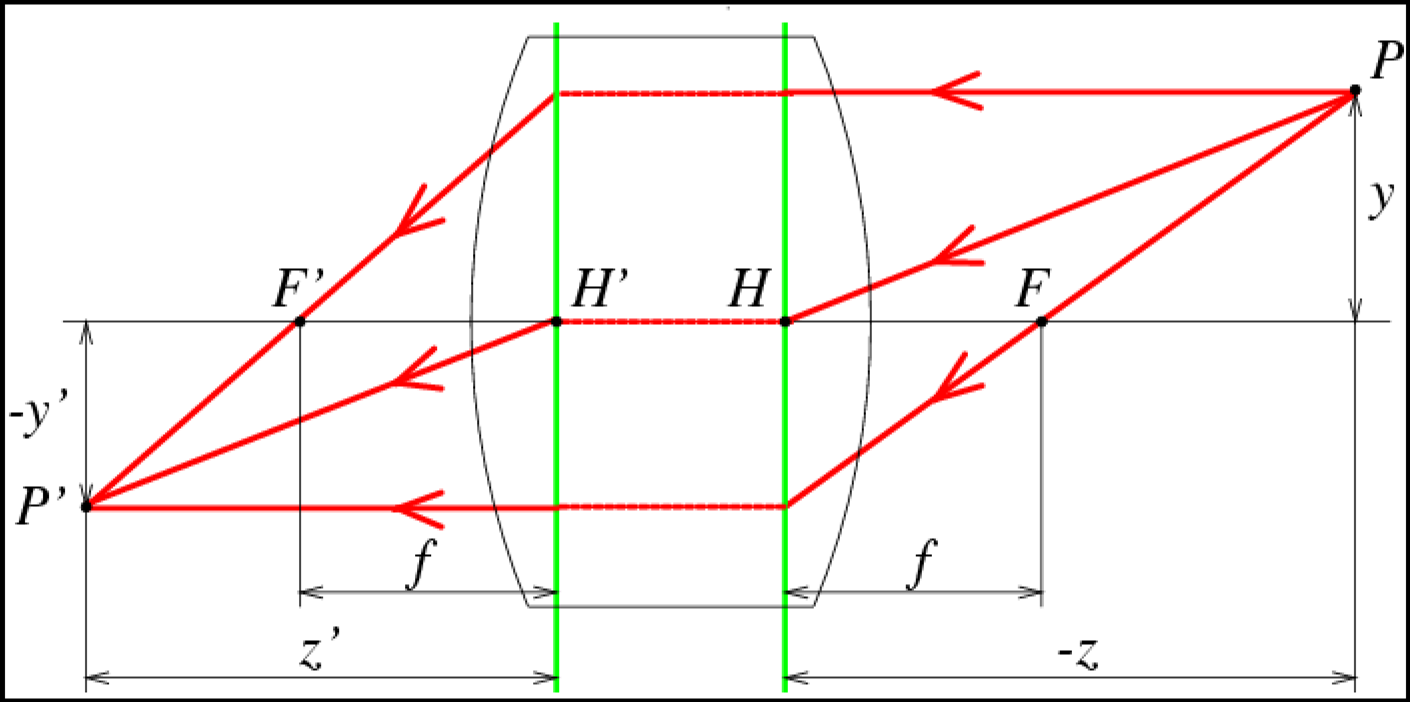


Thin Lenses (including paraxial approximation)

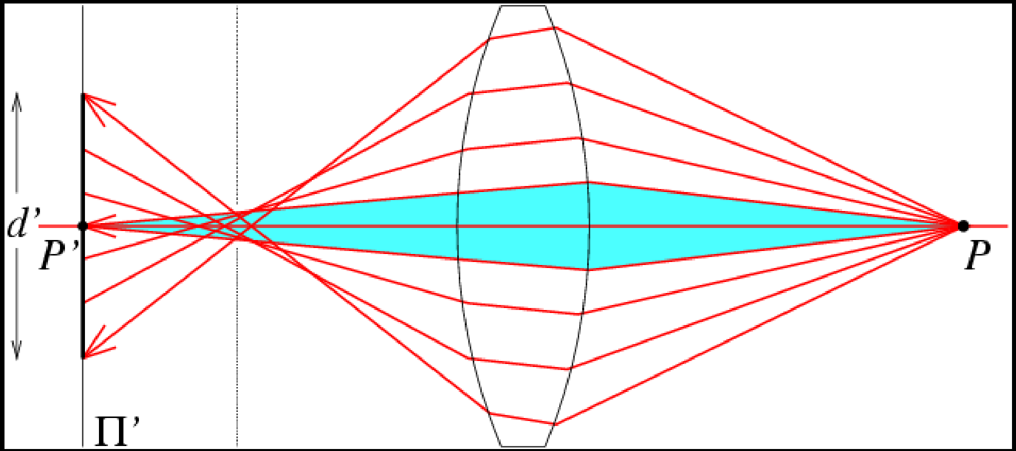


$$\begin{cases} x' = z'\dfrac{x}{z} \\ y' = z'\dfrac{y}{z} \end{cases}$$ where $$\boxed{\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}}$$ and $$\boxed{f = \frac{R}{2(n-1)}}$$

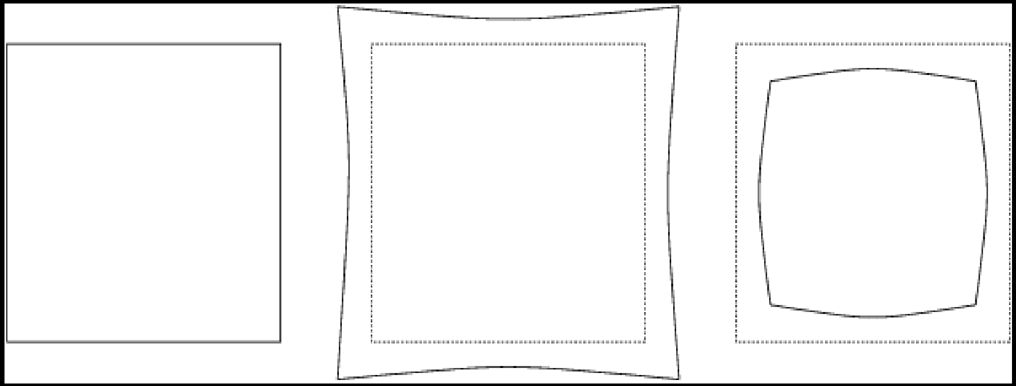
Thick Lenses



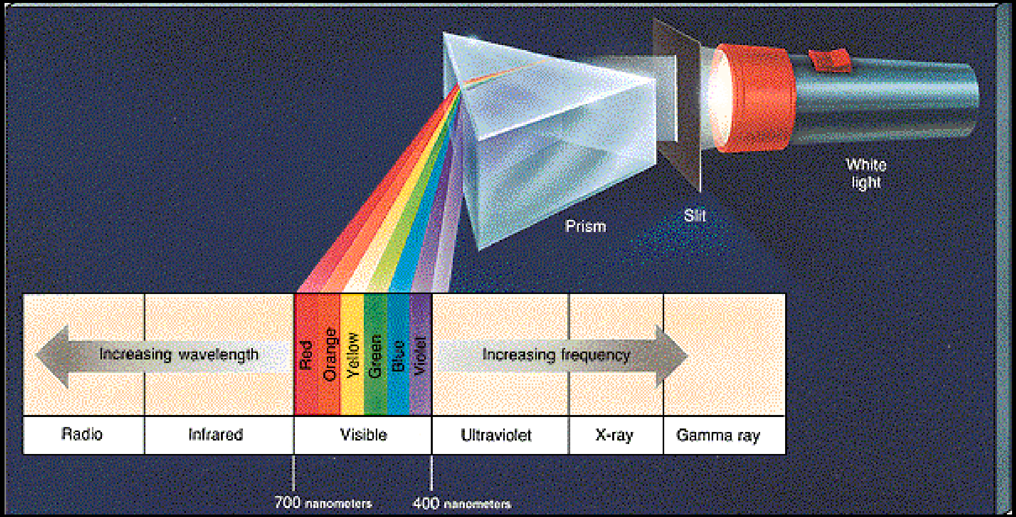
Spherical Aberration



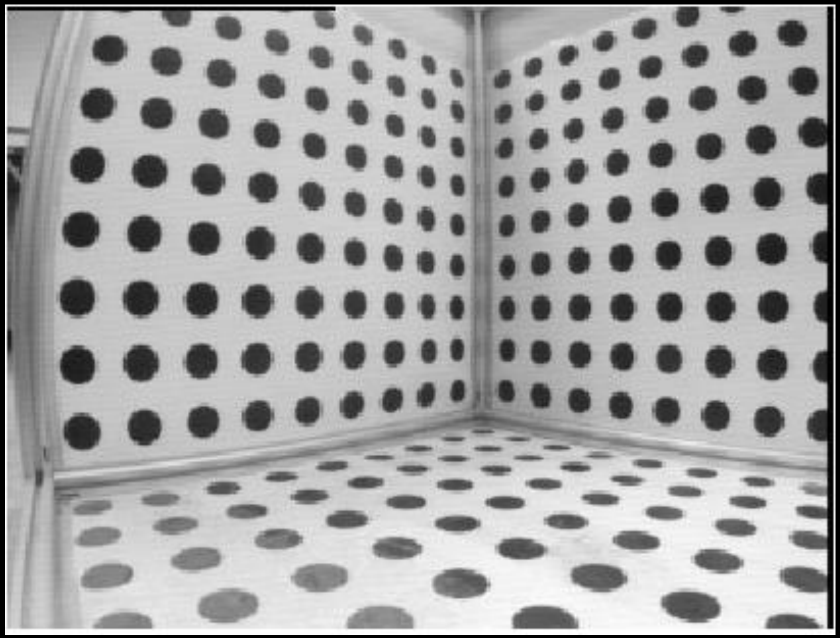
Distortion

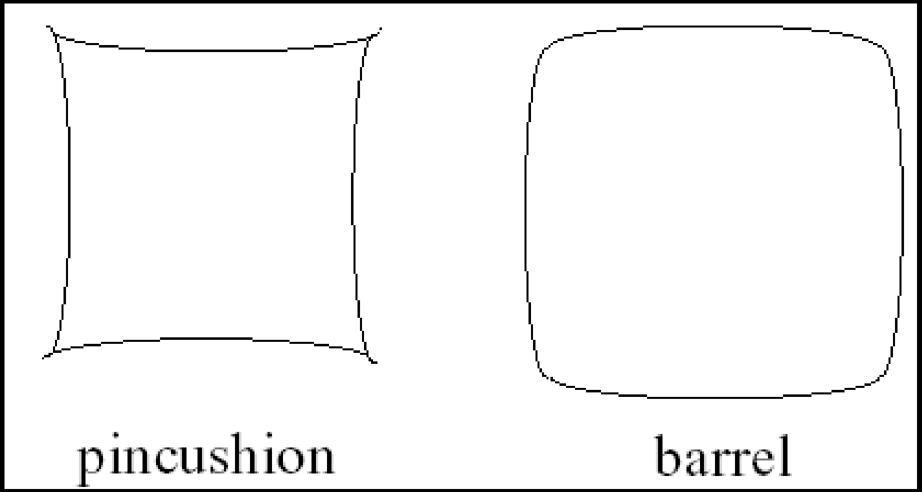


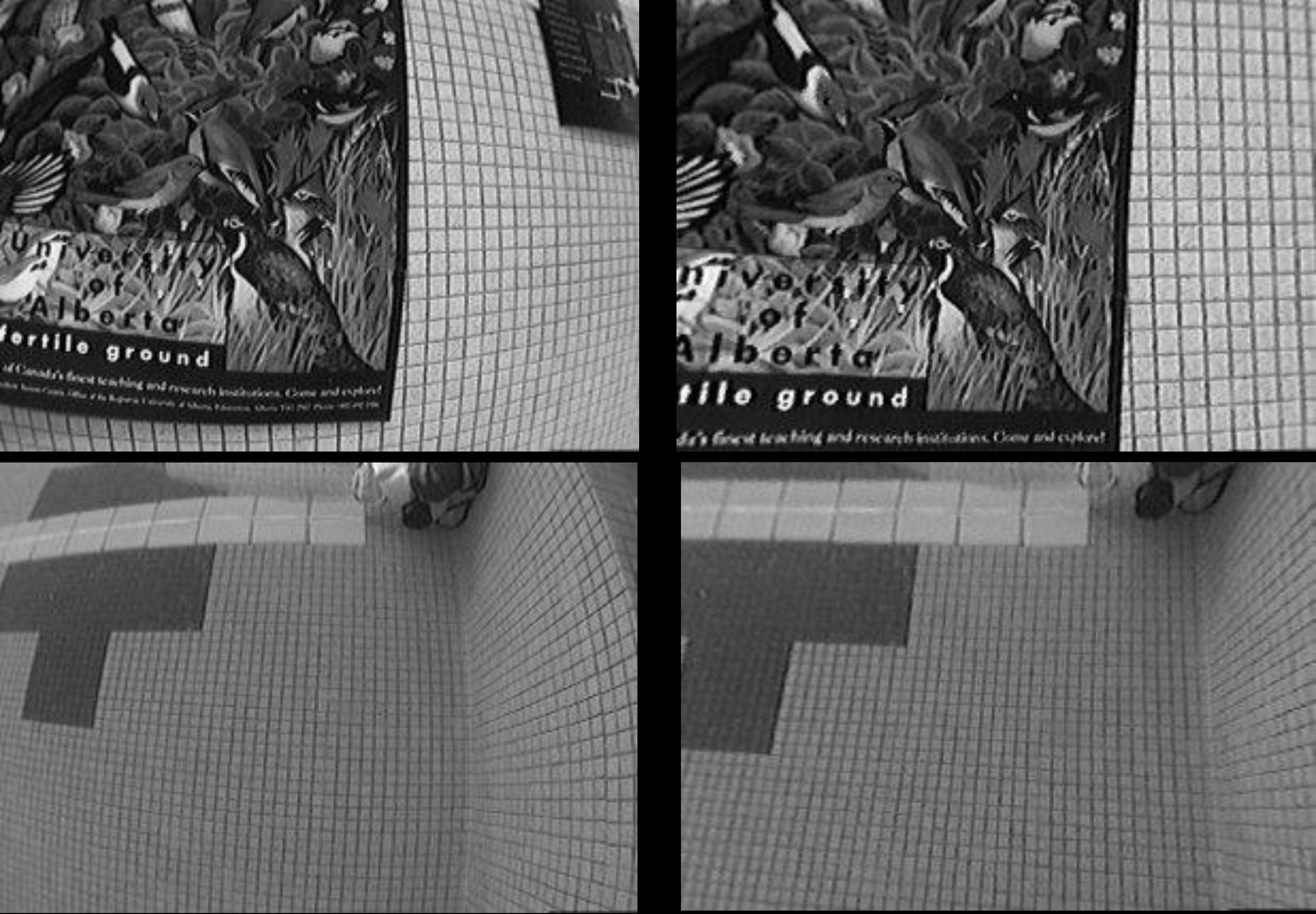
Chromatic Aberration



Geometric Distortion

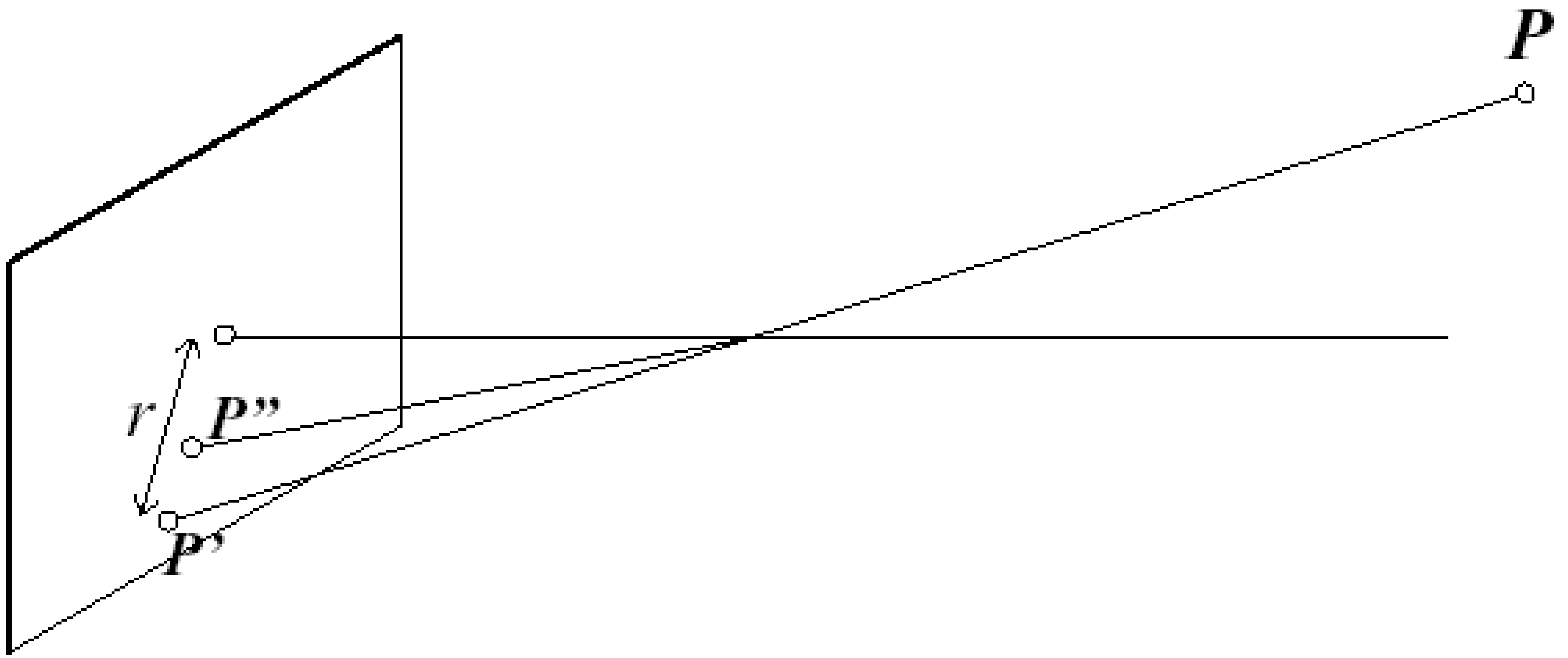






Rectification

Radial Distortion Model



Ideal:

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

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

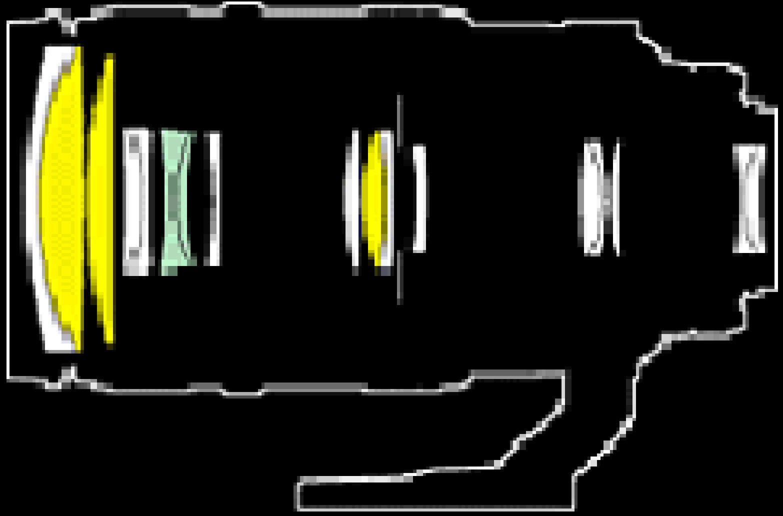
Distorted:

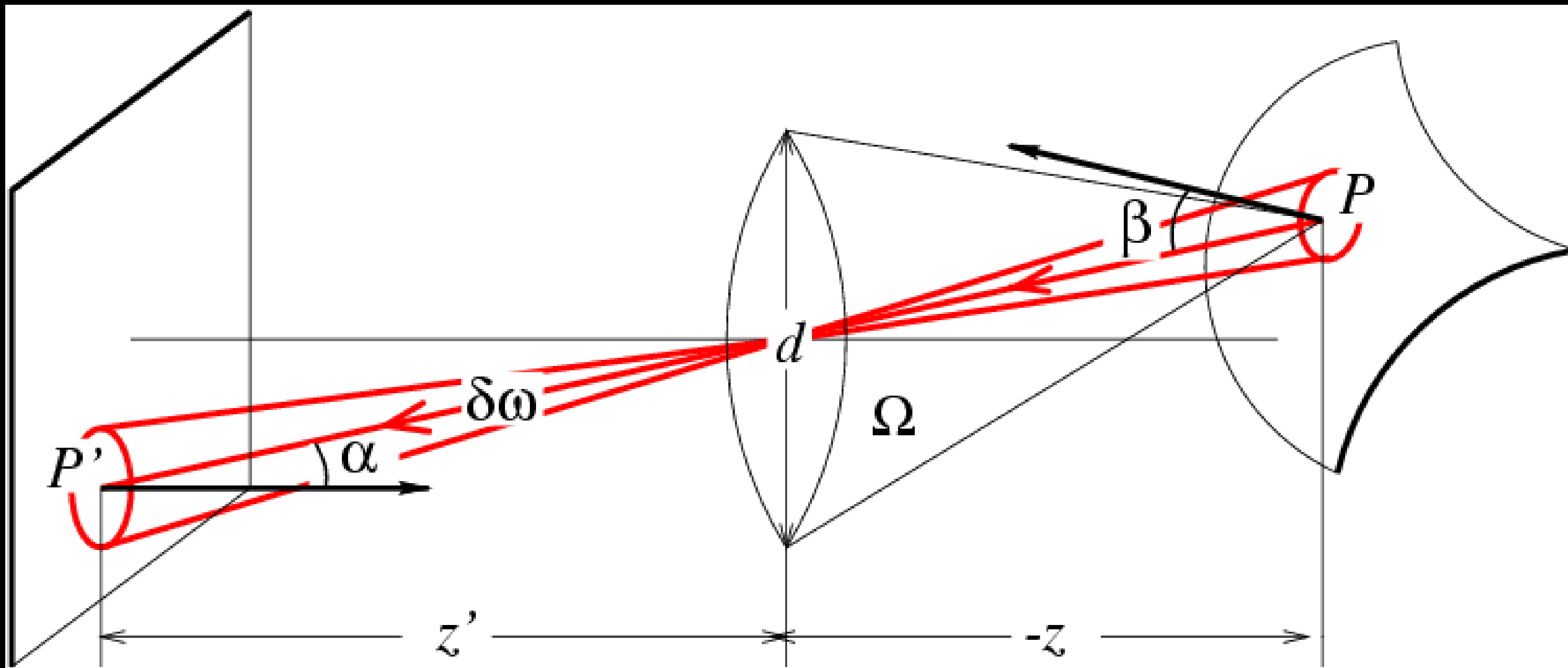
$$x'' = \frac{1}{\lambda}x'$$

$$y'' = \frac{1}{\lambda}y'$$

$$\lambda = 1 + k_1 r^2 + k_2 r^4 + \cdots$$

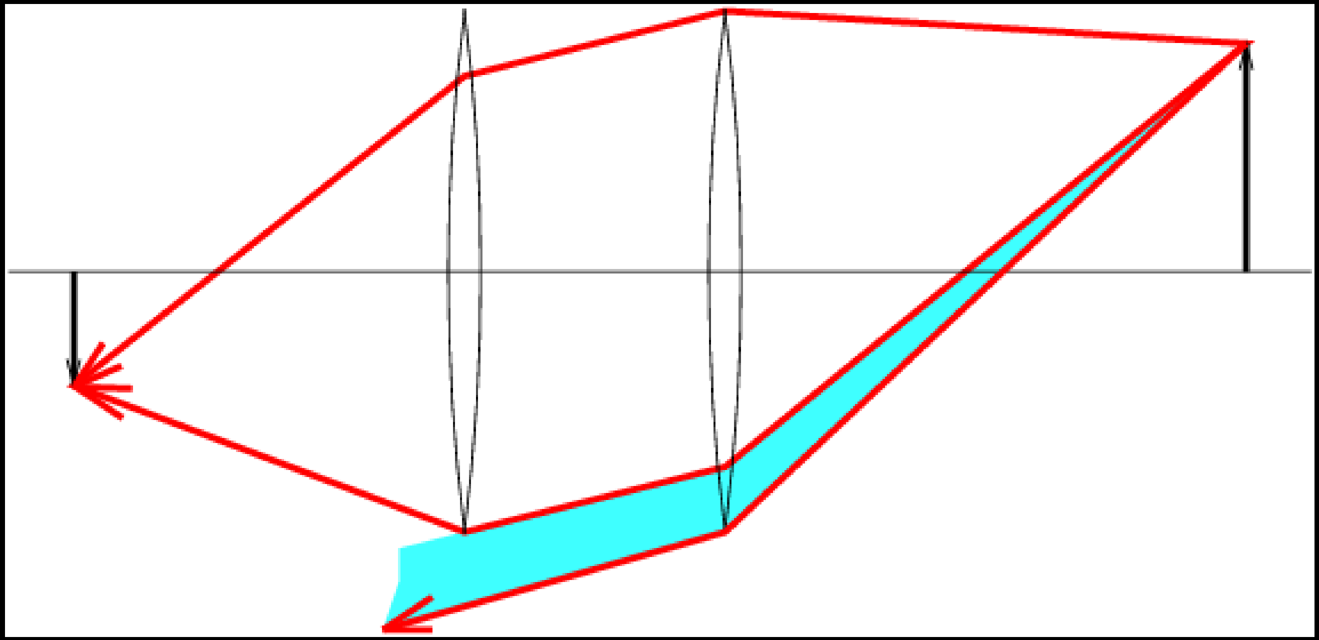
A compound lens





$$E = (\Pi/4)\left[\, (d/z')^2 \cos^4\alpha \,\right] L$$

Vignetting



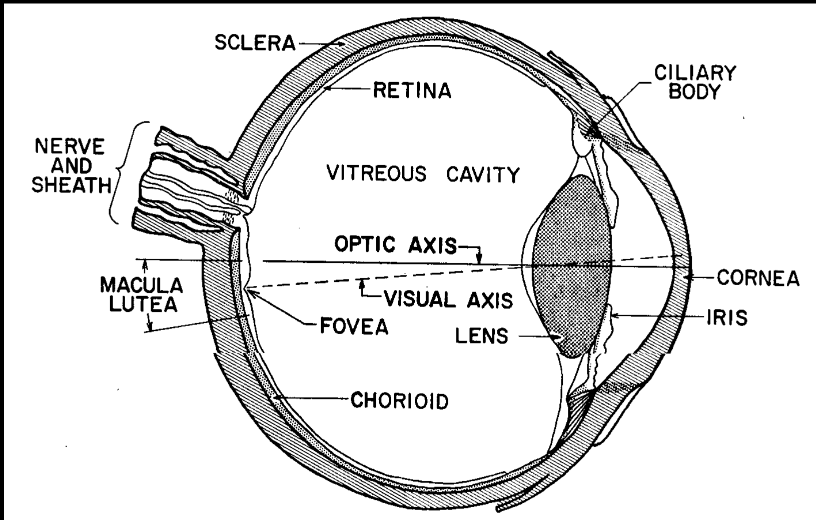
Challenge: Illumination – What is wrong with these pictures?

Introduction to computer vision 2.5

Jean Ponce and Elena Sizikova
jean.ponce@inria.fr, es5223@nyu.edu

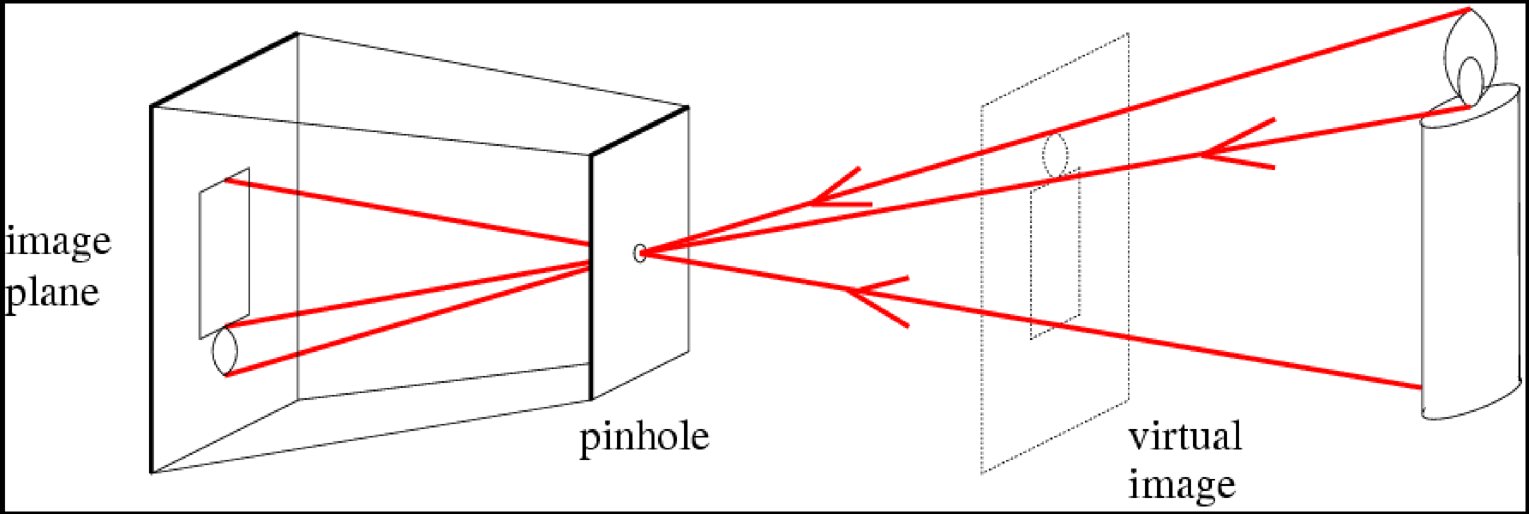
Camera geometry and calibration

- *A detour through sensing country*
- Intrinsic and extrinsic parameters



Animal eye: a looonnng time ago.

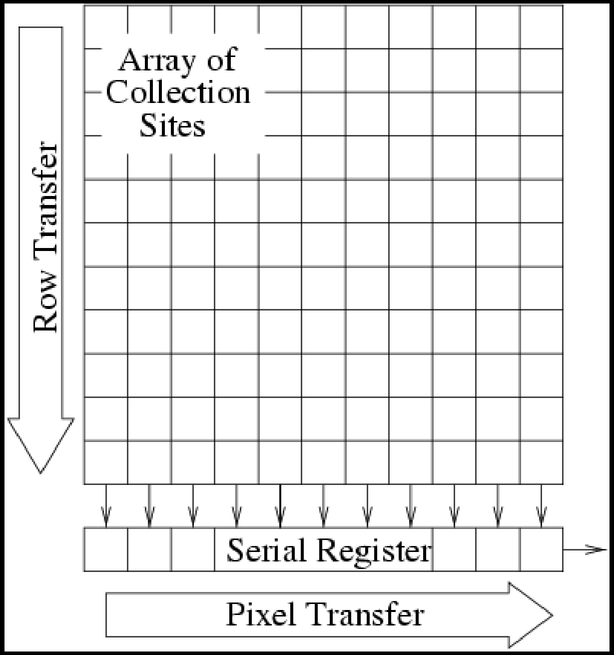
Photographic camera: Niepce, 1816.



Pinhole perspective projection: Brunelleschi, XVth Century.
Camera obscura: XVIth Century.

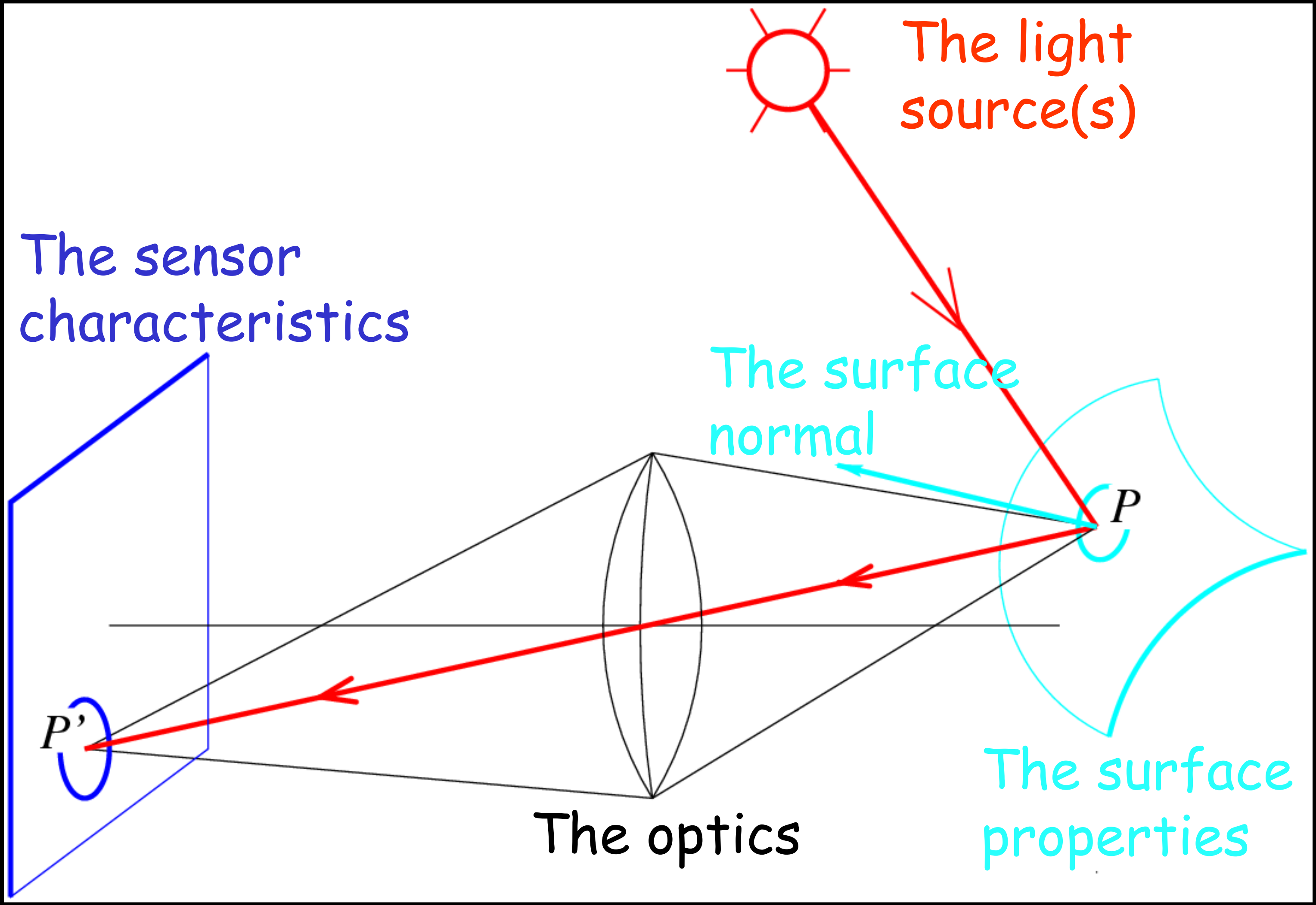
Photography
(Niepce, "La Table Servie," 1822)

Milestones:
- Daguerréotypes (1839)
- Photographic Film (Eastman, 1889)
- Cinema (Lumière brothers, 1895)
- Color Photography (Lumière brothers, again, 1908)
- Television (Baird, Farnsworth, Zworykin, 1920s)



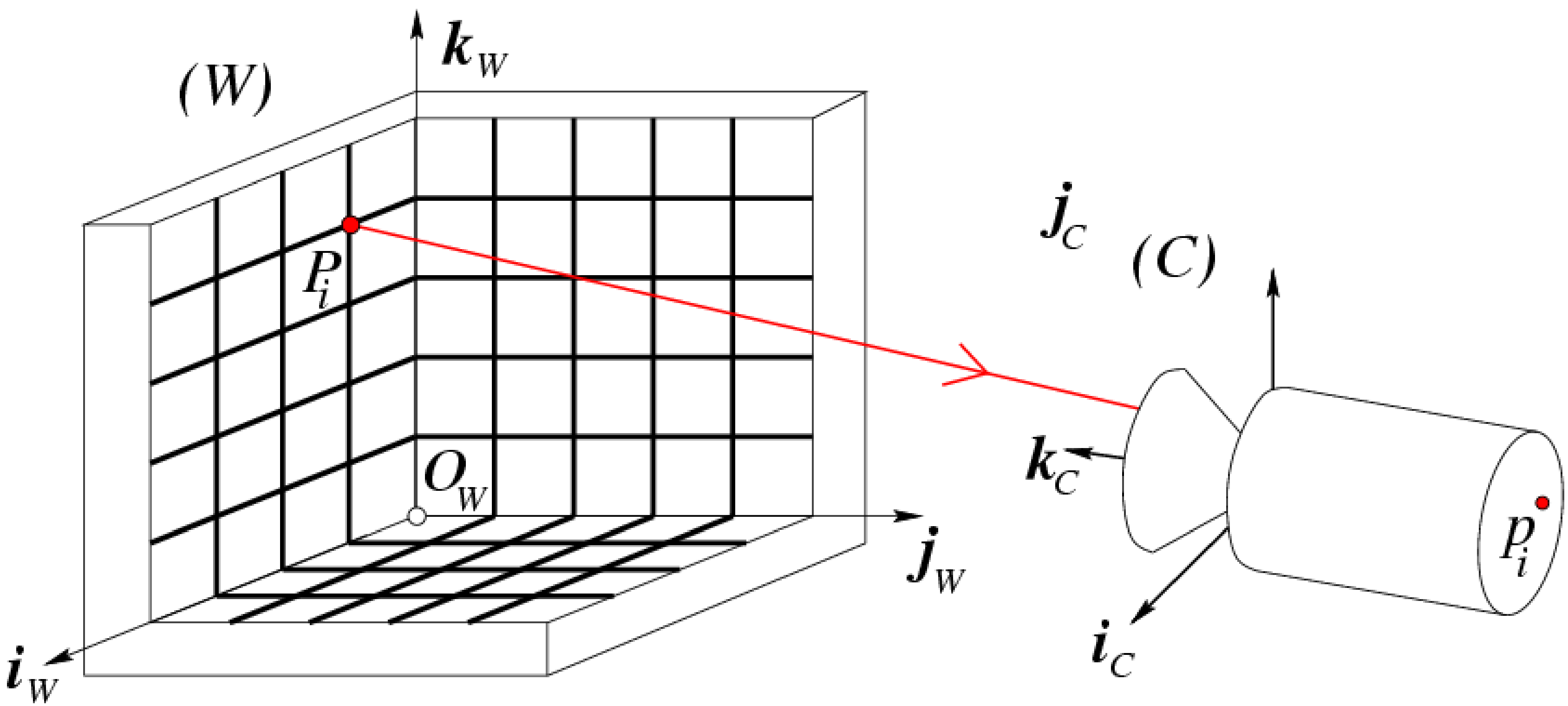
CCD Devices (1970), etc.

Image Formation: Radiometry

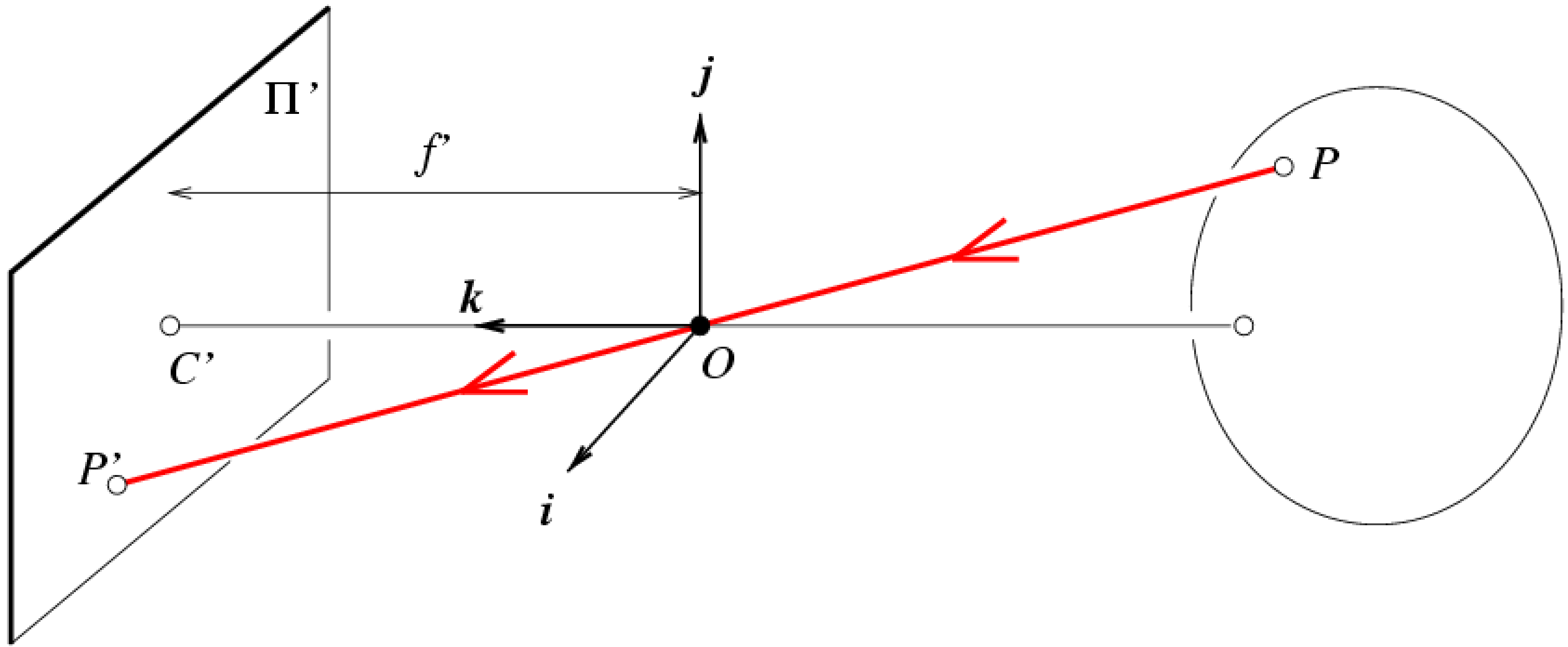


What determines the brightness of an image pixel?

Perspective Projection	$x' = f\dfrac{x}{z}$ $y' = f\dfrac{y}{z}$	*x*,*y*: World coordinates *x'*,*y'*: Image coordinates *f*: pinhole-to-retina distance
Weak-Perspective Projection (Affine)	$x' \approx -mx$ $y' \approx -my$ $\quad m = -\dfrac{f'}{\bar{z}}$	*x*,*y*: World coordinates *x'*,*y'*: Image coordinates *m*: magnification
Orthographic Projection (Affine)	$x' \approx -mx$ $y' \approx -my$ $\quad m = -\dfrac{f'}{\bar{z}}$	*x*,*y*: World coordinates *x'*,*y'*: Image coordinates
Common distortion model	$x'' = \dfrac{1}{\lambda}x'$ $y'' = \dfrac{1}{\lambda}y'$ $\lambda = 1 + k_1 r^2 + k_2 r^4 + \cdots$	*x'*,*y'*: Ideal image coordinates *x''*,*y''*: Actual image coordinates



$\boldsymbol{k}_W$
(W)
P_i
O_W
$\boldsymbol{j}_W$
$\boldsymbol{i}_W$
$\boldsymbol{j}_C$
(C)
$\boldsymbol{k}_C$
$\boldsymbol{i}_C$
p_i



$$\begin{cases} x' = f' \dfrac{x}{z} \\ y' = f' \dfrac{y}{z} \end{cases}$$

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

$$v = f' \frac{x}{z}$$

$$p = \frac{1}{z}\left[\begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}\right] \mathrm{P}$$

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Homogeneous coordinates

Conversion

Converting to *homogeneous* coordinates

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

homogeneous image coordinates

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

homogeneous scene coordinates

Converting *from* homogeneous coordinates

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

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

Homogeneous coordinates

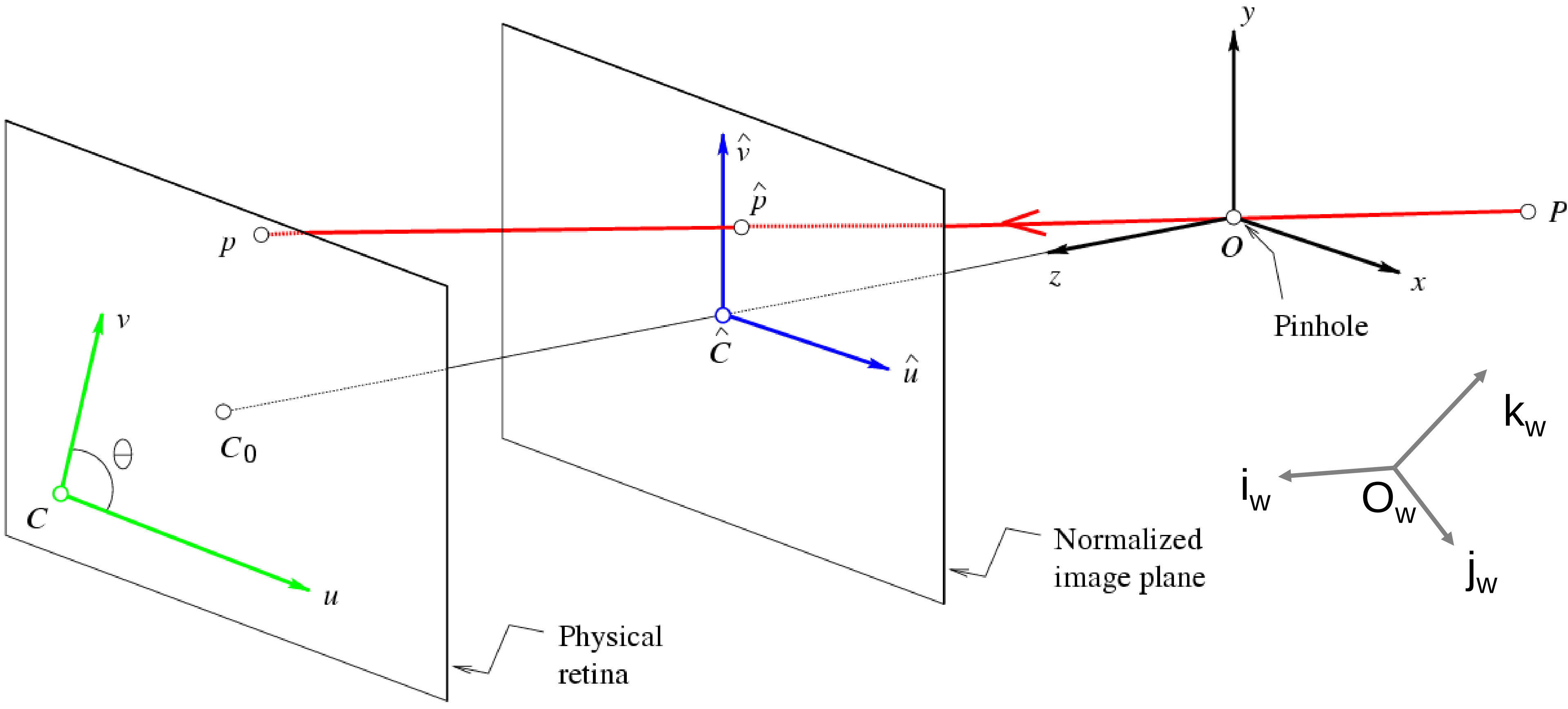
Invariant to scaling

$$k\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous Coordinates

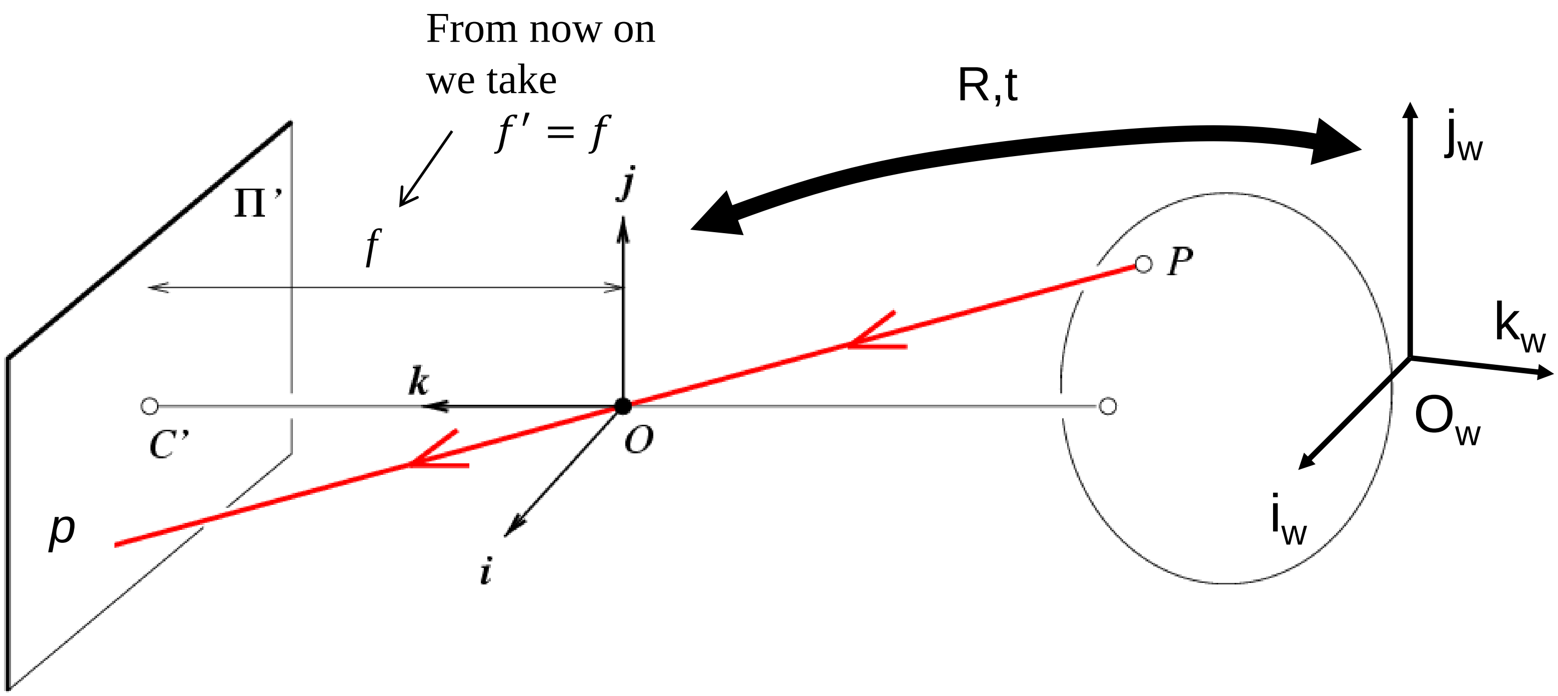
Cartesian Coordinates

A point in Cartesian coordinates is a ray in homogeneous ones



y
$\hat{v}$
$\hat{p}$
P
p
O
z
x
Pinhole
v
$\hat{C}$
$\hat{u}$
C_0
θ
C
u
k_w
i_w
O_w
j_w
Normalized
image plane
Physical
retina

Projection matrix

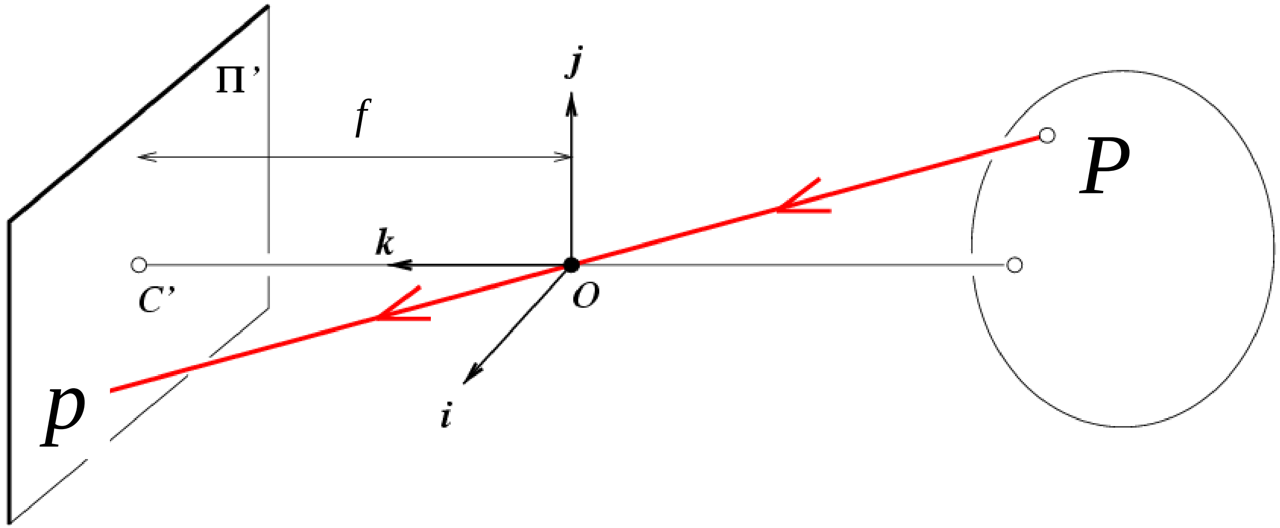


$$p \approx MP = K\,[R\ t]P$$

$$p = \lambda\, MP \ \text{ for some } \lambda \neq 0$$

p: Image Coordinates: (u,v,1)
M: 3x4 projection matrix
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
P: World Coordinates: (x,y,z,1)

Projection matrix



Intrinsic Assumptions

- Unit aspect ratio
- Image center at (0,0)
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$p \approx K\,[\mathrm{Id}\ 0]P \quad \Rightarrow \quad w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(Note: here $w = z$)

Slide Credit: Savarese

Remove assumption: known optical center

Intrinsic Assumptions

- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$p \approx K\ [\mathrm{Id}\ 0]P \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: square pixels

Intrinsic Assumptions
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$p \approx K\,[\mathrm{Id}\ 0]P \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: non-skewed pixels

Intrinsic Assumptions

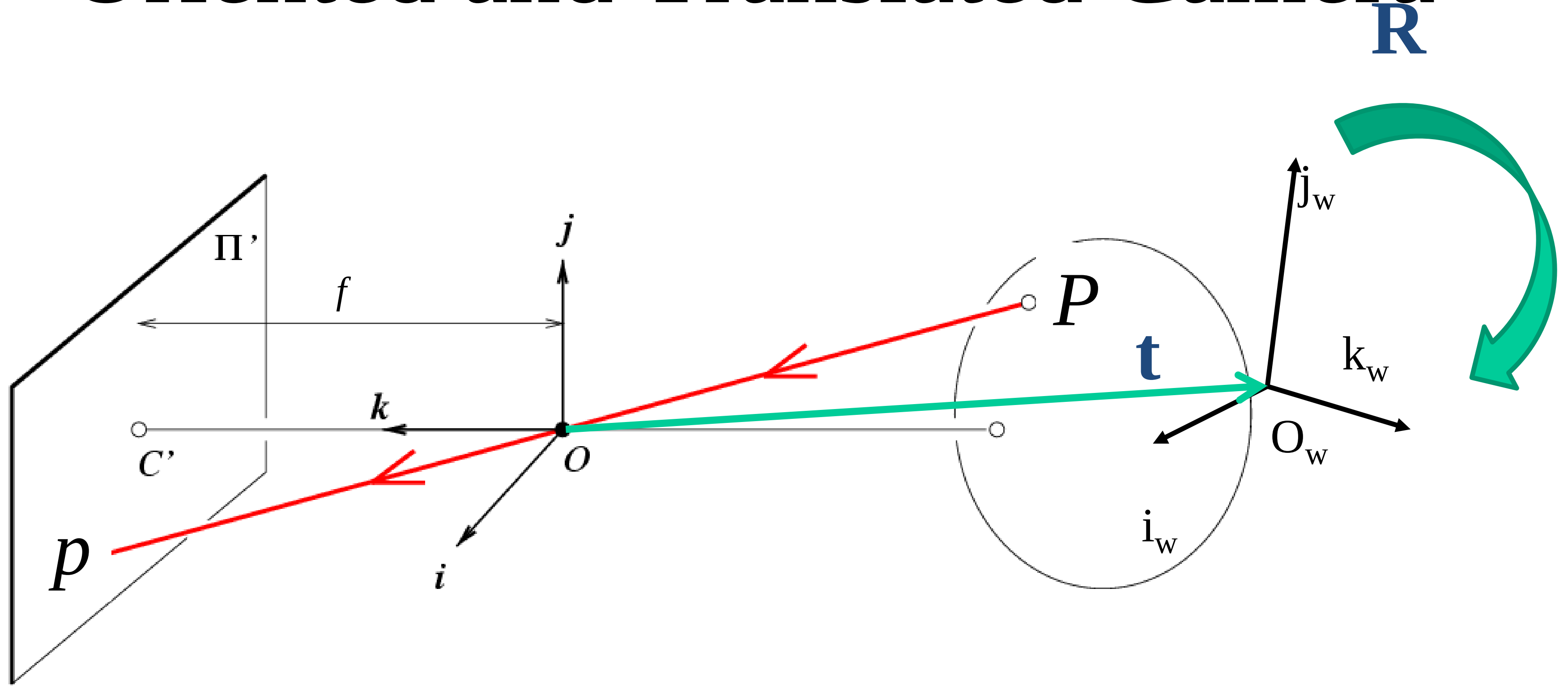
Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$p \approx K\ [\mathrm{Id}\ 0]P \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Note: different books use different notation for parameters

Oriented and Translated Camera



Allow camera translation

Intrinsic Assumptions

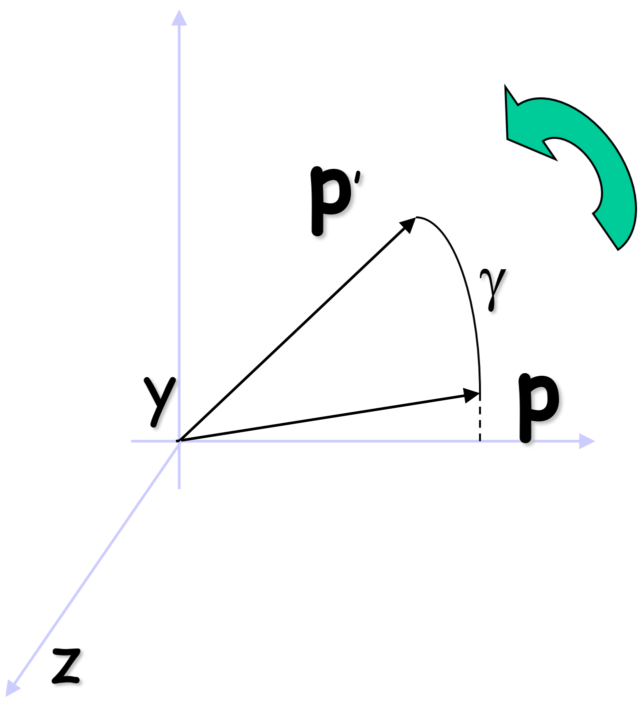
Extrinsic Assumptions

- No rotation

$$p \approx K\,[\mathrm{Id}\ t]P \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Allow camera rotation

$$p \approx K\,[R\ t]P$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Degrees of freedom

$$p \approx K\,[R\ t]P$$

$$\Downarrow$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overset{5}{\begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{6}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

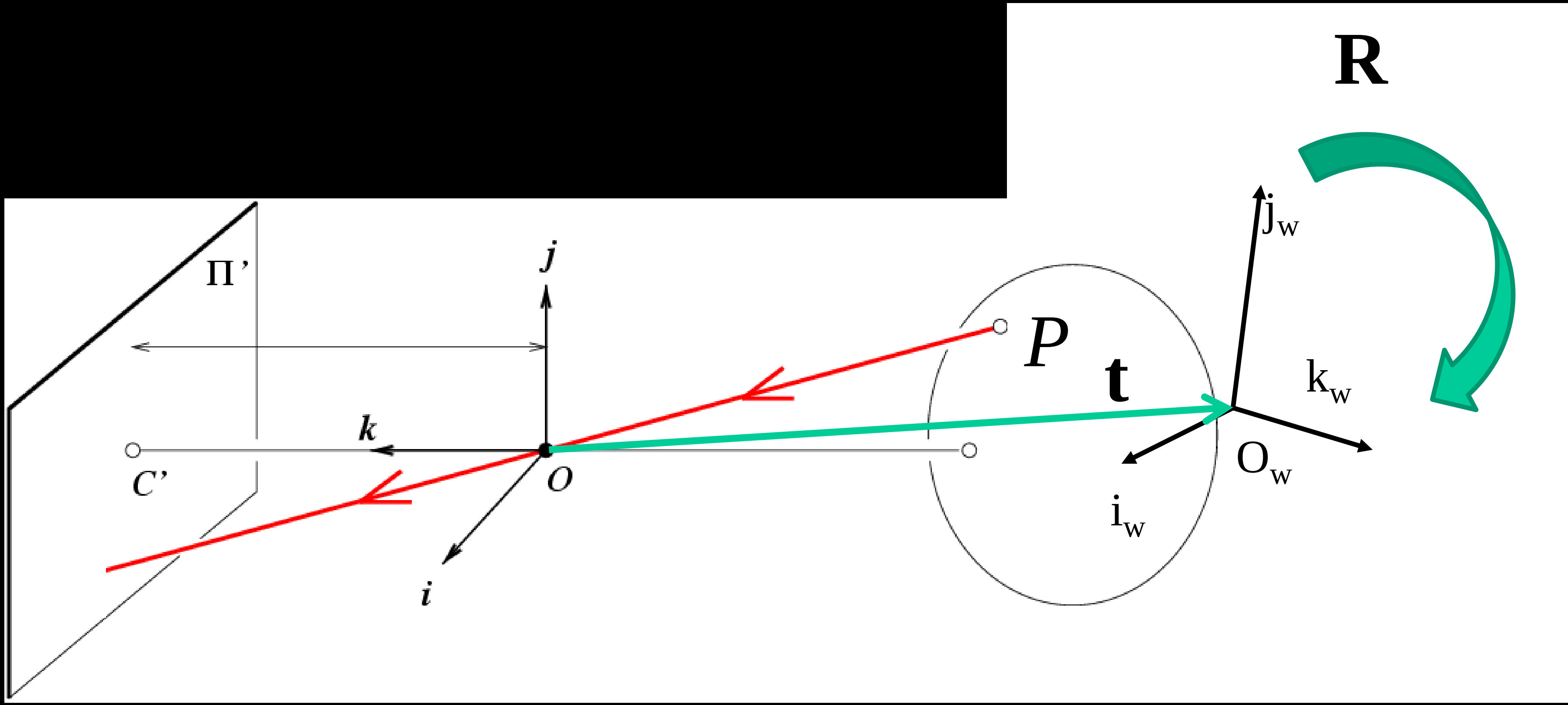
Introduction to computer vision 2.75

Jean Ponce and Elena Sizikova
jean.ponce@inria.fr, es5223@nyu.edu

Camera geometry and calibration

- Linear calibration
- Analytical photogrammetry

Oriented and Translated Camera



$$p \approx K\,[R\ t]P$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$p \approx MP$ or $p = \frac{1}{z} MP$

$M = K\,[R\ t]$

Note: $z = w$ here is depth relative to the camera coordinate system, it is not the last coordinate of (x, y, z) in world coordinates

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

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

$$p \approx MP \text{ or } p = \frac{1}{z} MP$$

$$M = K\,[R\ t]$$

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

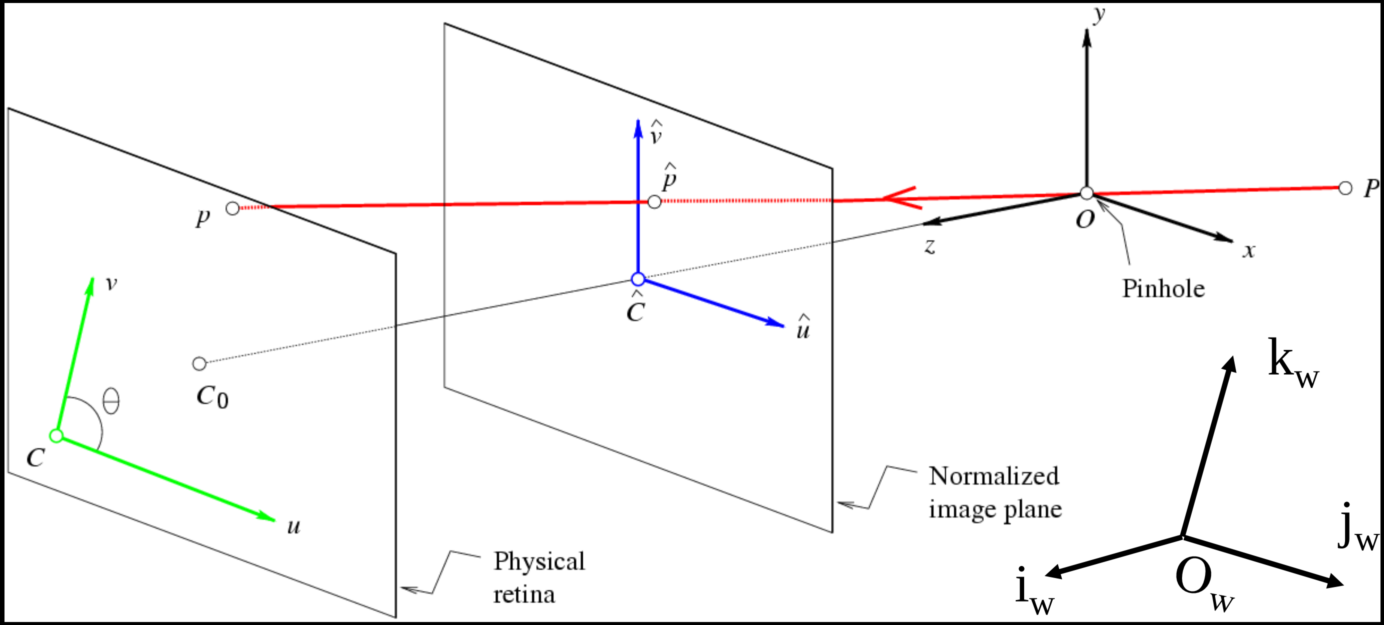
$$\mathcal{M} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \end{pmatrix} \Longrightarrow z = \boldsymbol{m}_3 \cdot \boldsymbol{P}, \quad \text{or} \quad \begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}, \\ v =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

$$p = K\hat{p} \text{ and } \hat{p} = \frac{1}{z} \widehat{M} P$$

normalized coordinates

$$\widehat{M} = [R\ t]$$

Explicit form of the projection matrix



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

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!!

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

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

Reminder:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \, \boldsymbol{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$$

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

$$(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{a} = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{b} = 0$$

If $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ is an orthonormal basis of R^3

$$\boldsymbol{i} \times \boldsymbol{j} = \boldsymbol{k}, \qquad \boldsymbol{j} \times \boldsymbol{k} = \boldsymbol{i},$$

$$\boldsymbol{k} \times \boldsymbol{i} = \boldsymbol{j}$$

Detour: weak perspective

$$\boldsymbol{p} = \frac{1}{z_r}\mathcal{M}\boldsymbol{P} \quad \rightarrow \quad \boldsymbol{p} = \mathcal{M}\boldsymbol{P} \quad \rightarrow \quad \boldsymbol{p} = \mathcal{A}\boldsymbol{P} + \boldsymbol{b}$$

Definition: A 2x4 matrix **M** = [**A b**], where **A** is a rank-2 2x3 matrix, is called an affine projection matrix.

Theorem: All affine projection models can be represented by affine projection matrices.

General form of the weak-perspective projection equation:

$$\mathbf{M} = \frac{1}{z_r}\begin{bmatrix} k & s \\ 0 & 1 \end{bmatrix}\begin{bmatrix}\mathbf{R}_2 & \mathbf{t}_2\end{bmatrix} \qquad (1)$$

Theorem: An affine projection matrix can be written uniquely (up to a sign amibguity) as a weak perspective projection matrix as defined by (1).

Linear Perspective Camera Calibration

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

Remember: $a \cdot b = a^T b$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \end{pmatrix} \Longleftrightarrow \begin{pmatrix} \boldsymbol{m}_1 - u_i \boldsymbol{m}_3 \\ \boldsymbol{m}_2 - v_i \boldsymbol{m}_3 \end{pmatrix} \cdot P_i = 0$$

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1 \boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \ldots & \ldots & \ldots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n \boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0$$

Homogeneous Linear Systems

$$A \, x = 0$$

Square system:

- unique solution: 0
- unless Det(A)=0

$$A \, x = 0$$

Rectangular system ??

- 0 is always a solution

Minimize $||Ax||^2$
under the constraint
$||x||^2 = 1$

How do you solve overconstrained homogeneous linear equations ?? Homogeneous linear least squares

$$E = \|\mathcal{U}\boldsymbol{x}\|^2 = \boldsymbol{x}^T(\mathcal{U}^T\mathcal{U})\boldsymbol{x}$$

- Orthonormal basis of eigenvectors: $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_q$.
- Associated eigenvalues: $0 \le \lambda_1 \le \ldots \le \lambda_q$.
- Any vector can be written as

such that $||x|| = 1$

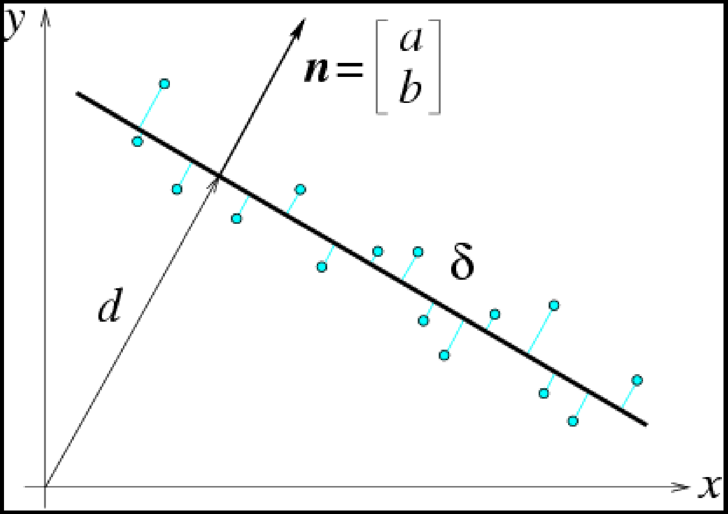
$$\boldsymbol{x} = \mu_1 \boldsymbol{e}_1 + \ldots + \mu_q \boldsymbol{e}_q$$

for some μ_i $(i = 1, \ldots, q)$ such that $\mu_1^2 + \ldots + \mu_q^2 = 1$.

$$\begin{aligned} E(\mathbf{x})-E(\mathbf{e}_1) &= \mathbf{x}^T(U^TU)\mathbf{x}-\mathbf{e}_1{}^T(U^TU)\mathbf{e}_1 \\ &= \lambda_1\mu_1{}^2+ \ldots +\lambda_q\mu_q{}^2-\lambda_1 \\ &\geq \lambda_1(\mu_1{}^2+ \ldots +\mu_q{}^2-1)=0 \end{aligned}$$

$\boxed{\text{The solution is } \mathbf{e}_1 \text{ .}}$

Example: Line Fitting



Problem: minimize

$$E(a,b,d) = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

with respect to (*a*,*b*,*d*).

- Minimize *E* with respect to *d*:

$$\frac{\partial E}{\partial d} = 0 \Longrightarrow d = \sum_{i=1}^{n} \frac{ax_i + by_i}{n} = a\bar{x} + b\bar{y}$$

- Minimize *E* with respect to *a*,*b*:

$$E = \sum_{i=1}^{n} [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = |\mathcal{U}\boldsymbol{n}|^2$$ where $$\mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \ldots & \ldots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

- Done !!

Linear Camera Calibration

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \\ \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3 \cdot \boldsymbol{P}_i} \end{pmatrix} \Longleftrightarrow \begin{pmatrix} \boldsymbol{m}_1 - u_i \boldsymbol{m}_3 \\ \boldsymbol{m}_2 - v_i \boldsymbol{m}_3 \end{pmatrix} \boldsymbol{P}_i = 0$$

$$\mathcal{P}\boldsymbol{m} = 0 \text{ with } \mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1 \boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \ldots & \ldots & \ldots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n \boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \text{ and } \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0$$

Minimize $||Pm||^2$ under the constraint $||m||^2 = 1$

Once *M* is known, you still got to recover the intrinsic and extrinsic parameters !!!

This is a decomposition problem, not an estimation problem.

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

- Intrinsic parameters
- Extrinsic parameters

Degenerate Point Configurations

Are there other solutions besides *M* ??

$$\mathbf{0} = \mathcal{P}\boldsymbol{l} = \begin{pmatrix} \boldsymbol{P}_1^T & \mathbf{0}^T & -u_1\boldsymbol{P}_1^T \\ \mathbf{0}^T & \boldsymbol{P}_1^T & -v_1\boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \mathbf{0}^T & -u_n\boldsymbol{P}_n^T \\ \mathbf{0}^T & \boldsymbol{P}_n^T & -v_n\boldsymbol{P}_n^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_1^T\boldsymbol{\lambda} - u_1\boldsymbol{P}_1^T\boldsymbol{\nu} \\ \boldsymbol{P}_1^T\boldsymbol{\mu} - v_1\boldsymbol{P}_1^T\boldsymbol{\nu} \\ \dots \\ \boldsymbol{P}_n^T\boldsymbol{\lambda} - u_n\boldsymbol{P}_n^T\boldsymbol{\nu} \\ \boldsymbol{P}_n^T\boldsymbol{\mu} - v_n\boldsymbol{P}_n^T\boldsymbol{\nu} \end{pmatrix}$$

$$\Downarrow$$

$$\begin{cases} \boldsymbol{P}_i^T\boldsymbol{\lambda} - \dfrac{\boldsymbol{m}_1^T\boldsymbol{P}_i}{\boldsymbol{m}_3^T\boldsymbol{P}_i}\boldsymbol{P}_i^T\boldsymbol{\nu} = 0 \\ \boldsymbol{P}_i^T\boldsymbol{\mu} - \dfrac{\boldsymbol{m}_2^T\boldsymbol{P}_i}{\boldsymbol{m}_3^T\boldsymbol{P}_i}\boldsymbol{P}_i^T\boldsymbol{\nu} = 0 \end{cases} \Rightarrow \begin{cases} \boldsymbol{P}_i^T(\boldsymbol{\lambda}\boldsymbol{m}_3^T - \boldsymbol{m}_1\boldsymbol{\nu}^T)\boldsymbol{P}_i = 0 \\ \boldsymbol{P}_i^T(\boldsymbol{\mu}\boldsymbol{m}_3^T - \boldsymbol{m}_2\boldsymbol{\nu}^T)\boldsymbol{P}_i = 0 \end{cases}$$

- Coplanar points: $(\lambda, \mu, \nu) = (\pi,0,0)$ or $(0,\pi,0)$ or $(0,0,\pi)$
- Points lying on the intersection curve of two quadric surfaces = straight line + twisted cubic

Does not happen for 6 or more random points!

Analytical Photogrammetry

Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$

Find $\boldsymbol{i}$ and $\boldsymbol{e}$ such that

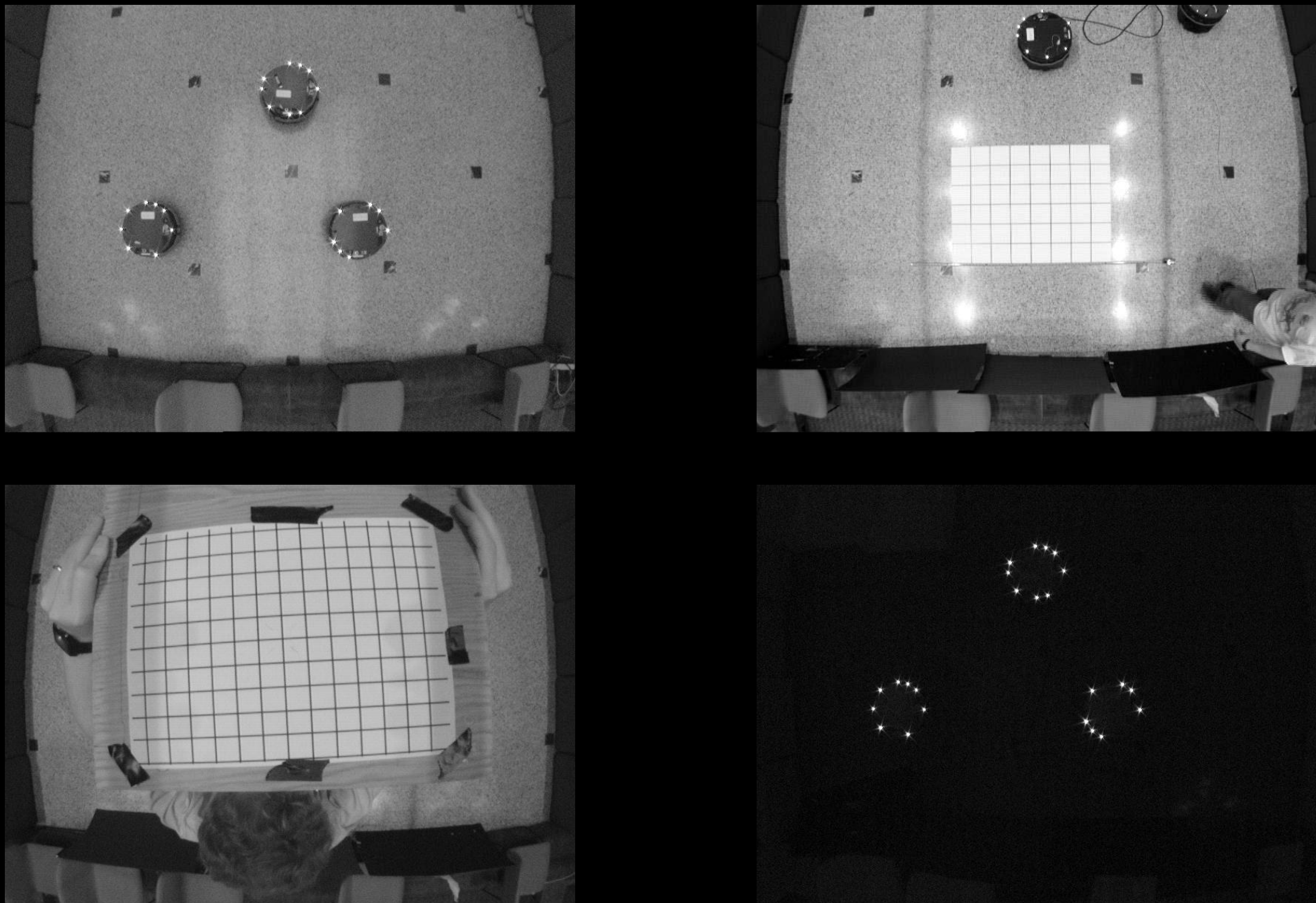
$$\sum_{i=1}^{n} \left[\left(u_i - \frac{\boldsymbol{m}_1(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 + \left(v_i - \frac{\boldsymbol{m}_2(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i}{\boldsymbol{m}_3(\boldsymbol{i}, \boldsymbol{e}) \cdot \boldsymbol{P}_i} \right)^2 \right] \quad \text{is minimized}$$

Non-Linear Least-Squares Methods

- Newton
- Gauss-Newton
- Levenberg-Marquardt

Iterative, quadratically convergent in favorable situations

Applications: Mobile Robot Localization (Devy et al., 1997)



(Rothganger, Sudsang, Ponce, 2002)