Image Formation and Cameras (Part 3)

Computer Vision I

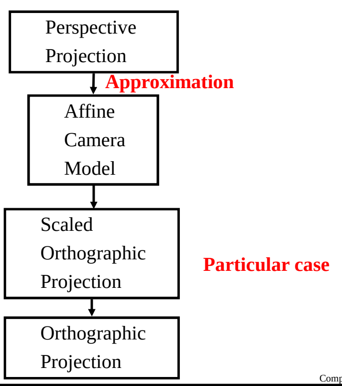
CSE 252A

Lecture 5

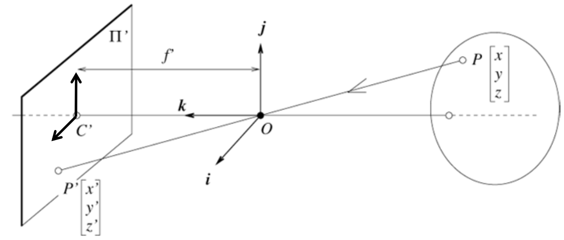
Announcements

- Instructor office hours TBD
- Homework 1 is due Oct 23, 11:59 PM
- Wait list

Simplified Camera Models



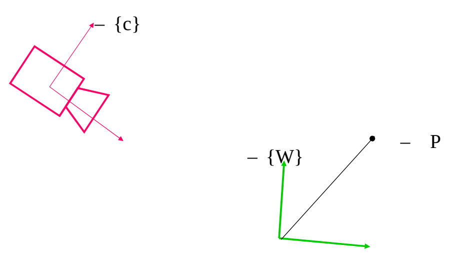
Equation of Perspective Projection



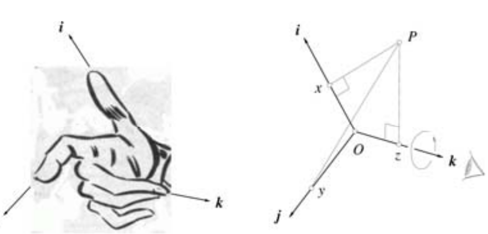
– Cartesian coordinates:

- We have, by similar triangles, that (x', y', z') = (f' x/z, f' y/z, f')
- Establishing an image plane coordinate system at C' aligned with i and j, image coordinates of the projection of P are $(x,y,z) \rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$

What if camera coordinate system differs from object coordinate system

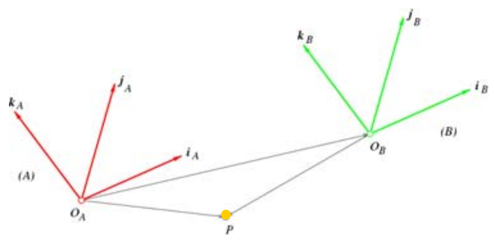


–Euclidean Coordinate Systems



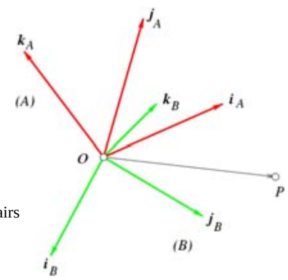
$$\begin{cases} x = \overrightarrow{OP}.\mathbf{i} \\ y = \overrightarrow{OP}.\mathbf{j} \\ z = \overrightarrow{OP}.\mathbf{k} \end{cases} \Leftrightarrow \overrightarrow{OP} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \Leftrightarrow \mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

–Coordinate Changes: Pure Translations



$$\overrightarrow{O_B P} = \overrightarrow{O_B O_A} + \overrightarrow{O_A P} \qquad -{}^{B}P = {}^{A}P + {}^{B}O_A$$

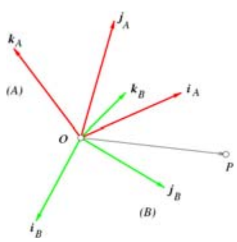
–Rotation Matrix



–Dot Products between all pairs of coordinate axis of both systems

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

–Coordinate Changes: Pure Rotations

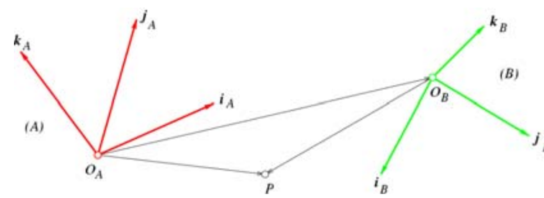


$$\overrightarrow{OP} = \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} {}^{A}P = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix} {}^{B}P$$

$$\Rightarrow \quad {}^{B}P = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} {}^{A}P = {}^{B}_{A}R\, {}^{A}P$$

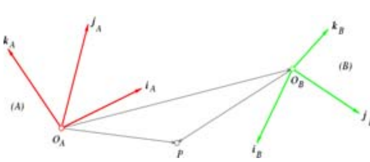
–Coordinate Changes: Rigid Transformations



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

–A convenient notation

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



– Points: ${}^{A}P_1$
  – Leading superscript indicates the coordinate system that the coordinates are with respect to
  – Subscript – an identifier
– Rotation Matrices ${}^{B}_{A}R$
  – Lower left (Going from this system)
  – Upper left (Going to this system)
– To add vectors, coordinate systems must agree
– To rotate a vector, points coordinate system must agree with lower left of rotation matrix

Some points about SO(n)

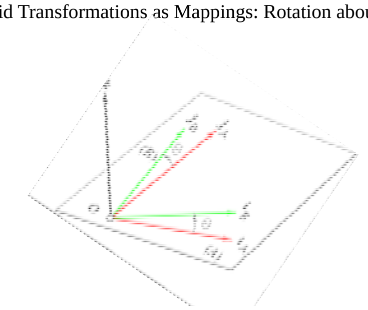
- SO(n) = { $R \in \Re^{n\mathrm{x}n}$: $R^T R = I$, det(R) = 1}
  - SO(2): rotation matrices in plane $\Re^2$
  - SO(3): rotation matrices in 3-space $\Re^3$
- Forms a Group under matrix product operation:
  - Identity
  - Inverse
  - Associative
  - Closure
- Closed (finite intersection of closed sets)
- Bounded $R_{i,j} \in [-1, +1]$
- Does not form a vector space.
- Manifold of dimension n(n-1)/2
  - Dim(SO(2)) = 1
  - Dim(SO(3)) = 3

Parameterizations of SO(3)

–Even though a rotation matrix is 3x3 with nine numbers, it only has three degrees of freedom, it can be parameterized with three numbers. There are many parameterization.

- 3-D manifold, so between 3 parameters and 2n+1 parameters (Whitney's Embedding Thm.)
  - Roll-Pitch-Yaw
  - Euler Angles
  - Axis Angle (Rodrigues formula)
  - Cayley's formula
  - Matrix Exponential
  - Quaternions (four parameters + one constraint)

–Rigid Transformations as Mappings: Rotation about the **k** Axis

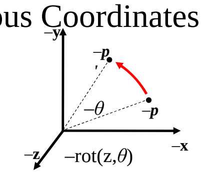


$$^{F}P' = \mathcal{R}^{F}P, \quad \text{where} \quad \mathcal{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

– = rot(k,θ)

Rotation: Homogenous Coordinates

- About z axis



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation

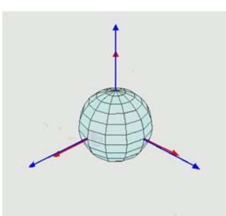
- About x axis:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- About y axis:

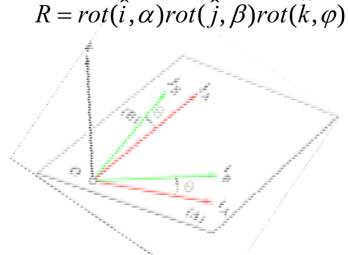
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Composition of Rotations



Roll-Pitch-Yaw

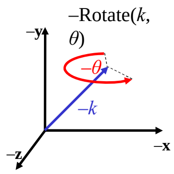
$$R = rot(\hat{i},\alpha)rot(\hat{j},\beta)rot(\hat{k},\varphi)$$



–Euler Angles

$$R = rot(\hat{k}'',\alpha)rot(\hat{j}',\beta)rot(\hat{k},\varphi)$$

Rotation



- About (k_x, k_y, k_z), a unit vector on an arbitrary axis (Rodrigues Formula)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} k_xk_x(1-c)+c & k_zk_x(1-c)-k_zs & k_xk_z(1-c)+k_ys & 0 \\ k_yk_x(1-c)+k_zs & k_zk_x(1-c)+c & k_yk_z(1-c)-k_xs & 0 \\ k_zk_x(1-c)-k_ys & k_zk_x(1-c)-k_xs & k_zk_z(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where $c = \cos\theta$ & $s = \sin\theta$

–Quaternions

–q = $(a,\boldsymbol{\alpha})$

–q is a quaternion (generalization of
– imaginary numbers)
–$a \in$ R is its real part
–$\boldsymbol{\alpha} \in$ R^3 is its imaginary part.

–Operations on quaternions:

- Sum of quaternions: $(a,\boldsymbol{\alpha}) + (b,\boldsymbol{\beta}) \equiv ((a+b),(\boldsymbol{\alpha}+\boldsymbol{\beta}))$
- Multiplication by a scalar: $\lambda(a,\boldsymbol{\alpha}) \equiv (\lambda a,\lambda\boldsymbol{\alpha})$
- Quaternion product:

– $(a,\boldsymbol{\alpha})(b,\boldsymbol{\beta}) \equiv ((a\,b - \boldsymbol{\alpha}\cdot\boldsymbol{\beta}), (a\,\boldsymbol{\beta} + b\,\boldsymbol{\alpha} + \boldsymbol{\alpha}\times\boldsymbol{\beta}))$

- Conjugate: q = $(a,\boldsymbol{\alpha})$ q' ≡ $(a, -\boldsymbol{\alpha})$
- Norm: $|q|^2 \equiv q\,q' = q'\,q = a^2 + |\boldsymbol{\alpha}|^2$

–Unit Quaternions and Rotations

- Let R denote the rotation of angle θ about the unit vector $\boldsymbol{u}$.
- Define unit quaternion q = (cos θ/2, sin θ/2 $\boldsymbol{u}$).
- Note |q| = 1 (i.e., q lies on unit sphere for any $\boldsymbol{u}$ and θ.
- Then for any vector $\boldsymbol{\alpha}$,

–$R\,\boldsymbol{\alpha}$ = imaginary(q $\boldsymbol{\alpha}^*$ q')

– where $\boldsymbol{\alpha}^* = (0, \boldsymbol{\alpha})$

•q and –q define the same rotation matrix.

–If q = $(a, (b, c, d)^T)$ is a unit quaternion, the corresponding rotation matrix is:

$$\mathcal{R} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

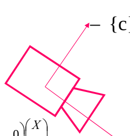
–Homogeneous Representation of Rigid Transformations

$$\begin{bmatrix} {}^BP \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B_AR\,{}^AP + {}^BO_A \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B_AR & {}^BO_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^AP \\ 1 \end{bmatrix} = {}^B_AT \begin{bmatrix} {}^AP \\ 1 \end{bmatrix}$$

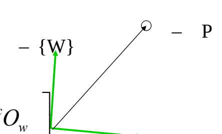
–Transformation represented by 4 by 4 Matrix

$$\begin{bmatrix} {}^B_AR & {}^BO_A \\ \mathbf{0}^T & 1 \end{bmatrix}$$

What if camera coordinate system differs from object coordinate system

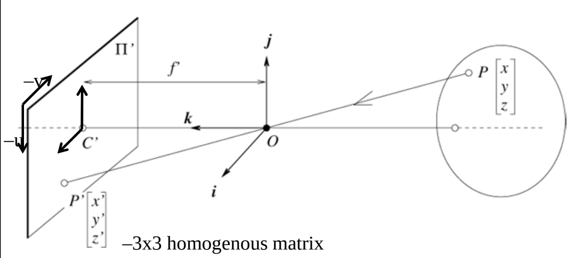


$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$



$${}^c_wT = \begin{bmatrix} {}^c_wR & {}^cO_w \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Intrinsic parameters



–3x3 homogenous matrix
–Focal length:
–Principal Point: C'
–Units (e.g. pixels)
–Orientation and position of image coordinate system
–Pixel Aspect ratio

Camera parameters

- Extrinsic Parameters: Since camera may not be at the origin, there is a rigid transformation between the world coordinates and the camera coordinates
- Intrinsic parameters: Since scene units (e.g., cm) differ image units (e.g., pixels) and coordinate system may not be centered in image, we capture that with a 3x3 transformation comprised of focal length, principal point, pixel aspect ratio, angle between axes, etc.

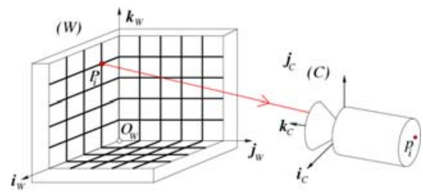
$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{represented by} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Rigid Transformation} \\ \text{represented by} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

–3 x 3

–4 x 4

$^{c}_{w}T$

Camera Calibration



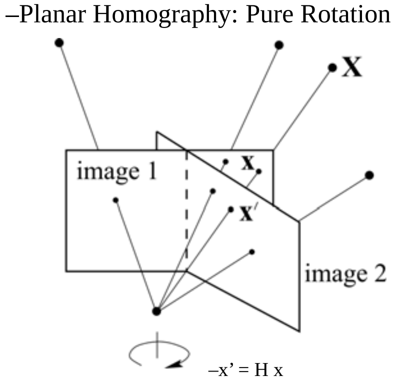
Given n points $P_1, \ldots, P_n$ with *known* positions and their images $p_1, \ldots, p_n$ –, estimate intrinsic and extrinsic camera parameters

- See Text book for how to do it.
- Camera Calibration Toolbox for Matlab (Bouguet)
  – http://www.vision.caltech.edu/bouguetj/calib_doc/

Application: Panoramas

Coordinates between pairs of images are related by projective transformations

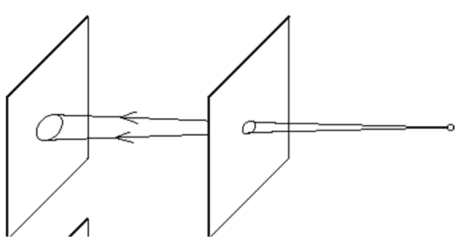
–Planar Homography: Pure Rotation



–x' = H x

–Figure borrowed from Hartley and Zisserman "Multiple View Geometry in computer vision"

Beyond the pinhole Camera Getting more light – Bigger Aperture

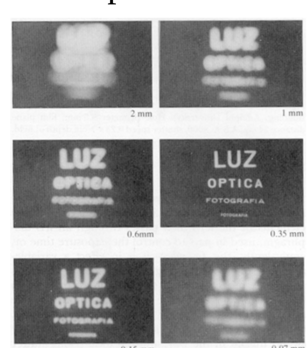


Pinhole Camera Images with Variable Aperture

2 mm **1mm**

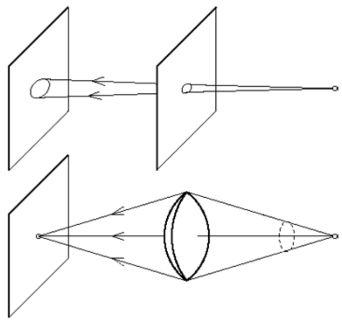
.6 mm **.35 mm**

.15 mm **.07 mm**

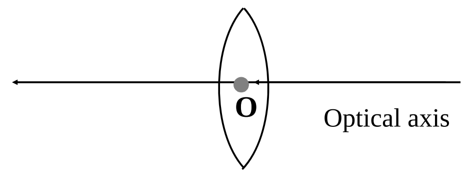


The reason for lenses

We need light, but big pinholes cause blur.

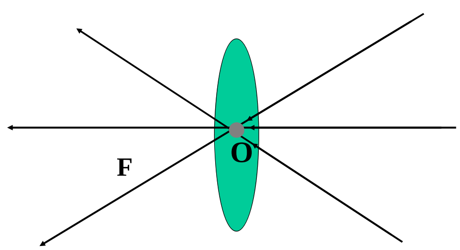


Thin Lens



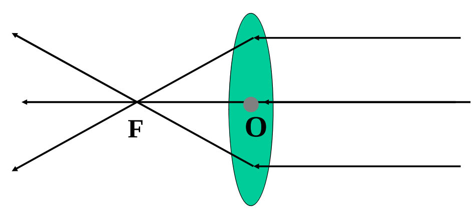
- Rotationally symmetric about optical axis.
- Spherical interfaces.

Thin Lens: Center



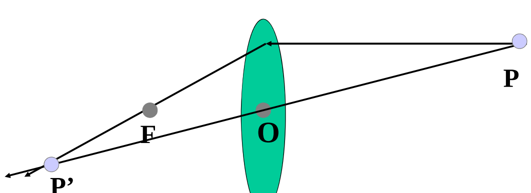
- All rays that enter lens along line pointing at **O** emerge in same direction.

Thin Lens: Focus



Parallel lines pass through the focus, F

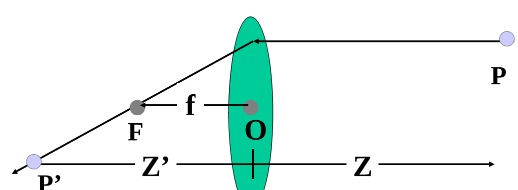
Thin Lens: Image of Point



– All rays passing through lens and starting at **P** converge upon **P'**

– So light gather capability of lens is given the area of the lens and all the rays focus on P' instead of become blurred like a pinhole

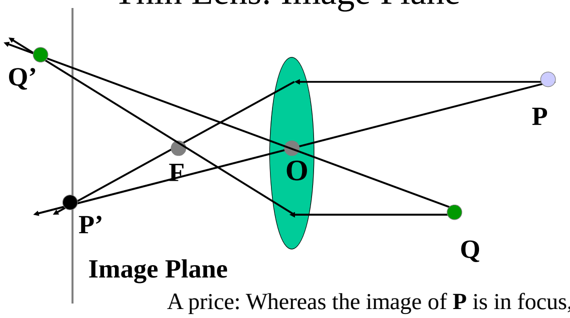
Thin Lens: Image of Point



$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

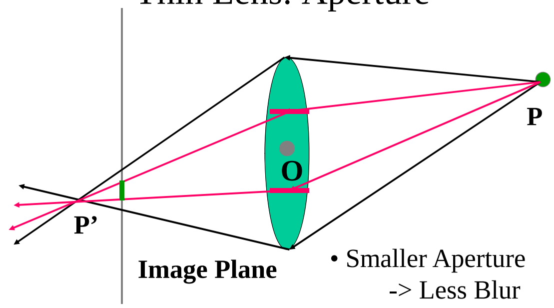
Relation between depth of Point (Z) and the depth where it focuses (Z')

Thin Lens: Image Plane



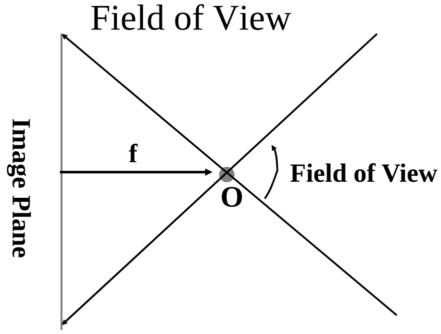
A price: Whereas the image of **P** is in focus, the image of **Q** isn’t.

Thin Lens: Aperture



- Smaller Aperture -> Less Blur
- Pinhole -> No Blur

Field of View



–Field of view is a function of **f** and size of image plane.

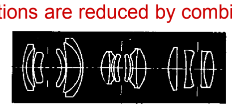
Deviations from the lens model

Deviations from this ideal are *aberrations*

Two types

1. geometrical
   - spherical aberration
   - astigmatism
   - distortion
   - coma
2. chromatic

Aberrations are reduced by combining lenses

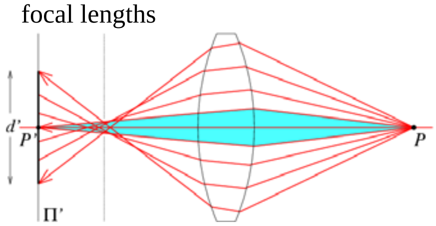


Compound lenses

Spherical aberration

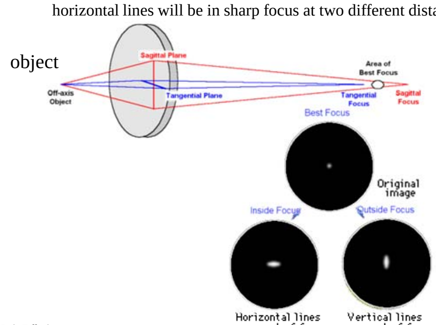
Rays parallel to the axis do not converge

Outer portions of the lens yield smaller focal lengths



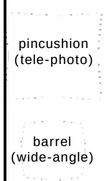
Astigmatism

An optical system with astigmatism is one where rays that propagate in two perpendicular planes have different focus. If an optical system with astigmatism is used to form an image of a cross, the vertical and horizontal lines will be in sharp focus at two different distances.



Distortion

magnification/focal length different
for different angles of inclination



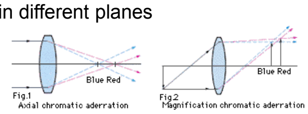
Can be corrected! (if parameters are know)

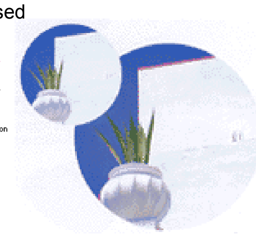
Chromatic aberration

(great for prisms, bad for lenses)

Chromatic aberration

rays of different wavelengths focused in different planes





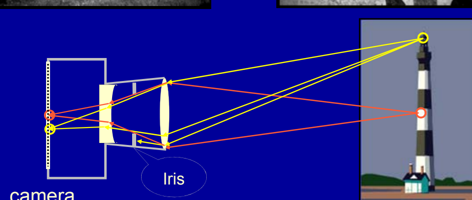
The image is blurred and appears colored at the fringe.

cannot be removed completely

Vignetting

- Only part of the light reaches the sensor
- Periphery of the image is dimmer

Vignetting: Spatial Non-Uniformity



Litvinov & Schechner, *radiometric nonidealities*