Computer Vision Group
Prof. Daniel Cremers

Technische Universität München

Robotic 3D Vision

Lecture 2: Image Formation, Multiple View Geometry Basics

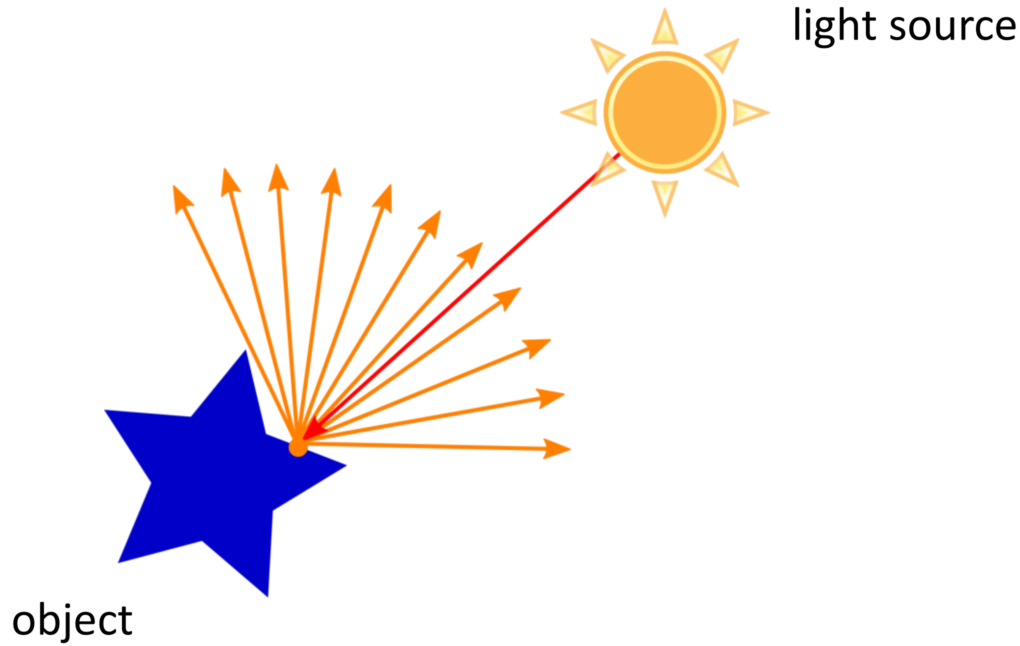
WS 2020/21

Dr. Niclas Zeller
Artisense GmbH

What We Will Cover Today

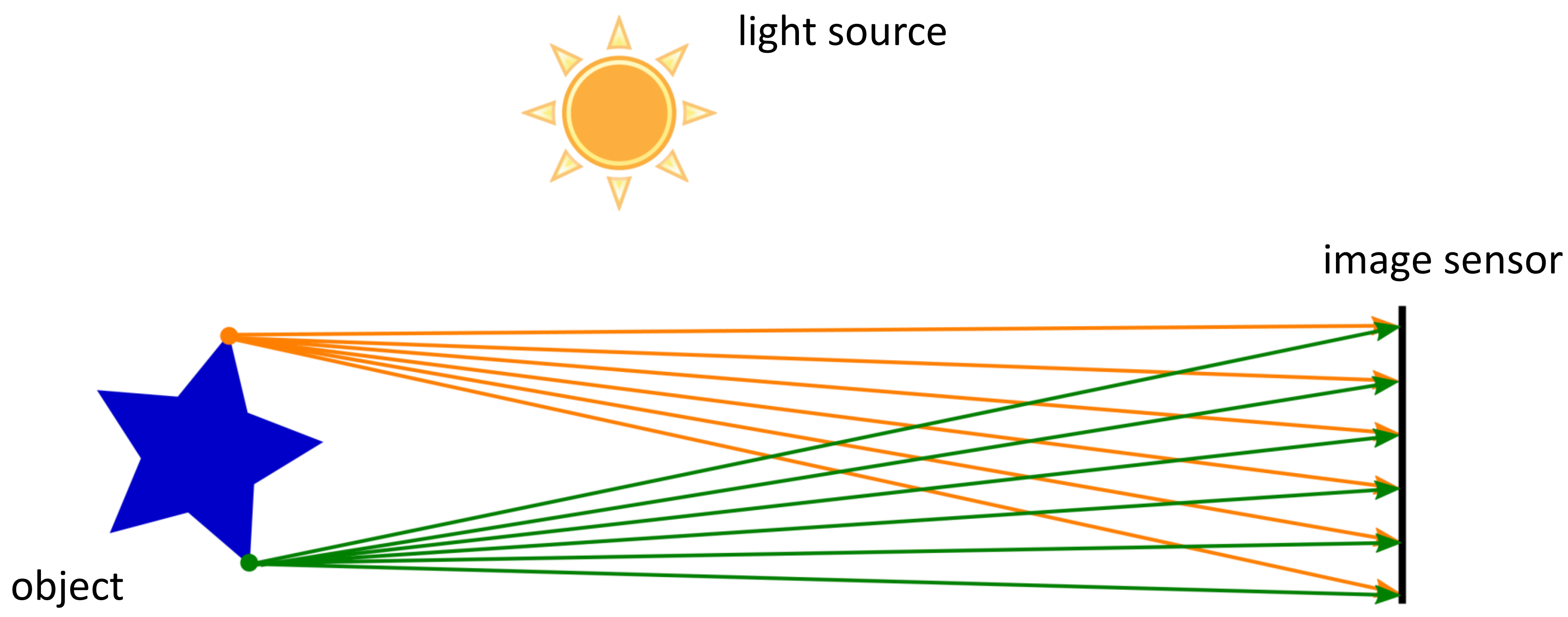
- **Image formation**
  - Pinhole camera
  - Lenses, thin lens equation, pinhole approximation
  - Focus, depth of field, field of view
  - Digital cameras
  - Camera response function and vignetting
  - Pinhole projection and intrinsic camera parameters
  - Lens distortion
- Multiple view geometry basics
  - Camera extrinsics
  - Epipolar geometry

How to Capture an Image?



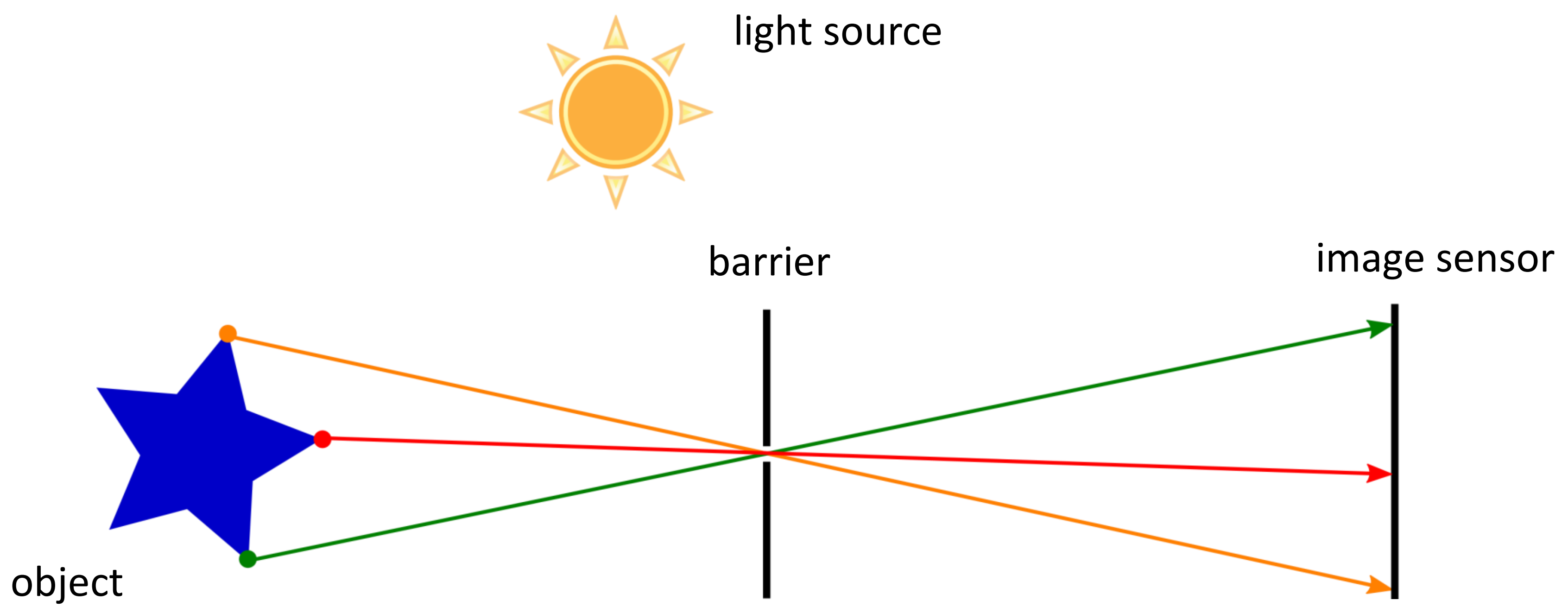
- Lambertian reflectance: object reflects light with a constant brightness at any angle

How to Capture an Image?



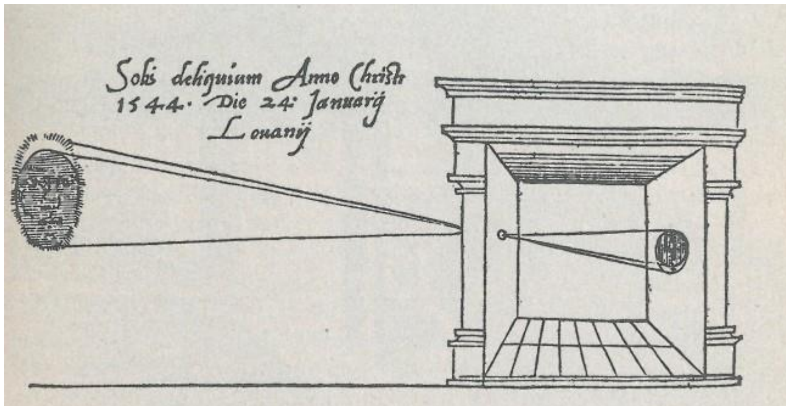
- What if we place an image sensor in front of the object?
- A pixel receives a mixture of light from visible object points
- Strong blur! We don't get a useful image

How to Capture an Image?



- Let's place a barrier with an aperture between object and sensor
- Sensor receives light from a small set of rays
- Blur is reduced

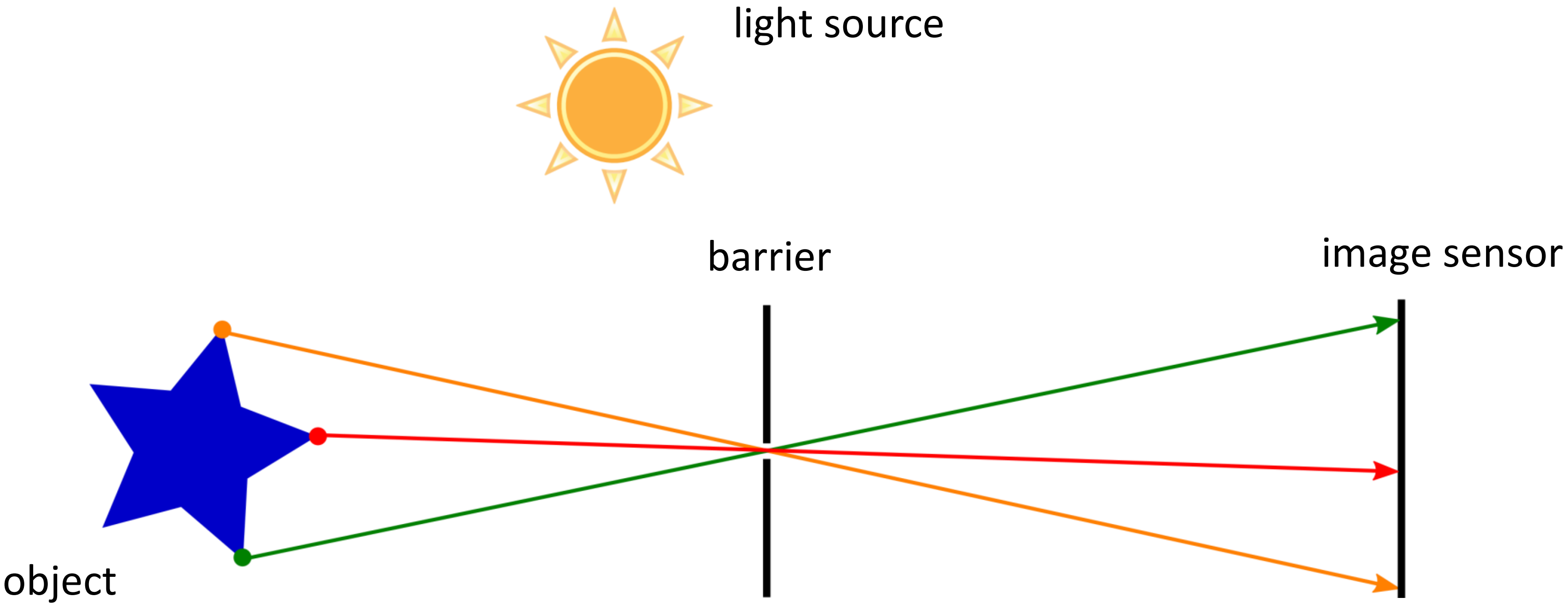
How to Capture an Image?



Camera obscura (lat., „dark room“)
illustrated by Gemma Frisius 1545

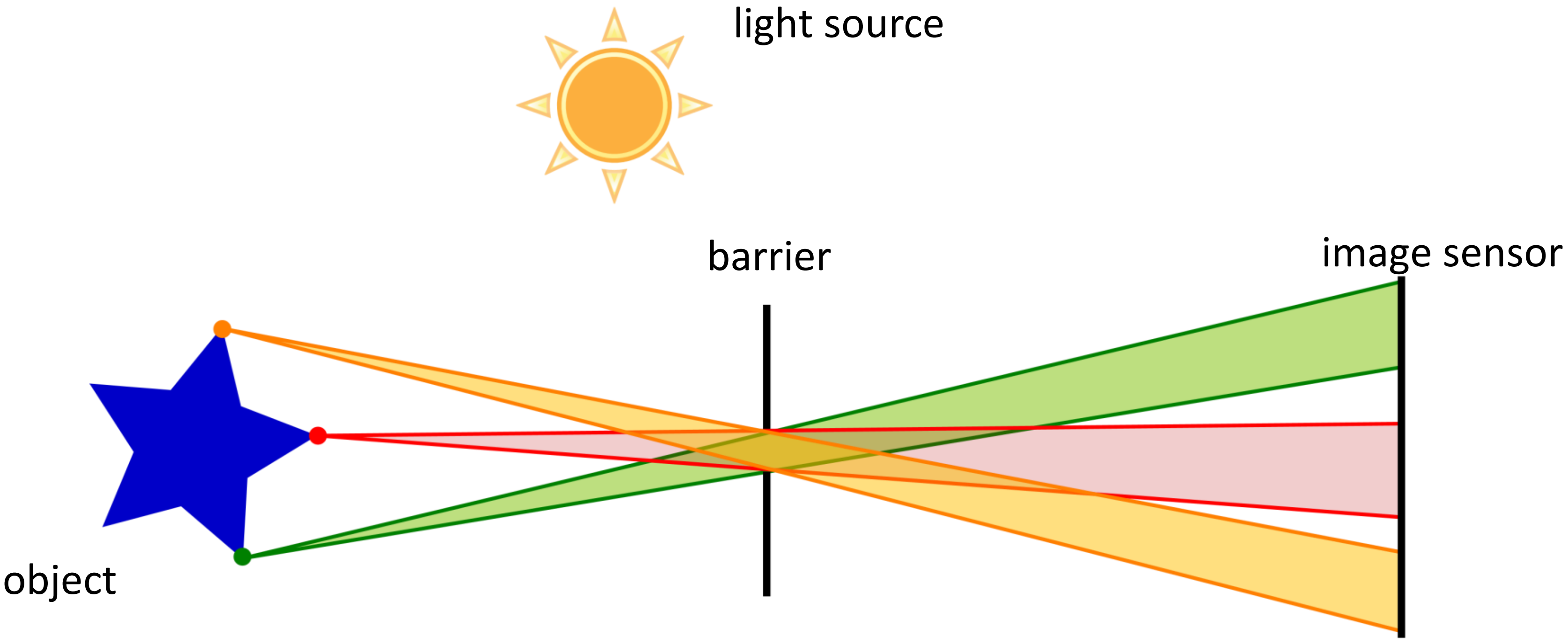
- Observation: Images are still blurry
  - What causes the blur?
  - How can we reduce the blur further?

How to Capture an Image?



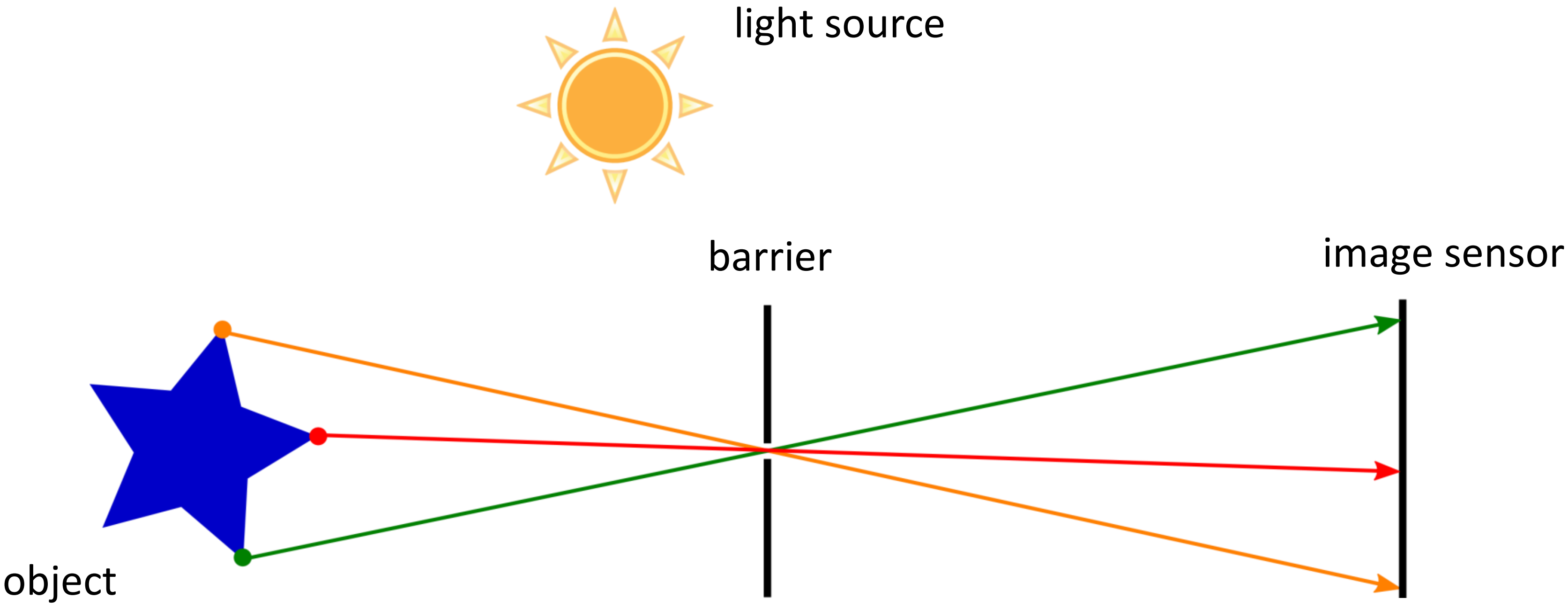
- For an ideal pinhole, only a single ray passes per sensor point
- No blur, but image is dim

How to Capture an Image?



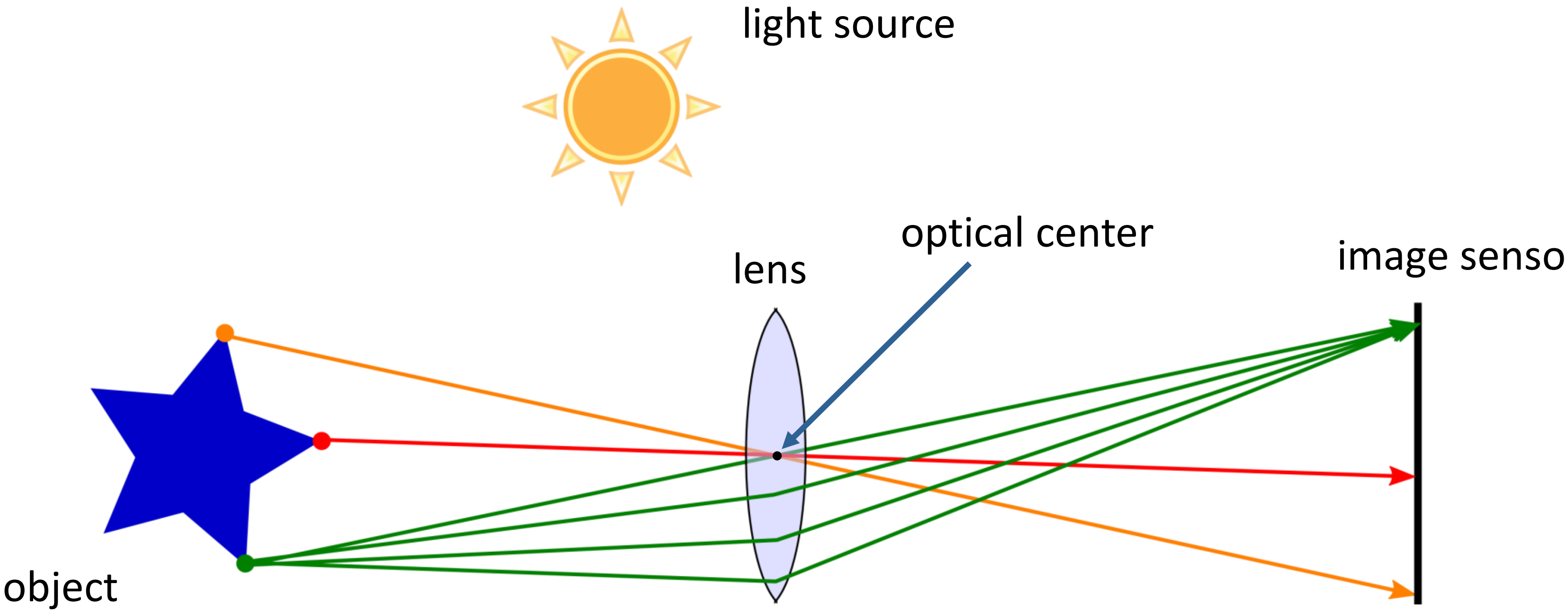
- The larger the aperture, the more light arrives at sensor
- The larger the aperture, the blurry the image

How to Capture an Image?



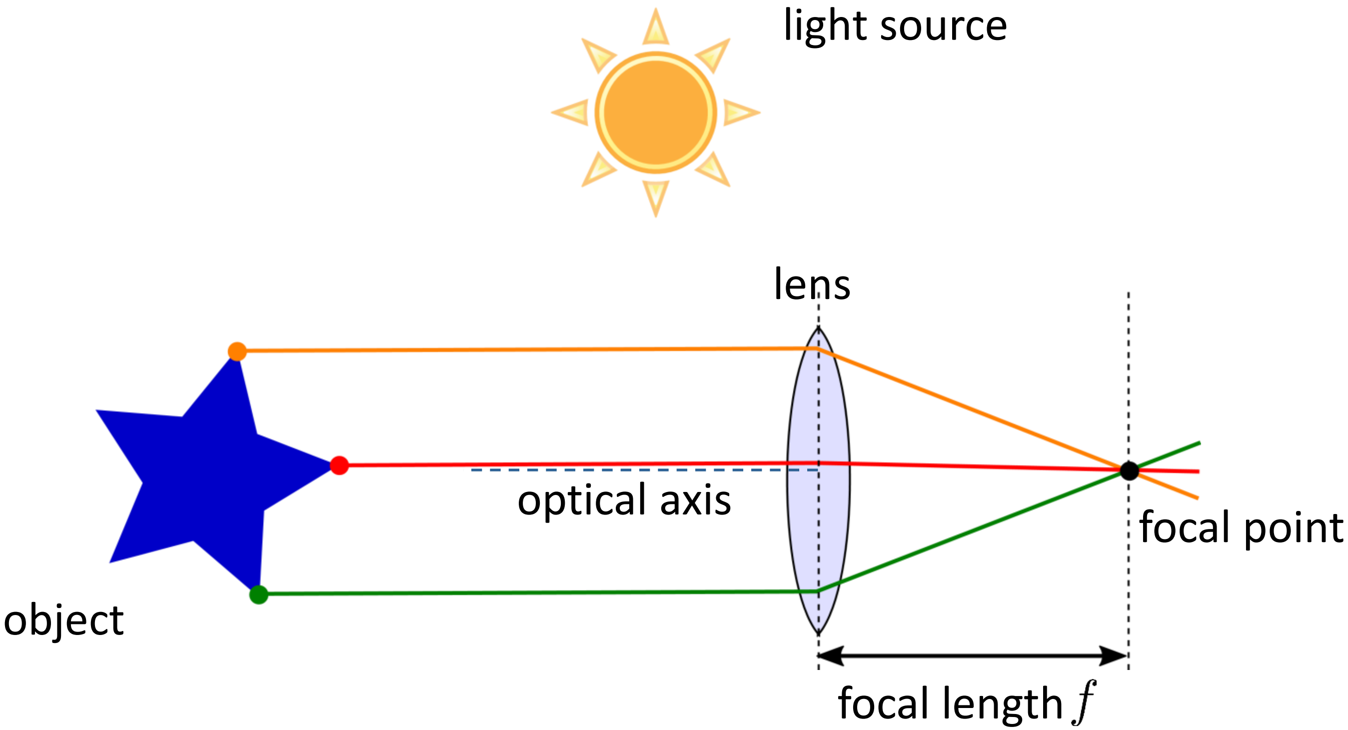
- How can we increase the collected light for small aperture?
  - We can increase the exposure time!
  - Disadvantage: motion blur increases with exposure time
- Diffraction limits the aperture size from below

Converging Lenses



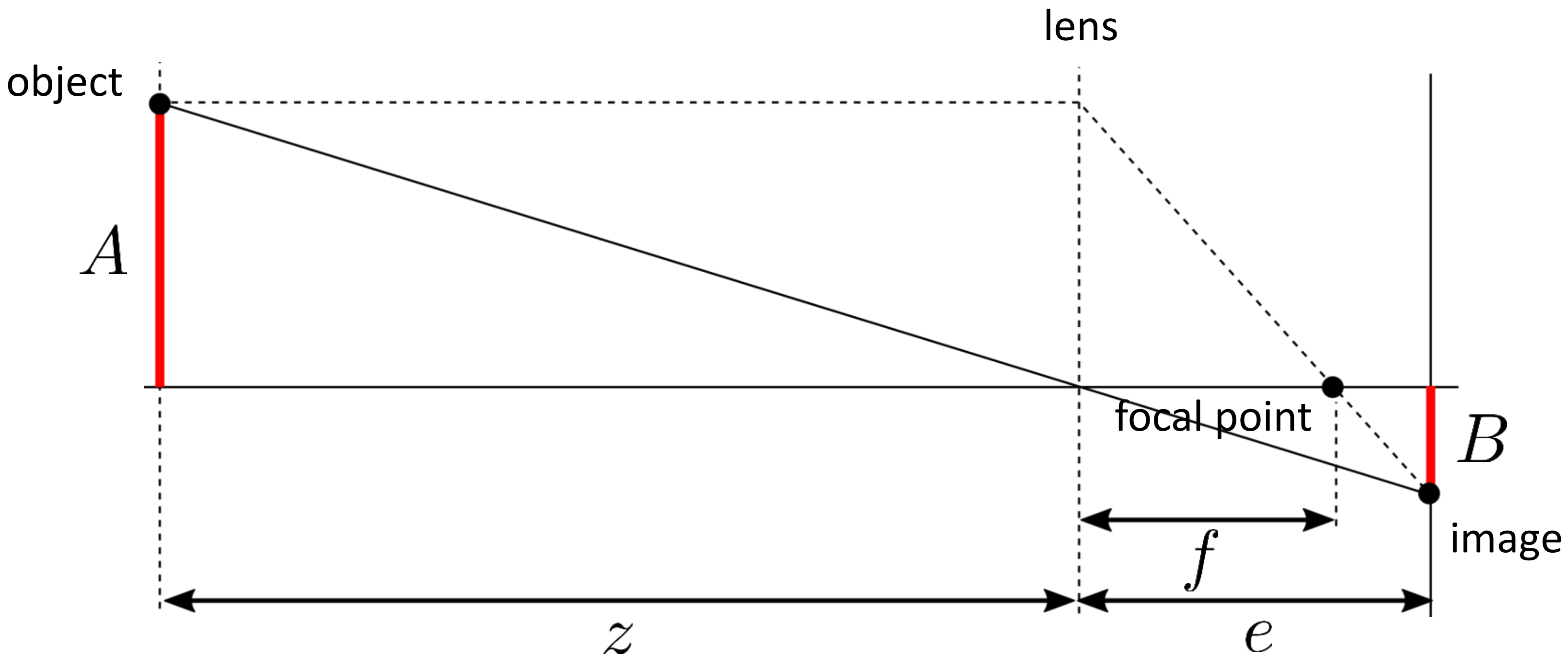
- New idea: use a lens to focus rays from the same object point on the sensor
- Rays go straight through the lens' optical center
  - Central ray

Focal Point



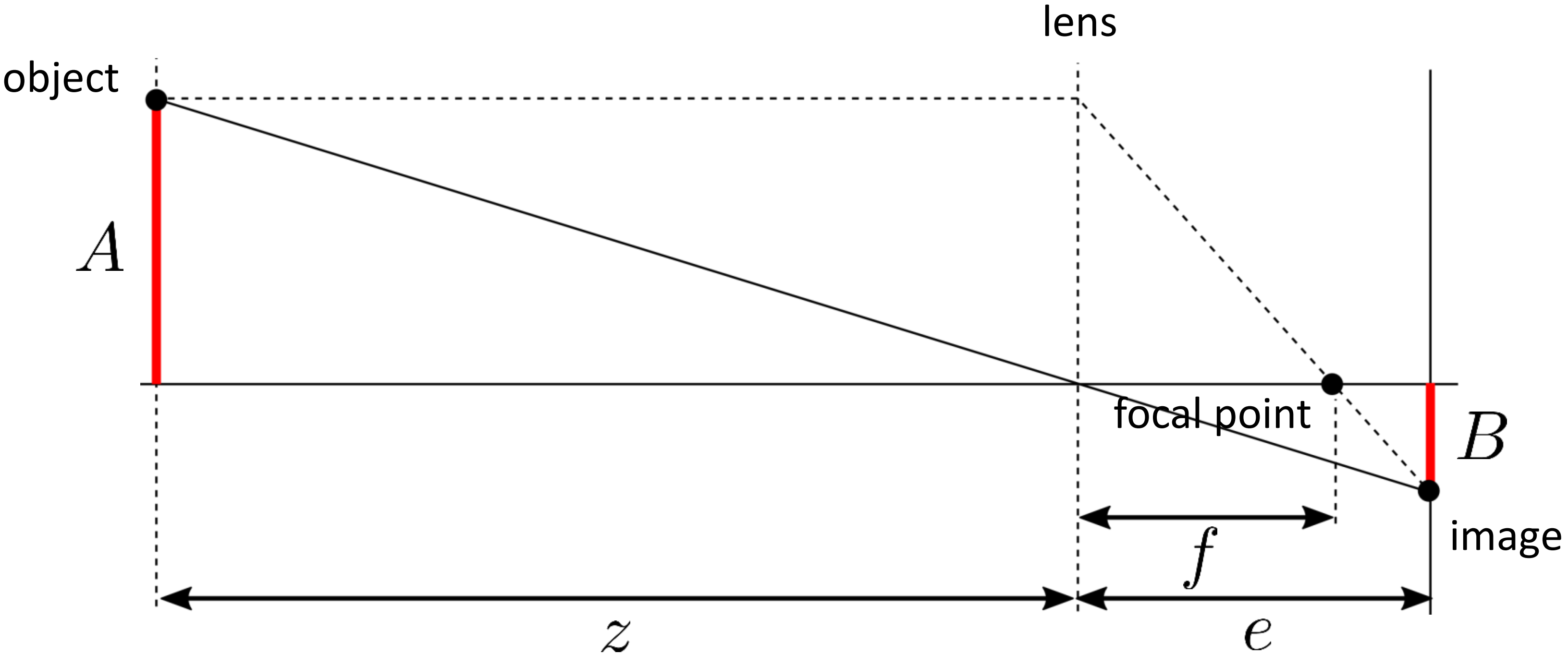
- Rays parallel to the optical axis of the lens converge at the focal point

Thin Lens Equation



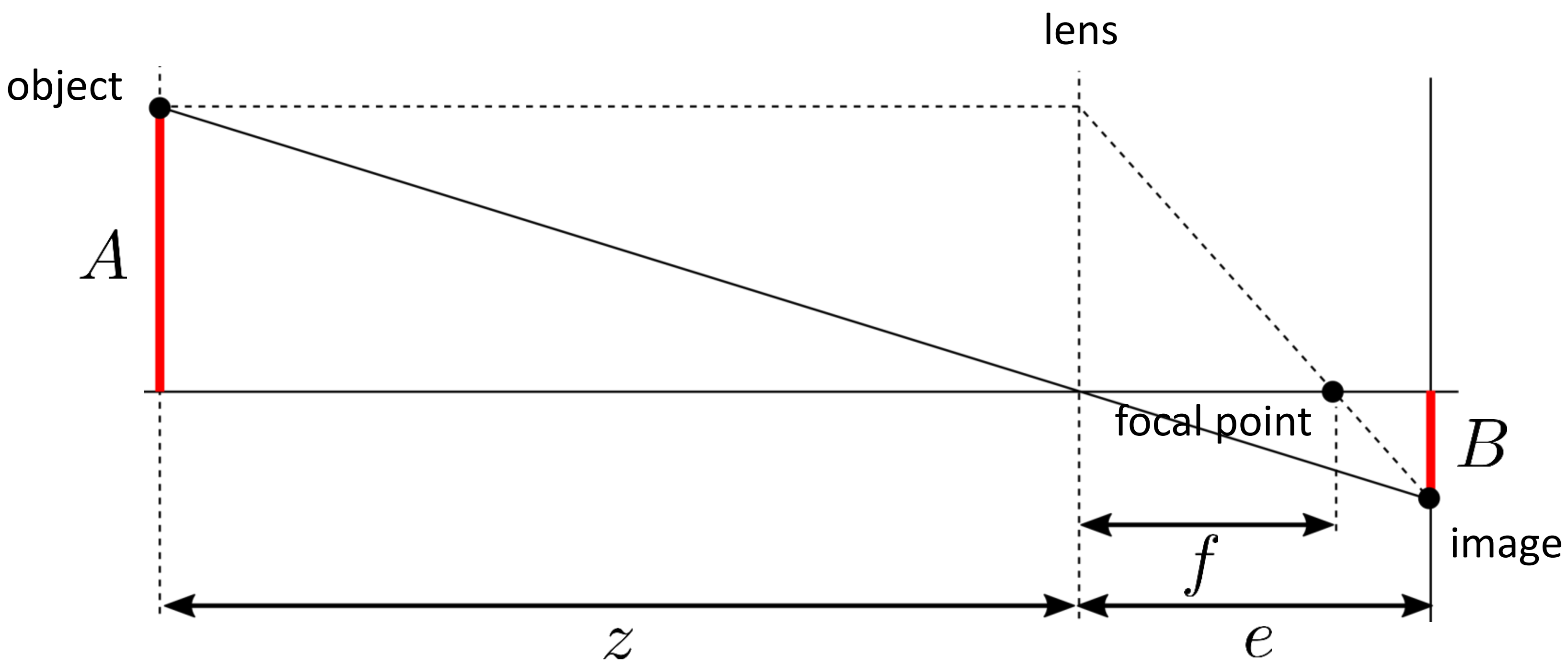
- Relationship f, z, e?

Thin Lens Equation



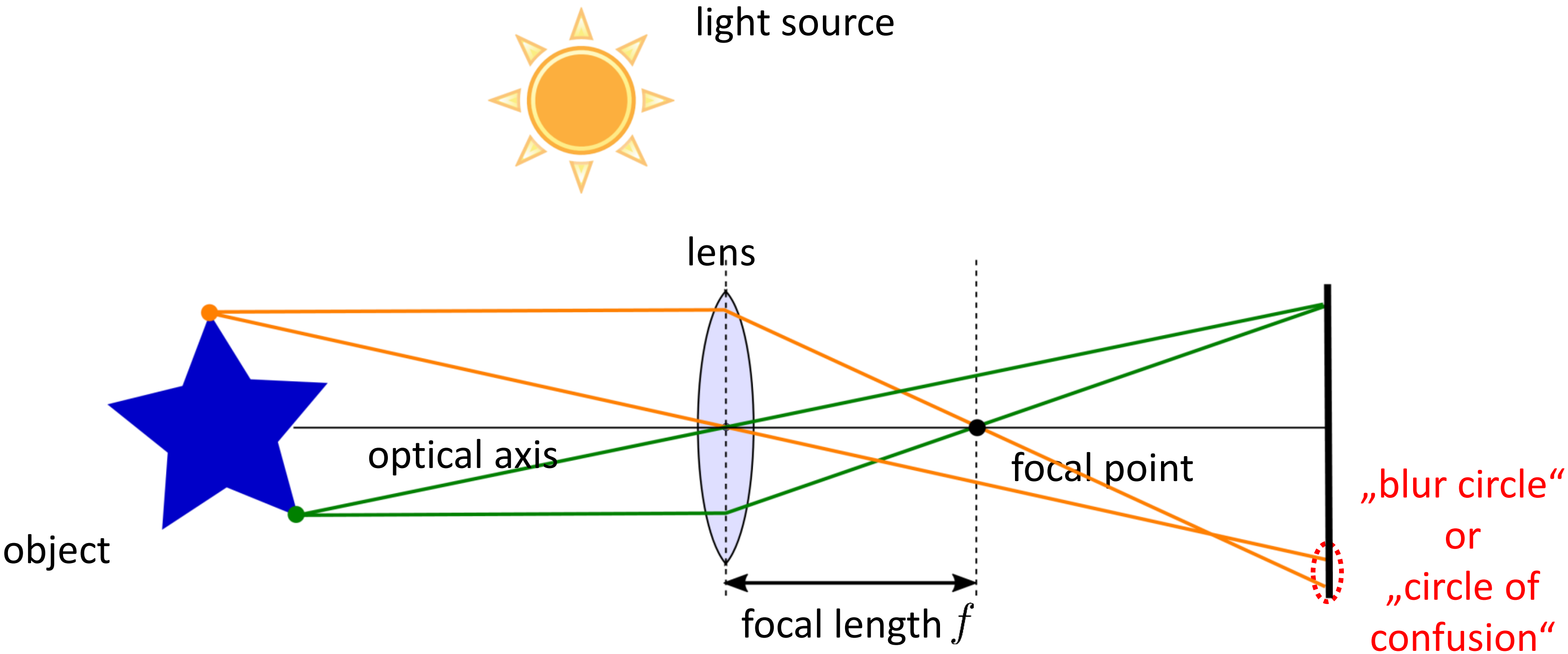
$$\left.\begin{aligned} \frac{B}{A} &= \frac{e}{z} \\ \frac{B}{A} &= \frac{e-f}{f} = \frac{e}{f} - 1 \end{aligned}\right\} \quad \frac{e}{z} = \frac{e}{f} - 1 \Leftrightarrow \frac{1}{f} = \frac{1}{z} + \frac{1}{e}$$

Thin Lens Equation



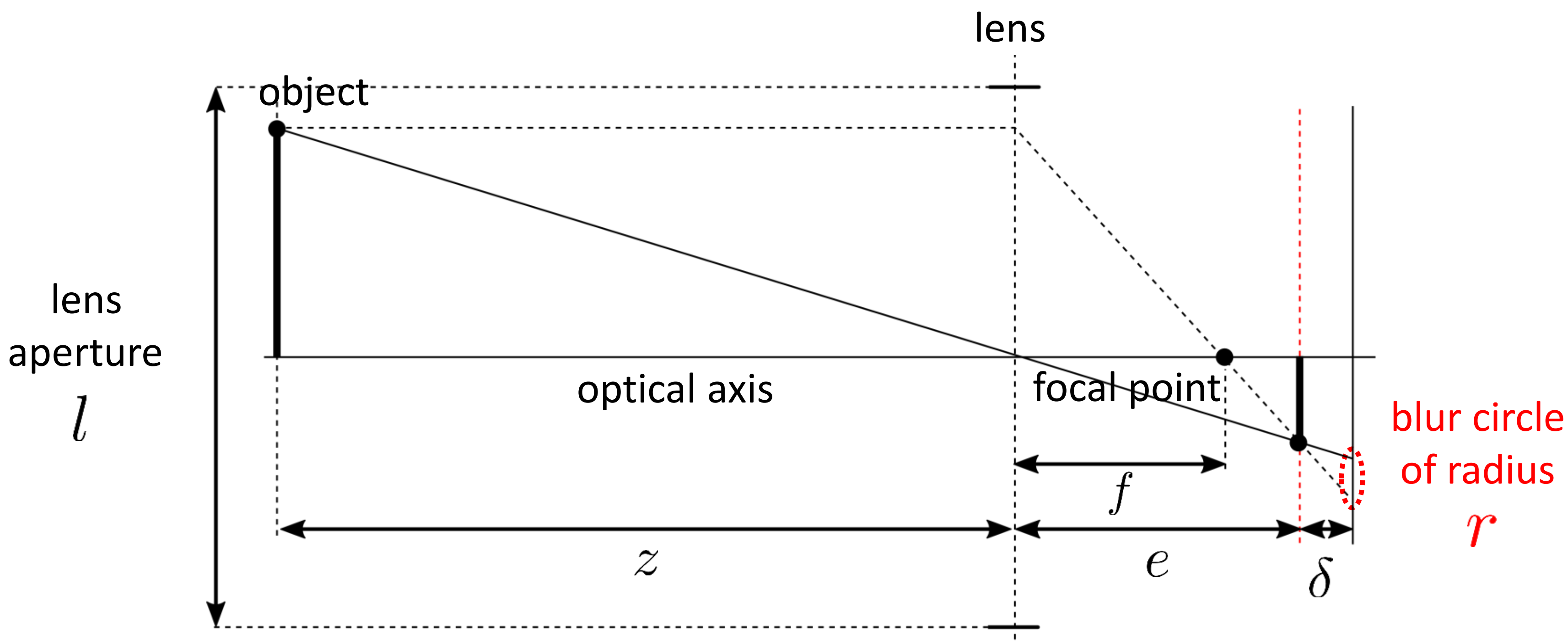
- Thin lens equation: $\frac{1}{f} = \frac{1}{z} + \frac{1}{e}$
- Objects satisfying this equation appear in focus on the image

Points in Focus



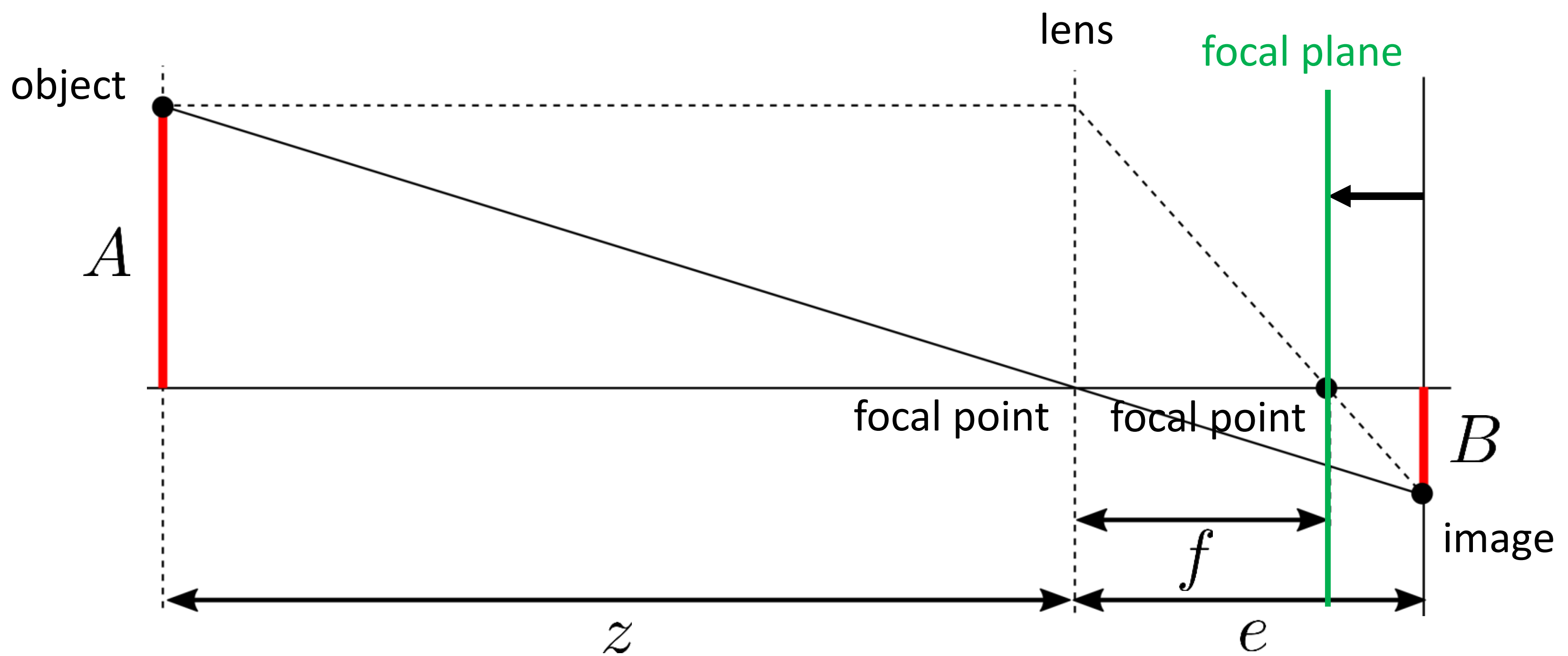
- Objects are in focus at a specific distance from the lens along the optical axis (i.e. depth)
- At other distances, objects project to a “blur circle” on image

Blur Circle



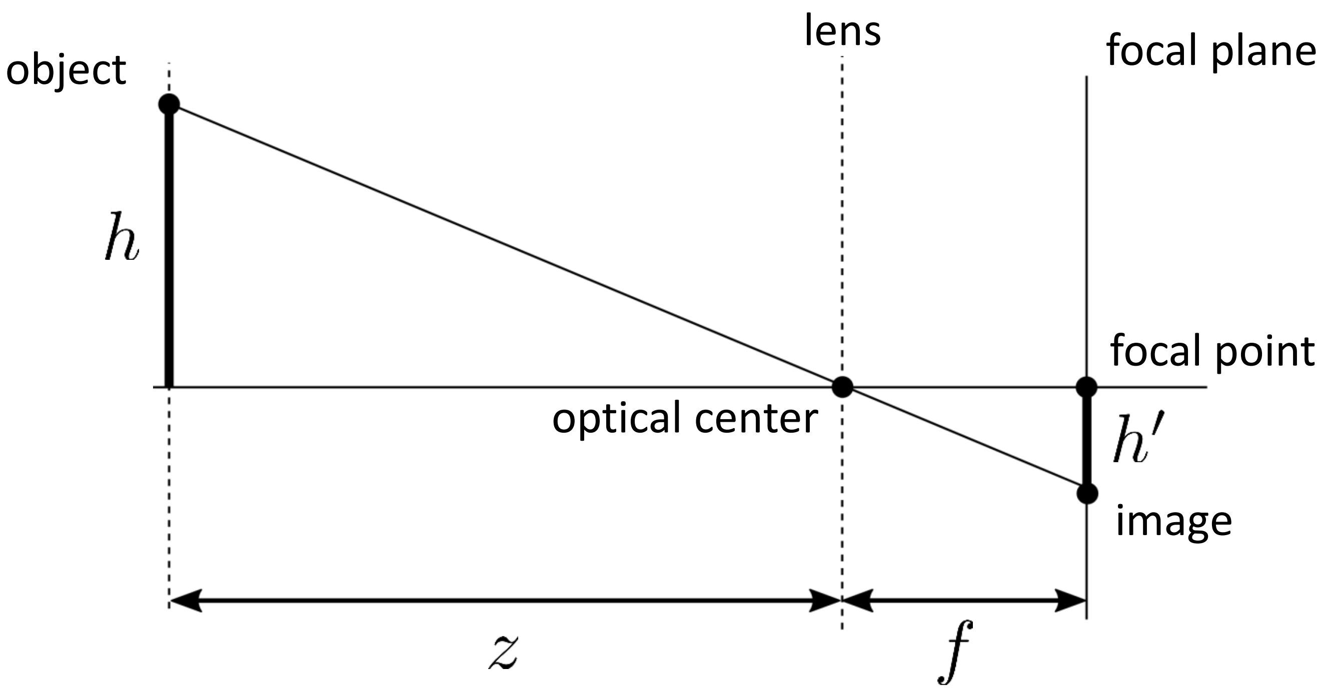
- Object out of focus: blur circle has radius $r = \frac{l\delta}{2e}$
  - Infinitesimally small aperture gives minimal radius
  - "Good image": adjust camera settings to achieve smaller radius than pixel size

Pinhole Approximation



- What happens for $z \gg f$?
  - For $z \rightarrow \infty$, we obtain $\frac{1}{f} = \frac{1}{z} + \frac{1}{e} \approx \frac{1}{e} \Rightarrow f \approx e$
  - Image plane needs to be adjusted towards focal plane for focus

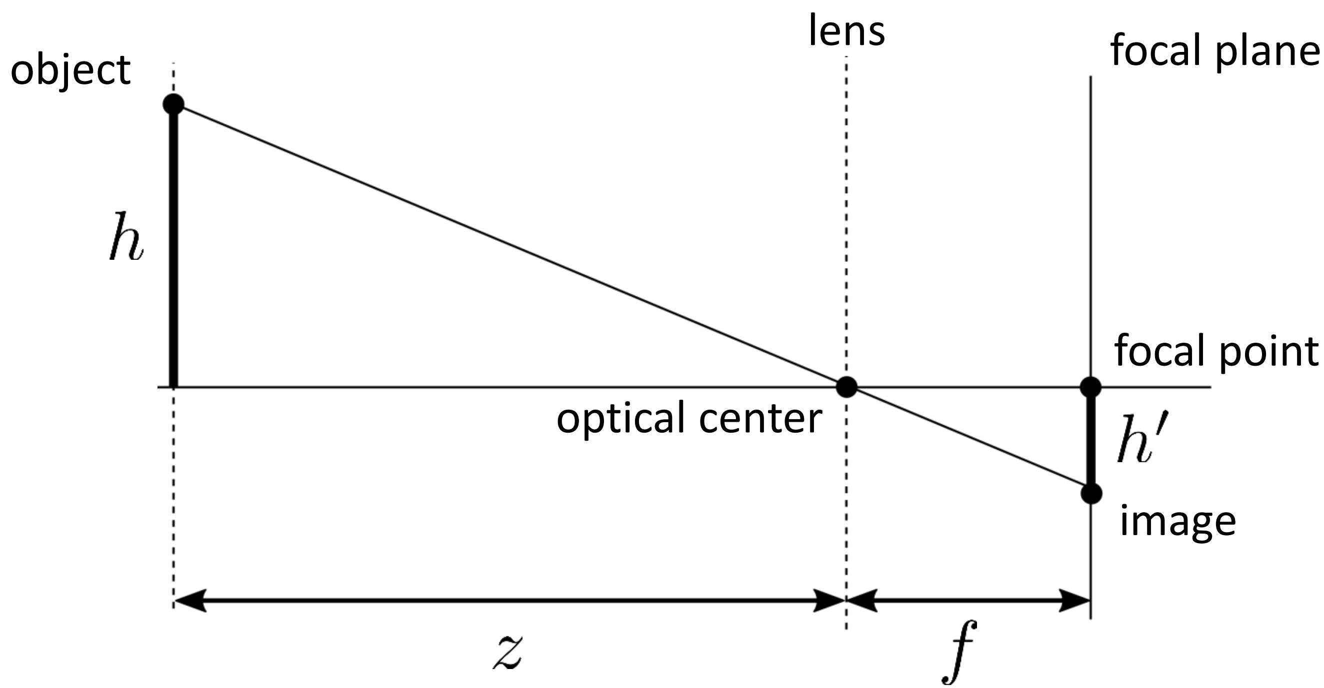
Pinhole Approximation



- In the limit (focus at infinity): image plane at focal plane
- Object point at h projects to image according to

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

Pinhole Approximation



- Pinhole approximation holds also for closed focus points
  - However, only in a very limited range (Depth of Field)
  - Pinhole focal length ≠ thin lens focal length

Perspective Effects

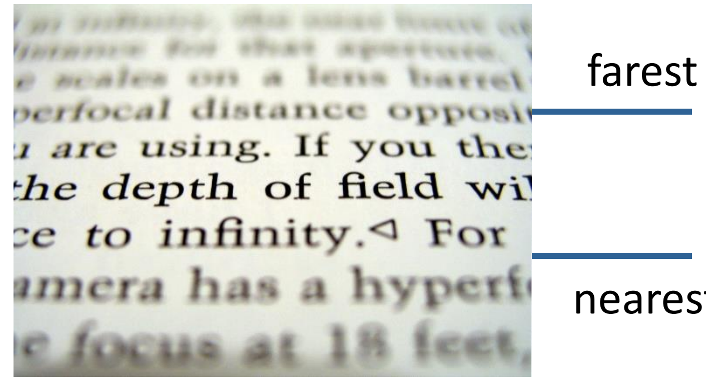
- More distant objects appear smaller in the image
- Ratio between object and image size directly relates to object distance

Depth of Field

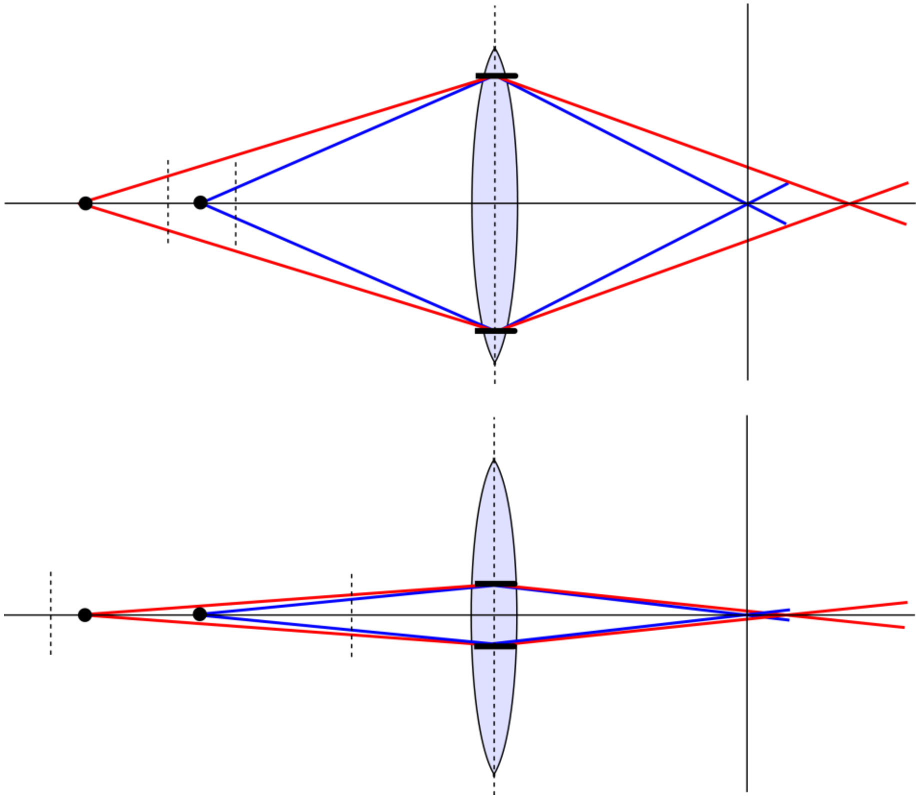
- Depth of Field: Depth of nearest and farthest object that appear acceptably sharp in image

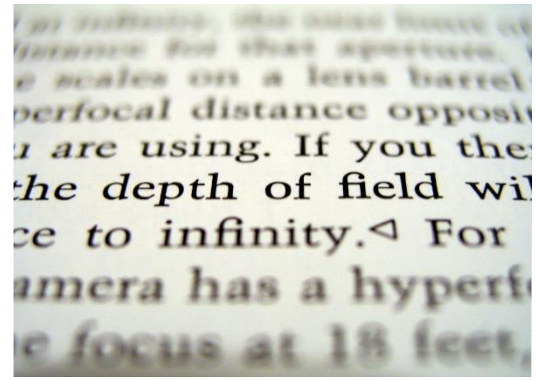
- Lens only precisely focuses on a single depth

- Blur circle increases gradually with depth



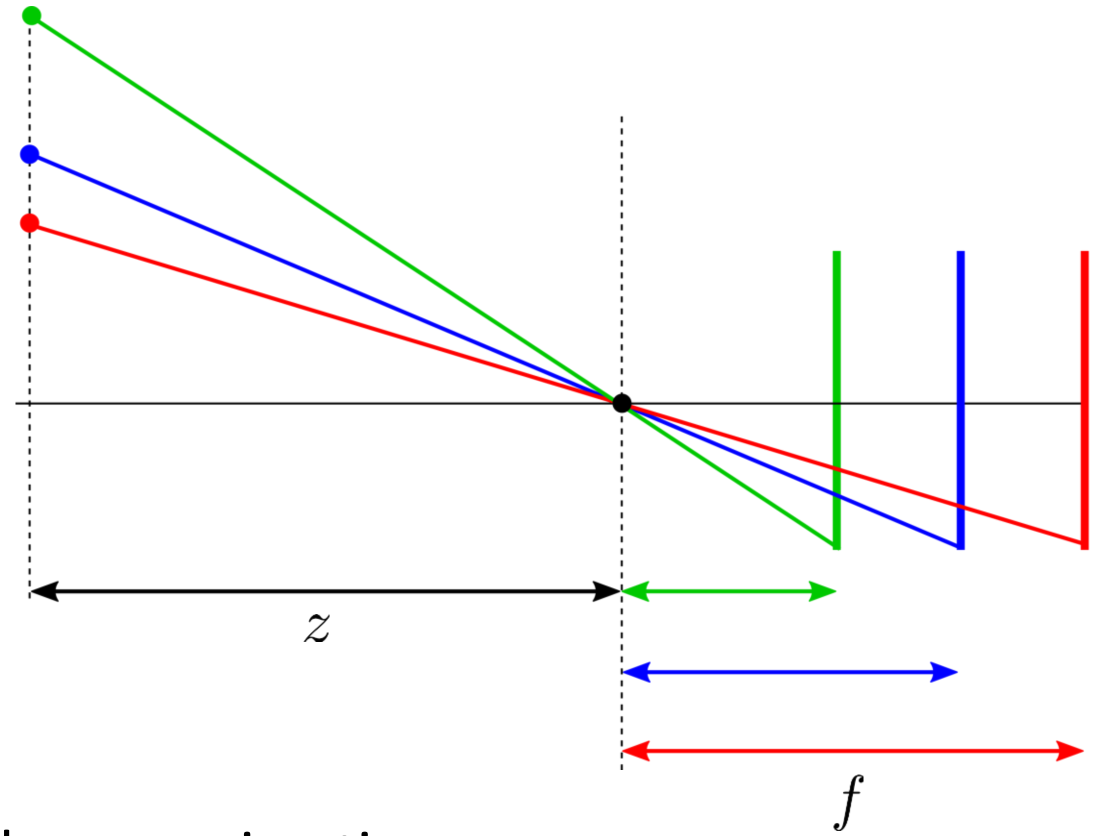
Depth of Field





- The smaller the lens aperture …
  - the larger the depth of field
  - the less light reaches the sensor in a given exposure time

Field of View



- Pinhole approximation
- The smaller f, the larger the maximum view angle
- focal length together with sensor size defines field of view

Field of View


28 mm lens, 65.5° × 46.4°

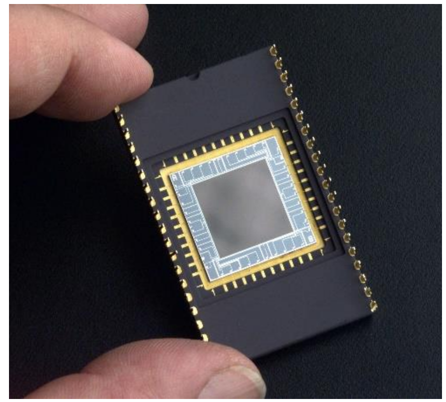

50 mm lens, 39.6° × 27.0°


70 mm lens, 28.9° × 19.5°


210 mm lens, 9.8° × 6.5°

- Choose lens with appropriate focal length for application

Digital Cameras



- Image sensor: array of light-sensitive semi-conducter pixels
- CCD (charge coupled device) or CMOS (complementary metal-oxide-semiconductor) technology
- Pixel: photosensitive diode
  - converts photons (light energy) to electrons
- Optical lens mounted on top of image sensor

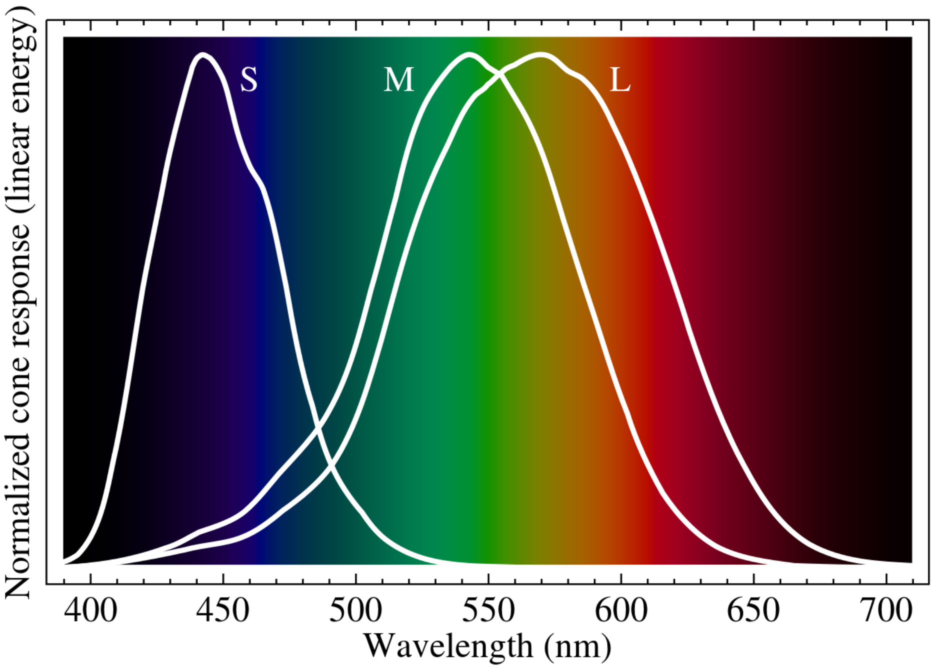
Digital Image

- Digital image is
  an array of D-dim. pixel
  values (RGB values)

- We will also denote an image by a function $I : \Omega \rightarrow \mathbb{R}^D$
  that maps pixels on a continuous image domain $\Omega \subset \mathbb{R}^2$
  to their D-dim. values

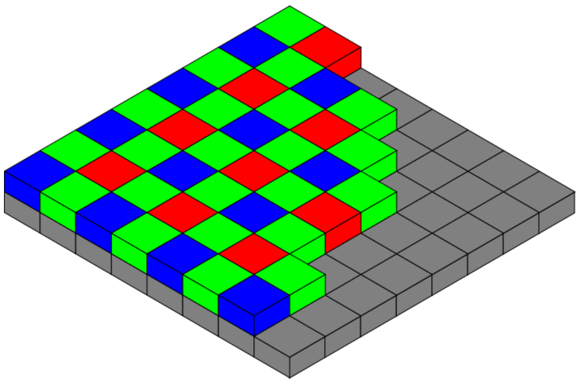
Color Vision

- For humans luminance is mainly perceived from green color
- Human visual system much more sensitive to high frequencies in luminance than in chrominance
- Spectral sensitivity of human cone cells



Bayer Pattern

- Bayer pattern (introduced by Bryce Bayer in 1967) arranges red, green, blue sensitive pixels
  - Half the pixels measure green light spectrum in a checkerboard pattern
  - Other pixels are sensitive to red or blue alternatingly



- “Demosaicing” to obtain RGB-value at each pixel
  - Interpolation of missing pixel colors based on neighboring pixels

Chromatic Aberration and Fringing

- Lenses may focus light of differing wavelengths to different focal points
- This leads to chromatic aberration (“purple fringing”)
- Other sources of fringing:
  - Lens flare
  - Different sensitivity to colors
  - Bayer pattern demosaicing algorithm

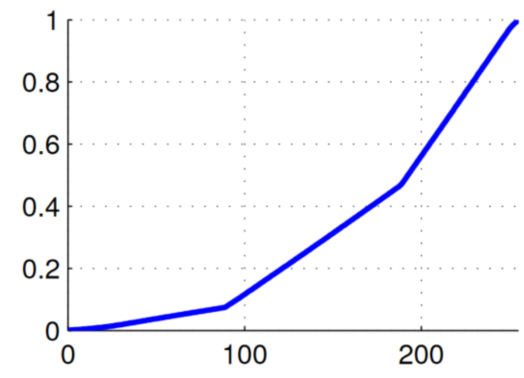
Global vs. Rolling Shutter

- Rolling shutter: Line-by-line exposure/readout of pixels
  - Causes distortions of objects that are in relative motion
- Global shutter: All pixels are exposed/read out at the same time

Camera Response Function

- The objects in the scene radiate light which is focused by the lens onto the image sensor
- The pixels of the sensor observe an irradiance $B : \Omega \rightarrow \mathbb{R}$ for an exposure time t
- The camera electronics translates the accumulated irradiance into intensity values according to a non-linear camera response function $G : \mathbb{R} \rightarrow [0, 255]$

example inv. G



- The measured intensity is $I(\mathbf{x}) = G(tB(\mathbf{x}))$

Vignetting

uncorrected

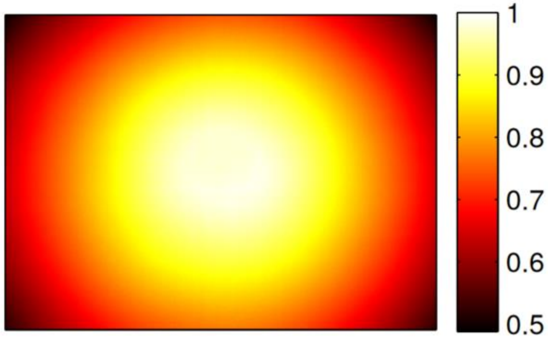
corrected

- Lenses gradually focus more light at the center of the image than at the image borders
- The image appears darker towards the borders
- Also called “lens attenuation”
- Lense vignetting can be modelled as a map $V : \Omega \to [0, 1]$

- Intensity measurement model

$$I(\mathbf{x}) = G(tV(\mathbf{x})B(\mathbf{x}))$$

$V(\mathbf{x})$



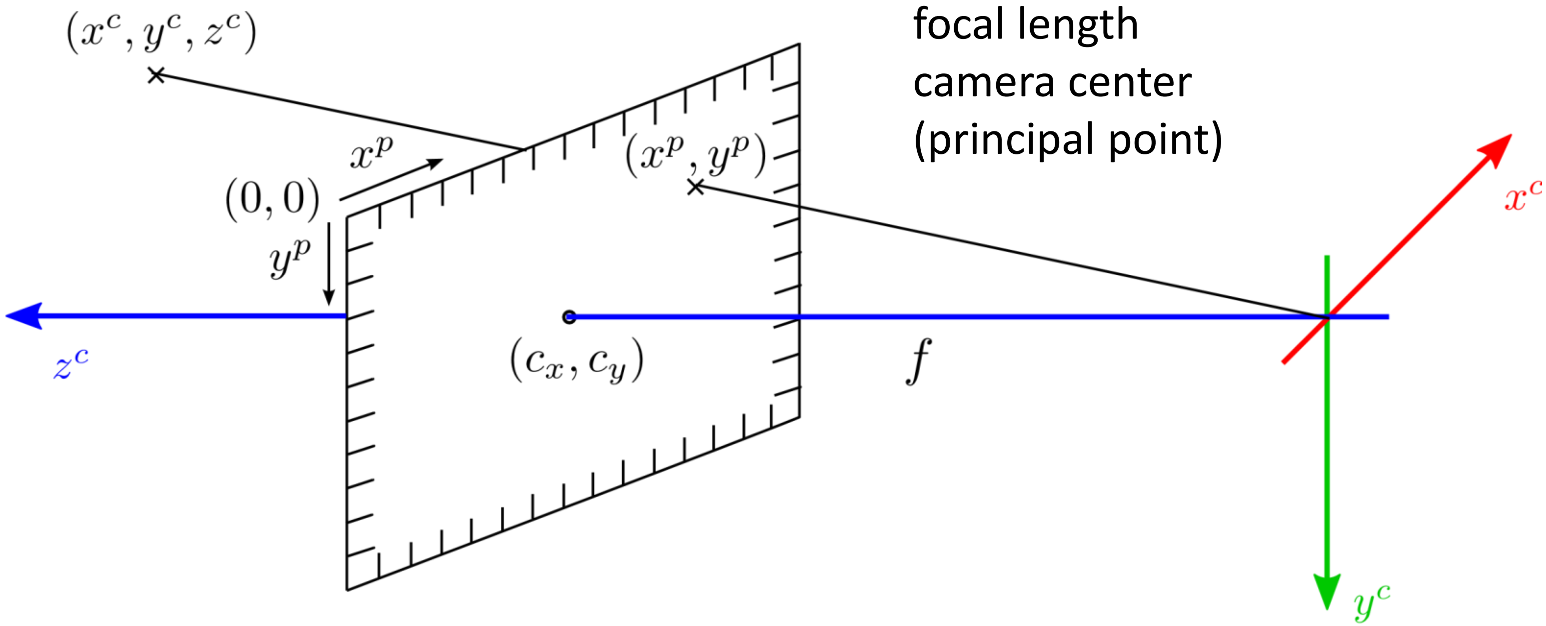
Geometric Point Primitives

	2D	3D
• Point	$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$	$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3$
• Augmented vector	$\bar{\mathbf{x}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{R}^3$	$\bar{\mathbf{x}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \in \mathbb{R}^4$
• Homogeneous coordinates	$\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{pmatrix} \in \mathbb{P}^2$	$\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{pmatrix} \in \mathbb{P}^3$

$$\tilde{\mathbf{x}} = \tilde{w}\bar{\mathbf{x}}$$

Pinhole Camera Model

world coordinates	$\bar{\mathbf{x}} = (x^c, y^c, z^c, 1)^\top$
image pixel coordinates	$\bar{\mathbf{y}}^p = (x^p, y^p, 1)^\top$
focal length	f
camera center (principal point)	c_x, c_y



$$\begin{pmatrix} x^p \\ y^p \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\substack{C \\ \text{(camera matrix)}}} \underbrace{\begin{pmatrix} x^c/z^c \\ y^c/z^c \\ 1 \end{pmatrix}}_{\substack{\bar{y} \\ \text{(normalized image coordinates)}}}$$

Pinhole Camera Model

$$\begin{pmatrix} x^p \\ y^p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^c/z^c \\ y^c/z^c \\ 1 \end{pmatrix}$$

$$z^c \begin{pmatrix} x^p \\ y^p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \tilde{x}^p \\ \tilde{y}^p \\ \tilde{w}^p \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ 1 \end{pmatrix}$$

$$\tilde{w}^p = z^c$$

Lens Distortion

- Lens imperfections cause radial distortion of image
- Deviations stronger towards the image borders
- Typically compensated using a low-order polynomial, for example,

$$x_d = x_n(1 + \kappa_1 r_n^2 + \kappa_2 r_n^4)$$

$$y_d = y_n(1 + \kappa_1 r_n^2 + \kappa_2 r_n^4)$$

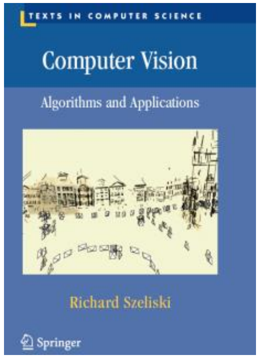
$$(x_n, y_n)^\top := (x_c/z_c, y_c/z_c)^\top$$

$$r_n = \left\| (x_n, y_n)^\top \right\|_2$$

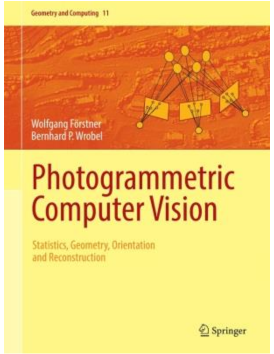
- There are also more complex/complete distortion models

Further Readings

- Further readings on image formation and camera models



Computer Vision – Algorithms and Applications, R. Szeliski, Springer, 2006

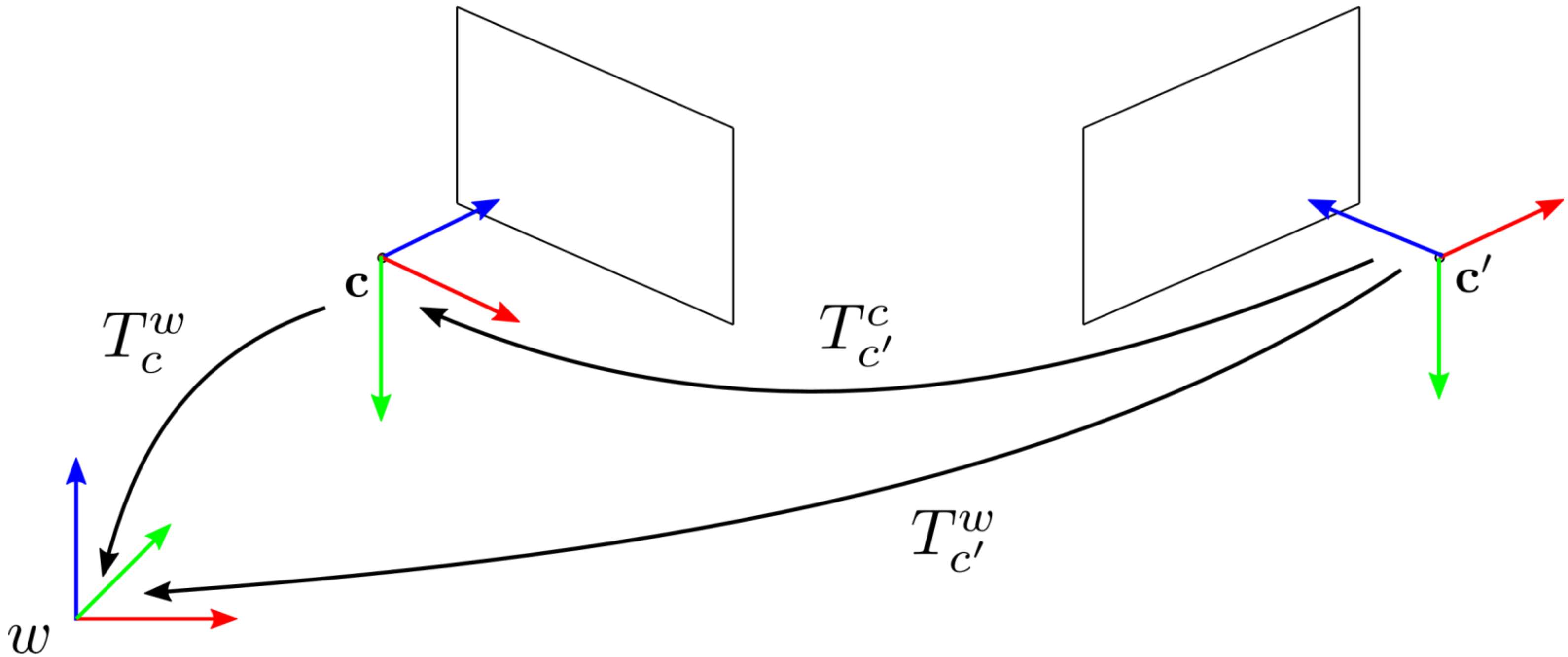


Photogrammetric Computer Vision, W. Förstner, Springer, 2016

What We Will Cover Today

- Image formation
  - Pinhole camera
  - Lenses, thin lens equation, pinhole approximation
  - Focus, depth of field, field of view
  - Digital cameras
  - Camera response function and vignetting
  - Camera intrinsics for pinhole camera model
  - Lens distortion
- **Multiple view geometry basics**
  - Camera extrinsics
  - Epipolar geometry

Camera Extrinsics



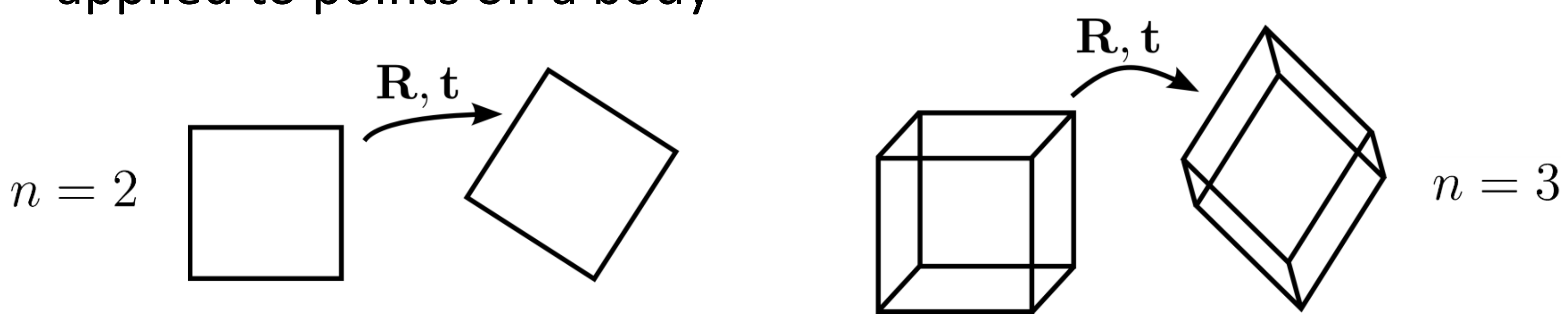
- Euclidean transformations ($T_c^w, T_{c'}^w, T_{c'}^c$) between camera view poses and world frame

(Special) Euclidean Transformations

- (Special) Euclidean transformations apply rotation $\mathbf{R} \in \mathbf{SO}(n) \subset \mathbb{R}^{n\times n}$ and translation $\mathbf{t} \in \mathbb{R}^n$

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t} \qquad \bar{\mathbf{x}}' = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \bar{\mathbf{x}}$$

- Correspond to rigid-body motion
- Rigid-body motion: preserves distances and angles when applied to points on a body



Special Orthogonal Group SO(n)

- Rotation matrices have a special structure

$$\mathbf{R} \in \mathbf{SO}(n) \subset \mathbb{R}^{n \times n}, \det(\mathbf{R}) = 1, \mathbf{R}\mathbf{R}^T = \mathbf{I}$$

  i.e. orthonormal matrices that preserve distance and angle

- They form a group which we denote as Special Orthogonal Group $\mathbf{SO}(n)$
  - The group operator is matrix multiplication - associative, but not commutative!
  - Inverse and neutral element exist

- 2D rotations only have 1 degree of freedom (DoF), i.e. angle of rotation
- 3D rotations have 3 DoFs, several parametrizations exist such as Euler angles and quaternions

3D Rotation Representations – Matrix

- Straight-forward: **Orthonormal matrix**

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

- Pro: Easy to concatenate and invert

$$\mathbf{R}_C^A = \mathbf{R}_B^A \mathbf{R}_C^B \qquad \mathbf{R}_A^B = \left(\mathbf{R}_B^A\right)^{-1}$$

- Con: Overparametrized (9 parameters for 3 DoF) - problematic for optimization

3D Rotation Representations – Euler Angles

- **Euler Angles**: 3 consecutive rotations around coordinate axes
  Example: roll-pitch-yaw angles α, β, γ (X-Y-Z):

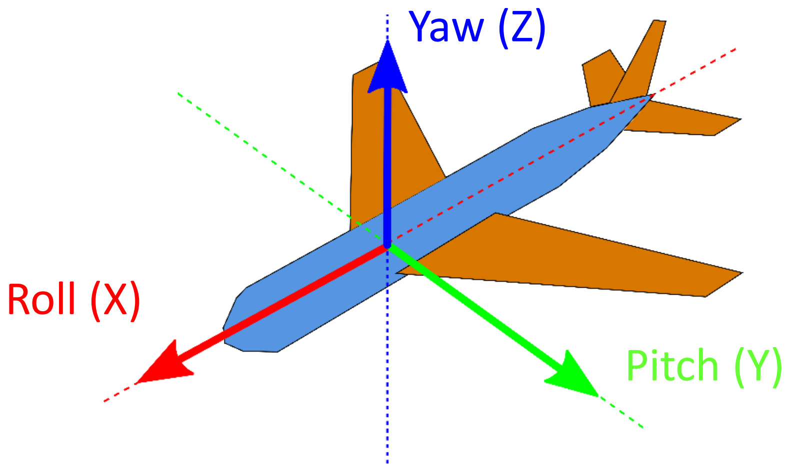
$$\mathbf{R}_{XYZ}(\alpha,\beta,\gamma) = \mathbf{R}_Z(\gamma)\,\mathbf{R}_Y(\beta)\,\mathbf{R}_X(\alpha)$$

  with

$$\mathbf{R}_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$\mathbf{R}_Y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

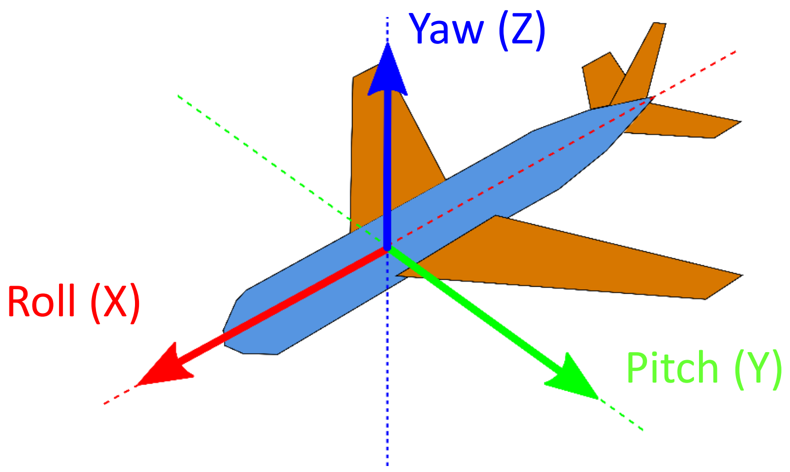
$$\mathbf{R}_Z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



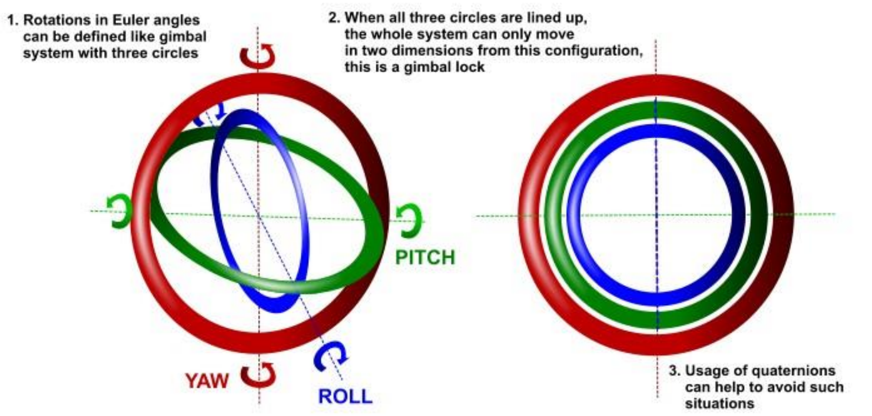
- 12 possible orderings of rotation axes (f.e. Z-X-Z)

3D Rotation Representations – Euler Angles

- Pro: Minimal with 3 parameters



- Con:
  - Singularities (gimbal lock)
  - concatenation/inversion
    via conversion from/to matrix

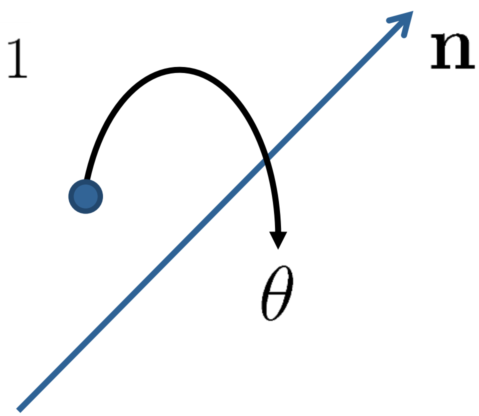


Loss in DoF

3D Rotation Representations – Axis-Angle

- **Axis-Angle:** Rotate along axis $\mathbf{n} \in \mathbb{R}^3$ by angle $\theta \in \mathbb{R}$:

$$\mathbf{R}(\mathbf{n}, \theta) = \mathbf{I} + \sin(\theta)\widehat{\mathbf{n}} + (1 - \cos(\theta))\widehat{\mathbf{n}}^2 \quad \|\mathbf{n}\|_2 = 1$$



where $$\widehat{\mathbf{x}} := \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \quad \widehat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$$

- Reverse: $\theta = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right)$ $\quad \mathbf{n} = \frac{1}{2\sin(\theta)}\begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$

- 4 parameters: $(\mathbf{n}, \theta)$
- 3 parameters: $\omega = \theta\mathbf{n}$

3D Rotation Representations – Axis-Angle

- Pro: minimal representation for 3 parameters

- Con:
  - $(\mathbf{n}, \theta)$ has unit norm constraint on $\mathbf{n}$ which can be problematic for optimization
  - both parametrizations not unique
  - concatenation/inversion via $\mathbf{SO}(3)$

3D Rotation Representations – Quaternion

- **Unit Quaternions:** $\mathbf{q} = (q_x, q_y, q_z, q_w)^\top \in \mathbb{R}^4$ **,** $\|\mathbf{q}\|_2 = 1$

- Relation to axis-angle representation:

  - Axis-angle to quaternion:

$$\mathbf{q}(\mathbf{n}, \theta) = \left(\mathbf{n}^\top \sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right)\right)$$

$$\mathbf{n}(\mathbf{q}) = \begin{cases} (q_x, q_y, q_z)^\top / \sin(\theta/2), & \theta \neq 0 \\ \mathbf{0}, & \theta = 0 \end{cases}$$

  - Quaternion to axis-angle: $\theta = 2 \arccos(q_w)$

3D Rotation Representations – Quaternion

- Pros:
  - Unique up to opposing sign $\mathbf{q} = -\mathbf{q}$
  - Direct rotation of a point:

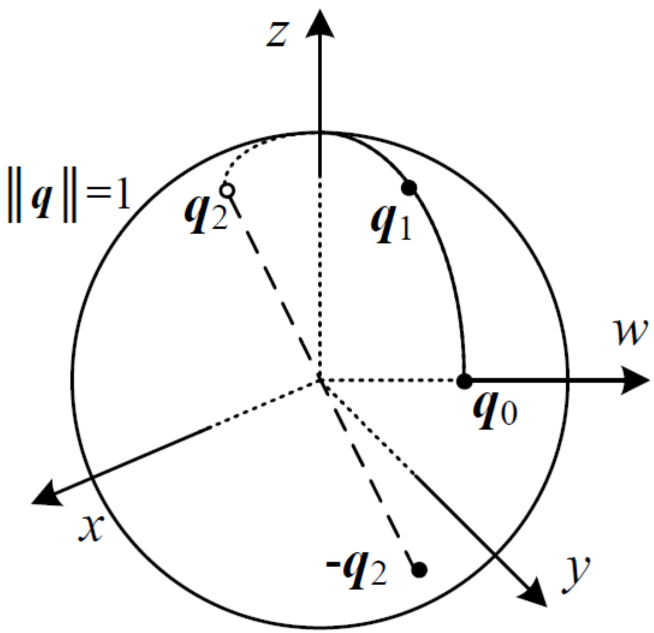
$$\mathbf{p}' = \mathbf{q}(\mathbf{R})\mathbf{p}\mathbf{q}(\mathbf{R})^{-1}$$

  - Direct concatenation of rotations:

$$\mathbf{q}(\mathbf{R}_2\mathbf{R}_1) = \mathbf{q}(\mathbf{R}_2)\mathbf{q}(\mathbf{R}_1)$$

  - Direct inversion of a rotation:

$$\mathbf{q}(\mathbf{R}^{-1}) = \mathbf{q}(\mathbf{R})^{-1}$$



with $\mathbf{q}^{-1} = (-\mathbf{q}_{xyz}^\top, q_w)^\top$, $\mathbf{p} = (\mathbf{p}_{xyz}^\top, 0)^\top$

$$\mathbf{q}_1\mathbf{q}_2 = (q_{1,w}\mathbf{q}_{2,xyz} + q_{2,w}\mathbf{q}_{1,xyz} + \mathbf{q}_{1,xyz} \times \mathbf{q}_{2,xyz}, q_{1,w}q_{2,w} - \mathbf{q}_{1,xyz}\mathbf{q}_{2,xyz})$$

- Con: Normalization constraint is problematic for optimization

Special Euclidean Group SE(3)

- Euclidean transformation matrices have a special structure as well:

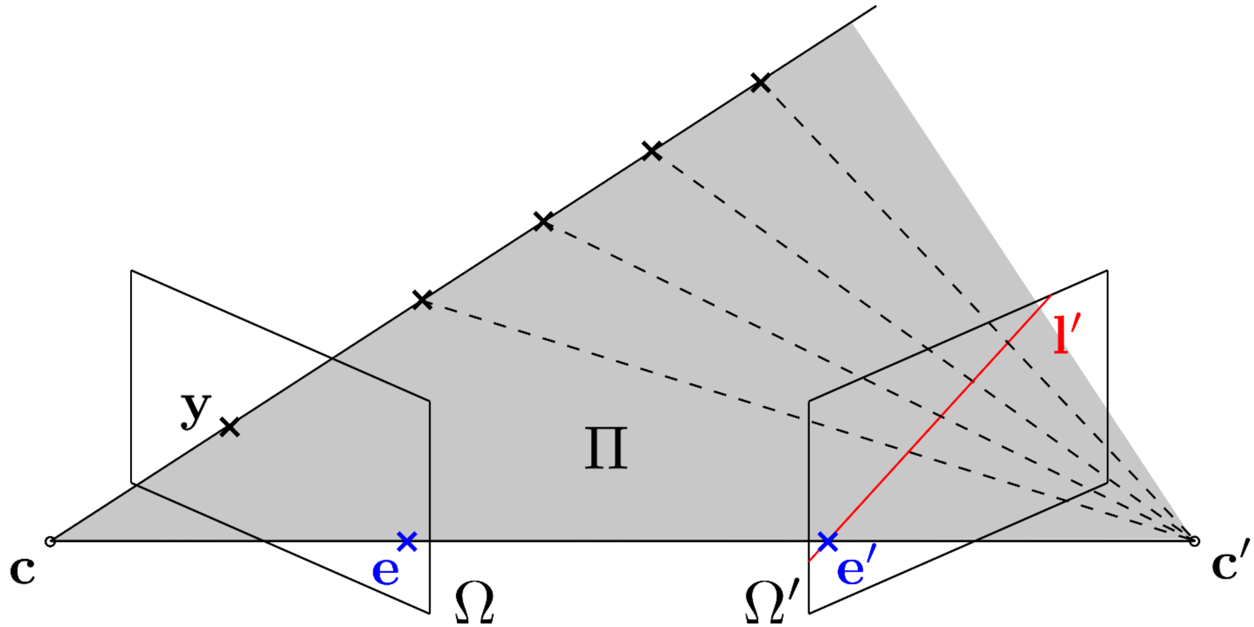
$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbf{SE}(3) \subset \mathbb{R}^{4\times 4}$$

  - Translation $\mathbf{t}$ has 3 degrees of freedom
  - Rotation $\mathbf{R} \in \mathbf{SO}(3)$ has 3 degrees of freedom

- They also form a group which we call $\mathbf{SE}(3)$. The group operator is matrix multiplication:

$$\cdot : \mathbf{SE}(3) \times \mathbf{SE}(3) \rightarrow \mathbf{SE}(3)$$
$$\mathbf{T}_B^A \cdot \mathbf{T}_C^B \mapsto \mathbf{T}_C^A$$

Epipolar Geometry

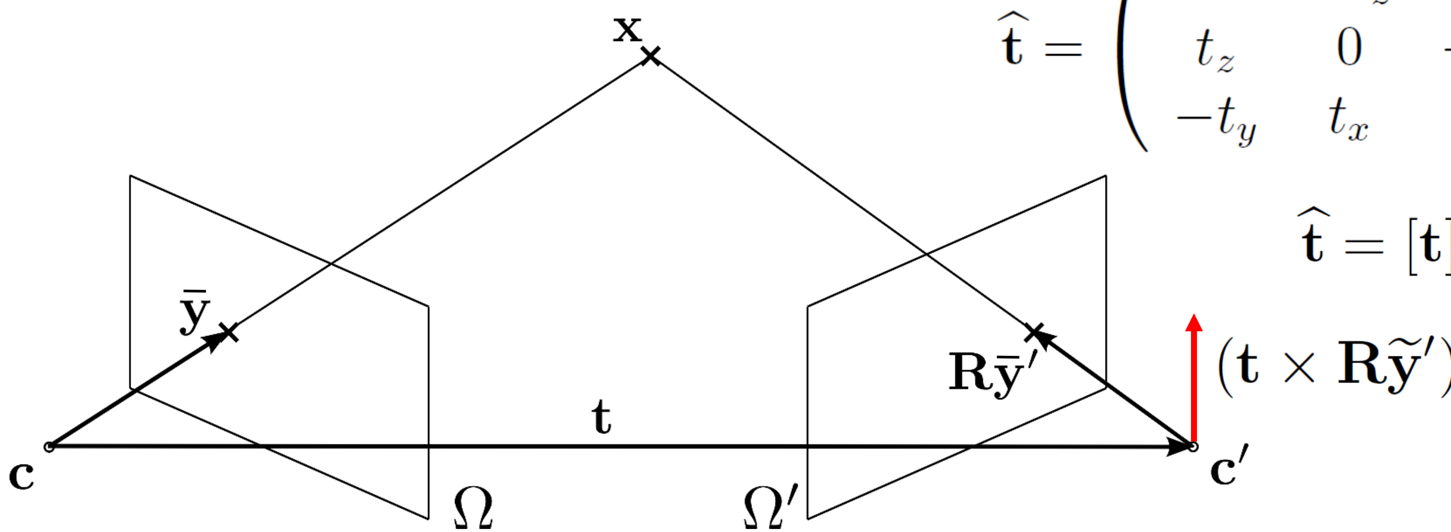


- Camera centers $\mathbf{c}$,$\mathbf{c}'$ and image point $\mathbf{y} \in \Omega$ span the epipolar plane Π
- The ray from camera center $\mathbf{c}$ through point $\mathbf{y}$ projects as the epipolar line $\mathbf{l}'$ in image plane Ω'
- The intersections of the line through the camera centers with the image planes are called epipoles $\mathbf{e}$, $\mathbf{e}'$

Essential Matrix

$$\hat{\mathbf{t}} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

$$\hat{\mathbf{t}} = [\mathbf{t}]_\times$$



- The rays to the 3D point and the baseline $\mathbf{t}$ are coplanar

$$\tilde{\mathbf{y}}^\top (\mathbf{t} \times \mathbf{R}\tilde{\mathbf{y}}') = 0 \Leftrightarrow \tilde{\mathbf{y}}^\top \hat{\mathbf{t}} \mathbf{R} \tilde{\mathbf{y}}' = 0$$

- The essential matrix $\mathbf{E} := \hat{\mathbf{t}}\mathbf{R}$ captures the relative camera pose
- Each point correspondence provides an „epipolar constraint“
- 5 correspondences suffice to determine $\mathbf{E}$ (simpler: 8-point algorithm)

Lessons Learned Today

- Image formation
  - Lenses focus light on image sensor
  - Approximation as pinhole camera
  - Camera settings determine focus, depth of field and field of view
  - Focus, depth of field, field of view
  - Digital cameras transfer irradiance to intensity
  - Lenses are imperfect: radial distortion and vignetting
- 3D rotation representations
- Recap of basic notions of multiple view geometry

Thanks for your attention!

Slides Information

- These slides have been initially created by Jörg Stückler as part of the lecture "Robotic 3D Vision" in winter term 2017/18 at Technical University of Munich.
- The slides have been revised by myself (Niclas Zeller) for the same lecture held in winter term 2020/21
- Acknowledgement of all people that contributed images or video material has been tried (please kindly inform me if such an acknowledgement is missing so it can be added).