Perception II

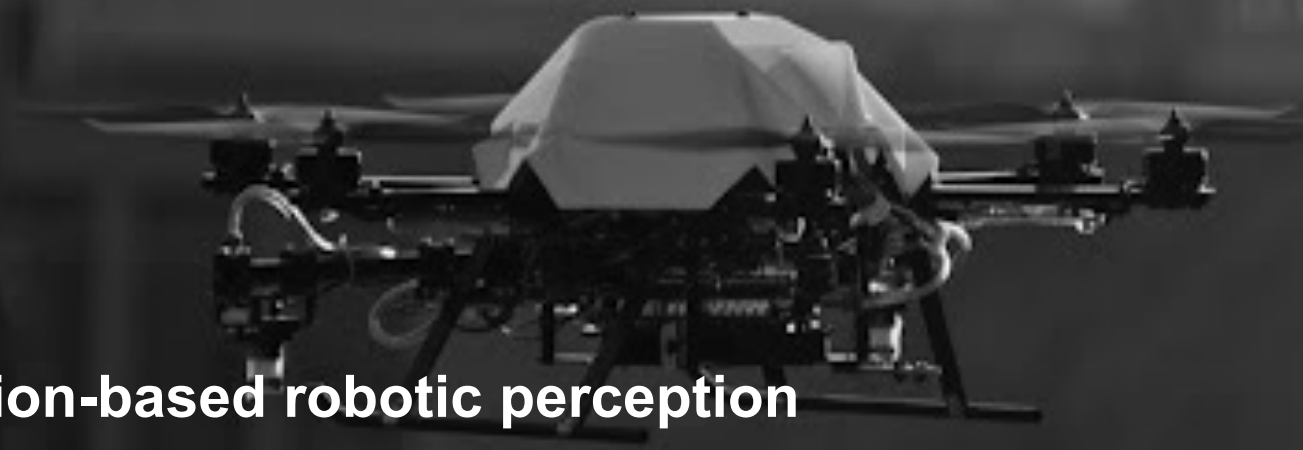
Fundamentals of Computer Vision

Autonomous Mobile Robots

Margarita Chli
Roland Siegwart and Nick Lawrance

V4RL
VISION FOR ROBOTICS LAB

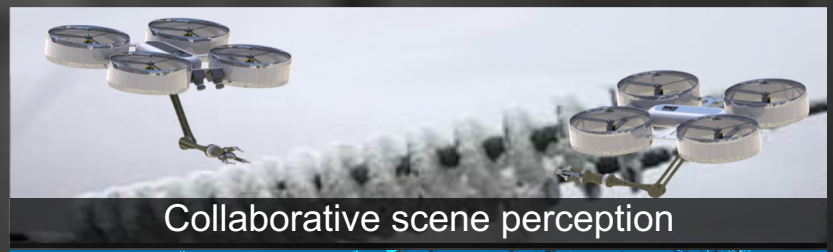
www.v4rl.ethz.ch



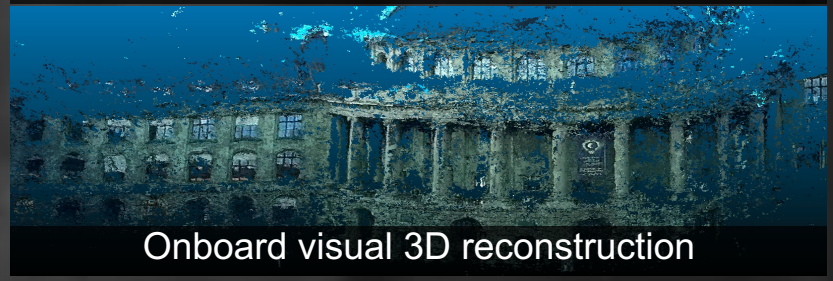
Vision-based robotic perception

Focus on small aircraft

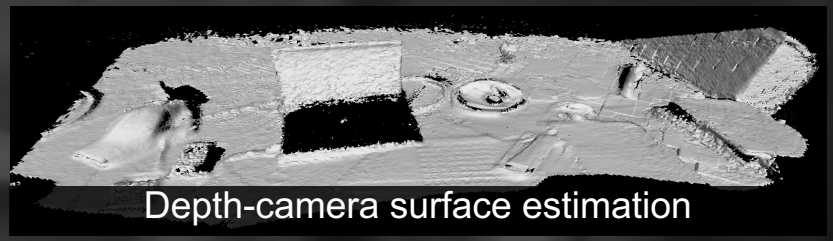
- Tracking & mapping
- Collaborative sensing & mapping for multi-robot
- Experimentation with a variety of cameras



Collaborative scene perception



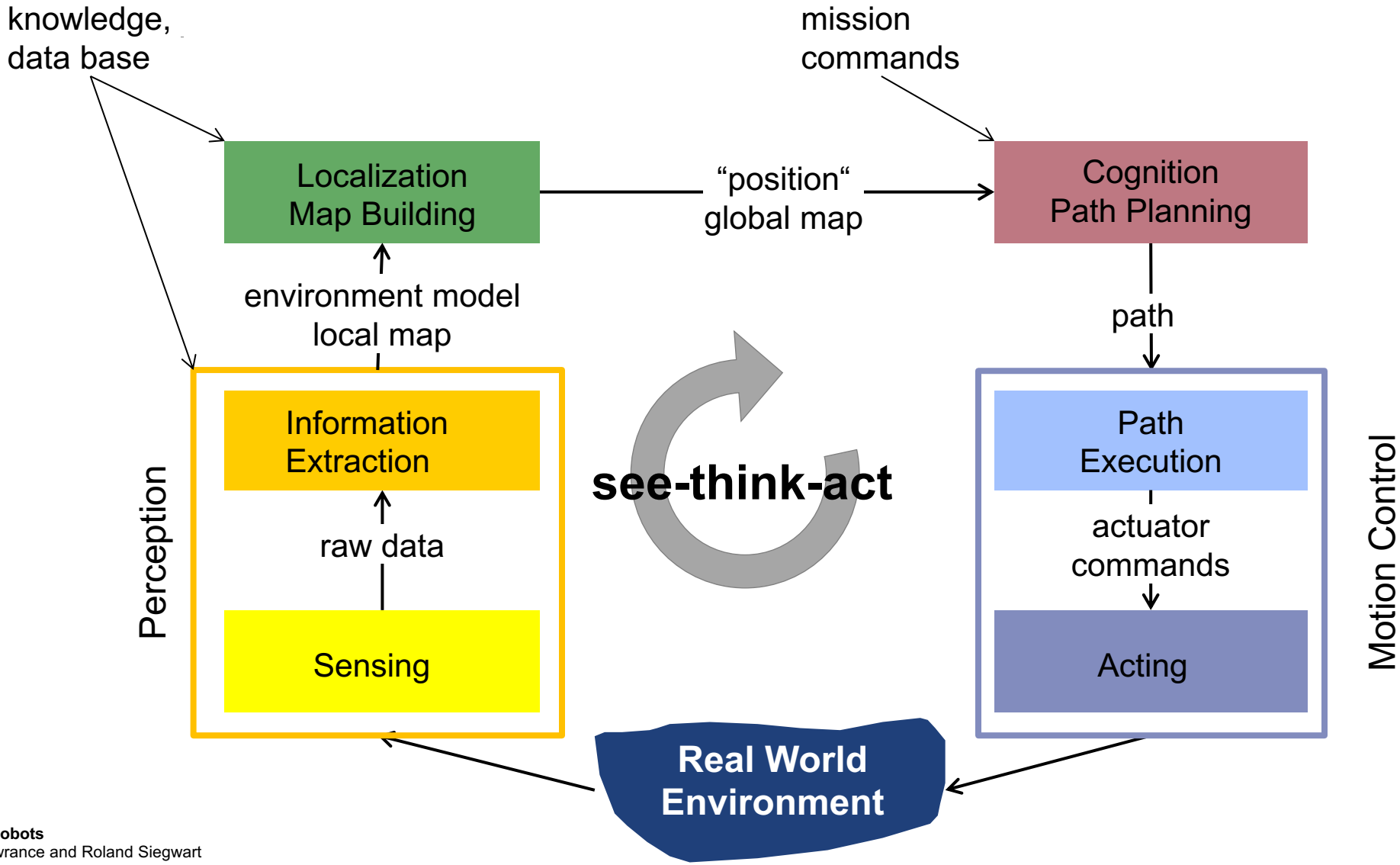
Onboard visual 3D reconstruction



Depth-camera surface estimation

5	23.03.2021	Perception II (to 4.4)	M. Chli
6	30.03.2021	Perception III: Image Saliency (to 4.5)	M. Chli
7	13.04.2021	Perception IV: Place Recognition & Line Fitting (to 4.5)	M. Chli
Ex3	13.04.2021	Line Extraction	L. Bernreiter, N. Alatur, I. Alzugaray
Q1	13.04.2021	Quiz 1	M. Breyer, P. Wulkop
8	20.04.2021	Localization I (to 5.2)	R. Siegwart
9	27.04.2021	Localization II	R. Siegwart
Ex4	27.04.2021	Line-based Extended Kalman Filter	L. Bernreiter, N. Alatur, I. Alzugaray
10	04.05.2021	SLAM I	M. Chli
11	11.05.2021	SLAM II	M. Chli
Ex5	11.05.2021	EKF SLAM	F. Tschopp, P. Schmuck, C. von Einem

Mobile Robot Control Scheme



Today's Topics

Section 4.2 in the book : Fundamentals of Computer Vision

- Pinhole Camera Model
- Perspective Projection
- Camera Calibration
- Stereo Vision, Epipolar Geometry & Triangulation
- Structure from Motion & Optical Flow

Additional, optional reading on Computer Vision: "Computer Vision: Algorithms and Applications", by Richard Szeliski (Springer)

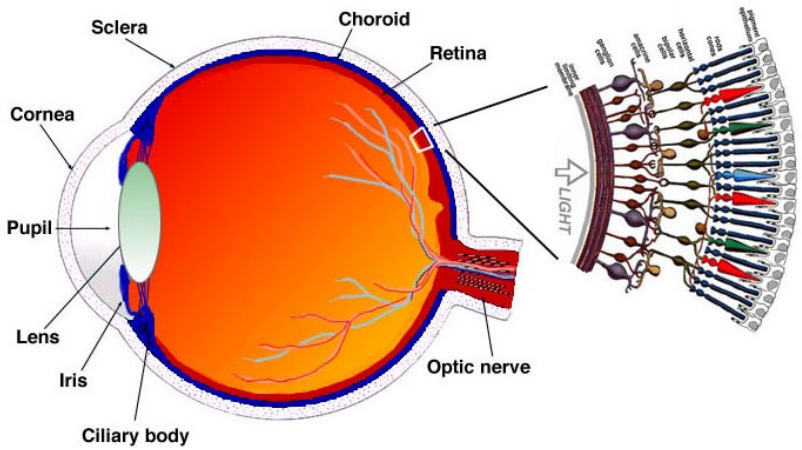
Vision

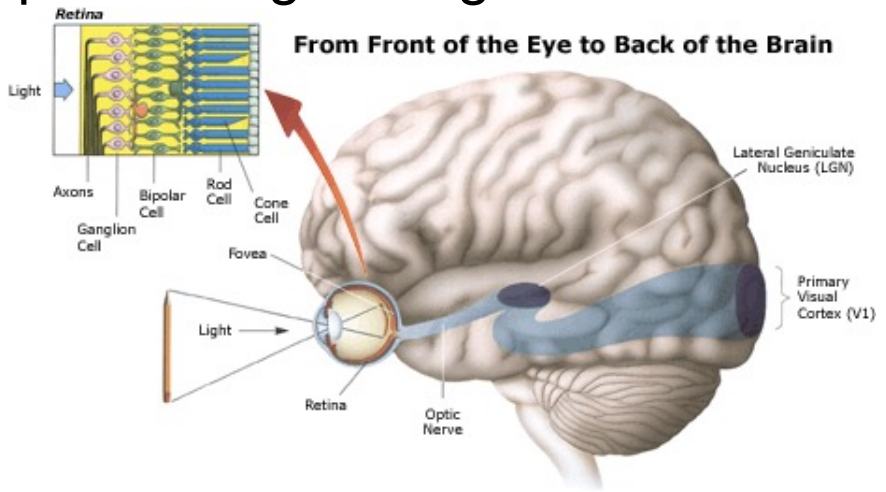
Sony Cybershot WX1

One picture, a thousand words

- **Vision** is our most powerful sense in aiding our perception of the 3D world around us.
- Retina is ~10cm^2. Contains millions of **photoreceptors**
  (120 mil. rods and 7 mil. Cones for colour sampling)
- Provides **enormous** amount of information: data-rate of **~3 *GBytes/s***

⇨ a large proportion of our brain power is dedicated to processing the signals from our eyes





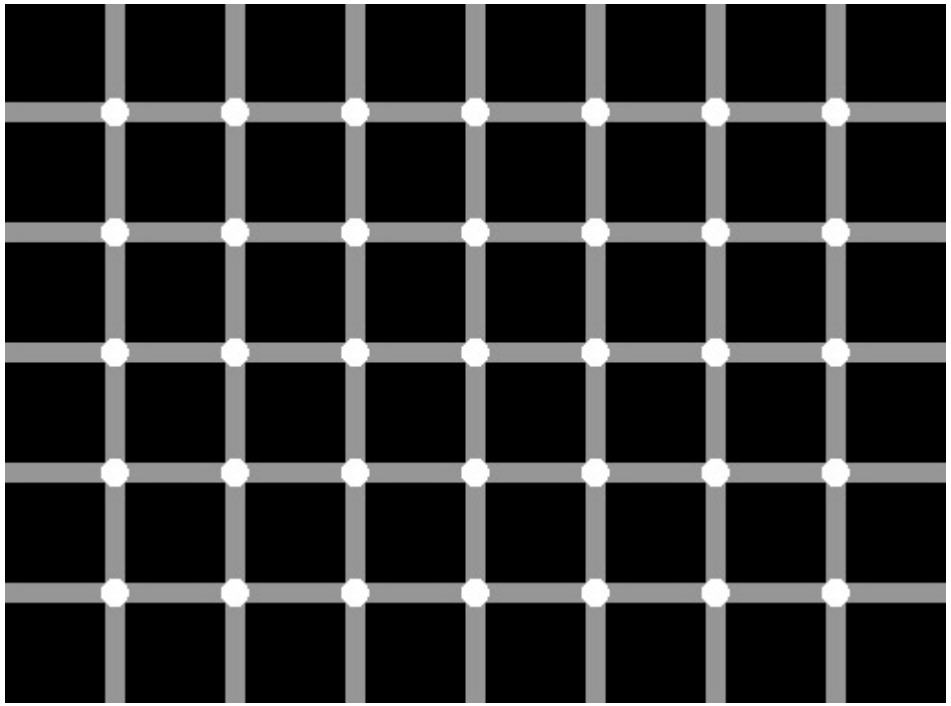
http://webvision.med.utah.edu/sretina.html

Human Visual Capabilities

- Our visual system is very sophisticated
- Humans can interpret images successfully under a wide range of conditions – even in the presence of very limited cues

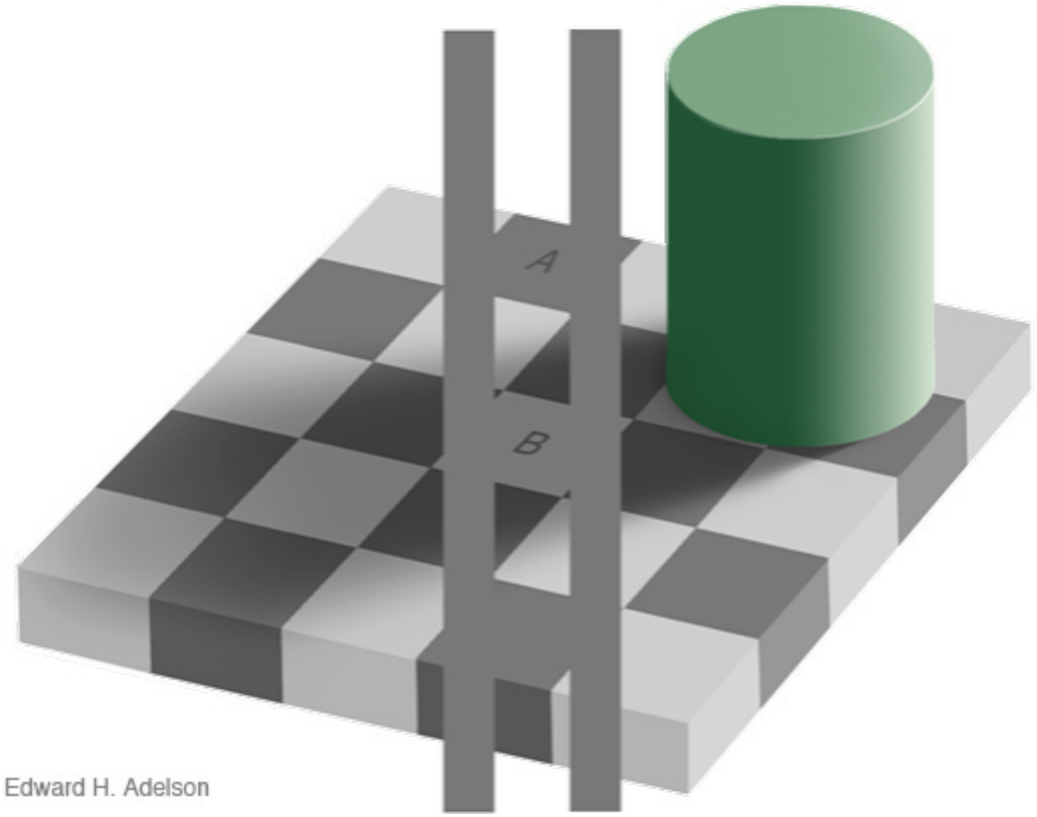
Do we get it always right?

Count the black spots



Do we get it always right?

Which square is darker, A or B?



Edward H. Adelson

Computer Vision for Robotics

- Enormous descriptability of images ⇨ a lot of data to process
  (human vision involves 60 billion neurons!)
- Vision provides humans with a great deal of useful cues
  ⇨ explore the power of vision towards intelligent robots

Cameras:

- Vision is increasingly popular as a sensing modality:
  - descriptive
  - compactness, compatibility, …
  - low cost
  - HW advances necessary to support the processing of images

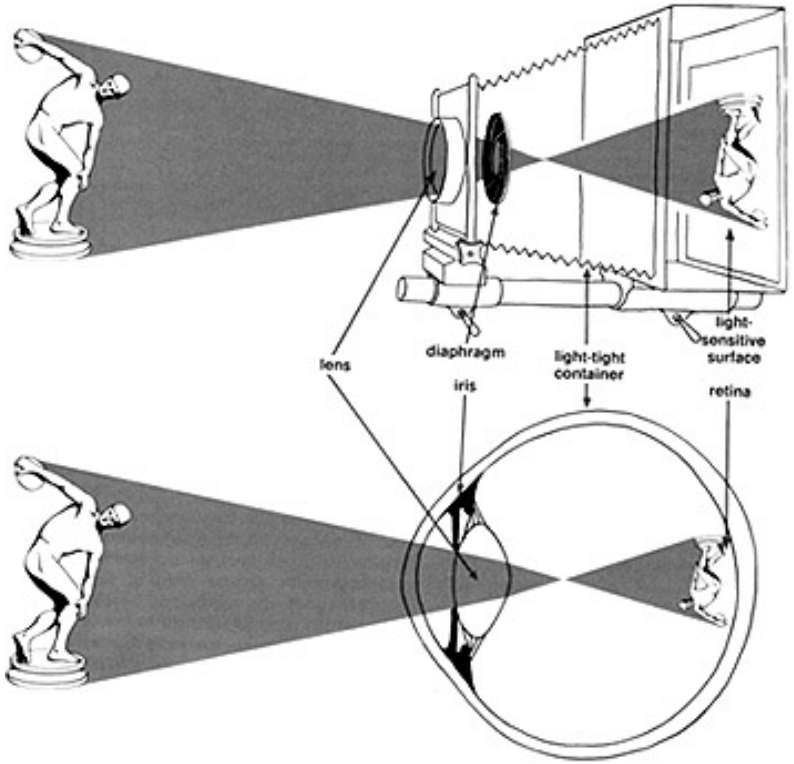
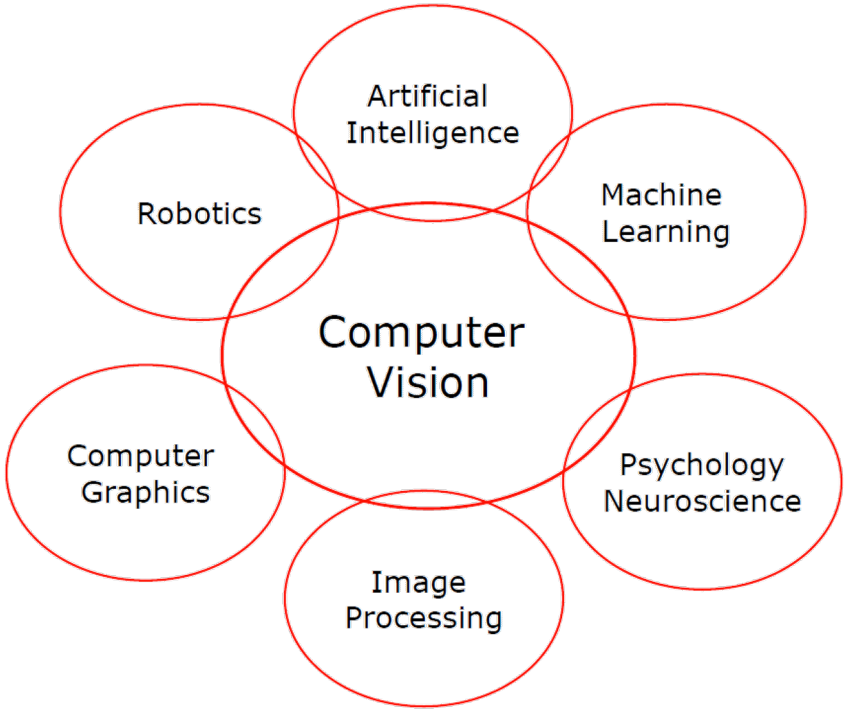


Image: Nicholas M. Short

Computer Vision | applications

q97country.com



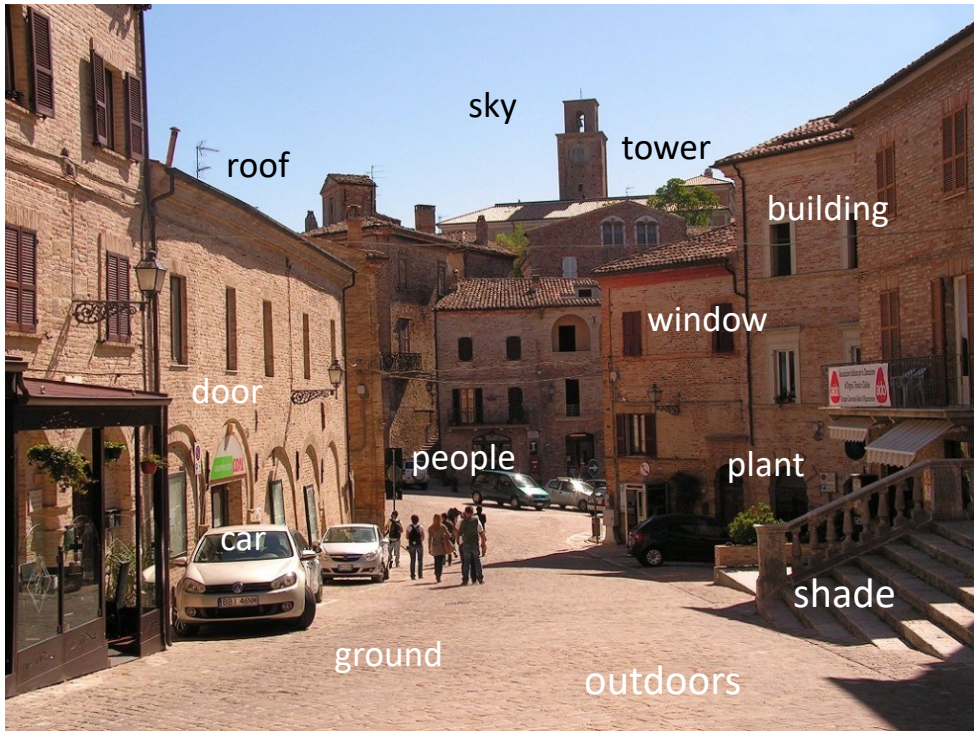
www.robertlpeters.com

www.samsung.com

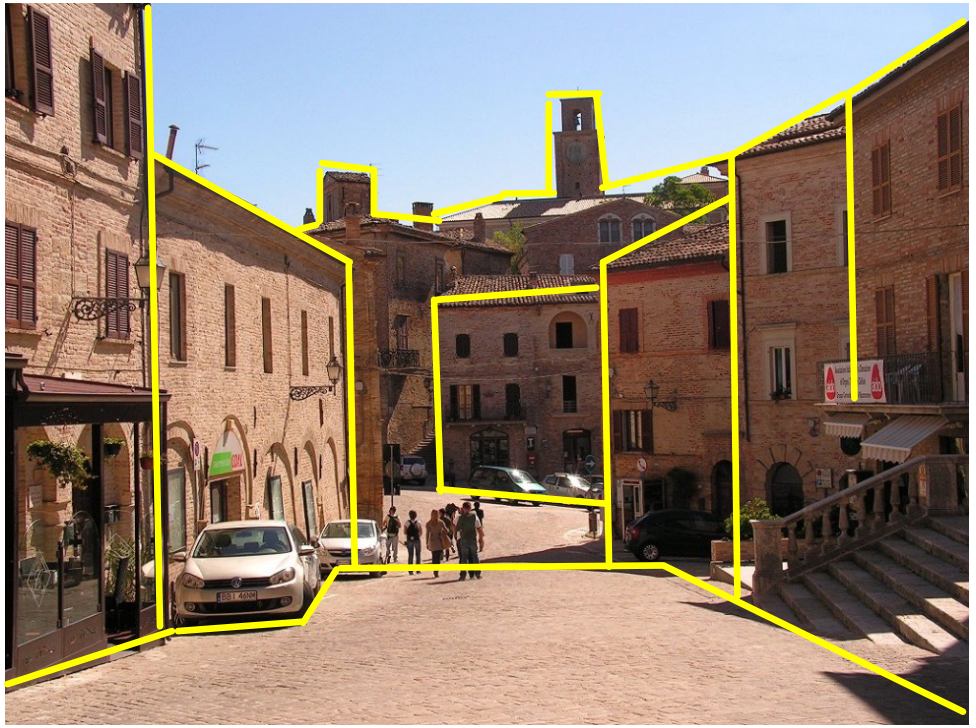

news.nationalgeographic.com

Computer Vision | what is it?

- Automatic extraction of “meaningful” information from images and videos – varies depending on the application



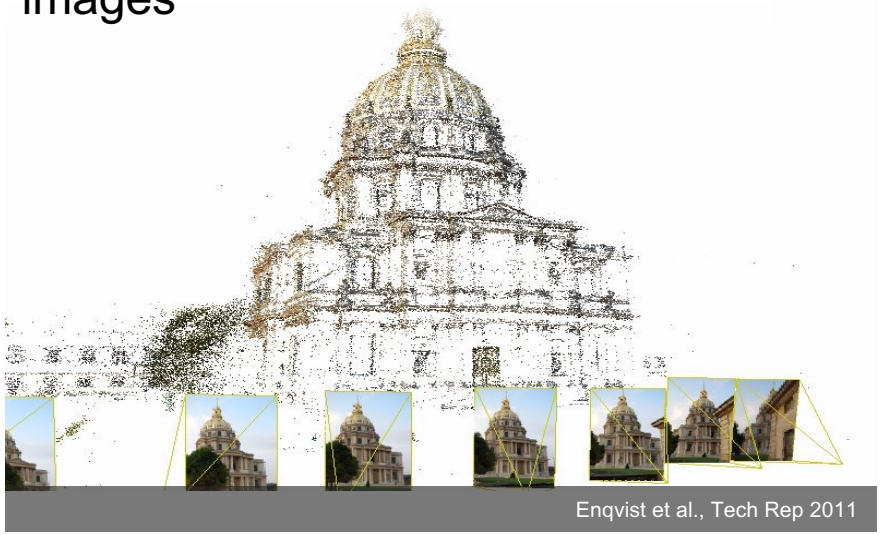
Semantic information



Geometric information

Computer Vision | how far along are we?

- Partial 3D model from overlapping images

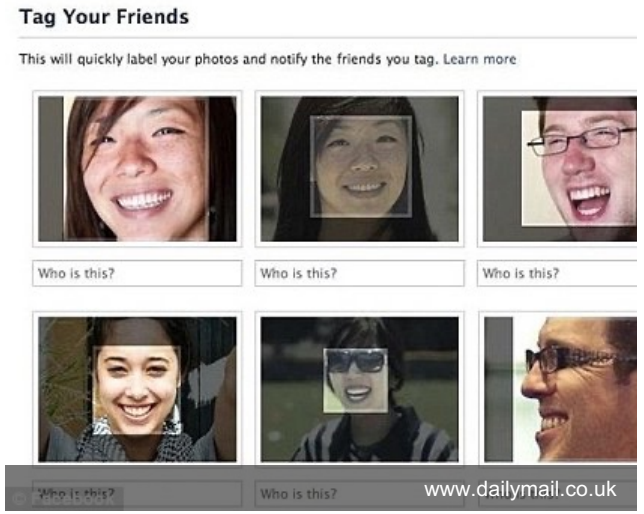


Enqvist et al., Tech Rep 2011

- Dense 3D surface model (from more overlapping images)

Goesele et al., ICCV 2007

- Face recognition



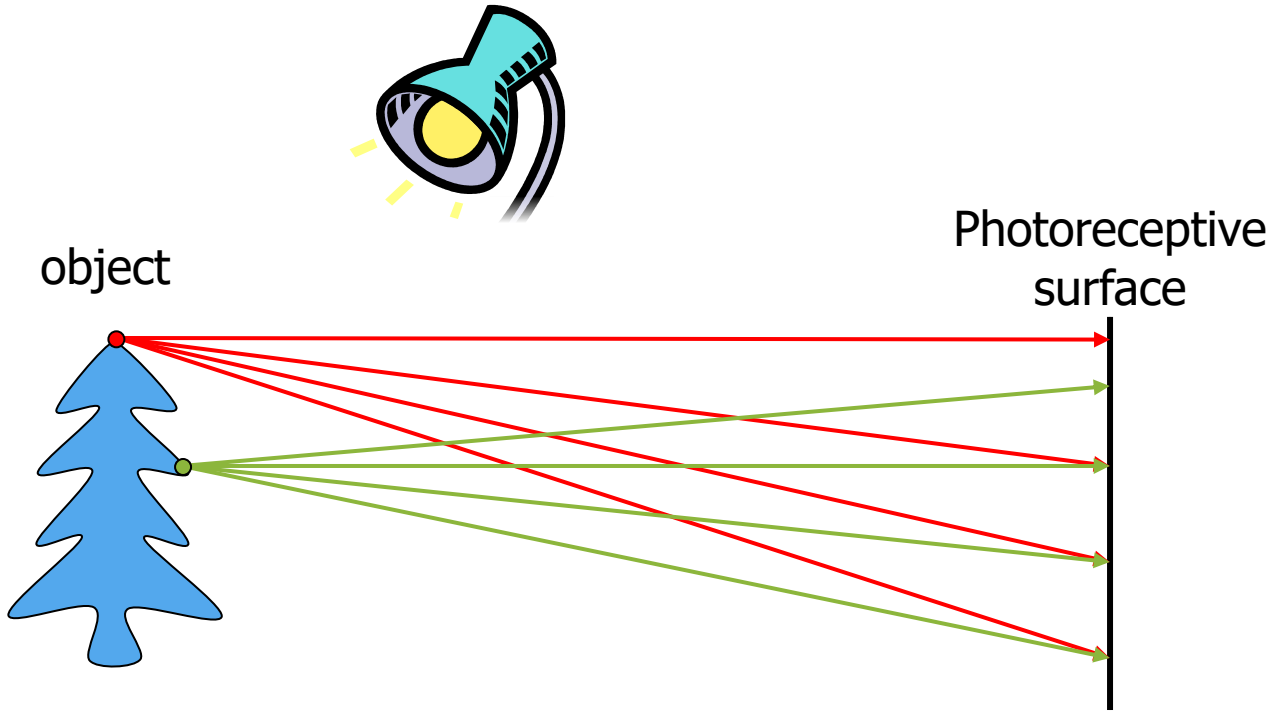
www.dailymail.co.uk

http://articles.orlandosentinel.com

- How many animals in the image?
- Resort to physics-based probabilistic models to disambiguate solutions

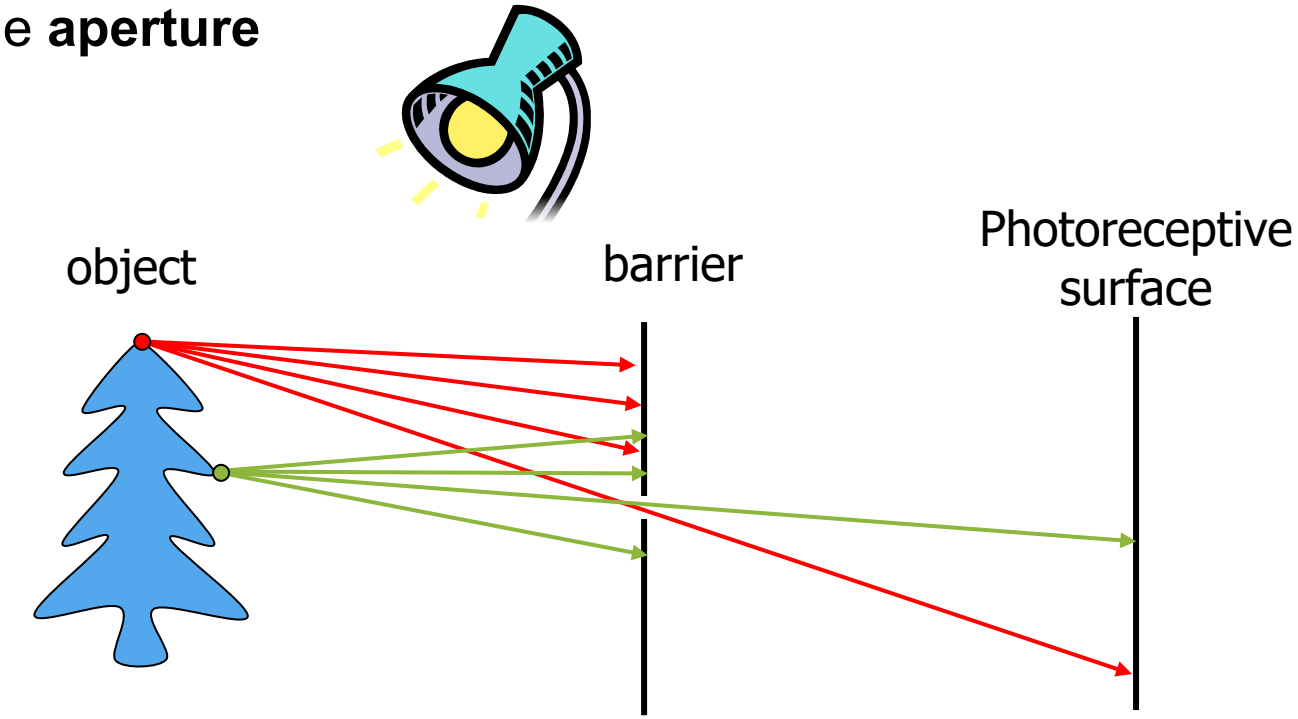
The camera | image formation

- If we place a piece of film in front of an object, do we get a reasonable image?

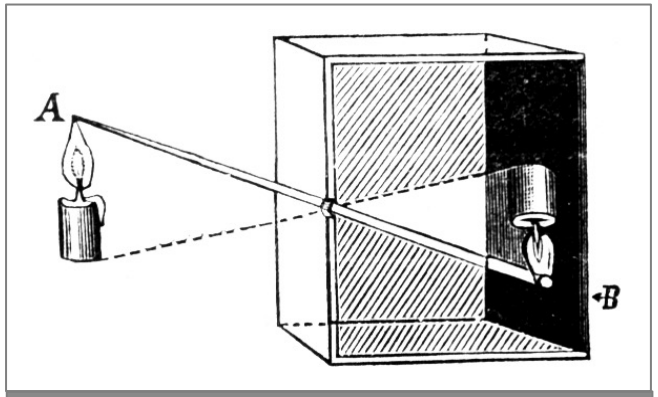


The camera | image formation

- If we place a piece of film in front of an object, do we get a reasonable image?
- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening is known as the **aperture**



The camera | camera obscura

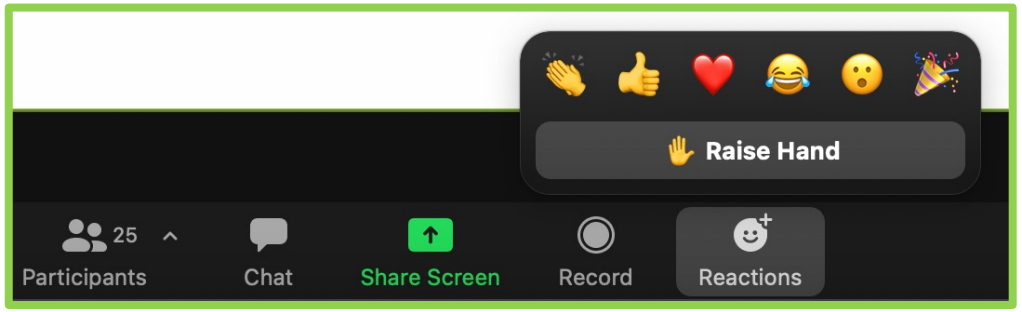


commons.wikimedia.org

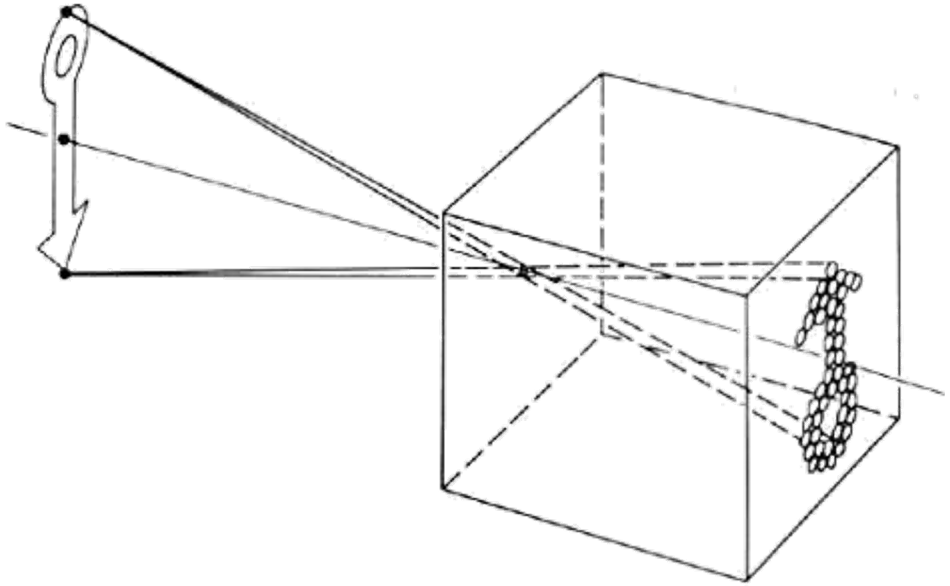
- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC), Euclid (323-283 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)
- Image is inverted
- Depth of the room (box) is the effective focal length

The camera | camera obscura

- Notice anything particular about this image?



The camera | the pinhole camera model



- Pinhole model:
  - Captures **beam of rays** – all rays through a single point (note: no lens!)
  - The point is called **Center of Projection** or **Optical Center**
  - The image is formed on the **Image Plane**

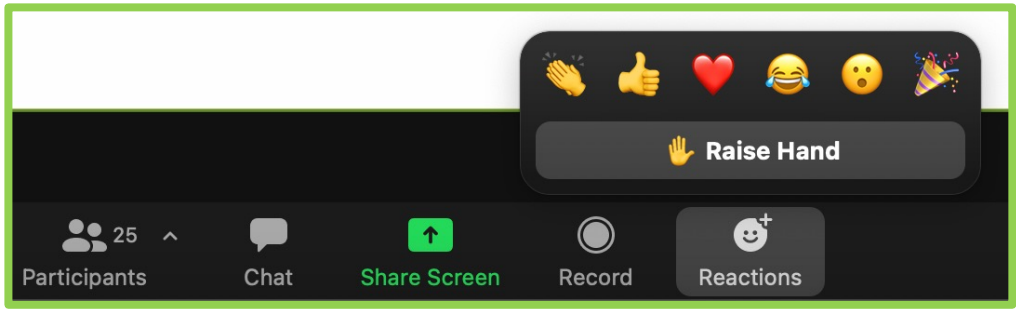
- We will use the pinhole camera model to describe how the image is formed

Based on slide by Steve Seitz

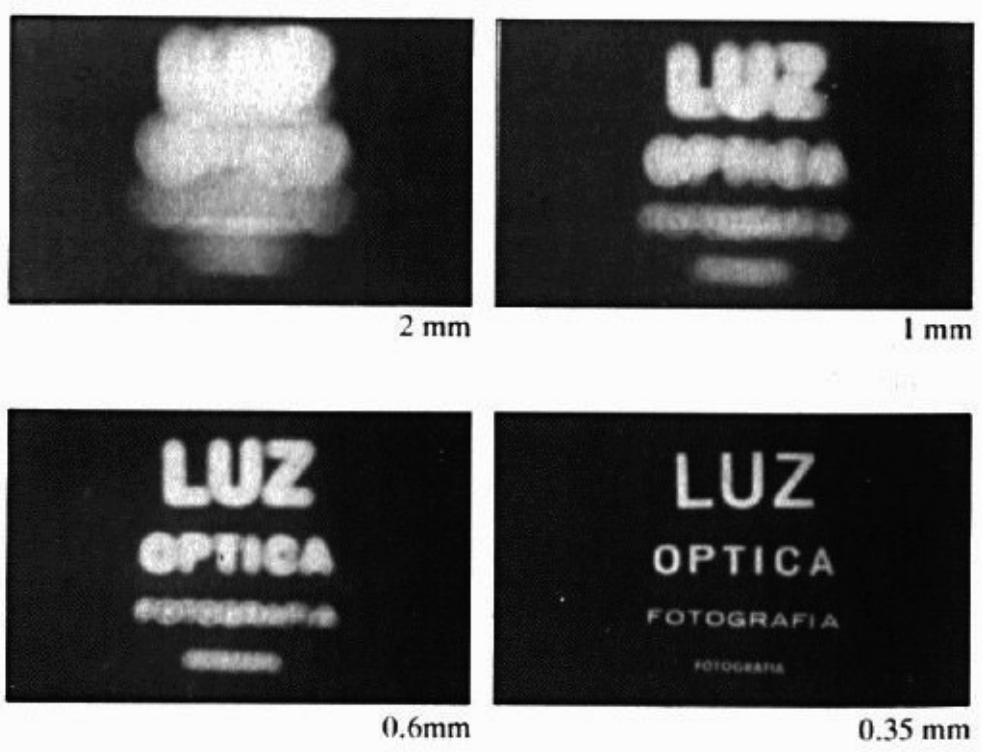
The camera | home-made pinhole camera

What can we do to reduce the blur?

Think about it for a minute & raise your hand to share your idea

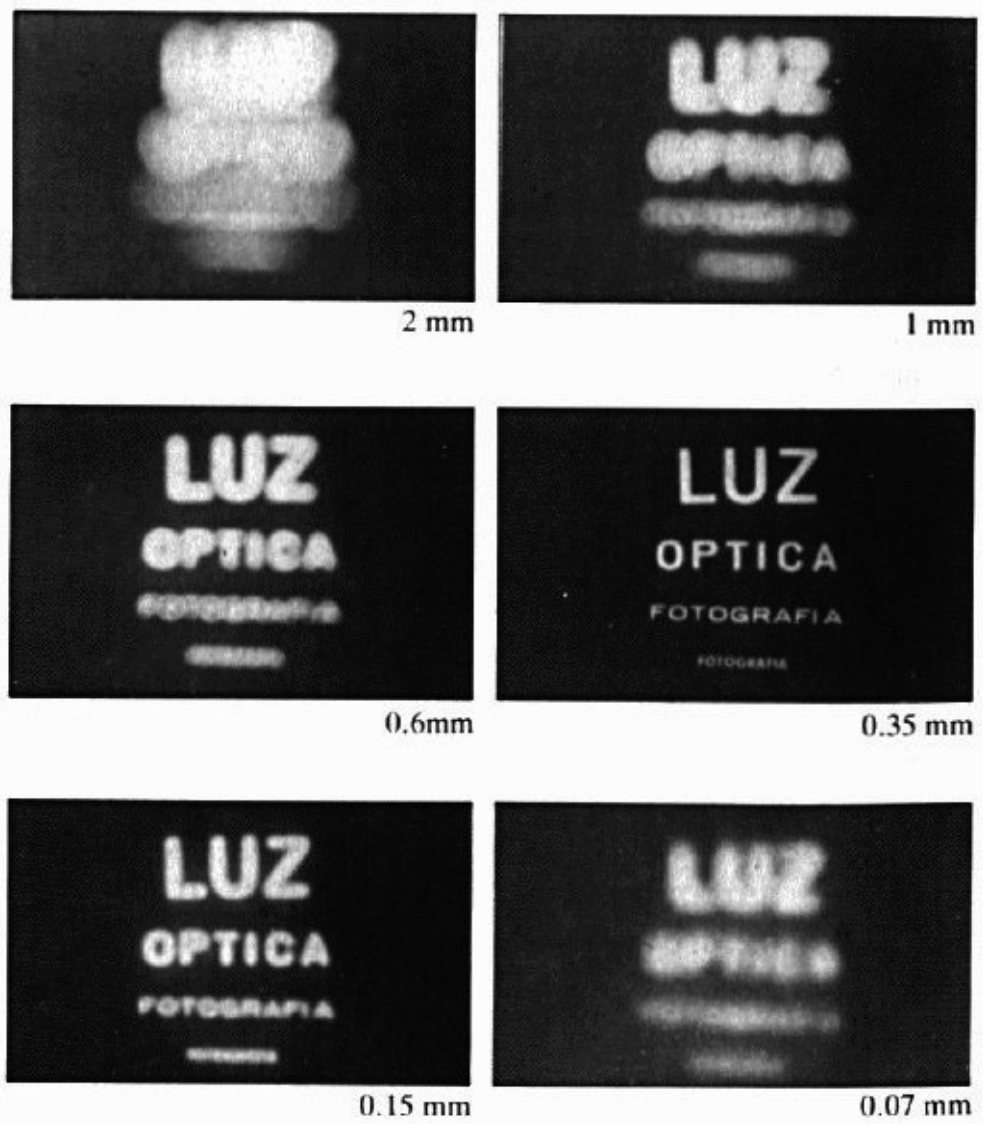


The camera | shrinking the aperture



Why not make the aperture as small as possible?

The camera | shrinking the aperture



Why not make the aperture as small as possible?

- Less light gets through (must increase the exposure)
- Diffraction effects…

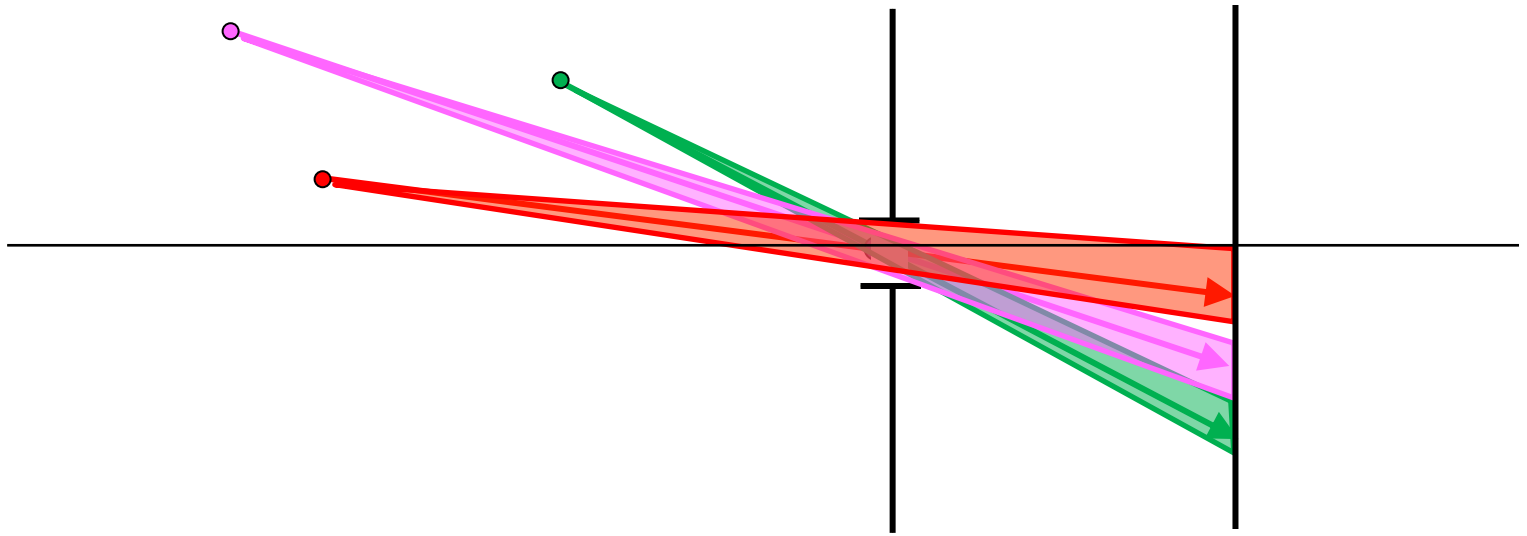
The camera | why use a lens?

- *The ideal pinhole*:

  only one ray of light reaches each point on the film

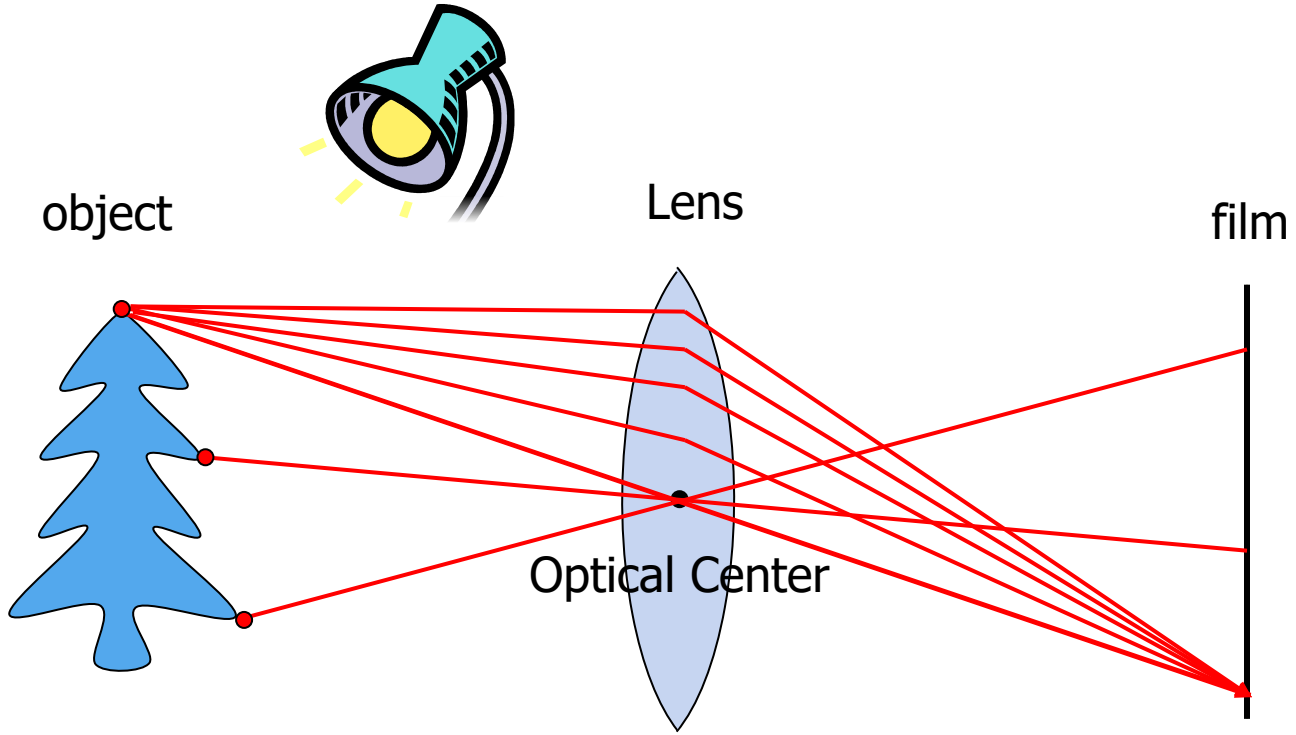
  ⇨ image can be very dim; gives rise to diffraction effects

- Making the pinhole bigger (i.e. aperture) makes the image blurry



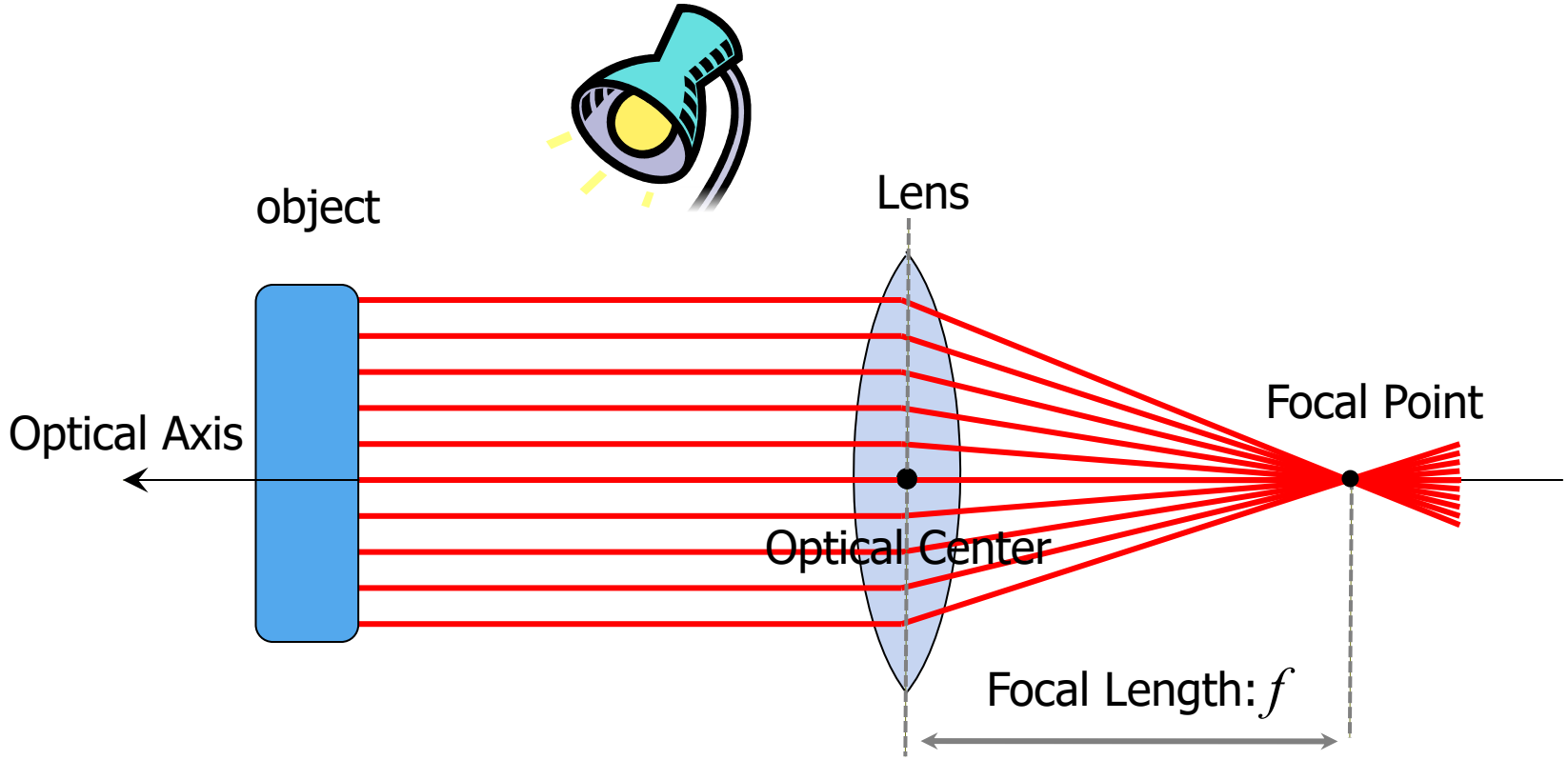
The camera | why use a lens?

- A lens focuses light onto the film
- Rays passing through the **optical center** are not deviated

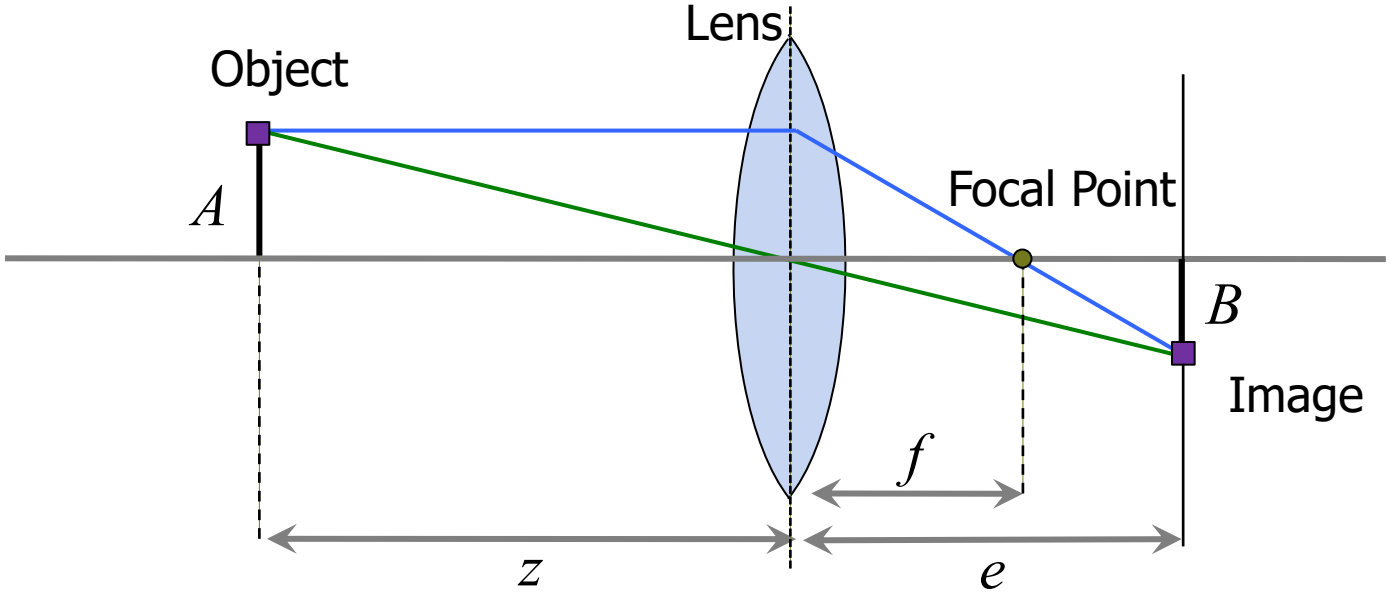


The camera | why use a lens?

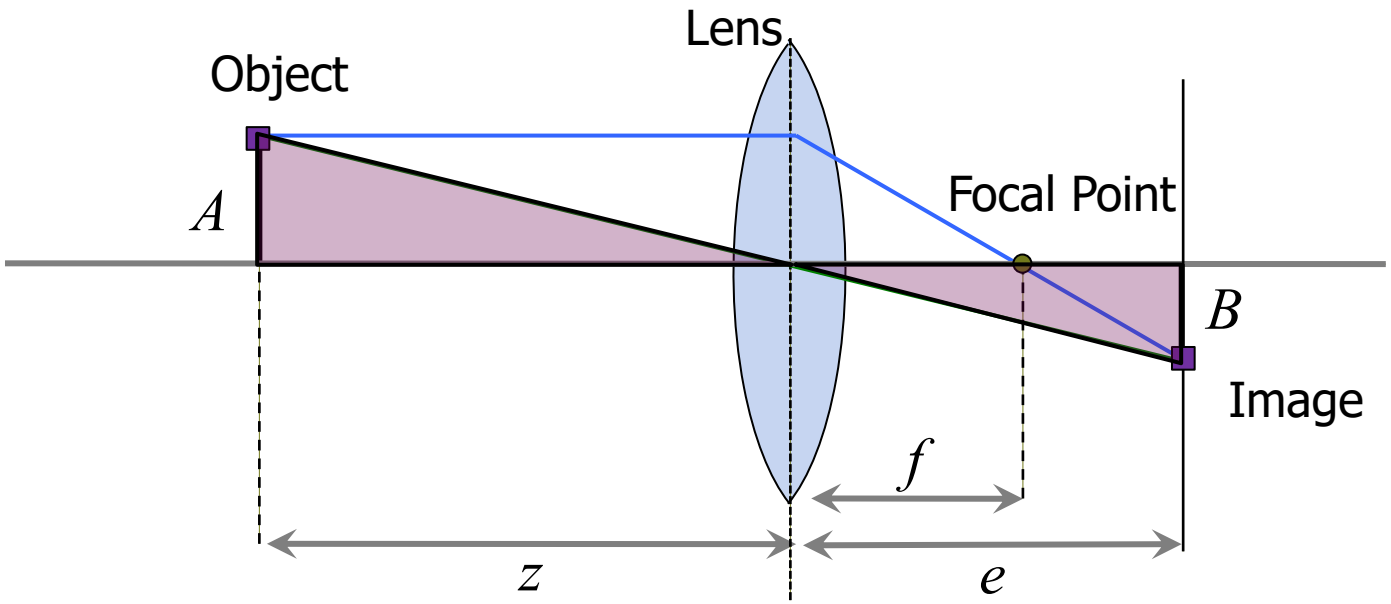
- A lens focuses light onto the film
- Rays passing through the **optical center** are not deviated
- All rays parallel to the **optical axis** converge at the **focal point**



The camera | how to create a focused image?

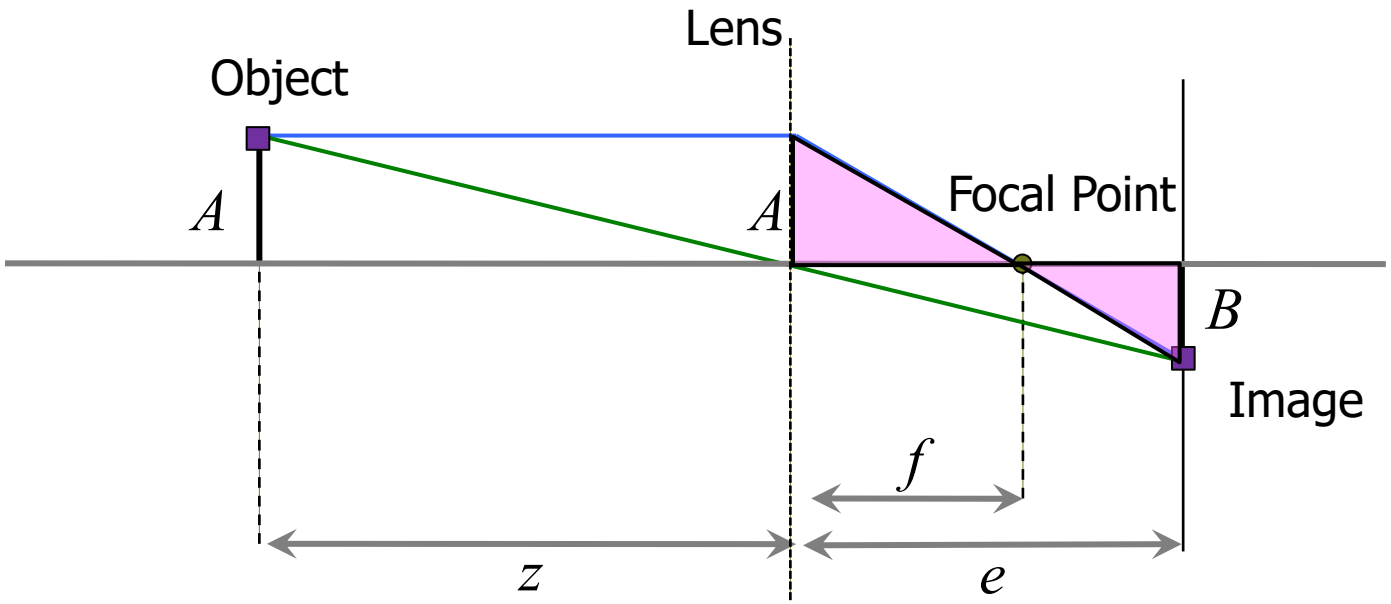


The camera | how to create a focused image?



Find a relationship between f, z and e

The camera | the thin lens equation

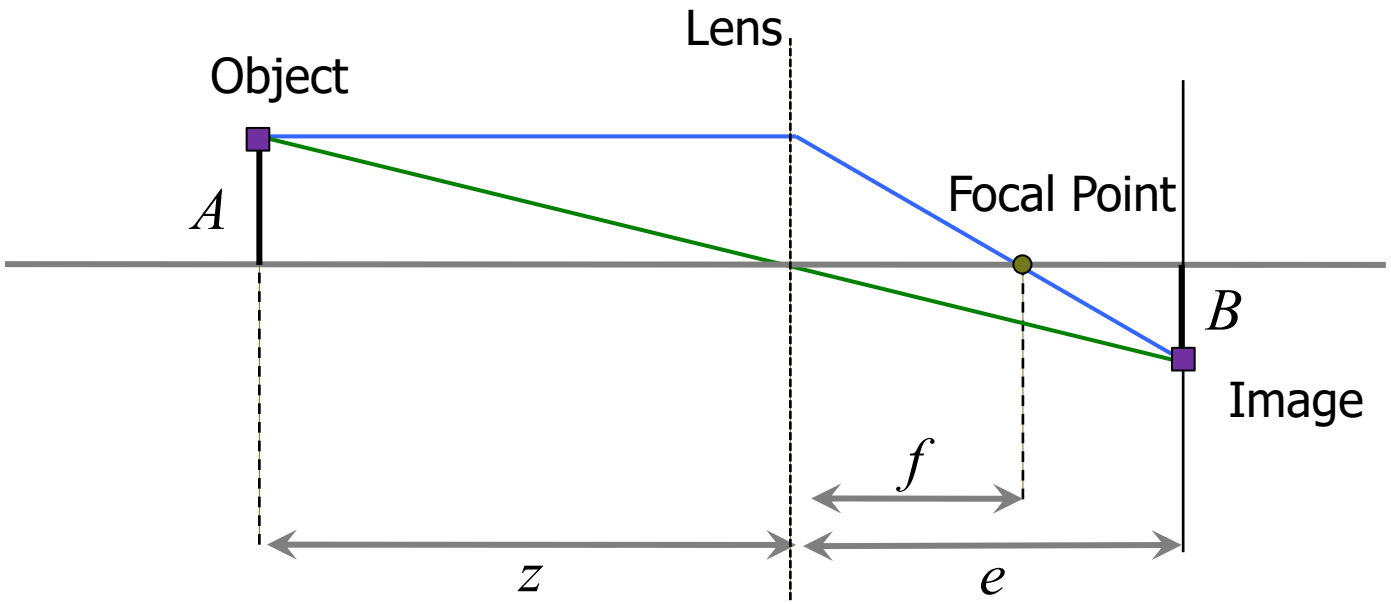


- Similar Triangles:

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

"Thin lens equation"

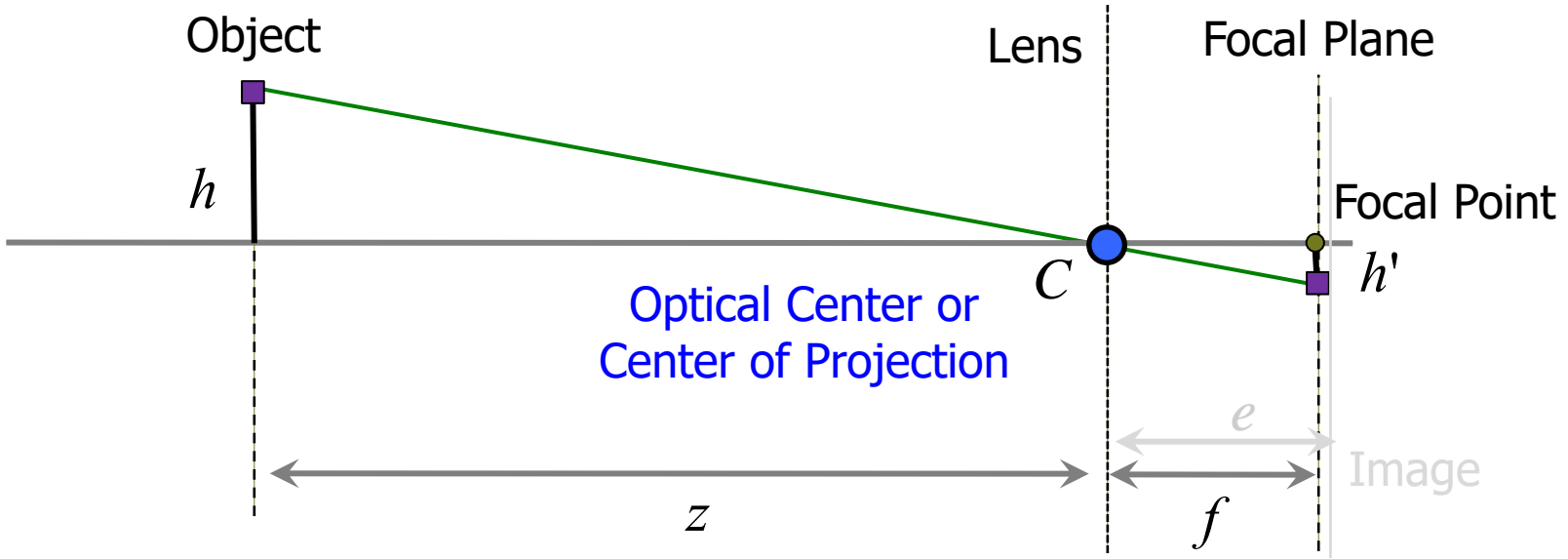
The camera | pinhole approximation



- What happens if z is very big, i.e. $z >> f$?

$$\frac{1}{f} = \boxed{\frac{1}{z}} + \frac{1}{e}$$
$$\cong 0$$

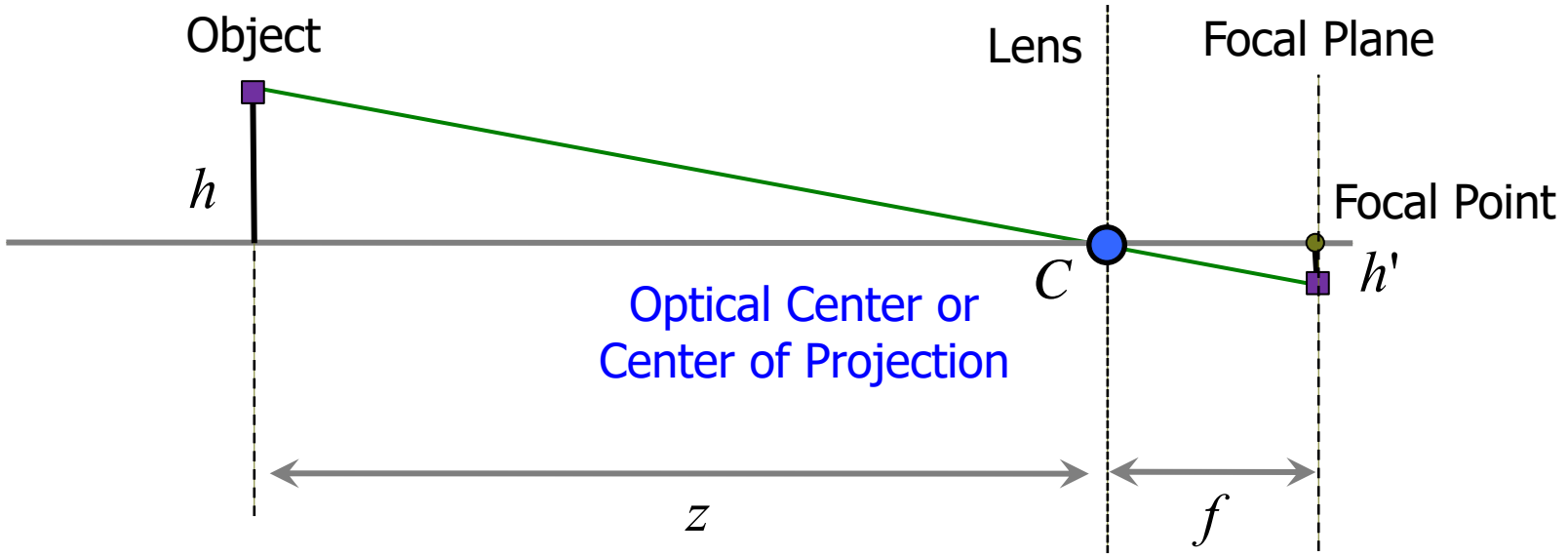
The camera | pinhole approximation



- What happens if z is very big, i.e. $z >> f$?
- We adjust the image plane s.t. objects at infinity are in focus

$$\frac{1}{f} = \underbrace{\frac{1}{z}}_{\cong 0} + \frac{1}{e} \quad \Rightarrow \frac{1}{f} \approx \frac{1}{e} \Rightarrow f \approx e$$

The camera | pinhole approximation

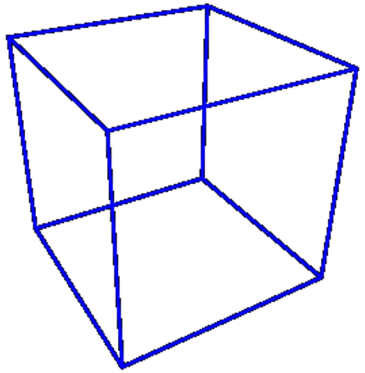


- The dependence of the apparent size of an object on its depth (i.e. distance from camera) is known as ***perspective***.

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

Playing with Perspective

- Perspective gives us very strong depth cues
  ⇨ hence we can perceive a 3D scene by viewing its 2D representation (i.e. image)

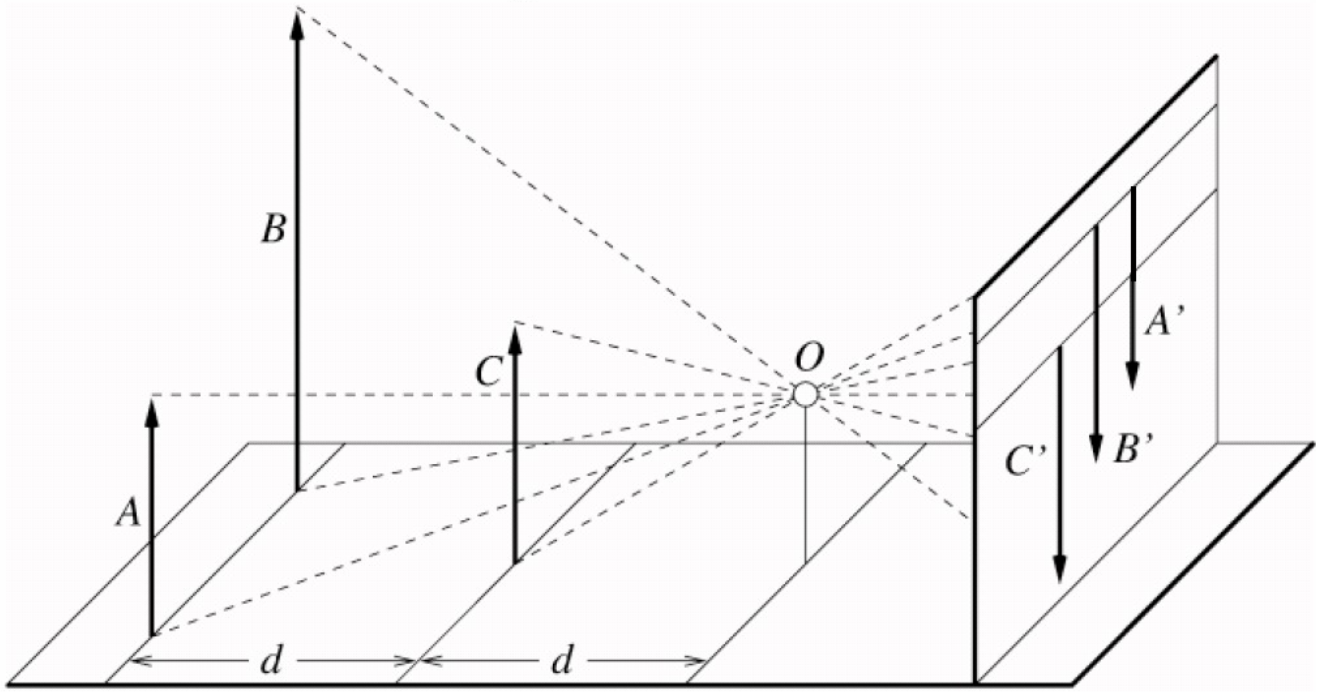


"Ames room"

A clip from "The computer that ate Hollywood" documentary. Dr. Vilayanur S. Ramachandran.

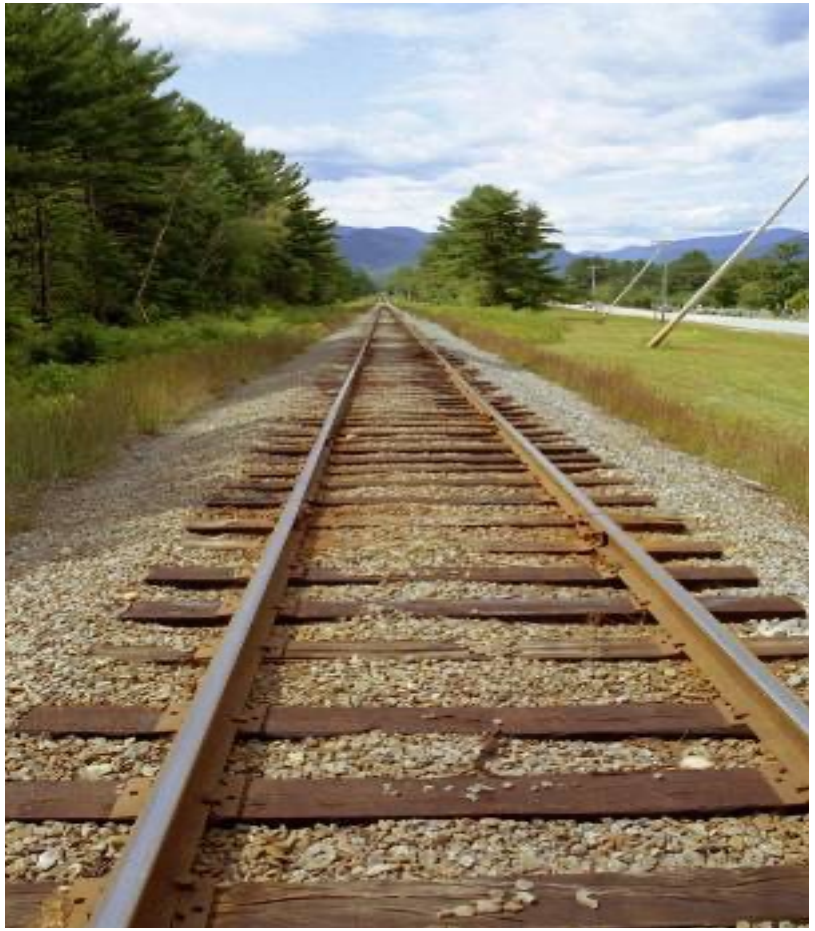
Perspective effects

- Far away objects appear smaller

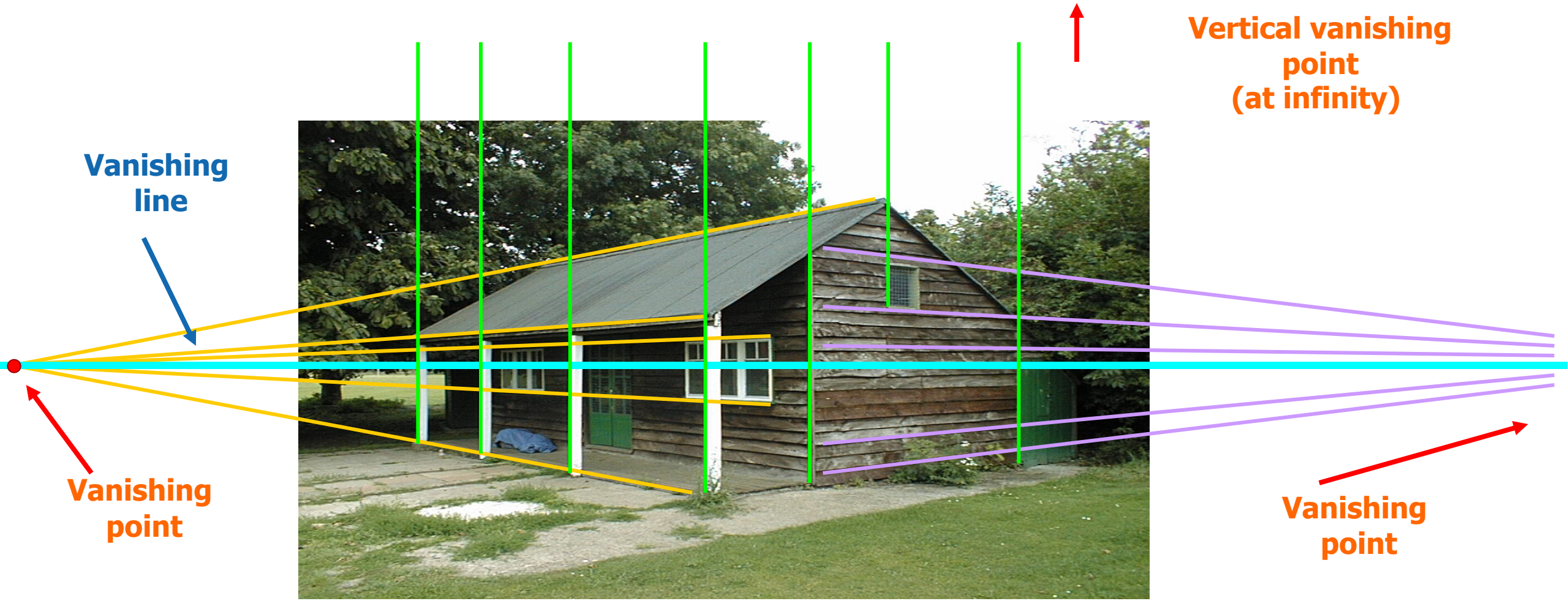


Vanishing points and lines

- Parallel lines in the world intersect in the image at a “vanishing point”



Vanishing points and lines



Today's Topics

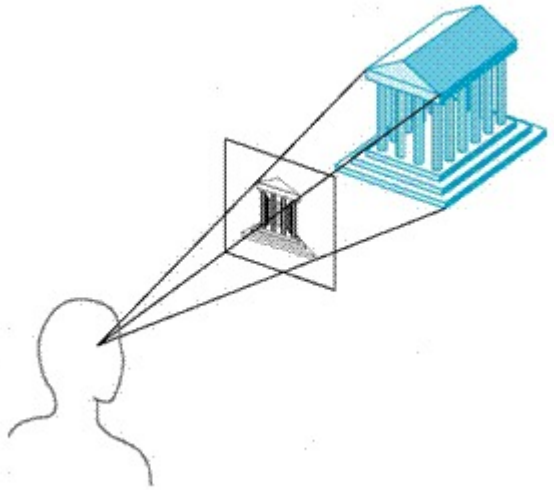
Section 4.2 in the book : Fundamentals of Computer Vision

- Pinhole Camera Model
- Perspective Projection
- Camera Calibration
- Stereo Vision, Epipolar Geometry & Triangulation
- Structure from Motion & Optical Flow

Additional, optional reading on Computer Vision: "Computer Vision: Algorithms and Applications", by Richard Szeliski (Springer)

Perspective projection

How world points map to pixels in the image?

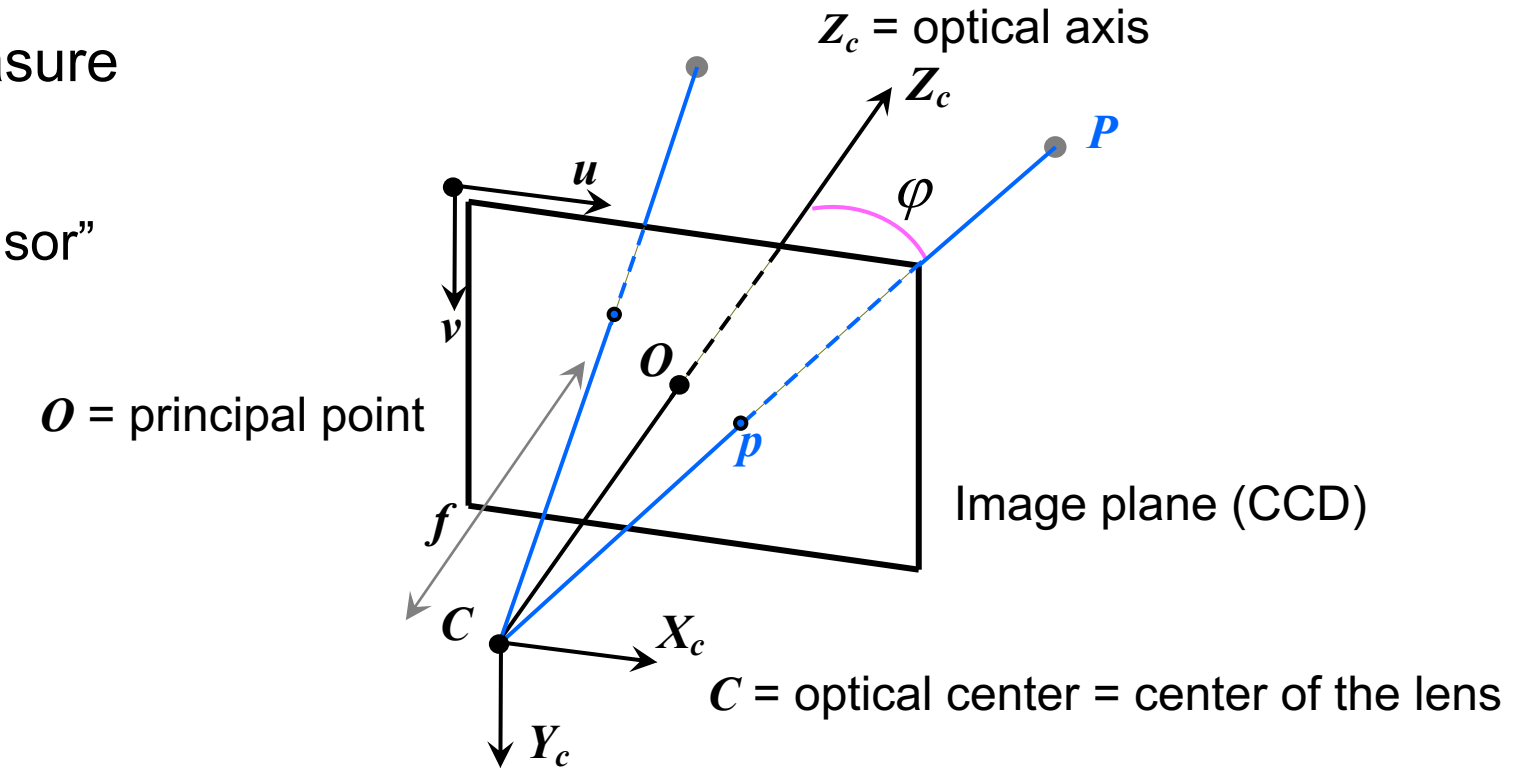


Perspective projection

- For convenience: the image plane is usually represented in front of C, such that the image preserves the same orientation (i.e. not flipped)

- A camera does not measure distances but angles!

⇨ a camera is a “bearing sensor”



Perspective projection | from world to pixel coordinates

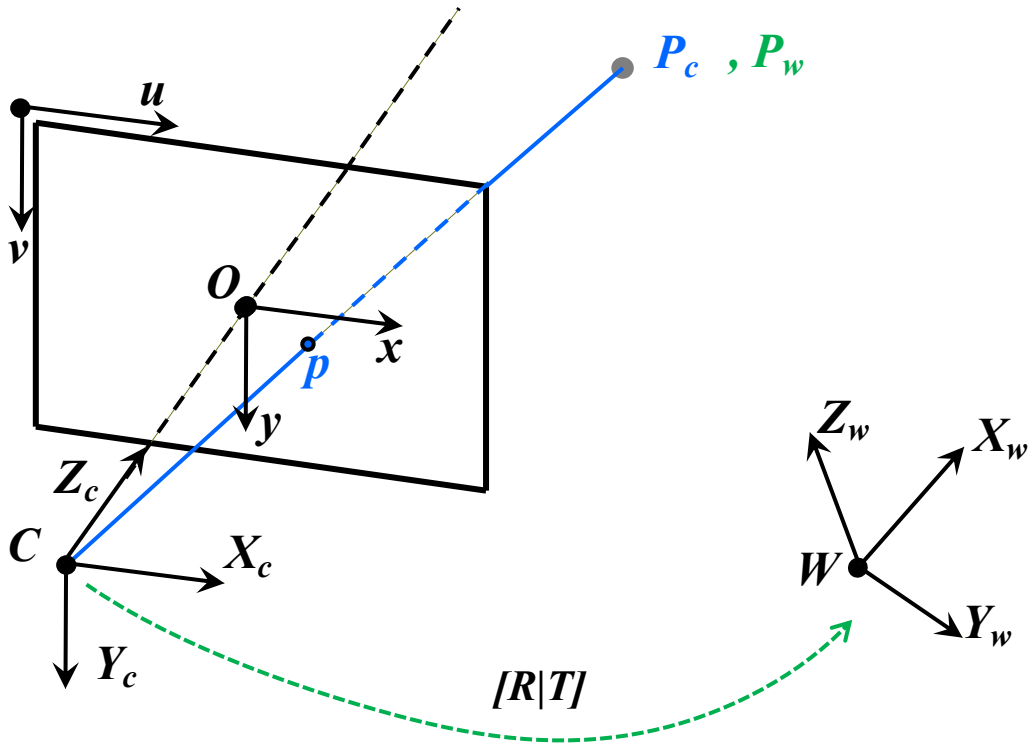
Find pixel coordinates (u,v) of point P_w in the world frame:

0. Convert world point P_w to camera point P_c

Find pixel coordinates (u,v) of point P_c in the camera frame:

1. Convert P_c to image-plane coordinates (x,y)

2. Convert P_c to (discretised) pixel coordinates (u,v)

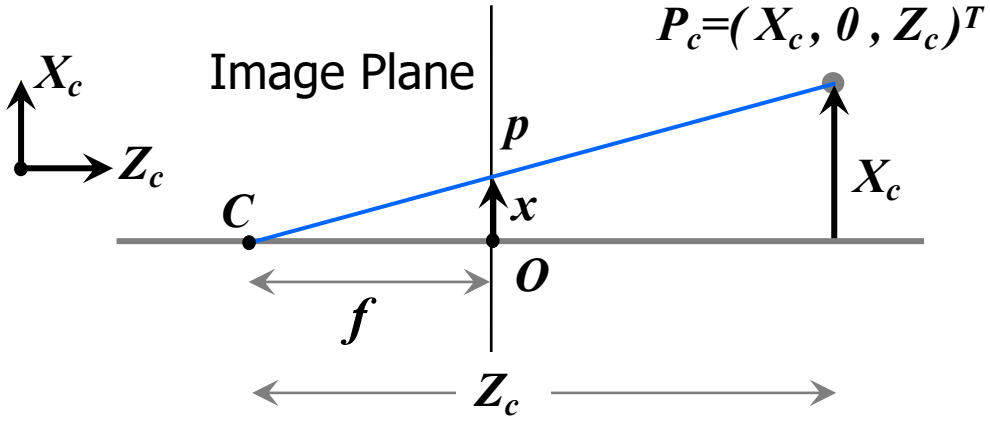


Perspective projection | from camera frame to image plane

- The Camera point $\boldsymbol{P_c = (X_c, 0, Z_c)^T}$ projects to $\boldsymbol{p = (x, y)}$ onto the image plane

- From similar triangles:

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$



- Similarly, in the general case:

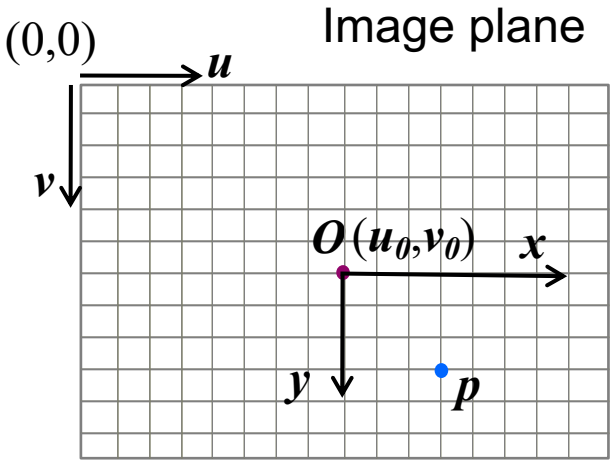
$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$

1. Convert $\boldsymbol{P_c}$ to image-plane coordinates $\boldsymbol{(x,y)}$

2. Convert $\boldsymbol{P_c}$ to (discretised) pixel coordinates $\boldsymbol{(u,v)}$

Perspective projection | from camera frame to pixel coords.

- Convert ***p*** from the local image plane coords ***(x,y)*** to the pixel coords ***(u,v)***, we need to account for:
  - the pixel coords of the camera optical center $O = (u_0, v_0)$
  - scale factors k_u, k_v for the pixel-size in both dimensions



- So: $u = u_0 + k_u x \Rightarrow u = u_0 + \frac{k_u f X_c}{Z_c}$

$$v = v_0 + k_v y \Rightarrow v = v_0 + \frac{k_v f Y_c}{Z_c}$$

- Use **Homogeneous Coordinates** for linear mapping from 3D to 2D, by introducing an extra element (scale):

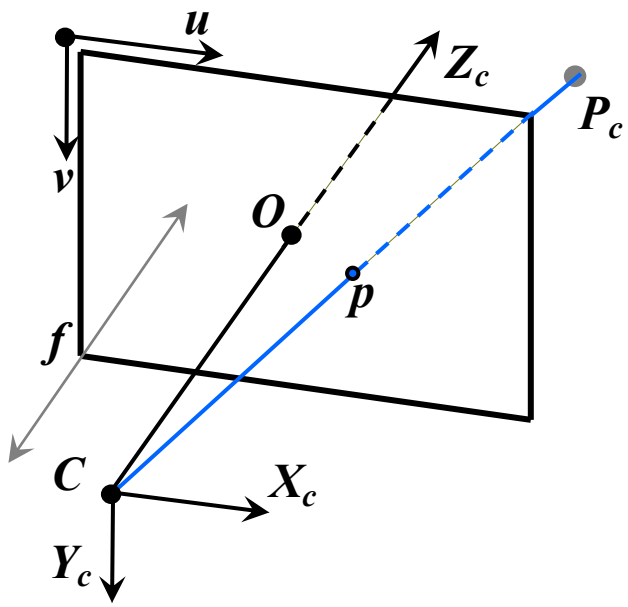
$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

and similarly for the world coordinates. Note, usually $\lambda = 1$

Perspective projection | from camera frame to pixel coords.

$$u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + \frac{k_v f Y_c}{Z_c}$$



- Expressed in matrix form and homogeneous coordinates:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

- Or alternatively

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Focal length in u-direction

Focal length in v-direction

"Calibration matrix"/
"Matrix of Intrinsic Parameters"

Perspective projection | from the world to the camera frame

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

in homogeneous coordinates

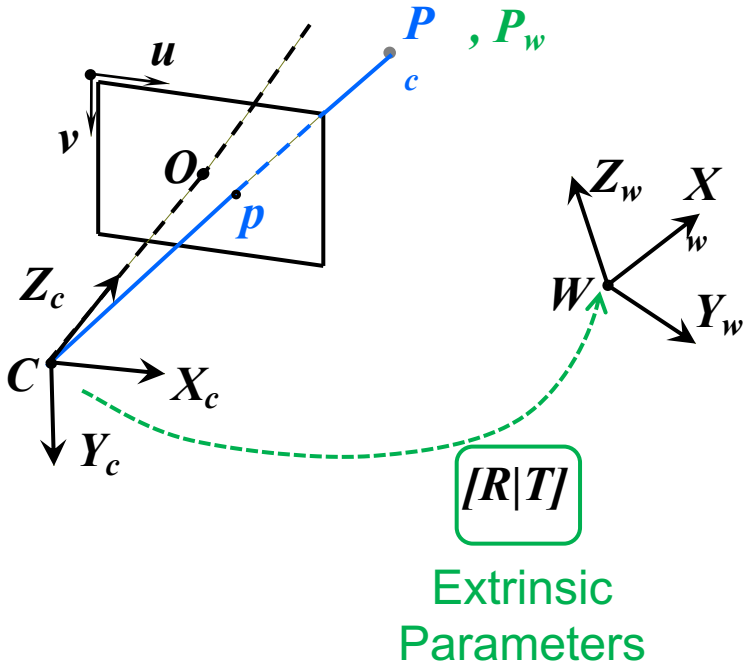
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

From the previous slide:

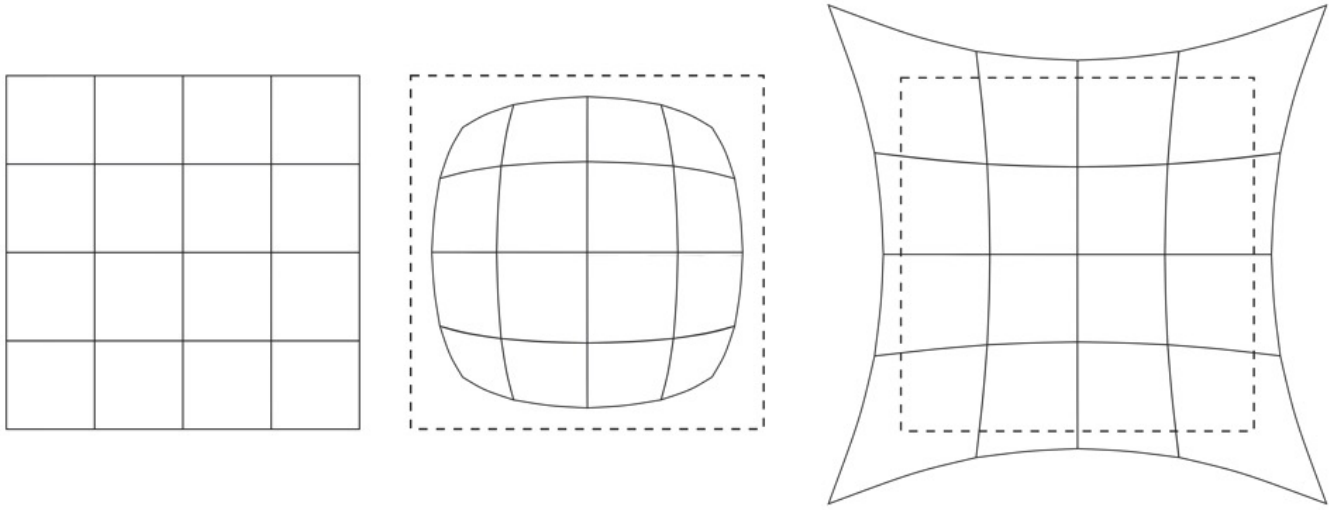
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Projection Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



Perspective projection | radial distortion



No distortion | Barrel distortion | Pincushion distortion

- Amount of distortion is a non-linear f^n of distance from center of image

From ideal (u,v) to distorted pixel coordinates (u_d, v_d):

- Simple quadratic distortion model:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

Radial Distortion parameter

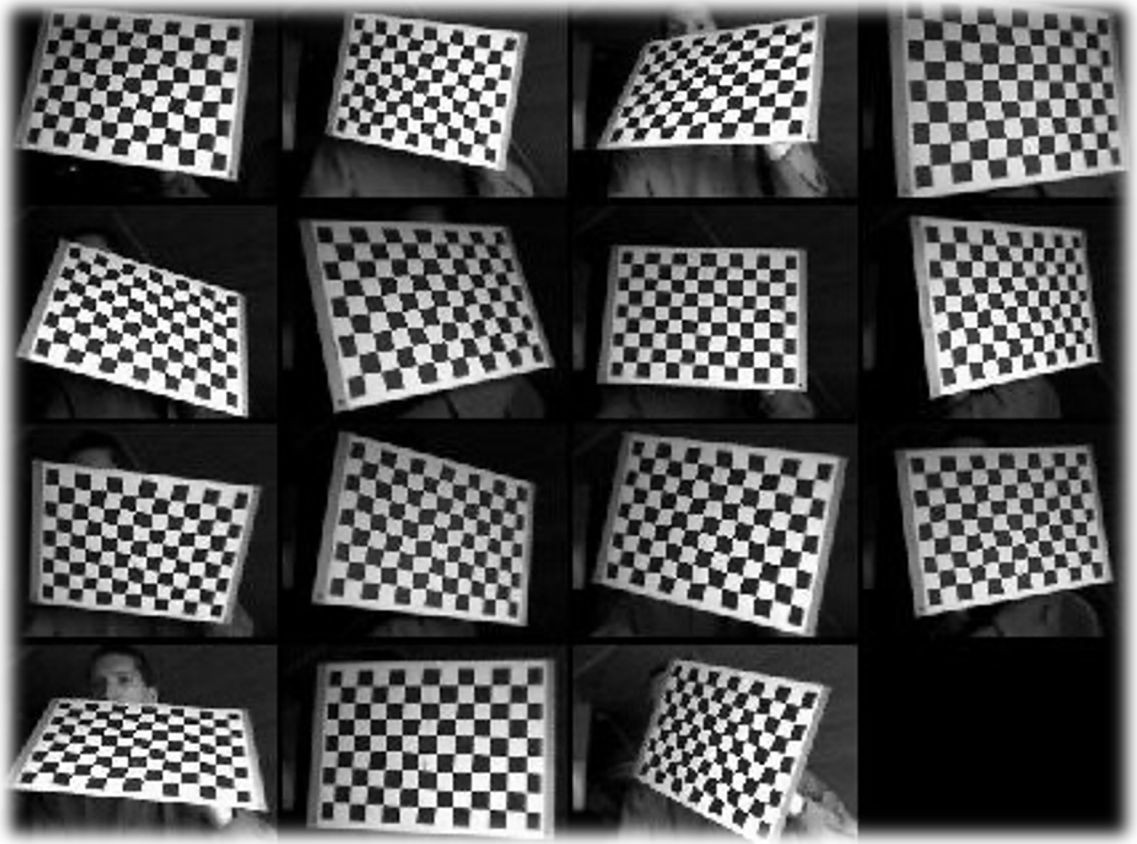
where

$$r^2 = (u - u_0)^2 + (v - v_0)^2 .$$

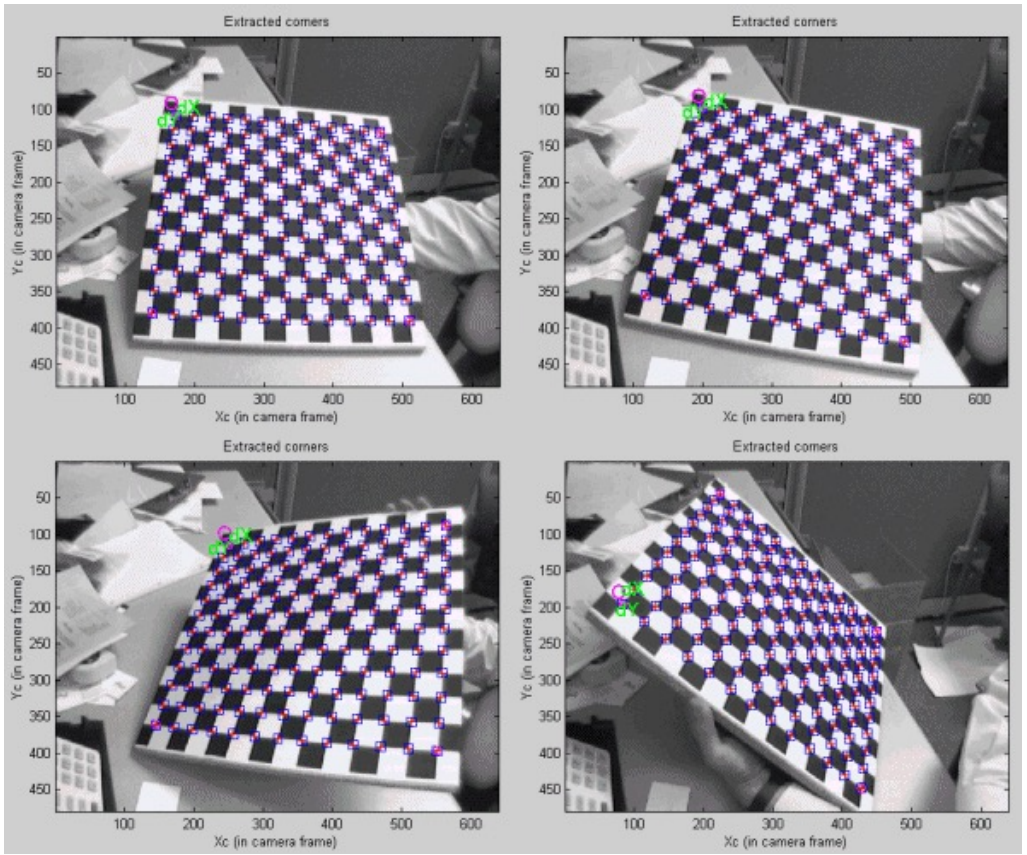
- Works well for most lenses

Camera Calibration

- Finding the *intrinsic parameters* of your camera



Camera Calibration



- Use camera model to interpret the projection from world to image plane
- Using known correspondences of $\boldsymbol{p} \Leftrightarrow \boldsymbol{P}$, we can compute the unknown parameters $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{T}$ by applying the perspective projection equation
- … so associate known, physical distances in the world to pixel-distances in image

Projection Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R | T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

How can we find out the values of the matrices K, R, T?

Camera Calibration | direct linear transform (DLT)

- We know that : $\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R | T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$

- So there are 11 values to estimate:

  (the overall scale doesn't matter, so e.g. m_{34} could be set to 1)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- Each observed point gives us a pair of equations:

$$u_i = \frac{\lambda u_i}{\lambda} = \frac{m_{11}X_i + m_{12}Y_i + m_{13}Z_i + m_{14}}{m_{31} + m_{32} + m_{33} + m_{34}}$$

$$v_i = \frac{\lambda v_i}{\lambda} = \frac{m_{21}X_i + m_{22}Y_i + m_{23}Z_i + m_{24}}{m_{31} + m_{32} + m_{33} + m_{34}}$$

- To estimate 11 unknowns, we need **at least 6** points to calibrate the camera ⇨ solved using linear least squares

Camera Calibration | direct linear transform (DLT)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = K[R \mid T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- **what we obtained**: the 3 x 4 projection matrix,

  what we need: its decomposition into the camera calibration matrix $\boldsymbol{K}$, and the rotation $\boldsymbol{R}$ and position $\boldsymbol{T}$ of the camera.

- Use QR factorization to decompose the 3x3 submatrix (m_{11}:m_{33}) into the product of an upper triangular matrix $\boldsymbol{K}$ and a rotation matrix $\boldsymbol{R}$ (orthogonal matrix)

- The translation $\boldsymbol{T}$ can subsequently be obtained by: $T = K^{-1} \begin{bmatrix} m_{14} \\ m_{24} \\ m_{34} \end{bmatrix}$

Today's Topics

Section 4.2 in the book : Fundamentals of Computer Vision

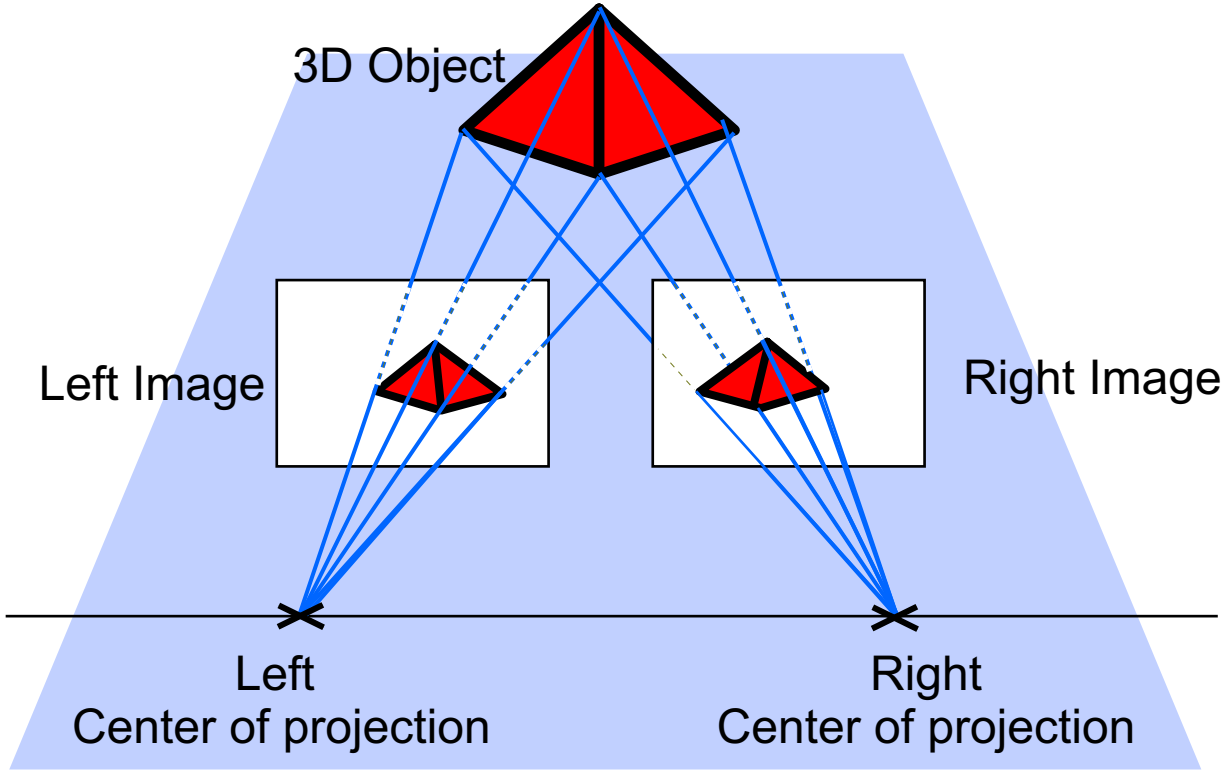
- Pinhole Camera Model
- Perspective Projection
- Camera Calibration
- Stereo Vision, Epipolar Geometry & Triangulation
- Structure from Motion & Optical Flow

Additional, optional reading on Computer Vision: "Computer Vision: Algorithms and Applications", by Richard Szeliski (Springer)

How do we measure distances with cameras?

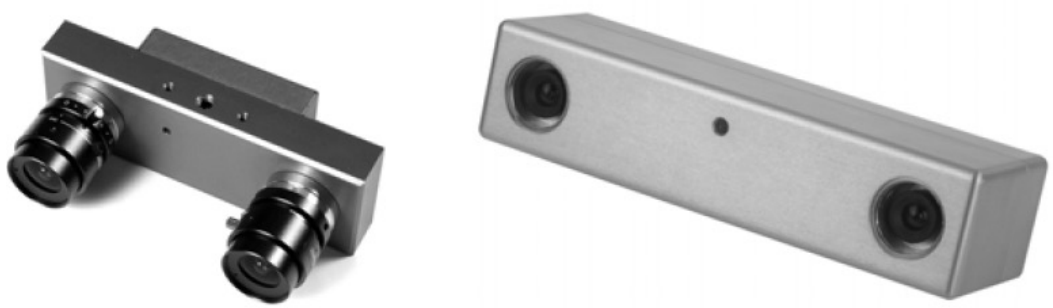
How do we measure distances with cameras?

- From a single image: we can only deduct the **ray** along which each image-point lies



How do we measure distances with cameras?

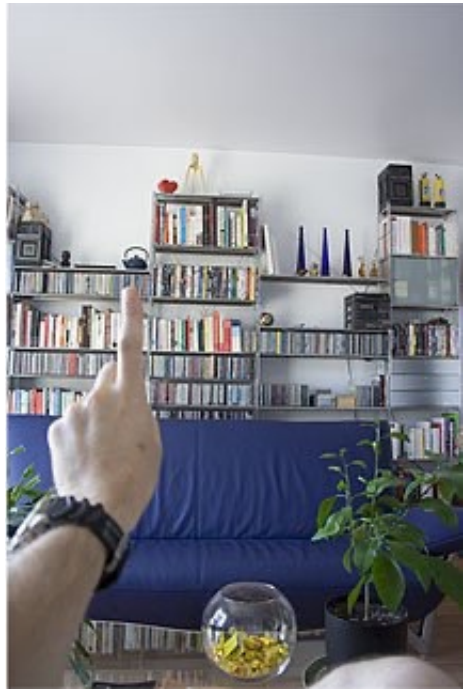
- **Stereo vision:**
  using 2 cameras with **known** relative position $\boldsymbol{T}$ and orientation $\boldsymbol{R}$, recover the **3D scene information**

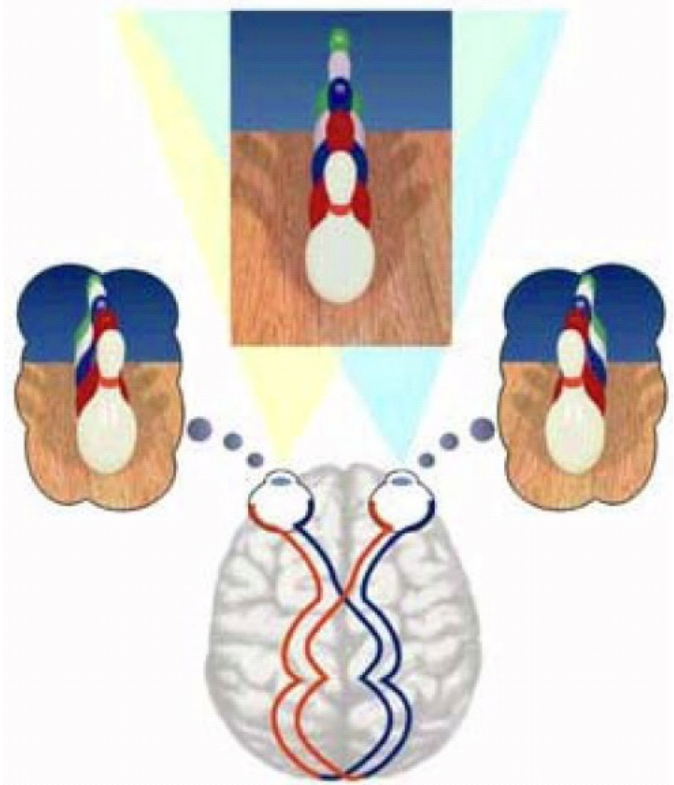


- **Structure from Motion:**
  recover the **3D scene structure & the camera poses** (up to scale) from multiple images, from potentially **unknown** cameras

Disparity in the human retina

- **Stereopsys:** the brain allows us to see the left and right retinal images as a single 3D image
- Observe ***image disparity***





Stereo Vision | simplified case

- An ideal, simplified case: assume both cameras are **identical** and are **aligned** with the x-axis

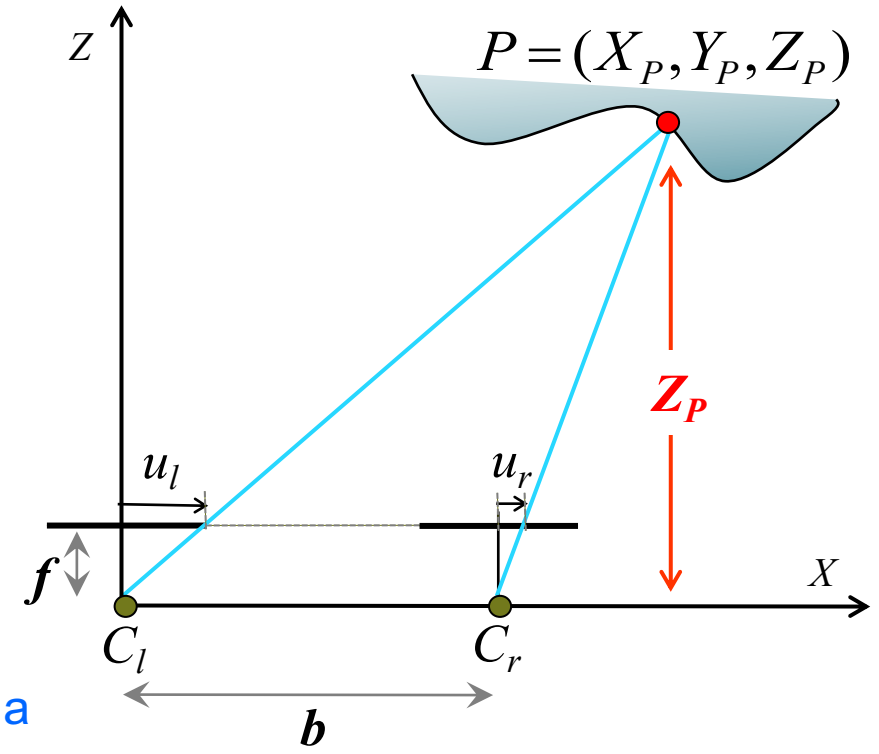
From Similar Triangles:

$$\frac{f}{Z_P} = \frac{u_l}{X_P}$$

$$\frac{f}{Z_P} = \frac{u_r}{X_P - b}$$

$$Z_P = \frac{bf}{u_l - u_r}$$

Disparity
difference in image location of the projection of a 3D point in two image planes



Baseline
distance between the optical centers of the two cameras

Stereo Vision | disparity map

- **Disparity:** the difference in image location of corresponding 3D points as projected to the left & right images

- **Disparity map** holds the disparity value at every pixel:
  - Identify correspondent points of **all image pixels** in the original images
  - Compute the disparity for each pair of correspondences

Left image

Right image

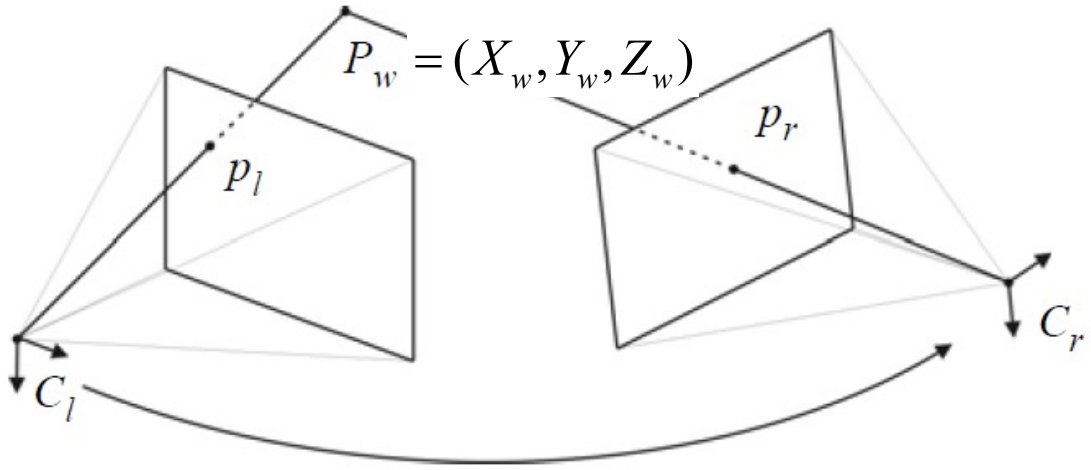
- Near-by objects experience bigger disparity:

$$Z_P = \frac{bf}{u_l - u_r}$$

Stereo Vision | general case

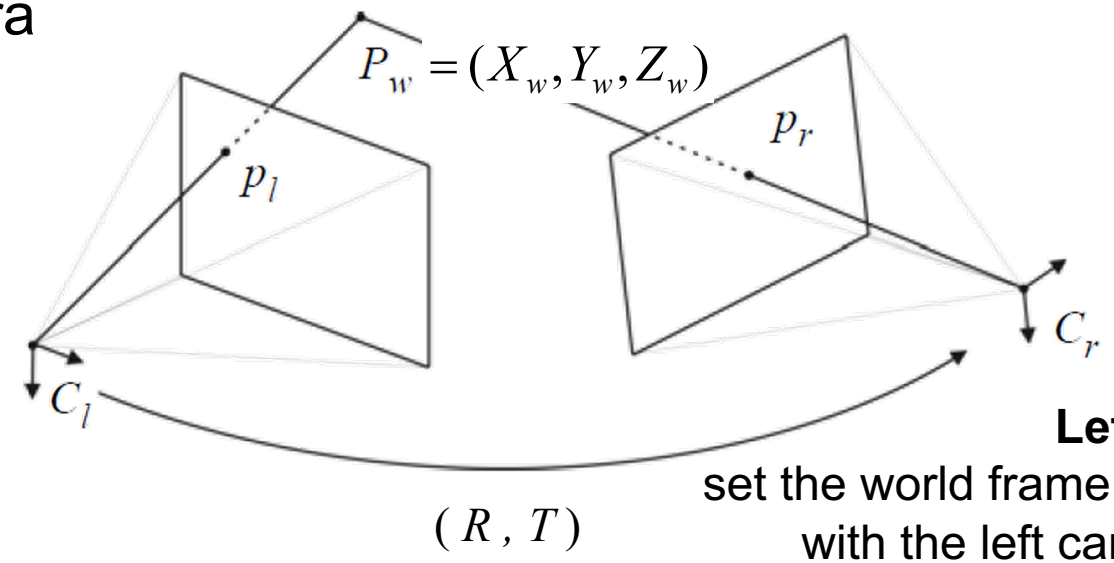
- Two identical cameras do not exist in nature!
- Aligning both cameras on a horizontal axis is very difficult
- In order to use a stereo camera, we need:
  - **relative pose** between the cameras (rotation, translation), and ⇨ "extrinsics"
  - the **focal length**, **optical center**, **radial distortion** of each ⇨ "intrinsics"

  ⇨ Use calibration method



Stereo Vision | general case

- To estimate the 3D position of P_w we construct the system of equations of the left and right camera



Left camera: set the world frame to coincide with the left camera frame

$$\tilde{p}_l = \lambda_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = K_l \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

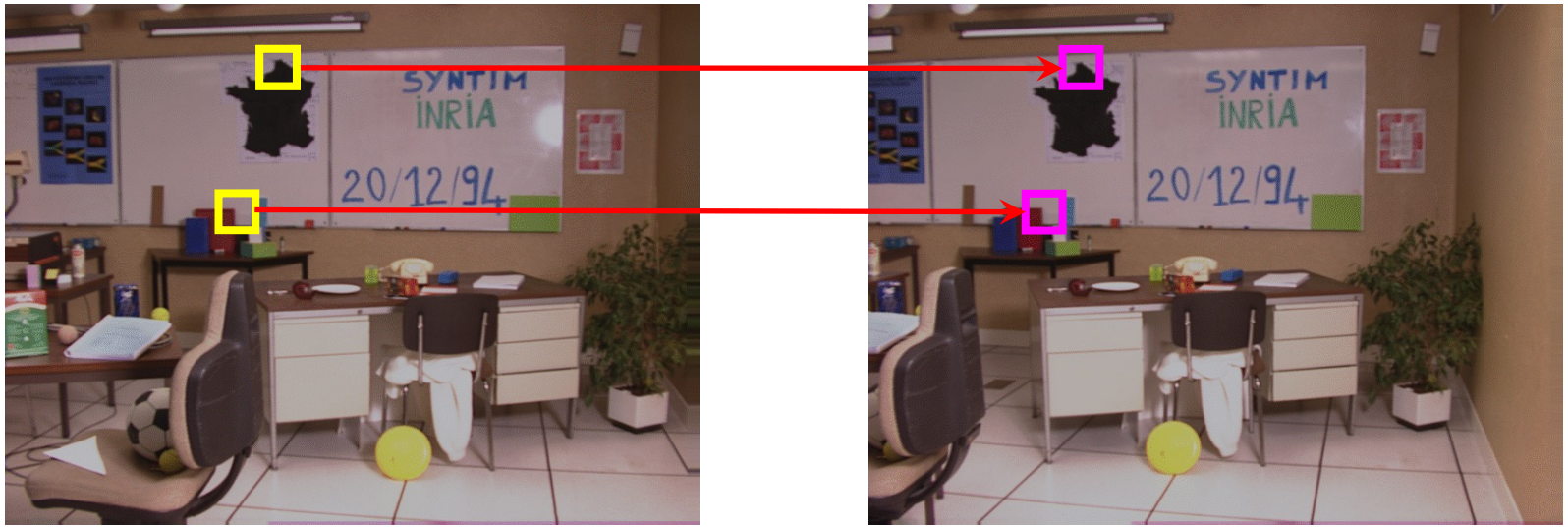
- Triangulation: the problem of determining the 3D position of a point, given a set of *corresponding* image locations & known camera poses.

Right camera:

$$\tilde{p}_r = \lambda_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = K_r \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

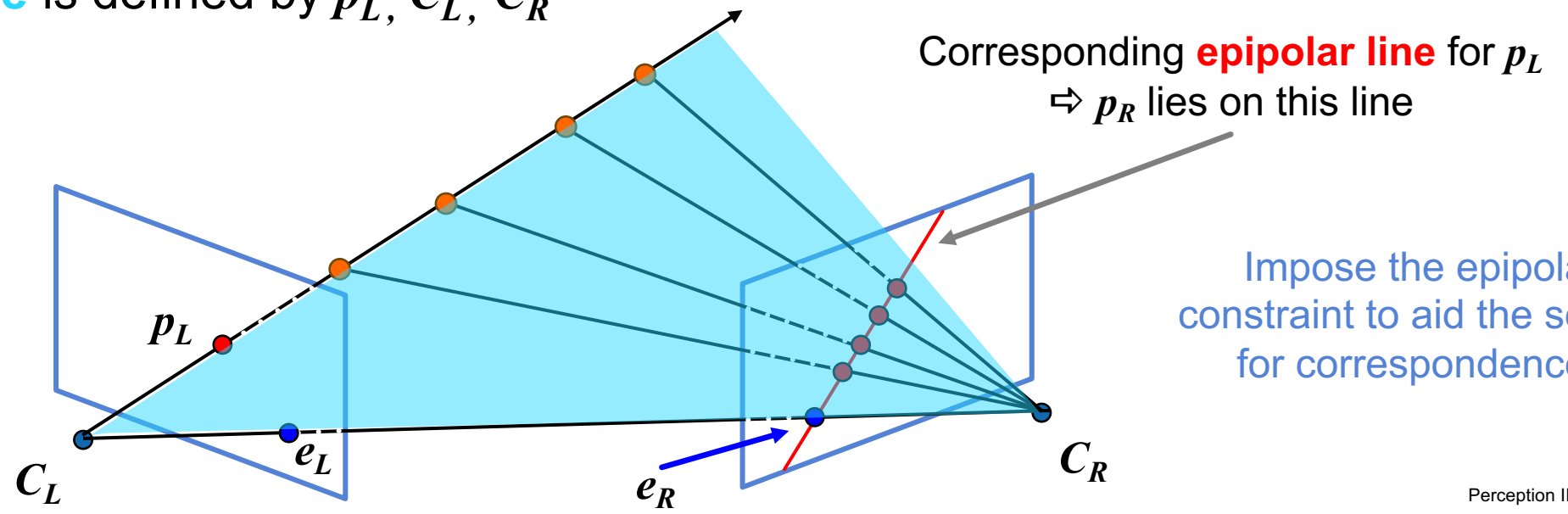
Correspondence Search | the problem

- **goal**: identify image regions / patches in the left & right images, corresponding to the same scene structure
  - Typical **similarity measures**: Normalized Cross-Correlation (NCC) , Sum of Squared Differences (SSD), Sum of Absolute Differences (SAD), …
  - **Exhaustive** image search can be computationally very expensive!
    Can we search for correspondences more efficiently?



Correspondence Search | the epipolar constraint

- Potential matches for p_L have to lie on the corresponding epipolar line
- **Epipolar line**: the projection of the infinite ray that connects C_L and p_L, onto the image plane of C_R (and vice versa)
- **Epipole**: the projection of the optical center C_L onto the image plane of C_R (and vice versa)
- **Epipolar plane** is defined by p_L, C_L, C_R

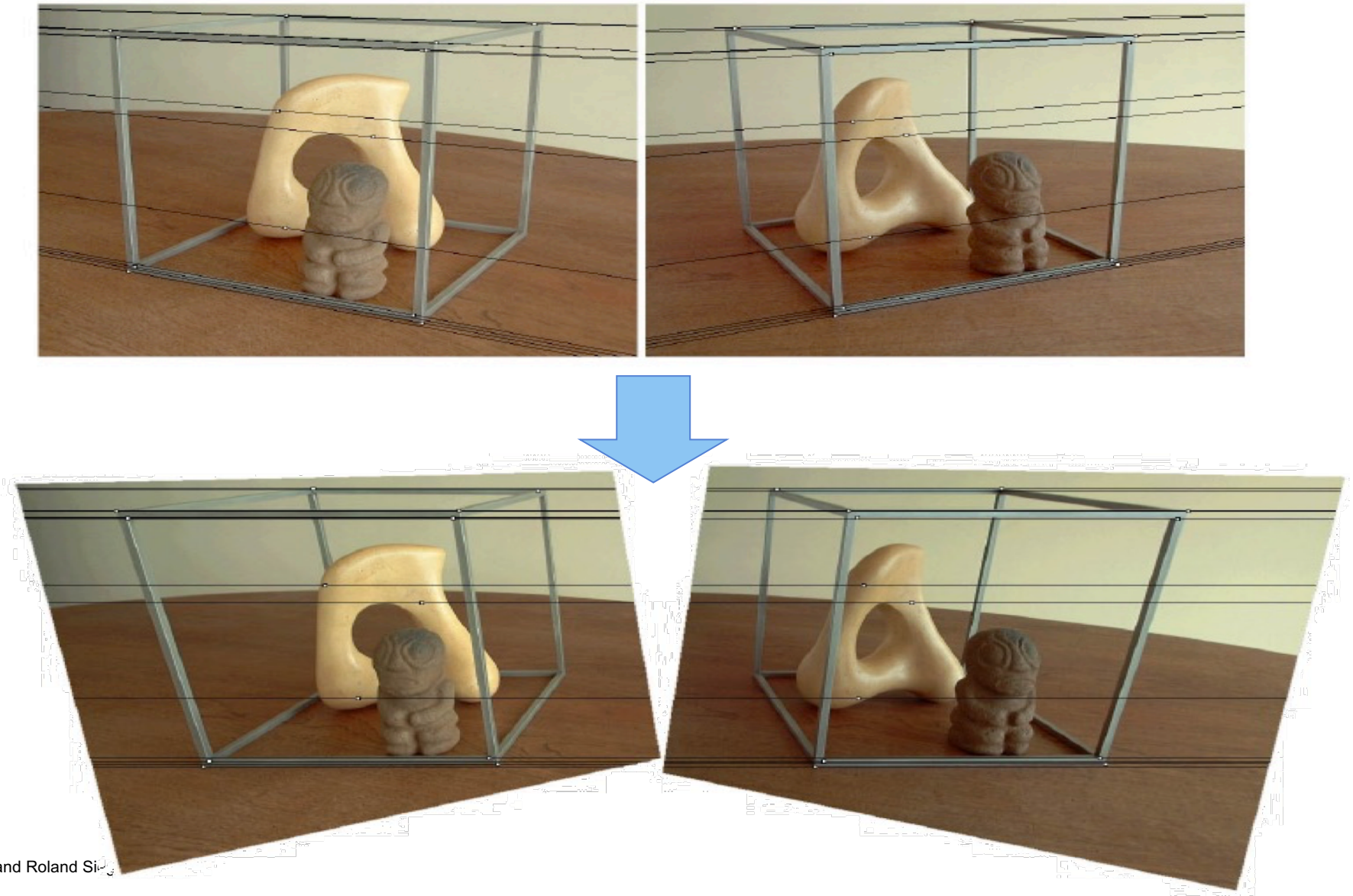


Impose the epipolar constraint to aid the search for correspondences

Correspondence Search | the epipolar constraint

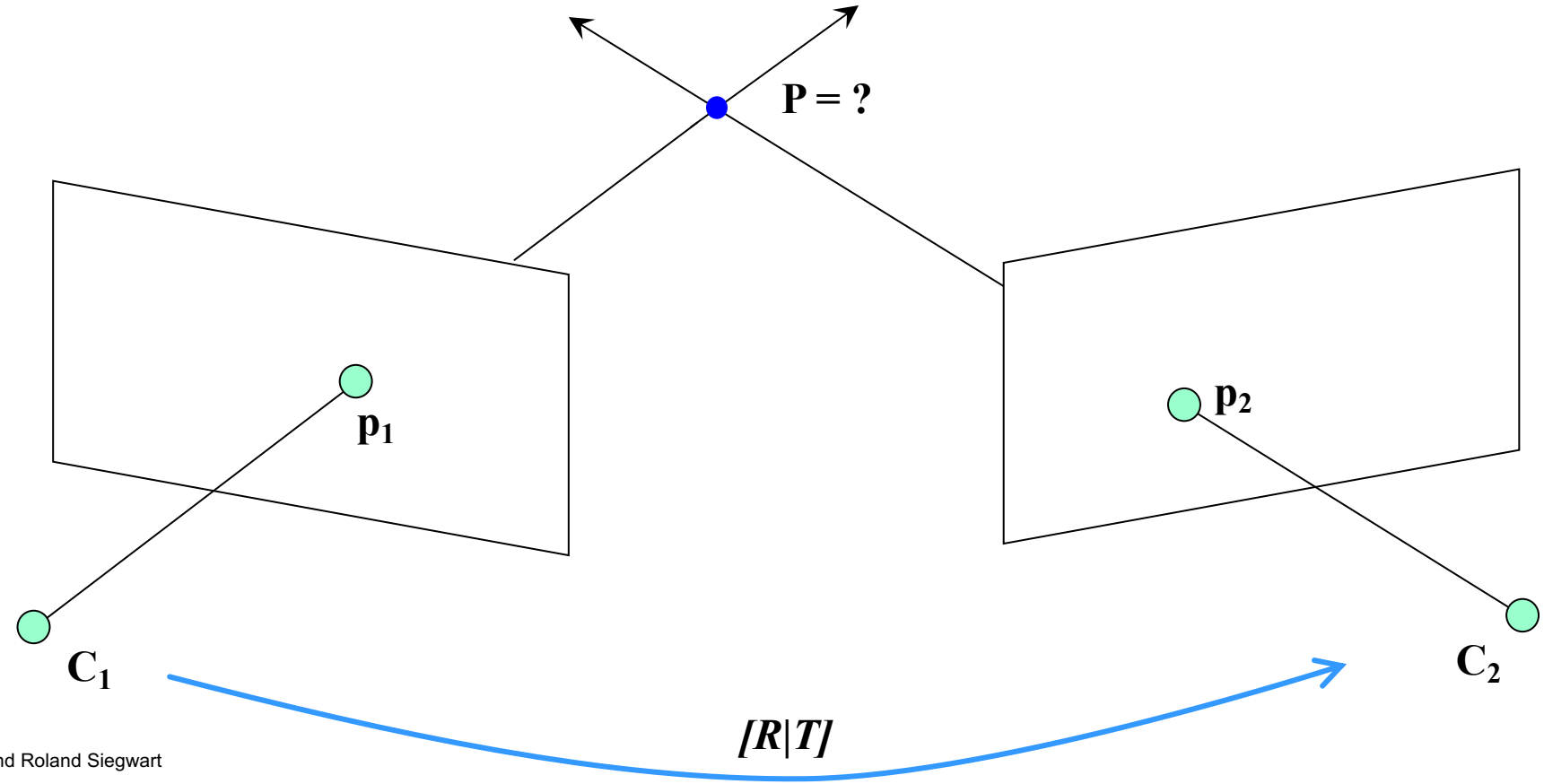
- Thanks to the epipolar constraint, corresponding points can be searched for, along epipolar lines
  ⇒ computational cost reduced to 1 dimension!

Epipolar Rectification | example



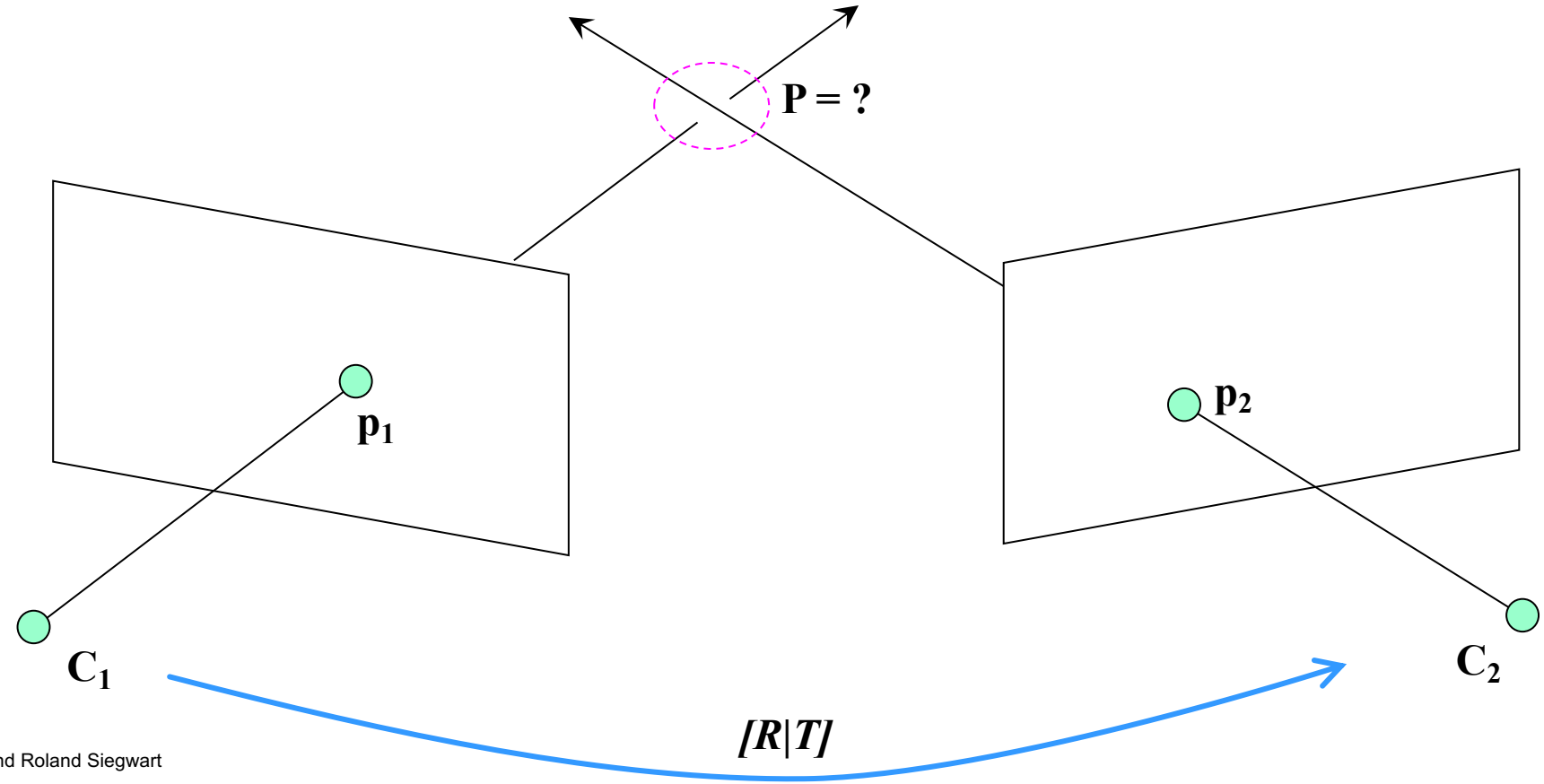
Triangulation

- Given the projections $\mathbf{p}_1$ and $\mathbf{p}_2$ of a 3D point $\mathbf{P}$ in two or more images (with known camera matrices $\boldsymbol{R}$ and $\boldsymbol{T}$), find the coordinates of the 3D point



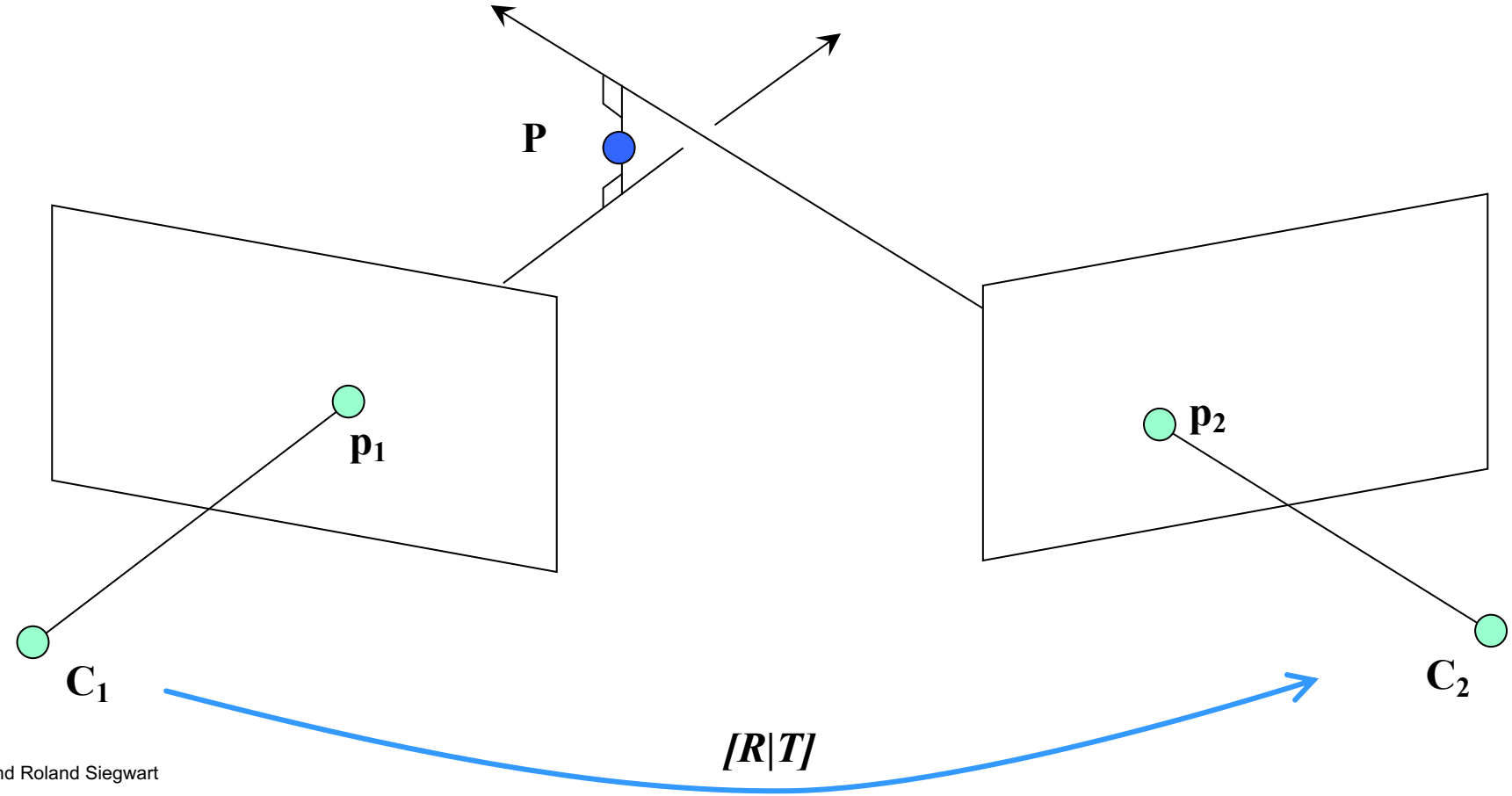
Triangulation

- We want to intersect the two visual rays corresponding to $\mathbf{p}_1$ and $\mathbf{p}_2$, but because of noise and numerical errors, they don't meet exactly



Triangulation | geometric approach

- Find shortest segment connecting the two viewing rays and let P be the midpoint of that segment



Triangulation | linear approach

Left camera:

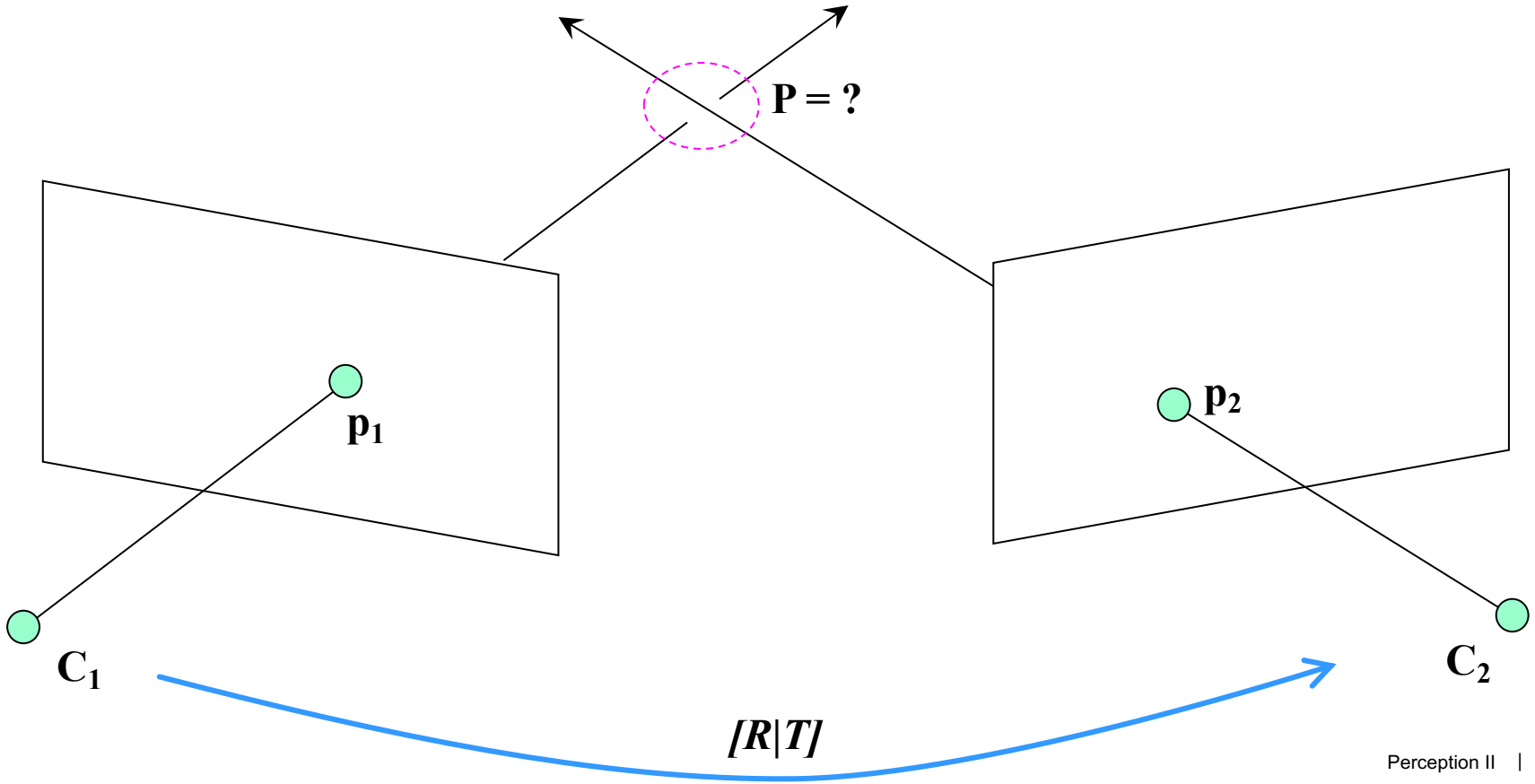
$$\tilde{p}_1 = K_1 \left[I | 0 \right] \tilde{P}$$

M_1

Right Camera:

$$\tilde{p}_2 = K_2 \left[R | T \right] \tilde{P}$$

M_2



Triangulation | linear approach

Left camera:

$$\tilde{p}_1 = M_1 \tilde{P} \qquad \Rightarrow M_1 \tilde{P} - \tilde{p}_1 = 0$$

Right Camera:

$$\tilde{p}_2 = M_2 \tilde{P} \qquad \Rightarrow M_2 \tilde{P} - \tilde{p}_2 = 0$$

Triangulation | linear approach

Left camera:

Cross-multiply by $\tilde{p}_1$

$$\tilde{p}_1 = M_1 \tilde{P} \quad \Rightarrow M_1 \tilde{P} - \tilde{p}_1 = 0 \Rightarrow \tilde{p}_1 \times M_1 \tilde{P} - \tilde{p}_1 \times \tilde{p}_1 = 0$$

$\tilde{p}_1 \times \tilde{p}_1 = 0$

Right Camera:

Cross-multiply by $\tilde{p}_2$

$$\tilde{p}_2 = M_2 \tilde{P} \quad \Rightarrow M_2 \tilde{P} - \tilde{p}_2 = 0 \Rightarrow \tilde{p}_2 \times M_2 \tilde{P} - \tilde{p}_2 \times \tilde{p}_2 = 0$$

$\tilde{p}_2 \times \tilde{p}_2 = 0$

Triangulation | linear approach

Left camera:

$$\tilde{p}_1 = M_1 \tilde{P} \qquad \Rightarrow M_1 \tilde{P} - \tilde{p}_1 = 0 \Rightarrow \tilde{p}_1 \times M_1 \tilde{P} = 0 \Rightarrow \left[\tilde{p}_{1\times}\right] M_1 \tilde{P} = 0$$

Right Camera:

$$\tilde{p}_2 = M_2 \tilde{P} \qquad \Rightarrow M_2 \tilde{P} - \tilde{p}_2 = 0 \Rightarrow \tilde{p}_2 \times M_2 \tilde{P} = 0 \Rightarrow \left[\tilde{p}_{2\times}\right] M_2 \tilde{P} = 0$$

Cross product as matrix multiplication:

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

Triangulation | linear approach

Left camera:

$$\tilde{p}_1 = M_1 \tilde{P} \quad \Rightarrow M_1 \tilde{P} - \tilde{p}_1 = 0 \Rightarrow \tilde{p}_1 \times M_1 \tilde{P} = 0 \Rightarrow \left[\tilde{p}_{1\times}\right] M_1 \tilde{P} = 0$$

Right Camera:

$$\tilde{p}_2 = M_2 \tilde{P} \quad \Rightarrow M_2 \tilde{P} - \tilde{p}_2 = 0 \Rightarrow \tilde{p}_2 \times M_2 \tilde{P} = 0 \Rightarrow \left[\tilde{p}_{2\times}\right] M_2 \tilde{P} = 0$$

3 unknowns (P) : X_w, Y_w, Z_w

6 independent equations

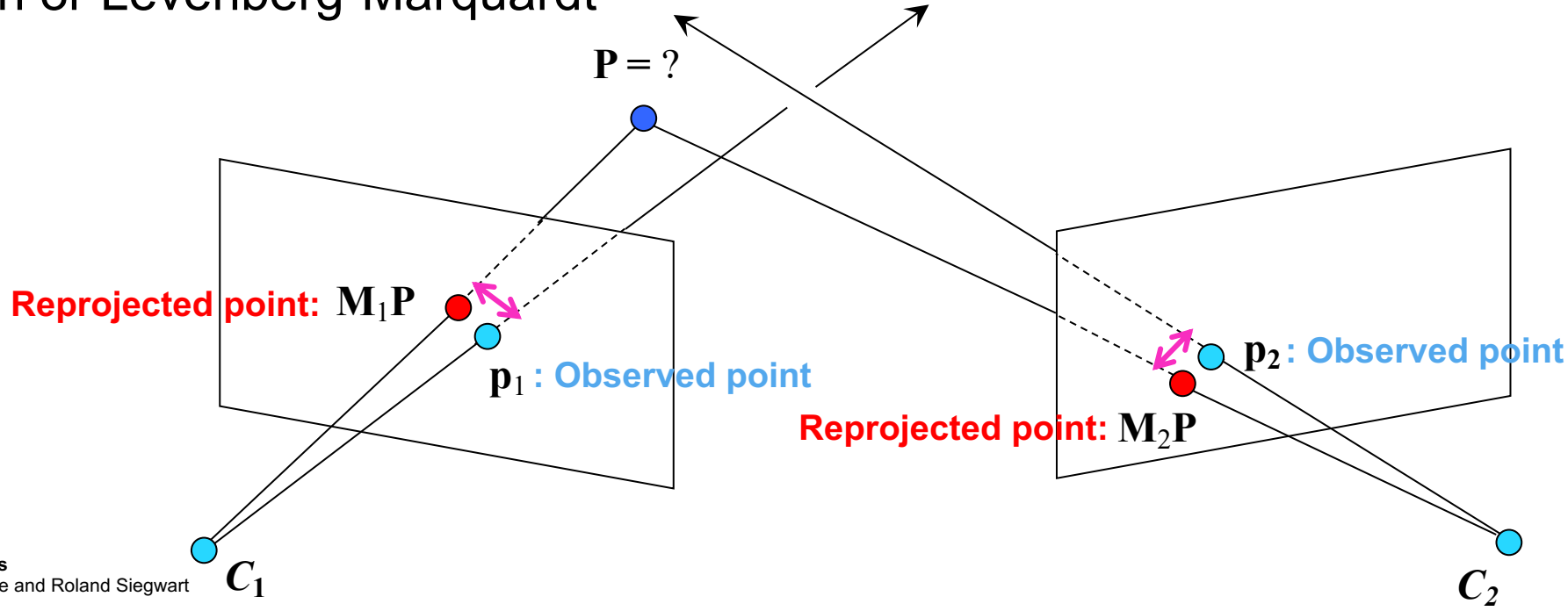
⇒ Solve system: $A \cdot P = 0$ using SVD

Triangulation | non-linear approach

- Find $\boldsymbol{P}$ that minimizes the Sum of Squared **Reprojection Error**

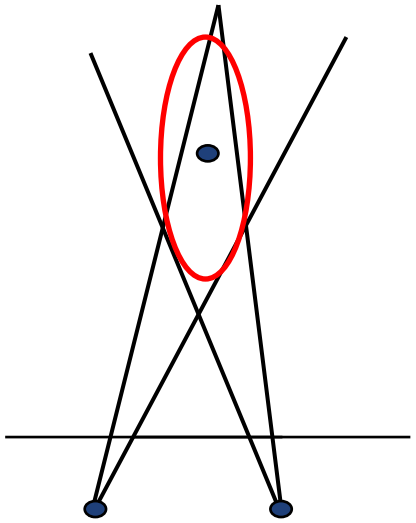
$$SSRE = \left\| p_1 - M_1 P \right\|^2 + \left\| p_2 - M_2 P \right\|^2$$

- In practice: initialize $\boldsymbol{P}$ using linear approach and then minimize SSRE using Gauss-Newton or Levenberg-Marquardt

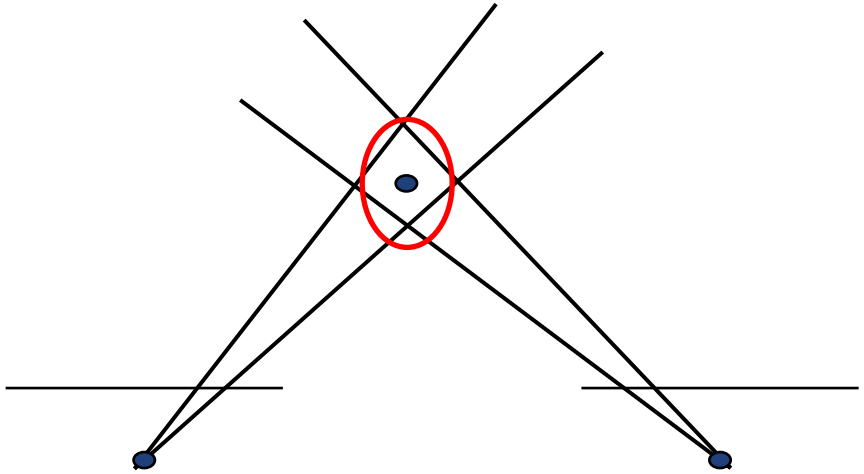


Triangulation | sensitivity to baseline

- What's the optimal baseline?
  - **Too small:**
    - Large depth error
    - Can you quantify the error as a function of the disparity?
  - **Too large:**
    - Object might be visible from one camera, but not the other
    - Difficult search problem for close objects



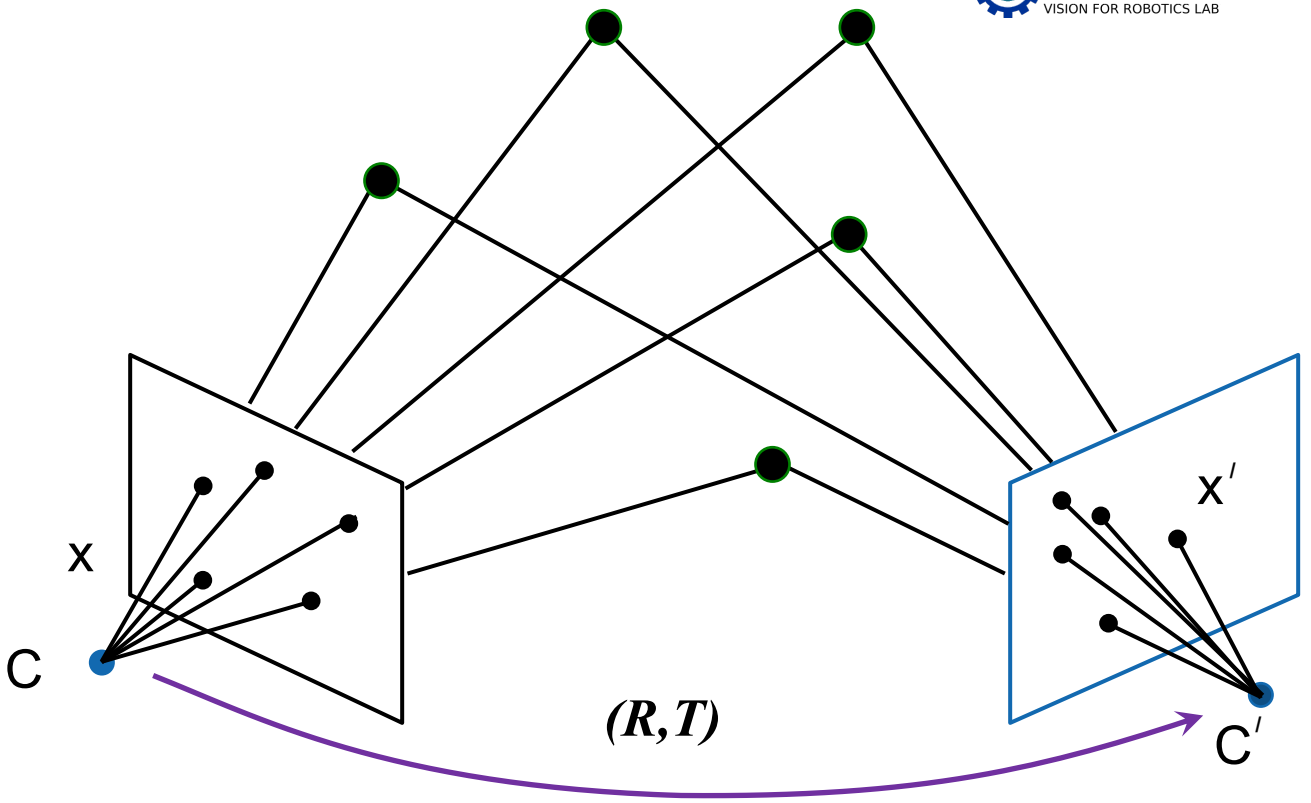
Small Baseline



Large Baseline

SFM: Structure From Motion

- Given image point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$, determine $\boldsymbol{R}$ and $\boldsymbol{T}$



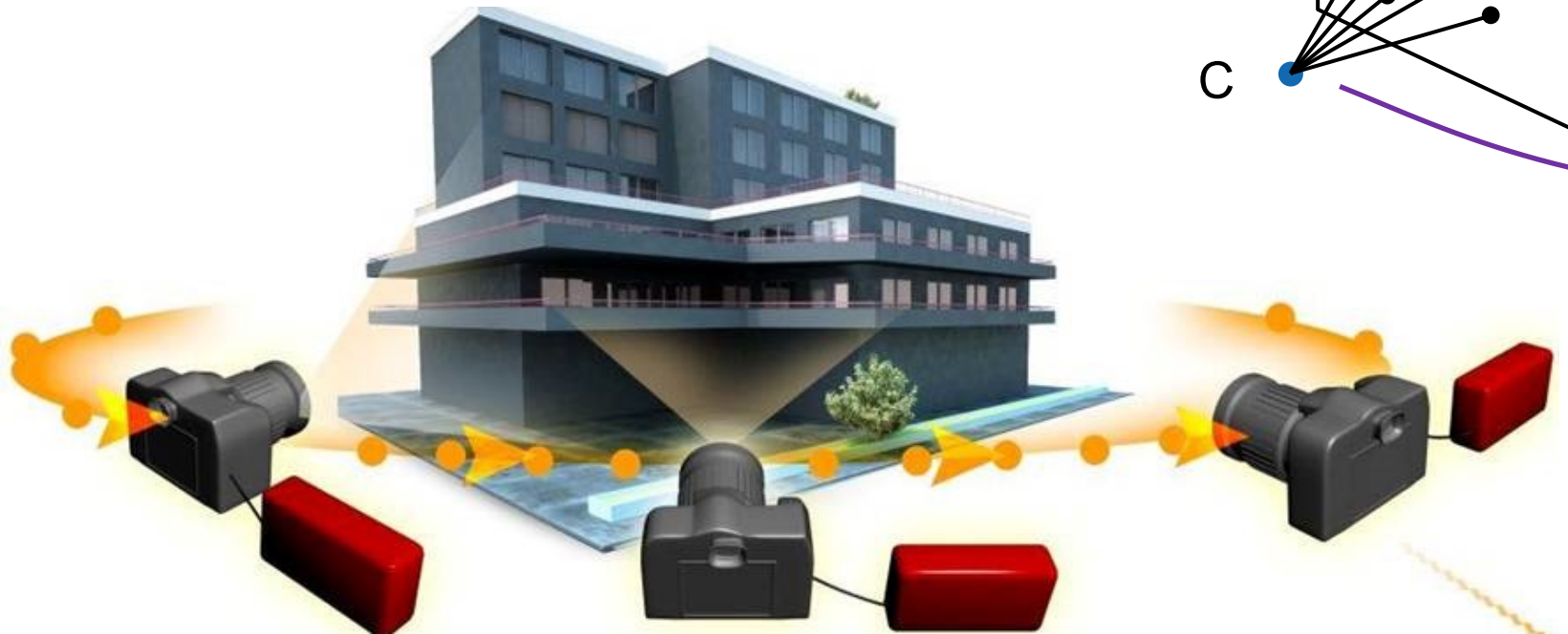
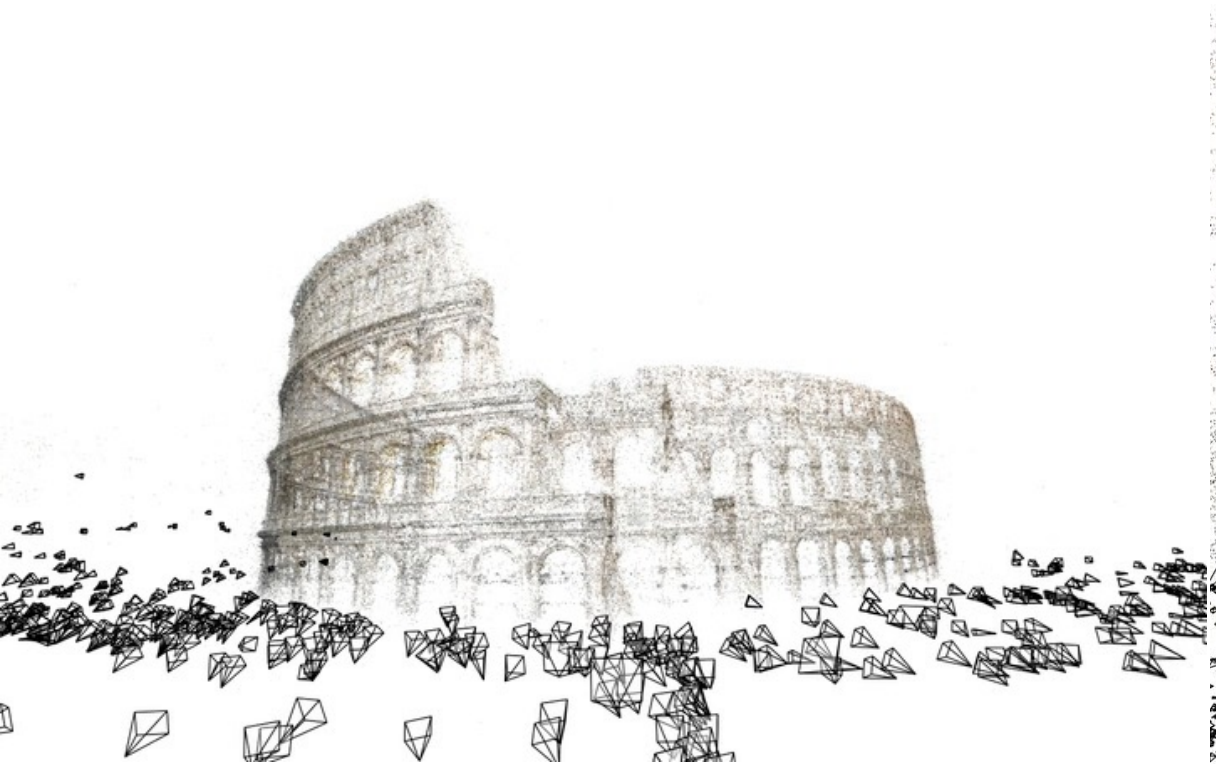


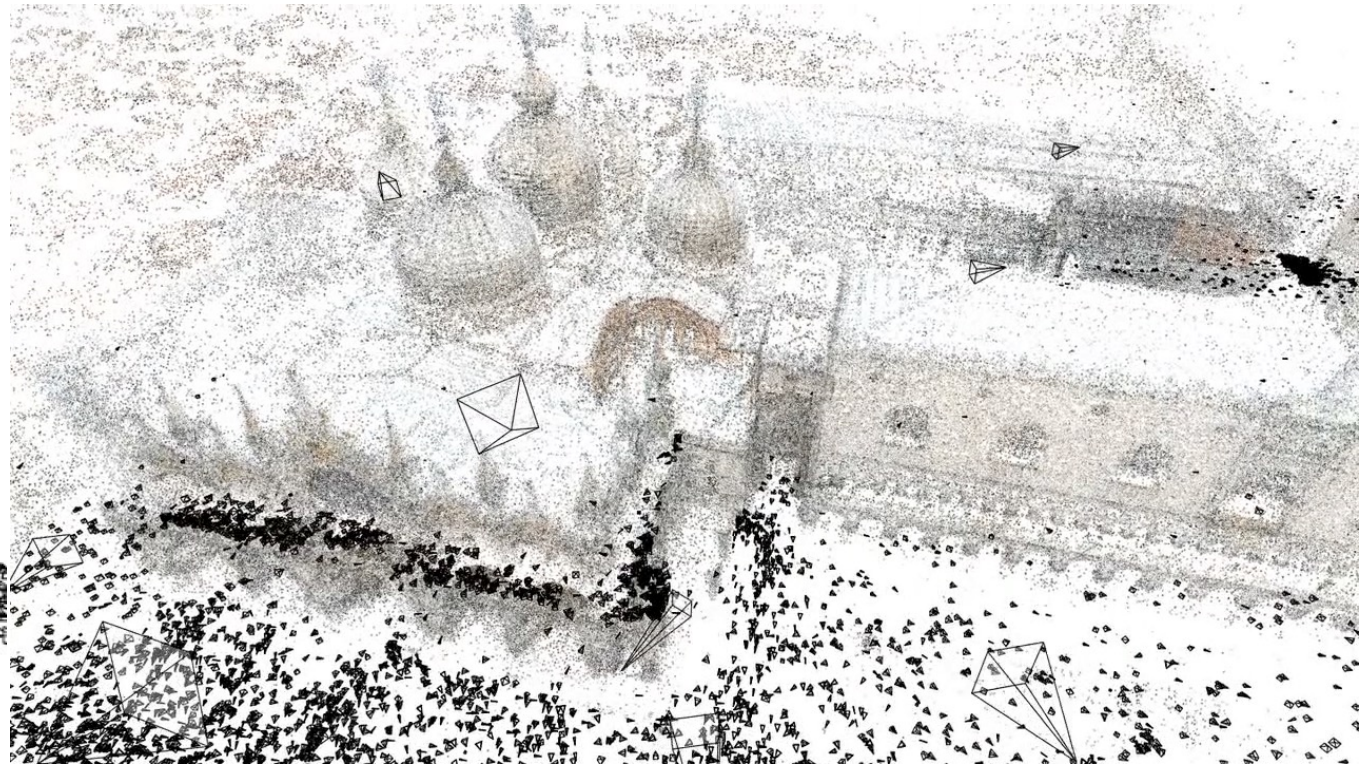
Image courtesy of Nader Salman

Multi-view SFM

- Results of Structure from motion from user images from flickr.com [Seitz & Szeliski, ICCV 2009]

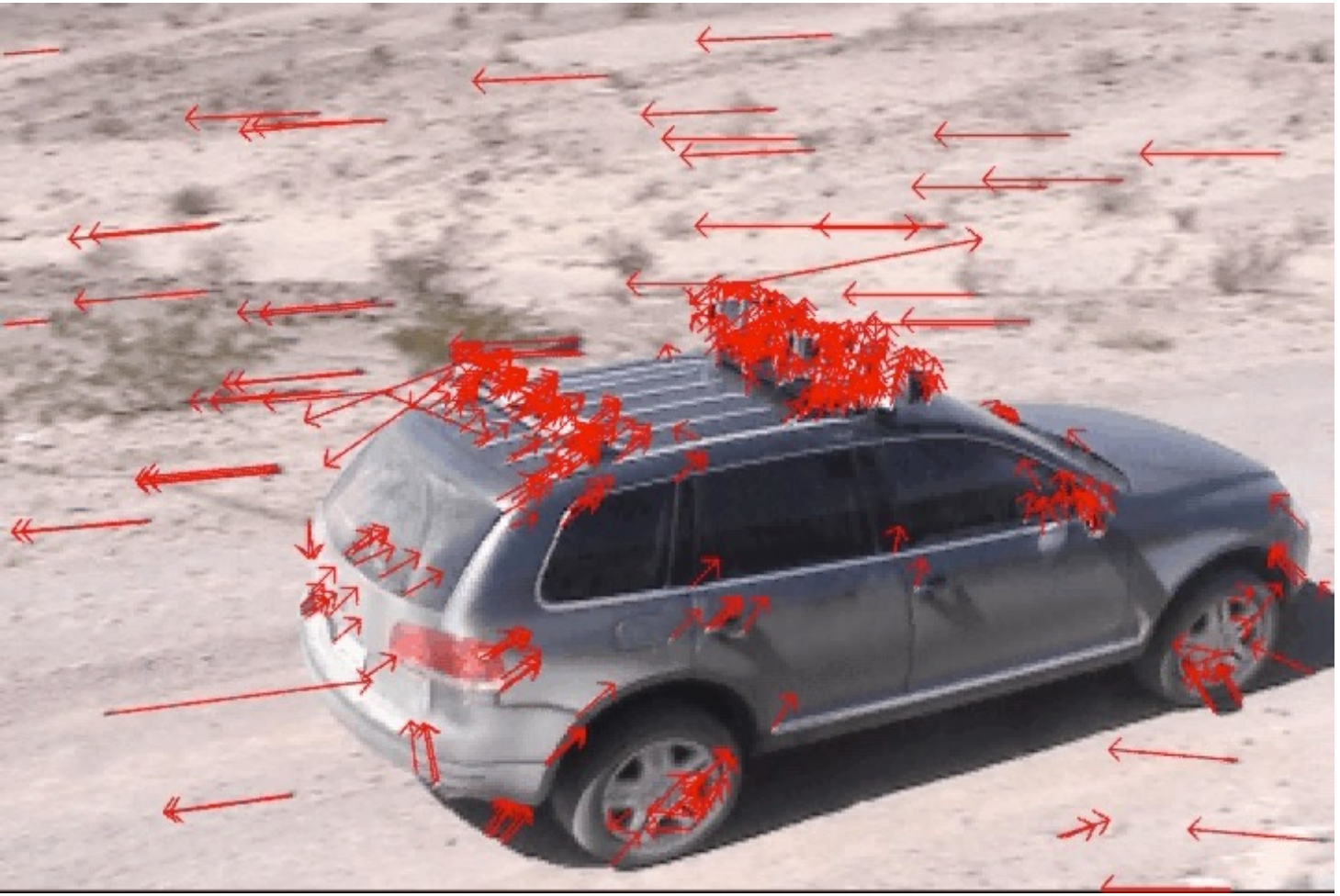


Colosseum, Rome
2,106 images, 819,242 points

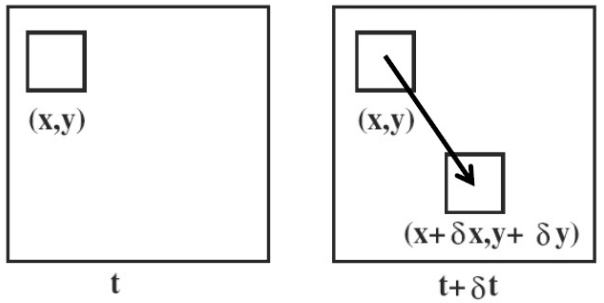


San Marco square, Venice
14,079 images, 4,515,157 points

Optical Flow -- the most general (and challenging) version of motion estimation: compute the motion of *each* pixel



- **Optical flow:** vector field representing motion of parts of an image in subsequent frames
- It computes the motion vectors of pixels in the image (or a subset of them to be faster)



by minimizing the brightness/color difference between corresponding pixels in the image

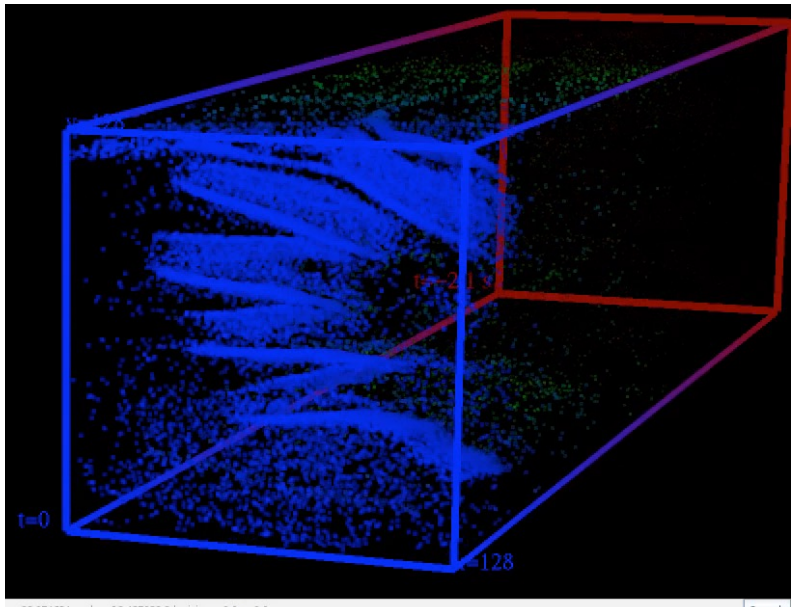
Event-based Cameras

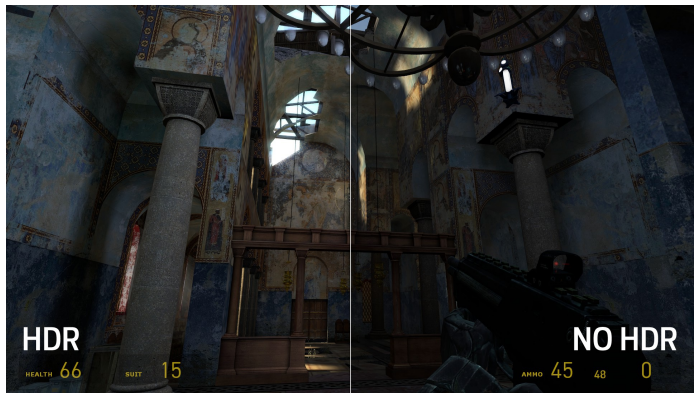
V4RL VISION FOR ROBOTICS LAB

- Dynamic Vision Sensor (DVS)
- Similar to the human retina:

  captures intensity changes asynchronously instead of capturing image frames at a fixed rate

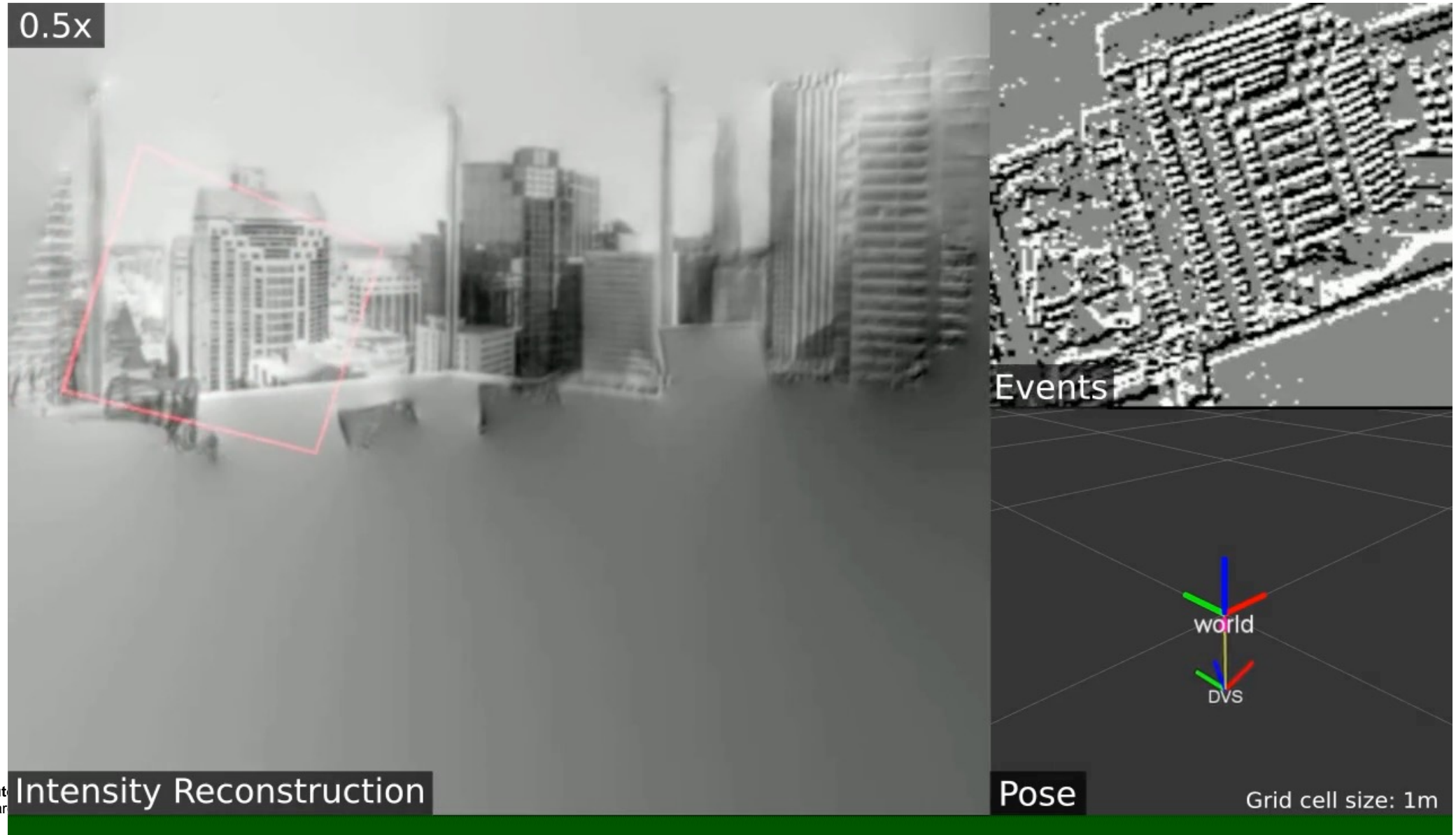
  - ✓ Low power
  - ✓ High temporal resolution ➔ tackle motion blur
  - ✓ High dynamic range



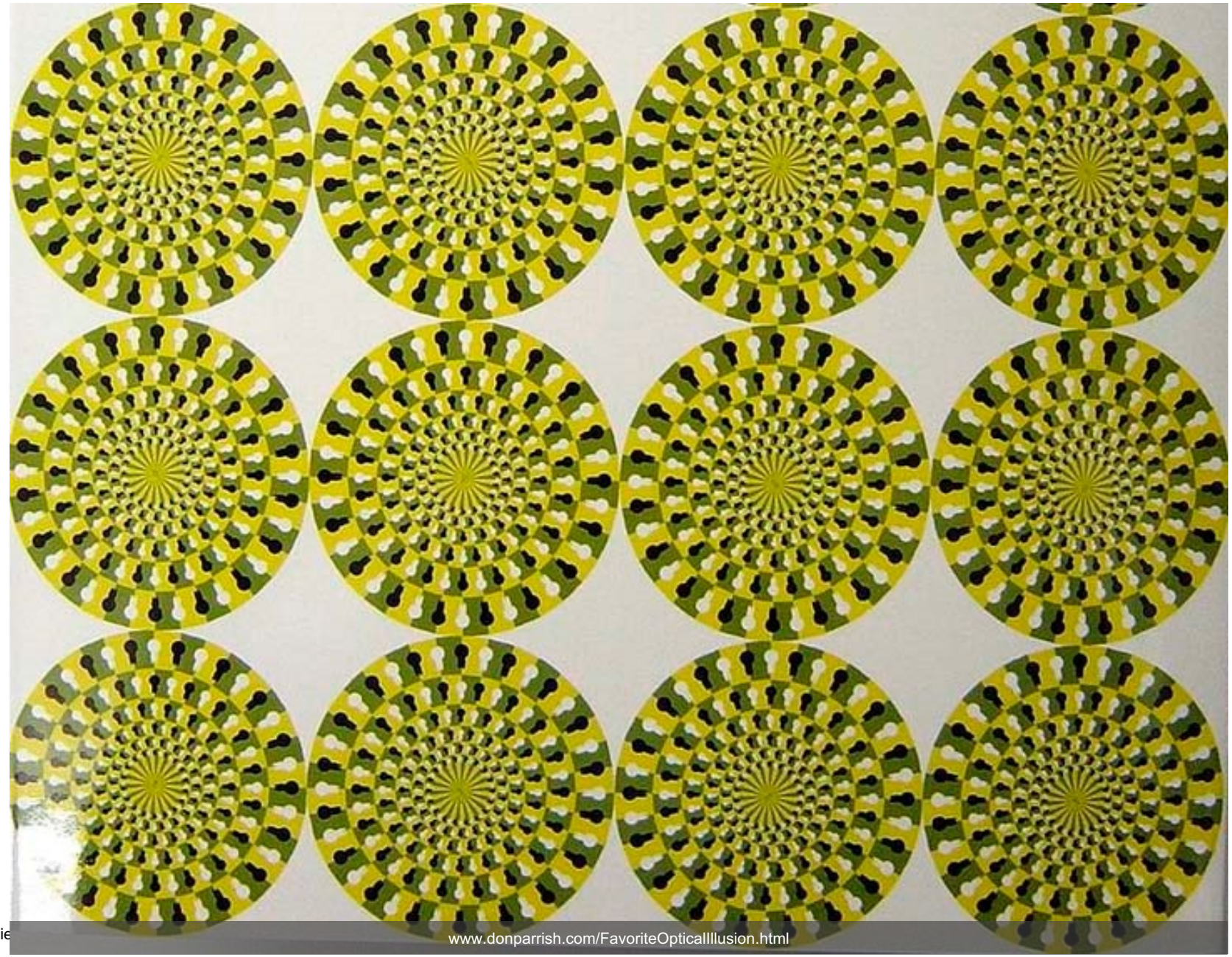


Work with Wilko Schwarting and Dragos Stanciu

Event-based Cameras for Robot Navigation



Work with I. Alzugaray, W. Schwarting & D. Stanciu



www.donparrish.com/FavoriteOpticalIllusion.html