



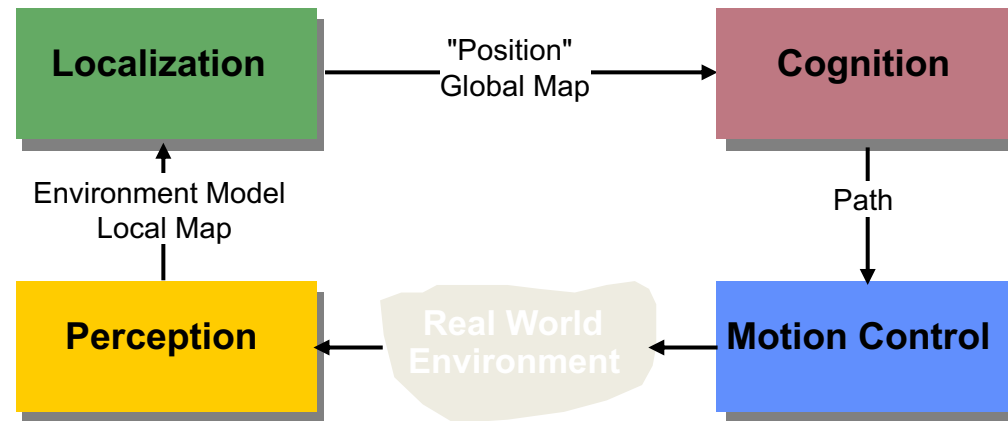
上海科技大学
ShanghaiTech University

CS283: Robotics Fall 2020: Vision

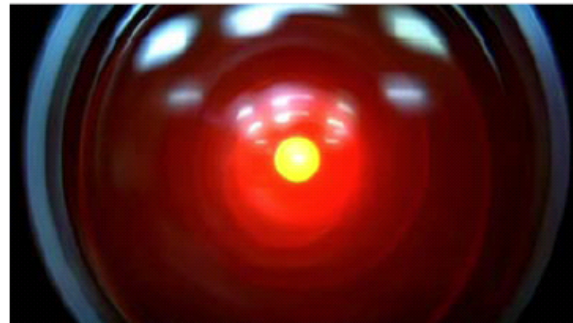
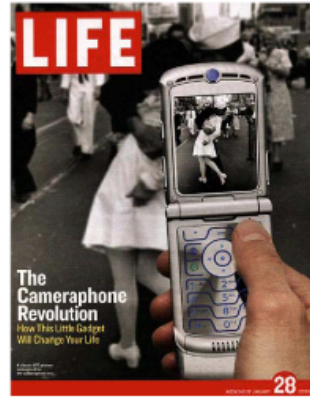
Qingwen Xu Sören Schwertfeger

ShanghaiTech University

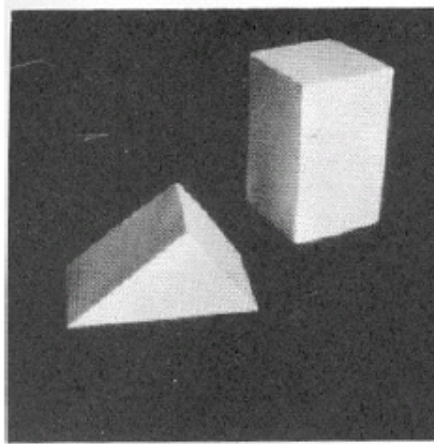
VISION



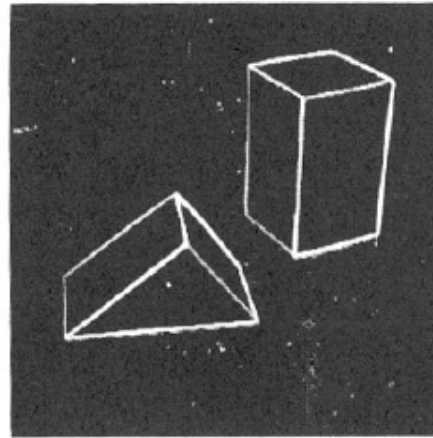
Computer Vision



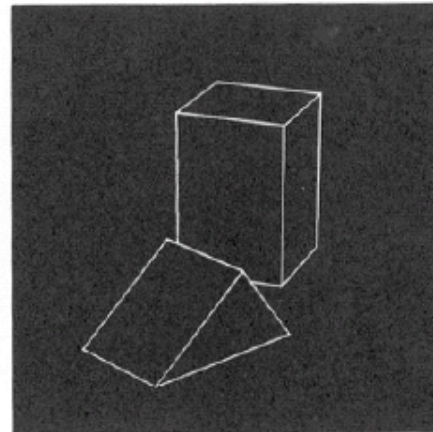
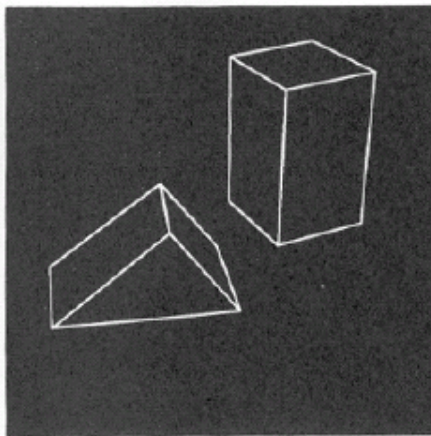
Origins of Computer Vision



(a) Original picture.

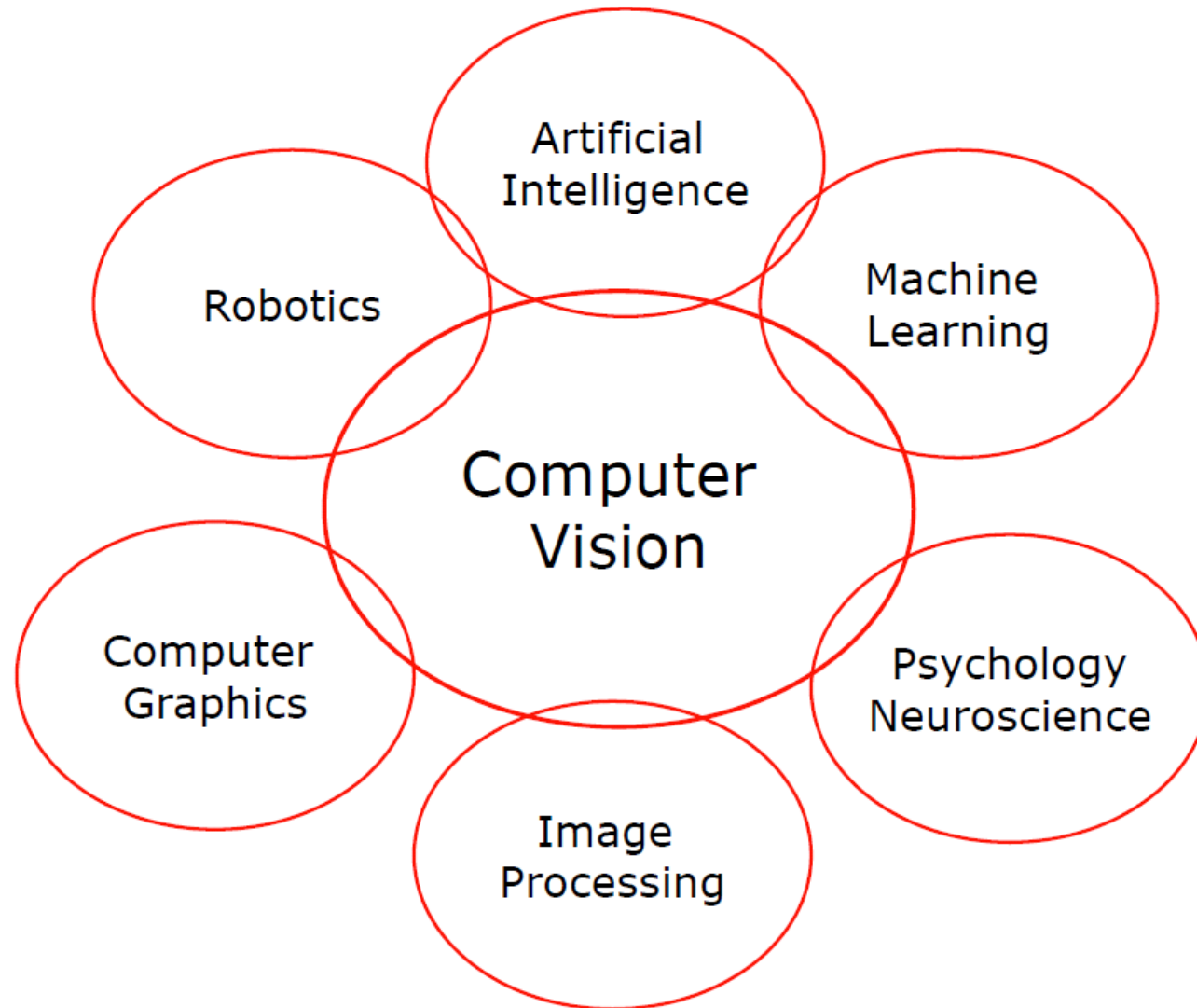


(b) Differentiated picture.



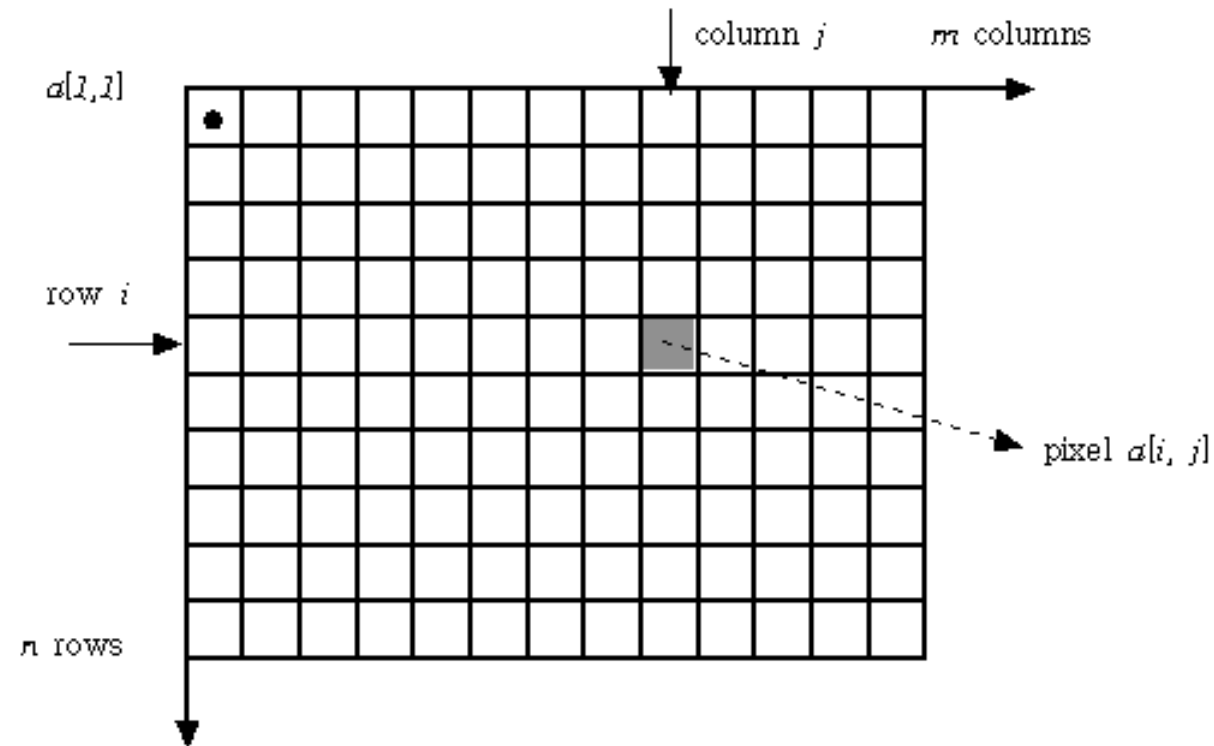
L. G. Roberts, *Machine Perception of Three Dimensional Solids*, Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

Connection to other disciplines



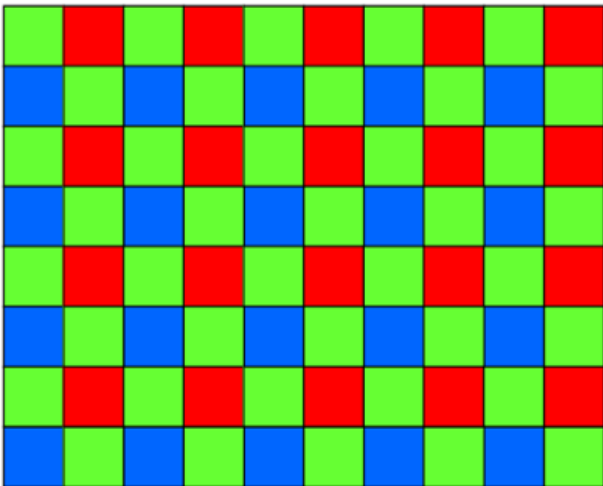
Image

- Image : a two-dimensional array of pixels
- The indices $[i, j]$ of pixels : integer values that specify the rows and columns in pixel values

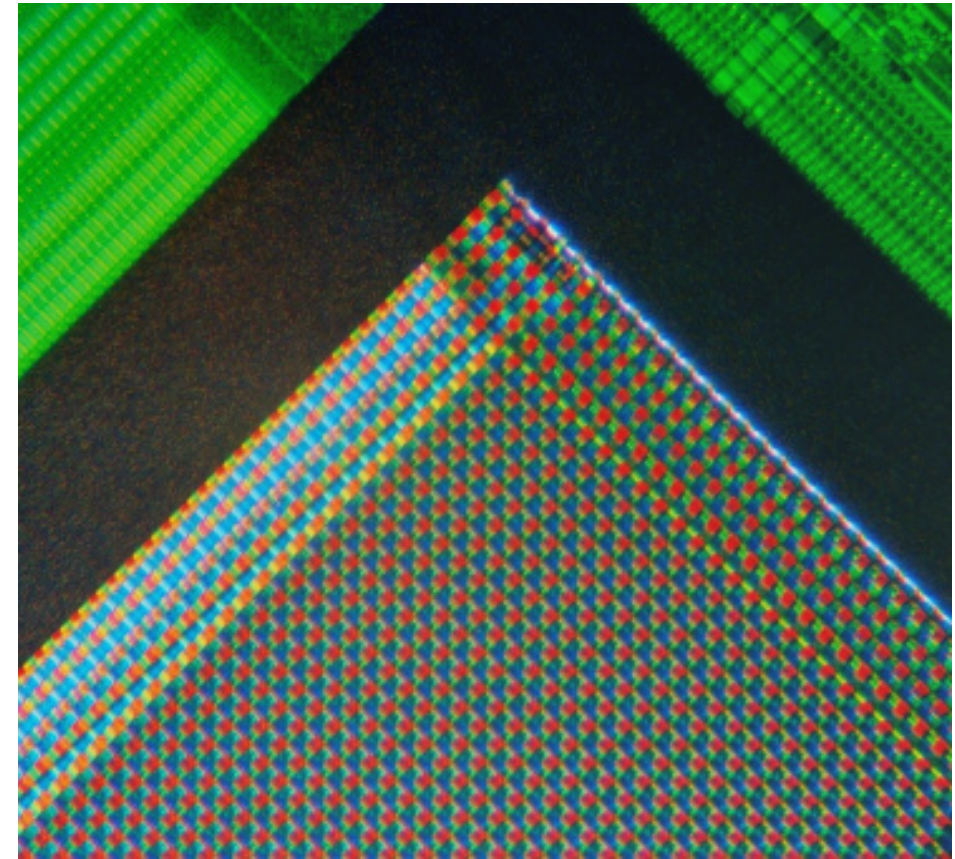


Digital Color Camera

- Bayer Pattern:
 - 50% green, 25% red and 25% blue =>
 - RGBG or GRGB or RGGB.
 - 1 Byte per square
 - 4 squared per 1 pixel
 - More green: eyes are more sensitive to green (nature!)

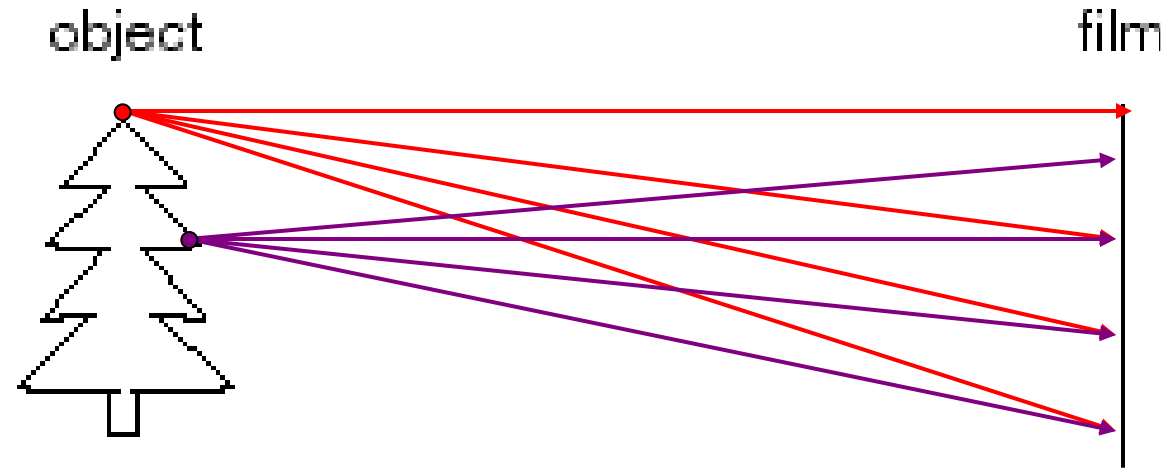


https://en.wikipedia.org/wiki/Bayer_filter



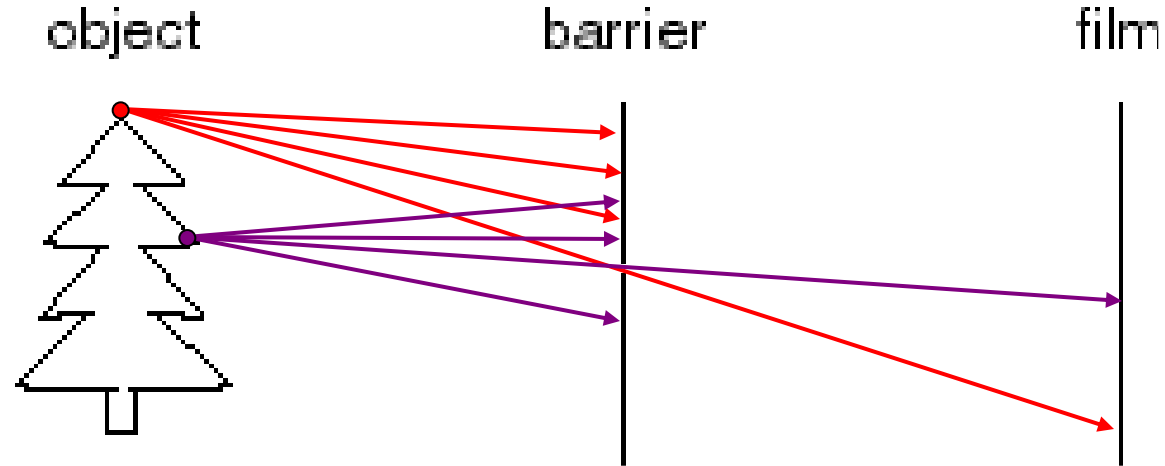
A micrograph of the corner of the photosensor array of a 'webcam' digital camera.
(Wikimedia)

How do we see the world?



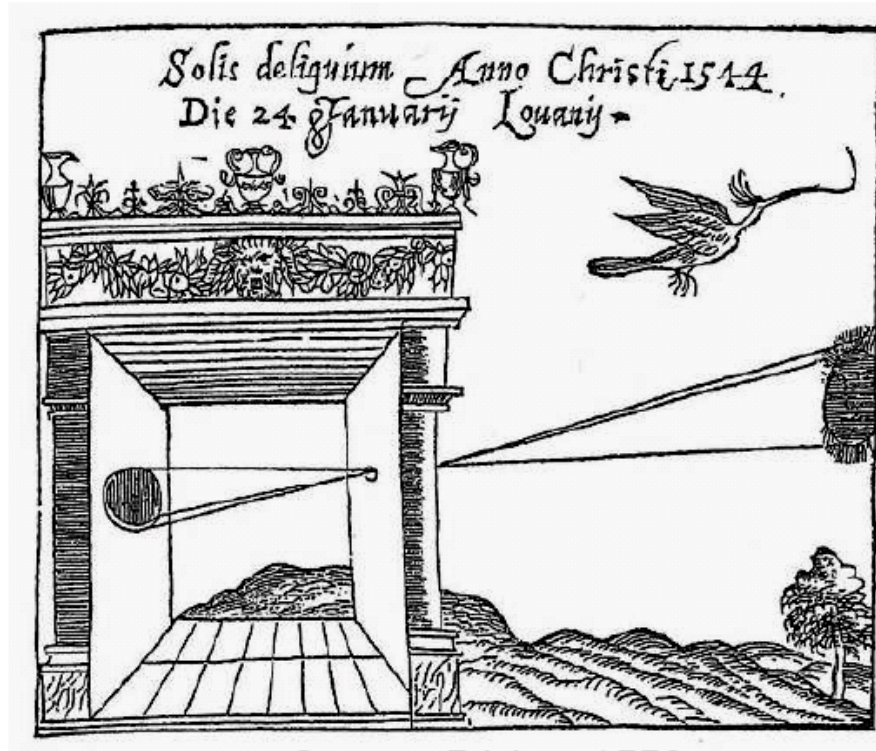
- Let's design a camera
 - Idea 1: put a piece of film in front of an object
 - Do we get a reasonable image?

Pinhole camera



- Add a barrier to block off most of the rays
 - This reduces blurring
 - The opening known as the **aperture**

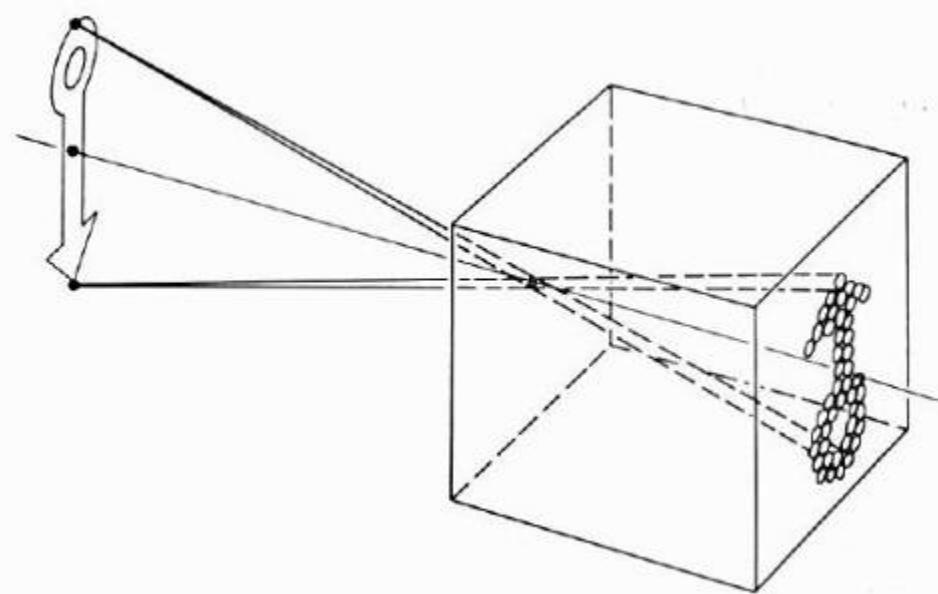
Camera obscura



Gemma Frisius, 1558

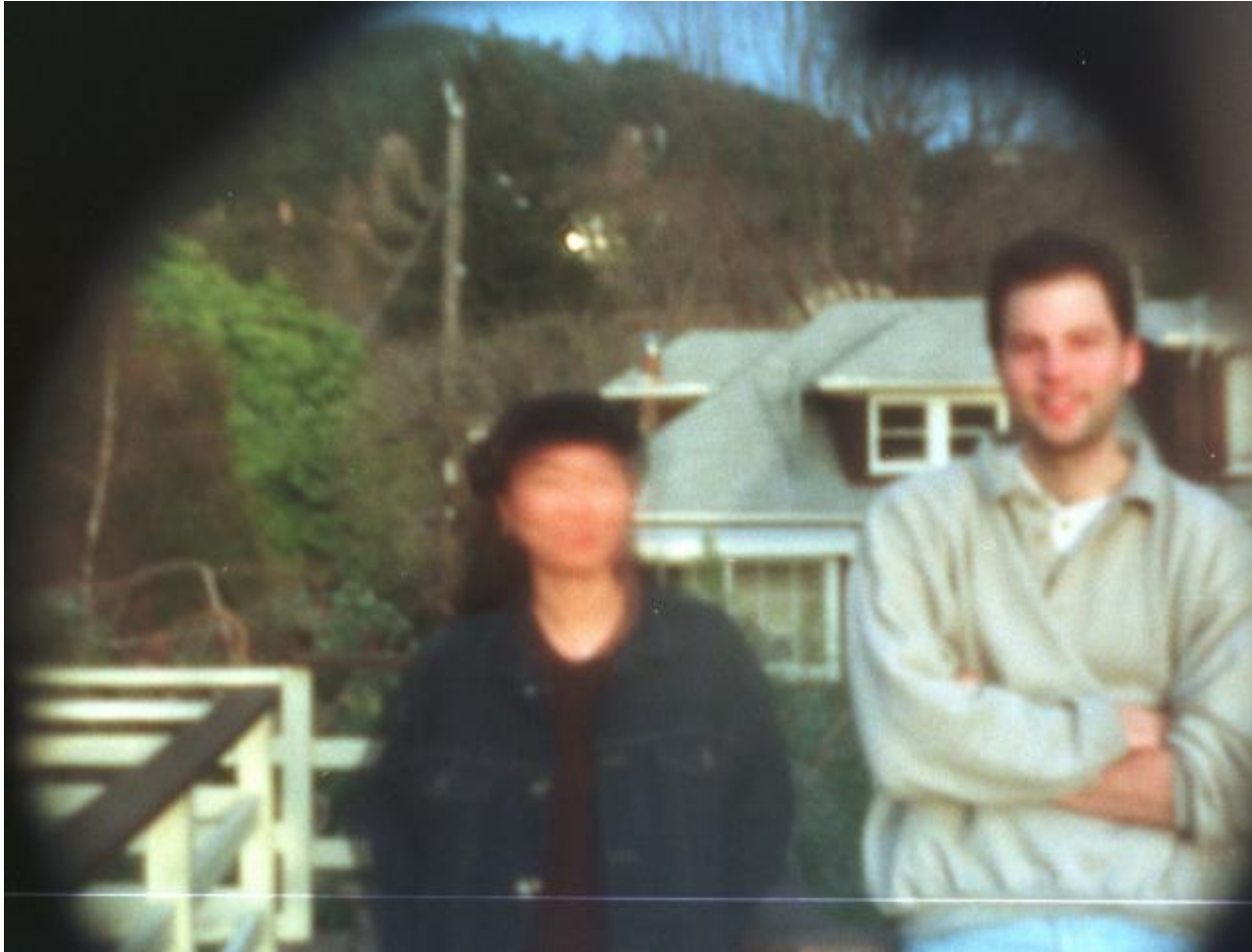
- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)
- Depth of the room (box) is the effective focal length

Pinhole camera model



- Pinhole model:
 - Captures **pencil of rays** – all rays through a single point
 - The point is called **Center of Projection**
 - The image is formed on the **Image Plane**

Home-made pinhole camera

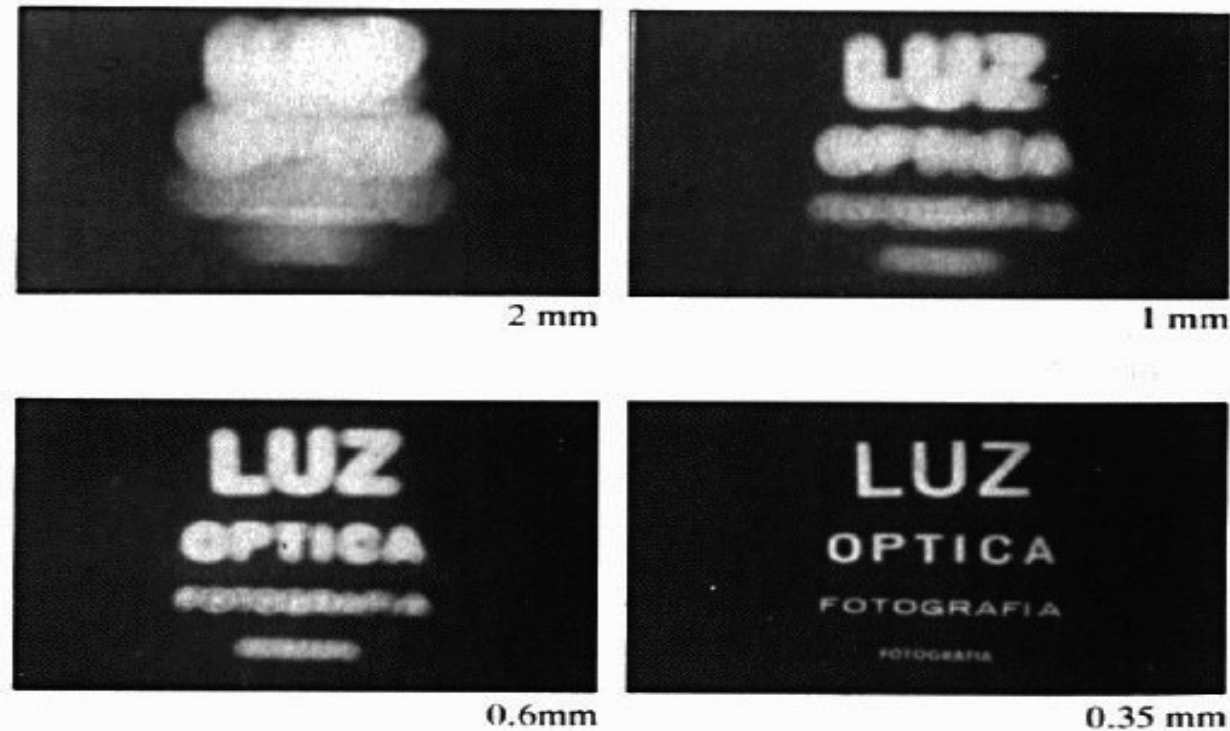


Why so blurry?



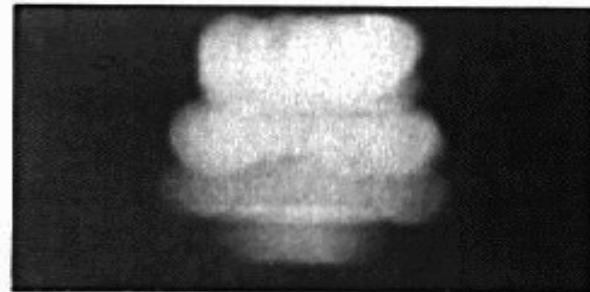
<http://www.debevec.org/Pinhole/>

Shrinking the aperture

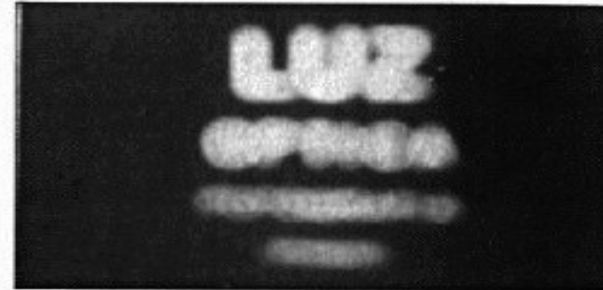


- Why not make the aperture as small as possible?
 - Less light gets through (must increase the exposure)
 - Diffraction effects...

Shrinking the aperture



2 mm



1 mm



0.6mm



0.35 mm

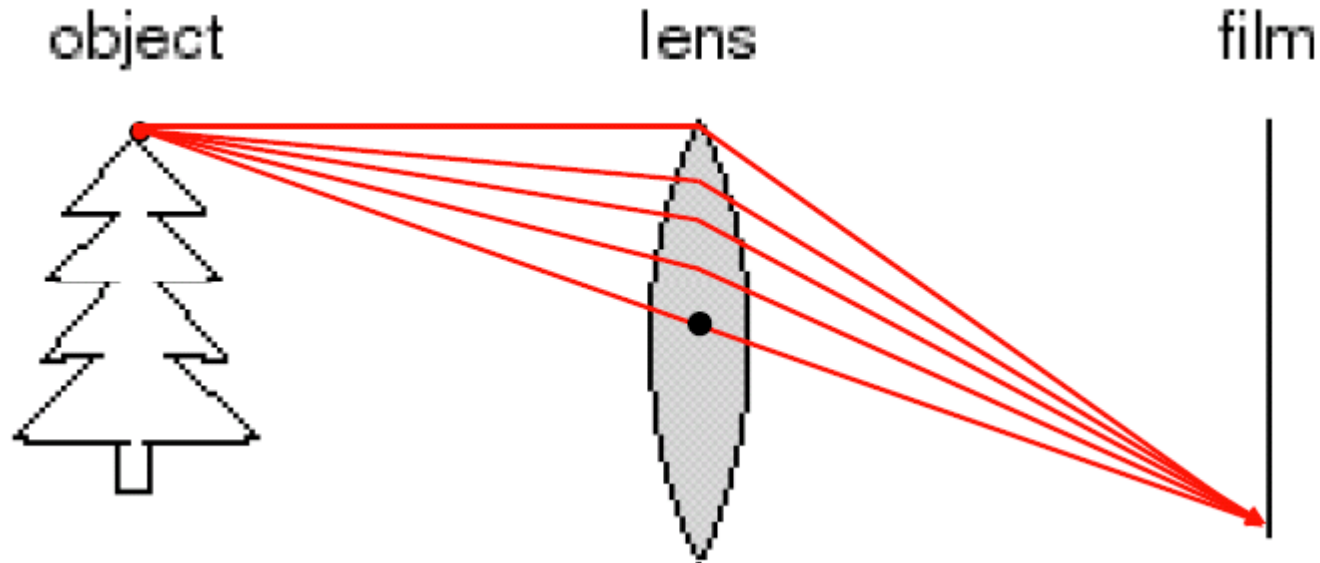


0.15 mm



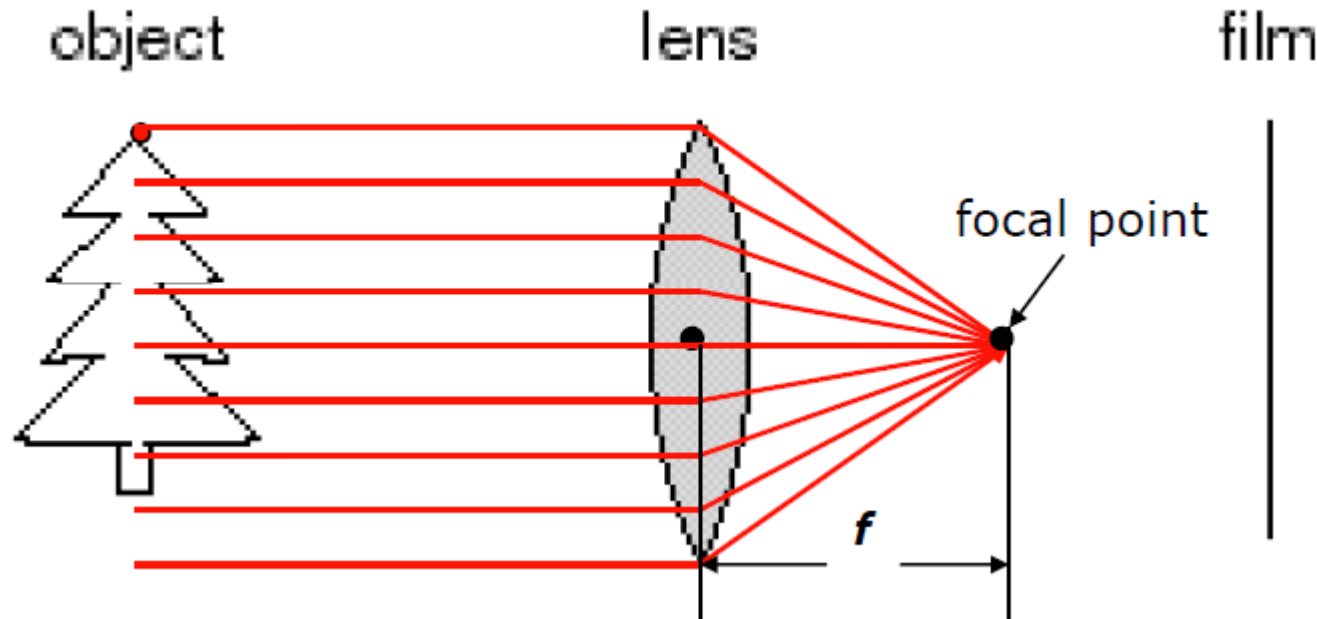
0.07 mm

Solution: adding a lens



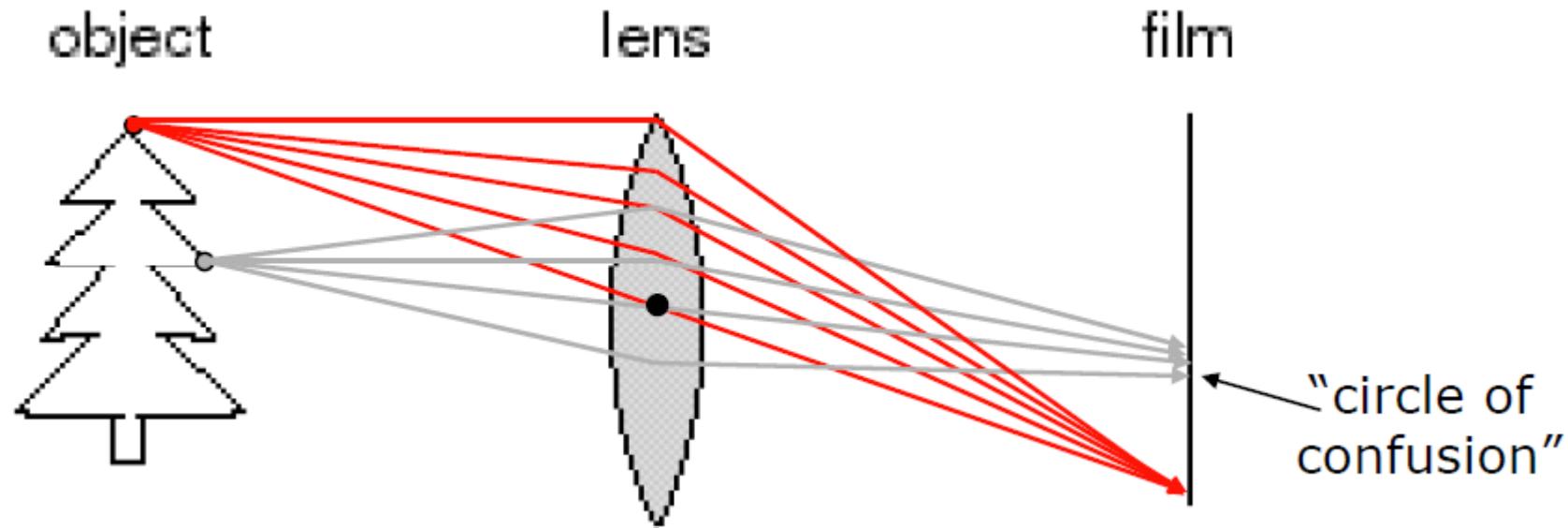
- A lens focuses light onto the film
 - Rays passing through the center are not deviated

Solution: adding a lens



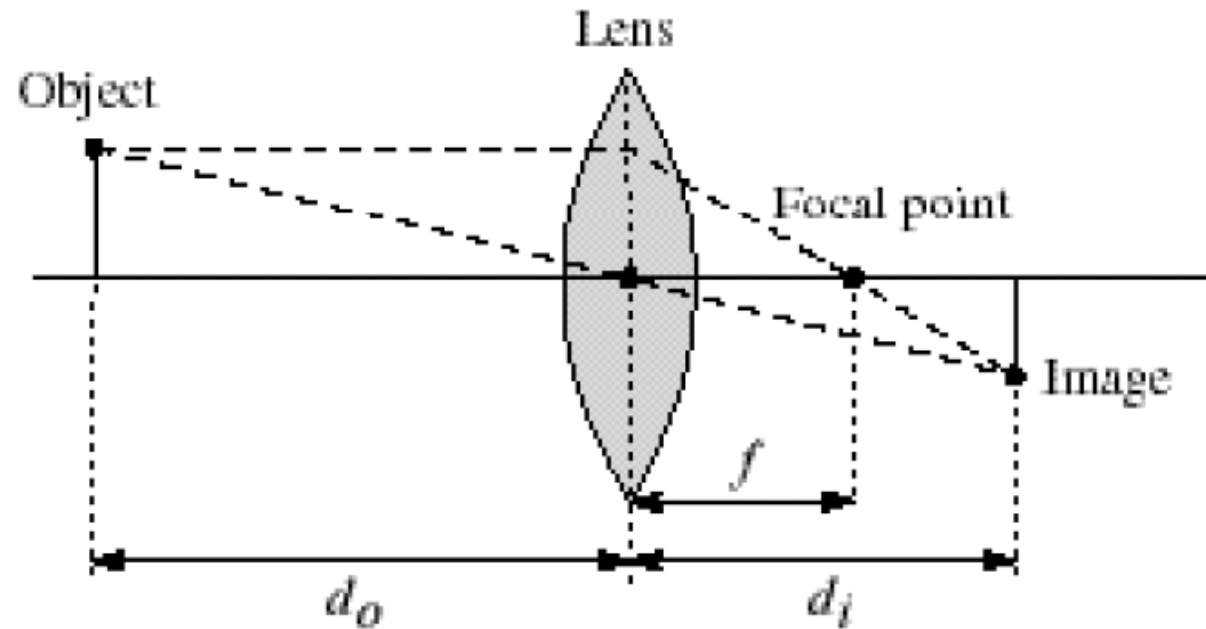
- A lens focuses light onto the film
 - Rays passing through the center are not deviated
 - All parallel rays converge to one point on a plane located at the *focal length* f

Solution: adding a lens



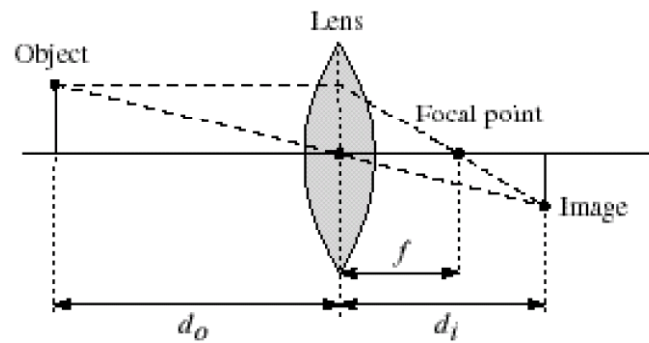
- A lens focuses light onto the film
 - There is a specific distance at which objects are “in focus”
 - other points project to a “circle of confusion” in the image

Thin lenses



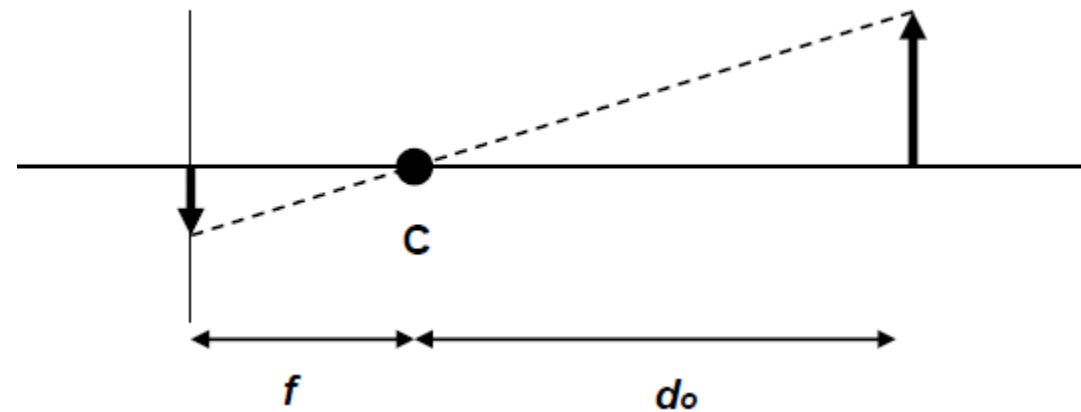
- Thin lens equation: $\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$
 - Any object point satisfying this equation is in focus
 - This formula can also be used to estimate roughly the distance to the object ("Depth from Focus")

Pin-hole approximation



$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

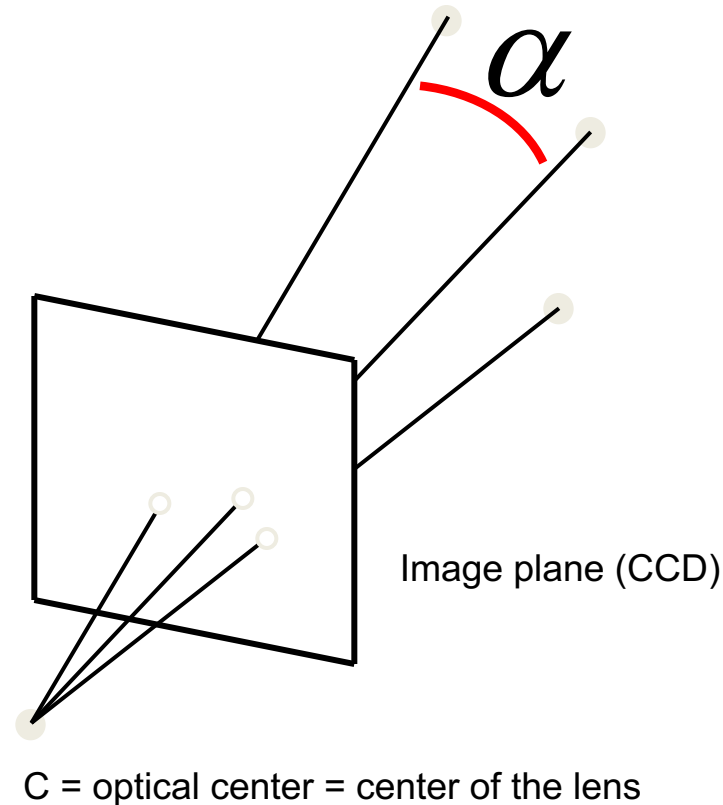
$$d_o \gg d_i \Rightarrow \frac{1}{d_i} \approx \frac{1}{f} \Rightarrow d_i \approx f$$



C = "optical center"

Pin-hole Model

Perspective camera



- For convenience, the image plane is usually represented in front so that the image preserves the same orientation (i.e. not flipped)
- **Notice: a camera does not measure distances but angles! Therefore it is a “bearing sensor”**

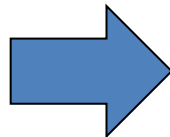
Perspective Projection onto the image plane

- To project a 3D scene point $P = (x, y, z)$ [meters] onto the camera image plane $p = (u, v)$ [pixels] we need to consider:
 - Pixelization: size of the pixel and position of the CCD with respect to the optical center
 - Rigid body transformation between camera and scene
- $u = v = 0$: where z-Axis passes through center of lens – z-Axis perpendicular to lens (coincident with optical axis)

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

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

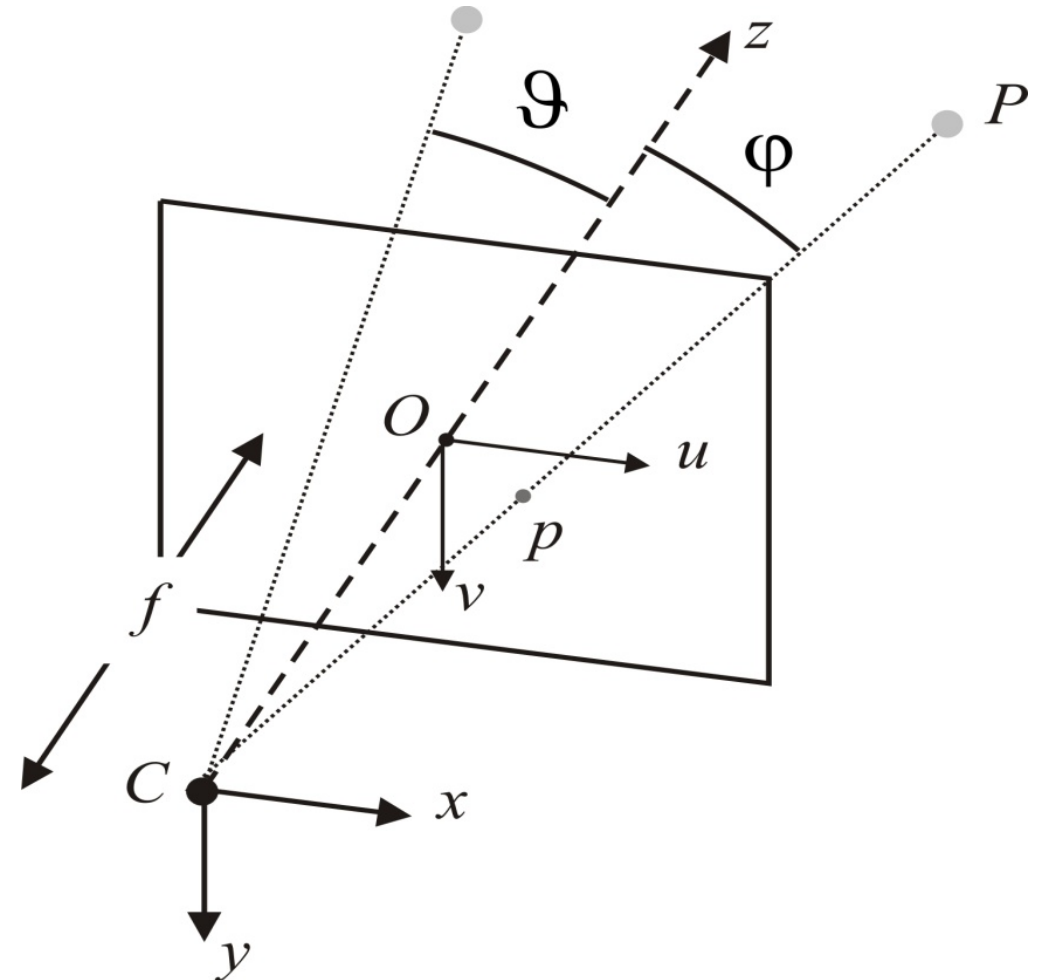
Simple case
(without pixelization)



$$u = k_u \frac{f}{z} \cdot x + u_0$$

$$v = k_v \frac{f}{z} \cdot y + v_0$$

With pixelization
 u_0, v_0 are the coordinates
of the optical center
 k_u and k_v are in [pxl/m]



Projection onto the image plane

- Observe that we can also rewrite this

$$u = k_u \frac{f}{z} \cdot x + u_0$$

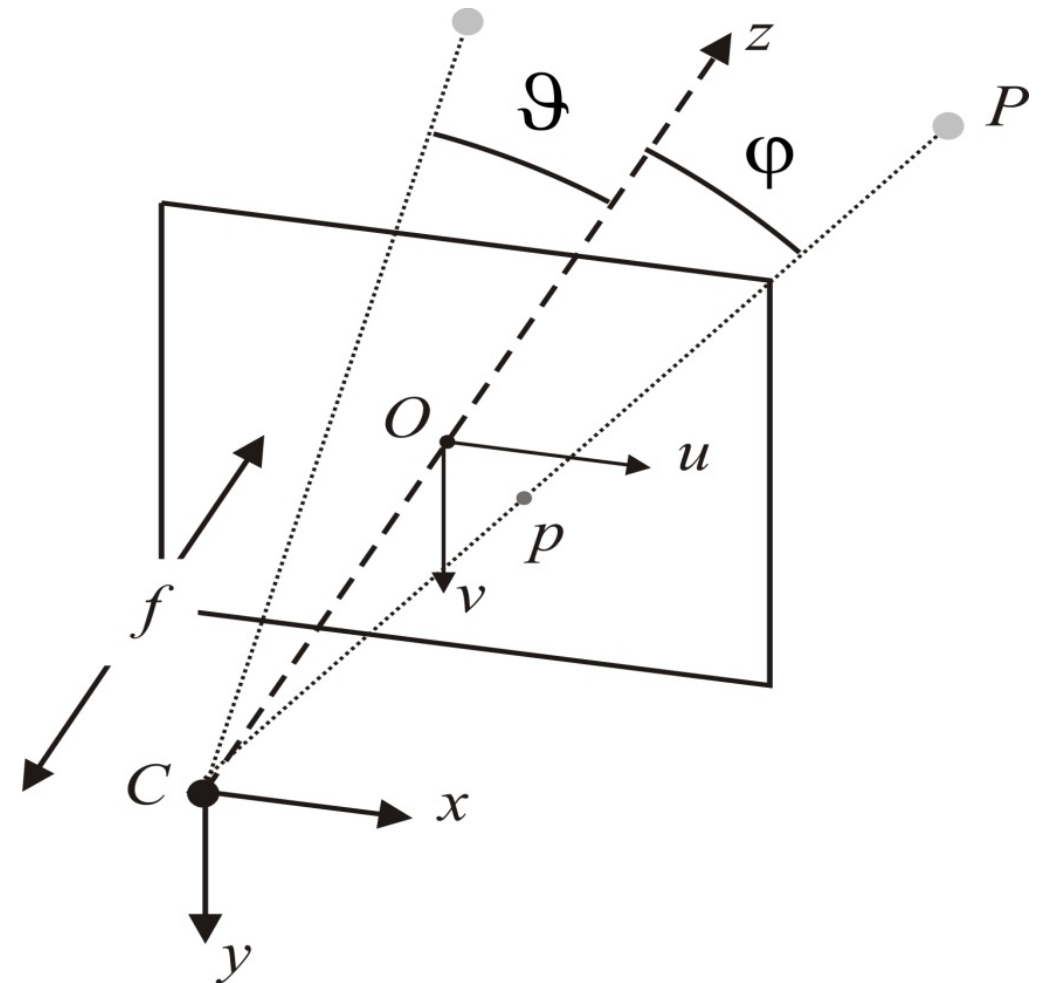
$$v = k_v \frac{f}{z} \cdot y + v_0$$

in matrix form (λ - homogeneous coordinates)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Or alternatively

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



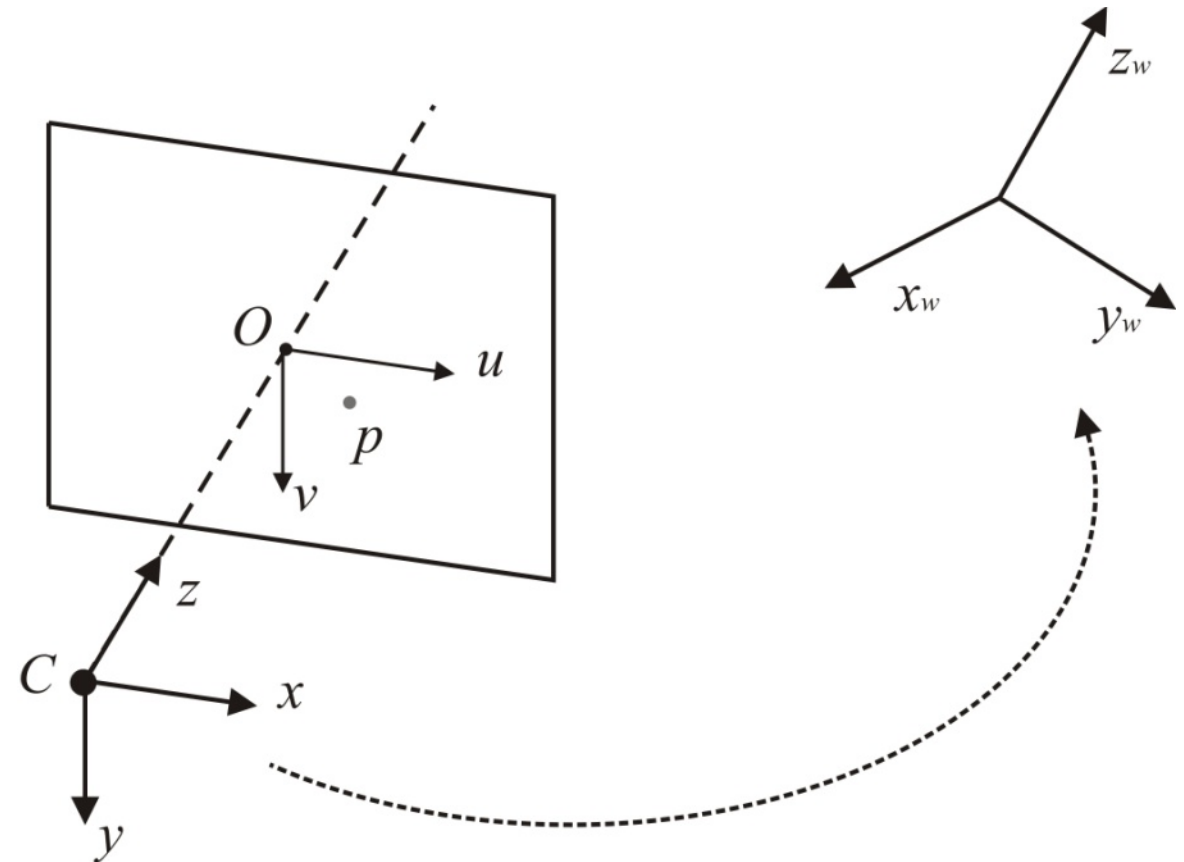
Projection onto the image plane

- Rigid body transformation from the World to the Camera reference frame

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = A \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$

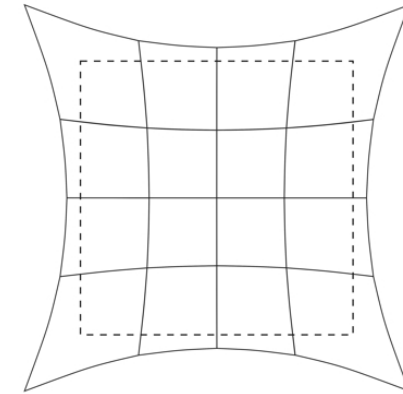
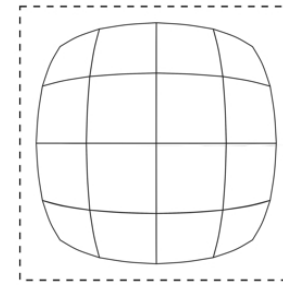
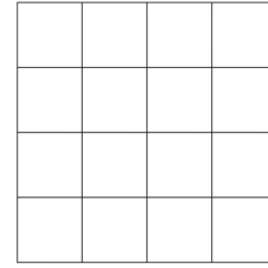


Radial distortion

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

where

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



Barrel distortion



Pincushion distortion

Camera Calibration

- How many parameters do we need to model a camera?

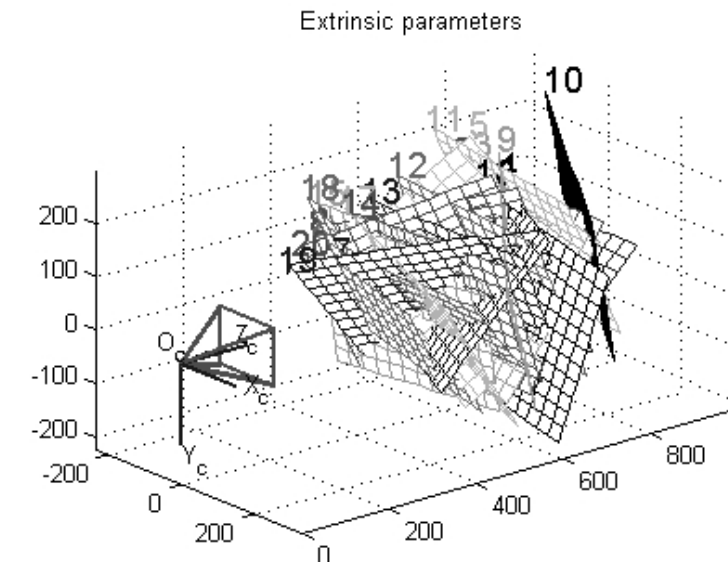
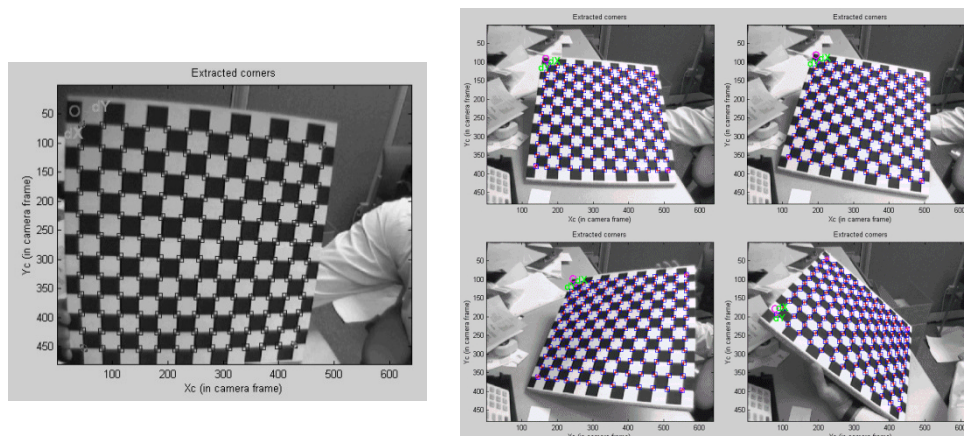
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad \begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 \rho^2) \cdot \begin{bmatrix} u \\ v \end{bmatrix}$$

- 5 “intrinsic” parameters: $\alpha_u, \alpha_v, u_0, v_0, k_1$
- Camera pose?
- 6 “extrinsic” parameters (or 0 if the world and the camera frames coincide)

Camera Calibration: how does it work?

- Calibration: measuring accurately **intrinsic** + **extrinsic** parameters of the camera model.
- Parameters: govern mapping from scene points to image points
- Idea: known:
 - pixel coordinates of image points p
 - 3D coordinates of the corresponding scene points P
 - => compute the unknown parameters A , R , T by solving the perspective projection equation

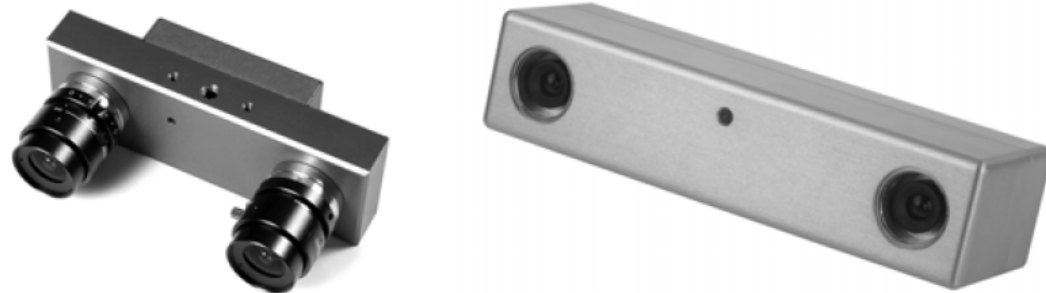
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T$$



STEREO VISION

How do we measure distances with cameras?

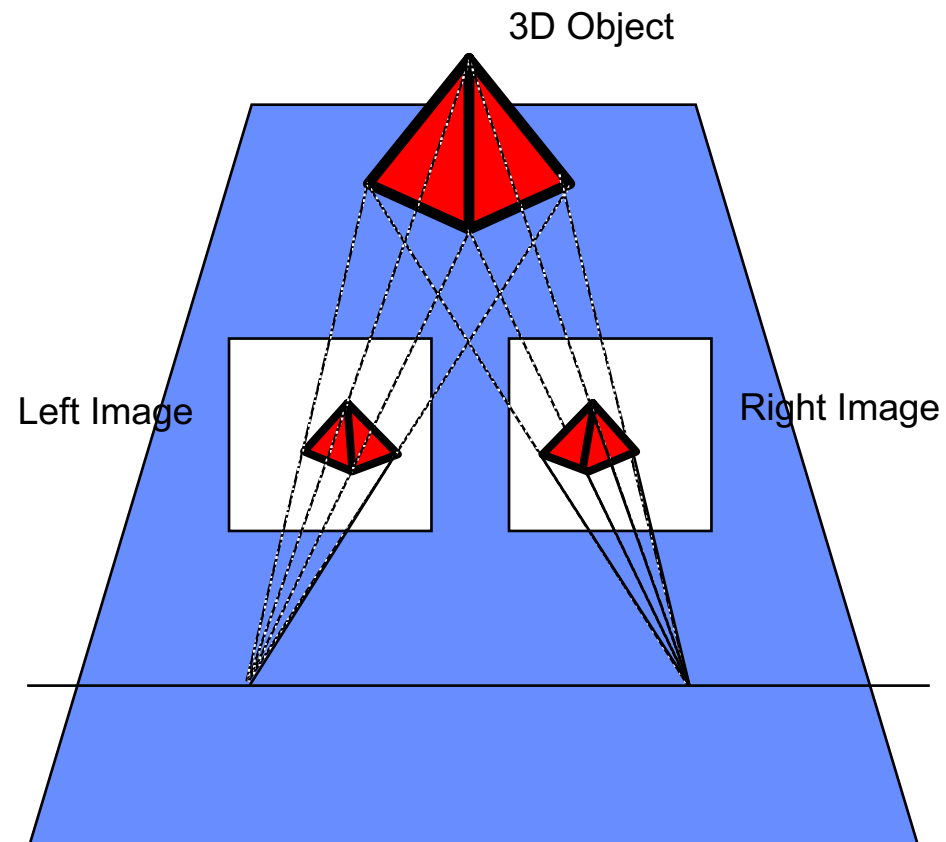
- Structure from stereo (Stereo-vision):
 - use two cameras with known relative position and orientation



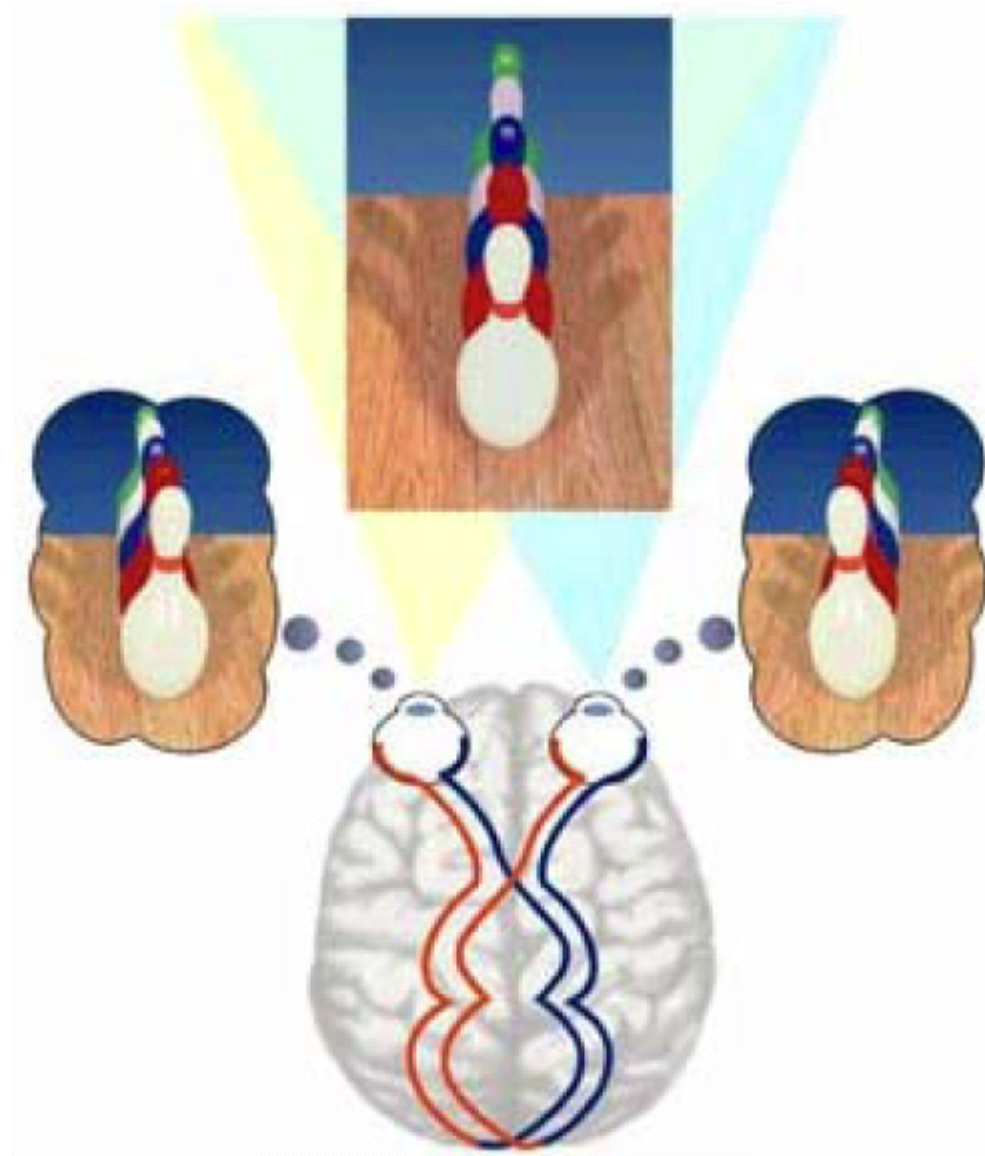
- Structure from motion:
 - use a single moving camera: both 3D structure and camera motion can be estimated up to a scale

Stereo Vision

- Allows to reconstruct a 3D object from two images taken at different locations

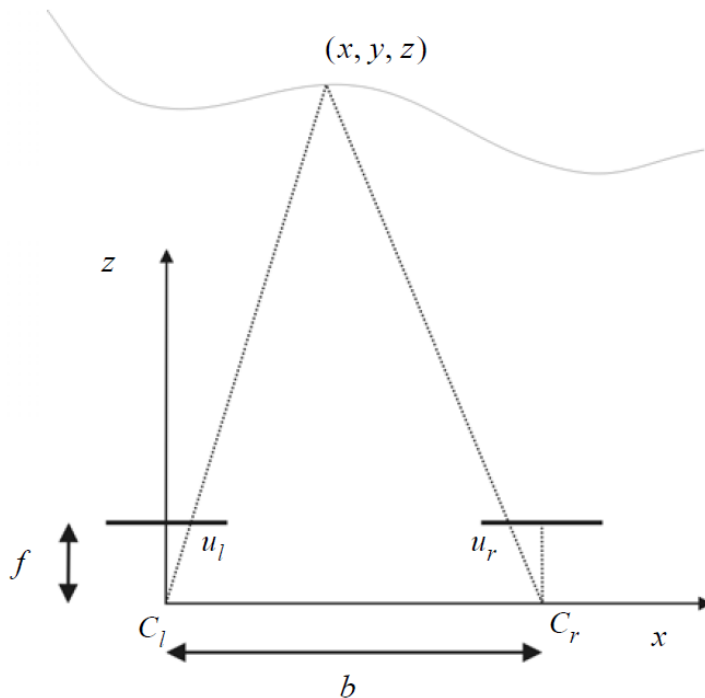


Disparity in the human retina



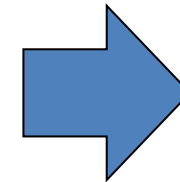
Stereo Vision - The simplified case

- The simplified case is an ideal case. It assumes that both cameras are identical and are aligned on a horizontal axis



$$\frac{f}{z} = \frac{u_l}{x},$$

$$\frac{f}{z} = \frac{u_r}{b - x}$$



$$z = b \frac{f}{u_l - u_r}$$

Distance

- b = baseline, distance between the optical centers of the two cameras
- f = focal length
- $u_l - u_r$ = disparity

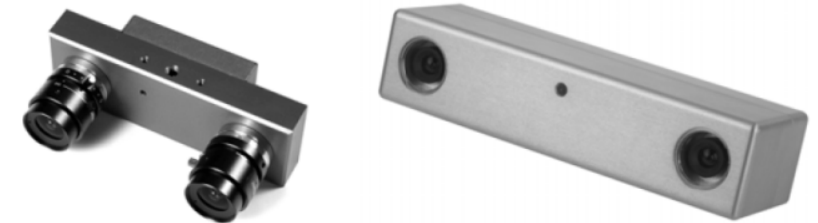
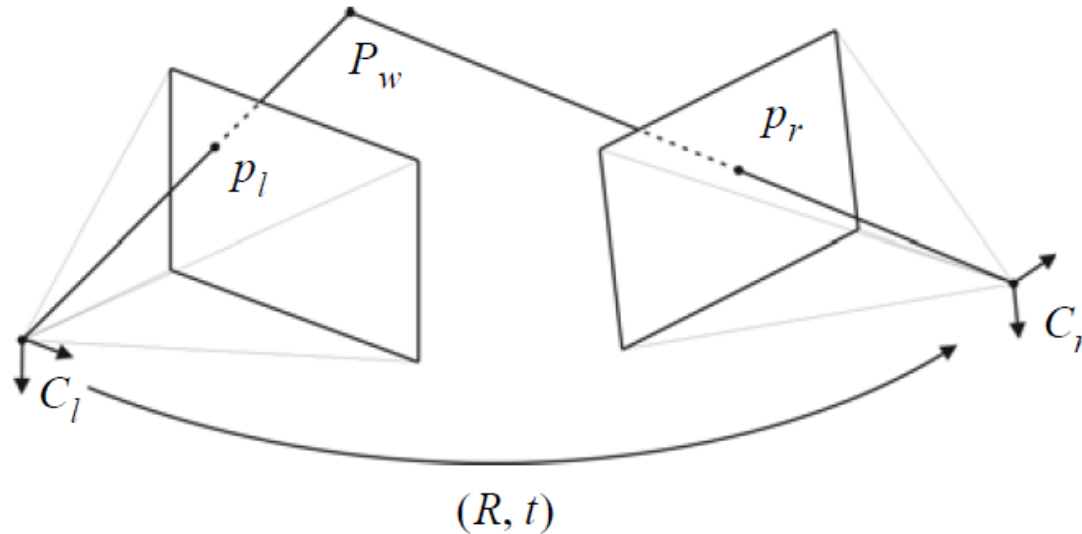
Stereo Vision: how to improve accuracy?

$$z = b \frac{f}{u_l - u_r}$$

1. Distance is inversely proportional to disparity ($u_l - u_r$)
 - closer objects can be measured more accurately
2. Disparity is proportional to b
 - For a given disparity error, the accuracy of the depth estimate increases with increasing baseline b .
 - However, as b is increased, some objects may appear in one camera, but not in the other.
3. Increasing image resolution improves accuracy

Stereo Vision – the general case

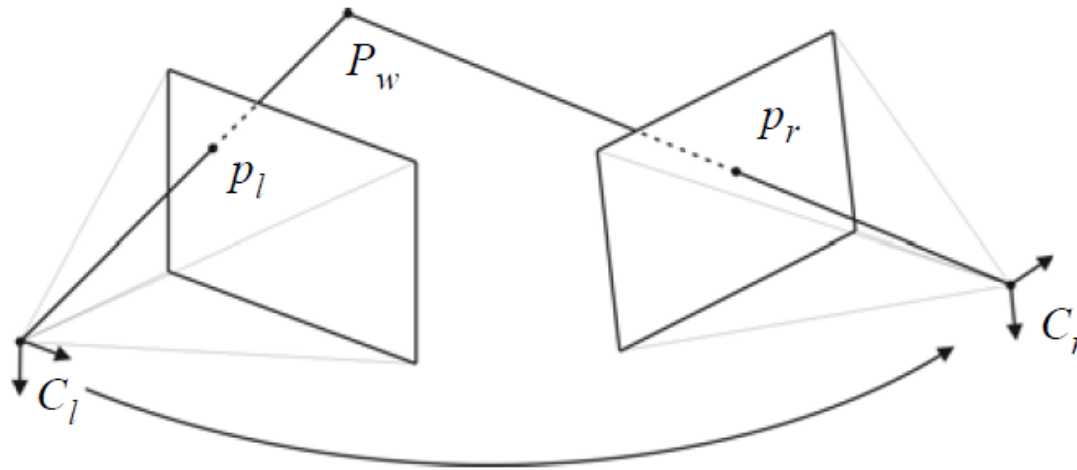
- Two identical cameras do not exist in nature!
- Aligning both cameras on a horizontal axis is very hard, also with the most expensive stereo cameras!



- In order to be able to use a stereo camera, we need first to estimate the relative pose between the cameras, that is, **Rotation** and **Translation**
- However, as the two cameras are not identical, we need to estimate: **focal length, image center, radial distortion**

Stereo Vision – the general case

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



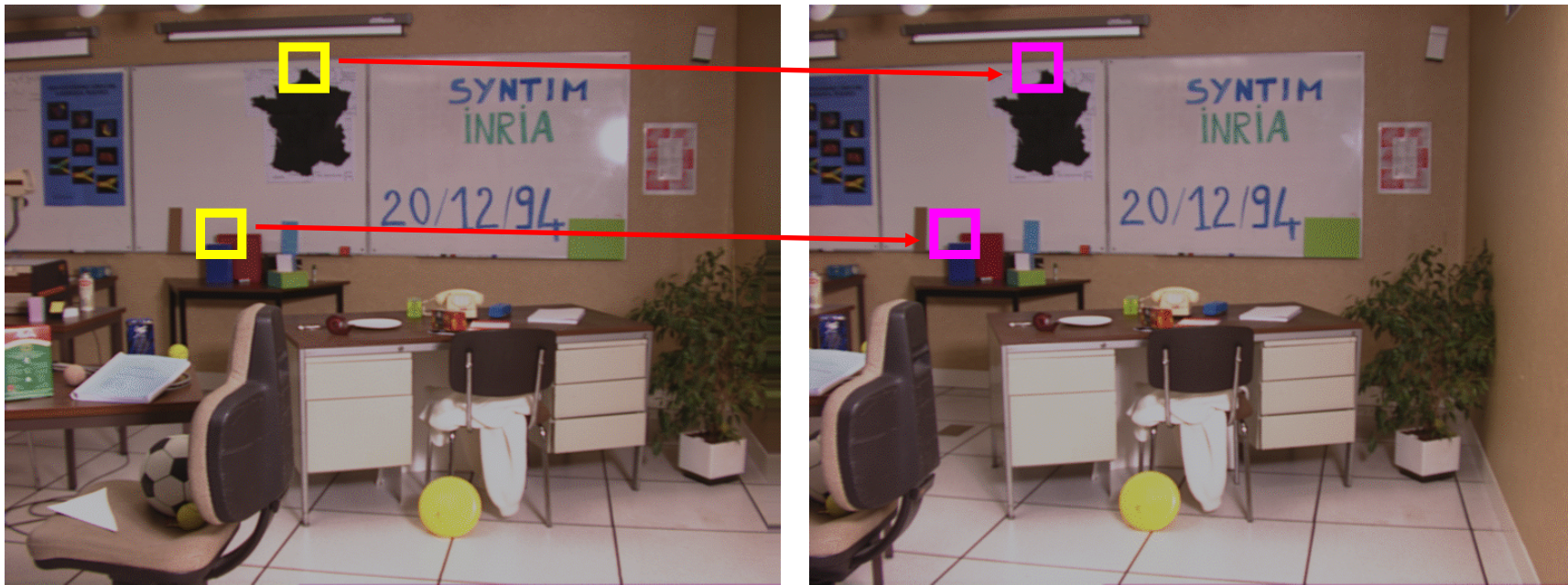
$$\lambda_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = A_l \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (R, t)$$

Left camera; world frame coincides with the left camera frame

$$\lambda_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = A_r \cdot R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad \text{Right camera}$$

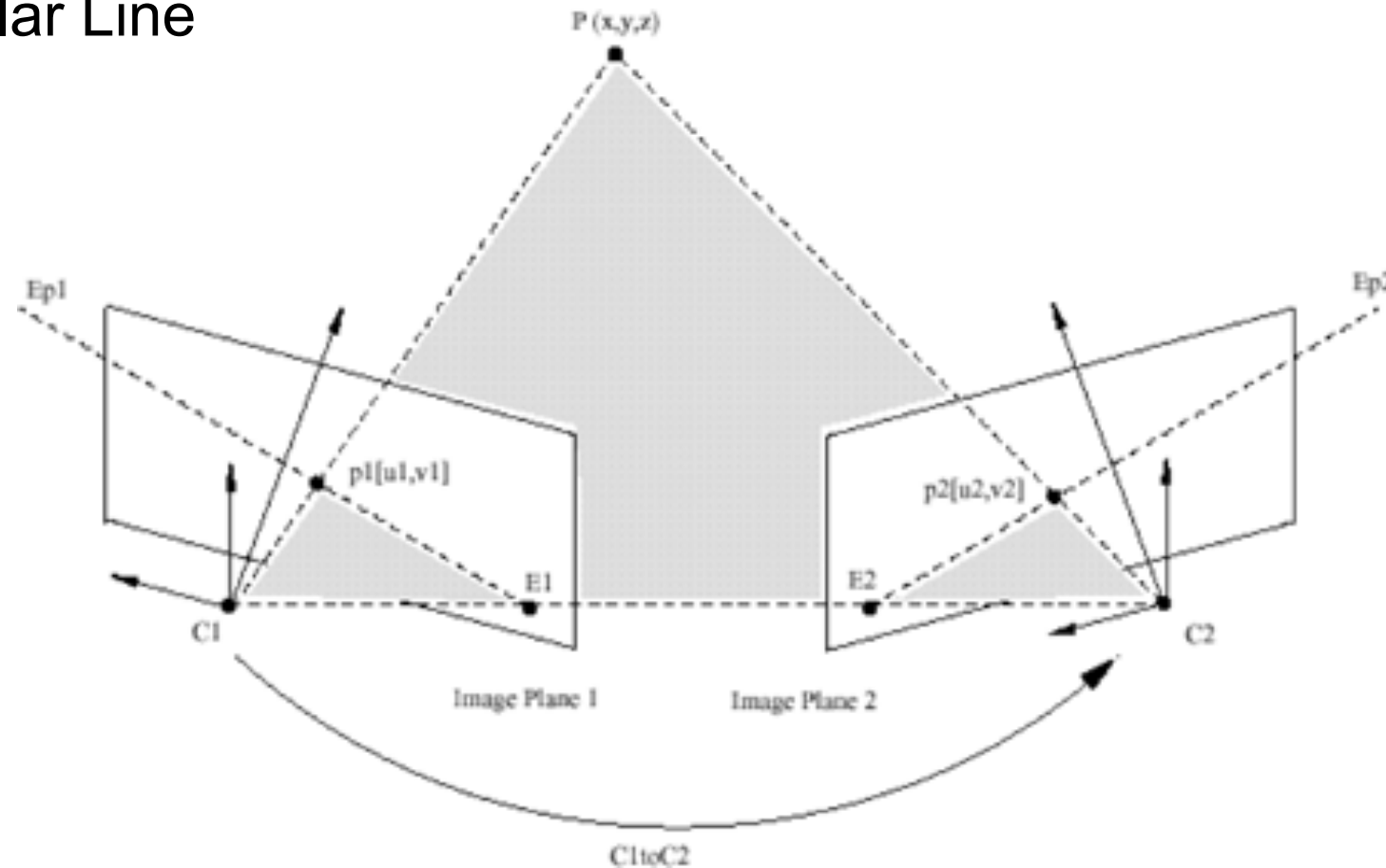
Stereo Vision: Correspondence Problem

- Matching between points in the two images which are projection of the same 3D real point
- Correspondence search could be done by comparing the observed points with all other points in the other image. Typical similarity measures are the Correlation and image Difference.
- This image search can be computationally very expensive! Is there a way to make the correspondence search 1 dimensional?



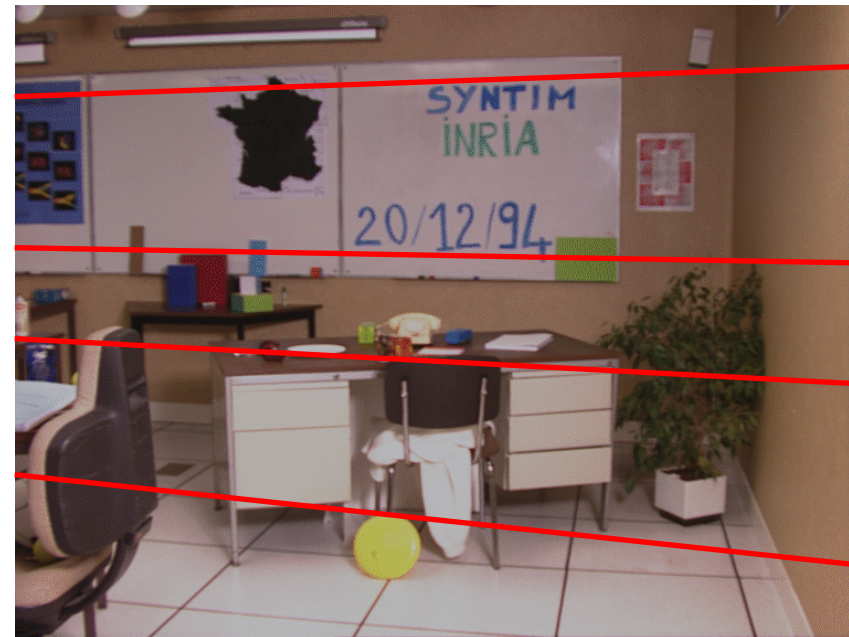
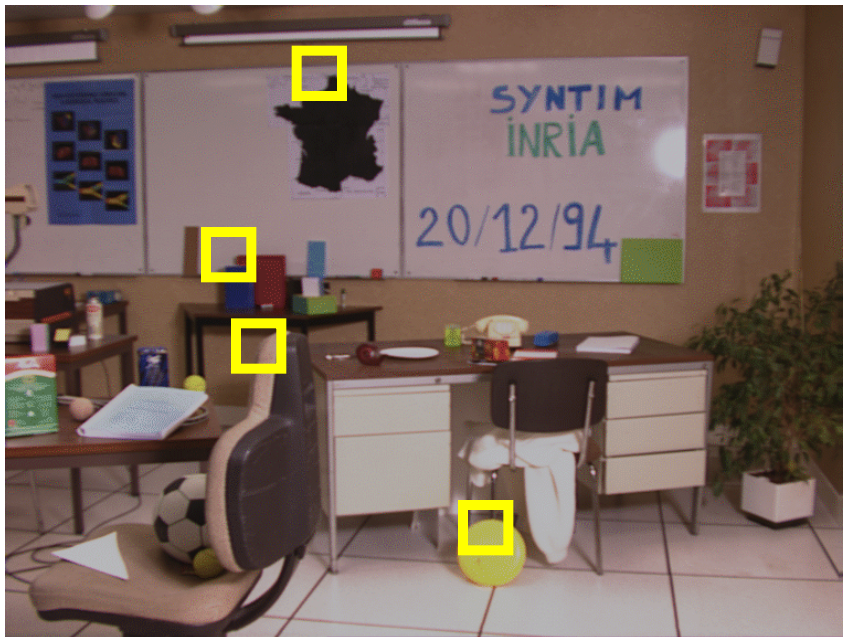
Correspondence Problem: Epipolar Constraint

- The correspondent of a point in an image must lie on a line in the other image, called Epipolar Line



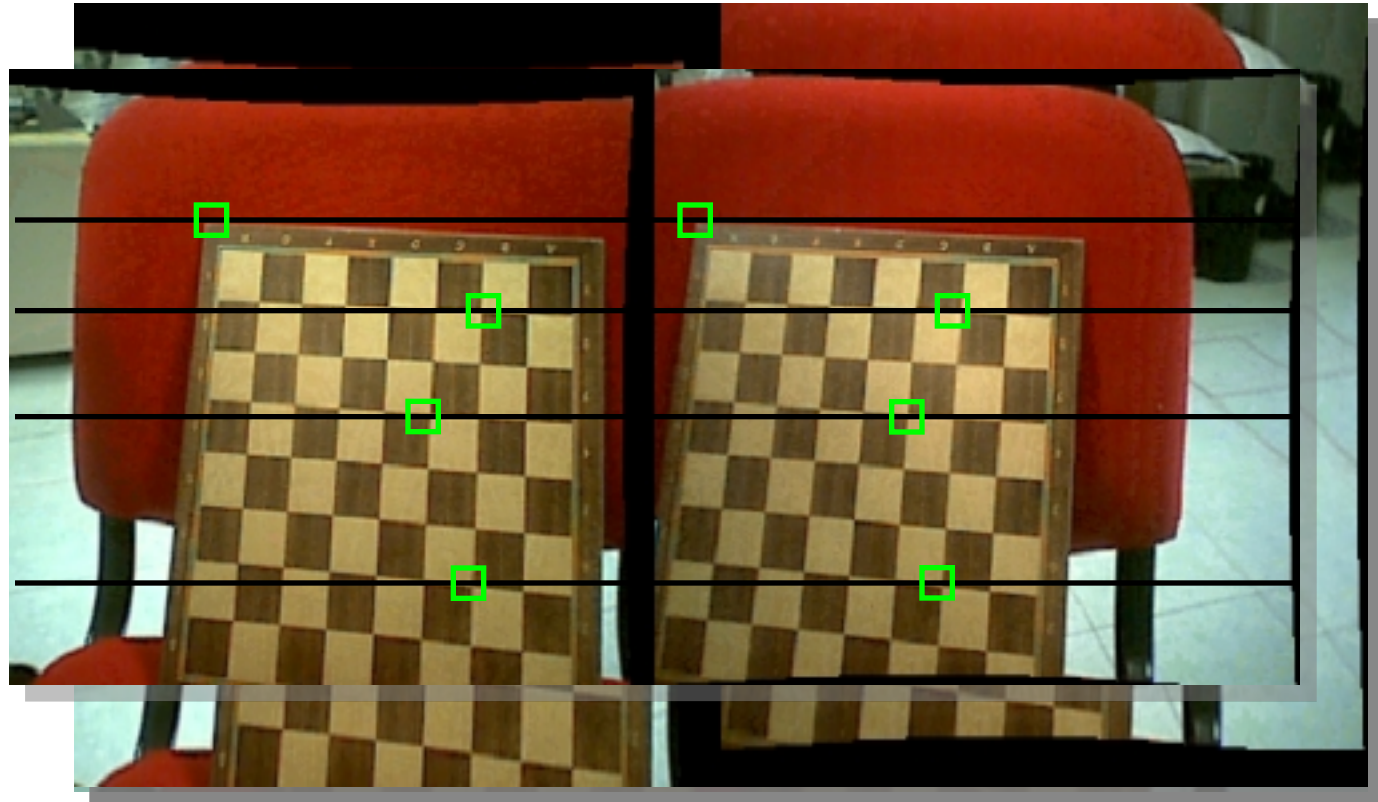
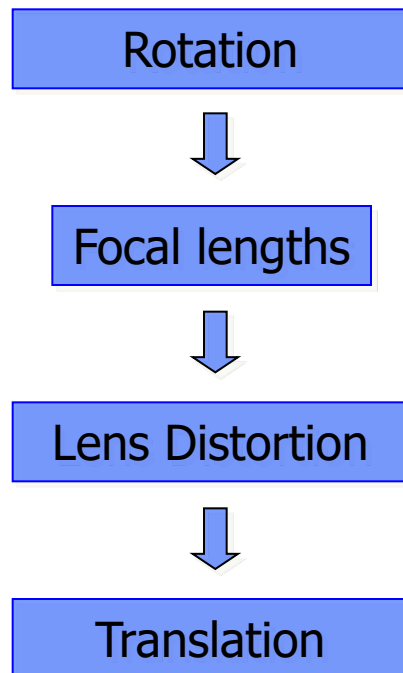
Correspondence Problem: Epipolar Constraint

- Thanks to the epipolar constraint, conjugate points can be searched along epipolar lines: this reduces the computational cost to 1 dimension!



Epipolar Rectification

- Determines a transformation of each image plane so that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes (usually the horizontal one)



Stereo Vision Output 1 – Disparity map

- Find the correspondent points of all image pixels of the original images
- For each pair of conjugate points compute the disparity $d = v - v'$
- $d(x,y)$ is called Disparity map.

- Disparity maps are usually visualized as grey-scale images. Objects that are closer to the camera appear lighter, those who are further appear darker.



Left image

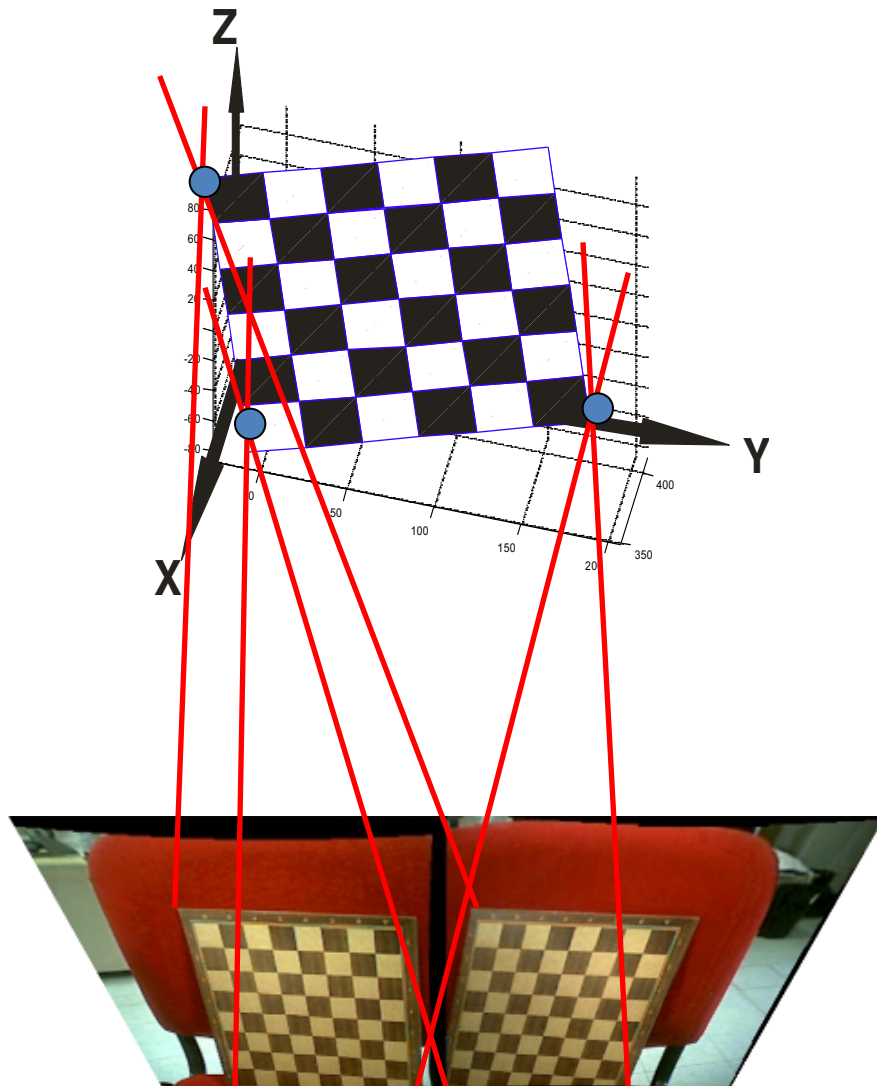


Right image



Disparity map

Stereo Vision Output 2 - 3D Reconstruction via triangulation



Stereo Camera Calibration

Estimates the parameters that manage the 3D – 2D transformation

$$\begin{cases} \begin{bmatrix} u_L \\ v_L \end{bmatrix} = f \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ \begin{bmatrix} u_R \\ v_R \end{bmatrix} = f \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \end{cases}$$

Stereo Camera Calibration

Totally:

Estimates the parameters that manage the 3D – 2D transformation
10 parameters for each camera need to be estimated

$$\left\{ \begin{array}{l} \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} = \begin{bmatrix} f_{u,L} & 0 & u_{0,L} \\ 0 & f_{v,L} & v_{0,L} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\ \begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix} = \begin{bmatrix} f_{u,R} & 0 & u_{0,R} \\ 0 & f_{v,R} & v_{0,R} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \end{array} \right.$$

Stereo Camera Calibration

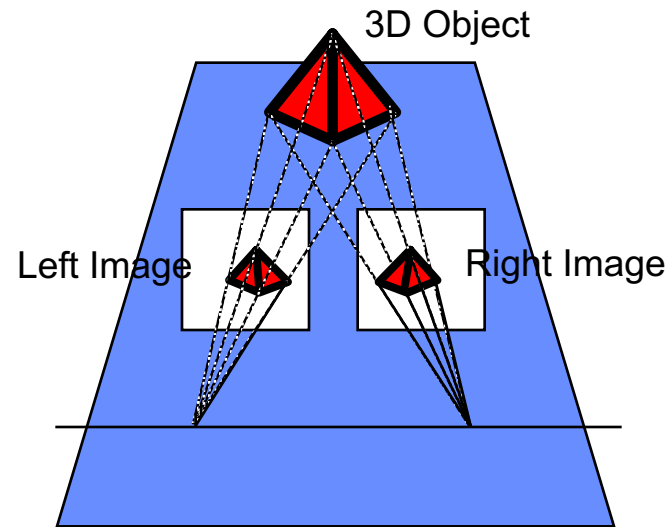
PLUS

5 parameters for each camera in order to compensate for lens distortion (radial & tangential distortion)

$$\begin{cases} \begin{bmatrix} u_{d,L} \\ v_{d,L} \end{bmatrix} = (1 + kc_{1,L}\rho^2 + kc_{2,L}\rho^4 + kc_{5,L}\rho^6) \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 2kc_{3,L} \cdot u \cdot v + kc_{4,L}(\rho^2 + 2u^2) \\ kc_{3,L}(\rho^2 + 2v^2) + 2kc_{4,L} \cdot u \cdot v \end{bmatrix} \\ \begin{bmatrix} u_{d,R} \\ v_{d,R} \end{bmatrix} = (1 + kc_{1,R}\rho^2 + kc_{2,R}\rho^4 + kc_{5,R}\rho^6) \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 2kc_{3,R} \cdot u \cdot v + kc_{4,R}(\rho^2 + 2u^2) \\ kc_{3,R}(\rho^2 + 2v^2) + 2kc_{4,R} \cdot u \cdot v \end{bmatrix} \end{cases}$$



Stereo Vision - summary

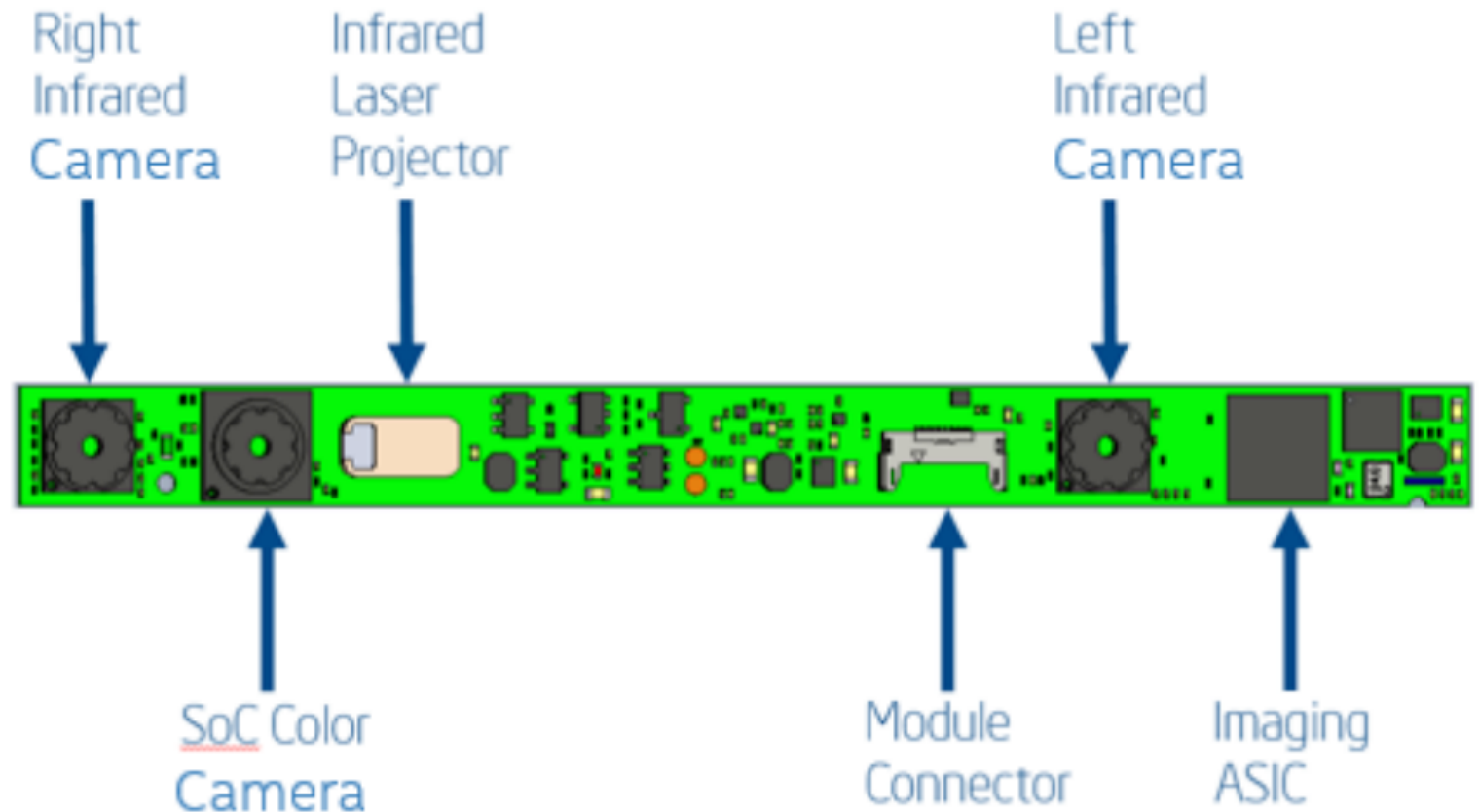


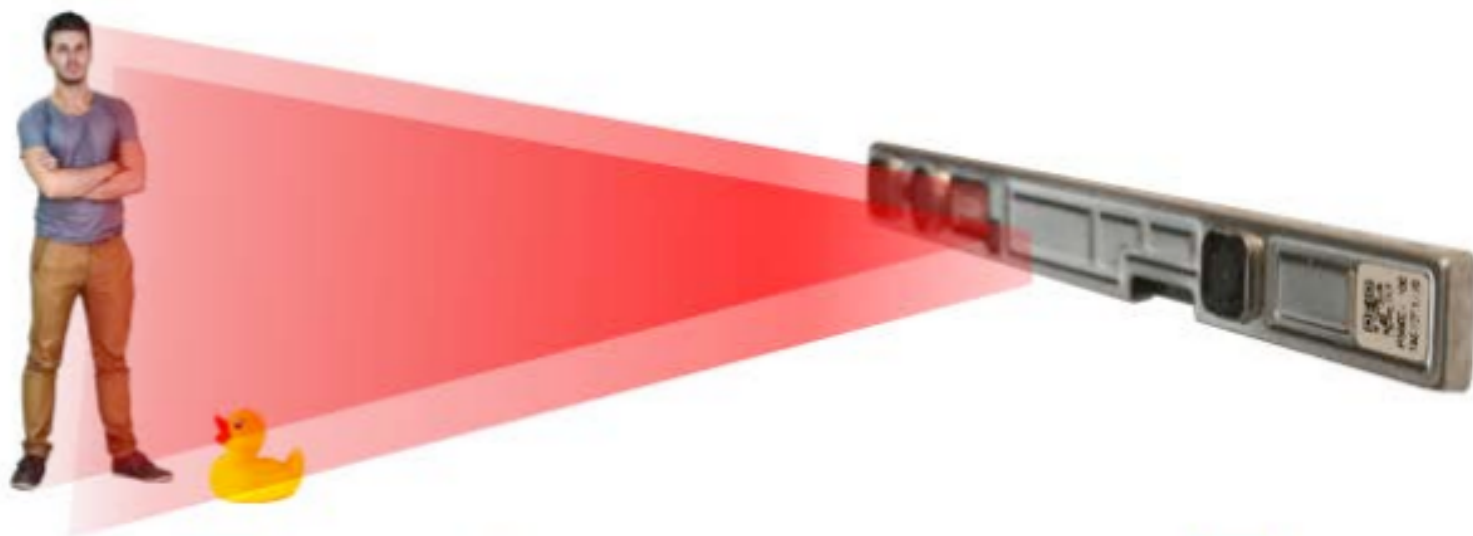
1. Stereo camera calibration -> compute camera relative pose
2. Epipolar rectification -> align images
3. Search correspondences
4. Output: compute stereo triangulation or disparity map
5. Consider baseline and image resolution to compute accuracy!

Device example:



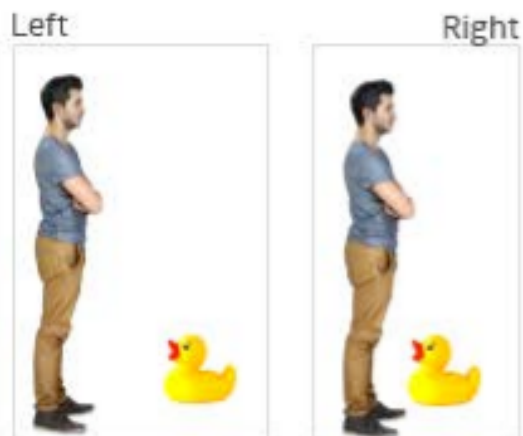
- 640x480 stereo infrared cameras
- 1920x1080 HD RGB camera
- Infrared Laser Projector
- Up to 60Hz





1 CAPTURE

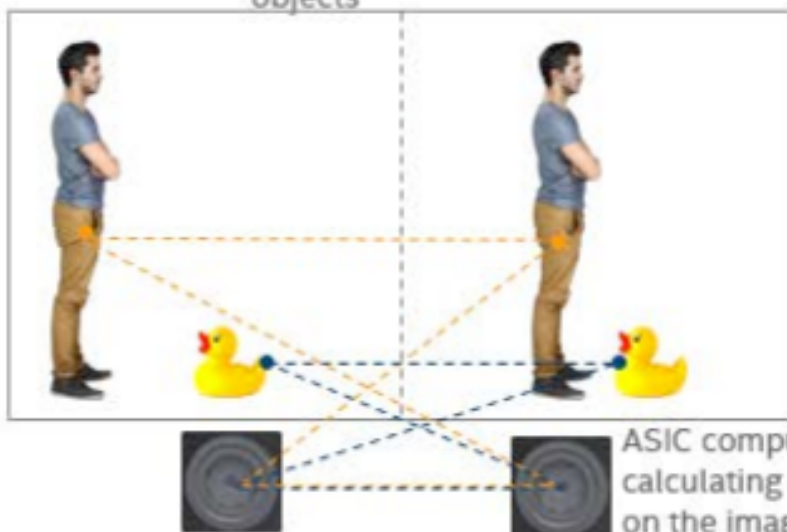
Project invisible light pattern on low texture surfaces like plain walls



Each camera sees a slightly different viewpoint

2 SEARCH

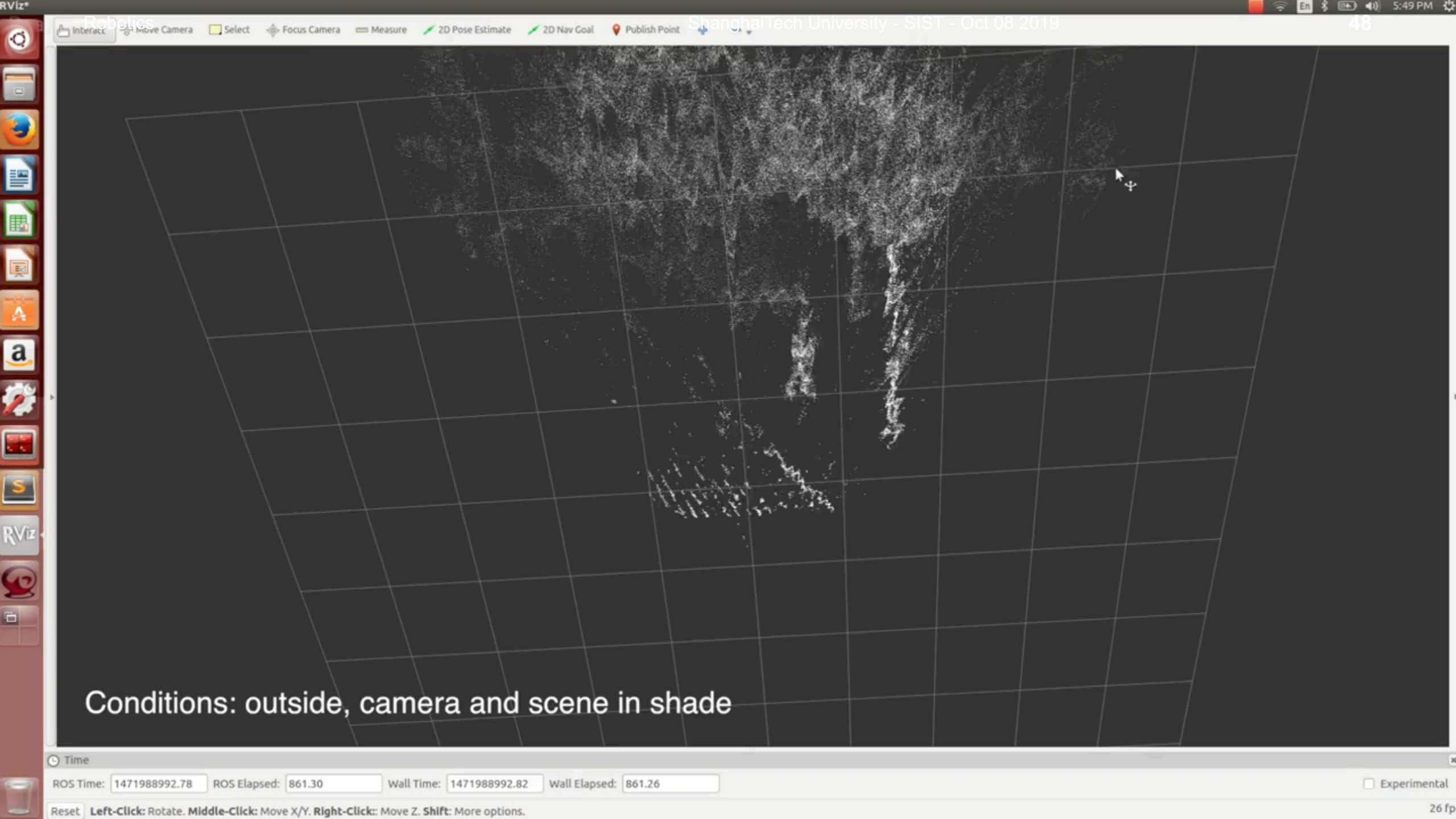
Closer objects shift more than further away objects



ASIC computes depth by calculating shift of every pixel on the image between the left and right images

3 DEPTH





Conditions: outside, camera and scene in shade

Time

ROS Time: 1471988992.78 ROS Elapsed: 861.30 Wall Time: 1471988992.82 Wall Elapsed: 861.26

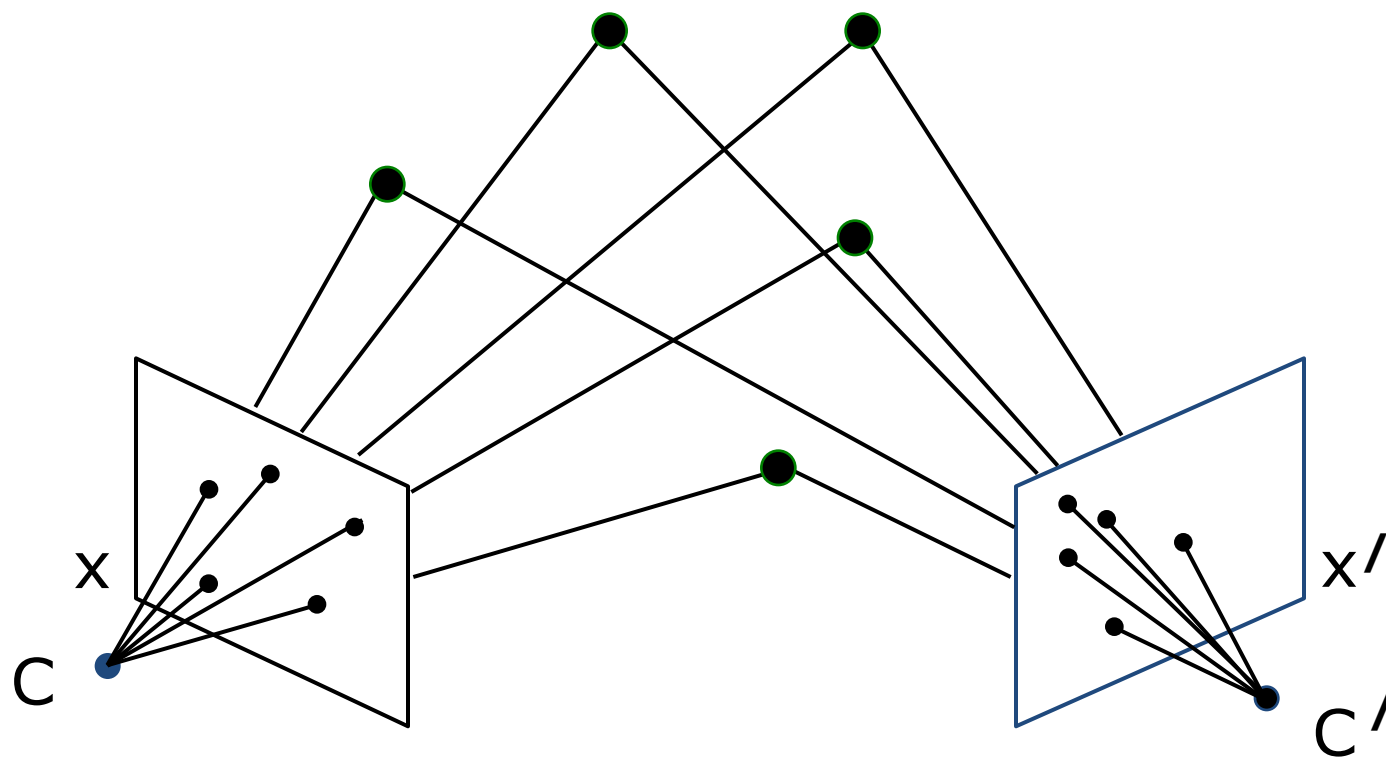
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Move Z. Shift: More options.

☐ Experimental

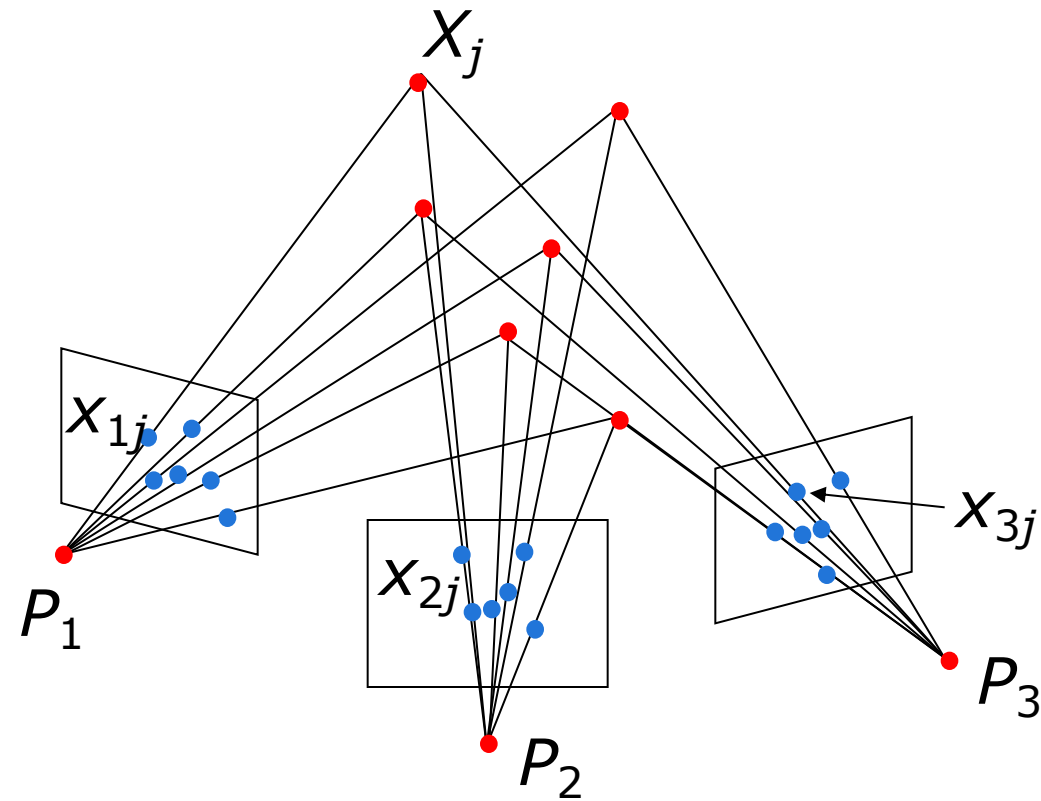
26 fp

Structure from motion

- Given image point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, determine R and T
- Rotate and translate camera until stars of rays intersect
- At least 5 point correspondences are needed



Multiple-view structure from motion



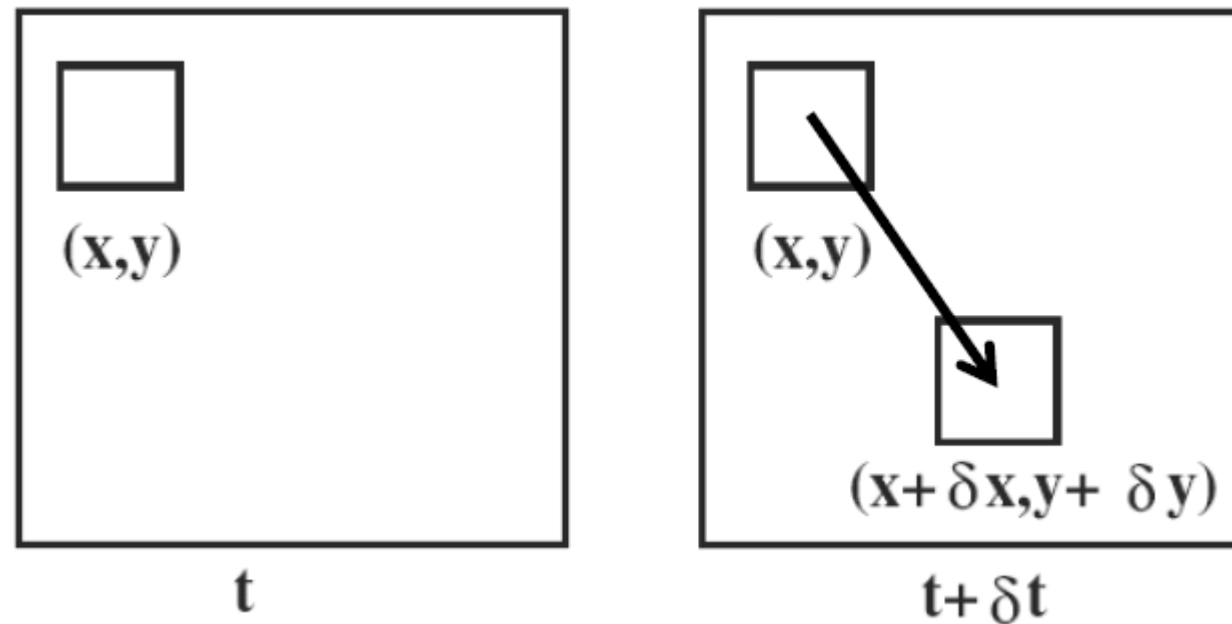
Multiple-view structure from motion

- Results of Structure from motion from 2 million user images from flickr.com - 2,106 images selected, 819,242 points



Optical Flow

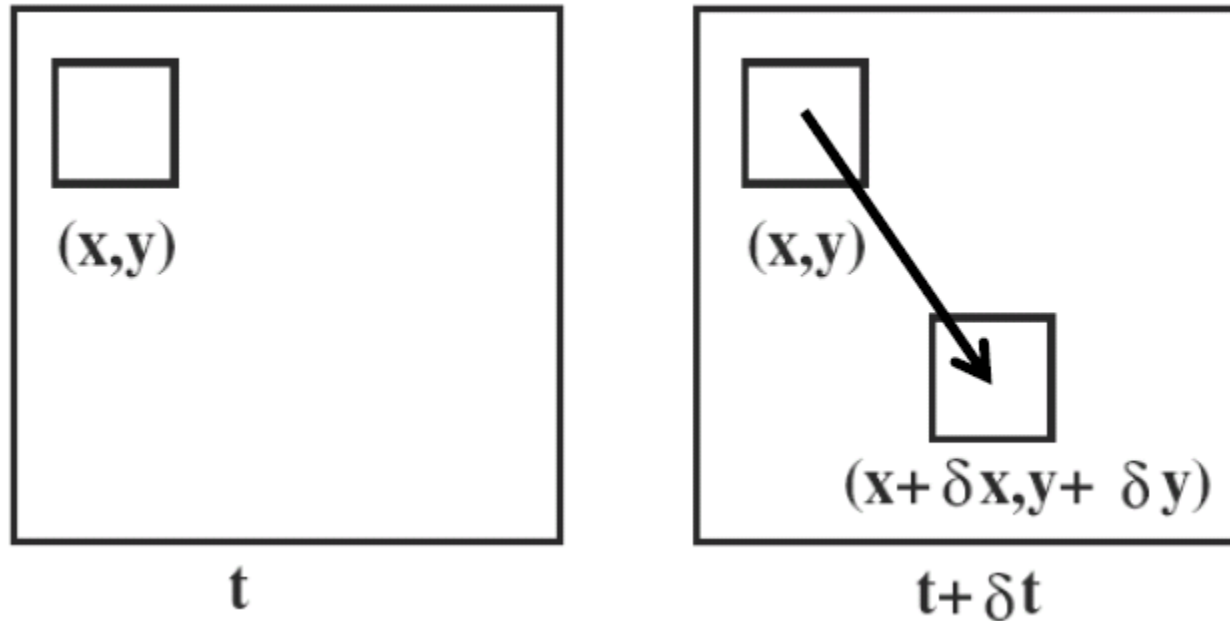
- It computes the motion vectors of all pixels in the image (or a subset of them to be faster)



$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

- Applications include collision avoidance

Optical Flow



$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0$$

$$\frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} = 0$$

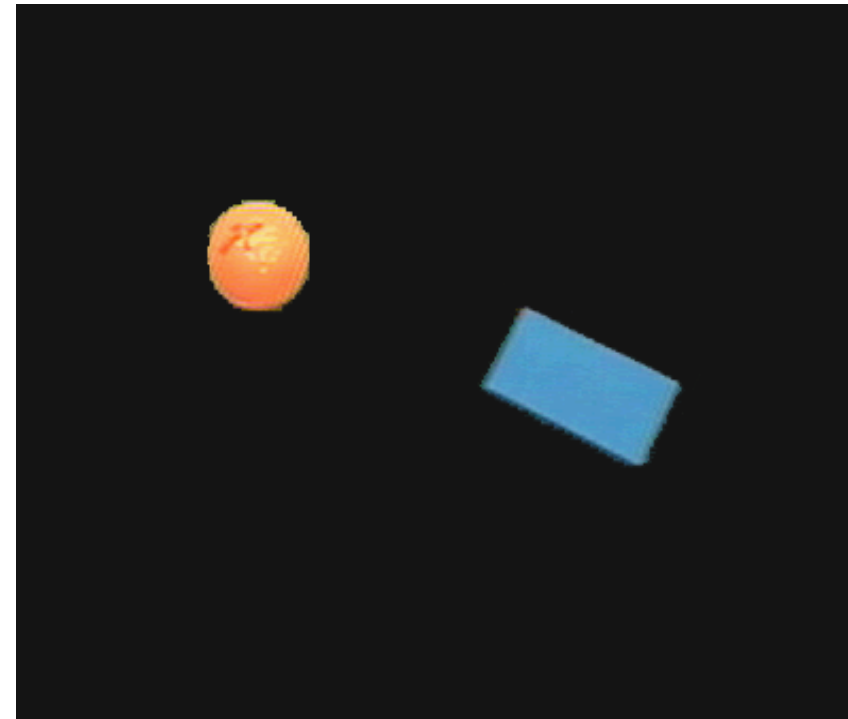
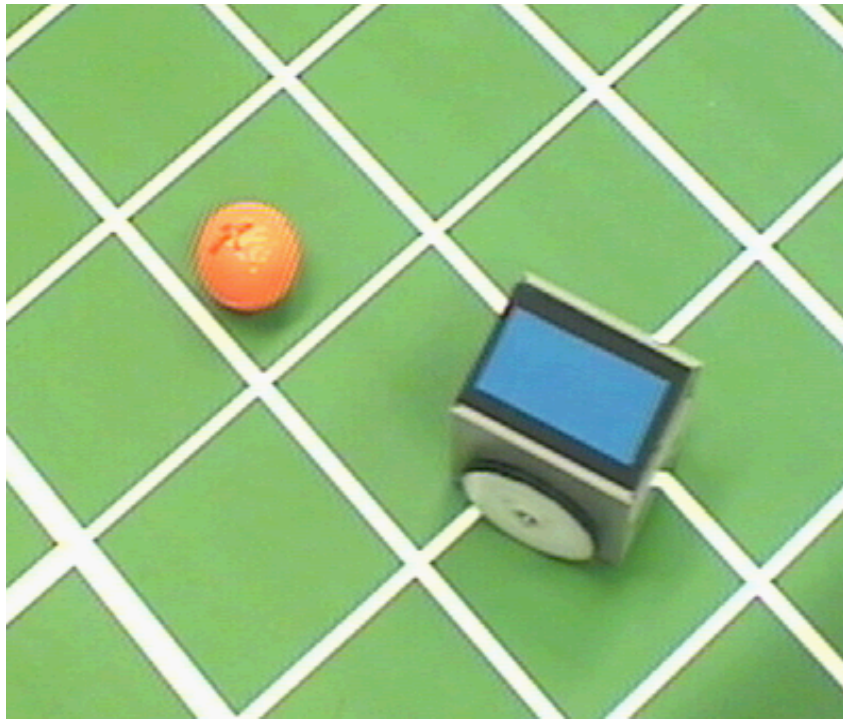
$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + \dots$$

Color Tracking

- Motion estimation of ball and robot for soccer playing using color tracking



Color segmentation with fixed thresholds

- Simple: constant thresholding:
 - selection only **iff** RGB values (r,g,b) simultaneously in R, G, and B ranges:
 - six thresholds [Rmin,Rmax], [Gmin,Gmax], [Bmin,Bmax]:

$$R_{min} < r < R_{max} \text{ and } G_{min} < g < G_{max} \text{ and } B_{min} < b < B_{max}$$

- Alternative: YUV color space
 - RGB values encode intensity of each color
 - YUV:
 - U and V together color (or chrominance)
 - Y brightness (or luminosity)
 - bounding box in YUV space => greater stability wrt. changes in illumination