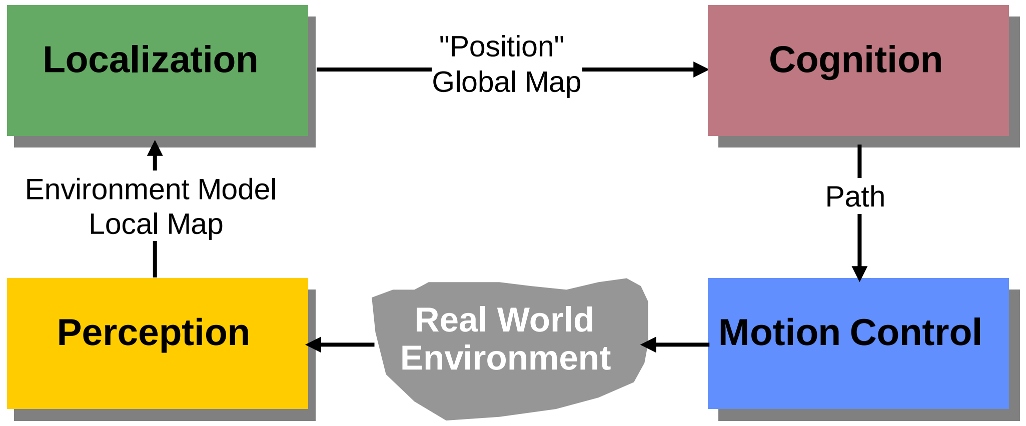
Autonomous Mobile Robots



Perception

Sensors

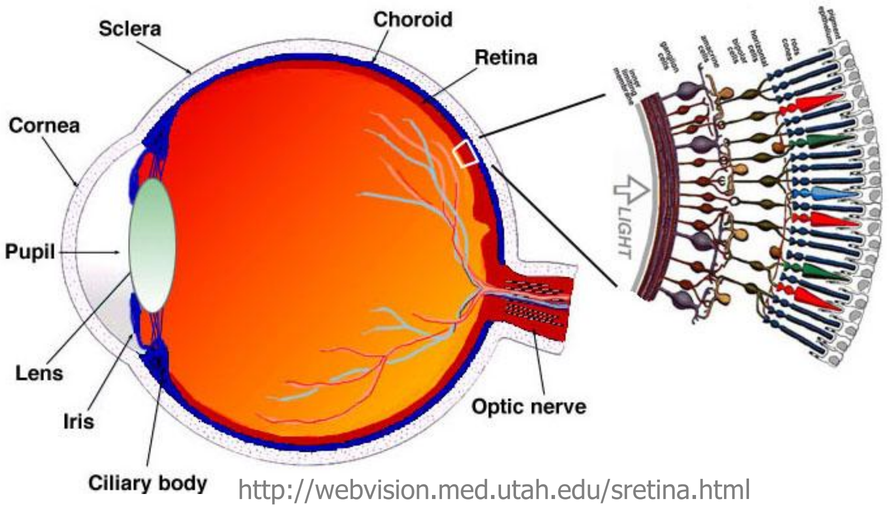
Vision

Uncertainties, Line extraction from laser scans

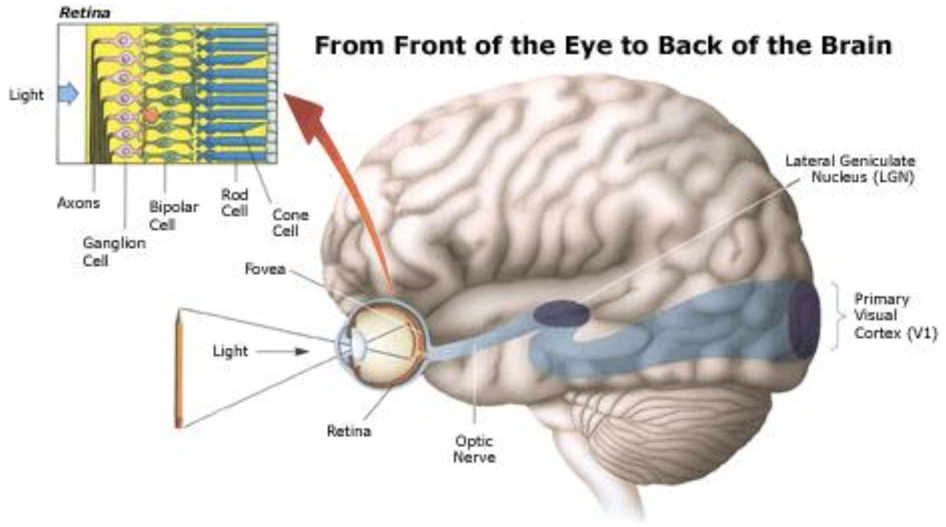
One picture, a thousand words

- Of all our senses, **vision** is the most powerful in aiding our perception of the 3D world around us.
- Retina is ~1000m^2. Contains millions of **photoreceptors** (120 mil. rods and 7 mil. Cones for colour sampling)
- Provides **enormous** amount of information: data-rate of ~3GBytes/s

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



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


Retina
Light
Axons
Ganglion Cell
Bipolar Cell
Rod Cell
Cone Cell
From Front of the Eye to Back of the Brain
Lateral Geniculate Nucleus (LGN)
Fovea
Light
Primary Visual Cortex (V1)
Retina
Optic Nerve

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

Vision for Robotics

- Enormous descriptability of images
  ➪ a lot of data to process (human vision involves 60 billion neurons!)
- Not sensible to copy the biology, but learn from it
- Capture light ➪ Convert to digital image
  ➪ Process to get 'salient' information
- Vision is increasingly popular as a sensing modality:
  - compactness,
  - compatibility,
  - low cost, …
  - HW advances necessary to support the processing of images

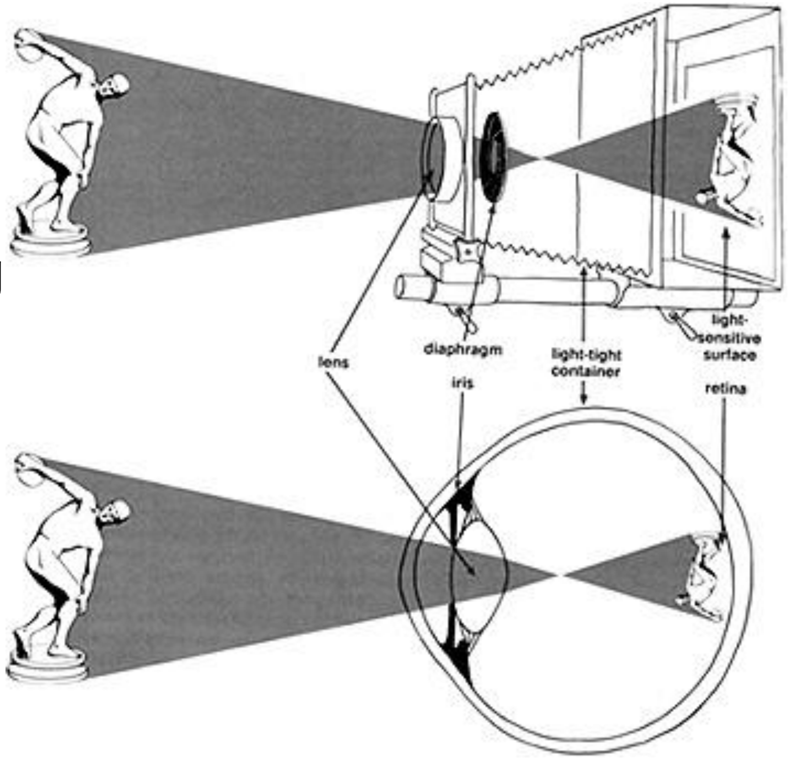
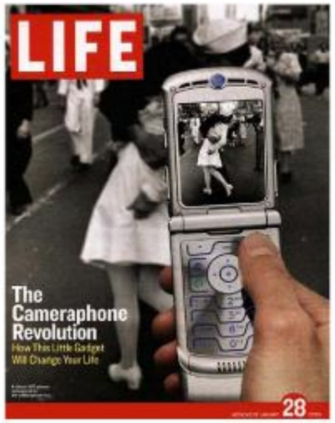
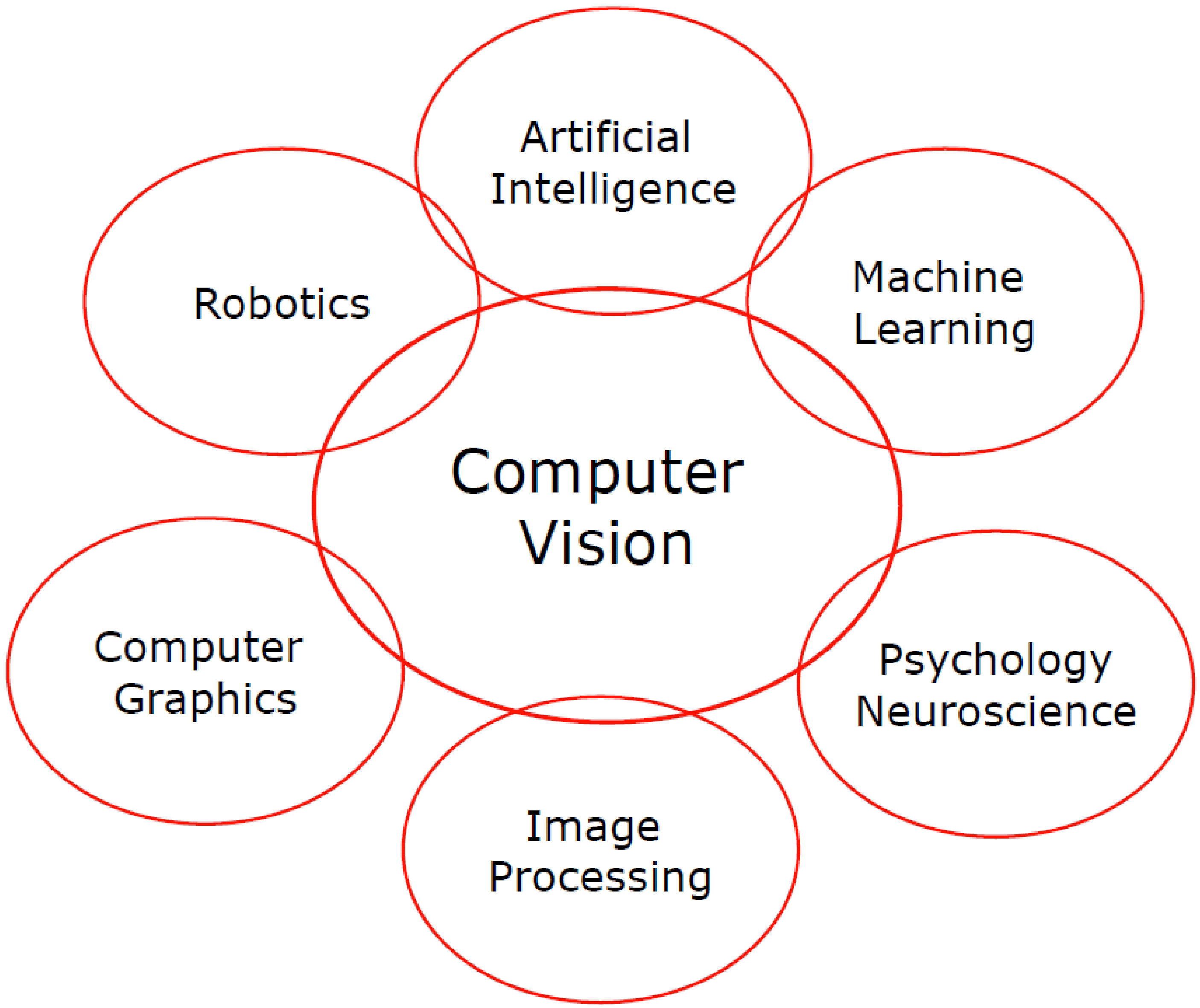


Image: Nicholas M. Short

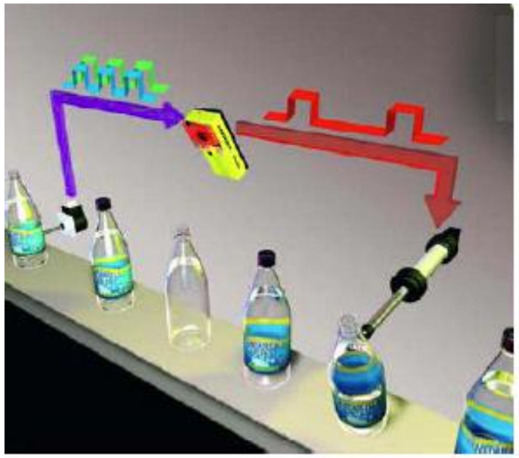
Computer Vision


LIFE
The Cameraphone Revolution
How This Little Gadget Will Change Your Life
28

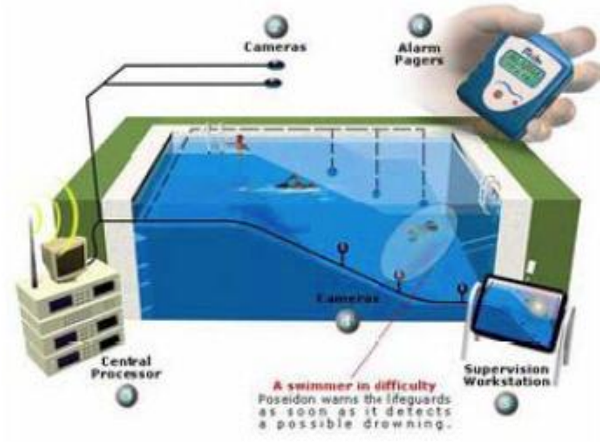
Connection to other disciplines



Applications of Computer Vision


Factory inspection

Reading license plates, checks, ZIP codes



Monitoring for safety (Poseidon)

Surveillance

Autonomous driving, robot navigation

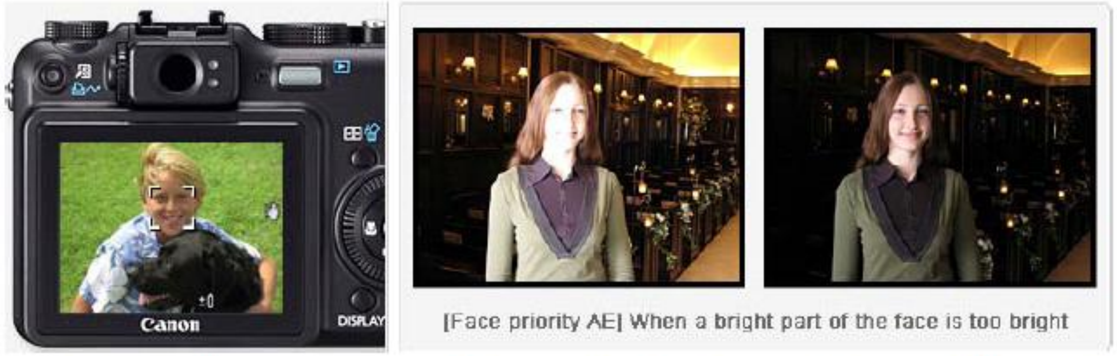
Driver assistance (collision warning, lane departure warning, rear object detection)

Applications of Computer Vision

Assistive technologies

Entertainment (Sony EyeToy)

Movie special effects


Canon
DISPLAY
[Face priority AE] When a bright part of the face is too bright
Digital cameras (face detection for setting focus, exposure)

Visual search http://www.kooaba.com/

Today’s Topics

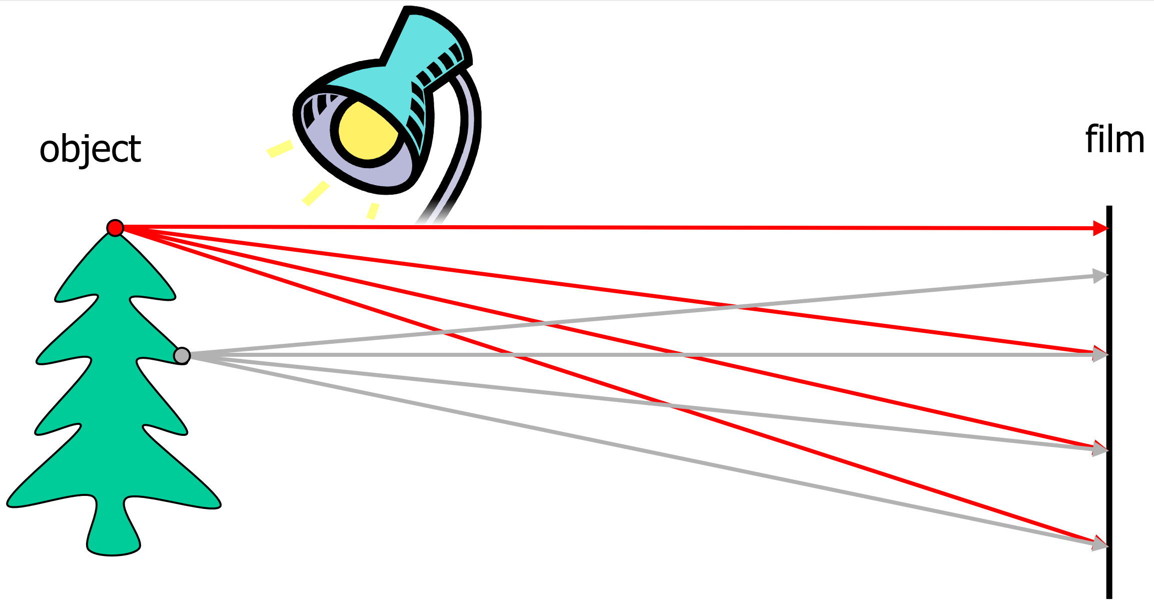
Section 4.2 in the book

- Pinhole Model
- Perspective Projection
- Stereo Vision
- Optical Flow
- Color Tracking

The camera

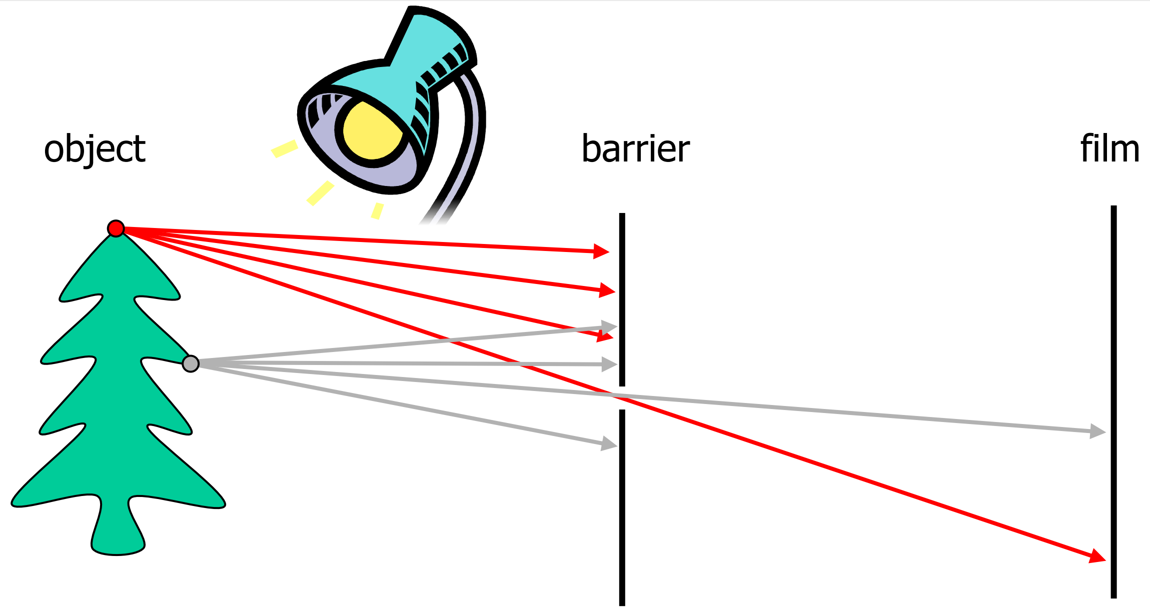
Sony Cybershot WX1

How do we see the world?



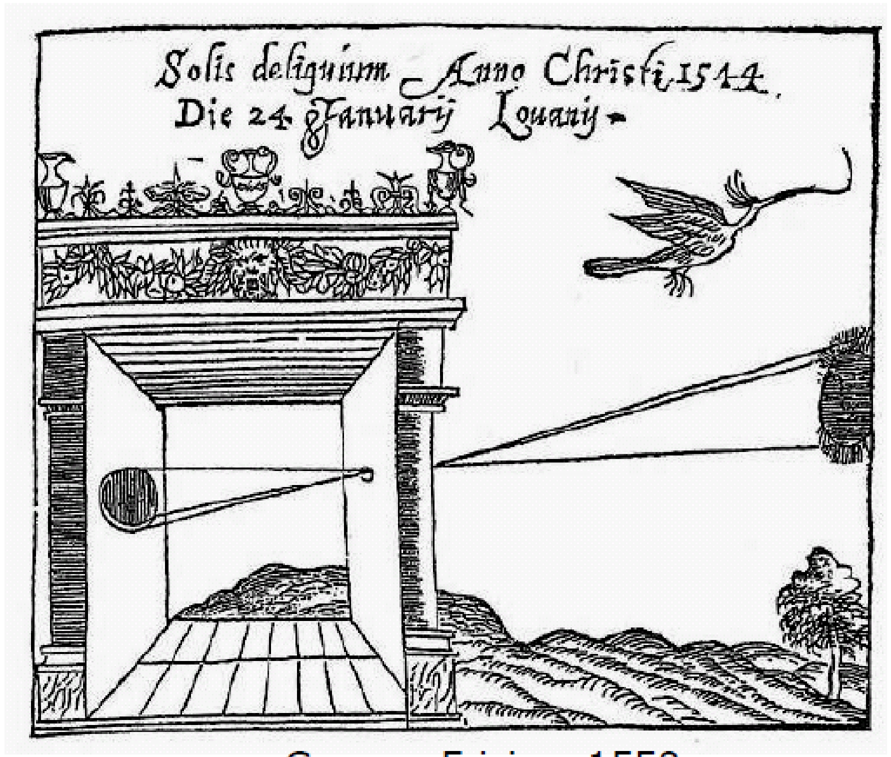
- Place a piece of film in front of an object
  ⇨ Do we get a reasonable image?

Pinhole camera



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

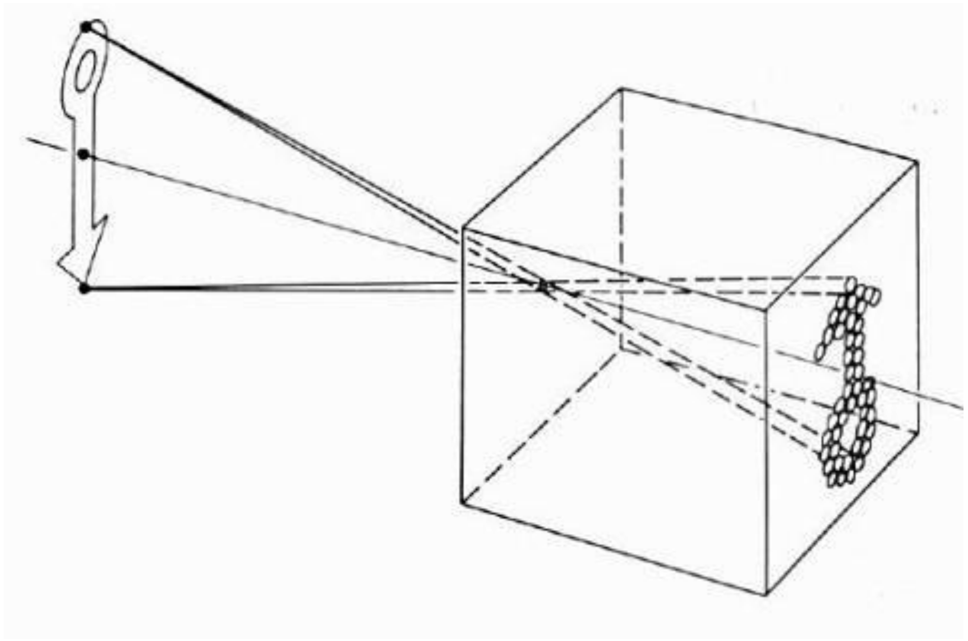
Camera obscura



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

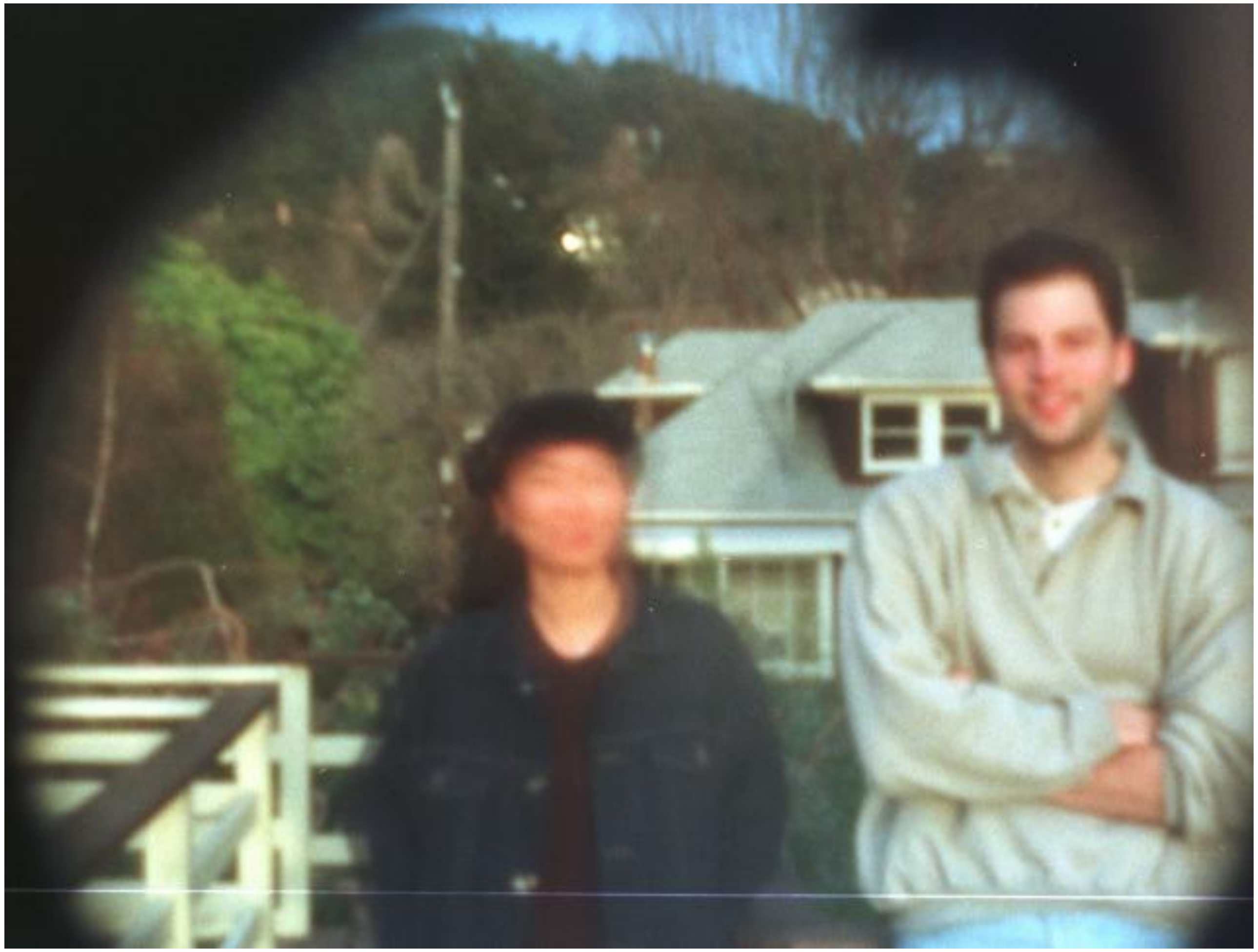
Pinhole camera model



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

Slide by Steve Seitz

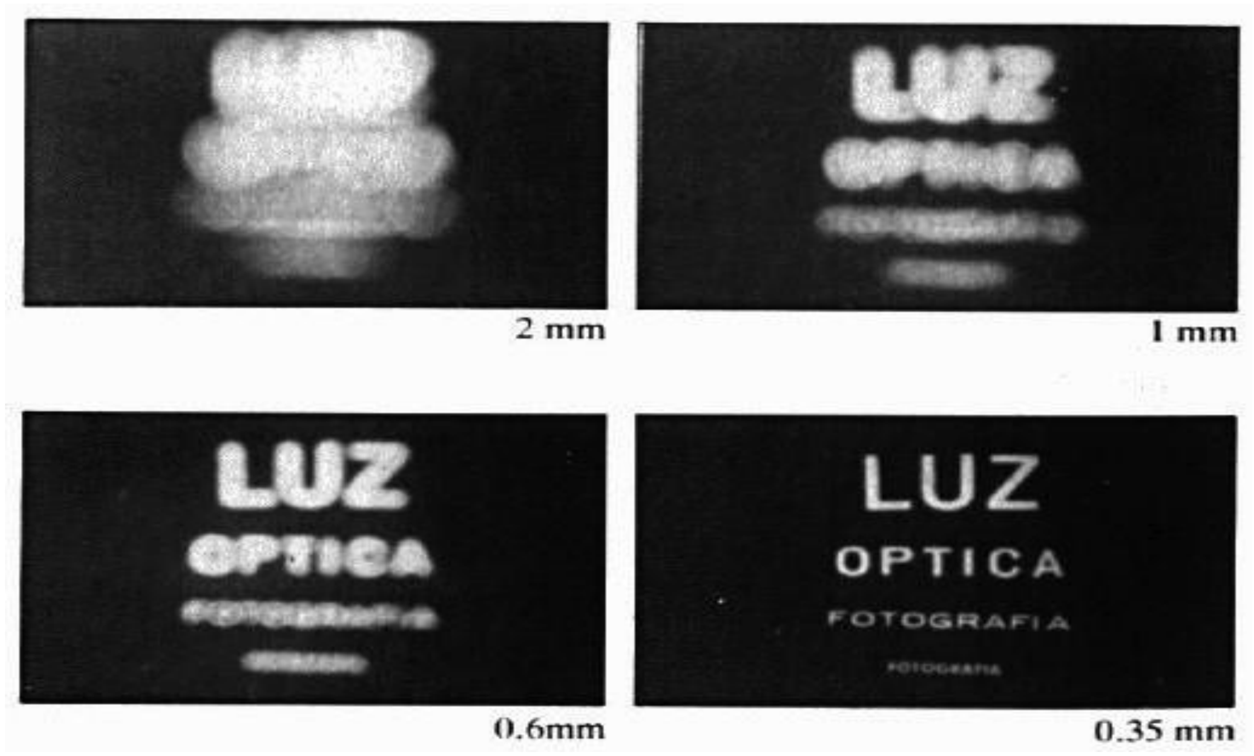
Home-made pinhole camera



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

Why so
blurry?

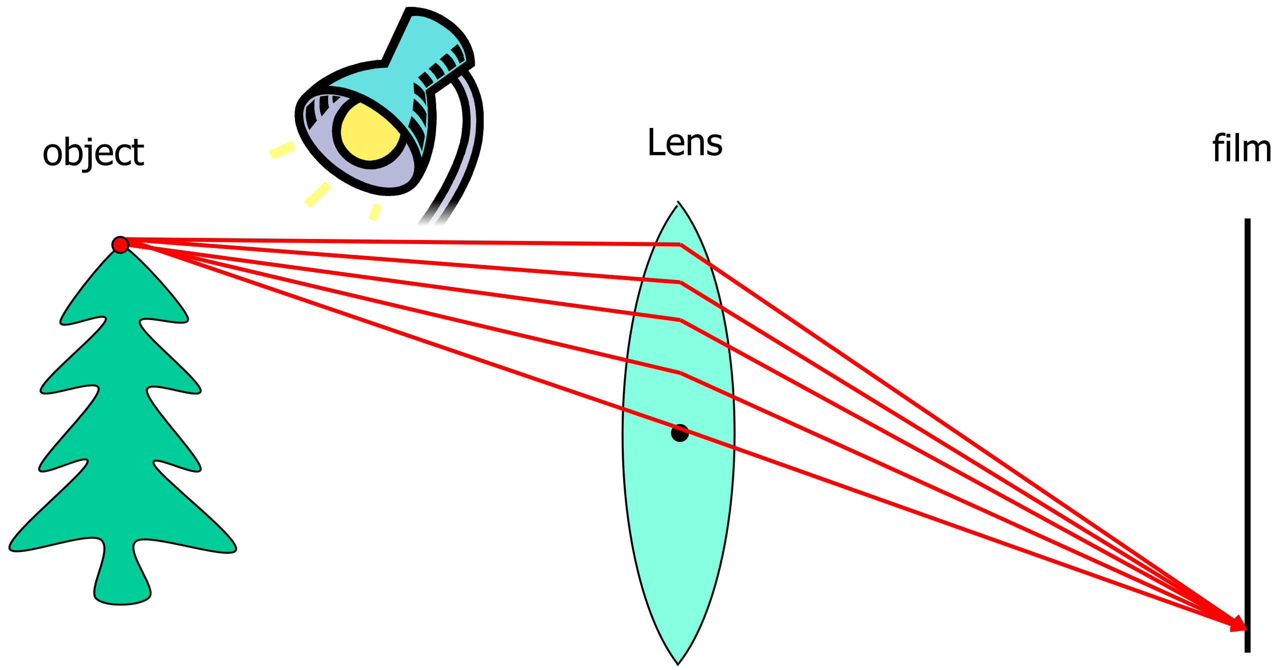
Shrinking the aperture



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

Slide by Steve Seitz

Solution: adding a lens



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

CS4733 Class Notes, Stereo Imaging

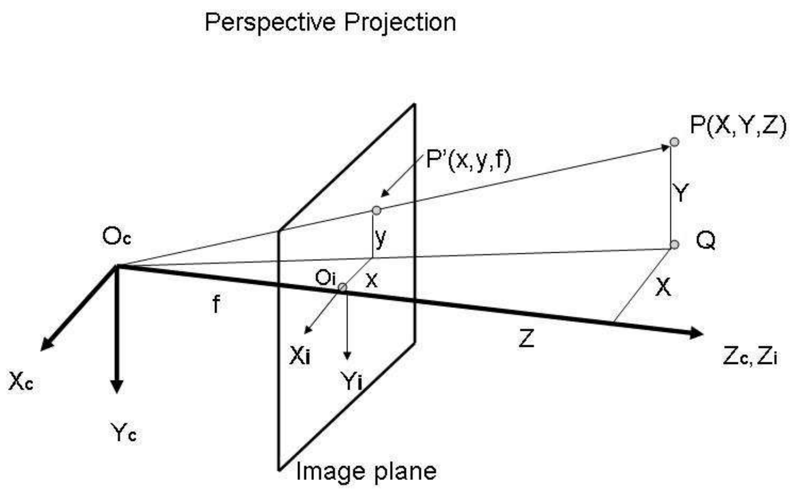


Figure 1: Perspective imaging geometry showing relationship between 3D points and image plane points.

1 Stereo Imaging: Camera Model and Perspective Transform

We typically use a pinhole camera model that maps points in a 3-D camera frame to a 2-D projected image frame. In figure 1, we have a 3D camera coordinate frame X_c, Y_c, Z_c with origin O_c, and an image coordinate frame X_i, Y_i, Z_i with origin O_i. The focal length is f. Using similar triangles, we can relate image plane and world space coordinates. We have a 3D point $P = (X, Y, Z)$ which projects onto the image plane at $P' = (x, y, f)$. O_c is the origin of the camera coordinate system, known as the *center of projection* (COP) of the camera.

Using similar triangles, we can write down the folowing relationships:

$$\frac{X}{x} = \frac{Z}{f} \quad ; \quad \frac{Y}{y} = \frac{Z}{f} \quad ; \quad x = f \cdot \frac{X}{Z} \quad ; \quad y = f \cdot \frac{Y}{Z}$$

If $f = 1$, note that perspective projection is just scaling a world coordinate by its Z value. Also note that all 3D points along a line from the COP through a designated position (x, y) on the image plane will have the same image plane coordinates.

We can also describe perspective projection by the matrix equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where s is a scaling factor and $[x, y, 1]^T$ are the projected coordinates in the image plane.

We can generate *image space* coordinates from the projected camera space coordinates. These are the actual pixels values that you use in image processing. Pixels values (u, v) are derived by scaling the camera image plane coordinates in the x and y directions (for example, converting mm to $pixels$), and adding a translation to the origin of the image space plane. We can call these scale factors D_x and D_y, and the translation to the origin of the image plane as (u_0, v_0).

If the pixel coordinates of a projected point (x,y) are (u,v) then we can write:

$$\frac{x}{D_x} = u - u_0; \;\; \frac{y}{D_y} = v - v_0;$$

$$u = u_0 + \frac{x}{D_x}; \;\; v = v_0 + \frac{y}{D_y}$$

where D_x, D_y are the physical dimensions of a pixel and (u_0, v_0) is the origin of the pixel coordinate system. $\frac{x}{D_x}$ and $\frac{y}{D_y}$ are simply the number of pixels, and we center them at the pixel coordinate origin. We can also put this into matrix form as:

$$\begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$P^{image} = T^{image}_{persp} T^{persp}_{camera} P^{camera}$$

In the above, we assumed that the point to be imaged was in the camera coordinate system. If the point is in a previously defined world coordinate system, then we also have to add in a standard $4x4$ transform to express the world coordinate point in camera coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s\cdot u \\ s\cdot v \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^wX \\ {}^wY \\ {}^wZ \\ 1 \end{bmatrix}$$

$$P^{image} = T^{image}_{persp} T^{persp}_{camera} T^{camera}_{world} P^{world}$$

Summing all this up, we can see that we need to find the following information to transform an arbitrary 3D world point to a designated pixel in a computer image:

- 6 parameters that relate the 3D world point to the 3D camera coordinate system (standard 3 translation and 3 rotation): (R, T)
- Focal Length of the camera: f
- Scaling factors in the x and y direcitons on the image plane: (D_x, D_y)
- Translation to the origin of the image plane: (u_0, v_0).

  This is 11 parameters in all. We can break these parameters down into *Extrinsic* parameters which are the 6-DOF transform between the camera coordinate system and the world coordinate system, and the *Intrinsic* parameters which are unique to the actual camera being used, and include the focal length, scaling factors, and location of the origin of the pixel coordinate system.

2 Camera Calibration

Camera calibration is used to find the mapping from 3D to 2D image space coordinates. There are 2 approaches:

- Method I: Find both extrinsic and intrinsic parameters of the camera system. However, this can be difficult to do. The instinsic parameters of the camera may be unknown (i.e. focal length, pixel dimension) and the 6-DOF transform also may be difficult to calculate directly.
- Method 2: An easier method is the "Lumped" transform. Rather than finding individual parameters, we find a composite matrix that relates 3D to 2D. Given the equation below:

  $$P^{image} = T^{image}_{persp} T^{persp}_{camera} T^{camera}_{world} P^{world}$$

  we can lump the 3 T matrices into a 3x4 calibration matrix C:

  $$P^{image} = C\, P^{world}$$

  $$C = T^{image}_{persp} T^{persp}_{camera} T^{camera}_{world}$$

- C is a <u>single</u> 3×4 transform that we can calculate empirically.

$$\overbrace{\begin{bmatrix} C \end{bmatrix}}^{3\times4} \quad \underbrace{\overbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}^{4\times1}}_{\text{3-D homo. vec}} = \underbrace{\overbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}^{3\times1}}_{\text{2-D homo. vec}} \triangleq \underbrace{\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}}_{\text{Pixels}} \quad \begin{array}{l} where \\ u' = \frac{u}{w} \\ v' = \frac{v}{w} \end{array}$$

- Multiplying out the equations, we get:

$$c_{11}x + c_{12}y + c_{13}z + c_{14} = u$$

$$c_{21}x + c_{22}y + c_{23}z + c_{24} = v$$

$$c_{31}x + c_{32}y + c_{33}z + c_{34} = w$$

- Substituting $u = u'w$ and $v = v'w$, we get:

  1. $c_{11}x + c_{12}y + c_{13}z + c_{14} = u'(c_{31}x + c_{32}y + c_{33}z + c_{34})$
  2. $c_{21}x + c_{22}y + c_{23}z + c_{24} = v'(c_{31}x + c_{32}y + c_{33}z + c_{34})$

- How to interpret <u>1</u> and <u>2</u>:

  1. If we know all the c_{ij} and x, y, z, we can find u', v'. This means that if we know calibration matrix C and a 3-D point, we can predict its image space coordinates.
  2. If we know $x,\ y,\ z,\ u',\ v'$, we can find c_{ij}. Each 5-tuple gives 2 equations in c_{ij}. This is the basis for empirically finding the calibration matrix C (more on this later).
  3. If we know $c_{ij},\ u',\ v'$, we have 2 equations in x, y, z. They are the equations of 2 planes in 3-D. 2 planes form an intersecton which is a line. These are the equations of the line emanating from the center of projection of the camera, through the image pixel location u', v' and which contains point x, y, z.

- We can set up a linear system to solve for c_{ij}: $AC = B$

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u'_1x & -u'_1y & -u'_1z \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v'_1x & -v'_1y & -v'_1z \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u'_2x & -u'_2y & -u'_2z \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v'_2x & -v'_2y & -v'_2z \\ . \\ . \\ . \\ . \\ . \\ . \\ . \end{bmatrix} \underbrace{\begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix}}_{\text{We can assume } c_{34}=1} = \begin{bmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ u'_3 \\ v'_3 \\ . \\ . \\ . \\ u'_N \\ v'_N \end{bmatrix}$$

- Each set of points x, y, z, u', v' yields 2 equations in 11 unknowns (the c_{ij}'s).
- To solve for C, A needs to be invertible (square). We can overdetermine A and find a Least-Squares fit for C by using a pseudo-inverse solution.

  If A is $N \times 11$, where $N > 11$,

$$AC = B$$
$$A^TAC = A^TB$$
$$C = \underbrace{(A^TA)^{-1}}_{\text{pseudo inverse}} A^TB$$

3 COMPUTATIONAL STEREO

Stereopsis is an identified human vision process. It is a passive, simple procedure that is robust to changes in lighting, scale, etc. Humans can fuse random dot stereograms that contain no high-level information about the objects in the fused images, yet they can infer depth from these stereograms. The procedure is:

- Camera-Modeling/Image-acquisition
- Feature extraction - identify edges, corners, regions etc.
- Matching/Correspondence - find same feature in both images
- Compute depth from matches - use calibration information to back project rays from each camera and intersect them (triangulation)

- Interpolate surfaces - Matches are sparse, and constraints such as smoothness of surfaces are needed to “fill in” the depth between match points.

Camera Modeling: An important consideration in computational stereo is the setup of the cameras. The **baseline** between the camera centers determines the accuracy of the triangulation. Large baseline means more accuracy; however as the baseline gets larger, the same physical event in each image may not be found.

The cameras also have to be calibrated and registered. Calibration is relatively straightforward, and a variety of methods exist. Some methods extend the simple least squares model we discussed to include non-linear effects of lens distortion (particularly true with short a focal length lens).

Registration is needed to make use of the epipolar constraint. This constraint consists of a plane that includes both camera’s optical centers and a point in 3-D space. This **epilolar plane** intersects both image planes in a straight line.

Feature Extraction: Identifying features in each image that can be matched is an important part of the stereo process. It serves 2 purposes: 1) data reduction so we are not forced to deal with every single pixel as a potential match, and 2) stability - features are seen to be more stable than a single gray level pixel.

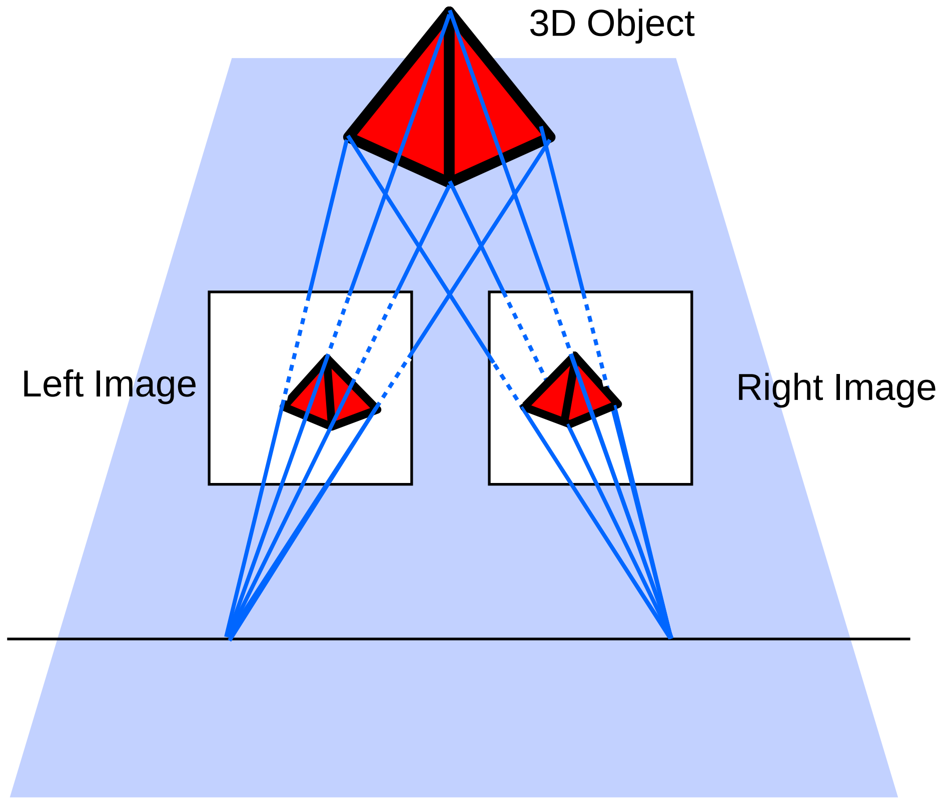
There are 2 approaches: feature-based methods which find primitives such as edges, corners, lines, arcs in each image and match them; and area-based methods that identify regions or areas of pixels that can be matched using correlation based methods. Sometimes both methods are used, with feature-based methods proposing a match and area-based methods centered on the feature used to verify it.

Correspondence: The heart of the stereo problem is a search procedure. Given a pixel in image 1, it can potentially match each of N^2 pixels in the other image. To cut down this search space, cameras are often registered along scan lines. This means that he epipolar plane intersects each image plane along the same scan line. A pixel in image 1 can now potentially match only a pixel along the corresponding scan line in image 2, reducing the search from $O(N^2)$ to $O(N)$. The match criteria can include not only the location of a feature like an edge, but also the edge direction and polarity.

Problems in Matching: A number of problems occur during matching to create false matches: These are occlusions, periodic features such as texture, homogeneous regions without features, baseline separation errors, and misregistered images. Stereo can usually only provide sparse 3-D data at easily identified feature points.

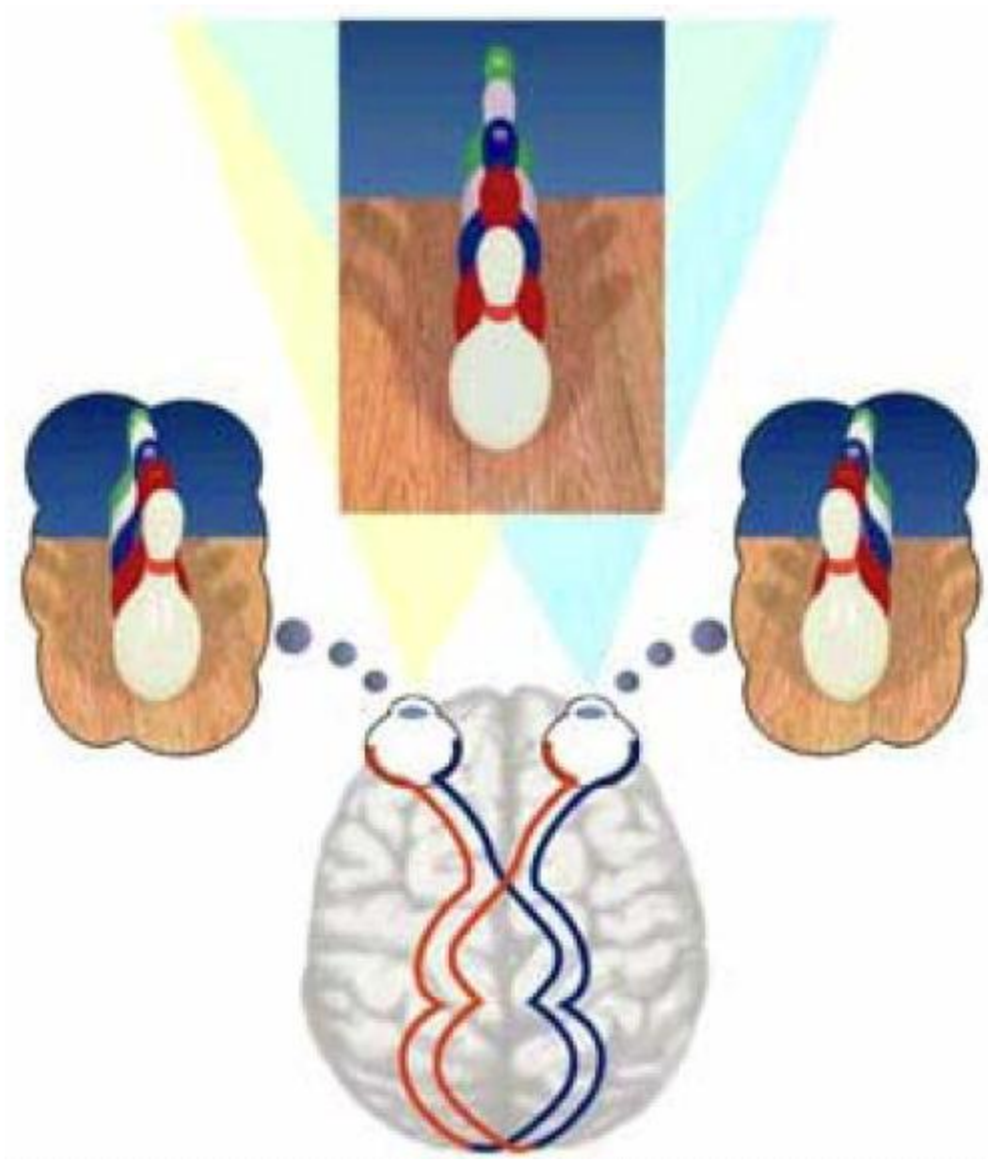
How do we measure distances with cameras?

- Impossible to capture 3D structure from a **single** image. We can only deduct the **ray** on which each image point lies.



- Observe scene from 2 different viewpoints ⇨solve for the intersection of the rays and recover the 3D structure

Disparity in the human retina



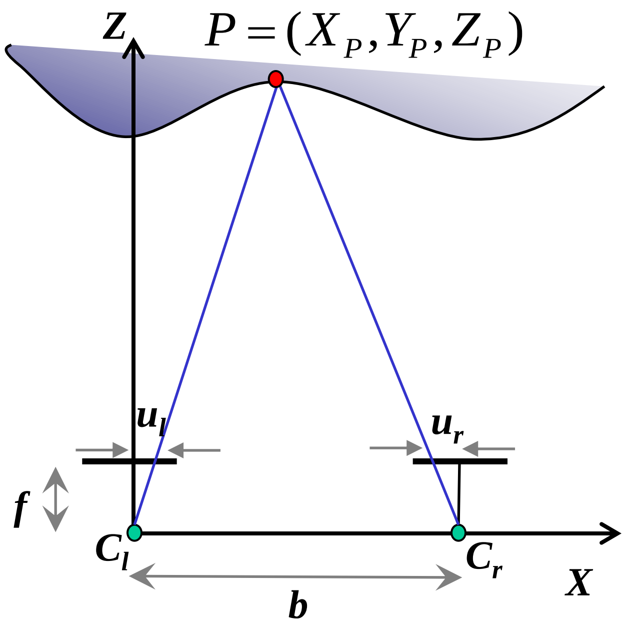
How do we measure distances with cameras?

- 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 - The simplified case

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



Baseline
distance between the optical centers of the two cameras

From Similar Triangles:

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

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

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

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

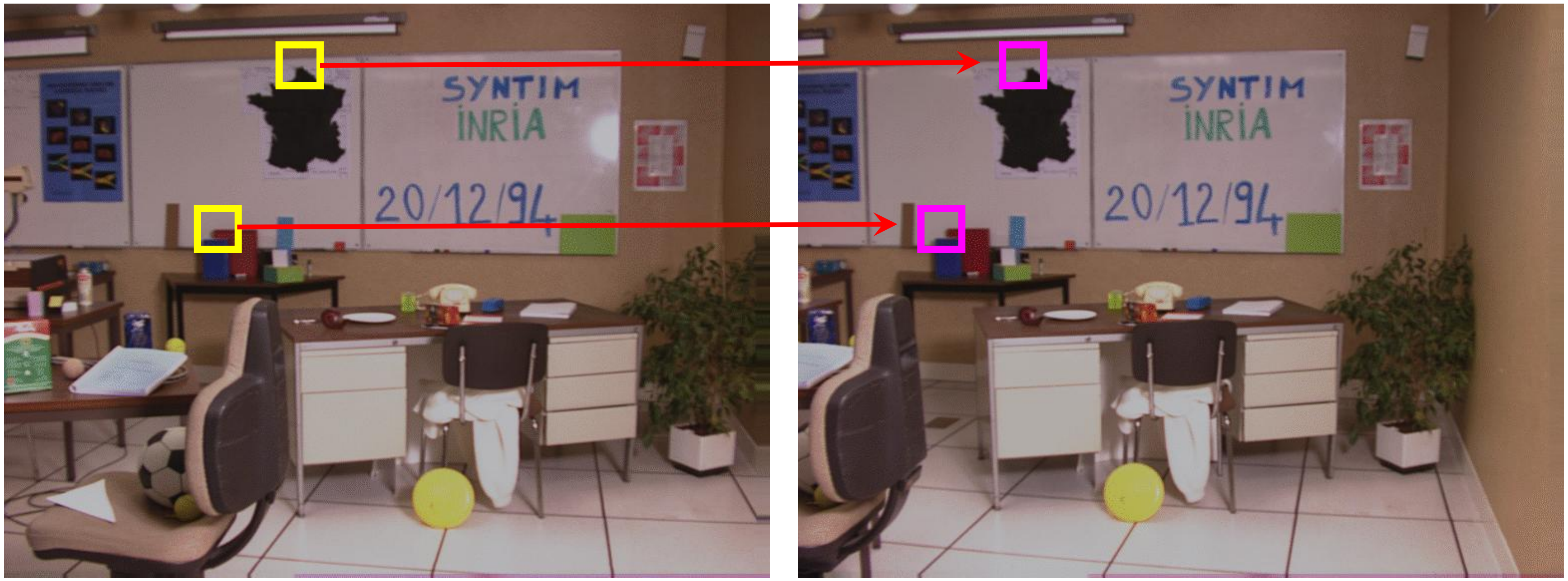
Stereo Vision facts

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

1. Depth is inversely proportional to disparity $(u_l - u_r)$
   - Foreground objects have bigger disparity than background objects
2. Disparity is proportional to stereo-baseline $\boldsymbol{b}$
   - The smaller the baseline $\boldsymbol{b}$ the more uncertain our estimate of depth
   - However, as $\boldsymbol{b}$ is increased, some objects may appear in one camera, but not in the other (remember both cameras have parallel optical axes)
3. The projections of a single 3D point onto the left and the right stereo images are called **'correspondence pair'**

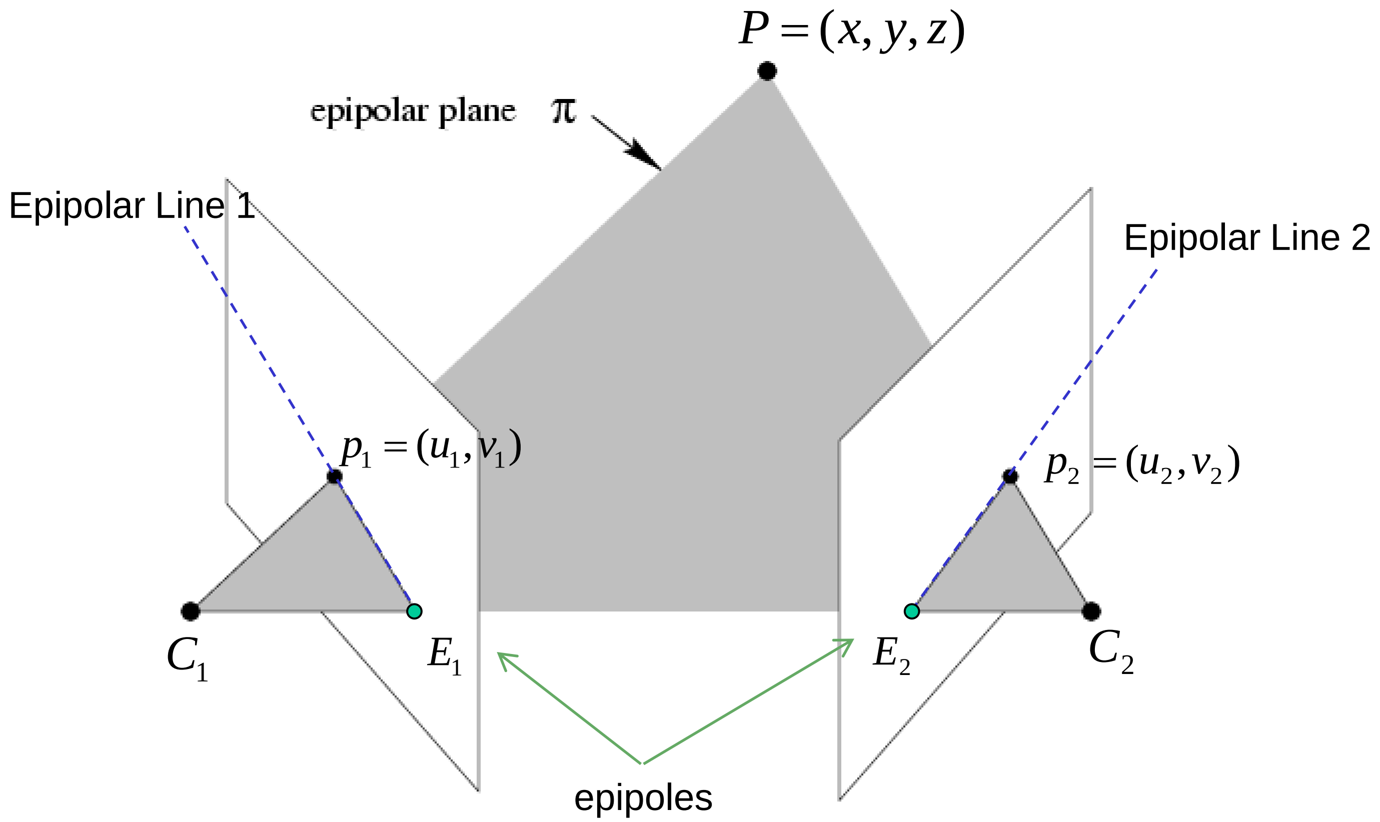
Correspondence Problem

- Which patch in the Left image, corresponds to the projection of the same 3D scene point on the Right image?
- Correspondence search could be done by testing the query-patches at all pixel positions in the other image. Typical **similarity measures** are the Cross-Correlation and Sum of Squared Differences
- **Exhaustive** image search can be computationally very expensive! Is there a way to make the correspondence search 1-dimensional?



Epipolar Geometry

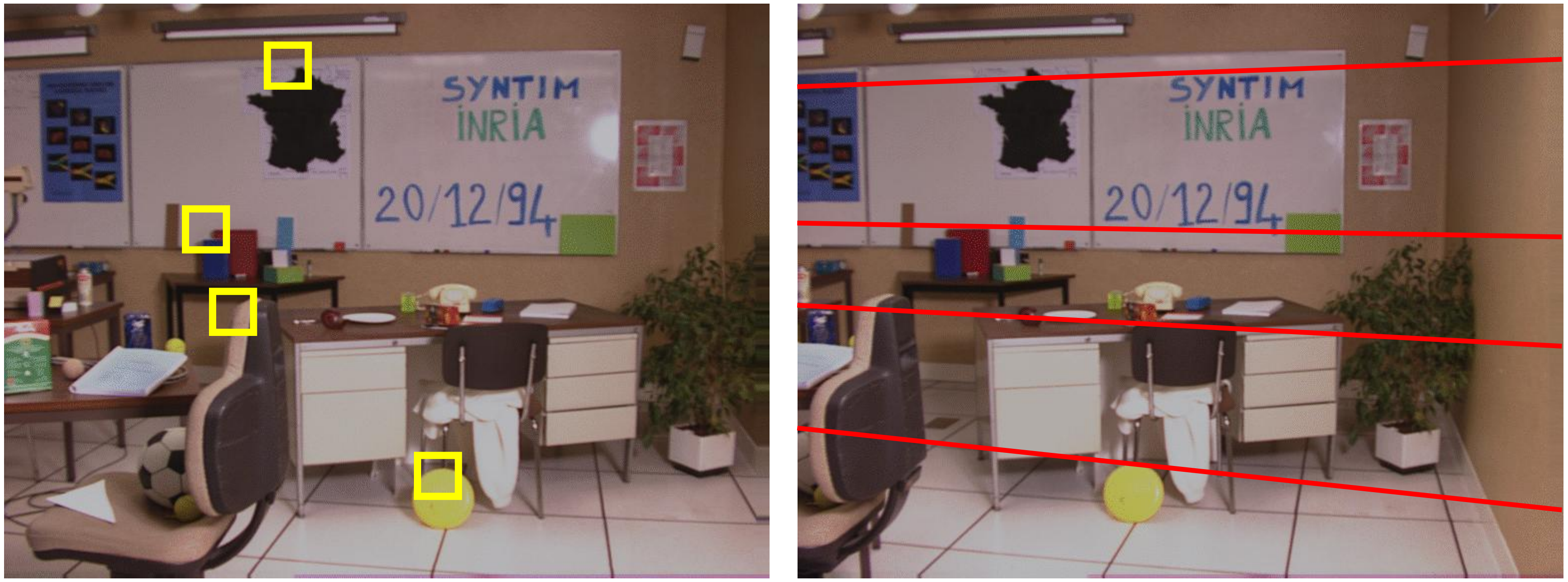
- Triangulating 2 rays from image correspondences
- Impose the **epipolar constraint** to aid matching: search for a correspondence along the epipolar line



- The epipolar plane is defined by a 3D point P and the optical centers.

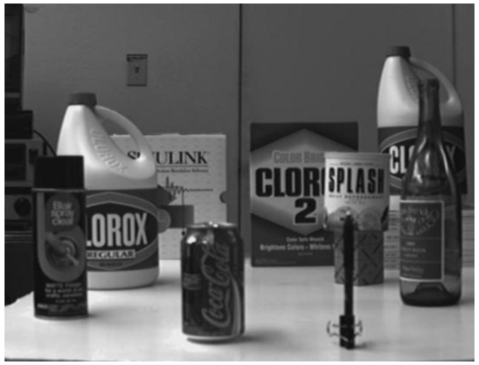
Correspondence Problem: Epipolar Constraint

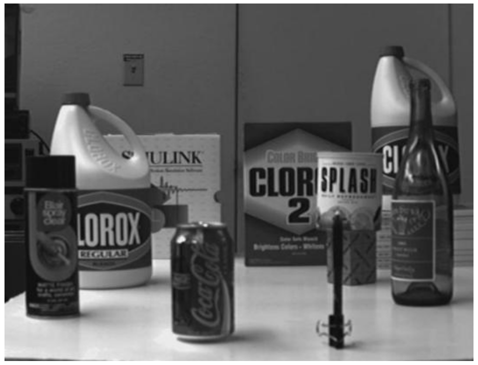
- Thanks to the epipolar constraint, conjugate points can be searched along epipolar lines: this reduces the computational cost to 1 dimension!

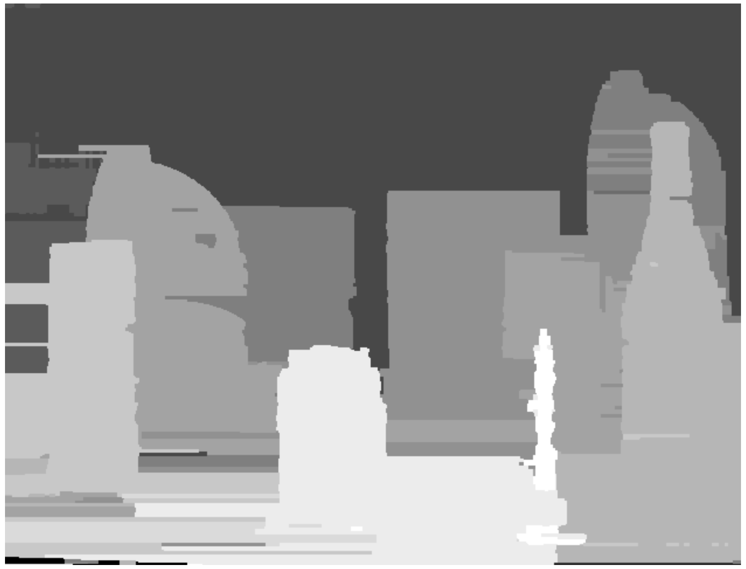


Stereo Vision Output 1 – Disparity map

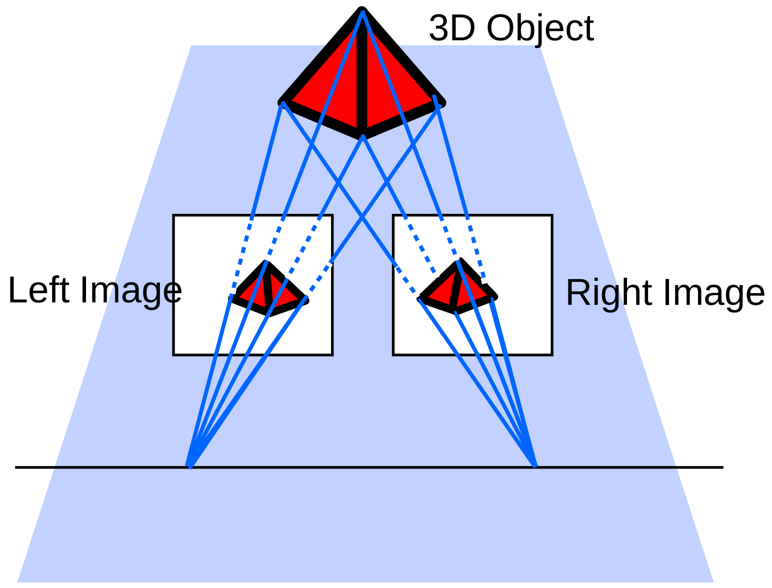
- Find the correspondent points of **all image pixels** of the original images
- Compute the disparity for each pair of correspondences
- **Disparity map**: holds the disparity value at every pixel
- Usually visualized as grey-scale images. If lighter color corresponds to larger disparities, then objects closer to the camera appear lighter, than those further away.


Left image


Right image


Disparity map

Stereo Vision - summary



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

Structure from motion

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