Introduction to 3D Imaging: Perceiving 3D from 2D Images

How can we derive 3D information from one or more 2D images?

There have been 2 approaches:

1. intrinsic images: a 2D representation that stores some 3D properties of the scene
2. 3D shape from X: methods of inferring 3D depth information from various sources

What can you determine about

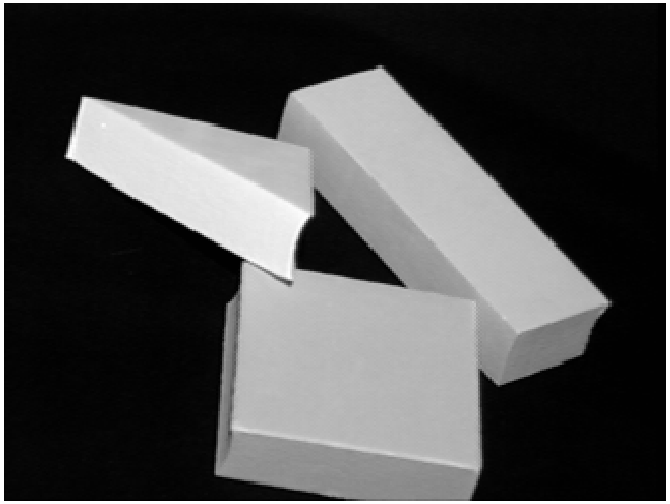
1. the sizes of objects
2. the distances of objects from the camera?

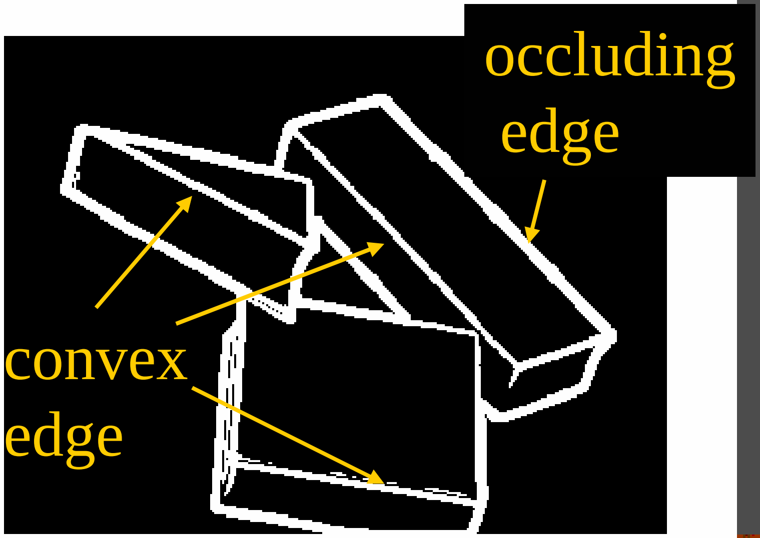
What knowledge do you use to analyze this image?

What objects are shown in this image?
How can you estimate distance from the camera?
What feature changes with distance?

Intrinsic Images: 2.5 D

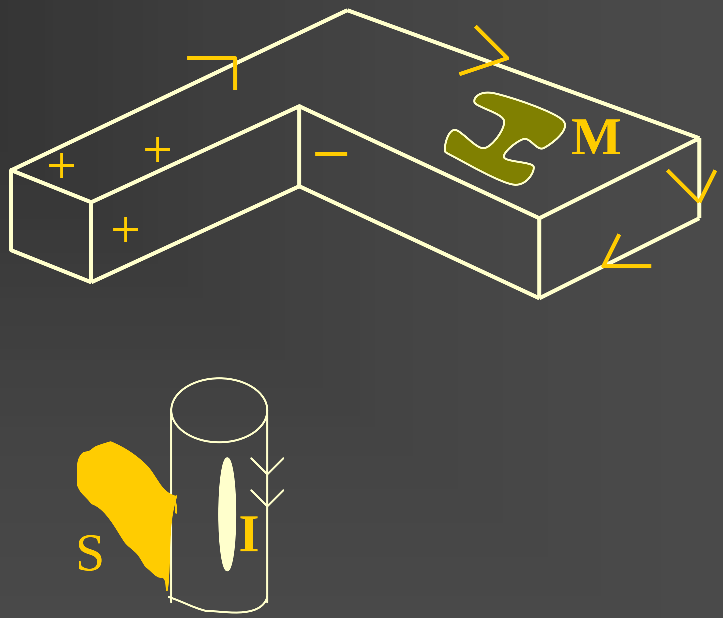
The idea of intrinsic images is to label features of a 2D image with information that tells us something about the 3D structure of the scene.





Contour Labels for Intrinsic Images

- convex crease (+)
- concave crease (-)
- blade (>)
- limb (>>)
- shadow (S)
- illumination boundary (I)
- reflectance boundary (M)



Labeling Simple Line Drawings

- Huffman and Clowes showed that blocks world drawings could be labeled (with +, -, >) based on real world constraints.
- Labeling a simple blocks world image is a consistent labeling problem!
- Waltz extended the work to cracks and shadows and developed one of the first discrete relaxation algorithms, known as Waltz filtering.

Problems with this Approach

- Research on how to do these labelings was confined to perfect blocks world images
- There was no way to extend it to real images with missing segments, broken segments, nonconnected junctions, etc.
- It led some groups down the wrong path for a while.

3D Shape from X

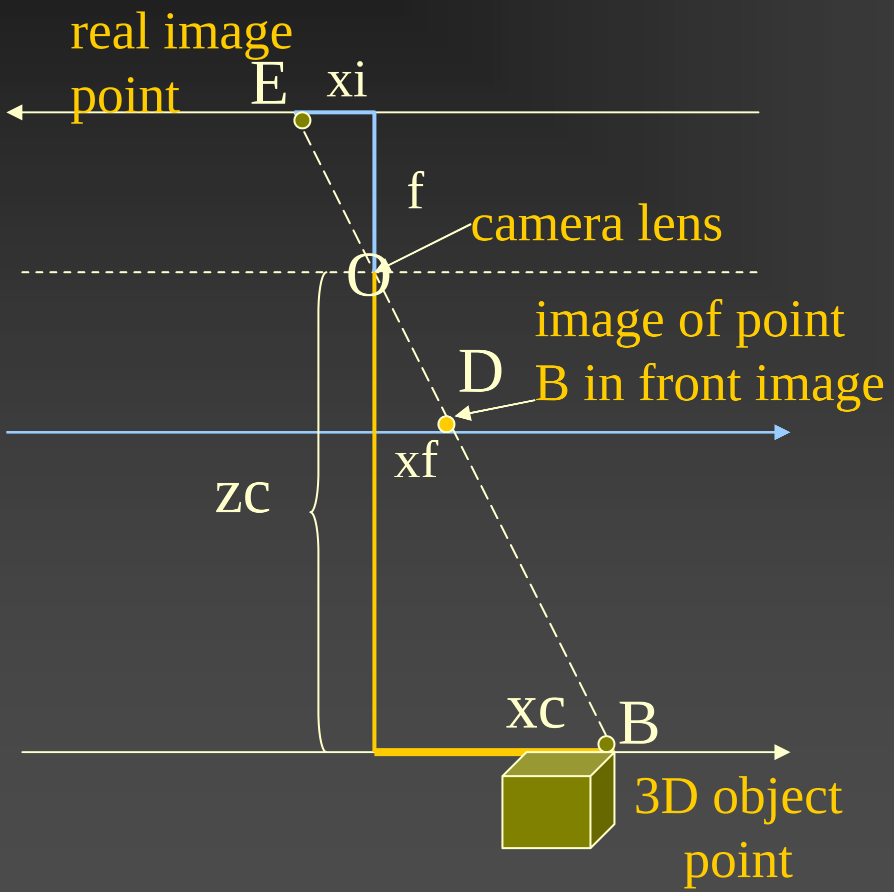
- shading
- silhouette
- texture

mainly research

- stereo
- light striping
- motion

used in practice

Perspective Imaging Model: 1D



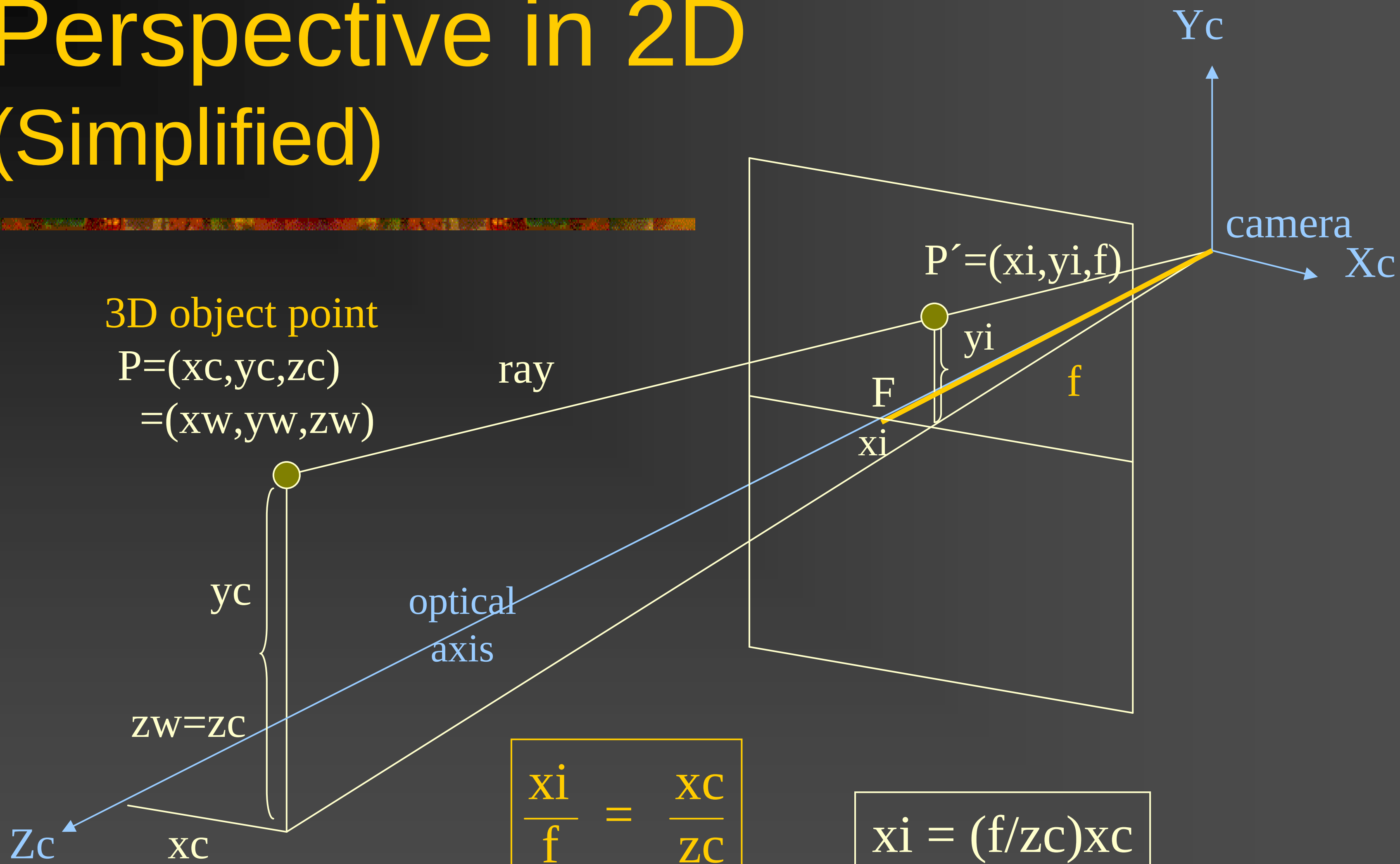
This is the axis of the real image plane.

O is the center of projection.

This is the axis of the front image plane, which we use.

$$\frac{xi}{f} = \frac{xc}{zc}$$

Perspective in 2D (Simplified)



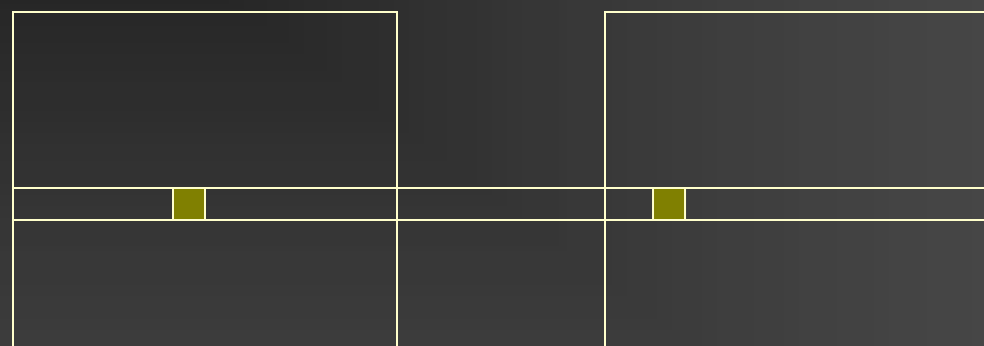
$$\frac{xi}{f} = \frac{xc}{zc}$$

$$\frac{yi}{f} = \frac{yc}{zc}$$

$$xi = (f/zc)xc$$
$$yi = (f/zc)yc$$

Here camera coordinates equal world coordinates.

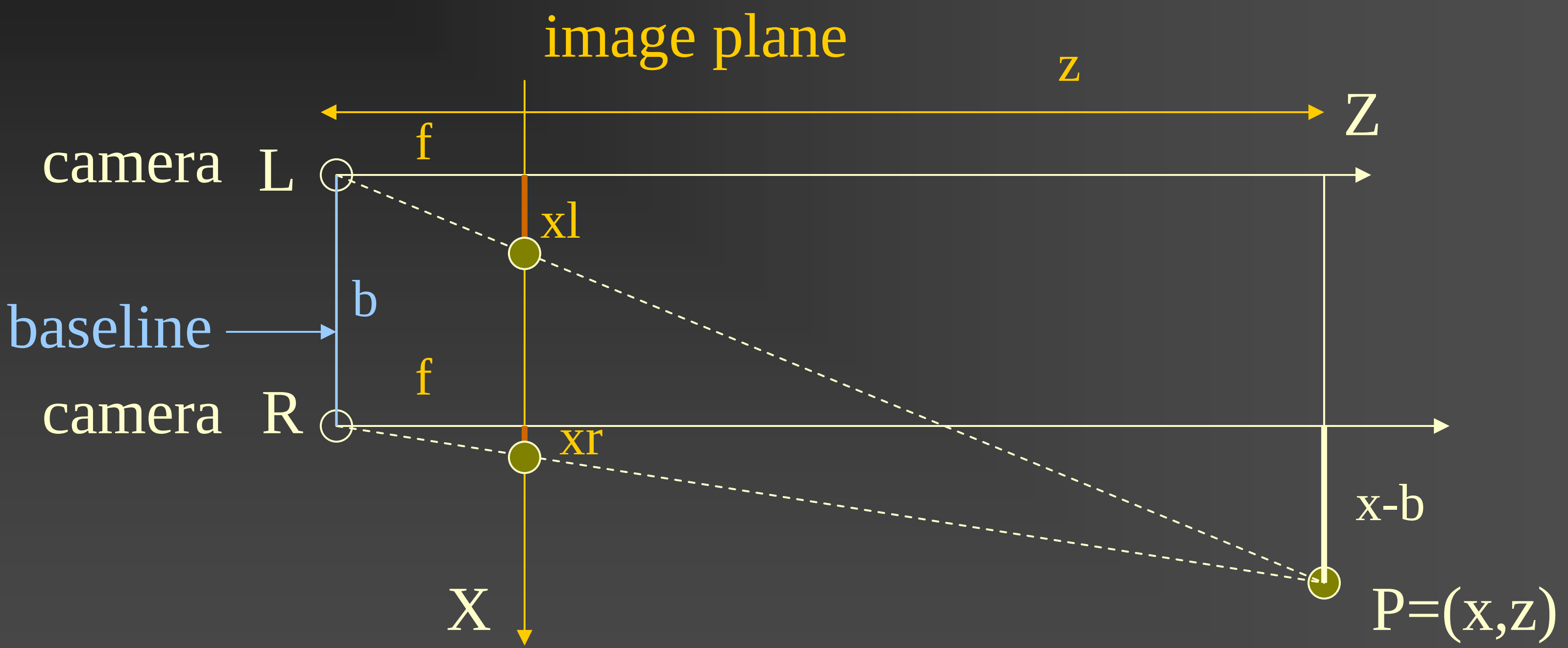
3D from Stereo



disparity: the difference in image location of the same 3D point when projected under perspective to two different cameras.

d = xleft - xright

Depth Perception from Stereo
Simple Model: Parallel Optic Axes



$$\frac{z}{f} = \frac{x}{xl}$$

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

$$\frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

y-axis is perpendicular to the page.

Resultant Depth Calculation

For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr) with disparity d:

z = f*b / (xl - xr) = f*b/d

x = xl*z/f or b + xr*z/f

y = yl*z/f or yr*z/f

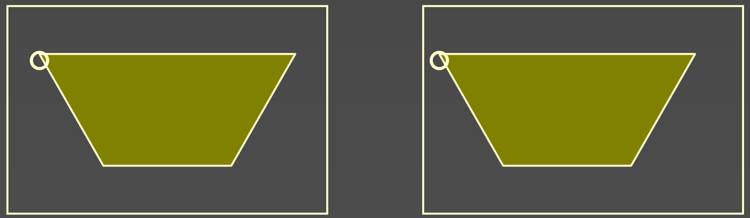
This method of determining depth from disparity is called **triangulation.**

Finding Correspondences

- If the correspondence is correct, triangulation works **VERY** well.

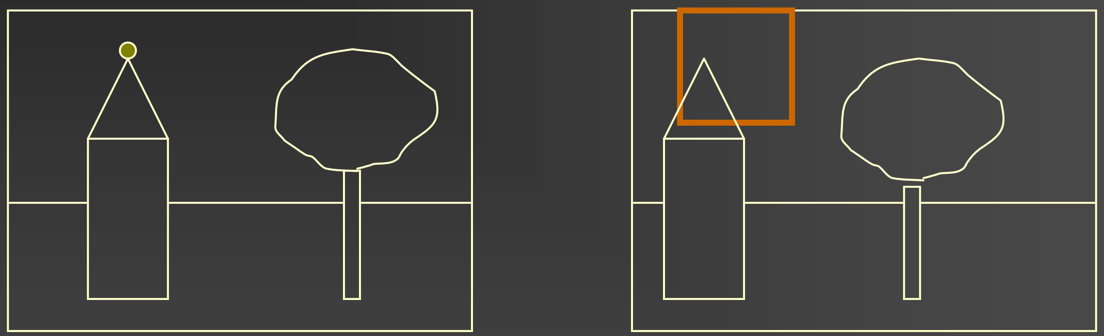
- But correspondence finding is not perfectly solved. for the general stereo problem.

- For some very specific applications, it can be solved for those specific kind of images, e.g. windshield of a car.



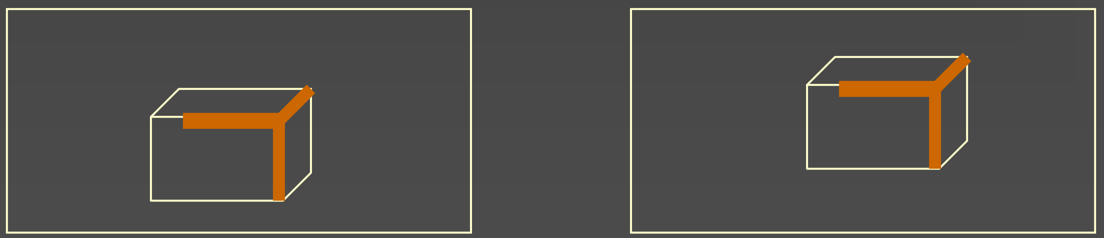
2 Main Matching Methods

1. Cross correlation using small windows.



dense

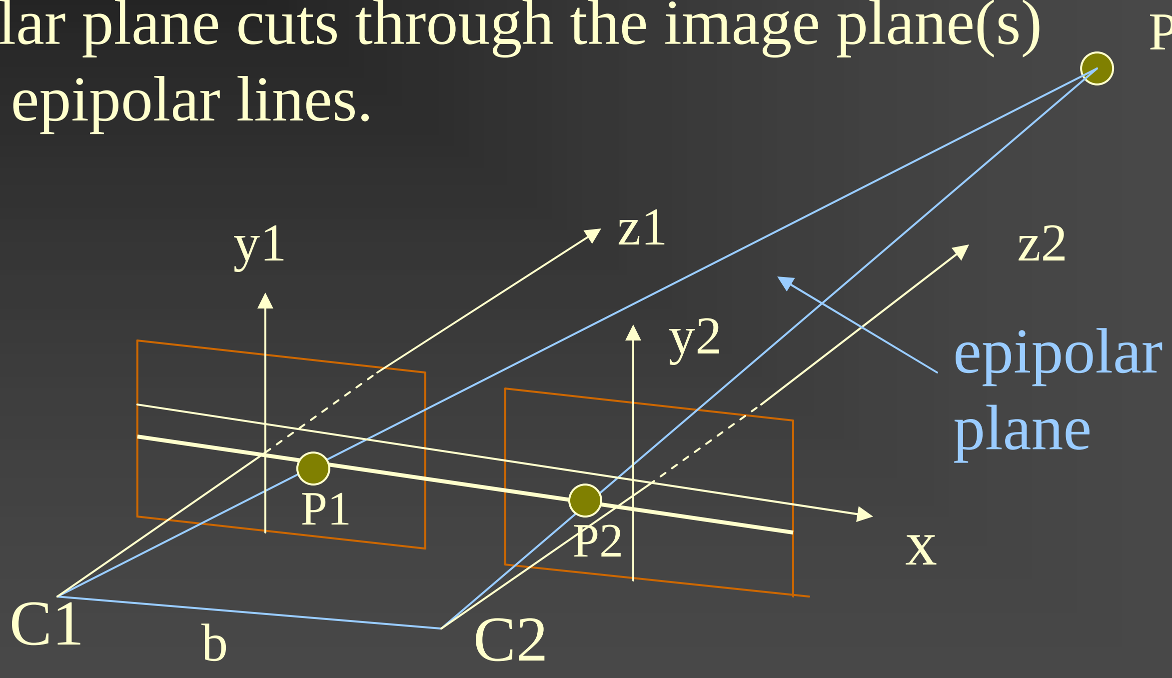
2. Symbolic feature matching, usually using segments/corners.



sparse

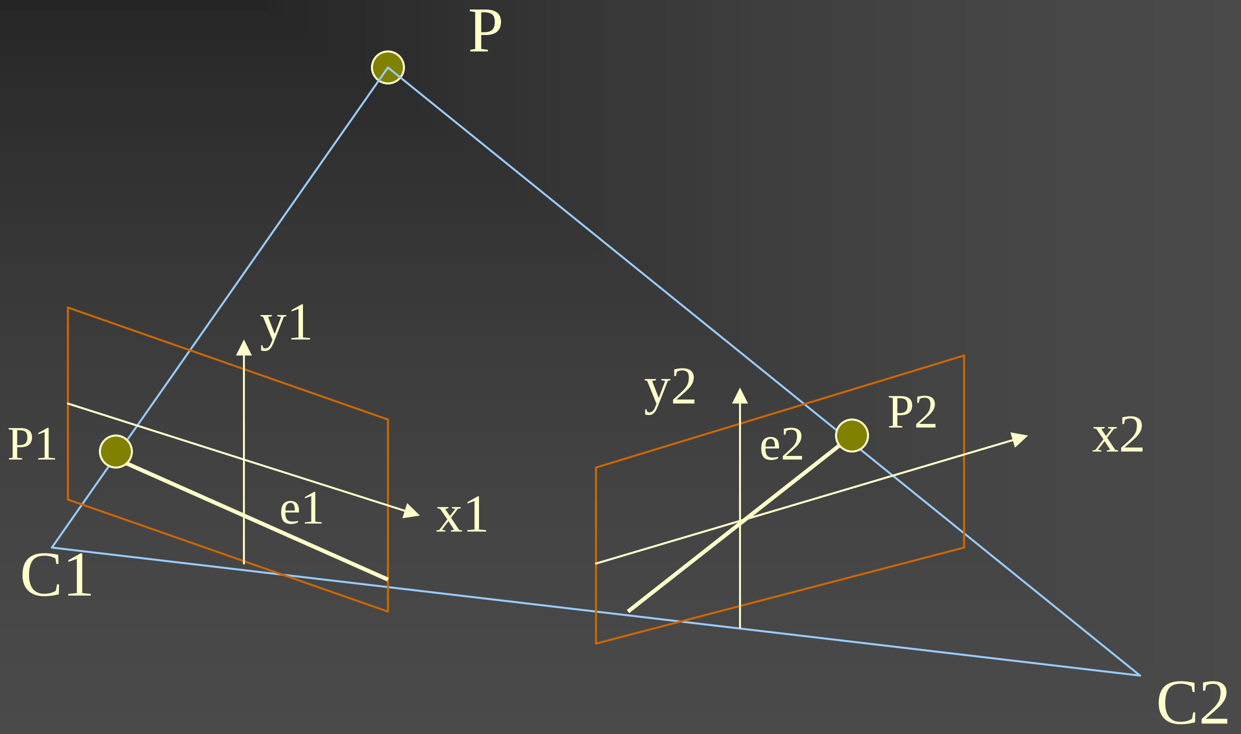
Epipolar Geometry Constraint: 1. Normal Pair of Images

The epipolar plane cuts through the image plane(s) forming 2 epipolar lines.

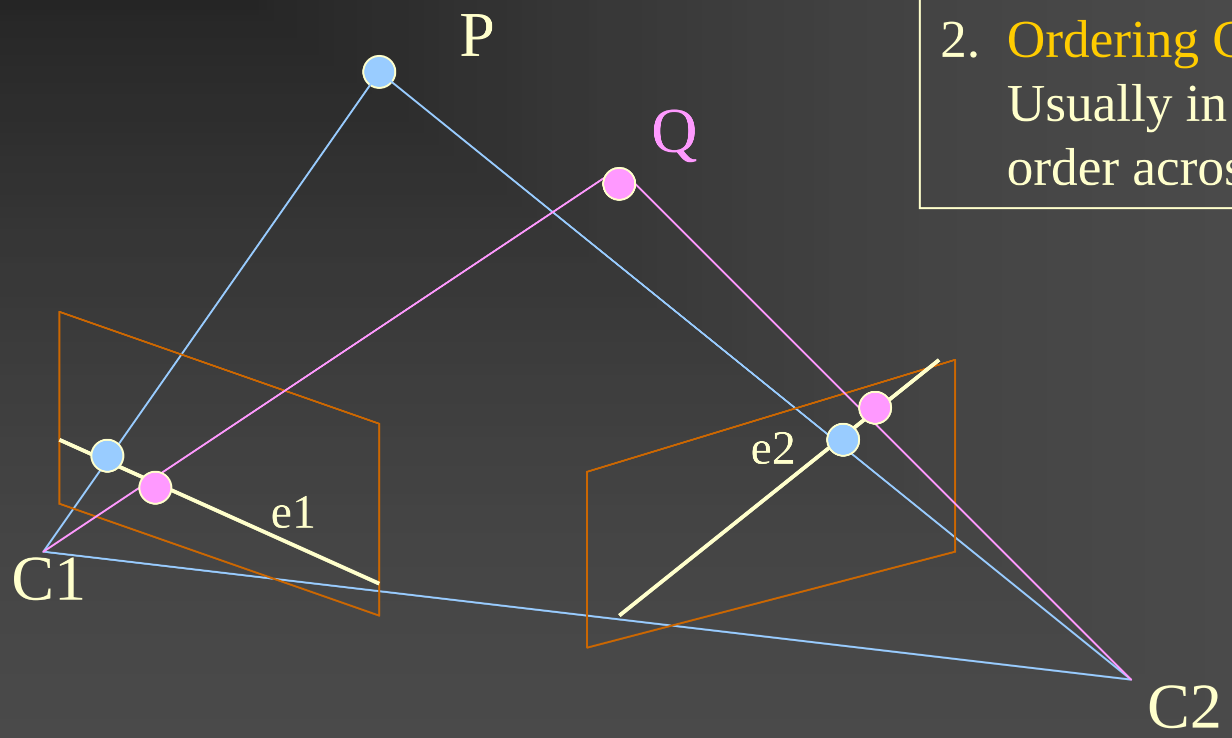


The match for P1 (or P2) in the other image, must lie on the same epipolar line.

Epipolar Geometry: General Case



Constraints



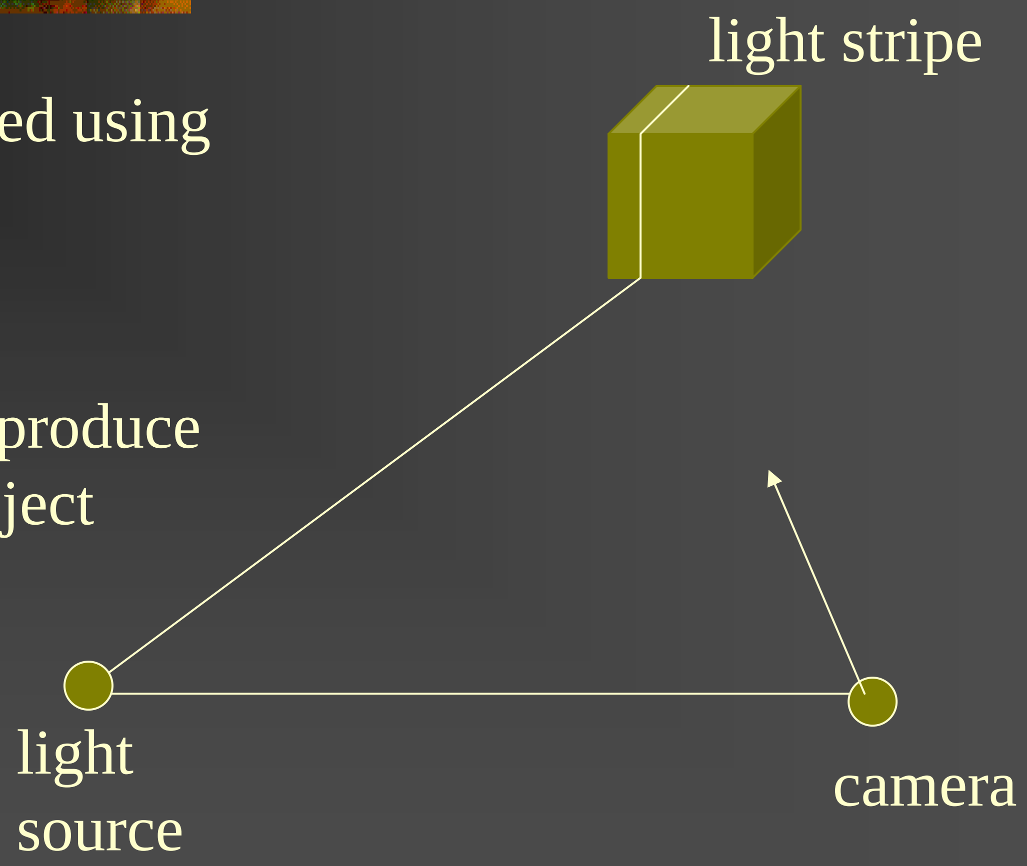
1. Epipolar Constraint: Matching points lie on corresponding epipolar lines.

2. Ordering Constraint: Usually in the same order across the lines.

Structured Light

3D data can also be derived using

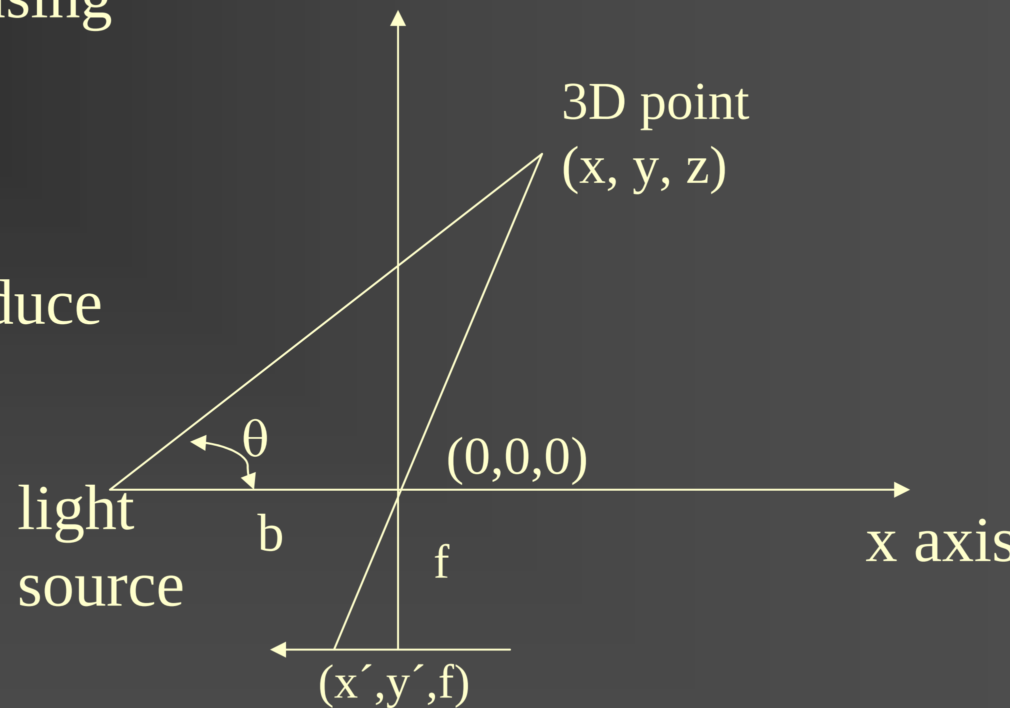
- a single camera
- a light source that can produce stripe(s) on the 3D object



Structured Light
3D Computation

3D data can also be derived using

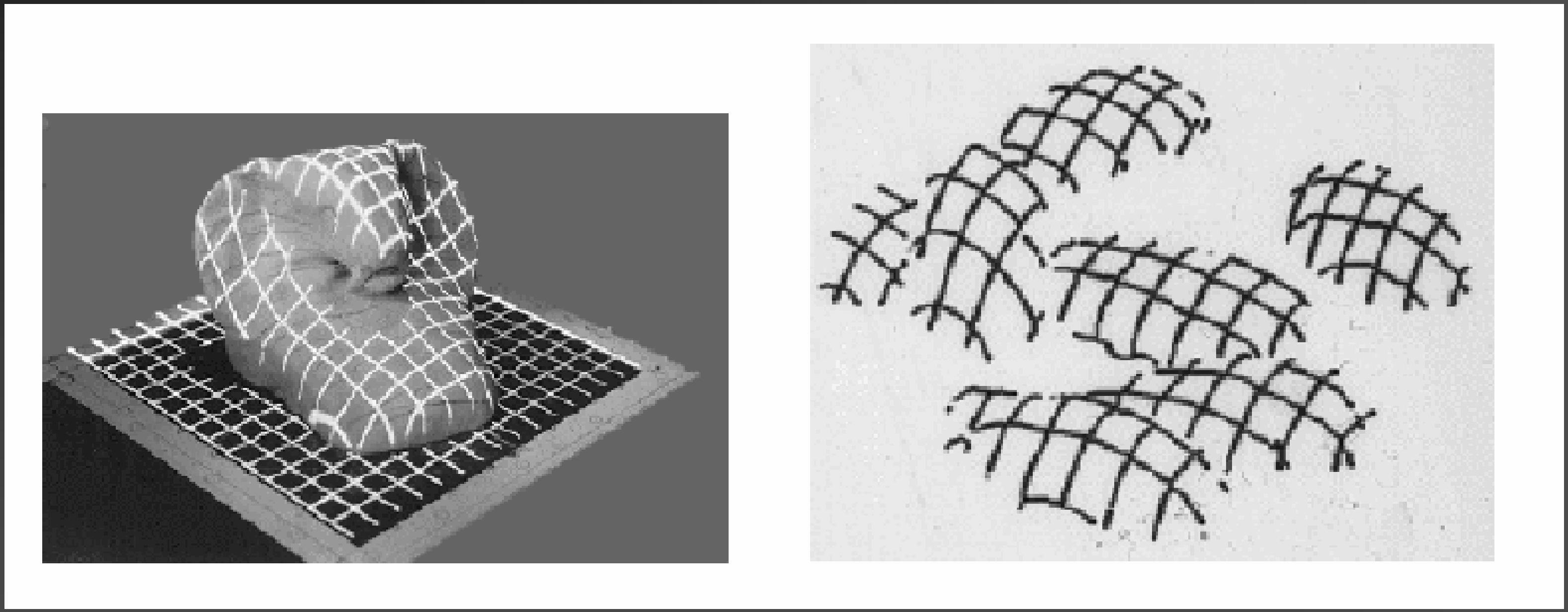
- a single camera
- a light source that can produce stripe(s) on the 3D object



$$[x\ y\ z] = \frac{b}{f \cot \theta - x'} [x'\ y'\ f]$$

3D image

Depth from Multiple Light Stripes



What are these objects?

Our (former) System
4-camera light-striping stereo