Introduction to 3D Vision

- How do we obtain 3D image data?
- What can we do with it?

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?

3D Shape from X

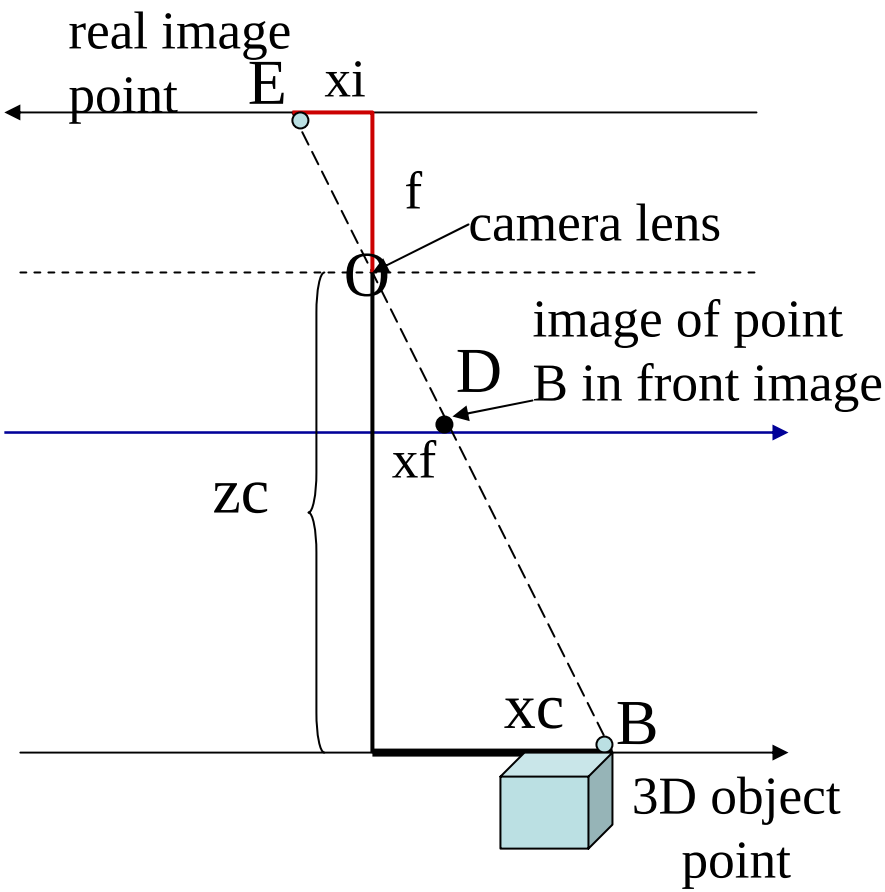
- shading
- silhouette
- texture

mainly research

- stereo
- light striping
- motion

used in practice

Perspective Imaging Model: 1D



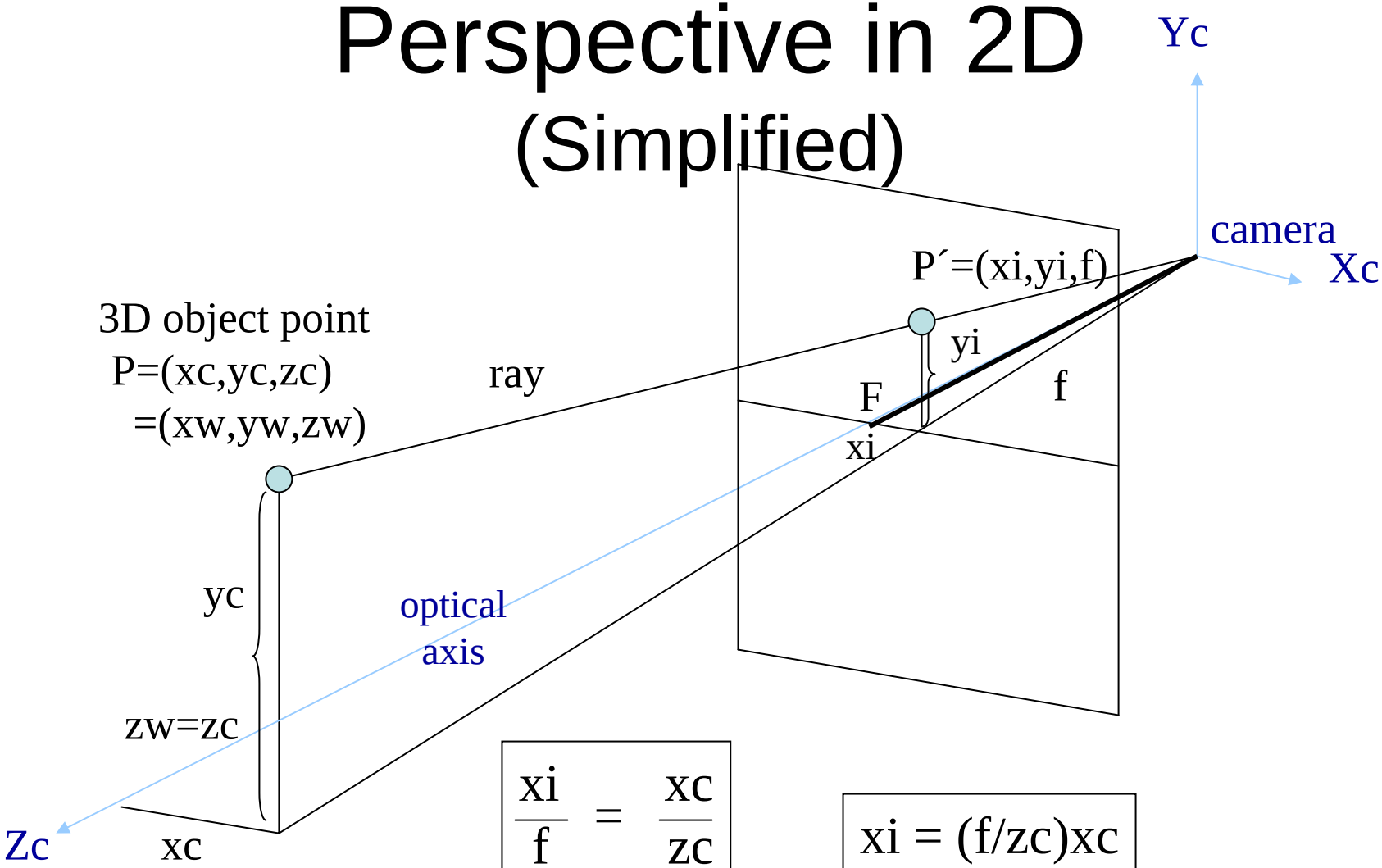
This is the axis of the real image plane.

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)



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

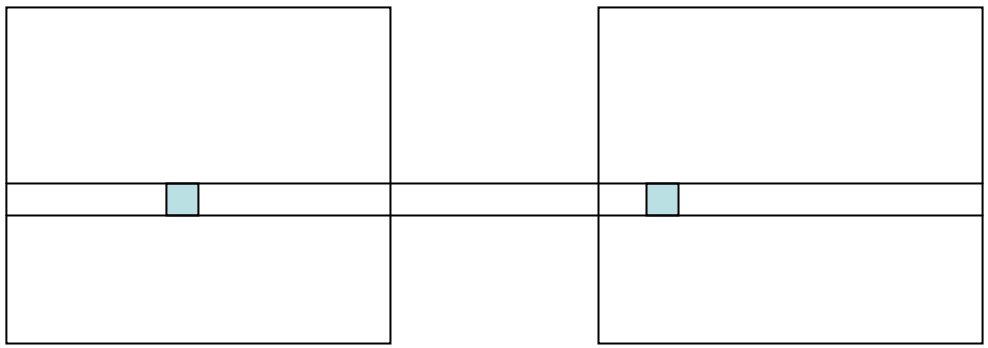
$$xi = (f/zc)xc$$
$$yi = (f/zc)yc$$

Here camera coordinates equal world coordinates.

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

3D from Stereo

- 3D point



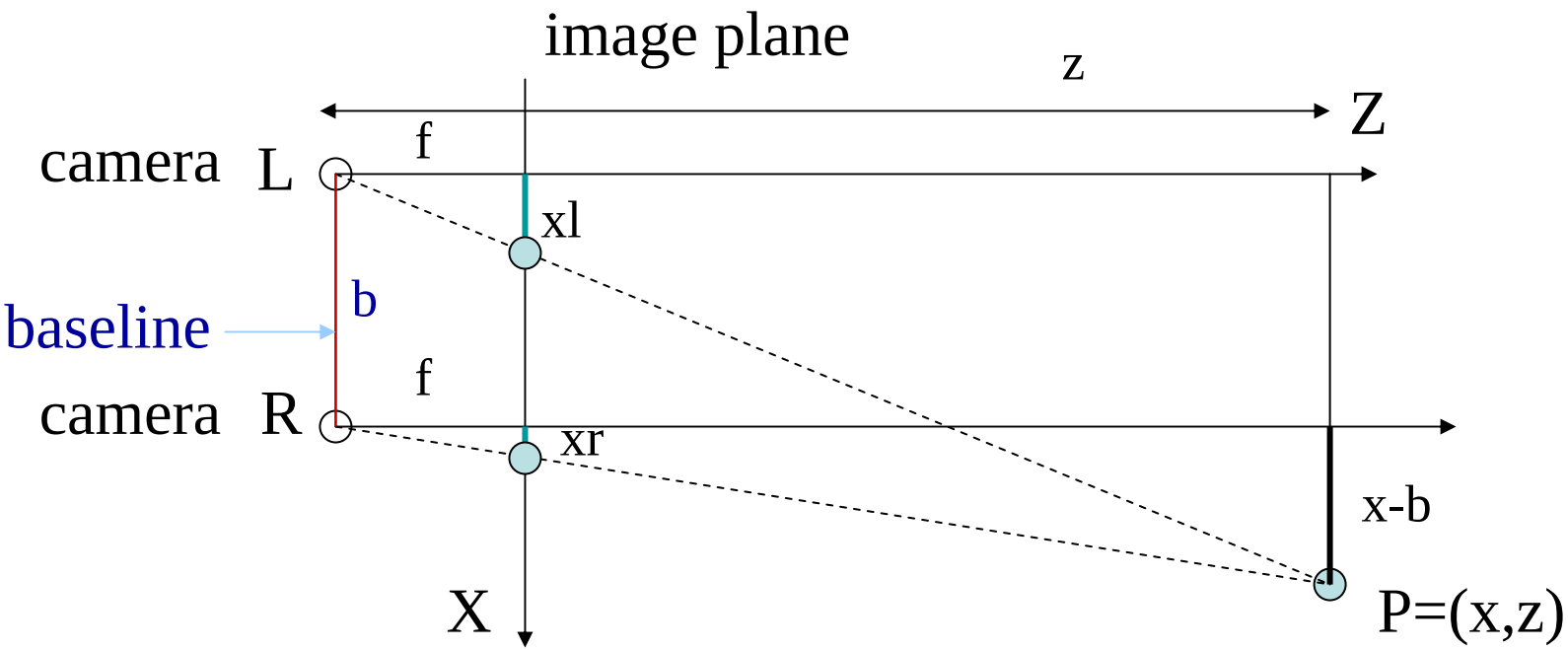
left image right image

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

$$d = x_{left} - x_{right}$$

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:

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

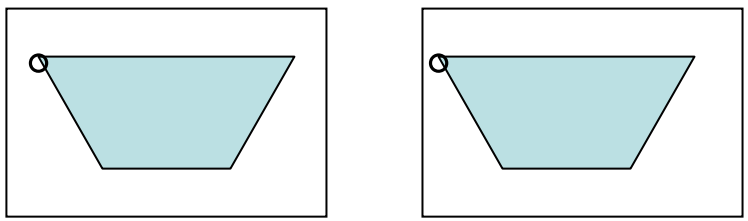
$$x = xl*z/f \quad \text{or} \quad b + xr*z/f$$

$$y = yl*z/f \quad \text{or} \quad yr*z/f$$

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

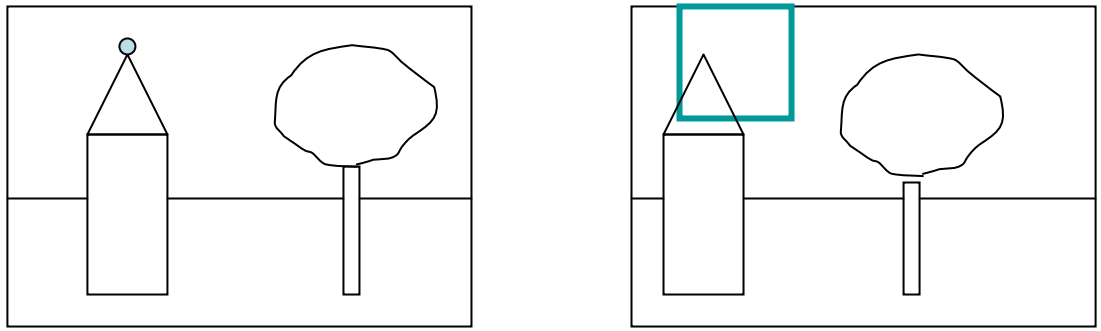
Finding Correspondences

- 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.



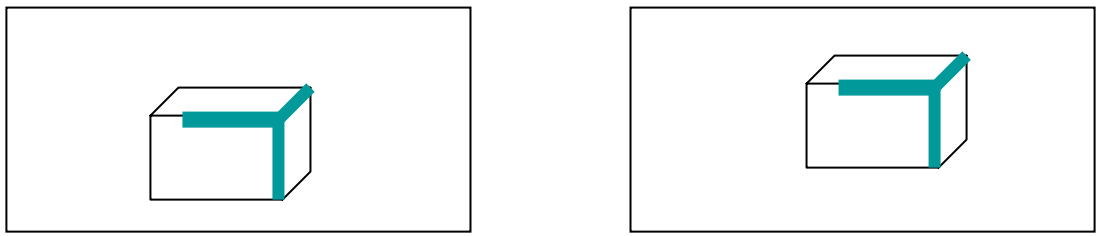
3 Main Matching Methods

1. Cross correlation using small windows.



dense

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



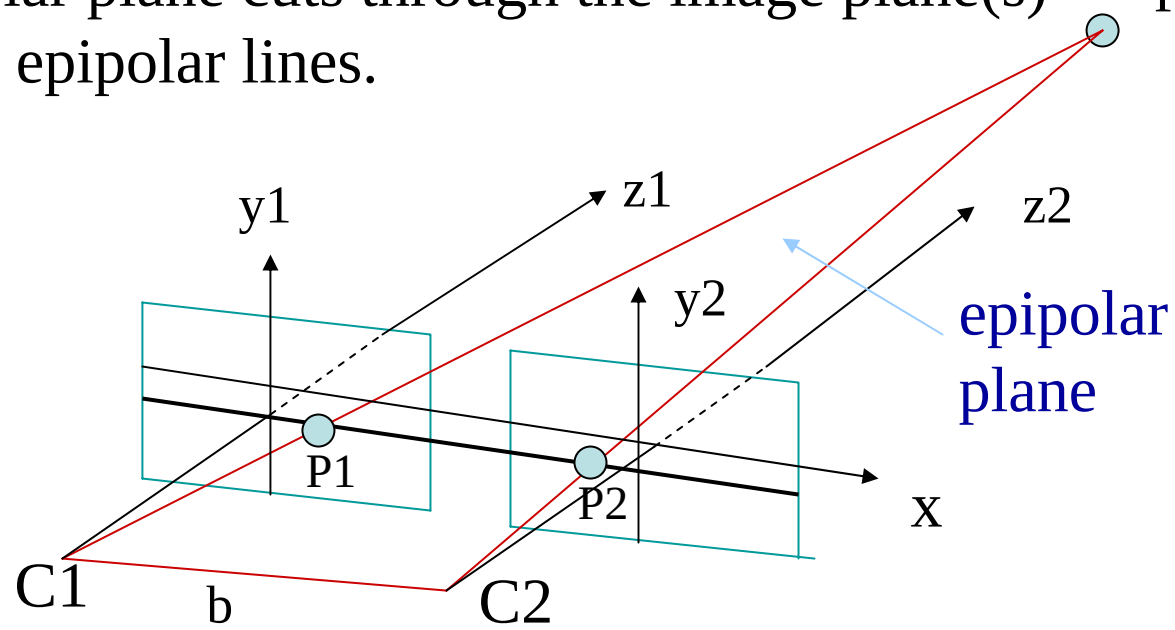
sparse

3. Use the newer interest operators, ie. SIFT.

sparse

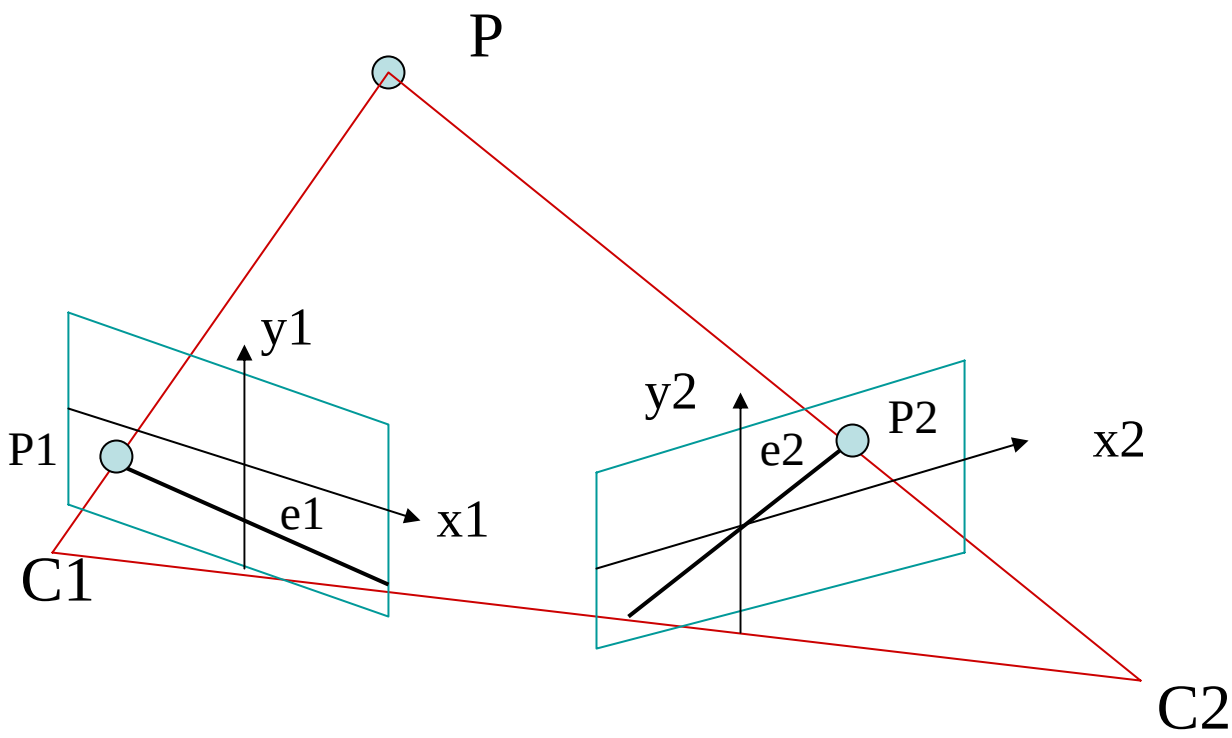
Epipolar Geometry Constraint: 1. Normal Pair of Images

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.

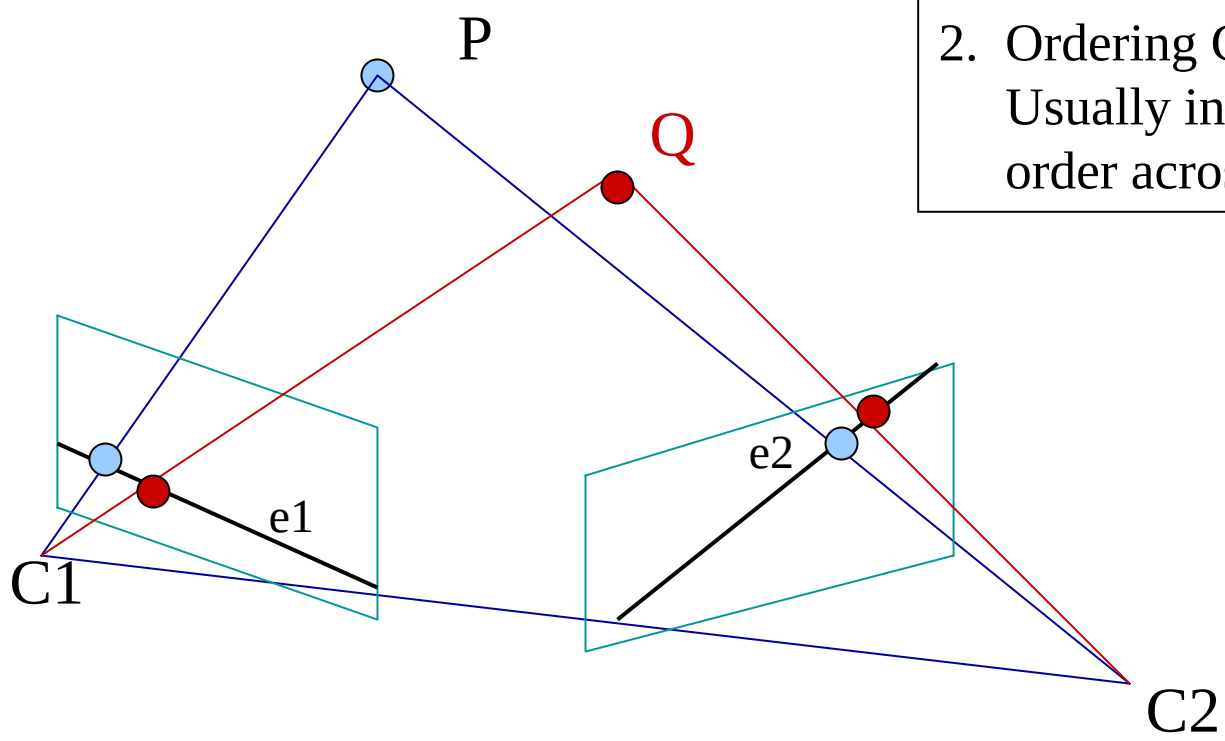
Epipolar Geometry: General Case



Constraints

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

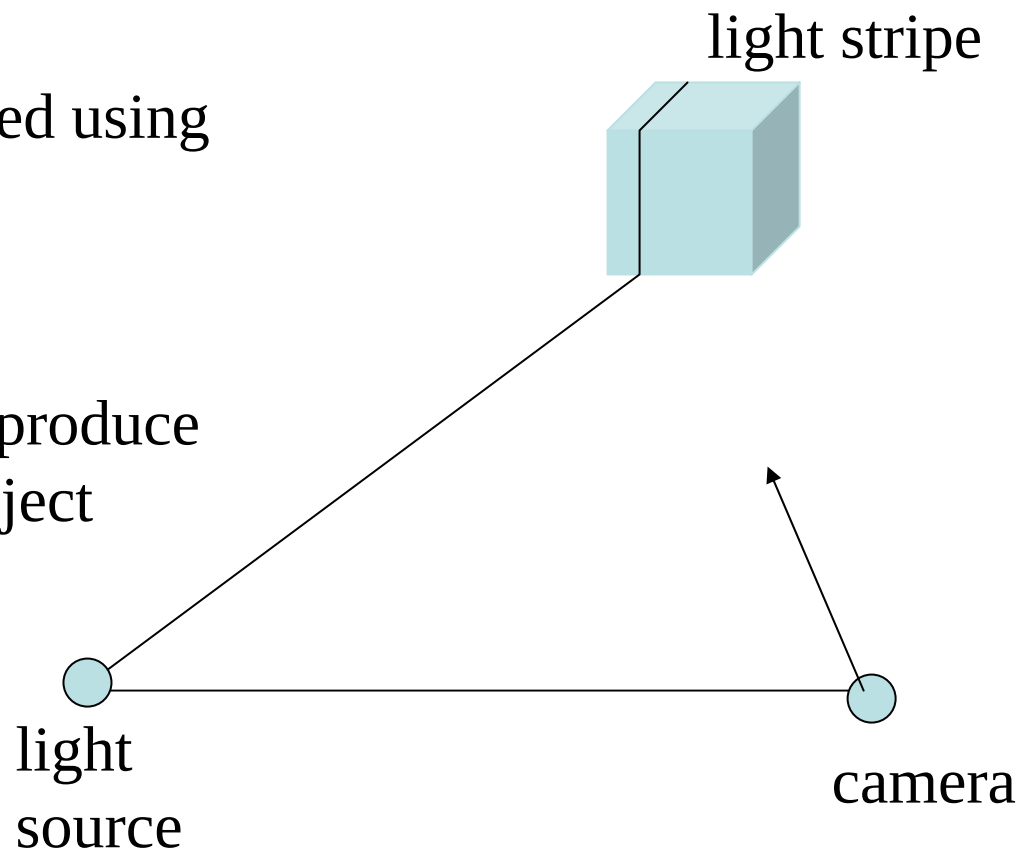
2. Ordering Constraint: Usually in the same order across the lines.



Structured Light

3D data can also be derived using

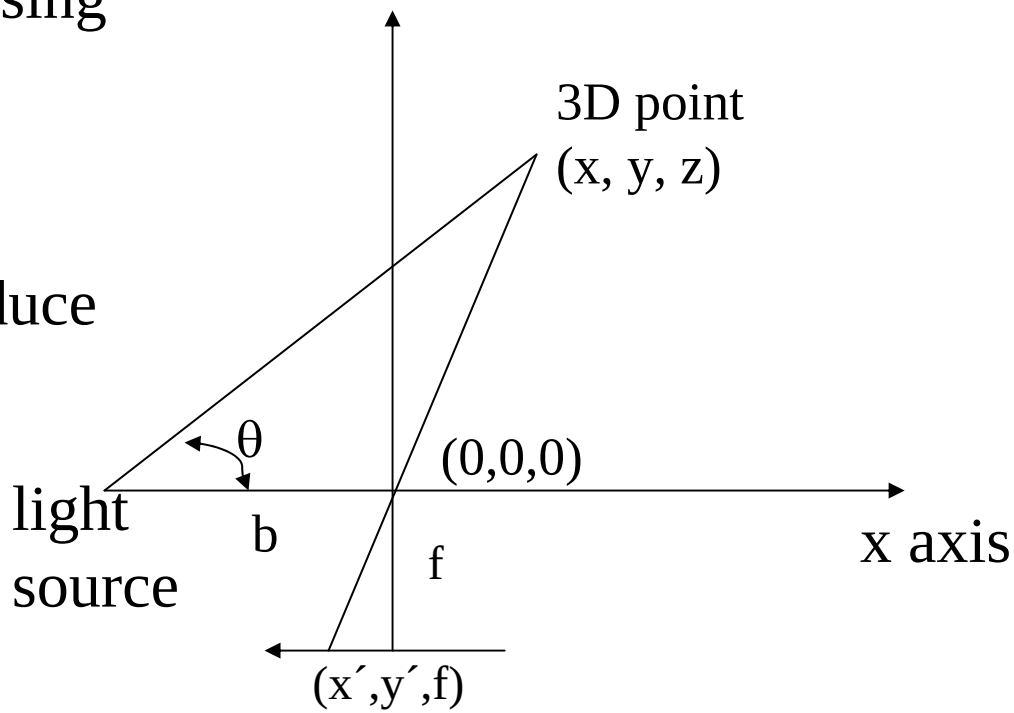
- a single camera
- a light source that can produce stripe(s) on the 3D object



Structured Light
3D Computation

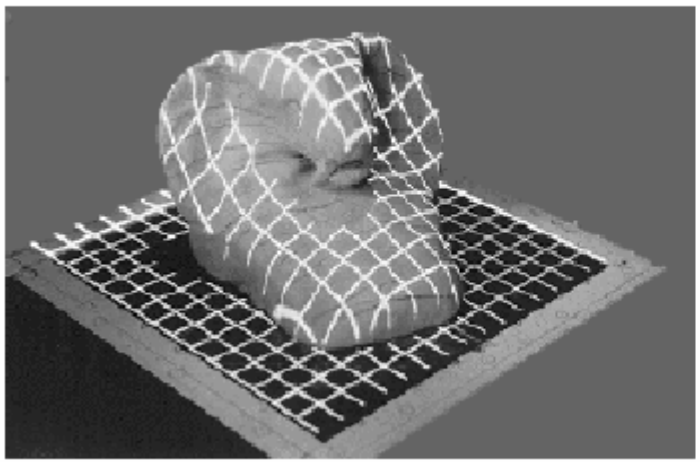
3D data can also be derived using

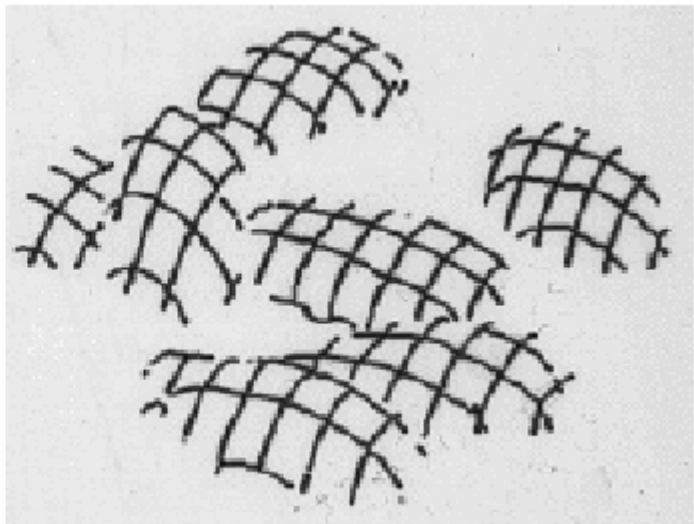
- a single camera
- a light source that can produce stripe(s) on the 3D object



$$[x\ y\ z]_{3D} = \frac{b}{f \cot\theta - x'} [x'\ y'\ f]_{image}$$

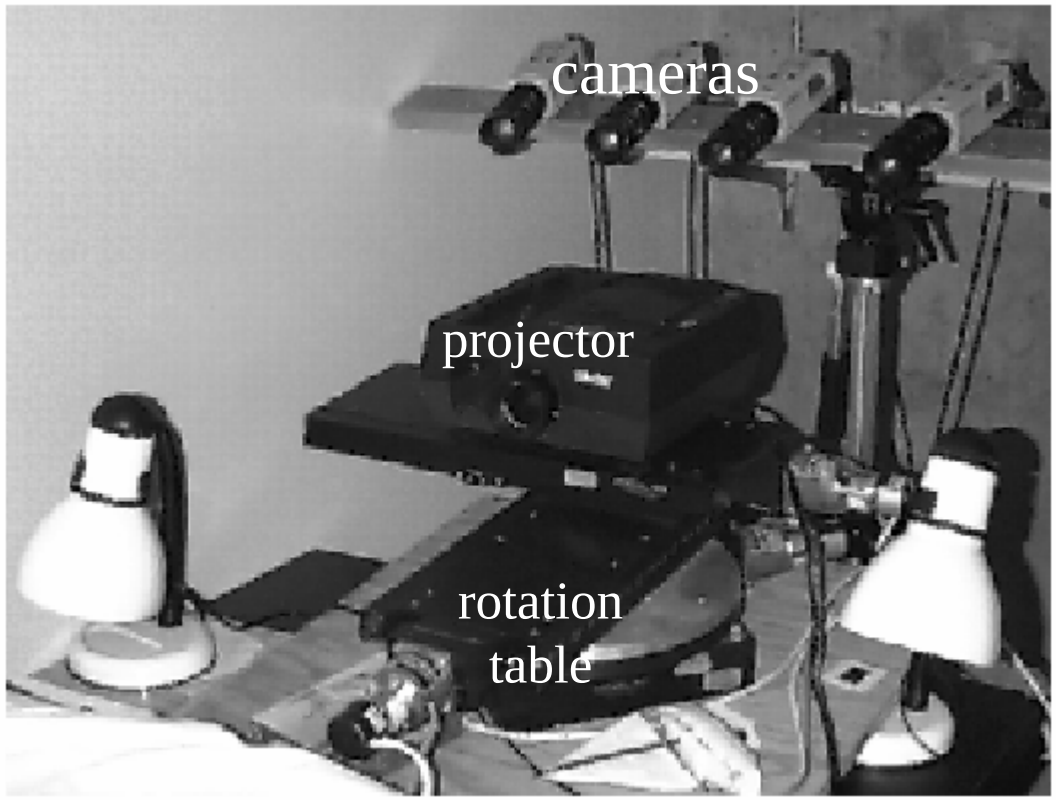
Depth from Multiple Light Stripes

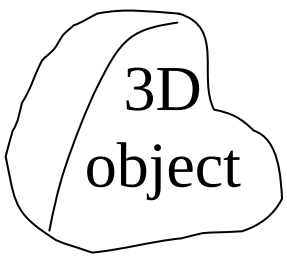




What are these objects?

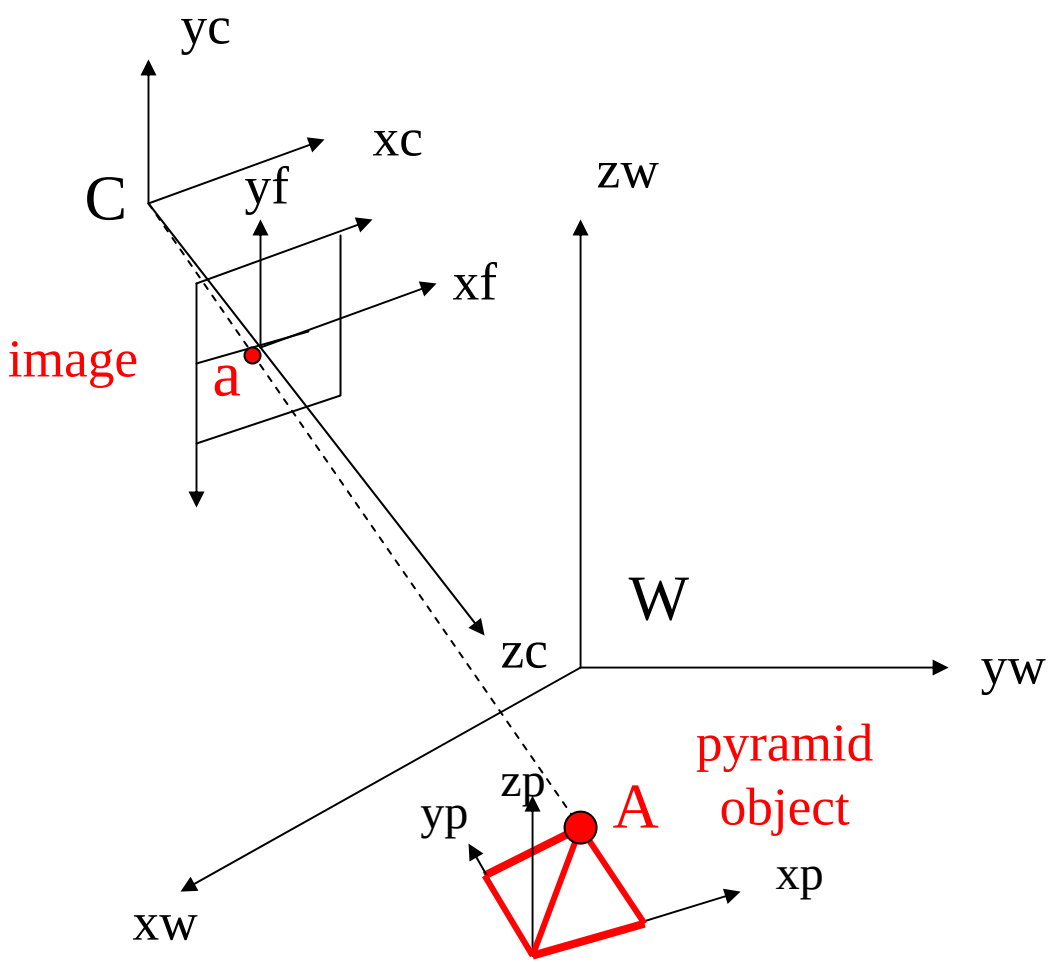
Our (former) System
4-camera light-striping stereo



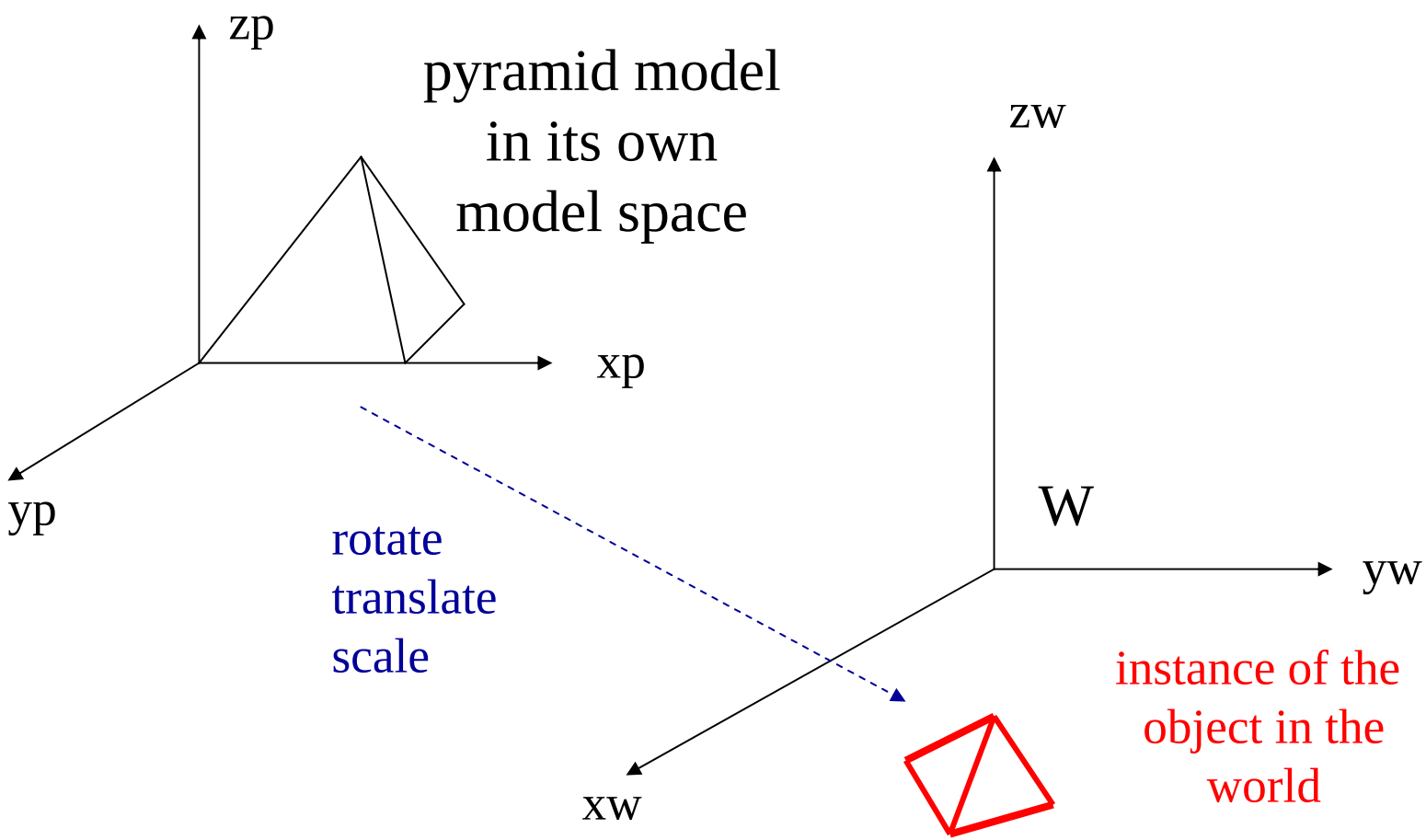


Camera Model: Recall there are 5 Different Frames of Reference

- **Object**
- **World**
- **Camera**
- **Real Image**
- **Pixel Image**



Rigid Body Transformations in 3D

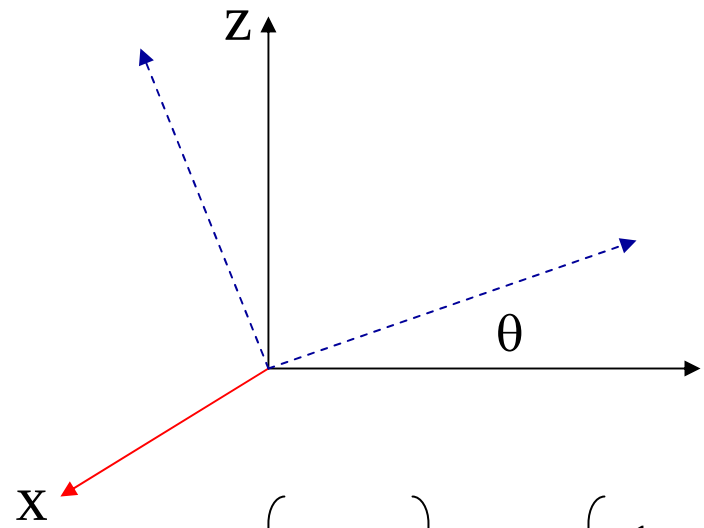


Translation and Scaling in 3D

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} {}^2P_x \\ {}^2P_y \\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x\ {}^2P_x \\ s_y\ {}^2P_y \\ s_z\ {}^2P_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1P_x \\ {}^1P_y \\ {}^1P_z \\ 1 \end{bmatrix}$$

Rotation in 3D is about an axis

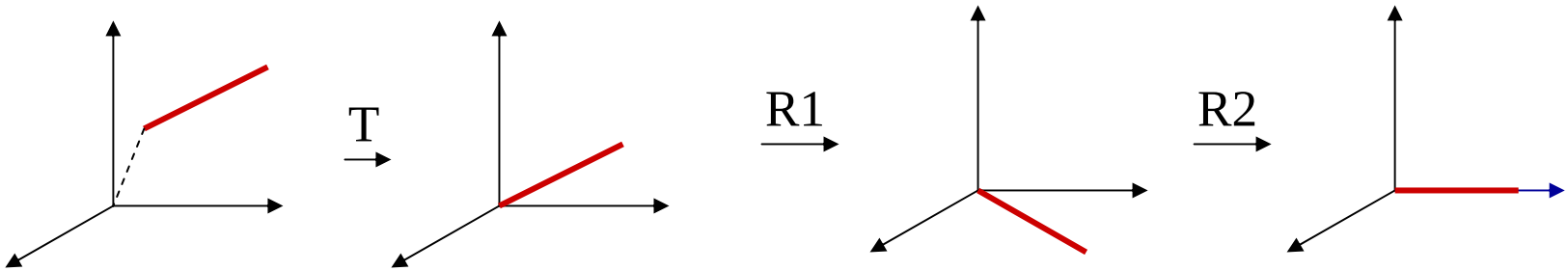


rotation by angle θ

about the x axis

$$\begin{pmatrix} Px' \\ Py' \\ Pz' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Px \\ Py \\ Pz \\ 1 \end{pmatrix}$$

Rotation about Arbitrary Axis



One translation and two rotations to line it up with a major axis. Now rotate it about that axis. Then apply the reverse transformations (R2, R1, T) to move it back.

$$\begin{pmatrix} Px' \\ Py' \\ Pz' \\ 1 \end{pmatrix} = \begin{pmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Px \\ Py \\ Pz \\ 1 \end{pmatrix}$$

The Camera Model

How do we get an image point IP from a world point P?

$$\begin{pmatrix} s\,Ipr \\ s\,Ipc \\ s \end{pmatrix} = \begin{pmatrix} c11 & c12 & c13 & c14 \\ c21 & c22 & c23 & c24 \\ c31 & c32 & c33 & 1 \end{pmatrix} \begin{pmatrix} Px \\ Py \\ Pz \\ 1 \end{pmatrix}$$

image point

camera matrix C

world point

What's in C?

The camera model handles the rigid body transformation from world coordinates to camera coordinates plus the perspective transformation to image coordinates.

1. $CP = T\,R\,WP$
2. $IP = \pi(f)\,CP$

$$\begin{pmatrix} s\,Ipx \\ s\,Ipy \\ s \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} CPx \\ CPy \\ CPz \\ 1 \end{pmatrix}$$

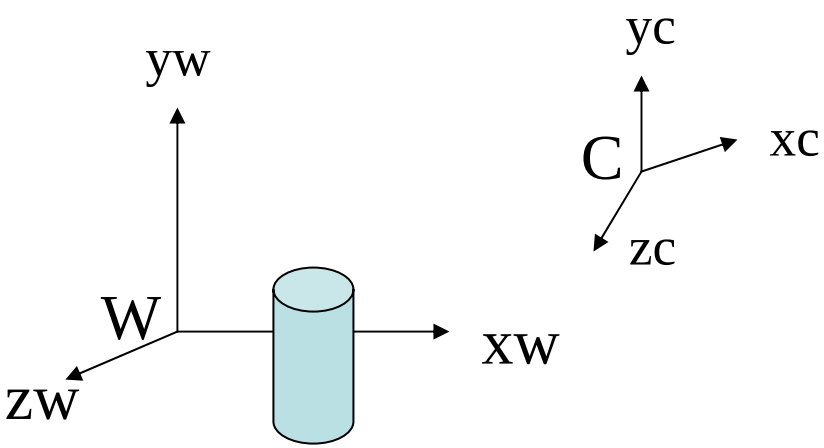
image point

perspective transformation

3D point in camera coordinates

Camera Calibration

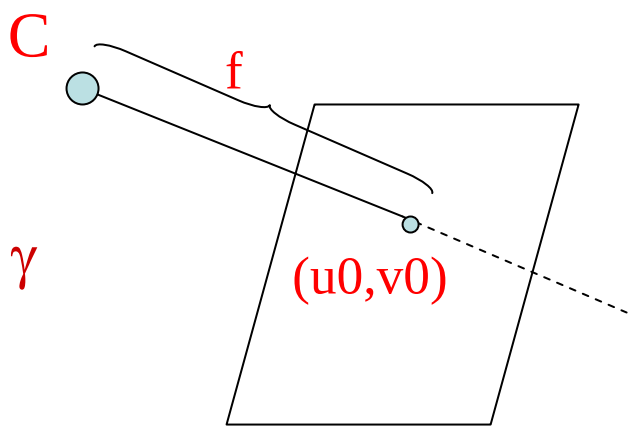
- In order work in 3D, we need to know the parameters of the particular camera setup.
- Solving for the camera parameters is called calibration.



- intrinsic parameters are of the camera device
- extrinsic parameters are where the camera sits in the world

Intrinsic Parameters

- principal point (u0,v0)
- scale factors (dx,dy)
- aspect ratio distortion factor γ
- focal length f
- lens distortion factor κ
  (models radial lens distortion)



Extrinsic Parameters

- translation parameters
  t = [tx ty tz]

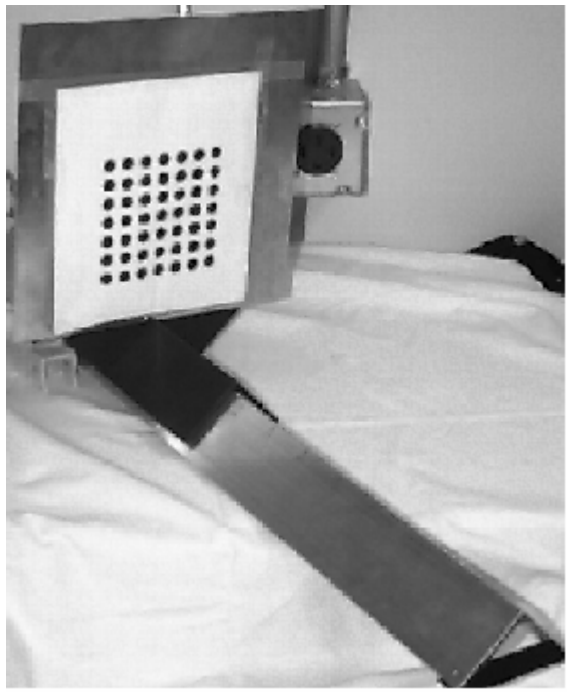
- rotation matrix

$$R = \begin{pmatrix} r11 & r12 & r13 & 0 \\ r21 & r22 & r23 & 0 \\ r31 & r32 & r33 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Are there really nine parameters?

Calibration Object

The idea is to snap images at different depths and get a lot of 2D-3D point correspondences.



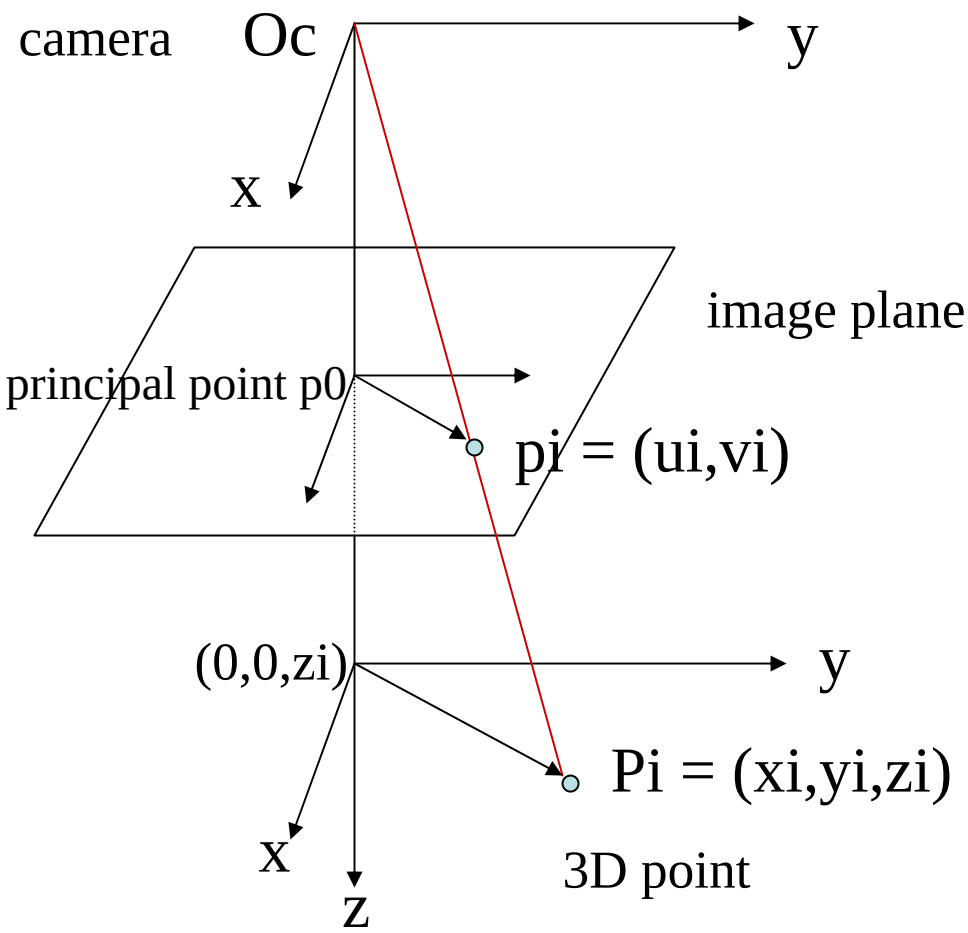
The Tsai Procedure

- The Tsai procedure was developed by Roger Tsai at IBM Research and is most widely used.
- Several images are taken of the calibration object yielding point correspondences at different distances.
- Tsai's algorithm requires n > 5 correspondences

$$\{(x_i, y_i, z_i), (u_i, v_i)) \mid i = 1,\ldots,n\}$$

between (real) image points and 3D points.

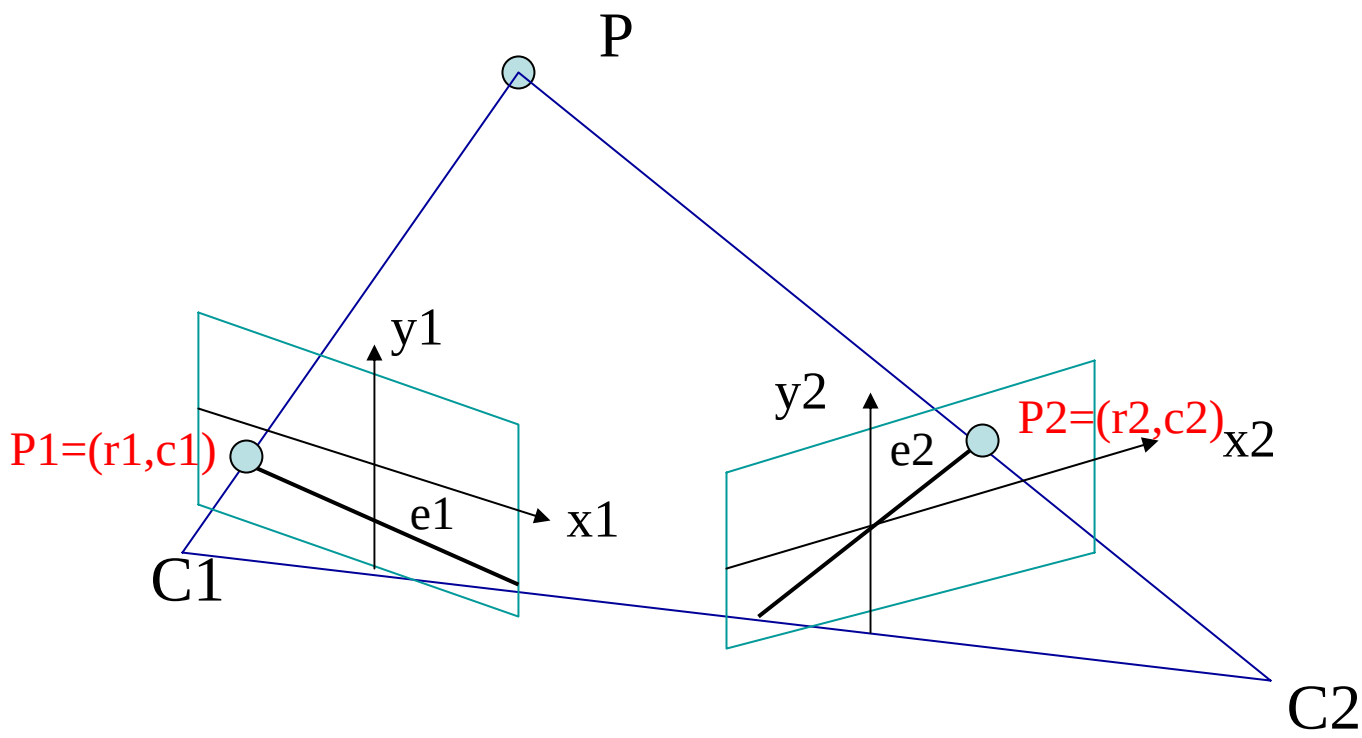
Tsai's Geometric Setup



Tsai's Procedure

- Given n point correspondences ((xi,yi,zi), (ui,vi))
- Estimates
  - 9 rotation matrix values
  - 3 translation matrix values
  - focal length
  - lens distortion factor
- By solving several systems of equations

We use them for general stereo.



For a correspondence (r1,c1) in image 1 to (r2,c2) in image 2:

1. Both cameras were calibrated. Both camera matrices are then known. From the two camera equations we get

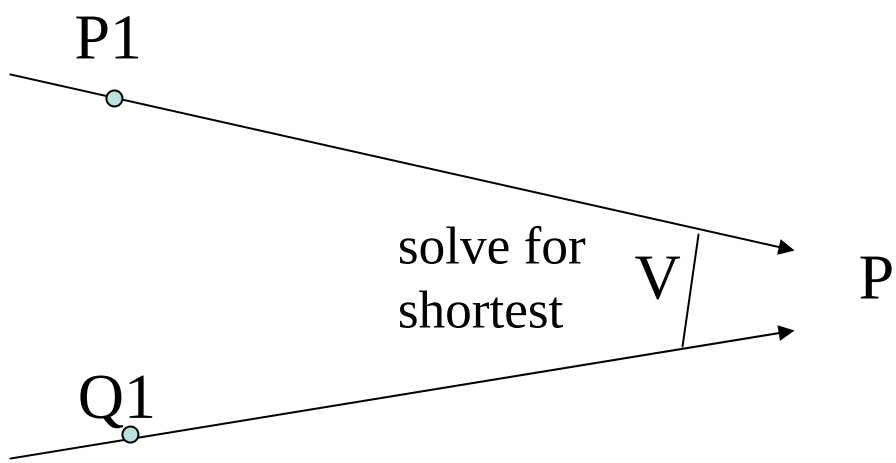
   4 linear equations in 3 unknowns.

$$r1 = (b11 - b31*r1)\mathbf{x} + (b12 - b32*r1)\mathbf{y} + (b13-b33*r1)\mathbf{z}$$
$$c1 = (b21 - b31*c1)\mathbf{x} + (b22 - b32*c1)\mathbf{y} + (b23-b33*c1)\mathbf{z}$$

$$r2 = (c11 - c31*r2)\mathbf{x} + (c12 - c32*r2)\mathbf{y} + (c13 - c33*r2)\mathbf{z}$$
$$c2 = (c21 - c31*c2)\mathbf{x} + (c22 - c32*c2)\mathbf{y} + (c23 - c33*c2)\mathbf{z}$$

Direct solution uses 3 equations, won't give reliable results.

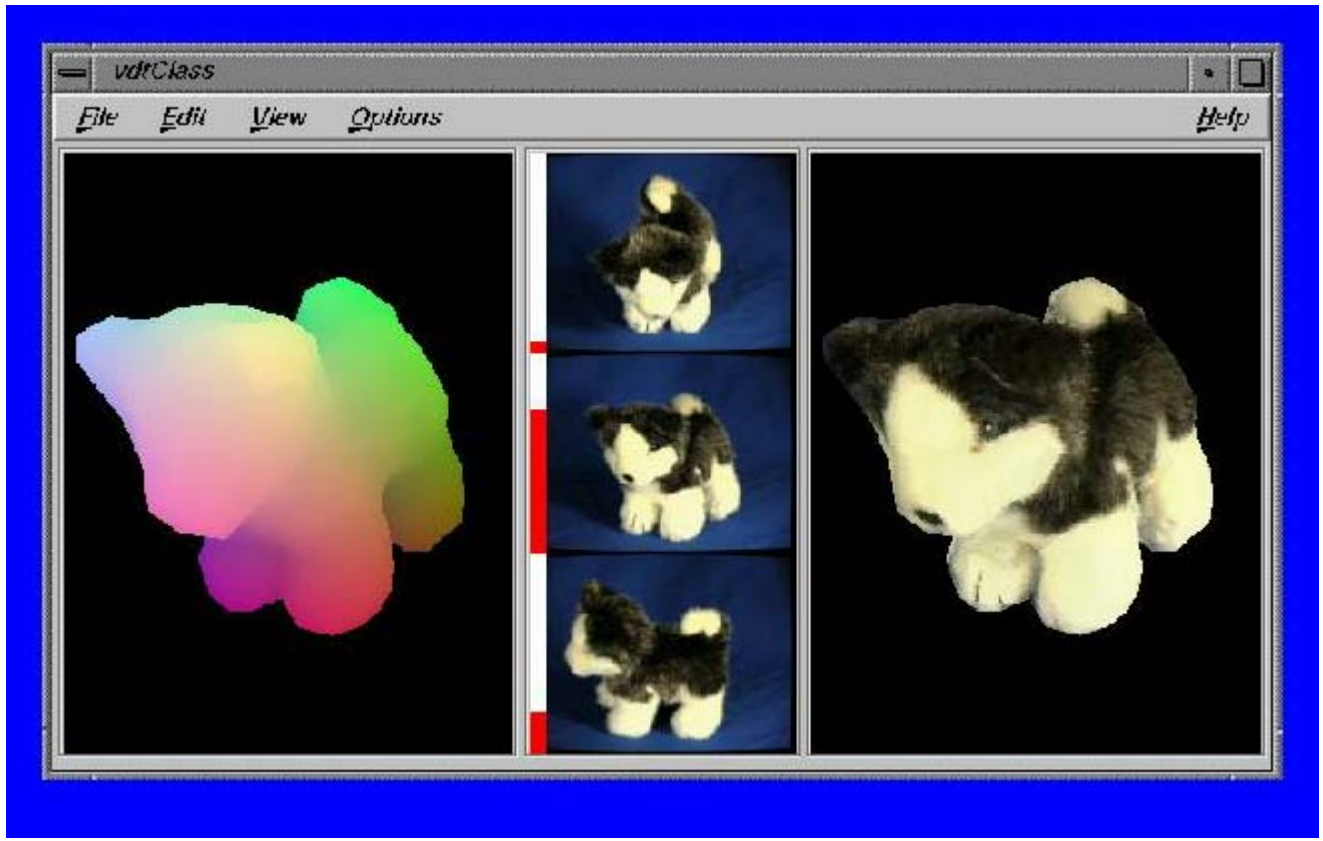
Solve by computing the closest approach of the two skew rays.



If the rays intersected perfectly in 3D, the intersection would be P.

Instead, we solve for the shortest line segment connecting the two rays and let P be its midpoint.

Application: Kari Pulli's Reconstruction of 3D Objects from light-striping stereo.



Application: Zhenrong Qian's 3D Blood Vessel Reconstruction from Visible Human Data