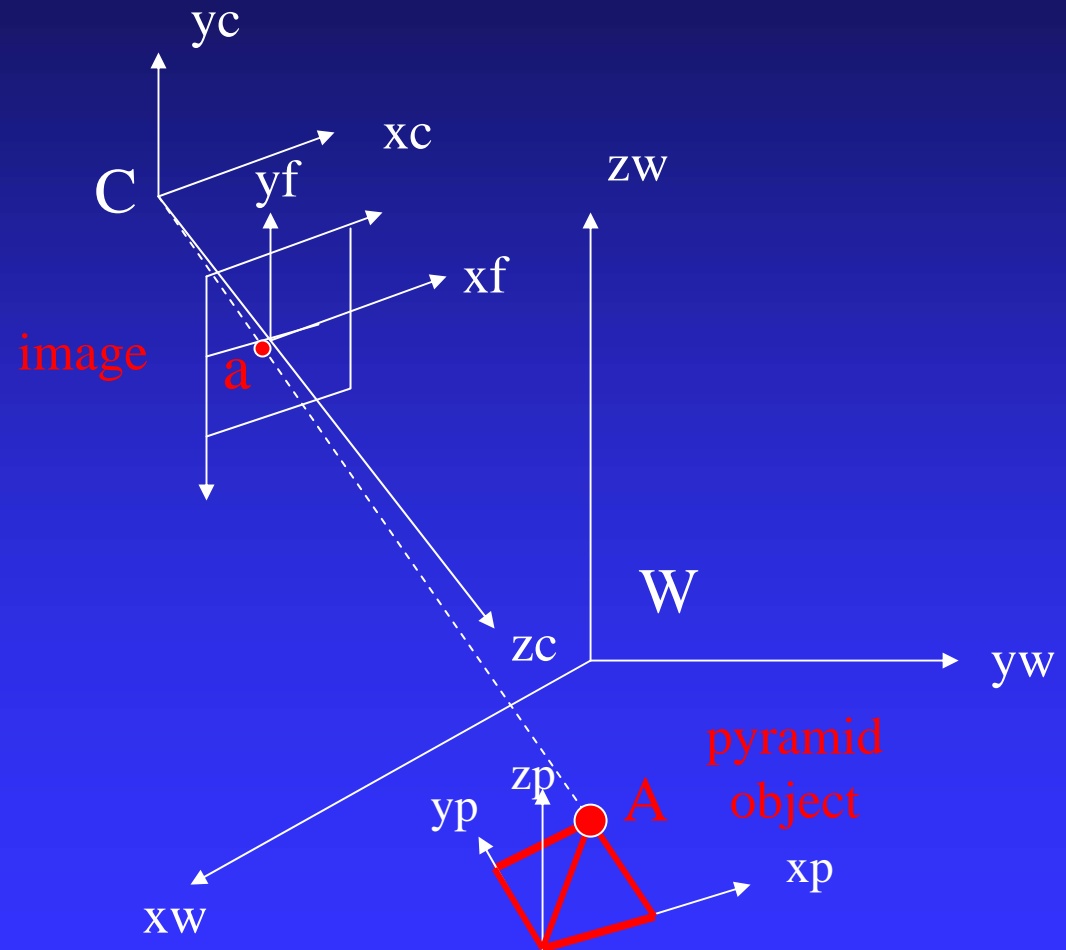


3D Sensing

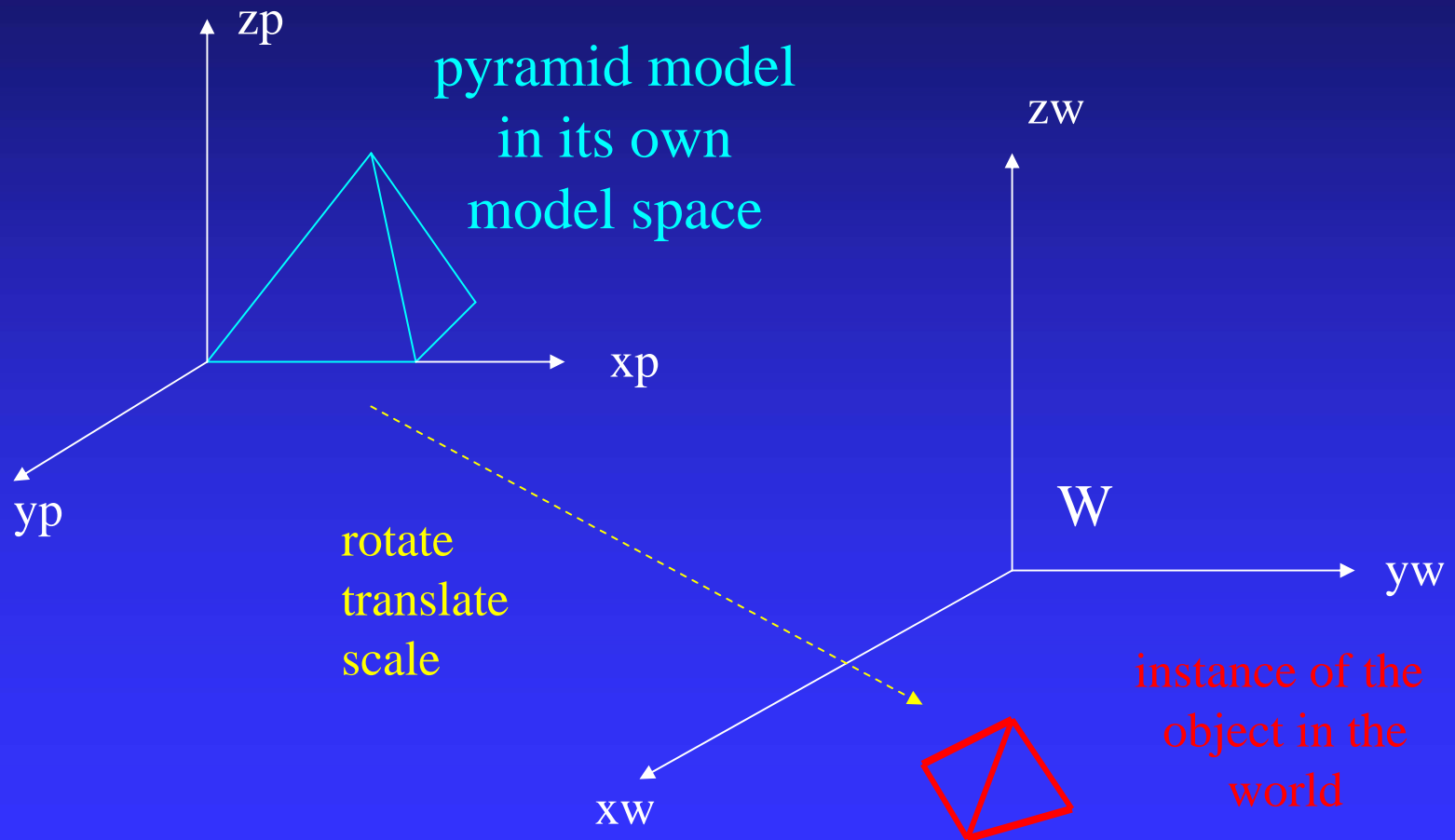
- Camera Model and 3D Transformations
- Camera Calibration (Tsai's Method)
- Depth from General Stereo (overview)
- Pose Estimation from 2D Images (skip)
- 3D Reconstruction

Camera Model: Recall there are 5 Different Frames of Reference

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



Rigid Body Transformations in 3D

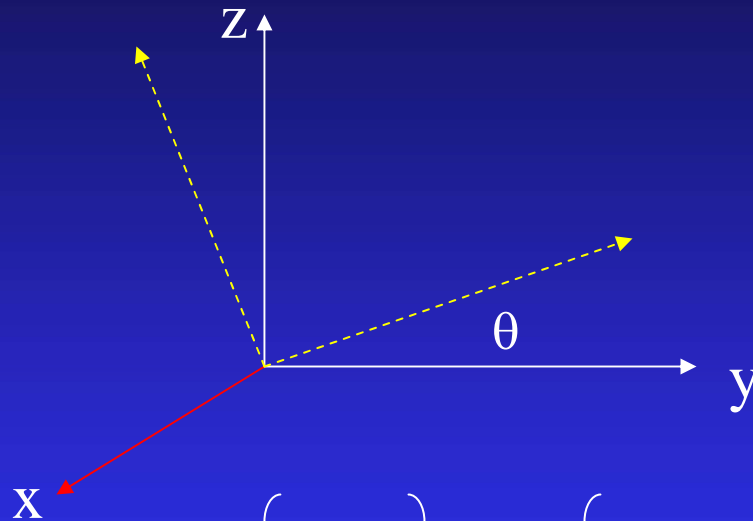


Translation and Scaling in 3D

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

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

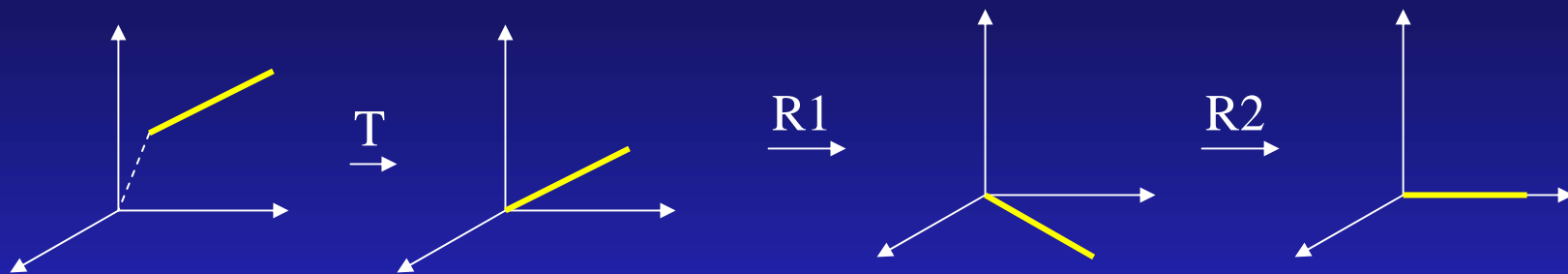
Rotation in 3D is about an axis



rotation by angle θ
about the x axis

$$\begin{pmatrix} P_{x'} \\ P_{y'} \\ P_{z'} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

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.

$$\begin{pmatrix} P_x' \\ P_y' \\ P_z' \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

The Camera Model

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

$$\begin{pmatrix} s \text{ Ipr} \\ s \text{ Ipc} \\ s \end{pmatrix} = \begin{pmatrix} c11 & c12 & c13 & c14 \\ c21 & c22 & c23 & c24 \\ c31 & c32 & c33 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

image
point

camera matrix C

world
point

What's in C?

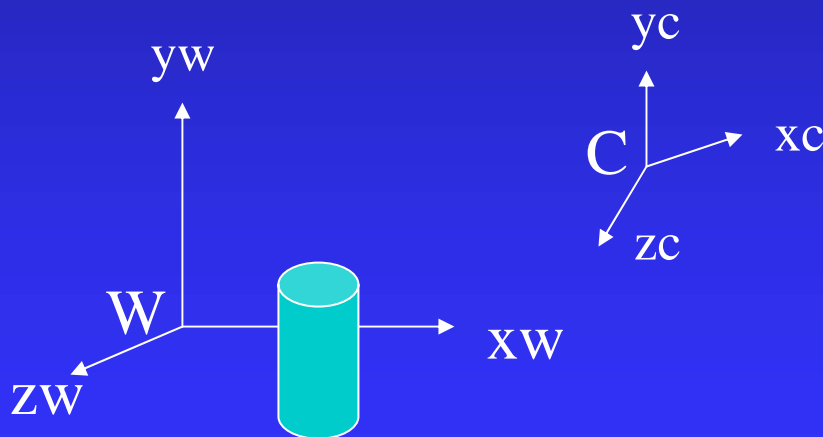
The camera model handles the **rigid body** transformation from world coordinates to camera coordinates plus the **perspective** transformation to image coordinates.

$$\begin{array}{l}
 1. \quad CP = TR WP \\
 2. \quad IP = \pi(f) CP
 \end{array}$$

$$\begin{array}{ccc}
 \begin{pmatrix} s \text{ Ipx} \\ s \text{ Ipy} \\ s \end{pmatrix} = & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} & \begin{pmatrix} \text{CPx} \\ \text{CPy} \\ \text{CPz} \\ 1 \end{pmatrix} \\
 \text{image point} & \text{perspective transformation} & \text{3D point in camera coordinates}
 \end{array}$$

Camera Calibration

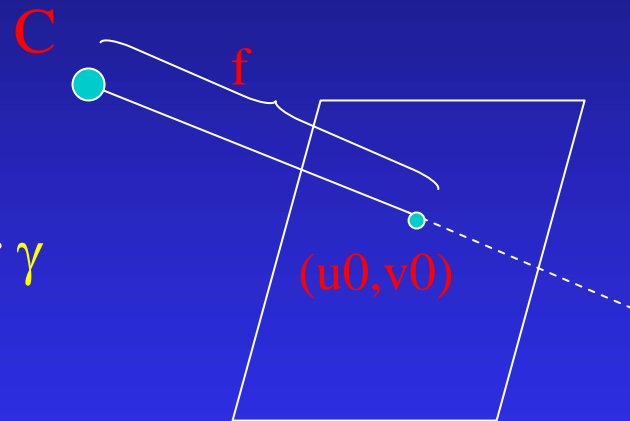
- In order work in 3D, we need to know the parameters of the particular camera setup.
- Solving for the camera parameters is called calibration.



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

Intrinsic Parameters

- principal point (u_0, v_0)
- scale factors (dx, dy)
- aspect ratio distortion factor γ
- focal length f
- lens distortion factor κ
(models radial lens distortion)



Extrinsic Parameters

- translation parameters

$$t = [t_x \ t_y \ t_z]$$

- rotation matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

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

between (real) image points and 3D points.

In this* version of Tsai's algorithm,

- The real-valued (u,v) are computed from their pixel positions (r,c) :

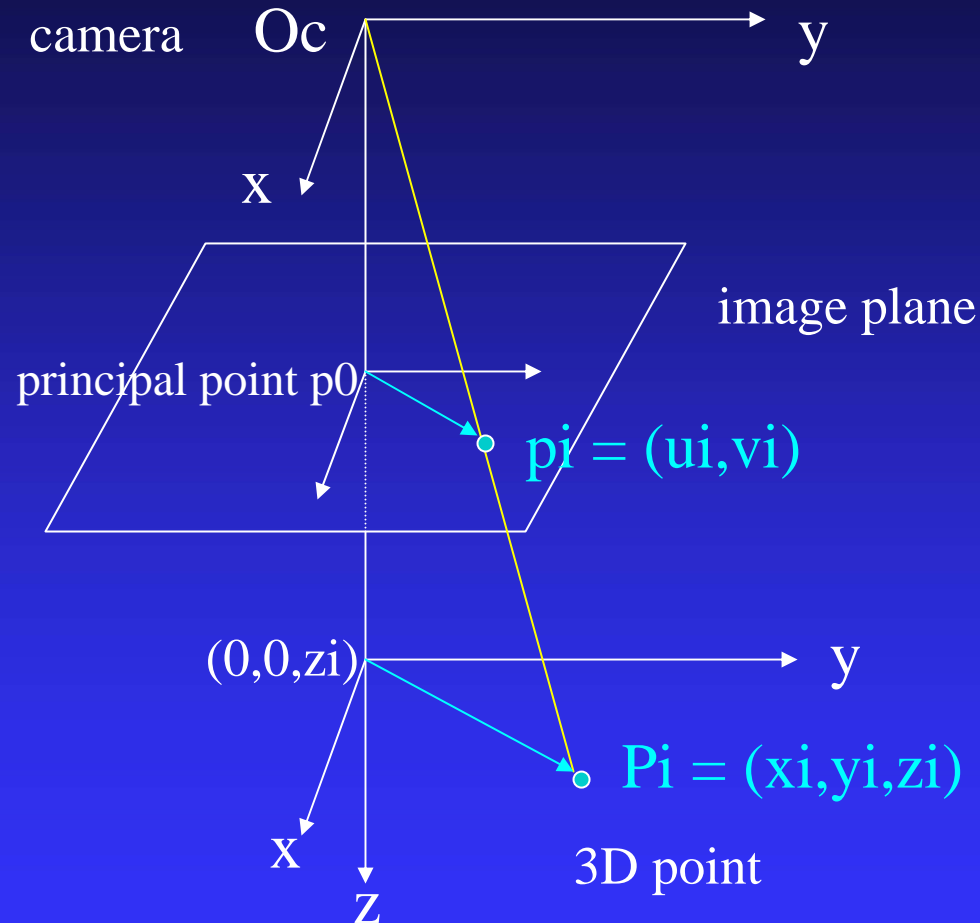
$$u = \gamma dx (c - u_0) \quad v = -dy (r - v_0)$$

where

- (u_0, v_0) is the **center of the image**
- dx and dy are the **center-to-center (real) distances** between pixels and come from the camera's specs
- γ is a **scale factor** learned from previous trials

* This version is for single-plane calibration.

Tsai's Geometric Setup



Tsai's Procedure

1. Given the n point correspondences $((x_i, y_i, z_i), (u_i, v_i))$

Compute matrix A with rows a_i

$$a_i = (v_i * x_i, v_i * y_i, -u_i * x_i, -u_i * v_i, v_i)$$

These are known quantities which will be used to solve for intermediate values, which will then be used to solve for the parameters sought.

Intermediate Unknowns

2. The vector of **unknowns** is $\mu = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$:

$$\mu_1=r_{11}/t_y \quad \mu_2=r_{12}/t_y \quad \mu_3=r_{21}/t_y \quad \mu_4=r_{22}/t_y \quad \mu_5=t_x/t_y$$

where the **r's** and **t's** are unknown rotation and translation parameters.

3. Let vector $\mathbf{b} = (u_1, u_2, \dots, u_n)$ contain the **u image coordinates**.

4. **Solve** the system of linear equations

$$\mathbf{A} \mu = \mathbf{b}$$

for unknown parameter vector μ .

Use μ to solve for t_y , t_x , and 4 rotation parameters

5. Let $U = \mu_1^2 + \mu_2^2 + \mu_3^2 + \mu_4^2$. Use U to calculate t_y^2 . (see text)

$$t_y^2 = \begin{cases} \frac{U - [U^2 - 4(\mu_1\mu_4 - \mu_2\mu_3)^2]^{1/2}}{2(\mu_1\mu_4 - \mu_2\mu_3)^2} & \text{if } (\mu_1\mu_4 - \mu_2\mu_3) \neq 0 \\ \frac{1}{\mu_1^2 + \mu_2^2} & \text{if } (\mu_1^2 + \mu_2^2) \neq 0 \\ \frac{1}{\mu_3^2 + \mu_4^2} & \text{if } (\mu_3^2 + \mu_4^2) \neq 0 \end{cases}$$

6. Try the positive square root $ty = (ty^2)^{1/2}$ and use it to compute translation and rotation parameters.

$$\begin{aligned} r_{11} &= \mu_1 ty \\ r_{12} &= \mu_2 ty \\ r_{21} &= \mu_3 ty \\ r_{22} &= \mu_4 ty \\ tx &= \mu_5 ty \end{aligned}$$

Now we know
2 translation parameters and
4 rotation parameters.

except...

Determine true sign of t_y and compute remaining rotation parameters.

7. Select an object point P whose image coordinates (u, v) are far from the image center.
8. Use P 's coordinates and the translation and rotation parameters so far to estimate the image point that corresponds to P .

If its coordinates have the same signs as (u, v) , then keep t_y , else negate it.

9. Use the first 4 rotation parameters to calculate the remaining 5.

Calculating the remaining 5 rotation parameters:

$$r_{13} = (1 - r_{11}^2 - r_{12}^2)^{1/2}$$

$$r_{23} = (1 - r_{21}^2 - r_{22}^2)^{1/2}$$

$$r_{31} = \frac{1 - r_{11}^2 - r_{12}r_{21}}{r_{13}}$$

$$r_{32} = \frac{1 - r_{21}r_{12} - r_{22}^2}{r_{23}}$$

$$r_{33} = (1 - r_{31}r_{13} - r_{32}r_{23})^{1/2}$$

Solve another linear system.

10. We have t_x and t_y and the 9 rotation parameters.
Next step is to find t_z and f .

Form a matrix A' whose rows are:

$$a_i' = (r_{21} * x_i + r_{22} * y_i + t_y, \quad v_i)$$

and a vector b' whose rows are:

$$b_i' = (r_{31} * x_i + r_{32} * y_i) * v_i$$

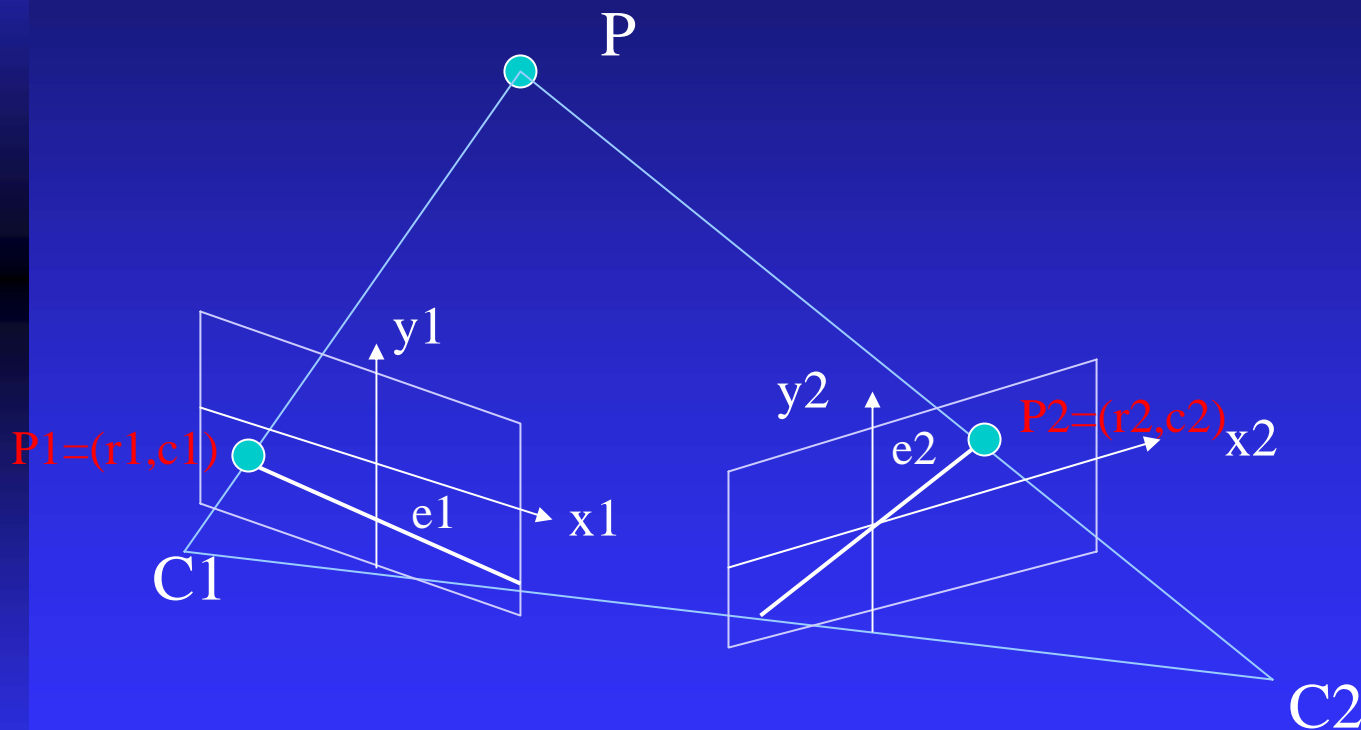
11. Solve $A' * \mathbf{v} = \mathbf{b}'$ for $\mathbf{v} = (f, t_z)$.

Almost there

12. If f is negative, change signs (see text).
13. Compute the lens distortion factor κ and improve the estimates for f and tz by solving a nonlinear system of equations by a **nonlinear regression**.
14. All parameters have been computed.

Use them in 3D data acquisition systems.

We use them for general stereo.



For a correspondence (r_1, c_1) in image 1 to (r_2, c_2) 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.

$$r_1 = (b_{11} - b_{31} * r_1) \mathbf{x} + (b_{12} - b_{32} * r_1) \mathbf{y} + (b_{13} - b_{33} * r_1) \mathbf{z}$$

$$c_1 = (b_{21} - b_{31} * c_1) \mathbf{x} + (b_{22} - b_{32} * c_1) \mathbf{y} + (b_{23} - b_{33} * c_1) \mathbf{z}$$

$$r_2 = (c_{11} - c_{31} * r_2) \mathbf{x} + (c_{12} - c_{32} * r_2) \mathbf{y} + (c_{13} - c_{33} * r_2) \mathbf{z}$$

$$c_2 = (c_{21} - c_{31} * c_2) \mathbf{x} + (c_{22} - c_{32} * c_2) \mathbf{y} + (c_{23} - c_{33} * c_2) \mathbf{z}$$

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

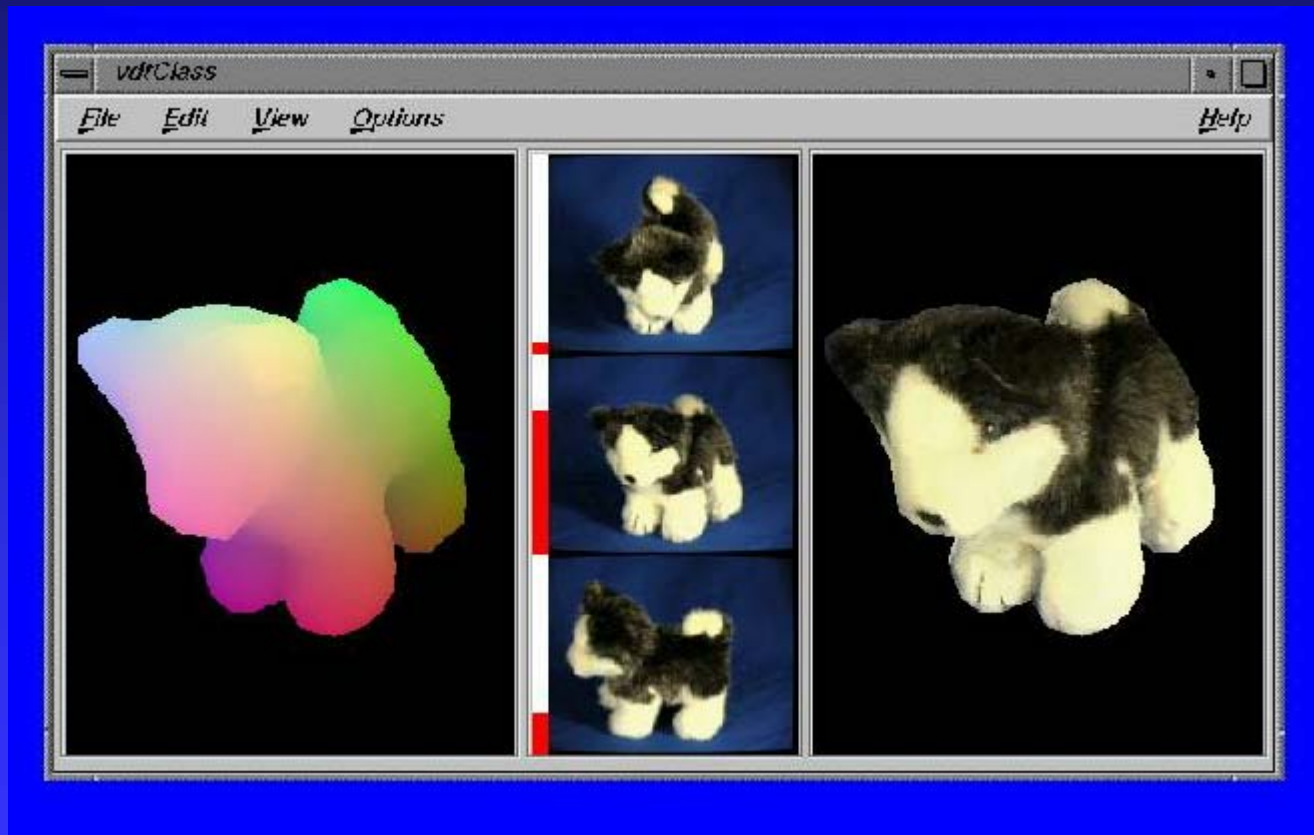
Solve by computing the closest approach of the two skew rays.



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

