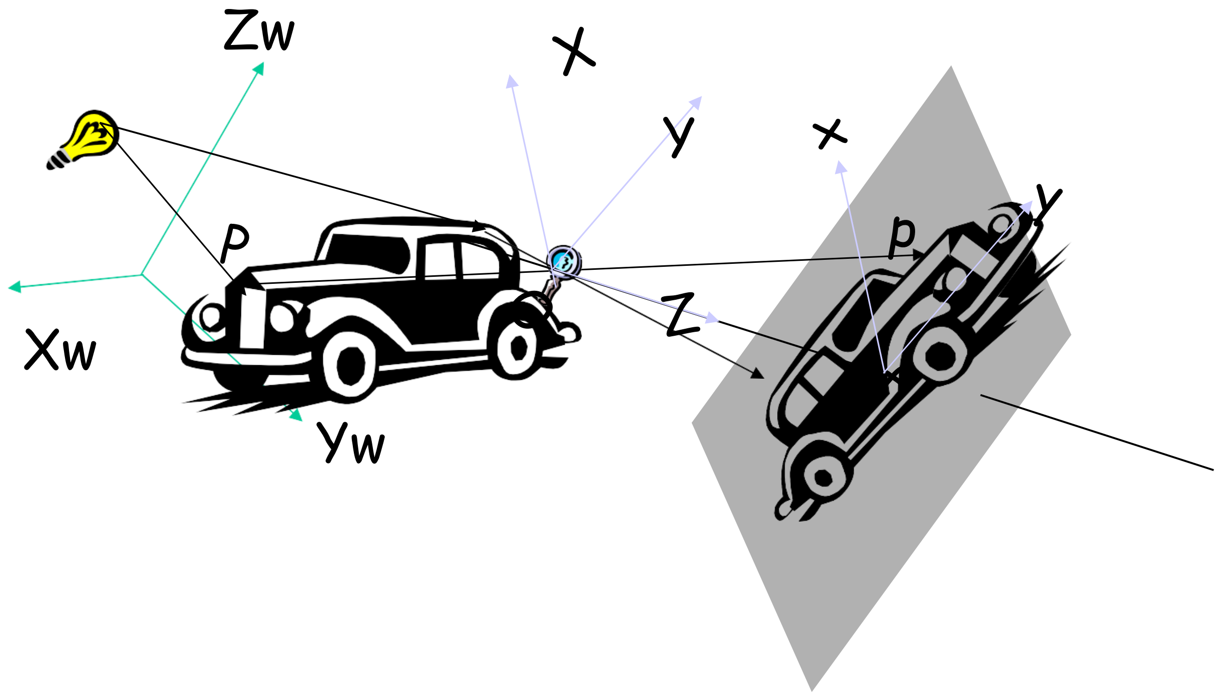
CS 554
Computer Vision

Camera Geometry, Calibration, and Multiple View

Hamdi Dibeklioğlu

Slide Credits: P. Duygulu Sahin, T. Darrell, O. Camps, D. Forsyth, and J. Ponce

Coordinate systems



WORLD, CAMERA and Image Coordinate Systems

Adapted from Octavia Camps

Geometric Camera Models

Issue

- camera may not be at the origin, looking down the z-axis

  extrinsic parameters
- one unit in camera coordinates may not be the same as one unit in world coordinates

  intrinsic parameters

Intrinsic parameters

- Do not depend on the camera location
  - Focal length, CCD dimensions, lens distortion

Extrinsic parameters

- Depend on the camera location
  - Translation, and Rotation parameters

Notions of Geometry

- Homogeneous coordinates
- Matrix representation of geometric transformations
- Extrinsic and intrinsic parameters that relate the world and the camera coordinate frames

Reminder

Dot product

$$\boldsymbol{u} = (u_1, \ldots, u_n)^T$$

$$\boldsymbol{v} = (v_1, \ldots, v_n)$$

$$\boldsymbol{u} \cdot \boldsymbol{v} = u_1 v_1 + \ldots + u_n v_n,$$

$$\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}^T \boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{u}$$

When u has unit norm u.v is sign length of projection of v onto u

Cross product

$$\boldsymbol{u} = (u_1, u_2, u_3)^T$$

$$\boldsymbol{v} = (v_1, v_2, v_3)^T$$

$$\boldsymbol{u} \times \boldsymbol{v} \stackrel{\text{def}}{=} \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$

u x v is orthogonal to these two
If u and v have same direction u x v = 0

$$(\boldsymbol{u} \cdot \boldsymbol{v})^2 = |\boldsymbol{u}|^2 |\boldsymbol{v}|^2 \cos^2 \theta,$$

$$|\boldsymbol{v} \times \boldsymbol{v}|^2 = |\boldsymbol{u}|^2 |\boldsymbol{v}|^2 \sin^2 \theta$$

Forsyth & Ponce

Recap: Homogeneous coordinates

- Add an extra coordinate and use an equivalence relation
- for 3D
  - equivalence relation k*(X,Y,Z,T) is the same as (X,Y,Z,T)
- Motivation
  - Possible to write the action of a perspective camera as a matrix

Adapted from David Forsyth, UC Berkeley

Recap: Homogeneous coordinates

- We are used to describing a location in *Cartesian coordinates*:

$$\mathbf{x} = [x \quad y]^{\mathrm{T}} \qquad\qquad \mathbf{x} = [x \quad y \quad z]^{\mathrm{T}}$$

- Alternatively, we can describe locations in *homogeneous coordinates*:

$$\tilde{\mathbf{x}} = [\tilde{x} \quad \tilde{y} \quad \tilde{w}]^{\mathrm{T}} \qquad\qquad \tilde{\mathbf{x}} = [\tilde{x} \quad \tilde{y} \quad \tilde{z} \quad \tilde{w}]^{\mathrm{T}}$$

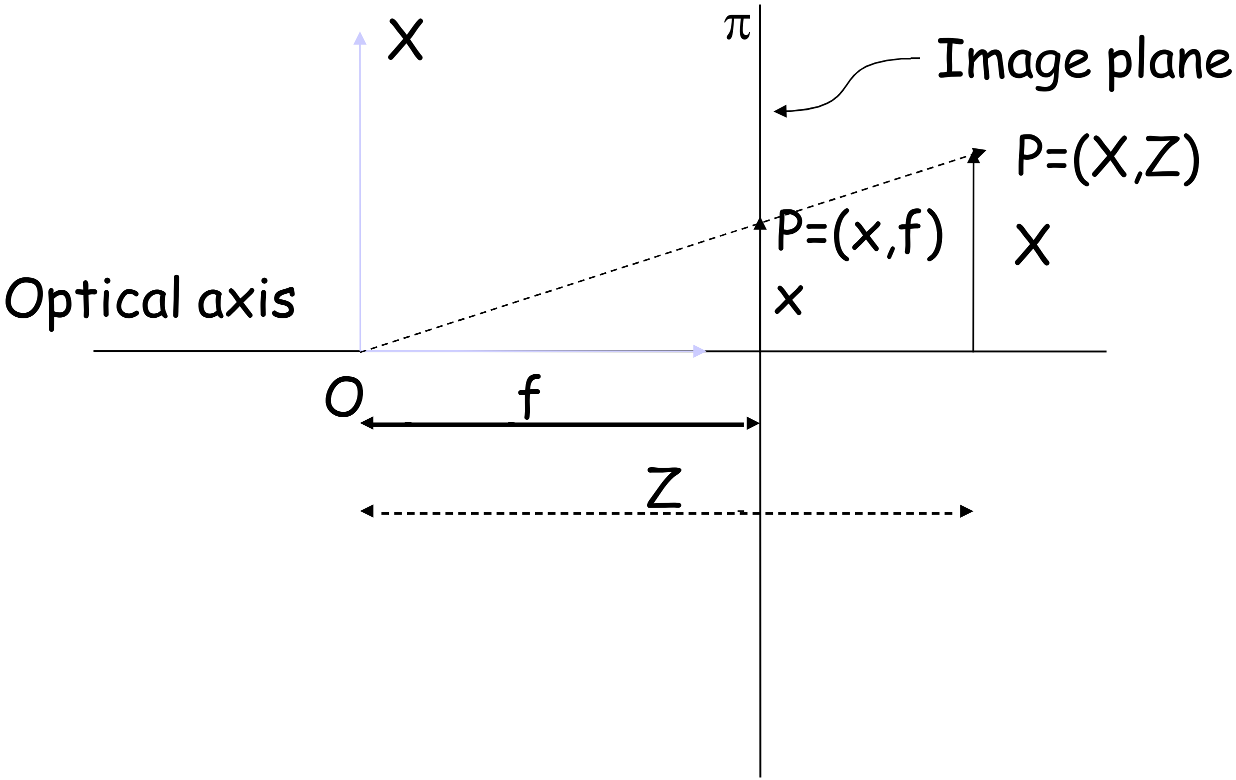
- The corresponding Cartesian coordinates are given by:

$$\mathbf{x} = [\tilde{x}/\tilde{w} \quad \tilde{y}/\tilde{w}]^{\mathrm{T}} \qquad\qquad \mathbf{x} = [\tilde{x}/\tilde{w} \quad \tilde{y}/\tilde{w} \quad \tilde{z}/\tilde{w}]^{\mathrm{T}}$$

- Essentially, you can think of $\tilde{w}$ as a way to deal with object scale (*"disparity"*)

- Homogeneous coordinates are very useful when working with *perspective transformations* (*homographies*)

Recap: Pinhole Camera Model



$$x = f\,\frac{X}{Z}$$

$$y = f\,\frac{Y}{Z}$$

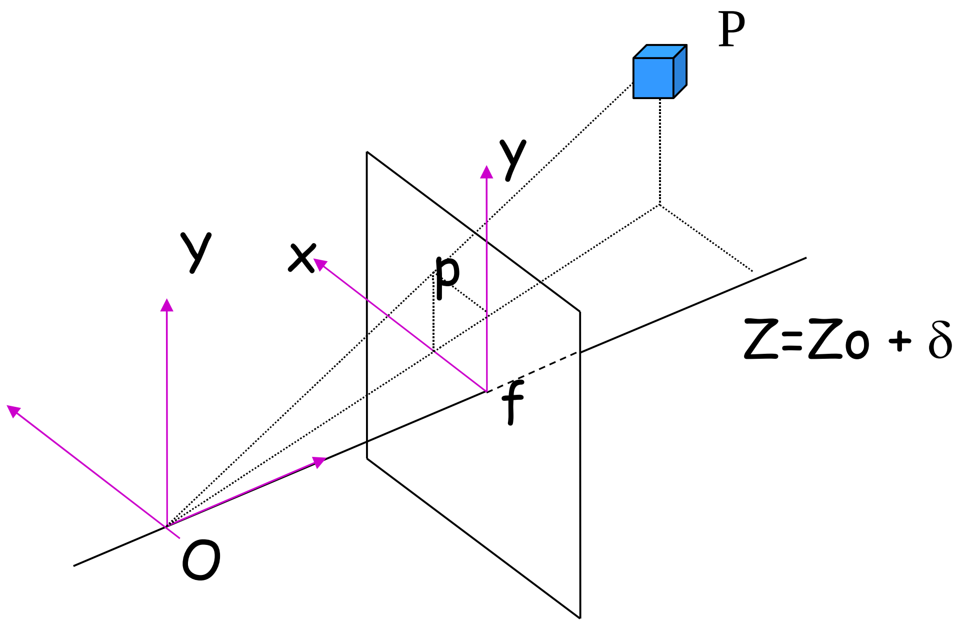
Adapted from Octavia Camps

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates: $(x, y, z) \to (x\frac{f}{z}, y\frac{f}{z})$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow (x\frac{f}{z}, y\frac{f}{z})$$

Weak Perspective Projection



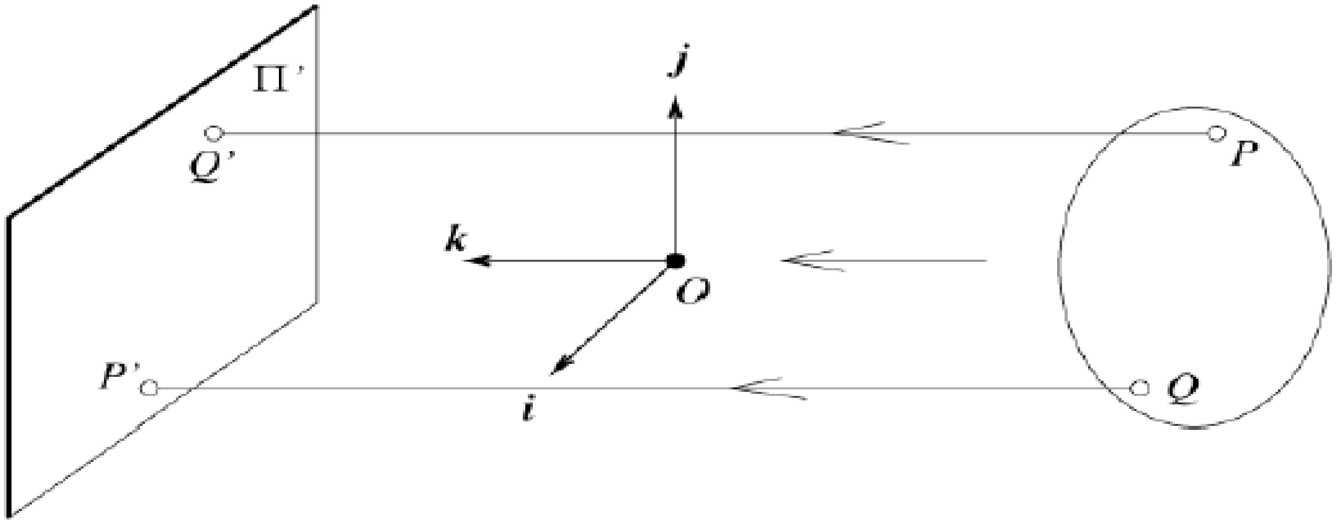
$$x = f\,\frac{X}{Z_0}$$

$$y = f\,\frac{Y}{Z_0}$$

- Object depth δ << Camera distance Zo
- Linear equations !!!

Adapted from Octavia Camps, PennState

Orthographic Projection



$$u = x$$

Assume f is at infinity

$$v = y$$

Forsyth & Ponce

Orthographic Projection Matrix

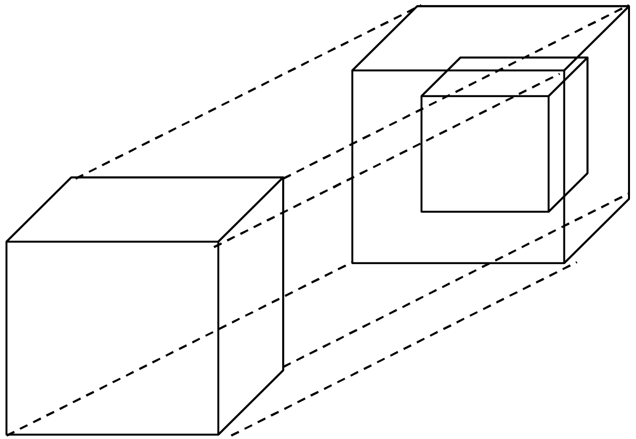
$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

$$= \begin{pmatrix} X \\ Y \\ T \end{pmatrix} \rightarrow \frac{1}{T} \begin{pmatrix} X \\ Y \end{pmatrix}$$

HC Non-HC

Adapted from David Forsyth, UC Berkeley

Weak Perspective vs Orthographic Projection



Weak perspective = Orthographic projection + Isotropic Scaling

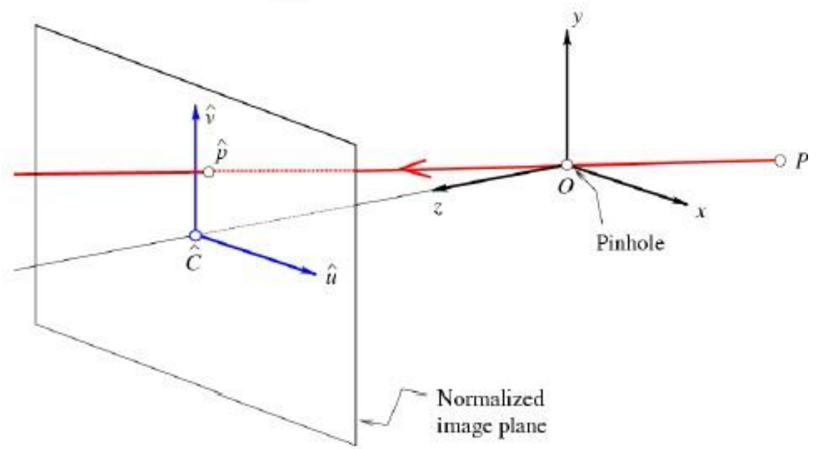
Adapted from Octavia Camps, PennState

Camera parameters

- Intrinsic parameters
  - Focal length, principal point, aspect ratio, angle between axes
- Extrinsic parameters
  - Translation, and Rotation parameters

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} Transformation \\ representing \\ intrinsic\ parameters \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} Transformation \\ representing \\ extrinsic\ parameters \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Intrinsic parameters



Forsyth&Ponce

Perspective projection

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

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

Adapted from Trevor Darrell, MIT

Intrinsic parameters: focal length

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

$$(U, V, W) \rightarrow \left(\frac{U}{W}, \frac{V}{W}\right) = (u, v)$$

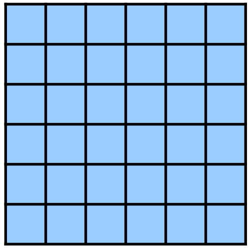
$$p = M_{int} . P$$

Adapted from Octavia Camps, PennState

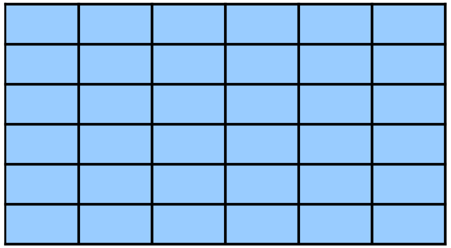
Intrinsic parameters: aspect ratio

- The CCD sensor is made of a rectangular grid $n \times m$ of photosensors.
- Each photosensor generates an analog signal that is digitized by a frame grabber into an array of $N \times M$ pixels.

Pixels may not be square



vs



$$u = \alpha \frac{x}{z}$$

$$v = \beta \frac{y}{z}$$

$$M_{int} = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}$$

Adapted from Octavia Camps, PennState

Intrinsic parameters: origin

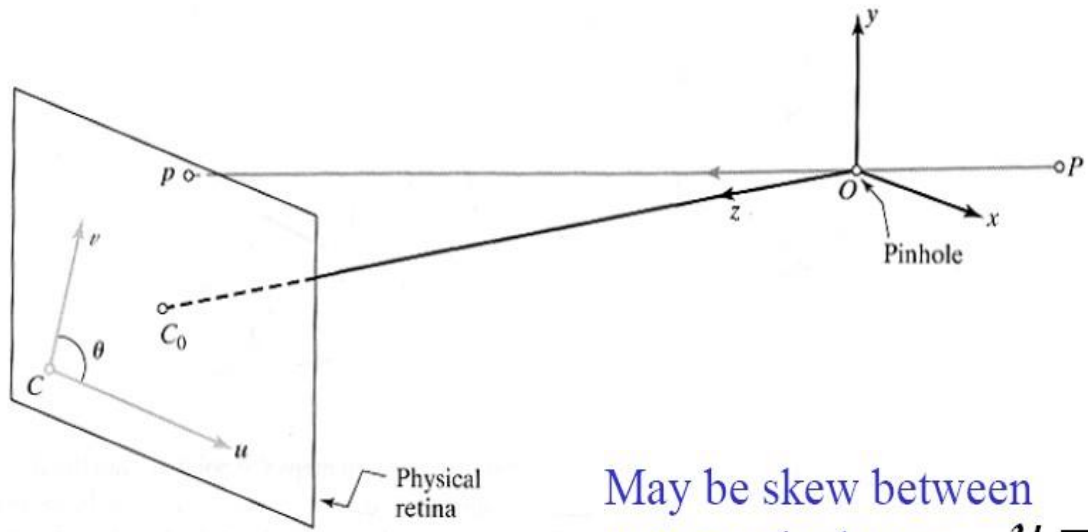
We don't know the origin of our camera pixel coordinates

$$u = \alpha \frac{x}{z} + u_0$$

$$v = \beta \frac{y}{z} + v_0$$

$$M_{int} = \begin{bmatrix} \alpha & 0 & uo & 0 \\ 0 & \beta & vo & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}$$

Intrinsic parameters: angle between axes

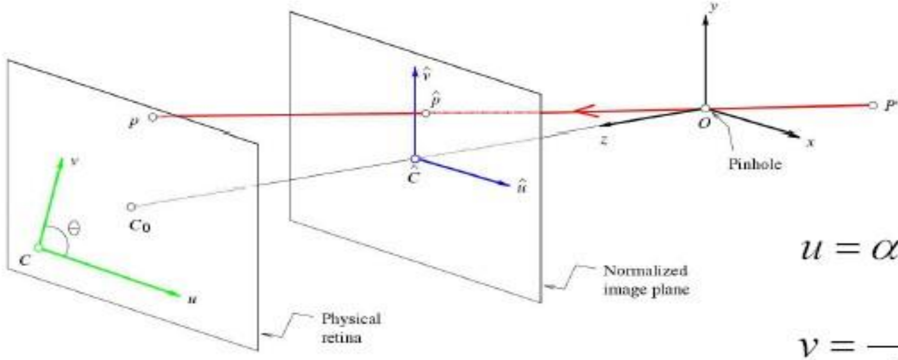


May be skew between
camera pixel axes

$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Intrinsic parameters



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Using homogenous coordinates,
we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} \alpha & -\alpha \cot(\theta) & u_0 & 0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

or:

$$\boxed{\vec{p} = \frac{1}{z} \begin{pmatrix} K & \vec{0} \end{pmatrix} \vec{P}}$$

Adapted from Trevor Darrell, MIT

Extrinsic parameters

Translation and rotation of camera frame

$$ {}^{C}P = {}^{C}_{W}R \, {}^{W}P + {}^{C}O_{W} $$

Non-homogeneous coordinates

$$ \begin{pmatrix} C_X \\ C_Y \\ C_Z \\ 1 \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{C}_{W}R & - & {}^{C}O_{W} \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} W_X \\ W_Y \\ W_Z \\ 1 \end{pmatrix} $$

Homogeneous coordinates

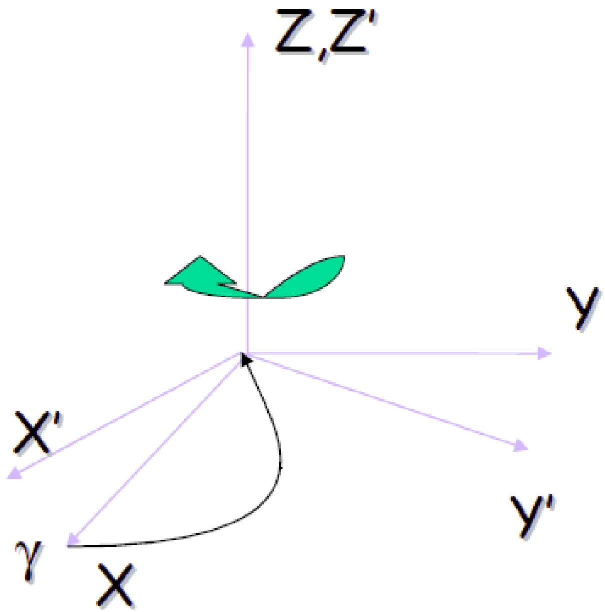
$$ \begin{pmatrix} {}^{C}P \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{C}_{W}\mathcal{R} & {}^{C}O_{W} \\ \mathbf{0}^{T} & 1 \end{pmatrix} \begin{pmatrix} {}^{W}P \\ 1 \end{pmatrix} $$

Block matrix form

Adapted from Trevor Darrell, MIT

3D Rotation of Coordinates Systems

Rotation around the coordinate axes, counter-clockwise (right hand rule):



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

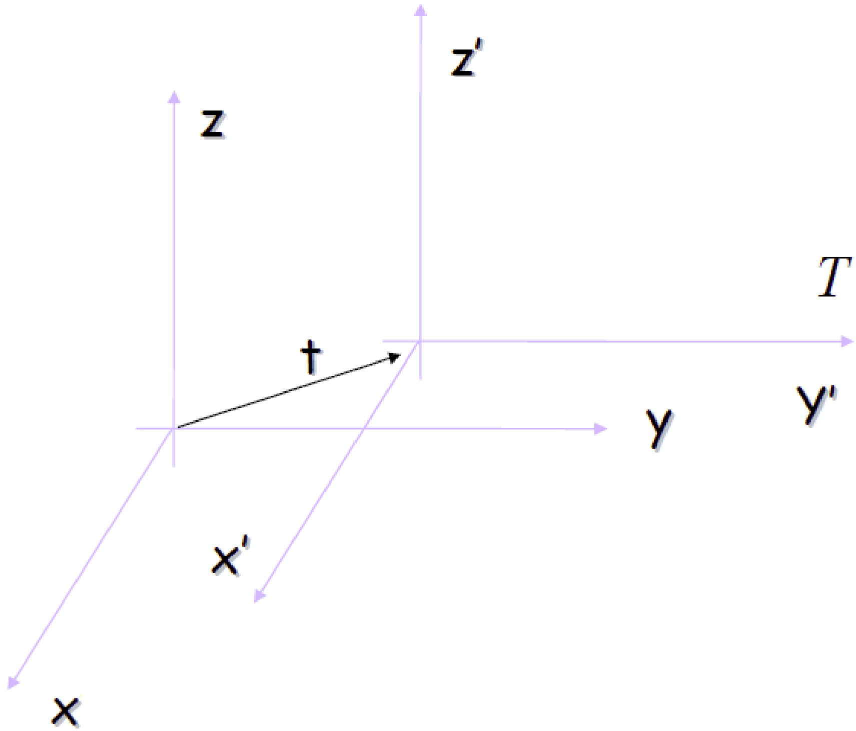
$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Adapted from Octavia Camps

3D Translation of Coordinate Systems

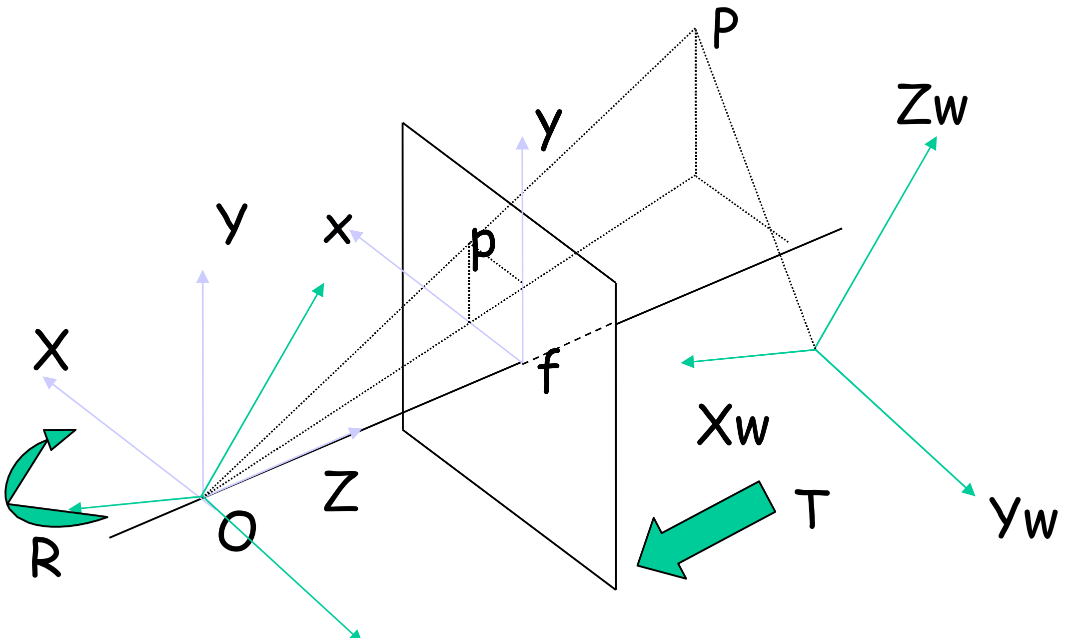
Translate by a vector $t=(t_x,t_y,t_x)^T$:



$$T = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Adapted rom Octavia Camps

Combining Extrinsic and Intrinsic Parameters



$$P = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

$$p = M_{\text{int}} P = M_{\text{int}} M_{\text{ext}} \cdot P_w$$

Adapted from Octavia Camps

Combining Extrinsic and Intrinsic Parameters

$$\vec{p} = \frac{1}{z} \begin{pmatrix} K & \vec{0} \end{pmatrix} \vec{P}$$

Intrinsic

$$^{C}P = {}^{C}_{W}R \; {}^{W}P + {}^{C}O_{W}$$

Extrinsic

$$\vec{p} = \frac{1}{z} K \begin{pmatrix} {}^{C}_{W}R & {}^{C}O_{W} \end{pmatrix} \vec{P}$$

$$\vec{p} = \frac{1}{z} M \; \vec{P}$$

Forsyth & Ponce

Combining Extrinsic and Intrinsic Parameters

$$p = \frac{1}{z}\mathcal{M}P, \quad \text{where} \quad \mathcal{M} = \mathcal{K}(\mathcal{R} \quad t), \tag{2.15}$$

$\mathcal{R} = {}^C_W\mathcal{R}$ is a rotation matrix, $t = {}^CO_W$ is a translation vector, and $P = ({}^Wx, {}^Wy, {}^Wz, 1)^T$ denotes the *homogeneous* coordinate vector of P in the frame (W).

A projection matrix can be written explicitly as a function of its five intrinsic parameters (α, β, u_0, v_0, and θ) and its six extrinsic ones (the three angles defining $\mathcal{R}$ and the three coordinates of t), namely,

$$\mathcal{M} = \begin{pmatrix} \alpha r_1^T - \alpha \cot\theta r_2^T + u_0 r_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} r_2^T + v_0 r_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ r_3^T & t_z \end{pmatrix}, \tag{2.17}$$

where r_1^T, r_2^T, and r_3^T denote the three rows of the matrix $\mathcal{R}$ and t_x, t_y, and t_z are the coordinates of the vector t.

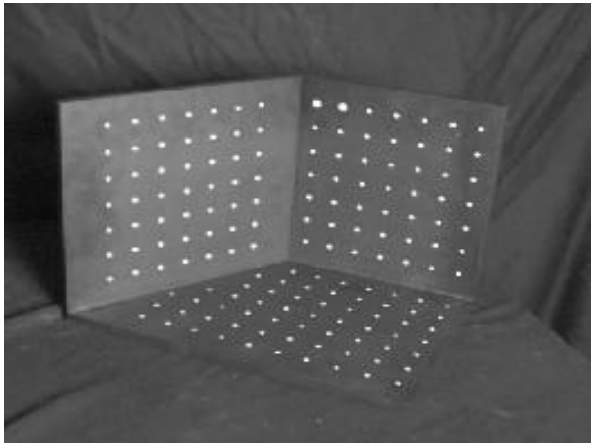
Forsyth & Ponce

Camera Calibration

Compute the camera intrinsic and extrinsic parameters using only observed camera data

Place a known object in the scene

- identify correspondence between image and scene
- compute mapping from scene to image



Camera Calibration

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

Adapted from Trevor Darrell

Camera Calibration

$$u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}$$
$$v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}$$

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Adapted from Trevor Darrell

Camera Calibration

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

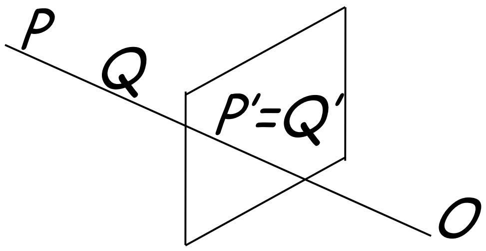
M has 12 entries
Each image point provides 2 equations
m_{ij}'s can be computed by Least Square Solution

Adapted from Trevor Darrell

Multiple View Geometry

Multiple Views

Despite the wealth of information contained in a a photograph, the **depth** of a scene point along the corresponding projection ray **is not directly accessible in a single image**



3D Points on the same viewing line have the same 2D image:
2D imaging results in depth information loss

With at least two pictures, depth can be measured by triangulation.

Human/Animal Visual System

It is the reason that most animals have at least two eyes and/or move their head when looking around

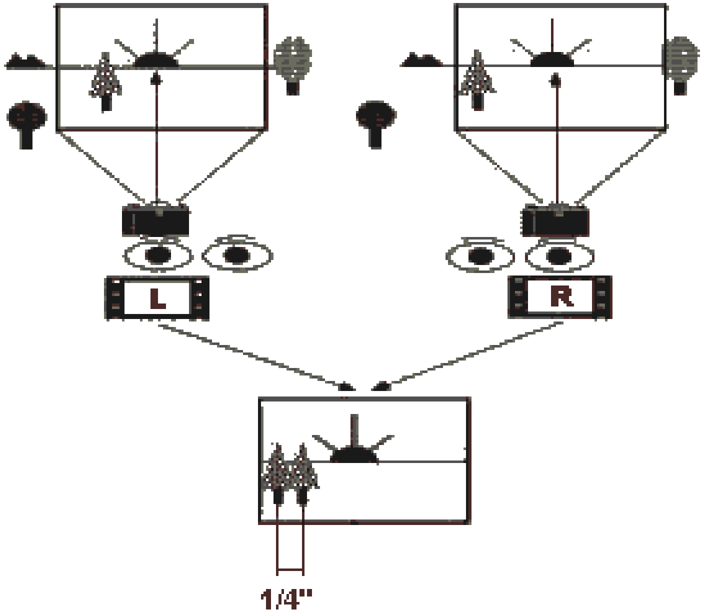
Adapted from David Forsyth, UC Berkeley

Visual Robot/Vehicle Navigation

This is also the motivation for equipping autonomous robots and vehicles with a stereo or motion analysis systems.

Human Vision

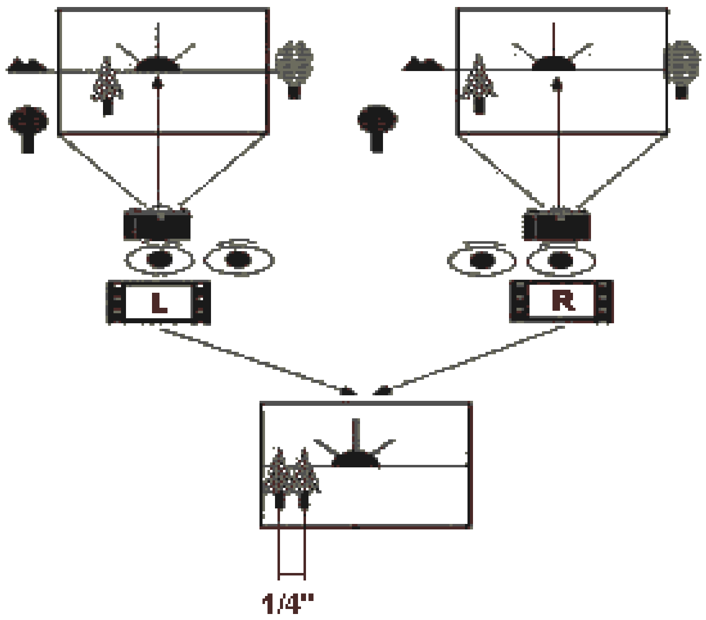
- Humans have two eyes, both forward facing but horizontally spaced by approximately 60mm.
- When looking at an object, each eye will produce a slightly different image, as it will be looking at a slightly different angle.
- The human brain combines both these images into one to give a perception of depth.
- This processing is so quick and seamless that the perception is that we are looking through one big eye rather than two.



http://www.photostuff.co.uk/stereo.htm

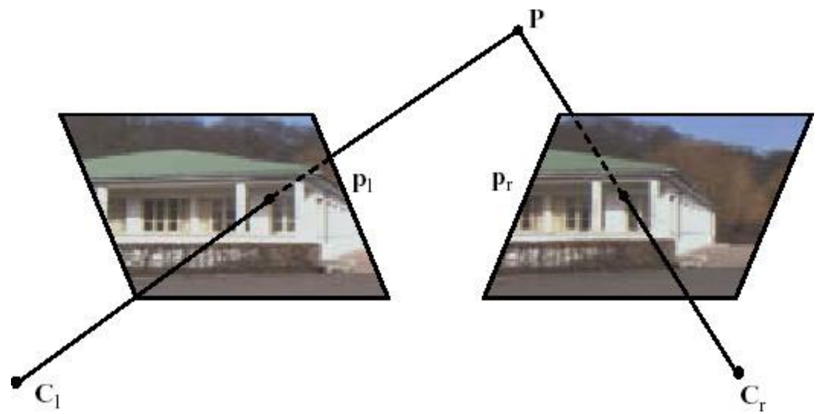
Human Vision

- The brain can also determine depth and how far objects are away from each other by the amount of difference between the two images that it receives.
- The further the subject is from the eye, the less will be the difference between the two images and conversely the nearer the subject, the greater the difference.
- The left and right eyes see the sun in the same place as it is in the distance. The tree being much closer is seen in slightly different places.



http://www.photostuff.co.uk/stereo.htm

Stereo vision = correspondences + reconstruction



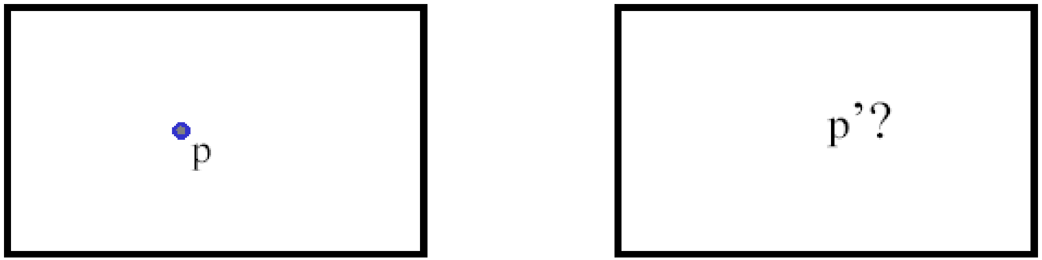
Stereovision involves two problems:

Correspondence:
Given a point p_l in one image, find the corresponding point in the other image

Reconstruction:
Given a correspondence (p_l, p_r) compute the 3D coordinates of the corresponding point in space, P

Adapted from Martial Hebert, CMU

Stereo constraints



Given p in left image, where can corresponding point p' be?

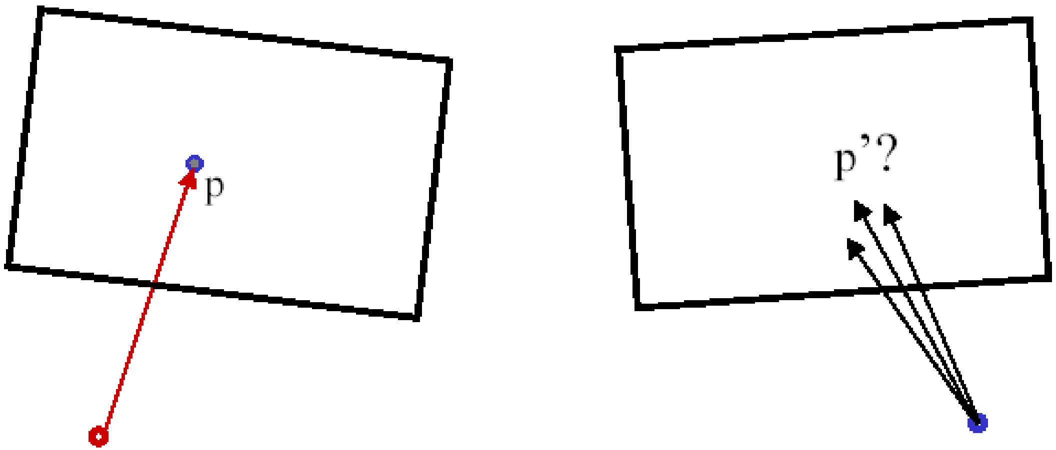
Could be anywhere! Might not be same scene!

… Assume pair of pinhole views of static scene:

Adapted from Trevor Darrell, MIT

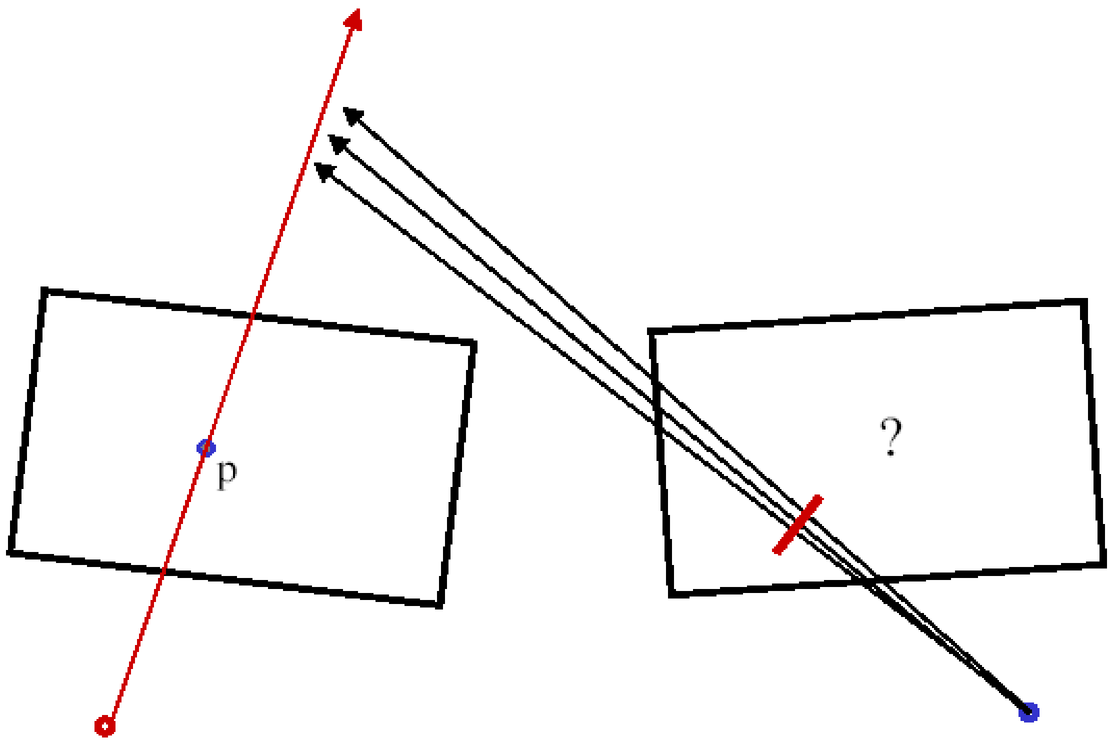
Stereo constraints

Given p in left image, where can p' be?



Adapted from Trevor Darrell, MIT

Epipolar line

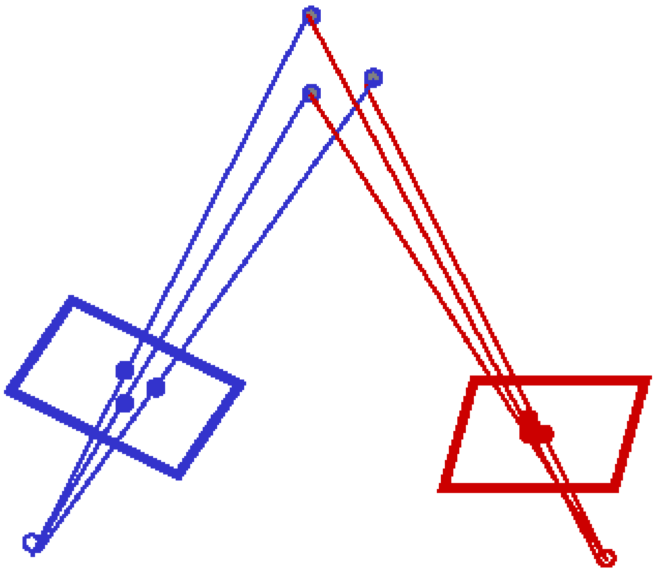


Adapted from Trevor Darrell, MIT

Multiple View Geometry

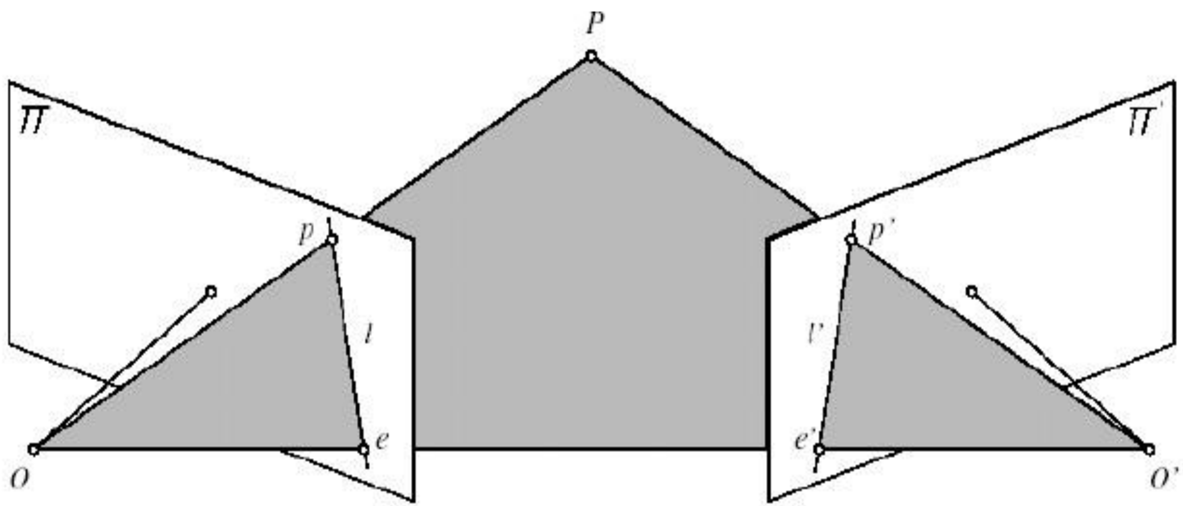
Relate

- 3-D points
- Camera centers
- Camera orientation
- Camera intrinsics



Adapted from Trevor Darrell, MIT

Epipolar constraint

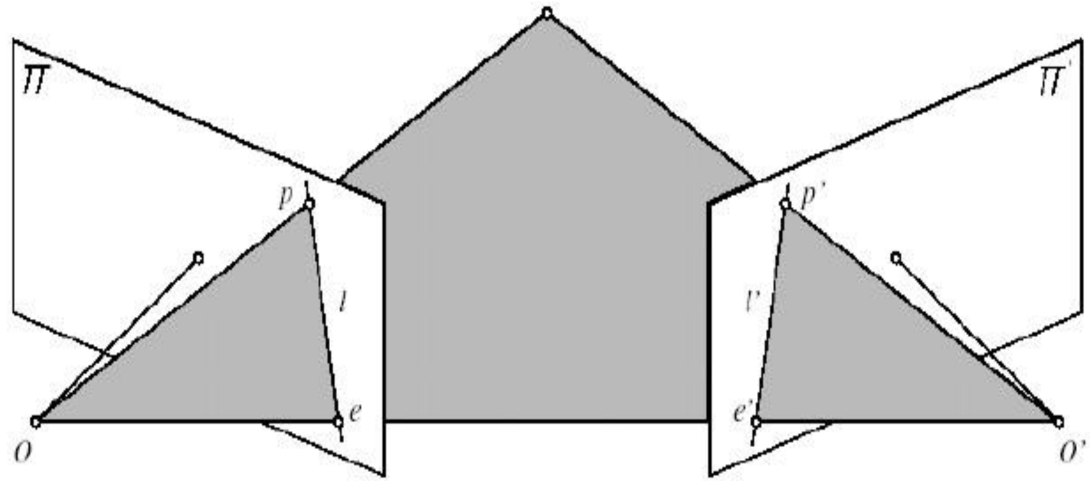


All epipolar lines contain epipole, the image of other camera center.

O, O' : optical centers
p & p' are the images of P → These 5 points all belong to epipolar plane

Forsyth & Ponce

Epipolar constraint

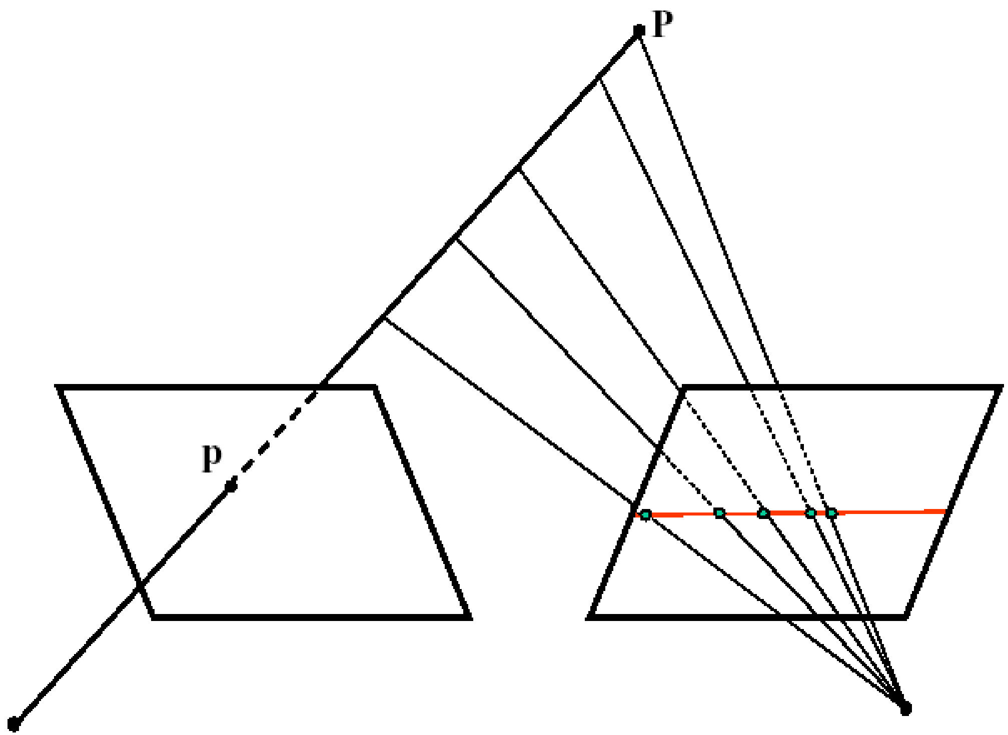


- Point p' lies on the line l' where epipolar plane and the retina π' intersect.
- The line l' is the epipolar line associated with the point p
  - It passes through the point e' where the baseline joining the optical centers O and O' intersects
- The points e and e' are called the epipoles of the cameras

- If p and p' are the images of the same point P, then p' must lie on the epipolar line associated with p → Epipolar constraint

Adapted from Trevor Darrell, MIT

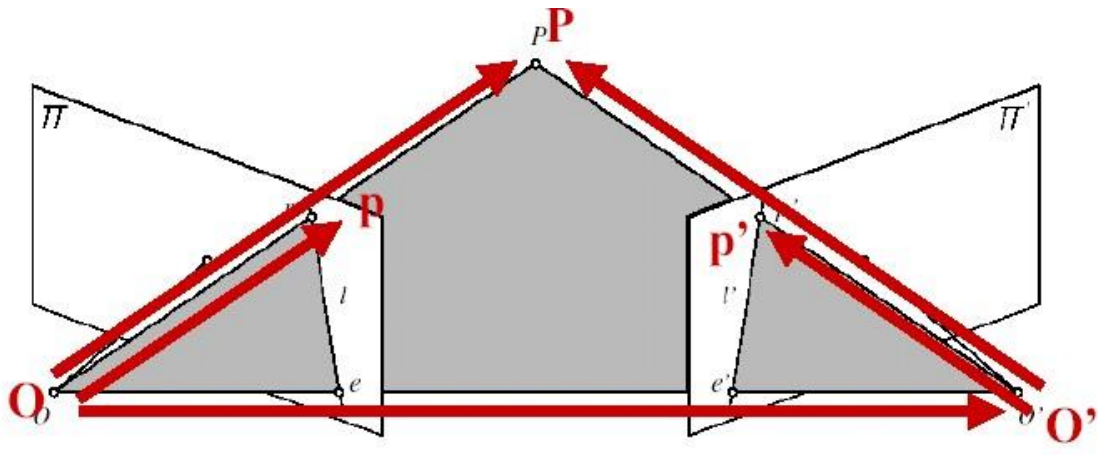
Epipolar constraint



Epipolar constraint greatly limits the search of corresponding points.

Adapted from Martial Hebert, CMU

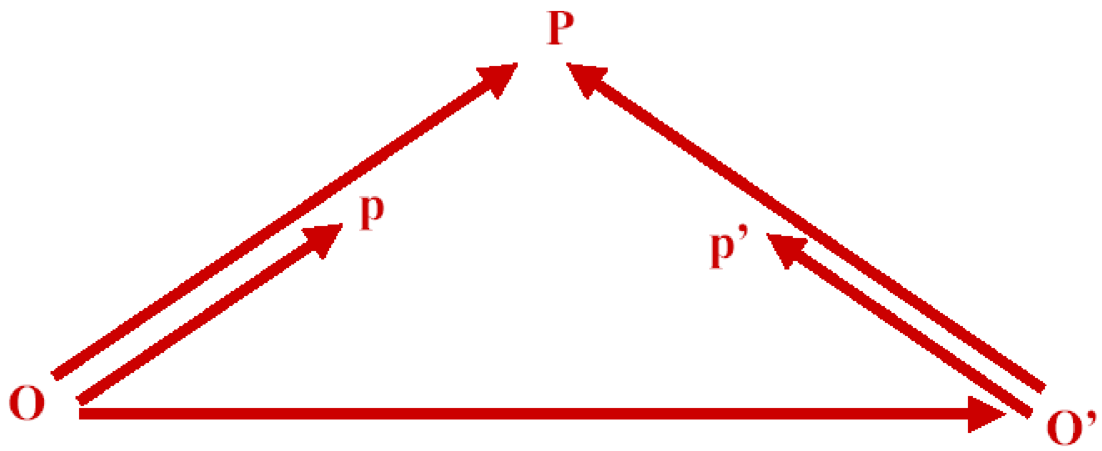
Epipolar constraint – Calibrated case



Assume that the intrinsic parameters of each camera are known

Adapted from Trevor Darrell, MIT

Epipolar constraint – Calibrated case

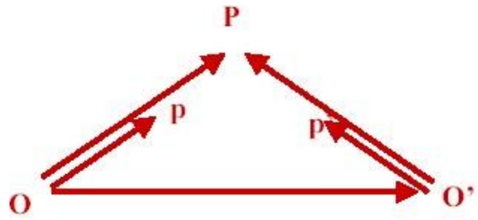


The epipolar constraint: these vectors are coplanar:

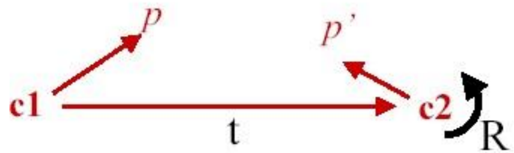
$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$

Adapted from Trevor Darrell, MIT

Epipolar constraint – Calibrated case



$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$



p,p' are image coordinates of P in c1 and c2…

c2 is related to c1 by rotation R and translation t

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0$$

$p = (u,v,1)^T \quad p' = (u',v',1)^T$

Adapted from Trevor Darrell, MIT

Epipolar constraint – Calibrated case

Review: matrix form of cross-product

The vector cross product also acts on two vectors and returns a third vector. Geometrically, this new vector is constructed such that its projection onto either of the two input vectors is zero.

$$\vec{a} \times \vec{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$

$$\vec{b} \cdot \vec{c} = 0$$

Adapted from Trevor Darrell, MIT

Epipolar constraint – Calibrated case

Review: matrix form of cross-product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$

$$\vec{b} \cdot \vec{c} = 0$$

$$[a_x] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$$\boxed{\vec{a} \times \vec{b} = [a_x]\vec{b}}$$

Adapted from Trevor Darrell, MIT

Epipolar constraint – Calibrated case

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0$$

$$\vec{a} \times \vec{b} = [a_x]\vec{b}$$

$$p^T [t_x] \Re p' = 0$$

$$\varepsilon = [t_x] \Re$$

$$\boxed{\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0}$$

Adapted from Trevor Darrell, MIT

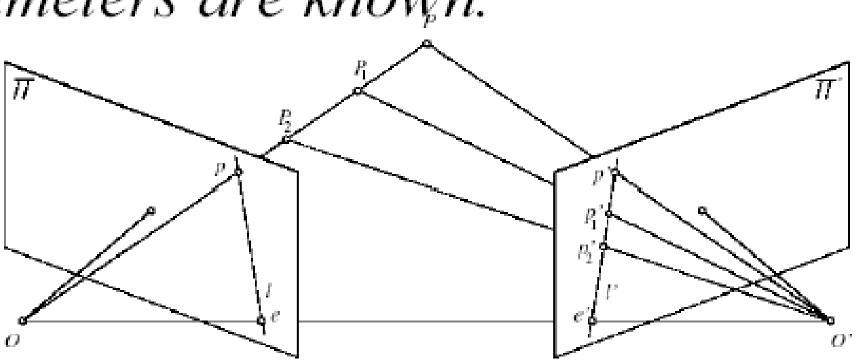
The essential matrix

Matrix that relates image of point in one camera to a second camera, given translation and rotation.

5 independent parameters (up to scale)

Assumes intrinsic parameters are known.

$$\mathcal{E} = [t_x]\Re$$



$$\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0$$

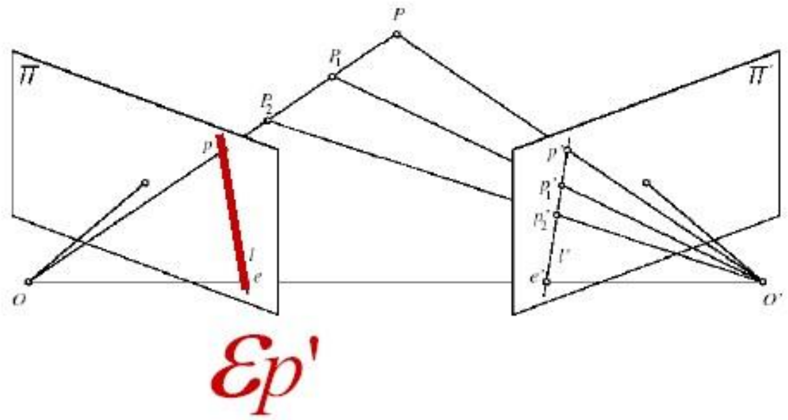
$$\vec{a} \times \vec{b} = [a_x]\vec{b}$$

Adapted from Trevor Darrell, MIT

The essential matrix

$\mathcal{E}p'$ is the epipolar line corresponding to p' in the left camera.

$$au + bv + c = 0$$



$$p = (u, v, 1)^T$$

$$l = (a, b, c)^T$$

$$l \cdot p = 0$$

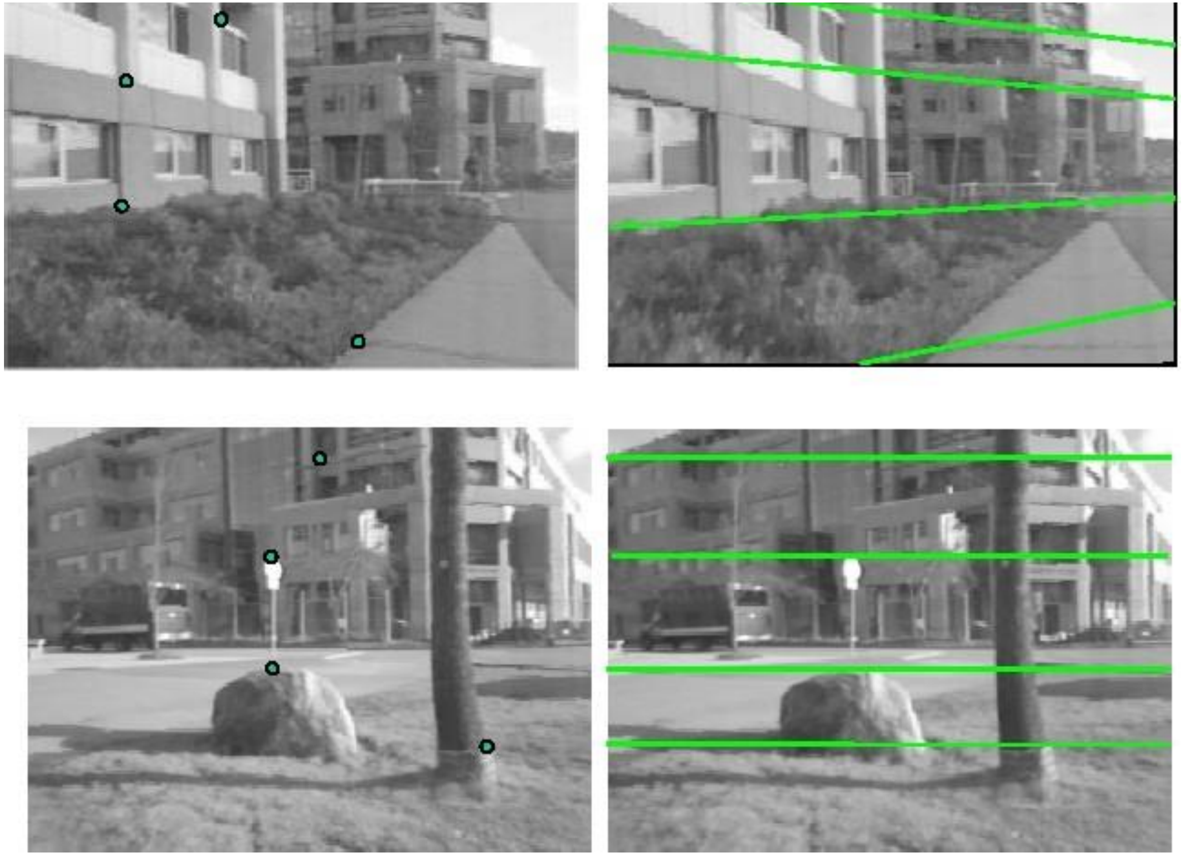
$$\mathcal{E}p' \cdot p = 0$$

$$\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0$$

p lies on the epipolar line associated with the point p'

Adapted from Trevor Darrell, MIT

Epipolar geometry example

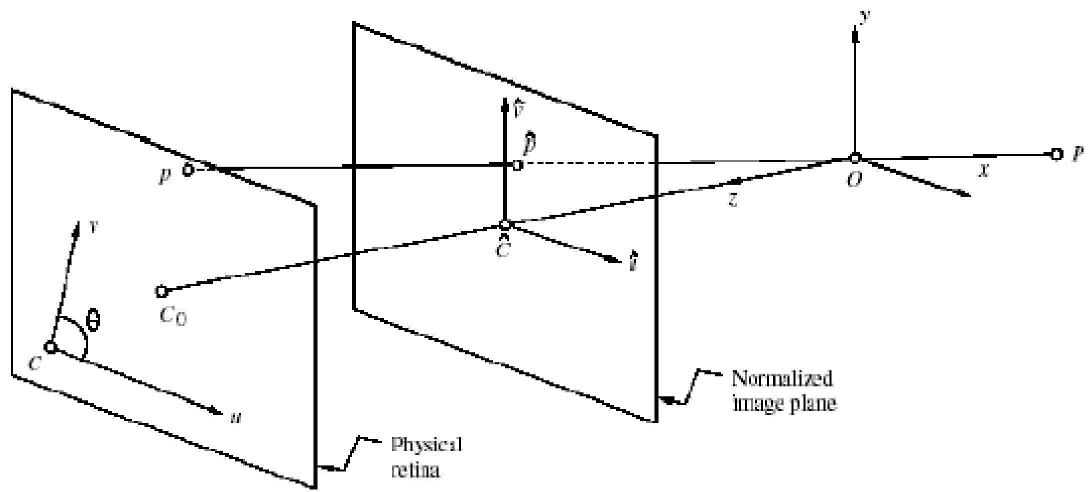


Adapted from Martial Hebert, CMU

What if calibration is unknown?

Recall calibration eqn:

$$\boldsymbol{p} = \mathcal{K}\hat{\boldsymbol{p}}, \quad \text{where} \quad \boldsymbol{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{and} \quad \mathcal{K} \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$



Adapted from Trevor Darrell, MIT

Fundamental Matrix

Essential matrix for points on normalized image plane,

$$\hat{p}^T \mathcal{E} \hat{p}' = 0$$

assume unknown calibration matrix:

$$p = K\hat{p}$$

yields:

$$\boxed{\boldsymbol{p}^T \mathcal{F} \boldsymbol{p}' = 0} \qquad \mathcal{F} = \mathcal{K}^{-T} \mathcal{E} \mathcal{K}'^{-1}$$

Adapted from Trevor Darrell, MIT

Estimation of the Fundamental Matrix

$$\boldsymbol{p}^T \mathcal{F} \boldsymbol{p}' = 0$$

Each point correspondence can be expressed as a single linear equation

$$(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

Adapted from Trevor Darrell, MIT

Estimation of the Fundamental Matrix

$$\boldsymbol{p}^T \mathcal{F} \boldsymbol{p}' = 0$$

Each point correspondence can be expressed as a single linear equation

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \Leftrightarrow (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

Adapted from Trevor Darrell, MIT

The 8 point algorithm (Longuet-Higgins, 1981)

8 corresponding points, 8 equations.

$$\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Invert and solve for $\mathcal{F}$.

under the constraint: $|F|^2 = 1$

(Use more points if available; find least-squares solution to minimize $\sum_{i=1}^{n} (\boldsymbol{p}_i^T \mathcal{F} \boldsymbol{p}_i')^2$)

Adapted from Trevor Darrell, MIT

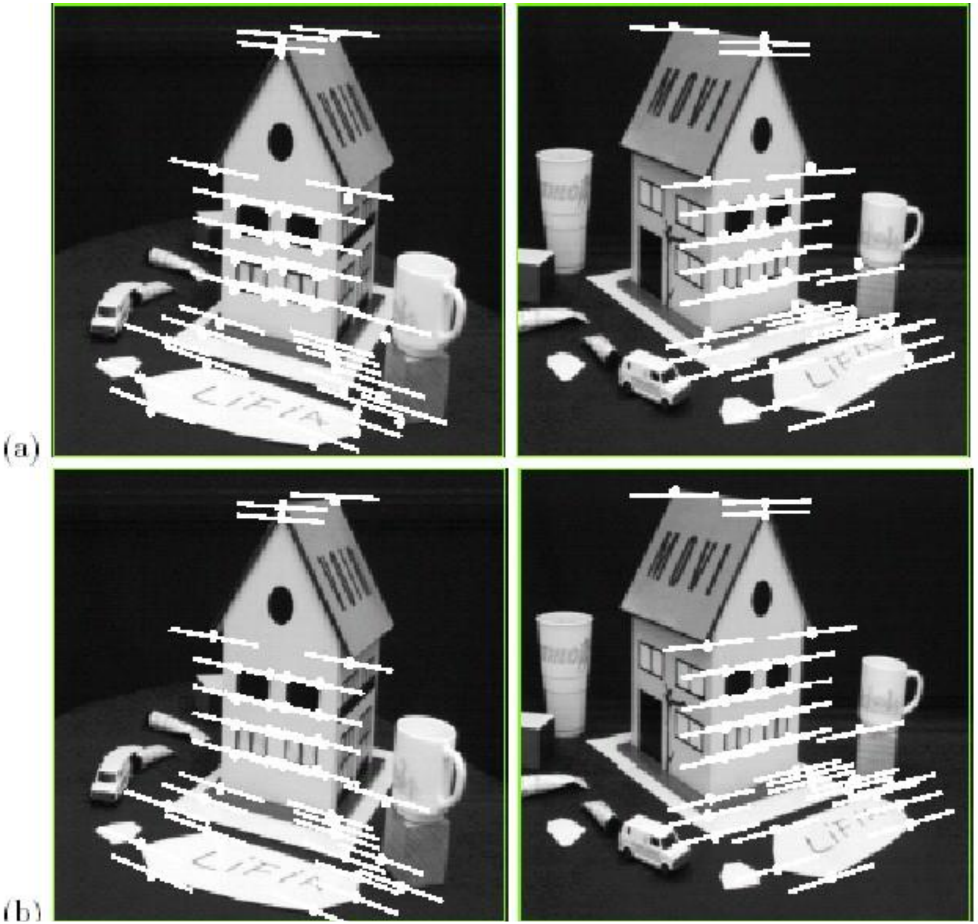
The normalized 8 point algorithm (Hartley, 1995)

Hartley 1995: use SVD.

1. Transform to centered and scaled coordinates
2. Form least-squares estimate of F
3. Set smallest singular value to zero.

Adapted from Trevor Darrell, MIT

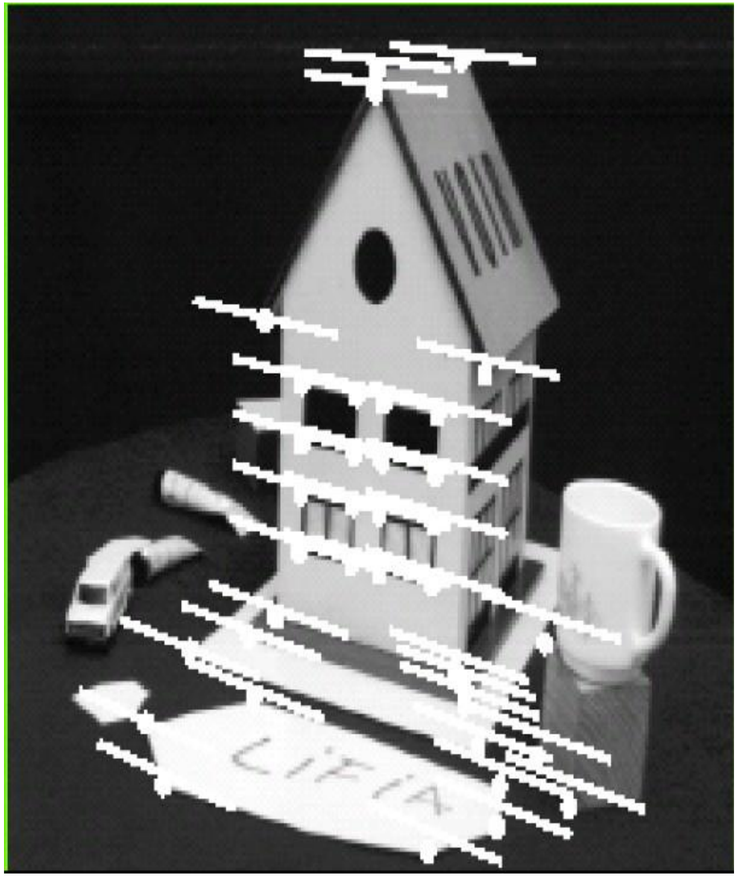
8 point algorithm



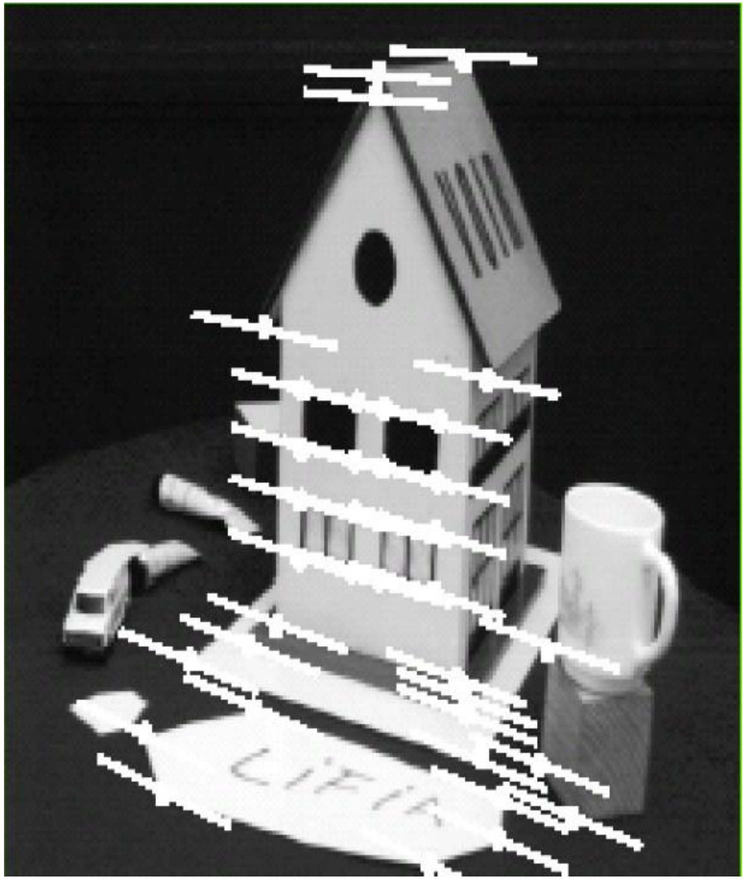
	Linear Least Squares	[Hartley, 1995]
Av. Dist. 1	2.33 pixels	0.92 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel

Adapted from Trevor Darrell, MIT

8 point algorithm

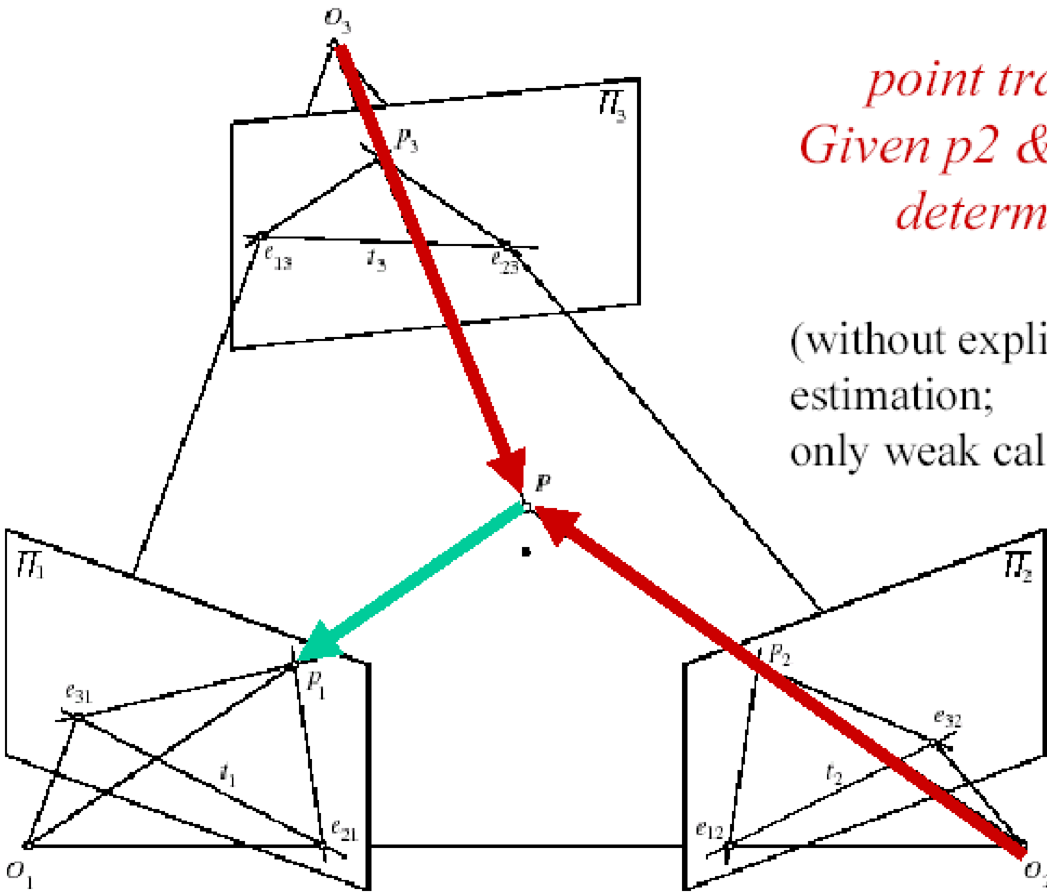


8 point algorithm (Normalized)



Trifocal Geometry

Trifocal plane
formed from
trifocal lines

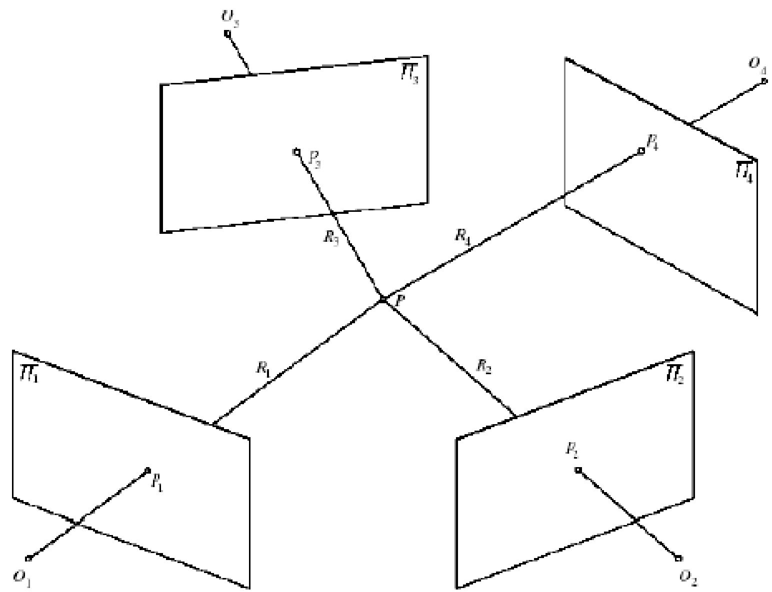


point transfer:
Given p2 & p3, p1 is
determined!

(without explicit depth
estimation;
only weak calibration)

Adapted from Trevor Darrell, MIT

Quadrifocal Geometry



Can form a "quadrifocal tensor"

Faugeras and Mourrain (1995) have shown that it is algebraically dependent on associated essential/fundamental matricies and trifocal tensor: no new constraints added.

No additional independent constraints from more than 3 views.

Adapted from Trevor Darrell, MIT