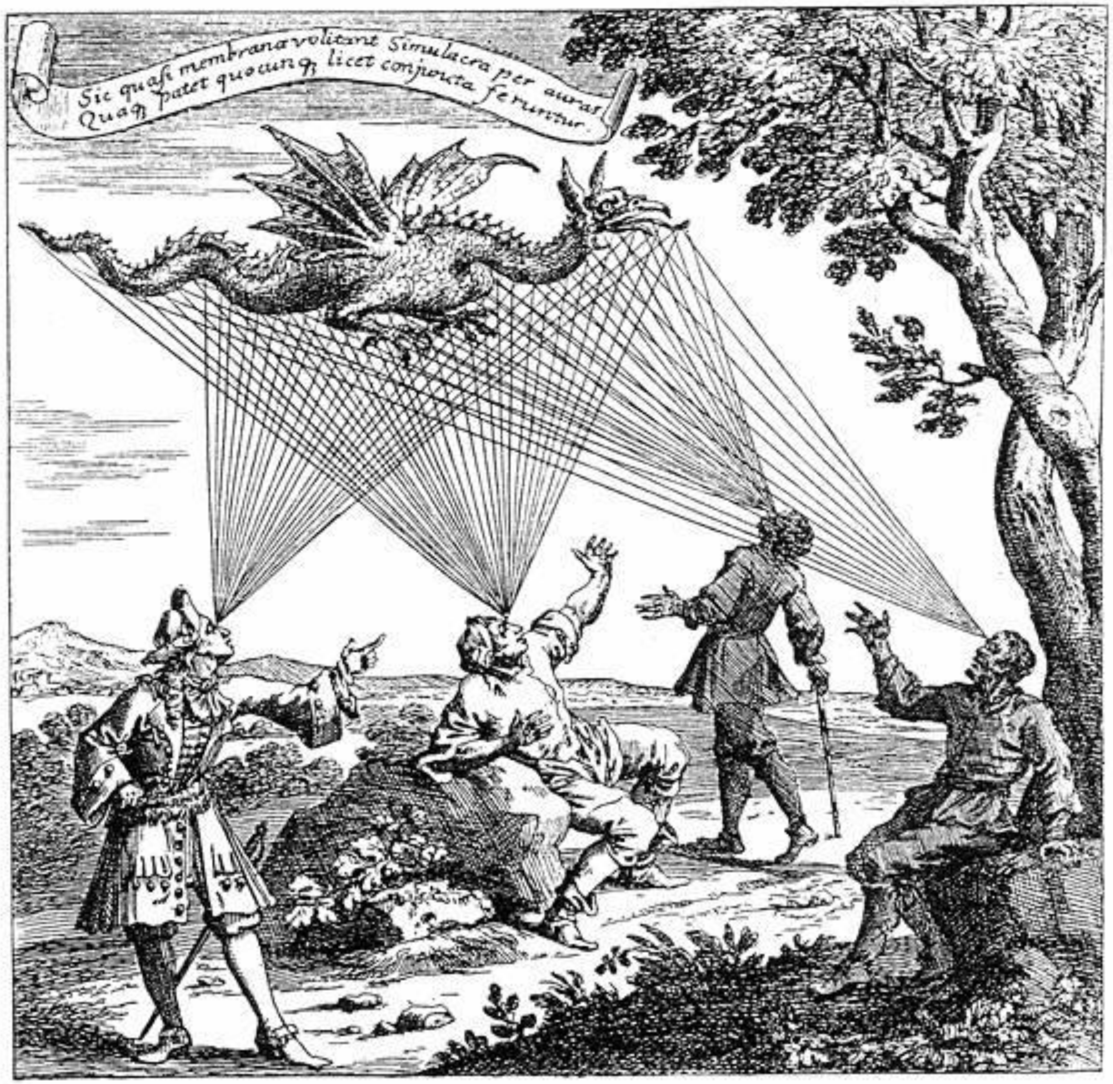
Structure-from-Motion Analysis

2D back to 3D

Structure from motion



Драконъ, видимый подъ различными углами зрѣнія
По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus" Цана. 1702 года.

Multiple-view geometry questions

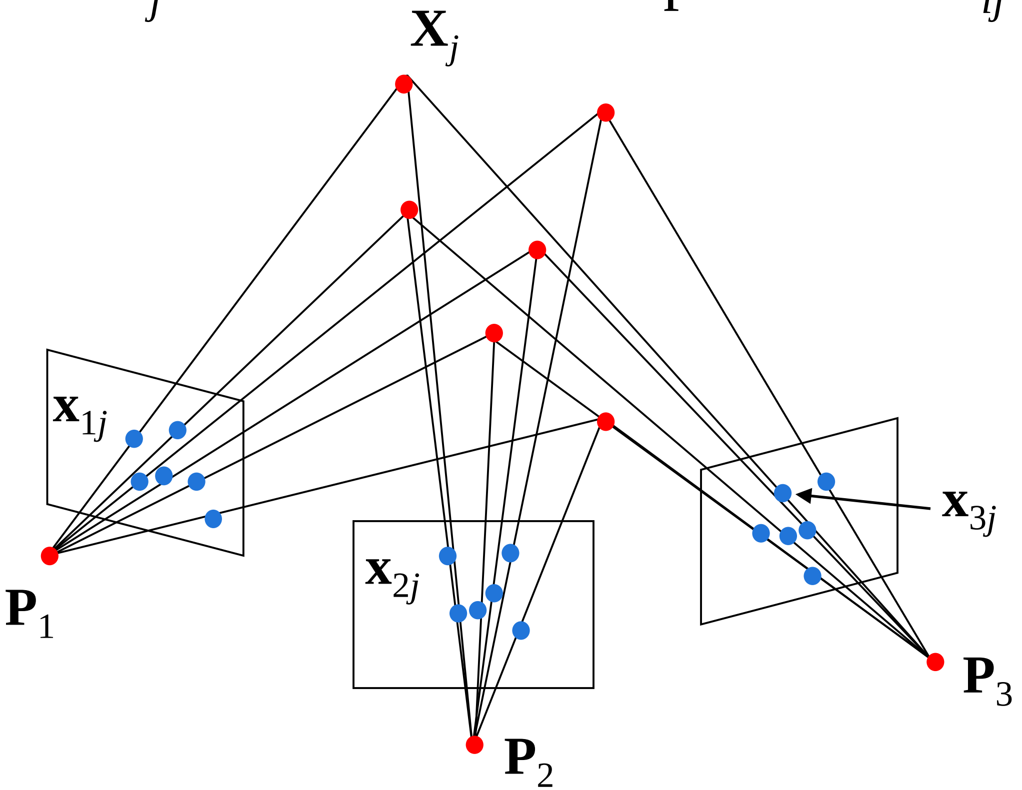
- **Scene geometry (structure):** Given 2D point matches in two or more images, where are the corresponding points in 3D?

- **Correspondence (stereo matching):** Given a point in just one image, how does it constrain the position of the corresponding point in another image?

- **Camera geometry (motion):** Given a set of corresponding points in two or more images, what are the camera matrices for these views?

Structure from motion

- Given: m images of n fixed 3D points
  - $\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- Problem: estimate m projection matrices $\mathbf{P}_i$ and n 3D points $\mathbf{X}_j$ from the mn correspondences $\mathbf{x}_{ij}$



Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

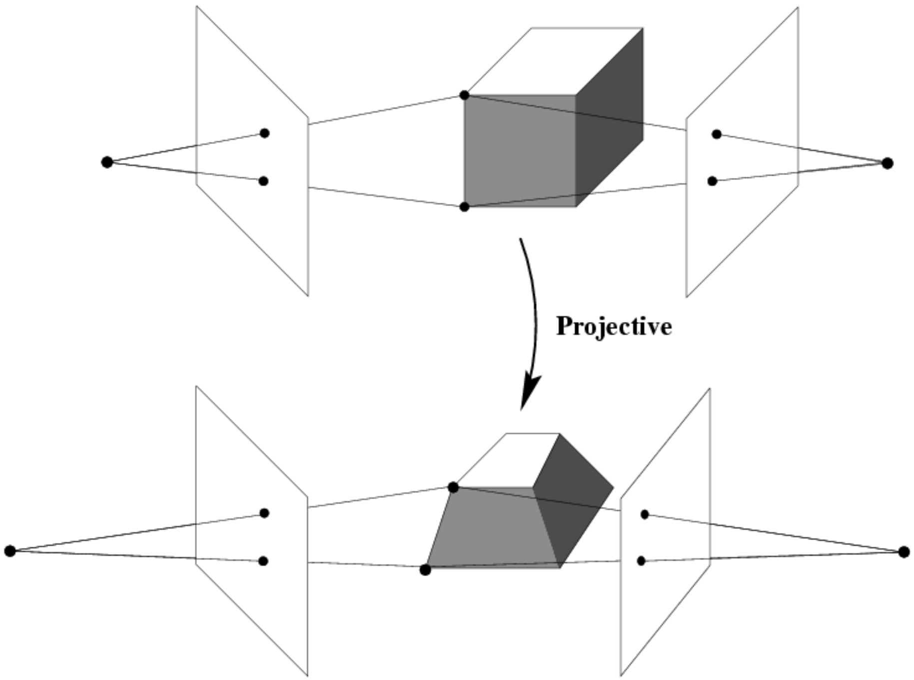
Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same

- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, then the images do not change

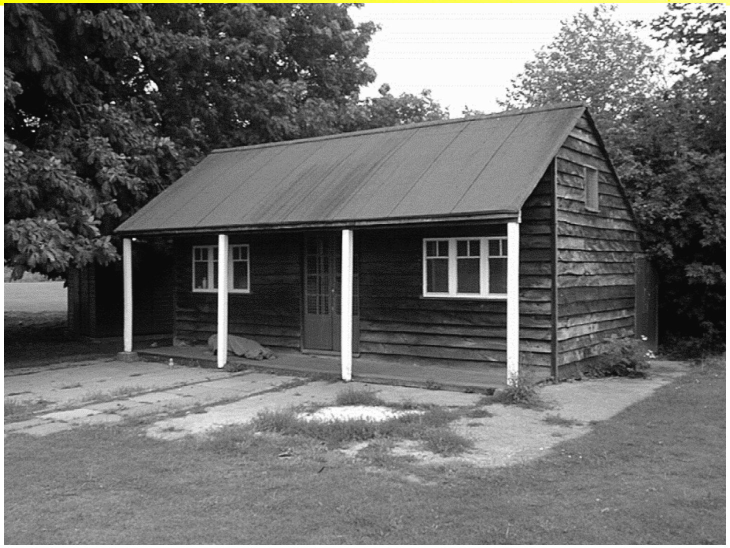
$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

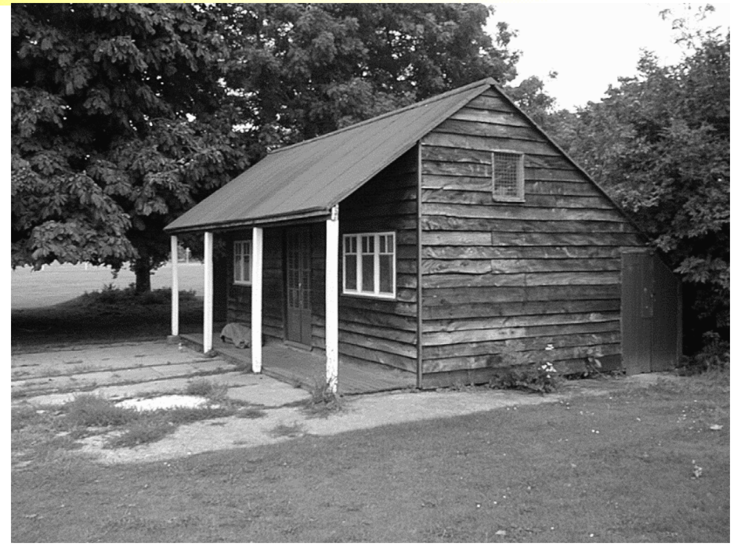
Projective ambiguity

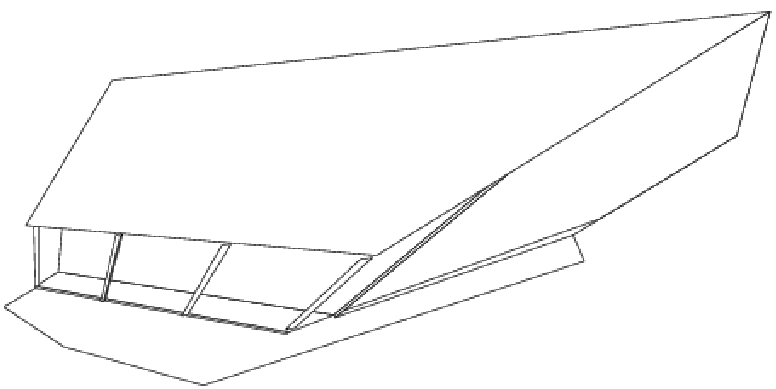


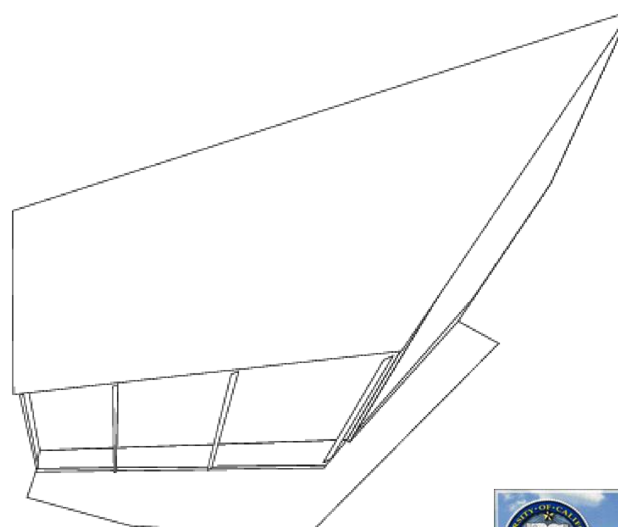
$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{P}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{P}}\mathbf{X}\right)$$

Projective ambiguity

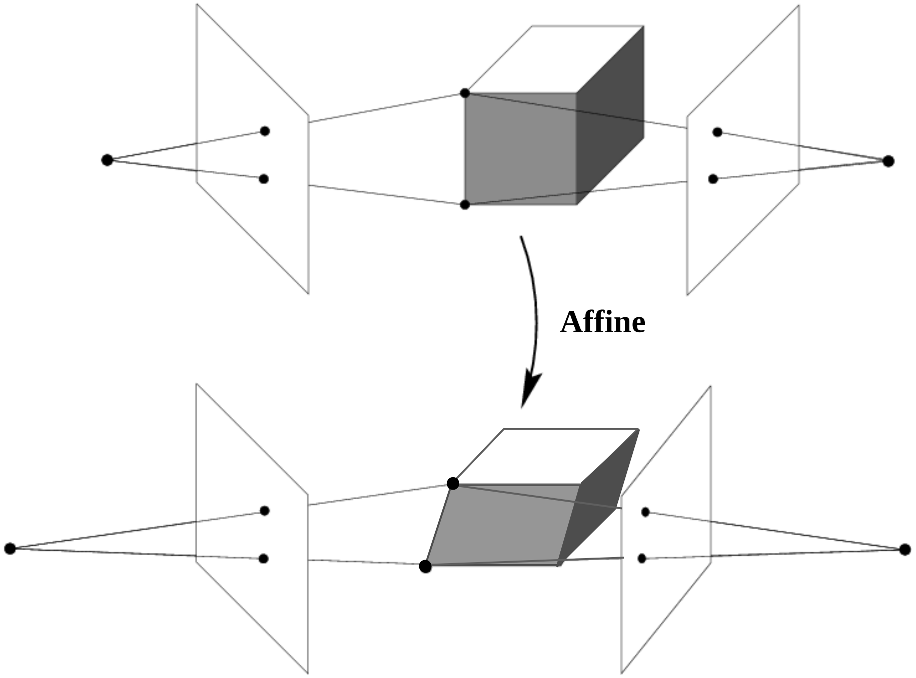






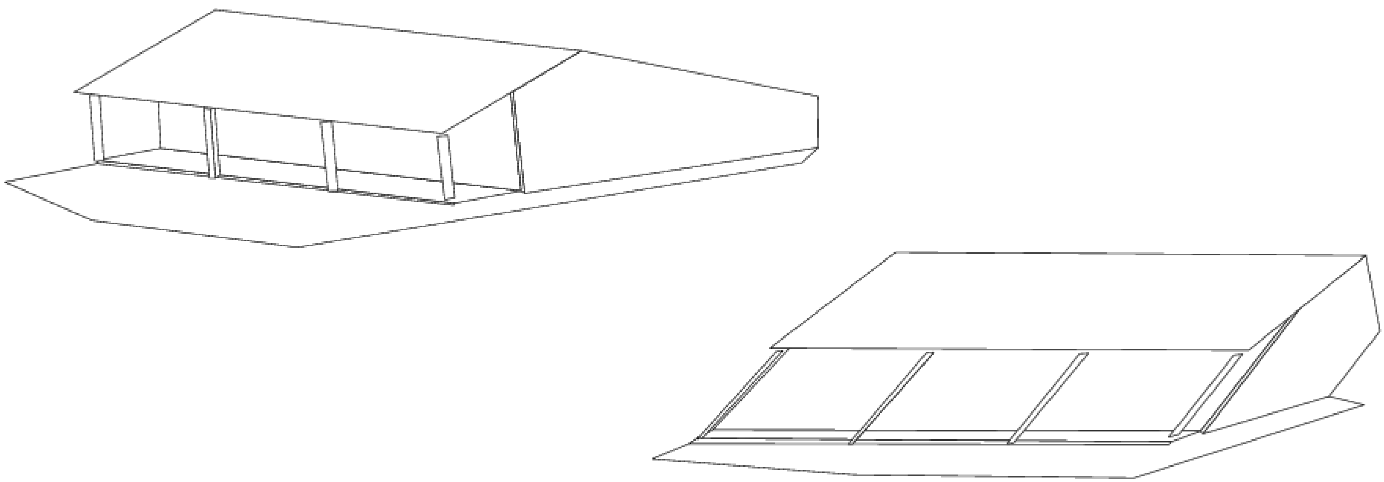


Affine ambiguity

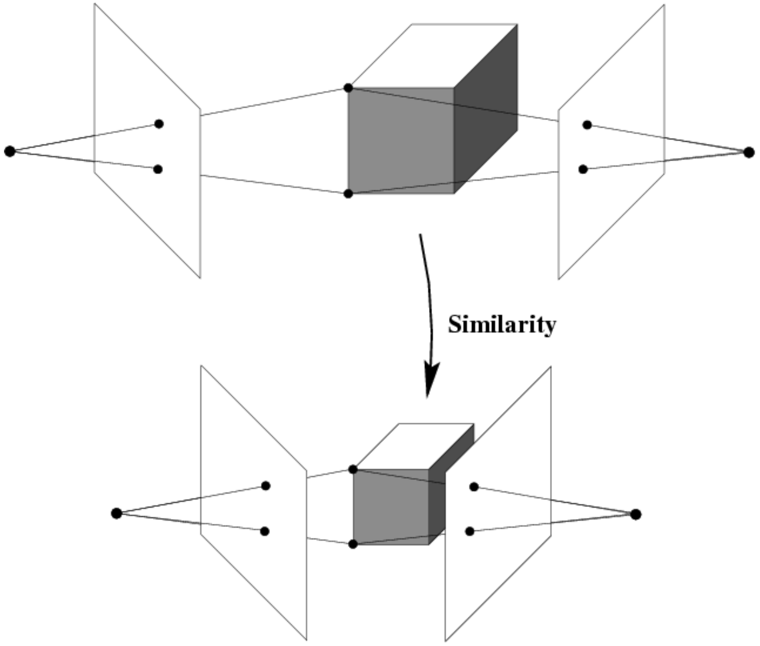


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{A}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{A}}\mathbf{X}\right)$$

Affine ambiguity

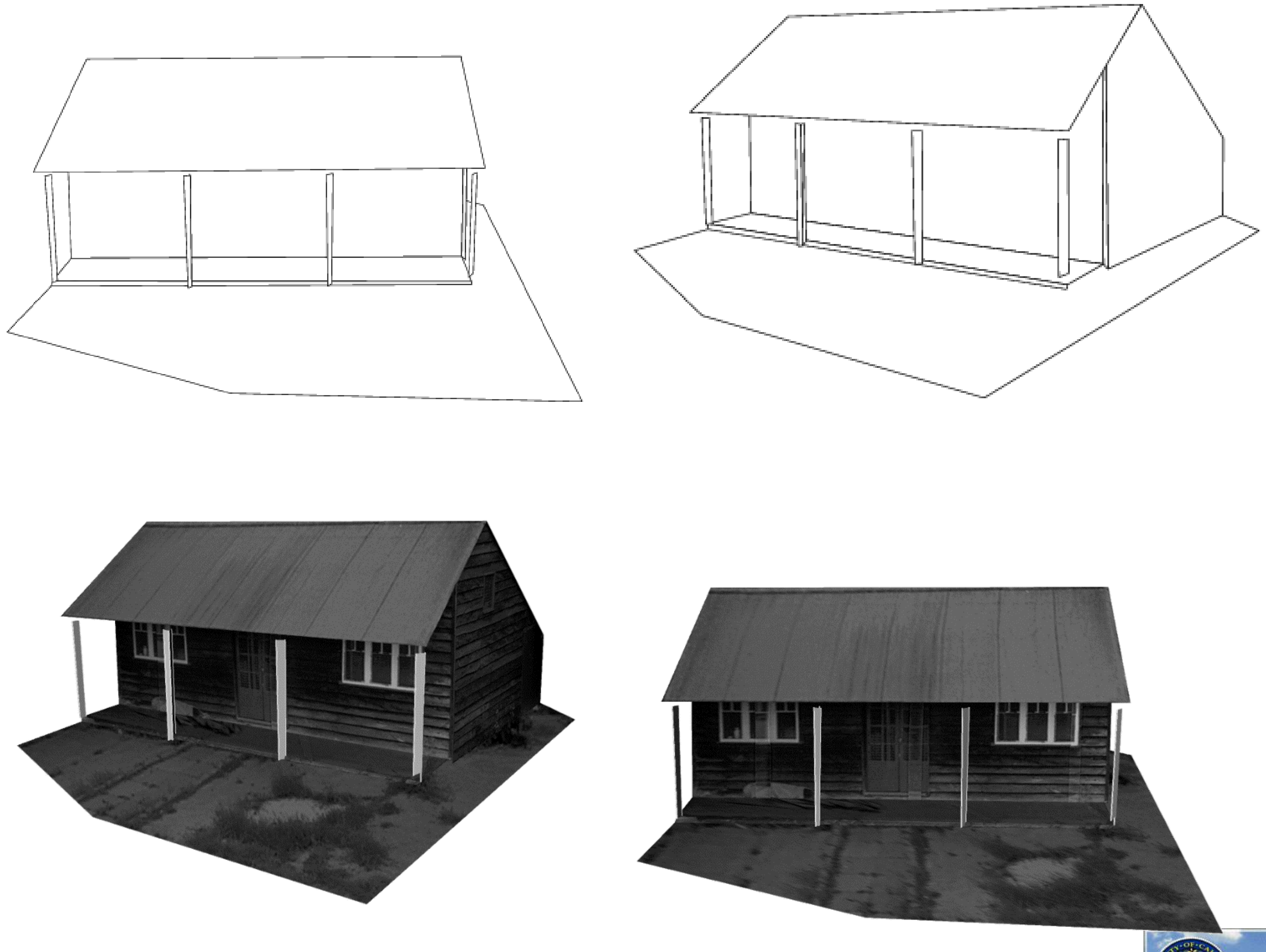


Similarity ambiguity

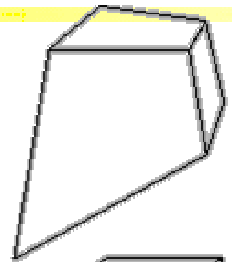
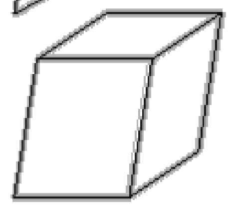
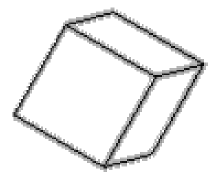
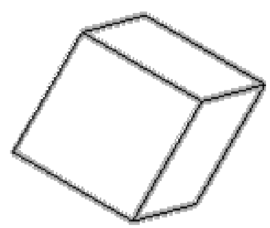


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_S^{-1}\right)\left(\mathbf{Q}_S\mathbf{X}\right)$$

Similarity ambiguity



Hierarchy of 3D transformations

Transformation	Matrix	Figure	Properties
Projective 15dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$		Preserves intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Preserves parallellism, volume ratios
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		Preserves angles, ratios of length
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		Preserves angles, lengths

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean

Sidebar: Matrix form of cross product

- The cross product of two vectors is a third vector, perpendicular to the others (right hand rule)

$$\mathbf{a}\times\mathbf{b}=\begin{bmatrix} a_2b_3-a_3b_2 \\ a_3b_1-a_1b_3 \\ a_1b_2-a_2b_1 \end{bmatrix}=\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}\mathbf{b} \quad =[\mathbf{a}_\times]\mathbf{b}$$

$$\mathbf{a}\cdot(\mathbf{a}\times\mathbf{b})=0$$

$$\mathbf{b}\cdot(\mathbf{a}\times\mathbf{b})=0$$

Side Bar (Useful Duality Relationship)

- Points to line
  - Two points determine a line

$$[x_1, y_1, w_1] \,\&\, [x_2, y_2, w_2]$$

$$[x - \frac{x_1}{w_1}, y - \frac{y_1}{w_1}] \begin{bmatrix} \frac{y_1}{w_1} - \frac{y_2}{w_2} \\ \frac{x_2}{w_2} - \frac{x_1}{w_1} \end{bmatrix} = 0$$

$$\left(\frac{y_1}{w_1} - \frac{y_2}{w_2}\right)x + \left(\frac{x_2}{w_2} - \frac{x_1}{w_1}\right)y - \frac{x_1}{w_1}\left(\frac{y_1}{w_1} - \frac{y_2}{w_2}\right) - \frac{y_1}{w_1}\left(\frac{x_2}{w_2} - \frac{x_1}{w_1}\right) = 0$$

$$(w_2 y_1 - y_2 w_1)x + (w_1 x_2 - w_2 x_1)y + (x_2 y_1 - x_1 y_2) = 0$$

$$\Rightarrow \begin{bmatrix} a \\ b \\ c \end{bmatrix} \propto \begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix} \times \begin{bmatrix} x_2 \\ y_2 \\ w_2 \end{bmatrix}$$

- Lines to point
  - Intersection of two lines is a point

$$a_1 x + b_1 y + c_1 = 0$$

$$a_2 x + b_2 y + c_2 = 0$$

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}, y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

$$\Rightarrow \begin{bmatrix} x \\ y \\ w \end{bmatrix} \propto \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \times \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix}$$

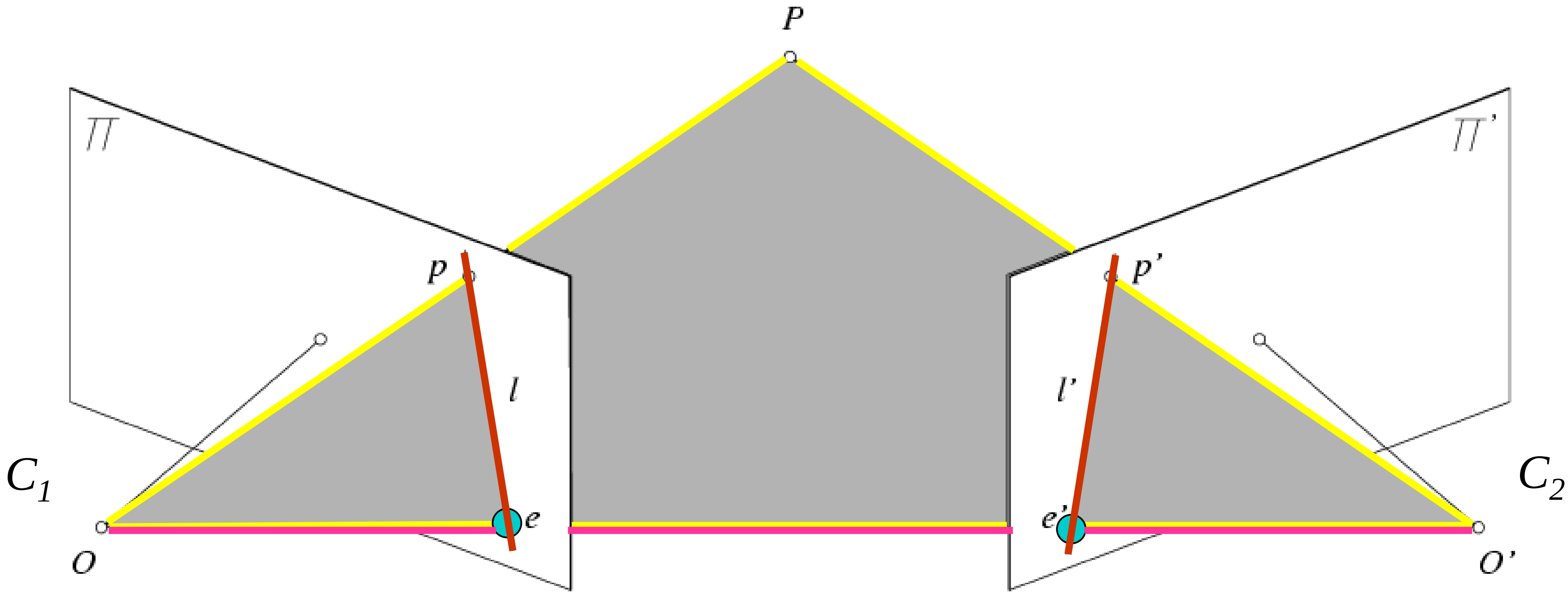
Sidebar: Eigen Decomposition

- Real symmetric Matrix
  - **UDU^T**
  - Eigen vectors are orthogonal
  - Eigen values are real
- Real anti-symmetric (Skew-symmetric) Matrix
  - **UBU^T**
  - Eigen vectors are orthogonal
  - Eigen values are all imaginary and appear in pair
  - Skew-symmetric matrices of odd dimension are singular (a row and a column of zero)

Harder Problem

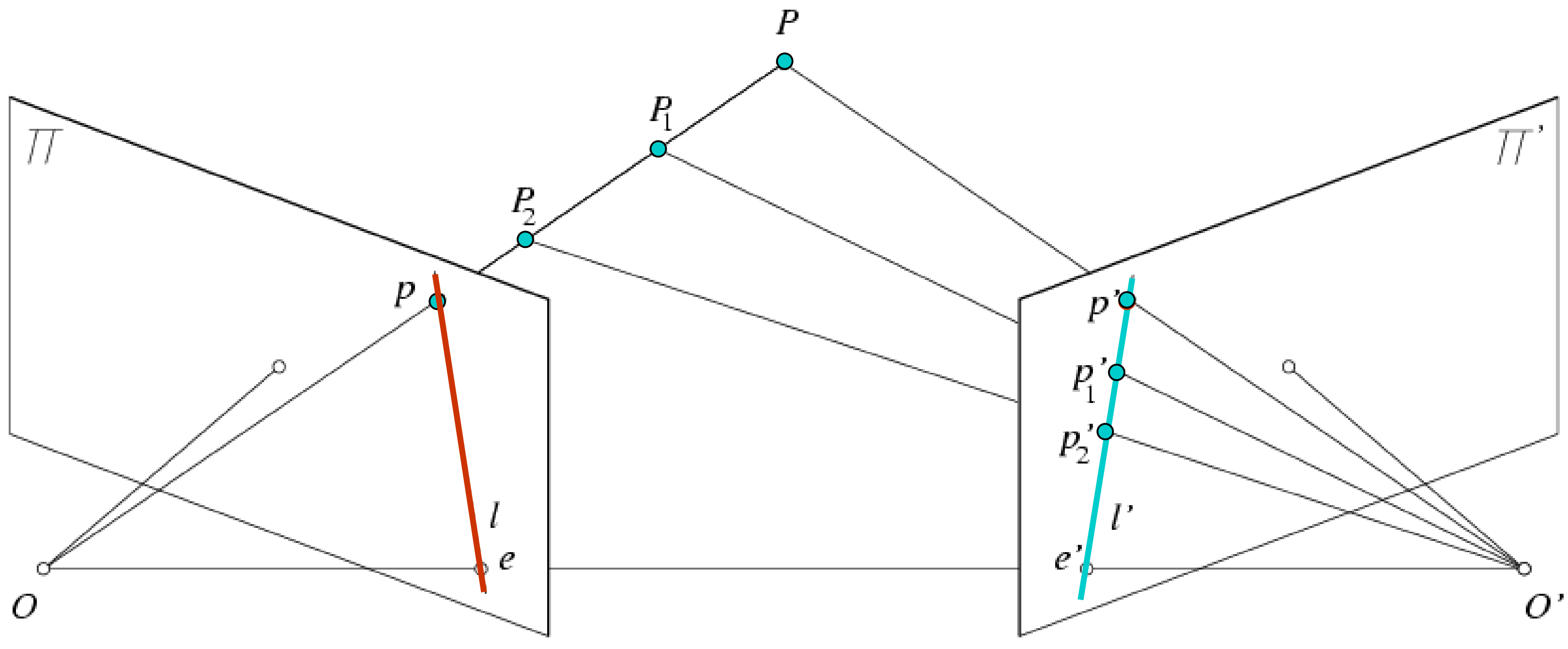
- What happens if we do not know the stereo configuration (baseline, camera orientation, etc.)?
  - If the stereo configuration can be recovered and then rectified, we can again apply the standard stereo matching algorithms described above
- Recovering **R** and **T** (rigid camera motion) using
  - 2 views (fundamental matrix)
  - 3 views (trifocal tensors)
  - N views (factorization)
  - Epipolar geometry is again the key
    - Correct **R** and **T** are not incidental, they match corresponding points and lines in different views

Epipolar geometry



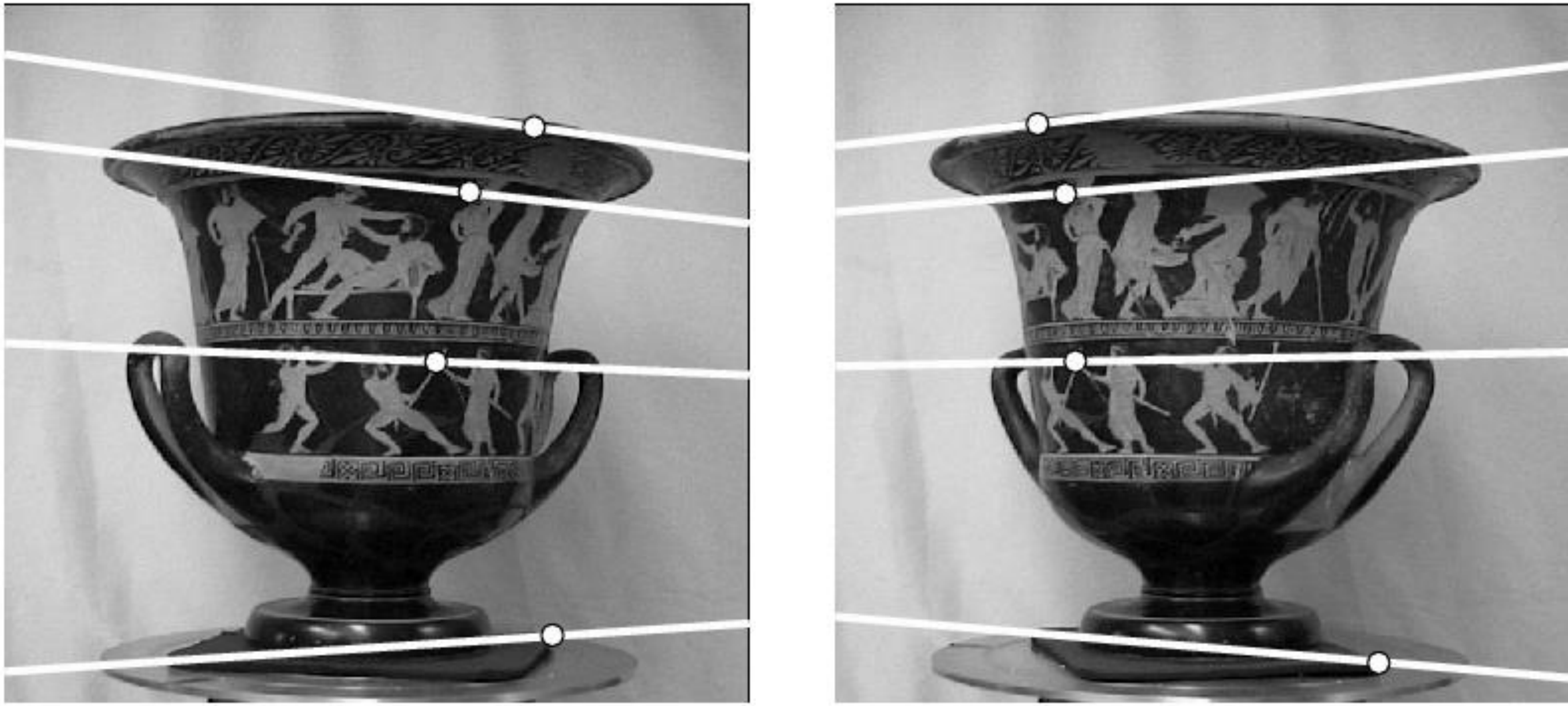
- Epipolar Plane
- Epipolar Lines
- Epipoles
- Baseline

Epipolar constraint

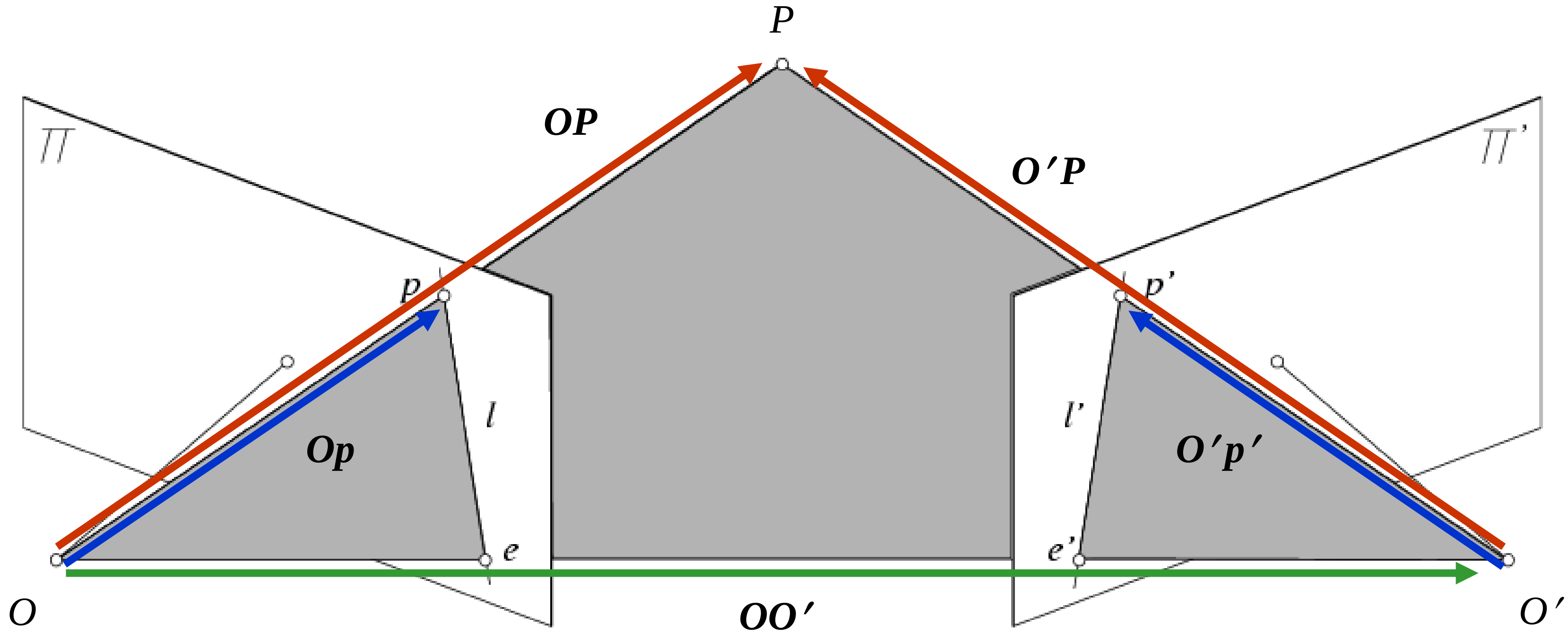


- Potential matches for *p* have to lie on the corresponding epipolar line *l'*
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*

Epipolar lines example



Case 1: Calibrated camera



$\mathbf{Op} \cdot (\mathbf{OO}' \times \mathbf{Op}) = ?$

$\mathbf{Op} \cdot (\mathbf{OO}' \times \mathbf{Op}) = 0$

$$E = [t_\times] R = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} R$$

[R t] – rigid trans. from O to O'

$\mathbf{p} \cdot (\mathbf{t} \times \mathbf{R}\mathbf{p}') = 0$

This can be written in matrix form as:

$\mathbf{p}^T \mathbf{E}\, \mathbf{p}' = 0$

Don't believe it?

$$P_1 = R_{c1\leftarrow c2} P_2 + T_{c1\leftarrow c2}$$

$$Z_1 p_1 = Z_2 R_{c1\leftarrow c2} p_2 + T_{c1\leftarrow c2}$$

$$p_1 = \frac{1}{Z_1}(Z_2 R_{c1\leftarrow c2} p_2 + T_{c1\leftarrow c2})$$

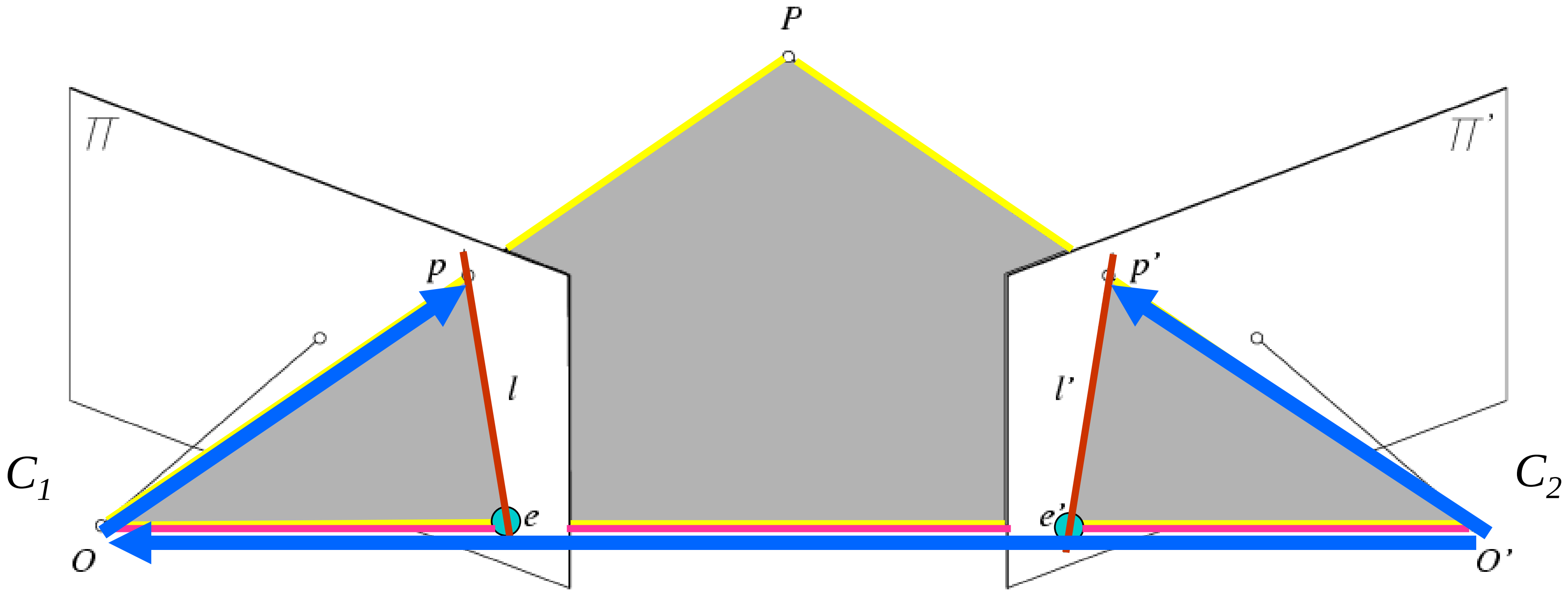
$$T_{c1\leftarrow c2} \times p_1 = \frac{Z_2}{Z_1}(T_{c1\leftarrow c2} \times R_{c1\leftarrow c2} p_2)$$

$$p_1 \cdot (T_{c1\leftarrow c2} \times p_1) = \frac{Z_2}{Z_1} \underline{p_1} \cdot \underline{T_{c1\leftarrow c2}} \times \underline{R_{c1\leftarrow c2} p_2} = 0$$

$$p_1 E p_2 = 0$$

$$E = [T_{c1\leftarrow c2}] R_{c1\leftarrow c2}$$

Calibrated Camera: Essential Matrix



$$O'p' \cdot (O'O \times Op) = 0$$

- Coplanar (3D, regular coordinates)
- Colinear (2D, homogeneous coordinates)

3D Analysis

- What is O in unprimed frame?
  - $[0,0,0]^T$
- What is p in unprimed frame?
  - $[x,y,1]^T$
- What is O in primed frame (or what is the vector O'O)?
  - **RO+T= T**
- What is p in primed frame?
  - **Rp+T**
- What is p' in primed frame?
  - $[x',y',1]^T$

$$O'p' \cdot (O'O \times Op) = 0$$

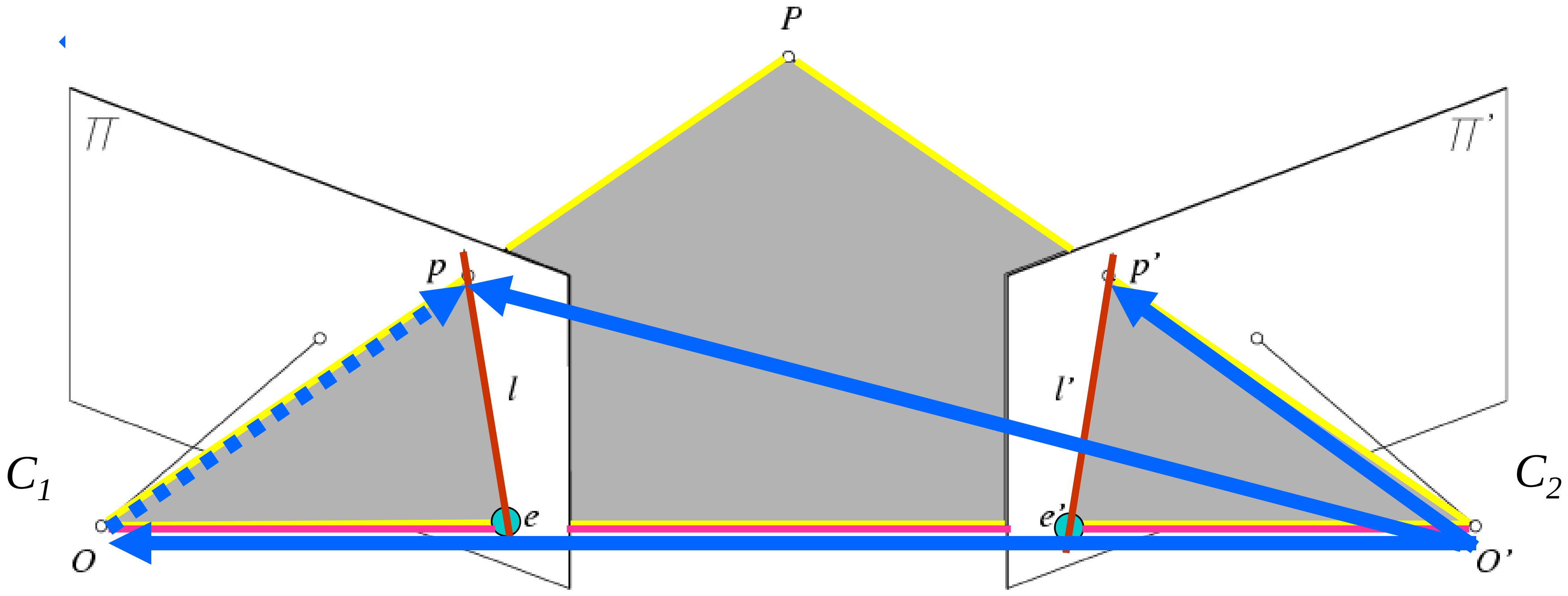
$$p'^T (T \times (Rp + T)) = 0$$

$$p'^T (T \times Rp) = 0$$

$$p'^T \boxed{(T \times R)} p = 0$$

Fundamental matrix

Graphically in 3D



$$O'p' \cdot (O'O \times Op) = 0$$

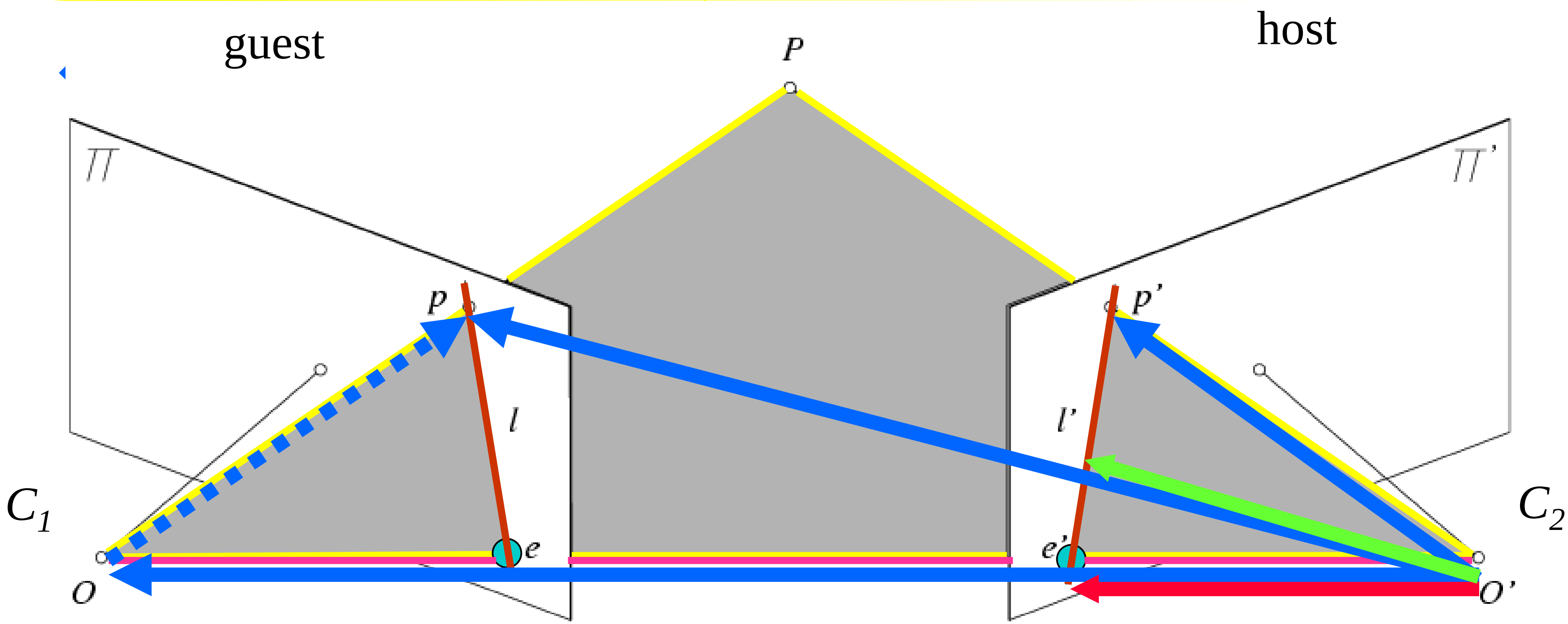
$$O'p' \cdot (O'O \times O'p) = 0$$

All quantities are now expressed in the same coordinate system (O')

2D Analysis

- [x.y,z] can be treated
  - As a 3D regular coordinate (what we did in the previous slide)
  - As a 2D homogeneous coordinate (or x/z and y/z are projections onto the image plane)
- Now O'O is **T**, if it is treated as a 2D homogeneous coordinate, then it is the epipole of the unprimed camera in the prime frame
- Now the image of Op in the primed frame is **Rp+T**
- Hence, **TxRp** is the line equation that passes through the two points (that is exactly the epipolar line!, next slide)
- Hnece, p' is on **TxRp** and **pTxRp=0**

Graphically in 2D

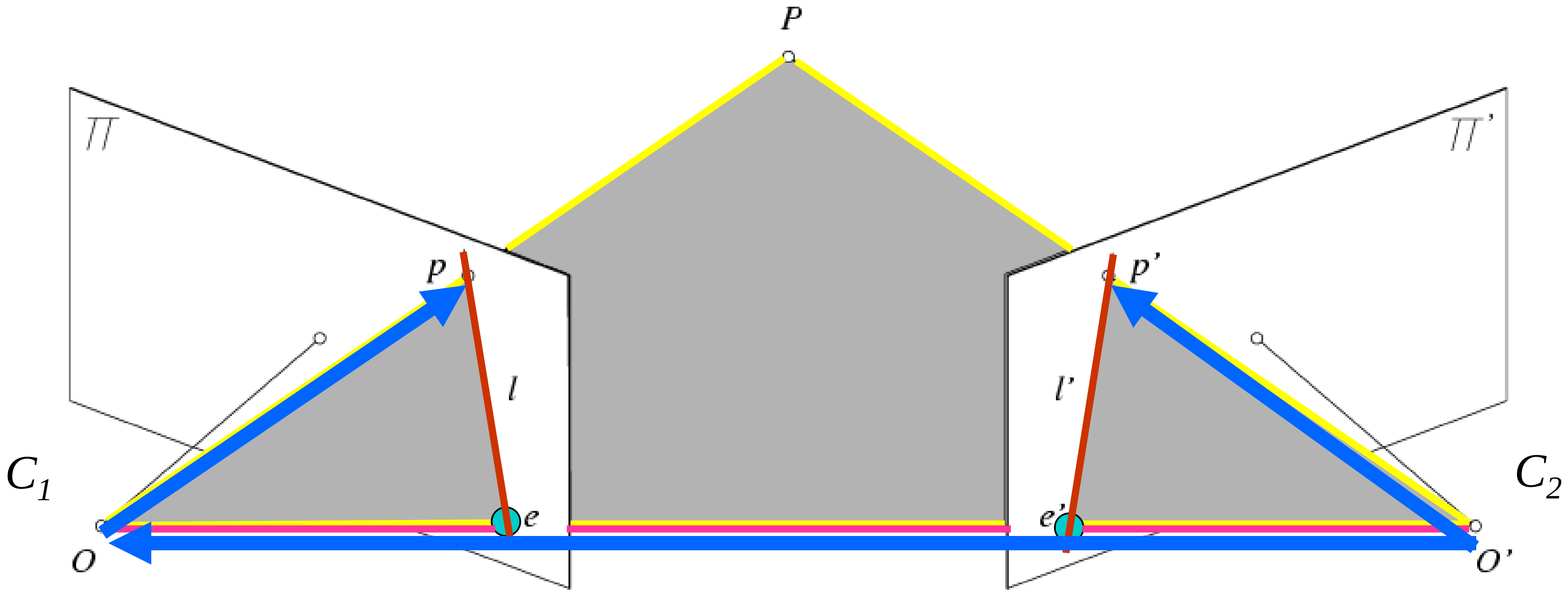


$$O'p' \cdot (O'O \times Op) = 0$$

$$O'p' \cdot (O'O \times O'p) = 0$$

Fp: The line in the host formed by
epipole of the guest in the host
the point of the guest in the host

Physical Meaning #1: Point View



- O'O, Op, O'p' all lie in the same plane (Epipolar plane, spanned by OO'P)
- Hence (O'O x Op) . O'p' = 0
- There are two interpretations:
  - Coplanar (3D, regular coordinates)
  - Colinear (2D, homogeneous coordinates)

The Essential Matrix

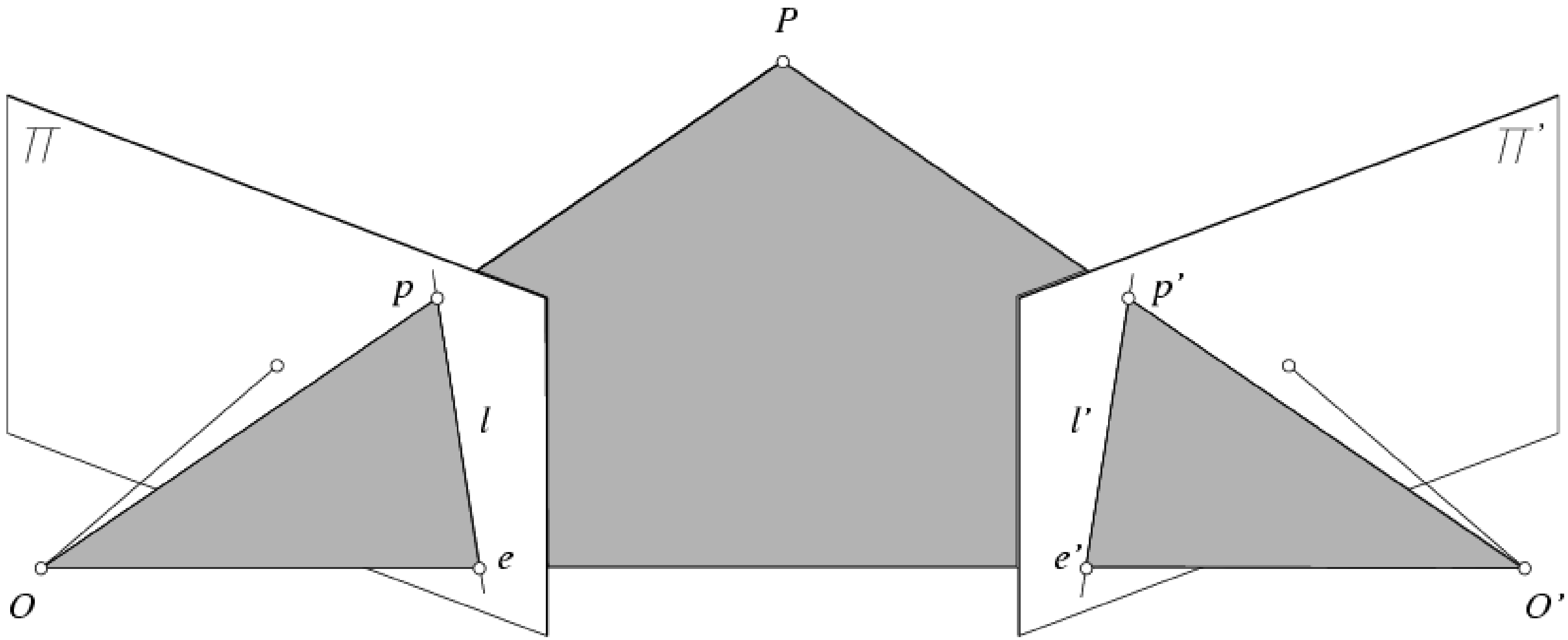
- ***E*** describes the transformation between camera coordinate frames

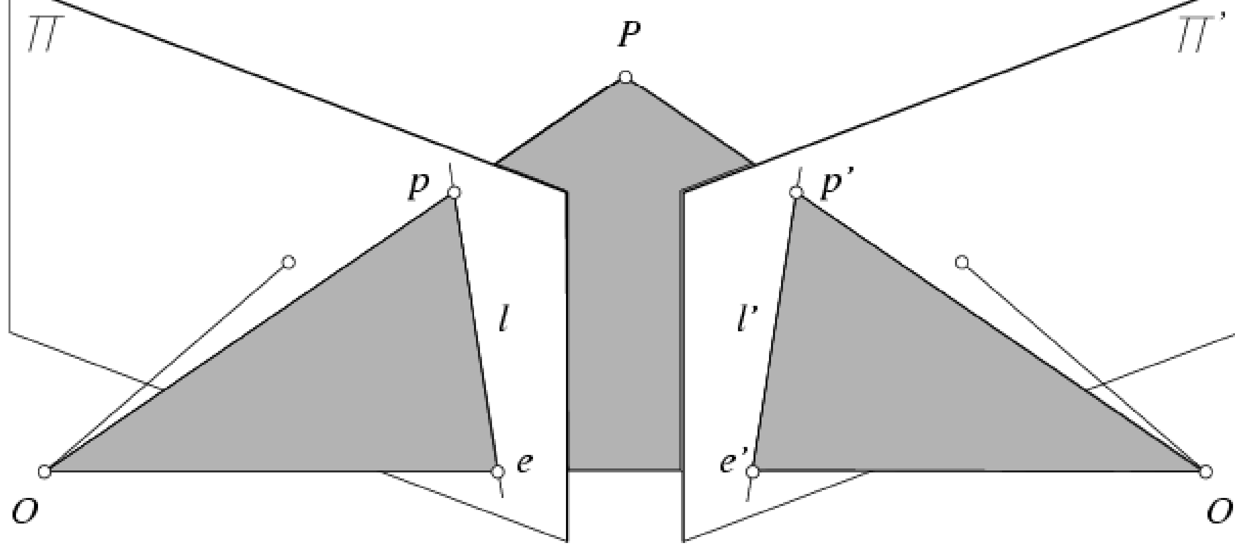
- ***E*** has five degrees of freedom
  - Defined up to a scale factor, since

$$\mathbf{p}^T \mathbf{E} \mathbf{p}' = 0$$

- Why only five?
  - A rigid transformation has <u>six</u> degrees of freedom

- 3 rotation parameters, 2 translation direction parameters
  - Why only translation <u>direction</u>?


P
Π
Π'
p
p'
l
l'
e
e'
O
O'


P
Π
Π'
p
p'
l
l'
e
e'
O
O'

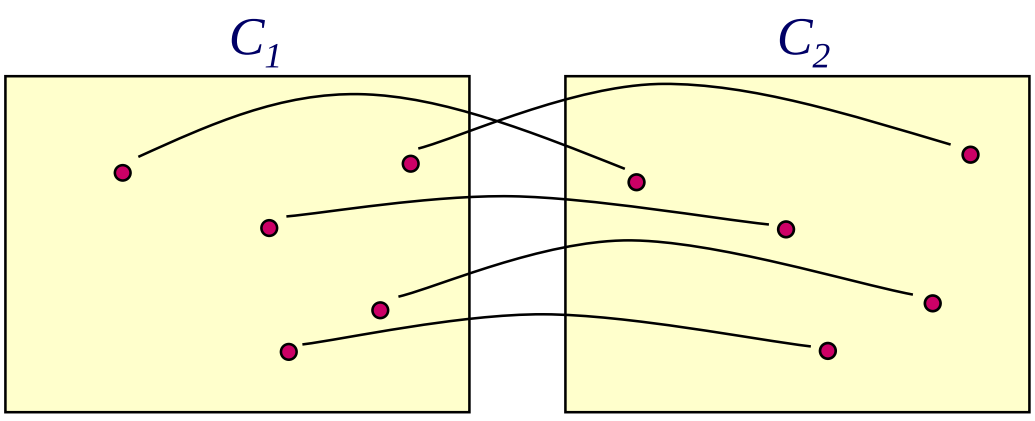
"Up to a scale factor"

- This is always the case with camera calibration and stereo
  - Shrink everything 10x and it all looks the same!

- Typically there is *something* we know that we can use to specify the scale factor
  - E.g., the baseline, the size of an object, the depth of a point/plane

Camera calibration from **E**

- With five unknowns, theoretically we can recover the essential matrix ***E*** by writing $\mathbf{p^T E p' = 0}$ for five corresponding pairs of points
  - 5 equations and 5 unknowns
  - We don't need to know anything about the points (e.g., their depth), only that they project to p_i and p_i'
  - There are, however, limitations…
- This is used for camera calibration (extrinsic parameters)



Direct Solutions of **E**

- Step one:
  - $\mathbf{A}^T\mathbf{A}$ is symmetric and semi-definite
  - It has positive (or zero) eigenvalues
  - The solution corresponds to the eigenvector of the smallest eigenvalue
- Step two:
  - $\mathbf{F} = \mathbf{U\Sigma V}^T$, with (r, s, t) as the singular values in non-increasing order
  - $\mathbf{F'} = \mathbf{U\,\Sigma'\,V}^T$, with (r, r, 0) as the singular values of $\mathbf{\Sigma'}$
    - Zero out the smallest singular value
    - Make the first two eigenvalues the same
- You can also infer essential matrix from fundamental matrix (more later)
- We will present the detailed procedure later with fundamental matrix

Decomposition of **E**

- There are four solutions
- If **t**, **R** is one decomposition (**E**= **t** x **R**)
- **-t** will be ok too (**E** is defined up to a scale factor)
- **UR** will be ok too (**U** is an 180 rotation about **t**)
- So there are four solutions
  - **t, R**
  - **-t, R**
  - **t, UR**
  - **-t, UR**

Why? Algebraic Explanation

- Skew-symmetric matrix has a block diagonal eigenvalue decomposition, for 3x3 matrices, we have

$$\mathbf{S} = \mathbf{U}(\sigma\mathbf{Z})\mathbf{U}^T = \sigma\mathbf{U}\,diag(1,1,0)\mathbf{W}^T\mathbf{U}^T = -\sigma\mathbf{U}\,diag(1,1,0)\mathbf{W}\mathbf{U}^T$$

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -diag(1,1,0)\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = diag(1,1,0)\mathbf{W}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Two ways to match **Z** with diag(1,1,0): **W** or $\mathbf{W}^T$
- E is determined up to a sign and a scale, so both are ok

Why? Algebraic Explanation

- Essential matrix has two identical singular values and a zero singular value (if and only if condition)
- If (->) essential matrix has two identical singular values and a zero singular value

$$\mathbf{E} = \mathbf{TR} = \mathbf{U}(\sigma\mathbf{Z})\mathbf{U}^T\mathbf{R}$$

$$= \mathbf{UZU}^T\mathbf{R}$$

$$= \mathbf{U}diag(1,1,0)(\mathbf{WU}^T\mathbf{R}) = \mathbf{U}diag(1,1,0)(\mathbf{W}^T\mathbf{U}^T\mathbf{R})$$

$$= \mathbf{U}\Sigma V^T$$

Up to a scale factor

Up to a sign

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -diag(1,1,0)\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = diag(1,1,0)\mathbf{W}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Why? Algebraic Explanation

- Essential matrix has two identical singular values and a zero singular value (if and only if condition)
- Only if (<-) a matrix has two identical singular values and a zero singular value is an essential matrix

$$
\begin{aligned}
\mathbf{E} &= \mathbf{U}\,diag(1,1,0)\mathbf{V}^T \\
&= \mathbf{UZW}^T\mathbf{V}^T = \mathbf{UZWV}^T \\
&= \mathbf{UZU}^T\mathbf{UW}^T\mathbf{V}^T = \mathbf{UZU}^T\mathbf{UWV}^T \\
&= (\mathbf{UZU}^T)(\mathbf{UW}^T\mathbf{V}^T) = (\mathbf{UZU}^T)(\mathbf{UWV}^T) \\
&= \mathbf{SR}
\end{aligned}
$$

← Up to a sign

Up to a scale factor

←

Why? Algebraic Explanation

- Given SVD of E as $\mathbf{U}$ diag(1,1,0)$\mathbf{V}^T$, there are two decompositions
  - See previous page for proof
- Given SVD of E as $\mathbf{U}$ diag(1,1,0)$\mathbf{V}^T$, there are four camera matrices
- **P=[I|0]**

An additional rotation about **t**

- $\mathbf{P}' = [\mathbf{UWV}^T|\mathbf{u}_3], [\mathbf{UWV}^T|-\mathbf{u}_3], [\mathbf{UW}^T\mathbf{V}^T|\mathbf{u}_3], [\mathbf{UW}^T\mathbf{V}^T|-\mathbf{u}_3]$

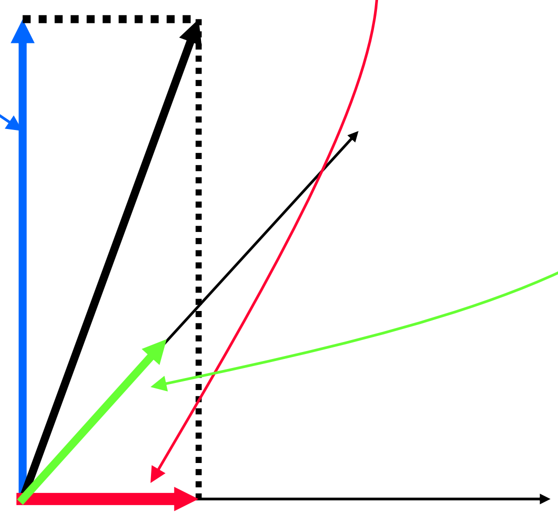
$$[\mathbf{t}]_x = \mathbf{UZU}^T = \begin{bmatrix} | & | & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} - & \mathbf{u}_1{}^T & - \\ - & \mathbf{u}_2{}^T & - \\ - & \mathbf{u}_3{}^T & - \end{bmatrix}$$

$$= \begin{bmatrix} | & | & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} - & \mathbf{u}_2{}^T & - \\ - & -\mathbf{u}_1{}^T & - \\ - & \mathbf{0} & - \end{bmatrix}$$

$$[\mathbf{t}]_x \mathbf{t} = 0 \Rightarrow \mathbf{t} = \mathbf{u}_3$$

Why? Physical Interpretation

$$R_{\mathbf{n}}(\theta) = \mathbf{n}\mathbf{n}^T + (\mathbf{I} - \mathbf{n}\mathbf{n}^T)\cos\theta + [\mathbf{n}_x]\sin\theta$$



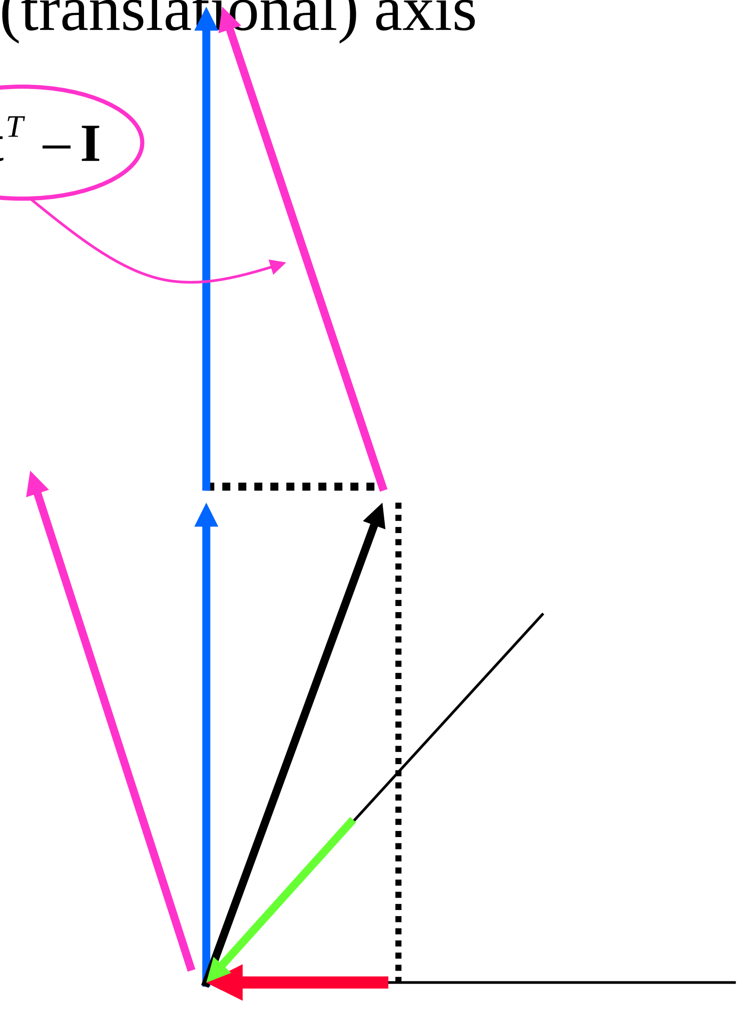
- A useful formula for rotation, a vector after rotation is made of three components
  - Component in the direction of n (no change)
  - Component perpendicular to the direction of n
    - Cos(θ) in the projected direction
    - Sin(θ) in the projected + 90^o direction

Physical Interpretation

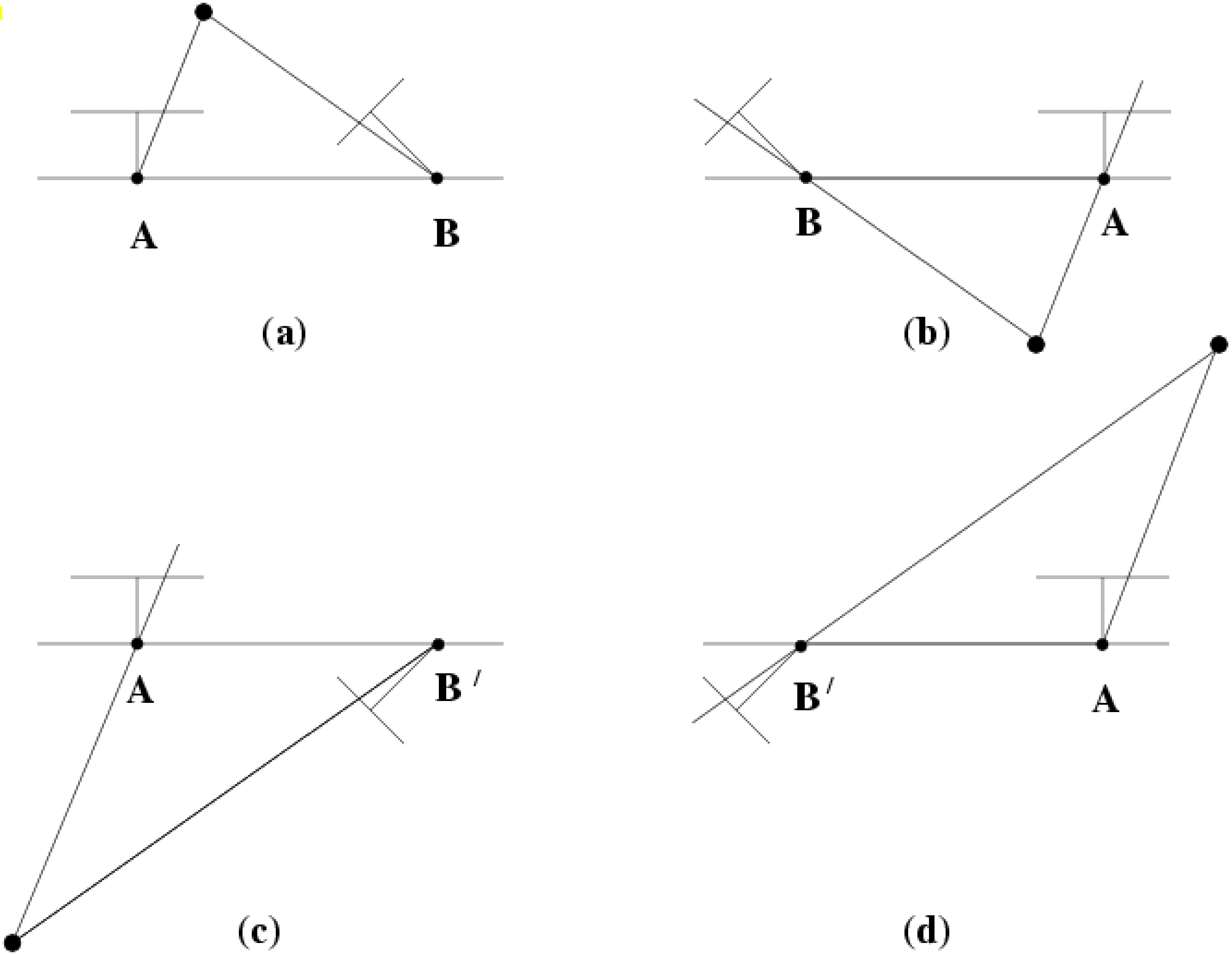
❖ Additional rotation of 180° about **t** (translational) axis

$$\mathbf{U} = \mathbf{R_t}(\pi) = \mathbf{tt}^T + (\mathbf{I} - \mathbf{tt}^T)\cos\pi + [\mathbf{t}_x]\sin\pi = 2\mathbf{tt}^T - \mathbf{I}$$

$$\mathbf{t} \times (2\mathbf{tt}^T - \mathbf{I}) = -\mathbf{t}$$



Four possible reconstructions from **E**



(only one solution where points is in front of both cameras)

Case 2: Uncalibrated camera

- Intrinsic parameters not known

Points in the *normalized* image plane

$$p = K_1 \hat{p}$$

$$p' = K_2 \hat{p}'$$

$$K = \begin{bmatrix} \alpha & -\alpha \cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$(K_1^{-1} p)^T E (K_2^{-1} p') = 0$$

$$(p^T K_1^{-T}) E (K_2^{-1} p') = 0$$

$$p^T F p' = 0$$

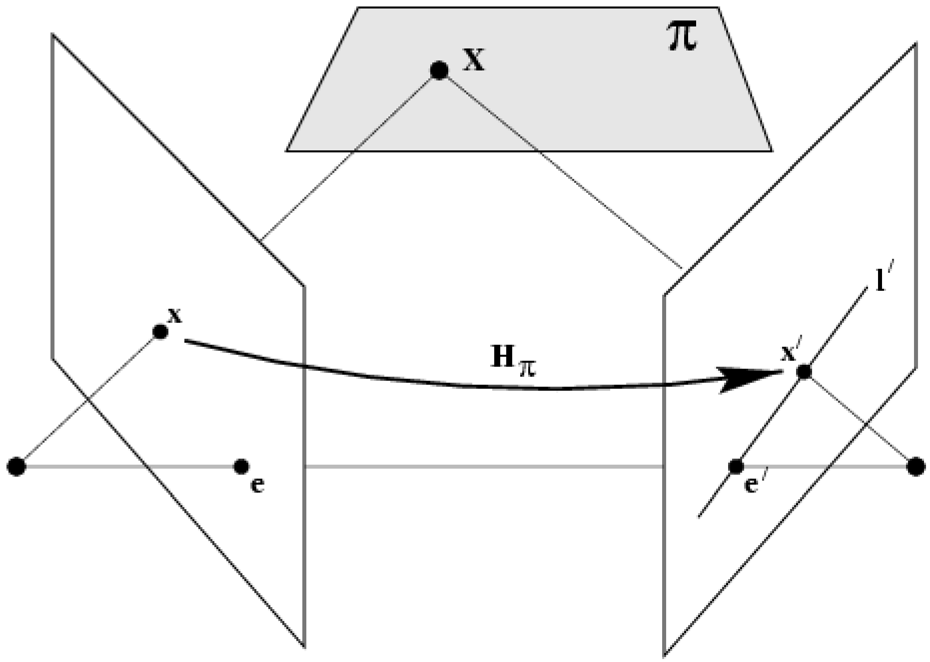
$$F = K_1^{-T} E K_2^{-1}$$

Fundamental Matrix

The Problem

- Given projection matrices of two views to find fundamental matrix is unique
- Given fundamental matrix to find projection matrices of two views are *not* unique
  - **Theorem**, F is the fundamental matrix of multiple two-view projection matrices if and only if the multiple interpretations are related by a projective transform
- The reconstruction is up to a projective transform
- That is, not many property can be measured except incidence and collinearity
- Again, the importance of calibration cannot be overstated

Physical Meaning #2: Plane View



$$x' = H_\pi x$$

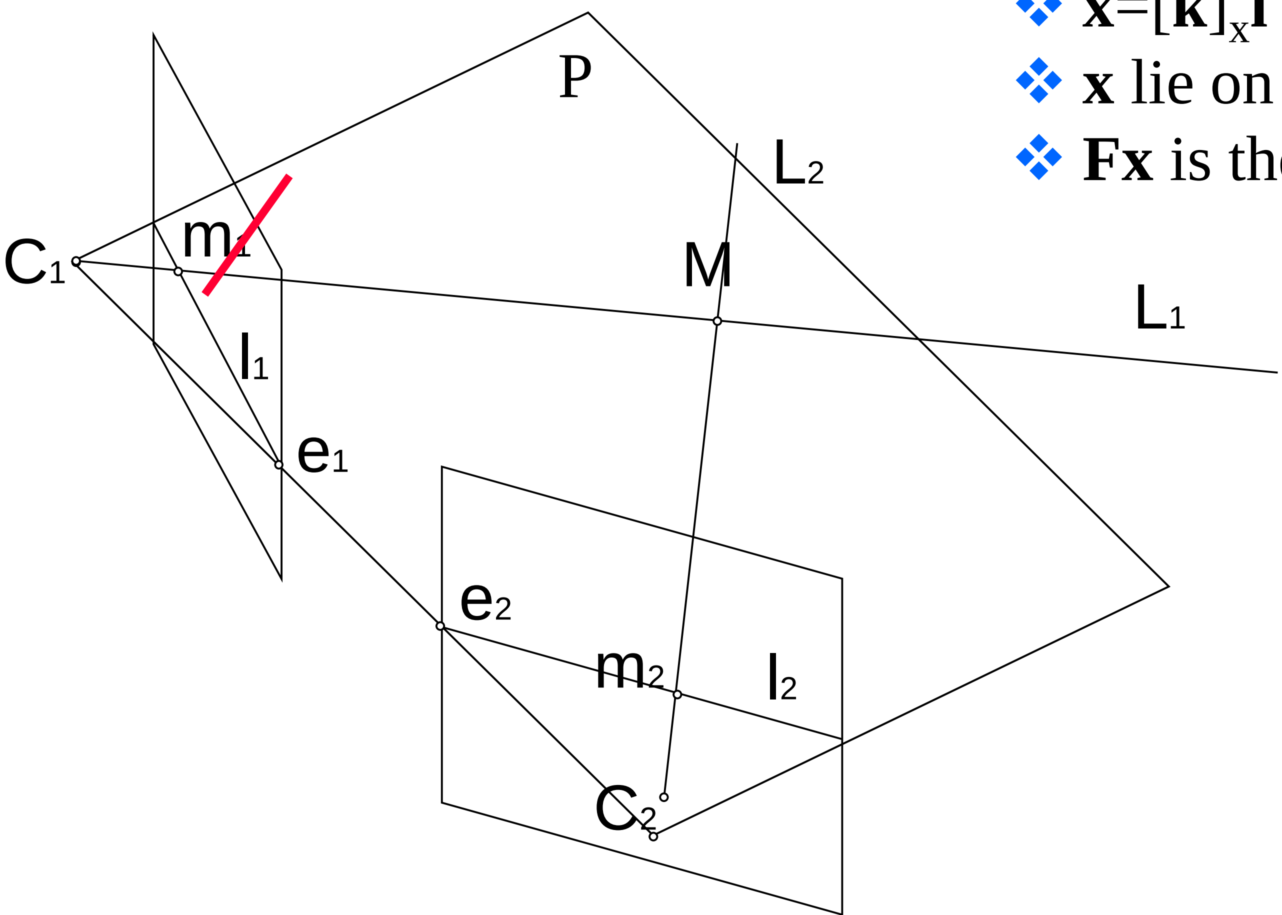
$$l' = e' \times x' = [e']_\times H_\pi x = Fx$$

mapping from 2-D to 1-D family (rank 2)

Physical Meaning #3: Line View

- Take an arbitrary line **k** not passing through **e** (the epipole)

$$\mathbf{l}_2 = \mathbf{F}[\mathbf{k}]_x \mathbf{l}_1$$



- $\mathbf{x}=[\mathbf{k}]_x\mathbf{l}$ is the intersection
- **x** lie on **l**
- **Fx** is then the epipolar line

Direct Derivation

- Given a point
- Back project it into space
- Project 3D point into the second frame
- Form the line with epipole

$$\mathbf{x}(\lambda) = \mathbf{P}^{+}\mathbf{x} + \lambda\mathbf{C}$$

$$\Rightarrow \mathbf{x}'(\lambda) = \mathbf{P}'(\mathbf{P}^{+}\mathbf{x} + \lambda\mathbf{C})$$

$$\Rightarrow \mathbf{l}' = [\mathbf{e}']_{x}\mathbf{P}'(\mathbf{P}^{+}\mathbf{x} + \lambda\mathbf{C}) = [\mathbf{e}']_{x}\mathbf{P}'\mathbf{P}^{+}\mathbf{x} = \mathbf{F}\mathbf{x}$$

$$\Rightarrow \mathbf{F} = [\mathbf{e}']_{x}\mathbf{P}'\mathbf{P}^{+}$$

Confusion

- Q: There are so many ways that one can derive fundamental matrix, are all derivations give the same results?
- A: Yes and no
- Yes: They all serve the same purpose (defining a line from a point to epipole)
- No: There can be multiple matrices (line representation is not unique)

The Fundamental Matrix

- ***F*** has seven independent parameters
- A simple, linear technique to recover ***F*** from corresponding point locations is the “eight point algorithm”
- From ***F***, we can recover the epipolar geometry of the cameras
  - Not saying how…
- This is called *weak calibration*

The eight-point algorithm

$$p^T F \, p' = 0$$

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

$$\mathbf{AF} = \mathbf{0}$$

Detailed Algorithm

- First Important Observation:
  - **A** is rank deficient (its null space contains more than zero)
  - In fact, **A** has rank of 8, hence there is a unique solution (up to a scale factor)
- Second Important Observation:
  - The fundamental matrix is rank deficient (it is 3x3 of rank 2)

Solutions

- Step one:
  - $\mathbf{A}^T\mathbf{A}$ is symmetric and semi-definite
  - It has positive (or zero) eigen values
  - The solution corresponds to the eigen vector of the smallest eigen value
  - Hint: use Lagrange multiplier, similar procedure as shown in the camera calibration slides
- Step two:
  - $\mathbf{E} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, with (r, s, t) as the singular values in non-increasing order
  - $\mathbf{E}' = \mathbf{U}\ \boldsymbol{\Sigma}\ '\mathbf{V}^T$, with (r, s, 0) as the singular values of $\boldsymbol{\Sigma}\ '$
    - Zero out the smallest singular value

Don't Believe It?

$$e = \|\mathbf{AF}\|^2 + \lambda(1 - \|\mathbf{F}\|^2)$$

$$\frac{\partial e}{\partial \mathbf{F}} = \mathbf{A}^T\mathbf{AF} - \lambda\mathbf{F} = 0$$

$$\Rightarrow \mathbf{A}^T\mathbf{AF} = \lambda\mathbf{F}$$

F is the eigen vector of $A^T A$

$$e = \|\mathbf{AF}\|^2 + \lambda(1 - \|\mathbf{F}\|^2)$$

$$= \mathbf{F}^T\mathbf{A}^T\mathbf{AF} + \lambda - \lambda\mathbf{F}^T\mathbf{F}$$

$$= \lambda\mathbf{F}^T\mathbf{F} + \lambda - \lambda\mathbf{F}^T\mathbf{F} = \lambda$$

Error is $\boldsymbol{\lambda}$

F is the eigen vector of $A^T A$

With the smallest eigen value

The eight-point algorithm

$$p^T F \, p' = 0$$

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

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

$$\mathbf{AX} = \mathbf{0}$$

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

Invert and solve for F

Least squares approach

If $n > 8$

Minimize:

$$\sum_{i=1}^{n} (\boldsymbol{p}_i^T \mathcal{F} \boldsymbol{p}_i')^2$$

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

Nonlinear least-squares approach

Point in image 1

Epipolar line in image 1 caused by p′

Minimize

$$\sum_{i=1}^{n}[\mathrm{d}^2(\boldsymbol{p}_i, \mathcal{F}\boldsymbol{p}'_i) + \mathrm{d}^2(\boldsymbol{p}'_i, \mathcal{F}^T\boldsymbol{p}_i)]$$

with respect to the coefficients of $\mathcal{F}$

Nonlinear – initialize it from the results of the eight-point algorithm

Least squares 8-point algorithm

(a)

Hartley's normalized 8-point alg.

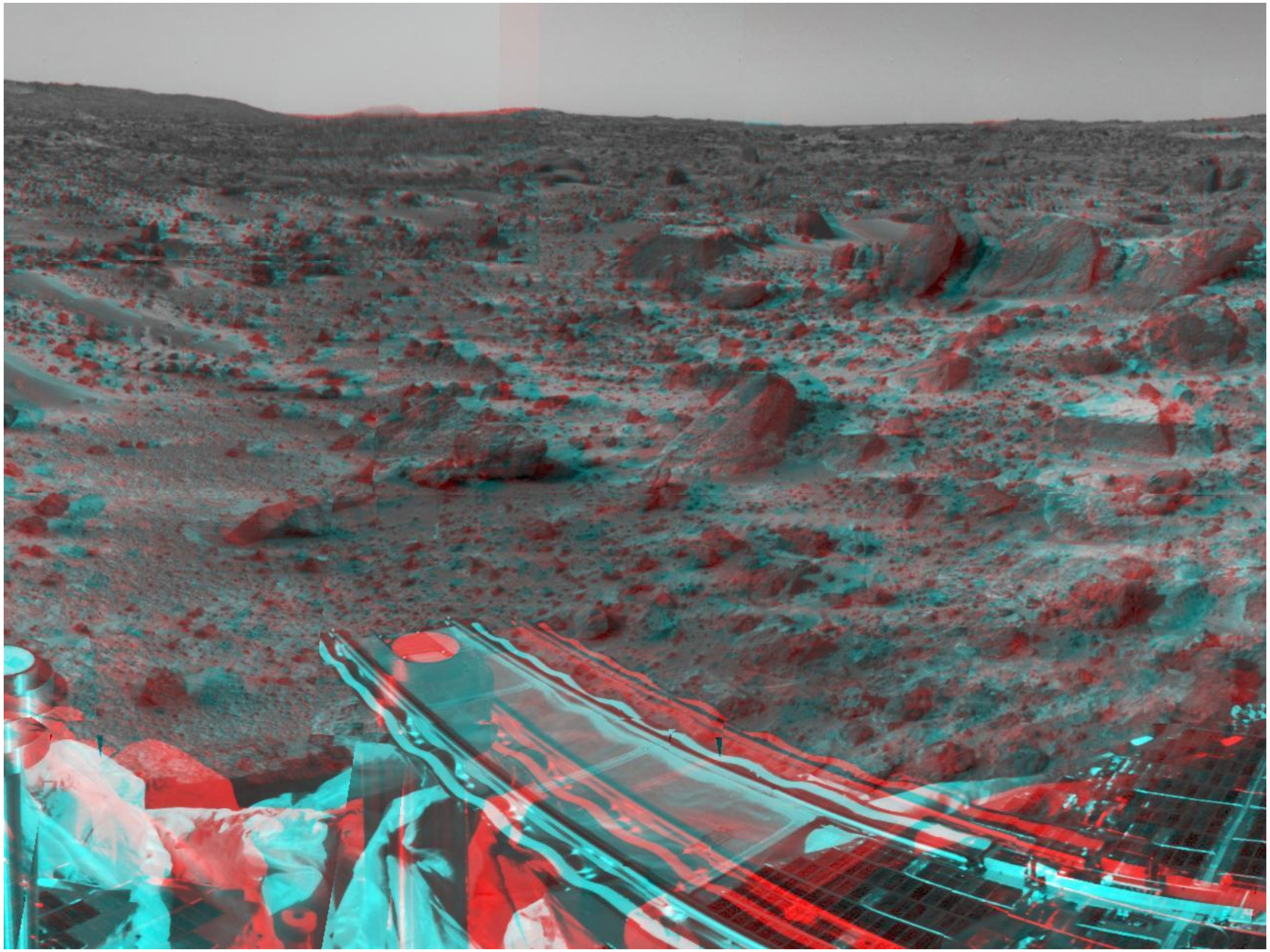
(b)

	Linear Least Squares	(Hartley, 1995)	(Luong et al., 1993)
Av. Dist.	2.33 pixels	0.92 pixels	0.86 pixels

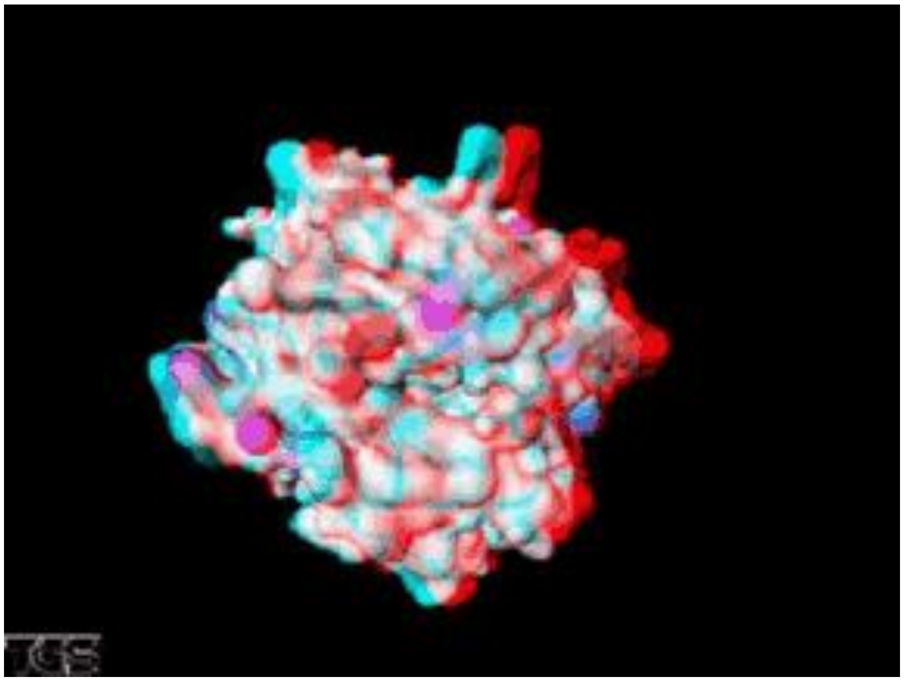
Figure 10-4

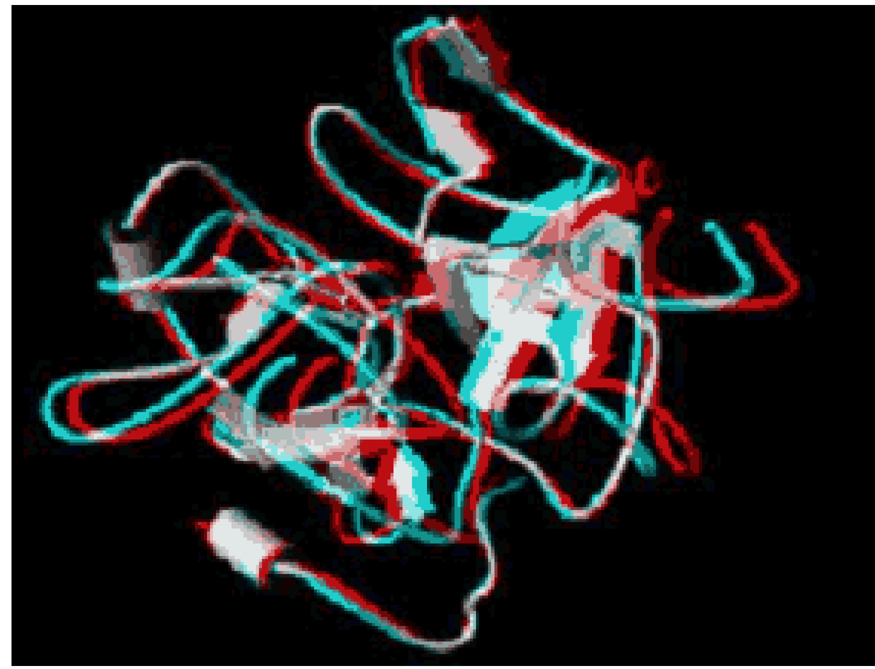
Weak-calibration experiment using 37 point correspondences between two images of a toy house. The figure shows the epipolar lines found by (a) the least-squares version of the eight-point algorithm, and (b) the normalized variant of this method proposed by Hartley (1995). Note, for example, the much larger error in (a) for the feature point close to the bottom of the mug. Quantitative comparisons are given in the table, where the average distances between the data points and corresponding epipolar lines are shown for both techniques as well as the nonlinear algorithm of Luong *et al.* (1993). *Data courtesy of Boubakeur Boufama and Roger Mohr.*

Red/Green stereo display



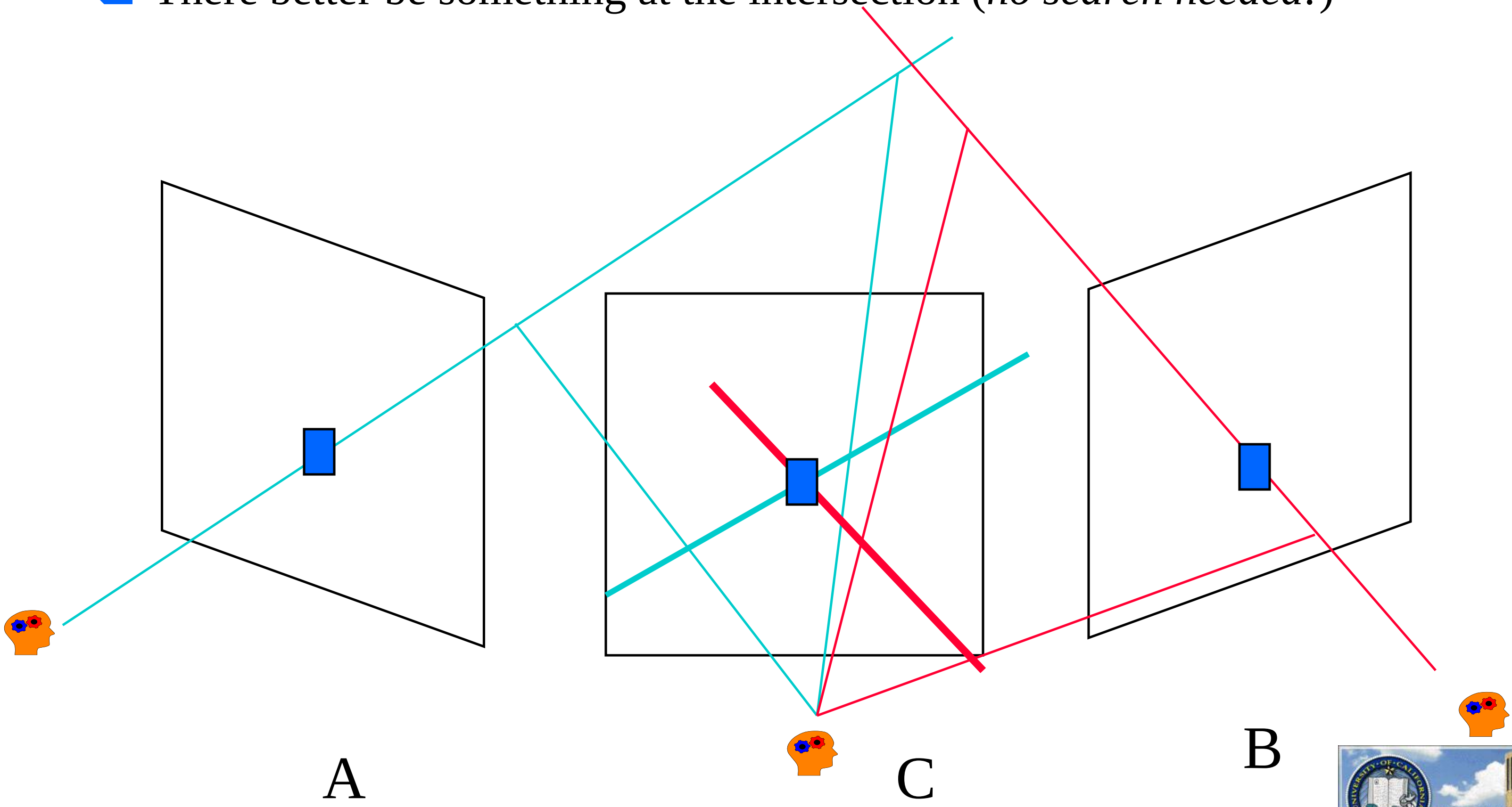
From Mars Pathfinder





Three Camera Stereo

- A powerful way of eliminate spurious matches
  - Hypothesize matches between A & B
  - Matches between A & C on green epipolar line
  - Matches between B & C on red epipolar line
  - There better be something at the intersection (*no search needed!*)



Mathematically

- Given two corresponding points p1 and p2 in views 1 and 2, the point p3 in the third view of the point P of intersection of the optical ray of p1 and p2 is

$$\mathbf{p}_3 = \mathbf{F}_{13}\mathbf{p}_1 \times \mathbf{F}_{23}\mathbf{p}_2$$

- Why? ($\mathbf{F}_{guest,host}$)
  - $\mathbf{F}_{13}\mathbf{p}_1$ is the epipole line of p from 1st frame in 3rd frame
  - $\mathbf{F}_{23}\mathbf{p}_2$ is the epipole line of p from 2nd frame in 3rd frame

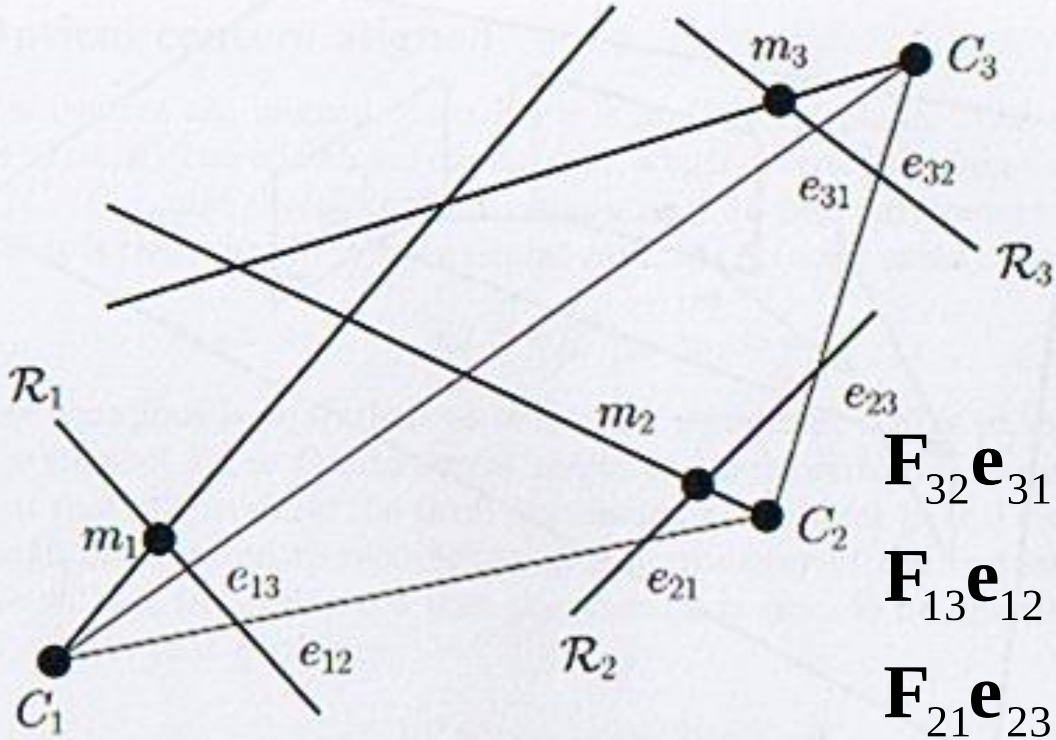
Special Cases – Many

- If p corresponds to epipole, then there is no epipolar line
- If the optical centers are colinear, epipolar lines will coincide and intersect everywhere
  - If you mount the camera on a translational stage without rotation, the three optical centers will be aligned (colinear). More views do not help
- If the optical centers are not colinear and P is in the trifocal plane (the plane formed by O1, O2 and O3), the same as above
- More problems
  - Given point correspondences in three views, the above equation is no longer linear in terms of the two fundamental matrices

Multiple camera stereo

- Using multiple camera in stereo has advantages and disadvantages
- Some disadvantages
  - Computationally more expensive
  - More correspondence matching issues
  - More hardware ($)
- Some advantages
  - Extra view(s) reduces ambiguity in matching
  - Wider range of view, fewer “holes”
  - Better noise properties
  - Increased depth precision

Trifocal Geometry



$$\mathbf{F}_{32}\mathbf{e}_{31} \propto \mathbf{e}_{21} \times \mathbf{e}_{23} \propto \mathbf{t}_2$$

$$\mathbf{F}_{13}\mathbf{e}_{12} \propto \mathbf{e}_{32} \times \mathbf{e}_{31} \propto \mathbf{t}_3$$

$$\mathbf{F}_{21}\mathbf{e}_{23} \propto \mathbf{e}_{13} \times \mathbf{e}_{12} \propto \mathbf{t}_1$$

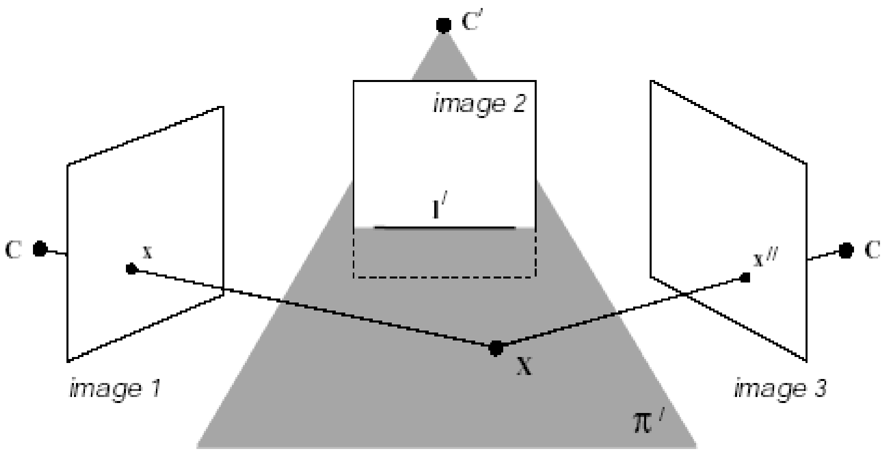
Figure 8.5: The three points m_1, m_2 and m_3 belong to the three Trifocal lines (e_{12}, e_{13}), (e_{23}, e_{21}), (e_{31}, e_{32}): They satisfy the equations (8.1) but are not the images of a single 3D point.

Mathematically

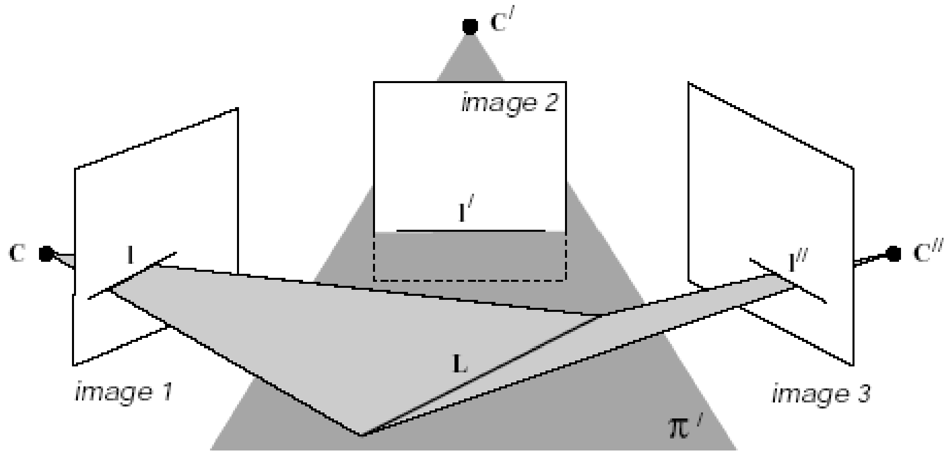
- Mathematically, trifocal geometry is formulated in terms of trifocal tensor expression
- Two popular formulations (among many) involve
  - All lines:
    - From two views, back project the lines into planes
    - Two planes intersect in space into a line
    - Project that line into the third view, and it should be the same line as in the third view
  - A point in one and lines in the other two:
    - From two views with lines, back project the lines into planes
    - Two planes intersect in space into a line
    - Project that line into the third view, and the point should lie in that projected line

Geometrically

- A planar homography can be established by a line in image 2 (or 1, 3) for features in images 1 and 3 (or 2 and 3, 1 and 2)



Point transfer



Line transfer

Sidebar: 2D line & 3D plane

- Given
  - A line $\mathbf{l}=[a,b,c]^T$ in image
  - A projection matrix $\mathbf{P}$, with $\mathbf{U}^T,\mathbf{V}^T,\mathbf{W}^T$ as its three rows, or $\mathbf{P}^T=[\mathbf{U},\mathbf{V},\mathbf{W}]$
- Then the space plane whose image is $\mathbf{l}$ under $\mathbf{P}$ is $a\mathbf{U}+b\mathbf{V}+c\mathbf{W}$ or $\mathbf{P}^T\mathbf{l}$
- Any point $\mathbf{M}$ that is on the plane satisfy the plane equation, and hence, the projection satisfies the line equation

$$(a\mathbf{U}^T + b\mathbf{V}^T + c\mathbf{W}^T)\mathbf{M} = 0$$

$$[a,b,c]\cdot[\mathbf{U}^T\mathbf{M}, \mathbf{V}^T\mathbf{M}, \mathbf{W}^T\mathbf{M}] = 0$$

$$[a,b,c]\cdot\begin{bmatrix}\mathbf{U}^T\\ \mathbf{V}^T\\ \mathbf{W}^T\end{bmatrix}\mathbf{M} = 0$$

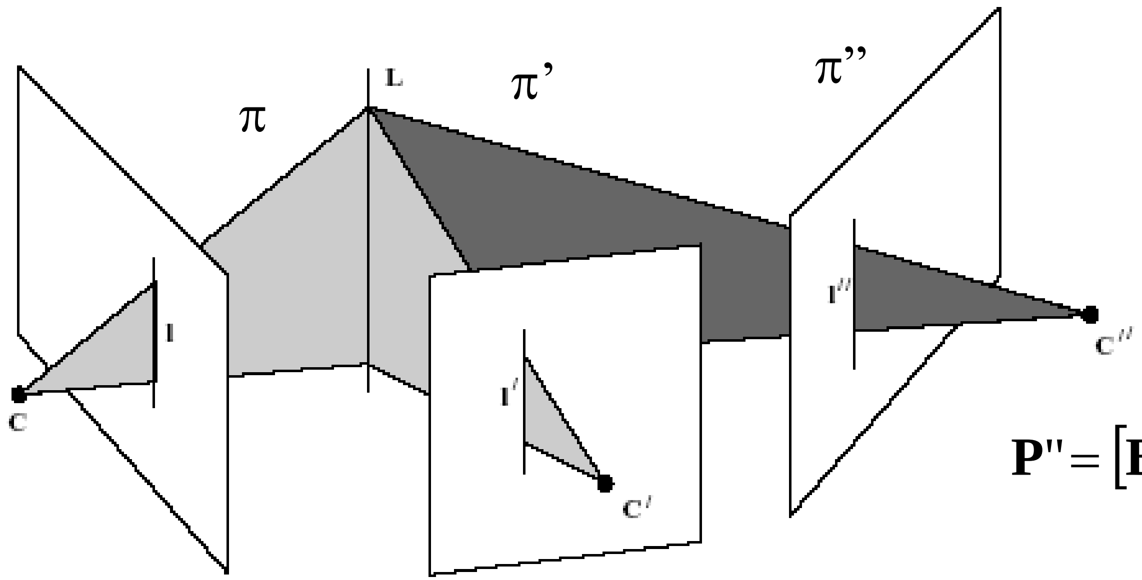
$$[a,b,c]\cdot\mathbf{PM} = 0$$

Sidebar: 3D Line Equation

- Plane equations
  - $\mathbf{N_1} \cdot \mathbf{p} = d_1$
  - $\mathbf{N_2} \cdot \mathbf{p} = d_2$
- Line equation
  - $\mathbf{l} = c_1 \mathbf{N_1} + c_2 \mathbf{N_2} + t\, \mathbf{N_1} * \mathbf{N_2}$
- Solving for c_1 and c_2
  - $\mathbf{N_1} \cdot \mathbf{l} = d_1 = c_1 \mathbf{N_1} \cdot \mathbf{N_1} + c_2 \mathbf{N_1} \cdot \mathbf{N_2}$
  - $\mathbf{N_1} \cdot \mathbf{l} = d_2 = c_1 \mathbf{N_1} \cdot \mathbf{N_2} + c_2 \mathbf{N_2} \cdot \mathbf{N_2}$
  - **$c_1 = (d_1 N_2 \cdot N_2 - d_2 N_1 \cdot N_2) / \text{determinant}$**
  - **$c2 = (d_2 N_1 \cdot N_1 - d_1 N_1 \cdot N_2) / \text{determinant}$**
  - **$\text{determinant} = (N_1 \cdot N_1) (N_2 \cdot N_2) - (N_1 \cdot N_2)^2$**

Detail on Trifocal Tensor

- For three lines
  - Warning: We are not using tensor notation here. Instead, we use matrix-vector notation that is more readily accessible to most people



$$\mathbf{P}'' = [\mathbf{B} \mid \mathbf{b}_4]$$

$$\mathbf{P}' = [\mathbf{A} \mid \mathbf{a}_4]$$

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$$

$$\pi = \mathtt{P}^{\mathsf{T}}\mathbf{l} = \begin{pmatrix} \mathbf{l} \\ 0 \end{pmatrix} \quad \pi' = \mathtt{P}'^{\mathsf{T}}\mathbf{l}' = \begin{pmatrix} \mathtt{A}^{\mathsf{T}}\mathbf{l}' \\ \mathbf{a}_4^{\mathsf{T}}\mathbf{l}' \end{pmatrix} \quad \pi'' = \mathtt{P}''^{\mathsf{T}}\mathbf{l}'' = \begin{pmatrix} \mathtt{B}^{\mathsf{T}}\mathbf{l}'' \\ \mathbf{b}_4^{\mathsf{T}}\mathbf{l}'' \end{pmatrix}$$

University of California Santa Barbara

Detail on Trifocal Tensor (cont.)

- M is 3x3, but has only one independent column

$$\mathtt{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] = \begin{bmatrix} \mathbf{l} & \mathtt{A}^\top \mathbf{l}' & \mathtt{B}^\top \mathbf{l}'' \\ 0 & \mathbf{a}_4^\top \mathbf{l}' & \mathbf{b}_4^\top \mathbf{l}'' \end{bmatrix}$$

$$\mathbf{m}_1 = \alpha \mathbf{m}_2 + \beta \mathbf{m}_3 \qquad \alpha = k(\mathbf{b}_4^\top \mathbf{l}'') \text{ and } \beta = -k(\mathbf{a}_4^\top \mathbf{l}')$$

$$\mathbf{l} = (\mathbf{b}_4^\top \mathbf{l}'')\mathtt{A}^\top \mathbf{l}' - (\mathbf{a}_4^\top \mathbf{l}')\mathtt{B}^\top \mathbf{l}'' = (\mathbf{l}''^\top \mathbf{b}_4)\mathtt{A}^\top \mathbf{l}' - (\mathbf{l}'^\top \mathbf{a}_4)\mathtt{B}^\top \mathbf{l}''$$

$$l_i = \mathbf{l}''^\top (\mathbf{b}_4 \mathbf{a}_i^\top)\mathbf{l}' - \mathbf{l}'^\top (\mathbf{a}_4 \mathbf{b}_i^\top)\mathbf{l}'' = \mathbf{l}'^\top (\mathbf{a}_i \mathbf{b}_4^\top)\mathbf{l}'' - \mathbf{l}'^\top (\mathbf{a}_4 \mathbf{b}_i^\top)\mathbf{l}''$$

$$l_i = \mathbf{l}'^\top \mathtt{T}_i \mathbf{l}''. \qquad \mathtt{T}_i = \mathbf{a}_i \mathbf{b}_4^\top - \mathbf{a}_4 \mathbf{b}_i^\top$$

$$\mathbf{l}^\top = \mathbf{l}'^\top [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3] \mathbf{l}''$$

Important Observations

- Similar to fundamental matrix **F**
  - The expressions are linear in **T**
  - Expressed in term of image observables (line orientation)
  - Given enough correspondences, we can solved for trifocal tensors
  - Then we can compute fundamental matrices and projection matrices from trifocal tensors

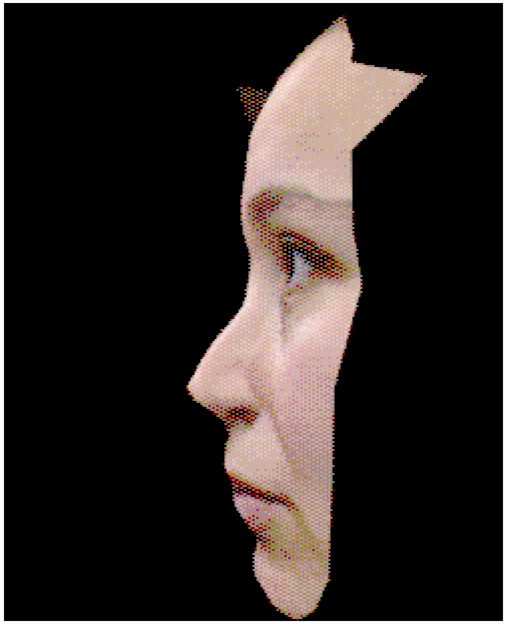
Multiple Views (>3)

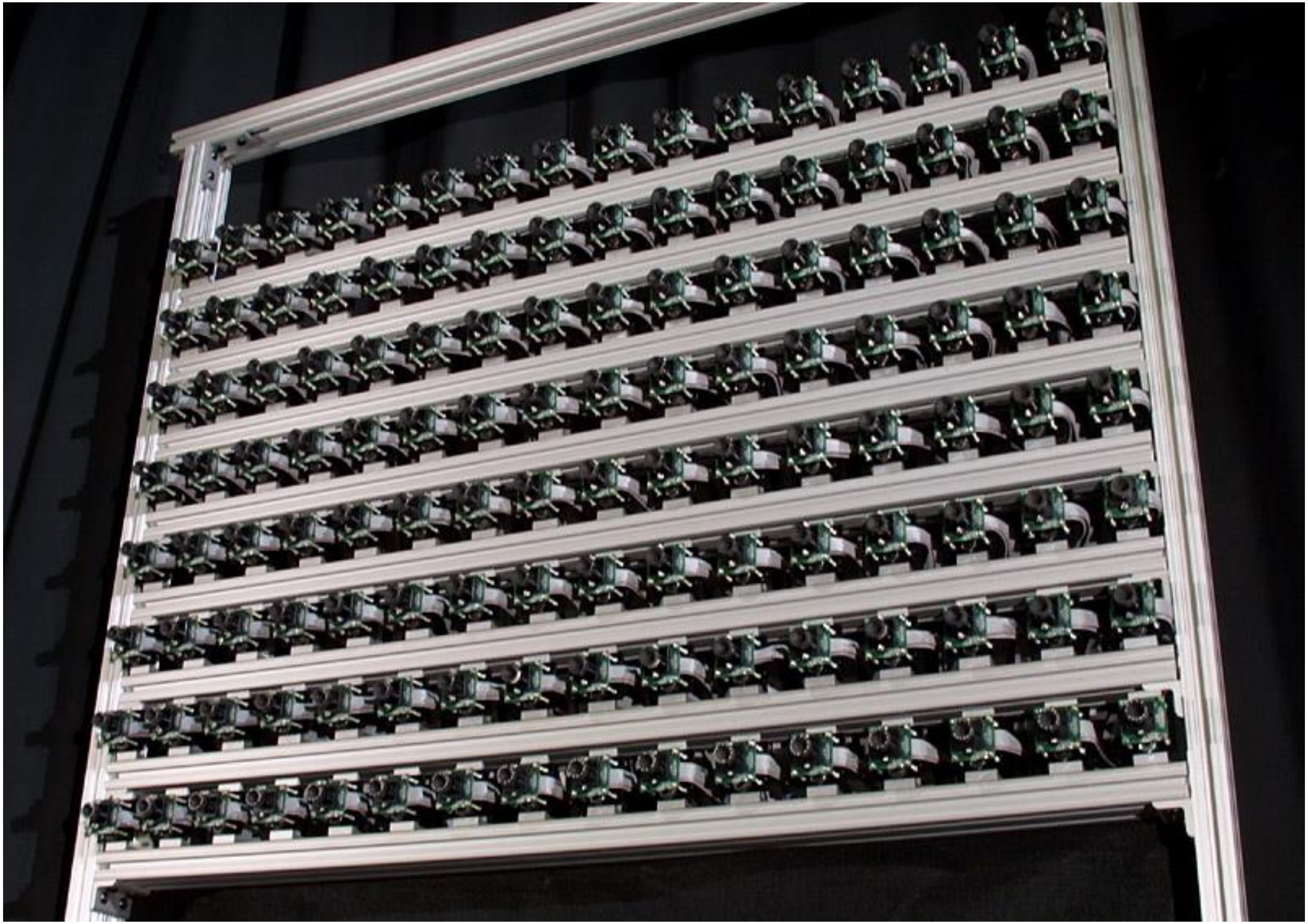
- Math becomes really involved
- In fact, quadrifocal tensor does not provide new information beyond trifocal tensor (for 3 views) + fundamental matrix (for 2 views)
- When the projection model is parallel, there is an elegant formulation based on factorization
- When the projection model is perspective, factorization does not generalize well
- The common approach:
  - Local: 2 views (fundamental matrix) or 3 views (trifocal tensor)
  - Global: bundle adjustment

Example: Four views

Input images

Texture input

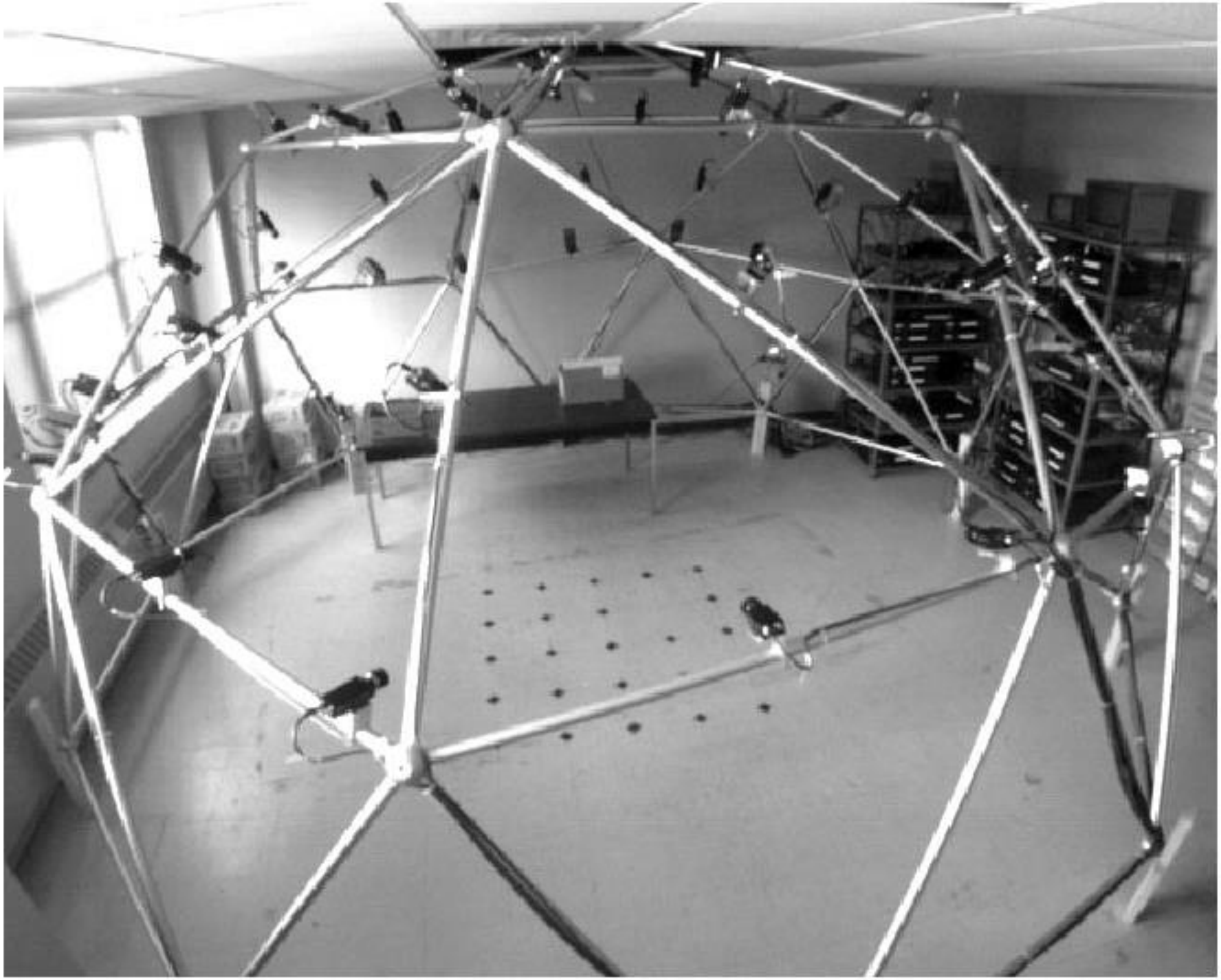




The Stanford Multi-Camera Array

128 CMOS cameras, 2” baseline

5x5 racks version: 125 CMOS cameras, 9" baseline
4 capture PCs, 4 electronics racks (1 board per camera)



CMU multi-camera stereo

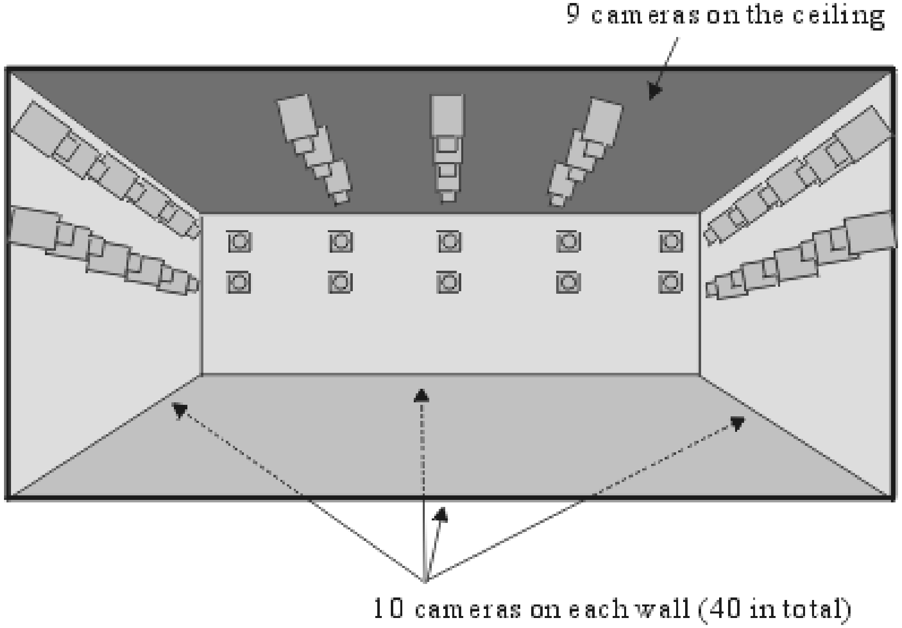
51 video cameras mounted on a 5-meter diameter geodesic dome

Video 1

Video 2

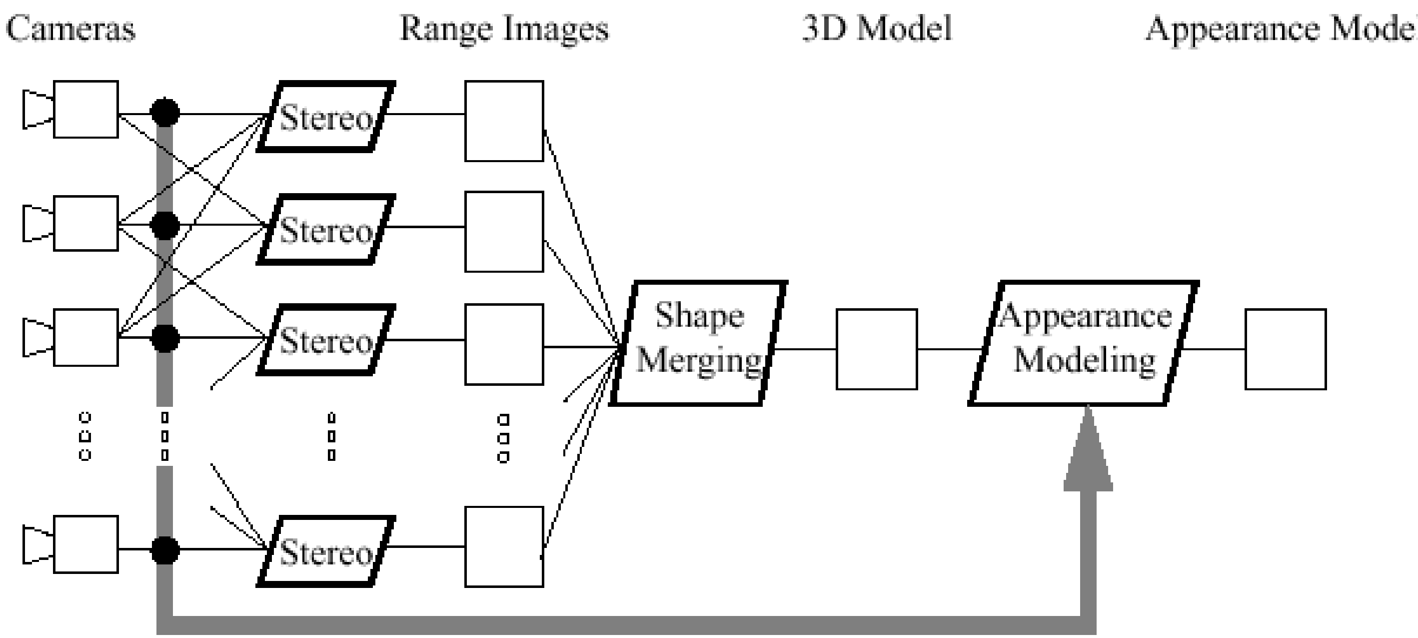
Video 3

Virtualized Reality: CMU 3D Room



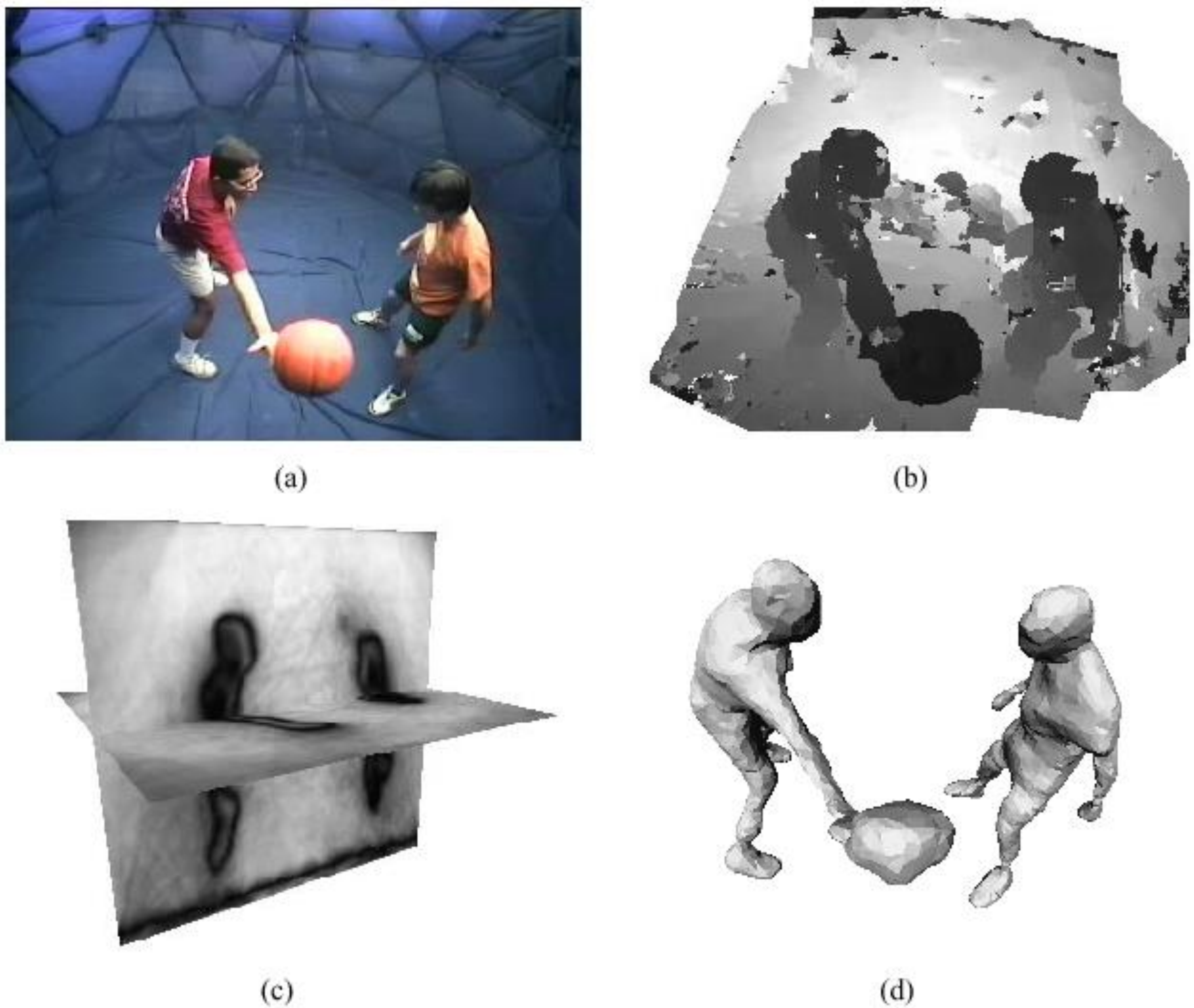
49 cameras
30 Hz
512x512 color
17 PCs

System Overview



[PhD thesis Peter Rander, CMU, 1999]

Example: Basketball



a) Original scene

b) Range Image

c) Integrated range images

d) 3D model extraction

Example: Basketball (cont.)

(e)

(f)

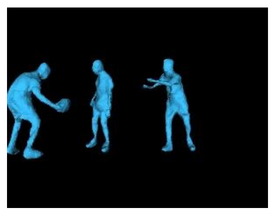
e) Rendered view of model with texture

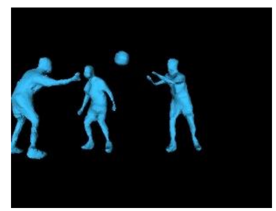
f) Rendered view of model from a virtual camera and combined with another digitized scene

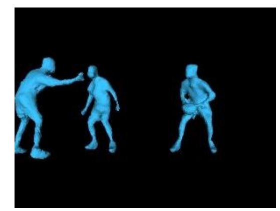
Inputs (two separate events)

Video 1

Reconstructed 3D shape







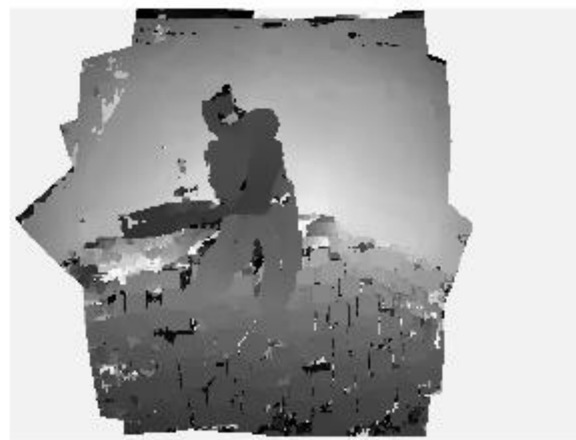
Video 2

Virtual View of combined event

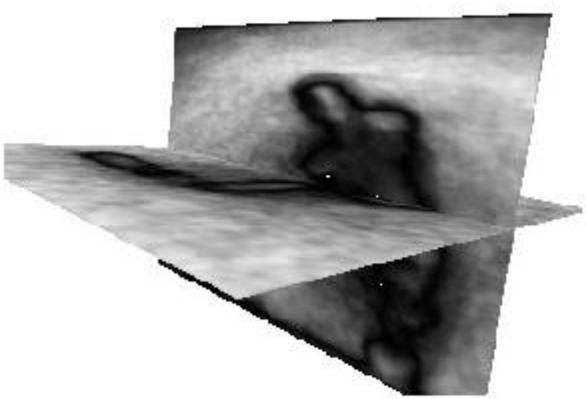
Video 3

Example: Baseball

(a)



(b)



(c)

(d)

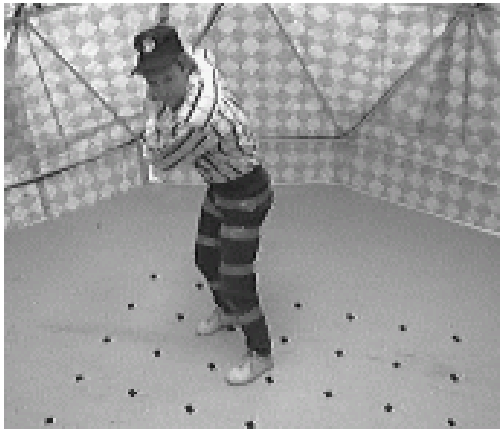
a) Original scene

b) Range Image

c) Integrated
range images

d) 3D model
extraction

Example: Baseball (cont.)



This example features a person swinging a baseball bat inside the recording studio. A director might select a single camera that provides a good view of the swing from the side (as in the above), but you might prefer to

- circle around as the batter swings...
- or stop the batter
- drop from above...
- *be the BALL!*

Example: Dance

Video 1

Video 2

Video 3

Example: Chair

Video 1

Video 2

Video 3