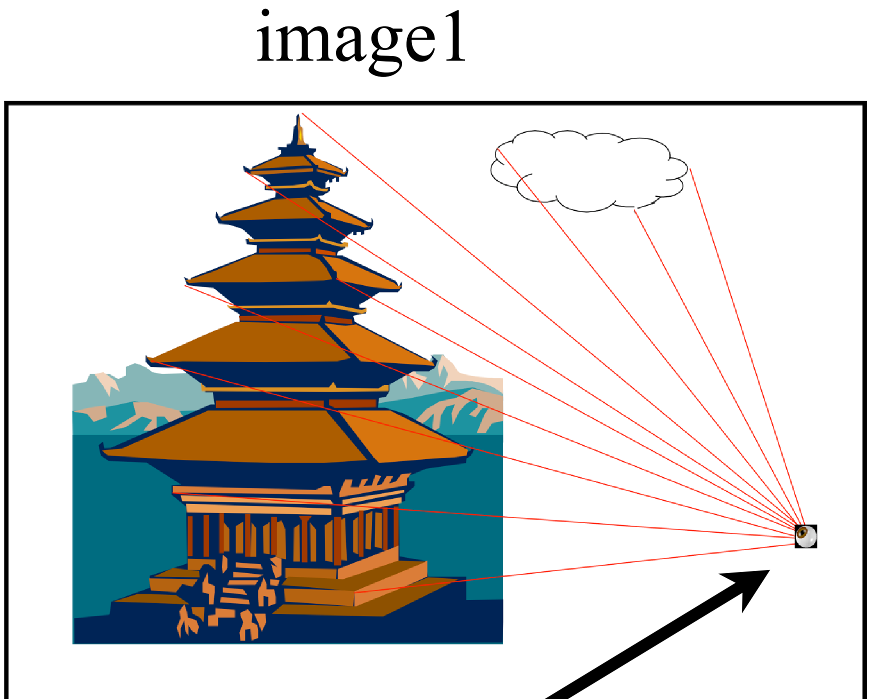
Generalized Stereo Reconstruction

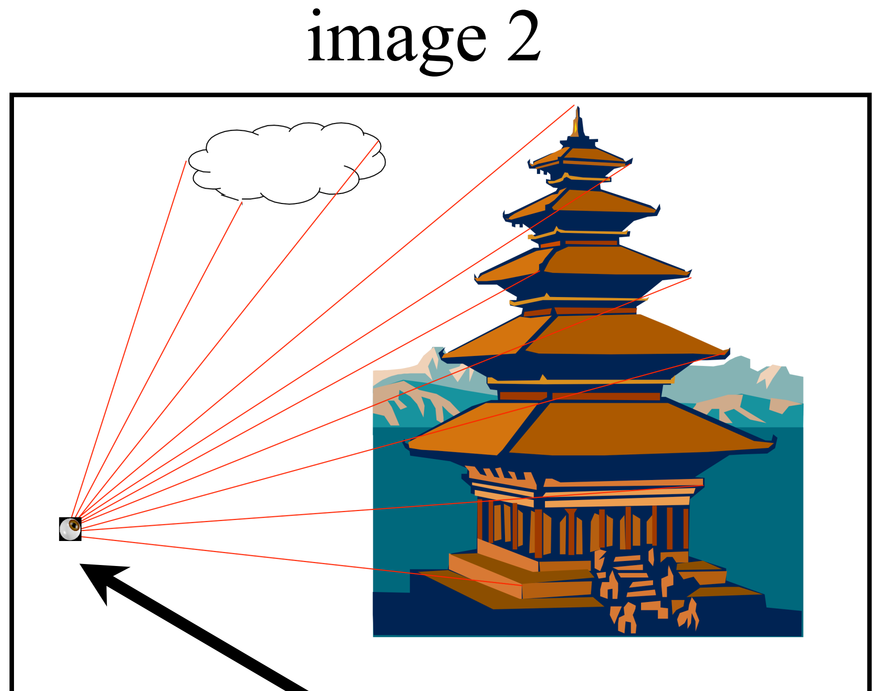
Vinay P. Namboodiri

- Slide credit to Robert Collins

Robert Collins
CSE486, Penn State

Epipolar Geometry

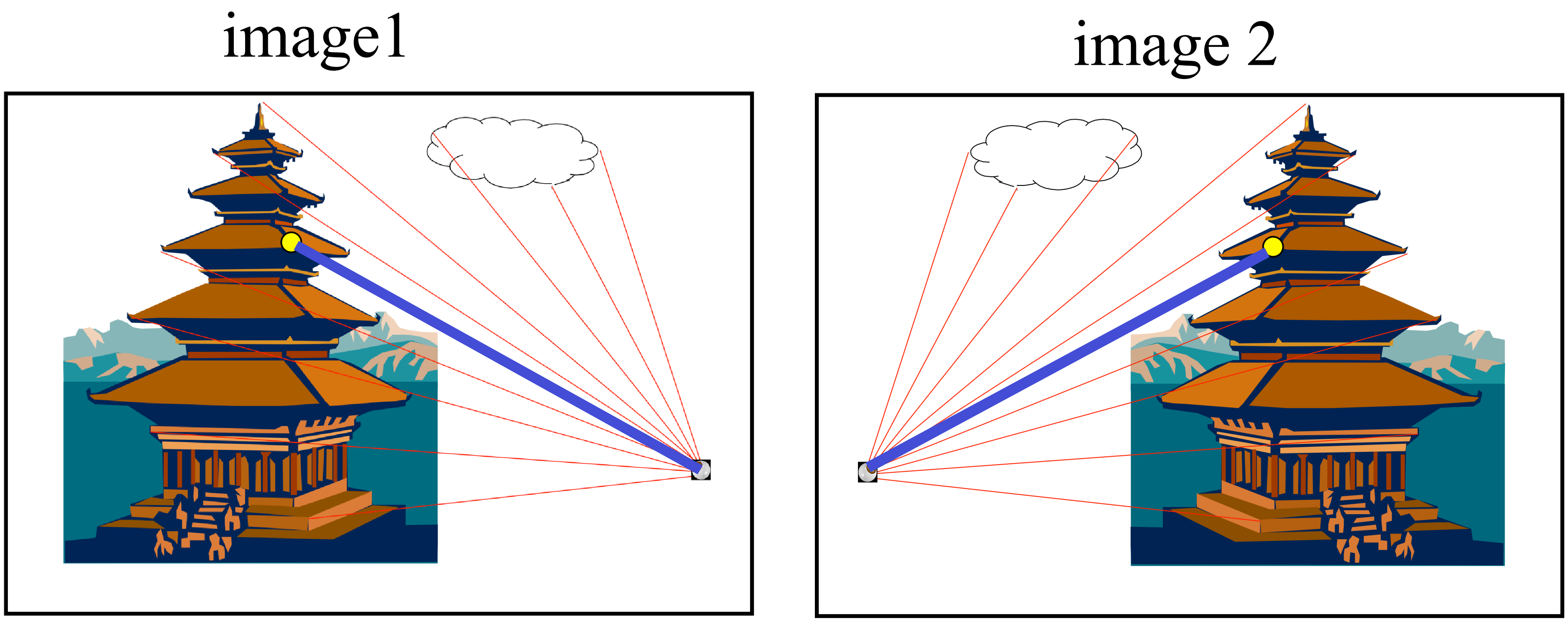




Epipole : location of cam2 as seen by cam1.

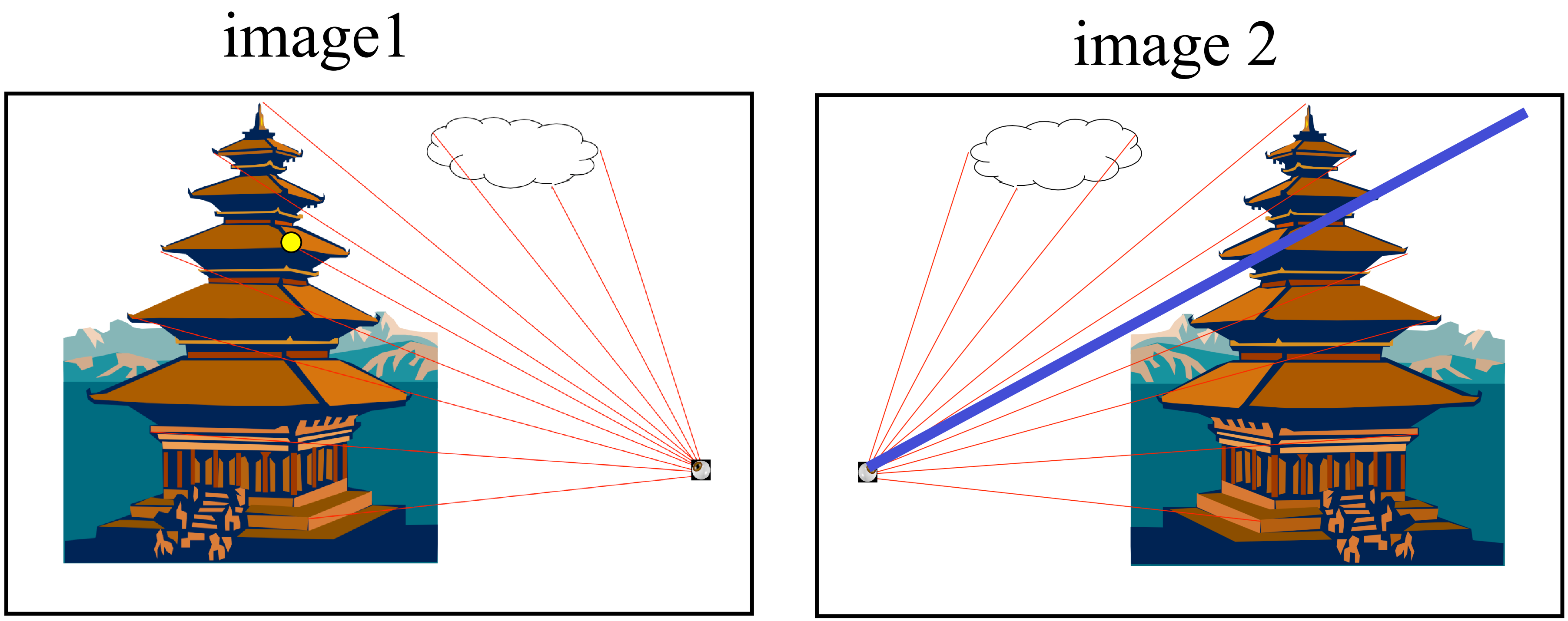
Epipole : location of cam1 as seen by cam2.

Epipolar Geometry



Corresponding points
lie on conjugate epipolar lines

This Lecture…



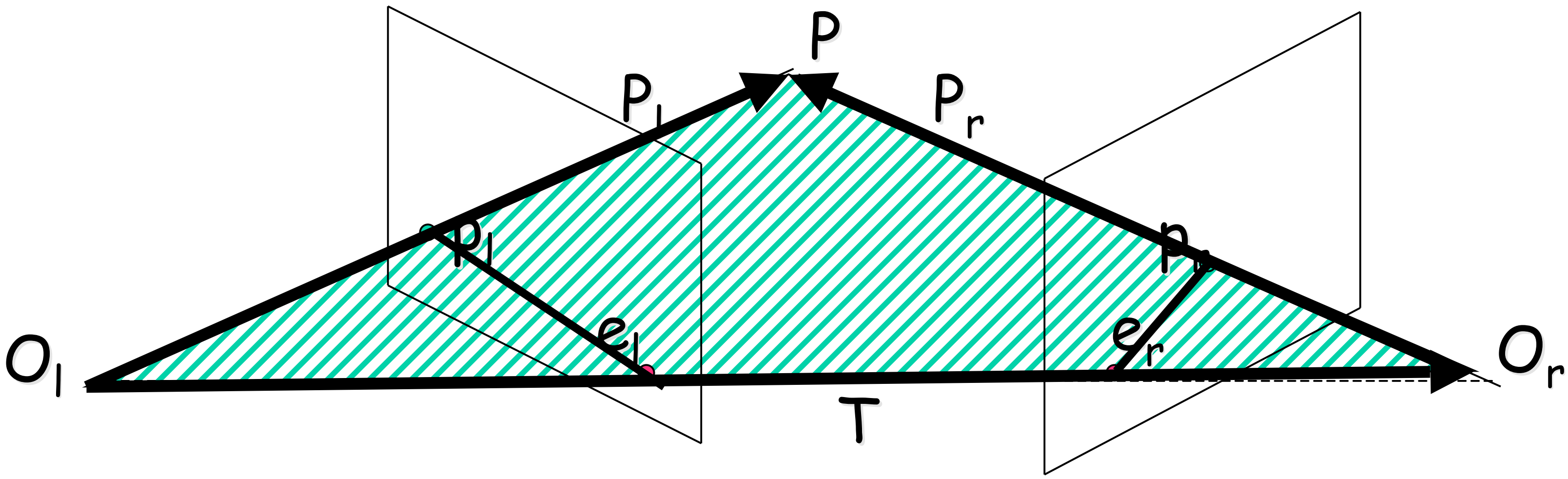
Given a point in one image, how do we determine the corresponding epipolar line to search along in the second image?

Essential Matrix

The essential and fundamental matrices are 3x3 matrices that “encode” the epipolar geometry of two views.

Motivation: Given a point in one image, multiplying by the essential/fundamental matrix will tell us which epipolar line to search along in the second view.

Essential Matrix



$$P_r = R(P_l - T)$$

R,T = rotation, and translation

$$\Rightarrow P_r^T R S P_l = 0$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

$$\Rightarrow P_r^T E P_l = 0$$

E=RS is “essential matrix”

Essential Matrix Properties

$$E = RS$$

- has rank 2
  ➔ has both a left and right nullspace (important!!!!)

- depends only on the EXTRINSIC Parameters (R & T)

Longuet-Higgins equation

$$P_r^T E P_l = 0$$

$$p_l = \frac{f_l}{Z_l} P_l \qquad\qquad p_r = \frac{f_r}{Z_r} P_r$$

$$\left(\cancel{\frac{Z_r}{f_r}} p_r\right)^T E \left(\cancel{\frac{Z_l}{f_l}} p_l\right) = 0$$

$$\boxed{p_r^T E p_l = 0}$$

Longuet-Higgins equation

$$P_r^T E P_l = 0$$

This relates viewing rays

Importance of Longuet-Higgens ...

This relates 2D film points

$$\boxed{p_r^T E p_l = 0}$$

Longuet-Higgins Makes Sense

- Note, there is nothing magic about Longuet-Higgins equation.
- A film point can also be thought of as a viewing ray. They are equivalent.
  - (u,v) 2D film point
  - (u,v,f) 3D point on film plane
  - k(u,v,f) viewing ray into the scene
  - k(X, Y, Z) ray through point P in the scene
    [hint: k=f/Z, and we have u=fX/Z, v=fY/Z].

Epipolar Lines

- Let l be a line in the image:

$$au + bv + c = 0$$

- Using homogeneous coordinates:

$$\tilde{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \tilde{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \boxed{\tilde{p}^T \tilde{l} = \tilde{l}^T \tilde{p} = 0}$$

Epipolar Lines

- Remember:

$$p_r^T E p_l = 0$$

$$\tilde{l}_r = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

p_r belongs to epipolar line in the right image defined by

$$\tilde{l}_r = E p_l$$

Epipolar Lines

- Remember:

$$p_r^T E p_l = 0$$

$$\tilde{l}_l^{\,T} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}^T$$

p_l belongs to epipolar line in the left image defined by

$$\tilde{l}_l = E^T p_r$$

Epipoles

- Remember: epipoles belong to the epipolar lines

$$e_r^T E p_l = 0 \qquad\qquad p_r^T E e_l = 0$$

- And they belong to all the epipolar lines

$$e_r^T E = 0 \qquad\qquad E e_l = 0$$

We can use this to compute the location of the epipoles. There will be an example, shortly...

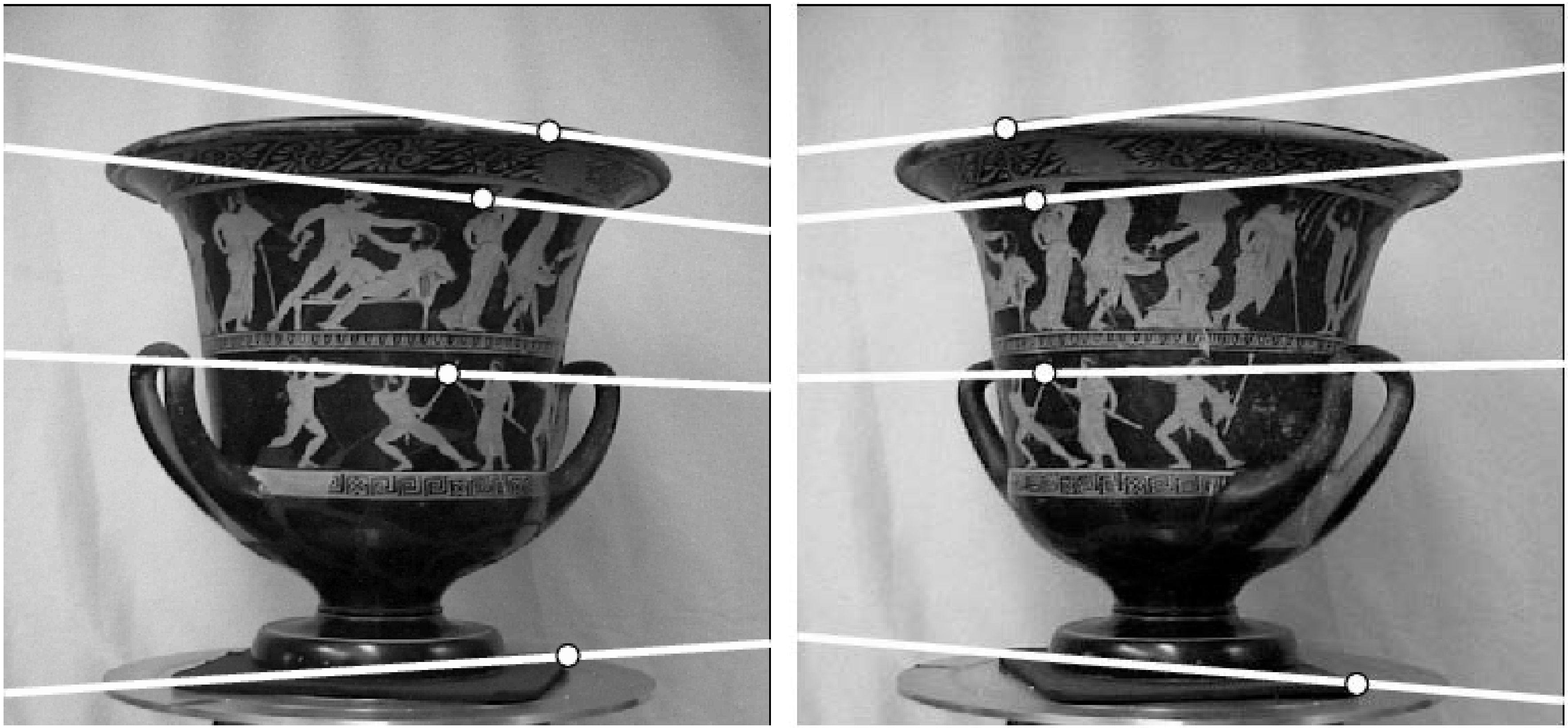
Essential Matrix Summary

Longuet-Higgins equation $$p_r^T E p_l = 0$$

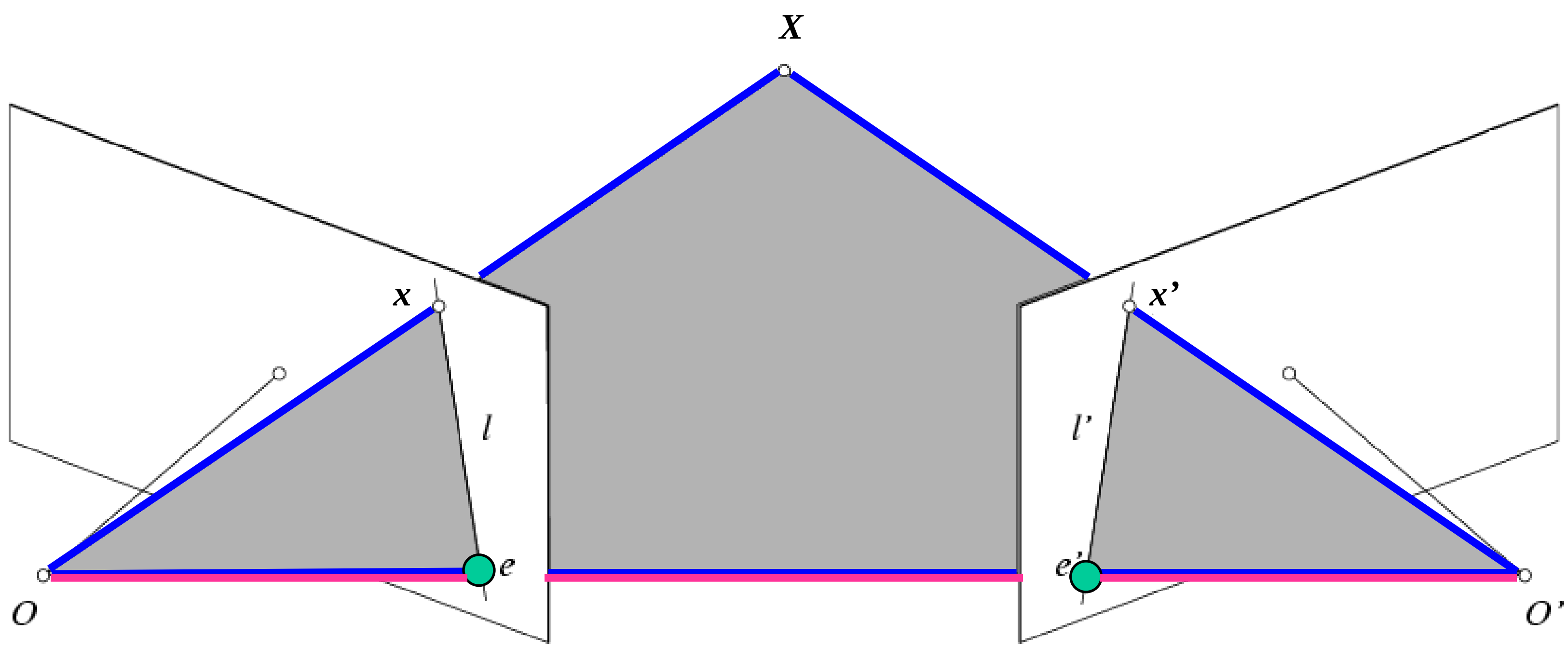
Epipolar lines:
$$\tilde{p_r}^T \tilde{l_r} = 0 \qquad \tilde{p_l}^T \tilde{l_l} = 0$$
$$\tilde{l_r} = E p_l \qquad \tilde{l_l} = E^T p_r$$

Epipoles:
$$e_r^T E = 0 \qquad E e_l = 0$$

Two-view geometry



Epipolar geometry



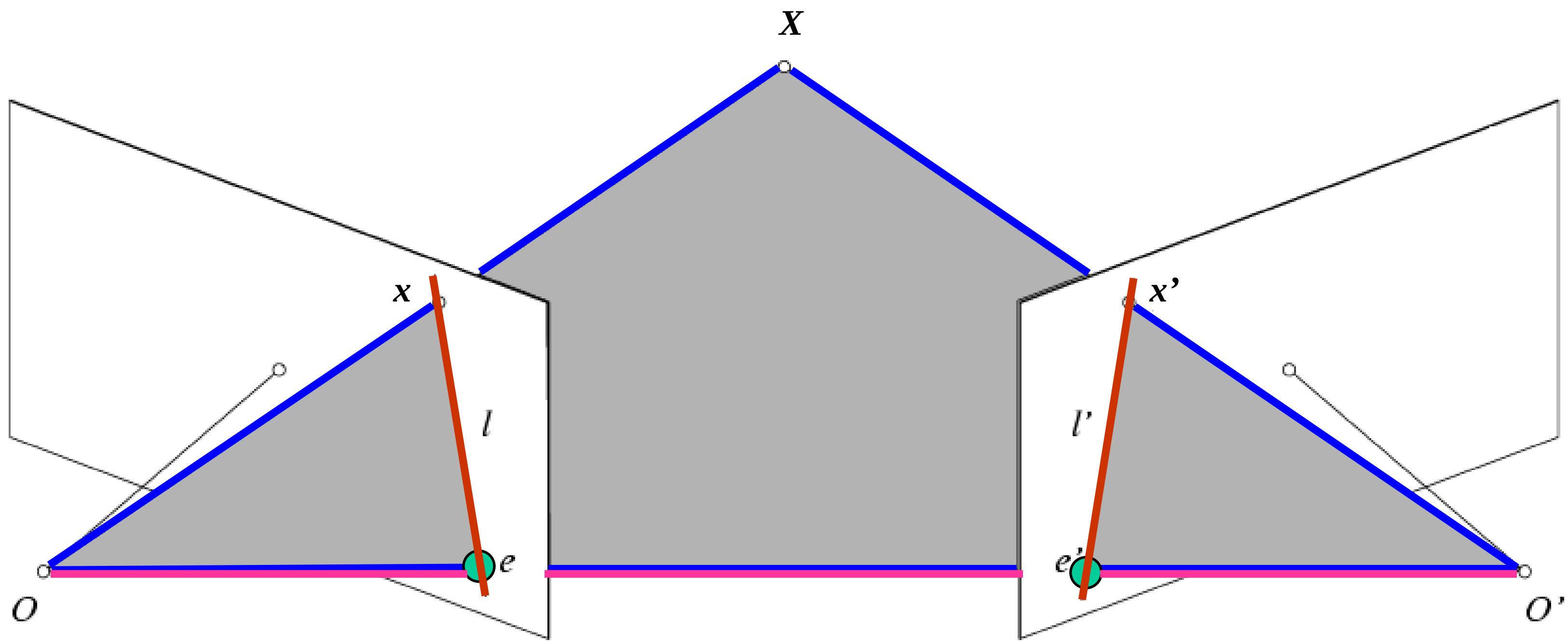
- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**

= intersections of baseline with image planes
= projections of the other camera center

The Epipole

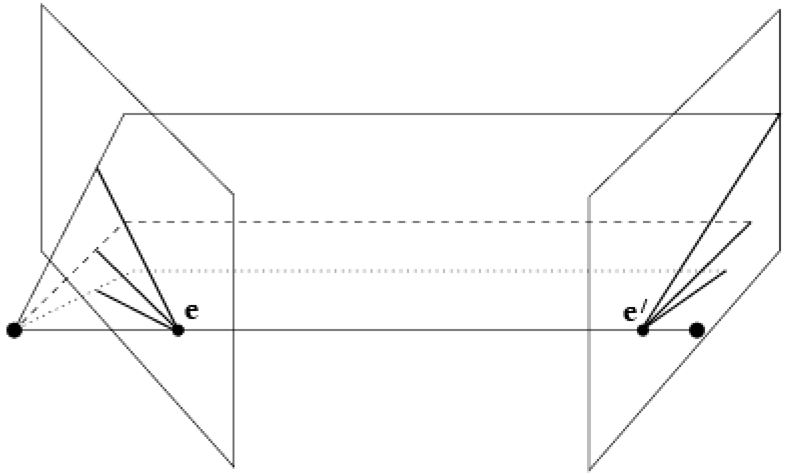
Photo by Frank Dellaert

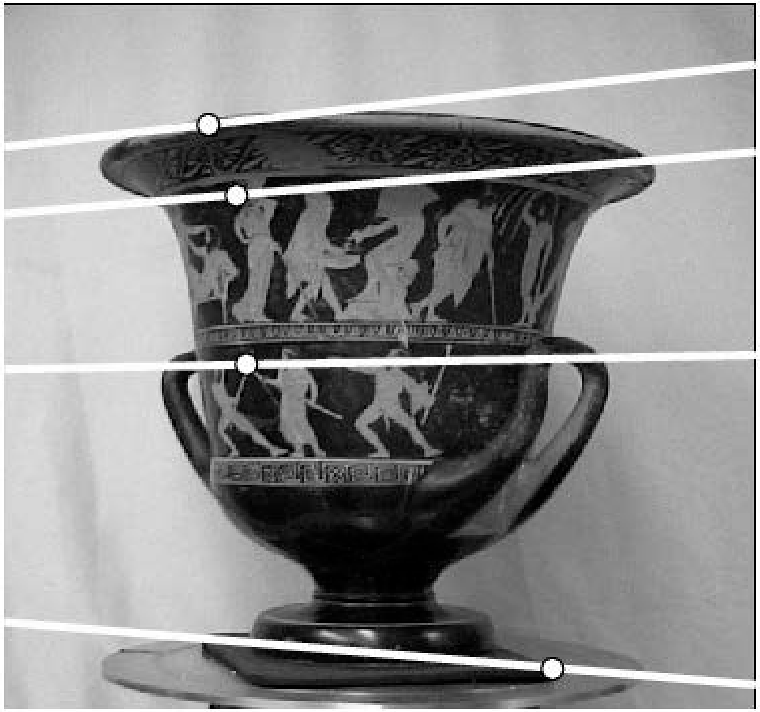
Epipolar geometry



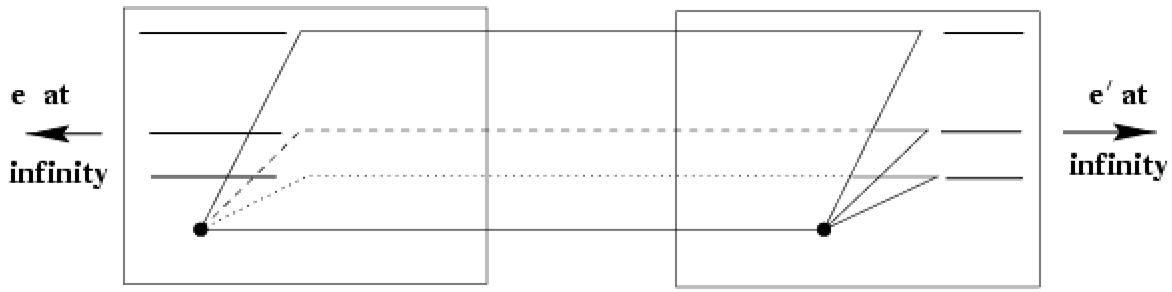
- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**
  = intersections of baseline with image planes
  = projections of the other camera center
- **Epipolar Lines -** intersections of epipolar plane with image planes (always come in corresponding pairs)

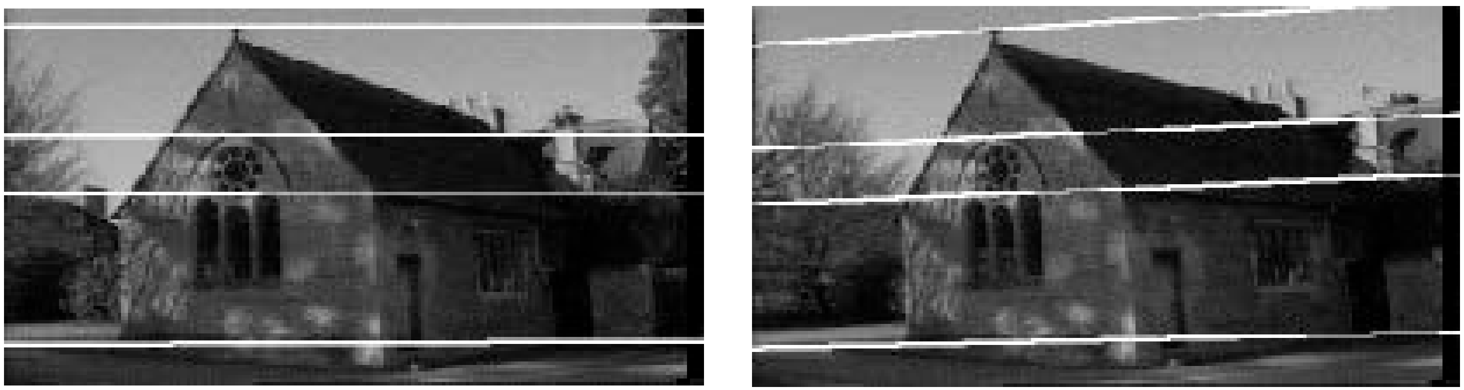
Example: Converging cameras





Example: Motion parallel to image plane

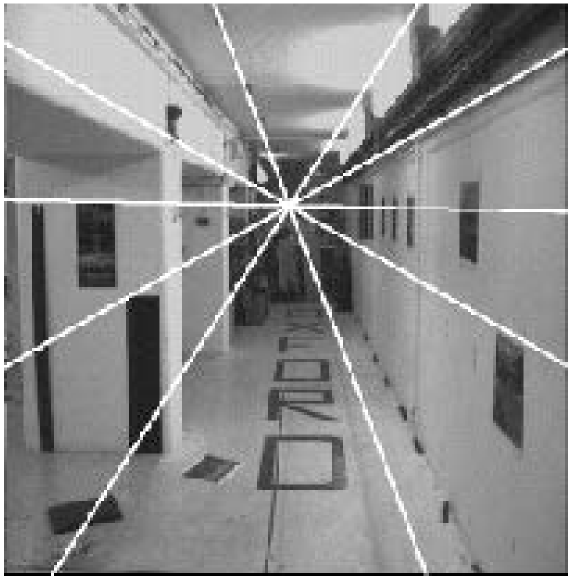


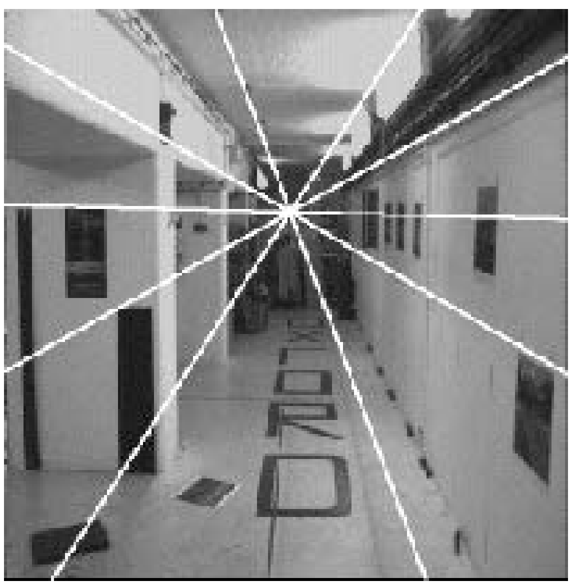


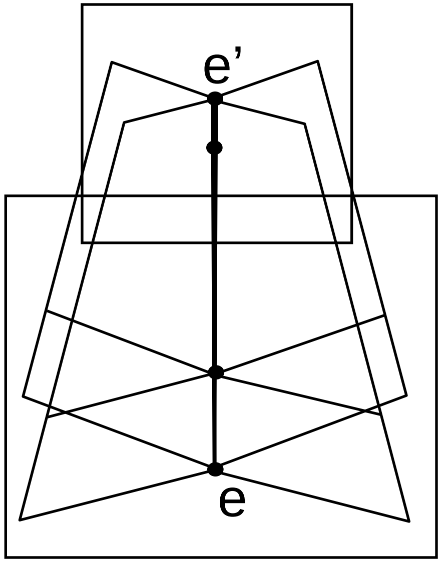
Example: Motion perpendicular to image plane

Example: Motion perpendicular to image plane

Example: Motion perpendicular to image plane

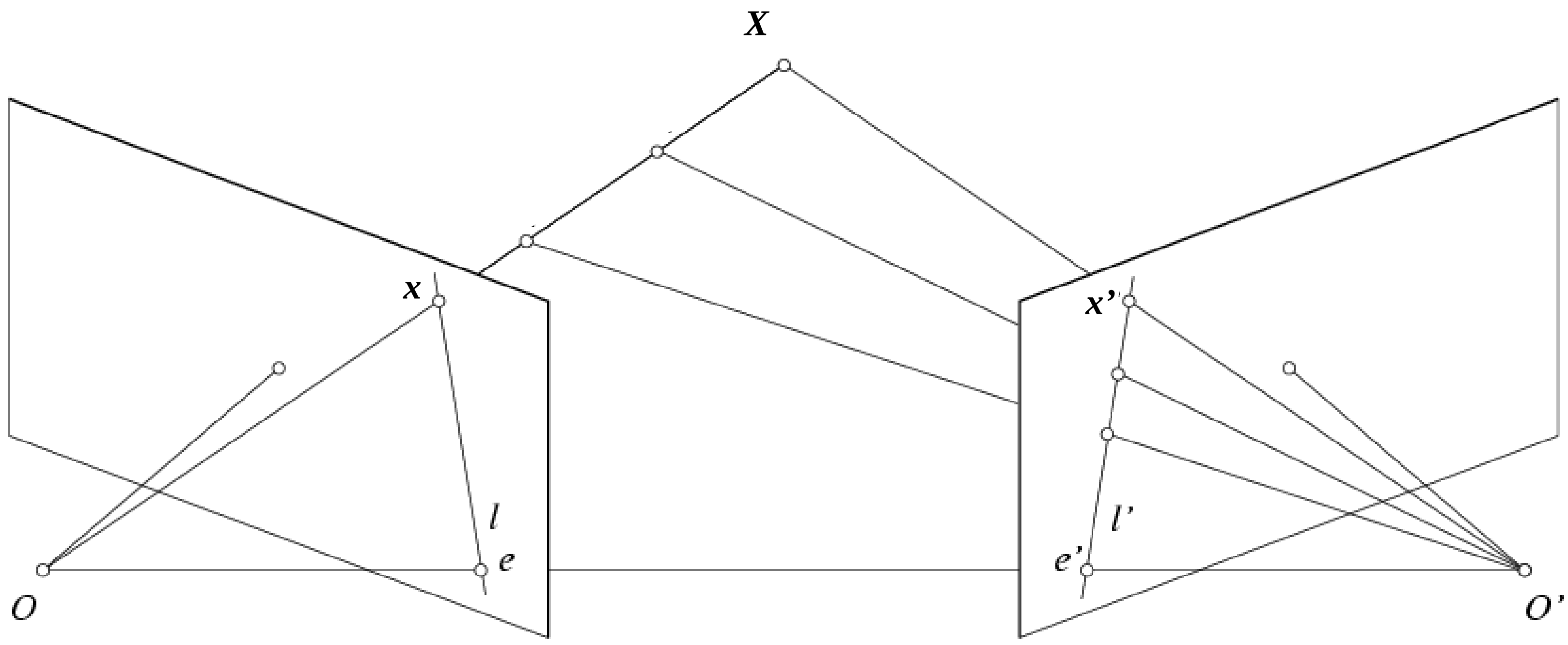






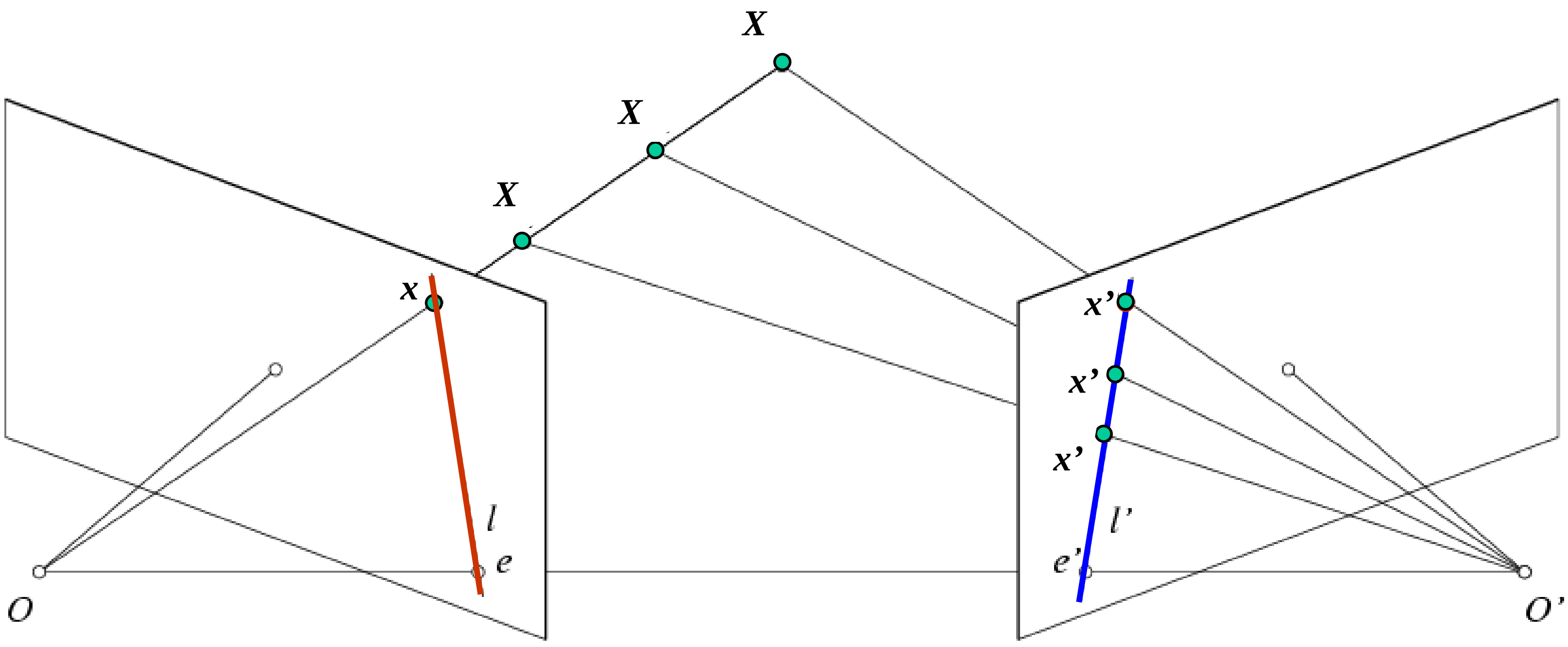
Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Epipolar constraint



- If we observe a point x in one image, where can the corresponding point x' be in the other image?

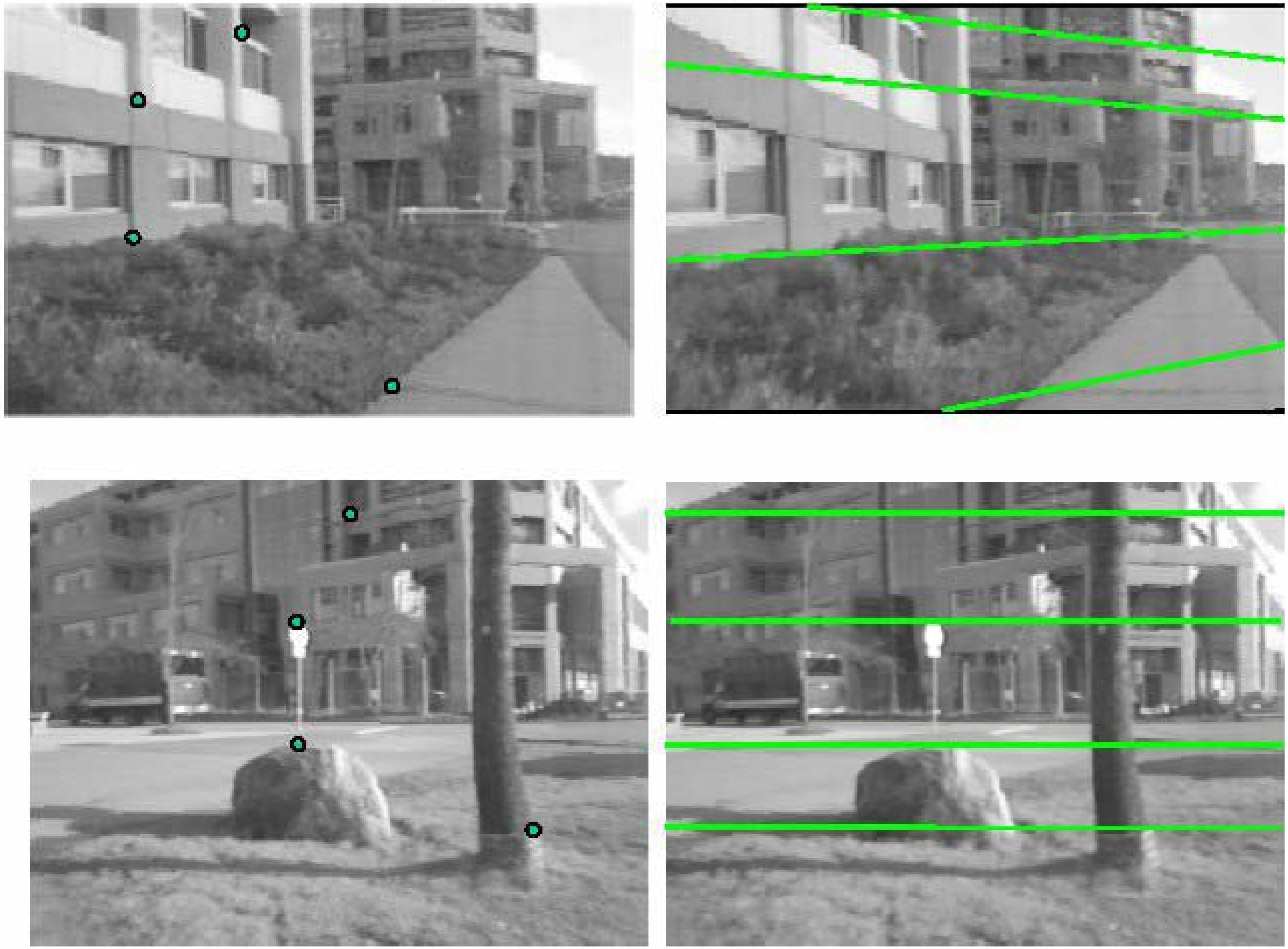
Epipolar constraint



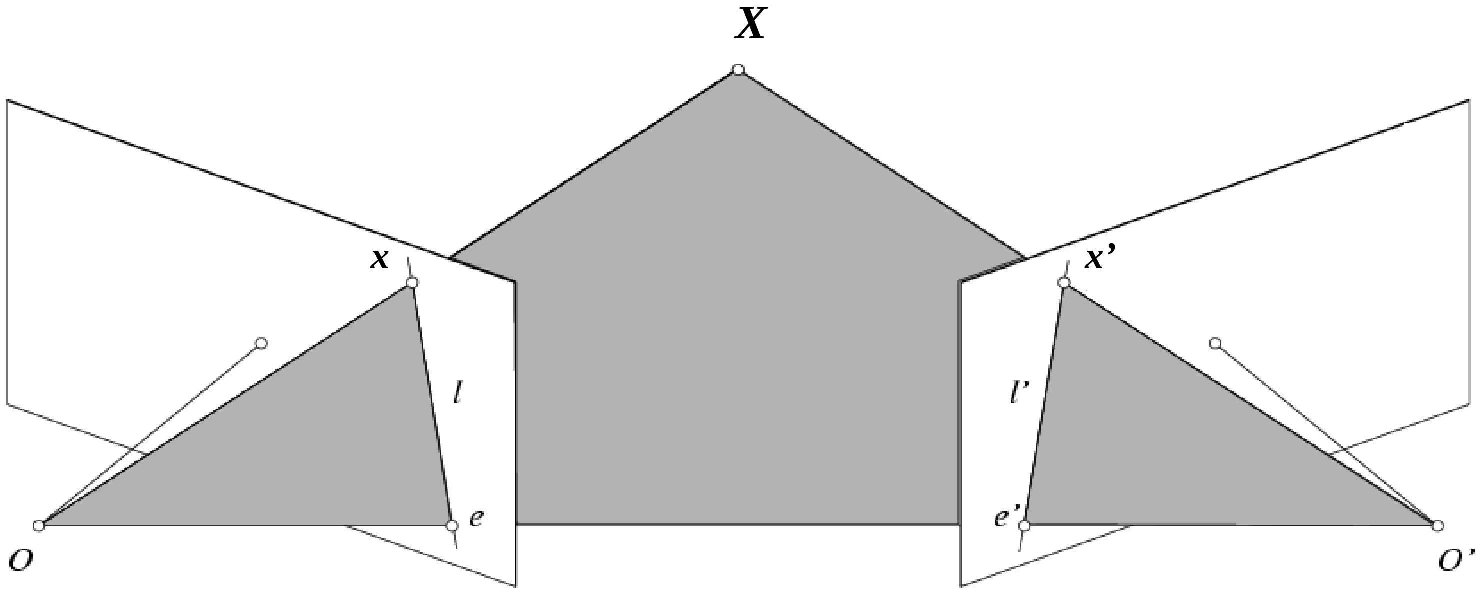
- Potential matches for *x* have to lie on the corresponding epipolar line *l'*.

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

Epipolar constraint example

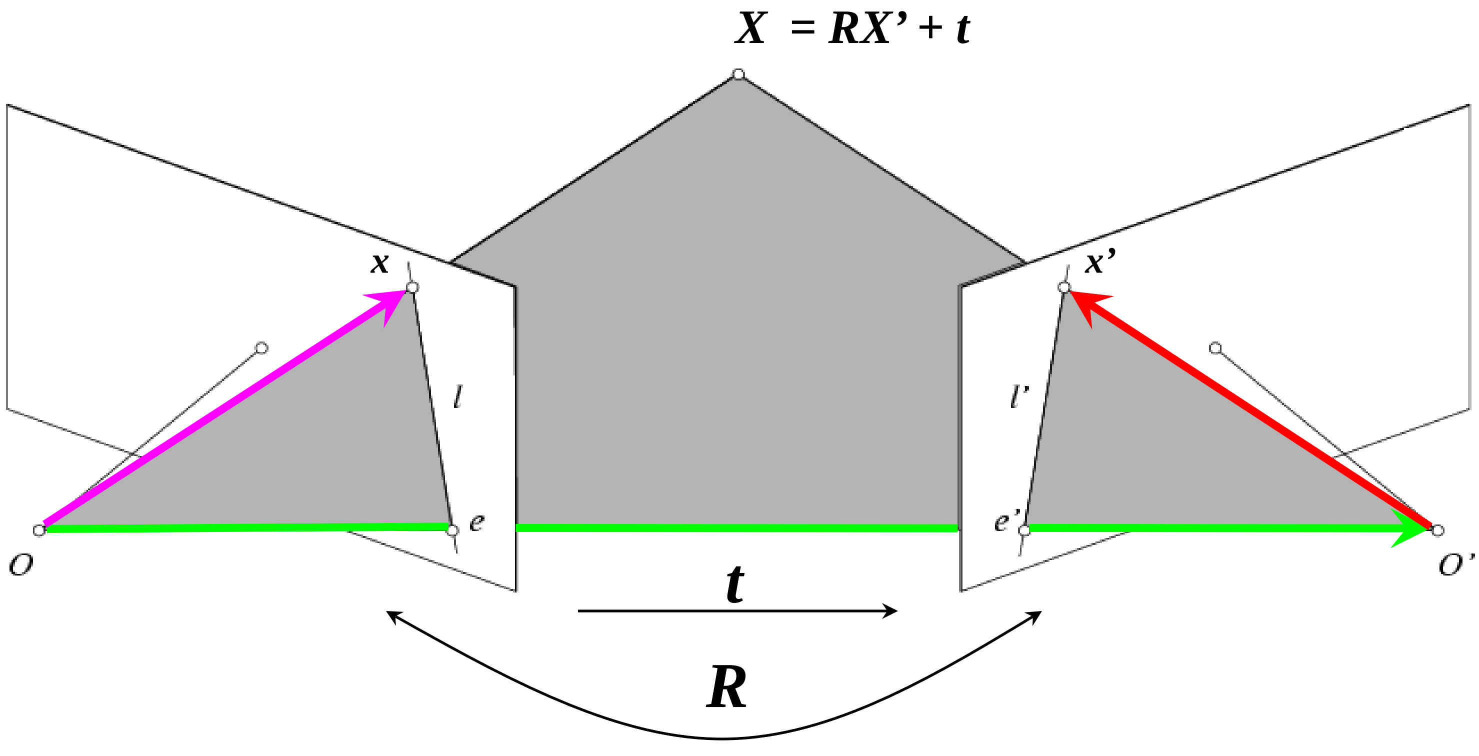


Epipolar constraint: Calibrated case



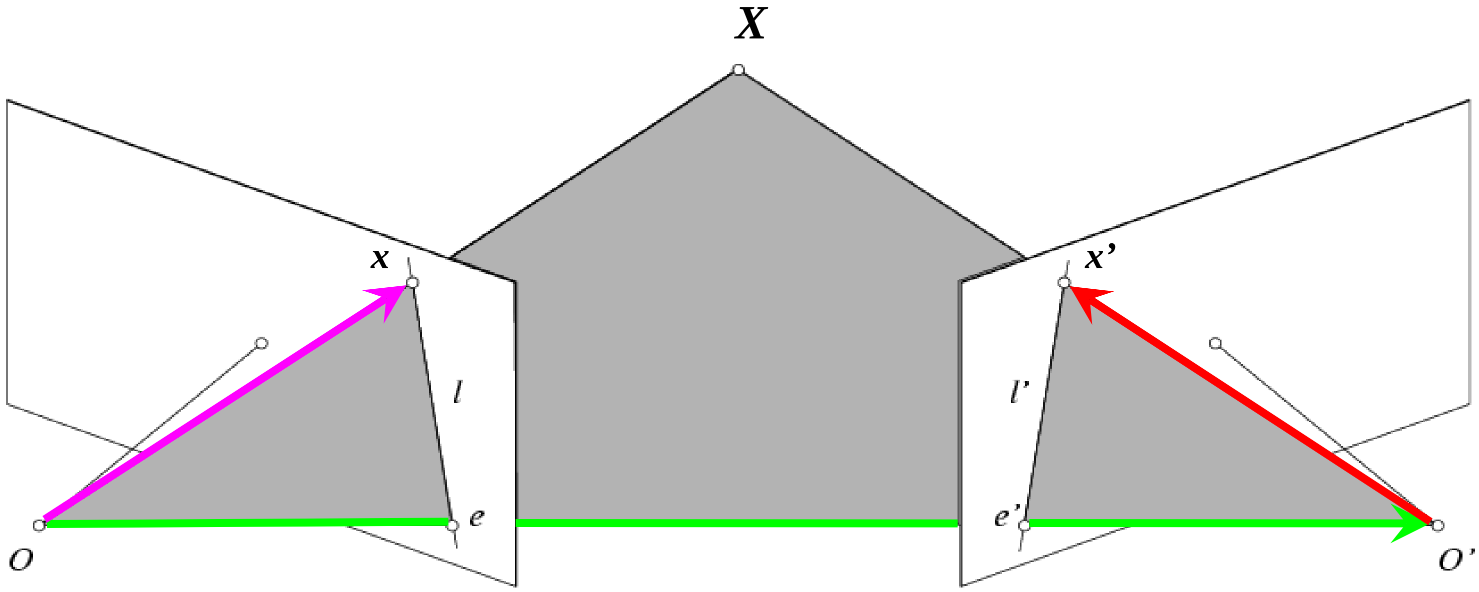
- Assume that the intrinsic and extrinsic parameters of the cameras are known
- We can multiply the projection matrix of each camera (and the image points) by the inverse of the calibration matrix to get *normalized* image coordinates
- We can also set the global coordinate system to the coordinate system of the first camera. Then the projection matrix of the first camera is $[\mathbf{I} \mid \mathbf{0}]$.

Epipolar constraint: Calibrated case



The vectors ***x***, ***t***, and ***Rx'*** are coplanar

Epipolar constraint: Calibrated case

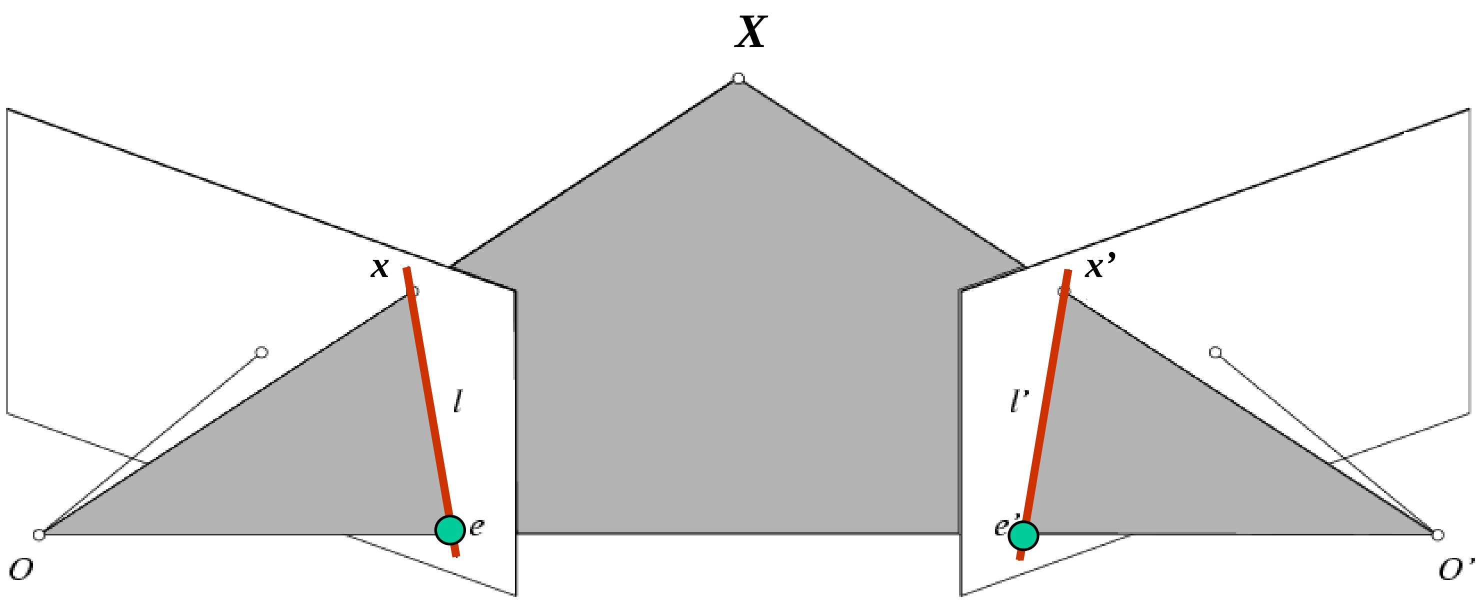


$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

Essential Matrix
(Longuet-Higgins, 1981)

The vectors ***x***, ***t***, and ***Rx'*** are coplanar

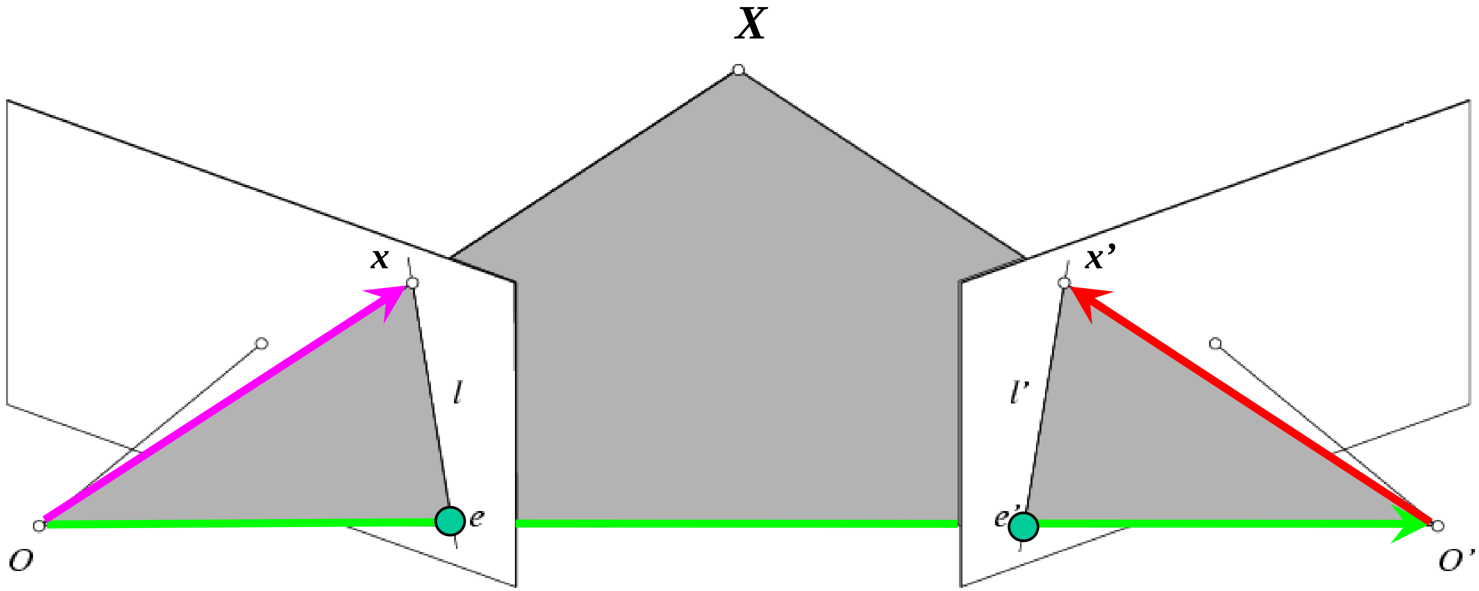
Epipolar constraint: Calibrated case



$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

- *E x'* is the epipolar line associated with *x'* (*l = E x'*)
- $E^T x$ is the epipolar line associated with *x* ($l' = E^T x$)
- *E e'* = 0 and $E^T e = 0$
- *E* is singular (rank two)
- *E* has five degrees of freedom

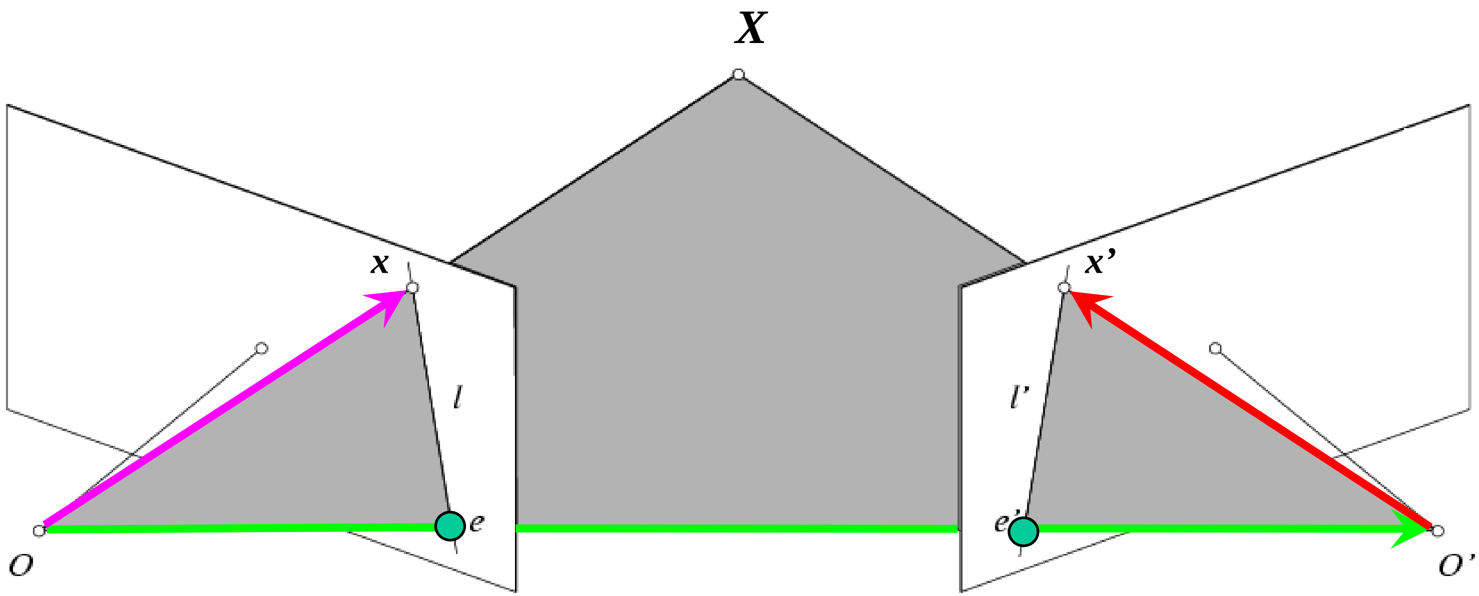
Epipolar constraint: Uncalibrated case



- The calibration matrices *K* and *K'* of the two cameras are unknown
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad\qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Epipolar constraint: Uncalibrated case



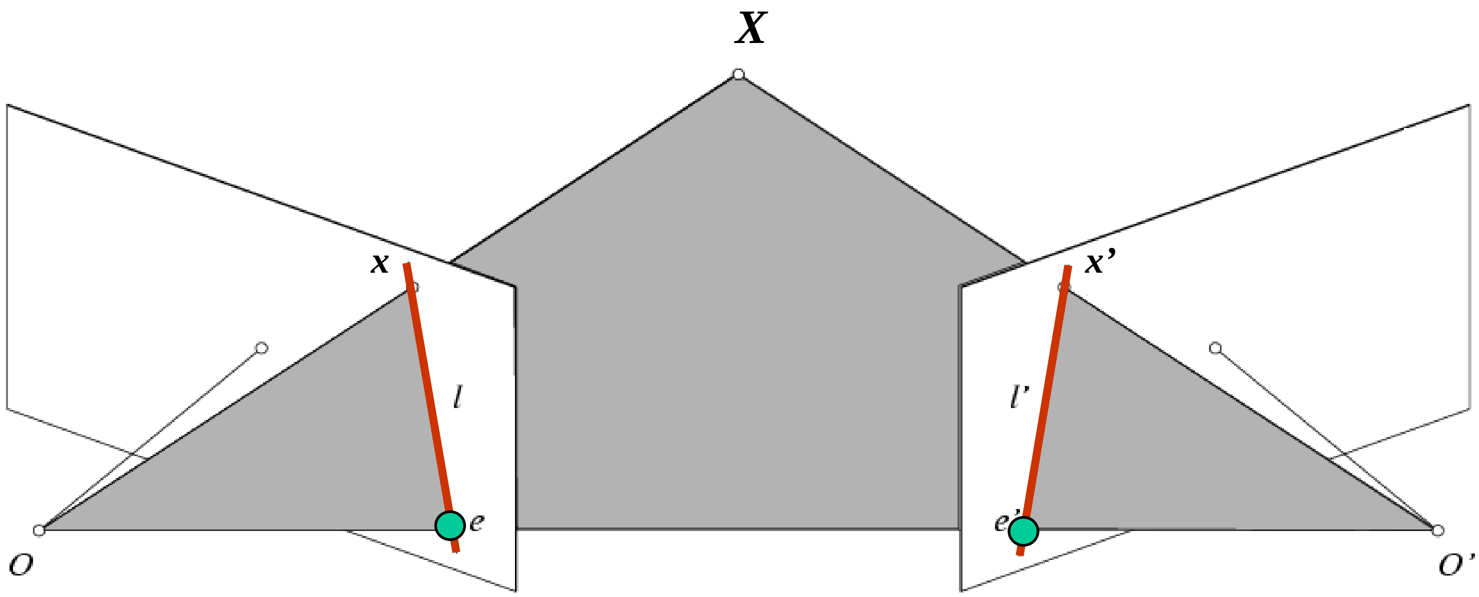
$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

$$\hat{x} = K^{-1} x$$

$$\hat{x}' = K'^{-1} x'$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar constraint: Uncalibrated case



$$\hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F x'$ is the epipolar line associated with x' ($l = F x'$)
- $F^T x$ is the epipolar line associated with x ($l' = F^T x$)
- $F e' = 0$ and $F^T e = 0$
- F is singular (rank two)
- F has seven degrees of freedom

The eight-point algorithm

$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x'} = (u', v', 1)^T$

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

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

Minimize:

$$\sum_{i=1}^{N} (x_i^T F x_i')^2$$

under the constraint

$$F_{33} = 1$$

The eight-point algorithm

- Meaning of error $\sum_{i=1}^{N}(x_i^T F x_i')^2$:

  sum of Euclidean distances between points x_i and epipolar lines Fx'_i (or points x'_i and epipolar lines $F^T x_i$) multiplied by a scale factor

- Nonlinear approach: minimize

$$\sum_{i=1}^{N}\left[\mathrm{d}^2(x_i, F x_i') + \mathrm{d}^2(x_i', F^T x_i)\right]$$

Problem with eight-point algorithm

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

Problem with eight-point algorithm

$$
\begin{bmatrix}
250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 \\
2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 \\
416374.23 & 871684.30 & 935.47 & 408110.89 & 854384.92 & 916.90 & 445.10 & 931.81 \\
191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 \\
48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 \\
164786.04 & 546559.67 & 813.17 & 1998.37 & 6628.15 & 9.86 & 202.65 & 672.14 \\
116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 \\
135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48
\end{bmatrix}
\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix}
= -\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
$$

Poor numerical conditioning

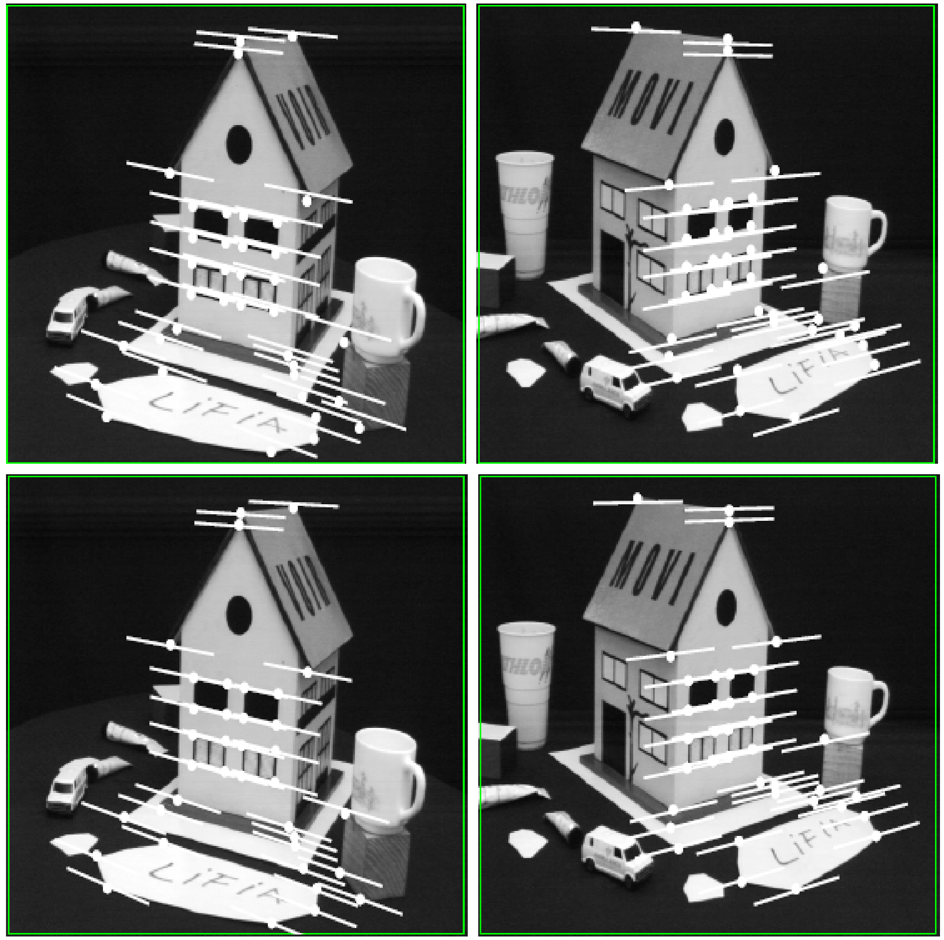
Can be fixed by rescaling the data

The normalized eight-point algorithm

(Hartley, 1995)

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
- Use the eight-point algorithm to compute *F* from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of *F* and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if T and T' are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T F T'$

Comparison of estimation algorithms



	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

From epipolar geometry to camera calibration

- Estimating the fundamental matrix is known as “weak calibration”
- If we know the calibration matrices of the two cameras, we can estimate the essential matrix: $E = K^T F K'$
- The essential matrix gives us the relative rotation and translation between the cameras, or their extrinsic parameters

Robert Collins
CSE486, Penn State

Fundamental Matrix

The essential matrix uses CAMERA coordinates

To use image coordinates we must consider the INTRINSIC camera parameters:

Pixel coord (row,col)

Affine transform matrix

Camera (film) coord

$$\bar{p}_l = M_l p_l \qquad p_l = M_l^{-1} \bar{p}_l$$

$$\bar{p}_r = M_r p_r \qquad p_r = M_r^{-1} \bar{p}_r$$

Fundamental Matrix

$$p_l = M_l^{-1} \bar{p}_l$$

$$p_r = M_r^{-1} \bar{p}_r$$

$$p_r^T E p_l = 0$$

$$(M_r^{-1} \bar{p}_r)^T E (M_l^{-1} \bar{p}_l) = 0$$

$$\bar{p}_r{}^T (M_r^{-T} E M_l^{-1}) \bar{p}_l = 0$$

$$\boxed{\bar{p}_r{}^T F \bar{p}_l = 0}$$

short version: The same equation works in pixel coordinates too!

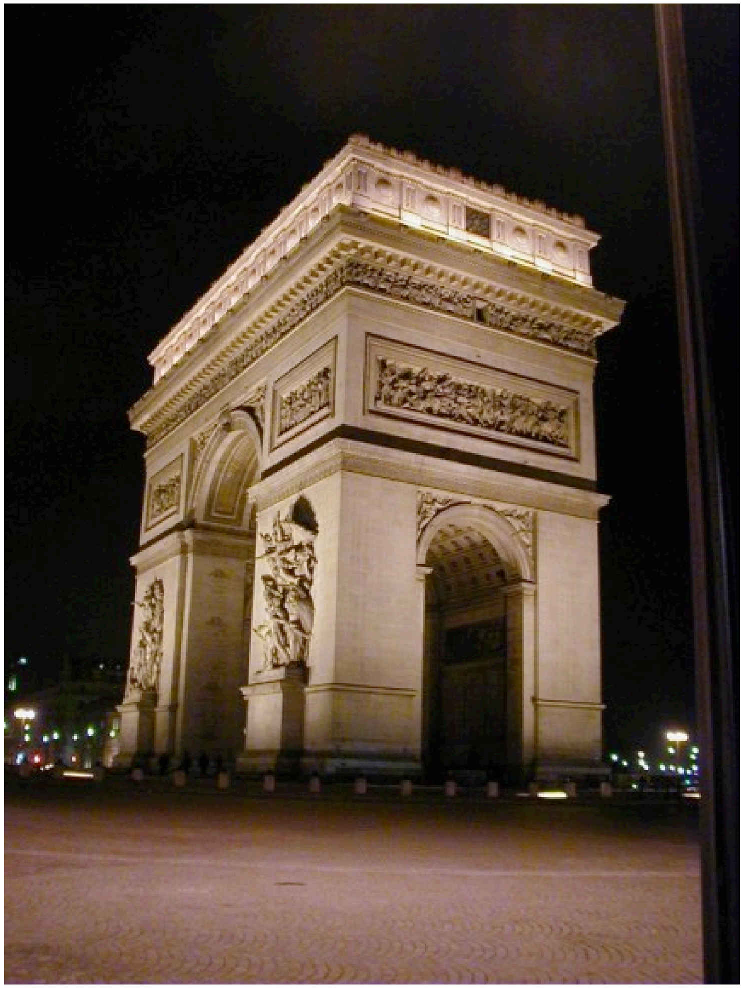
Fundamental Matrix Properties

$$F = M_r^{-T} R S M_l^{-1}$$

- has rank 2
- depends on the INTRINSIC and EXTRINSIC Parameters (f, etc ; R & T)

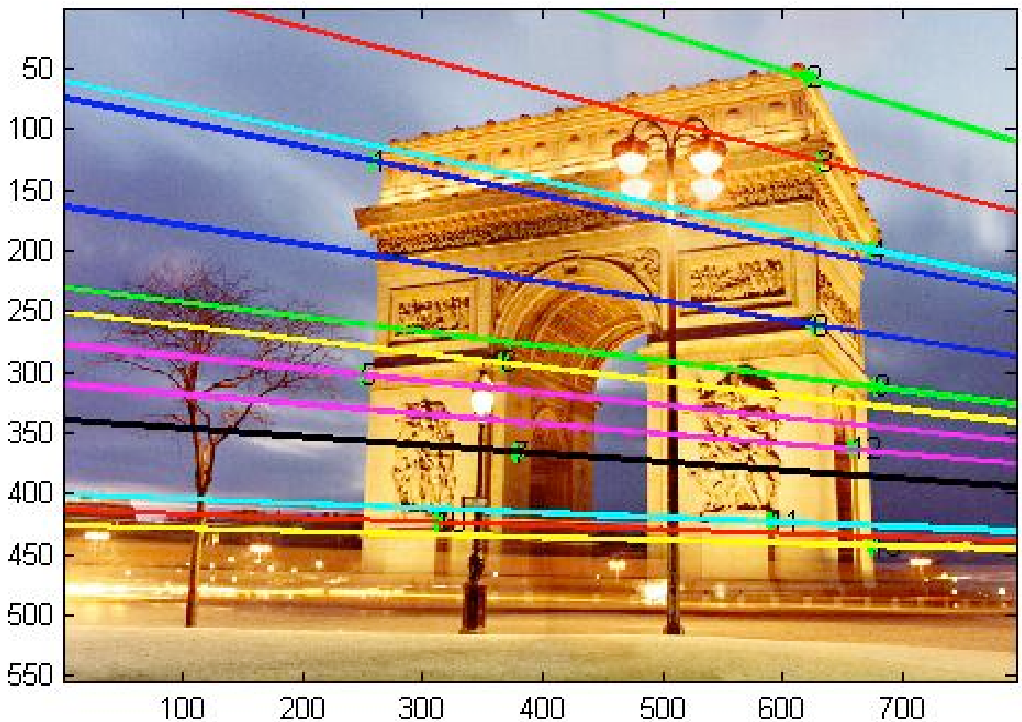
Analogous to essential matrix. The fundamental matrix also tells how pixels (points) in each image are related to epipolar lines in the other image.

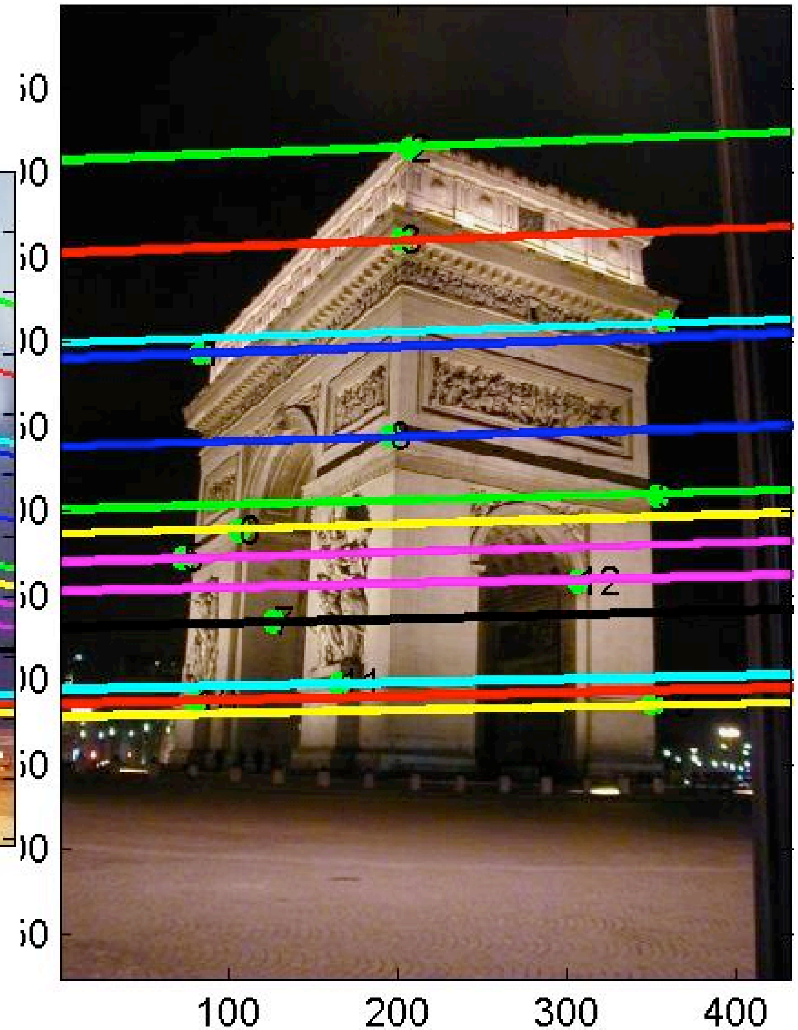
Example



Robert Collins
CSE486, Penn State

Example





Example

$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

Example

$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix} \begin{pmatrix} 343.53 \\ 221.70 \\ 1.0 \end{pmatrix}$$

x = 343.5300 y = 221.7005

$$\begin{matrix} 0.0001 \\ 0.0045 \\ -1.1942 \end{matrix} \rightarrow \begin{matrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{matrix}$$

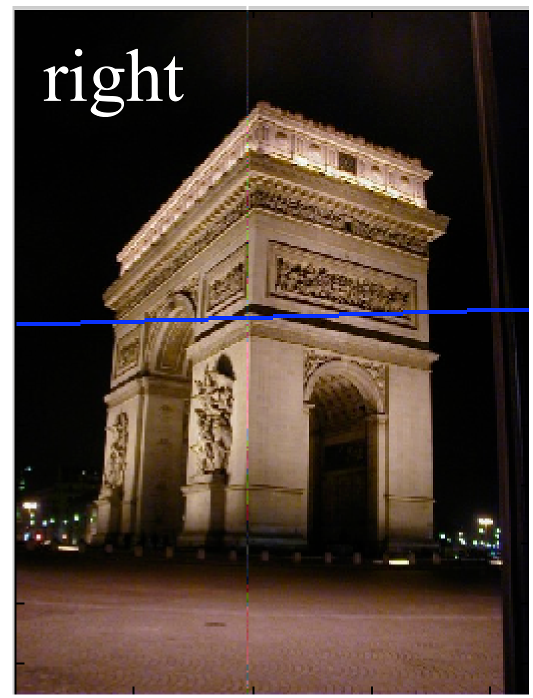
normalize so sum of squares
of first two terms is 1 (optional)

Example

$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix} \begin{pmatrix} 343.53 \\ 221.70 \\ 1.0 \end{pmatrix}$$

x = 343.5300 y = 221.7005

$$\begin{matrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{matrix}$$

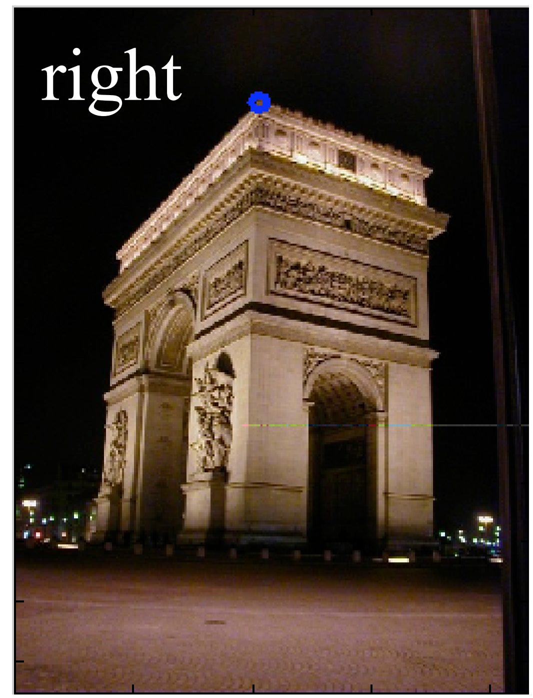


Example

$$
\begin{pmatrix} 205.5526 & 80.5 & 1.0 \end{pmatrix}
\begin{pmatrix}
-0.00310695 & -0.0025646 & 2.96584 \\
-0.028094 & -0.00771621 & 56.3813 \\
13.1905 & -29.2007 & -9999.79
\end{pmatrix}
$$

$$L = (0.0010 \quad -0.0030 \quad -0.4851)$$

$$\rightarrow (0.3211 \quad -0.9470 \;\; -151.39)$$



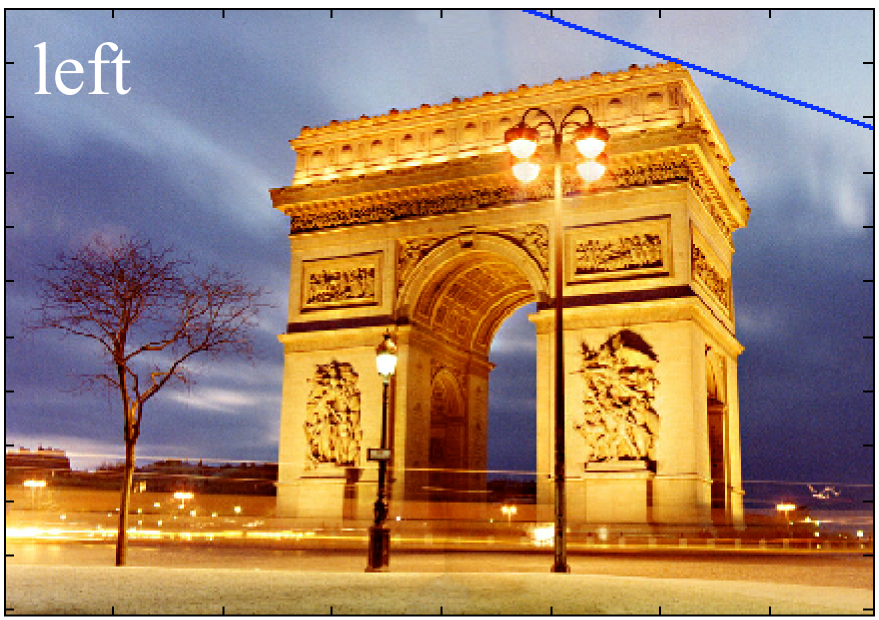
x = 205.5526 y = 80.5000

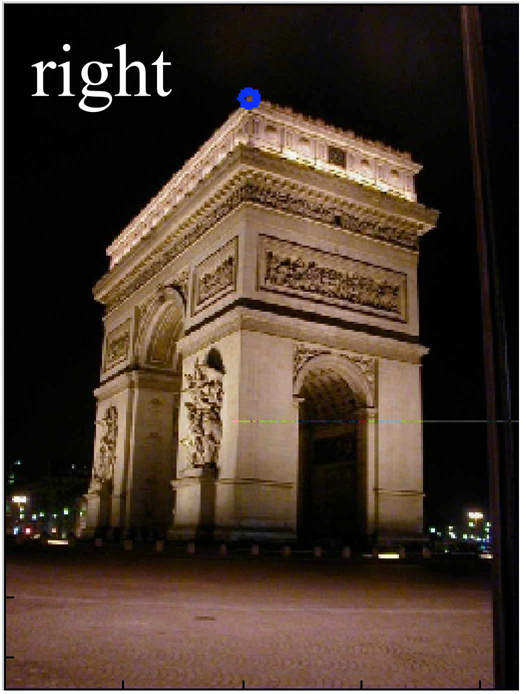
Robert Collins
CSE486, Penn State

Example

$$\begin{pmatrix} 205.5526 & 80.5 & 1.0 \end{pmatrix} \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

$$L = \begin{pmatrix} 0.3211 & -0.9470 & -151.39 \end{pmatrix}$$



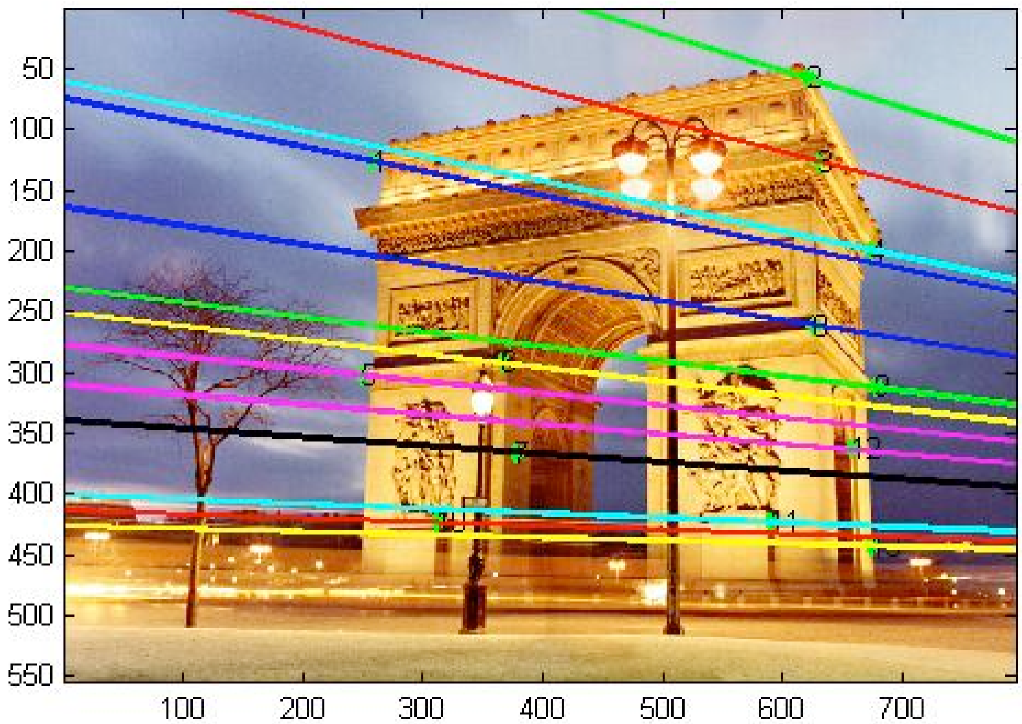


x = 205.5526 y = 80.5000

Robert Collins
CSE486, Penn State

Example

where is the epipole?

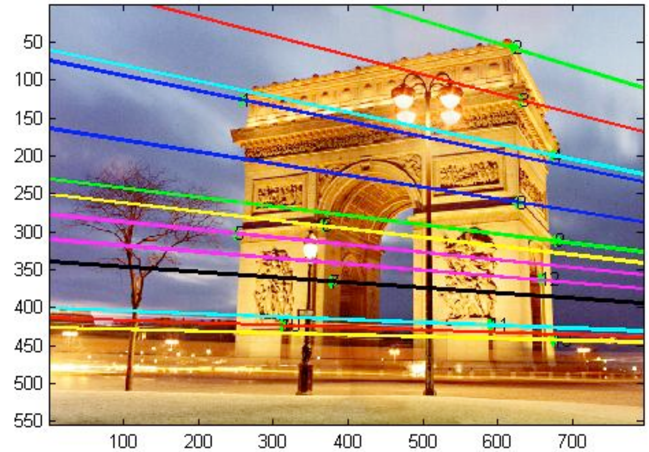


$$F * e_L = 0$$

vector in the right nullspace of matrix F

However, due to noise, F may not be singular. So instead, next best thing is eigenvector associated with smallest eigenvalue of F

Example



>> [u,d] = eigs(F' * F)

u =

$$\begin{array}{ccc} -0.0013 & 0.2586 & -0.9660 \\ 0.0029 & -0.9660 & -0.2586 \\ 1.0000 & 0.0032 & -0.0005 \end{array}$$

d = 1.0e8*

$$\begin{array}{ccc} -1.0000 & 0 & 0 \\ 0 & -0.0000 & 0 \\ 0 & 0 & -0.0000 \end{array}$$

eigenvector associated with smallest eigenvalue

>> uu = u(:,3)

uu = (-0.9660 -0.2586 -0.0005)

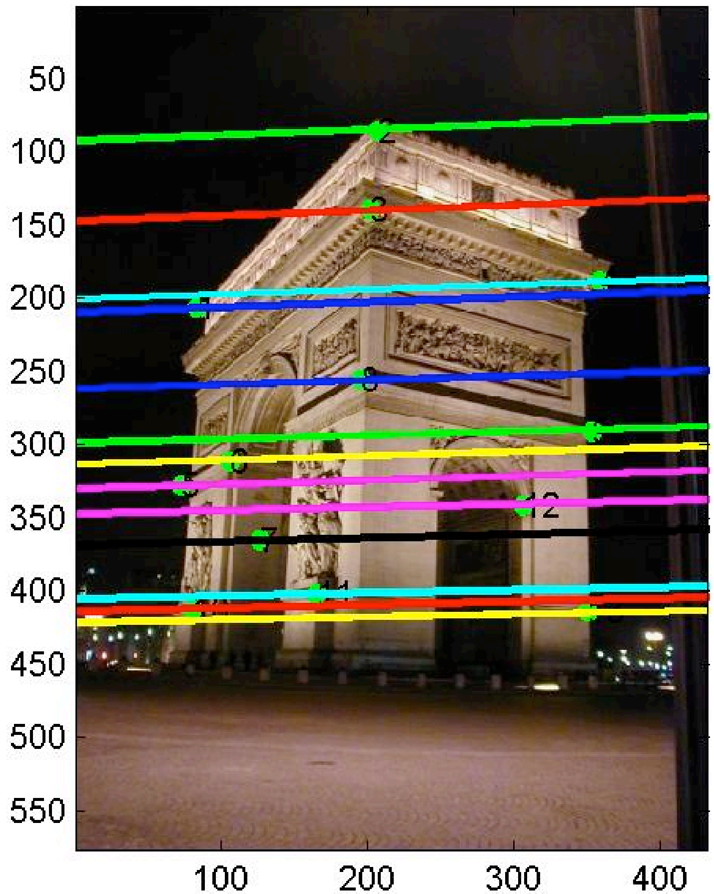
>> uu / uu(3) : to get pixel coords

(1861.02 498.21 1.0)

Robert Collins
CSE486, Penn State

Example

where is the epipole?



$e'_r * F = 0$

$\rightarrow F' * e_r = 0$

vector in the right nullspace of matrix F'

However, due to noise, F' may not be singular. So instead, next best thing is eigenvector associated with smallest eigenvalue of F'

Essential/Fundamental Matrix

The essential and fundamental matrices are 3x3 matrices that “encode” the epipolar geometry of two views.

Motivation: Given a point in one image, multiplying by the essential/fundamental matrix will tell us which epipolar line to search along in the second view.

Robert Collins
CSE486, Penn State

E/F Matrix Summary

Longuet-Higgins equation $$p_r^T E p_l = 0$$

Epipolar lines:

$$\tilde{p_r}^T \tilde{l_r} = 0 \qquad \tilde{p_l}^T \tilde{l_l} = 0$$

$$\tilde{l_r} = E p_l \qquad \tilde{l_l} = E^T p_r$$

Epipoles:

$$e_r^T E = 0 \qquad E e_l = 0$$

E vs F: E works in film coords (calibrated cameras)
F works in pixel coords (uncalibrated cameras)

Computing F from Point Matches

- Assume that you have m correspondences
- Each correspondence satisfies:

$$\bar{p}_{r_i}^{\,T} F \bar{p}_{l_i} = 0 \quad i = 1, \ldots, m$$

- F is a 3x3 matrix (9 entries)
- Set up a **HOMOGENEOUS** linear system with 9 unknowns

Computing F

$$\bar{p}_{l_i} = (x_i \;\; y_i \;\; 1)^T \quad \bar{p}_{r_i} = (x'_i \;\; y'_i \;\; 1)^T$$

$$\bar{p}_{r_i}^{\,T} F \bar{p}_{l_i} = 0 \quad i = 1, \ldots, m$$

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

Robert Collins
CSE486, Penn State

Computing F

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$x_i x_i' f_{11} + x_i y_i' f_{21} + x_i f_{31} +$$
$$y_i x_i' f_{12} + y_i y_i' f_{22} + y_i f_{32} +$$
$$x_i' f_{13} + y_i' f_{23} + f_{33} = 0$$

Robert Collins
CSE486, Penn State

Computing F

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

Given m point correspondences…

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

Think: how many points do we need?

How Many Points?

Unlike a homography, where each point correspondence contributes two constraints (rows in the linear system of equations), for estimating the essential/fundamental matrix, each point only contributes one constraint (row). [because the Longuet-Higgins / Epipolar constraint is a scalar eqn.]

Thus need at least 8 points.

Hence: The Eight Point algorithm!

Solving Homogeneous Systems

Self-study

Assume that we need the non trivial solution of:

$$A\mathbf{x} = 0$$

with m equations and n unknowns, m >= n – 1 and rank(A) = n-1

Since the norm of x is arbitrary, we will look for a solution with norm ||x|| = 1

Least Square solution

Self-study

We want Ax as close to 0 as possible and $||x|| = 1$:

$$\min_{\mathbf{x}} ||A\mathbf{x}||^2 \text{ s.t. } ||x||^2 = 1$$

$$||A\mathbf{x}||^2 = (A\mathbf{x})^T(A\mathbf{x}) = \mathbf{x}^T A^T A\mathbf{x}$$

$$||\mathbf{x}||^2 = \mathbf{x}^T\mathbf{x} = 1$$

Optimization with constraints

Self-study

Define the following cost:

$$\mathcal{L}(\mathbf{x}) = \mathbf{x}^T A^T A \mathbf{x} - \lambda(\mathbf{x}^T \mathbf{x} - 1)$$

This cost is called the LAGRANGIAN *cost* and λ is called the LAGRANGIAN *multiplier*

The Lagrangian incorporates the constraints into the cost function by introducing extra variables.

Optimization with constraints

Self-study

$$\min_{\mathbf{x}} \left\{ \mathcal{L}(\mathbf{x}) = \mathbf{x}^T A^T A \mathbf{x} - \lambda(\mathbf{x}^T \mathbf{x} - 1) \right\}$$

Taking derivatives wrt to x and λ:

$$A^T A \mathbf{x} - \lambda \mathbf{x} = 0$$

$$\mathbf{x}^T \mathbf{x} - 1 = 0$$

- The first equation is an eigenvector problem
- The second equation is the original constraint

Optimization with constraints

Self-study

$$A^T A\mathbf{x} - \lambda\mathbf{x} = 0$$
$$A^T A\mathbf{x} = \lambda\mathbf{x}$$

- $\mathbf{x}$ is an eigenvector of $A^T A$ with eigenvalue λ: e_λ

$$\mathcal{L}(\mathrm{e}_\lambda) = \mathrm{e}_\lambda{}^T A^T A\mathrm{e}_\lambda - \lambda(\mathrm{e}_\lambda{}^T\mathrm{e}_\lambda - 1)$$
$$\mathcal{L}(\mathrm{e}_\lambda) = \lambda\mathrm{e}_\lambda{}^T\mathrm{e}_\lambda = \lambda$$

- We want the eigenvector with <u>smallest</u> eigenvalue

Self-study

We can find the eigenvectors and eigenvalues of A^TA by finding the Singular Value Decomposition of A

Self-study

Singular Value Decomposition (SVD)

Any m x n matrix *A* can be written as the product of 3 matrices:

$$A = UDV^T$$

Where:

- U is m x m and its columns are orthonormal vectors
- V is n x n and its columns are orthonormal vectors
- D is m x n diagonal and its diagonal elements are called the singular values of *A*, and are such that:

$$\sigma_1 \geq \sigma_2 \geq \dots \sigma_n \geq 0$$

Self-study

SVD Properties

$$A = UDV^T$$

- The columns of U are the eigenvectors of AA^T
- The columns of V are the eigenvectors of A^TA
- The squares of the diagonal elements of D are the eigenvalues of AA^T and A^TA

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 & 1 \\ & & & & \vdots & & & & \\ x_8 x'_8 & x_8 y'_8 & x_8 & y_8 x'_8 & y_8 y'_8 & y_8 & x'_8 & y'_8 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

$$A\mathbf{x} = 0$$ <u>A has rank 8</u>

$$\min_{\mathbf{x}} ||A\mathbf{x}||^2 \text{ s.t. } ||x||^2 = 1$$

- Find eigenvector of $A^T A$ with <u>smallest</u> eigenvalue!

Algorithm EIGHT_POINT

The input is formed by m point correspondences, m >= 8

- Construct the m x 9 matrix A
- Find the SVD of A: $A = UDV^T$
- The entries of F are the components of the column of V corresponding to the least s.v.

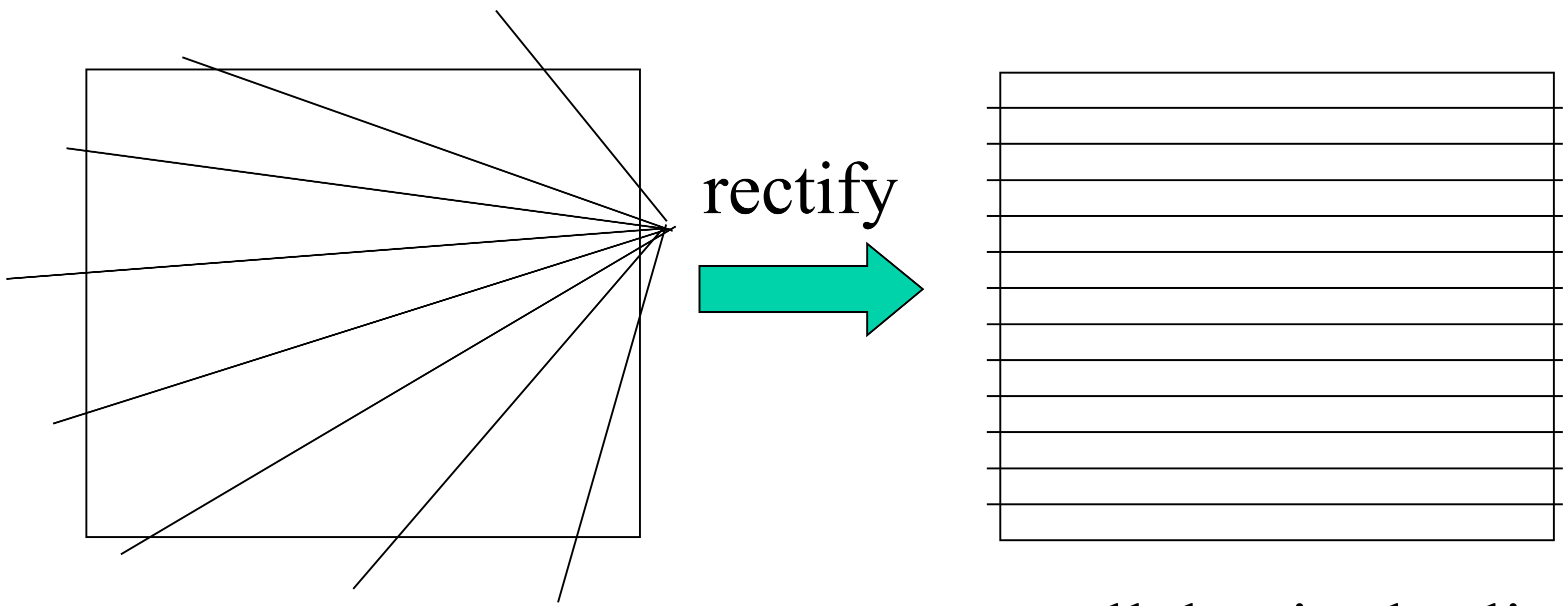
Algorithm EIGHT_POINT

F must be singular (remember, it is rank 2, since it is important for it to have a left and right nullspace, i.e. the epipoles). To enforce rank 2 constraint:

- Find the SVD of F: $F = U_f D_f V_f^T$
- Set smallest s.v. of F to 0 to create D'_f
- Recompute F: $F = U_f D'_f V_f^T$

A Practical Issue

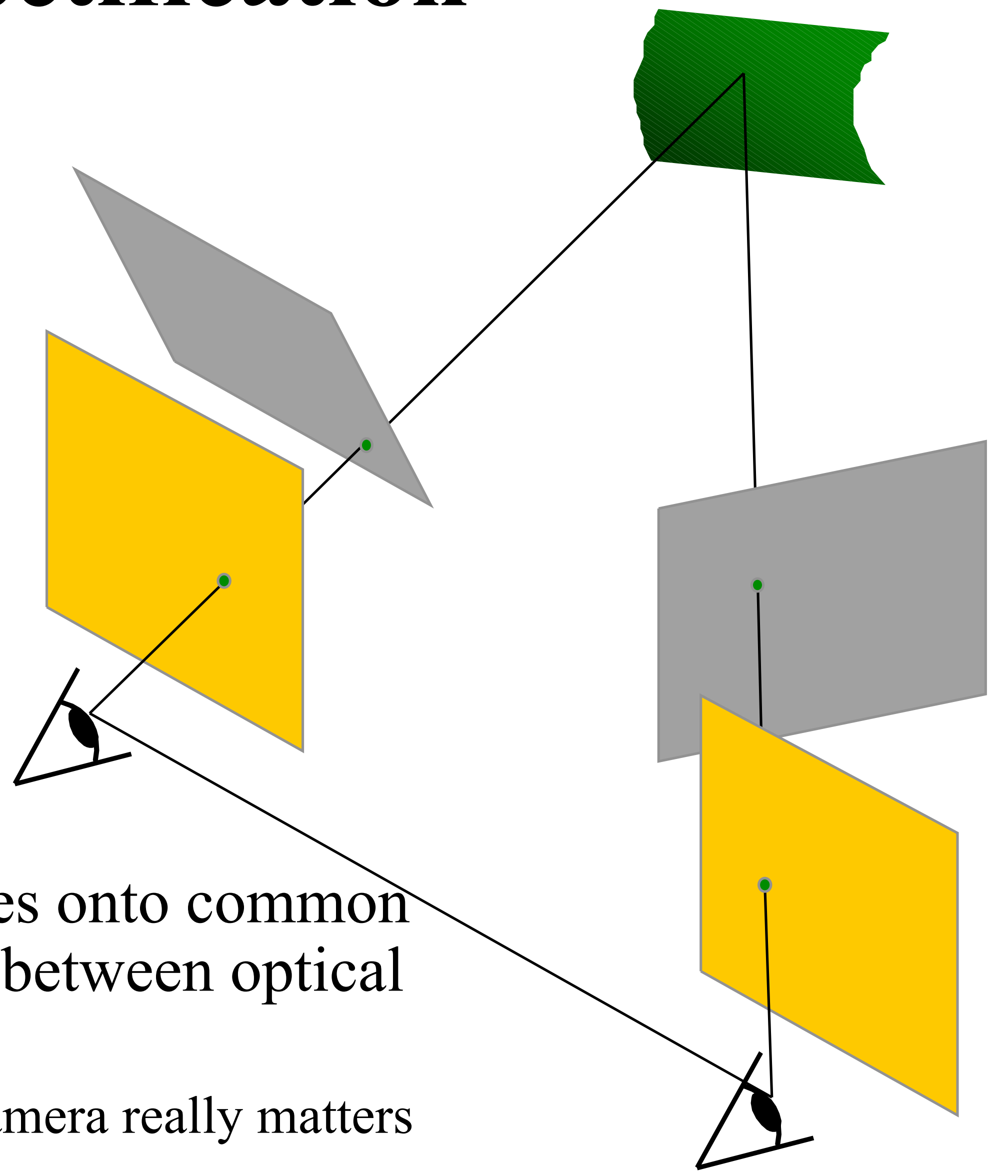
How to "rectify" the images so that any scan-line stereo alorithm that works for simple stereo can be used to find dense matches (i.e. compute a disparity image for every pixel).



general epipolar lines

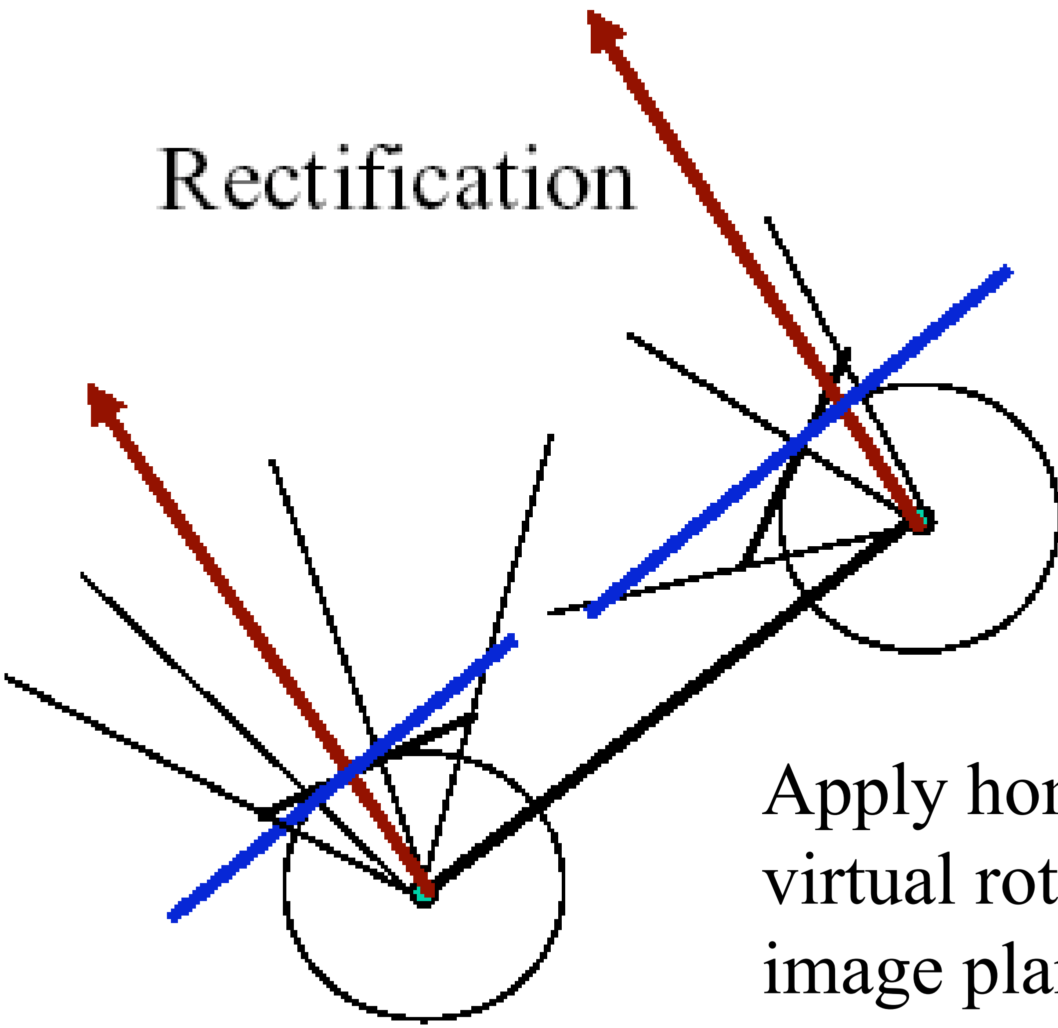
parallel epipolar lines

Stereo Rectification



- Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers
- Notice, only focal point of camera really matters

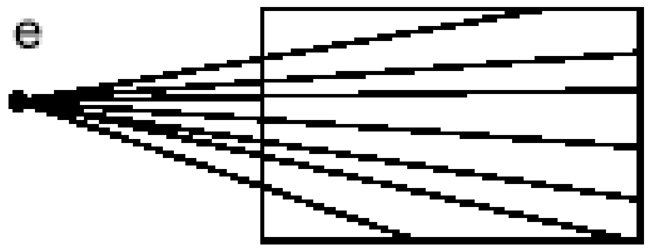
General Idea

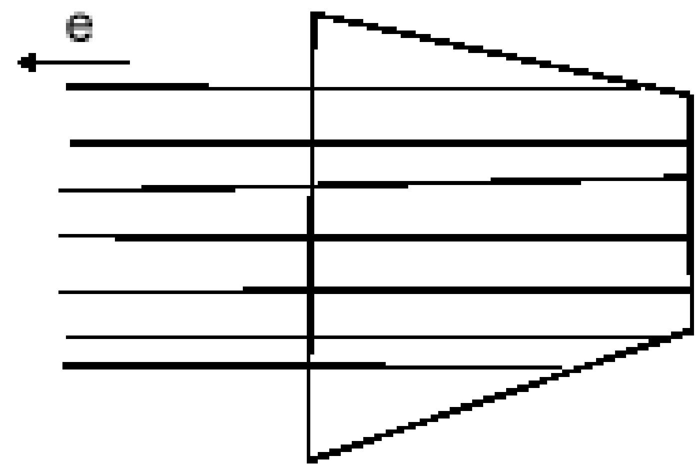


Apply homography representing a virtual rotation to bring the image planes parallel with the baseline (epipoles go to infinity).

General Idea, continued

Apply projective transformation so that epipolar lines correspond to horizontal scanlines





$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = He$$

map epipole e to (1,0,0)

try to minimize image distortion

Note that rectified images usually not rectangular

Image Rectification

Build the rotation:

$$R_{rect} = \begin{bmatrix} \mathbf{e_1}^T \\ \mathbf{e_2}^T \\ \mathbf{e_3}^T \end{bmatrix}$$

with:

$$\mathbf{e}_1 = \frac{\mathbf{T}}{||\mathbf{T}||}$$

where T is just a unit vector representing the epipole in the left image. We know how to compute this from E, from last class.

Algorithm Rectification

- Build the matrix R_{rect}
- Set $R_l = R_{rect}$ and $R_r = R.R_{rect}$
- For each left point $p_l = (x,y,f)^T$
  - compute $R_l\, p_l = (x', y', z')^T$
  - Compute $p'_l = f/z'\, (x', y'\, z')^T$
- Repeat above for the right camera with R_r

Example

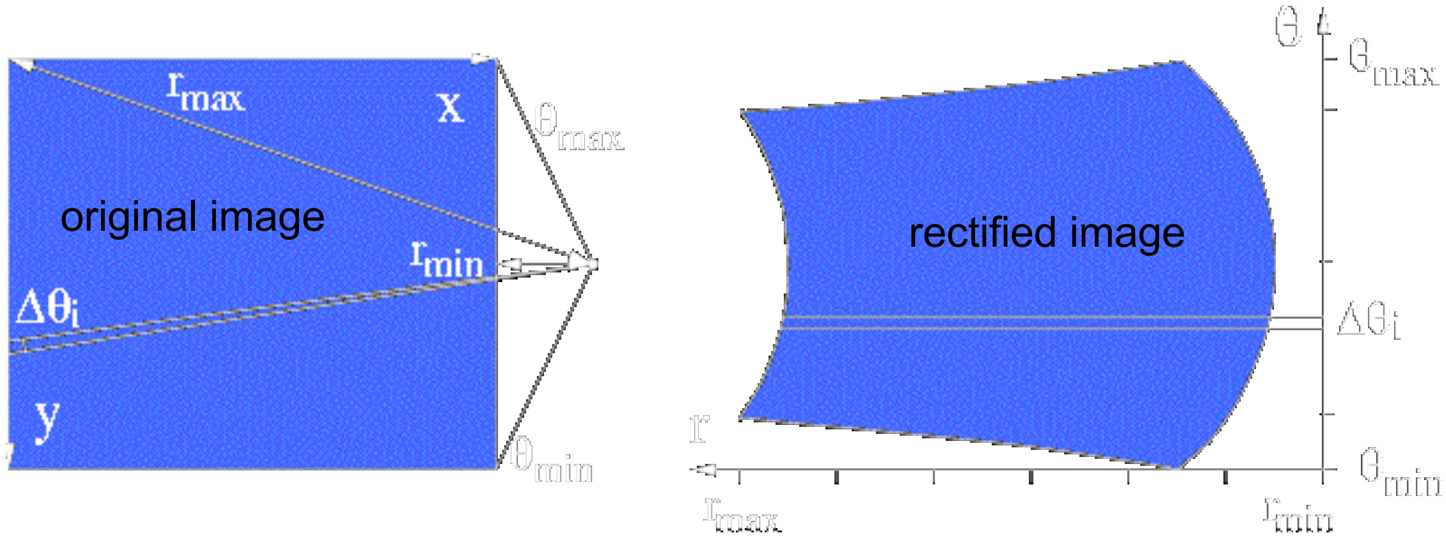
Example

Rectified Pair

Polar rectification

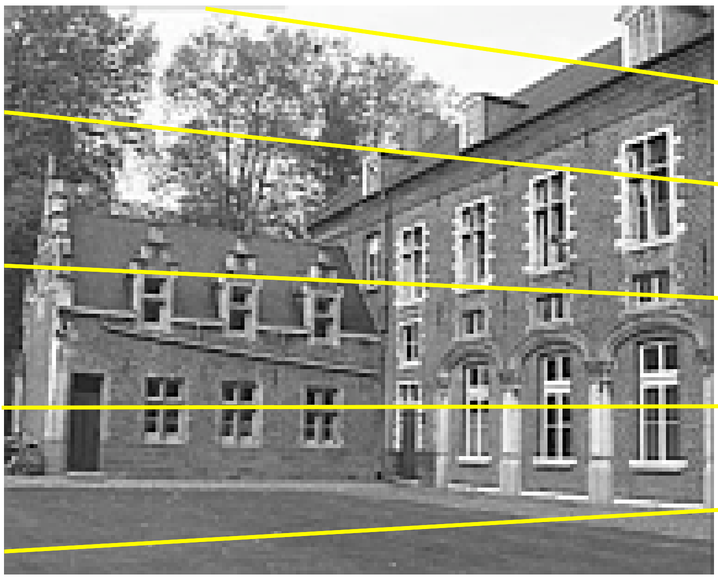
(Pollefeys et al. ICCV'99)

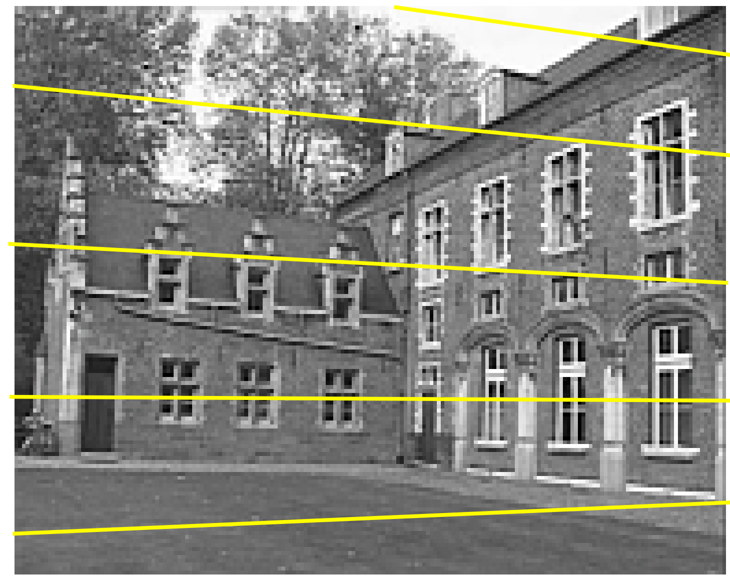
Polar re-parameterization around epipoles
Requires only (oriented) epipolar geometry
Preserve length of epipolar lines
Choose $\Delta\theta$ so that no pixels are compressed



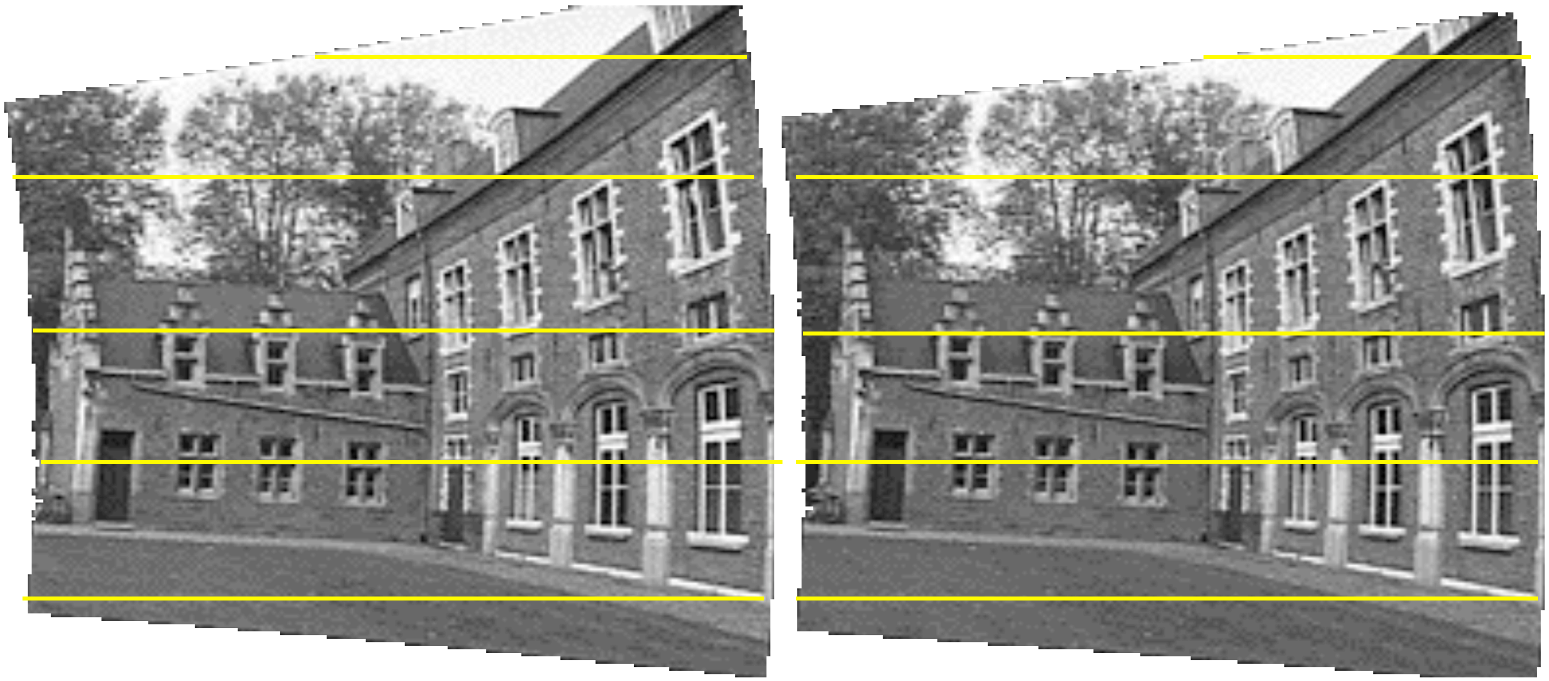
Works for all relative motions
Guarantees minimal image size

polar rectification: example



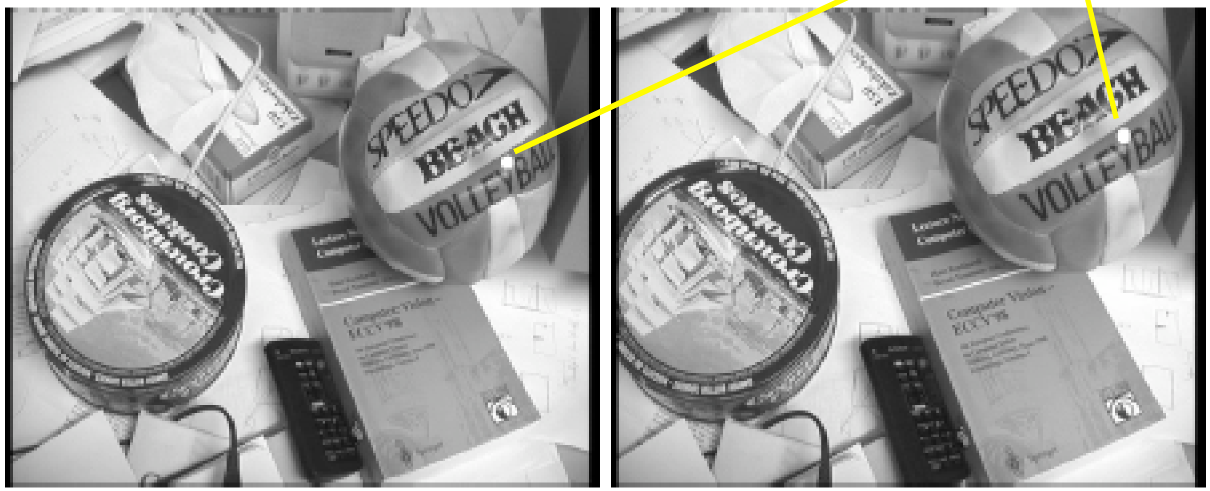


polar rectification: example



Example

epipoles

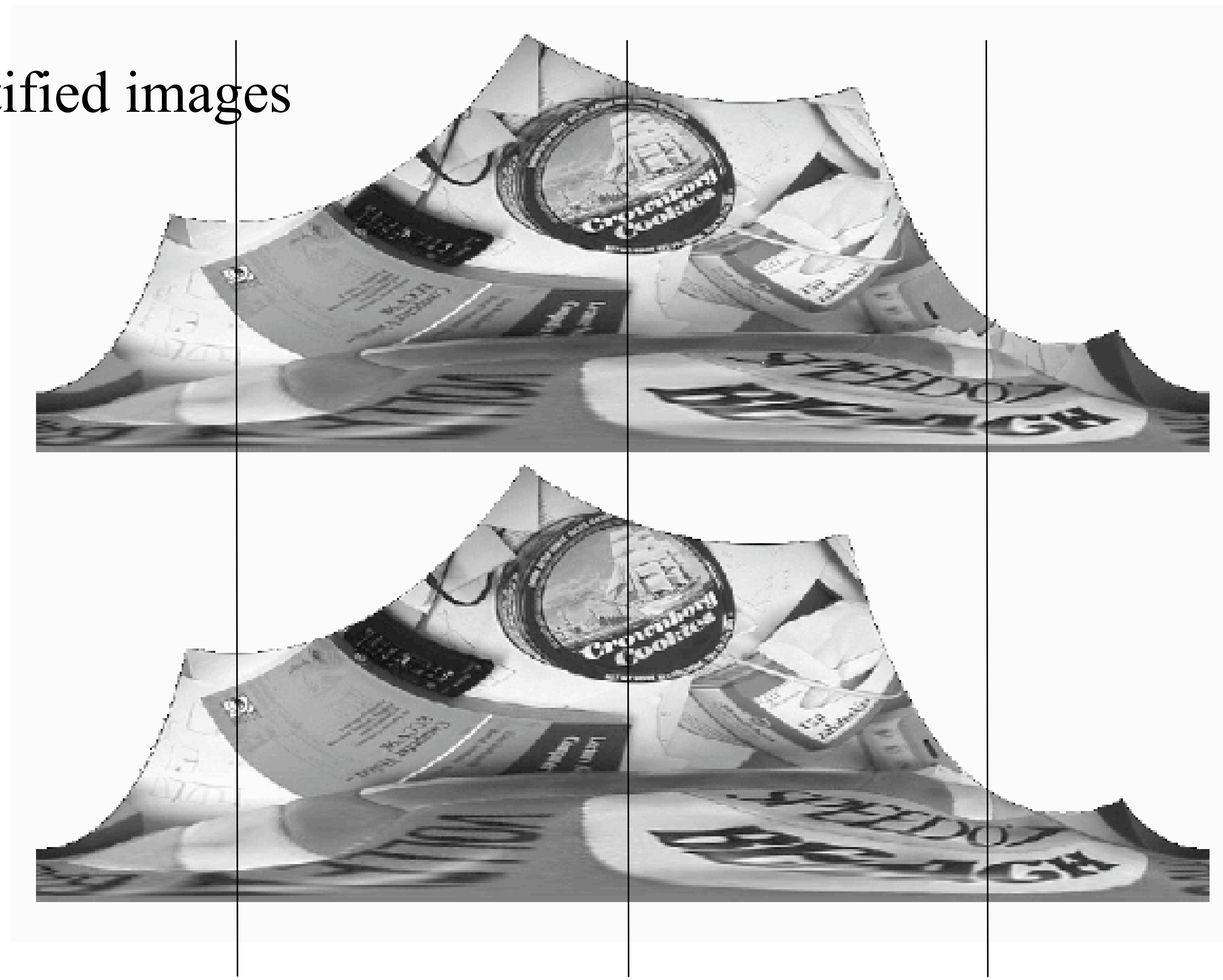


Input images

Robert Collins
CSE486, Penn State

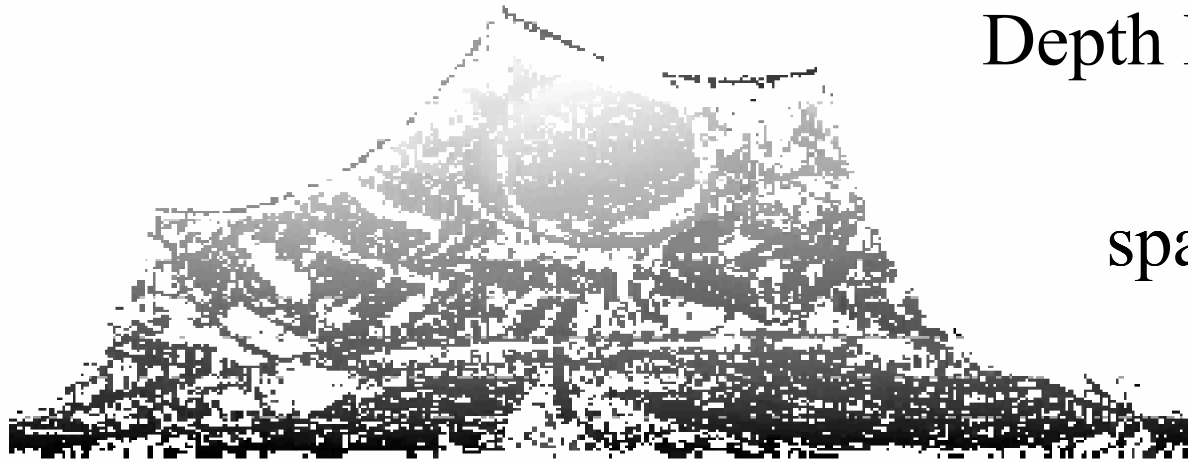
Example (cont)

Rectified images

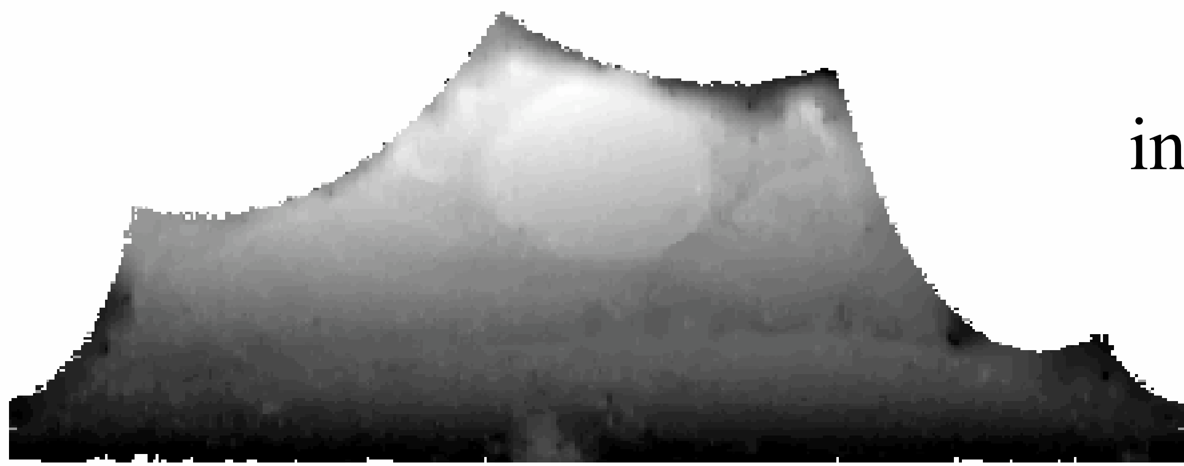


Example (cont)

Depth Maps

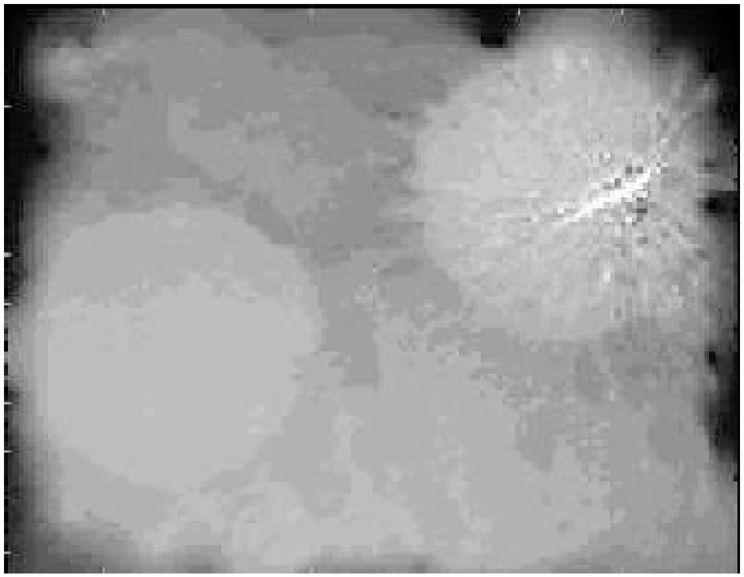


sparse



interpolated

Example (cont)

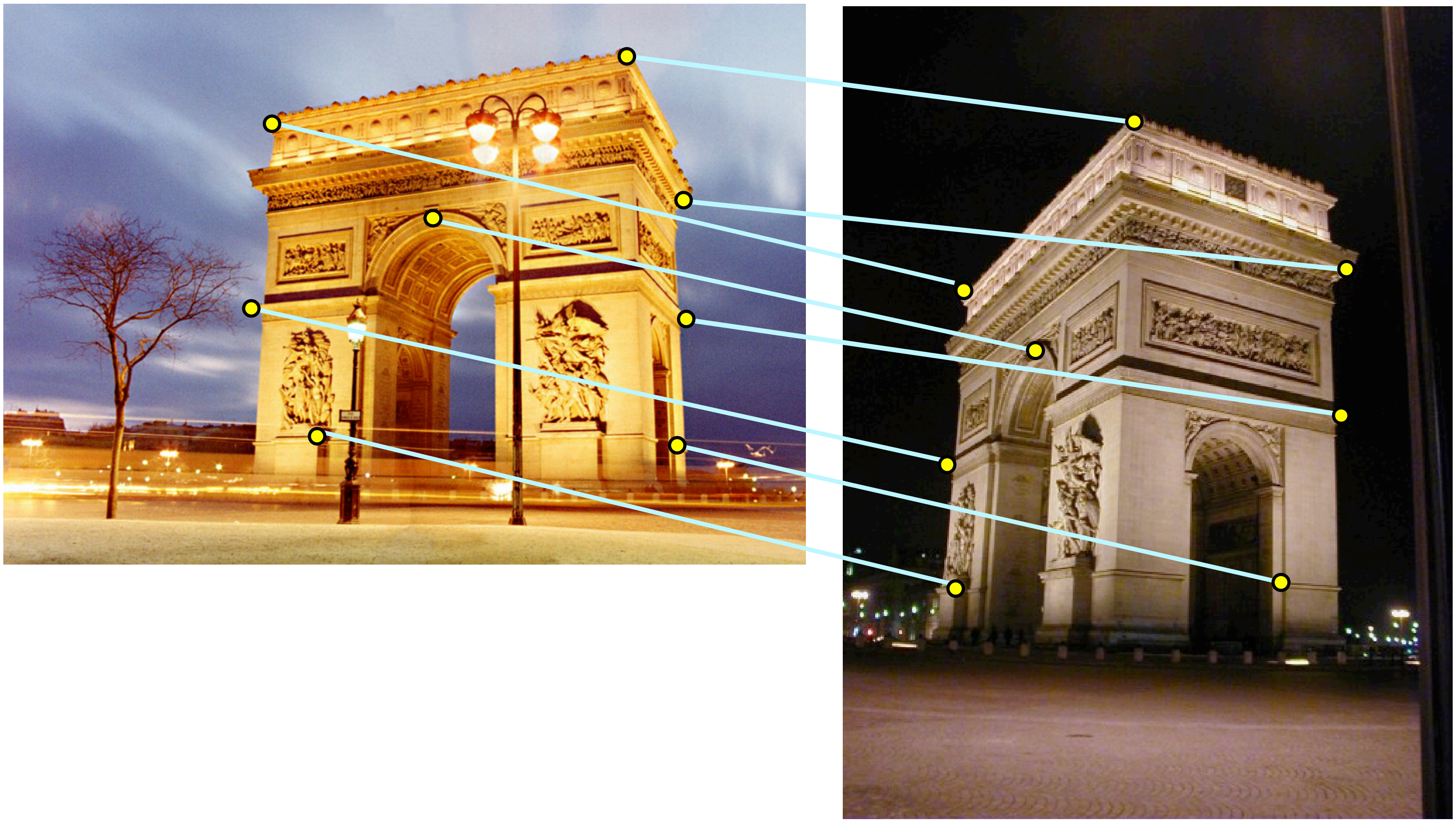


Depth map in pixel coords

Views of texture mapped depth surface

Steps to General Stereo

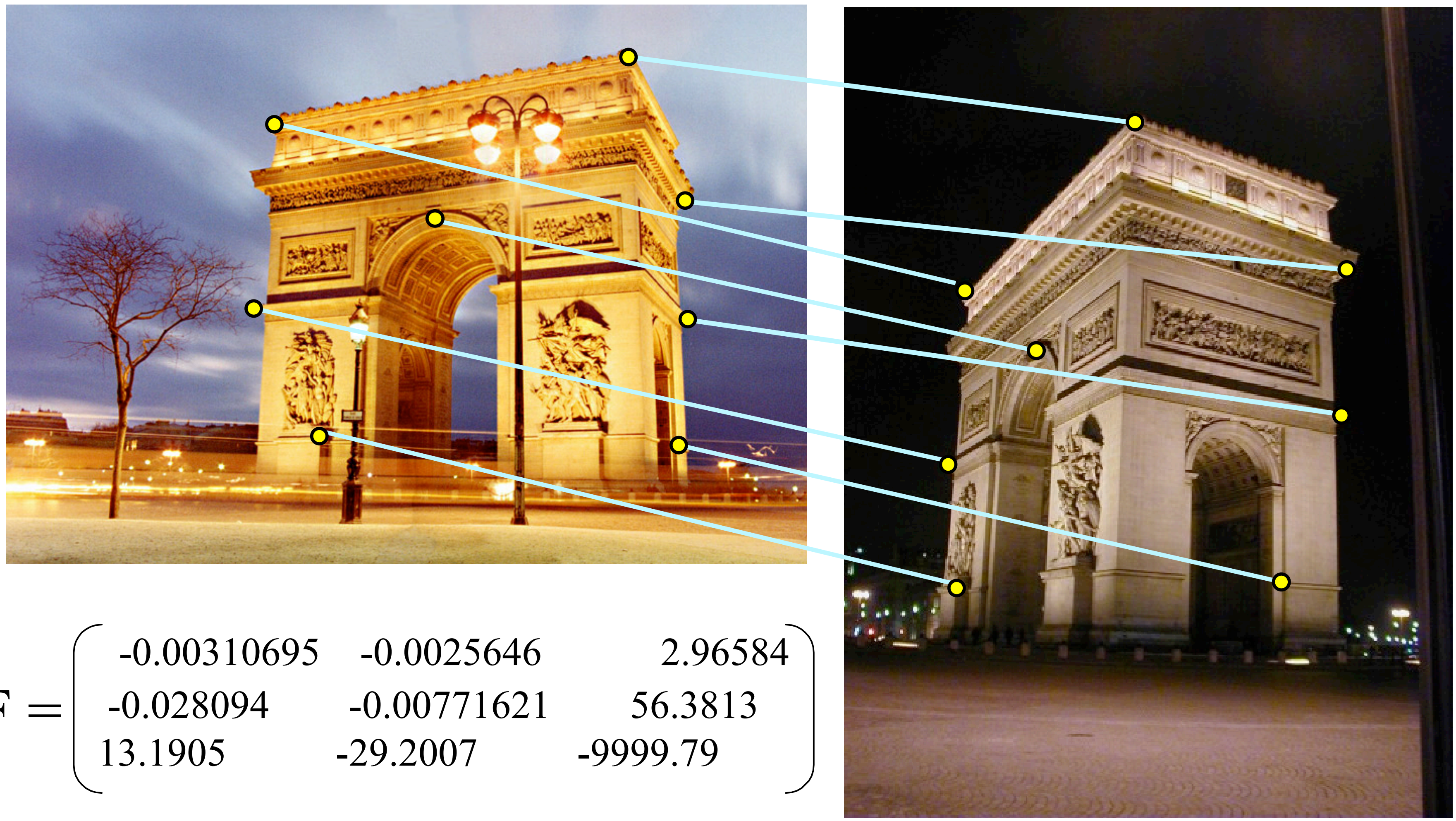
find 8 or more initial point matches (somehow)



Robert Collins
CSE486, Penn State

Steps to General Stereo

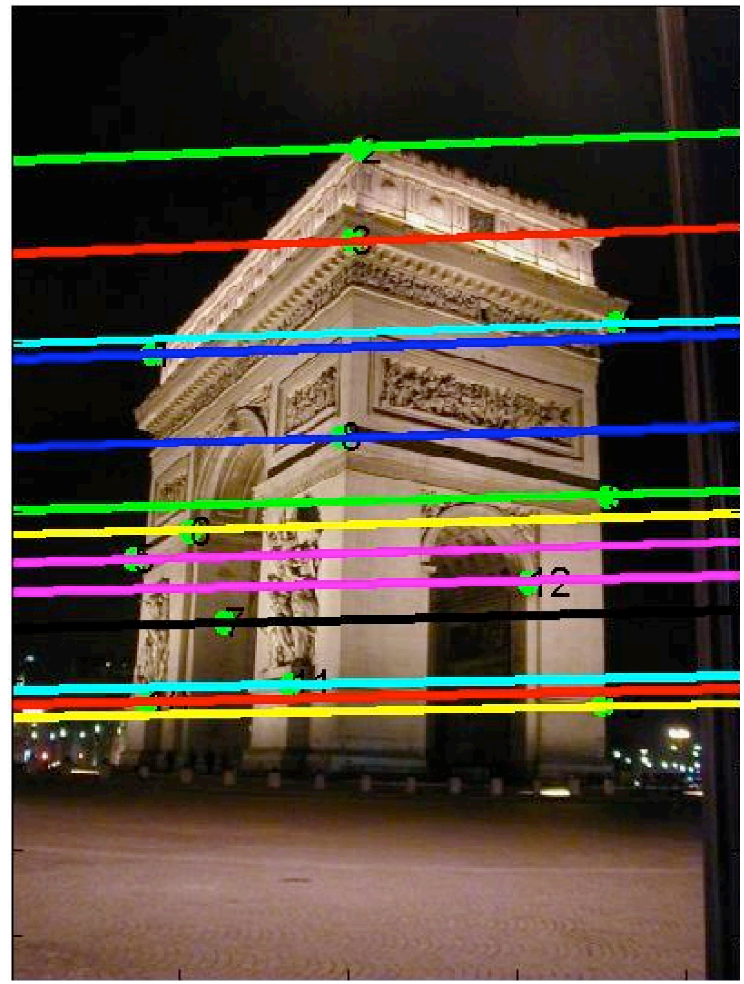
Compute F matrix using 8-point algorithm



$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

Steps to General Stereo

Infer epipolar geometry (epipoles, epipolar lines) from F.

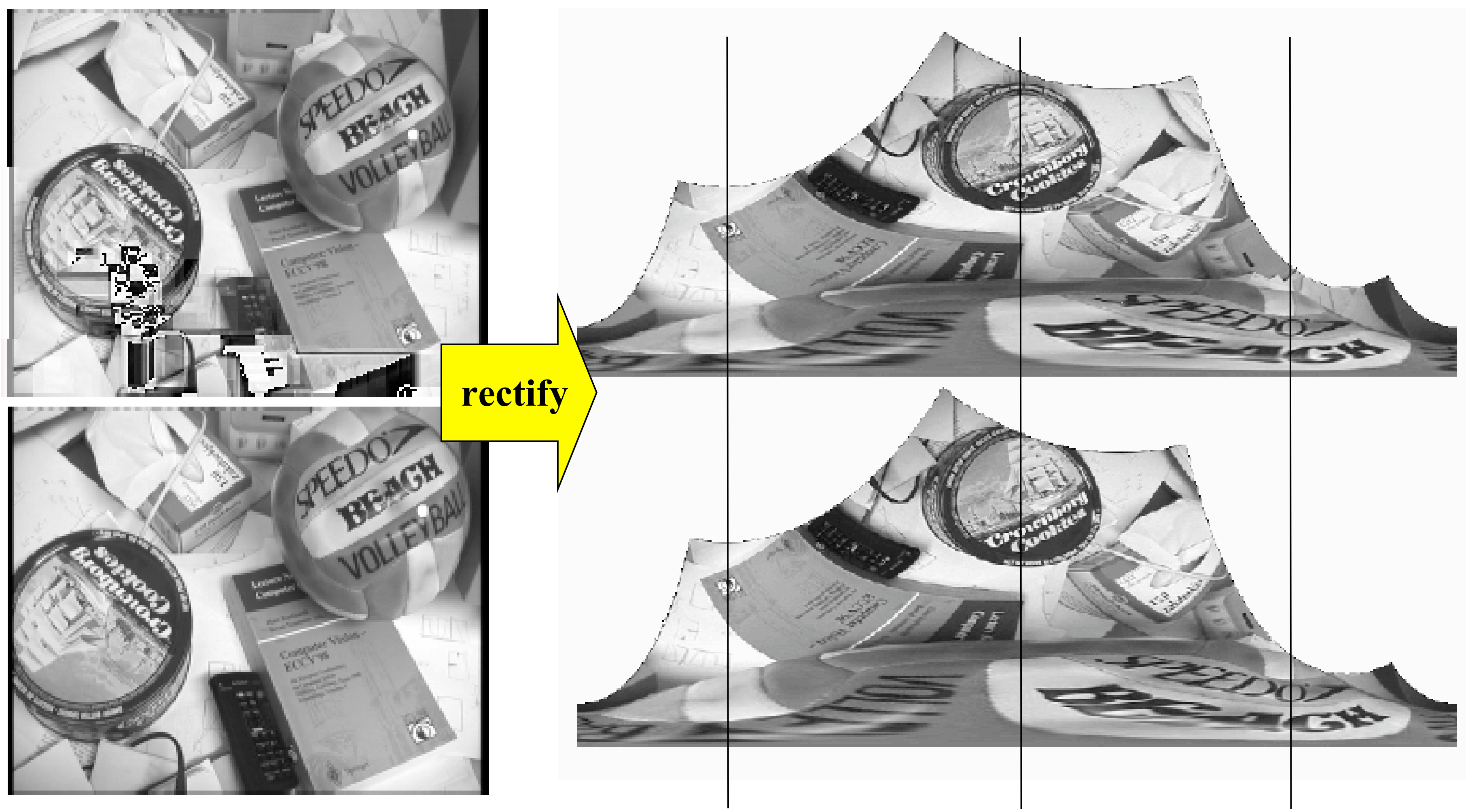


$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

Robert Collins
CSE486, Penn State

Steps to General Stereo

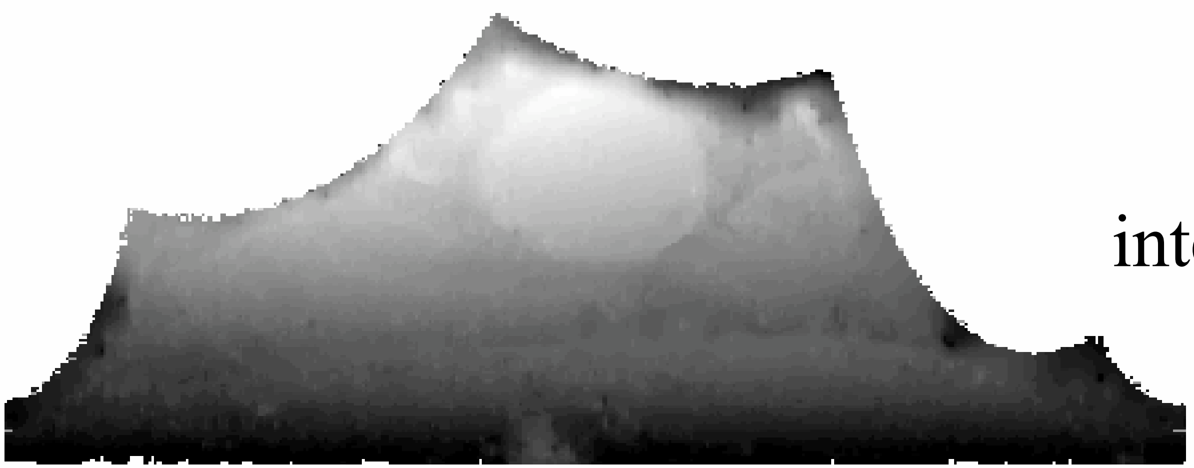
Rectify images to get simple scanline stereo pair.



Steps to General Stereo

Compute disparity map (correspondence matching)

sparse

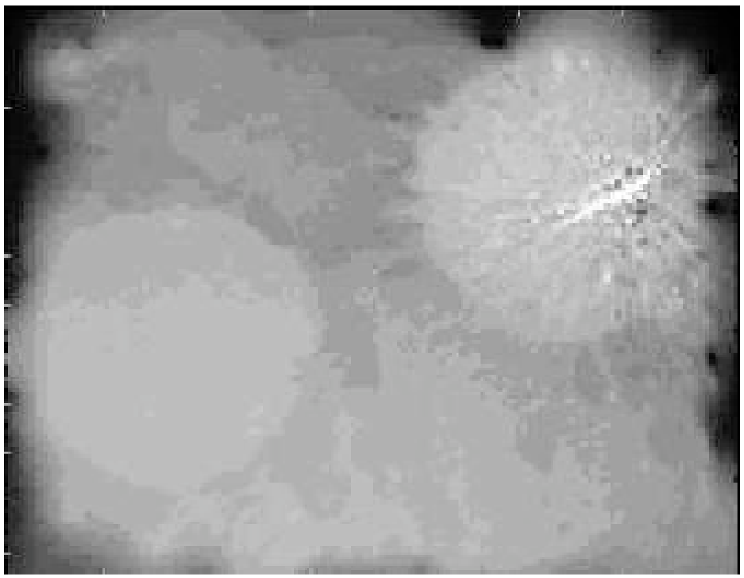


interpolated

Robert Collins
CSE486, Penn State

Steps to General Stereo

Compute 3D surface geometry from disparity map

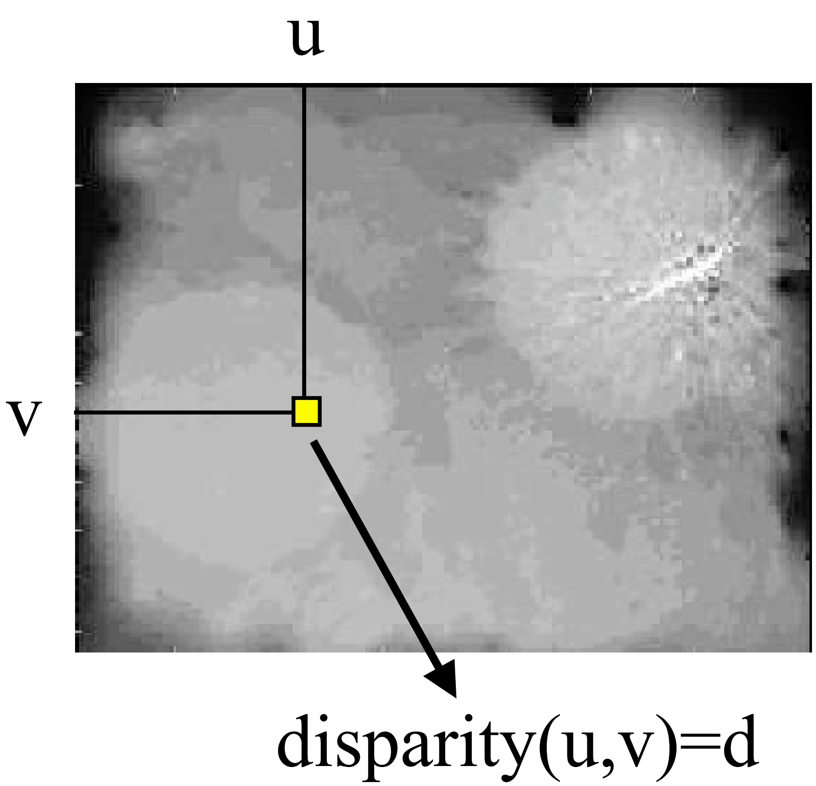


disparity map in
pixel coords

Views of texture mapped
depth surface

Reconstruction

Even though we have a dense disparity map, when talking about recovering 3D scene structure, we will consider it to just be a set of point matches.



<u>point match:</u>

(u,v) in left image
matches
(u-d,v) in right image

Stereo Reconstruction

Given point correspondences, how to compute 3D point positions using triangulation.

<u>Results depend on how calibrated the system is:</u>

1) Intrinsic and extrinsic parameters known
   Can compute metric 3D geometry
2) Only intrinsic parameters known
   Unknown scale factor
3) Neither intrinsic nor extrinsic known
   Recover structure up to an unknown projective transformation of the scene

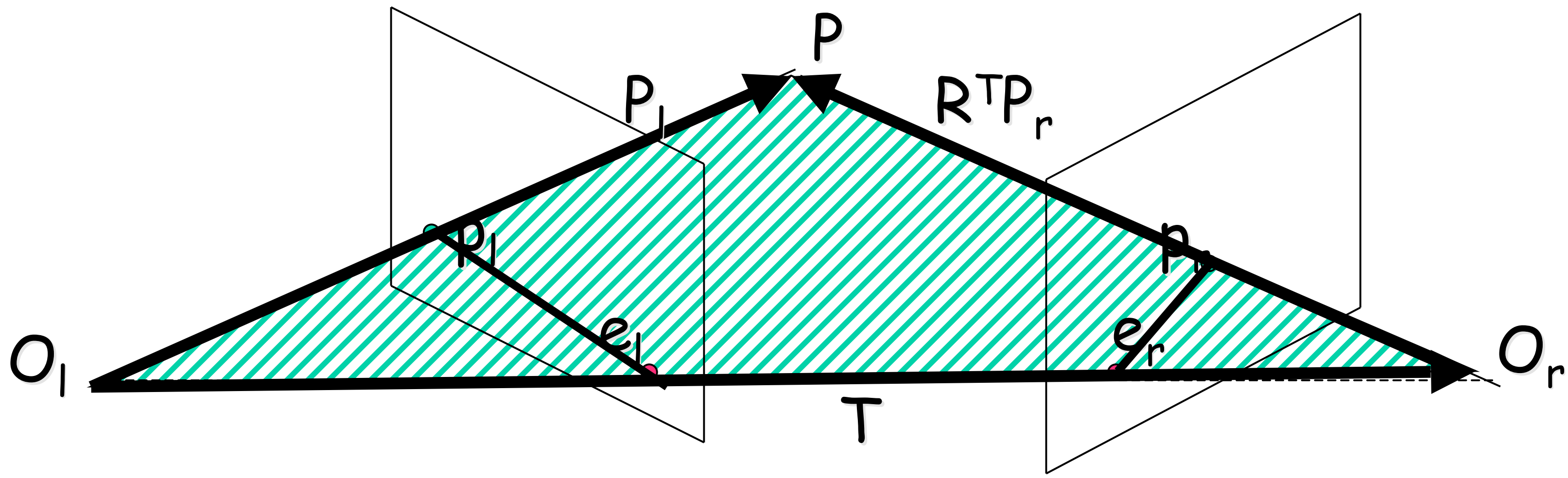
Fully Calibrated Stereo

Known intrinsics -- can compute viewing rays in camera coordinate system

Know extrinsics -- know how rays from both cameras are positioned in 3D space

Reconstruction: triangulation of viewing rays

Calibrated Triangulation



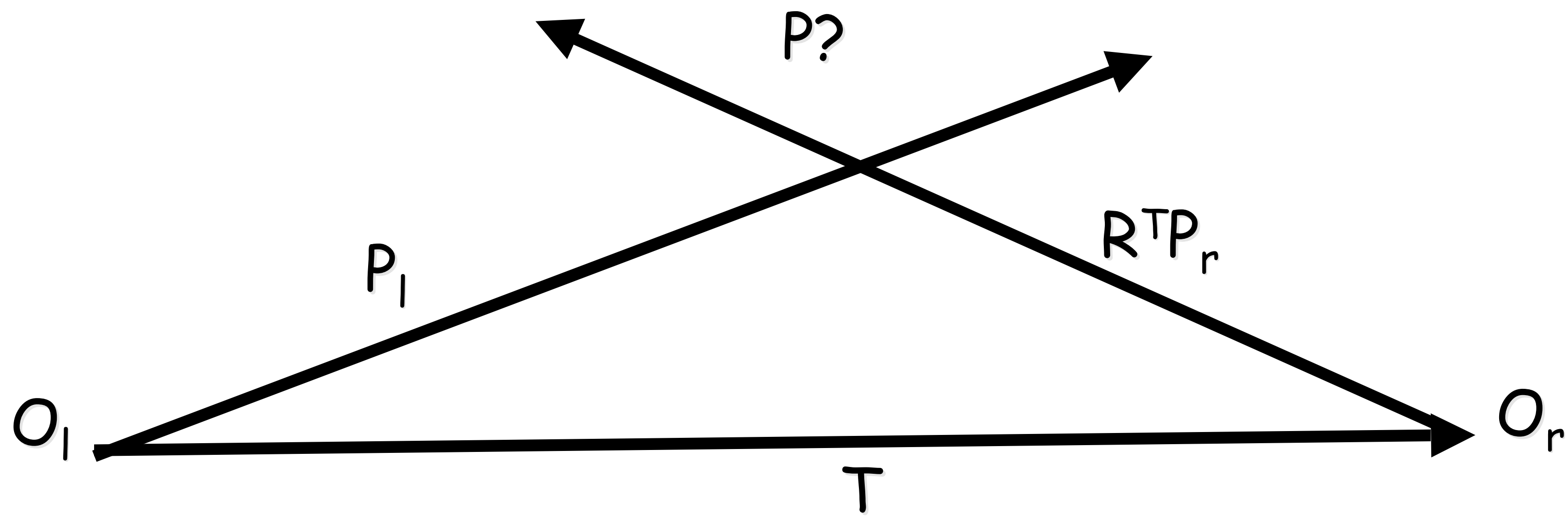
ideally, P is the point of intersection of two 3D rays:

ray through O_l with direction P_l

ray through O_r with direction R^TP_r

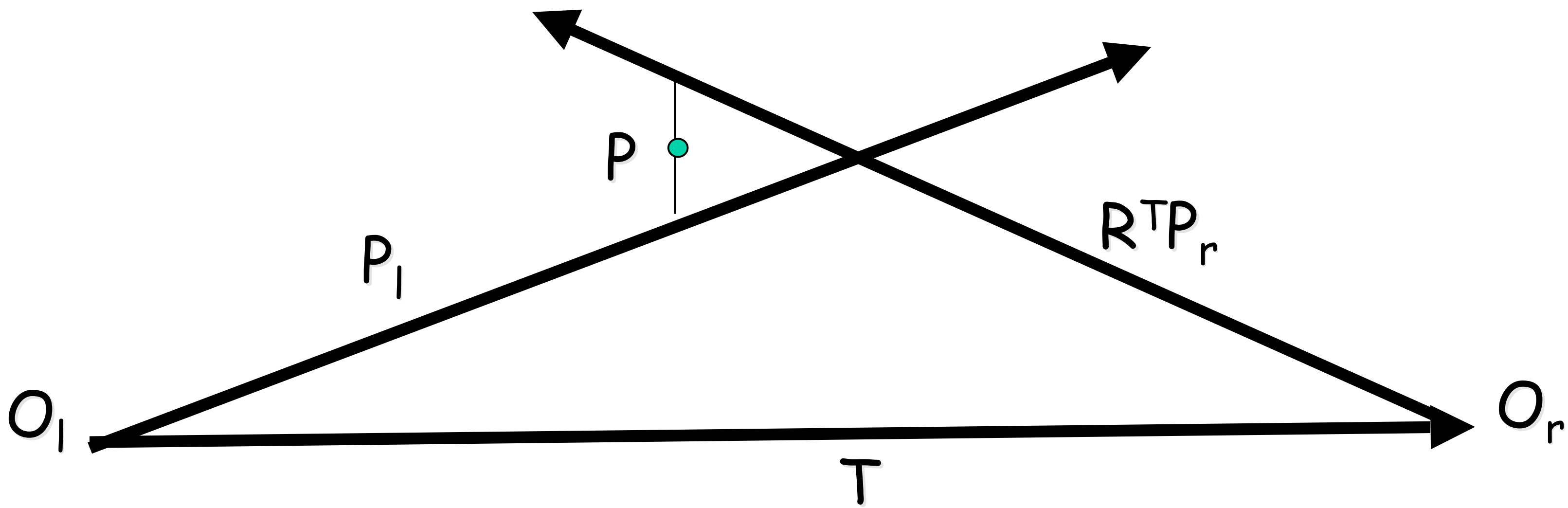
Triangulation with Noise

Unfortunately, these rays typically don't intersect due to noise in point locations and calibration params



Triangulation with Noise

Unfortunately, these rays typically don't intersect due to noise in point locations and calibration params

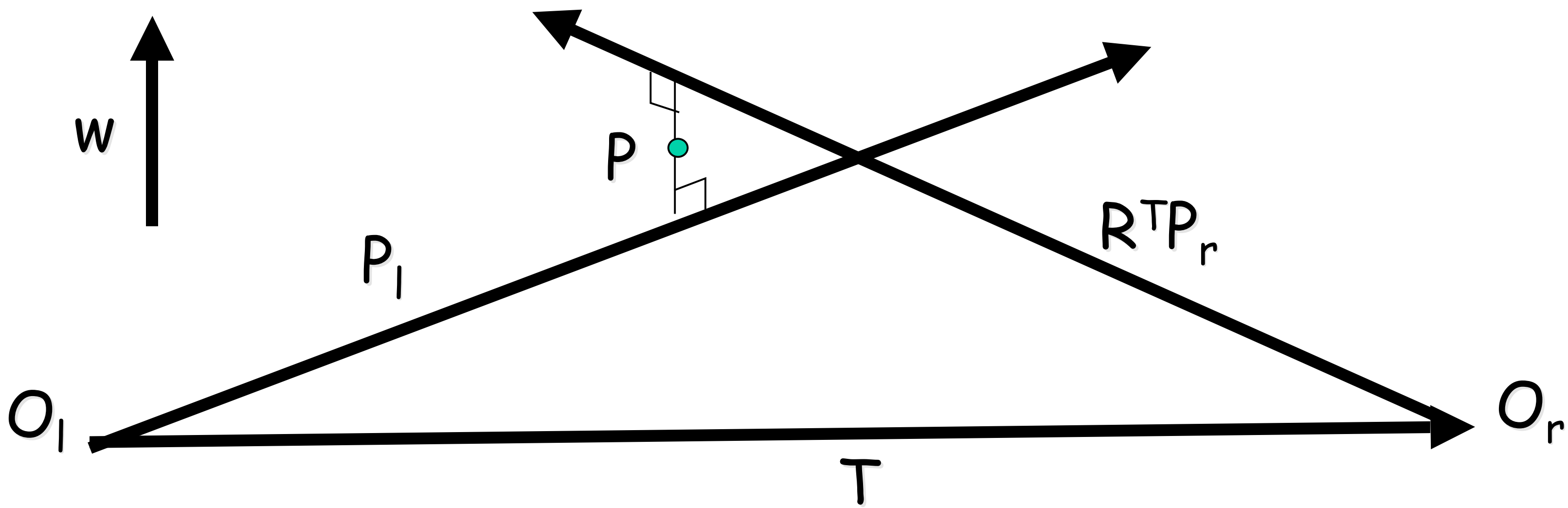


Solution: Choose P as the "pseudo-intersection point". This is point that minimizes the sum of squared distance to both rays. (The SSD is 0 if the rays exactly intersect)

Solution from T&V Book

P is midpoint of the segment perpendicular to P_l and $R^T P_r$

Let $w = P_l \times R^T P_r$ (this is perpendicular to both)



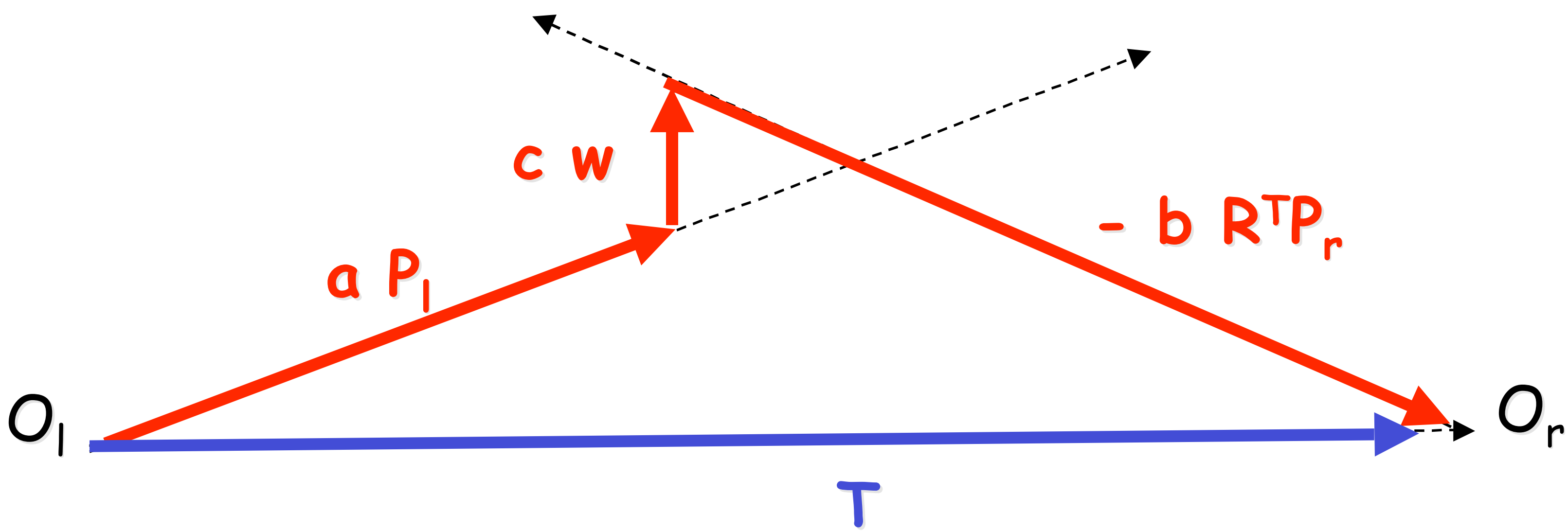
Introducing three unknown scale factors a,b,c we note we can write down the equation of a "circuit"

Robert Collins
CSE486, Penn State

Solution from T&V Book

Writing vector "circuit diagram" with unknowns a,b,c

$$a\,P_l \;+ c\,(P_l \,X\, R^T P_r)\; - b\,R^T P_r \;=\; T$$

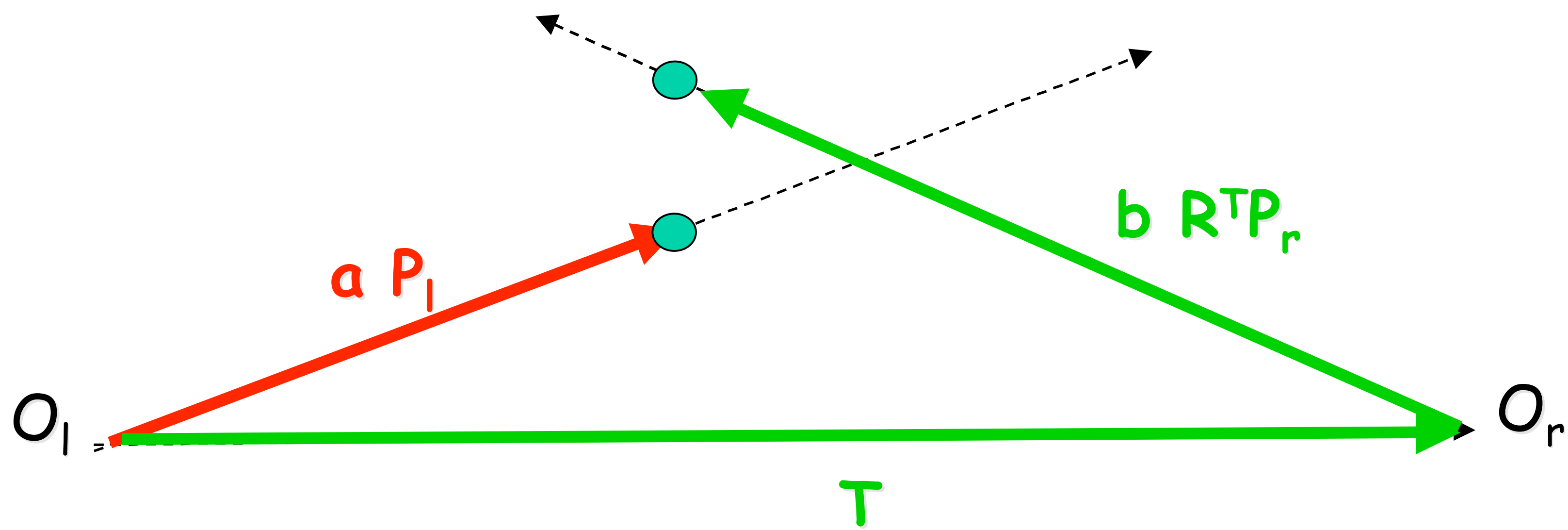


note: this is three linear equations in three unknowns a,b,c
=> can solve for a,b,c

Robert Collins
CSE486, Penn State

Solution from T&V Book

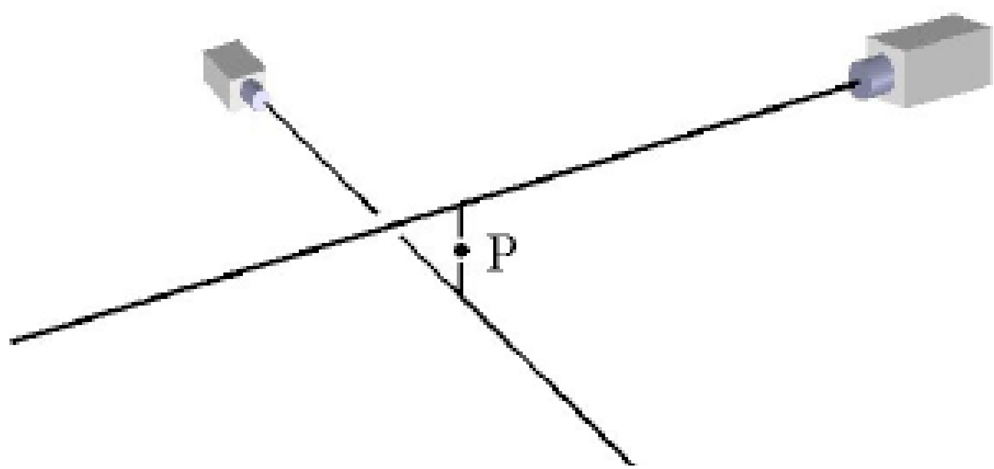
After finding a,b,c , solve for midpoint of line segment between points $\mathbf{O_l + a\,P_l}$ and $\mathbf{O_l + T + b\,R^T\,P_r}$



Robert Collins
CSE486, Penn State

Alternate Solution

I prefer an alternate solution based on using least squares to solve for an unknown point P that minimizes SSD to viewing rays. Why? it generalizes readily to N cameras.



$$[\sum_{i}^{n} w_i (I - u_i u_i')]\, P \;=\; \sum_{i}^{n} w_i (I - u_i u_i') c_i$$

Stereo Reconstruction

Given point correspondences, how to compute 3D point positions using triangulation.

Results depend on how calibrated the system is:

1) Intrinsic and extrinsic parameters known
   Can compute metric 3D geometry
2) Only intrinsic parameters known
   Unknown scale factor
3) Neither intrinsic nor extrinsic known
   Recover structure up to an unknown projective transformation of the scene

Only Intrinsic Params Known

General outline of solution (see book for details)

Use knowledge that E = R S to solve for R and T, then use previous triangulation method.

Note: since E is only defined up to a scale factor, we can only determine the direction of T, not its length. So... 3D reconstruction will have an unknown scale.

Only Intrinsic Params Known

Using E to solve for extrinsic params R and T

E = R S where elements of S are functions of T

Then $E^T E = S^T R^T R S = S^T S$ (because $R^T R = I$)

Thus $E^T E$ is only a function of T.
Solve for elements of T assuming it is a unit vector.

After determining T, plug back into E = R S to determine R.

Only Intrinsic Params Known

Unfortunately, four different solutions for (R,T) are possible (due to choice of sign of E, and choice of sign of T when solving for it).

However, only one choice will give consistent solutions when used for triangulation, where consistent means reconstructed points are in front of the cameras (positive Z coordinates).

So, check all four solutions, choose the correct one, and you are done.

Stereo Reconstruction

Given point correspondences, how to compute 3D point positions using triangulation.

Results depend on how calibrated the system is:

1) Intrinsic and extrinsic parameters known
   Can compute metric 3D geometry
2) Only intrinsic parameters known
   Unknown scale factor
3) Neither intrinsic nor extrinsic known
   Recover structure up to an unknown projective transformation of the scene

Stereo when "Nothing" is Known

What if we don't known intrinsic nor extrinsic params? (we just look at two pictures of the scene with no prior information)

Can we recover any 3D information?

It wasn't clear that you could, but then in 1992...

- Faugeras "What can be seen in three dimensions from an uncalibrated stereo rig", ECCV 1992
- Hartley et.al., "Stereo from Uncalibrated Cameras", CVPR 1992
- Mohr et.al., "Relative 3D Reconstruction using Multiple Uncalibrated Images, LIFIA technical report, 1992

Stereo when "Nothing" is Known

Of course, we don't really know "nothing".

We know point correspondences, and because of that, we can compute the fundamental matrix F

Sketch of solution: use knowledge of F and use 5 points in the scene to define an arbitrary projective coordinate system. These points will have coordinates:

(1 0 0 0) (0 1 0 0) (0 0 1 0) (0 0 0 1) (1 1 1 1)

Result: You can recover 3D locations of other points with respect to that projective coordinate system.

Stereo when "Nothing" is Known

Result: You can recover 3D locations of other points with respect to that projective coordinate system.

Why would that be practical?

Often, you can then use other information to determine how your arbitrary projective coordinate system relates to the "real" Euclidean scene coordinates system.
e.g. use prior knowledge of lengths and angles
of some items in the world (like a house)