PITT | SWANSON ENGINEERING
ELECTRICAL & COMPUTER

ECE 1390/2390 Image Processing and Computer Vision – Fall 2021

Multiple Views – Essential and Fundamental Matrices

Ahmed Dallal

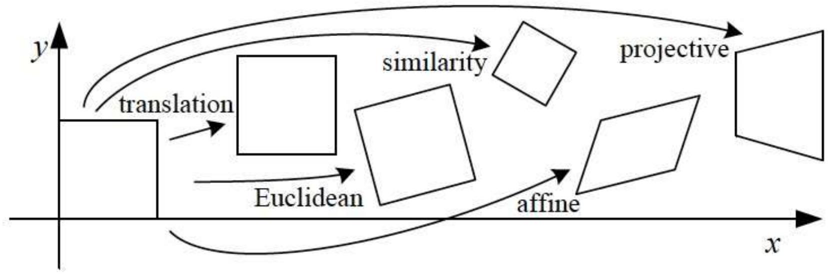
Two views…and two lectures

- Projective transforms from image to image
- Some more projective geometry
  - Points and lines and planes
- Two arbitrary views of the same scene
  - Calibrated – "Essential Matrix"
  - Two uncalibrated cameras "Fundamental Matrix"
    - Gives epipolar lines

Essential matrix

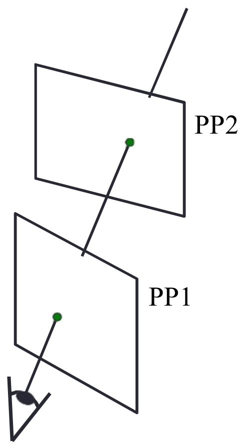
Last time

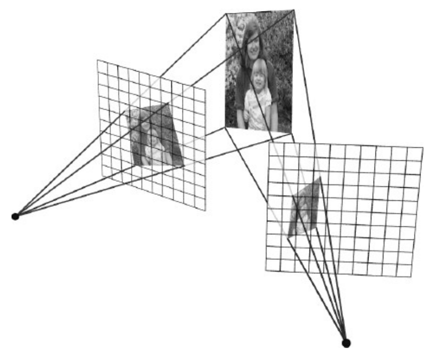
- Projective Transforms: Matrices that provide transformations including translations, rotations, similarity, affine and finally general (or perspective) projection.
- When 2D matrices are 3x3.



Last time: Homographies

- Provide mapping between images (image planes) taken from same center of projection; also mapping between any images of a planar surface.

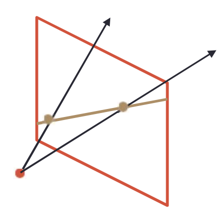




Last time: Projective geometry - Lines

- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

in vector notation:
$$0 = \underbrace{\begin{bmatrix} a & b & c \end{bmatrix}}_{\mathbf{l}} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{p}}$$



- A line is also represented as a homogeneous 3-vector **l**

Projective Geometry: lines and points

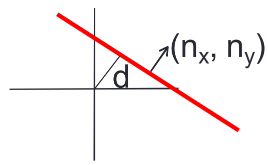
2D Lines: $ax + by + c = 0$

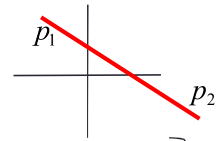
$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

Eq of line

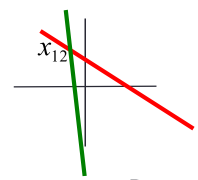
$$\mathbf{l}^T \mathbf{x} = 0$$

$$l = \begin{bmatrix} a & b & c \end{bmatrix} \Rightarrow \begin{bmatrix} n_x & n_y & d \end{bmatrix}$$





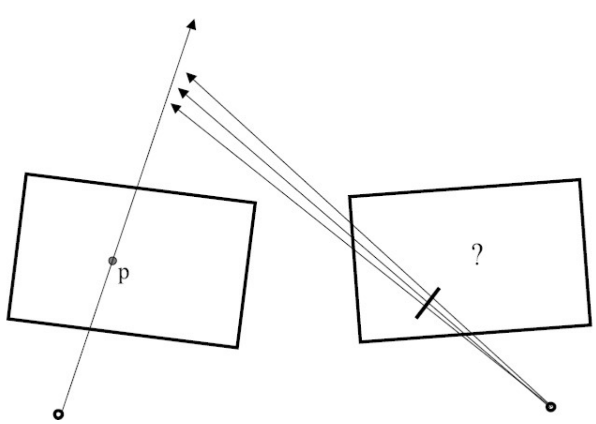
$$\left.\begin{aligned} p_1 &= \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \\ p_2 &= \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \end{aligned}\right\} l = p_1 \times p_2$$



$$\left.\begin{aligned} l_1 &= \begin{bmatrix} a_1 & b_1 & c_1 \end{bmatrix} \\ l_2 &= \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix} \end{aligned}\right\} x_{12} = l_1 \times l_2$$

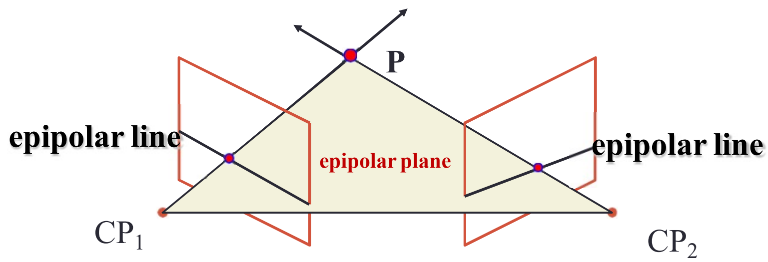
Motivating the problem: stereo

- Given two views of a scene (the two cameras not necessarily having optical axes) what is the relationship between the location of a scene point in one image and its location in the other?



Stereo correspondence

- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point

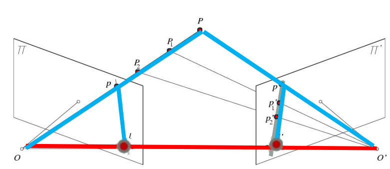


Epipolar Constraint

- Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Epipolar geometry: terms

- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with image plane
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane



- All epipolar lines intersect at the epipole
- An epipolar plane intersects the left and right image planes in epipolar lines

Example: converging cameras

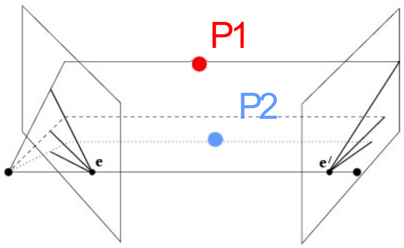
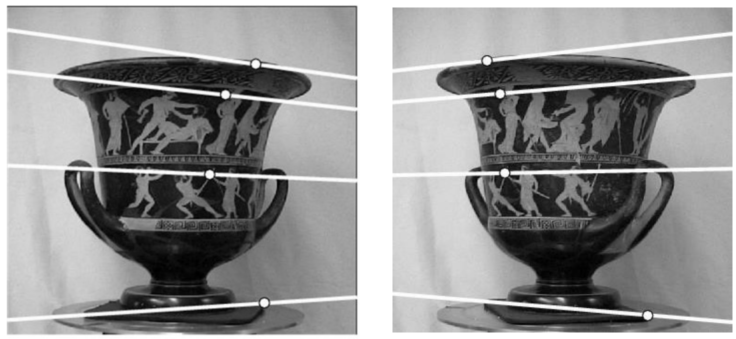


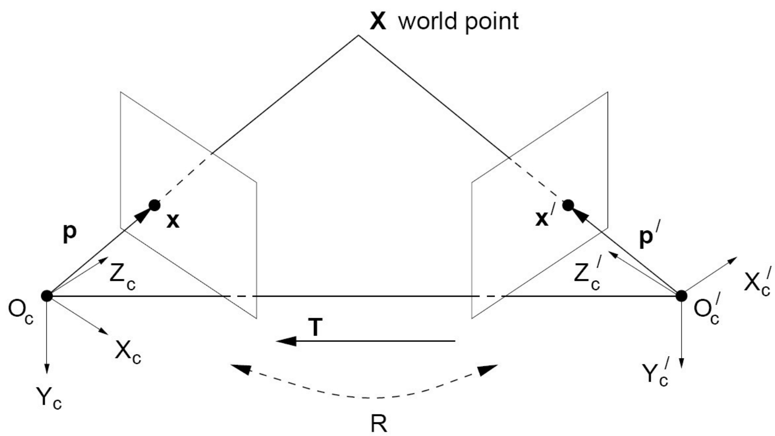
Figure from Hartley & Zisserman



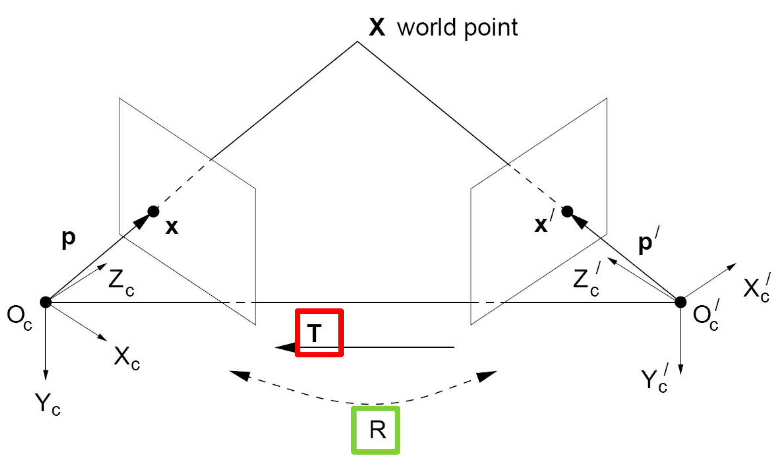
From Geometry to Algebra

- So far, we have the explanation in terms of geometry.
- Now, how do we express the epipolar constraints algebraically?

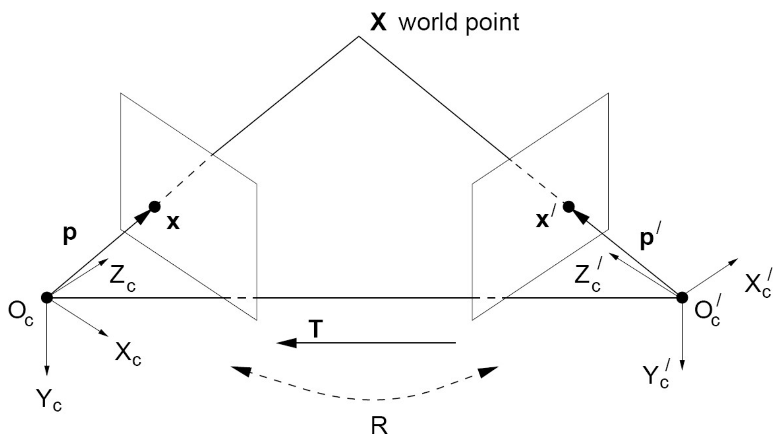
Stereo geometry, with calibrated cameras



Stereo geometry, with calibrated cameras



Stereo geometry, with calibrated cameras

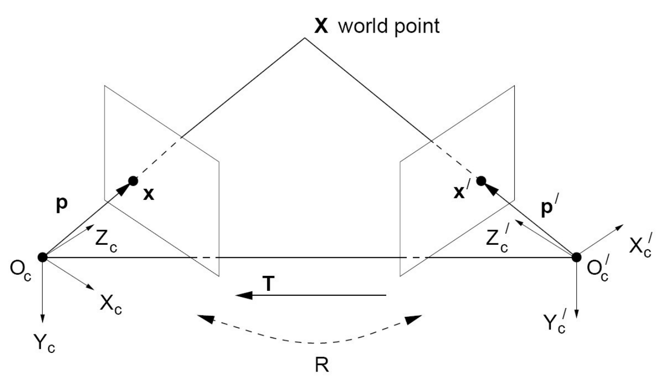


If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix **R**; translation: 3 vector **T**.

From geometry to algebra



$$\mathbf{X}'_c = \mathbf{R}\mathbf{X}_c + \mathbf{T}$$

Aside 1: Reminder of cross product

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

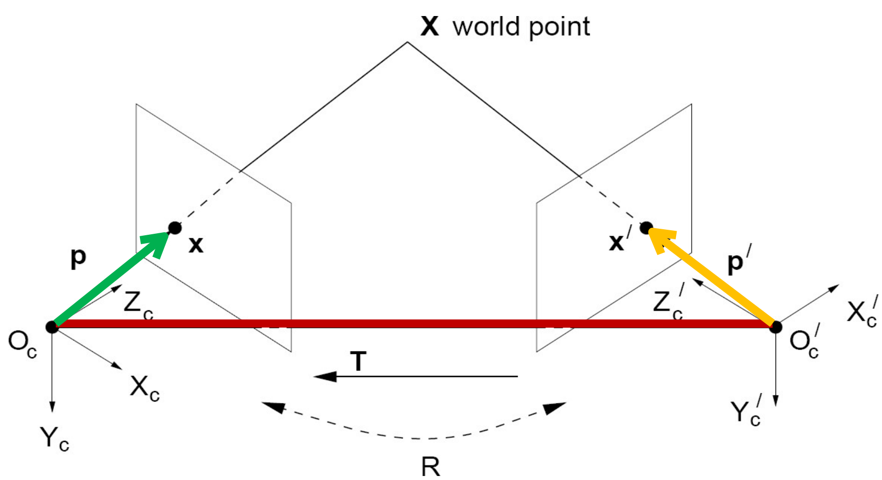
$$a \times b = c$$

Here c is perpendicular to both a and b, i.e. the dot product = 0.

$$a \cdot c = 0$$

$$b \cdot c = 0$$

Also A x A = 0 for all A



$$\boxed{\mathbf{X}' = \mathbf{R}}\,\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\,\mathbf{X} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\,\mathbf{X})$$

$$\boxed{0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\,\mathbf{X})}$$

Aside2: Matrix form of cross product

$$a \times b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = c$$

Can be expressed as a matrix multiplication!!!

Aside2: Matrix form of cross product

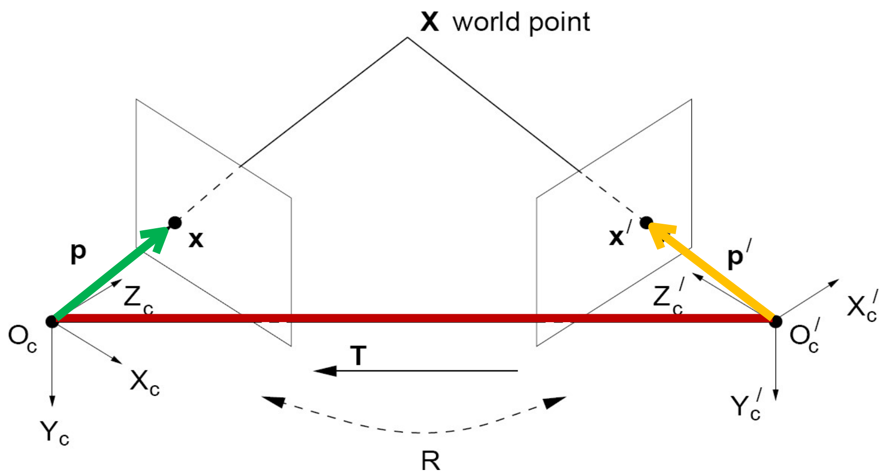
Can define a cross product matrix operation:

$$[a_x] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Notation:

$$a \times b = [a_\times] b$$

Has rank 2!



$$\boxed{\mathbf{X}' = \mathbf{R}}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\,\mathbf{X} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\,\mathbf{X})$$

$$\boxed{0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\,\mathbf{X})}$$

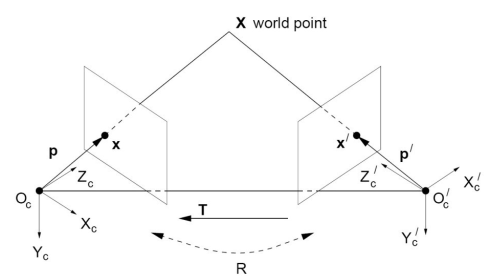
Essential matrix

$$\boxed{\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\,\mathbf{X}) = 0}$$

$$\mathbf{X}' \cdot ([\mathrm{T}_x]\mathbf{R}\,\mathbf{X}) = 0$$

Let $\mathbf{E} = [\mathrm{T}_x]\mathbf{R}$

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

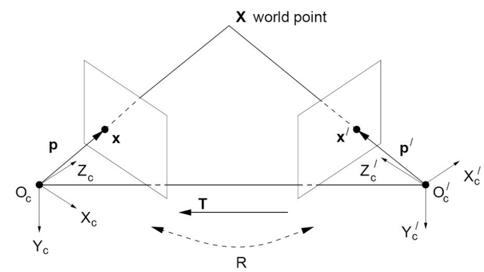


E is called the "essential matrix".

it relates the point X and the same point, but it described in the other camera frame

Essential matrix

$$\boxed{\mathbf{X}'^{T}\mathbf{E}\mathbf{X} = 0}$$



E relates corresponding image points between both cameras, given the rotation and translation.

Note: these points are in **each camera coordinate systems**.

We know if we observe a point in one image, its position in other image is constrained to lie on line defined by above.

Quiz

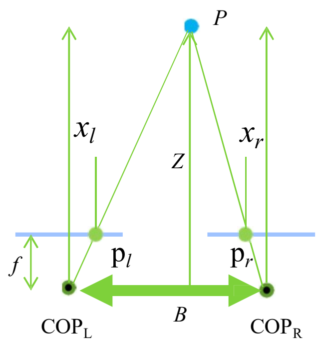
- That's fine for some converged cameras. But what if the image planes are parallel. What happens?

a) That is a degenerate case. You'll see in a bit.

b) That's fine. R is just the identity and the math works.

c) I have no idea.

Essential matrix example: parallel cameras

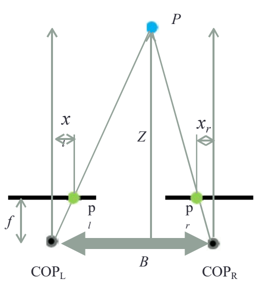


$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-B, 0, 0]^T$$

$$\mathbf{E} = [\mathrm{T}_x]\mathbf{R} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{matrix}$$

Essential matrix example: parallel cameras



$$\mathbf{E} = [\mathrm{T}_x]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{pmatrix}$$

$$\mathbf{p}'^t\mathbf{E}\mathbf{p} = 0 : \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\mathbf{p} = [X, Y, Z] = [\frac{Zx}{f}, \frac{Zy}{f}, Z] \cong [x, y, f]$$

$$\mathbf{p}' = [X', Y', Z] = [\frac{Zx'}{f}, \frac{Zy'}{f}, Z] \cong [x', y', f]$$

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ Bf \\ -By \end{bmatrix} = 0$$

$$Bfy' = Bfy \Rightarrow \boxed{\mathbf{y}' = \mathbf{y}}$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

Given a known point (x,y) in the original image, this is a ***line*** in the (x',y') image.

Fundamental matrix

Weak calibration

Main idea:

- Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras

From before: Projection matrix

$$\begin{bmatrix} w x_{im} \\ w y_{im} \\ w \end{bmatrix} = \mathbf{K}_{\text{int}} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

From before: Projection matrix

$$\begin{bmatrix} w x_{im} \\ w y_{im} \\ w \end{bmatrix} = \mathbf{K}_{\text{int}} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{\Phi}_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^{\mathrm{T}}\mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^{\mathrm{T}}\mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^{\mathrm{T}}\mathbf{T} \end{bmatrix}$$

From before: Projection matrix

$$\begin{bmatrix} w x_{im} \\ w y_{im} \\ w \end{bmatrix} = \mathbf{K}_{int} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{K}_{\text{int}} = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Note: Invertible, scale x and y, assumes no skew

From before: Projection matrix

$$\begin{bmatrix} w x_{im} \\ w y_{im} \\ w \end{bmatrix} = \mathbf{K}_{int} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{p}_{im} = \mathbf{K}_{int} \underbrace{\mathbf{\Phi}_{ext} \mathbf{P}_w}_{\mathbf{p}_c}$$

$$\boxed{\mathbf{p}_{im} = \mathbf{K}_{int} \mathbf{p}_c}$$

Uncalibrated case

For a given camera: $\mathbf{p}_{im} = \mathbf{K}_{int}\mathbf{p}_c$

And since invertible: $\mathbf{p}_c = \mathbf{K}_{int}^{-1}\mathbf{p}_{im}$

Uncalibrated case

So, for **two** cameras (left and right):

$$\mathbf{p}_{c,left} = \mathbf{K}_{int,left}^{-1}\,\mathbf{p}_{im,left}$$

$$\mathbf{p}_{c,right} = \mathbf{K}_{int,right}^{-1}\,\mathbf{p}_{im,right}$$

Internal calibration matrices, one per camera

Uncalibrated case

$$\mathbf{p}_{c,right} = \mathbf{K}^{-1}_{int,right}\,\mathbf{p}_{im,right} \qquad \mathbf{p}_{c,left} = \mathbf{K}^{-1}_{int,left}\,\mathbf{p}_{im,left}$$

From before, the ***essential*** *matrix* ***E.***

$$\boxed{\mathbf{p}_{c,right}^{\mathrm{T}}\,\mathbf{E}\,\mathbf{p}_{c,left} = 0}$$

$$\left(\mathbf{K}^{-1}_{int,right}\mathbf{p}_{im,right}\right)^{\mathrm{T}}\mathbf{E}\left(\mathbf{K}^{-1}_{int,left}\mathbf{p}_{im,left}\right) = 0$$

Uncalibrated case

$$\left(\mathbf{K}^{-1}_{int,right}\mathbf{p}_{im,right}\right)^{\mathrm{T}}\mathbf{E}\left(\mathbf{K}^{-1}_{int,left}\mathbf{p}_{im,left}\right) = 0$$

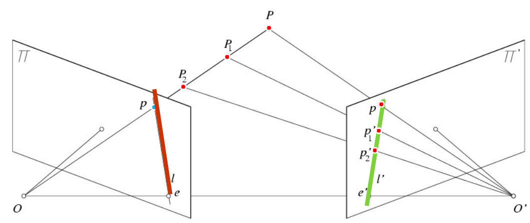
$$\mathbf{p}^{\mathrm{T}}_{im,right}\left(\underbrace{\left(\mathbf{K}^{-1}_{int,right}\right)^{T}\mathbf{E}\,\mathbf{K}^{-1}_{int,left}}_{\mathbf{F}}\right)\mathbf{p}_{im,left} = 0$$

"Fundamental matrix": $\mathbf{F}$

$$\boxed{\mathbf{p}^{\mathrm{T}}_{im,right}\,\mathbf{F}\,\mathbf{p}_{im,left} = 0} \quad or \quad \mathbf{p}^{T}\mathbf{F}\,\mathbf{p}' = 0$$

Properties of the Fundamental Matrix

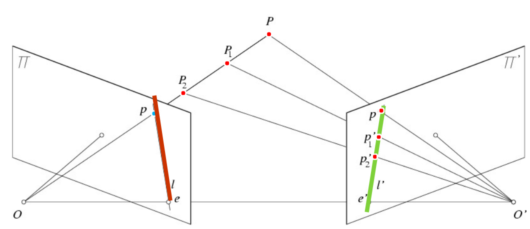
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



$\mathbf{l} = \mathbf{F}\,\mathbf{p}'$ is the epipolar ***line*** in the *p* image associated with p’

Properties of the Fundamental Matrix

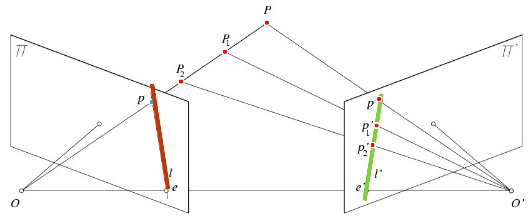
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



$\mathbf{l}' = \mathbf{F}^{T}\mathbf{p}$ is the epipolar line in the prime image associated with p

Properties of the Fundamental Matrix

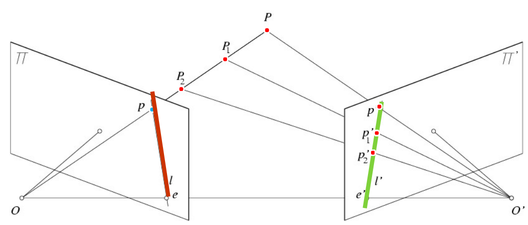
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



Epipoles found by $Fp' = 0$ and $F^{T}p = 0$

Properties of the Fundamental Matrix

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



F is singular (mapping from homogeneoues 2-D point to 1-D family so rank 2 – more later)

Fundamental matrix

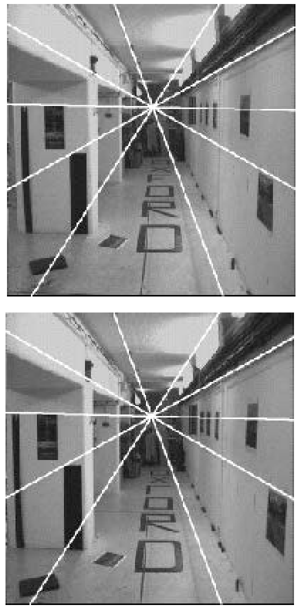
- Relates pixel coordinates in the two views
- More general form than essential matrix:

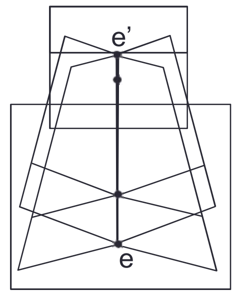
  We remove the need to know intrinsic parameters

Fundamental matrix

- If we estimate fundamental matrix from correspondences in pixel coordinates, can reconstruct epipolar geometry without intrinsic or extrinsic parameters.

Different Example: forward motion





courtesy of Andrew Zisserman

Computing F from correspondences

$$\mathbf{p}_{im,right}^{T}\,\mathbf{F}\,\mathbf{p}_{im,left} = 0$$

Each point correspondence generates ***one*** constraint on F

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

Computing F from correspondences

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

Multiply out:

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Computing F from correspondences

Collect N of these:

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_n u_n & u'_n v_n & u'_1 & v'_n u_n & v'_n v_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

And solve for ***f*** the elements of F***…***

The (in)famous “eight-point algorithm”

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1.00
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1.00
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1.00
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1.00
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1.00
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1.00
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1.00
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1.00

$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

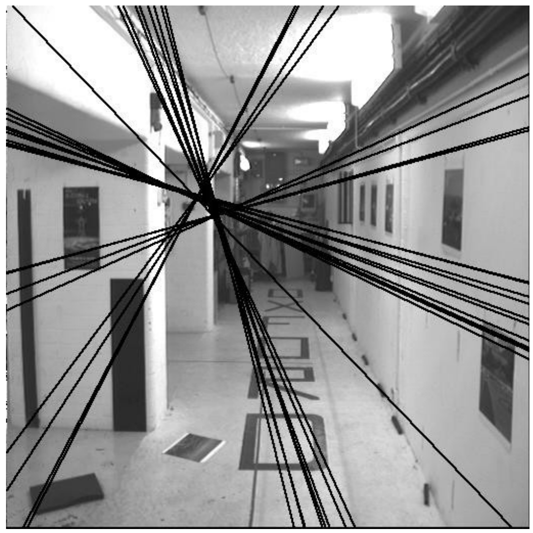
The (in)famous “eight-point algorithm”

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1.00
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1.00
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1.00
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1.00
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1.00
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1.00
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1.00
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1.00

$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

- In principal can solve with 8 points.
- What happens when there is noise?
- Better with more – yields homogeneous linear least-squares:
  - Find unit norm vector F yielding smallest residual
  - Remember SVD?

Just solving for F...

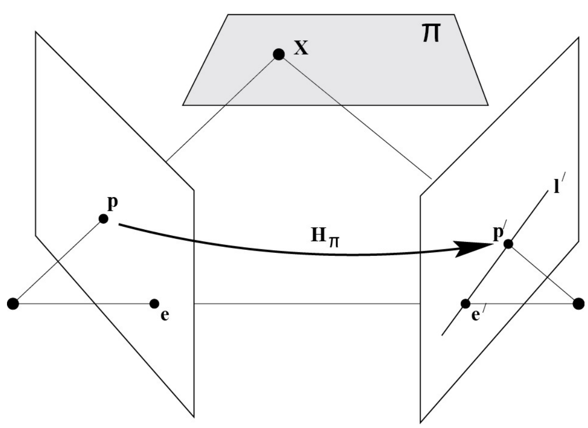


Rank of F

- Assume we know the homography H_π that maps from Left to Right (Full 3x3)

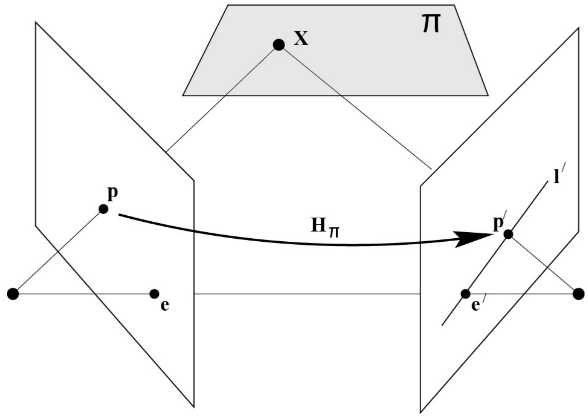
$$\mathbf{p}' = \mathbf{H}_\pi \mathbf{p}$$

- Let line $\mathbf{l}'$ be the epiloar line corresponding to $\mathbf{p}$ – goes through epipole $\mathbf{e}'$



Rank of F

- Let line **l'** be the epiloar line corresponding to **p** – goes through epipole **e'**



$$\mathbf{l}' = \mathbf{e}' \times \mathbf{p}'$$
$$= \mathbf{e}' \times \mathbf{H}_\pi \mathbf{p}$$
$$= \boxed{[\mathbf{e}']_\times \mathbf{H}_\pi} \mathbf{p}$$

But **l'** is the epipolar line for p: $\mathbf{l}' = \boxed{\mathbf{F}}\,\mathbf{p}$

Rank of F is rank of $[e']_x$=2

Fix the linear solution

- 1- Use SVD or other method to do linear computation for F
- 2- Decompose the estimated F using SVD (not the same SVD):

$$\mathbf{F} = UDV^T$$

Fix the linear solution

- Use SVD or other method to do linear computation for F
- Decompose F using SVD (not the same SVD):

$$\mathbf{F} = U D V^T$$

- Set the last singular value to zero:

$$D = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & t \end{bmatrix} \Rightarrow \hat{D} = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

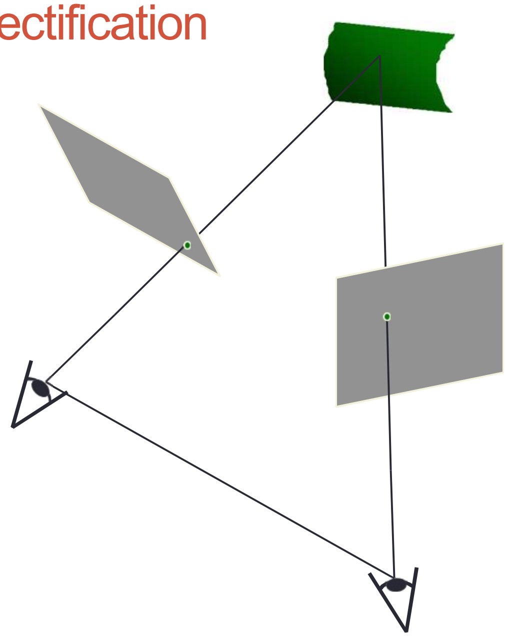
Fix the linear solution

- Estimate new F from the new $\hat{D}$

$$\hat{\mathbf{F}} = U \hat{D} V^T$$

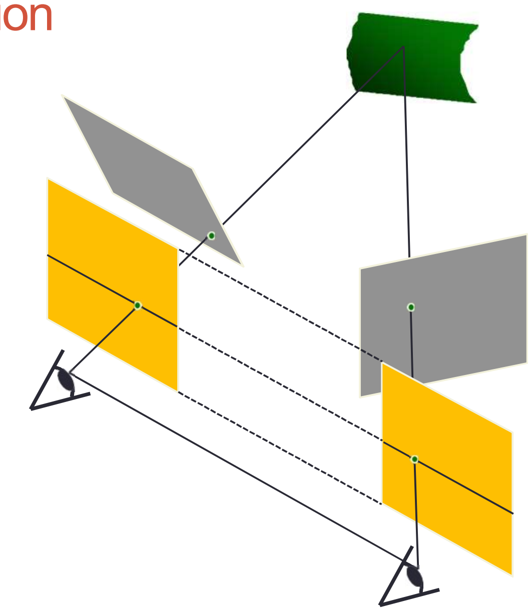
That's better…

Stereo image rectification



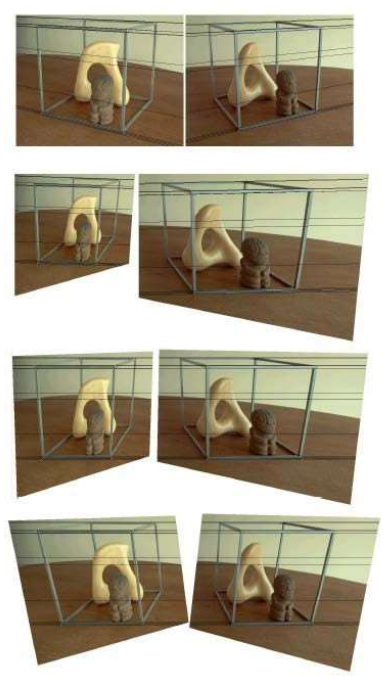
Stereo image rectification

- Reproject image planes onto a common plane parallel to theline between optical centers – each a homography
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection
  - C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



Rectification Example

C. Loop and Z. Zhang, Computing Rectifying Homographies for Stereo Vision, IEEE Conf. Computer Vision and Pattern Recognition, 1999.



(a) Original image pair overlayed with several epipolar lines.

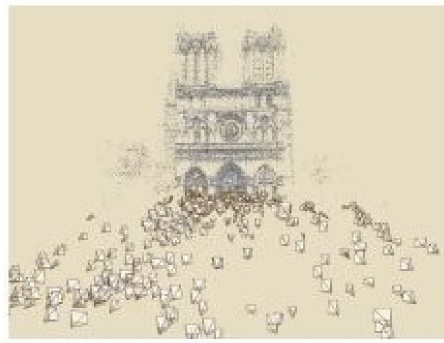
(b) Image pair transformed by the specialized projective mapping H_p and H'_p. Note that the epipolar lines are now parallel to each other in each image.

(c) Image pair transformed by the similarity H_r and H'_r. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

(d) Final image rectification after shearing transform H_s and H'_s. Note that the image pair remains rectified, but the horizontal distortion is reduced.

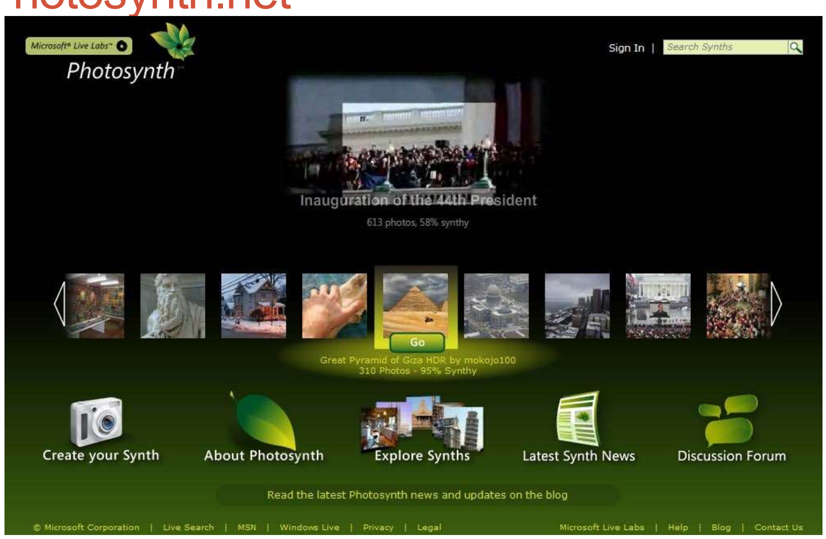
Photo synth

Noah Snavely, Steven M. Seitz, Richard Szeliski, "Photo tourism: Exploring photo collections in 3D," SIGGRAPH 2006



http://photosynth.net/

Photosynth.net



Based on Photo Tourism
by Noah Snavely, Steve Seitz, and Rick Szeliski

3D from multiple images

Building Rome in a Day: Agarwal et al. 2009

Summary

- For 2-views, there is a geometric relationship that define the relations between rays in one view to rays in the other – epipolar geometry.
- These relationships can be captured algebraicly as well:
  - Calibrated – Essential matrix
  - Uncalibrated – Fundamental matrix.
- This relation can be estimated from point correspondences.