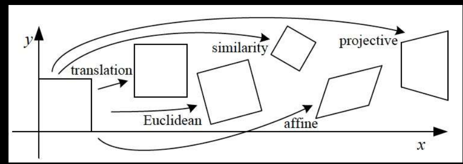
Computer Vision

Essential matrix

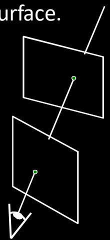
Last time

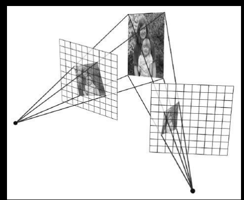
- Projective Transforms: Matrices that provide transformations including translations, rotations, similarity, affine and finally general (or perspective) projection.
- When 2D matrices are 3x3; for 3D they are 4x4.



Last time: Homographies

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

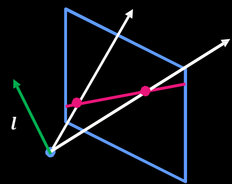




Projective lines

In Vector Notation:

$$0 = \underset{l}{[a \quad b \quad c]}\underset{p}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}$$



A line is also represented as a homogeneous 3-vector!

Projective Geometry: Lines and points

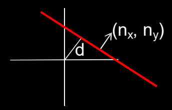
2D Lines: $ax + by + c = 0$

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

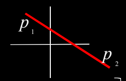
$$l = \begin{bmatrix} a & b & c \end{bmatrix} \Rightarrow \begin{bmatrix} n_x & n_y & -d \end{bmatrix}$$

Eq of line

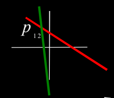
$$\mathbf{l}^T \mathbf{x} = 0$$



Projective Geometry: Lines and points



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

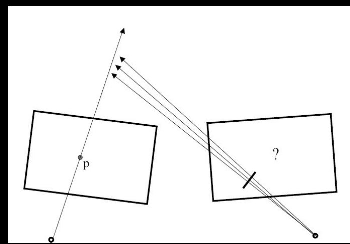


$$\left.\begin{array}{l} l_1 = \begin{bmatrix} a_1 & b_1 & c_1 \end{bmatrix} \\ l_2 = \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix} \end{array}\right\} p_{12} = l_1 \times l_2$$

Motivating the problem: Stereo

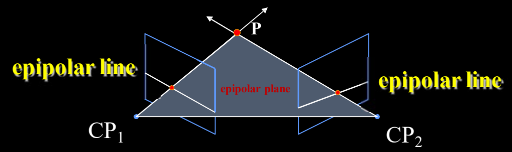
- Given two views of a scene (the two cameras not necessarily having optical axes) what is the relationship between the location of a scene point in one image and its location in the other?

Motivating the problem: Stereo



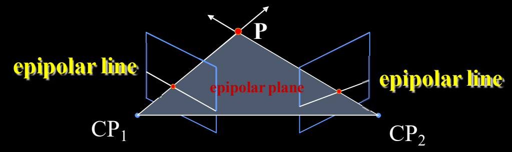
Stereo correspondence

- Find pairs of points that correspond to same scene point



Stereo correspondence

Epipolar Constraint reduces correspondence problem to 1D search along conjugate epipolar lines



Example: Converging cameras

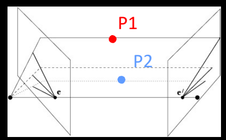
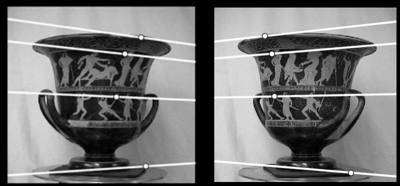
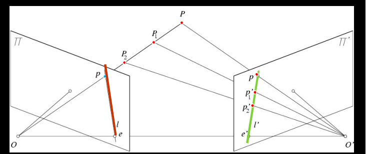


Figure from Hartley & Zisserman



Epipolar geometry: Terms

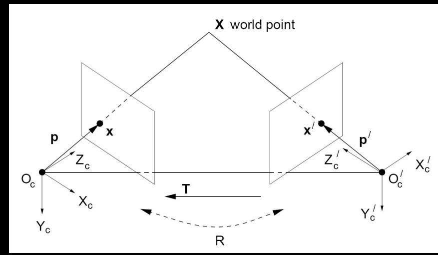
- ***Baseline***: line joining the camera centers
- *Epipolar plane*: plane containing baseline and world point
- *Epipolar line*: intersection of epipolar plane with the image plane – come in pairs
- *Epipole*: point of intersection of baseline with image plane



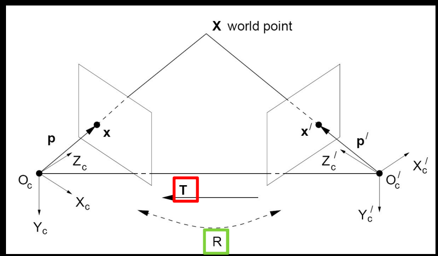
From Geometry to Algebra

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

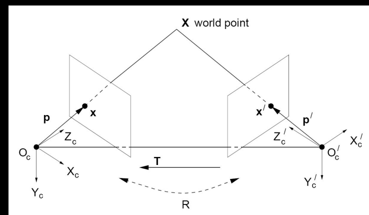
Stereo geometry, with calibrated cameras



Stereo geometry, with calibrated cameras



From geometry to algebra



$$X'_c = R X_c + T$$

Aside 1: Reminder of cross product

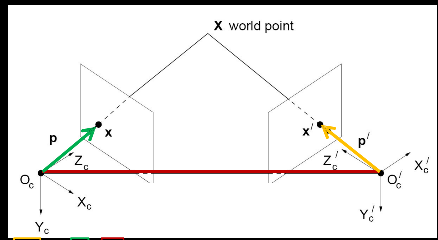
Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

$$\vec{a} \times \vec{b} = \vec{c}$$

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

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

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



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

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

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

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

Aside 2: Matrix form of cross product

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

Can be expressed as a matrix multiplication!!!

Aside 2: Matrix form of cross product

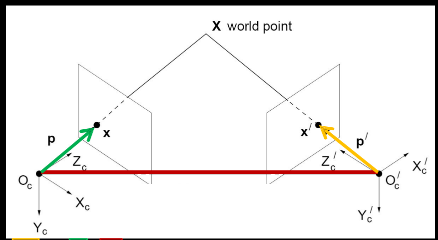
Can define a cross product matrix operation:

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

Notation:

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

Has rank 2!



$$\boxed{X'} = \boxed{RX} + \boxed{T}$$

$$\underbrace{T \times X'}_{\text{Normal to the plane}} = T \times RX$$

$$X' \cdot (T \times X') = X' \cdot (T \times RX)$$

$$\boxed{0 = X' \cdot (T \times RX)}$$

Essential matrix

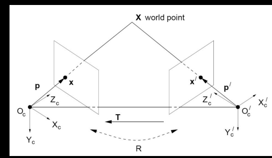
$$\boxed{X' \cdot (T \times RX) = 0}$$

$$X' \cdot ([T_\times] RX) = 0$$

$$\text{Let} \quad E = [T_\times] R$$

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

E is called the "essential matrix".



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.

Quiz – answer

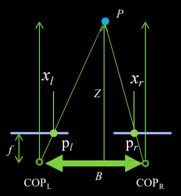
- That's fine for some converged cameras. But what if the image planes are parallel. What happens?

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

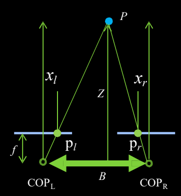


$$\mathbf{R} = \mathbf{I}$$

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

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

Essential matrix example: parallel cameras

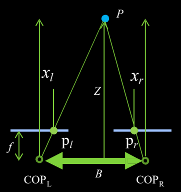


$$\mathbf{p}'^T \mathbf{E}\,\mathbf{p} = 0 \qquad \mathbf{p} = [X, Y, Z] = [\frac{Zx}{f}, \frac{Zy}{f}, Z]$$

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

$$[x' \quad y' \quad f]\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

Essential matrix example: parallel cameras



$$[x' \quad y' \quad f]\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

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

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

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

Computer Vision

Fundamental matrix

Weak calibration

Main idea:

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

From before: Projection matrix

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

From before: Projection matrix

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

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

From before: Projection matrix

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

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

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):

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

$$\mathbf{p}_{c,right} = \underbrace{\mathbf{K}_{int,right}^{-1}}_{\text{Internal calibration matrices, one per camera}} \mathbf{p}_{im,right}$$

Uncalibrated case

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

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

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

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

Uncalibrated case

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

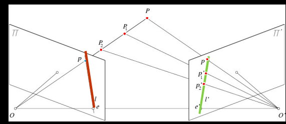
$$\mathbf{p}_{im,right}^{T}\,\underbrace{(\mathbf{K}_{int,right}^{-1})^{T}\,\mathbf{E}\,\mathbf{K}_{int,left}^{-1}}\,\mathbf{p}_{im,left} = 0$$

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

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

Properties of the Fundamental Matrix

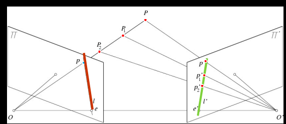
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



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

Properties of the Fundamental Matrix

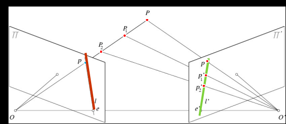
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



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

Properties of the Fundamental Matrix

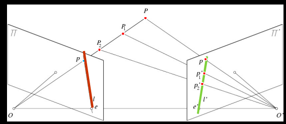
$$\mathbf{p}^{T}\,\mathbf{F}\,\mathbf{p}' = 0$$



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)

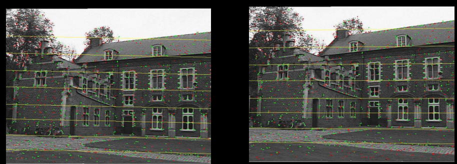
Fundamental matrix

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

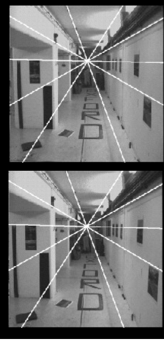
  We remove the need to know intrinsic parameters

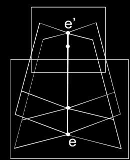
Fundamental matrix

- 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

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

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:

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Computing F from correspondences

Collect N of these:

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_nu_n & u'_nv_n & u'_n & v'_nu_n & v'_nv_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

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

The (in)famous "eight-point algorithm"

$$\begin{bmatrix} 250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 & 1.00 \\ 2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 & 1.00 \\ 416374.23 & 871684.30 & 935.47 & 408110.89 & 854384.92 & 916.90 & 445.10 & 931.81 & 1.00 \\ 191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 & 1.00 \\ 48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 & 1.00 \\ 164786.04 & 546559.67 & 813.17 & 1998.37 & 6628.15 & 9.86 & 202.65 & 672.14 & 1.00 \\ 116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 & 1.00 \\ 135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48 & 1.00 \end{bmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

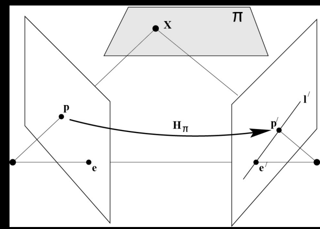
Just solving for F…

Rank of F

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

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

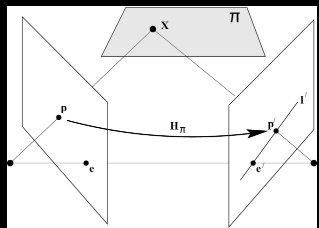
- Let line **l'** be the epiloar line corresponding to **p** – goes through epipole **e'**



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

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

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

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:

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

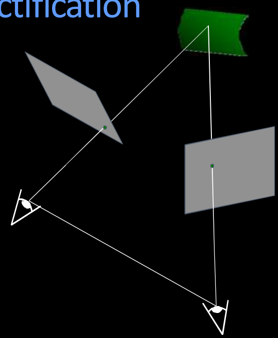
Fix the linear solution

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

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

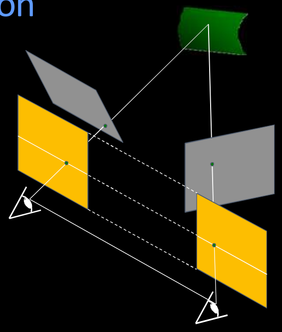
That's better…

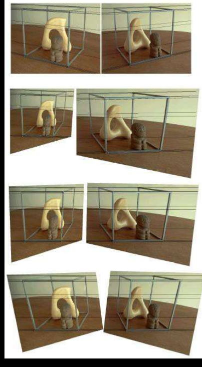
Stereo image rectification



Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between optical centers –each a homography
- Pixel motion is horizontal after this transformation
- 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.

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

Algorithm Steps

Step 1 — Detect feature points

1. For each image i:
   - Detect the 2D coordinates of feature points (e.g. chessboard corners) → (u_{ij}, v_{ij})
   - Get the corresponding 3D world coordinates $(X_j, Y_j, 0)$ on the pattern plane.

Step 2 — Compute homography

2. For each view i:
   - Compute the **homography** H_i such that:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H_i \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

   - Use Direct Linear Transform (DLT) to solve for H_i.

Step 3 — Estimate intrinsic matrix K

3. For each H_i, form the constraint:

$$H_i = K[r_{1i}\ r_{2i}\ t_i]$$

   - From this, derive two linear equations involving K's parameters.
   - Stack all equations from all views → solve for K using linear least squares.

Step 4 — Estimate extrinsic parameters

4. For each view i:
   - Compute:

$$r_1 = \lambda K^{-1} h_1, \quad r_2 = \lambda K^{-1} h_2, \quad t = \lambda K^{-1} h_3$$

   where h_j are columns of H_i, and $\lambda = 1/\|K^{-1} h_1\|$
   - Form $r_3 = r_1 \times r_2$ to complete the rotation matrix R_i.

3.1 Homography relationship

For each calibration image i, the homography between the **pattern plane** ($Z = 0$) and the image plane is:

$$H_i = K[r_{1i}\ r_{2i}\ t_i]$$

Let h_{i1}, h_{i2}, h_{i3} denote the **three columns** of H_i.

Then:

$$h_{i1} = K\,r_{1i}, \quad h_{i2} = K\,r_{2i}, \quad h_{i3} = K\,t_i$$

The rotation vectors r_{1i}, r_{2i} are **orthonormal**, so they satisfy:

$$r_{1i}^T r_{2i} = 0, \quad r_{1i}^T r_{1i} = r_{2i}^T r_{2i}$$

Substitute $r_{1i} = K^{-1}h_{i1}, r_{2i} = K^{-1}h_{i2}$:

$$h_{i1}^T K^{-T} K^{-1} h_{i2} = 0, \quad h_{i1}^T K^{-T} K^{-1} h_{i1} = h_{i2}^T K^{-T} K^{-1} h_{i2}$$

Let

$$B = K^{-T} K^{-1}$$

which is a **symmetric 3×3 matrix** containing 6 unknowns.

3.2 Linear constraints

Define a 6-vector $b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$.

For each homography H_i, define:

$$v_{ij} = [h_{i1}h_{j1},\ h_{i1}h_{j2} + h_{i2}h_{j1},\ h_{i2}h_{j2},\ h_{i3}h_{j1} + h_{i1}h_{j3},\ h_{i3}h_{j2} + h_{i2}h_{j3},\ h_{i3}h_{j3}]^T$$

Then the orthogonality constraints become:

$$v_{12}^T b = 0$$

$$(v_{11} - v_{22})^T b = 0$$

Each view gives two linear equations in the six unknowns of b.

3.3 Solve for B and K

- Stack all constraints from n images into one matrix V:

$$Vb = 0$$

- Solve using **SVD** — b is the right singular vector corresponding to the smallest singular value.

From b, reconstruct B:

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

Then compute K analytically (Zhang's closed-form solution):

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2}$$

$$\lambda = B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}}$$

$$\alpha = \sqrt{\frac{\lambda}{B_{11}}}, \quad \beta = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda, \quad u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda$$

Finally:

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

This K is the **intrinsic camera matrix**.

4.1 Recover rotation and translation

From

$$H_i = K[r_{1i}\ r_{2i}\ t_i]$$

Compute:

$$K^{-1}H_i = [\tilde{r}_{1i}\ \tilde{r}_{2i}\ \tilde{t}_i]$$

Let $\lambda_i = 1/\|\tilde{r}_{1i}\|$
(ensures unit length for rotation columns)

Then:

$$r_{1i} = \lambda_i \tilde{r}_{1i}, \quad r_{2i} = \lambda_i \tilde{r}_{2i}, \quad r_{3i} = r_{1i} \times r_{2i}, \quad t_i = \lambda_i \tilde{t}_i$$

This gives the full **rotation matrix** $R_i = [r_{1i}, r_{2i}, r_{3i}]$
and **translation vector** t_i.

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.