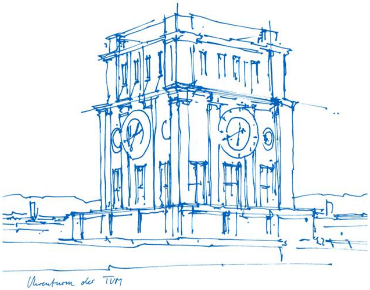
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 06 2D-2D Geometry
(Part 1 Overview and Fundamentals)

Dr. Haoang Li

01 June 2023 11:00-11:45



Outline

- Overview of 2D-2D Geometry
- Two-view Geometric Constraints
- Eight-point Method

Overview of Two-view Geometry

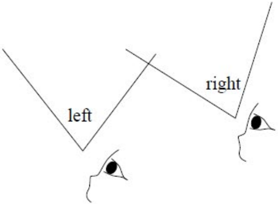
➢ Intuitive Illustration

✓ Camera pose estimation

We can easily imagine that the image A is obtained by the left camera (eye) and image B is obtained by the right camera (eye).

Image A

Image B



Camera motion can be inferred from two consecutive image frames.

Overview of Two-view Geometry

➢ Intuitive Illustration

✓ Camera pose estimation

We infer the camera motion from some object **correspondences**. Objects can be further abstracted by points and lines.

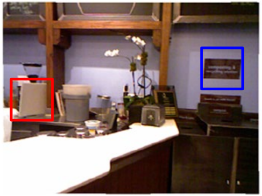


Image A

Image B

Object correspondences

Image A

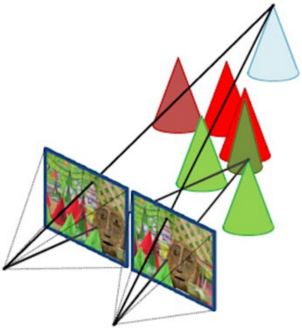
Image B

Point correspondences

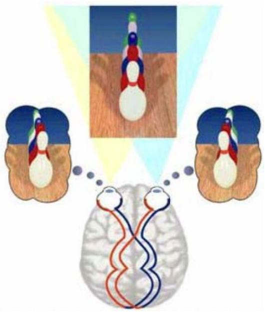
Overview of Two-view Geometry

- ➢ Intuitive Illustration

✓ 3D reconstruction



3D perception from two human eyes



Two human eyes

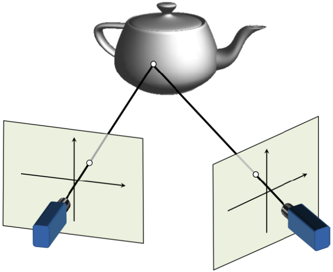
Brain

Overview of Two-view Geometry

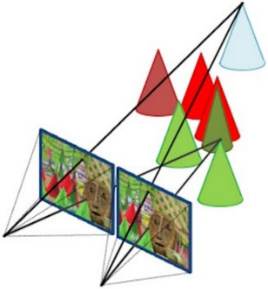
➢ Intuitive Illustration

✓ 3D reconstruction

Given a pair of 2D points in two images, the 3D point's position in space is found as the intersection of the two projection rays.



Triangulation

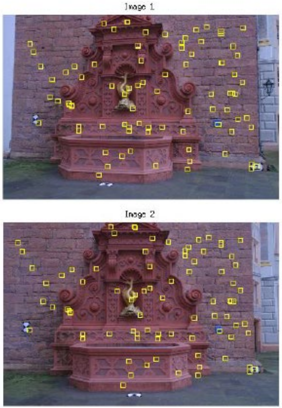


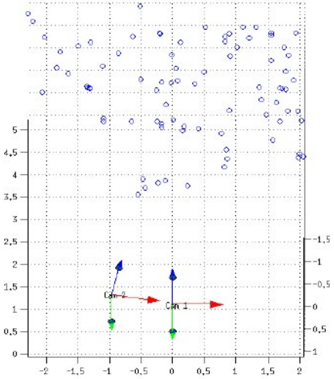
Depth from disparity

Overview of Two-view Geometry

➢ Problem Formulation

2D-2D point correspondence





Estimated poses and 3D structure

Overview of Two-view Geometry

➢ Problem Formulation

✓ Can we solve the estimation of relative motion (R,T) without any prior information about 3D points? Yes! The next couple of slides prove that this is possible.

✓ Once (R,T) are known, the 3D points can be triangulated using the triangulation algorithm (i.e., least square approximation plus reprojection error minimization)

$P_i = ?$

$K_1, R_1, T_1 = ?$ $K_2, R_2, T_2 = ?$

=

First step

$K_1, R_1, T_1 = ?$ $K_2, R_2, T_2 = ?$

+

Second step $P_i = ?$

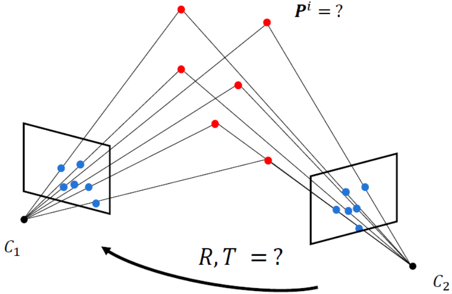
K_1, R_1, T_1 K_2, R_2, T_2

Overview of Two-view Geometry

➢ Problem Formulation

✓ Recover simultaneously 3D scene structure and camera poses (up to scale) from two images. (More specifically, camera pose first, followed by 3D structure.)

✓ Intrinsic parameters of camera is known from calibration. We can also handle uncalibrated case.



Overview of Two-view Geometry

➢ Problem Formulation

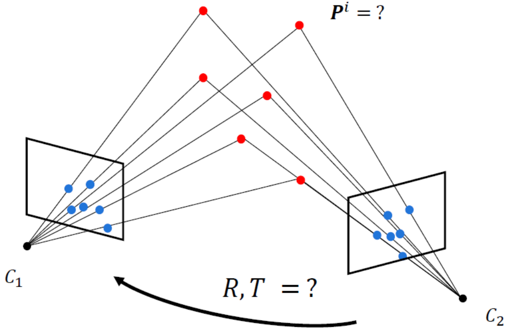
✓ Given a set of n point correspondences $\{p^i{}_1 = (u^i{}_1, v^i{}_1),\ p^i{}_2 = (u^i{}_2, v^i{}_2)\}$ between two images, where $i = 1 \dots n$, the goal is to simultaneously

- estimate the 3D points $\boldsymbol{P}^i$ and
- the camera relative-motion parameters $(\boldsymbol{R}, \boldsymbol{T})$

$$\lambda_1^i \begin{bmatrix} u^i{}_1 \\ v^i{}_1 \\ 1 \end{bmatrix} = K_1 [I|0] \cdot \begin{bmatrix} X^i{}_w \\ Y^i{}_w \\ Z^i{}_w \\ 1 \end{bmatrix}$$

$$\lambda_2^i \begin{bmatrix} u^i{}_2 \\ v^i{}_2 \\ 1 \end{bmatrix} = K_2 [R|T] \cdot \begin{bmatrix} X^i{}_w \\ Y^i{}_w \\ Z^i{}_w \\ 1 \end{bmatrix}$$

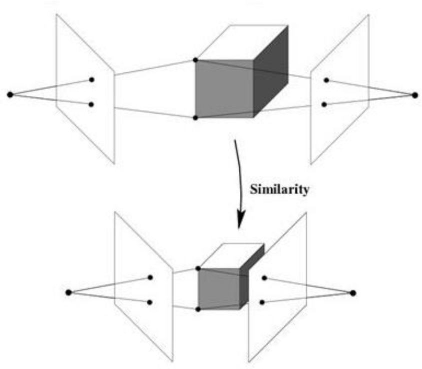
Perspective projection



Overview of Two-view Geometry

➢ Scale Ambiguity

If we rescale the entire scene and camera views, the projections (in pixels) of the scene points in both images remain exactly the same:



- Reduce the size of 3D object: smaller projection
- Reduce the distance from camera to 3D object: bigger projection
- **Simultaneously** reduce the size of 3D object and reduce the distance from camera to 3D object?

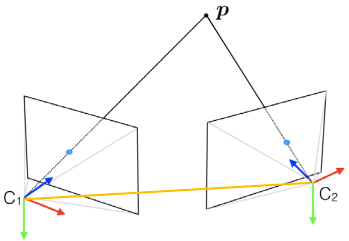
Overview of Two-view Geometry

➢ Scale Ambiguity

✓ For monocular case, it is not possible to recover the absolute scale of the scene.

✓ Thus, only 5 degrees of freedom are measurable:

- Three parameters to describe the rotation
- Two parameters for the parameters for the translation up to a scale (we can only compute the direction of translation but not its length)



Overview of Two-view Geometry

➢ Number of Point Correspondences

✓ $\boldsymbol{4n}$ knowns:

- n correspondences; each one $(u^i{}_1, v^i{}_1)$ and $(u^i{}_2, v^i{}_2)$, i=1…n

✓ $\boldsymbol{5+3n}$ unknowns

- 5 for the motion up to a scale (3 for rotation, 2 for translation)
- $3n$ is number of coordinates of n 3D points (*x, y, z*)

✓ If and only if the number of independent equations ≥ number of unknowns

$$4n \geq 5 + 3n \quad \Rightarrow \quad \boxed{n \geq 5}$$

Overview of Two-view Geometry

➢ Number of Point Correspondences

✓ In 1913, Kruppa showed that 5 image correspondences is the minimal case and that there can be at up to 11 solutions [1].

✓ In 1981, the first popular solution uses 8 points and is called the 8 point algorithm or Longuet Higgins algorithm [2].

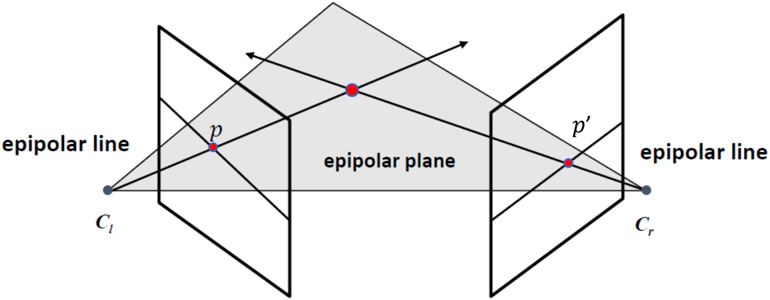
✓ In 2004 , Nister proposed the first efficient and non-iterative solution . It uses Groebner basis decomposition [3].

[1] E. Kruppa, Zur Ermittlung eines Objektes aus zwei Perspektiven mit Innerer Orientierung, Sitz. Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa. IIa., 1913
[2] H. Christopher Longuet Higgins, A computer algorithm for reconstructing a scene from two projections, Nature, 1981.
[3] D. Nister , An Efficient Solution to the Five Point Relative Pose Problem, PAMI, 2004.

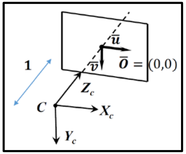
Geometric Constraints

➢ Epipolar planes and lines

- ✓ The camera centers C_l and C_r and the image point p and p' determine the so-called epipolar plane.
- ✓ The intersections of the epipolar plane with the two image planes are called epipolar lines.

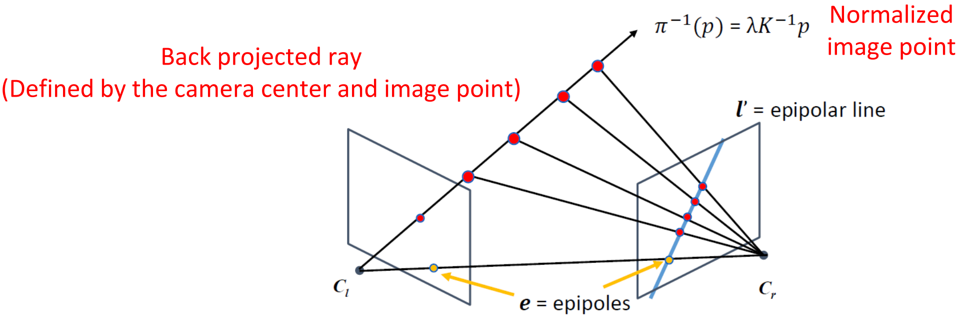


Geometric Constraints



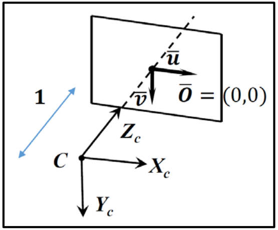
➢ Epipolar planes and lines

- ✓ The epipolar line is the projection of a back projected ray $\pi^{-1}(p)$ onto the other camera image
- ✓ The epipole is the projection of the optical center on the other camera image
- ✓ A pair of images has two epipoles.



Geometric Constraints

➢ Essential Matrix

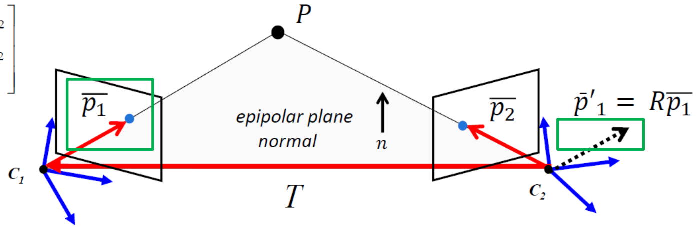


Coplanarity constraint

R and T denote the rotation and translation from C_1 to C_2. (arrow of T doesn't represent the direction of T, but "the coordinates of the center of C_1" in C_2.)

$$\bar{p}_1 = \begin{bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{bmatrix} \qquad \bar{p}_2 = \begin{bmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{bmatrix}$$

Normalized image coordinates



$\bar{p}_1, \bar{p}_2, T$ **are coplanar**

Right camera frame

Left camera frame

$$\underbrace{\bar{p}_2^T \cdot n}_{\text{orthogonality}} = 0 \Rightarrow \bar{p}_2^T \cdot \underbrace{(T \times \bar{p}'_1))}_{\text{Normal of epipolar plane}} = 0 \Rightarrow \bar{p}_2^T (T \times (R\bar{p}_1)) = 0$$

orthogonality

Normal of epipolar plane

From dot product to matrix multiplication

Geometric Constraints

- Essential Matrix

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

Coplanarity constraint

Skew-symmetric matrix

$$\bar{p}_2^T (T \times (R\bar{p}_1)) = 0 \quad \Rightarrow \bar{p}_2^T \; [T_\times] R \; \bar{p}_1 \; = 0 \quad \Rightarrow \boxed{\bar{p}_2^T \; E \; \bar{p}_1 \; = 0}$$

Associative law

Definition of essential matrix

$$\boxed{\mathrm{E} = \; [T_\times] R \quad \textit{essential matrix}}$$

R and T can be computed from E

Geometric Constraints

➢ From Essential Matrix to Fundamental Matrix

So far, we have assumed that the camera intrinsic parameters are **known** and we have used **normalized** image coordinates to get the epipolar constraint for calibrated cameras:

$$\bar{p}_2^T \, E \, \bar{p}_1 = 0 \qquad \begin{bmatrix} \bar{u}_2^i \\ \bar{v}_2^i \\ 1 \end{bmatrix}^T E \begin{bmatrix} \bar{u}_1^i \\ \bar{v}_1^i \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} \bar{u}_1^i \\ \bar{v}_1^i \\ 1 \end{bmatrix} = K_1^{-1} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} \qquad \begin{bmatrix} \bar{u}_2^i \\ \bar{v}_2^i \\ 1 \end{bmatrix} = K_2^{-1} \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}$$

Normalized image coordinates

$$\Rightarrow \quad \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T K_2^{-T} \, E \, K_1^{-1} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0$$

Geometric Constraints

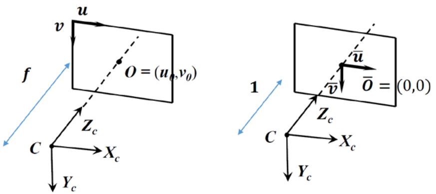
- Fundamental Matrix

✓ Definition of fundamental matrix

$$\begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T \boxed{K_2^{-T}\ E\ K_1^{-1}} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0 \quad \Rightarrow \quad \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T \boxed{F} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0$$

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

Advantage: Based on fundamental matrix, we work directly in ordinary image plane, instead of normalized image plane.



Geometric Constraints

➢ Computation of Fundamental/Essential Matrix

✓ Eight-point method (Direct linear transform--DLT)

- Essential matrix
- Fundamental matrix

✓ Five-point method (introduce in the next class)

- Essential matrix

Eight-point Method

- Classical Version

✓ We first take essential matrix estimation for example [1].

✓ Each pair of point correspondences $\bar{\boldsymbol{p}}_1 = (\bar{u}_1, \bar{v}_1, 1)^T, \quad \bar{\boldsymbol{p}}_2 = (\bar{u}_2, \bar{v}_2, 1)^T$ provides a linear equation:

$$\bar{p}_2^T \, E \, \bar{p}_1 = 0$$

Normalized image coordinates

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

[1] H. Christopher Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature, 1981

Eight-point Method

➢ Classical Version

$$\bar{u}_2\bar{u}_1 e_{11} + \bar{u}_2\bar{v}_1 e_{12} + \bar{u}_2 e_{13} + \bar{v}_2\bar{u}_1 e_{21} + \bar{v}_2\bar{v}_1 e_{22} + \bar{v}_2 e_{23} + \bar{u}_1 e_{31} + \bar{v}_1 e_{32} + e_{33} = 0$$

Note:

- ✓ The 8-point algorithm assumes that the entries of E are all independent. This is not true since, for the calibrated case, they depend on 5 parameters (R and T).
- ✓ The 5-point algorithm (introduced later) uses the epipolar constraint considering the dependencies among all entries.

Eight-point Method

➤ Classical Version

For n points, we can write

Normalized image coordinates

$$\underbrace{\begin{bmatrix} \bar{u}_2^{\,1}\bar{u}_1^{\,1} & \bar{u}_2^{\,1}\bar{v}_1^{\,1} & \bar{u}_2^{\,1} & \bar{v}_2^{\,1}\bar{u}_1^{\,1} & \bar{v}_2^{\,1}\bar{v}_1^{\,1} & \bar{v}_2^{\,1} & \bar{u}_1^{\,1} & \bar{v}_1^{\,1} & 1 \\ \bar{u}_2^{\,2}\bar{u}_1^{\,2} & \bar{u}_2^{\,2}\bar{v}_1^{\,2} & \bar{u}_2^{\,2} & \bar{v}_2^{\,2}\bar{u}_1^{\,2} & \bar{v}_2^{\,2}\bar{v}_1^{\,2} & \bar{v}_2^{\,2} & \bar{u}_1^{\,2} & \bar{v}_1^{\,2} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{u}_2^{\,n}\bar{u}_1^{\,n} & \bar{u}_2^{\,n}\bar{v}_1^{\,n} & \bar{u}_2^{\,n} & \bar{v}_2^{\,n}\bar{u}_1^{\,n} & \bar{v}_2^{\,n}\bar{v}_1^{\,n} & \bar{v}_2^{\,n} & \bar{u}_1^{\,n} & \bar{v}_1^{\,n} & 1 \end{bmatrix}}_{\text{Q (this matrix is known)}} \underbrace{\begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{bmatrix}}_{\bar{E}} = 0 \quad \Rightarrow \quad \mathrm{Q} \cdot \bar{\mathrm{E}} = 0$$

Q (this matrix is **known**)

$\bar{E}$ (this matrix is **unknown**)

$$\underbrace{\begin{pmatrix} X_w^1 & Y_w^1 & Z_w^1 & 1 & 0 & 0 & 0 & 0 & -u_1X_w^1 & -u_1Y_w^1 & -u_1Z_w^1 & -u_1 \\ 0 & 0 & 0 & 0 & X_w^1 & Y_w^1 & Z_w^1 & 1 & -v_1X_w^1 & -v_1Y_w^1 & -v_1Z_w^1 & -v_1 \\ & & & & & & & \vdots & & & & \\ X_w^n & Y_w^n & Z_w^n & 1 & 0 & 0 & 0 & 0 & -u_nX_w^n & -u_nY_w^n & -u_nZ_w^n & -u_n \\ 0 & 0 & 0 & 0 & X_w^n & Y_w^n & Z_w^n & 1 & -v_nX_w^n & -v_nY_w^n & -v_nZ_w^n & -v_n \end{pmatrix}}_{\text{Q (this matrix is known)}} \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix}}_{\text{M (this matrix is unknown)}} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow \mathrm{Q} \cdot \mathrm{M} = 0$$

Linear system of Tsai's method

Eight-point Method

$$Q \cdot \bar{E} = 0$$

➢ Classical Version

Minimal solution

- $Q_{(n \times 9)}$ should have rank 8 to have a unique (up to a scale) non-trivial solution $\bar{E}$
- Different E matrices up to scale lead to the same result of rotation and translation (norm=1)
- Each point correspondence provides 1 independent equation
- Thus, 8 point correspondences are needed

Over-determined solution

- n > 8 points
- A solution is to minimize $||Q\bar{E}||^2$ subject to the constraint $||\bar{E}||^2 = 1$
- The solution is the eigenvector corresponding to the smallest eigenvalue of the matrix $Q^T Q$

Eight-point Method

- Extension to Fundamental matrix

✓ Similarly, eight-point method for fundamental matrix can be formulated as

$$\begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^{\mathrm{T}} \mathrm{F} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0 \quad \Rightarrow \quad \begin{bmatrix} u_2^1 u_1^1 & u_2^1 v_1^1 & u_2^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2 u_1^2 & u_2^2 v_1^2 & u_2^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^n u_1^n & u_2^n v_1^n & u_2^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n & u_1^n & v_1^n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

✓ We use the original image coordinates instead of normalized image coordinates.

Eight-point Method

➢ Normalized Version

$$\begin{bmatrix} u_2^1u_1^1 & u_2^1v_1^1 & u_2^1 & v_2^1u_1^1 & v_2^1v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2u_1^2 & u_2^2v_1^2 & u_2^2 & v_2^2u_1^2 & v_2^2v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^nu_1^n & u_2^nv_1^n & u_2^n & v_2^nu_1^n & v_2^nv_1^n & v_2^n & u_1^n & v_1^n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Motivation

✓ Orders of magnitude difference between column of data matrix.

✓ Least-square method yields poor results.

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
~10000	~10000	~100	~10000	~10000	~100	~100	~100	1

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

Orders of magnitude difference
between column of data matrix
→ least-squares yields poor results

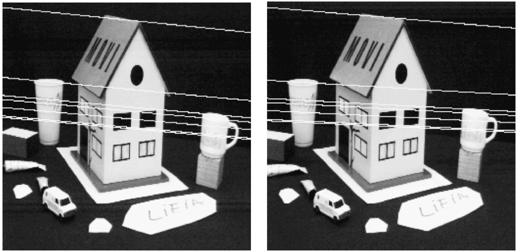
Eight-point Method

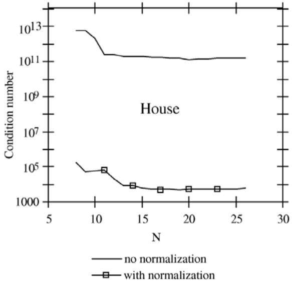
➢ Normalized Version

Motivation

✓ Poor numerical conditioning, which makes results very sensitive to noise

✓ This problem can be fixed by rescaling the data: Normalized 8-point algorithm [1]



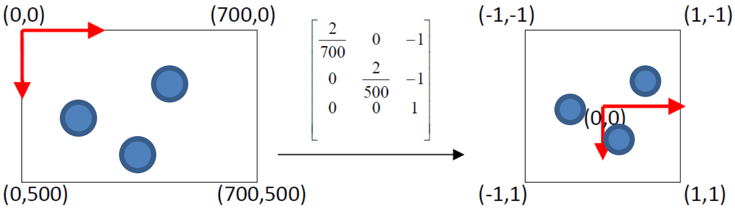


[1] R. Hartley, In defense of the eight point algorithm, IEEE Transactions of Pattern Analysis and Machine Intelligence (TPAMI), 1997

Eight-point Method

➢ Normalized Version

In the original 1997 paper, Hartley proposed to rescale the two 2D point sets such that the centroid of each set is 0 and the mean standard deviation $\sqrt{2}$ (equivalent to having the points distributed around a circled passing through the four corners of the $[-1,1] \times [-1,1]$ square).



Eight-point Method

➢ Normalized Version

✓ This can be done for every point as follows $\widehat{p^i} = \frac{\sqrt{2}}{\sigma}(p^i - \mu)$

where $\mu = (\mu_x, \mu_y) = \frac{1}{N}\sum_{i=1}^{n} p^i$ is the centroid and $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left\|p^i - \mu\right\|^2}$ is the standard deviation of the point set.

✓ This transformation can be expressed in matrix form using homogeneous coordinates:

$$\widehat{p^i} = \begin{bmatrix} \frac{\sqrt{2}}{\sigma} & 0 & -\frac{\sqrt{2}}{\sigma}\mu_x \\ 0 & \frac{\sqrt{2}}{\sigma} & -\frac{\sqrt{2}}{\sigma}\mu_y \\ 0 & 0 & 1 \end{bmatrix} p^i$$

Eight-point Method

$$\begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T \mathrm{F} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0$$

Fundamental Matrix

- Normalized Version

The Normalized 8-point algorithm can be summarized in three steps:

1. Normalize the point correspondences: $\widehat{p_1} = B_1 p_1$, $\widehat{p_2} = B_2 p_2$
2. Estimate normalized $\widehat{F}$ with 8 point algorithm using normalized coordinates $\widehat{p_1}$, $\widehat{p_2}$
3. Compute unnormalized *F* from $\widehat{F}$

Normalized 2D coordinates (known)

$$\widehat{p_2}^T \widehat{\mathrm{F}} \widehat{p_1} = 0$$

Output of second step

Original 2D coordinates

$$p_2^T B_2^T \; \widehat{\mathrm{F}} \; B_1 p_1$$

$$\mathrm{F} = \mathrm{B}_2^T \widehat{\mathrm{F}} \mathrm{B}_1$$

Associative law

Eight-point Method

- Normalized Version

Comparison between Normalized and non-normalized versions

	8-point	Normalized 8-point	Nonlinear refinement
Avg. Ep. Line Distance	2.33 pixels	0.92 pixel	0.86 pixel

Summary

- Overview of 2D-2D Geometry
- Two-view Geometric Constraints
- Eight-point Method

Thank you for your listening!

If you have any questions, please come to me :-)