4/20 Scribe

## *§ Structure from Motion*

**Problem Statement:**

-Given video sequence, want to find out parameters of camera or 3D structures of an object or a scene.

**What to solve?**

-Camera Parameters or 3D Structures.

***[Note] Match Move using Camera Calibration could get the most accuracy, however not easy to implement.***

☆ Basic Technique: SVD (Singular Value Decomposition)

Given $b_{(m\times1)}$:

$b_{(m\times1)}=A_{(m\times n)}x_{(n\times1)}$ → Transformation of **n** dimension to **m** dimension. Ex. 2D to 3D

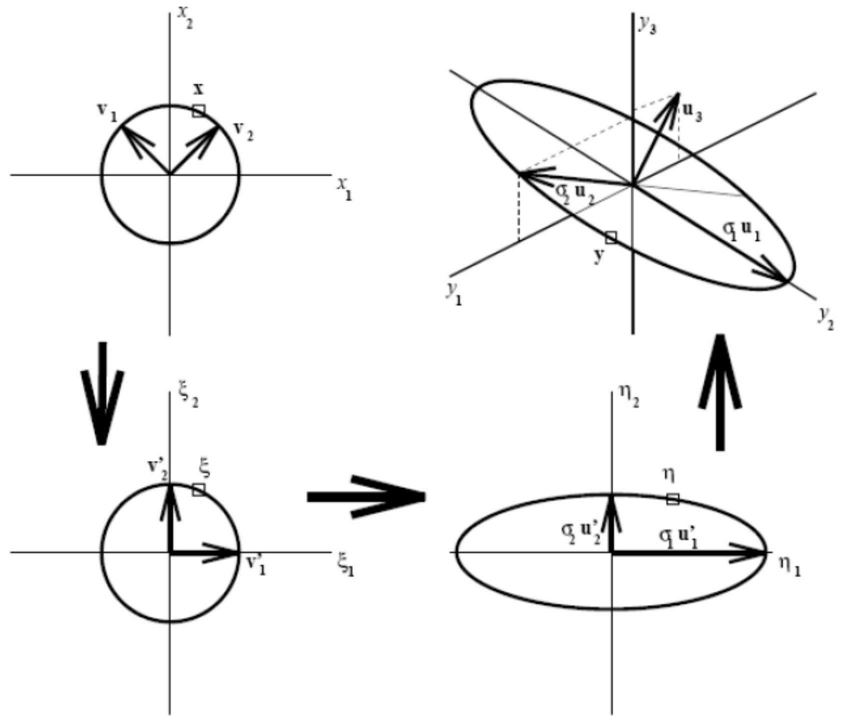
***[Note] the scaling may be anisotropic scaling.***

If $A_{(m\times n)}$ ➔ $A = U\Sigma V^T$

, where U(m×m), V(n×n) are both orthonormal matrices and Σ is the m×n diagonal matrix.

We can view the above system as:

V and Σ are for rotation(V) and scaling(Σ) transformation on **n dimension**, while U is for rotation transformation on **m dimension**. (Illustrated below)



◇To solve ***Ax=b,*** we'd like to find out the min norm least squares solution to

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|.$$

As a result, we need to look for the minimum of || ***A-A'***||, where rank(***A'***)=r < rank(***A***),

***[Note] Why do we find SVD?***

***∵A maybe affected by noise, such that it seems full-rank, while actually it isn't.***

➔ ***A'***= $V\Sigma^{\dagger}U^{T}$, and

$$\Sigma^{\dagger} = \begin{bmatrix} 1/\sigma_1 & & & & & 0 & \cdots & 0 \\ & \ddots & & & & & & \\ & & 1/\sigma_r & & & \vdots & & \vdots \\ & & & 0 & & & & \\ & & & & \ddots & & & \\ & & & & & 0 & 0 \cdots & 0 \end{bmatrix}$$

***A'*** is the pseudoinverse of ***A.***

Therefore $\hat{\mathbf{x}} = V\Sigma^{\dagger}U^{T}\mathbf{b}$ is the solution to $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|$.

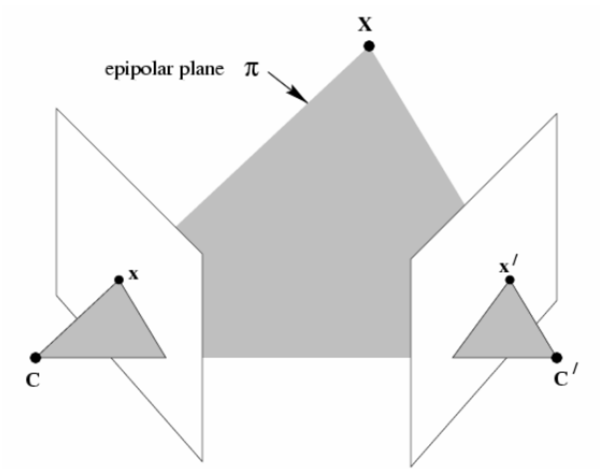
◇Important Applications of SVD:

a) Given one linear over-constrained matrix, find $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|$

b) Look for ***A'*** with regard to ***A***, such that rank(***A'***)<rank(***A***).

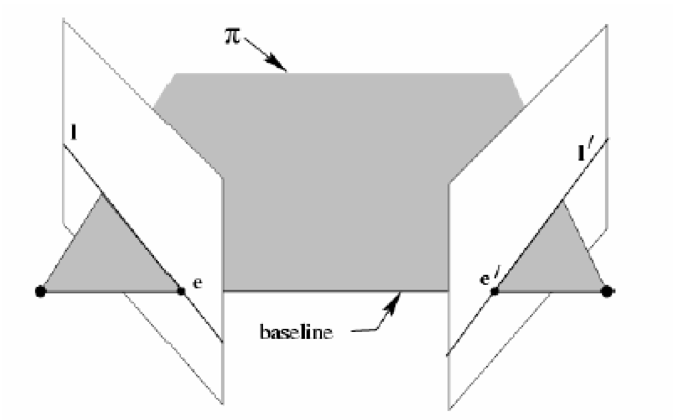
☆ Epipolar Geometry and Fundamental Matrix

The Geometry of the real object and camera positions:

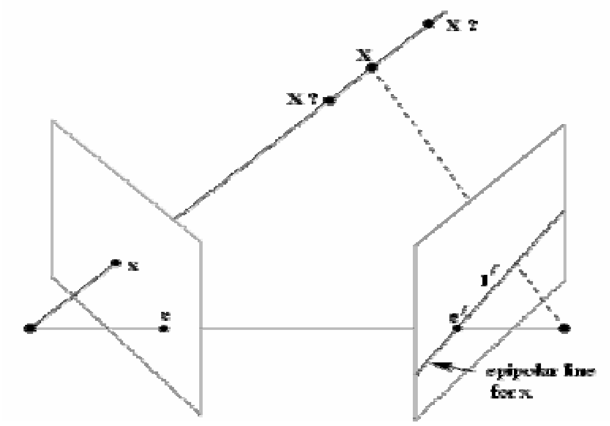



X: object position; C,C' : cameras; x: projection of X captured by C; x': projection of X captured by C'

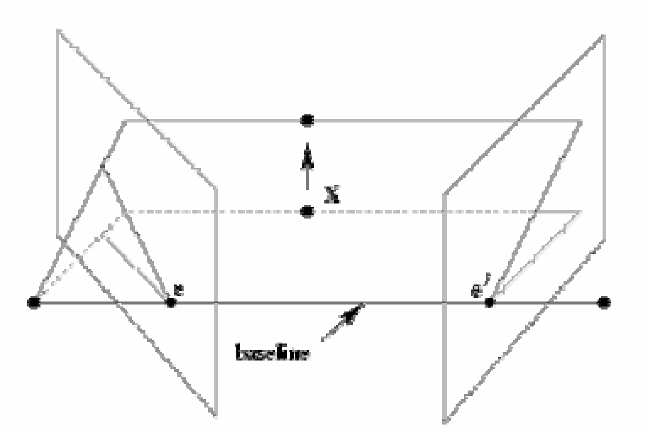
◇Expression of Epipolar pole, line, plane:
epipolar pole
= intersection of baseline with image plane
= projection of projection center in other image
epipolar plane = plane containing baseline
epipolar line = intersection of epipolar plane with image




◇ *If only C, C’, x are known…*




➔ x’ must lie on the epipolar line of the image plane determined by C’,
***[Note] Ex: Suppose there are 2 frames capturing the same object in different orientations, once we have the epipolar lines, to find the corresponding positions of some particular point in Frame1&2, the only positions we need to search from are only on the epipolar lines.***

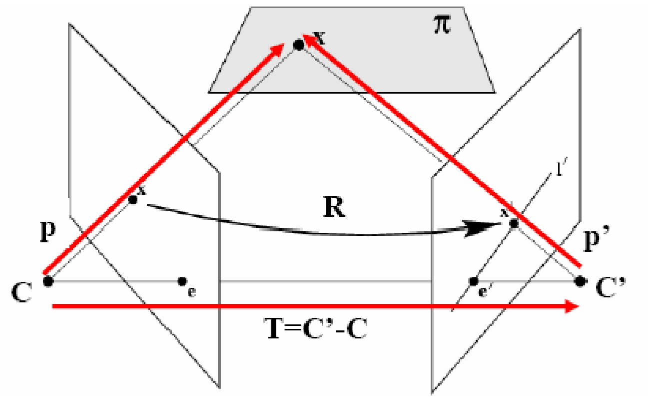



*Family of planes* $\pi$ *and lines l and l' intersects on epipolar poles e and e'.*

◇ Let

p=line of sight connecting X and C (X-C),

p'=line of sight connecting X and C' (X-C'), and T=C'-C




→ p'= R(p-T), and ∵ X,C,C' are coplanar

→ $(\mathbf{p}-\mathbf{T})^T(\mathbf{T}\times\mathbf{p})=0$

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

→

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

$$\mathbf{T}\times\mathbf{p}=\mathbf{S}\mathbf{p}$$

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

$$(\mathbf{R}^T\mathbf{p}')^T(\mathbf{S}\mathbf{p})=0$$

$$(\mathbf{p}'^T\mathbf{R})(\mathbf{S}\mathbf{p})=0$$

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

Therefore we get: $\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p}=0$ (1)

Let **M** and **M'** be the intrinsic parameters, then

$\mathbf{p}=\mathbf{M}^{-1}\mathbf{x}$ $\mathbf{p}'=\mathbf{M}'^{-1}\mathbf{x}'$ (2)

(2) substitute into (1), we get : $\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x}=0$ , where **F**= $\mathbf{M}'^{-\mathrm{T}}\mathbf{E}\mathbf{M}^{-1}$

F is the unique 3x3 rank 2 matrix that satisfies $x'^{T}Fx=0$ for all x↔x'.

*Given 2 images, if we can find the Fundamental Matrix F (note: rank(F)=2 with 7 degrees of freedom) ➔ we know where x in Frame#1 maps to the Frame#2*

◇ Solving F such that $\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x}=0$

Let $\mathbf{x}=(u,v,1)^{T}$, $\mathbf{x}'=(u',v',1)^{T}$

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

For each match:

$$uu'f_{11}+vu'f_{12}+u'f_{13}+uv'f_{21}+vv'f_{22}+v'f_{23}+uf_{31}+vf_{32}+f_{33}=0$$

***[Note] ∵ rank(F)=2, we let $f_{33}$ =1 and therefore we need only <u>8 equations</u> to solve the linear system Af=0.***

➔

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix}\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}=0$$

Instead of solving **Af=0**, we seek f to minimize **||Af||** .

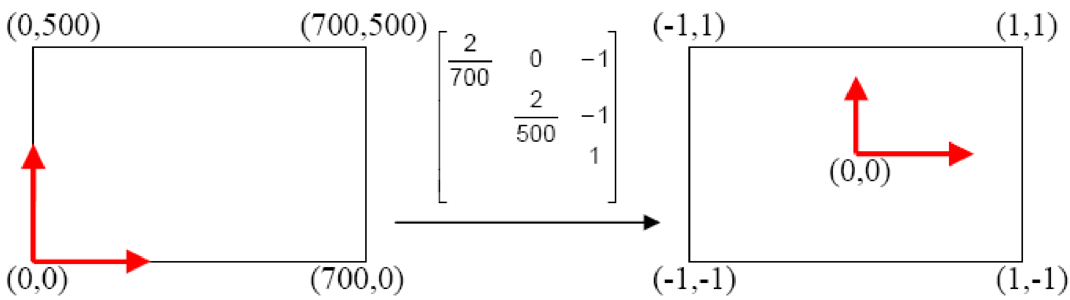
Note that F is of rank 2, so we replace F by F’ that minimizes ||F-F’||, det(F’)=0

- Find F’ using SVD!

→ $\mathbf{F}' = \mathbf{U}\Sigma'\mathbf{V}^{T}$ is the solution.

Though this “8 point algorithm” is linear and easy to implement, it is **susceptible** to noise because the ***orders of magnitude difference between column of data matrix are so large that least-squares yields poor results***.

Therefore, we *normalize the image size* to be within [-1,-1]~[1,1], shown as below:



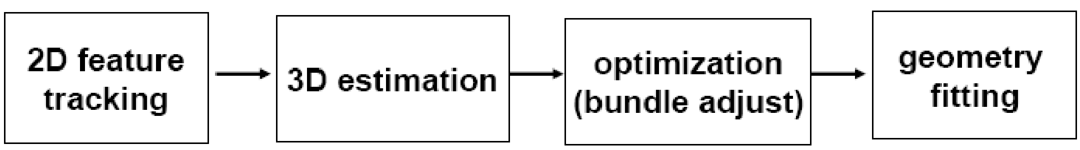

so that values of all u,v’s would lie in [-1,1] → least-squares yields good result!

**Now that we know how to solve for the fundamental matrix F, we could use *RANSAC* algorithm to repeatedly estimate the F to get the one with the largest portion of inliers!**

☆ Structure from Motion

◇The Idea: automatic recovery of **camera motion** and **scene structure** from two or more images. It is a self calibration technique and called *automatic camera tracking* or *matchmoving*.

◇Pipeline:




Step1: Track Features

Detect good and representing features, find correspondence points between frames.

Step2: Estimate Motion and Structures

Step3: Refine Estimate (Ex. Bundle Adjustment)

Step4: Use the Result from above to recover the surfaces

***[Note]***

1. ***Bundle Adjustment needs good initial guess.***
2. ***SIFT does not do the tracking. We could utilize the KLT tracking.***
3. ***Assume the scene captured by cameras is still.***
4. ***The estimation of lens distortion is very important to the recover of 3D structures.***
5. ***There are track life time for certain features, i.e. missing data.***

☆ Factorization Method

◇Idea: Given the 3D scene, and pictures are taken around the object, we'd like to recover the projection matrix and functions of 3D scene to the pictures.

$$\mathbf{q}_{ij} = \pi(\Pi_j \mathbf{p}_i)$$

, p: 3D scene point, q: 2D image point, $\Pi$ : projection matrix, $\pi$ : projection function

, j: jth image, i: ith point

The above equation could be reduced to:

2D image point → $\mathbf{q}$; orthographic projection matrix → $\Pi$; 3D scene point → $\mathbf{p}$; image offset → $\mathbf{t}$

$$\underset{2\times 1}{\mathbf{q}} = \underset{2\times 3}{\Pi}\,\underset{3\times 1}{\mathbf{p}} + \underset{2\times 1}{\mathbf{t}}$$

and with the trick of moving the origin to the centroid of the 3D & 2D points, we get

$$\mathbf{q} = \Pi\mathbf{p}$$

→

projection of *n* features in *m* images

$$\underset{2\mathrm{m}\times\mathrm{n}}{\begin{bmatrix} \mathbf{q}_{11} & \mathbf{q}_{12} & \cdots & \mathbf{q}_{1n} \\ \mathbf{q}_{21} & \mathbf{q}_{22} & \cdots & \mathbf{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_{m1} & \mathbf{q}_{m2} & \cdots & \mathbf{q}_{mn} \end{bmatrix}} = \underset{2\mathrm{m}\times 3}{\begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_m \end{bmatrix}} \underset{3\times\mathrm{n}}{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix}}$$

**W** measurement **M** motion **S** shape ***[Note] rank(W)<=3***

Now we know the relation between measurement and shape, and the measurement **W** are known, therefore we need to solve for **M, S**

➔ use ***SVD*** to decompose **W** !

$$\underset{2\mathrm{m}\times\mathrm{n}}{\mathbf{W}} = \underset{2\mathrm{m}\times 3}{\mathbf{M}'}\,\underset{3\times\mathrm{n}}{\mathbf{S}'} \rightarrow \mathbf{W} = \mathbf{M}'\mathbf{S}' = (\mathbf{M}\mathbf{A}^{-1})(\mathbf{A}\mathbf{S})$$

and with the constraint : $\mathbf{M}'\mathbf{A} = \mathbf{M}$, we could solve for **A !**

If affected by noise:

$$\underset{2m\times n}{\mathbf{W}} = \underset{2m\times 3}{\mathbf{M}}\ \underset{3\times n}{\mathbf{S}} + \underset{2m\times n}{\mathbf{E}}$$

→ SVD gives this solution

– Provides optimal rank 3 approximation **W'** of **W**

$$\underset{2m\times n}{\mathbf{W}} = \underset{2m\times n}{\mathbf{W}'} + \underset{2m\times n}{\mathbf{E}}$$

☺☺☺