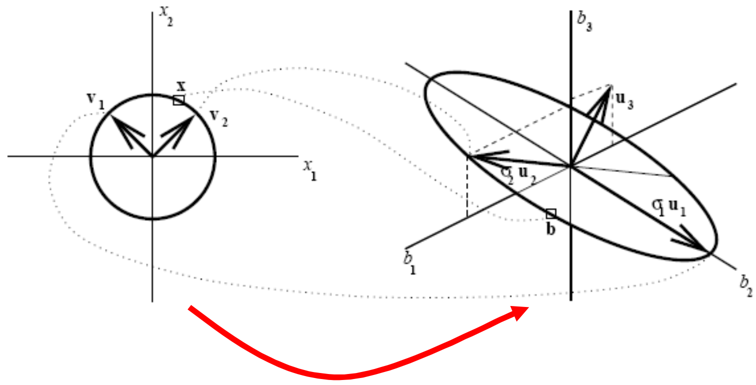
(1) Singular value decomposition(SVD)

(a)每個 matrix 都可表示成一個 transformation，如下圖：



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

$$A = \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{3} & \sqrt{3} \\ -3 & 3 \\ 1 & 1 \end{bmatrix}$$

reference：http://www.uwlax.edu/faculty/will/svd/index.html

(b)Singular value decomposition(SVD)的做法：

Theorem 3.2.1 *If A is a real $m \times n$ matrix then there exist orthogonal matrices*

$$\begin{aligned} U &= \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix} \in \mathcal{R}^{m\times m} \\ V &= \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} \in \mathcal{R}^{n\times n} \end{aligned}$$

such that

$$U^T AV = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p) \in \mathcal{R}^{m\times n}$$

where $p = \min(m, n)$ and $\sigma_1 \geq \ldots \geq \sigma_p \geq 0$. Equivalently,

$$A = U\Sigma V^T\ .$$

The SVD reveals a great deal about the structure of a matrix. If we define r by

$$\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = 0\ ,$$

that is, if σ_r is the smallest nonzero singular value of A, then

$$\mathrm{rank}(A) \quad = \quad r$$

其中 Matrix A 可拆成如下：

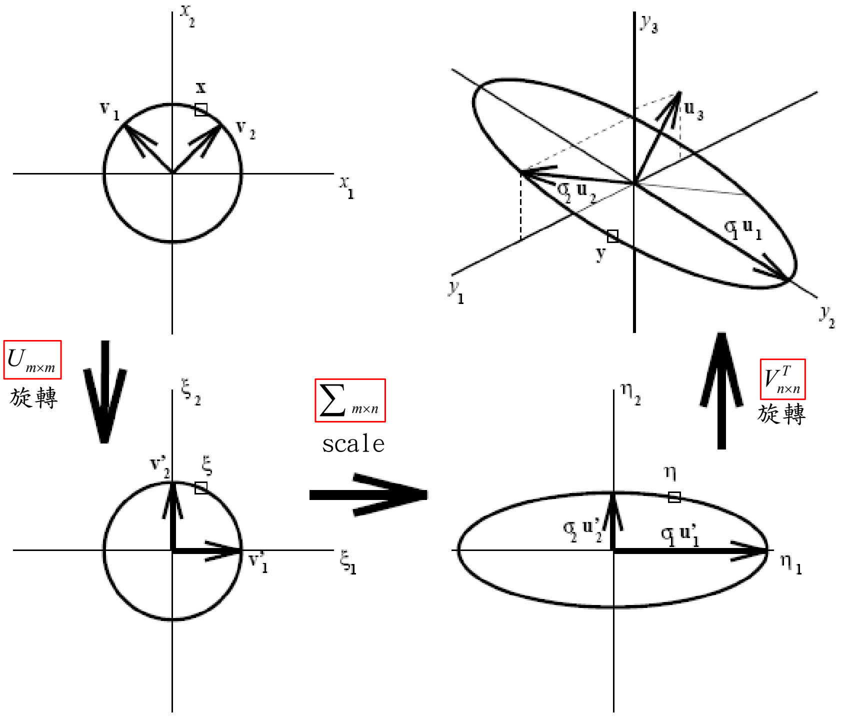
$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}$$

$U_{m\times m}$ ：為 orthogonal matrix，代表旋轉。

$\sum_{m\times n}$：為 diagonal matrix，代表 scale，如右：$\begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 \end{bmatrix}$

$V_{n\times n}^T$：為 orthogonal matrix，代表旋轉。

如下圖：



(c) Pseudoinverse：

Theorem 3.3.1 *The minimum-norm least squares solution to a linear system $A\mathbf{x} = \mathbf{b}$, that is, the shortest vector $\mathbf{x}$ that achieves the*

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

is unique, and is given by

$$\hat{\mathbf{x}} = V\Sigma^{\dagger}U^T\mathbf{b}$$

where

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

is an $n \times m$ diagonal matrix.

The matrix

$$A^{\dagger} = V\Sigma^{\dagger}U^T$$

is called the *pseudoinverse* of A.

由 A 找一個與 A 最近的 matrix A'，即$\|A - A'\|$是最小。其中

$rank(A') = r < rank(A)$，對 A' 做 SVD，即 $A' = U\sum{}'V^T$，A' 的 rank 變小，至於多小，自己指定。

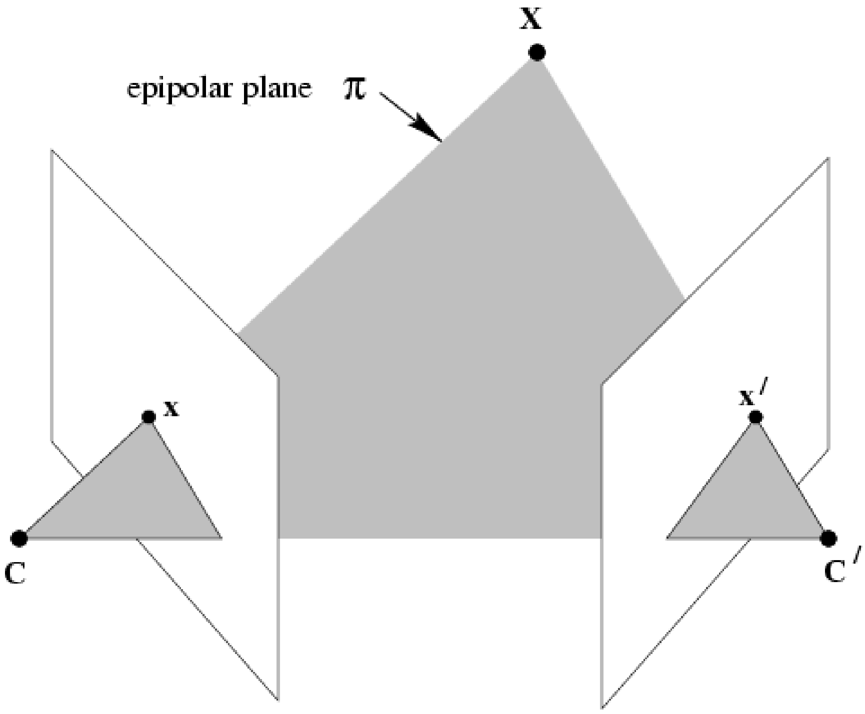
共 r 項

$$\sum{}' = \begin{bmatrix} x & & & & & 0 & 0 & 0 \\ & x & & & & 0 & 0 & 0 \\ & & x & & & 0 & 0 & 0 \\ & & & 0 & & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \end{bmatrix}$$

(2) Epipolar geometry and fundamental matrix

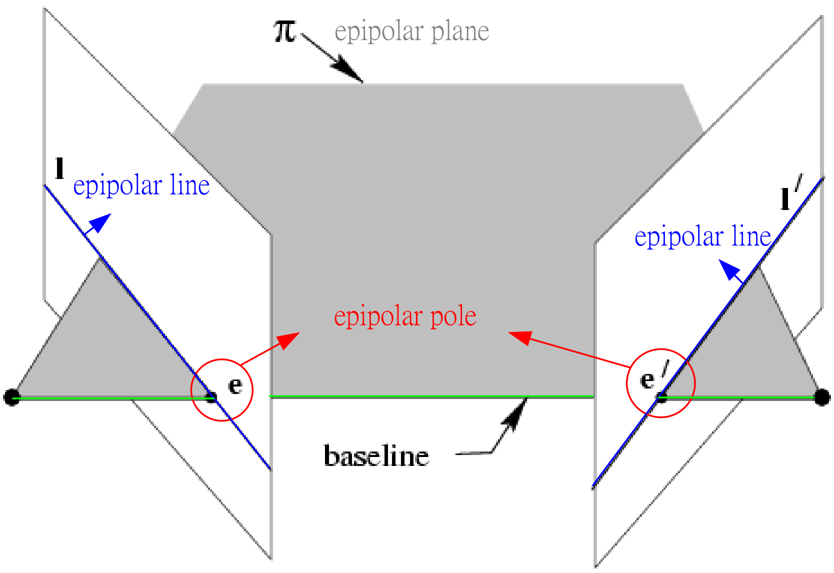
(a) The epipolar geometry：

C、C'、x、x'和 X 五點共平面，其中 C 為相機 1，C'為相機 2，X 為 C 和 C'共同見到之 3D 的 scene，x 和 x'分別為 X 在 C 和 C'上的投影 (2D point)。



epipolar pole(e，e')
= intersection of baseline with image plane
= projection of projection center in other image

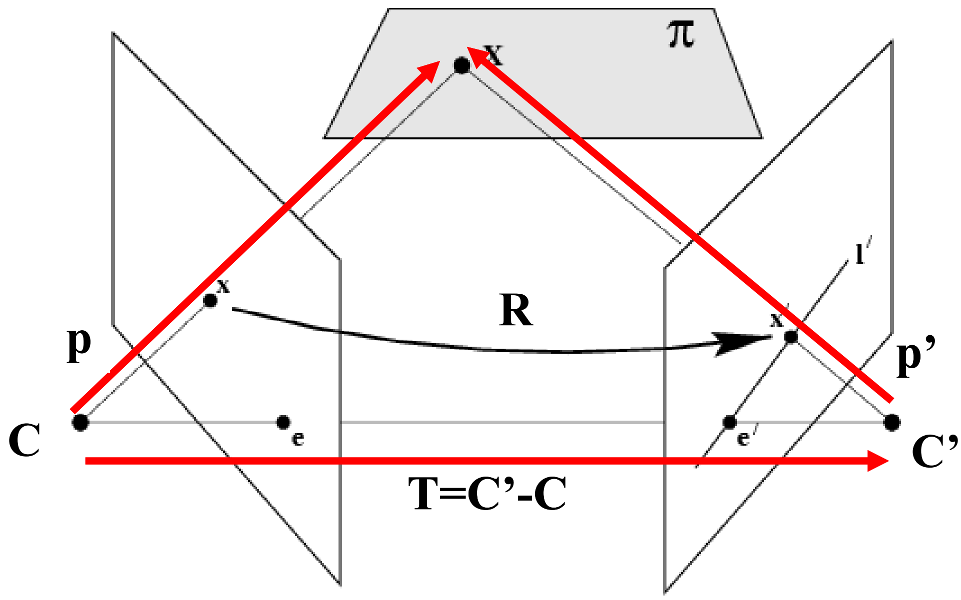
epipolar plane(π) = plane containing baseline
60epipolar line(l，l') = intersection of epipolar plane with image



epipolar geometry demo：

http://www-sop.inria.fr/robotvis/personnel/sbougnou/Meta3DViewer/EpipolarGeo.html

(b) The fundamental matrix F：



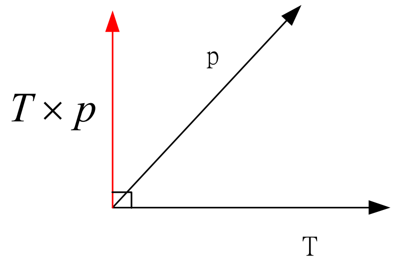
Two reference frames are related via the extrinsic parameters

$$p' = R(p-T)$$

The equation of the epipolar plane through X is

$$(p-T)^T(T\times p)=0 \;\Rightarrow\; (R^T p')^T(T\times p)=0$$

其中 $(p-T)$ 為 epipolar plane，$(T\times p)$ 為 epipolar plane 的法線，所以兩個垂直，內積為 0。令 $p-T\ =R^{-1}p'=R^T p'$，R 為 orthonormal matrix，所以 $R^{-1}=R^{T}$ 。



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

其中 $S=(x1,y1,z1)\times(x2,y2,z2)=\begin{vmatrix} i & j & k \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{vmatrix}=\begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$

➔ $(R^{T}p')^{T}(Sp)=0$ ➔ $(p'^{T}R)(Sp)=0$

令 $RS=E$ ，R 表示 Rotation，S 表示 Transformation，E 爲 Essential matrix。

➔ $(p'^{T}Ep)=0$

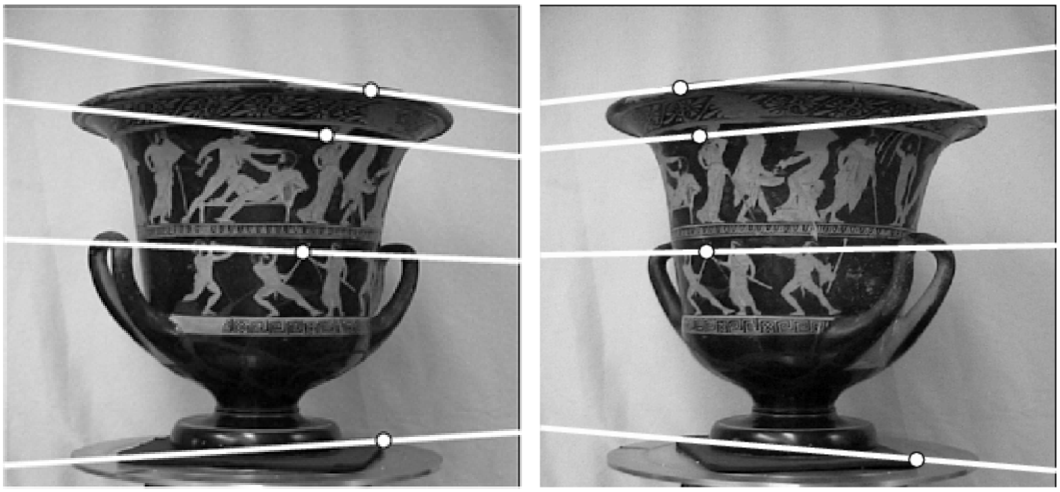
假如我們知道 p 和 p' 的關系，那能否找到 x 和 x' 的關系?
令 M 為 intrinsic matrix：

$$x=Mp \quad ➔ \quad p=M^{-1}x$$
$$x'=M'p' \quad ➔ \quad p'=M'^{-1}x'$$

➔ $(M'^{-1}x)^{T}E(M^{-1}x)=0$ ➔ $x'^{T}M'^{-T}EM^{-1}x=0$ ➔ $x'^{T}Fx=0$

其中 F 為 fundamental matrix。

兩張 images 只要找出 F ，即可找到對應點，如下圖。其中 F 至少要有 8 個 match，rank(F)=2。如找到的 F 之 rank≠2, 則找與 F 最近的 matrix F' ，即 $\|F-F'\|$ 為最小，用 SVD(參考前面說明)解 F' ，最後得到 rank(F') = 2。



<u>(c)</u> Estimation of F — 8-point algorithm：

由 $x'^T Fx = 0$ 知，兩張 images 只要找出 F ，即可找到對應點，

令 $x = (u, v, 1)^T$ 和 $x' = (u', v', 1)^T$ ， $F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

而每組對應點可得一線性方程式如下：

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

八組對應點帶入方程式後可得以下的線性系統：

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

實際上，我們通常會找 f 使 $\|Af\|$ 為最小，且為 $A^T A$ 的 least eigenvector，而不直接解 $Af = 0$ 。

若得到之 F 的 rank 若不為 2，則找 F' 使得 $\|F - F'\|$ 為最小來取代 F ，可

用 SVD 來解，令 $F=U\sum V^T$，$\sum=\begin{bmatrix} x1 & 0 & 0 \\ 0 & x2 & 0 \\ 0 & 0 & x3 \end{bmatrix}$，

令 $\sum=\begin{bmatrix} x1 & 0 & 0 \\ 0 & x2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ ➔ rank 為 2，所以 $F'=U\sum{}'V^T$ 即為解。

優點：為 linear、易完成，速度快。

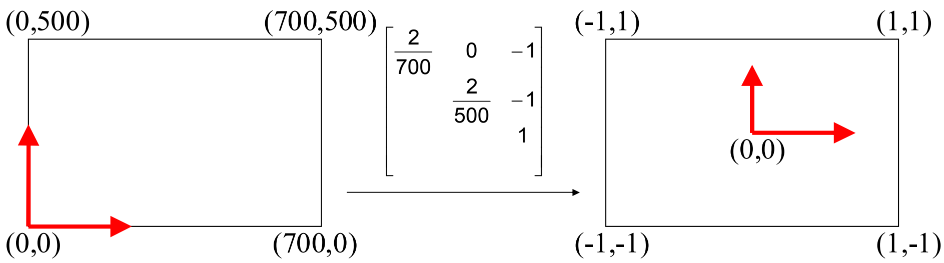
缺點：易受 noise 影響，原因是 Matrix A 的 column 間的值差異太多，即使是小 noise 也會造成大誤差。

解決方法：用 Normalized 8-point algorithm

(d) Normalized 8-point algorithm：

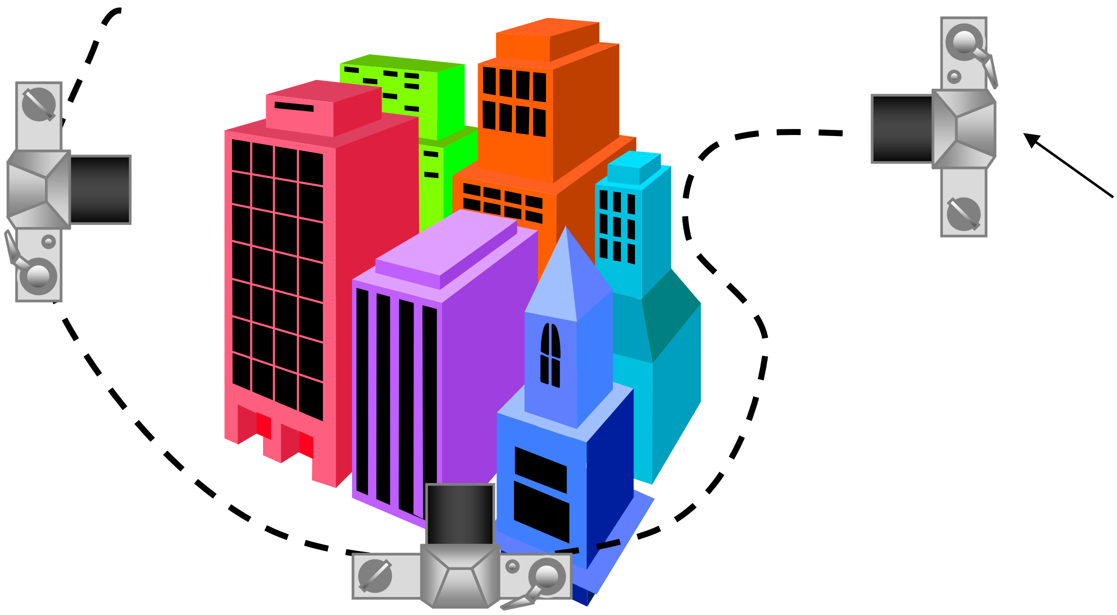
1. Transform 兩對應的輸入點 x 和 x'， $\hat{x}_i=Tx_i$，$\hat{x}'_i=Tx'_i$。
2. 以 $\hat{x}_i$ 和 $\hat{x}_i{}'$ 呼叫 8-point algorithm，以獲得 $\hat{F}$
3. $F=T'^T\hat{F}T$

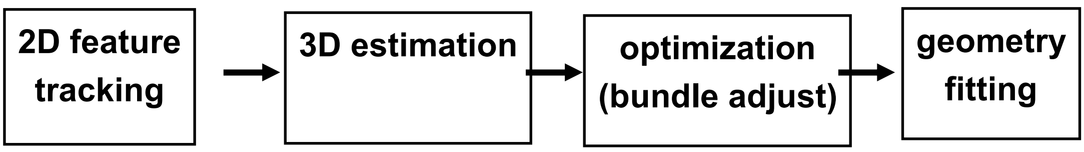
Normalized 後，將原點移至中心，範圍在[-1，1]x[-1，1]之間，如下圖。



(3)Structure from motion

自動從兩張或多張 images 重建 camera motion 和 scene structure。SFM 是一種自我較正的技術，稱 automatic camera tracking or matchmoving。通常應用在電腦視覺(CV)、multiple-view shape reconstruction， novel view synthesis and autonomous vehicle navigation。在電影製作過程中，也用來將 CGI 插入真實場影中，如電影 Jurassic Park 就使用此技術。





SFM pipeline

(a)SFM 的步驟：

- Step1：Track Features

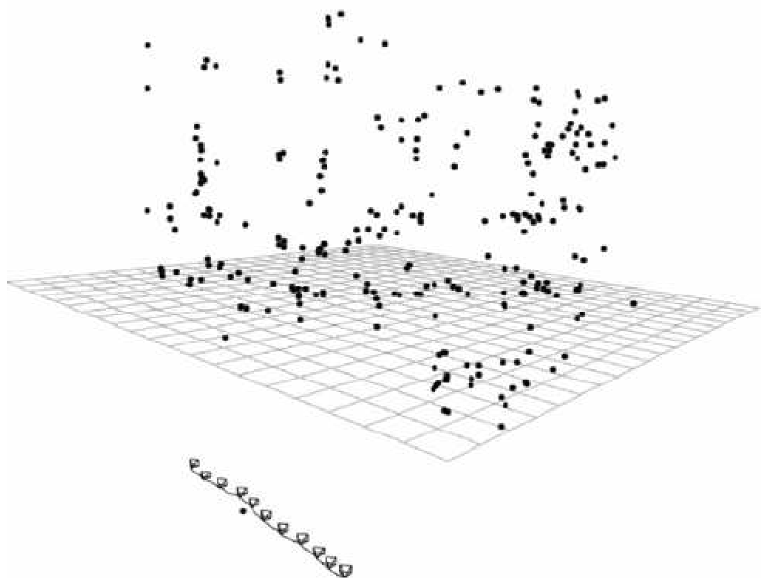
  -找出 frame 間相對應的 features，有幾種方法，其中 SIFT 效果比較好，KLT Tracking 效果較差

- Step 2：Estimate Motion and Structure

  -Simplified projection model，e.q.，[Tomasi 92]

  -2 or 3 views at a time [Hartley 00]

- Step 3：Refine estimates
  -"Bundle adjustment" in photogrammetry
  -Other iterative methods



- Step 4：Recover surfaces(image-based triangulation，silhouettes，stereo…)

(b) Issues in SFM

- Track lifetime
  說明每個 track point 生命期只有 n 個 frame，這樣可能造成 tracking 錯誤，所以每個 feature 應乘一個 weight，生命期愈長，乘的 weight 愈小。
- Nonlinear lens distortion
  3D 的 recovery 須考慮 lens distortion 的問題。
- Degeneracy and critical surfaces
- Prior knowledge and scene constraints
- Multiple motions

(4)Factorization methods

(a)Notation：

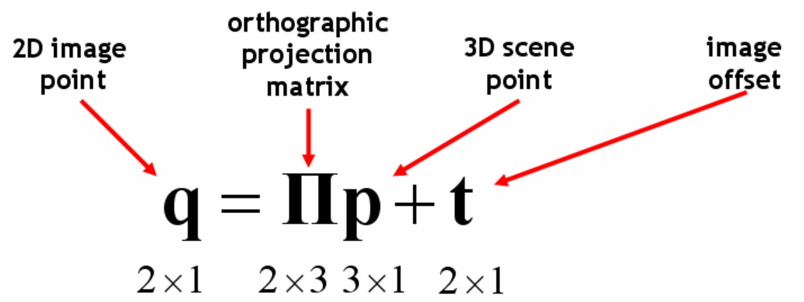
- n 3D points are seen in m views
- q=(u,v,1): 2D image point
- p=(x,y,z,1): 3D scene point
- Π: projection matrix

- π: projection function
- q_{ij} is the projection of the i-th point on image j
- λ_{ij} projective depth of qij

$q_{ij} = \pi(\Pi_j p_i)$，$\pi(x, y, z) = (x/z, y/z)$，$\lambda_{ij} = z$

<u>(b)</u> SFM under orthographic projection：

Homogeneous $q_{3\times1} = \Pi_{3\times4} p_{4\times1}$ ➔ $q_{3\times1} = \Pi_{3\times3} p_{3\times1} + t$
Orthographic $q_{2\times1} = \Pi_{2\times3} p_{3\times1} + t_{2\times1}$



爲了運算方便，將 translation 去掉，即將 3D 之 Scene 的原點移至 3D 的 centroid，2D image 的原點移至 2D 的 centroid，做法如下：

原輸入點(u_1, v_1)……. (u_n, v_n)，

$\bar{u} = \frac{1}{n}\sum u_i$ ，$\bar{v} = \frac{1}{n}\sum v_i$ ➔ centroid 即爲$(\bar{u}, \bar{v})$ 。

得到新的輸入點：

$u_i' = u_i - \bar{u}$ ，$v_i' = v_i - \bar{v}$ ，$q_i = \begin{pmatrix} u_i' \\ v_i' \end{pmatrix}$

最後得到：$q = \Pi p$

<u>(c)</u> factorization (Tomasi & Kanade)：

n features 在一張 image 的投影如下：

$$\underset{2\times n}{\begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}} = \underset{2\times 3}{\prod} \underset{3\times n}{\begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}}$$

則n features 在m張 image 的投影：

$$\underbrace{\underset{2m\times n}{\begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix}}}_{W_{measurement}} = \underbrace{\underset{2m\times 3}{\begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_m \end{bmatrix}}}_{M_{motion}} \underbrace{\underset{3\times n}{\begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}}}_{S_{shape}}$$

其中 rank(W) <= 3。

$$\text{known} — \underset{2m\times n}{\mathbf{W}} = \boxed{\underset{2m\times 3}{\mathbf{M}}\ \underset{3\times n}{\mathbf{S}}} — \text{solve for}$$

Fractorization 的技術是找一個 rank=3 的 array MS，去近似一個 $2m\times n$ 的 array W，其中 W 的 rank<=3(no noise)。

對 W 做 SVD➔ $W_{2m\times n} = U_{2m\times 2m} \Sigma_{2m\times n} V_{n\times n}{}^T$，其中

$$\Sigma = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & 0 & \\ & & \lambda_3 & & \\ & & & \ddots & \\ & 0 & & 0 & \\ & & & & 0 \end{bmatrix}$$

rank(Σ)=3, 所以我們可以強制將 Σ 轉成 $\Sigma'_{3\times 3}$，

$$W'_{2m\times 3} = U'_{2m\times 3} \Sigma'_{3\times 3} V'^T_{3\times n} = (U'\sqrt{\Sigma'})(\sqrt{\Sigma'}V'^T) = M'S'$$，

$M' = U'\sqrt{\Sigma'}$，$S' = \sqrt{\Sigma'}V'^T$

S' differs from S by a linear transformation A：

$$W = M'S' = (MA^{-1})(AS)$$

只要對 M 做 metric constraints 即可解 A。

由於相機的投影是 Orthographic，所以Π(projection matrix)的每個row vector 必須是 Orthonormal，即滿足：$\Pi\Pi^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$。因此 M 的 row vectors 也必須是 Orthonormal，有了這個 constraint，便可解 A，進而解 $M = M'A$。

另一方法是令 $\Pi = \Pi' A$，則

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \Pi\Pi' = \Pi' A(\Pi' A)^T = (\Pi' A)(A^T \Pi'^T) = \Pi' G \Pi'^T \text{，} G = AA^T$$

先寫出在 M 中所有 Π_i 的 equation，以解出 G；再對 G 做 SVD($U = V$)，then $G = AA^T$，便可解 A。