- $\mathbf{x}_i = P\,\mathbf{X}_i$
- $\mathbf{x}_i$：影像座標點，3×1 / 3×n，Homogeneous
- **P**：投影矩陣，3×4
- $\mathbf{X}_i$：世界座標點， 4×1 / 4×n，Homogeneous
- (Homogeneous) $x_i = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}$
- (Cartesian) $x_i = (p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}) / (p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34})$
- 移項： $x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = (p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14})$
- $(p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}) - x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = 0$

- 誤差項Ap(分為影像平面上，X方向和Y方向對應的):

$$(p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}) \ - x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34})$$

$$(p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}) \ - y_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34})$$

X_1	Y_1	Z_1	1	0	0	0	0	$-x_1X_1$	$-x_1Y_1$	$-x_1Z_1$	$-x_1$
0	0	0	0	X_1	Y_1	Z_1	1	$-y_1X_1$	$-y_1Y_1$	$-y_1Z_1$	$-y_1$
X_2	Y_2	Z_2	1	0	0	0	0	$-x_2X_2$	$-x_1Y_2$	$-x_2Z_2$	$-x_2$
0	0	0	0	X_2	Y_2	Z_2	1	$-y_2X_2$	$-y_2Y_2$	$-y_2Z_2$	$-y_2$
…											
X_n	Y_n	Z_n	1	0	0	0	0	$-x_nX_n$	$-x_nY_n$	$-x_nZ_n$	$-x_n$
0	0	0	0	X_1	Y_n	Z_n	1	$-y_nX_n$	$-y_nY_n$	$-y_nZ_n$	$-y_n$

A is a $2n \times 12$ matrix

- 分解Projection Matrix為$K_{(3\times3)}$ [$R_{(3\times3)}$ | $t_{(3\times1)}$]
- 其中，利用K為upper triangular，而R為orthonormal，可以先解出K、R，再透過K解出t，K對應RQ分解中的R，而R則對應Q
- 使用RQ(QR)分解，P1_stuff/QR_Decomp.m 使用Gram-Schmidt分解法，而P1_stuff/RQ_Decomp.m是基於QR_Decomp.m所解出之RQ=A，對應KR=M。使用純量積調整K、R，使得$K_{3,3}=1$。
- 最後再根據K和Kt的關係，即t = K \ Kt解出t。
    - Matlab鼓勵使用矩陣除法而不是先求反矩陣再相乘，他們說這樣可以有較小的誤差，此誤差來源可能是floating point運算時的truncation error

1b

P

0.443608834160014	-0.019974437336475	0.243106918114603	0.020847564064082
0.003944936056198	0.446877511518878	0.307997579388433	0.027968140786298
-0.023234343373376	0.023234216539569	0.664939316782719	0.068504542784123

K

0.679700979213296	-0.019012074507017	0.340415482404809
0	0.655073327196395	0.485287148265281
0	0	1

α_x	s	x_0
	α_y	y_0
		1

R

0.664939308847748	-0.022423565351371	0.024017864178132
0.023234443629113	0.664967152145422	-0.022423330049757
-0.023234343373377	0.023234216539569	0.664939316782718

t

-0.003862836985486
0.008054416514267

- $P_{(3\times4)} = K_{(3\times3)} [\ R_{(3\times3)} \mid t_{(3\times1)}\ ]$

1c

- 最小化誤差$Ap_{(2n\times1)}$各項的平方和
- 使用1a和1b分解後重組的$p_{(3\times4)}$皆為

0.443608834160014	-0.019974437336475	0.243106918114603	0.020847564064082
0.003944936056198	0.446877511518878	0.307997579388433	0.027968140786298
-0.023234343373376	0.023234216539569	0.664939316782719	0.068504542784123

- SSE , MSE, RMSE generated by P1(a) : 2.4338×10^{-6} , 1.2169×10^{-8} , 1.103121×10^{-4}
- SSE , MSE, RMSE generated by P1(b) : 2.4338×10^{-6} , 1.2169×10^{-8} , 1.103121×10^{-4}
- Error differences between P1(a) & P1(b) : 0 , 0 , 0
  - RMS error為誤差的平方平均(root mean square)，
    與樣本數量較無關，為標準化後的結果

- Homography 是將3D平面投影到2D影像平面時，所產生的特例，它可以減少一些自由度，使得計算上較為簡單。
- 右下角的表格描述$\mathbf{x}_i=\mathbf{P}\mathbf{X}_i$
- 然而，若$Z_i=0$，那麼我們可以忽視Z和P的第3個Column
- 因此，我們便能透過和P1(a)類似的方法，求出Homography

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

- Homography (由左邊海報到右邊海報)

0.008465291601476	0.001017393990707	0.257049195049130
0.005210794269711	0.004542552629287	-0.892567997592930
6.335458888592164e-06	7.072902027047134e-07	0.002845193648442

- 逆對應的Homography即為其反矩陣

置換左右兩張電影海報

2b

- 使用Bi-linear interpolation以增進影像品質(使其更平滑)
- 您可以透過調整P2.m的「這個參數」來開啟或關閉這個功能

```
%雙線性內插
img_post_R = getPatchTransform( im1f, mask_R, inv(hmgphyMat), true ); %把左邊的貼到右邊
img_post_L = getPatchTransform( im1f, mask_L, hmgphyMat, true ); %把右邊的貼到左邊
```

Using NN interpolation

Using bi-linear interpolation

- 使用Alpha Blending以增進影像品質
- 可以減少鋸齒邊緣的出現

Before

After