Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3 - Option IRV Premier Semestre 2008/2009

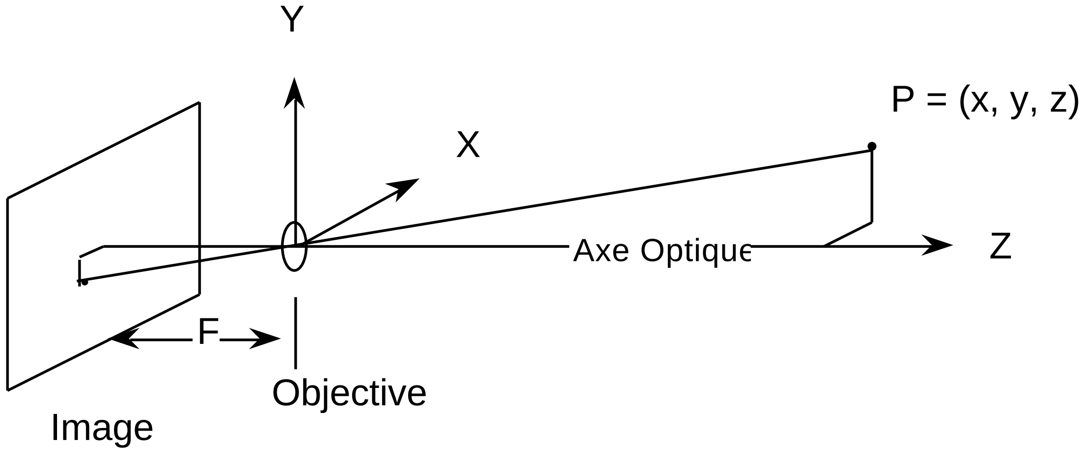
Séance 2 10 octobre 2008

Plan de la Séance :

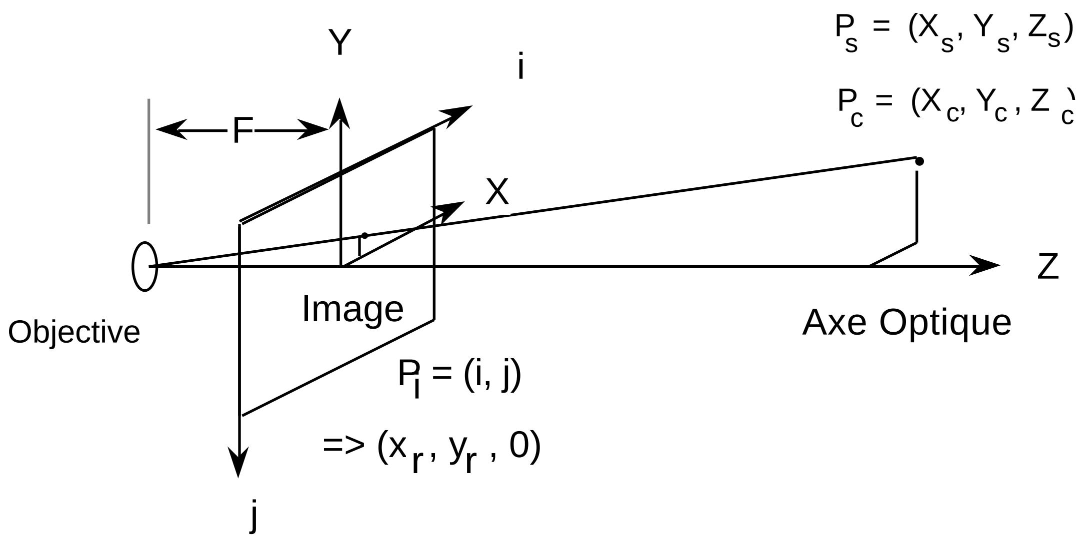
Modèle de la Caméra

Modèle de la Caméra

Modèle de la Caméra   Géométrie Physique



Modèle Mathématique: Projection Centrale



Les Repères

Coordonnées de la Scène :

Point Scène : $P^s = (x_S, y_S, z_S, 1)^T$

Coordonnées de la Caméra :

Point Caméra : $P^c = (x_C, y_C, z_C, 1)^T$

Point Image : $P^r = (x_r, y_r, 1)^T$

Coordonnées de l'Image :

Point Image : $P^i = (i, j, 1)^T$

Nota : En coordonnées homogènes, les points sont invariant au multiplication par un constant.

$(i, j, 1)^T = A\cdot(i, j, 1)^T = (Ai, Aj, A)^T$

Transformations entres reperes

La modèle de la caméra est composé d'une composition de transformations.

Transformation entre repère <u>Caméra</u> et repère <u>Scène</u> :

$$P^c = \mathbf{T}_s^c P^s$$

Matrice de Projection ${}_c^r\mathbf{P}$ du repère <u>Caméra</u> vers le repère <u>Rétine</u>

$$P^r = \mathbf{P}_c^r P^c$$

Transformation entre repère image et repère caméra

$$P^i = \mathbf{C}_r^i \; w \, P^r$$

Composition : $P^i = \mathbf{C}_r^i \; \mathbf{P}_c^r \; \mathbf{T}_s^c \; P^s = \mathbf{M}_s^i \; P^s$

La Transformation Scène - Caméra

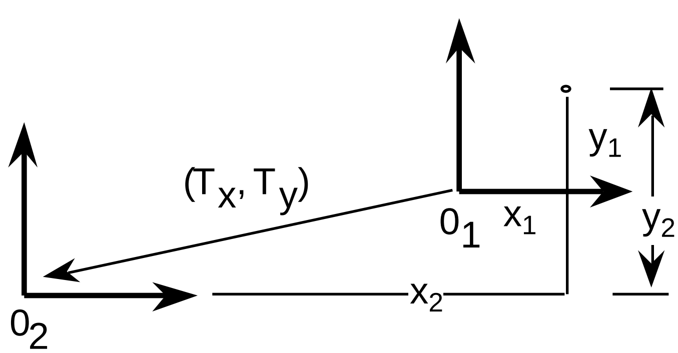
Dans les coordonnées homogènes, les translations, rotations, transformées affines et perspectives sont représentées par les produits des matrices.

La transformation $\mathbf{T}_s^c$ a la forme :

$$\mathbf{T}_s^c = \begin{pmatrix} & \mathbf{R}_s^c & & x_s \\ & & & y_s \\ & & & z_s \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

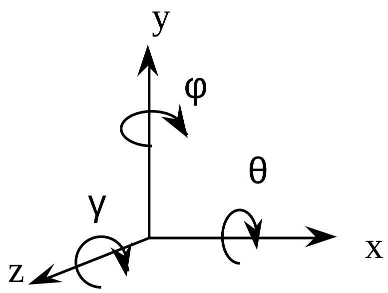
ou (x_s, y_s, z_s) est la position du repère scène dans le repere caméra.
et $\mathbf{R}_s^c$ est l'orientation du repère scène dans le repère caméra.

Translation Addition en espace cartésien
Multiplications en Coordonnées Homogènes



T est un vecteur (vecteur de translation) de P_1 vers P_2

En 3D



Au tour de l'axe X :

$$\mathbf{R}_X(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{Cos}(\theta) & \mathrm{Sin}(\theta) & 0 \\ 0 & -\mathrm{Sin}(\theta) & \mathrm{Cos}(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

l'axe Y :

$$\mathbf{R}_Y(\varphi) = \begin{pmatrix} \mathrm{Cos}(\varphi) & 0 & -\mathrm{Sin}(\varphi) & 0 \\ 0 & 1 & 0 & 0 \\ \mathrm{Sin}(\varphi) & 0 & \mathrm{Cos}(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

l'axe Z :

$$\mathbf{R}_Z(\gamma) = \begin{pmatrix} \mathrm{Cos}(\gamma) & \mathrm{Sin}(\gamma) & 0 & 0 \\ -\mathrm{Sin}(\gamma) & \mathrm{Cos}(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

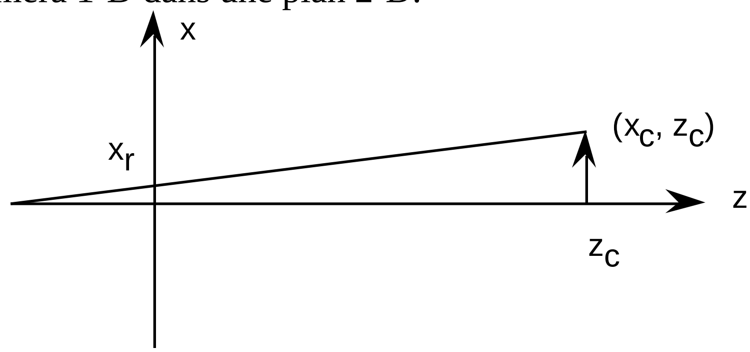
En Générale :

$$\mathbf{T} = \begin{pmatrix} & \mathbf{R} & & T_x \\ & & & T_y \\ & & & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{ou } \mathbf{R} = \mathbf{R}_Z(\gamma)\, \mathbf{R}_Y(\varphi)\, \mathbf{R}_X(\theta)$$

La Projection Caméra - Rétine (Le modèle Stenope)

La projection de la caméra vers la rétine est une projection perspective.

Considère une caméra 1-D dans une plan 2-D.



Dans le repère de la caméra :

$(x_c, y_c, z_c, 1)$: Point dans la scène en repère caméra
$(x_r, y_r, 1)$: Point dans la rétine en repère rétine.

Par triangles semblables :

$$\frac{x_r}{F} = \frac{x_c}{z_c} \quad \Leftrightarrow \quad x_r = x_c \frac{F}{z_c} \quad \Leftrightarrow \quad x_r \frac{z_c}{F} = x_c$$

$$\frac{y_r}{F} = \frac{y_c}{z_c} \quad \Leftrightarrow \quad y_r = y_c \frac{F}{z_c} \quad \Leftrightarrow \quad y_r \frac{z_c}{F} = y_c$$

soit : $w = \frac{z_c}{F}$

alors :
$$w\, x_r = x_c$$
$$w\, y_r = y_c$$
$$w = \frac{z_c}{F}$$

En matrice :

$$\begin{pmatrix} w x_r \\ w y_r \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

La transformation du repère <u>Caméra</u> vers le repère <u>Rétine</u> est :

$$P^r = \begin{pmatrix} w x_r \\ w y_r \\ w \end{pmatrix} = \mathbf{P}_c^r \; P^c \quad \text{tel que} \quad \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \begin{pmatrix} p^1/p^3 \\ p^2/p^3 \\ 1 \end{pmatrix}$$

et donc : $\mathbf{P}_c^r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 0 \end{pmatrix}$

Noter que $\mathbf{P}_c^r$ n'est pas inversible .

Si l'origine est dans la rétine, par triangles semblables :

$$\frac{x_r}{F} = \frac{x_c}{(F + z_c)} \qquad => \qquad x_r = \frac{x_c\, F}{(F + z_c)}$$

Equations de perspective:

$$x_r = \frac{x_c\, F}{(F+z_c)} \qquad\qquad y_r = \frac{y_c\, F}{(F+z_c)}$$

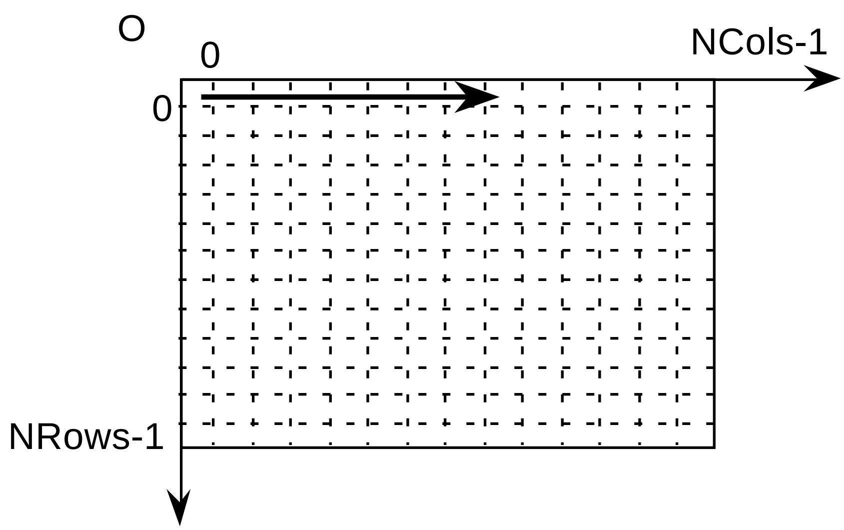
$$z_r = 0$$

et puis : $\mathbf{P}_c^r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 1 \end{pmatrix}$

La transformation Rétine - Image

Les paramètres Intrinsèques de la caméra

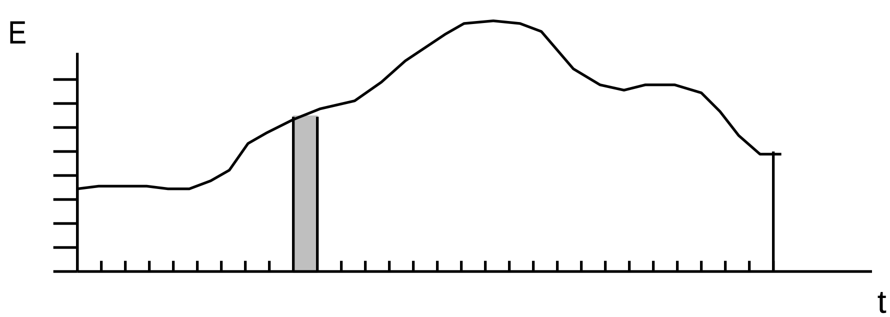
La Trame : L'image est composée des "pixels" (picture elements)



Question : Est-ce-que les pixels sont "carrés"?
Réponse : non.
Le nombre de pixels de l'image dépend de la matérielle.

Par exemple : VGA : 640 x 480

Echantillonage et Numérisation



Les Paramètre Intrinsèque de la Caméra

F : Distance Focale
C_i, C_j: Centre Optique de l'Image (en pixels)
D_i, D_j : Taille Physique d'un Pixel dans la Rétine (pixel/mm)

$i = x_r D_i$ (mm · pixel/mm) $+ C_i$ (pixel)
$j = y_r D_j$ (mm · pixel/mm) $+ C_j$ (pixel)

Transformation entre repère de l'image et repère de la rétine :

$$P^i = \mathbf{C}_r^i \; P^r$$

$$\begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} D_i & 0 & C_i \\ 0 & D_j & C_j \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix}$$

ou bien:

$$\begin{pmatrix} wi \\ wj \\ w \end{pmatrix} = \begin{pmatrix} D_i & 0 & C_i \\ 0 & D_j & C_j \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} wx_r \\ wy_r \\ w \end{pmatrix}$$

La Composition de la Projection Scène - Image

$$\boxed{P^i = \mathbf{C}_r^i \; \mathbf{P}_c^r \; \mathbf{T}_s^c \; P^s = \mathbf{M}_s^i \; P^s}$$

$$\begin{bmatrix} w\,i \\ w\,j \\ w \end{bmatrix} = \mathbf{M}_s^i \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

et donc

$$i = \frac{w\,i}{w} = \frac{M_s^1 \cdot P^s}{M_s^3 \cdot P^s} \qquad j = \frac{w\,j}{w} = \frac{M_s^2 \cdot P^s}{M_s^3 \cdot P^s}$$

ou bien

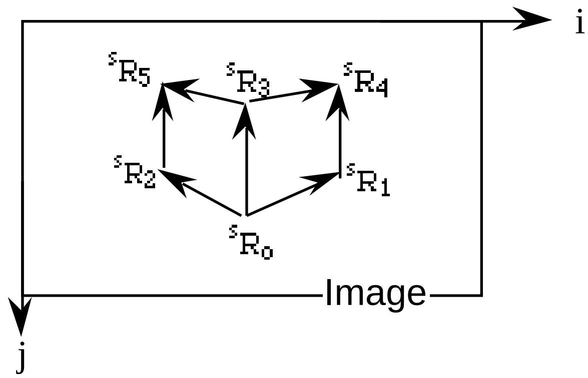
$$i = \frac{w\,i}{w} = \frac{M_{11}\,X_s + M_{12}\,Y_s + M_{13}\,Z_s + M_{14}}{M_{31}\,X_s + M_{32}\,Y_s + M_{33}\,Z_s + M_{34}}$$

$$j = \frac{w\,j}{w} = \frac{M_{21}\,X_s + M_{22}\,Y_s + M_{23}\,Z_s + M_{24}}{M_{31}\,X_s + M_{32}\,Y_s + M_{33}\,Z_s + M_{34}}$$

Calibrage de la Projection Scène - Image

Comment obtenir $\mathbf{M}_s^i$? Par calibrage.

On définit un mire composé de points pour lequels les positions sont connues R_k^s.



La matrice $\mathbf{M}_s^i$ est composé de 3x4=12 coéfficients. Cépendant, $\mathbf{M}_s^i$ est homogène, avec rang 12-1 = 11.

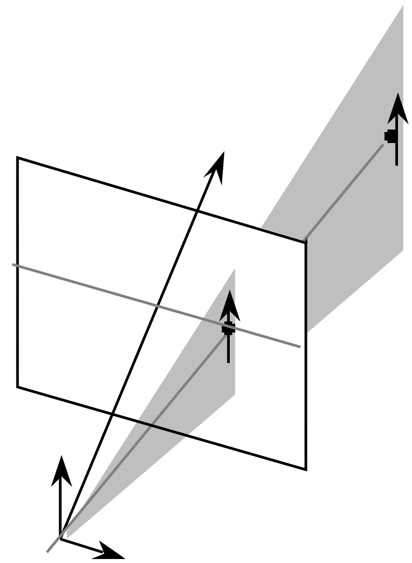
On détermine une correspondance entre les points dans le scènes (R_k^s) et leurs images (P_k^i). Chaque point donne deux équations pour les 11 inconnus.
Il faut au moins 5 $\frac{1}{2}$ correspondances pour les 11 inconnus.

Pour chaque point de calibrage R_k^s et sa projection P_k^s, on peut écrire :

$$i_k = \frac{w_k\, i_k}{w_k} = \frac{M_s^1 \cdot R_k^s}{M_s^3 \cdot R_k^s} \qquad j_k = \frac{w_k\, j_k}{w_k} = \frac{M_s^2 \cdot R_k^s}{M_s^3 \cdot R_k^s}$$

Avec lequel on peut déterminer 2 équations pour les 11 inconnus

$$(M_s^1 \cdot R_k^s) - i_k\,(M_s^3 \cdot R_k^s) = 0 \qquad (M_s^2 \cdot R_k^s) - j_k\,(M_s^3 \cdot R_k^s) = 0$$



L'équation $(M^1_s \cdot R^s_k) - i_k\,(M^3_s \cdot R^s_k) = 0$ est une plan passant par l'origine de la caméra est la colone $i=i_k$.

L'équation $(M^2_s \cdot R^s_k) - j_k\,(M^3_s \cdot R^s_k) = 0$ est une plan passant par l'origine de la caméra est la ligne $j=j_k$.

Avec notation tensorielle, les équations de la calibrage sont explicites.

soit $P^i = \begin{pmatrix} wi \\ wj \\ w \end{pmatrix}$ On écrit : $P^i = \mathbf{M}^i_s\, R^s$

Avec K points de la scène R^S_k et leur correspondance de l'image P^i_k on peut écrire

$$P^i_k = \mathbf{M}^i_s\, R^s_k$$

On ne connait pas P^i_k mais $i\,w = P^1_k/P^3_k$ et $j\,w = P^2_k/P^3_k$. Donc
Pour chaque point k, il y deux equations independants, i=1, 2.

$$\begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} wi \\ wj \\ w \end{pmatrix} \quad \text{donc} \quad \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} p^1/p^3 \\ p^2/p^3 \\ 1 \end{pmatrix}$$

et $P^3_k = \mathbf{M}^3_s\, R^3_k$

$$i = p^1/p^3 = \mathbf{M}^1_s\, R^s_k \,/\, \mathbf{M}^3_s\, R^s_k \qquad i\,\mathbf{M}^3_s\, R^s_k - \mathbf{M}^1_s\, R^s_k = 0$$
$$j = p^2/p^3 = \mathbf{M}^2_s\, R^s_k \,/\, \mathbf{M}^3_s\, R^s_k \qquad j\,\mathbf{M}^3_s\, R^s_k - \mathbf{M}^2_s\, R^s_k = 0$$

Pour chaque k, ceci donne deux équations :

$$\begin{pmatrix} R^1 & R^2 & R^3 & 1 & 0 & 0 & 0 & 0 & -iR^1 & -iR^2 & -iR^3 & -i \\ 0 & 0 & 0 & 0 & R^1 & R^2 & R^3 & 1 & -jR^1 & -jR^2 & -jR^3 & -j \end{pmatrix} \begin{pmatrix} \mathbf{M}_1^1 \\ \mathbf{M}_2^1 \\ \mathbf{M}_3^1 \\ \mathbf{M}_4^1 \\ \mathbf{M}_1^2 \\ \mathbf{M}_2^2 \\ \mathbf{M}_3^2 \\ \mathbf{M}_4^2 \\ \mathbf{M}_1^3 \\ \mathbf{M}_2^3 \\ \mathbf{M}_3^3 \\ \mathbf{M}_4^3 \end{pmatrix} = 0$$

Pour N points (non-coplanaires) on peut écrire un système de 2N équations :

$$\mathbf{A}\ \mathbf{M}_s^i = 0.$$

Le rank de A est 11. Le problème est de minimiser un critère :

$$\mathbf{C} = \| \mathbf{A}\ \mathbf{M}_s^i \|$$

On utilise les multiplieurs de Lagrange afin d'obtenir le $\mathbf{M}_s^i$ qui minimise **C**

Exemple : Soit un cube dont les coins sont détectés à :

$P^L{}_o = (101, 221)$ $P^L{}_1 = (144, 181)$ $P^L{}_2 = (22, 196)$
$P^L{}_3 = (105, 88)$ $P^L{}_4 = (145, 59)$ $P^L{}_5 = (23, 67)$

Par moindre de carré on obtient :

$$\mathbf{M}_s^i = \begin{pmatrix} 55.886873 & -79.292084 & 1.276703 & 101.917630 \\ -22.289319 & -17.878203 & -134.345576 & 221.300658 \\ 0.100734 & 0.038274 & -0.008458 & 1.000000 \end{pmatrix}$$

Dérivation alternative : Le produit croisé.

Dans une notation matricielle, on écrit :

$$\vec{P} \times \mathbf{M}_s^i \vec{R} = 0$$

Le terme $\vec{R}$ est facturé pour obtenir $\vec{P}\ \vec{R} \times \mathbf{M}_s^i = 0$

Ce qui donne : $$\begin{pmatrix} 0 & -wR^s & jwR^s \\ -wR^s & 0 & -iwR^s \\ wR^s & -wR^s & 0 \end{pmatrix} \begin{pmatrix} \mathbf{M}_s^1 \\ \mathbf{M}_s^2 \\ \mathbf{M}_s^3 \end{pmatrix} = 0$$

Parse que $\vec{R}$ et $\mathbf{M}_s^i$, sont les vecteurs, on obtient :

$$\begin{pmatrix} 0 & 0 & 0 & 0 & wX & wY & wZ & w1 & -jwX & -jwY & -jwZ & -jw1 \\ -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 & -iwX & -iwY & -iwZ & -iw1 \\ wX & wY & wZ & w & -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{M}_1^1 \\ \mathbf{M}_2^1 \\ \mathbf{M}_3^1 \\ \mathbf{M}_4^1 \\ \mathbf{M}_1^2 \\ \mathbf{M}_2^2 \\ \mathbf{M}_3^2 \\ \mathbf{M}_4^2 \\ \mathbf{M}_1^3 \\ \mathbf{M}_2^3 \\ \mathbf{M}_3^3 \\ \mathbf{M}_4^3 \end{pmatrix} = 0$$

Dont deux equations sont indépendants.

Homographie entre deux plans

La projection d'un plan vers un autre plan est une transformation projective entre deux plans. Cette transformation s'appele une "homographie.
L'homographie est bijective.

Elle est bijective est facile a estimate et de "recetifié.

$$\boxed{Q^B = \mathbf{H}_A^B \; P^A}$$

En notation "classique".

$$\begin{pmatrix} w\, x_B \\ w\, y_B \\ w \end{pmatrix} = \mathbf{H}_A^B \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix}$$

$$x_B = \frac{w\, x_B}{w} = \frac{m_{11}\, x_A + m_{12}\, y_A + m_{13}}{m_{31}\, x_A + m_{32}\, y_A + m_{33}}$$

$$y_B = \frac{w\, y_B}{w} = \frac{m_{21}\, x_A + m_{22}\, y_A + m_{23}}{m_{31}\, x_A + m_{32}\, y_A + m_{33}}$$

En notation "tensorielle"

$$\boxed{Q^B = \mathbf{H}_A^B \; P^A}$$

$$\begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \mathbf{H}_A^B \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} h_1^1 & h_2^1 & h_3^1 \\ h_1^2 & h_2^2 & h_3^2 \\ h_1^3 & h_2^3 & h_3^3 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}$$

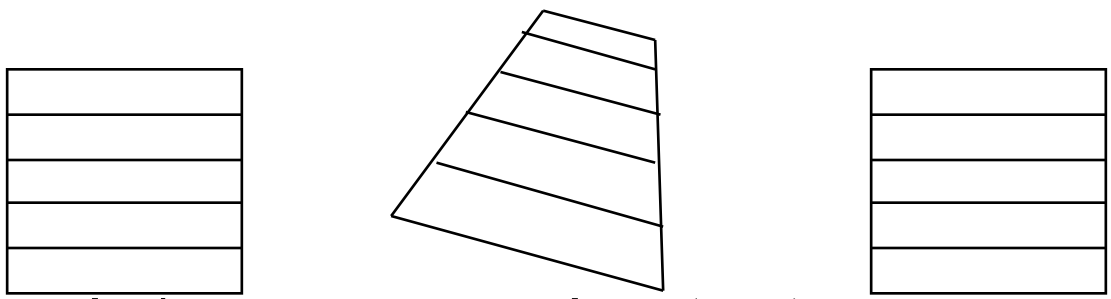
$$x_B = \frac{q^1}{q^3} \qquad y_B = \frac{q^2}{q^3}$$

Rectification de l'Image d'un plan.

La projection d'un plan 2D dans un scène 3D vers le retine 2D d'un caméra est une homographie. Elle est bijective est facile a "corriger".

Ceci a des nombreuse applications. Par exemple, l'image d'un tableau blanc peut être "rectifié" par l'homographie inverse.

$$\mathbf{H}_D^S = (\mathbf{H}_S^D)^{-1}$$



Dans le scène — Dans l'image (source) — L'image Rectifié (destination)

Pour chaque pixel de l'image destination, on calcul son position dans l'image source.

$$Q^S = \mathbf{H}_D^S \; P^D \quad \text{pour S, D = 1, 2, 3.}$$

Ensuite Pour chaque pixel P^D on détermine la valeur de la pixel au position

$$\begin{pmatrix} x_S \\ y_S \\ 1 \end{pmatrix} = \begin{pmatrix} q^1/q^3 \\ q^2/q^3 \\ 1 \end{pmatrix}$$

Ensuite

Pour chaque pixel, P^D :
SetPixel (Destination, P^D)= BiLinear_Interpolate(Source, ($\mathbf{H}_D^S$ P^D));