Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3 - Option IRV — Premier Semestre 2007/2008

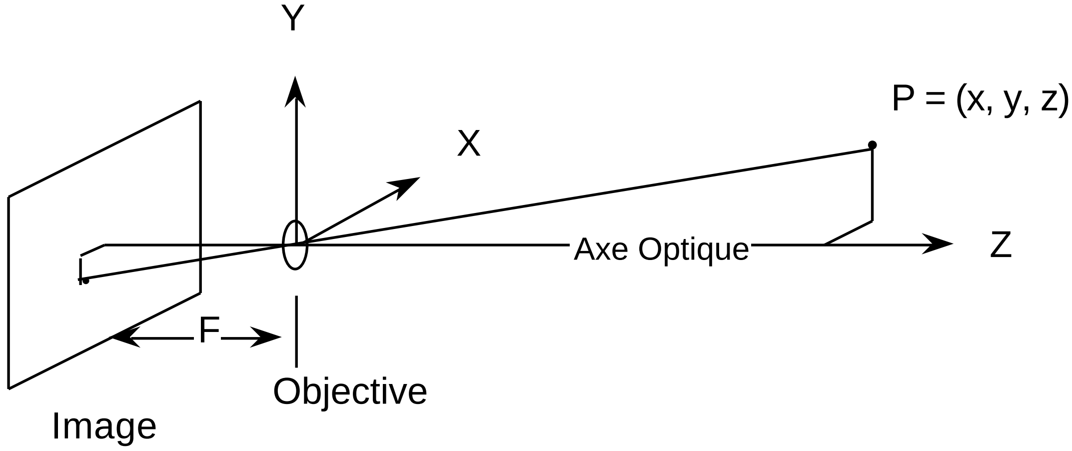
Séance 3 — 12 octobre 2007

Plan de la Séance :

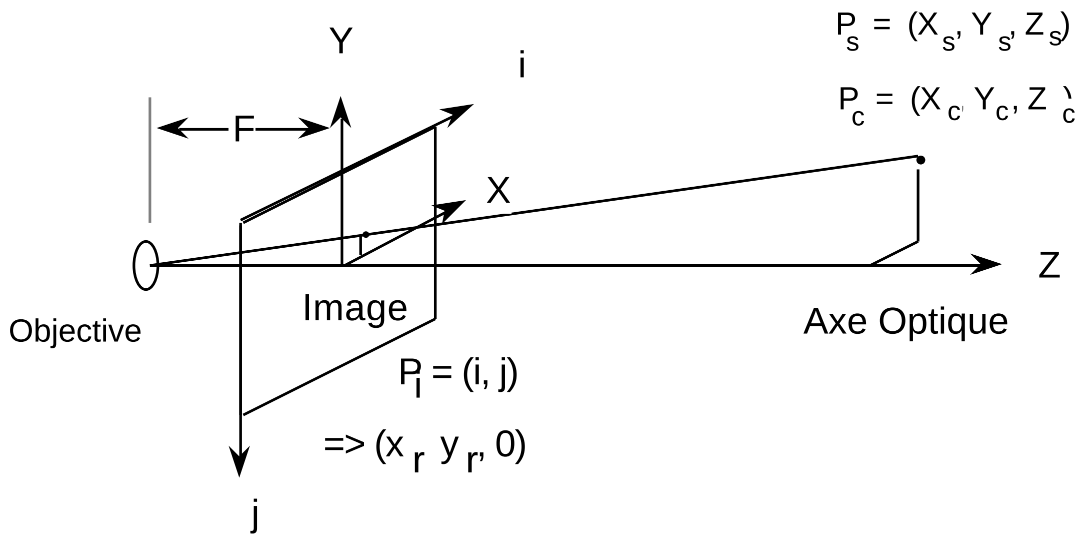
Calibrage du Modèle de la Caméra

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

La Composition de la Projection Scène - Image

$$\boxed{P^i = \mathbf{C}^i_r \ \mathbf{P}^r_c \ \mathbf{T}^c_s P^s = \mathbf{M}^i_s \ P^s}$$

$$\begin{bmatrix} w\ i \\ w\ j \\ w \end{bmatrix} = \mathbf{M}^i_s \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

et donc

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

ou bien

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

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

Projection inverse, du repère <u>image</u> vers le repère <u>scène</u> :

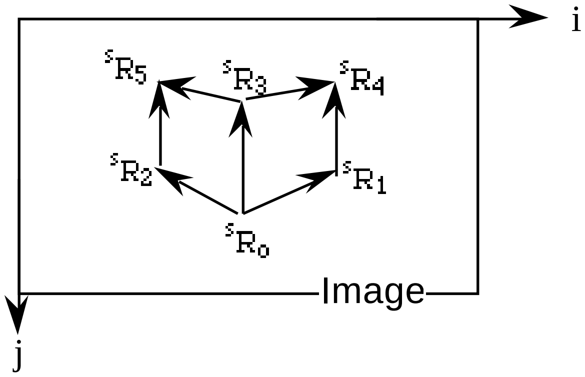
$$P^s = \mathbf{T}^s_c \ \mathbf{P}^c_r \ \mathbf{C}^r_i \ P^i$$

<u>Cependant</u>, l'inversion de la transformation perspective $\mathbf{P}^c_r$ implique la connaissance de la profondeur, z_C, pour chaque pixel.

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.

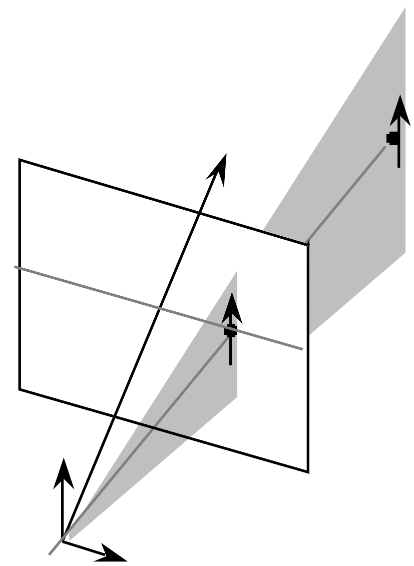
On détermine une correspondance entre les points dans le scènes (R_k^s) et leurs images (P_k^i). Chaque point donne deux équations pour les 11 inconnus.
Il faut au moins $5\frac{1}{2}$ correspondances pour les 11 inconnus.

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"

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

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

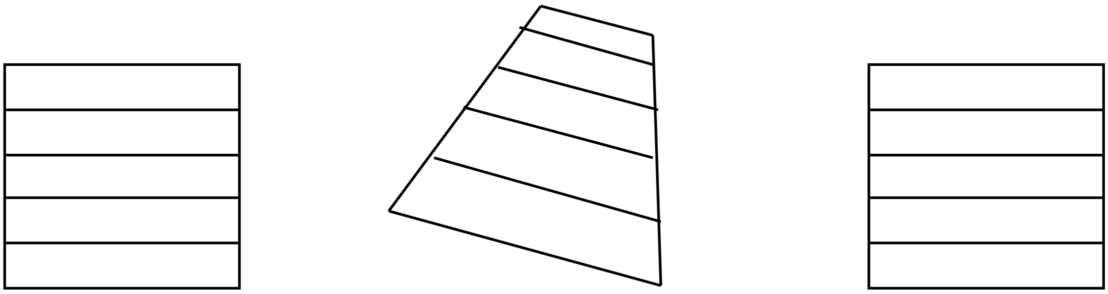
$$x_B = \frac{q^1}{q^3} \qquad y_B = \frac{q^2}{q^3}$$

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

$$\begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} = \begin{pmatrix} q^1/q^3 \\ q^2/q^3 \\ 1 \end{pmatrix}$$

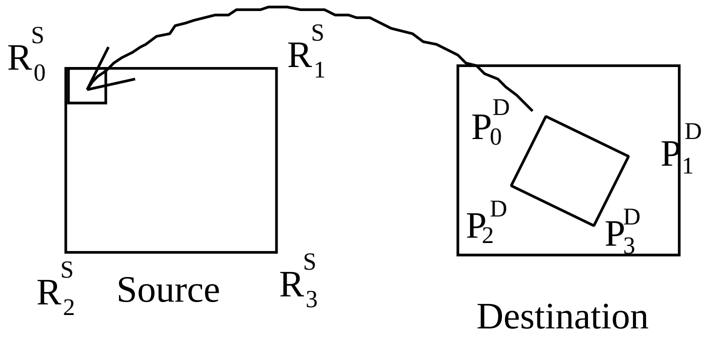
Ensuite

Pour chaque pixel, P^D :

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

Estimation et Rectification par Homographie

La Homographie peut être déterminer par observation des 4 coins d'un carré.



Soit les qautre coins, k=1, 2, 3, 4 de l'image dans repere destination :

$$P_k^D = \begin{pmatrix} p_0^1 & p_1^1 & p_2^1 & p_3^1 \\ p_0^2 & p_1^2 & p_2^2 & p_3^2 \\ p_0^3 & p_1^3 & p_2^3 & p_3^3 \end{pmatrix}$$

On dit qu'ils corresponde au sommets de l'image source

$$R_k^S = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

On note $R_k^S = \mathbf{H}_D^S \; P_k^D$

Le 3 x 3 matrice inconnu, $\mathbf{H}_D^S$ a 9 coefficients.
Parce que $\mathbf{H}_D^S$ est en coordonées homogénes, on peut fixer $\mathbf{H}_3^3 = 1$.
Il nous reste 8 coéfficients à estimer.

Donc :

$$R_k^1 / R_k^3 = \mathbf{H}_D^1 \; P_k^D / \mathbf{H}_D^3 \; P_k^D$$

et

$$R_k^2 / R_k^3 = \mathbf{H}_D^2 \; P_k^D / \mathbf{H}_D^3 \; P_k^D$$

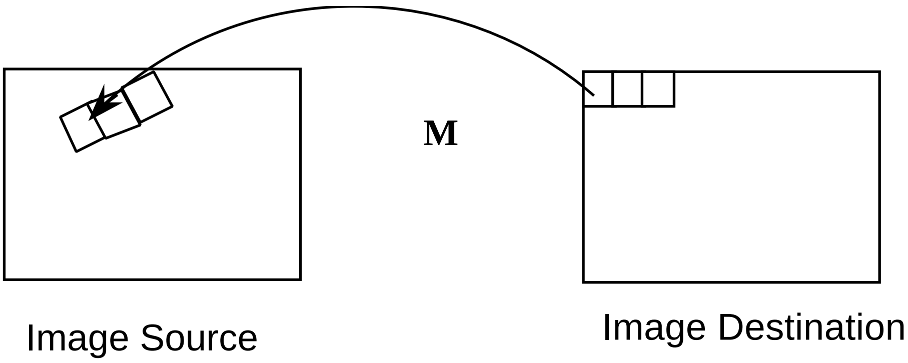
$$R_k^1 \; \mathbf{H}_D^3 \; P_k^D = R_k^3 \; \mathbf{H}_D^1 \; P_k^D$$
$$R_k^2 \; \mathbf{H}_D^3 \; P_k^D = R_k^3 \; \mathbf{H}_D^2 \; P_k^D$$

$$R_k^1 \; \mathbf{H}_D^3 \; P_k^D - R_k^3 \; \mathbf{H}_D^1 \; P_k^D = 0$$
$$R_k^2 \; \mathbf{H}_D^3 \; P_k^D - R_k^3 \; \mathbf{H}_D^2 \; P_k^D = 0$$

Ceci donne deux équations (S=1, 2) pour chaque coin (k=1,2,3,4) , donc 8 equations pour les 8 inconnues du $\mathbf{H}_D^S$. L'equation pour S =3 n'est pas indépendant de S =1 et S=2.

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

Transformations d'images



Pour chaque pixel de l'image de destination (x_d, y_d), on calcule une position (x_s, y_s) dans l'image de la source.

$$P^s = \mathbf{M}_d^s \, P^d$$

ou $$\begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} m_1^1 & m_2^1 & m_3^1 \\ m_1^2 & m_2^2 & m_3^2 \\ m_1^3 & m_2^3 & m_3^3 \end{pmatrix} \begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix}$$

$$x_s = \frac{p^1}{p^3} \qquad y_s = \frac{p^2}{p^3}$$

ou $$\begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} = \begin{pmatrix} p^1/p^3 \\ p^2/p^3 \\ 1 \end{pmatrix}$$

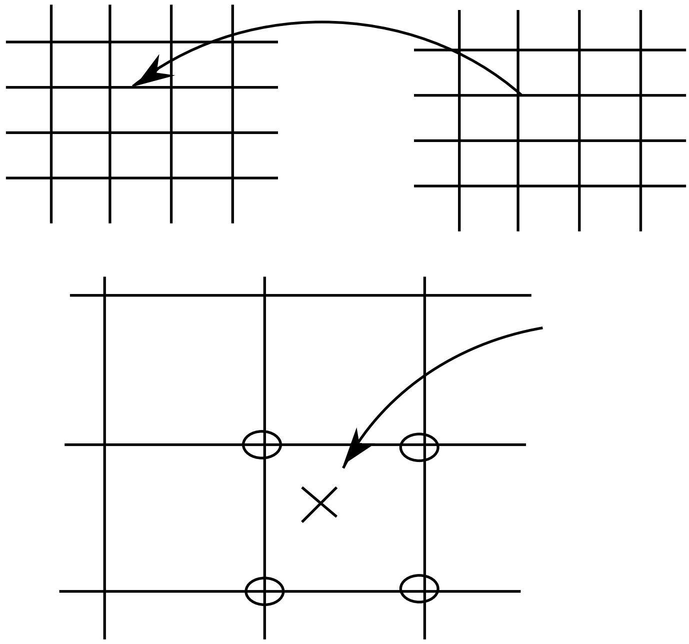
Ensuite la nouvelle valeur de
pixel destination, P^2, est calculé en fonction du voisinage de la position source.

MAIS, $P^s = \begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix}$ n'est pas des entier!

Quelle valeur faut-il prendre pour les pixels?

Pour chaque pixel du destination, (x_d, y_d) on calcul le position du source, (x_s, y_s). Ensuite, on détermine une valeur par interpolation avec les pixels voisins.

Pour chaque pixel du destination, (x_d, y_d) on calcul le position du source, (x_s, y_s). Ensuite, on détermine une valeur par interpolation avec les pixels voisins.



Il existe plusieurs fonctions d'interpolation.

Ordre zéro :	Plus proche voisin.
Ordre unité :	intérpolation linéaire et "bi-linéaire"
Ordre trois :	Spline cubic.