

Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3 - Option IRV

Premier Semestre 2009/2010

Séance 3

12 octobre 2009

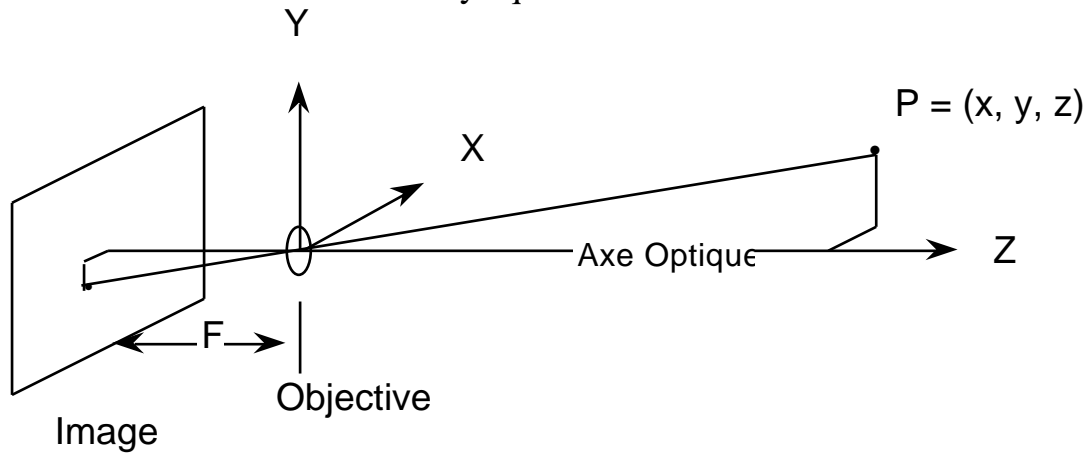
Plan de la Séance :

Calibration de Caméras et Transformations d'Images

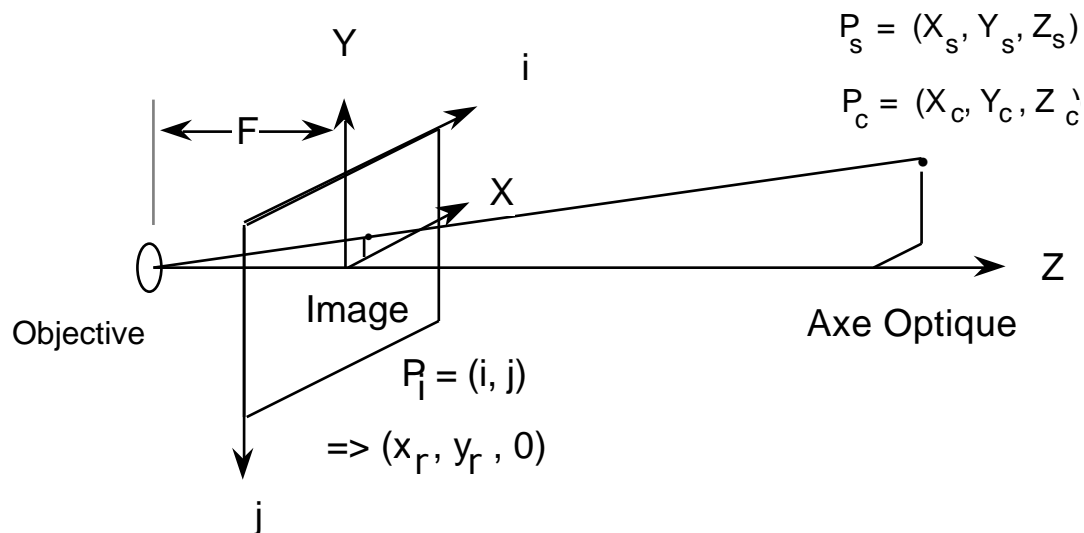
Modèle de la Caméra (Rappel)	2
Les Repères.....	2
La Composition de la Projection Scène - Image.....	3
Calibrage de la Projection Scène - Image.....	4
Dérivation.....	4
Dérivation alternative : Le produit croisé.....	7
Transformation des Images.....	8
Transformations d'échelle, rotation et translation.....	8
Homographie entre deux plans.....	9
Rectification de l'Image d'un plan.....	10
Estimation et Rectification par Homographie.....	11
Transformations d'images	12
Interpolation Linéaire.....	14
Interpolation Bi-linéaire.....	15

Modèle de la Caméra (Rappel)

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

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

$$\begin{pmatrix} w i \\ w j \\ w \end{pmatrix} = \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \mathbf{M}_s^i \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix}$$

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

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

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

Projection inverse, du repère image vers le repère scène :

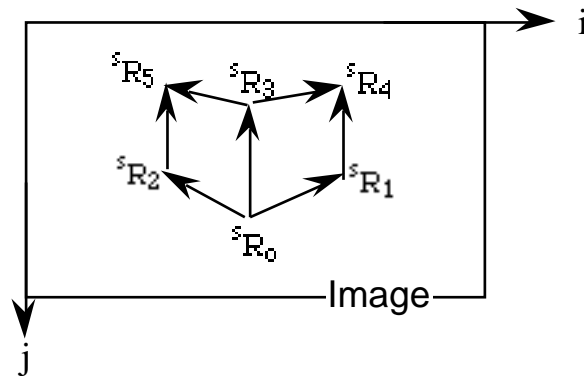
$$\mathbf{P}^s = \mathbf{T}_c^s \mathbf{P}_r^c \mathbf{C}_i^r \mathbf{P}^i$$

Cependant, l'inversion de la transformation perspective \mathbf{P}_r^c implique la connaissance de la profondeur, z_c , pour chaque pixel.

Calibrage de la Projection Scène - Image

Comment obtenir M_s^i ? Par calibrage.

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



La matrice M_s^i est composée de $3 \times 4 = 12$ coefficients. Cependant, M_s^i est homogène, avec rang $12 - 1 = 11$.

Dérivation

On détermine une correspondance entre les points dans le scène (R_k^s) et leurs images (P_k^i). Chaque point donne deux équations pour les 11 inconnus.

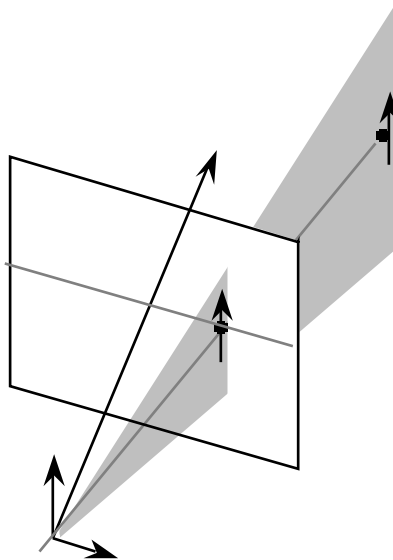
Il faut au moins $5 \frac{1}{2}$ correspondances pour les 11 inconnus.

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

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

Avec lequel on peut déterminer 2 équations pour les 11 inconnus

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



L'équation $(M_s^1 \cdot R_k^s) - i_k (M_s^3 \cdot R_k^s) = 0$ est une plan passant par l'origine de la caméra est la colone $i=i_k$.

L'équation $(M_s^2 \cdot R_k^s) - j_k (M_s^3 \cdot R_k^s) = 0$ 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 $P^i = \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} w_i \\ w_j \\ w \end{pmatrix}$ On écrit : $P^i = M_s^i R^s$

Avec K points de la scène R_k^s et leur correspondance de l'image P_k^i on peut écrire

$$P_k^i = M_s^i R_k^s$$

On ne connaît pas P_k^i mais $i w = P_k^1/P_k^3$ et $j w = P_k^2/P_k^3$. Donc Pour chaque point k, il y deux equations independants, $i=1, 2$.

$$\frac{p^1}{p^3} = \frac{w_i}{w} \quad \text{donc} \quad \frac{i}{1} = \frac{p^1/p^3}{1}$$

et $P_k^3 = M_s^3 R_k^3$

$$\begin{aligned} i = p^1/p^3 &= M_s^1 R_k^s / M_s^3 R_k^s & i M_s^3 R_k^s - M_s^1 R_k^s &= 0 \\ j = p^2/p^3 &= M_s^2 R_k^s / M_s^3 R_k^s & j M_s^3 R_k^s - M_s^2 R_k^s &= 0 \end{aligned}$$

Pour chaque k, ceci donne deux équations :

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

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 :

$$\mathbf{C} = \| \mathbf{A} \mathbf{M}_s^i \|^2$$

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 :

$$\begin{matrix}
 P_o^L = (101, 221) & P_1^L = (144, 181) & P_2^L = (22, 196) \\
 P_3^L = (105, 88) & P_4^L = (145, 59) & P_5^L = (23, 67)
 \end{matrix}$$

Par moindre de carré on obtient :

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

Dérivation alternative : Le produit croisé.

Dans une notation matricielle, on écrit :

$$P \times M_s^i R = 0$$

Le terme R est facturé pour obtenir $P R \times M_s^i = 0$

Ce qui donne :

$$\begin{matrix} 0 & -wR^s & jwR^s & M_s^1 \\ -wR^s & 0 & -iwR^s & M_s^2 \\ wR^s & -wR^s & 0 & M_s^3 \end{matrix} = 0$$

Parse que R et M_s^i , sont les vecteurs, on obtient :

$$\begin{matrix} 0 & 0 & 0 & 0 & wX & wY & wZ & w1 & -jwX & -jwY & -jwZ & -jw1 \\ -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 & -iwX & -iwY & -iwZ & -iw1 \\ wX & wY & wZ & w & -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 \end{matrix} \begin{matrix} M_1^1 \\ M_2^1 \\ M_3^1 \\ M_4^1 \\ M_1^2 \\ M_2^2 \\ M_3^2 \\ M_4^2 \\ M_1^3 \\ M_2^3 \\ M_3^3 \\ M_4^3 \end{matrix} = 0$$

Dont deux equations sont indépendants.

Transformation des Images

Transformations d'échelle, rotation et translation

Une transformation de similitude d'une image est définie par une rotation, une translation, et un changement de taille des axes.

Soit un changement d'échelle s_x et s_y des axes x_1 et y_1 (repère source) suivi d'une rotation d'angle θ dans le plan de l'image source, suivi d'une translation t_x , t_y s'exprimes dans le repère de la destination.

Ces paramètres donne une transformation $(s_x, s_y, \theta, t_x, t_y)$ de

- 1) Un changement d'échelle des axes, puis
- 2) Une rotation des axes, puis
- 3) Une translation.

$$\begin{array}{rcccl} x_2 & & s_x \cos(\theta) & s_y \sin(\theta) & t_x & x_1 \\ y_2 & = & -s_x \sin(\theta) & s_y \cos(\theta) & t_y & y_1 \\ 1 & & 0 & 0 & 1 & 1 \end{array}$$

ou bien

$$\begin{array}{l} x_2 = s_x \cos(\theta) x_1 + s_y \sin(\theta) y_1 + t_x, \\ y_2 = -s_x \sin(\theta) x_1 + s_y \cos(\theta) y_1 + t_y \\ 1 = 1 \end{array}$$

Et à quoi sert le dernier ligne de la matrice ?

Homographie entre deux plans

La projection d'un plan vers un autre plan est une transformation projective entre deux plans. Cette transformation s'appelle 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".

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

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

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

En notation "tensorielle"

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

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

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

Rectification de l'Image d'un plan.

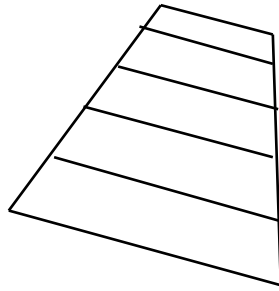
La projection d'un plan 2D dans un scène 3D vers le retine 2D d'un caméra est une homographie. Elle est bijective est facile a "corriger".

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

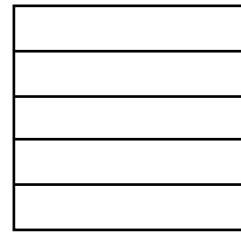
$$\mathbf{H}_D^S = (\mathbf{H}_S^D)^{-1}$$



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

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

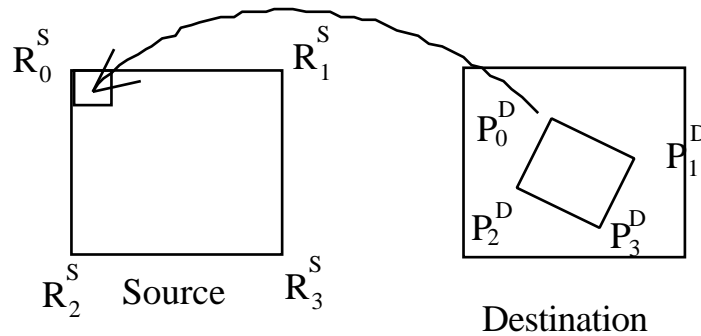
Ensuite

Pour chaque pixel, P^D :

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

Estimation et Rectification par Homographie

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



Soit les quatre 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 $R_k^S = H_D^S P_k^D$

Le 3×3 matrice inconnu, H_D^S a 9 coefficients.

Parce que H_D^S est en coordonnées homogènes, on peut fixer $H_3^S = 1$.

Il nous reste 8 coefficients à estimer.

Donc :

$$\begin{aligned} R_k^1 / R_k^3 &= H_D^1 P_k^D / H_D^3 P_k^D \\ \text{et } R_k^2 / R_k^3 &= H_D^2 P_k^D / H_D^3 P_k^D \end{aligned}$$

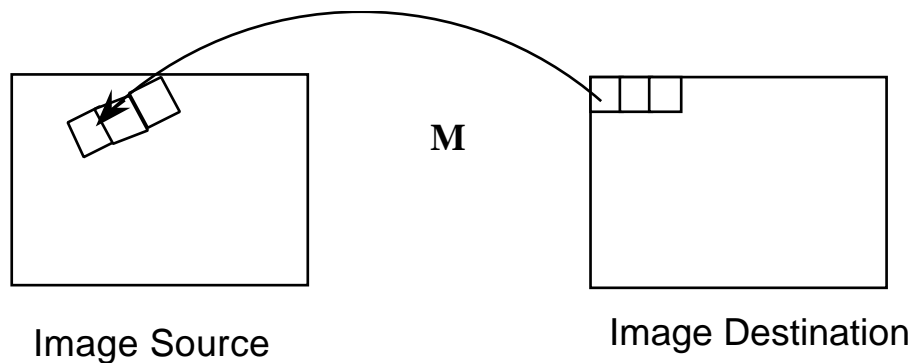
$$\begin{aligned} R_k^1 H_D^3 P_k^D &= R_k^3 H_D^1 P_k^D \\ R_k^2 H_D^3 P_k^D &= R_k^3 H_D^2 P_k^D \end{aligned}$$

$$\begin{aligned} R_k^1 H_D^3 P_k^D - R_k^3 H_D^1 P_k^D &= 0 \\ R_k^2 H_D^3 P_k^D - R_k^3 H_D^2 P_k^D &= 0 \end{aligned}$$

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

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

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.

$$P^s = M_d^s P^d$$

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

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

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

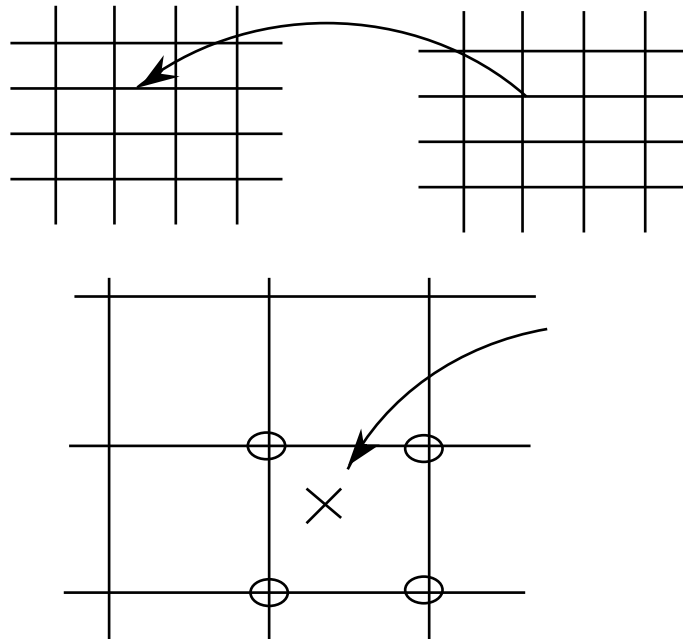
Ensuite la nouvelle valeur de pixel destination, P^2 , est calculé en fonction du voisinage de la position source.

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.

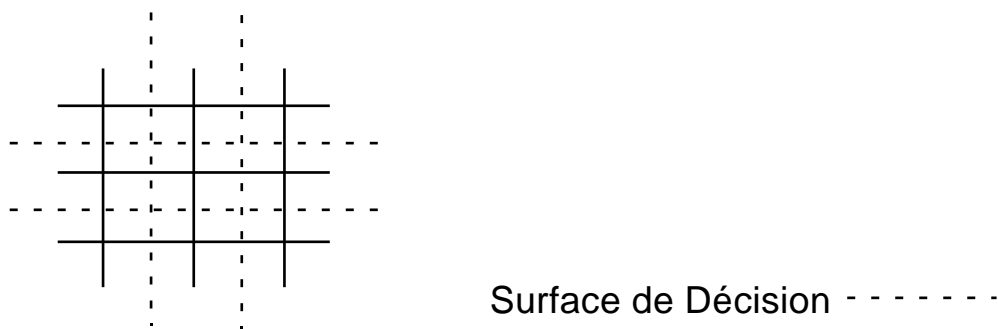


Il existe plusieurs fonctions d'interpolation.

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

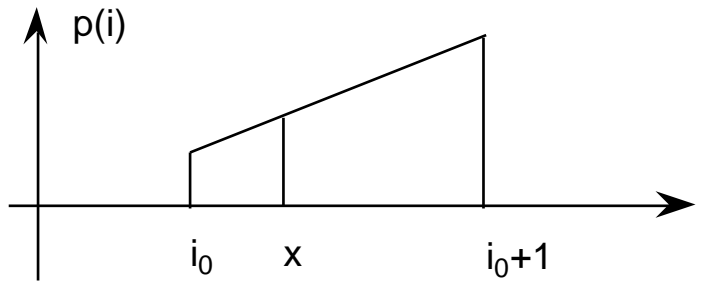
Interpolation d'ordre zéro

Pour les images Binaire, on peut faire que l'ordre zéro.
 La valeur de $p(i_2, j_2)$ déterminer par arrondis de $p(i_1, j_1)$.



Interpolation Linéaire

Interpolation Linéaire en 1-D. soit $i_0 \leq x \leq i_0+1$

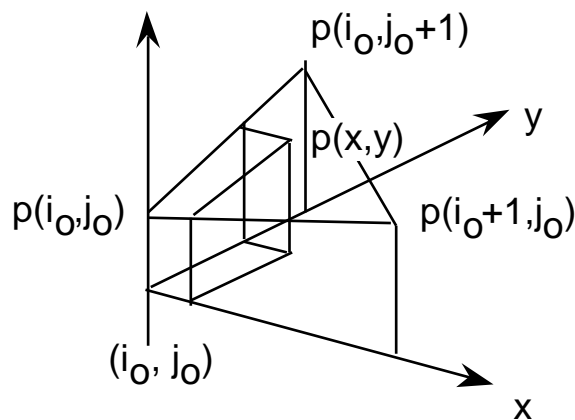


A partir de l'origine : $p(x) = p(0) + m_x x$

A partir de deux points i_0 et i_0+1 :

pende : $m_x = \frac{P}{x} = \frac{p(i_0+1) - p(i_0)}{i_0+1 - i_0}$

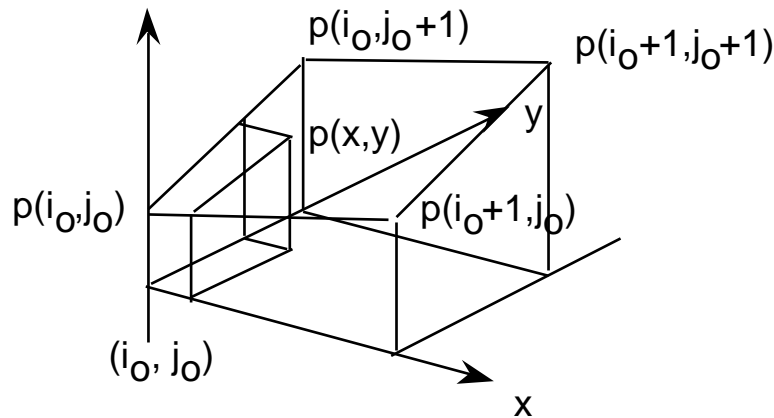
$$p(x) = (x - i_0) m_x + p(i_0)$$

Interpolation Linéaire en 2D

$$m_x = \frac{P}{x} = \frac{p(i_0+1, j_0) - p(i_0, j_0)}{i_0+1 - i_0}$$

$$m_y = \frac{P}{y} = \frac{p(i_0, j_0+1) - p(i_0, j_0)}{j_0+1 - j_0}$$

donc $p(x, y) = m_x \cdot (x - i_0) + m_y \cdot (y - j_0) + p(i_0, j_0)$

Interpolation Bi-lineaire

Forme Bilinéaire : Hyperbolic Paraboloïde

$$p(x, y) = a x + b y + c x y + d.$$

Une interpolation linéaire en "y" de deux interpolations linéaire en "x".

Dérivation :

$$p(x, 0) = p(0, 0) + x \cdot (p(1, 0) - p(0, 0))$$

$$p(x, 1) = p(0, 1) + x \cdot (p(1, 1) - p(0, 1))$$

$$p(x, y) = p(x, 0) + y \cdot (p(x, 1) - p(x, 0))$$

$$= p(0, 0) + x \cdot (p(1, 0) - p(0, 0))$$

$$+ y \cdot (p(0, 1) + x \cdot (p(1, 1) - p(0, 1)))$$

$$- y \cdot (p(0, 0) + x \cdot (p(1, 0) - p(0, 0)))$$

$$= p(0, 0) + x \cdot (p(1, 0) - p(0, 0)) + y \cdot (p(0, 1) - p(0, 0))$$

$$+ x \cdot y \cdot (p(1, 1) - p(0, 1) - p(1, 0) + p(0, 0))$$

Pour le point i_0, j_0 , remplace : 0 $i_0, 1$ i_0, x $(x - i_0), y$ $(y - i_0)$

$$a \quad m_x = \frac{P}{x} = p(i_0+1, j_0) - p(i_0, j_0)$$

$$b \quad m_y = \frac{P}{y} = p(i_0, j_0+1) - p(i_0, j_0)$$

$$c \quad m_{xy} = p(i_0+1, j_0) + p(i_0, j_0+1) - p(i_0, j_0) - p(i_0+1, j_0+1)$$

$$d = p(i_0, j_0)$$

$$p(x, y) = a \cdot (x - i_0) + b \cdot (y - j_0) + c \cdot (x - i_0) \cdot (y - j_0) + p(i_0, j_0)$$