

Introduction au traitement d'images

Enseignement intégré

ITEC2 | Informatique 2A | 2025-2026

Chapitre 7 : Transformations spatiales

Rémi Giraud

remi.giraud@enseirb-matmeca.fr

<https://remi-giraud.enseirb-matmeca.fr/>

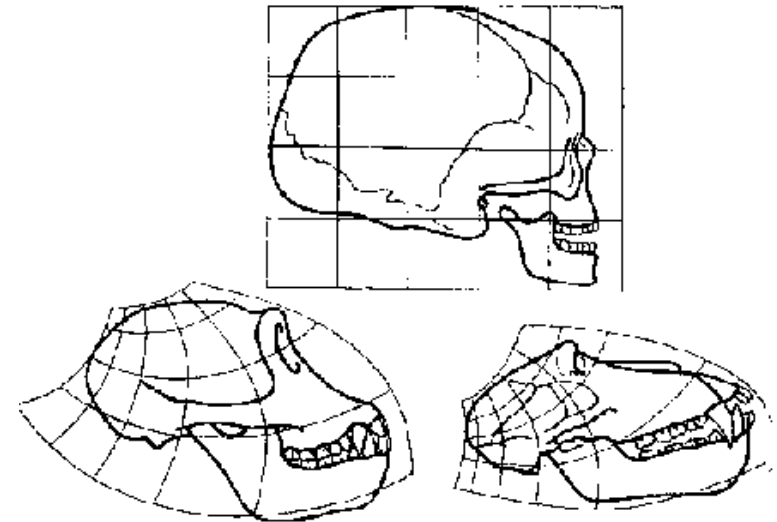
- **Introduction**
- **Formation / Acquisition**
- **Image couleur**
 - Format/Affichage/Synthèse
 - Espaces couleur caractéristiques (YCbCr)
 - Applications : compression, esquisse
- **Traitements**
 - Filtrage linéaire / non linéaire
 - Applications : débruitage, anonymisation
 - Détection de contours
 - Applications : réhaussement de contraste
- **Transformée de Fourier**
 - Application : recouvrement fréquentiel
- **Compression d'images**
 - Application : algorithme JPEG
- **Transformation spatiales**

PRINCIPE

Position du problème

$$I_s(x, y) = g(I_e(f(x, y)))$$

- $g(.)$ transformation d'intensité
- $f(.)$ transformation spatiale
 - Rigide, Affine, Projective etc.

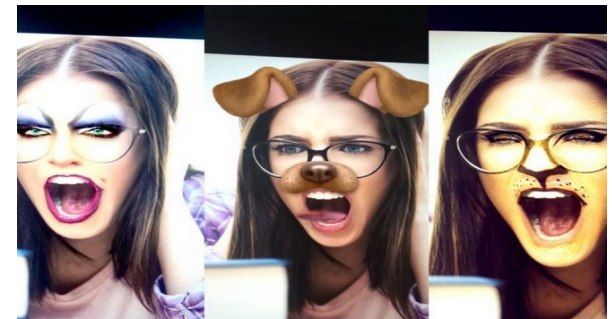


Skulls of a human, a chimpanzee and a baboon and transformations between them



Applications

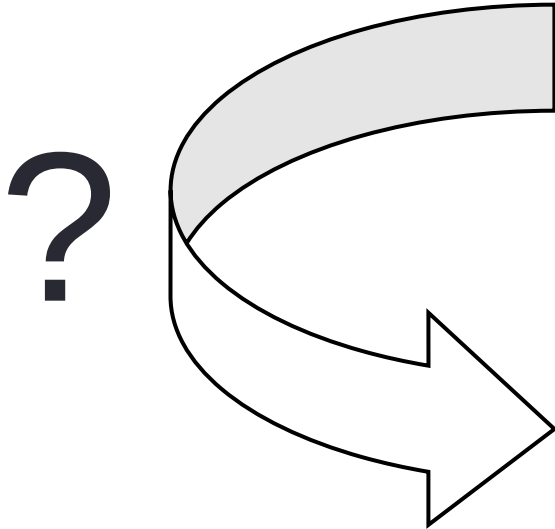
- Mise en correspondance d'images prises en des instants différents (panoramas, mosaïques, recalages, ...)
- Corriger des images suite à des distorsions géométriques (optique)
- Corriger l'orientation de la caméra
- Morphing d'images ou autres effets spéciaux



Transformation

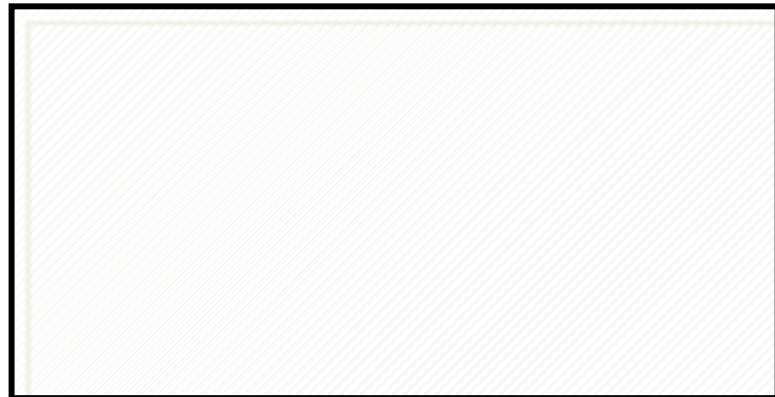
Exemple : Conversion de format

- Comment déformer l'image ?



4:3

$$\frac{w}{h}$$



16:9

$$\frac{4w}{3h}$$

Transformation

Exemple : Conversion de format

- « Zero-padding »



25 % de l'écran non utilisé

- Crop



25 % d'image perdue

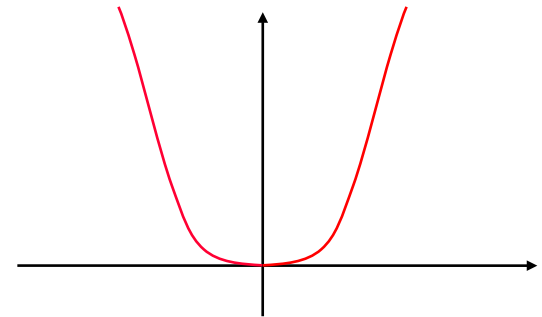
Exemple : Conversion de format

- Contraction horizontale



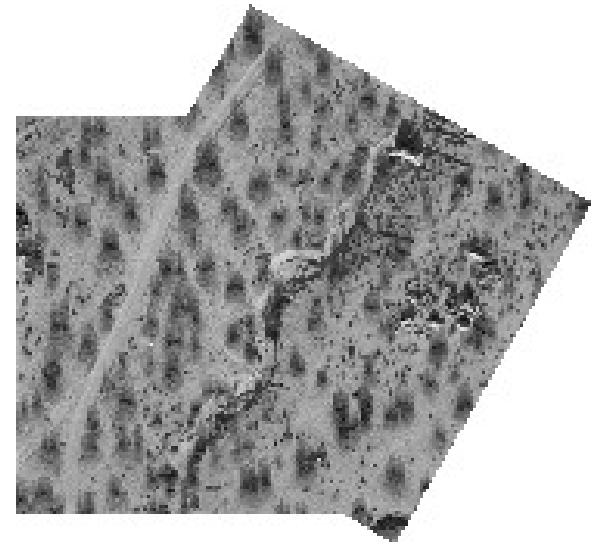
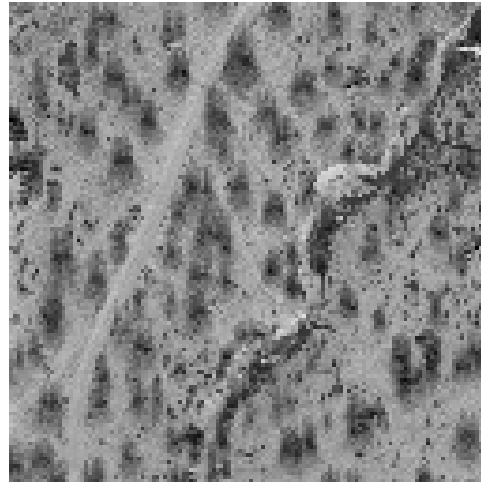
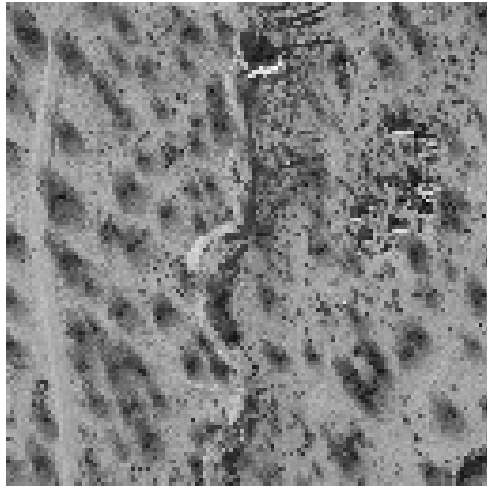
Transformation non linéaire
Plus prononcée aux bords

$$x' = ax$$



Transformation

Recomposition de large scène à partir de morceaux

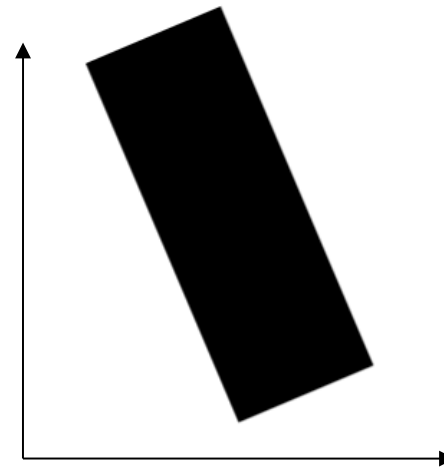
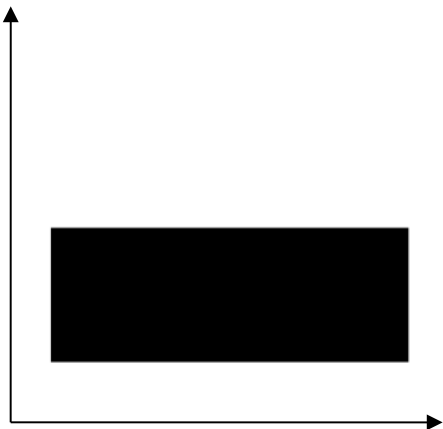


TRANSFORMATION DANS LE PLAN

Rotation, translation, similarité

$$\vec{p}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \xrightarrow{f(\cdot)} \vec{p}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\vec{p}_2 = \vec{t} + SR\vec{p}_1 \quad R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad \vec{s} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$$



Transformations spatiales

D'un point vue numérique

- Pas de bijection entre l'espace de départ et d'arrivée

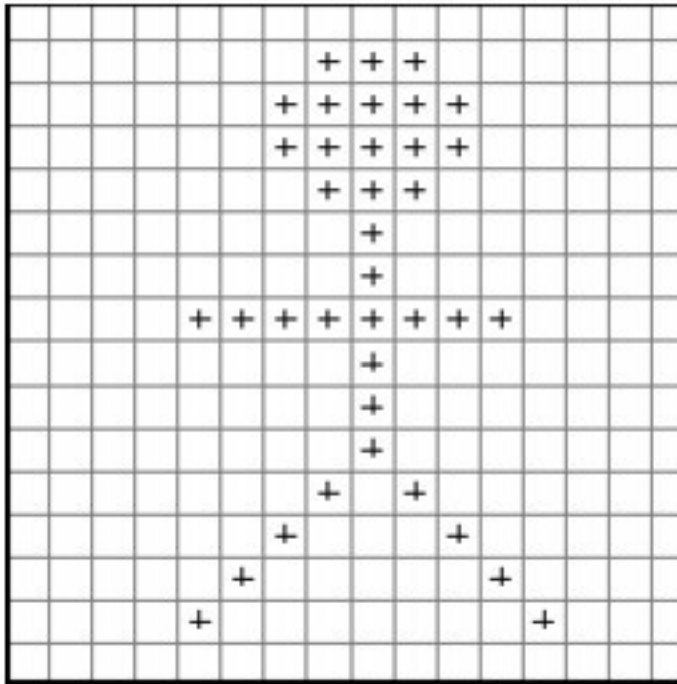


Image de départ

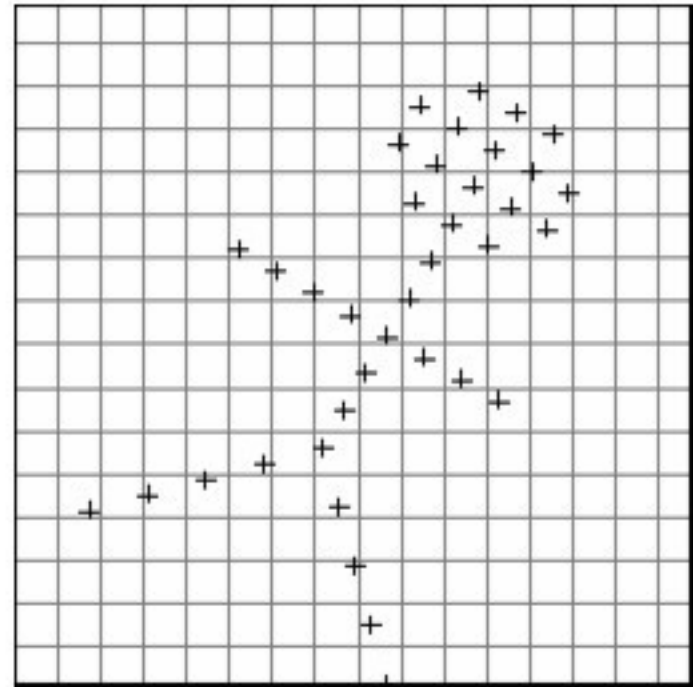
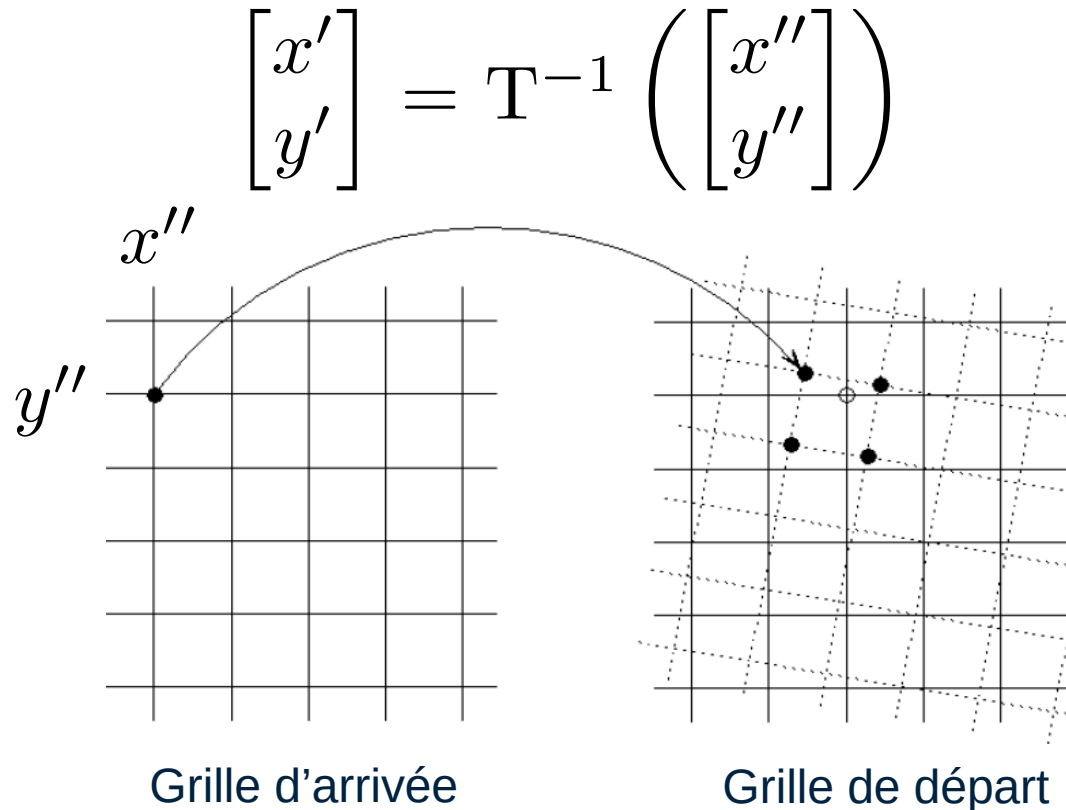


Image d'arrivée

D'un point vue numérique

- Pas de bijection entre l'espace de départ et d'arrivée
→ Application de la transformation inverse

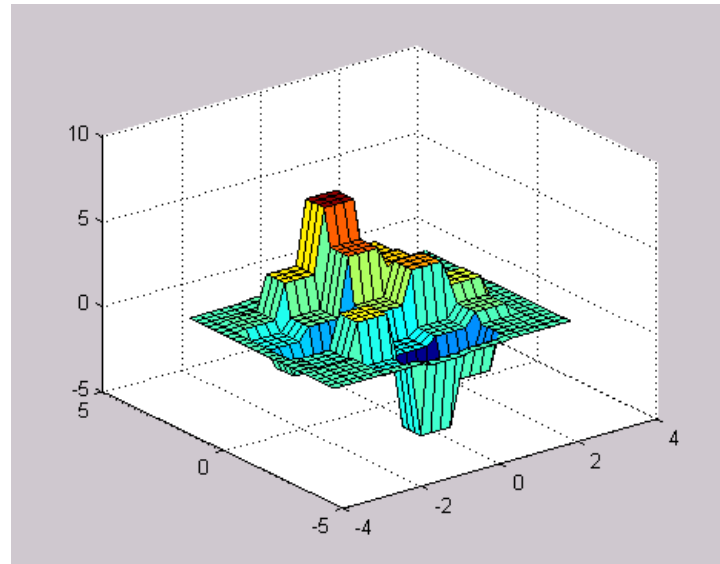
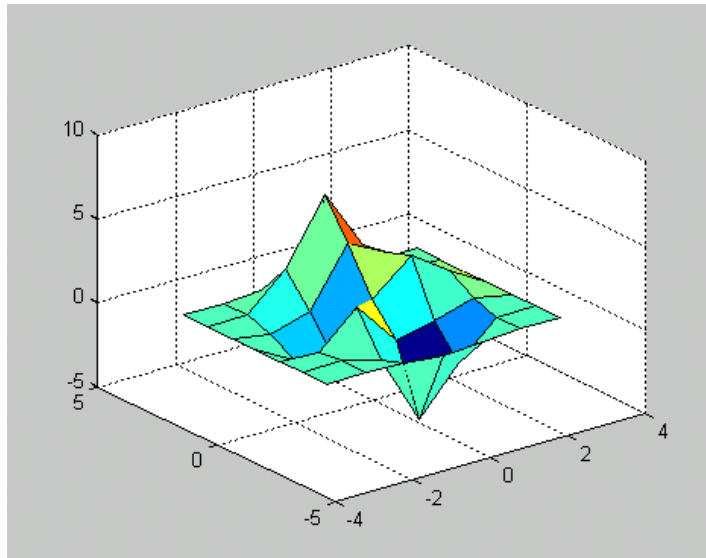
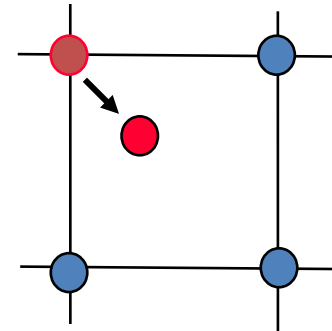


Méthode d'interpolation

- **Plus proche voisin** : sélection du pixel le plus proche

$$\hat{I}(\bullet) = I(\bullet)$$

$$\hat{I}(x', y') = I(\text{round}(x'), \text{round}(y'))$$



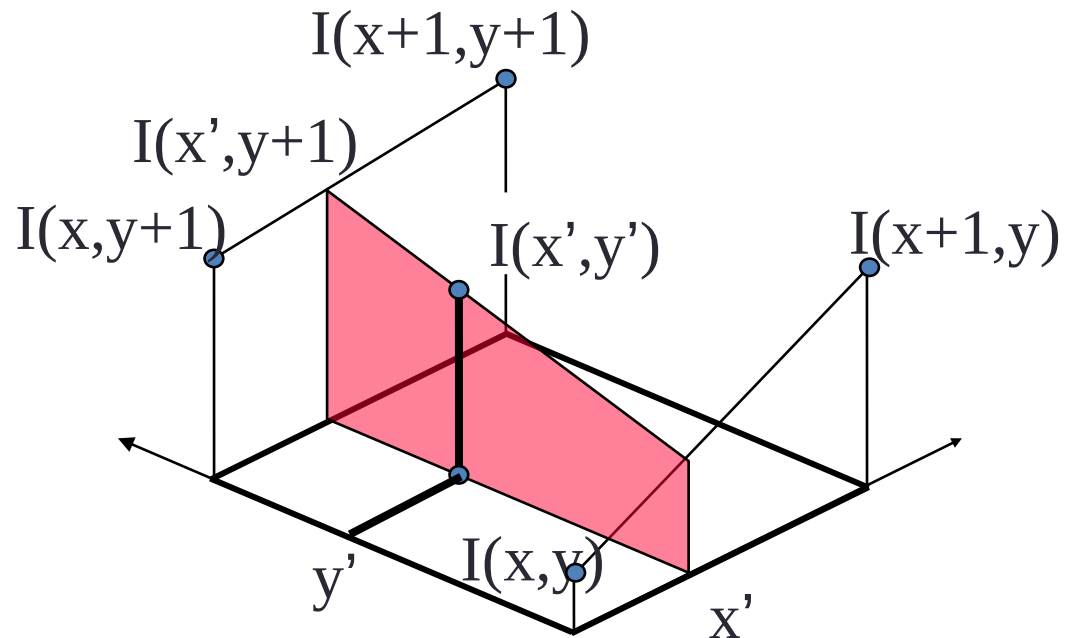
Méthode d'interpolation

- **Bilinéaire** : moyenne pondérée des 4 pixels les plus proches

$$I(x', y') = b(1 - a)I(y + 1, x) + baI(y + 1, x + 1) \\ + (1 - b)(1 - a)I(y, x) + a(1 - b)I(y, x + 1)$$

$$\begin{cases} a = x' - x \\ b = y' - y \end{cases}$$

$$\begin{cases} x = \text{floor}(x') \\ y = \text{floor}(y') \end{cases}$$



Plus proche voisin VS Interpolation

- Interpolation plus naturelle sur une image

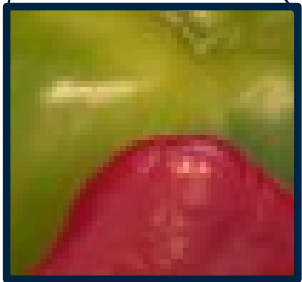
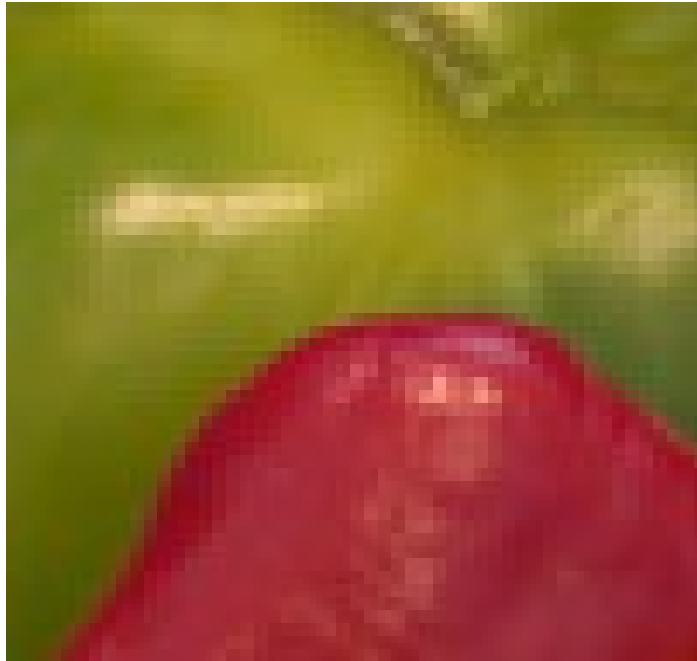
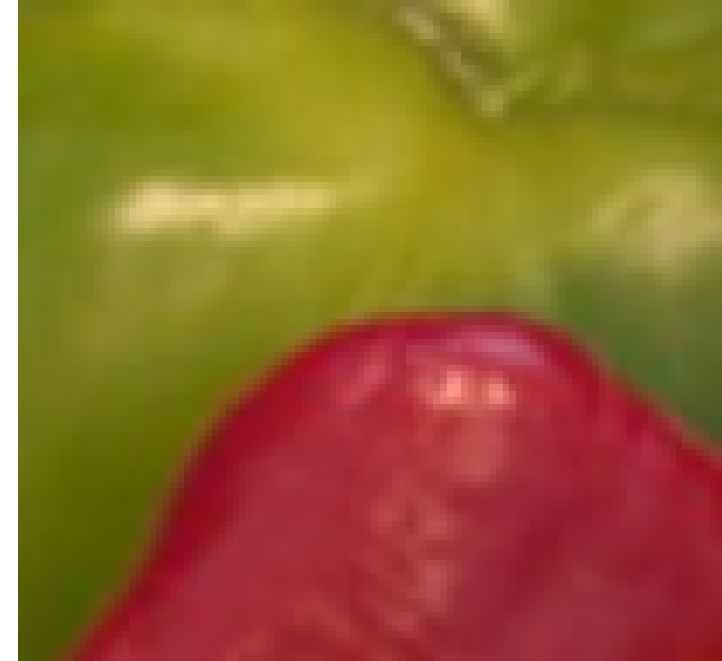


Image initiale



Agrandissement x2 avec
plus proche voisin



Agrandissement x2 avec
interpolation bilinéaire

Plus proche voisin VS Interpolation

- Interpolation plus naturelle sur une image
- Mais pas adapté pour les matrices type carte de labels



Image initiale



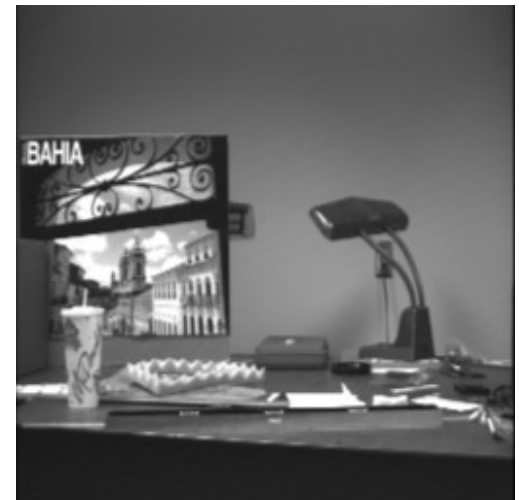
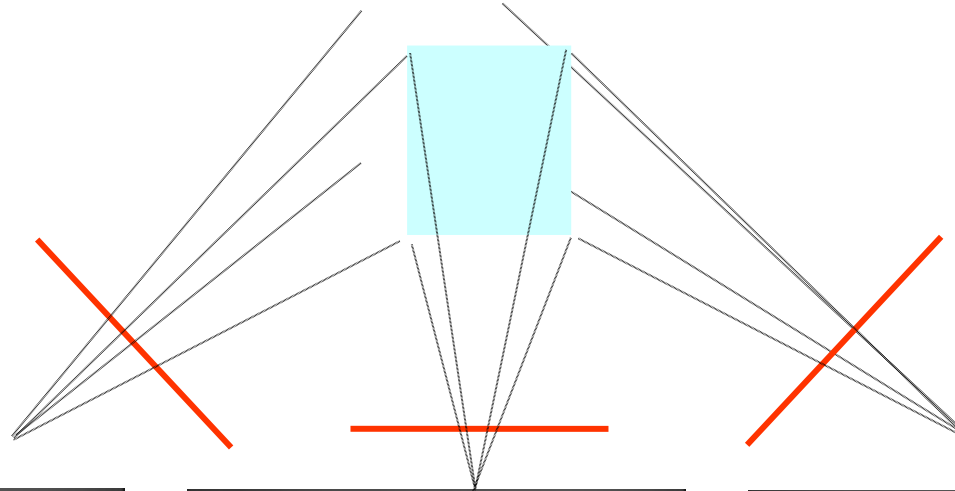
Agrandissement x2 avec
plus proche voisin



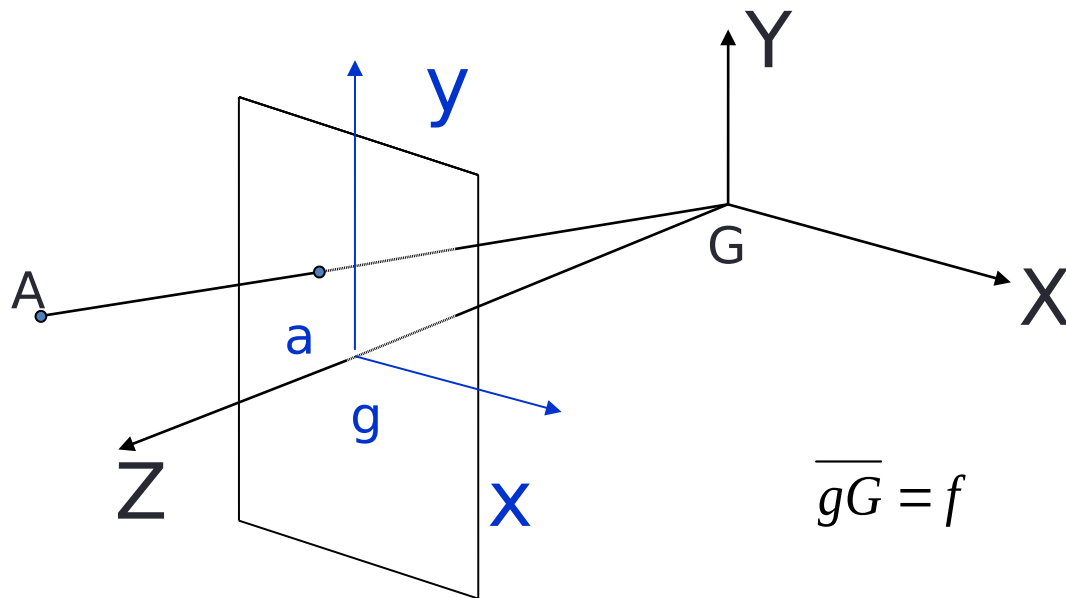
Agrandissement x2 avec
interpolation bilinéaire

ANNEXES TRANSFORMATION COMPATIBLE 3D

Plusieurs prises de vue de la même scène



Modèle 3D



$$\overline{gG} = f$$

$$\left\{ \begin{array}{l} x_A = f \frac{X_A}{Z_A} \\ y_A = f \frac{Y_A}{Z_A} \end{array} \right\}$$

$$\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = \begin{pmatrix} T_X \\ T_Y \\ T_Z \end{pmatrix} + R \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}$$

Approximation par rotation

$$R_X R_Y R_Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_X & -\sin \theta_X \\ 0 & \sin \theta_X & \cos \theta_X \end{pmatrix} \begin{pmatrix} \cos \theta_Y & 0 & -\sin \theta_Y \\ 0 & 1 & 0 \\ \sin \theta_Y & 0 & \cos \theta_Y \end{pmatrix} \begin{pmatrix} \cos \theta_Z & -\sin \theta_Z & 0 \\ \sin \theta_Z & \cos \theta_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

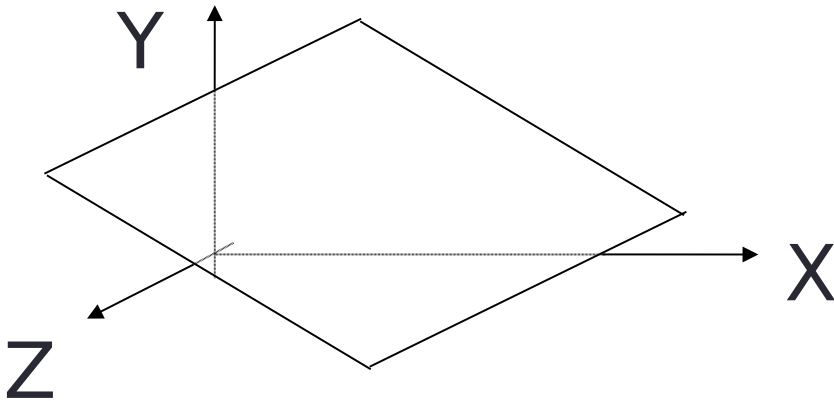
$$R = \begin{pmatrix} 1 & -\theta_Z & -\theta_Y \\ \theta_Z & 1 & -\theta_X \\ \theta_Y & \theta_X & 1 \end{pmatrix}$$

Modèle projectif

$$\left\{ \begin{array}{l} x_2 = f \frac{X_2}{Z_2} \\ y_2 = f \frac{Y_2}{Z_2} \end{array} \right\}$$

$$\left\{ \begin{array}{l} x_2 = M_x[T, R, P_1] \\ y_2 = M_y[T, R, P_1] \end{array} \right\}$$

Modèle homographique



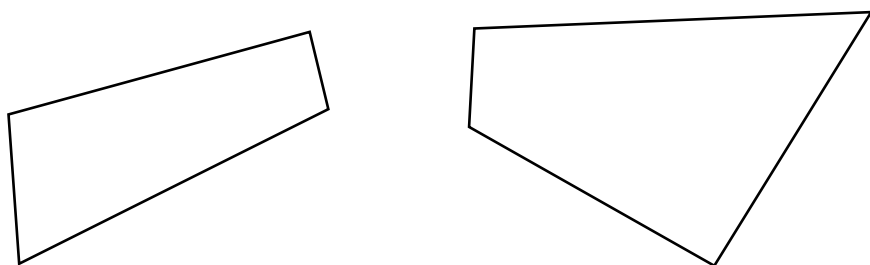
$$Z = aX + bY + Z_o$$



Garantir : une droite reste une droite

Modèle homographique

- Modèle à 8 paramètres



$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{a_1 + a_2 x_1 + a_3 y_1}{1 + a_7 x_1 + a_8 y_1} \\ \frac{a_4 + a_5 x_1 + a_6 y_1}{1 + a_7 x_1 + a_8 y_1} \end{bmatrix}$$



Mise en correspondance entre les couples de points



Estimation

$$x'_k = \frac{d_0 x_k + d_1 y_k + d_2}{d_6 x_k + d_7 y_k + 1}, \quad y'_k = \frac{d_3 x_k + d_4 y_k + d_5}{d_6 x_k + d_7 y_k + 1}.$$

$$\underbrace{\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0 x'_0 & -y_0 x'_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x'_1 & -y_1 x'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 x'_2 & -y_2 x'_2 \\ x_3 & y_3 & 0 & 0 & 0 & 0 & -x_3 x'_3 & -y_3 x'_3 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0 y'_0 & -y_0 y'_0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y'_1 & -y_1 y'_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2 y'_2 & -y_2 y'_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3 y'_3 & -y_3 y'_3 \end{bmatrix}}_M \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{bmatrix} = \begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \\ y'_0 \\ y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}$$

D
 X'

Résultats

