

Les opérations géométriques ont pour but de modifier la position des informations contenues dans l'image sans modifier le niveau de gris. Ces opérations peuvent s'appliquer à la totalité des points d'une image (les pixels), un objet particulier de l'image, voire à certains points spécifiques (recalage de points caractéristiques).

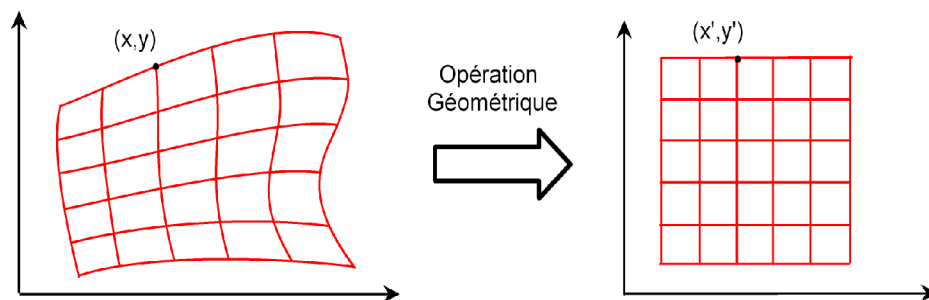


Fig 1 - Recalage géométrique d'une image

Ces opérations présentent un intérêt dans de nombreuses applications:

- correction des *distorsions* géométriques liées au matériel optique. La distorsion la plus fréquente est celle introduite par l'objectif lui-même (effet de "tonneau" ou de "barillet").
- correction de la perspective d'une scène 3D projetée sur un capteur 2D. Pour envisager des mesures précises des objets présents dans la scène, il peut être utile de restituer une image corrigée par une opération géométrique adaptée.
- compensation du mouvement de la caméra pour un système embarqué pour ramener l'image dans un repère de représentation fixe.
- mise en correspondance d'images prises dans des conditions différentes (prises de vue satellitaires, radiographie, tomographie). Si l'objectif est de comparer l'évolution des images entre deux vues prises à des intervalles très éloignés (un an par exemple). Les conditions ne seront jamais les mêmes; la première étape sera donc un *recalage géométrique* de la nouvelle vue sur la première prise comme référence, pour une stricte mise en correspondance des deux vues.

L'objet de cette présentation des opérations géométriques est de donner des outils de base 2D. La même approche est souvent utilisée en robotique mobile lorsque la caméra est embarquée sur le robot.

Notion d'opération géométrique

Les opérations géométriques de base sont la translation, la rotation, l'homothétie ou la symétrie. Pour chacune de ces opérations, on peut considérer que l'opération est une transformation de coordonnées.

Soient $X = (x, y)$ les anciennes coordonnées et $X' = (x', y')$ les nouvelles coordonnées.

Une opération géométrique Φ est une correspondance fonctionnelle entre les anciennes et les nouvelles coordonnées : $(x', y') = \Phi(x, y)$

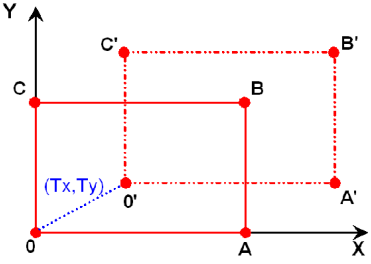
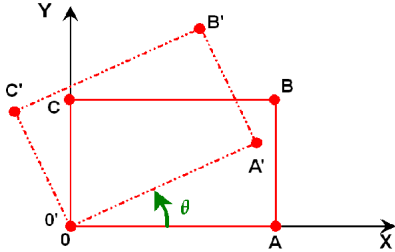
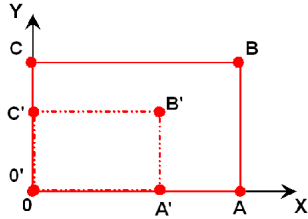
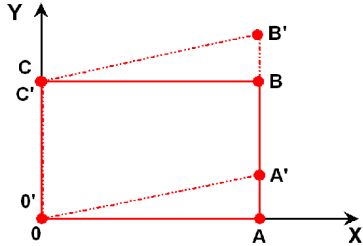
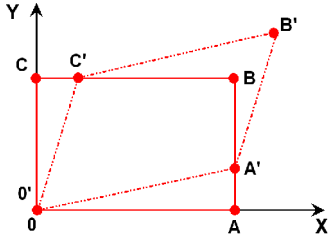
L'opération Φ peut présenter des propriétés spécifiques telles que:

- la linéarité par rapport à (x, y) ou la loi *affine*
- l'inversibilité (possibilité de retrouver (x, y) à partir de (x', y'))

Opérations géométriques élémentaires

Les opérations de base permettent de réaliser des transformations géométriques simples. Elles sont visibles lorsqu'elles affectent une structure spécifique (élément carré, maillage)

Les principales opérations ont les formes suivantes (remarquer que leur expression est une fonction affine des variables x et y):

Translation		$\begin{cases} x' = x + T_x \\ y' = y + T_y \end{cases}$
Rotation autour de l'origine		$\begin{cases} x' = \cos \theta x - \sin \theta y \\ y' = \sin \theta x + \cos \theta y \end{cases}$
Homothétie/ changement d'échelle/dilatation		$\begin{cases} x' = S_x x \\ y' = S_y y \end{cases}$ <p>Notes: ($S_x = -1, S_y = -1$) donne une symétrie centrale par rapport à l'origine, ($S_x = -1, S_y = +1$) une symétrie axiale par rapport à Ox et ($S_x = +1, S_y = -1$) une symétrie axiale par rapport à Oy</p>
Relèvement /cisaillement vertical		$\begin{cases} x' = x \\ y' = \alpha x + y \end{cases}$
Cisaillement double /transvection		$\begin{cases} x' = x + \alpha y \\ y' = \beta x + y \end{cases}$

⇒ Le point O a la propriété d'être invariant (O' est confondu avec O).

⇒ Une droite reste une droite dans la transformation affine. Cette propriété permet de mettre en évidence l'opération affine par un ensemble de droites dans l'espace d'origine et celui d'arrivée.

⇒ L'opération affine n'est pas une transformation conforme (les angles ne sont pas conservés).

Calcul homogène

Les matrices homogènes permettent de décrire toutes les opérations géométriques affine par rapport aux variables x et y . Le principe est d'introduire une troisième dimension fictive pour prendre en compte la translation.

La forme générale d'une transformation géométrique affine en matrice homogène est :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & T_x \\ c & d & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Cette représentation est utile pour comprendre la décomposition d'une opération géométrique. Par contre, il sera inutile de *programmer* une transformation homogène sous forme matricielle (la dernière ligne de calcul étant vérifiée systématiquement, le calcul effectif se réduit au calcul des lignes x' et y' !).

Cette représentation permet de décrire les opérations de base:

Translation	Rotation d'angle θ	Homothétie de rapport (S_x, S_y)	Cisaillement Vertical	Cisaillement
T	R	S	C_y	C
$\begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \alpha & 0 \\ \beta & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Le calcul homogène définit une *algèbre de calcul* pour les opérations géométriques:

- élément neutre $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ (matrice identité)

- inverse d'une opération géométrique: l'opération inverse d'une opération géométrique est caractérisée par la matrice homogène inverse de l'opération directe.

Exemple de la rotation:

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ soit une rotation d'angle } -\theta$$

Exemple de l'homothétie:

$$\begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ soit une homothétie de rapport inverse.}$$

- cascade d'opération: la cascade d'opération successives se traduit par un simple produit matriciel (attention, l'ordre des opérations successives intervient comme dans le produit matriciel).

Exemple: cascade d'une translation (T_{x1}, T_{y1}) suivie d'une translation (T_{x2}, T_{y2})

$$T_1 \circ T_2 = \begin{pmatrix} 1 & 0 & T_{x2} \\ 0 & 1 & T_{y2} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & T_{x1} \\ 0 & 1 & T_{y1} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_{x1} + T_{x2} \\ 0 & 1 & T_{y1} + T_{y2} \\ 0 & 0 & 1 \end{pmatrix} \quad \text{soit la somme des vecteurs de translation}$$

Cette algèbre de calcul permet de combiner une suite de plusieurs opération qui seront réalisées en une seule application sur l'image.

Exemple: Rotation θ suivie d'une Translation (T_x, T_y)

$$R \circ T = \begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & T_x \\ \sin \theta & \cos \theta & T_y \\ 0 & 0 & 1 \end{pmatrix}$$

Bien remarquer le caractère *non-commutatif* des deux opérations:

$$T \circ R = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & T_{x1} \\ 0 & 1 & T_{y1} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & T_x \cos \theta - T_y \sin \theta \\ \sin \theta & \cos \theta & T_x \sin \theta + T_y \cos \theta \\ 0 & 0 & 1 \end{pmatrix}$$

Dans cette deuxième écriture (translation puis rotation), on remarque que la rotation a été appliquée à la translation, d'où le nouveau vecteur de translation de l'opération globale.

Centre quelconque

Les opérations de base définies précédemment ont pour centre l'origine du repère (point O). De façon générale, le centre (C_x, C_y) peut être choisi de manière arbitraire. L'opération se décompose en trois temps:

- translation du centre sur l'origine (le vecteur de translation est l'inverse des coordonnées du centre de l'opération)
- exécution de l'opération demandée (rotation, cisaillement...)
- translation de l'origine vers le centre de l'opération (translation inverse)

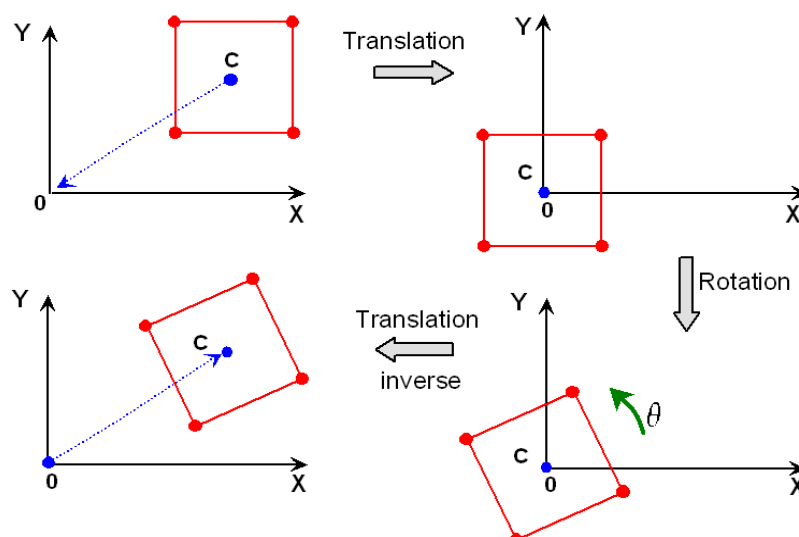


Fig 2 - Etape de la rotation de centre quelconque

L'écriture en matrice homogène montre les étapes:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \text{ avec } \begin{matrix} T_x = -C_x \\ T_y = -C_y \end{matrix}$$

Ou encore : $X' = (T^{-1} \cdot R \cdot T) X$

Selon les règles matricielles, il est évident que la deuxième translation n'annule pas la première (produit matriciel non commutatif).

Théorème

Cette algèbre permet d'affirmer le théorème suivant :

Toute opération géométrique affine par rapport aux paramètres (x, y) est décomposable en une suite d'opérations élémentaires de type translation, rotation, cisaillement et homothétie (cette décomposition n'est pas unique).

Cette propriété est très utile pour analyser la déformation d'une image et ainsi mettre en évidence les fonctionnalités de base (rotation, homothétie...)

L'opération affine étant définie par les 6 paramètres (a, b, c, d, T_x, T_y) , il suffit de connaître 3 points de l'espace transformé pour définir complètement les paramètres.

Opérations non-linéaires

Les opérations non-linéaires sont caractérisées par des fonctions de calcul de $(x', y') = \Phi(x, y)$ non-linéaires, comme par exemple des termes polynomiaux xy , x^2 ou y^2 .

• transformation bilinéaire dite "perspective"

Un terme dit *bilinéaire* est un terme de la forme xy . L'opération géométrique induite est une forme particulière de perspective (utilisée dans de nombreux logiciels de retouche photo)

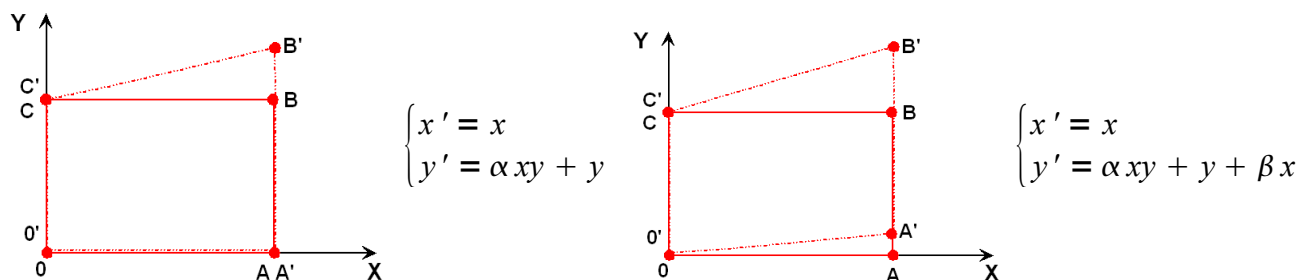


Fig 3 - Transformations bilinéaires de perspective

On remarque que la deuxième perspective est la cascade d'un cisaillement et d'une perspective simple

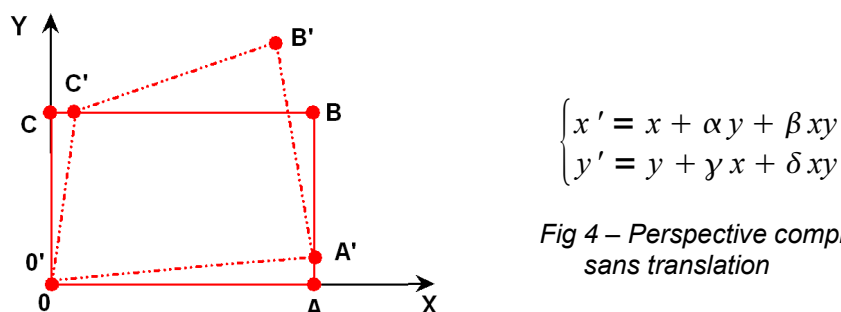
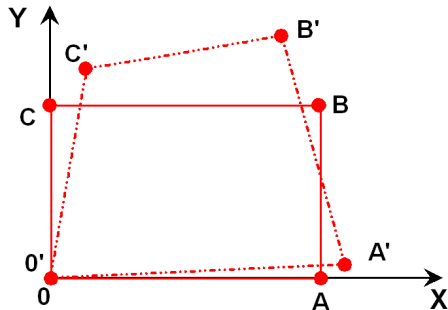


Fig 4 – Perspective complète sans translation

• **transformation bilinéaire générale**

L'expression de la transformation bilinéaire générale est :
$$\begin{cases} x' = a_1 x + a_2 y + a_3 xy + a_4 \\ y' = a_5 x + a_6 y + a_7 xy + a_8 \end{cases}$$

La transformation géométrique est la suivante :



Influence des coefficients:

- a_1 et a_6 définissent le rapport homothétique
- a_2 et a_5 définissent le relèvement et l'inclinaison
- a_3 et a_7 définissent la perspective
- a_4 et a_8 définissent la translation générale (nulle sur cet exemple).

Fig 5 - Transformation bilinéaire complète

On remarque que les 8 paramètres scalaires de la transformation peuvent être fixés par les 4 points de contrôle O', A', B', C' (deux coordonnées par point soit 8 équations pour 8 coefficients).

Attention : l'inverse d'une transformation bilinéaire (correction d'une perspective par exemple) n'est pas une transformation bilinéaire !

• **transformation non-linéaire**

Dans le cas le plus général, la transformation géométrique peut utiliser des termes polynomiaux en x^2 , y^2 , x^3 , y^3 voire des termes du genre $\sin \alpha x$ ou $\sin \beta y$.

Exemple d'une distorsion géométrique

Les objectifs de prise de vue induisent fréquemment des distorsions à caractère isotrope depuis l'axe optique, comme celle introduite par un objectif grand angle. Ces distorsions sont liées à la distance $\rho = \sqrt{x^2 + y^2}$ entre le point de l'image et l'axe optique. La correction de cette aberration géométrique utilise la transformation inverse de la distorsion.

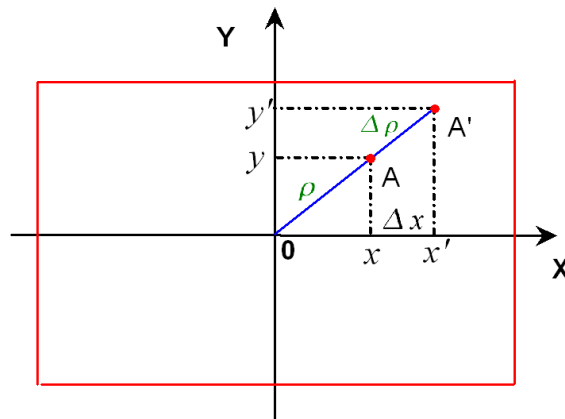


Fig 6 - Modèle de la distorsion en barillet

- exemple de modèle de la distorsion en barillet: $\Delta \rho = \alpha \rho^3$

$$\frac{\Delta x}{x} = \frac{\Delta \rho}{\rho} \Rightarrow \Delta x = \alpha x \rho^2$$

$$\text{d'où } \begin{cases} x' = x + \alpha x \rho^2 \\ y' = y + \alpha y \rho^2 \end{cases} \quad \text{soit } \begin{cases} x' = x + \alpha(x^3 + x y^2) \\ y' = y + \alpha(x^2 y + y^3) \end{cases}$$

L'opération fait bien appel à des expressions polynomiales en (x, y) .

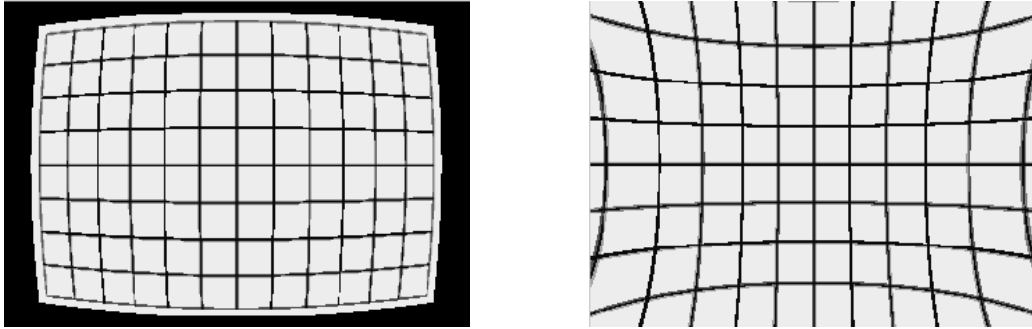


Fig 7 - Distorsion géométrique en barillet et coussinet d'une grille

Pour corriger une distorsion en barillet, une correction en coussinet de coefficient α adapté permet d'*approcher* la fonction inverse de la distorsion en coussinet.

Correction géométrique d'une image

Les opérations non-linéaires permettent de corriger de nombreuses situation de déformation géométrique d'une image. Pour définir les coefficients, il faut fixer un certain nombre de *points de contrôles*.

Les *points de contrôle* sont des points caractéristiques de l'image source pour lesquels la position dans l'image redressée est connue.

Soit $C(x,y)$ un point de contrôle dont l'image est $C'(x',y')$ dans l'image distordue (en supposant que la correction est le passage des coordonnées (x',y') vers (x,y)). Chaque point de contrôle fournit deux équations:

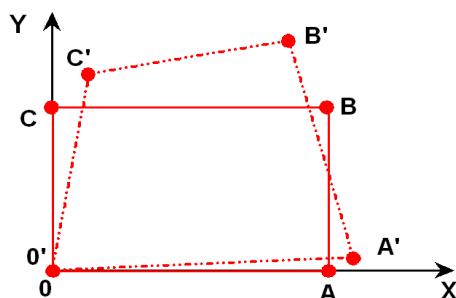
$$\begin{cases} x_C = \Phi_x(x_{C'}, y_{C'}) \\ y_C = \Phi_y(x_{C'}, y_{C'}) \end{cases}$$

Pour déterminer les paramètres d'une correction géométrique, le nombre de points de contrôle *indépendants* doit être égal à la *moitié* du nombre de paramètres à déterminer (si le nombre de paramètres est impairs, il suffit d'ignorer l'une des équations).

Attention: en raison des *symétries* éventuelles, certains points sont dépendants (par exemple, les 4 sommets de la distorsion en barillet sont symétriques et donneraient 4 systèmes d'équations dépendantes). De même, les points *invariants* d'une transformation géométrique ne peuvent pas servir de point de contrôle.

La correction est déterminée par la *résolution* du système d'équations. Si la position imposée par les points de contrôles est impossible à obtenir par la transformation géométrique utilisée, on pourra adopter une solution *optimale* au sens des moindres-carrés.

- Exemple de la perspective complète :



$$\begin{cases} x = a_1 x' + a_2 y' + a_3 x' y' \\ y = a_4 y' + a_5 x' + a_6 x' y' \end{cases}$$

Fig 8 – Correction d'une perspective par une transformation bilinéaire (valable pour une faible perspective)

Les points de contrôle peuvent être les 3 sommets du quadrilatère (A'B'C')

Pour ces points de contrôle, le résultat de la correction géométrique Φ est connu :

$$\begin{cases} x_A = \Phi_x(x_{A'}, y_{A'}) \\ y_A = \Phi_y(x_{A'}, y_{A'}) \end{cases} \quad \begin{cases} x_B = \Phi_x(x_{B'}, y_{B'}) \\ y_B = \Phi_y(x_{B'}, y_{B'}) \end{cases} \quad \begin{cases} x_C = \Phi_x(x_{C'}, y_{C'}) \\ y_C = \Phi_y(x_{C'}, y_{C'}) \end{cases}$$

On dispose de 6 équations pour 6 inconnues dans la description de la transformation. Il suffit donc de résoudre le système d'équations pour obtenir les 6 coefficients.

L'écriture matricielle permet de résoudre très facilement le système :

$$\begin{pmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{pmatrix} = \begin{pmatrix} x_{A'} & y_{A'} & x_{A'}y_{A'} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{A'} & y_{A'} & x_{A'}y_{A'} \\ x_{B'} & y_{B'} & x_{B'}y_{B'} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{B'} & y_{B'} & x_{B'}y_{B'} \\ x_{C'} & y_{C'} & x_{C'}y_{C'} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{C'} & y_{C'} & x_{C'}y_{C'} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \quad \text{soit } X = H\theta$$

La solution est : $\theta = H^{-1} \cdot X$

L'ajout de points de contrôle *supplémentaires* permet d'améliorer la précision de détermination des coefficients de la correction géométrique.

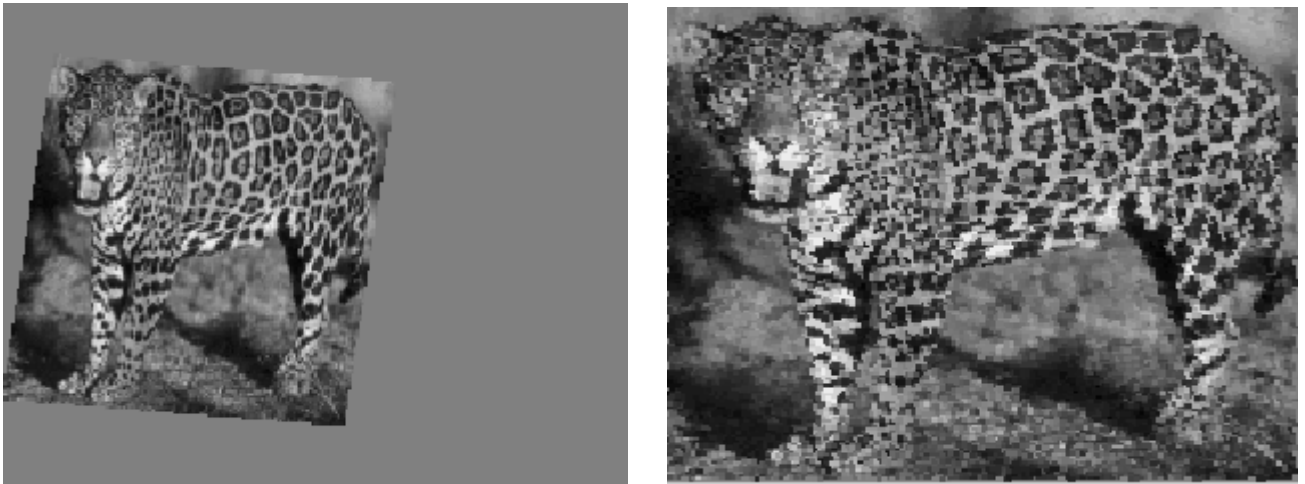


Fig 9- Recalage géométrique d'une image

Pour chaque point supplémentaire, le vecteur X augmente de 2 lignes ainsi que la matrice H qui devient rectangulaire. Le système comporte donc plus d'équations que d'inconnues. Sa résolution pourra faire appel à la méthode des moindres-carrés qui minimise le critère quadratique $J = (X - H\theta)(X - H\theta)^T$ (voir cours M1 ASE de Modélisation/Identification)

La solution optimale est : $\hat{\theta} = (H^T H)^{-1} H^T X$

Correction exacte d'une perspective

Lorsque des objets plans se projettent sur un capteur, les images observées depuis différents points de vue (droit ou oblique) sont liées par une transformation projective appelée Homographie de la forme : $P' = H P$ avec P et P' points en correspondance de coordonnées $(x, y, z)^T$ et $(x', y', z')^T$ (la dimension z' est fixée à 1 pour l'image plane)

L'Homographie H est caractérisée par une matrice homogène 3×3 . Cette transformation induit un facteur d'échelle et ne comporte que 8 coefficients indépendants ($h_{33}=1$)

Pour rectifier une image, il faut déterminer les 8 coefficients de la transformation H qui amène les points de l'image source plane à une position de référence. Les points sont liés par :

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad \text{et} \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Avec 4 points $P(x,y)$ de l'image source assignés à des positions $P'(x',y')$ de référence, on dispose de 8 équations avec 8 inconnues. Le système se met sous la forme suivante :

$$\begin{aligned} x'(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

Avec 4 points de contrôle, on obtient le système suivant :

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0x'_0 & -y_0x'_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0y'_0 & -y_0y'_0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \end{pmatrix} \times \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \\ y'_0 \\ y'_1 \\ y'_2 \\ y'_3 \end{pmatrix}$$

La résolution de ce système linéaire permet de déterminer la matrice homographique de redressement de la perspective.

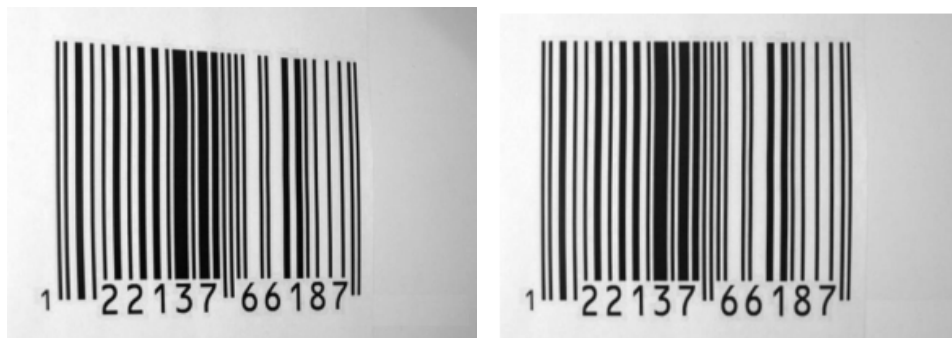


fig 10 Correction de la perspective par la transformation homographique (d'après L; Jagannathan)

Interpolation des niveaux de gris

La mise en oeuvre des opérations géométriques sur les images discrètes demande une algorithmique spécifique. En effet, l'idée de "construire" la nouvelle image à partir du balayage de l'original ne permet pas d'obtenir une image correcte.

En effet, la transformation des coordonnées d'origines discrètes (donc entières) peut donner des valeurs non-entières dans le résultat du calcul $(x_{A'}, y_{A'}) = \phi(x_A, y_A)$

La position d'un tel point est donc hors de la maille d'échantillonnage (fig 10).

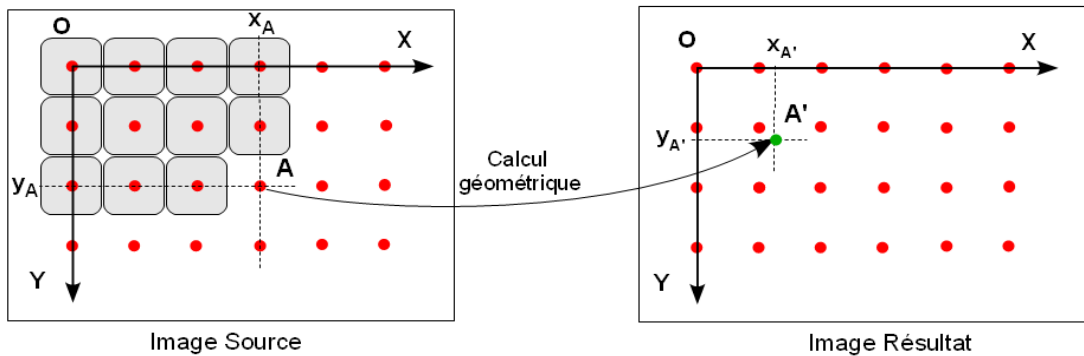


fig 10 - Transformation géométrique d'une image discrète

La solution d'affecter le résultat au pixel le plus proche par exemple ne résout pas le problème de la construction de l'image résultat; certains pixels seraient affectés de deux valeurs différentes, d'autres d'aucune valeur.

La méthode est de balayer les points de l'image résultat, dont les coordonnées sont dans la maille d'échantillonnage du résultat et à rechercher le niveau de gris équivalent du point source (fig 11). A noter que la maille d'échantillonnage du résultat pourra être différente de celle de l'image d'origine dans le cas d'un *ré-échantillonnage* (resampling en maille carrée, rectangulaire, hexagonale...)

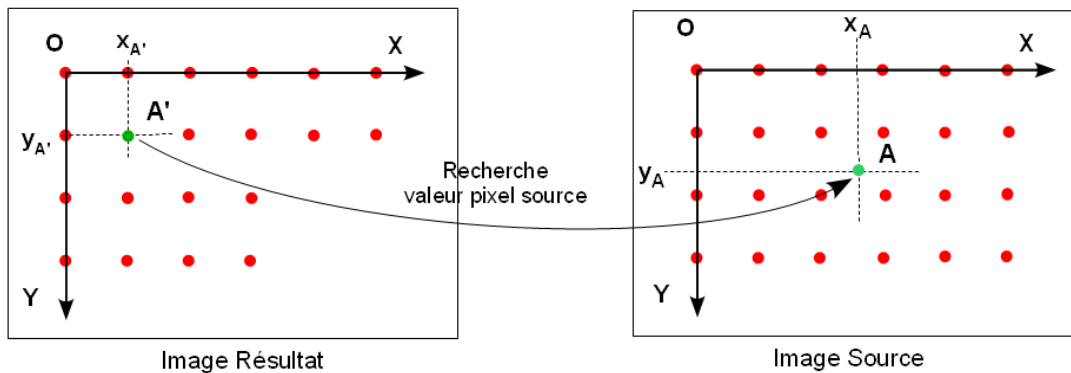


fig 11 - Algorithme de construction de l'image transformée

L'opération géométrique permettant de repasser des nouvelles coordonnées aux anciennes est l'inverse de l'opération directe de la forme $(x_A, y_A) = \phi^{-1}(x_{A'}, y_{A'})$ soit :

$$\begin{cases} x_A = \Phi_x^{-1}(x_{A'}, y_{A'}) \\ y_A = \Phi_y^{-1}(x_{A'}, y_{A'}) \end{cases}$$

Pour les opérations linéaires, il suffit de prendre la matrice inverse de l'opération. Les opérations bilinéaires sont inversibles aussi. Dans le cas général, la transformation inverse, si elle n'existe pas, pourra être approchée par un développement polynomial si nécessaire.

Plusieurs méthodes sont proposées pour interpoler la valeur du niveau de gris de A.

- **Interpolation du plus proche voisin**

C'est l'algorithme le plus simple, rapide à calculer. La fonction mise en oeuvre est celle de l'arrondi (*round*, *floor* ou *ceil*).

L'algorithme est donc (indichage des images et bornes à modifier selon le langage) :

```

for x = 1 à Xmax
  for y = 1 à Ymax
    x1 = round ( phi1_x ( x , y ))
    y1 = round ( phi1_y ( x , y ))
    if (x1>=1 & y1>=1 & x1<Xmax & y1<Ymax)
      Image_resultat( x , y ) = Image_source( x1 , y1 )
    end
  end
end
end

```

La qualité du résultat est assez faible : le phénomène de *crénage* est très visible en cas de rotation d'un élément à aspect géométrique.

- **Interpolation Bilinéaire du niveau de gris**

L'interpolation bilinéaire prend en compte pour le calcul les 4 pixels les plus proches du pixel inconnu, en donnant un poids évalué à partir de la distance du point calculé aux voisins.

Soient (m, n) les coordonnées du point P_0 le plus proche obtenues par arrondi par valeur inférieure (*floor*) et $(\delta x, \delta y)$ les coordonnées relatives du point A par rapport au point P_0 . L'interpolation du niveau de gris par la loi bilinéaire est donnée par :

$$I_A = (1 - \delta x)(1 - \delta y)I(m, n) + \delta x(1 - \delta y)I(m+1, n) \\ + (1 - \delta x)\delta y I(m, n+1) + \delta x \delta y I(m+1, n+1)$$

avec $\delta x, \delta y \in [0, 1]$

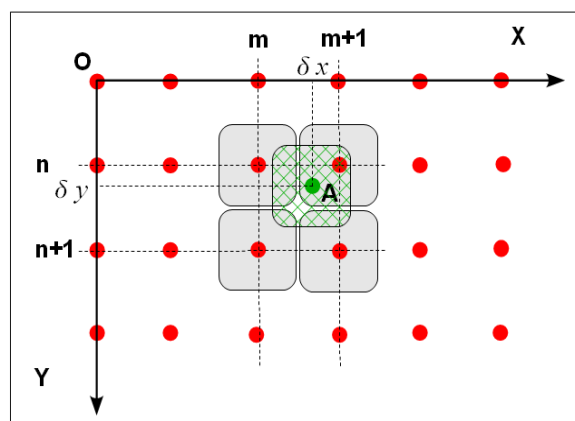


Image Source

fig 12 Interpolation bilinéaire

On remarque que cette interpolation revient à calculer la valeur du nouveau pavé à partir de son intersection avec ceux du maillage initial.

- **Interpolation Bicubique du niveau de gris**

L'interpolation bicubique va au-delà de la méthode bilinéaire, en prenant en compte les 16 pixels les plus proches du point à interpoler. L'idée est d'ajuster un modèle polynomial sur les 16 niveaux de gris de l'image source, puis d'en déduire le niveau du point interpolé par calcul de la valeur prise par le modèle.

Le modèle a la forme suivante :

$$M_g(x, y) = \sum_{p=0}^{p=3} \sum_{q=0}^{q=3} \alpha_{pq} x^p y^q$$

Le voisinage est centré autour du point A (même technique de positionnement par arrondi de la valeur des coordonnées (x, y) de A).

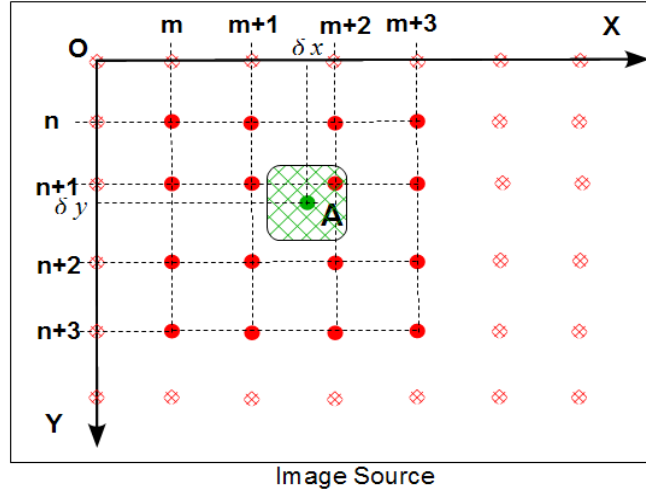


fig 13 - Domaine de modélisation de l'interpolateur bicubique

On remarque que le modèle bicubique possède 16 coefficients; le voisinage d'ajustement représentant 16 points de contrôle, l'ajustement sera exact.

Il suffit donc d'écrire les 16 équations et de les résoudre (par Cramer par exemple). La résolution donne l'expression des coefficients α_{pq} en fonction des niveaux de gris de l'image source. La valeur prise par le point A sera donc :

$$NG(A) = \sum_{p=0}^{p=3} \sum_{q=0}^{q=3} \alpha_{pq} \delta x^p \delta y^q$$

L'interpolation bicubique donne de meilleurs résultats que les méthodes précédentes; avec un bon rapport temps de calcul/qualité de rendu. Beaucoup de logiciels laissent le choix de la méthode à l'utilisateur.