Rappels mathématiques,

Transformations géométriques 2D et 3D

Michel Buffa, UNSA, UFR Sciences, 1996

1 Introduction

2 Rappels sur les vecteurs et leurs propriétés

Les vecteurs sont utilisés pour représenter la position des sommets des objets, l'orientation des surfaces, le comportement de la lumière inter-agissant avec les objets solides ou transparents, etc...
Nous utiliserons surtout les vecteurs de manière algébrique plutôt que géométrique, afin de constituer un ensemble d'équations concises d'algèbre linéaire, une partie de votre "livre de cuisine mathématique en infographie".

On notera les vecteurs *en italiques*, les réels à l'aide de caractères de l'alphabet grec, les matrices en majuscules.

2.1. Addition de deux vecteurs

- Commutative et associative : $u + v = v + u$ et $u + (v + w) = u + (v + w)$
- Il existe un vecteur, généralement noté 0, tel que pour n'importe quel vecteur u, on a $u + 0 = 0 + u = u$.
- Pour chaque vecteur u il existe un autre vecteur v tel que $u + v = v + u = 0$. On pourra ainsi dire que $v = -u$

L'addition se fait composante par composante.

$$\text{Soit } u = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \text{ et } v = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix}. \quad \text{(EQ 1)}$$

$$u + v = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \\ 12 \end{bmatrix} \quad \text{(EQ 2)}$$

2.2. Multiplication scalaire

Il s'agit de la multiplication d'un vecteur par un nombre réel. Voici les principales propriétés de la multiplication scalaire :

- $(\alpha\beta) \cdot v = \alpha(\beta \cdot v)$
- $1 \cdot v = v \cdot 1 = v$
- $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$
- $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$

Comme pour l'addition, cette opération se fait composante par composante :

$$u = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \alpha \cdot u = \begin{bmatrix} \alpha \cdot x \\ \alpha \cdot y \\ \alpha \cdot z \end{bmatrix} \tag{EQ 3}$$

2.3. Combinaison linéaire de vecteurs

Une combinaison linéaire de vecteurs $v_1, v_2 \ldots v_n$ est représentée par un vecteur de la forme $\alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + \ldots + \alpha_n \cdot v_n$.
Les combinaisons linéaires sont utilisées pour décrire de nombreux objets, par exemple les courbes et les surfaces.

2.4. Produit scalaire entre deux vecteurs

Etant donné deux vecteurs à n dimensions :

$$u_1 = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \text{ et } u_2 = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}, \tag{EQ 4}$$

Le produit scalaire entre ces deux vecteurs est défini de la manière suivante :

$$u_1 \cdot v_1 = x_1 y_1 + x_2 y_2 + \ldots x_n y_n, \text{ c'est un réel.} \tag{EQ 5}$$

Remarque : la distance du point (x, y) à l'origine (0, 0) est égale à $\sqrt{x^2 + y^2}$. En général, la distance du point $(x_1, x_2 \ldots x_n)$ à l'origine, dans un espace à n dimensions est égale à $\sqrt{x_1^2 + x_2^2 + \ldots x_n^2}$.
Soit $v = [x_1, x_2, \ldots x_n]^T$, alors la distance précédente est égale à $\sqrt{v \cdot v}$.
Ceci sera notre définition de la longueur d'un vecteur à n dimensions. On notera également cette longueur $\|v\|$ (norme de v).

La distance entre deux points dans un espace à n dimensions est définie de manière similaire : $d(P, Q)$

est égale à la norme du vecteur $\overrightarrow{QP}$.

Propriétés du produit scalaire :

- Symétrique : $v \cdot w = w \cdot v$
- Non dégénéré : $v \cdot v = 0$ si et seulement si $v = 0$
- Bilinéaire : $v \cdot (u + \alpha w) = v \cdot u + \alpha(v \cdot w)$
- Le produit scalaire peut être utilisé pour générer des vecteurs dont la longueur est égale à 1 (vecteurs *normalisés*).

  Pour normaliser un vecteur v, on calcule $v' = v / \|v\|$. Un vecteur de longueur 1 est également appelé un vecteur *unitaire*.
- Le produit scalaire peut servir *à mesurer des angles* : l'angle entre le vecteur u et le vecteur v est égal à :

$$\alpha = \operatorname{acos}\left(\frac{u \cdot v}{\|u\|\|v\|}\right) \tag{EQ 6}$$

  Si u et v sont normalisés, il est inutile de diviser par $\|u\|\|v\|$. Dans ce cas, le produit scalaire représente le cosinus de l'angle entre les deux vecteurs : $u \cdot v = \cos(\alpha)$.

  Conséquence de cette propriété : *si $u \cdot v = 0$ alors u est orthogonal à v.*

2.5. Produit vectoriel

Le produit vectoriel entre deux vecteurs u et v est un vecteur perpendiculaire à u et à v noté $w = u \wedge v$. La méthode permettant de calculer la valeur de w est proche de celle permettant de calculer le déterminant d'une matrice 3×3 (voir section "Déterminants" page 5).

En 3-D (le produit vectoriel n'existe que dans des dimensions supérieures ou égales à 3) :

$$\text{soit } u = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \text{ et } v = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \text{ on aura } w = \begin{bmatrix} y_1 z_2 - z_1 y_2 \\ z_1 x_2 - x_1 z_2 \\ x_1 y_2 - y_1 x_2 \end{bmatrix} \tag{EQ 7}$$

- La longueur d de w est égale à $d = \|u\|\|v\||\sin(\alpha)|$ si α est l'angle entre u et v.
- Si u et v sont normalisés, $w = u \wedge v$ l'est aussi *si et seulement si u et v sont perpendiculaires!*

  En effet, la norme de w représente l'aire du parallélogramme obtenu en faisant la somme (géométrique) des deux vecteurs u et v. Si $u = v$ l'aire est nulle et la longueur de w est égale à zéro ($d = |\sin(0)| = 0$). Si $u \perp v$, l'aire est maximale ($d = |\sin(\Pi/2)| = 1$).

3 Rappels sur les matrices

3.1. Définitions, conventions de notation

Une matrice est un ensemble rectangulaire de nombres, fréquemment utilisée lors d'opérations sur les vecteurs.

Les éléments d'une matrice, en général des nombres réels, sont doublement indexés, et par convention le premier indice indique la ligne et le second la colonne.

Les conventions mathématiques veulent que ces indices commencent à 1. Certains langages de programmation, comme le langage C, font commencer les indices à 0. Pour ce cours, nous nous en tiendrons aux conventions mathématiques, au programmeur de “shifter” les indices!

Ainsi, avec ces conventions, si A est une matrice, $a_{3,2}$ est l'élément de A situé à la 3$^{\text{ème}}$ ligne, 2$^{\text{ème}}$ colonne.

Lorsque des indices symboliques sont utilisés, comme dans a_{ij}, ou lorsqu'il n'y a pas d'ambiguïtés, on omet en général la virgule entre les indices.

Un vecteur $u = [x_1, x_2, \ldots x_n]^T$ peut-être considéré comme étant une matrice $n \times 1$ (n lignes, 1 colonne).

Nous allons faire un usage intensif des matrices pendant ce cours. Elles jouent un rôle très important car elles permettent de réaliser les transformations géométriques de base, la visualisation tridimensionnelle, etc...

3.2. Multiplication de matrices

Règle : si A est une matrice $n \times m$ avec des éléments a_{ij}, si B est une matrice $m \times p$ avec des éléments b_{ij}, alors la matrice $C = A \cdot B$ est une matrice $n \times p$ dont les éléments c_{ij} sont définis de la manière suivante :

$$c_{ij} = \sum_{s=1}^{n} a_{is} b_{sj} \qquad \text{(EQ 8)}$$

Si l'on s'imagine les colonnes de B comme étant des vecteurs $B_1, B_2 \ldots B_n$ et les lignes de A comme étant les vecteurs $A_1, A_2 \ldots A_n$ mais tournés de 90 degrés, à l'horizontale, alors on voit que chaque élément c_{ij} de C est égal au produit scalaire $A_i B_j$.

Les propriétés usuelles de la multiplication scalaire sont vraies à part la commutativité :

- En général $AB \neq BA$
- La multiplication matricielles est distributive par rapport à l'addition : $A(B+C) = AB+AC$.
- Il existe une matrice identité : $I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$ (rien que des 1 sur la diagonale, 0 ailleurs).

3.3. Déterminants

Le déterminant *d'une matrice carrée* est calculée à partir des éléments de cette dernière. Ce calcul est légèrement compliqué car sa définition est récursive. Partons d'une matrice 2×2 :

Soit $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, son déterminant D est égal au produit croisé $D = ad - bc$.

Le déterminant d'une matrice A de taille $n \times n$ est défini à partir des déterminants de ses sous-matrices. Soit D_{1i} le déterminant de la sous-matrice de taille $(n-1) \times (n-1)$ que l'on obtient en supprimant la $1^{\text{ère}}$ ligne et la $\text{i}^{\text{ème}}$ colonne de la matrice A (taille $n \times n$), alors le déterminant de A est égal à :

$$D_A = \sum_{i=1}^{n} (-1)^{1+i} \cdot D_{1i} \tag{EQ 9}$$

Cas particulier d'une matrice 3×3.

Soit $A = \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix}$, son déterminant D est égal à

$$D = (v_2 w_3 - v_3 w_2) - (v_1 w 3 - v_3 w_1) + (v_1 w_2 + v_2 w 1) \tag{EQ 10}$$

3.4. Transposition de matrice

Une matrice $n \times m$ peut être "flippée" (transposée) par rapport à sa diagonale. Si la matrice A possède des éléments $a_{ij}, i = 1\ldots n, j = 1\ldots m$, alors sa transposée B aura comme éléments $b_{ij}, (a_{ij}, i = 1\ldots m, j = 1\ldots n)$ avec $b_{ij} = a_{ji}$.

Cette nouvelle matrice est appelée *la transposée* de A et on la notera A^T.

Si l'on considère un vecteur à n dimensions comme une matrice $n \times 1$ alors son transposé est une matrice

$1 \times n$ (un vecteur ligne).

Si A et B sont des matrices, à l'aide de la transposée nous pouvons écrire :

- $(A \cdot B)^T = B^T \cdot A^T$

3.5. Matrice inverse

La multiplication matricielle diffère de la multiplication ordinaire : une matrice A peut très bien ne pas avoir d'inverse. En fait, seules les matrices carrées peuvent avoir un inverse, et seules celles dont le déterminant est non nul en ont un.

Si A et B sont deux matrices $n \times n$, et si $AB = BA = I$ (la matrice identité), alors B est l'inverse de A et on pourra le noter A^{-1}.

Pour des matrices possédant des éléments réels, il suffira de démontrer que $AB = I$ *ou bien* $BA = I$. Si l'un est vrai, l'autre l'est aussi.

Pour n'importe quelle matrice $n \times n$, la meilleure méthode pour trouver son inverse est par élimination selon la méthode de Gauss, en particulier pour les matrices de dimension supérieure ou égale à 3. En Infographie, nous manipulerons le plus souvent des matrices 4×4, cependant la plupart du temps, leur inverse sera trivial à calculer.

4 Transformations géométriques 2-D

4.1. Rappels

4.1.1. Translation

On peut translater un point (x, y) du plan vers une nouvelle position en ajoutant une valeur de translation à ses coordonnées. Pour chaque point $P(x, y)$ devant être déplacé de t_x unités de valeur le long de l'axe des X, et de t_y unités de valeur le long de l'axe des Y, vers une nouvelle position $P'(x', y')$, on peut écrire:

$$x' = x + t_x \text{ et } y' = y + t_y \qquad \text{(EQ 11)}$$

On peut écrire ceci sous forme matricielle en prenant les deux points sous la forme de vecteurs colonnes :

$$\text{Avec } P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \text{ on aura } P' = P + T \qquad \text{(EQ 12)}$$

En Infographie, pour déplacer un objet constitué de lignes, de sommets, de faces, il suffira de translater tous les sommets puis de retracer les lignes et les faces. Cette méthode est également vraie pour les autres types de transformations (rotations, zooms, etc...) que nous allons étudier maintenant.

4.1.2. Rotations

Pour faire tourner un point d'un angle α *autour de l'origine* :

$$\begin{aligned} x' &= x\cos\alpha - y\sin\alpha \\ y' &= x\sin\alpha + y\cos\alpha \end{aligned} \qquad \text{(EQ 13)}$$

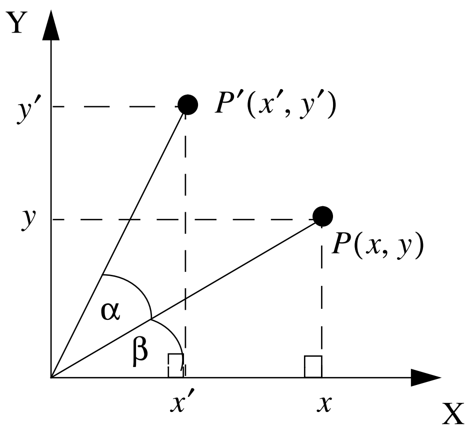
Sous forme matricielle :

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \\ P' &= R \cdot P \end{aligned} \qquad \text{(EQ 14)}$$

Démonstration de ces résultats :

Rappelons quelques conventions trigonométriques :

- Un angle est positif quand on tourne vers la gauche, de l'axe des X vers l'axe des Y.
- $\cos(-\alpha) = \cos(\alpha)$ et $\sin(-\alpha) = -\sin\alpha$



On a :

$$\begin{aligned} x &= R\cos\beta \\ y &= R\sin\beta \end{aligned} \qquad \text{(EQ 15)}$$

et :

$$\begin{aligned} x' &= R\cos(\alpha+\beta) = R\cos\alpha\cos\beta - R\sin\alpha\sin\beta \\ y' &= R\sin(\alpha+\beta) = R\cos\alpha\sin\beta + R\sin\alpha\cos\beta \end{aligned} \qquad \text{(EQ 16)}$$

En substituant (EQ 15) dans (EQ 16), on obtient bien :

$$
\begin{aligned}
x' &= x\cos\alpha - y\sin\alpha \\
y' &= x\sin\alpha + y\cos\alpha
\end{aligned}
\qquad \text{(EQ 17)}
$$

Attention : ici, on tourne *autour de l'origine*, pour tourner autour d'un point quelconque, voir la section "Composition de transformations 2-D" page 14.

4.2. Scaling

On peut "scaler" (retailler) un ensemble de points avec un facteur de grossissement s_x le long de l'axe des X, et s_y le long de l'axe des Y à l'aide de la multiplication :

$$
\begin{aligned}
\begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \\
P' &= \quad S \quad \cdot P
\end{aligned}
\qquad \text{(EQ 18)}
$$

Ici, on fait un "scaling" *par rapport à l'origine*. Si les facteurs de scaling sont inférieurs à 1, l'objet se rapprochera de l'origine, sinon il s'en éloignera.

Il est possible de "scaler" par rapport à un point quelconque, comme nous allons bientôt le voir.

Si $s_x \neq s_y$ les proportions changent, il s'agit d'un scaling *différentiel*, si $s_x = s_y$ le scaling est dit *uniforme*.

5 Coordonnées homogènes

5.1. Pourquoi ajouter une nouvelle coordonnée ?

Les représentations matricielles pour la translation, la rotation et le scaling sont les suivantes :

$$
\begin{aligned}
P' &= P + T \\
P' &= R \cdot P \\
P' &= S \cdot P
\end{aligned}
\qquad \text{(EQ 19)}
$$

...où P représente un point, et P' son transformé. Malheureusement, on le voit bien, la translation est traitée différemment du scaling et de la rotation (on utilise dans un cas une addition, dans les deux autres une multiplication). Nous aimerions traiter ces trois transformations de manière *homogène*, en appliquant dans tous les cas la même opération. Si on y arrive, cela permettra de combiner plus facilement différentes transformations.

Si les points sont exprimés en *coordonnées homogènes* ces trois opérations -nous allons le voir- peuvent être réalisées à l'aide de la seule multiplication :

- Lorsqu'on travaille en coordonnées homogènes, on ajoute une nouvelle coordonnée aux points. En 2-D chaque point, au lieu d'être représenté par une paire (x, y), est représenté par un triplet (x, y, w).
- En même temps, on dit que deux triplets de coordonnées homogènes représentent le même point s'ils sont multiples. Par exemple, le point (2, 3, 6) et le point (4, 6, 12) représentent le même point! Ainsi, un même point peut posséder plusieurs représentations en coordonnées homogènes.
- Si la coordonnée w est différente de zéro, on peut diviser les trois coordonnées par w : (x, y, w) représente en effet le même point que $(x/w, y/w, 1)$. Lorsque $w \neq 0$ on effectue en général cette opération et les nombres x/w et y/w sont appelés les *coordonnées cartésiennes* du point homogène.
- Les points avec $w = 0$ sont appelés "points à l'infini", ils serviront à décrire des directions.

Habituellement, des triplets de coordonnées servent à représenter des points en 3-D, mais ici on les utilise pour représenter des points 2-D. On peut relier ces deux représentations de la manière suivante : si on prend tous les triplets représentant le même point, c'est à dire tous les triplets de la forme (kx, ky, kw) avec $k \neq 0$, on obtient une droite dans l'espace 3-D. Si on *homogénéise* le point en divisant ses coordonnées par kw, on obtient un point de la forme $(X, Y, 1)$. Les points homogènes forment donc un plan défini par l'équation $w = 1$ dans l'espace (X, Y, W). Voir Figure 1.

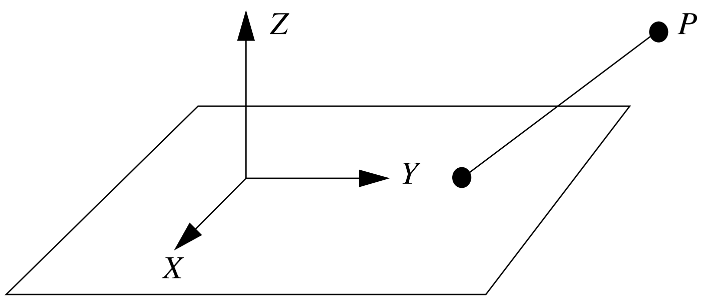


Figure 1 : L'espace des coordonnées homogènes (X, Y, W), le point P projeté sur le plan W=1

5.2. Transformations en coordonnées homogènes

Parce que les points sont maintenant des vecteurs à trois éléments, les matrices de transformation doivent aussi être de taille 3×3.

5.2.1. Translations en coordonnées homogènes :

La translation en coordonnées homogènes devient :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$P' = T(t_x, t_y) \cdot P \tag{EQ 20}$$

Que se passe-t-il si un point P est translaté de $T(t_{x1}, t_{y1})$ en P', puis translaté par $T(t_{x2}, t_{y2})$ en P'' ? Intuitivement, on suspecte que le résultat est une translation $T(t_{x1} + t_{x2}, t_{y1} + t_{y2})$.
En effet :

$$\begin{aligned} P' &= T(t_{x1}, t_{y1}) \cdot P \\ P'' &= T(t_{x2}, t_{y2}) \cdot P' \end{aligned} \tag{EQ 21}$$

On substitue P dans la deuxième équation et on obtient :

$$P'' = T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) \cdot P \tag{EQ 22}$$

...or :

$$T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) = \begin{bmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{bmatrix} = T(t_{x1} + t_{x2}, t_{y1} + t_{y2}) \tag{EQ 23}$$

La composée de deux translations est bien égale à la somme des translations.

5.2.2. Scaling en coordonnées homogènes

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$P' = S(s_x, s_y) \cdot P \tag{EQ 24}$$

On a deviné que la composition des translations était additive, on devine également que la composition des homothéties (autre nom plus général donné au scaling) est... multiplicative.
En effet :

$$\begin{aligned} P' &= S(s_{x1}, s_{y1}) \cdot P \\ P'' &= S(s_{x2}, s_{y2}) \cdot P' \end{aligned} \tag{EQ 25}$$

On substitue P dans la deuxième équation et on obtient :

$$P'' = S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) \cdot P \tag{EQ 26}$$

...or :

$$S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) = \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1}s_{x2} & 0 & 0 \\ 0 & s_{y1}s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= S(s_{x1}s_{x2}, s_{y1}s_{y2}) \tag{EQ 27}$$

5.2.3. Rotations en coordonnées homogènes

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$P' = R(\alpha) \cdot P \tag{EQ 28}$$

La composition des rotations est additive, on le démontre avec la même méthode que nous venons d'utiliser pour les translations et le scaling. Pour ne pas surcharger ce cours nous vous laissons le loisir de faire la démonstration chez vous.

5.2.4. Matrices orthogonales

Dans la sous-matrice 2×2 de la matrice de l'équation (EQ 28), en pointillés, considérons les deux lignes comme des vecteurs : $\overrightarrow{V_1} = \begin{bmatrix} \cos\alpha & -\sin\alpha \end{bmatrix}^T$ et $\overrightarrow{V_2} = \begin{bmatrix} \sin\alpha & \cos\alpha \end{bmatrix}^T$.

Ces deux vecteurs possèdent les propriétés suivantes :

1. Ce sont des vecteurs unitaires (leur norme est égale à 1).
2. Chacun est perpendiculaire à l'autre (leur produit scalaire est égal à 0).
3. Le déterminant de la sous-matrice $\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$ est égal à 1 : chacun des vecteurs doit être tourné (dans le sens trigonométrique) de α pour se retrouver aligné sur les axes X et Y (Figure 2):

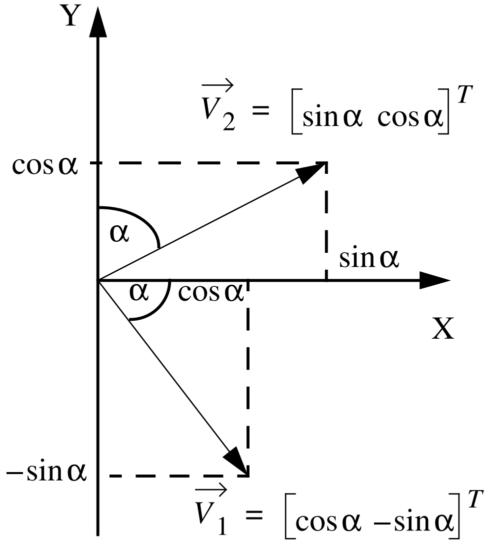


Figure 2 : les vecteurs lignes de la matrice de rotation forment un repère orthonormé. Si on tourne ce repère de l'angle de la rotation, ces vecteurs se confondent avec les axes.

Si l'on considère les deux vecteurs colonnes de la sous matrice $\vec{V_1} = [\cos\alpha\ \sin\alpha]$ et $\vec{V_2} = [-\sin\alpha\ \cos\alpha]$, alors les deux premières propriétés sont également vraies :

1. Ils sont unitaires.
2. Ils sont orthogonaux.
3. La transformation définie par $R(\alpha)$ appliquée aux deux vecteurs unitaires de l'axe des X et de l'axe des Y produit ces deux vecteurs colonnes (Figure 3) :

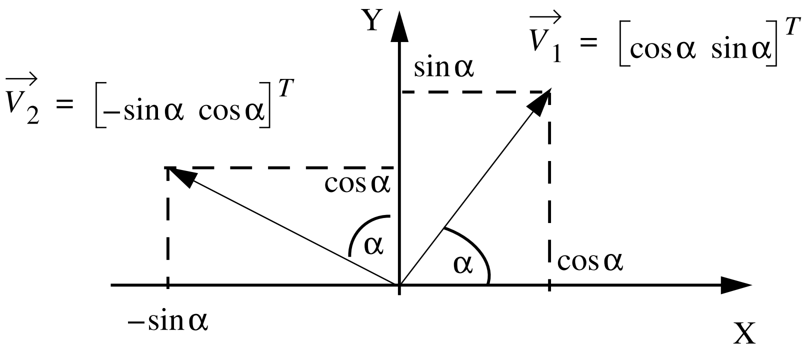


Figure 3 : Si on tourne de l'angle de la rotation les axes X et Y, ils se confondent avec les deux vecteurs colonnes de la matrice de rotation

Une matrice ayant ces propriétés est appelée une *matrice spéciale orthogonale*.

5.2.5. Transformations rigides, transformations affines

Une matrice de transformation quelconque du type :

$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 29)}$$

où la sous-matrice 2×2, en pointillés, est orthogonale, préservera les angles et les longueurs. Un carré de côté 1 sera toujours un carré de côté 1 une fois transformé par une matrice de ce type.

On dit qu'il s'agit d'une transformation *rigide*, car le corps des objets transformés n'est pas distordu après la transformation.

Une séquence arbitraire de rotations et de translations est toujours une matrice de la forme de celle présentée dans l'équation (EQ 29).

Que peut-on dire du résultat obtenu en composant des translations, des rotations *et des scalings* ? On obtient ce qu'on appelle *une transformations affine*. Les transformations affines possèdent la propriété suivante : elles conservent le parallélisme des lignes mais ne conservent ni les longueurs ni les angles.

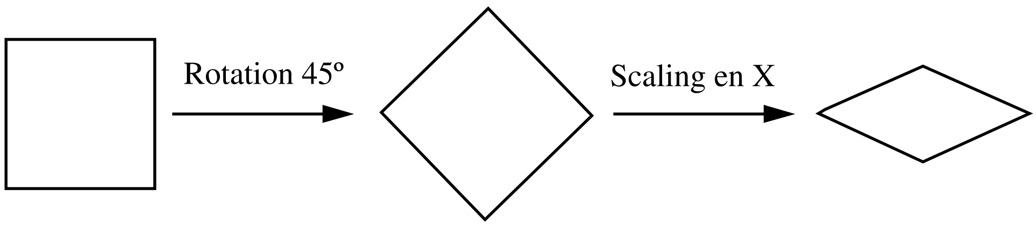


Figure 4 : Exemple de transformation affine

Les rotations, translations et scaling sont chacune des transformations affines. Si on applique n'importe laquelle de ces transformations à un parallélogramme, ses côtés demeurent parallèles.

5.2.6. Glissements

Il existe un autre type de transformation affine, le "glissement" ou encore "cisaillement" le long des axes X et/ou Y ("shear transformation" en anglais). La Figure 5 montre l'effet d'un glissement en X et en Y sur un carré.

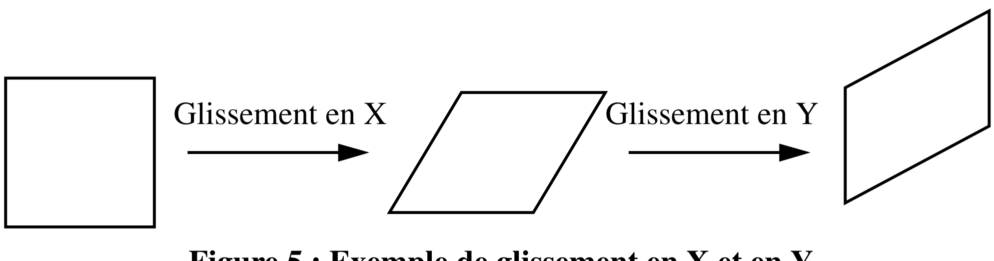


Figure 5 : Exemple de glissement en X et en Y

Cette opération est représentée par les matrices suivantes (a et b sont des constantes de proportionnalité) :

$$SH_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ et } SH_y = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 30)}$$

On a bien :

$$SH_x \cdot P = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + ay \\ y \\ 1 \end{bmatrix} \quad \text{(EQ 31)}$$

Le point $P' = SH_x \cdot P$ correspond bien au point P auquel on a fait subir un changement proportionnel de sa coordonnée x par rapport à sa coordonnée y.

6 Composition de transformations 2-D

Nous allons dans cette section composer les transformations 2-D fondamentales que nous venons d'étudier, pour en tirer des résultats plus généraux.
L'avantage principal de la composition des transformations est un gain en efficacité, car on applique finalement aux points d'un objet une seule transformation au lieu de leur appliquer une série de transformations.

Considérons la rotation d'un objet autour d'un point $P_1 = \begin{bmatrix} x_1 & y_1 \end{bmatrix}^T$ arbitraire. Puisque nous savons comment faire tourner un objet autour de l'origine, il suffit de transformer le problème de départ (complexe) en trois sous-problèmes simples :

1. Translater P_1 à l'origine.
2. Effectuer la rotation.
3. Translater P_1 vers son ancienne position.

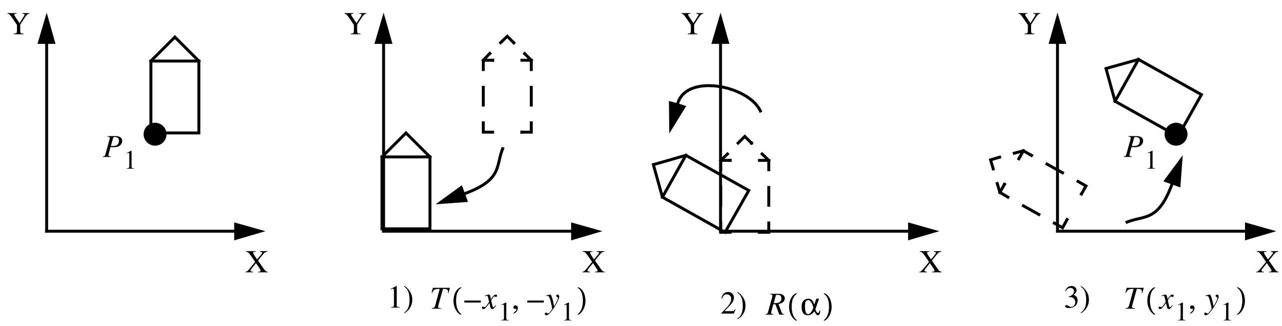


Figure 6 : Rotation autour d'un point arbitraire

Voyons ce que cela donne d'un point de vue mathématique. Soit $M = T(x_1, y_1) \cdot R(\alpha) \cdot T(-x_1, -y_1)$, si on développe ce résultat on obtient :

$$M = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha & -\sin\alpha & -x_1\cos\alpha + y_1\sin\alpha \\ \sin\alpha & \cos\alpha & -x_1\sin\alpha - y_1\cos\alpha \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha & -x_1\cos\alpha + y_1\sin\alpha + x_1 \\ \sin\alpha & \cos\alpha & -x_1\sin\alpha - y_1\cos\alpha + y_1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & x_1(1-\cos a) + y_1\sin\alpha \\ \sin\alpha & \cos\alpha & -x_1\sin\alpha + y_1(1-\cos\alpha) \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 32)}$$

On utilisera une approche similaire pour "scaler" un objet par rapport à un point P_1 arbitraire :

1. Translater P_1 à l'origine.
2. Effectuer le scaling.
3. Translater P_1 vers son ancienne position.

La transformation résultante $M = T(x_1, y_1) \cdot S(s_x, s_y) \cdot T(-x_1, -y_1)$ sera égale à :

$$M = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & x_1(1-s_x) \\ 0 & s_y & y_1(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 33)}$$

De la même manière, si l'on veut scaler, tourner et positionner en P_2 l'objet de la Figure 6 avec P_1 comme centre de rotation, on décomposera la transformation que l'on désire obtenir en une série de transformations élémentaires :

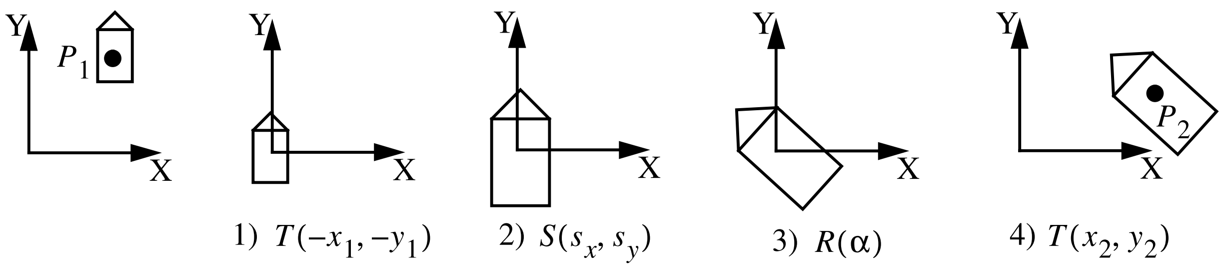


Figure 7 : Décomposition d'une transformation complexe en transformations élémentaires

Si M_1 et M_2 représentent une translation (T), une rotation (R) ou un scaling (S), quand a-t-on $M_1 \cdot M_2 = M_2 \cdot M_1$?
En général, bien sûr, la multiplication matricielle n'est pas commutative. Il est cependant facile de montrer que $M_1 \cdot M_2 = M_2 \cdot M_1$ dans les cas suivants :

Table 1: Cas où $M_1 \cdot M_2 = M_2 \cdot M_1$

M_2	M_2
T	T
S	S
R	R
S (avec $s_x = s_y$)	R

7 Transformation Window-Viewport (Zoom, etc...)

En général, les objets de la scène sont exprimés dans le repère du monde (world coordinate system), qui possède ses propres unités de mesure (mètres, centimètres, millimètres, peu importe). On utilise le terme *monde* car le programme d'application représente *un monde virtuel* créé par l'utilisateur.

Les routines graphiques quant à elles vont servir à visualiser les objets appartenant au monde sur l'écran, qui possède son propre système de coordonnées, exprimées *en pixels*.

En général, on est amené à définir deux "fenêtres" :

- L'une, appelée WINDOW, exprimée dans le repère du monde, qui contient la partie du monde que l'on souhaite visualiser.
- L'autre, appelée VIEWPORT, définit la partie de l'écran dans laquelle on veut afficher les objets du monde contenus dans la fenêtre WINDOW. La taille et la position de VIEWPORT sont exprimées en pixels.

Etant donné une WINDOW et un VIEWPORT, quelle est la matrice de transformation qui permet de passer de l'un vers l'autre ?

Cette transformation peut s'effectuer en plusieurs étapes (voir Figure 8) :

4. Translation de la fenêtre WINDOW à l'origine du repère du monde.
5. Mise à l'échelle de la fenêtre WINDOW pour qu'elle prenne la taille du VIEWPORT (passage du repère du monde vers le repère écran).
6. Translation du VIEWPORT vers la position désirée dans le repère écran.

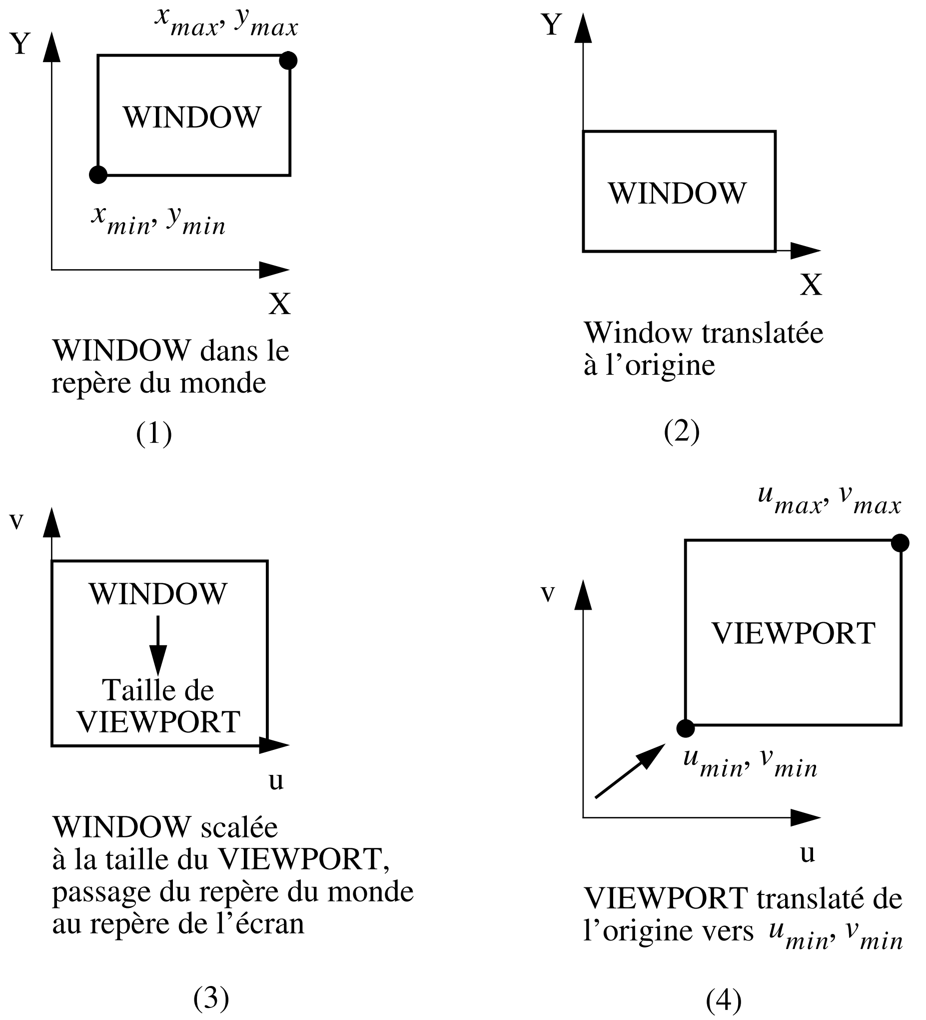


Figure 8 : Décomposition de la transformation WINDOW-VIEWPORT en 4 étapes

La transformation finale M_{wv} peut s'écrire sous la forme d'un produit de 3 transformations, chacune correspondant à une des trois étapes ci-dessus. Rappelons qu'on écrit les transformations consécutives de droite à gauche dans le produit matriciel.

$$M_{wv} = T(u_{min}, v_{min}) \cdot S\left(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}, \frac{v_{max} - v_{min}}{y_{max} - y_{min}}\right) \cdot T(-x_{min}, -y_{min}) \qquad \text{(EQ 34)}$$

Développons maintenant le résultat précédent :

$$M_{wv} = \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} & 0 & 0 \\ 0 & \frac{v_{max} - v_{min}}{y_{max} - y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 35)}$$

Soit :

$$M_{wv} = \begin{vmatrix} \frac{u_{max} - u_{min}}{x_{max} - x_{min}} & 0 & -x_{min}\left(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}\right) + u_{min} \\ 0 & \frac{v_{max} - v_{min}}{y_{max} - y_{min}} & -y_{min}\left(\frac{v_{max} - v_{min}}{y_{max} - y_{min}}\right) + v_{min} \\ 0 & 0 & 1 \end{vmatrix} \quad \text{(EQ 36)}$$

Donc finalement, si on a un point $P_w = [x, y, 1]^T$ appartenant à un objet dans le repère du monde. Si P_w se trouve dans la WINDOW, il se trouvera représenté sur l'écran en :

$$P_v = M_{wv} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{vmatrix} (x - x_{min})\left(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}\right) + u_{min} \\ (y - y_{min})\left(\frac{v_{max} - v_{min}}{y_{max} - y_{min}}\right) + v_{min} \\ 1 \end{vmatrix} \quad \text{(EQ 37)}$$

8 Optimisation

Nous l'avons vu, la composition des transformations de base (Rotation, Translation et Scaling) produit une transformation affine dont la matrice a la forme suivante :

$$M = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 38)}$$

La sous matrice 2×2 (en pointillés) est la matrice composant rotations et scaling alors que t_x et t_y représentent la translation finale.
Calculer $P' = M \cdot P$ revient à effectuer 9 multiplications et 6 additions, cependant la structure fixe de la dernière ligne de M permet de simplifier le calcul de $M \cdot P$:

$$x' = r_{11}x + r_{12}y + t_x$$
$$y' = r_{21}x + r_{22}y + t_y \qquad \text{(EQ 39)}$$

Ce calcul comporte seulement 4 additions et 4 multiplications.

Conclusion : les matrices sont très pratiques pour composer les différentes transformations, mais il est possible d'optimiser les calculs lorsqu'on exploite la transformation finale.

9 Représentation matricielle des transformations 3-D

9.1. Conventions, systèmes de coordonnées direct/indirect

De la même manière que les transformations 2-D sont représentées par des matrices 3×3 en utilisant des coordonnées homogènes, les transformations 3-D seront représentées à l'aide de matrices 4×4. Au lieu de représenter un point par un triplet (x, y, z), on utilisera un quadruplé homogène (x, y, z, w). Deux de ces quadruplets, différents de $(0, 0, 0, 0)$, représentent un même point si l'un est multiple de l'autre.

Comme en 2-D, une représentation standard d'un point $(x, y, z, w), w \neq 0$, est donnée par $(x/w, y/w, z/w, 1)$. Les points pour lesquels $w = 0$ sont dits *points à l'infini*.

Le système de coordonnées conventionnellement utilisé est *direct*, on dit aussi *orienté main-droite* (voir Figure 9).

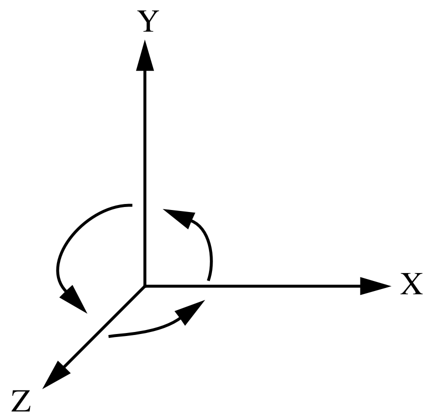


Figure 9 : système de coordonnées direct

Par convention, une rotation positive dans un système direct est telle que lorsqu'on s'embroche sur la pointe de l'un des axes, un rotation de 90° dans le sens inverse des aiguilles d'une montre transforme les axes de la manière suivante :

Table 2: Rotation positive dans un repère direct

Axe de rotation	Direction de la rotation positive
X	Y $\Rightarrow$Z
Y	Z $\Rightarrow$X
Z	X$\Rightarrow$Y

9.2. Translations en 3-D

$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{EQ 40}$$

$$T(t_x, t_y, t_z) \cdot [x, y, z, w]^T = [x + t_x, y + t_y, z + t_z, 1]^T \tag{EQ 41}$$

9.3. Scaling en 3-D

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{EQ 42}$$

$$S(s_x, s_y, s_z) \cdot [x, y, z, w]^T = [s_x \cdot x, s_y \cdot y, s_z \cdot z, 1]^T \tag{EQ 43}$$

9.4. Rotations en 3-D

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{EQ 44}$$

$R_z(\alpha)$ est une rotation d'angle α autour de l'axe des Z. Il s'agit en fait de la rotation 2-D adaptée à la 3-D, Z étant l'axe perpendiculaire au plan (X, Y), voir la Figure 10.

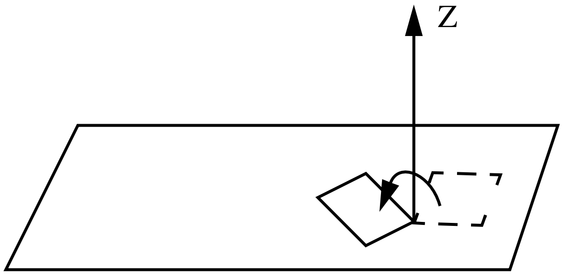


Figure 10 : Rotation autour de l'axe des Z

On exprime les rotations autour des axes X et Y de la manière suivante :

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ et } R_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 45)}$$

Les sous-matrices 3×3 en pointillés sont des matrices *spéciales orthogonales* (voir section 5.2.4. page 11). Elles possèdent les propriétés suivantes :

1. Les trois vecteurs lignes et les trois vecteurs colonnes sont unitaires
2. Les trois vecteurs lignes sont orthogonaux, les trois vecteurs colonnes aussi.
3. La transformation définie par $R(\alpha)$, appliquée aux trois vecteurs lignes, produit les vecteurs unitaires des axes X, Y et Z.
4. La transformation définie par $R(\alpha)$, appliquée aux vecteurs unitaires des axes X, Y et Z, produit les trois vecteurs colonnes.

Ces propriétés seront très utiles par la suite, elles nous permettront de construire très simplement des matrices de changement de repère.

En outre, la sous-matrice 3×3 d'une matrice résultant de la composition de multiples rotations, est spéciale orthogonale. Rappelons que les matrices spéciales orthogonales préservent les angles et les distances.

9.5. Transformations inverses

Chacune des matrices de transformation R (rotation), S (scaling) et T (translation) possède un inverse très simple à obtenir :

- On obtient l'inverse d'une translation $T(t_x, t_y, t_z)$ en inversant les signes de t_x, t_y et t_z.
- L'inverse d'une matrice de scaling $S(s_x, s_y)$ s'obtient en inversant s_x et s_y.
- L'inverse d'une rotation $R(\alpha)$ est la rotation $R(-\alpha) = (R(\alpha))^T$.
- Plus généralement, l'inverse de n'importe quelle *matrice orthogonale* M est égale à M^T.

Astuce : prendre la transposée d'une matrice ne signifie pas forcément déplacer les éléments de la matrice. On peut souvent se contenter d'inverser les indices des éléments de la matrice avant d'y accéder.

9.6. Glissements en 3-D

On exprimera en 3-D un glissement le long des axes X et Y à l'aide de la matrice : $SH_{xy} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

En effet : $\begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ \end{bmatrix} = \begin{bmatrix} x + sh_x \cdot z \\ y + sh_y \cdot z \\ z \\ 1 \end{bmatrix}$

9.7. Transformation d'un plan

Jusqu'à présent, nous nous sommes contentés de transformer des points (les sommets des arêtes des objets). Pour déplacer, tourner, retailler un objet, on transforme juste des points, puis on redessine les arêtes. Nous allons voir maintenant qu'il peut aussi être utile de transformer des plans.

Pour transformer un plan défini par 3 points, on agira de la même manière, on transformera les points pour obtenir la nouvelle équation du plan. Cependant, souvent les plans sont définis uniquement par leur équation; nous allons voir qu'il est possible de recalculer ces cœfficients en utilisant directement la transformation. On peut aussi vouloir non pas transformer le plan lui-même, mais son plan normal.

Représentons le plan par un vecteur colonne dont les éléments sont les coefficients de son équation : $N = [A, B, C, D]^T$. Dans ce cas, le plan est défini par tous les points $P = [x, y, z, 1]^T$ qui vérifient $N^T \cdot P = 0$:

$$\begin{bmatrix} A & B & C & D \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0 \qquad \text{(EQ 46)}$$

Supposons que l'on transforme les points du plan P à l'aide d'une matrice M :

$$P' = M \cdot P \qquad \text{(EQ 47)}$$

Pour conserver $N^T \cdot P = 0$ avec les points transformés, on aimerait transformer N en $N' = [A', B', C', D']$ par une matrice Q (que l'on cherche) telle que l'équation suivante reste vraie :

$$\underbrace{(Q \cdot N)^T} \cdot \underbrace{M \cdot P} = 0$$

$$\overbrace{(N')^T} \cdot P' = 0 \tag{EQ 48}$$

On peut réécrire cette équation de la manière suivante, puisque $(A \cdot B)^T = B^T \cdot A^T$:

$$N^T \cdot Q^T \cdot M \cdot P = 0 \tag{EQ 49}$$

Puisque $N^T \cdot P = 0$ (EQ 46), cette équation reste vraie si $Q^T \cdot M$ est un multiple de la matrice identité, en particulier pour $Q^T \cdot M = I$. Dans ce cas on a $Q^T = M^{-1}$ ou encore $Q = (M^{-1})^T$.

Ainsi, le vecteur colonne N' contenant les cœfficients de l'équation du plan transformé par M est donné par :

$$N' = Q \cdot N = (M^{-1})^T \cdot N \tag{EQ 50}$$

Remarque N° 1 : la matrice $(M^{-1})^T$ peut ne pas exister si le déterminant de M est égal à zéro. Cette situation apparaît lorsque M est une matrice de projection (que nous étudierons bientôt).

Remarque N° 2 : La normale au plan avant transformation est égale à $\vec{n} = [A, B, C, 0]^T$, après transformation à $\vec{n'} = [A', B', C', 0]^T$. (la normale est une direction, c'est pourquoi sa quatrième coordonnée est nulle).

10 Composition des transformations 3-D

10.1. Un petit problème à résoudre pour mieux comprendre

Dans cette section nous allons étudier la composition des transformations 3-D à l'aide d'un exemple qui nous sera bien utile par la suite.

Problème : Transformer deux segments de droite P_1P_2 et P_1P_3 de leur position courante (a) vers une nouvelle position (b) (voir Figure 11) :

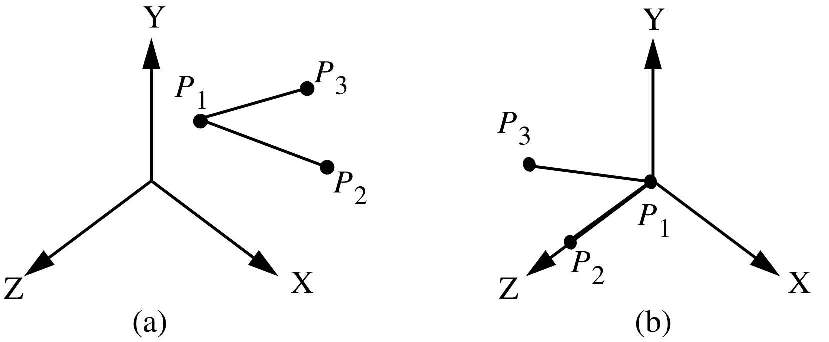


Figure 11 : Déplacer en 3-D un objet

Ainsi, le point P_1 doit être translaté à l'origine, P_1P_2 doit être aligné avec l'axe Z, et P_1P_3 doit se trouver dans le demi-plan positif (Y, Z). On ne souhaite pas changer la taille des segments.

Nous allons étudier deux moyens différents permettant de transformer les segments de (a) vers (b) :

- Le premier consiste à composer des transformations fondamentales (Translation + Rotation). Cette approche longue et fastidieuse est facile à illustrer, mais permet de bien comprendre comment on décompose un problème à priori complexe en une suite de transformations simples.
- Le second utilise les propriétés des matrices orthogonales. Il s'agit d'une méthode beaucoup plus courte mais plus abstraite.

10.1.1. Première approche

On peut décomposer le problème en quatre étapes :

1. Translater P_1 à l'origine.
2. Tourner autour de l'axe Y jusqu'à ce que P_1P_2 se trouve dans le plan (Y, Z).
3. Tourner autour de l'axe des X jusqu'à ce que P_1P_2 soit aligné avec l'axe Z.
4. Tourner autour de l'axe des Z jusqu'à ce que P_1P_3 se trouve dans le plan (Y, Z).

10.1.1.1. Etape 1

On a $P_1 = [x_1, y_1, z_1, 1]^T$, on veut le translater à l'origine.

$$\text{Soit } T(-x_1, -y_1, -z_1) = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 51)}$$

On transforme P_1, P_2 et P_3 par $T(-x_1, -y_1, -z_1)$:

$$P'_1 = T \cdot P_1 = [0, 0, 0, 1]^T \qquad \text{(EQ 52)}$$

$$P'_2 = T \cdot P_2 = [x_2 - x_1, y_2 - y_1, z_2 - z_1, 1]^T \qquad \text{(EQ 53)}$$

$$P'_3 = T \cdot P_3 = [x_3 - x_1, y_3 - y_1, z_3 - z_1, 1]^T \qquad \text{(EQ 54)}$$

10.1.1.2. Etape 2

Rotation autour de l'axe des Y pour amener P_1P_2 dans le plan (Y, Z). La Figure 12 présente la situation après l'étape 1 :

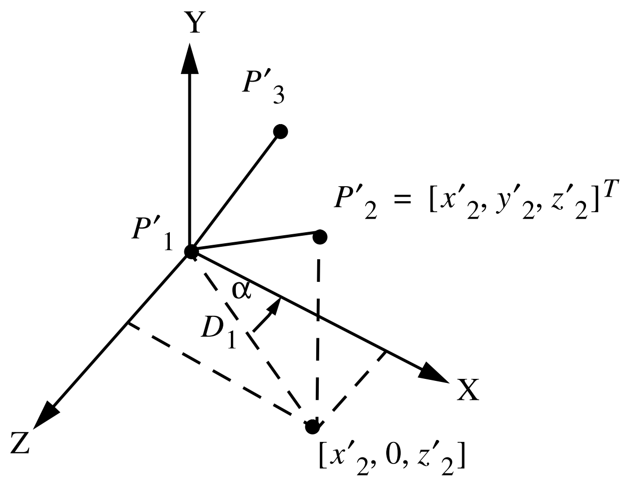


Figure 12 : Situation après l'étape 1

D'après la figure, l'angle de la rotation permettant d'amener $P'_1P'_2$ dans le plan (Y, Z) est égal à $-(90 - \alpha) = (\alpha - 90)$ degrés.

On a donc :

$$\cos(\alpha - 90) = \cos(\alpha - \pi/2) = \sin\alpha = z'_2/D_1 = (z_2 - z_1)/D_1$$
$$\sin(\alpha - 90) = \sin(\alpha - \pi/2) = -\cos\alpha = -x'_2/D_1 = -(x_2 - x_1)/D_1 \qquad \text{(EQ 55)}$$

$$\text{Avec} \quad D_1 = \sqrt{(x')^2 + (y')^2} = \sqrt{(x_2 - x_1)^2 + (z_2 - z_1)^2} \qquad \text{(EQ 56)}$$

On peut maintenant écrire la matrice de la rotation autour de l'axe Y :

$$R_y(\alpha - 90) = \begin{bmatrix} \cos(\alpha - 90) & 0 & \sin(\alpha - 90) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha - 90) & 0 & \cos(\alpha - 90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (z_2 - z_1)/D_1 & 0 & -(x_2 - x_1)/D_1 & 0 \\ 0 & 1 & 0 & 0 \\ (x_2 - x_1)/D_1 & 0 & (z_2 - z_1)/D_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 57)}$$

$$P''_2 = R_y(\alpha - 90) \cdot P'_2 = \begin{bmatrix} (z_2 - z_1)/D_1 & 0 & -(x_2 - x_1)/D_1 & 0 \\ 0 & 1 & 0 & 0 \\ (x_2 - x_1)/D_1 & 0 & (z_2 - z_1)/D_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ y_2 - y_1 \\ D_1 \\ 1 \end{bmatrix} \quad \text{(EQ 58)}$$

Comme on s'y attendait, la composante en X de P''_2 vaut zéro, et la composante en Z vaut 1.

10.1.1.3. étape 3

Tourner autour de l'axe des X pour aligner le segment $P''_1P''_2$ sur l'axe des Z.

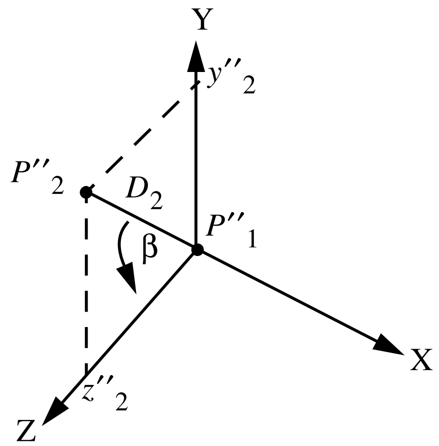


Figure 13 : Situation après l'étape 2

$$\text{On a } \cos\beta = (z''_2)/D_2 \text{ et } \sin\beta = (y''_2)/D_2\,, \quad \text{(EQ 59)}$$

$$\text{avec } D_2 = \|P''_1P''_2\| = \|P_1P_2\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad \text{(EQ 60)}$$

On peut calculer maintenant P'''_2 :

$$\begin{aligned} P'''_2 = R_x(\beta) \cdot P''_2 &= R_x(\beta) \cdot R_y(\alpha - 90) \cdot P'_2 \\ &= R_x(\beta) \cdot R_y(\alpha - 90) \cdot T \cdot P_2 \\ &= [0, 0, \|P_1P_2\|, 1]^T \end{aligned} \quad \text{(EQ 61)}$$

10.1.1.4. Etape 4

Tourner autour de l'axe des Z pour mettre P_1P_3 dans le plan (Y, Z).

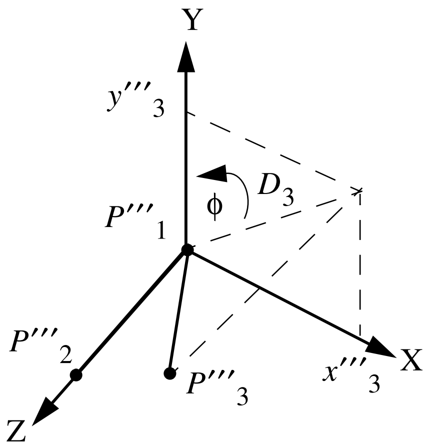


Figure 14 : Situation après l'étape 3

On a $\cos\phi = (y'''_3)/D_3$ et $\sin\phi = (x'''_3)/D_3$,avec $D_3 = \sqrt{(x'''_3)^2 + (y'''_3)^2}$., (EQ 62)

$$P''''_3 = R_z(\phi) \cdot P'''_3 \quad \text{(EQ 63)}$$

10.1.1.5. Conclusion

La matrice de transformation finale est la suivante :

$$M = R_z(\phi) \cdot R_x(\beta) \cdot R_y(\alpha - 90) \cdot T = R \cdot T \ . \quad \text{(EQ 64)}$$

Cette méthode est très fastidieuse et requiert de nombreux calculs. Nous allons maintenant étudier une autre méthode beaucoup plus simple.

10.1.2. Deuxième approche

L'autre manière d'obtenir la matrice de rotation R de l'équation (EQ 64) consiste à utiliser les propriétés des matrices orthogonales (voir la section "Matrices orthogonales" page 11). La méthode est très simple: utilisons les segments P_1P_2 et P_1P_3 de la Figure 11 pour construire un repère orthonormé. Nous avons vu que les vecteurs lignes d'une matrice de rotation sont les vecteurs directeurs d'un repère tel que si on applique la rotation à ce repère, ses vecteurs directeurs se confondent avec les axes X, Y et Z.
Etant donné que dans cet exercice nous voulons déplacer P_1P_2 de manière à amener P_1P_2 sur l'axe des Z avec P_1 à l'origine, nous allons construire un repère orthonormé tel que l'axe P_1P_2 soit le 3ème axe de ce repère.
On appellera $\overrightarrow{V_1}$. $\overrightarrow{V_2}$ et $\overrightarrow{V_3}$ les vecteurs du nouveau repère. $\overrightarrow{V_3}$ sera le vecteur unitaire le long de P_1P_2 puisque c'est ce segment qui doit se confondre avec le 3ème axe, l'axe des Z. On prendra donc $\overrightarrow{V_3} = \overrightarrow{P_1P_2} / \left\| \overrightarrow{P_1P_2} \right\|$.

On prendra $\overrightarrow{V_1}$, l'axe du nouveau repère qui va se confondre avec l'axe des X, perpendiculaire à $\overrightarrow{V_3}$, ou encore au plan $P_1P_2P_3$. Ainsi, on prendra $\overrightarrow{V_1}$ égal au produit vectoriel entre deux vecteurs de ce plan : $\overrightarrow{V_1} = (\overrightarrow{P_1P_3} \wedge \overrightarrow{P_1P_2}) / \left\| \overrightarrow{P_1P_3} \wedge \overrightarrow{P_1P_2} \right\|$.

Finalement, puisque $\overrightarrow{V_2}$ doit être perpendiculaire à $\overrightarrow{V_3}$ et à $\overrightarrow{V_1}$, on prendra $\overrightarrow{V_2} = \overrightarrow{V_3} \wedge \overrightarrow{V_1}$ (pas besoin de diviser par la norme puisque les deux vecteurs sont normalisés).

La matrice de rotation finale sera donc égale à :

$$R = \begin{bmatrix} \boxed{\begin{array}{c} \overrightarrow{V_1} \\ \hline \overrightarrow{V_2} \\ \hline \overrightarrow{V_3} \end{array}} & \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{EQ 65}$$

La matrice de transformation finale est $M = R \cdot T(-x_1, -y_1, -z_1)$.

10.2. Autre exemple

On a un objet composé de sommets, d'arêtes et de faces. Il possède son propre repère $(\overrightarrow{X_o}, \overrightarrow{Y_o}, \overrightarrow{Z_o})$ (tous les sommets sont exprimés dans ce dernier). On désire positionner cet objet dans l'espace à une position P donnée avec une certaine orientation, toutes deux exprimées dans le repère du monde dont les vecteurs directeurs sont $(\vec{X}, \vec{Y}, \vec{Z})$. Par exemple on veut que $\overrightarrow{Z_o}$ pointe dans une direction $\vec{V}$ donnée. Dans cet exemple, on ne donne pas plus de contraintes sur la direction des autres axes $\overrightarrow{X_o}$ et $\overrightarrow{Y_o}$.

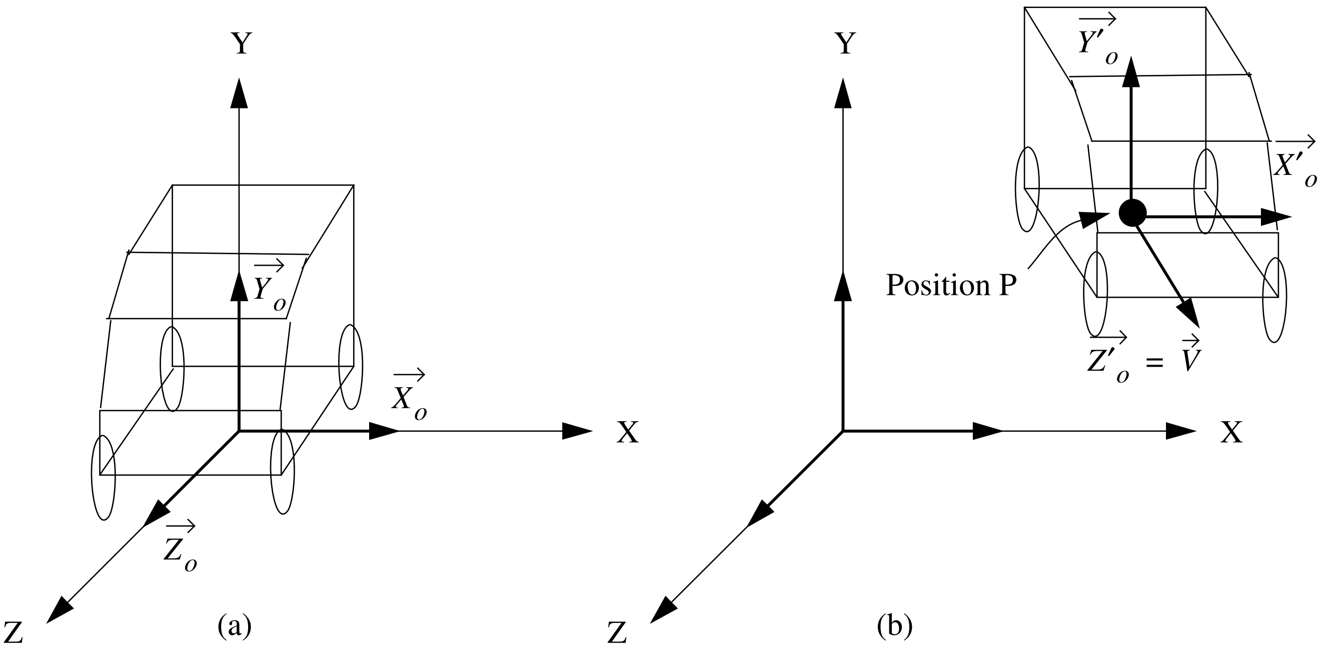


Figure 15 (a) l'objet est placé en (0, 0, 0). (b) L'objet est placé en P, avec son 3ème axe qui a pour direction $\vec{V}$

Au départ, l'objet est placé dans le monde de telle manière que les axes de son repère sont confondus avec les axes X, Y et Z du repère du monde (Figure 15a). On veut le placer en P de telle manière que $\overrightarrow{Z_o} = \vec{V}$, avec $\vec{V}$ quelconque (Figure 15b).

Le calcul de la transformation pour faire ce déplacement consiste à tourner l'objet pour qu'il pointe dans la direction voulue (celle de $\vec{V}$), puis à le translater en P. Pour trouver la matrice de rotation, on va juste déterminer dans quelle direction chacun des vecteurs $(\overrightarrow{X_o}, \overrightarrow{Y_o}, \overrightarrow{Z_o})$ doit pointer, s'assurer que ces directions sont normalisées et utiliser ces dernières comme colonnes de la matrice de rotation (voir section "Matrices orthogonales" page 11) :

- L'axe $\overrightarrow{Z_o}$ doit être transformé pour avoir la direction de $\vec{V}$ tout en étant normalisé : $\overrightarrow{Z'_o} = \vec{V}/\|\vec{V}\|$
- L'axe $\overrightarrow{X_o}$ doit être transformé en un vecteur perpendiculaire à $\overrightarrow{Z'_o}$. Comme il en existe une infinité et qu'on a pas fixé de contraintes supplémentaires sur cet axe, on va par exemple le prendre horizontal, donc perpendiculaire au vecteur directeur $\vec{Y}$ du repère du monde. Ainsi on aura $\overrightarrow{X'_o} = (\vec{Y} \wedge \overrightarrow{Z'_O})/\|\vec{Y} \wedge \overrightarrow{Z'_O}\|$. Remarque : on est obligé de normaliser car $\vec{Y}$ n'est pas

forcément orthogonal à $\overrightarrow{Z'_o}$, or le produit vectoriel de deux vecteurs donne un vecteur normalisé que si les deux vecteurs sont normalisés ET orthogonaux (voir section “Produit vectoriel” page 3).

- L’axe $\overrightarrow{Y'_o}$ est perpendiculaire à $\overrightarrow{X'_o}$ et à $\overrightarrow{Z'_o}$. On prendra donc $\overrightarrow{Y'_o} = \overrightarrow{Z'_o} \wedge \overrightarrow{X'_o}$

Ainsi, la matrice de rotation sera égale à :

$$R = \begin{bmatrix} \begin{bmatrix} \\ \overrightarrow{X'_o} \\ \\ \end{bmatrix} & \begin{bmatrix} \\ \overrightarrow{Y'_o} \\ \\ \end{bmatrix} & \begin{bmatrix} \\ \overrightarrow{Z'_o} \\ \\ \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(EQ 66)}$$

Remarque : le cas où $\vec{V}$ a la même direction que $\vec{Y}$ est dégénéré (et oui, 4 accents sur “dégénéré”) car il existe alors une infinité de vecteurs possibles pour calculer $\overrightarrow{X'_o}$. En effet, dans ce cas $\vec{Y} \wedge \vec{V} = 0$ et R n’est plus une matrice de rotation.

11 Transformations et changement de système de coordonnées

Nous avons vu comment transformer un ensemble de points appartenant à un objet, mais dans tous les cas les points originaux et leurs transformés étaient exprimés dans le même repère.
Avec cette approche, le repère ne change pas et l’objet est transformé par rapport à l’origine du repère.
Une autre manière de procéder consiste à calculer la transformation correspondant à un changement de repère. Cette approche est avantageuse lorsqu’on travaille avec plusieurs objets, chacun défini dans son propre repère, et qu’on a envie de les manipuler dans un repère unique, le repère du “monde” par exemple.
On appellera $M_{i \leftarrow j}$ la transformation qui fait passer un point représenté dans le repère j à sa représentation dans le repère i.
On notera $P^{(i)}$ un point représenté dans le repère i, $P^{(j)}$ un point représenté dans le repère j, etc...

$$P^{(i)} = M_{i \leftarrow j} \cdot P^{(j)} \quad \text{(EQ 67)}$$

$$P^{(j)} = M_{j \leftarrow k} \cdot P^{(k)} \quad \text{(EQ 68)}$$

En substituant on a :

$$P^{(i)} = M_{i \leftarrow j} \cdot M_{j \leftarrow k} \cdot P^{(k)} \quad \text{(EQ 69)}$$

Ainsi, on a :

$$M_{i \leftarrow k} = M_{i \leftarrow j} \cdot M_{j \leftarrow k} \quad \text{(EQ 70)}$$

Les changements de repères sont très importants en infographie, car il est peu réaliste d’imaginer que tous les objets d’une scène ont été créés dans un seul et même système de coordonnées. En général, les sommets de chaque objet sont exprimés dans un repère lié à l'objet. Lorsqu'on compose une scène, on prend les objets un par un, on les tourne, on modifie leur taille, puis on les positionne dans la scène. Cette dernière opération correspond à un changement de repère.

Exemple : on veut calculer le changement de repère permettant de passer du repère 2 vers le repère 1 de la figure Figure 16 :

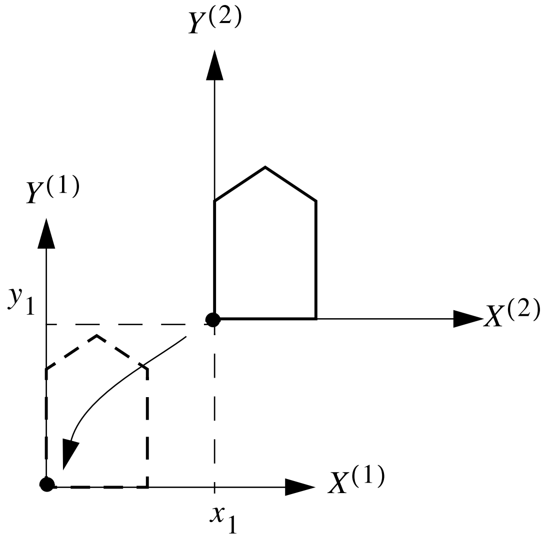


Figure 16 : Changement de repère et déplacement d'un objet

La transformation $M_{1 \leftarrow 2}$ est une translation. D'après la figure, on voit bien que l'origine du repère 2 se trouve dans le repère 1 à la position (x_1, y_1) , on en déduit que $M_{1 \leftarrow 2} = T(x_1, y_1)$. Ainsi, si on désire exprimer dans le repère 1 les points $P_i^{(2)}$ de la maison, définis au départ dans le repère 2, il faudra calculer $P_i^{(1)} = M_{1 \leftarrow 2} \cdot P_i^{(2)} = T(x_1, y_1) \cdot P_i^{(2)}$.

Autre remarque : si tous les points de la maison étaient déjà exprimés dans le repère 1, pour translater la maison à l'origine, la transformation aurait été une translation

$$T(-x_1, -y_1) = T(x_1, y_1)^{-1} = (M_{1 \leftarrow 2})^{-1} \qquad \text{(EQ 71)}$$

Règle générale : la transformation qui transforme un ensemble de points exprimés dans un unique repère d'une position vers une autre est l'inverse de la transformation qui permet de passer du repère objet se trouvant à la position de départ vers le repère objet se trouvant à la position d'arrivée.

12 Visualisation 3-D

La visualisation d'objets en 3-D est plus complexe qu'en 2-D où on se contente en général de transformations de type WINDOW-VIEWPORT. La complexité vient de la dimension supplémentaire, mais aussi parce que les supports de visualisation (l'écran de l'ordinateur) sont en 2-D. Cependant, comme toujours, si on décompose le processus de visualisation en étapes intermédiaires, tout devient plus simple.

12.1. La caméra synthétique

Pour créer des scènes 3-D il est nécessaire d'introduire la notion de caméra. La caméra est l'outil qui permet de faire le passage 3-D vers 2-D. En infographie, la cellule photosensible de la caméra correspondra à la fenêtre graphique dans laquelle une scène 3-D est visualisée. Si on déplace la caméra, on obtient une séquence animée permettant de voir les objets sous plusieurs angles.
En général, on associe un repère à la caméra et un repère au monde que l'on explore. Souvent les objets eux-mêmes ont leurs coordonnées exprimées dans leur propre repère.

Bien que le concept de caméra soit très simple, il ne suffit pas de presser un bouton de notre programme (vision très abstraite !) pour visualiser un objet 3-D, ce travail comporte plusieurs étapes importantes :

1. Spécifier le type de projection. On passe de la 3-D à la 2-D en réalisant des *projections* sur le plan écran. Nous étudierons pendant le cours deux types de projections : les projections *perspectives* et les projections *parallèles orthographiques*.
2. Spécifier des paramètres de visualisation tels que :
   - Position et orientation de la caméra.
   - Distance focale, angle d'ouverture de la caméra.
   - Taille de la fenêtre graphique dans laquelle on va visualiser.
3. Etape facultative : clipping 3-D. Si la scène contient de nombreux objets, on ne va pas forcément tous les traiter. Les objets trop éloignés ou en dehors du champs de vision peuvent être éliminés avant l'étape de projection. On appelle cette opération un *clipping 3-D*.
4. Effectuer la projection des sommets des objets sur le plan de projection, relier ces sommets par des arêtes pour former des faces et afficher le résultat dans une fenêtre graphique.

12.2. Les projections

La projection d'un objet 3-D est définie par des *rayons* qui émanent d'un *centre de projection* (qu'on appelle *centre optique* dans le cas d'une vraie caméra) et qui passent par chaque point de l'objet visé. Les intersections entre ces rayons et un *plan de projection* forment la projection (Figure 17) :

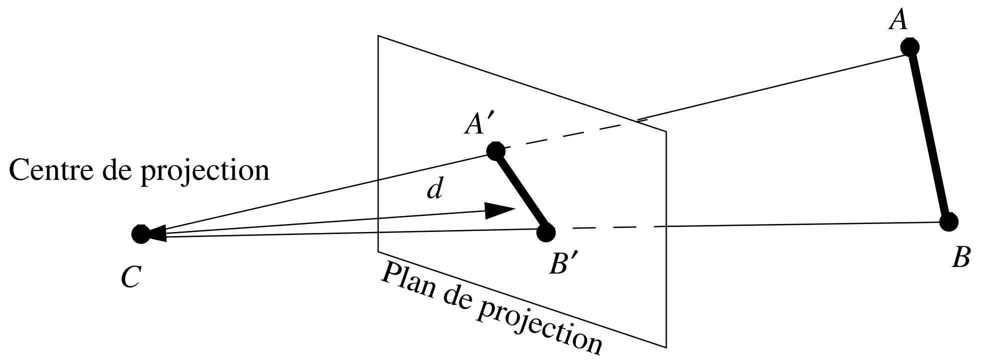


Figure 17 : Projection perspective d'un objet

Pour certains types de projection, le centre de projection peut être placé à l'infini, dans ce cas les rayons sont parallèles et orthogonaux au plan de projection. On appelle ces projections particulières des *projec-*

tions parallèles (Figure 18) :

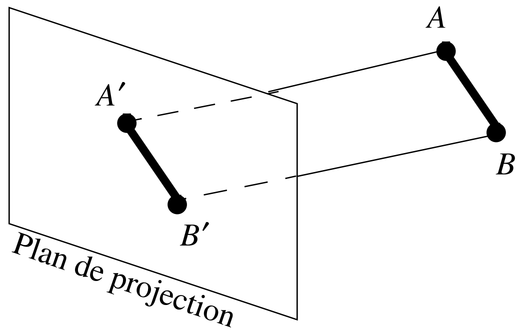


Figure 18 : projection parallèle d'un objet

Dans le cadre de ce cours nous n'étudierons que les projections *planaires* car on projette toujours sur un plan et non sur des surfaces courbes. Ne riez pas! Pensez aux géographes qui projettent leurs cartes sur des globes terrestres même pas sphériques!

Le rendu d'une projection perspective (Figure 17) est similaire à la photographie et à la vision humaine, c'est le type de projection le plus employé : la taille des objets varie inversement proportionnellement à la distance qui sépare l'objet du plan de projection. Plus un objet est éloigné, plus il est petit sur le plan de projection.

Les projections parallèles (Figure 18) sont surtout utilisées en C.A.O car la projection d'un objet conserve le parallélisme de ses arêtes (mais les angles ne sont conservés que sur les faces parallèles au plan de projection). Très utile pour représenter des vues de face, de dessous et de côté d'un objet. La taille ne varie pas en fonction de la distance de l'objet au plan de projection.

12.3. Mathématiques des projections géométriques planaires

12.3.1. Les différents repères

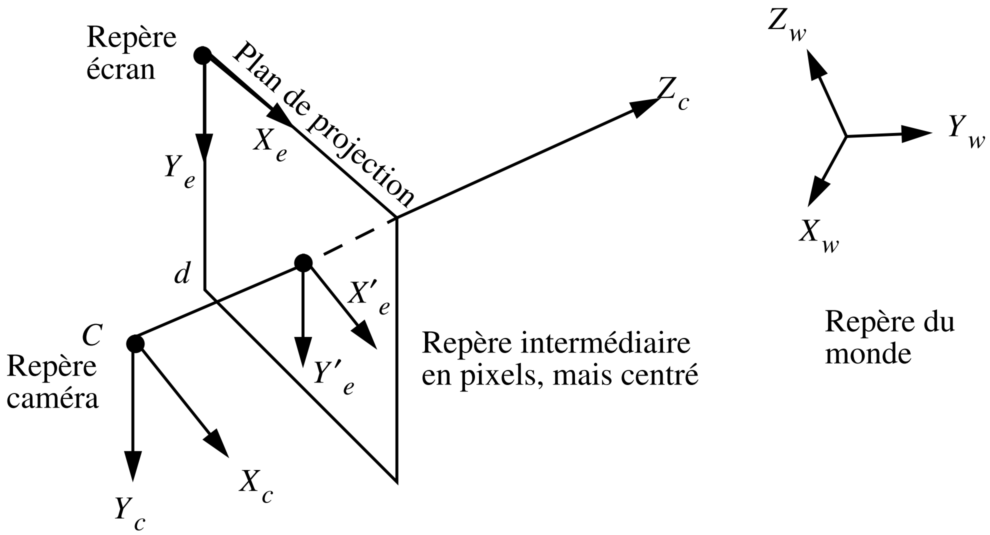


Figure 19 : Les différents repères

Supposons le plan de projection normal à l'axe des Z. Il se trouve à la distance $z = d$ du centre optique de la caméra C (voir Figure 19) :
En projection parallèle, le plan de projection se trouve à la distance $d = \infty$ du centre optique.

Problème : nous désirons calculer les coordonnées d'un point $P^{(w)} = [x, y, z]$ du monde, dans le repère écran situé en haut à gauche du plan de projection, dont les unités sont en pixels.

Nous avons étudié dans les chapitres précédents les méthodes permettant de passer d'un repère à un autre. Nous supposerons donc que la transformation $M_{c \leftarrow w}$ permettant de passer du repère du monde au repère caméra est connue. Le problème se ramène donc à calculer la projection d'un point $P^{(c)}$ en un point $P^{(e)}$ sur le plan de projection.

12.3.2. Projection perspective

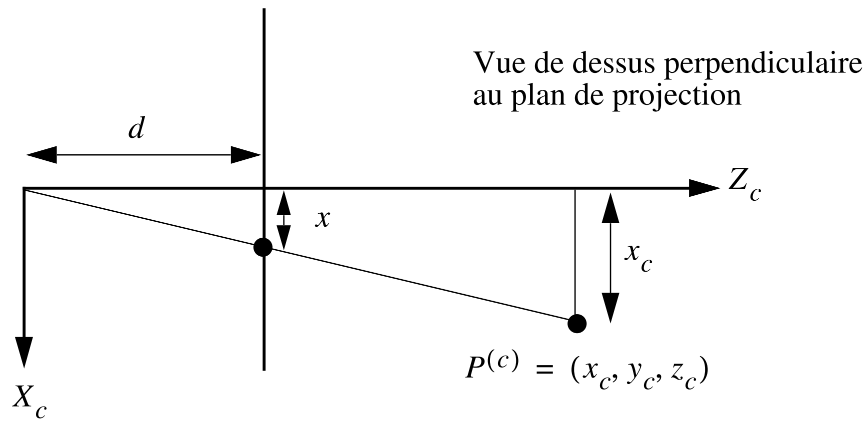


Figure 20 : Calcul de la projection d'un point sur le plan de projection

D'après Thalès, on a $x/d = x_c/z_c$. De même, il suffit d'imaginer la Figure 20 vue de côté pour deviner qu'on a aussi $y/d = y_c/z_c$.

En multipliant par d on obtient :

$$x = \frac{x_c}{z_c/d} \text{ et } y = \frac{y_c}{z_c/d} \qquad \text{(EQ 72)}$$

Cette transformation peut s'exprimer sous forme matricielle :

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \qquad \text{(EQ 73)}$$

En effet, multiplier un point $P = [x, y, z, 1]^T$ par M_p donne le point homogène :

$$P' = M_p \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} \qquad \text{(EQ 74)}$$

En divisant par z/d on obtient $[x/(z/d), y/(z/d), d, 1]^T$ qui correspond en coordonnées 2-D homogènes au point $[x/z, y/z, 1]^T$. On a juste enlevé la 4$^{\text{ème}}$ coordonnée puis homogénéisé par rapport à d.

Conclusion : la projection d'un point $P = [x, y, z]^T$ exprimé dans le repère caméra est égale au point $P' = [x/z, y/z, 1]$ (toujours dans le repère caméra).

12.3.3. Autre formulation, projection parallèle

On place le plan de projection en $z = 0$ et le centre optique de la caméra en $z = -d$.

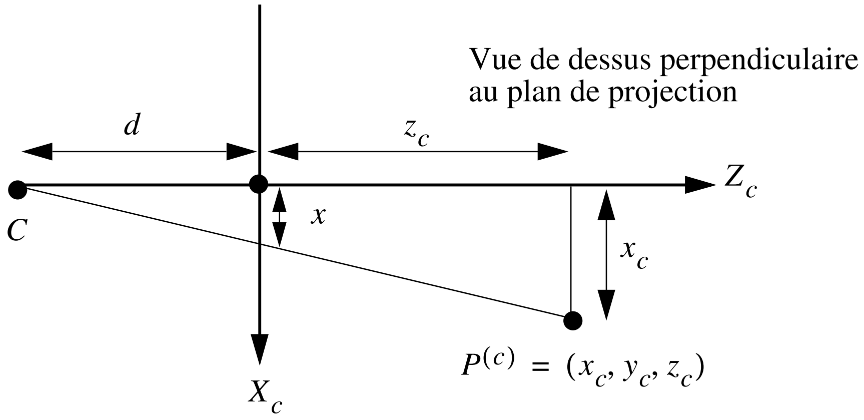


Figure 21 : Autre formulation de la projection perspective

D'après la figure, on a :

$$x/d = x_c/(z_c + d) \quad \text{et} \quad y/d = y_c/(z_c + d) \tag{EQ 75}$$

En multipliant par d on obtient :

$$x = \frac{x_c}{(z_c/d) + 1} \quad \text{et} \quad y = \frac{y_c}{(z_c/d) + 1} \tag{EQ 76}$$

La matrice de projection correspondante est :

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \tag{EQ 77}$$

Cette formulation permet à d de tendre vers l'infini. La projection sur un plan situé en $z = 0$ est évidente: la direction de projection est celle de l'axe des Z.

$$M_{parallele} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{EQ 78}$$

$$P' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \quad \text{(EQ 79)}$$

Comme on le voit, la projection parallèle est un cas particulier de la projection perspective.

12.3.4. Passage en pixels pour visualisation à l'écran

Nous avons vu comment calculer la projection d'un point sur l'écran. Cependant, ses coordonnées sont toujours dans le système d'unités utilisé par le repère caméra. Pour afficher ce point sur l'écran il faut (1) transformer ses coordonnées en pixels et (2) le translater pour l'exprimer non pas dans un repère centré au milieu de l'écran, mais dans le repère (X_e, Y_e) qui se trouve en haut à gauche de l'écran (voir figure Figure 19).

Soit α_x et α_y les ouvertures horizontales et verticales de la caméra, r_x et r_y ses résolutions (en pixels) horizontales et verticales. Pour régler la focale de la caméra, on peut soit faire varier la distance focale d, soit fixer cette focale et faire varier les angles α_x et α_y, le résultat sera équivalent. Pour simplifier les équations, on fixe $d = 1$.

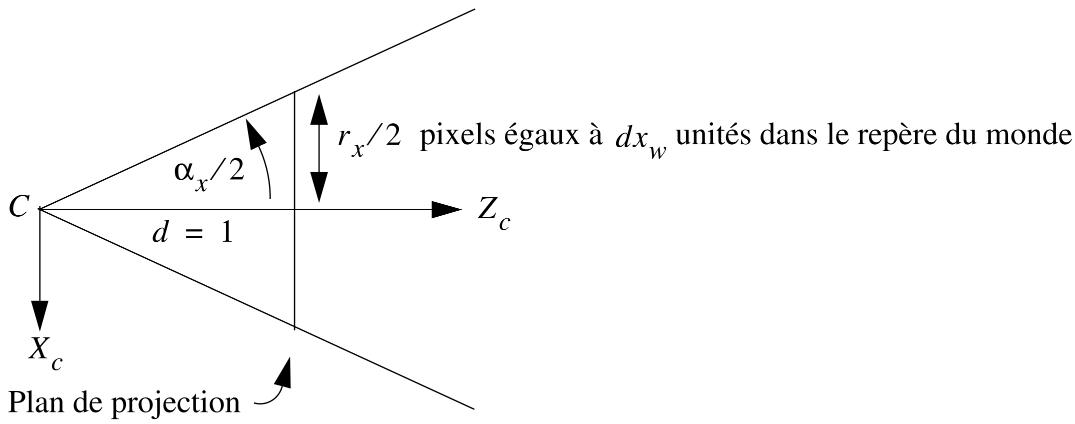


Figure 22 : passage du repère caméra au repère écran, vue de dessus

Sur la Figure 22, d'après Thalès on a :

$$dx_w = \tan(\alpha_x/2) \quad \text{(EQ 80)}$$

Rappelons que dx_w et $r_x/2$ représentent la même distance (voir figure) mais le premier est exprimé en unités quelconques, le second en pixels. Nous cherchons le facteur d'échelle s_x tel que :

$$r_x/2 = s_x \cdot dx_w \quad \text{(EQ 81)}$$

En substituant on a :

$$s_x = \frac{r_x}{2 \cdot dx_w} = \frac{r_x}{2 \cdot \tan(\alpha_x/2)} \qquad \text{(EQ 82)}$$

De la même manière, on peut démontrer que :

$$s_y = \frac{r_y}{2 \cdot dy_w} = \frac{r_y}{2 \cdot \tan(\alpha_y/2)} \qquad \text{(EQ 83)}$$

La matrice de changement d'échelle permettant de passer des unités caméra vers des pixels est égale à :

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 84)}$$

Une fois le point en pixels, il faut encore le translater vers le repère écran qui n'est pas centré (voir Figure 19 page 34). Il faut donc le translater de $[r_x/2, r_y/2]$.

La matrice de passage vers le repère écran est donc égale à :

$$M = T \cdot S = \begin{bmatrix} 1 & 0 & r_x/2 \\ 0 & 1 & r_y/2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & r_x/2 \\ 0 & s_y & r_y/2 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(EQ 85)}$$

$$P' = M \cdot P = \begin{bmatrix} s_x & 0 & r_x/2 \\ 0 & s_y & r_y/2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \cdot x + r_x/2 \\ s_y \cdot y + r_y/2 \\ 1 \end{bmatrix} \qquad \text{(EQ 86)}$$

Cette matrice dépend uniquement de 4 paramètres de la caméra :

- α_x et α_y qui interviennent dans le calcul de s_x et s_y. Voir équations (EQ 82) et (EQ 83).
- r_x et r_y, les résolutions en X et en Y de la caméra. Dans la pratique, ces deux variables seront égales à la largeur et à la hauteur (en pixels) de la fenêtre graphique dans laquelle on veut visualiser une scène.