

1 Introduction aux Transformations de Modélisation

Les transformations géométriques sont des opérations fondamentales en infographie. Elles permettent de manipuler la position, l'orientation, la taille et la forme des objets dans une scène 2D ou 3D.

1.1 Définition et objectif

Une transformation de modélisation est une fonction mathématique qui modifie les coordonnées des points définissant un objet. L'objectif principal est de positionner et d'arranger les objets dans une scène virtuelle (World Space) à partir de leurs définitions locales (Object Space). Elles sont essentielles pour créer des scènes complexes, animer des objets, et définir le point de vue de l'observateur.

1.2 Lien avec le pipeline graphique

Les transformations de modélisation constituent la première étape majeure du pipeline graphique. Après avoir défini la géométrie d'un objet dans son propre système de coordonnées (Object Space), ces transformations le placent dans la scène globale (World Space). Les étapes suivantes du pipeline incluent l'illumination, la transformation de vue, le clipping, la projection, et la rasterisation pour finalement afficher l'image à l'écran.

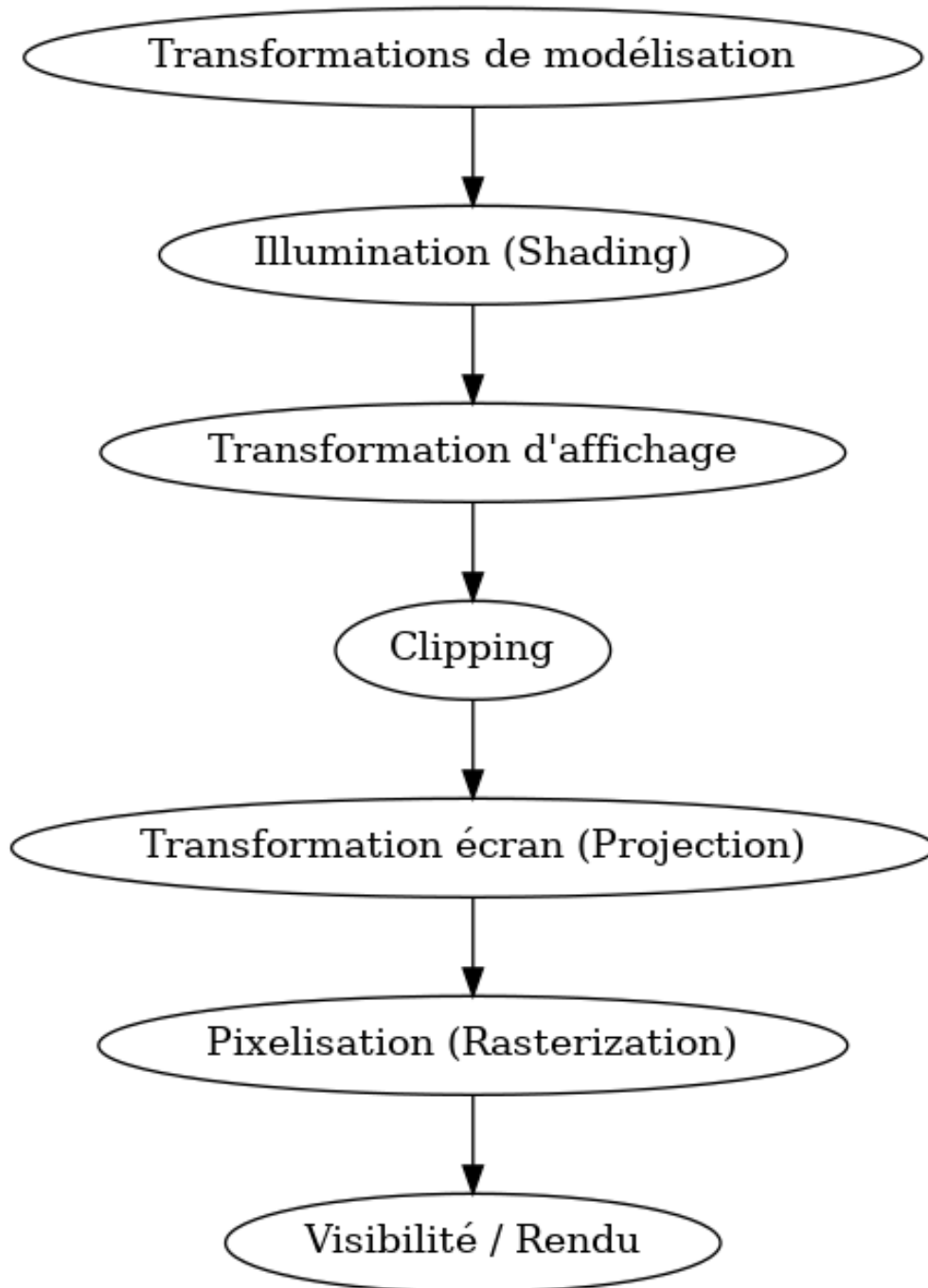


Figure 1: Étapes principales du pipeline graphique.

1.3 Transformation spatiale

Une transformation spatiale $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est une fonction qui assigne à chaque point de l'espace un nouvel emplacement. Nous nous concentrons principalement sur :

- Les transformations **linéaires** : rotation, mise à l'échelle, cisaillement.
- La transformation **non-linéaire** : translation.

L'utilisation de coordonnées homogènes permettra de représenter toutes ces transformations sous forme de multiplication matricielle.

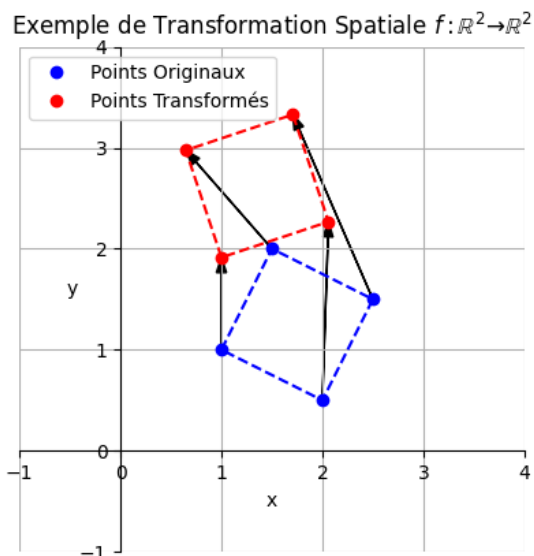


Figure 2: Illustration d'une transformation spatiale assignant de nouvelles positions aux points.

2 Rappels : Transformation Linéaire

Définition 2.1. Une application $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ est dite **linéaire** si elle satisfait les deux propriétés suivantes pour tous vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ et tout scalaire $c \in \mathbb{R}$:

1. Additivité : $f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$
2. Homogénéité : $f(c\mathbf{u}) = cf(\mathbf{u})$

2.1 Interprétation géométrique

Géométriquement, une transformation linéaire préserve certaines propriétés de l'espace :

- Elle transforme les lignes droites en lignes droites (ou en un point si la transformation est dégénérée).
- Elle préserve l'origine : $f(\mathbf{0}) = \mathbf{0}$.
- Elle préserve le parallélisme des lignes.

Les rotations et les mises à l'échelle centrées à l'origine sont des exemples de transformations linéaires. La translation ne l'est pas, car elle ne préserve pas l'origine ($f(\mathbf{0}) \neq \mathbf{0}$).

2.2 Propriétés algébriques

Algébriquement, une transformation linéaire préserve les opérations de base de l'espace vectoriel : l'addition de vecteurs et la multiplication par un scalaire. Cela signifie que l'on peut appliquer la transformation avant ou après avoir effectué ces opérations, le résultat sera le même.

Propriété d'additivité : $f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$

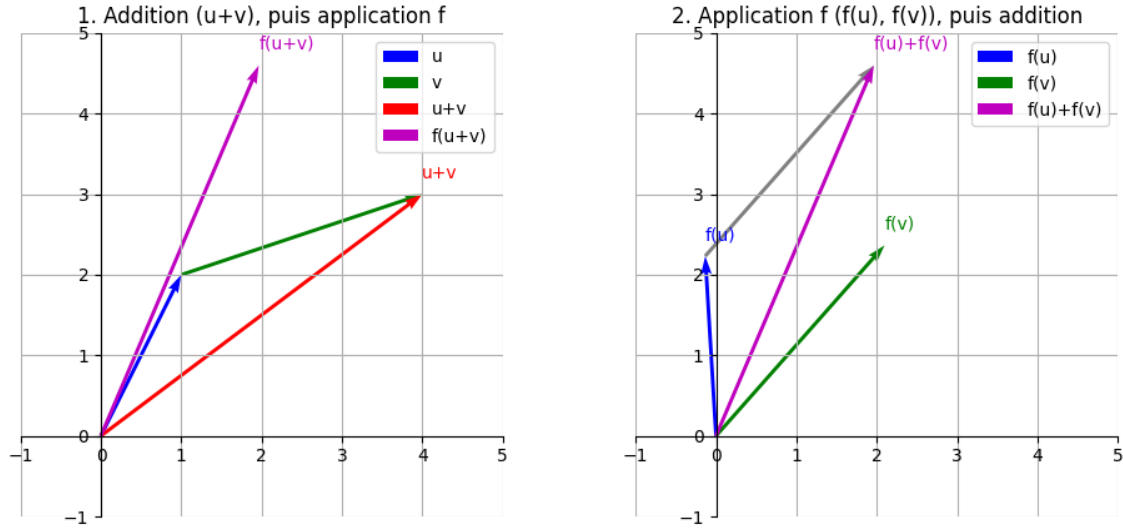


Figure 3: Illustration de la propriété d'additivité d'une transformation linéaire.

Propriété d'homogénéité : $f(c\mathbf{u}) = cf(\mathbf{u})$

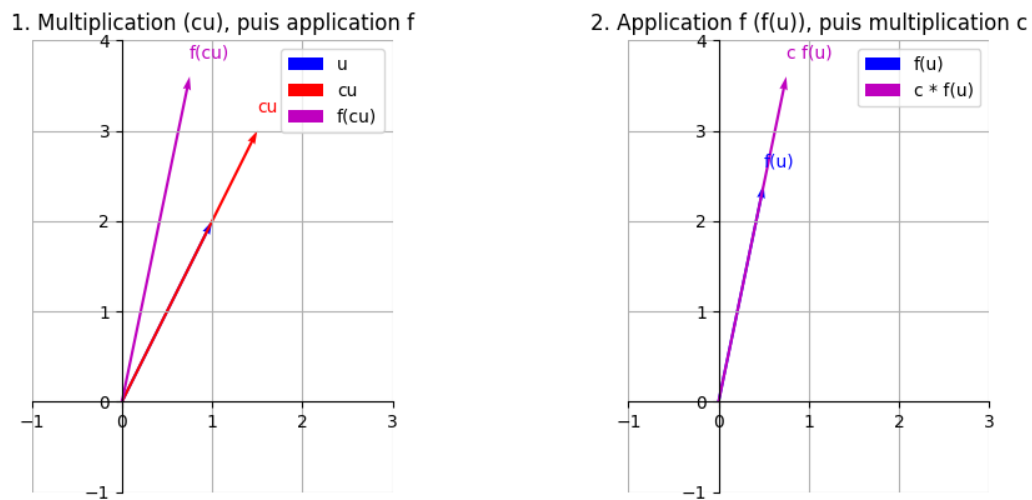


Figure 4: Illustration de la propriété d'homogénéité d'une transformation linéaire.

3 Transformations Spatiales : Utilisations et Types

3.1 Utilisations

Les transformations spatiales sont omniprésentes en infographie et modélisation :

- **Placement d'objets** : Positionner des objets (meubles dans une pièce, personnages dans un décor).
- **Animation** : Déplacer, tourner, ou déformer des objets au fil du temps.

- **Point de vue** : Déplacer et orienter la caméra virtuelle pour définir ce qui est visible.
- **Modélisation hiérarchique** : Construire des objets complexes en assemblant des parties plus simples (e.g., un bras composé d'une épaule, d'un coude, d'une main).
- **Réplication** : Créer des motifs ou des structures répétitives (e.g., une forêt d'arbres identiques placés différemment).
- **Déformation** : Modifier la forme d'un objet (souvent non-linéaire, mais peut inclure des cisaillements linéaires).
- **Projection** : Transformer des coordonnées 3D en coordonnées 2D pour l'affichage (bien que la projection perspective ne soit pas linéaire, elle utilise des matrices et est souvent groupée avec ces transformations).

3.2 Types de transformations fondamentales

Les transformations les plus courantes sont :

- **Translation** : Déplace un objet sans changer son orientation ou sa taille.
- **Rotation** : Tourne un objet autour d'un axe ou d'un point.
- **Mise à l'échelle (Scaling)** : Agrandit ou réduit un objet. Peut être uniforme (même facteur dans toutes les directions) ou non uniforme.
- **Cisaillement (Shear)** : Incline l'objet, comme si on poussait sur une face d'un livre.

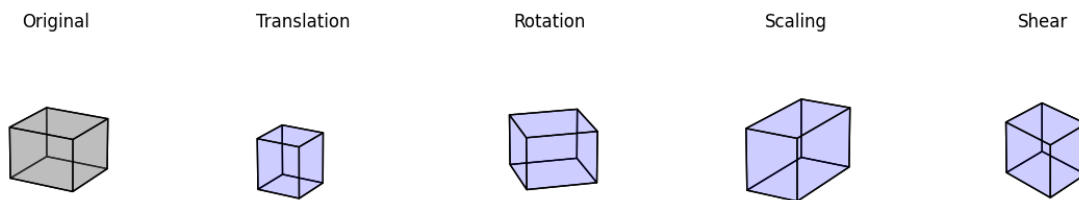


Figure 5: Types de transformations fondamentales appliquées à un cube.

3.3 Représentation des objets

Un objet 3D est typiquement décrit par un ensemble de sommets (vertices) qui définissent sa forme. Appliquer une transformation à un objet revient à appliquer cette transformation à chacun de ses sommets. On utilise la notation vectorielle pour représenter les sommets :

$$\mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

4 Transformations de Base : Rotation

4.1 Propriétés fondamentales

Les rotations sont des transformations linéaires définies par trois propriétés essentielles :

- **Origine fixe** : Le point origine (0,0,0) ne bouge pas.
- **Distances conservées** : La distance entre deux points quelconques reste la même après rotation (isométrie).
- **Orientation conservée** : L'orientation relative des axes est maintenue (pas de "retournement" comme un miroir). Une base orthonormée directe reste directe.

Les deux premières propriétés (origine fixe et conservation des distances) impliquent que les rotations (autour de l'origine) sont des transformations linéaires.

4.2 Rotation autour de l'axe Z

Considérons une rotation d'un angle θ autour de l'axe Z dans un repère orthonormé direct. Un point $\mathbf{p} = (x, y, z)$ est transformé en $\mathbf{p}' = (x', y', z')$.

4.2.1 Calcul vectoriel

En projetant le point sur le plan xy, on voit une rotation 2D standard. La coordonnée z reste inchangée.

$$\begin{aligned}x' &= x \cos \theta - y \sin \theta \\y' &= x \sin \theta + y \cos \theta \\z' &= z\end{aligned}$$

4.2.2 Matrice de rotation

Cette transformation peut s'écrire sous forme matricielle $\mathbf{p}' = R_z(\theta)\mathbf{p}$:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

La matrice $R_z(\theta)$ est la matrice de rotation autour de l'axe Z.

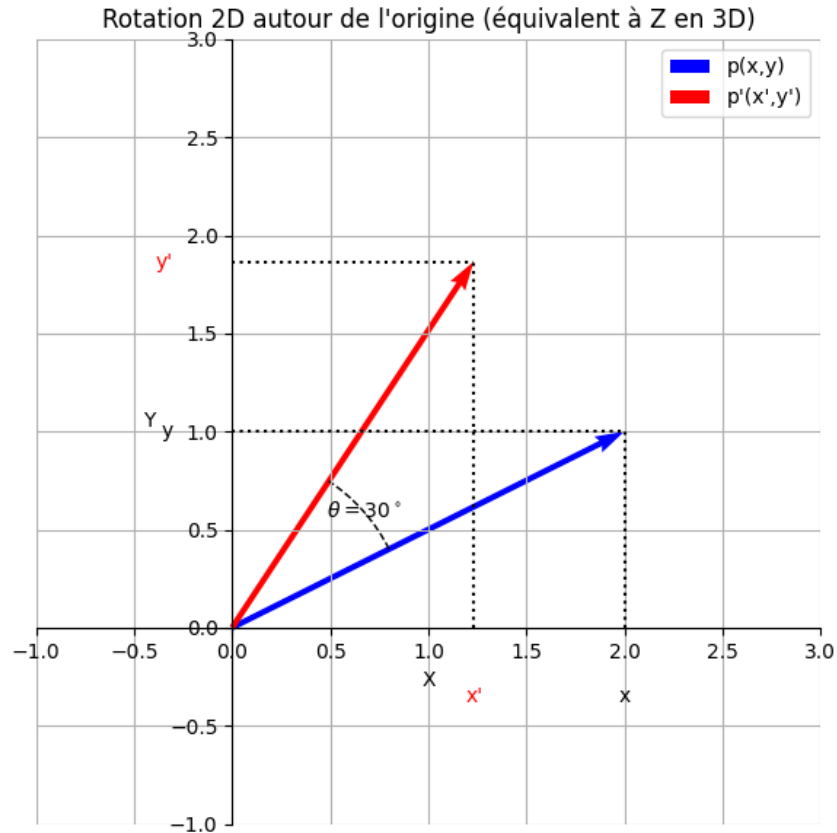


Figure 6: Rotation d'un point p autour de l'axe Z (vue dans le plan xy).

Example 4.1 (Rotation 3D). Calculer les coordonnées du point $\mathbf{p} = (3, 4, 5)$ après une rotation de 90° autour de l'axe Z.

Solution. Ici $\theta = 90^\circ$, donc $\cos \theta = 0$ et $\sin \theta = 1$. La matrice est :

$$R_z(90^\circ) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Appliquons la matrice au point \mathbf{p} :

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} (0)(3) + (-1)(4) + (0)(5) \\ (1)(3) + (0)(4) + (0)(5) \\ (0)(3) + (0)(4) + (1)(5) \end{pmatrix} = \begin{pmatrix} -4 \\ 3 \\ 5 \end{pmatrix}$$

Le point transformé est $\mathbf{p}' = (-4, 3, 5)$. □

5 Transformations de Base : Homothétie (Mise à l'échelle)

L'homothétie, ou mise à l'échelle (scaling), modifie la taille d'un objet. Elle peut être uniforme (même facteur dans toutes les directions) ou non uniforme.

5.1 Multiplication par les facteurs d'échelle

Une homothétie de facteurs S_x, S_y, S_z appliquée à un point $\mathbf{p} = (x, y, z)$ donne le point $\mathbf{p}' = (x', y', z')$ où :

$$x' = S_x \cdot x$$

$$y' = S_y \cdot y$$

$$z' = S_z \cdot z$$

5.2 Matrice d'homothétie

Cette transformation peut s'écrire sous forme matricielle $\mathbf{p}' = S(S_x, S_y, S_z)\mathbf{p}$:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Si $S_x = S_y = S_z = S$, l'homothétie est dite **isotrope**. Sinon, elle est **anisotrope**.

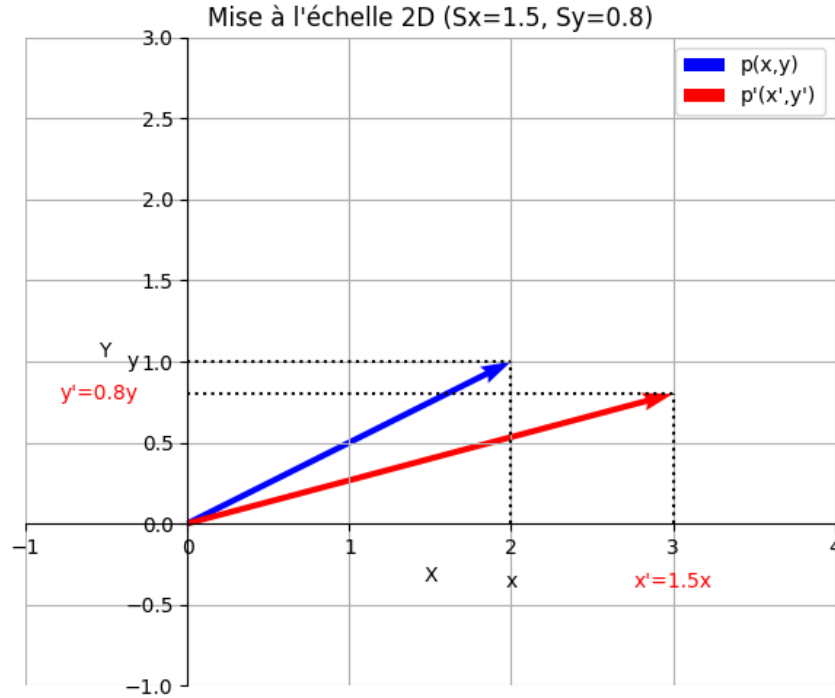


Figure 7: Mise à l'échelle (homothétie) non uniforme d'un point p .

5.3 Linéarité de la mise à l'échelle

La mise à l'échelle (centrée à l'origine) est une transformation linéaire. Elle satisfait les propriétés d'additivité et d'homogénéité.

- $S(\mathbf{u} + \mathbf{v}) = S\mathbf{u} + S\mathbf{v}$

- $S(cu) = c(Su)$

Les figures 8 et 9 illustrent ces propriétés pour la mise à l'échelle.

Linéarité de la mise à l'échelle : Additivité

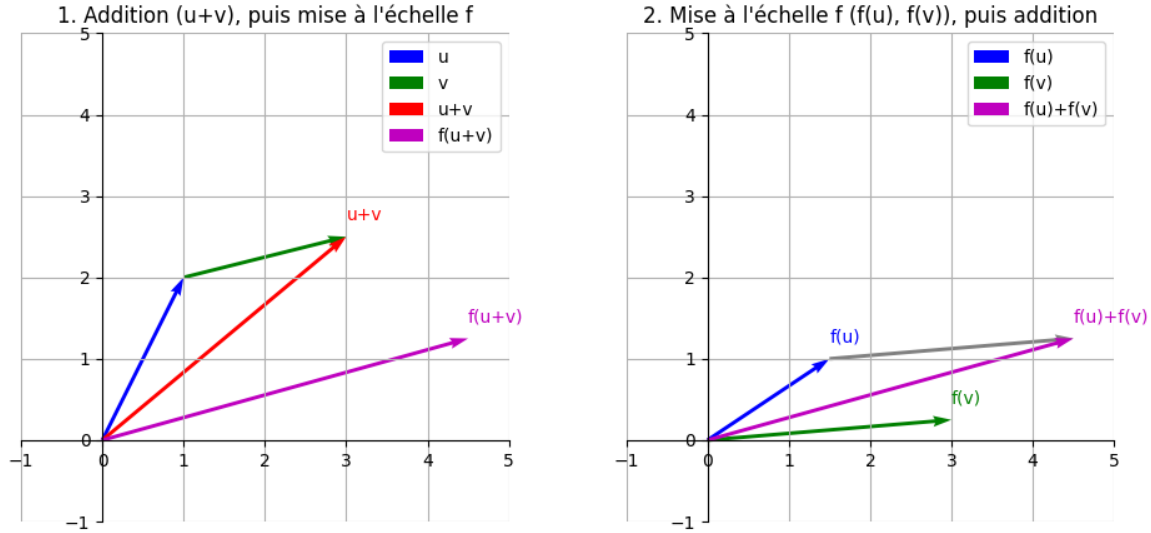


Figure 8: Illustration de l'additivité pour la mise à l'échelle.

Linéarité de la mise à l'échelle : Homogénéité

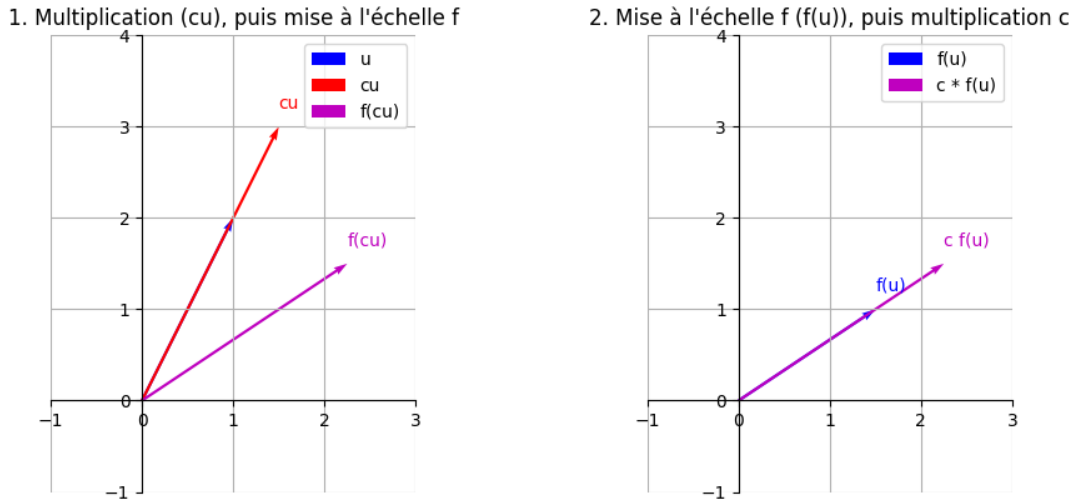


Figure 9: Illustration de l'homogénéité pour la mise à l'échelle.

6 Transformations de Base : Translation

La translation déplace un objet d'une certaine distance dans une direction donnée, sans changer sa taille ni son orientation.

6.1 Ajout des valeurs de translation aux coordonnées

Une translation par un vecteur $\mathbf{T} = (T_x, T_y, T_z)$ appliquée à un point $\mathbf{p} = (x, y, z)$ donne le point $\mathbf{p}' = (x', y', z')$ où :

$$x' = x + T_x$$

$$y' = y + T_y$$

$$z' = z + T_z$$

Ceci peut s'écrire comme une addition vectorielle : $\mathbf{p}' = \mathbf{p} + \mathbf{T}$.

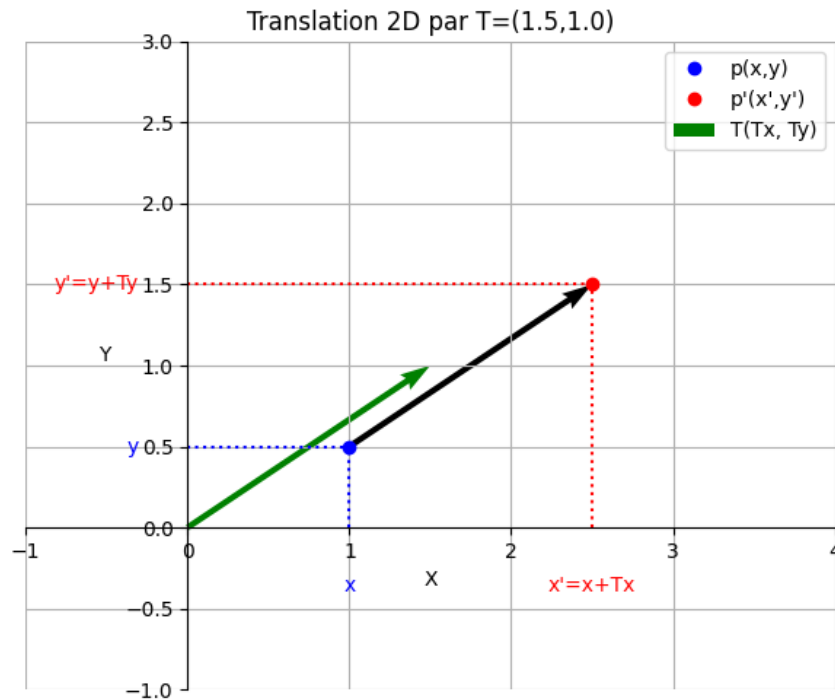


Figure 10: Translation d'un point p par un vecteur \mathbf{T} .

6.2 Notation matricielle

La translation, étant une addition $\mathbf{p}' = \mathbf{p} + \mathbf{T}$, ne peut pas être représentée par une multiplication matricielle $\mathbf{p}' = M\mathbf{p}$ en utilisant des matrices 3x3 et des vecteurs 3D standards. C'est une des motivations principales pour introduire les coordonnées homogènes.

6.3 Non-linéarité de la translation

La translation n'est pas une transformation linéaire car elle ne respecte pas les propriétés d'additivité et d'homogénéité, et surtout, elle ne laisse pas l'origine invariante (sauf si $\mathbf{T} = \mathbf{0}$).

$$f(\mathbf{0}) = \mathbf{0} + \mathbf{T} = \mathbf{T} \neq \mathbf{0} \quad (\text{si } \mathbf{T} \neq \mathbf{0})$$

7 Coordonnées Homogènes

7.1 Utilité

Les coordonnées homogènes sont un système de coordonnées utilisé en géométrie projective et en infographie pour simplifier la représentation et la composition des transformations géométriques. Leur principal avantage est de permettre de représenter **toutes** les transformations affines (y compris la translation) sous forme de **multiplication matricielle unique**.

7.2 Représentation des points

En coordonnées homogènes, un point 3D $\mathbf{p} = (x, y, z)$ est représenté par un vecteur 4D \mathbf{P} :

$$\mathbf{P} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Plus généralement, un point (X, Y, Z, W) avec $W \neq 0$ en coordonnées homogènes correspond au point cartésien $(X/W, Y/W, Z/W)$. La composante W est appelée coordonnée homogène. Pour les points de l'espace affine, on utilise conventionnellement $W = 1$. Les points avec $W = 0$ représentent des points à l'infini (directions).

7.3 Représentation matricielle unifiée

Avec les coordonnées homogènes, les transformations de rotation, mise à l'échelle et translation peuvent toutes être exprimées par une multiplication avec une matrice 4x4.

- **Translation** $T(T_x, T_y, T_z)$:

$$M_T = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Mise à l'échelle** $S(S_x, S_y, S_z)$:

$$M_S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Rotation** autour de Z $R_z(\theta)$:

$$M_{Rz} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Matrices de rotation autour de X et Y sont similaires).

Un point \mathbf{p} transformé devient \mathbf{p}' en calculant $\mathbf{P}' = M\mathbf{P}$, où M est la matrice 4x4 de la transformation et \mathbf{P}, \mathbf{P}' sont les représentations homogènes de \mathbf{p}, \mathbf{p}' .

$$\mathbf{P}' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

7.4 Structure de la matrice 4x4

La matrice 4x4 générale pour les transformations affines a la structure suivante :

$$M_H = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix}$$

Où $\mathbf{R}_{3 \times 3}$ représente la partie linéaire (rotation, échelle, cisaillement) et $\mathbf{T}_{3 \times 1}$ représente la translation. Les transformations de projection (perspective) utiliseront également la dernière ligne de la matrice.

Matrice Affine 4x4			
Rotation, Echelle, Cisaillement (Linéaire) 3x3			Translation 3x1
0	0	0	1

Figure 11: Structure typique d'une matrice de transformation affine 4x4 en coordonnées homogènes.

8 Manipulations Géométriques

En utilisant les coordonnées homogènes et les matrices 4x4, toute transformation affine (et même projective) peut être représentée par une unique matrice M .

- Soit un point \mathbf{p} , représenté par \mathbf{P} en coordonnées homogènes.
- Soit une transformation géométrique définie par la matrice M .
- Le point transformé \mathbf{p}' , représenté par \mathbf{P}' , est obtenu par :

$$\mathbf{P}' = M \cdot \mathbf{P}$$

Ceci inclut les transformations de Translation, Rotation, Changement d'échelle, Projection, Symétries, Affinités orthogonales, etc.

Exemple 8.1 (Translation via Matrice 4x4). Calculer la translation du point $\mathbf{p} = (2, 3, 4)$ par le vecteur $\mathbf{T} = (5, -1, 2)$ en utilisant les coordonnées homogènes.

Solution. Le point en coordonnées homogènes est $\mathbf{P} = (2, 3, 4, 1)^T$. La matrice de translation est :

$$M_T = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Calculons $\mathbf{P}' = M_T \mathbf{P}$:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 1(2) + 0(3) + 0(4) + 5(1) \\ 0(2) + 1(3) + 0(4) + (-1)(1) \\ 0(2) + 0(3) + 1(4) + 2(1) \\ 0(2) + 0(3) + 0(4) + 1(1) \end{pmatrix} = \begin{pmatrix} 7 \\ 2 \\ 6 \\ 1 \end{pmatrix}$$

Le point translaté est $\mathbf{p}' = (7, 2, 6)$. □

9 Rotation (Angles d'Euler)

Une rotation 3D quelconque autour de l'origine peut être décomposée en une séquence de trois rotations autour des axes principaux (X, Y, Z). Les angles de ces rotations sont appelés angles d'Euler. Une convention courante est Z-Y-X. La matrice de rotation globale R est obtenue en multipliant les matrices des rotations individuelles :

$$R = R_x(\alpha)R_y(\beta)R_z(\gamma)$$

(Attention, l'ordre de multiplication est important et dépend de la convention choisie : axes fixes ou axes mobiles). La matrice de rotation 4x4 M_R s'obtient en plaçant la matrice de rotation 3x3 R dans le coin supérieur gauche et en complétant avec des 0 et un 1 :

$$M_R = \begin{pmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix}$$

9.1 Règles de construction des matrices de rotation de base

Pour construire la matrice 3x3 (et par extension 4x4) d'une rotation d'angle θ autour d'un axe principal (X, Y ou Z) :

1. La ligne et la colonne correspondant à l'axe de rotation contiennent un 1 sur la diagonale et des 0 ailleurs. (En 4x4, le 1 de la coordonnée homogène est aussi présent).
2. Pour les deux autres axes, on place $\cos(\theta)$ sur la diagonale.
3. On place $\pm \sin(\theta)$ dans les positions hors diagonale pour "compléter le carré" 2x2 correspondant à ces deux axes.
4. Le signe du sinus : pour une rotation anti-horaire (sens direct), le sinus est positif lorsqu'on passe de l'axe i à l'axe j si (i, j, k) forme une permutation directe des axes (e.g., X vers Y pour rotation Z, Y vers Z pour rotation X, Z vers X pour rotation Y). Le signe est négatif pour le terme symétrique ($M_{ji} = -M_{ij}$). Une règle mnémotechnique issue des slides est : "Sur la ligne suivant celle de l'axe de rotation, le sinus est précédé d'un signe '+'".

Matrices de base (3x3) :

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

L'image des slides montre la composition $R = R_z R_y R_x$ (attention, l'ordre peut varier). La matrice finale R (partie 3x3) s'obtient par multiplication de ces matrices de base. L'image illustre visuellement comment les termes cos et sin des angles $\theta_x, \theta_y, \theta_z$ (correspondant aux rotations autour de x, y, z) se combinent dans la matrice résultante 4x4.

10 Homothétie et Affinités Orthogonales

10.1 Homothétie isotrope

C'est une mise à l'échelle uniforme avec le même facteur S dans toutes les directions ($S_x = S_y = S_z = S$). La matrice 4x4 est :

$$M_{S_{iso}} = \begin{pmatrix} S & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

L'application à un point \mathbf{P} donne $\mathbf{P}' = M_{S_{iso}} \mathbf{P} = (Sx, Sy, Sz, 1)^T$.

10.2 Affinités orthogonales

Ce sont des mises à l'échelle non uniformes le long des axes principaux. Elles sont parfois appelées "homothéties axiales" ou simplement "mises à l'échelle non uniformes". La matrice générale est M_S vue précédemment.

$$M_S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Les slides montrent des cas particuliers où l'échelle n'est appliquée que sur un ou deux axes (affinité d'axe X par rapport au plan yOz, etc.) :

- Affinité d'axe X (mise à l'échelle en X seulement) : $S_y = 1, S_z = 1$

$$M_{S_x} = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Affinité d'axe Y (mise à l'échelle en Y seulement) : $S_x = 1, S_z = 1$

$$M_{S_y} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Affinité d'axe Z (mise à l'échelle en Z seulement) : $S_x = 1, S_y = 1$

$$M_{S_z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

11 Glissement (Shear)

Le glissement, ou cisaillement (shear), est une transformation qui incline les objets. Les coordonnées d'un point sont modifiées proportionnellement à une ou plusieurs autres coordonnées.

Définition 11.1. Un **glissement parallèle à un axe** modifie la coordonnée de cet axe en fonction d'une autre coordonnée. Par exemple, un glissement parallèle à l'axe X avec un rapport k par rapport à Y modifie la coordonnée x : $x' = x + k \cdot y$. Les autres coordonnées (y, z) restent inchangées.

11.1 Matrice de glissement

- Glissement de X par Y (rapport k) : $x' = x + ky$

$$M_{Sh_{XY}} = \begin{pmatrix} 1 & k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Glissement de X par Z (rapport k) : $x' = x + kz$

$$M_{Sh_{XZ}} = \begin{pmatrix} 1 & 0 & k & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Glissement de Y par X (rapport k) : $y' = y + kx$

$$M_{Sh_{YX}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ k & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Et ainsi de suite pour les autres combinaisons YZ, ZX, ZY).

L'image des slides montre un glissement 2D parallèle à x , avec rapport k :

$$M = \begin{pmatrix} 1 & k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Elle montre aussi un glissement parallèle à x , de rapport k , dont l'intensité dépend de la distance à une ligne de base $y = y_{ref}$: $x' = x + k(y - y_{ref}) = x + ky - ky_{ref}$. Cela combine un cisaillement ($x' = x + ky$) et une translation ($x'' = x' - ky_{ref}$). La matrice devient :

$$M = \begin{pmatrix} 1 & k & 0 & -k \cdot y_{ref} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

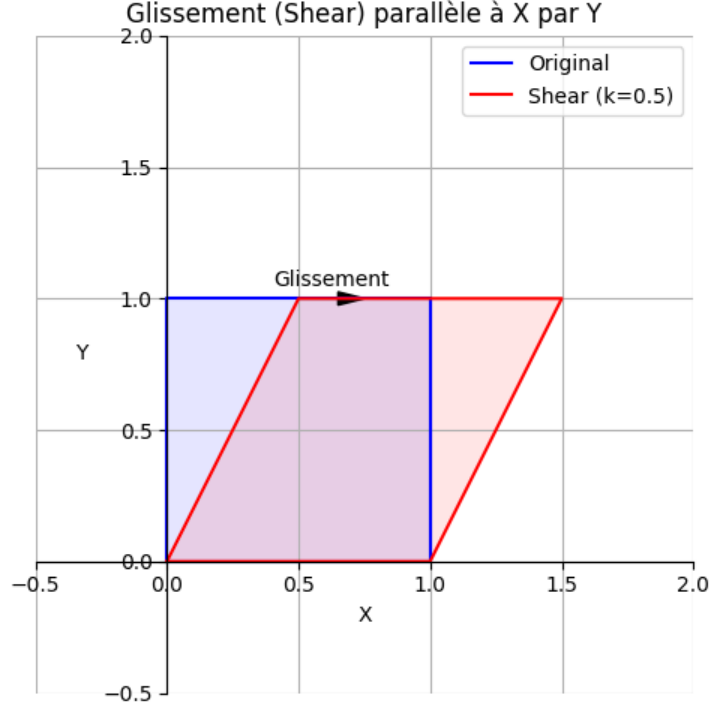


Figure 12: Exemple de glissement (shear) parallèle à l'axe X, proportionnel à Y.

12 Composition de Transformations

Souvent, on souhaite appliquer plusieurs transformations successivement à un objet (par exemple, une mise à l'échelle, puis une rotation, puis une translation). Grâce aux coordonnées homogènes, la composition de transformations se réduit à la multiplication des matrices correspondantes. Si on applique d'abord la transformation M_1 , puis M_2 , puis M_3 à un point \mathbf{P} , le point final \mathbf{P}' est :

$$\mathbf{P}' = M_3(M_2(M_1\mathbf{P})) = (M_3M_2M_1)\mathbf{P}$$

La matrice de transformation composite $M = M_3M_2M_1$ peut être pré-calculée et appliquée directement à tous les points de l'objet.

12.1 Ordre d'application

Important : La multiplication de matrices n'est pas commutative ($M_1M_2 \neq M_2M_1$ en général). L'ordre dans lequel les transformations sont appliquées est crucial. La convention est que les transformations sont appliquées dans l'ordre où elles sont multipliées de **droite à gauche**. Dans l'expression $M = M_3M_2M_1$, la transformation M_1 est appliquée en premier, suivie de M_2 , puis M_3 .

12.2 Exemple : Scale puis Translate vs Translate puis Scale

Considérons un point $\mathbf{p} = (1,1)$ (donc $\mathbf{P} = (1,1,1)^T$ en 2D homogène). Appliquons une mise à l'échelle $S(2,2)$ et une translation $T(3,1)$. **Cas 1 : Scale(2,2) puis Translate(3,1)** La matrice composite est $M = M_T \cdot M_S$.

$$M_S = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M_T = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M = M_T M_S = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Appliquons à \mathbf{P} :

$$\mathbf{P}' = M\mathbf{P} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2+3 \\ 2+1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \\ 1 \end{pmatrix}$$

Le point final est $\mathbf{p}' = (5, 3)$. **Cas 2 : Translate(3,1) puis Scale(2,2)** La matrice composite est $M' = M_S \cdot M_T$.

$$M' = M_S M_T = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Appliquons à \mathbf{P} :

$$\mathbf{P}'' = M'\mathbf{P} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2+6 \\ 2+2 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 1 \end{pmatrix}$$

Le point final est $\mathbf{p}'' = (8, 4)$. Les résultats sont différents, illustrant la non-commutativité.

Non-commutativité de la composition des transformations

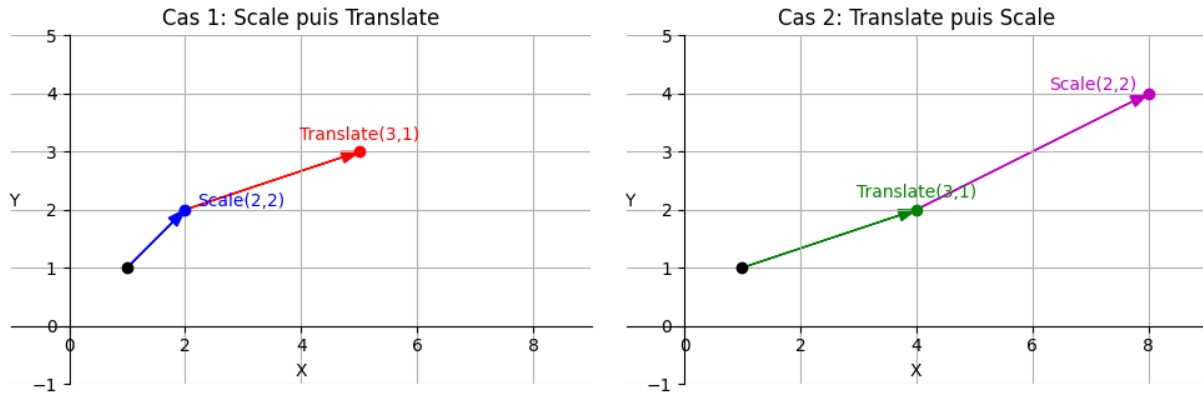


Figure 13: Illustration de la non-commutativité : l'ordre des transformations importe.

12.3 Composition pour transformations complexes

Les transformations élémentaires (Translation, Rotation, Echelle) sont souvent définies par rapport à l'origine et aux axes globaux. Pour effectuer une rotation ou une mise à l'échelle par rapport à un point arbitraire P ou un axe arbitraire, on utilise une composition :

1. Translater pour amener le point P (ou l'axe) à l'origine. $T(-P)$
2. Effectuer la rotation ou la mise à l'échelle voulue (par rapport à l'origine). $R(\theta)$ ou $S(s)$
3. Translater inversement pour remettre le point P (ou l'axe) à sa position initiale. $T(P)$

La matrice composite est $M = T(P) \cdot R(\theta) \cdot T(-P)$.

Exemple 12.1 (Rotation autour d'un point P). Trouver la matrice qui effectue une rotation d'angle θ_z autour de l'axe Z passant par le point $P = (T_x, T_y, T_z)$.

Solution. La séquence est :

1. Translation $T(-T_x, -T_y, -T_z)$ pour amener P à l'origine.
2. Rotation $R_z(\theta_z)$ autour de l'axe Z (qui passe maintenant par l'origine "locale" de l'objet translaté).
3. Translation $T(T_x, T_y, T_z)$ pour revenir.

La matrice composite M est :

$$M = T(T_x, T_y, T_z) \cdot R_z(\theta_z) \cdot T(-T_x, -T_y, -T_z)$$

En multipliant les matrices 4x4 correspondantes :

$$M = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 & 0 \\ \sin \theta_z & \cos \theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le calcul donne (comme montré dans les slides) :

$$M = \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 & -T_x \cos \theta_z + T_y \sin \theta_z + T_x \\ \sin \theta_z & \cos \theta_z & 0 & -T_x \sin \theta_z - T_y \cos \theta_z + T_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Note: la coordonnée Z de translation T_z n'intervient pas car la rotation est autour de Z). □

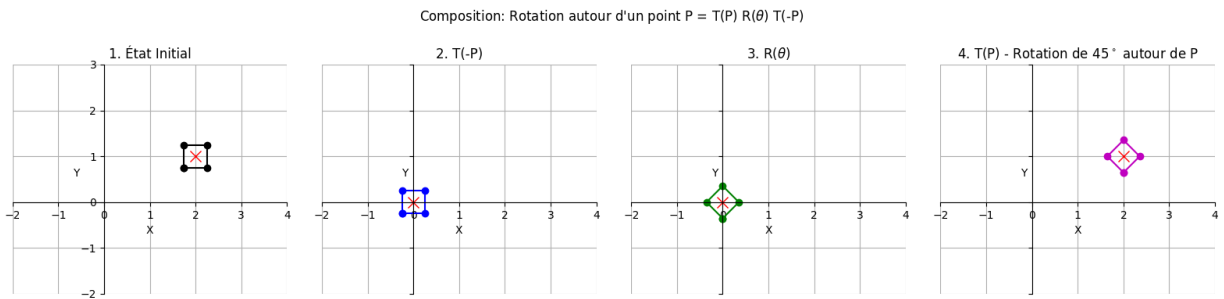


Figure 14: Étapes de la rotation d'un objet autour d'un point pivot P.

13 Modélisation Hiérarchique

Pour représenter des objets complexes composés de plusieurs parties (par exemple, un personnage, un véhicule, un bâtiment), on utilise souvent une structure hiérarchique, typiquement un arbre ou un graphe orienté acyclique (DAG).

13.1 Définition et principe

- La scène est organisée comme une structure arborescente (ou DAG).

- Les nœuds internes représentent des groupes ou des sous-objets.
- Les nœuds feuilles (terminaux) représentent des primitives géométriques simples (cubes, sphères, polygones...).
- Chaque nœud (sauf la racine) est défini par rapport au système de coordonnées de son nœud parent.
- Chaque arc (ou lien parent-enfant) est associé à une matrice de transformation qui positionne l'enfant par rapport au parent.

Le repère associé à la racine est le repère global de la scène (World Space). La transformation absolue d'un objet dans la scène est obtenue en composant (multipliant) toutes les matrices de transformation rencontrées en descendant depuis la racine jusqu'au nœud de l'objet.

13.2 Arbre de scène (Scene Graph)

C'est la structure de données (arbre ou DAG) qui représente la hiérarchie des objets et leurs transformations relatives.

- **Avantages :**
- Facilite la manipulation d'objets complexes (déplacer un bras entier en bougeant juste le nœud "épaule").
- Permet l'instanciation (réutilisation d'une même géométrie à plusieurs endroits avec des transformations différentes).
- Simplifie l'animation (animer les transformations associées aux arcs).
- **Structure DAG :** Si un objet (comme une roue) est utilisé plusieurs fois dans la scène, on peut utiliser un DAG où plusieurs nœuds parents pointent vers le même nœud enfant (la définition de la roue).

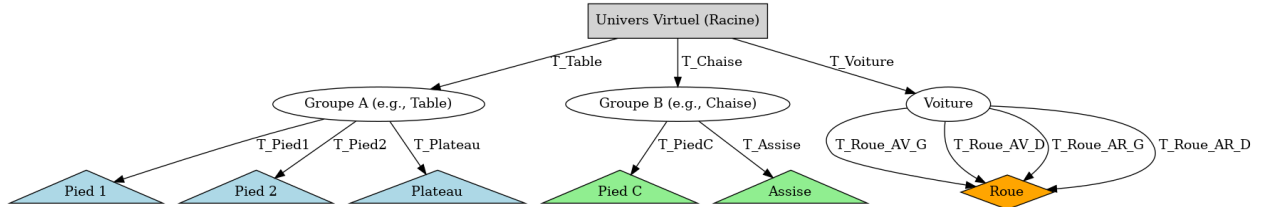


Figure 15: Exemple simplifié d'un graphe de scène (DAG).

13.3 Construction et utilisation

On construit le graphe généralement "top-down" (de la racine vers les feuilles), en décomposant récursivement les objets complexes en parties plus simples. Pour obtenir la transformation globale (World Transform) M_{world} d'un nœud N , on multiplie sa transformation locale M_{local} (associée à l'arc entrant) par la transformation globale de son parent M_{parent_world} :

$$M_{world} = M_{parent_world} \cdot M_{local}$$

On calcule cela récursivement depuis la racine, dont la transformation globale est l'identité.

13.4 Exemple : Modélisation d'une maison

L'image des slides montre un graphe de scène pour une maison simple.

1. Racine (Maison, T_m)
 - (a) Garage (T_g)
 - i. Porte A (T_{ga}) - > Feuille
 - ii. Porte B (T_{gb}) - > Feuille
 - (b) Corps Principal (T_c)
 - i. Façade (T_{fa})
 - A. Fenêtre 1 (T_{f1})
 - B. Fenêtre 2 - > Feuille
 - C. ...
 - D. Fenêtre 5 - > Feuille
 - ii. Toit - > Feuille

Pour trouver la transformation de la Vitre A dans le repère monde, on calcule : $M_{VitreA} = T_m \cdot T_c \cdot T_{fa} \cdot T_{f1} \cdot T_{va}$

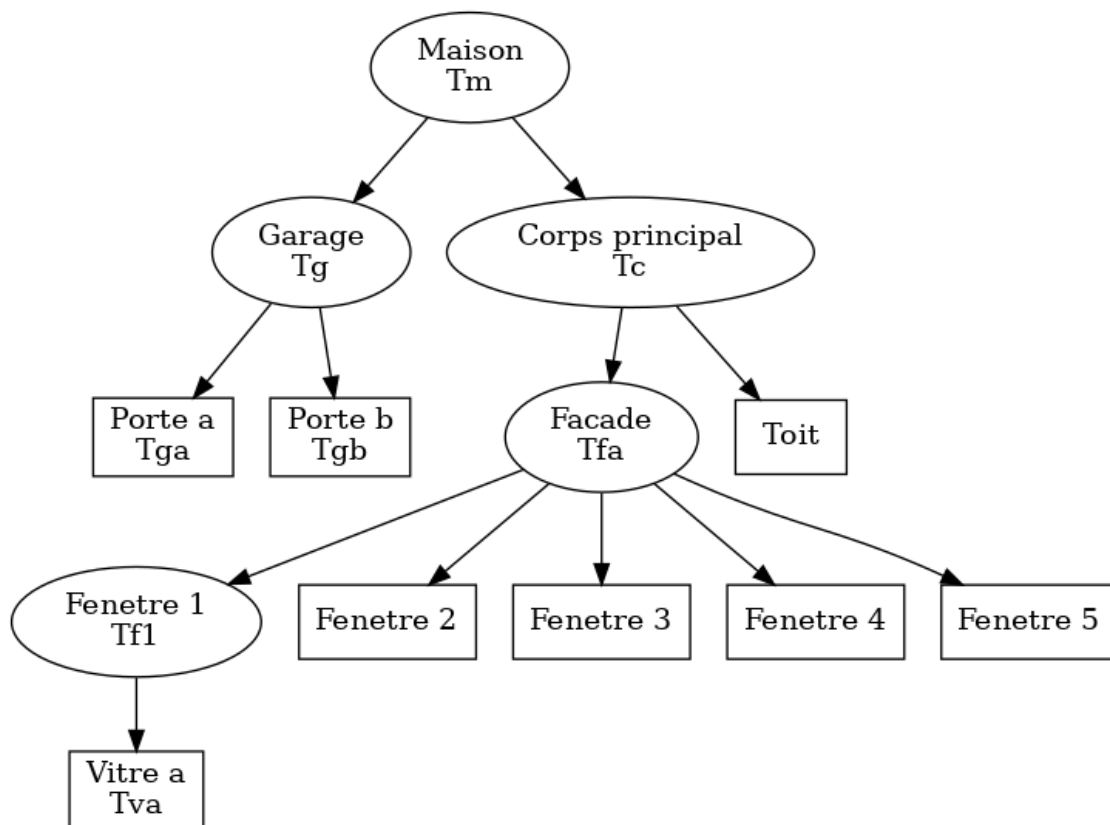


Figure 16: Exemple de graphe de scène pour une maison.

14 Instanciation

L'instanciation est une technique clé en modélisation hiérarchique qui permet de réutiliser la définition géométrique d'un objet (une "instance maîtresse" ou "prototype") à plusieurs endroits dans la scène.

14.1 Principe

Plutôt que de dupliquer la géométrie (sommets, faces) pour chaque copie d'un objet (comme chaque arbre dans une forêt, ou chaque roue sur une voiture), on stocke la géométrie une seule fois. Dans le graphe de scène, les nœuds qui représentent les copies de cet objet contiennent simplement une référence (un "pointeur") vers la géométrie maîtresse. Chaque arc menant à une instance peut avoir sa propre matrice de transformation, permettant de positionner, orienter et dimensionner chaque copie différemment.

14.2 Utilisation dans l'arbre de scène

Dans un DAG, plusieurs nœuds "parents" peuvent pointer vers le même nœud "enfant" qui définit la géométrie. Chaque chemin depuis la racine vers cet enfant via différents parents correspondra à une instance différente de l'objet, avec une transformation globale potentiellement unique.

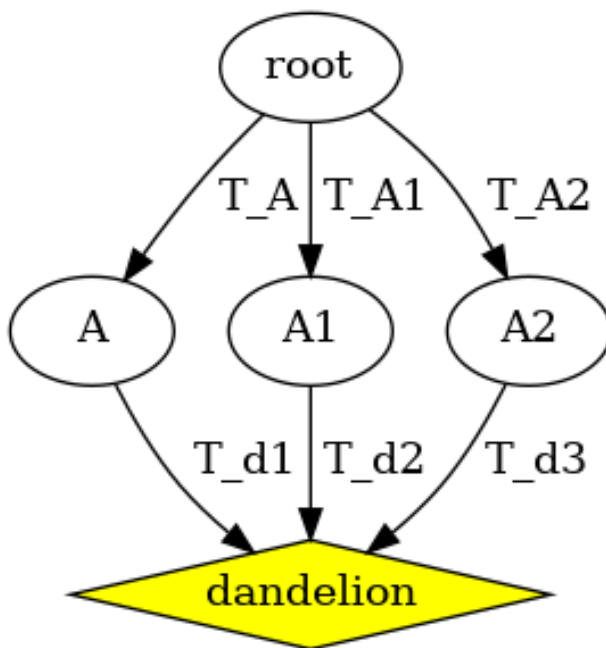


Figure 17: Graphe de scène (DAG) illustrant l'instanciation : plusieurs nœuds (A, A1, A2) utilisent la même définition géométrique (dandelion) via des transformations différentes (Td1, Td2, Td3).

15 Changement de Repère

Il est souvent nécessaire d'exprimer les coordonnées d'un point ou la description d'un objet dans différents systèmes de coordonnées (repères). Par exemple :

- Repère objet (Object Space) : Coordonnées locales définissant la géométrie de l'objet.
- Repère scène (World Space) : Repère global où tous les objets sont positionnés.
- Repère caméra (View Space / Camera Space) : Repère lié à l'observateur.

Le passage d'un repère à un autre est une transformation géométrique.

15.1 Définition et objectif

Transformer les coordonnées d'un point \mathbf{p} exprimées dans un repère R_2 en coordonnées équivalentes dans un repère R_1 .

15.2 Transformation des coordonnées

Soit M la matrice 4×4 qui transforme les vecteurs de base et l'origine du repère R_2 pour les aligner avec ceux de R_1 . Si $\mathbf{P}_2 = (x_2, y_2, z_2, 1)^T$ sont les coordonnées d'un point dans R_2 , alors ses coordonnées $\mathbf{P}_1 = (x_1, y_1, z_1, 1)^T$ dans R_1 sont données par :

$$\mathbf{P}_1 = M \cdot \mathbf{P}_2$$

La matrice M est appelée **matrice de passage** de R_2 à R_1 . Inversement, pour passer de R_1 à R_2 , on utilise la matrice inverse M^{-1} :

$$\mathbf{P}_2 = M^{-1} \cdot \mathbf{P}_1$$

15.3 Matrice de passage

Si le repère R_2 (origine O_2 , axes $\mathbf{u}_2, \mathbf{v}_2, \mathbf{w}_2$) est exprimé dans les coordonnées de R_1 , la matrice de passage M de R_2 vers R_1 est construite comme suit :

$$M = \begin{pmatrix} u_{2x} & v_{2x} & w_{2x} & O_{2x} \\ u_{2y} & v_{2y} & w_{2y} & O_{2y} \\ u_{2z} & v_{2z} & w_{2z} & O_{2z} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Où (u_{2x}, u_{2y}, u_{2z}) sont les coordonnées du vecteur \mathbf{u}_2 dans R_1 , etc., et (O_{2x}, O_{2y}, O_{2z}) sont les coordonnées de l'origine O_2 dans R_1 . Cette matrice M transforme un point dont les coordonnées sont connues dans R_2 pour donner ses coordonnées dans R_1 .

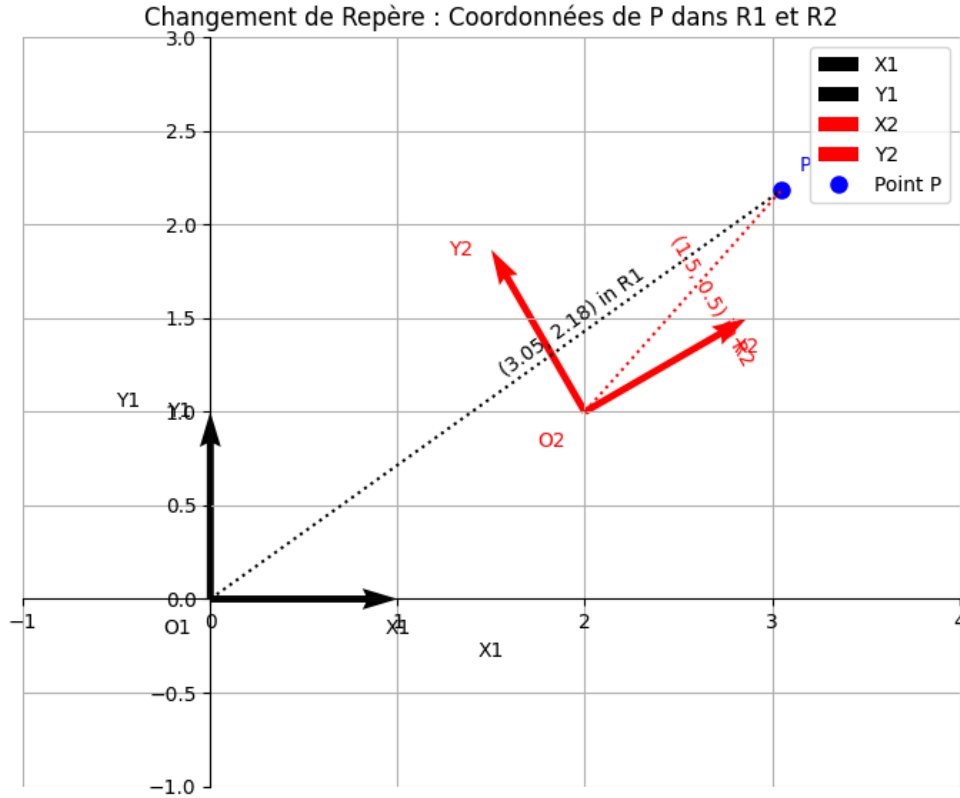


Figure 18: Illustration du changement de repère. Le même point P a des coordonnées différentes dans le repère R_1 (noir) et le repère R_2 (rouge).

16 Projection

La projection est l'étape du pipeline graphique qui transforme les objets 3D de l'espace caméra (ou espace vue) en une représentation 2D sur le plan image. C'est une réduction du nombre de dimensions.

16.1 Introduction

Imaginez regarder une scène 3D à travers une fenêtre. La projection détermine comment cette scène apparaît sur la surface plane de la fenêtre. Les lignes allant de l'œil (ou centre de projection) aux points de l'objet sont appelées **projecteurs**. L'image projetée est formée par les intersections de ces projecteurs avec le **plan de projection** (la fenêtre).

16.2 Types de projection

Il existe deux grandes familles de projections géométriques planes :

- **Projection Perspective** : Le centre de projection (CP) est à une distance finie du plan de projection. Les projecteurs sont des lignes convergentes (comme les rayons de lumière entrant dans un appareil photo ou l'œil). C'est la projection la plus réaliste, car elle simule la façon dont nous percevons la profondeur (les objets éloignés apparaissent plus petits).
- **Projection Parallèle** : Le centre de projection est à l'infini. Les projecteurs sont des lignes parallèles entre elles. Moins réaliste, mais utile en dessin technique, architecture, ou jeux vidéo (vues isométriques) car elle préserve certaines propriétés comme le parallélisme et les rapports de longueurs le long des directions principales.

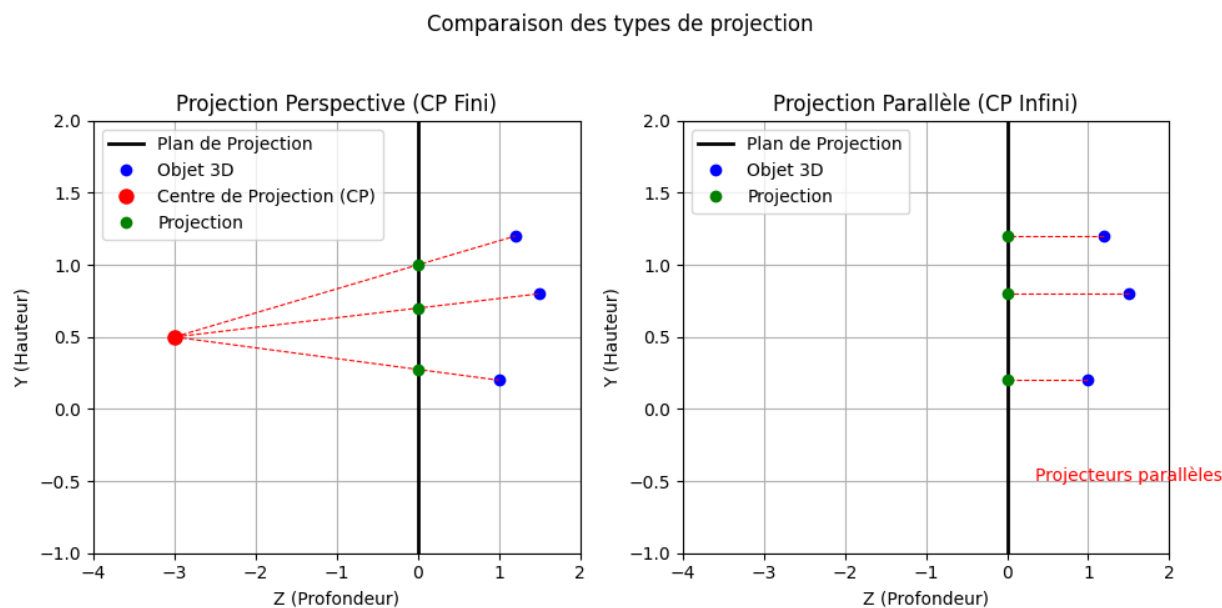


Figure 19: Projection perspective (projecteurs convergents vers CP) vs. Projection parallèle (projecteurs parallèles).

17 Projection Parallèle

Dans ce type de projection, les projecteurs sont parallèles entre eux. La direction de ces projecteurs est appelée **direction de projection (DP)**.

17.1 Définition et propriétés

- Le centre de projection est à l'infini.
- Les lignes parallèles dans l'espace 3D restent parallèles dans l'image 2D projetée (sauf si elles sont parallèles à la direction de projection).
- Les rapports des longueurs sur des lignes parallèles à un plan donné sont conservés.
- Moins réaliste que la perspective (pas d'effet de raccourci avec la distance).
- Utile pour les dessins techniques, les mesures, certaines vues de jeux (isométrique).

17.2 Types de projection parallèle

On distingue deux sous-catégories selon l'angle entre la direction de projection (DP) et la normale au plan de projection :

- **Projection Orthographique** : La direction de projection est **perpendiculaire** au plan de projection. Les projecteurs sont orthogonaux au plan.
- **Projection Oblique** : La direction de projection est **non perpendiculaire** au plan de projection. Les projecteurs coupent le plan de manière oblique.

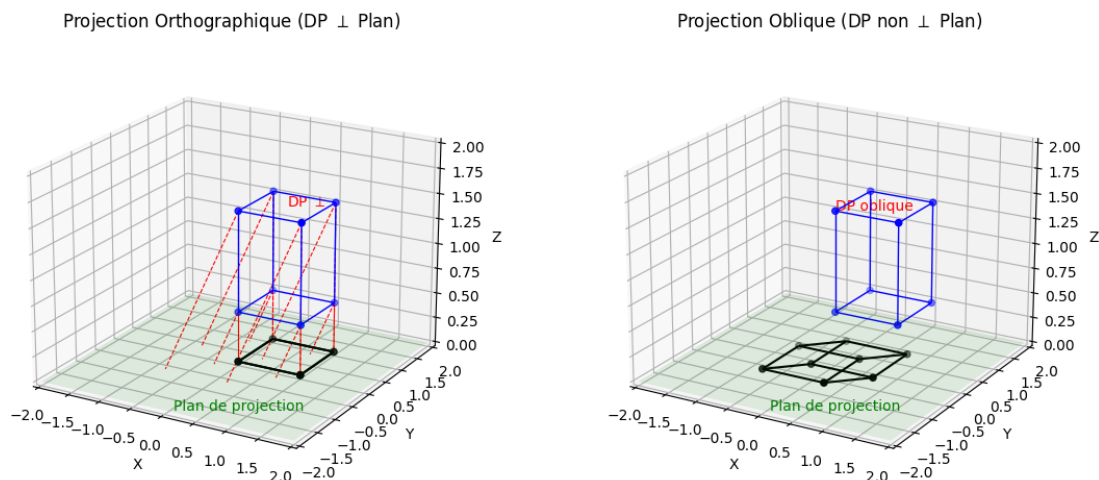


Figure 20: Projection parallèle orthographique (projecteurs rouges perpendiculaires au plan vert) vs. oblique (projecteurs rouges non perpendiculaires).

18 Projection Orthographique

C'est le type le plus simple de projection parallèle. La direction de projection est orthogonale au plan de projection.

18.1 Principe

La projection orthographique consiste essentiellement à "écraser" l'objet sur le plan de projection en supprimant une dimension (celle correspondant à la direction de projection). Si le plan de projection est l'un des plans principaux (xy, yz, xz), cela revient simplement à mettre à zéro la coordonnée correspondante.

18.2 Vues orthographiques

Les projections orthographiques sur les plans principaux sont très utilisées en dessin technique et CAO :

- **Vue de face (Front View)** : Projection sur le plan XY (DP = axe Z). On ignore la coordonnée Z.
- **Vue de dessus (Top View)** : Projection sur le plan XZ (DP = axe Y). On ignore la coordonnée Y.
- **Vue de côté (Side View)** : Projection sur le plan YZ (DP = axe X). On ignore la coordonnée X.

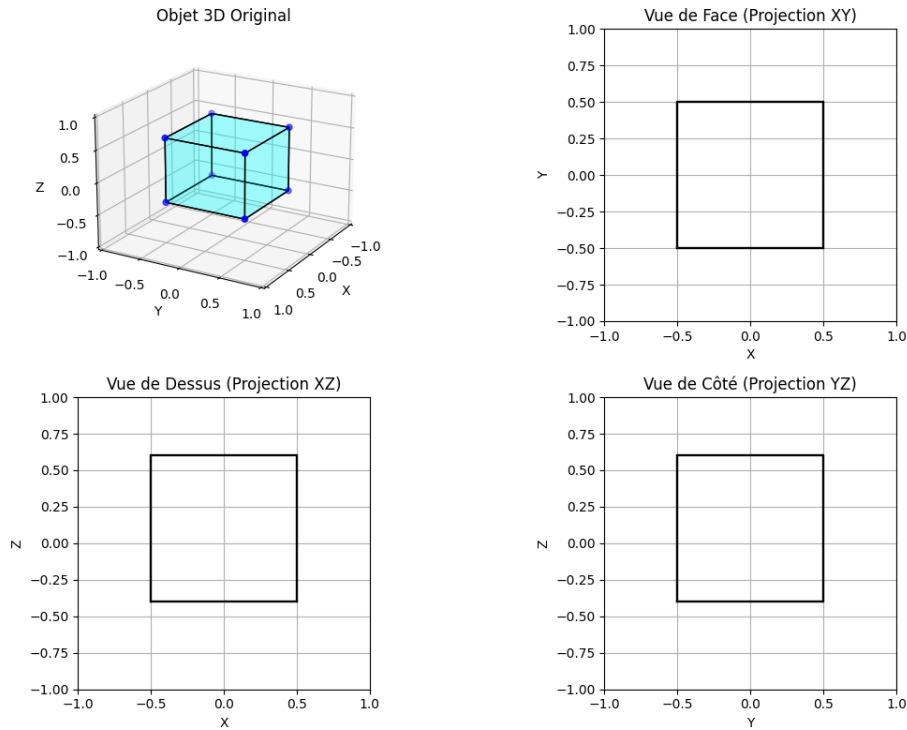


Figure 21: Objet 3D et ses vues orthographiques principales.

18.3 Forme matricielle

En coordonnées homogènes, la projection orthographique sur le plan $z = 0$ (vue de face) est représentée par la matrice M_{PZ} qui met la coordonnée Z à zéro :

$$M_{PZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Appliquer cette matrice à un point $\mathbf{P} = (x, y, z, 1)^T$ donne :

$$\mathbf{P}' = M_{PZ}\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}$$

Le point projeté a les coordonnées (x, y) dans le plan de projection. De même pour les autres vues :

- Projection sur XZ (Vue de dessus, ignorer Y) : $M_{PY} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Projection sur YZ (Vue de côté, ignorer X) : $M_{PX} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

18.4 Projection axonométrique

C'est une projection orthographique où le plan de projection n'est **pas** parallèle à l'un des plans principaux (XY, YZ, XZ). Le vecteur normal au plan de projection n'est parallèle à aucun des axes de coordonnées. Cela permet de montrer plusieurs facettes de l'objet simultanément. Les vues isométriques, dimétriques et trimétriques sont des types de projections axonométriques.

19 Projection Oblique

Dans ce type de projection parallèle, la direction de projection (DP) n'est pas perpendiculaire au plan de projection.

19.1 Définition

Les projecteurs sont parallèles entre eux mais coupent le plan de projection selon un angle $\phi \neq 90^\circ$. Cela a pour effet de décaler les points projetés par rapport à une projection orthographique. La matrice d'une projection oblique est souvent obtenue en combinant une transformation de cisaillement (shear) suivie d'une projection orthographique.

19.2 Types courants

Deux types de projections obliques sont fréquemment mentionnés :

- **Projection Cavalière** : L'angle ϕ entre les projecteurs et le plan est souvent de 45° . Les longueurs le long de l'axe perpendiculaire au plan de projection sont conservées (pas de raccourci). Cela donne une apparence un peu déformée.
- **Projection Cabinet** : L'angle ϕ est tel que $\tan(\phi) = 2$ (soit $\phi \approx 63.4^\circ$). Les longueurs le long de l'axe perpendiculaire au plan de projection sont réduites de moitié. Cela donne une apparence plus réaliste que la projection cavalière.

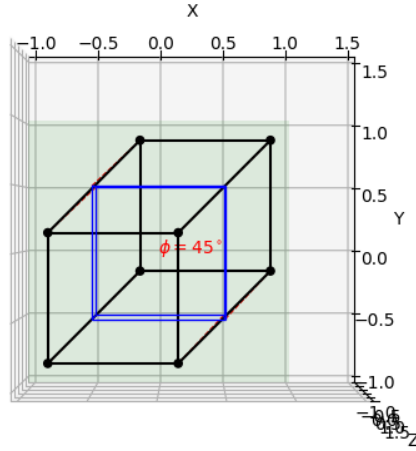
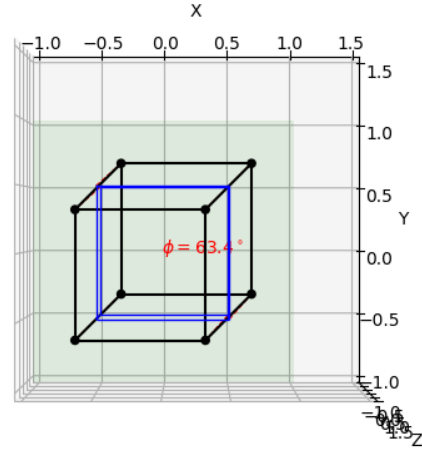
Projection Cavaliere ($\phi = 45^\circ$, $L=1$)Projection Cabinet ($\phi = 63.4^\circ$, $L=0.5$)

Figure 22: Exemples de projections obliques : Cavaliere et Cabinet (vues du dessus du plan de projection XY).

20 Projection Perspective

C'est le type de projection le plus réaliste, simulant la vision humaine ou l'objectif d'un appareil photo.

20.1 Définition et fonctionnement

- Le centre de projection (CP), souvent appelé "œil" ou "caméra", est situé à une distance finie du plan de projection.
- Les projecteurs sont les lignes droites passant par le CP et chaque point de l'objet 3D.
- L'image projetée est l'ensemble des points d'intersection de ces projecteurs avec le plan de projection.
- **Effet de perspective** : Les objets plus éloignés du CP apparaissent plus petits sur le plan de projection. Les objets de même taille à des distances différentes auront des tailles projetées différentes.
- **Non-conservation du parallélisme** : Les lignes parallèles dans l'espace 3D ne restent parallèles sur l'image 2D que si elles sont parallèles au plan de projection. Les autres ensembles de lignes parallèles semblent converger vers un point appelé **point de fuite**.

20.2 Calcul (Vue de côté simple)

Considérons une configuration simple : le CP est à l'origine (0,0,0), le plan de projection est à $z = f$ (où f est la distance focale ou distance au plan image), et on projette un point $P = (x, y, z)$ sur ce plan en $q = (x_p, y_p, f)$. Par similarité des triangles (vu de côté, en considérant l'axe Y et Z par exemple) :

$$\frac{y_p}{f} = \frac{y}{z} \implies y_p = \frac{y \cdot f}{z} = \frac{y}{z/f}$$

De même, vu de dessus (en considérant X et Z) :

$$\frac{x_p}{f} = \frac{x}{z} \implies x_p = \frac{x \cdot f}{z} = \frac{x}{z/f}$$

Si le plan de projection est à $z = 1$ (caméra normalisée), alors $x_p = x/z$ et $y_p = y/z$. On voit que les coordonnées projetées dépendent inversement de la profondeur z .

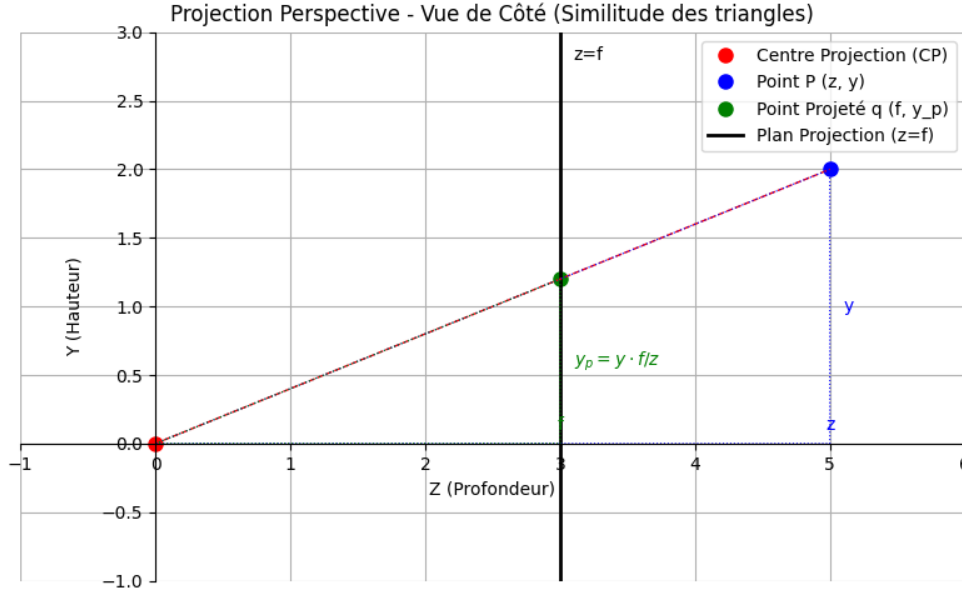


Figure 23: Calcul de la projection perspective par similarité des triangles (vue de côté, plan YZ).

20.3 Point de fuite

Un ensemble de lignes parallèles dans l'espace 3D qui ne sont pas parallèles au plan de projection convergera vers un unique point sur l'image projetée, appelé point de fuite.

- **Perspective à 1 point de fuite** : Seulement un des axes principaux (généralement la profondeur Z) converge. Les lignes parallèles aux axes X et Y restent parallèles sur l'image.
- **Perspective à 2 points de fuite** : Deux axes (e.g., X et Z) convergent vers deux points de fuite distincts. Les lignes parallèles à l'axe Y restent verticales. Courant pour les vues architecturales.
- **Perspective à 3 points de fuite** : Les trois axes X, Y, Z convergent vers trois points de fuite distincts. Donne une vue plus dramatique, souvent utilisée pour regarder des objets hauts depuis le bas ou inversement.

20.4 Représentation matricielle

La projection perspective peut être représentée par une matrice 4x4 en coordonnées homogènes. La matrice de projection perspective la plus simple (CP à l'origine, projection sur $z = f$) est :

$$M_{persp} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix}$$

Appliquons cette matrice à $\mathbf{P} = (x, y, z, 1)^T$:

$$\mathbf{P}' = M_{persp} \mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/f \end{pmatrix}$$

Les coordonnées homogènes résultantes sont $(x, y, z, z/f)$. Pour retrouver les coordonnées cartésiennes, on divise par la 4ème composante $W = z/f$:

$$\mathbf{p}' = \left(\frac{x}{z/f}, \frac{y}{z/f}, \frac{z}{z/f} \right) = \left(\frac{x \cdot f}{z}, \frac{y \cdot f}{z}, f \right)$$

On retrouve bien les coordonnées projetées x_p, y_p sur le plan $z = f$. La profondeur z est "perdue" au profit de f , mais l'information de profondeur originale est stockée dans la coordonnée W avant la division (division perspective). Une matrice de projection plus générale prenant en compte un frustum de vue (pyramide tronquée définissant le volume visible) est présentée plus loin (Section ??).

21 Plans de Clipping

Avant la projection finale et la rastérisation, il est nécessaire de délimiter le volume d'espace qui sera effectivement visible et affiché. Les primitives géométriques (triangles, lignes) qui se trouvent partiellement ou totalement en dehors de ce volume doivent être découpées (clipped).

21.1 Utilité et fonctionnement

- **View Frustum** : Le volume visible en projection perspective est généralement une pyramide tronquée appelée "frustum de vue". Il est défini par 6 plans :
 - **Near Plane (Plan proche, z-near)** : Empêche les objets trop proches de la caméra d'être affichés (évite division par zéro ou erreurs de Z-buffer).
 - **Far Plane (Plan lointain, z-far)** : Empêche les objets trop éloignés d'être affichés (limite la profondeur de champ, économise calculs).
 - **Left, Right, Top, Bottom Planes** : Définissent les limites latérales du champ de vision.
- **Clipping** : Algorithmes (e.g., Sutherland-Hodgman pour les polygones, Cohen-Sutherland pour les lignes) qui éliminent les parties des primitives situées en dehors du frustum de vue.
- **Problèmes sans Clipping** :
 - Rastérisation incorrecte de primitives traversant le plan z-near (triangles derrière l'œil).
 - Problèmes de précision du Z-buffer (tampon de profondeur) si la plage [z-near, z-far] est trop grande, surtout près de $z=0$ où la précision des flottants est limitée. Une valeur très petite (proche de 0) pour z-near concentre une grande partie de la précision du Z-buffer très près de l'œil, laissant peu de précision pour le reste de la scène.

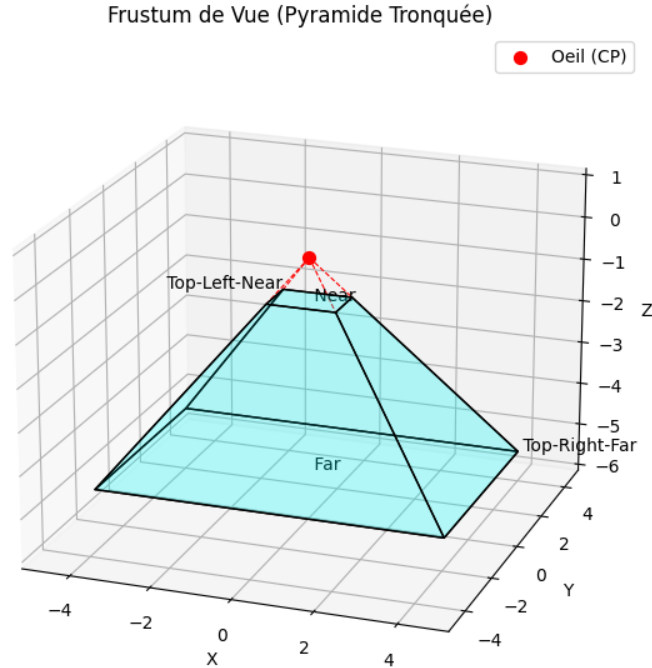


Figure 24: Visualisation du frustum de vue défini par les plans Near, Far, Top, Bottom, Left, Right.

22 Passage en Coordonnées Normalisées (NDC)

Après le clipping dans l'espace caméra (ou parfois après la projection), le frustum de vue est transformé en un volume canonique simple, généralement un cube unitaire centré à l'origine, appelé **Normalized Device Coordinates (NDC)**. Ce cube a typiquement des coordonnées allant de -1 à +1 pour X, Y et Z.

22.1 Objectif

- Simplifier les étapes ultérieures comme le clipping (clipping contre un cube est plus simple) et le mapping vers l'écran (Viewport Transform).
- Standardiser le volume visible indépendamment des paramètres de la caméra (FOV, aspect ratio, near/far).

22.2 Transformation

La transformation qui mappe le frustum de vue perspectif vers le cube NDC $[-1, 1]^3$ est réalisée par la **matrice de projection perspective** (souvent appelée M_{proj}). Cette matrice combine la projection elle-même et la déformation (warp) de l'espace pour transformer le frustum en cube. Les slides montrent un exemple de dérivation de la matrice A qui transforme 8 points x_i (coins du frustum) en 8 points y_i (coins du cube NDC). Cette matrice A est la matrice de projection M_{proj} . Le but est de trouver A telle que $y_i = Ax_i$ pour tous les coins i . (Attention : les x_i, y_i dans les slides sont des vecteurs homogènes 4D). La dérivation aboutit à la matrice A (qui est M_{proj}) suivante (pour une caméra regardant vers -Z, et

un mapping NDC de $[-1, 1]$:

$$A = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

où l, r, b, t, n, f sont les coordonnées des plans Left, Right, Bottom, Top, Near, Far du frustum dans l'espace caméra (n, f sont des distances, donc positives, mais souvent utilisées comme coordonnées $z = -n, z = -f$).

23 Transformation de la Perspective

La matrice $M = A$ dérivée ci-dessus est la **matrice de transformation de perspective** complète. Elle prend un point \mathbf{P}_{view} en coordonnées caméra (dans le frustum) et le transforme en coordonnées normalisées \mathbf{P}_{NDC} :

$$\mathbf{P}_{NDC} = M \cdot \mathbf{P}_{view}$$

Après multiplication par M , les coordonnées (x', y', z', w') de \mathbf{P}_{NDC} sont encore homogènes. La division par w' (division perspective) donne les coordonnées finales dans le cube $[-1, 1]^3$. $(x_{ndc}, y_{ndc}, z_{ndc}) = (x'/w', y'/w', z'/w')$ La coordonnée z_{ndc} est particulièrement importante car elle est souvent utilisée comme valeur de profondeur dans le Z-buffer. La transformation non-linéaire de z_{view} en z_{ndc} (due à la division par w' , qui dépend de z_{view}) fait que la précision de la profondeur n'est pas uniforme dans l'espace NDC.

24 Transformations Géométriques (Synthèse)

Pour afficher un objet défini dans son propre repère (Object Space) sur l'écran, une série de transformations est appliquée :

- **Modèle vers Scène (Model to World)** : M_{model} Placer l'objet dans la scène globale. Utilise les transformations de modélisation (translation, rotation, échelle) stockées dans le graphe de scène. $\mathbf{P}_{world} = M_{model} \cdot \mathbf{P}_{object}$
- **Scène vers Caméra (World to View)** : M_{view} Transformer les coordonnées de la scène dans le repère lié à la caméra. C'est l'inverse de la transformation qui placerait la caméra dans la scène. Si M_{cam} positionne la caméra dans le monde, alors $M_{view} = (M_{cam})^{-1}$. $\mathbf{P}_{view} = M_{view} \cdot \mathbf{P}_{world}$
- **Caméra vers NDC (Projection)** : M_{proj} Appliquer la projection (perspective ou orthographique) et normaliser les coordonnées dans le cube $[-1, 1]^3$. $\mathbf{P}_{clip} = M_{proj} \cdot \mathbf{P}_{view}$ (Les coordonnées sont souvent appelées "clip coordinates" avant la division perspective).
- **Division Perspective** : Diviser par la composante W pour obtenir les coordonnées NDC. $(x_{ndc}, y_{ndc}, z_{ndc}) = (x_{clip}/w_{clip}, y_{clip}/w_{clip}, z_{clip}/w_{clip})$
- **Transformation Fenêtre (Viewport Transform)** : $M_{viewport}$ Mapper les coordonnées NDC $[-1, 1]^2$ (on ignore souvent Z ici pour le mapping 2D) vers les coordonnées de la fenêtre d'affichage (pixels) sur l'écran. $(x_{screen}, y_{screen}) = M_{viewport} \cdot (x_{ndc}, y_{ndc})$

La transformation totale de l'objet à l'écran peut être vue comme la composition de ces matrices : $M_{total} = M_{viewport} \cdot M_{proj} \cdot M_{view} \cdot M_{model}$ (en ignorant la division perspective pour le moment).

24.1 Exemple de transformations combinées (Caméra)

Les slides (page 75) montrent un exemple où l'on veut placer l'objet dans le repère caméra. Considérons une caméra située en $(4, 2, 0)$ dans le repère monde, regardant vers l'origine le long de l'axe -X (donc sa direction de vue est $(-1, 0, 0)$). On veut transformer un objet du repère monde vers le repère caméra, où la caméra est à l'origine et regarde le long de l'axe -Z. Il faut deux transformations : 1. **Translation** $T(-4, -2, 0)$ pour amener la caméra (et le monde avec elle) à l'origine. 2. **Rotation** pour aligner l'axe de vue (-X monde)

avec l'axe de vue caméra (-Z caméra). Il faut tourner de $+90^\circ$ autour de l'axe Y (pour amener -X vers -Z). La matrice de vue M_{view} est la composition $M_{view} = R_y(90^\circ) \cdot T(-4, -2, 0)$.

$$R_y(90^\circ) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

$$M_{view} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Appliquer M_{view} à un point en coordonnées monde \mathbf{P}_{world} donne ses coordonnées \mathbf{P}_{view} dans le repère caméra.