Visualisation 3D

Principes mathematiques

Christian Nguyen

Departement d'informatique

Universite de Toulon et du Var

Transformations géométriques

Transformations simples

Translation

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotations

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Theta & \sin\Theta & 0 \\ 0 & -\sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

autour de l'axe X

$$R_y = \begin{pmatrix} \cos\Theta & 0 & -\sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

autour de l'axe Y

Déformations

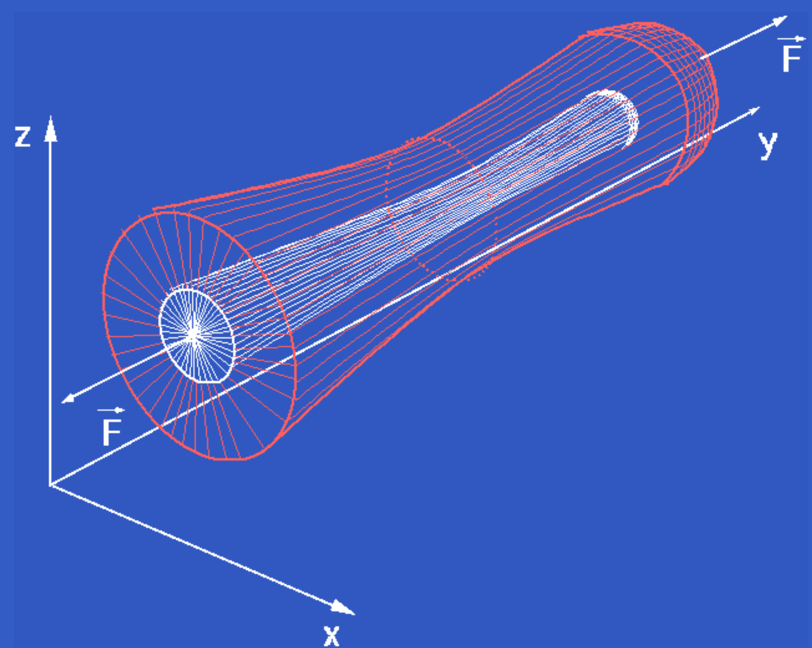
Formule générale :

$$\begin{cases} x' = F_a(x) \\ y' = F_b(y) \\ z' = F_c(z) \end{cases}$$

Une dilatation est une forme particulière de déformation.

Effilement (*tapering*) suivant Y

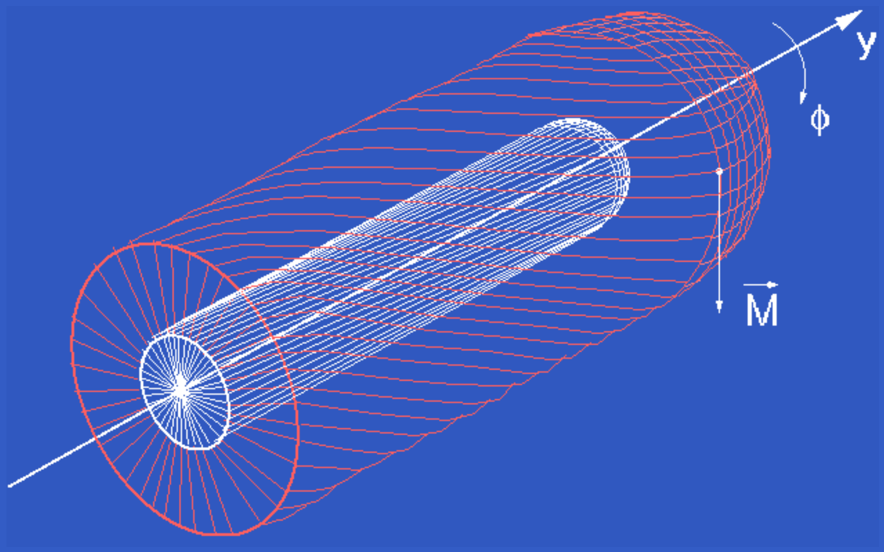
$$\begin{cases} x' = f(y)x \\ y' = y \\ z' = f(y)z \end{cases}$$



Torsion (*twisting*) autour de Z

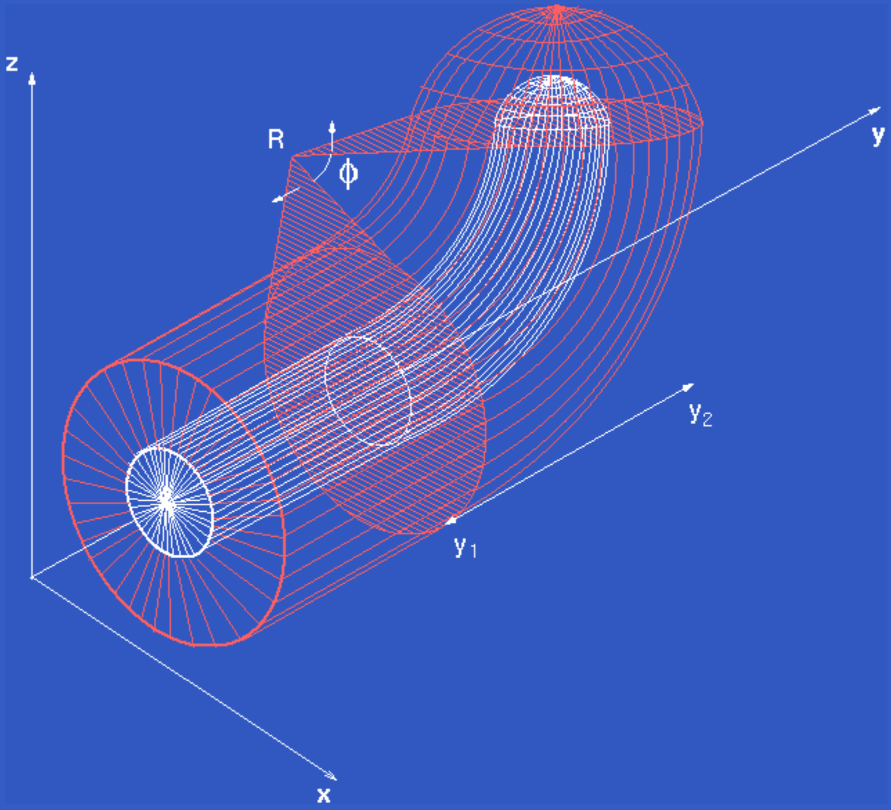
$f(z)$: grandeur de torsion, fonction de la distance.

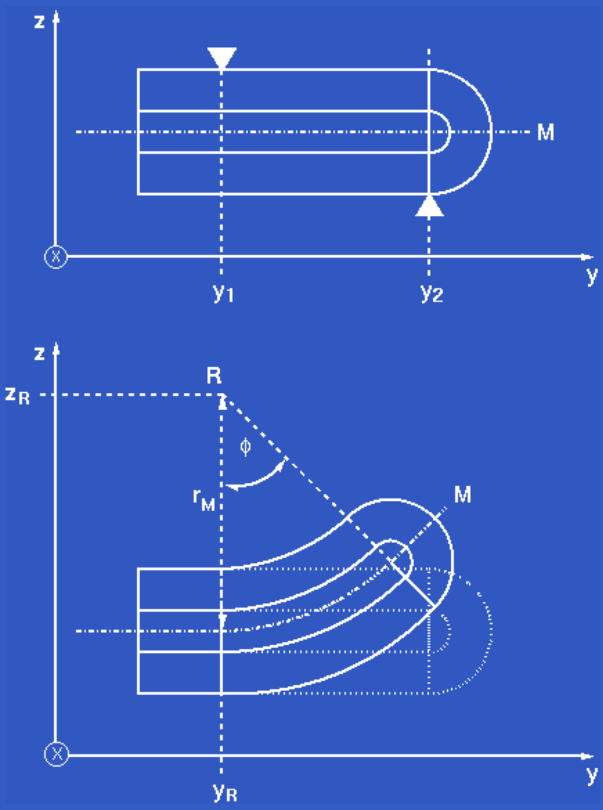
$$\begin{cases} x' = x\cos(f(y)) + z\sin(f(y)) \\ y' = y \\ z' = -x\sin(f(y)) + z\cos(f(y)) \end{cases}$$



Courbure, pliure (*bending*)

Transformation composite (rotation, translation). Exemple :





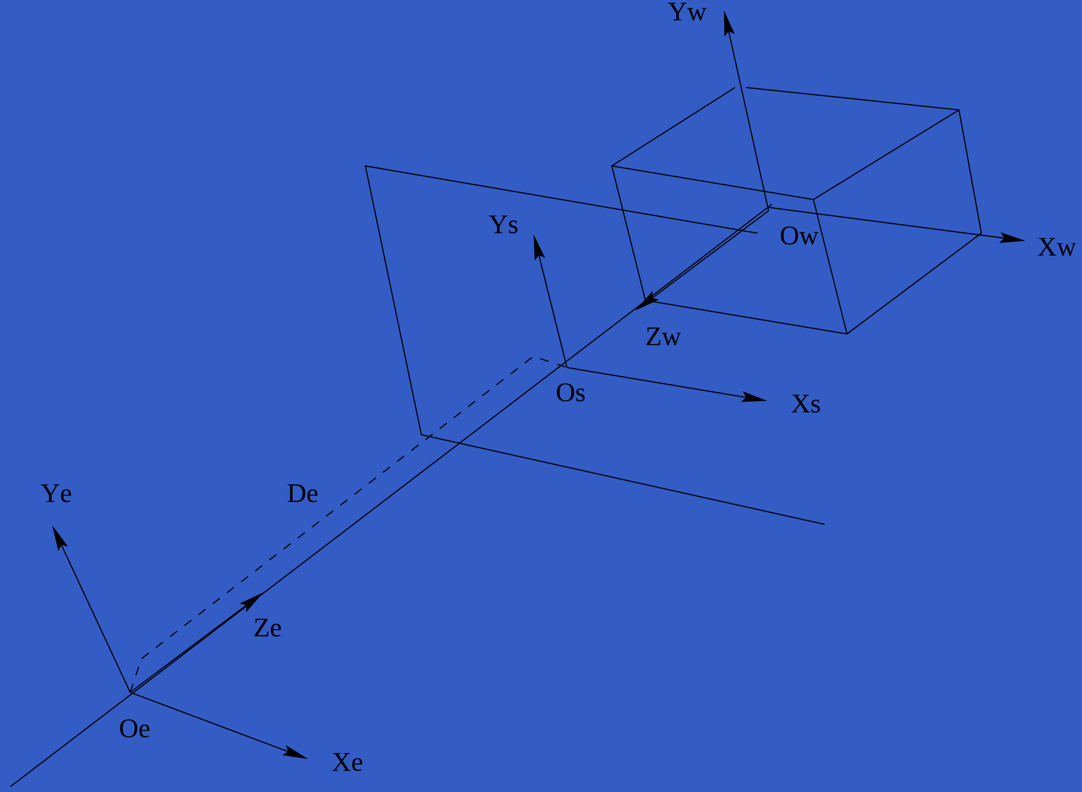
Courbure, pliure (*bending*)

$$
\begin{cases}
x' = x \\
y' = \begin{cases}
-\sin\Phi(z-r) + y_r & y_1 \le y \le y_2 \\
-\sin\Phi(z-r) + y_r + \cos\Phi(y-y_1) & y < y_1 \\
-\sin\Phi(z-r) + y_r + \cos\Phi(y-y_2) & y > y_2
\end{cases} \\
z' = \begin{cases}
\cos\Phi(z-r) + r & y_1 \le y \le y_2 \\
\cos\Phi(z-r) + r + \sin\Phi(y-y_1) & y < y_1 \\
\cos\Phi(z-r) + r + \sin\Phi(y-y_2) & y > y_2
\end{cases}
\end{cases}
$$

Projection

Projection simple

On utilise trois systèmes de coordonnées O_w (world), O_e (eyes) et O_s (screen). On suppose ici que Z_e est colinéaire à Z_w.



Projection simple

Trois étapes :

1. changement de repère : du système de coordonnées global (O_w) au système de coordonnées lié à la caméra (O_e) :

$$\begin{cases} x_e = x_w \\ y_e = y_w \\ z_e = -z_w + O_eO_w \end{cases}$$

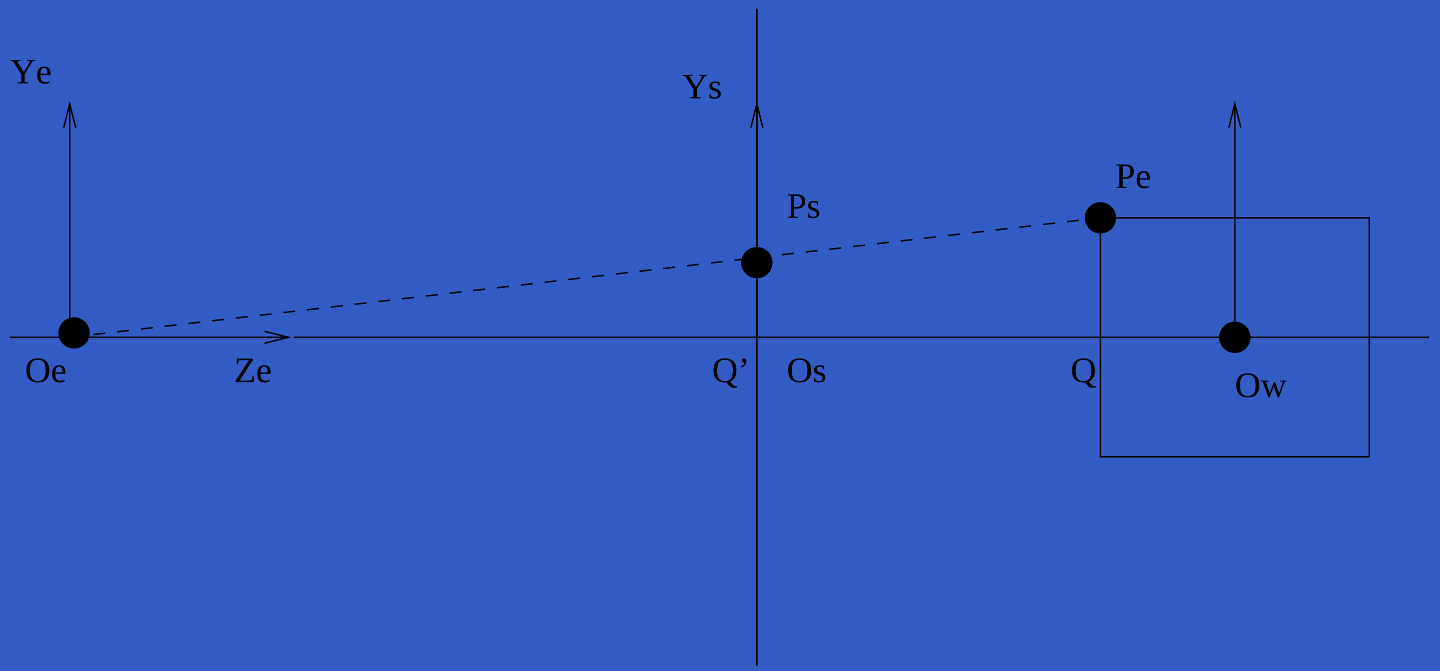
Projection simple

2. projection :
   – parallèle, on ignore la coordonnée en z de chaque point :

$$\begin{cases} x_s = x_e \\ y_s = y_e \end{cases}$$

Projection simple

– perspective, on tient compte de la coordonnée z_e. La vue en coupe, selon le plan (Y_e, Z_e) montre que les triangles $O_eQ'P_s$ et O_eQP_e sont similaires :



Projection simple

Soit, dans le plan (Y_e, Z_e) :

$$\frac{y_s}{D_e} = \frac{y_e}{z_e} \text{ soit } y_s = D_e \cdot \frac{y_e}{z_e}$$

De même pour le plan (X_e, Z_e) :

$$\frac{x_s}{D_e} = \frac{x_e}{z_e} \text{ soit } x_s = D_e \cdot \frac{x_e}{z_e}$$

$$z_s = z_e$$

Projection simple

3. formatage des coordonnées en fonction de la résolution de l'écran. En considérant que l'origine $O_s(S_x, S_y)$ de l'écran est un point situé au centre de cet écran, les formules de projection et de mise à l'echelle deviennent :

$$x_s = D_e \cdot \frac{x_e}{z_e} + S_x$$

$$y_s = D_e \cdot \frac{y_e}{z_e} + S_y$$

D'où le pipe-line 3D "première version" :

$$P(x_w, y_w, z_w) \overset{\text{changement de repère}}{\longrightarrow} P_e(x_e, y_e, z_e) \overset{\text{projection}}{\longrightarrow} P_s(x_s, y_s)$$

Projection générale

Dans le cas général, la caméra est située à un endroit $F(x_f, y_f, z_f)$ (*from*) dans le repère de la scène O_w, et pointe vers un endroit $A(x_a, y_a, z_a)$ (*at*) toujours exprimé dans ce même repère.

Paramètres de visualisation (projection et découpage) :

- 2 points : PRP (*Projection Reference Point*) et VRP (*View Reference Point*),
- 2 vecteurs : VPN (*View Point Normal*) et VUP (*View Up Vector*),
- 6 scalaires : (u_{min}, v_{min}) et (u_{max}, v_{max}) définissant une partition du plan de projection, d_f et d_b les distances des plans de coupe avant et arrière.

Projection générale

On définit u et v les deux autres vecteurs du modèle de caméra à partir de la projection de VUP sur le plan de projection défini par VPN :

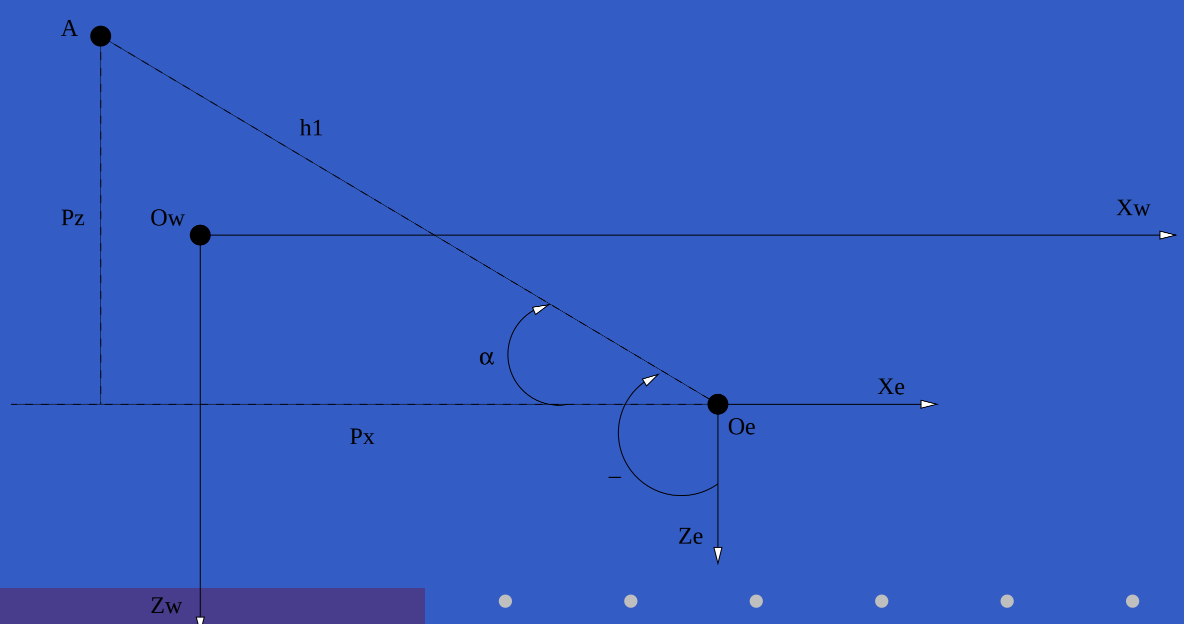
$$u' = VUP - (VPN \bullet VUP).VPN$$
$$u = \frac{u'}{|u'|}$$
$$v = u \wedge VPN$$

Projection générale

Le calcul de la matrice de changement de repère fait alors intervenir trois rotations en plus de la classique translation [Newman79] :

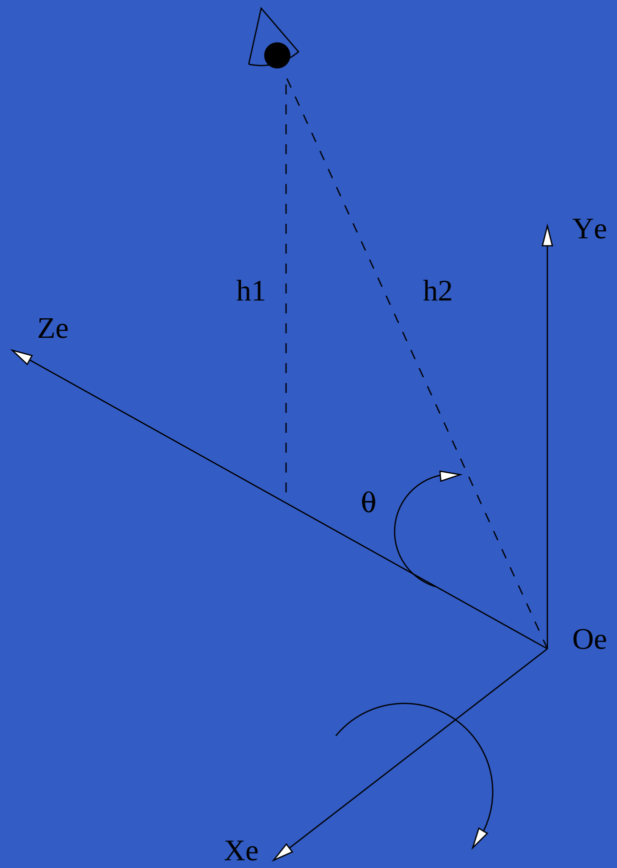
– translation du repère O_e au point F,

– rotation autour de l'axe Y_e afin d'aligner l'axe Z_e du repère avec la projection parallèle dans le plan (X_w, Z_w) du vecteur défini par les deux points F et A,



Projection générale

– rotation du repère O_e autour de l'axe X_e,



Projection générale

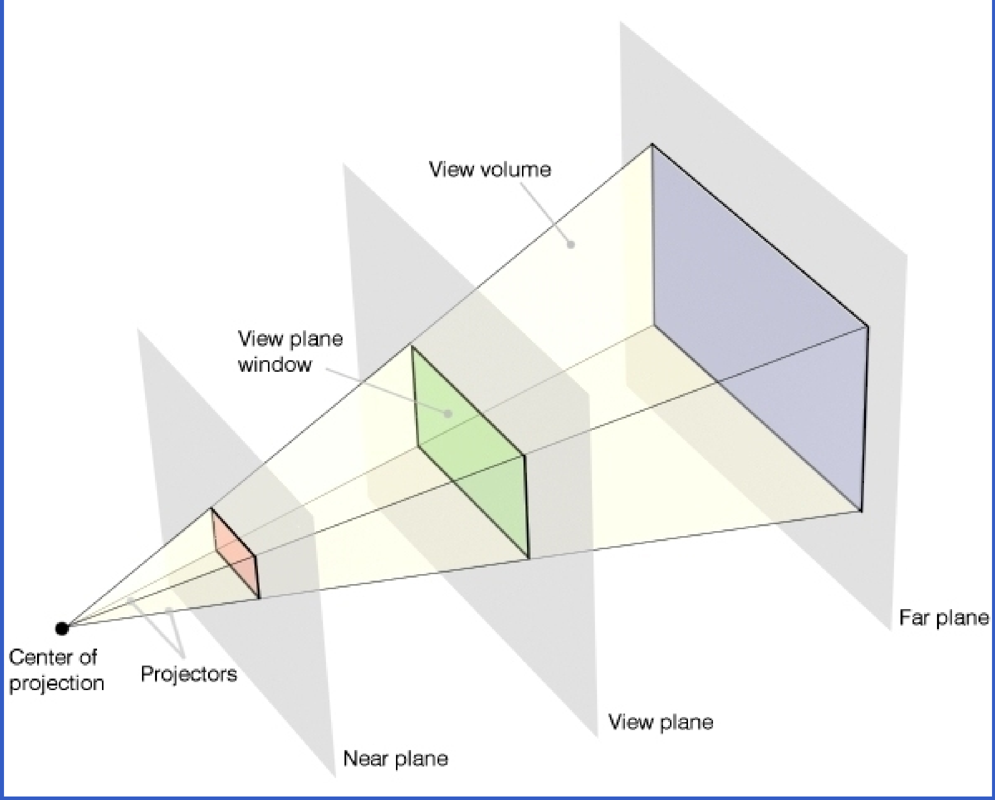
- inclinaison de la caméra le long de son axe Z_e (modification de la "verticale" Y_e),
- changement de signe, qui permet de transformer le repère O_e, intialement direct, en un repère indirect.

Composées, ces cinq transformations donnent la matrice de changement de repère finale. Les autres phases du pipeline graphique (la projection) restent les mêmes.

Fenêtrage

Fenêtrage

Cette opération passe d'abord par la définition d'un volume de visualisation.



Fenêtrage

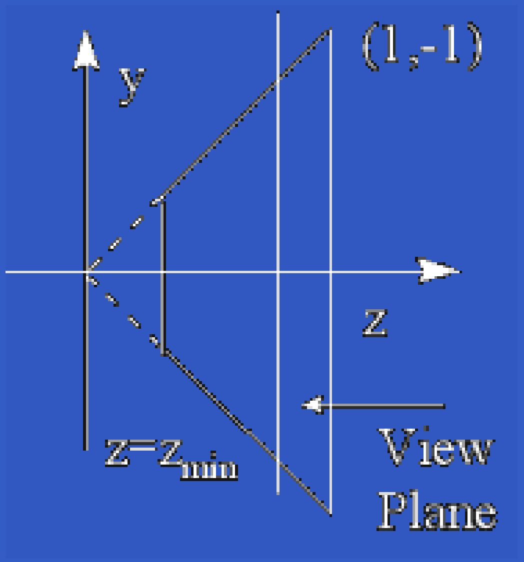
Calcul de l'intersection des segments de droites avec chacun des 6 plans définissant le volume de vue (transpositions directes des algorithmes en 2D de Cohen-Sutherland, Sutherland-Hodgman, . . .).

– projection parallèle :

$$\left\{ \begin{array}{l} x_1 = -1,\ x_2 = 1 \\ y_1 = -1,\ y_2 = 1 \\ z_1 = z_{min},\ z_2 = z_{max} \end{array} \right.$$

Fenêtrage

– projection perspective :



$$\begin{cases} x_1 = -z, \; x_2 = z \\ y_1 = -z, \; y_2 = z \\ z_1 = z_{min}, \; z_2 = z_{max} \end{cases}$$