Synthèse d'images

(5) Composition de scène

Plan de l'exposé :

1- Transformations géométriques

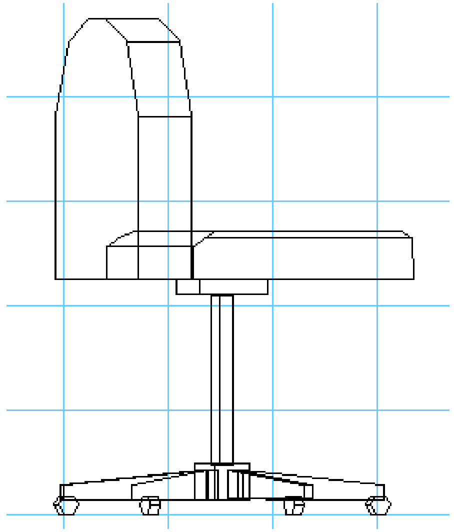
2 - Déformations

3- Opérateurs booléens

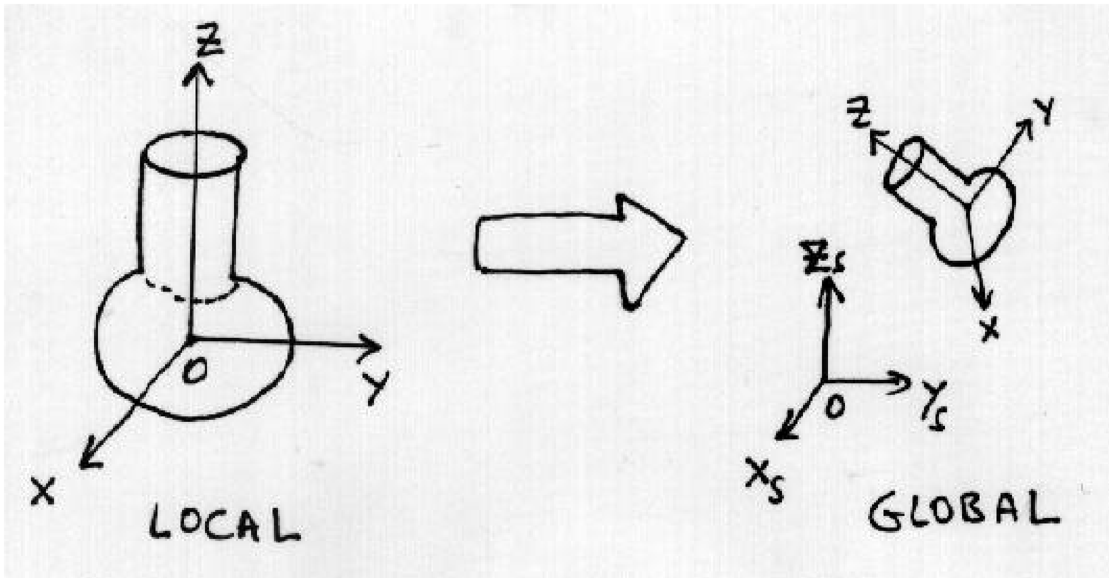
4- Hiérarchie d'objets

v5 P. CUBAUD <cubaud@cnam.fr> Dept. informatique CNAM, Paris

1. Transformations géométriques



- Pour construire de nouveaux objets à partir de volumes élémentaires

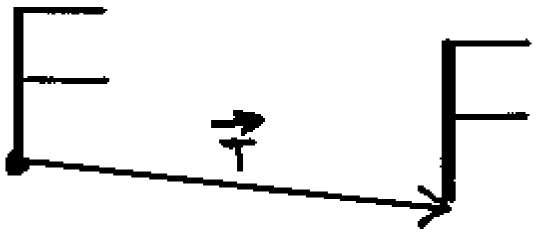


- Pour changer le système des coordonnées utilisées pour décrire un objet.

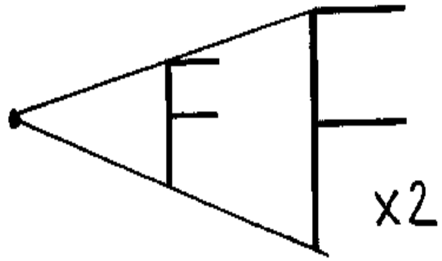
Différents types

D'après [QUID] p. 278-9

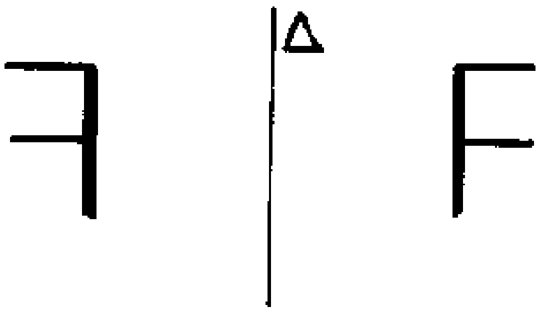
Translation



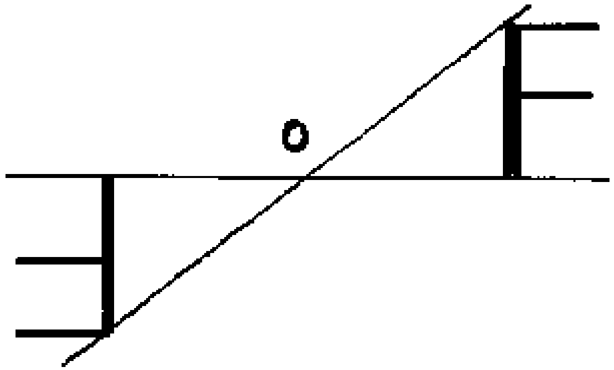
Homothétie



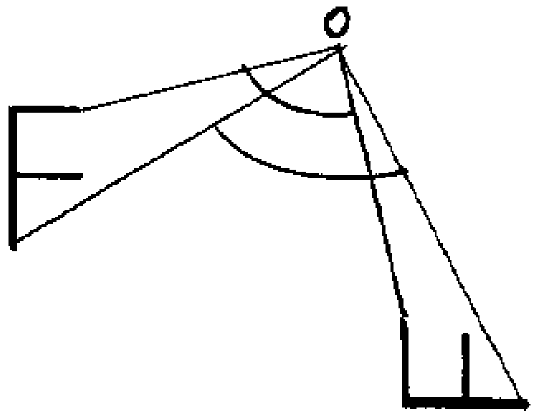
Symétrie orthogonale



Symétrie centrale



Rotation



Opérations matricielles (2D)

- Matrices homogènes

  - Toutes les opérations précédentes se ramènent à la multiplication d'une matrice par le vecteur des coordonnées de départ, SAUF pour la translation (addition de 2 vecteurs)

  - On va représenter un point P(x,y) par une famille de vecteurs de 3 éléments (x,y,W). W est le facteur d'échelle.

  Ainsi (x,y,W) et (x / W, y / W, 1) représentent le même point.

  - Si W=0, on dira que le point est à l'infini…

- Translation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Homothétie

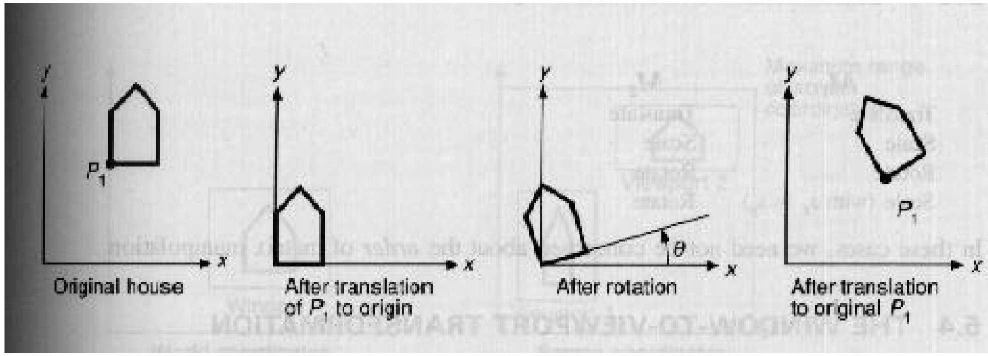
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Rotation par rapport à l'origine

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

Exemple de composition

On veut faire une rotation de la maison autour du point P1



[FOLEY] pp. 208-9

Suite d'opérations à réaliser :

Translation de l'objet à l'origine & Rotation & Translation de l'objet à sa position de départ

$$T(x_1, y_1) \circ R(\theta) \circ T(-x_1, -y_1)$$

Matrices résultantes :

$$= \begin{pmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & x_1(1-\cos\theta) + y_1 \sin\theta \\ \sin\theta & \cos\theta & y_1(1-\cos\theta) - x_1 \sin\theta \\ 0 & 0 & 1 \end{pmatrix}$$

Attention à l'ordre !!

Commutativité pas toujours vérifiée : T.R≠R.T

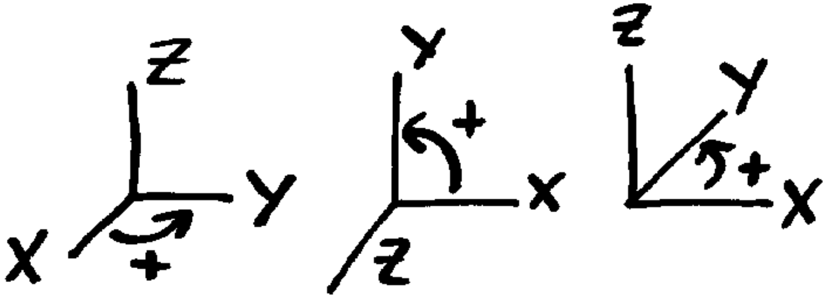
Opérations matricielles (3D)

- Translation : $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

- Homothétie : $\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

- Rotations par rapports aux axes X, Y et Z

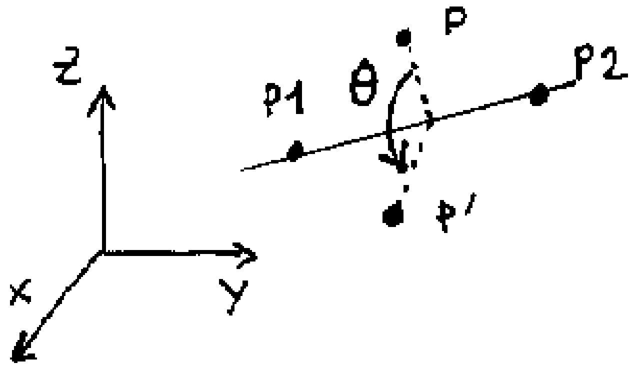
Si répère "main droite" !!! θ positif en allant de X à Y



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

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

- Rotation autour d'un axe quelconque :



1- Translation T(-x1,-y1,-z1) pour amener P1 à l'origine

$$x' = x_2 - x_1 \quad y' = y_2 - y_1 \quad z' = z_2 - z_1$$

2- Rotation Ry(-α) autour de l'axe des Y pour amener P'2 dans le plan (O,Y,Z)

$$d = \sqrt{x'^2 + z'^2} \quad \cos(-\alpha) = \frac{z'}{d} \quad \sin(-\alpha) = \frac{-x'}{d}$$

3- Rotation Rx(β) autour de l'axe des X pour amener P''2 sur l'axe des Z

$$D = \sqrt{d^2 + y'^2} \quad \cos(\beta) = \frac{d}{D} \quad \sin(\beta) = \frac{y'}{D}$$

4- Rotation Rz(θ) de θ sur l'axe des Z

5- Opération inverse de 3 : Rx(-β)

6- Opération inverse de 2 : Ry(α)

7- Opération inverse de 1 : T(x1,y1,z1)

- Un exemple : rotation de 60° selon l'axe (0,0,0)-(1,1,1)

On a :

$$x' = x_2 - x_1 = 1 \quad y' = y_2 - y_1 = 1 \quad z' = z_2 - z_1 = 1$$

$$d = \sqrt{2} \quad \cos(-\alpha) = \frac{1}{\sqrt{2}} \quad \sin(-\alpha) = \frac{-1}{\sqrt{2}} \quad \alpha = \frac{\pi}{4}$$

donc $R_Y(\frac{\pi}{4}) = \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

$$D = \sqrt{3} \quad \cos(\beta) = \sqrt{\frac{2}{3}} \quad \sin(\beta) = \sqrt{\frac{1}{3}} \quad \beta = \frac{\pi}{5}$$

donc $R_X(\frac{\pi}{5}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2/3} & 1/\sqrt{3} & 0 \\ 0 & -1/\sqrt{3} & \sqrt{2/3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

on a aussi $R_Z(\frac{\pi}{3}) = \begin{pmatrix} 1/2 & \sqrt{3}/2 & 0 & 0 \\ -\sqrt{3}/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

La matrice de transformation est :

$$M = R_Y\left(\frac{\pi}{4}\right) \circ R_X(-\frac{\pi}{5}) \circ R_Z(\frac{\pi}{3}) \circ R_X(\frac{\pi}{5}) \circ R_Y\left(-\frac{\pi}{4}\right)$$

Soit $M = \frac{1}{3}\begin{pmatrix} 2 & 2 & -1 & 0 \\ -1 & 2 & 2 & 0 \\ 2 & -1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

- Application à un cube :

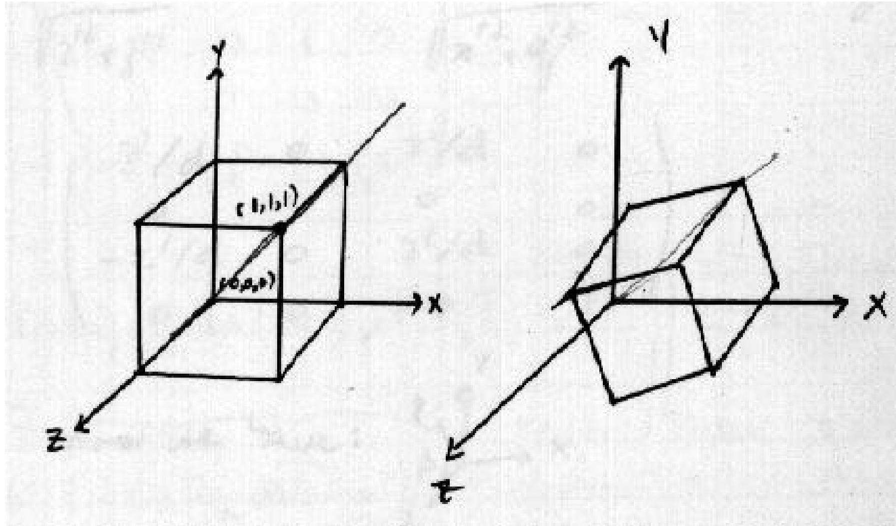


Table des sommets avant rotation :

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Table des sommets après rotation :

0	0	0
2/3	-1/3	2/3
-1/3	2/3	2/3
1/3	1/3	4/3
2/3	2/3	-1/3
4/3	1/3	1/3
1/3	4/3	1/3
1	1	1

2. Déformations d'objets

- Des techniques surtout utilisées en animation (donc un peu hors sujet…)

- Appliquée directement sur la représentation polyédrale

  => risque : un cube tordu n'est plus un polyèdre à 6 faces

  => algorithme de subdivision adaptative

- Appliquée sur des surfaces paramétrées (Bézier, Bspline)

  => facile, action sur les points de contrôles

  => possibilité de déformation locale si B-spline

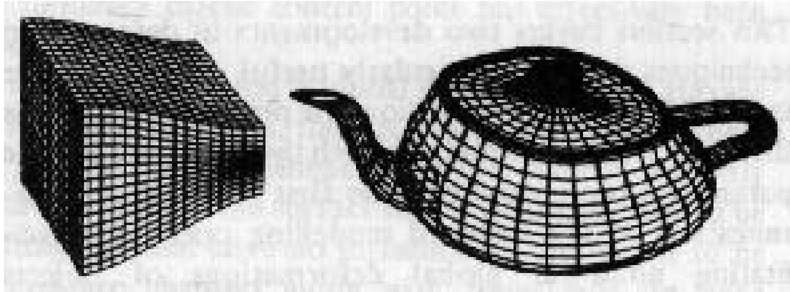
- Bibliographie : [WATT] chapitre 17, sur l'animation des objets mous.

Déformations non-linéaires (Barr, 1984)

La transformation est variable selon chaque point du polyèdre.

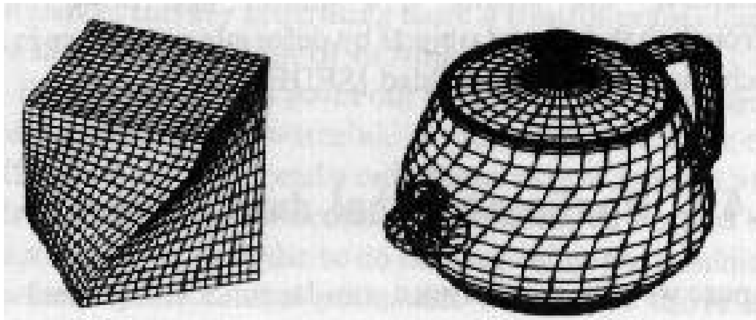
- "tapering" (effilage) :

$$\begin{pmatrix} x' & y' & z' \end{pmatrix} = \begin{pmatrix} xf(z) & yf(z) & z \end{pmatrix}$$

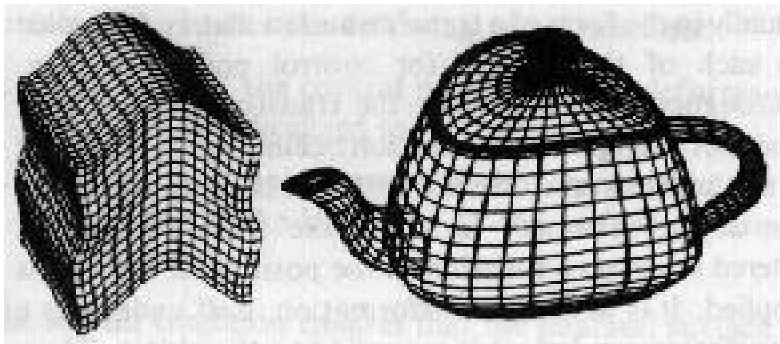


- "twisting" (torsion) :

$$\begin{pmatrix} x' & y' & z' \end{pmatrix} = R_Z(f(z))$$



- "bending" (pliage)

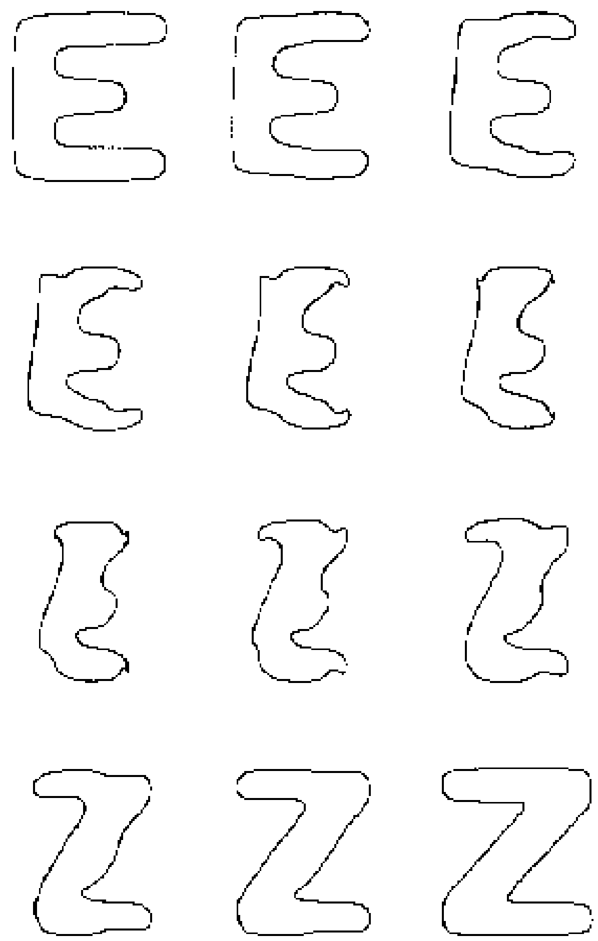


Le "morphing"

- On se donne :

  - Une image (une courbe, un polygone…) de départ

  - Une image d'arrivée



[WATT] p. 413

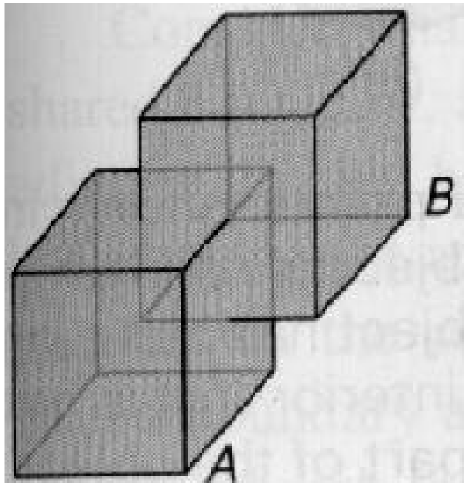
- On veut calculer N images intermédiaires

Par exemple : par interpolation linéaire

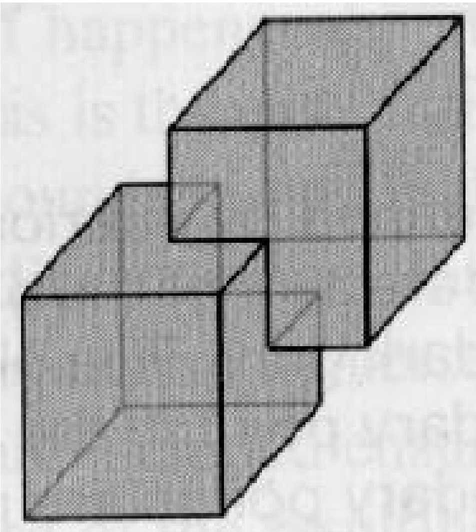
$x_i = x_D + i \frac{x_A - x_D}{N+1}$, idem avec yi et zi

3. Opérateurs booléens

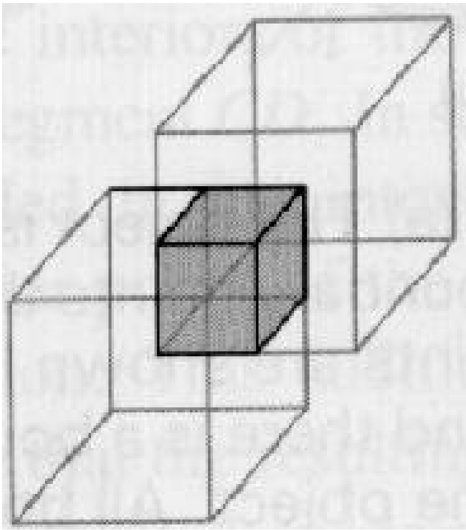
[FOLEY] p.535



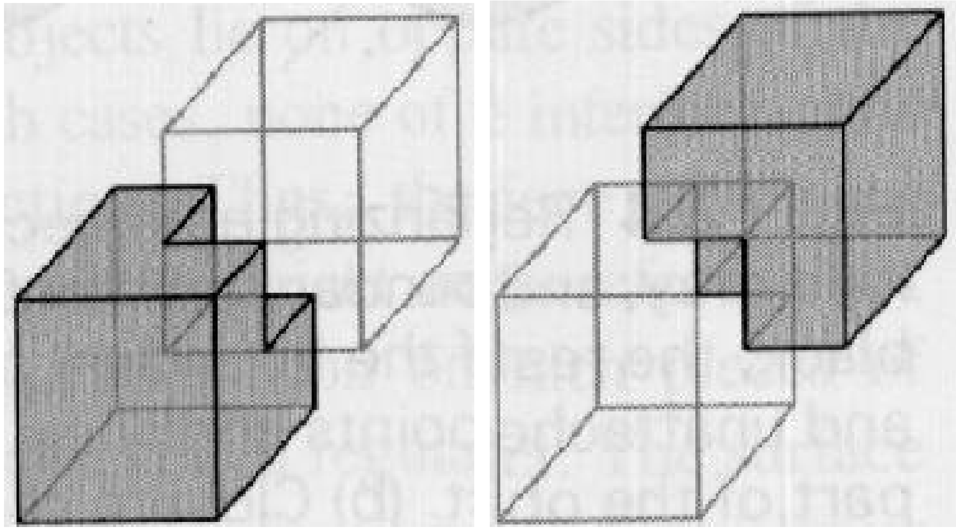
Union



Intersection



Différence

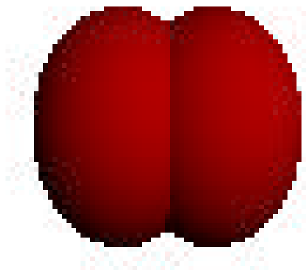


v5 P. CUBAUD <cubaud@cnam.fr> Dept. informatique CNAM, Paris

Exemples en POV

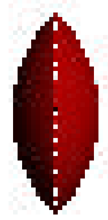
- Union de deux sphères

```
union{
 sphere { <0, 0, 0>, 1
   translate -0.5*x
  }
 sphere { <0, 0, 0>, 1
   translate 0.5*x
  }
  pigment { Red }
}
```



- Intersection de deux sphères (extrusion)

```
intersection {
 sphere { <0, 0, 0>, 1
   translate -0.5*x
  }
 sphere { <0, 0, 0>, 1
   translate 0.5*x
  }
  pigment { Red }
}
```



Représentation interne

- Pour les opérateurs ensemblistes :

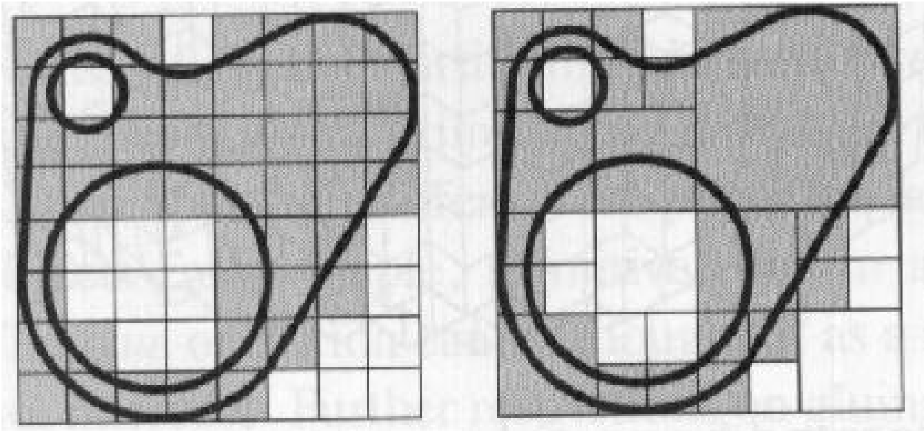
  solide ≠ liste de faces

  solide = ensemble d'éléments de volume

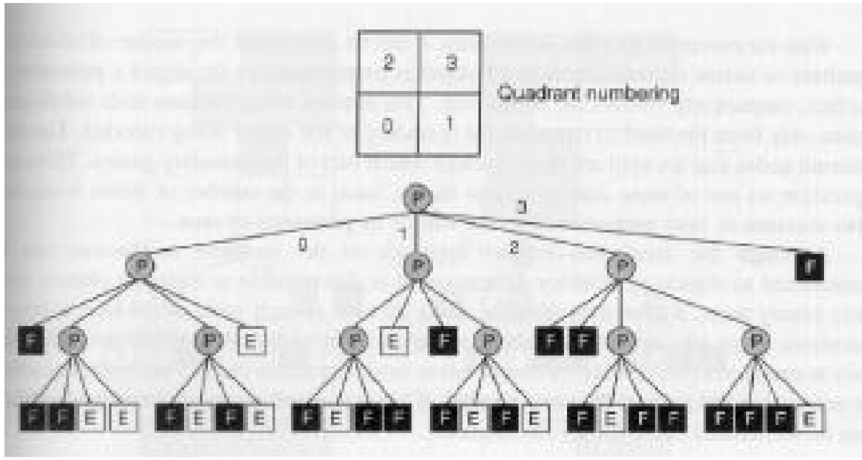
  => suppose une représentation interne adaptée (plus sophistiquée)

  => nombreuses techniques (surtout pour la CAO)

- Les "quadtrees" et les "octrees"



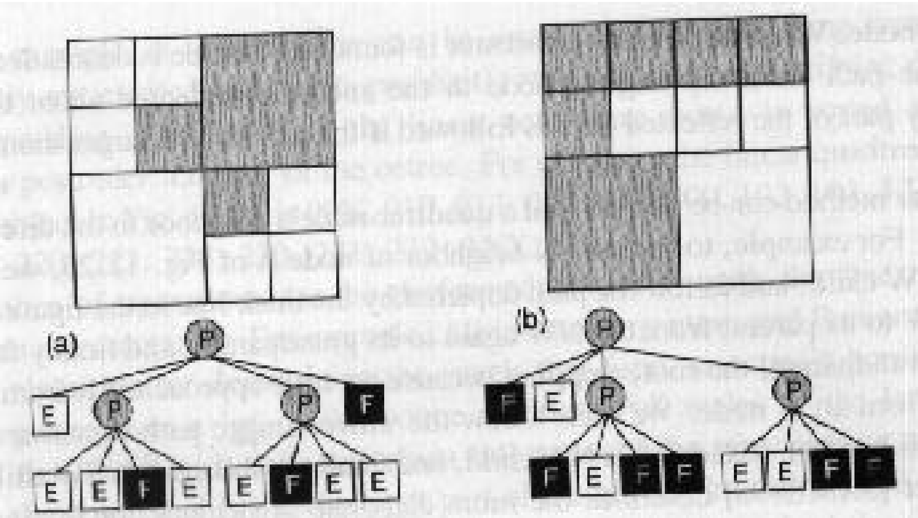
[FOLEY] p. 550-1



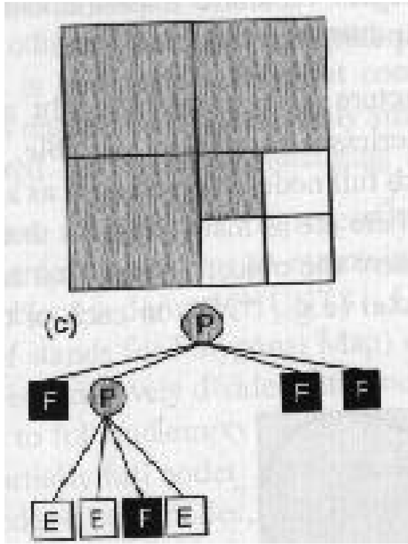
• Exemple d'opérations booléenes :

[FOLEY] p. 553

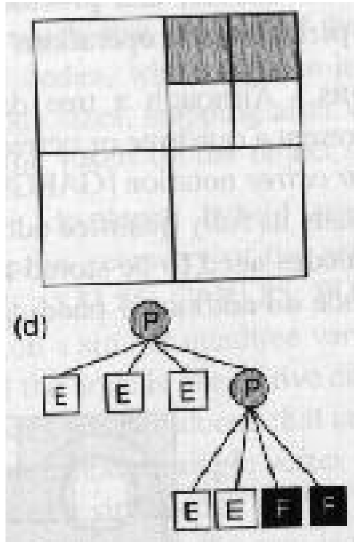
Les objets et leurs quadtrees :



Union



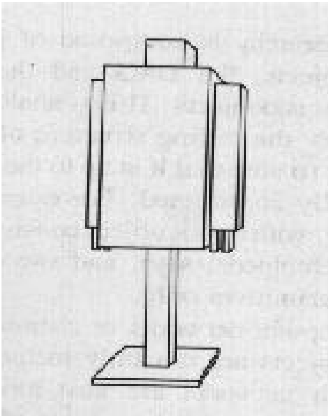
Intersection



4. Hiérarchie d'objets

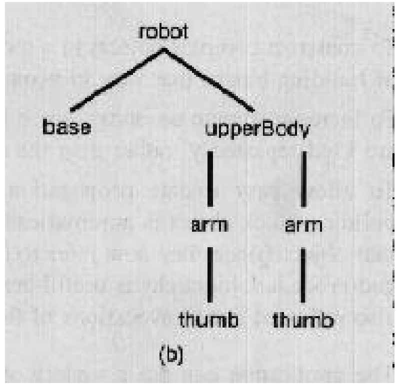
• Le concept de hiérarchie d'objets graphiques est présent dans toutes les bibliothèques de programmation 3D, ainsi que dans les langages spécialisés

• Exemple d'un robot :



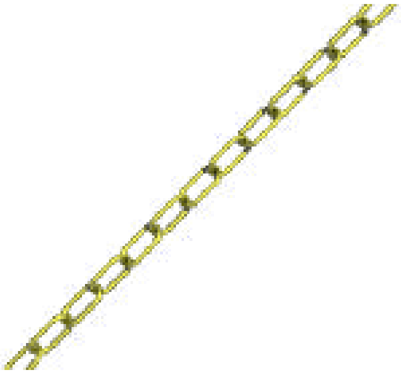
[FOLEY] p. 289

La scène peut être décrite par un <u>arbre</u> :



Exemple 1 : POV

On veut créer un pendantif :



- Structure répétitive : anneau = 2 demi-tores et deux cyclindres

- Les anneaux sont pivotés de 90° à chaque fois.

```
// Copyright 1996 by the POV-Ray Team
#include "colors.inc"
camera {
  location <0, .1, -25>
  look_at 0
  angle 36
}
background { color White }
light_source{ <300, 300, -1000> White }
#declare Half_Torus = difference {
  torus { 4,1
    sturm
    rotate x*-90  // so we can see it from the top
  }
  box { <-5, -5, -1>, <5, 0, 1> }
}
```

```
#declare Flip_It_Over = x*180
#declare Torus_Translate = 8
#declare Link_Translate = Torus_Translate*2-2*y
#declare Chain_Segment = cylinder { <0, 4, 0>, <0, -4, 0>, 1 }
#declare Chain_Gold = texture {
  pigment { BrightGold }
  finish {
    ambient .1
    diffuse .4
    reflection .25
    specular 1
    metallic
  }
}
#declare Link = union {
  object { Half_Torus
    translate y*Torus_Translate/2
  }
  object { Half_Torus
    rotate Flip_It_Over
    translate -y*Torus_Translate/2
  }
  object { Chain_Segment
    translate x*Torus_Translate/2
  }
  object { Chain_Segment
    translate -x*Torus_Translate/2
  }
  texture { Chain_Gold }
}
#declare Link_Pair = union {
  object { Link }
  object { Link translate y*Link_Translate rotate y*90 }
}
#declare Chain = union {
  object { Link_Pair}
  object { Link_Pair translate  y*Link_Translate*2 }
  object { Link_Pair translate  y*Link_Translate*4 }
  object { Link_Pair translate  y*Link_Translate*6 }
  object { Link_Pair translate -y*Link_Translate*2 }
  object { Link_Pair translate -y*Link_Translate*4 }
  object { Link_Pair translate -y*Link_Translate*6 }
}
object { Chain scale .1 rotate <0, 45, -45> }
```

Exemple 2 : VRML

VRML permet de décrire des objets graphiques (au sens large) et leurs actions : des "nodes".

- Les "nodes" sont organisés en une hiérarchie appelée le "scene graph".

- Certains "nodes" affectent le comportement des "nodes" suivants ("rotation" ou "material", par exemple)

- Le "separator" permet de limiiter la portée de telle ou telle propriété à des parties de la scène.

```
#VRML V1.0 ascii
Separator {
   DirectionalLight {
      direction 0 0 -1
   }
   PerspectiveCamera {
      position    -8.6 2.1 5.6
      orientation -0.1352 -0.9831 -0.1233  1.1417
      focalDistance      10.84
   }
   Separator {   # sphere 1
      Material {
         diffuseColor 0.25 0.50 0.25   # Red
      }
      Translation { translation 3 0 1 }
      Sphere { radius 2.3 }
   }
   Separator {  # The blue cube
      Material {
         diffuseColor 0 0 1  # Blue
      }
      Transform {
         translation -2.4 .2 1
         rotation 0 1 1  .9
      }
      Cube {}
   }
}
```

Représentation interne

Exemple pour PHIGS, extrait de [FOLEY] p. 318