

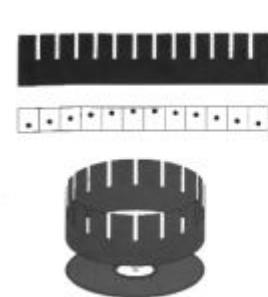
# Animation Habillage

C Gentil / M Neveu

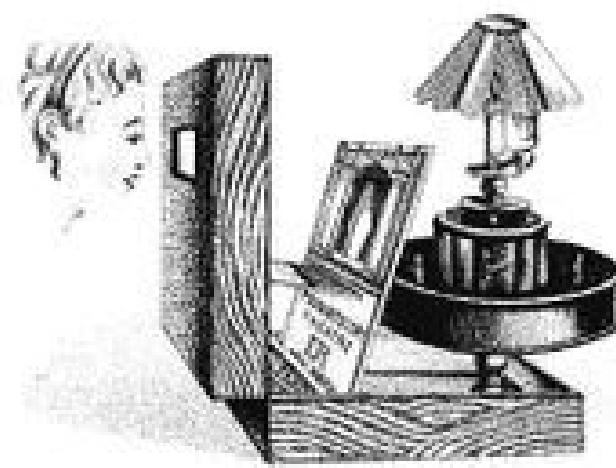
# Les précurseurs



Le phénaskistiscope  
Joseph Plateau, 1833

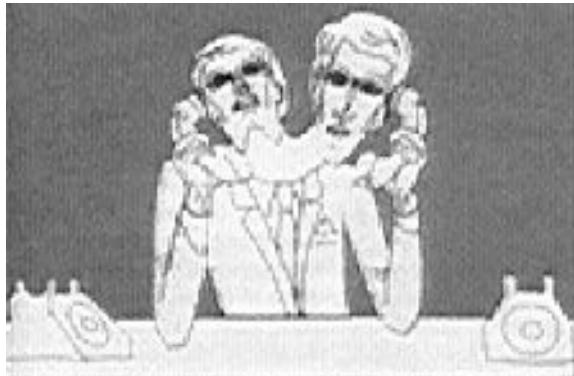


zootrope W. G. Horner, 1834

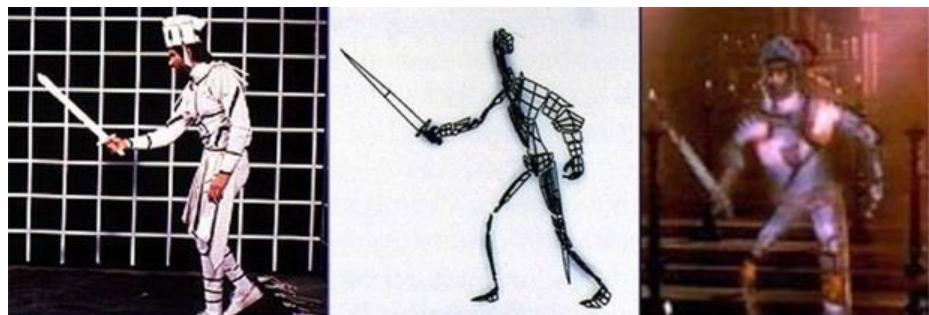


praxinoscope Émile Reynaud 1877

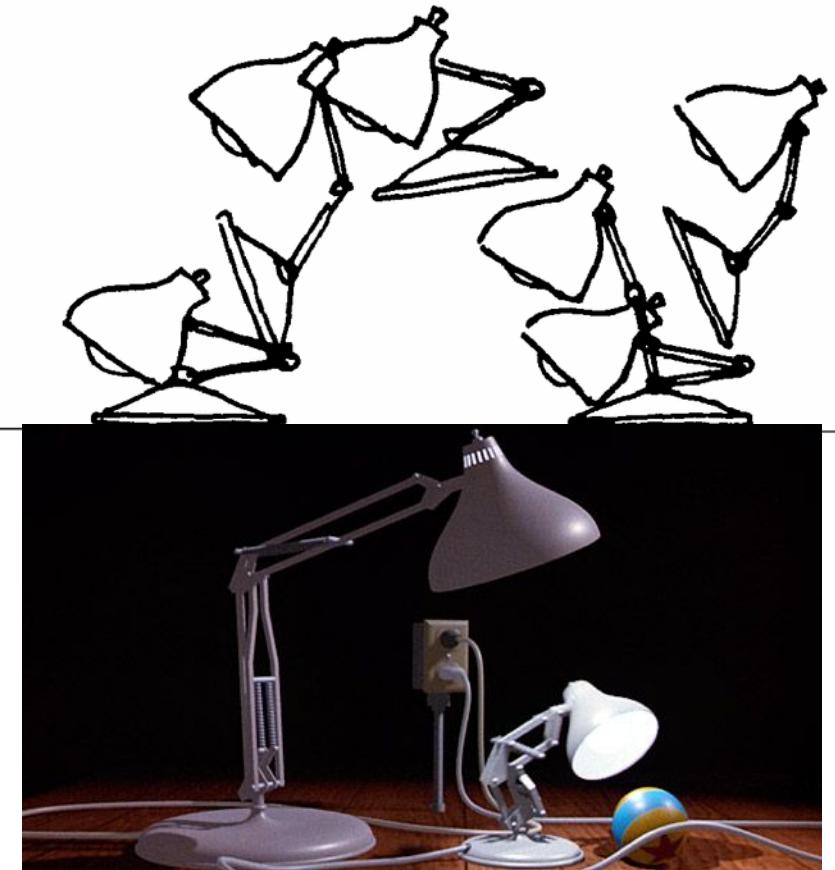
# Le numérique- les débuts



**La Faim** - Peter Foldes (1974)



secret de la pyramide - Barry Levinson (1985)



Luxo. Pixar.  
(1986)

# Le numérique



The adventures of André and Wally B.- John Lasseter (1984)



Jurassic Park – Steven Spielberg (1993) **Geri's game** - Jan Pinkava (1997)



Pixar



Shrek - Dreamworks  
(2001)



Ice Age - Blue Sky Studios (2002)



The Minions - Illumination Entertainment / Illumination Mac Guff (2015)



BUF- 2006



Le Voyage de Chihiro - H. Miyazaki - studio Ghibli (2001)



Le petit prince - M. Osborne - On Animation Studios (2015)

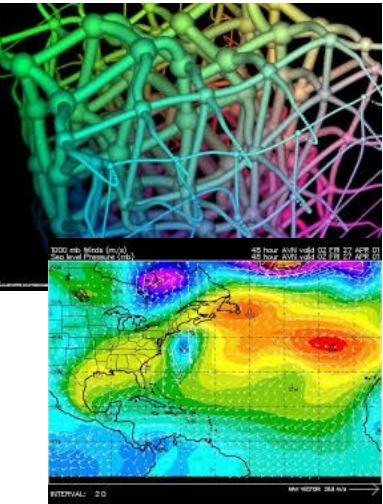


Mac Guff - 2006

# Applications



Audiovisuel



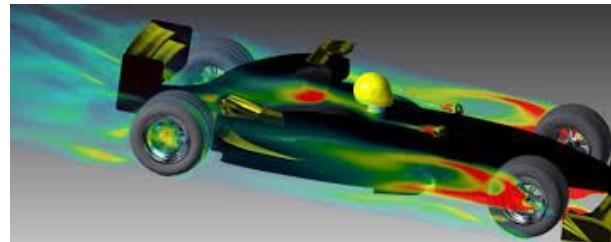
Visualisation  
scientifique



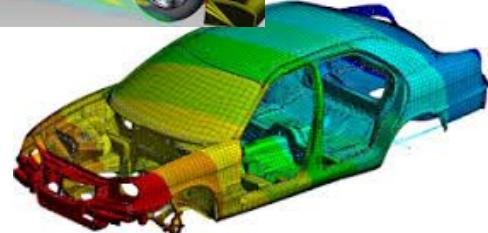
Jeux video



Serious games



Simulation



# Les différentes techniques

- Interpolation entre positions clés
  - Très souple mais fastidieux
- Capture de mouvements
  - Grande qualité mais spécifique
- Édition de mouvement
  - Pratique mais limité et risqué
- Modèles générateurs
  - Automatique mais complexe
- Idéalement, automatisé avec un contrôle précis de l'animateur

# Capture de Mouvement

## Optique :

capteurs réflectifs (**marqueurs**) attachés au corps de la vraie personne.

positions des marqueurs → positions correspondantes pour le modèle animé.

Ex : petits capteurs aux articulations d'une personne et on enregistre la position de ces capteurs selon plusieurs directions.  
d'où la reconstruction des positions 3D de chaque point-clé à chaque instant.

Avantages : liberté de mouvement, pas de câblage.

Inconvénients : occlusions, difficulté de distinguer 2 marqueurs proches

Solutions : plus de caméras, mais le coût devient prohibitif.

La plupart des systèmes opèrent avec 4-6 caméras.

Exemple de systèmes optiques: Elite, MultiTrax.

# Capture de Mouvement

## Magnétique :

une personne porte un ensemble de capteurs magnétiques (capables de mesurer leur relation spatiale par rapport à un transmetteur magnétique centralisé).

La position et l'orientation de chaque capteur sont utilisées ensuite pour faire mouvoir l'acteur digital.

=> synchronisation des récepteurs.

Ex : pour le mouvement du corps humain, 11 capteurs sont nécessaires:

- un sur la tête
- un sur chaque bras
- un sur chaque main
- un autre au centre de la poitrine
- un sur le bas du dos
- un à chaque cheville
- un sur chaque pied

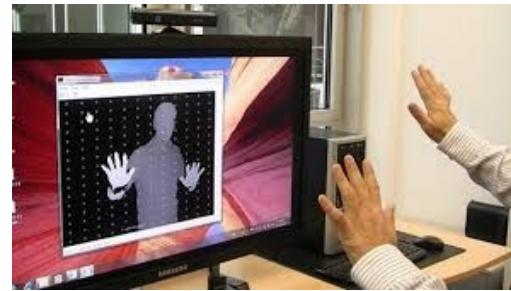
On utilise la cinématique inverse pour calculer le reste de l'information nécessaire.

Les systèmes les plus populaires sont: Polhemus Fastrack et Ascension Flock of Birds

# Capture de Mouvement

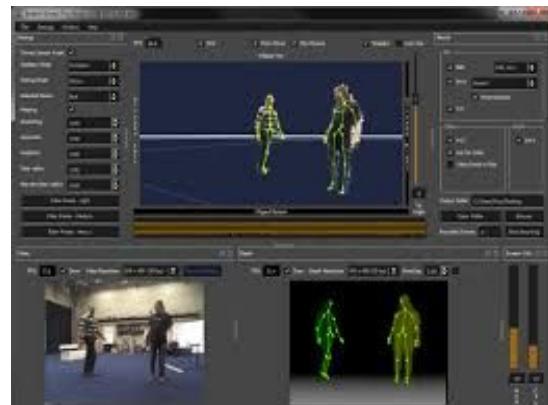
Avec des caméras de type Kinect :

- Pas de marqueurs
- Calibrage
- Surfaces lambertiennes
- Peu coûteux et logiciels nombreux





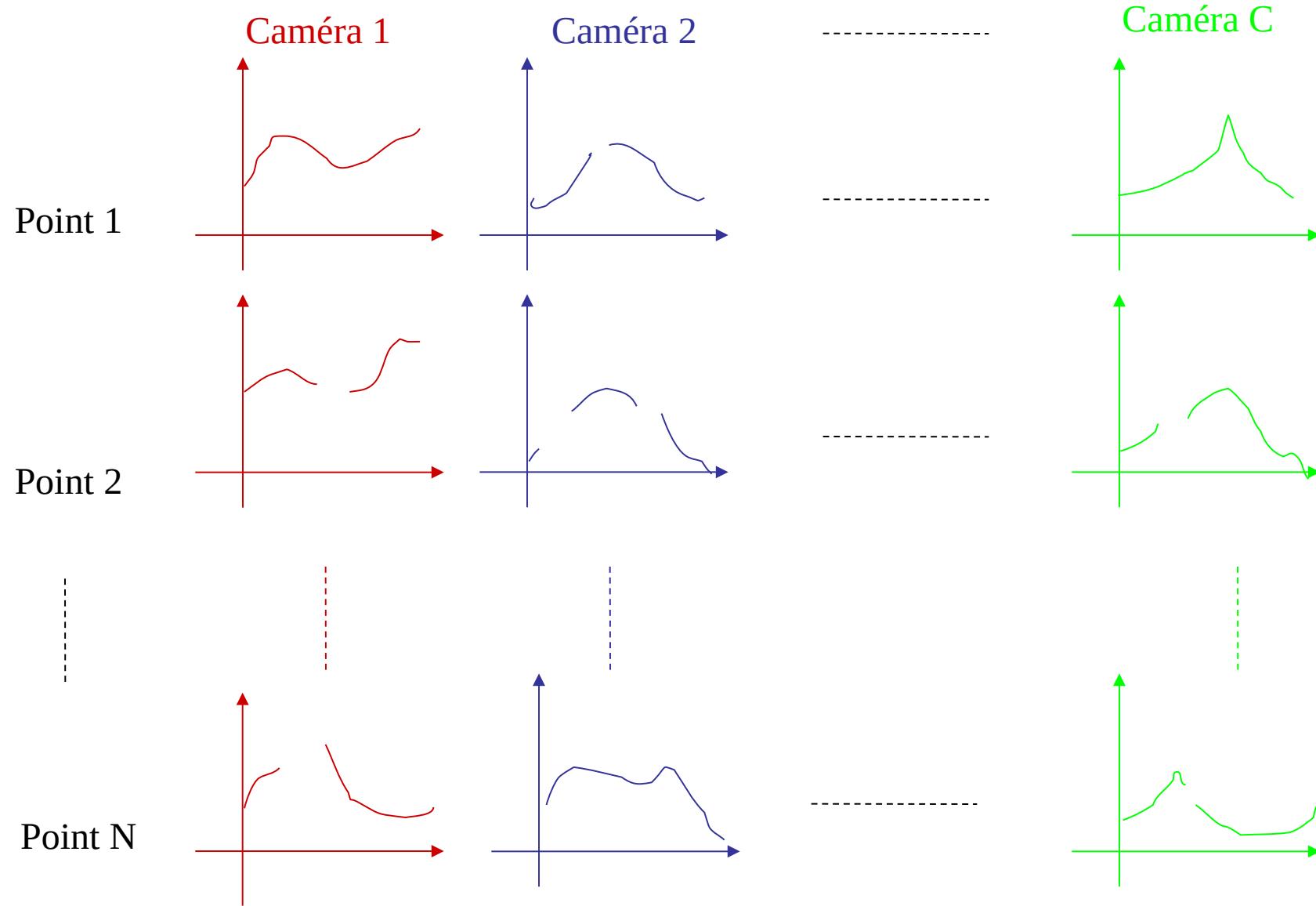
MOCAP :  
environnement contraint,  
avec ou sans marqueurs



# Mise en correspondance

N points  $P_j$  repérés dans C repères 2D  $R_i$  ( $x_i^j, y_i^j$ )  $\rightarrow$  N points 3D ( $x_j, y_j, z_j$ )

$i=1..C, j=1..N$



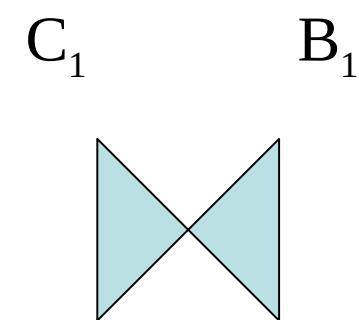
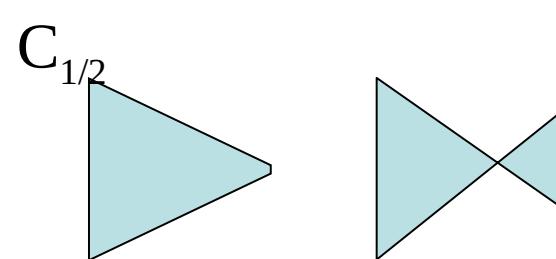
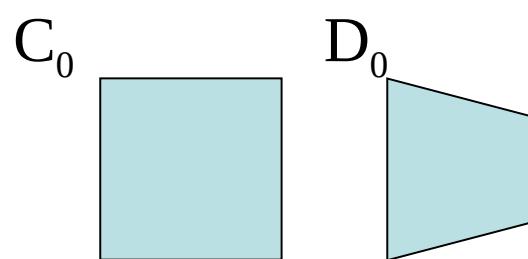
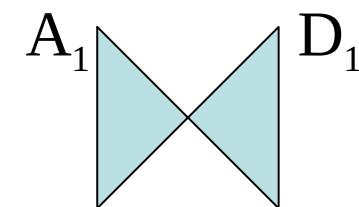
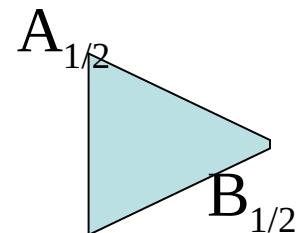
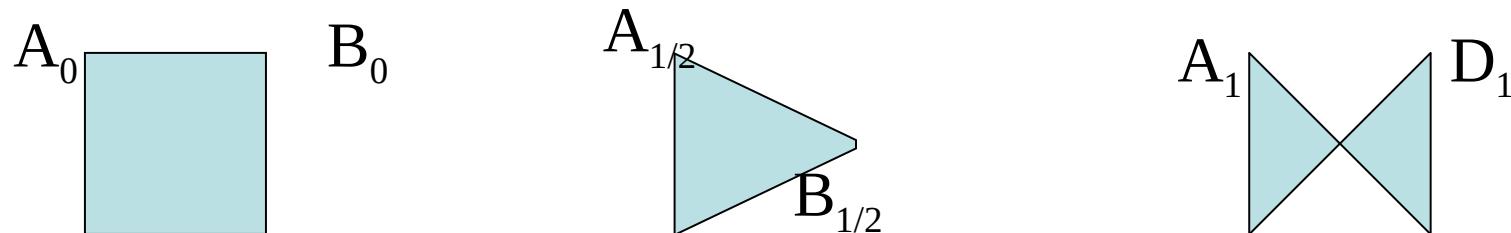
# Animation par images-clés

## Principe

En entrée : une série d'images à des temps donnés

En sortie : les images intermédiaires par interpolation.

Ex : transformer une forme géométrique en une autre lors d'une animation.



T = 0

$\frac{1}{4}$

$\frac{1}{2}$

$\frac{3}{4}$

1

# Animation par images-clés

Soient N1 et N2 les nombres de sommets des 2 lignes polygonales.

Si  $N1 > N2$  alors

$$RT := (N1 - 1) \text{ div } (N2 - 1)$$

$$RS := (N1 - 1) \text{ mod } (N2 - 1)$$

on ajoute RT points aux RS premiers segments et RT-1 aux autres.

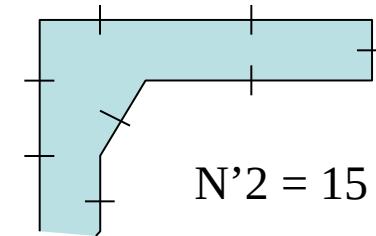
$N1 = 15$



$N2 = 7$

$N2 = 7$

$RT = 2, RS = 2$



$T = 0$

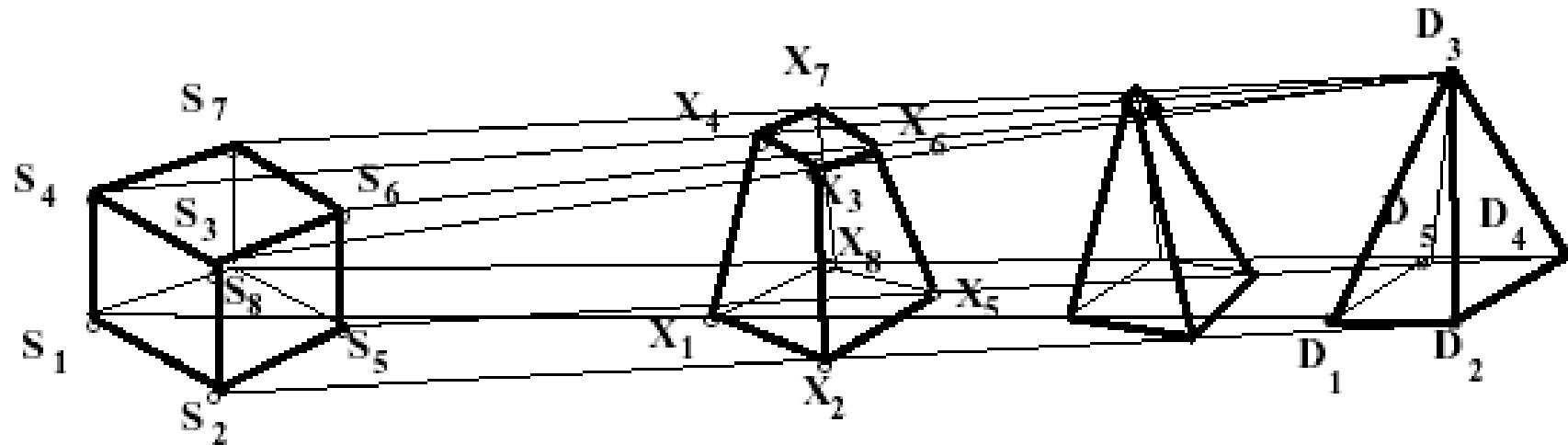
$\frac{1}{2}$

$1$

# Animation par images-clés

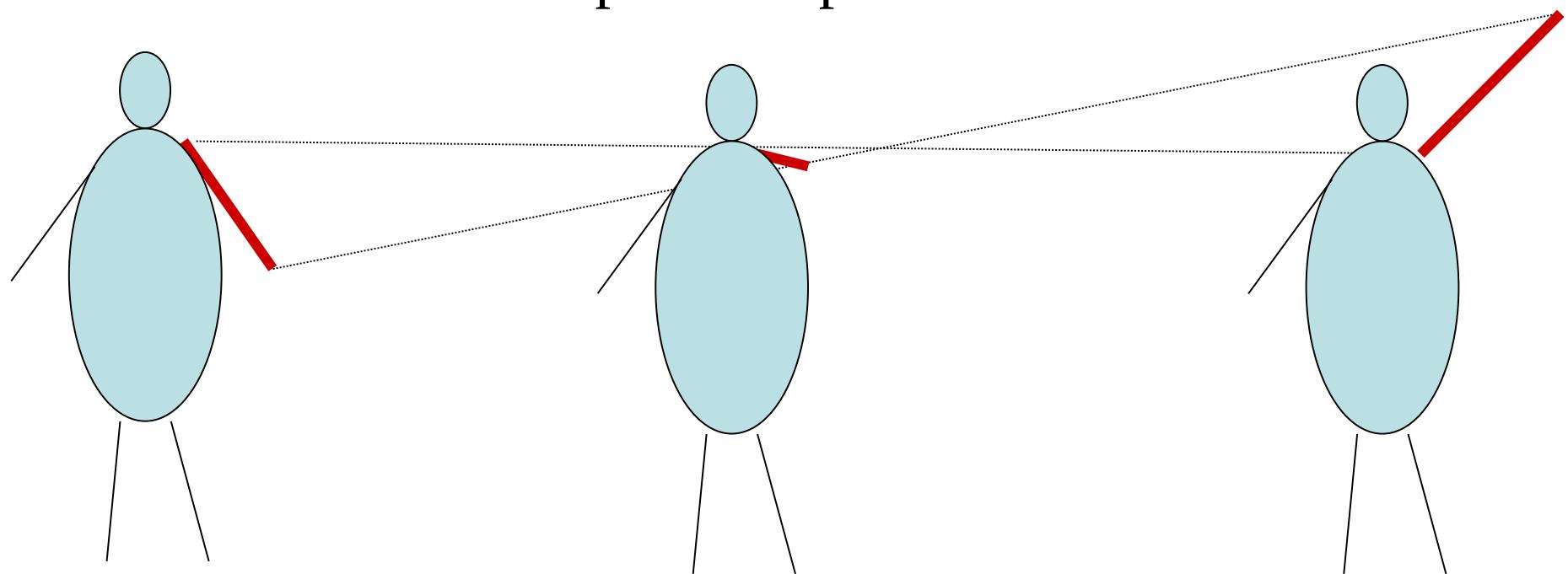
morphing 2D : extension de la méthode des images-clés à une interpolation selon les pixels plutôt que les sommets de figures.

- La méthode de prétraitement vue pour les segments peut être étendue aux facettes d'objets 3D
- mais plus complexe car il faut assurer une correspondance entre facettes et sommets. Il faut donc ajouter des facettes et des sommets pour que les 2 images clés en aient le même nombre (= morphing 3D).



# Animation par images-clés

Problème : si on interpolate des positions



=>Interpolation paramétrique (paramètres de position et de forme, lumière, caméra, etc...)

# Animation procédurale

- Algorithme décrivant le mouvement

Ex : chute libre d'un objet

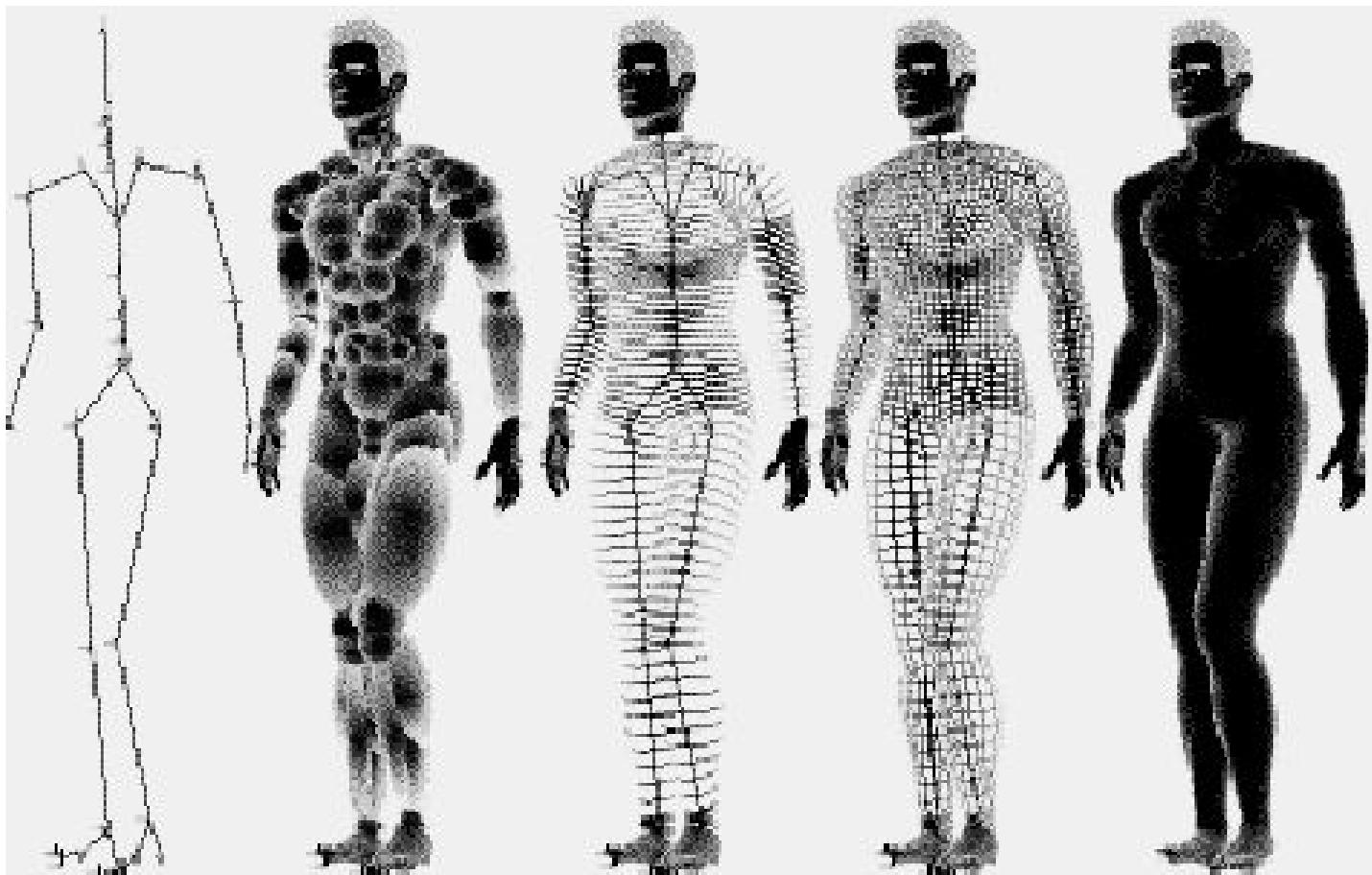
```
créer OBJET (...);  
TEMPS = 0;  
tantque Y > 0  
    Y = INITIALE - ½*G*TEMPS^2  
    déplacer (OBJET, X,Y,Z);  
    dessiner OBJET;  
    enregistrer l'image  
    effacer OBJET  
    TEMPS:=TEMPS+1/25;
```

≈simulation

Ex : pendule

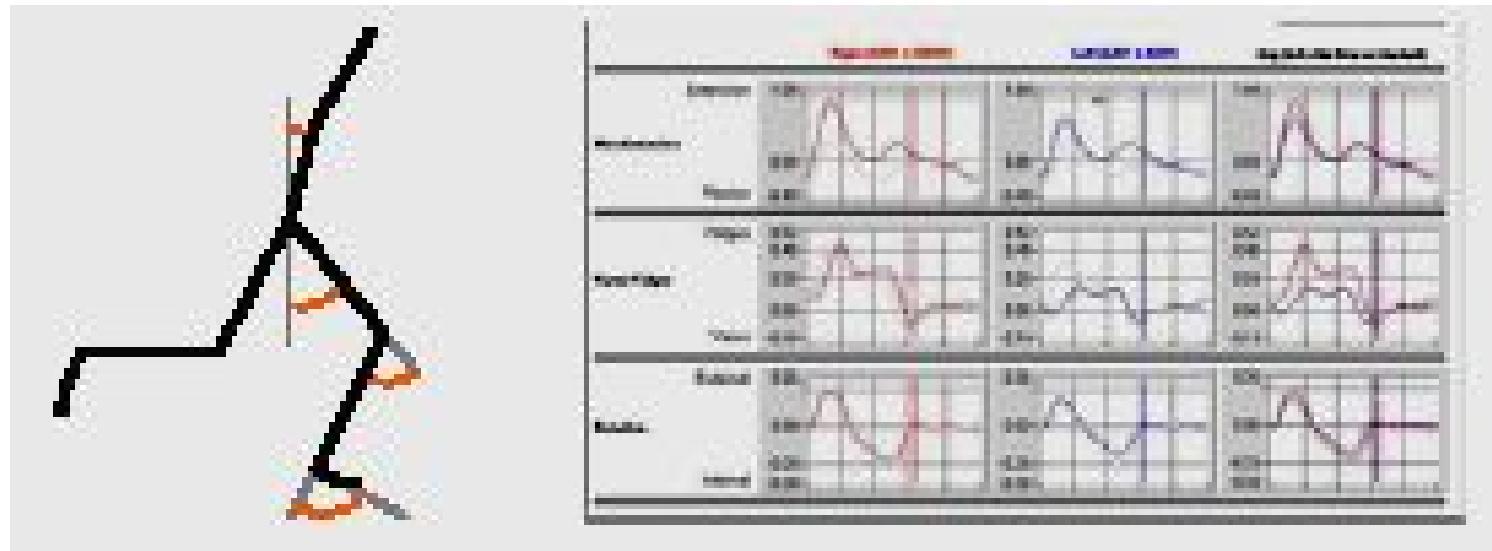
```
créer HORLOGE (...);  
pour IMAGE:=1 à NB_IMAGES  
    TEMPS:=TEMPS+1/25;  
    ANGLE:=A*SIN (OMEGA*TEMPS+PHI);  
    MODIFIER (HORLOGE, ANGLE);  
    dessiner HORLOGE;  
    enregistrer l'image  
    effacer HORLOGE
```

# Squelette + habillage



# Interpolation de positions clefs

- Compléter lorsqu'on a assez d'information
- Interpoler position, angle, forme, transparence...
- Demande une grande technique



# Fonctions d'interpolation

- Modifications des points de contrôle
- Influence plus ou moins locale
- Interpolation vs approximation
- Forte continuité ou points de rebroussement
- Dérivées réglables (tension, biais contrôle)

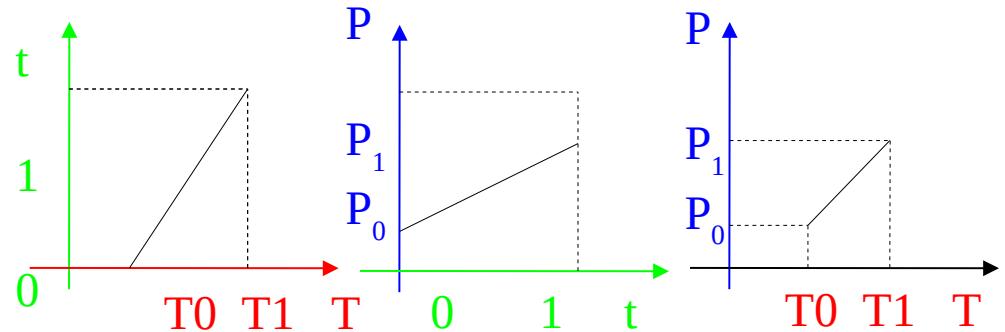
Splines de Catmull-Rom de Kochanek-Bartels, etc...



# Paramétrisation de la courbe

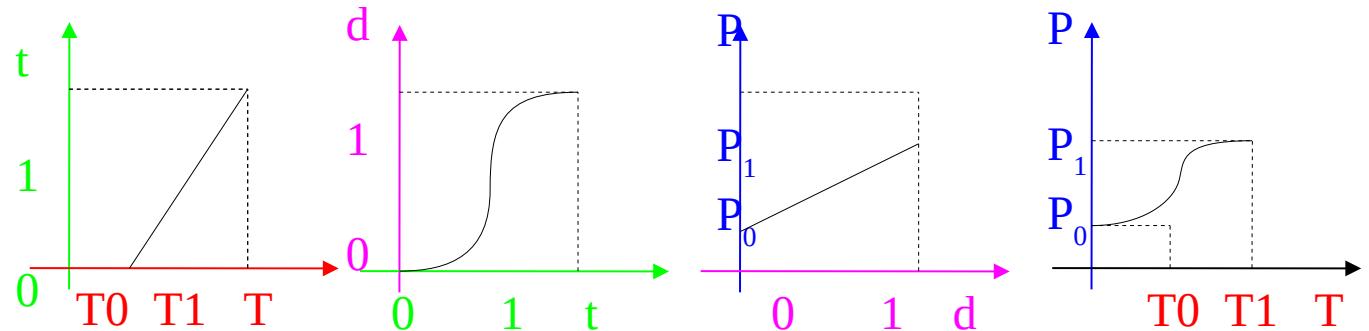
## Paramétrisation

- Par morceaux  
 $P(T) = P(t(T))$

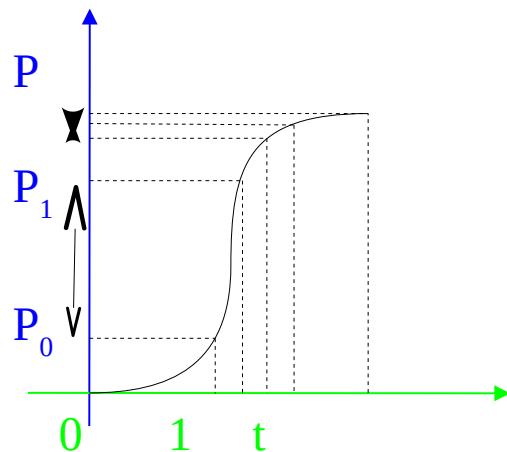


- Contrôle de la vitesse : on introduit distance =  $f(\text{temps})$

$$P(T) = P(d(t(T)))$$



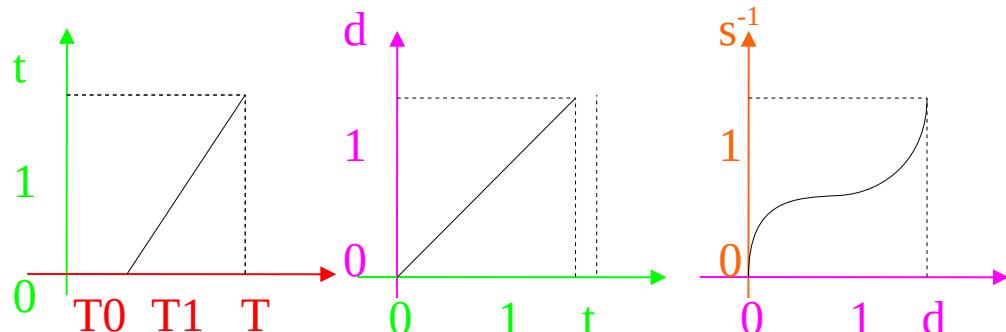
# Paramétrisation de la courbe



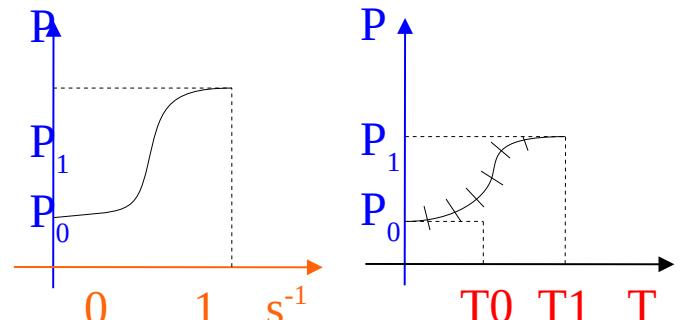
$\Delta t$  cst  $\rightarrow \Delta P$  non cst

« Redressement » avec abcisse curviligne

$$s(t) = \int_0^t \left\| \frac{dP}{du} \right\| du$$

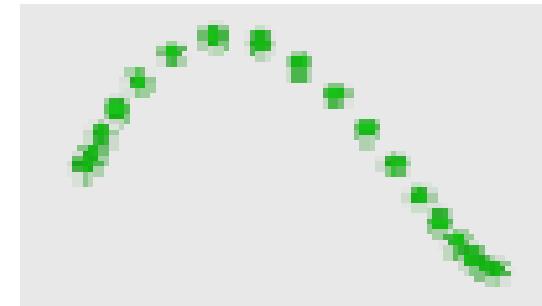
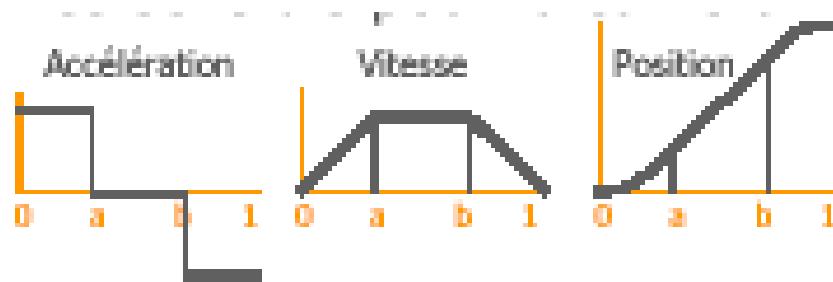


$$P(T) = P(s^{-1}(d(t(T))))$$



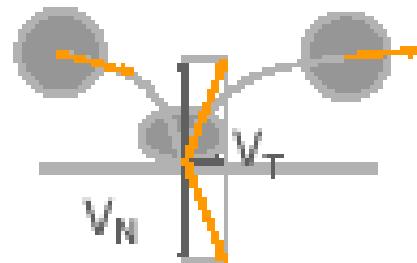
# Paramétrisation de la courbe

- Vitesse non constante
- Reparamétrisation par la longueur d'arc
- Calcul analytique ou discret
- Vitesse réglable
- Départ et fin souples (Ease In/Out)



# Effets

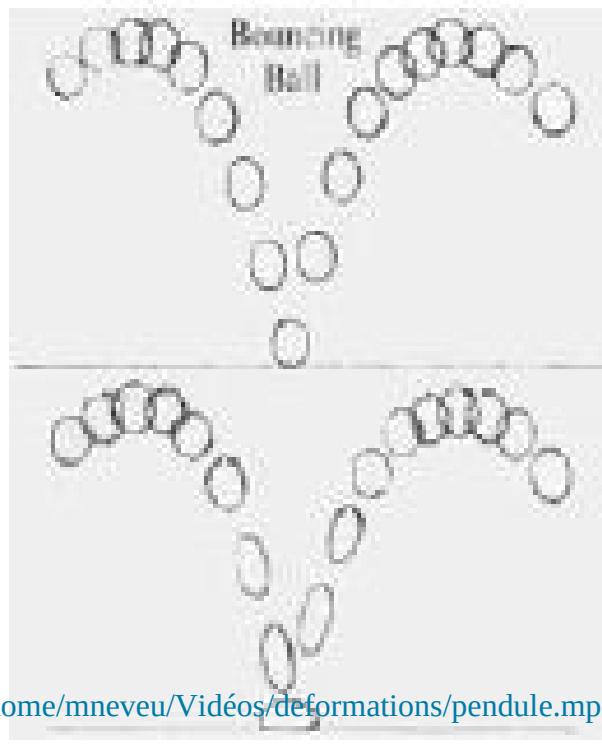
- Anticipation des mouvements
- Rebonds, points d'inflexion



$$v'_T = \alpha v_T$$
$$v'_N = -\beta v_N$$

- Adapter la vitesse à la masse
- Etirements / écrasements (volume constant)
- Bruiter le mouvement

# Effets



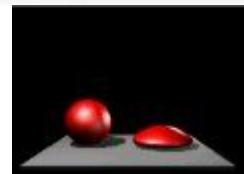
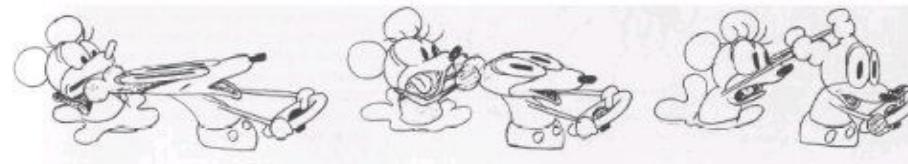
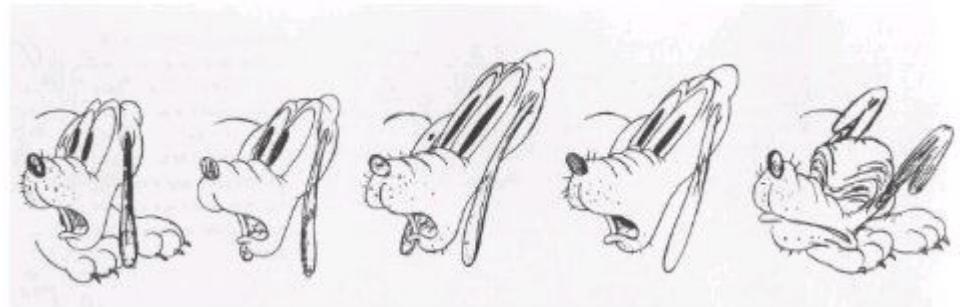
/home/mneveu/Vidéos/deformations/pendule.mpe  
g

*Principles of traditional animation applied to 3D computer animation.* J. Lasseter.  
ACM Computer Graphics, 21(4), 1987.

- ▶ squash and stretch
- ▶ slow in and out
- ▶ timing
- ▶ arcs
- ▶ anticipation
- ▶ exaggeration
- ▶ staging
- ▶ secondary action
- ▶ follow through and overlapping action
- ▶ appeal
- ▶ straight ahead and pose-to-pose action

# Squash and Stretch

- Extension/  
écrasement  
donnent de la vie  
(les êtres vivants se  
déforment quand ils  
se déplacent)
- Souvent : volume  
constant



# Timing

- Le nombre d'inbetweens donne un aspect plus ou moins lisse au mouvement
- Utilisé pour rendre +/- rapide, +/- dynamique, +/- heurté,....
- Change l'expressivité



# Anticipation

- préparation physique
- pour expliquer
- pour attirer l'attention
- pour accélérer le mouvement qui suit
- pour souligner la difficulté d'une action

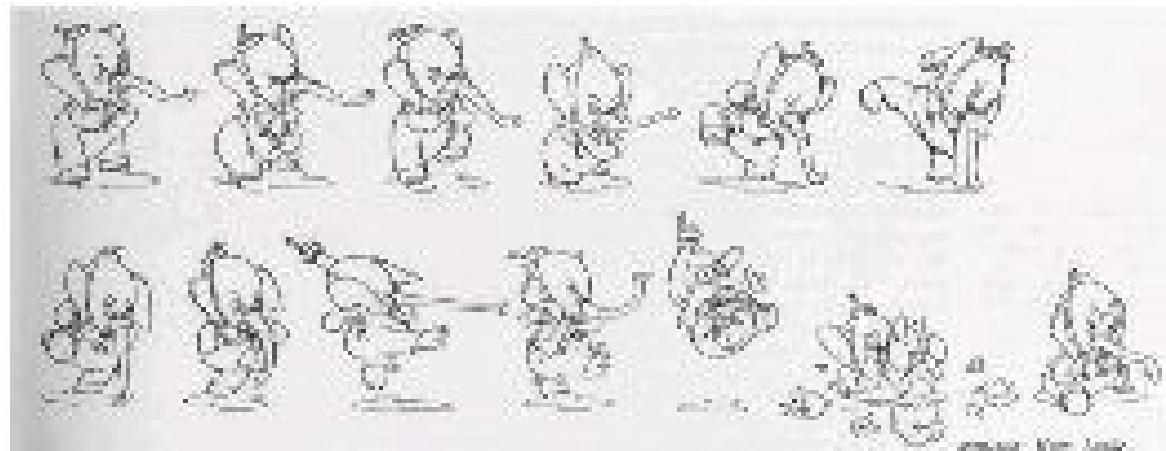


# Anticipation



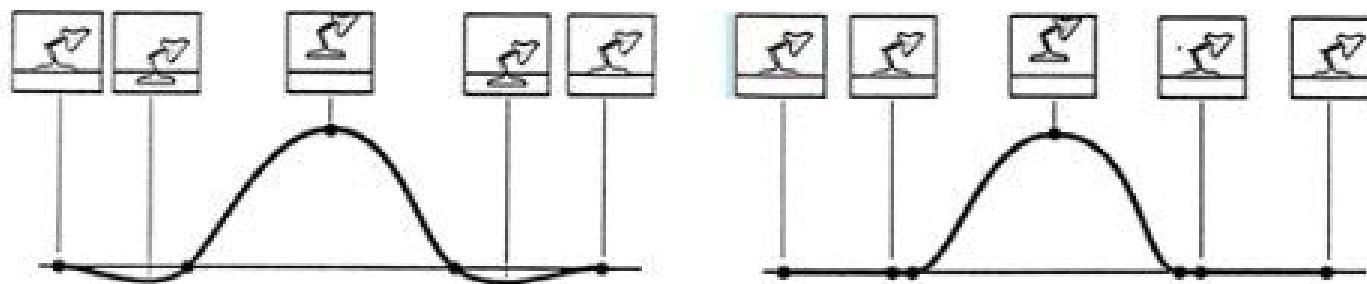
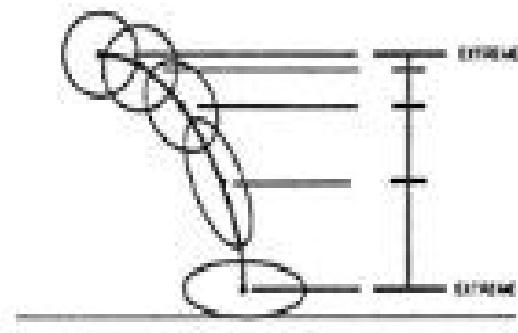
# Follow through, overlapping action

- FT : terminaison d'une action (jambe reste en l'air après un tir au foot)
- succession d'actions : yeux-tête-corps, bras-avantbras-main, ....
- Découpage de parties du corps par FTOA

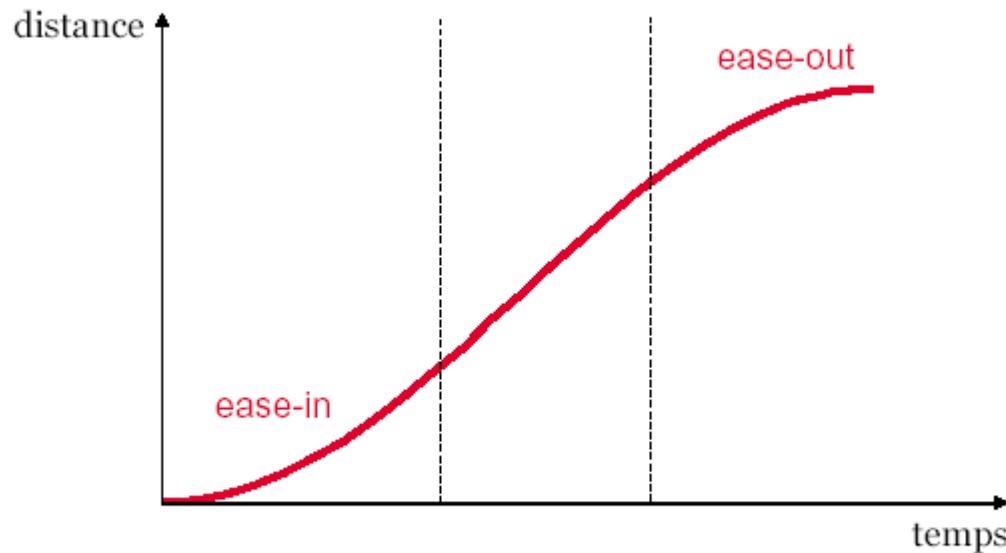


# Slow in slow out

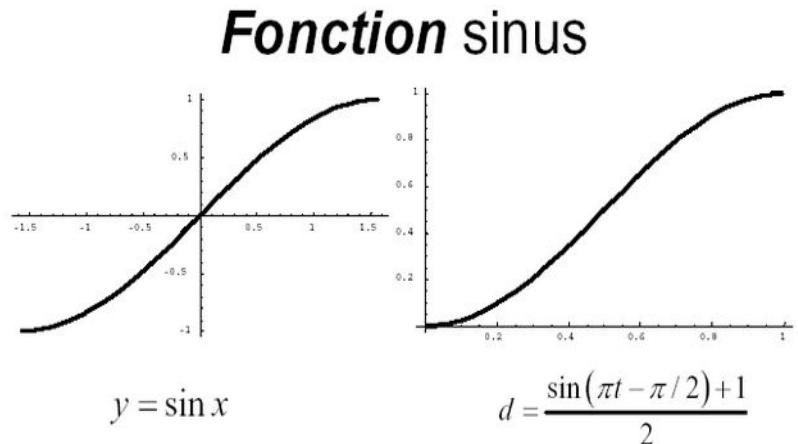
- Flow-chart pour placer les images intermédiaires
- Gestion de la continuité du mouvement par splines (accélération/ralentissement)
- Corrections



# Contrôle de la vitesse



Lissage de la progression :  
accélération (ease-in) et décélération  
(ease-out)

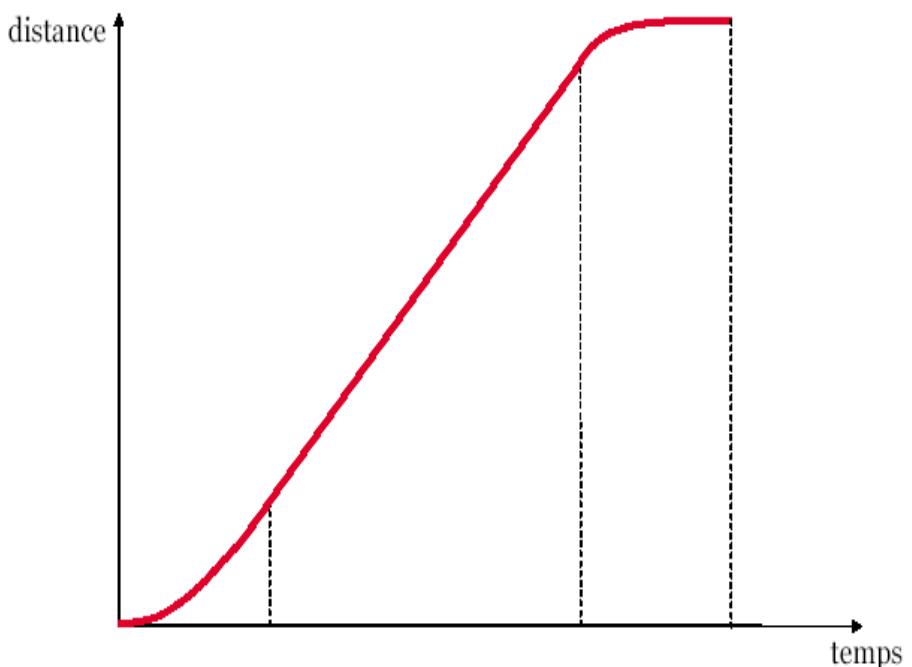
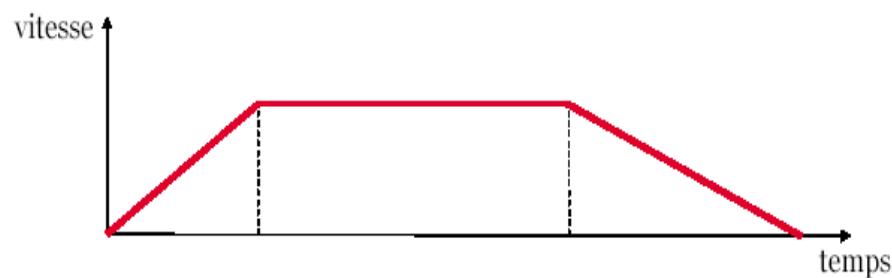
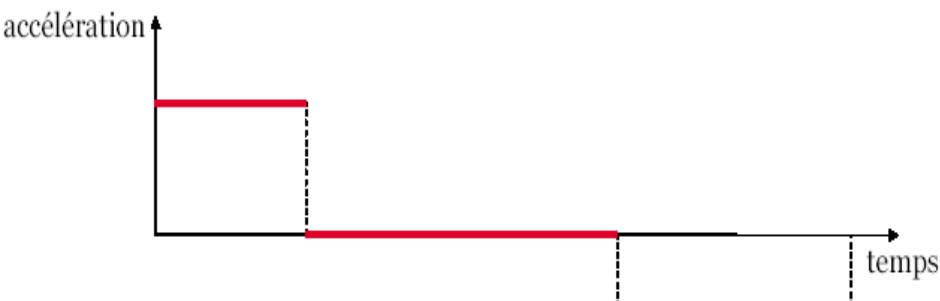


Courbe distance / temps **normalisée**.

$$0 \leq t \leq 1 \quad \text{et} \quad 0 \leq d \leq 1$$

# *Accélération, vitesse, ...*

*..., distance*



# Récréation



Luxo - John Lasseter (1986)



Geri's game - Jan Pinkava (1997)



[https://www.youtube.com/watch?v=K\\_xT4H1rFn8](https://www.youtube.com/watch?v=K_xT4H1rFn8)

The Adventures of André and Wally B (1984)



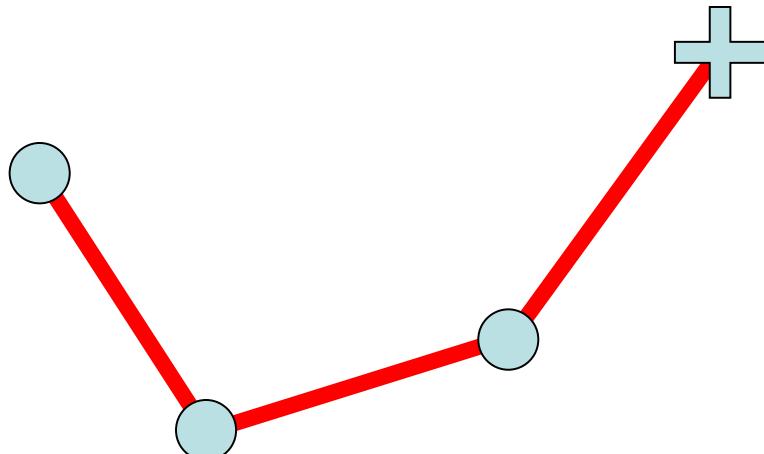
For the Birds - Ralph Eggleston (2000)

# Cinématique directe

- Objet décomposé en une hiérarchie de repères
- Humain, animaux, robots...
- Interpolation de positions
- Représentation des orientations
  - Matrice 3x3 et angles d'Euler mal adaptés
  - Utiliser les quaternions et l'interpolation par Slerps
- Composition depuis la racine

# Cinématique inverse

- Mouvement fixé par la position d'une extrémité



# Modèles hiérarchiques

# Modélisation procédurale

- Comme le bonhomme de neige
- Problèmes :
  - Modèle de plus en plus complexe
  - Écrire le code sans erreurs
  - Modifier le modèle
  - Manipuler le modèle

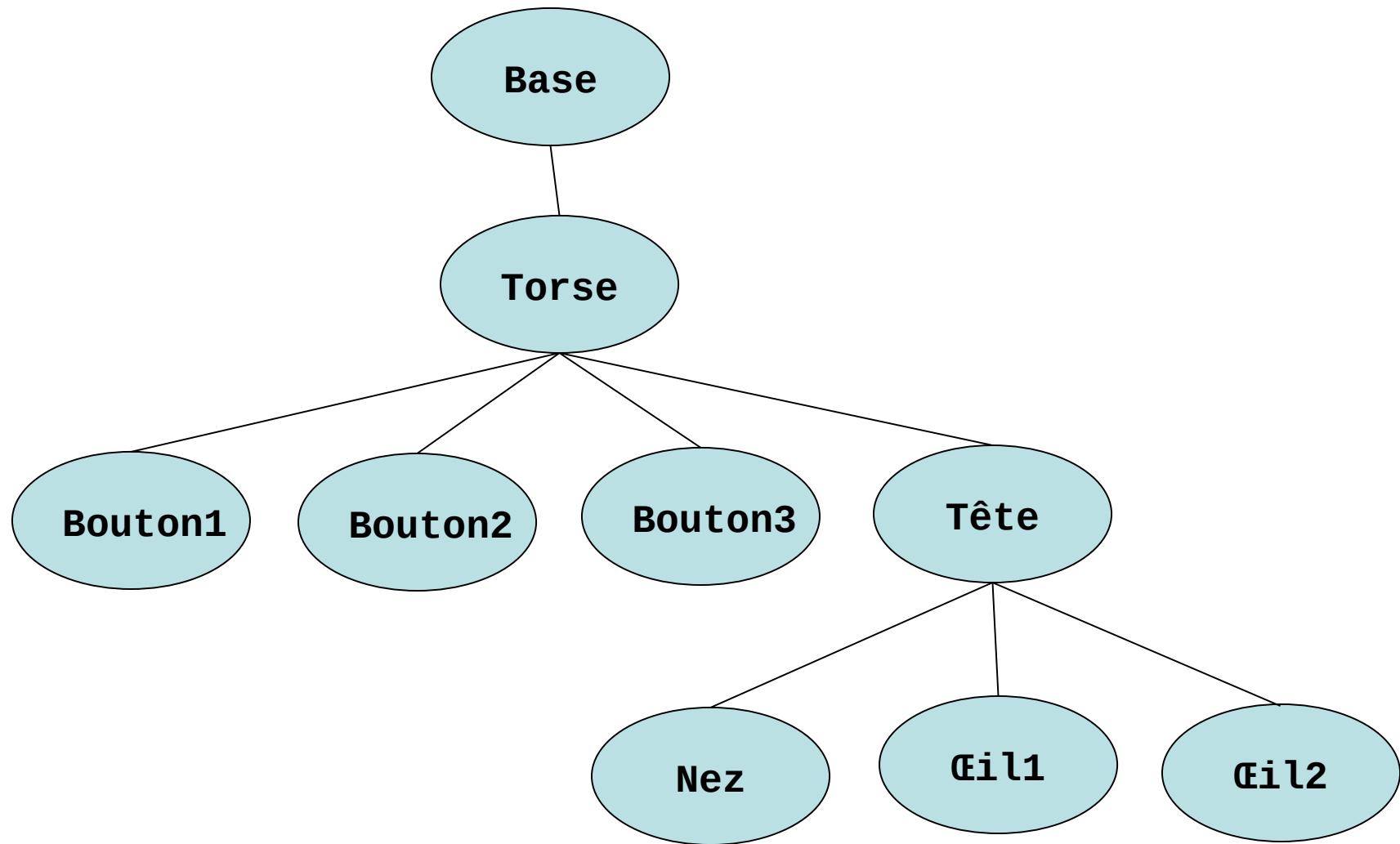
# Modèle hiérarchique

- Modèle divisé en morceaux
- Relations entre les morceaux
- Paramètres pour :
  - Forme des morceaux (écrasement)
  - Position des morceaux (inclinaison)
- Hiérarchie du modèle :
  - Base se déplace : tout suit
  - Torse se déplace : tête et boutons suivent
  - Tête se déplace : yeux et nez suivent

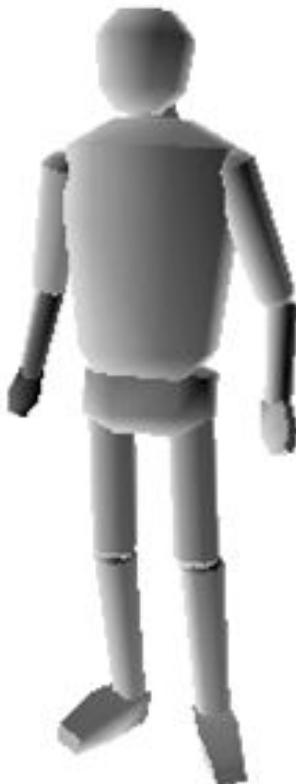
# Hiérarchie de nœuds

- Chaque nœud :
  - Transformation en coordonnées locales
  - Modèle simple en coordonnées locales
  - Dessin des enfants (qui héritent des coordonnées locales)
- Après le nœud :
  - Retour aux coordonnées locales précédentes
  - Couleur, matériau, style, etc. ?
    - Pas fixé

# Bonhomme de neige



# Personnage



- Squelette
- Hanches
  - Hanche1
  - Buste
    - Buste1
    - Cou
    - Tête
  - Buste2
    - Epaulegauche
    - Avbrasgauche
    - Brasgauche
    - Maingauche
  - Buste3
    - Epauledroite
    - Avbrasdroit
    - Brasdroit
    - Maindroite
  - Hanche2
    - Cuissegauche
    - Jambegauche
    - Piedgauche
  - Hanche3
    - Cuissedroite
    - Jambedroite
    - Pieddroit

# Descente de l'arbre

- Méthode simple:

```
void draw(node) {  
    glPushMatrix();  
    glTranslate(..., ..., ...);  
    glRotate(..., ..., ...);  
    drawGeometry(node);  
    for (i=0; i<numChildren; i++)  
        draw(children[i]);  
    glPopMatrix();  
}
```

- Convient bien aux langages à classe (Java, C++)
  - Méthode de base object.draw(),...

# Intérêts des arbres

- Structure hiérarchique sur l'objet
- Boites englobantes :
  - Construites hiérarchiquement
  - Collision, contact
  - Affichage/ Niveau de détail (LOD)
  - Sélection à la souris
- Édition interactive du modèle

# Inconvénients des arbres

- Trop limité :
  - Morceaux bien distincts
  - Trous, pénétration, discontinuités
  - Besoin de suivre toute la hiérarchie
- Deux extensions :
  - Graphes de scène
  - Squelette

# Graphes de scène

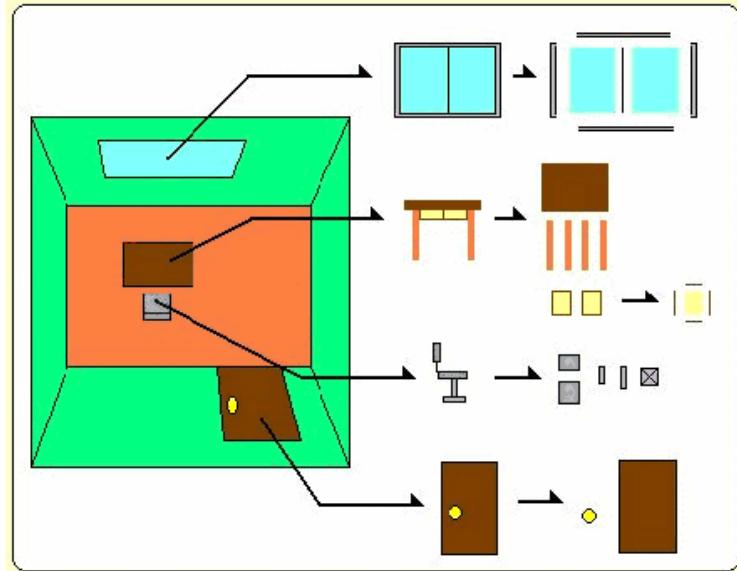
- issus de l'architecture des boîtes à outils 3D [Open Inventor ~ 1990] .
- avec des librairies graphiques de bas niveau, manipulation d'entités de plus haut niveau
- Principe : graphe acyclique orienté pour organiser hiérarchiquement une scène 3D
- structure arborescente manipulée par sa racine ; nœud = un élément de la scène à représenter

# Graphes de scène

- avantages et possibilités :
- adéquation avec la programmation objet et bonne structuration de la scène;
- optimisation et simplification de la gestion de l'ordre d'affichage et du détourage («clipping»);
- transmission et partage aisés du contexte graphique (propriétés) entre nœuds parents et enfants.

# Graphes de scène

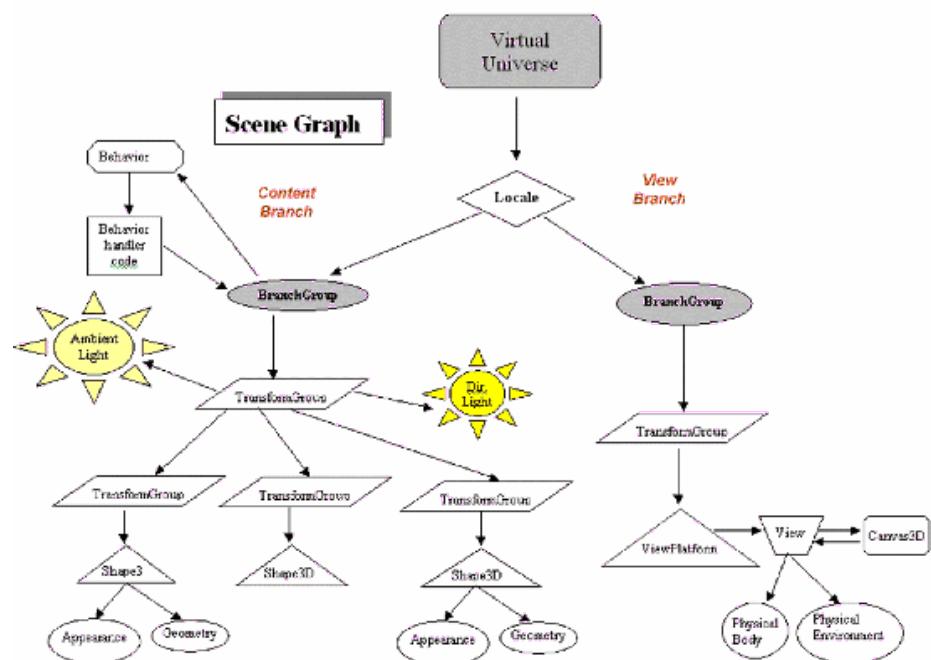
- Nœuds distincts :
  - Caméra
  - Sources lumineuses
  - Transformations
  - Groupes
  - Surfaces
  - Matériaux
- Surfaces simples
  - Sphère, cône, cylindre, cube...
  - Paramètres donnés par transformation précédente



(a) Une organisation sémantique.

(a) présente un graphe de scène ne représentant que les objets géométriques de la scène.

Cette visualisation permet d'appréhender clairement l'organisation de la scène.

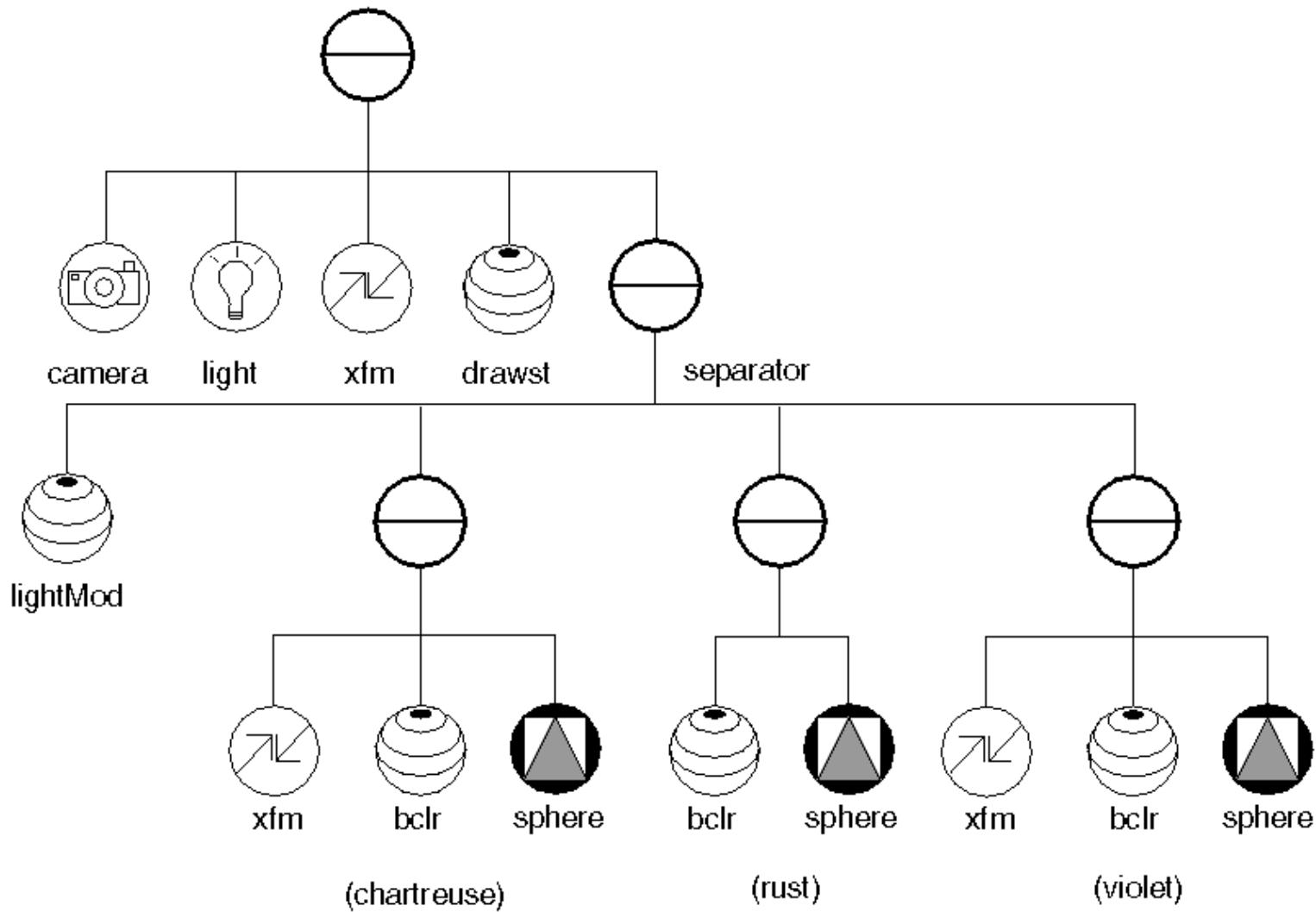


(b) Une représentation complète.

(b) est une représentation complète d'un graphe de scène,

faisant apparaître les transformations et autres nœuds graphiques.

# Graphe de scène OpenInventor



# Graphe de scène Java3D

