

# Images de Synthèse Animées

Nicolas Holzschuch  
Cours d'Option Majeure 2  
Nicolas.Holzschuch@imag.fr

---

---

---

---

---

---

---

---

## Plan

- Plan du cours
  - Questions techniques
- Vue d'ensemble de la synthèse
- Coordonnées homogènes
- Modélisation paramétrique

2

---

---

---

---

---

---

---

---

## Plan du cours (1)

- Techniques d'image de synthèse
  - Techniques de base
  - Formant un tout cohérent
  - Vues de façon très pratique
- La technique n'est pas tout :
  - Art
  - Interface utilisateur
  - Non traités en cours, mais essentiels

3

---

---

---

---

---

---

---

---

## À propos de vous

- Motivations, envies ?
- Arrière-plan, connaissances :
  - Informatique :
    - Majeure 1 Informatique ?
    - C, C++ ?
    - OpenGL ?
  - Math/Physique :
    - ODE ? RK4 ? Maxwell ? Cinématique inverse ?

4

---

---

---

---

---

---

---

---

## Questions techniques

- Les TD :
  - Avec OpenGL
  - Langage C
- Examen :
  - Oral
  - Projets ?

5

---

---

---

---

---

---

---

---

## Livre de cours ?

- Les diapos de chaque cours sont (seront) sur le site web :  
[http://artis.imag.fr/~Nicolas.Holzschuch/cours/isa\\_maj2.html](http://artis.imag.fr/~Nicolas.Holzschuch/cours/isa_maj2.html)
- Si vous voulez en savoir plus :
  - Foley, vanDam, Feiner et Hughes
    - *Computer Graphics, Principles and Practice*
      - Gros, complet, cher
    - Ou bien : *Introduction to Computer Graphics*
      - Moins gros, moins complet, moins cher
      - Traduit en français : *Introduction à l'infographie*
  - E. Haines & T. Möller, *Real-Time Rendering*

6

---

---

---

---

---

---

---

---

## Plan du cours (2)

- Modélisation
  - Modèles paramétriques
  - Modèles hiérarchiques
  - Déformations du modèle
- Animation
  - Cinématique inverse
  - Résolution des équations différentielles
  - Particules, masse-ressort,...
- Rendu (temps-réel)
  - Modèles complexes de matériaux
  - Ombres en temps-réel
  - Affichage temps-réel : niveaux de détail, etc.

7

---

---

---

---

---

---

---

---

## Images de Synthèse

- Processus à plusieurs étapes
  - Modélisation (+ artiste)
  - Animation
    - Keyframe, cinématique, contraintes, dynamique
    - Aspects réaliste
    - Contraintes de l'histoire
  - Rendu
    - Affichage
    - Aspect réaliste
    - Contraintes de temps

8

---

---

---

---

---

---

---

---

## Films d'animation

- Story-board
- Modèle géométrique
  - Dessin, sculpture, scanner 3D
- Animation :
  - Mouvements à grande échelle
  - Mouvements précis
- Rendu :
  - Rendu rapide pour vérification (tps réel)
  - Rendu complet :
    - 90 mn par image, 25 img/sec, 1h30 de film = 202500 h de calcul

9

---

---

---

---

---

---

---

---

## Du concret

- DVD

10

---

---

---

---

---

---

---

---

## Rendu (suite)

- Rendu temps-réel de plus en plus sophistiqué :



10 fps en 2001  
(NVIDIA demo,  
Siggraph 2001.)

11

---

---

---

---

---

---

---

---

## Plan

- Plan du cours
  - Questions techniques
- Vue d'ensemble de la synthèse
- Coordonnées homogènes
- Modélisation paramétrique

12

---

---

---

---

---

---

---

---

## Transformations géométriques

- Représentation vectorielle des points
  - Points attachés aux primitives graphiques
  - Sommets, centres, données volumiques...
- Transformations sur ces données
  - Translation, rotation, changement d'échelle...
  - Projections :
    - Perspective, parallèle...
  - Notation unifiée ?

13

---

---

---

---

---

---

---

---

## En 2 dimensions

- On commence en 2D
  - Plus facile à représenter
- Chaque point est transformé:
  - $x' = f(x,y)$
  - $y' = g(x,y)$
- Comment représenter la transformation ?

14

---

---

---

---

---

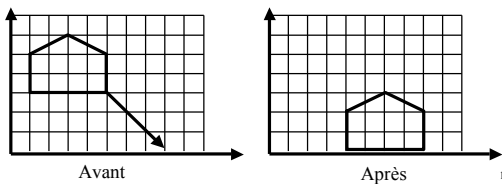
---

---

---

## Translations

- Modification simple :
  - $x' = x + t_x$
  - $y' = y + t_y$



---

---

---

---

---

---

---

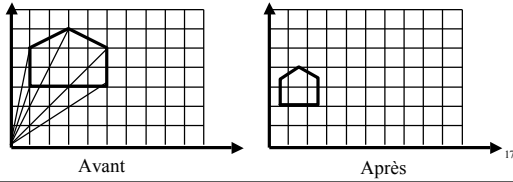
---

## Changement d'échelle

- Les coordonnées sont multipliées par le facteur de changement d'échelle :

- $x' = s_x \cdot x$

- $y' = s_y \cdot y$




---

---

---

---

---

---

---

---

## Notation matricielle

- C'est une multiplication matricielle :

$$P' = SP$$

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

18

---

---

---

---

---

---

---

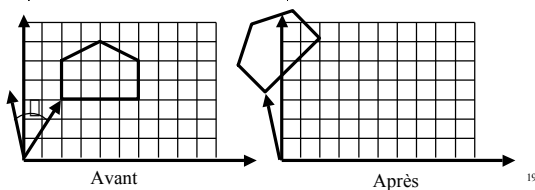
---

## Rotation

- Rotation en 2D :

- $x' = \cos \theta \cdot x - \sin \theta \cdot y$

- $y' = \sin \theta \cdot x + \cos \theta \cdot y$



19

---

---

---

---

---

---

---

---

## Notation matricielle

- Rotation = multiplication matricielle :

$$P' = RP$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos & \sin \\ \sin & \cos \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

20

---

---

---

---

---

---

---

---

## Unification

- Notation simple, concise
- Mais pas vraiment unifiée
  - Addition ou bien multiplication
  - Comment faire pour concaténer plusieurs transformations ?
- On veut une notation unique
  - Qui permette de noter aussi les combinaisons de transformations
  - Comment faire ?

21

---

---

---

---

---

---

---

---

## Coordonnées homogènes

- Outil géométrique très puissant :
  - Utilisé partout en Infographie (Vision, Synthèse)
  - cf. aussi géométrie projective
- On ajoute une troisième coordonnée,  $w$
- Un point 2D devient un vecteur à 3 coordonnées :

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

22

---

---

---

---

---

---

---

---

## Coordonnées homogènes

- Deux points sont égaux si et seulement si :
  - $x'/w' = x/w$  et  $y'/w' = y/w$
- $w=0$ : points « à l'infini »
  - Très utile pour les projections, et pour certaines splines

23

---

---

---

---

---

---

---

---

## Et en 3 dimensions ?

- C'est pareil
- On introduit une quatrième coordonnée,  $w$ 
  - Deux vecteurs sont égaux si :  $x/w = x'/w'$ ,  $y/w = y'/w'$  et  $z/w = z'/w'$
- Toutes les transformations sont des matrices 4x4

24

---

---

---

---

---

---

---

---

## Translations en c. homogènes

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 0 & t_x \\ 1 & t_y \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\begin{bmatrix} x/w \\ y/w \\ w/w \end{bmatrix} = \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix} = \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix} + \begin{bmatrix} t_x/w \\ t_y/w \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} x + wt_x \\ y + wt_y \\ w \end{bmatrix}$$

25

---

---

---

---

---

---

---

---



## Changement d'échelle

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

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ w \end{pmatrix}$$

$$\begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \end{pmatrix} = \begin{pmatrix} s_x \frac{x}{w} \\ s_y \frac{y}{w} \end{pmatrix}$$

26

---

---

---

---

---

---

---

---

## Rotation

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

$$\begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \end{pmatrix} = \begin{pmatrix} \cos \theta \frac{x}{w} + \sin \theta \frac{y}{w} \\ -\sin \theta \frac{x}{w} + \cos \theta \frac{y}{w} \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} \cos \theta x + \sin \theta y \\ -\sin \theta x + \cos \theta y \\ w \end{pmatrix}$$

27

---

---

---

---

---

---

---

---

## Composition des transformations

- Il suffit de multiplier les matrices :
  - composition d'une rotation et d'une translation:  
 $\mathbf{M} = \mathbf{RT}$
- Toutes les transformations 2D peuvent être exprimées comme des matrices en coord. homogènes
  - Notation très générale

28

---

---

---

---

---

---

---

---

## Rotation autour d'un point Q

• Rotation autour d'un point Q:

- Translater Q à l'origine ( $T_Q$ ),
- Rotation autour de l'origine ( $R_Q$ )
- Translater en retour vers Q ( $-T_Q$ ).

$$\longrightarrow P' = (-T_Q)R_Q T_Q P$$

29

---

---

---

---

---

---

---

---

## Translations en 3D

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

$$\begin{cases} x' = x + wt_x \\ y' = y + wt_y \\ z' = z + wt_z \\ w' = w \end{cases}$$

30

---

---

---

---

---

---

---

---

## Changement d'échelle en 3D

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

$$\begin{cases} x' = s_x x \\ y' = s_y y \\ z' = s_z z \\ w' = w \end{cases}$$

31

---

---

---

---

---

---

---

---

## Rotations en 3D

- Rotation : un axe et un angle
- La matrice dépend de l'axe et de l'angle
- Expression directe possible, en partant de l'axe et de l'angle, et quelques produits vectoriels
  - Passage par les quaternions
- Fait par la librairie graphique :
  - `glRotatef(angle, x, y, z)`

32

---

---

---

---

---

---

---

---

## Toutes les transformations 3D

- Toute transformation 3D s'exprime comme combinaison de translations, rotations, changement d'échelle
  - Et donc comme une matrice en coordonnées homogènes
- Fournies par la librairie graphique :
  - `glTranslatef(x, y, z);`
  - `glRotatef(angle, x, y, z);`
  - `glScalef(x, y, z);`

36

---

---

---

---

---

---

---

---

## Transformations 3D (suite)

- On peut faire ses transformations soi-même :
  - `glLoadIdentity();`
  - `glLoadMatrixf(pm);`
  - `glMultMatrixf(pm);`
- Pile de transformations :
  - `glPushMatrix();`
  - `glPopMatrix();`

37

---

---

---

---

---

---

---

---

## Exemple

```

drawHighLevelObject(parameters) {
  glPushMatrix()
  glRotate(...)
  glTranslate(...)
  glScale(...)
  drawSimpleShape()
  glPopMatrix()
}
drawModel() {
  glPushMatrix()
  drawHighLevelObject1(...)
  glTranslate(...)
  drawHighLevelObject2(...)
  [etc...]
  glPopMatrix()
}

```

38

---

---

---

---

---

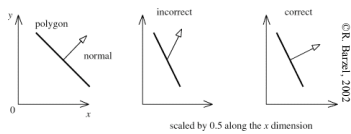
---

---

---

## Transformation des normales

- Vecteur normal (à la surface)
- Pas vraiment un vecteur
  - Définit une relation sur les vecteurs
  - Une forme linéaire, un co-vecteur



- Transformation en utilisant la transposée de l'inverse de  $M$

39

---

---

---

---

---

---

---

---

## Supplément : Projection perspective

- Projection sur le plan  $z=0$ , avec le centre de projection placé à  $z=-d$ :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

40

---

---

---

---

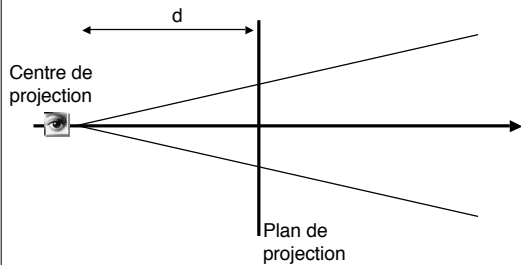
---

---

---

---

## Supplément : perspective (suite)



41

---

---

---

---

---

---

---

---

## Supplément : perspective (suite)

- Coord. homogènes essentielles pour perspective
- La rétrécissement des objets utilise  $w$

$$w \square = \frac{z}{d} + w$$
$$\frac{x \square}{w \square} = \frac{x}{\frac{z}{d} + w}$$

- Impossible sans coordonnées homogènes

42

---

---

---

---

---

---

---

---

## Perspective : en pratique

- Fait par la librairie graphique :
  - `gluLookAt(Eyex, Eyey, Eyez, Centerx, Centery, Centerz, upx, upy, upz);`
  - `gluPerspective( fovy, aspect, zNear, zFar );`

43

---

---

---

---

---

---

---

---

## Modélisation paramétrique

- Dit aussi *modélisation procédurale*
  - Modèle fait par un programme
  - Fondamental en infographie
- Paramètres de position
- Paramètres de forme
  - Forme des parties de l'objet
  - Position relative des parties de l'objet

44

---

---

---

---

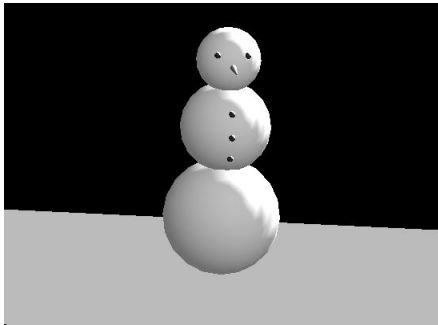
---

---

---

---

## Exemple : bonhomme de neige



45

---

---

---

---

---

---

---

---

## Exemple : bonhomme de neige

- Trois sphères empilées
- Paramètres :
  - Écrasement
  - Inclinaison
  - Pour chaque sphère
- Sujet du TD 1
  - Modèle 2D, représentation 3D

46

---

---

---

---

---

---

---

---

## Modélisation paramétrique

- Routine draw()
- OpenGL effectue les actions de base
  - Élimination des parties cachées,
  - Modèle simple d'éclairage
- OpenGL conserve l'état courant :
  - Matériau, taille des lignes, polygones remplis
  - Transformation courante
    - Plus pile de transformations

47

---

---

---

---

---

---

---

---

## Modèle paramétrique

- Animation faite sur les paramètres
- Animation :
  - Key-frame interpolation
  - *Motion capture*
  - Entrée par l'utilisateur
  - Cinématique
  - Cinématique inverse
  - Dynamique
- Choix en fonction des contraintes
  - Scenario, réalisme...
  - Mélange de méthodes

48

---

---

---

---

---

---

---

---

## Animation : key-frame interpolation

- Animateur fournit données entrée
  - Position, vitesse... à temps  $t_i$
- Interpolation entre points de contrôle
- Courbes de Bézier 2D
- Re-paramétrisation

49

---

---

---

---

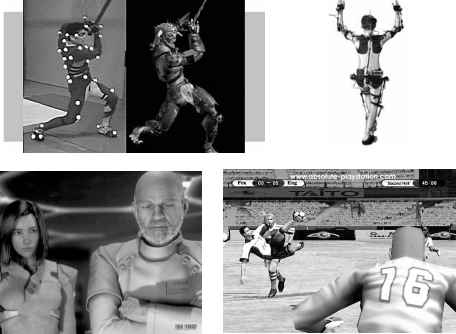
---

---

---

---

## Animation : *Motion Capture*



50

---

---

---

---

---

---

---

---

## Animation : donné par l'utilisateur

- Action directe de l'utilisateur sur les paramètres
- Fourni par la souris
  - $(x_0, y_0)$  et  $(x_p, y_p)$
- Relation entre action de la main et du modèle
  - Perception logique
- Difficile d'agir sur modèle complexe
  - Quel partie de l'objet ?

51

---

---

---

---

---

---

---

---

## Animation : cinématique

- Vitesse donnée en entrée
  - Programme calcule la position
- Utile pour des objets simples, trajectoires simples
- Contrôle complet de l'objet
  - Pratique pour suivre le *scenario*
- ...mais *besoin* d'un contrôle complet de l'objet
  - Difficile pour l'animateur

52

---

---

---

---

---

---

---

---



## Animation : cinématique inverse

- Objets complexes
  - Bras articulé
- Animation d'une partie de l'objet
- Calcul des positions du reste de l'objet
- Simple pour animateur/joueur
- Problème complexe
  - Non-linéaire, pas d'unicité, pas de continuité...

53

---

---

---

---

---

---

---

---

## Animation : dynamique

- Lois de la dynamique, appliquées au modèle
- Trajectoires réalistes
  - Si modèle réaliste
- Complexité pour imposer résultat
- Utile pour particules, objets secondaires...

54

---

---

---

---

---

---

---

---

## Contenu du cours

- Modélisation, animation, rendu
- Coordonnées homogènes
  - Transformations 3D
  - Perspective
- Modèle paramétrique
- Techniques d'animation

55

---

---

---

---

---

---

---

---