

## Animation de personnage basée squelette

Alexandre Meyer  
M2pro

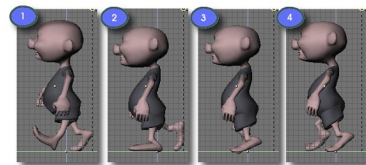


1

## Idée 1 : Animation personnage 3D

- 1ère manière d'animer un personnage
  - Série de maillage correspondant à des positions clés de l'animation

Affiche : mesh1, mesh2, mesh3, mesh4, mesh1, mesh2, etc.



M2R 2

## Idée 1 : Animation personnage 3D

- 1ère manière d'animer un personnage
  - Positions clés de l'animation
  - Interpolation de maillage (BlendShape)
  - Beaucoup utilisé pour les visages
- Inconvénients : création des données et stockage

3% mesh1  
+ 40% mesh2  
+ 17% mesh3  
+ ...

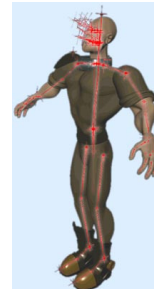
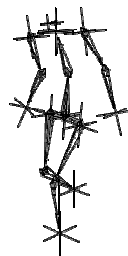


M2R 3

## Idée 2 : Animation personnage 3D

2<sup>e</sup> manière d'animer un personnage

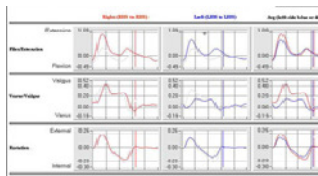
- Personnage 3D = 1 squelette + 1 maillage(avec texture)



M2R 4

## Idée 2 : Animation personnage 3D

Interpolation des positions clefs du squelette  
+ Déformation du maillage (skinning)



M2R 5

## Idée 2 : Vocabulaire

- Manipulation directe (**cinématique directe**)
  - Angles → squelette en 3D
- **Skinning** pour déformer le maillage
- **Cinématique inverse**
  - Position extrémité → calcul angles

M2R 6

## Idée 2 : Animation personnage 3D

Intérêts de l'animation basé squelette

### ■ Squelette

- Structure hiérarchique sur l'objet  
Epaule tourne → tout le bras tourne
- Plus facile de capturer le squelette d'un acteur que sa surface → *Motion Capture (MoCap)*

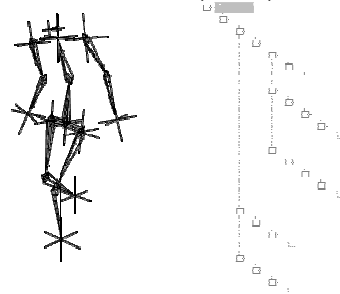
### ■ Déformation du maillage

- Peu coûteux en stockage (1 maillage+qq infos)
- Calcul de déformation suffisamment simple pour être temps réel (et possible sur GPU)

M2R 7

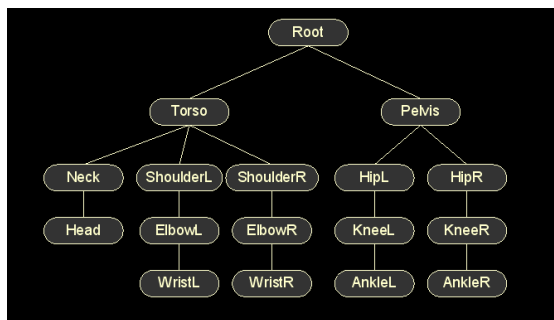
## Squelette

- Pas de géométrie, que des « os »
- Notion de hiérarchie (arbre)



M2R 8

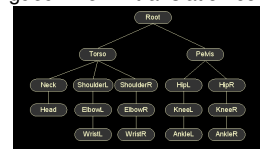
## Squelette = arbre



M2R 9

## Squelette = arbre

- 1 nœud de l'arbre = 1 **articulation, os (joint, bone)**
  - une matrice relative au père
  - Souvent Matrice = rotation + translation
    - Rotation = orientation de l'os
    - Translation → position au repos
- La matrice est contrainte selon le type de joint
  - Ex. coude simple = 1-DOF (degree of freedom)
  - Os de longueur fixe → translation constante durant anim



M2R 10

## Intérêts des arbres

- Structure hiérarchique sur l'objet
  - L'épaule tourne → tout le bras tourne
- Similaire à l'idée de graphe de scène
- Boîtes englobantes :
  - Construites hiérarchiquement
  - Collision, contact
  - Affichage/LOD
- Édition interactive du modèle
  - Animer un squelette est plus simple que d'animer un maillage

M2R 11

## Affichage du squelette

Affichage du squelette = parcours de l'arbre

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

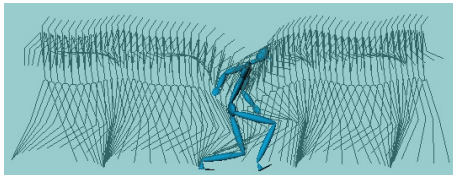
M2R 12

## Animation par positions clés du squelette

- Animation par position clés (key framing)
  - Position initiale (au repos) = un squelette à J joints
  - Une pose = J rotations (avec 1 rotation=1 angle, 2 angles ou 3 angles)
  - Une animation = n poses

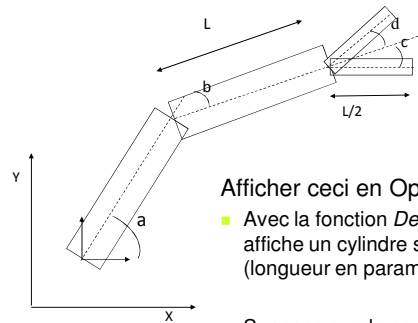


Pose au repos



M2R 13

## Un exemple : robot



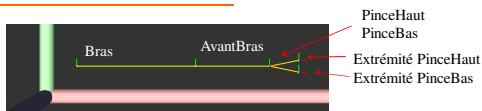
Afficher ceci en OpenGL

- Avec la fonction *DessineCylindre* qui affiche un cylindre sur l'axe des X (longueur en paramètre)
- Suppose que le squelette au repos est  $a=b=c=d=0$

M2R 14

## Description plus générale : robot

**Position au repos**



- **Bras**

Origine du bras par rapport au monde=(20,20,0)

- **AvantBras**

Origine de l'AvantBras par rapport au Bras=(80,0,0)

- **PinceHaut**

Origine de PinceHaut=(50,0,0) par rapport à AvantBras

Extrémité de PinceHaut par rapport origine de PinceHaut=(20,2,0)

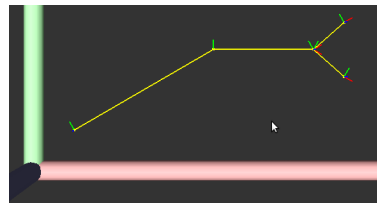
- **PinceBas**

Origine de PinceBas=(50,0,0) par rapport à AvantBras

Extrémité de PinceBas par rapport origine de PinceBas=(20,-2,0)

M2R 15

## Description plus générale : robot



■ Animation : 4 paramètres

- Angle Bras (par rapport au repos)
- Angle AvantBras
- Angle PinceHaut
- Angle PinceBas

M2R 16

## Description plus générale : BVH=format de fichier

**ROOT Bras**

OFFSET 20.00 20.00 0.00

CHANNELS 1 Zrotation

**JOINT AvantBras**

OFFSET 80.00 0 0.00

CHANNELS 1 Zrotation

**JOINT PinceHaut**

OFFSET 50.00 0 0.00

CHANNELS 1 Zrotation

End Site OFFSET

20.00 4 0.00

**JOINT PinceBas**

OFFSET 50.00 0 0.00

CHANNELS 1 Zrotation

End Site OFFSET

20.00 -4 0.00

Description du squelette au repos

M2R 17

```

HIERARCHY
ROOT Bras
{
  OFFSET 20.00 20.00 0.00
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT AvantBras
  {
    OFFSET 80.00 0 0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT PinceHaut
    {
      OFFSET 50.00 0 0.00
      CHANNELS 3 Zrotation Xrotation Yrotation
      End Site
      {
        OFFSET 20.00 4 0.00
      }
    }
    JOINT PinceBas
    {
      OFFSET 50.00 0 0.00
      CHANNELS 3 Zrotation Xrotation Yrotation
      End Site
      {
        OFFSET 20.00 -4 0.00
      }
    }
  }
}
MOTION
Frames: 4
Frame Time: 0.33333
0 0 0 10 0 0 -10 0 0 10 0 0 -10 0 0
0 0 0 20 0 0 -20 0 0 20 0 0 -20 0 0
0 0 0 30 0 0 -30 0 0 30 0 0 -30 0 0
    
```

```

HIERARCHY
ROOT Bras
{
  OFFSET 20.00 20.00 0.00
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT AvantBras
  {
    OFFSET 80.00 0 0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT PinceHaut
    {
      OFFSET 50.00 0 0.00
      CHANNELS 3 Zrotation Xrotation Yrotation
      End Site
      {
        OFFSET 20.00 4 0.00
      }
    }
  }
  JOINT PinceBas
  {
    OFFSET 50.00 0 0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
      End Site
      {
        OFFSET 20.00 -4 0.00
      }
    }
  }
}

MOTION
Frames: 4
Frame Time: 0.33333
0 0 0 1000 -10 0 0 -10 0 0
0 0 0 20 0 0 0 0 0 -20 0 0
0 0 0 30 0 0 0 0 0 -30 0 0

```

+ autre exemple de .bvh

## BVH matrice

```

Joint PinceHaut {
  OFFSET 50.00 10.00 2.00
  CHANNELS 1 Zrotation
}

```

Signifie en GL

```

glTranslatef( 50, 10, 2)
glRotatef( 0, 0, 1, a); // a=angle_des_donnees_MOTION

```

Ou

$$\begin{pmatrix} \cos(a) & -\sin(a) & 0 & 50 \\ \sin(a) & \cos(a) & 0 & 10 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \text{Matrice}_{Bone_{i-1} \leftarrow Bone_i} = \text{Matrice}_{pere \leftarrow local}$$

M2R 20

## Affichage récursive du squelette

Affichage du squelette issue du BVH précédent

```

void draw(BVHJoint joint) {
  glPushMatrix();
  Draw_Bone(); // draw line from (0,0,0) to (OffsetX,OffsetY,OffsetZ);
  glTranslate(OffsetX, OffsetY, OffsetZ);
  glRotate( 0,0,1, a);
  for (i=0; i<joint.numChildren; i++)
    draw( joint.children[i]);
  glPopMatrix();
}

```

M2R 21

## Affichage non récursif

Utile pour le skinning ou la comparaison de pose dans le graphe d'animation

```

Struct CAJoint {
  int m_fatherId; // Le numéro du père dans le tableau de CAJoint de CASkeleton
  Mat4f m_local2world; // La matrice passant du repère de l'articulation vers le monde
  Mat4f m_world2local; // Si besoin : La matrice passant du repère du monde vers l'articulation
};

class CASkeleton {
public:
  // Créer un squelette ayant la même structure que définit dans le BVH
  CASkeleton(const BVH&);
  // Positionne ce squelette dans la position n du BVH
  void setPose(const BVH& bv, const int frameNumber);
  // Calcule la distance entre deux postures. Precond: les deux squelettes doivent être identique
  float distance(const CASkeleton& skel) const;
  // Affiche en OpenGL le squelette, les liens entre les articulations sont donnés par le champ m_fatherId
  void drawGL() const; private:
  // L'ensemble des articulations. Remarque : hiérarchie (arbre) est remplacé l'information "fatherID"
  vector<CAJoint> m_joint;
};

```

M2R 22

## Affichage non récursif

### Pour obtenir la matrice

- Mat4f m\_local2world; // monde<-local
- On fait
  - M\_monde<-local = M\_monde<-pere X M\_pere<-local
  - Pour un joint i
    - Mat4f local2father( ... ); // On construit la matrice local2father avec le transparent 20
    - int pere\_i = m\_joints[i].fatherId;
    - m\_joints[i].m\_local2world = m\_joints[pere\_i].m\_local2world \* local2father

M2R 23

## Interpolation

### Interpolation entre 2 poses d'animations

- ne pas interpoler les positions 3D (sinon longueur des membres non garantie)
- Interpolation des angles, Possible mais attention
  - angle1 = -10
  - angle2 = 10 ==> angle\_milieu = (-10+10)/2 = 0
- Mais
  - angle1=angle1' = 360-10 = 350 (identique à l'angle 1)
  - angle2 = 10 ==> angle\_milieu = (350+10)/2 = 180 != 0 (!!!!!)
- Interpolation des rotations en passant par des quaternions
  - Le mieux

M2R 24

## Matrice VS Quaternion/Translation

- Dans notre cas, les matrices ne comportent qu'une Rotation et une Translation (pas de scale ou autre)

Mat4f M\_A;

$$M_A = \begin{pmatrix} R_A & T_A \\ 0 & 1 \end{pmatrix} \text{ avec } R_A = \begin{pmatrix} - & - & - \\ - & - & - \\ - & - & - \end{pmatrix} = \text{rotation}$$

et

$$T_A = \begin{pmatrix} Tx \\ Ty \\ Tz \end{pmatrix} = \text{vecteur\_translation}$$

- On peut représenter ceci avec un quaternion + vecteur
  - Quaternion Q\_A; // cf. cours de R. Chaine et
  - Vec3f T\_A; // [http://fr.wikipedia.org/wiki/Quaternions\\_et\\_rotation\\_dans\\_l'espace](http://fr.wikipedia.org/wiki/Quaternions_et_rotation_dans_l'espace)

M2R 25

## Matrice VS Quaternion/Translation

- Combiner des transformations

- Mat4f M\_a;
- Mat4f M\_b;
- Mat4f M\_ab = M\_a \* M\_b;

$$M_a = \begin{pmatrix} R_a & T_a \\ 0 & 1 \end{pmatrix} \quad M_b = \begin{pmatrix} R_b & T_b \\ 0 & 1 \end{pmatrix}$$

$$M_a \times M_b = \begin{pmatrix} R_a & T_a \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} R_b & T_b \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_a \times R_b & R_a \times T_b + T_a \\ 0 & 1 \end{pmatrix}$$

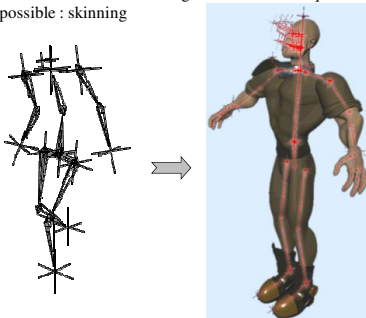
- Transformation (=Quaternion + vecteur) A et B

- Quaternion Q\_a; Vec3f T\_a; // Transfo A
- Quaternion Q\_b; Vec3f T\_b; // Transfo B
- Composition A o B
  - Quaternion Q\_ab = Q\_a \* Q\_b;
  - Vec3f T\_ab = Q\_a \* T\_b + T\_a;

M2R 26

## Introduction "Mesh Skinning"

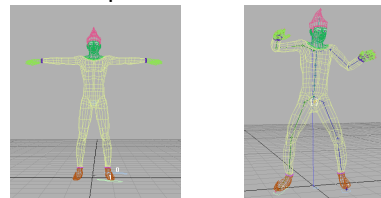
Supposons qu'un artiste nous produit un modèle correspondant au squelette  
Ou inversement il produit un squelette correspondant à un maillage donné  
→ **Problème** : comment animer le maillage en fonction du squelette  
Une réponse possible : skinning



M2R 27

## Introduction "Mesh Skinning"

- Déformer le maillage en fonction des mouvement du squelette
- Vu comme la peau qui se déforme au dessus du squelette



M2R 28

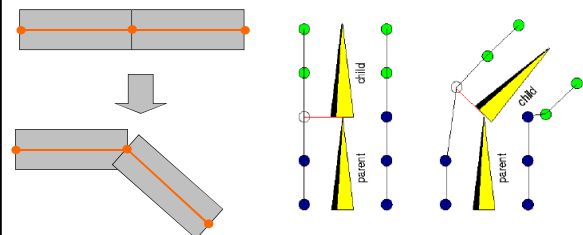
## Skinning

- Maillage 3D + squelette
- Chaque sommet est attaché à un ou plusieurs os
  - Somme des poids = 1
- Pose « au repos »
  - Maillage non déformé
- Pose « animée »
  - Position squelette donnée
  - Trouver position des sommets

M2R 29

## Skinning

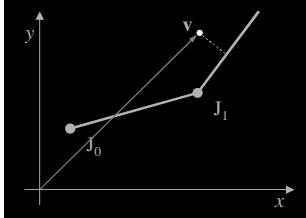
- Sommet(vertex) du maillage associé à un os
- Déplacement de l'os → le maillage suit
  - transformation des sommets



M2R 30

### Comment ça marche (1)

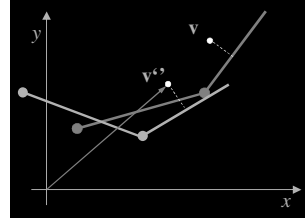
- Sommet  $v$ , attaché à l'os  $J_1$ . Au repos :



© J. Gregory, 2003

### Comment ça marche (2)

- Nouvelle position du squelette
  - Trouver nouvelle position du sommet



© J. Gregory, 2003

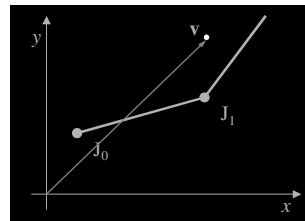
### Comment ça marche (3)

- On transforme le sommet  $v$  :
  - De l'espace du modèle au repos (ou Monde)
  - Dans l'espace lié à l'os au repos (local)  $\rightarrow v'$
- Le sommet est *fixe* par rapport à l'os
  - Donc on peut bouger l'os dans sa position d'animation
- Re-transformation de l'espace local animé vers l'espace du monde
  - Donne nouvelles coordonnées du sommet  $\rightarrow v''$

MGR 33

### Comment ça marche (4)

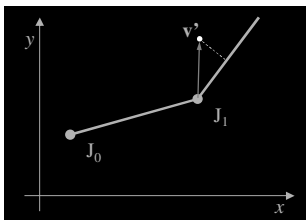
- Passage du sommet  $v$  du repère du modèle (monde) dans l'espace lié à l'os (local) au repos



© J. Gregory, 2003

### Comment ça marche (4)

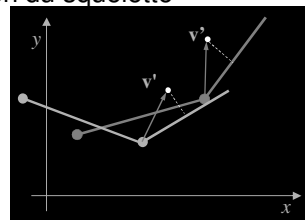
- Sommet, dans l'espace lié à l'os (local) au repos



© J. Gregory, 2003

### Comment ça marche (5)

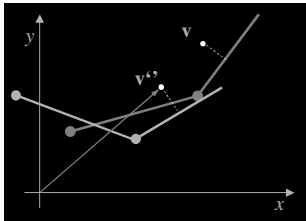
- Animation du squelette
- Sommet toujours dans la même position par rapport à l'os, quelle que soit la position du squelette



© J. Gregory, 2003

## Comment ça marche (6)

- Retour à l'espace du modèle (monde), position du sommet

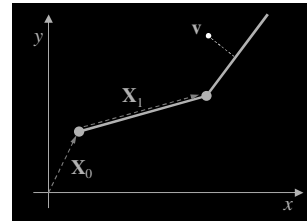


© J. Gregory, 2003

## Avec des maths

### ■ Squelette au repos

$X_i$  = translation = position de la base de l'os  $i$  par rapport à l'os père  $i-1$

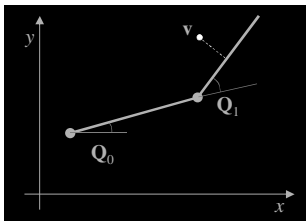


© J. Gregory, 2003

## Avec des maths (2)

### ■ Squelette au repos

$Q_i$  rotation de l'os  $i$  par rapport à l'os père  $i-1$   
Souvent  $Q_i$  est l'identité (cas des fichiers BVH)



© J. Gregory, 2003

## Avec des maths (3)

### ■ Position au repos de l'os $J_i$

- Produit matriciel des translations et rotations depuis la racine :

$$B_j = \prod_{i=0}^j X_i Q_i = \text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{repos}}$$

avec

$$X_i Q_i = \text{Matrice}_{\text{Local}_{i-1} \leftarrow \text{Local}_i}^{\text{repos}}$$

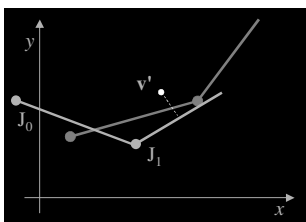
donc

$$v' = B^{-1}_j \cdot v$$

M2R 40

## Avec des maths (4)

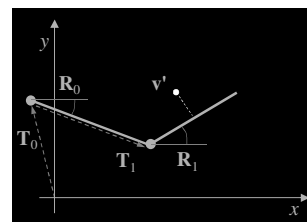
- On déplace le squelette (animation) :



© J. Gregory, 2003

## Avec des maths (5)

- $T_i$  la translation pour l'os  $i$ ,  $R_i$  sa rotation :



© J. Gregory, 2003

## Avec des maths (6)

- Pose *actuelle* décrite par matrice :

$$\mathbf{P}_j = \prod_{i=0}^j \mathbf{T}_i \mathbf{R}_i = \text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{animation}}$$

similaire à la matrice de la pose *au repos* :

$$\mathbf{B}_j = \prod_{i=0}^j \mathbf{X}_i \mathbf{Q}_i = \text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{repos}}$$

M2R 43

## Avec des maths (7)

- Multiplication du sommet  $v$  du maillage par  $\mathbf{B}_j^{-1}$  pour se ramener dans l'espace lié à l'os/local (au repos)
  - Peut être fait une fois pour toutes puisque la position au repos reste la même
- Puis multiplication du résultat par  $\mathbf{P}_j$  pour se ramener dans l'espace du modèle/monde, dans la pose actuelle

M2R 44

## Avec des maths (8)

- Ce qui nous donne, si  $v$  est influencé par l'os  $j$

$$\begin{aligned} \mathbf{v}'' &= \mathbf{P}_j \cdot \mathbf{B}_j^{-1} \cdot \mathbf{v} \\ &= \text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{animation}} \times \\ &\quad \text{Inverse}(\text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{repos}}) \times \mathbf{v} \\ &\text{avec} \\ \mathbf{v}' &= \text{Inverse}(\text{Matrice}_{\text{Monde} \leftarrow \text{Local}_j}^{\text{repos}}) \times \mathbf{v} \end{aligned}$$

M2R 45

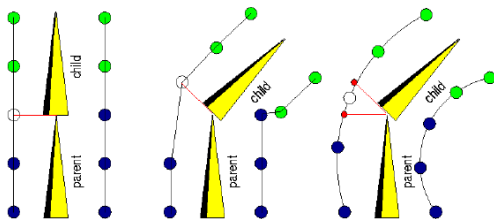
## Avec des maths (9)

- Calculs effectués pour chaque sommet du maillage
  - Chaque sommet du maillage est attaché à au moins un os
- La connectivité du maillage reste identique
  - Dessine les sommets obtenus avec la connectivité du maillage au repos
- Remarque :  
Interpolation possible des positions pour les sommets qui dépendent de plusieurs os

M2R 46

## Poids

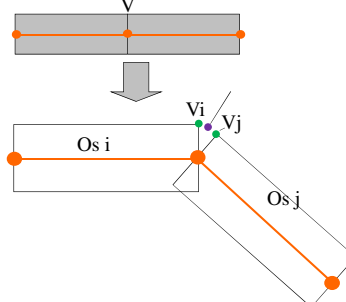
- Sommets(vertex) modifiés par plusieurs os
- Moyenne pondérée des déplacements
- Ajuster les poids



M2R 47

## Smooth Skinning

- Un sommet  $V$  est influencé par plusieurs os
  - Ici l'os  $i$  et l'os  $j$



Sommet  $V$  =  
moyenne pondérée  
des sommets  $V_i$  et  $V_j$

M2R 48



## Skinning : en résumé

$$v' = \sum_i^n w_i M_i v \quad \text{avec} \quad \sum_i w_i = 1$$

où:

- $n$  est le nombre d'os influençant le vertex
- $v$  est la position du sommet
- $w_i$  est le poids associé
- $M_i$  est la matrice de transformation P.B<sup>-1</sup>

M2R 49

## Skinning : en résumé

- Pour les normales:

$$n' = \sum_i^N w_i M_i^{-1^T} n \quad \text{avec} \quad \sum_i w_i = 1$$

Où

$N$  est le nombre de matrice

$n$  est la normale

$w_i$  est le poids associé

$M_i^{-1^T}$  est la transposée inverse de la matrice  $M_i$

M2R 50

## Skinning

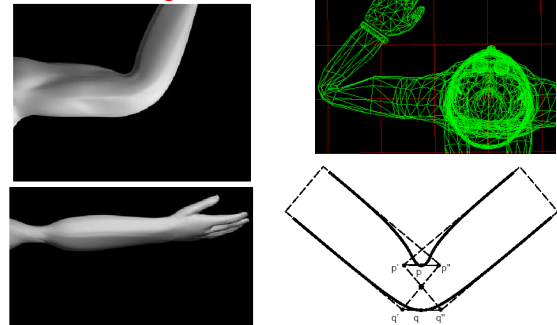
- Problème : étant donné un maillage
  - Comment trouver un squelette approprié
  - Pour chaque point du maillage, comment trouver le ou les os associés? + poids?
- Différentes pistes
  - Artiste à la main
  - Squelette : avec une contraction du maillage
  - Poids
    - Distance à l'os (cf. TP1 Blender)
    - Diffusion de la chaleur [cf. SIGGRAPH2007]

M2R 51

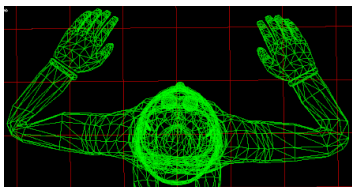
## Skinning: limitations

Le skinning que nous venons de voir est le

**Smooth skinning.**



## Skinning++



Problème avec le smooth skinning classique (coude gauche)

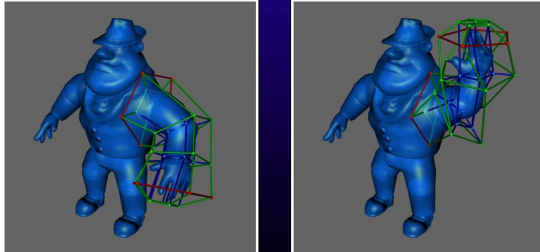
## Skinning++

- Puisque problème avec smooth skinning
- Autres réponses (cf. transparents suivants)
  - Skinning + Free Form Deformation (FFD)
  - Skinning classique avec dual quaternion
  - « Simulation » de la déformation de la peau
    - [Larboulette05]: cherche à conserver une surface de peau constante
    - en utilisant des muscles
  - etc.

M2R 54

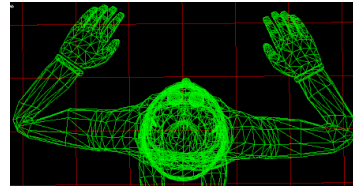
## Skinning+ FFD

- Placer squelette simplifié sur modèle
- Squelette se transforme en armature puis FFD=Free Form Deformation



M2R 55

## Skinning+FFD : résultat



Smooth skinning

Skinning + FFD

M2R 56

## Skinning : dual quaternion

- Quasiment identique au smooth skinning
  - Les rotations et translation sont regroupés dans des quaternions++ = dual quaternions dont l'interpolation se passe mieux



Smooth skinning

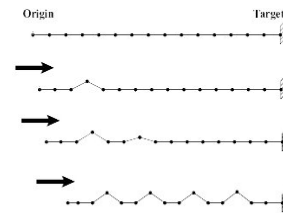
Dual quaternion

VIDEO

M2R 57

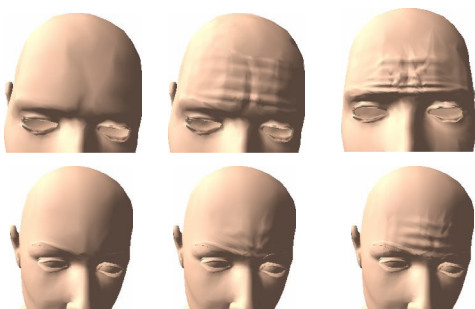
## "Dynamic Wrinkles" [Larboulette05]

- Conservation de l'aire de la surface durant la pliure
- Par un système de masse-ressort peu dense donc peu coûteux à résoudre



M2R 58

## Résultats : "Dynamic Wrinkles"



M2R 59

## Résultats : "Dynamic Wrinkles"

Skinning classique



Dynamic Wrinkles

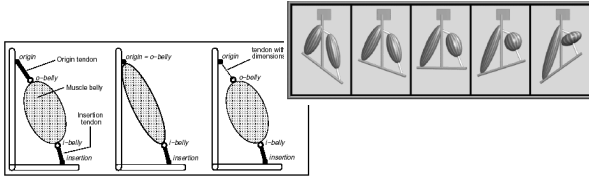


VIDEO

M2R 60

## Modèle Anatomique

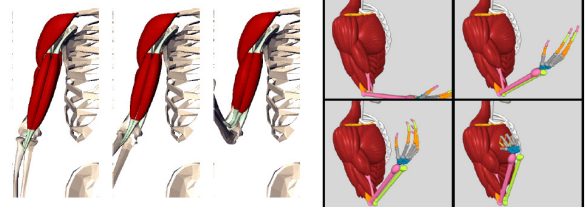
- Des muscles sont attachés aux os (avec des tendons)
- Contraction des muscles avec calcul du volume du muscles



M2R 61

## Modèle Anatomique

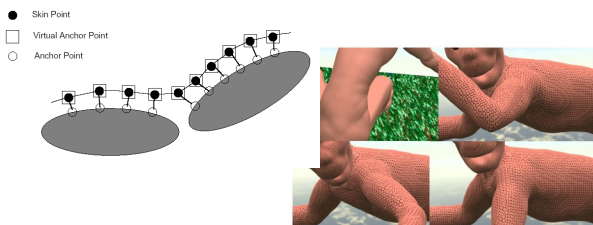
- Muscles complexes construits à partir de plusieurs muscles simples



M2R 62

## Modèle Anatomique

- Peau est attachée aux muscles avec un système masse-ressorts et une simulation physique avec collision os/muscle/peau



## Modèle Anatomique

- Couplé à des paramètres physiques réels  
→ résultats très réaliste
- Couteux en calculs
- Mise en place lourde
- Ne gère que les déformations de la peau
  - (ce n'est pas une animation physique, pas de notion d'équilibre, etc.)
- Voir : <http://www.cgcharacter.com>

M2R 64

## Modèle Anatomique + VIDEO

