

# Déformation de surfaces

SIA Ensimag 3A

Estelle Duveau

# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion

# Plan

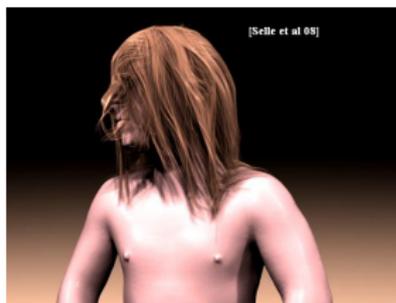
- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion

# Animation 3D pour quoi?

- Crédibilité vs Expressivité



- Réalisme vs Contrôlabilité

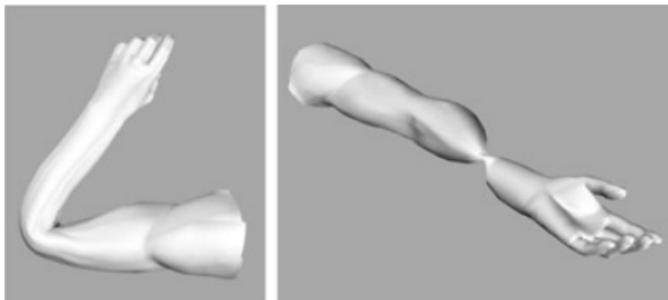


# Animation de surfaces 3D

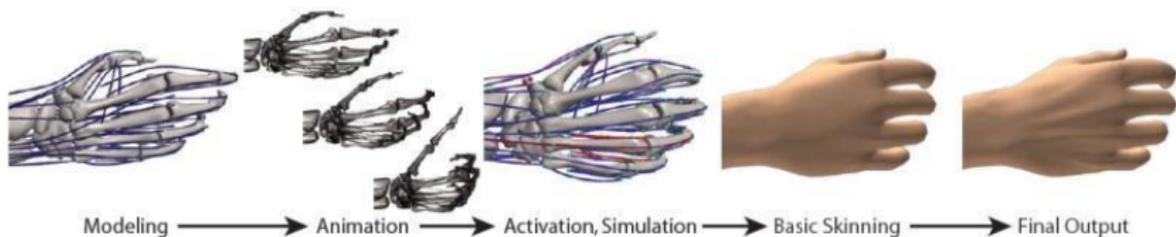
- De quel **niveau de détails** se satisfaire dans l'animation ?
- Souvent, on se satisfait de l'**approximation par une surface 3D** de la réalité :
  - ▶ rapport direct avec la modélisation (forme et texture)
  - ▶ structure légère facile à animer (en temps réel)
- Modèles tous basés sur points de contrôle  
⇒ **Déformer un modèle** ⇔ **Agir sur les points de contrôle**

# Rappels

Besoin d'une solution entre



et



Sueda 2008

# Rappels

Deux questions subsistent :

- Génération de données cinématiques
- Déformation de surfaces rapide et réaliste

# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion

# Principe

- Modification **globale** des formes 3D
- Définir une fonction dans l'espace :

$$P \in \mathbb{R}^3 \rightarrow M(P)$$

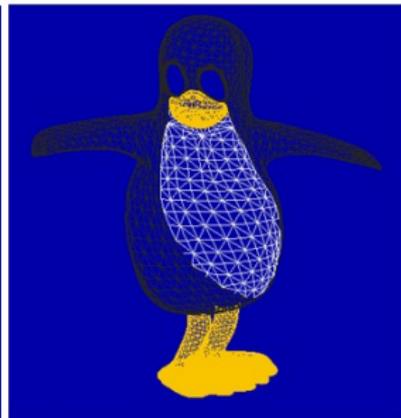
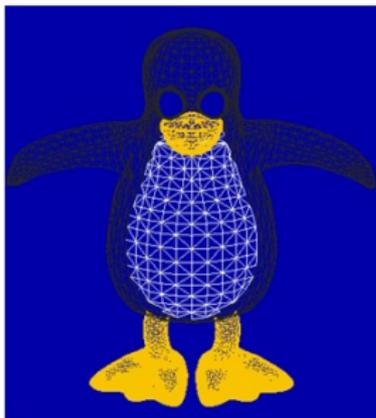
où  $M(P)$  est une matrice de transformation

- Action sur un point  $P$  :
  - ▶ évaluer  $M$  au point  $P$
  - ▶ faire agir  $M$  sur  $P$  :  $P' = M(P)P$

## Exemple : twist (rotation non-uniforme)

$$r(z) = \begin{cases} 0 & z \leq z_0 \\ \frac{z - z_0}{z_1 - z_0} \theta_{\max} & z_0 \leq z \leq z_1 \\ \theta_{\max} & z_1 \leq z_0 \end{cases}$$

$$P' = \begin{bmatrix} \cos(r(p_z)) & -\sin(r(p_z)) & 0 \\ \sin(r(p_z)) & \cos(r(p_z)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

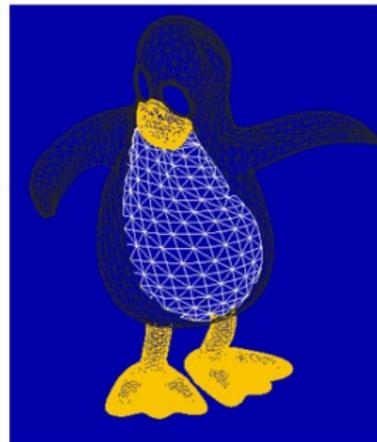
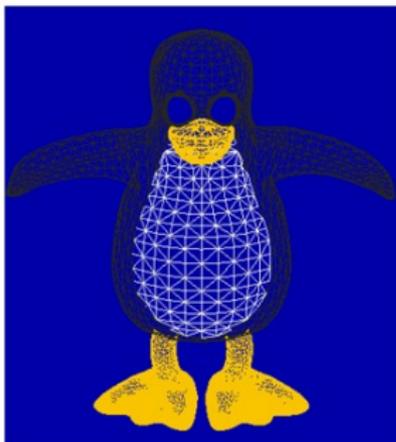


## Exemple : vortex

$$r(z) = \begin{cases} 0 & z \leq z_0 \\ \frac{z - z_0}{z_1 - z_0} \theta_{\max} & z_0 \leq z \leq z_1 \\ \theta_{\max} & z_1 \leq z \end{cases}$$

$$\alpha(P) = r(p_z) e^{-(p_x^2 + p_y^2)}$$

$$P' = \begin{bmatrix} \cos(\alpha(P)) & -\sin(\alpha(P)) & 0 \\ \sin(\alpha(P)) & \cos(\alpha(P)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$



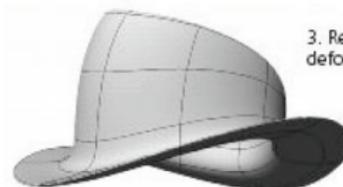
## Exemple : combinaison



1. The original hat



2. Result of a deformation (bend)



3. Result of a second deformation (twist)



4. Muting the Bend yields a twist.

# Bilan

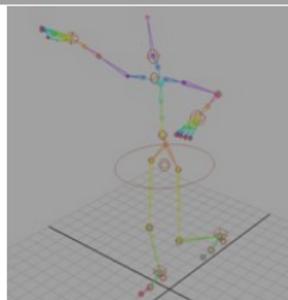
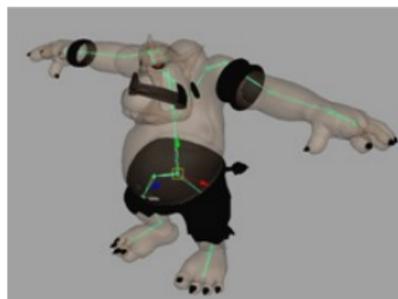
- Avantages :
  - ▶ pratique
  - ▶ simple
- Inconvénients :
  - ▶ pas de contrôle fin des déformations
  - ▶ le modèle peut se recouper

# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire**
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion

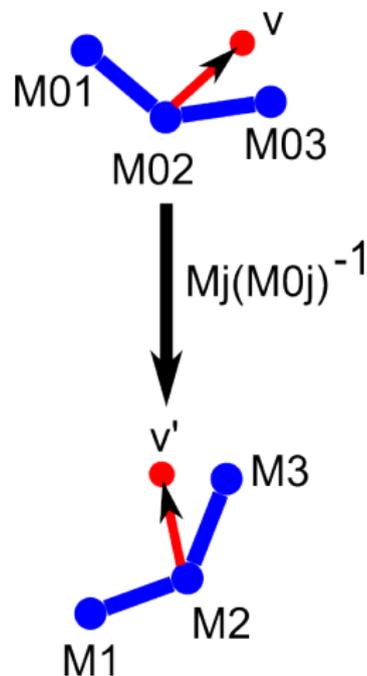
# Skinning

- **Modèle 3D (peau) + squelette d'animation**  
⇒ comment **attacher** la peau au squelette?
- **Pose au repos** : modèle non déformé
- **Pose animée** : position squelette donnée  
⇒ trouver les nouvelles positions des sommets du modèle



# Principe

- On connaît  $v$  dans l'espace du modèle
- On exprime  $v$  dans l'espace lié à l'os
- **Le sommet est fixe par rapport à l'os**
- On fait bouger le squelette
- On exprime  $v$  dans l'espace du modèle  
⇒ nouvelles coordonnées de  $v$
- Position au repos de l'os  $j$  :  $M_{0,j}$
- Pose actuelle :  $M_j$
- $v' = M_2 M_{0,2}^{-1} v$



# Sommet influencé par plusieurs os

- **Interpolation** des positions

- Si  $P$  lié à os 1 et 2 :

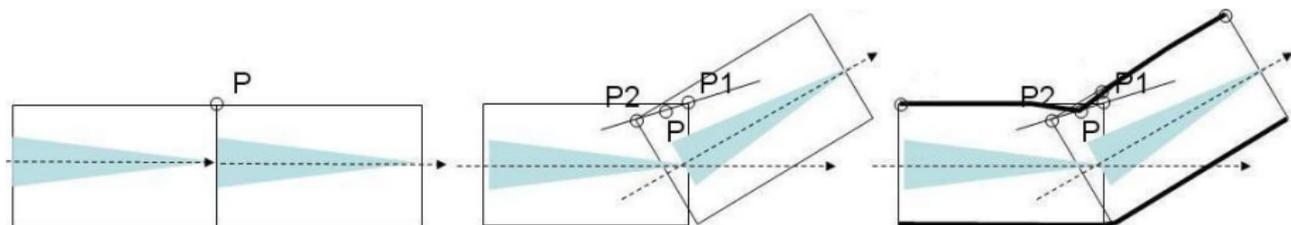
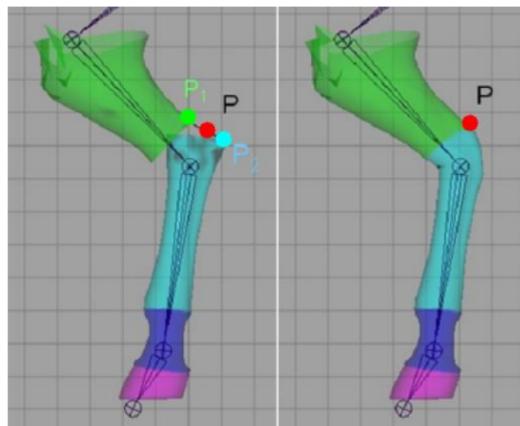
$$P' = wP_1 + (1 - w)P_2$$

- Généralisation :

$$P' = \sum_i w_i M_i M_{0,i}^{-1} P$$

avec  $\sum_i w_i = 1$

⇒ Comment calculer les poids?



# Distance euclidienne

- Inverse de la distance :

$$\alpha_i(P) = \frac{1}{d_i}$$

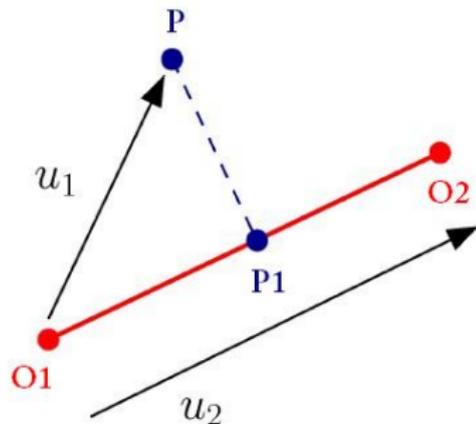
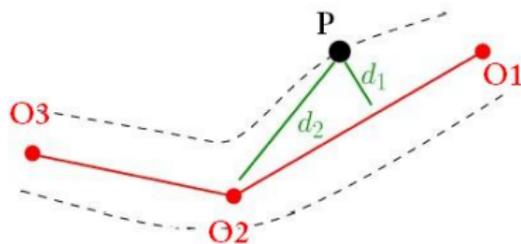
$$w_i(P) = \frac{\alpha_i(P)}{\sum_j \alpha_j(P)}$$

- Quelle distance? Distance cylindrique :  
projection du point sur le segment :

$$p = \frac{u_1 \cdot u_2}{\|u_2\|^2} \text{ tronqué sur } [0, 1]$$

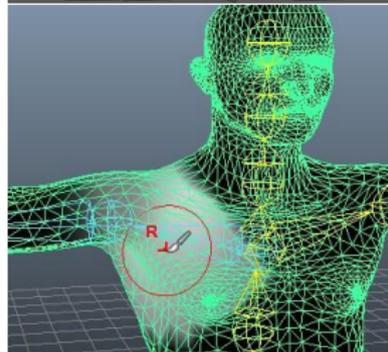
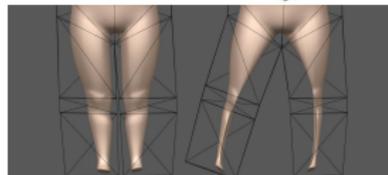
$$P_i = O_i + pu_2$$

$$d_i = \|P - P_i\|$$



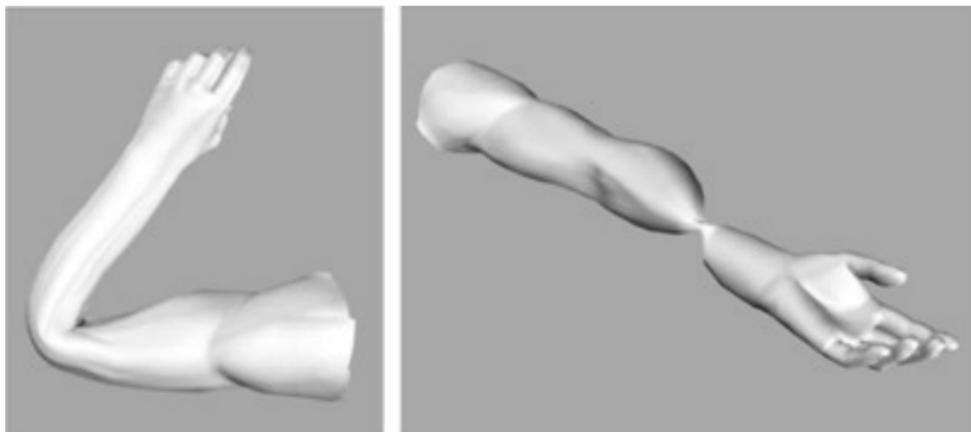
# Autres distances

- Distance curviligne
- Distance géodésique
- Un artiste peint directement sur les maillages...



# Bilan

- Rapide
- Intuitif à construire et animer
- $v' = Qv$  avec  $Q$  transformation **non rigide**  
⇒ artefacts pour angles importants



# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire**
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion

# Pose Space Deformation

- Skeleton Subspace Deformation (SSD) → **Pose Space Deformation (PSD)** [Lewis et al 00]
- SSD : configuration des articulations

$$\Rightarrow P' = \sum_i w_i M_i M_{0,i}^{-1} P$$

- PSD : utiliser des exemples pour corriger les positions obtenues par skinning

# Pose Space Deformation

- Configuration des articulations :

$$P_{skinning} = \sum_i w_i M_i M_{0,i}^{-1} P$$

- Modifications des points sur certaines poses :

$$P_{ex} = P_{skinning} + \delta P_{ex}$$

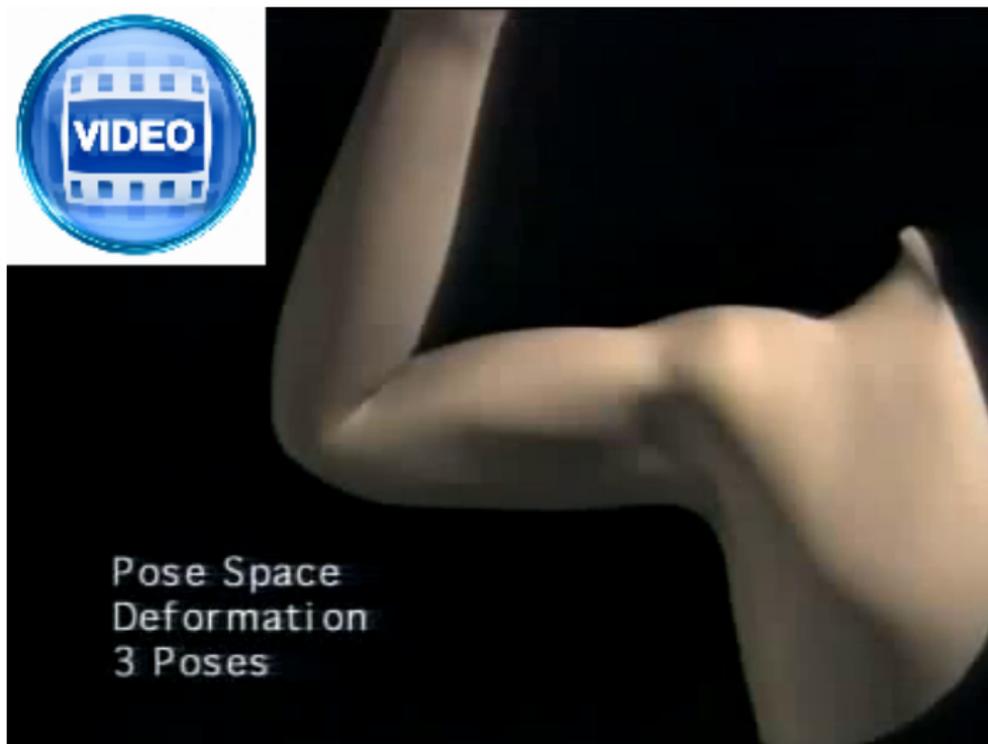
- Nouvelle configuration :

$$\Rightarrow P' = P_{skinning} + \delta P$$

où  $\delta P$  est interpolé à partir des  $\delta P_{ex}$

# Pose Space Deformation

Pose Space Deformations : A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation - J.P. Lewis, M. Cordner, N. Fong



# Utilisation d'exemples

- PSD requiert de garder en mémoire les exemples  $\Rightarrow$  **grande consommation mémoire**
- Apprendre des exemples pour ne plus en avoir besoin à l'exécution [Mohr et Gleicher 03] : Modéliser les déformations en **ajoutant des articulations**
  - ▶ **Twist** (grande rotation) : ajouter des articulations pour mieux interpoler la rotation
  - ▶ **Contraction** de muscles (angle relatif) : ajouter une articulation en scale
  - ▶ ...
- Etudier la séquence exemple pour ajouter **seulement** les articulations et poids nécessaires
  - $\Rightarrow$  appliquer le skinning linéaire sur le nouveau squelette d'animation

## Utilisation d'exemples

Building Efficient, Accurate Character Skins from Examples - A. Mohr, M. Gleicher



Original



Our Method



# Quaternions duaux

- Utilisation d'exemples : correction des artefacts, ne s'attaque pas au problème du skinning linéaire
- $v' = Qv$  avec  $Q = \sum_i w_i M_i M_{0,i}^{-1}$  transformation affine, **non rigide**
- Utiliser les **quaternions duaux** [Kavan et al 07] pour interpoler entre les  $M_i M_{0,i}^{-1}$  de manière rigide

# Quaternions duaux

- **Quaternions duaux :**

$$\hat{q} = q_0 + \epsilon q_{\text{epsilon}} \text{ avec } \epsilon^2 = 0$$

$$\hat{q} = q_0 + \frac{\epsilon}{2}(t_0j + t_1j + t_2k)q_0$$

$$\Rightarrow T = (t_0, t_1, t_2), R = q_0$$

- Calculer les matrices  $Q$  en passant par les quaternions duaux :

$$DLB(w; \hat{q}_1, \dots, \hat{q}_n) = \frac{w_1 \hat{q}_1 + \dots + w_n \hat{q}_n}{\|w_1 \hat{q}_1 + \dots + w_n \hat{q}_n\|}$$

- Cette transformation est **rigide, indépendante des référentiels et retourne la plus petite trajectoire**
- Facilement implémentable sur **GPU**

# Quaternions duaux

Skinning with dual quaternions - L. Kavan, S. Collins, J. Zara, C. O'Sullivan



Matrix Blending  
0 FPS

Dual Quaternions  
198.3 FPS

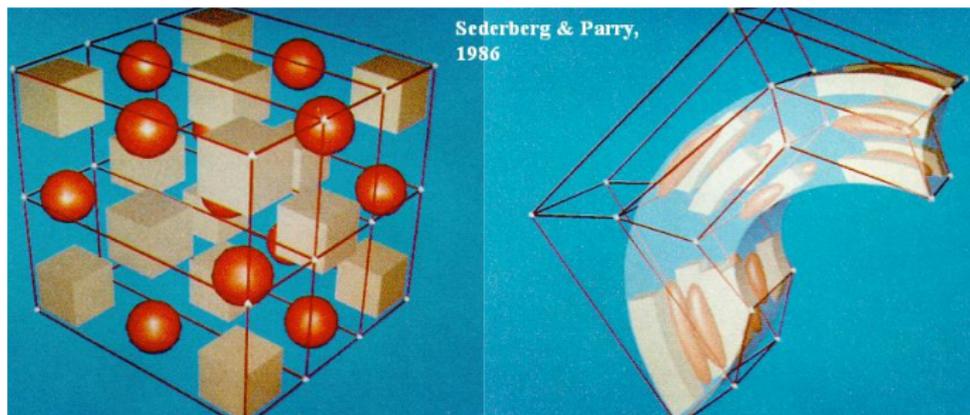


# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage**
- 6 Animation de visages
- 7 Conclusion

# Free-form deformations - Principe

- **Déformer l'espace** autour du modèle
- Maillage de points de contrôle : le modèle est plongé dans du caoutchouc 3D
- Déformer le maillage
- L'espace suit le maillage



# Free-form deformations - Détails

- Parallélépipède de l'espace  $(S,T,U)$
- **Paramétrisation locale** : conversion  $(x,y,z) \rightarrow (s,t,u)$
- Points de contrôle  $P_{ijk}$
- Déplacement des points de contrôle
- Nouvelle position  $(x',y',z')$  en fonction de  $(s,t,u)$

# Free-form deformations - Plongement

- Paramétrisation locale :

$$M = M_0 + sS + tT + uU$$

$$s = \frac{T \times U \cdot (M - M_0)}{T \times U \cdot S}$$

$$t = \frac{S \times U \cdot (M - M_0)}{S \times U \cdot T}$$

$$u = \frac{S \times T \cdot (M - M_0)}{S \times T \cdot U}$$

- Points de contrôle : par ex, régulier :

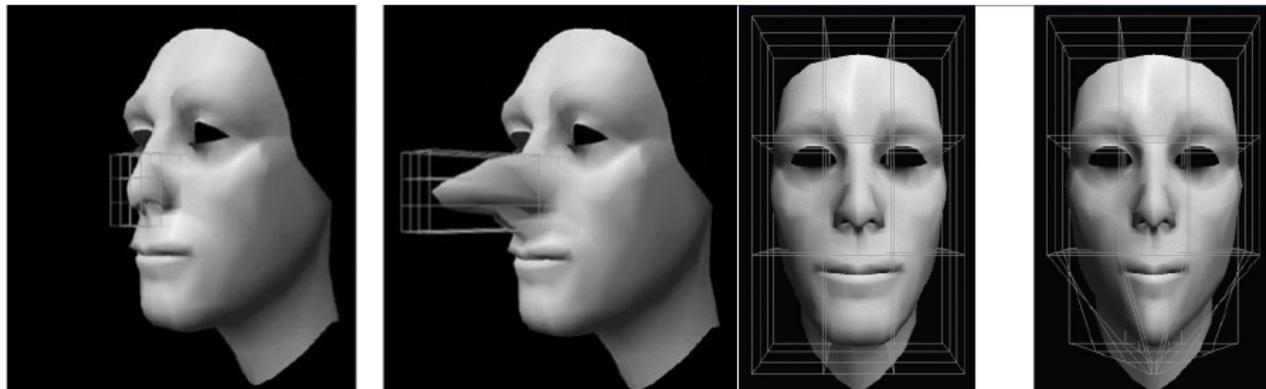
$$P_{ijk} = M_0 + \frac{i}{i_{\max}} S + \frac{j}{j_{\max}} T + \frac{k}{k_{\max}} U$$

# Free-form deformations - Plongement

- Nouvelle position :

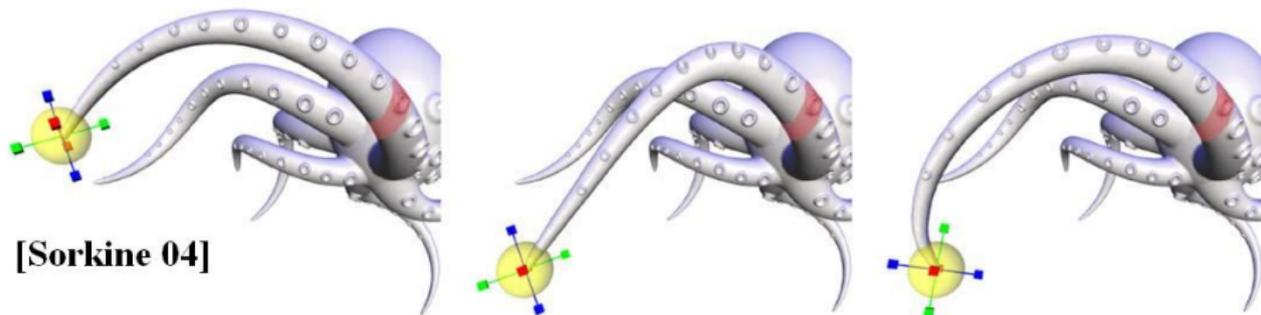
$$M_{FFD} = \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} \sum_{k=0}^{k_{max}} B_i^{i_{max}}(s) B_j^{j_{max}}(t) B_k^{k_{max}}(u) P_{ijk}$$

avec B polynôme de Bernstein



# Laplacian mesh edition - Principe

- Animation de personnages sans squelette
- Ensemble de sommets **déformés localement** tout en préservant les détails
- Basée sur la **géométrie différentielle discrète**



## Laplacian mesh edition - Détails

- Chaque coordonnée de sommet est remplacée par la différence avec ses voisins :  $D = LV$  ( $L$  : poids)
- Ajout des contraintes  $\Leftrightarrow$  ajout de lignes à  $L$  et  $D \Rightarrow L'$  et  $D'$
- Reconstruction de  $V$  par approximation :  $V' = \operatorname{argmin}_V (\|L'V - D'\|)$

# Laplacian mesh edition

VIDEO

Reload

Info +

Export files +

Editing -

ROI -

Edit params

Free ring radius 0.5

Fixed ring radius 0.06

Handle radius 0.03

ROI selection type

Euclidean radius

Geodesic radius

Edit Mode

Render anchors

System data -

Settings +

Store result

Save to IJV

Matrix size: 33486

Geometry sources and visualization +

Rendering modes +

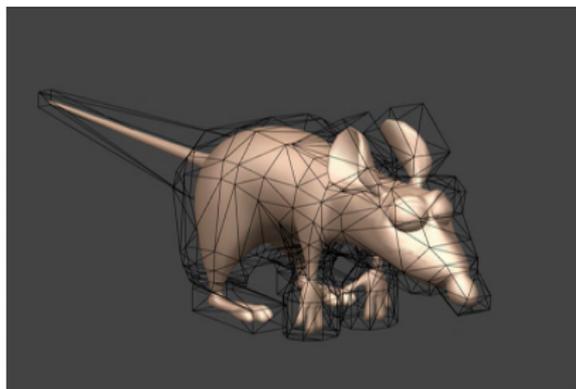
Lights +

Windows +

Blue Light Golden Light White Light Red Light

## Autres systèmes de coordonnées

- Utilisation d'une **cage** (maillage à géométrie libre) pour contrôler l'animation d'un maillage
- **Points de contrôle** de déformation = Sommets de la cage  $p_j$
- $v = \sum_j w_{v,j} p_j$  avec précalcul des  $w_{v,j}$
- Comment calculer les **influences** des sommets de la cage sur le maillage à animer?  
⇒ Quelles **coordonnées** utiliser?



## Mean-Value Coordinates

- Construction d'une **fonction qui interpole** des valeurs définies aux sommets d'un maillage : shading de Gouraud, déformation, ... :

$$f[v] = \frac{\sum_j w_j f[p_j]}{\sum_j w_j}$$

- **Mean-value coordinates** : [Floater 2003]

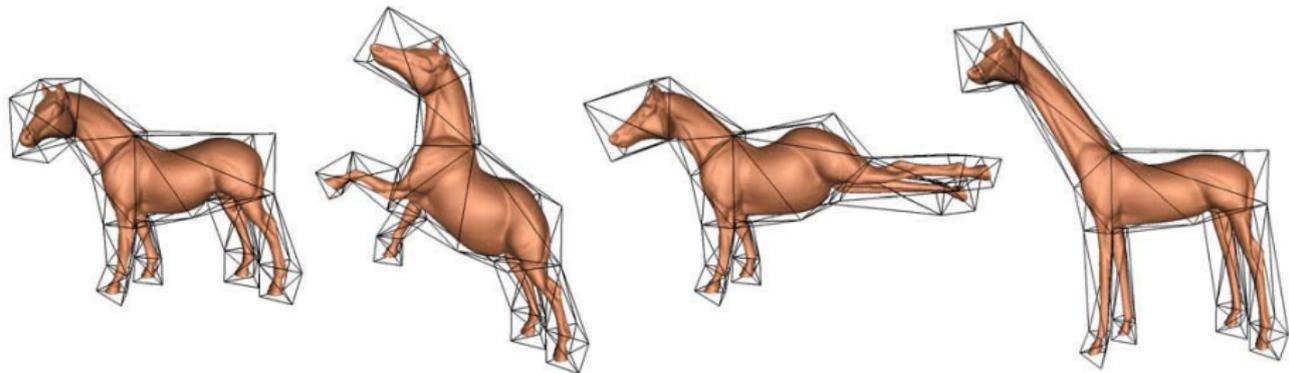
$$w_j = \frac{\tan \frac{\alpha_{j-1}}{2} + \tan \frac{\alpha_j}{2}}{|p_j - v|}$$

$$\alpha_i = \text{angle}(p_j - v, p_{j+1} - v)$$

- **Solution analytique** pour les formes linéaires par morceau
- Cas de la **déformation de maillages** pour des cages triangulaires fermées :
  - ▶  $f[v]$  = position du sommet  $v$  du maillage
  - ▶  $f[p_j]$  = position du sommet  $p_j$  de la cage

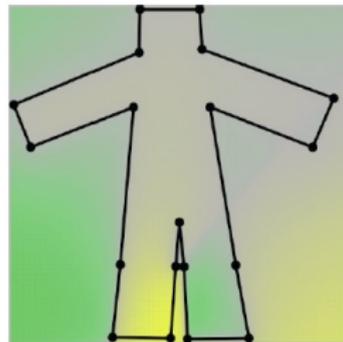
# Mean-Value Coordinates

Mean Value Coordinates for Closed Triangular Meshes - T. Ju, S. Schaefer, J. Warren - 2005

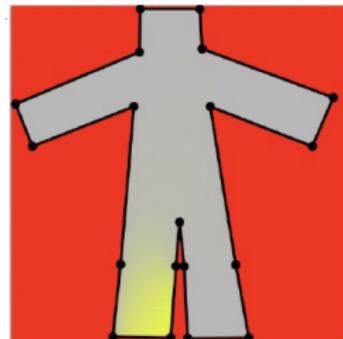


# Coordonnées harmoniques

- Mean-value coordinates : basées sur distance euclidienne, ne respectent pas les propriétés de **localité interne** et **non-négativité interne**
- **Localité interne** : pas d'extrema à l'intérieur de la cage
- Utilisation de l'**équation de Laplace**  
⇒ coordonnées harmoniques
- Non-négativité et localité interne
- Possibilité d'ajouter des **cages internes**
- **Pas de solution analytique**, nécessité d'un solveur numérique



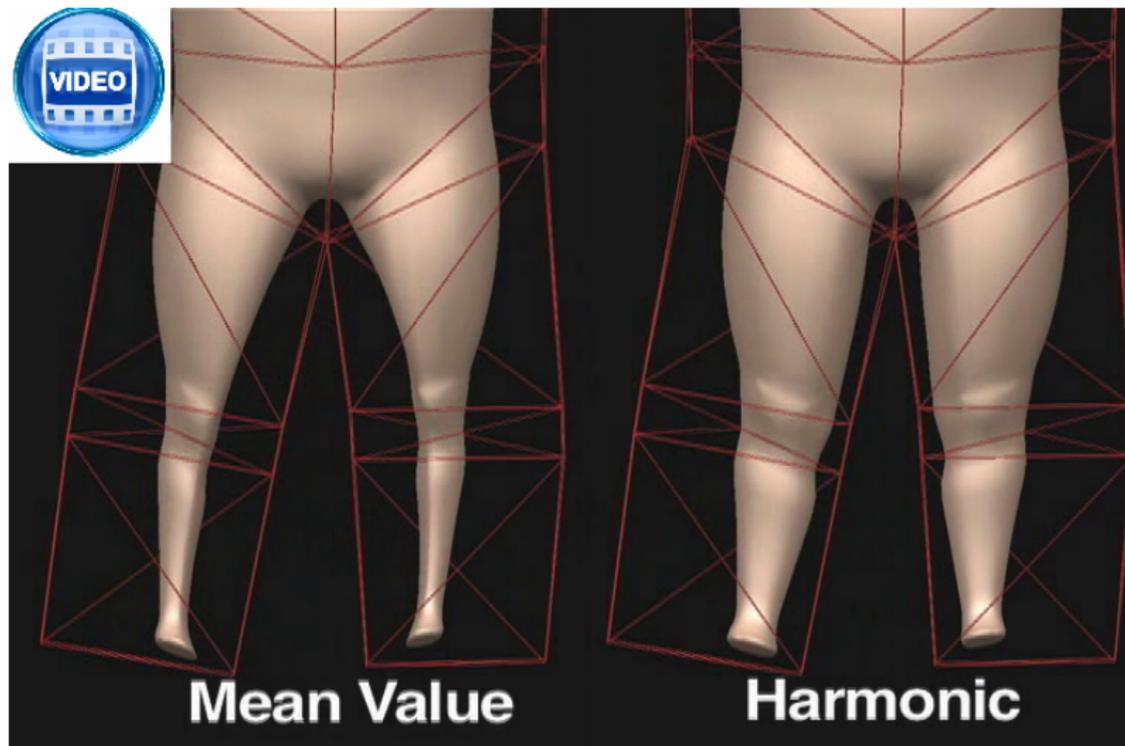
Mean-value



Harmoniques

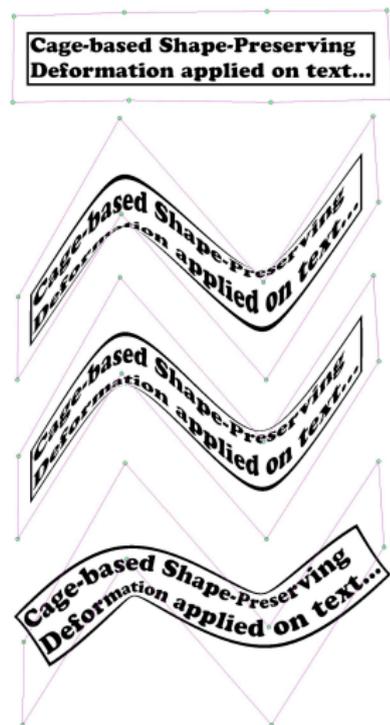
## Coordonnées harmoniques

Harmonic Coordinates for Character Articulation - P. Joshi, M. Meyer, T. DeRose - 2007



# Coordonnées de Green

- Basées sur la 3ème identité intégrale de Green (dérivée de fonctions harmoniques) [Nehari 1952]
- Prend en compte à la fois les **positions** et **normales** de la cage
- **Conformes** en 2D, quasi-conformes en 3D (préservent les angles)
- Extension à l'extérieur de la cage pour faire des **cages partielles**
- **Solution analytique**



# Coordonnées de Green

Green Coordinates - Y. Lipman, D. Levin, D. Cohen-Or - 2008

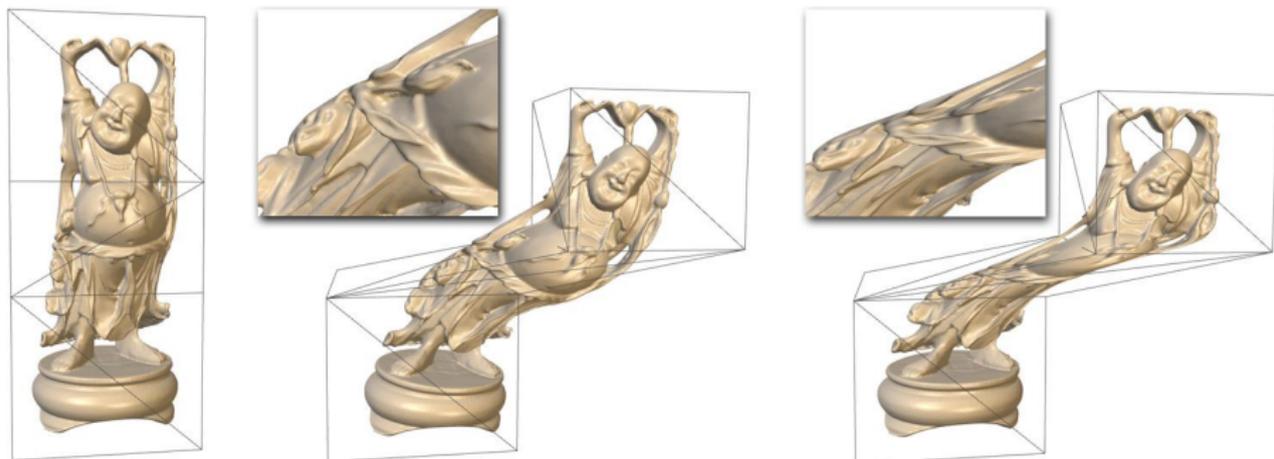


Figure 15: Deformation of a large model (1087K triangles) in real-time is shown in the middle (See the accompanying video), and on the right is the result using MVC.

# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages**
- 7 Conclusion

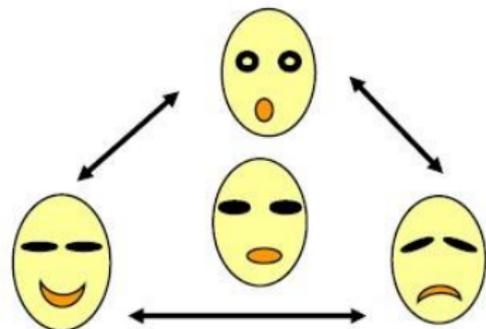
# Morphing

- Interpolation de **formes clés**
- Surtout pour animation faciale
- **Interpolation temporelle** : modéliser et stocker toutes les keyframes

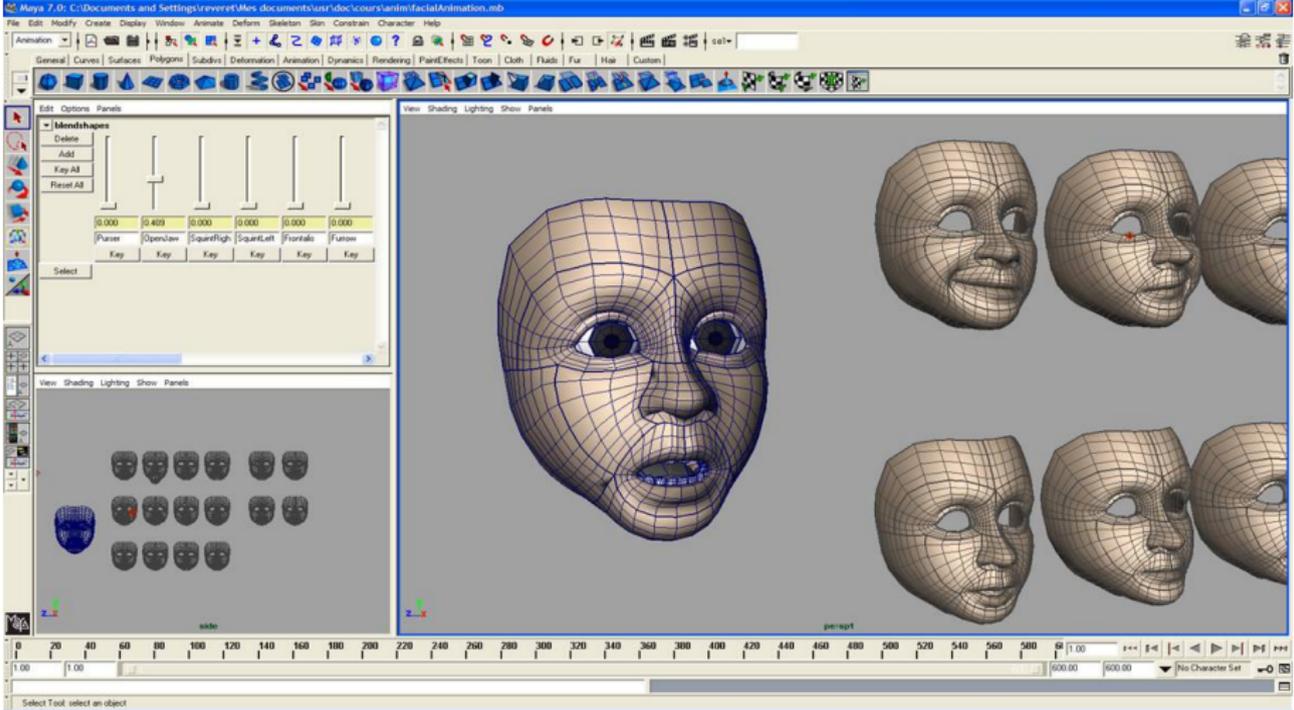


# Interpolation multi-cibles

- **Blendshapes**
- Modéliser des visages extrêmes à partir d'un visage neutre  
⇒ **formes de références**
- La forme 3D pour une key-frame donnée est une **combinaison linéaire** des formes de références
- Chaque sommet est l'interpolation linéaire de ce sommet aux key-frames



# Blenshapes



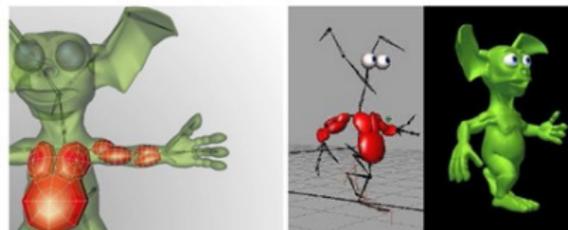
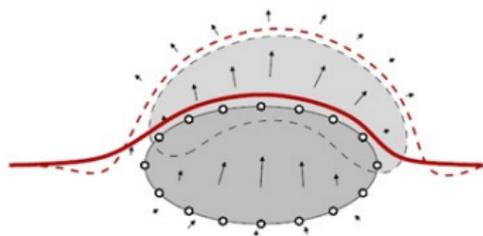
# Plan

- 1 Introduction
- 2 Déformations globales
- 3 Skinning linéaire
- 4 Améliorations du skinning linéaire
- 5 Animation par cage
- 6 Animation de visages
- 7 Conclusion**

- Animation par squelette :
  - ▶ Skinning linéaire rapide mais génère des artefacts
  - ▶ Correction des artefacts en utilisant des exemples fournis par des artistes
  - ▶ Utilisation des quaternions duaux qui imposent de bonnes propriétés à la transformation appliquée aux points du maillage
- Animation par cage :
  - ▶ Contrôle par un maillage simplifié
  - ▶ Choix des coordonnées primordial pour assurer de bonnes propriétés aux déformations

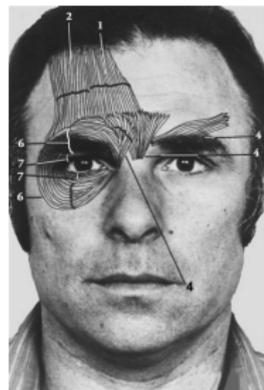
# Animation de surfaces - Questions ouvertes

- **Préservation de volume?** [Angelidis 07]



- **Animation faciale** : 2 domaines principaux :

- ▶ **expressions** : toute expression est une combinaison d'expressions de base communes à tout homme [Ekman 75]
- ▶ **conversation** : perception visuelle de la production de paroles (**lip-synching**)
  - ★ problème difficile
  - ★ solution courante : utilisation de **visèmes** (p,b,m/t,v)
  - ★ effet de **co-articulation**



- Cette semaine : skinning classique
- Projet : améliorations