

Animation et déformation

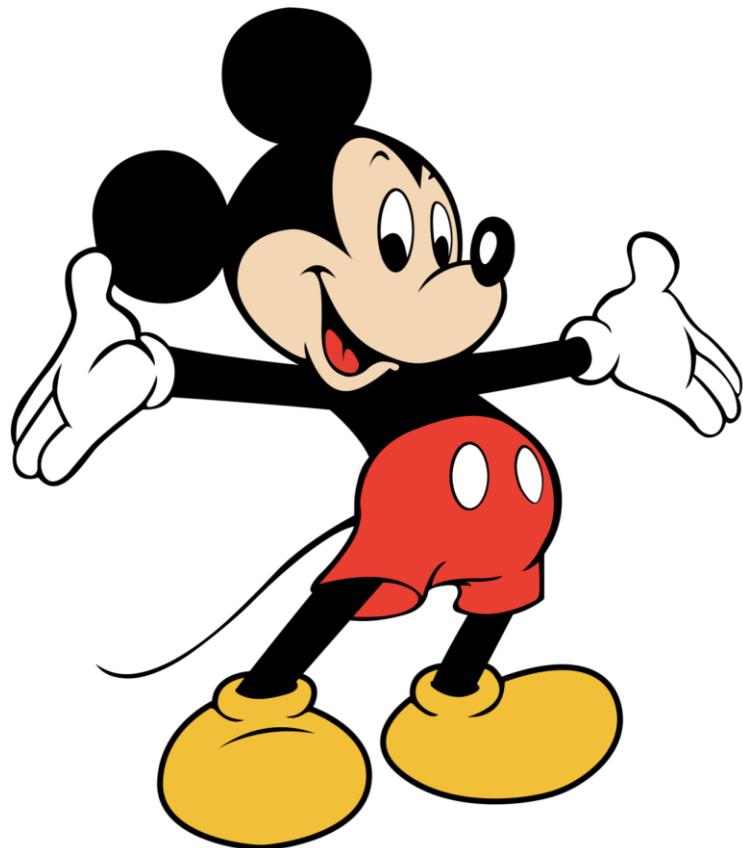
Master IMAGINA

Plan

- Introduction
- Animation
 - Images clés
 - Squelette
 - Skinning
- Deformation
 - Surface
 - Volume

Double héritage

Dessin animé



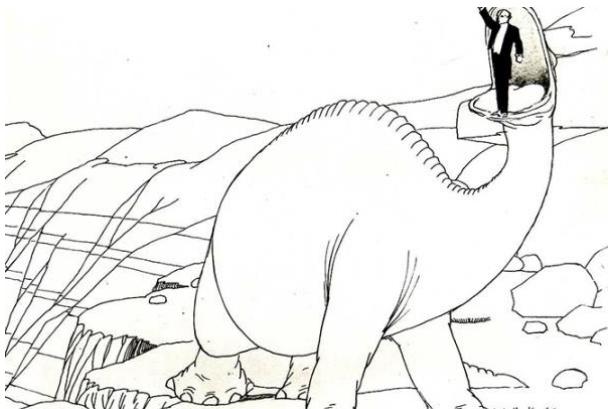
Robotique



Dessin animé

- Animation

Gertie le dinosaure (1914)



1^{ère} animation par image clés



- Données par un animateur et dessin des images intermédiaires par des « petites mains »

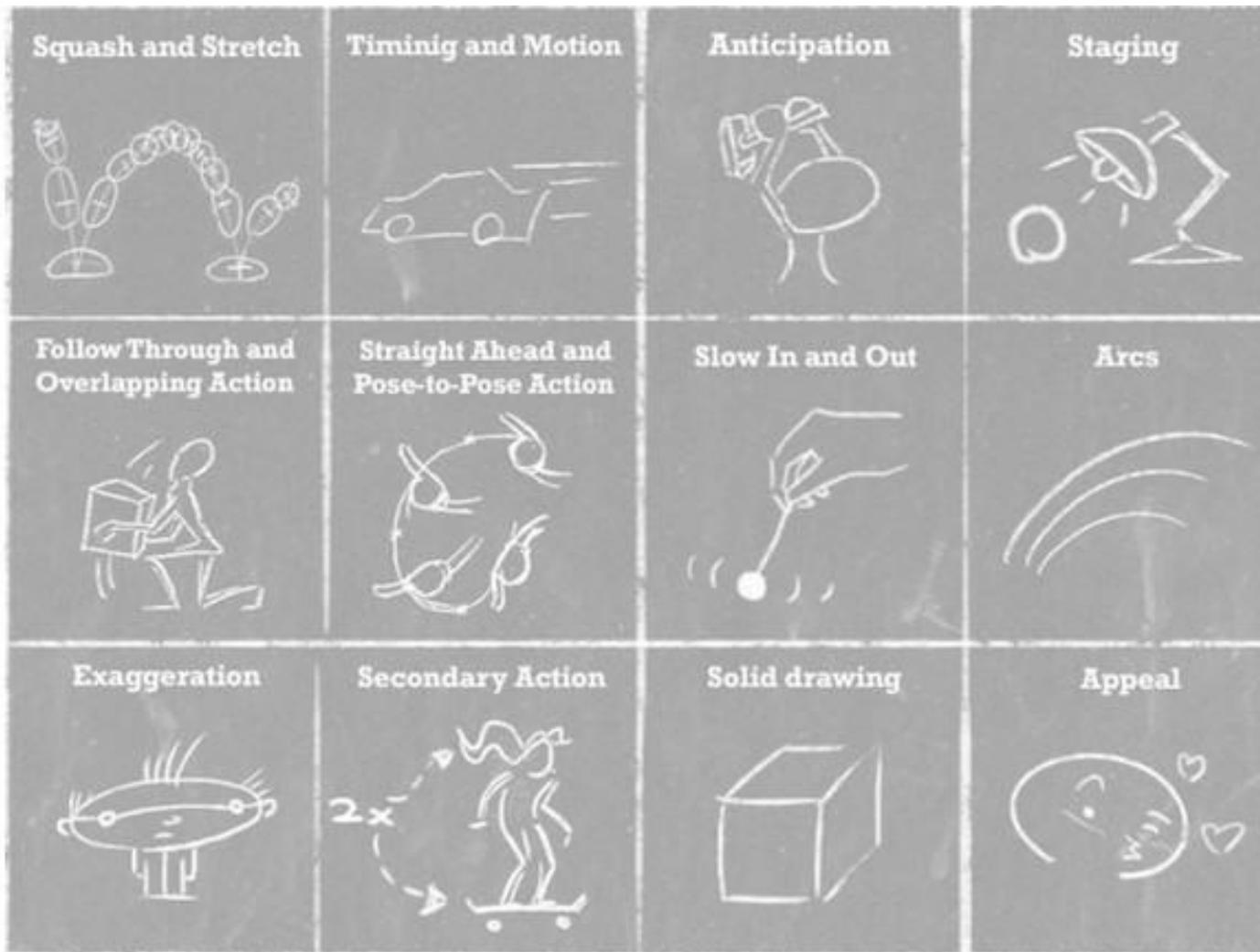
Animation par ordinateur

- Automatiser ce processus



12 principes d'animation

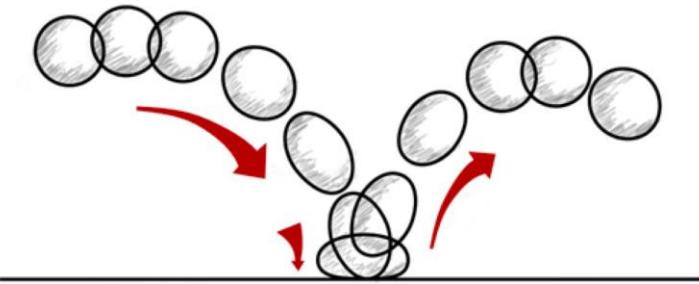
Disney Studios, 1930s



[Voir vidéo...](#)

1. Squash and stretch

- Distorsion de la forme pour accentuer le mouvement (avec préservation du volume)

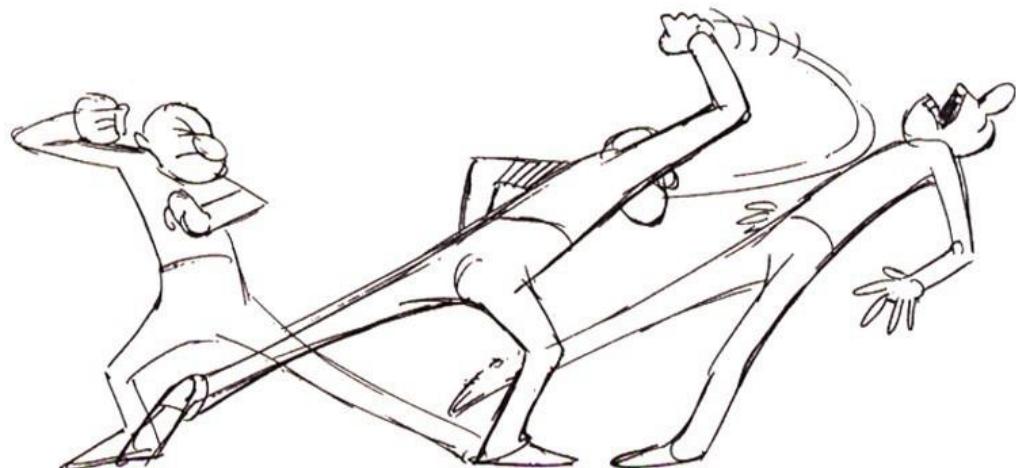


© Richard William



4. Anticipation

- Action inverse au mouvement pour l'accentuer



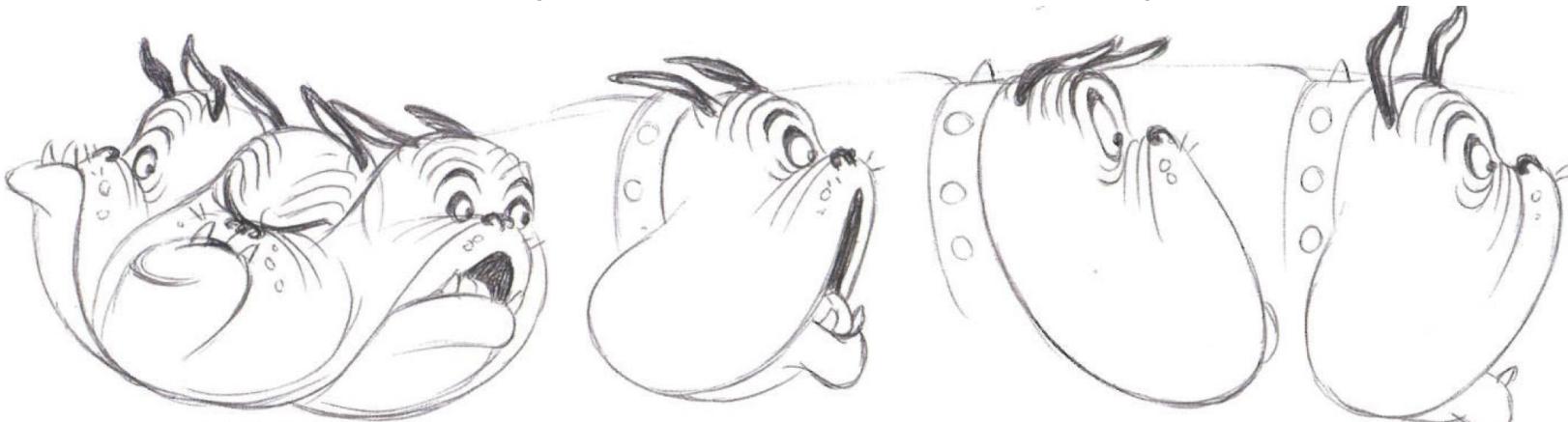
© Richard William



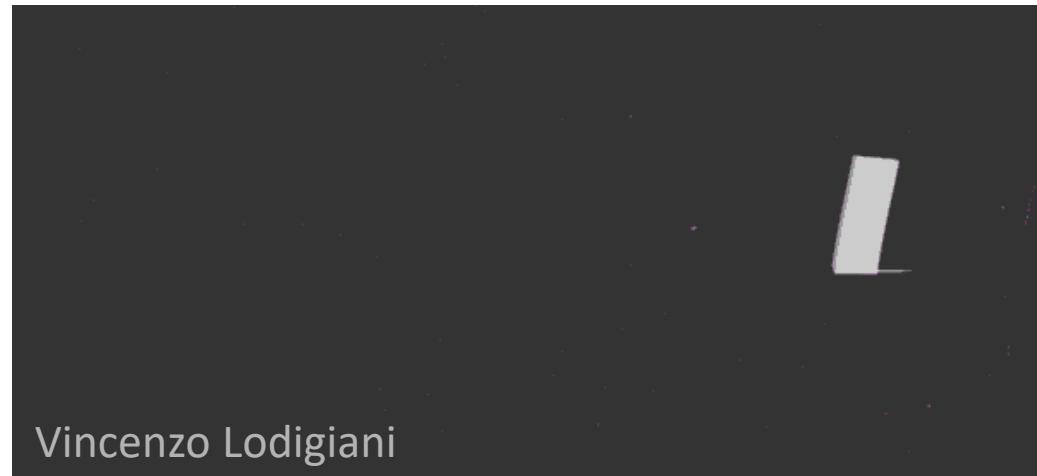
Vincenzo Lodigiani

5. Follow through & Overlapping

- Retards et anticipations de certaines parties de la forme



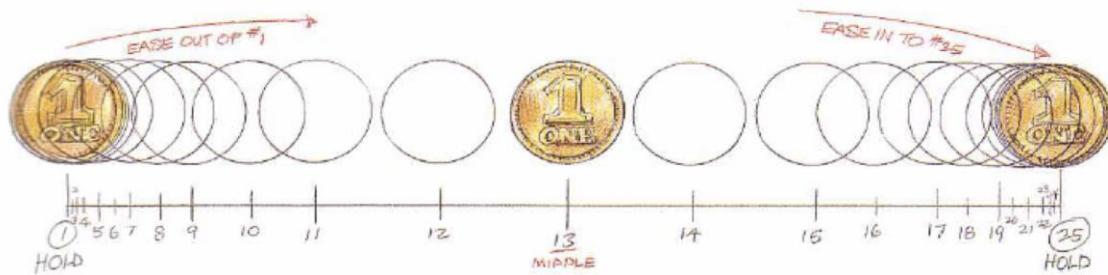
© Richard William



Vincenzo Lodigiani

6. Slow in, slow out

- Exagération de l'accélération et de la décélération aux extrémités de la trajectoire

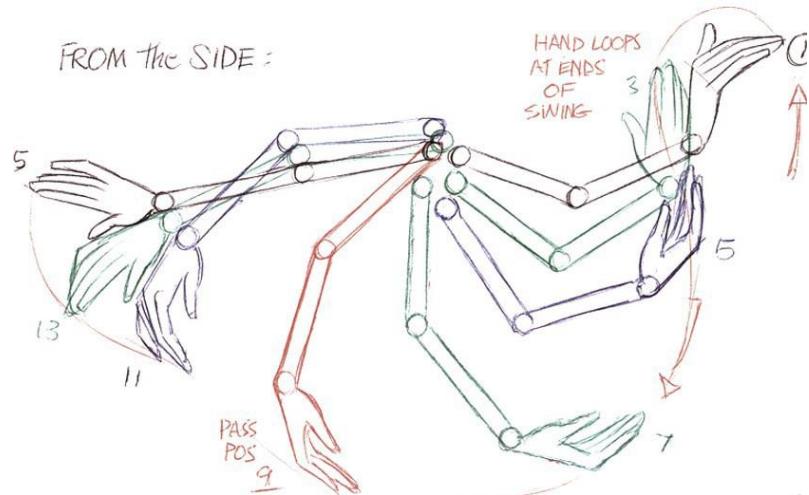


© Richard William



7. Arcs

- Rien ne bouge en ligne droite mais selon un arc



© Richard William



Débuts de l'animation 3D

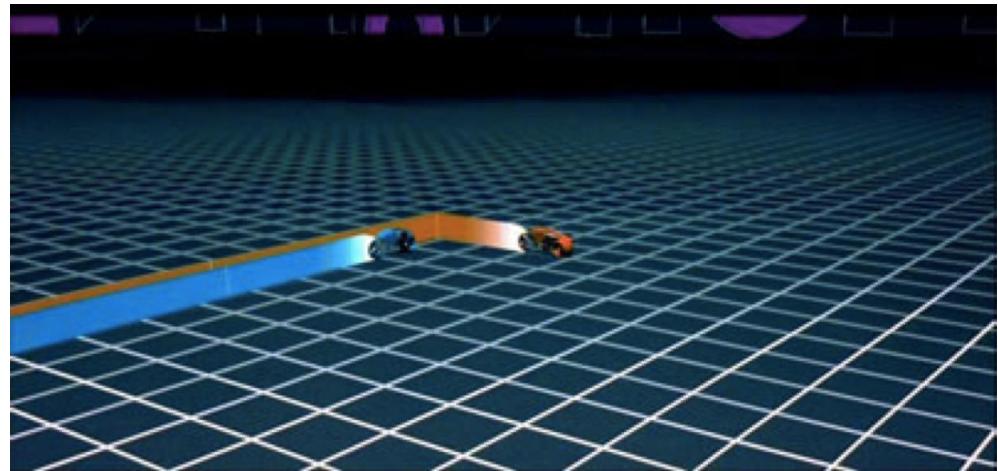
- Recherche académique



A Computer Animated Hand (1972)
Ed Catmull - Fred Parke

Débuts de l'animation 3D

- Effets spéciaux



Tron (1986)



Abysse (1989)

Débuts de l'animation 3D

- Court métrage d'animation 3D



Luxo Jr. (Pixar 1986)

Débuts de l'animation 3D

- Premier long métrage d'animation 3D



Toy Story, (Pixar 1995)



Animation par ordinateur

- Les modèles ont des paramètres :
 - Position de sommets
 - Normales
 - Points de contrôles de courbes
 - Angle de jointures
 - Couleurs,
 - Lumières...
- n paramètres définissent un espace d'état de n dimension
- Un jeu de valeurs de ces n paramètres = point dans cet espace

Animation par ordinateur

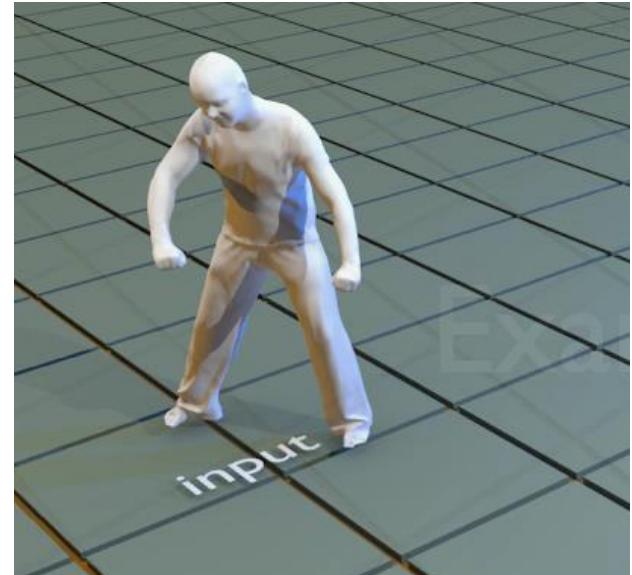
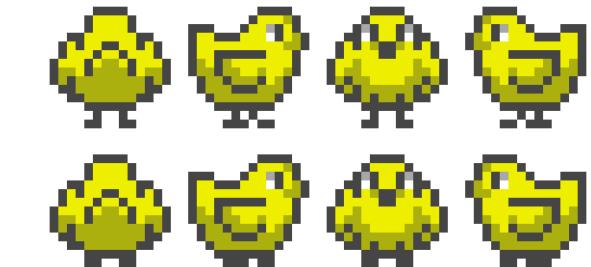
- Une animation est définie par un chemin dans cet espace d'état
- Pour produire une animation :
 1. Commencer au début du chemin
 2. Fixer les paramètres de votre modèle
 3. Rendre une image
 4. Bouger au point suivant du chemin dans l'espace d'état
 5. Aller à 2
- Le chemin est défini par un ensemble de courbe de mouvement (une pour chaque parameter)
- Animation = spécifier la trajectoire dans l'espace d'état

Techniques d'animation

- Traditionnelle (image par image)
- Par image clés (keyframing)
- Techniques procédurales
- Techniques comportementales (e.g., flocking)
- Basées sur la performance (motion capture)
- Basées sur la physique (dynamique)

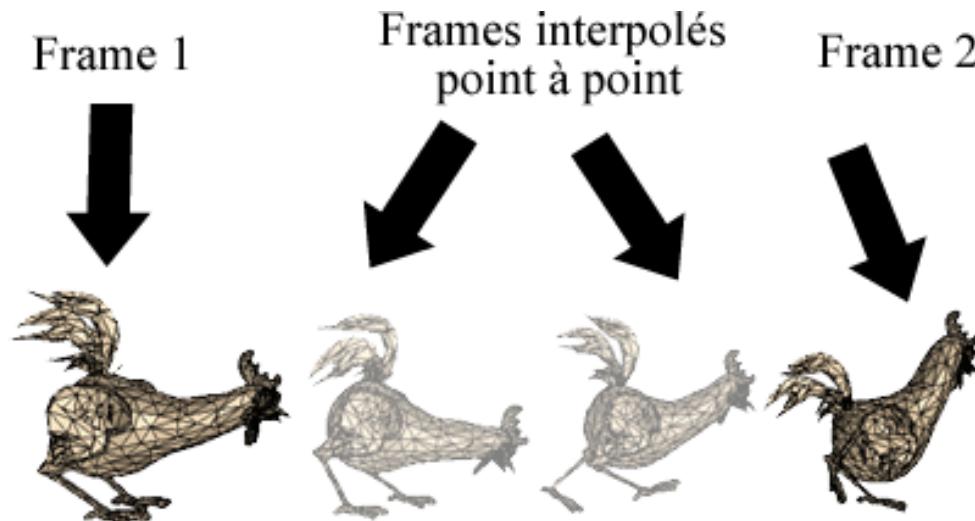
Animation classique

- Chaque image de notre animation est stockée dans une unité complète, séparée du reste.
- En 2D on parle de “Sprite”
 - Images 2D affichées en séquence.
- En 3D on parle d'une suite de maillages (mesh)
 - Suite de modèles 3D
 - rendus en séquence.



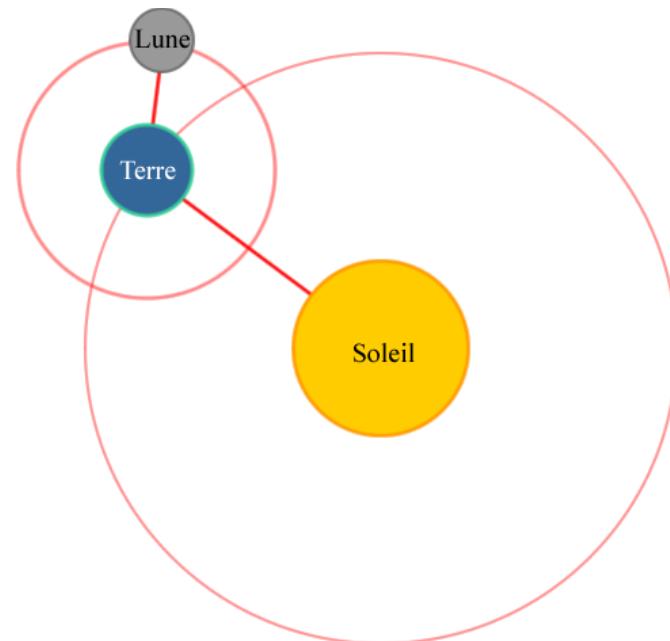
Animation par images clés

- Variation pour l'animation par key-frame classique en 3D :
 - Le modèle contient un nombre de frame réduit pour l'animation.
 - Chaque vertex est interpolé un à un entre les images pour combler les trous



Animation hiérarchique

- Exprimer l'animation d'un objet en fonction de transformations relatives à celles d'un autre objet.
- Exemple: Modélisation hiérarchique d'un système solaire (simplifié)

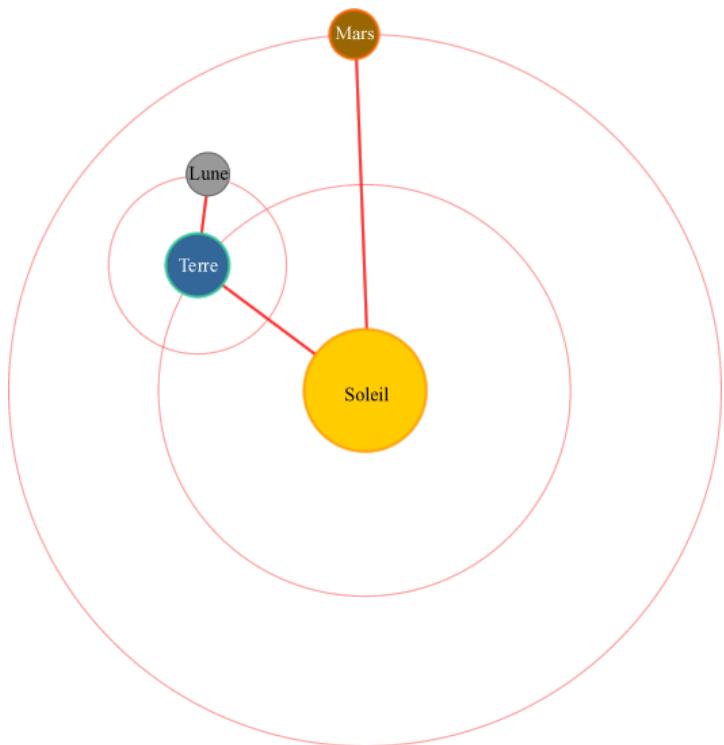


Animation hiérarchique

- Vocabulaire:
 - Chaque objet dans la scène est appelé un noeud.
 - Si la transformation d'un noeud a dépend d'un noeud b, on dit que a est l'enfant de b.
 - Le noeud dont un enfant dépend est appelé son parent.
 - Un noeud n'ayant aucun parent est appelé la racine.
 - Un noeud n'ayant aucun enfant est une feuille.
 - Un noeud peut avoir plusieurs enfants mais un seul parent.

Animation hiérarchique

- Exemple:
 - Soleil → Racine
 - Lune et Mars → Feuilles
 - Terre → Parent de Lune
 - Lune → Enfant de Terre
 - Soleil → Parent de Terre et mars
 - Etc...

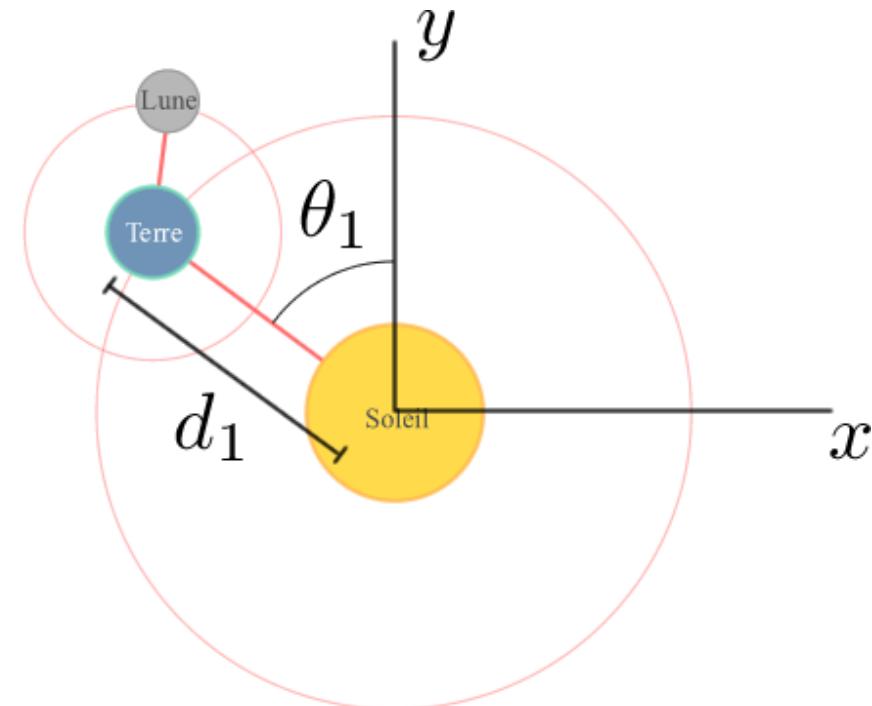


Animation hiérarchique

- Transformations d'un nœud par rapport à son parent → réduction de dimensionnalité
 - Contraintes de distance (translation)
 - Contraintes d'orientation (rotation)

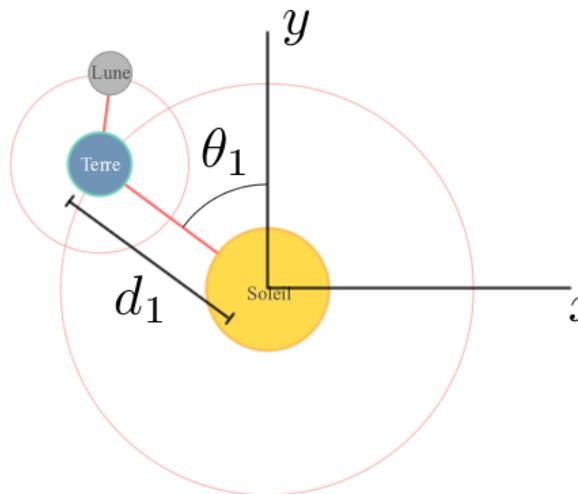
Exemple:

La position d'une planète/lune se définie par une distance et un angle par rapport au corps autour duquel elle tourne. (On sauve la complexité requise pour calculer la position de l'objet de façon absolue).

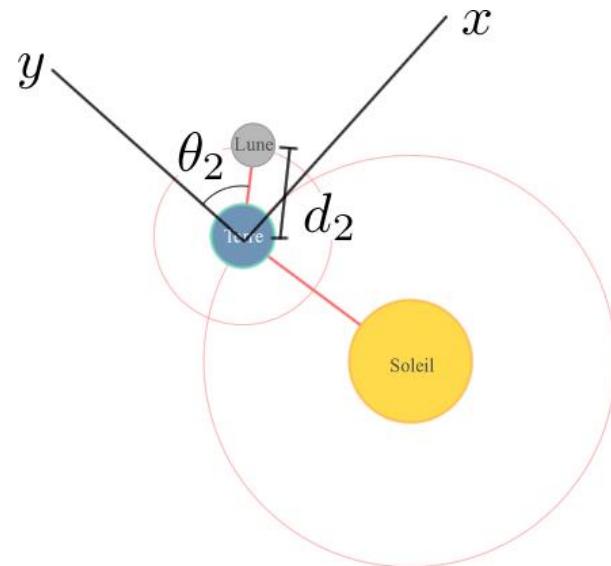


Animation hiérarchique

- Chaque noeud possède son propre système de coordonnées locales.



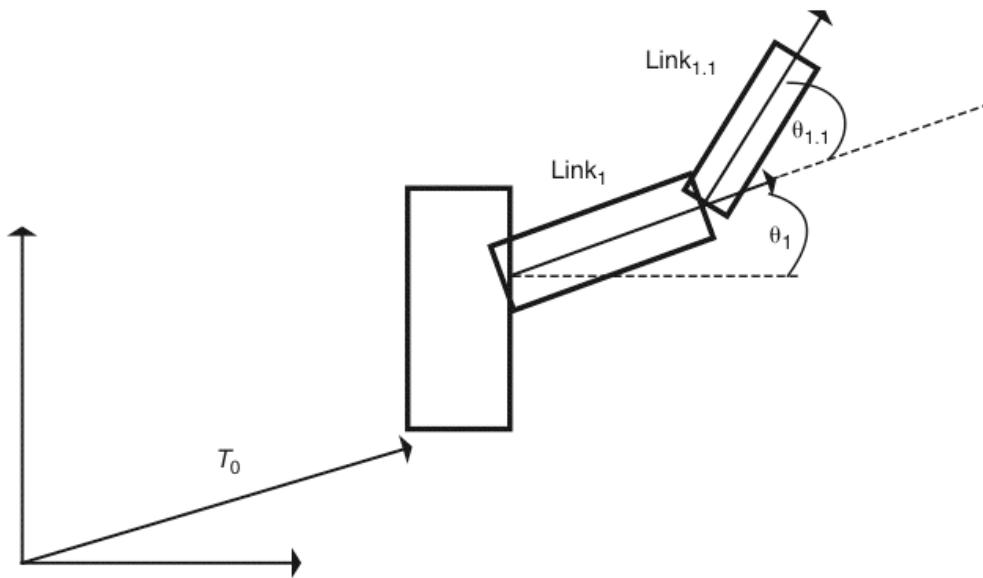
Système local du Soleil, la position de la Terre étant exprimée par rapport celui-ci.



Système local de la terre, la position de la lune étant exprimée par rapport à celui-ci

Animation hiérarchique

- Transformation finale d'un nœud :
 - Concaténation des transformations en partant de la racine jusqu'au nœud en question.



Rick Parent, 2008

$$\text{Transf}_{1.1} = T_0 T_1 R_1(\theta_1) T_{1.1} R_{1.1}(\theta_{1.1})$$

Animation hiérarchique

- Transformation globale → transformer les points associés se fait simplement.

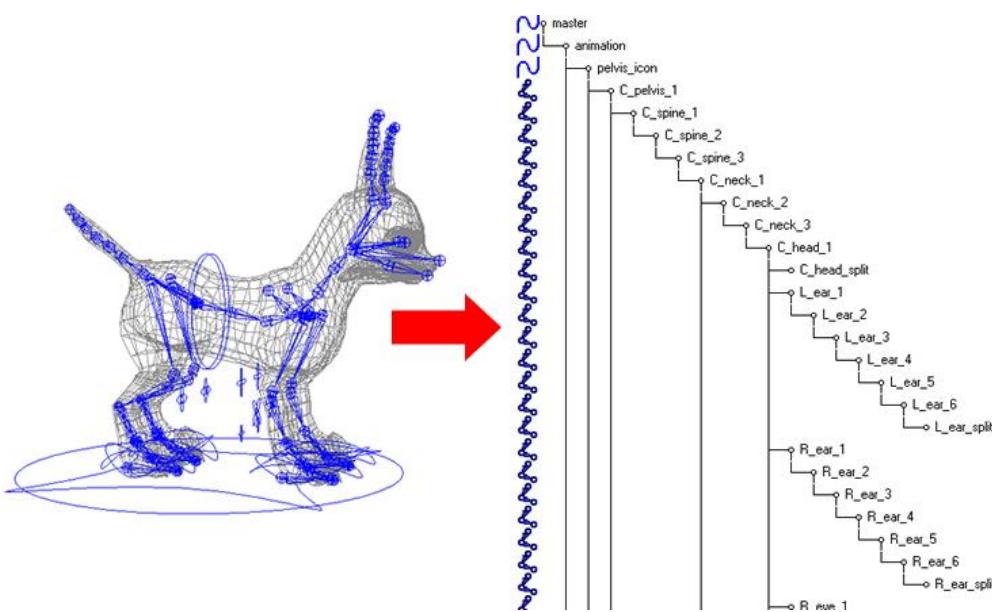
Ex : (transformation du point V1 associé au nœud 1.1)

$$Transf_{1.1} = T_0 T_1 R_1(\theta_1) T_{1.1} R_{1.1}(\theta_{1.1})$$

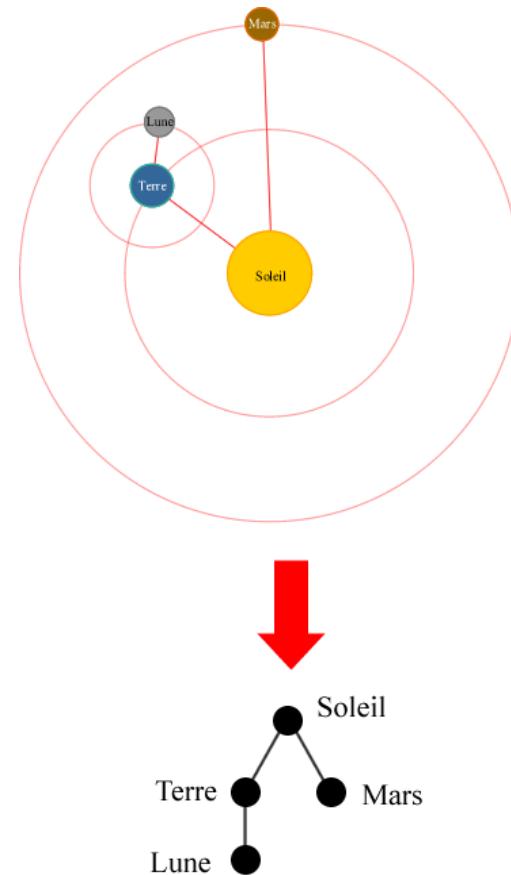
$$V'_1 = (Transf_{1.1})(V_1)$$

Animation hiérarchique

- La hiérarchie entre noeuds peut être représentée sous la forme d'un graphe dirigé acyclique.



Ariane Vaillancourt, 2009

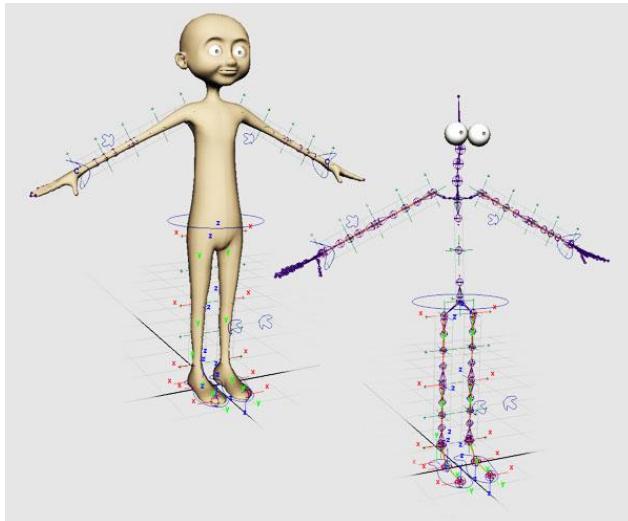


Pourquoi utiliser les squelettes ?

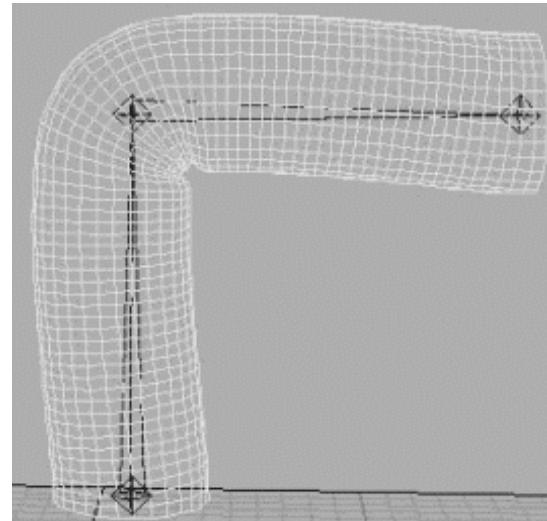
- Beaucoup plus simple à animer.
 - Manipulation d'articulations (sinon chaque vertex indépendamment)
- Beaucoup plus compact que l'animation par keyframe classique.
 - Modèle + squelette stocké 1 fois en mémoire → animation du squelette.
- Assez simple pour permettre l'interaction avec l'environnement
- Possibilité de réutiliser le même squelette pour plusieurs modèles 3D. (Mêmes animations, apparences différentes.)
- Utilisation de méthodes d'interpolation de complexité supérieure à linéaire.
 - Rendu possible puisque l'interpolation se fait sur les articulations, moins nombreuses que les points qui composent le modèle.

Squelette d'animation

- Le modèle 3D contenant la hiérarchie peut être perçu comme une enveloppe.
 - Les différents points (vertices) qui composent le modèle sont reliés à différents nœuds de la hiérarchie.
 - L'association d'un point à un élément de la hiérarchie est appelé “skinning”



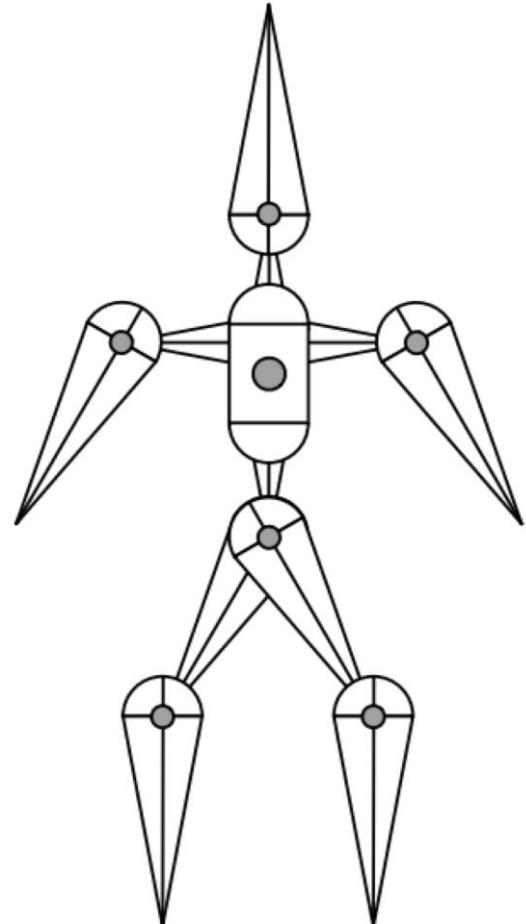
“Lucien”, Mathias Aubry, 2008



David Lanier, 2002

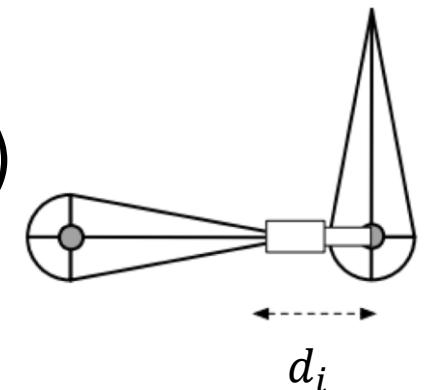
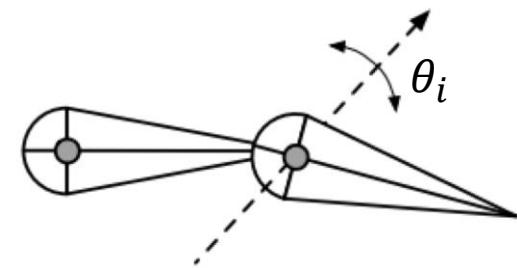
Squelette d'animation

- Modèle décomposé en une **hiérarchie de repères contraints**
- Pas de géométrie, que des solides appelés **os (*bones*)** et des liaisons appelés **articulations (*joins*)**

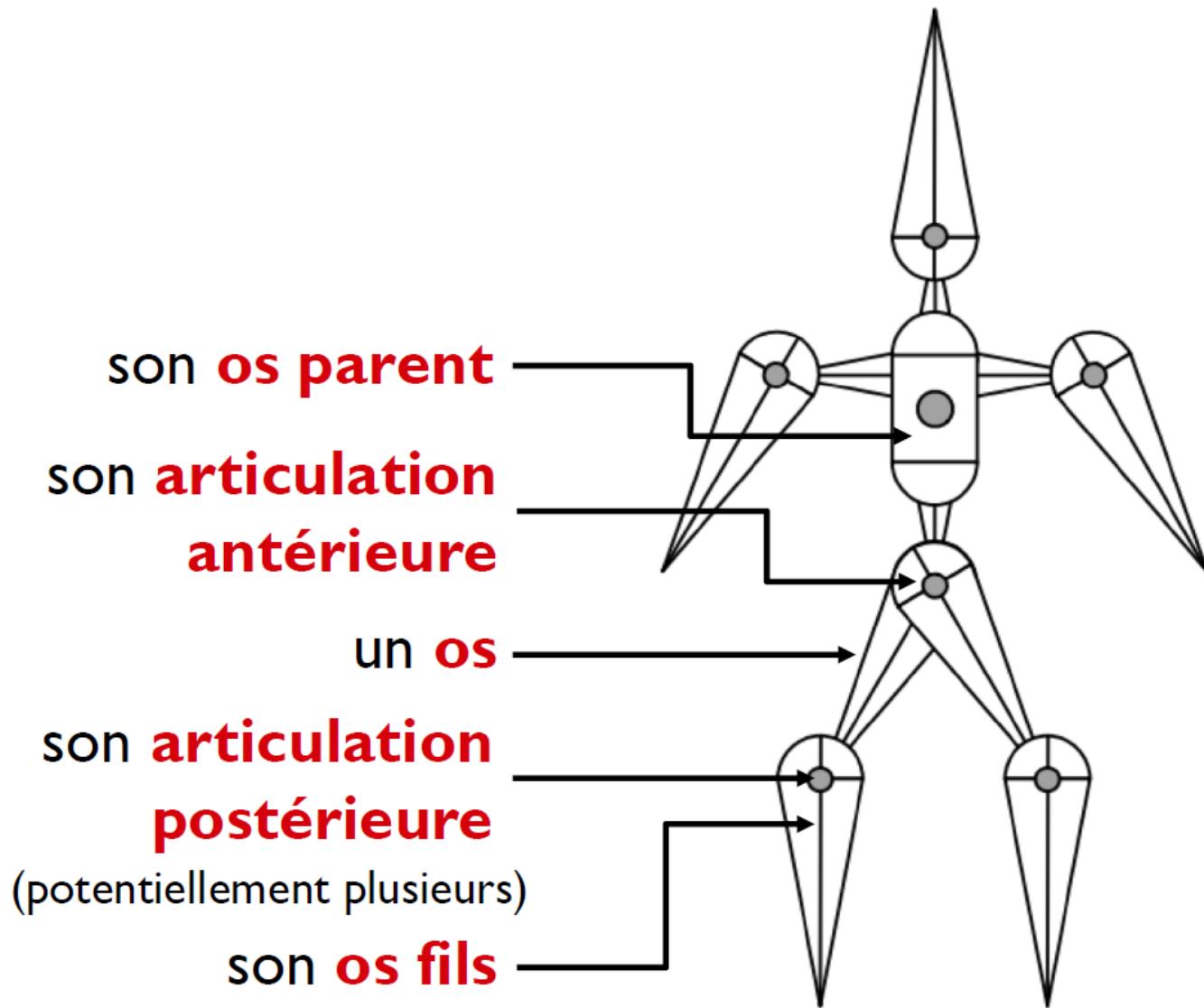


Articulations

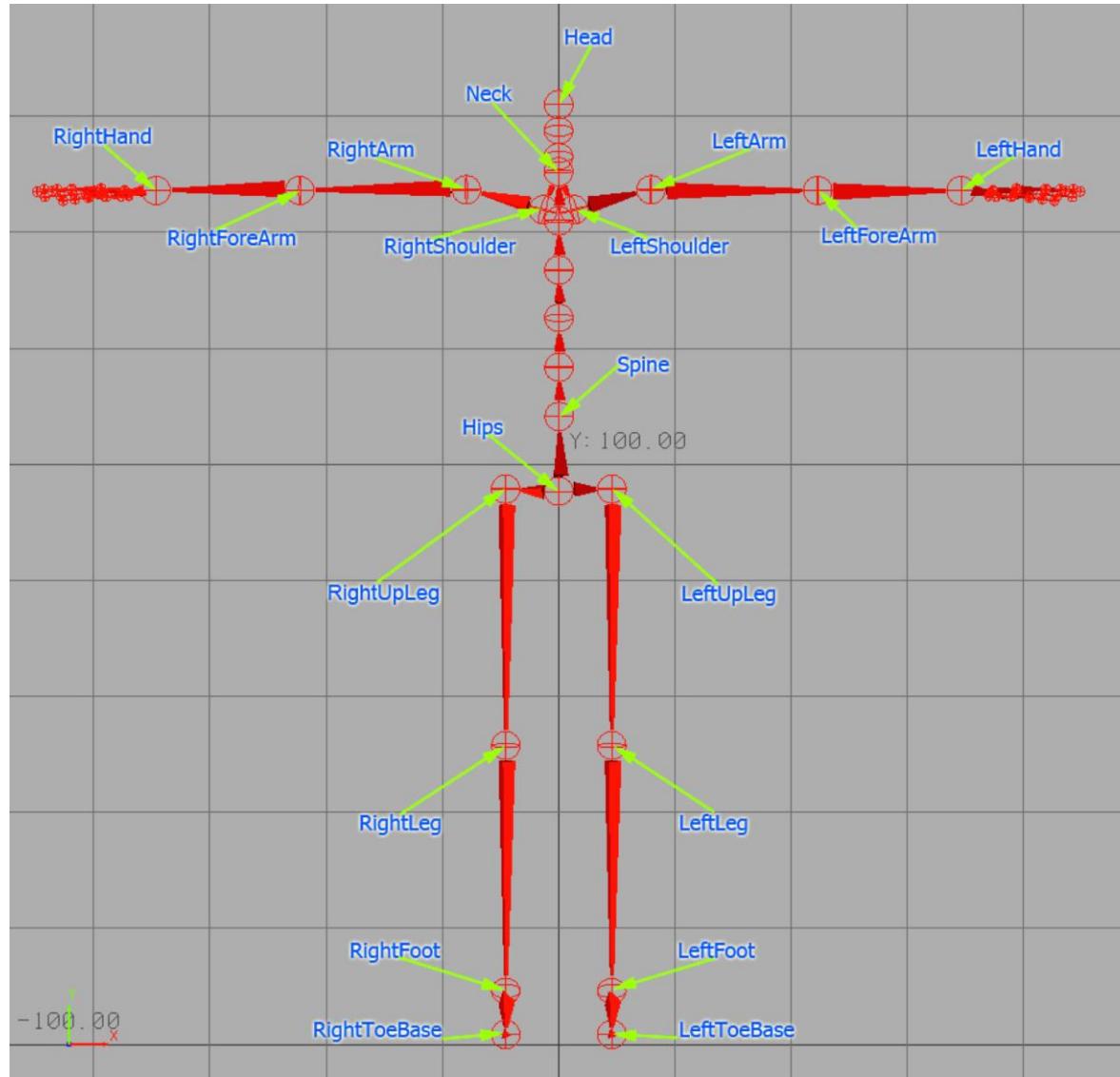
- En général **6 degrés de liberté (DOF)** :
 - 3 translations
 - 3 rotations
- En pratique, souvent des **contraintes** :
- **Pin** – rotation autour d'un axe (1 DOF)
- **Ball** – rotation arbitraire (3 DOF)
- **Prism** – translation selon un axe (1 DOF)



Chaîne articulée

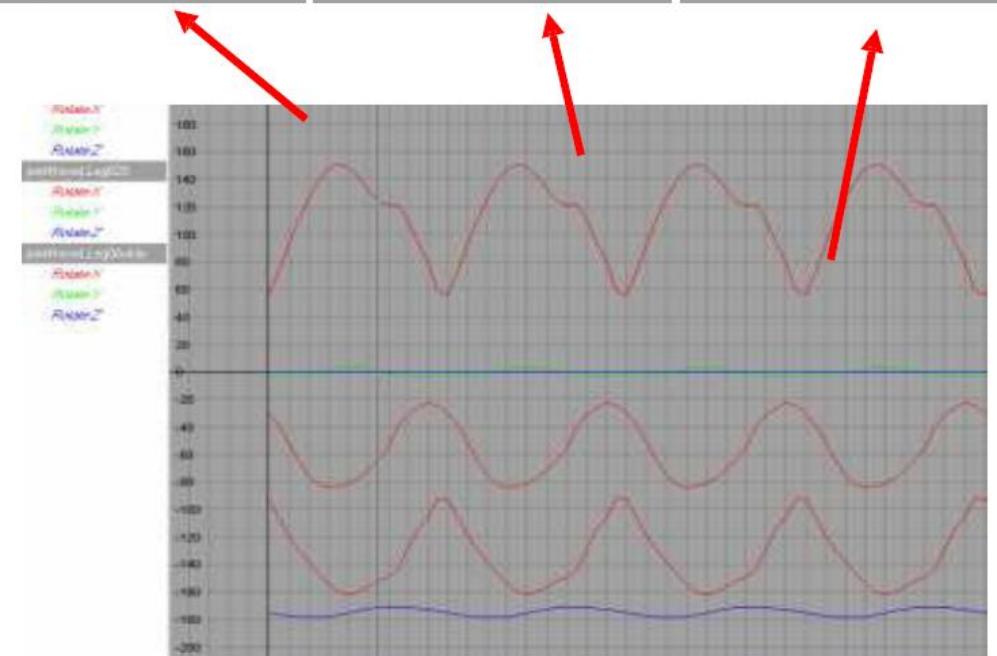
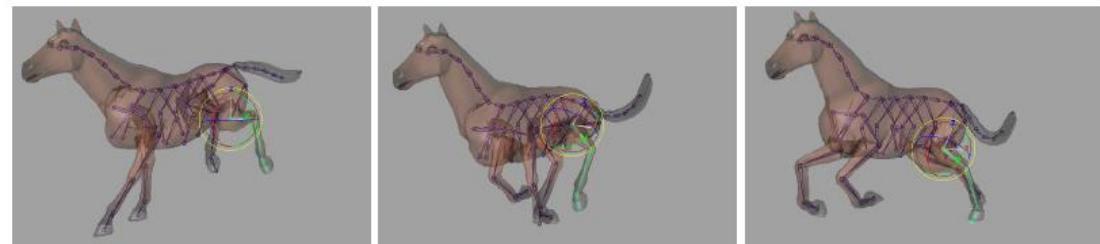
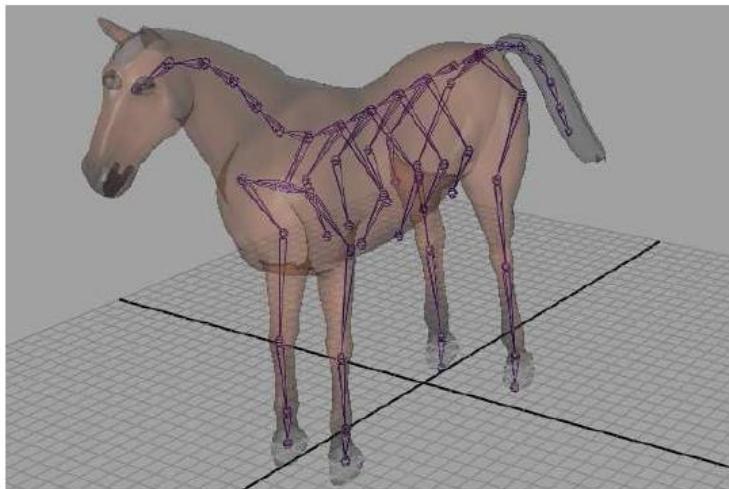


Squelette d'animation

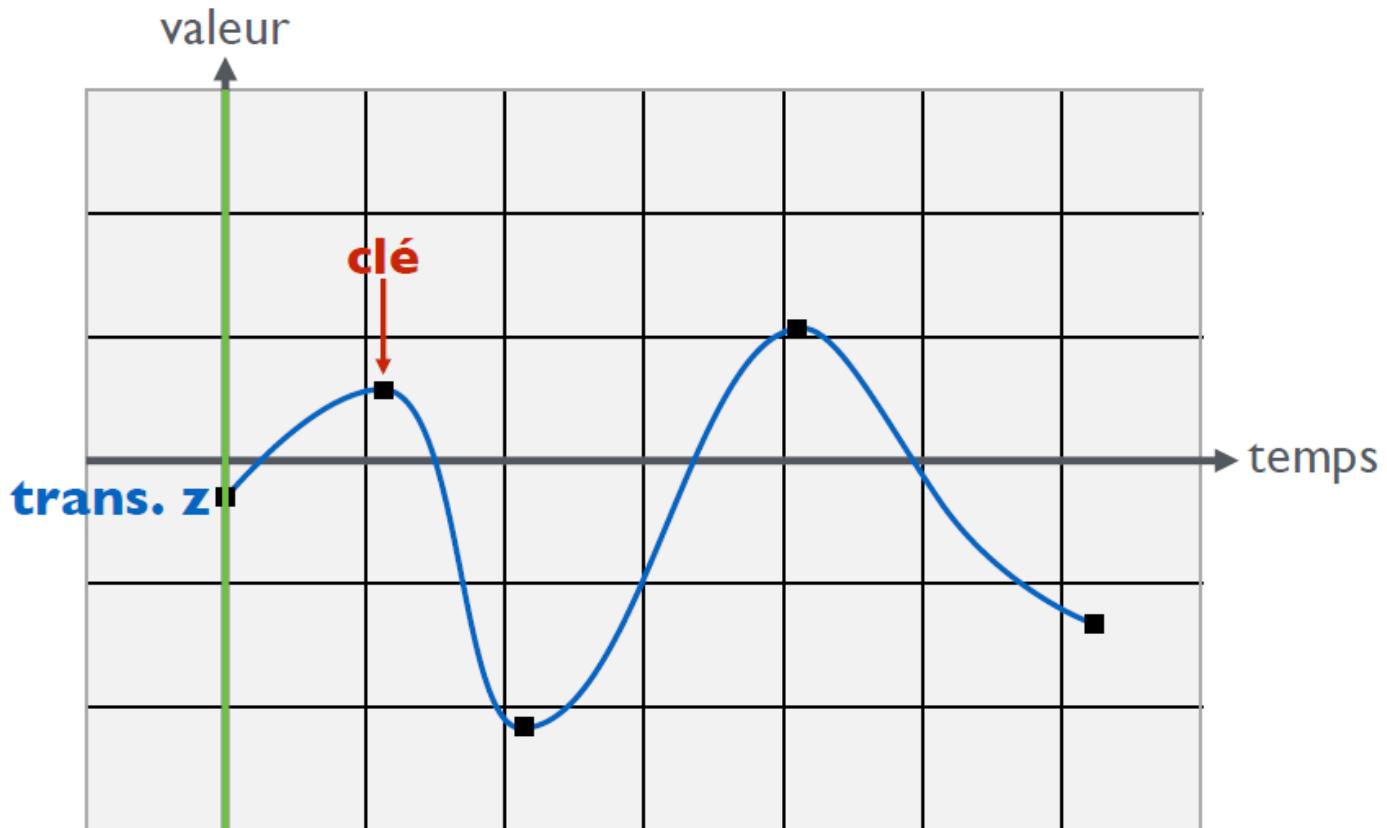


Animation 3D

- **Squelette + interpolation selon une courbe**



Courbe d'animation

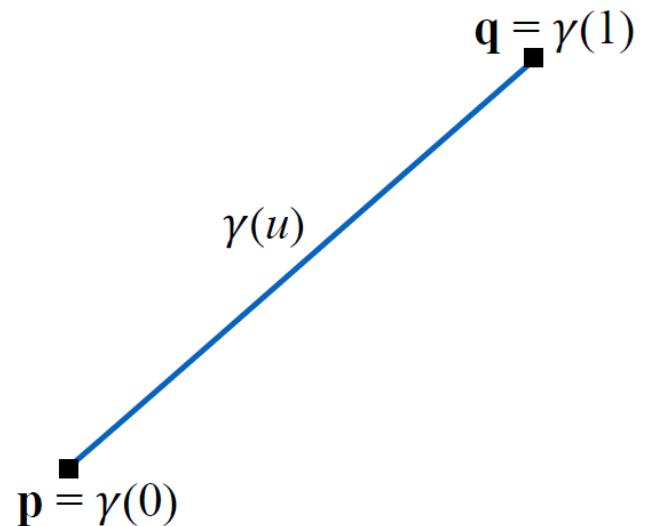


Interpolation

- **Entrées** : valeurs clés (positions)
- **Objectif** : interpoler ces valeurs clés par une fonction continue \Rightarrow courbes d'animation
- **Fonction linéaire** :

$$\gamma(u) = (1-u) \mathbf{p} + u \mathbf{q}$$

Continuité C^0
(position)



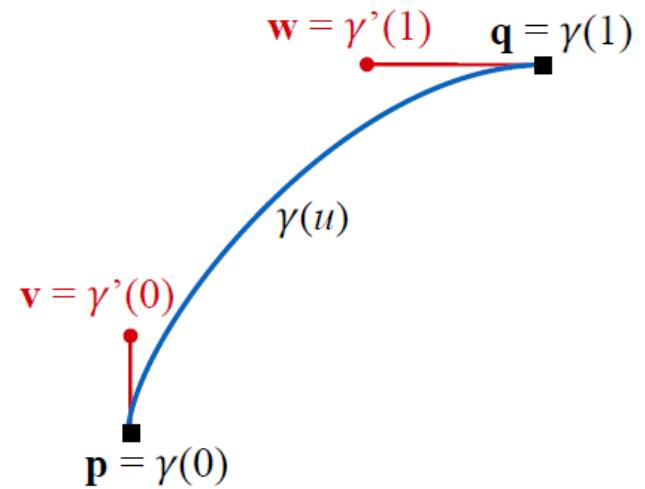
Interpolation

- **Entrées** : valeurs clés (positions et vitesses)
- **Objectif** : interpoler ces valeurs clés par une fonction continue \Rightarrow courbes d'animation

- **Polynôme de degré 3** de la forme :

$$\gamma(u) = \sum_{n=0 \dots 3} a_n u^n$$

Continuité C^1 entre deux clés
(positions et tangentes)



- Déterminer les paramètres an interpolant (ou approximant) au mieux les valeurs clés
- Degré 3 \Rightarrow 4 contraintes suffisantes pour résolution exacte

Interpolation

- Polynômes d'Hermitte
- Splines de Catmull-Rom
- Interpolation de Bézier cubique
- ...

Interpolation des rotations

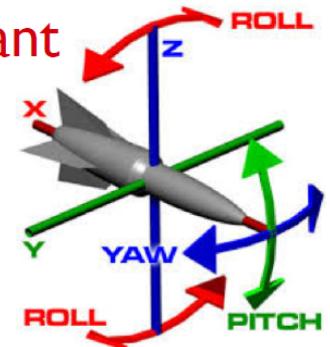
Angles d'Euler

- 1 angle par axe : $(\theta_x, \theta_y, \theta_z)$

- $M = R_{x,\theta_x} R_{y,\theta_y} R_{z,\theta_z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$

⇒ attention, l'ordre des multiplications est important

- Interpoler chaque angle
⇒ intuitif et peu couteux



Limitation

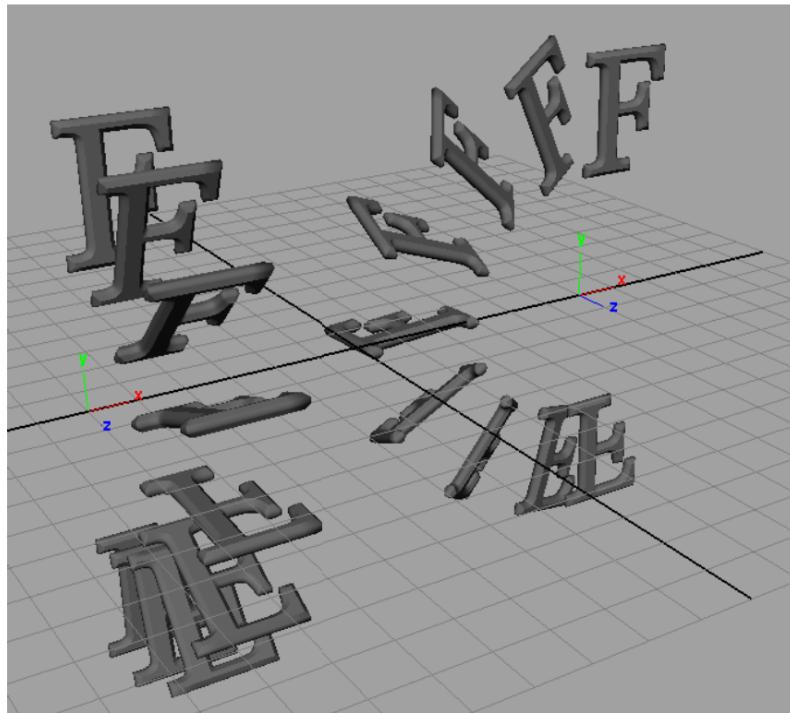
Non-unicité de la trajectoire

$$[\theta_x, \theta_y, \theta_z] = [0, 0, 0]$$



$$[\theta_x, \theta_y, \theta_z] = [\pi, 0, 0]$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$



$$[\theta_x, \theta_y, \theta_z] = [0, 0, 0]$$



$$[\theta_x, \theta_y, \theta_z] = [0, \pi, \pi]$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Quaternions

Méthode la plus populaire pour représenter une orientation dans le domaine de l'animation par ordinateur.

- Composé de 4 nombres, soient 1 angle et 3 nombres qui composent un vecteur.

Se note:

$$q = [s, x, y, z]$$

Ou

$$q = [s, \boldsymbol{v}], \boldsymbol{v} \text{ étant le vecteur } (x, y, z)$$

Rotation par quaternion

Rotation 3D = quaternion unitaire

($\|q\|=1 \Leftrightarrow$ point sur la sphère unité 4D)

$$\mathbf{v}' = R_{\theta, \mathbf{n}} \mathbf{v} \Leftrightarrow [0, \mathbf{v}'] = q[0, \mathbf{v}]q^{-1}$$

avec $q = [\cos(\theta/2), \sin(\theta/2)\mathbf{n}]$

et $q^{-1} = [\cos(\theta/2), -\sin(\theta/2)\mathbf{n}]$

(...en développant et simplifiant, on retrouve la formule de Rodrigues)

Propriétés :

- rotation de $\theta + 2\pi$: $[-\cos(\theta/2), -\sin(\theta/2)\mathbf{n}] = -q$
- rotation de $-\theta$: $[\cos(\theta/2), -\sin(\theta/2)\mathbf{n}] = q^{-1}$

Composition de rotations = multiplication

si q_1 représente la rotation R_1 et q_2 représente R_2 ,

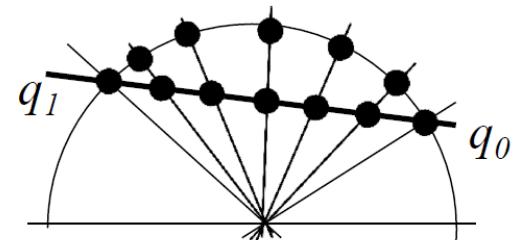
alors q_1q_2 représente la rotation R_1R_2

Interpolation des rotations

Interpolation linéaire (lerp)

$$\text{lerp}(q_0, q_1, \alpha) = (1-\alpha)q_0 + \alpha q_1$$

\Rightarrow vitesse angulaire non constante



Puissance

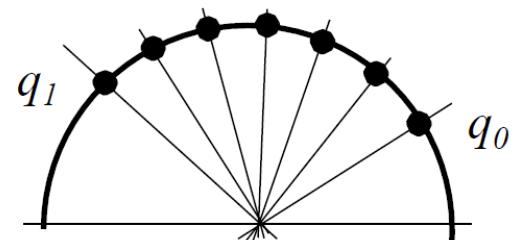
$$[\cos(\theta/2), \sin(\theta/2)\mathbf{n}]^\alpha = [\cos(\alpha\theta/2), \sin(\alpha\theta/2)\mathbf{n}]$$

avec $\alpha \in [0,1]$

Interpolation sphérique (slerp)

[Ken Shoemake 85]

$$\begin{aligned} \text{slerp}(q_0, q_1, \alpha) &= q_0(q_0^{-1}q_1)^\alpha \\ &= (q_1q_0^{-1})^\alpha q_0 \end{aligned}$$



Cinématique

Mouvement non-constraint

- Hucher la tête, faire un signe de la main...
⇒ Cinématique directe (FK)

Mouvement constraint

- Attraper un objet, marcher sur le sol...
⇒ Cinématique inverse (IK)

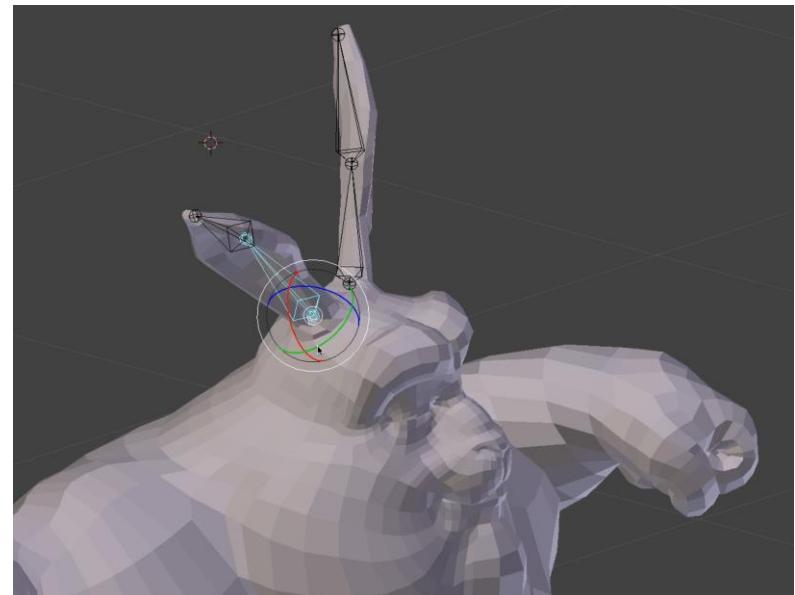
Cinématique directe

Entrées : configuration de la chaîne = valeurs des paramètres des articulations à un temps donnée :

$$\mathbf{q} = (\theta_1, \theta_2, \theta_3, \dots, d_1, d_2, \dots)$$

Sortie : position \mathbf{p} d'une extrémité de la chaîne (coordonnées cartésiennes + orientation, 3 ou 6D)

On **calcule** : $\mathbf{p} = f(\mathbf{q})$



Cinématique inverse

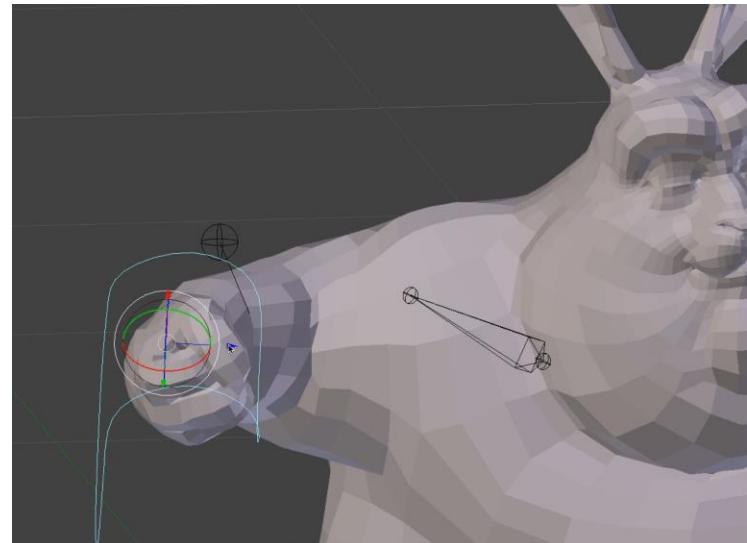
Entrée : position à atteindre \mathbf{p}

Sortie : valeurs des paramètres des articulations

$$\mathbf{q} = (\theta_1, \theta_2, \theta_3, \dots, d_1, d_2, \dots)$$

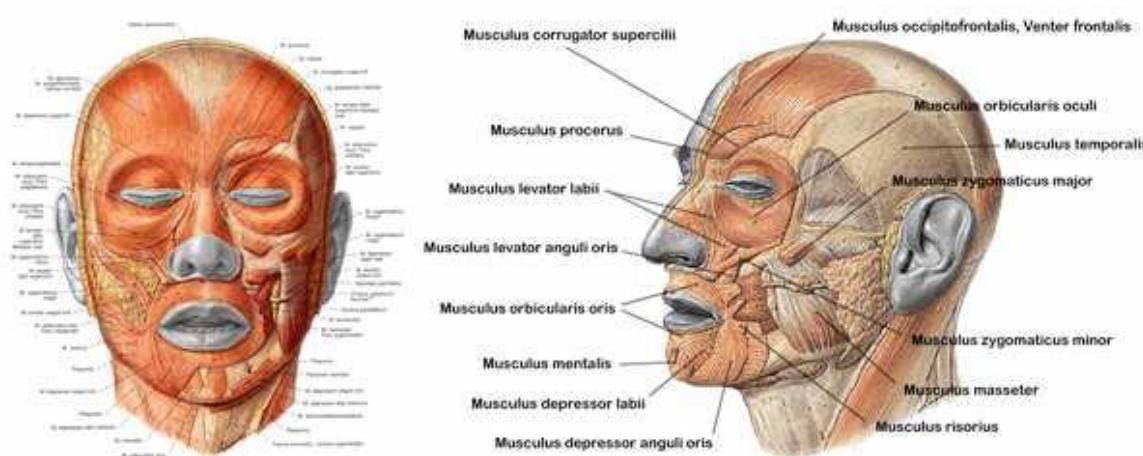
On veut trouver \mathbf{q} tel que

$$f(\mathbf{q}) = \mathbf{p} \text{ c'est à dire } \mathbf{q} = f^{-1}(\mathbf{p})$$



Déformation de surface

- Squelette d'animation vs. modèle visuel
- En réalité, forme visible composée de tissus organiques



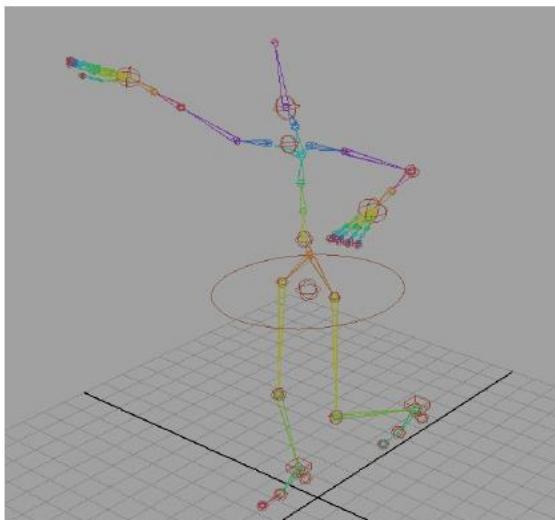
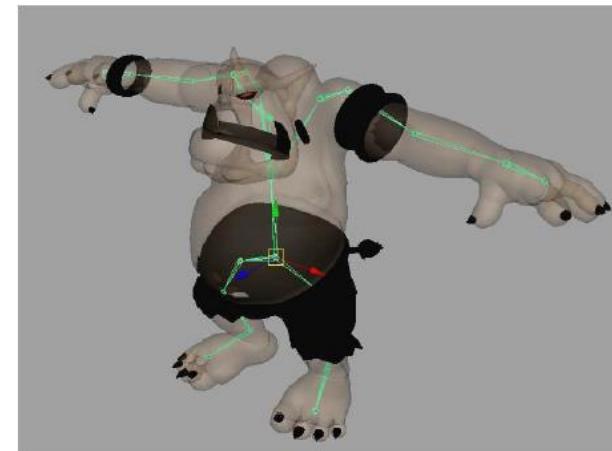
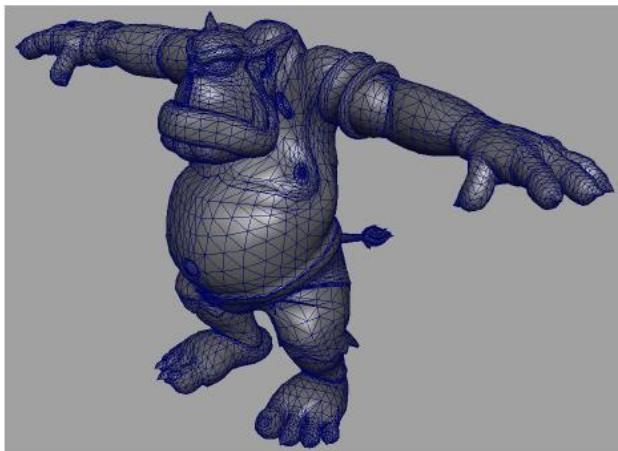
Modèle visuel

- Le niveau de réalisme dépend de l'application

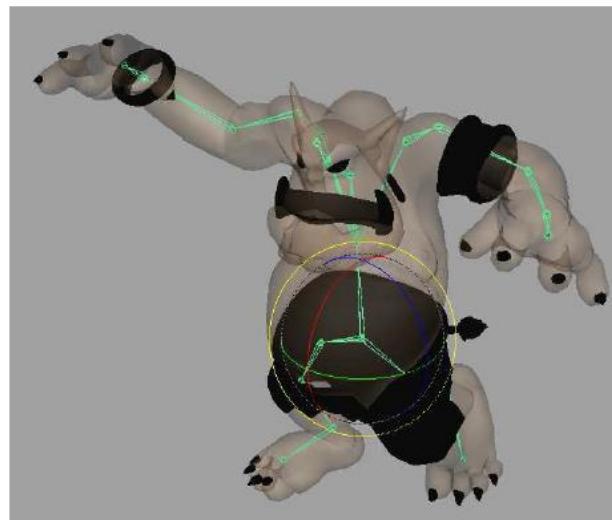


Pipeline 3D

**Squelettage ou rigging
+ calcul des poids de skinning**



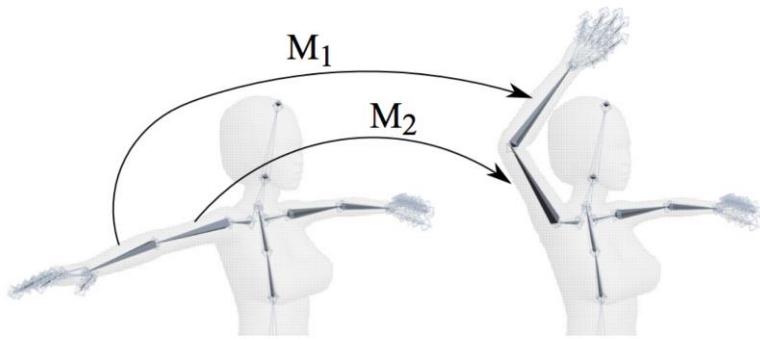
Animation



Calcul des positions
des sommets du maillage



Rigging

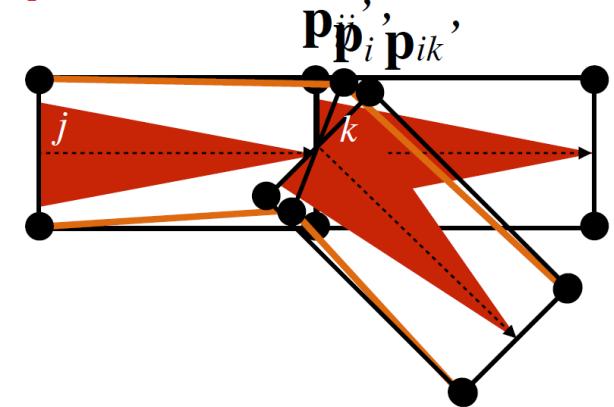


- **Entrées** : sommets p_i de la surface 3D (*skin*) et squelette dans une pose de base (*bind/rest pose*)
 - **Sortie** : positions p_i' après mouvement du squelette
 - On connaît p_i dans le repère monde (ou objet)
 - On connaît la matrice B_j de passage de l'os j au repos au repère monde
- ⇒ nouvelles coordonnées : $p_i' = T_j B_j^{-1} p_i = M_j p_i$
- T_j la matrice de transformation de l'os j

Skinning linéaire

- Sommet influencés par **plusieurs os**
⇒ **interpolation linéaire des positions**
- Si p_i lié aux os j et k :
$$p_i' = w p_{ij}' + (1-w) p_{ik}'$$

 $w \in [0,1] \text{ skin weight}$



Cas général

$$\begin{aligned} p_i' &= \sum_j w_{ij} T_j B_j^{-1} p_i \\ &= \sum_j w_{ij} M_j p_i \end{aligned}$$

avec $\sum_j w_{ij} = 1$

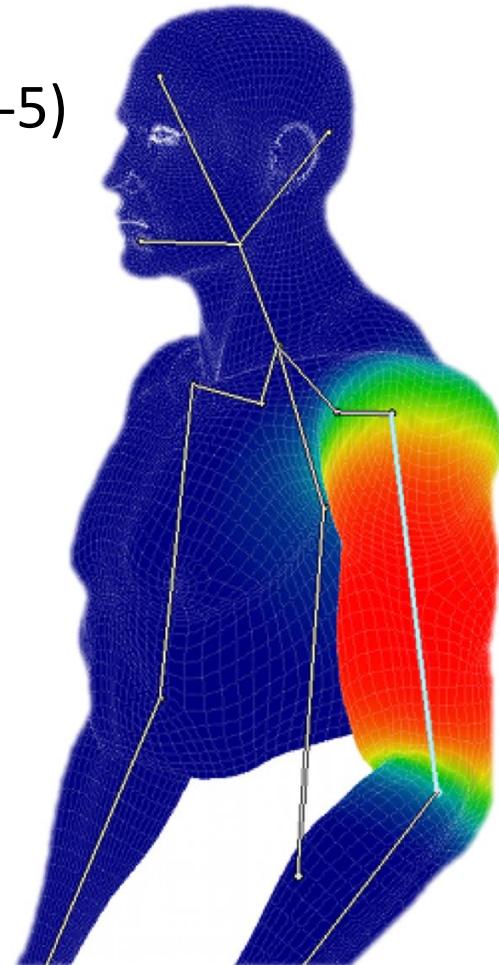
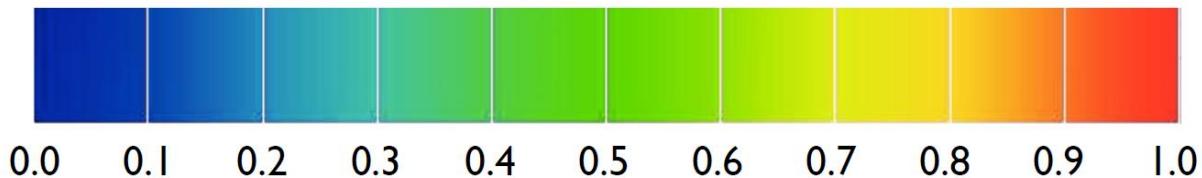
Poids de skinning

Spécifiés (« peints ») par un artiste os par os

- $w_{ij} = 1 \Rightarrow$ mouvement rigide avec l'os
- Limiter le nombre de poids par sommet (4-5)
 - ⇒ coût de la sommation
 - ⇒ stockage mémoire

et/ou calculés automatiquement

(distance euclidienne, harmonique, géodésique)



Skinning linéaire

1. *Rigging* – pré-calcul

⇒ matrices de pose : B_j

⇒ poids de *skinning* : w_{ij}

2. Cinématique direct ou inverse – à chaque temps t

⇒ matrices de transformation $M_j(t) = T_j(t)B_j^{-1}$

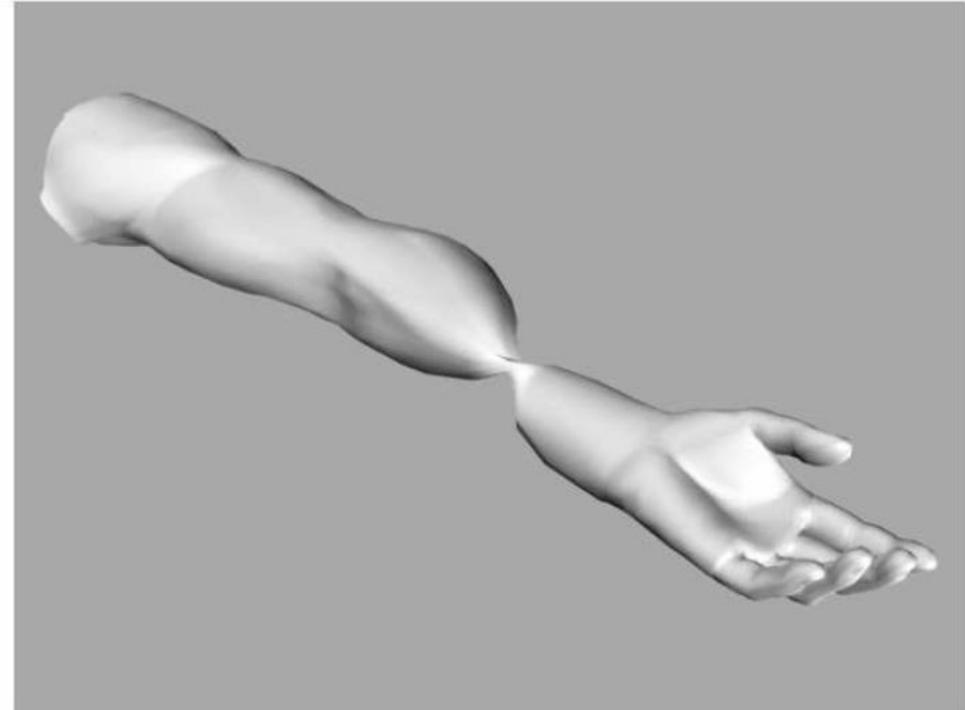
3. Skinning linéaire – *vertex shader*

⇒ $p_i' = \sum_j w_{ij} M_j(t) p_i$

Skinning linéaire

- **Limitation : rotations**

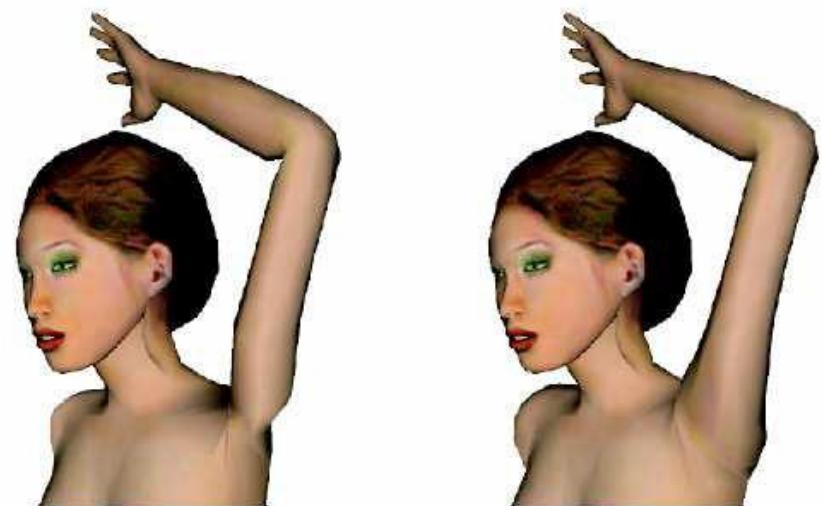
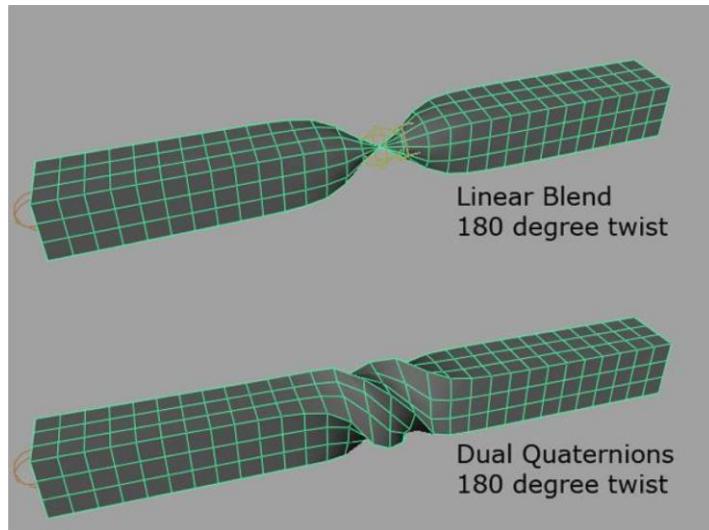
(la moyenne de 2 rotations n'est pas une rotation !)



Skinning + quaternions duaux

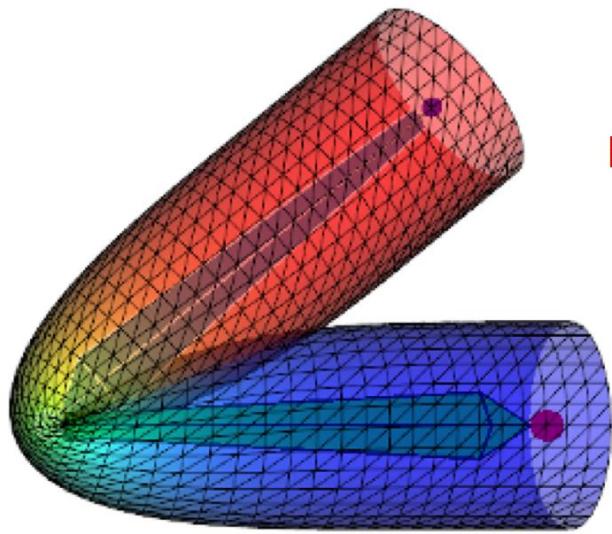
[Kavan et al. 08]

Interpolation des matrices de transformation à l'aide de
quaternions duaux (encodant rotation et translation)



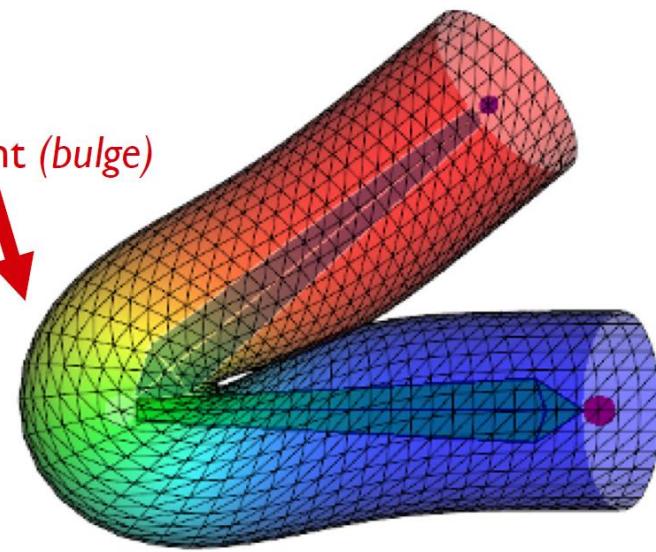
Skinning + quaternions duaux

[Kavan et al. 08]

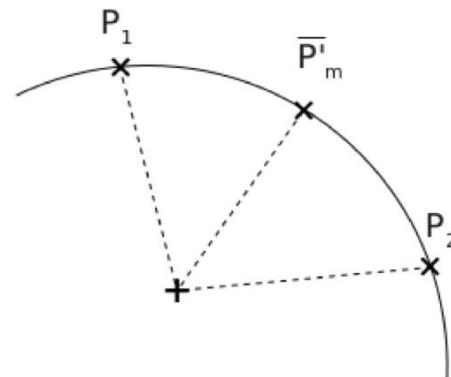
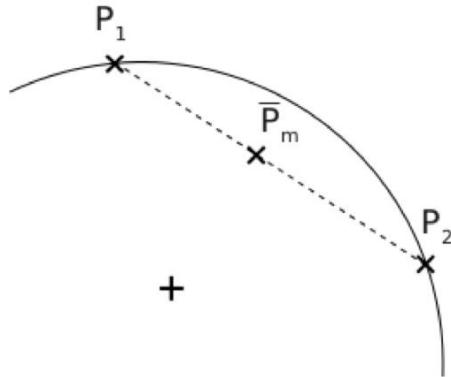


Skinning linéaire

Renflement (*bulge*)



Dual quaternions



Recalcule des normales

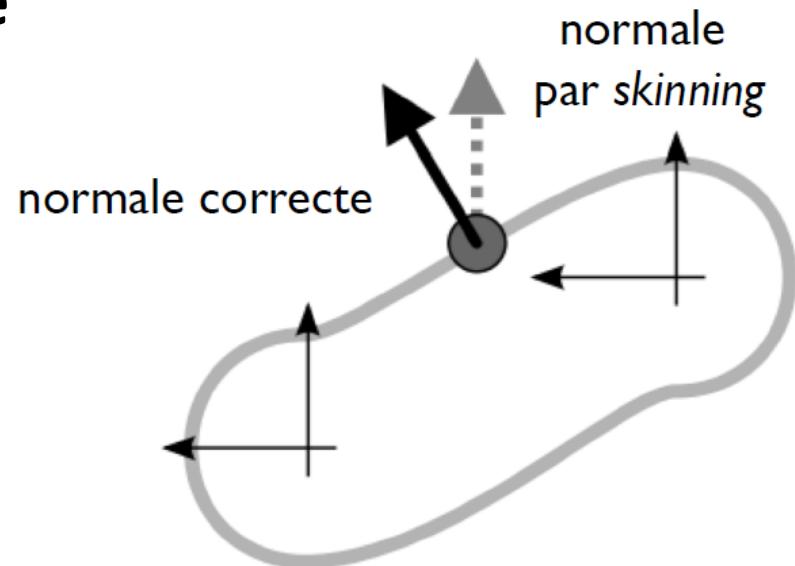
Calculer la transformation linéaire moyenne pour le sommet i : $C_i = \sum_j w_{ij} M_j$

...et transformer sa normale par $(C_i^{-1})^T$

⇒ **approximation grossière**



Pose de base

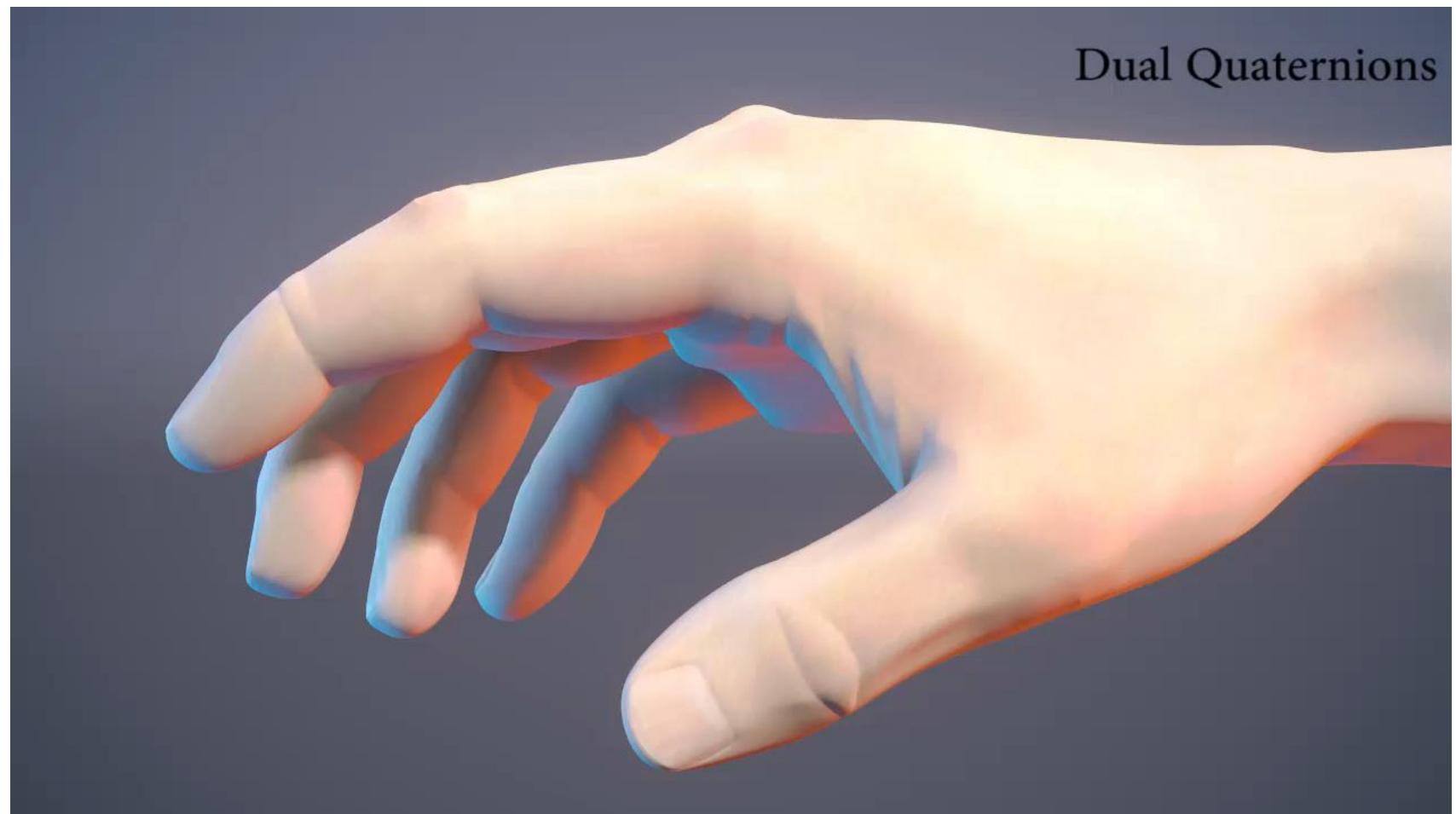


Pose déformée

Skinning + surface implicites

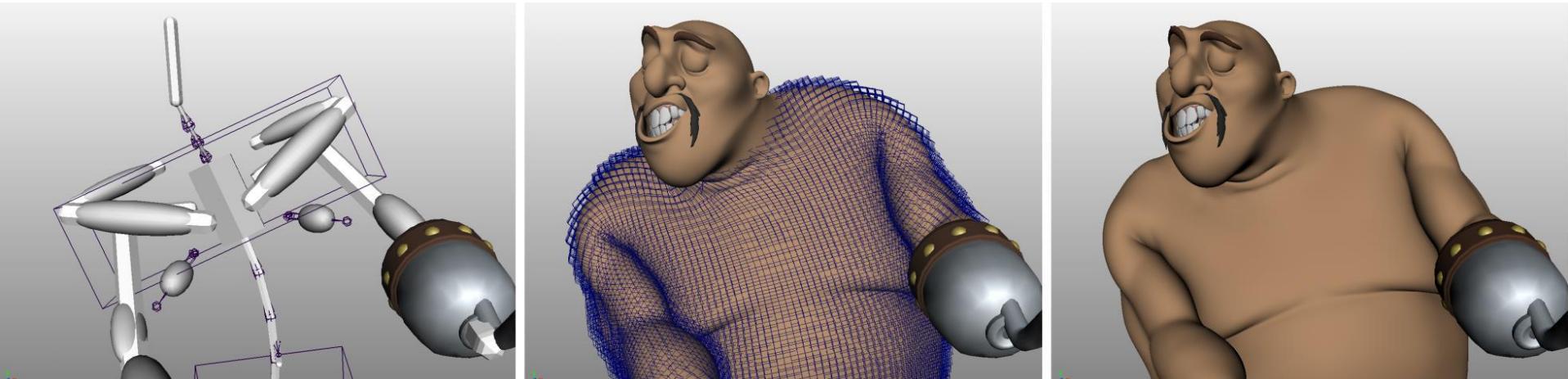
[Vaillant et al. 13 & 14]

Modélisation des **contacts**



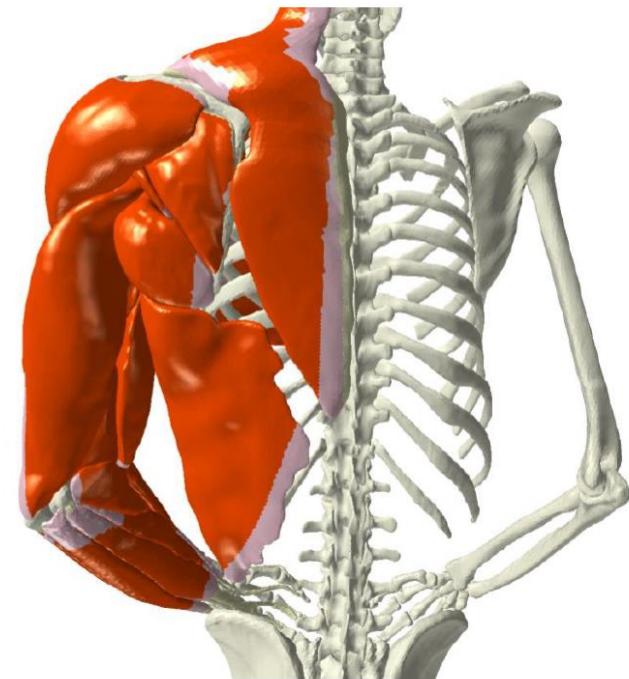
Skinning basé-physique

- Inclure le squelette dans un modèle volumique (e.g. maillage tétraédrique). Simulation de matière élastique (chaire, muscle) guidé par le squelette.



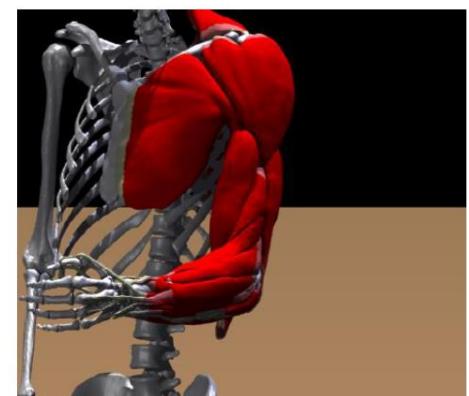
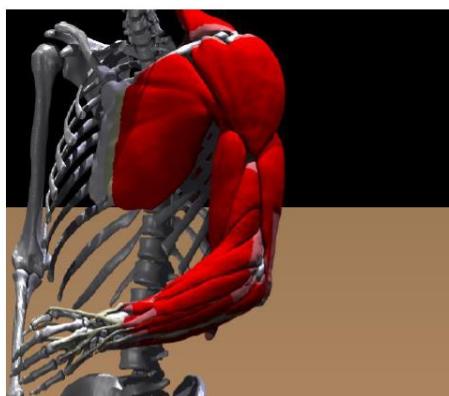
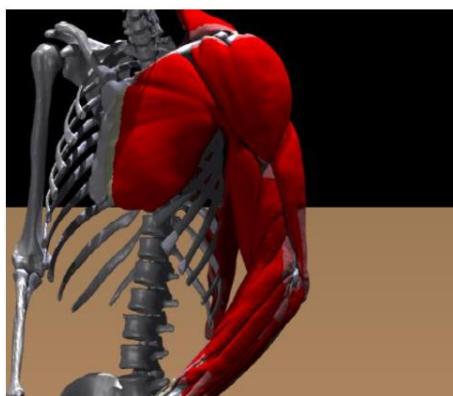
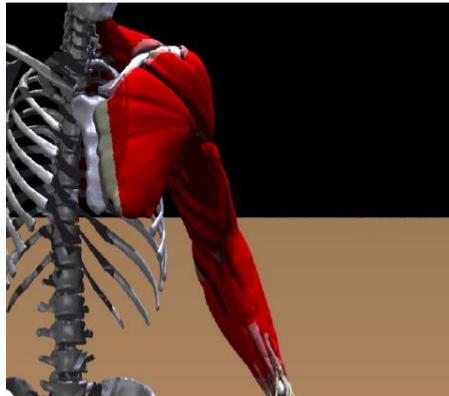
Muscles

- Amélioration du réalisme en ajoutant des muscles se contractant quand ils sont activés et exercent des forces sur les tissus mous (chair)
- Données réelles pour la forme des muscles

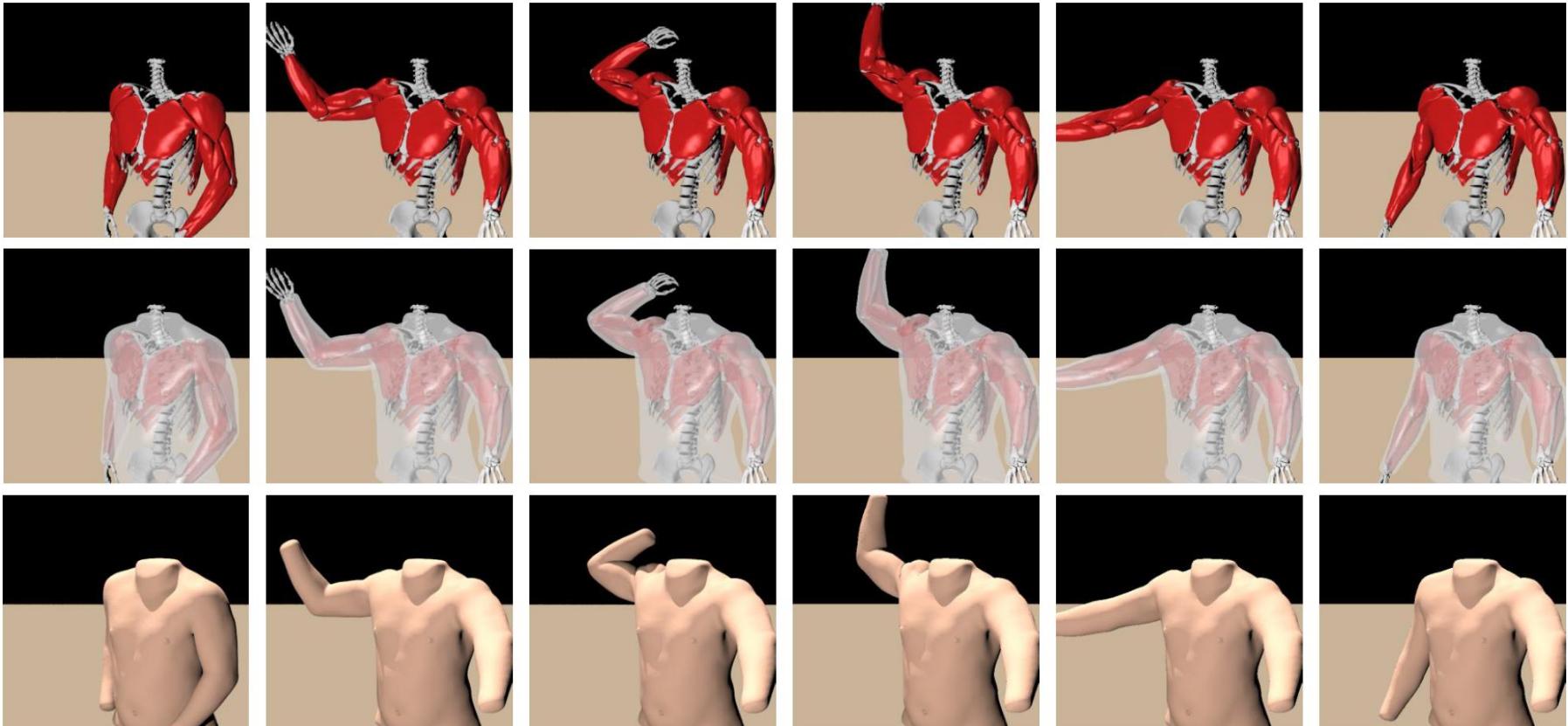


Muscles

Résoudre un problème inverse pour déduire quels muscles doivent s'activés d'après le mouvement des os



Muscles

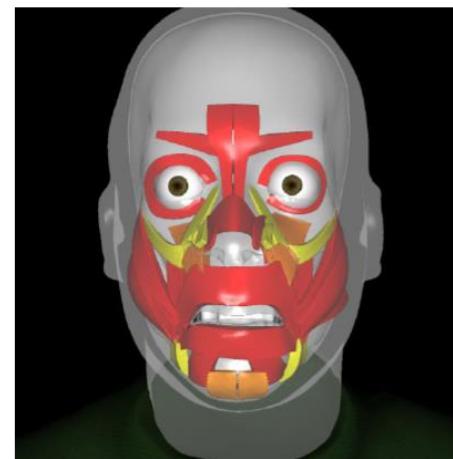
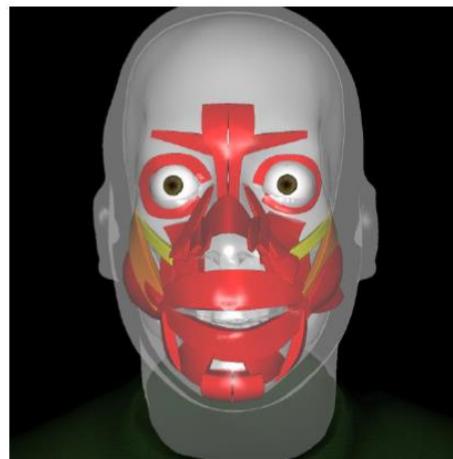


Modèle anatomique du visage

Même principe pour obtenir des expressions réalistes du visage

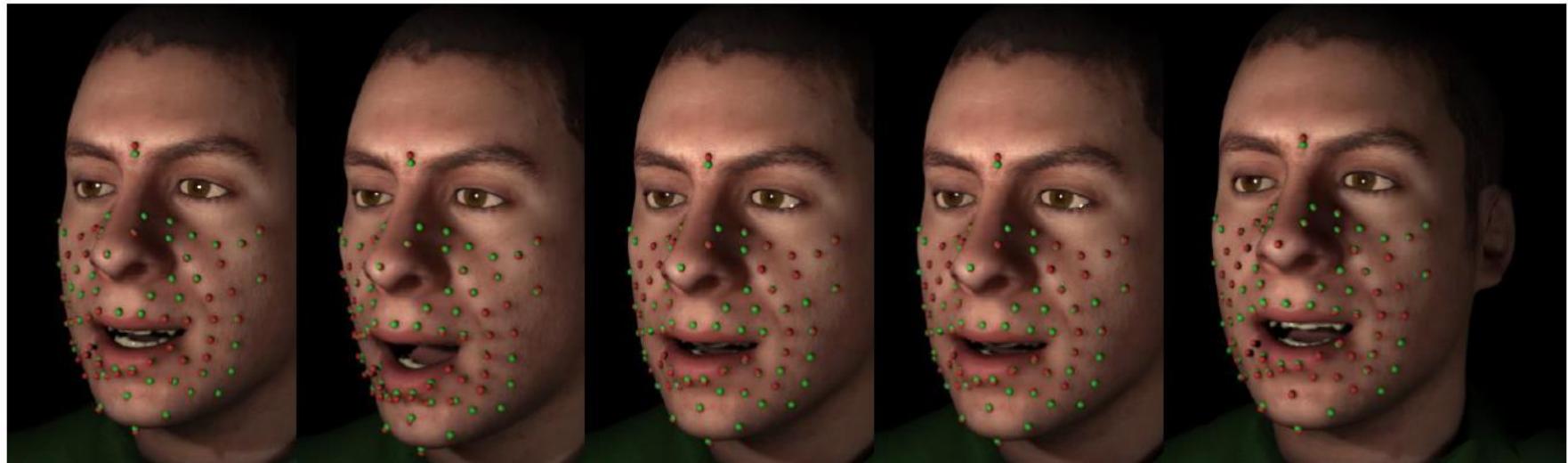
Activation :

- Jaune : complètement
- Rouge : inactif



Activation des muscles

- Utilisation de la capture de mouvement avec des marqueurs à matcher avec des positions du maillage
- Modification des expressions en interpolant les angles et les activations entre les keyframes



Capture de mouvements



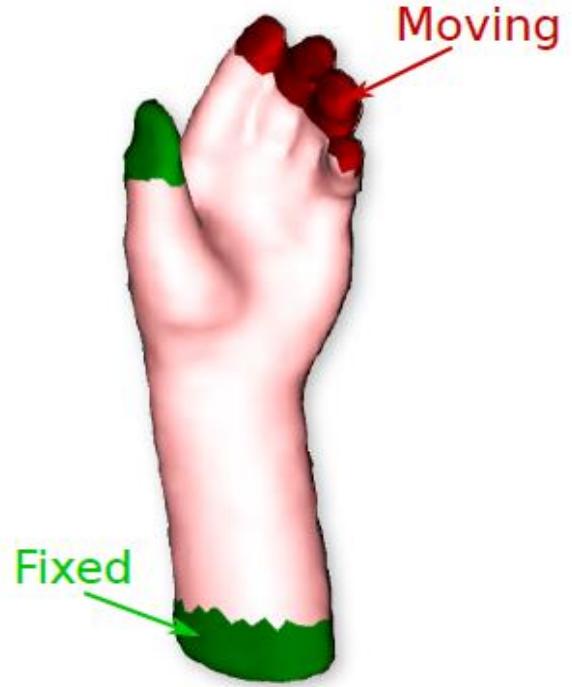
Discovery, “Avatar: Motion Capture Mirrors Emotions”,
<https://youtu.be/1wK1lxr-UmM>

Déformation

- Animation de personnages
 - Sans squelette ou en complément
 - Expressions du visage
 - Changement d'épaisseur
- Modélisation simplifiée : générer de nouvelles formes en déformant une existante
- Poser les personnages pour l'animation

Déformation

- Déformation de surface avec des manipulateurs
 - Linéaires
 - Non-linéaires
- Déformation de l'espace
 - Modèle base resolution guide la deformation de l'espace à l'intérieur de celui-ci
 - Pas de contrainte sur la representation du modèle



Déformation surfacique

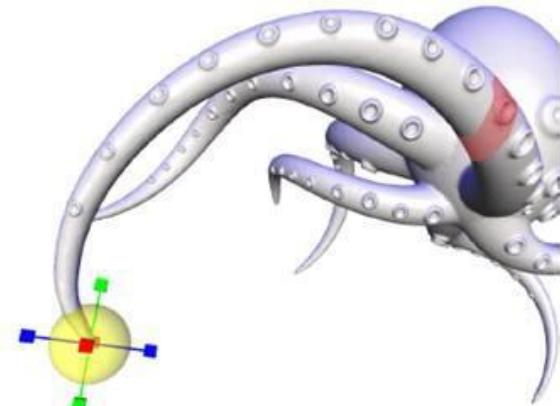
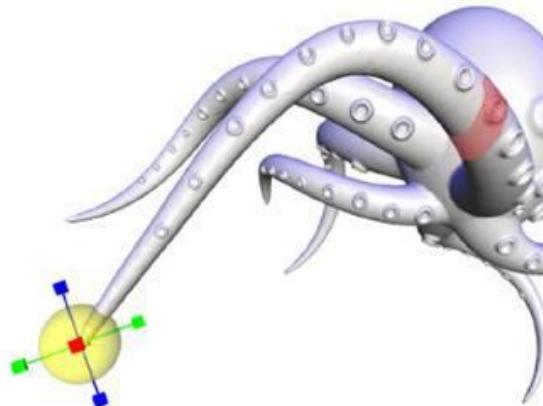
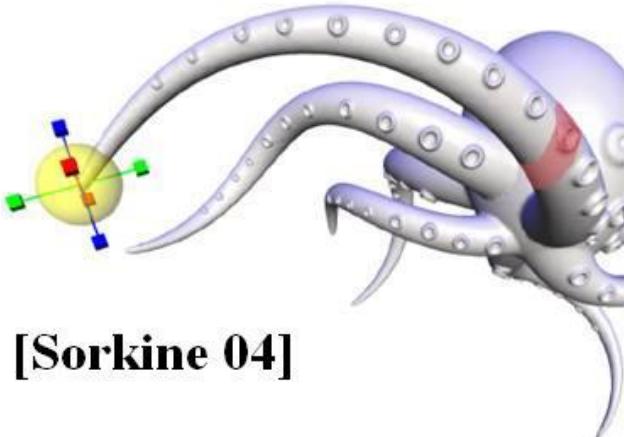
- Déformation de surface
 - Surface déformation
 - Forme est vide
- Courbe pour les déformations 2D
- Surface pour les déformations 3D
- Déformation :
 - définie seulement **sur** la forme
 - couplée avec la representation de la forme

Principes

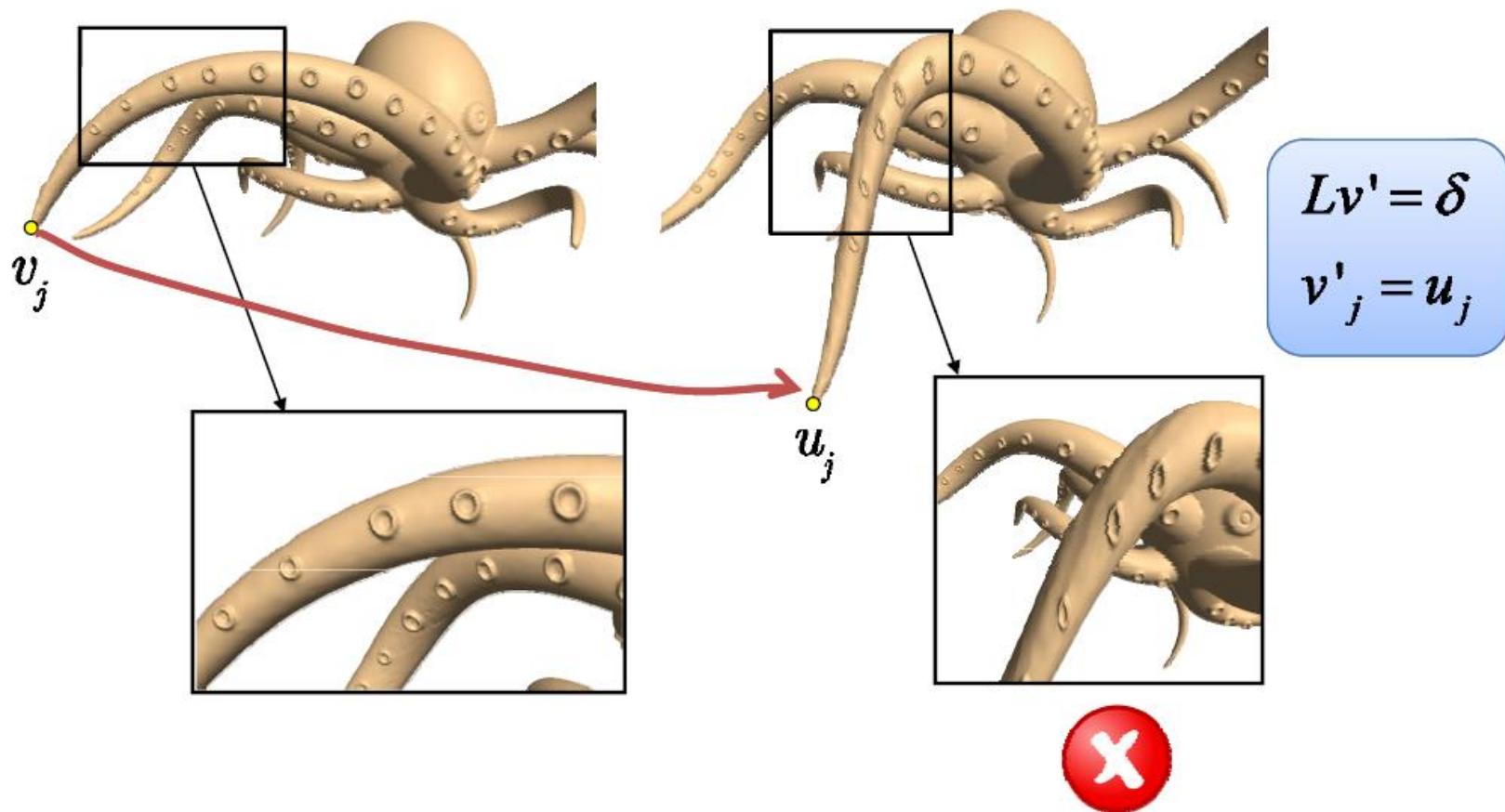
- Donner le moins de contraintes/informations possibles et laisser l'algorithme faire le reste
- Ensemble de sommets déformes localement tout en préservant les détails
- Basée sur la géométrie différentielle discrète

Coordonnées laplaciennes

- 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 : $V' = \operatorname{argmin}_V (\|L'V - D'\|)$

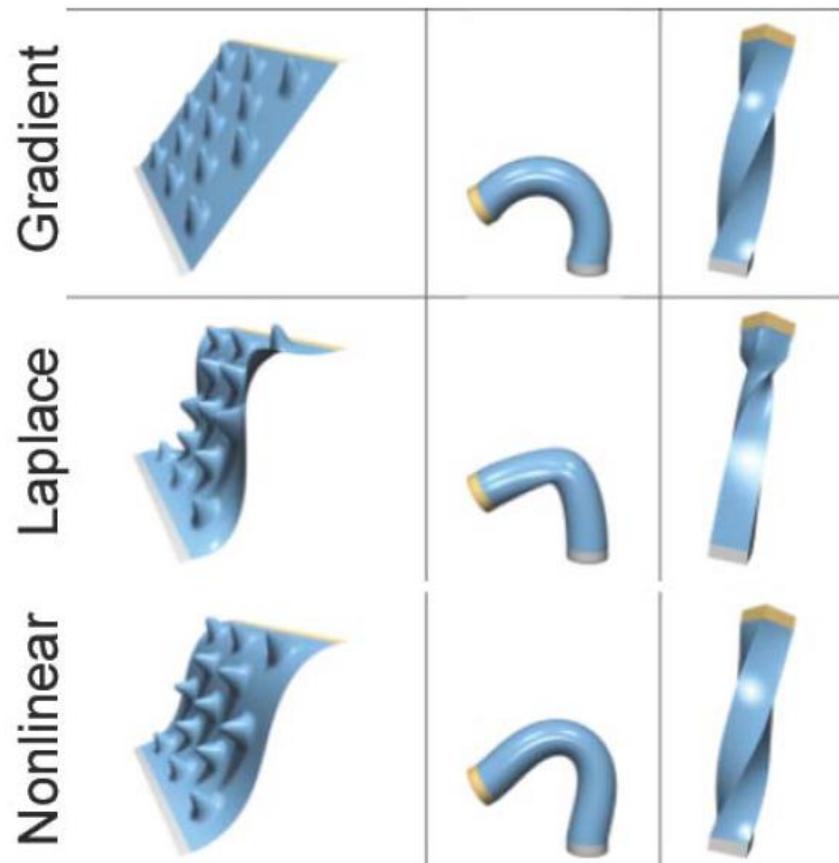


Limitations



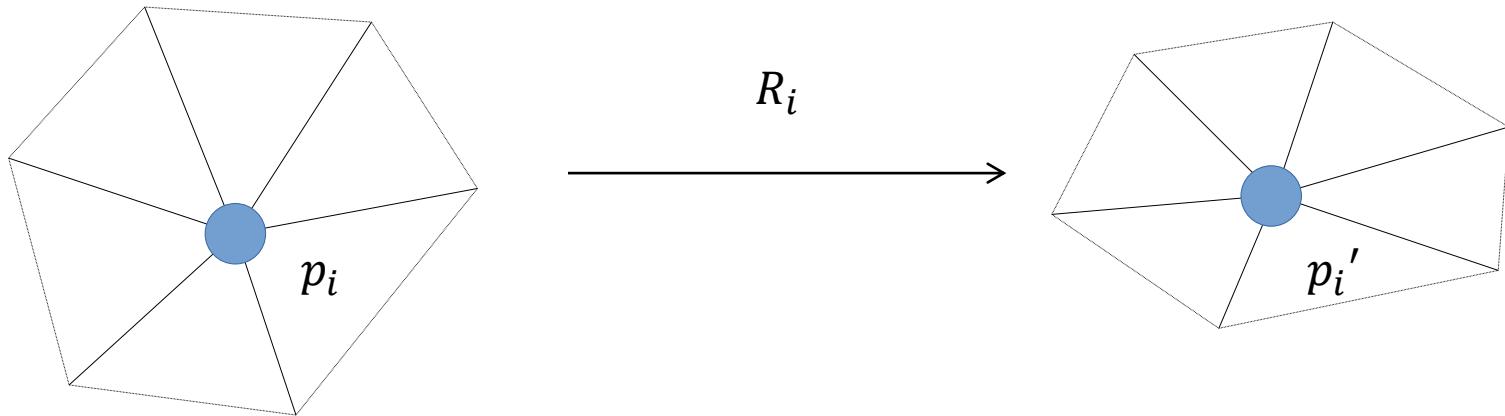
Qualité de déformation

- Méthodes non-linéaires fonctionnent pour les grandes rotations et les translations
- Beaucoup plus long à calculer



Energie rigide

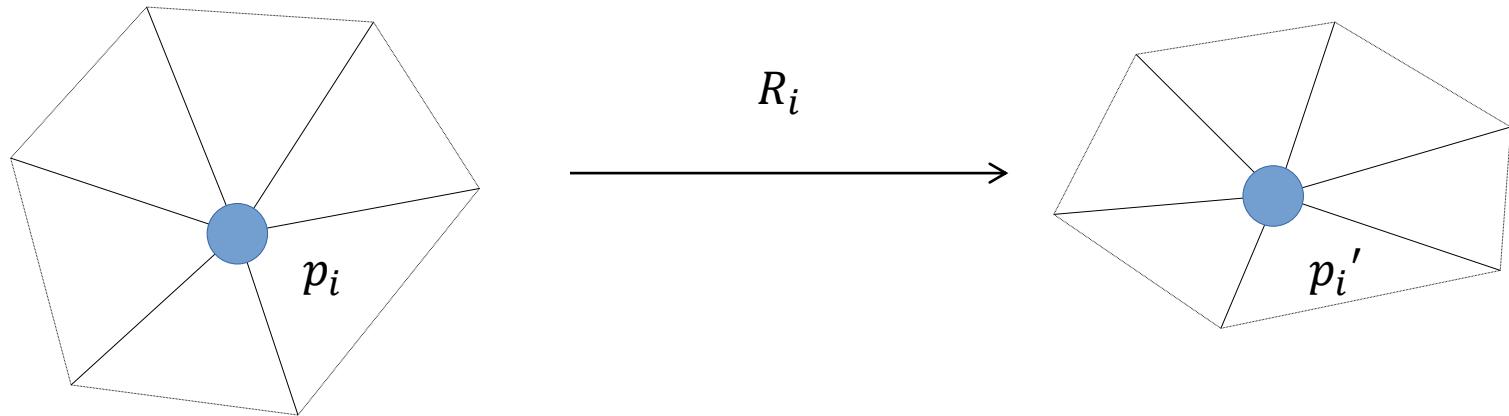
- Déformation ARAP [Sorkine & Alexa] :
 - Aussi-rigide que possible
 - Minimiser le stretch (forçant la rigidité)



$$\mathbf{p}'_i - \mathbf{p}'_j = \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j), \quad \forall j \in \mathcal{N}(i)$$

Energie rigide

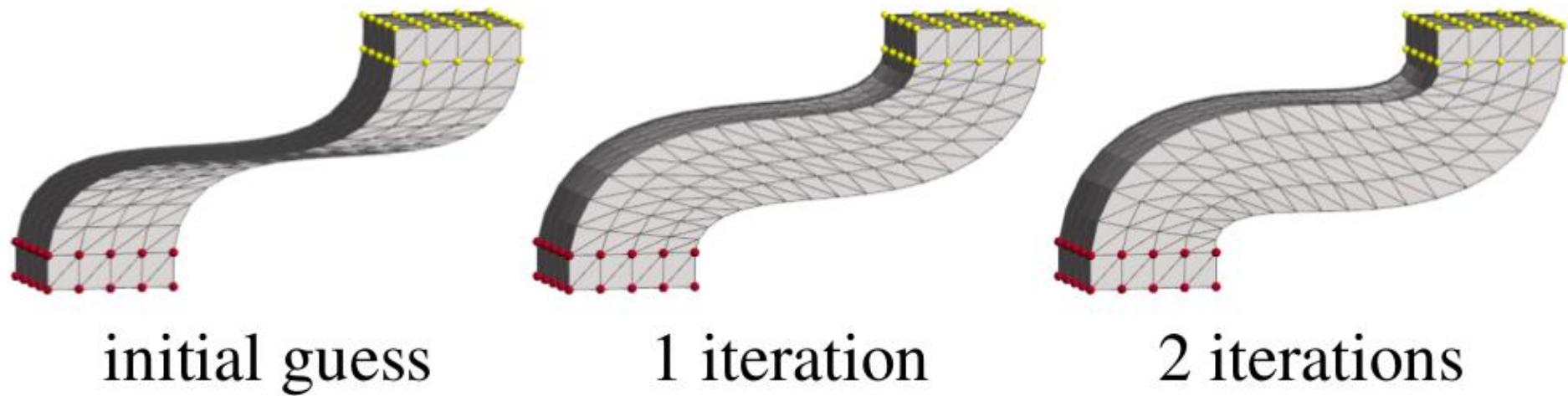
- A minimiser :



$$E(\mathcal{C}_i, \mathcal{C}'_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} \| (\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{R}_i (\mathbf{p}_i - \mathbf{p}_j) \|^2$$

Ad hoc solution

- Fixer R, optimiser p'
- Fixer p', optimiser R
- Fixer R, optimiser p'
- ...



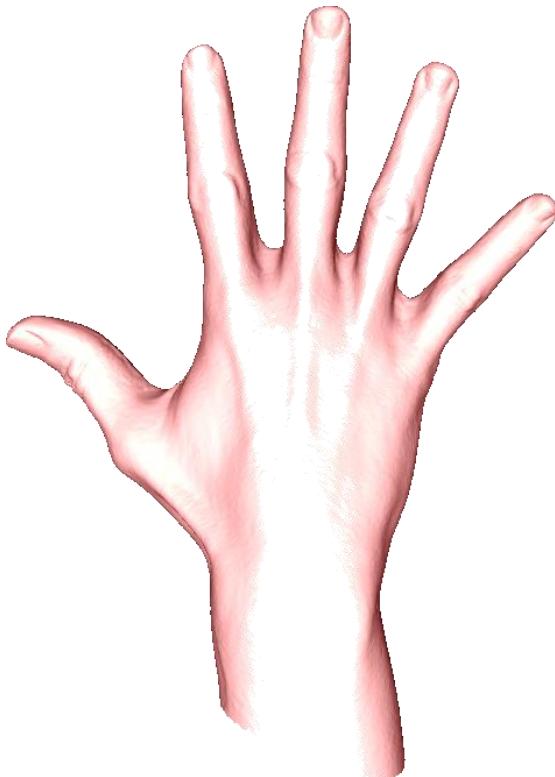
Déformation de l'espace

- Peut-être appliqué à toutes les géométries :
 - maillages (non-variétés, plusieurs composantes)
 - Soupe de polygones
 - Nuages de points
 - Données volumiques
- Complexité découpée de la complexité de la géométrie
 - Choix de la complexité adaptée à la déformation

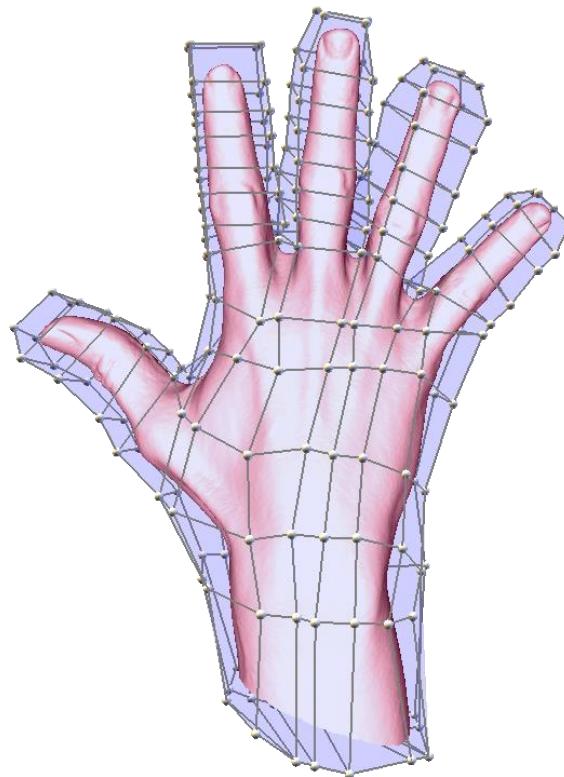
Déformation de l'espace

- Utilisation d'une cage (maillage à géométrie libre) pour contrôler l'animation d'un maillage
- Points de contrôle de déformation = Sommets de la cage \mathbf{p}_j
- $\mathbf{v}_i = \sum_j w_{ij} \mathbf{p}_i$ avec précalcule des w_{ij}
- Comment calculer les influences des sommets de la cage sur le maillage à animer? Quelles coordonnées utiliser?

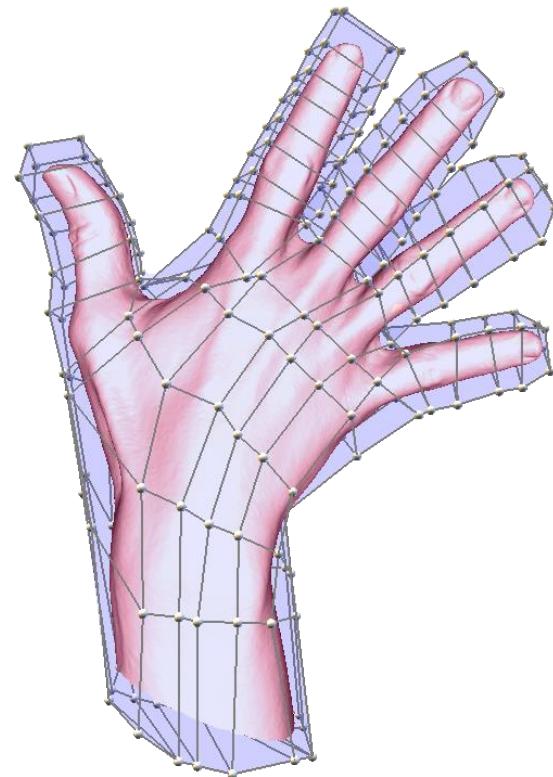
Déformation de l'espace



High resolution
model



Cage coordinates
computation

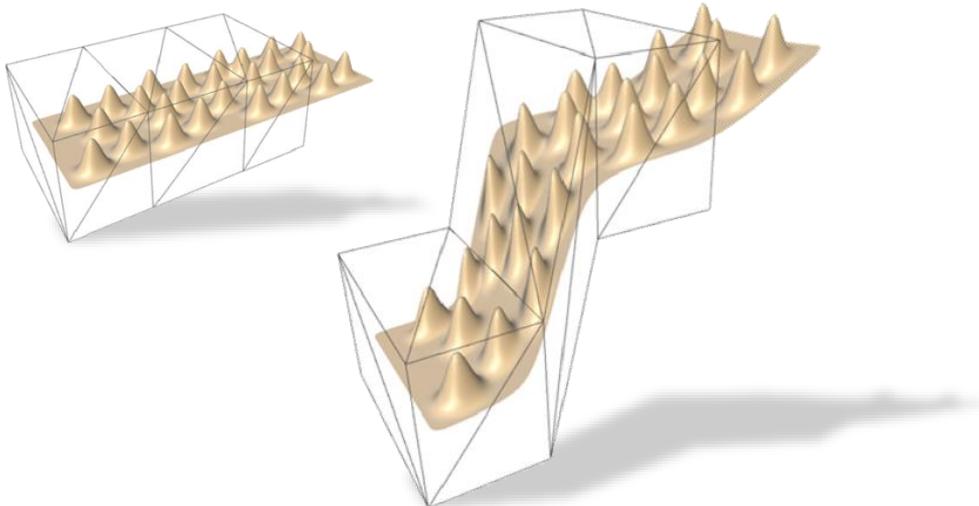


High resolution
model deformation

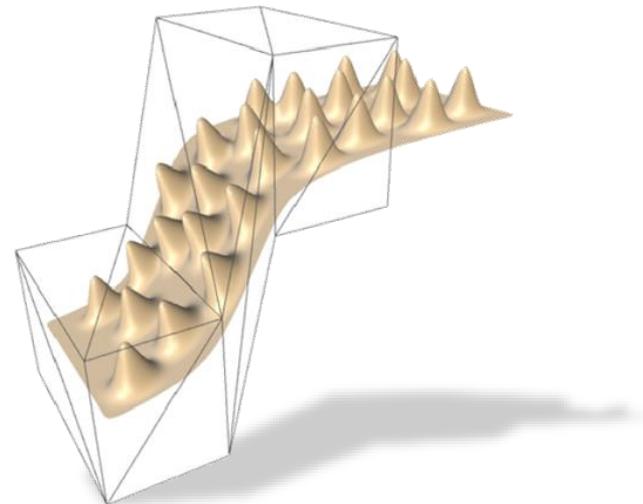
Coordonnées de cage

Green Coordinates [Lipman 2008]

- Déformation quasi conforme de l'espace
- Préservation des caractéristiques



Mean Value Coordinates
MVC [JSW05]



Green Coordinates
GC [LL08]

$$F(\boldsymbol{\eta}; P) = \sum_{i \in I_{\mathbb{V}}} \varphi_i(\boldsymbol{\eta}) \mathbf{v}_i$$

$$F(\boldsymbol{\eta}; P) = \sum_{i \in I_{\mathbb{V}}} \phi_i(\boldsymbol{\eta}) \mathbf{v}_i + \sum_{j \in I_{\mathbb{T}}} \psi_j(\boldsymbol{\eta}) \mathbf{n}(t_j)$$

Crédits

- Pierre Benard, Olivier Vaillancourt, Olivier Godin, Estelle Duveau, Marie-Paule Cani, Lionel Revert, François Faure, Ravi Ramamoorthi, Ronen Barzel, Bill Baxter...

Input

