

Animation de personnages virtuels

Déformation de la peau

Déformation basée squelette - Skinning

Objectif : Déformer un personnage articulé

Idée : Utiliser un squelette pour contrôler les membres

Articulations comme rotations

Squelette d'animation

Ensemble de repères M_i : position, orientation

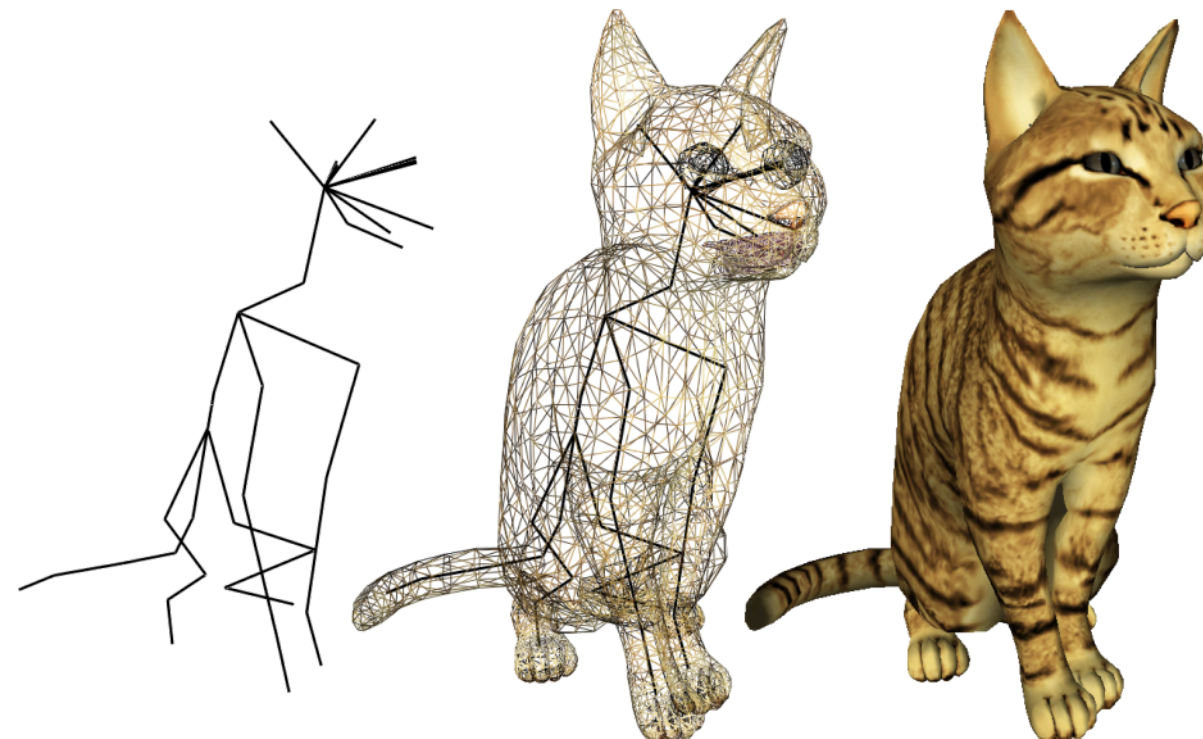
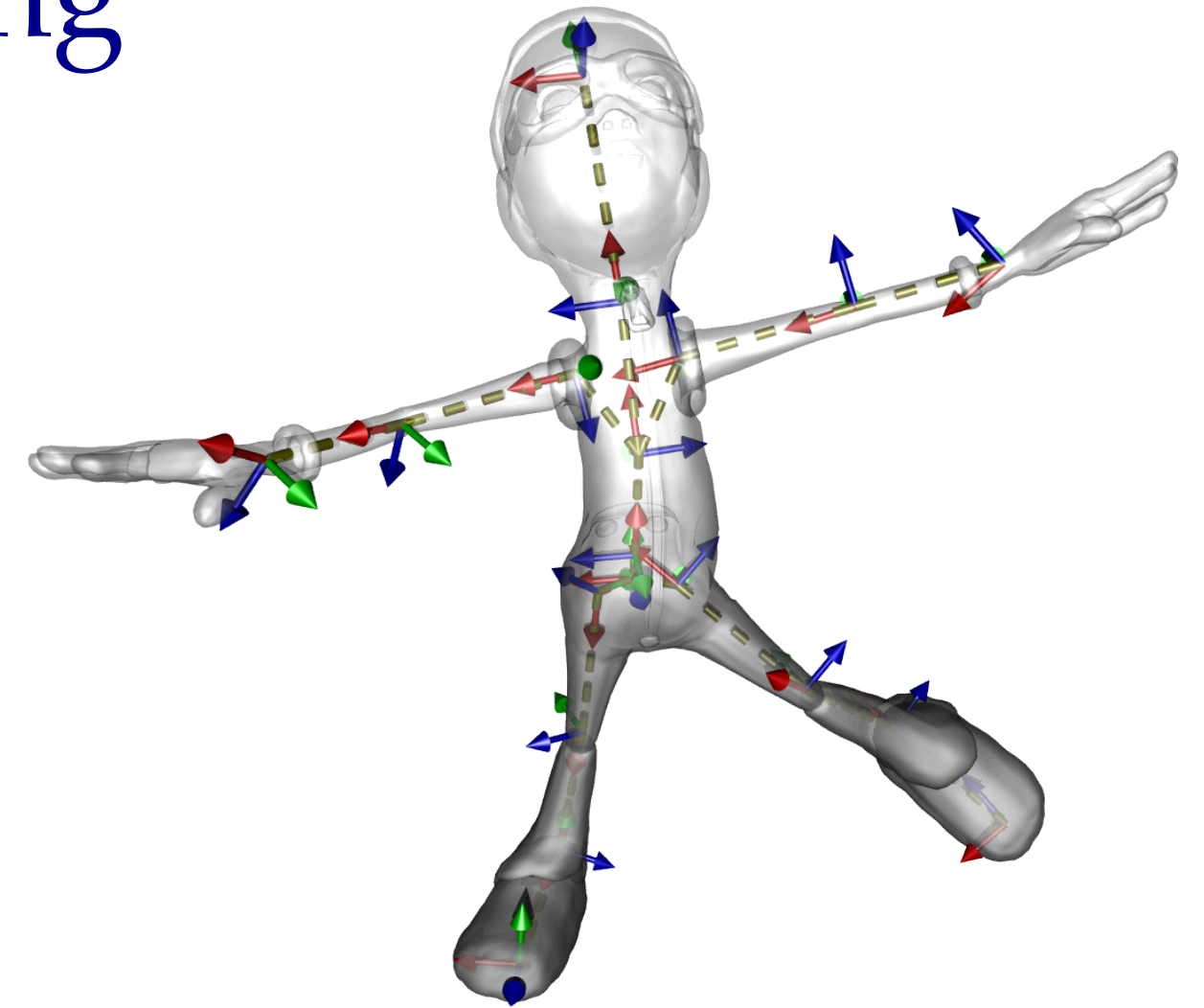
Décrire les parties non rigides du personnage (DDL/DOF)

Terminologie

Articulation = Un repère M_i

Os = Segment entre deux articulations

Note : Squelette d'animation $\neq$ Squelette anatomique



Skinning rigide

Attache rigide d'un sous-ensemble de sommets à des os spécifiques, décrits par le repère de l'articulation.

Les sommets suivent la déformation rigide de leur repère.

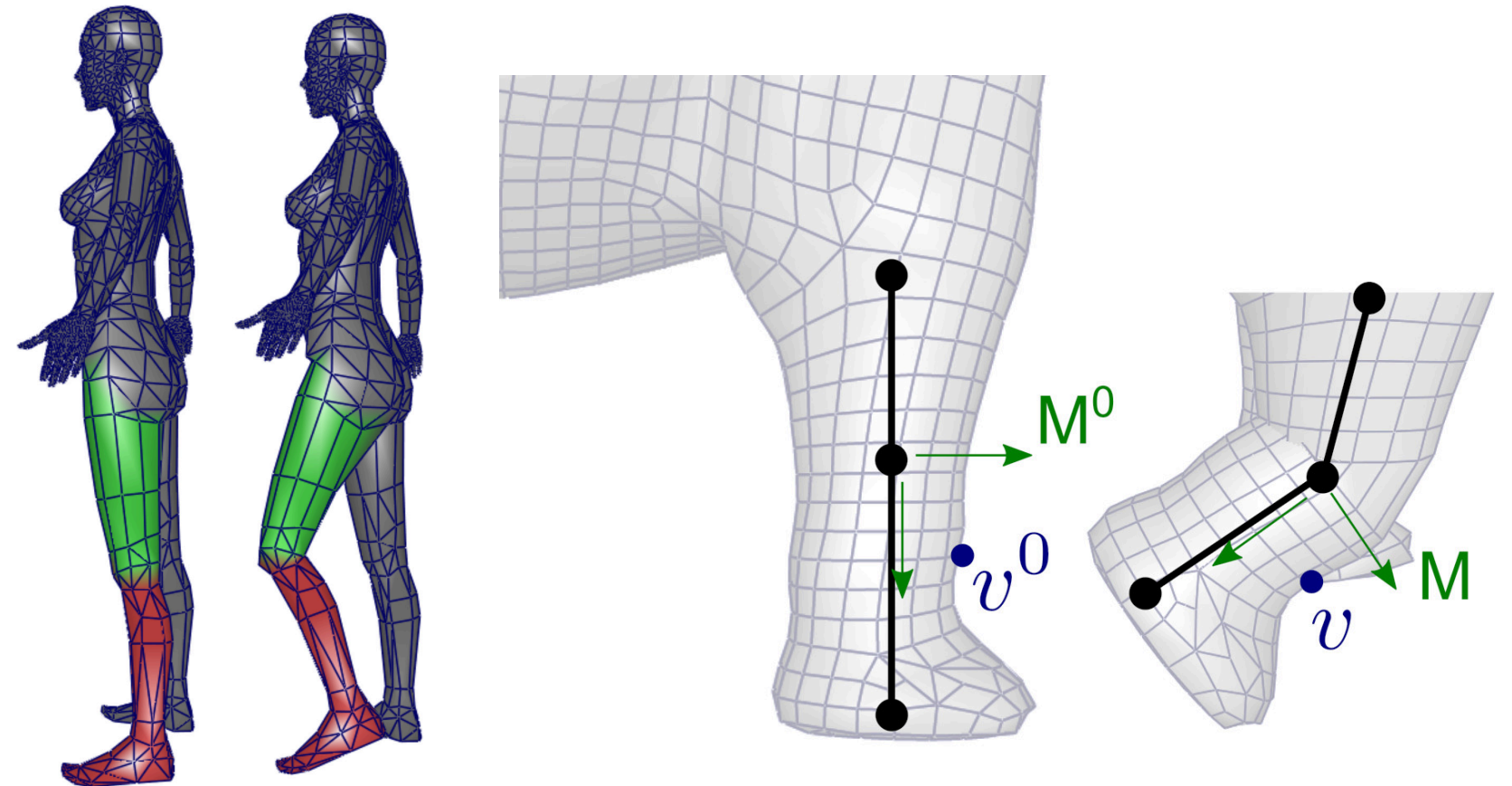
Formulation de la déformation

Considérons, en pose de repos / état initial / pose de liaison

- Un repère M^0
- Un sommet v^0 attaché à ce repère

Après déformation

- Nouveau repère M
- Nouvelle position du sommet v



Question : Quelles sont les nouvelles coordonnées v par rapport à M , M^0 , v^0 ?

Skinning rigide

Attache rigide d'un sous-ensemble de sommets à des os spécifiques, décrits par le repère de l'articulation.

Les sommets suivent la déformation rigide de leur repère.

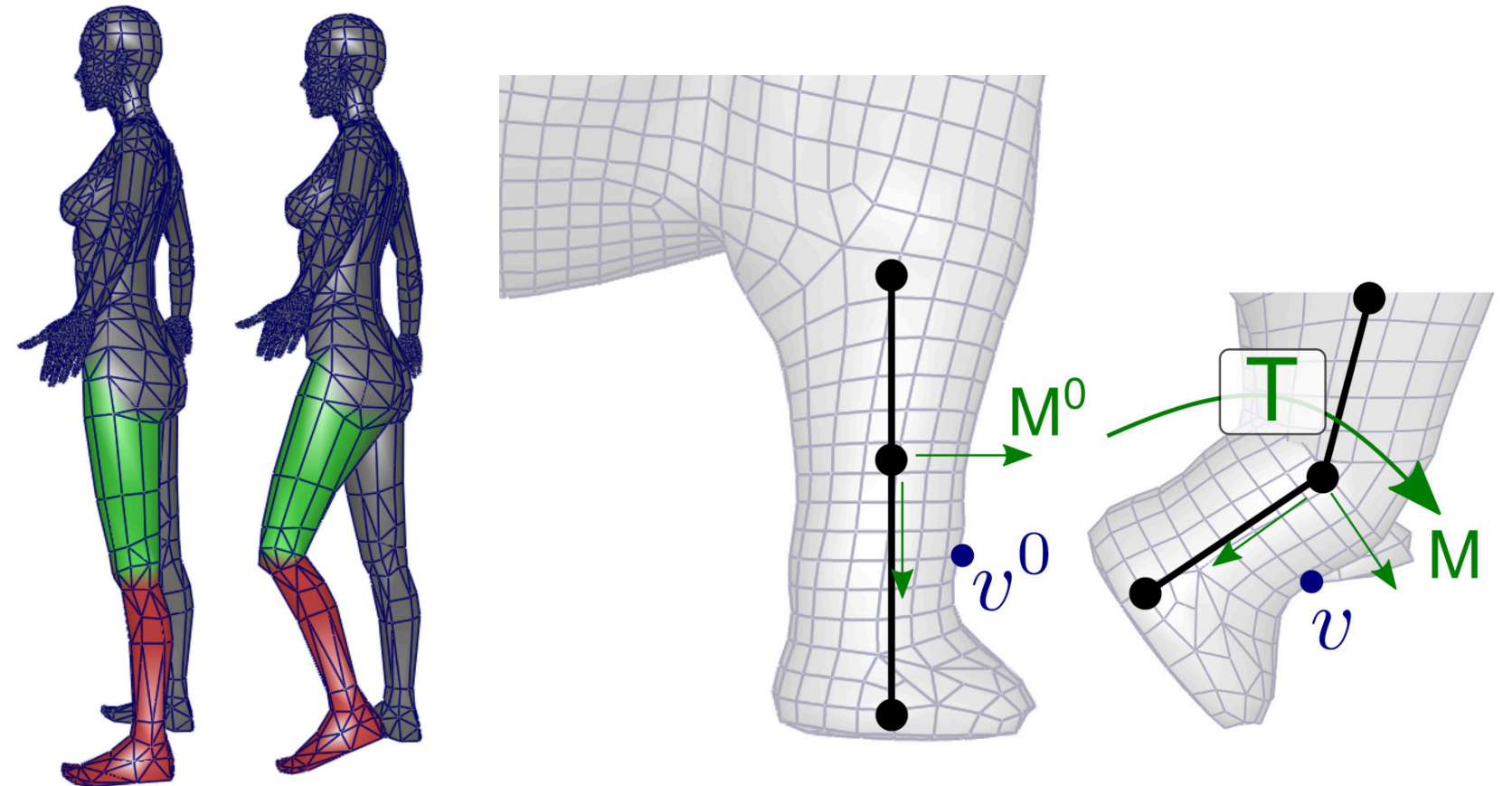
Formulation de la déformation

Considérons, en pose de repos / état initial / pose de liaison

- Un repère M^0
- Un sommet v^0 attaché à ce repère

Après déformation

- Nouveau repère M
- Nouvelle position du sommet v



Question : Quelles sont les nouvelles coordonnées v par rapport à M , M^0 , v^0 ?

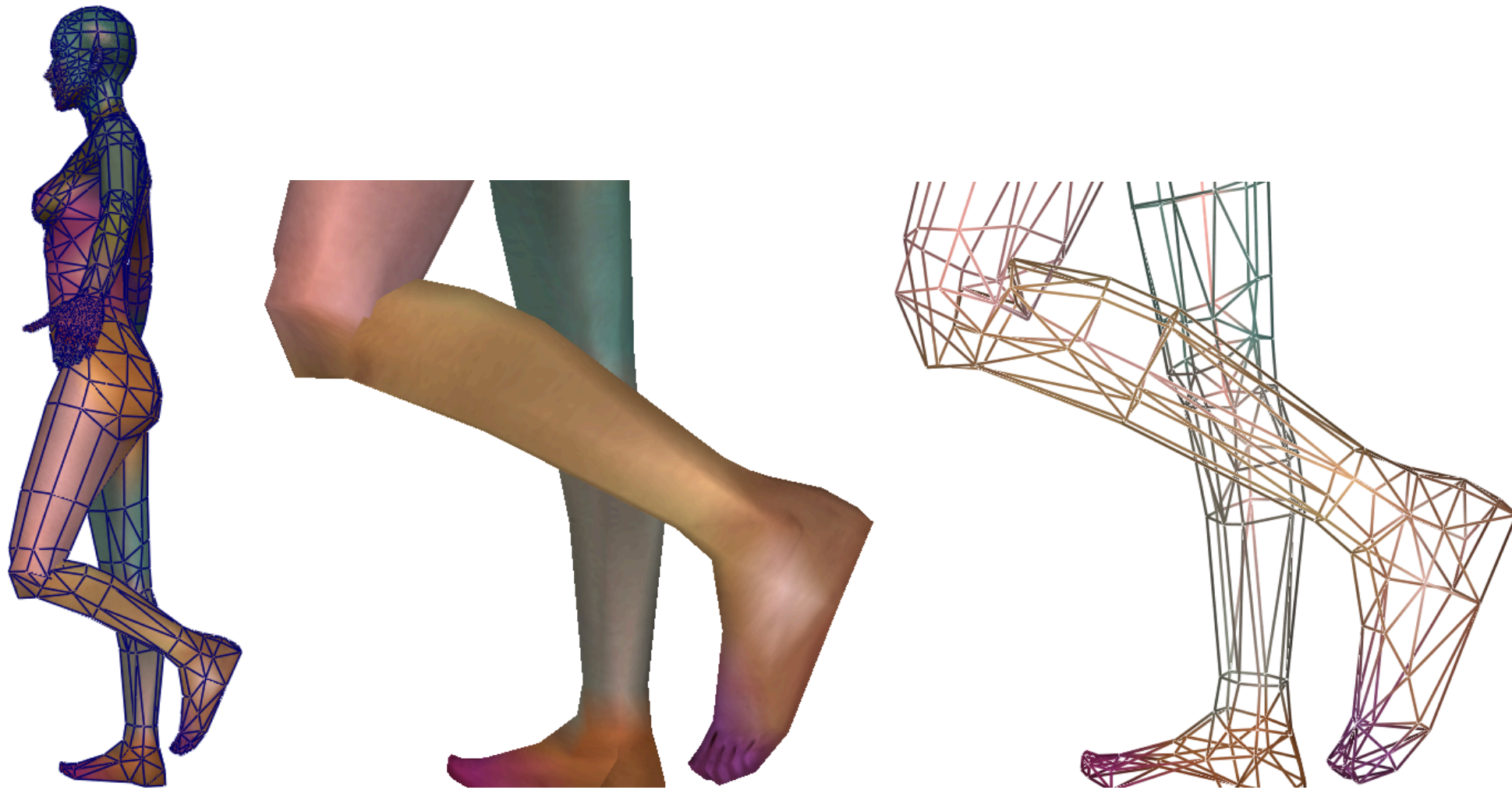
v^0 et v ont des coordonnées locales similaires par rapport à M^0 et M

$$\Rightarrow M^{-1}v = (M^0)^{-1}v_0$$

$$\Rightarrow v = M (M^0)^{-1}v_0 = T v_0$$

Rigid Skinning

- (+) Le squelette est facile à construire
- (+) L'interaction du squelette est intuitive pour modéliser une articulation rigide
- (-) Discontinuités / Interpénétrations



Idée du smooth skinning : Mélanger les transformations discontinues autour de l'articulation

Skinning lisse (/ Smooth skinning)

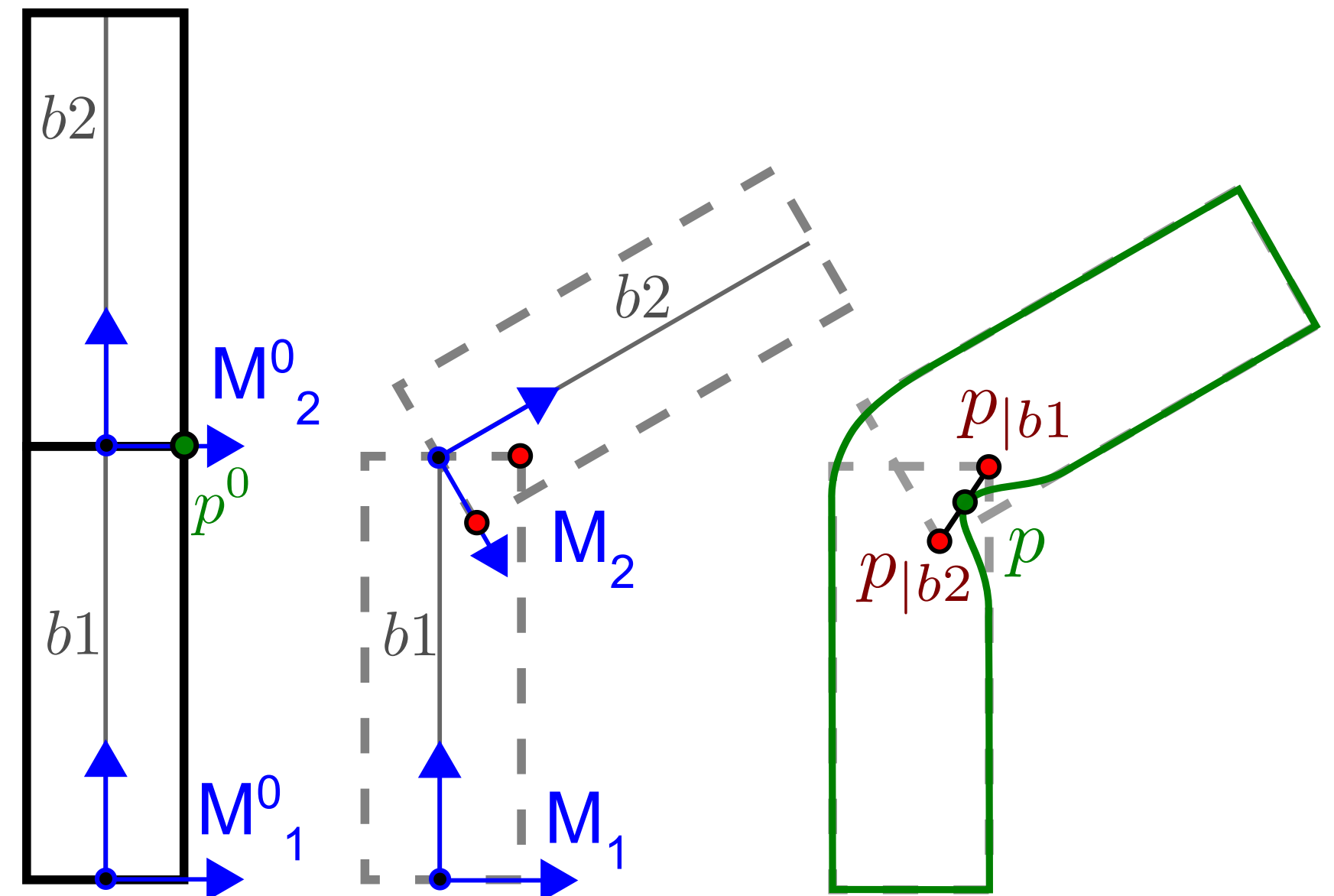
Linear Blend Skinning (LBS) : interpolation linéaire des positions entre les repères associées

Exemple à la position du sommet médian d'un cylindre plié

$$p = 0.5 p_{|b1} + 0.5 p_{|b2}$$

$$p = 0.5 \underbrace{M_1 (M_1^0)^{-1}}_{T_1} p^0 + 0.5 \underbrace{M_2 (M_2^0)^{-1}}_{T_2} p^0$$

$$p = (0.5 T_1 + 0.5 T_2) p^0$$



Skinning lisse (/ Smooth skinning)

Linear Blend Skinning (LBS) : interpolation linéaire des positions entre les repères associées

Exemple à la position du sommet médian d'un cylindre plié

$$p = 0.5 p_{|b1} + 0.5 p_{|b2}$$

$$p = 0.5 \underbrace{M_1 (M_1^0)^{-1}}_{T_1} p^0 + 0.5 \underbrace{M_2 (M_2^0)^{-1}}_{T_2} p^0$$

$$p = (0.5 T_1 + 0.5 T_2) p^0$$

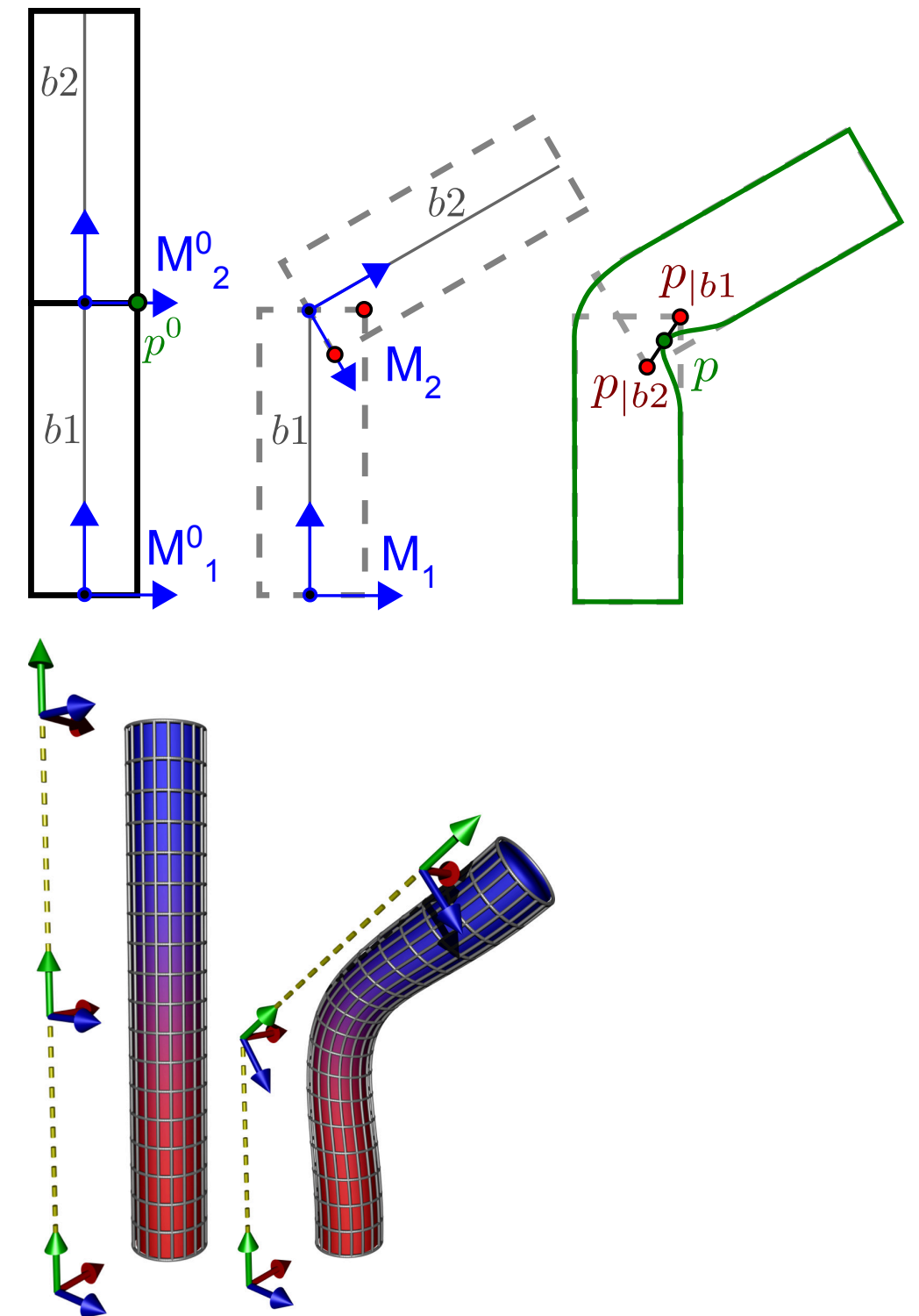
Peut être généralisé à une interpolation arbitraire entre deux os

$$p = \alpha p_{|b1} + (1 - \alpha) p_{|b2} = (\alpha T_1 + (1 - \alpha) T_2) p^0$$

α : poids de skinning

Peut être généralisé à un nombre quelconque d'os

$$p = \sum_{j=0}^{N-1} \alpha_j p_{|bj} = \left(\sum_{j=0}^{N-1} \alpha_j T_j \right) p^0, \quad \sum_j \alpha_j = 1$$



Skinning lisse - Résumé

$$p_i = \left(\sum_{j=0}^{N-1} \alpha_{ij} T_j \right) p_i^0 = \left(\sum_{j=0}^{N-1} \alpha_{ij} M_j (M_j^0)^{-1} \right) p_i^0$$

$\forall j, \alpha_{ij} \in [0, 1]$, et $\sum_{j=0}^{N-1} \alpha_{ij} = 1$

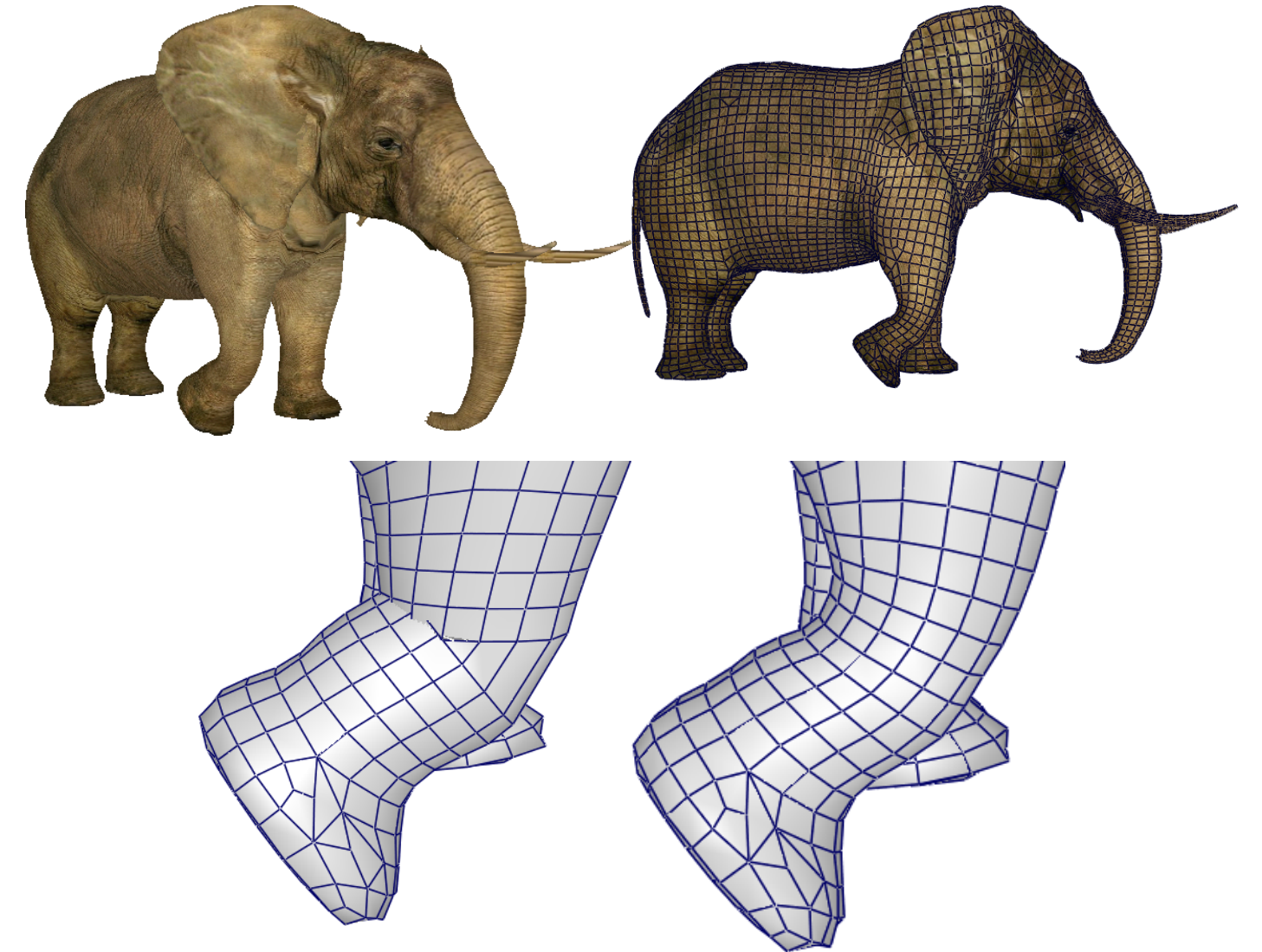
M_j^0 : Bind Pose

La **norme** actuelle pour presque toutes les déformations de personnages articulés

- Déformation intuitive
- Forme contrôlable (*via les poids*)
- Rapide à calculer (compatible GPU)

moyenne de matrices, multiplication matrice-vecteur

⇒ Très utilisée en animation cinéma & jeux vidéo



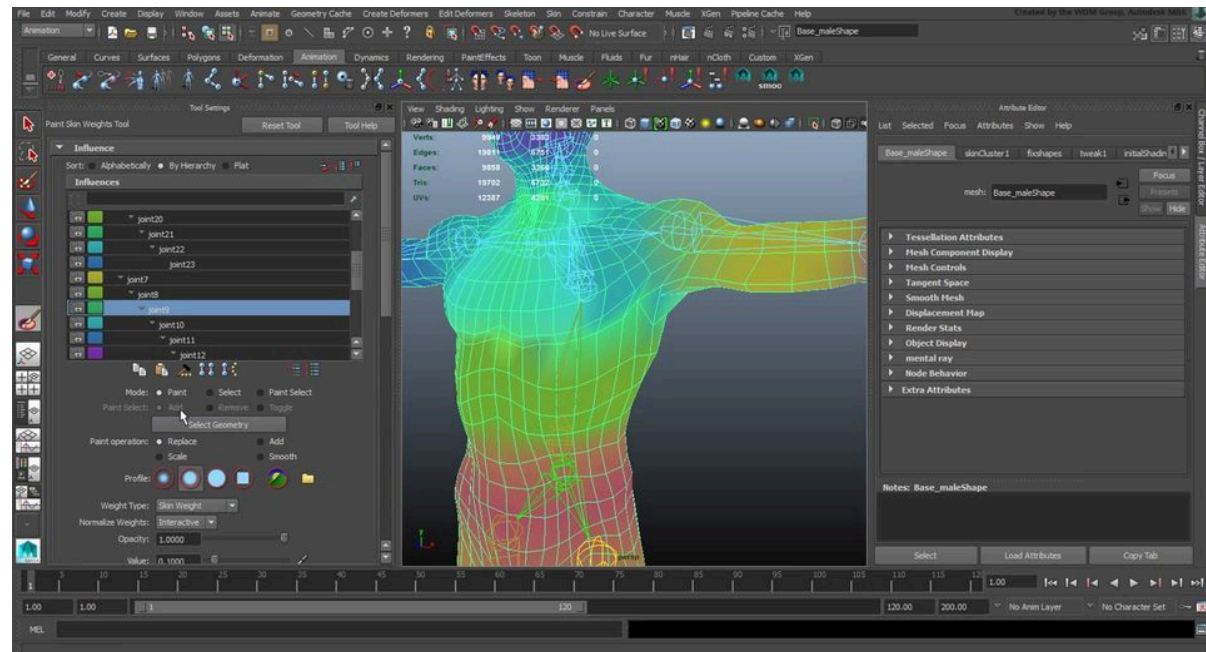
[Joint-Dependent Local Deformations for Hand Animation and Object Grasping. Nadia Magnenat-Thalmann, Rochard Laperrière, Daniel Thalmann. Graphics Interface, 1988]

[Over My Dead, Polygonal Body. Jeff Lander. Game Developer Magazine, 1998]

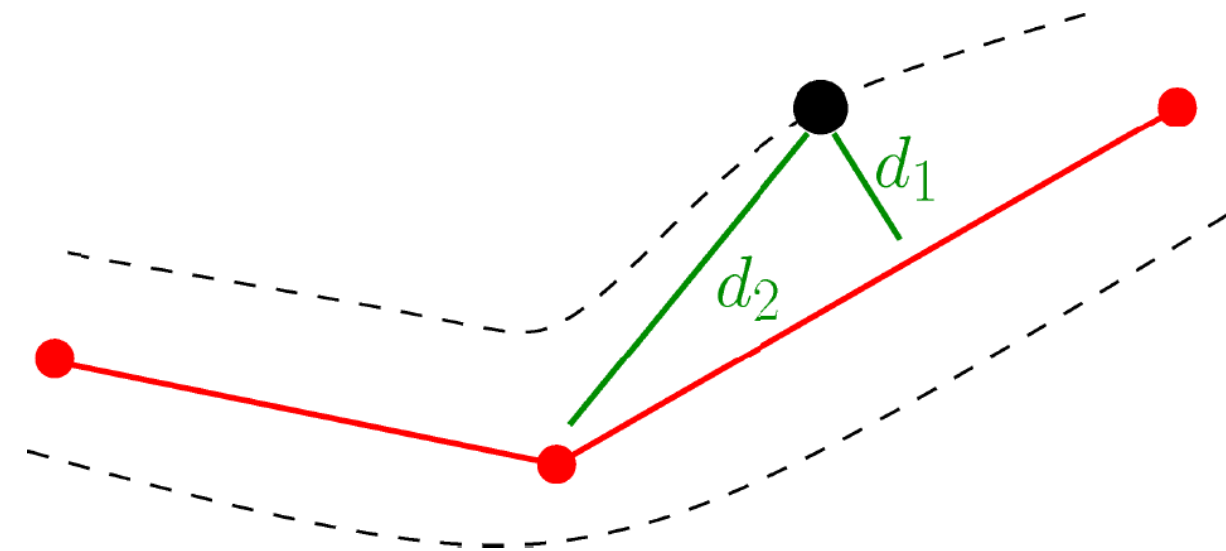
Poids de skinning

Comment générer les poids de skinning ?

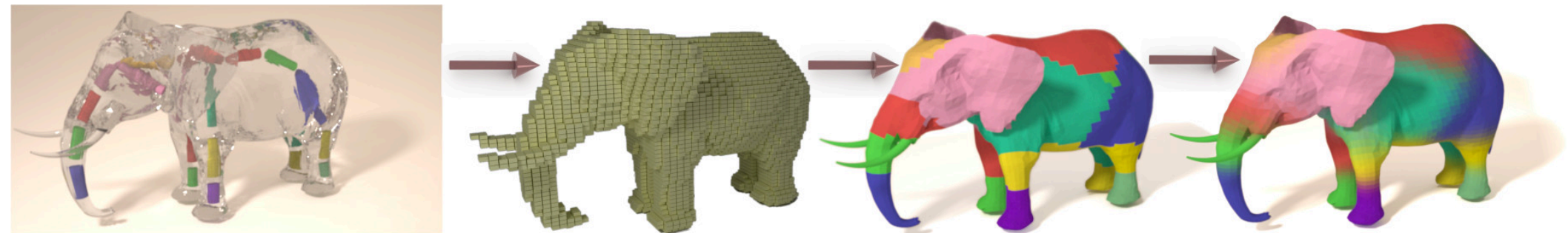
- Les peindre manuellement
- Calcul automatique



ex. Utilisation des distances cartésiennes : $\alpha_1 = d_1^{-1} / (d_1^{-1} + d_2^{-1})$



Ou en utilisant la diffusion sur le volume/surface



Rigging : Association des os et des poids de skinning (ou de toute poignée d'animation) aux parties du maillage

Skinning : Format de fichier

Malheureusement, peu de formats ouverts standard pour stocker les données d'animation de skinning

Principaux formats ouverts : Collada (XML), glTF (JSON)

Formats liés aux logiciels courants : FBX, Blend, 3DS, ...

```
krissie Maya 7.0 | ColladaMaya v2.04 | FCollada v1.14 Collada Maya Export Options:
bakeTransforms=0;exportPolygonMeshes=1;bakeLighting=0;isSampling=0;
curveConstrainSampling=0;exportCameraAsLookat=0; exportLights=0;exportCameras=1;exportJointsAndSkin=1;
exportAnimations=1;exportTriangles=0;exportInvisibleNodes=0;
exportNormals=1;exportTexCoords=1;exportVertexColors=0;exportTangents=0;
exportTexTangents=0;exportConstraints=0;exportPhysics=0;exportXRefs=0;
dereferenceXRefs=0;cameraXFov=0;cameraYFov=1 file:///E:/maya_projets/girafe_skin/scenes/giranitex_33_bak3.ma 2008-
04-16T21:56:42Z 2008-04-16T21:56:42Z Y_UP 0.040000 0.080000 0.120000 0.160000 0.200000 0.240000 0.280000
0.320000 0.360000 0.400000 0.440000 0.480000 0.520000 0.560000 0.600000 0.640000 0.680000 0.720000 0.760000
0.800000 0.840000 0.880000 0.920000 0.960000 1.000000 1.040000 1.080000 1.120000 1.160000 1.200000 1.240000
1.280000 1.320000 1.360000 1.400000 1.440000 1.480000 1.520000 1.560000 1.600000 1.640000 1.680000 1.720000
1.760000 1.800000 1.840000 1.880000 1.920000 1.960000 2.000000 2.040000 2.080000 2.120000 2.160000 2.200000
2.240000 2.280000 2.320000 2.360000 2.400000 2.440000 2.480000 2.520000 2.560000 2.600000 2.640000 2.680000
2.720000 2.760000 2.800000 2.840000 2.880000 2.920000 2.960000 3.000000 3.040000 3.080000 3.120000 3.160000
3.200000 3.240000 3.280000 3.320000 3.360000 3.400000 3.440000 3.480000 3.520000 3.560000 3.600000 3.640000
3.680000 3.720000 3.760000 3.800000 3.840000 3.880000 3.920000 3.960000 4.000000 4.040000 4.080000 4.120000
4.160000 4.200000 4.240000 4.280000 4.320000 4.360000 4.400000 4.440000 4.480000 4.520000 4.560000 4.600000
4.640000 4.680000 4.720000 4.760000 4.800000 4.840000 4.880000 4.920000 4.960000 5.000000 5.040000 5.080000
5.120000 5.160000 5.200000 5.240000 5.280000 5.320000 5.360000 5.400000 5.440000 5.480000 5.520000 5.560000
5.600000 5.640000 5.680000 5.720000 5.760000 5.800000 5.840000 5.880000 5.920000 5.960000 6.000000 6.040000
6.080000 6.120000 6.160000 6.200000 6.240000 6.280000 6.320000 6.360000 6.400000 6.440000 6.480000 6.520000
6.560000 6.600000 6.640000 6.680000 6.720000 6.760000 6.800000 6.840000 6.880000 6.920000 6.960000 7.000000
7.040000 7.080000 7.120000 7.160000 7.200000 7.240000 7.280000 7.320000 7.360000 7.400000 7.440000 7.480000
7.520000 7.560000 7.600000 7.640000 7.680000 7.720000 7.760000 7.800000 7.840000 7.880000 7.920000 7.960000
8.000000 8.040000 8.080000 8.120000 8.160000 8.200000 8.240000 8.280000 8.320000 8.360000 8.400000 8.440000
8.480000 8.520000 8.560000 8.600000 8.640000 8.680000 8.720000 8.760000 8.800000 8.840000 8.880000 8.920000
8.960000 9.000000 9.040000 9.080000 9.120000 9.160000 9.200000 9.240000 9.280000 9.320000 9.360000 9.400000
9.440000 9.480000 9.520000 9.560000 9.600000 9.640000 9.680000 9.720000 9.760000 9.800000 9.840000 9.880000
9.920000 9.960000 10.000000 10.040000 10.080000 10.120000 10.160000 10.200000 10.240000 10.280000 10.320000
10.360000 10.400000 10.440000 10.480000 10.520000 10.560000 10.600000 10.640000 10.680000 10.720000 10.760000
10.800000 10.840000 10.880000 10.920000 10.960000 11.000000 0.016364 -0.049779 1.881053 0.017034 -0.052046
1.877202 0.018112 -0.055759 1.865879 0.019565 -0.060865 1.847425 0.021359 -0.067310 1.822185 0.023464 -0.075041
1.790502 0.025845 -0.084004 1.752719 0.028471 -0.094146 1.709178 0.031309 -0.105413 1.660224 0.034326 -0.117752
1.606199 0.037499 -0.121199 1.545447 0.040769 -0.145431 1.484219 0.044139 -0.169665 1.417133 0.047529 -0.176756
```

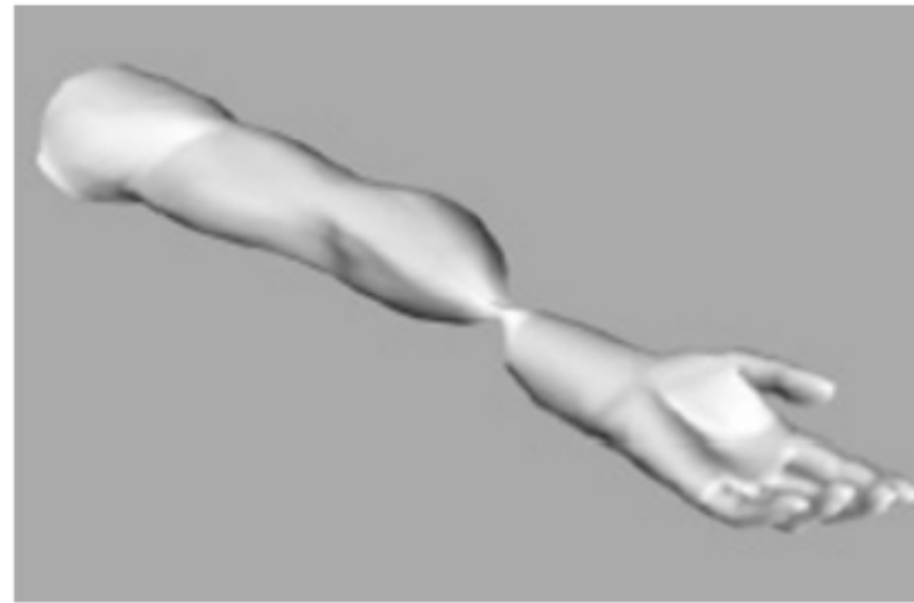
collada.dae

```
Pretty print ☐
{
  "images": [{
    "url": "MarineCv2_color.jpg",
    "uuid": "1F3E9A1C-DDD2-3F4D-80C1-BC5F609DC23B",
    "name": "MarineCv2_color.jpg"
  }],
  "geometries": [{
    "type": "Geometry",
    "uuid": "4181C04A-B75A-3FDC-9A89-89A1D2BCE0AF",
    "data": {
      "uvs":
[[[0.270354,0.313478,0.271947,0.319847,0.275729,0.309655,0.274243,0.3087,0.272832,0.307868,0.269884,0.3
10585,0.226132,0.380806,0.222211,0.383513,0.22838,0.387803,0.230469,0.384471,0.220584,0.387519,0.22777
8,0.393438,0.236551,0.373639,0.235651,0.37412,0.236463,0.379508,0.237612,0.379488,0.239952,0.371796,0.
240752,0.379751,0.245983,0.382396,0.244072,0.372847,0.029897,0.380277,0.026492,0.378217,0.024403,0.381
866,0.027378,0.383736,0.23588,0.388524,0.23554,0.386494,0.234758,0.383916,0.050509,0.560407,0.047071,0
.557888,0.045167,0.558824,0.048244,0.563377,0.064365,0.566106,0.055853,0.562263,0.05439,0.567096,0.064
598,0.572284,0.04127,0.560623,0.045026,0.567068,0.051922,0.572939,0.058767,0.576525,0.037322,0.561994,
0.040429,0.570751,0.047131,0.578106,0.034887,0.563887,0.036649,0.573688,0.041729,0.581221,0.051832,0.5
87997,0.054417,0.583009,0.033166,0.564334,0.033893,0.574381,0.038276,0.583055,0.046687,0.590068,0.0292
28,0.564585,0.030582,0.575915,0.035745,0.587145,0.043417,0.594797,0.012878,0.575167,0.025786,0.576822,
0.024107,0.563407,0.011428,0.55775,0.019835,0.589523,0.031426,0.590171,0.056273,0.542792,0.059073,0.54
5878,0.065176,0.5447,0.063035,0.542068,0.048447,0.546891,0.053936,0.549727,0.04472,0.552324,0.048873,0
.553322,0.055773,0.539823,0.044725,0.545943,0.064416,0.538785,0.040454,0.553447,0.054384,0.536842,0.04
1786,0.545212,0.065807,0.534961,0.037489,0.554075,0.049928,0.535922,0.039805,0.54519,0.035552,0.554559
,0.046737,0.533753,0.037925,0.544124,0.033877,0.554064,0.043337,0.529892,0.034539,0.543152,0.030422,0.
55413,0.04709,0.510318,0.042159,0.50679,0.026639,0.514508,0.031141,0.519839,0.038545,0.525958,0.02404,
0.532227,0.030178,0.540512,0.016619,0.542977,0.02613,0.552544,0.019243,0.52681,0.072806,0.54418,0.0680
36,0.541536,0.069568,0.544295,0.072537,0.546332,0.074457,0.542001,0.070694,0.540527,0.075113,0.538587,
0.072083,0.53632,0.070949,0.565762,0.071836,0.562397,0.067553,0.560436,0.064941,0.562816,0.080185,0.55
7515,0.078931,0.555483,0.076466,0.562489,0.076226,0.565704,0.058861,0.594811,0.064195,0.589419,0.06979
1,0.550726,0.066581,0.555646,0.071219,0.555536,0.07257,0.547845,0.061829,0.558444,0.058373,0.559956,0.
058247,0.554864,0.053888,0.556612,0.529304,0.679074,0.528585,0.691852,0.525233,0.680917,0.521958,0.670
57,0.005351,0.580703,0.020611,0.59504,0.002847,0.554083,0.040637,0.597389,0.01118,0.540222,0.054475,0.
599187,0.046209,0.601951,0.073623,0.5477,0.073842,0.556482,0.06595,0.597735,0.05175,0.602881,0.062732,
0.603138,0.513246,0.652152,0.507764,0.643571,0.505233,0.633772,0.522766,0.650073,0.06824,0.506816,0.07
5348,0.508814,0.073568,0.50543,0.068519,0.503123,0.062065,0.55231,0.067781,0.548426,0.055314,0.517723,
0.050992,0.513294,0.062031,0.506743,0.064039,0.51368,0.068682,0.511233,0.061823,0.58071,0.081265,0.529
```

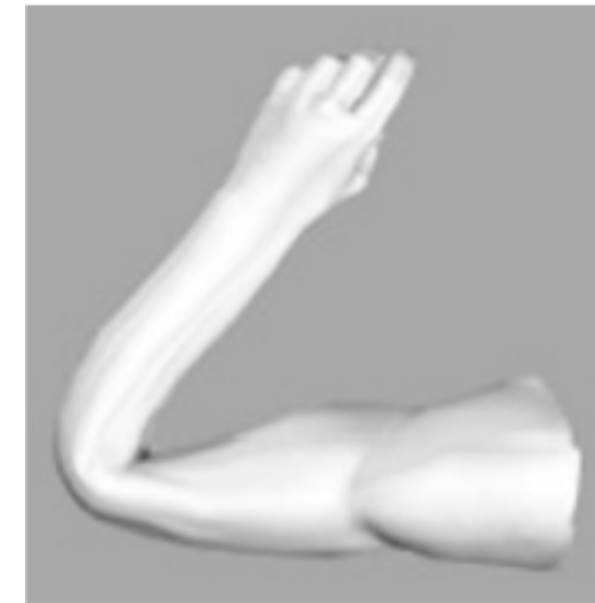
.json

Linear Blend Skinning - Limitations

- Paramètres de rigging non triviaux
- Artéfacts pour de grandes rotations : Candy wrapper, Collapsing elbow

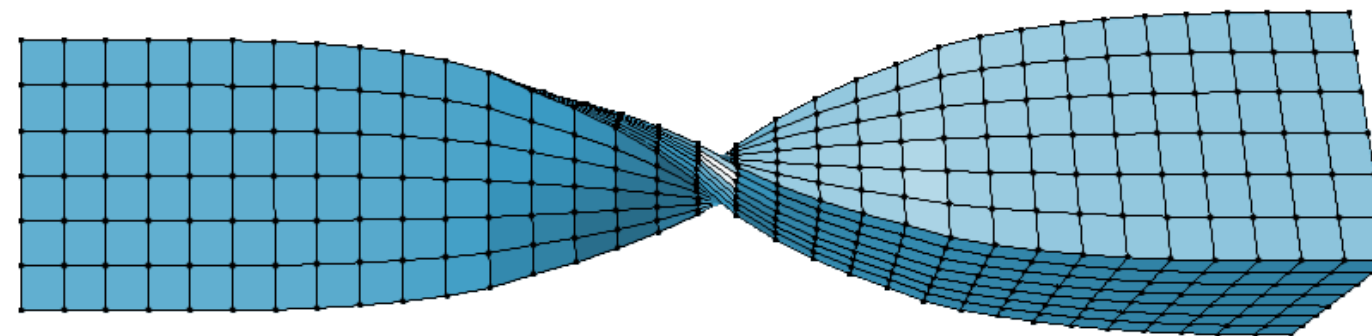


Candy wrapper



Coude qui s'effondre

Interpolation linéaire entre matrices de transformation rigides



Amélioration du skinning

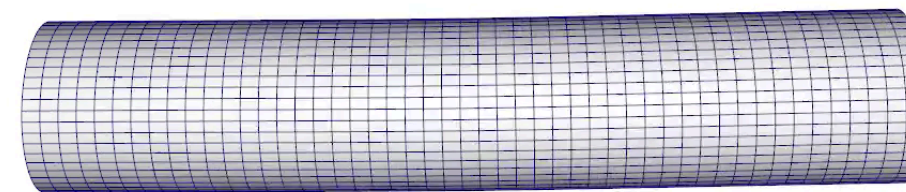
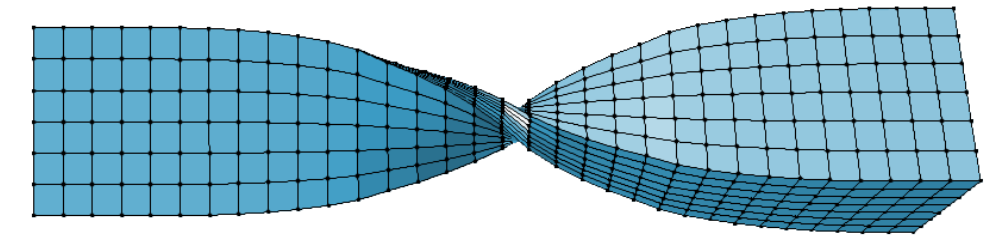
Éviter le mélange linéaire entre matrices de transformation affine

Première idée : Séparer les matrices de transformation rigide en

- partie rotation : mélanger en utilisant des quaternions (résoudre le problème du candy wrapper)
- partie translation : mélanger linéairement

Ne fonctionne pas pour une déformation générale : centre de rotation arbitraire

Rotation et translations sont traitées séparément



Possibilité de calculer un centre de rotation optimal à chaque image

[Spherical Blend Skinning: A Real-time Deformation of Articulated Models. L. Kavan, J. Zara. I3D 2005.]

Amélioration du skinning : Quaternions duaux

Deuxième idée : Utilisation du **quaternion dual**

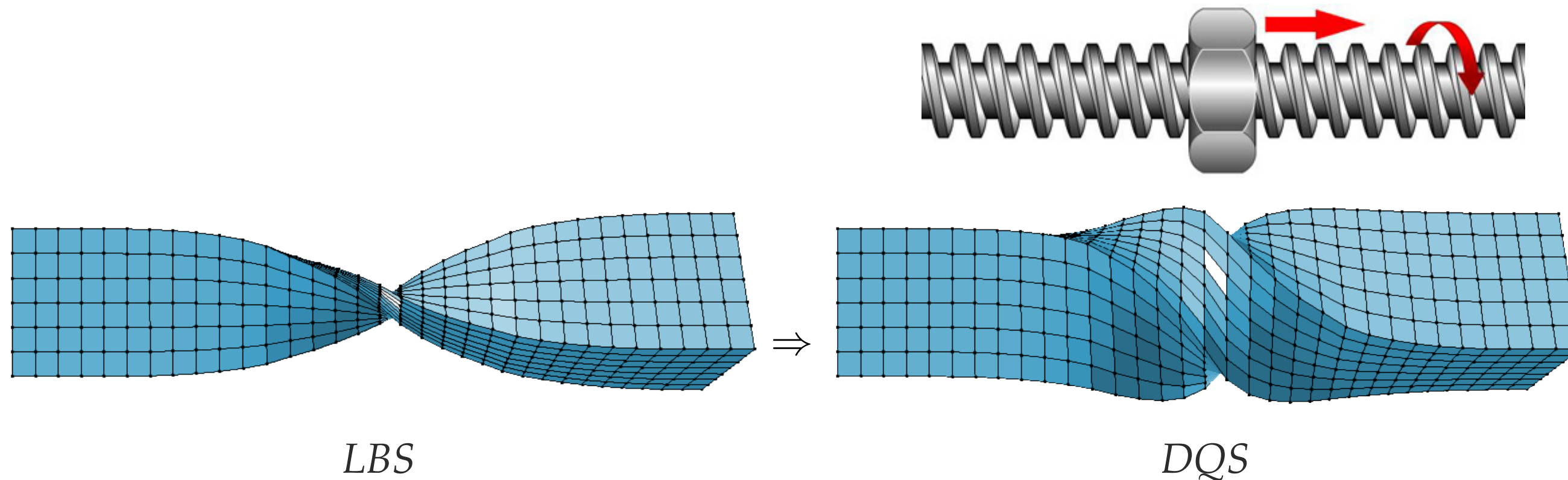
$$\hat{q} = q_0 + \epsilon q_\epsilon, \quad \epsilon \text{ élément dual.}$$

Quaternion dual = généralisation du quaternion pour gérer la *transformation rigide*

Rotation + translation

Le quaternion dual unitaire modélise la transformation rigide comme des **mouvements hélicoïdaux**

= rotation autour d'un axe suivie d'une translation dans la direction de cet axe



Nombre dual et quaternion dual

Nombre dual $a = a_0 + \epsilon a_\epsilon$

L'élément dual ϵ est nilpotent : $\epsilon \neq 0, \epsilon^2 = 0$

ϵ est couramment utilisé pour modéliser une quantité infinitésimale (ex. différentiation automatique)

Quaternion dual $\hat{q}$ = Généralisation du quaternion aux nombres duaux

$$\hat{q} = q_0 + \epsilon q_\epsilon$$

- q_0 : composante de rotation pure
- q_ϵ : encode la composante de translation (partie duale)

Étant donné un quaternion unitaire q_0 , et une translation $t = (t_x, t_y, t_z)$, le quaternion dual unitaire associé est

$$\hat{q} = q_0 + \frac{\epsilon}{2} q_t q_0 \quad q_t = (t_x, t_y, t_z, 0)$$

Le quaternion dual unitaire ($\|\hat{q}\| = 1$) décrit l'ensemble des transformations rigides comme un mouvement hélicoïdal.

Représentation axe-angle avec quaternion dual

De manière similaire au quaternion : correspondance "angle / axe"

- $\hat{q} = \cos(\hat{\theta}/2) + \hat{n} \sin(\hat{\theta}/2)$

- $\hat{\theta} = \theta_0 + \epsilon \theta_\epsilon$

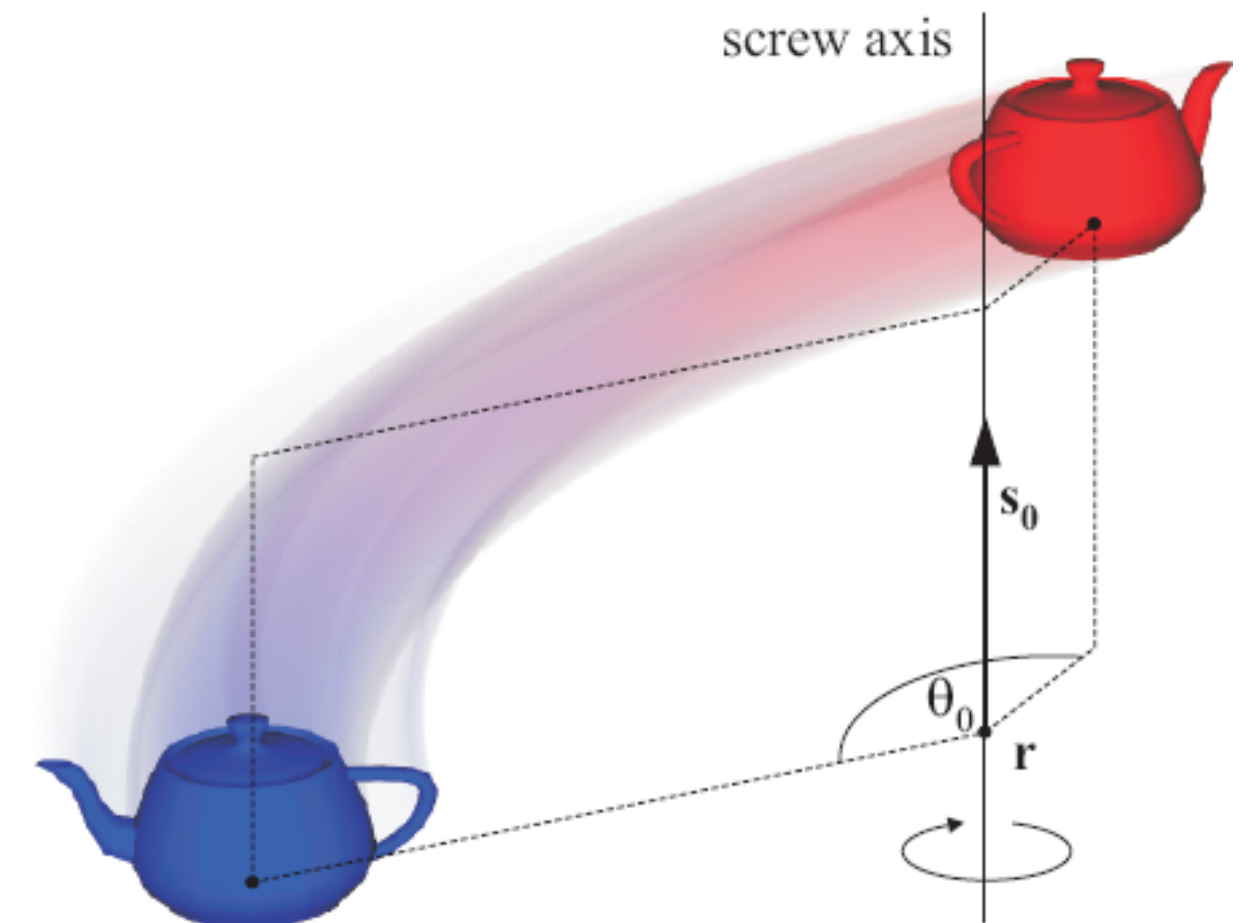
- θ_0 : angle de rotation

- $\theta_\epsilon = t \cdot n_0$: quantité de translation le long de n_0

- $\hat{n} = n_0 + \epsilon n_\epsilon$

- n_0 : axe de rotation

- $n_\epsilon = \frac{1}{2} ((n_0 \times t) \cotan(\theta_0/2) + t) \times n_0$: appelé le moment de l'axe de rotation.



Conversion d'un dual quaternion en rotation / translation

Étant donné un dual quaternion non unitaire en entrée $\hat{q}' = q'_0 + \epsilon q'_\epsilon$

Comment calculer les composantes de $\hat{q}' / \|\hat{q}'\| = \hat{q} = q_0 + \epsilon q_\epsilon$?

$\Rightarrow$ Forcer la paramétrisation en rotation-translation :

$$\hat{q} = q_0 + \frac{\epsilon}{2} q_t q_0, \quad q_t = (t_x, t_y, t_z, 0)$$

Premièrement, normaliser avec la composante non duale : $\hat{q}' / \|q'_0\|$

$$\Rightarrow q_0 = q'_0 / \|q'_0\|, \quad q_\epsilon = q'_\epsilon / \|q'_0\|$$

Deuxièmement, imposer la paramétrisation de la composante duale : $\frac{1}{2} q_t q_0 = q_\epsilon$

$$\Rightarrow q_t = 2 q_\epsilon q_0^*, \quad q_0^* \text{ conjugué de } q_0$$

Skinning par Quaternions Doubles (DQS)

1- Encoder la transformation rigide (q^i, t^i) en dual quaternion

$$\hat{q}^i = q_0^i + \frac{\epsilon}{2} q_t^i q_0^i, \quad q_t^i = (t_x^i, t_y^i, t_z^i, 0)$$

2- Calculer le blending dans l'espace des quaternions duaux (ScLERP)

$$\hat{q}' = \sum_i \omega_i \hat{q}^i = q_0' + \epsilon q_\epsilon'$$

3- Extraire les composantes (q, t) de $\hat{q}'$

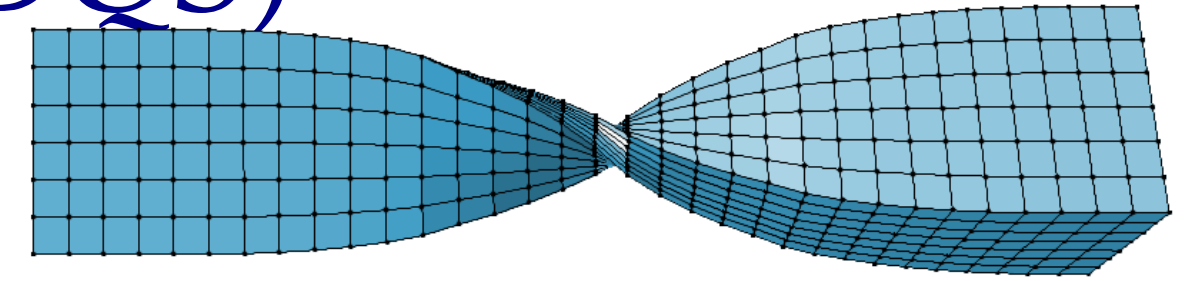
$$- q_0 = q_0' / \|q_0'\|, \quad q_\epsilon = q_\epsilon' / \|q_\epsilon'\|$$

$$- (t_x, t_y, t_z, 0) = 2 q_\epsilon q_0^*$$

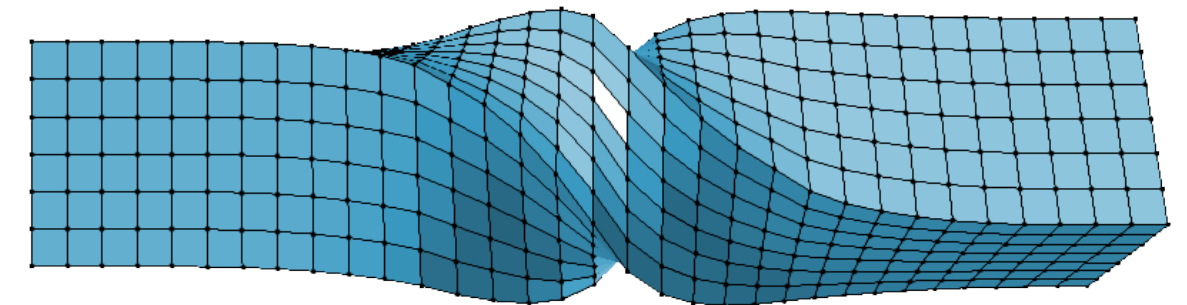
q_0^* : conjugué de q_0 .

4- Appliquer finalement la transformation (q_0, t) à la position p_0

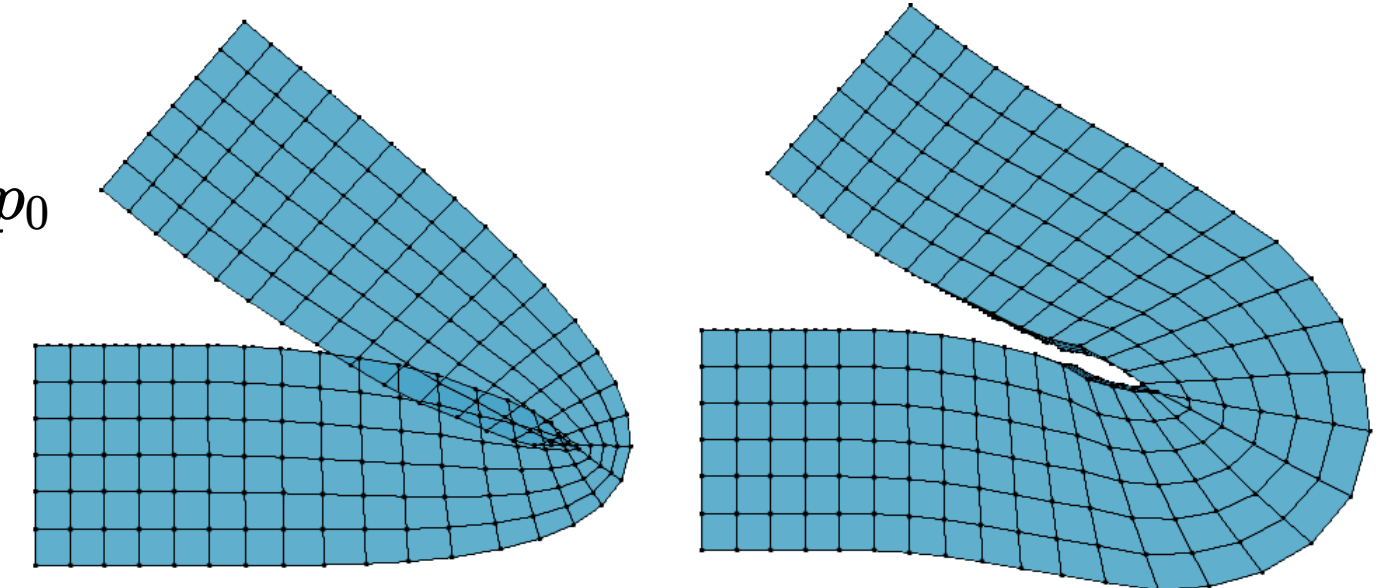
[*Skinning with Dual Quaternions*. Kavan et al. I3D, 2007]



LBS



DQS



LBS

DQS

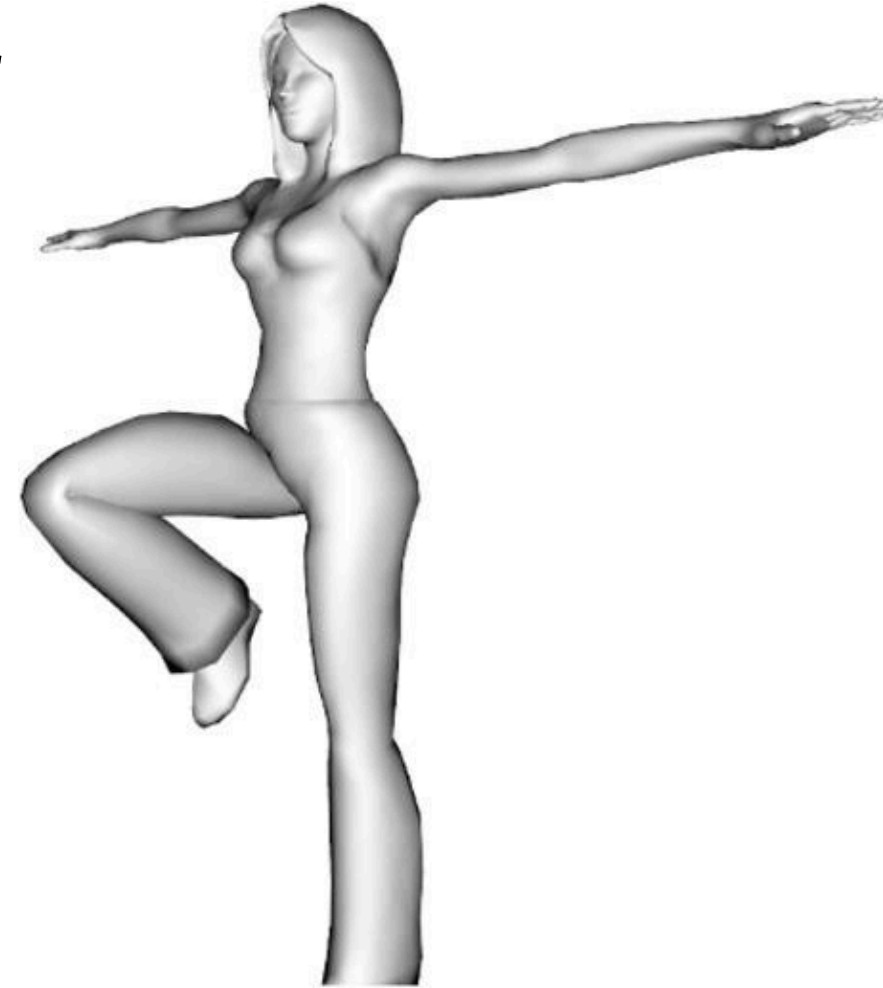
Quaternion Dual VS LBS

Quaternion dual

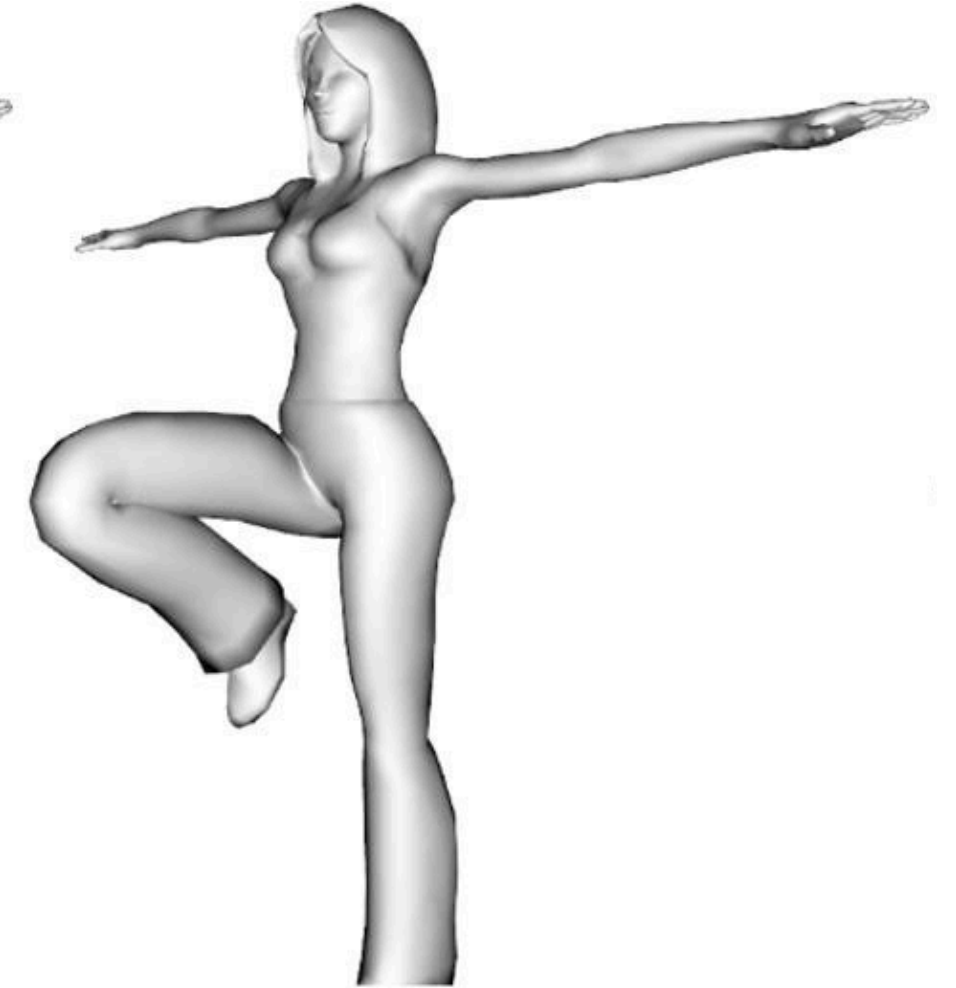
- (+) Résout complètement l'artefact *candy wrapper*
- (+) Presque aussi efficace que LBS
- (-) Peut créer un renflement artificiel / non désiré

Pas toujours préféré à LBS

Les deux solutions (LBS, DQS) sont proposées dans les outils standards



LBS



DQS

Animation de personnage

Animation squelettique

Structure du squelette

Caractéristiques

- Représentation hiérarchique

Tous les enfants suivent la transformation du parent

Nécessité de définir une racine : généralement au niveau des hanches/pelvis

- Pratique pour exprimer la transformation relative par rapport au repère parent

ex. La cheville est à 20° par rapport au genou

Conversion des repère locaux en repère globaux / coordonnées articulaires

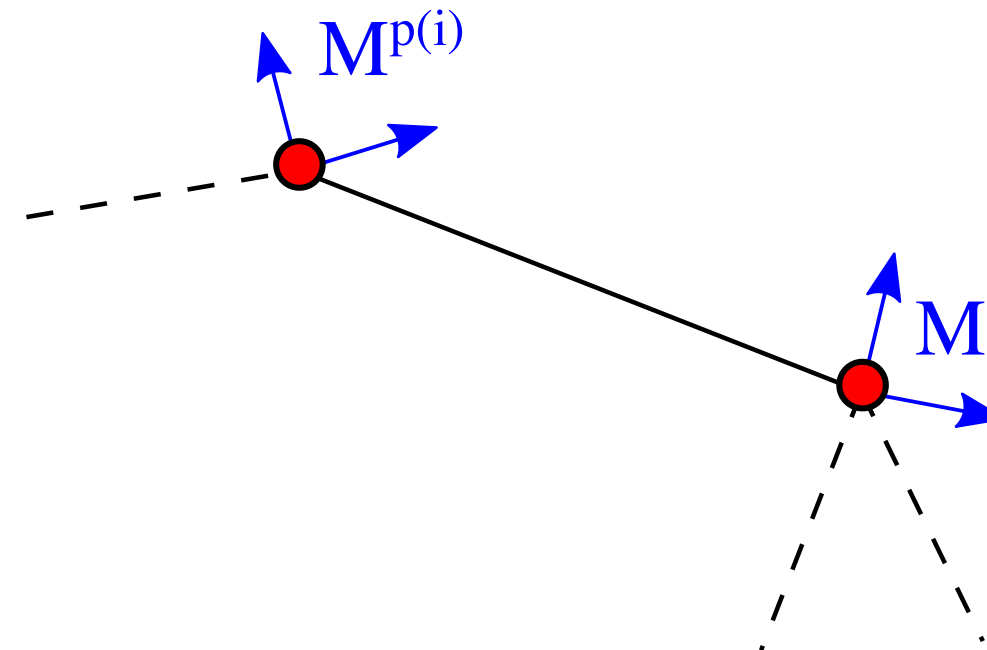
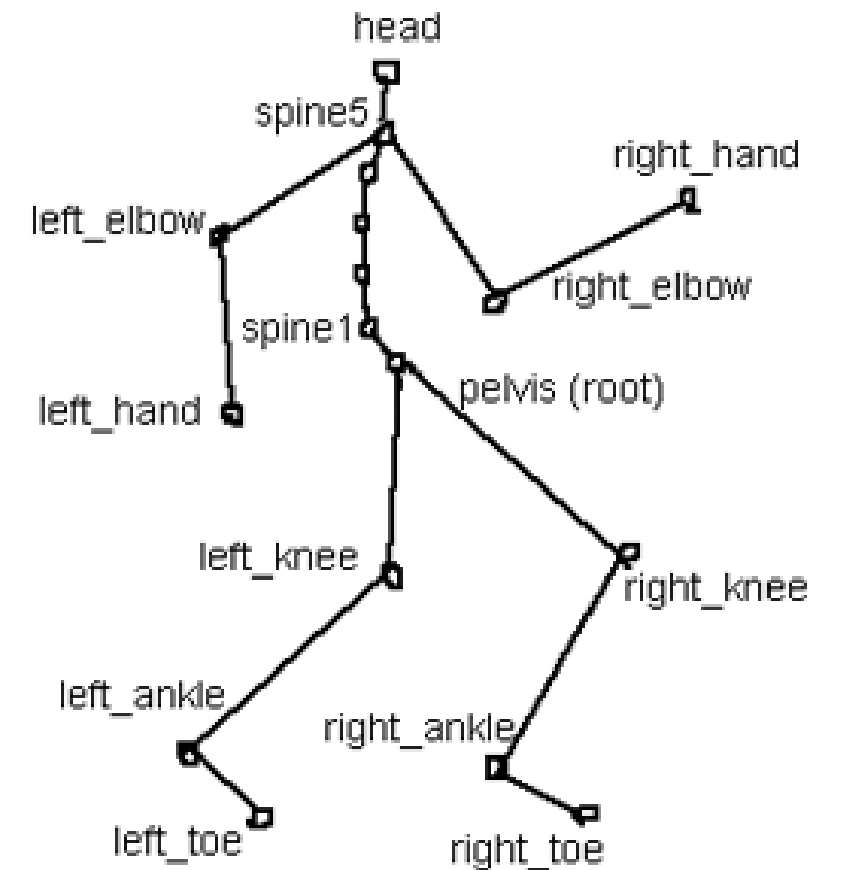
- Avec des matrices 4x4 M

$$M_{global}^i = M_{global}^{p(i)} M_{local}^i$$

- Avec translation t , rotation R

$$R_{global}^i = R_{global}^{p(i)} R_{local}^i$$

$$t_{global}^i = t_{global}^{p(i)} + R_{global}^{p(i)} t_{local}^i$$



Encodage du squelette hiérarchique

- Encodage le plus simple basé sur l'indice dans le vecteur

```
Geometry=[M0, M1, M2, M3, M4, M5]
```

```
Parent = [-1,0,1,1,0,4]
```

- Conversion des coordonnées locales en coordonnées globales

```
local (Geometry) <- std::vector of rotation (r), translation (p)
```

```
global (Geometry) <- std::vector of rotation (r), translation (p)
```

```
global[0] = local[0];
```

```
for(size_t k=1; k<N; ++k)
```

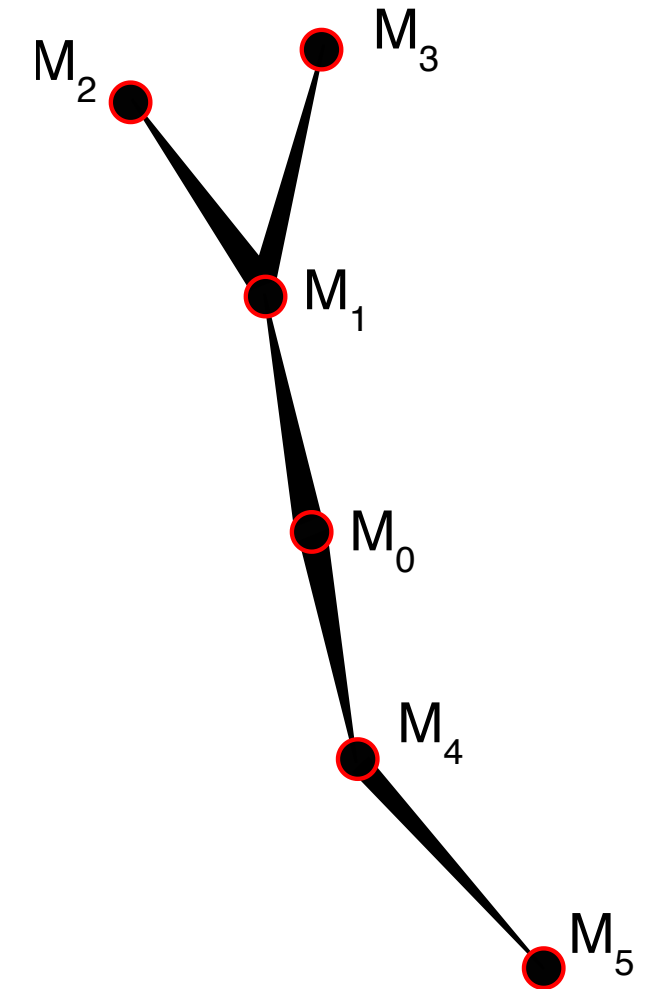
```
{
```

```
    int parent = Parent[k];
```

```
    global[k].r = global[parent].r * local[k].r;
```

```
    global[k].p = global[parent].r * local[k].p + global[parent].p;
```

```
}
```

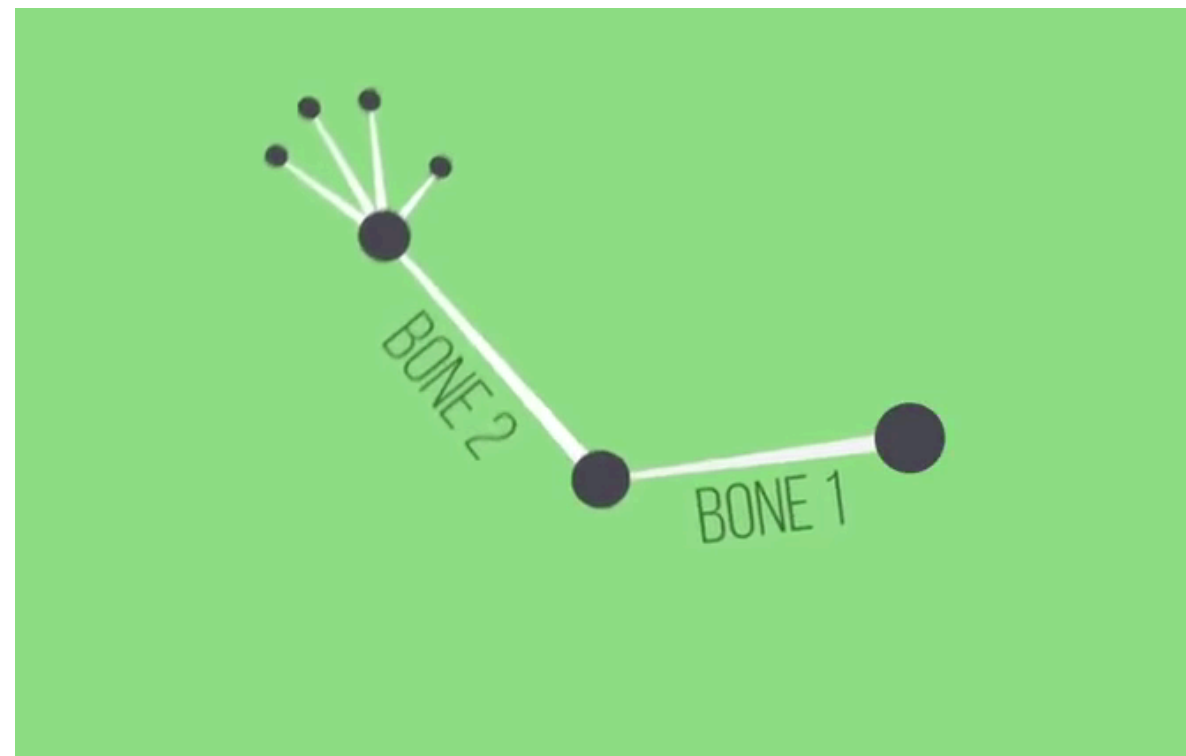
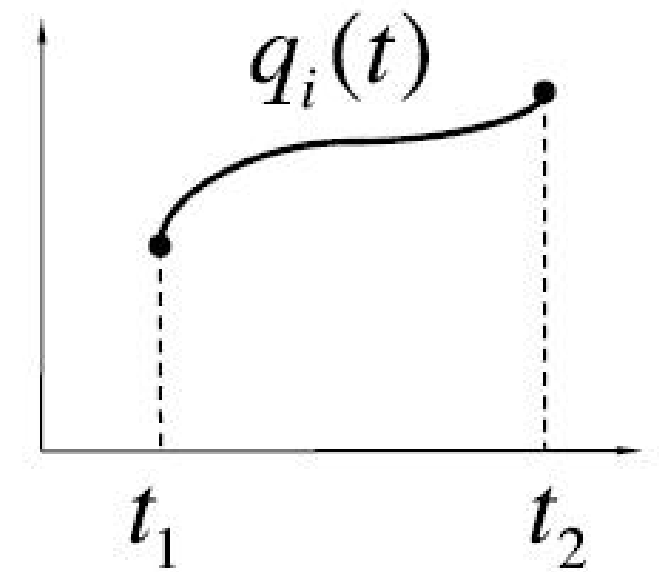
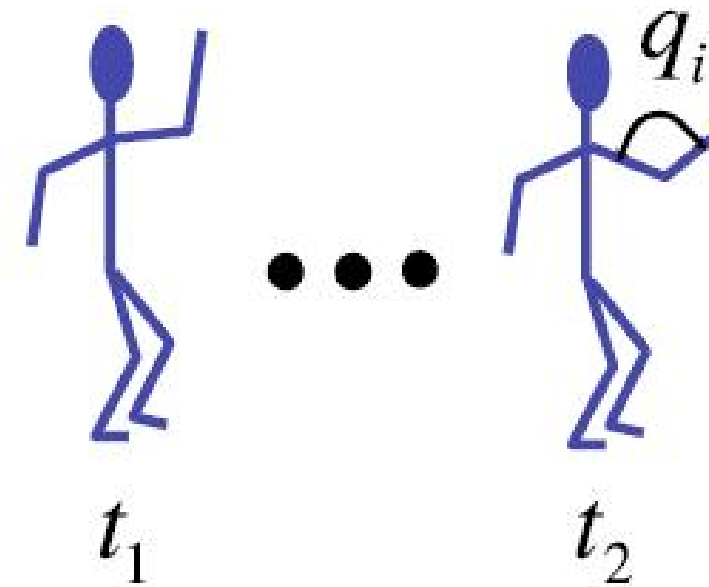


Cinématique directe

FK - Cinématique Directe

- Chaque angle d'articulation est réglé manuellement
- Adapté pour définir l'orientation de parties spécifiques
- Interpoler les rotations pendant l'animation

(+) Génère naturellement une trajectoire courbe



MiloScerny Animation

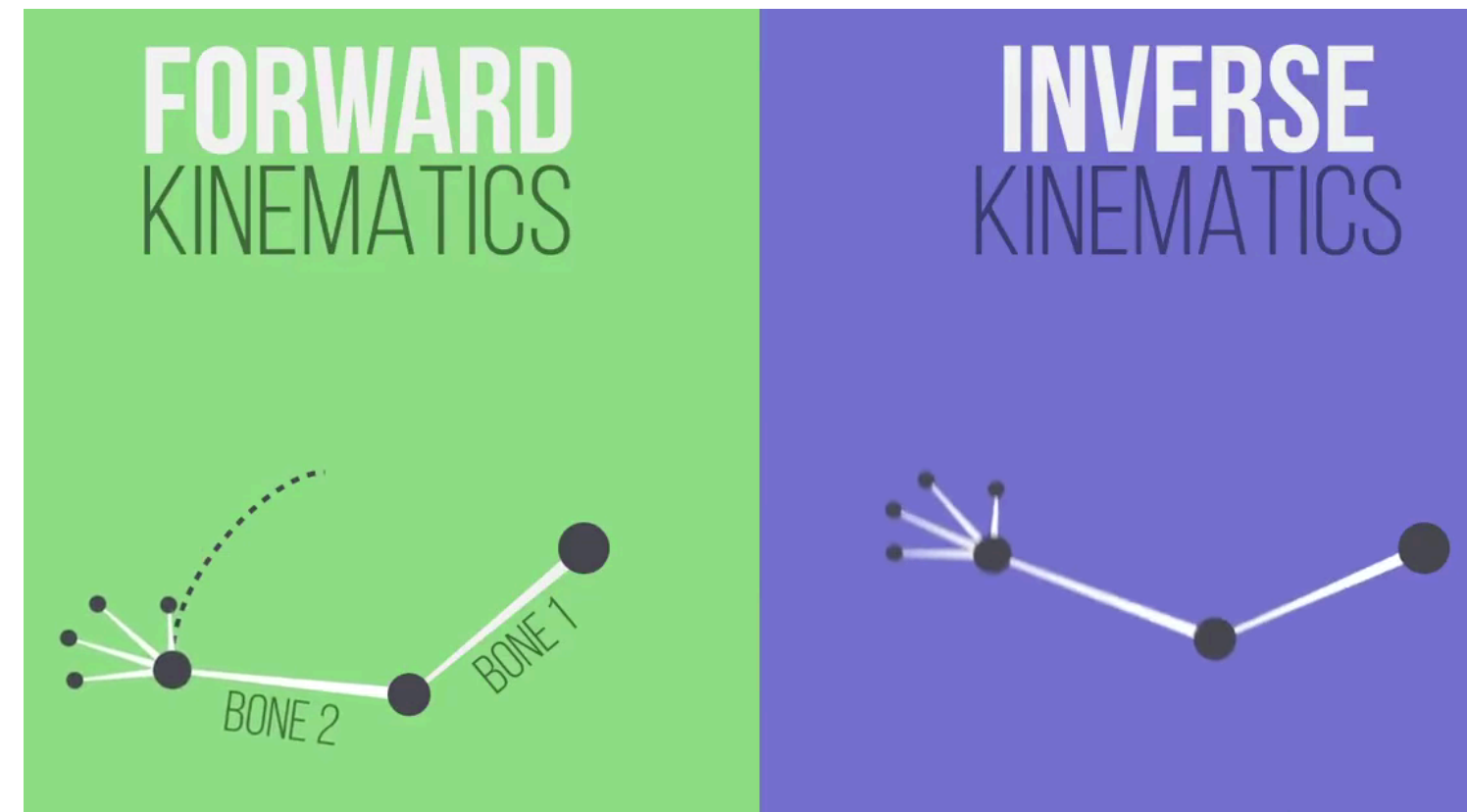
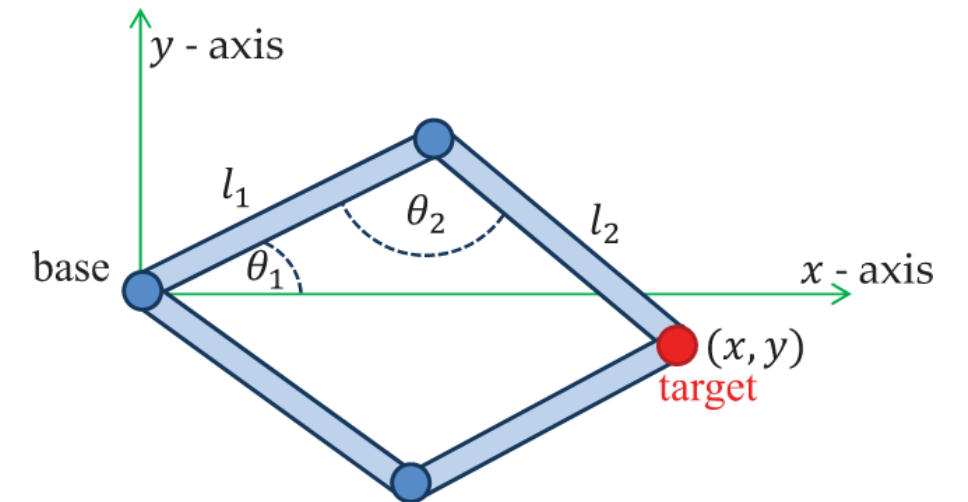
Cinématique Inverse

IK : Inverse Kinematics

- Décrire la position (/ et l'orientation) des *effecteurs finaux* (contact, marche, etc)
- Calculer les angles des articulations pour atteindre cette position

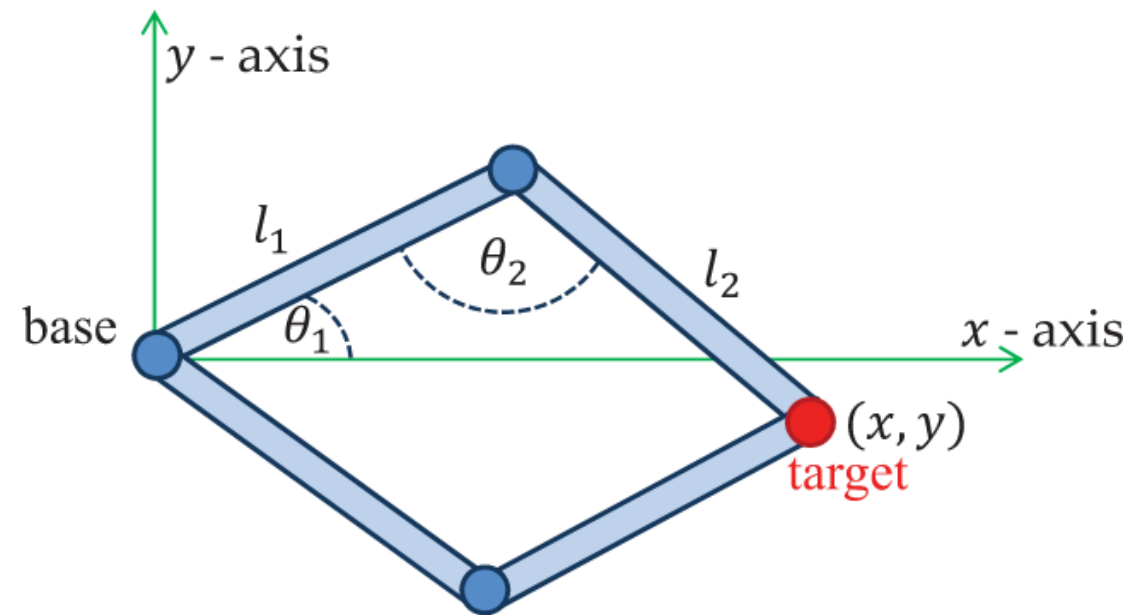
$$p_k = p_0 + \sum_{i=0}^{k-1} l_i R_i u_i$$

R_i peut être exprimé avec divers paramètres de rotation (Matrices, angles d'Euler, axe/angle, quaternions, etc)



MiloScerny Animation

Exemple d'IK avec deux os



[*Inverse Kinematics Techniques in Computer Graphics: A Survey.* A. Aristidou et al. STAR EG. 2017.]

En général, le cas général $p_k = f(\theta_i)$

- Chercher $\theta_i = f^{-1}(p_k)$
- f est une fonction non linéaire
- Il peut exister plusieurs solutions (ou aucune)
- Les solutions peuvent présenter des discontinuités
- Les solutions en forme fermée ne sont pas disponibles

Deux solutions définies par

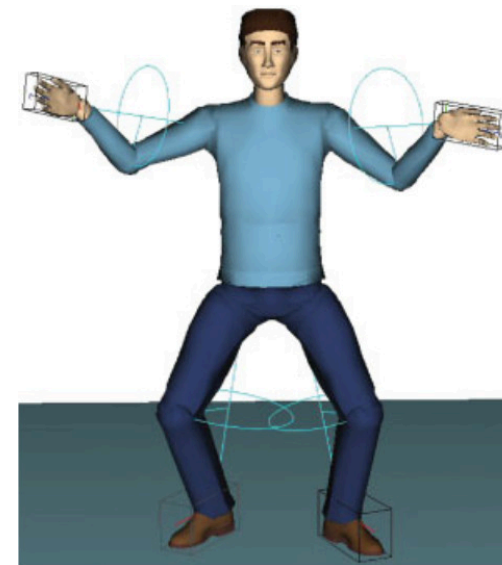
$$\cos(\theta_1) = \frac{l_1^2 + x^2 + y^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}}$$

$$\cos(\theta_2) = \frac{l_1^2 + l_2^2 - (x^2 + y^2)}{2l_1l_2}$$

Quelques tentatives de solutions explicites dans des cas spécifiques

[*Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs.* D. Tolani et al. *Graphical Models*, 2000.]

[*Analytical inverse kinematics with bodyposture control.* M. Kallmann. *Comp. Anim. & Virt. Worlds*, 2008]



IK : Méthodes numériques

Inversion numérique de $p = f(\theta)$, $\theta = (\theta_0, \dots, \theta_{N-1})$.

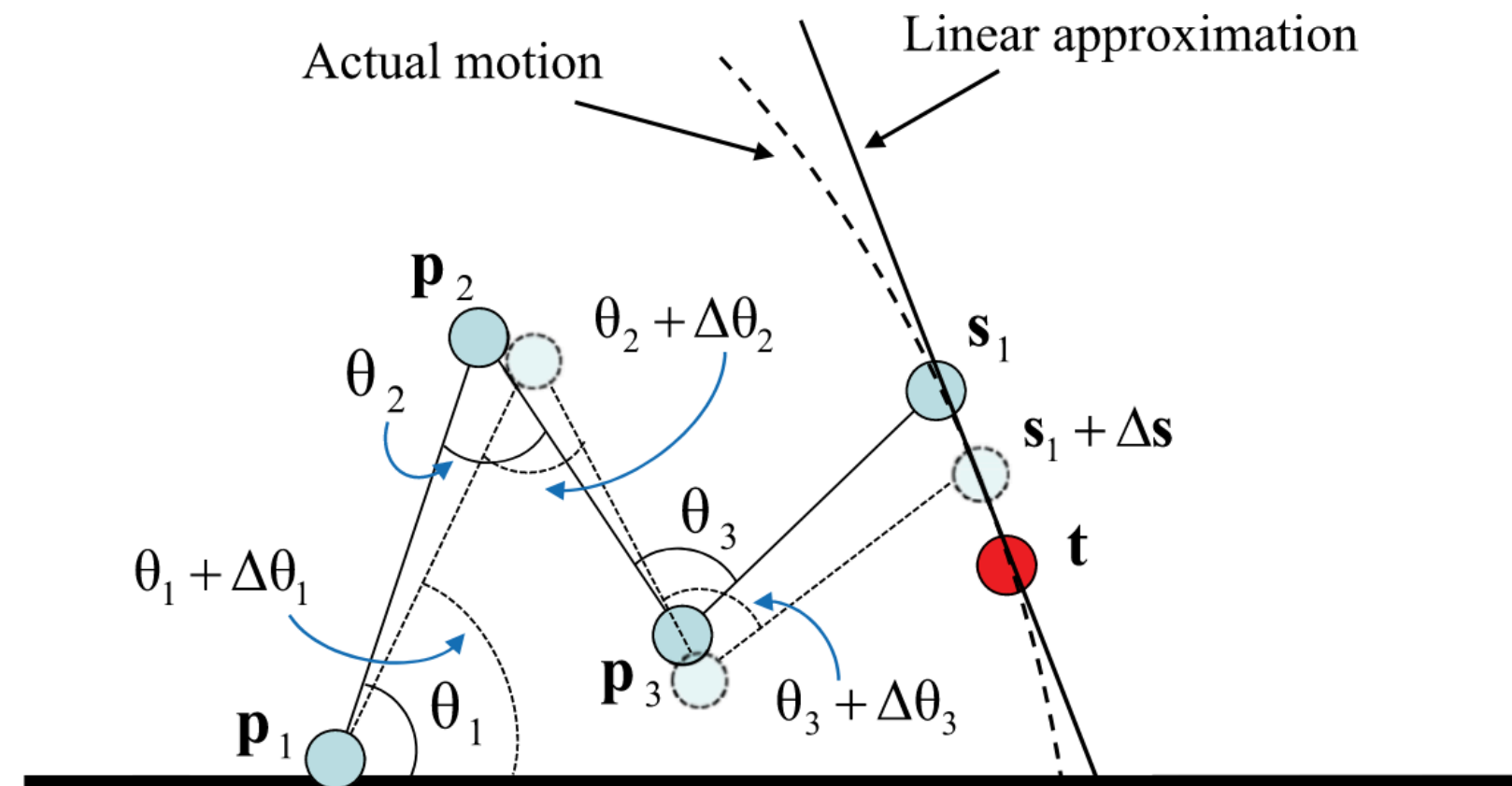
Considérer un petit pas $p \rightarrow p + \Delta p$

$$\Delta p \simeq \underbrace{\left(\frac{\partial f}{\partial \theta} \right)}_{\text{J}} \Delta \theta$$

J - matrice jacobienne.

→ Non carrée ($3 \times N$), non inversible.

Inconnues > # contraintes



IK : Méthodes numériques

Plusieurs approches possibles pour résoudre $J \Delta\theta = \Delta p$

- Pseudo-inverse

$$\Delta\theta = J^+ \Delta p, \text{ avec } JJ^+ = I$$

$$J^+ = J^T (J J^T)^{-1}$$

- Peut aussi être calculée en utilisant la SVD : $J^+ = V \Sigma^+ U^T$

$$\Sigma_{ii} = \sigma_i, \Sigma_{ii}^+ = 1/\sigma_i \text{ si } \sigma_i \neq 0, 0 \text{ sinon.}$$

- Ajout d'un amortissement pour compenser les singularités

$$\Delta\theta = J^T (J J^T + \lambda^2 I)^{-1} \Delta p$$

- Utilisation des méthodes de Newton

[*Inverse Kinematics Techniques in Computer Graphics: A Survey*. A. Aristidou. STAR EG 2017]

Cinématique Inverse

Exemple

Synthèse et contrôle de l'animation de squelette

Mélange d'animation de squelette

Pré-enregistrer plusieurs animations bouclants

Blending entre animations lors de transitions

`three.js` - Skeletal Animation Blending (model from realitymeltdown.com)

camera orbit/zoom/pan with left/middle/right mouse button

Note: crossfades are possible with blend weights being set to (1,0,0), (0,1,0) or (0,0,1)

Your graphics card does not seem to support [WebGL](#).
Find out how to get it [here](#).

Motion graphs

Aussi appelés Motion trees (jeux vidéo)

- Pré-stockent plusieurs animations pré-calculées

FK/IK, capture de mouvement, etc.

- Trouvent les transitions optimales entre différents mouvements

- [Mizuguchi et al., Data driven motion transitions for interactive games, EG short paper, 2001]

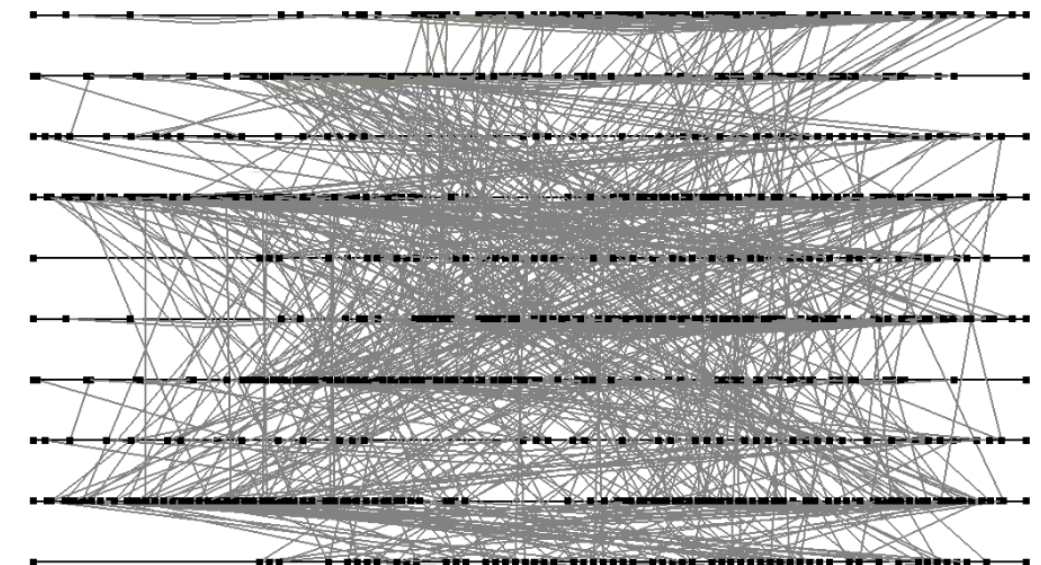
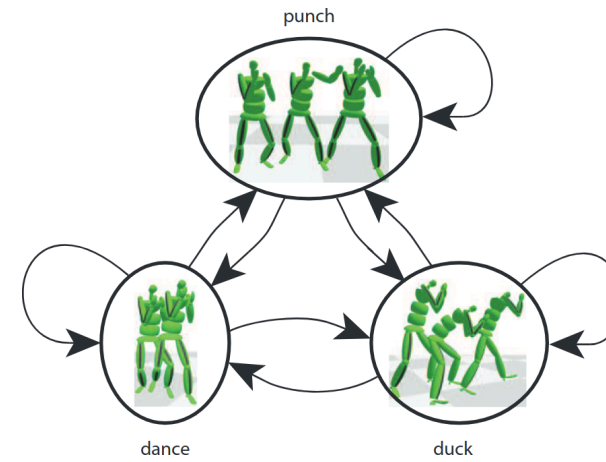
- [Kovar et al., Motion Graphs, ACM SIGGRAPH 2002]

- [Heck and Gleicher, Parametric Motion Graphs, ACM SIGGRAPH 2007]

- Plus récent : Utilisation du "Motion Matching"

Prédit dynamiquement la meilleure frame d'animation possible
basée sur l'état actuel et l'état futur désiré.

[Simon Clavet, Motion Matching and The Road to Next-Gen Animation, GDC 2016]



Contrôleurs

Mix entre mouvements prédéfinis et physique → permet les perturbations utilisateur

1 - Définir la cible $(\theta_{target}(t), \theta'_{target}(t))$

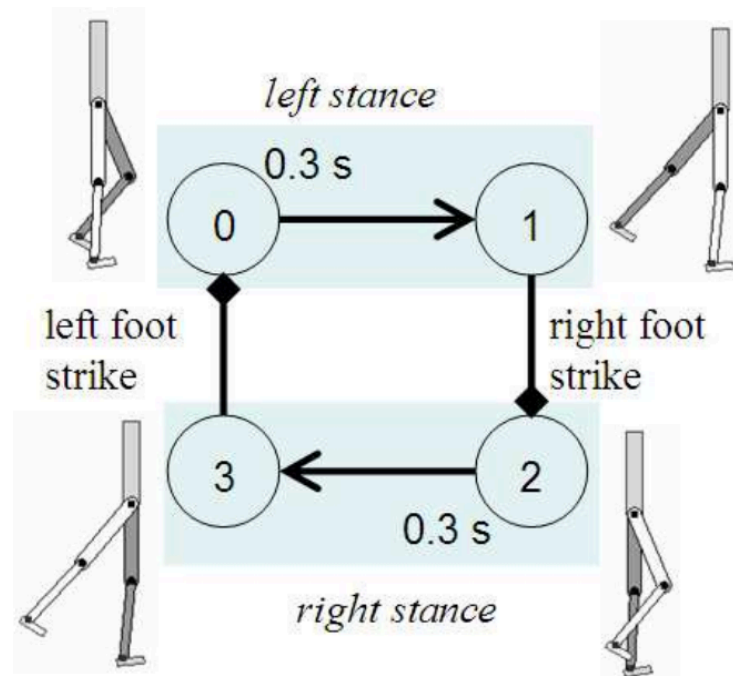
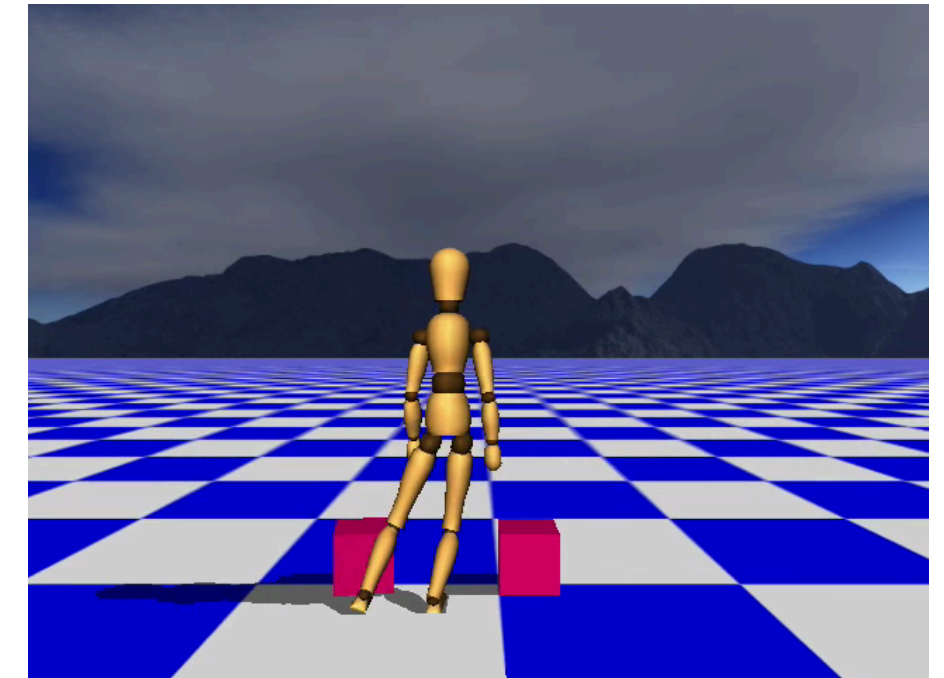
machine à états finis pré-définie (modèle de marche)

2 - Ajouter la perturbation utilisateur à l'état actuel

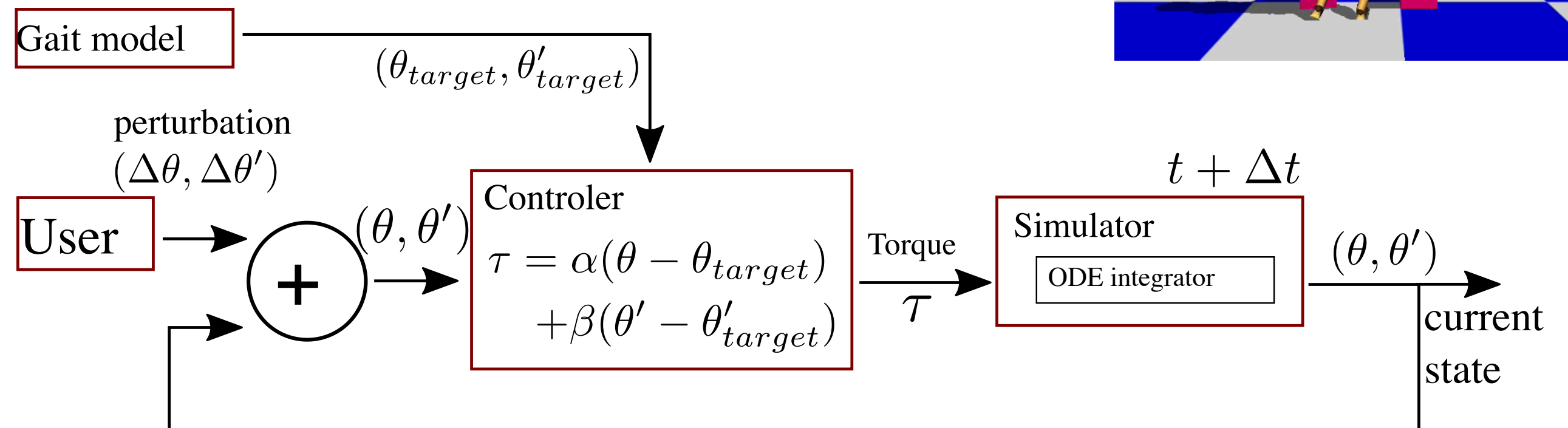
3 - Utiliser des contrôleurs proportionnels dérivatifs pour calculer le couple articulaire τ

4 - Intégrer le couple en utilisant un simulateur de corps rigides

5 - Itérer



Gait model

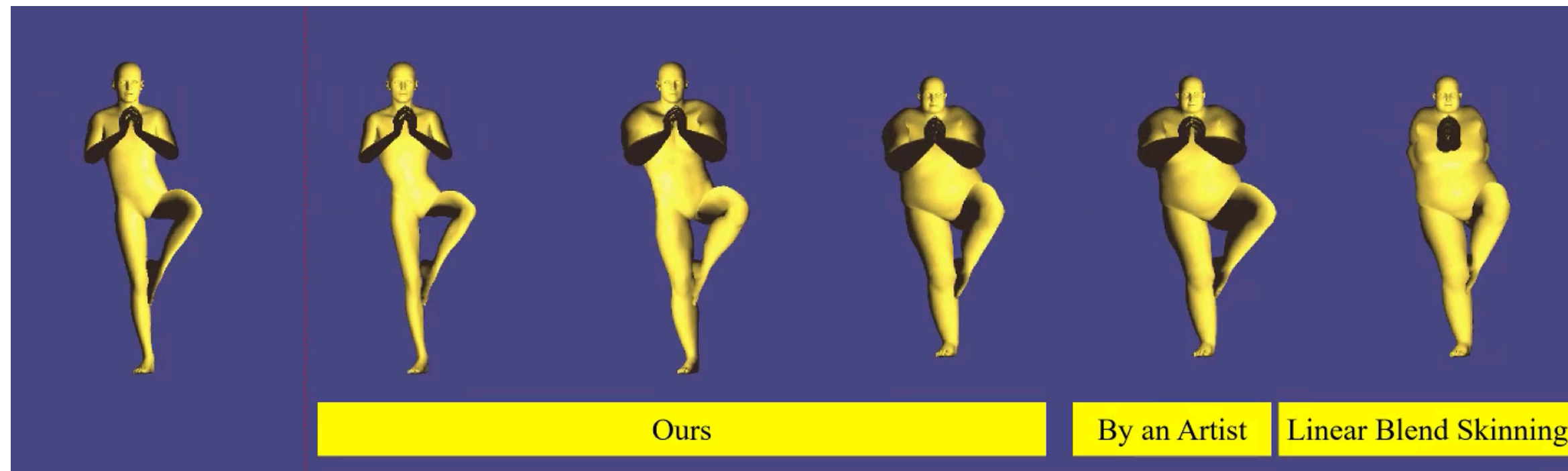
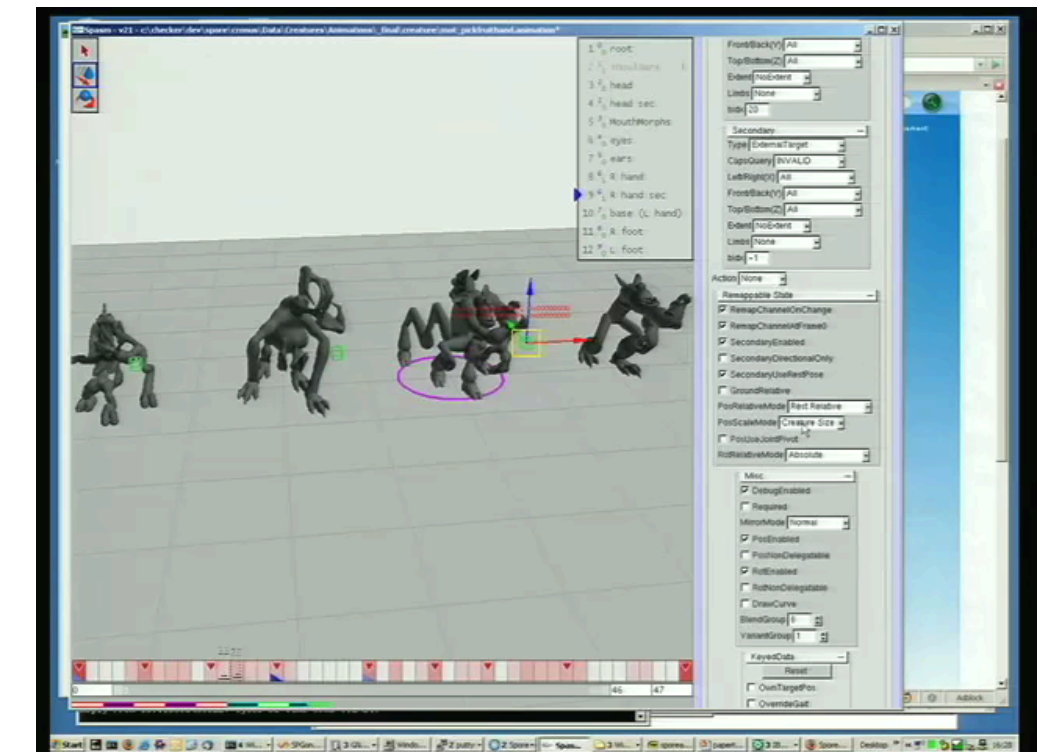


[M. Raibert et J. Hodgins. *Animation of Dynamic Legged Locomotion*, ACM SIGGRAPH 2001]

[K. Yin et al., *SIMBICON: Simple Biped Locomotion Control*, ACM SIGGRAPH 2007]

Transfert de mouvement

- Cartographie des coordonnées locales à partir des mouvements squelettiques
 - [C. Hecker et al., Real-time Motion Retargeting to Highly Varied User-Created Morphologies, SIGGRAPH 2008] (Spore)
- Incluant la morphologie de la forme
 - [Z. Liu et al., Surface based Motion Retargeting by Preserving Spatial Relationship, MIG 2018]



Conception d'animation

Basé sur la *Line of Action*

- [Guay et al., The Line of Action: an Intuitive Interface for Expressive Character Posing, ACM SIGGRAPH Asia 2013]
- [Guay et al., Space-time sketching of character animation, ACM SIGGRAPH 2015]
- [Choi et al., SketchiMo: Sketch-Based Motion Editing for Articulated Characters, ACM SIGGRAPH 2016]



0:00 / 0:27

0:00 / 0:35

Synthèse automatique d'animation squelettique

Travaux fondamentaux

- [Evolving Virtual Creatures. Karl Sims. SIGGRAPH 1994]
- [Automated Learning of Muscle-Actuated Locomotion Through Control Abstraction. Radek Grzeszczuk and Demetri Terzopoulos. SIGGRAPH 1995]
- *Optimisation vers une fonction objective couplée à des simulations de corps rigides*
- *Variation morphologique à partir d'un algorithme génétique*

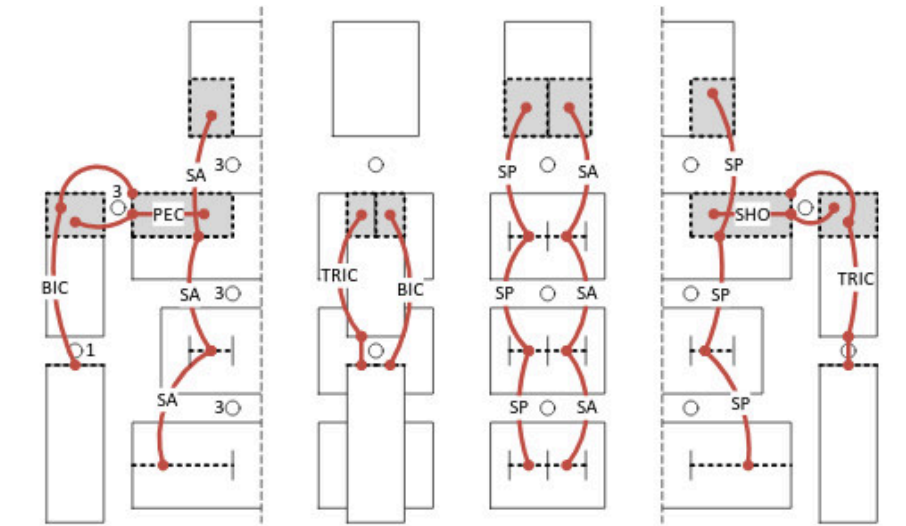
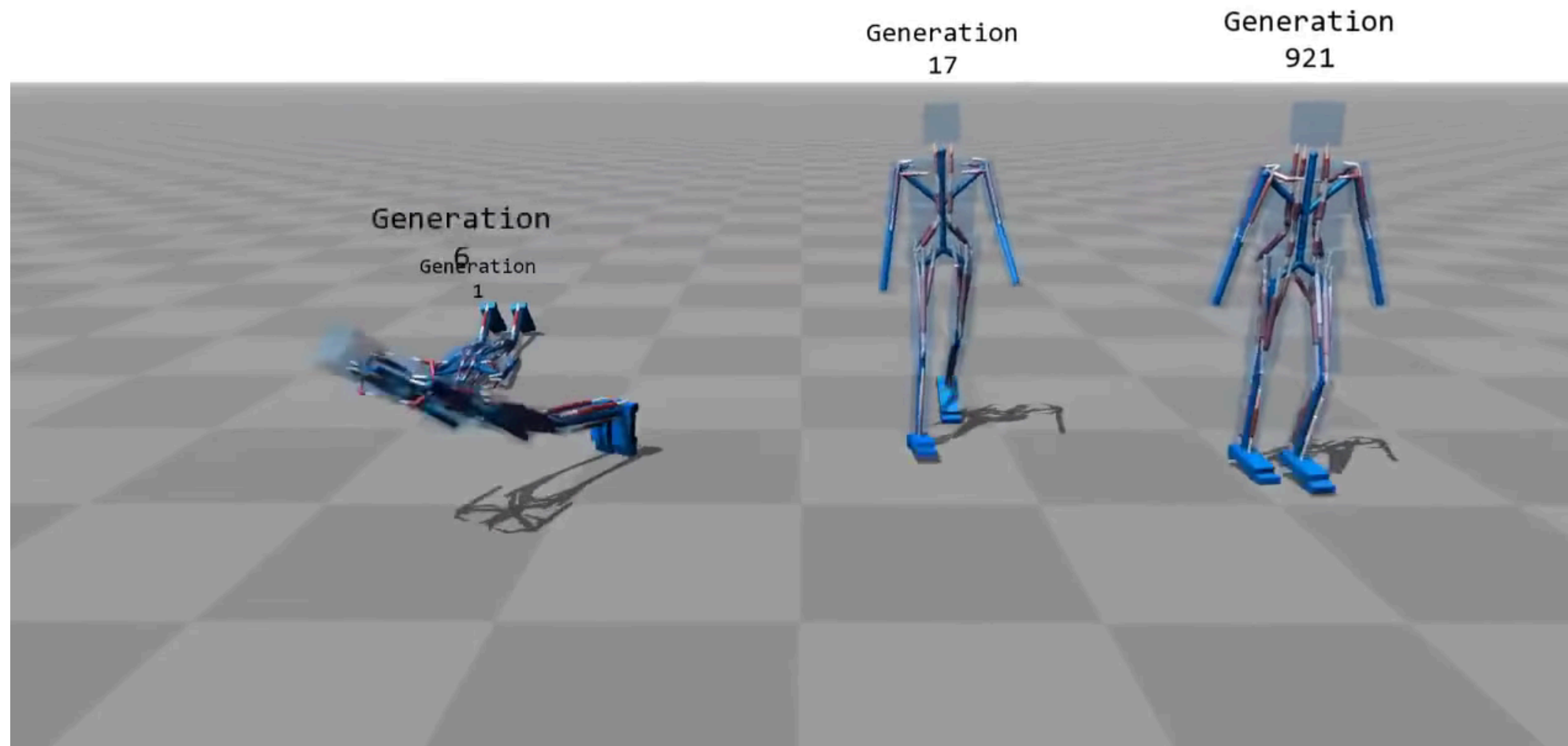
0:00 / 4:08

A horizontal progress bar with a light gray track and a white playhead marker at the beginning.

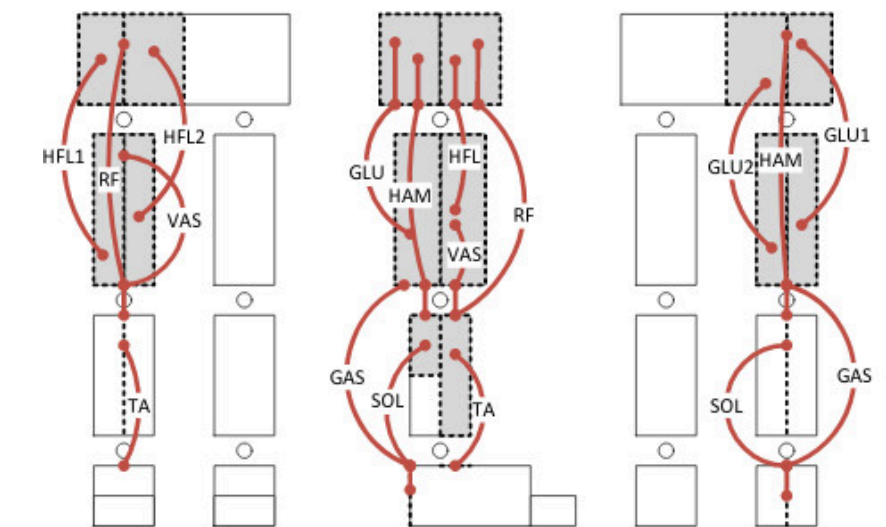
Optimisation de l'activation musculaire

- Prendre en compte un modèle biomécanique simple
- Optimiser la séquence d'activation via l'apprentissage par renforcement

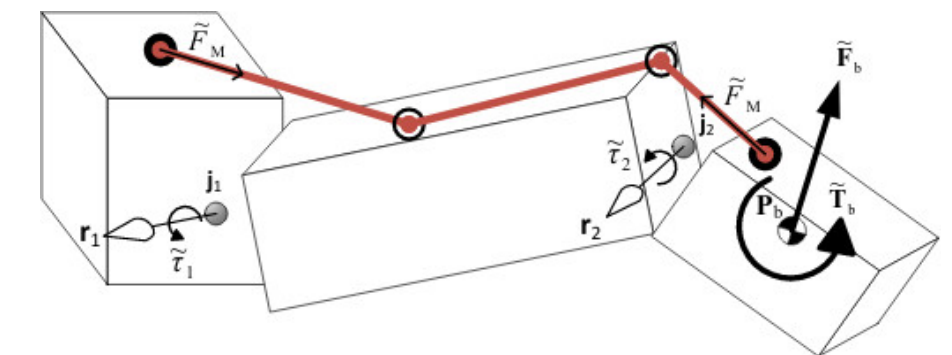
[*Flexible Muscle-Based Locomotion for Bipedal Creatures*, T. Geijtenbeek et al. SIGGRAPH Asia 2013.]



(a) Humanoid upper-body model: front, side arm, side body, and back.



(c) Humanoid lower-body model: front, side and back.

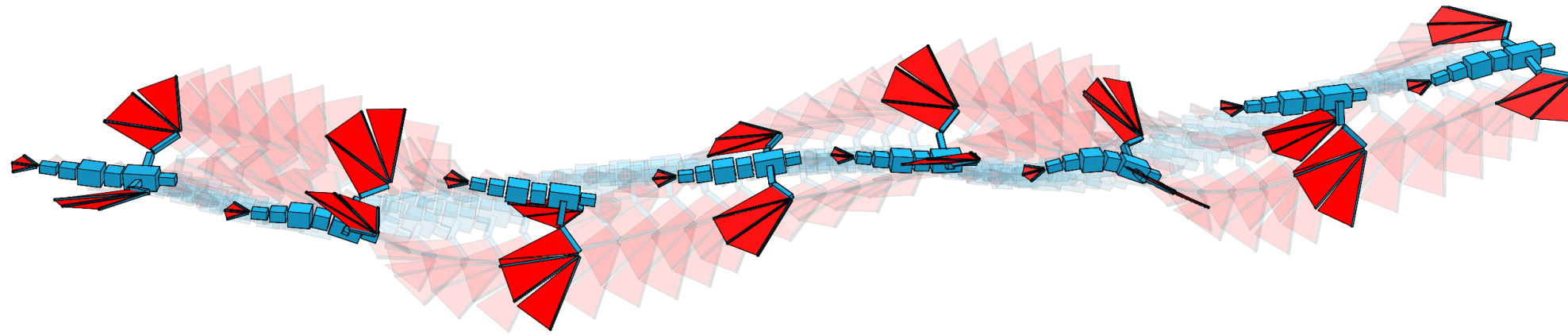


Utilisation de l'apprentissage profond

Apprentissage par renforcement profond pour l'optimisation complexe

Apprendre l'activation musculaire

[Won et al. *How to Train Your Dragon: Example-Guided Control of Flapping Flight*, ACM SIGGRAPH Asia 2017]



Apprentissage profond pour le contrôle du mouvement en temps réel

Apprendre la phase du cycle de mouvement.

Utiliser une grande base de données de capture de mouvement

[Holden et al., *Phase-Functioned Neural Networks for Character Control*, ACM TOG 2017]



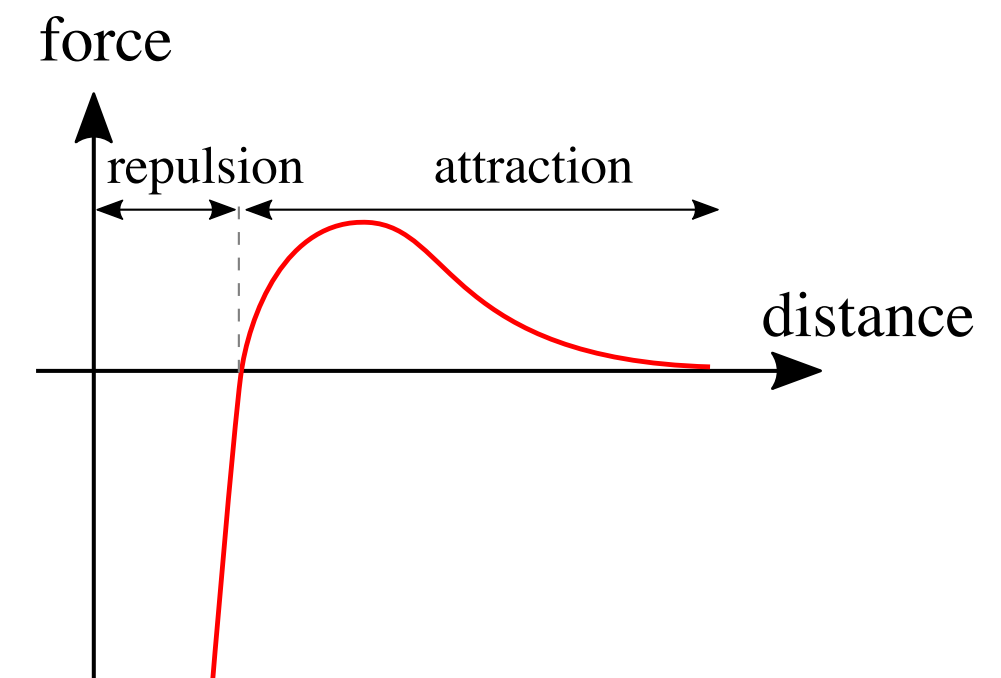
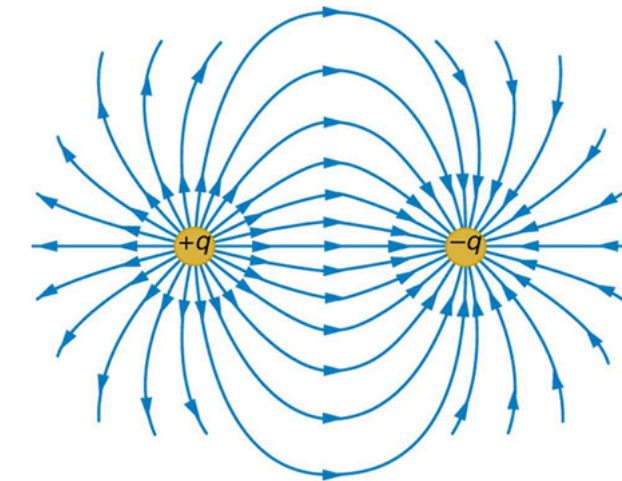
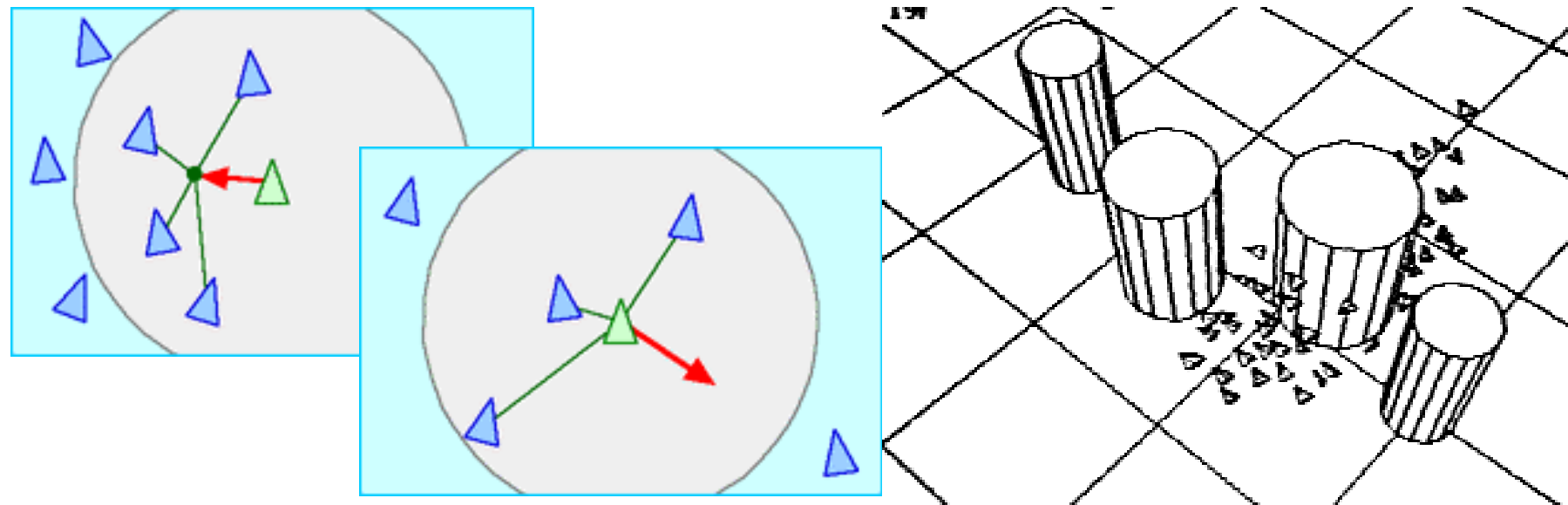
Animation de foules de personnages

Interaction entre particules

Interaction en tant que **champ de force** (interaction à distance)

Exemple d'utilisation

- Modélise une foule de personnages réalistes à grande échelle
Inspiré des forces en physique (ex. potentiel de Lennard-Jones)
Attraction à *longue portée*
Répulsion à *courte portée*
- Premier modèle : **Boids** Craig Reynolds 1987
- Étendu plus tard à la modélisation des foules humaines



Modèle de Boids

Introduit par

[Craig Reynolds. *Flocks, Herds, and Schools: A Distributed Behavioral Model*, SIGGRAPH 1987]

[Craig Reynolds. *Steering Behaviors For Autonomous Characters*. *Proceedings of Game Developers*, 1999]

Un boid est défini par sa

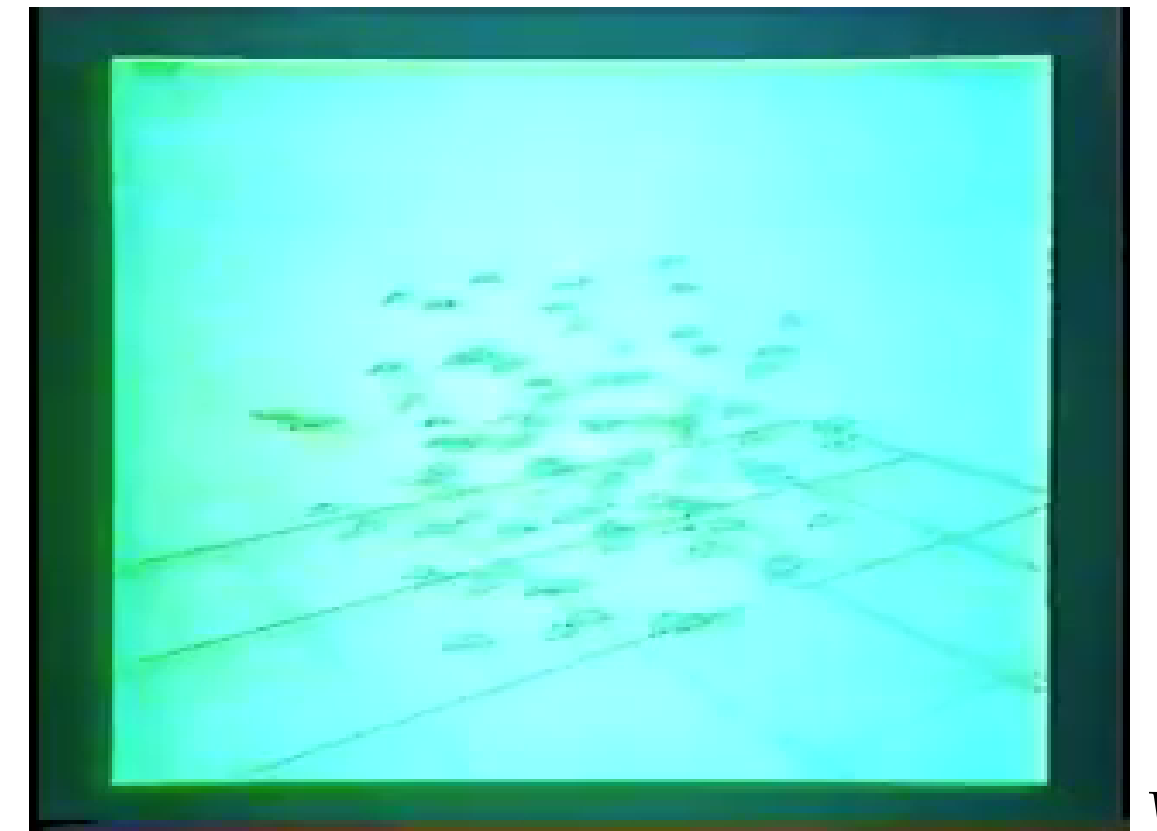
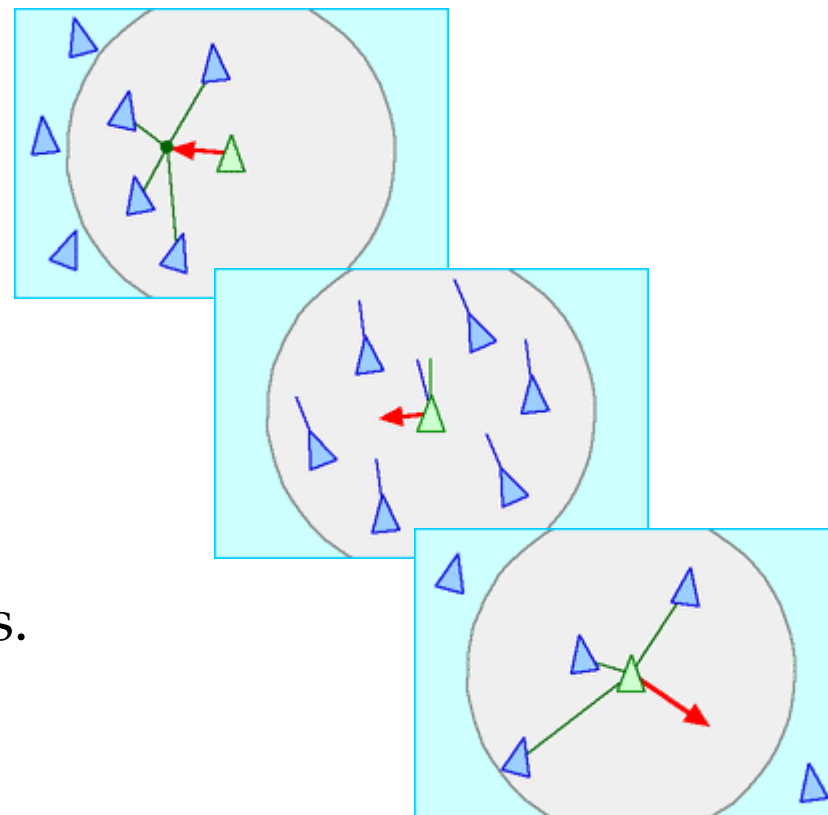
- Position
- Vitesse
- Forces qui s'exercent sur lui

Comportement guides de base modélisant les volées

Steering behavior

- **Cohésion** entre particules locales
- **Alignement** entre particules locales
- **Séparation** entre particules trop proches

=> Conduit à des comportements globaux émergents.



originale en 1986 (C. Reynolds)

Vidéo

Modèle de Boids - Modèle de base.

- Définir une position/vitesse initiale aléatoire pour N particules.
- Définir la force d'attraction/répulsion en fonction des distances par paires

$$F(p_i) = \sum_j f(\|p_j - p_i\|) \frac{p_j - p_i}{\|p_j - p_i\|}$$

Exemple

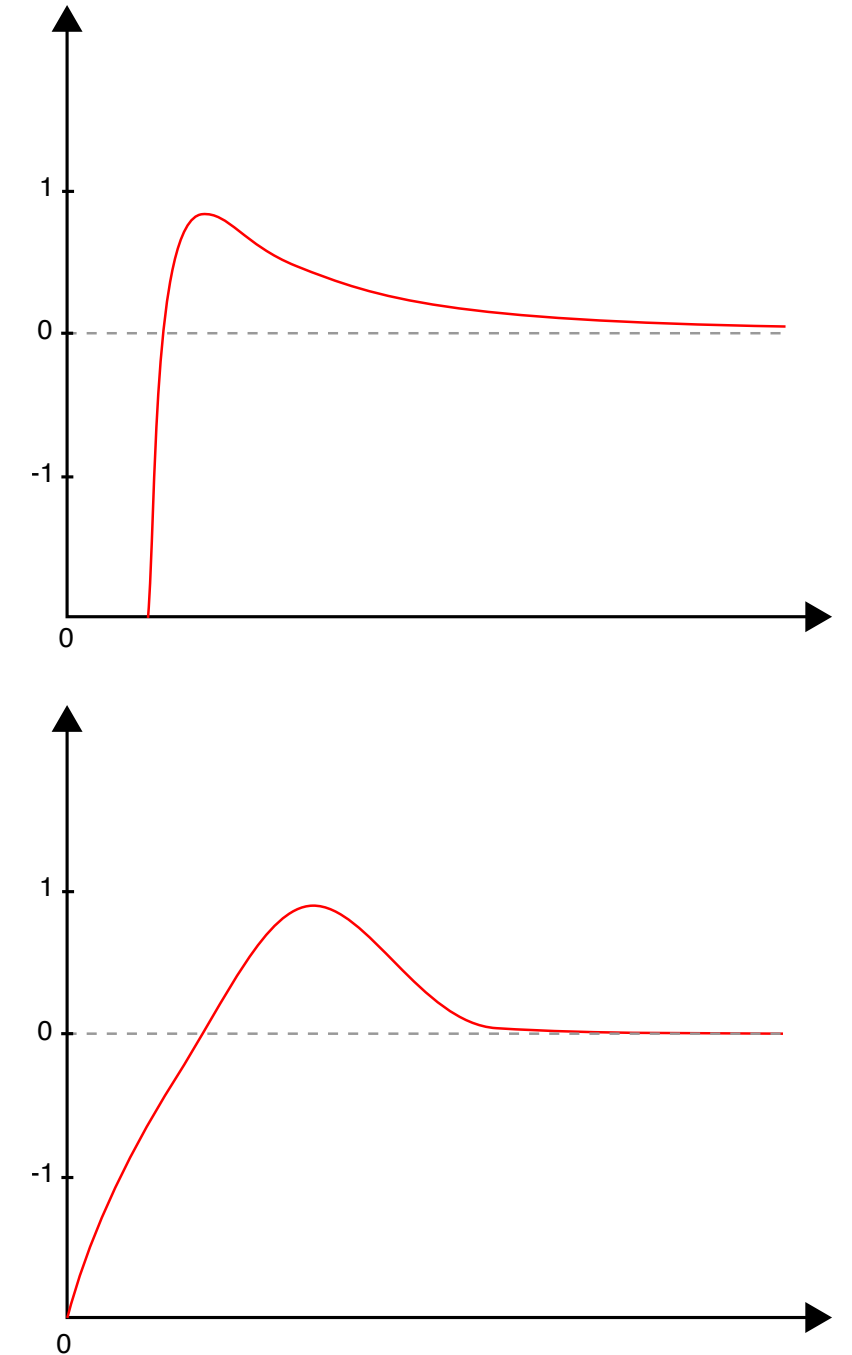
- Inverse de la distance $f(x) = \frac{\alpha_1}{x^2} - \frac{\alpha_2}{x^4}$

- Exponentielle/Gaussienne $f(x) = \alpha_1 \exp\left(-k \left(\frac{x - x_0}{x_0}\right)^2\right) - \alpha_2 \exp\left(\frac{x}{x_0}\right)$

- Intégrer la position et la vitesse au cours du temps

$$v^{t+\Delta t}(p_i) = v^t(p_i) + \Delta t F(p_i^t)$$

$$p^{t+\Delta t}(p_i) = p^t(p_i) + \Delta t v^{t+\Delta t}(p_i)$$



Modèle de Boids - Complexité.

Implémentation triviale :

```
struct particle { vec3 p, v, f; };

std::vector<particle> boids;

// Initialiser N boids ...
// ...

// calculer la force par paire
for(int i=0; i<N; ++i)
{
    for(int j=0; j<N; ++j)
    {
        if( i!=j )
        {
            const vec3& pi = boids[i].p;
            const vec3& pj = boids[j].p;

            boids[i].f += force( norm(pi,pj)) * (pi-pj)/norm(pi-pj);
        }
    }
}

// intégration
for(int i=0; i<N; ++i)
{
    boids[i].v = boids[i].v + dt * boids[i].f;
    boids[i].p = boids[i].p + dt * boids[i].v;
}
```

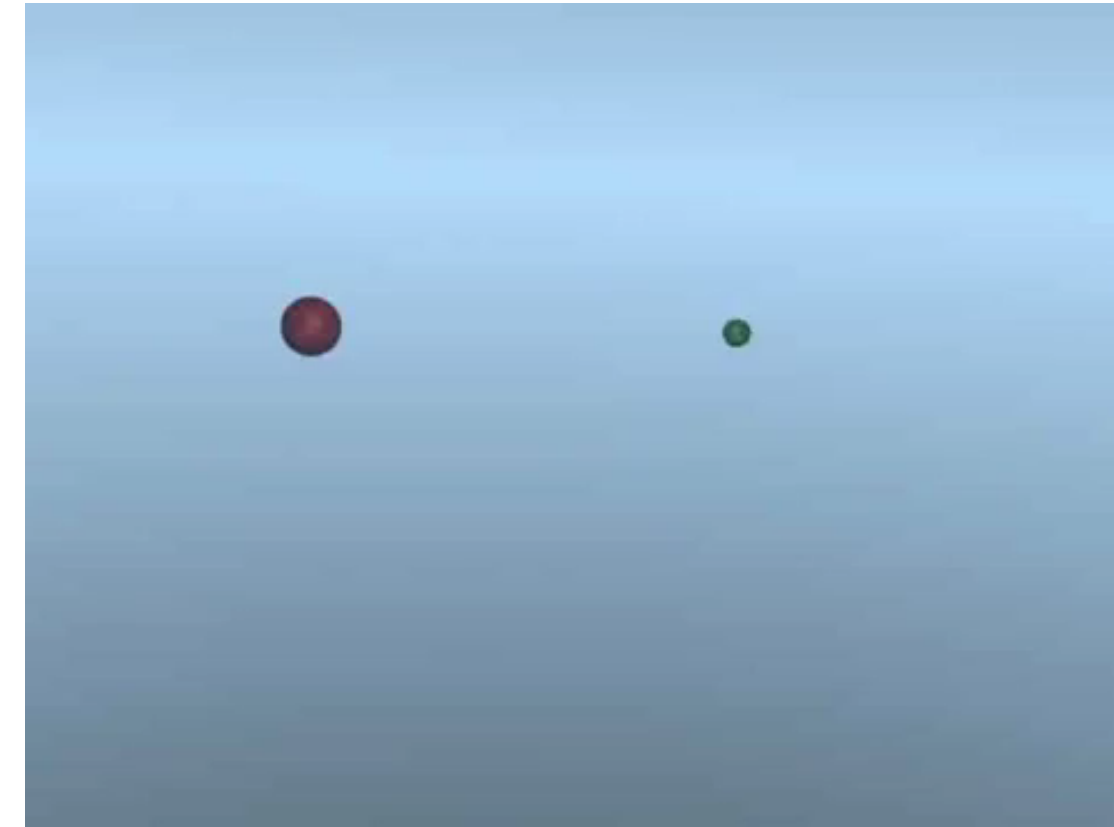
- Quelle est la complexité (par rapport à N) de cet algorithme ?
- Pouvez-vous penser à une manière d'être plus efficace pour un grand N ?

Modèle Boids - Utilisation et limites

- Adapté aux volées (oiseaux, poissons - observation / réaction)
- Affiche les particules en utilisant un modèle 3D animé

Comportements supplémentaires

- Position / vitesse objectif
 - Contraintes : évitement d'obstacles, vitesse limitée
 - Poursuivre et échapper à une cible / autres particules - suivre le leader, prédateurs, etc.
-
- *La collision entre particules est-elle possible ?*
 - *Les déplacements humains sont principalement guidés par la vision, quel élément clé manque dans le modèle de base des boids basé sur les forces ?*



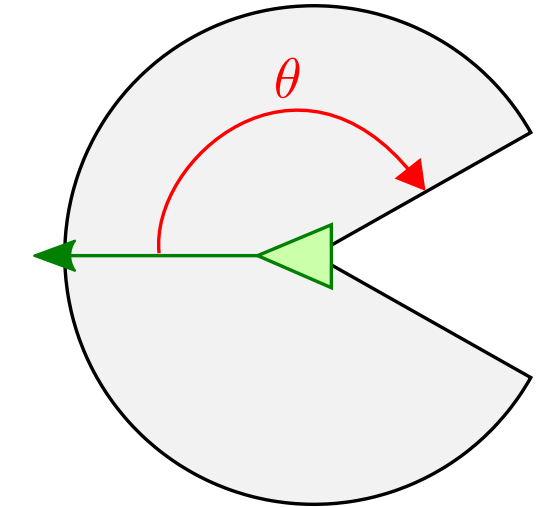
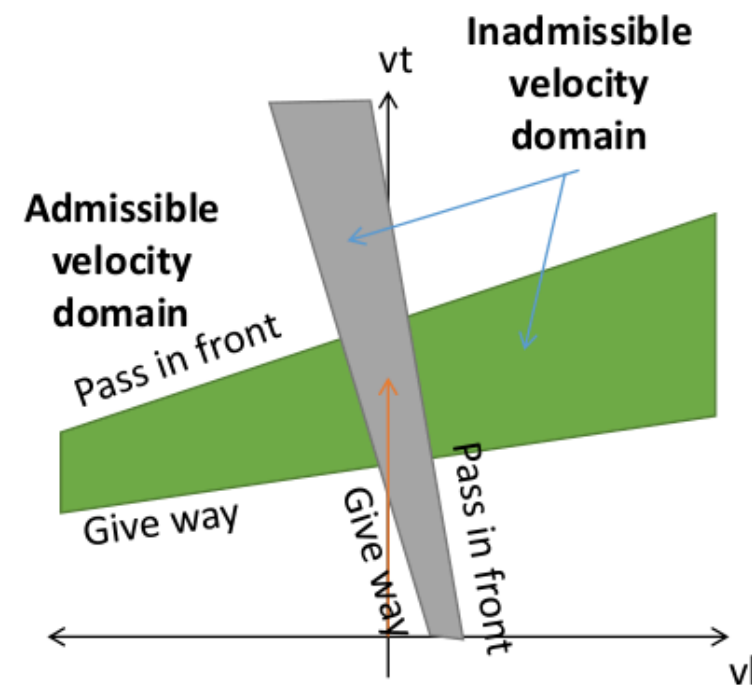
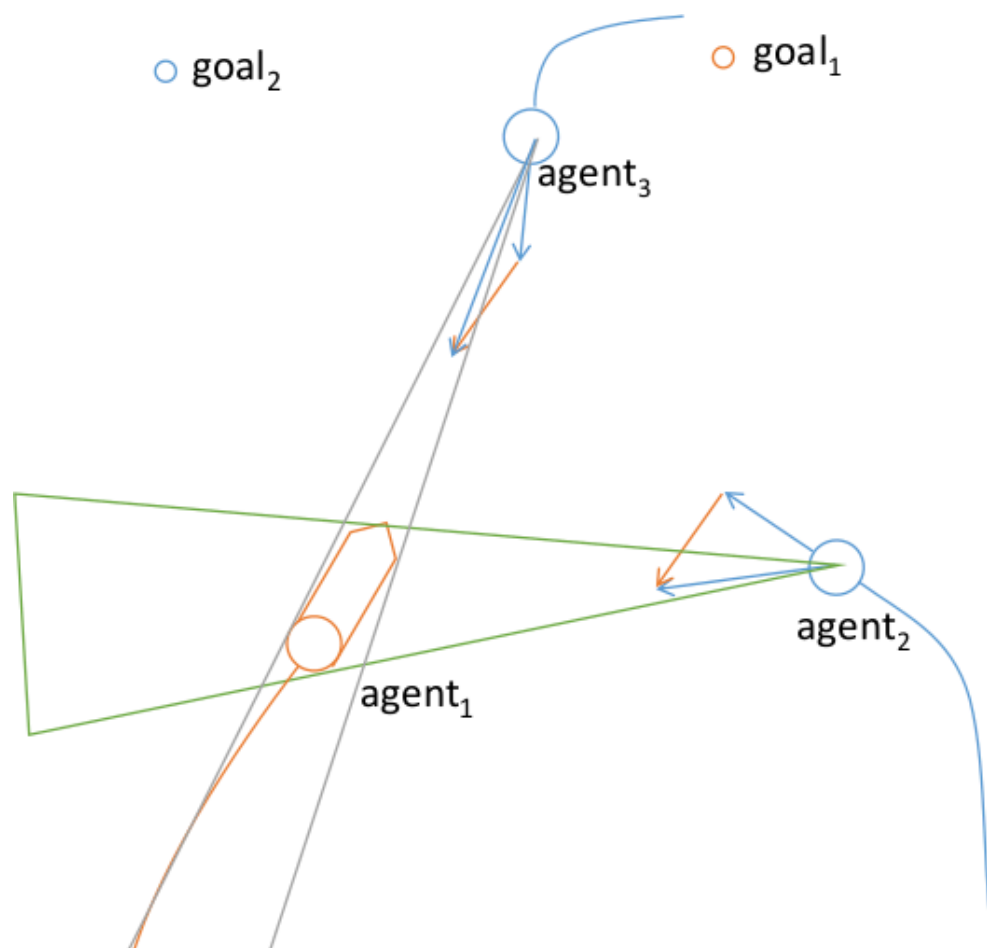
Modèle de Boids - Extensions

Amélioration du calcul des forces

- Calcul des forces basé sur la vision (angle, vitesse) : angle de vue limité

Amélioration de l'évitement des collisions

- Basé sur la vitesse



[*Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers.* J. Pettré et al. SCA 2009]

Déplacement basé sur la vision

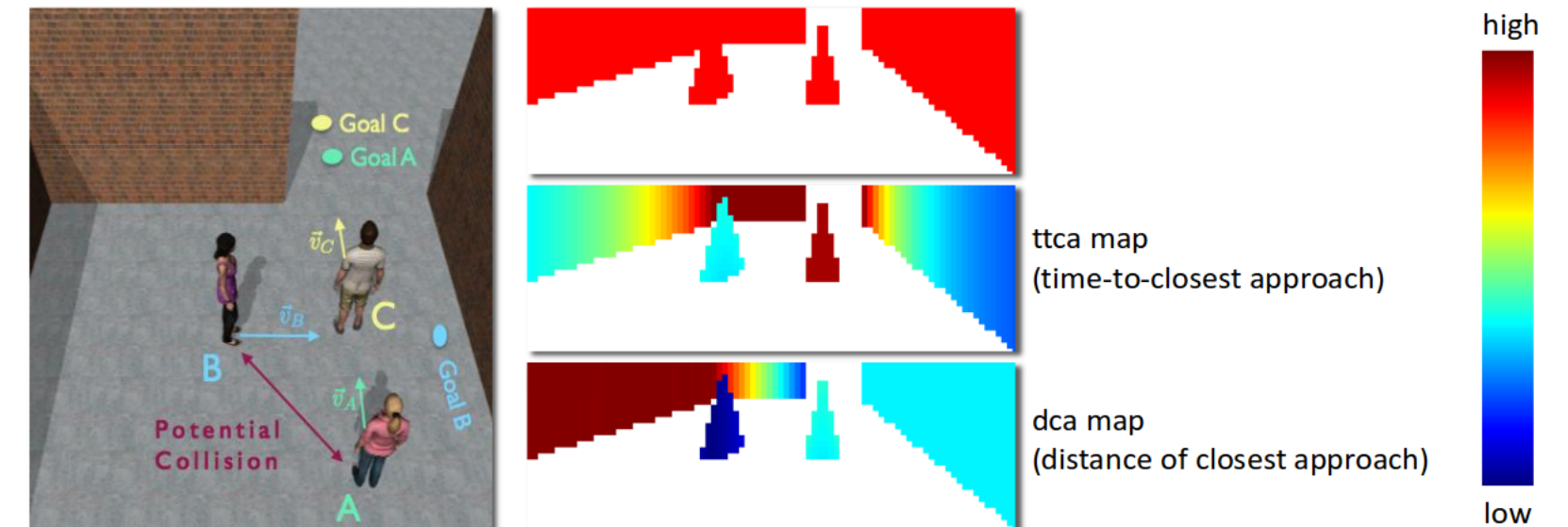
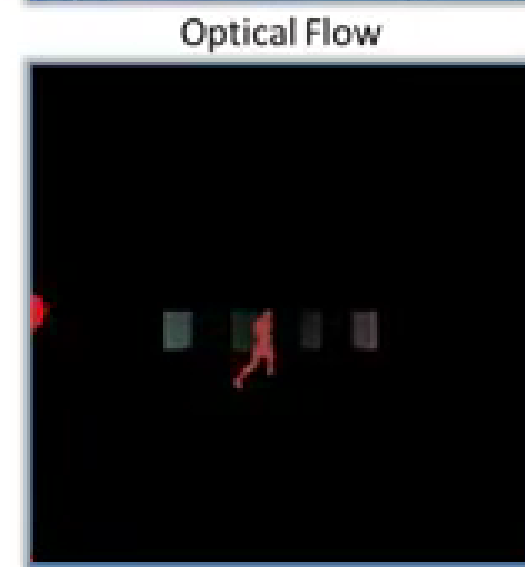
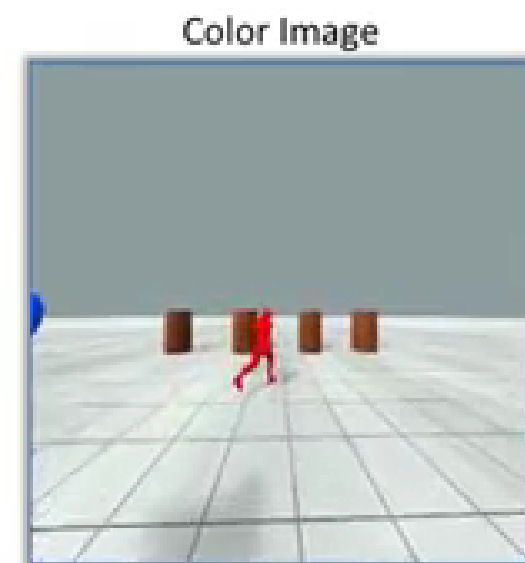
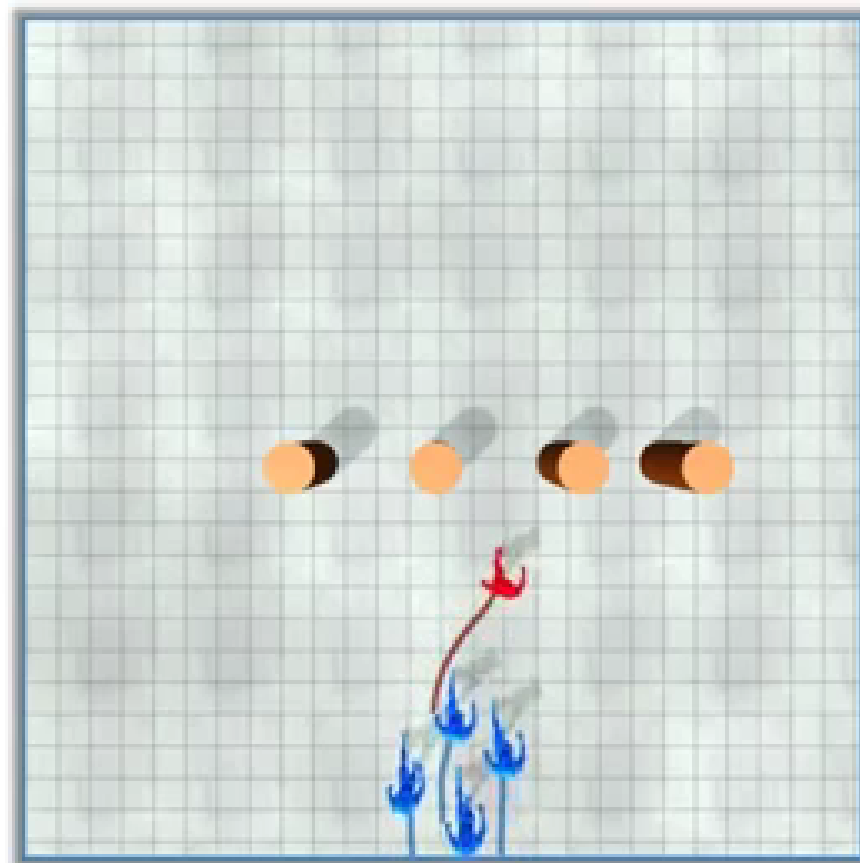
Rendu de la vision pour chaque agent individuel

Calcul du flux optique u .

Les obstacles s'approchent lorsque $\text{div}(u) > 0$

Meilleurs résultats, mais coût de calcul élevé.

Following with Obstacles



[A Synthetic-Vision Based Steering Approach for Crowd Simulation. J. Ondrej et al. SIGGRAPH 2010]

[Character navigation in dynamic environments based on optical flow. A. Lopez et al. EG 2019]

Exemple de logiciel de simulation de foule

Golaem

À Rennes

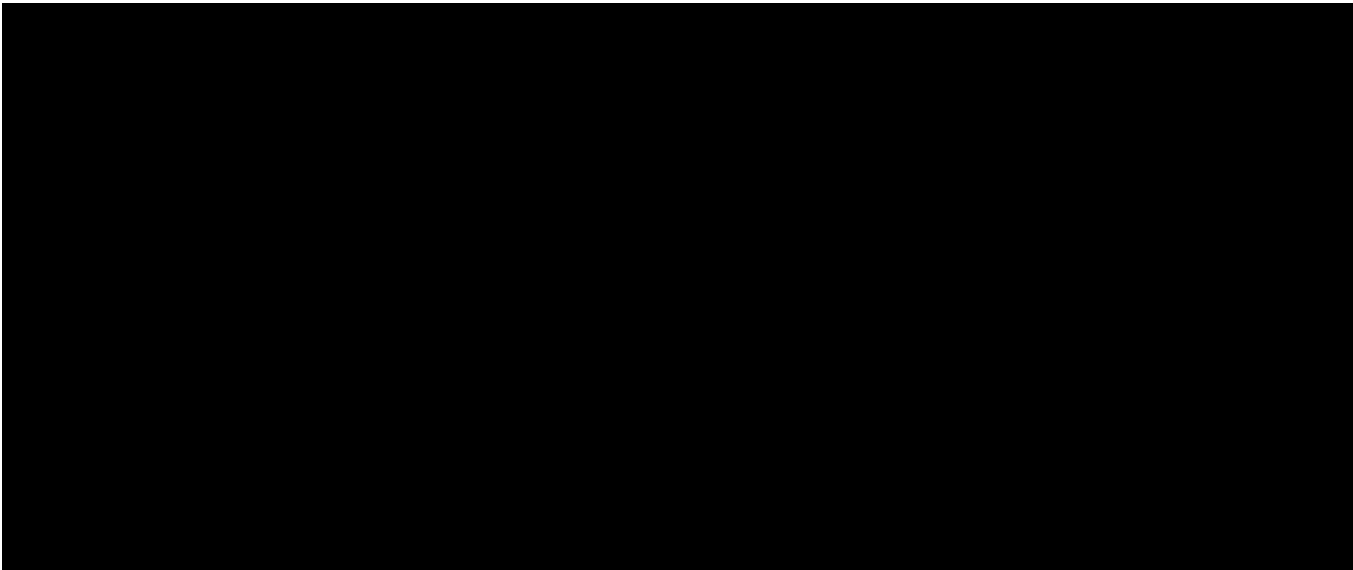
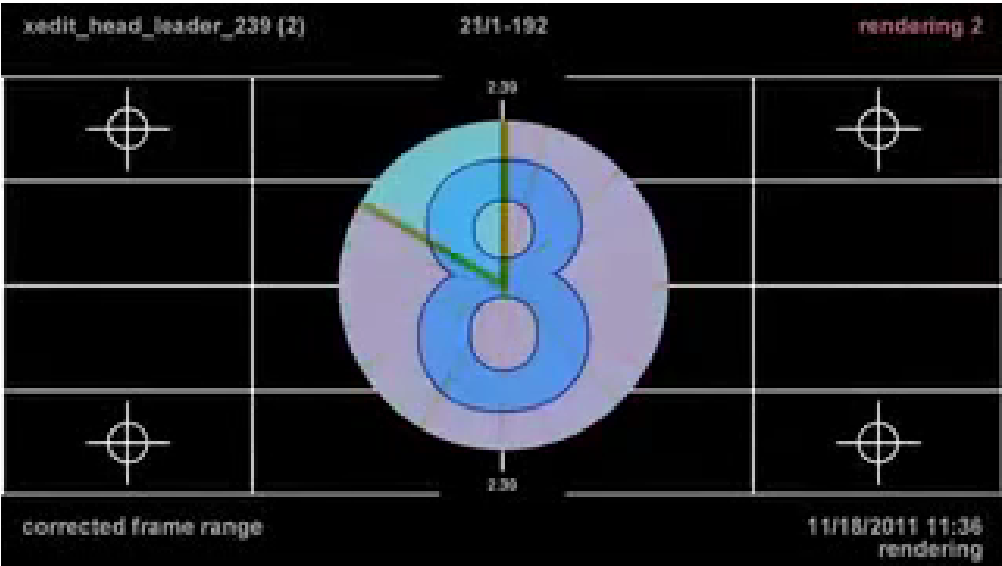
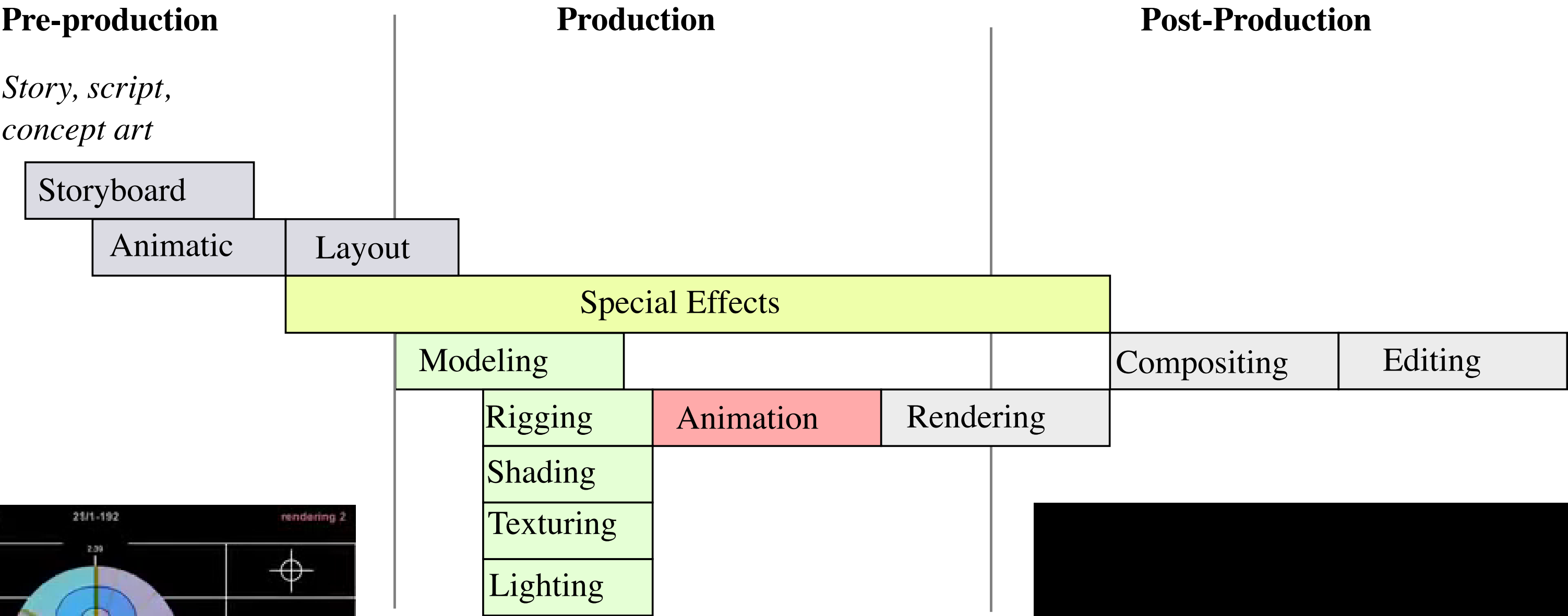


Pipeline de Production

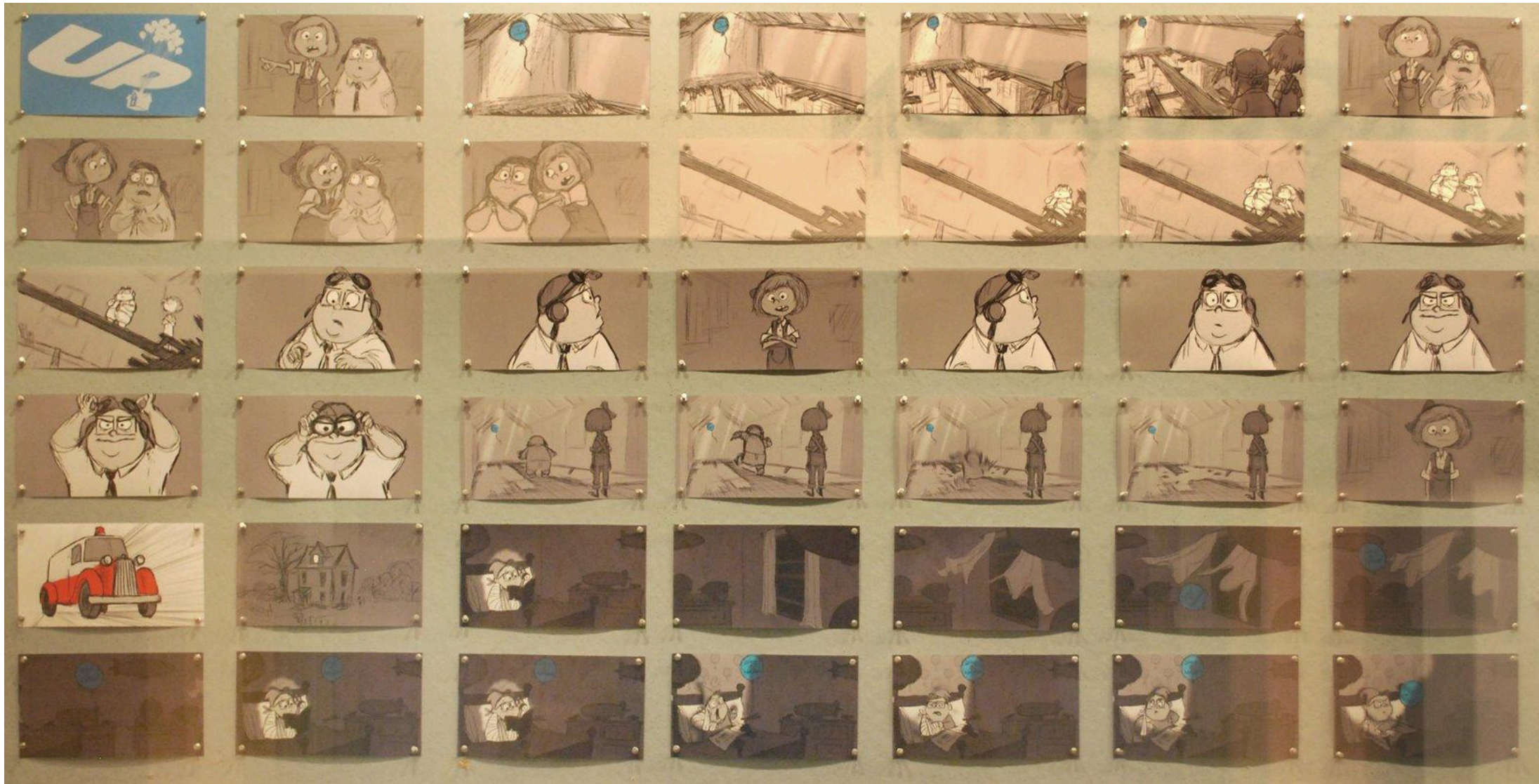
Cas de la production de cinéma d'animation

Pipeline de Production

Structure fondamentale pour la production de films VFX/animation.



Story-board



Quelques dessins 2D : expriment l'histoire

Avant principalement non technique - basé sur l'art/ la créativité

Actuellement augmentation des recherches liées au Storytelling en infographie

ex. *Disney Research Studios*

Animatic

≈ Story-board animé, divers formats

Sens approximatif du timing, visuel, action: détails pas nécessairement suivis précisément dans la version finale.



Mise en page : Passage en 3D

1ère étape : Modélisation 3D approximative et placement de

- Caméra : champ visuel, perspective

La 3D est beaucoup plus contraignante que les dessins 2D

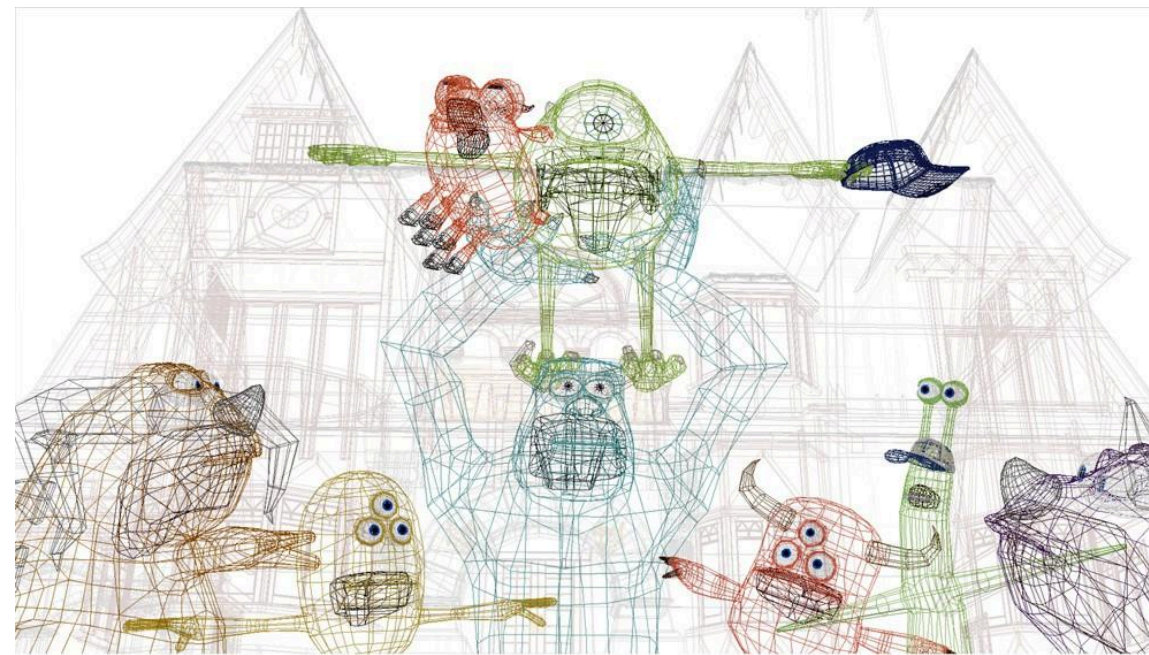
- Volumes des formes, continuité

Pas de détails : visage, etc.

Choix entre éléments 3D/2D



"MONSTERS UNIVERSITY" Progression Image 1 of 6: STORY
©2013 Disney•Pixar. All Rights Reserved.



"MONSTERS UNIVERSITY" Progression Image 3 of 6: MODELING
©2013 Disney•Pixar. All Rights Reserved.



"MONSTERS UNIVERSITY" Progression Image 4 of 6: LAYOUT
©2013 Disney•Pixar. All Rights Reserved.

Modélisation 3D

En production

- Modélisation par maillage polygonal

Approche du grossier au fin

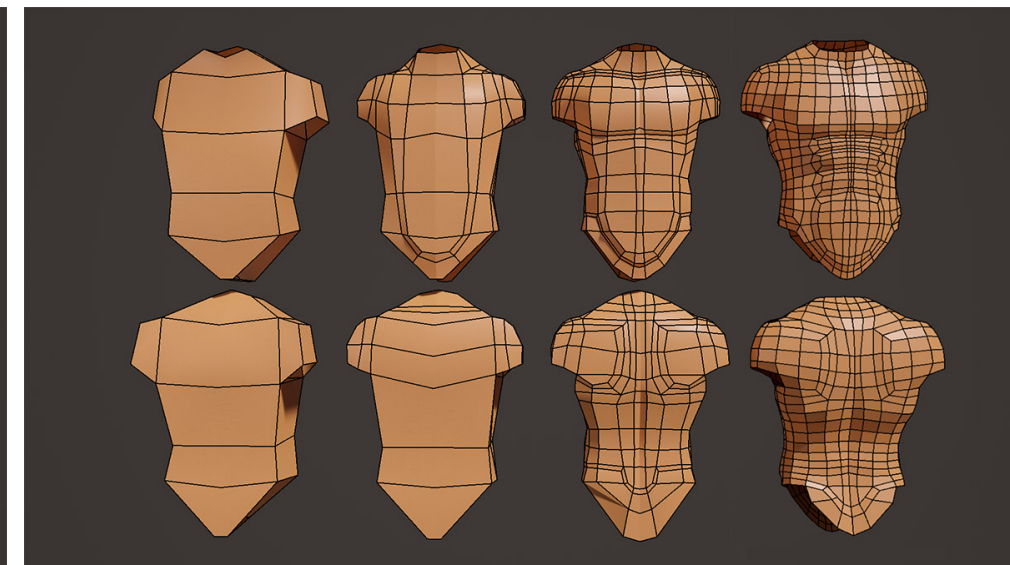
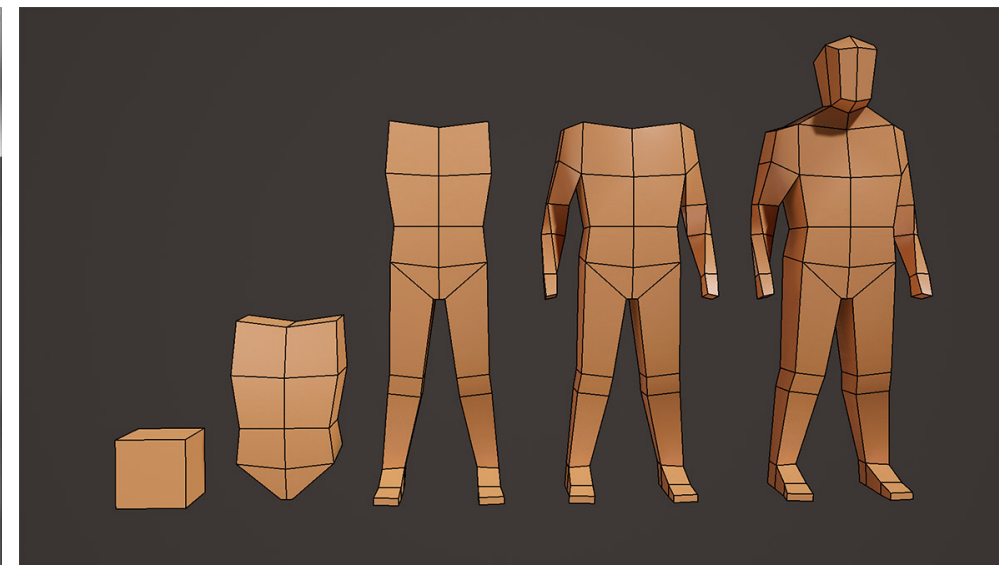
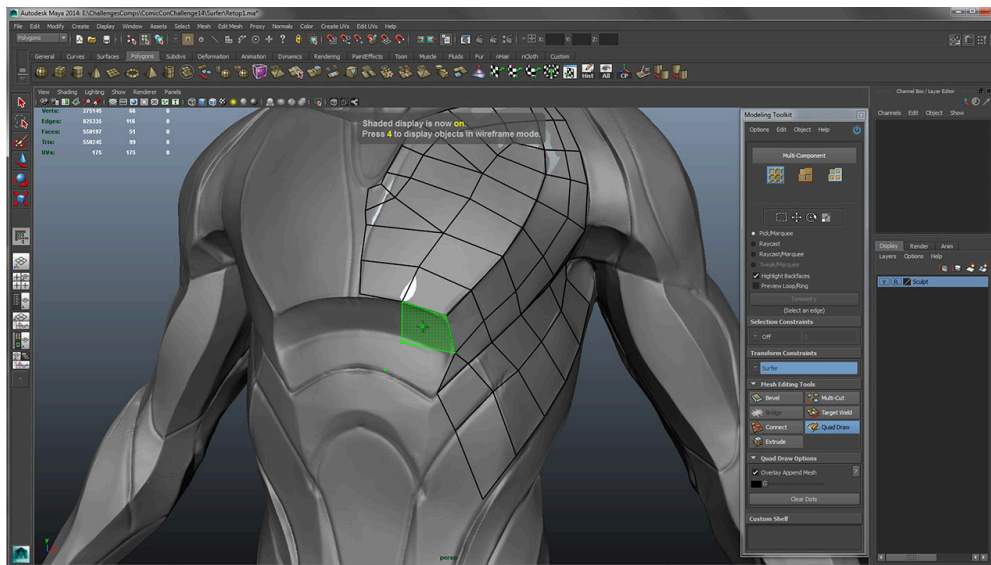
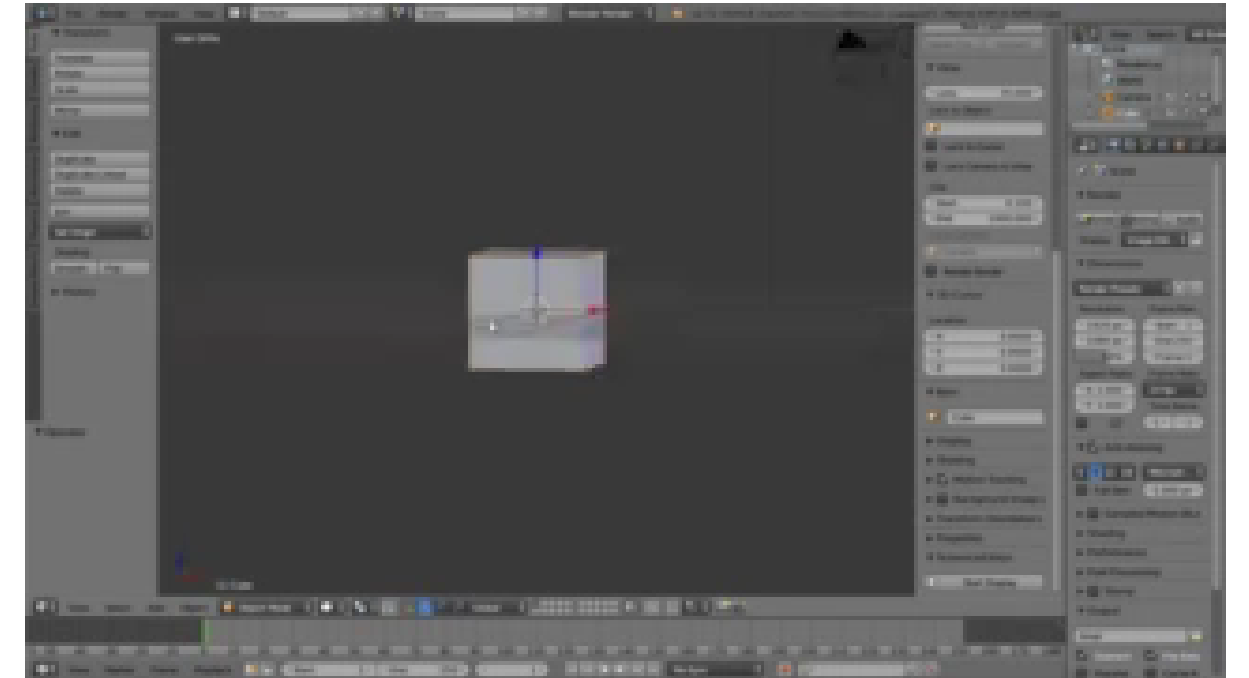
1. *Modélisation basse résolution (extrusion)*

2. *Subdivision, Affinement*

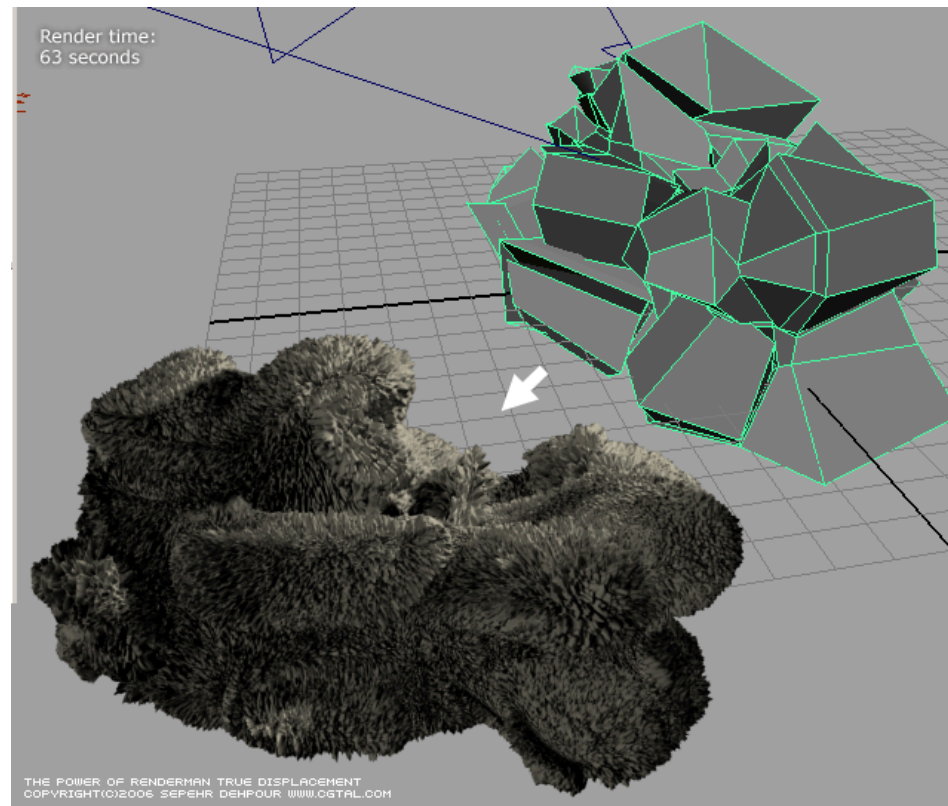
- Modélisation paramétrique (NURBS)

Tout le reste est du "VFX"

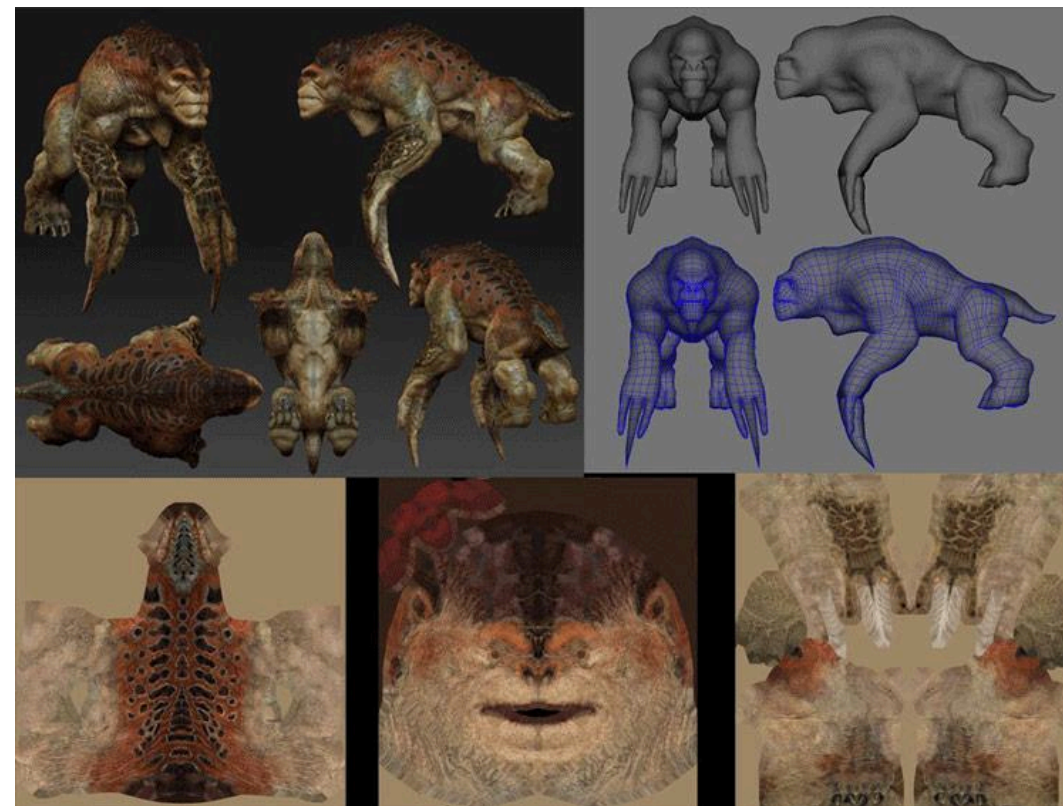
Outils courants : Maya, 3DSMax, Cinema4D, Blender, etc.



Modélisation de l'apparence - But du rendu



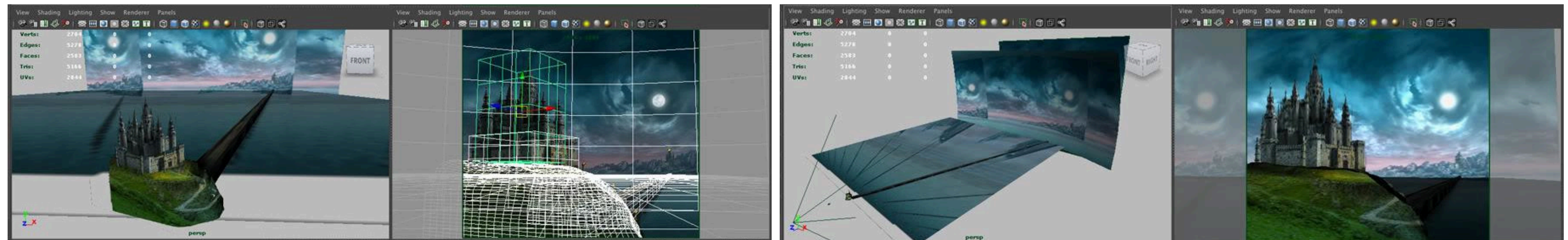
Shading



Texturing



Lighting



Material painting

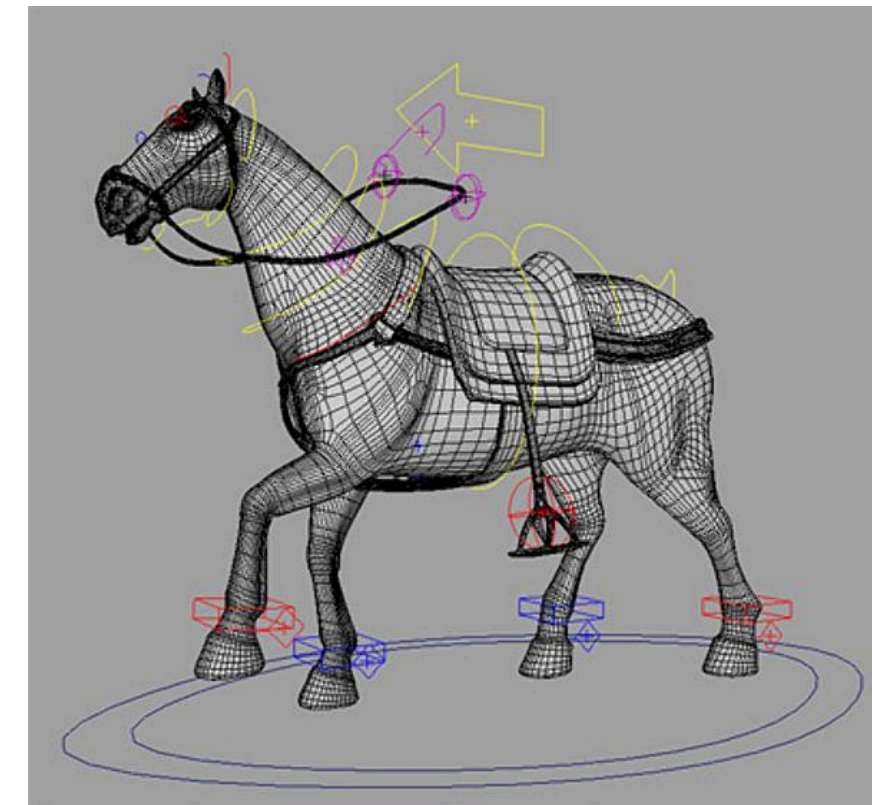
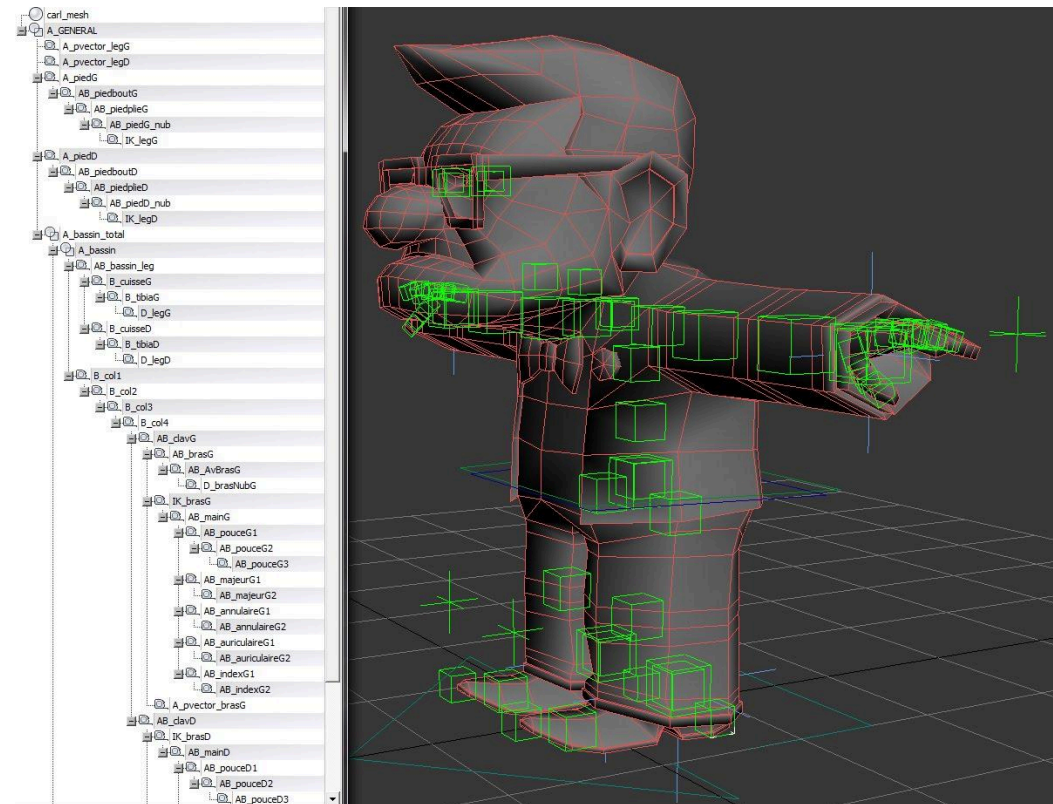
Rigging

Attacher des poignées de déformation au maillage

Chaque poignée (contrôleur) est associée à une déformation - degrés de liberté

Le rigging est une partie technique

Script Python, Mel (Maya), Lua, etc.

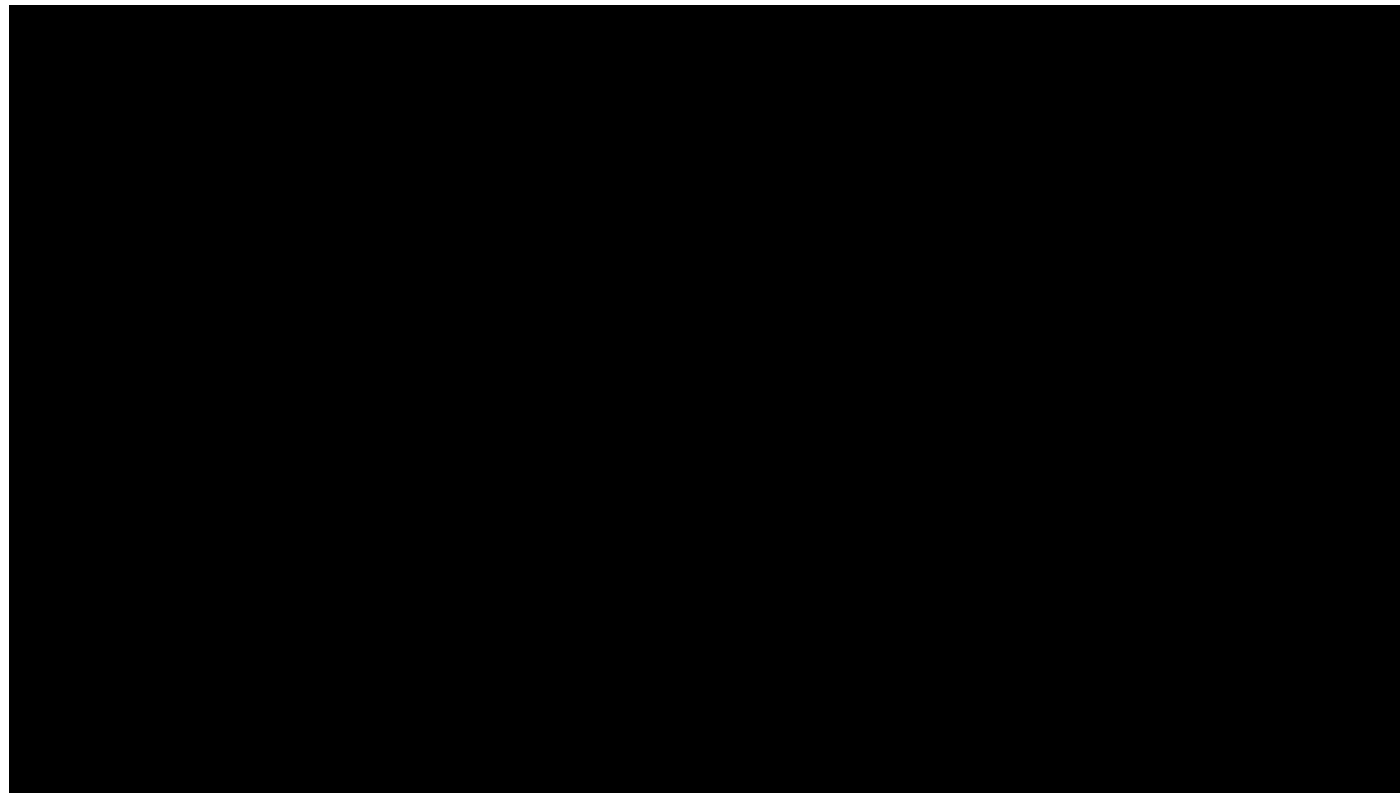


Animation

En terminologie de production : Animation = Animation par pose clé de personnage riggé

Définir les courbes d'animation sur les contrôleurs du rig

Tout le reste est VFX



Animation du cycle de marche (×40)



Résultat

Jusqu'à 75 % des artistes dans les studios de production sont des animateurs

Animation = L'élément clé - coût le plus élevé - pour les studios de production

Un animateur → 1-10 s d'animation par jour

Sous-parties de l'animation

1- Poser les images clés

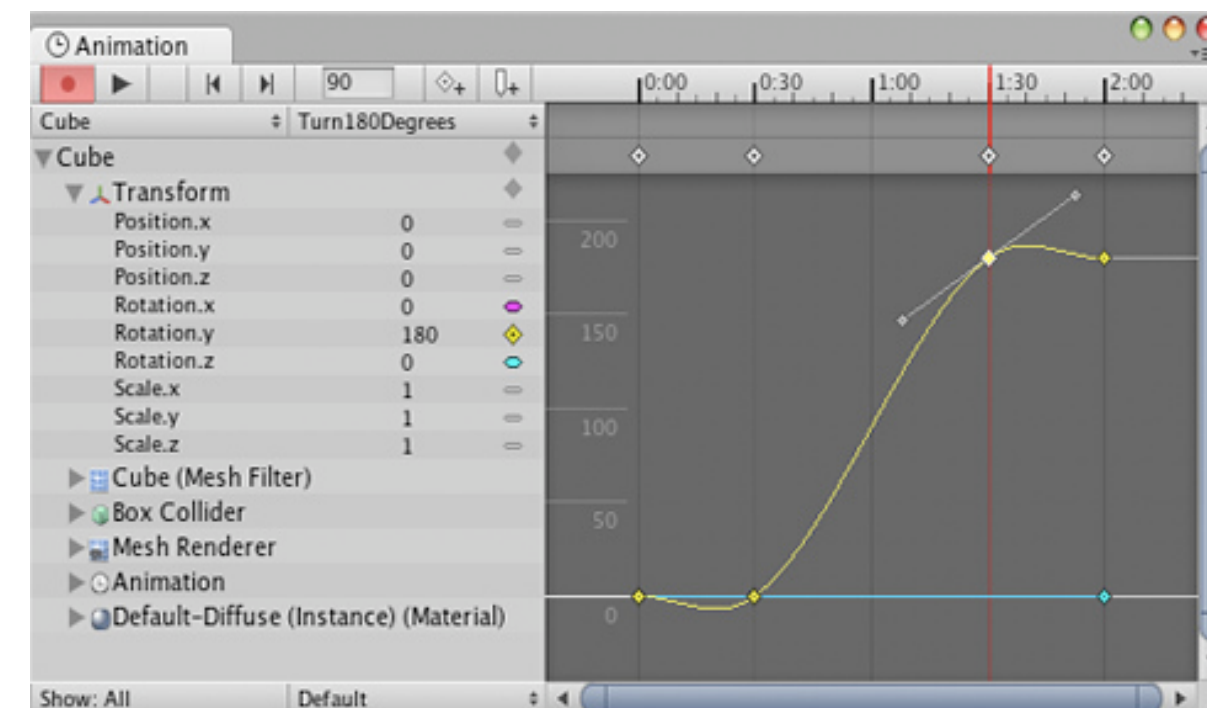
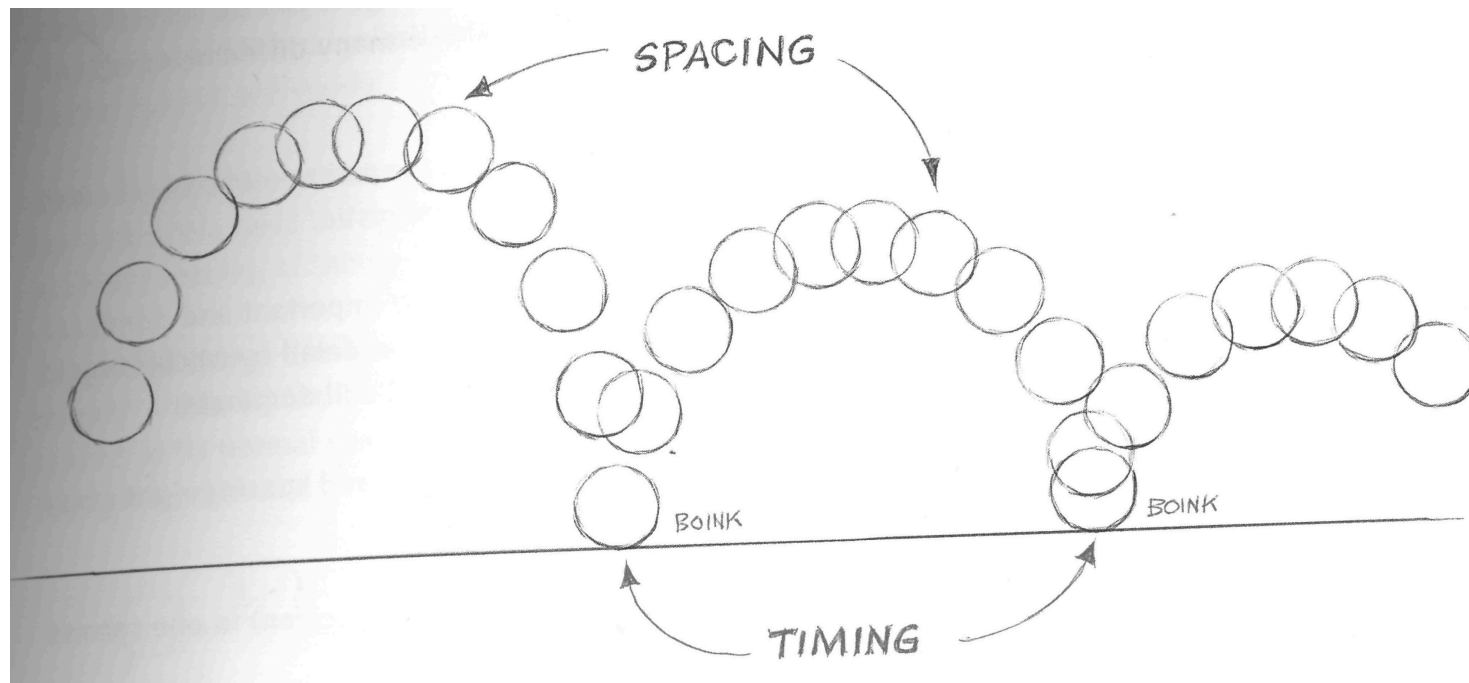
Définir la posture générale principale du personnage

*Lié à la **déformation géométrique** du personnage, le temps n'est pas impliqué*



2- Animer les images intermédiaires

- **Timing** : Placer les images clés à des moments spécifiques (durée globale d'une action)
- **Spacing** : Vitesse de l'interpolation (dynamique de l'action)



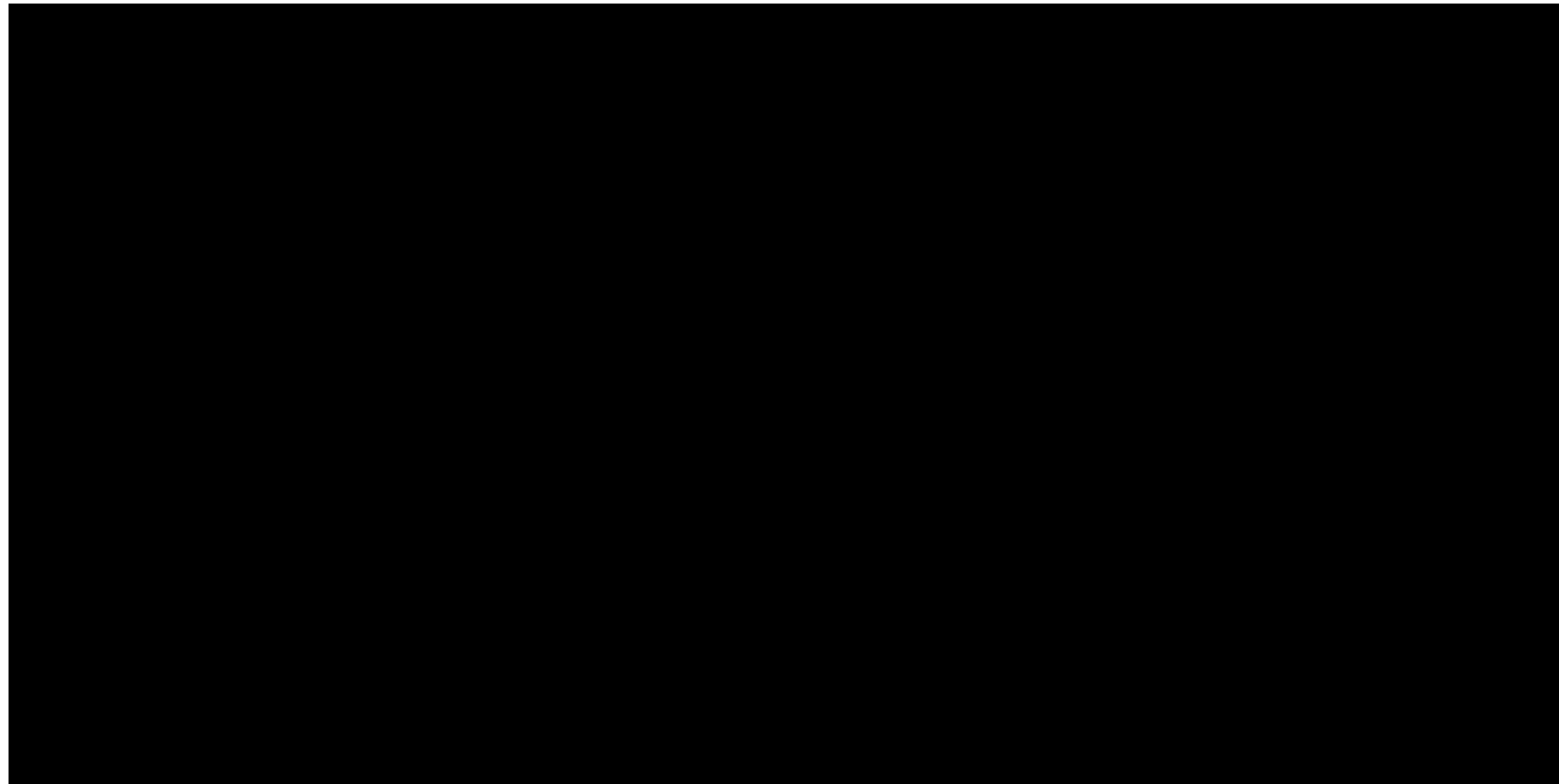
Effets Spéciaux (VFX)

Tout ce qui n'est pas pris en charge par la modélisation / rigging / animation traditionnelle

Physique (explosions, fluides, cheveux dynamiques, tissus, ...), systèmes de particules, formes complexes, foule, etc.

Partie R&D technique : Un élément peut conduire au développement d'un système dédié.

Logiciel principal : Houdini (SideFX)



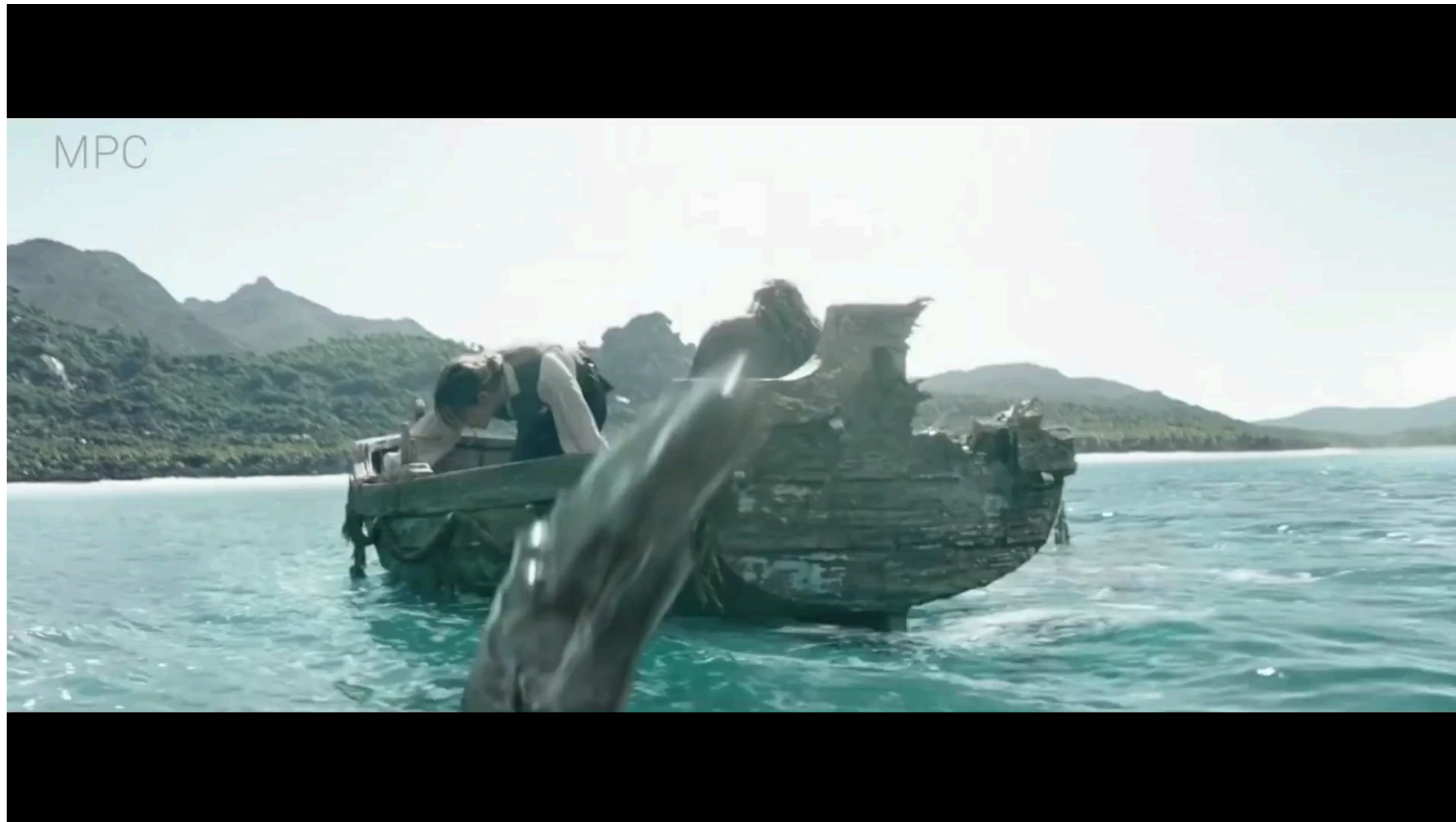
Post Production

Compositing

Fusionner toutes les couches : rendues et réelles

Note : Rendu des couches de couleur mais aussi de la profondeur et des normales.

Logiciel principal : Nuke (Foundry)

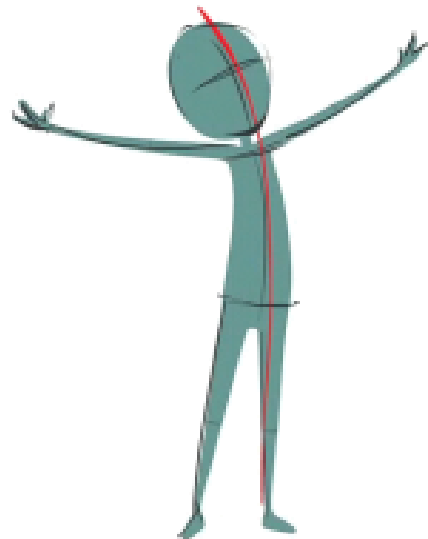


Animation expressive

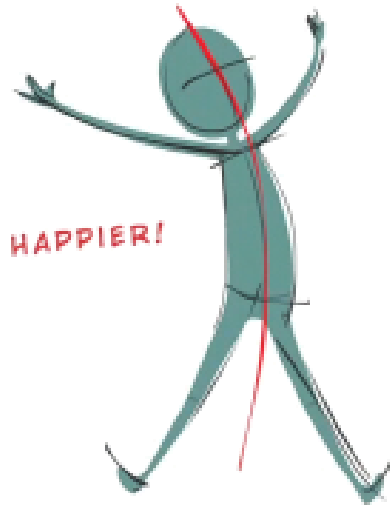
Pose d'animation de personnage - Ligne d'action

- Ligne d'action : *Axe médian* exprimant la pose du personnage
- Exprimer *statiquement* la dynamique de l'action
 - Pose instable $\Rightarrow$ Action/mouvement dynamique

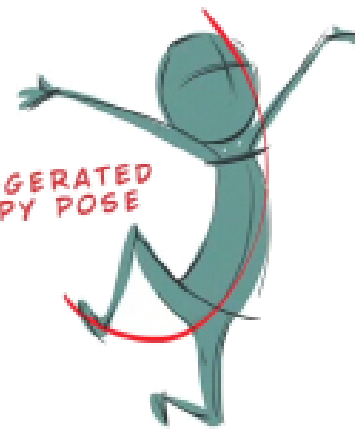
SLIGHT CURVE



HAPPIER!

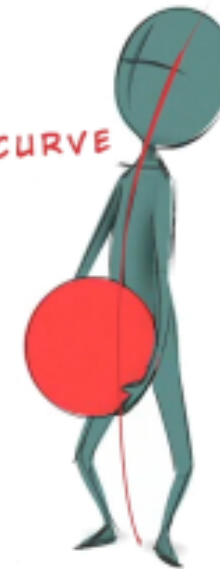


EXAGGERATED
HAPPY POSE



SLIGHT CURVE

SEEMS LIGHTER



EXAGGERATED EFFORT POSE

SEEMS HEAVIER

ACCENTUATED
CURVE



SLIGHT CURVE



ACCENTUATED
CURVE



EXAGGERATED
SAD POSE

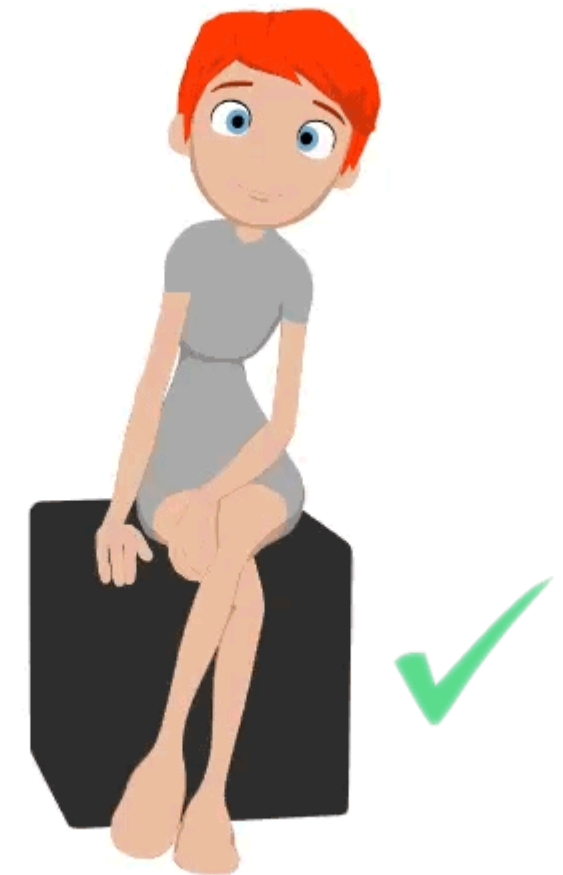


(C) Chiara Porri

Principes d'animations

- L'interpolation entre des poses réalistes ne suffit pas pour une animation expressive
- *12 principes d'animation* par Disney *Illusion of Life*, 1981

1. Timing
2. Spacing
3. Slow-in, Slow-out
4. Squash & Stretch
5. Anticipation
6. Follow Through
7. Secondary Action
8. Exaggeration
9. Appeal
10. Arcs
11. Staging
12. Straight Ahead/Pose to Pose

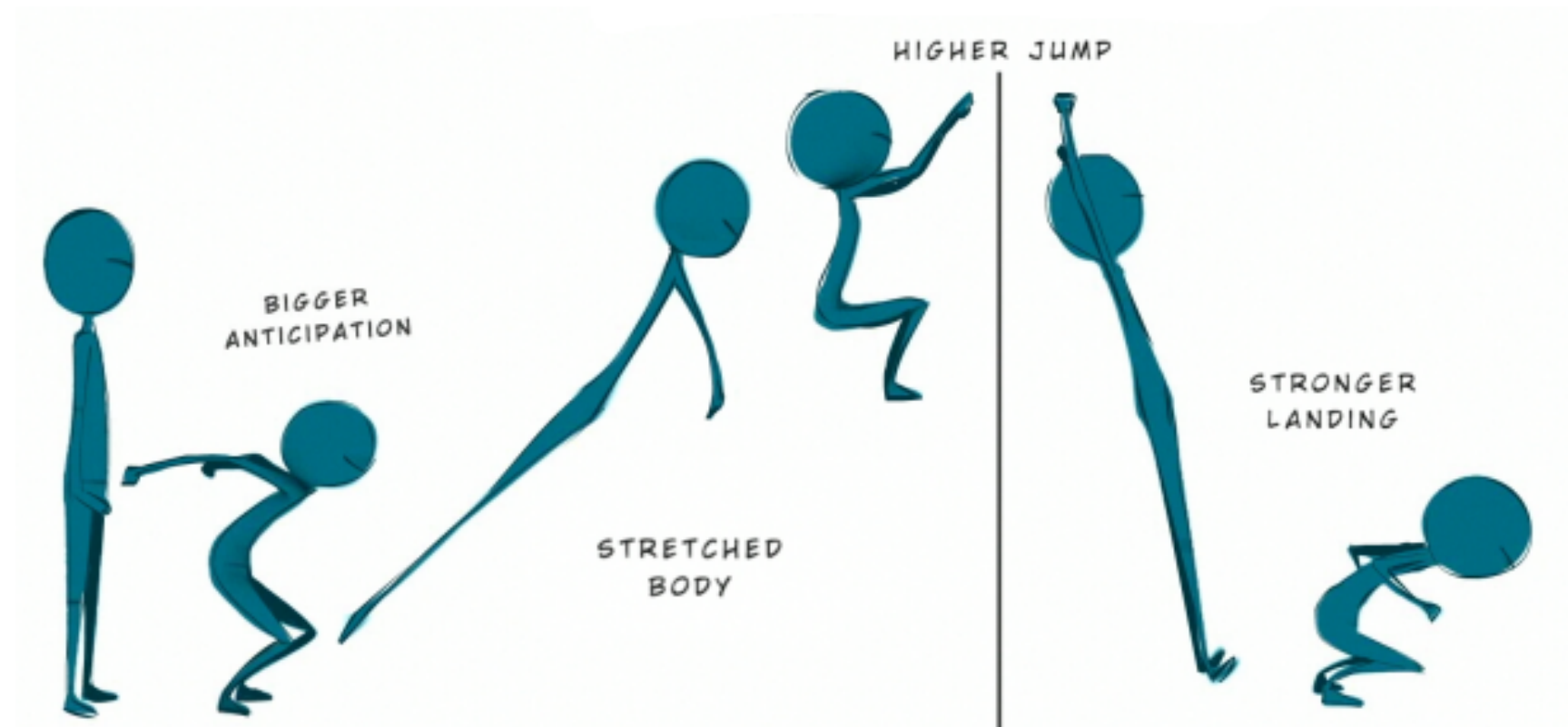
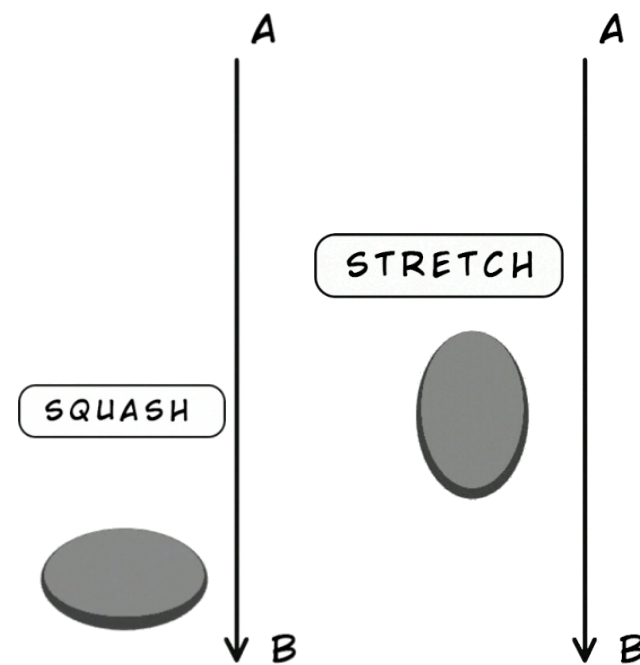


(C) Chiara Porri

Animation expressive

Squash & Stretch

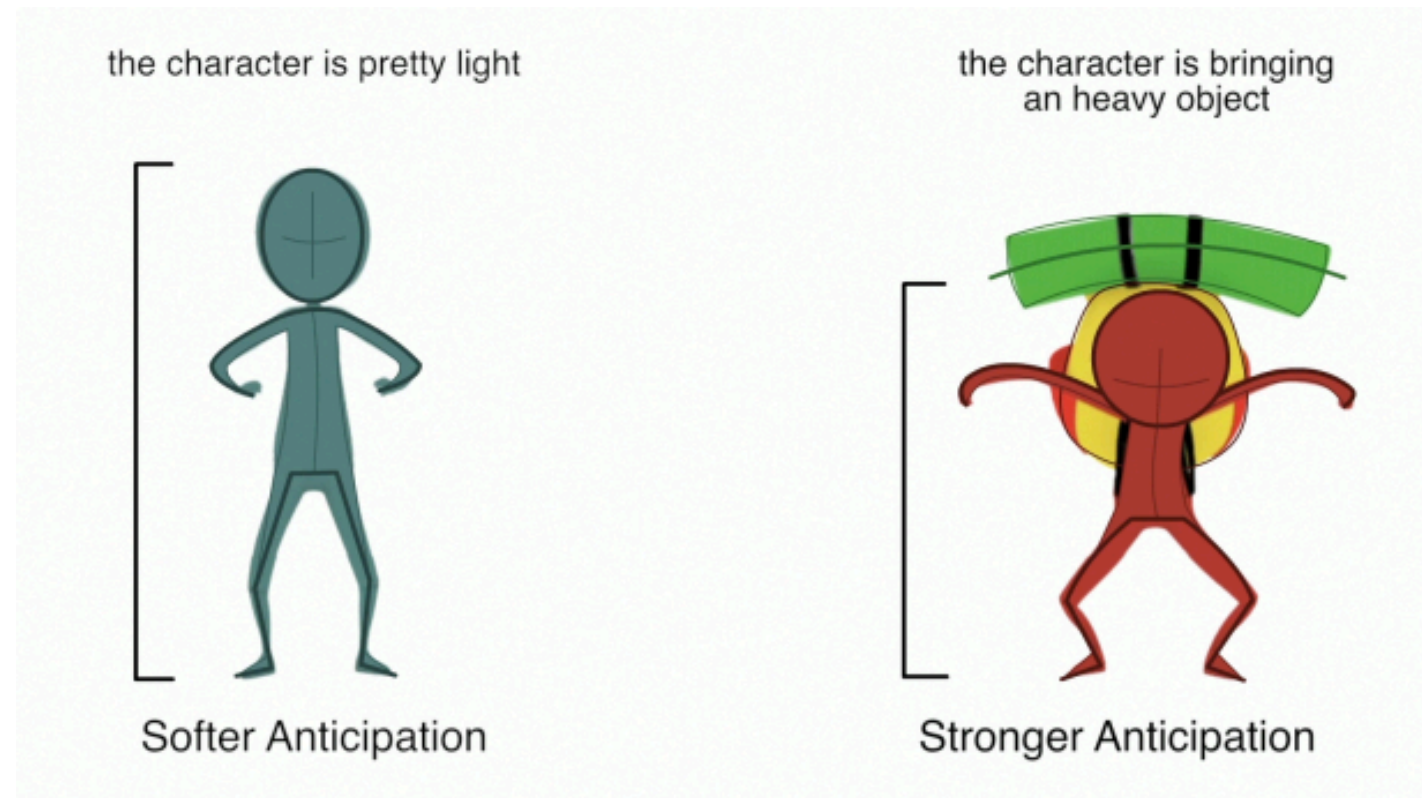
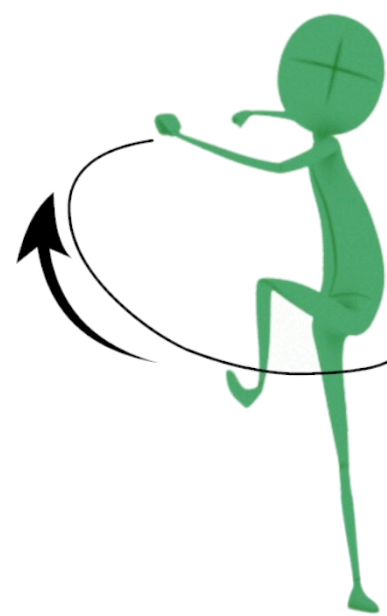
- Très courant dans les dessins animés
- Irréaliste, mais étonnamment *plausible*



(C) Chiara Porri

Animation expressive

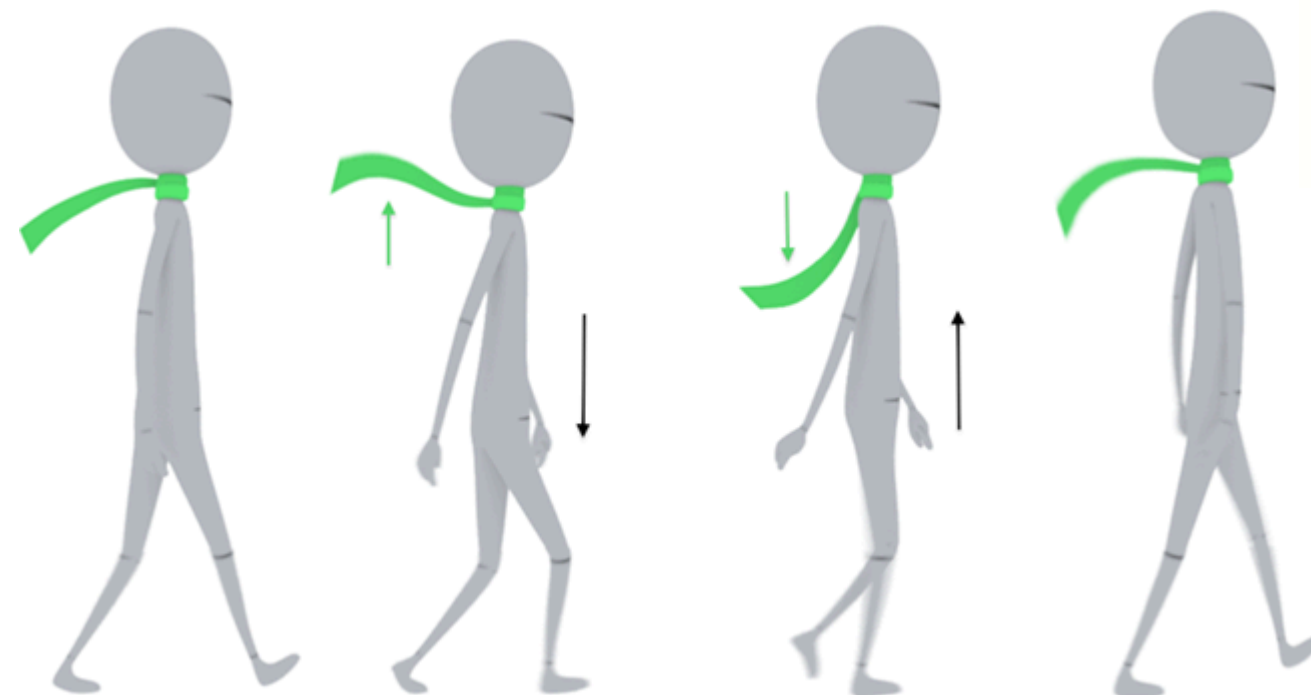
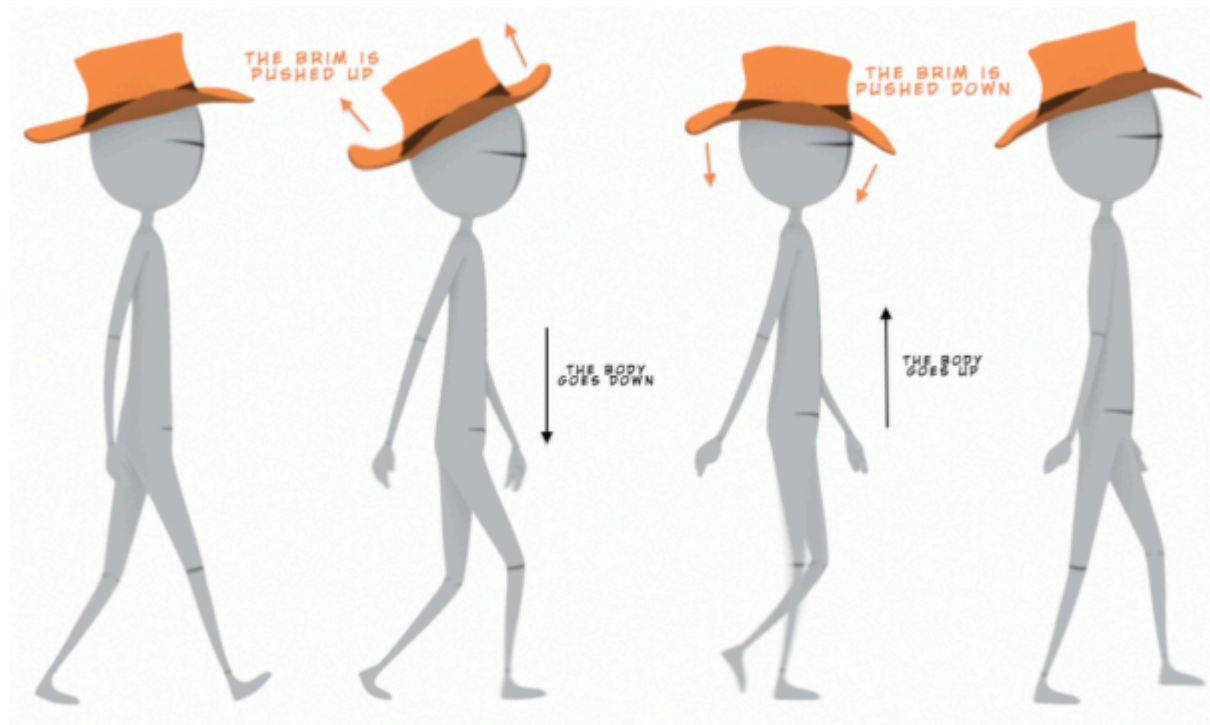
Anticipation



Animation expressive

Suivi / Mouvements secondaires

Follow-through



(C) Chiara Porri