

IFT3355: Infographie Animation

© Pierre Poulin

Dép. I.R.O.

Université de Montréal

Animation - Mouvement dans le temps

- Objet:
 - position / orientation
 - déformation
 - couleur, transparence, texture
 - croissance, apparition, disparition
 - liens entre les objets

Animation - Mouvement dans le temps

- Lumière:
 - position / orientation
 - intensité, distribution
 - mêmes paramètres que ceux des objets

Animation - Mouvement dans le temps

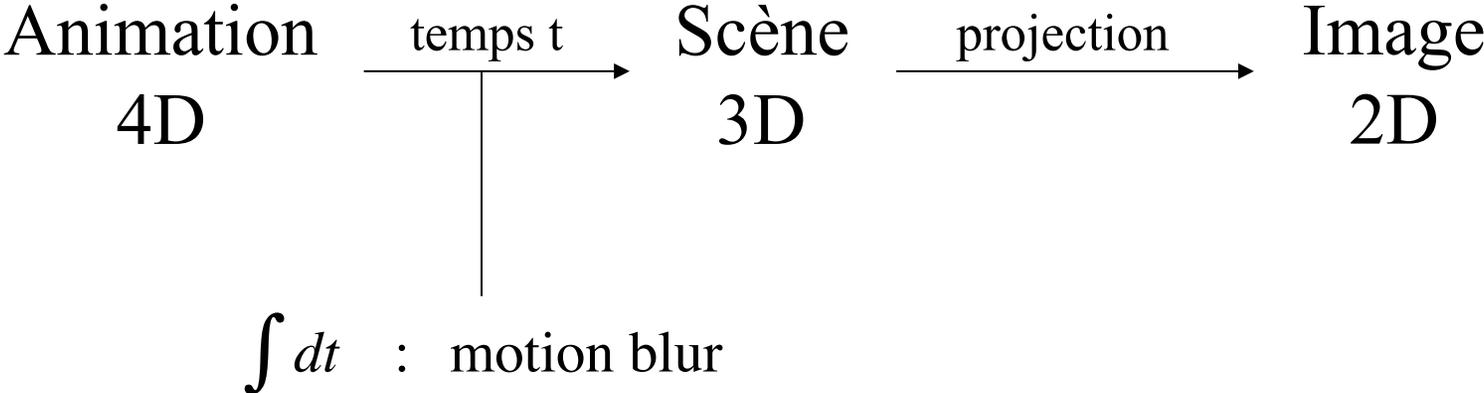
- Caméra:
 - position / orientation
 - angle de vue (pyramide)
 - résolution
 - profondeur de champ

Animation par ordinateur

+ tout est virtuel:

- pas de caméra dans un miroir
- pas d'ombres de microphone
- mouvements et tailles impossibles
- simulateurs de vol sans aucun danger
- effets spéciaux sans dégâts
- pleine liberté des points de vue

De 4D à 2D



Animer un modèle 3D

- Chaque modèle contient un certain nombre de paramètres
 - position (x,y,z) de ses points
 - orientations (θ,ϕ) relatives de ses parties
 - couleur, etc.
- Animer un modèle correspond donc à donner à tous ses paramètres des valeurs dans le temps
- Autrement dit, un modèle animé possède un *état* pour ses n paramètres, ce qui correspond à un point dans un espace d'états en n -D

Animer un modèle 3D

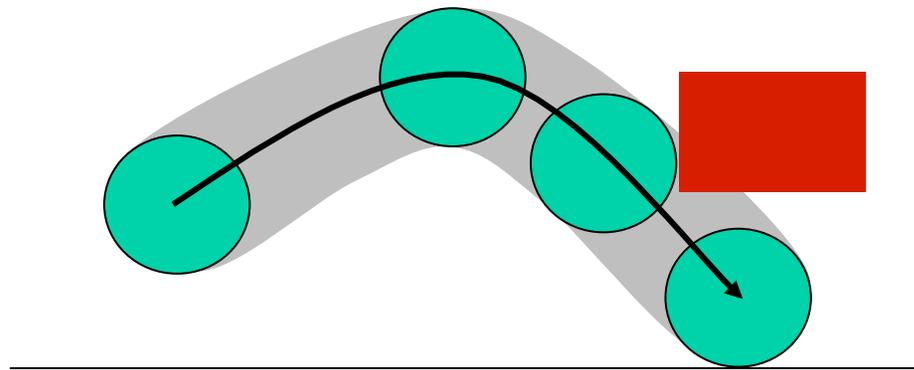
- Le mouvement correspond à la trajectoire d'un paramètre dans le temps
- Animer un modèle consiste donc à spécifier une trajectoire dans l'espace d'états du modèle
- Un modèle et son animation sont donc intimement liés:
 - modélisation définit les contrôles
 - animation utilise ces contrôles
- Modèle hiérarchique permet de contrôler hiérarchiquement son animation

Illusion de mouvement

- Affiche plusieurs images par seconde
- Persistance de la vision humaine
 - intégration des images en une sensation de continuité visuelle
 - moins de 40 Hz produit du *flicker*
 - film: 24 frames par sec. (48 Hz, 2x même image)
 - grand écran brillant dans un environnement sombre
 - vidéo: 30 frames par sec. + *interlaced* (60 Hz)
 - infographie: 30-60-72 frames par sec.

Cohérence temporelle

- Plusieurs aspects des cohérences 2D/3D peuvent être étendus à la cohérence temporelle
 - collision entre deux volumes évoluant dans le temps
 - position près de la position au temps précédent
 - etc.

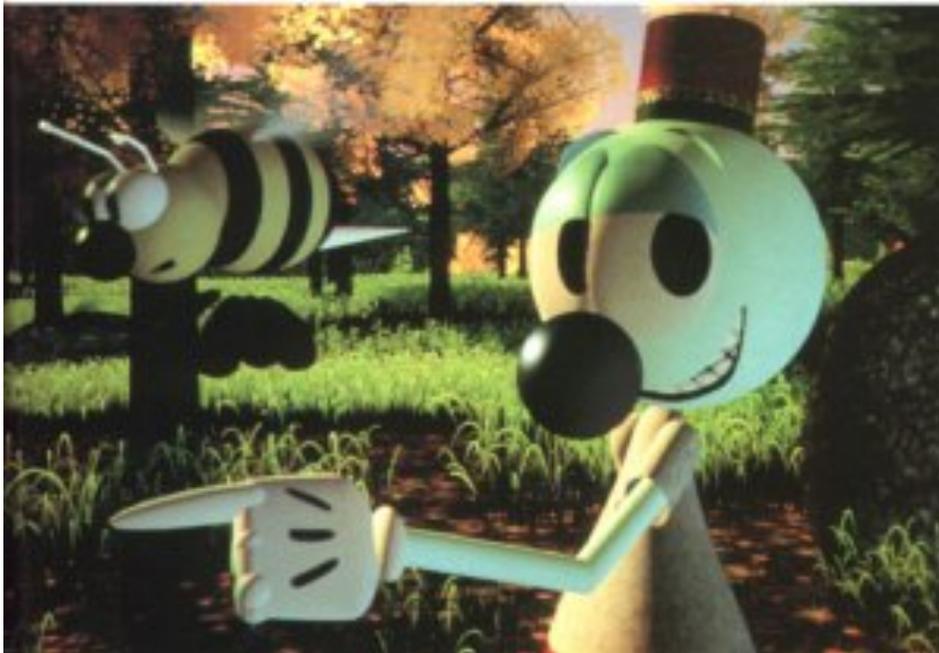


Aliassage temporel

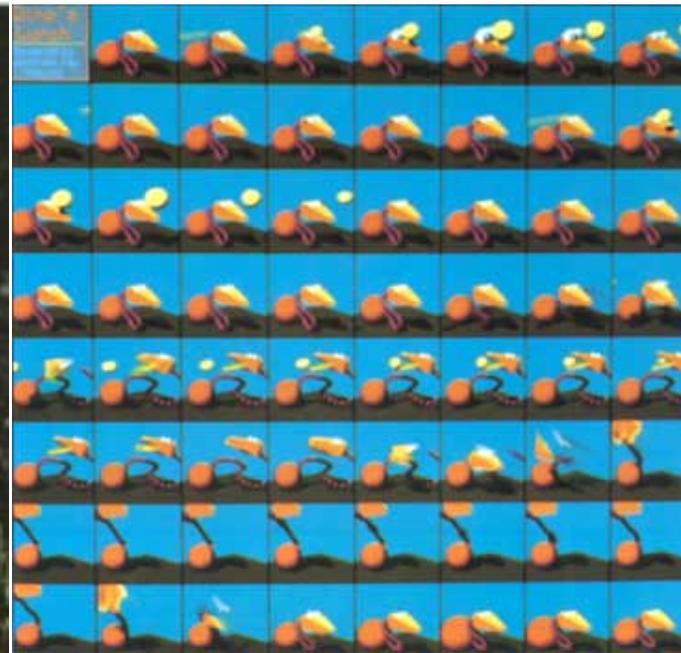
- Changements trop rapides relativement au nombre d'images affichées par seconde
 - *strobing*: disco et roue/hélice qui tourne à l'envers

Aliassage temporel

- Filtre temporel appliqué sur un pixel pour plusieurs images consécutives
 - flou de mouvement (*motion blur*)



Lucasfilm

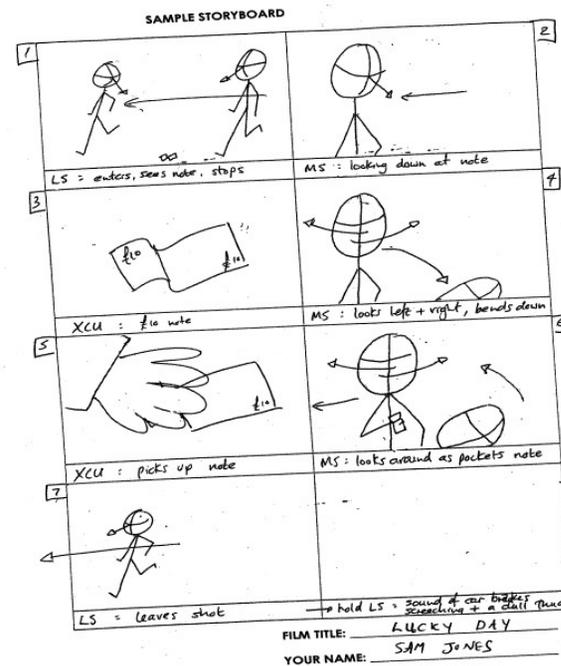


Glassner



Animation traditionnelle

- *Story board*
 - lignes directrices de l'histoire en esquisses

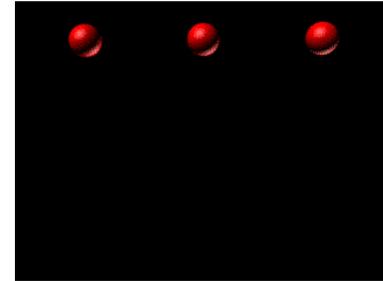


Animation traditionnelle

- Bande sonore
 - paroles et musique
- *Keyframes*
 - positions (extrêmes) des actions
- Images intermédiaires (*in-betweens*)
- *Painting*
- Assemblage des transparents (*cels*) et photographie

Les principes de l'animation traditionnelle

- *Squash and stretch*
 - Rigidité et masse d'un objet en mouvement
- *Slow in - slow out, and arcs*
 - Modifier l'espacement entre les images consécutives
- *Straight ahead vs. pose to pose*
 - Dessiner au fur et à mesure, ou in-between
- *Timing, secondary actions, anticipation, and staging*
- *Follow through, overlapping action*
 - Lancer de balle, commencer une action avant la fin de la précédente (aucun temps mort)
- *Exaggeration and appeal*



L'infographie dans l'animation traditionnelle

- *Painting*
 - numérisation des dessins
 - souris ou tablette
 - digitaliseur optique (*scanner*)
 - dessins faits par ordinateur (plus rare)
 - filtrage pour en adoucir les contours
 - coloriage par remplissage à germes
 - souvent la partie la plus longue si produite à la main
 - composition des divers plans
 - transfert sur film

L'infographie dans l'animation traditionnelle

- Génération des images intermédiaires
 - interpolation des formes



La Faim - ONF 1973

L'infographie dans l'animation traditionnelle

- Génération des images intermédiaires
 - interpolation des images
- *fade in - fade out*
- *morphing*

Morphing

- Associe des points et des courbes entre deux images (mise en correspondances)
- Crée un maillage sur chaque image (ou une fonction de distance) selon les correspondances
- Pour toute image intermédiaire, interpole la position des correspondances et les couleurs
- *Inverse warping*
- *Morphing* de modèles 3D n'est pas encore très populaire (quoique les FFD le sont)

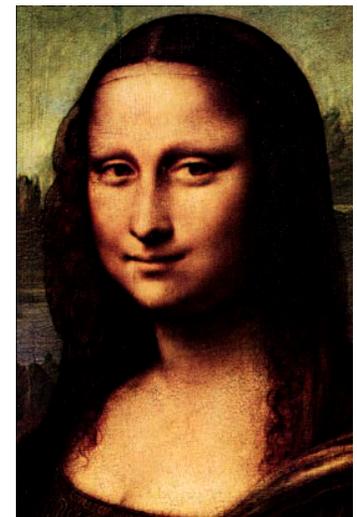
Exemples de *Morphing*



Free Morphing 2.1



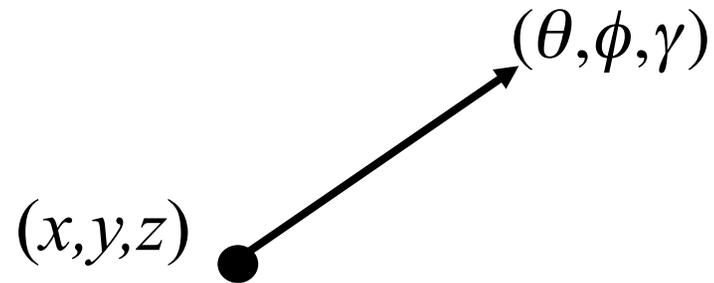
Black or white - PDI



Seitz

Animation 3D - Degrés de liberté (*DOF*)

- Animer un objet rigide (*flying logo*) de 6 *DOF* à 30 frames/sec pendant 5 sec : 900 chiffres



- Figure humaine possède plus de 200 *DOF*

Degrés de liberté

- Nombre de variables pour spécifier un état
- Vecteur d'états
 - ensemble de variables définissant toutes les configurations possibles
 - forme une base dans l'espace d'états
 - animation: chemin dans l'espace d'états
- Exemple pour un solide:

$$E = (x, y, z, \theta, \phi, \gamma)$$

Techniques d'animation

- Bas niveau
 - script
 - *keyframe*
 - interpolation (spline)
- Capture de mouvements (*dataglove*, senseurs)
- Cinématique inverse (et directe)
- Hiérarchies et automatismes

Techniques d'animation

- Animations procédurales
 - simulation dynamique
 - gravité, déformations, friction
 - systèmes de particules
 - herbe, fumée, chutes d'eau
 - effets secondaires (tissus)
 - animation comportementale (*flocking*)
 - perception locale
 - ensemble de règles

Animation par scripts

- Populaire lorsque les systèmes n'étaient pas interactifs

1. Listes linéaires d'événements

Frame	Table	Action	Axe	Angle
Début - Fin	Interpolation			
43,54	B	rotate palm	X	30
50,77	D	translate balle		

Animation par scripts

2. Fonctions dans un langage de programmation standard

- + flexible (peut tout faire)
- + objets et animations procéduraux
- animateur doit aussi être un programmeur
- *Animator*: contrôle l'animation à un haut niveau
 - active et interrompt les *actors*
 - change les paramètres globaux (ex: horloge)
- *Actor*: procédure exécutée une fois par frame
 - souvent associé à un objet

Animation par scripts

3. Représentation graphique

- Séquence d'actions disposées dans un diagramme en fonction du temps

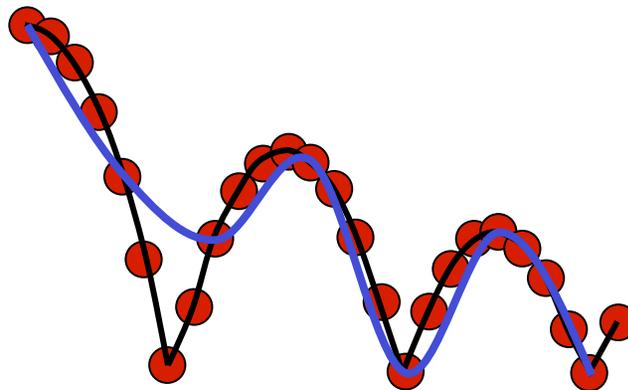
Vol de rêve

Animation par *keyframes*

- Traditionnel chez Disney
 - animateur produit les frames les plus importants pour le mouvement et les formes: *key-frames*
 - dessinateurs produisent les frames intermédiaires: *in-betweens*
- Ordinateur peut parfois produire les *in-betweens* par interpolation, mais c'est souvent trop difficile
- Peut interpoler n'importe quel paramètre: *key-parameters*

Animation par *keyframes*

- *Key-frame* correspond à un modèle dans une configuration donnée
- *In-betweens* correspondent à une interpolation entre les valeurs de chaque paramètre (point en n -D)
- Souvent on utilise des courbes (splines) en n -D, formant une trajectoire dans le domaine d'états du modèle



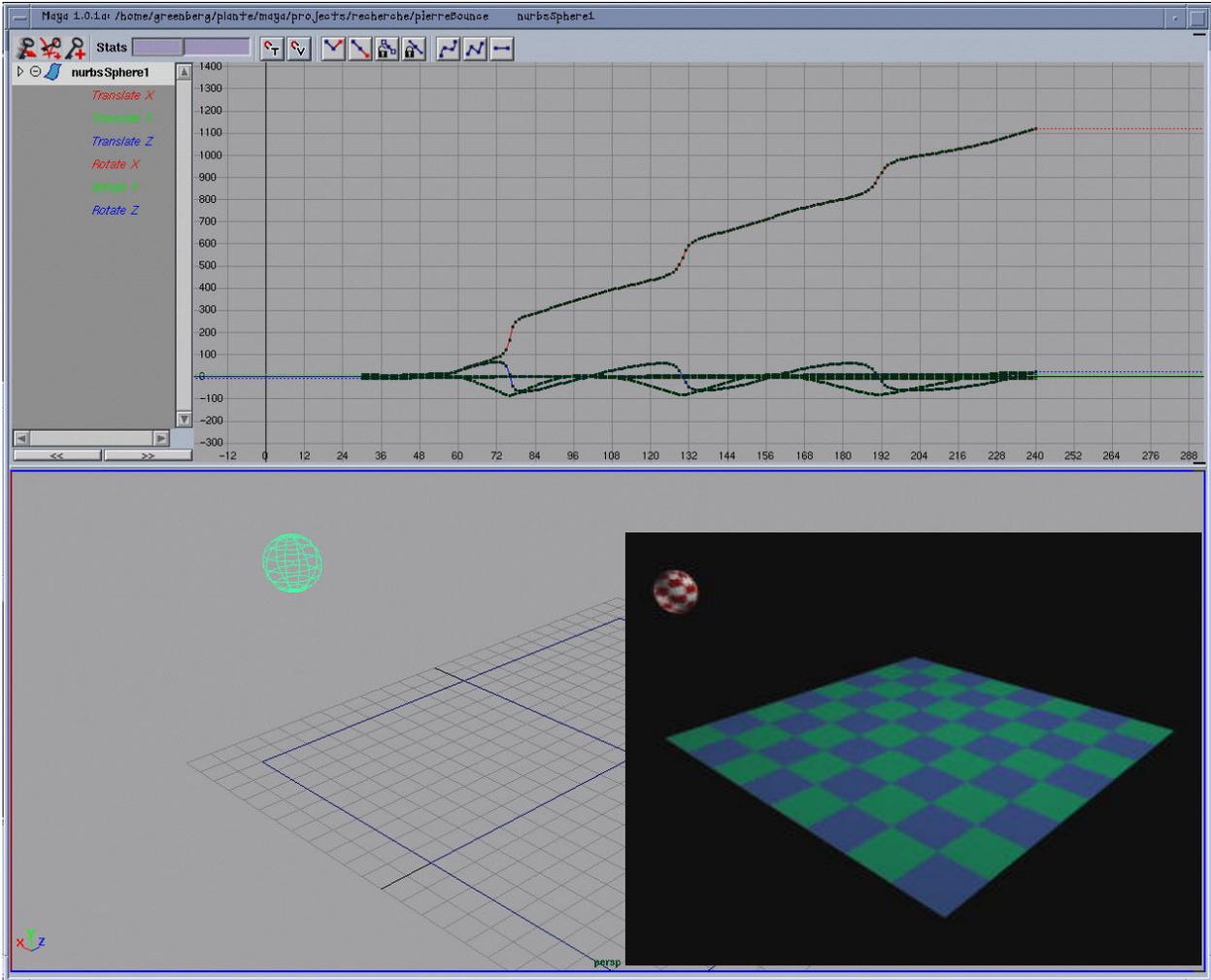
Animation par *keyframes*

- Le son est régulièrement utilisé pour déterminer la durée des actions d'un film de synthèse
- Seulement les positions/orientations sont habituellement fournies par l'animateur, les dérivées (vitesses, accélérations) sont laissées libres
- Une courbe de type *C1* est souvent suffisante
- Spline de Catmull-Rom est populaire avec contrôles additionnels de tension, continuité et biais
- Vitesse et accélération peuvent être aussi reparamétrisées entre les *keyframes*

Animation par *keyframes*

- + fondamental à la plupart des systèmes d'animation
- seulement approprié pour des mouvements simples de corps rigides, très difficile lorsque les modèles se complexifient
- c'est tout aussi un art que le *in-between* traditionnel et demande presque autant de travail
- attention aux mouvements indésirables
- meilleures animations jusqu'à présent ont été réalisées avec un recours important aux *keyframes*

Animation par *keyframes*



Maya

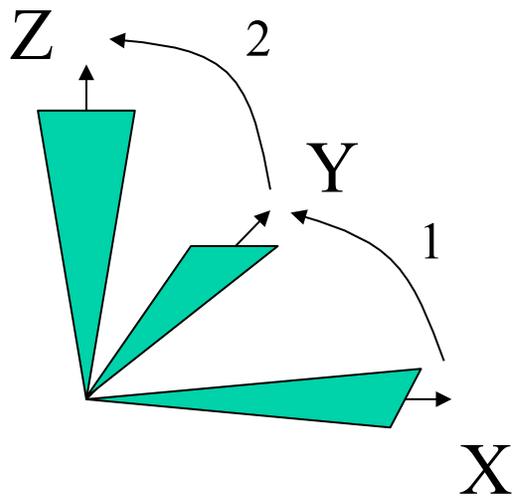
Interpolations

- Linéaire (*LERP*)

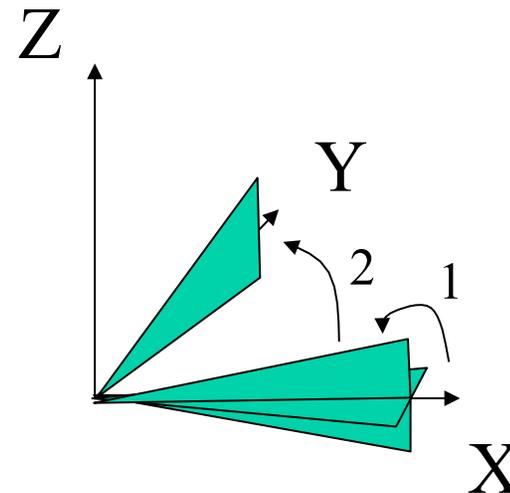
$$v_0 \text{ à } v_1 \rightarrow v_t = (1-t)v_0 + tv_1$$

- Angulaire - Euler
 - On ne doit pas interpoler les entrées entre deux matrices de rotation (ne correspond pas normalement à une rotation)
 - Les rotations ne sont pas commutatives: ordre des rotations donne des résultats différents
 - Interpoler linéairement autour des trois axes ne donne pas vraiment le résultat désiré (non-linéaire)
 - *Gimbal lock*

Problèmes d'interpolation angulaire avec Euler

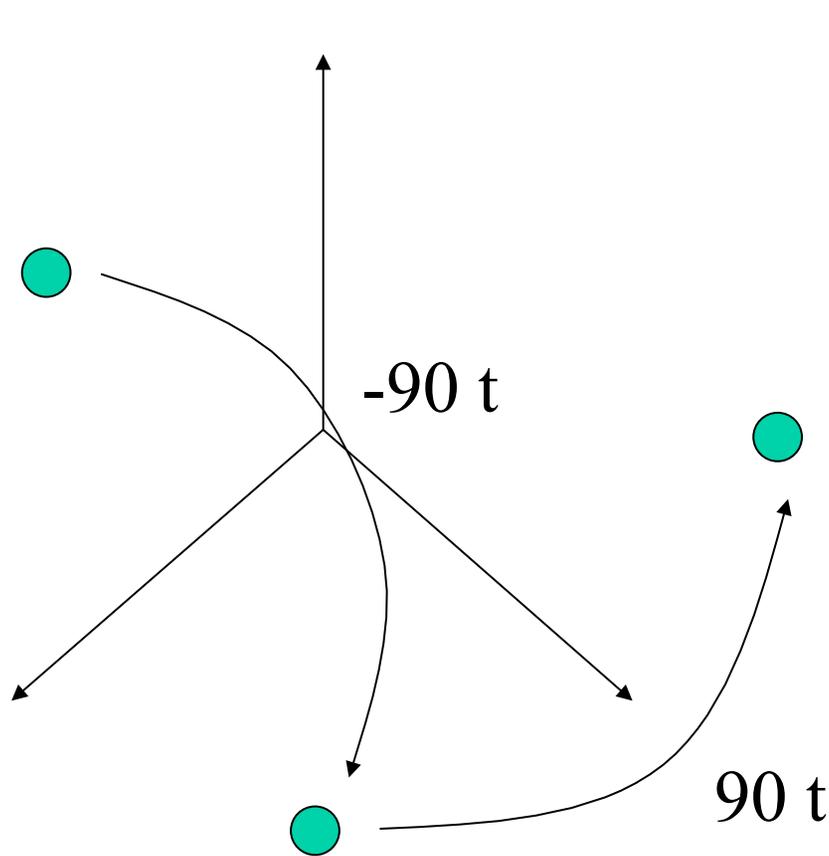


1. Rotation 90 Z
2. Rotation 90 X



1. Rotation 90 X
2. Rotation 90 Z

Problèmes d'interpolation angulaire avec Euler



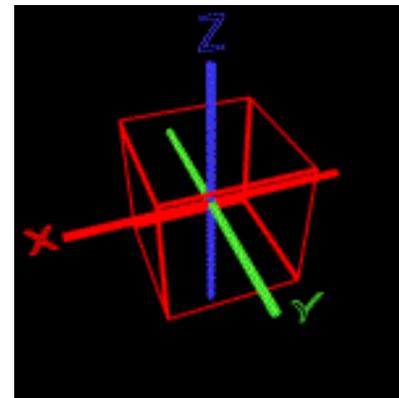
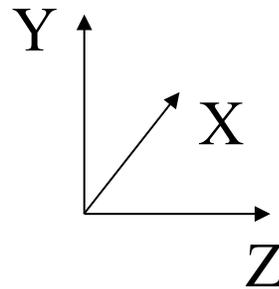
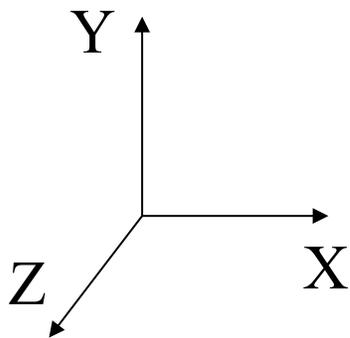
<u>t</u>	<u>t-Δt</u>
0.1	0.313
0.2	0.313
0.3	0.313
0.4	0.313
0.5	0.311
0.6	0.309
0.7	0.304
0.8	0.298
0.9	0.290
1.0	0.278

Problèmes d'interpolation angulaire avec Euler

- Difficile de développer une rotation à partir de trois rotations séparées
- Chemin interpolé est difficile à prédire
- Mouvements de rotation sont non-linéaires
- *Gimbal lock*: perte d'un degré de liberté

Gimbal Lock

- Une rotation de 90 degrés autour de Y aligne les axes X et Z
- Tourner autour de X est équivalent à tourner autour de Z: perte d'un degré de liberté
- Critique lorsque l'on doit interpoler autour des trois axes



Quaternions

- Un quaternion, une matrice de rotation et trois angles sont trois façons de spécifier une rotation
- Un quaternion ne spécifie pas une orientation dans l'espace, mais un changement d'orientation
- Un quaternion est représenté par un vecteur (b, c, d) en 3D et un scalaire (a)

$$a + bi + cj + dk$$

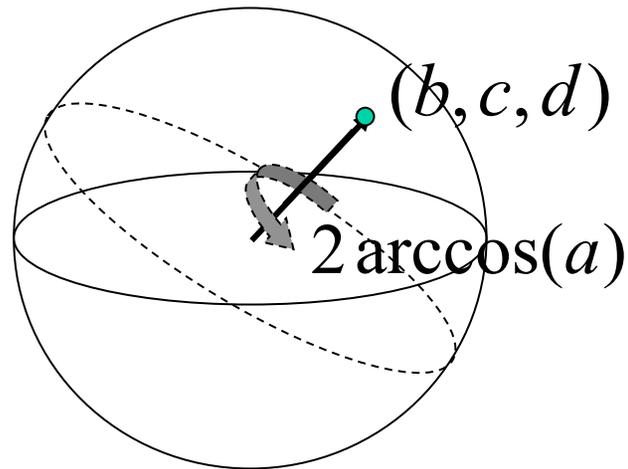
- Propriétés: $i^2 = j^2 = k^2 = ijk = -1$: parties imaginaires
 $ij = k = -ji$
 $a^2 + b^2 + c^2 + d^2 = 1$: point sur sphère 4D

Quaternions

- Une rotation de θ autour du vecteur (b, c, d) s'exprime en quaternion unité comme:

$$\cos \frac{\theta}{2} + b \sin \frac{\theta}{2} i + c \sin \frac{\theta}{2} j + d \sin \frac{\theta}{2} k$$

(a, b, c, d)



Opérations sur les quaternions

- L'addition de deux quaternions $(s_1, \vec{v}_1), (s_2, \vec{v}_2)$

$$(s_1 + s_2, \vec{v}_1 + \vec{v}_2)$$

- La multiplication de deux quaternions $(s_1, \vec{v}_1), (s_2, \vec{v}_2)$ correspond à faire une première rotation suivie de la deuxième

$$(s_1 s_2 - \vec{v}_1 \cdot \vec{v}_2, s_1 \vec{v}_2 + s_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2)$$

- L'inverse d'un quaternion correspond à

$$q^{-1} = \frac{[s, -\vec{v}]}{|q|^2} = \frac{[s, -\vec{v}]}{s^2 + v_x^2 + v_y^2 + v_z^2}$$

Appliquer une rotation via quaternions

- Pour tourner un vecteur v (ou un point) par un quaternion q , on considère le vecteur comme le quaternion $(0, v)$ et on applique

$$\text{Rot}(v, q) = qvq^{-1}$$

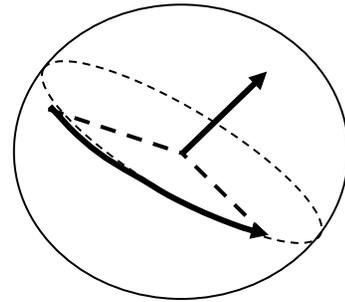
De quaternion à une matrice de rotation

- Il existe un petit algorithme pour passer d'une matrice de rotation à un quaternion
- Passer d'un quaternion à une matrice de rotation s'exprime comme :

$$q = [s, (x, y, z)] = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2sz & 2xz - 2sy & 0 \\ 2xy - 2sz & 1 - 2x^2 - 2z^2 & 2yz + 2sx & 0 \\ 2xz - 2sy & 2yz - 2sx & 1 - 2x^2 - 2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Quaternions

- Interpolation entre deux quaternions
 - grand arc entre deux points sur la sphère en 4D

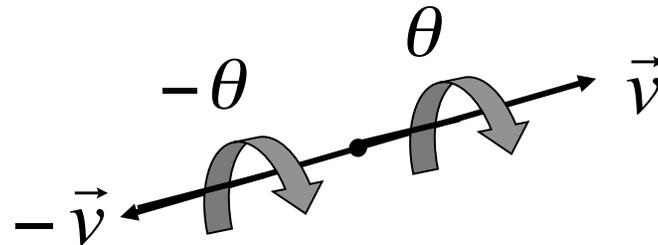


- Interpolation linéaire entre deux quaternions (*SLERP*) séparés par un angle θ

$$q(t) = \frac{\sin((1-t)\theta)}{\sin\theta} q_1 + \frac{\sin t\theta}{\sin\theta} q_2 \quad 0 \leq t \leq 1$$

Problèmes avec les quaternions

- Une même orientation peut être représentée par deux quaternions

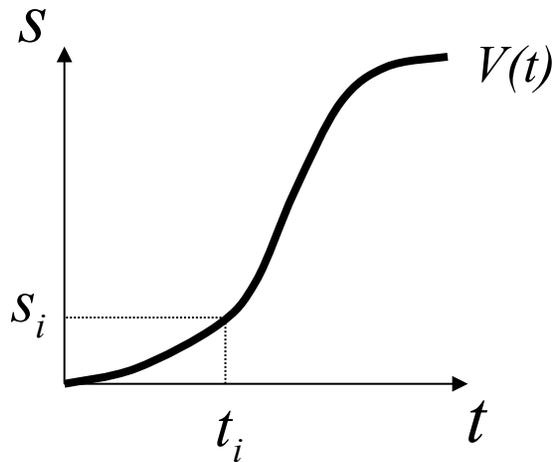


- Les rotations de 0 et 360 degrés sont représentées identiquement mais l'interpolation entre ces deux rotations est différente
- Aucun concept pour conserver la notion du vecteur **UP** lors de l'interpolation de rotations pour une caméra

Animation par splines

- Spécifie le mouvement (chemin) par une spline en 3D
 - *keyframe* est un sous-ensemble (linéaire) de ce mouvement
- Utilise deux courbes
 - $Q(u)$ définit le chemin en 3D
 - $V(u)$ définit la vitesse paramétrique sur ce chemin
- Une distance en espace paramétrique u correspond rarement à une distance équivalente en espace euclidien sauf si on utilise une paramétrisation par longueur d'arc (*arc length*)

Reparamétrisation



$$t_i \Rightarrow V(t_i) \Rightarrow s_i$$

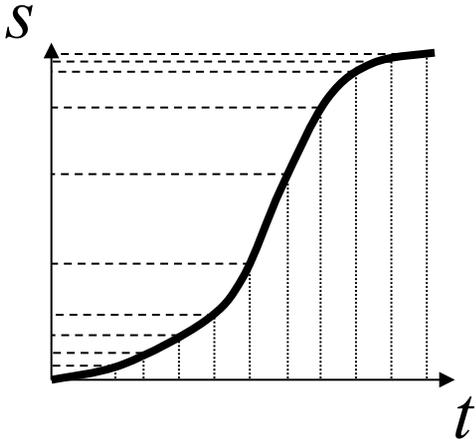
évalue $Q(s_i)$: reparamétrisation

$V(t)$ doit être une fonction croissante monotone

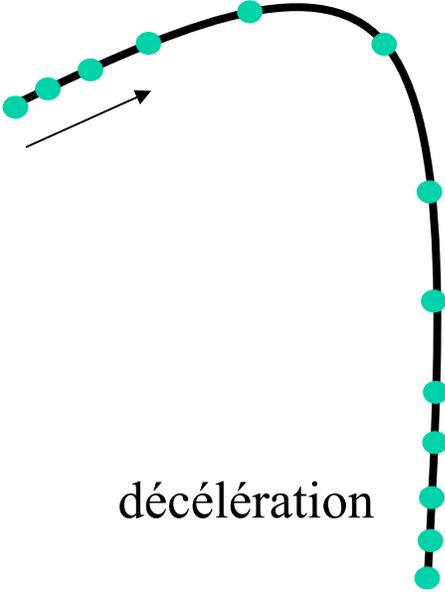
$$\text{i.e. } t_1 < t_2 \Rightarrow s_1 < s_2$$

Une pente négative de $V(t)$ correspond à un mouvement arrière

Reparamétrisation - *Ease-in ease-out*



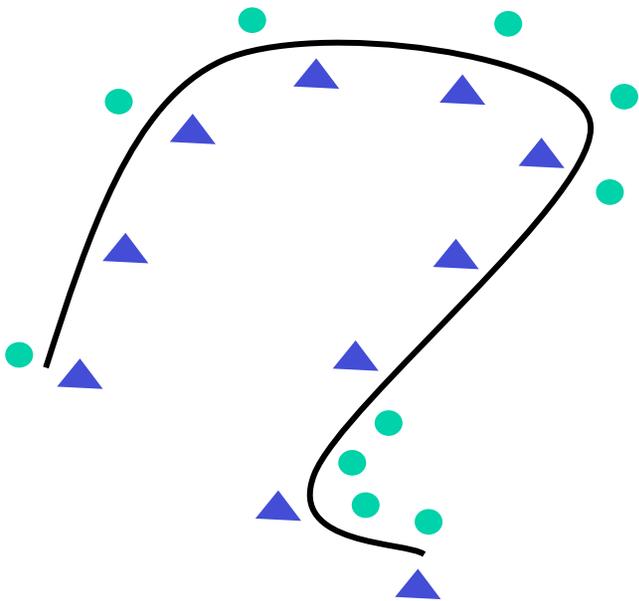
accélération



décélération

Paramétrisation longueur d'arc (*arc length*)

- $Q(t)$ intervalles égaux en paramétrique
- ▲ $Q(V(t))$ intervalles égaux en 3D

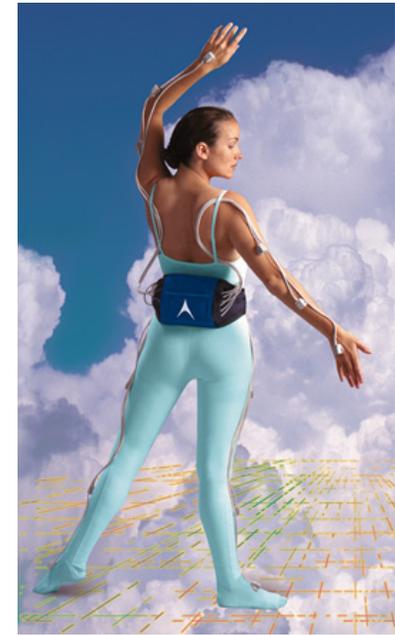


$$ds = \int_{t_0}^t \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$$

intégration numérique

Capture de mouvements

- Senseurs utilisés sur des acteurs réels
 - *magnetic flock of birds*
 - *optical video et motion tracking*
- Résultats simples et naturels
- Souvent bruit dans les données
 - doit corriger les données, lisser
- Il faut inférer les déplacements des joints d'après les senseurs
- Attention particulière au glissement des pieds (*foot skating*) et généralement des *end-effectors*



Ascension

Capture de mouvements

- Difficultés pour coordonner entre deux acteurs
 - *tracking* simultané
 - *tracking* non-simultané
 - danse, combat
- Différences de tailles et de morphologie (structure)
- Edition, réutilisation, changement de parcours

Cinématique

- Etude du mouvement en fonction du temps, indépendamment des forces qui produisent ce mouvement
- Propriétés du mouvement reliées au temps et à la géométrie:
 - position
 - vitesse : taux de changement de position en fonction du temps
 - accélération : taux de changement de vitesse en fonction du temps

Cinématique (linéaire et angulaire) en 2D

position : x

vitesse : $\frac{dx}{dt} = \dot{x} = v$

accélération : $\frac{d^2x}{dt^2} = \ddot{x} = \frac{d\dot{x}}{dt} = \frac{dv}{dt} = \dot{v} = a$

orientation : Ω

vitesse angulaire : $\frac{d\Omega}{dt} = \dot{\Omega} = \omega$

accélération angulaire : $\frac{d^2\Omega}{dt^2} = \ddot{\Omega} = \frac{d\omega}{dt} = \dot{\omega} = \alpha$

Figure articulée (squelette)

- Figure articulée formée de pièces (ou segments) reliées entre elles par des articulations (*joints*), le tout relié ensemble sous forme d'arbre
- Chaque pièce est modélisée à l'origine et son axe de rotation (ou deux) est spécifié dans ce système de coordonnées
- Chaque pièce possède une matrice de transformation qui la joint à sa pièce parent (repère local)
- Animer un modèle correspond à parcourir l'arbre à partir de la racine et accumuler (concaténer) les transformations pour atteindre ce noeud (sous forme de *push-pop* sur une pile de matrices)
- On peut aussi animer la longueur des segments (étirements) mais cela peut produire des mouvements étranges

Cinématique directe (*FK: forward kinematics*)

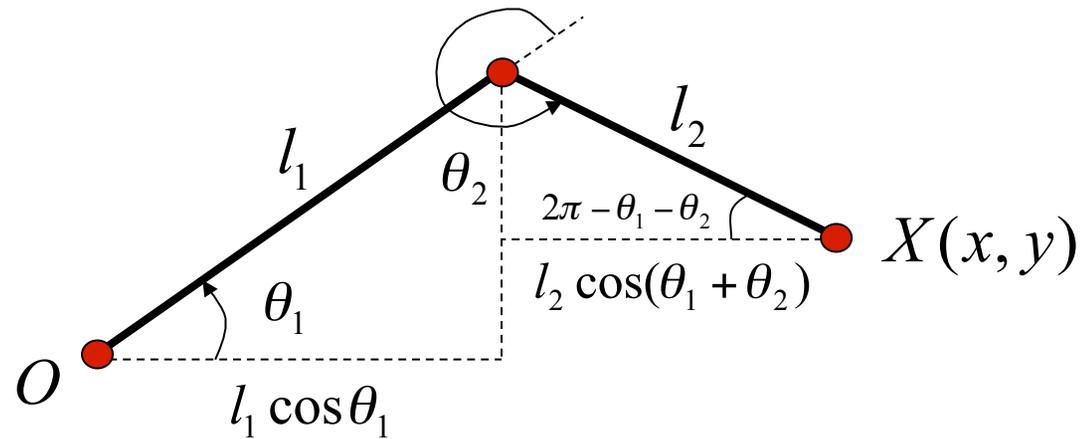
- Spécifie le mouvement de toutes les pièces

$$f(E) \rightarrow X$$

- Effecteur terminal (*end effector*)
 - dernier maillon d'une chaîne (racine vers branche finale)
- Le mouvement du *end effector* est déterminé en parcourant l'arbre de transformations

$$\text{pied} \leftarrow [\text{hanche}] \cdot [\text{genou}] \cdot [\text{cheville}]$$

Cinématique directe



$$X = (l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2), l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2))$$

Cinématique directe

- + interactif: liberté et simplicité
 - modifie les courbes du mouvement et affiche
- + connaissances de l'animation traditionnelle s'incorporent facilement à la cinématique
- + peut créer des bibliothèques paramétrables de mouvements communs
- devient de plus en plus contre-intuitif lorsque le nombre de *joints* à contrôler augmente (dans un objet articulé)
 - robotique
 - bras canadien

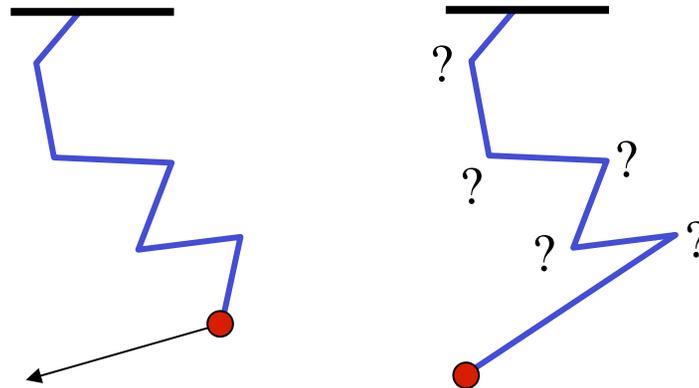
Cinématique inverse (*IK: inverse kinematics*)

- Spécifie seulement la position des *end effectors*

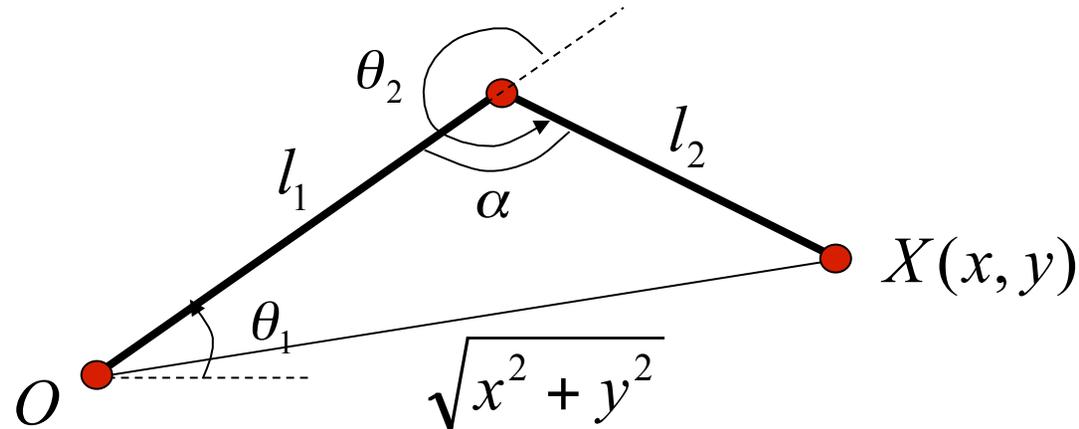
$$f^{-1}(X) \rightarrow E$$

- Le système doit calculer les positions et les orientations de toutes les autres pièces

pied \rightarrow [hanche] · [genou] · [cheville]



Cinématique inverse



règle des cosinus :

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos \alpha$$

$$\theta_2 = \pi + \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \right)$$

$$\theta_1 = \sin^{-1} \left(\frac{l_2 \sin \alpha}{\sqrt{x^2 + y^2}} \right)$$

Cinématique inverse

$$X = F \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \quad \partial X = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \cdots & \frac{\partial f_2}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \theta_1} & \frac{\partial f_n}{\partial \theta_2} & \cdots & \frac{\partial f_n}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} \partial \theta_1 \\ \partial \theta_2 \\ \vdots \\ \partial \theta_n \end{bmatrix}$$

inconnues
↓

(1) $X_{t+1} - X_t$ (2) relations géométriques précalculées

(3) $\theta_{t+1} = \theta_t + \partial \theta$ à résoudre, mais système sous-contraint...
itérations de (1), (2) et (3)

Cinématique inverse

- + commandes de haut niveau orientées *but*
- + automatique
 - marche sur terrain à vitesse v pour atteindre position p
- aucune ou plusieurs solutions possibles
 - contraintes pour réduire le nombre de solutions
 - minimise l'énergie
 - équations non-linéaires à résoudre numériquement
- + combine FK avec IK sur des parties séparées
- plus coûteux lorsque la complexité augmente
- difficile de spécifier le *naturel* d'un mouvement

Skinning

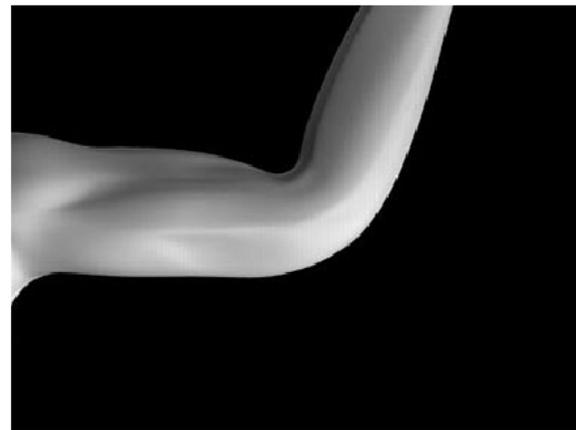
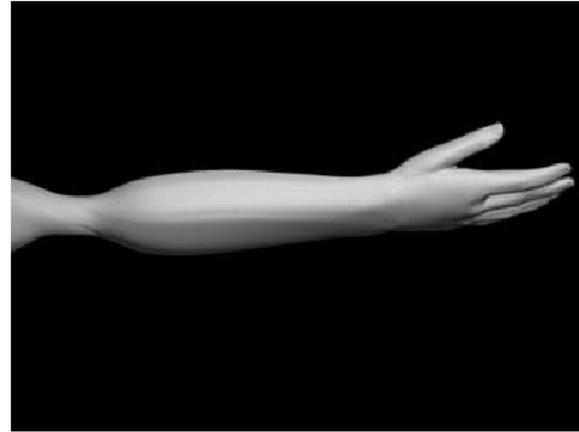
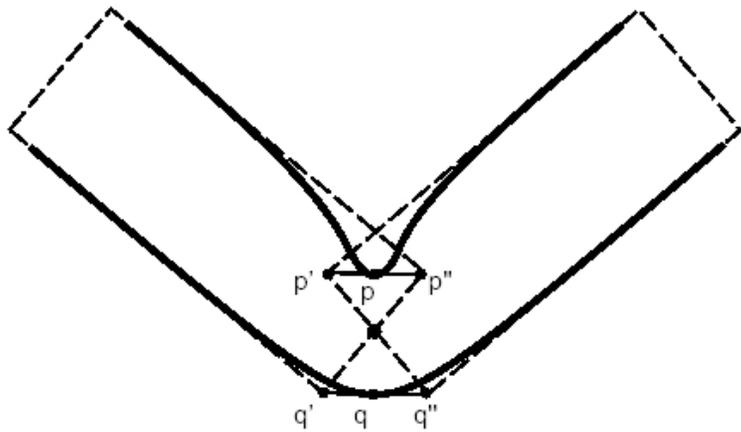
- Chaque sommet d'un objet rigide est attaché à son segment du squelette, et donc transformé comme son segment
- Pour une surface plus douce, on peut attacher un sommet à plus d'un segment (*smooth skin*) et transformer ce sommet par une pondération des transformations appliquées à ses segments $v' = Mv$

$$v' = \sum (w_i M_i v) \quad \text{où} \quad \sum w_i = 1$$

- Mais les M doivent être définies par rapport à la matrice au joint dans le repère du monde... et ça se complique...

$$v' = \sum \left(w_i M_i (M_i^{ref})^{-1} v \right) \quad \text{référentiel local de la configuration initiale}$$

Skinning



Dynamique

- En cinématique, on doit décrire explicitement tout mouvement en fonction du temps
- En dynamique, on décrit les propriétés physiques (forces et masses) et le système en déduit les mouvements résultants
 - résolution numérique d'équations différentielles ou d'équations aux dérivées partielles
 - au moins une des variables de dérivation est le temps
 - au moins deux grandeurs physiques
 - position-force, vitesse-force, vitesse-pression, position-énergie, orientation-moment de force, etc.

Dynamique directe

- Position en fonction des forces $\vec{F} \rightarrow \vec{X}$
- La seconde loi de Newton $\sum_i \vec{F}_i = m\vec{A}$
 - \vec{F}_i sont les forces qui s'appliquent sur cet objet
 - \vec{A} est l'accélération d'un objet de masse m

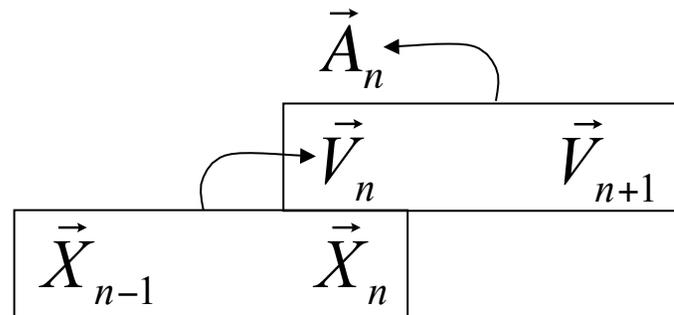
$$\vec{A} = \frac{d\vec{V}}{dt} = \frac{d^2\vec{X}}{dt^2}$$

- Pour calculer la position $\vec{X}(t)$ de l'objet

$$\vec{X} = \int \vec{V} dt = \iint \vec{A} dt^2 = \frac{1}{m} \iint \left(\sum_i \vec{F}_i \right) dt^2$$

Discrétisation dans le temps

- Au lieu d'intégrer dans le temps, on peut discrétiser les équations (différences) selon Δt
- Un exemple de discrétisation:



$$\vec{A}_n = \frac{\vec{V}_{n+1} - \vec{V}_n}{\Delta t} \quad \Leftrightarrow \quad \vec{V}_{n+1} = \vec{V}_n + \vec{A}_n \Delta t = \vec{V}_n + \frac{\sum \vec{F}_i}{m} \Delta t$$

$$\vec{V}_n = \frac{\vec{X}_n - \vec{X}_{n-1}}{\Delta t} \quad \Leftrightarrow \quad \vec{X}_n = \vec{X}_{n-1} + \vec{V}_n \Delta t$$

Dynamique inverse

- Force en fonction des positions

$$d = \|\vec{X}_2 - \vec{X}_1\|$$

Elastiques

$$F = -kd$$

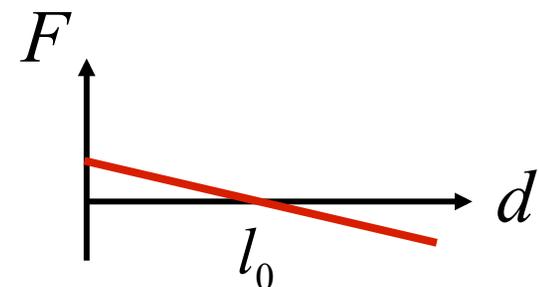
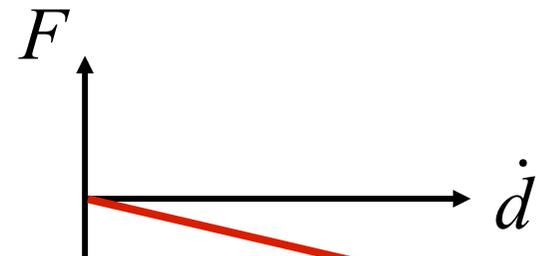
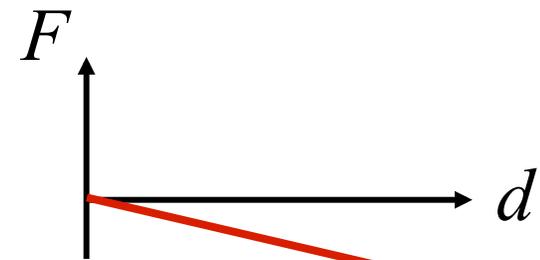
Frottements
visqueux

$$F = -z\dot{d}$$

Elastiques
ressorts

$$F = -k(d - l_0)$$

$$\vec{X} \rightarrow \vec{F}$$

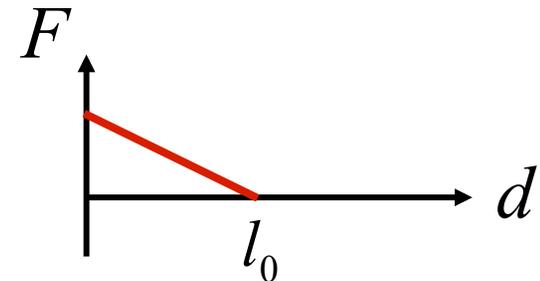


Dynamique inverse

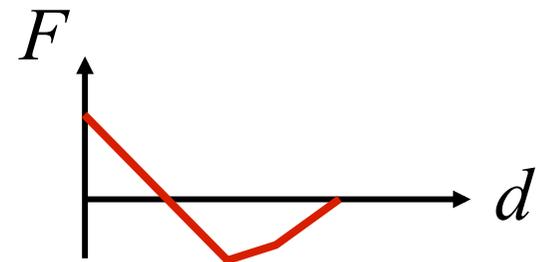
- Force en fonction des positions

$$\vec{X} \rightarrow \vec{F}$$

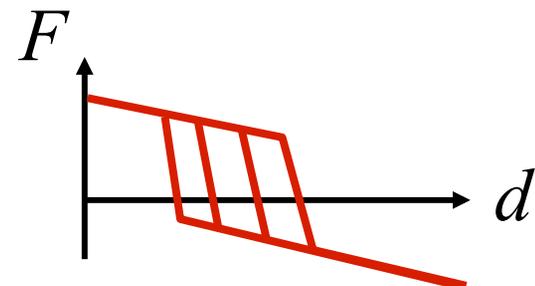
$$\text{Collision élastique} \begin{cases} F = -k(d - l_0) & \text{si } d \leq l_0 \\ F = 0 & \text{sinon} \end{cases}$$



Cohésion
Lennard - Jones



Elastiques
plastiques



Particule

- Particule est simple: masse, position, vitesse
 - Particule n'a pas de géométrie (seulement un point)
 - Particule réagit aux forces
- + Facile à simuler
- + Peut les joindre par des ressorts pour simuler des structures non-rigides
- + Des particules spéciales de contrôle peuvent servir comme point d'attache ou de pivot
- + Peut générer des comportements de fluides (SPH) en fonction de la densité (pression) de particules à un endroit

Simulation d'une particule

$$\mathbf{F} = m\mathbf{a}$$

$$\ddot{\mathbf{x}} = \dot{\mathbf{v}} = \mathbf{F}/m$$

$$\dot{\mathbf{x}} = \mathbf{v}$$

$$\begin{bmatrix} x_x \\ x_y \\ x_z \\ v_x \\ v_y \\ v_z \end{bmatrix}$$

Espace de
phase
d'une
particule

Systeme de particules

- Particule
 - {float m , * x , * v , * f }
- Systeme de particules
 - {Particule * p , int n ; float $time$ }
- Forces
 - Efface toutes les forces du temps precedent
 - Traite la liste des forces (gravite, vent, ressorts, etc.)
 - Une force applique la methode *ApplyForce* sur sa liste de particules, et retourne v et F/m pour chaque particule de sa liste

Particules: forces unaires

- Gravité

$$\mathbf{F} = m\mathbf{g}$$

- Frottement (*drag*)

$$\mathbf{F} = -k_d \mathbf{v}$$

- Résistance aux mouvements
- Arrête éventuellement la particule si aucune autre interaction
- Un peu de frottement augmente la stabilité numérique
- Trop de frottement donne l'impression de bouger dans de la mélasse

Particules: forces n-naires

- Deux particules aux positions a et b se déplaçant à vitesses v_a et v_b , reliées par un ressort
- Loi de Hooke

$$\mathbf{F}_a = - \left(k_s (|a - b| - l) + k_d \frac{(v_a - v_b) \cdot (a - b)}{|a - b|} \right) \frac{a - b}{|a - b|}$$

$$\mathbf{F}_b = -\mathbf{F}_a$$

l : longueur du ressort au repos

k_s : constante du ressort

k_d : *damping* proportionnel à la différence de vitesses

Particules: forces spatiales

- Attraction et répulsion en fonction des positions des particules
 - ex: fluides, planètes
 - $O(n^2)$: utilisation de structures spatiales

Particules: collisions

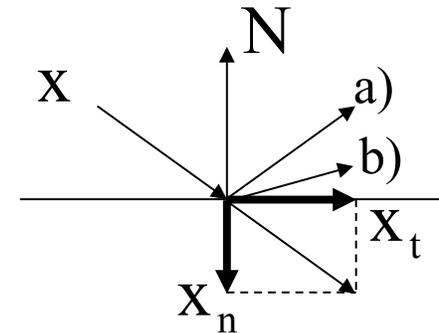
- Détection
 - collision avec un plan si $(\mathbf{x} - \mathbf{p}_p) \cdot \mathbf{N} < 0$
- Traitement
 - Correct
 - Calcule le temps du contact
 - Recule le temps de simulation à ce moment
 - Traite la collision (changements de vitesse)
 - Continue la simulation
 - Plus facile
 - Déplace la particule qui était sous la surface
 - Aucune garantie que la nouvelle position est correcte

Particules: collisions

- Réponse à la collision

$\mathbf{x}_n = (\mathbf{N} \cdot \mathbf{x})\mathbf{N}$: composante normale

$\mathbf{x}_t = \mathbf{x} - \mathbf{x}_n$: composante tangentielle



- a) Elastique sans friction

- La composante normale de la vitesse est multipliée par -1.0

- b) Inelastique sans friction

- La composante normale de la vitesse est multipliée par $-r$ où r est entre 0 et 1

- Frottement

$$N \cdot F < 0 \quad \rightarrow \quad F_{contact} = -(N \cdot F)N \quad -k_f (-F \cdot N) v_t$$

Objets articulés

- Intégrer les forces et les moments

$$\sum_i \vec{F}_i = m\vec{A}$$

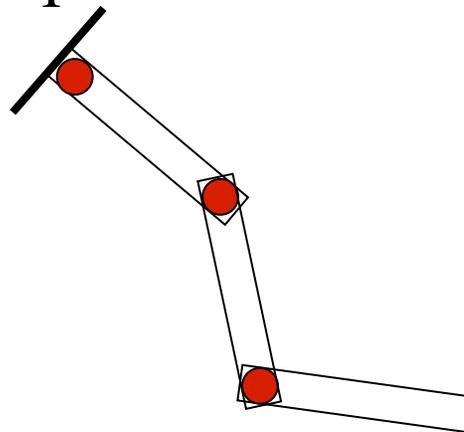
$$\sum_i \vec{C}_i = I\dot{\omega} + \omega \times I\omega$$

C: couples (*torques*)

ω : vitesse angulaire

I: inertie

- Tenir compte des contraintes

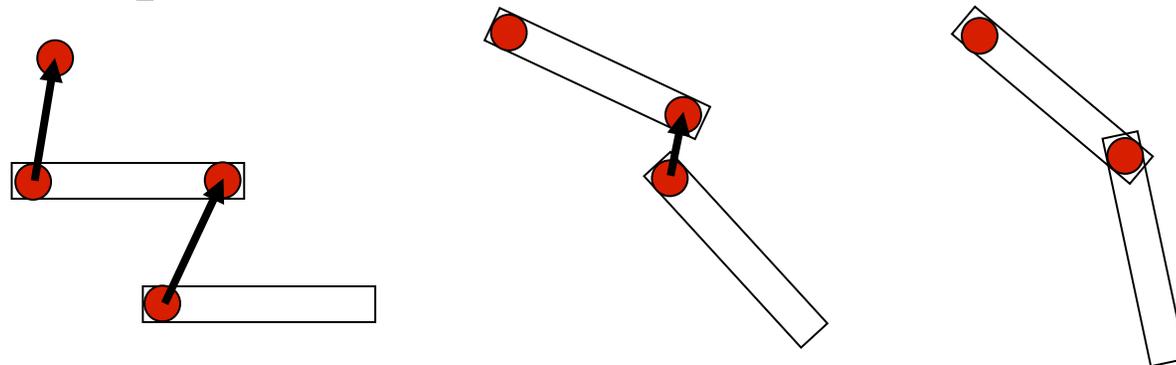


Simulation de chaînes articulées

- Méthode récursive:
 - calcul des équations de la dynamique pour chaque segment de la structure hiérarchique
 - calcul de la relation de dépendance entre accélération de chaque segment fils et de son père
 - calcul de l'accélération de la racine, en remontant les valeurs calculées précédemment
 - propagation pour calculer les nouvelles positions et vitesses de tous les segments

Simulation par contraintes

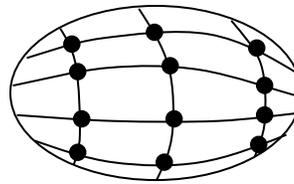
- L'utilisateur spécifie des contraintes à respecter durant l'animation, par exemple:
 - deux points coïncident
 - un point réside sur une courbe
- La dynamique inverse calcule les forces à appliquer sur les objets pour que les contraintes soient respectées



Objets déformables

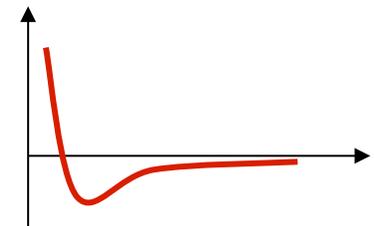
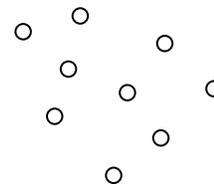
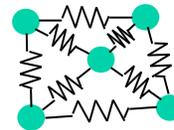
- Solide élastique, pâte visqueuse, liquide, fumée, etc.
 - partir d'une équation des milieux continus

- élasticité
- fluides



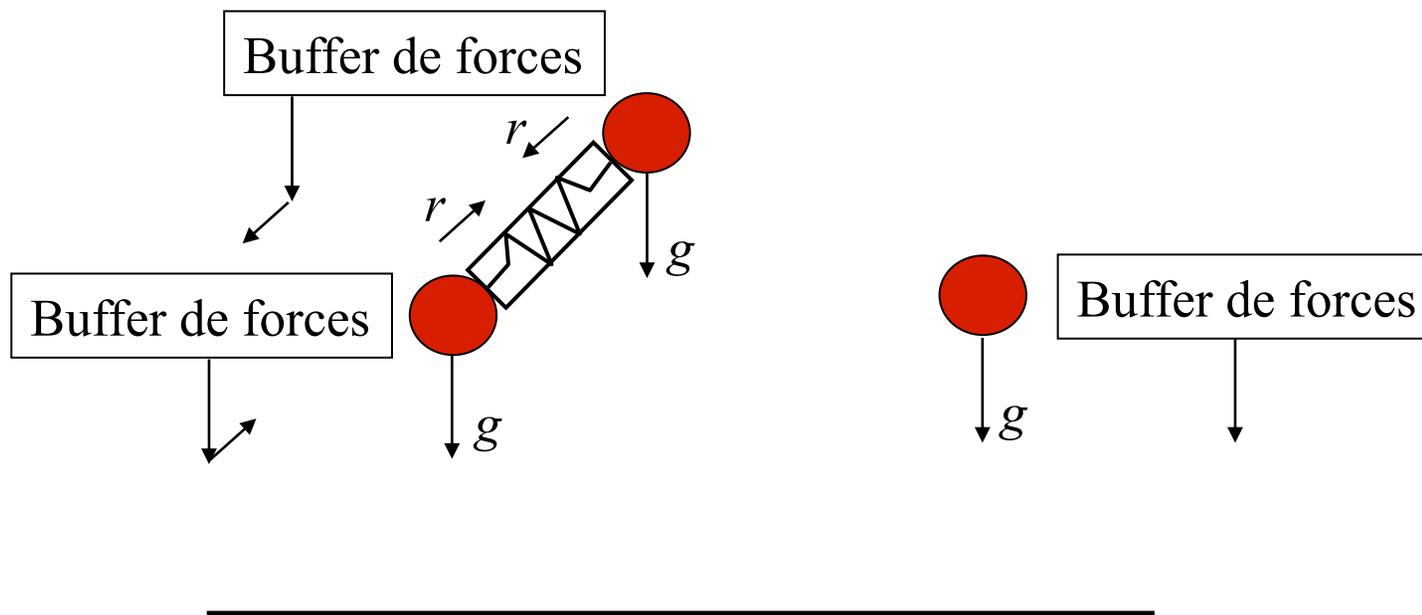
- combiner des composants mécaniques simples

- réseaux masses-ressorts
- systèmes de particules



Systeme de particules

- Pour un systeme de n particules, on applique les forces globales à toutes les particules et les forces d'interactions par paire de particules



Dynamique d'un système de particules

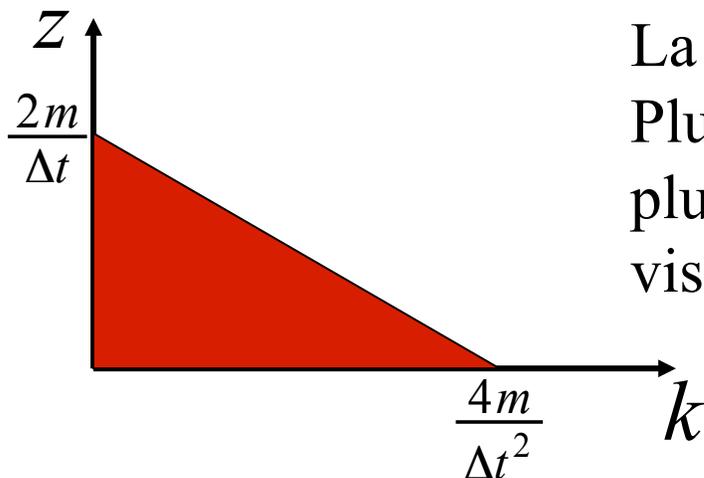
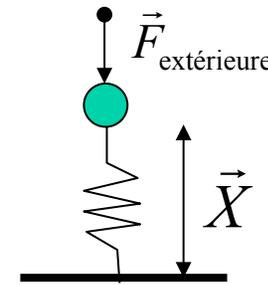
- Pour chaque pas de simulation dans le temps
 - initialise tous les buffers de force
 - pour tout couple (M_i, M_j)
 - calcule le vecteur unité $\vec{u}_{ij} = \frac{M_j - M_i}{\|M_j - M_i\|}$
 - calcule la force F exercée par M_i sur M_j
 - ajoute $F\vec{u}_{ij}$ dans le buffer de force de M_j
 - ajoute $-F\vec{u}_{ij}$ dans le buffer de force de M_i
 - calcule la nouvelle position de chaque particule en fonction des forces dans son buffer

Instabilité de la discrétisation dans le temps

$$\sum_i \vec{F}_i = m\vec{A} \Leftrightarrow \vec{F}_{\text{élastique}} + \vec{F}_{\text{visqueuse}} + \vec{F}_{\text{extérieure}} = m\vec{A}$$

$$m\vec{A}_n + z\vec{V}_n + k\vec{X}_n = \vec{F}_{\text{extérieure}}$$

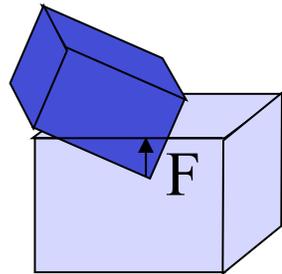
$$\alpha\vec{X}_{n+1} + \beta\vec{X}_n + \gamma\vec{X}_{n-1} + \vec{\delta} = 0$$



La zone rouge est dite zone stable.
Plus la discrétisation temporelle est fine,
plus on peut modéliser des objets rigides,
visqueux et légers.

Interactions entre objets

- Détection des interpénétrations



- Réponse aux collisions
 - choc rigide: modification des vitesses
 - choc mou ou contact: force de réaction
- Méthode de pénalité
 - force en fonction de la pénétration

Dynamique

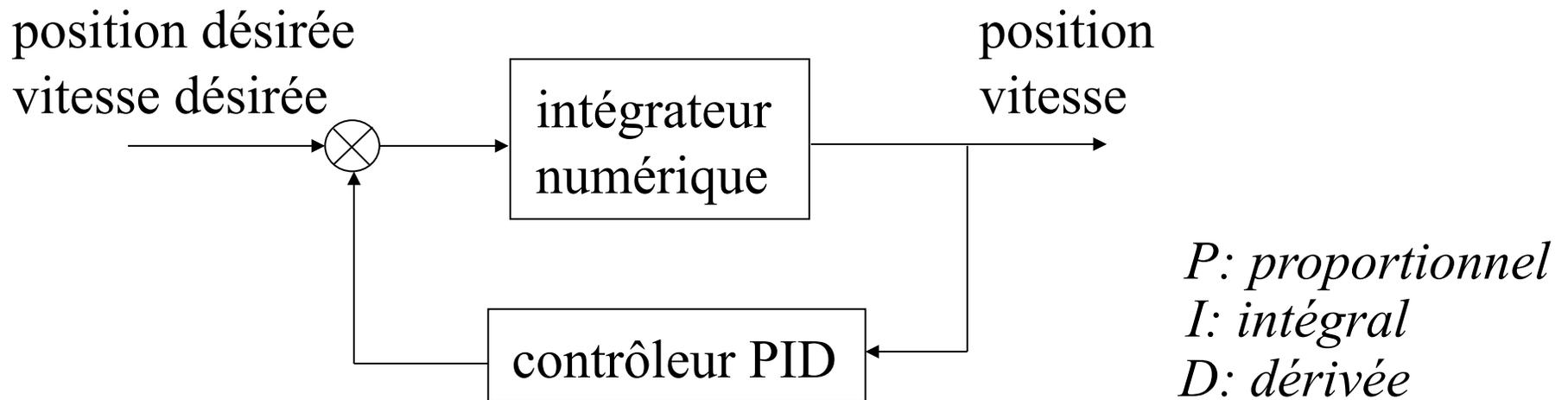
- + Un seul modèle dynamique peut produire une grande variété de comportements
- Difficile d'obtenir un comportement particulier dans toute cette variété de mouvements
- + Lorsque plusieurs éléments interagissent, la dynamique réussit là où de bons animateurs éprouvent des difficultés:
 - liquides, flammes, fumée, etc.

Dynamique vs. cinématique

- Cinématique
 - + grand contrôle (mais beaucoup de travail)
 - faible réalisme
- Dynamique
 - faible contrôle
 - + grand réalisme

Compromis cinématique-dynamique

- Optimisation de trajectoire:
 - *space-time constraints*: contraintes
 - + critère énergétique
- Contrôleurs:



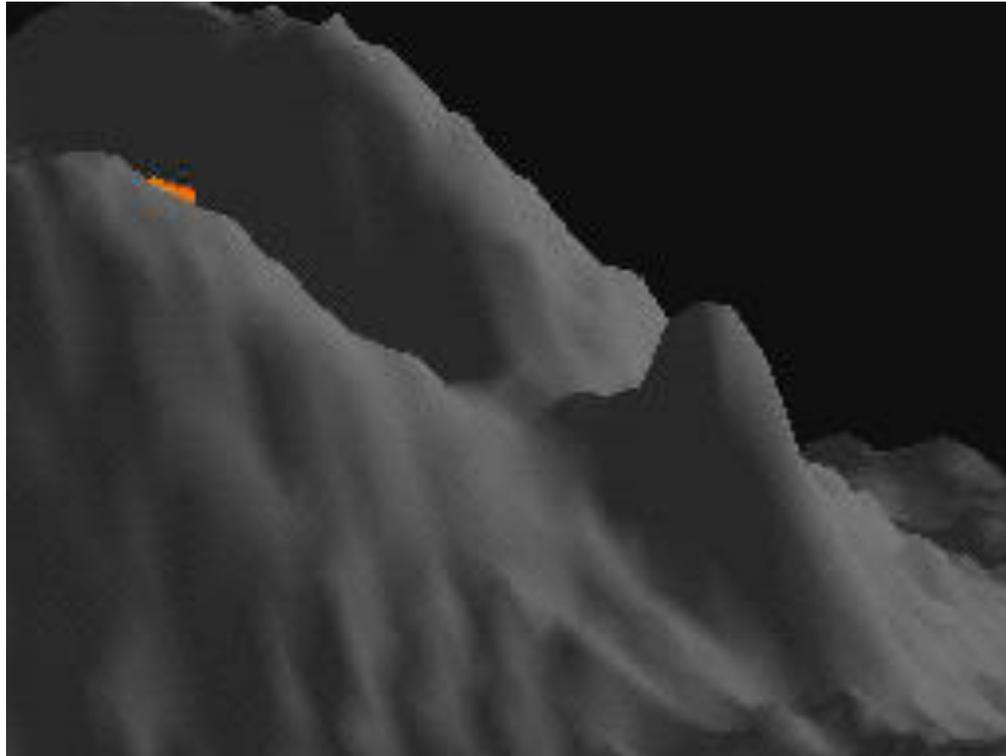
$$F = -k_p (X - X^*) - k_d (\dot{X} - \dot{X}^*) - k_i \int (X - X^*) dX$$

Films dynamiques



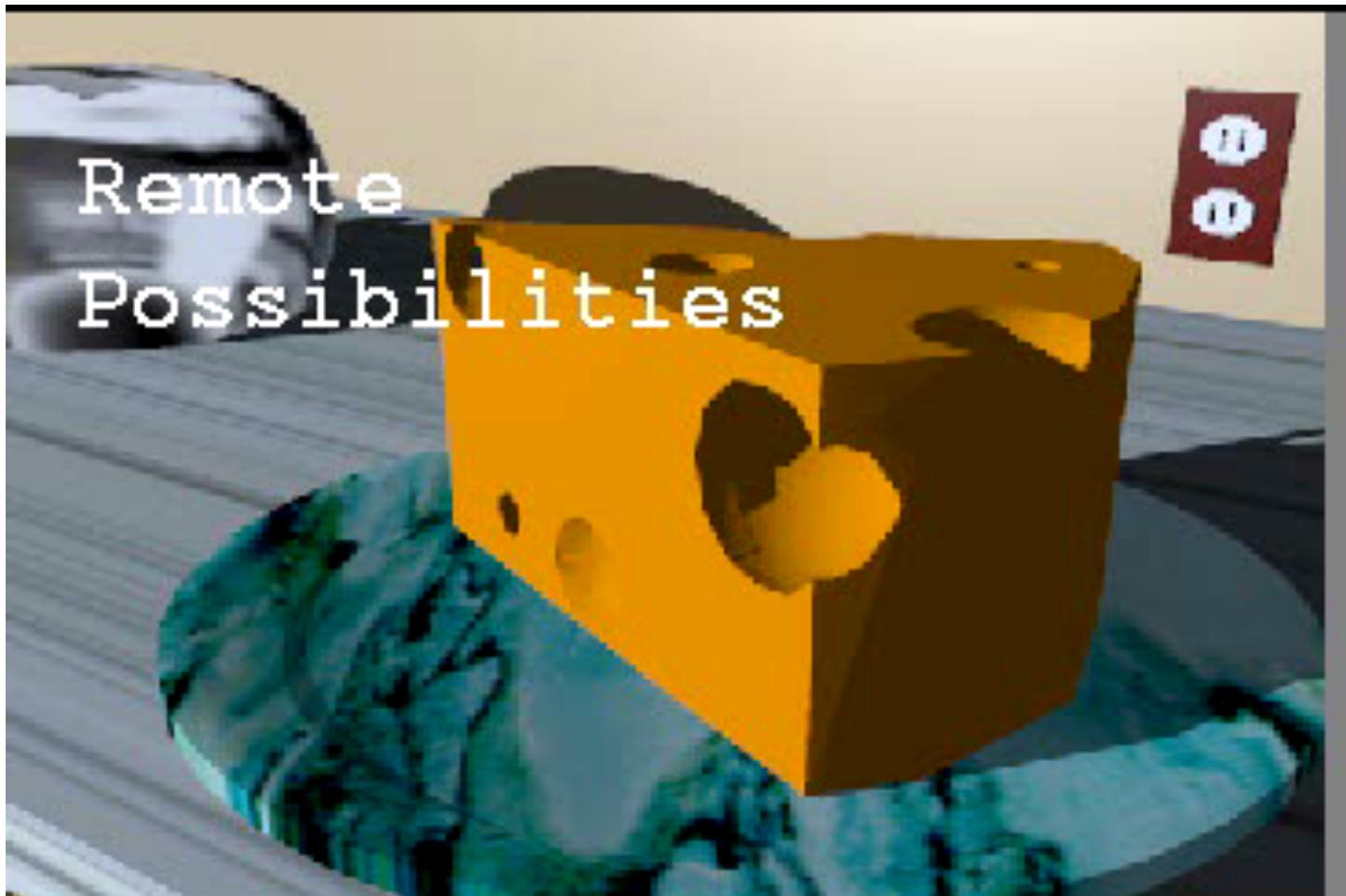
Anderson

Films dynamiques



iMAGIS

Films dynamiques



WSU

Films dynamiques



Films dynamiques



Animation comportementale (*behavior*)

- Comportement d'un agent est géré par des règles de base
- *flocking*: suit le groupe mais évite tes voisins et l'environnement

COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPOULOS

"BOIDS DEMOS"
CRAIG REYNOLDS
SILICON STUDIOS, MS 3L-980
2011 NORTH SHORELINE BLVD.
MOUNTAIN VIEW, CA 94039-7311

COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPOULOS

"ARTIFICIAL FISHES"
DEMETRI TERZOPOULOS
DEPT. OF COMPUTER SCIENCE
UNIVERSITY OF TORONTO
10 KING'S COLLEGE ROAD
TORONTO, ON
CANADA M5S 1A4

Animation comportementale (*behavior*)

- On peut aussi donner des buts plus complexes aux agents

COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPOULOS

"ALIVE DEMO"
PATTIE MAES
MEDIA LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
20 AMES STREET
CAMBRIDGE, MA 02139

COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPOULOS

"EVOLVED VIRTUAL CREATURES"
KARL SIMS
THINKING MACHINES CORP.
245 FIRST STREET
CAMBRIDGE, MA 02142

Quand on allie toutes ces techniques



Alias!wavefront