Dynamic Skinning

Kinematics-Driven Cartoon Effects for Articulated Characters

Damien Rohmer[1], Karim Salem[2], Niranjan Kalyanasundaram[2], Victor Zordan[2]

[1]LIX, École polytechnique, IP Paris [2]Clemson University

IEEE TVCG 2026



Motivation : animation expressive

Effet d'exagération ⇒ **perception du mouvement**

visible même sur une image fixe : vitesse, poids, souplesse





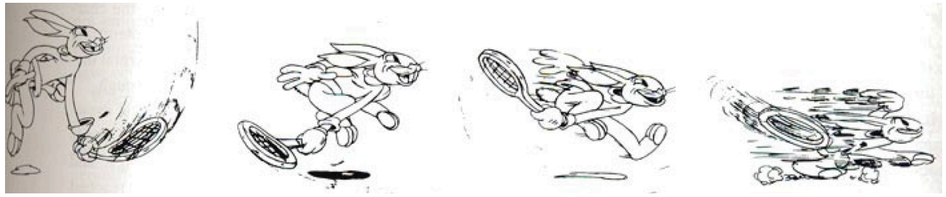
Codifiée dans les **12 principes** de l'animation

Drag : la forme traîne pendant le mouvement

→ Expressivité

Follow-through : la déformation se poursuit après l'arrêt

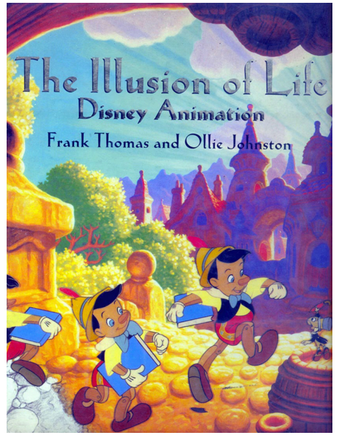
→ Animation secondaire



Idée : s'appuyer sur le **skinning** du rig

pour générer **animation secondaire** et **expressivité**

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \underbrace{\mathcal{D}(p_i, t)}_{\text{fonction dynamique}}$$





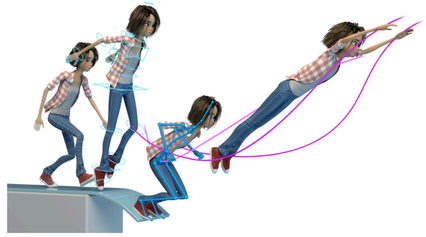
Raiponce (Disney, 2010) ; The Tortoise and the Hare (Disney, 1935) ; F. Thomas, O. Johnston, The Illusion of Life, 1981 ; animation : Chiara Porri

1 min

État de l'art : 4 grandes approches

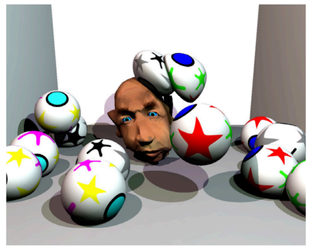
Animation

Conception de trajectoires



Guay 2015, Choi 2016, Ciccone 2017

Simulation physique

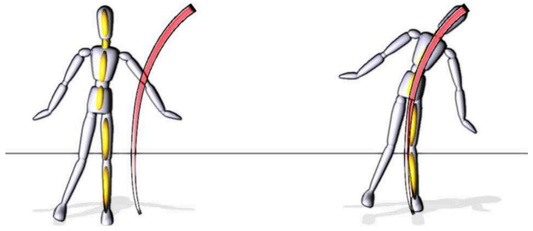


PD, modèles réduits, Wu 2023, CODY 2023

Dynamic Skinning

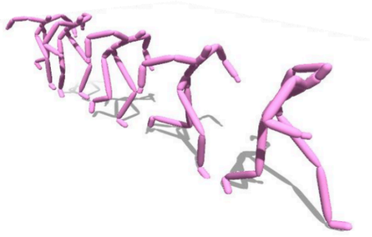
Manuel — Automatique

Esquisses



Guay 2013, Hahn 2015, Dvorožňák 2017

Transfert de pose / apprentissage



Villegas 2018, Zhang 2023

Pose clé

Dynamic Skinning :

Temps réel
même ordre de grandeur que le LBS

Réutilise le rig
reprend la base existante de l'artiste
pas de nouveaux paramètres de surface / volume

Purement procédural
évaluation directe, prédictible

2 min

Dynamic Skinning

Une formule, trois ingrédients

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Une position de skinning + une somme de contributions cartoon

Skinning (LBS) + déformeur (drag) + filtre temporel



Base : p_i^{LBS}
Le personnage suit le squelette



Espace : $\beta_{ij}\, \psi$
La peau traîne pendant le mouvement



Temps : $\mathcal{F}(\Omega_j, t)$
Elle oscille aussi après l'arrêt

3 min

Rappel : Linear Blend Skinning (LBS)

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Un sommet i, plusieurs articulations j

Formulation classique

$$p_i^{\mathrm{LBS}} = \sum_j \alpha_{ij}\, T_j\, (T_j^0)^{-1}\, p_i^0$$

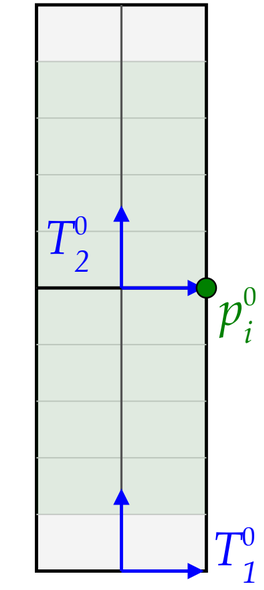
α_{ij} : Skinning weights

$p_{i/j} = T_j\, (T_j^0)^{-1}\, p_i^0$: sommet transporté rigidement par j

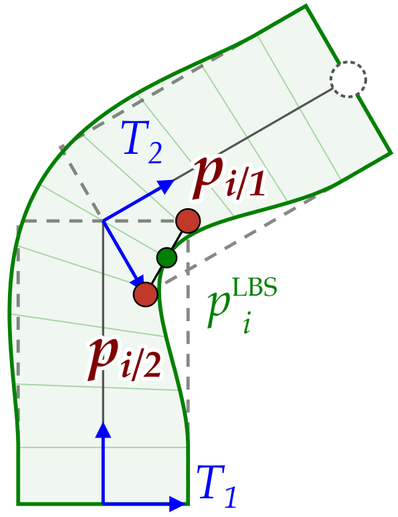
LBS = Blending de positions

$$p_i^{\mathrm{LBS}} = \sum_j \alpha_{ij}\, p_{i/j}$$

Idée : pour la dynamique $\rightarrow$ expression en vitesse



Repos



5 min

LBS en vitesse (1/2)

$$p_i^{\text{DS}}(t) = p_i^{\text{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\text{LBS}}\big)$$

1. **Dériver le LBS** : un blending de vitesses

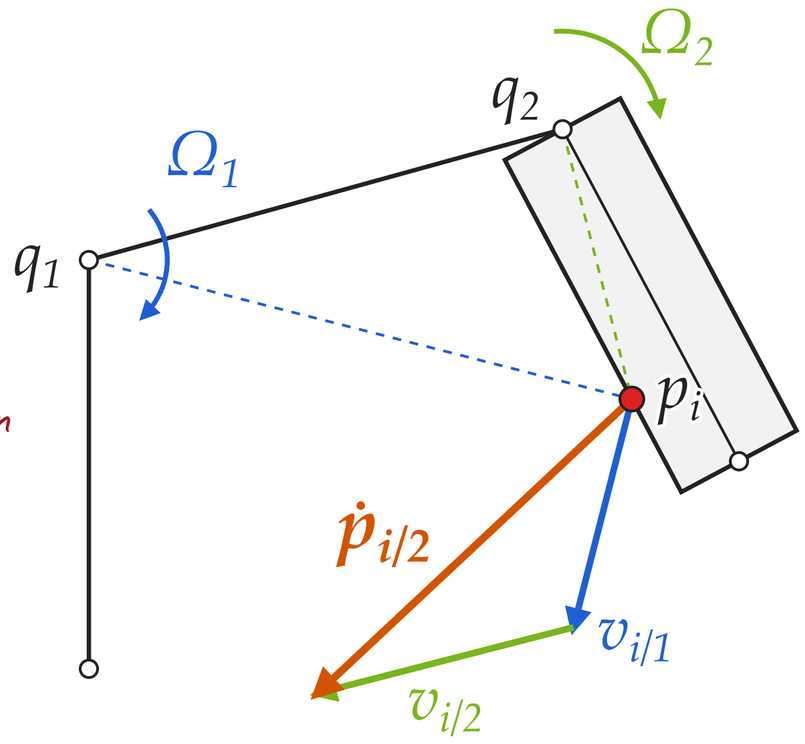
$$\dot{p}_i^{\text{LBS}} = \sum_k \alpha_{ik}\, \dot{p}_{i/k}$$

2. **Chaîne articulée** : les vitesses s'additionnent

$$\dot{p}_{i/k} = \sum_{j \in \mathcal{A}(k)} v_{i/j} \qquad \text{ex. } \dot{p}_{i/2} = v_{i/1} + v_{i/2}$$

$$\dot{p}_i^{\text{LBS}} = \underbrace{\sum_k \alpha_{ik} \sum_{j \in \mathcal{A}(k)}}_{\text{double somme}} v_{i/j}$$

vitesse du sommet i, relative à la rotation de l'os j uniquement



3. **Une articulation** : rotation autour de q_j

$$v_{i/j} = \Omega_j \times (p_i - q_j)$$

q_j : position du repère j

Ω_j : vitesse angulaire ; translations omises

6 min

LBS en vitesse (2/2)

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Regrouper les contributions d'une **même articulation**

$$\dot{p}_i^{\mathrm{LBS}} = \sum_k \alpha_{ik} \sum_{j \in \mathcal{A}(k)} v_{i/j} = \sum_j \beta_{ij}\, v_{i/j}$$

on inverse les sommes : $\beta_{ij} = \sum_{k \in \mathcal{D}(j)} \alpha_{ik}$
(j et ses descendants)

β_{ij} : Hierarchical weights

Calculé **une fois**, depuis le rig existant

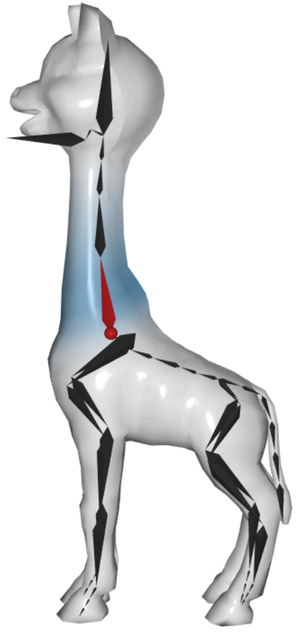
α pondère les **positions** ; β pondère les **mouvements locaux**

Skinning en vitesse

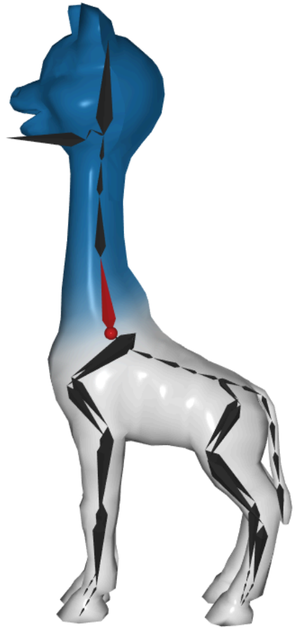
$$\dot{p}_i^{\mathrm{LBS}} = \sum_j \beta_{ij}\, v_{i/j} = \sum_j \beta_{ij}\, \Omega_j \times (p_i - q_j)$$

Intérêt : fonction de la vitesse angulaire Ω_j

$\rightarrow$ espace paramétrique **compréhensible**



Poids de skinning α_{ij}



Poids hiérarchiques β_{ij}

7 min

Déformeur : interpréter une vitesse en drag

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

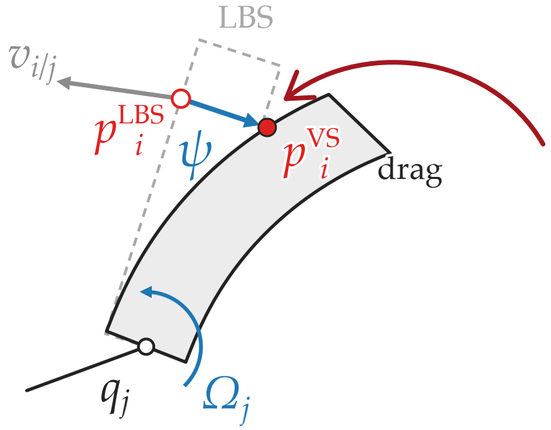
Vitesse $\dot{p}_i^{\mathrm{LBS}} = \sum_{\text{joints } j} \beta_{ij} \quad \Omega_j \times (p_i - q_j)$

Déformation $p_i^{\mathrm{VS}} = p_i^{\mathrm{LBS}} + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\Omega_j,\, q_j,\, p_i^{\mathrm{LBS}}\big)$

VELOCITY SKINNING

ψ : fonction procédurale, effet de **drag**

la peau traîne derrière le mouvement de rotation



$\psi(\Omega_j, q_j, p_i) = \big(R - \mathrm{Id}\big)\,(p_i - q_j)$

avec R : rotation d'axe Ω_j

d'angle $-\gamma\, \|\Omega_j \times (p_i - q_j)\|$

Ω_j γ Auto

[Velocity Skinning for Real-time Stylized Skeletal Animation.
D. Rohmer, M. Tarini, N. Kalyanasundaram, F. Moshfeghifar, M.-P. Cani, V. Zordan. Eurographics, 2021]

8 min

Follow-through

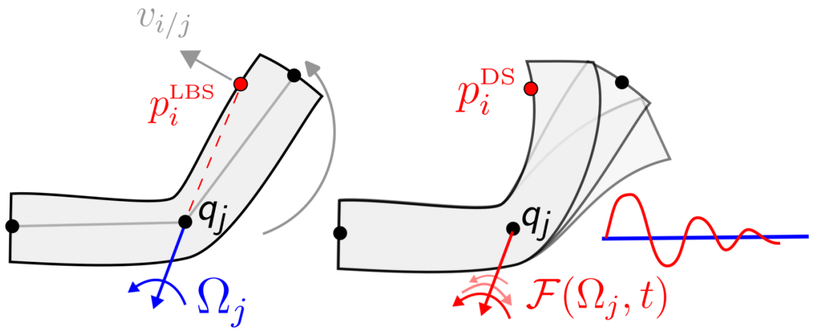
Filtrer temporellement la vitesse angulaire

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Drag : $\Omega_j = 0 \Rightarrow \psi = 0$, la peau s'arrête net, sans osciller

Idée : filtrer temporellement la vitesse angulaire

$\Omega_j(t) \longrightarrow \mathcal{F}(\Omega_j, t)$, même ψ , mêmes β_{ij}



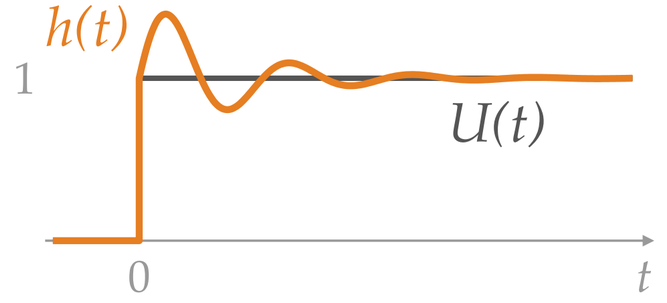
Quel filtre ?

1. **Contraintes** : linéaire, **causal**, sur une fenêtre W
2. **On prescrit** la réponse à un **échelon** $U(t)$
   $h(t) = \big(1 + \kappa(t)\big)\, U(t) \qquad \kappa(0) = 0, \quad \kappa(W) \simeq 0$
3. **On en déduit** le filtre

   réponse impulsionnelle : on dérive h, $\dot{h} = \delta + \dot{\kappa}\, U$

   réponse calculée par convolution : $\mathcal{F}(\Omega_j, \cdot) = \dot{h} * \Omega_j$

$$\mathcal{F}(\Omega_j, t) = \Omega_j(t) + \int_0^W \dot{\kappa}(u)\, \Omega_j(t-u)\, du$$



9 min

Noyau d'oscillation : une sinusoïde amortie

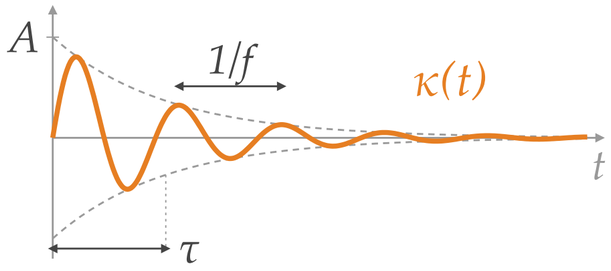
$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

$$\kappa(t) = A\, e^{-t/\tau} \sin(2\pi f\, t)$$

A : amplitude

f : fréquence

τ : amortissement



Fréquence

f = 0,75 Hz



f = 4 Hz



Amortissement

τ = 2 s



τ = 0,2 s



10 min

Résultats : une formule, des personnages variés

$$p_i^{\mathrm{DS}}(t) = \underbrace{p_i^{\mathrm{LBS}}(t)}_{\substack{\text{skinning}\\ \text{existant}}} + \sum_{\text{joints } j} \underbrace{\beta_{ij}}_{\substack{\text{rig}\\ \text{(hiérarchie)}}} \underbrace{\psi}_{\substack{\text{forme}\\ \text{(drag)}}} \Big(\underbrace{\mathcal{F}(\Omega_j, t)}_{\substack{\text{mouvement filtré}\\ \text{(oscillation)}}}, q_j, p_i^{\mathrm{LBS}} \Big)$$

Adaptation automatique : β_{ij} issus des poids de skinning



Vache



Dragon



Clown

Efficacité temps réel

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Aucune simulation : formule explicite, chaque sommet indépendant ⇒ GPU (vertex shader)

```
// CPU : une fois par image, pour chaque articulation j
F[j] = Omega[j](t) + sum_k dkappa[k] * Omega[j](t - k dt);

// GPU, vertex shader : chaque sommet i en parallèle
vec3 p_lbs = sum_j alpha[i][j] * T[j] * p0[i];
vec3 p = p_lbs;
for (j in joints(i))
    p += beta[i][j] * psi(F[j], q[j], p_lbs);
gl_Position = projection * view * vec4(p, 1.0);
```

	ms / polygone	polygones
Dynamic skinning	**0,00003**	**1,5 M**
Wu et al. *SCA 2023* Y. Wu, N. Umetani *Two-Way Coupling of Skinning Transformations and Position Based Dynamics*	0,004	4,2 k
CODY *SIGGRAPH 2023* O. Benchekroun, J. E. Zhang, S. Chaudhuri, E. Grinspun, Y. Zhou, A. Jacobson *Fast Complementary Dynamics via Skinning Eigenmodes*	0,05	0,3 M

12 min

Contrôle artistique : changer les ingrédients

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\Big(\underbrace{\mathcal{F}(\Omega_j, t)}_{\substack{\text{mouvement filtré} \\ \text{(noyau)}}} , \underbrace{q_j}_{\text{origine}} , p_i^{\mathrm{LBS}} \Big)$$

Origine q_j de la déformation

Déplacée au début d'un membre souple :
le corps reste rigide, bras et chapeau fléchissent

Centre = origine de la déformation

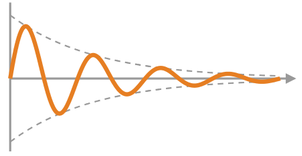


Centres locaux

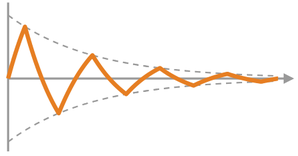


Forme du noyau d'oscillation

Mêmes A, f, τ : seule la forme change



Sinus (article)



Triangle





Extension : propager avec un retard

$$p_i^{\mathrm{DS}}(t) = p_i^{\mathrm{LBS}}(t) + \sum_{\text{joints } j} \beta_{ij}\, \psi\big(\mathcal{F}(\Omega_j, t), q_j, p_i^{\mathrm{LBS}}\big)$$

Retarder le signal selon la **distance le long du squelette**

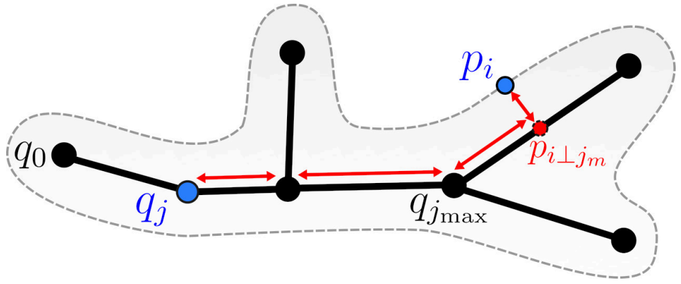
$\mathcal{F}^{delay}(x,t) = x\big(t - d_{ij}/c\big)$, c : vitesse de propagation

Delayed LBS : appliquer le retard aux transformations du LBS

cette extension modifie la **position de base** de la formule

$$p_i^{\mathrm{dLBS}} = \sum_j \alpha_{ij}\ \mathrm{T}_j\big(t - d_{ij}/c\big)\, p_i^0$$

l'effet se propage le long de la chaîne



15 min

Synthèse

$$p_i^{\mathrm{DS}}(t) = \underbrace{p_i^{\mathrm{LBS}}(t)}_{\text{base}} + \sum_j \underbrace{\beta_{ij}\,\psi}_{\text{espace}}\Big(\underbrace{\mathcal{F}(\Omega_j, t)}_{\text{temps}}, q_j, p_i^{\mathrm{LBS}}\Big)$$

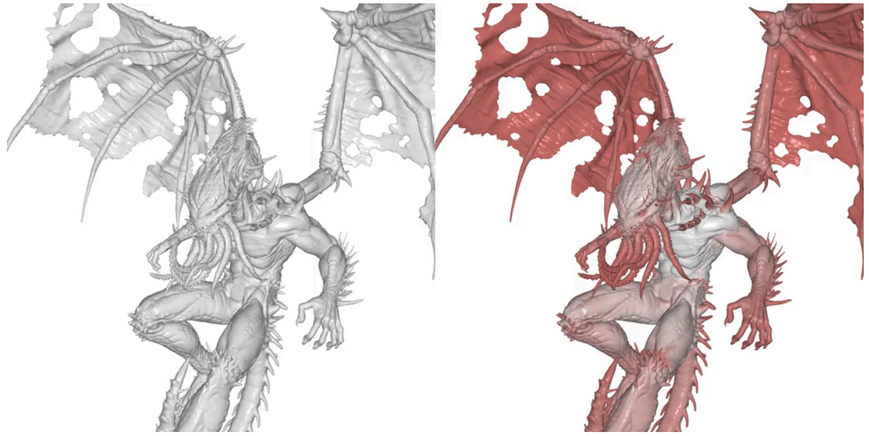
1 **LBS en vitesse**
par articulation : $\beta_{ij}\,\Omega_j \times (p_i - q_j)$ → 2 **Conversion en déformeurs**
ψ guidé par le rig → 3 **Filtre temporel** sur Ω_j
$\mathcal{F}(\Omega_j, t) = \Omega_j(t) + \sum_k \dot{\kappa}(k\Delta t)\,\Omega_j(t - k\Delta t)\,\Delta t$

Purement procédural
formule explicite, sans intégration en temps : stable et prévisible

S'appuie sur le rig existant
squelette et poids de skinning, aucune donnée en plus

Déformation par sommet
compatible GPU, coût proche du LBS

Paramètres lisibles
A, f, $\tau \cdot \gamma$, poids peints



LBS | Dynamic skinning avec poids peints

2021 **Velocity Skinning** *Eurographics*
drag, déformeurs procéduraux → 2022 **Acceleration Skinning** *Graphics Interface*
accélération, follow-through → 2026 **Dynamic Skinning** *IEEE TVCG*
filtre temporel, retard

16 min

Limites et perspectives

Limites

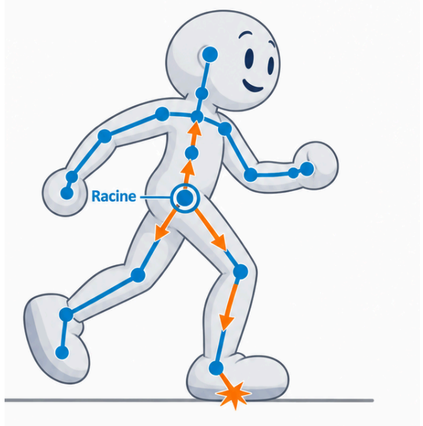
- Pas de gestion des **auto-collisions**
- Paramètres extrêmes : déformations non plausibles
- Pas de « vraie » dynamique
  - aucune réaction aux contacts ni aux forces externes
- Propagation depuis la racine du squelette



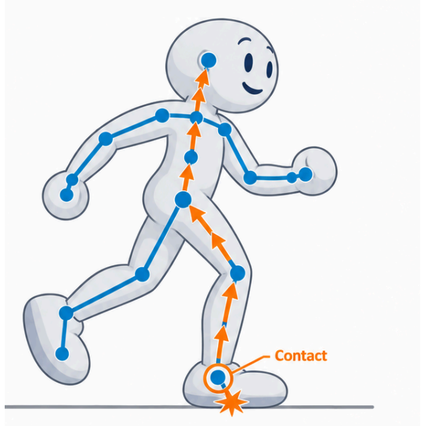
Flexion exagérée et auto-collision
(drag $\gamma = 2{,}5$)

Perspectives

- Déplacer l'**origine de la propagation** lors d'un contact
  - impact sur un pied, une main
- Nouveaux noyaux, combinaison de déformeurs
- Réglage automatique des paramètres depuis des exemples



Actuel : depuis la racine



Perspective : depuis le contact

17 min

Merci pour votre attention

Démo en ligne : dynamicskinning.com/demo

Code C++ (MIT) : github.com/drohmer/dynamic_skinning



Extension Blender (4.4+)

extensions.blender.org/add-ons/dynamic-skinning

18 min