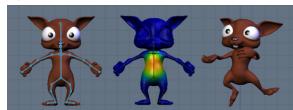


## SKINNING ET POIDS METHODES AUTOMATIQUES

Alexandre Meyer  
Equipe SAARA, Laboratoire LIRIS  
M2pro Image

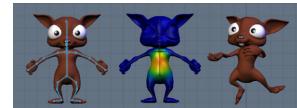


## SKINNING ET POIDS

À PARTIR D'UN UNIQUE MAILLAGE

- GÉOMÉTRIQUE
- ÉQUATION DE LA CHALEUR
- PAR TRANSFERT

À PARTIR DE PLUSIEURS MAILLAGES  
LORS DE LA MODÉLISATION



### Linear Blend Skinning : rappel

- LBS : Skinning "classique"

$$v'_i = \sum_j w_{ij} M_j v_i \quad \text{avec} \quad \sum_j w_{ij} = 1$$

où:

- $n$  est le nombre d'os influençant le sommet i
- $v_i$  est la position du sommet i
- $w_{ij}$  est le poids associé de l'os j pour le sommet i
- $M_j$  est la matrice de transformation P.B<sup>-1</sup>

### Linear Blend Skinning : rappel

- \* Problème

Attacher un sommet i aux os = calculer les poids d'influence des n os

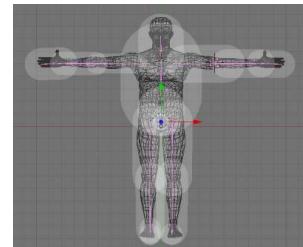
$$\text{poids du sommet } i \quad w_i = (w_{i0} \dots w_{ij} \dots w_{in})$$

## SKINNING ET POIDS

À PARTIR D'UN UNIQUE MAILLAGE  
- GÉOMÉTRIQUE

## Poids et distances : méthode de base

- Attaché chaque sommet à l'os le plus proche pondéré par la distance D (capsule autour de l'os)

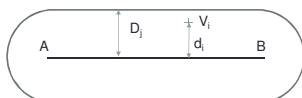


## Poids et distances : méthode de base

- Attaché chaque sommet à l'os le plus proche pondéré par la distance D (capsule autour de l'os)
- $w_{ij}$  = le poids du sommet i ( $V_i$ ) attaché à l'os j (AB sur la figure)

$$w'_{ij} = \begin{cases} \frac{d_i}{D_j} & \text{si } d_i \leq D_j \\ 0 & \text{sinon} \end{cases} \quad \text{donc } w'_i = \left( \frac{d_0}{D_0} \text{ ou } 0 \dots \frac{d_j}{D_j} \text{ ou } 0 \dots \frac{d_n}{D_n} \text{ ou } 0 \right)$$

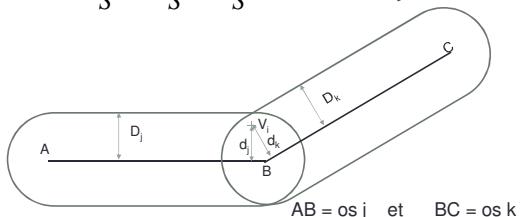
' signifie non normalisé



## Poids et distances : méthode de base

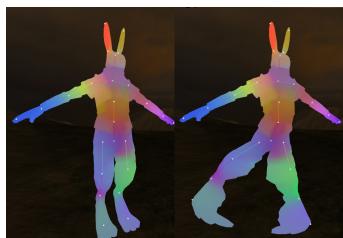
- Normalisation

$$w_i = \left( \frac{w'_{i0}}{S} \dots \frac{w'_{ij}}{S} \dots \frac{w'_{in}}{S} \right) \quad \text{avec } S = \sum_{j=1}^n w'_{ij}$$



## Poids et distances : méthode de base

- Problème : un sommet peu être influencé par un os d'un membre opposé. Voir la jambe ici :

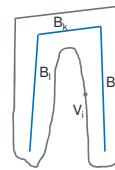


<http://blog.wolfire.com/2009/11/volumetric-heat-diffusion-skinning/>

## Poids et distances : méthode de base

- Problème
  - Un sommet peut être influencé par un os opposé
- Solutions possibles
  - Notion de visibilité entre un sommet et l'os
  - Distance calculée à l'intérieur du volume

Ici  $V_i$  peut être influencé par  $B_j$

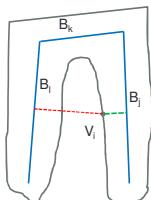


## Poids et distances : méthode de base

- Solutions possibles

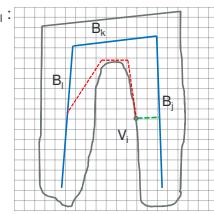
- Notion de visibilité entre un sommet et l'os
  - $B_j$  influence  $V_i$  car la droite directe ne traverse pas le maillage
  - $B_j$  n'influence pas  $V_i$  car la droite directe traverse le maillage

Peut se résoudre par lancer de rayons  
→ Test : est-ce que le segment entre  $V_i$  et le point le plus proche sur l'os traverse un triangle ?



## Poids et distances : méthode de base

- Solutions possibles
  - Distance calculée à l'intérieur du volume
    - Voxelise l'espace : voxel interieur/extérieur
    - Visibilité peut se faire sur les voxels
    - Ou calcul de distance entre  $V_i$  et os  $B_j$  : plus court chemin discret (sur la grille) entre  $V_i$  et le point le plus proche sur  $B_j$   
→ distance grande, poids nul (seuil)



**Geodesic Voxel Binding for Production Character Meshes**  
Olivier Dionne and Martin de Lasa\*  
Autodesk Inc.

Symposium on Computer Animation (SCA) 2013

Figure 1: Starting from a character skeleton and mesh (A), which may contain degenerate geometry (in red) (B), we voxelize the mesh using graphics hardware (C), and compute bind weights using geodesic distances from each bone (D). Resulting weights are applied to existing closed-form skinning methods to deform character geometry (E).

**Geodesic Voxel Binding for Production Character Meshes**  
Olivier Dionne and Martin de Lasa\*  
Autodesk Inc.

- Voxelisation : rendu en changeant le 'near' plane

Figure 3: Visualizations of the 'Buster' model. (A) The character world space anti-aligned bounding box defines the initial orthogonal view volumes. (B) For every pair of view directions in x, y and z axes we position the camera in the proper axis and slice the model by moving the near plane. (C) After compiling votes for each view using equations (5) and (6) and performing an octree hit test we obtain a voxelization of the input geometry. (D) A cross section view of the resulting voxelization. Red and blue represent internal and boundary voxels respectively.

**Geodesic Voxel Binding for Production Character Meshes**  
Olivier Dionne and Martin de Lasa\*  
Autodesk Inc.

**Algorithm 1: Distance Computation**

```

input: Character skeleton S and voxelized mesh V
foreach bone  $b_i$  of S do
    // Initialize voxel distance values
    foreach non-exterior voxel  $v_i$  of V do
         $d_{i,j} = \infty$ ;
    end
    Create empty voxel queue Q;
    // Initialize bone voxels and add to queue
    foreach non-exterior voxel  $v_i$  of V intersecting with  $b_i$  do
         $d_{i,i} = 0$ ;
        Push  $v_i$  to Q;
    end
    // Compute geodesic distances
    while Q is not empty do
        Pop  $v_i$  from Q;
        foreach non-exterior voxel neighbor  $v_j$  to  $v_i$  do
             $dist = d_{i,j} + |\mathbf{p}_{vertex} - \mathbf{p}_{voxel}|$ 
            if  $d_{i,j} > dist$  then
                 $d_{i,j} = dist$ ;
                Push  $v_j$  to Q;
            end
        end
    end
end

```

Figure 4: Computing distances through the voxelized model where white, red and blue colors represent skeleton, interior and boundary voxels.

$$d_j^i = \frac{d_{i,j} + |\mathbf{p}_{vertex} - \mathbf{p}_{voxel}|}{D}$$

$$\omega_j^i = \left( \frac{1}{(1-\alpha)(d_j^i) + \alpha(d_j^i)^2} \right)^2$$

**Geodesic Voxel Binding for Production Character Meshes**  
Olivier Dionne and Martin de Lasa\*  
Autodesk Inc.

Figure 2: Comparison of other automatic weighting schemes (B-D) with our method (E). (A) Initial bind pose. (B) Closer distance (Maya) (C) Closer hierarchy (Maya) (D) Heat map weighting [Baran and Popović 2007] (E) Geodesic voxel binding. Distance-based methods that ignore mesh structure (B,C) can produce severe artifacts. For watertight meshes we obtain similar results to Heat map weighting (D,E). See Figure 8 for more complete discussion.

+VIDEO

## Poids et distances : méthode de base

- Bilan
  - Avec la notion de visibilité, résultats bons (sans retouche manuel)
  - Manque parfois de transition douce (discontinuités) aux frontières entre les zones d'influences
- diffusion [Baran and Popovic 2007 Pinocchio]

## SKINNING ET POIDS

À PARTIR D'UN UNIQUE MAILLAGE  
- ÉQUATION DE LA CHALEUR

---

Automatic Rigging and Animation of 3D Characters  
Ilya Baran, Jovan Popović  
SIGGRAPH 2007

Code disponible sur leur page

## Automatic Rigging and Animation of 3D Characters

- Entrée
  - 1 maillage (statique)
  - 1 squelette (générique)
- Ajuste le squelette sur le maillage
- Attache le maillage au squelette
  - Skinning (Linear Blend Skinning)
  - Calcul automatique des poids par diffusion de chaleur

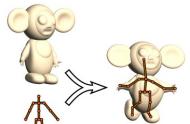


Figure 1: Our method takes a static character mesh and an input skeleton and rigs the character so that it can be animated by controlling the skeleton.

## Automatic Rigging and Animation of 3D Characters

- Ajuste le squelette sur le maillage

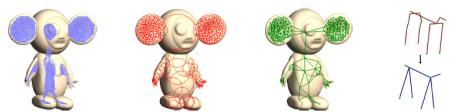


Figure 2: Approximate Medial Surface  
Figure 3: Packed Spheres  
Figure 4: Constructed Graph  
Figure 5: The original and reduced quadruped skeleton

## Equation de la chaleur

- Idée
  - Tous les sommets ont une température initiale de 0 degré
  - Un unique os émet de la chaleur (ici celui de droite en rouge)
  - Simulation de la diffusion
  - La température arrivant au sommet donne le poids



Figure 8: Top: heat equilibrium for two bones. Bottom: the result of rotating the right bone with the heat-based attachment

## Equation de la chaleur (wikipedia)

$$\forall x \in \Omega, \quad \frac{\partial T}{\partial t}(x, t) = D \Delta T(x, t) + \frac{P}{\rho c}$$

- où
  - $\Delta$  est l'opérateur [Laplacien](#) (somme des dérivées seconde),  

$$\Delta \phi = \nabla^2 \phi = \nabla \cdot (\nabla \phi) = \operatorname{div}(\overrightarrow{\operatorname{grad}} \phi) \quad \Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$
  - D est le [coefficent de diffusivité thermique](#) (en  $\text{m}^2/\text{s}$ ),
  - P une éventuelle production volumique de chaleur (en  $\text{W}/\text{m}^3$ ),
  - $\rho$  est la [masse volumique](#) du matériau (en  $\text{kg}/\text{m}^3$ ),
  - c la [chaleur spécifique](#) massique du matériau (en  $\text{J}/\text{kg}\cdot\text{K}$ ).

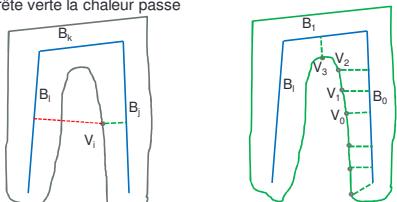
## Equation de la chaleur : volumique

- Pour calculer la diffusion de chaleur
  - Peut-être calculé en voxelisant le maillage dans une grille 3D
  - Assez précis
  - Pas forcément nécessaire



## Equation de la chaleur : maillage

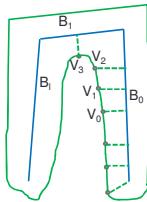
- La chaleur passe par les arêtes du maillage
- + des arêtes entre chaque sommet et les os quand il y a visibilité directe (voir transparent 11)
  - Arête rouge la chaleur ne passe pas
  - Arête verte la chaleur passe



## Equation de la chaleur : maillage

- On traite chaque os séparément
- Initialisation
  - Os j:  $w=1$  (chaud), autre os  $w=0$  (froid)
  - Température des os = constante
  - Poids du sommet i avec l'os j =  $w_{ij}=0$
- Chaleur sur tout le maillage

$$\frac{dw_j}{dt} = \Delta w_j + H(p_j - w_j)$$



## Equation de la chaleur : maillage

- Sur tout le maillage

$$\frac{dw_j}{dt} = \Delta w_j + H(p_j - w_j)$$

avec

- $p_j$  est un vecteur avec  $p_{ij}=1$  quand l'os le plus proche du sommet i est l'os j, 0 sinon
- H matrice diagonale avec  $H_{ii}$  la contribution de chaleur de l'os le plus proche du sommet i, c'est-à-dire

$$H_{ii}=c/d_i^2 \text{ si l'os } i \text{ est le plus proche, 0 sinon}$$

avec  $d_i$  la distance entre le sommet i et l'os le plus proche  
(cela correspond aux arêtes en pointillées de la figure du slide précédent)

c=constante réglée empiriquement à 0.22 dans l'article

## Equation de la chaleur : maillage

- Deux manières de calculer les poids à partir de cette éq.
- Simulation par différence finie et itérations

$$\frac{w_j^{t+1} - w_j^t}{dt} = \Delta w_j^t + H(p_j - w_j^t)$$

- Équilibre est atteint quand  $dw/dt=0$   
→ résolution du système (matrice creuse, résolution de Cholesky)  
le papier utilise la librairie TAUCS

$$\begin{aligned} \frac{dw_j}{dt} &= \Delta w_j + H(p_j - w_j) = 0 \\ \Leftrightarrow -\Delta w_j + H w_j &= H p_j \end{aligned}$$

## Equation de la chaleur : maillage

- Deux manières de calculer les poids à partir de cette éq.
- Simulation par différence finie et itérations

$$\frac{w_j^{t+1} - w_j^t}{dt} = \Delta w_j^t + H(p_j - w_j^t)$$

- Équilibre est atteint quand  $dw/dt=0$   
→ résolution du système (matrice creuse, résolution de Cholesky)  
le papier utilise la librairie TAUCS

$$\begin{aligned} \frac{dw_j}{dt} &= \Delta w_j + H(p_j - w_j) = 0 \\ \Leftrightarrow -\Delta w_j + H w_j &= H p_j \end{aligned}$$

## Equation de la chaleur : maillage

- Pour l'os 0, température os 0 = 1, autres os t=0
  - Refaire ce calcul pour os 1, 2, etc.
- Système à résoudre pour trouver l'influence de l'os 0 :

$$\begin{aligned} \Delta w_j + H(p_j - w_j) &= 0 \\ (\Delta w_0 & \quad c/d_0 \quad 0 \quad \dots \quad 1 \quad w_0) \\ \Delta w_1 & \quad c/d_1 \quad \dots \quad 1 \quad w_1 \\ \Delta w_2 & \quad c/d_2 \quad \dots \quad 1-w_2 \quad 0 \quad w_3 \\ \Delta w_3 & \quad 0 \quad \dots \quad 0 \quad w_3 \\ \dots & \quad \dots \quad \dots \quad \dots \quad \dots \end{aligned} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \end{pmatrix}$$

Laplacien

Ecart de température entre le sommet i et son os le plus proche (la source de chaleur) multiplié par l'inverse du carré de la distance

## Eq. chaleur : Laplacien

The mesh is represented as a graph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , with vertices  $\mathbf{V}$  and edges  $\mathbf{E}$ , where  $\mathbf{V} = [v_1^T, v_2^T, \dots, v_n^T]^T$ ,  $v_i = [v_{ix}, v_{iy}, v_{iz}]^T \in \mathbb{R}^3$  is the original geometry,  $\tilde{\mathbf{V}}$  denotes the displaced geometry. Furthermore,  $\delta_i$  is the Laplacian of  $v_i$ , the result of applying the discrete Laplace operator to  $v_i$ , i.e.

$$\delta_i = \sum_{(i,j) \in E} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) = \left[ \sum_{(i,j) \in E} w_{ij} \mathbf{v}_j \right] - \mathbf{v}_i, \quad (1)$$

where  $\sum_{(i,j) \in E} w_{ij} = 1$ , and the choice of weights

$$w_{ij} = \frac{\omega_{ij}}{\sum_{(i,k) \in E} \omega_{ik}} \quad (2)$$

defines the nature of  $\delta_i$ . Some popular choices are

$$\omega_{ij} = 1, \quad (3)$$

$$\omega_{ij} = \cot \alpha + \cot \beta, \quad (4)$$

where (3) are the uniform and (4) the cotangent weights. The angles used in these equations are shown in Fig. 4. In the remainder of

## Eq. chaleur : Laplacien

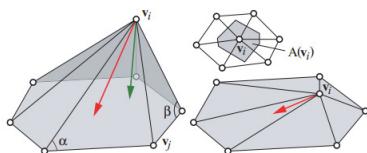


Figure 4: Left: uniform (red) and cotangent (green) Laplacian vectors for a vertex  $v_i$  and its (in this case planar) 1-ring, as well as the angles used in Eqn. 4 for one  $v_j$ . Bottom right: the effect of flattening  $v_i$  into the 1-ring plane. While the cotangent Laplacian vanishes, the uniform Laplacian generally does not. Right top: the Voronoi region  $A(v_i)$  around a vertex.

## Eq. chaleur : Laplacien

- Pour la Laplacien un choix populaire et simple est
  - Equation (3), dans deux transparents précédents
  - Moyenne des écarts avec les sommets voisins

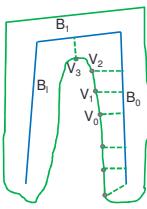
$$\Delta w_i = \frac{1}{\text{nombre de sommets voisins}} \sum_{\text{sommets voisins } k} (w_k - w_i)$$

$$\Delta w_i = w_i + \frac{\sum w_k}{\text{nombre de sommets voisins}}$$

## Eq. chaleur : Laplacien

- Sur notre exemple ca donne en notation matricielle
  - 1 sur la diagonale
  - 1 pour les voisins divisé par le nombre de voisins
  - 0 pour les autres

$$\begin{pmatrix} \Delta w_0 \\ \Delta w_1 \\ \Delta w_2 \\ \Delta w_3 \\ \dots \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 & \dots \\ 1/2 & -1 & 1/2 & 0 & \dots \\ 0 & 1/2 & -1 & 1/2 & \dots \\ 0 & 0 & 1/2 & -1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \dots \end{pmatrix}$$



## Equation de la chaleur : maillage

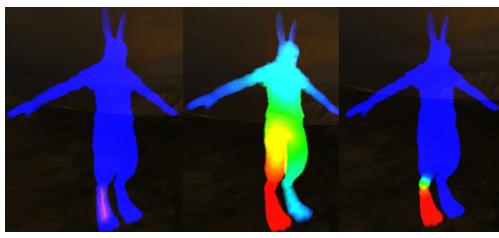
- Pour l'os 0, température os 0 = 1, autres os t=0
  - Refaire ce calcul pour os 1, 2, etc.
- Système à résoudre pour trouver l'influence de l'os 0 :

$$\Delta w_j + H(p_j - w_j) = 0$$

$$\underbrace{\begin{pmatrix} -1 & 1 & 0 & 0 & \dots \\ 1/2 & -1 & 1/2 & 0 & \dots \\ 0 & 1/2 & -1 & 1/2 & \dots \\ 0 & 0 & 1/2 & -1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}}_{\text{Laplacien}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \dots \end{pmatrix}}_{\text{Ecart de température entre le sommet } i \text{ et son os le plus proche (la source de chaleur)}} + \underbrace{\begin{pmatrix} c/d_0 & & 0 & \dots & 1-w_0 \\ c/d_1 & \dots & & 1-w_1 & 0 \\ c/d_2 & \dots & & 1-w_2 & 0 \\ 0 & \dots & & 0 & w_3 \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}}_{\text{multiplié par l'inverse du carré de la distance}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \end{pmatrix}$$

## Poids par eq. chaleur : résultats

+ VIDEO + DEMO de Pinocchio



<http://blog.wolfire.com/2009/11/volumetric-heat-diffusion-skinning/>

## SKINNING ET POIDS

À PARTIR D'UN UNIQUE MAILLAGE  
- PAR TRANSFERT

Transferring the Rig and Animations from a Character to Different Face Models.  
V. Orvalho, E. Záčur and A. Súšin, Computer Graphics Forum, 2008

Easy Rigging of Face by Automatic Registration and Transfer of Skinning Parameters. L. Dutrevet, A. Meyer, V. Orvalho, S. Bouakaz. Dans International Conference on Computer Vision and Graphics, . 2010.

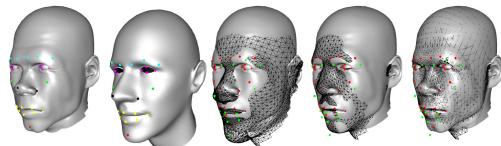
## Transfert des poids : principe

- Entrée
  - Un maillage R de référence + un squelette + les poids de skinning
  - Un maillage M à attacher à un squelette
- Besoin de points de correspondance (PdC) entre les 2 maillages
  - Entrés à la main (le plus sûr et efficace), pas nécessairement liés au squelette



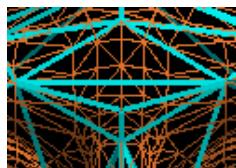
## Transfert des poids : principe

- Avec les PdC entre R et M, calcul une correspondance dense entre les 2 maillages
  - PdC définissent une déformation de l'espace entre M et R (par RBF)
  - Les sommets de M sont déformés puis projeté pour arriver sur R
- Donc chaque sommet de M à sa correspondance dans un triangle de R



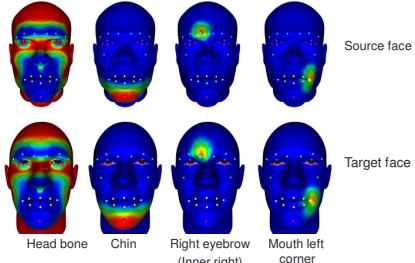
## Transfert des poids

- Les poids de chaque sommet de M sont calculés par interpolation barycentrique des poids des 3 sommets du triangle de R
  - En orange le maillage M
  - En vert le maillage R de référence avec les poids



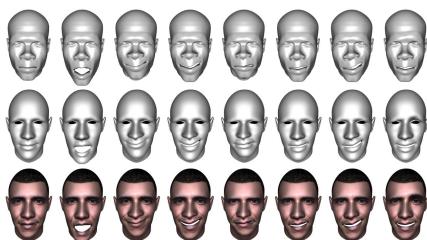
## Transfert : résultats sur un visage

- Transfert des poids de skinning d'un visage à un autre
  - Résolution des 2 maillages potentiellement différent



## Transfert : exemples d'animations

- Exemples d'animation après transfert (+VIDEO)
- Possible également sur autres maillages que visage



## SKINNING ET POIDS

À PARTIR DE PLUSIEURS MAILLAGES

Entrée : n maillages avec squelette OU juste n maillages

## Principes

- Entrée : n maillages avec squelette bien positionné
- Sortie : poids
  - Moindre carrés
  - Multi-Weight Enveloping: Least-Squares Approximation Techniques for Skin Animation. Xiaohuan Corina Wang, Cary Phillips. SCA 2002
  - un poids pour chaque case de la matrice de skinning

$$\mathbf{p}' = \sum_{k=1}^n w_k \mathbf{p} M_k \quad \mathbf{p}' = \sum_{k=1}^n \mathbf{p} \begin{bmatrix} w_{00_k} m_{00_k} & w_{01_k} m_{01_k} & w_{02_k} m_{02_k} & 0 \\ w_{10_k} m_{10_k} & w_{11_k} m_{11_k} & w_{12_k} m_{12_k} & 0 \\ w_{20_k} m_{20_k} & w_{21_k} m_{21_k} & w_{22_k} m_{22_k} & 0 \\ w_{30_k} m_{30_k} & w_{31_k} m_{31_k} & w_{32_k} m_{32_k} & 1 \end{bmatrix}$$

Skinning classique

Skinning avec poids différent pour chaque case de la matrice

## Principes

- Entrée : n maillages
- Sortie : un squelette (sans hiérarchie) + poids
  - Context-Aware Skeletal Shape Deformation. Ofir Weber, Olga Sorkine, Yaron Lipman, Craig Gotsman. Eurographics 2007 - Computer Graphics Forum
  - Fast Automatic Skinning Transformations. SIGGRAPH 2012
  - De nombreuses variantes sur ce thème

VIDEO

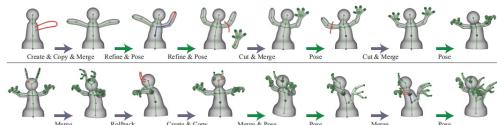
## SKINNING ET POIDS LORS DE LA MODÉLISATION

RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation  
Peter Borosan, Ming Jin, Doug DeCarlo, Yotam Gingold, Andrew Nealen  
SIGGRAPH Asia 2012.

## Lors de la modélisation

- RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation. Peter Borosan, Ming Jin, Doug DeCarlo, Yotam Gingold, Andrew Nealen. SIGGRAPH Asia 2012
  - Cylindres généralisés avec os
  - Équation de la chaleur pour les poids
  - Branchements possibles de squelettes

PPT du papier  
VIDEO  
DEMO



Embracing  
the digital  
convergence

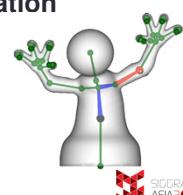
CONFERENCE 28 Nov - 1 Dec  
EXHIBITION 29 Nov - 1 Dec  
Singapore EXPO

Sponsored by ACM SIGGRAPH

[www.SIGGRAPH.org/ASIA2012](http://www.SIGGRAPH.org/ASIA2012)

## RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation

Péter Borosán  
Ming Jin  
Doug DeCarlo  
Yotam Gingold  
Andrew Nealen



48

**49 Problem**

- Creating ready-to-animate 3D models is **hard**

- Fundamental problem: static sequential pipeline [MasahiroUshiyama]

**50 Sketch-based Modeling**

- Teddy [Igarashi et al. 99]
- ShapeShop [Schmidt et al. 05]
- FiberMesh [Nealen et al. 07]
- SurfaceTrees [Schmidt and Singh 08]

**51 Automatic Rigging**

- Skeleton extraction  
[Sharf et al. 07] [Pan et al. 09] [Ju et al. 11]  
  
[Cornea et al. 07]
- Automatic skinning  
Pinocchio [Baran and Popović 07]

**52 Solution?**

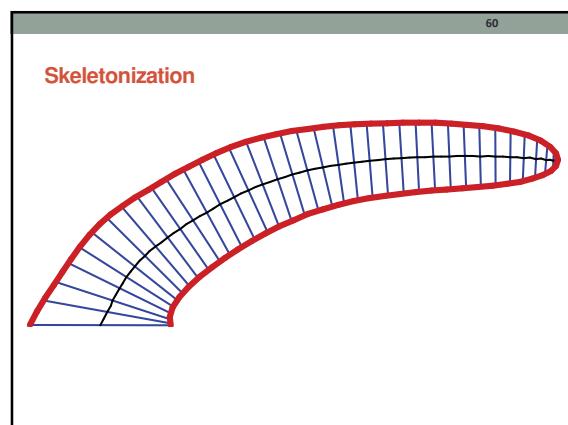
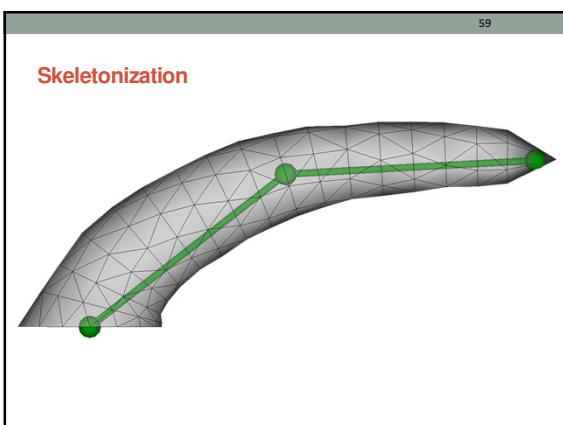
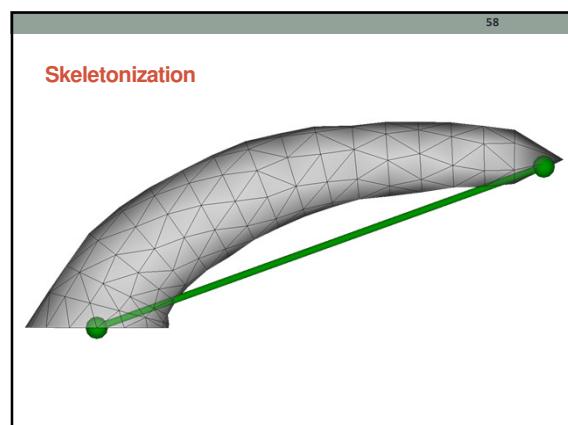
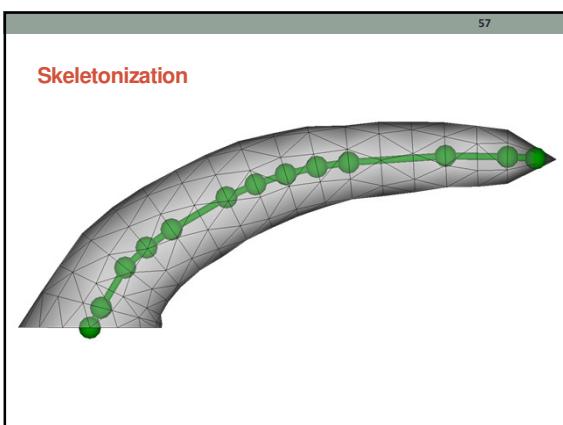
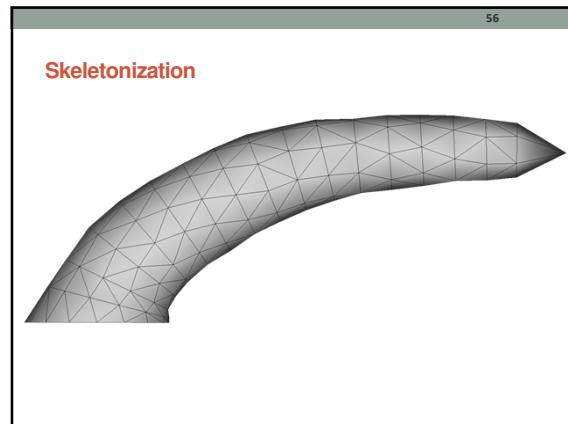
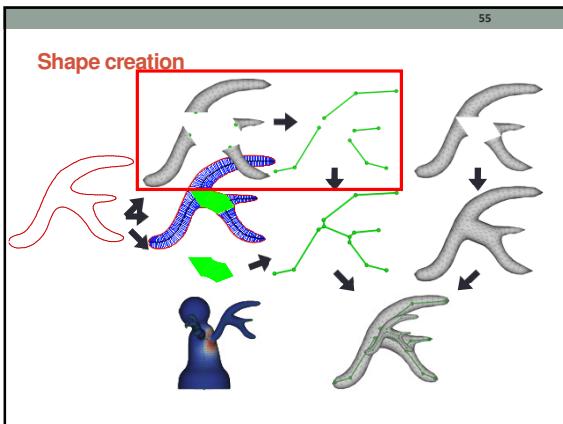
**Models rigged at all times**

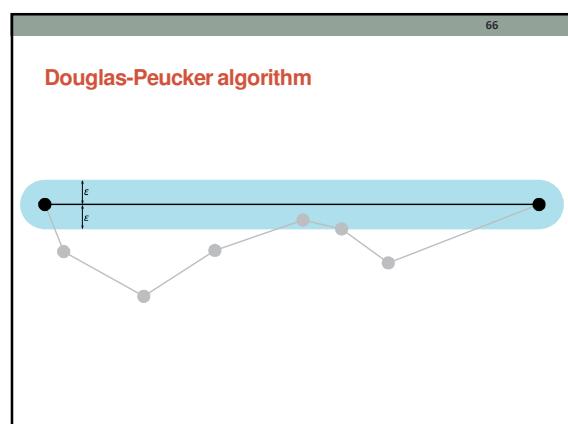
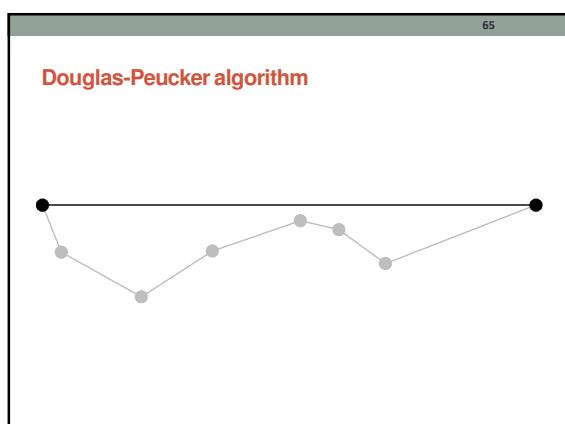
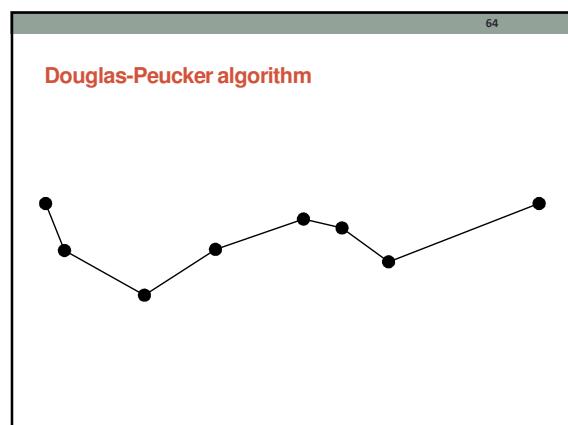
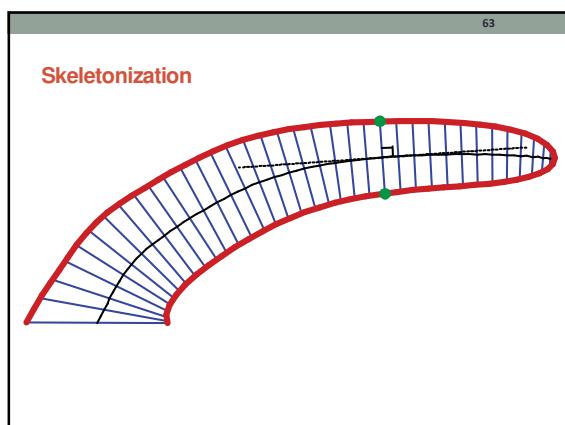
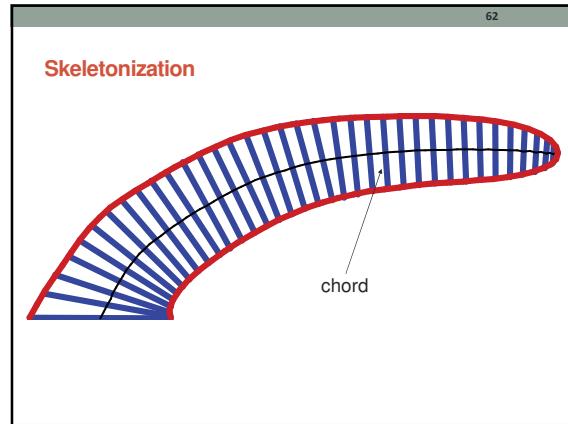
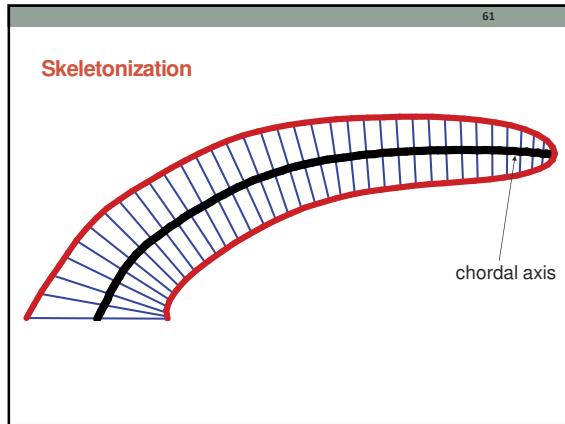
- Non-linear editing
- Modeling by parts

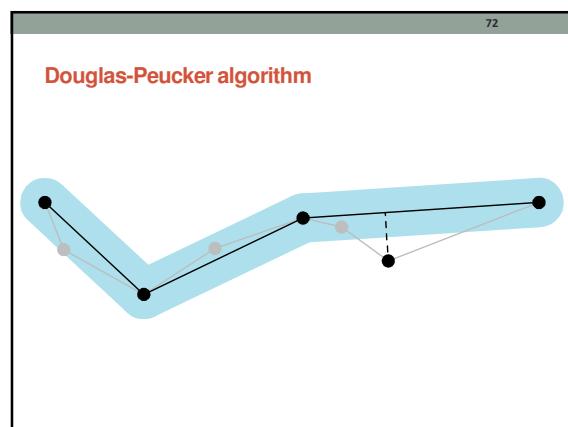
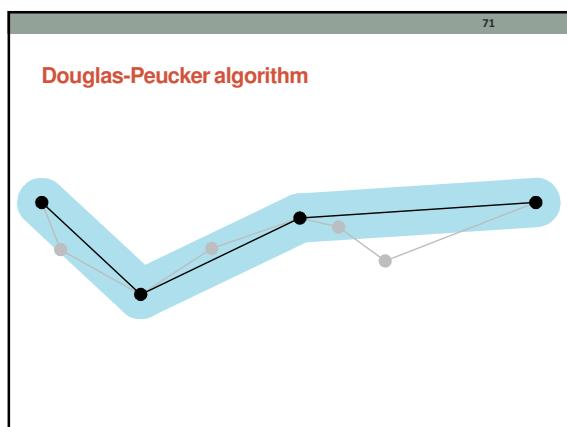
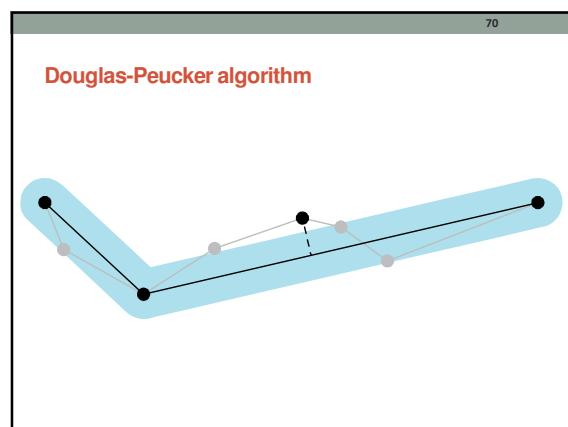
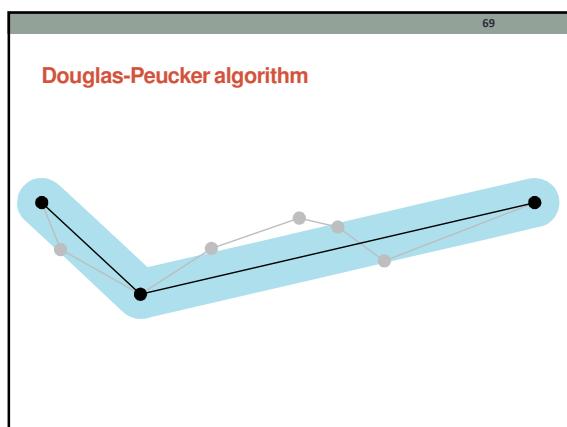
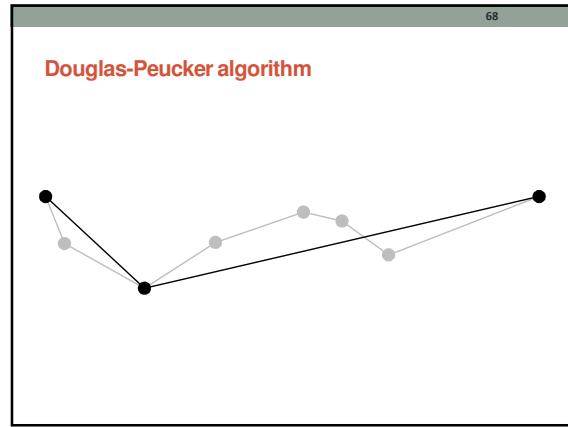
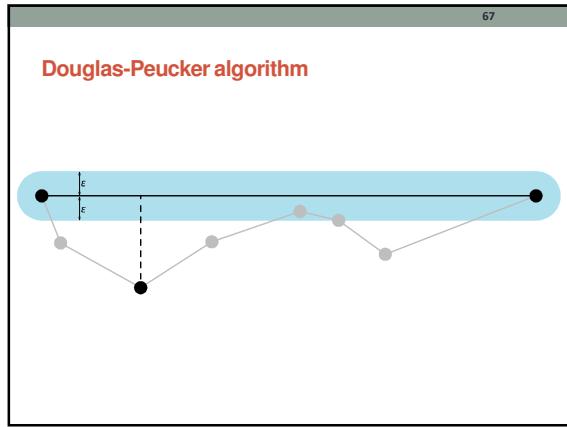
**53 RigMesh**  
Contributions

- : Sketch, Cut & Merge
- Automatically determining skeletal structure from sketch
- Models rigged at all times**
- Efficient local **non-linear editing**
- Modeling by parts

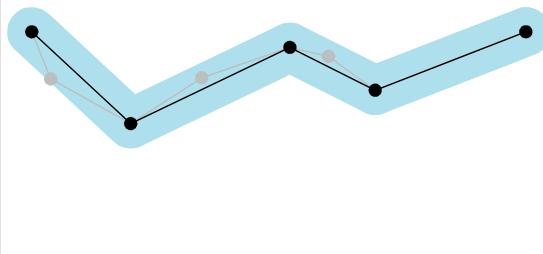
**54 Demo**



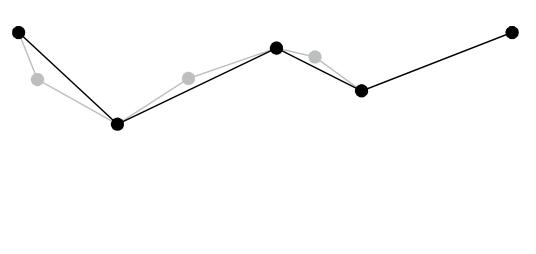




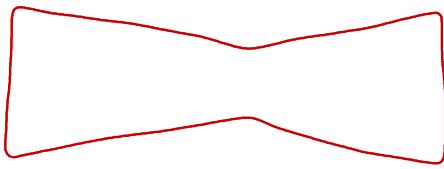
Douglas-Peucker algorithm



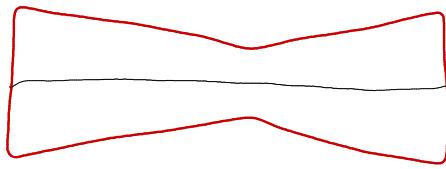
Douglas-Peucker algorithm



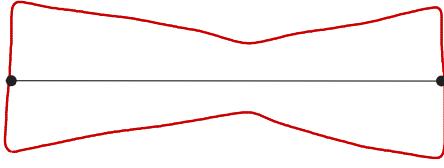
What happens if...



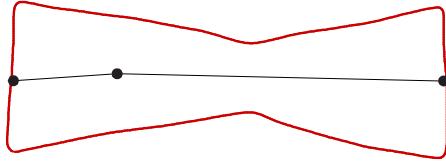
...the chordal axis is almost straight

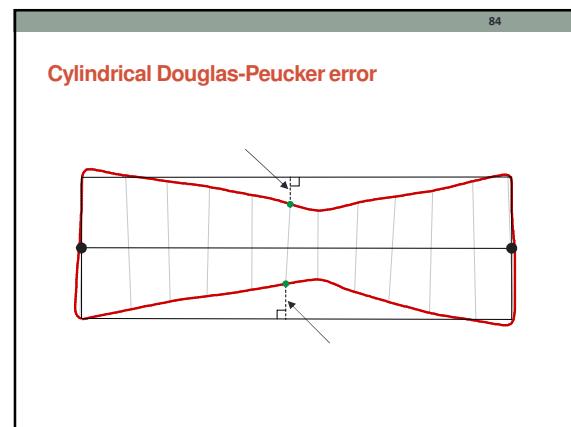
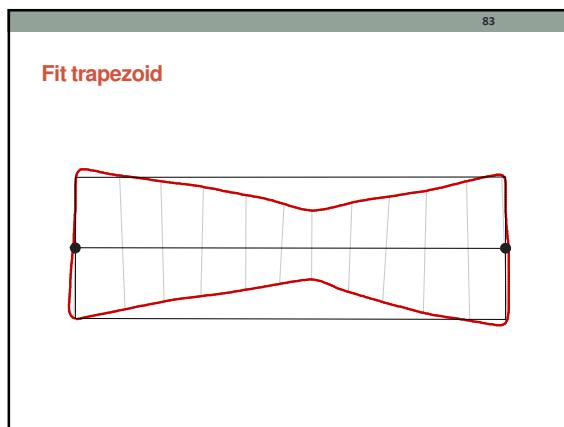
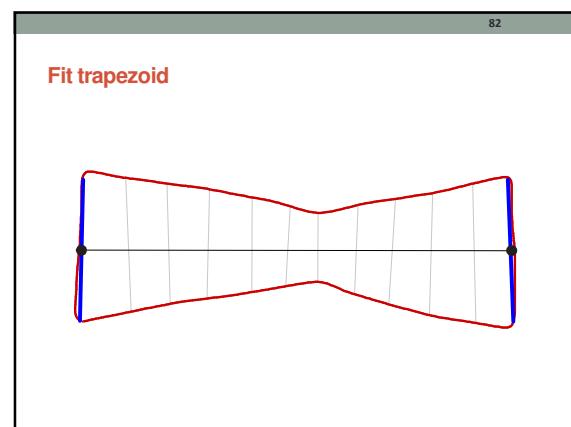
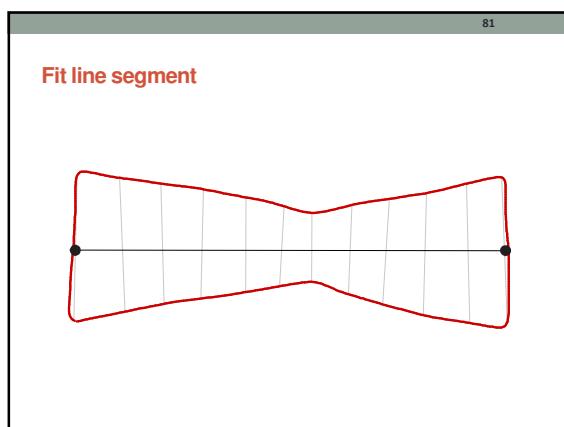
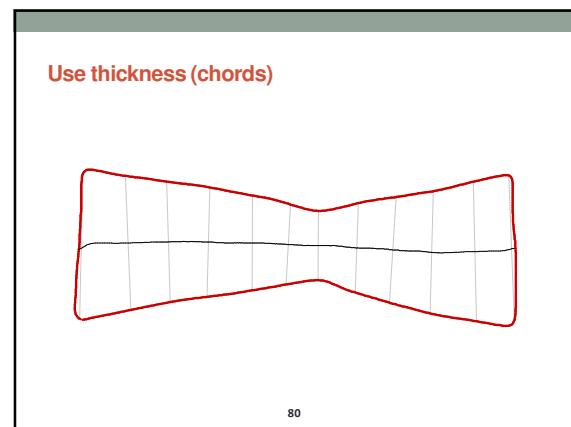
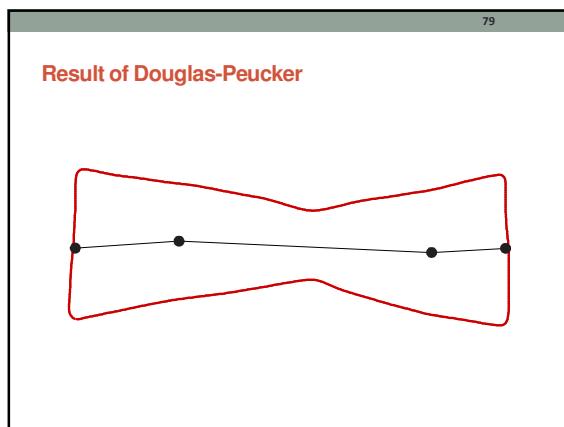


Result of Douglas-Peucker

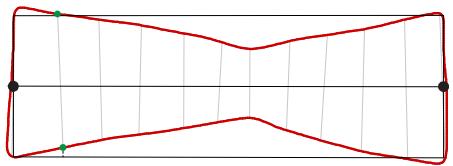


Result of Douglas-Peucker



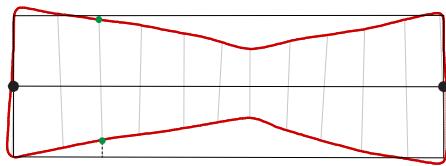


Cylindrical Douglas-Peucker error



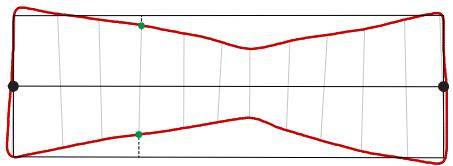
85

Cylindrical Douglas-Peucker error



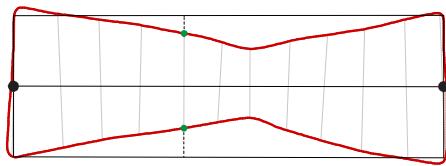
86

Cylindrical Douglas-Peucker error



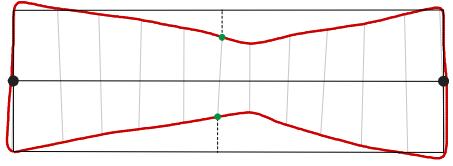
87

Cylindrical Douglas-Peucker error



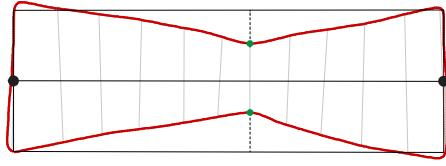
88

Cylindrical Douglas-Peucker error



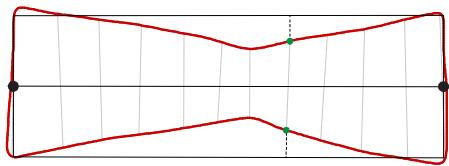
89

Cylindrical Douglas-Peucker error



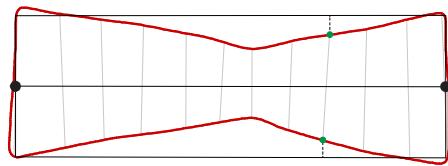
90

Cylindrical Douglas-Peucker error



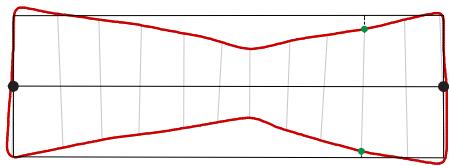
91

Cylindrical Douglas-Peucker error



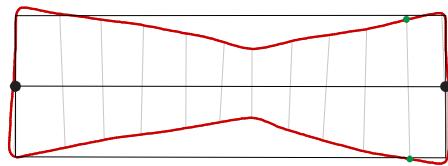
92

Cylindrical Douglas-Peucker error



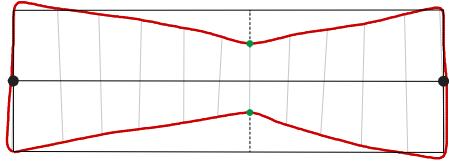
93

Cylindrical Douglas-Peucker error



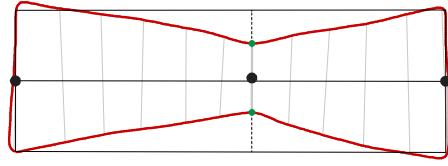
94

Cylindrical Douglas-Peucker error

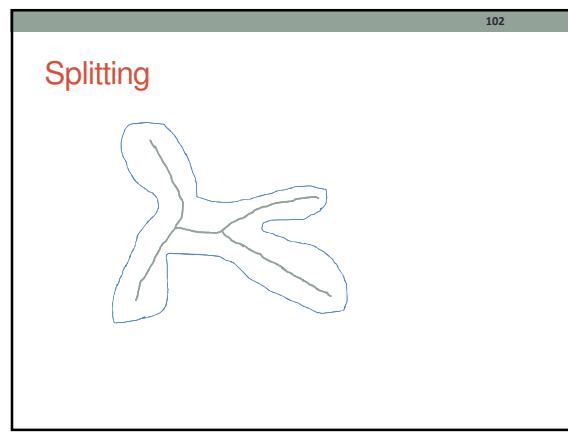
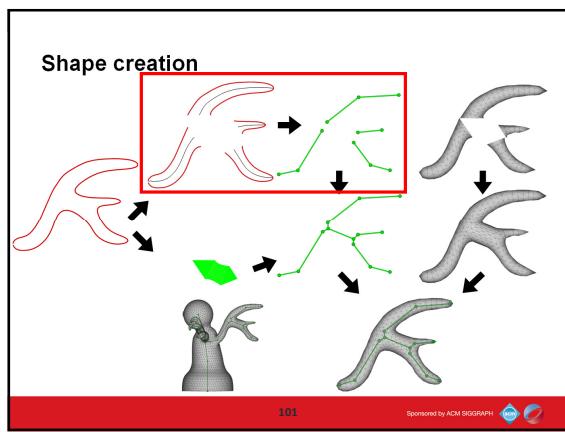
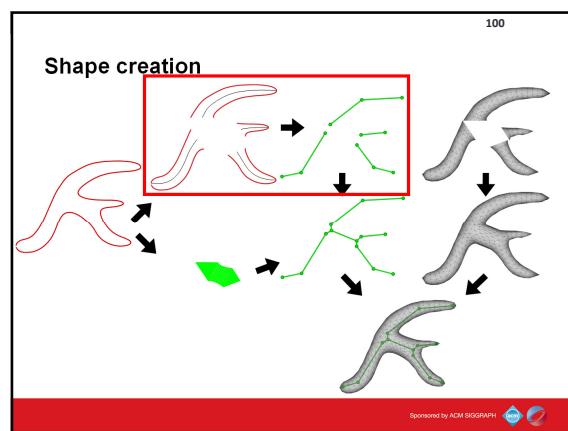
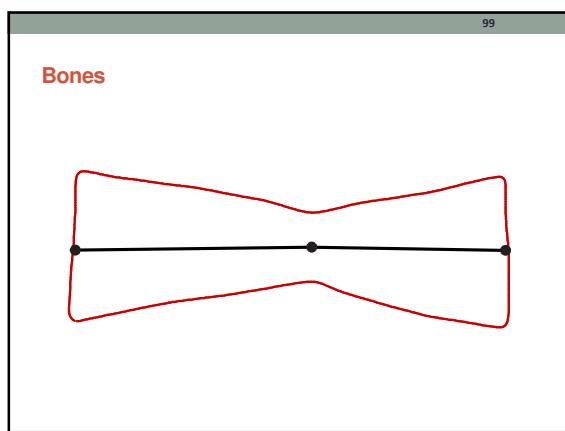
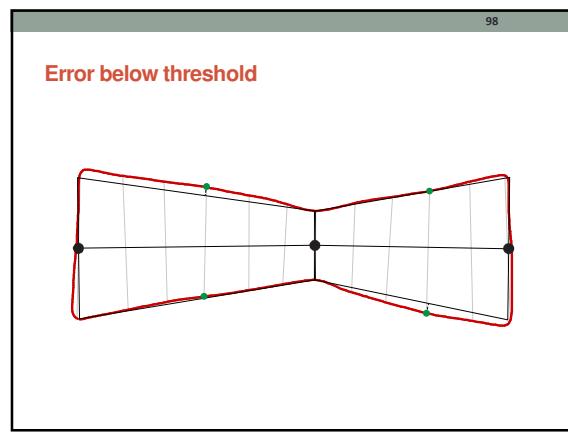
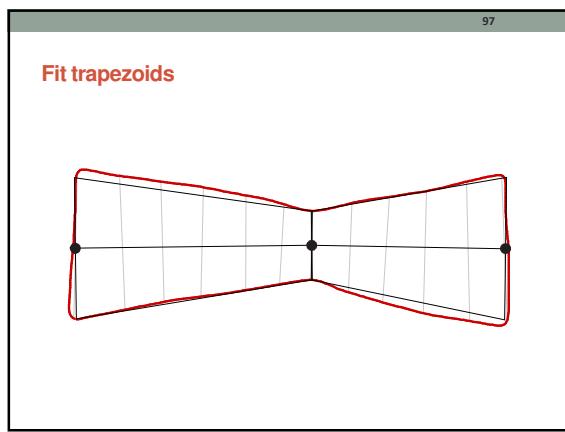


95

Add new point



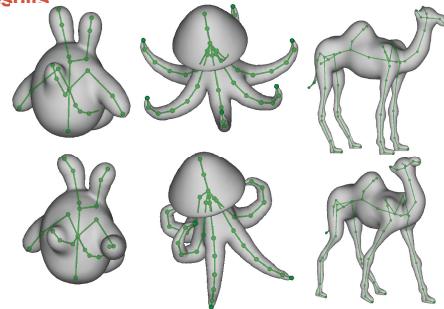
96



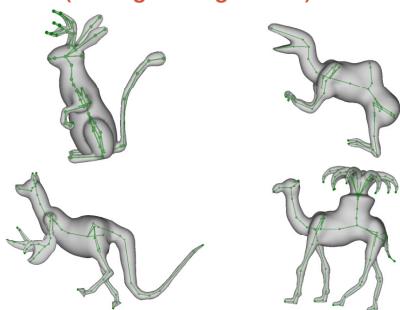
## Collapsing

- Click
- Draw lines, maybe separate

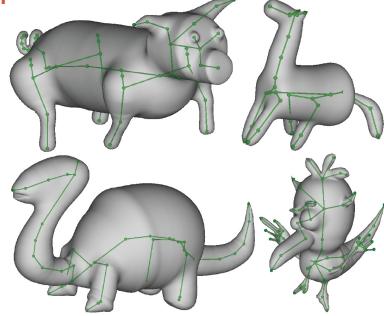
## Results



## Results (reusing existing models)



## Results (1st time usage)

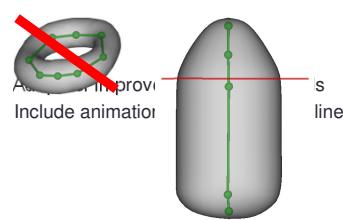


## Conclusions

Unified approach to modeling and rigging  
 Algorithm for creating high quality skeletons for sketched shapes  
 Methods for locally re-computing skin weights  
 Modeling by parts

## Limitations & Future work

Only acyclic skeletons  
 Objects without strong, stable axial symmetry



07/01/2014

