Fabricating Articulated Characters from Skinned Meshes

6.S079 Computational Fabrication
Wojciech Matusik
CSAIL & EECS MIT
The Plan for Today

• Linear Blend Skinning Review

• Fabricating Articulated Characters from Skinned Meshes
  Bächer et al., Siggraph 2012
Skinning

- We know how to animate a bone hierarchy
  - Change the joint angles, i.e., bone transformations, over time (keyframing)
Skinning

• We know how to animate a bone hierarchy
  – Change the joint angles, i.e., bone transformations, over time (keyframing)
• Embed a skeleton into a detailed character mesh
• Bind skin vertices to bones
  – Animate skeleton, skin will move with it
  – But how?
Skinning/Enveloping

• Need to infer how skin deforms from bone transformations.

• Most popular technique: Skeletal Subspace Deformation (SSD), or simply Skinning
  - Other aliases
    • vertex blending
    • matrix palette skinning
    • linear blend skinning
Each bone has a deformation of the space around it (rotation, translation)
SSD / Skinning

- Each bone has a deformation of the space around it (rotation, translation)
  - What if we attach each vertex of the skin to a single bone?
SSD / Skinning

• Each bone has a deformation of the space around it (rotation, translation)
  – What if we attach each vertex of the skin to a single bone?
    • Skin will be rigid, except at joints where it will stretch badly
SSD / Skinning

- Each bone has a deformation of the space around it (rotation, translation)
  - What if we attach each vertex of the skin to a single bone?
    - Skin will be rigid, except at joints where it will stretch badly
  - Let’s attach a vertex to many bones at once!
    - In the middle of a limb, the skin points follow the bone rotation (near-rigidly)
    - At a joint, skin is deformed according to a “weighted combination” of the bones
Examples

Colored triangles are attached to 1 bone
Black triangles are attached to more than 1

Note how they are near joints

James & Twigg 2005
Examples

Colored triangles are attached to 1 bone
Black triangles are attached to more than 1

Note how they are near joints

James & Twigg 2005
Vertex Weights

• We’ll assign a weight $w_{ij}$ for each vertex $p_i$ for each bone $B_j$.
  – “How much vertex $i$ should move with bone $j$”
  – $w_{ij} = 1$ means $p_i$ is rigidly attached to bone $j$. 
Vertex Weights

• We’ll assign a weight $w_{ij}$ for each vertex $p_i$ for each bone $B_j$.
  – “How much vertex $i$ should move with bone $j$”
  – $w_{ij} = 1$ means $p_i$ is rigidly attached to bone $j$. 

Wang & Phillips 2002
Vertex Weights

• We’ll assign a weight $w_{ij}$ for each vertex $p_i$ for each bone $B_j$.
  – “How much vertex $i$ should move with bone $j$”
  – $w_{ij} = 1$ means $p_i$ is rigidly attached to bone $j$.

• Weight properties
  – Usually want weights to be non-negative
Vertex Weights

• We’ll assign a weight $w_{ij}$ for each vertex $p_i$ for each bone $B_j$.
  - “How much vertex $i$ should move with bone $j$”
  - $w_{ij} = 1$ means $p_i$ is rigidly attached to bone $j$.

• Weight properties
  - Usually want weights to be non-negative
  - Also, want the sum over all bones to be 1 for each vertex
How to compute vertex positions?
Linear Blend Skinning

- **Basic Idea 1:** Transform each vertex $p_i$ with each bone as if it was tied to it rigidly.
Linear Blend Skinning

- **Basic Idea 1**: Transform each vertex $p_i$ with each bone as if it was tied to it rigidly.
- **Basic Idea 2**: Then blend the results using the weights.
Computing Vertex Positions

- **Basic Idea 1**: Transform each vertex $p_i$ with each bone as if it was tied to it rigidly.
- **Basic Idea 2**: Then blend the results using the weights.

\[
p'_{ij} = T_j p_i
\]

\[
p'_i = \sum_j w_{ij} p'_{ij}
\]

$p'_{ij}$ is the vertex $i$ transformed using bone $j$.
$T_j$ is the current transformation of bone $j$.
$p'_i$ is the new skinned position of vertex $i$. 
Computing Vertex Positions

Rest ("bind") pose

Bone 1: $T_1$

Vertex $p_0$ has weights $w_{01}=0.5$, $w_{02}=0.5$

Bone 2: $T_2$

"Skin"
Computing Vertex Positions

- Vertex $p_0$ has weights $w_{01}=0.5$, $w_{02}=0.5$
- Transform by $T'_1$ and $T'_2$ yields $p'_{01}$, $p'_{02}$
Computing Vertex Positions

Rest ("bind") pose

Bone 1: $T_1$
Bone 2: $T_2$

$\bullet$ Vertex $p_0$ has weights $w_{01}=0.5$, $w_{02}=0.5$

$\bullet$ Transform by $T'_1$ and $T'_2$ yields $p'_{01}$, $p'_{02}$

$\bullet$ the new position is $p'_0 = 0.5*p'_1 + 0.5*p'_2$

"Skin"
Vertex $p_0$ has weights $w_{01}=0.5$, $w_{02}=0.5$

Transform by $T'_1$ and $T'_2$ yields $p'_{01}$, $p'_{02}$

the new position is $p'_0 = 0.5*p'_1 + 0.5*p'_2$

Rest ("bind") pose

Bone 1: $T_1$
Bone 2: $T_2$

After rotations

Bone 1: $T'_1$
Bone 2: $T'_2$

"Skin"
SSD is Not Perfect

After rotations
The Plan for Today

• Linear Blend Skinning Review

• Fabricating Articulated Characters from Skinned Meshes
  Bächer et al., Siggraph 2012
Motivation

Animation software

[Maya, Autodesk]
Motivation

Animation software

[Maya, Autodesk]

3D printer

[OBJET]
The Approach
Challenges

• Rig joints
  - non-physical points
Challenges

• Rig joints
  - non-physical points
Challenges

• Rig joints
  - non-physical points
Challenges

• Rig joints
  - non-physical points

• Mechanical joints
  - volumetric entities
  - avoid overlaps between them
Challenges

• Rig joints
  - non-physical points

• Mechanical joints
  - volumetric entities
  - avoid overlaps between them
Overview

Input

Geometry

Articulation

Output
Overview

Input
Geometry
Articulation

Output
Fab. Mesh
Overview

Input
- Geometry
- Articulation

Output

Fab. Mesh

Segmentation
Overview

Input
- Geometry
- Articulation

Output

Geometry

Fab. Mesh

Joint Estimation

Joint Optimization

Segmentation
Overview

Input
- Geometry
- Articulation

Fabricated Mesh
- Joint Estimation
- Joint Optimization
- Segmentation
- Joint Carving

Output
Overview

Input
- Geometry
- Articulation

Joint Estimation

Output

Joint Optimization

Joint Carving

Segmentation
Overview

Input
- Geometry
- Articulation

Output

Joint Optimization
Joint Carving

Segmentation

Joint Estimation

Fabric Mesh
Skinned Character
Skinned Character

- Geometry
Skinned Character

- Geometry
Skinned Character

• Geometry
Skinned Character

• Geometry

• Articulation (LBS)

\[ \mathbf{v}_i' = \sum_l w_{il} \mathbf{T}_l \mathbf{v}_i \]
Skinned Character

- Geometry
- Articulation (LBS)

\[ \mathbf{v}'_i = \sum_l w_{il} \mathbf{T}_l \mathbf{v}_i \]
Skinned Character

- Geometry
- Articulation (LBS)

\[ \mathbf{v}'_i = \sum_l w_{il} T_l \mathbf{v}_i \]
Skinned Character

- Geometry
- Articulation (LBS)

\[ \mathbf{v}_i' = \sum_l w_{il} \mathbf{T}_l \mathbf{v}_i \]
Skinned Character

- Geometry
- Articulation (LBS)

\[ \mathbf{v}'_i = \sum_l w_{il} \mathbf{T}_l \mathbf{v}_i \]
Skinned Character

• Geometry

• Articulation (LBS)

\[ \mathbf{v}'_i = \sum_{l} w_{il} \mathbf{T}_l \mathbf{v}_i \]

• Appearance
  - Diffuse texture
  - Normal map

[Nehab et al. 2005]
Overview

Input
- Geometry
- Articulation

Output
- Joint Optimization
- Joint Carving

Process:
1. Fab. Mesh
2. Joint Estimation
3. Joint Carving

Segmentation
Overview

Input
- Geometry
- Articulation

Output

1. Fab. Mesh
2. Joint Optimization
3. Joint Carving
4. Joint Estimation
5. Segmentation

Overview of the process:
- Geometry and articulation are inputs.
- Fab. mesh is generated.
- Joint optimization is applied.
- Joint carving follows.
- Joint estimation is performed.
- Segmentation is done.
- The output is shown.
Estimating Rigid Parts
Estimating Rigid Parts

\[ \hat{l} = \arg \max_l w_{il} \]
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Transitions
Estimating Transitions
Estimating Transitions

transition edges
Estimating Transitions

transition edges
Estimating Transitions

\[ \omega_{j, \hat{t}_j} \quad v_j \quad \omega_{k, \hat{t}_k} \quad P_{jk} \quad v_k \]

transition point
Estimating Transitions

\[ p_{jk} = \frac{w_{j,k} + w_{k,\hat{k}}}{w_{j,k} + w_{k,\hat{k}}} v_j + \frac{w_{j,k} + w_{k,\hat{k}}}{w_{j,k} + w_{k,\hat{k}}} v_k \]
Estimating Transitions
Estimating Transitions
Estimating Transitions

Principal Component Analysis (PCA)
Estimating Transitions

\[ \bar{n} = \pm e_3 \]
Estimating Transitions

parent

child

\(p\)

\(n\)
Estimating Transitions

parent

child
Estimating Transitions
Estimating Transitions

parent → child

μ

μ
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Estimating Rigid Parts
Overview

Input
- Geometry
- Articulation

Output
- Joint Optimization
- Joint Carving

Intermediate Steps:
- Fab. Mesh
- Segmentation
- Joint Estimation
Overview

Input
- Geometry
- Articulation

Output
- Joint Optimization
- Joint Carving

Joint Estimation

Fab. Mesh

Segmentation
Filtering Transitions
Filtering Transitions
Filtering Transitions
Filtering Transitions

degenerate
Filtering Transitions

\[ (\lambda_1, e_1) \]

\[ (\lambda_2, e_2) \]

degenerate

\[ \lambda_1 > f \lambda_2 \]
Filtering Transitions
Placing Joint Centers
Medial Axis
Medial Axis

[Miklos et al. 2010]
Placing Joint Centers

[Blum 1967]

[Miklos et al. 2010]
Placing Joint Centers

[Blum 1967]

[Miklos et al. 2010]
Placing Joint Centers

[Blum 1967]

[Blum et al. 2010]
Placing Joint Centers

[Miklos et al. 2010]
Oriented Joint Locations
Overview

Input
- Geometry
- Articulation

Joint Estimation
- Fab. Mesh

Joint Optimization
- Segmentation

Joint Carving
- Output

Segmentation
- Joint Optimization

Joint Estimation
- Output
Overview

Input

Geometry

Articulation

Output

Fab. Mesh

Joint Optimization

Joint Carving

Segmentation

Joint Estimation
Manufacturing

• Additive manufacturing
Manufacturing

• Additive manufacturing
Manufacturing

- Additive manufacturing
  - build vs. support material
Manufacturing

- Additive manufacturing
  - build vs. support material
Manufacturing

- Additive manufacturing
  - build vs. support material
Manufacturing

• Additive manufacturing
  - build vs. support material
  - minimal distance $d$
Manufacturing

• Additive manufacturing
  - build vs. support material
  - minimal distance $d$
Manufacturing

- Additive manufacturing
  - build vs. support material
  - minimal distance $d$
Manufacturing

- Additive manufacturing
  - build vs. support material
  - minimal distance $d$

- Structural stability
  - identify & adjust weakest link
Manufacturing

- Additive manufacturing
  - build vs. support material
  - minimal distance $d$

- Structural stability
  - identify & adjust weakest link
  - critical area $A_{min}$
Manufacturing

- Additive manufacturing
  - build vs. support material
  - minimal distance $d$

- Structural stability
  - identify & adjust weakest link
  - critical area $A_{min}$
Ball and Socket Joint
Ball and Socket Joint
Ball and Socket Joint
Ball and Socket Joint

\[ A_1(r, h) \]
\[ A_2(r, h) \]
\[ A_3(r, h) \]
Ball and Socket Joint

$A_1(r, h)$

$A_2(r, h)$

$A_3(r, h)$
Ball and Socket Joint

$A_1(r, h)$

$A_2(r, h)$

$A_3(r, h)$
Ball and Socket Joint

$A_1(r, h)$

$A_2(r, h)$

$A_3(r, h)$

$max_{r, h} \ min_{i \in \{1, 2, 3\}} A_i(r, h)$
Ball and Socket Joint

\[
\max_{r,h} \min_{i \in \{1,2,3\}} A_i(r, h)
\]
Ball and Socket Joint
Ball and Socket Joint

opening angle $\phi(\theta)$

$\theta$
Ball and Socket Joint

opening angle $\phi(\theta)$

$A_1(r, h)$

$A_2(r, h)$

$A_3(r, h)$

$$\max_{r, h} \min_{i \in \{1, 2, 3\}} A_i(r, h)$$
Ball and Socket Joint

Opening angle $\phi(\theta)$

$A_1(r, h)$  $A_2(r, h)$  $A_3(r, h)$

$$\max_{r, h} \min_{i \in \{1, 2, 3\}} A_i(r, h)$$
Hinge Joint
Hinge Joint
Hinge Joint

$\gamma_b$  $\gamma_f$

swing angles
Hinge Joint

\[ \gamma_b \quad \gamma_f \]

swing angles
Estimating Mechanical Joints

• Automation
  - default: ball-and-socket joint
  - globally constant opening angle: $\phi(\theta) = \alpha$

• User Intervention
  - switch default to hinge
  - specify ranges
Estimating Mechanical Joints
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

\[ t = t_0 \]

- non-colliding
- colliding

stack

- \( C^t_3 \)
- \( C^t_2 \)
- \( C^t_1 \)
- \( C^t_0 \)
Resolving Collisions

\[ t = t_0 \]
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

\[ t = t_1 \]

- non-colliding
- colliding

stack

\[ C^{t_3} \]
\[ C^{t_2} \]
\[ C^{t_1} \]
\[ C^{t_0} \]
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

\[ t = t_2 \]

- non-colliding
- colliding

stack

\[
\begin{array}{c}
C^t_0 \\
C^t_1 \\
C^t_2 \\
C^t_3 \\
\end{array}
\]
Resolving Collisions

$t = t_2$

- non-colliding
- colliding

stack

<table>
<thead>
<tr>
<th>C_{3,2}^t</th>
<th>C_{3,1}^t</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_2^t</td>
<td>C_1^t</td>
</tr>
<tr>
<td>C_0^t</td>
<td></td>
</tr>
</tbody>
</table>

group split
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

$t = t_3$

- non-colliding
- colliding

stack

$C^t_2$
$C^t_1$
$C^t_0$
Resolving Collisions

- non-colliding
- colliding
Resolving Collisions

\[ t = t_4 \]

- non-colliding
- colliding

stack

outside collision
Resolving Collisions
Resolving Collisions
Resolving Collisions

infeasible joint
Resolving Collisions

exclusion
Resolving Collisions

group reset
Resolving Collisions
Resolving Collisions
Overview

Input
- Geometry
- Articulation

Output
- Joint Carving

Segmentation
Joint Estimation
Joint Optimization
Fab. Mesh

Joint Estimation - Joint Carving = Joint Optimization
Joint Carving
Joint Carving
Joint Carving
Results
Results
More Results
More Results
Summary

- Skinned meshes to toy characters
  - 3D print button

- Mechanical Joint Placement
  - non-physical rig joints
  - rigid body parts from skinning weights

- Mechanical Joint Optimization
  - geometric models of joint strength
  - collision handling

3D Print
Another Related Project [Cali et al., Siggraph Asia 2012]
That’s All for Today