Building Efficient, Accurate Character Skins from Examples

Alex Mohr    Michael Gleicher
University of Wisconsin, Madison

Karen Lin

Motivation
- Interactive animation tools using SSD
  + fast computation and small memory size
  + widely used in industry
- difficult for animator to manipulate
- artifacts (candy-wrapper collapse effect)

Goal
- easy authoring
- new poses through examples without artifacts

Approach
- Use example data set
- Add extra joints

SSD Review
- Weighted sum of key shapes
  \[ S = \sum_k w_k S_k \]

Linear Blend Skinning Artifacts
- A rotational deformation that nears 180° will result in a “candy wrapper” artifact.

Linear Blend Skinning Artifacts
- Add one new joint to same position in space with a halfway spherical linear interpolation
Linear Blend Skinning Artifacts
- Add one new joint to same position in space with a halfway spherical linear interpolation

Linear Blend Skinning Artifacts
- Muscle bulge is lost
- Observe that bulge scales according to angle of joint

Muscle Bulge Fix
- Add 4 new joints on upstream bone
- Scale along orthogonal axes v1 and v2

Muscle Bulge Fix
- Add 4 new joints on upstream bone
- Scale along orthogonal axes v1 and v2

Muscle Bulge Fix
- Add 4 new joints on upstream bone
- Scale along orthogonal axes v1 and v2

Muscle Bulge Fix
- Add 4 new joints on upstream bone
- Scale along orthogonal axes v1 and v2

\[ s = 1 + \frac{1}{2} \left( \frac{|b_1|}{|b_2|} + 1 \right) \]
Fitting the Skinning Model

- Solve for parameters of Extended SSD Model
  - Don’t need to save example data
  - Runtime memory won’t scale by # of inputs/examples
- Heuristic influence set creation
  - Speed up authoring
  - Increase performance

Examples

- A sampling of IK skeleton (joint transformations) paired with sampling of mesh surface (vertices)

Influence Set

- Joints, transformation matrix to local coordinate system of a joint ($M_{ie}$)
- Weights ($w_i$)
- Dress pose vertex position ($v_d$)

\[ v_e = \sum_{j=1}^{n} w_j M_{ie} M_{ej} v_d \]

Finding Influencing Set

- Rigidity score computation
  - Smallest rigidity score joints are added to the set
  - Found 3 to 8 joints per vertex works well
- $M_{ie}^{-1} v_e$ gives vertex in local coordinates
Finding Influence Set
- Joints, transformation matrix to local coordinate system of a joint \( (M_{i,e}) \)
- Weights \( (w_i) \)
- Dress pose vertex position \( (v_d) \)

\[
v_f = \sum_{i=1}^{n} w_i M_{i,e} M_e J^T v_d
\]

Solving Bilinear Problem
- Use alternation technique (weights and vertices)
- Ensure resulting weights are affine
  \[
  w_1 = 1 - \sum_{i=2}^{n} w_i
  \]
- Reformulate as matrix to solve for \( w \)

\[
\begin{bmatrix}
(T_{x_i} - T_{x_e}) v_x & \cdots & (T_{x_i} - T_{x_e}) v_x \\
\vdots & \ddots & \vdots \\
(T_{x_i} - T_{x_e}) v_x & \cdots & (T_{x_i} - T_{x_e}) v_x
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_n
\end{bmatrix}
= \begin{bmatrix}
v_{x_i} - T_{x_e} v_d \\
\vdots \\
v_{x_i} - T_{x_e} v_d
\end{bmatrix}
\]

Finding Influence Set
- Use singular value decomposition
- Compensates for possible rank deficient matrix

\[
\begin{bmatrix}
\sum_{i=1}^{n} w_i J_{i,1} \\
\vdots \\
\sum_{i=1}^{n} w_i J_{i,n}
\end{bmatrix}
\begin{bmatrix}
v_d
\end{bmatrix}
= \begin{bmatrix}
v_{c1} \\
\vdots \\
v_{cn}
\end{bmatrix}
\]

Results
- Video Demo
- Applications
  - Video games
  - Skin retargeting
  - Real-time high-end animation tool

Gains and Limitations
+ Doesn’t grow in size of example input
+ Compatible with current graphics hardware accelerators and existing game engines
- Poses restricted if example set too small
- Adds new joints to every part of skeleton, sometimes unnecessary
Conclusion

- Better approximation of natural body deformations
- Quick authoring with preprocessed influence sets
- Real-time animation tool

Discussion