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

$$S = \sum_k w_k S_k$$

SSD Review

- Weighted sum of key shapes

$$S = \sum_k w_k S_k$$

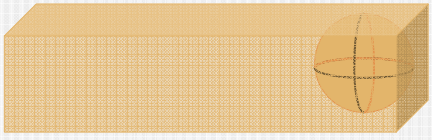
$$\bar{\mathbf{v}}_e = \sum_{i=1}^{n} w_i M_{i,e} M_{i,d}^{-1} \mathbf{v}_d$$

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

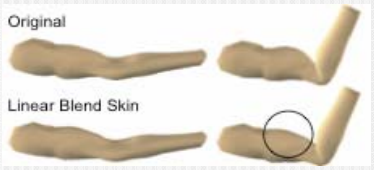


Linear Blend Skinning Artifacts

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

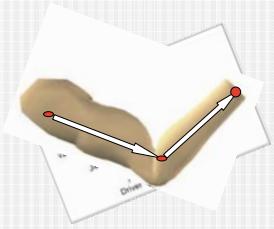
Linear Blend Skinning Artifacts

- Muscle bulge is lost
- Observe that bulge scales according to angle of joint



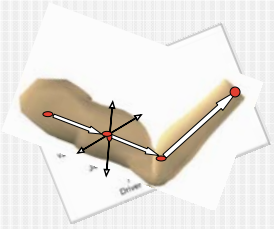
Muscle Bulge Fix

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



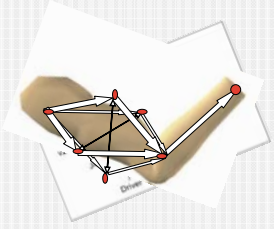
Muscle Bulge Fix

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



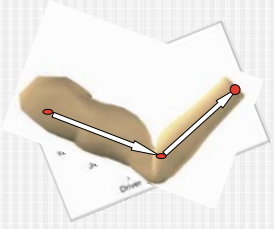
Muscle Bulge Fix

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



Muscle Bulge Fix

$$s = 1 + \frac{k}{2}\left(\frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\|\mathbf{b}_1\|\|\mathbf{b}_2\|} + 1\right)$$

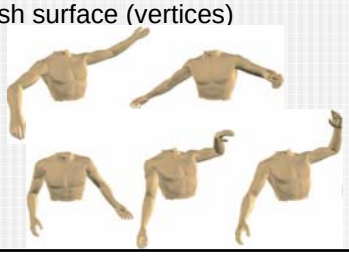


Fitting the Skinning Model

- 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 ($M_{i,e}$)
- Weights (w_i)
- dress pose vertex position ($\mathbf{v}_d$)

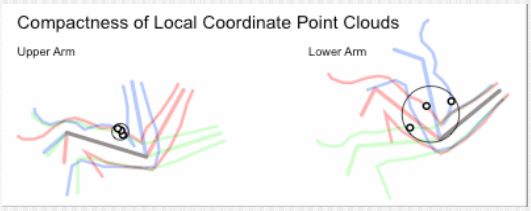
$$\bar{\mathbf{v}}_e = \sum_{i=1}^{n} w_i M_{i,e} M_{i,d}^{-1} \mathbf{v}_d$$

Influence Set

- Joints, transformation matrix to local coordinate system of a joint ($M_{i,e}$)
- Weights (w_i)
- dress pose vertex position ($\mathbf{v}_d$)

$$\bar{\mathbf{v}}_e = \sum_{i=1}^{n} w_i M_{i,e} M_{i,d}^{-1} \mathbf{v}_d$$

Finding Influencing Set



Finding Influence Set

- Rigidity score computation
  - Smallest rigidity score joints are added to the set
  - Found 3 to 8 joints per vertex works well
  - $M_{i,e}^{-1} \mathbf{v}_e$ 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 ($\mathbf{v}_d$)

$$\bar{\mathbf{v}}_e = \sum_{i=1}^{n} w_i M_{i,e} M_{i,d}^{-1} \mathbf{v}_d$$

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**

$$\begin{bmatrix} (T_{2,e_1} - T_{1,e_1})\mathbf{v}_d & \cdots & (T_{n,e_1} - T_{1,e_1})\mathbf{v}_d \\ \vdots & \ddots & \vdots \\ (T_{2,e_k} - T_{1,e_k})\mathbf{v}_d & \cdots & (T_{n,e_k} - T_{1,e_k})\mathbf{v}_d \end{bmatrix} \begin{bmatrix} w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{e_1} - T_{1,e_1}\mathbf{v}_d \\ \vdots \\ \mathbf{v}_{e_k} - T_{1,e_k}\mathbf{v}_d \end{bmatrix}$$

Finding Influence Set

- Joints, transformation matrix to local coordinate system of a joint ($M_{i,e}$)
- Weights (w_i)
- dress pose vertex position ($\mathbf{v}_d$)

$$\bar{\mathbf{v}}_e = \sum_{i=1}^{n} w_i M_{i,e} M_{i,d}^{-1} \mathbf{v}_d$$

Finding Influence Set

- Use singular value decomposition
  - Compensates for possible rank deficient matrix

$$\begin{bmatrix} \sum_{i=1}^{n} w_i T_{i,e_1} \\ \vdots \\ \sum_{i=1}^{n} w_i T_{i,e_k} \end{bmatrix} [\mathbf{v}_d] = \begin{bmatrix} \mathbf{v}_{e_1} \\ \vdots \\ \mathbf{v}_{e_k} \end{bmatrix}$$

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