Stretchable and Twistable Bones for Skeletal Shape Deformation

Alec Jacobson and Olga Sorkine
New York University and ETH Zurich
Skeleton-based skinning provides direct metaphor for character animation
Skeleton-based skinning provides direct metaphor for character animation
Linear Blend Skinning remains standard due to simplicity and efficiency

place skeleton in shape
Linear Blend Skinning remains standard due to simplicity and efficiency

place skeleton in shape

compute/paint weights
Linear Blend Skinning remains standard due to simplicity and efficiency

place skeleton in shape  
compute/paint weights  
deform bones
Linear Blend Skinning remains standard due to simplicity and efficiency

\[ p' = \sum_{i \in B} w_i(p) T_i p \]
But stretching results in shape explosion...
But stretching results in shape explosion...

LBS [Magnenat-Thalmann et al. 1988]
... and twisting must be packed at joints

LBS [Magnenat-Thalmann et al. 1988]
... and twisting must be packed at joints

LBS [Magnenat-Thalmann et al. 1988]
Fixing candy-wrapper effect is not enough

DQS [Kavan et al. 2008]
We expand deformation space to include stretching and twisting

our method
We expand deformation space to include stretching and twisting

our method
Stretchable bones statue at Ten Thousand Buddhas Monastery in Hong Kong
LBS cannot properly handle stretching...
LBS cannot properly handle stretching...

uniform scaling
LBS cannot properly handle stretching...
LBS cannot properly handle stretching

anisotropic scaling
LBS cannot properly handle stretching

anisotropic scaling

our method
Previous methods with extra weights did not solve stretching problem

LBS, DQS, [Wang and Phillips 2002], [Merry et al. 2006]

our method
LBS and previous improvements cannot twist along bone lengths

LBS, DQS, [Wang and Phillips 2002], [Merry et al. 2006]  

our method
LBS and previous improvements cannot twist along bone lengths.

LBS, DQS, [Wang and Phillips 2002], [Merry et al. 2006]  
our method
LBS and previous improvements cannot twist along bone lengths

LBS
DQS
our method
LBS and previous improvements cannot twist along bone lengths
Other improvements change too much or rely on examples

• Automatic extra bones [Mohr and Gleicher 2003]
  • Anatomically incorrect
  • Needs example poses

• Curve or spline skeletons [Fortsmann and Ohya 2006; Yang et al. 2006; Fortsmann et al. 2007]
  • Extra weights ignore input shape
  • Rigging tools and controls inconsistent with existing pipeline
Bone weights capture rigid parts well...
... but fail to control along bone lengths
Point weights stretch correctly ...
... but cannot bend like bone joints
Utilizing bone weights and point weights allows bending like bones ...
... and stretching like points
Decomposing Linear Blend Skinning exposes constant terms

\[ p' = \sum_{i \in B} w_i(p) T_i p \]
Decomposing Linear Blend Skinning exposes constant terms
Decomposing Linear Blend Skinning exposes constant terms
Decomposing Linear Blend Skinning exposes constant terms

\[ a'_i + R_i(-a_i + p) \]
Decomposing Linear Blend Skinning exposes constant terms

\[ a'_i + R_i(-a_i + p) \]
Decomposing Linear Blend Skinning exposes constant terms

\[ a_i' + R_i(-a_i + p) \]
Decomposing Linear Blend Skinning exposes constant terms

\[ T_ip \]
Anisotropic scaling term is constant for each bone weight

$$\mathbf{a}'_i + R_i(S_i(-\mathbf{a}_i + \mathbf{p}))$$
Anisotropic scaling term is constant for each bone weight

\[ a_i' + R_i(S_i(-a_i + p)) \]
Anisotropic scaling term is constant for each bone weight

\[ a_i' + R_i(S_i(-a_i + p)) \]
Anisotropic scaling term is constant for each bone weight

\[ \mathbf{a}_i' + R_i \left( S_i \left( -\mathbf{a}_i + \mathbf{p} \right) \right) \]
Anisotropic scaling term is constant for each bone weight

$$a'_i + R_i (S_i (-a_i + p))$$
Anisotropic scaling term is constant for each bone weight

\[ a_i' + R_i \left( S_i \left( -a_i + p \right) \right) \]
Anisotropic scaling term is constant for each bone weight

$$\mathbf{a}_i' + R_i(S_i(-\mathbf{a}_i + \mathbf{p}))$$

Missing information: where is $\mathbf{p}$ attached to the bone?
Replace constant scaling term with translation varied by *endpoint weights*.

\[
a'_i + R_i(e_i(p)s_i + (-a_i + p))
\]
Replace constant scaling term with translation varied by \textit{endpoint weights}.
Replace constant scaling term with translation varied by *endpoint weights*

\[ \mathbf{a}_i' + R_i(e_i(p)s_i + (-\mathbf{a}_i + \mathbf{p})) \]

\[ \mathbf{s}_i = \left( \frac{\| \mathbf{b}_i' - \mathbf{a}_i' \|}{\| \mathbf{b}_i - \mathbf{a}_i \|} - 1 \right) (\mathbf{b}_i - \mathbf{a}_i) \]
Replace constant scaling term with translation varied by *endpoint weights*

\[ a_i' + R_i(e_i(p)s_i + (-a_i + p)) \]
In 3D, use endpoint weights to blend additional twists at either endpoint.
In 3D, use endpoint weights to blend additional twists at either endpoint.

\[
K_i(t) = (1 - t)\theta_{a_i} + t\theta_{b_i}
\]

\[
a'_i + R_i K_i(e_i(p))(e_i(p)s_i + (-a_i + p))
\]
In 3D, use endpoint weights to blend additional twists at either endpoint.

\[ a'_i + \]

\[ R_i K_i(e_i(p))(e_i(p)s_i + (-a_i + p)) \]
In 3D, use endpoint weights to blend additional twists at either endpoint

\[ a_i' + R_i K_i(e_i(p))(e_i(p)s_i + (-a_i + p)) \]
Per-vertex, per-bone transformations are non-constant, but still rigid

\[ p' = \sum_{i \in B} w_i(p) \{ a'_i + R_i K_i(e_i(p))(e_i(p)s_i + (-a_i + p)) \} \]
Per-vertex, per-bone transformations are non-constant, but still rigid

\[
p' = \sum_{i \in B} w_i(p) \{ T_i(e_i(p)) + R_i(e_i(p)) p \}
\]
Per-vertex, per-bone rigid transformations may be blended as dual quaternions

\[ p' = \sum_{i \in B} w_i(p) \left\{ T_i(e_i(p)) + R_i(e_i(p)) \right\} p \]

Instead of blending transformation matrix elements linearly, blend rigid transformations as dual quaternions
Simple modification leads to powerful expression

LBS

our method extending LBS
Simple modification leads to powerful expression

LBS

our method extending LBS
Simple modification leads to powerful expression

DQS

our method extending DQS
Endpoint weights have a clear geometric meaning

Feasibly painted manually, or design automatic methods without relying on examples
Good endpoint weights maintain desirable properties

- Smooth
- Local, shape-aware
- Bound between 0 and 1
- Interpolate endpoints
- In 2D, vary linearly along bone
Endpoint weights are independent of bone weights and other endpoint weights.

\[ 1 = \sum_{i \in B} e_i(p) \]

\[ 1 = \sum_{i \in B} w_i(p) + e_i(p) \]
Endpoint weights may be conveniently defined in terms of point weights at joints.

\[ e_i = \frac{1}{2} \left( (1 - j_{a_i}) + j_{b_i} \right) \]

Point weights defined at \( a_i \)

Point weights defined at \( b_i \)
Naïve endpoint weights lack many qualities

\[ j_{IEDW_i}(p) = \sum_{k \in J} \frac{d(p, v_k)^{-\alpha}}{d(p, v_i)^{-\alpha}} \]
Naïve endpoint weights lack many qualities

\[
\mathbf{j}_{\text{IEDW}_i}(\mathbf{p}) = \frac{d(\mathbf{p}, \mathbf{v}_i)^{-\alpha}}{\sum_{k \in J} d(\mathbf{p}, \mathbf{v}_k)^{-\alpha}}
\]
Naïve endpoint weights lack many qualities

\[ j_{\text{IEDW}_i}(p) = \frac{d(p, v_i)^{-\alpha}}{\sum_{k \in J} d(p, v_k)^{-\alpha}} \]
Naïve endpoint weights lack many qualities

\[ j_{\text{IEDW}, i}(p) = \frac{d(p, v_i)^{-\alpha}}{\sum_{k \in J} d(p, v_k)^{-\alpha}} \]
Naïve endpoint weights lack many qualities

\[ e_{\text{proj}_i}(\mathbf{p}) = \frac{\|\text{proj}_i(\mathbf{p}) - \mathbf{a}_i\|}{\|\mathbf{b}_i - \mathbf{a}_i\|} \]

\[ e_{\text{proj}_i}(\mathbf{p}) = 1 \]
Naïve endpoint weights lack many qualities

\[ e_{\text{proj}_i}(p) = \frac{\|\text{proj}_i(p) - a_i\|}{\|b_i - a_i\|} \]

\[ e_{\text{proj}_i}(p) = 1 \]
Naïve endpoint weights lack many qualities

\[ e_{\text{proj}_i}(\mathbf{p}) = \frac{\|\text{proj}_i(\mathbf{p}) - \mathbf{a}_i\|}{\|\mathbf{b}_i - \mathbf{a}_i\|} \]

\[
e_{\text{proj}_i}(\mathbf{p}) = 0.5
\]
Naïve endpoint weights lack many qualities

\[ e_{\text{proj}_i}(p) = \frac{\| \text{proj}_i(p) - a_i \|}{\| b_i - a_i \|} \]

\[ e_{\text{proj}_i}(p) = 0 \]
Naïve endpoint weights lack many qualities

\[ e_{\text{proj}_i}(p) = \frac{\| \text{proj}_i(p) - a_i \|}{\| b_i - a_i \|} \]

\[ e_{\text{proj}_i}(p) = 0 \]
Recent automatic methods produce high quality endpoint weights

- [Weber et al. 2007], [Wang et al. 2007]
  - rely on extra input
- Bone heat [Baran and Popović 2007]
  - requires visibility computation
- BBW [Jacobson et al. 2011]
  - requires meshing volume and quadratic programming
Recent automatic methods produce high quality endpoint weights

- [Weber et al. 2007], [Wang et al. 2007]
  - rely on extra input
- Bone heat [Baran and Popović 2007]
  - requires visibility computation
- BBW [Jacobson et al. 2011]
  - requires meshing volume and quadratic programming
Advanced methods pay off in final quality

$e_{\text{IEDW}} \quad e_{\text{proj}} \quad e_{\text{BH}} \quad e_{\text{BBW}}$
Advanced methods pay off in final quality

$e_{IEDW}$  $e_{proj}$  $e_{BH}$  $e_{BBW}$
Good endpoint weights cannot save insufficient bone weights
Good endpoint weights cannot save insufficient bone weights

$w_{BH}$

$w_{BBW}$
Like LBS, runtime implementation is simple and efficient

- **Once per session:** load additional endpoint weights into memory
- **Each update:** pass bone transformations and extra twist parameters
- LBS or DQS form of:

\[
p' = \sum_{i \in B} w_i(p) \{ a'_i + R_i K_i(e_i(p)) (e_i(p) s_i + (-a_i + p)) \}
\]
Stretching facilitates exaggeration, a basic principle of life-like animation
Stretching facilitates exaggeration, a basic principle of life-like animation
In 2D, stretching manipulates foreshortening

LBS without allow bones to change length

STBS with *stretchable* bones
In 2D, stretching manipulates foreshortening

LBS without allow bones to change length

STBS with stretchable bones
Dragable joints simplify user interface
Dragable joints simplify user interface
Dragable joints simplify user interface
Stretchable, Twistable Bones Skinning expands space of real-time deformations

- Real-time:
  - simple and embarrassingly parallel
- Extra endpoint weights have geometric meaning
  - May be painted manually
  - Or use recent automatic methods
- Existing rigs (skeletons and bone weights) are unmodified
Future work

- Treat stretching and twisting separately
- Inverse Kinematics and procedural animation
- Fit existing mesh animations
- Explore other roles for endpoint weights
  - e.g. Muscle bulging via simple filters
We are grateful to Ofir Weber and Ilya Baran for illuminating discussions. We thank the United States Library of Congress for its collection of public domain photographs including the half-portrait of Max Schmeling. Special thanks to Felix Hornung for beautifying the teaser image.
Stretchable and Twistable Bones for Skeletal Shape Deformation

http://igl.ethz.ch/projects/skinning/stretchable-twistable-bones/

Alec Jacobson (jacobson@inf.ethz.ch)
Olga Sorkine
New York University and ETH Zurich

August 13, 2012