Delta Mush: Smoothing Deformations While Preserving Detail

Joe Mancewicz
Rhythm & Hues Studio
Delta Mush In Action: proximity binding to skeleton with delta mush smoothing.
Definitions

• Delta => tangent space displacement vector
• Mush => smoothing
• Pin Perimeter => boundary conditions
• Weight By Distance => apply Lapacian coefficients
• Prox => proximity binding (wrap deformer)
Motivation

• Rigging efficiency
  • More characters
  • Fewer riggers
  • Quicker rigs
  • Fast to setup and evaluate
  • Eliminate fix shapes and PSD
What it has become

- Added the Delta option to the Mush deformer.
- Recent refactor improved performance ~2x.
- GPU improved performance 75x.
Weighted Laplacian

\[ \vec{x}_i' = \frac{1}{N} \sum_{j=1}^{N} \vec{x}_j' \]

\[ \vec{x}_i' = \frac{1}{N+1} \left\{ \vec{x}_i + \sum_{j=1}^{N} \vec{x}_j' \right\} \]

\[ \vec{x}_i' = \frac{1}{1 + \sum_{j=1}^{N} \alpha_{ij}} \left\{ \sum_{j=1}^{N} \alpha_{ij} \vec{x}_j' \right\} \]

where \( \alpha_{ij} = \frac{1}{\| \vec{x}_i' - \vec{x}_j' \|} \)

- Standard mesh Laplacian treats all neighbors the same.
- Weighting by distance limits redistribution of points
Weighted Laplacian

- Standard mesh Laplacian treats all neighbors the same.
- Weighting by distance limits redistribution of points

\[
\vec{x}_i' = \frac{1}{N} \sum_{j=1}^{N} \vec{x}_j'
\]

\[
\vec{x}_i' = \frac{1}{N+1} \left\{ \vec{x}_i + \sum_{j=1}^{N} \vec{x}_j' \right\}
\]

\[
\vec{x}_i' = \frac{1}{1 + \sum_{j=1}^{N} \alpha_{ij}} \left\{ \sum_{j=1}^{N} \alpha_{ij} \vec{x}_j' \right\}
\]

where \( \alpha_{ij} = \frac{1}{\| \vec{x}_i' - \vec{x}_j' \|} \)
Vector displacement

\[ \vec{x}'_i = \vec{r}_i S_i^{-1} S'_i \]

or

\[ \vec{d}_i = \vec{r}_i S_i^{-1} \]

\[ \vec{x}'_i = \vec{d}_i S'_i \]
In practice

- On a 40,000 point mesh
- 15 iterations
- 4% of a frame at 24 fps (K5200)
What this lead to

• Interactive Rides
• Video Games
• Augmented Reality / Virtual Reality
Current and Future Work

- Ported our simple bone bind, proximity bind, and blend shapes to cuda
- Have characters authored in voodoo, running through our cuda deformation library, in UnrealEngine4
- Looking to port a few more key deformers
- Looking to reduce CPU<=>GPU memcpys
Go forth and implement

- Easy to implement and easy to use
- Perfect for the GPU
- Game Engines