Shape Modeling

Skeletal deformation
Believable character animation

- Computers games and movies
- Skeleton: intuitive, low-dimensional subspace
Discrete representation

- **Skeleton:**
  - collection of line segments
  - connected by joints

- **Skin:**
  - discrete samples of the surface
  - polygonal mesh
Skin + skeleton

- Skeleton defines the overall motion
- Skin moves with the skeleton

The process of building the skeleton and binding it to the skin mesh is called **rigging**.
Skeletal subspace deformation (SSD)

The artist needs to specify, for each point on the skin, how much it is influenced by the skeleton bones.
Skeletal subspace deformation (SSD)

- Affine combination of transformations

\[ v_j' = \sum_{k=1}^{K} w_{kj} \cdot T_k \cdot v_j \]

- De facto standard for interactive applications – simple + fast + works on the GPU
Skeletal subspace deformation (SSD)

- Hard to set up
- Visual artifacts
- No context
Pose space deformation (PSD)

[Lewis et al. 2000, Sloan et al. 2001]

Each degree of freedom of the skeleton is a dimension:

\[ \mathbf{P} = (\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, \ldots, \alpha_K, \beta_K, \gamma_K) \]
Pose space deformation (PSD)

Radial Basis functions:

\[ a_j(P) = h_j(P) + \sum_{i=1}^{D} m_{i,j} \Phi_i(\|P - P_i\|) \]

Deformed Shape
Pose space deformation (PSD)
PSD limitations

- SSD – artifacts, requires many examples + setup
- Linear displacements – no rotation
- High memory consumption, performance
Rotation interpolation and extrapolation
Linear displacements (PSD)
Context-Aware Skeletal Shape Deformation

Eurographics 2007

Ofir Weber
Olga Sorkine
Yaron Lipman
Craig Gotsman
The contributions

- Replace SSD by detail-preserving mesh deformation

- Easy setup

- Differential morphing

- Sparse representation of example shapes
Other previous work


  Survey: [Botsch and Sorkine 2008]

- **MeshIK** [Sumner et al. 2005, Der et al. 2006]

- **SCAPE** [Anguelov et al. 2005]
Detail-preserving deformation

\[ \Delta w_k = 0 \]

Dirichlet boundary conditions:
\[ w_k(t_n) = 1 \text{ for } t_n \in H_k \]
\[ w_k(t_n) = 0 \text{ for } t_n \in H_l \text{ where } l \neq k. \]
Blending rotations

For each face $t$:

$$R(t) = w_1(t)R_1 \oplus w_2(t)R_2 \oplus \ldots \oplus w_K(t)R_K$$

$\oplus$: [Buss 93] log-quaternion

Poisson equation [Yu et al. 2004]

$$\Delta [x\ y\ z] = \text{div}[R]$$

Sparse linear system
Poisson stitching

- The Poisson equation averages the different vertex positions
- Tries to preserve the shape and orientation of the triangles as much as possible
Poisson stitching

- The Poisson equation averages the different vertex positions
- Tries to preserve the shape and orientation of the triangles as much as possible
Comparison to SSD

Context-aware SSD

SSD
Comparison to SSD

SSD

CASSD

Video: 0:0:18
Using context – examples
Relative encoding

\[ A(t) = T(t) \times R(t) \]
\[ T(t) = A(t) \times R^T(t) \]
Relative encoding

\[ A(t) = T(t) \times R(t) \]

\[ T(t) = A(t) \times R^T(t) \]

\[ R^T(t) \]

applied to the example shape (+ stitched)

\[ = T(t) \]
Blending transformations

Polar Decomposition

\[ T_0 \]
\[ Q_0 S_0 \]
\[ T_1 \]
\[ Q_1 S_1 \]
\[ T_2 \]
\[ Q_2 S_2 \]
\[ \vdots \]
\[ \vdots \]
\[ T_D \]
\[ Q_D S_D \]

\[ R(t) \]

\[ (\bigoplus a_j Q_j) \left( \sum_{j=0}^{D} a_j S_j \right) \]
Smooth Difference

Deformation without Examples

Smooth

Example
Compact Representation

- Transformations varies smoothly
- Laplace equation
- Less than 5% memory
- Evaluation only at anchors – performance
- Greedy selection

\[ \Delta T = 0 \]

Boundary conditions:
known T’s at anchors

See Least-squares Meshes
[Sorkine and Cohen-Or 2004]
One more result...
Conclusions

• Detail-preserving skeletal shape deformation
• Easy setup
• No or small number of examples
• Interpolation and meaningful extrapolation
• Sparse representation of examples
Limitations and extensions

- No dynamics
- The greedy algorithm is not optimal
- Map to GPU → Wang et al. SIGGRAPH 2007