Subspace Gradient Domain Mesh Deformation

Scott Schaefer
Contributions

- Framework for constrained deformation
  - Skeletal constraints
  - Volume preservation
  - Projection-based manipulation
  - Detail preservation function
  - Fast non-linear, sub-space solver
Skeletal Constraints
Constructing Bones

- User drags a line in screen space
- For each pixel
  - Find first two intersections with surface
- Fit a least squares line to all midpoints
Region of Influence

- Construct supporting planes (normal perpendicular to \( ab \)) at end-points
- Flood from intersection triangles outward until all connected
Mathematical Constraint

- For each sample point along ab
  - Compute MV coordinates with respect to vertices in region of influence
  - Close mesh by fanning to centroid on ends
Volume Preservation

with constraint

no constraint
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area

\[ \frac{1}{2} \begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} \]
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area
Calculating Area

\[
\begin{vmatrix}
1 & x_0 & y_0 & 1 \\
\frac{1}{2} & x_1 & y_1 & 1 \\
\frac{1}{2} & x_2 & y_2 & 1
\end{vmatrix}
\]
Calculating Area

\[
\frac{1}{2} \begin{vmatrix}
  x_0 & y_0 & 1 \\
  x_1 & y_1 & 1 \\
  0 & 0 & 1 \\
\end{vmatrix}
\]
Calculating Area

\[
\text{area} = \sum_i \frac{1}{2} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}
\]
Calculating Volume

\[\text{volume} = \sum_{i} \frac{1}{6} \begin{vmatrix} x_{i,1} & y_{i,1} & z_{i,1} \\ x_{i,2} & y_{i,2} & z_{i,2} \\ x_{i,3} & y_{i,3} & z_{i,3} \end{vmatrix}\]
Projection-Based Manipulation
Non-linear Laplacian coordinates

\[-\kappa n \propto \sum_{i} \frac{1}{2} (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) (p_i - p_0)\]
Detail Preservation

- **Non-linear Laplacian coordinates**

\[-\alpha \bar{k} n = \sum_{i} \frac{1}{2}(\cot(\theta_{r,i}) + \cot(\theta_{l,i}))(p_i - p_0)\]
Detail Preservation

- Non-linear Laplacian coordinates

\[
p_0 = \frac{\sum_i \frac{1}{2} (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i + \alpha \bar{k} n}{\sum_i \frac{1}{2} (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))}
\]
Detail Preservation

- **Non-linear Laplacian coordinates**

\[
p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + \nu
\]
Detail Preservation

- Non-linear Laplacian coordinates

\[ p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v \]
Detail Preservation

- **Non-linear Laplacian coordinates**

\[
p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v
\]
Detail Preservation

- Non-linear Laplacian coordinates

\[ p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v \]
Detail Preservation

- Non-linear Laplacian coordinates

\[ \sum_{i} \alpha_{i} (p_{i} - p_{0}) \times (p_{i+1} - p_{0}) = v \]

Find \( \alpha_{i} \) through pseudoinverse
Non-linear Laplacian coordinates

\[
\hat{p}_0 = \sum_i \beta_i \hat{p}_i + \frac{\sum \alpha_i (\hat{p}_i - \hat{p}_0) \times (\hat{p}_{i+1} - \hat{p}_0)}{\sum \alpha_i (\hat{p}_i - \hat{p}_0) \times (\hat{p}_{i+1} - \hat{p}_0)} |\mathbf{v}| 
\]
Non-linear Minimization
Subspace Solver

- Construct a low-res approximation of mesh
- Express constraints in terms of MV coordinates
Subspace Solver
Subspace Solver

- Constraints are on the high-res mesh... NOT the low-res mesh
- A variant of a multi-grid solver
- Speeds up convergence and helps stability
Subspace Solver

- Constraints are on the high-res mesh... NOT the low-res mesh
- A variant of a multi-grid solver
- Speeds up convergence and helps stability

<table>
<thead>
<tr>
<th>model</th>
<th># vertices (original mesh)</th>
<th># vertices (coarse mesh)</th>
<th>full space</th>
<th>subspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armadillo</td>
<td>30,002</td>
<td>220</td>
<td>2.8</td>
<td>9.1</td>
</tr>
<tr>
<td>Horse</td>
<td>14,285</td>
<td>427</td>
<td>6.9</td>
<td>8.2</td>
</tr>
<tr>
<td>Tweety</td>
<td>10,240</td>
<td>286</td>
<td>12</td>
<td>23.8</td>
</tr>
<tr>
<td>Dinosaur</td>
<td>10,002</td>
<td>159</td>
<td>9.5</td>
<td>34.5</td>
</tr>
<tr>
<td>DNA</td>
<td>19,184</td>
<td>194</td>
<td>NA*</td>
<td>16.7</td>
</tr>
<tr>
<td>Santa</td>
<td>25,777</td>
<td>448</td>
<td>NA*</td>
<td>5.3</td>
</tr>
</tbody>
</table>
Resolving low-res mesh affects quality of deformation.
Results
Conclusions

- Most useful deformation constraints involve non-linear functions
- MV coordinates accelerate solving
- Don’t be afraid of non-linear optimization!!!