Mikhail Bessmeltsev

Image from ‘As-Rigid-As-Possible Shape Modeling’ by Sorkine & Alexa, 2007
What for?

• Animation!
• Mesh editing
• Image warping (2D)
Warning:

TMI

This topic is immense
We’ll only see a few samples
Deformation: user interface

• Handles
• Cages
• Skeletons
• ...

Ju et al., SIGGRAPH 2007

More on those later
Deformation models

Direct

\[ v' = (\sum w_j T_j) v \]

- Linear Blend Skinning
- Dual Quaternion Skinning
- ...

Variational

\[ v' = \arg\min_x E(x) \]

- Multiresolution editing
- As-Rigid-As-Possible
- Laplacian Mesh Editing
- ...


Deformation models

Direct

\[ \mathbf{v}' = \left( \sum w_j T_j \right) \mathbf{v} \]

• Linear Blend Skinning
• Dual Quaternion Skinning
• ...

Variational

\[ \mathbf{v}' = \arg \min_x E(x) \]

• Multiresolution editing
• As-Rigid-As-Possible
• Laplacian Mesh Editing
• ...

Deformation: user interface

- **Handles**
- Cages
- Skeletons
- ...

![Diagram showing deformation process](image.png)
Modeling

Paint three surface areas:

• Constrained
• Smooth falloff
• Fixed
Formulation

Find displacement vector field $d$

- Smooth
- Satisfies constraints

$$S' = \{ p + d(p) | p \in S \}$$
Simplest idea

- $d = s(p) \cdot D$
- $s(p)$ is a smooth function:
  - $-1$ on green vertices
  - $0$ on grey ones

\[ d \text{ known } = D \]
\[ d = 0 \]
Geodesic distance?

Reality vs Expectation
How to find $s(p)$?

- Something inversely proportional to geodesic distance
- Or our favorite:

\[
\Delta s(p_i) = 0, \quad p_i \in \mathcal{R}, \\
s(p_i) = 1, \quad p_i \in \mathcal{H}, \\
s(p_i) = 0, \quad p_i \in \mathcal{F}.
\]
Physically-Based

Find a deformation that preserves both fundamental forms

Express the fundamental forms of $S'$ via vector field $d$

\[ \int_{\Omega} k_s \left\| \mathbf{I} - \mathbf{I}' \right\|_F^2 + k_b \left\| \mathbf{I} - \mathbf{I}' \right\|_F^2 \, dudv \]

Expensive to optimize!
Shell-Based Deformation

Find a deformation that preserves both fundamental forms

**Linearize** Express the fundamental forms of $S'$ via vector field $d$

$$\int_{\Omega} k_s \left( \|d_u\|^2 + \|d_v\|^2 \right) + k_b \left( \|d_{uu}\|^2 + 2 \|d_{uv}\|^2 + \|d_{vv}\|^2 \right) dudv$$

- stretching
- bending
Physically-Based

\[
\int_{\Omega} k_s \left( \|d_u\|^2 + \|d_v\|^2 \right) + k_b \left( \|d_{uu}\|^2 + 2\|d_{uv}\|^2 + \|d_{vv}\|^2 \right) \, dudv
\]

stretching \hspace{1cm} bending

Gateaux derivative =>

\[-k_s \Delta d + k_b \Delta^2 d = 0\]
Physically-Based

\[
\int_{\Omega} k_s \left( \| d_u \|^2 + \| d_v \|^2 \right) + k_b \left( \| d_{uu} \|^2 + 2 \| d_{uv} \|^2 + \| d_{vv} \|^2 \right) dudv
\]

- **Streching**: \( k_s \left( \| d_u \|^2 + \| d_v \|^2 \right) \)
- **Bending**: \( k_b \left( \| d_{uu} \|^2 + 2 \| d_{uv} \|^2 + \| d_{vv} \|^2 \right) \)

Gateaux derivative =>

\[-k_s \Delta d + k_b \Delta^2 d = 0\]

\( (x, y, z) \)

Bi-Laplacian
Deformation Energies

Initial state

$\Delta p = 0$

$\Delta^2 p = 0$

$\Delta d = 0$

(Membrane)

$\Delta^2 d = 0$

(Bilaplacian)
Deformation Energies

Initial state

$\Delta p = 0$

$\Delta^2 p = 0$

Higher order => more boundary conditions

Initial state

$\Delta d = 0$

(Membrane)

$\Delta^2 d = 0$

(Bilaplacian)
Solved?

- Very fast
  - One linear solve!
- Physically-based
- Linearization => lose details

Original

Linear deformation

Non-linear deformation
Issue

• We need to rotate details
  • Local rotation is nonlinear!

• Can we still survive with linear solves?

Original  Linear deformation  Non-linear deformation
Multiresolution Editing

Frequency decomposition

Change low frequencies

Add high frequency details, stored in local frames
Multiresolution Editing

Decomposition -> Multiresolution Modeling -> Detail Information

Freeform Modeling

Reconstruction
How to represent details?

• For example, normal displacements
Global deformation with intuitive detail preservation
Limitations

Neighboring displacements are not coupled
- Surface bending changes their angle
- Leads to volume changes or self-intersections
Limitations

Neighboring displacements are not coupled
- Surface bending changes their angle
- Leads to volume changes or self-intersections
New coordinates?

Express shape in *differential coordinates*

Transform those, then reconstruct the new shape
Mean Value Property

\[ L_{vw} = A - D = \begin{cases} 
1 & \text{if } v \sim w \\
-\text{degree}(v) & \text{if } v = w \\
0 & \text{otherwise}
\end{cases} \]

\[
(Lx)_v = 0
\]

Value at v is average of neighboring values
Laplacian Mesh Editing

Graph Laplacian:

$$\delta_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j$$

$$\delta = L\mathbf{v}$$
Laplacian Mesh Editing

• Represent mesh using only $\delta$
• Find a surface whose Laplacian coordinates are as close as possible to $\delta$

$$\int_S \| \Delta p' - \delta' \|^2 \, dS \rightarrow \min$$

s.t. $p'_i = p_i, i \in \{\text{point constraints}\}$
Laplacian Mesh Editing

Find a surface whose Laplacian coordinates are as close as possible to $\delta$

$$\min \sum \| \delta_i - L(p'_i) \|^2 + \sum_{i \in c} \| p'_i - p_i \|^2$$
Laplacian Mesh Editing

Find a surface whose Laplacian coordinates are as close as possible to $\delta$

\[
\int_{\mathcal{S}} \left\| \Delta p' - \delta' \right\|^2 \, dS \rightarrow \min
\]

s.t. $p'_i = p_i, \; i \in \{\text{point constraints}\}$

Gateaux derivative $\Rightarrow \quad \Delta^2 p' = \Delta \delta'$
Physically-Based

\[ \int_{\Omega} k_s \left( \| d_u \|^2 + \| d_v \|^2 \right) + k_b \left( \| d_{uu} \|^2 + 2 \| d_{uv} \|^2 + \| d_{vv} \|^2 \right) dudv \]

stretcing

bending

Gateuax derivative =>

\[ -k_s \Delta d + k_b \Delta^2 d = 0 \]

(almost) the same equation?

Bi-Laplacian
Issue

Reconstructing from differential coordinates makes sense only if they are *rotation and translation invariant*. Otherwise, you get this

Translating a handle induces local rotations!
Laplacian Coordinates

- Translation invariant
- Not rotation/scale invariant

\[ \delta_i = L(v_i) = L(v_i + t); \forall t \in \mathbb{R}^3 \]
Solutions

1. Transform, ignoring rotations or details

2. **while** (not converged)
   - Estimate rotations *(from normals)*
   - Rotate differential coordinates and solve

\[
E(V') = \sum_{i=1}^{n} \| R_i \delta_i - L(p'_i) \|^2 + \sum_{i \in c} \| p'_i - p_i \|^2
\]

Solutions

1. Transform, ignoring rotations or details
2. while (not converged)
   – Estimate rotations (from normals)
   – Rotate differential coordinates and solve

Rotations + **scaling** – invariant?

Add local transformations $T_i$ as variables

$$E(V') = \sum_{i=1}^{n} \| T_i \delta_i - L(p_i') \|^2 + \sum_{i \in c} \| p_i' - p_i \|^2$$

Rotations + **scaling** – invariant?

Add local transformations $T_i$ as variables

$$E(V') = \sum_{i=1}^{n} \| T_i \delta_i - L(p'_i) \|^2 + \sum_{i \in c} \| p'_i - p_i \|^2$$

$$\min_{T_i} \left( \| T_i v_i - v'_i \|^2 + \sum_{j \in N_i} \| T_i v_j - v'_j \|^2 \right).$$

Rotations + **scaling** – invariant?

Add local transformations $T_i$ as variables

$$\min_{T_i} \left( \| T_i v_i - v'_i \|^2 + \sum_{j \in N_i} \| T_i v_j - v'_j \|^2 \right).$$

$T_i = \text{translation} + \text{rotation} + \text{scaling}$

Represent (a linearization of) $T_i$ using translation/rotation/scaling parameters

Rotations + **scaling** – invariant?

Add local transformations $T_i$ as variables

$$E(V') = \sum_{i=1}^{n} ||T_i \delta_i - L(p'_i)||^2 + \sum_{i \in c} ||p'_i - p_i||^2$$

$\Rightarrow$ $T_i$ is a linear function of $V'$

$\Rightarrow$ Quadratic optimization

$\Rightarrow$ Linear solve!

As-Rigid-As-Possible Surface Modelling
As-rigid-as-possible (ARAP)
As-rigid-as-possible (ARAP)

• “Intuitive” deformations
  ◦ Smooth deformations at large scale
• Preserve local features
• Fast, for interactive mesh editing
ARAP in a nutshell...

1. Break mesh into overlapping pieces
2. Try to move each piece rigidly
3. Combine all local transformations into a smooth one
Pieces

Vertex Umbrella

– Covers entire surface
– One cell per vertex
– All triangles exist in 3 cells
Rigid motion

If cell $i$ moved rigidly:

$$p_j' - p_i' = R_i (p_j - p_i)$$

$$\forall j \in N(i)$$
Deviation from rigid motion

If cell $i$ moved rigidly:

$$p_j' - p_i' = R_i(p_j - p_i)$$

$$\forall j \in N(i)$$

$$E = \sum_{j \in N(i)} \|p_j' - p_i' - R_i(p_j - p_i)\|^2$$
For the whole mesh

\[ E = \sum_{i} \sum_{j \in N(i)} \| p_j' - p_i' - R_i (p_j - p_i) \|^2 \]
For the whole mesh

\[ E = \sum_i \sum_{j \in N(i)} w_{ij} \| p'_j - p'_i - R_i (p_j - p_i) \|^2 \]
Orthogonal Procrustes problem

How to find the best rotation matrix aligning $V$ with $V'$?
Orthogonal Procrustes problem

How to find the best rotation matrix aligning $V$ with $V'$?

$$\arg\min_{R} \|RA - B\|_F$$

s.t. $R^T R = I$
1. Build covariance matrix $S = VV^T$
2. SVD: $S = U\Sigma W^T$
3. $R_i = UW^T$

Closed-form solution!
Mesh Deformation

\[
\min \sum_i \sum_{j \in N(i)} w_{ij} \| p'_j - p'_i - R_i (p_j - p_i) \|^2 \\
\text{s.t. } p'_i = \tilde{p}_i
\]

Caveats:
- \{p'_i\} and \{R_i\} are unknown
- Non-linear optimization problem
Mesh Deformation

1. Start with initial guess of \( \{p'_i\} \), find \( \{R_i\} \)
2. Given \( \{R_i\} \), minimize energy to find \( \{p'_i\} \)
3. Repeat

\[
\sum_{j \in N(i)} w_{ij} (p'_i - p'_j) = \sum_{j \in N(i)} \frac{w_{ij}}{2} (R_i + R_j)(p_i - p_j)
\]

\[
Lp' = b
\]
Advantages

Laplacian
  – Depends only on original mesh
  – Only needs to be factored once!

Rotations can be computed in parallel
  • Each iteration reduces energy
    ◦ Updating rotations guaranteed to reduce cell-error
    ◦ Updating positions guaranteed to reduce global error

Guaranteed Convergence
Results (vs Poisson)

Poisson:

ARAP:
Deformation models

Direct

\[ v' = (\sum w_j T_j) v \]

- Linear Blend Skinning
- Dual Quaternion Skinning
- ...

Variational

\[ v' = \arg\min_x E(x) \]

- Multiresolution editing
- As-Rigid-As-Possible
- Laplacian Mesh Editing
- ...


Deformation models

Direct

\[ \nu' = (\sum w_j T_j) \nu \]

- Linear Blend Skinning
- Dual Quaternion Skinning
- ...

Variational

\[ \nu' = \arg\min_x E(x) \]

- Multiresolution editing
- As-Rigid-As-Possible
- Laplacian Mesh Editing
- ...

\[ \nu' = (\sum w_j T_j) \nu \]
1) Rest pose
2) Skinning transformations

\[ \mathbf{T}_1 \quad \mathbf{T}_2 \]

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
3) Skinning weights

$w_{i,1}$
3) Skinning weights

$w_{i,2}$
Linear blend skinning (LBS)

\[ \mathbf{v}' = (\sum w_j T_j) \mathbf{v} \]
LBS is used widely in the industry

Halo 3

Bolt

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
LBS: candy-wrapper artifact
LBS: candy-wrapper artifact

Figure 2: Typical “candy-wrapper” artifacts of linear blend skin-ning.
What went wrong?

\[ v' = (\sum w_j T_j) v \]
What went wrong?

\[ \mathbf{v}' = (\sum w_j T_j) \mathbf{v} \]

\[
\mathbf{R}_1 = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \quad \mathbf{R}_2 = \begin{bmatrix}
-1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
What went wrong?

\[ \mathbf{v}' = (\sum w_j \mathbf{T}_j) \mathbf{v} \]

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
-1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Why can’t we just sum up rotation matrices?
Geometry of linear blending

$\text{SE}(3)$
Geometry of linear blending

$SE(3) \quad T_1 \quad + \quad T_2$
Geometry of linear blending

\[\text{SE}(3)\]

\[\mathbf{T}_1\quad \mathbf{T}_{\text{blend}}\quad \mathbf{T}_2\]
Geometry of linear blending

$SE(3)$

$T_1$ $T_{\text{blend}}$ $T_2$

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
Intrinsic blending

\[ T_{\text{blend}} \]

\[ T_1 \]

\[ T_2 \]
Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
Intrinsic blending using Lie algebras


\[
\arg\min_X \sum_X w_j d(X, T_j)
\]

\[
d(X, Y) = \|\log(\text{YX}^{-1})\|^2
\]

Karcher / Frechet mean

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Ladislav Kavan
Dual Quaternion Skinning

Dual Quats

Linear Blend
Where do the weights come from?
Manual?
Automatic skinning weight computation
Weights should obtain a few basic qualities
Inverse Euclidean distance weights are too crude, show obvious artifacts

\[ w_j(v) = \frac{1}{\|c_i - v\|^2} \]

weights optimized *inside* shape

[Shepard 1968], [Schaefer et al. 2006], etc.
Inverse Euclidean distance weights are too crude

\[ w_j(v) = \frac{1}{\| c_i - v \|^2} \]

weights optimized \textit{inside} shape

[Shepard 1968], [Schaefer et al. 2006], etc.
Discontinuous projection onto surface can be smoothed out

[Baran & Popović 2007]

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Alec Jacobson
Discontinuous projection onto surface can be smoothed out

\[ \arg \min_{w_j} \int_{\Omega} \| \nabla w_j \|^2 + h_j (w_j - \hat{w}_j)^2 \, dA \]

[Baran & Popović 2007]
Discontinuous projection onto surface can be smoothed out

\[
\argmin_{w_j} \int_{\Omega} \| \nabla w_j \|^2 + h_j (w_j - \hat{w}_j)^2 \, dA
\]

[Baran & Popović 2007]

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Alec Jacobson
Discontinuous projection onto surface can be smoothed out

\[ \arg\min_{w_j} \int_{\Omega} \| \nabla w_j \|^2 + h_j(w_j - \hat{w}_j)^2 \; dA \]

[Baran & Popović 2007]

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Alec Jacobson
Discontinuous projection onto surface can be smoothed out

\[
\argmin_{w_j} \int_\Omega \left( \| \nabla w_j \|^2 + h_j (w_j - \hat{w}_j)^2 \right) dA
\]

[Baran & Popović 2007]
Gradient energy weights not smooth at handles

\[
\arg\min_{w_j} \int_{\Omega} (\Delta w_j)^2 \, dA
\]

\[
\arg\min_{w_j} \int_{\Omega} ||\nabla w_j||^2 \, dA
\]

Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Alec Jacobson
Gradient energy weights not smooth at handles

\[
\arg\min_{w_j} \int_{\Omega} (\Delta w_j)^2 dA
\]

\[
\arg\min_{w_j} \int_{\Omega} \| \nabla w_j \|^2 dA
\]
Gradient energy weights not smooth at handles

\[ \Delta^2 w_j = 0 \]

\[ \Delta w_j = 0 \]
Point constraints for Laplace equation

https://www.facebook.com/521399544544480/photos/a.523048724379562/800968259920939/?type=1&theater, Facebook group “Circus tents and circus equipment”
Non-negative, local weights are mandatory

\[ 0 \leq w_j \leq 1 \]

\[ \text{argmin}_{w_j} \int_{\Omega} (\Delta w_j)^2 \, dA \]

[Bohstch & Kobbelt 2004]
Spurious extrema cause distracting artifacts

\[ 0 \leq w_j \leq 1 \]
Must explicitly prohibit spurious extrema

\( w_j \) is “monotonic”
Previous methods fail in one way or another

<table>
<thead>
<tr>
<th></th>
<th>$\Delta w_j$</th>
<th>$\Delta^2 w_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>smooth</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>non-negative</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>shape-aware</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>local</td>
<td>-/✓</td>
<td>–</td>
</tr>
<tr>
<td>monotonic</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>arbitrary handles</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shepard 1968, Sibson 1980, Schaefer et al. 2006</td>
</tr>
<tr>
<td>Baran &amp; Popovic 2007, Joshi et al. 2007</td>
</tr>
</tbody>
</table>
Constrained optimization ensures satisfaction of all properties

\[
\arg\min_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV
\]

+ shape-aware
+ smoothness

Constrained optimization ensures satisfaction of all properties

\[
\arg\min_{w_j,j=1,\ldots,m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV
\]

\[
w_j(v) = \begin{cases} 
1 & v \in h_j, \\
0 & v \in h_k \\
\text{linear on cage facets} & 
\end{cases}
\]

Constrained optimization ensures satisfaction of all properties

\[ \arg\min_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV \]

\[ 0 \leq w_j \leq 1, \]

\[ \sum_{j=1}^{m} w_j = 1 \]

[Jacobson et al. 2011]
Constrained optimization ensures satisfaction of all properties

$$\arg\min_{w_j, j=1,\ldots,m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV$$

$$0 \leq w_j \leq 1,$$

$$\sum_{j=1}^{m} w_j = 1$$

[Jacobson et al. 2011]
Constrained optimization ensures satisfaction of all properties

$$\text{argmin}_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV$$

$$\|w\|_1 = 1$$

+ shape-aware
+ smoothness
+ arbitrary handles
+ non-negativity
+ locality

[Rustamov 2011]
Constrained optimization ensures satisfaction of all properties

\[ \arg\min_{w_j, j=1,\ldots,m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV \]

\[ \|w\|_1 = 1 \rightarrow \sum_{j=1}^{m} |w_j| = 1 \]

[Rustamov 2011]
Constrained optimization ensures satisfaction of all properties

$$\arg\min_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \left( \Delta w_j \right)^2 dV$$

$$\|w\|_1 = 1 \rightarrow \sum_{j=1}^{m} |w_j| = 1 \rightarrow \sum_{j=1}^{m} w_j = 1,$$

$$0 \leq w_j \leq 1$$

[Rustamov 2011]
Constrained optimization ensures satisfaction of all properties

\[
\arg\min_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV
\]

\[
\nabla w_j \cdot \nabla u_j > 0
\]

+ shape-aware
+ smoothness
+ arbitrary handles
+ non-negativity
+ locality
+ monotonicity

Previous methods fail in one way or another

<table>
<thead>
<tr>
<th></th>
<th>Euclidean</th>
<th>$\Delta w_j = u$</th>
<th>$\Delta^2 w_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>smooth</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>non-negative</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>shape-aware</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>local</td>
<td>–/✓</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>monotonic</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>arbitrary handles</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>


Slides from Skinning: Real-time Shape Deformation Course, Direct Skinning Methods and Deformation Primitives by Alec Jacobson
Constrained optimization ensures satisfaction of all properties

$$\arg\min_{w_j, j=1, \ldots, m} \sum_{j=1}^{m} \int_{\Omega} (\Delta w_j)^2 \, dV$$

$$\nabla w_j \cdot \nabla u_j > 0$$

Weights retain nice properties in 3D