Detail Preservation
Local & Global
Complex Models
Shape Deformation

- Linear Def.
- Nonlinear Def.
- Space Def.
- Surface Def.

Model complexity vs. Deformation complexity matrix:
- Linear Def.: Simple models with simple deformations.
- Nonlinear Def.: Complex models with complex deformations.
- Space Def.: Models deformed in space.
- Surface Def.: Surface-level deformations.
Shape Deformation

Model complexity

Deformation complexity

Linear Def.
Nonlinear Def.
Space Def.
Surface Def.
Linear Surface-Based Deformation

- Shell-Based Deformation
- Multi-Scale Deformation
- Differential Coordinates
Mesh deformation by displacement function $d$
- Interpolate prescribed constraints
- Smooth, intuitive deformation
$\rightarrow$ Physically-based principles

$$d(p_i) = d_i$$

$d : S \rightarrow \mathbb{R}^3$
$p \leftrightarrow p + d(p)$
Shell Deformation

- **Stretching**
  - Change of local distances
  - Captured by 1st fundamental form

- **Bending**
  - Change of local curvature
  - Captured by 2nd fundamental form

- **Stretching & bending is sufficient**
  - 1st and 2nd fundamental forms determine a surface up to rigid motion.

\[
\int \Omega \, k_s \left\| I - \bar{I} \right\|^2
\]

\[
I = \begin{bmatrix}
  x_u^T x_u & x_u^T x_v \\
  x_v^T x_u & x_v^T x_v
\end{bmatrix}
\]

\[
\int \Omega \, k_b \left\| \Pi - \bar{\Pi} \right\|^2
\]

\[
\Pi = \begin{bmatrix}
  x_{uu}^T n & x_{uv}^T n \\
  x_{vu}^T n & x_{vv}^T n
\end{bmatrix}
\]
Shell Deformation

• Nonlinear stretching & bending energies

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

stretching  \hspace{2cm} bending

• Linearize terms \rightarrow Quadratic energy

\[
\int_{\Omega} k_s \left( \left\| \frac{\partial d}{\partial u} \right\|^2 + \left\| \frac{\partial d}{\partial v} \right\|^2 \right) + k_b \left( \left\| \frac{\partial^2 d}{\partial u^2} \right\|^2 + 2 \left\| \frac{\partial^2 d}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 d}{\partial v^2} \right\|^2 \right) \, dudv
\]

stretching  \hspace{2cm} bending
Shell Deformation

• Minimize linearized shell energy

\[ \int_{\Omega} k_s \left( \left\| \frac{\partial d}{\partial u} \right\|^2 + \left\| \frac{\partial d}{\partial v} \right\|^2 \right) + k_b \left( \left\| \frac{\partial^2 d}{\partial u^2} \right\|^2 + 2 \left\| \frac{\partial^2 d}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 d}{\partial v^2} \right\|^2 \right) \, dudv \]

\[ f(x) \rightarrow \min \]

• Variational calculus → Euler-Lagrange PDE

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

\[ f'(x) = 0 \]

⇒ “Best” deformation that satisfies constraints
Stretching & Bending

Initial state

\[ \Delta d = 0 \]
(Membrane)

\[ \Delta^2 d = 0 \]
(Thin plate)
PDE Discretization

• Euler-Lagrange PDE

\[ \Delta^2 d = 0 \]
\[ d = 0 \]
\[ d = \delta h \]

• Laplace discretization

\[ \Delta d_i = \frac{1}{2A_i} \sum_{j \in N_i} (\cot \alpha_{ij} + \cot \beta_{ij})(d_j - d_i) \]

\[ \Delta^2 d_i = \Delta(\Delta d_i) \]
Linear System

- Sparse linear system (19 nz/row)

\[
\begin{pmatrix}
\Delta^2 & 0 & 0 \\
0 & I & 0 \\
0 & 0 & I
\end{pmatrix}
\begin{pmatrix}
\vdots \\
d_i \\
\vdots
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
\delta h_i
\end{pmatrix}
\]

  - Turn into symmetric system

- Solve this system *each frame*
  - Only right hand side changes
  - Symmetric positive definite matrix
  - Use efficient linear solvers !!!
Sparse SPD Solvers

• Cholesky factorization
  – Cubic complexity
  – High memory consumption (doesn’t exploit sparsity)

• Iterative conjugate gradients
  – Quadratic complexity
  – Need sophisticated preconditioning

• Multigrid solvers
  – Linear complexity
  – But rather complicated to develop (and to use)

• Sparse Cholesky factorization!
Dense Cholesky Solver

Solve $Ax = b$

1. Cholesky factorization $A = LL^T$

2. Solve system $y = L^{-1}b$, $x = L^{-T}y$
Sparse Cholesky Factorization

\[ \mathbf{A} = \mathbf{L} \mathbf{L}^T \]

500×500 matrix
3500 non-zeros

Reordering
\[ \mathbf{P}^T \mathbf{A} \mathbf{P} \]

Cholesky Factorization
\[ \mathbf{L} \]
36k non-zeros

500×500 matrix
3500 non-zeros

14k non-zeros
Sparse Cholesky Solver

Solve $Ax = b$

**Pre-computation**

1. Matrix re-ordering  
   $\tilde{A} = P^TAP$

2. Cholesky factorization  
   $\tilde{A} = LL^T$

**Per-frame computation**

3. Solve system  
   $y = L^{-1}P^Tb$,  
   $x = PL^{-T}y$
Linear System Solver

Per frame computational costs

- Conjugate Gradients
- Multigrid
- Sparse Cholesky
Derivation Steps

1. Nonlinear Energy
2. Linearization
3. Quadratic Energy
4. Variational Calculus
5. Linear PDE
6. Discretization
7. Linear Equations
CAD-Like Deformation

[Botsch & Kobbelt, SIGGRAPH 04]
Face Animation

Mocap Markers

Large-Scale Deformation

Skin Rendering

[Bickel et al, SIGGRAPH 07]
• Kobbelt et al, *Interactive multi-resolution modeling on arbitrary meshes*, SIGGRAPH 1998

• Botsch & Kobbelt, *An intuitive framework for real-time freeform modeling*, SIGGRAPH 2004
Linear Surface-Based Deformation

- Shell-Based Deformation
- Multi-Scale Deformation
- Differential Coordinates
Multi-Scale Modeling

• Even pure translations induce local rotations!
  ➡ Inherently non-linear coupling

• Alternative approach
  – Linear deformation + multi-scale decomposition...

Original

Linear

Nonlinear
Multi-Scale Editing

Frequency decomposition

Change low frequencies

Add high frequency details, stored in local frames
Multi-Scale Editing

[Multi-Scale Modeling]

S

Decomposition

B

Freeform Modeling

B'

Detail Information

Reconstruction

[Kobbelt et al, SIGGRAPH 98]
Normal Displacements

[Kobbelt et al, SIGGRAPH 98]
Limitations

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

[Original, Normal Displ., Nonlinear]

[Botsch et al, EG 03, VMV 06]
Limitations

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

[Botsh et al, EG 03, VMV 06]
Literature

• Kobbelt et al, *Interactive multi-resolution modeling on arbitrary meshes*, SIGGRAPH 1998


• Botsch & Kobbelt, *Multiresolution surface representation based on displacement volumes*, Eurographics 2003

• Botsch et al, *Deformation transfer for detail-preserving surface editing*, VMV 2006
Linear Surface-Based Deformation

- Shell-Based Deformation
- Multi-Scale Deformation
- Differential Coordinates
Differential Coordinates

1. Manipulate differential coordinates
   – Gradients, Laplacians, local frames
   – Intuition: Close connection to surface normal

2. Find mesh with these differential coords
   – Cannot be solved exactly
   – Formulate as variational minimization
Differential Coordinates

Original → Rotated Diff-Coords → Reconstructed Mesh
Differential Coordinates

• Which differential coordinate $\delta_i$?
  – Gradients
  – Laplacians
  – ...

• How to get local transformations $T_i(\delta_i)$?
  – Smooth propagation
  – Implicit optimization
  – ...

Gradient-Based Editing

• Manipulate gradient of a function (e.g. a surface)

\[ g = \nabla f \quad g \mapsto T(g) \]

• Find function \( f' \) whose gradient is closest to \( g' \)

\[ f' = \arg\min_f \int_\Omega \| \nabla f - T(g) \|^2 \, du \, dv \]

• Variational calculus \( \rightarrow \) Euler-Lagrange PDE

\[ \Delta f' = \text{div} \, T(g) \]

[Yu et al, SIGGRAPH 04]
Gradient-Based Editing

• Consider piecewise linear coordinate function

\[ p(u, v) = \sum_{v_i} p_i \cdot \phi_i(u, v) \]

• Its gradient is

\[ \nabla p(u, v) = \sum_{v_i} p_i \cdot \nabla \phi_i(u, v) \]
Gradient-Based Editing

• Consider piecewise linear *coordinate function*

\[ p(u, v) = \sum_{v_i} p_i \cdot \phi_i(u, v) \]

• Its gradient is

\[ \nabla p(u, v) = \sum_{v_i} p_i \cdot \nabla \phi_i(u, v) \]

• It is constant per triangle

\[ \nabla p|_{f_j} =: g_j \in \mathbb{R}^{3 \times 3} \]
Gradient-Based Editing

- Gradient of coordinate function \( p \)

\[
\begin{pmatrix}
  g_1 \\
  \vdots \\
  g_F 
\end{pmatrix} =
\begin{pmatrix}
  p_1^T \\
  \vdots \\
  p_V^T 
\end{pmatrix}
\]

- Manipulate per-face gradients

\[
g_j \rightarrow T_j(g_j)
\]
Gradient-Based Editing

• Reconstruct mesh from new gradients
  – Overdetermined \((3F \times V)\) system
  – Weighted least squares system

\[ \Rightarrow \text{Linear Poisson system } \Delta p' = \text{div } T(g) \]

\[
\begin{align*}
\text{div} \nabla &= \Delta \\
G^T DG \cdot \begin{pmatrix} p'_1^T \\ \vdots \\ p'_V^T \end{pmatrix} &= G^T D \cdot \begin{pmatrix} T_1(g_1) \\ \vdots \\ T_F(g_F) \end{pmatrix}
\end{align*}
\]
Laplacian-Based Editing

• Manipulate Laplacians field of a surface

\[ l = \Delta(p) , \quad l \mapsto T(l) \]

• Find surface whose Laplacian is closest to \( \delta' \)

\[ p' = \arg\min_p \int_{\Omega} \| \Delta p - T(l) \|^2 \, dudv \]

• Variational calculus yields Euler-Lagrange PDE

\[ \Delta^2 p' = \Delta T(l) \]

[Sorkine et al, SGP 04]
Careful Discretization!

Irregular mesh

\[ L^T L p' = L^T \delta' \]

\[ L^2 p' = L \delta' \]

[Botsch & Sorkine, TVCG 08]
Differential Coordinates

• Which differential coordinate $\delta_i$?
  – Gradients
  – Laplacians
  – ...

• How to get local transformations $T_i(\delta_i)$?
  – Smooth propagation
  – Implicit optimization
  – ...

Smooth Propagation

1. Compute handle’s deformation gradient
2. Extract rotation and scale/shear components
3. Propagate damped rotations over ROI
Deformation Gradient

- Handle has been transformed \textit{affinely}
  \[ T(x) = Ax + t \]

- Deformation gradient is
  \[ \nabla T(x) = A \]

- Extract rotation $\mathbf{R}$ and scale/shear $\mathbf{S}$ by \textit{polar decomposition}
  \[ A = U \Sigma V^T \quad \Rightarrow \quad R = UV^T, \quad S = V \Sigma V^T \]
Smooth Propagation

• Construct smooth scalar field [0,1]
  • $s(x)=1$: Full deformation (handle)
  • $s(x)=0$: No deformation (fixed part)
  • $s(x)\in(0,1)$: Damp handle transformation in between
Damp Handle Transformation

- Full handle transformation
  - Rotation: \( R(c, a, \alpha) \)
  - Scaling: \( S(s) \)

- Damped by scalar \( \lambda \)
  - Rotation: \( R(c, a, \lambda \cdot \alpha) \)
  - Scaling: \( S(\lambda \cdot s + (1-\lambda) \cdot 1) \)
Differential Coordinates

Original → Rotated → Diff-Coords → Reconstructed Mesh
Limitations

• Differential coordinates work well for rotations
  – Represented by deformation gradient

• Translations don’t change deformation gradient
  – Translations don’t change differential coordinates
  – “Translation insensitivity”
Literature

• Yu et al, *Mesh editing with Poisson-based gradient field manipulation*, SIGGRAPH 2004

• Sorkine et al, *Laplacian surface editing*, SGP 2004
Surface-Based Deformation

- Problems with
  - Highly complex models
  - Topological inconsistencies
  - Geometric degeneracies
Freeform Deformation

• Deform object’s bounding box
  – Implicitly deforms embedded objects
Freeform Deformation (FFD)

• Trivariate tensor-product spline

\[ d(u, v, w) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} d_{ijk} N_i(u) N_j(v) N_k(w) \]

[Sederberg & Perry, SIGGRAPH 87]
Direct Manipulation FFD

• How to prescribe displacement constraints?
  – Solve linear system for control points
  – Can be over- or under-determined
  – Pseudo-inverse: least squares, least norm

[Hsu et al, SIGGRAPH 92]
Direct Manipulation

• Depends a lot on grid resolution
  – Minimize control point movement $\neq$ minimize physical energies!
Cage Deformation

- Deform object through *control cage*
  - Spline control points $\rightarrow$ cage vertices
  - Spline basis $\rightarrow$ generalized barycentric coordinates

[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]
Cage Deformation

- Deform object through \textit{control cage}
  - Spline control points $\rightarrow$ cage vertices
  - Spline basis $\rightarrow$ generalized barycentric coordinates

[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]
Cage Deformation

- Deform object through *control cage*
  - More flexible than spline control grids
  - Same limitation for direct manipulation

[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]
Mesh deformation by displacement function $d$
  - Interpolate prescribed constraints
  - Smooth, intuitive deformation

$\xrightarrow{\text{Physically-based principles}}$

$$d(p_i) = d_i$$

$p \leftrightarrow p + d(p)$

Space Deformation
Volumetric Energy Minimization

• Minimize similar energies to surface case

\[ \int_{\mathbb{R}^3} \left( \|d_{uu}\|^2 + \|d_{uv}\|^2 + \ldots + \|d_{ww}\|^2 \right) dV \rightarrow \min \]

• But displacement function lives in 3D...
  – Need a volumetric space tessellation?
  – No, same functionality provided by RBFs
Radial Basis Functions

- Represent deformation by RBFs

\[ d(x) = \sum_j w_j \cdot \varphi(\|c_j - x\|) + p(x) \]

- Choose basis function \( \varphi(r) = r^3 \)
  - Function \( d \) is triharmonic \( \Delta^3 d = 0 \)
  - Minimizes fairness energy

\[ \int_{\mathbb{R}^3} \|d_{uuu}\|^2 + \|d_{vuu}\|^2 + \ldots + \|d_{www}\|^2 \, du \, dv \, dw \rightarrow \min \]

[Botsch & Kobbelt, EG 05]
RBF Deformation

• Represent deformation by RBFs

\[ d(x) = \sum_j w_j \cdot \varphi(||c_j - x||) + p(x) \]

1. RBF fitting
   – Interpolate displacement constraints
   – Solve linear system for \( w_j \) and \( p \)

[Botsch & Kobbelt, EG 05]
RBF Deformation

• Represent deformation by RBFs

\[ d(x) = \sum_j w_j \cdot \varphi(\|c_j - x\|) + p(x) \]

2. RBF evaluation

– Function \( d \) transforms points
– Jacobian \( (\nabla d)^{-T} \) transforms normals
– Evaluate on the GPU!

[Botsch & Kobbelt, EG 05]
RBF Deformation

1M vertices

[Botsh & Kobbelt, EG 05]
“Bad Meshes”

- 3M triangles
- 10k components
- Not oriented
- Not manifold

[Botsch & Kobbelt, EG 05]
Local & Global Deformations

[Botsch & Kobbelt, EG 05]
• Sederberg & Parry, *Free-Form Deformation of Solid Geometric Models*, SIGGRAPH 1986

• Botsch & Kobbelt, *Real-time shape editing using radial basis functions*, Eurographics 2005

• Ju et al, *Mean value coordinates for closed triangular meshes*, SIGGRAPH 2005


• Lipman et al, *Green coordinates*, SIGGRAPH 2008
Shape Deformation

- Linear Def.
- Nonlinear Def.
- Space Def.
- Surface Def.

Model complexity vs. Deformation complexity
Linear vs. Nonlinear

Surface-Based

RBF

Nonlinear
Linear Approaches

Nonlinear Energy \rightarrow Linearization

Quadratic Energy \rightarrow Variational Calculus

Linear PDE \rightarrow Discretization

Linear Equations

causes artifacts for large deformations
Linearizations / Simplifications

• Shell-based deformation

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

\[ \int_{\Omega} k_s \left( \left\| \frac{\partial d}{\partial u} \right\|^2 + \left\| \frac{\partial d}{\partial v} \right\|^2 \right) + k_b \left( \left\| \frac{\partial^2 d}{\partial u^2} \right\|^2 + 2 \left\| \frac{\partial^2 d}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 d}{\partial v^2} \right\|^2 \right) \, dudv \]
Linearizations / Simplifications

- Gradient-based editing

\[ \nabla T(x) = A \]
Linear vs. Nonlinear

Original

Shell

Gradient

Nonlinear
Linear vs. Nonlinear

- Analyze existing methods
  - Some work for translations
  - Some work for rotations
  - No method works for both

[Botsch & Sorkine, TVCG 08]
Literature

• Botsch et al, PriMo: Coupled prisms for intuitive surface modeling, SGP 2006

• Botsch & Sorkine, On linear variational surface deformation methods, TVCG 2008
Shape Deformation

Model complexity

Deformation complexity

Linear Def.

Nonlinear Def.

Space Def.

Surface Def.
Nonlinear Deformation?

- Sounds easy: “Just don’t linearize.”

- Not so easy though...
  - Solve nonlinear problems (Newton, Gauss-Newton)
  - No convergence guarantees
  - Robustness issues
  - Considerably slower
Nonlinear Surface Deformation

- Shell-Based Deformation
- Rigid Cells
- As-rigid-as-possible deformation
Nonlinear Discrete Shells

\[
E(x_1, \ldots, x_m) = \lambda \sum_{e} w_{s,e} (l_e - L_e)^2 + \mu \sum_{e} w_{b,e} (\theta_e - \Theta_e)^2
\]

**Stretching:** change of edge length

**Bending:** change of dihedral angle
Gauss-Newton Minimization

- **Residual function**

\[
\begin{bmatrix}
x_1 \\
y_1 \\
z_1 \\
\vdots \\
x_n \\
y_n \\
z_n \\
\end{bmatrix} \mapsto \begin{bmatrix}
\sqrt{\lambda w_{s,1}} (l_1 - L_1) \\
\vdots \\
\sqrt{\lambda w_{s,m}} (l_m - L_m) \\
\sqrt{\mu w_{b,1}} (\theta_1 - \Theta_1) \\
\vdots \\
\sqrt{\mu w_{b,m}} (\theta_m - \Theta_m) \\
\end{bmatrix}
\]

\[E(x) = f(x)^T f(x) \rightarrow \min\]

- **Iterate until convergence**

\[
J(x)^T J(x) \delta = -J(x)^T f(x)
\]

\[x \leftarrow x + h \delta\]
Deformation Results

[Fröhlich & Botsch, CGF 11]

[Botsch & Sorkine, TVCG 08]
Deformation Results

Global stiffness control

[Fröhlich & Botsch, CGF 11]
Deformation Results

Local stiffness control

[Fröhlich & Botsch, CGF 11]
Nonlinear Face Animation

Add nonlinear wrinkle effects & realistic rendering

[Bickel et al, SCA 2008]
Nonlinear Surface Deformation

- Shell-Based Deformation
- Rigid Cells
- As-rigid-as-possible deformation
Cells

- Qualitatively emulate thin-shell behavior
- Thin volumetric layer around center surface
- Extrude polygonal cell $C_i$ per mesh face

[Botsch et al, SGP 06]
Rigid Cells

- Aim for robustness
  - Prevent cells from degenerating
  - Keep cells *rigid*

[Botsch et al, SGP 06]
Elastically Connected Rigid Cells

- Connect cells along their faces
  - Nonlinear elastic energy
  - Measures bending, stretching, twisting, ...
Cell-Based Surface Deformation

1. Prescribes position/orientation for cells
2. Find optimal rigid motions per cell
3. Update vertices by average cell transformations

[Botsch et al, SGP 06]
Elastically Connected Rigid Cells

• Pairwise energy

\[ E_{ij} = \int_{[0,1]^2} \| f^{i\rightarrow j}(u) - f^{j\rightarrow i}(u) \|^2 \, du \]

• Global energy

\[ E = \sum_{\{i,j\}} w_{ij} \cdot E_{ij} \quad w_{ij} = \frac{\| e_{ij} \|^2}{|F_i| + |F_j|} \]

[Botsch et al, SGP 06]
Nonlinear Minimization

• Find \textit{rigid} motion $T_i$ per cell $C_i$

$$\min_{\{T_i\}} \sum_{i,j} w_{ij} \int_{[0,1]^2} \|T_i(f^{i\rightarrow j}(u)) - T_j(f^{j\rightarrow i}(u))\|^2 \, du$$

• Generalized global \textit{shape matching} problem
  – Robust geometric optimization
  – Nonlinear Newton-type minimization
  – Hierarchical multi-grid solver

[Botsch et al, SGP 06]
Robustness

[Botsch et al, SGP 06]
PriMo

[Botsch et al, SGP 06]
Character Posing

[Botsch et al, SGP 06]
Nonlinear Surface Deformation

- Shell-Based Deformation
- Rigid Cells
- As-rigid-as-possible deformation
Surface Deformation

- Smooth large scale deformation
- Local as-rigid-as-possible behavior
  - Preserves small-scale details

[Sorkine & Alexa, SGP 07]
Deformation Energy

• Vertex neighborhoods should deform rigidly

$$\sum_{j \in N(i)} \| (p_j' - p_i') - R_i (p_j - p_i) \|^2 \to \text{min}$$

[Sorkine & Alexa, SGP 07]
Cell Deformation Energy

- If $p, p'$ are known then $R_i$ is uniquely defined

$\Rightarrow$ Shape matching problem

[Sorkine & Alexa, SGP 07]
Total Deformation Energy

• Sum over all vertex

\[
\min_{p'} \sum_{i=1}^{n} \sum_{j \in N(i)} \| (p'_j - p'_i) - R_i (p_j - p_i) \|^2
\]

• Treat \( p' \) and \( R_i \) as separate variables

• Allows for alternating optimization
  – Fix \( p' \), find \( R_i \) : Local shape matching per one-ring
  – Fix \( R_i \), find \( p' \) : Solve Laplacian system

[Sorkine & Alexa, SGP 07]
As-Rigid-As-Possible Modeling

- Start from naïve Laplacian editing as initial guess

![Initial guess](image1)
![1 iteration](image2)
![2 iterations](image3)

![Initial guess](image4)
![1 iterations](image5)
![4 iterations](image6)

[Sorkine & Alexa, SGP 07]
As-Rigid-As-Possible Modeling

[Sorkine & Alexa, SGP 07]
• Botsch et al, *PriMo: Coupled prisms for intuitive surface modeling*, SGP 2006

• Sorkine & Alexa, *As-rigid-as-possible surface editing*, SGP 2007

• Grinspun et al, *Discrete shells*, SCA 2003

• Fröhlich & Botsch, *Example-driven deformations based on discrete shells*, CGF 2011
Shape Deformation

- Linear Def.
- Nonlinear Def.
- Space Def.
- Surface Def.

Model complexity vs. Deformation complexity
Space PriMo

Volumetric Discretization → Cell-Based Deformation → Space Deformation

[Botsch et al, EG 07]
Space PriMo

[Botsch et al, EG 07]
Space PriMo

[Botsch et al, EG 07]
Space PriMo

14k components

Deformed triangle soup

[Botsch et al, EG 07]
Space PriMo

[Botsch et al, EG 07]
Space PriMo

[Botsch et al, EG 07]
Embedded Deformation

- Parameterize model with *deformation graph*
- Find optimal transformation for each node
  - Affine transformation per node
  - Weakly enforce rigidity on matrices

[Sumner et al, SIGGRAPH 07]
Embedded Deformation

Live edit

[Sumner et al, SIGGRAPH 07]
Literature

- Botsch et al, *Adaptive space deformations based on rigid cells*, Eurographics 2007
- Sumner et al, *Embedded deformations for shape manipulation*, SIGGRAPH 2007
Shape Deformation

- Linear Def.
- Nonlinear Def.
- Space Def.
- Surface Def.

Model complexity vs. Deformation complexity
Summary

Bending Energy
- Precise control of continuity
- Requires multi-resolution hierarchy
- Problems with large rotations

Differential Coords
- Designed for large rotations
- Problems with translations
- How to determine local rotations?
Summary

**Surface-Based**
- More precise control of surface properties
- Depends on surface complexity & quality

**Space Deformation**
- Doesn’t know about embedded surface
- Works for complex and “bad” input

vs.
Summary

Linear
  + Highly efficient & numerically robust
  – Many constraints for large-scale edits

Nonlinear
  – Numerically much more complex
  + Easier edits, fewer constraints

vs.
Literature

- Polygon Mesh Processing, Chapter 9