Surface Deformations Driven by Vector-Valued 1-Forms

Gabriel Taubin and Çağatay Demiralp
Handle-based interactive deformations

- Unified Poisson-based image and geometry editing
- Additional degrees of freedom under user control
- ASAP “As Simple As Possible” formulation

Recent Survey

- Botsch, M. & Sorkine, O.
  On Linear Variational Surface Deformation Methods
  *IEEE Transactions on Visualization and Computer Graphics, 2008, 14, 213-230*
A Lot of Prior Work (partial list)

- Interactive Multi-Resolution Modeling on Arbitrary Meshes (Kobbelt:sg1998)
- Multiresolution Signal Processing for Meshes (Guskov:sg1999)
- Local Control For Mesh Morphing (Alexa:smi2001)
- Linear Combination of Transformations (Alexa:sg2002)
- Surface Parameterization: a Tutorial and Survey (Floater:amgm2003)
- Poisson Image Editing (Perez:sg2003)
- Differential Coordinates for Local Mesh Morphing and Deformation (Alexa:vc2003)
- A Remeshing Approach to Multiresolution Modeling (Botsch:sgp2004)

- Differential Coordinates for Interactive Mesh Editing (Lipman:smi2004)
- Laplacian Surface Editing (Sorkine:sgp2004)
- Mesh Editing with Poisson-Based Gradient Field Manipulation (Yu:sg2004)
- Mesh Editing with Curvature Flow Laplacian (Au:sgp2005)
- As-Rigid-As-Possible Shape Manipulation (Igarashi:sg2005)
- A Global Laplacian Smoothing Approach with Feature Preservation (Ji:gagdgcg2005)
- Mean Value Coordinates for Closed Triangular Meshes (Ju:sg2005)
- Linear Rotation-Invariant Coordinates for Meshes (Lipman:sg2005)
- Laplacian Mesh Processing (Sorkine:eg2005star)
A Lot of Prior Work (partial list)

- Geometry-Aware Bases for Shape Approximation \{Sorkine:tvcg2005\}
- Poisson Shape Interpolation \{Xu:spm2005\}
- Harmonic Guidance for Surface Deformation \{Zayer:eg05\}
- Dual Laplacian Editing for Meshes \{Au:tvcg2006\}
- PriMo: Coupled prisms for intuitive surface modeling \{Botsch:sgp2006\}
- Deformation Transfer for Detail-Preserving Surface Editing \{Botsch:vmv2006\}
- Discrete one-forms on meshes and applications to 3D mesh parameterization \{Gortler:cagd2006\}
- Subspace Gradient Domain Mesh Deformation \{Huang:sg2006\}
- Material-Aware Mesh Deformations \{Popa:sma2006\}
- A Fast Multigrid Algorithm for Mesh Deformation \{Shi:sg2006\}
- Design of Tangent Vector Fields \{Fisher:sg2007\}
- Geometric Modeling in Shape Space \{Kilian:sg2007\}
- Dual Laplacian Morphing for Triangular Meshes \{Hu:cavw2007\}
- Stretch-based Tetrahedral Mesh Manipulation \{Song:gi2007\}
- As-Rigid-As-Possible Surface Modeling \{Sorkine:sgp2007\}
- Free-Form Motion Processing \{Kircher:tog2008\}

Goals

- Create intuitive tools for interactive shape design: If results are not satisfactory modify the parameters and try again
- Produce realistic-looking and aesthetically pleasing shape deformations
- Preserve local features
- Definitely not necessarily physically correct
Main motivation: Poisson Image Editing

- gradient-based approach to image editing

\[ I(u_1, u_2) \]

Modified gradient \( g(u_1, u_2) \)

Minimize \( \int_{[0,1] \times [0,1]} \| \nabla I(u) - g(u) \|^2 \, du \)

Poisson equation \( \Delta I = \nabla \cdot g \)

Extension to Surface Patches

\[ x_1(u) = \frac{\partial x}{\partial u_1} \]

\[ x_2(u) = \frac{\partial x}{\partial u_2} \]

\[ E(x) = \int_U \| dx(u) - v(u) \|^2 \, du \]

\[ \nabla I \mapsto dx = (x_1, x_2) \]

\( x \) - 3D 0-Form

\( dx \) - 3D 1-Form: differential of \( x \)

\[ v = (v_1, v_2) \] - 3D 1-Form

(point,direction) \( \mapsto \) vector

\[ (u, w) \mapsto \frac{\partial x}{\partial w} = w_1 \frac{\partial x}{\partial u_1} + w_1 \frac{\partial x}{\partial u_1} \]

\[ (u, w) \mapsto w_1 v_1(u) + w_2 v_2(u) \]
Discrete Vector-Valued Forms on Graphs

\[ G = (V, E) \]

**0-FORM**

\[ x : V \rightarrow \mathbb{R}^D \]

**1-FORM**

\[ v_j = \sum_{i} v_{ij} \]

\[ v : E' \rightarrow \mathbb{R}^D \]

Differential of a 0-form

\[ x : V \rightarrow \mathbb{R}^D \]

\[ dx : E' \rightarrow \mathbb{R}^D \]

\[ dx_{ij} = x_j - x_i \]
Exact (Integrable) 1-forms

If the graph has no loops all the 1-forms are exact!

There exists a 0-form $x$ so that $v = dx$

The deformation algorithm

1. Given a 0-form $x$
2. Rotate and stretch the differential $v_{ij} = \sigma_{ij} R_{ij} dx_{ij}$
3. Impose constraints (optional)
4. Integrate the deformed 1-form $v \mapsto x'$
Integrating a 1-form in the LS sense

Minimize \( E(x) = \| dx - v \|_\mu^2 \)

\[
E(x) = \sum_{(ij)} \mu_{ij} \| x_j - x_i - v_{ij} \|_2^2
\]

Edge weights \( \mu_{ij} = \mu_{ji} > 0 \) (not a 1-form)
Laplacian System of Linear Equations

Minimizing
\[ E(x) = \sum_{j \in i^*} \mu_{ij} \| x_i - x_j - v_{ij} \|^2 \]

Reduces to solving the linear system
\[ 0 = \frac{\partial E}{\partial x_i} = \sum_{j \in i^*} \mu_{ij} (x_i - x_j - v_{ij}) \quad i \in V \]

Discrete Poisson Problem
\[ \Delta x_i = \sum_{j \in i^*} \mu_{ij} (x_j - x_i) = \sum_{j \in i^*} \mu_{ij} v_{ji} \]

Solving Sparse Linear Systems

- **Direct Sparse Linear Solvers**
  - Sparse Cholesky decomposition (e.g. TAUCS)

- **Iterative Solvers**
  - Jacobi, Gauss-Seidel, Conjugate Gradients, LSQR
  - Easy to implement on existing data structures
  - Slow to converge
  - Where to start?
  - Problem-specific heuristics to initialize
  - In our case: integration of 1-forms along trees
Integrating a 1-form along a path

path \( \gamma = (i_0, i_1, i_2, \ldots, i_{N-1}, i_N) \)

\[
\int_{\gamma} v = v_{i_0i_1} + v_{i_1i_2} + \cdots + v_{i_{N-1}i_N}
\]

Integrating a 1-form along a Tree

\( i = \text{parent}(j) \Rightarrow x_j = x_i + v_{ij} \)
Integration along a spanning tree

Heuristics to Initialize for Iterative Solver

1. Start with undeformed 0-form \( x = x^0 \)
2. Compute edge errors \( \epsilon_{ij} = \mu_{ij} \| dx_{ij} - v_{ij} \|^2 \)
3. Build maximal spanning tree \( T \)
4. Integrate 1-form along spanning tree \( \mapsto x^1 \)
5. Minimize along line \( E(t) = E(x^0 + t \cdot x^1) \mapsto \hat{t} \)
6. New estimate \( x = x^0 + \hat{t} x^1 \)
7. Repeat N (2-6) times
spanning tree integration heuristic

Creating smooth rotation fields

- Exponential parameterization of rotations.
  - Single 3D vector

\[
R = e^{\hat{\omega}} \\
\hat{\omega} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} \quad \alpha = \|\omega\| \\
\hat{\omega} q = w \times q
\]

- Evaluation: Rodrigues Formula
- Supports very large rotations in one step
Creating smooth rotation and stretch fields

- Rotation field: one 3D vector per edge
  \[ w_{ij} = w_{ji} \]

- Stretch field: one positive scalar per edge
  \[ \sigma_{ij} = \sigma_{ji} \]

- Constrain values at handles
- Aggressive constrained smoothing

Relation to Laplacian Smoothing

- Jacobi iteration for our energy function:
  \[
  E(x) = \sum_{j \in i^*} \mu_{ij} \| x_i - x_j - v_{ij} \|^2 \\
  \delta x_i = \mu_i^{-1} \sum_{j \in i^*} \mu_{ij} \left( x_j - x_i - v_{ij} \right) \\
  x' = x + \lambda \delta x
  \]

- Laplacian smoothing: \( v = 0 \)
- Same algorithm used to smooth fields and deform 0-form
Results

Results
Results