Mesh Smoothing Algorithms

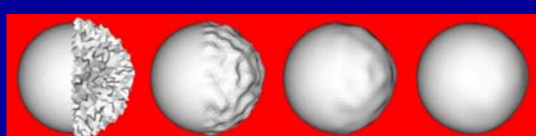
ENGN2911I
3D Photography and Geometry Processing
Brown Spring 2008
Gabriel Taubin

Overview

- Laplacian Smoothing
- Problems and fixes
- Vertex and Normal Constraints
- Normal Constraints at Boundary Vertices
- Isotropic vs. Anisotropic
- Linear vs. Nonlinear
- Filtering of Normal Fields
- Filters that Integrate Normal Fields
- Related Problems

Different Approaches

- Digital Signal Processing
- Physics-based / PDE Surfaces
- Variational / Regularization
- Multi-resolution
- Subdivision Surfaces



Classical Digital Signal Processing

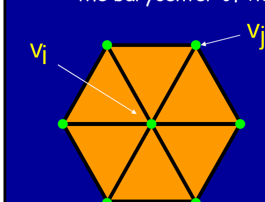
- Signals defined on regular grids
  - 1D : music / speech
  - 2D : images / video
  - 3D : medical imaging
- Solid theoretical foundation and practical algorithms
  - Sampling Theorem
  - DFT/FFT Fourier Analysis
  - FIR/IIR Linear Filters / Convolution
  - Non-linear filtering
  - Multi-rate filtering / up-sampling / down-sampling
  - Etc.

Graph and Mesh Signals

- Graph signal
  Signal defined on a graph (irregular grids)
  - $x = (x_1, \ldots, x_V)^T$
  - $G = (V,E)$
    - V = {i,j,...} --- vertices
    - E = {e={i,j},...} --- non-oriented edges
    - E = {e=(i,j),...} --- oriented edges
- Mesh Signal
  Signal defined on the graph of a polygonal mesh

Laplacian Smoothing in Mesh Generation

- Used to improve quality of 2D meshes for FE computations
- Keep boundary vertices fixed
- Move each internal vertex to the barycenter of its neighbors



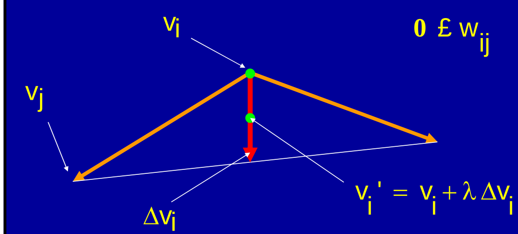
$$v_i' = \frac{1}{n_i} \sum_{j \in i^*} v_j$$

$$v_i' = v_i + \lambda \Delta v_i$$

The Laplacian Operator

$$\Delta v_i = \sum_j w_{ij}(v_j - v_i) \qquad 1 = \sum_j w_{ij}$$

0 £ w_{ij}

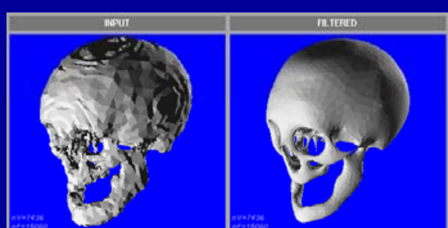


Laplacian Smoothing : Advantages

- Algorithm Simplicity
- Linear time and storage
- Edge length equalization
  (advantage depending on the application)
- Constraints and special effects
  by weight control

$$\Delta v_i = \sum_j w_{ij}(v_j - v_i) \qquad v_i' = v_i + \lambda \Delta v_i$$

Laplacian Smoothing Demo



Laplacian Smoothing : Disadvantages

- Overall Shrinkage
  - Solved by Taubin's Low-Pass filter algorithm
  - Why ? Fourier Analysis
- Edge length equalization
  (disadvantage depending on the application)
  - Solved by non-linear filtering
    - Fujiwara / Desbrun-et-al weights (curvature flow)
- Shrinkage at boundaries
  - Solved by hierarchical filtering ?
- Smoothing of ridges
  - Solved by Anisotropic diffusion

Laplacian Smoothing : Challenges

- How to solve all the problems preserving
  - Algorithm Simplicity
  - Linear time and storage
- Proposed Solution:
  - Modify the Laplacian Operator
    - Isotropic / Anisotropic
    - Linear / non-linear (avoid!)
  - Define Laplacian Operator on Normal fields
  - Use FIR linear filters
  - Dynamic connectivity resampling

$$\Delta v_i = \sum_j w_{ij}(v_j - v_i)$$

Fourier Analysis

$$\Delta x_i = \sum_j w_{ij}(x_j - x_i) \qquad K x = -\Delta x$$

- Eigenvalues of $K = I - W$ (FREQUENCIES)

$$0 = k_0 \leq k_1 \leq \cdots \leq k_N \leq 2$$

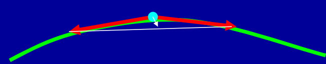
- Right eigenvectors of K (NATURAL VIBRATION MODES)

$$e_0, e_1, \ldots, e_N$$

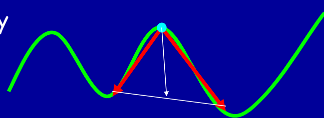
Geometry of low and high frequencies

$$k_h e_{hi} = K e_{hi}' = -\sum_j w_{ij}(e_{hj} - e_{hi})$$

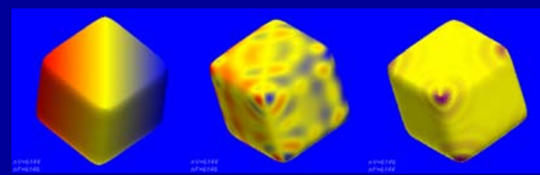
- Low frequency



- High frequency



Natural vibration modes



The Discrete Fourier Transform

- Eigenvectors form a basis of N-space
- Every signal can be written as a linear combination

$$x = \hat{x}_0 e_0 + \hat{x}_1 e_1 + \cdots + \hat{x}_N e_N$$

- Discrete Fourier Transform (DFT)

$$\hat{x} = (\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_N)^t$$

FIR Linear Filters

- Polynomial Transfer Function

$$x' = f(K)x \qquad Kx = -\Delta x$$

- $f(k)$ is a univariate polynomial
- $f(K)$ is a matrix
- Eigenvectors of K and $f(K)$ are the same
- Eigenvalues of $f(K)$ are

$$f(k_0)\,,\, f(k_1)\,,\ldots,\, f(k_N)$$

FIR Linear Filters

- After filtering

$$f(K)x = f(k_0)\hat{x}_0 e_0 + \cdots + f(k_N)\hat{x}_N e_N$$

- Evaluation of $f(K)\,x$ based on matrix multiplication
- It does not require the computation of eigenvalues and eigenvectors (DFT)
- Low-Pass : need univariate polynomial $f(k)$ such that

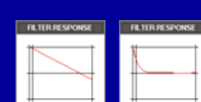
$$f(k_h) \approx 1 \qquad k_L \le k_{PB}$$
$$f(k_h) \approx 0 \qquad k_L > k_{PB}$$

Laplacian Smoothing is not Low-Pass

- After filtering

$$f(K)x = f(k_0)\hat{x}_0 e_0 + \cdots + f(k_N)\hat{x}_N e_N$$

- For Laplacian smoothing



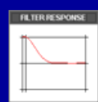
$$f(k_0) = 1$$

$$f(k_j) = (1 - \lambda\, k_j)^N \to 0 \qquad j \ne 0 \qquad 0 \le \lambda < 1$$

Taubin Smoothing (Siggraph'95)

- Minor modification of Laplacian smoothing algorithm
- Two Laplacian smoothing steps
- First shrinking step with positive factor
- Second unshrinking step with negative factor
- Use inverted parabola as transfer function

$$f(k) = ((1-\mu k)(1-\lambda k))^{N/2} \quad \text{with} \quad -\mu > \lambda > 0$$



Taubin-Zhang-Golub (ECCV'96) FIR Filter Design

- Efficient algorithm
  to evaluate any polynomial transfer function
- Based on Chebyshev polynomials defined by three term recursion
- All classical Finite Impulse Response (FIR) filter design techniques can be used with no modifications
- Implemented method of "windows" based on truncated Fourier series expansion of ideal transfer function and coefficient weighting to remove Gibbs phenomenon

Parameters

$$\Delta v_i = \sum_j w_{ij}(v_j - v_i)$$

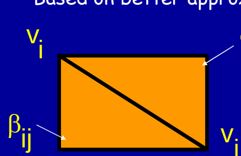
- Weights
  - Neighborhoods = non-zero weights
  - Prevention of Tangential drift
  - Edge-length equalization
- Boundaries and creases / hierarchical smoothing
- Vertex-dependent smoothing parameters

Linear / Non-Linear

- Linear Laplacian Operator
  - Weights are computed once and kept constant for all iterations
- Non-Linear Laplacian Operator
  - Weights are recomputed at every iteration

Preventing tangential drift

- Fujiwara (P-AMS'95)
  - Weights inversely proportional to edge length
- Desbrun-Meyer-Schroder-Barr (SG'99)
  - Based on better approximation of curvature normal



$$c_{ij} = \cot(\alpha_{ij}) + \cot(\beta_{ij})'$$

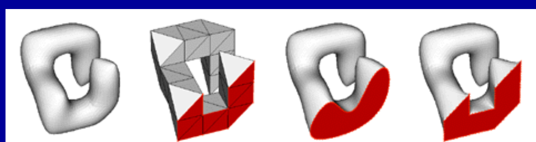
- Guskov-et-al (SG'99) based on divided differences and second order neighborhood

Mesh Signal Processing

- **A Signal Processing Approach to Fair Surface Design**, by G. Taubin, in Proceedings of Siggraph 1995
- **Optimal Surface Smoothing as Filter Design**, by G. Taubin, T. Zhang, and G. Golub, Fourth European Conference on Computer Vision (ECCV'96)
- **Interactive Multi-Resolution Modeling on Arbitrary Meshes**, by L. P. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel, in Proceedings of Siggraph 1998
- **Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow**, by M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, in Proceedings of Siggraph 1999
- **A Discrete Spring Model for Generating Fair Curves and Surfaces**, by A. Yamada, K. Shimada, T. Furuhata, and K.-H. Hou, in Proceedings of Pacific Graphics 1999, October 1999
- **Geometric Signal Processing on Polygonal Meshes**, by G. Taubin, Eurographics 2000 State of The Art Report (STAR), September 2000

Hierarchical Neighborhoods

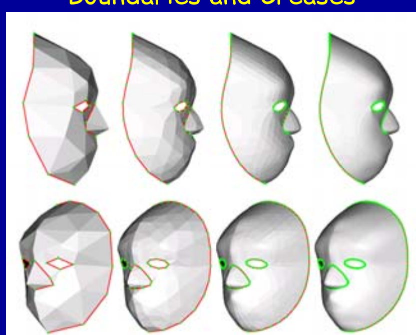
- Assign a numeric label to each vertex
- Vertex j is a neighbor of vertex i only if i and j are connected by an edge, and the label of i is less or equal than the label of j



Boundaries and Creases

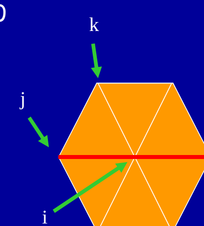
- Use hierarchical neighborhoods
- Assign label 1 to boundary and crease vertices
- Assign label 0 to all internal vertices
- The graph defined by the boundary and crease edges and vertices is smoothed independently of the rest of the mesh
- The rest of the mesh "follows" the graph defined by the boundary and crease edges and vertices
- Eigenvalues of K are complex, but $1-k_i| \leq 1$

Boundaries and Creases

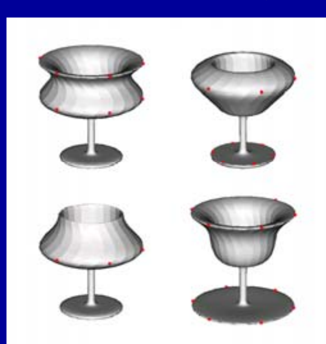


Hierarchical neighborhoods and weights

- W_{ij} > 0
- W_{ji} > 0
- W_{ik} = 0 but W_{ki} > 0



Vertex Constraints and Surface Design



Vertex Position Constraints

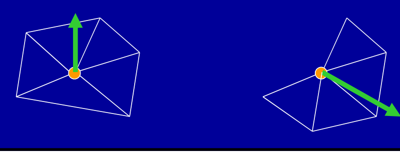
- Hard vs. soft constraints
- Hard vertex position constraints are easy to impose but produce artifacts because of lack of normal control
- Kobbelt-et-al Variational Fairing (SG'98)
  - Minimize square norm of Laplacian operator
- Yamada-et-al Discrete Spring Model (PCCGA'98)
  - impose soft normal constraints with a spring model that adds an extra term to the smoothing step
- Slow convergence and/or high computational cost

Variational Fairing

- Minimize $\sum_j \left\|\Delta v_i\right\|^2$
- Under linear constraints

The Boundary Shrinkage Problem

- Laplacian operator approximates
  - Mean curvature X normal vector X mean edge length
- Not for boundary vertices !
  - Has a strong tangencial component
- Fix : project onto normal direction



Modified Laplacian for Boundary Vertices

- Project onto normal direction

$$\Delta v_i = \sum_j w_{ij} n_i n_i^t (v_j - v_i)$$

- Define weights as 3x3 matrices

$$W_{ij} = w_{ij} n_i n_i^t$$

- Linear Anisotropic Laplacian Operator

Anisotropic Laplacian Operators

$$\Delta v_i = \sum_j W_{ij} (v_j - v_i)$$

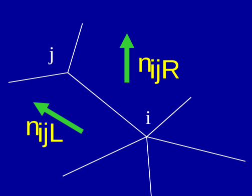
$$W_{ij} = C_i^{\dagger} C_{ij}$$

C_{ij} Symmetric non-negative definite

$$C_i = \sum_j C_{ij}$$

Preventing Tangential Drift

- Use Laplacian Operator that fixes boundary shrinkage
- But, how to define the vertex normals ?
- Use smooth face normal field instead



$$C_{ij} = n_{ijL} n_{ijL}^t + n_{ijR} n_{ijR}^t$$

Smoothing Normal Fields

- Signal is defined on dual graph with values in the unit sphere
- Only need to define Laplacian Operator
- Then can apply any Linear Filter
- Displacement $n_j - n_i$ is the Rotation defined by the vector product $n_i \times n_j$
- Laplacian Operator Δn_i is the average Rotation

Rodrigues Formula

- Local parameterization of Rotations

$$\{u : |u| \leq 1\} \rightarrow SO(3)$$

$$R(u) = cI + (1-c)rr^t + sr^{\Lambda}$$

- If n_1 and n_2 are two unit vectors, then

$$R(n_1 \times n_2)n_1 = n_2$$
$$R(n_1 \times n_2)n_1 \times n_2 = n_1 \times n_2$$

Laplacian for Normal Fields

- Definition

$$\lambda \Delta n_i = R\left(n_i \times \left(\lambda \sum_j w_{ij} n_j \right) \right)$$

$$n_i' = R\left(n_i \times \left(\lambda \sum_j w_{ij} n_j \right) \right) n_i$$

Constrained Normal Filtering

- Like vertex position constraints in the Euclidean case
- Just do not update the constrained values
- Face normals are filtered independently of vertex positions
- Then vertex positions are filtered with the linear anisotropic filter defined by the face normals
- Can impose both face normal constraints and vertex position constraints

Application : Hole filling

- Triangulate hole with internal vertices
- Smooth normal field in the graph defined by the hole faces and the incident faces
- Fix normals on incident faces
- Filter normals with boundary constraints
- Filter vertices with boundary constraints
- Use dynamic connectivity rules to resample if needed, and iterate

What Next ?

- Combine with Dynamic Connectivity Rules for adaptive resampling
- Ridge detection and enhancement
- Non-linear isotropic and anisotropic filtering

Irregular Mesh Resampling

- **Multiresolution Shape Deformations for Meshes with Dynamic Vertex Connectivity**, by L.P. Kobbelt, T. Bareuther, and H.-P. Seidel, in Proceedings of Eurographics 2000.

  - **Define Min-Max target edge lengths**
  - **Collapse short edges**
  - **Optimize vertex valences by flipping edges**
  - **Smooth mesh**
  - **Split long edges**

Non-Linear Anisotropic Diffusion

- **Scale-Space and Edge Detection Using Anisotropic Diffusion**, by P. Perona, and J. Malik, in IEEE Trans. on Pattern Analysis and Machine Intelligence, July 1990.
- **Anisotropic Feature-Preserving Denoising of Height Fields and Bivariate Data**, by M. Desbrun, M. Meyer, P. Schroder, and A. Barr, in Proceedings of Graphics Interface 2000, May 2000
- **Polyhedral Surface Smoothing with Simultaneous Mesh Regularization**, by Y. Ohtake, A.G. Belyaev, and I.A. Bogaevski, in Proceedings of the Geometric Modeling and Processing 2000, April 2000
- **Anisotropic Geometric Diffusion in Surface Processing**, by U. Clarenz, U. Diewald, and M. Rumpf, in Proceedings of IEEE Visualization 2000, October 2000
- **Mesh Regularization and Adaptive Smoothing**, by Y. Ohtake, A.G. Belyaev, and I.A. Bogaevski, Computer Aided Design, 2001