Geometric Signal Processing on Polygonal Meshes

Gabriel Taubin

IBM T.J.Watson Research Center

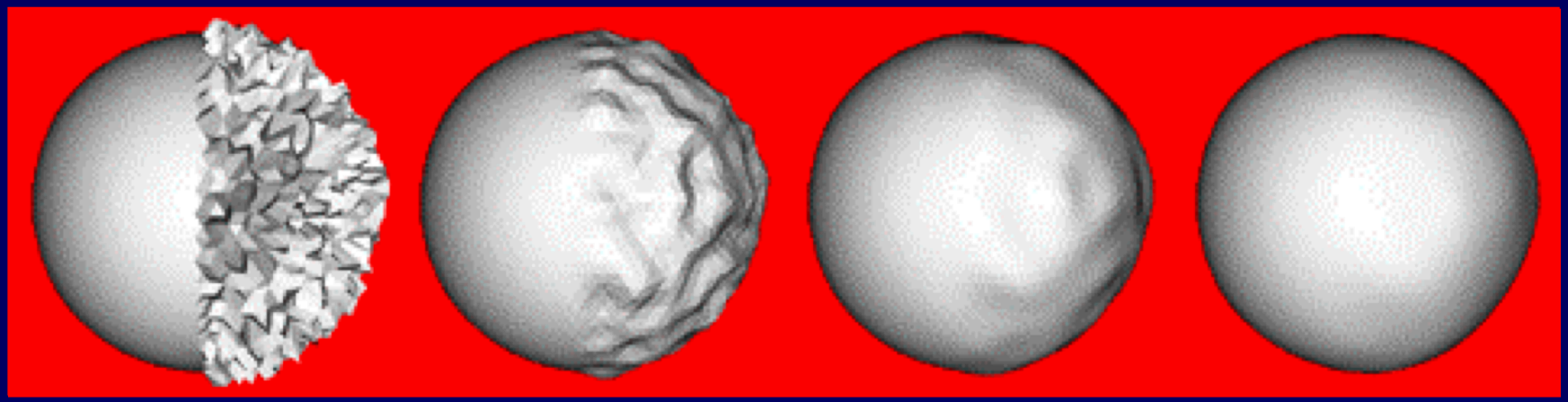
http://www.research.ibm.com/people/t/taubin

Large dense polygonal meshes

- Are becoming standard representation for surface data
  - 3D Scanning (Reverse engineering, Art)
  - Isosurfaces (Scientific Visualization, Medical)
  - Subdivision Surfaces (Modeling, Animation)
- But have too many degrees of freedom (vertices)
- How to ?
  - Smooth / De-noise
  - Edit / Deform / Constrain / Animate
  - Represent / Compress / Transmit
- BUT FAST !

Different approaches

- Signal Processing
- Physics-based / PDE Surfaces
- Variational / Regularization
- Multiresolution
- Subdivision Surfaces



About this talk

- Initial goal was to present a comprehensive survey
- Final result is not quite comprehensive
- Only way to verify claims is to implement yourself
- Which I did for most algorithms covered in the talk
- But run out of time to implement all
- Demo software (Java) available in my web pages (to be updated soon)
- The talk is biased
- There is much more to understand and do in this area

The Signal Processing Approach

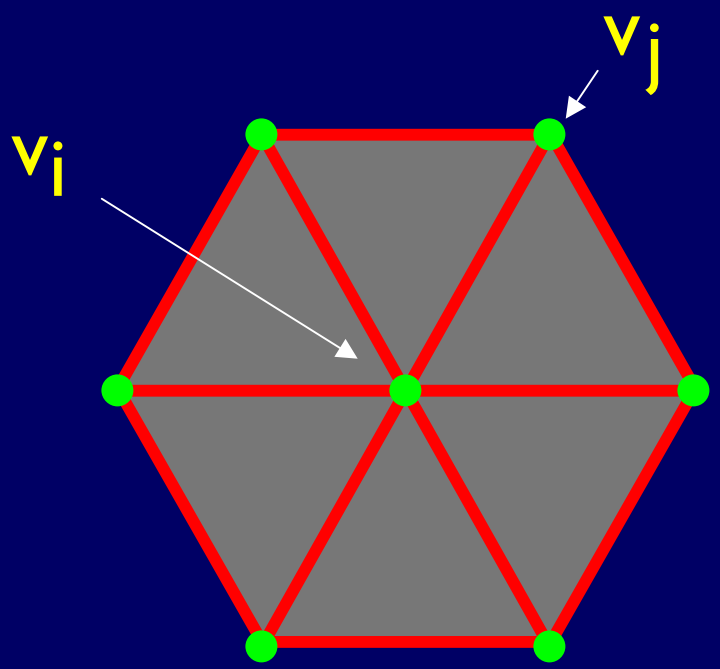
- Laplacian smoothing
  - The shrinkage problem
  - Fourier analysis on meshes
  - Smoothing by partial Fourier expansion
  - Smoothing as low-pass filtering
  - Taubin $\lambda|\mu$ smoothing
  - FIR/IIR filter design
  - Implicit Fairing / Multiresolution modeling
  - Weights / Hard and soft constraints
- Compression of geometry information

Main references

- Taubin λ|μ smoothing (SG'95)
- Taubin-et-al FIR filter design (ECCV'96)
- Desbrun-et-al Implicit smoothing (SG'99)
- Kobelt-et-al Multiresolution smoothing (SG'98)
- Tani-Gotsman Spectral compression (SG'00)
- Balan-Taubin prediction by filtering (CAD'00)
- Khodakovsky-Schroder-Sweldens Progressive Geometry Compression (SG'00)
- Guskov-et-al Multiresolution Signal Processing (SG'99)
- …

Laplacian smoothing in mesh generation

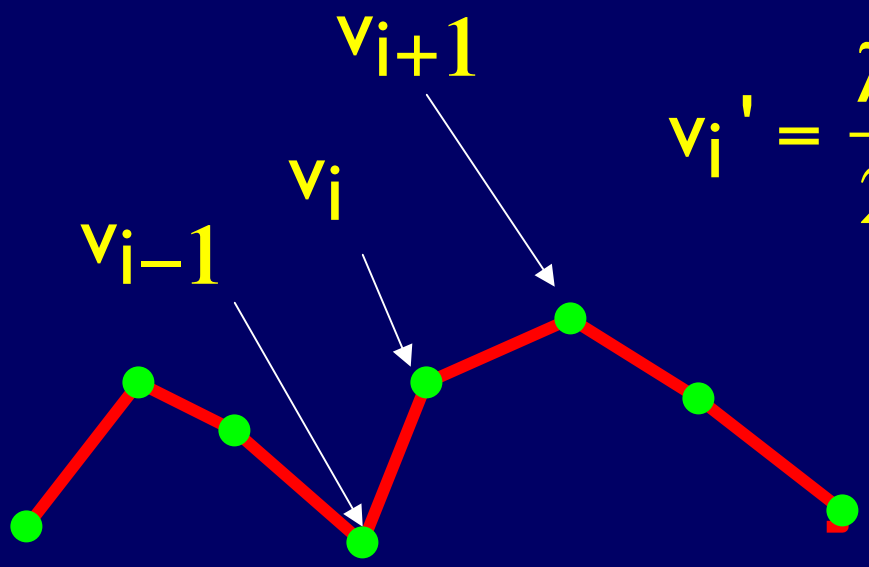
- Used to improve quality of 2D meshes for FE computations
- Move each vertex to the barycenter of its neighbors
- But keep boundary vertices fixed



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

Laplacian smoothing of 1D discrete signals

- Known as Gaussian smoothing
- Convolution of 1D signal with Gaussian kernel
- Also for 2D discrete and continuous signals



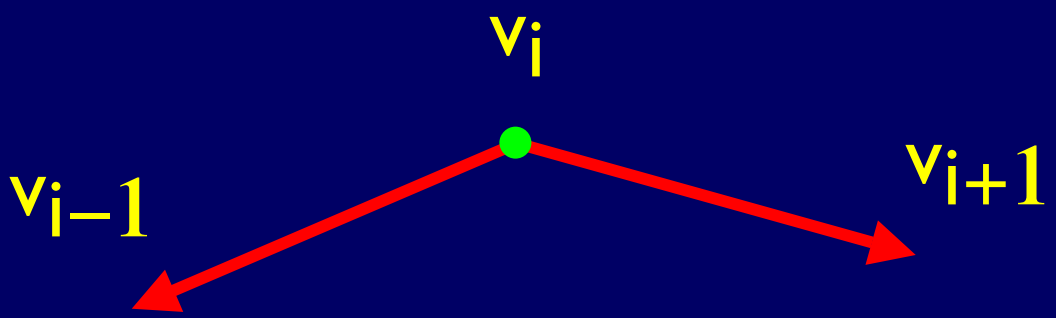
$$v_i' = \frac{\lambda}{2} v_{i-1} + (1-\lambda) v_i + \frac{\lambda}{2} v_{i+1}$$

$$0 < \lambda < 1$$

Laplacian smoothing of 1D discrete signals

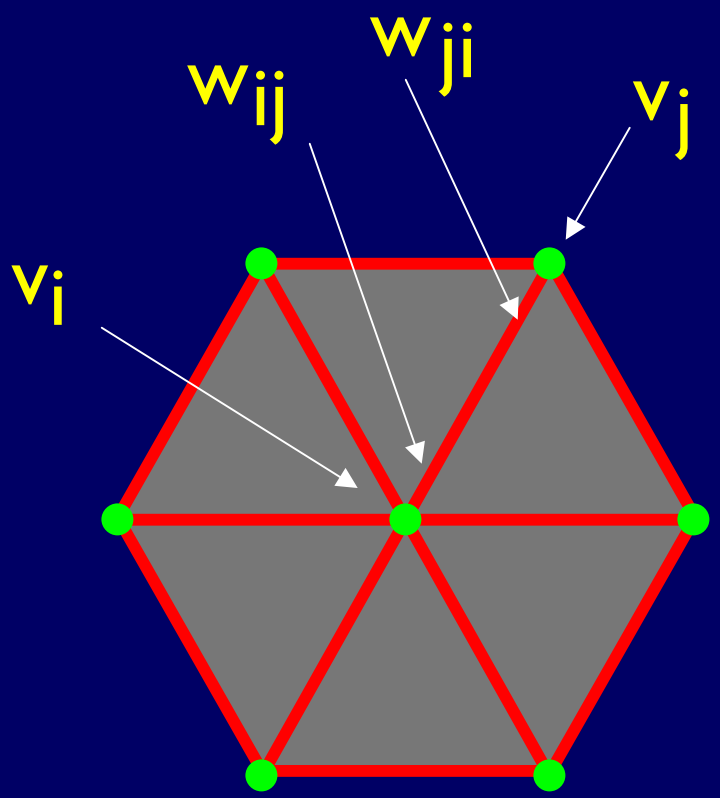
$$v_i' = \frac{\lambda}{2} v_{i-1} + (1-\lambda) v_i + \frac{\lambda}{2} v_{i+1}$$

$$v_i' = v_i + \lambda \left(\frac{1}{2}(v_{i-1} - v_i) + \frac{1}{2}(v_{i+1} - v_i) \right)$$



- Preserves DC

Laplacian smoothing with general weights



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

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

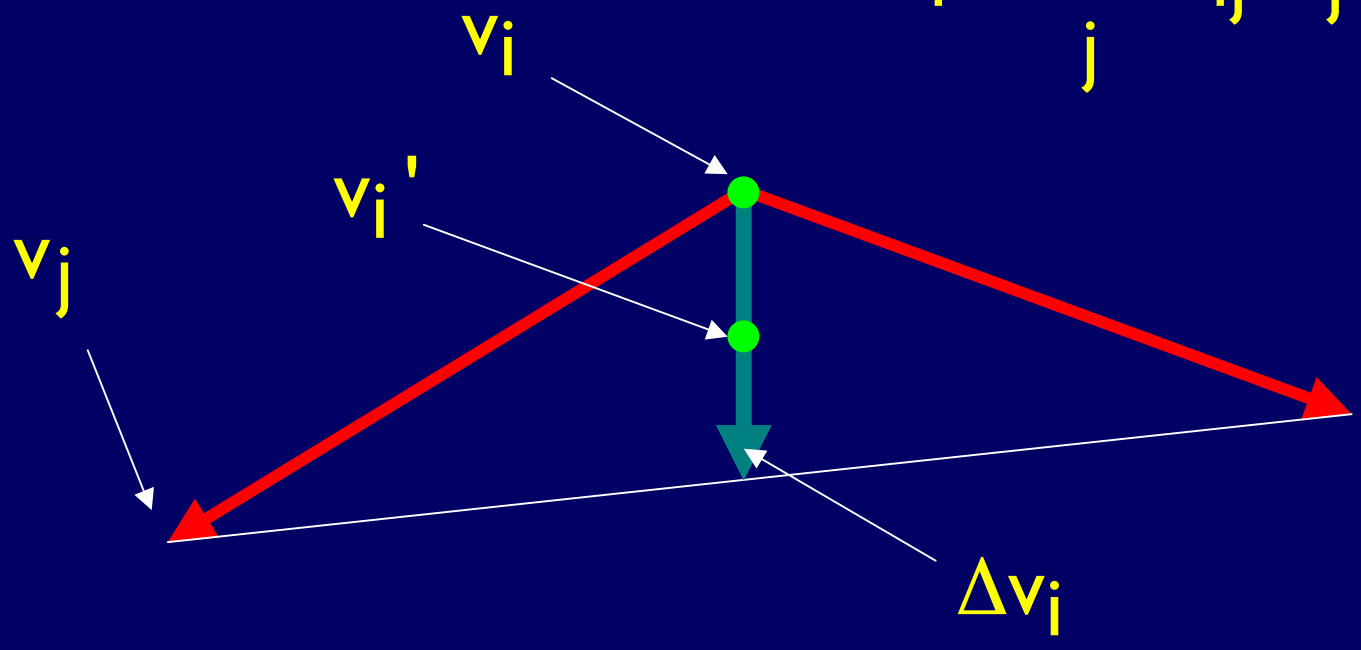
$$1 = \sum_j w_{ij}$$

$$0 \leq w_{ij}$$

The Laplacian operator

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

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

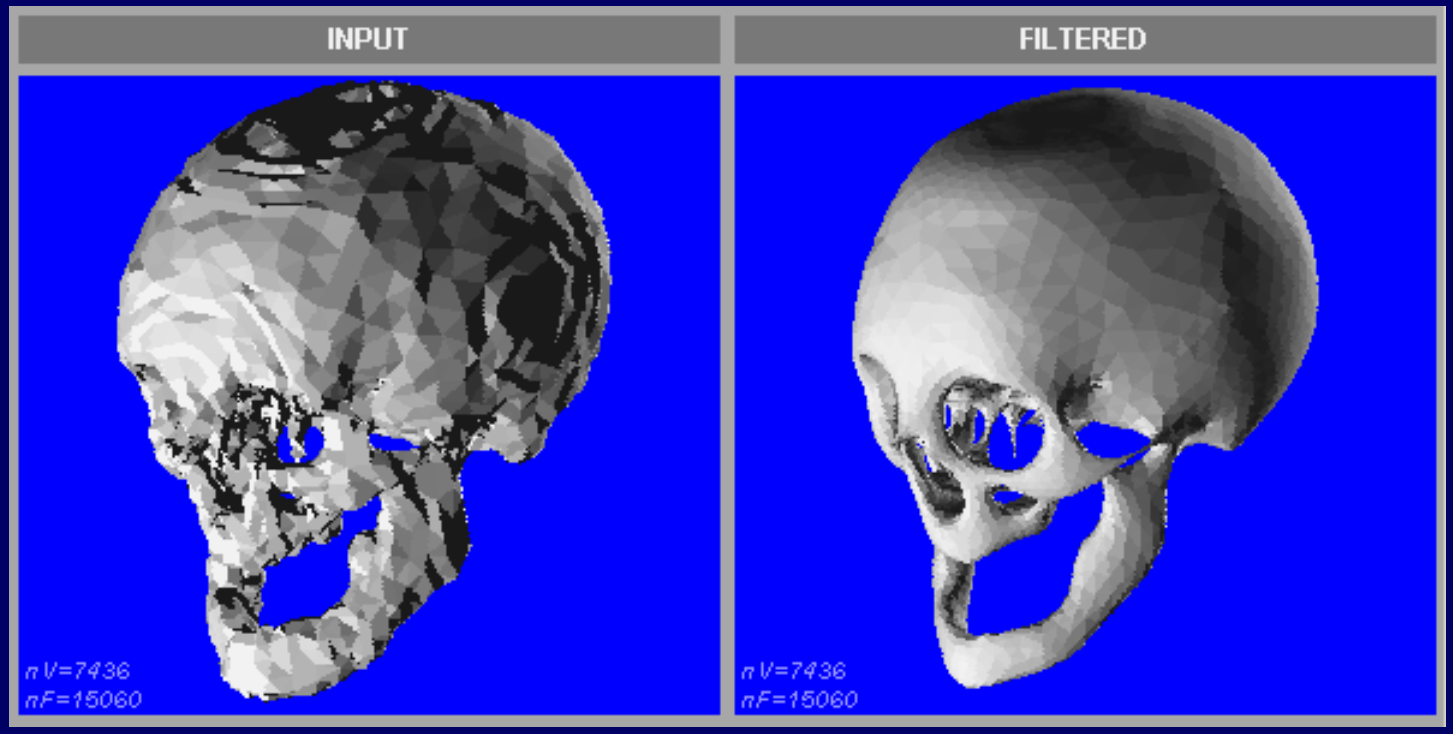


Laplacian smoothing : advantages

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

Shrinkage of Laplacian smoothing

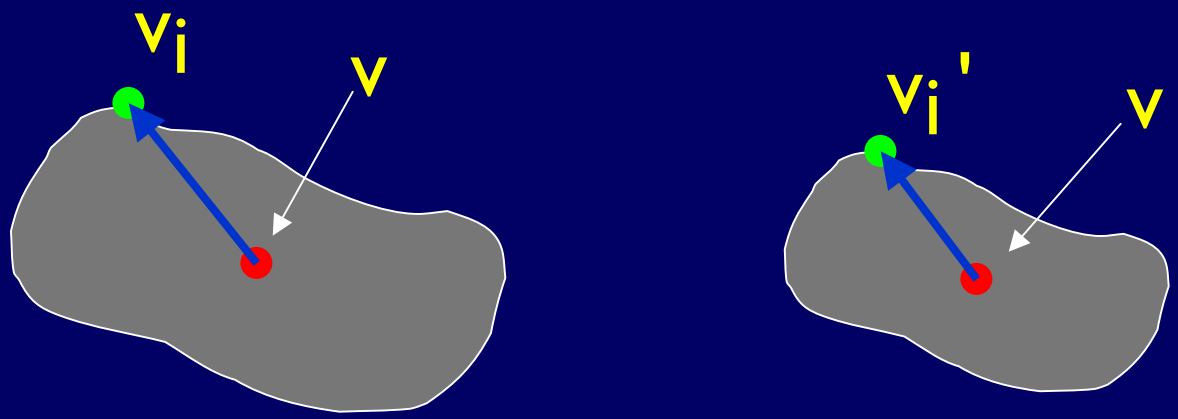
- DEMO !!!



Laplacian smoothing : disadvantages

- Shrinkage
  - Solve by scale adjustment for closed shapes
  - What is going on? Stochastic matrices
  - What is going on? Fourier analysis
  - Solved by Taubin's algorithm for general shapes
- Edge length equalization (depending on the application)
  - Fujiwara weights
  - Desbrun-et-al weights

Fixing shrinkage by renormalizing scale



- Adjust scale s to keep distance to barycenter v constant

$$\sum_i \|v_i - v\|^2 = \sum_i \| s(v_i' - v)\|^2$$

$$v_i'' = v + s(v_i' - v)$$

Fixing shrinkage by renormalizing scale

- It is a global solution
- Local perturbation changes shape everywhere
- For a better solution we need to understand why shrinkage occurs
  - Stochastic matrices
  - Fourier analysis

Stochastic matrices

- Square matrices with non-negative elements
- Sum of rows equal to one
- Related to the asymptotic behavior of Markov chains
- Represent probability of transition from state to state

$$M = (m_{ij}) \qquad m_{ij} \geq 0 \qquad \sum_j m_{ij} = 1$$

- Magnitude of other eigenvalues less than 1
- Powers converge to matrix with eigenvector 1 as rows

$$M^n \rightarrow M^\infty$$

Stochastic matrix of Laplacian smoothing

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

$$v' = Mv \qquad M = (I - \lambda)\, I + \lambda W$$

- Converges to the centroid (barycenter) of the vertices

$$v^n = M^n v \rightarrow M^\infty v = \bar{v}$$

- Why ? Analyze eigenvalues / eigenvectors

Fourier analysis on meshes

$$x_i' = x_i + \lambda \sum_j w_{ij}(x_j - x_i) \qquad x' = (I - \lambda K)\, x$$

- Eigenvalues of $K = I\text{-}W$ (FREQUENCIES)

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

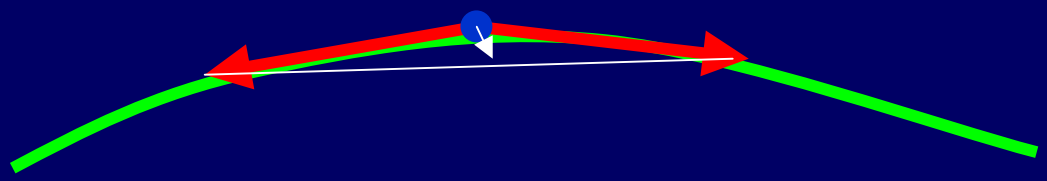
- Right eigenvectors of K (NATURAL VIBRATION MODES)

$$e_0, e_1, \cdots, e_N$$

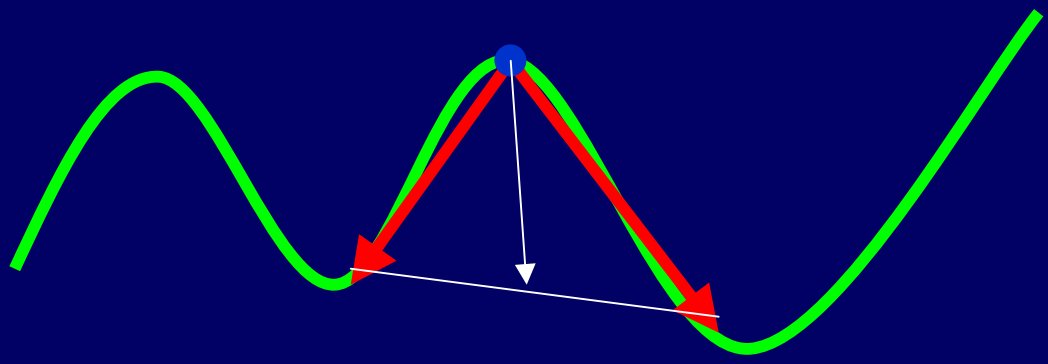
Geometry of low and high frequencies

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

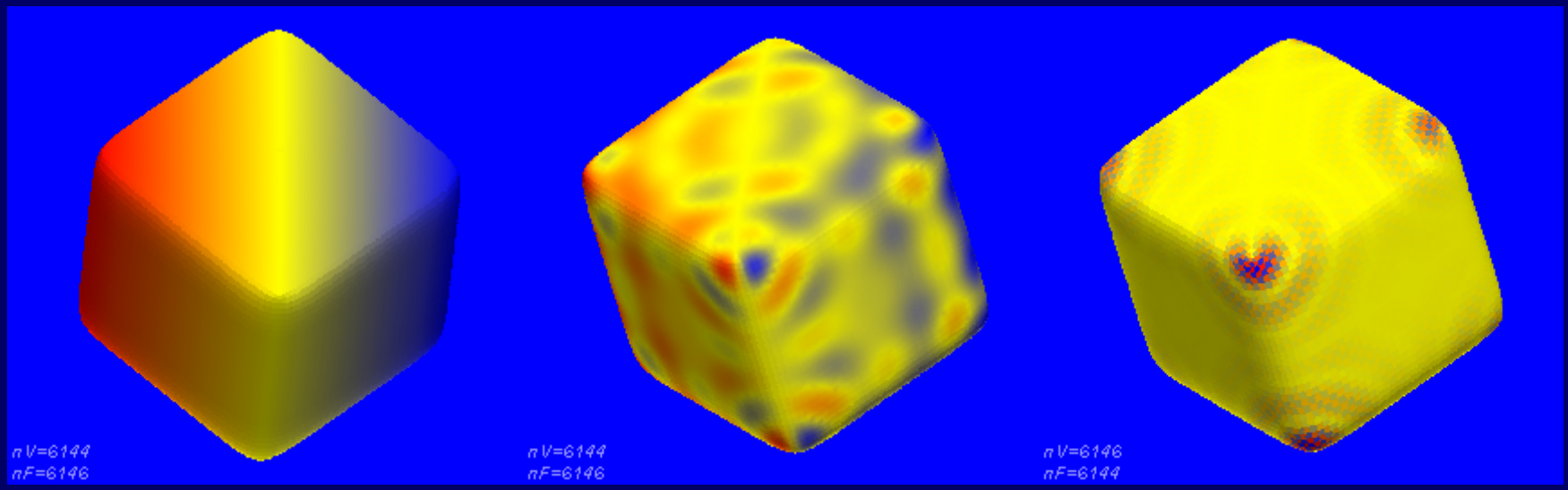
- Low frequency



- 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

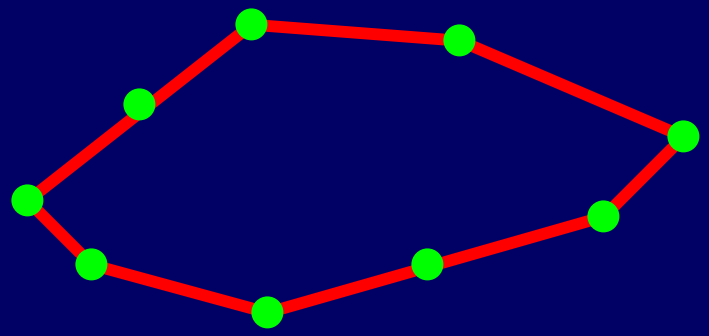
$$x = \hat{x}_0 e_0 + \hat{x}_1 e_1 + \cdots + \hat{x}_N e_N$$

- Discrete Fourier Transform (DFT)

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

The Discrete Fourier Transform

- Corresponds to the classical definition for 1D periodic signals

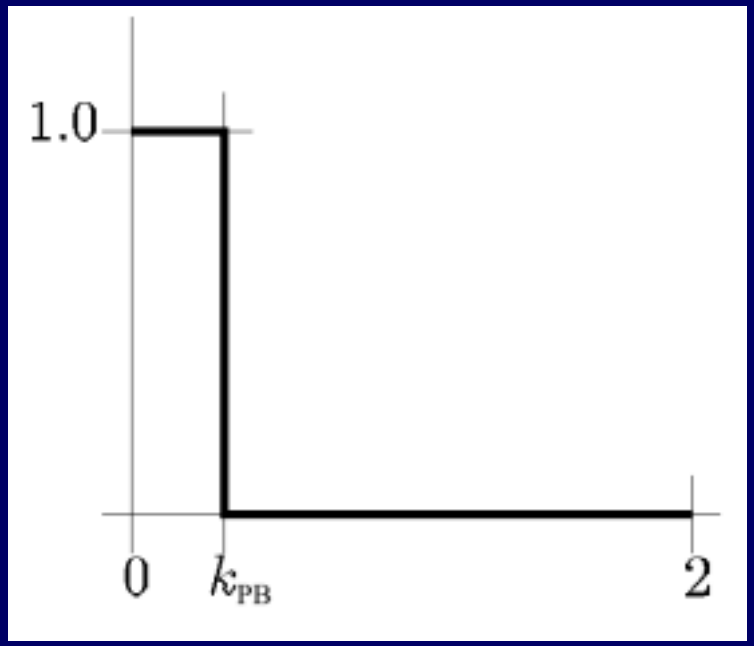


- For 1D periodic signals there is a fast algorithm to compute the DFT : the FFT
- For the general case of signals defined on irregular meshes, DFT is almost impossible to compute

The Ideal Low-Pass Filter

- Truncated Fourier expansion

$$x' = \hat{x}_0 e_0 + \hat{x}_1 e_1 + \cdots + \hat{x}_L e_L$$



$$k_L \le k_{PB}$$

The Discrete Fourier Transform

- Ideal low-pass filtering = truncated Fourier expansion

$$x' = 1\hat{x}_0 e_0 + \cdots + 1\hat{x}_L e_L$$

$$+ 0\hat{x}_{L+1} e_{L+1} + \cdots + 0\hat{x}_N e_N$$

- But eigenvectors cannot be computed !
- Compute an approximation instead : Linear filtering

Analysis of Laplacian smoothing

- Laplacian smoothing transfer function

$$x^N = (I - \lambda K)^N x = f(K)\, x$$

- f(k) univariate polynomial (rational later)
- f(K) matrix
- K and f(K) have same eigenvectors
- Eigenvalues of f(K)

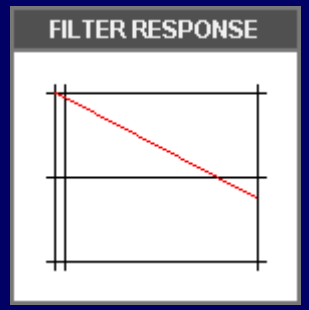
$$f(k_0)\ ,\ f(k_1)\ ,\ \ldots\ ,\ f(k_N)$$

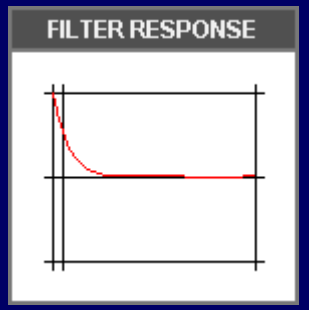
Laplacian Smoothing is a Linear Filter

- 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





$$f(k_0) = 1$$

$$f(k_j) = (1 - \lambda\, k_j)^N \rightarrow 0 \qquad j \neq 0 \qquad 0 \leq \lambda < 1$$

- Laplacian smoothing is not a low-pass filter !

Linear Filtering

- After filtering

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

- Evaluation of $f(K)\,x$ is matrix multiplication
- It does not require the computation of eigenvalues and eigenvectors (DFT)

Low-Pass Linear Filtering

- After filtering

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

- Need to find univariate polynomial f(k) such that

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

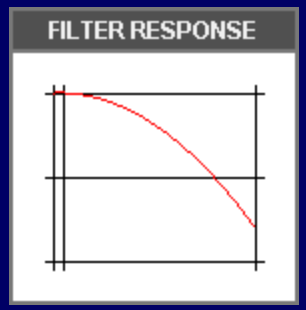
$$f(k_h) \approx 0 \qquad k_L > k_{PB}$$

- Need to define efficient evaluation algorithm

Taubin smoothing (Siggraph'95)

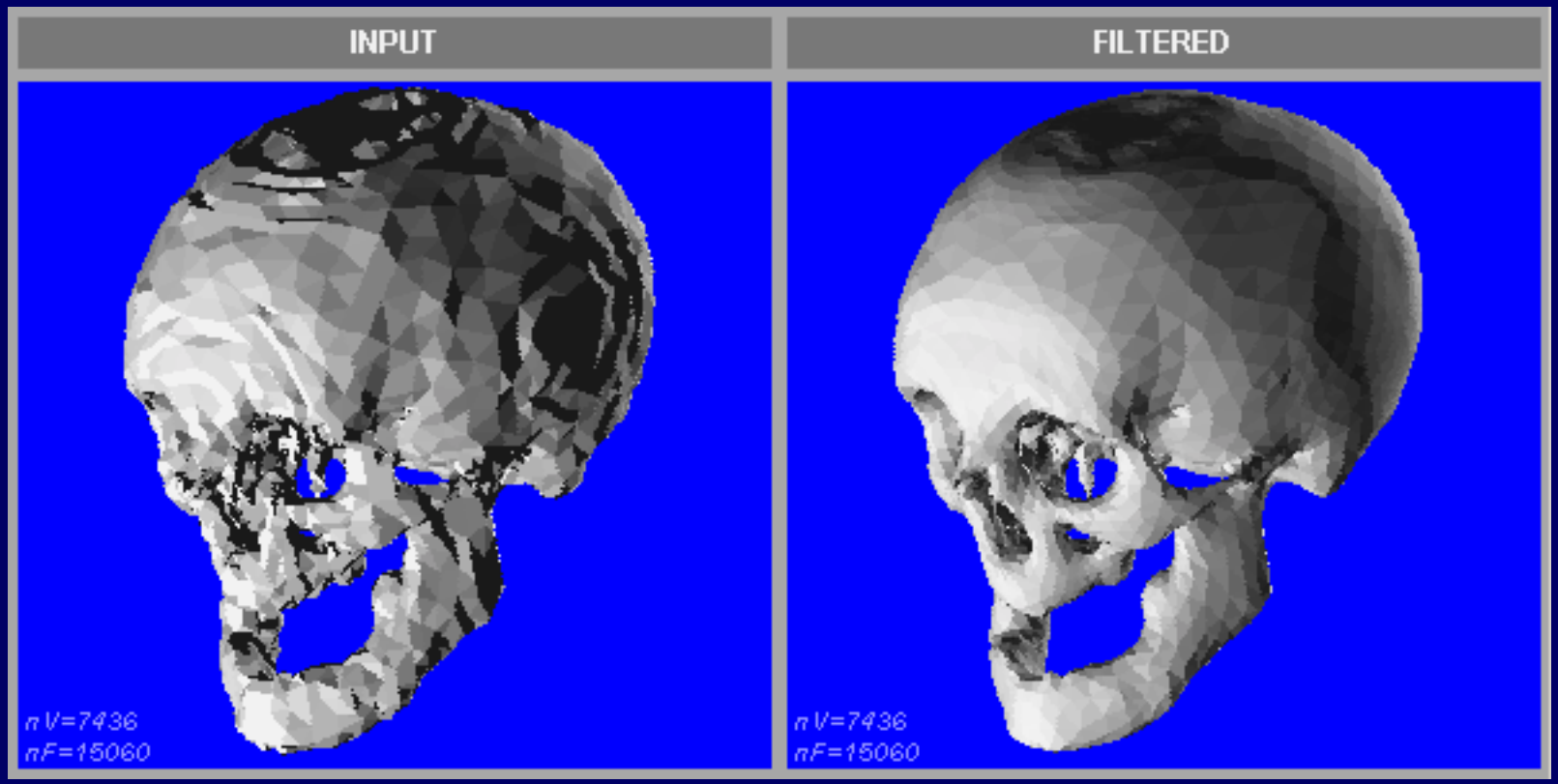
- Two steps of Laplacian smoothing
- First shrinking step with positive factor
- Second unshrinking step with negative factor
- Use inverted parabola as transfer function

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



Taubin smoothing (Siggraph'95)

- DEMO !!!



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
- DEMO !!!

FIR filters vs. IIR filters

- Sharp transitions and narrow pass-bands require very high degree polynomial transfer functions
- Infinite Inpulse Response (IIR) filters with rational transfer functions can produce good approximations using polynomials of low degree
- But require the solution of sparse linear systems
- Is it worth the effort ?

IIR filters

- If $f(k)=g(k)/h(k)$, with $h(k)$ non-zero in $[0,2]$
- Filtering a signal x requires solving the system

$$h(K)\,x' = g(K)\,x$$

- $y = g(K)\,x$ is an FIR filter
- With $H = h(K)$ solving $H\,x' = y$ with the Preconditioned Biconjugate Gradients algorithm (PBCG) only requires methods to multiply a vector z by H and by H^t and a preconditioner H'

Desbrun-Meyer-Schroder-Barr (SG'99) Implicit fairing

- Corresponds to the classical Butterworth filter with transfer function

$$f(k) = \frac{1}{1 + (k / k_{PB})^N}$$

- Need to solve sparse (for small N) linear system

$$(I + (I / k_{PB})^N K^N) x' = x$$

- But with PDE formulation in the paper

Implicit fairing

- Laplacian smoothing corresponds to the numerical solution of

$$\frac{\partial x}{\partial t} = \lambda dt \, \Delta x$$

- using the forward Euler method

$$x' = x + \lambda dt \, \Delta x = (I + \lambda dt \, \Delta) x$$

- They use the backward Euler method instead

$$(I - \lambda dt \, \Delta) x' = x$$

- Stable for large time steps (true or false ?)
- DEMO !!!

Kobelt-et-al Multiresolution modeling (Siggraph'98)

- Minimize membrane energy $E_M = \|\Delta x\|^2$
- or thin plate energy $E_{TP} = \|\Delta^2 x\|^2$
- Requires boundary vertex position constraints
- Speed-up by multi-grid approach
- Jacobi updates similar to Laplacian and Taubin updates
- How does it compare with single-res FIR filters ?
- DEMO !!!

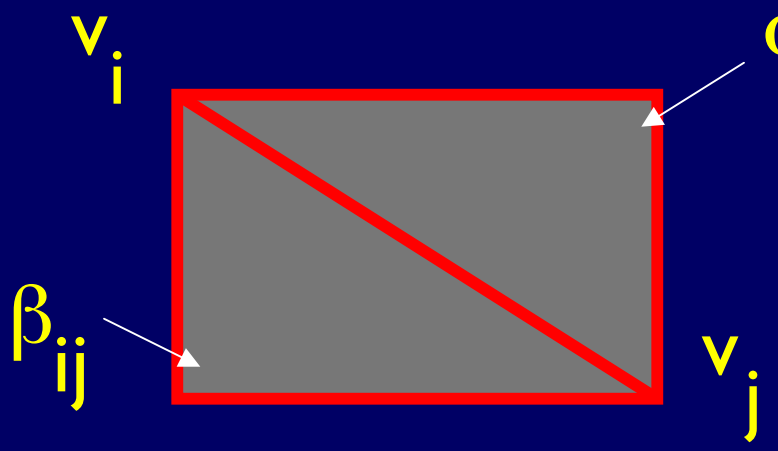
Parameters

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

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

Preventing tangencial drift

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

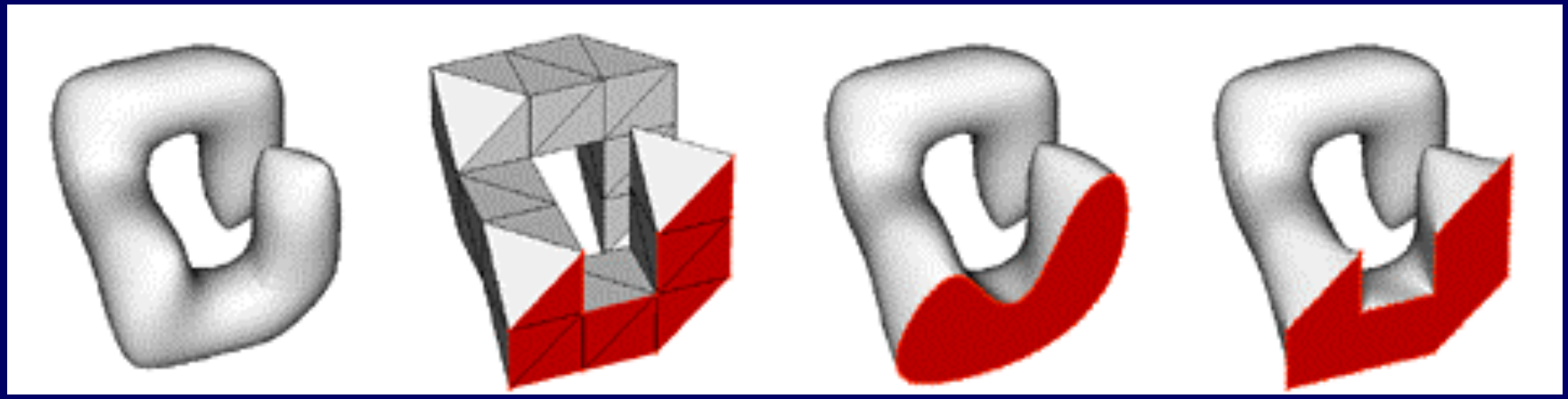


$$c_{ij} = \cot(\alpha_{ij}) + \cot(\beta_{ij})'$$

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

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

Position and normal constraints

- Hard vs. soft constraints
- Hard vertex position constraints are easy to impose
- General hard linear constraints require solving small linear systems
- 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
- Multi-resolution helps
- More work needed

Geometry compression

- Static or single-resolution vs. progressive
- Connectivity, geometry, and properties
- Geometry and properties cost much more than connectivity
- Commercial grade single-resolution methods available
  - Taubin-Rossignac Topological Surgery (MPEG-4/ IBM HotMedia)
  - Touma-Gotsman (Virtue Ltd.)
- Need better geometry prediction/compression schemes

Tani-Gotsman (Siggraph'00) Spectral Compression

- Based on partial DFT expansion
- Connectivity is transmitted first
- Encoder computes Eigenvalues/Eigenvectors of matrix K to evaluate Fourier coefficients
- Fourier coefficients are transmitted
- Decoder computes Eigenvalues/Eigenvectors of matrix K to reconstruct the partial sum
- Mesh partition into smaller submeshes to be able to deal with the numerical restrictions
- Need to compute lots of Eigenvalues/Eigenvectors

Balan-Taubin prediction by filtering (CAD'00)

- Based on a vertex clustering hierarchy (PM, PFS, etc.)
- Connectivity is transmitted progressively interlieved with geometry data
- Fine Geometry is predicted from coarse geometry by filtering the coarse geometry on the fine mesh
- Filter coefficients are determined by solving a LS problem
- Corrections are not transmitted

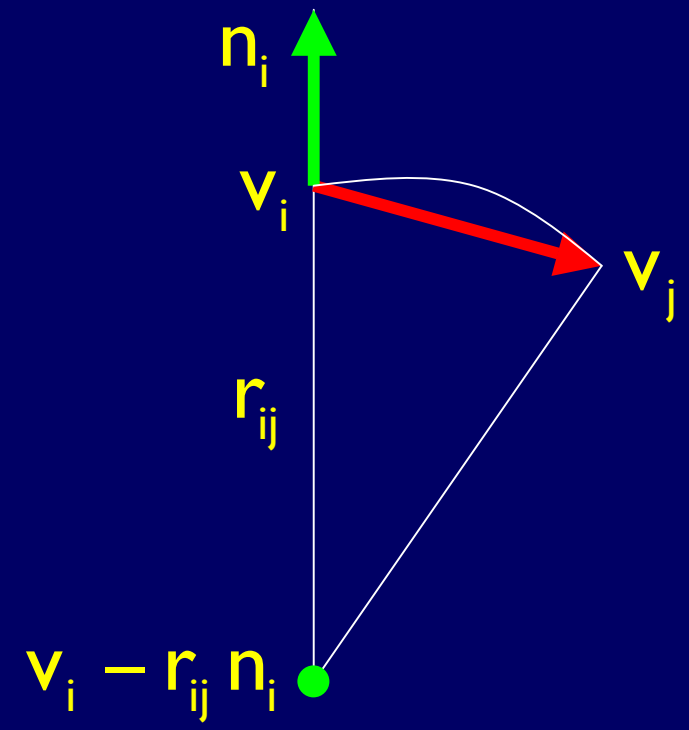
$$\min_f \left\| x_F - f(K)\, x_C \right\|^2$$

Khodakovsky-Schroder-Sweldens Progressive Geometry Compression (SG'00)

- Good for large densely sampled meshes with low topological complexity (3D scanning, etc.)
- MAPS Remeshing produces subdivision surface
- Wavelet compression
- Zero-tree encoding
- Very good results reported

Curvature-based Sampling

- Silva-Taubin Curvature-based sampling (SIAM-GD'99)
- Taubin Tensor of curvature (ICCV'95)



$$\left\| v_j - v_i + r_{ij}\, n_i \right\|^2 = r_{ij}^{\,2}$$

$$\sigma_{ij} = \frac{\left\| v_j - v_i \right\|}{r_{ij}} = \frac{2\, n_i^{\,t} (v_j - v_i)}{\left\| v_j - v_i \right\|}$$

Conclusion / To Do

- Fast and efficient methods to smooth with hard and soft constraints
- Relation to subdivision surfaces
- Global vs. local behavior of smoothing operators
- Goal: interactive free-form modeling based on intuitive user interface to manipulate constraints, remesh, simplify, etc.
- Goal: practical and effective methods for the compression of geometry data.
- Implementation of other popular SP operations