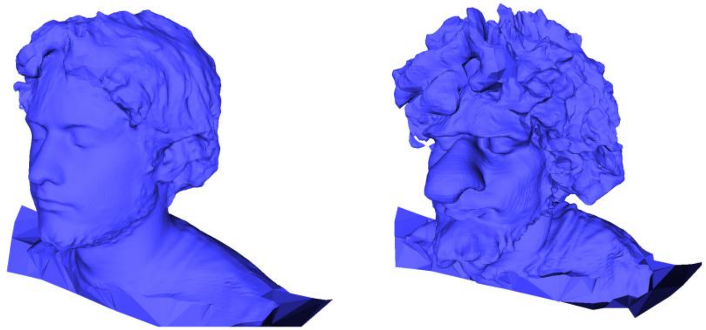
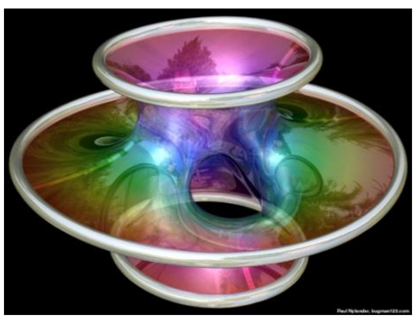
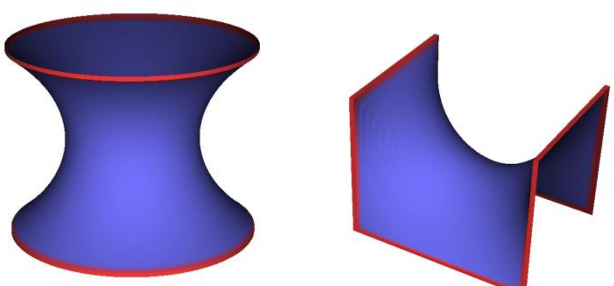
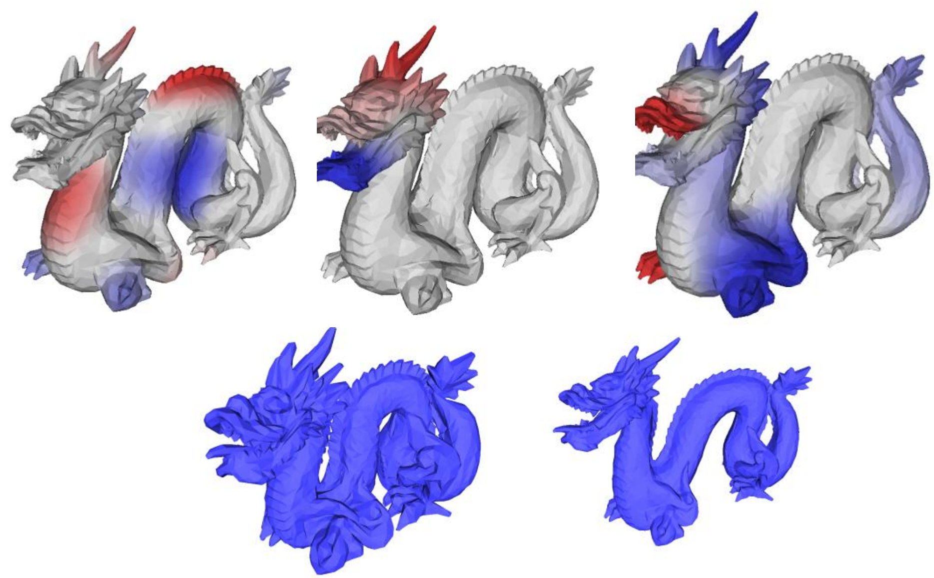
Assignment 4

Mesh Laplacians and Applications









Laplacians

- Discussed in the differential geometry chapter of the mesh processing book.

- Discretizations:
  - Uniform Laplacian (page 44)

$$\Delta f(v_i) = \frac{1}{|\mathcal{N}_1(v_i)|} \sum_{v_j \in \mathcal{N}_1(v_i)} (f_j - f_i)$$

  - Cotangent Laplacian (page 46, Assignment 1)

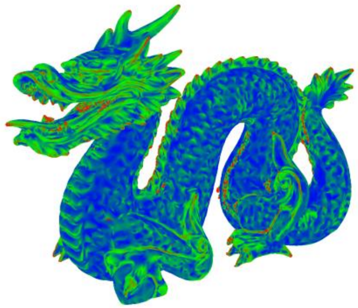
$$\Delta f(v_i) := \frac{1}{2A_i} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot\alpha_{i,j} + \cot\beta_{i,j})(f_j - f_i) \, .$$

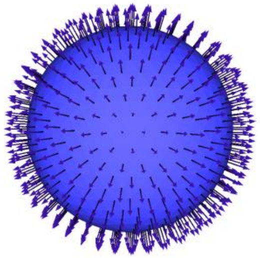
Laplacians

- Properties:
  - Meancurvature normal property

$$\Delta_{\mathcal{S}}\,\mathbf{x} = -2H\mathbf{n}.$$

Only true for the Laplacian associated to S!!

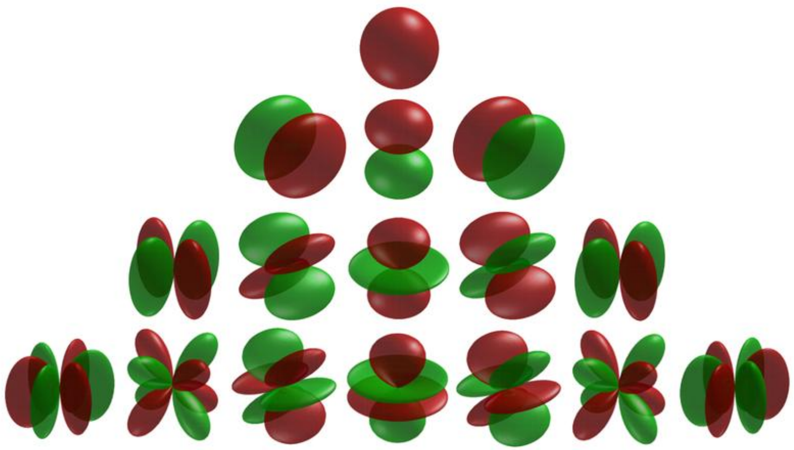




Where x are the positional surface coordinates of the surface S, $\Delta_{\mathcal{S}}$ is the Laplacian associated to S, H is the mean curvature

Laplacians

- Properties:
  - Very interesting eigenfunctions!



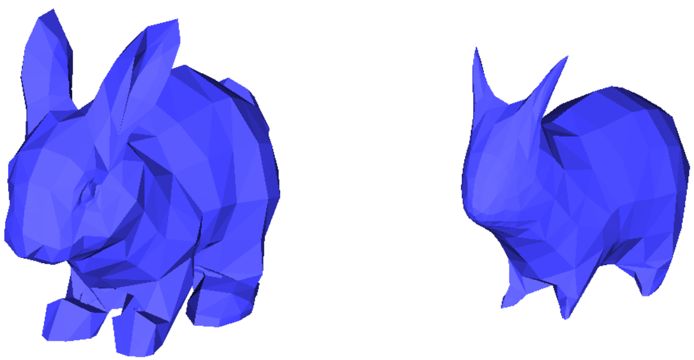
Laplacians: Pros and Cons

- Uniform Laplacian:
  - Simple to implement
  - Numerically stable, as its definition does not take triangle shapes into count.
  - Only a coarse approximation of the Laplace operator

- Cotangent Laplacian:
  - Better approximation of the Laplace operator
  - Degenerates when the triangles degenerate too much.

Implicit Smoothing

- Scheme discussed in the lecture.
- Problem: shrinking!



- Solution: rescale mesh such that the volume is preserved, i.e. scale the mesh with

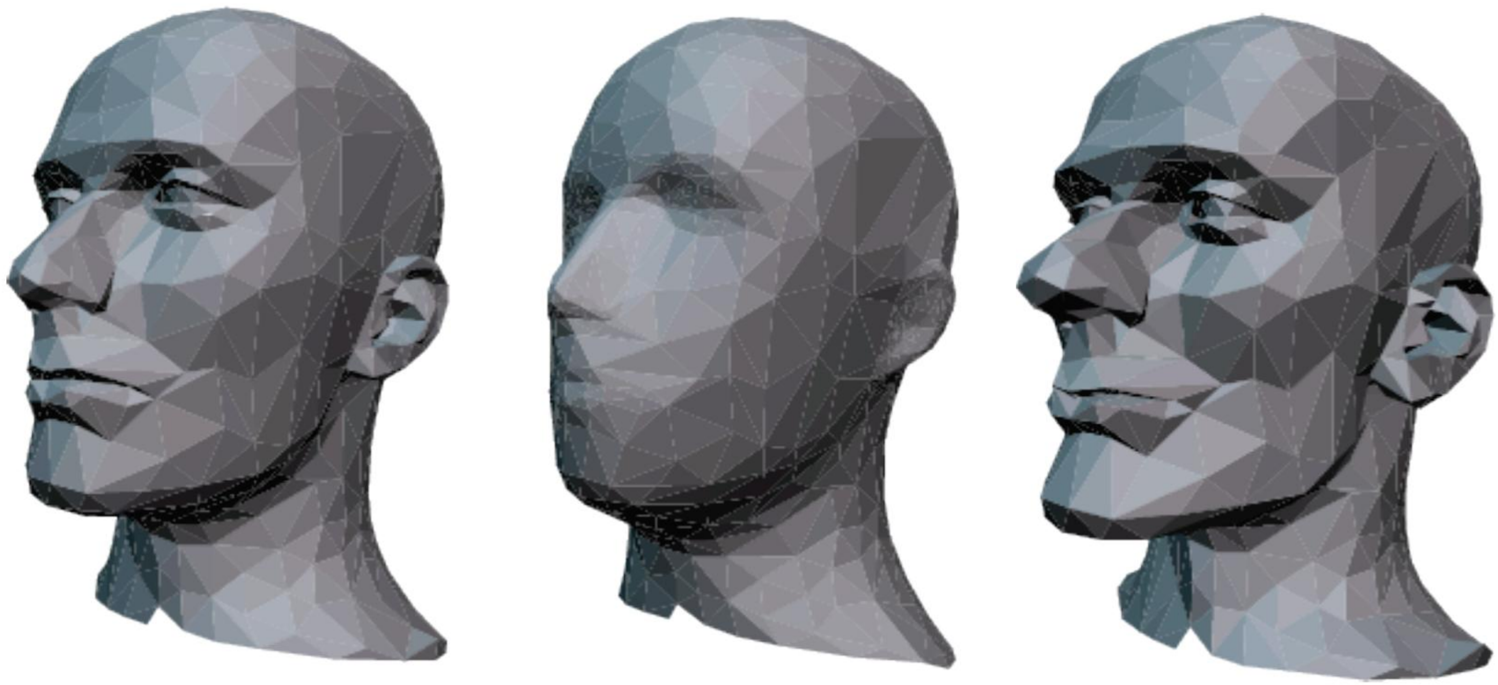
$$\left(\frac{vol_{old}}{vol_{new}}\right)^{1/3}$$

Implicit Smoothing

- Volume of a mesh

$$\sum_{triangles\{p1,p2,p3\}} \langle p1, p2 \times p3 \rangle / 6$$

Unsharp Masking



Input

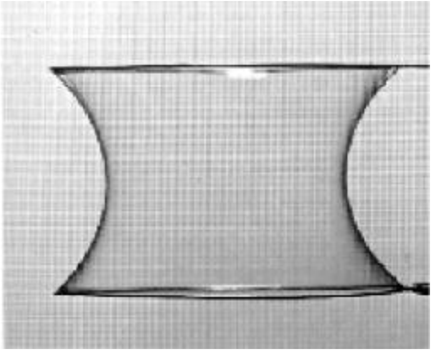
Smoothed

Enhanced details
(input-smoothed)*s+smoothed,
s>1

http://www.multires.caltech.edu/pubs/irrsub.pdf

Minimal Surfaces

- Examples



- Surface area is minimal under some (boundary) constraints

Minimal Surfaces

http://rkneufeld.wordpress.com/2010/10/27/minimal-surfaces-and-the-area-functional-2/

- Theory
  - Area functional:

$$A(S) = \int_S dA$$

  - Perturbation by some phi

$$A(S + t\phi) \quad t \in \mathbb{R}$$

  - Surface is minimal if

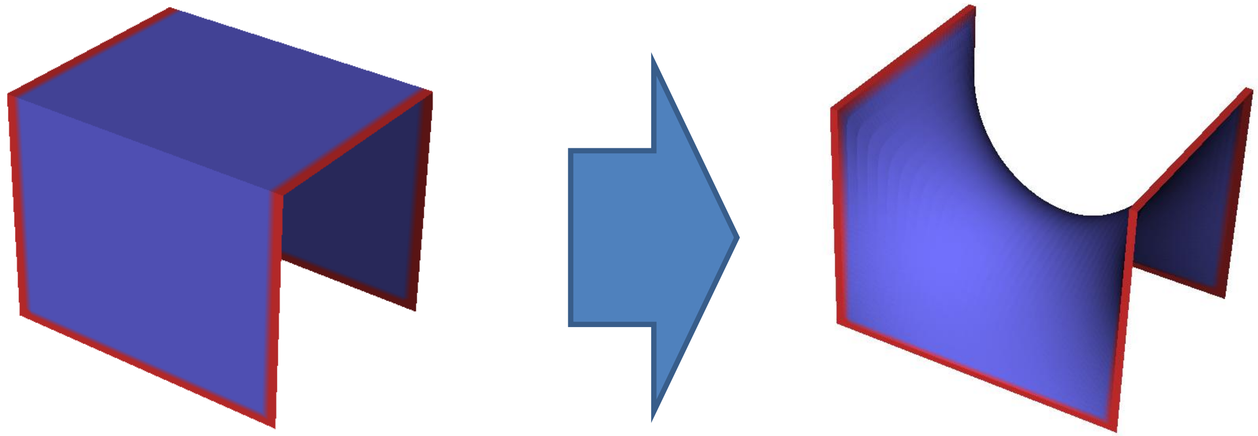
$$\frac{\partial}{\partial t} A(S + t\phi) = 0 \quad \text{for all perturbations } \phi$$

  - Some Math later....

$$\Rightarrow meancurvature(A) = 0$$

Minimal Surfaces

- Your task:
  - Input: a mesh with the correct connectivity
  - Output: mesh with same boundary and zero mean curvature.



Minimal Surfaces

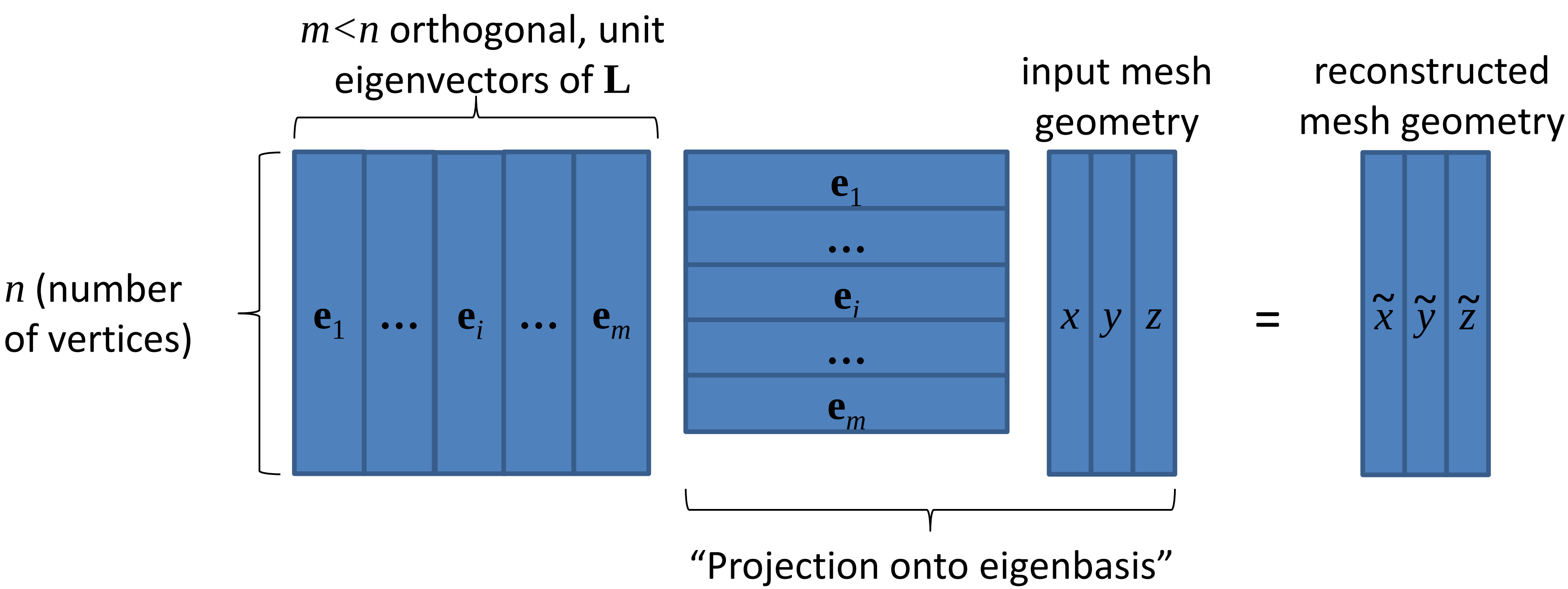
- Algorithm : Input mesh S with positions (x,y,z)
  - Solve for zero curvature mesh using the mesh Laplacian L_S

$$
\begin{aligned}
L_S(x', y', z') &= 0 && \text{inside the mesh} \\
I(x', y', z') &= (x, y, z) && \text{On the mesh boundary}
\end{aligned}
$$

  - Update: (x,y,z) ← (x‘,y‘,z‘)
  - Recompute L_S, solve again, until convergence.

Spectral Mesh Processing

- In General:
  - Compute eigenvectors/ values of Laplacian
  - Reproject onto this new basis



Spectral Mesh Processing

- Problem:
  - Need symmetric Laplacian matrix
  - To guarantee existence of real eigenvectors.

- Adapt cotangent matrix:
  - Instead of

$$\Delta_{ij} = -\frac{1}{\mathcal{A}_i}(cotan(P) + cotan(Q))$$

$$\Delta_{ii} = -\sum_{j \neq i} \Delta_{ij}$$

  - Use

$$\Delta_{ij} = -\frac{1}{\sqrt{\mathcal{A}_i \mathcal{A}_j}}(cotan(P) + cotan(Q))$$

$$\Delta_{ii} = -\sum_{j \neq i} \Delta_{ij}$$

Spectral Mesh Processing

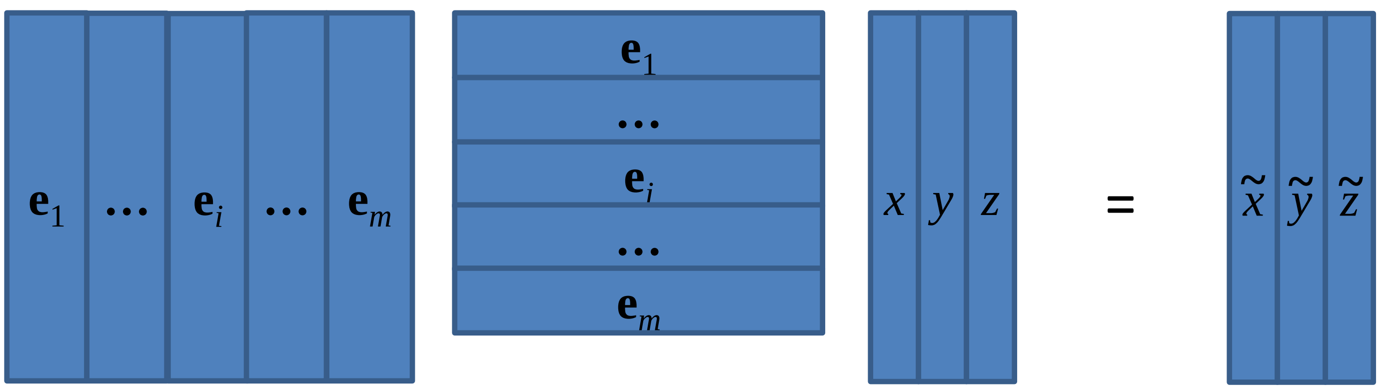
- Eigen vectors and eigenvalues
  - Relation to frequencies

$$freq = \sqrt{|Eigenvalue|}$$

  - Low frequencies = coarse shape
  - High frequencies = details
  - Spectral filtering, steal image from paper.

Spectral Mesh processing

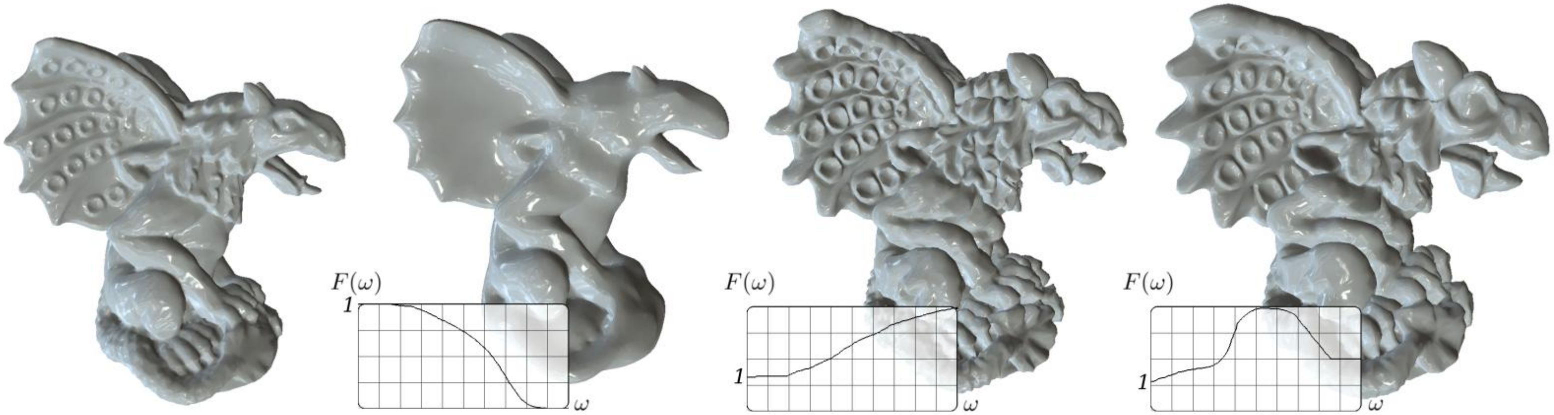
- Spectral Filtering. Instead of only projecting onto the eigenfunctions, increase/decrese importance of some frequencies



$$\begin{pmatrix} e_1 & \cdots & e_n \end{pmatrix} \begin{pmatrix} f(freq(e_1)) & & & \\ & f(freq(e_2)) & & \\ & & \ddots & \\ & & & f(freq(e_n)) \end{pmatrix} \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} (x, y, z) = (\tilde{x}, \tilde{y}, \tilde{z})$$

Spectral Mesh Processing

- Example filters.



$$\omega = \sqrt{|Eigenvalue|}$$

The Numerical Side......

- Linear equations:
  - SciPy Solver: Slow and reliable
  - JMTSolver: Fast, but does not always converge.
  - For this assignment the JMTSolver usually does the job.

- Eigenvalue decomposition:
  - SCIPYEVD.java
  - Calls Python script
  - Works only for small (< 10‘000 x 10‘000) matrices.

Questions?