Lecture 12
Mesh Smoothing

Overview

- Mesh smoothing

  = design + computation of smooth functions on a triangle mesh $f: S \rightarrow R^d$

  → A fundamental tool

  → f can be describing vertex positions, texture coordinates, vertex displacements…

- Applications
  - Denoising: to remove high-frequency noise
  - Fairing: deform a surface to a smooth patch
  - Surface parameterization, remeshing, hole filling, deformation…

Surface Smoothing Techniques

- Spectrum approaches
  - Discrete Cosine Transform (DCT) (used in JPEG)
  - Spherical Harmonics
  - Manifold Harmonics [Vallet and Levy, “Spectral Geometry Processing with Manifold Harmonic,” CGF’08]
  - Eigenvectors of Discrete Laplace matrix
  - Filter out high frequency components
  - Elegant but expensive to compute
- Diffusion flow
  - Cheaper and therefore more widely used

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = \lambda \Delta f(\mathbf{x},t).$$

Heat Diffusion

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = \lambda \Delta f(\mathbf{x},t).$$

- A second-order linear partial differential equation (PDE)
- f changes over time by a scalar diffusion coefficient λ times its spatial Laplacian Δf
- f(x,t) usually denotes the temperature at time t of a point x, the equation describes the temporal heat diffusion in an object; on a surface, replace the regular Laplace operator → Laplace-Beltrami
- Can be used to smooth function on a manifold surface S

Discrete Heat Diffusion

- Usually a continuous time-dependent PDE → discreteize it both in space and in time
- Spatial discretization
  - Consider sample values at the vertices and use discrete Laplace matrix:

$$\frac{\partial}{\partial t} f(v_i, t) = \lambda \Delta f(v_i, t), \quad i = 1, \dots, n,$$

- Temporal discretization

$$\frac{\partial \mathbf{f}(t)}{\partial t} \approx \frac{\mathbf{f}(t+h) - \mathbf{f}(t)}{h}$$

$$\mathbf{f}(t+h) = \mathbf{f}(t) + h \frac{\partial \mathbf{f}(t)}{\partial t} = \mathbf{f}(t) + h \lambda \mathbf{L} \mathbf{f}(t).$$

Discrete Heat Diffusion

- can be updated on vertex positions iteratively

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + h\lambda\Delta\mathbf{x}_i.$$

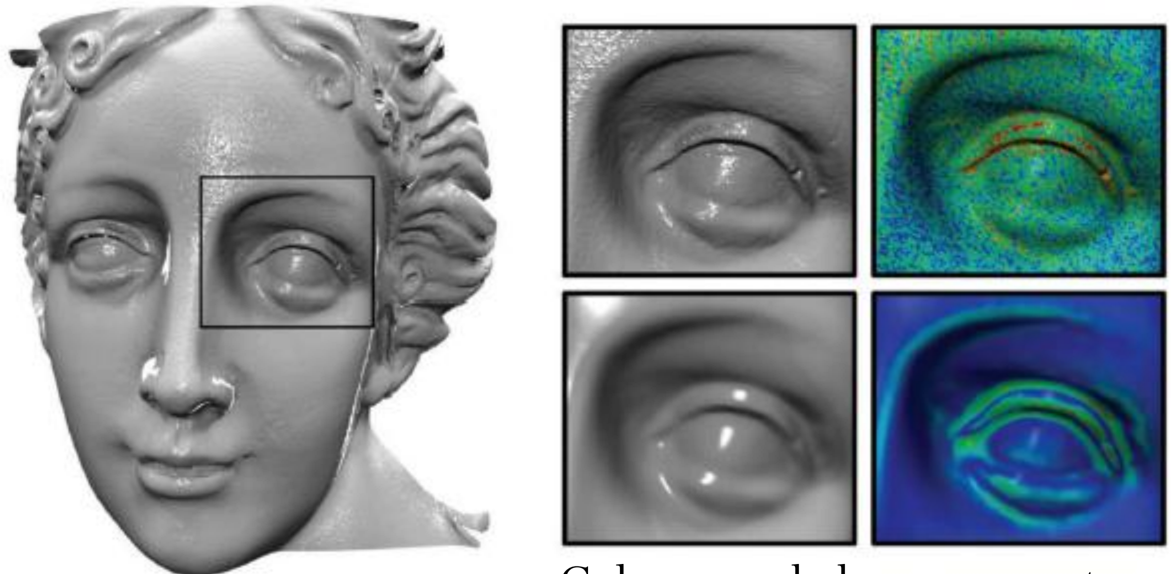
- It is called Laplacian smoothing

- Laplace-Beltrami of vertex positions = mean curvature normal

$$\Delta\mathbf{x} = -2H\mathbf{n}$$

  - Vertices move in the normal direction by an amount determined by the mean curvature H
  - Therefore, also called mean curvature flow

Denoising using Heat Diffusion



Color-encoded mean curvature

Surface Fairing

- Denoising: to remove high frequency noise
- Fairing: to construct shapes that are smooth to fill holes, or to blend two patches
- Frequently used functional:
  - Membrane energy (Dirichlet energy)

  → Membrane surface or minimal surface

$$\tilde{E}_{\mathrm{M}}(\mathbf{x}) = \iint_{\Omega} \|\mathbf{x}_u\|^2 + \|\mathbf{x}_v\|^2 \, \mathrm{d}u \mathrm{d}v,$$

The membrane surface must have $\Delta f = 0.$

Surface Fairing

- Higher-order flows can also be applied

  $$\partial \mathbf{f} / \partial t = \lambda \Delta^k \mathbf{f}$$

  - More expensive, but better smoothness between the smoothed/constructed and the fixed region
- Higher order functional:
  - Thin-plate energy (minimizing curvature)

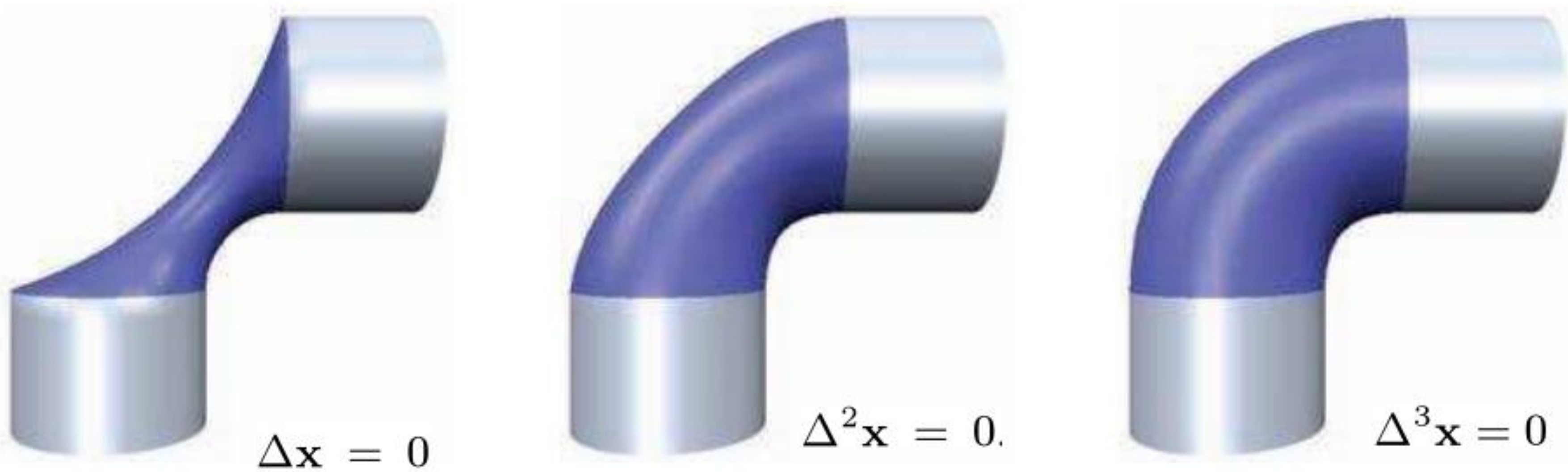
  $$\tilde{E}_{\mathrm{TP}}(\mathbf{x}) = \iint_{\Omega} \|\mathbf{x}_{uu}\|^2 + 2\|\mathbf{x}_{uv}\|^2 + \|\mathbf{x}_{vv}\|^2 \,\mathrm{d}u\mathrm{d}v.$$

  $$\Delta^2 \mathbf{x}(u,v) = 0$$

  - Minimizing variation of curvature

  $$\iint_{\Omega} \left(\frac{\partial \kappa_1}{\partial \mathbf{t}_1}\right)^2 + \left(\frac{\partial \kappa_2}{\partial \mathbf{t}_2}\right)^2 \mathrm{d}u\mathrm{d}v, \qquad \Delta^3 \mathbf{x} = 0$$

Higher order fairing



[Botsch and Kobbelt, an intuitive framework for real-time freeform modeling, TOG'04]