Outline

• Differential surface representation
• Ideas and applications
  – Compact shape representation
  – Mesh editing and manipulation
  – Membrane and flattening
Motivation

• Meshes are great, but:
  – Geometry is represented in a global coordinate system
    • Single Cartesian coordinate of a vertex doesn’t say much
Laplacian Mesh Editing

- Meshes are difficult to edit
Motivation

• *Meshes are difficult to edit*
Motivation

- Meshes are difficult to edit
Differential Coordinates

- Represent a point *relative* to its neighbors
- Represent *local detail* at each surface point
  - better describe the shape
- Linear transition from global to differential
- Useful for operations on surfaces where surface details are important
Differential Coordinates

- Detail = surface – smooth(surface)
- Smoothing = averaging

\[
\delta_i = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j
\]

\[
\delta_i = \sum_{j \in N(i)} \frac{1}{d_i} (v_i - v_j)
\]

"Local control for mesh morphing", Alexa 01
Connection to the Smooth Case

- The direction of $\delta_i$ approximates the normal
- The size approximates the mean curvature $H$
  \[ H \approx \frac{1}{\text{radius of local best-fit sphere}} \]
- Laplace-Beltrami operator on surface (like Laplacian of a 2D function)

\[
\delta_i = \frac{1}{d_i} \sum_{v \in N(i)} (v_i - v)
\]

\[
\frac{1}{\text{len}(\gamma)} \int_{v \in \gamma} (v_i - v) \, ds
\]

\[
\lim_{\text{len}(\gamma) \to 0} \frac{1}{\text{len}(\gamma)} \int_{v \in \gamma} (v_i - v) \, ds = H(v_i) n_i
\]
Laplacian Matrix

- Coefficient of each vertex in computation of Laplacian at every other vertex
Weighting Schemes

- Ignore geometry
  \[ \delta_{\text{umbrella}} : w_{ij} = 1 \]

- Integrate over circle around vertex
  \[ \delta_{\text{mean value}} : w_{ij} = \tan \phi_{ij}/2 + \tan \phi_{ij+1}/2 \]

- Integrate over Voronoi region of vertex
  \[ \delta_{\text{cotangent}} : w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij} \]
Laplacian Mesh Representation

- Vertex positions are represented by Laplacian coordinates \((\delta_x, \delta_y, \delta_z)\)

\[
\delta_i = \sum_{j \in N(i)} w_{ij} (v_i - v_j)
\]

\[
\begin{align*}
v_x &= \delta_x \\
v_y &= \delta_y \\
v_z &= \delta_z
\end{align*}
\]
Basic Properties

- $\text{rank}(L) = n - c$  \( (n - 1 \text{ for connected meshes}) \)
- Can reconstruct geometry from $\delta$ up to translation
  - Add constraint on one vertex for unique solution
Reconstruction

• Constrain additional vertices: overdetermined system

\[
\arg\min_x \left( \|Lx - \delta_x\|^2 + \sum_{k=1}^{n_c} \|x_k - c_k\|^2 \right)
\]

• Cool underlying idea: shape defined as minimizer of an objective function
So Far…

- Laplacian coordinates $\delta$
  - Local representation
  - Translation-invariant

- Linear transition from $\delta$ to $xyz$
  - can constrain more than 1 vertex
  - least-squares solution
Editing Using Laplacian Coordinates

The editing process from the user’s point of view:

1. Set ROI, anchors, and a *handle* vertex
2. Move the handle, interactively see effect on mesh
Editing Using Laplacian Coordinates

Behind the scenes…

- ROI defines vertices that are included in the solve
- Constraints at anchors: responsible for smooth transition of the edited part to the rest of the mesh
  - Increasing weight with distance away from handle
- Precomputation enables interactivity

\[
\begin{align*}
Ax &= b \\
A^TAx &= A^Tb \\
x &= (A^TA)^{-1}A^Tb
\end{align*}
\]
Mesh Editing Example
Mesh Editing Example

Original  Regular Laplacian editing  Solve for transformations
Mesh Editing Example
What Else Can We Do with It?

• By modifying Laplacians or positional constraints, can achieve a variety of other effects
Detail Transfer

- “Peel” the detail off one surface and transfer to another
Detail Transfer
Detail Transfer
Mixing Laplacians

- Take weighted average of $\delta_i$ and $\delta'_i$
Mesh Transplanting

- Geometrical stitching via Laplacian mixing
Mesh Transplanting

- Details gradually change in the transition area
Mesh Transplanting

• Details gradually change in the transition area
Feature Preserving Smoothing

- Weighted positional and smoothing constraints

\[ \begin{align*}
V & = \begin{bmatrix}
L & 0
\end{bmatrix}
\end{align*} \]
Feature Preserving Smoothing

- Weighted positional and smoothing constraints
Parameterization

- Use zero Laplacians.

\[ \mathbf{L} \mathbf{V} = \mathbf{0} \]

In 2D:
Texture Mapping
Texture Mapping

[Piponi 2000]