Re-Meshing Surfaces

- Given input mesh, generate new mesh which is "better"
  - Element sizing
  - Element shape
- BUT is near (geometrically) to original surface
Hausdorff Metric

- Given two sets (surfaces) $P$ and $Q$
  \[ H_P(Q) = \max_{p} \min_{q} ||p,q|| \]
  \[ H(P,Q) = \max(H_P(P), H_Q(Q)) \]
- Point to point
- On mesh approximate by
  - measuring vertex to surface distance
  - measuring vertex to vertex distance
- Expensive to compute
  - Public Domain Code: Metro

How to (re)mesh surfaces?

- Can we apply Delaunay triangulation?
  - What is Delaunay criterion on surface?
    - Option 1: Use sphere instead of circle
      - Works for volumetric meshes (tets)
    - Option 2: Use pairwise test only
      - Theoretical Delaunay properties do not hold
  - Boundary recovery = Approximation quality
Approaches

- Mesh adaptation/Local Remeshing
  - Modify existing mesh using sequence of local operations
    - Evaluate approximation quality at every step

- Reduction to 2D/Global Remeshing
  - Segment surface into parameterizable pieces
  - Parameterize in 2D
  - Mesh in 2D (Delaunay)
  - Project back

Mesh adaptation

- Store original for approximation evaluation
- Connectivity modification
  - Edge flips
  - Refinement/Coarsening
- Geometry modification - mesh smoothing
- Typical Sequence:
  - Refine/Coarsen to satisfy sizing
  - Smooth mesh (sizing + quality)
  - Perform flips after every other operation
  - Repeat entire sequence several times
Mesh adaptation: Edge Flip

- Given 2 triangles flip one diagonal if longer than the other
- 3D equivalent of Delaunay test in 2D (why?)
- Test impact on approximation (why?)
  - Approximate Hausdorff metric
    - Normal error
    - Smoothness
  - Test self-intersection
    - Distance based
    - Very expensive test - often ignored

Normal Error

- Based on normals only
- Defined for a face
- Normals $N_j$, $N_2$, $N_3$ - from original mesh
  - Store original locations for each vertex
- $N_f$ - current face normal
- Distance:
  \[
  E_{gap}(f) = \max_{i \in \{1,2,3\}} \langle N_{f,i}, N_{f,i+1} \rangle < \cos \theta_{gap}
  \]
- Why?
Smoothness

\[ E_{\text{smooth}}(f) = \max_{i \in \{1,2,3\}} \langle N_I, N_i \rangle \leq \cos \theta_{\text{smooth}} \]

Example

Before (avg min 30)

After (avg min 33)
Mesh Adaptation: Refinement

- Add vertices - reach desired sizing or element count
- Strategy:
  - Split long edges - insert mid-points
- Vertex positioning
  - Project to original mesh
- Hard to achieve good spacing
  - Improve by smoothing

Example

Before (avg min 33)

Second round of flips (avg min 37)
### Projection to Original Mesh

- **Nearest point**
  - Requires search
    - Find original face closest to (estimated) new vertex
  - Expensive
  - Unlimited Hausdorff error

- **Local parameterization**
  - Compute new location in terms of barycentric coordinates of new face vertices
  - Locally parameterize old mesh to get corresponding location
    - Use sophisticated data structures for efficiency
  - Better approximation

### Local Parameterization

**Input:**
- New mesh face: 3 vertices
  - known locations (face+ coords) on original mesh
- New vertex: barycentric coordinates on new face
- Compute small patch of old mesh containing all 3 vertices
- Parameterize patch in 2D
- Compute new location on parameterized patch
- Project to 3D
Mesh Adaptation: Coarsening

- Similar to simplification
- Operations:
  - Vertex removal
  - Edge collapse
    - Project new vertex to original surface as in refinement
- Approximation Error
  - Quadrics
  - Normal based

Mesh Adaptation: Smoothing

- Move vertices ON surface to improve sizing/quality
- Moving One Vertex:
  - Compute vertex location as function of neighbors in new mesh
    - E.g. convex combination
    - Use local parameterization
  - Project to original mesh
  - Check approximation error
    - If too large, keep previous location
Local Parameterization

- Project vertex + neighbors to current normal plane
- Relocate vertex in plane
- Find new triangle in which vertex is located
- Compute barycentric coordinates in this triangle
- Use for placement on original mesh

Example

Before (avg min 30)

Smoothing + Flips (avg min 45)
Smoothing: Centroidal Voronoi Diagram

- Relocate vertices (smoothing) to control sizing (sampling)
- Lloyd algorithm on surface mesh
  - On 2D umbrella compute VD of vertex + neighbors
  - Place vertex at center of mass of it’s cell
  - Repeat

Michelangelo's David

Original: 350k faces
Remesh: 100k faces
Mesh Adaptation

- Modify existing mesh using sequence of local operations
  - Fast
  - Simple to implement
    - Depending on choice of local operations
  - Hard to find GOOD spacing of vertices
    - WCVD does the trick but at a cost...
  - Heuristic
    - How many iterations of each operation to do?