600.657: Mesh Processing

Chapter 7
Variational Shape Approximation

Goal:
Compute lower-complexity polygonization(s) of a surface capturing the detail(s): $M_0 \rightarrow M_1 \rightarrow ...$

[Cohen-Steiner et al., 2004] [Marinov and Kobbelt, 2005]
Variational Shape Approximation

**Goal:**
Compute lower-complexity polygonization(s) of a surface capturing the detail(s): $M_0 \rightarrow M_1 \rightarrow \ldots$

- Capture detail $\rightarrow$ polygons align with anisotropy.
- Simple base models $\rightarrow$ topology is removed.
Variational Shape Approximation

Approach:
Partition $M$ into $k$ regions $R=\{R_1,...,R_k\}$ which are a best fit to $k$ planar proxies $P=\{P_1,...,P_k\}$:

$$E(R, P) = \sum_{i=1}^{k} \int_{R_i} d^2(q, P_i) dq$$

where the planar proxy $P_i$ can be represented by a point $x_i$ on the plane and a unit-normal $n_i$. 

Variational Shape Approximation

**Approach:**
Partition $M$ into $k$ regions $R=\{R_1,...,R_k\}$ which are a best fit to $k$ planar proxies $P=\{P_1,...,P_k\}$:

$$E(R,P) = \sum_{i=1}^{k} \int_{R_i} d^2(q,P_i) dq$$

where the planar proxy $P_i$ can be represented by a point $x_i$ on the plane and a unit-normal $n_i$.

Need to simultaneously solve for the partition regions $R$ and the proxies $P$. 
Variational Shape Approximation

Approach:
The distance function $d$ can be:

- The $L^2$ distance: $d^2(q, P_i) = \min_{p \in P_i} \|q - p\|^2 = \langle q - x_i, n_i \rangle^2$
- The $L^{2,1}$ distance: $d^2(q, P_i) = \|n(q) - n_i\|^2$

[Coherent-Steiner et al., 2004]
Variational Shape Approximation

Implementation:

- [Cohen-Steiner et al., 2004]: Use a Lloyd’s algorithm type approach – iterate between growing regions and fitting proxies.
- [Marinov and Kobbelt, 2005]: Use a decimation type approach, merging triangles into regions, and using the regions to define the proxies.
Initialization:

– Choose $k$ triangles at random, and set $P_i$ to be the plane going through triangle $t_i$. 

[Cohen-Steiner et al., 2004]
 Initialization:
   – Choose $k$ triangles at random, and set $P_i$ to be the plane going through triangle $t_i$.

 Iteratively:
   – Grow the regions $R_i$.
   – Fit the proxies $P_i$.
   – Seed new triangles by choosing the triangle $t_i \in R_i$ that is closest to $P_i$. 

[Cohen-Steiner et al., 2004]
[Cohen-Steiner et al., 2004]

Growing $R_i$:

- For each assigned triangle $t$:
  - For each unassigned neighbor $t'$:
    - Compute the fit of $t'$ to the proxy of $t$.
  - Assign the best fitting $t'$.
Fitting $P_i$:
Given a region $R_i$, we would like to solve for the position $x_i$ and normal $n_i$ minimizing:

$$E_{R_i}(x_i,n_i) = \int_{R_i} d^2(q,P_i)dq$$
[Cohen-Steiner et al., 2004]

**Fitting** $P_i$:

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$
Fitting $P_i$:

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$

Taking the derivative w.r.t. $x_i$, we get:

$$\frac{\partial E_{R_i}}{\partial x_i} = -2\int_{R_i} \langle q - x_i, n_i \rangle n_i dq = -2\left< \int_{R_i} q dq - |R_i| x_i, n_i \right>n_i$$
[Cohen-Steiner et al., 2004]

**Fitting $P_i$:**

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$

Taking the derivative w.r.t. $x_i$, we get:

$$\frac{\partial E_{R_i}}{\partial x_i} = -2 \int_{R_i} \langle q - x_i, n_i \rangle n_i dq = -2 \left( \int_{R_i} q dq - |R_i| x_i, n_i \right) n_i$$

So the distance is minimized when $x_i$ is the average of the points in $R_i$:

$$\frac{\partial E_{R_i}}{\partial x_i} = 0 \iff \int_{R_i} q dq - |R_i| x_i = 0 \iff x_i = \frac{1}{|R_i|} \int_{R_i} q dq$$
Fitting $P_i$:

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$

Setting $x_i$ to the average, we get:

$$E_{R_i}(x_i, n_i) = n_i \left( \int_{R_i} (q - x_i)(q - x_i)' dq \right) n_i$$
Fitting $P_i$:

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$

Setting $x_i$ to the average, we get:

$$E_{R_i}(x_i, n_i) = n_i^i \left( \int_{R_i} (q - x_i)(q - x_i)' dq \right) n_i$$

Thus, the unit-vector $n_i$ minimizing the distance is the smallest eigenvector of:

$$C_{R_i} = \int_{R_i} (q - x_i)(q - x_i)' dq$$
Fitting $P_i$:

If $d^2$ is the $L^2$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \langle q - x_i, n_i \rangle^2 dq$$

Setting $x_i$ to the average, we get:

$$E_{R_i}(x_i, n_i) = n_i^t \left( \int_{R_i} (q - x_i)(q - x_i)' dq \right) n_i$$

Thus, the unit-vector $n_i$ minimizing the distance is the smallest eigenvector of:

$$C_{R_i} = \int_{R_i} (q - x_i)(q - x_i)' dq$$

Note that $C_{R_i}$ is the covariance matrix of $R_i$. 
Fitting $P_i$:

If $d^2$ is the $L^{2,1}$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \| n(q) - n_i \|^2 dq$$
Fitting $P_i$:

If $d^2$ is the $L^{2,1}$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \|n(q) - n_i\|^2 dq$$

Since $R_i$ is made up of triangles, the normal of a point is the normal of its triangle.
Fitting $P_i$:

If $d^2$ is the $L^{2,1}$ distance:

$$E_{R_i}(x_i, n_i) = \int_{R_i} \left\| n(q) - n_i \right\|^2 dq$$

Since $R_i$ is made up of triangles, the normal of a point is the normal of its triangle. So, the normal $n_i$ minimizing the distance is the average of the triangle’s normals:

$$n_i = \frac{\sum_{t \in R_i} N(t)t}{\left\| \sum_{t \in R_i} N(t)t \right\|}$$
In Sum:
Iterating this process, we get a set of connected regions $R_i$ that partition the original surface.
In Sum:
Iterating this process, we get a set of connected regions $R_i$ that partition the original surface.

- Identify corners
  - >2 patches in interior
  - >1 patch on boundary
**In Sum:**
Iterating this process, we get a set of connected regions $R_i$ that partition the original surface.

- **Identify corners**
  - $>2$ patches in interior
  - $>1$ patch on boundary

- **Connect corners**
  - May need to divide edges
In Sum:

Iterating this process, we get a set of connected regions $R_i$ that partition the original surface.

- Identify corners
  - >2 patches in interior
  - >1 patch on boundary

- Connect corners
  - May need to divide edges

- Triangulate polygons
  - Using “discrete CDT”
Advantages:
- Fast
- Can be made hierarchical*

Disadvantage:
- Convergence?
- Not guaranteed to be manifold

*Can grow regions made up of polygons, not just triangles
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them.
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.
- Each triangle starts in its own patch
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.

- Each triangle starts in its own patch
- Edges are successively removed
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.

– Each triangle starts in its own patch
– Edges are successively removed
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.

- Each triangle starts in its own patch
- Edges are successively removed
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.

- Each triangle starts in its own patch
- Edges are successively removed
Approach:
Instead of using a Lloyd’s algorithm approach to grow regions, use face-merge to collapse them. Two regions can be collapsed into each other iff. their projection onto their proxy is injective.

– Each triangle starts in its own patch
– Edges are successively removed
– Valence-2 vertices are removed
Invariants:

At each step of the iteration:

– The region $R_i$ tracks its:
  • Boundary: $b(R_i)$
  • Area: $a(R_i)$
  • Centroid: $x(R_i)$
  • Normal: $n(R_i)$

– $R_i$ maps injectively to its proxy $(x(R_i), n(R_i))$
[Marinov and Kobbelt, 2005]

**Invariants:**

At each step of the iteration:

– The region $R_i$ tracks its:
  
  • Boundary: $b(R_i)$
  • Area: $a(R_i)$
  • Centroid: $x(R_i)$
  • Normal: $n(R_i)$

– $R_i$ maps injectively to its proxy $(x(R_i), n(R_i))$

**Note:** The area of $R_i$ is not tracked!
Injectivity:
Since only the boundary of a region $R_i$ is tracked, injectivity is determined by testing that the projection of the boundary is injective:

- That the boundary curves do not self-intersect
- That the exterior boundary contains interior boundaries.
- That the interior boundary curves do not nest.
[Marinov and Kobbelt, 2005]

Initialization:

– Set each triangle to be its own region.
[Marinov and Kobbelt, 2005]

**Initialization:**
- Set each triangle to be its own region.

**Merging:**

When merging adjacent regions $R \leftarrow (R_i \cup R_j)$:

- $b(R_i \cup R_j) = b(R_i) \cup a(R_j) - b(R_i) \cap a(R_j)$
- $a(R_i \cup R_j) = a(R_i) + a(R_j)$
- $x(R_i \cup R_j) = \frac{x(R_i) \cdot a(R_i) + x(R_j) \cdot a(R_j)}{a(R_i) + a(R_j)}$
- $n(R_i \cup R_j) = \frac{n(R_i) \cdot a(R_i) + n(R_j) \cdot a(R_j)}{|n(R_i) \cdot a(R_i) + n(R_j) \cdot a(R_j)|}$
Initialization:
- Set each triangle to be its own region.

Iteratively:
- Evaluate the cost of every feasible merge
- Perform the lowest-cost merge
- Repeat
Cost of a Merge:

In order to evaluate the merge, we need to associate a surface with each region $R_i$. 

[Marinov and Kobbelt, 2005]
Cost of a Merge:
In order to evaluate the merge, we need to associate a surface with each region $R_i$. We do this by:

– Projecting the boundary of $R_i$ onto the proxy
– Computing the CDT of the boundary
– Raising the triangulation to 3D.
Cost of a Merge:

If $d^2$ is the $L^2$ distance, merging $R_i$ and $R_j$ costs:

$$E(R_i, R_j) = \sum_{t \in R_i \cup R_j} \int \left( p - x(R_i \cup R_j), n(R_i \cup R_j) \right)^2 dp$$
Cost of a Merge:

If $d^2$ is the $L^2$ distance, merging $R_i$ and $R_j$ costs:

$$E(R_i, R_j) = \sum_{t \in R_i \cup R_j} \int \left( p - x(R_i \cup R_j), n(R_i \cup R_j) \right)^2 dp$$

If $d^2$ is the $L^{2,1}$ distance, merging $R_i$ and $R_j$ costs:

$$E(R_i, R_j) = a(R_i) \left\| n(R_i) - n(R_i \cup R_j) \right\|^2 + a(R_j) \left\| n(R_j) - n(R_i \cup R_j) \right\|^2$$
Advantages:

– Partition is hierarchical
– Patches are well-behaved

Disadvantage:

– Use of CDT can make it slow.*
– Errors are not measured w.r.t. original triangles

*May be able to get by without using a CDT