Variational Shape Approximation

By David Cohen-Steiner, Pierre Alliez and Mathieu Desbrun

Math 6645 Final Presentation

Xia Wu
Outline

- Background and related works
- Shape proxies and error metrics
- Algorithm for shape approximation
- Applications and results
Background & Related Works
In computer graphics, a surface is usually expressed by triangles.

Approximation theory has provided robust results on approximating functions, but they are hard to extend to surfaces due to the randomness and no mapping.

The problem of finding the best geometric approximation of a 3D surface is NP-hard. Thus the computational time is considered significantly.

Figure 1: Example of shape approximation
Related Works

- **Partitioning** using greedy algorithms
  - May result in suboptimal meshes

- Cast mesh simplification to **global optimization** problems
  - May generate irregular meshes

- Take use of **anisotropy** (physical property that having different values when measured in different directions)
  - May increase the computational time
Shape Proxies & Error Metrics
Shape Proxies

Partition surface $\mathcal{S}$ into $k$ regions: $\{\mathcal{R}_i\}_{i=1}^k$

Associate each region $\mathcal{R}_i$ with a Proxy $P_i$

Proxy $P_i$ is expressed by an “average” point $X_i$ and normal $N_i$: $P_i = (X_i, N_i)$
Shape Proxies (cont.)

A set $P$ of proxies $\{P_i\}_{i=1}^k$ forms the approximation of the whole geometry

$X_i$ - Barycenter of region $R_i$

$N_i$ - Area-weighted average of the normals
Error Metrics

The commonly used error metrics in computer graphics are:

\[ \mathcal{L}^p(X, Y) = \left( \frac{1}{|X|} \int \int_{x \in X} \|d(x, Y)\|^p \, dx \right)^{\frac{1}{p}} \]

\[ d(x, Y) = \inf_{y \in Y} \|x - y\| \]

- \( X \) : input surface
- \( Y \) : approximating surface
- \( |.| \) : surface area
- \( \|.| \) : Euclidean distance

Hausdorff distance:

\[ \mathcal{H}(X, Y) = \max_{x \in X} d(x, Y) \]
Error Metrics (cont.)

Extension of $\ell^2$ norm in this paper:

$$\ell^2(\mathcal{R}_i, P_i) = \iiint_{x \in \mathcal{R}_i} \|x - \Pi_i(x)\|^2 \, dx$$

$\Pi(\cdot)$ - Orthogonal projection of argument on the “proxy” plane going through $X_i$ and normal to $N_i$

That is, this error metric is checking how “far” is the proxy plane from all the plane associated with an unique triangle in the corresponding region, i.e. “MSE”
Error Metrics (cont.)

The previous error metric matches geometry via approximating the position of the object in the space. They found that, however, our visual system is more sensitive to changes in normal other than position. Therefore, new error metric based on previous measure of the normal filed is introduced:

New error metric:

\[ \mathcal{L}^{2,1}(R_i, P_i) = \int \int_{x \in R_i} \| n(x) - n_i \|^2 dx \]

\( n \): normal

Figure 3: Example with new error metric
Comparison

Pros of new error metric:

- Can capture the anisotropy of the surface better.
- The new metric generates equal or better results visually according authors’ tests.
- To find the best normal proxy, just need to average the normals over the associated regions without computing covariance matrix. Thus it saves great amount of computational time.

Figure 4: Comparison of approximation results between two error metrics
Optimal Shape Proxies

Together with the regions and proxies defined previously, we now define the total distortion over all the partitions:

\[ E(\mathcal{R}, P) = \sum_{i=1}^{k} E(\mathcal{R}_i, P_i) \]

Optimal shape proxies: a set \( P \) of proxies \( P_i \) associated to the regions \( \mathcal{R}_i \) of a partition \( \mathcal{R} \) of \( S \) that minimizes the total distortion above.
Algorithms of Shape Approximation
K-Means Clustering

1) Define k random centers (then we have k regions).
2) Assign each point to its nearest center (and the associated region).
3) Update center to be the barycenter (centroid) of each region.
4) Repeat 2) and 3) until stopping criteria have been met.

Stopping criteria: minimize the following cost function to some predefined tolerance:

\[ E = \sum_{i=1}^{k} \sum_{x_j \in R_i} \| x_j - c_i \|^2 \]

\[ \{ c_i \}_{i=1}^{k} : \text{Barycenter of region} \]
\[ \{ x_j \}_{j=1}^{N} : \text{Data points in one region} \]
K-Proxy Clustering

1) Initial $k$ random proxies by picking $k$ random triangles (then we have $k$ planes and their associated barycenters and normals).

2) Using priority queue, assign each triangle to a proxy that connected to it in terms of the minimum distortion error until the triangles are exhausted (BFS).

3) Update each proxy and its associated barycenter and normal.

4) Repeat 2) and 3) until stopping criteria have been met.

Stopping criteria: minimize the total distortion, which is defined in previous slides with respect to one of the error metrics, to some predefined tolerance:

$$E(\mathcal{R}, P) = \sum_{i=1}^{k} E(\mathcal{R}_i, P_i)$$

$$\mathcal{L}^2(\mathcal{R}_i, P_i) = \int_{x \in \mathcal{R}_i} \| x - \prod_i(x) \|^2 dx$$

Or

$$\mathcal{L}^{2,1}(\mathcal{R}_i, P_i) = \int_{x \in \mathcal{R}_i} \| n(x) - n_i \|^2 dx$$
Improvements

- **Number of proxies** can be chosen by users
- Implement **region teleportation** when stuck in local optima, for example, flat region.
- Special **farthest-point initialization** for non-smooth objects
- While global convergence cannot be guaranteed, **convergence** over new defined norm can be guaranteed on convex objects.
- Allow users to add “importance” or weight to some specific regions
Applications & Results
Applications and Results

Figure 5: Results of two iterations after user interactively added a proxy.

Figure 6: Convergence of authors’ algorithm
Applications and Results (cont.)

Figure 7: Results of $\mathcal{L}^2,1$-approximation and QEM

Figure 8: Comparison of the Hausdorff error of QEM and authors’ $\mathcal{L}^2,1$-technique


Thank you!

Any questions?