Superfaces: Polyhedral Mesh Simplification with Bounded Error

Alan D. Kalvin
Russell H. Taylor
Problems with Polyhedra

- Common 3D image to surface boundary reconstruction algorithms produce many small faces
- Shapes are complex
- Voxel-based methods cannot span > 1 voxel
- Contour tiling methods cannot span > 2 slices
Problems with Polyhedra

- Common 3D image to surface boundary reconstruction algorithms produce many small faces
- Shapes are complex
- Voxel-based methods cannot span > 1 voxel
- Contour tiling methods cannot span > 2 slices
Superfaces Algorithm: Summary

Automatic simplification of complex polyhedral models with bounded error:

- Applied to biomedical models derived from CT
- Useful on other models with similar characteristics
- Typical performance on 350K triangle skull model and 1 voxel diameter error bound:
  - 4:1 reduction in triangle count
  - 7:1 reduction data structure size
  - 6 minutes on (slow) RS/6000
Superfaces Algorithm: Summary

Automatic simplification of complex polyhedral models with bounded error:

– Applied to biomedical models derived from CT
– Useful on other models with similar characteristics
– Typical performance on 350K triangle skull model and 0.5 pixel units error bound:
  • 3:1 to 6:1 reduction in triangle count
  • Mean approx. error 0.05-0.09 pixel units
  • Run time 8.5 to 9.5 minutes on (slow) RS/6000
11 Original skull model (349,792 triangles).
12 Simplified skull (a) mesh and (b) color-coded approximation errors in pixel units: $\varepsilon = 0.5$ (36.60 percent of original triangles).
Simplified skull (a) mesh and (b) color-coded approximation errors in pixel units—with aggressive border straightening: $\epsilon = 0.5$ (15.58 percent of original triangles).
Algorithm Properties

- Fast, “greedy” method
- Preserves geometric error bound
- Preserves topology
- Simplified model is imbedded in original
- Applicable to any polyhedral model
- No *a priori* knowledge of surface required
Related Work

- Schmidt, Barsky, Du (1986)
  - top-down refinement of surface of bicubic patches
  - for objects $z=f(x,y)$
- Kalvin (1991)
  - adaptive merging of redundant faces
- Schroeder, Zarge, Lorensen (1992)
  - “triangle decimation” to reduce size by given percentage
- Turk (1992)
  - retiling surface by triangulating new set of vertices
- Rossignac & Borel (1993)
  - multi-resolution 3D approximations
Related work

• Cutting, et. al (1991)
  – Registration to anatomical atlas
• Bloomenthal (1988)
• Hall and Warren (1990)
• Ning and Hasselink (1991)
• Gueziec (1996)
• Cohen (1997?)

• many more
Algorithm Outline

• Phase 1: Merge faces into superfaces
  – Greedy, bottom-up algorithm
  – Runs in \( O(n) \) time, where \( n \) = number of faces

• Phase 2: Straighten borders
  – Create “superedges”
  – Several variations with different degrees of aggressiveness

• Phase 3: Pick triangulation points
  – Usually, triangulation is not done explicitly
Algorithm outline
Phase 1: Greedy Merging

1. Pick a seed face to start a new superface
   - Options include random choice, pincushion search, etc.

2. Keep adding adjacent faces to the superface as long as can find a feasible approximating plane & meet some other technicalities

3. Repeat steps 1 & 2 as long as there are faces not assigned to superfaces


Phase 1: Quasi-planar merging

Consider the approximating plane $P$ of a superface:

$$P = \{(x, y, z) \mid ax + by + z = d\}$$

in some local coordinate system of the face. The parameters $(a, b, d)$ thus represent $P$.

Note the duality: $(a, b, d)$ constrains $(x, y, z)$, but $(x, y, z)$ also constrains $(a, b, d)$.

In general, $(a, b, d)$ will obey the bounded approximation constraints

$$-\varepsilon - z \leq ax + by - d \leq \varepsilon - z$$

and some other (linear) constraints to be discussed later.
Approximating Plane
Set of feasible approximating planes

The set of feasible planes is described by a polytope

\[ E = \{ (a, b, d) | C \cdot (a, b, d)^T \leq g \} \]
Set of feasible approximating planes

Conservatively approximate $E$ by an ellipsoid
Ellipsoidal approximation

\[ E^* = \|k(k - k_0)^T M(k - k_0) \| \leq 1 \]

\[ M = Q^T \beta_1^2 \beta_2^2 \ldots \beta_n^2 \]

\[ E = \|kC \cdot k \| \leq g \]
Growing a Superface

1. Select $f_b$ face on current perimeter

   $\Rightarrow f_b$ generates new linear constraints \{\(C_j\}\}

2. Compute $\Sigma' =$ linear-time adjustment of ellipsoid $\Sigma$ based on \{\(C_j\}\}
Growing a Superface

3. if $\Sigma' \neq \{\}$ then $f_b$ satisfies merging criteria
   - merge $f_b$ into superface
   - $\Sigma \leftarrow \Sigma'$

4. Iterate above until:
   - no more acceptable faces to merge
   - bad aspect ratio
Merging Rules

1. Planarity rule:
   - All vertices of \( f_b \) must be within bounded distance of approximating plane \( p \)
   - 2 constraints: \( \|(a, b, 1, -d). (v_x, v_y, v_z, 1)\| \leq \Delta_{\text{max}} \)

2. Face-axis rule:
   - orientation \( f_b \approx \) orientation \( p \)
   - constraint: \( an_x + bn_y \geq \cos(\theta_{\text{max}}) - n_z \)
   - \( (n_x, n_y, n_z) = \) outward-facing normal of \( f_b \)

3. No-foldover rule:
   - \( f_b \) cannot “tuck-under” superface \( F \)
   - \( \forall v \in f_b, v \) outside \( F_p \) (projection of perim \( F \) into \( p \))
   - constraint: \( aK_1 + bK_2 \leq K_3 \)
Gerrymandering check

Optional constraint

- to prevent "irregular" shaped surfaces
- stop growing if Irreg($F$) > $\text{Irreg}_{\text{max}}$
- simple estimate: $\text{Irreg}(F) = \text{perim}^2/\text{area}$

Polyhedra from Alligator algorithm

- perim $\approx$ no. edges
- area $\approx$ no. faces
- Irreg($F$) needs 3 floating point ops.
Phase 1 output
Phase 2 Strategy

1. Replace edges between adjacent superfaces with single superedge
2. Recursively split the big superedge into smaller edges until every boundary vertex in any edge merged into superedge is within a bounded distance of one of adjacent superfaces
3. Repeat until done

Alternative: Merge aggressively and check all subsumed vertices. Only split if boundedness condition is violated.
Phase 2

Figure 5: Superface borders (a) before straightening, (b) after edge merging, (c) after edge splitting.
Phase 3: Triangulation

1. Project superface perimeter into the nominal approximating plane

2. Decompose 2D polygon into star polygons

3. (Implicitly) triangulate star polygons
Star Polygon Decomposition

Method 1 (fast, try first):

1. Determine conservative approximation for the polygon kernel of projection of superface onto its approximating plane.
2. Pick point in polytope.

Method 2 (if that fails)

1. Decompose superface into monotone polygons.
2. Determine triangulation point for each monotone polygon.
Star Polygon Decomposition

• Avis & Toussaint (1981) - $O(n \log n)$
  – efficient
  – does not attempt to limit number of star polygons
  – does not handle polygons with holes
• Keil (1985) - $O(n^5k^2\log n)$
  – minimizes number of star polygons
  – does not handle polygons with holes
• What we did - $O(n^2)$
  – Attempts to limit number of star polygons
  – Can handle polygons with holes
  – First step is decomposition into monotone polygons
  – Second step is conversion of monotone polygons to star polygons
Monotone polygons & Star Decomp.

Figure 6: Monotone polygon $P$.

Figure 7: Creating internal diagonal $[B, C]$. 
Results for skull

<table>
<thead>
<tr>
<th>error bound</th>
<th>approx. error</th>
<th>triangles</th>
<th>running time (m:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>max.</td>
<td>count</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0544</td>
<td>0.4723</td>
<td>128,040</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1289</td>
<td>0.9231</td>
<td>78,002</td>
</tr>
<tr>
<td>1.5</td>
<td>0.2017</td>
<td>1.4387</td>
<td>50,442</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2559</td>
<td>1.8690</td>
<td>37,438</td>
</tr>
<tr>
<td>3.0</td>
<td>0.3088</td>
<td>2.6119</td>
<td>28,388</td>
</tr>
<tr>
<td>4.0</td>
<td>0.3358</td>
<td>2.7684</td>
<td>24,170</td>
</tr>
</tbody>
</table>

Table 1: Results of simplifying the skull mesh of 349,792 triangles.

<table>
<thead>
<tr>
<th>error bound</th>
<th>approx. error</th>
<th>triangles</th>
<th>running time (m:ss)</th>
<th>vertices above e limit</th>
<th>superfaces to adjust / total superfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>max.</td>
<td>count</td>
<td>% of original</td>
<td>time</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0947</td>
<td>1.4240</td>
<td>53,790</td>
<td>15.38</td>
<td>8:49</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2187</td>
<td>1.3973</td>
<td>23,704</td>
<td>6.78</td>
<td>7:12</td>
</tr>
<tr>
<td>1.5</td>
<td>0.3402</td>
<td>2.3713</td>
<td>15,470</td>
<td>4.42</td>
<td>6:28</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4523</td>
<td>5.8117</td>
<td>11,994</td>
<td>3.43</td>
<td>6:20</td>
</tr>
<tr>
<td>3.0</td>
<td>0.5984</td>
<td>3.3584</td>
<td>9,820</td>
<td>2.81</td>
<td>6:03</td>
</tr>
<tr>
<td>4.0</td>
<td>0.6714</td>
<td>3.6544</td>
<td>8,934</td>
<td>2.55</td>
<td>5:52</td>
</tr>
</tbody>
</table>

Table 2: Results of simplifying the skull mesh of 349,792 triangles – with aggressive border straightening.)
16 Original femur model (179,916 triangles).
17 Simplified femur (a) mesh and (b) color-coded approximation errors in pixel units: $\epsilon = 4.0$ (12.14 percent of original triangles).
18 Simplified femur (a) mesh and (b) color-coded approximation errors in pixel units—with aggressive border straightening: \( \varepsilon = 4.0 \) (4.41 percent of original triangles).
Results for femur

<table>
<thead>
<tr>
<th>error bound $\epsilon$</th>
<th>approx. error mean</th>
<th>approx. error max.</th>
<th>triangles count</th>
<th>% of original</th>
<th>running time (m:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.0378</td>
<td>0.4826</td>
<td>97,010</td>
<td>53.92</td>
<td>4:54</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1060</td>
<td>0.8821</td>
<td>66,318</td>
<td>36.86</td>
<td>3:31</td>
</tr>
<tr>
<td>1.5</td>
<td>0.1708</td>
<td>1.3265</td>
<td>46,748</td>
<td>25.98</td>
<td>3:10</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2263</td>
<td>1.7860</td>
<td>36,018</td>
<td>20.02</td>
<td>2:54</td>
</tr>
<tr>
<td>2.5</td>
<td>0.2745</td>
<td>2.2530</td>
<td>30,766</td>
<td>17.10</td>
<td>2:45</td>
</tr>
<tr>
<td>3.0</td>
<td>0.3196</td>
<td>2.7049</td>
<td>26,832</td>
<td>14.91</td>
<td>2:36</td>
</tr>
<tr>
<td>4.0</td>
<td>0.3921</td>
<td>3.5481</td>
<td>21,840</td>
<td>12.14</td>
<td>2:28</td>
</tr>
</tbody>
</table>

Table 3: Results of simplifying the femur mesh of 179,916 triangles.

<table>
<thead>
<tr>
<th>error bound $\epsilon$</th>
<th>approx. error mean</th>
<th>approx. error max.</th>
<th>triangles count</th>
<th>% of original</th>
<th>running time (m:ss)</th>
<th>vertices above $\epsilon$ limit</th>
<th>superfaces to adjust / total superfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.0733</td>
<td>0.9509</td>
<td>56,294</td>
<td>51.29</td>
<td>4:22</td>
<td>12</td>
<td>10 / 18,792</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1797</td>
<td>2.1814</td>
<td>28,216</td>
<td>15.68</td>
<td>3:07</td>
<td>17</td>
<td>8 / 7,599</td>
</tr>
<tr>
<td>1.5</td>
<td>0.2778</td>
<td>1.6823</td>
<td>19,348</td>
<td>10.75</td>
<td>2:46</td>
<td>2</td>
<td>2 / 4,908</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4000</td>
<td>2.3609</td>
<td>12,762</td>
<td>7.09</td>
<td>2:33</td>
<td>42</td>
<td>4 / 3,016</td>
</tr>
<tr>
<td>3.0</td>
<td>0.5516</td>
<td>3.7760</td>
<td>9,046</td>
<td>5.03</td>
<td>2:21</td>
<td>59</td>
<td>2 / 2,152</td>
</tr>
<tr>
<td>4.0</td>
<td>0.6797</td>
<td>4.1972</td>
<td>7,942</td>
<td>4.41</td>
<td>2:17</td>
<td>4</td>
<td>1 / 1,885</td>
</tr>
</tbody>
</table>

Table 4: Results of simplifying the femur mesh of 179,916 triangles (with aggressive border straightening).
19 Map of topographic data of the earth.

20 Simplified map of topographic data of the earth: \( r = 32.0 \) meters.
Some published comparisons

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>triangles</th>
<th>running times (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original mesh</td>
<td>reduced mesh</td>
</tr>
<tr>
<td><strong>Superfaces</strong></td>
<td>30,876</td>
<td>2,038</td>
</tr>
<tr>
<td>IBM RS/6000 (model 550)</td>
<td>179,916</td>
<td>21,840</td>
</tr>
<tr>
<td></td>
<td>349,792</td>
<td>24,170</td>
</tr>
<tr>
<td><strong>Triangle Decimation [13]</strong></td>
<td>38,394</td>
<td>17,799</td>
</tr>
<tr>
<td>SGI Onyx Reality Engine</td>
<td>186,630</td>
<td>71,485</td>
</tr>
<tr>
<td>2-processor model</td>
<td>334,643</td>
<td>84,342</td>
</tr>
<tr>
<td></td>
<td>1,049,476</td>
<td>29,507</td>
</tr>
<tr>
<td><strong>Geometric Optimization [10]</strong></td>
<td>315,812</td>
<td>295,636</td>
</tr>
<tr>
<td>(hardware not specified)</td>
<td>1,019,373</td>
<td>642,204</td>
</tr>
<tr>
<td><strong>Mesh Optimization [7]</strong></td>
<td>3,832</td>
<td>432</td>
</tr>
<tr>
<td>DEC Alpha</td>
<td>18,272</td>
<td>1,348</td>
</tr>
<tr>
<td><strong>Multi-resolution Approximation [9]</strong></td>
<td>349,792</td>
<td>N/A</td>
</tr>
<tr>
<td>IBM RS/6000 (model 560)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Running times of different mesh simplification algorithms.
21 3D hard copy of a simplified skull (9,820 triangles).