Interpolating and Approximating Implicit Surfaces from Polygon Soup

Chen Shen  James F. O’Brien  Jonathan R. Shewchuk

Presented by Yuan Chen

Outline
- Overview
- Algorithm
- Value Constraints
- Normal Constraints
- Interpolation and Approximation
- Preprocessing
- Results & Discussion

Overview
- Input: Arbitrary set of polygons
  -- holes, gaps, self-intersections, non-manifold
- Output: implicit surfaces
  -- watertight, with some other nice properties.

Algorithm
- Use MLS to generate the implicit surface
- Enforce true normal constraints
- Adjust the implicit surface to fit tightly around the input polygons and enclose all input vertices.
- A hierarchical fast evaluation scheme
- Optional preprocessing

Value Constraints at Points
- N points located at positions $p_i$, $i \in [1...N]$
- Build a function $f(x)$ which approximates the values $\phi_i$ at those points
- $b(x)$ is the vector of basis functions
- $c$ is the unknown vector of coefficients

Value Constraints at Points
- Standard least-squares fit:
  $$\begin{bmatrix} b^T(p_1) \\ \vdots \\ b^T(p_N) \end{bmatrix} c = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$
  , the resulting function $f(x) = b^T(x)c$
Value Constraints at Points
- Moving least-squares fit:
  \[ \begin{bmatrix} w(x, p_1) & \cdots & \cdots & w(x, p_n) \\ \vdots & \ddots & \ddots & \vdots \\ w(x, p_1) & \cdots & \cdots & w(x, p_n) \end{bmatrix} \begin{bmatrix} b_1^T(p_j) \\ \vdots \\ b_n^T(p_j) \end{bmatrix} c = \begin{bmatrix} w(x, p_1) & \cdots & \cdots & w(x, p_n) \\ \vdots & \ddots & \ddots & \vdots \\ w(x, p_1) & \cdots & \cdots & w(x, p_n) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} \]
- Weighting function \( w(x, p_j) \) is used to vary \( c \) depending on \( x \).

Images from B-CAM website.

Value Constraints at Points
- Using point constraints scattered over input polygons
  -- potentially a large number of points
  -- undesirable bumps and dimples
- So, scatter an infinite number of points continuously across the surface of each polygon.

Generated using integrated polygonal constraints
Generated using different densities of scattered point constraints
Images from Shen04 paper.

Value Constraints over Polygons
- MLS formulation for point constraints:
  \[ B^T(W(x))^2 B c(x) = B^T(W(x))^2 \phi \]

\[ \left( \sum_{j=1}^{K} w^2(x, p_j) b(p_j) b^T(p_j) \right) c(x) = \sum_{j=1}^{K} w^2(x, p_j) b(p_j) \phi \]

Value Constraints over Polygons
- Integrate value constraints over \( K \) polygons, \( \Omega_k \), \( k \in [1..K] \):
  \[ \left( \sum_{j=1}^{K} w^2(x, p_j) b(p_j) b^T(p_j) \right) c(x) = \sum_{j=1}^{K} w^2(x, p_j) b(p_j) \phi \]
  \[ \sum_{k=1}^{K} a_k = \sum_{k=1}^{K} a_k, \quad a_k = \int_{\Omega_k} w^2(x, p) b(p) b^T(p) dp \]
  \[ a_k = \int_{\Omega_k} w^2(x, p) b(p) \phi dp \]
Normal Constraints

- One of popular normal constraints: pseudo-normal constraints: [Turk and O'Brien, 1999]
  - A zero constraint at a point on the surface;
  - A positive constraint offset slightly outside the surface
  - A negative one slightly inside
  - This approach does not work well.
  - An essential singularity at that point.
  - Undesirable oscillatory behavior.

A function $S(x)$ associated with polygon $\Omega_k$ is defined, with normal constraints:

$$S_k(x) = \phi_k + (x - q_k)^T \tilde{n}_k = \psi_{sk} + \psi_{xk} x + \psi_{yk} y + \psi_{zk} z$$

$\tilde{n}_k$ is the normal of polygon $\Omega_k$, $\phi_k$ is an arbitrary point on the polygon.

These functions are blended using MLS instead of constant values.

Interpolation and Approximation

- The weighting function in the paper is set to be:
  $$w(x, p_i) = \frac{1}{\|x - p_i\|^2 + \epsilon^2}$$

The parameter $\epsilon$ allows a degree of control over the function's behavior.

- $\epsilon = 0$, interpolation
- $\epsilon \neq 0$,

Two problems in approximation

- Surfaces may fall away
  - $\epsilon$ is large

- The original vertices may fall outside of the implicit surface.
Adjustment
- Sample the average value of MLS function with the desired $\epsilon$ over the input polygons.
- Extract a surface using that as iso-value.
- Iteratively adjust the $\phi$ values assigned to the vertices

Preprocessing
- Remove interior polygons
- Force the normal to point outward and be consistent.

Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Reg</th>
<th>P/l</th>
<th>f</th>
<th>v</th>
<th>v, Out</th>
<th>Time</th>
<th>c</th>
<th>v, Out</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horse Loader</td>
<td>1</td>
<td>27</td>
<td>4</td>
<td>269</td>
<td>11.42</td>
<td>0.05</td>
<td>588</td>
<td>44.86</td>
<td>5</td>
</tr>
<tr>
<td>Trap</td>
<td>18</td>
<td>6.3</td>
<td>1</td>
<td>1800</td>
<td>5.80</td>
<td>0.6</td>
<td>380</td>
<td>10.28</td>
<td>10</td>
</tr>
<tr>
<td>Cow</td>
<td>18</td>
<td>6.6</td>
<td>1</td>
<td>1800</td>
<td>5.77</td>
<td>0.6</td>
<td>380</td>
<td>9.94</td>
<td>7</td>
</tr>
<tr>
<td>Beaver</td>
<td>18</td>
<td>6.8</td>
<td>1</td>
<td>1800</td>
<td>5.35</td>
<td>0.4</td>
<td>380</td>
<td>39.34</td>
<td>10</td>
</tr>
<tr>
<td>Dragon</td>
<td>18</td>
<td>8.5</td>
<td>1</td>
<td>2690</td>
<td>12.23</td>
<td>0.6</td>
<td>480</td>
<td>12.81</td>
<td>7</td>
</tr>
<tr>
<td>Scorpion</td>
<td>18</td>
<td>27</td>
<td>1</td>
<td>1500</td>
<td>4.54</td>
<td>0.5</td>
<td>480</td>
<td>41.50</td>
<td>5</td>
</tr>
<tr>
<td>Batwing Bat</td>
<td>11</td>
<td>0.05</td>
<td>1</td>
<td>1800</td>
<td>4.41</td>
<td>1</td>
<td>380</td>
<td>4.18</td>
<td>10</td>
</tr>
<tr>
<td>Icebox</td>
<td>11</td>
<td>0.6</td>
<td>8</td>
<td>1800</td>
<td>5.45</td>
<td>0.8</td>
<td>380</td>
<td>5.74</td>
<td>7</td>
</tr>
<tr>
<td>Rock Climber</td>
<td>11</td>
<td>3.9</td>
<td>6</td>
<td>1800</td>
<td>5.81</td>
<td>0.6</td>
<td>380</td>
<td>9.29</td>
<td>10</td>
</tr>
<tr>
<td>Armadillo</td>
<td>11</td>
<td>3.15</td>
<td>1</td>
<td>1800</td>
<td>4.95</td>
<td>0.4</td>
<td>380</td>
<td>7.53</td>
<td>7</td>
</tr>
<tr>
<td>Cuda Ripper</td>
<td>11</td>
<td>0.6</td>
<td>8</td>
<td>1800</td>
<td>4.41</td>
<td>0.6</td>
<td>380</td>
<td>5.31</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure from Shihfit paper.

Discussion
- Watertight & Have to be watertight
- Adjustment procedure…
- Time-consuming…
Building
Interpolating and Approximating
Implicit Surfaces from
Polygon Soup

Chen Shen  James F. O'Brien  Jonathan R. Shewchuk
University of California, Berkeley

SIGGRAPH 2004