Paint Mesh Cutting

Lubin Fan    Ligang Liu    Kun Liu
Zhejiang University
Outline

• Related work & Motivation
• Basic algorithm
  – Graph cuts based optimization
• Paint mesh cutting system
  – Global and local optimization
• Results & conclusion
  – Results
  – User study
  – Conclusion
Motivation

• How to cut out its tail?
Motivation

• **Automatic** algorithms

- Graph cuts [Katz et al. 2003]
- Hierarchical clustering [Gelfand et al. 2004]
- Primitive fitting [Attene et al. 2006]
- Random walks [Lai et al. 2008]
- Randomized cuts [Golovinskiy et al. 2008]
- Spectral clustering [Liu et al. 2004]
- Core extraction [Katz et al. 2005]
- Survey [Attene et al. 2006]
- Survey [Shamir et al. 2008]
- Survey [Chen et al. 2009]

**Complicated human perception**
Motivation

- **Interactive** tools for mesh segmentation
  - Direct UI

Direct UI
[Funkhouser et al. 2004, Chen et al. 2009]
Motivation

• **Interactive** tools for mesh segmentation
  – Direct UI
  – Sketch-based UI
Motivation

• **Interactive** tools for mesh segmentation
  – Direct UI
  – Sketch-based UI

Foreground/background Brushes (FBB)
[Ji et al. 2006, Zhang et al. 2010]
Motivation

- **Interactive** tools for mesh segmentation
  - Direct UI
  - Sketch-based UI

Cross-boundary Brushed (CBB)
[Zheng et al. 2010]
Motivation

- **Interactive** tools for mesh segmentation
  - Direct UI
  - Sketch-based UI

Foreground/background Brushes (FBB) [Ji et al. 2006, Zhang et al. 2010]

Cross-boundary Brushes (CB) [Zheng et al. 2010]

Can they be more intuitive?
Related Work

• Interactive image segmentation
  – Paint Selection [Liu et al. 2009]

image size: 24.5M

Intuitive UI
Instant feedback
Motivation

2D

Paint Selection [Liu et al. 2009]

3D

Mesh Segmentation
Motivation

• **Our Goal**
  – Easy and simple
  – Natural manner
  – Specify user intention intuitively
  – Instant feedback

*What you paint is what you get!*
This Work
This Work
Optimization

• Minimize the Energy

\[ E(L) = \sum_{v \in V} E_d(l_v) + \lambda \sum_{(v,u) \in \mathcal{E}} E_s(l_v, l_u) \]

- **Data Term** \( E_d(l_v) \), the penalty of assigning a label \( l_v \) to vertex \( v \) (1-foreground, 0-background).

- **Smoothness Term** \( E_s(l_v, l_u) \), the penalty for assigning different labels to two adjacent vertices \( v \) and \( u \).
Data Term – $E_d(\cdot)$

- How to define the penalty in data term?

$$E_d(l_v) = l_v \cdot L^f_v + (1 - l_v) \cdot L^b_v$$

$$L^f_v = -\ln\left(p_f M(v) + \varepsilon\right)$$

$$L^b_v = -\ln\left(p_b M(v) + \varepsilon\right)$$

$M(v)$ Surface Metric

$\varepsilon$ Probability
Surface Metric

- **Shape diameter function (SDF)** [Shamir et al. 2008]
  - Rely on volume information
  - Insensitive to noise
  - Insensitive to pose variation
Build SDF Models
Build SDF Models

Gaussian Mixture Model (GMM)

\[ p_f(\square) \]

\[ p_b(\square) \]
Data Term – $E_d(\cdot)$

**Data Term**

$$E_d(l_v) = \begin{cases} (1-l_v) \cdot K, & \forall v \in S^f \\ l_v \cdot L^f_v + (1-l_v) \cdot L^b_v, & \text{otherwise} \end{cases}$$

$$L^f_v = -\ln\left( p_f M(v) + \varepsilon \right)$$

$$L^b_v = -\ln\left( p_b M(v) + \varepsilon \right)$$
Energy Terms

- **Data Term**
- **Smoothness Term**

\[
E_s(l_v, l_u) = -|l_v - l_u| \cdot \ln\left((1 - \beta)n(v, u) + \beta g(v, u)\right)
\]

\[
n(v, u) = \frac{1 - n_v \cdot n_u}{2}
\]

\[
g(v, u) = \frac{e(v, u) - e_{\text{min}}}{e_{\text{max}} - e_{\text{min}}}
\]
Graph Cuts

Foreground (Source)

Background (Sink)

Min Cut

[Boykov and Jolly 2001]
System Overview

• Progressive expansion algorithm

• **Goal**
  – simple and easy to use
  – instant feedback (usually under 0.1 sec.)
  – expand the foreground continuously
Initial Global Optimization

Algorithm

- Compute SDF values.
- Construct global graph.
- Build the background GMM model $p_b(\cdot)$ with 4 components.
- Build the foreground GMM model $p_f(\cdot)$ with 2 components.
- Apply the graph cuts optimization.
Progressive Local Optimization
Progressive Local Optimization
Progressive **Local** Optimization

**Algorithm**

- Construct local graph.
- Build $p_f(\cdot)$ with 1 components.
- Update background sample vertices.
- Update $p_b(\cdot)$.
- Apply graph cuts optimization to local graph.
Progressive **Local** Optimization

**Algorithm**

- Construct local graph.
- Build $p_f(\cdot)$ with 1 components.
- Update background sample vertices.
- Update $p_b(\cdot)$.
- Apply graph cuts optimization to local graph.
Final Global Optimization

Algorithm

- Update $p_f(\cdot)$ with 2 components.
- Update $p_b(\cdot)$ with 4 components.
- Apply the graph cuts optimization.
Flow Chart

Original Model → Initial Global Optimization → Progressive Local Optimization → Final Global Optimization
Implementation Details
Implementation Details

• Cutting boundary refinement
  – Boundary smoothing by *snakes* on mesh [Ji et al. 2006]
Implementation Details

- Cutting boundary refinement
- Background painting

\[
E_d (l_v) = \begin{cases} 
(1-l_v) \cdot K, & \forall v \in S^f \\
l_v \cdot K, & \forall v \in S^b \\
l_v \cdot L_v^f + (1-l_v) \cdot L_v^b, & \text{otherwise}
\end{cases}
\]
Implementation Details

• Cutting boundary refinement
• Background painting
• Speedup
  – Computation of SDF values
    • Interpolation using the Poisson equation [Kovacic et al. 2010]
  – Graph cuts optimization
    • Parallel graph-cut method [Srandmark et al. 2010]
Results

Demo: armadillo
Results

Demo: patch: bunny
Results

• Independent on specific brushes
Results

• Insensitive to pose variation
Results

• Insensitive to noise
Results

- Running time

<table>
<thead>
<tr>
<th>Model</th>
<th># Vertex</th>
<th>T₁ (ms)</th>
<th>T₂ (ms)</th>
<th>T₃ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dino</td>
<td>28,150</td>
<td>53</td>
<td>10</td>
<td>178</td>
</tr>
<tr>
<td>Woman</td>
<td>5,691</td>
<td>8</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>Airplane</td>
<td>6,797</td>
<td>12</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>Armadillo</td>
<td>25,193</td>
<td>36</td>
<td>10</td>
<td>120</td>
</tr>
<tr>
<td>Bunny</td>
<td>34,835</td>
<td>54</td>
<td>11</td>
<td>248</td>
</tr>
</tbody>
</table>

* T₁, T₂, T₃ denote the computation time of the three steps in our algorithm, i.e., the initial global optimization, averaged local optimization, and the final global optimization, respectively.
Results

• More
**User Study**

- **Three sketch-based user interface algorithms**
  - Foreground/background brushes (FBB) [Ji et al. 2006]
  - Cross boundary brushes (CBB) [Zheng et al. 2010]
  - Foreground brushes (FB) - Paint Mesh Cutting
User Study

• **Assignment**
  
  – 16 participants
  
  – 16 models
  
  – Each participant test 6 models by using 3 algorithms respectively.
  
  – A short questionnaire
    
    • Accuracy
    
    • Efficiency
    
    • User intention
    
    • The favorite algorithm

Corpus
Analysis

• Interaction time

Averaged time and standard error of user interactions

Averaged time and standard error of the segmentation algorithm
Analysis

• Accuracy
  – Region-based measure
    [McGuinness et al. 2010]
    \[ BJI(S_1, S_2) = \frac{\|S_1^0 \cap S_2^0\|}{\|S_1^0 \cup S_2^0\|} \]

• Subjective evaluation

<table>
<thead>
<tr>
<th>Order</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FB</td>
</tr>
<tr>
<td>2</td>
<td>CBB</td>
</tr>
<tr>
<td>3</td>
<td>FBB</td>
</tr>
</tbody>
</table>

Comparison of accuracy for three tools: averaged BJI value and standard error.
Limitations & Future Work

• It is difficult to cut out the partial part for smooth surfaces.

• User need to specify many strokes to cut out some semantic parts from highly-detailed regions.
Conclusion

• Novel tool for interactive mesh segmentation
• Obtain the cutting results instantly
• Provide users a favorable experience on cutting mesh surfaces
• **What you paint is what you get!**
Thanks!
Acknowledgements

• Funding agencies:
  – National Natural Science Foundation of China (61070071)
  – 973 National Key Basic Research Foundation of China (No. 2009CB320801)

• Jie Xu for video narration