Interactive Implicit Modeling with Hierarchical Spatial Caching

Ryan Schmidt¹, Brian Wyvill¹, Eric Galin²

¹University of Calgary, Canada
²LIRIS-CNRS, Université Claude Bernard Lyon 1, France
Outline

• We present a hierarchical spatial caching technique that:
  – Enables interactive visualization of BlobTree implicit volume models
  – Is suitable for use in an interactive BlobTree modeling system

• We demonstrate an order-of-magnitude improvement in polygonization time

• We explore several applications
Background
Which Implicits?

- Many, many alternatives
- We choose Hierarchical Implicit Volume Modeling (BlobTrees [Wyvill et al 99])
  - Complex user-created models have been demonstrated
BlobTrees

• Hierarchical implicit model data structure

• Leaves are implicit volumes
  – Must have *bounded* scalar fields

• Internal nodes are *composition operators*
  – CSG, many types of blending, PCM, warping, texturing, etc…

• Implicit volume is procedurally defined at the root of the tree
BlobTree Scalar Fields

• Volume defined as \( f(p) \geq v_{iso} \)

• \( f(p) \) is *bounded* at all nodes of tree
  – non-zero values are contained inside a finite bounding box
  – local influence guaranteed
Shape Modeling with BlobTrees

- **Volume Modeling**
  - Blending, Warping and CSG → free-form and CAD

- **Functional representation**
  - Sharp edges, smooth surfaces

- **Scene Graph**
  - Animation

- **Construction History**
  - Non-linear editing
Interactive Modeling with BlobTrees

• Too Slow

• Visualization is the bottleneck
  – Designers need interactive feedback (surface)
  – Visualization algorithms do not scale interactively
    • Require many potential field evaluations

• Local updates? [Jevans88]
  – Do not help for large-scale assembly, expensive primitives, high-frequency details
Fundamental Problem

• BlobTree evaluations are too expensive
  – Tree traversal cost increases with each new primitive / operator
  – Tree optimization schemes are insufficient

• Our Solution: Hierarchical Spatial Caching
  – Discretely approximate tree branches
  – Reduces the cost of all field evaluations
  – Reduces tree depth
Hierarchical Spatial Caching
Related Work

• Akleman & Chen 99
  – Similar approach for ray-linear & ray-quadric implicits (only requires 2D cache)
  – No BlobTree hierarchy

• Barthe et al 02
  – Incremental modeling w/ volume data sets
  – Ray-traced triquadratic reconstruction
  – Scalability limits
Observations

• Polygonization algorithms run interactively on volume data sets  [Ferley et al 2000]

• For interactive modeling, most of the BlobTree structure is not changing frame-to-frame

• Memory is cheap
  – Less than $200 USD for 1GB of RAM
  – Easy to use
Global Spatial Caching

- Cache $f(p)$ in a volume data set:
  - $O(N) \rightarrow O(1)$ after lazy evaluation
  - Invalidate cache using field bounds of modified region

- Resolution is limited

- Does not fundamentally reduce evaluation cost
  - Large / expensive updates are still very slow
Hierarchical Spatial Caching

• Use multiple Spatial Caches

• Insert into the BlobTree as *Cache Nodes*
  – $O(m) \rightarrow O(1)$ for cache node subtree

• Faster updates than global spatial cache
  – Interactive modification only affects parent cache node(s)
  – Large / complex updates are possible
Standard BlobTree

Union

Blend
Blob Tree With Cache Node

Union

Cache

Blend
Cache Implementation
Spatial Cache Implementation

- Uniform Grids
- Trilinear and Triquadratic sampling
- Lazy evaluation
  - minimizes start-up overhead
Reconstruction Filters

• Trilinear Filter
  – 8 samples
  – $C^0$ Continuous [Marschner & Lobb 94]
  – Interpolating

• Triquadratic Filter  [Barthe et al 02]
  – 27 samples required
  – $C^1$ Continuous
  – Approximating
Gradient Continuity

Trilinear

Triquadratic
Interactive Visualization

• Trilinear filter is twice as fast
  – But gradient is unacceptable

• Gradient evaluation is only 10% of the polygonization cost

• Hybrid Solution:
  – Trilinear reconstruction for field value
  – Triquadratic reconstruction for gradient

• Perceptually smooth surface
Cache Sampling w/ Lazy Evaluation

← Cache (Initially empty)

← Cached Subtree
Step 1: Incoming Point Query
Step 2: Cached Subtree Evaluation
Step 3: Cache Initialization
Step 4: Field Approximation
Step 1: Incoming Point Query
Step 2: Subtree Evaluation
Step 3: Cache Initialization
Step 4: Field Approximation
Step 1: Incoming Point Query
Step 4: Field Approximation
Sharp Feature Reconstruction

No Cache  128³ Cache  256³ Cache

(polygonizer resolution is much higher than cache resolution)
Profiling Results
Test Model

9490 point primitives
## Static Polygonization

- Caches cleared before each test
- Time increase when resolution doubles:
  - Cache $\rightarrow$ 1.7x
  - No Cache $\rightarrow$ $\sim$4x

<table>
<thead>
<tr>
<th>Cubes</th>
<th>Cache</th>
<th>No Cache</th>
<th>Ratio</th>
<th>Triangles (Approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$32^3$</td>
<td>5.77</td>
<td>4.90</td>
<td>0.8x</td>
<td>4,000</td>
</tr>
<tr>
<td>$64^3$</td>
<td>10.34</td>
<td>14.36</td>
<td>1.4x</td>
<td>16,000</td>
</tr>
<tr>
<td>$128^3$</td>
<td>17.40</td>
<td>51.97</td>
<td>3x</td>
<td>61,000</td>
</tr>
<tr>
<td>$256^3$</td>
<td>29.23</td>
<td>199.37</td>
<td>6.5x</td>
<td>239,000</td>
</tr>
<tr>
<td>$512^3$</td>
<td>49.83</td>
<td>809.66</td>
<td>16x</td>
<td>955,000</td>
</tr>
</tbody>
</table>
Head Translation Test
(No Hair)

Cache
No Cache

Polygonization Time in Seconds

Timestep

- Cached
- Cached (No Hair)
- No Cache
- Cached (2x Resolution)
- No Cache (No Hair)
Head Translation Test
(2x Polygonizer Resolution)

Polygionization Time in Seconds

Timestep

Cached
Cached (No Hair)
Cached (2x Resolution)
No Cache
No Cache (No Hair)
Local Update Polygonization

- Recompute polygonization only in modified region
- Point / Hair Test simulates detail work on hair component (6500 point primitives)

<table>
<thead>
<tr>
<th>Test</th>
<th>Cubes</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Translation</td>
<td>$60^3$</td>
<td>7×</td>
</tr>
<tr>
<td>Head Translation</td>
<td>$120^3$</td>
<td>12×</td>
</tr>
<tr>
<td>Point / Hair Test</td>
<td>$60^3$</td>
<td>30×</td>
</tr>
<tr>
<td>Point / Hair Test</td>
<td>$120^3$</td>
<td>47×</td>
</tr>
</tbody>
</table>
Applications
Interactive Model Assembly
Demo
Proof-of-concept: Interactive Sketch-Based BlobTree Modeling

(Come see our SIGGRAPH 05 sketch)
Future Work

• Improving Accuracy
  – Alternate spatial caching schemes?

• Automatic Cache Management Algorithms
  – Take advantage of interactive context

• Minimize invalidation regions
  – Better bounds

• Interactive implicit modeling interfaces
  – Largely unexplored territory…
Summary

• We described hierarchical spatial caching for the BlobTree implicit modeling system

• We compared polygonization times with traditional approaches
  – An order-of-magnitude improvement was found

• We described several interactive modeling applications enabled by this technique
Questions?