Massively Parallel Batch Neural Gas for BVH Construction

R. Weller\textsuperscript{1}, D. Mainzer\textsuperscript{2}, A. Srinivas\textsuperscript{1}, M. Teschner\textsuperscript{3} and G. Zachmann\textsuperscript{1}

\textsuperscript{1} University of Bremen, Germany
\textsuperscript{2} Clausthal University, Germany
\textsuperscript{3} University of Freiburg, Germany

VRIPHYS Sept 2014, Bremen
Motivation for BVHs

- Commonly used in wide variety of graphics problems
  - Collision detection, Ray tracing, Culling

- Bounding Volumes:
  - AABB, OBB, Sphere, Convex Hull, ...

- Branching factor and splitting criteria
Previous Work

- **Binary axis aligned bounding box tree [Bergen97]**
  - Splitting AABB along longest side

- **Surface area heuristic BVHs on GPU [LGS*09]**
  - Slower build time than CPU version
  - Faster traversal of BVH

- **Oriented bounding box-tree construction [GLM96]**
  - Presented a new construction method
  - Optimal Solution in $O(n^3)$ and hard to implement [O’R85]
Volume-based approaches

- Volumetric method: volumetric poly-disperse sphere packing
  - All sphere are *inside* and *do not overlap* each other
- Need different approach for constructing hierarchy tree
- Investigate on other factors such as
  - Splitting criteria
  - Branching factor
Biggest Sphere Splitting Criterion

Motivation

Previous Work

Volume-based approaches

BNG

Parallelization

Results

Conclusions/Future
Outer Sphere Splitting Criterion

Motivation

Previous Work

Volume-based approaches

BNG Work

Parallelization

Results

Conclusions/Future
Hierachy Creation

Motivation           Previous Work

Sweep-Plane           Scene Subdivision Work           Our Algorithm           Results           Conclusion/Future

Work
Batch Neural Gas

- Cost function minimize mean squared Euclidean distance for each data point to its nearest center point
- Very robust – independent of initialization of center points
- Extendable to define importance of a data point
- Rank for prototypes (with \( n \) prototypes):
  \[ k_{ij} := | \{ w_k : d(x_j, w_k) < d(x_j, w_i) \} | \in \{0, \ldots, n\} \]
- Position for prototypes:
  \[ w_i := \frac{\sum_{j=0}^{m} h_\lambda(k_{ij})x_j}{\sum_{j=0}^{m} h_\lambda(k_{ij})} \]
- Convergence rate controlled by monotonically decreasing function \( h_\lambda(\ldots) \)
- BNG only utilizes location of the center of the spheres
- Ignores the extent of the spheres
  - Prototypes avoid regions covered with a very large sphere
  - Regions treated as outlier \(\rightarrow\) Non-Optimal IST
- Extended BNG Version
  - Magnification controlled BNG [HHV06]
    \[
    w_i := \frac{\sum_{j=0}^{m} h_{\lambda}(k_{ij}) v(x_j) x_j}{\sum_{j=0}^{m} h_{\lambda}(k_{ij}) v(x_j)}
    \]
  - Use volume of the sphere: \(v(x_j) = \frac{4}{3} \pi r^3\)
  - Runtime: \(\mathcal{O}(n \log n)\)
- BNG only utilizes location of the center of the spheres
- Ignores the extent of the spheres
  - Prototypes avoid regions covered with a very large sphere
  - Regions treated as outlier → Non-optimal bounding spheres
- Extended BNG Version
  - Magnification controlled BNG [HHV06]
    \[
    w_i := \frac{\sum_{j=0}^{m} h_\lambda(k_{ij})v(x_j)x_j}{\sum_{j=0}^{m} h_\lambda(k_{ij})v(x_j)}
    \]
  - Use volume of the sphere:
    \[
    v(x_j) = \frac{4}{3} \pi r^3
    \]
  - Runtime: \( O(n \log n) \)
Non-Optimal Subdivision

Motivation           Previous Work           Sweep-Plane
Scene Subdivision
Our Algorithm           Results           Conclusion/Future
Work
Batch Neural Gas

- Need an algorithm which considers volume during IST construction
- BNG clustering with a modified cost function which considers volume
- Very robust – independent of initialization of center points.
Batch Neural Gas Clustering

\[ k_{ij} := \left\{ w_k : d(x_i, w_k) < d(x_j, w_i) \right\} \sum_{j=0}^{m} h_\lambda(k_{i,j}) v(x_j) x_j \]

\[ w_i = \frac{\sum_{j=0}^{m} h_\lambda(k_{i,j}) v(x_j) x_j}{\sum_{j=0}^{m} h_\lambda(k_{i,j})} \]
Parallel BNG

- BNG Hierarchy construction has a complexity of $O(n \log n)$
- BNG perfectly suited for parallelization
- Parallelization in first level of hierarchy is straightforward
  - Ordering of $k_{ij}$ and $h_\lambda(k_{ij})v(x_j)x_j$ can be computed independently for each sphere
  - Summing up $\rightarrow$ using parallel scan algorithm [SHG08]
  - Distance sphere to all prototypes $\rightarrow$ assignment sphere to prototypes (a sphere is assigned to exactly one prototype)
- Triggering an own parallel process for each sub-tree is not efficient for parallel processing and is not memory efficient
<table>
<thead>
<tr>
<th>Prototype index</th>
<th>Sub tree 1</th>
<th>Sub tree 2</th>
<th>Sub tree 3</th>
<th>Sub tree 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>$w_{1,1}(x_i)$</td>
<td>$w_{1,2}(x_i)$</td>
<td>$w_{1,3}(x_i)$</td>
<td>$w_{1,4}(x_i)$</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>$w_{2,1}(x_i)$</td>
<td>$w_{2,2}(x_i)$</td>
<td>$w_{2,3}(x_i)$</td>
<td>$w_{2,4}(x_i)$</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>$w_{3,1}(x_i)$</td>
<td>$w_{3,2}(x_i)$</td>
<td>$w_{3,3}(x_i)$</td>
<td>$w_{3,4}(x_i)$</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>$w_{4,1}(x_i)$</td>
<td>$w_{4,2}(x_i)$</td>
<td>$w_{4,3}(x_i)$</td>
<td>$w_{4,4}(x_i)$</td>
</tr>
</tbody>
</table>

Prev. Prototype

<table>
<thead>
<tr>
<th>Sphere</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>6</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>10</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
- BNG hierarchy construction on CPU has complexity of $\mathcal{O}(n \log n)$
- Parallelization of BNG reduces complexity to $\mathcal{O}(\log^2 n)$
Results: Hierarchy Construction Performance

- Intel i7 CPU with 8GB RAM & NVIDIA GeForce GTX 780 with 3GB

**Motivation**

**Previous Work**

Volume-based approaches  
BNG  
Parallelization

**Results**

**Conclusions/Future Work**
Results: Runtime Performance

Motivation

Previous Work

Volume-based approaches

BNG

Parallelization

Results

Conclusions/Future

collision test between pig and statue

Collision test between pig and cow

Graphs showing runtime performance for different numbers of spheres.
Conclusions/Future Work

- Completely GPU-based method for BVH for volumetric object representations
- Parallel BNG reduce complexity from $O(n \log n)$ to $O(\log^2 n)$
- Outperforms CPU version by factor 15
- Better hierarchy: faster than naïve splitting approaches for collision queries
- Our method can also be used with different branching factors

- Apply approach to other volumetric object representations than sphere packing's, e.g. tetrahedral or ellipses
- Use this approach for classical outer BVH
- Ray tracing and occlusion culling
Thank you!
Any Questions?

DFG grant TRR 8/3-2013