SAH guided spatial split partitioning for fast BVH construction

Per Ganestam and Michael Doggett
Lund University
Opportunistic triangle splitting for higher quality BVHs

- Bounding Volume Hierarchies (BVH) are a simple, compact 3D data structure commonly used in Ray Tracing
- Objective is fast construction, fast tracing, and minimal memory
- Traversal cost: Whiter pixels mean higher cost
BVH construction

• Focus on a top down approach
• Recursively subdivide triangles
  • Decide where to divide
  • Surface Area Heuristic (SAH) measures cost
  • Sweep SAH is a greedy top down method
• Creating a binary tree
Surface Area Heuristic

\[ C_i \sum_{n \in I} \frac{A(n)}{A(root)} + C_l \sum_{n \in L} \frac{A(n)}{A(root)} + C_t \sum_{n \in L} \frac{A(n)}{A(root)} N(n), \]

- \( A(n) \) is the surface area
- Find the ratio of current triangle’s BB area to parent’s
- Consider cost, \( C \), of
  - intersection of internal, \( i \), and leaf, \( l \), nodes
  - \( t \), traversal
- Which partitioning gives the lowest cost?
Bonsai BVH construction algorithm

- From "Bonsai: Rapid Bounding Volume Hierarchy Generation using Mini Trees", Ganestam, Barringer, Doggett, and Akenine-Möller, JCGT, Sep. 2015
Bonsai Partitioning using triangle mid-points

- Guarantees no empty partitions
Bonsai algorithm with pruning

- For mini-trees with large bounds
- Delete node and promote children

Before pruning

After pruning

Original mini tree

New mini trees

Deleted nodes
Triangle Splitting

- Problem: Big and small triangles don’t mix well
- Split clipping splits the bounding box
- Previous algorithms split before construction
  - Early Split Clipping [EG07] and EVH [DHK08]
  - Can result in more triangles and poor splits
  - Also adds to the overall construction time
- Some algorithms need per scene parameters
- Our solution: SAH Spatial Split Partitioning
SAH Spatial Split Partitioning
SAH Spatial Split Partitioning

• Split triangles while partitioning
• Create triangle sets
• Only split triangles when SAH cost is lower than not splitting
SAH Spatial Split Partitioning

- Brown box is mid-point bounds and used to find split plane.
SAH Spatial Split Partitioning

Triangle Sets
- Disjoint set
- $D_L, D_R$

- Brown box is mid-point bounds and used to find split plane
SAH Spatial Split Partitioning

Triangle Sets
- Disjoint set
  - $D_L$, $D_R$
- Overlap sets
  - $O_L$, $O_R$

- Brown box is mid-point bounds and used to find split plane
- Overlap sets have triangles that overlap the split plane
SAH Spatial Split Partitioning

Triangle Sets
- Disjoint set
  - $D_L$, $D_R$
- Overlap sets
  - $O_L$, $O_R$
- Split sets
  - $S_L$, $S_R$

- Brown box is mid-point bounds and used to find split plane
- Overlap sets have triangles that overlap the split plane
- Split sets contain all overlap triangles
SAH Spatial Split Partitioning algorithm

- Recursively partition using mid-point and SAH cost
  - Take lowest cost of split vs. not split
  - Until mini trees are between 512 and 8K triangles (based on overall scene size)
- Mini trees processed using Sweep SAH
Efficient memory allocation

- Problem: Using separate arrays and merging
- Solution: In place, memory growing
- Memory is allocated for all triangles + 20%
- One half of split triangle overwritten in-place
  - and the other half added to the other side
- What empty space is left is rebalanced
- If split triangles don’t fit
  - Allocate more memory for right set + 20%
Efficient memory allocation

Initial triangle array

After partitioning

After splitting and memory reordering

Left first

Left last

Right first

Right last

$D_L$  $O_L$

$D_R$  $O_R$

$S_L$

$I_L$ $I_L$ $I_L$

$S_R$

$I_R$ $I_R$

Left last free space

Right last free space
Results

• Implemented in Intel’s Embree Ray Tracer, version 2.7.1
• Results generated on laptop **CPU** (MacBook Pro) with AVX2 support
• Consider both build and tracing times
• Compare to algorithms
  • Without splitting
    • SweepSAH
    • Bonsai
    • BinnedSAH (included in Embree)
  • With splitting
    • BonsaiS (Proposed algorithm)
    • SweepPre (Proposed algorithm as a pre splitting pass for Sweep)
    • BinnedPre (Pre-split algorithm in Embree)
    • BinnedS (Spatial split algorithm in Embree, based on SBVH [SFD09])
## Results

<table>
<thead>
<tr>
<th>Triangles</th>
<th>Arabic City</th>
<th>Crown</th>
<th>Dragon</th>
<th>Fairy Forest</th>
<th>Italian City</th>
<th>Kalabsha</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>416,236</td>
<td>4,868,924</td>
<td>7,349,978</td>
<td>174,117</td>
<td>382,029</td>
<td>4,542,705</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWEEP SAH</td>
<td>254</td>
<td>295 [100%]</td>
<td>4680</td>
<td>325 [100%]</td>
<td>7595</td>
<td>129 [100%]</td>
<td>117</td>
<td>299 [100%]</td>
<td>240</td>
<td>300 [100%]</td>
<td>4910</td>
</tr>
<tr>
<td>BONSAI</td>
<td>34</td>
<td>291 [99%]</td>
<td>469</td>
<td>331 [102%]</td>
<td>664</td>
<td>131 [102%]</td>
<td>16</td>
<td>299 [100%]</td>
<td>31</td>
<td>292 [97%]</td>
<td>451</td>
</tr>
<tr>
<td>BINNED SAH</td>
<td>33</td>
<td>302 [102%]</td>
<td>447</td>
<td>322 [99%]</td>
<td>674</td>
<td>126 [98%]</td>
<td>14</td>
<td>296 [99%]</td>
<td>31</td>
<td>297 [99%]</td>
<td>459</td>
</tr>
<tr>
<td>SWEEP PRE</td>
<td>374</td>
<td>243 [82%]</td>
<td>5414</td>
<td>322 [99%]</td>
<td>8087</td>
<td>134 [104%]</td>
<td>147</td>
<td>301 [100%]</td>
<td>322</td>
<td>239 [80%]</td>
<td>6098</td>
</tr>
<tr>
<td>BONSAI S</td>
<td>53</td>
<td>247 [84%]</td>
<td>835</td>
<td>316 [97%]</td>
<td>827</td>
<td>132 [102%]</td>
<td>24</td>
<td>297 [99%]</td>
<td>47</td>
<td>231 [77%]</td>
<td>765</td>
</tr>
<tr>
<td>BINNED PRE</td>
<td>91</td>
<td>267 [91%]</td>
<td>980</td>
<td>317 [98%]</td>
<td>1577</td>
<td>134 [104%]</td>
<td>40</td>
<td>305 [102%]</td>
<td>77</td>
<td>254 [85%]</td>
<td>811</td>
</tr>
<tr>
<td>BINNED S</td>
<td>404</td>
<td>237 [80%]</td>
<td>4130</td>
<td>304 [94%]</td>
<td>2205</td>
<td>124 [96%]</td>
<td>129</td>
<td>289 [97%]</td>
<td>316</td>
<td>229 [76%]</td>
<td>4646</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangles</th>
<th>Mini</th>
<th>Power Plant</th>
<th>Sala</th>
<th>San Miguel</th>
<th>Sibenik</th>
<th>Sponza</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>912,411</td>
<td>12,759,246</td>
<td>400,637</td>
<td>7,880,512</td>
<td>79,380</td>
<td>262,267</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
<th>Build</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWEEP SAH</td>
<td>719</td>
<td>593 [100%]</td>
<td>14347</td>
<td>617 [100%]</td>
<td>283</td>
<td>298 [100%]</td>
<td>8208</td>
<td>394 [100%]</td>
<td>49</td>
<td>222 [100%]</td>
<td>178</td>
</tr>
<tr>
<td>BONSAI</td>
<td>74</td>
<td>581 [98%]</td>
<td>1261</td>
<td>606 [98%]</td>
<td>34</td>
<td>298 [100%]</td>
<td>711</td>
<td>363 [92%]</td>
<td>9</td>
<td>225 [101%]</td>
<td>22</td>
</tr>
<tr>
<td>BINNED SAH</td>
<td>83</td>
<td>592 [100%]</td>
<td>1311</td>
<td>700 [113%]</td>
<td>35</td>
<td>332 [111%]</td>
<td>798</td>
<td>386 [98%]</td>
<td>7</td>
<td>222 [100%]</td>
<td>22</td>
</tr>
<tr>
<td>SWEEP PRE</td>
<td>952</td>
<td>548 [92%]</td>
<td>17603</td>
<td>469 [76%]</td>
<td>359</td>
<td>296 [100%]</td>
<td>9301</td>
<td>359 [91%]</td>
<td>53</td>
<td>222 [100%]</td>
<td>213</td>
</tr>
<tr>
<td>BONSAI S</td>
<td>106</td>
<td>538 [91%]</td>
<td>1881</td>
<td>437 [71%]</td>
<td>47</td>
<td>286 [96%]</td>
<td>835</td>
<td>339 [86%]</td>
<td>11</td>
<td>218 [98%]</td>
<td>29</td>
</tr>
<tr>
<td>BINNED PRE</td>
<td>166</td>
<td>617 [104%]</td>
<td>3625</td>
<td>606 [98%]</td>
<td>90</td>
<td>339 [114%]</td>
<td>2168</td>
<td>388 [98%]</td>
<td>14</td>
<td>229 [103%]</td>
<td>56</td>
</tr>
<tr>
<td>BINNED S</td>
<td>739</td>
<td>572 [96%]</td>
<td>13835</td>
<td>441 [71%]</td>
<td>342</td>
<td>292 [98%]</td>
<td>7118</td>
<td>341 [87%]</td>
<td>53</td>
<td>213 [96%]</td>
<td>168</td>
</tr>
</tbody>
</table>

All measurements in ms
## Results

### Without splitting

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Build</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crown</td>
<td>4,868,924</td>
<td></td>
</tr>
<tr>
<td>Dragon</td>
<td>7,349,978</td>
<td></td>
</tr>
<tr>
<td>Fairy Forest</td>
<td>174,117</td>
<td></td>
</tr>
<tr>
<td>Italian City</td>
<td>382,029</td>
<td></td>
</tr>
<tr>
<td>Kalabsha</td>
<td>4,542,705</td>
<td></td>
</tr>
</tbody>
</table>

### Primer Test

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Build</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWEEP-SAH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BONSAI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BINNED-SAH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWEEP-PRE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BONSAI-S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BINNED-PRE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BINNED-S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Triangles Test

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Build</th>
<th>Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini</td>
<td>912,411</td>
<td></td>
</tr>
<tr>
<td>Power Plant</td>
<td>12,759,246</td>
<td></td>
</tr>
<tr>
<td>Sala</td>
<td>400,637</td>
<td></td>
</tr>
<tr>
<td>San Miguel</td>
<td>7,880,512</td>
<td></td>
</tr>
<tr>
<td>Sibenik</td>
<td>79,380</td>
<td></td>
</tr>
<tr>
<td>Sponza</td>
<td>262,267</td>
<td></td>
</tr>
</tbody>
</table>

### Summary

- **SWEEP-SAH**: High efficiency across all datasets.
- **BONSAI**: Improves slightly over SWEEP-SAH.
- **BINNED-SAH**: Further improvement with better performance.

**Overall**: The proposed algorithms outperform previous methods in terms of build time and trace size.

---

**Notation**:
- % values indicate percentage improvement.
- [ ] values indicate absolute improvement in build or trace size.

---

**References**:
## Results

<table>
<thead>
<tr>
<th>Triangles</th>
<th>Arabic City</th>
<th>Crown</th>
<th>Dragon</th>
<th>Fairy Forest</th>
<th>Italian City</th>
<th>Kalabsha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>416,236</td>
<td>4,868,924</td>
<td>7,549,978</td>
<td>174,117</td>
<td>382,029</td>
<td>4,542,705</td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWEEP SAH</td>
<td>254</td>
<td>4680</td>
<td>7595</td>
<td>117</td>
<td>240</td>
<td>4,910</td>
</tr>
<tr>
<td>BONSAI</td>
<td>34</td>
<td>469</td>
<td>664</td>
<td>16</td>
<td>31</td>
<td>451</td>
</tr>
<tr>
<td>BINNED SAH</td>
<td>33</td>
<td>447</td>
<td>674</td>
<td><strong>14</strong></td>
<td>31</td>
<td>459</td>
</tr>
<tr>
<td>SWEEP PRE</td>
<td><strong>374</strong></td>
<td><strong>5414</strong></td>
<td>8087</td>
<td>147</td>
<td>322</td>
<td>6098</td>
</tr>
<tr>
<td>BONSAI S</td>
<td>53</td>
<td><strong>535</strong></td>
<td>827</td>
<td>24</td>
<td>47</td>
<td>765</td>
</tr>
<tr>
<td>BINNED PRE</td>
<td>91</td>
<td>980</td>
<td>1577</td>
<td>40</td>
<td>77</td>
<td>811</td>
</tr>
<tr>
<td>BINNED S</td>
<td><strong>404</strong></td>
<td><strong>4,130</strong></td>
<td><strong>2,205</strong></td>
<td><strong>129</strong></td>
<td><strong>316</strong></td>
<td><strong>4,464</strong></td>
</tr>
<tr>
<td>With splitting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangles</th>
<th>Mini</th>
<th>Power Plant</th>
<th>Sala</th>
<th>San Miguel</th>
<th>Sibenik</th>
<th>Sponza</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>912,411</td>
<td>12,759,246</td>
<td>400,637</td>
<td>7,880,512</td>
<td>79,380</td>
<td>262,267</td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWEEP SAH</td>
<td>719</td>
<td>14347</td>
<td>283</td>
<td>8208</td>
<td>49</td>
<td>178</td>
</tr>
<tr>
<td>BONSAI</td>
<td>74</td>
<td>1261</td>
<td>34</td>
<td>711</td>
<td>9</td>
<td>22</td>
</tr>
<tr>
<td>BINNED SAH</td>
<td>83</td>
<td>1311</td>
<td>35</td>
<td>798</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>SWEEP PRE</td>
<td><strong>952</strong></td>
<td><strong>17,603</strong></td>
<td><strong>347</strong></td>
<td><strong>379</strong></td>
<td><strong>53</strong></td>
<td><strong>213</strong></td>
</tr>
<tr>
<td>BONSAI S</td>
<td><strong>106</strong></td>
<td><strong>1881</strong></td>
<td><strong>47</strong></td>
<td><strong>800</strong></td>
<td><strong>11</strong></td>
<td><strong>29</strong></td>
</tr>
<tr>
<td>BINNED PRE</td>
<td>166</td>
<td>3625</td>
<td>90</td>
<td>2168</td>
<td>14</td>
<td>56</td>
</tr>
<tr>
<td>BINNED S</td>
<td>739</td>
<td>13,835</td>
<td>342</td>
<td>7118</td>
<td>53</td>
<td>168</td>
</tr>
</tbody>
</table>

**With splitting**
BonsaiS is one of 2 best tracing 10/12 times
Results

• For 4 scenes BonsaiS traces faster than BinnedS (considered to be the HQ builder)
• SweepPre is twice in the top two tracers
  • and is typically between BinnedS and BinnedPre (Embree builders)
• BinnedS always improves tracing performance of BinnedSAH, but at much longer build times
• BinnedS allocated 34x more memory compared to BonsaiS (measured with Valgrind)
• For Arabic City BonsaiS allocated 735MB, BinnedS 25GB
Parallel scaling

- Intel E5-2643V3 dual socket 12 core CPU
- Powerplant scene
- Per frame tracing performance on the 12 core CPU
  - BonsaiS 105ms
  - BinnedSAH 162ms

![Graph showing parallel scaling with BonsaiS and BinnedSAH]
Conclusion

• Simple top down triangle splitting and BVH construction algorithm
• Integrated or preprocess triangle splitter
• Fast BVH construction and fast ray tracing

• Source code will be available in Embree and on paper web page

• Acknowledgements: Thanks to ELLIIT and the Intel Visual Computing Institute for funding.
Thanks for listening!
Bonus slides
Binary Trees

- GPU (Intel Iris Pro 5200) traversal, CPU shading
- BonsaiS improves tracing performance compared to SweepSAH for all scenes
- Similar tracing improvement to [KA13], but with fewer split triangles
Triangle Counts and SAH costs

• BonsaiS creates fewer additional triangles than the two triangle split methods available in Embree
• BonsaiS can split far fewer triangles and get the same and faster tracing
• BonsaiS consistently produces lower SAH costs
SAH Spatial Split Partitioning

Triangle Sets
- Disjoint set
  - $D_L$, $D_R$
- Overlap sets
  - $O_L$, $O_R$
- Split sets
  - $S_L$, $S_R$

- Brown box is mid-point bounds and used to find split plane
- Overlap sets have triangles that overlap the split plane
- Split sets contain all overlap triangles