Eurographics 2023

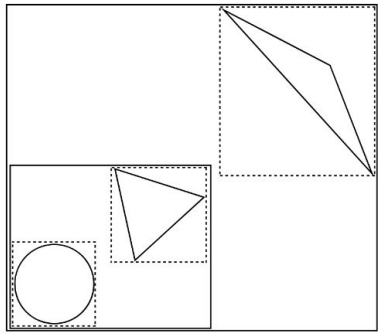
Parallel Transformation of Bounding Volume Hierarchies into Oriented Bounding Box Trees

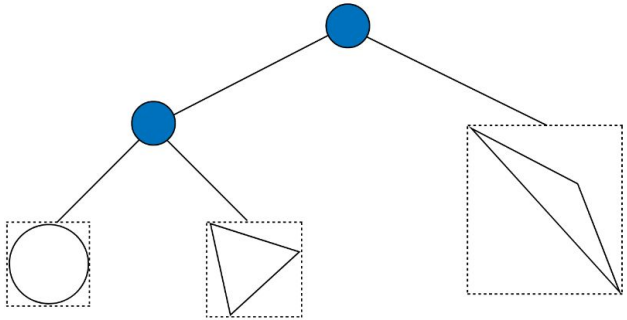
Nick Vitsas
Iordanis Evangelou
Georgios Papaioannou
Anastasios Gkaravelis

Department of Informatics
Athens University of Economics and Business
Greece

Introduction

Bounding Volume Hierarchy

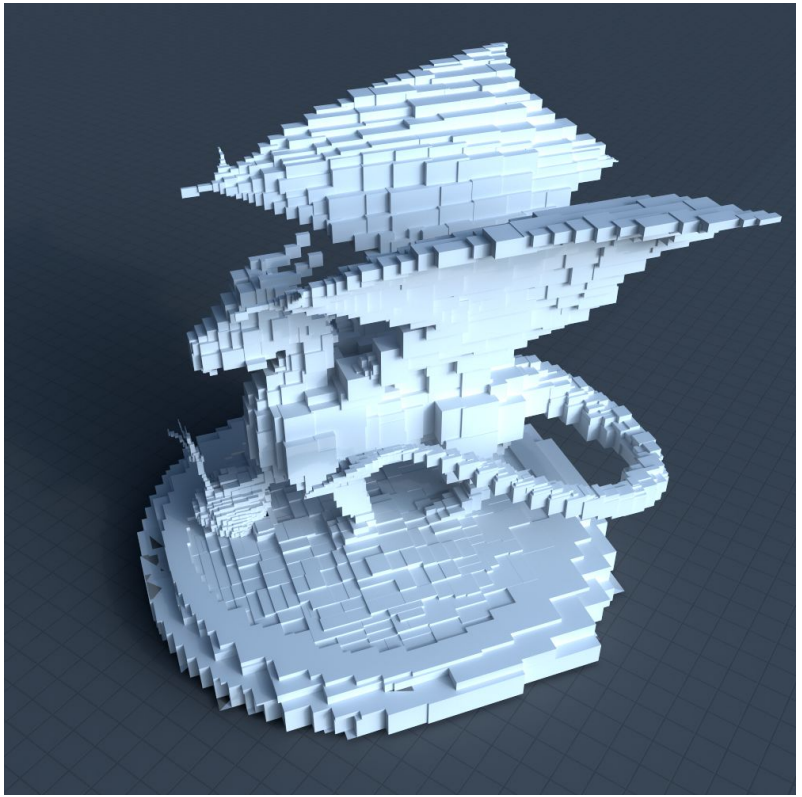




Introduction

Bounding Volume Hierarchy

Axis-Aligned Bounding Boxes (AABBs) are very common



Introduction

Bounding Volume Hierarchy

Axis-Aligned Bounding Boxes (AABBs) are very common

Bounding Volume
- Any closed geometric shape

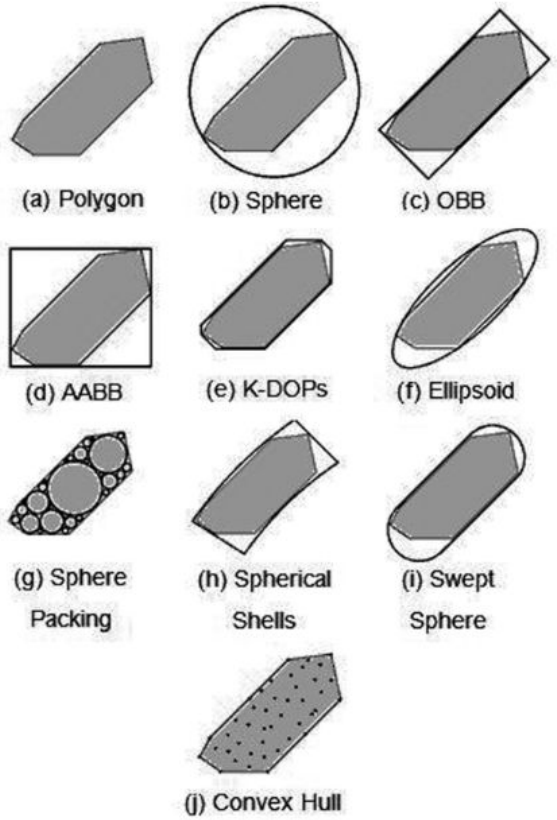
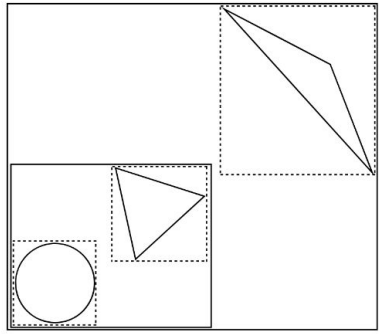


Image from A literature review of bounding volumes hierarchy focused on collision detection, Dinas, Simena, and José M. Bañón

Related Work

- **Surface Area Heuristic (SAH)**
- Axis-Aligned Bounding Boxes (AABBs)
- Object Partitioning
  - Recursively subdivide the list of primitives
  - Keep the partition that minimizes the cost



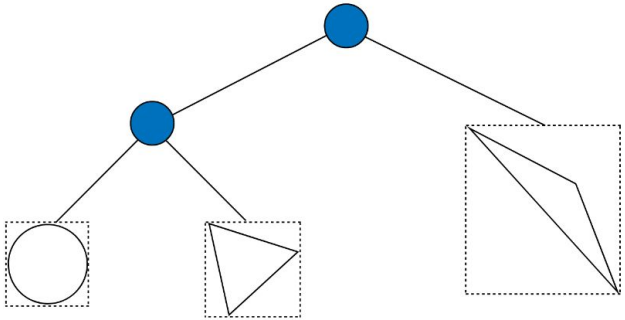
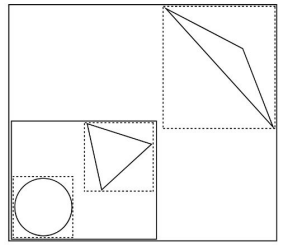
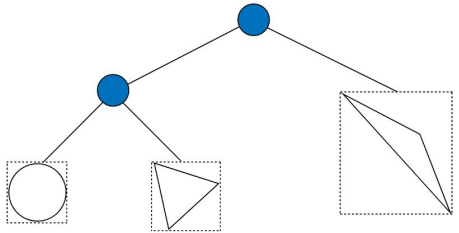


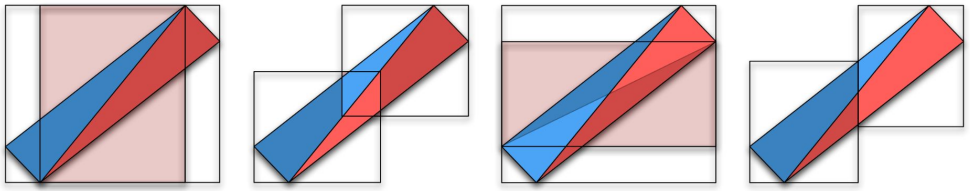
Image from PBRT v3

Related Work

- **Surface Area Heuristic (SAH)**
- Axis-Aligned Bounding Boxes (AABBs)
- Object Partitioning
  - Recursively subdivide the list of primitives
  - Keep the partition that minimizes the cost
- **Spatial Partitioning**



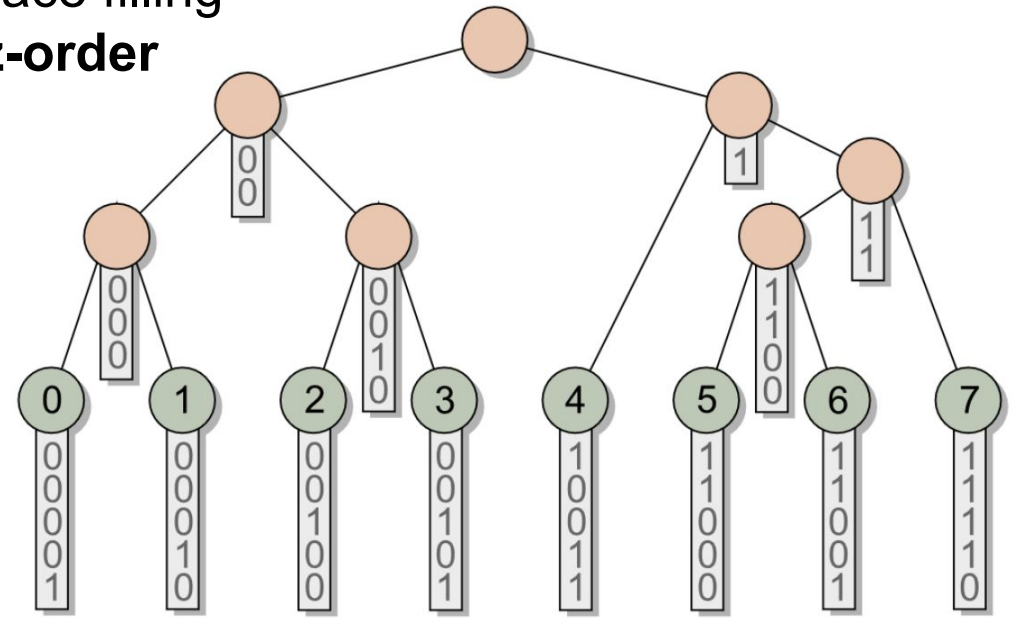




SBVH, Spatial splits in bounding volume hierarchies, Stitch et al.

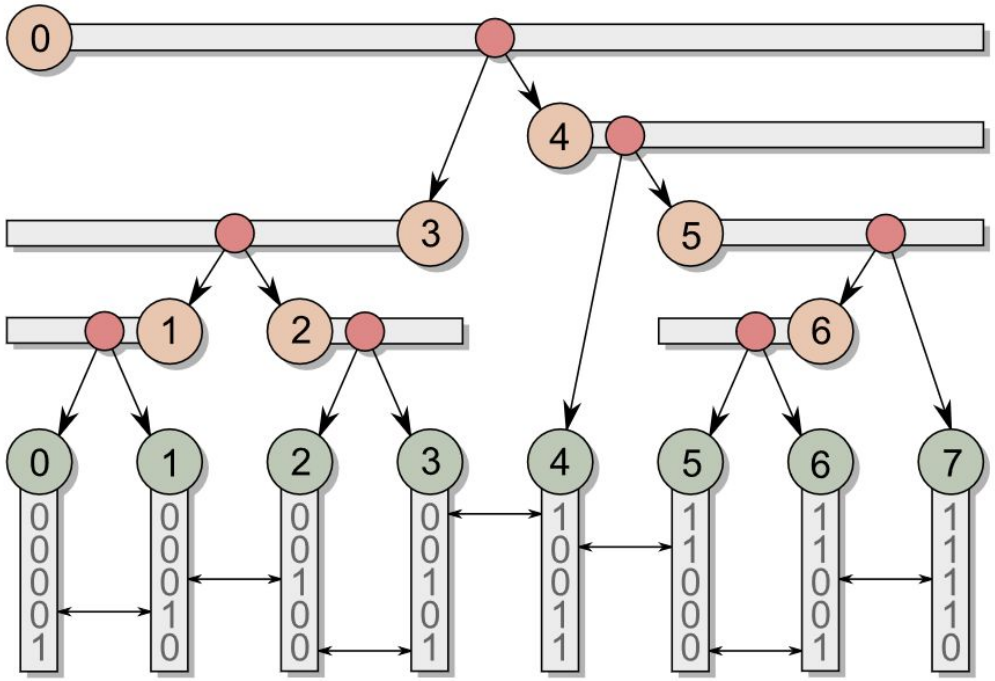
Related Work

Using some space filling curve like the **z-order** curve



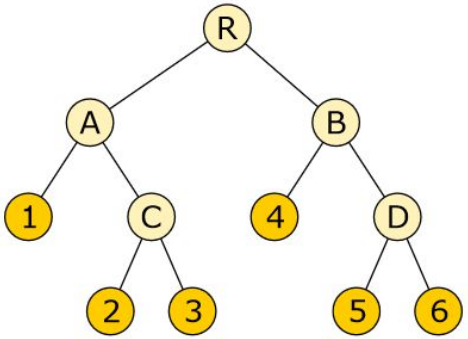
LBVH, Maximizing Parallelism in the Construction of BVHs, Octrees and k-d Trees [PJH16]

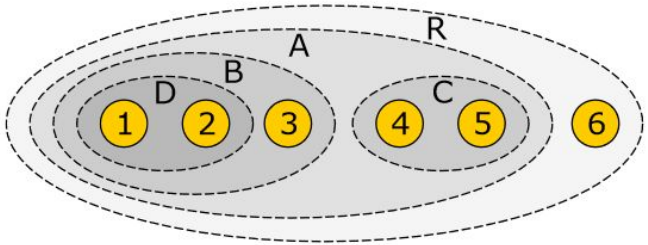
Related Work

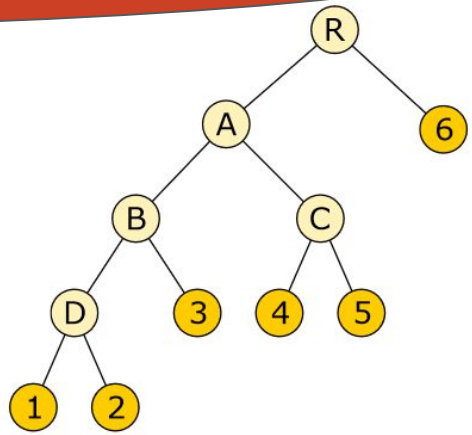


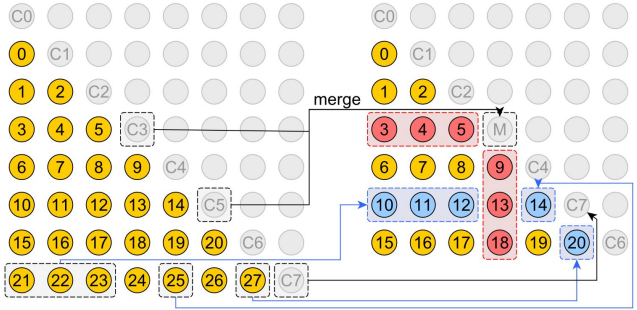
LBVH, Maximizing Parallelism in the Construction of BVHs, Octrees and k-d Trees [PJH16]

Related Work







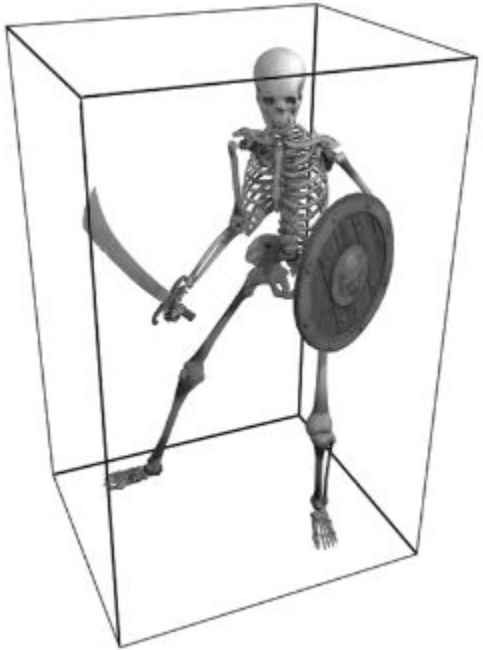


ATRBVH, Domingues and Pedrini et al. [PJH16]

Introduction

Tighter fitting
- Less wasteful intersections

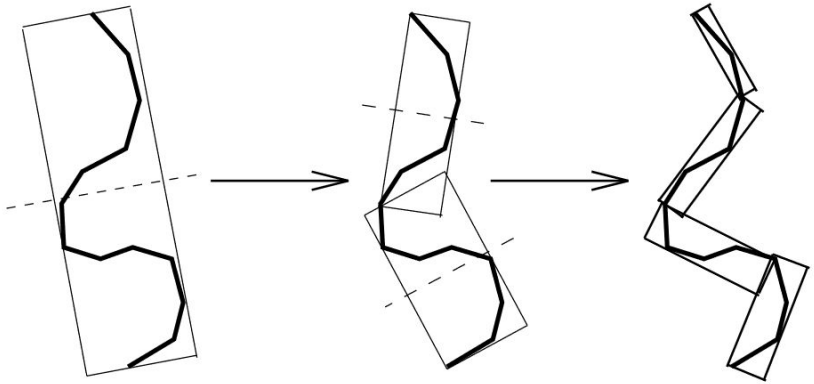
Robust to transformation

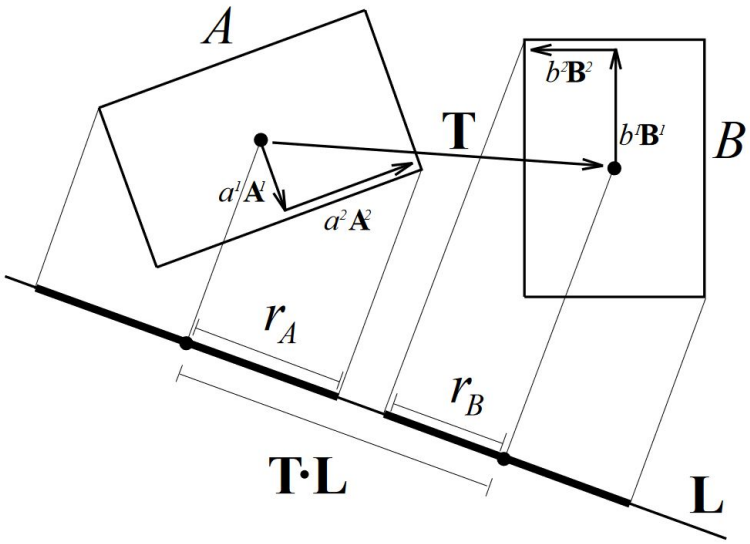


AABB

OBB

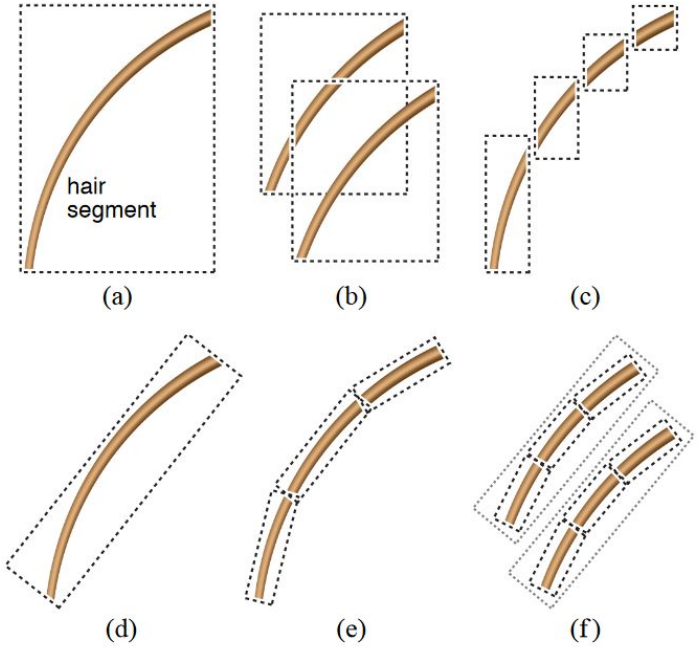
Related Work





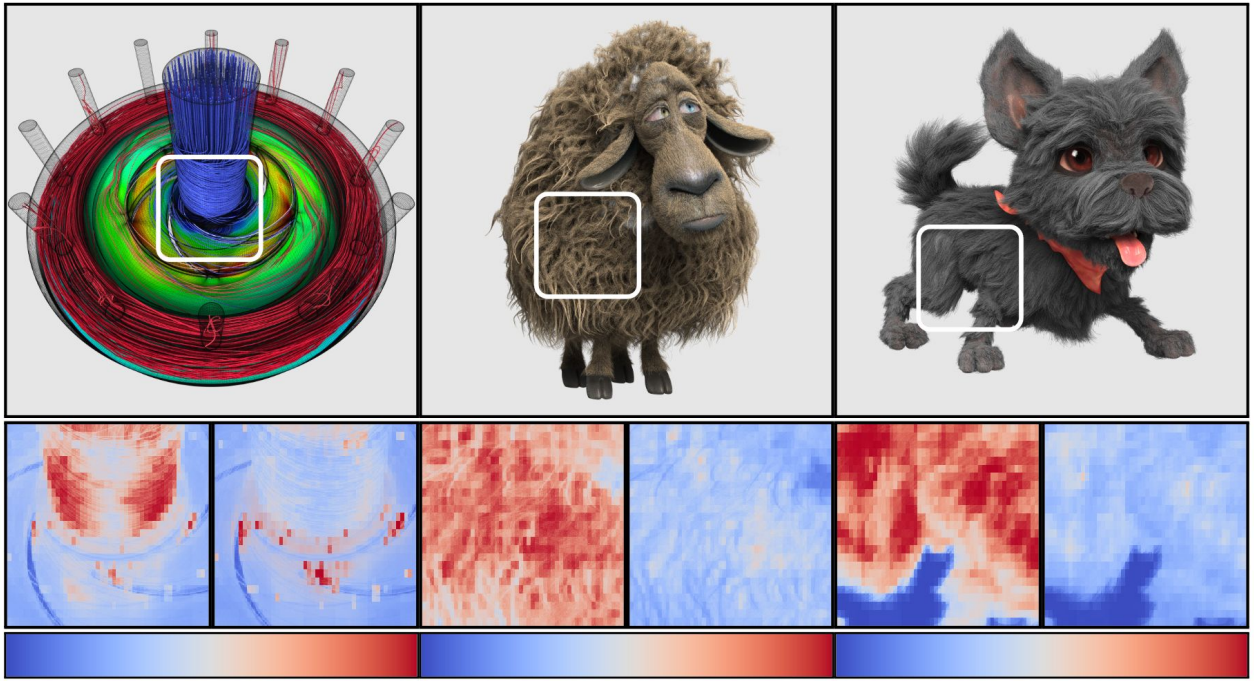
OBBTree A Hierarchical Structure for
Rapid Interference Detection, Gottschalk et al.

Related Work



Exploiting Local Orientation Similarity for
Efficient Ray Traversal of Hair and Fur, Woop et al.

Related Work



Using Hardware Ray Transforms to Accelerate
Ray/Primitive Intersections for Long, Thin Primitive Types, Wald et al.

Idea

We know that OBBs offer tighter fitting.

We have very fast algorithms for building AABB hierarchies.

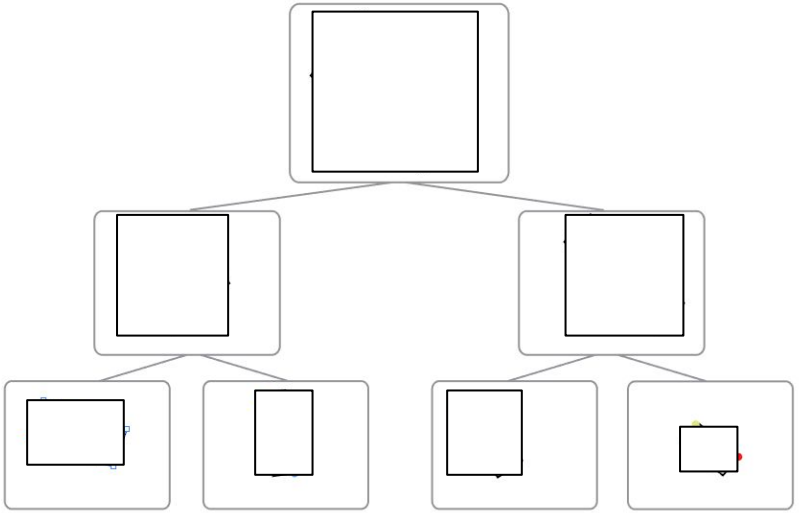
Idea

We know that OBBs offer tighter fitting.

We have very fast algorithms for building AABB hierarchies.

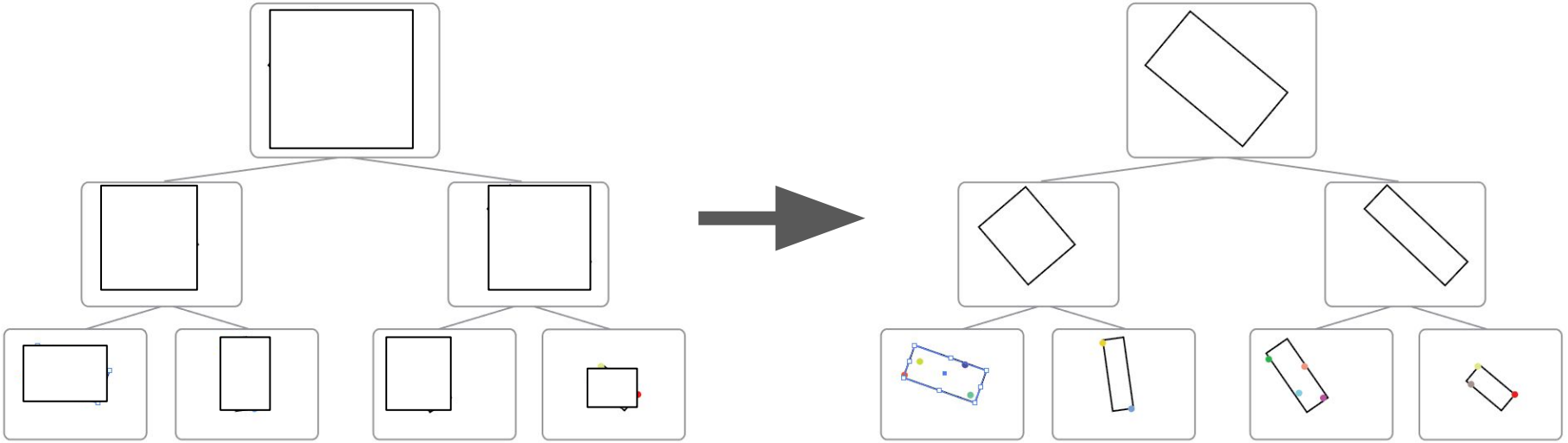
Can we adapt an OBB extraction algorithm to work fast on an existing tree hierarchy?
Can we minimally adapt one of the existing traversal kernels for improved ray tracing performance?

Introduction



AABB Hierarchy

Introduction



AABB Hierarchy

OBB Hierarchy

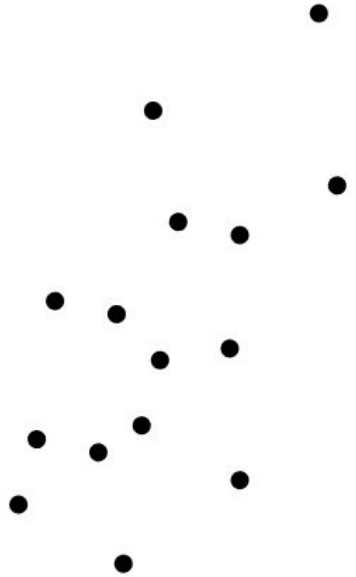
Di-tetrahedron OBB algorithm (DiTO)

Point clouds, polygon meshes, or polygon soups e.t.c.

Fast

Approximate

Robust



Larsson and Källberg, Fast computation of tight-fitting oriented bounding boxes, Game Engine Gems 2

Di-tetrahedron OBB algorithm (DiTO)

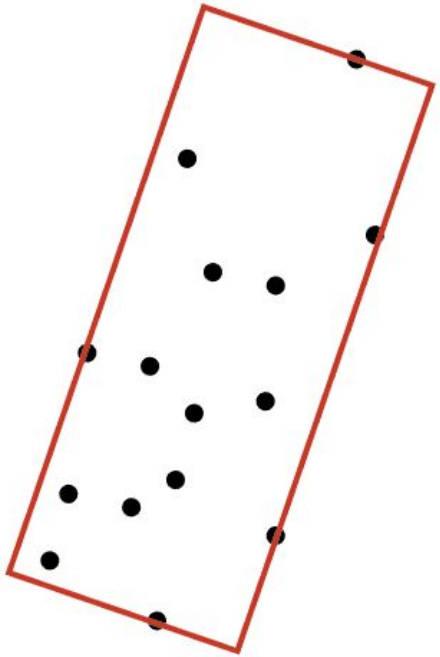
Point clouds, polygon meshes, or polygon soups e.t.c.

Fast

Approximate

Robust

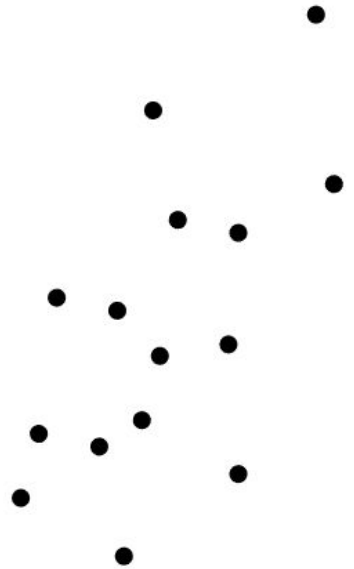
Parallelizable



Larsson and Källberg, Fast computation of tight-fitting oriented bounding boxes, Game Engine Gems 2

DiTO - Projection Extrema

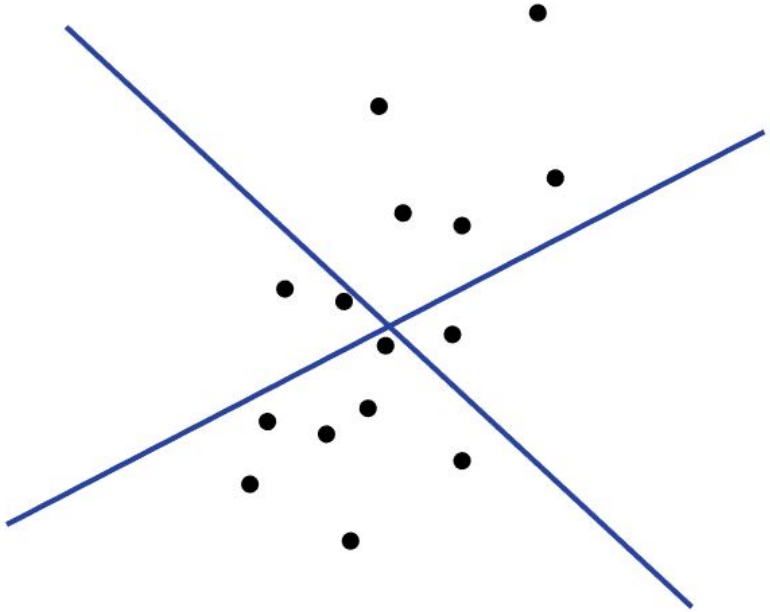
Consider a point set **P**



DiTO - Projection Extrema

Consider a point set **P**

Select **K** (7) standard axes



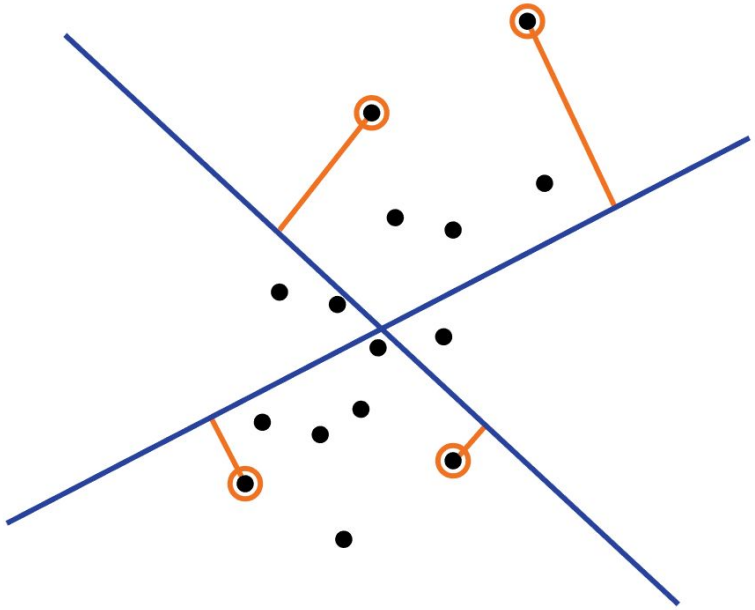
DiTO - Projection Extrema

Consider a point set **P**

Select **K** (7) standard axes

Find subset **S** of min and max projections for all **K** axes

- Values
- Points



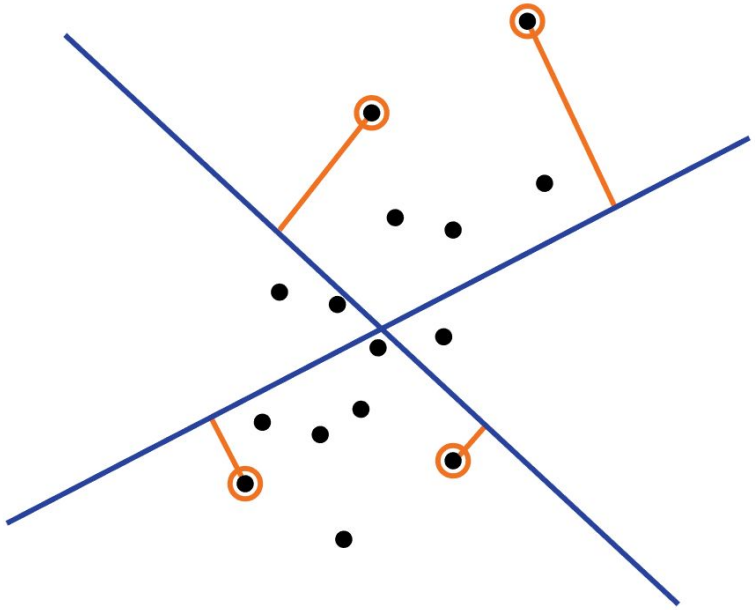
DiTO - Projection Extrema

Consider a point set **P**

Select **K** (7) standard axes

Find subset **S** of min and max projections for all **K** axes
- Values
- Points

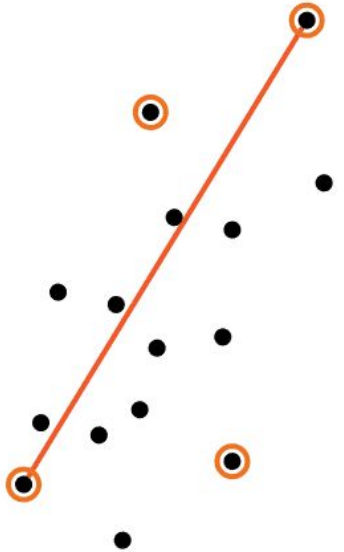
Typically, **S << P**



DiTO - Base Triangle

Construct the **large base triangle**

- From points in **S**

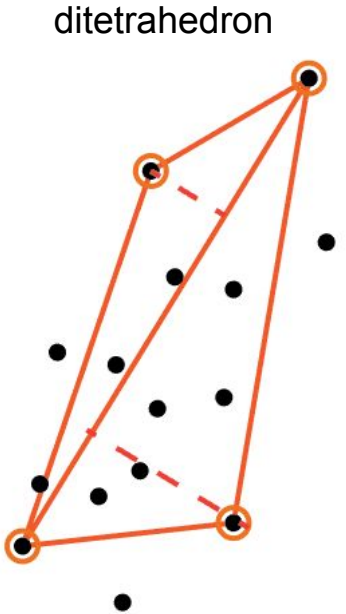


large base triangle

DiTO - Base Triangle

Construct the **large base triangle**
- From points in **S**

Construct the **ditetrahedron**
- From points in **S** and the base triangle



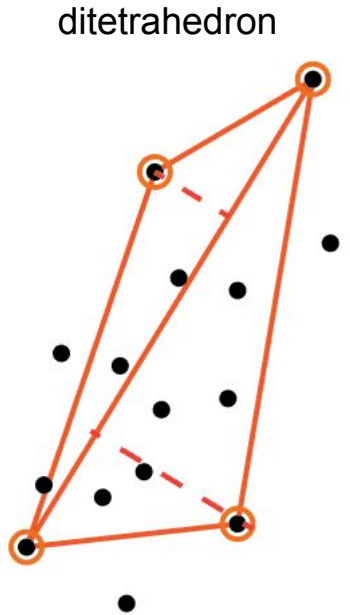
DiTO - Base Triangle

Construct the **large base triangle**
- From points in **S**

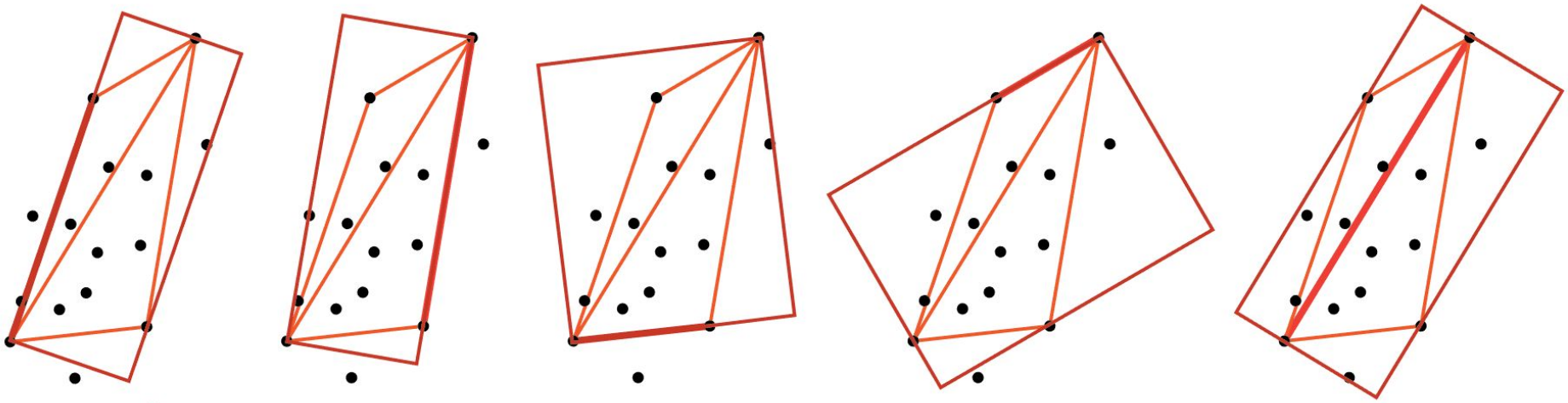
Construct the **ditetrahedron**
- From points in **S** and the base triangle

Construct candidate OBBs

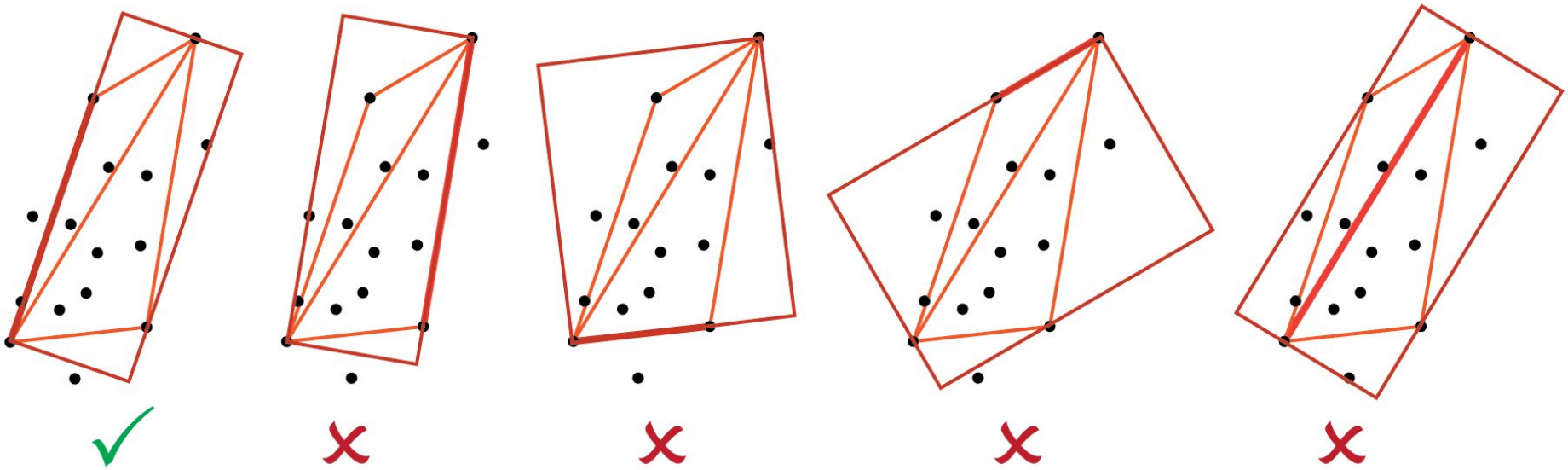
Using the edges and normals of all the constructed triangles we can find very good candidate OBBs.



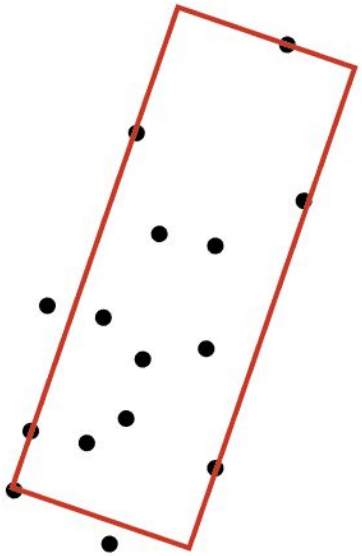
DiTO - Candidate OBBs



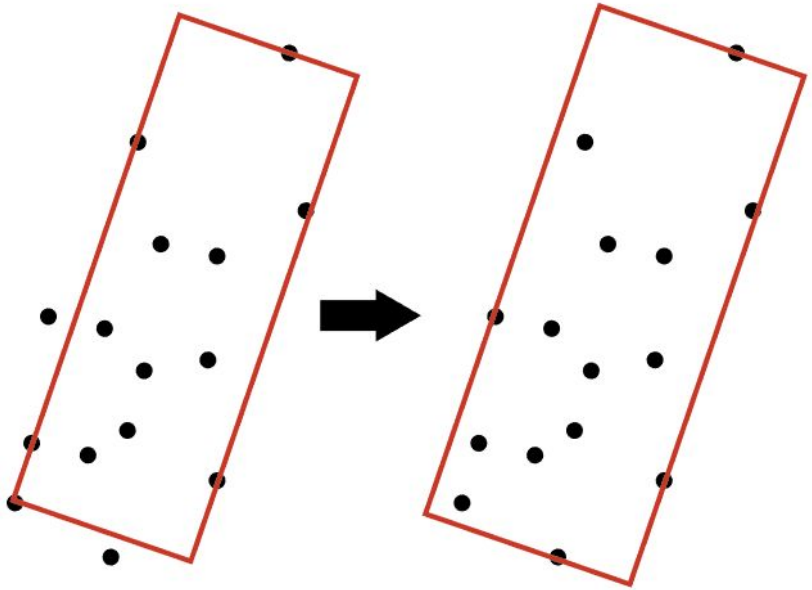
DiTO - Candidate OBBs



DiTO - OBB Refitting



DiTO - OBB Refitting



Parallel OBB Calculation

Parallel min/max scan operation over the **K** axes to find **S**

- Values
- Points

Construct and evaluate candidate OBBs

- Keep the one with the smallest surface area

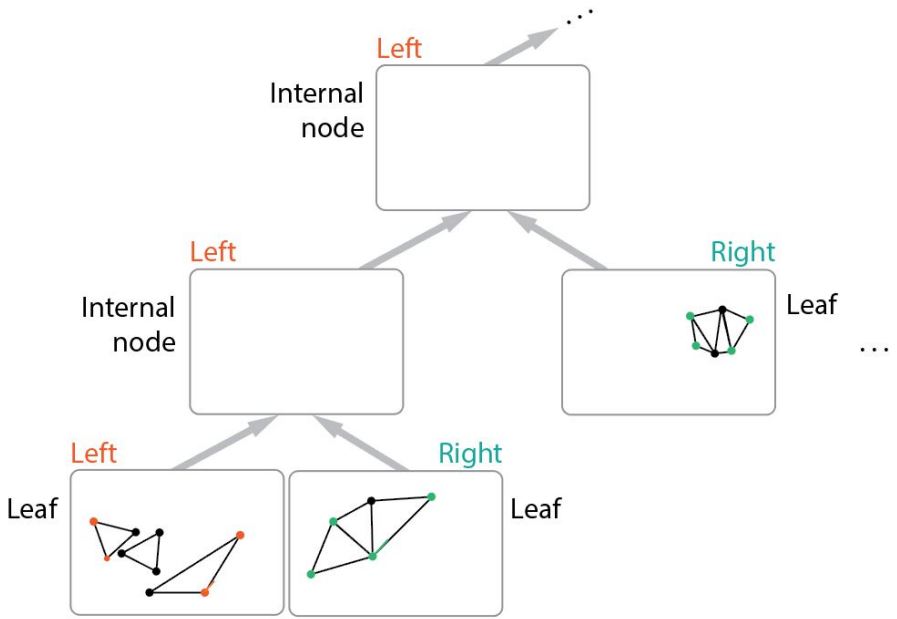
Parallel min/max scan operation over the **3** axes of the best OBB candidate on **P**

- We could end up with an AABB

Parallel Transformation

Prerequisites

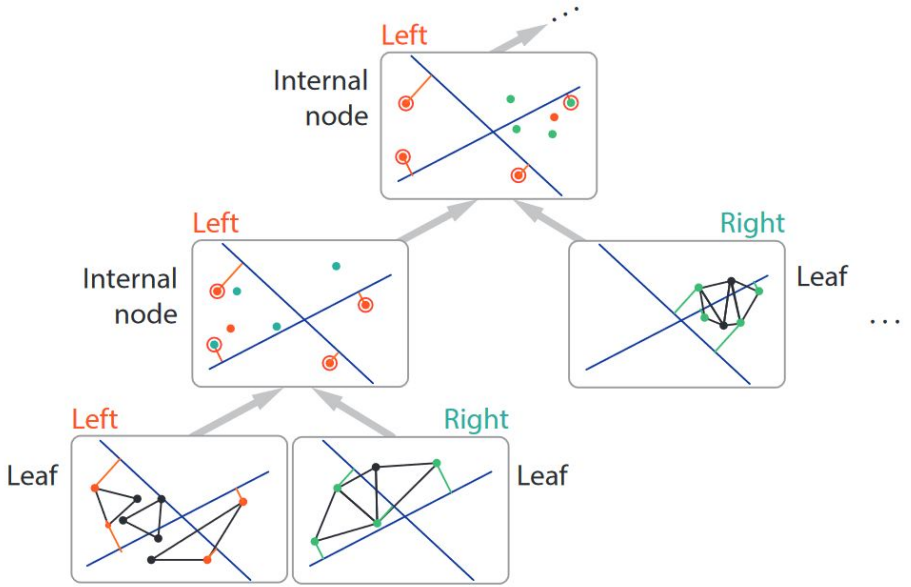
- **Parent** information
- Access to vertex data
- Not only triangles



Parallel Transformation

Key insights

- Nodes only need to know the **K** min/nax vertices of their subtrees
- Agglomerative approach



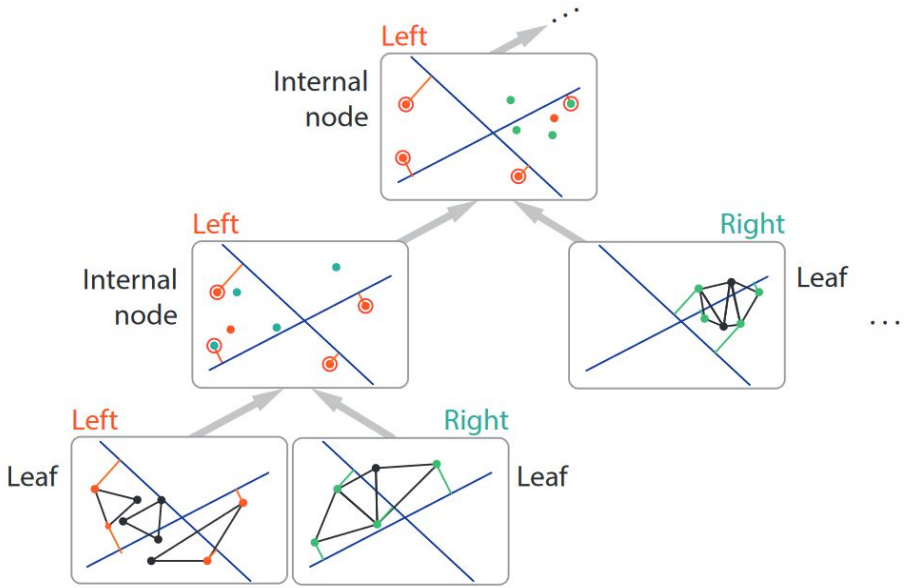
Parallel Transformation - Projection Extrema

Parallel bottom-up traversal

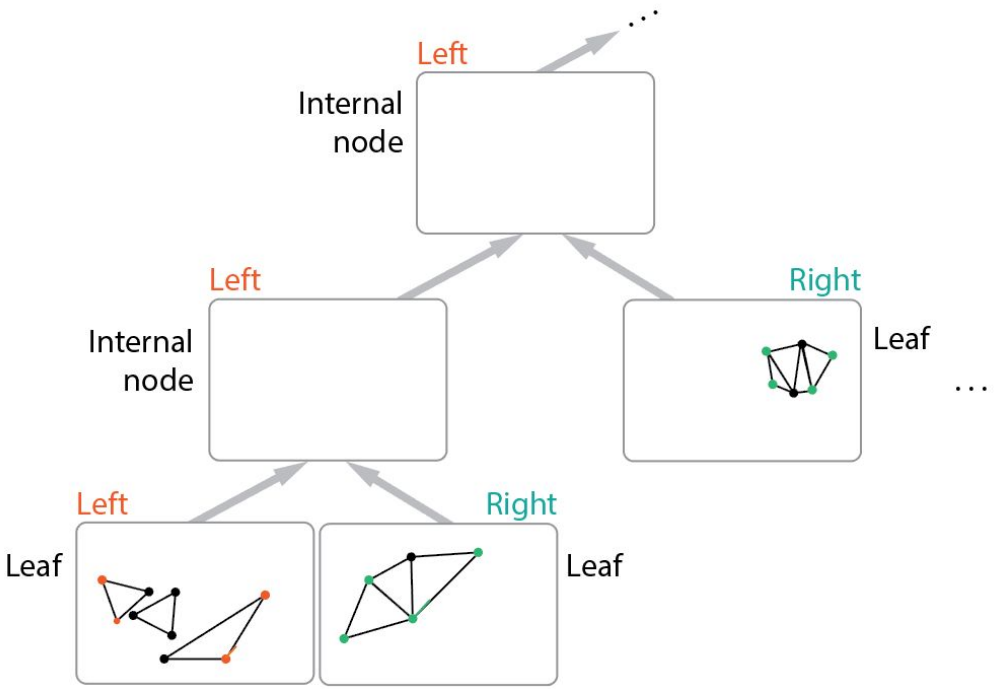
Find the **K** min/max projections at the leaves in parallel
- Stored in memory

Follow uplinks to parent nodes
- Only <u>one</u> of the children reach the parent

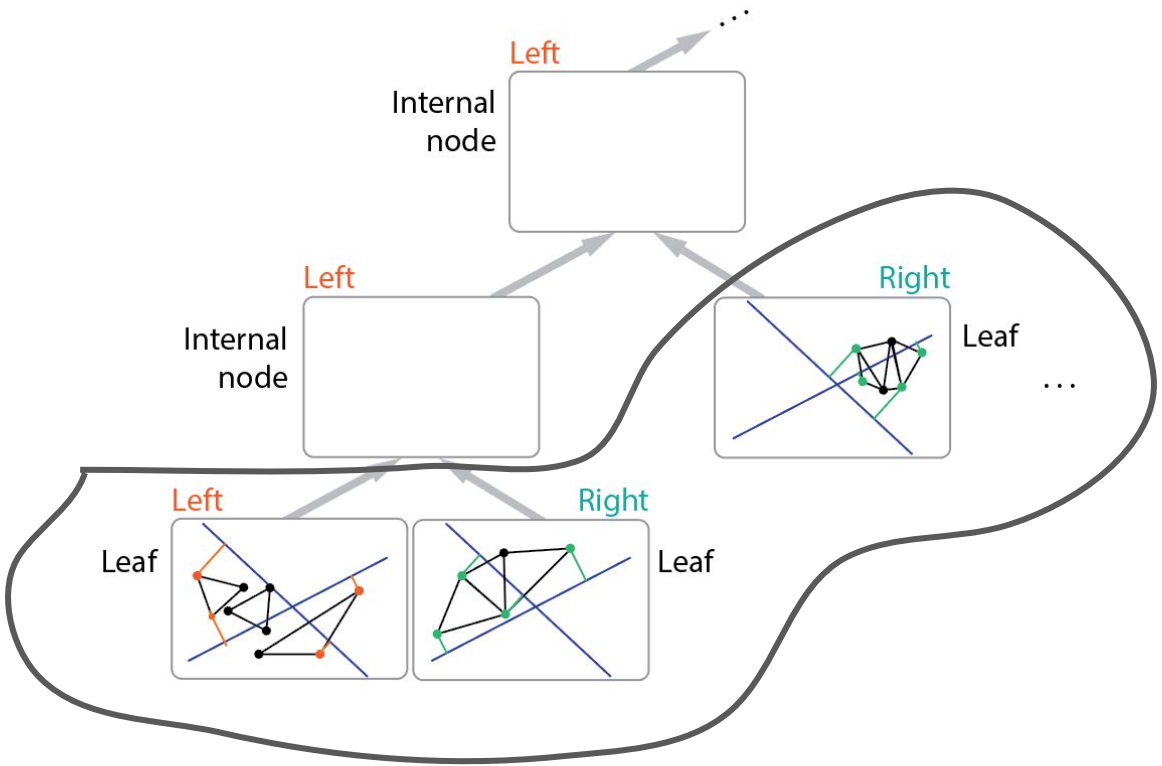
Find node's projection extrema from its children



Parallel Transformation - Projection Extrema



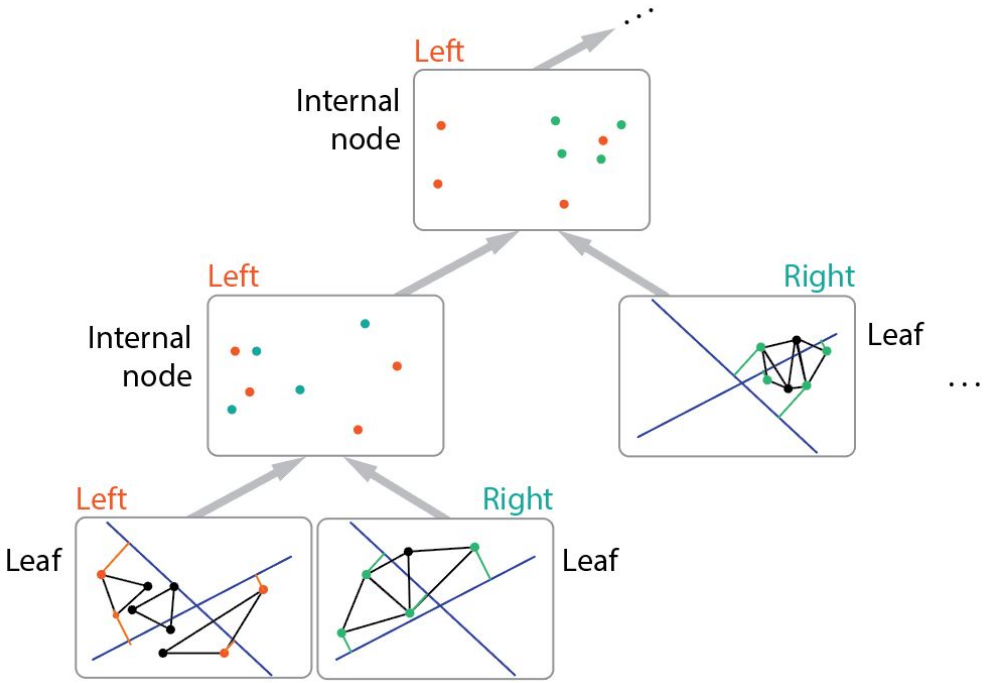
Parallel Transformation - Projection Extrema



N-threads

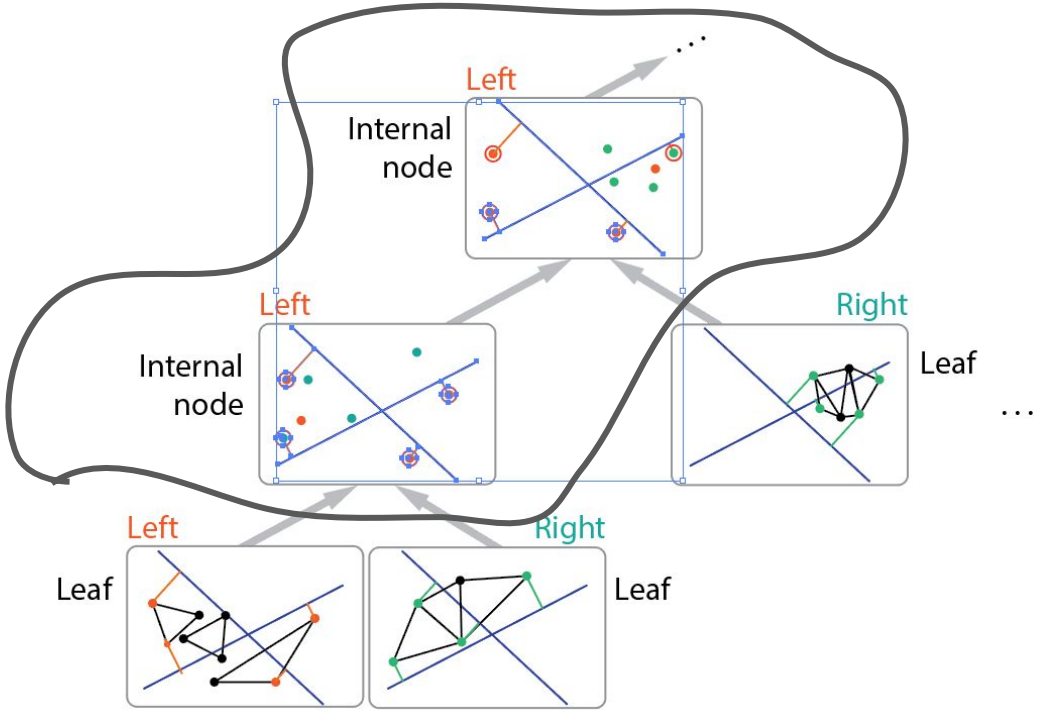
Parallel Transformation - Projection Extrema

Store results to memory



Parallel Transformation - Projection Extrema

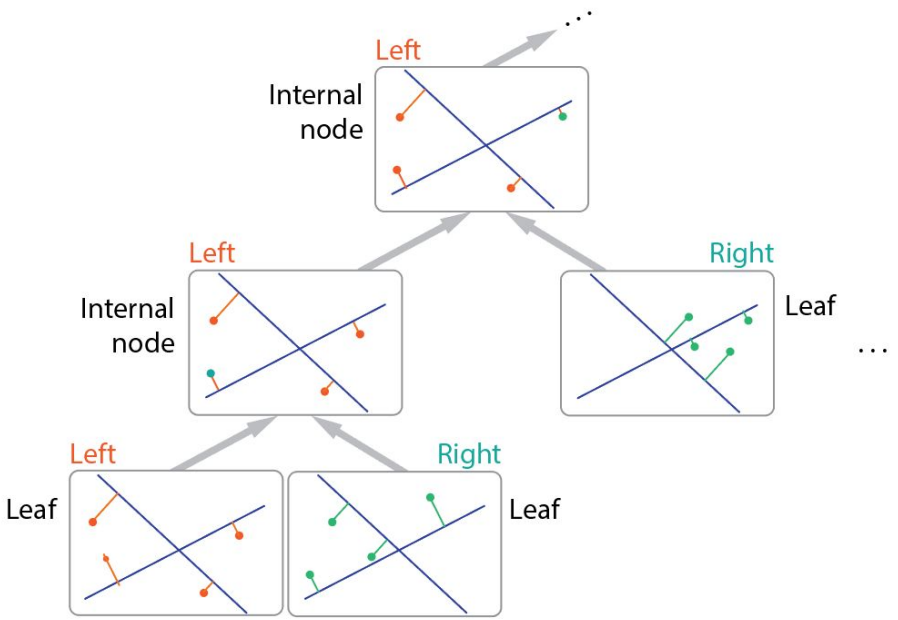
N/2-threads



Parallel Transformation - Node OBB candidates

For each node in parallel

Calculate candidate OBBs independently for each node



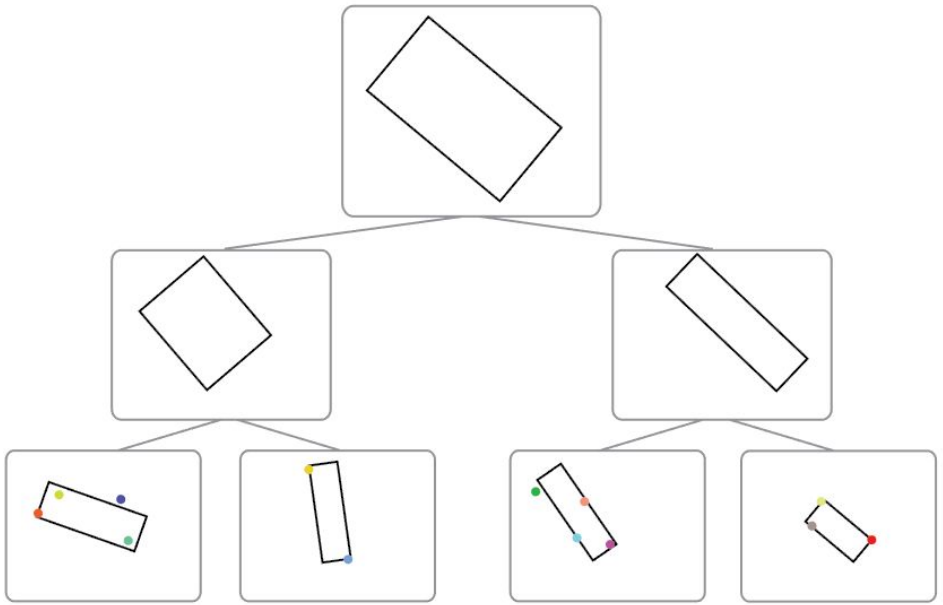
Parallel Transformation - Node OBB candidates

For each node in parallel

Calculate candidate OBBs independently for each node

For the best candidate store in global memory

- Axes
- Extents



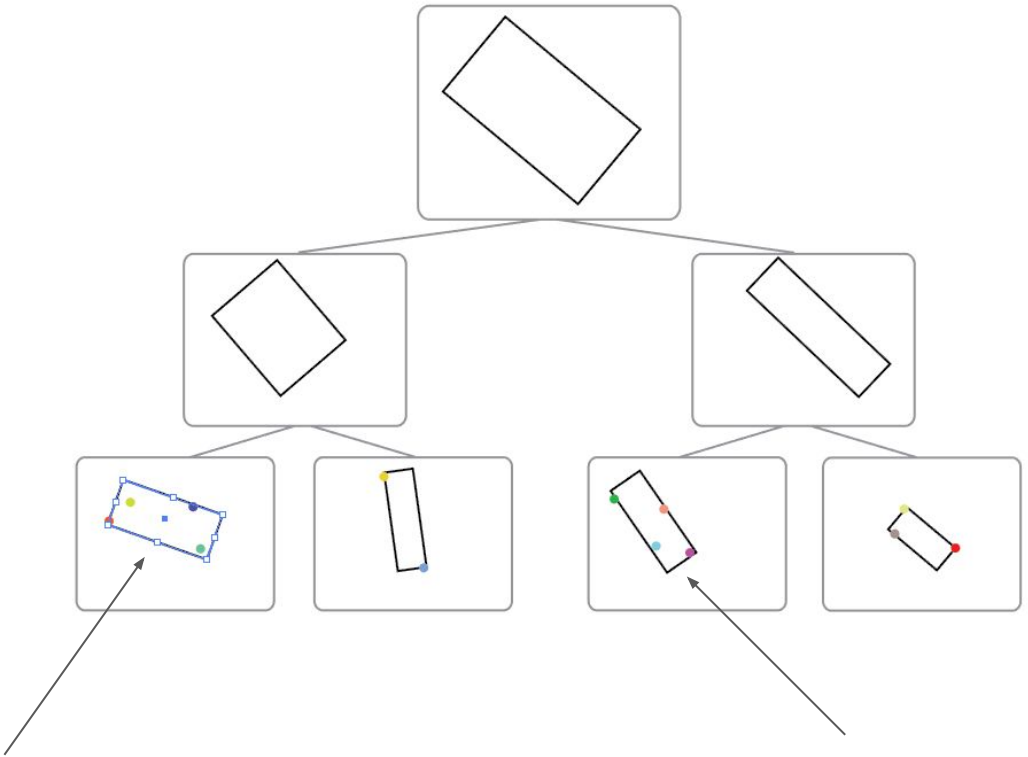
Parallel Transformation - OBB refitting

Data parallel bottom-up traversal

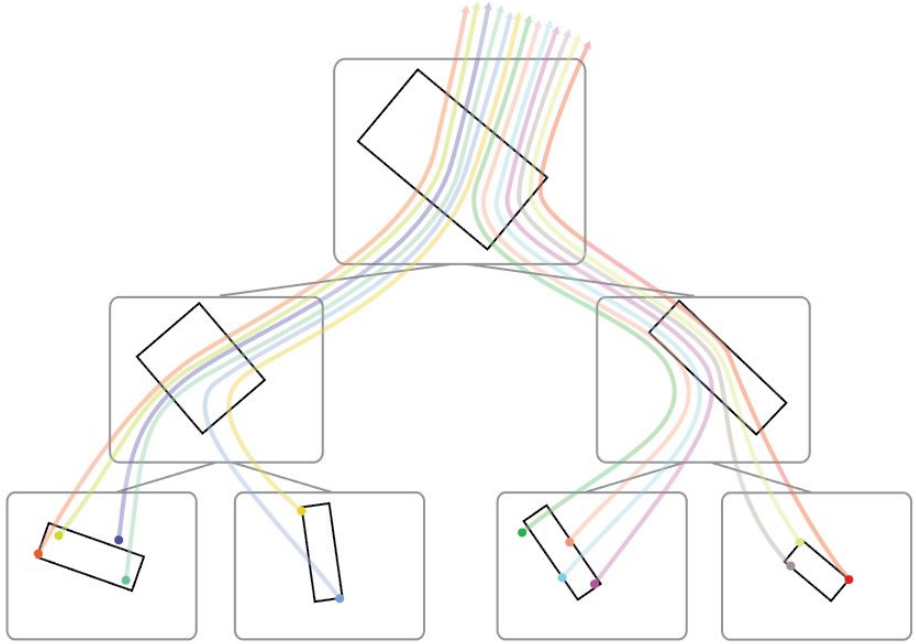
Ensure all nodes fully enclose the primitives of their subtree

Where needed, atomically update

- Axes
- Extents

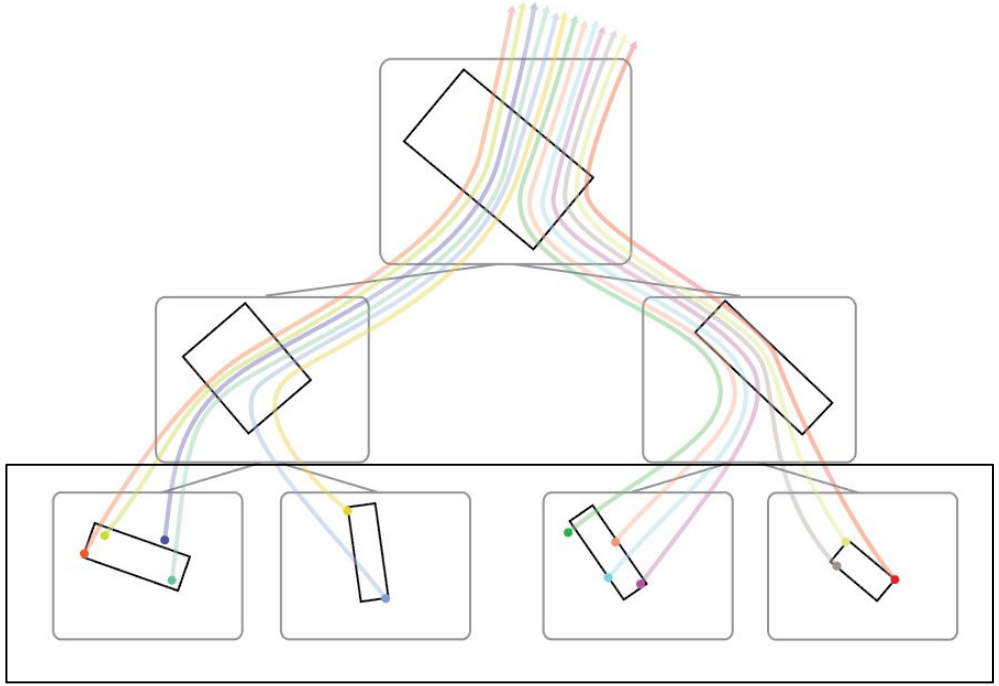


Parallel Transformation - OBB refitting



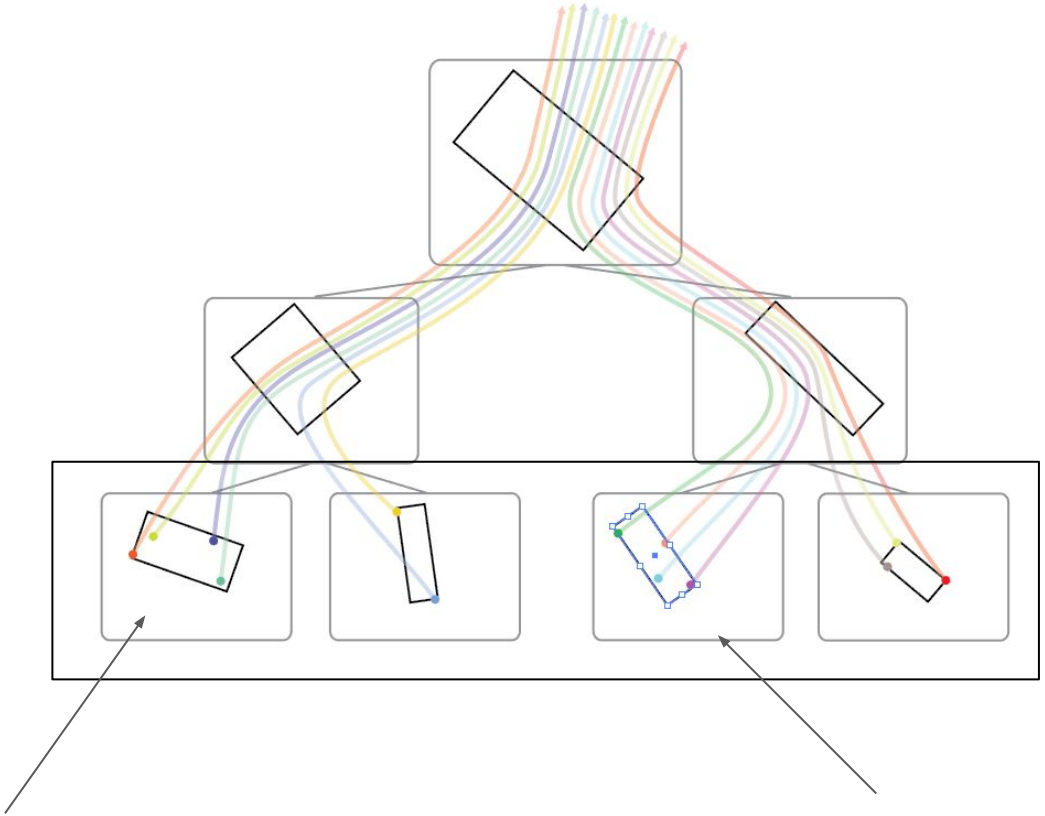
Parallel Transformation - OBB refitting

N-threads



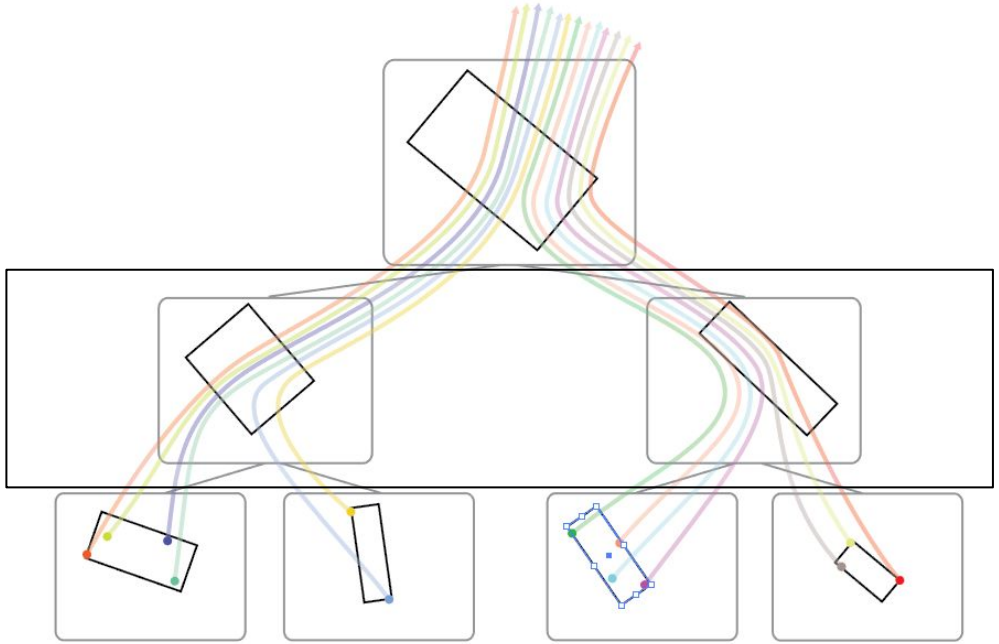
Parallel Transformation - OBB refitting

N-threads



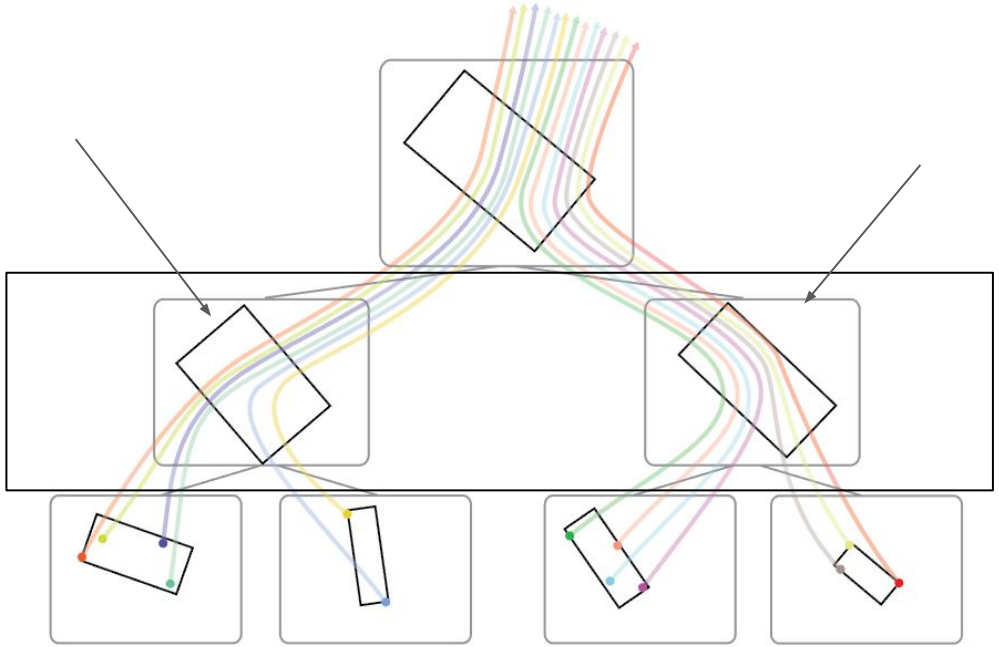
Parallel Transformation - OBB refitting

N-threads



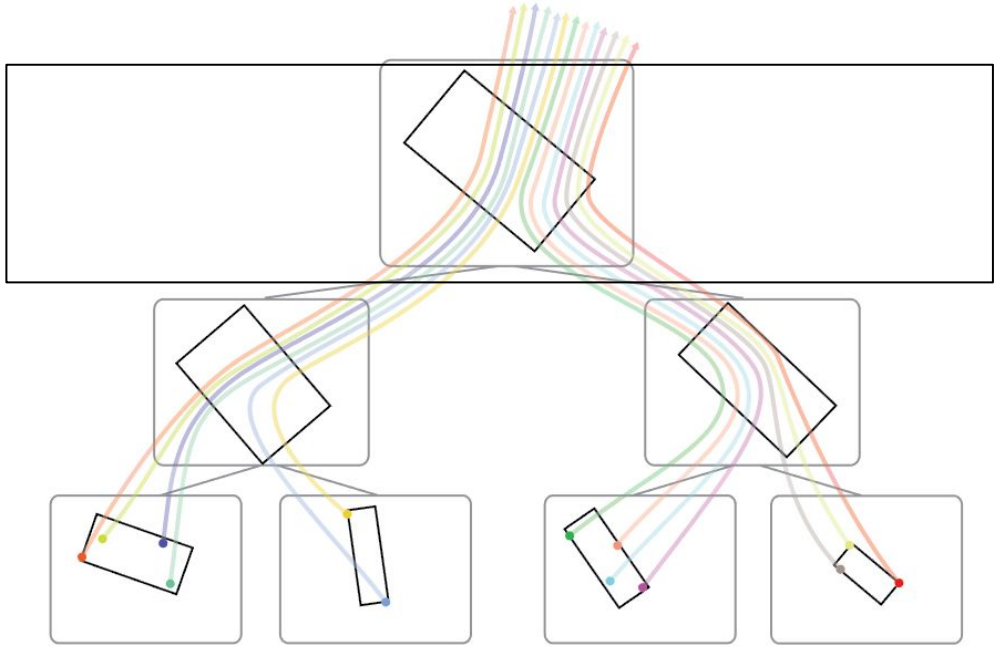
Parallel Transformation - OBB refitting

N-threads



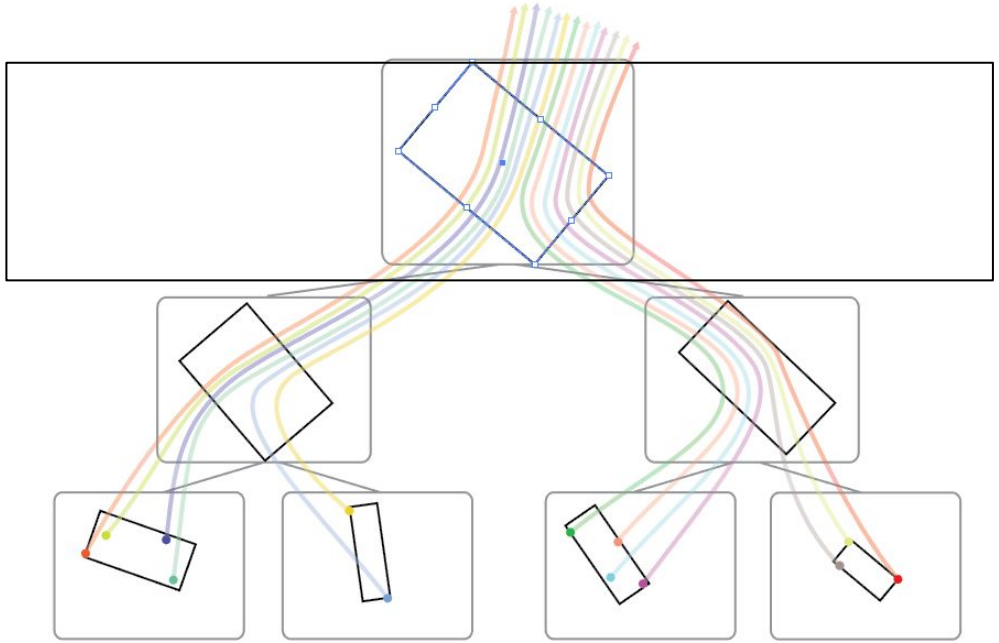
Parallel Transformation - OBB refitting

N-threads

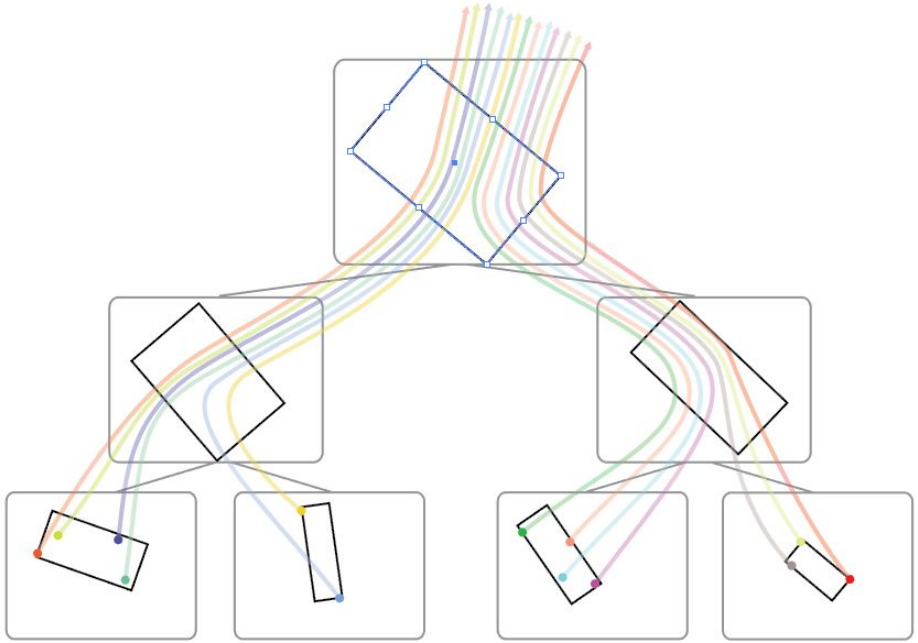


Parallel Transformation - OBB refitting

N-threads



Parallel Transformation - OBB refitting



Tree Traversal

Persistent while-while loop

Struct-Of-Arrays node layout

- Parents hold intersection data for their children
- min and max for AABBs
- 4x3 inverse transform for OBBs

Leaf nodes only contain the corresponding range of the primitive data

Understanding the efficiency of ray traversal on GPUs, Aila and Laine

Tree Traversal

```
while rays available do
    Fetch ray
    while ray not terminated do
        while internal node do
            Fetch AABBs of children
            Intersect AABBs
        end
        while leaf node do
            Fetch primtive information
            Intersect
        end
    end
end
```

```
while rays available do
    Fetch ray
    while ray not terminated do
        while internal node do
            Fetch transformation matrix of children
            Transform ray
            Intersect unit AABB
        end
        while leaf node do
            Fetch primtive information
            Intersect
        end
    end
end
```

Fast and Robust Ray/OBB Intersection Using the Lorentz Transformation, Sabino, Rodolfo, et al

Tree Traversal

```
while rays available do
    Fetch ray
    while ray not terminated do
        while internal node do
            Fetch AABBs of children
            Intersect AABBs
        end
        while leaf node do
            Fetch primtive information
            Intersect
        end
    end
end
```

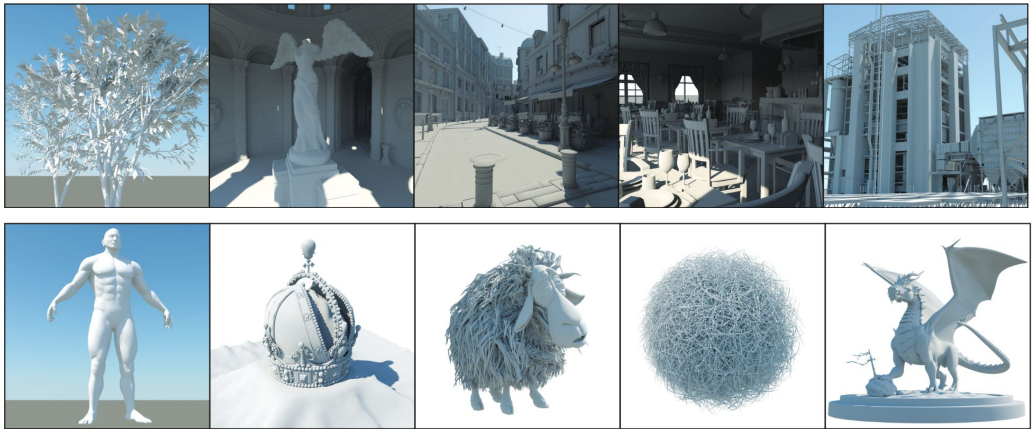
```
while rays available do
    Fetch ray
    while ray not terminated do
        while internal node do
            Fetch transformation matrix of children
            Transform ray
            Intersect unit AABB
        end
        while leaf node do
            Fetch primtive information
            Intersect
        end
    end
end
```

Fast and Robust Ray/OBB Intersection Using the Lorentz Transformation, Sabino, Rodolfo, et al

Evaluation

Standalone OBB calculation on unordered point sets

OBB Trees for ray tracing



Hardware

- NVIDIA RTX 3080Ti 12GB
- Intel i7 12700K CPU

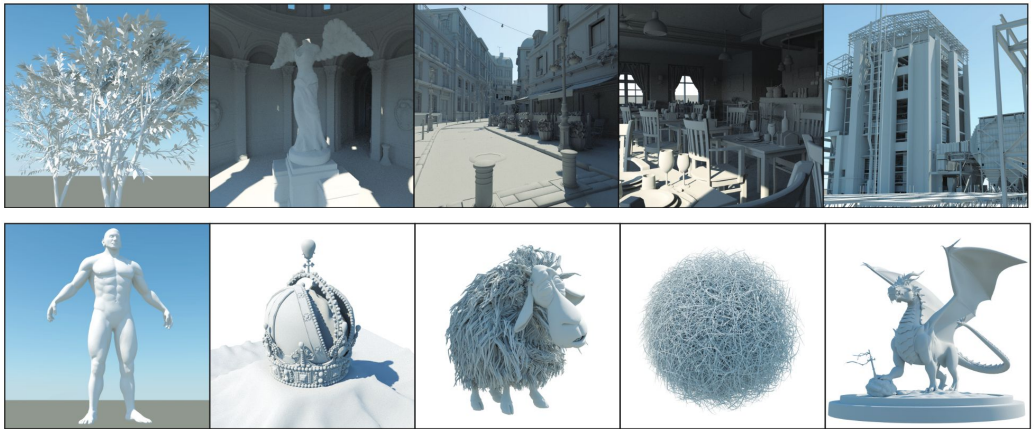
Evaluation

Scenes

- Single object scenes
- Non-hierarchical/flat scenes

Tree transformation

- Transformation overhead
- Ray tracing
  - Primary rays
  - Secondary rays
  - Time-to-render



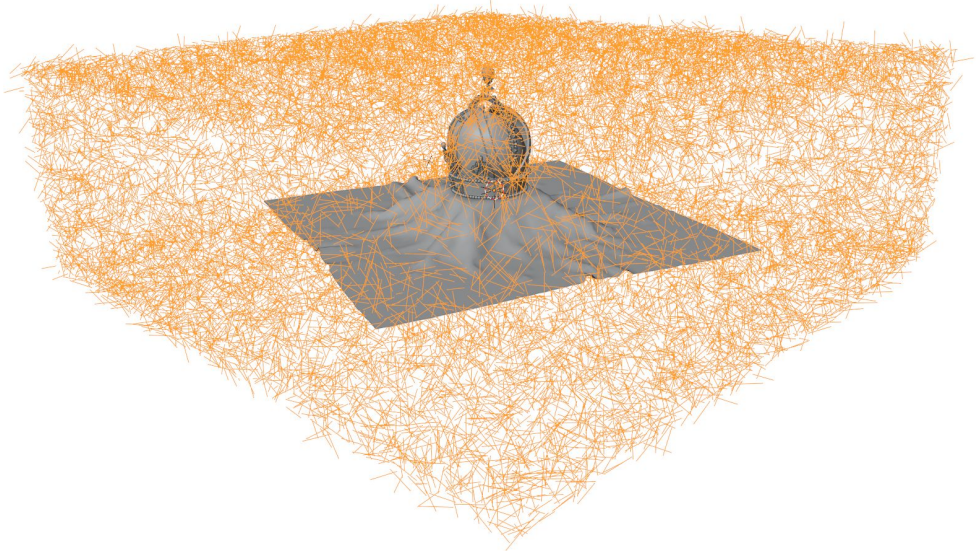
Evaluation

Scenes

- Single object scenes
- Non-hierarchical/flat scenes

Tree transformation

- Transformation overhead
- Ray tracing
  - Primary rays
  - Secondary rays
  - Time-to-render
  - Bounding Volume rays

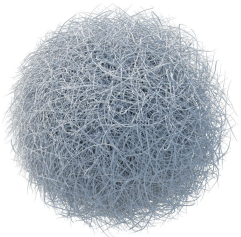


Evaluation - OBB calculation on the GPU

Scene	Point Count	Timings (ms)	
		Reference	Ours
Trees	270K	1.88	0.28 (6.71×)
Human	370K	2.38	0.3 (7.93×)
Dragon	420K	2.68	0.33 (8.12×)
Hairball	1.5M	9.21	0.55 (16.74×)
Crown	1.8M	11.66	0.67 (17.19×)
Sheep	23.3M	149.23	5.65 (26.41×)








Evaluation - Builders

Builders

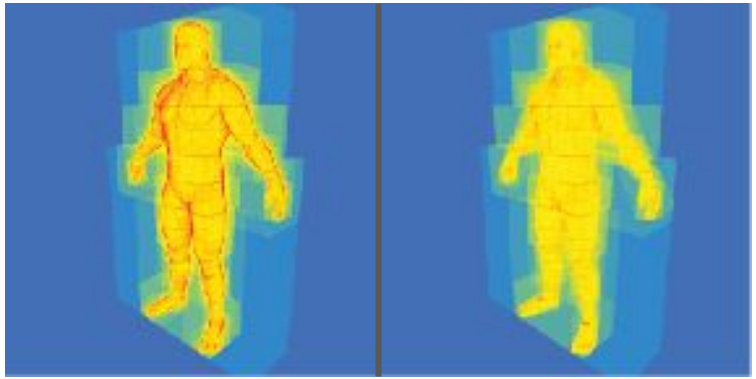
- **LBVH**
- **ATRBVH**
- **Sweep SAH BVH (SwpBVH)**

Options

- Node collapsing (up to 8 primitives per leaf)
- Default options of each builder

Results

Human (1.2M triangles)

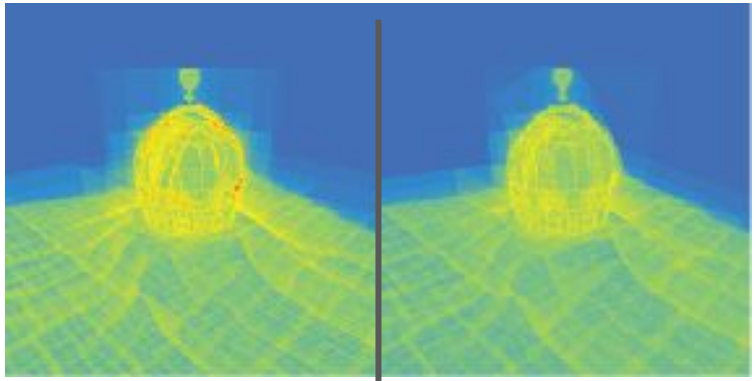


Conversion Time 5.92ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	1.07 / **0.96** (1.11x)
	ATRBVH	1.00 / **0.90** (1.10x)
	swpBVH	0.94 / **0.75** (1.25x)
Time (ms) Secondary rays	LBVH	3.06 / **1.92** (1.59x)
	ATRBVH	2.76 / **1.78** (1.54x)
	swpBVH	2.27 / **1.47** (1.54x)
Time (ms) Random BV rays	LBVH	1.64 / **1.59** (1.03x)
	ATRBVH	1.54 / **1.48** (1.04x)
	swpBVH	1.28 / **1.17** (1.09x)
Avg. intersections swpBVH		5.99 / **4.38** (1.36x)
Time (sec.) to render	LBVH	2.11 / **1.75** (1.21x)
	ATRBVH	1.93 / **1.62** (1.19x)

Results

Crown (3.5M triangles)

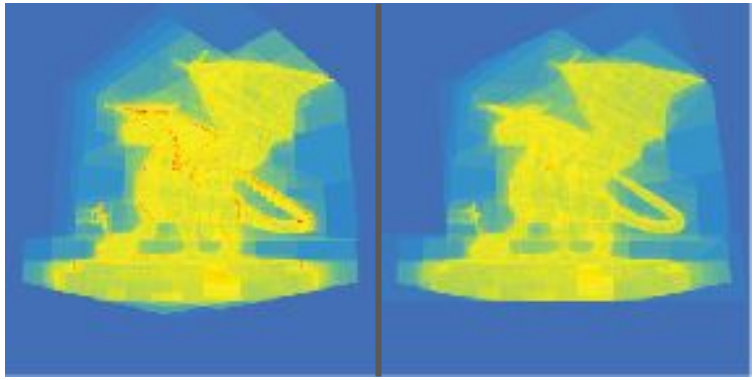


Conversion Time 16.85ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	2.56 / **1.91** (1.33x)
	ATRBVH	1.81 / **1.42** (1.27x)
	swpBVH	1.88 / **1.35** (1.39x)
Time (ms) Secondary rays	LBVH	17.18 / **13.75** (1.24x)
	ATRBVH	13.39 / **10.69** (1.25x)
	swpBVH	12.66 / **9.66** (1.31x)
Time (ms) Random BV rays	LBVH	1.10 / **1.01** (1.09x)
	ATRBVH	0.94 / **0.84** (1.11x)
	swpBVH	0.90 / **0.77** (1.17x)
Avg. intersections swpBVH		8.53 / **6.05** (1.40x)
Time (sec.) to render	LBVH	6.82 / **5.81** (1.17x)
	ATRBVH	5.06 / **4.07** (1.24x)

Results

Dragon (800K triangles)

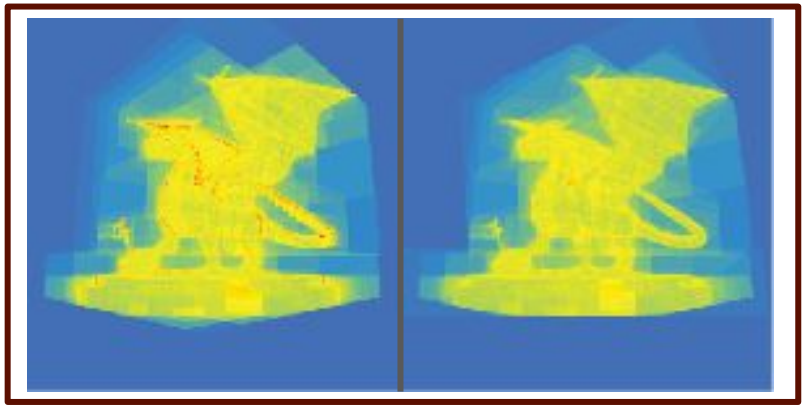


Conversion Time 4.64ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	1.23 / **0.91** (1.35x)
	ATRBVH	1.01 / **0.74** (1.36x)
	swpBVH	1.01 / **0.69** (1.46x)
Time (ms) Secondary rays	LBVH	5.64 / **4.34** (1.3x)
	ATRBVH	4.47 / **3.58** (1.24x)
	swpBVH	4.16 / **3.21** (1.29x)
Time (ms) Random BV rays	LBVH	1.64 / **1.46** (1.12x)
	ATRBVH	1.26 / **1.18** (1.06x)
	swpBVH	1.18 / **1.07** (1.10x)
Avg. intersections swpBVH		15.15 / **12.01** (1.26x)
Time (sec.) to render	LBVH	2.56 / **1.93** (1.33x)
	ATRBVH	2.05 / **1.56** (1.32x)

Results

Dragon (800K triangles)

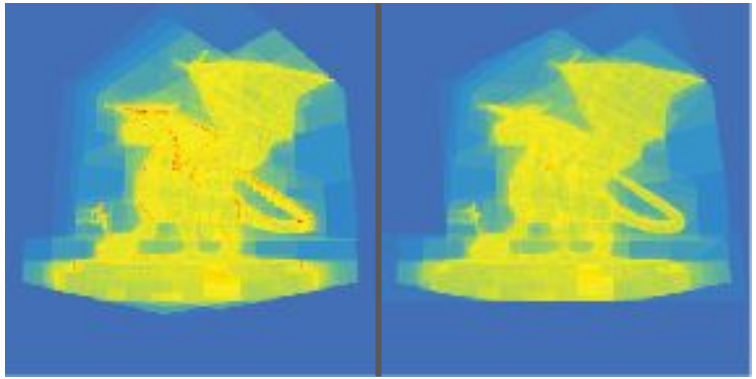


Conversion Time 4.64ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	1.23 / **0.91** (1.35x)
	ATRBVH	1.01 / **0.74** (1.36x)
	swpBVH	1.01 / **0.69** (1.46x)
Time (ms) Secondary rays	LBVH	5.64 / **4.34** (1.3x)
	ATRBVH	4.47 / **3.58** (1.24x)
	swpBVH	4.16 / **3.21** (1.29x)
Time (ms) Random BV rays	LBVH	1.64 / **1.46** (1.12x)
	ATRBVH	1.26 / **1.18** (1.06x)
	swpBVH	1.18 / **1.07** (1.10x)
Avg. intersections swpBVH		15.15 / **12.01** (1.26x)
Time (sec.) to render	LBVH	2.56 / **1.93** (1.33x)
	ATRBVH	2.05 / **1.56** (1.32x)

Results

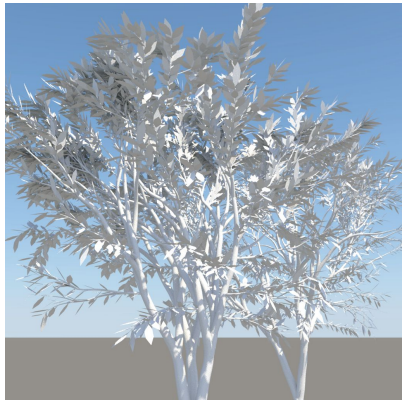
Dragon (800K triangles)



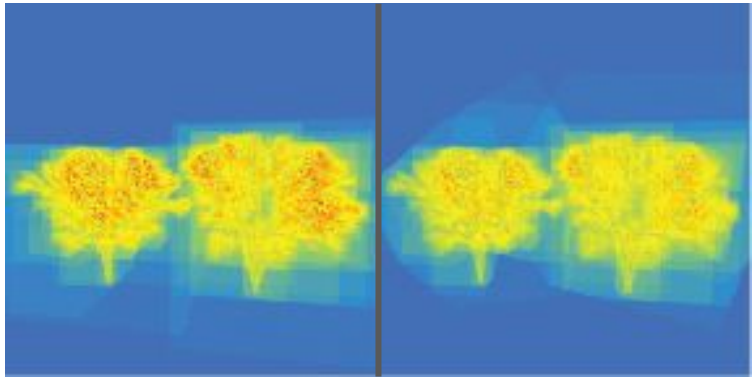
Conversion Time 4.64ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	1.23 / **0.91** (1.35x)
	ATRBVH	1.01 / **0.74** (1.36x)
	swpBVH	1.01 / **0.69** (1.46x)
Time (ms) Secondary rays	LBVH	5.64 / **4.34** (1.3x)
	ATRBVH	4.47 / **3.58** (1.24x)
	swpBVH	4.16 / **3.21** (1.29x)
Time (ms) Random BV rays	LBVH	1.64 / **1.46** (1.12x)
	ATRBVH	1.26 / **1.18** (1.06x)
	swpBVH	1.18 / **1.07** (1.10x)
Avg. intersections swpBVH		15.15 / **12.01** (1.26x)
Time (sec.) to render	LBVH	2.56 / **1.93** (1.33x)
	ATRBVH	2.05 / **1.56** (1.32x)

Results



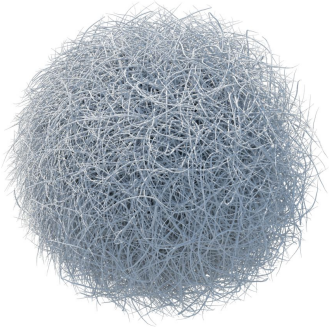
Trees (170K triangles)



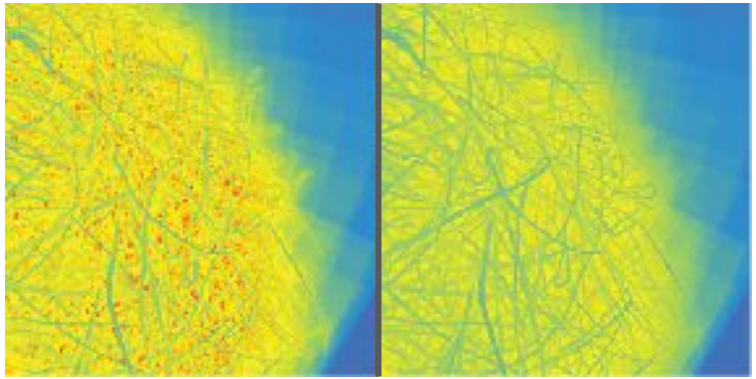
Conversion Time 0.97ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	3.38 / **2.94** (1.14x)
	ATRBVH	2.90 / **2.42** (1.19x)
	swpBVH	3.29 / **2.44** (1.34x)
Time (ms) Secondary rays	LBVH	15.22 / **12.89** (1.18x)
	ATRBVH	12.45 / **10.58** (1.17x)
	swpBVH	12.12 / **10.08** (1.2x)
Time (ms) Random BV rays	LBVH	2.62 / **2.17** (1.20x)
	ATRBVH	2.15 / **1.77** (1.21x)
	swpBVH	1.95 / **1.72** (1.13x)
Avg. intersections swpBVH		33.73 / **25.03** (1.34x)
Time (sec.) to render	LBVH	6.79 / **5.60** (1.21x)
	ATRBVH	5.59 / **4.55** (1.23x)

Results



Hairball (2.8M triangles)

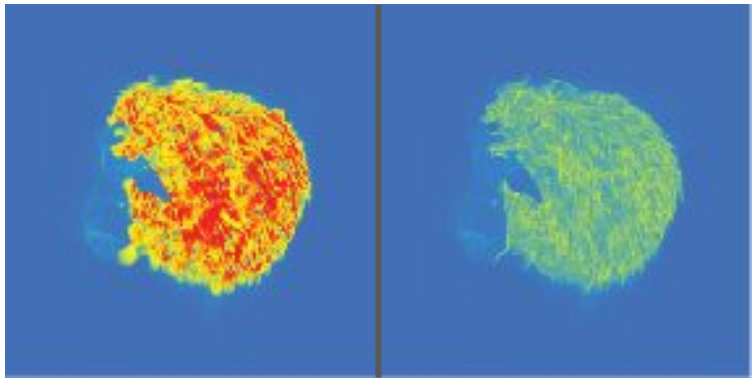


Conversion Time 12.54ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	7.91 / **5.13** (1.54x)
	ATRBVH	6.46 / **4.20** (1.53x)
	swpBVH	6.30 / **4.01** (1.57x)
Time (ms) Secondary rays	LBVH	9.41 / **8.79** (1.07x)
	ATRBVH	10.90 / **7.80** (1.39x)
	swpBVH	7.11 / **6.81** (1.04x)
Time (ms) Random BV rays	LBVH	5.26 / **4.57** (1.14x)
	ATRBVH	5.06 / **3.71** (1.36x)
	swpBVH	4.17 / **3.41** (1.22x)
Avg. intersections swpBVH		48.83 / **41.33** (1.18x)
Time (sec.) to render	LBVH	10.53 / **7.56** (1.39x)
	ATRBVH	8.81 / **6.34** (1.39x)

Results

Sheep (10.7M triangles)

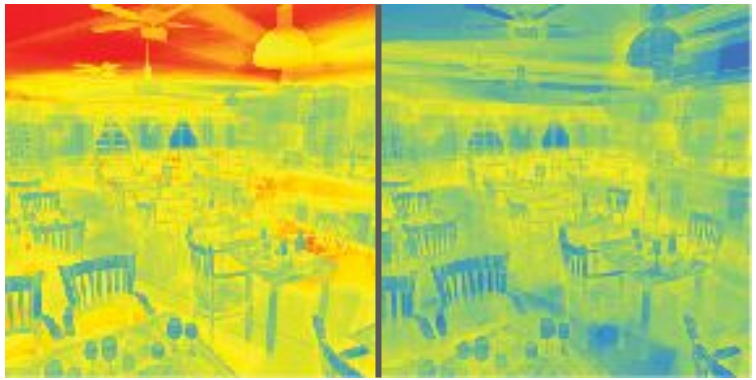


Conversion Time 44.91ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	147.82 / **15.25** (9.68x)
	ATRBVH	147.10 / **13.72** (10.71x)
	swpBVH	134.43 / **11.63** (11.55x)
Time (ms) Secondary rays	LBVH	518.08 / **41.95** (12.34x)
	ATRBVH	519.10 / **37.43** (13.86x)
	swpBVH	476.64 / **30.87** (15.44x)
Time (ms) Random BV rays	LBVH	127.82 / **19.77** (6.46x)
	ATRBVH	123.45 / **17.46** (7.06x)
	swpBVH	107.10 / **13.74** (7.79x)
Avg. intersections swpBVH		1282.80 / **128.36** (9.91x)
Time (sec.) to render	LBVH	268.81 / **25.86** (10.39x)
	ATRBVH	264.32 / **23.09** (11.45x)

Results

Bistro Interior (1M triangles)



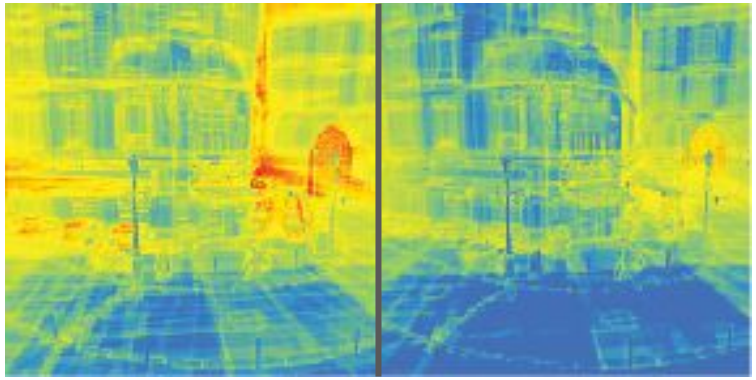
Conversion Time 4.84ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	6.72 / **4.78** (1.40x)
	ATRBVH	4.62 / **3.48** (1.32x)
	swpBVH	4.77 / **3.15** (1.51x)
Time (ms) Secondary rays	LBVH	137.61 / **70.83** (1.94x)
	ATRBVH	103.70 / **48.28** (2.14x)
	swpBVH	89.69 / **83.15** (1.07x)

Avg. intersections swpBVH		126.53 / **57.68** (2.19x)
Time (sec.) to render	LBVH	28.91 / **16.60** (1.74x)
	ATRBVH	21.22 / **11.79** (1.80x)

Results

Bistro Exterior (2.8M triangles)



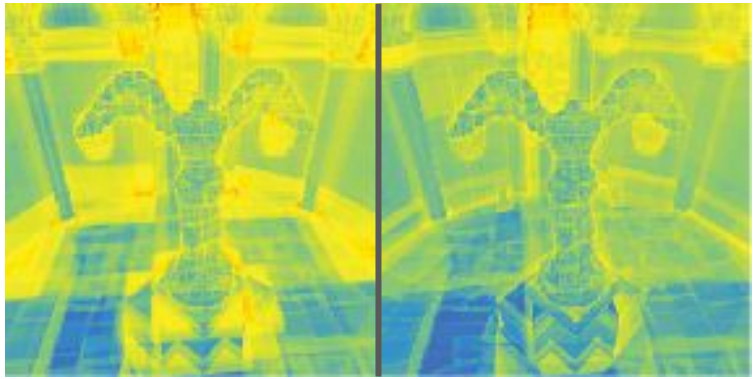
Conversion Time 12.78ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	7.74 / **5.37** (1.43x)
	ATRBVH	5.12 / **4.05** (1.26x)
	swpBVH	4.33 / **3.22** (1.34x)
Time (ms) Secondary rays	LBVH	148.09 / **131.96** (1.12x)
	ATRBVH	117.72 / **96.04** (1.22x)
	swpBVH	91.13 / **72.57** (1.25x)

Avg. intersections swpBVH		94.97 / **58.75** (1.61x)
Time (sec.) to render	LBVH	33.69 / **29.49** (1.14x)
	ATRBVH	26.88 / **22.13** (1.21x)

Results

Temple (400K triangles)



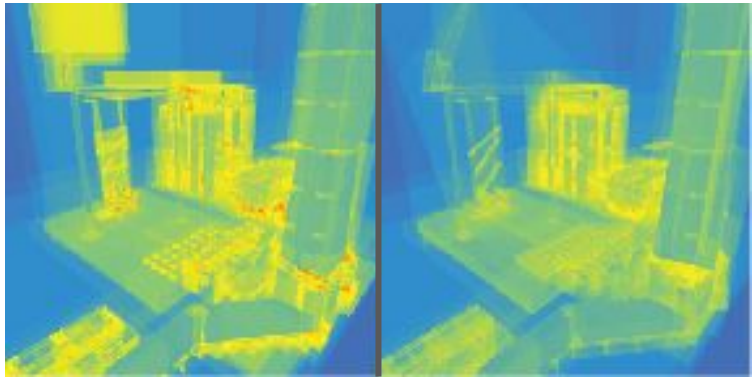
Conversion Time 1.98ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	**3.35** / 3.41 (0.98x)
	ATRBVH	**2.14** / 2.19 (0.97x)
	swpBVH	**1.76** / 1.83 (0.96x)
Time (ms) Secondary rays	LBVH	60.48 / **58.57** (1.03x)
	ATRBVH	**37.72** / 38.43 (0.98x)
	swpBVH	31.13 / **29.52** (1.05x)

Avg. intersections swpBVH		55.82 / **41.21** (1.35x)
Time (sec.) to render	LBVH	16.83 / **14.65** (1.15x)
	ATRBVH	10.48 / **9.52** (1.10x)

Results

Powerplant (12.7M triangles)



Conversion Time 52.63ms

		AABB BVH / OBB BVH
Time (ms) Primary rays	LBVH	5.91 / **4.01** (1.47x)
	ATRBVH	3.26 / **2.58** (1.26x)
	swpBVH	2.63 / **2.36** (1.11x)
Time (ms) Secondary rays	LBVH	29.03 / **26.67** (1.08x)
	ATRBVH	21.01 / **17.66** (1.18x)
	swpBVH	18.63 / **13.98** (1.33x)

Avg. intersections swpBVH		32.99 / **24.11** (1.36x)
Time (sec.) to render	LBVH	14.15 / **11.18** (1.27x)
	ATRBVH	9.20 / **6.96** (1.32x)

Results

Scene	Builder		Conversion				Total construction time (relative)	
	LBVH	ATRBVH	Project	Select	Refit	Finalise	LBVH	ATRBVH
Human	5.56	10.96	3.28	0.84	1.15	0.66	11.48 (× 2.06)	16.89 (× 1.54)
Crown	12.94	25.36	8.97	1.92	4.18	1.78	29.80 (× 2.30)	42.22 (× 1.66)
Sheep	41.40	77.40	27.00	5.30	7.50	5.12	86.31 (× 2.08)	122.32 (× 1.58)
Hairball	12.01	22.62	7.36	1.59	2.14	1.45	24.55 (× 2.04)	35.15 (× 1.55)
Dragon	3.98	6.91	2.94	0.51	0.76	0.44	8.63 (× 2.17)	11.55 (× 1.67)
Trees	1.31	2.52	0.52	0.14	0.20	0.12	2.28 (× 1.74)	3.49 (× 1.38)
Temple	2.25	4.35	1.10	0.30	0.37	0.23	4.24 (× 1.88)	6.34 (× 1.46)
Bistro (exterior)	10.88	21.29	7.21	1.83	2.32	1.42	23.67 (× 2.18)	34.07 (× 1.60)
Bistro (interior)	4.50	8.56	2.72	0.70	0.88	0.54	9.35 (× 2.08)	13.41 (× 1.57)
Powerplant	41.44	79.56	29.20	8.91	8.69	5.83	94.07 (× 2.27)	132.19 (× 1.66)

Results

Scene	Builder		Conversion				Total construction time (relative)	
	LBVH	ATRBVH	Project	Select	Refit	Finalise	LBVH	ATRBVH
Human	5.56	10.96	3.28	0.84	1.15	0.66	11.48 (× 2.06)	16.89 (× 1.54)
Crown	12.94	25.36	8.97	1.92	4.18	1.78	29.80 (× 2.30)	42.22 (× 1.66)
Sheep	41.40	77.40	27.00	5.30	7.50	5.12	86.31 (× 2.08)	122.32 (× 1.58)
Hairball	12.01	22.62	7.36	1.59	2.14	1.45	24.55 (× 2.04)	35.15 (× 1.55)
Dragon	3.98	6.91	2.94	0.51	0.76	0.44	8.63 (× 2.17)	11.55 (× 1.67)
Trees	1.31	2.52	0.52	0.14	0.20	0.12	2.28 (× 1.74)	3.49 (× 1.38)
Temple	2.25	4.35	1.10	0.30	0.37	0.23	4.24 (× 1.88)	6.34 (× 1.46)
Bistro (exterior)	10.88	21.29	7.21	1.83	2.32	1.42	23.67 (× 2.18)	34.07 (× 1.60)
Bistro (interior)	4.50	8.56	2.72	0.70	0.88	0.54	9.35 (× 2.08)	13.41 (× 1.57)
Powerplant	41.44	79.56	29.20	8.91	8.69	5.83	94.07 (× 2.27)	132.19 (× 1.66)

Results

Scene	Builder		Conversion				Total construction time (relative)	
	LBVH	ATRBVH	Project	Select	Refit	Finalise	LBVH	ATRBVH
Human	5.56	10.96	3.28	0.84	1.15	0.66	11.48 (× 2.06)	16.89 (× 1.54)
Crown	12.94	25.36	8.97	1.92	4.18	1.78	29.80 (× 2.30)	42.22 (× 1.66)
Sheep	41.40	77.40	27.00	5.30	7.50	5.12	86.31 (× 2.08)	122.32 (× 1.58)
Hairball	12.01	22.62	7.36	1.59	2.14	1.45	24.55 (× 2.04)	35.15 (× 1.55)
Dragon	3.98	6.91	2.94	0.51	0.76	0.44	8.63 (× 2.17)	11.55 (× 1.67)
Trees	1.31	2.52	0.52	0.14	0.20	0.12	2.28 (× 1.74)	3.49 (× 1.38)
Temple	2.25	4.35	1.10	0.30	0.37	0.23	4.24 (× 1.88)	6.34 (× 1.46)
Bistro (exterior)	10.88	21.29	7.21	1.83	2.32	1.42	23.67 (× 2.18)	34.07 (× 1.60)
Bistro (interior)	4.50	8.56	2.72	0.70	0.88	0.54	9.35 (× 2.08)	13.41 (× 1.57)
Powerplant	41.44	79.56	29.20	8.91	8.69	5.83	94.07 (× 2.27)	132.19 (× 1.66)

Results

Scene	Builder		Conversion				Total construction time (relative)	
	LBVH	ATRBVH	Project	Select	Refit	Finalise	LBVH	ATRBVH
Human	5.56	10.96	3.28	0.84	1.15	0.66	11.48 (× 2.06)	16.89 (× 1.54)
Crown	12.94	25.36	8.97	1.92	4.18	1.78	29.80 (× 2.30)	42.22 (× 1.66)
Sheep	41.40	77.40	27.00	5.30	7.50	5.12	86.31 (× 2.08)	122.32 (× 1.58)
Hairball	12.01	22.62	7.36	1.59	2.14	1.45	24.55 (× 2.04)	35.15 (× 1.55)
Dragon	3.98	6.91	2.94	0.51	0.76	0.44	8.63 (× 2.17)	11.55 (× 1.67)
Trees	1.31	2.52	0.52	0.14	0.20	0.12	2.28 (× 1.74)	3.49 (× 1.38)
Temple	2.25	4.35	1.10	0.30	0.37	0.23	4.24 (× 1.88)	6.34 (× 1.46)
Bistro (exterior)	10.88	21.29	7.21	1.83	2.32	1.42	23.67 (× 2.18)	34.07 (× 1.60)
Bistro (interior)	4.50	8.56	2.72	0.70	0.88	0.54	9.35 (× 2.08)	13.41 (× 1.57)
Powerplant	41.44	79.56	29.20	8.91	8.69	5.83	94.07 (× 2.27)	132.19 (× 1.66)

Conclusions

Method

- A GPU accelerated OBB extraction implementation
- Parallel conversion of Bounding Volume Hierarchies to Oriented Bounding Box Trees on the GPU

Conclusions

Method

- A GPU accelerated OBB extraction implementation
- Parallel conversion of Bounding Volume Hierarchies to Oriented Bounding Box Trees on the GPU

Limitations

- Not suitable for all scenes
- Memory overhead
  - Construction
  - Traversal

Future Work

Future Work

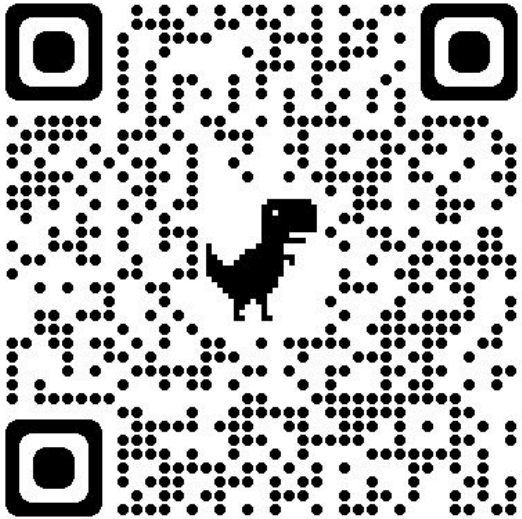
- OBB-aware construction
- Hybrid construction and traversal
- Revisit kernel traversal
- Hardware accelerated Ray-OBB intersection that exists for instancing

Acknowledgments

This research was funded by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “3rd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers” (Project No: 7310).

Questions

Thank you for your time!



References

- Domingues, Leonardo R., and Helio Pedrini. "Bounding volume hierarchy optimization through agglomerative treelet restructuring." Proceedings of the 7th Conference on High-Performance Graphics. 2015.
- Wald, Ingo, et al. "Using hardware ray transforms to accelerate ray/primitive intersections for long, thin primitive types." Proceedings of the ACM on Computer Graphics and Interactive Techniques 3.2 (2020): 1-16.
- Woop, Sven, et al. "Exploiting Local Orientation Similarity for Efficient Ray Traversal of Hair and Fur." High Performance Graphics 3 (2014).
- Karras, T. (2012, June). Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics (pp. 33-37).
- Pharr, Matt, Wenzel Jakob, and Greg Humphreys. Physically based rendering: From theory to implementation. Morgan Kaufmann, 2016.

References

- Larsson, Thomas, and Linus Källberg. "Fast computation of tight-fitting oriented bounding boxes." Game Engine Gems 2 (2011): 1.
- Stich, Martin, Heiko Friedrich, and Andreas Dietrich. "Spatial splits in bounding volume hierarchies." Proceedings of the Conference on High Performance Graphics 2009. 2009.
- Wald, Ingo. "On fast construction of SAH-based bounding volume hierarchies." 2007 IEEE Symposium on Interactive Ray Tracing. IEEE, 2007.
- Aila, Timo, and Samuli Laine. "Understanding the efficiency of ray traversal on GPUs." Proceedings of the conference on high performance graphics 2009. 2009.
- Sabino, Rodolfo, et al. "Fast and Robust Ray/OBB Intersection Using the Lorentz Transformation." Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX (2021): 519-528.
- Dinas, Simena, and José M. Bañón. "A literature review of bounding volumes hierarchy focused on collision detection." Ingeniería y competitividad 17.1 (2015): 49-62

Supplemental Material

Introduction

Queries on geometric primitives

Ray tracing

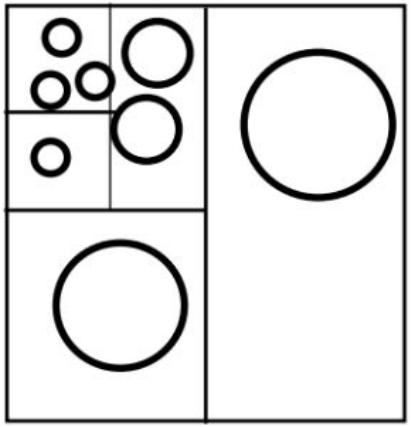
Image synthesis
- Production
- Real-time

Thanks to advances in both
- Hardware
- Software

Introduction

Space-partitioning data structure

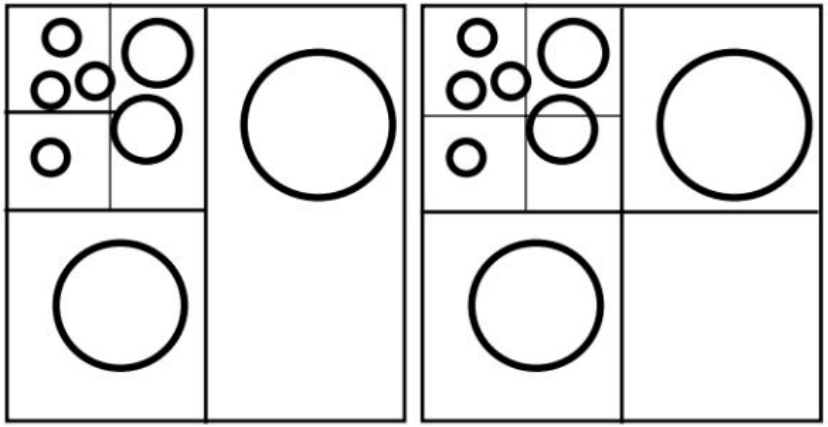
- KD-tree



Introduction

Space-partitioning data structure

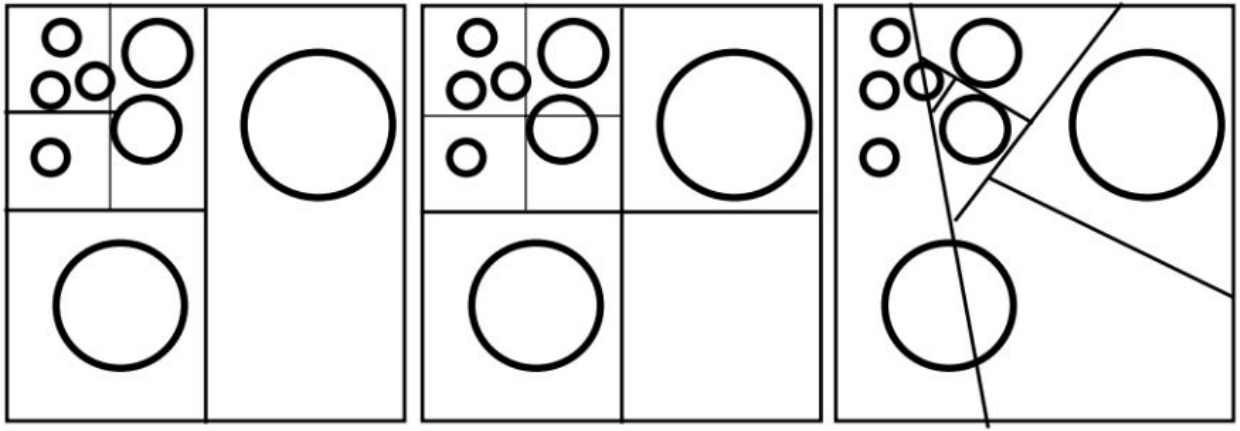
- KD-tree
- Octree



Introduction

Space-partitioning data structure

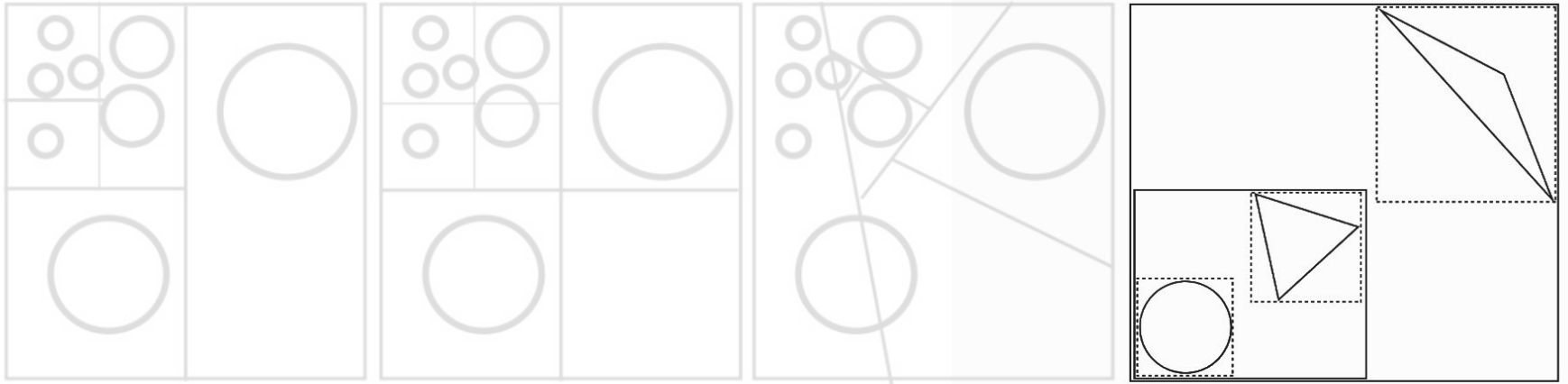
- KD-tree
- Octree
- BSP-tree



Introduction

Space-partitioning data structure

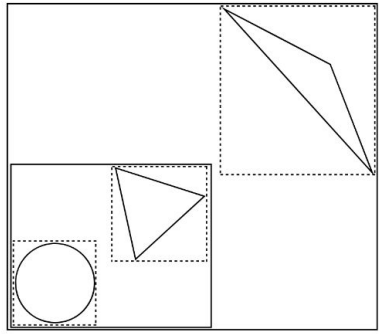
- KD-tree
- Octree
- BSP-tree
- **Bounding Volume Hierarchy**



Introduction

Bounding Volume Hierarchy

Bounding Volume
- Any closed geometric shape



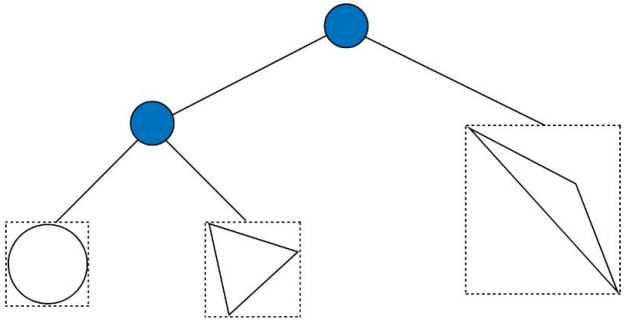


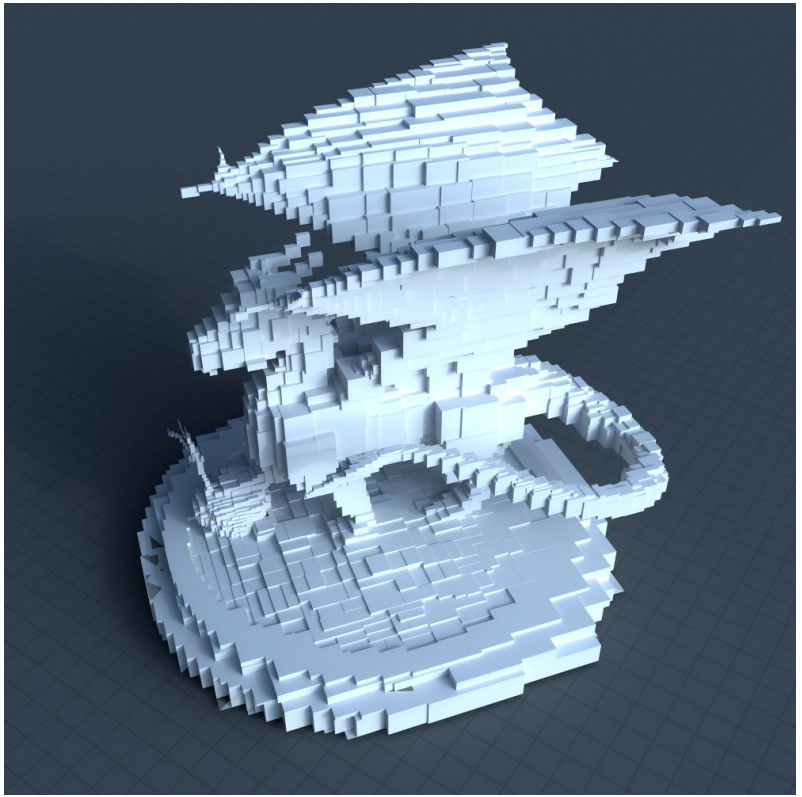
Image from PBRT v3 [6]

Introduction

Bounding Volume Hierarchy

Bounding Volume
- Any closed geometric shape

Axis-Aligned Bounding Boxes are very common



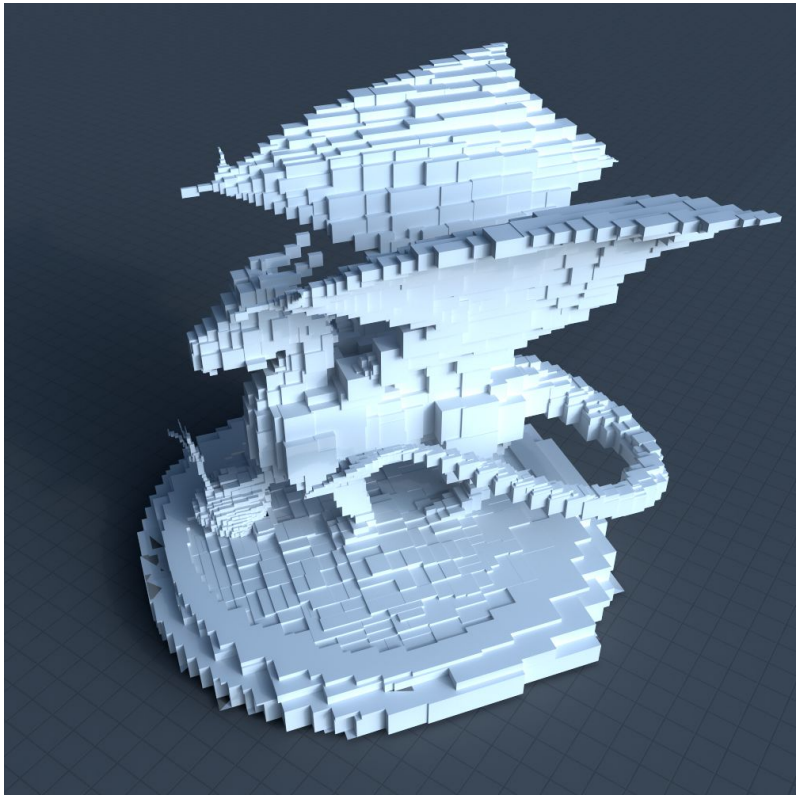
Introduction

Bounding Volume Hierarchy

Bounding Volume
- Any closed geometric shape

Axis-Aligned Bounding Boxes are very common

We focus on the potential of OBBs



Challenges

- Finding the best OBB for a collection of primitives is not an easy task
  - Complexity
  - Computation
  - Parallelizability
- Traversal
  - GPU friendliness
  - Required changes

Challenges

- Finding the best OBB for a collection of primitives is not an easy task
  - Complexity
  - Computation
  - Parallelizability
- Traversal
  - GPU friendliness
  - Required changes

Slow, optimal. Joseph O'Rourke

Fast, approximate. Principal Component Analysis

Introduction

Pros

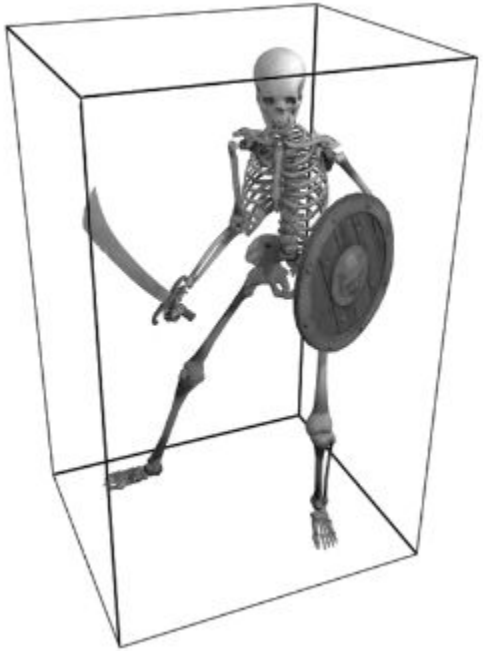
Tighter fitting
- Less wasteful intersections

Robust to transformation

Cons

Finding a good OBB

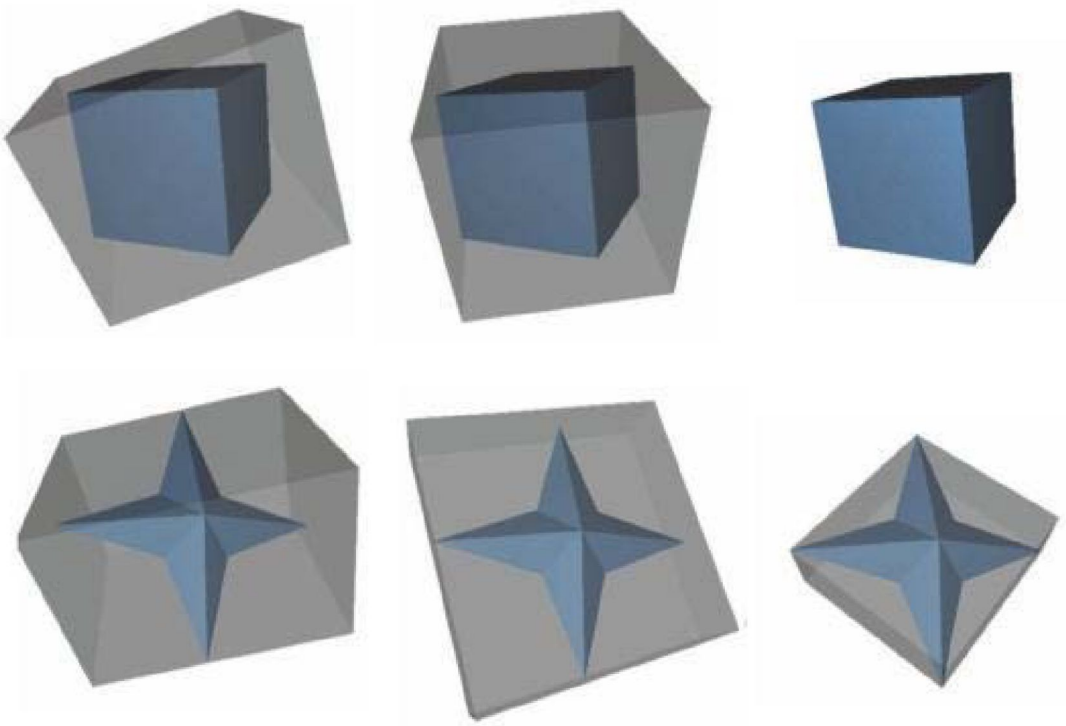
More involved intersection tests



AABB

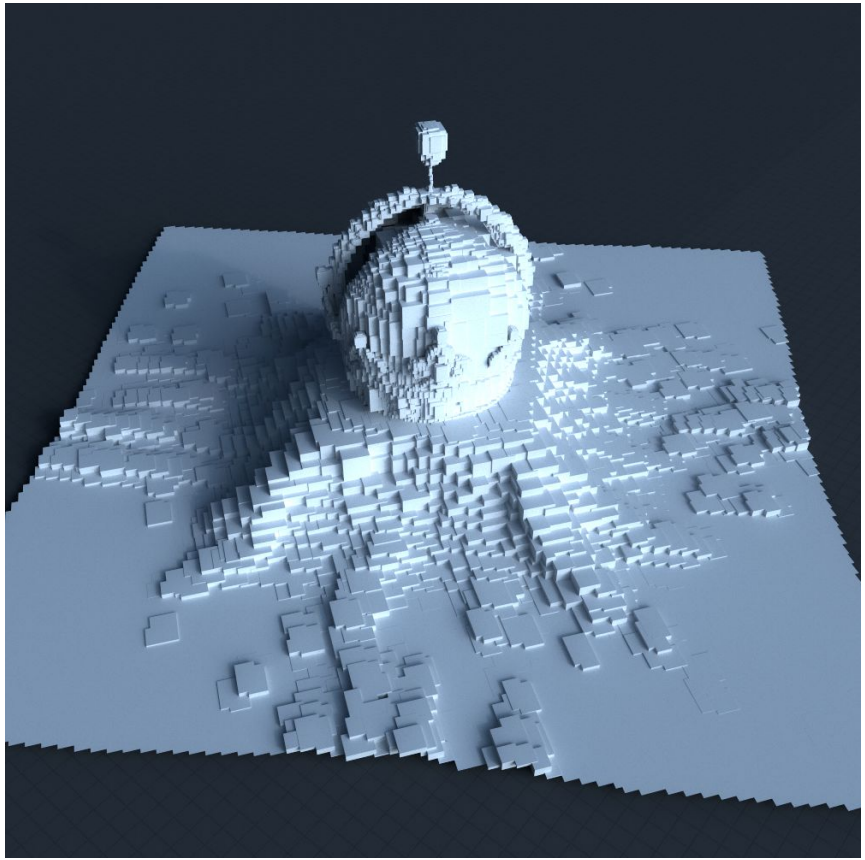
OBB

Di-tetrahedron OBB algorithm (DiTO)

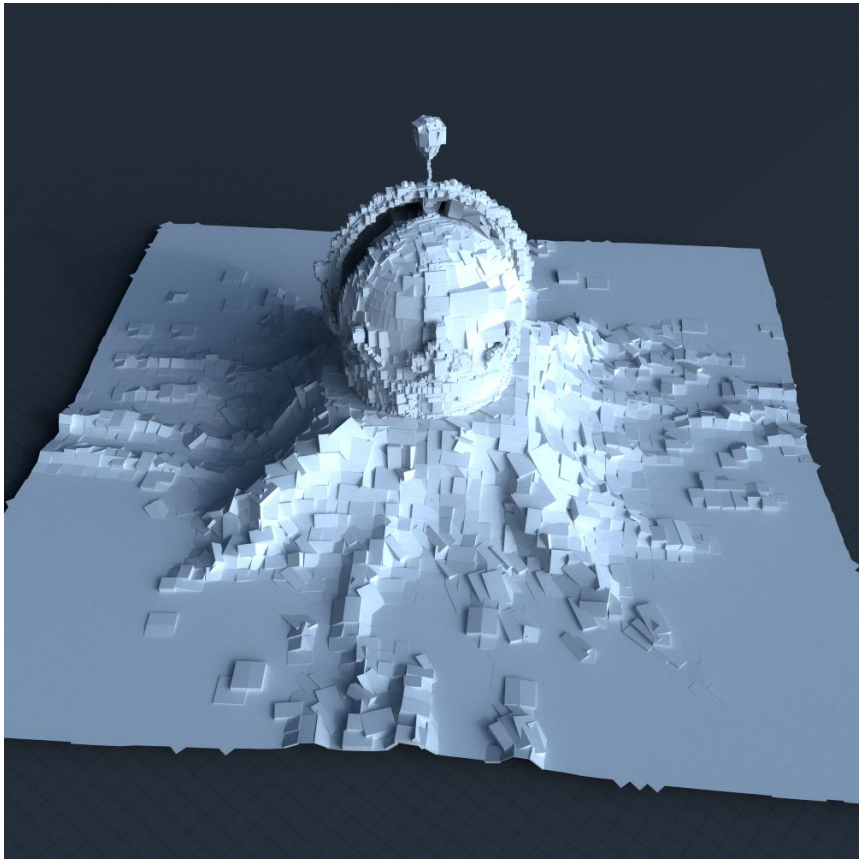


Larsson and Källberg, Fast computation of tight-fitting oriented bounding boxes, Game Engine Gems 2

Results

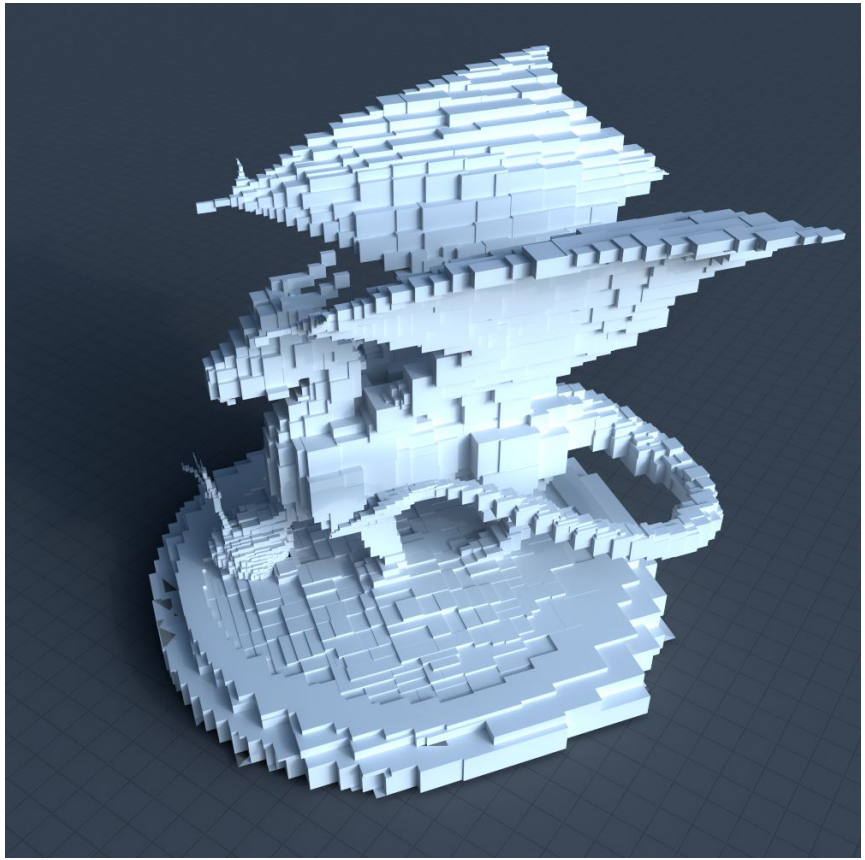


AABBs

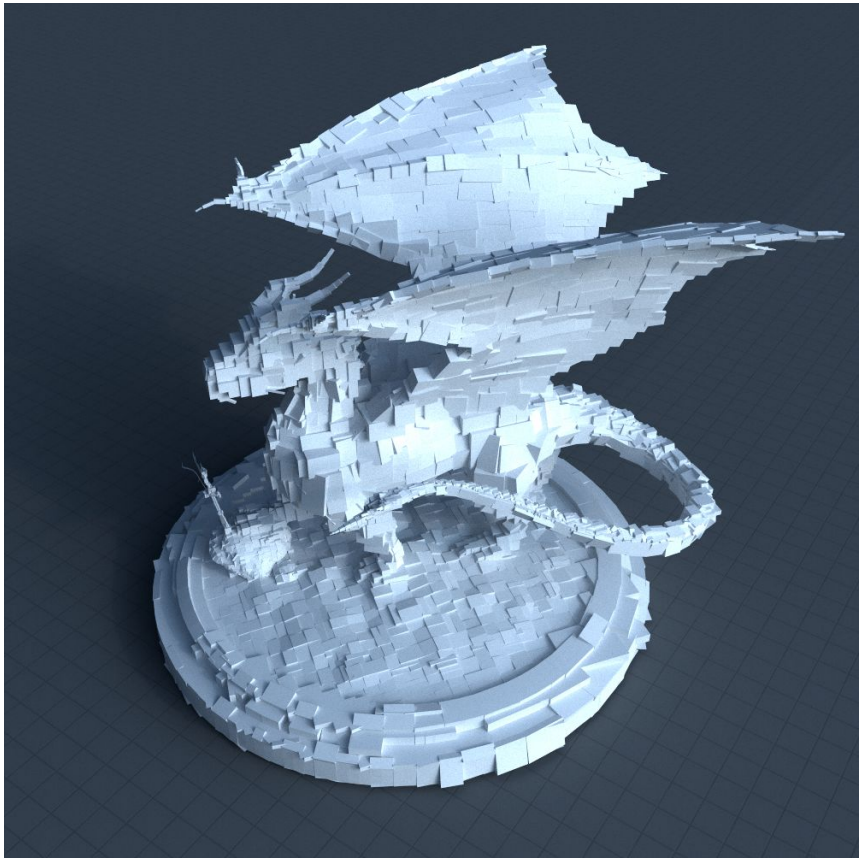


OBBs

Results

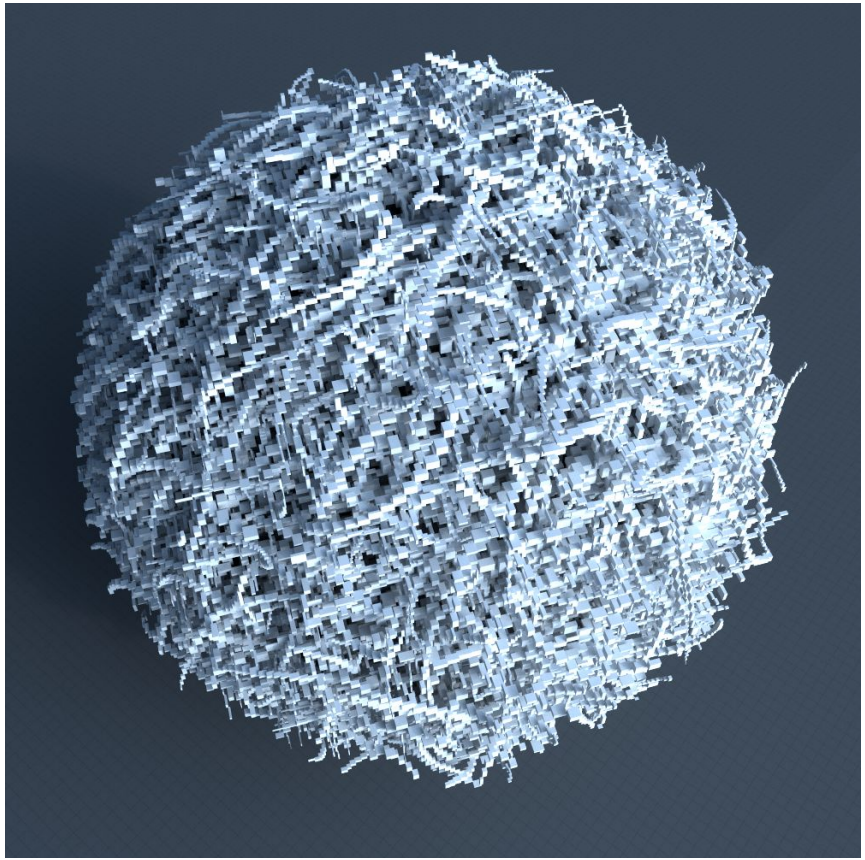


AABBs

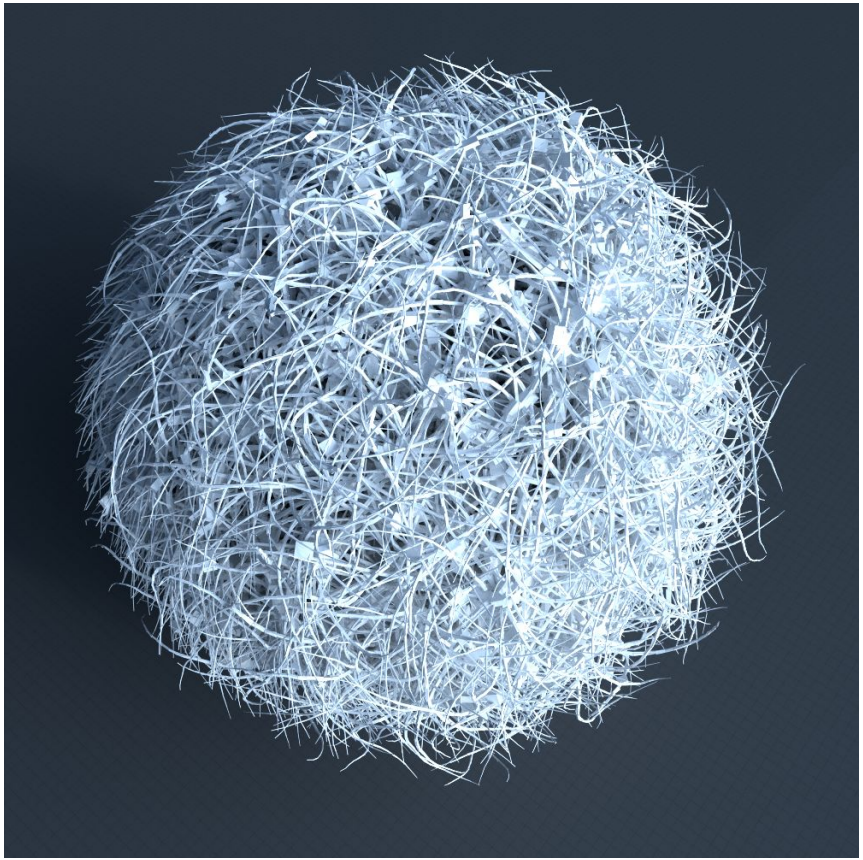


OBBs

Results

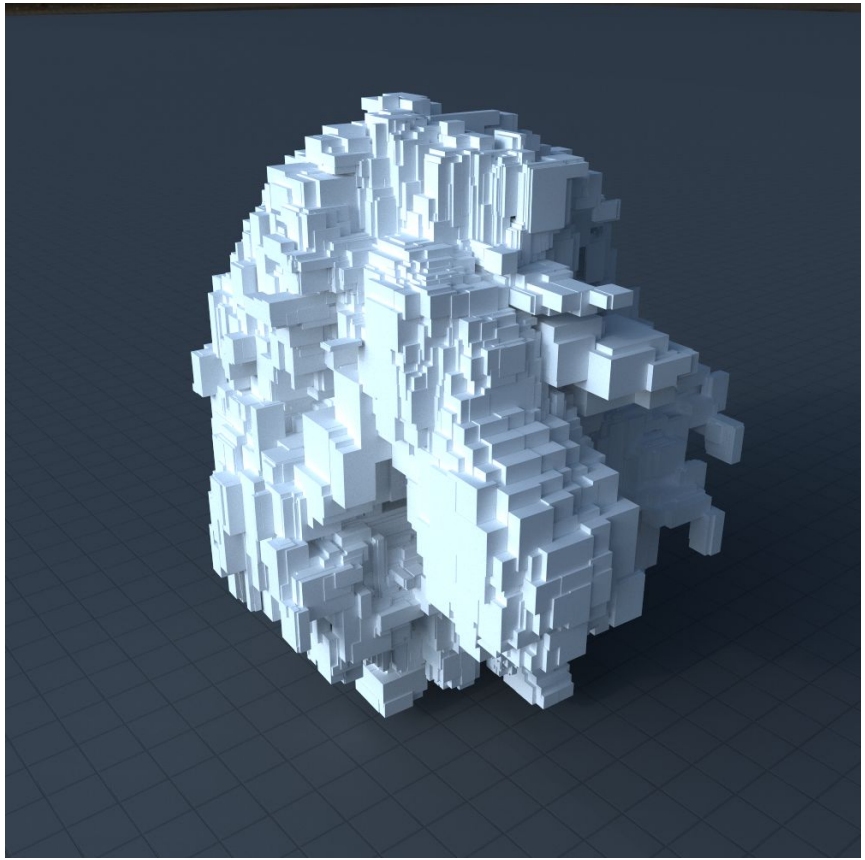


AABBs

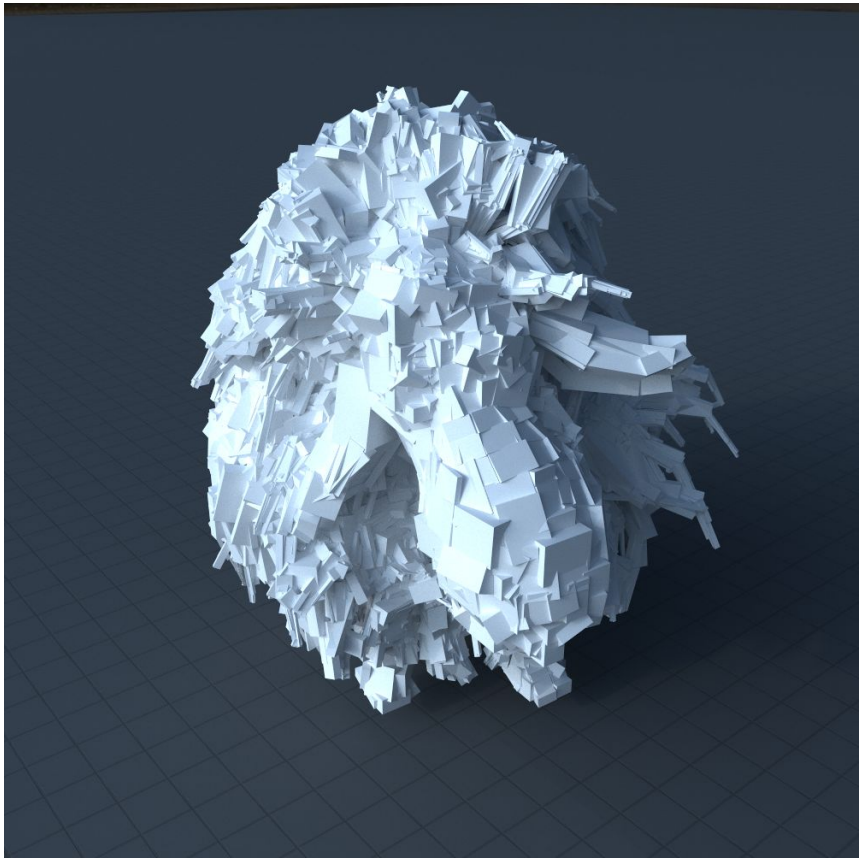


OBBs

Results



AABBs



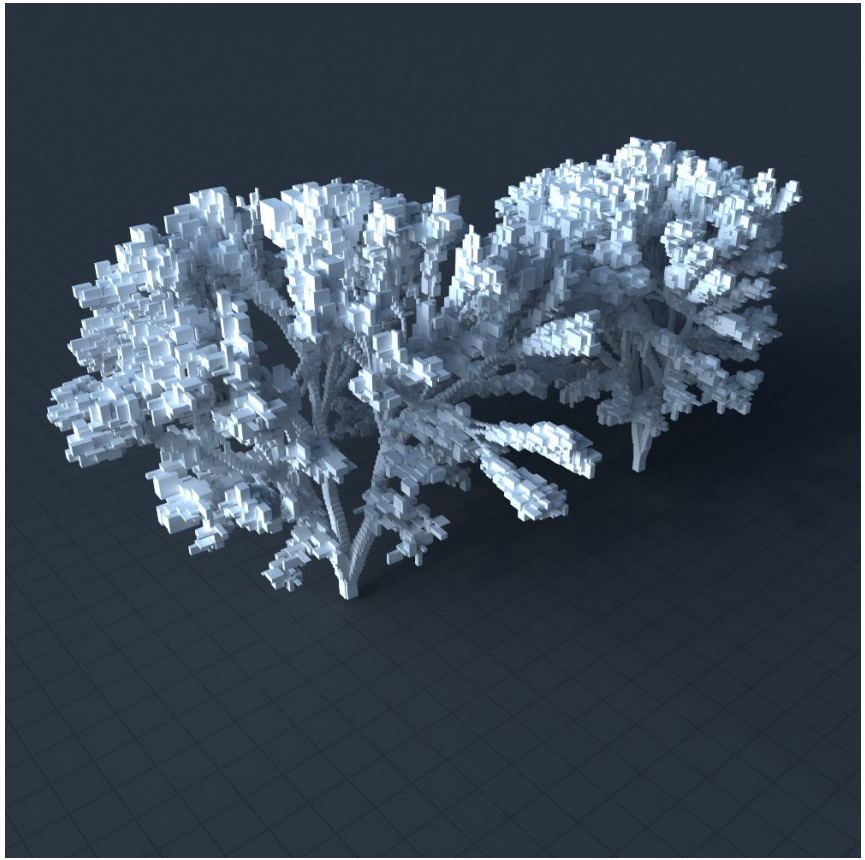
OBBs

Results

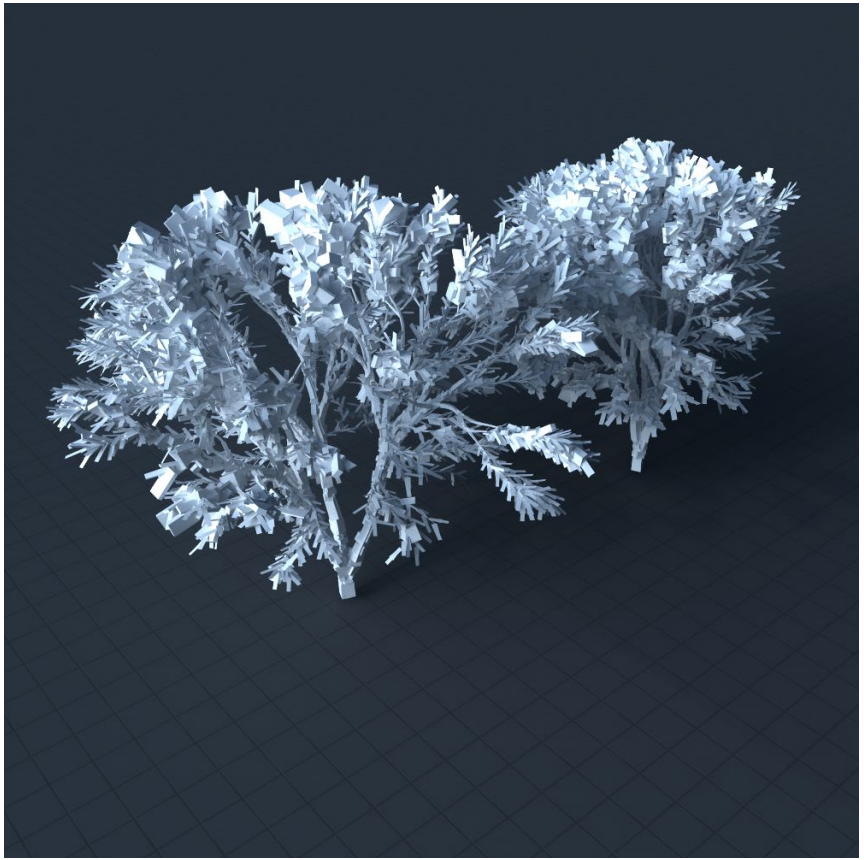
AABBs

OBBs

Results



AABBs



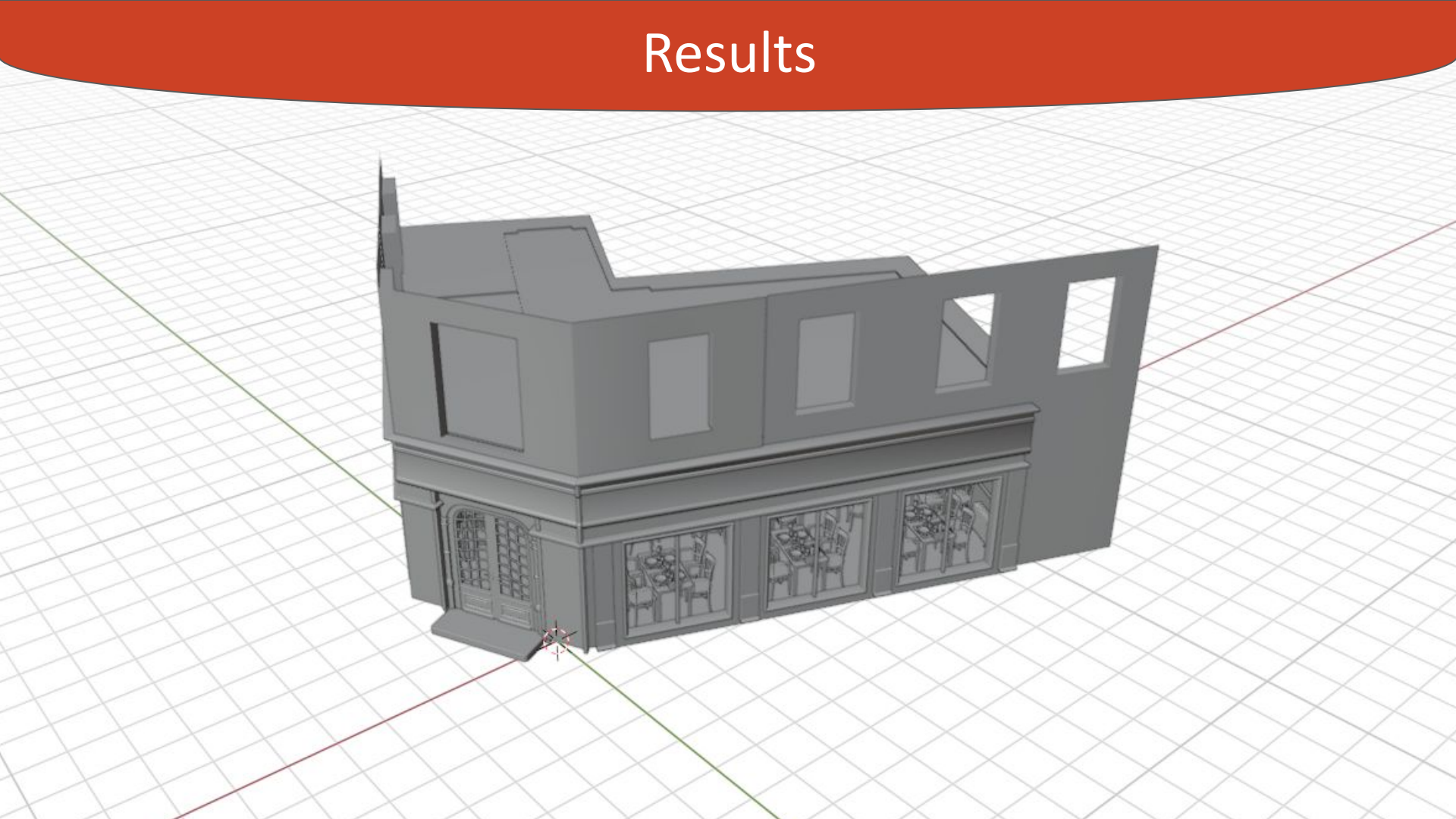
OBBs

Results

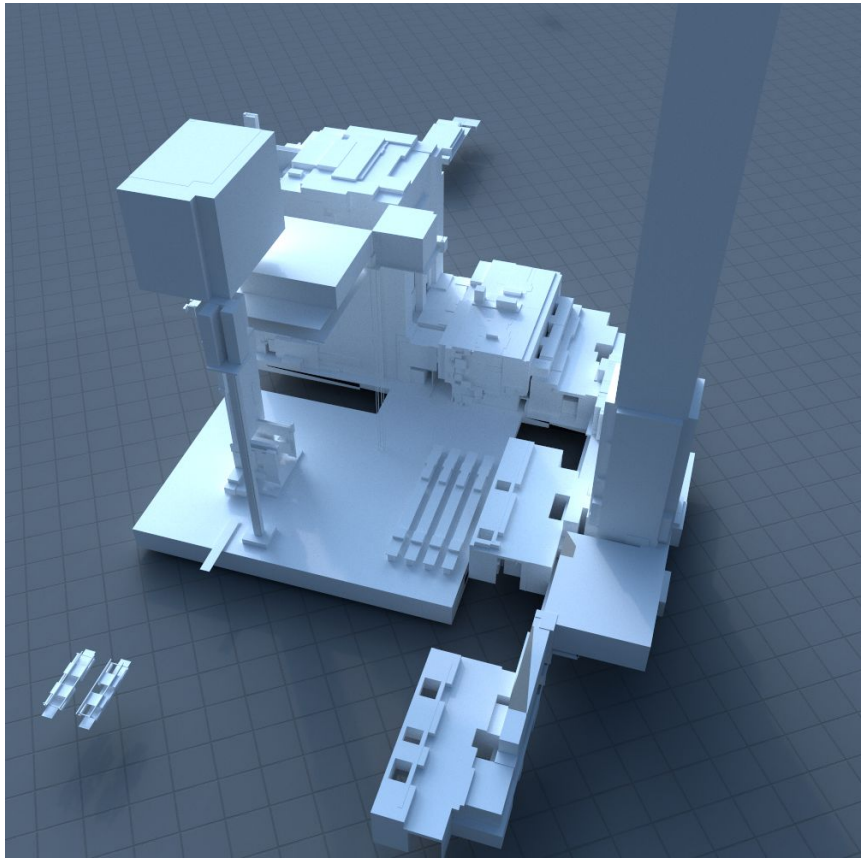
AABBs

OBBs

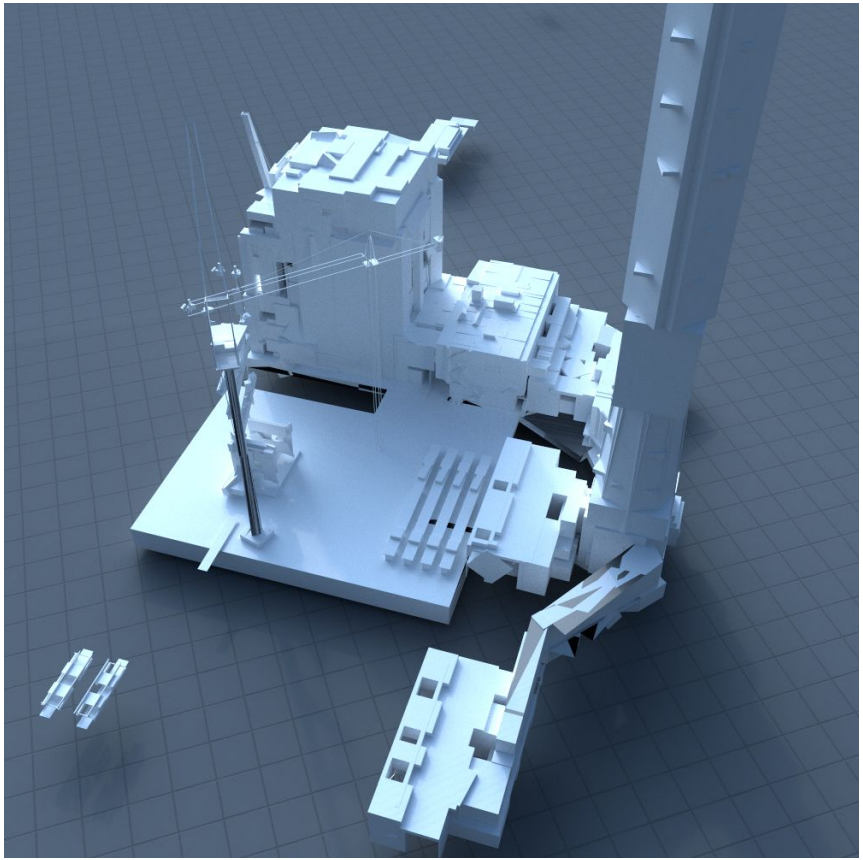
Results



Results

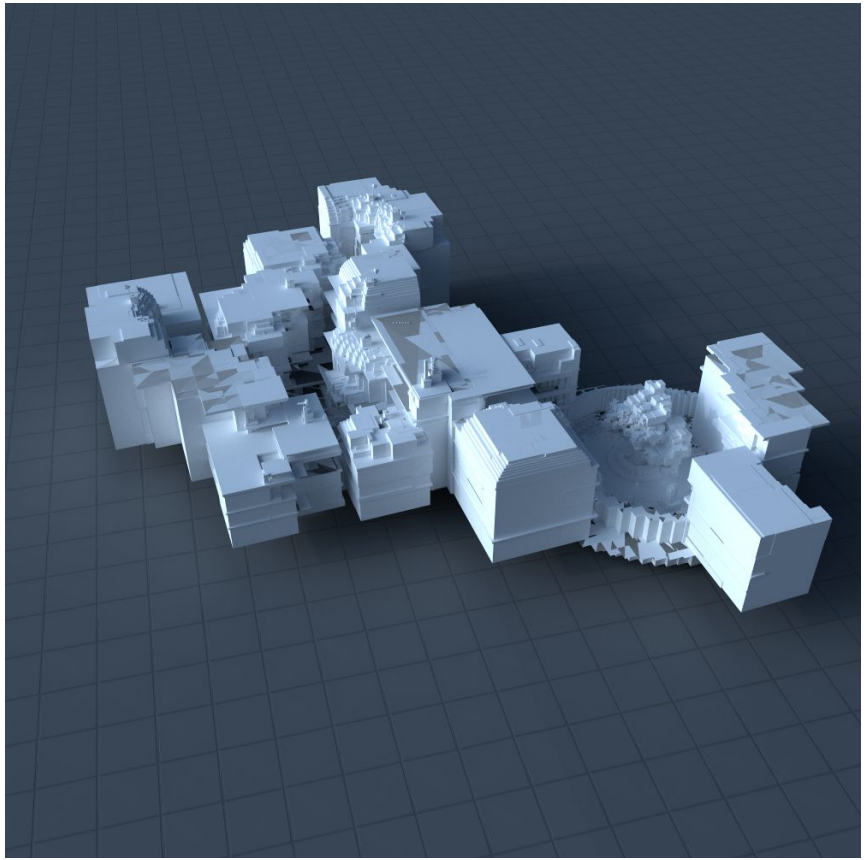


AABBs

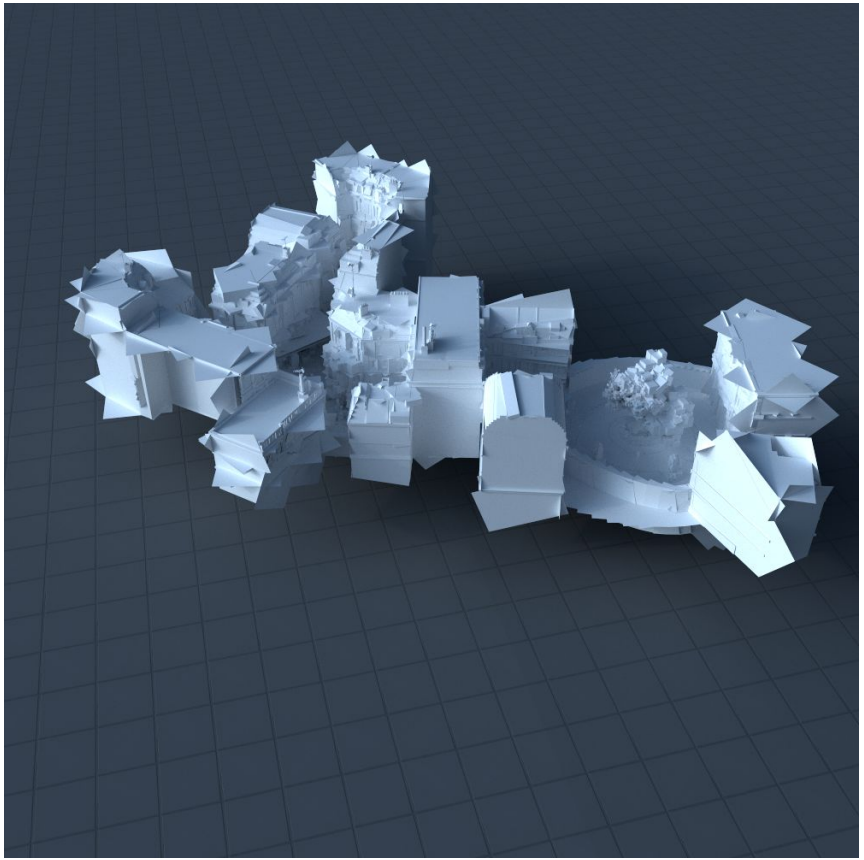


OBBs

Results



AABBs



OBBs

Results

Scene	Builder		Conversion				Total construction time (relative)	
	LBVH	ATRBVH	Project	Select	Refit	Finalise	LBVH	ATRBVH
Human	5.56	10.96	3.28	0.84	1.15	0.66	11.48 (× 2.06)	16.89 (× 1.54)
Crown	12.94	25.36	8.97	1.92	4.18	1.78	29.80 (× 2.30)	42.22 (× 1.66)
Sheep	41.40	77.40	27.00	5.30	7.50	5.12	86.31 (× 2.08)	122.32 (× 1.58)
Hairball	12.01	22.62	7.36	1.59	2.14	1.45	24.55 (× 2.04)	35.15 (× 1.55)
Dragon	3.98	6.91	2.94	0.51	0.76	0.44	8.63 (× 2.17)	11.55 (× 1.67)
Trees	1.31	2.52	0.52	0.14	0.20	0.12	2.28 (× 1.74)	3.49 (× 1.38)
Temple	2.25	4.35	1.10	0.30	0.37	0.23	4.24 (× 1.88)	6.34 (× 1.46)
Bistro (exterior)	10.88	21.29	7.21	1.83	2.32	1.42	23.67 (× 2.18)	34.07 (× 1.60)
Bistro (interior)	4.50	8.56	2.72	0.70	0.88	0.54	9.35 (× 2.08)	13.41 (× 1.57)
Powerplant	41.44	79.56	29.20	8.91	8.69	5.83	94.07 (× 2.27)	132.19 (× 1.66)