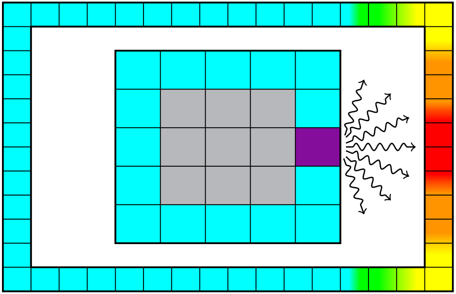
Turbocharging View Factor Computation with Quad- and Octrees

Justin Droba

Lyndon B. Johnson Space Center

MSU SURIEM REU
July 3, 2015

Concept of View Factor



- Have a mesh with its boundary elements. Pick one.
- Heat as blackbody radiation radiates from this face
- Heats up other faces the radiation impinges

A view factor quantifies radiation transfer between areas:

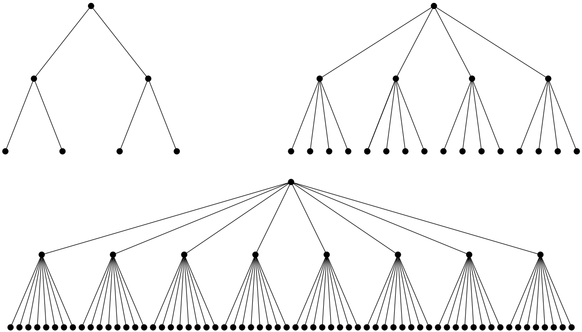
$$F(\mathbf{A}_1, \mathbf{A}_2) = \frac{\text{radiation leaving } \mathbf{A}_1 \text{ and impinging upon } \mathbf{A}_2}{\text{all radiation leaving } \mathbf{A}_1}$$

Binary Trees, Quadtrees, and Octrees

- Can compute view factors by Monte Carlo ray casting
- No info in ray on where it lands; must check **all** faces in set

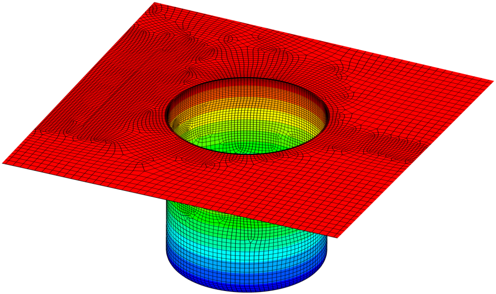
✗ Ok for small meshes, but gets glacially slow quickly

- Binary trees used in computer science for efficient searches
- Because meshes are 2D and 3D, quadtrees and octrees are the weapon of choice to speed up the searches

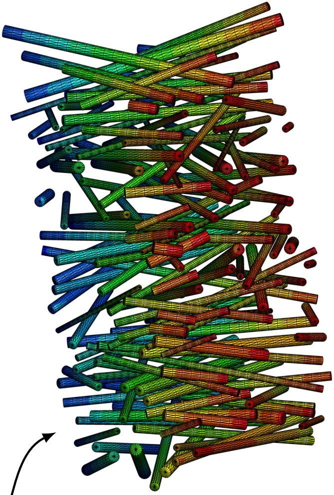


"Glacially Slow"?

Orion tile cavity problems take too long:



"The Monster"
26,000+ faces

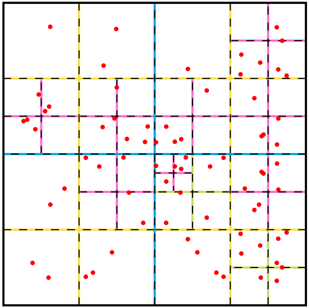


"The Nightmare"
75,000+ faces

Bonus: add capability to solve Eric's fiber problem.

A Tree Grows in Data

Suppose a data set that has some sort of spatial association.



Bin Size b:
99
32
16
8
4

Binary trees are formed by dividing data set in half repeatedly. Do that in each dimension until each box has b items.

Data associated with space: Boundary faces $\mathcal{F}$
Placement criterion: $\mathcal{F} \hookrightarrow \mathcal{B}$ if $|\mathcal{F} \cap \mathcal{B}|_{d-1} > 0$

The Hyperplane Separation Theorem

We pull out the artillery: convex analysis. We will power our intersection algorithm with this theorem:

Theorem (Hyperplane Separation Theorem)

Let $\mathcal{A} \subset \mathbb{R}^d$ be closed and $\mathcal{K} \subset \mathbb{R}^d$ be compact with both convex. Then $\mathcal{A} \cap \mathcal{K} = \emptyset$ *if and only if* there is a separating hyperplane $\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^d : \mathbf{x} \cdot \mathbf{p} = \alpha \right\}$ for some $\alpha \in \mathbb{R}$ and $\mathbf{p} \in \mathbb{R}^d \backslash \{\mathbf{0}\}$ such that

1. $\mathbf{p} \cdot \mathbf{a} > \alpha$ for all $\mathbf{p} \in \mathcal{P}$, $\mathbf{a} \in \mathcal{A}$ and $\mathbf{p} \cdot \mathbf{k} < \alpha$ for all $\mathbf{p} \in \mathcal{P}$, $\mathbf{k} \in \mathcal{K}$

or

2. $\mathbf{p} \cdot \mathbf{a} < \alpha$ for all $\mathbf{p} \in \mathcal{P}$, $\mathbf{a} \in \mathcal{A}$ and $\mathbf{p} \cdot \mathbf{k} > \alpha$ for all $\mathbf{p} \in \mathcal{P}$, $\mathbf{k} \in \mathcal{K}$

Theorem (Plain Language Version)

Two nice sets $\mathcal{A}$ and $\mathcal{K}$ don't intersect if we can divide space into $\mathcal{A}$'s half and $\mathcal{K}$'s half with a line (2D) or plane (3D).

The Separating Axis Theorem

Two vectors span a plane in 3D, so testing for hyperplanes directly is expensive. Cheaper to test for separating axes:

Definition (Separating Axis)

Let $\mathcal{P} \subset \mathbb{R}^d$ be a separating hyperplane. $\boldsymbol{\xi} \subset \mathbb{R}^d$ is a *separating axis* if $\boldsymbol{\xi} \perp \mathcal{P}$.

Because $\dim \mathcal{P} = d - 1$, $\dim \boldsymbol{\xi} = 1$. This leads to an obvious result:

Theorem

Let $\mathcal{A} \subset \mathbb{R}^d$ be closed and $\mathcal{K} \subset \mathbb{R}^d$ be compact with both convex. Then there exists a separating hyperplane for $\mathcal{A}$ and $\mathcal{K}$ if and only if there exists a separating axis between them.

The key: when $\mathcal{A}$ and $\mathcal{K}$ are orthogonally projected onto $\boldsymbol{\xi}$, $\mathcal{A} \cap \mathcal{K} = \emptyset$ if and only if the projection intervals do not overlap.

Collecting the Candidates

There are six ways for intersection to occur:

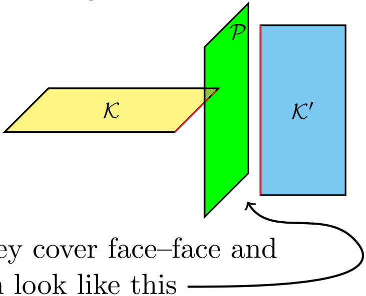
- face–face
- face–edge
- edge–edge

} Same in 2D

- ✓ face–node
- ✓ edge–node
- ✓ node–node

Can perturb non-intersecting faces and still maintain separation. Gives these candidate separating axes:

1. Face normals from $\mathcal{K}$
2. Face normals from $\mathcal{K}'$
3. Cross product of edge from $\mathcal{K}$ with one from $\mathcal{K}'$ (3D only)



In 2D, these are sufficient. In 3D, they cover face–face and face–edge cases. Edge–edge cases can look like this

SAT for Symmetric Objects

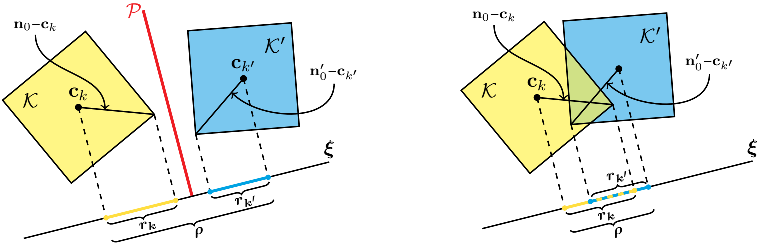
Algorithm (SAT for Symmetric Objects)

1. Let $\mathbf{c}_k \in \mathcal{K}$ and $\mathbf{c}_{k'} \in \mathcal{K}'$ be the centroids of $\mathcal{K}$ and $\mathcal{K}'$.
2. Let $\Pi(\mathbf{x})$ denote the projection of $\mathbf{x} \in \mathbb{R}^d$ onto $\boldsymbol{\xi}$. Define

$$r_k \triangleq \max_{0 \le i \le N-1} |\Pi(\mathbf{n}_i - \mathbf{c}_k)| \qquad r_{k'} \triangleq \max_{0 \le i \le N-1} |\Pi(\mathbf{n}'_i - \mathbf{c}_{k'})|$$

$$\rho \triangleq |\Pi(\mathbf{c}_k - \mathbf{c}_{k'})|$$

3. If $r_k + r_{k'} < \rho$, then $\boldsymbol{\xi}$ is a separating axis.



How It Works

The SAT for symmetric objects works so efficiently because

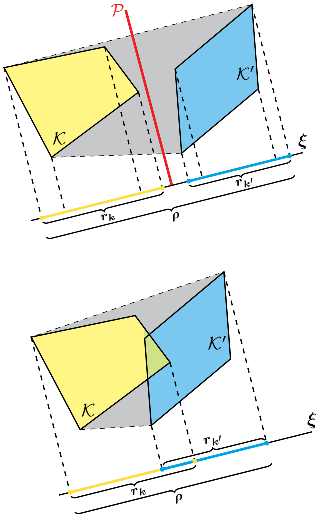
1. $\mathcal{K}$ and $\mathcal{K}'$ are symmetric about centroids $\mathbf{c}_k$ and $\mathbf{c}_{k'}$.
   $\implies$ Projection intervals symmetric about $\Pi(\mathbf{c}_k)$ and $\Pi(\mathbf{c}_k')$.
   $\implies$ It is sufficient to project the "radii" of $\mathcal{K}$ and $\mathcal{K}'$.
2. By linearity of projection operator, we can project the centroid-to-centroid segment.

For non-symmetric objects, it is a bit more complicated:

1. Must project every node of $\mathcal{K}$ and $\mathcal{K}'$.
2. Must project every node of the convex hull of $\mathcal{K}$ and $\mathcal{K}'$.

Because $\mathcal{K}$ and $\mathcal{K}'$ are convex, extreme points of convex hull come from nodes of $\mathcal{K}$ and $\mathcal{K}'$. Can reuse values from Step 1.

SAT for Non-symmetric Objects



Algorithm

1. Let $\{\mathbf{n}_i\}_{i=0}^{N-1}$ and $\{\mathbf{n}'_i\}_{i=0}^{N'-1}$ be nodes of $\mathcal{K}$ and $\mathcal{K}'$. Put

$$p_i \triangleq \mathbf{n}_i \cdot \boldsymbol{\xi} \qquad p'_i \triangleq \mathbf{n}'_i \cdot \boldsymbol{\xi}$$

2. Compute lengths of proj. intervals:

$$r_k \triangleq \max_{0\le i\le N-1} p_i - \min_{0\le i\le N-1} p_i$$

$$r_{k'} \triangleq \max_{0\le i\le N-1} p'_i - \min_{0\le i\le N-1} p'_i$$

$$\rho \triangleq \max\left\{ \max_{0\le i\le N-1} p_i, \max_{0\le i\le N-1} p'_i \right\} - \min\left\{ \min_{0\le i\le N-1} p_i, \min_{0\le i\le N-1} p'_i \right\}$$

3. $\boldsymbol{\xi}$ is separating axis if $r_k + r_{k'} < \rho$.

SAT for Insertion in 2D

Candidates for separating axes are face normals of box and normal to the face/edge $\mathbf{w} \triangleq \mathbf{n}_1 - \mathbf{n}_0$:

$$\boldsymbol{\xi}_1 = (1,0)^T \qquad \boldsymbol{\xi}_2 = (0,1)^T \qquad \boldsymbol{\xi}_3 = \left(-w^{(1)}, w^{(0)}\right)^T$$

Both $\mathcal{B}$ and $\mathcal{F}$ are symmetric: can use SAT for symmetric objects. Project $\mathbf{w}$, box radius $\mathbf{d} \triangleq \mathbf{x}_{\max} - \mathbf{x}_{\min}$, and centroid-to-centroid vector $\mathbf{m} \triangleq \mathbf{n}_0 + \mathbf{n}_1 - \mathbf{x}_{\max} - \mathbf{x}_{\min}$ onto $\boldsymbol{\xi}$:

Axis	$\boldsymbol{\xi}$	r_k	$r_{k'}$	$\boldsymbol{\rho}$
1	$(1,0)^T$	$\left\|d^{(0)}\right\|$	$\left\|w^{(0)}\right\|$	$\left\|m^{(0)}\right\|$
2	$(0,1)^T$	$\left\|d^{(1)}\right\|$	$\left\|w^{(1)}\right\|$	$\left\|m^{(1)}\right\|$
3	$\left(-w^{(1)}, w^{(0)}\right)^T$	$\left\|d^{(0)}w^{(1)}\right\| + \left\|d^{(1)}w^{(0)}\right\|$	0	$\left\|w^{(0)}m^{(1)} - w^{(1)}m^{(0)}\right\|$

Because $d^{(i)} > 0$, we can lose some absolute values in column 3.

SAT for Insertion in 3D

In 3D, we do not expect $\mathcal{F}$ to be symmetric. But $\mathcal{B}$ is! Can compute the max and min of projection nodes $\mathbf{v}_i$ of $\mathcal{B}$ directly:

$$M \triangleq \max_i(\mathbf{v}_i \cdot \boldsymbol{\xi}) = \boldsymbol{\xi} \cdot \mathbf{x}_+ \qquad m \triangleq \min_i(\mathbf{v}_i \cdot \boldsymbol{\xi}) = \boldsymbol{\xi} \cdot \mathbf{x}_-$$

$$x_+^{(i)} = \begin{cases} x_{\max}^{(i)} & \text{if } \xi^{(i)} \geq 0 \\ x_{\min}^{(i)} & \text{if } \xi^{(i)} < 0 \end{cases} \qquad x_i^{(i)} = \begin{cases} x_{\min}^{(i)} & \text{if } \xi^{(i)} \geq 0 \\ x_{\max}^{(i)} & \text{if } \xi^{(i)} < 0 \end{cases}$$

First set of candidate are $\mathcal{B}$'s face normals $\boldsymbol{\xi}_1 = (1,0,0)^T$, $\boldsymbol{\xi}_2 = (0,1,0)^T$, and $\boldsymbol{\xi}_3 = (0,0,1)^T$ and normal $\boldsymbol{\nu}$ from $\mathcal{F}$:

Axis	$\boldsymbol{\xi}$	$\boldsymbol{r_k}$	$\boldsymbol{r_{k'}}$	$\boldsymbol{\rho}$
1	$(1,0,0)^T$	$d^{(0)}$	$\max n_i^{(0)} - \min n_i^{(0)}$	$\max\{x_{\max}^{(0)}, \max n_i^{(0)}\} - \min\{x_{\min}^{(0)}, \min n_i^{(0)}\}$
2	$(0,1,0)^T$	$d^{(1)}$	$\max n_i^{(1)} - \min n_i^{(1)}$	$\max\{x_{\max}^{(1)}, \max n_i^{(1)}\} - \min\{x_{\min}^{(1)}, \min n_i^{(1)}\}$
3	$(0,0,1)^T$	$d^{(2)}$	$\max n_i^{(2)} - \min n_i^{(2)}$	$\max\{x_{\max}^{(2)}, \max n_i^{(2)}\} - \min\{x_{\min}^{(2)}, \min n_i^{(2)}\}$
4	$\boldsymbol{\nu}$	$M - m$	$\max \boldsymbol{\nu}\cdot\mathbf{n}_i - \min \boldsymbol{\nu}\cdot\mathbf{n}_i$	$\max\{M, \max \boldsymbol{\nu}\cdot\mathbf{n}_i\} - \min\{m, \min \boldsymbol{\nu}\cdot\mathbf{n}_i\}$

SAT for Insertion in 3D (continued)

Second set of axes given by cross products of box edges with face edges $\mathbf{f}_i \triangleq \mathbf{n}_{i+1} - \mathbf{n}_i$:

$$\boldsymbol{\xi}_{5,i} = (1,0,0)^T \times \mathbf{f}_i \qquad \alpha_{j,i} \triangleq \mathbf{n}_j \cdot \boldsymbol{\xi}_{5,i}$$
$$\boldsymbol{\xi}_{6,i} = (0,1,0)^T \times \mathbf{f}_i \qquad \beta_{j,i} \triangleq \mathbf{n}_j \cdot \boldsymbol{\xi}_{6,i}$$
$$\boldsymbol{\xi}_{7,i} = (0,0,1)^T \times \mathbf{f}_i \qquad \gamma_{j,i} \triangleq \mathbf{n}_j \cdot \boldsymbol{\xi}_{7,i}$$

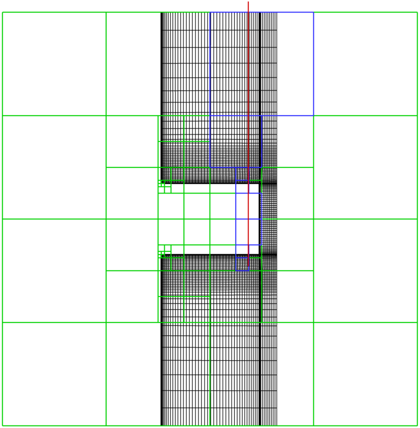
Can compute everything without forming $\mathbf{f}_i$ or ever computing cross product. Also have $\alpha_{j,i} = \alpha_{j+1,i}$, similarly for $\beta_{j,i}$ and $\gamma_{j,i}$.

Axis	$\boldsymbol{\xi}$	$\boldsymbol{r_k}$	$\boldsymbol{r_{k'}}$	$\boldsymbol{\rho}$
$5,i$	$\left(0,-f_i^{(2)},f_i^{(1)}\right)^T$	$M_{5,i} - m_{5,i}$	$\max\limits_{j\neq i} \alpha_{j,i} - \min\limits_{j\neq i} \alpha_{j,i}$	$\max\left\{M_{5,i}, \max\limits_{j\neq i} \alpha_{j,i}\right\} - \min\left\{m_{5,i}, \min\limits_{j\neq i} \alpha_{j,i}\right\}$
$6,i$	$\left(f_i^{(2)},0,-f_i^{(0)}\right)^T$	$M_{6,i} - m_{6,i}$	$\max\limits_{j\neq i} \beta_{j,i} - \min\limits_{j\neq i} \beta_{j,i}$	$\max\left\{M_{6,i}, \max\limits_{j\neq i} \beta_{j,i}\right\} - \min\left\{m_{6,i}, \min\limits_{j\neq i} \beta_{j,i}\right\}$
$7,i$	$\left(-f_i^{(1)},f_i^{(0)},0\right)^T$	$M_{7,i} - m_{7,i}$	$\max\limits_{j\neq i} \gamma_{j,i} - \min\limits_{j\neq i} \gamma_{j,i}$	$\max\left\{M_{7,i}, \max\limits_{j\neq i} \gamma_{j,i}\right\} - \min\left\{m_{7,i}, \min\limits_{j\neq i} \gamma_{j,i}\right\}$

$M_{j,i}$ and $m_{j,i}$ are defined/computed like M and m of last slide.

Trees Are Made for Climbing

- Have a mesh. With a tree built from it.
- Shoot a ray from one of the faces.
- Now want to identify which boxes the ray visits.

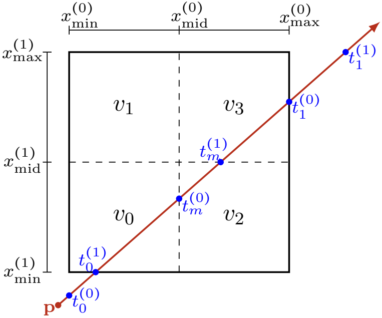


Intersection with Child Boxes

SAT gives no info about intersection point, so poor for recursion. Traversal algorithm will look like "first attempt."

Suppose we have ray $\mathfrak{r}(t) = \mathbf{p} + t\mathbf{r}$. $\mathbf{p}$ is inside mother box, so we need to move it out: $\mathbf{p} \leftarrow \mathbf{p} - n\mathbf{r}\Delta t$, where

$$\Delta t \triangleq \frac{1}{5} \min_{1 \le i \le d} \left\{ \frac{x_{\max}^{(i)} - x_{\min}^{(i)}}{\left| r^{(i)} \right|} \right\}$$



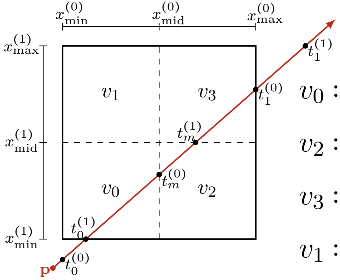
Compute times:

$$\text{Entry}: \quad t_0^{(i)} \triangleq \frac{x_{\text{entry}}^{(i)} - p_0^{(i)}}{r^{(i)}}$$

$$\text{Exit}: \quad t_1^{(i)} \triangleq \frac{x_{\text{exit}}^{(i)} - p_0^{(i)}}{r^{(i)}}$$

$$\text{Midpoint}: \quad t_m^{(i)} \triangleq \frac{1}{2}\left[t_0^{(i)} + t_1^{(i)} \right]$$

2D Ray Traversal



Time spent in:

$$v_0 : \quad \left[t^{(1)}_0, t^{(0)}_m\right] = \left[t^{(0)}_0, t^{(0)}_m\right] \bigcap \left[t^{(1)}_0, t^{(1)}_m\right]$$

$$v_2 : \quad \left[t^{(0)}_m, t^{(1)}_m\right] = \left[t^{(0)}_m, t^{(0)}_1\right] \bigcap \left[t^{(1)}_0, t^{(1)}_m\right]$$

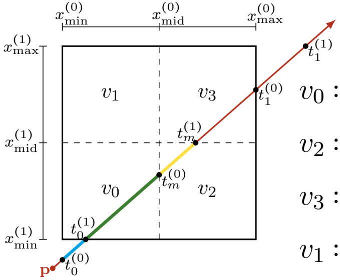
$$v_3 : \quad \left[t^{(0)}_m, t^{(0)}_1\right] = \left[t^{(0)}_m, t^{(0)}_1\right] \bigcap \left[t^{(1)}_m, t^{(1)}_1\right]$$

$$v_1 : \quad \emptyset = \left[t^{(0)}_0, t^{(0)}_m\right] \bigcap \left[t^{(1)}_m, t^{(1)}_1\right]$$

Child Box	Condition for Entry	Entry Times		Exit Times	
v_0	$\max\left\{t^{(0)}_0, t^{(1)}_0\right\} < \min\left\{t^{(0)}_m, t^{(1)}_m\right\}$	$t^{(0)}_0$	$t^{(1)}_0$	$t^{(0)}_m$	$t^{(1)}_m$
v_1	$\max\left\{t^{(0)}_0, t^{(1)}_m\right\} < \min\left\{t^{(0)}_m, t^{(1)}_1\right\}$	$t^{(0)}_0$	$t^{(1)}_m$	$t^{(0)}_m$	$t^{(1)}_1$
v_2	$\max\left\{t^{(0)}_m, t^{(1)}_0\right\} < \min\left\{t^{(0)}_1, t^{(1)}_m\right\}$	$t^{(0)}_m$	$t^{(1)}_0$	$t^{(0)}_1$	$t^{(1)}_m$
v_3	$\max\left\{t^{(0)}_m, t^{(1)}_m\right\} < \min\left\{t^{(0)}_1, t^{(1)}_1\right\}$	$t^{(0)}_m$	$t^{(1)}_m$	$t^{(0)}_1$	$t^{(1)}_1$

...but this table will only be valid if $r^{(i)} > 0$ for each i.

2D Ray Traversal



Time spent in:

$$v_0: \quad [t_0^{(1)}, t_m^{(0)}] = [t_0^{(0)}, t_m^{(0)}] \bigcap [t_0^{(1)}, t_m^{(1)}]$$

$$v_2: \quad [t_m^{(0)}, t_m^{(1)}] = [t_m^{(0)}, t_1^{(0)}] \bigcap [t_0^{(1)}, t_m^{(1)}]$$

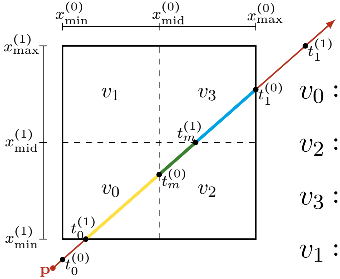
$$v_3: \quad [t_m^{(0)}, t_1^{(0)}] = [t_m^{(0)}, t_1^{(0)}] \bigcap [t_m^{(1)}, t_1^{(1)}]$$

$$v_1: \quad \emptyset = [t_0^{(0)}, t_m^{(0)}] \bigcap [t_m^{(1)}, t_1^{(1)}]$$

Child Box	**Condition for Entry**	**Entry Times**		**Exit Times**	
v_0	$\max\{t_0^{(0)}, t_0^{(1)}\} < \min\{t_m^{(0)}, t_m^{(1)}\}$	$t_0^{(0)}$	$t_0^{(1)}$	$t_m^{(0)}$	$t_m^{(1)}$
v_1	$\max\{t_0^{(0)}, t_m^{(1)}\} < \min\{t_m^{(0)}, t_1^{(1)}\}$	$t_0^{(0)}$	$t_m^{(1)}$	$t_m^{(0)}$	$t_1^{(1)}$
v_2	$\max\{t_m^{(0)}, t_0^{(1)}\} < \min\{t_1^{(0)}, t_m^{(1)}\}$	$t_m^{(0)}$	$t_0^{(1)}$	$t_1^{(0)}$	$t_m^{(1)}$
v_3	$\max\{t_m^{(0)}, t_m^{(1)}\} < \min\{t_1^{(0)}, t_1^{(1)}\}$	$t_m^{(0)}$	$t_m^{(1)}$	$t_1^{(0)}$	$t_1^{(1)}$

...but this table will only be valid if $r^{(i)} > 0$ for each i.

2D Ray Traversal



Time spent in:

$$v_0: \quad \left[t_0^{(1)}, t_m^{(0)}\right] = \left[t_0^{(0)}, t_m^{(0)}\right] \bigcap \left[t_0^{(1)}, t_m^{(1)}\right]$$

$$v_2: \quad \left[t_m^{(0)}, t_m^{(1)}\right] = \left[t_m^{(0)}, t_1^{(0)}\right] \bigcap \left[t_0^{(1)}, t_m^{(1)}\right]$$

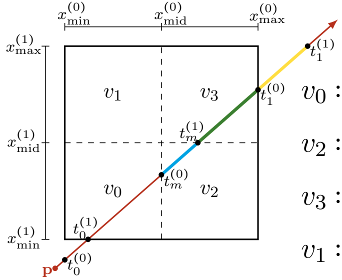
$$v_3: \quad \left[t_m^{(0)}, t_1^{(0)}\right] = \left[t_m^{(0)}, t_1^{(0)}\right] \bigcap \left[t_m^{(1)}, t_1^{(1)}\right]$$

$$v_1: \quad \emptyset = \left[t_0^{(0)}, t_m^{(0)}\right] \bigcap \left[t_m^{(1)}, t_1^{(1)}\right]$$

Child Box	**Condition for Entry**	**Entry Times**		**Exit Times**	
v_0	$\max\left\{t_0^{(0)}, t_0^{(1)}\right\} < \min\left\{t_m^{(0)}, t_m^{(1)}\right\}$	$t_0^{(0)}$	$t_0^{(1)}$	$t_m^{(0)}$	$t_m^{(1)}$
v_1	$\max\left\{t_0^{(0)}, t_m^{(1)}\right\} < \min\left\{t_m^{(0)}, t_1^{(1)}\right\}$	$t_0^{(0)}$	$t_m^{(1)}$	$t_m^{(0)}$	$t_1^{(1)}$
v_2	$\max\left\{t_m^{(0)}, t_0^{(1)}\right\} < \min\left\{t_1^{(0)}, t_m^{(1)}\right\}$	$t_m^{(0)}$	$t_0^{(1)}$	$t_1^{(0)}$	$t_m^{(1)}$
v_3	$\max\left\{t_m^{(0)}, t_m^{(1)}\right\} < \min\left\{t_1^{(0)}, t_1^{(1)}\right\}$	$t_m^{(0)}$	$t_m^{(1)}$	$t_1^{(0)}$	$t_1^{(1)}$

...but this table will only be valid if $r^{(i)} > 0$ for each i.

2D Ray Traversal



Time spent in:

$$v_0 : \quad \left[t^{(1)}_0, t^{(0)}_m\right] = \left[t^{(0)}_0, t^{(0)}_m\right] \bigcap \left[t^{(1)}_0, t^{(1)}_m\right]$$

$$v_2 : \quad \left[t^{(0)}_m, t^{(1)}_m\right] = \left[t^{(0)}_m, t^{(0)}_1\right] \bigcap \left[t^{(1)}_0, t^{(1)}_m\right]$$

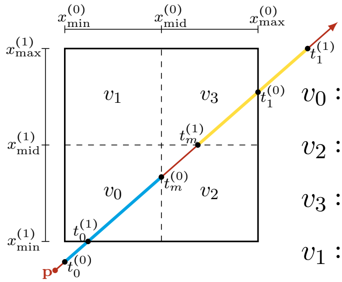
$$v_3 : \quad \left[t^{(0)}_m, t^{(0)}_1\right] = \left[t^{(0)}_m, t^{(0)}_1\right] \bigcap \left[t^{(1)}_m, t^{(1)}_1\right]$$

$$v_1 : \quad \emptyset = \left[t^{(0)}_0, t^{(0)}_m\right] \bigcap \left[t^{(1)}_m, t^{(1)}_1\right]$$

Child Box	Condition for Entry	Entry Times		Exit Times	
v_0	$\max\left\{t^{(0)}_0, t^{(1)}_0\right\} < \min\left\{t^{(0)}_m, t^{(1)}_m\right\}$	$t^{(0)}_0$	$t^{(1)}_0$	$t^{(0)}_m$	$t^{(1)}_m$
v_1	$\max\left\{t^{(0)}_0, t^{(1)}_m\right\} < \min\left\{t^{(0)}_m, t^{(1)}_1\right\}$	$t^{(0)}_0$	$t^{(1)}_m$	$t^{(0)}_m$	$t^{(1)}_1$
v_2	$\max\left\{t^{(0)}_m, t^{(1)}_0\right\} < \min\left\{t^{(0)}_1, t^{(1)}_m\right\}$	$t^{(0)}_m$	$t^{(1)}_0$	$t^{(0)}_1$	$t^{(1)}_m$
v_3	$\max\left\{t^{(0)}_m, t^{(1)}_m\right\} < \min\left\{t^{(0)}_1, t^{(1)}_1\right\}$	$t^{(0)}_m$	$t^{(1)}_m$	$t^{(0)}_1$	$t^{(1)}_1$

...but this table will only be valid if $r^{(i)} > 0$ for each i.

2D Ray Traversal



Time spent in:

$$v_0: \quad [t^{(1)}_0, t^{(0)}_m] = [t^{(0)}_0, t^{(0)}_m] \bigcap [t^{(1)}_0, t^{(1)}_m]$$

$$v_2: \quad [t^{(0)}_m, t^{(1)}_m] = [t^{(0)}_m, t^{(0)}_1] \bigcap [t^{(1)}_0, t^{(1)}_m]$$

$$v_3: \quad [t^{(0)}_m, t^{(0)}_1] = [t^{(0)}_m, t^{(0)}_1] \bigcap [t^{(1)}_m, t^{(1)}_1]$$

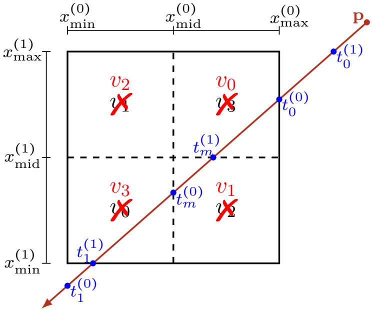
$$v_1: \quad \emptyset = [t^{(0)}_0, t^{(0)}_m] \bigcap [t^{(1)}_m, t^{(1)}_1]$$

Child Box	**Condition for Entry**	**Entry Times**		**Exit Times**	
v_0	$\max\{t^{(0)}_0, t^{(1)}_0\} < \min\{t^{(0)}_m, t^{(1)}_m\}$	$t^{(0)}_0$	$t^{(1)}_0$	$t^{(0)}_m$	$t^{(1)}_m$
v_1	$\max\{t^{(0)}_0, t^{(1)}_m\} < \min\{t^{(0)}_m, t^{(1)}_1\}$	$t^{(0)}_0$	$t^{(1)}_m$	$t^{(0)}_m$	$t^{(1)}_1$
v_2	$\max\{t^{(0)}_m, t^{(1)}_0\} < \min\{t^{(0)}_1, t^{(1)}_m\}$	$t^{(0)}_m$	$t^{(1)}_0$	$t^{(0)}_1$	$t^{(1)}_m$
v_3	$\max\{t^{(0)}_m, t^{(1)}_m\} < \min\{t^{(0)}_1, t^{(1)}_1\}$	$t^{(0)}_m$	$t^{(1)}_m$	$t^{(0)}_1$	$t^{(1)}_1$

...but this table will only be valid if $r^{(i)} > 0$ for each i.

Rays with Negative Components

Consider the reversal of the previous example:



	v_1 **(Failure)**
Reversal	$[t_m^{(0)}, t_1^{(0)}] \cap [t_0^{(1)}, t_m^{(1)}]$
Previous	$[t_0^{(0)}, t_m^{(0)}] \cap [t_m^{(1)}, t_1^{(1)}]$

	v_3	v_2	v_0
Reversal	$[t_0^{(0)}, t_m^{(0)}] \cap [t_0^{(1)}, t_m^{(1)}]$	$[t_0^{(0)}, t_m^{(0)}] \cap [t_m^{(1)}, t_1^{(1)}]$	$[t_m^{(0)}, t_1^{(0)}] \cap [t_m^{(1)}, t_1^{(1)}]$
Previous	$[t_m^{(1)}, t_1^{(0)}] \cap [t_m^{(1)}, t_1^{(1)}]$	$[t_m^{(0)}, t_1^{(0)}] \cap [t_0^{(1)}, t_m^{(1)}]$	$[t_0^{(0)}, t_m^{(0)}] \cap [t_0^{(1)}, t_m^{(1)}]$

A simple relabeling of boxes will allow reuse of previous table!

The Relabeling Scheme

The general mapping for box relabeling is $\ell \mapsto \ell \oplus a$, where

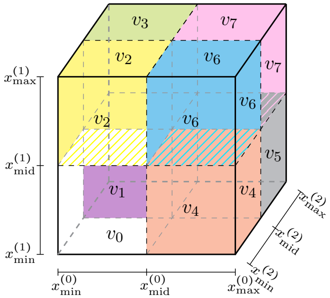
$$a \triangleq \begin{cases} 2\sigma(0) + \sigma(1) & \text{if } d = 2 \\ 4\sigma(0) + 2\sigma(1) + \sigma(2) & \text{if } d = 3 \end{cases}$$

$$\sigma(i) \triangleq \begin{cases} 0 & \text{if } r^{(i)} \geq 0 \\ 1 & \text{if } r^{(i)} < 0 \end{cases}$$

Written as binary string, a encodes the signs of the components of $\mathbf{r}$ (1 negative, 0 nonnegative).

In 3D, the box labeling order compatible with the relabeling scheme is this goofy thing:

With this, we can make a table for 3D very similar to 2D one.



One Last Thing: Infinite Arithmetic Module

- Computation of entry/exit times totally fails if $r^{(i)} = 0$.
- If $\left|r^{(i)}\right| < \varepsilon \ll 1$, then the numerics are bad too.
- Small (positive) direction value is inducer of exit only if

$$x_{\max}^{(i')} - p^{(i')} > \left(x_{\max}^{(i)} - p^{(i)}\right)\sqrt{\frac{1}{2\varepsilon^2} - 1}$$

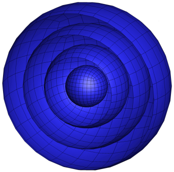
- Solution: when $\left|r^{(i)}\right| < \tau$, set

$$t_0^{(i)} = -\infty \qquad t_1^{(i)} = +\infty$$

- Says that $x_{\min}^{(i)} \leq \mathfrak{r}^{(i)}(t) \leq x_{\max}^{(i)}$ for all t.
- Computation of $t_m^{(i)}$ as before is undefined. Instead, define

$$t_m^{(i)} = \begin{cases} +\infty & \text{if } p^{(i)} < x_{\text{mid}}^{(i)} \\ -\infty & \text{if } p^{(i)} \geq x_{\text{mid}}^{(i)} \end{cases}$$

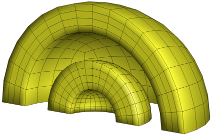
Rogues Gallery (Part I)



768 faces

3D Concentric Spheres (5k Rays)
768 faces, Trunk: 47.69s

Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	9.72s	4.9x	2452	0.063854%
20	7.99s	6.0x		
10	5.97s	8.0x		
5	4.44s	10.7x		

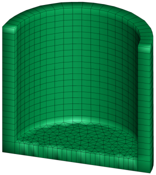


192 faces

3D Concentric Spheres, 2-Plane Symmetry (5k Rays)
192 faces, Trunk: 20.52s

Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	2.59s	7.9x	2951	0.255420%
20	2.15s	9.5x		
10	1.86s	11.0x		
5	1.51s	13.6x		

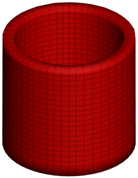
Rogues Gallery (Part II)



764 faces

3D Cylinder, 1-Plane Symmetry (10k Rays)
764 faces, Trunk: 77.22s

Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	17.16s	4.5x	8	0.000104%
20	9.03s	8.5x		
10	7.37s	10.5x		
8	5.90s	13.1x		

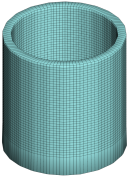


1528 faces

3D Cylinder (10k Rays)
1528 faces, Trunk: 307.05s

Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	41.41s	7.4x	16	0.000104%
20	21.83s	14.1x		
10	17.95s	17.1x		
8	14.07s	21.8x		

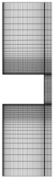
Rogues Gallery (Part III)



6414 faces

3D Cylinder, Fine Mesh (10k Rays)
6414 faces, Trunk: 9892.82s

Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	203.97s	48.5x	34	0.000053%
20	121.03s	81.5x		
10	84.40s	117.2x		
8	75.02s	131.9x		



216 faces

2D Cavity (5k Rays)
216 faces, Trunk: 3.01s

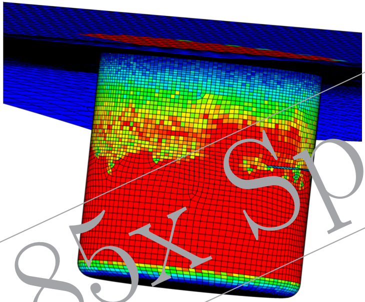
Bin Size	Run Time	Speed Up	Rays Diff.	Corrected
30	1.48s	2.0x	0	—
20	1.39s	2.2x		
10	1.27s	2.4x		
5	1.02s	3.0x		

"The Monster"

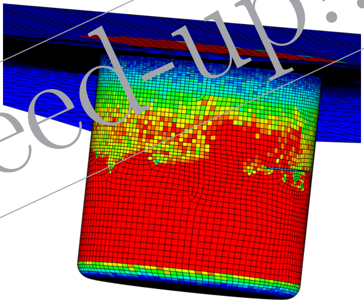
The primary motivation for doing all this work was this beast:

26,232 Faces
1,000,000 Rays

185x Speed-up!!!



"Brute Force"
96 hours
12 Restarts



Tree
31 *minutes*
0 Restarts

Summary and Conclusions

Tree-based search:

- Has three speeds: Fast, Blazing, and Ludicrous
- Numerically robust thanks to Separating Axis Test
- Pretty simple to implement