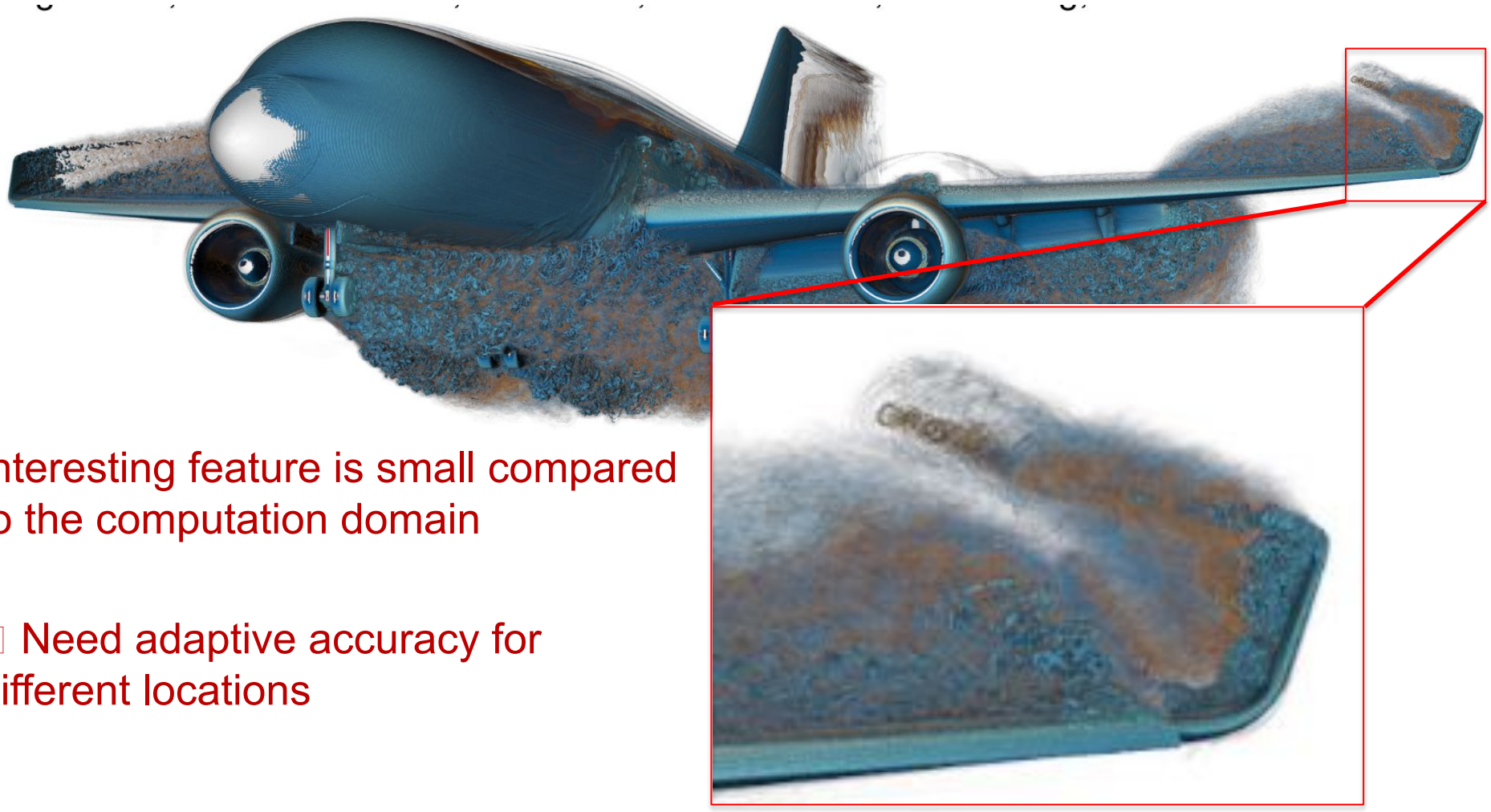
THE OHIO STATE UNIVERSITY

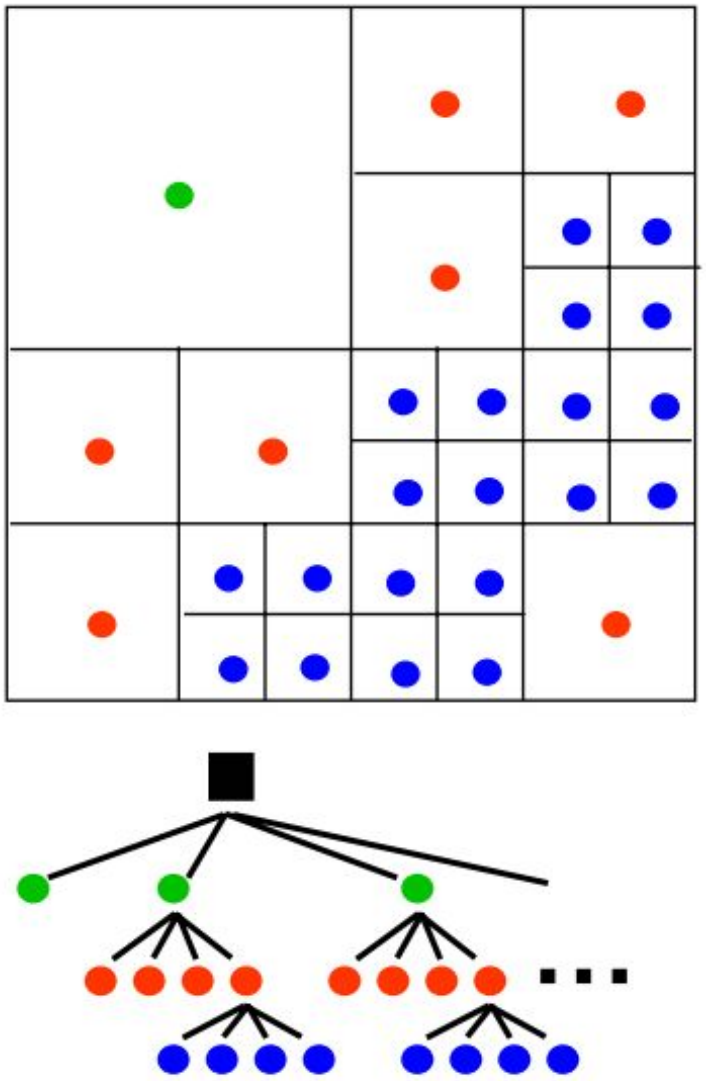
Ray Tracing Structured AMR Data Using ExaBricks

Authors: Ingo Wald, Stefan Zellmann, Will Usher, Nate Morrical,
Ulrich Lang, and Valerio Pascucci



Interesting feature is small compared to the computation domain

- Need adaptive accuracy for different locations

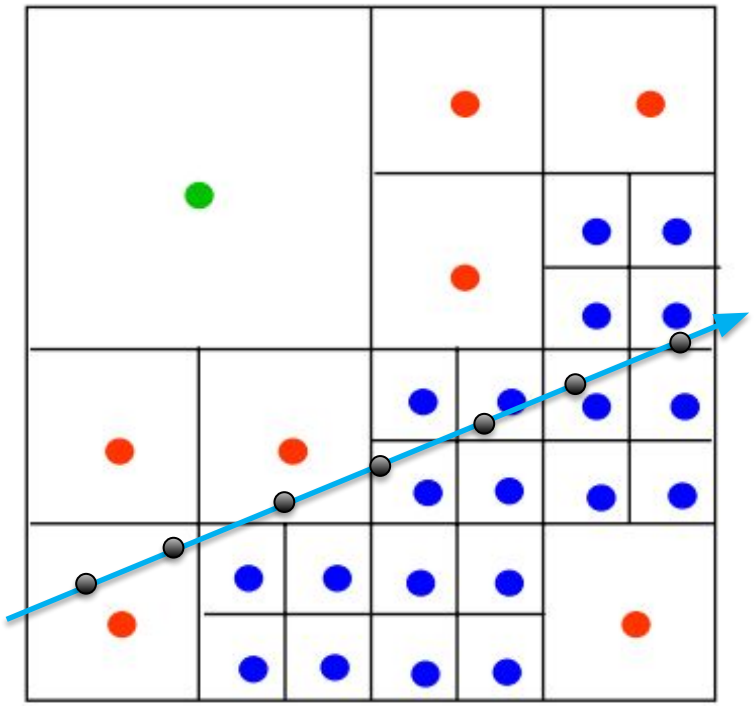


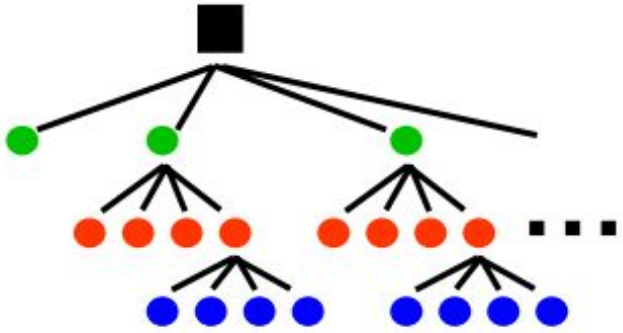
AMR (Adaptive Mesh Refinement)

Adaptive accuracy for different location

Hierarchical Structure (e.g. Octree)

- How to do ray tracing on AMR data structure?





Straight-forward way

Interpolate the value at every sample point (black dot)

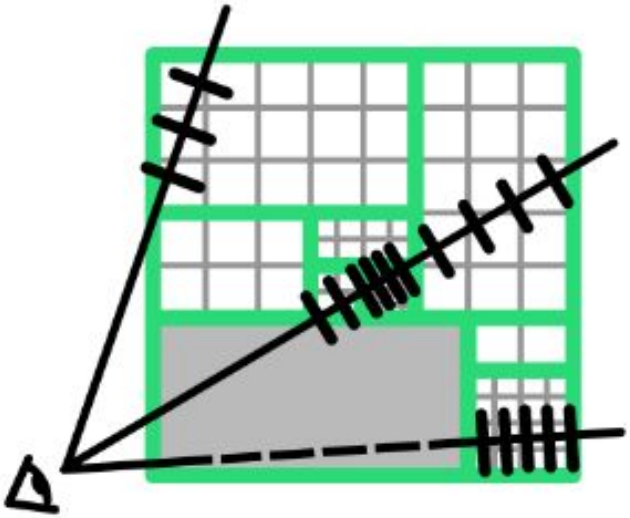
Tree traversal for every sample.

Extremely inefficient, hard to use GPU acceleration.

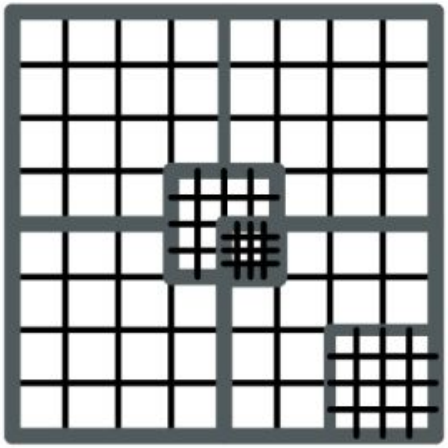
From hierarchy to bricks

Separate the cell into bricks of the same-level cell

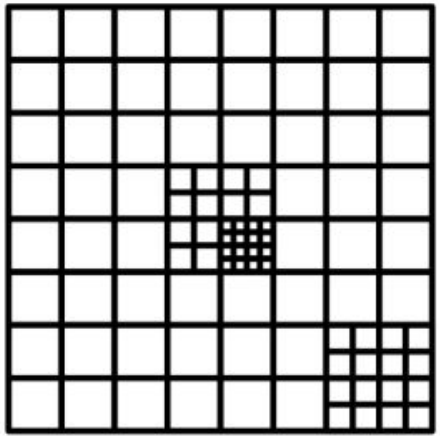
Sample values in each bricks can be evaluated efficiently.



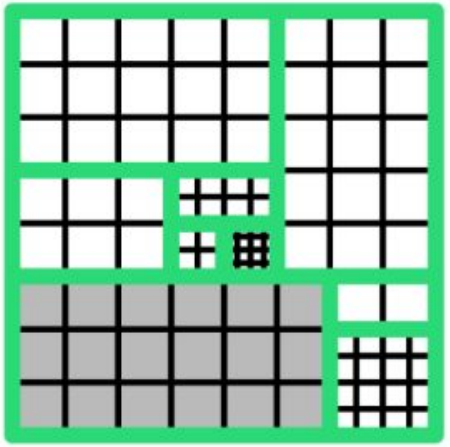
(e)



(a)



(b)



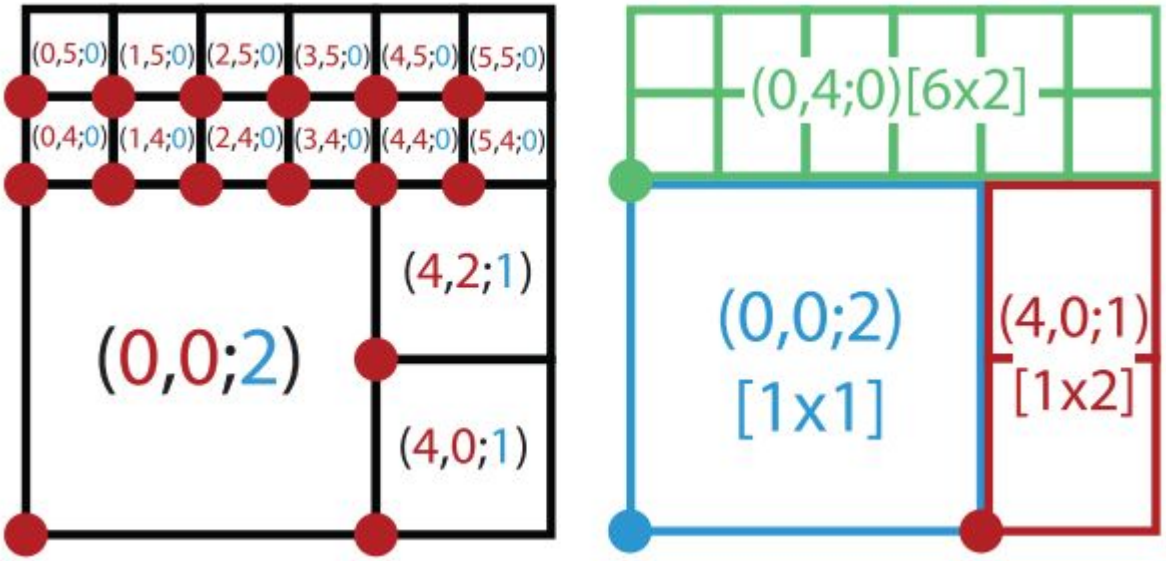
(c)

How are bricks built

Top-down built k-d tree.

Split the node if the node contain more than one AMR level.

Split along the longest axis.



Sample Reconstruction

Nearest Neighbor

- Artifacts near the level-change location

Basis method

- Filter:

$$\hat{H}_C(p) = \hat{h}\left(\frac{|C_{p_x} - p_x|}{C_w}\right)\hat{h}\left(\frac{|C_{p_y} - p_y|}{C_w}\right)\hat{h}\left(\frac{|C_{p_z} - p_z|}{C_w}\right)$$

$$B(p) = \frac{\sum_{C_i} \hat{H}_{C_i}(p) C_{v_i}}{\sum_{C_i} \hat{H}_{C_i}(p)}$$

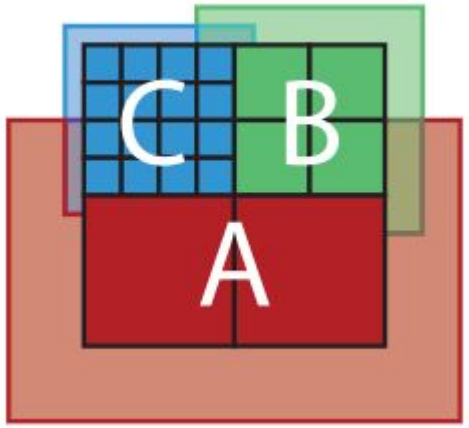
C_p Cell center, p sample location, C_w Cell Width, $\hat{h}(x) = \max(1 - x, 0)$

Active Brick Regions

Cell is active if the distance is smaller than cell width.

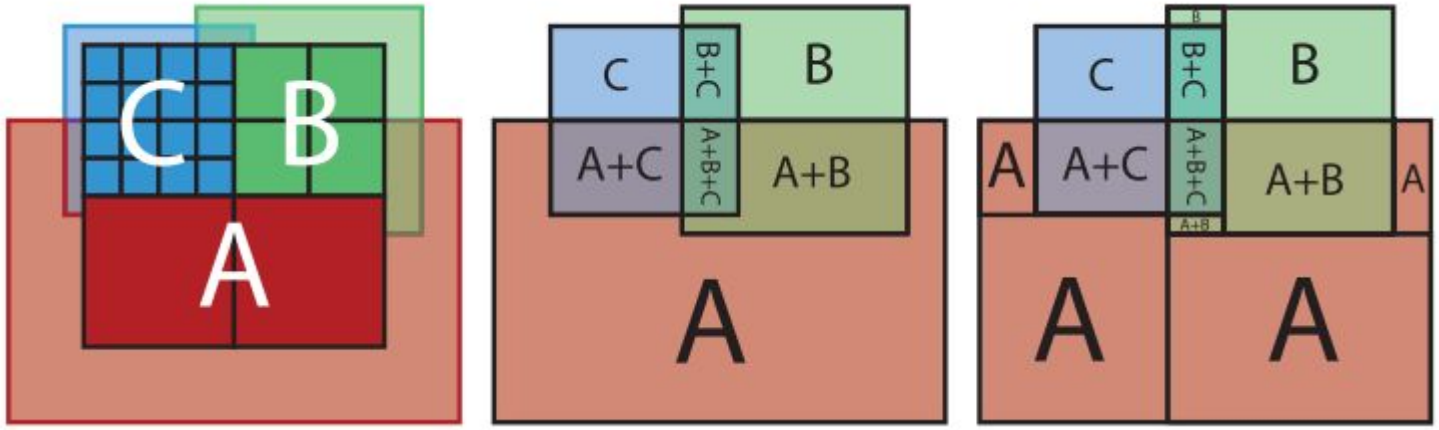
$$\hat{h}\left(\frac{|C_{p_x} - p_x|}{C_w}\right) \qquad \hat{h}(x) = \max(1 - x, 0)$$

A brick is affecting the region of brick size + ½ cell width



Efficiently Data Structure

Re-separate the bricks into Active Brick Regions.



Building Method is similar with k-d tree.

Ray Tracing and RTX BVH (Bounding Volume Hierarchy)

RTX BVH is a hardware acceleration to determine if the ray hits any bounding box of a geometry.

In last step ABR are created, BVH tree is built upon the created regions.

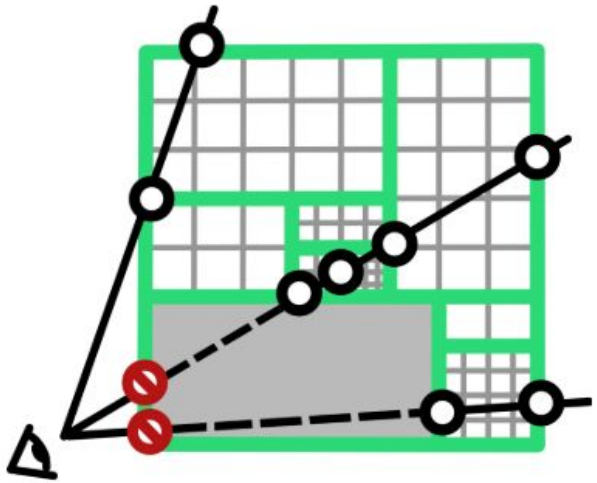
BVH details are skipped in the paper.

Space Skipping

Skip the brick which is empty or has zero max opacity.

Opacity is determined by transfer function.

- Precompute the opacity range when building ABR



The grey brick is skipped

Adaptive Sampling

Determine the sample rate on the ray based on AMR-level.

$$t_{(i+1)} = t_i + dt$$

$$dt = scale * region\_cell\_size$$

Opacity Correction

Sample rate should not affect the opacity of the region.

Remember how to calculate the color in ray tracing:

$$C_{\text{acc}} += (1 - \alpha_{\text{acc}}) \times (\alpha_{\text{sample}} C_{\text{sample}})$$

$$\alpha_{\text{acc}} += (1 - \alpha_{\text{acc}}) \times \alpha_{\text{sample}}$$

(forward)

Opacity calculation is wrong if sampling rate is different.

Correction:

$$\alpha_{corrected} = 1 - (1 - \alpha)^{\frac{s}{s_1}}$$

s is the current step size, s_1 is the base step size

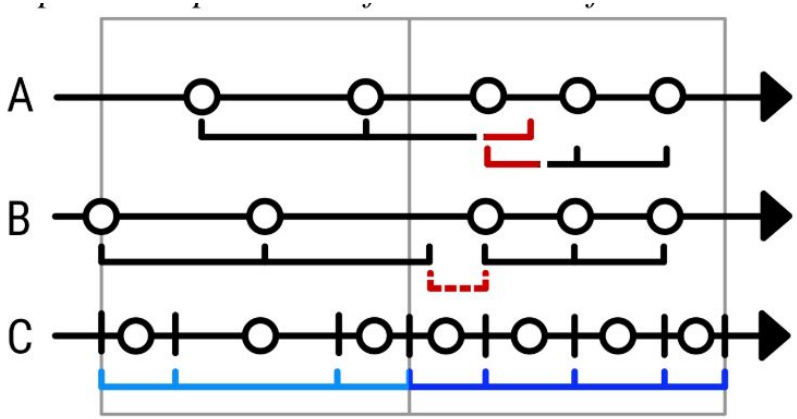
Opacity Artifacts

Due to higher or lower sample rate at the region edge, the opacity correction maybe wrong.

Fix by place node at the edge and use the interval middle point as sample

(a) w/o sample correction, $dt = 0.3$

(b) w/ sample correction, $dt = 0.3$



Evaluations

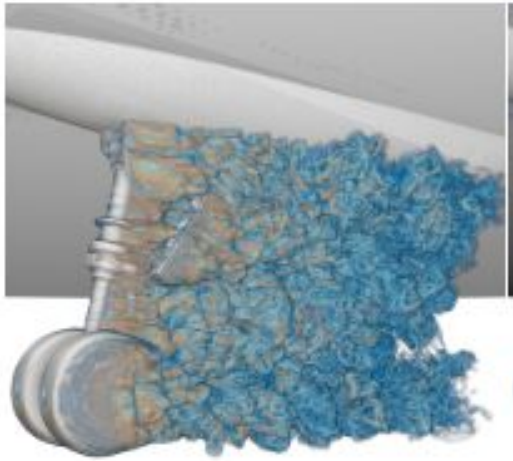
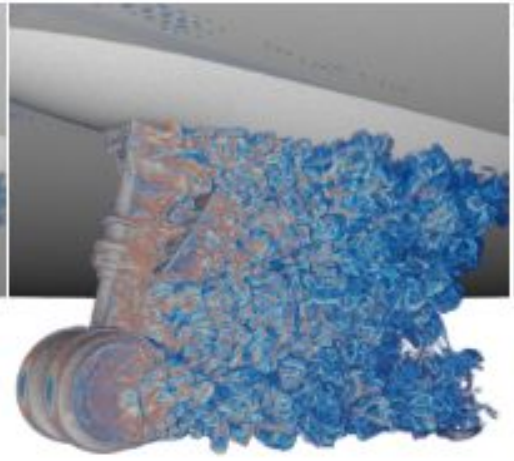
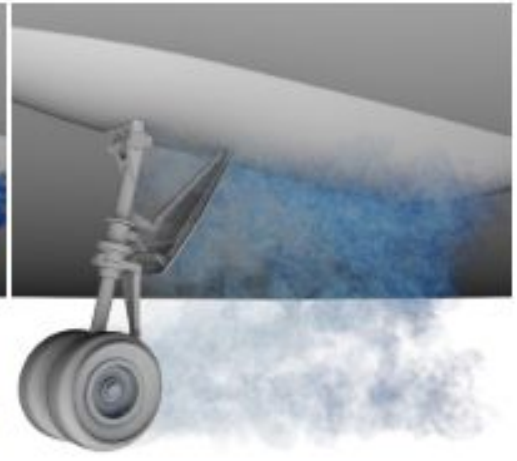
Time Comparison

Compared to other GPU ray tracing method.

Model	via cell loc. (Wald et al. [37])	from regions (ours)	speedup
Cloud	88.5	37.8	2.3×
Impact-5K	80.6	38.9	2.1×
Impact-20K	258	130	2.0×
Impact-46K	391	202	1.9×
Wind	151	75.8	2.0×
Gear	288	56.5	5.1×
Exajet (rear)	218	80.6	2.7×
Exajet (wing)	88.5	36.2	2.4×

Time Comparison

Compared to other CPU ray tracing method.

Ours (Default)	OSPRay AMR Similar Quality	OSPRay AMR Similar Perf.
		
(a) 0.06 sec	(b) 12.5 sec	(c) 0.24 sec

Memory

	Volume Data			Surface Data	Total
Model	Scalars	Bricks	Regions		
Cloud	307MB	31.8MB	1.00GB	n/a	2.06GB
Impact-5K	102MB	17.7MB	676MB	n/a	2.19GB
Impact-20K	604MB	104MB	4.82GB	n/a	12.9GB
Impact-46K	1.06GB	95.1MB	4.15GB	n/a	11.7GB
Wind	1.53GB	767KB	36.3MB	n/a	2.16GB
Gear	1.96GB	813KB	42.2MB	38.2MB	2.70GB
Exajet	2.45GB	95.0MB	2.95GB	1.52GB	13.4GB

Contribution Summary

Novel data structure that supports RTX accelerated ray tracing for AMR data.

Adaptive sampling, opacity correction etc…