SIROH: A Cost Metric for Scene-Interior Ray Origins

Bartosz Fabianowski, Colin Flower and John Dingliana
GV2 Group, Trinity College, Dublin, Ireland

Abstract

The highest ray tracing speeds are currently achieved by using acceleration structures built according to the Surface Area Heuristic. We present a heuristic based on more realistic assumptions that yields even better acceleration in many scenes.

Introduction

Ray tracing speed crucially depends on an acceleration structure to reduce the number of intersection tests per ray. The best such structures are obtained by top-down construction, minimizing a cost metric to determine where to split a parent node P:

$$C = p_P \cdot C_T + p_L \cdot N_L \cdot C_I + p_R \cdot N_R \cdot C_I$$

For each potential split, N_L, N_R are the numbers of primitives in the child nodes, C_T is the cost of traversing an inner node and C_I the cost of intersecting a single primitive. To compute the total, the probabilities p_P, p_L, p_R of a ray visiting each node must be approximated. The Surface Area Heuristic (SAH) [MB90] makes three assumptions for this purpose:

- Ray origins are uniformly distributed outside the scene's bounding box S
- Ray directions are uniformly distributed
- Rays miss all primitives

Under these assumptions, the probability of a node N being visited is the ratio of the surface areas of N and S.

Our Assumptions

We make the observation that assuming rays to originate *inside* the scene is more realistic. Reflected and refracted rays always begin at surfaces within the scene. For indoor scenes, camera and light sources are also commonly located inside the scene's bounding box. We therefore change the first assumption into:

- Ray origins are uniformly distributed *inside* the scene's bounding box S

SIROH

A ray may now originate either inside node N or in the remainder of the scene, $S \setminus N$. In the first case, the probability of visiting N is 1 and in the latter, $\alpha_{\vec{x}}/4\pi$, the fractional solid angle subtended by N at the ray's origin $\vec{x}$. As the integral of $\alpha_{\vec{x}}$ over $S \setminus N$ has no closed-form solution, we propose an approximation.

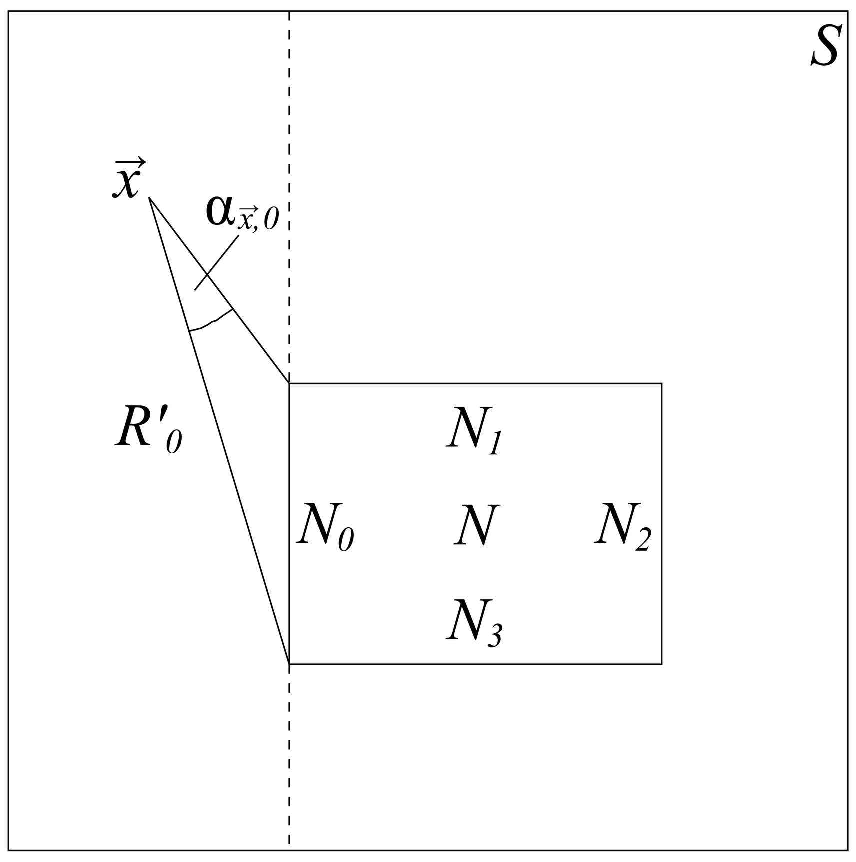

FIGURE 1: SIROH approximation

We divide $S \setminus N$ into six overlapping regions R'_i, each containing all potential origins $\vec{x}$ from which a face N_i of node N can be seen. Throughout every R'_i, the fractional solid angle $\alpha_{\vec{x},i}/4\pi$ subtended by N_i should be integrated. Taking inspiration from the SAH, we approximate the fraction as a ratio of surface areas instead:

$$p_N \approx \frac{V(N)}{V(S)} + \frac{1}{V(S)} \sum_i V\left(R'_i\right) \frac{SA(N_i)}{SA\left(R'_i\right)}$$

Results

We compare the frame rates of our CUDA-based ray tracer using acceleration structures built according to the SAH and SIROH for a number of benchmark scenes. All figures are obtained using a 512×512 viewport on an NVidia GeForce 280GTX, averaging over a flight through each scene. SIROH can be seen to improve the frame rate by an average of 3.2%, only one scene showing no speed-up over the SAH.

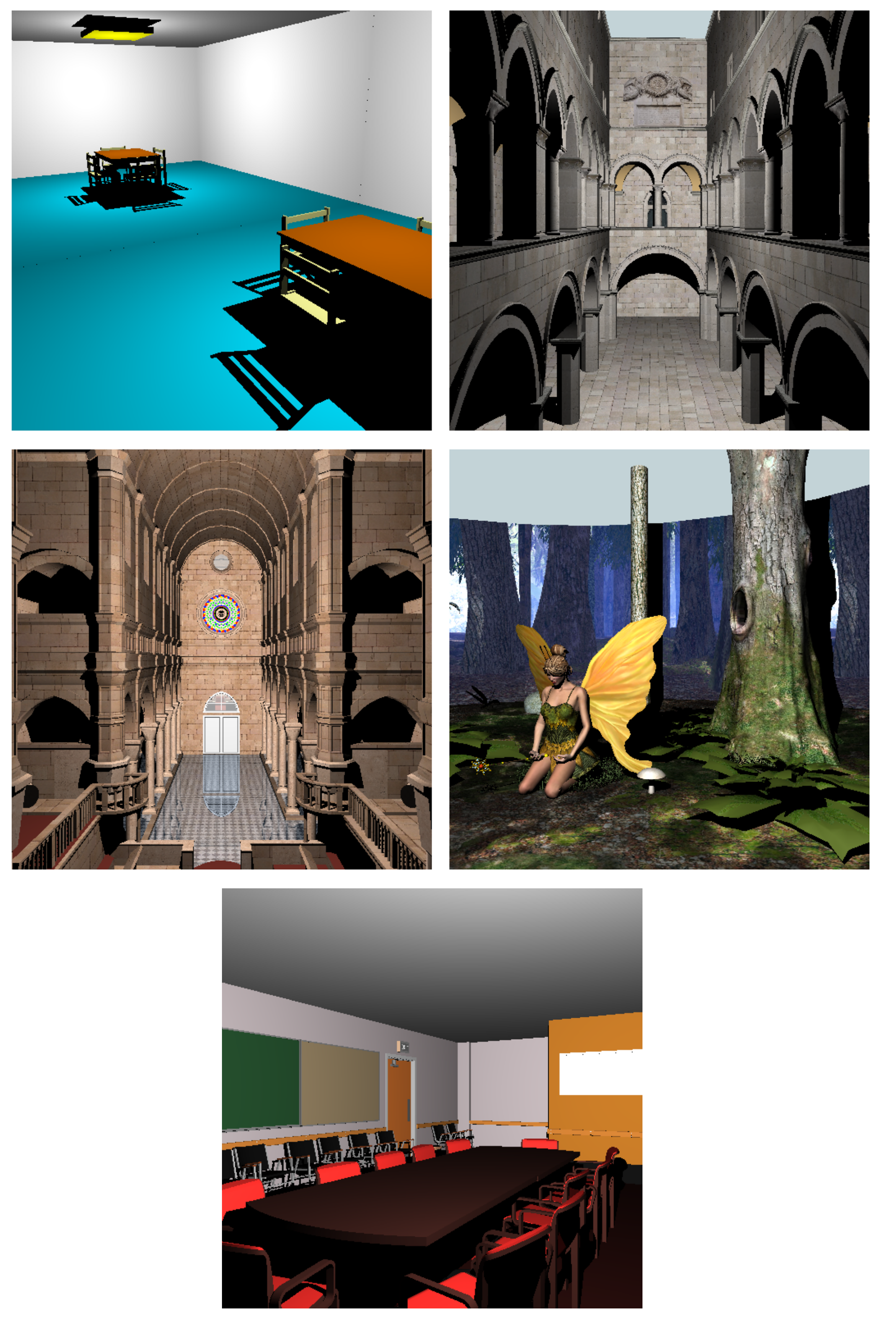

FIGURE 2: Benchmark scenes

Scene	SAH	SIROH			
	FPS			Construction	
Scene 6	382.2	409.1	**+7.0%**	0.01	+7.4%
Sponza	99.6	99.6	**0.0%**	0.96	+5.8%
Sibenik	101.0	103.1	**+2.1%**	0.97	+2.3%
Fairy	75.9	78.0	**+2.7%**	3.85	+13.0%
Conference	71.9	74.9	**+4.2%**	3.50	+7.6%
Average			**+3.2%**		+7.2%
Std. Dev.			2.6%		3.9%

Building the acceleration structure has the same complexity as with the SAH and on average takes 7.2% longer. For more details of the SIROH heuristic and experiments confirming the results with other ray tracers, see our paper at [FFD09].

Acknowledgments

The first two authors are funded by IRCSET under the Embark Initiative, the second additionally by Intel Corporation.

References

[FFD09] B. Fabianowski, C. Flower, and J. Dingliana. A cost metric for scene-interior ray origins. In *Eurographics Short Papers*, pages 49–52, 2009.

[MB90] J.D. MacDonald and K.S. Booth. Heuristics for ray tracing using space subdivision. *The Visual Computer*, 6(3):153–166, 1990.