A Shadow Culling Algorithm for Interactive Ray Tracing

Jae-ho Nah, Kyung-ho Lee, Woo-chan Park, Tack-don Han
Yonsei University, Korea
E-mail : jhnah@msl.yonsei.ac.kr

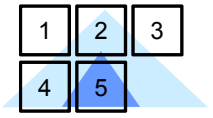
Abstract

We present a novel shadow culling algorithm in ray tracing. For interactive ray tracing, our approach exploits frame-to-frame coherence instead of preprocessing of building shadow data. In this algorithm, shadow results are stored to each primitive and used in the next frames. We also present a novel occlusion testing method to guarantee exact shadows. In experiments with BART scenes, our algorithm shows 7-19 percent reduction in cost of traversal and 9-24 percent reduction in cost of intersection test.

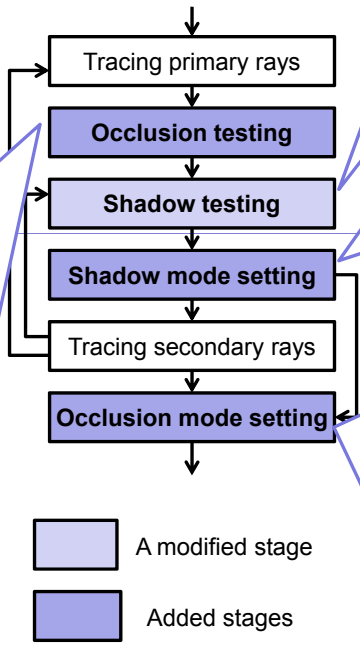
KEYWORDS: Ray tracing, real-time rendering, shadow algorithm

Overall Flow of the Shadow Culling Algorithm

- Why this occlusion testing is needed?
  - Updating shadow results is valid if and only if the primitive is not occluded on the view point
- Algorithm details
  (1) Compare primitives between the current pixel and two reference pixels.
  (2) If the primitives are different, an intersection test is executed between the current primary ray and the primitive on the reference pixel.
  (3) If the hit result is true, the primitive on the reference pixel is partially occluded by the primitive on the current pixel.
- Example



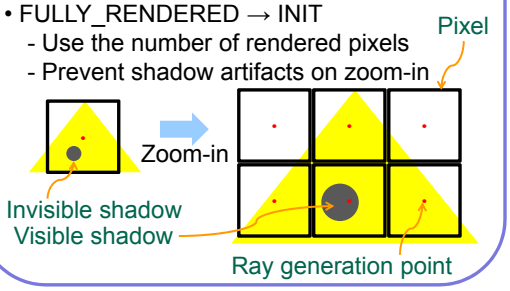
- The current pixel : 5
- Reference pixels : 2 and 4
- Result : The faint blue triangle is partially occluded



- Detect shadows and execute the culling
- FULLY_RENDERED && FULL_SHADOW
  - Shadow tracing is skipped
- FULLY_RENDERED && NON_SHADOW
  - Shadow tracing is executed about only dynamic objects

- Simple OR operation
  - current shadow mode | new shadow mode
  - ex) 11(PARTIAL_SHADOW) = 01(FULL_SHADOW) | 10(NON_SHADOW)

- Set occlusion modes of rendered primitives
- INIT→ FULLY_RENDERED
  - Non-occluded primitive
- FULLY_RENDERED → INIT
  - Use the number of rendered pixels
  - Prevent shadow artifacts on zoom-in



Required Addition Information

- An occlusion mode (2bit)
- A shadow mode (2bit*light sources) { can be merged to a 32bit integer in less than ten lights

Occlusion mode (value)	
	INIT(00)
	PARTIALLY_ OCCLUDED(01)
	FULLY_ RENDERED(10)
	DYNAMIC(11)

Shadow mode (value)	
	INIT(00)
	FULL_SHADOW (01)
	NON_SHADOW (10)
	PARTIAL_ SHADOW (11)

- The number of current rendered pixels (1 integer) { used in the occlusion mode setting stage
- The highest number of rendered pixels (1 integer)

Benchmark Results

Scene	Cost reduction ratio of shadow ray traversal		Cost reduction ratio of shadow ray/primitive intersection test		Additional intersection tests for occlusion testing
	Avg.	Max.	Avg.	Max.	
BART Kitchen 	19.8%	59.5% (20th frame)	24.5%	59.8% (19th frame)	0.173 per pixel
BART Robot 	7.4%	19.5% (31th frame)	9.5%	29.6% (41th frame)	0.149 per pixel

- 512*512 resolution, kd-tree ray tracer

Future Works

- Apply shadow caching – reduce the calculation of partially occluded primitives
- Apply triangle subdivision – reduce the ratio of primitives which have partial shadows
- Support soft shadows

Acknowledgment

This work was supported by the forth stage of the Brain Korea 21 Project in 2009.