DXR Photon Mapper

Anantha Srinivas
Connie Chang

Acknowledgement

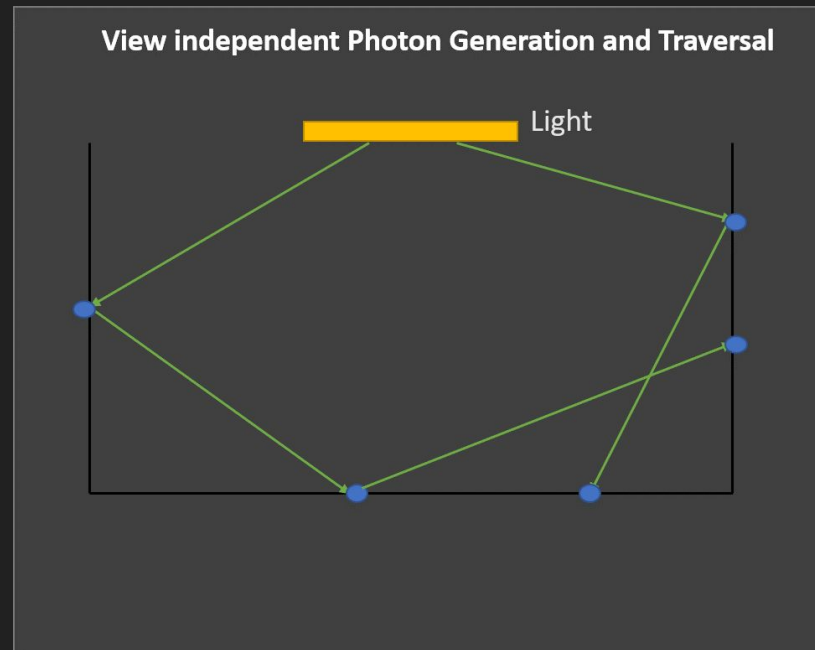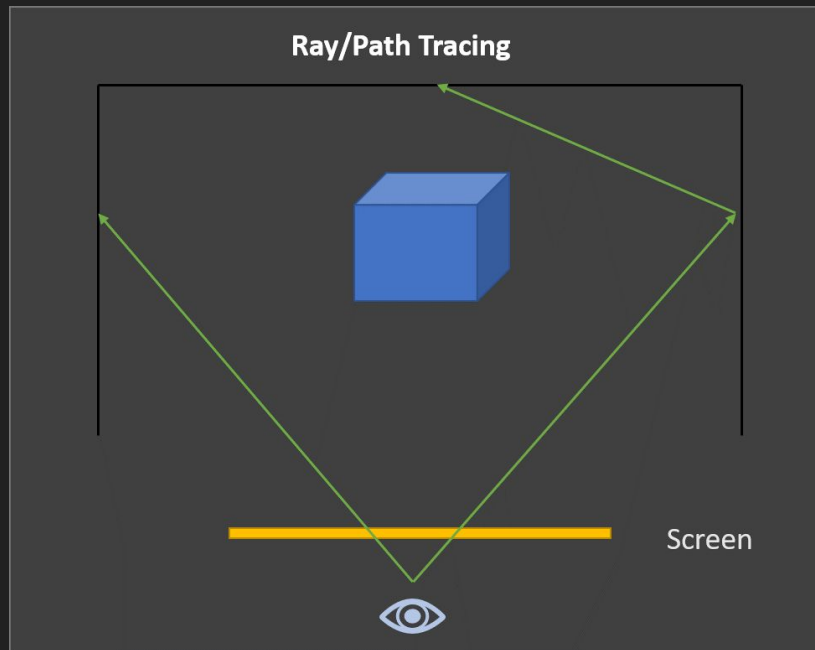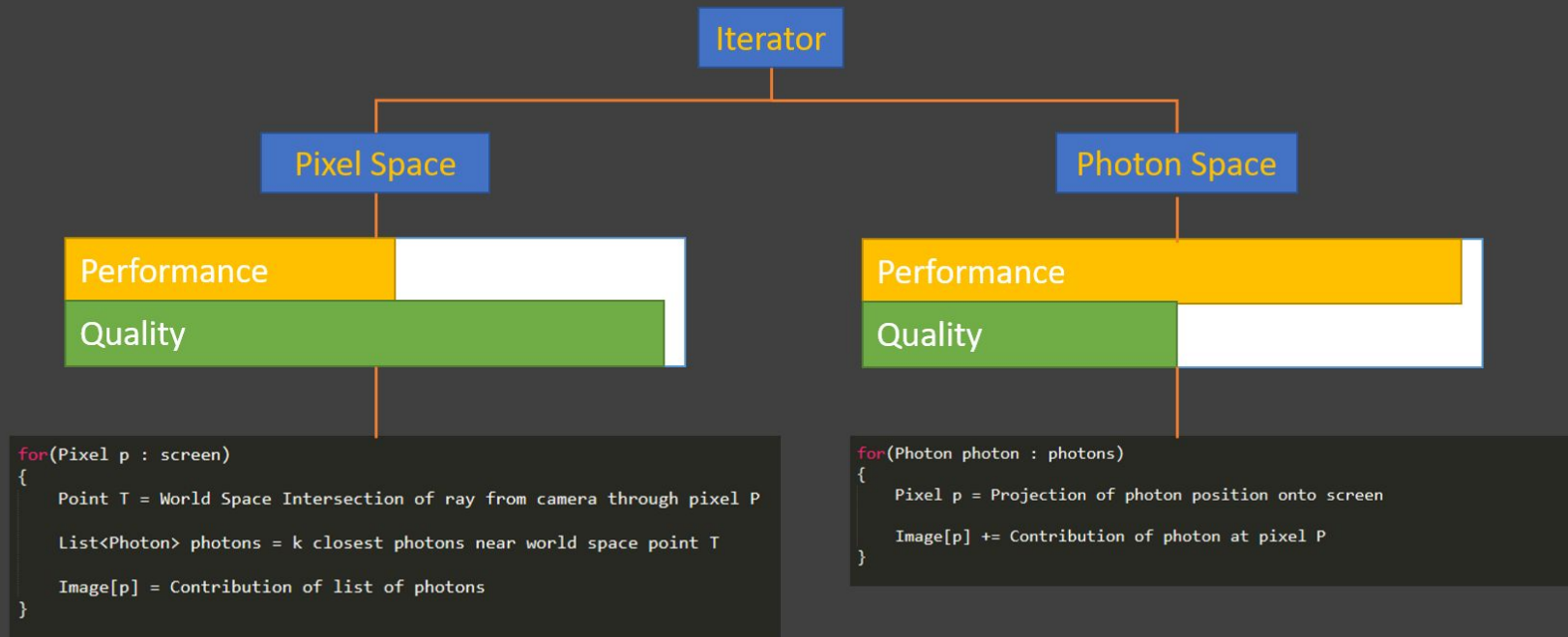# Goals of the project

- ❖ ***Implementation of Photon Mapping using DXR***
- ❖ ***Possible search time optimizations***
- ❖ ***Alternate approaches to spatial data structure construction***
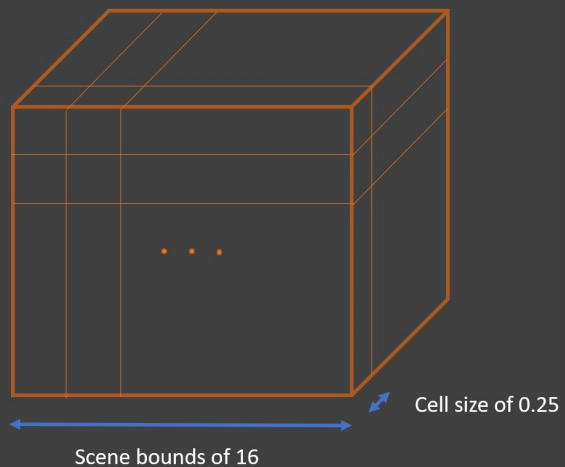
Ray Tracing vs Photon Mapping

Pixel Major vs Photon Major Iterator

Iterator

Pixel Space

Photon Space

Performance

Quality

Performance

Quality

```
for(Pixel p : screen)
{
    Point T = World Space Intersection of ray from camera through pixel P

    List<Photon> photons = k closest photons near world space point T

    Image[p] = Contribution of list of photons
}
```

```
for(Photon photon : photons)
{
    Pixel p = Projection of photon position onto screen

    Image[p] += Contribution of photon at pixel P
}
```

Pixel Major Iterator

Scene Division into cell structure

Cell size of 0.25

Scene bounds of 16

Customization of photon storage

Photon Count per cell

2	0	1	3	5	• • •

Exclusive Scan on Count – Starting index for sorted photons

0	2	2	3	6	• • •

Photon Map

P1	P2	P3	P4	P5	• • •

Photon Major Iterator

View independent Photon Generation and Traversal

Light

Shadow Ray Culling

Camera

Screen
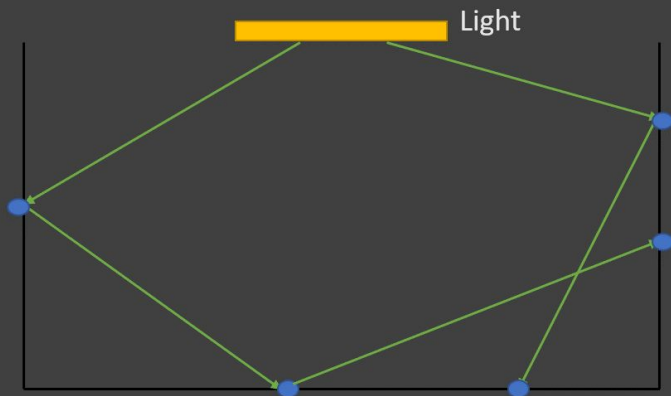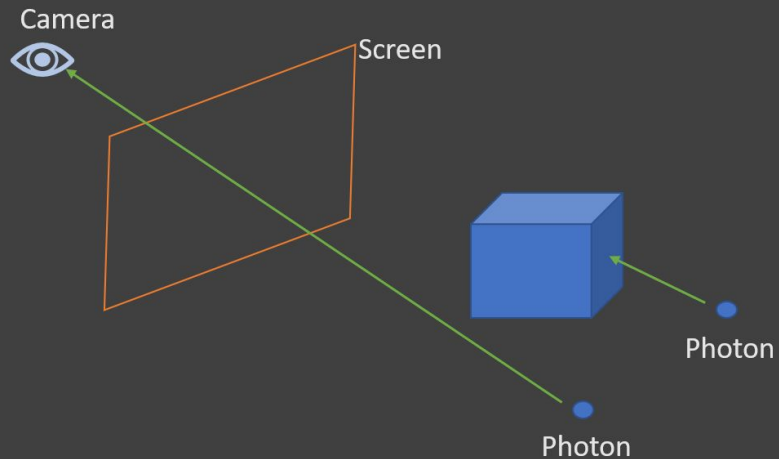
Photon

Photon
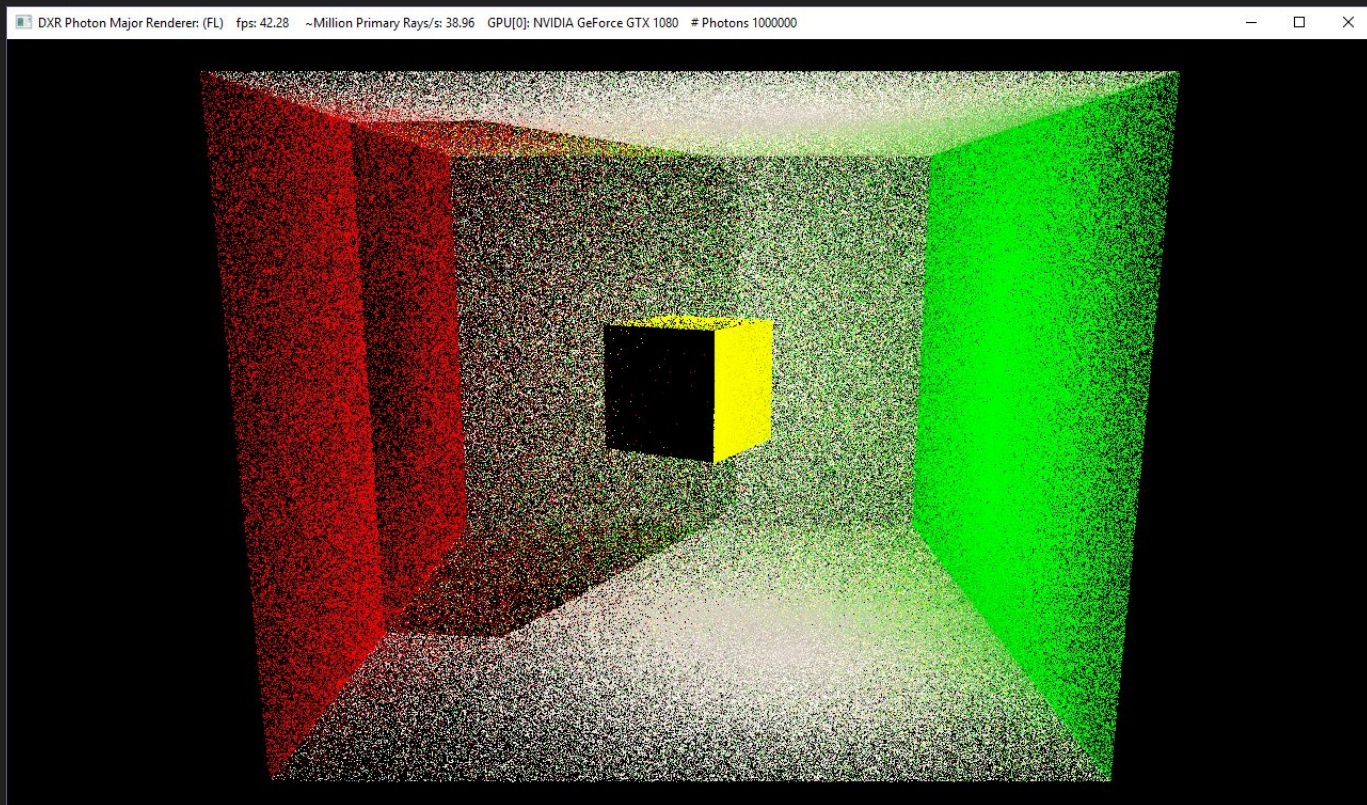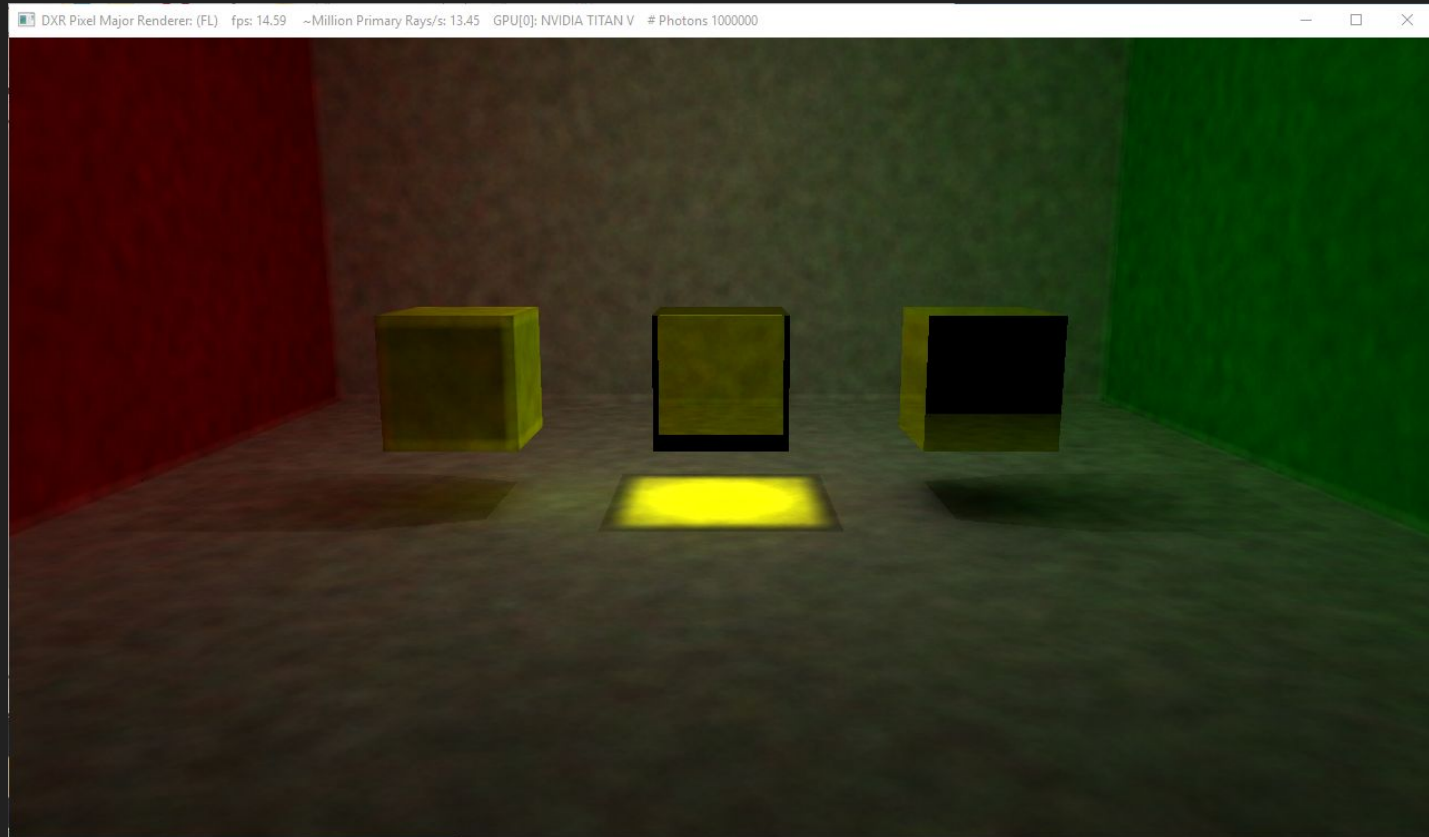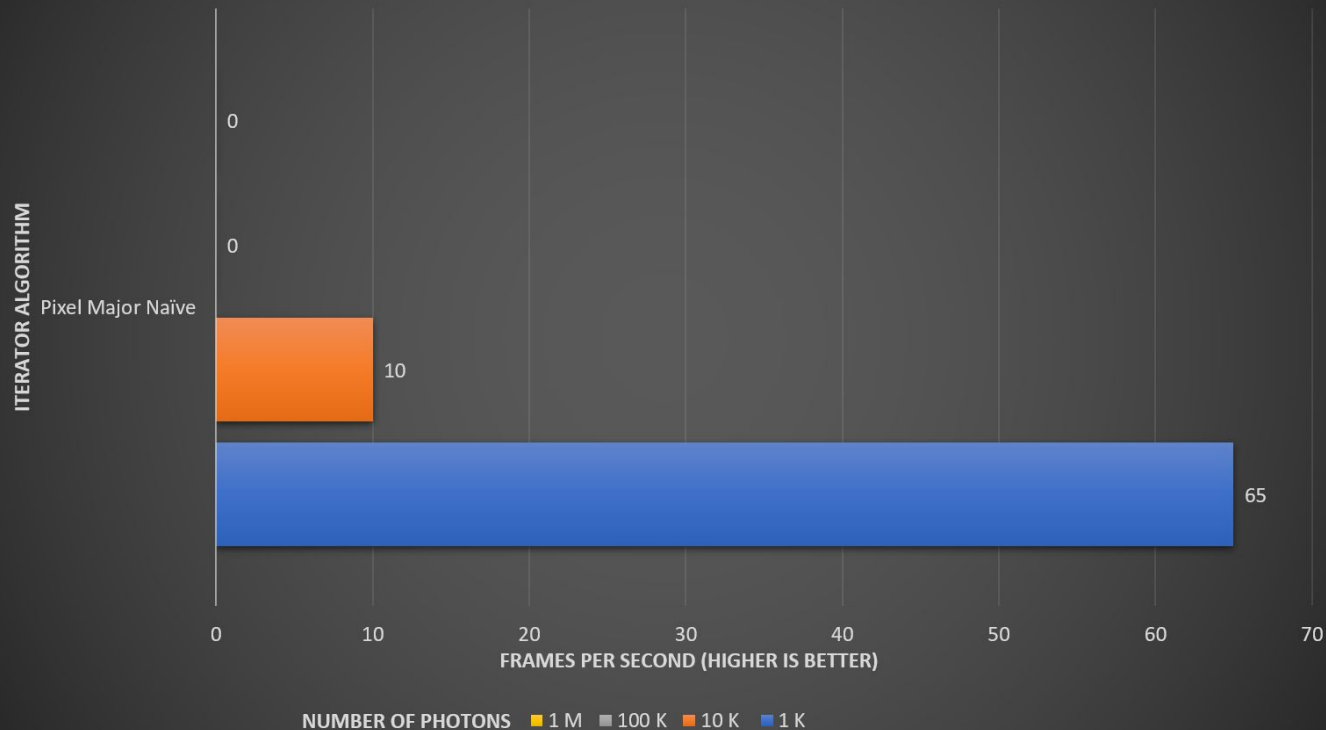
Pixel Major Results
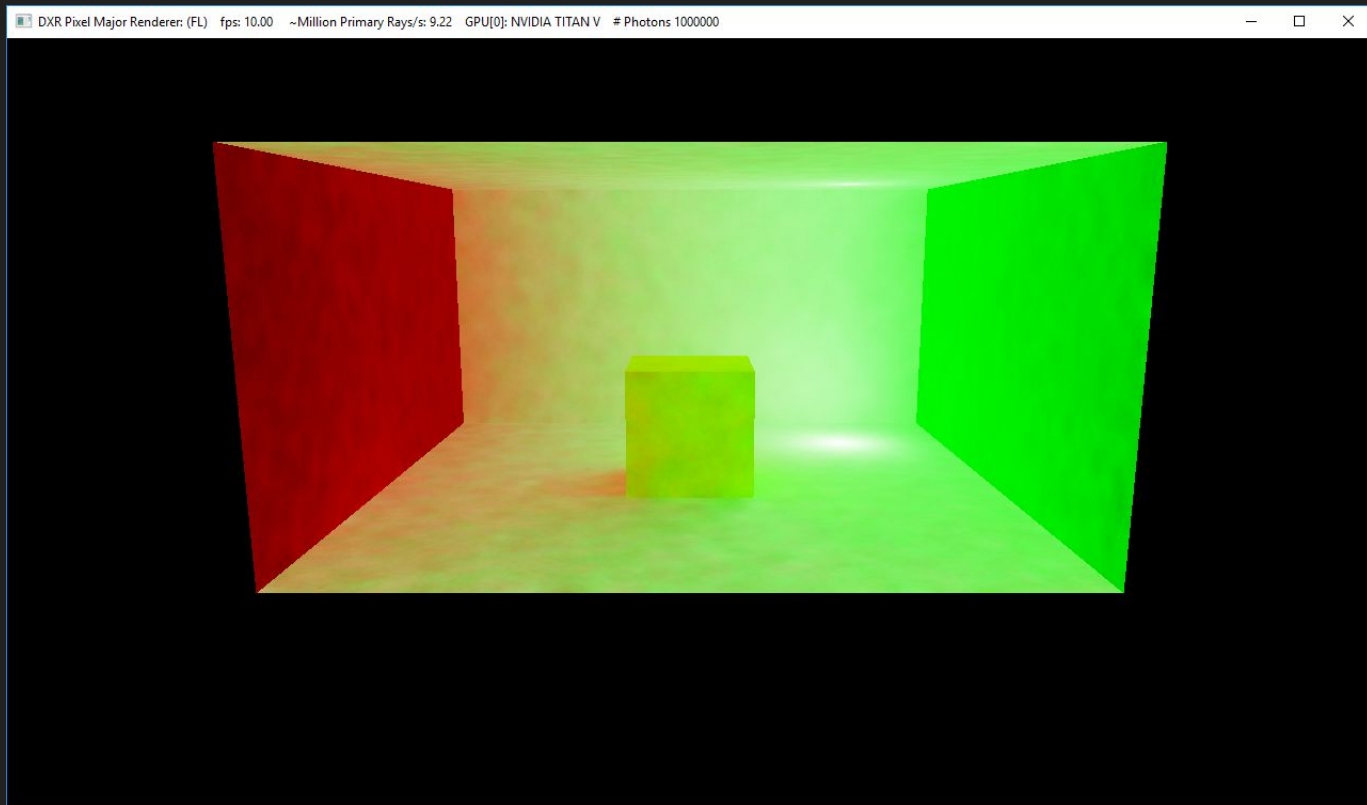
Photon Major Iterator results

Perfectly Specular materials

Performance Analysis - Pixel Major Naive
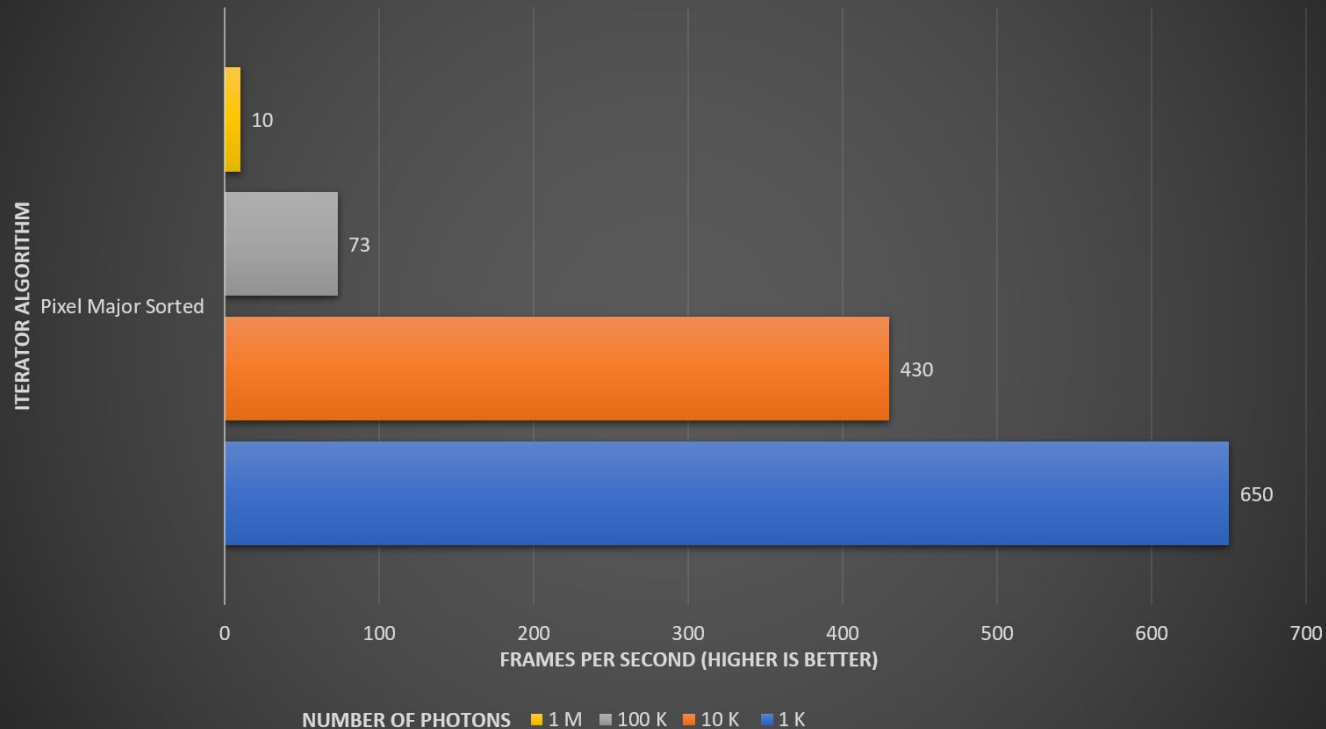

Performance of Naïve Search

Test Conditions

- NVIDIA Titan X
- DirectX Fallback Layer
- V-Sync Off
- 8 Bounces per Photon

Performance Analysis - Pixel Major Sorted

Performance Pixel Major – Sorted Photons

ITERATOR ALGORITHM

Pixel Major Sorted
- 1 M: 10
- 100 K: 73
- 10 K: 430
- 1 K: 650

FRAMES PER SECOND (HIGHER IS BETTER)

0   100   200   300   400   500   600   700

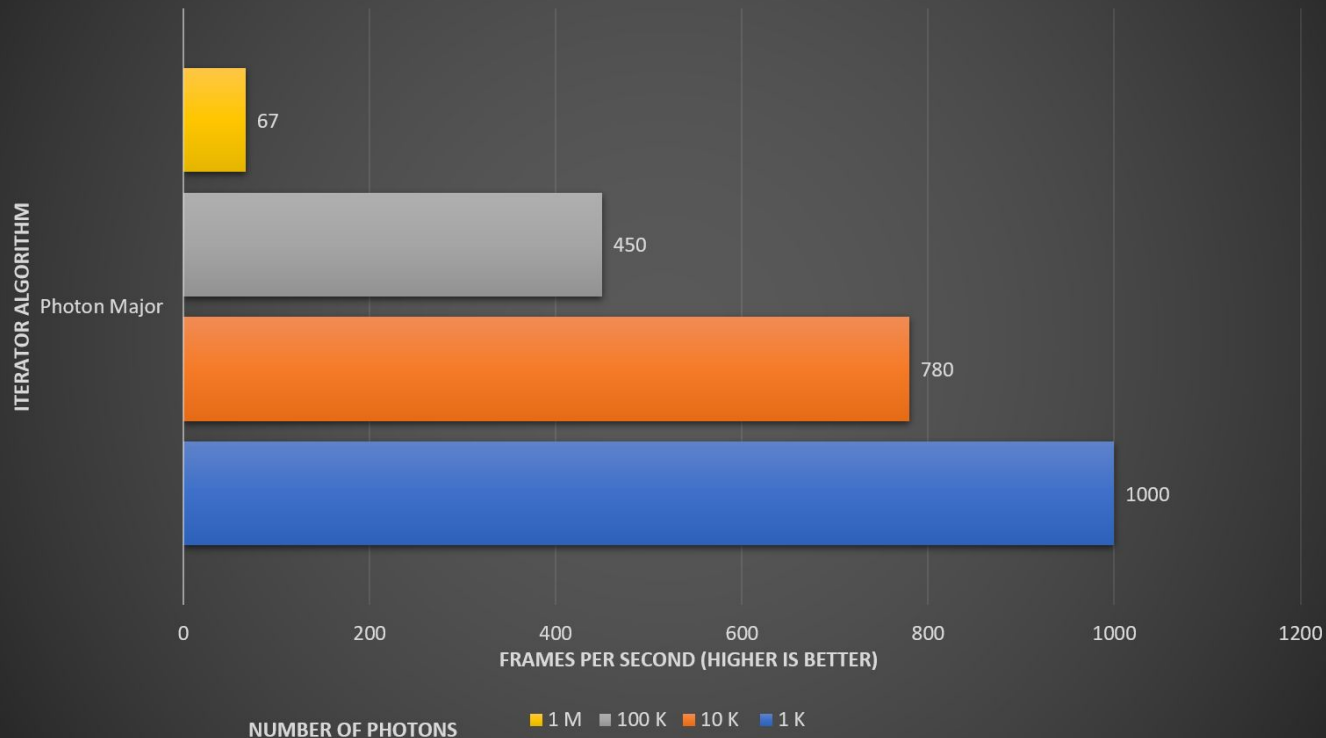**NUMBER OF PHOTONS** ■ 1 M ■ 100 K ■ 10 K ■ 1 K

Test Conditions

- NVIDIA Titan X
- DirectX Fallback Layer
- V-Sync Off
- 8 Bounces per Photon

Performance Analysis - Photon Major

Performance of Photon Major Iterator

Chart axis: ITERATOR ALGORITHM (vertical), FRAMES PER SECOND (HIGHER IS BETTER) (horizontal)

Photon Major:
- 1 M: 67
- 100 K: 450
- 10 K: 780
- 1 K: 1000

NUMBER OF PHOTONS — ■ 1 M ■ 100 K ■ 10 K ■ 1 K

Test Conditions

- NVIDIA Titan X
- DirectX Fallback Layer
- V-Sync Off
- 8 Bounces per Photon

Possible Future research?

1) Improve sorting and searching of photon -
   a) Better Memory Management
   b) Faster search
2) Denoising in Photon Major

Questions?