
Real-time Application of LTC for Shadow Rendering

신은혁 - Eun Hyouk Shin
조인영 - In Young Cho

Overview

- 1. Limitations of LTC**
- 2. Project Scope: Shadow Rendering**
- 3. Our Solution**
 - a. Geometric Approach**
 - b. Practical Problem & Solution**
- 4. Results: Live Demo**
- 5. Practical Benefits & Drawbacks**
- 6. Roles of Each Member**

Limitations of LTC

- **No shadows**

- Not considered a **visibility** term ($V = 1$, always)
- No obstacles between light and material

$$\int_{\Omega} \text{Light} \times \text{BRDF} \times \text{Visibility}$$

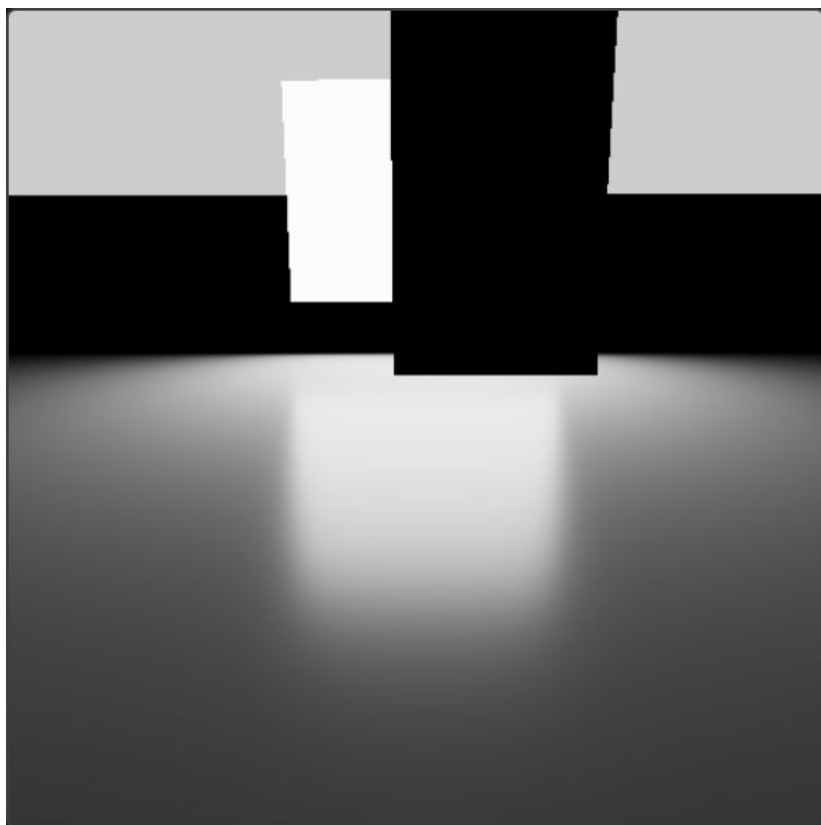


Our Project Scope

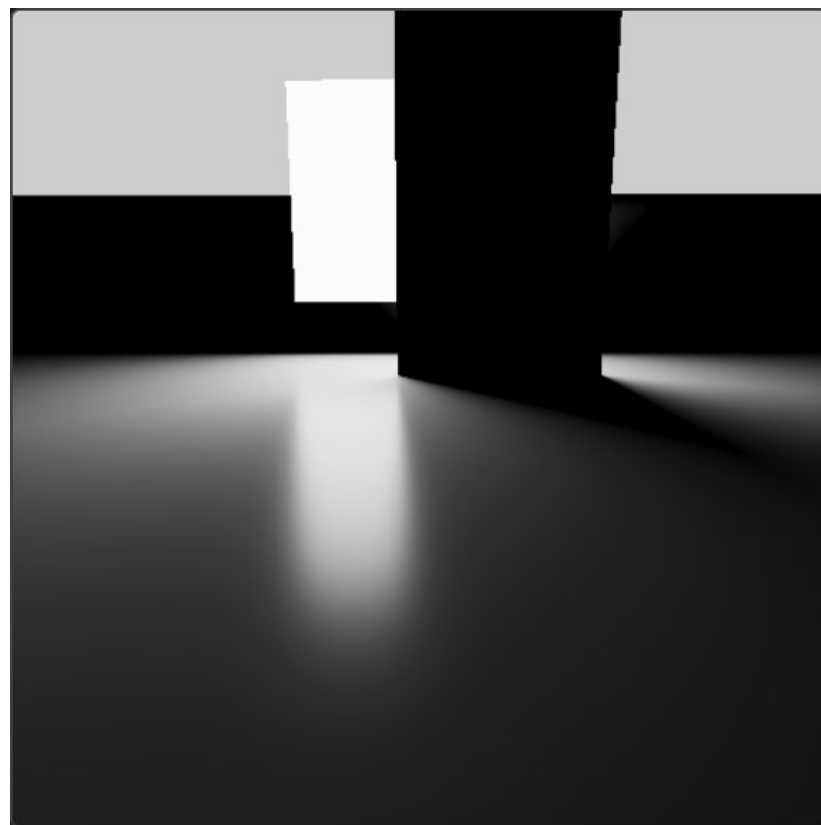
“Efficient **Shadow** Rendering Using LTC”

Result Overview

Baseline(LTC paper)

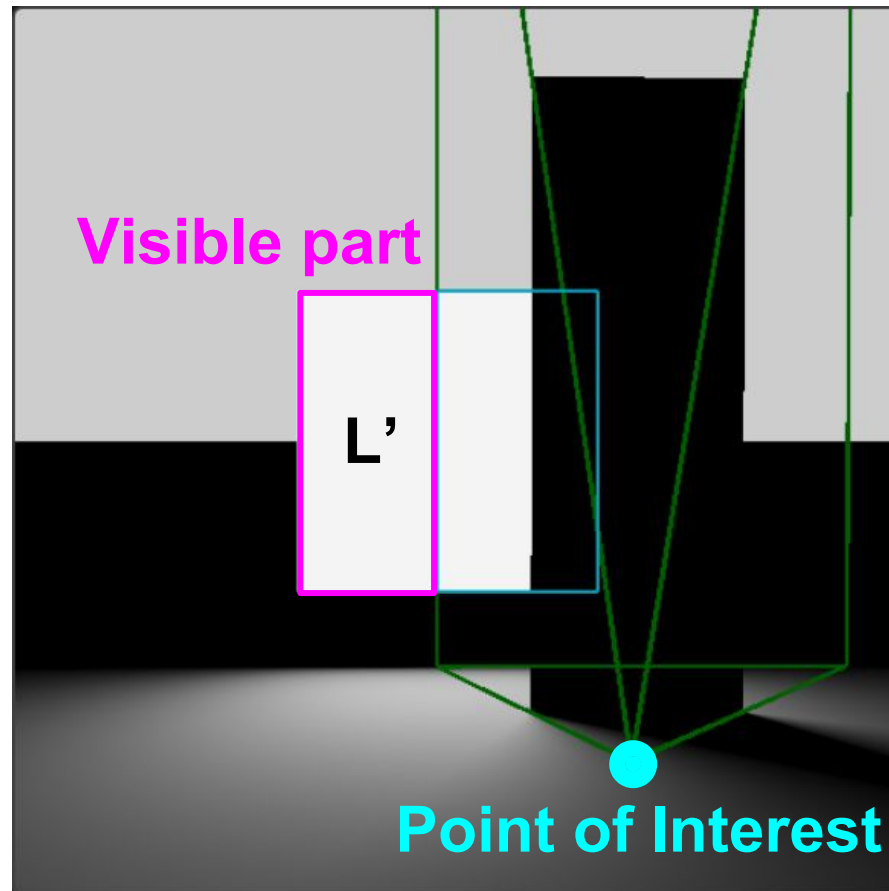


Our improvement!



Our Idea: Geometric Approach

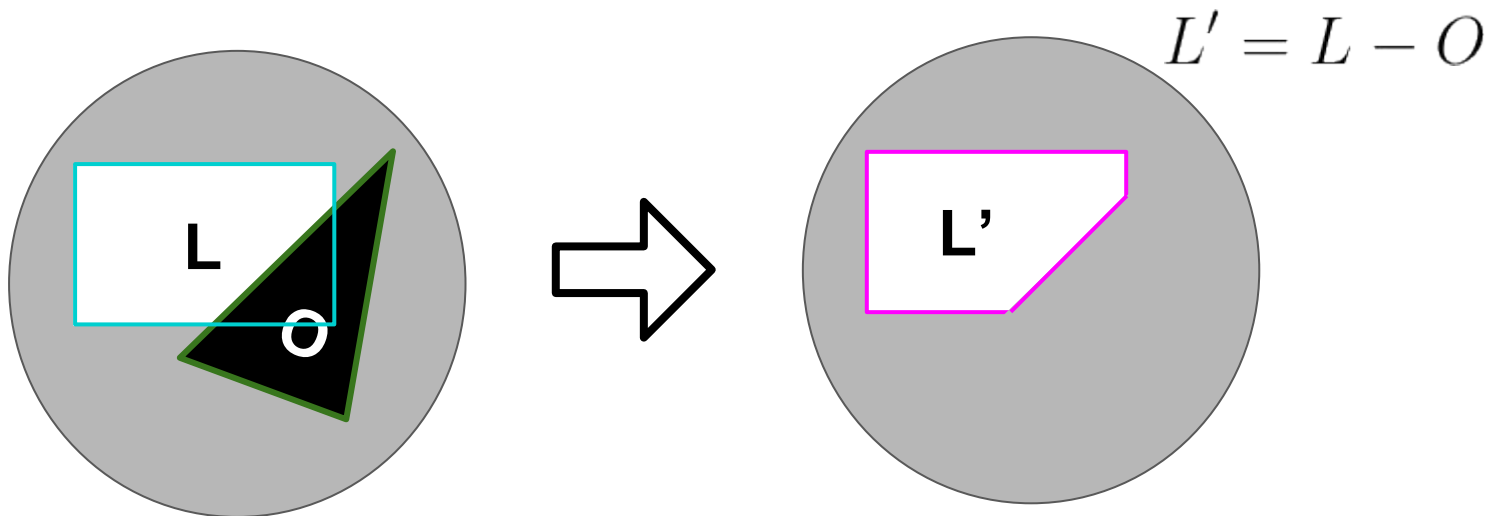
- Limit the integration domain to **visible** area light



Our Idea: Geometric Approach

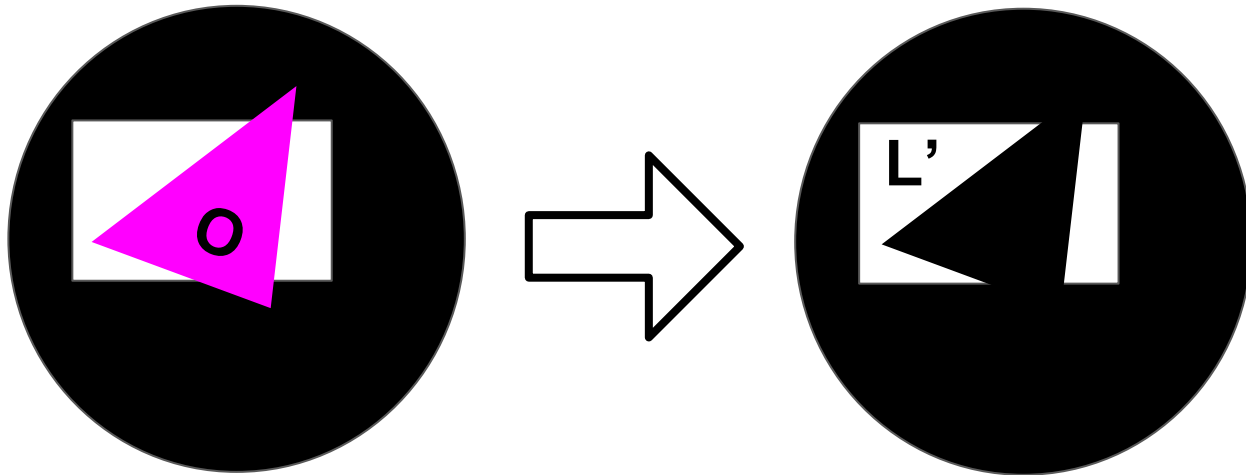
- Limiting integration domain is equivalent to applying visibility term

$$\int_L \text{Radiance} \times \text{BRDF} \times \text{Visibility} = \int_{L-O} \text{Radiance} \times \text{BRDF}$$



Practical Problem: Complex Shape

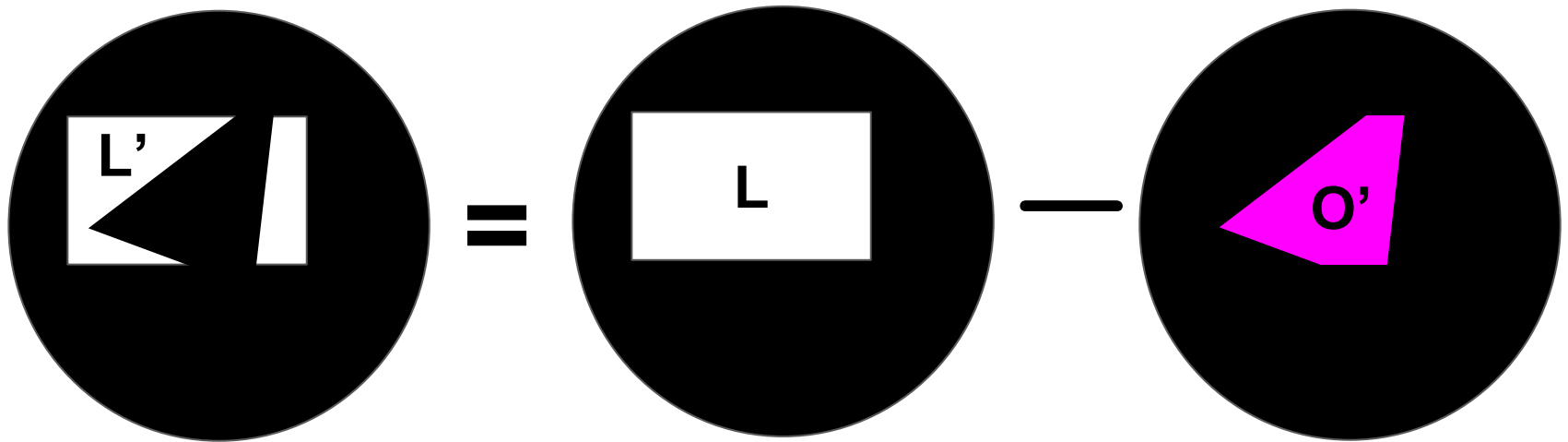
- **Hard** to compute the integral **over L'**



- **Visible light (L') can be non-convex, **not simply connected****

Solution: Separate Integration

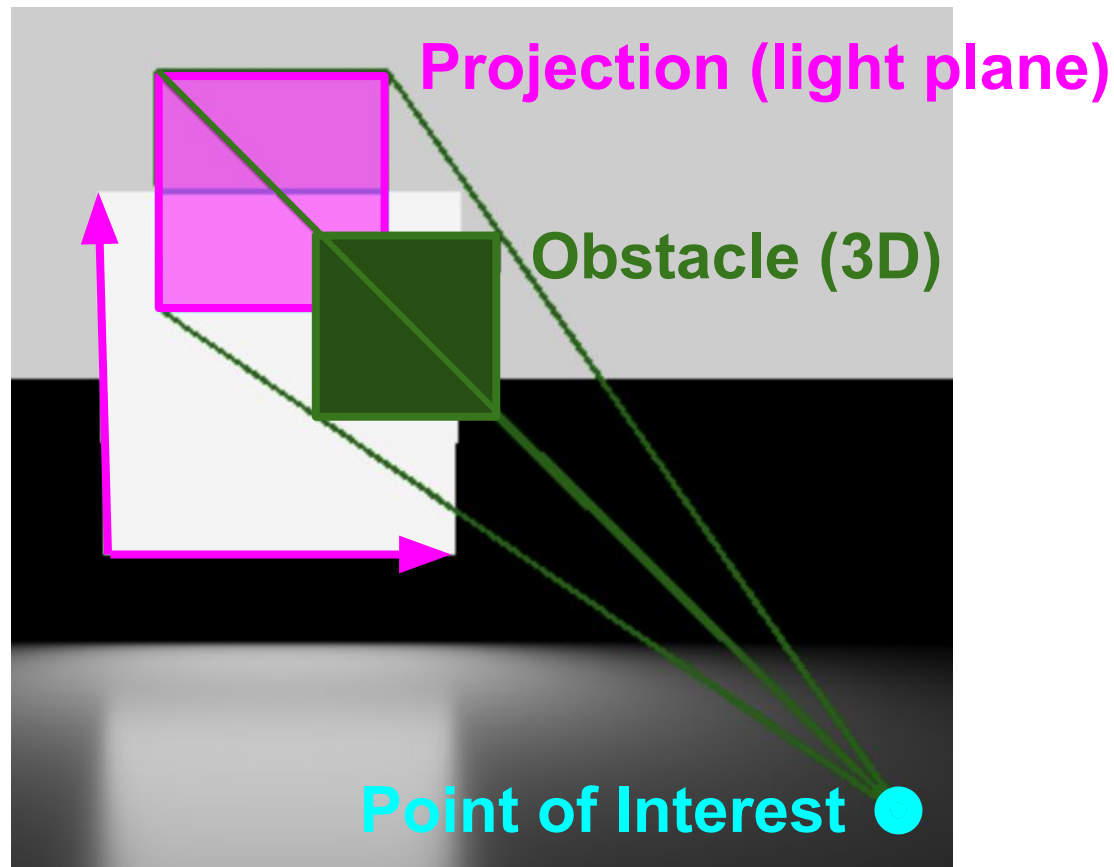
- Integrate separately, then calculate difference
 - 1. Original light
 - 2. Projected clipped obstacle (simpler shape)



$$\int_{L'} = \int_{L-O'} = \int_L - \int_{O'}$$

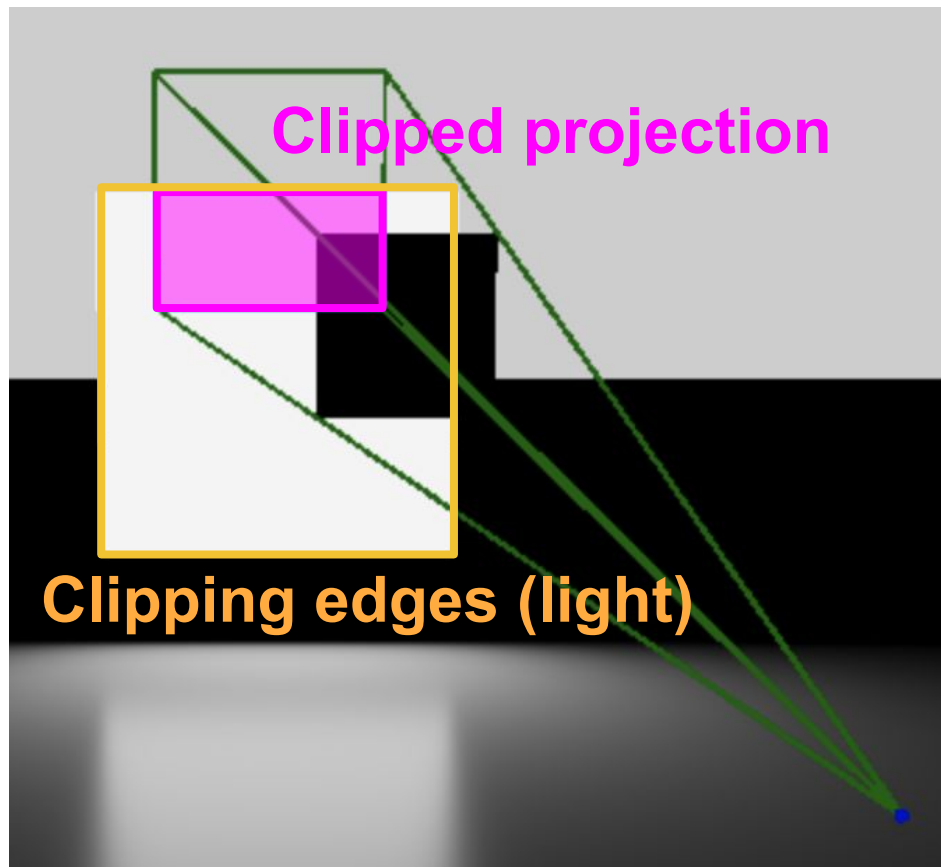
Projection

- **Perspective projection of obstacle to light plane**
 - **Giving 2D light plane coordinates**



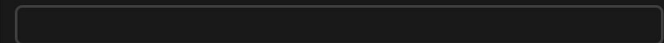
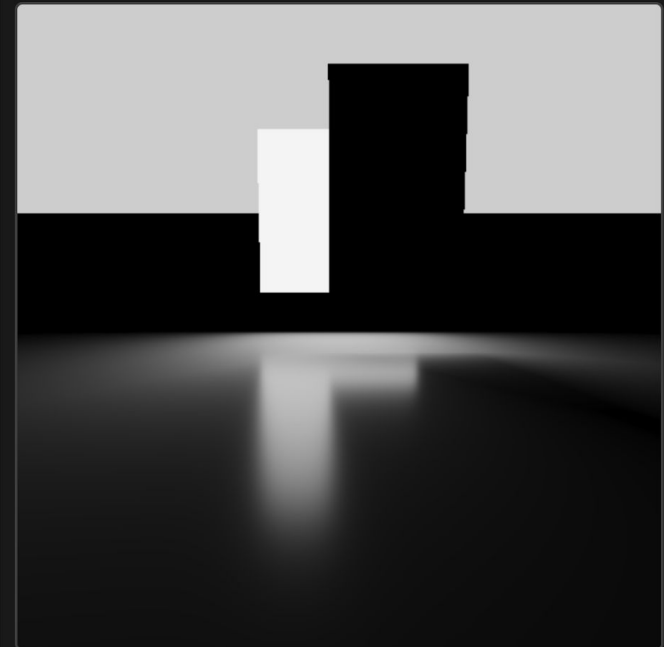
Clipping

- **Clipping by light edges**
 - **Sutherland-Hodgman alg. (2D polygon intersection)**



Results: Real-Time Live Demo

```
0 // bind shadow      {label:"Shadow On", default:true}
1 // bind shadow_debug {label:"Debug mode", default:false}
2 // bind second_obstacle {label: "Second Obstacle", default:false}
3
4 // bind targetu     {label:"Target u", default:0.0, min:-1.0, max:1.0, step:0.01}
5 // bind targetv     {label:"Target v", default:0.0, min:-1.0, max:1.0, step:0.01}
6
7 // bind width_obstacle {label:"Obstacle Width", default: 8, min:0.1, max:15, step:0.1}
8 // bind height_obstacle {label:"Obstacle Height", default: 8, min:0.1, max:15, step:0.1}
9 // bind roty_obstacle {label:"Obstacle Rotation Y", default: 0, min:0, max:1, step:0.001}
10 // bind rotz_obstacle {label:"Obstacle Rotation Z", default: 0, min:0, max:1, step:0.001}
11 // bind posx        {label:"Obstacle Position X", default: 5, min:0, max:10, step:0.1}
12
13 // bind roughness   {label:"Roughness", default:0.25, min:0.01, max:1, step:0.001}
14 // bind dcolor      {label:"Diffuse Color", r:1.0, g:1.0, b:1.0}
15 // bind scolor      {label:"Specular Color", r:0.23, g:0.23, b:0.23}
16 // bind intensity   {label:"Light Intensity", default:4, min:0, max:10}
17 // bind width       {label:"Width", default: 8, min:0.1, max:15, step:0.1}
18 // bind height      {label:"Height", default: 8, min:0.1, max:15, step:0.1}
19 // bind roty        {label:"Rotation Y", default: 0, min:-0.5, max:0.5, step:0.001}
20 // bind rotz        {label:"Rotation Z", default: 0, min:0, max:1, step:0.001}
21 // bind twoSided    {label:"Two-sided", default:false}
22 // bind clipless    {label:"Clipless Approximation", default:false}
23
24
25 uniform float roughness;
26 uniform vec3 dcolor;
27 uniform vec3 scolor;
28
29 uniform float intensity;
30 uniform float width;
31 uniform float height;
32 uniform float roty;
33 uniform float rotz;
34 uniform float targetu;
35 uniform float targetv;
36
37 // obstacle shape
38 uniform float width_obstacle;
39 uniform float height_obstacle;
40 uniform float roty_obstacle;
41 uniform float rotz_obstacle;
42 uniform float posx;
43
44 bool twoSided = false;
45 uniform bool clipless;
46 uniform bool shadow_debug;
47 uniform bool shadow;
48 uniform bool second_obstacle;
49
50 uniform sampler2D tex1;
```



Shadow On

Debug mode

Second Obstacle

Target u

Target v

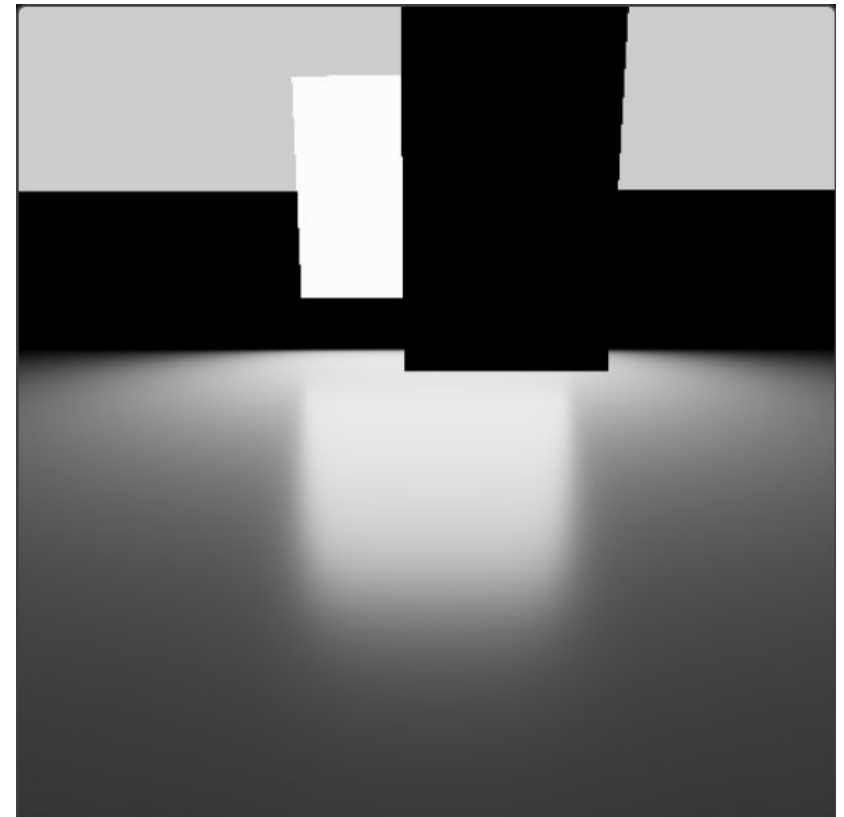
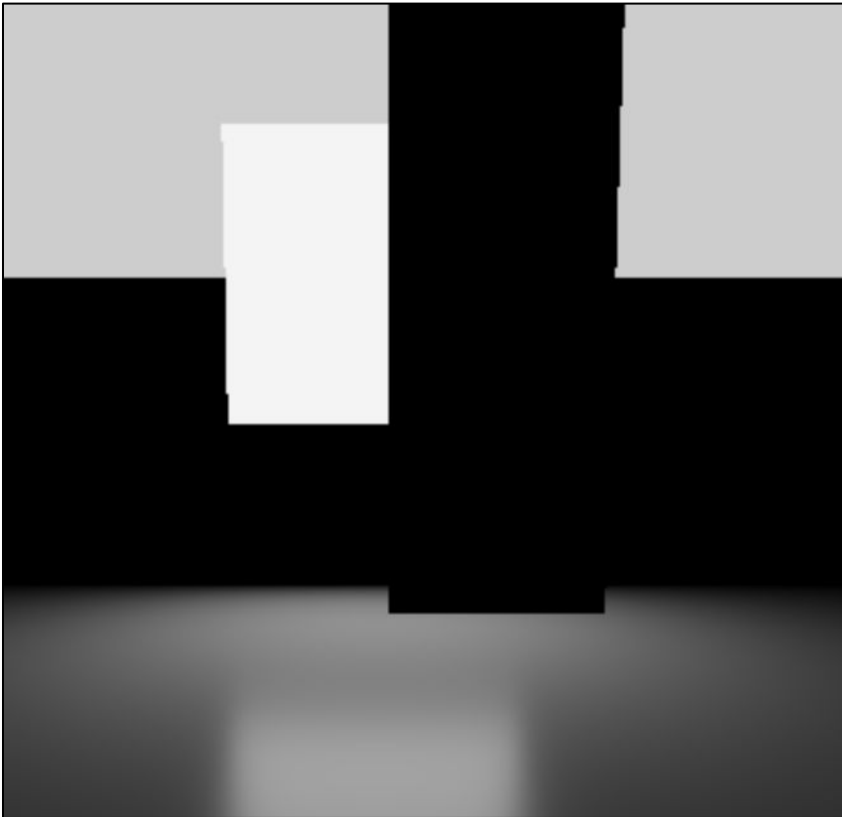
Obstacle Width

Obstacle Height

Obstacle Rotation Y

Practical Benefits

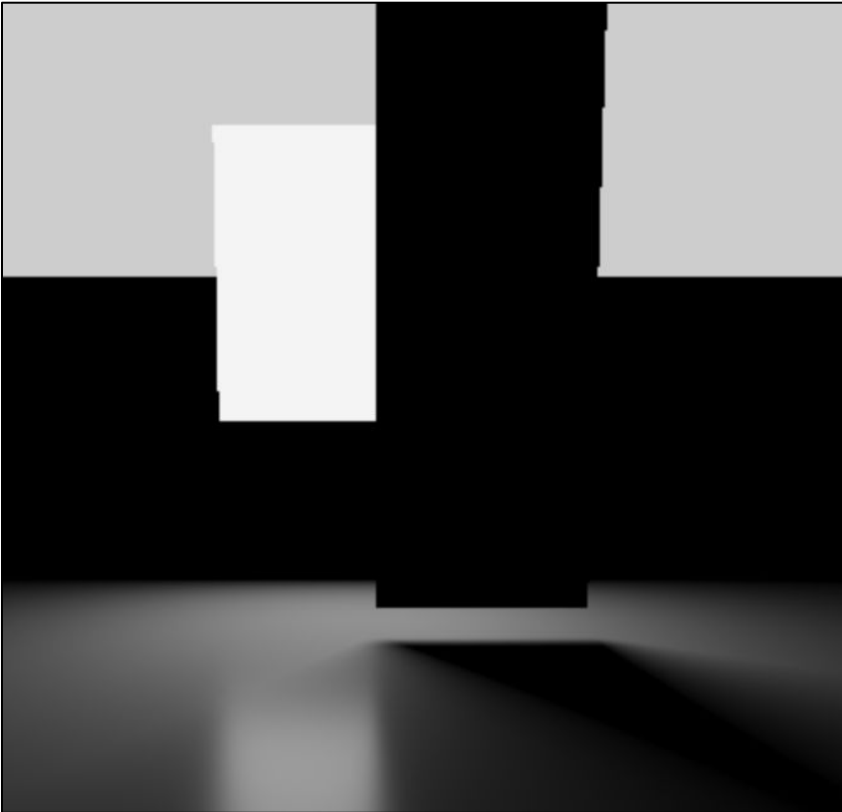
- Drastic enhancement of the **sense of height**



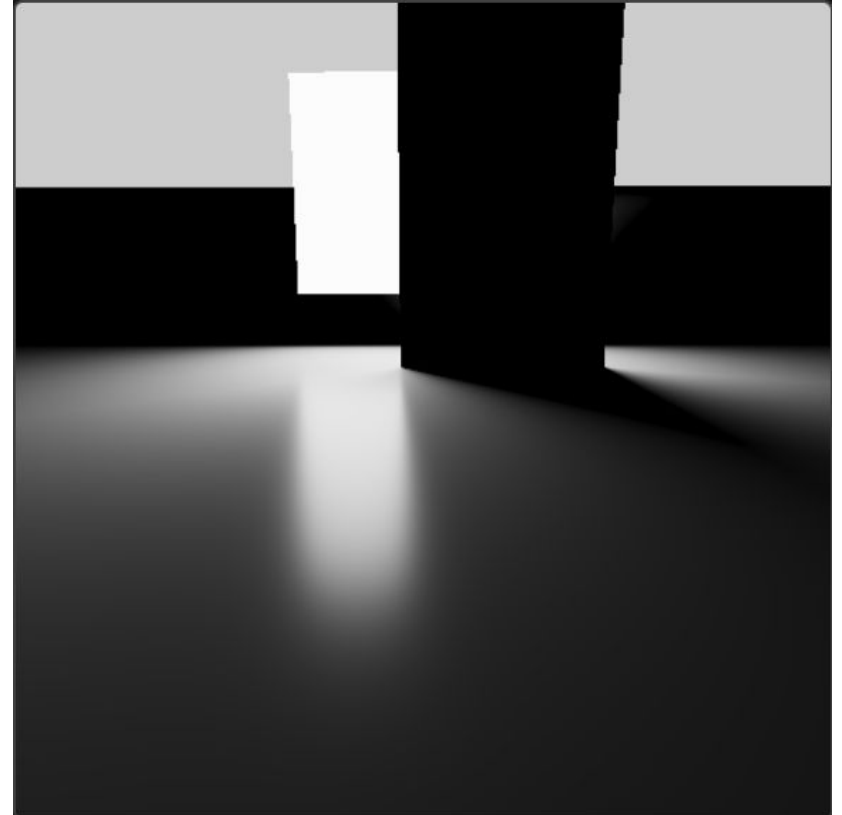
Are they floating in the air?

Practical Benefits

- Drastic enhancement of the **sense of height**



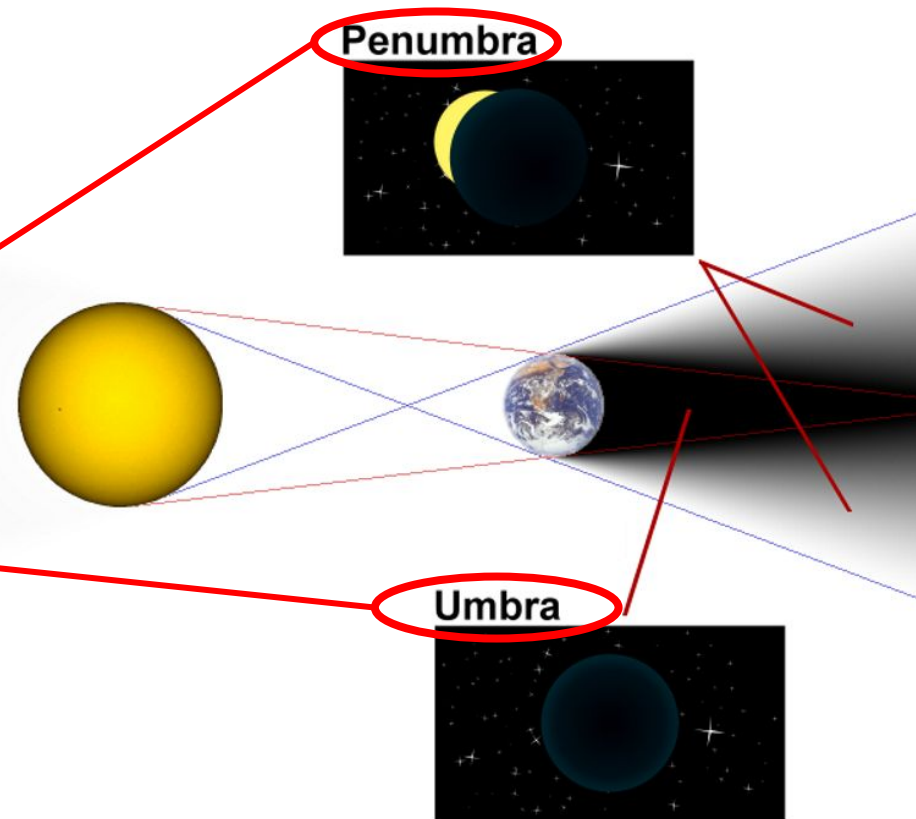
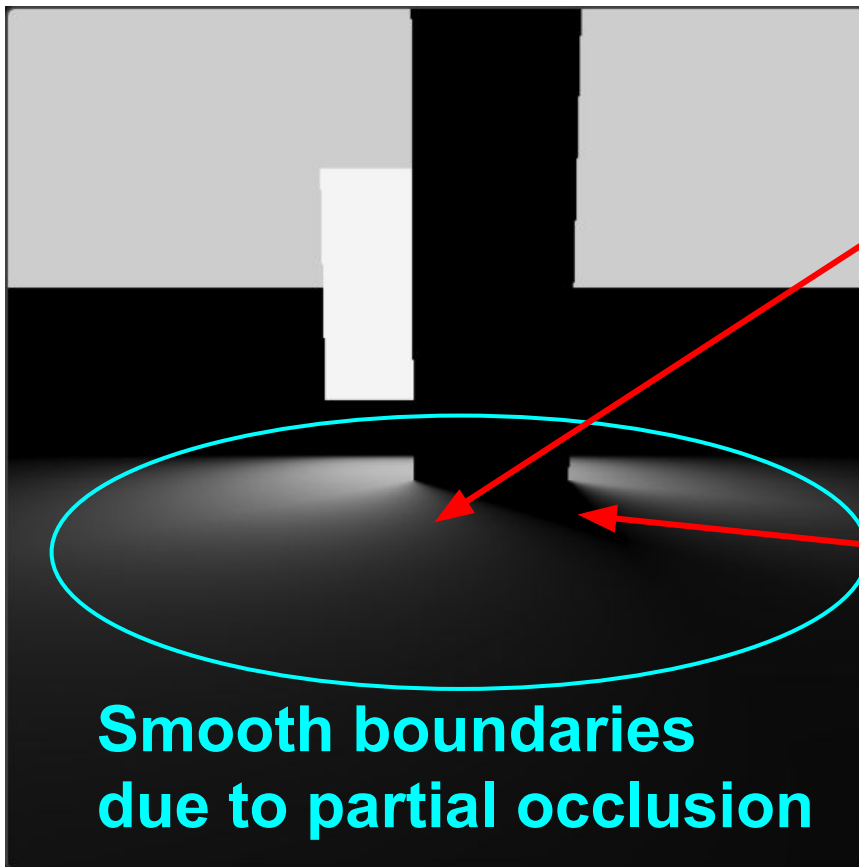
Yes.



No.

Practical Benefits

- **Soft shadow** effect for **area light** source



[Wikipedia] umbra,
penumbra and antumbra

Drawbacks

- As usual, additional graphical effects **increase computational cost**, so does our algorithm
- **Main bottleneck**
 - 1 projection & 1 clipping / **every** camera ray
- **Need advanced optimization**
 - e.g., vertex simplification or parametrization

The Roles of Each Member

- **Eun Hyouk Shin**
 - **Perspective projection**
 - **Clipping**
 - **Visualization tools**
- **In Young Cho**
 - **Draw obstacles**
 - **Generalize the original LTC demo code**
 - **LTC integration for obstacles**
- **Together**
 - **Algorithm design & breakthroughs**

At the End of Our Journey....

- **Shadow rendering**
 - **Visibility term**

- **Separate integration**
 - **integral over the original light**
 - **integral over a projected clipped obstacle**

- **Realistic scenes**