Hello everyone, today we want to share our group work about computer graphics. Since we have a lot to share, we assume you have enough knowledge about real time render. So, what do we want to do? You know there are many amazing video games in the world, the image quality and fluency have a major impact on players. Today we will introduce some technology, including lighting and shadow.

First thing is Phong lighting model, the key idea is supposed the surface of an object is perfect surface and use some basic lighting equation. But around 2010, physically based rendering became popular. This idea emphasizes on using real-world physics to simulate and render. In lighting, people created a model called microfacet surface model, to simulate the surface roughness of the object. Also, it was thought that real radiation was needed to describe the energy of light and this is called energy conserving. The metallic described the proportion of metal in the material, as you know, metal materials have no diffuse reflection. You can see, there are some renderings and have a great effect. A very important reason people use PBR is TA(technical artist) won’t need to tuning parameters! They can directly use real world parameters and quickly get amazing pictures!

Here are some formula derivations but we won’t talk the details. In fact, we only need to calculate this integration. In 2003, an epoch-making paper was published. They were using Spherical Harmonics to estimate this integration. In recent years, thanks to the rapid development of GPU, Unreal Engine 4 gives another approximation. This method is very fast, and the photos shown before is the result of our reproduction.

Second thing we want to share is shadow. Traditional shadow technique was shadow map. It has a lot of bugs, like shadow acne, peter panning and need over sampling to get better quality. Also, it doesn’t create soft shadow! Recall your life experiences, all the shadows you’ve seen are soft shadows! People improved it by using Percentage-Closer Soft Shadow (Based on PCF) and Variance Soft Shadow Mapping (Faster PCSS).

Nowadays, a new technique has become popular. It’s called Signed Distance Field Soft Shadow. We start from each shader point, emit a ray to light point, and go along this step by step. The SDF can tell us, there cannot be any occlusion inside these circles. So, an intuitive way is the smallest theta describes the visibility of this shader point.

But, when we try to reproduction this method, we faced a lot of difficulties. A lot of details didn't share to public, and we don't have the ability to read source code of UE4. We had tried our best to do it. One is we need to draw a sphere on light point too, only on the path is not enough, otherwise you can see this. Another thing is the generation of Signed Distance Field is hard. Just a 64\*64\*64 pixels SDF is already very huge. But if we use some approximation, you can see this.

Also, we have a question that why UE won't use vectors to describe distance? In this way we can interpolate better approximation, also, this whole method doesn’t require a lot of floating point precision, easily to achieve same memory storage. We also apply this, and you can see a little bit of improvement (\*).

So, in our group work, we explored the technique about real time rendering, and studied how to render more realistic lighting and shadows. Also, we raised some questions and made a little progress. The code is all wrote by OpenGL and we wrote it by ourselves. This project is public on Github if you are interested.