高质量渲染

—Stochastic Screen-Space Reflections
随机屏幕反射

巧妙的环境光照

--基于图像光照（IBL）

细节补完SSSR

I'M
Luminance -H

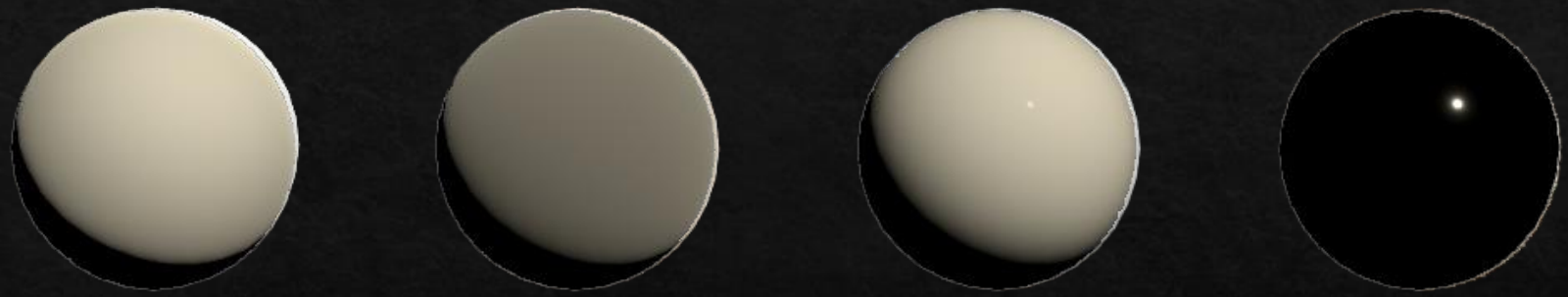
Agenda

- Previous Work 前人工作
- Motivation 动机
- Technology Structure and Details 技术结构和细节
- NPR SSSR 基于卡渲的拓展

Previous Work

- BRDF And IBL

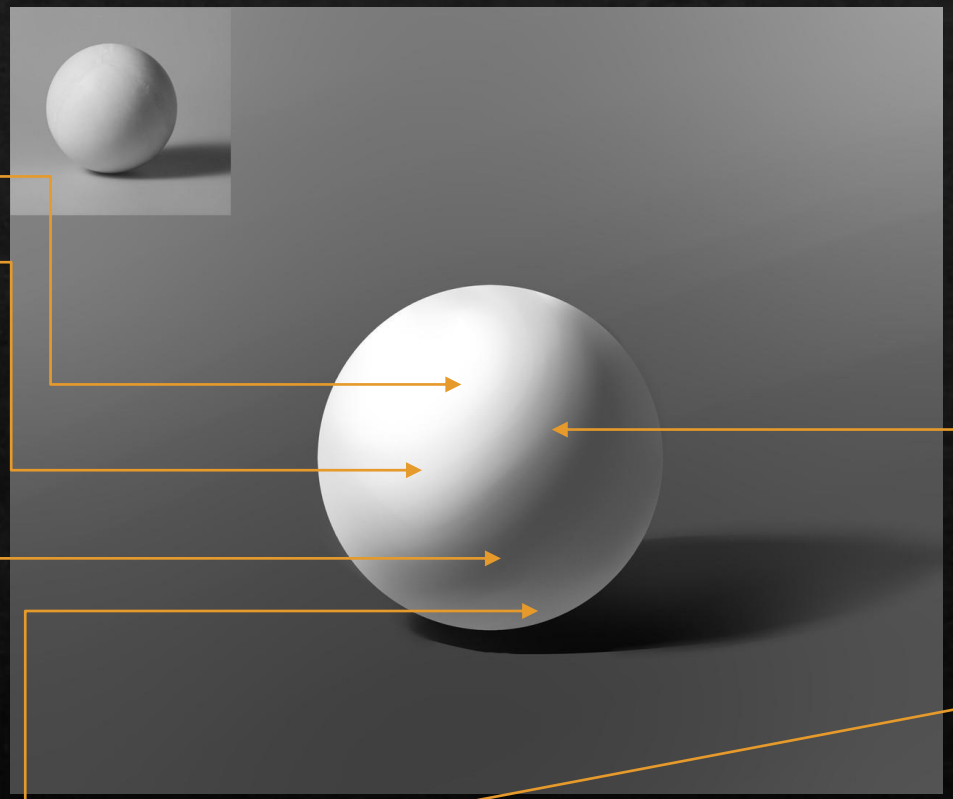
BRDF in your mind but Actually....



what happened ?

Subsurface Scattering Introducition

- Artistic perspective



In the case that just single light source in scence,BRDF is similar as Blinn–Phong

Previous Work

- BRDF And IBL

Luminance(BRDF) = Light Source + Environmental light

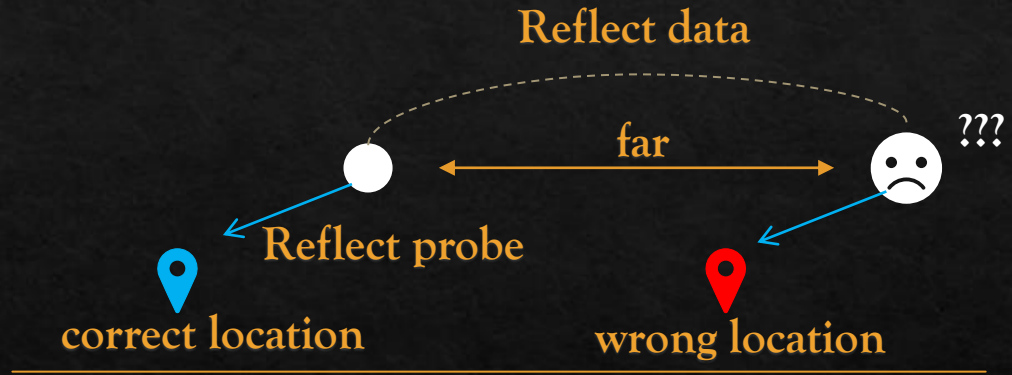
For Eample .....

Motivation

- IBL and SSSR

BRDF+IBL still have some flaws...

Environmental light



If a lot of object between you and the probe, the flaws will be conspicuous.
Just like the film celled 《FREE GUY》

Motivation

- IBL and SSSR

So, the SSSR is coming

Stochastic Screen-Space Reflections is base SSR and render base Physical
随机屏幕反射是基于SSR的PBR渲染形式

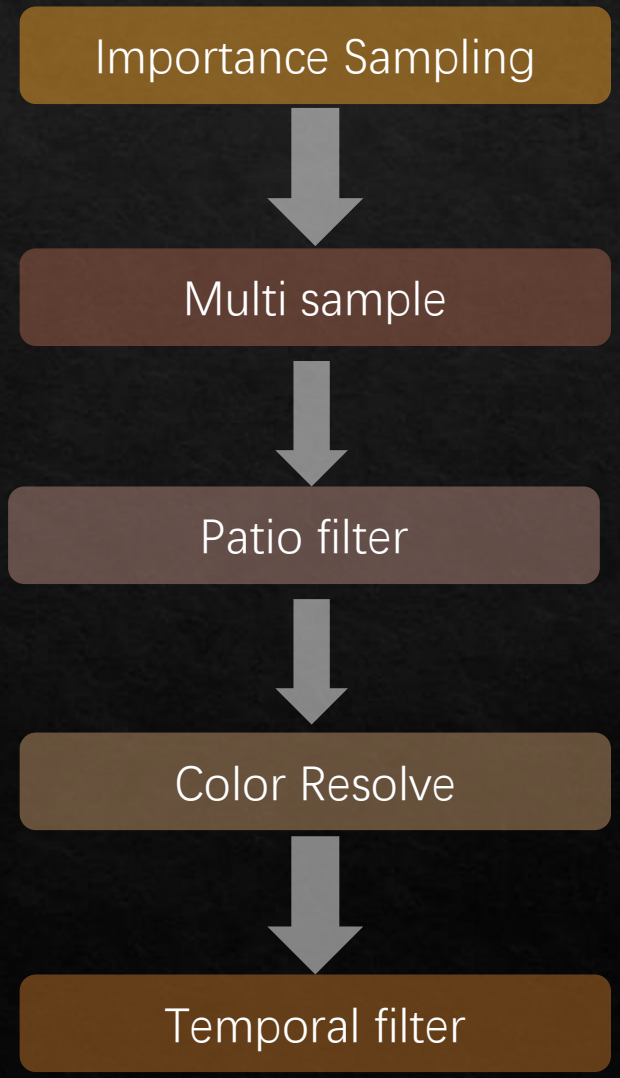
We use cube map in IBL，and use screen texture in SSSR
我们在IBL中使用的是Cube map，而在SSSR中将使用屏幕图像

SSR + IBL +BRDF= SSSR

It works the same as ray tracing

Technology Structure and Details

- Structure



Our Work

Technology Structure and Details

- Ray Marching SSR

And if you don't know what is SSR, you should click here first. You also can find them in the summary
如果还不知道什么是屏幕反射技术，我认为你应该先点击以下链接，SSSR是基于SSR的PBR表现形式，是IBL技术的补完，而在本章内我们并不会过多的去讲IBL和SSR，因为他们的内容太多了，我们更多的是讲述SSSR技术，也就是随机屏幕空间反射技术。

【1】Screen Space Reflection | 3D Game Shaders For Beginners (lettier.github.io)
【2】(14条消息) 图形学基础|PBR回顾_桑来93的博客-CSDN博客

如何有不懂的地方也可以私信问我，间接中也有对应的文章，稍后我们整理的SSSR HLSL源码也会放在Github中

Technology Structure and Details

Importance Sampling

We use ray tracing to get reflect Light, but it is too Expensive

So We use Importance Sampling to get an approximate solution

Actrually, Importance Sampling is a cheap Ray tracing but more complex.

It will get solution by less data

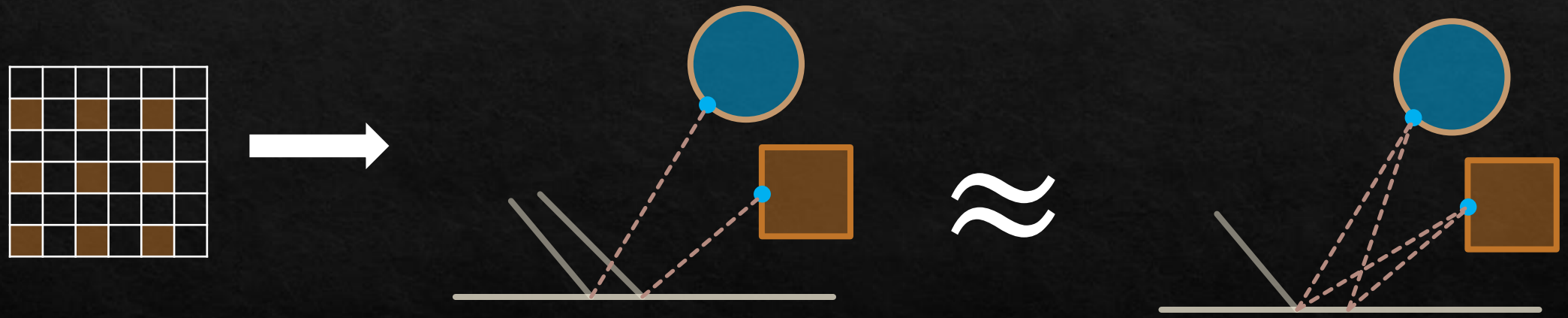
$$\int f(x)\mathrm{d}x = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{f(x_k)}{p_k}$$

GXX Reflect Models

Technology Structure and Details

- Multi sample and Patio filter

And we will use the data of near pixel to assume we still have more ray from the pixel.
我们会使用临近的像素数据假设是该点射出的
这样采样的Ray就从空间上增加了维度，从N条变成了N*M条



Still use:

$$\int f(x)\mathrm{d}x = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\frac{f(x_k)}{p_k}$$

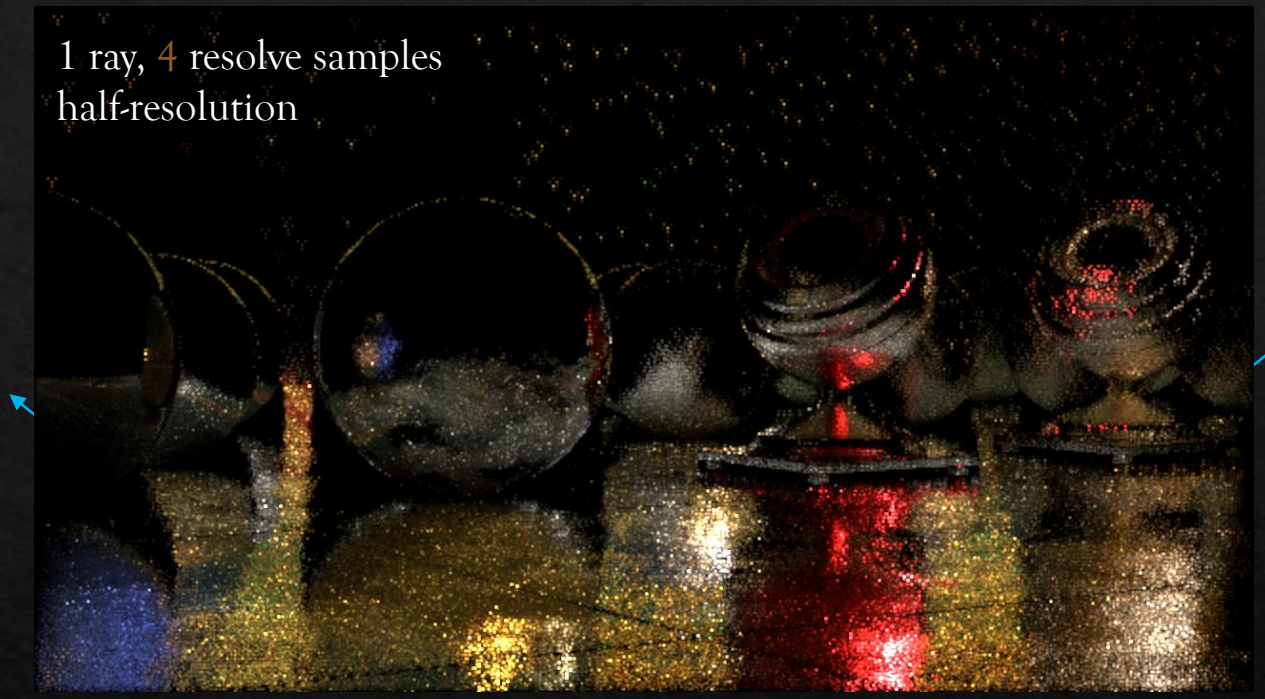
Technology Structure and Details

- Patio filter

But we will get a wrong results

$$\int f(x)\mathrm{d}x = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{f(x_k)}{p_k}$$

In fact, the different ray of different pixel just give us a different Pk. So if we will got a wrong results by wrong PDF



And you can see it

Because there are so many differences between them

Technology Structure and Details

- Structure Importance Sampling

Solution of Frostbite

$$L_o = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{L_i(l_k) f_s(l_k \to v) \cos\theta_{l_k}}{p_k}$$

Variance!

Frostbite try to cripple the effect of pk:

$$L_o = \frac{\int L_i(l) f_s(l \to v) \cos\theta_l \mathrm{d}l}{\int f_s(l \to v) \cos\theta_l \mathrm{d}l} \int f_s(l \to v) \cos\theta_l \mathrm{d}l$$

And if we think more, such as use IBL:

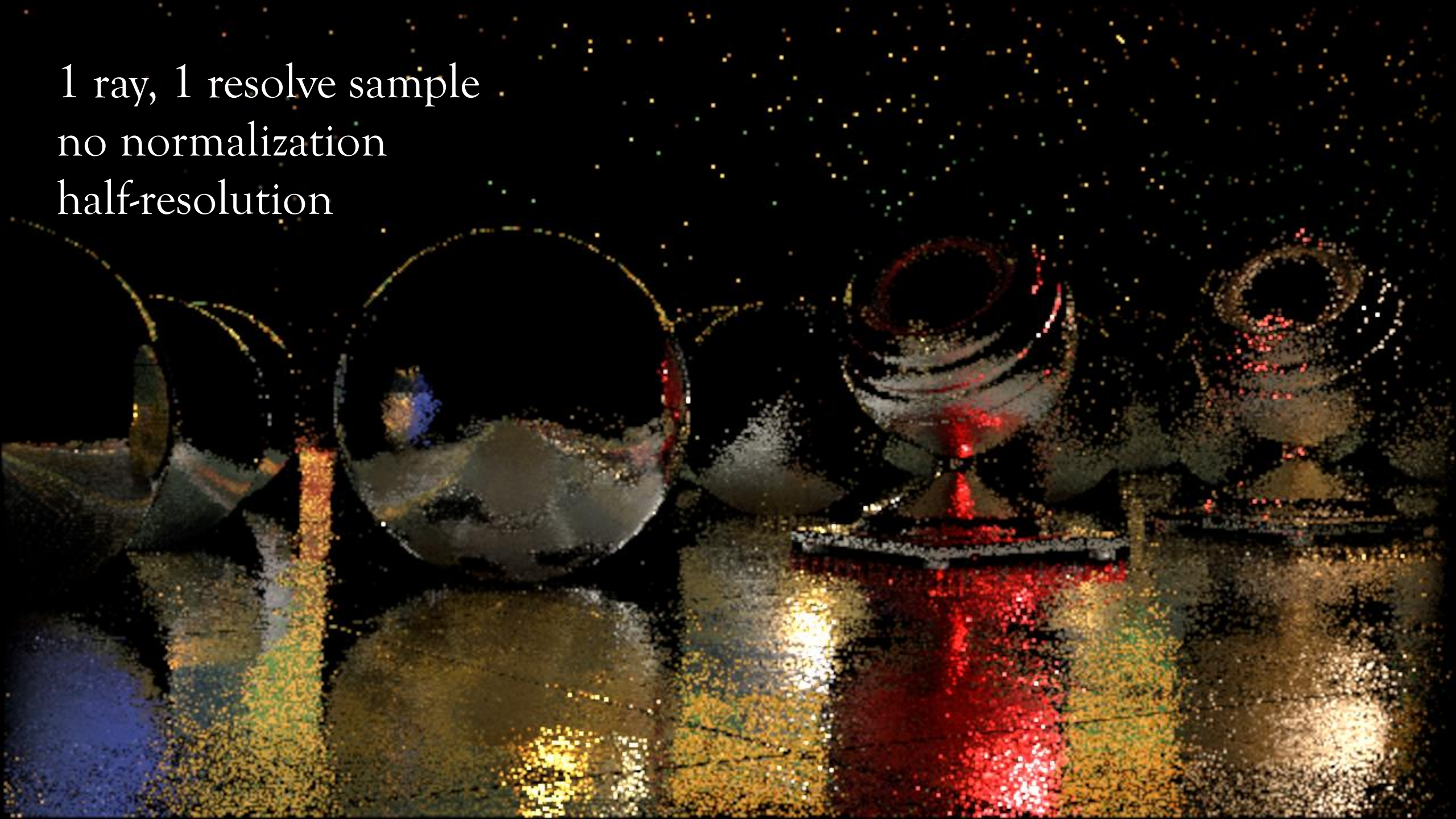
$$L_o \approx \frac{\int L_i(l) f_s(l \to v) \cos\theta_l \mathrm{d}l}{\int f_s(l \to v) \cos\theta_l \mathrm{d}l} FG$$

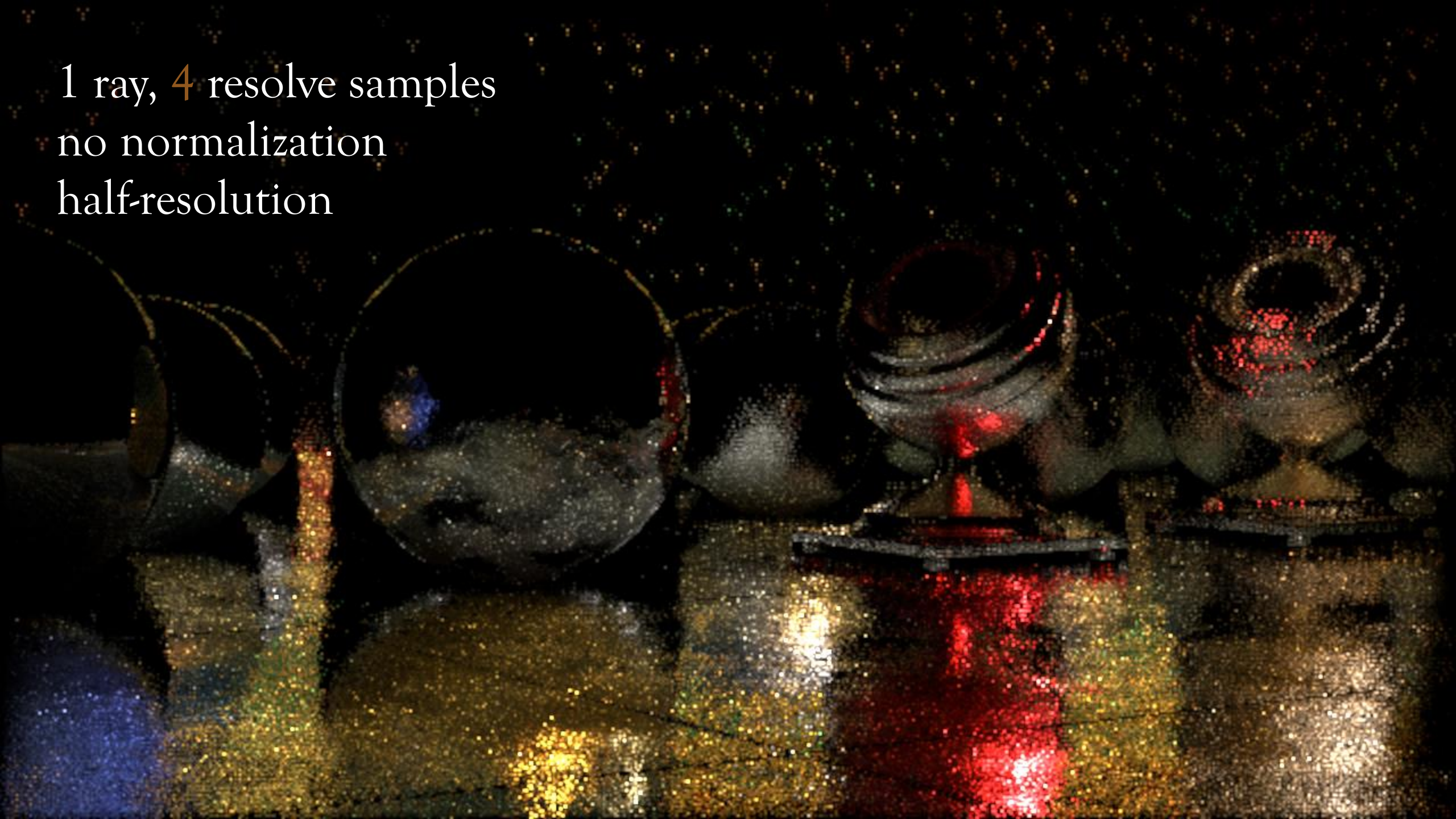
Technology Structure and Details

- Structure Importance Sampling

… and pseudocode

```
result    = 0.0
weightSum = 0.0
for pixel in neighborhood:
    weight = localBrdf(pixel.hit) / pixel.hitPdf
    result += color(pixel.hit) * weight
    weightSum += weight
result /= weightSum
```


1 ray, 1 resolve sample
no normalization
half-resolution


1 ray, 4 resolve samples
no normalization
half-resolution

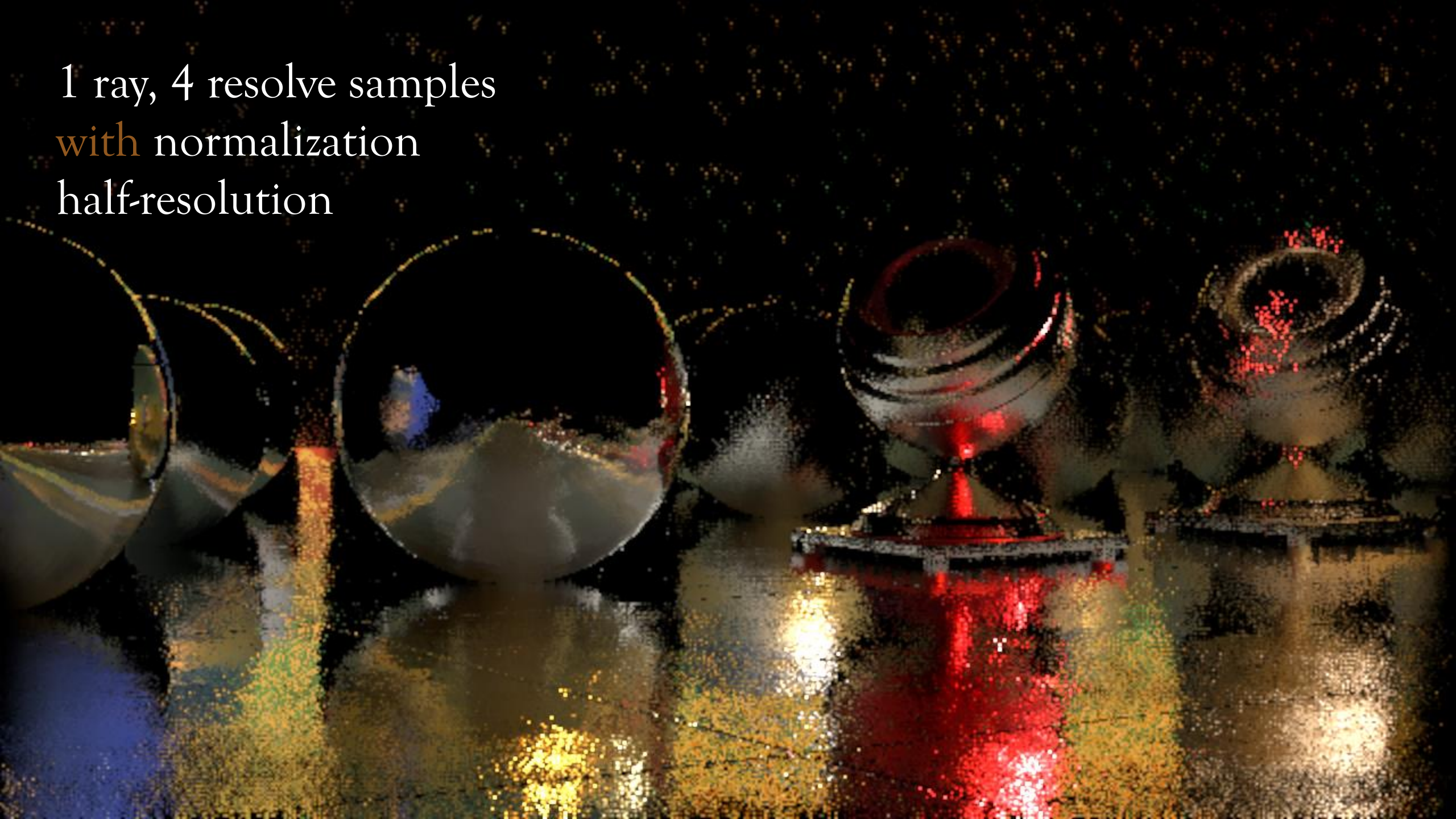
Technology Structure and Details

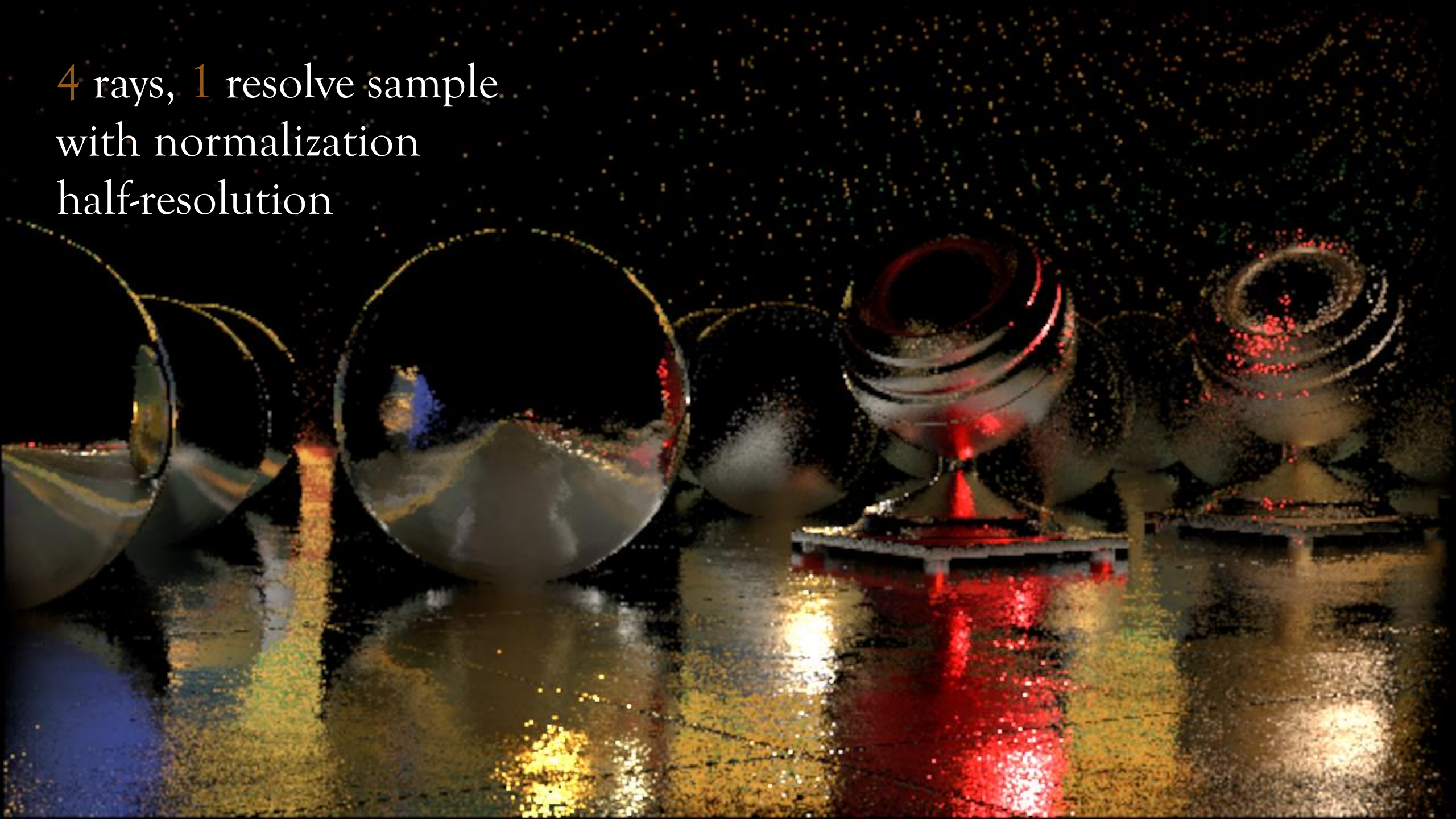
- Structure Importance Sampling

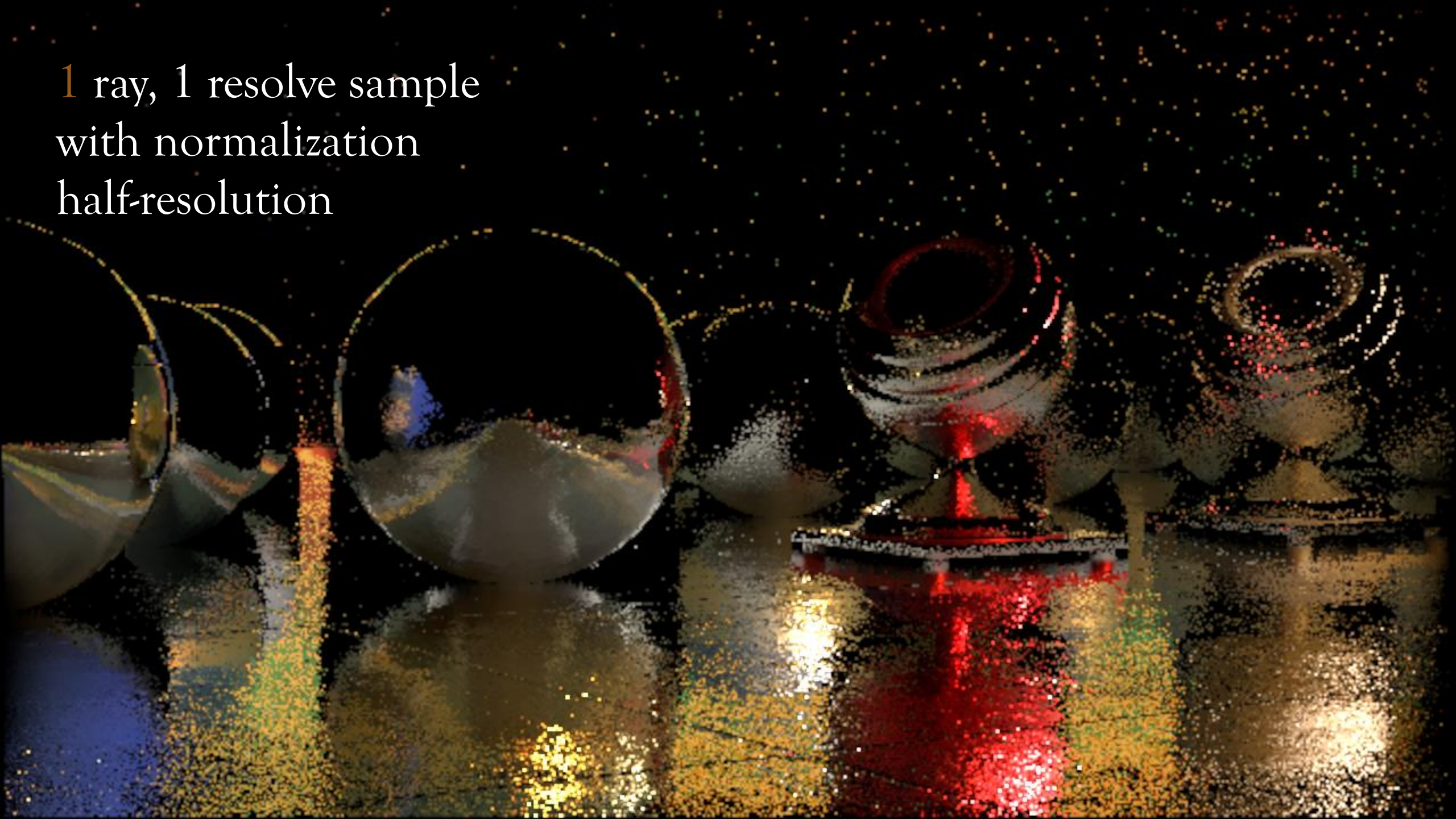
$$L_o \approx \frac{\int L_i(l) f_s(l \to v) \cos\theta_l \mathrm{d}l}{\int f_s(l \to v) \cos\theta_l \mathrm{d}l} \quad FG$$

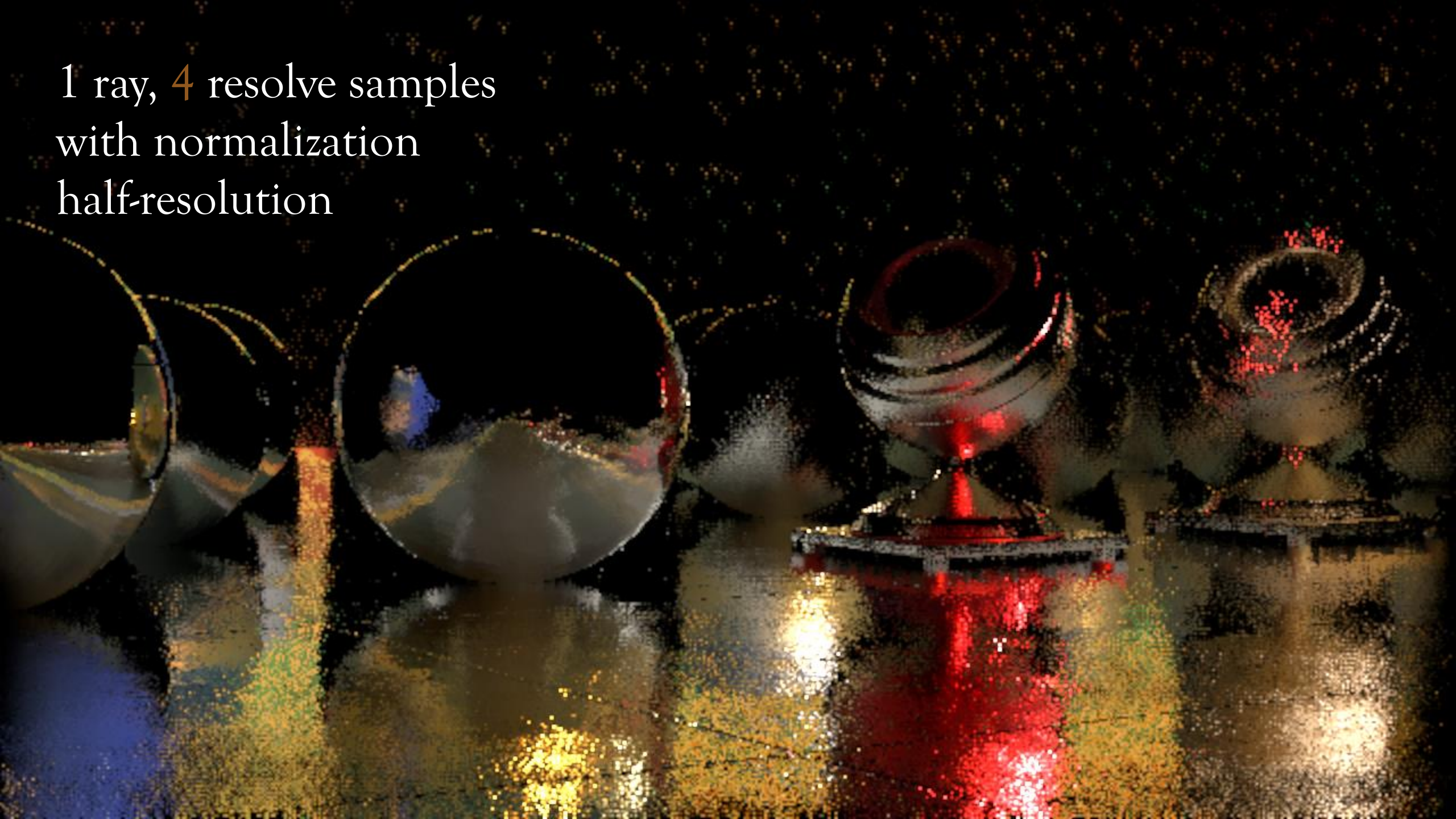
normalization

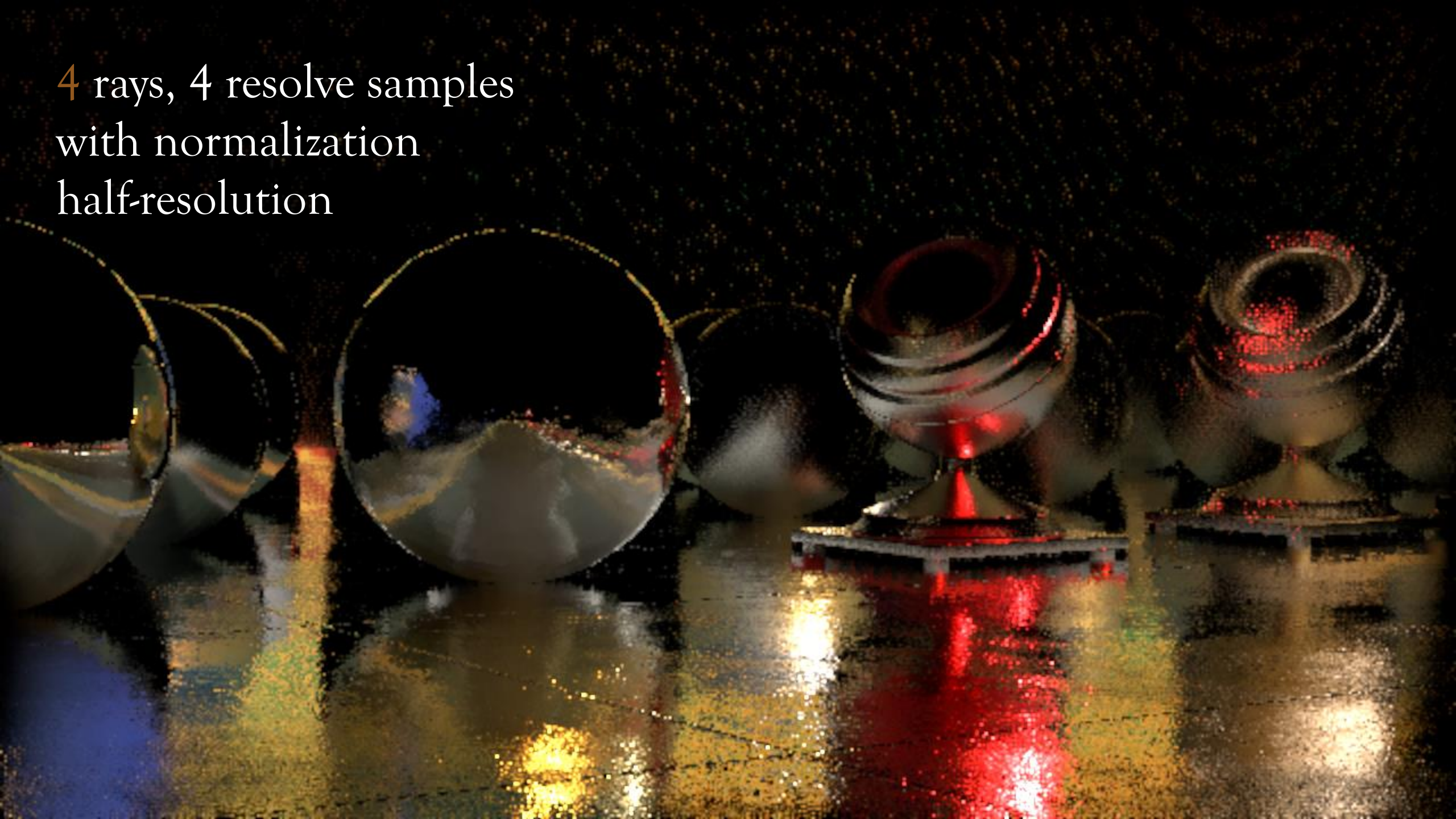
FG 归一化可以得到噪点更少的结果


1 ray, 4 resolve samples
with normalization
half-resolution


4 rays, 1 resolve sample
with normalization
half-resolution

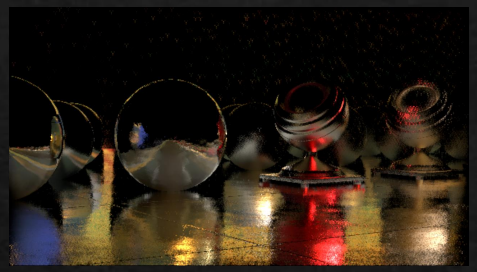

1 ray, 1 resolve sample
with normalization
half-resolution


1 ray, 4 resolve samples
with normalization
half-resolution

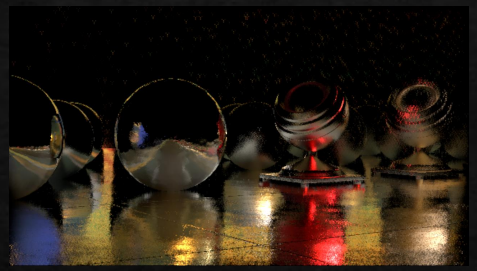

4 rays, 4 resolve samples
with normalization
half-resolution

Technology Structure and Details

- temporal filter

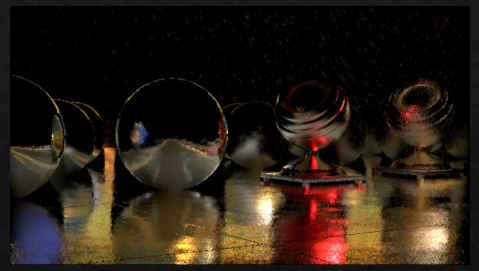


Frame 1



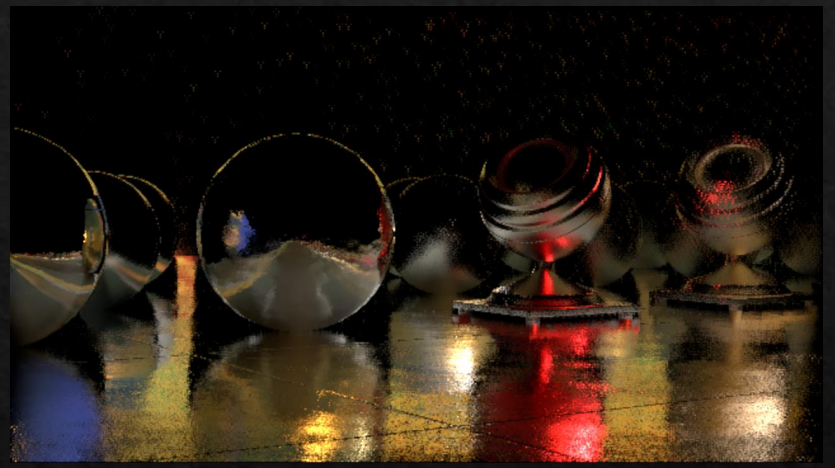
Frame 2

....



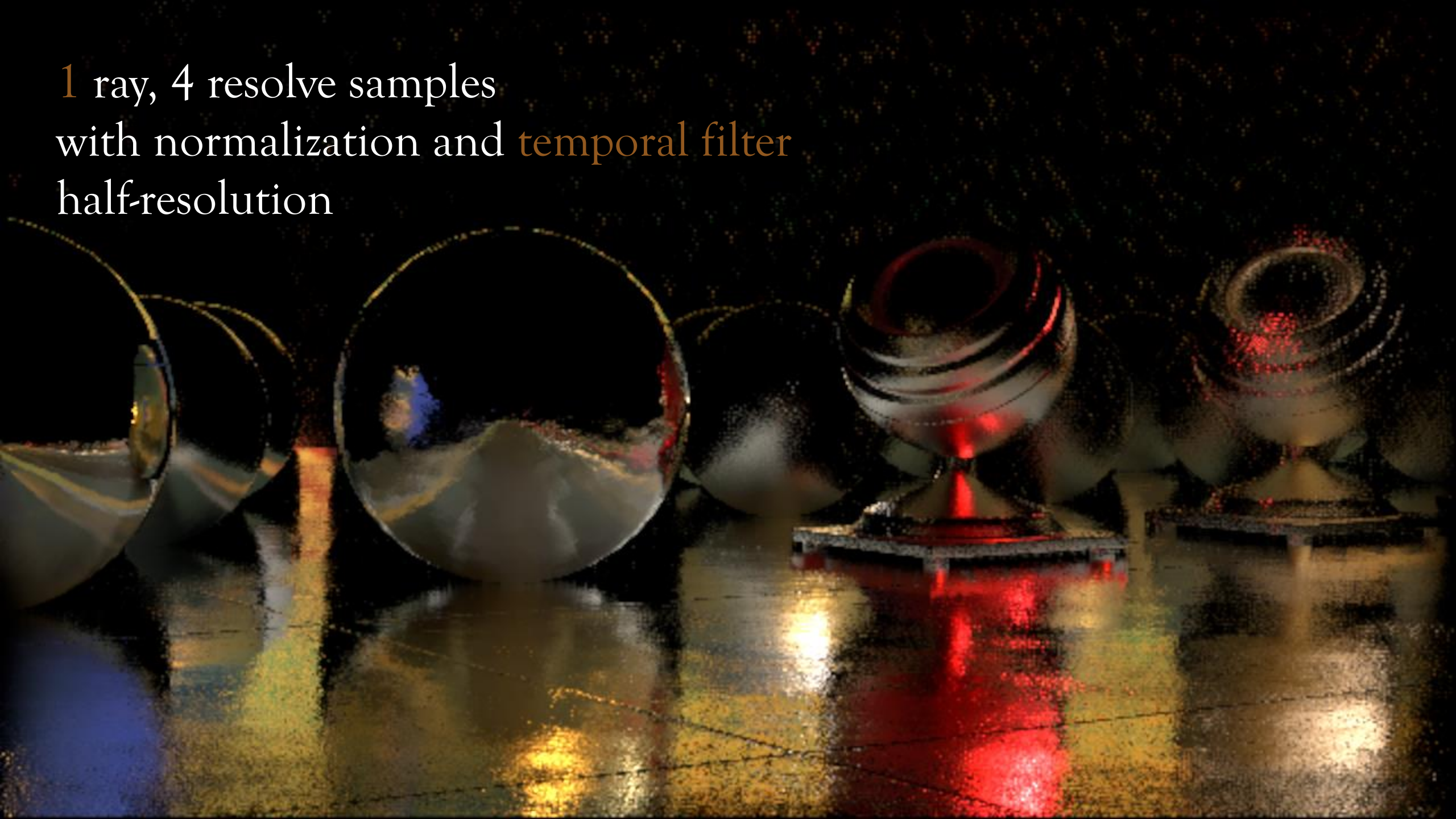
Frame now(Filtered)

Mix



Out Color

The operating principle of temporal filter is very Simple(but not when implementing)
It just us the data of old fames to mix now frame
时间滤波的原理很简单，就是用这以前几帧的结果来扩大射出的Ray数据，从N*M扩展到了N*M*∞，在理想情况下，摄像机不动，则可得到的光线数量可以达到无穷（理想情况）


1 ray, 4 resolve samples
with normalization and temporal filter
half-resolution

Technology Structure and Details

- Structure Importance Sampling

  - Lots of noise from BRDF tail
  - Shift samples toward mirror direction
    - Still need accurate PDF values
  - Truncated distribution sampling

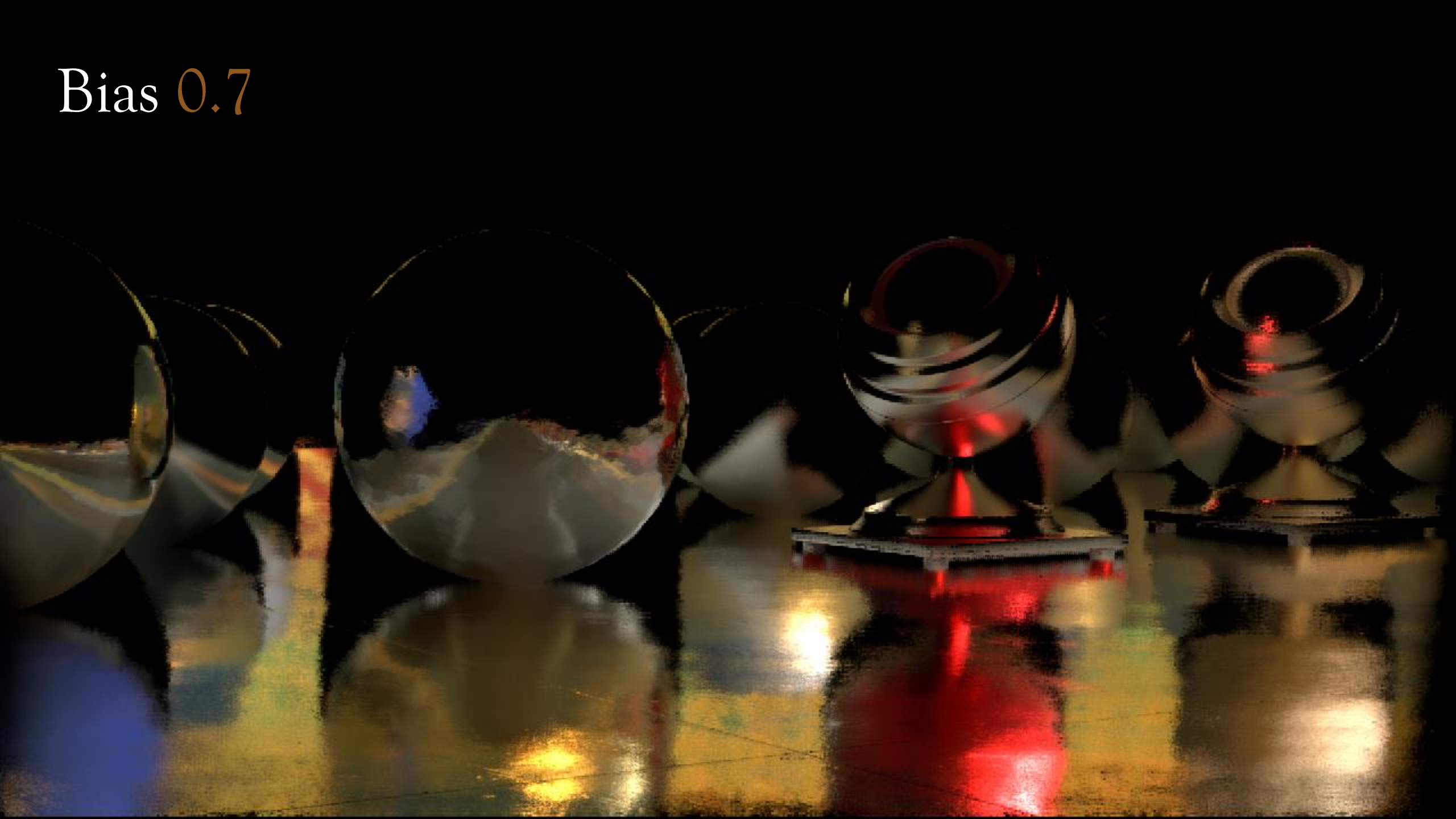
```
float2 u = halton(sampleIdx);
u.x = lerp(u.x, 1.0, bias);
importanceSample(u);
```

  - Different normalization constant
    - Our variance reduction re-normalizes!

Remap to(0,1)

Bias 0.0

Bias 0.7




光照需要重建（28 未构建对象）
Roughnees 0


光照需要重建（28未构建对象）
Roughnees 1

Technology Structure and Details

- NPR SSSR

NPR can also use SSSR-NPR

In my opinion:

NPR $= L * (Diffuse + Specular)$

$Specular \infty$ Normal It means we can get the corresponding probability distribution formula

And use Importance Sampling :

$$\int f(x)\mathrm{d}x = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{f(x_k)}{p_k}$$

mhy has not use that, image is MMD

I mean if your Shading model It has something to do with Math so you can try to use that
我的意思是，如果你的NPR模型是可以用数学公式表示的，那么你可以尝试去加入GI，SSSR，IBL去提升渲染质量，他们的效果将会和光追一样

I'M
Luminance -H

Thanks For Watching