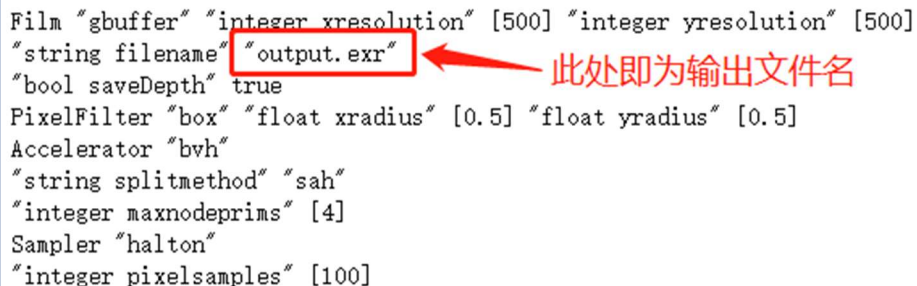


小米场景部分打包说明

1. 打包内容: pbrt 渲染程序 + Exr2ISET 的代码
2. 输入文件: pbrt 文件
3. 输出文件: 场景 mat 文件 (一个 struct), 其文件名与 exr 文件名一致, 这个文件名是在 pbrt 文件中进行设置的, 因此可从 **pbrt 文件** 中读取到。见下图:

```
Film "gbuffer" "integer xresolution" [500] "integer yresolution" [500]
"string filename" "output.exr"
"bool saveDepth" true
PixelFilter "box" "float xradius" [0.5] "float yradius" [0.5]
Accelerator "bvh"
"string splitmethod" "sah"
"integer maxnodeprims" [4]
Sampler "halton"
"integer pixelsamples" [100]
```



4. 参考程序

PBRT 渲染:

<https://github.com/scienstanford/pbrt-v4/tree/omni>

EXR2ISET:

<https://github.com/ISET/iset3d/blob/main/utilities/pbrt/pbrt-v4/piEXR2ISET.m>

简化后的 EXR2ISET:

见示例文件夹

5. 具体说明

(1) 首先第一步是打包 pbrt 渲染程序, 使其可以根据输入的 pbrt 文件输出得到 .exr 文件。

(2) 然后读取 exr 以及 pbrt 文件的信息, 并把他们组织成 matlab 结构体的形式, 具体字段和值的说明如下:

字段	值
type	‘scene’
metadata	[]
spectrum	spectrum.wave=400:10:700, 见示例 mat 文件
distance	1
magnification	1
data	data.photons 是 exr 文件中 31 个通道的 radiance 数据, 具体说明见 (3)
	data.luminance = []
illuminant	已废弃, 按照示例文件给相同的值就可以
WAngular	从 pbrt 中读取 fov 值, 然后计算, 说明见 (4)
depthMap	单通道深度图, 计算方法说明见 (5)
name	按照生成 exr 文件的名称给就可以, 去掉后缀名

(3) exr 文件中通过 31 个通道存储了 radiance 数据, 其存储的数据单位是 watts/sr/m²/nm, 我们需要转成 photons/sr/m²/nm, 具体计算方法参考 iset-3d 中的 Energy2Quanta 函数。

(4) Pbrt 文件里面的 fov 表示的是垂直方向的 HAngular (角度制), 换算成水平方向的 WAngular (角度制) 的方法如下 (计算时请注意角度制和弧度制是否需要转换):

$$\text{half_HAngular} = \text{HAngular} / 2$$

$$\text{half_WAngular} = \arctan(\text{half_HAngular} / \text{HH} * \text{WW})$$

$$\text{WAngular} = \text{half_Wangular} * 2$$

其中 HH 和 WW 分别是 radiance 数据的垂直分辨率和水平分辨率

(5) 在 exr 中, depthmap 的通道为 ‘Px’, ‘Py’ 和 ‘Pz’

$$\text{depthMap} = \sqrt{\text{Px}.^2 + \text{Py}.^2 + \text{Pz}.^2}$$

注: 若 pbrt 文件中设置了不渲染 depthmap, 那么 exr 文件中将没有 depthmap 的三个通道, 此时 depthmap 给全 1 的单通道矩阵即可, 分辨率与 radiance 数据的空间分辨率相同。