

🔗 6. gltf后处理颜色异常(伽马校正)

你打开上节课代码，可以发现，加载gltf模型，如果使用EffectComposer添加后处理功能，模型颜色可能会出现异常，对于这种情况，首先要先分析产生颜色偏差原因。

颜色偏差的原因

首先你可以回顾6.3和6.9节内容，加载gltf模型如果出现颜色偏差，需要设置 `renderer.outputEncoding` 解决。

如果你使用threejs后处理功能EffectComposer， `renderer.outputEncoding` 会无效，自然会出现颜色偏差。

```
renderer.outputEncoding = THREE.sRGBEncoding;
```

js

引入 `GammaCorrectionShader.js`

GammaCorrectionShader.js扩展库的目录： `examples/jsm/shaders/`

`examples/jsm/shaders/` 目录下有很多不同功能的shader文件,GammaCorrectionShader.js的功能就是进行伽马校正，具体点说就是可以用来解决gltf模型后处理时候，颜色偏差的问题。

```
// 伽马校正后处理Shader
import {GammaCorrectionShader} from 'three/addons/shaders/GammaCorrectionShader.js';
```

js

引入 `ShaderPass.js`

ShaderPass.js扩展库目录： `examples/jsm/postprocessing/`

```
// ShaderPass功能：使用后处理Shader创建后处理通道
import {ShaderPass} from 'three/addons/postprocessing/ShaderPass.js';
```

js

创建伽马校正后处理通道

threejs并没有直接提供伽马校正的后处理通道，提供了一个伽马校正的Shader对象GammaCorrectionShader，这时候可以把Shader对象作为ShaderPass的参数创建一个通道。

```
// 创建伽马校正通道
const gammaPass= new ShaderPass(GammaCorrectionShader);
composer.addPass(gammaPass);
```

js

← [5. gltf工厂模型设置发光描边](#)

[7. 抗锯齿后处理](#) →