

🟡 3. 阴影范围.shadow.camera

上节课2.平行光阴影计算讲解过，通过光源的阴影相机属性 `.shadow.camera` ,来控制阴影的渲染范围，本节课继续 `.shadow.camera` 的讲解。

平行光阴影相机属性 `.shadow.camera` 回顾

- 平行光 `DirectionalLight` 的 `.shadow` 属性是平行光阴影对象 `DirectionalLightShadow`
- 平行光阴影对象 `DirectionalLightShadow` 有一个相机属性 `.camera`
- `.shadow.camera` 属性值(正投影相机 `OrthographicCamera`)

CameraHelper 可视化 `.shadow.camera`

为了方便观察阴影渲染的范围，需要把平行光对应的阴影相机 `.shadow.camera` 可视化显示。

```
new THREE.CameraHelper(directionalLight.shadow.camera);
```

js

参数测试

下面先在上节课代码基础上给大家改参数体验下相机渲染范围参数对阴影的影响。

课件演示文件中，提供了批量创建了多个长方体，用于阴影范围设置测试。

设置相机 `.shadow.camera` 长方体范围

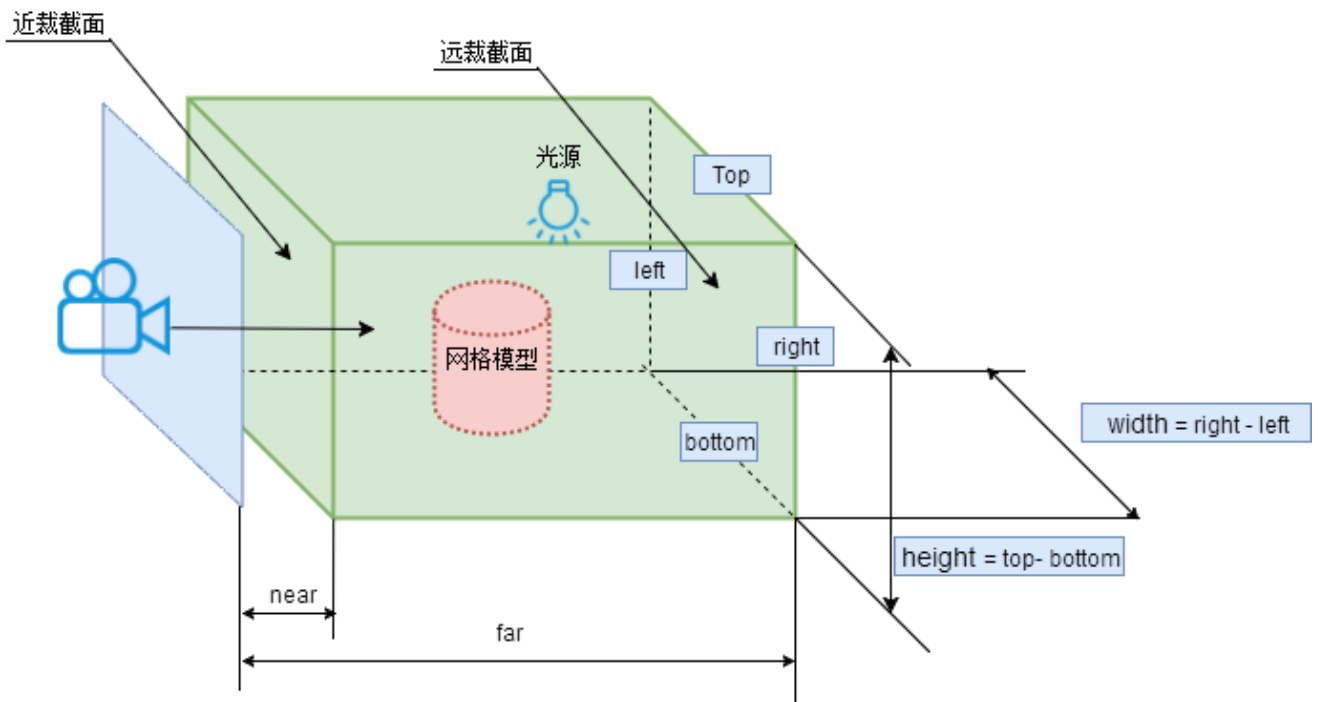
根据3D场景渲染范围，去设置 `.shadow.camera` 长方体尺寸参数，一般比你渲染的范围稍微大一些即可，过小阴影不显示或显示不完整，过大很多可能阴影偏模糊，你可以比较下面两个参数的阴影渲染差异。

```
directionalLight.shadow.camera.left = -50*10;  
directionalLight.shadow.camera.right = 50*10;
```

js

js

```
directionalLight.shadow.camera.left = -50*100;
directionalLight.shadow.camera.right = 50*100;
```



调节光源位置

光源位置影响平行光阴影相机 `.shadow.camera` 的位置，所以要根据渲染范围调整光源的位置。

你可以比较测试两个不同的光源位置，对应阴影渲染效果。

js

```
directionalLight.position.set(100, 60, 50);
```

js

```
directionalLight.position.set(100*2, 60*2, 50*2);
```

确定阴影计算范围

其实平行光阴影范围的设置，你可以类比以前正投影机位置、长方体可视化空间的设置。

- 1. 先大概确定下3D场景中需要阴影计算范围，不用精确，有一个数量级就行，比如几百、几千。
- 2. `.shadow.camera` 的 `.left`、`.right`、`.top`、`.bottom`、`.near`、`.far` 属性定义的长方体空间

- 3. `.shadow.camera` 的位置(光源位置影响 `.shadow.camera` 的位置)

需要阴影范围数量级：z方向尺寸1000左右，xy方向100左右。

```
for (let i = -3; i < 4; i++) {  
  const mesh2 = mesh.clone();  
  // 设置产生投影的网格模型  
  mesh2.castShadow = true;  
  mesh2.position.z = 100 * i;  
  group.add(mesh2);  
}
```

js

根据尺寸数量级设置阴影渲染范围

比如光线是从斜上方照射下来，模型y方向高度100左右，这时候y可以设置为100左右，xz也可以根据渲染范围先给个大概尺寸。

```
directionalLight.position.set(100, 100, 100);  
// 平行光默认从.position指向坐标原点
```

js

光线方向，你可以改变xz坐标来调整

```
directionalLight.position.set(-100, 100, -100);
```

js

渲染范围可以都先给个几百量级的值，不用精准，先设置，在微调。

```
// 设置三维场景计算阴影的范围  
directionalLight.shadow.camera.left = -100;  
directionalLight.shadow.camera.right = 100;  
directionalLight.shadow.camera.top = 100;  
directionalLight.shadow.camera.bottom = -100;  
directionalLight.shadow.camera.near = 0.5;  
directionalLight.shadow.camera.far = 100;
```

js

