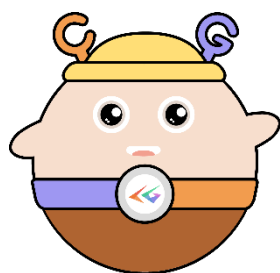


華中科技大學



计算机图形学课程

实验：混合（透明物体处理）

目录

1.混合（Blending）	3
2.开启混合和设置混合模式	3
3.绘制顺序	4
4.排序透明物体.....	4
5.绘制	5
6.实验结果	5

1 混合（Blending）

首先介绍一下 OpenGL 中混合的基本概念。混合是实现物体透明的一种技术。当一个物体是透明的，我们看到的颜色是物体本身的颜色和它背后其它物体的颜色的混合。例如，一个有色玻璃窗是一个透明的物体，玻璃有它自己的颜色，但我们看玻璃所看到的最终的颜色还包含了玻璃之后所有物体的颜色。

1.1 开启混合和设置混合模式

和 OpenGL 大多数的功能一样，我们可以启用 GL_BLEND 来启用混合。

```
// 开启混合
glEnable(GL_BLEND);
```

启用了混合之后，我们还需要告诉 OpenGL 如何混合：

```
//设置混合的源和目标因子
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

上面的代码设置混合的源和目标因子，这里使用源颜色向量（纹理图片的不透明度值）的 **alpha** 作为源因子，使用 **1-alpha** 作为目标因子。这样绘制出来的物体的最终颜色=纹理的颜色向量×**alpha**+颜色缓冲中的颜色向量×（1-alpha）

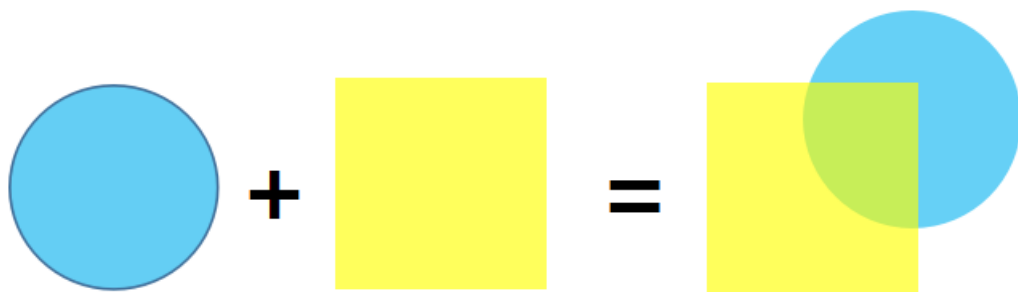


图 1 透过黄色透明物体看圆

1.2 绘制顺序

在绘制一个有不透明和透明物体的场景的时候，必须遵循如下原则：

（1）先绘制不透明物体；

（2）后绘制透明物体：需要按距离摄像头（视点）由远到近的顺序绘制所有透明的物体。

思考一下为什么需要按距离摄像头（视点）由远到近的顺序绘制所有透明的物体？

原因 1：如果先绘制近的物体，后绘制远的物体时，远物被近物相应被遮挡的片段因为深度测试而被丢弃。会出现如图 2 的错误结果。

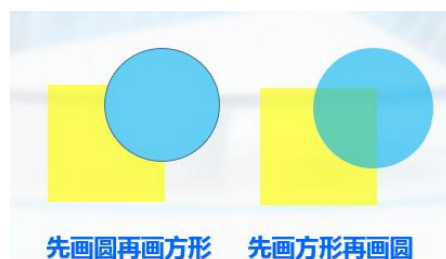


图 2 开启深度测试情况下

原因 2：在关闭深度测试情况下，混合得到的颜色也不对。会出现如图 3 的错误结果。



图 3 关闭深度测试

1.3 排序透明物体

通过需要绘制的物体和摄像头的距离 `distance` 来对物体进行从大到小排序，这里使用 c++ 的 `map` 容器来实现，定义一个 `map<key, value>` 对象，`key` 的值使用物体和摄像头的距离 `distance`，`value` 的值存储物体的坐标，`map` 会自动根据键的值对元素进行从小到大排序，之后我们只需要反向遍历 `map`，就可以按照由远及近的顺序取到物理的位置值。

```
std::map<float, glm::vec3> sorted;
for (unsigned int i = 0; i < windows.size(); i++)
{
    float distance = glm::length(camera.Position - windows[i]);
    sorted[distance] = windows[i];
}
```

1.4 绘制

在绘制循环中，先绘制不透明的物体（箱子和地板）。最后按照距离摄像头由远到近的顺序绘制透明的物体 `window`。

因为 `sorted` 这个 `map` 中的元素是按照从小到大的顺序排序，所有需要使用反向迭代器从后往前，按照物体和摄像头之间距离从大到小的顺序遍历 `map`。

```
for (std::map<float, glm::vec3>::reverse_iterator it = sorted.rbegin();
it != sorted.rend(); ++it)
{
    model = glm::mat4(1.0f);
    model = glm::translate(model, it->second);
    shader.setMat4("model", model);
    glDrawArrays(GL_TRIANGLES, 0, 6);
}
```

1.5 实验结果

实验的最终结果如图 4 所示：

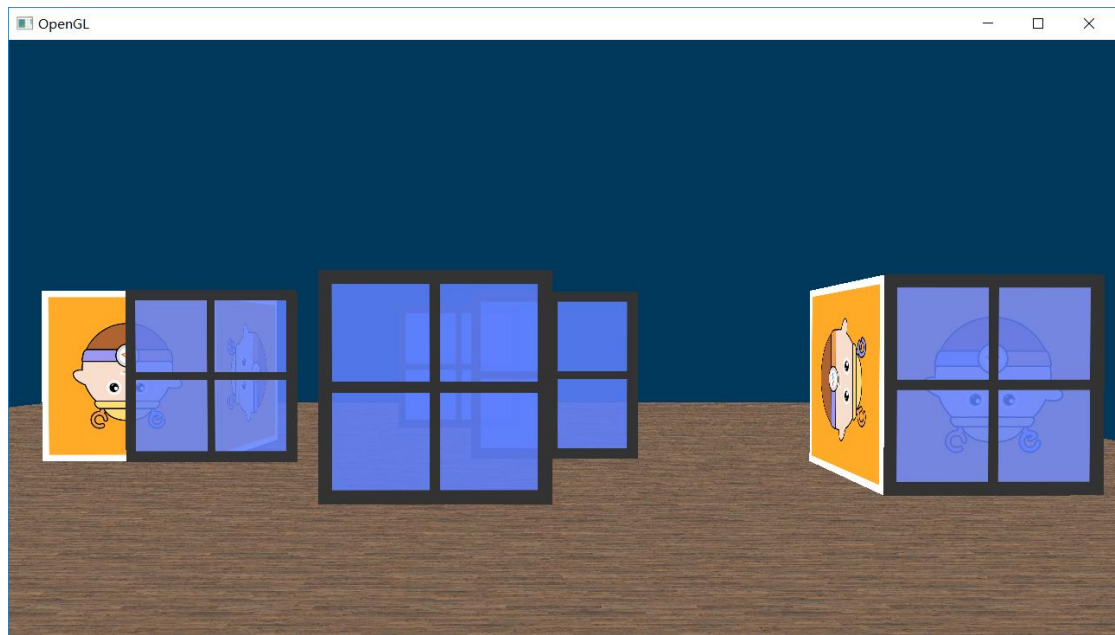


图 4 实验最终结果