

Ray Tracer Week2 Report

范昊翀

bfd08d4

Week2 又是周一交了，好在这次没有熬大夜。不过我看了一下 Book3，内容好像很多，感觉在周四之前写完非常有挑战性... Week2 的内容基本是比较零散的，我将按照 Cornell Box 来分割。

1 BVH 和多线程

虽然 BVH 是中间部分的内容，而多线程书中未做要求，但是这里还是把他们单独拎出来写一段，毕竟如果不写多线程那得渲染到猴年马月。个人感觉下来，BVH 提速大约 30%，多线程提速了 6~7 倍。当然，BVH 的表现之所以不显著，大概是由于在 Week1 的 Final Scene 中，物体个数不是太多（相比之下，Week2 的 Final Scene 则有一处 1000 个球的团块），测一下太久了，实际上 BVH 的提速应该不止这么点。

2 更多球的样式

在 Week1 中，我们已经搭建好了一个基本的光追框架，是时候让我们的世界变得更好看一些了！

2.1 动态模糊

我们的动机还是模拟真实相机的表现，说到真实相机，我们此前一直在拍摄静态的物体，当相机拍摄动态物体的时候，会由于曝光时间并非无限小，使得动态物体的轨迹会被显示在照片上。我们的代码采用了一种聪明的方式实现了这一点。如果我们规定一个球在时间 $t \in [0, 1]$ 内的轨迹，我们把每束“光”随机地在这段时间内发出，这束光判断碰到的物体的位置都是采用的物体在光发出时刻的位置。多束光的结果混合起来，我们就得到了比较符合真实世界的动态模糊效果。

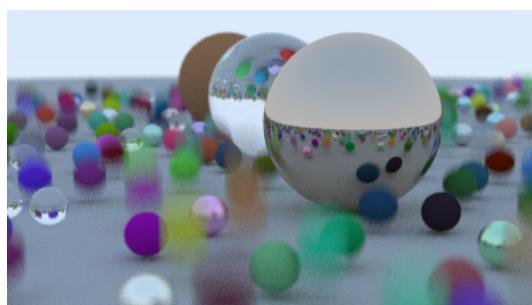


图 1: 弹跳小球的动态模糊

2.2 球的表面贴图

我们在 Games101 中给牛牛表面贴上了奶牛的材质，我们在光追里面也想要做类似的事情。从最简单的贴图开始：我们根据球表面点的 xyz 坐标，取整为一个值，以此决定其颜色，可以得到一个棋盘状的贴图，效果如下：

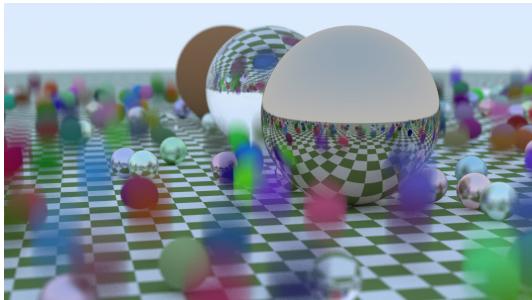


图 2: 把地板替换为了棋盘格材质

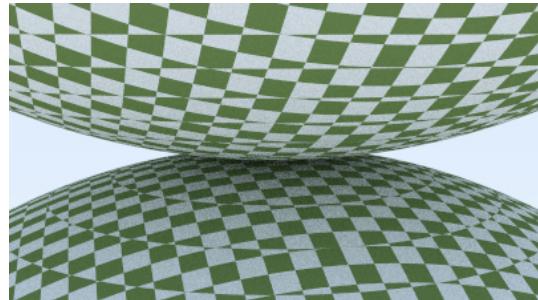


图 3: 两个棋盘格材质的球

接着，我把 Games101 部分的代码移植了过来，让 texture 返回某一坐标对应贴图坐标的颜色，注意到图片导入的时候是在线性空间中，我应该仿照 Week1 中把各个颜色平方，换算到 gamma 空间的颜色。



图 4: 伽马值未修正



图 5: 伽马值修正后

是时候加入更多材质了，在这次的报告中，我邀请了室友百花协助友情出镜，我仿照地球贴图的方式，将他的帅照导入：



图 6: 百花的帅照



图 7: 产生了奇怪的效果

导入材质后产生了奇怪的效果，我意识到这是因为他的帅照是竖屏拍摄的，因此投影到球面上，照片会被横向拉长，因此我重新修改了他的照片，通过拷贝加宽，避免图片失真。



图 8: 修改后的贴图

拉长之后，脸的比例变得正常了很多，我把之前场景中的 lambertian 全部替换为百花的帅照材质，得到的效果如下图所示：

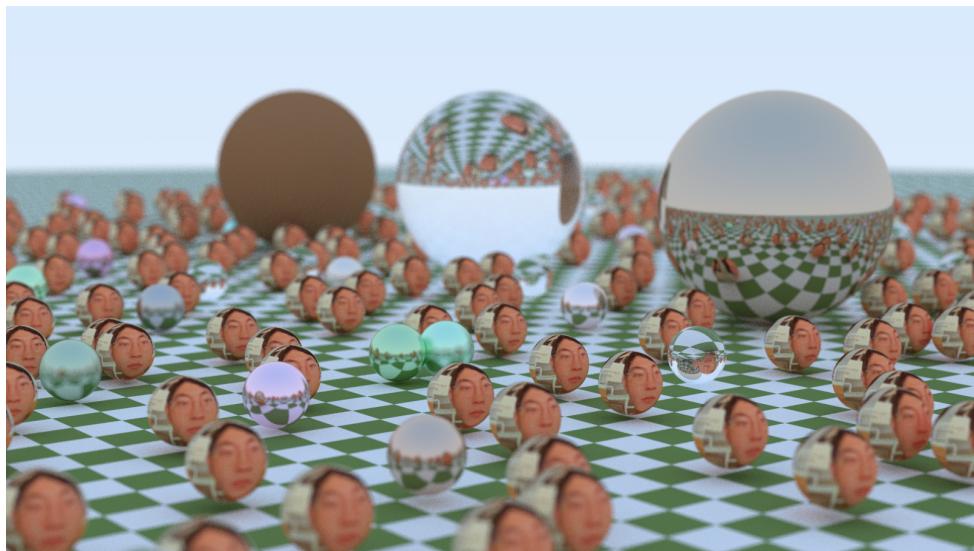


图 9: 遍地百花

实际上，上图是使用多线程进行渲染的。我采用了王宇轩同学所介绍的方式，将图片分割为了若干块（此处为 400 块），交由各个线程渲染，每个线程先将颜色写到 buffer 中，只有在自己这一部分完全渲染完毕，才尝试拿锁，写入图片。此外，我将最大线程数限制在了 20 个，从而平衡多线程的高效率和与之对应的更大的线程间交流的开销。

2.3 柏林噪声

有点懒，直接放最终的图了。（我演讲在讲柏林噪声的时候完全不知道这个居然就是第二周的内容）柏林噪声主要是为了解决普通的白噪声贴图过于 chucky 的问题，我们希望材质更加地连续一点，即相邻两个点的颜色过渡比较连续一些。我们考虑在晶格的角落上放置随机向量，在晶格内部的点的噪声值通过与这些角落向量点积得到。

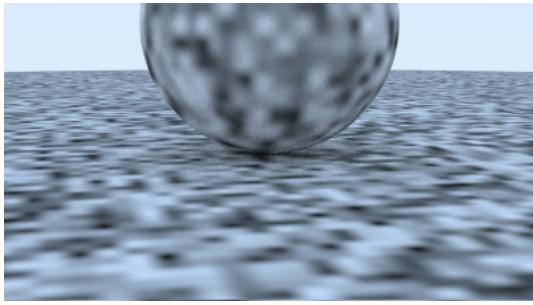


图 10: 朴素的白噪声

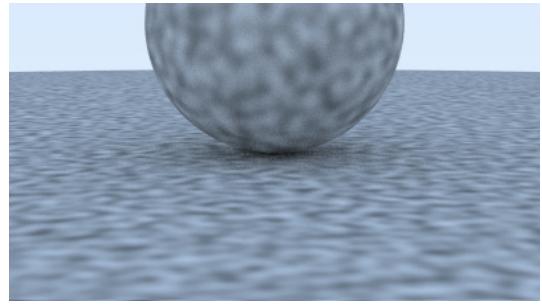


图 11: 点积得出的柏林噪声

为了做出大理石般的材质，我们引入一个叫做“turbulence”的机制，即用高频波函数去干涉在这一点的颜色，这能够保持一定的连续性，同时做出一些表面的纹理，在第一次测试中我忘记把整个东西用 \sin 包起来了，做出了雪球一般的材质，比较有意思，所以放上来了，效果如下。

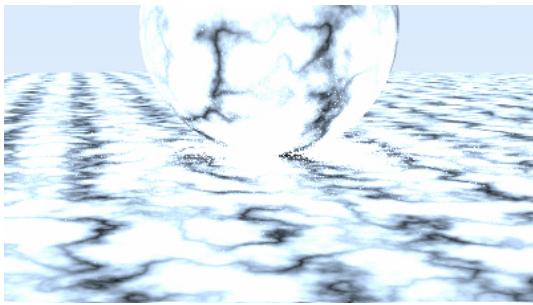


图 12: 雪球材质（迫真雪球）

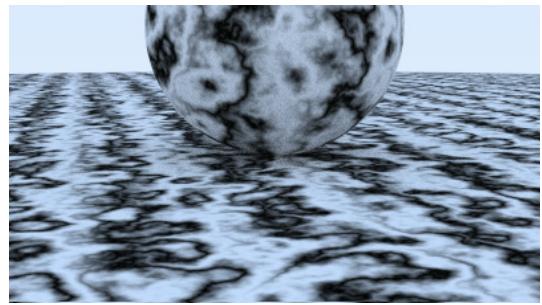


图 13: 大理石材质

3 平行四边形与灯光

在跟随书本实现平行四边形时，我遇到了一个比较弱智的问题，即多次运行得出结果不一样，且都是错的。我在场景中放了三个板，每次都会有一个随机消失：

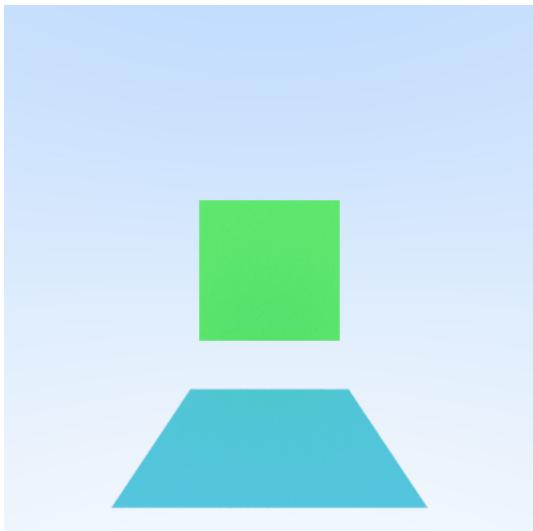


图 14: WTF?

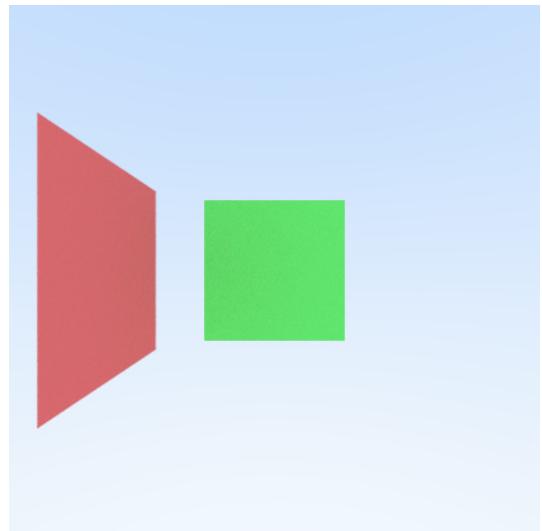


图 15: WTF??

并且，我在场景中放置了一个玻璃球后，他们又显现出来了（什么情况？）

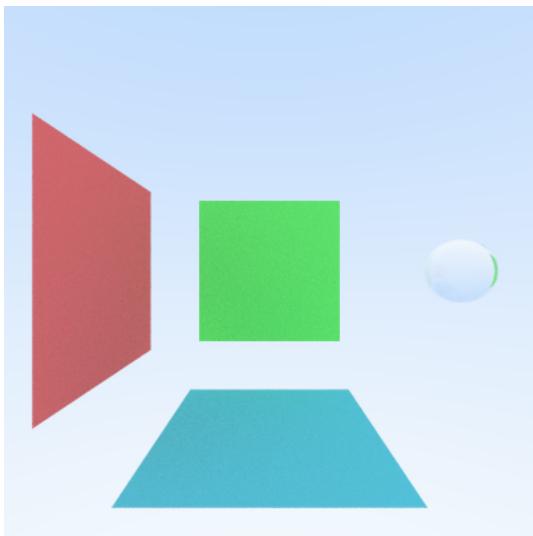


图 16: WTF???

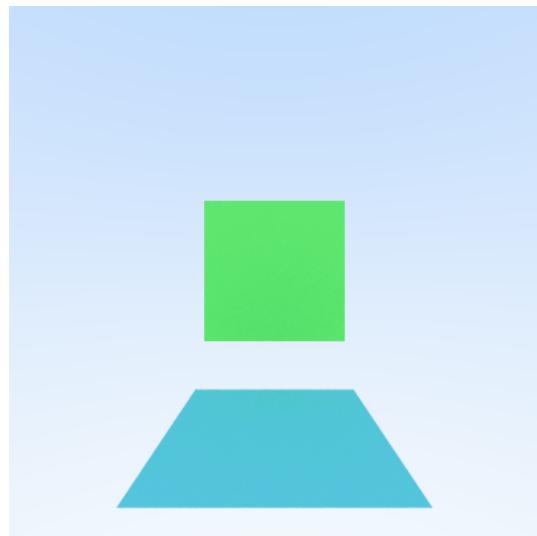


图 17: WTF????

It turns out, 我在 bbox 的其中一个构造函数里面对 bbox 过小的情况做了判断，但另一种忘记了；此外，之前运行结果不同的情况，是因为我仍然采用的是随机选轴排序的 BVH 处理方式，我顺手把这个优化为了每次选最长的轴。

最后可以顺利地做出教程中要求的图片：

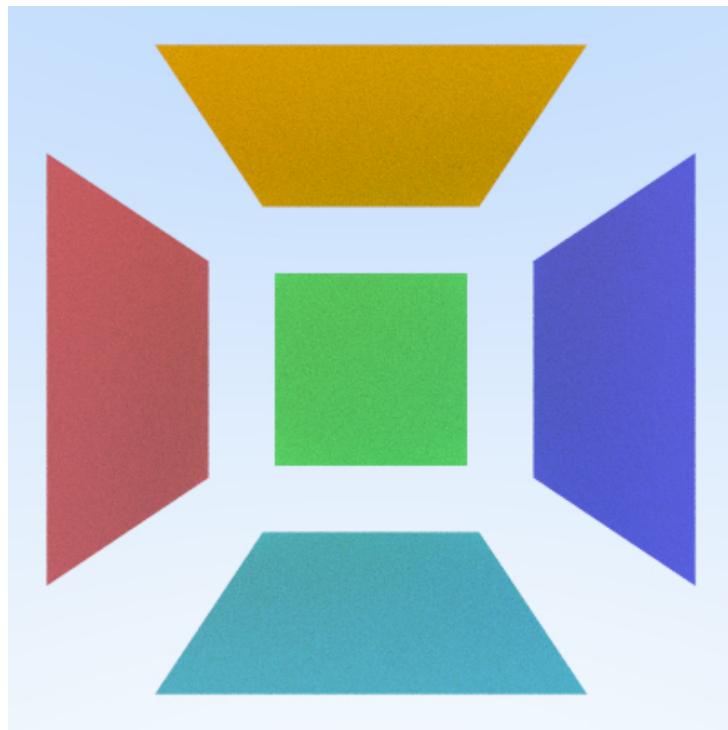


图 18: 五个平行四边形

随后，我们希望做出真正的灯光，于是对于一个 ray，我们将其获取到的颜色分为两部分，一个是其本身的放出的灯光，其次是它本身的颜色，二者加起来可以得到最终的真实颜色。

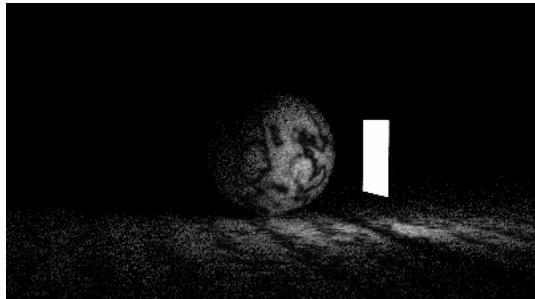


图 19: 单一光源

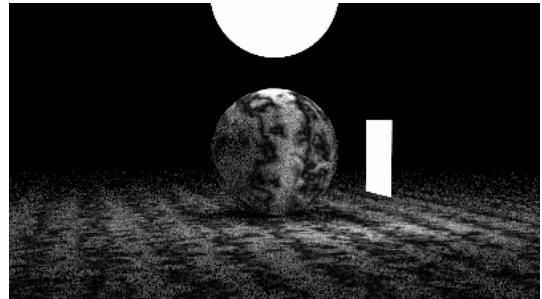


图 20: 双重光源

4 Cornell Box

现在我们已经可以做出一个空的 Cornell Box 了，我们对物体的平移和旋转进行进一步的适配。书中还要求我们做两个新的材质，放在这里有点意义不明，不过观感不错：

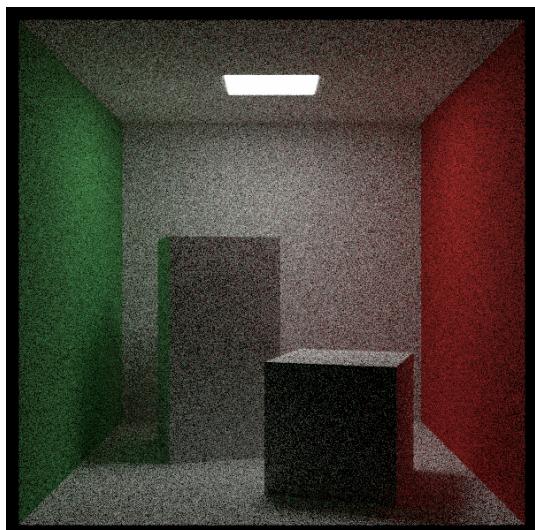


图 21: 物体的旋转平移

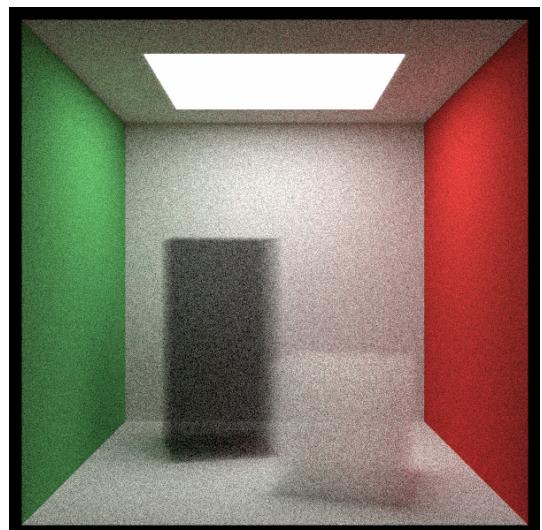


图 22: 烟雾材质

最后是 Final Scene，我的参数设置为 800*800 大小，像素采样 5000 次，最大反射次数为 40

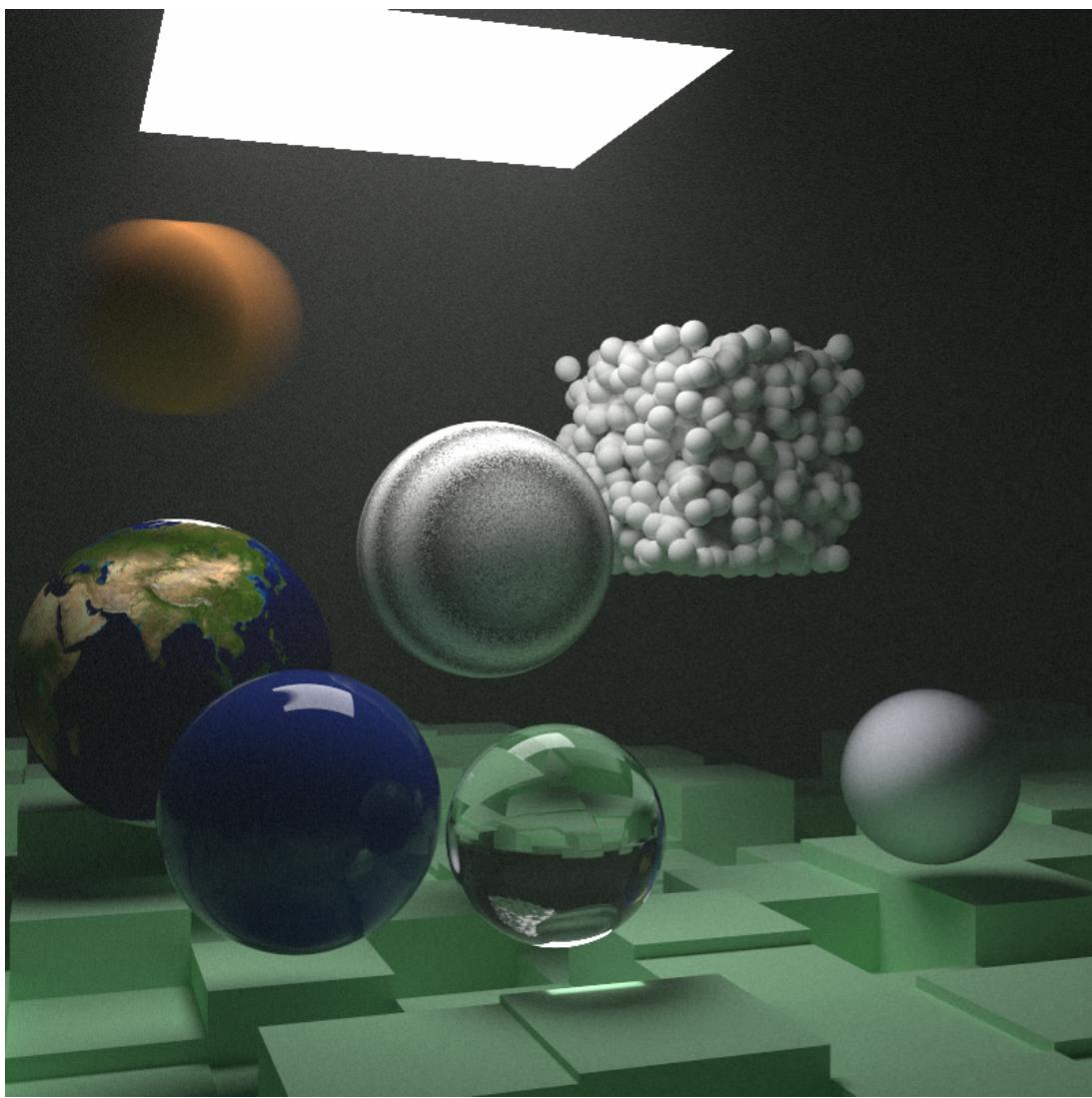


图 23: Final Scene

额外的一个小 bonus, 感谢室友 bubble, 室友 baihua 和室友 Valencia 友情出演

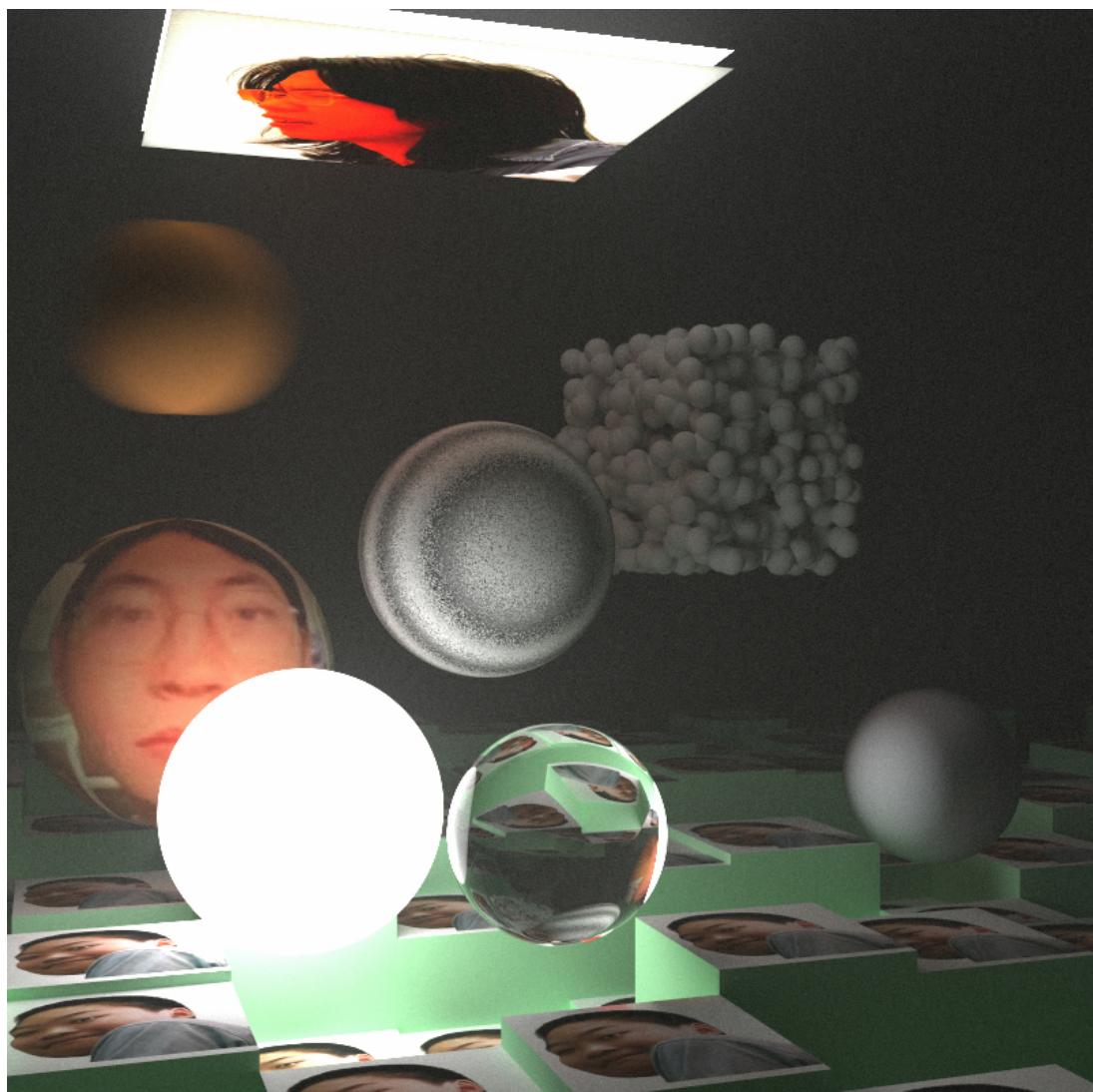


图 24: Final Scene with Everyone