

图形学大作业报告

姓名：袁方舟 学号：2019012376

本作业使用随机渐进式光子映射 (sppm) 算法，支持如下效果：景深、抗锯齿、贴图、凹凸贴图、参数曲面求交、复杂网格模型以及算法加速。结果在results文件夹下，其中result1.bmp和result2.bmp除了一些参数的调整场景基本一样，result3.bmp使用了景深相机。在这个场景里，苹果是贴图和凹凸贴图，墙壁、桌子和书是贴图，旁边的阿尔托花瓶是参数曲面，苹果和兔子（一共7w三角面片）是复杂网格模型以及算法加速，玻璃球和兔子后面是焦散。SPPM自带了抗锯齿效果。result4.bmp是一个人物模型，它展现了程序读取复杂obj的能力。

使用到的资源有：stanford-bunny.obj, galaxy.ppm, desk/desk.obj, desk/desk.ppm, food_apple_01_8k.obj, book_final/book.obj, book_final/book.ppm, miku/append.obj, miku/append.mtl。

由于是Windows下用vscode cmake+mingw64 GCC8.1.0编译，所以有一些路径设置为绝对路径。如果要在linux下编译，可将main函数里的path设定为对应path。path指的是resource路径的上一级。

源码细节

代码在src文件夹和include文件夹下。

Object: `bspline.hpp`, `bspline.cpp`, `group.hpp`, `group.cpp`, `mesh.hpp`, `mesh.cpp`, `plane.hpp`, `sphere.hpp`, `triangle.hpp`。所有基本物体都继承于Object抽象类，它包含生成AABB包围盒、计算交点等函数。Mesh类实现了读入obj文件中的顶点坐标、纹理坐标、法线坐标的方法。场景中所有的Object用ObjectGroup类管理，它不继承任何类。它储存Object指针，和用AABB包围盒生成的KD树结构。KD树加速求交的代码在 `group.cpp`。

Material: `material.hpp`, `material_read.hpp`, `material_read.cpp`。使用Phong模型，可以支持读.mtl文件中的Material。

Color: `color.hpp`, `texture.hpp`。Color基类储存一个基本颜色，Texture类和TriangleTexture类继承它，用于实现纹理。

Ray: `ray.hpp`。储存光线的原点和方向。

Light: `light.hpp`。使用点光源。

Camera: `camera.hpp`。里面有PerspectiveCamera是基本的针孔相机模型，DepthFieldCamera是景深相机，它接受透镜位置和物体位置和光圈大小生成光线。

参数曲面: `bspline.hpp`, `bspline.cpp`, `utils.hpp`。使用17个控制点的3次B样条曲线, 由于场景里做的是一个玻璃花瓶(由两层曲面和上面一个圆环, 下面两个底组成), 所以只需要解3次方程即可。解3次方程的实现在 `utils.hpp`。方程如下:

$$O_x + D_x t = f_x(\theta), O_y + D_y t = f_y(\theta), O_z + D_z t = z$$

其中 θ, t, z 需要在一定的范围。 O, D, f 分别是光线的原点、光线的方向、曲线的函数。由于B样条是分段函数, 所以在具体实现上是每4个控制点组成一个单独的Object计算。首先通过前两个方程可以把 θ 解出来, 然后再判断 θ, t, z 是否合法。3次方程的求解使用先计算判别式再使用三角函数和反三角函数的计算方法。

圆环的实现方法是定一个曲线内部的点, 然后将曲线向这个点收缩一定比例。由于圆环设定为与xy平面平行, 可以先算出x,y,z坐标, 再判定是否有交点。这个新的方程如下:

$$x = f_x(\theta)\beta, y = f_y(\theta)\beta$$

仍然需要解一个三次方程, 不过判定条件有所不同。

SPPM: `sppmhit.cpp`, `scene.hpp`, `scene.cpp`, `main.cpp`。SPPM的算法由Scene和main函数本身实现。main函数里每个round里实现两个pass, 一开始先设定捕捉光子的范围, 对于第一个pass, 它发射相机光线, 调用Scene::trace, 将命中点储存在real_sppmhits。然后它调用Scene::buildHashGrid建立hash grid。然后到第二个pass, 它发射1e6个光子, 光子每到一个漫反射面就调用Scene::updateHit更新命中点参数。每个round的最后使用real_sppmhits更新sppmhit, 将颜色做平均。