图形学大作业实验报告

2018011316

计 82 刘妮琦

一、 框架结构

以 PA1 为基础框架，扩充的内容如下：

头文件/扩充函数	功能
$Constants.h$	宏定义，如面光源采样率，RT 最大递归深度等
$curve.hpp$	$Bezier$曲线的定义，旋转$Bezier$曲面的解析法求交
$object3d.hpp:getColor()$	纹理贴图（继承类中具体实现）
$camera.hpp:generateRay()$	景深效果
$light.hpp-class\ AreaLight$	面光源子类，实现软阴影
$octTree.h+octTree.cpp$	八叉树求交加速
$triangle.hpp$	$AABB$包围盒的定义，三角片构造时包围盒计算
$Square.hpp$	矩形面（纹理贴图时代替无穷大平面）
$RayTracing.hpp$	RT 算法实现
$main.cpp$	超采样抗锯齿

二、 算法选型 RT

典型的 RT 算法。对于每一根从视点发出的光线，与场景求交：若无交点，返回背景色；否则：

1) 对于非阴影点，用$Phong$模型计算该点的局部光强;

2) 若递归深度小于上限，依据交点材质计算反射光线与折射光线，光强衰减，递归跟踪;

3) 记：交点法相 N，视线方向 V（指向视点），入射角θ_1，反射角θ_2，入射介质折射率η_1，折射介质折射率η_2,相对折射率$\eta=\frac{\eta_1}{\eta_2}$

反射光：

$$S = 2 * dot(N,V) * N - V$$

折射光：

$$T = \eta(\cos\theta_1 * N - V) - cos\theta_2 * N$$

三、 实现功能

a) 旋转曲面解析法求交 *curve.hpp_intersect*()

［注：记$P(t)$为参数 t 在$Bezier$曲线上对应的点；固定 y 轴为旋转轴］

原理：记光线与旋转曲面交点P：$(x,y(t),z)$,有（A）：$x^2 + z^2 = x(t)^2$,其中$x(t)$为 P 到旋转轴的距离；联立直线方程$P = \vec{o} + u\vec{d}$，有（B）：$\vec{o}.y + u\vec{d}.y = y(t)$，且（A）中的$x,z$可以用$\vec{o}$、$\vec{d}$表示。最终得到目标函数：

$$h(t) = sqrt(A(y(t)^2 - B) + C) - x(t)$$

$$A = \frac{\vec{d}.x^2 + \vec{d}.z^2}{\vec{d}.y^2}$$

$$B = \vec{o}.y - \vec{d}.y * \frac{\vec{d}.x * \vec{o}.x + \vec{d}.z * \vec{o}.z}{\vec{d}.x^2 + \vec{d}.z^2}$$

$$C = \frac{\left(\vec{d}.x * \vec{o}.z + \vec{d}.z * \vec{o}.x\right)^2}{\vec{d}.x^2 + \vec{d}.z^2}$$

方法：用牛顿迭代法求$h(t)$的零点即可得到 t 的数值解,代入（B）中求 u，得到 P 点坐标，进而计算基曲线的切向量和旋转角$\theta = actan(\frac{P.z}{P.x})$，最终得到交点的法向量。

若$\vec{d}.y$接近 0，则$y(t) = \vec{o}.y$,用牛顿迭代法求出 t 的近似数值，问题化简为求光线与$y = \vec{o}.y$平面上，以$(0,\vec{o}.y,0)$为圆心、$x(t)$为半径的圆的交点。

难点 1：牛顿迭代的初值问题：$h(t)$可能存在多解，只有与视点距离最近的点满足条件；在拒绝不合理的解后，需要更换初值重新迭代。

解决方案：将（0，1）区间划分为多个子段，以各子段的中值作为初值多次迭代。

不足：对于不相交的情况，遍历子段计算，运行速度较慢。

难点 2：对于$\vec{d}.y$较小的点，系数 ABC 较大，算法难以收敛。

解决：对于这些点，先采用$y(t) = \vec{o}.y$的做法求近似解，作为牛顿迭代的初值。

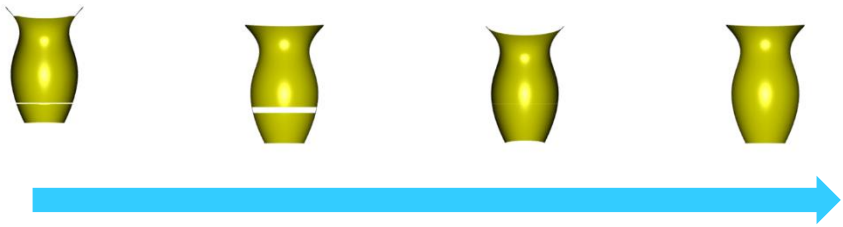

图 1. 旋转曲面——花瓶进化史

b） **景深**_*camera.hpp_generateRay*()

设定相机光圈$Aperture$、光心到焦平面的距离$Focus_Dis$。计算通过各像素点的光线与焦平面的交点，在光圈范围内随机采样作为偏置，得到修正的光线起始点$origin$，连接该点与焦平面上的交点，得到修正的光线方向。

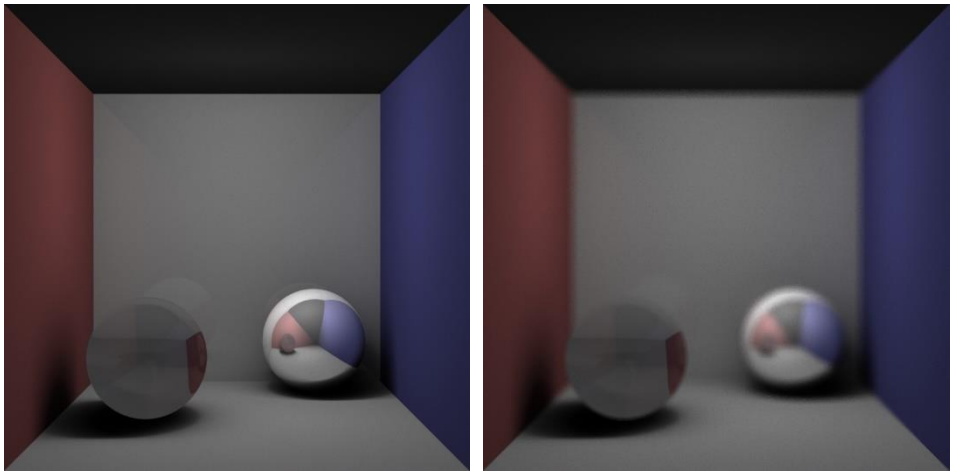

图 2. 景深效果对比（聚焦左侧球）

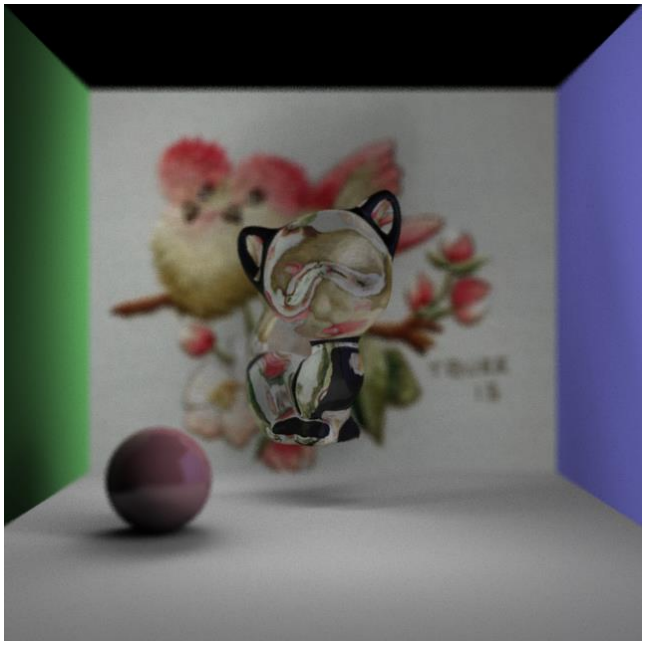

图 3. 景深

c) **抗锯齿**_*main.cpp*

遍历各像素点时，做 SSAA 超采样：随机地取像素点邻域内的四个点发出光线，将结果求均值，作为该像素点的最终颜色。

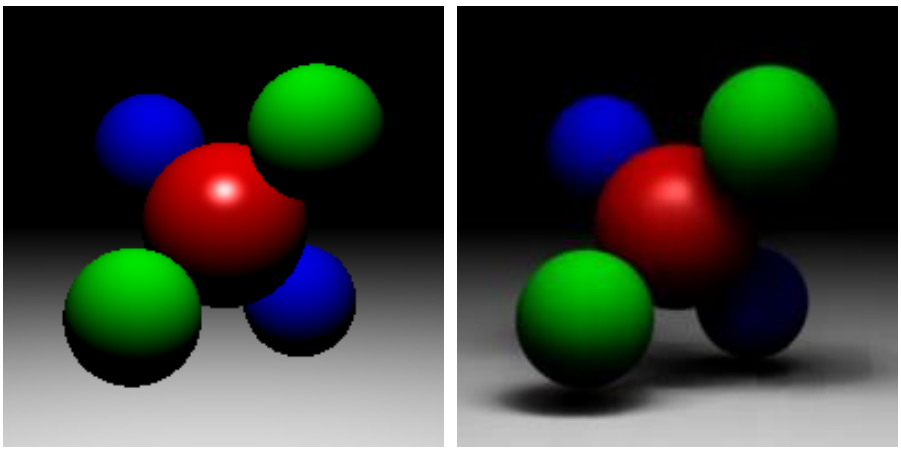

图 4. 抗锯齿效果对比（左侧为 PA1 图）

d) **贴图映射关系**_继承类$addTexture() + getColor()$

[注：记交点为 P，材质纹理高h宽w, 对应点的坐标(x, y)]

- **球**_*sphere.hpp*

场景内的球各自建立球坐标系，计算 P 与 z 轴非负半轴夹角θ、P 在xy平面投影与 x 轴非负半轴夹角φ：

$$x = \frac{\varphi}{2\pi} * w, y = \frac{\theta}{\pi} * h$$

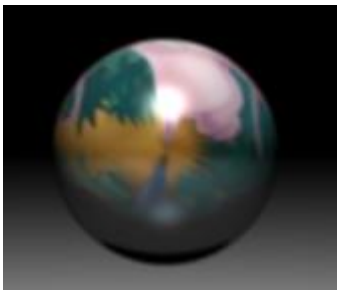

图 5. 纹理贴图——球

- ***Bezier*旋转曲面**_*curve.hpp*

计算$P.y$在$Bezier$基曲线上的y分位数、基曲线旋转角度θ：

$$mp = \frac{P.y - minY}{maxY - minY}，\ \theta = actan(\frac{P.z}{P.x})$$

则：

$$x = \frac{\theta}{2\pi} * w, y = mp * h$$

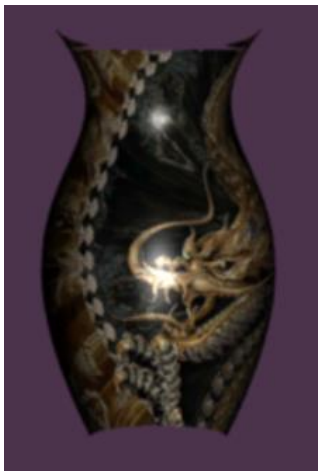

图 6.纹理贴图——旋转曲面

- **矩形面**$\_Square.hpp$

  [从右上角开始，逆时针方向四个顶点记为 ABCD]

  分别计算$\overrightarrow{CP}$在$\overrightarrow{CB}$、$\overrightarrow{CD}$方向的投影yy、xx,则：

  $$x = xx * w, y = yy * h$$

e) **软阴影**$\_light.hpp$

增加圆形面光源$AreaLight$类：$RayTracing$计算局部光强时，面光源随机采样（圆盘均匀采样），采样点视作点光源，计算各自光强后取平均。

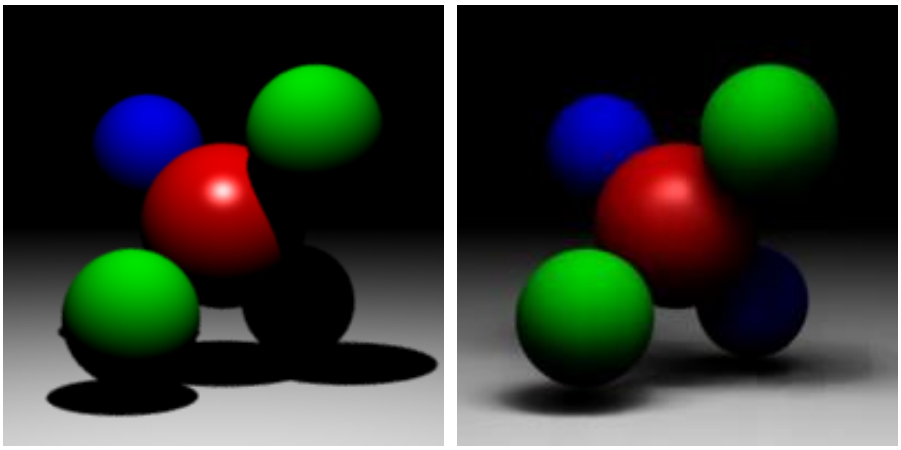

图 7.软阴影

f) **求交加速**——AABB 包围盒+八叉树$_octTree.cpp + triangle.hpp$

每一个三角片在构造函数中计算其 AABB 包围盒。网格模型中增加$octTree$成员，加载所有三角片后建树：子空间采用均匀划分方式，当三角片的 AABB 包围盒与子空间有交叠时，将该三角片分配给节点；树高不超过 8 层。求交时，将光线方向向量归一到第一卦限，与子空间进行求交测试；若在叶节点相交，则遍历节点内三角片，逐片求交。

速度对比（以 1000 片斯坦福兔子为例，有抗锯齿等附加效果）：

八叉树+包围盒	直接遍历各三角片
13.140625s	300.21875s

g） 法向插值_*mesh.cpp + triangle.hpp*

网格加载三角片时：

1）计算各三角片的法向量；

2）对于每个顶点，取关联的所有三角片的法向量的均值作该顶点的法向量

求交时，以三角片的三个顶点为基底表示交点；取基底坐标的三个分量分别为三个顶点法向量的权值，计算交点的法向量。

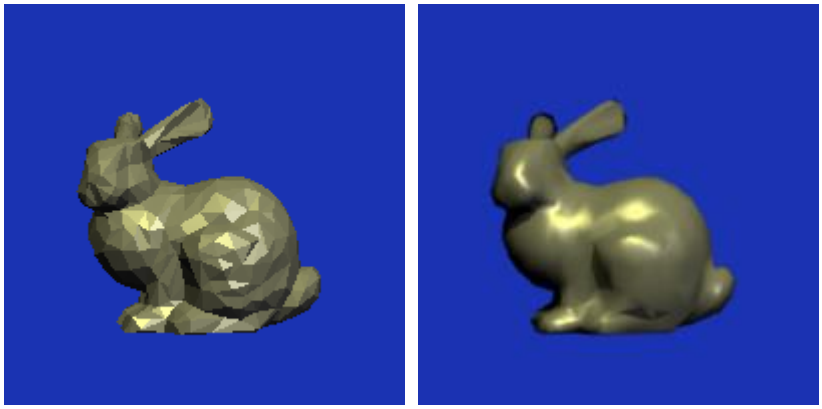

图 8.法向插值

四、 效果图说明

两张成图（见“Result”文件夹下.png 文件）大小均为 640*640。花瓶图实现旋转曲面、软阴影、抗锯齿、纹理贴图（光圈较小，景深不明显）；兔子图实现景深、软阴影、抗锯齿、纹理贴图、求交加速、法向插值。