

# 作业 6: 加速结构

## GAMES101, 2020 年春季

教授: 闫令琪

计算机图形学与混合现实研讨会

GAMES: Graphics And Mixed Environment Seminar

发布日期为北京时间 2020 年 3 月 31 日 (星期二) 晚上**23: 59**

截止日期为北京时间 2020 年 4 月 10 日 (星期五) 晚上**23: 59**

---

### 注意:

- 任何更新或更正都将发布在论坛上, 因此请偶尔检查一下。
  - 论坛链接: <http://games-cn.org/forums/forum/graphics-intro/>。
  - 你必须独立完成自己的作业。
  - 你可以在论坛上发布帖子求助, 但是发布问题之前, 请仔细阅读本文档。
  - 在截止时间之前将你的作业提交到 SmartChair 上。
-

## 1 总览

在之前的编程练习中，我们实现了基础的光线追踪算法，具体而言是光线传输、光线与三角形求交。我们采用了这样的方法寻找光线与场景的交点：遍历场景中的所有物体，判断光线是否与它相交。在场景中的物体数量不大时，该做法可以取得良好的结果，但当物体数量增多、模型变得更加复杂，该做法将会变得非常低效。因此，我们需要加速结构来加速求交过程。在本次练习中，我们重点关注物体划分算法 Bounding Volume Hierarchy (BVH)。本练习要求你实现 Ray-Bounding Volume 求交与 BVH 查找。

首先，你需要从上一次编程练习中引用以下函数：

- **Render()** in `Renderer.cpp`: 将你的光线生成过程粘贴到此处，并且按照新框架更新相应调用的格式。
- **Triangle::getIntersection** in `Triangle.hpp`: 将你的光线-三角形相交函数粘贴到此处，并且按照新框架更新相应相交信息的格式。

在本次编程练习中，你需要实现以下函数：

- **IntersectP(const Ray& ray, const Vector3f& invDir, const std::array<int, 3>& dirIsNeg)** in the `Bounds3.hpp`: 这个函数的作用是判断包围盒 `BoundingBox` 与光线是否相交，你需要按照课程介绍的算法实现求交过程。
- **getIntersection(BVHBuildNode\* node, const Ray ray)** in `BVH.cpp`: 建立 BVH 之后，我们可以用它加速求交过程。该过程递归进行，你将在其中调用你实现的 `Bounds3::IntersectP`。

## 2 开始实现

### 2.1 编译运行

基础代码只依赖于 CMake，下载基础代码后，执行下列命令，就可以编译这个项目：

```
1 $ mkdir build
2 $ cd ./build
3 $ cmake ..
4 $ make
```

在此之后，你就可以通过 `./Raytracing` 来执行程序。

### 2.2 代码框架

我们修改了代码框架中的如下内容：

- Material.hpp: 我们从将材质参数拆分到了一个单独的类中，现在每个物体实例都可以拥有自己的材质。
- Intersection.hpp: 这个数据结构包含了相交相关的信息。
- Ray.hpp: 光线类，包含一条光的源头、方向、传递时间  $t$  和范围 `range`。
- Bounds3.hpp: 包围盒类，每个包围盒可由 `pMin` 和 `pMax` 两点描述（请思考为什么）。`Bounds3::Union` 函数的作用是将两个包围盒并成更大的包围盒。与材质一样，场景中的每个物体实例都有自己的包围盒。
- BVH.hpp: BVH 加速类。场景 `scene` 拥有一个 `BVHAccel` 实例。从根节点开始，我们可以递归地从物体列表构造场景的 BVH。

## 3 提交与评分

评分：

- [5 points] 提交格式正确, 包含所有需要的文件; 代码可以在虚拟机下正确编译运行。
- [20 points] 包围盒求交: 正确实现光线与包围盒求交函数。
- [15 points] BVH 查找: 正确实现 BVH 加速的光线与场景求交。
- [加分项 20 points] SAH 查找: 自学 SAH(Surface Area Heuristic), 正确实现 SAH 加速, 并且提交结果图片, 并在 README.md 中说明 SVH 的实现方法, 并对比 BVH、SVH 的时间开销。(可参考 [http://15462.courses.cs.cmu.edu/fall2015/lecture/acceleration/slide\\_024](http://15462.courses.cs.cmu.edu/fall2015/lecture/acceleration/slide_024), 也可以查找其他资料)
- [-5 points] 未提交 README.md, 未提交 CMakeLists.txt, 未提交结果图片, 未完整提交代码, 提交包中多余文件 (比如 /build, /.vs) 未清除。



提交:

- 当你完成作业后, 请清理你的项目, 记得在你的文件夹中包含 CMakeLists.txt 和所有的程序文件 (无论是否修改);
- 同时, 请新建一个 /images 目录, 将所有实验结果图片保存在该目录下;

- 再添加一个 README.md 文件写清楚自己完成了上述得分点中的哪几点（如果完成了，也请同时提交一份结果图片），并简要描述你在各个函数中实现的功能；
- 最后，将上述内容打包，并用“姓名\_Homework6.zip”的命名方式提交到 SmartChair 平台。