

计算机图形学第一次作业报告

张雯莉

5090379039

zwl.sjtu@gmail.com

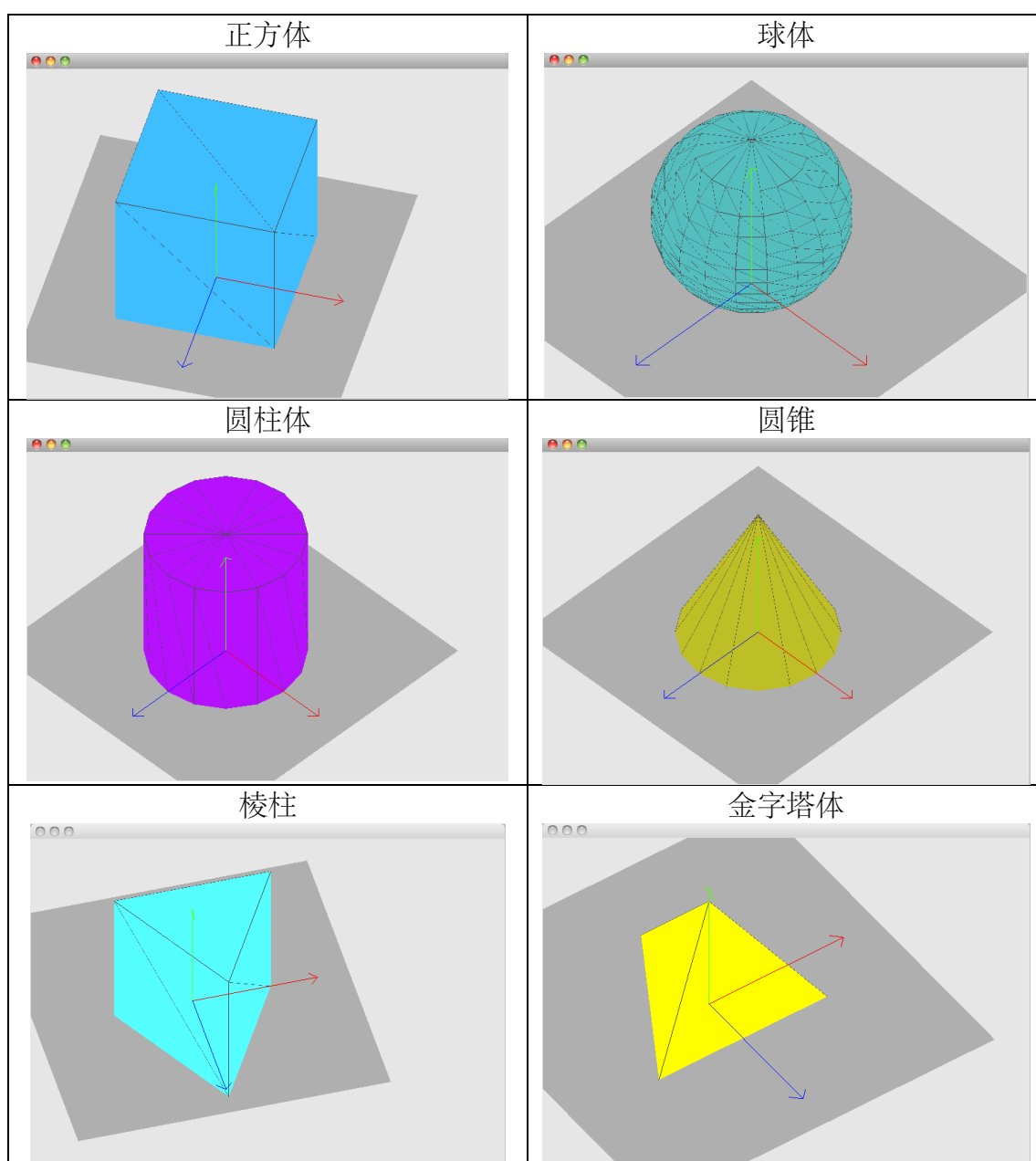
作业要求

实现 CSG 实体建模。实现五种以上体素的 CSG 建模，暂只实现并操作，用户可交互修改任何体素的几何参数和其他属性。体素种类超过 5 种可获加分，实现体素交和差计算可获加分。

完成内容

1. 六种体素的建模

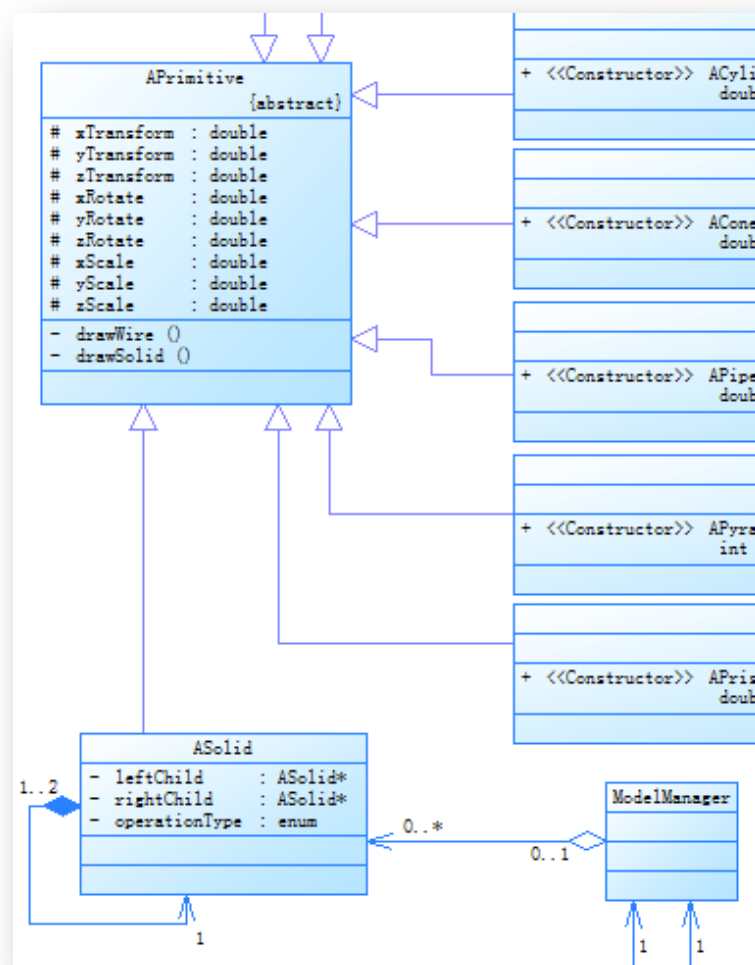
正方体、球体、圆柱体、圆锥、棱柱、金字塔体



为了方便布尔操作，并没有直接调用库函数绘制，而是全部采用三角面片的方式表示体素。用 Vector3d 结构记录每个点的坐标值，在用 Vector3i 结构记录组成三角面片的点的序号。

2. 布尔操作

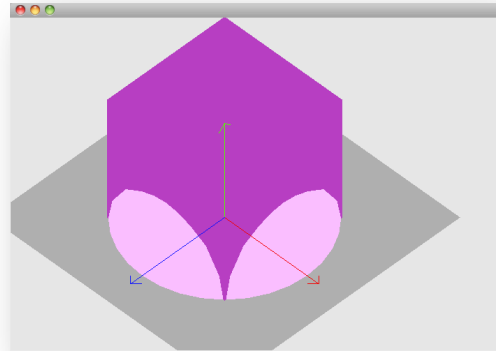
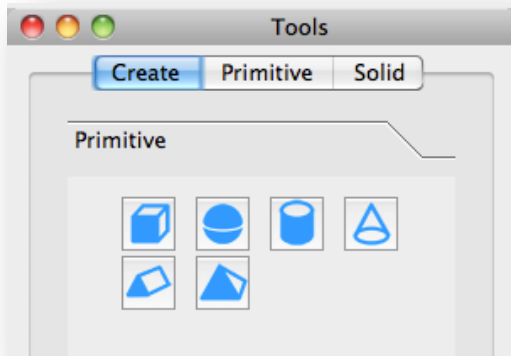
我花了大量的时间研究 CSG 建模布尔操作的方法，阅读了一些论文和网上的资料。经过研究发现，现有 OpenCSG 库提供了很好的方法，但为了学习，我并没有直接调用库函数，而是仔细阅读了作者写的论文。基本理解了基于光线透视的 Gold Feather 算法。由于时间关系，目前还没有把这算法实现。但考虑到将来会采用这种算法，在建立数据结构的时候已经把 CSG 树实现了。具体关系如下：



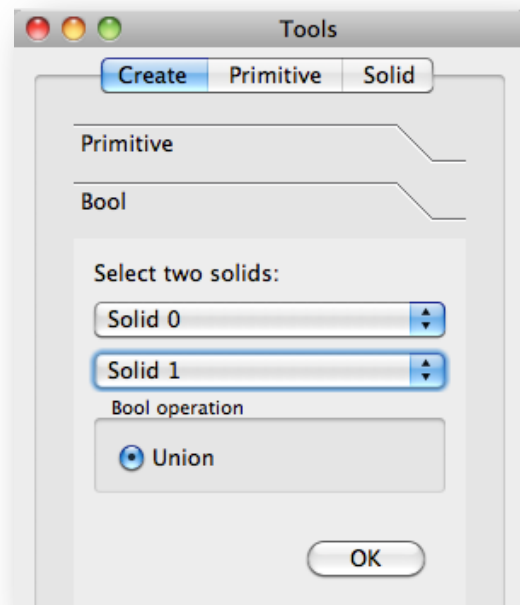
其中，`ASolid` 是 CSG 树，分别持有左右子树 `ASolid` 指针类型的 `leftChild` 和 `rightChild`，以及表示交并差操作的枚举类型。`ASolid` 的叶子节点是 `APrimitive`，这是基本体素（正方体 `ACube` 等）的基类。通过基本体素的布尔操作，实现各种复杂的体素。

由于花了大量的时间研读论文，没有充裕的时间完成交、差操作，将会在以后的迭代中完成。现在实现了体素的并操作，具体操作方式如下：

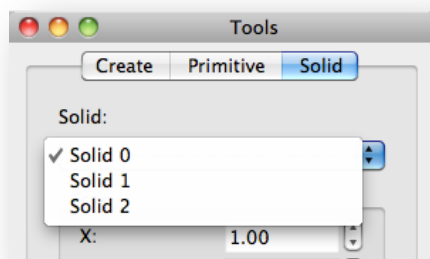
- a) 建立两个基本体素（图示为 Cube 和 Sphere 的组合）



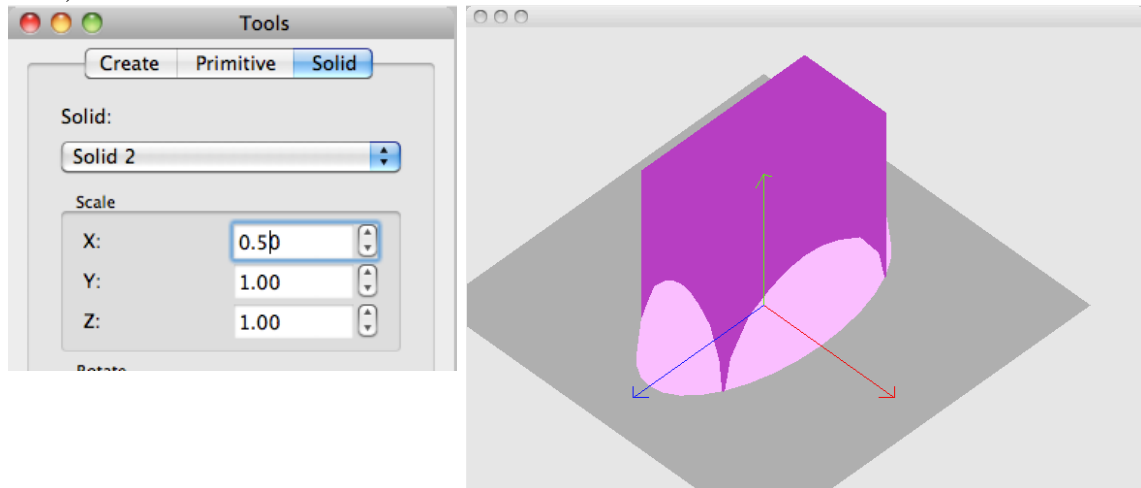
- b) 在 Create 面板中选择 Bool
c) 选择基本体素和布尔操作（目前只有并操作）



- d) 点击 OK 按钮
e) 可以在 Solid 面板中发现多了一项，就是组合后的结果

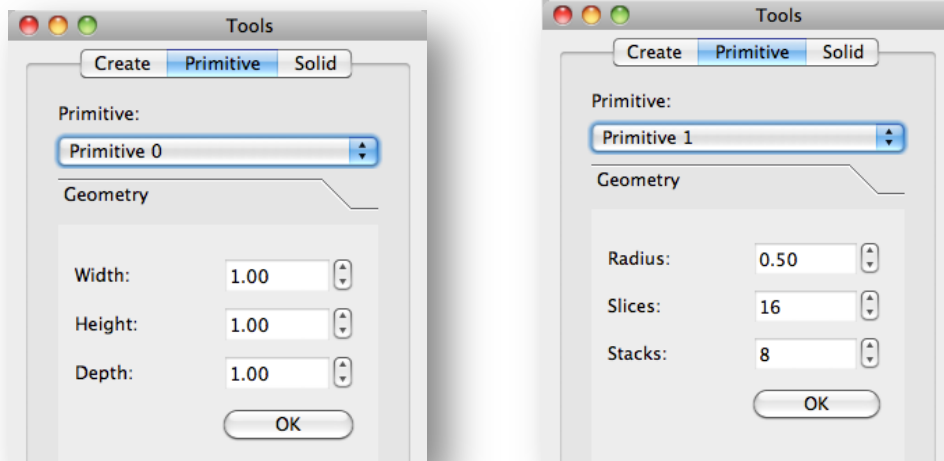


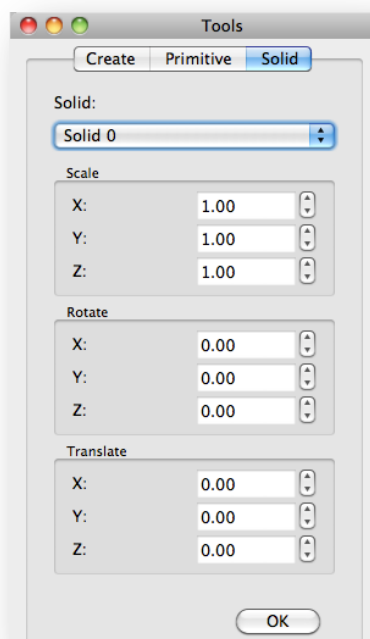
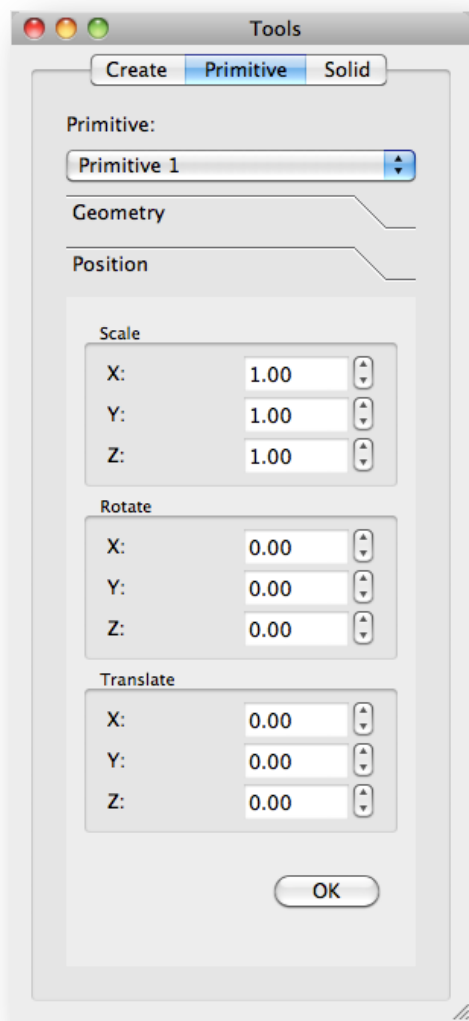
f) 在 Solid 面板中的操作将作用于组合整体



3. 修改参数

用户可以通过 GUI 修改基本体素（对应 APrimitive 类）和组合后的 CSG 树（对应 ASolid 类）的几何属性以及旋转、移动、缩放参数。其中，不同的基本体素对应不同的参数，如球体有 radius, slices, stacks 属性。





请注意，Primitive 面板中对应的参数是指对基本体素的设置，而 Solid 面板中的参数是指经过布尔操作后的组合的设置。

4. Wire 和 Solid 两种显示方式

借鉴 glut 中 glutSolidxxx 和 glutWirexxx，提供两种显示方式。工具栏中

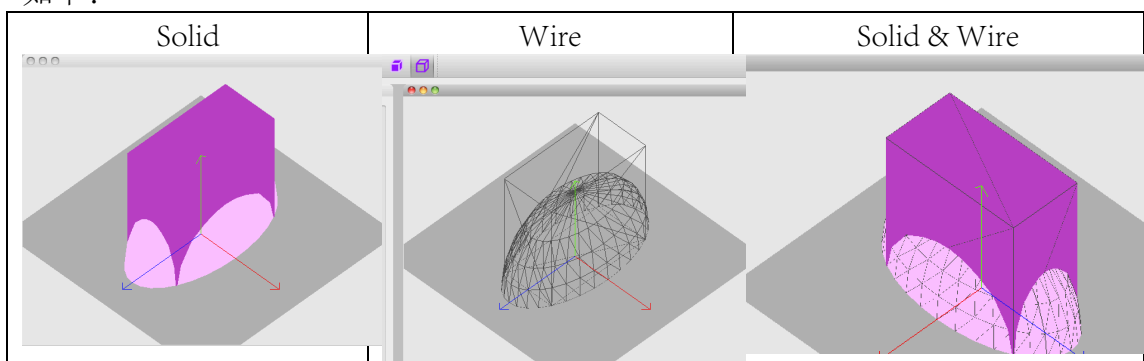


表示 Solid，



表示 Wire，可以同时开启，但至少必修开启一个。效果

如下：



5. 场景参数的设置



工具栏的左边部分是控制场景的，从左到右分别表示：移动场景、放大场景、缩小场景、绕 X 轴旋转、绕 Y 轴旋转、绕 Z 轴旋转。

由于时间关系，只有移动场景是可以拖动鼠标完成的，其他操作都是点击一次执行一次，会在将来的迭代中进行改进，学习像 3dmax 等建模软件更有好的交互方式。

环境配置

本项目采用 Qt 跨平台开发，bin 文件夹下的 exe 文件是在 Windows 下编译的可执行文件，应该可以直接执行。如果有问题的话，src 文件夹内有 pro 文件，是 Qt Creator 的工程文件，可以用来新建项目。

阶段小结

这是我第一次阅读论文，最初的困难在于，即使能读懂意思，对算法真正在说什么显得很困惑。在读到第三天的时候，突然有点豁然开朗的感觉，也逐渐发现 Gold Feather 算法的精妙之处。其中让我印象最深刻的是“只有交集的前面和差集的背面可能被绘制”，我原先一直不明白如何把画出来的某些区域擦除，而 Gold Feather 算法就另辟蹊径，直接计算出哪些部分是从视角处看得见的。通过这阶段的项目，让我接触到了 CSG 建模的方法，一开始毫无头绪只是想挑战一下这个看上去比较困难的项目。在了解了实现的方法后，我被复杂的算法搞得一筹莫展，甚至有过放弃的念头。经过肖老师的指点，我慢慢找到了突破口，并在耗费了大量学习时间的前提下完成了代码的实现。虽然现在做到的功能和我原先预期的还有一点差距，但我也在其中找到了乐趣，会在将来的迭代中对其不断改进。

参考资料

1. OpenCSG: A Library for Image-Based CSG Rendering
介绍了 OpenCSG 和相关算法
http://static.usenix.org/event/usenix05/tech/freenix/full_papers/kirsch/kirsch.pdf
2. Rendering Techniques for Hardware-Accelerated Image-Based CSG
对 Gold Feather 和其他算法的详细描述，让我豁然开朗的文章
http://wscg.zcu.cz/wscg2004/Papers_2004_Full/M11.pdf
3. A z-buffer csg rendering algorithm for convex objects
其中包含对 Gold Feather 的伪代码描述，颇有裨益
<http://www.nigels.com/research/wscg2000.pdf>
4. Constructive Solid Geometry ppt
CSG 数据结构的介绍
<http://www.cs.bath.ac.uk/~djp/30075-CG-7-csg.ppt>