自适应的空间划分法加速三角形求交

报告人: 谭焱

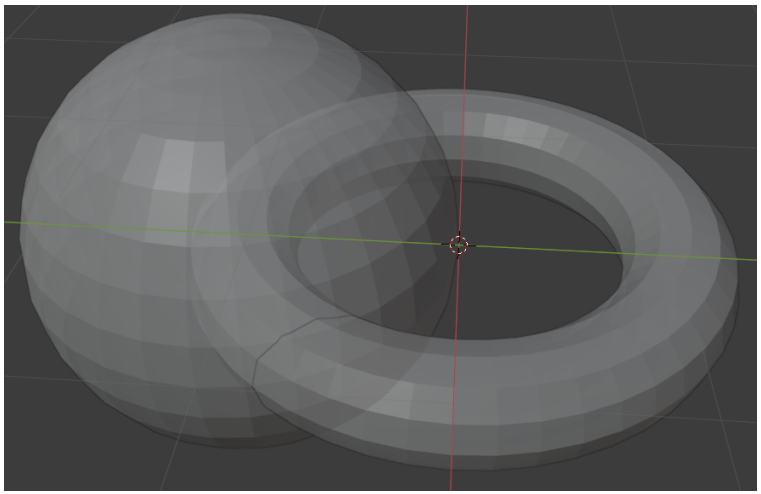
小组成员: 邱云昊，谭焱

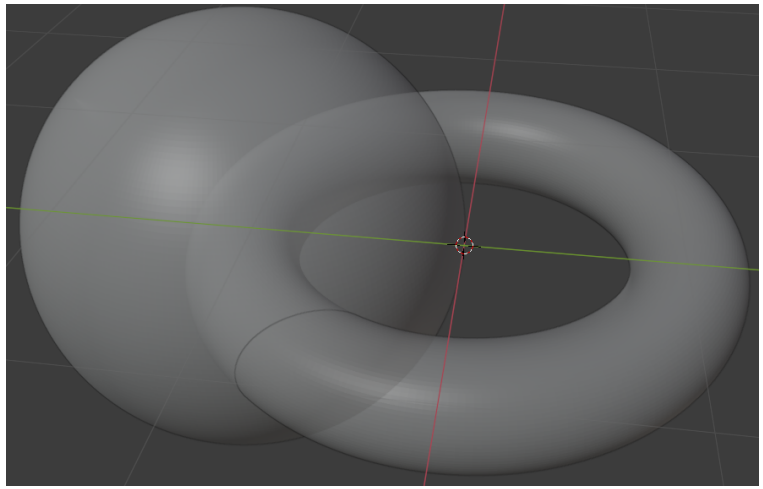
December 22, 2021

提纲

研究背景

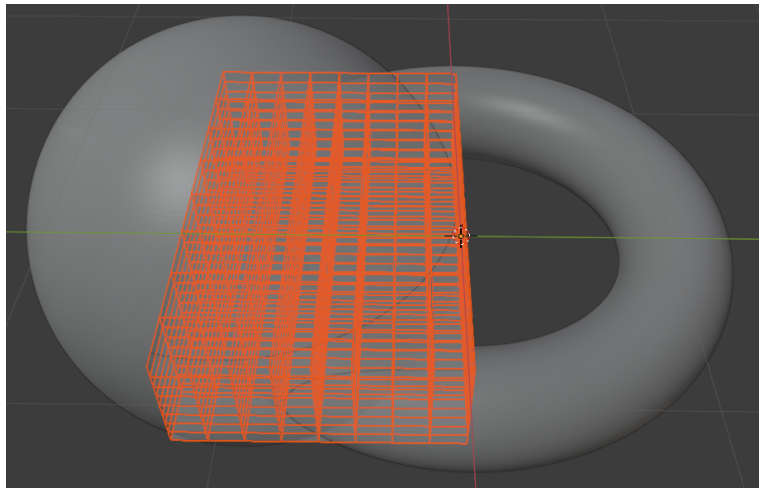
1. 使用般集对三维空间中有物理意义的区域建模.
2. 为般集实现布尔代数, 提供研究流相拓扑变化的工具.
3. 实现布尔运算时需要计算空间三角形的交线.
4. 已有求交算法将所有三角形两两求交,
   - 时间复杂度是 $O(N^2)$.
   - 是当前布尔运算的时间瓶颈 (2112 个三角形时占总时长 0.20 / 0.52, 359424 个三角形时为 115.39 / 140.36).
   - 次要时间瓶颈是三角剖分 (分别占用时间 0.26 / 0.52, 19.65 / 140.36).
5. 需要高效的大量空间三角形求交方法, 有效的提高布尔运算速度.



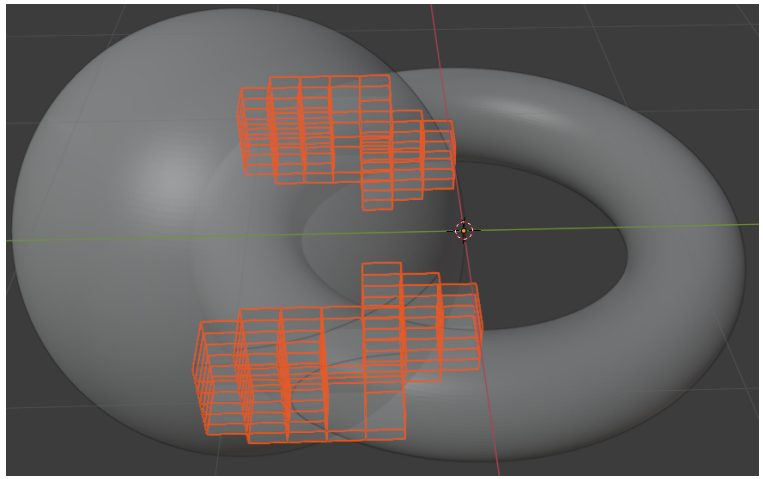


空间划分

1. 三角形相交是局部的, 考虑局部区域中的三角形两两求交. 如图所示

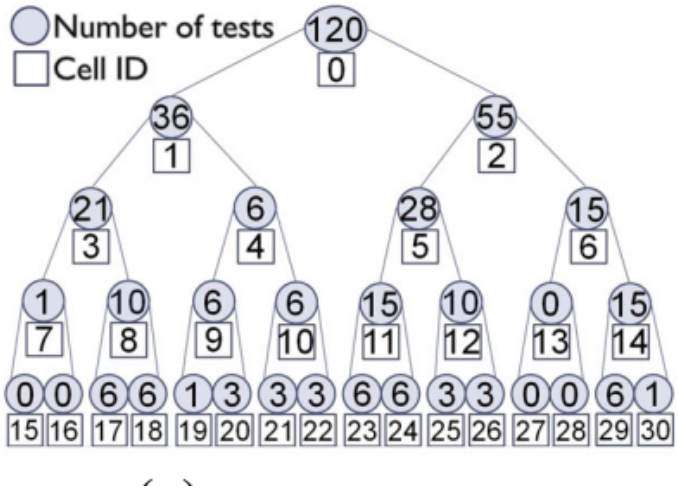


2. 不考虑一个般集边界上的三角形相交, 大部分区域可以忽略.

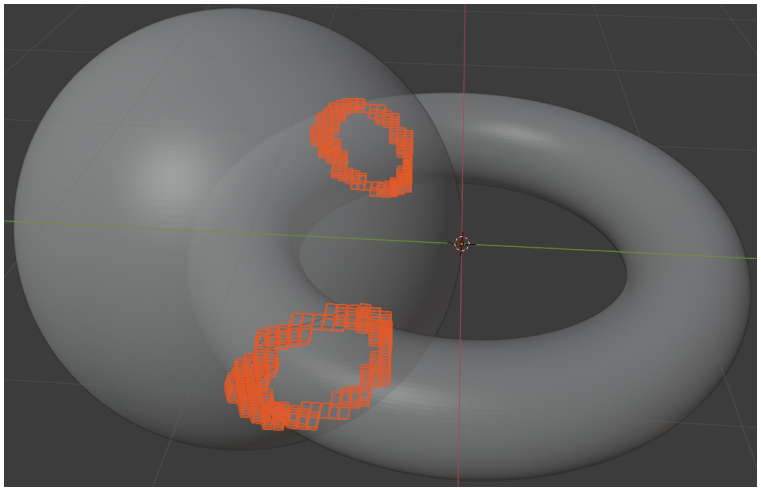


自适应的空间划分

1. 在三角形密集的局部进行更细致的划分, 结合剪枝算法.



2. 得到的自适应空间划分如下



代码结构

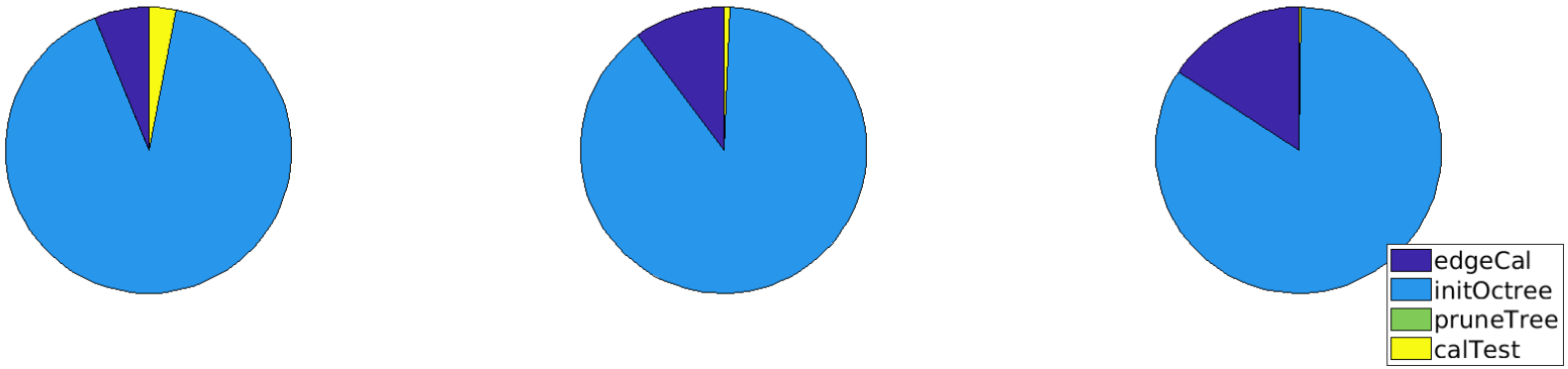
1. 类接口
   - 输入: 两组空间三角形, 空间划分树的最大深度.
   - 输出: 每个三角形上的所有交线段.
2. 建立空间划分树的节点

```
1  template <class T>
2  struct OctreeNode {
3    T val;
4    std::vector<int> tris[2];
5    std::vector<OctreeNode<T>*> child;
6  };
```

3. initOctree() 生成八叉树, 每个节点的子节点是长方体的 8 等分之一.
4. pruneTree() 对树进行剪枝, 最小化三角形求交计算的次数.
5. calTest() 遍历八叉树, 计算所有叶节点长方体中的三角形求交.

时间复杂度分析

1. 设两个般集边界分别约有 N 个三角形.
2. 令八叉树深度为 $\log_8 N$, 可理解为八叉树将计算空间 N 等分.
3. 不妨设般集的三角形不会过大, 即覆盖的长方体数量为常数.
4. initOctree() 的时间复杂度为 O(NlogN).
5. pruneTree() 的时间复杂度为 O(N).
6. calTest() 最坏情况时间复杂度为 $O(N^2)$, 但在普遍情况是 $O(N^{\frac{1}{2}})$.
7. 综上, 该算法时间复杂度为 O(NlogN).

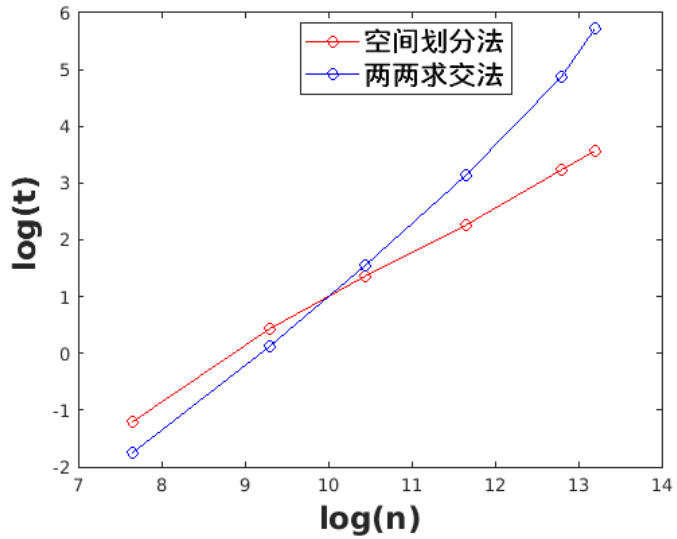


(a) (960, 1152) (b) (16128, 18432) (c) (261120, 98304)

Figure 1: 边界上 (x1, x2) 个三角形的般集求交

计算时间测试

1. 自适应的空间划分法和已有方法计算时间比较如下



三角形数量/个	2112	10880	34560	114176	359424	539392
空间划分法/s	0.302020	1.549209	3.878483	9.649845	25.118267	35.477708
两两求交法/s	0.175366	1.140628	4.671456	22.934571	129.951804	306.352480

Table 1: 新老方法计算时间对比

2. 在当前测试图形中三角剖分消耗时间略小于空间划分法求交.

待解决的问题

1. 程序没有通过所有测试.
2. 判断长方体包含三角形的程序实现低效.
3. 测试样例简单单一.
4. 时间复杂度分析不严谨.
5. 是否要在殷集内部添加可以表示三角形相邻关系的数据结构?
6. 是否要继续优化?