

复杂场景中基于拓扑空间网格的碰撞检测算法^①

王振文¹, 徐 华²

¹(北京化工大学 信息科学与技术学院, 北京 100029)

²(北京石油化工学院 信息工程学院, 北京 102617)

摘 要: 为了提高复杂场景的碰撞检测效率, 提出一种基于拓扑空间网格的碰撞检测算法. 由于场景中存在着众多形状复杂、尺寸不一且运动状态不同的物体, 首先采取场景预处理对空间进行均匀八叉树网格划分, 建立物体方向包围盒层次树与空间网格拓扑结构, 利用静态大尺寸物体分割策略提升定位精确性, 然后在实时检测中利用拓扑空间网格及投影相交测试排除大量不相交物体对, 利用层次包围盒算法对潜在碰撞对进行精确检测并计算出碰撞点. 实验结果表明, 本算法有效地提高了实时检测的效率, 适用于复杂虚拟场景中的碰撞检测.

关键词: 碰撞检测; 拓扑空间网格; 方向包围盒; 大尺寸物体分割; 层次包围盒

引用格式: 王振文, 徐华. 复杂场景中基于拓扑空间网格的碰撞检测算法. 计算机系统应用, 2017, 26(12): 116-123. <http://www.c-s-a.org.cn/1003-3254/6031.html>

Collision Detection Algorithm in Complex Scenes Based on the Topological Spatial Grid

WANG Zhen-Wen¹, XU Hua²

¹(College of Information Science & Technology, Beijing University of Chemical Technology, Beijing 100029, China)

²(College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China)

Abstract: To improve the efficiency of collision detection in complex scenes, a collision detection algorithm based on topological spatial grid is proposed in this paper. Because there are numerous objects with complex shapes, different scales and motion states in the scene, the algorithm firstly partitions the scene with uniform octree, builds oriented bounding box hierarchical tree, sets up the topology of spatial grids and conducts static large-scale object decomposition for accurate positioning in preprocess stage. Then, the algorithm excludes plenty of non-intersect objects with topological spatial grid and projection intersection test, by using bounding volume hierarchy to conduct accurate detection of potential collision pairs and calculates collision points in real-time detection process. Experimental results show that this algorithm improves the real-time detection efficiency, which is suitable for collision detection in complex virtual scenes.

Key words: collision detection; topological spatial grid; oriented bounding box; large-scale object decomposition; bounding volume hierarchy

碰撞检测在计算机动画模拟、虚拟仿真、游戏开发等领域都是关键技术. 随着虚拟现实的发展, 虚拟场景越为复杂, 物体对象逐步增多, 对碰撞检测算法的实时性提出了更高的要求^[1].

基于物体空间的碰撞检测算法中比较经典的两类

碰撞检测算法是空间剖分法和层次包围盒^[2]. 空间剖分法适用于物体在空间均匀分布的稀疏环境, 简单地从大量物体中排除不相交的物体^[3]. 但对于物体分布密集的场景, 则需要进行更深的递归分割, 空间存储量大且灵活性差. 经典的算法主要有: BSP 树, 八叉树以及均

① 基金项目: 国家重点研发计划项目 (2016YFC0801801); 国家自然科学基金 (41430318); 北京市自然科学基金 (4142015)

收稿时间: 2017-02-11; 修改时间: 2017-02-23; 采用时间: 2017-03-02

匀网格剖分法^[4]. 层次包围盒算法是用简单的包围盒来包裹整个物体及物体所有的基本单元子集, 构造起层次树, 通过遍历两物体的层次包围盒来进行逐步求精相交检测^[5]. 比较常用的有包围球 (Bounding sphere), 方向包围盒 (Axis-aligned bounding box, AABB), OBB 包围盒 (Oriented bounding box, OBB) 等^[6]. 而对于场景物体数目较多的情形, 若仅使用层次包围盒算法往往会产生不必要的检测, 影响整体检测效率. 针对简单的多物体场景, 孙劲光^[7]、王巍^[8]、Zhao W^[9]等人提出了结合空间剖分及层次包围盒的混合算法, 快速排除不相交物体对, 有效地提高检测效率, 但对存在静态大尺寸物体的复杂场景及物体的定位精确性上具有一定局限性, 从而影响了排除效率.

因此, 本文针对复杂场景中物体数目众多、尺寸不一、形状不规则的特性, 在方向包围盒 (Oriented bounding box, OBB) 层次树^[10]的基础上提出了一种基于拓扑空间网格的碰撞检测算法 (Topological spatial grid-OBB, TSGO). 通过分割大型物体实现更精确的定位, 利用空间网格单元与物体之间的拓扑结构以及投影相交测试排除明显不相交物体, 提升检测效率, 最终计算得到碰撞物体对的精确碰撞点, 可为碰撞响应服务.

1 算法流程

相较于包围球及轴向包围盒, OBB 包围盒能够对形态复杂的物体模型达到较好的贴合度, 一方面能够更加精确的表征物体模型, 增强不相交物体对的排除效果, 另一方面在层次树的构造上减少树的深度, 简化模型二叉树结构. 因此, 对于本文研究场景下形态复杂、规模庞大的物体模型, 选用 OBB 层次包围盒具有更好的适用性. 算法整体流程如图 1 所示.

算法共分为场景预处理以及实时检测两大部分, 图 2(a) 为整体场景示意图, 图 2(b) 为局部场景示意图, 以二维场景均匀四叉树划分进行说明, 三维场景均匀八叉树划分同理.

(1) 场景预处理阶段. 首先建立所有物体 OBB 层次包围盒, 进行均匀八叉树场景空间划分, 随后对场景中静态大尺寸物体进行分割, 将大尺寸物体分解为多个尺寸较小的子物体进行替代, 最终建立网格与物体之间的拓扑结构. 拓扑建立过程中对物体进行初次定

位, 利用物体根节点包围盒 4 个体对角线进行定位, 提升了定位准确性, 大尺寸物体分解为多个尺寸较小的子物体再定位, 解决大尺寸物体定位准确性问题. 精确的定位有助于在实时检测阶段有效地排除不相交物体对, 减少实时检测耗时.

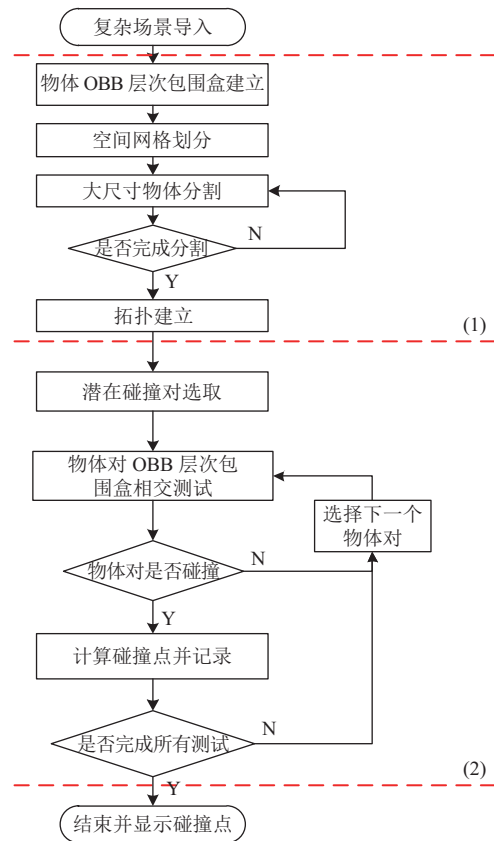


图 1 算法流程图

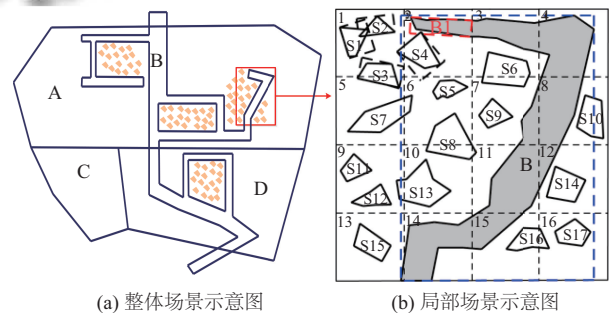


图 2 场景及网格划分示意图

算法对物体建立 OBB 层次包围盒, 将图 2(b) 场景划分成 16 个均匀的小正方形空间网格. 由于大尺寸物体 B 的根节点包围盒占据网格 1~16, 而实际物体并不占据网格 5、6、9、10、16, 为提升网格定位准确度,

将 B 分割为多个子物体, 图中 B_1 为 B 分割后的小尺寸物体集之一. 最后, 对所有物体建立网格的拓扑结构, 完成物体初次定位.

(2) 实时检测阶段. 利用物体与网格的拓扑关系排除不在同一网格内的物体对, 随后以投影相交测试排除同一网格内投影不相交物体对, 最终对剩余的潜在碰撞对利用层次包围盒进行精确测试获取碰撞物体对并计算碰撞点.

取图 2(b) 中物体 S_1 为例, 物体 S_1 、 S_2 、 S_3 、 S_4 的包围盒都占据网格 1, 因此 S_1 仅与 S_2 、 S_3 、 S_4 进行投影相交测试. 经投影相交测试之后排除物体 S_4 , 得到潜在碰撞对 (S_1, S_2) 、 (S_1, S_3) . 对潜在碰撞对物体对利用 OBB 层次包围盒相交测试算法进行精确检测, 排除物体 S_3 , 确定最终发生碰撞的物体对为 (S_1, S_2) , 得到 (S_1, S_2) 相交的三角形对. 最终利用三角形相交算法计算交点作为碰撞点.

2 场景预处理

2.1 OBB 层次包围盒建立

OBB 包围盒因其方向的任意性, 对形态复杂, 不规则的物体具有良好的包围效果. 为改善物体模型三角形面片以及各顶点的非均匀分布导致计算所得包围盒中心点以及方向矩阵的偏差, 采用三角形面积加权方式减少偏差^[11], OBB 层次包围盒建立方法如下:

设模型共有 n 个三角形. 第 i 个三角形的中心点 O^i , 顶点为 p^i 、 q^i 、 r^i , 面积 S^i . 构成模型的所有三角面片的总面积 $SumS$.

$$O^i = \frac{1}{3}(p^i + q^i + r^i) \quad (1)$$

$$S^i = \frac{1}{2} |(p^i - q^i) \times (p^i - r^i)| \quad (2)$$

$$SumS = \sum_{i=1}^n S^i \quad (3)$$

加权后物体包围盒中心点 u , 协方差矩阵 C 如下所示^[10-12]:

$$u = \frac{1}{SumS} \sum_{i=1}^n O^i S^i \quad (4)$$

$$C_{jk} = \sum_{i=1}^n \frac{S^i}{12SumS} (9O_j^i O_k^i + p_j^i p_k^i + q_j^i q_k^i + r_j^i r_k^i) - u_j u_k \quad (5)$$

j 、 k 代表点的 x 、 y 、 z 分量并确定矩阵 C 元素.

矩阵 C 为实对称矩阵, 三个特征向量相互正交, 作为包围盒的方向基底. 物体所有顶点投影到三个方向轴上, 得到三个方向上的最大最小值, 从而确定包围盒的尺寸.

层次树的构造采取自顶向下的二叉树构造方式. 以整个物体模型对象为根节点, 递归划分模型对象并建立对应的 OBB 包围盒, 直到盒子中的元素仅有一个模型几何单元 (三角形) 时终止分裂.

2.2 空间网格划分

进行虚拟三维场景均匀八叉树网格划分时, 需要确定网格单元的尺寸大小. 本文研究场景如图 2(a) 所示, 其中 A、B、C、D 为静态大尺寸物体, 存在多个动态小尺寸物体呈区域性非均匀分布, 因此网格划分过大或过小都会对碰撞检测的效率产生不好的影响^[13]. 此类场景可适用于矿井灾害模拟仿真、山体隧道坍塌模拟等. 针对此类场景, 空间网格划分算法如下:

(1) 设总场景空间为立方体, 利用场景内所有模型顶点集获取 x 、 y 、 z 三个轴向上的极值点, 计算三个轴向极值点差值, 取差值结果中最大值作为总场景空间边长 L , 建立起立方体为总场景空间网格;

(2) 对空间网格进行八叉树网格划分, 将当前空间网格划分成八个尺寸相同的子空间网格;

(3) 若当前划分深度小于 k , 对当前场景内所有空间网格执行步骤 (2), 直至划分深度为 k 时划分结束.

划分深度 k 值确定如下:

设最大动态小物体尺寸 $MaxSmallL$, 网格单元的边长 $GridL$. 则网格单元总数为 8^k , $GridL=L/2^k$, k 的取值最大满足 $L/2^{k_{max}+1} < MaxSmallL < L/2^{k_{max}}$. 针对图 2(a) 所示场景分别取不同数目的动态小物体, 以不同划分深度 $k(0 \leq k \leq k_{max})$ 执行碰撞检测, 检测耗时对比如图 3. 由图 3 可知, 对该类场景划分深度取 2 至 4 时较为合适.

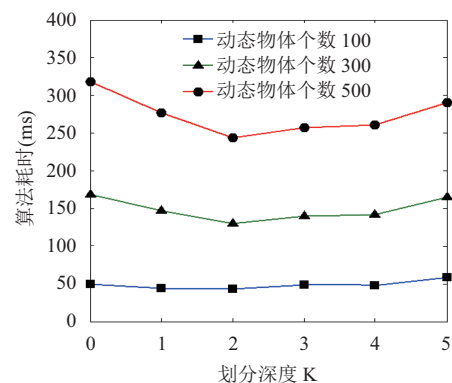


图 3 划分深度与检测耗时关系

2.3 大尺寸物体分割

复杂虚拟场景中存在静态大尺寸物体,所包含的三角形单元数目庞大,导致其层次包围盒根节点包围盒占据多余空间网格.为消除冗余网格,提升定位准确性,对其按一定条件进行分割.以图2(b)中大尺寸物体B为例,分割过程如下:

(1) 以物体B的层次包围盒根节点为当前节点;

(2) 判断当前节点包围盒边长是否满足:当前节点的包围盒边长<网格单元边长.若不满足则将其孩子节点作为新的当前节点,重复此步骤;若满足或当前节点为叶子节点,则转到(3);

(3) 抽离当前节点作为分割后的小物体,该小物体包含当前节点的所有三角形集合.

分割形成的子物体如果尺寸过小会导致单一网格内子物体数目众多,增加后续潜在碰撞对选取的耗时.因此步骤(2)的终止条件保证了分割后包围盒尺寸与网格单元大致相同,既能实现较精准的定位又控制了子物体数目.分割的实质是将大尺寸物体的层次树分解为若干个符合要求的子树,每个子树形成的子物体为大物体的一部分.

2.4 拓扑结构建立

网格单元结构体中包含网格单元ID号 $Grid_ID$; 网格顶点的最小点与最大点 min, max ; 邻域网格单元ID集合 $neighbor_ID$; 网格内包含物体的身份索引 $Including_Object$. 其描述及具体建立过程如下:

2.4.1 网格单元ID

场景划分后网格单元总数为 8^k 个, k 为八叉树划分迭代次数,因此划分好的空间网格在 x, y, z 单一轴向上的网格数为 $m=2^k$, 用 a, b, c 分别表示 x, y, z 方向的序号, 则 (a, b, c) 可以唯一地表示网格单元的序号:

$$Grid_ID = a + b \cdot m + c \cdot m^2$$

2.4.2 最值点

网格单元6个顶点中在 x, y, z 三个方向皆为最小值的点为 min , 皆为最大值的点为 max .

2.4.3 邻域网格单元

记录当前网格单元相邻网格单元ID号及自身ID. 一个网格单元的 $neighbor_ID$ 至多27个, 计算方式如下:

如图4中网格单元6上 x, y, z 坐标轴所示, 其相邻网格可利用三个轴向以及任意两个轴向或三个轴向的组合进行计算:

(1) 第 i 个网格单元的 $neighbor_ID$ 分别为 $i, i\pm 1, i\pm m, i\pm m^2, i\pm 1\pm m, i\pm 1\pm m^2, i\pm m\pm m^2, i\pm 1\pm m\pm m^2$;

(2) 上述27个结果值若小于0或大于网格总数 8^k , 则此相邻单元序号无效, 舍去该序号;

(3) 计算剩余邻域网格单元与当前网格单元的距离. 若距离大于网格单元体对角线长度 $|max-min|$ 则舍去此相邻单元序号.

2.4.4 网格所含物体集身份索引

网格单元包含的物体集身份索引为该网格内所有物体的身份信息集合, 利用该索引可确定网格内包含的所有物体. 要确定某一网格是否包含该物体, 需要先确定物体所占据的网格. 文献[7]中所采用的方法是: 已知物体包围盒主对角线上两顶点, 计算两顶点所在的网格 $Grid_ID_1$ 和 $Grid_ID_2$, 则物体所占据的空间网格为 $Grid_ID_1$ 到 $Grid_ID_2$ 之间所有的连续网格. 如图4所示, P_1P_8 为某物体包围盒的主对角线. P_1 在网格1中, P_8 在网格5中, 则利用上述算法得到物体所占空间网格为1、2、3、4、5, 而实际只占据网格1和网格5, 精确性较差.

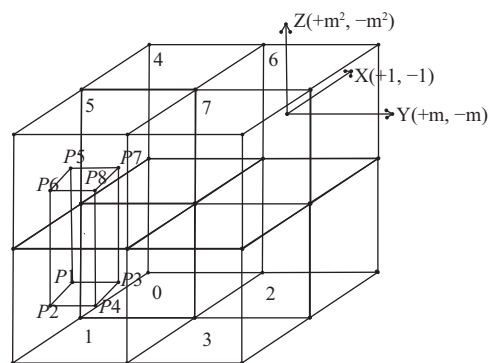


图4 邻域网格及物体网格定位示意图

为解决上述方法缺陷, 本文提出一种基于包围盒体对角线来定位物体所占空间网格的方法, 具体如下:

(1) 以4个体对角线 ($P_1P_8, P_2P_7, P_3P_6, P_4P_5$) 代替整体包围盒; 选择一个对角线, 如 P_1P_8 , 转到(2).

(2) 寻找该对角线所占网格. 以向量形式 $P=E+tD$ 表示对角线, E 为线段的端点 (如 P_1P_8 的端点为 P_1), t 为参数, D 为从 E 出发的对角线方向向量, 模值为对角线长度. 对某网格最值点 min, max 解不等式:

$$\begin{cases} x_{min} \leq x_E + txD \leq x_{max} \\ y_{min} \leq y_E + tYD \leq y_{max} \\ z_{min} \leq z_E + tZD \leq z_{max} \end{cases} \quad (6)$$

若 t 有解且 $0 \leq t_{min} \leq t_{max} \leq 1$, 则此条对角线占据该网格, 转到步骤 (3).

(3) 判断剩余 3 个体对角线是否占据 (2) 中所占网格的邻域网格, 最终记录 4 个体对角线占据的所有非

重复网格单元为物体所占据的网格.

得到物体占据的网格单元后将其对应的物体身份索引信息记录在 *Including_Object* 中, 构造出物体与网格之间的对应关系, 形成最终拓扑结构. 拓扑结构如图 5 所示.

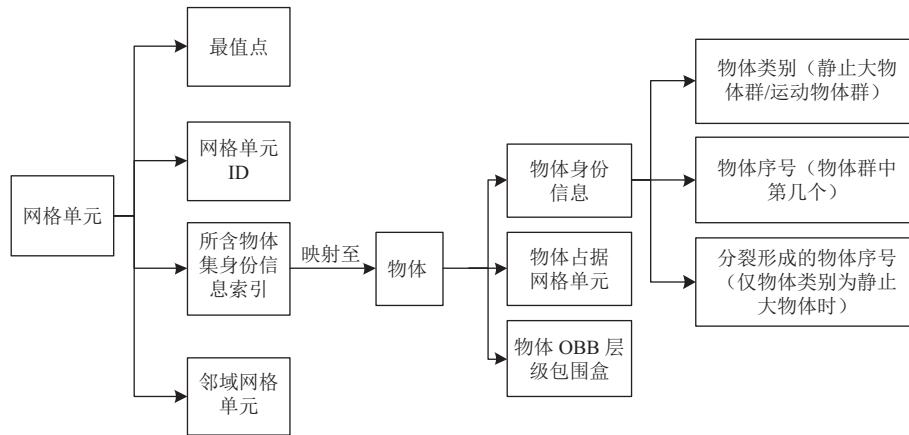


图 5 网格单元拓扑结构

3 实时检测

当场景中大量物体时, 若仅用两两物体配对检测, 无法达到高效的检测效率^[14]. 因此, 本文算法利用拓扑空间网格以及投影相交测试剔除场景中明显不相交物体, 再对剩余潜在碰撞对物体进行 OBB 层次包围盒相交测试, 以提升检测过程实时效率.

3.1 潜在碰撞对选取

算法利用网格拓扑结构中的所含物体集身份信息 *Including_Object* 进行排除. 确立某物体占据的网格, 仅对该物体和该物体占据网格内包含的所有物体配对进行投影相交测试.

投影相交测试利用包围盒的各坐标轴投影来排除明显不相交的物体. 具体内容为: 两物体层次包围盒的根节点包围盒向 x 、 y 、 z 三个坐标轴中的每一条坐标轴做投影, 计算在该坐标轴下其包围盒的投影区间是否有重叠. 若 x 、 y 、 z 三条轴上存在任意一条轴上的投影区间不相交, 则证明该物体对一定不相交, 可排除该物体对. 将剩余物体对作为潜在碰撞对存入潜在碰撞对集合.

潜在碰撞对选取算法伪代码如下:

输入: 运动物体集 MO ; 大物体分裂所得物体集 BO ;

输出: 潜在碰撞对集合 PCP ;

```
void PotentialPairsPickUp( $MO, BO, PCP$ )
```

```
{
```

```
for (int i = 0; i < MO.size(); i++){
```

```
     $IO \leftarrow MO[i]$ 所占网格中包含的所有物体集合;
```

```
    for (int j = 0; j < IO.size(); j++){
```

```
        if ( $IO[j] \in BO$ )
```

$MO[i]$ 与 $IO[j]$ 执行投影相交测试, 若相交将两者存入 PCP ;

```
        if ( $IO[j] \in MO$  且  $IO[j]$  的物体序号大  $MO[i]$ )
```

$MO[i]$ 与 $IO[j]$ 执行投影相交测试, 若相交将两者存入 PCP ;

```
    }
```

```
}
```

3.2 层次包围盒相交测试

如图 6 所示, 潜在碰撞对共三种类型, 而真实发生相交的仅为图 (c) 所示类型, 因此需要通过遍历两物体 OBB 层次包围盒树进行逐步求精的相交检测, 过程如下^[15].

利用某潜在碰撞对中 $Object_1$ 的 OBB 层次包围盒根节点去遍历 $Object_2$ 的 OBB 层次包围盒, 若能达到 $Object_2$ 的叶子节点, 再利用该叶子节点去遍历 $Object_1$ 的层次包围盒, 若也能达到 $Object_1$ 的叶子节点, 则 $Object_1$ 与 $Object_2$ 发生碰撞, 两叶子节点中包含的三角形相交. 如果在遍历过程中, 当前两节点的包围盒不相交, 则它们包含的对象的三角形子集必定不相

交,也就不需要对子集中的元素进行进一步的遍历检测.其中,两节点包围盒之间的相交测试利用的是分离轴理论^[16].一对OBB包围盒,其分离轴有15个:两盒子共6个面的法向量,盒子三条边向量与另一盒子三条边向量的两两叉积,共9个.当15条分离轴都无法分离时,两包围盒之间相交.

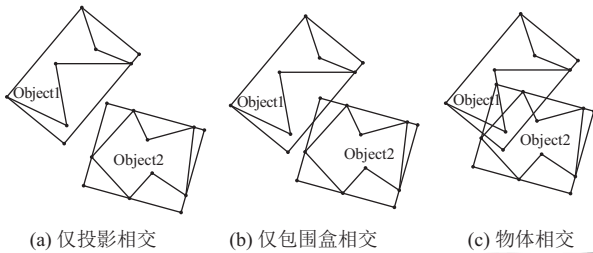


图6 潜在碰撞对类型图

层次包围盒相交测试利用二叉树遍历避免了对执行该阶段的两物体的所有几何基元(三角形)进行穷举碰撞测试.设单个物体三角形个数 m ,则层次包围盒遍历检测的时间复杂度为 $O(mlbm)$,达到高效的精确检测效率.

3.3 相交点计算

通过层次包围盒相交测试得到最终发生相交的物体及三角形后进行交点计算并作为碰撞点.三维空间内两三角形的空间分布分为异面与共面两种,针对两

种情况的相交类型如下.

(1) 异面相交

图7中(a)、(b)、(c)显示了异面三角形的三种相交类型及交点.交点利用射线-三角形相交法^[17]计算:

在三角形 (V_0, V_1, V_2) 上的点 T 可以用如下方式表示:
 $T(u, v) = (1 - u - v) \cdot V_0 + u \cdot V_1 + v \cdot V_2 (u + v \leq 1, u \geq 0, v \geq 0)$ (7)

射线使用 $R(t) = O + tD$ 来表示, O 为原点, D 为射线方向,为三角形线段起始点指向线段终点,模值为线段长度.可利用下式求交点:

$$O + tD = (1 - u - v) \cdot V_0 + u \cdot V_1 + v \cdot V_2 \quad (8)$$

解得:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{[(D \times E_2) \cdot E_1]} \begin{bmatrix} (E_3 \times E_1) \cdot E_2 \\ (D \times E_2) \cdot E_3 \\ (E_3 \times E_1) \cdot D \end{bmatrix} \quad (9)$$

其中 $E_1 = V_1 - V_0, E_2 = V_2 - V_0, E_3 = O - V_0$.交点为 $O + tD$.若 t 不在 $[0, 1]$ 区间内,则交点超出线段长度,认为无交点;若 $(D \times E_2) \cdot E_1 = 0$,如图7中(c)所示,三角形一边与另一三角形重合,也认为线段与平面无交点.

(2) 共面相交

图7中(d)、(e)、(f)、(g)显示了共面三角形相交的四种类型以及交点.首先判断一个三角形的顶点是否在另一三角形内,若在内部则记录该点.然后利用线段相交求交点,该类型交点计算分解为线段的相交判断.

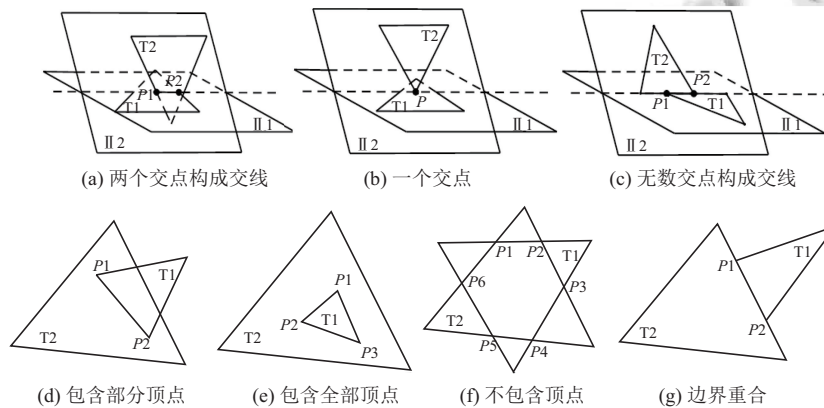


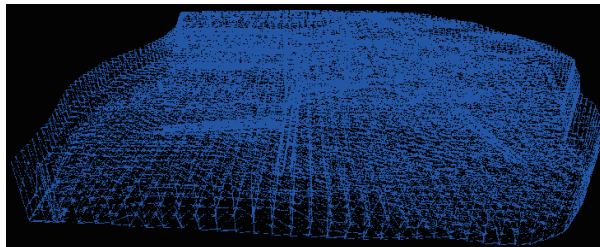
图7 异面/共面三角形相交类型

线段上的点用 $P = E + tD$ 表示, E 为线段的端点, D 为 E 点出发的方向向量,其值为线段长度, t 为参数,则线段相交有: $E_1 + t_1D_1 = E_2 + t_2D_2$.解得若 $t_1, t_2 \in [0, 1]$ 则线段相交且交点为 $E_1 + t_1D_1$.若发生线段重合,该式无解,也判定线段不相交.

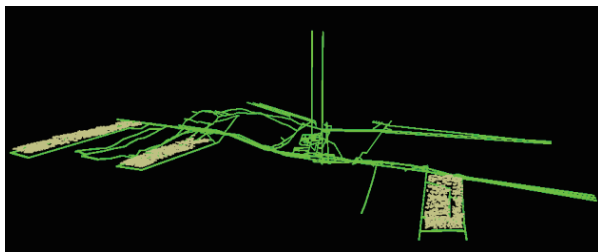
4 实验分析

为测试本文算法在复杂虚拟场景中的碰撞检测效率,以河北省张家口某矿区模型为虚拟地矿实验场景.场景中有6个复杂的静态大尺寸模型:包含5个地层模型及1个巷道模型,三角面片数目共93079个;采掘

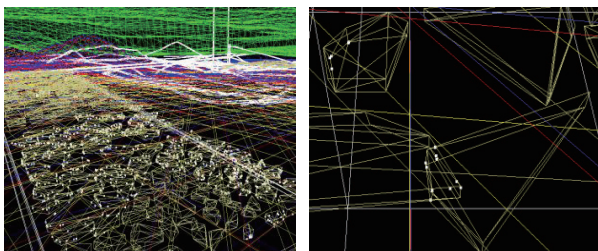
区域内存在大量不规则运动岩体碎块,进行碎块与碎块、碎块与巷道及碎块与地层之间的碰撞检测.本文算法在 CPU 主频为 3.2GHz,内存 4G 的 PC 机上进行实验并用 OpenGL 进行显示.图 8(a)为整体虚拟场景,图 8(b)为场景中巷道及碎块模型.



(a) 整体场景图



(b) 巷道及采掘区碎块模型



(c) 碰撞点效果图

(d) 碰撞点局部放大图

图 8 实验场景与碰撞点结果

实验分别设定 50、100、200、300、400、500、600 个运动岩体碎块进行检测,空间划分深度取 2.针对上述场景,分别选用基于 OBB 层次包围盒算法的经典碰撞检测算法 RAPID^[10]、基于空间剖分与层次包围盒的混合算法^[7-9]以及本文提出的 TSGO 算法执行 100 次测试实验.三种算法执行碰撞检测并得出碰撞点的平均耗时如表 1 所示,图 8(c)、8(d)为最终的碰撞点.

通过图 9 及表 1 中的实验结果可以看出,随着运动物体数目增多,相较于 RAPID 算法,混合算法与 TSGO 算法利用空间网格排除明显不相交物体使总体检测耗时减小;相较于混合算法,TSGO 算法利用空间网格拓扑、大物体分割及投影相交测试使得算法耗时进一步减小,且随着碎块数目的增长,TSGO 算法的检

测效率优势越为明显.

表 1 实验结果对比

碎块数 目	场景三角面片 数	碰撞点数 目	算法耗时(ms)		
			RAPID算 法	混合算法	TSGO算法
50	94847	640	57.06	56.94	55.88
100	95723	808	82.99	82.10	74.85
200	97427	1230	165.53	164.70	157.23
300	99159	1364	193.64	192.20	167.95
400	100859	1463	231.27	226.17	186.49
500	102615	1510	278.26	262.68	204.06
600	104319	1716	358.42	332.30	247.08

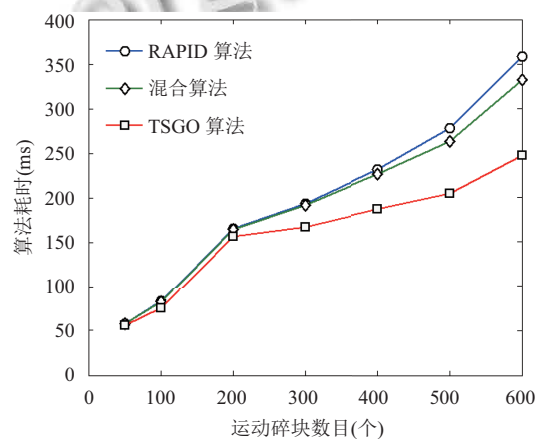


图 9 算法耗时对比图

5 结束语

针对复杂虚拟场景物体数目多、形状复杂、尺寸不一的特性,本文提出一种基于拓扑空间网络的碰撞检测算法.算法以大尺寸物体分割及包围盒体对角线替代有效改善了物体定位准确性,利用均匀八叉树拓扑网格及投影相交测试排除不相交物体对,提升了检测效率,最终分析了空间三角形的相交类型并通过计算得到精确的碰撞点.实验结果表明:相比较 RAPID 算法及单纯的空间剖分与层次包围盒的混合算法,本文算法可有效地减少检测耗时,对存在大尺寸物体的复杂场景中多物体的碰撞检测有良好的适用性和检测效率.所得碰撞点可为碰撞响应提供可靠支持.

参考文献

- 刘秀玲,王冬雨,陈栋,等.复杂场景中快速碰撞检测算法及 GPU 加速.计算机工程与设计,2012,33(5): 1847-1851.
- 沈学利,吴琼.基于包围盒和空间分割的混合碰撞检测算法.计算机工程,2012,38(6): 256-258.
- 冯立颖.碰撞检测技术研究综述.计算机时代,2014,(8):

- 7-10.
- 4 Ericson C. 实时碰撞检测算法技术. 刘天慧, 译. 北京: 清华大学出版社, 2010.
- 5 Chang JW, Wang WP, Kim MS. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer-Aided Design*, 2010, 42(1): 50-57. [doi: [10.1016/j.cad.2009.04.010](https://doi.org/10.1016/j.cad.2009.04.010)]
- 6 潘仁宇, 孙长乐, 熊伟, 等. 虚拟装配环境中碰撞检测算法的研究综述与展望. *计算机科学*, 2016, 43(11A): 136-139.
- 7 孙劲光, 吴素红. 基于空间分割与椭球包围盒的碰撞检测算法. *计算机工程与应用*, 2016, 52(4): 217-222.
- 8 王崑, 周诚, 杨云, 等. 面向虚拟维修的碰撞检测算法. *计算机应用与软件*, 2016, 33(4): 235-238.
- 9 Zhao W, Ye LM. A collision detection algorithm based on spatial partitioning and bounding volume. *Applied Mechanics and Materials*, 2013, 433-435: 932-935. [doi: [10.4028/www.scientific.net/AMM.433-435](https://doi.org/10.4028/www.scientific.net/AMM.433-435)]
- 10 Gottschalk S, Lin MC, Manocha D. OBBTree: A hierarchical structure for rapid interference detection. *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA, 1996. 171-180.
- 11 李苗. 实时碰撞检测算法分析与比较. *计算机与现代化*, 2011, (6): 88-90.
- 12 田瑜, 关正西, 许平, 等. 3D 关节角色基于 OBB 的实时动态碰撞检测. *计算机工程与应用*, 2006, 42(33): 91-93. [doi: [10.3321/j.issn:1002-8331.2006.33.029](https://doi.org/10.3321/j.issn:1002-8331.2006.33.029)]
- 13 鲍义东, 吴冬梅. 自适应细分及优化编码八叉树碰撞检测算法. *上海交通大学学报*, 2015, 49(8): 1114-1122.
- 14 王磊, 王毅刚. 基于 GPU 加速的多物体碰撞检测方法. *计算机工程与科学*, 2009, 31(12): 52-55. [doi: [10.3969/j.issn.1007-130X.2009.12.015](https://doi.org/10.3969/j.issn.1007-130X.2009.12.015)]
- 15 彭晏飞, 卢真真. 基于空间剖分和包围盒的快速碰撞检测算法. *计算机应用与软件*, 2015, 32(8): 150-153, 165.
- 16 孔中武, 石林锁, 王涛. 导弹飞行视景仿真中的碰撞检测算法. *计算机系统应用*, 2015, 24(1): 176-179.
- 17 Havel J, Herout A. Yet faster ray-triangle intersection (using SSE4). *IEEE Trans. on Visualization and Computer Graphics*, 2010, 16(3): 434-438. [doi: [10.1109/TVCG.2009.73](https://doi.org/10.1109/TVCG.2009.73)]