

# 基于 AABB 树的聚变堆形变部件碰撞检测算法<sup>①</sup>

邓峻生<sup>1</sup>, 毛世峰<sup>1</sup>, 刘旭峰<sup>2</sup>, 叶民友<sup>1,2</sup>

<sup>1</sup>(中国科学技术大学 物理学院, 合肥 230027)

<sup>2</sup>(中国科学院 等离子体物理研究所, 合肥 230031)

通讯作者: 毛世峰, E-mail: [sfmao@ustc.edu.cn](mailto:sfmao@ustc.edu.cn)

**摘要:** 针对工程设计中形变部件的有限元模型的碰撞检测问题, 提出了一种基于 AABB 树的快速碰撞检测算法. 对于需要分析的有限元, 对几何表面进行三角化, 随后建立 AABB 包围盒, 并采用优化的 AABB 树算法进行空间划分; 利用 AABB 树与包围盒排除不相交图形, 采用 Devillers 算法测试三角形相交, 并利用并行方式加快计算. 实验结果表明, 本算法有效提高了碰撞检测的效率, 适用于复杂有限元模型的碰撞检测.

**关键词:** 碰撞检测; AABB 树; 形变部件; 有限元; 工程设计; 聚变堆

引用格式: 邓峻生, 毛世峰, 刘旭峰, 叶民友. 基于 AABB 树的聚变堆形变部件碰撞检测算法. 计算机系统应用, 2018, 27(11): 161-167. <http://www.c-s-a.org.cn/1003-3254/6616.html>

## Collision Detecting Algorithm for Deformable Components in Fusion Reactor Based on AABB Tree

DENG Jun-Sheng<sup>1</sup>, MAO Shi-Feng<sup>1</sup>, LIU Xu-Feng<sup>2</sup>, YE Min-You<sup>1,2</sup>

<sup>1</sup>(School of Physical Sciences, University of Science and Technology of China, Hefei 230027, China)

<sup>2</sup>(Institute of Plasma Physics, Chinese Academy of Sciences, Hefei 230031, China)

**Abstract:** To perform fast collision detection on the Finite Element Model (FEM) of deformable components in engineering design, an algorithm of collision detection based on the Axis-Aligned Bounding Box (AABB) tree is developed. For the FEMs to be checked, the facial surfaces are triangulated at first. AABBs are then generated for the triangles, and optimized AABB trees are built to divide the space. The AABBs and AABB tree are used to exclude disjoint graphics, and the Devillers & Guigue algorithm are applied to perform fast triangle intersection test. Parallel computing is also applied to accelerate the calculation. This proposed algorithm is applied for testing models, and the result shows that the efficient collision detection algorithm could give reliable result for complex FEMs.

**Key words:** collision detection; AABB tree; deformable components; Finite Element Model (FEM); engineering design; fusion reactor

中国聚变工程实验堆 (CFETR) 是正在设计中的超导聚变实验堆, 旨在弥补聚变实验堆 ITER 和示范反应堆 DEMO 之间的空白<sup>[1-3]</sup>. 为了支持 CFETR 设计工作高效协同的展开, CFETR 集成设计平台<sup>[4,5]</sup>目前正在开发中.

在工程设计中, 由于外部环境等因素影响, 部件会产生应力、导致形变, 影响零部件的装配设计与空间

布局<sup>[6]</sup>. 因此对形变部件的碰撞检测, 是工程设计中的必不可少的检测环节. 工程设计中通常利用有限元方法分析计算部件的应力、形变, 计算结果以有限元网格的形式存储. 目前对于刚体模型的碰撞检测算法研究比较充分, 针对复杂场景的快速检测算法也正在不断发展<sup>[7]</sup>. 对有限元结果的碰撞检测, 由于模型的复杂

① 基金项目: 国家磁约束核聚变能发展研究专项 (2014GB110002)

Foundation item: National Magnetic Confinement Fusion Development Research Project of China (2014GB110002)

收稿时间: 2018-03-21; 修改时间: 2018-04-24; 采用时间: 2018-05-03; csa 在线出版时间: 2018-10-24

度较高,需要采取多种措施降低计算复杂度。

对形变的有限元模型的碰撞检测问题, Sean Curtis 等提出了基于特征三角形的快速检测方法<sup>[8]</sup>, 吴峥等提出了三角片与包围盒混合的快速碰撞检测算法<sup>[9]</sup>。这些算法主要针对三角片网格与四面体网格, 不适合工程设计中复杂多样的有限元网格, 且在三角形相交、包围盒处理、并行化等方面存在不足, 计算的效率存在提升空间。

本文提出了一种基于三角形碰撞检测与轴对齐包围盒 (Axis-Aligned Bounding Box, AABB) 的、针对有限元模型的通用快速碰撞检测算法: 将有限元的几何表面三角化后, 选择合适的三角形相交算法, 利用 AABB 包围盒紧密性好、相交测试简单的优势并采用优化后的 AABB 树算法, 结合并行化的设计, 实现对有限元模型的快速碰撞检测。利用本算法可以处理多种有限元网格, 对工程设计中复杂的形变有限元模型实现高效快速的碰撞检测。本文将该算法应用于聚变堆的工程设计中, 能够有效提高碰撞检测的效率, 并给出清晰直观的检测结果。

## 1 碰撞检测算法

有限元模型由大规模的网格和网格节点组成, 网格类型包括四面体、六面体等几何体。在有限元模型的碰撞检测中, 通过提取几何表面、将表面的几何面用三角形重构, 可以将有限元模型的碰撞问题, 转换为表面三角片的碰撞检测问题; 同时利用包围盒算法简化碰撞计算、引入 AABB 树算法提高碰撞检测效率<sup>[10]</sup>, 最终实现部件有限元模型的快速碰撞检测。算法的处理步骤如图 1 所示。

核心步骤如下所述:

Step 1. 读取有限元计算结果, 提取模型的表面节点, 用三角片重构, 以三角形网格描述几何体的轮廓。

Step 2. 计算三角片对应的 AABB 包围盒, 将包围盒与三角形对应。

Step 3. 对包围盒集, 构造 AABB 树, 将所有包围盒存储在二叉树中。

Step 4. 对两棵 AABB 树中的包围盒进行碰撞检测。如果包围盒间存在碰撞, 则继续检测包围盒对应的三角片的碰撞问题。

Step 5. 跳过未碰撞的包围盒、三角片, 记录所有碰撞的三角片对, 构成碰撞结果集。

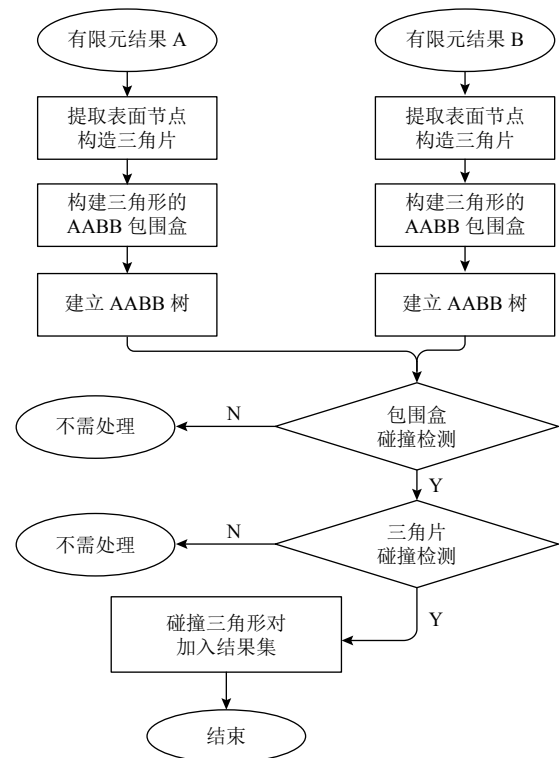


图 1 碰撞检测算法计算流程

Step 6. 结束计算, 将碰撞区域存储为 STL (stereolithography, 立体光刻) 格式的三角片模型文件。

### 1.1 包围盒算法

包围盒算法在数字化模型的碰撞检测计算中有着广泛的应用<sup>[11,12]</sup>, 其核心思路为: 将复杂几何体用一个简单几何体包围, 如立方体、球体, 然后计算简单几何体的碰撞, 排除不碰撞对象, 最后计算被简单几何体包含的几何体间关系。

针对本文静态检测的场景, 选择构造 AABB 包围盒进行计算。在读取三角片网格后, 首先对所有三角片构造 AABB 包围盒, 即用一个与  $x$ 、 $y$ 、 $z$  三轴平行的立方体包围几何面。如果一对立方体不碰撞, 则立方体内的几何面元不会碰撞。

对一个三角形面元, 其包围盒可以两个点表示: 如式 (1)(2) 所示, 两个点的坐标由所有顶点在  $x$ 、 $y$ 、 $z$  方向上的最大值最小值决定。

$$P_{\max} = x_{\max}\vec{i} + y_{\max}\vec{j} + z_{\max}\vec{k} \quad (1)$$

$$P_{\min} = x_{\min}\vec{i} + y_{\min}\vec{j} + z_{\min}\vec{k} \quad (2)$$

检测两个轴对齐包围盒是否碰撞, 需要将包围盒分别投影到  $x$ 、 $y$ 、 $z$  轴上, 判断其投影是否相交。对于两个包围盒 A、B, 将其投影到  $x$  轴上, 对应关系为:

$$A \rightarrow (x_{a-\min}, x_{a-\max}) \quad (3)$$

$$B \rightarrow (x_{b-\min}, x_{b-\max}) \quad (4)$$

式(3)(4)表示包围盒在  $x$  轴上的投影区间, 两区间不相交的条件为式(5):

$$(x_{a-\min} > x_{b-\max}) \vee (x_{a-\max} < x_{b-\min}) \quad (5)$$

当包围盒在  $x$ 、 $y$ 、 $z$  轴向的投影都相交, 则判定包围盒相交. 本方法只需进行 6 次大小判断、3 次或操作, 即可完成包围盒碰撞检测.

相比直接进行空间三角形的检测, AABB 包围盒排除相交的计算量更小, 因此计算速度更快.

### 1.2 AABB 树算法

#### 1.2.1 层次包围盒与 AABB 树

当两个三角形相距较远时, 采用 AABB 包围盒可以快速排除且降低了求交的计算量, 但是采用 AABB 包围盒算法, 依然需要对 A 的  $m$  个包围盒与 B 的  $n$  个包围盒求交, 没有降低求交计算的次数.

为了降低求交计算的时间复杂度, 引入 AABB 树的结构. AABB 树基于层次包围盒原理, 即用大的包围盒包裹小的层次包围盒, 并用大包围盒进行求交计算; 层次包围盒基于如下原理:

对集合 A、B 与 P, 存在  $A \in P, B \in P$ , 如果  $Q \cap P = \emptyset$ , 则  $Q \cap A = Q \cap B = \emptyset$ . 因此如果大包围盒与某个包围盒不相交, 则其中包含的包围盒也不相交; 如此可以降低求交的次.

AABB 树是一种以二叉树存储的层次包围盒, 树的每个节点都是一个包围盒, 除叶节点外每个包围盒包含两个子盒; 所有叶节点为几何体的包围盒.

AABB 树的层次包围盒结构如图 2 所示, 对应的二叉树存储结构如图 3 所示.

#### 1.2.2 AABB 树的构建

AABB 树的构建流程如图 4, 核心步骤如下文.

- (1) 新建包围盒作为节点, 包含两个子节点的包围盒.
- (2) 新包围盒向树中添加时, 与节点包围盒测试相交.
- (3) 如果不相交则新建包围盒包裹当前节点和新包围盒并作为头结点, 当前节点和新包围盒作为子节点.
- (4) 如果与其相交, 则首先新建大包围盒替换当前节点, 然后新包围盒与两个子节点合并, 选择合并后体积较小的节点, 进行迭代操作.

通过这种方法, 可以建立层次包围盒结构, 并存储在二叉树中.

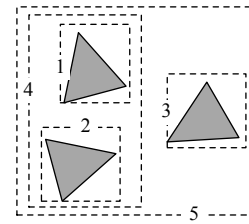


图 2 层次包围盒示意图

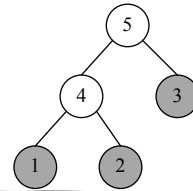


图 3 二叉树结构示意图

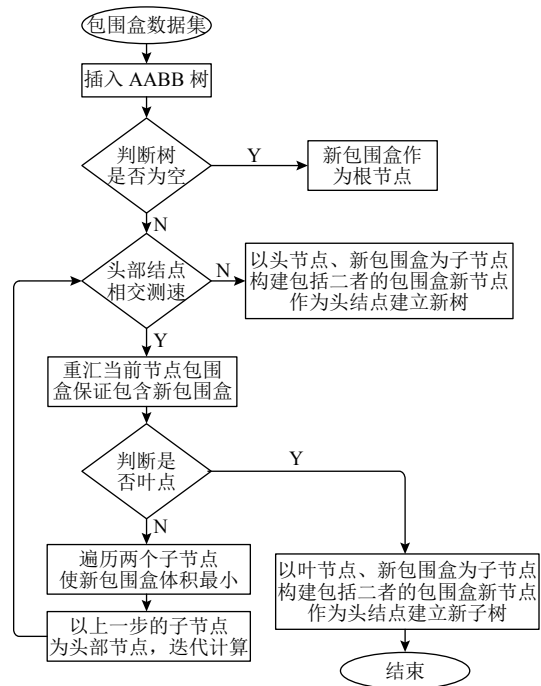


图 4 AABB 树构造流程

#### 1.2.3 AABB 树的碰撞检测

一个包围盒, 与一棵 AABB 树的碰撞检测过程, 从根节点对应的包围盒开始计算:

- (1) 如果与节点不相交, 则该节点的所有子节点都不相交.
- (2) 如果包围盒与一个节点相交, 则继续与节点的两个子节点求交, 进行迭代直到叶节点. 理想情况下, 包围盒每次只与一个子包围盒相交, 相当于二分查找. 对于  $n$  个节点的树, 理想情况下的求交次数为  $\log n$  级别.

因此采用 AABB 树后, 算法复杂度可以降低为  $O(m \log n)$ .

对两棵 AABB 树的计算有所区别: 从两棵树的根节点开始遍历, 如果当前两个节点相交, 则计算二者的子节点包围盒的相交关系; 通过递归, 直到叶节点包围盒为止.

#### 1.2.4 AABB 树的优化

王晓荣等从 AABB 树的平衡角度, 提出了 AABB 树构建过程中的建议<sup>[13]</sup>. 当 AABB 树的左右节点近似于空间二分, 碰撞检测的效率最高; 为了实现这个效果, 引入分割操作:

(1) 选择整个模型在  $x$ 、 $y$ 、 $z$  轴上最长的轴, 计算模型在该方向的最大值、最小值.

(2) 将所有包围盒, 按照其在最长轴上的投影, 在最小值、最大值方向分成两组, 并尽可能保证平衡.

通过这种方法, 直到二分后每一组的三角片数量到达一定程度; 如此建立的二叉树比不分组情况下更加平衡, 计算效率更高.

### 1.3 三角形相交检测

对于三角形的快速相交检测算法研究很多, 典型的如标量判别型的 Möller 算法<sup>[14]</sup>, 矢量判别型的 Devillers & Guigue 算法<sup>[15]</sup> (以下简称 Devillers 算法). 本节将对两种算法进行比较, 选择本计算场景下性能更优的算法.

#### 1.3.1 标量判别型的 Möller 算法

标量判别型算法是通过准确计算来获知两三角形相交关系的一类算法. 算法的基本思路如下所述:

设有两个三角形  $T_1$  和  $T_2$ , 顶点分别为  $V_{10}$ 、 $V_{11}$ 、 $V_{12}$  与  $V_{20}$ 、 $V_{21}$ 、 $V_{22}$ , 三角形所在的平面分别为  $S_1$ 、 $S_2$ , 法向量为  $N_1$ 、 $N_2$ , 则有  $S_1$  的方程表示:

$$N_1 \times (x - V_{10}) = 0 \quad (6)$$

其中,  $x$  为任意一点的坐标.

空间中一点坐标代入方程, 值为 0 则在面上, 大于 0 则在面上方, 小于 0 则在面下方. 据此进行如下计算.

Step 1. 计算  $T_2$  的三个顶点在  $S_1$  平面方程的值.

Step 2. 如果  $T_2$  三点在  $S_1$  平面同侧, 则不相交.

Step 3. 如果  $T_2$  三点在  $S_1$  平面上, 转换为共面三角形的相交计算. 如果一个点在面上, 且其它两点同号, 则计算该点是否在三角形内部; 如果两个点在面上, 计算线段与三角形的关系. 总之, 可以转换为平面内计算.

Step 4. 如果  $T_2$  的三个顶点在  $S_1$  的两侧, 则计算  $T_1$  的三个顶点与  $S_2$  平面的关系.

Step 5. 经过排除计算, 只剩下  $T_1$ 、 $T_2$  在两侧的情况.

如图 4 所示, 设平面的交线为  $k$ , 则两个三角形一定都与  $k$  相交. 直线  $k$  的方程可以表示为:

$$k = K \times i + O \quad (7)$$

其中  $K$  为直线的方向向量,  $i$  为标量化的点坐标.

Möller 算法的核心, 就是计算出三角形与  $k$  的交点的  $i$  值. 设  $T_1$  和  $T_2$  与  $k$  的交点参数值分别为  $i$ 、 $j$  和  $k$ 、 $l$ , 则区间  $(i, j)$  与  $(k, l)$  相交, 等价于三角形相交.

#### 1.3.2 矢量判别型的 Devillers 算法

矢量判别型算法是通过一系列计算值的符号来确定两个三角形的位置关系, 继而判别其相交情况的一类算法. 算法的核心思想如下所述:

设空间中四点坐标分别为  $(a_x, a_y, a_z)$ ,  $(b_x, b_y, b_z)$ ,  $(c_x, c_y, c_z)$ ,  $(d_x, d_y, d_z)$ , 则有行列式:

$$[a, b, c, d] = \begin{pmatrix} a_x & a_y & a_z & 1 \\ b_x & b_y & b_z & 1 \\ c_x & c_y & c_z & 1 \\ d_x & d_y & d_z & 1 \end{pmatrix} \quad (8)$$

行列式的值表示点  $d$  与  $a$ 、 $b$ 、 $c$  组成平面的位置关系, 等于 0 在面上, 大于 0 在面上方, 小于 0 在面的下方. 通过该公式, 对三角形间的位置关系进行判定.

与 Möller 算法的过程类似, 对于三角形在平面一侧的情况可以进行排除, 对于共面或者点、线段等情况可以进行计算; 通过  $T_1$ 、 $T_2$  的最终校验, 最终需要计算剩下  $T_1$ 、 $T_2$  分别在平面两侧的情况.

Devillers 算法不计算具体的交点坐标, 而是计算行列式的值, 进行相交判断. 如图所示得情况下, 循环置换顶点, 保证  $V_{10}(V_{20})$  在面  $S_2(S_1)$  的一面, 同时交换  $V_{11}$ 、 $V_{12}(V_{21}$ 、 $V_{22})$ , 保证  $V_{10}(V_{20})$  在面的上方.

如图 5 所示, 考虑三角形与交线  $k$  的相交区域, 为保证二者存在重叠, 即区间  $(i, j)$  与  $(k, l)$  相交, 只需  $k \leq j$  且  $i \leq l$ , 用判别式表示即:

$$[V_{10}, V_{11}, V_{20}, V_{21}] \leq 0 \cap [V_{10}, V_{12}, V_{22}, V_{20}] \leq 0 \quad (9)$$

#### 1.3.3 算法比较

矢量型的 Devillers 算法, 与标量型的 Möller 算法相比, 存在性能优势:

(1) 内存使用上, Möller 算法使用 55 个变量, 而 Devillers 算法只需 18 个变量.

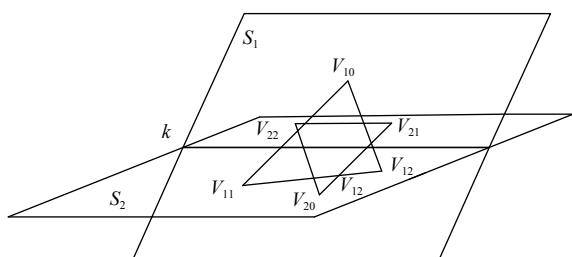


图5 三角形相交示意图

(2) 在计算点与面的关系时, 面方程方法与行列式方法计算量相同; 两种算法的区别在处理三角形在两侧的场景下, Möller 算法要求解四个坐标的值, 而 Devillers 算法只判断两个行列式大小, 后者计算量更小.

(3) 排除不相交的三角形时, Möller 算法进行三次排除, 而 Devillers 算法进行四次排除; 前两次排除的计算相同, Devillers 算法在后面的排除中, 通过行列式的正负号排除, 比 Möller 算法效率更高.

邹益胜等对两种算法进行了性能测试<sup>[16]</sup>, 用随机生成的三角形进行计算, 结果表明 Devillers 算法的计算效率比 Möller 算法高约 15%. 因此本文选择 Devillers 算法, 对三角形进行相交检测.

#### 1.4 并行优化

实际应用场景中, 复杂模型的三角片网格一般为十万、百万量级, 为了充分利用计算资源, 加快计算速度, 本算法采用多线程的方式, 提高碰撞检测的计算效率.

在文件读取、包围盒构建、AABB 树构建过程中, 处理两个模型, 可以利用 2 个线程并行完成.

在碰撞检测的过程中, 可以将待检测的模型拆分, 利用多线程方法加快计算速度. 设有两棵 AABB 树 A 与 B, 并行的基本思路如下所述:

(1) 根据计算机核心数确定线程数, 一般为  $2^n$ .

(2) 将二叉树 A, 从头结点开始迭代, 按左右节点拆分为  $2^n$  棵子树.

(3) 在每个线程中, 每棵子树与二叉树 B 进行碰撞检测; 全部计算结束后合并结果.

并行没有降低计算量, 而是充分利用多核优势, 缩短碰撞检测需要的时间.

## 2 算法应用与分析

本文选择 CFETR 中两个相邻部件作为测试用例, 对算法性能进行测试. 待分析模型为有限元模型, 经三角化处理后重构为三角片网格模型.

算法由 C++ 代码实现, 程序的运行平台为 Windows 10 系统, 硬件参数为: CPU 参数 4 核 3.30 GHz, 内存 8 GB.

### 2.1 算法效率

待处理模型的基本参数, 包括三角形面元、碰撞三角形、碰撞区域占比等参数如表 1 所示.

表 1 模型面元数量与碰撞区域数据

模型名称	三角形面元数量	碰撞三角形数量	碰撞区域百分比 (%)
真空室	404268	7491	1.9
冷屏	128544	4733	3.7

为比较算法的性能与优化效果, 在碰撞检测过程中, 使用多种不同策略进行计算:

(1) 采用 Möller 算法遍历计算三角形相交.

(2) 采用 Devillers 算法遍历计算三角形相交.

(3) 在方法 2 基础上, 为三角形建立包围盒, 先对包围盒测试相交, 再计算三角形相交.

(4) 在方法 3 基础上引入 AABB 树方法, 计算包围盒相交, 然后求解三角形相交.

(5) 对方法 4 中 AABB 树的构建进行优化.

(6) 在 5 基础上, 采用并行计算加速; 本文中选择了 4 线程进行计算.

采用不同策略时, 算法的性能如表 2 所示.

表 2 不同策略下的算法性能比较

计算策略	处理耗时 (s)	碰撞检测耗时 (s)	时间复杂度
1	2580	2579	$O(mn)$
2	2178	2177	$O(mn)$
3	1530	1529	$O(mn)$
4	4.8	2.5	$O(m \log n)$
5	1.7	0.3	$O(m \log n)$
6	1.2	0.16	$O(m \log n)$

对多种优化策略的比较如下:

(1) 三角形相交算法

设 A 与 B 模型分别包含  $m$ 、 $n$  个三角形, 在计算碰撞时, 如果采用直接遍历方法, 对 A 中的每一个三角形, 与 B 中每一个三角形进行相交测试, 相交测试的次数为  $m \times n$ , 时间复杂度为  $O(mn)$ . 策略 1 与策略 2 相比, 计算复杂度相同. 实验中 1 与 2 的对比表明, 矢量判别型的 Devillers 算法比标量判别型的 Möller 算法效率高 20% 左右.

(2) AABB 包围盒

在有限元计算的场景中, 每个三角片面积很小, 三角形间不相交是大概率事件. 采用 AABB 包围盒算法

时,依然需要将A中 $m$ 个包围盒与B中 $n$ 个求交,计算次数为 $m \times n$ ,时间复杂度为 $O(mn)$ ;但是AABB包围盒可以将减少每次排除相交的计算量:AABB包围盒只需要最多9次计算可以排除相交,而Devillers算法在排除相交时,最少需要进行43次计算.实验结果表明,采用AABB包围盒,相比直接进行三角形相交测试,可以减少30%左右的计算时间.

### (3) AABB 树

采用层次包围盒算法时,考虑理想情况下,包围盒与一棵 $n$ 个叶子的二叉树求交时,每次在两个子节点中只与一个相交,则只需要 $\log_2 n$ 次查询,因此层次包围盒算法的时间复杂度为 $O(m \log_2 n)$ .在实际的二叉树建立过程中,由于包围盒的位置随机分布,导致建立的二叉树平衡性较差,复杂度高于到 $\log_2 n$ 级别.3与4的对比显示,采用AABB树可以降低计算耗时的量级.

### (4) 优化的 AABB 树

方法4与5的对比说明,采取空间划分的方式对包围盒进行二分,通过提高二叉树的平衡性,可以优化AABB树的结构,从而降低求交的次数.实验表明,采取优化的AABB树,相比不优化情况下碰撞检测的时间降低60%以上.

### (5) 并行计算

方法6采用并行方法进行计算,利用多个线程,进行二叉树构建、AABB树的碰撞检测.本次实验中,并行方法可以将计算耗时降低30%左右.

## 2.2 算法应用

将该快速碰撞检测算法应用于CFETR的部件检测中,所得计算结果如图6所示.

在实际工程设计中通常使用建模软件CATIA进行数字化建模、碰撞检测,因此本文选择CATIA的碰撞计算结果进行对比.CATIA的计算结果如图7所示.

碰撞计算结果的对比显示,快速检测算法给出的碰撞区域与CATIA计算结果基本一致.CATIA计算碰撞时计算区域的交线,而算法简化为计算碰撞的三角面对,大幅提高效率的同时,能够给出可靠的结果.

在实际设计计算中,有限元模型的网格数量往往在百万、千万量级,此时算法的处理效率更加重要.

本文通过将模型B进行网格重,衡量算法处理大规模网格时的性能.

模型的网格面数量和计算耗时对比如表3所示.

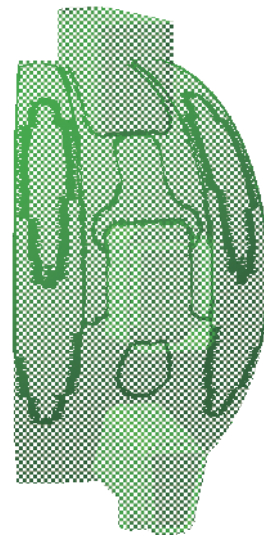


图6 算法的碰撞计算结果

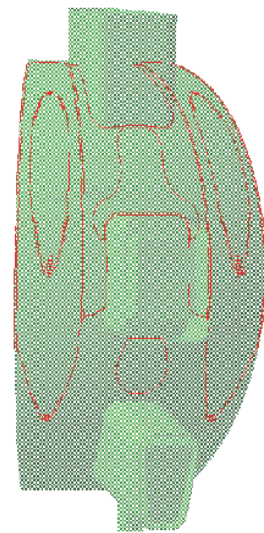


图7 CATIA的碰撞计算结果

表3 大规模网格计算的性能比较

模型A面元数量	模型B面元数量	快速算法耗时(s)	CATIA耗时(s)
404 268	128 544	1.2	75
	461 728	1.3	180
	1391 840	3.6	330

实验结果表明,本文提出的快速检测算法在处理复杂有限元网格时,相比CATIA的计算,效率上存在明显优势.将本文提出的算法应用在聚变堆工程设计中,可以大幅缩短计算时间,提高了工程设计的效率.

## 3 结论

针对工程设计中存在的有限元模型碰撞问题,本

文设计并实现了一种基于三角形相交检测、AABB包围盒的快速碰撞检测算法,通过采用三角形的 Devillers 算法、优化的 AABB 树和并行化设计,实现高效的碰撞检测.将算法应用于聚变堆形变部件检测,测试结果表明,本算法在计算效率、处理大规模网格方面都取得了良好的效果.

在进一步的工作中,该算法将被集成至 CFETR 集成设计平台,为支持聚变堆设计提供快速的形变部件碰撞检测功能.

### 参考文献

- 1 Wan YX, Li JG, Liu Y, *et al.* Overview of the present progress and activities on the CFETR. *Nuclear Fusion*, 2017, 57(10): 102009. [doi: [10.1088/1741-4326/aa686a](https://doi.org/10.1088/1741-4326/aa686a)]
- 2 Wan BN, Ding SY, Qian JP, *et al.* Physics design of CFETR: Determination of the device engineering parameters. *IEEE Transactions on Plasma Science*, 2014, 42(3): 495–502. [doi: [10.1109/TPS.2013.2296939](https://doi.org/10.1109/TPS.2013.2296939)]
- 3 Song YT, Wu ST, Li JG, *et al.* Concept design of CFETR Tokamak machine. *IEEE Transactions on Plasma Science*, 2014, 42(3): 503–509. [doi: [10.1109/TPS.2014.2299277](https://doi.org/10.1109/TPS.2014.2299277)]
- 4 Ye MY, Wang ZW, Mao SF, *et al.* Integration design platform of the CFETR. *Fusion Engineering and Design*, 2017, 123: 87–90. [doi: [10.1016/j.fusengdes.2017.05.093](https://doi.org/10.1016/j.fusengdes.2017.05.093)]
- 5 Ye MY, Wang SJ, Mao SF, *et al.* Development of CFETR integration design platform: Modular structure. *IEEE Transactions on Plasma Science*, 2017, 45(3): 512–518. [doi: [10.1109/TPS.2017.2655548](https://doi.org/10.1109/TPS.2017.2655548)]
- 6 Park SH, Kim JY, Park WW, *et al.* The effect of plastic deformation on low temperature mechanical and magnetic properties of Austenite 316LN tube for ITER TF conductor. *IEEE Transactions on Applied Superconductivity*, 2012, 22(3): 7800204. [doi: [10.1109/TASC.2011.2176894](https://doi.org/10.1109/TASC.2011.2176894)]
- 7 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.
- 8 Curtis S, Tamstorf R, Manocha D. Fast collision detection for deformable models using representative-triangles. *Proceedings of 2008 Symposium on Interactive 3D Graphics and Games*. Redwood City, CA, USA. 2008. 61–69.
- 9 吴峥,谢叻,马浩博.虚拟手术实时物体碰撞检测和软组织变形研究. *计算机仿真*, 2010, 27(2): 255–259. [doi: [10.3969/j.issn.1006-9348.2010.02.061](https://doi.org/10.3969/j.issn.1006-9348.2010.02.061)]
- 10 Van Den Bergen G. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 1997, 2(4): 1–13. [doi: [10.1080/10867651.1997.10487480](https://doi.org/10.1080/10867651.1997.10487480)]
- 11 马登武,叶文,李瑛.基于包围盒的碰撞检测算法综述. *系统仿真学报*, 2006, 18(4): 1058–1061, 1064. [doi: [10.3969/j.issn.1004-731X.2006.04.063](https://doi.org/10.3969/j.issn.1004-731X.2006.04.063)]
- 12 赵亮,张义德,胡旭晓.基于网格包络的工业机器人仿真碰撞检测算法. *中国机械工程*, 2017, 28(3): 316–321. [doi: [10.3969/j.issn.1004-132X.2017.03.011](https://doi.org/10.3969/j.issn.1004-132X.2017.03.011)]
- 13 王晓荣,王萌,李春贵.基于AABB包围盒的碰撞检测算法的研究. *计算机工程与科学*, 2010, 32(4): 59–61.
- 14 Möller T. A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 1997, 2(2): 25–30. [doi: [10.1080/10867651.1997.10487472](https://doi.org/10.1080/10867651.1997.10487472)]
- 15 Guigue P, Devillers O. Fast and robust triangle-triangle overlap test using orientation predicates. *Journal of Graphics Tools*, 2003, 8(1): 25–32.
- 16 邹益胜,丁国富,何邕,等.空间三角形快速相交检测算法. *计算机应用研究*, 2008, 25(10): 2907–2909.