

建造场景动态体素化碰撞检测方法研究

林昊¹, 吴志铭¹, 金季岚²

(1. 厦门理工学院土木工程与建筑学院, 福建 厦门 361024;
2. 厦门海迈科技股份有限公司, 福建 厦门 361008)

摘 要: 在建造场景所有安全事故中, 碰撞事故被认为是最常见的伤害之一。为能有效预防监测碰撞事故的发生, 采用计算机图形分析技术辅助碰撞检测分析, 具有一定成效, 但在检测的实时性与高精度的平衡上仍存在局限。为了解决这个问题, 提出了一种基于动态体素化的碰撞检测方法, 即融合空间动态体素树生成与资源动态球状体素化计算, 构建了一种碰撞检测分析机制。核心思路在于: ①基于拥挤度阈值, 递归分割空间生成动态体素树, 有效过滤非碰撞风险区域; ②依据资源间相对距离和资源体积动态计算体素单元边长, 实现体素粒度的自适应调节; ③采用球状体素替代传统立方体体素, 避免非轴对齐检测的计算负担; ④引入空心化处理剔除内部无效体素, 进一步优化检测效率。该方法能够在复杂动态建造环境中精准捕捉资源交互, 显著提升检测精度并优化计算效率。实验结果表明, 相较于传统方法, 该方法在检测精度上显著提高, 精确率与准确率分别达到 94.64% 与 96.67%。在碰撞检测时间上, 比多数现有方法更具效率, 计算速度至少提升了 11.36%。同时, 研究分析了体素树深度、根节点尺寸和体素边长参数对性能的影响, 并分析了不同规模场景的 CPU 资源与内存资源的消耗。消耗量处于可接受范围内, 验证了其在建造场景的适用性。该方法为提升建造安全管理智能化水平提供了有效的信息化处理新思路。

关键词: 建造场景; 计算机图形分析; 空间动态体素树; 动态球状体素化; 自适应调节; 碰撞检测

中图分类号: TU 712.4

DOI: 10.11996/JGj.2095-302X.2026010204

文献标识码: A

文章编号: 2095-302X(2026)01-0204-12

Research on dynamic voxelization-based collision detection in construction scenarios

LIN Hao¹, WU Zhiming¹, JIN Jilan²

(1. School of Civil Engineering and Architecture, Xiamen University of Technology, Xiamen Fujian 361024, China;

2. Xiamen Hymake Technology Co., Ltd., Xiamen Fujian 361008, China)

Abstract: Among all safety accidents in construction scenarios, collision accidents are regarded as one of the most common types of injury. To effectively prevent and monitor the occurrence of collision accidents, the computer graphics analysis technology has been used to assist collision detection and analysis; however, limitations remain in balancing the real-time performance with high precision of detection. To address this, a collision-detection method based on dynamic voxelization was proposed. This method integrated the generation of dynamic spatial voxel tree with the dynamic spherical voxelization calculation of resources to construct a collision detection and analysis mechanism. The core ideas are as follows: ① Based on the crowding-degree threshold, the space was recursively divided to generate a dynamic voxel tree, effectively filtering out non-collision risk areas. ② The side length of voxel units were dynamically calculated according to the relative distance between resources and resource volume, realizing the adaptive adjustment of voxel granularity. ③ Spherical voxels were used instead of traditional cubic voxels to

收稿日期: 2025-03-24; 定稿日期: 2025-07-21; 通信作者: 吴志铭, E-mail: 2014990901@xmut.edu.com

Received: 24 March, 2025; Finalized: 21 July, 2025; Corresponding author: WU Zhiming, E-mail: 2014990901@xmut.edu.com

基金项目: 国家自然科学基金(51808474); 福建省自然科学基金(2023J011441)

Foundation items: National Natural Science Foundation of China (51808474); Fujian Provincial Natural Science Foundation of China (2023J011441)

avoid the computational burden of non-axis-aligned detection. ④ A hollowing-out procedure was introduced to eliminate internal invalid voxels, further optimizing detection efficiency. This method can accurately capture resource interactions in complex dynamic construction environments, significantly improving detection accuracy and optimizing computational efficiency. Experimental results showed that compared with traditional methods, the proposed method significantly improved the detection accuracy, with precision and accuracy reaching 94.64% and 96.67%, respectively. In terms of collision detection time, it was more efficient than most existing methods, with a calculation speed increase of at least about 11.36%. At the same time, the study analyzed the impact of key parameters such as voxel-tree depth, root-node size, and voxel side length on performance, and analyzed the consumption of CPU resources and memory resources by the method in scenarios of different scales. The consumption was within an acceptable range, verifying the applicability of the method in construction scenarios. The method provided an effective new idea of information processing for enhancing the intelligent level of construction safety management.

Keywords: construction scenarios; computer graphics analysis; dynamic spatial voxel tree; dynamic spherical voxelization; adaptive adjustment; collision detection

在建造场景资源密度高且环境多变的施工条件下, 人员、设备、材料等各类建造资源之间的时空冲突和碰撞风险将显著增加^[1]。如何确保建造资源之间的无碰撞安全生产, 一直是建造管理的目标与重要挑战。由于现场人工监控和预判存在明显的滞后性与局限性, 因此, 如何应用计算机图形技术辅助碰撞检测分析, 充分发挥其自动化效率, 提高建造场景自动碰撞检测的精度与速度, 成为建设项目信息化安全管理中的关键问题。计算机图形技术辅助碰撞检测分析的方法, 是指使用计算机技术自动判定在同一时间点, 一对或多对物体是否占据相同或重叠的空间^[1-4]。

欧几里得距离的邻近碰撞检测方法是计算机图形技术辅助碰撞检测分析的常见方法之一。其基于资源坐标, 将资源或危险区域视为“点对点”“点对面”或“点对体”等情况进行分析。监测目标资源相对于危险区域的位置坐标, 并通过计算其之间的相对距离, 判断碰撞风险及输出相对应的分析结果^[5-8]。然而, 当被检测物为一个体积或跨度很大的物体时, 基于欧几里得距离的碰撞检测方法存在检测精度不足的问题。

此外, 一些研究考虑到被检测物的三维形状, 相比基于欧几里得距离的邻近碰撞检测, 在满足资源空间实体精度上具有优势。一些方法考虑到资源本身的几何参数与特征, 如轴对齐包围盒(Axis-Aligned Bounding Box, AABB)、定向包围盒(Oriented Bounding Box, OBB)和球体包围盒(Sphere Bounding Box, Sphere)算法^[9-13]。然而, 涉

及资源数量多且复杂的建造项目时, 上述方法则需要较长的计算时间, 且碰撞检测精度不足(包围盒无法贴合资源本身三维形状)。而空间分割(Spatial Partitioning, SP)算法的原理是将空间分割成基元, 然后进行碰撞检测, 且只检查周围的障碍物, 检测速度较快。SP 方法是将检测资源简化表示为点的形式并进行分析^[14], 在碰撞检测上缺少精度, 不适用于大体积资源间的相交检测。有研究尝试将 2 种方法进行结合或改进, 试图解决碰撞检测时间长, 精度不足的问题。LI 和 SHEN^[15]使用 OBB 以及空间二叉树划分, 实现在虚拟世界中较快的碰撞检测速度。TU 和 YUAN^[16]提出了新的 Sphere-OBB, 并以单次遍历的方式提高了碰撞检测的速度。以 Sphere 和 OBB 构建层次包围树, 当树具备一定深度时, 可减少 40%的时间。GAN 和 DONG^[17]提出 AABB-OBB 层次包围盒方法, 使用了同步下降规则, 并在构建 OBB 时, 将三角片网格面积作为加权参数计算其中心点位置。在实验场景的运算速度加快了 30%。程琦甫^[18]将 AABB-Sphere 层次包围盒用于资源自碰撞检测, 并使用深度神经网络提高 Sphere 的碰撞检测效率。然而, 该算法在面临多目标物体时将会增加碰撞检测时间。马晓萌等^[19]利用 AABB 结合空间八叉树原理, 将工作空间分割为不同的子节点, 将各节点与 AABB 进行关联, 达成工作空间中高效而精准的碰撞检测效果。ZHU 等^[20]通过改进八叉树的编码方式, 提出适用于装配式建筑的莫顿码八叉树与 VERTICAL-OBB 结合的检测方式, 实现了快速的碰撞检测。有效的碰撞检测策

略通常需要在“计算时间”和“检测精度”之间取得平衡,但方法多采取外包包围盒拟合资源实际的三维形状进行碰撞检测,仍无法达到高碰撞检测精度下具有较少的计算时间的目的。

文献[21-22]考虑到外包包围盒在碰撞检测精度上的局限性,结合层次包围盒或八叉树分割,使用资源的三角网格进行碰撞检测。此类方法检测精度极高,但碰撞检测仍会消耗较长的时间,且受形状复杂度影响较大,更适用于集中的规则建筑构件之间进行碰撞检测,不适用于资源形状复杂的建造场景(包含各种机械,人员等)。一些研究将体素化后的资源体素单元用于碰撞检测。LI 等[23]提出一种以 BIM 为基础的碰撞检测方法,利用资源体素来侦测高空坠落危险,在不同场景模型上,检测结果精确率达到 85.98%以上。但由于体素单元过多,检测耗时较长,最高可达 19 285 s,不适用于动态的施工现场。谢震鹏[24]通过对比 2 个资源体素坐标的相对位置,与 2 个资源外包球体的球心的相对位置来判断碰撞结果,其计算复杂且检测耗时。徐进等[25]依据点云数据,建立体素化碰撞模型,进行机械臂与零件之间的碰撞检测,其检测精度高,但未考虑非轴对齐问题,仍存在碰撞检测精度损失的情况,且对于小零件精度损失更大。

现有碰撞检测方法在应对计算机辅助建造场景时,常面临精度与效率难以兼顾的挑战。基于欧氏距离或包围盒的方法虽计算高效,但因对复杂三维资源形状的拟合度不足,导致检测精度普遍较低。而基于三角网格的精确检测方法虽能保证高精度,其计算耗时却使其难以适用于动态建造过程。体素化方法作为一种折中方案,其性能与精度严重依赖于体素粒度与空间划分策略;传统的均匀、轴对齐体素化会因“阶梯效应”与大量无效的内部体素而引入误差并增加计算负担,且未能充分考虑资源体积与方向多样性带来的影响。因此,本研究提出一种基于动态体素化的碰撞检测方法,通过阈值分割空间、自适应调节体素粒度、采用球状体素及空心化剔除无效体素等核心优化,旨在在动态建造场景中同步提升检测精度、计算效率与应用适应性。

1 基于动态体素化的碰撞检测方法

基于动态体素化的碰撞检测方法融合了空间动态体素树生成与资源动态球状体素化计算,检测流程如图 1 所示。

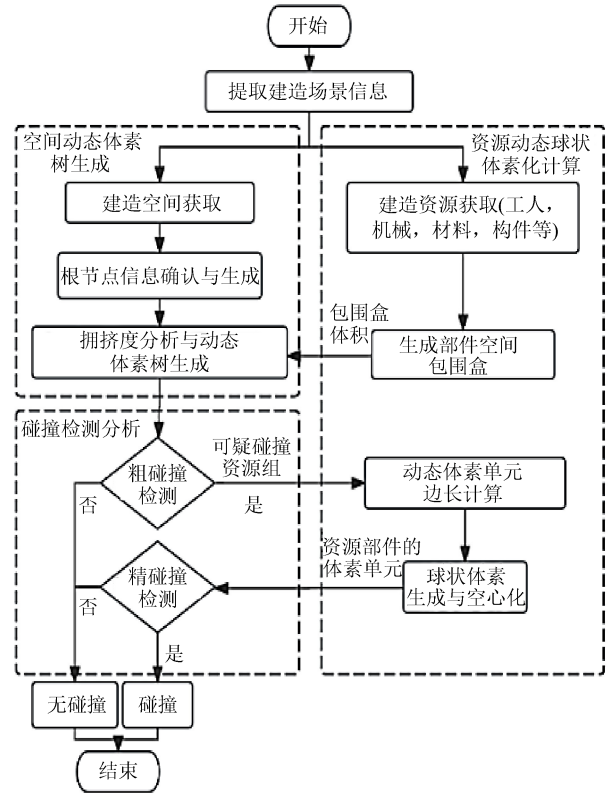


图 1 基于动态体素化的碰撞检测流程图

Fig. 1 Flow chart of collision detection based on dynamic voxelization

1.1 空间动态体素树生成

通过快速过滤可能存在的碰撞风险的资源进入精碰撞检测,以减少不必要的精碰撞检测次数及检测时间。设定拥挤度阈值,动态地将建造现场空间进行体素化递归分割,生成体素树。体素树的根节点表示包围建造场景的轴对齐立方体区域,通过获取建造场景的活动空间,可确定体素树根节点的尺寸 L_0 , 及设置合适的根节点中心点 B_0 , 使得根节点能够包围全部的建造资源。由于体素树的每个节点的空间区域都是轴对齐立方体,通过每次递归可将其分割为 8 个相同的子空间。设定体素树的节点为 B_i , 中心点为 C_i , 边长为 L_i , 则下一级子节点、中心点和边长可分别表示为

$$B_{i+1} = (C_{i+1}, L_{i+1}) \quad (1)$$

$$C_{i+1} = C_i + (\pm L_i/4, \pm L_i/4, \pm L_i/4) \quad (2)$$

$$L_{i+1} = L_i/2 \quad (3)$$

式中:正负号决定了子节点的不同方向; B_{i+1} 表示节点; C_{i+1} 表示中心点; L_{i+1} 表示边长。

拥挤度表示当前体素节点的预分割子节点中资源的空间占据程度,可通过计算每个预分割的下一级节点内资源的占用体积,及体积比率得到。预分割的下一级节点内资源的占用体积为与其相交

的资源部件的轴对齐包围盒的体积相同。

如图 2 所示, 若当前节点的下一预分割节点内资源的拥挤度比率超过所设阈值, 则继续分割, 直至达到设定的体素树最大深度为止。不同深度的节点具有不同的拥挤度比率阈值, 其计算式为

$$Y = \begin{cases} 0.05, & dp \leq 3 \\ 0.1 \times (dp - 2), & dp > 3 \end{cases} \quad (4)$$

式中: Y 表示拥挤度阈值; dp 表示当前节点的深度。

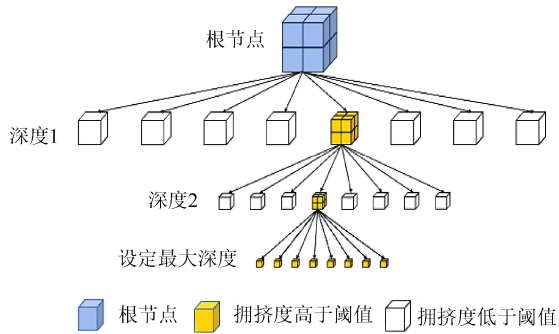


图 2 体素树结构与分割条件

Fig. 2 Voxel tree structure and segmentation conditions

区别于传统八叉树或依据资源数量进行分割的方法^[26], 结合拥挤度生成体素树方法在保证一定计算速度的前提下, 减少了计算机运行内存的占用, 同时可根据动态场景的变化, 选择最优的分割方式, 达到最优的计算效率, 更加适合规模大且复杂的建造场景。如图 3(a)~(b)所示, 当资源数量少且拥挤度比率或局部拥挤度比率很高时, 传统八叉树分割方式不再继续分割, 但体素树方法将继续进行分割, 从而过滤出真正可疑碰撞的资源, 加快后续的碰撞检测速度。如图 3(c)所示, 当资源数量极多, 且资源体积相对于场景体积较小时, 拥挤度比率也较低。因为资源数量极多, 传统的八叉树分割方式会依据数量阈值进行多层分割, 将显著增加计算机内存占用, 及碰撞检测时间。而体素树方法不会进行过深的体素树分割, 而在适当分割后就进行精碰撞检测, 因为资源体积较小, 生成的体素单元较少, 即使进入后续的精碰撞检测分析, 其增加时间也相对有限。如图 3(d)所示, 当资源数量较少, 资源的体积相对于场景体积较大时, 拥挤度比率较高。因为资源数量较少, 且未达到传统八叉树分割的数量阈值时不会进行分割。但体素树方法会进行进一步的分割, 从而过滤出真正可疑碰撞的资源, 进入精碰撞检测, 避免更多的体素单元生成, 加快了碰撞检测速度。由于资源数量较少, 体素树的分割也不会产生较多的子体, 占据过多的运行内存。

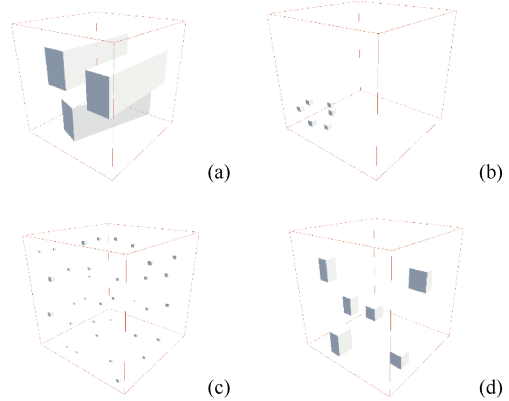


图 3 建造场景拥挤度示例((a) 资源数量少, 全场拥挤度比率高; (b) 资源数量少, 局部拥挤度比率高; (c) 资源数量极多, 拥挤度比率低; (d) 资源数量较少, 拥挤度比率较高)

Fig. 3 Examples of crowdedness in architectural scenarios ((a) Small quantity of resources, high overall crowding ratio; (b) Small quantity of resources, high local crowding ratio; (c) Extremely large quantity of resources, low crowding ratio; (d) Relatively small quantity of resources, relatively high crowding ratio)

根据拥挤度阈值生成的空间动态体素树如图 4 所示。图中人员所处位置拥挤度较高, 需进行更深层次的分割, 至设定的最大深度为止, 以加快碰撞检测速度。其他空间则无需继续分割, 由此减少了计算机内存的占用。

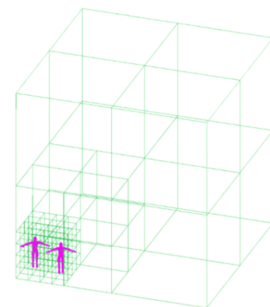


图 4 空间动态体素树示例

Fig. 4 Dynamic spatial voxel tree example

1.2 资源动态球状体素化计算

以获取建造场景资源作为被检测物时, 为提高方法的实际应用效率和针对性, 无需对所有场景资源都进行碰撞检测。可从资源的动态性, 风险等级及管理人员经验确定被检测资源, 一般包括工人、机械设备、建筑材料与构件等。本研究对资源所占据的工作空间进行轴对齐包围盒的生成。为了更好地拟合资源三维形状, 依据资源每个部件的最大点 P_{max} 与最小点 P_{min} , 对资源的每个部件分别进行轴对齐包围盒生成, 如图 5 所示。其中 P_{max} 与 P_{min} 的

定义为

$$P_{\max} = (\max(x_i), \max(y_i), \max(z_i)) \quad (5)$$

$$P_{\min} = (\min(x_i), \min(y_i), \min(z_i)) \quad (6)$$

式中: (x_i, y_i, z_i) 表示资源部件顶点三维坐标。

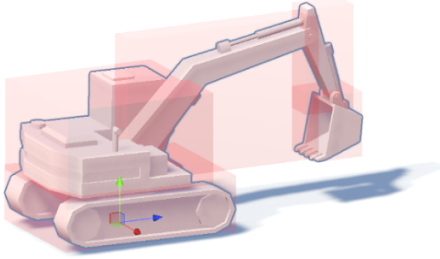


图5 部件轴对齐包围盒

Fig. 5 Component axis-aligned bounding box

轴对齐包围盒是将资源进行体素化分割,以生成体素单元,再进行精碰撞检测。设定当前检测资源的体素单元边长为 N ,其随资源间的相对距离与资源的体积动态而变化。当资源距离较远且体积较大时,使用的体素边长较大。相反,当资源距离较近且体积较小时,使用的体素边长较小。相对距离为

$$d_{AB} = \|C_A - C_B\| \quad (7)$$

式中: C_A 表示被检测资源的质心; C_B 表示与被检测资源相对距离最近的资源质心; d_{AB} 表示 2 个资源之间的质心距离。

体素单元边长为

$$N = \frac{N_{\max}}{1 + \alpha \cdot (d_{AB} / D_{\max})} \cdot \frac{1}{1 + \beta \cdot (V_{AB} / V_{\text{scale}})} \quad (8)$$

$$V_{AB} = \sqrt[3]{V_A \cdot V_B} \quad (9)$$

式中: N_{\max} 表示最大的体素单元边长,由资源类别决定; D_{\max} 表示影响 N 的最大距离阈值,超过此距离 N 不再变化; V_{AB} 表示待测资源的几何平均体积,用于综合考虑资源的体积大小; V_{scale} 表示设定的资源体积的标定参数,决定资源体积大小对 N 的影响; α 表示距离的权重因子,用于控制距离对 N 的影响; β 表示体积大小的权重因子,用于控制资源体积大小对 N 的影响。

确定体素边长 N 后,每个维度的体素数量为

$$N_x = \text{ceil}[(P_{\max} - P_{\min})_x / N] \quad (10)$$

$$N_y = \text{ceil}[(P_{\max} - P_{\min})_y / N] \quad (11)$$

$$N_z = \text{ceil}[(P_{\max} - P_{\min})_z / N] \quad (12)$$

式中: N_x , N_y 和 N_z 分别表示在 X , Y 和 Z 轴上的体素单元数量; ceil 表示向上取整函数。

因为施工现场不会出现资源相互嵌入的情况,所以依据资源模型的三角网格来生成体素单元,并进行碰撞检测。通过计算资源模型的三角网格与体素单元是否相交来确定资源的体素单元。将每个三角网格顶点分配到相应的体素中,即确定顶点所在的体素索引

$$P_i = \text{floor}[(v_i - P_{\min}) / N] \quad (13)$$

式中: P_i 表示三角网格顶点 v_i 的索引位置; floor 表示向下取整函数。

根据体素索引得到的可能与三角网格相交的体素单元,基于此使用分离轴定理(Separating Axis Theorem, SAT),通过计算三角网格与体素单元的边界框进行碰撞相交检测,以对三角网格与体素单元的相交检测进行算法优化。具体方法为

$$T_{\max} \geq V_{\min} \quad (14)$$

$$V_{\max} \geq T_{\min} \quad (15)$$

式中: T_{\max} 和 T_{\min} 分别表示体素单元在某一分离轴上的最大投影值与最小投影值; V_{\max} 和 V_{\min} 分别表示三角网格在某一分离轴上的最大投影值与最小投影值。若在所有分离轴上同时满足式(14)~(15),则表示体素单元与三角网格相交。分离轴包括 X 轴、 Y 轴、 Z 轴、三角网格法线轴及体素单元边长与三角网格边长叉积方向轴。资源的初步体素化需通过保留与三角网格相交的体素单元实现。

生成的立方体体素单元不适合直接进行碰撞检测^[27]。如图 6 中 C 和 D 所示,在场景中,当立方体体素单元边框与坐标轴平行时,碰撞检测较为简单,只需比较单元的最大顶点 C_{\max} , D_{\max} 与最小顶点 C_{\min} , D_{\min} 坐标之间的大小关系即可^[28]。当资源转向时,单元边框与坐标轴不保持平行,如图 6 中 A 和 B 所示,会出现非轴对齐情况。立方体体素单元的边框不与坐标轴平行时,碰撞检测将使用分离轴定理进行,2 个立方体单元之间最多需要 15 次计算,计算耗时^[29]。图中示例以 X 轴、 Y 轴为分离轴,需比较各轴上的 A_{\max} , B_{\max} , A_{\min} , B_{\min} 距离来判断碰撞。其中 $A_{\max} = B_{\max} = \max(p_{i+1}, p_{i+2}, p_{i+3})$, $A_{\min} = B_{\min} = \min(p_{i+1}, p_{i+2}, p_{i+3})$ 。如图 6 中 E 和 F 所示,采用球状体素时将不会出现非轴对齐的问题^[30],只需要计算球体与球体之间的距离 L_{EF} 与各半径之间的大小关系来判断碰撞与否。此外,使用原立方体体素的内切球代替立方体体素时,只需存储球心位置和半径这 2 个参数即可,能够显著减少计算机内存的占用。因此根据已生成的立方体体素单元,生成其内切球,代替原先

的立方体体素单元进行后续的碰撞检测。

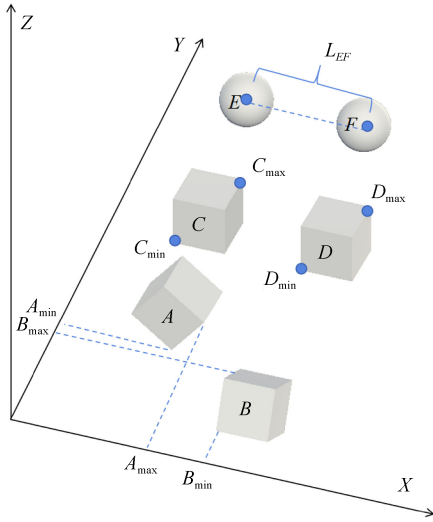


图 6 体素单元碰撞示意

Fig. 6 Schematic diagram of voxel unit collision

三角网格本身包含双层结构, 资源内外层都会生成体素。针对建造场景一些特殊资源, 图 7(a)为一双层结构的特殊资源, 图 7(b)为其球状体素化后的水平剖面图。内部生成的体素对碰撞检测结果意义不大, 将其删除可加速后续的碰撞检测速度。假设从资源球状体素的质心 P_v 沿各个三角网格面方向发射射线, 即

$$r(t_m) = P_v + t_m \cdot D_f \quad (16)$$

式中: D_f 表示射线方向向量, 射线均与三角网格面的法向量平行; t_m 表示控制射线沿着 D_f 方向的移动程度; P_v 表示射线的起点坐标。当射线 $r(t_m)$ 与资源三角网格面相交时, 计算射线与所有三角网格面交点个数。其中, 射线 $r(t_m)$ 与三角网格面相交的条件为

$$t_m = \mathbf{n}_f \cdot (v_i - P_v) / (\mathbf{n}_f \cdot D_f) \quad (17)$$

式中: \mathbf{n}_f 表示平面法向量; 如果 $t_m \geq 0$, 则射线与资源的三角网格相交。如果每一条射线与三角网格面的相交数量为偶数, 则该体素位于网格的内部, 否则位于外部。删除位于内部的体素, 只保留模型外表面的体素, 进入精碰撞检测。

1.3 碰撞检测分析

首先, 将资源部件的轴对齐包围盒与生成的动态体素树进行基于轴对齐包围盒的粗碰撞检测, 输出当前与体素树各个分枝最深子节点相交的所有资源的部件。接着将与同一个最深子节点相交的所有资源部件视为一个可疑碰撞资源组。最后对各个组内资源部件进行精碰撞检测。精碰撞检测基于

资源部件分割的动态球状体素单元, 进行球状体素单元之间的遍历碰撞检测, 从而得到最后的碰撞结果。

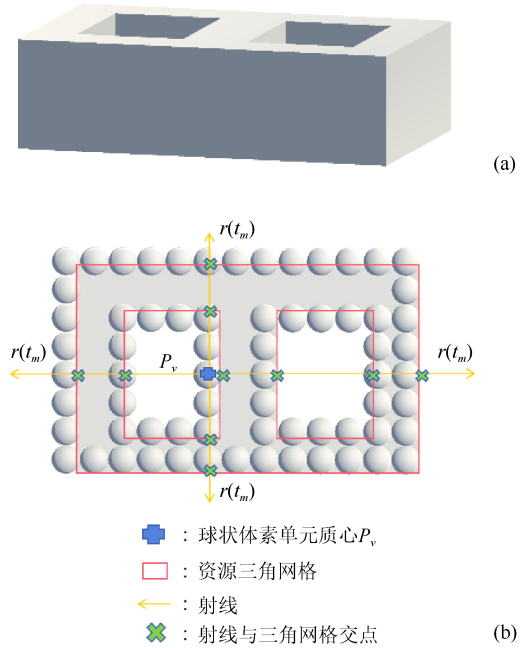


图 7 空心化示例((a) 双层结构的特殊资源; (b) 球状体素化后的水平剖面图)

Fig. 7 Hollowing example ((a) Special resources with a double-layer structure; (b) Horizontal cross-sectional view after sphere voxelization)

2 实验测试与结果分析

2.1 测试环境与场景

所有实验都在 i9-13900H 处理器和 16 GB RAM 的计算机上运行, 使用基于 c# 语言的 Unity3D2023.2.20f1c1 编辑器进行模拟与实验, 并一个模拟场景进行评估。该场景存在 56 个风险资源, 包括一栋拟建的 4 层建筑的首层与正在施工的第 4 层, 总共 36 个结构构件(高风险资源, 首层构件与挖掘机有碰撞的可能, 正在施工的第 4 层构件则可能与人员有不安全接触, 一旦发生危险碰撞, 可能引发结构安全事故与人员伤亡的后果严重)以及 2 台挖掘机设备、8 名工人(动态高风险资源)与 10 处静态建筑材料堆料(高风险资源)。其中挖掘机与工人按照设计的路线进行移动与回转, 包括资源远离与靠近。如图 8(a)所示, 模拟检测的场景范围为一个 $50 \text{ m} \times 50 \text{ m} \times 50 \text{ m}$ 的立方体, 用细线显示, 拟建建筑位于场景的一角, 建筑构件较多, 空间较为拥挤, 用粗线显示。模拟检测场景的南立面图, 东立面图, 俯视图如图 8(b)~(d)所示, 红色虚线框

内为高风险拟建建筑构件。测试的资源参数见表 1。

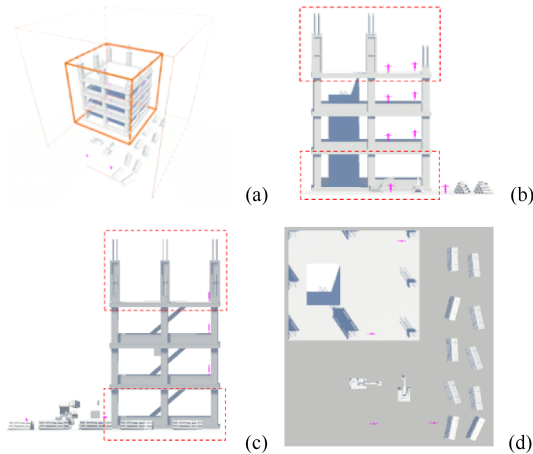


图 8 模拟测试场景((a) 常规视角; (b) 南立面图; (c) 东立面图; (d) 俯视图)

Fig. 8 Simulate the test scenario ((a) Conventional view; (b) South elevation; (c) East elevation; (d) Top view)

表 1 资源参数

Table 1 Resource parameter

资源	长×宽×高/m	资源体积 V/m^3
挖掘机	6.26×2.56×3.58	7.900
工人	1.71×0.37×1.78	0.090
静态堆料	2.2×2×2.8	6.920
拟建建筑	22×19×18	5 643
拟建建筑(梁构件)	4.2×0.6×0.6	1.512
拟建建筑(柱构件)	0.75×0.75×4.5	2.530
拟建建筑(板构件)	5.5×4.75×0.12	3.140
挖掘机移动速度	0.5 m/s	
挖掘机机身回转速度	10 °/s	
工人移动速度	1.7 m/s	

表 2 体积误差比较

Table 2 Comparison of volume errors

资源	(1)	(2)	(3)	(4)	(5)	(6)
拟建建筑(梁构件)	0~6.13	0	28.56	0	0	0.91~2.01
静态堆料	3.97~8.82	3.97	31.49	3.97	0.02	0.87~1.95
挖掘机(水平状态)	6.01~13.16	6.01	21.55	6.01	0.06	0.81~1.79
挖掘机(倾斜状态)	11.22~23.11	6.01	21.55	11.22	0.06	0.81~1.79
工人(臂展状态)	12.26~44.29	12.26	39.33	12.26	0.09	1.12~2.93

注: (1)表示 AABB; (2)表示 OBB; (3)表示 Sphere; (4)表示 VERTICAL-OBB; (5)表示三角网格; (6)表示本文方法(非空心化)。

图 9 展示了不同拟合方法在形状上的拟合效果,除了使用三角网格的方法,即使是最大的体素单元边长,球状体素化方法对资源的形状拟合仍然最好,最贴合资源本身三维形状。综上,与传统方式对比,除了使用三角网格的方法,采用体素化的方法对资源进行拟合的精度最高。

2.3 检测精度与时间比较

体素树的分割方式只是加快碰撞检测速度,影响碰撞检测精度的是各种拟合方式。为了验证本方

2.2 拟合精度比较

为了验证本方法在拟合精度上的成效,使用模拟场景对资源在不同三维拟合方法下的拟合精度进行比较。将所提出的球状体素化方法与传统的 AABB、OBB、Sphere、VERTICAL-OBB 及三角网格方法在不同资源上进行拟合精度的比较。因为拟建建筑构件为长方体,细长型几何特征对拟合方法的精度要求更高,故采用梁构件参与比较。对资源内部空间进行体素填充,以完成整个资源体素化。基于模拟测试场景,由式(8)可得,梁构件、静态堆料、挖掘机及工人的体素单元边长范围分别为 0.81~1.76 m, 0.86~1.82 m, 0.35~1.02 m 和 0.12~0.33 m。为了更好地验证本方法在拟合精度上的优势,施工现场挖掘机载体分为与地面水平及与地面夹角 45°的倾斜状态,工人为臂展状态。

衡量不同拟合方法在体积上的误差采用体积误差计算,即

$$V_c = |V_n - V_w| / V_w \quad (18)$$

式中: V_c 表示体积误差; V_n 表示拟合物体体积; V_w 表示资源体积。使用体积误差来比较资源拟合精度,是因为其对资源体积大小变化的敏感度是一致的。体积误差越小,表示拟合物体体积越接近资源本身的体积,拟合精度越好。比较结果见表 2,除了三角网格的方法,球状体素化的方法体积误差次好,精度最高。

法在碰撞检测精度上的成效,使用模拟场景,将本方法与传统的 AABB、OBB、Sphere、VERTICAL-OBB 及三角网格方法在碰撞检测精度上进行比较。实验采用时间离散方式,设定检测时间间隔 Δt 为 0.1 s,实验总时间为 15.0 s,输出 150 帧的碰撞检测结果,并分析碰撞检测精度。比较结果如图 10 所示,可知本方法的碰撞检测准确率、误检率和精确率分别为 96.67%, 4.12%和 94.64%,除了三角网格的方法,都优于其他方法。

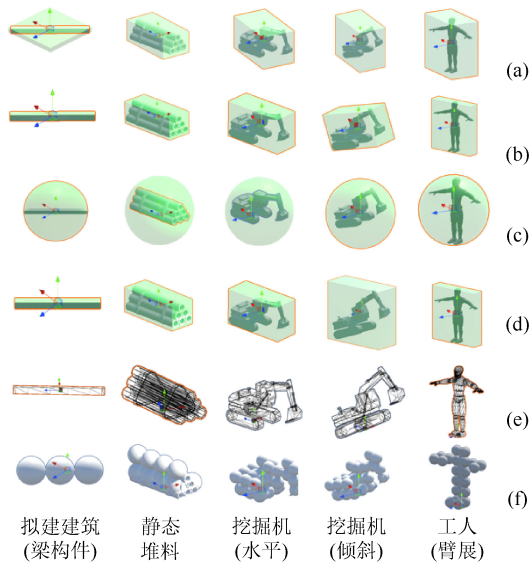


图 9 不同方法在形状上的拟合效果((a) AABB; (b) OBB; (c) Sphere; (d) VERTICAL-OBB; (e) 三角网格; (f) 球状体素化)

Fig. 9 The fitting effects of different methods in terms of shape ((a) AABB; (b) OBB; (c) Sphere; (d) VERTICAL-OBB; (e) Triangular mesh; (f) Spherical voxelization)

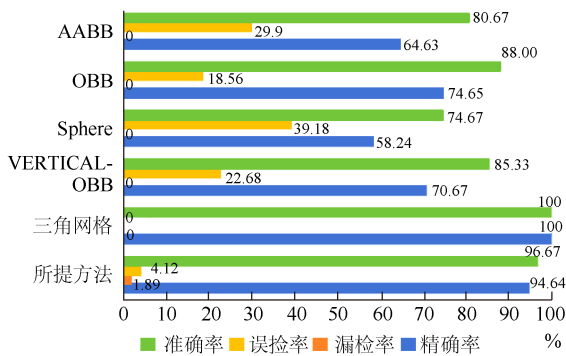


图 10 检测精度比较

Fig. 10 Comparison of detection accuracy

本方法的漏检率为 1.89%。漏检与误检的原因如图 11 所示, 绿色和红色分别表示资源 A 和 B 的三角网格与基于其生成的球状体素单元。图 11(a)为漏检情况, 球状体素单元未覆盖全部三角网格

面, 虽然资源 A 和 B 的三角网格已经相切, 但各自球状体素单元并未相交, 所以未检测出碰撞。图 11(b)为误检的情况, 可见资源 A 和 B 的三角网格未相交, 但各自球状体素单元已相交, 输出 2 资源碰撞, 造成误检。虽然本方法在碰撞检测会出现误检与漏检的情况, 由于误检率与漏检率极低, 均在实际应用可以接受范围。

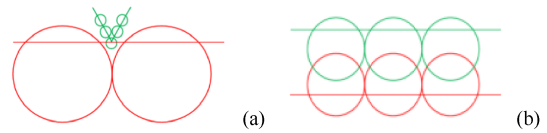


图 11 漏检与误检示例((a) 漏检情况; (b) 误检情况)
Fig. 11 Examples of missed detection and false detection ((a) Missed detection situation; (b) False detection situation)

为了验证本方法在碰撞检测时间上的成效, 使用模拟场景, 将本文方法与传统的 AABB、OBB、Sphere、VERTICAL-OBB、莫顿码八叉树与 VERTICAL-OBB 包围盒结合的方法^[20]和八叉树分割与包围盒结合三角网格的方法^[22]进行碰撞检测比较。此外, 为了体现所提出动态体素树空间生成方法的优势, 将传统八叉树空间分割方法与动态球状体素化精确碰撞检测方法相结合, 与本方法进行对比。

实验采用时间离散方式, 设定检测时间间隔 Δt 为 0.1 s。实验总时间为 15.0 s, 并记录包围盒生成、更新与碰撞检测所消耗的总时间(ms)。体素树设定的最大深度为 7, 传统八叉树的数量阈值为 3。根节点坐标为(0,0,22.5), 尺寸为 55 m, 比较结果见表 3。可知与使用传统包围盒的方式相比, 除了 AABB 方法外, 采用体素树与球状体素结合的方法在计算速率上更具优势。另外, 本方法比莫顿码八叉树与 VERTICAL-OBB 包围盒结合的方法快了 11.36%, 比八叉树分割与包围盒结合三角网格的方

表 3 检测时间比较/ms

Table 3 Comparison of detection times/ms

方法	最大值	最小值	绝对差值	平均值
AABB	103.27	99.01	4.26	99.850
OBB	1 347.57	1 283.18	64.39	1 310.600
Sphere	1 295.34	1 192.79	102.55	1 238.170
VERTICAL-OBB	1 245.87	1 123.11	122.76	1 179.900
莫顿码八叉树与 VERTICAL-OBB 结合的方法 ^[20]	693.12	604.52	88.60	662.500
八叉树分割与包围盒结合三角网格的方法 ^[22]	7 635.98	7 682.33	46.35	7 647.820
传统八叉树与动态球状体素结合方法	4 128.39	3 945.67	179.72	4 076.820
本方法	639.13	567.88	71.25	587.230

法快了 92.32%，优于现有方法。比传统八叉树与球状体素结合方法快了 85.58%，绝对差值小，更稳定。分析原因为占据场景一角的拟建建筑构件，人员较多、空间较拥挤，而传统的八叉树基于数量阈值的分割空间方法无法进行更进一步的空间分割，显著增加了计算机在精碰撞检测的计算量。

2.4 树深度的敏感性

分析本方法设定的不同体素树最大深度下碰撞时间，验证设定的不同体素树最大深度对于碰撞检测时间的影响程度。研究采用时间离散方式，设定检测时间间隔 Δt 为 0.1 s。实验总时间为 15.0 s，并记录其时间内包围盒生成、更新与碰撞检测所消耗的总时间(ms)。根节点坐标为(0, 0, 22.5)，尺寸为 55 m。实验结果如图 12 所示，随着设定的体素树最大深度的增加，精碰撞检测时间减少，粗碰撞检测时间增加。计算总时间呈先减少后增加趋势，在设定的体素树最大深度为 7 时，达到最快的计算速度。随着设定的体素树最大深度的增加，需要递归进行粗碰撞检测的体素树节点增加，而更深的体素树将空间分割地更加精细，而需要进行精确体素球碰撞的物体数减少。减少的精碰撞检测时间大于增加的粗碰撞检测时间。当体素树节点增加到一定数值时，增加的粗碰撞检测时间大于减少的精碰撞检测时间，总时间增加。即对于该模拟场景，在设定的体素树最大深度为 7 时，是一个平衡点，实现了最低的计算时间成本。

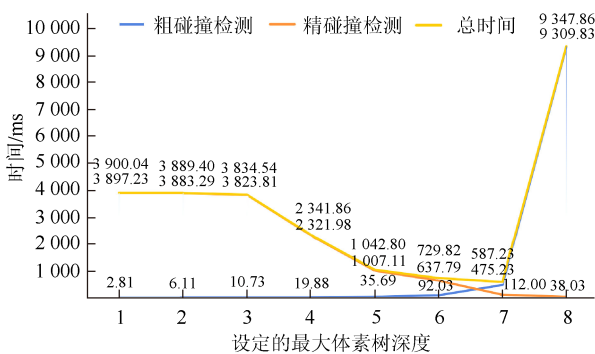


图 12 本文方法在不同深度下的计算时间

Fig. 12 The calculation time of the proposed method at different depths

2.5 根节点尺寸的敏感性

根节点的尺寸会影响生成体素树后的各个子节点的空间位置，进而影响碰撞检测时间，设置不同的根节点尺寸，可帮助分析其对碰撞检测时间的

影响程度。实验采用时间离散方式，设定检测时间间隔 Δt 为 0.1 s。实验总时间为 15.0 s，并记录其时间内包围盒生成、更新与碰撞检测所消耗的总时间(ms)。设定的体素树最大深度为 7，根节点坐标为(0, 0, 22.5)，实验结果见表 4。模拟场景使用根节点包围全部的建造资源时，根节点尺寸最小值为 50 m，当尺寸逐渐增大时，总碰撞检测时间先减再增，并不是尺寸最小时，碰撞检测时间越快。原因是当尺寸逐渐增大，且树深度相同时，尺寸大的子节点比尺寸小的子节点体积大，在资源体积保持不变，计算拥挤度值降低，达不到拥挤度阈值时，即停止划分子节点。尺寸大的子节点个数减少，从而导致粗碰撞检测时间减少。当尺寸逐渐增大时，最深层子节点的体积会更大，所包含的资源就会更多，从而增加资源之间体素球的遍历碰撞检测的速度，及精碰撞检测的时间。在尺寸为 65 m 时，两者达到一个平衡，总碰撞检测时间最少。

表 4 本文方法在不同根节点尺寸下的检测时间
Table 4 The detection time of the proposed method under different root node sizes

根节点尺寸/m	粗碰撞检测时间/ms	精碰撞检测时间/ms	总碰撞检测时间/ms
50	553.20	127.38	680.58
55	475.23	112.00	587.23
60	381.72	182.43	564.15
65	287.88	201.98	489.86
70	247.21	702.35	949.56
75	196.23	927.73	1 123.96

2.6 体素单元边长的敏感性

体素单元边长影响每个资源球状体素的个数与拟合精度，球状体素个数决定遍历次数，进而影响精碰撞检测时间，拟合精度影响碰撞检测精度。针对挖掘机与工人设定不同的体素单元边长组合，分析其对检测时间与碰撞检测精度的影响。研究采用时间离散方式，设定检测时间间隔 Δt 为 0.1 s。实验总时间为 15.0 s，并记录其时间内包围盒生成、更新与碰撞检测所消耗的总时间(ms)与 150 帧的碰撞检测结果。根节点坐标为(0, 0, 22.5)，尺寸为 55 m。为了更凸显体素单元边长对碰撞检测时间的敏感性，使用较浅的体素树结构，设定体素树最大深度为 3。实验结果见表 5，1~4 号实验为挖掘机与工人单元边长同步减小的测试结果。5~8 号实验为挖掘机的体素单元边长逐渐增加，工人的体素单元边长逐渐减少。

表 5 本方法在不同体素单元边长下的检测时间
Table 5 The detection time of the proposed method under different side lengths of voxel units

序号	体素单元边长/m		检测时间/ms
	挖掘机	工人	
1	1.2	1.2	212.98
2	0.9	0.9	373.59
3	0.6	0.6	909.87
4	0.3	0.3	8 532.30
5	0.3	1.2	6 169.54
6	0.6	0.9	397.23
7	0.9	0.6	454.79
8	1.2	0.3	2 713.62

由表 5 的 1~4 组数据可知, 随着体素单元边长的减小, 碰撞检测时间增加。这主要是因为球状体素数量增加, 精碰撞检测时间增加。同时, 由表 5 的 5~8 组数据可知, 当挖掘机的体素单元边长增大, 工人的体素单元边长减小时, 碰撞检测时间呈先减少再增加的趋势。但当工人的体素单元边长达到最小的 0.3 m, 挖掘机的体素单元边长达到最大的 1.2 m, 碰撞检测时间仍然小于第 5 组。可知, 大体积资源的体素单元边长大小对计算时间的影响更大。此外, 如图 13 所示, 当单元边长增大时, 对资源的拟合效果较差, 呈膨胀状, 但单元个数少。当单元边长减小时, 拟合效果好, 但单元个数急剧增多。图 14 给出了不同序号组的碰撞检测精度结果。由表 5 的 1~4 组数据可知, 当单元边长减小时, 碰撞检测的精确率、准确率都会增加, 误检率降低。由表 5 的 5~8 组数据可知, 当单元边长一大一小时, 就易出现漏检, 如图 11(a)所示。因此在资源相互靠近时采用边长较小的体素单元, 资源相互远离时采用较大的体素单元进行碰撞检测是有效且合理的, 可以达到保证碰撞检测精度的条件下, 增快碰撞检测速度。总之, 根据资源体积与距离动态地进行体素单元的计算生成是有必要的。

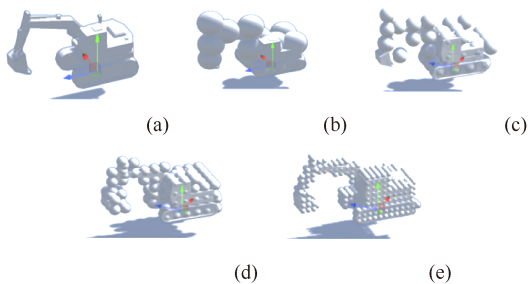


图 13 本方法在不同体素单元边长下示例

Fig. 13 Examples of the proposed method under different side lengths of voxel units ((a) Original view; (b) $N=1.2$ m, $N_x \cdot N_y \cdot N_z=29$; (c) $N=0.9$ m, $N_x \cdot N_y \cdot N_z=48$; (d) $N=0.6$ m, $N_x \cdot N_y \cdot N_z=135$; (e) $N=0.3$ m, $N_x \cdot N_y \cdot N_z=663$)

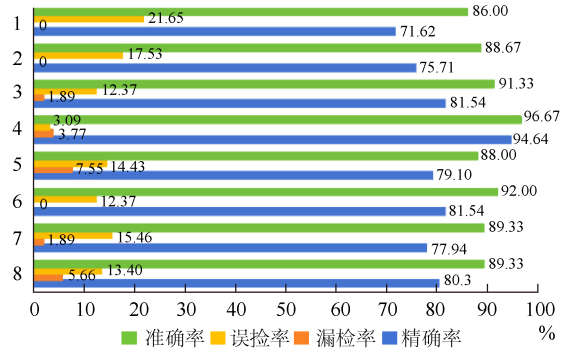


图 14 本方法在不同体素单元边长下的精度

Fig. 14 Accuracy of the proposed method under different voxel unit edge lengths

2.7 计算资源消耗分析

分析本方法在不同体量场景的资源消耗情况。设计大规模(根节点尺寸 214 m, 资源数量 152 个)、中规模(根节点尺寸 107 m, 资源数量 106 个)和小规模(根节点尺寸 55 m, 资源数量 56 个) 3 种情况的场景进行 CPU 资源与内存资源消耗的比较。CPU 资源的消耗核心指标是碰撞检测时间, 时间检测实验采用时间离散的方式, 树深度为 5~8, 根节点坐标为各场景中点。设定检测时间间隔 Δt 为 0.1 s, 实验总时间为 15.0 s, 记录其时间内包围盒生成、更新与碰撞检测所消耗的总时间(ms)。时间检测结果如图 15 所示, 当设定最大树深度为 8 时, 中规模与大规模场景碰撞检测时间减少到较低值, 而小规模场景在树深度为 7 时就达到碰撞检测时间的最小值。可知对于更复杂、更大规模的场景, 当最大树深度设定合适, 本文方法也具有较快的碰撞检测速度, 所消耗计算资源较少。

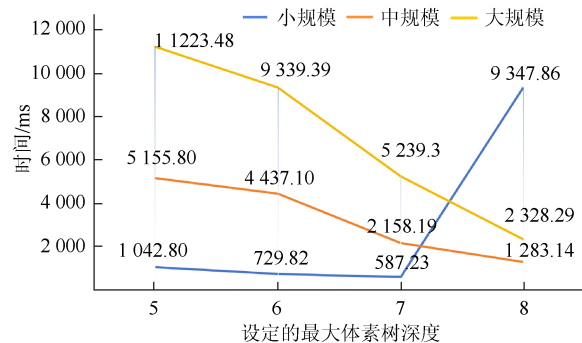


图 15 本方法在不同规模场景下的检测时间

Fig. 15 Detection time of the proposed method in scenarios with different scales

内存检测结果如图 16 所示, 当树深度较浅时, 内存占用成上升趋势, 升幅较小, 原因是体素树的生成增加内存占用, 但由于树深度较浅, 不能有效

地减少体素单元的数量。当树深度继续增加时,内存占用减少,由于体素单元数量减少,使得减少的内存占用大于树深度增加的内存占用。在树深度达到7时,内存占用开始快速增加,原因是体素树的分割开始极大地占据内存。当体素树深度达到8时,大规模场景的内存占用低于中规模场景,是因为拟建建筑的高度一般不会超过100 m,位于根节点高度107 m之上的体素子体将不会划分,减少了体素树分割所占用的内存。综上,本方法中体素树的分割占用的计算机内存最大,但只要设置合适的最大深度,在大规模场景中仍具有较少的资源消耗。

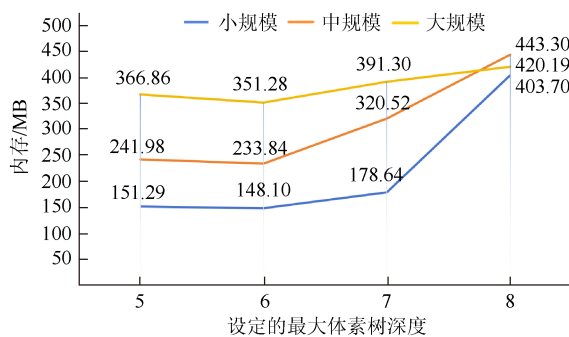


图16 本方法在不同规模场景下的内存占用

Fig. 16 Memory usage of the proposed method in scenarios with different scales

3 结论

传统的建造场景碰撞检测方法无法保证在高检测精度的前提具有较少的碰撞检测时间。本文提出了一种基于动态体素化的碰撞检测方法,融合空间动态体素树生成与资源动态球状体素化计算,构建了一种碰撞检测分析机制。主要具有以下优势。

本文通过结合拥挤度比率动态生成体素树,相较于八叉树全部分割到最大深度或依据物体数量来划分的方法,在保证碰撞检测时间的前提下,减少了计算机运行内存的占用,更加适合规模大且复杂的建造场景。依据物体自身体积大小与碰撞相对距离动态决定体素单元的边长大小,以生成体素单元进行精确的碰撞检测增强了碰撞检测的实时性与精度。另外,使用球状体素代替原立方体体素可消除碰撞检测时遇到的非轴对齐问题,结合空心体素化过程,加快了碰撞检测的速度。

此外,本研究还存在一些不足之处,如本方法仅适用于相对简化和理想化的实验环境中,且缺乏客观量化的动态风险评估体系,在实际应用中依赖

各类物联网传感器技术的支持。未来的实验测试,将融合强化学习与多源数据,建立动态风险评估机制,实现高风险目标的自动聚焦。并结合合适和建造场景的物联网传感器技术(如计算机视觉),在更为复杂、多变的实际场景中开展应用测试。

参考文献 (References)

- [1] DAGAN D, ISAAC S. Planning safe distances between workers on construction sites[J]. *Automation in Construction*, 2015, 50: 64-71.
- [2] VAN DEN BERGEN G. Collision detection in interactive 3D environments[M]. London: CRC Press, 2003: 1-277.
- [3] DINAS S, BAÑÓN J M. A literature review of bounding volumes hierarchy focused on collision detection[J]. *Ingeniería y Competitividad*, 2015, 17(1): 49-62.
- [4] 李占利, 付敬鼎, 李洪安, 等. 虚拟正畸治疗中的错位牙齿自动排列方法[J]. *图学学报*, 2019, 40(2): 225-234.
- [5] LI Z L, FU J D, LI H A, et al. Automatic alignment method for malocclusion in virtual orthodontics treatment[J]. *Journal of Graphics*, 2019, 40(2): 225-234 (in Chinese).
- [6] CHI S, CALDAS C H. Image-based safety assessment: automated spatial safety risk identification of earthmoving and surface mining activities[J]. *Journal of Construction Engineering and Management*, 2012, 138(3): 341-351.
- [7] KIM K, KIM M. RFID-based location-sensing system for safety management[J]. *Personal and Ubiquitous Computing*, 2012, 16(3): 235-243.
- [8] LUO X C, LI H, HUANG T, et al. A field experiment of workers' responses to proximity warnings of static safety hazards on construction sites[J]. *Safety Science*, 2016, 84: 216-224.
- [9] YANG B, ZHANG H R, SHEN Z R. Minimum distance calculation method for collision issues in lifting construction scenarios[J]. *Journal of Computing in Civil Engineering*, 2024, 38(6): 04024041.
- [10] HU S B, FANG Y H, GUO H L. A practicality and safety-oriented approach for path planning in crane lifts[J]. *Automation in Construction*, 2021, 127: 103695.
- [11] ZHU A M, ZHANG Z Q, PAN W. Crane-lift path planning for high-rise modular integrated construction through metaheuristic optimization and virtual prototyping[J]. *Automation in Construction*, 2022, 144: 104434.
- [12] 李世林, 李红军. 改进的最小包围球增量算法[J]. *图学学报*, 2016, 37(2): 166-171.
- [13] LI S L, LI H J. An improved randomized incremental algorithm for generating minimum enclosing ball of discrete point set[J]. *Journal of Graphics*, 2016, 37(2): 166-171 (in Chinese).
- [14] 王佳, 苏鼎丁, 周小平, 等. BIM 模型相似度计算方法[J]. *图学学报*, 2020, 41(4): 624-631.
- [15] WANG J, SU D D, ZHOU X P, et al. Similarity calculation method of BIM model[J]. *Journal of Graphics*, 2020, 41(4): 624-631 (in Chinese).
- [16] 柳有权, 李婉, 王愿超, 等. 基于碰撞检测强化的实时爆炸视图自动生成算法研究[J]. *图学学报*, 2019, 40(2): 235-239.
- [17] LIU Y Q, LI W, WANG Y C, et al. Real-time automatic generation algorithm of exploded view with collision detection enhancement[J]. *Journal of Graphics*, 2019, 40(2): 235-239 (in Chinese).
- [18] ZHOU Y, ZHANG E D, GUO H L, et al. Lifting path planning of mobile cranes based on an improved RRT algorithm[J].

- Advanced Engineering Informatics, 2021, 50: 101376.
- [15] LI Y, SHEN X K. A real-time collision detection between virtual and real objects based on three-dimensional tracking of hand[C]//2010 International Conference on Audio. New York: IEEE Press, 2010: 1346-1351.
- [16] TU C Q, YUAN L Z. Study on collision detection algorithm of hybrid bounding box[C]//2009 International Forum on Information Technology and Applications. New York: IEEE Press, 2009: 190-192.
- [17] GAN B Q, DONG Q P. An improved optimal algorithm for collision detection of hybrid hierarchical bounding box[J]. Evolutionary Intelligence, 2022, 15(4): 2515-2527.
- [18] 程琦甫. 融合深度神经网络的自碰撞检测算法研究[D]. 太原: 中北大学, 2020.
CHENG Q F. Research on self-collision detection algorithm combined with deep neural network[D]. Taiyuan: North University of China, 2020 (in Chinese).
- [19] 马晓萌, 孙红岩, 孙晓鹏. 层次八叉的三维模型并行碰撞检测[J]. 计算机工程与设计, 2019, 40(4): 1077-1084.
MA X M, SUN H Y, SUN X P. Parallel collision detection of 3D mesh using hierarchical octree[J]. Computer Engineering and Design, 2019, 40(4): 1077-1084 (in Chinese).
- [20] ZHU A M, ZHANG Z Q, PAN W. Developing a fast and accurate collision detection strategy for crane-lift path planning in high-rise modular integrated construction[J]. Advanced Engineering Informatics, 2024, 61: 102509.
- [21] 胡振中, 张建平, 张新. 基于四维时空模型的施工现场物理碰撞检测[J]. 清华大学学报(自然科学版), 2010, 50(6): 820-825.
HU Z Z, ZHANG J P, ZHANG X. Construction collision detection for site entities based on 4-D space-time model[J]. Journal of Tsinghua University (Science & Technology), 2010, 50(6): 820-825 (in Chinese).
- [22] 张建平, 李丁, 胡振中. 一种集成空间分解与占用的精确碰撞检测算法及其在建筑工程中的应用[J]. 工程力学, 2014, 31(5): 79-85.
ZHANG J P, LI D, HU Z Z. An accurate collision detection algorithm integrating spatial decomposition and spatial occupancy and its applications in constructions[J]. Engineering Mechanics, 2014, 31(5): 79-85 (in Chinese).
- [23] LI P, WANG Q K, GUO Z, et al. Identifying falling-from-height hazards in building information models: a voxelization-based method[J]. Journal of Construction Engineering and Management, 2022, 148(2): 04021203.
- [24] 谢震鹏. 基于深度学习和视觉 SLAM 的语义地图构建及碰撞检测研究[D]. 南宁: 广西大学, 2023.
XIE Z P. Research on semantic map construction and collision detection based on deep learning and visual slam[D]. Nanning: Guangxi University, 2023 (in Chinese).
- [25] 徐进, 柳宁, 李德平, 等. 一种基于抓取簇和碰撞体素的工业零件抓取姿态检测算法[J]. 机器人, 2022, 44(2): 153-166.
XU J, LIU N, LI D P, et al. A grasping poses detection algorithm for industrial workpieces based on grasping cluster and collision voxels[J]. Robot, 2022, 44(2): 153-166 (in Chinese).
- [26] LIN X, HAN Y, GUO H L, et al. Lift path planning for tower cranes based on environmental point clouds[J]. Automation in Construction, 2023, 155: 105046.
- [27] 于瑞云, 赵金龙, 余龙, 等. 结合轴对齐包围盒和空间划分的碰撞检测算法[J]. 中国图象图形学报, 2018, 23(12): 1925-1937.
YU R Y, ZHAO J L, YU L, et al. Collision detection algorithm based on AABB bounding box and space division[J]. Journal of Image and Graphics, 2018, 23(12): 1925-1937 (in Chinese).
- [28] QU H Y. Research on parallel algorithm based on AABB bounding box coordinate chain method[J]. Journal of Physics: Conference Series, 2021, 1732: 012067.
- [29] HU A L, HE Y Y. Research on hybrid collision detection algorithm based on separation distance[J]. Journal of Physics: Conference Series, 2022, 2258: 012011.
- [30] LU S N, XU Z D, WANG B R. Human-robot collision detection based on the improved camshift algorithm and bounding box[J]. International Journal of Control, Automation and Systems, 2022, 20(10): 3347-3360.