

文章编号 1004-924X(2015)03-0895-09

## 大规模地下空间在线展示关键技术

刘小军,张东培,谢 宁,贾金原\*  
(同济大学 软件学院,上海 201804)

**摘要:**简述了通过网页 3D 来分享信息及进行 3D 交互的相关技术。针对大规模的地下空间场景以及对该场景进行网页实时展示的需求,提出了一套基于互联网进行大规模地下空间实时展示的轻量级场景管理机制。首先,对地下空间场景数据进行轻量化预处理,构建轻量级场景管理结构。然后,根据场景数据特点提出相应的场景管理策略,即基于兴趣区域的室外场景管理策略和基于入口的室内场景管理策略。通过视点位置驱动实现室内外场景切换,并自动选择管理策略,达到在网页上进行远程实时漫游大规模地下空间场景的效果。实验表明:该方法能大幅度剔除不必要的场景数据,提高资源利用率,满足在线展示大规模地下空间场景的需求。由于网页应用具有跨平台优势,提出的系统支持用户跨平台对大规模地下空间场景进行实时漫游与操控。

**关键词:** Web3D; 地下空间; 场景管理; 在线展示; 轻量化

**中图分类号:** TP391.9 **文献标识码:** A **doi:** 10.3788/OPE.20152303.0895

### Key technologies for online visualization of large-scale underground space

LIU Xiao-jun, ZHANG Dong-pei, XIE Ning, JIA Jin-yuan\*

(School of Software Engineering, Tongji University, Shanghai 201804, China)

\* Corresponding author, E-mail: jyjia@tongji.edu.cn

**Abstract:** The corresponding technologies for sharing information and 3D interaction by Web3D were described briefly. An efficient scene management scheme for real-time visualization of a large-scale underground scene was proposed to meet its demands for webpage visualization. Firstly, the raw data from the underground scene were preprocessed in lightweight, and a shell structure-aware scene analysis algorithm was employed for constructing a scene management structure. Then, some strategies were put forward according to the characteristics of the underground scene data, and those are a SOI-ExteriorShell strategy based on Sector Of Interest (SOI) management for outdoor scenes and a Portal-InteriorShell strategy supporting progressive loading for indoor ones. With the switch between indoor and outdoor scenes, the method dynamically chooses the corresponding strategy and makes it possible to roam in the large-scale underground scene online. From the final experiment, it concludes that the proposed method efficiently culls unnecessary scenes in a greatly large extent and is capable of meeting the requirements for real-time visualization of the large-scale underground scene over Internet. As the webpage has an advantage of cross-platform, the system supports users to roam

**收稿日期:** 2014-11-18; **修订日期:** 2014-12-31.

**基金项目:** 国家自然科学基金面上项目(No. 61272276); 国家十二五计划重大科技支撑项目(No. 2012BAC11B00-04-03); 高等学校博士学科点专项科研基金资助项目(No. 20130072110035); 吉林省重点科技攻关项目(No. 20140204088GX); 长春高新区"长白慧谷"英才计划资助项目(No. 3-2013006)

and control the large-scale underground scene under cross-platform.

**Key words:** Web3D; underground space; scene management; online visualization; lightweight

## 1 引言

随着网络技术的蓬勃兴起,通过 3D 在线应用进行信息共享以及协同交互已受到诸多行业的青睐,同时也出现了更复杂的需求,需要处理的数据量也更多。然而,由于目前网页浏览器能力的限制,其 3D 显示等各方面性能还无法与 PC 端软件相比。在此情况下,针对规模庞大且结构复杂的 3D 场景,如何使用合适的策略,使得网页端能处理远远超出系统承载量的大规模 3D 场景,已成为极具挑战性的问题。

目前,市场上主流的虚拟展示系统中普遍采用场景图对整个场景进行组织<sup>[1]</sup>。然而,这些场景图大都面向 PC 软件或需要安装第三方客户端,无法满足移动互联网上跨平台应用的需求。

本文主要针对大规模的地下空间场景及该场景的网页实时展示进行研究。结合地下空间场景的特点,提出一整套有效的方案,包括:原始数据的轻量化预处理、基于体结构感知的场景结构分析、基于层次化网格的室外场景管理等。由于网页应用自身的跨平台优势,本系统支持用户跨平台对大规模地下空间场景进行实时漫游与操控。

地下空间场景建筑是指由人工挖掘建造的位于地表层以下的建筑,包括复杂构筑物和简单空间对象。构筑物由外体、内体和连通体组成,其结构比较复杂,如空间结构复杂且包含大量附属公共设施的地下车站等。简单空间对象虽较清晰,但数量庞大,如众多类型的管线等。构筑物的外体是指由外表面形成的几何体;内体位于外体内部,表示房间等内部基本空间;连通体则用于连接多个体对象或构筑物,如楼梯、走廊和通道等。在几何结构上,体的结构通常相似,由表示顶、底和侧的多边形面组成其外壳,内部则可能存在附属对象等,这些附属对象同样由多边形面组成。每个构筑物中包含一个外体、多个内体和多个连通体,体与体之间可能存在连接关系,这种关系用一对包含相同连接标识的多边形面表示,这些成对的多边形面又称为连面。

与地表场景相比,地下空间场景不仅结构复

杂,也有其自身的特点。由于施工难度等多方面原因,地下空间场景的纵向范围往往在地下 30m 以内,远远小于横向范围。而且数量庞大且种类繁多的管线数据也是地表场景所没有的。

要实现大规模地下空间场景的三维在线展示,主要面临如下挑战:

(1)场景结构复杂。地下空间建筑数据主要由多边形面组成,面的数量成千上万。逻辑上,这些多边形面间是存在某种空间关系的,如两个面之间是否相邻。但实际中却缺少这样的关系数据,致使整个场景的结构复杂且难于管理。因此,使用何种场景管理结构能更加有效地组织结构复杂的地下空间场景,提高数据存储、查询、加载等效率是场景管理时需要考虑的首要问题。

(2)数据冗余度高。一方面,现有数据中多边形面间的共用顶点重复出现,造成数据冗余;另一方面,地下空间建筑的体数据中有一大部分呈拉伸状态,完全可以通过几何运算来对该部分数据进行轻量化,减少后续处理步骤中需要处理的数据量。

(3)难于在网页端实时渲染。大规模地下空间场景的数据量远远超出网页应用的最大承载量,无法满足在线实时展示的需求。此时,采用何种场景剔除策略显得尤为重要。好的剔除策略能在不影响用户体验质量的前提下,剔除大部分不必要的数据库。

## 2 相关研究工作

当前已出现较多与场景图相关的研究,如早期的 DIVE、Blue-c、VRML 以及 Java3D 等。这些技术均把场景图作为图形系统的重要组成部分,在很大程度上提高了场景管理的性能,但仍存在模块独立性低和使用不够灵活等缺点<sup>[2-5]</sup>。OSG(Open Scene Graph)等对此进行了扩充,将文件读取、转换和绘制等功能以插件形式进行封装,提高了模块独立性<sup>[6]</sup>。但其使用的公用节点、指针引用等方法又致使场景节点过于庞大,维护较困难。

在可见性剔除方面, Cohen-Or 等预先把场

景剔除方式进行分类,再根据场景实际情况选择合适的方式来进行场景剔除<sup>[7]</sup>;普建涛等给出可见性判断算法的基本步骤和算法优劣判别标准<sup>[8]</sup>;Zheng等通过射线检测方式有效剔除复杂模型的不可见区域<sup>[9]</sup>,有效提高了场景剔除效果。

汪国平等、李静等依次通过场景和资源独立管理<sup>[10]</sup>、递归网格结构<sup>[11]</sup>(Optimized Recursive Grids, ORG)等方法优化场景结构组织;贾金原等则进一步引入扇形兴趣区域<sup>[12]</sup>(Sector of Interest, SOI)来简化或管理整个场景,提高场景管理效率。

当然也出现了其他相关方面的进展,如孙军华等<sup>[13]</sup>、周志勇等<sup>[14]</sup>均对点集配准算法进行研究,为场景中重复对象的检测等提供了新思路;Deng等<sup>[15]</sup>则讨论了分布式虚拟环境的动态加载及延时调整,考虑了实时加载和网络延时的折衷方法。

以上方法虽解决了PC端虚拟场景管理的诸多问题,但较少考虑支持网页三维(Web3D)的大规模场景实时展示。因此,随着智慧城市、大数据等相关技术的发展,通过网页端共享大规模3D场景的需求也更加强烈,迫切需要结合场景数据和网页应用的特点,提出一套适合在网上实时互动展示大规模地下空间场景的解决方案。

### 3 技术路线

大规模地下空间在线展示系统分为4部分(如图1所示):原始模型结构重组、场景数据轻量化、场景管理结构构建和场景管理策略设计。其中前3部分主要通过后台预处理方式预先完成,以减少第4部分中前端网页的实时数据处理量及计算量。具体如下:

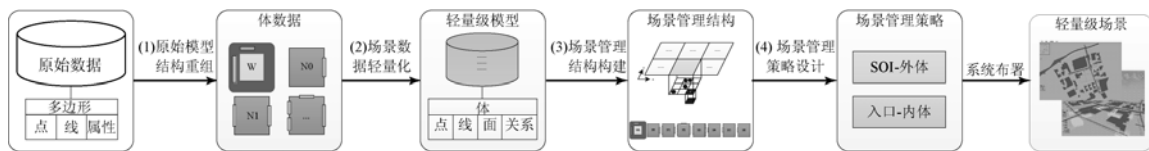


图1 系统流程图

Fig. 1 Flowchart of proposed system

(1)原始模型结构重组:针对地下建筑空间结构复杂等特点,先将整个场景统一为子场景和单位模型两类节点。通过结构感知的结构重组算法生成并记录模型间的所属关系,使后续处理中能快速准确地进行不必要数据的剔除等操作(详见4.1章)。

(2)场景数据轻量化:在形成的体数据中通过建立点索引、拉伸简化等方式去除冗余数据,实现数据的轻量化(详见4.2章)。

(3)场景管理结构构建:为了更好地支持室外、室内漫游,从室外、室内两方面出发构建场景管理结构。室外通过层次化网格来进行组织,室内则采用以体为单位的构筑物空间结构图。两种结构相结合,组成整个场景管理结构(详见第5章)。

(4)场景管理策略设计:在场景管理结构的支持下,引入室外、室内两种场景管理策略。室外时首先剔除内体,再通过SOI-外体管理策略将当前处理的场景范围限制在有限的视角范围内;室内时不仅只显示当前构筑物中的内体,还采用入口-内体管理策略进一步剔除与当前体无关的内体数

据。最后,通过视点位置驱动进行室外、室内场景切换,实现整个场景的即时漫游(详见第6章)。

### 4 场景数据预处理

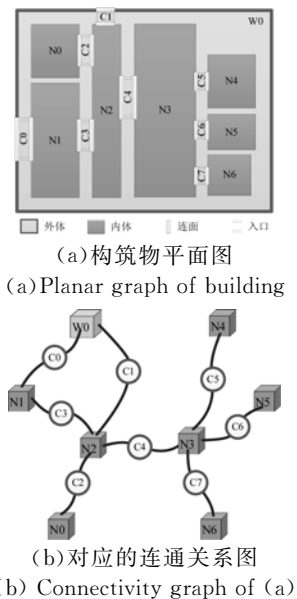
地下空间场景的预处理过程主要包括两部分:场景模型关系的生成和场景数据的轻量化。

#### 4.1 基于结构感知的模型关系生成算法

地下空间场景数据主要由地下构筑物和管线数据组成。构筑物内部结构复杂,管线虽然结构简单却数量庞大。为了统一管理这些数据,引入子场景和单位模型两类场景节点,故整个场景由多个子场景组成,而每个子场景内又包含多个单位模型。将构筑物和管线数据分别统一到这两类节点中,并记录它们的内部关系,为构建场景结构做准备。

对于结构相似但复杂的地下构筑物,以子场景节点进行处理。通过空间结构分析方法将其分解成数据量较小的、空间关系相对独立的体作为其单位模型。图2(a)为包含1个外体(记为W0)

和 7 个内体 (N0 到 N6) 的某构筑物平面图, 图 2(b) 为对应的连通关系图, C0 到 C7 则表示体之间的入口, 每个入口中包含一对连面。图 3 则为抽取出的体数据。可见统一后的体数据中不仅包含当前体的几何数据, 还记录了当前体的连接信息。



(b) Connectivity graph of (a)

图 2 地下构筑物示例

Fig. 2 Example of underground building



图 3 结构抽取并统一后的体数据

Fig. 3 Shells extracted from underground building

对于结构简单但数量庞大的地下管线, 则当作单位模型来处理, 考虑到其空间形态呈线状且总是与路段绑定等特点, 将路段作为子场景节点, 位于该路段下的管线对象都被当作其单位模型。

#### 4.2 基于拉伸简化的轻量化算法

前期生成的体数据中仍存在较多的冗余, 如多个面共边时重复出现的顶点数据、拉伸体中包含的拉伸顶点等。本文通过建立点索引、几何运算生成拉伸顶点等方式对场景数据进行轻量化处理。图 4 为通过几何运算轻量化拉伸体的过程, 式(1)表示了拉伸体轻量化前后的顶点数量关系。

$$V' = V / (2k) + n, \quad (1)$$

其中:  $V$  表示轻量化前的顶点数;  $V'$  为轻量化后的顶点数;  $k$  为顶点重复度;  $n$  则代表拉伸向量个数。

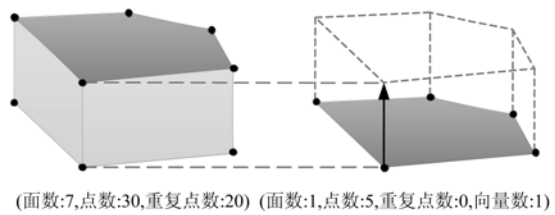


图 4 拉伸体简化为底面和拉伸向量的情况

Fig. 4 Simplification for extruded shell

根据式(1)可知, 轻量化程度与顶点的重复度成正比, 重复度越大, 轻量化程度也越高。图 4 中, 重复度为 3, 使得轻量化后数据量约为轻量化前的 1/6, 取得了较好的轻量化效果。

数据预处理在降低场景数据冗余度的同时, 还整理出整个场景的所有子场景节点及其单位模型节点。配合整个室外场景管理结构给每个子场景赋予 ID, 进而实现大部分的场景管理操作, 无需每次都加载场景数据, 从而大大减少了数据的访问次数, 为后续场景管理策略的实施提供必要的支持。

## 5 场景管理结构的构建

在实际应用的大规模场景中, 受视点位置、视域范围等的影响, 当前实际需要处理的场景区域往往是整个场景的一小部分。本文综合考虑室外和室内两种模式, 构建了合适的场景管理结构, 为快速确定尽可能小的当前处理区域提供支持。

### 5.1 基于层次化网格的室外场景管理结构

场景管理结构主要以二维的平面网格和三维的八叉树等为主, 然而地下空间场景的纵向空间范围远小于横向, 为了降低场景管理结构的复杂度, 采用层次化网格来组织室外场景。

目前已存在一些较优秀的网格划分方法, 如文献[11]中的网格划分算法具有计算效率高、划分时间短等优点, 但该类算法大部分都以优化算法本身的计算复杂度或空间复杂度等为目的, 较少考虑输入数据本身的特性, 更不讨论场景管理结构构建等过程。因此, 本文针对这 3 点对其做进一步扩展:

(1) 在体数据的支持下, 不需要所有模型的三角面均参与计算, 只需外体的底面(或路段面)参与即可, 从而不仅保证了计算精度更降低了计算成本。

(2) 将场景管理结构以场景描述文件的形式

输出。其中记录子场景 ID 或描述文件名称,通过该文件可进行大面积的室外场景剔除,大大减少了前端网页展示时所处理的数据量,更为场景必要数据的动态加载等提供支持。

(3)将网格划分算法扩展到支持网格绑定对象的添加、删除及更新等操作,做到场景子模型变化时场景管理结构的动态更新。

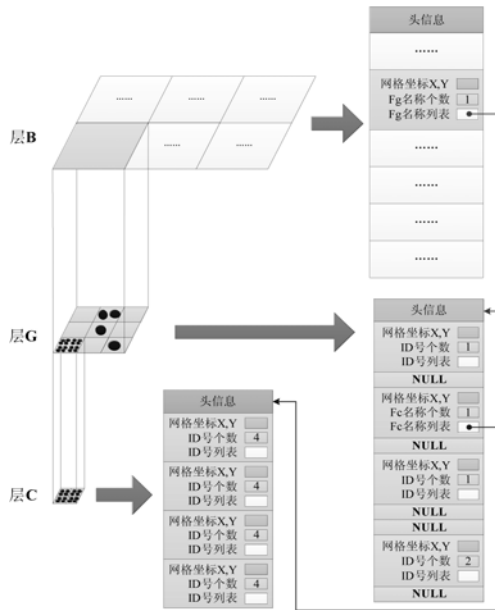


图 5 室外场景管理结构示意图

Fig. 5 Structure for outdoor scene

本文构建的室外场景管理结构如图 5 所示,在该结构中,网格层 B 主要绑定的是网格层 G 生成的场景描述文件名称。目的为控制当前加载的场景管理结构信息量,防止加载远远超出视域范围、暂时不会用到的场景管理结构信息。网格层 G 为后续漫游时实施兴趣区域、场景调度及碰撞检测区域等判断的主要网格层;层 C 则是为了防止在极端情况下,过多小尺寸子场景集中于同一网格单元中,导致该单元记录信息急剧增大,增加实时调度难度,使用 C 可以在不影响其他网格单元的基础上,集中处理该复杂单元,从地下建筑场景的特点来看,出现该极端情况的机率较小。

扩展后的场景管理结构构建过程见算法 1,算法 2 则描述了子场景删除时的算法过程。

算法 1:基于网格划分的室外场景管理结构生成算法

输入:预处理后的模型数据, MaxID, MaxCell 和 M

输出:基于层次化网格结构的室外场景描述文件

1. begin
2. 计算场景横向范围;
3. 用文献[11]中的方法生成网格层 G;
4. 绑定所有模型到 G 网格层并记录到文件 Fg;
5. for 所有 Fg 中的网格单元 g do
6. if g 中的 ID 号个数 > MaxID then
7. 创建新的场景描述文件 Fc;
8. 以 g 为场景按[11]中的方法生成新的网格层 C;
9. 重新绑定 g 中的模型到网格层 C 中并记录到文件 Fc;
10. 用 Fc 文件名替换 Fg 中原 g 网格单元绑定的所有 ID 号;
11. end if
12. end for
13. if Fg 中的网格单元个数 > MaxCell then
14. 新的分辨率 = 层 G 的分辨率 / 10<sup>2M</sup>;
15. 通过新的分辨率创建网格层 B;
16. 根据新的分辨率将 Fg 分割成多个较小的文件;
17. 将分割后的小文件名绑定到层 B 并记录到文件 Fb;
18. end if
19. end

其中参数 MaxID 表示单个网格中允许容纳的最多 ID 号个数;MaxCell 表示单个场景描述文件中允许包含的最多网格数;M 则为调整 B 层网格分辨率的参数信息。

算法 2:子场景自动删除算法

输入:子场景 ID 及其位置信息,室外场景描述文件输出;删除子场景后的室外场景描述文件

1. begin
2. for 所有的 Fb 文件 do
3. 定位当前子场景所在的 Fg 文件名;
4. if 返回为真 then
5. 转到第 9 行;
6. end for
7. if 定位失败 then
8. 转至第 22 行;

9. 根据当前  $F_g$  的分辨率定位子场景所在的网格单元;
10. for 所有定位到的网格单元 do
11. 查找指定 ID 是否被绑定到当前网格单元;
12. if 返回为真 then
13. 删除该 ID 号;
14. 转至第 22 行;
15. if 查找到  $F_c$  文件 then
16. 转至第 20 行;
17. end for
18. if 查找失败 then
19. 转至第 22 行;
20. if 查找到  $F_c$  then
21. 在  $F_c$  中重复第 9 到 17 行的过程;
22. end

### 5.2 以体为单位的室内场景管理结构

室内场景管理主要用于地下构筑物,在此采用基于入口的室内场景管理策略(具体见 6.2),其主要思想是只处理构筑物中视点所在的体及其相关的体数据,所有不属于该范围的数据均被剔除。而本文在前期的数据预处理过程中,已进行了多方面的分析,生成的体数据中已包含了足够的室内场景管理结构信息(见图 2 和图 3),能全面支持后续策略的实施。因此,直接使用该数据即可满足室内场景的漫游需求。

## 6 场景管理策略

为了达到更好的效果,在场景管理结构的基础上,本文提出相应的场景管理策略与其配合。

### 6.1 SOI-外体室外场景管理策略

在进行大规模地下空间场景的室外漫游时,一方面由于外体对内体的遮挡,在抽取到体数据后,可通过只处理外体数据的方式剔除大部分结构复杂的内体数据;另一方面,受限于视野范围等因素,往往只对视域范围内的场景区域发生兴趣,对兴趣区域<sup>[12]</sup>的有效管理可进一步大幅过滤不必要的场景数据信息。

在基于层次化网格的室外场景管理结构的支持下,根据视点位置在  $G$  网格层上进行 SOI 定位(如图 6(a)所示),只处理 SOI 内的模型数据,大大提高了系统资源的利用率。其中网格单元结构体定义如下:

```

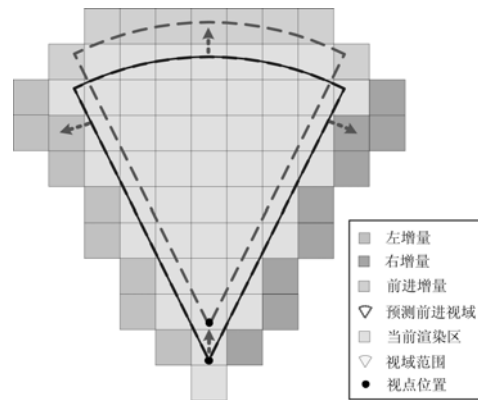
struct Cell
{
long x, y; //网格坐标
vector<string> vIDs;//绑定的 ID 号
}

```

对于视域范围内的模型,为了与真实世界保持一致,总是优先加载和渲染离视点  $v$  较近的场景模型,在确定视域范围的网格单元集  $S$  时,同时给每个网格单元赋予一定的加载优先级  $w$ (计算公式见式(2)),优先加载  $w$  较小的网格单元,计算结果及加载扩展过程如图 6(b)所示。

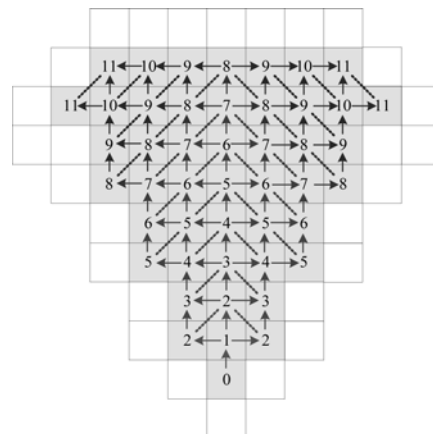
$$w = |s.x - v.x| + |s.y - v.y|, \quad (2)$$

其中:  $s \in S$ 。



(a) 兴趣区域示意图

(a) Diagram of sector of interest



(b) 兴趣区域中的网格优先级

(b) Grid priorities in sector of interest

图 6 扇形兴趣区域

Fig. 6 Sector of interest

进行场景漫游时,图 6 中的视域范围将沿着漫游方向移动,系统将根据漫游方向对视域中的

增量区进行预测,并按计算的优先级预先加载相应的网格单元,同时在移动完成后卸载已不在数据缓冲区的网格单元,做到数据的及时更新。

### 6.2 入口-内体室内场景管理策略

室内漫游与室外不同,受所在体侧面等的遮挡,采用入口-内体策略进行室内场景管理,部分思路得益于文献[7]。漫游过程见图 7,为了描述清晰,图中假设视点从图 2 中的内体 N2 出发,通过入口 C3 并进入内体 N1 中。

整个漫游过程使用多线程参与,用于算法控制、模型渲染和数据加载。其中模型渲染和数据加载线程的行为由当前体中的入口对象进行驱动,当视点到入口的距离小于某临界值时,调用数据加载线程加载该入口对面的体数据,并在加载完成后调用模型渲染线程进行渲染。通过视点与连面的碰撞检测、视点与入口之间的距离来判断后续要执行的操作。

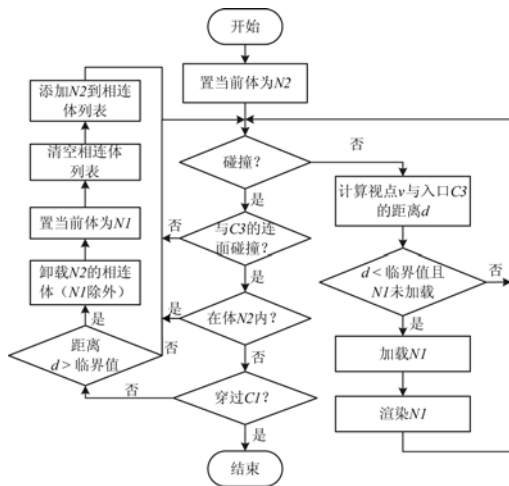


图 7 室内场景管理流程

Fig. 7 Flowchart of indoor scene management

为了避免视点在入口附近徘徊导致反复装/卸载同一体数据,根据漫游速度、数据加载速度等引入距离临界值,根据视点到入口的距离与该临界值的比较结果来选择加载(即小于)或卸载(即大于)相应的体数据。同时在进入某个体时先判断与该个体相连的体的个数,若个数等于 2,则直接加载并渲染对应体数据,简化图 7 中的判断流程。

在引入上述策略的同时,还通过其他方式来减少资源损耗,如对管线检修井等呈现相同几何外形的场景对象采用实例化渲染;对外体、管线等对象采用细节层次转换策略,根据对象与视点的距离,选择细节合适的模型进行渲染。

## 7 实验

为了验证本文方法的有效性,通过仿真试验进行性能测试,分别选择不同数量、不同大小的一系列原始数据参与试验。试验中采用的计算机配置为: Intel i5-3210M 处理器、2.0 GB 内存、NVIDIA GeForce 610M 显卡及 64 位 Windows 8 操作系统;后台预处理用 C++ 语言实现,前端网页展示采用 Flash3D 技术,使用 ActionScript 3.0 语言实现;运行过程中设置的参数如下:兴趣区域半径等于 10 倍的 G 层网格单元长度、视角大小为 60°、每个网格单元中的平均模型数为 8、最大刷新率为 30 frame/s。通过记录并统计部分运行过程中的性能参数,得到如表 1 和图 8 所示的结果。图 9 则显示了仿真实验的部分效果图。

表 1 室外场景漫游实验结果对比

Tab. 1 Performance comparison of outdoor scenes with and without our method

模型个数	T w/o (10 <sup>4</sup> )	T w (10 <sup>4</sup> )	M w/o (M)	M w (M)	FPS w/o (frame/s)	FPS w (frame/s)
100	35.38	0.76	329.8	64.8	28.3	30.4
200	69.52	0.98	672	78.6	13.4	30.1
500	170.94	1.12	1300	88.1	3.3	29.6
1 000	353.21	1.2	—	87.8	—	30.1
1 500	526.62	1.18	—	88.4	—	29.9
2 000	705.72	1.24	—	88	—	30.6
2 500	884.26	1.26	—	88.2	—	29.8
3 000	1 030.15	1.23	—	88.4	—	30.2

T:三角形数量 M:内存损耗  
w/o:未使用本文方法 w:使用本文方法

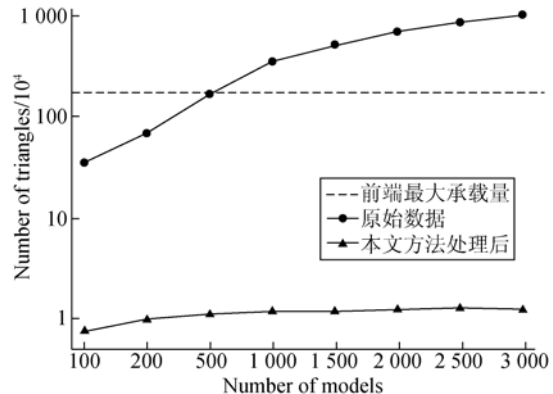


图 8 室外场景剔除结果统计

Fig. 8 Statistic of culling face in outdoor scene

从表 1 可以看出,在采用本文方法前,随着加

载模型数量的增加系统的刷新率迅速下降。当加载模型达到 500 个时系统出现不稳定,甚至死机现象。而使用本文方法后,尽管模型数量增加到 3 000 个,刷新率始终保持在最大刷新率 30 frame/s 左右,且类似的结果同样出现在内存消耗对比中,这充分说明本文方法的有效性。

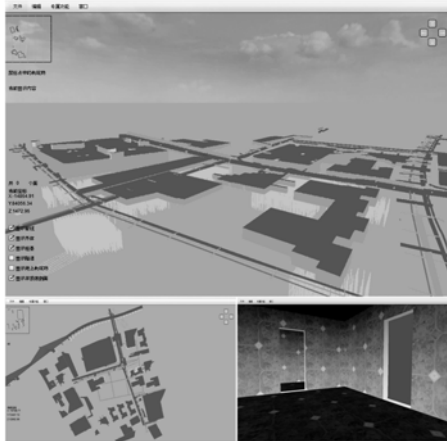


图 9 仿真实验效果图

Fig. 9 Snapshots from experiment demo

图 8 是注重体现室外场景策略的实施效果。经结果统计和分析,室外场景策略的实施使得前端网页展示时整个场景的三角面数呈指数级下降,且在模型个数接近 650 时,剔除后的整个场景三角面数已开始趋于收敛并保持稳定,使得系统

当前需要处理的数据量与场景规模呈弱相关状态。这一方面说明前期构建的场景管理结构对场景策略的实施起到了较好的支持作用;另一方面也体现了整个策略的设计较好地结合了地下空间场景数据的特点。

## 8 结 论

本文从地下空间场景的数据特点等出发,在统一场景节点的基础上,通过改进和简化部分算法,构建了合理的场景管理结构。通过进一步引入合适的场景管理策略,去除冗余数据、剔除不必要的场景区域等方式对整个地下空间场景进行了轻量化。最后的实验统计结果表明该方法及其构建的场景管理结构适合于地下空间场景管理,大幅度降低了前端网页展示时的系统资源开销,提高了计算效率。

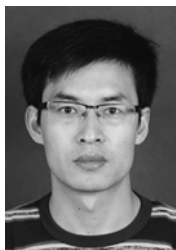
当然,本文在某些方面还存在一定的不足,如现有的轻量化算法虽对拉伸体等体数据有很好的轻量化效果,但还不适用于处理其他特征的场景数据,导致该结果的主要原因是未能准确把握该数据的几何特征,后续将对这些方面进行分析或从另外的角度寻找合适的轻量化算法,进一步降低前端网页展示的负担。

## 参考文献:

- [1] SOWIZRAL H. Scene graphs in the new millennium [J]. *IEEE Computer Graphics and Applications*, 2000, 20(1): 56-57.
- [2] FRÉCON E, STENIUS M. DIVE: A scalable network architecture for distributed virtual environments [J]. *Distributed Systems Engineering*, 1998, 5(3): 91-100.
- [3] NAEF M, LAMBORAY E, STAADT O, et al.. The blue-c distributed scene graph [C]. *Proceedings of the Workshop on Virtual Environments*, Zurich, Switzerland, 2003:125-133.
- [4] TRAMBEREND H. Avocado: A distributed virtual reality framework [C]. *Proceedings of IEEE Virtual Reality*, Los Alamito, CA, 1999: 14-21.
- [5] The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1 [EB/OL]. 1997, [2014-9-14]. [http://www.web3d.org/fs\\_technicalinfo.htm](http://www.web3d.org/fs_technicalinfo.htm).
- [6] BURNS D, OSFIELD R. Tutorial: Open scene graph A: introduction tutorial; open scene graph, B: Examples and applications [C]. *Proceedings of IEEE Virtual Reality*, Washington, DC, 2004: 265.
- [7] COHEN-OR D, CHRYSANTHOU Y L, SILVA C T, et al.. A survey of visibility for walkthrough applications [J]. *IEEE Transactions on Visualization and Computer Graphics*, 2003, 9(3):412-431.
- [8] 普建涛, 查红彬. 大规模复杂场景的可见性问题研究[J]. *计算机研究与发展*, 2005, 42(2): 236-246. PU J T, ZHA H B. Research on visibility for large-scale and complex scenes [J]. *Journal of Computer Research and Development*, 2005, 42(2):236-246. (in Chinese)

- [9] ZHENG Z, PRAKASH E, CHAN T K Y. Interactive view-dependent rendering over networks [J]. *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14(3):576-589.
- [10] 汪国平, 李胜, 王少荣, 等. ViWOSG: 超大规模分布式虚拟环境中的分布式场景图[J]. *中国科学 (F辑: 信息科学)*, 2009, 39(1): 96-106.  
WANG G P, LI SH, WANG SH R, *et al.*. ViWoSG: A distributed scene graph of ultra-massive distributed virtual environments [J]. *Science China: Information Sciences*, 2009, 39(1): 96-106. (in Chinese)
- [11] 李静, 王文成. 线性复杂度的网格优化划分[J]. *软件学报*, 2011, 22(10): 2488-2496.  
LI J, WANG W CH. Optimizing grid construction in linear complexity [J]. *Journal of Software*, 2011, 22(10): 2488-2496. (in Chinese)
- [12] 贾金原, 王伟, 王明飞, 等. 基于多层增量式可扩展扇形兴趣区域的大规模 DVE 场景对等渐进式传输机制 [J]. *计算机学报*, 2014, 37(6): 1324-1334.  
JIA J Y, WANG W, WANG M F, *et al.*. Multi-Layered Incremental & Scalable Sector of Interest (MISSOI) based efficient progressive transmission of large-scale DVE scenes [J]. *Chinese Journal of Computers*, 2014, 37(6):1324-1334. (in Chinese).
- [13] 孙军华, 射萍, 刘震, 等. 基于分层块状全局搜索的三维点云自动配准[J]. *光学精密工程*, 2013, 21(1):174-180.  
SUN J H, SHE P, LIU ZH, *et al.*. Automatic 3D point cloud registration based on hierarchical block global search [J]. *Opt. Precision Eng.*, 2013, 21(1):174-180. (in Chinese)
- [14] 周志勇, 薛维琴, 郑健, 等. 基于 t 分布混合模型的非刚性配准算法[J]. *光学精密工程*, 2013, 21(9):2405-2420.  
ZHOU ZH Y, XUE W Q, ZHENG J, *et al.*. Point set non-rigid registration using t-distribution mixture model [J]. *Opt. Precision Eng.*, 2013, 21(9):2405-2420. (in Chinese)
- [15] DENG Y, LAU R W H. On delay adjustment for dynamic load balancing in distributed virtual environments [J]. *IEEE Transactions on Visualization and Computer Graphics*, 2012, 18(4): 529-537.

#### 作者简介:



刘小军(1981—),男,山西吕梁人,博士研究生,2008年于中国矿业大学(北京)获得硕士学位,主要研究方向有计算机图形学,计算几何,3D模型处理等。E-mail: xjliu1204@126.com。

#### 导师简介:



贾金原(1963—),男,山东乐陵人,教授,博士生导师,2004年于香港科技大学获得博士学位,现为同济大学软件学院图形图像研究中心主任,主要从事 Web Graphics、虚拟现实、几何造型、游戏引擎等方面的研究。E-mail: jyjia@tongji.edu.cn。

(版权所有 未经许可 不得转载)