



基于 Vega Prime 的“数字港口” 三维可视化系统与构架设计

王 艺, 李彩霞, 李 智, 程小林
(中国卫星海上测控部, 江苏 江阴 214431)

摘要: 结合基于 Vega Prime 的“数字港口”三维可视化系统, 分析了系统构架设计和场景数据库优化管理, 针对三维重建可视化效果和运行效率之间的矛盾, 提出了系统运用的图形数据动态调度、动态 LOD (Levels of Detail)、实时软阴影优化等关键技术, 并通过试验证明了该方法能够在系统流畅性和高仿真性之间获取较好平衡。

关键词: Vega Prime; 数字港口; 可视化; 三维重建; 图形数据库

中图分类号: U 651

文献标志码: A

文章编号: 1002-4972(2014)08-0071-06

Architecture design of 3D visualization technology for digital port based on Vega Prime

WANG Yi, LI Cai-xia, LI Zhi, CHENG Xiao-lin

(China Satellite Maritime Tracking and Control, Jiangyin 214431, China)

Abstract: After analyzing 3D visualization system for the digital port based on Vega Prime, we propose the architecture design of system and scene graphics database optimization and management. Concerning the confliction between getting better visualization of 3D reconstruction and system efficiency, we present some key methods, including dynamic data schedule, dynamic LOD (Levels of Detail), as well as optimization for real-time soft shadows. A decent balance between the system fluency and high simulation can be achieved by these techniques, as shown by experimental results.

Key words: Vega Prime; digital port; visualization; 3D reconstruction; graphics database

计算机技术和现代测绘技术的迅猛发展, 虚拟现实、多媒体技术和 GIS (Geographic Information Systems, 地理信息系统)^[1]的紧密结合, 为空间信息系统的构筑提供了全新的表现方式。随着现代航运经济发展和港口港区功能日渐复杂多样, 使用传统信息表达和二维呈现方式, 已经越来越难以表现信息数据的多样性和复杂性, 探索和建设数字化、网络化、智能化、可视化的“数字港口”也成为趋势。

“数字港口”是结合现代化航运要求和信息技术发展需求, 对港口信息进行多分辨率、多尺度、多时空和多种类的可视化描述, 为港口经营企业、

港口服务企业及其他客户提供互联、高效、智能的战略及运作决策支持^[2]。本文研究了“数字港口”三维可视化系统的构架设计和场景数据库优化管理, 从空间信息的可视化呈现问题, 即如何将现实世界中的空间数据和专题信息映射到计算机中进行表达来入手, 探索了在三维重建视效和实时运行效率之间获取较好的平衡的手段。

1 系统分析与构架设计

1.1 “数字港口”可视化系统构架设计

本系统综合应用了三维重建技术和 GIS 技术, 对二维信息进行仿真, 提供了对现实场景和专题

收稿日期: 2013-12-13

作者简介: 王艺 (1987—), 男, 工程师, 研究方向为计算机应用。

信息自由视角、多层次、高精度三维可视化，同时实现了整个场景的用户交互和可操纵性。系统构架设计如图 1 所示。

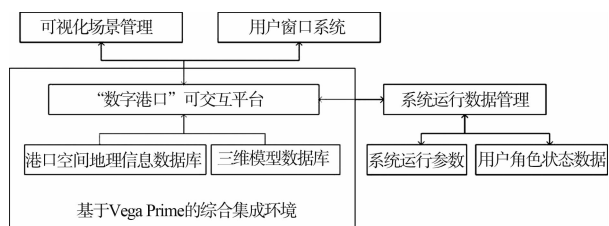


图 1 基于 Vega Prime 的“数字港口”三维可视化系统构架设计

综合集成环境包括了港口空间地理信息数据库、三维图形数据库和基于 Vega Prime 的可交互系统平台。

1) 港口空间地理信息数据库。

港口空间地理信息数据库是描述地理特征及自

然、人文特征和属性的数据仓储 (FeatureDataSet)^[3]，是可视化的基础和系统构建的数据核心。为了实现多数据源、多分辨率、多时相的数据整合以及多种类复杂元数据的管理、检索和共享，首先利用数据库和 GIS 技术，集合港口数字正射影像 (DOM, Digital Orthophoto Map)、数字地形模型 (DEM, Digital Elevation Model)、全要素数字划线图 (DLG, Digital Line Graphic) 为空间定位的基本数据，建立基础地理数据平台；然后基于 GeoDatabase 数据逻辑设计，构筑空间数据仓储，建立基于专题信息的要素类 (FeatureClass) 和空间数据分层分类组织结构，形成地形地貌、港口建筑、植被、道路交通、港区管线等基础专题数据，建立空间数据的分层和分类组织结构 (图 2)。

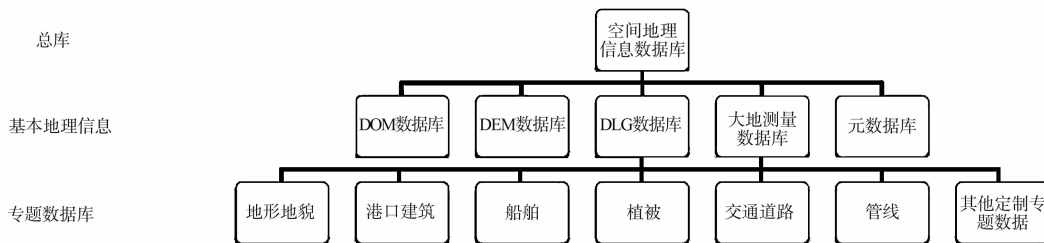


图 2 港口空间地理基础信息数据库逻辑层次结构

2) 三维模型数据库。

在空间地理信息数据库的基础上，基于 OpenGL 和 OpenFlight 对 DOM 数据库、DEM 数据库和专题数据库进行三维场景重构^[4]。首先通过 Creator DED Builder 将 DOM 和 DEM 转换为地形模型后与高精度卫图、专题数据库进行匹配，制作具备全要素专题信息的地形承载平台。在确定仿真目标、场景需求、硬件能力的情况下，需要对地形数据库进行反复试验测试，由于角色漫游是全方位、全视角的近景呈现，所以必须在保证系统实时效率的同时，满足仿真的高精度要求。地形测试成功后，需要对区域范围内映射纹理、地理属性、人文属性等其他特征数据进行优化筛选，明确各子系统模块的需求，保证数据量高效、精简。

基于地形数据库的位置信息和全要素信息，利用 MultiGen-Paradigm Creator 对各专题数据三维

重建，把地形地貌、港口建筑库、植被库、道路交通库等进行可视化呈现和空间还原。模型数据库基于 OpenFlight 图形格式进行树状层次化组织，不仅便于依据图形数据的特征值进行组织和管理，更利于实时系统的遍历操作。

3) 基于 Vega Prime 的可交互系统平台。

系统基于 Vega Prime/VSG 搭建，部分功能模块和视觉效果通过 OpenGL 混编实现，整个可交互系统平台不仅实现了三维场景漫游、模拟交互操纵，还能完成对各类专题的信息查阅。

可视化场景管理是“数字港口”可视化渲染核心，一方面进行对图形数据库的遍历读取、裁剪 (Cull)、绘制 (Draw) 等渲染流程；另一方面负责可操控角色人物 Bump 检测线和 LOS 检测线的绘制，通过对检测线是否碰撞的识别实现人物对地物的碰撞检测、全方位漫游和拾取

操作。

系统运行数据管理系统对实时系统运行参数和角色状态数据等进行记录和读取,基于数据管理系统的数据库接口,方便实现场景参数、漫游路径和操纵数据的统计和重放等功能。

1.2 场景图形数据库优化管理

三维模型数据库组织结构中不同层级的几何体节点不仅表达物体的几何信息,还承载着相应的拓扑信息,这些节点的层级结构组织方式在很大程度上决定了模型数据库的实时应用性能,特别是直接影响了实时系统的剔除和绘制过程的执

行效率。所以合理规划、统筹安排图形数据库的结构和层次,对提高系统运行效率有着极大的意义。

本系统采用了基于数据专题的空间结构方式组织图形数据库。图3和4为某港口三维可视化效果及场景图形数据结构。



图3 港口三维可视化效果图

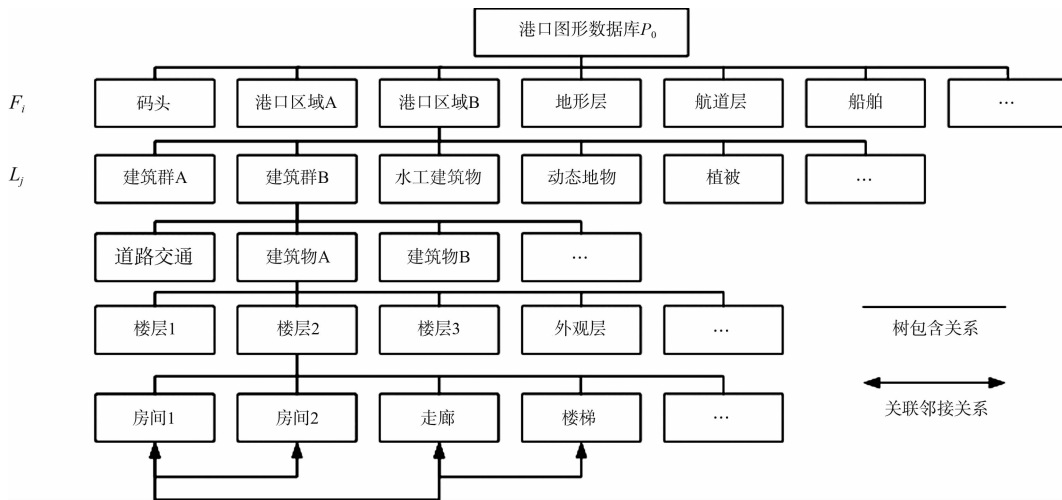


图4 港口场景图形数据结构

数据结构中, F_i 表示空间区域数据库, 如 F_1 码头、 F_2 港口区域 A、 F_3 港口区域 B..., 总库 P_0 的构成如式(1)所示:

$$P_0 = \cup_i^\gamma F_i + \cup_k a_k \quad (1)$$

式中: γ 为特定排序读取方法; a_k 为特殊元数据补充。在逻辑层面, 每个数据库 F_i 又由基于专题信息的子数据库 L_j 组成, 如港口区域 A 的构成如式(2)所示:

$$F_2 = \cup_j^\delta L_j + \cup_l b_l \quad (2)$$

其中: δ 为特定排序读取方法, b_l 为特殊元数据补充。空间关系一般由树包含关系和图形关联关系构成, 在组织构建中, 很多节点的实体采用了实例引用技术, 有效降低了数据库规模; 而关联邻接关系和逻辑专题库构建不仅较好表现了空间对

象的关系, 而且严格描述了图节点的连接属性。

通过对数据库的优化管理, 一方面, 将场景数据依据空间区域规划管理, 提升实时系统的遍历裁剪和绘制效率; 另一方面, 在同一区域内, 按照诸如水工建筑物、植被、管线等不同专题组织图形数据, 便于对数据的统一管理。

2 关键技术研究

港口虚拟场景的高精度实时描述, 需要处理的数据量极其庞大, 整个空间信息和图形数据库的精细程度及复杂程度与交互实时性之间的平衡就成为仿真系统需要权衡的关键之一。为了实现交互运行的流畅性和高仿真性, 系统采用了一些新技术。

2.1 基于空间信息的图形数据动态调度

由于实时系统对图形数据承载能力是有限的，本系统采用了基于空间信息的图形数据动态调度技术。港口和建筑内部模型数据量庞大，为了便于快速判断读取区域和控制精度，对场景地物和影像进行四叉树剖分。图5为场景剖分过程。

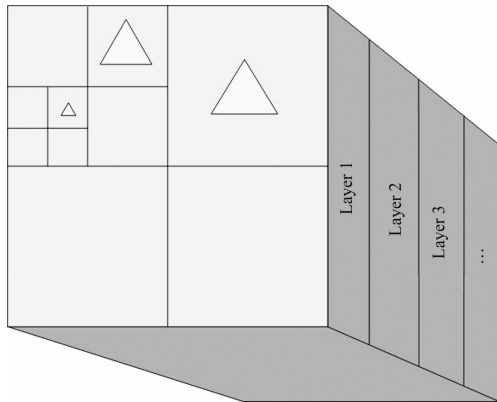


图5 场景四叉树剖分

其中，Layer1、Layer2、Layer3...等体现了在可视化系统数据处理中的分层组织思想：一方面，直观地说明了场景各种精度细节呈现方式，如LOD技术^[5]；另一方面，也是对纹理等影像信息的不同分辨率调度方法的体现，如Mipmap技术^[6]。

场景四叉树剖分后，以实时位置参数为控制参量，建立包围体，处于包围体范围外的数据直接释放，不再进入裁剪和绘制过程。动态调度过程见图6。

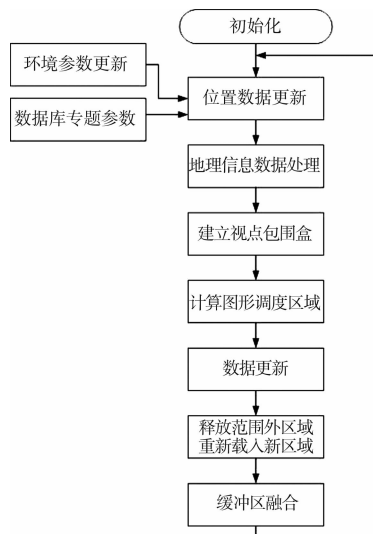


图6 场景动态调度

为了测试数据动态调度算法的性能，对系统算法和传统同步载入算法进行了试验比较。以系统全场景作为试验图形数据库，总面片数670 000。运行环境：Windows 7 64位，Intel Core i5，4G内存，ATI Radeon HD 4770显卡。表1为实时系统使用两种不同图形数据库载入算法后的运行数据，数据比较结果显示：采用了动态载入方法后，不仅直接提高了实时裁剪、绘制速度和帧数率，还间接提高了系统启动速度。

表1 不同技术的系统效率对比

方式	启动时间/s	帧数率/Hz	裁剪 Cull/ms	绘制 Draw/ms
同步载入	78	28	0.20	4.80
动态载入	60	35	0.13	3.80

2.2 基于细度剖分和分级思想的动态LOD技术

LOD技术是加速图形生成渲染的主要方法之一，当物体覆盖视锥中较小区域时采用低分辨率模型，反之只用较高分辨率模型，便于大范围复杂场景的快速渲染（图7）。

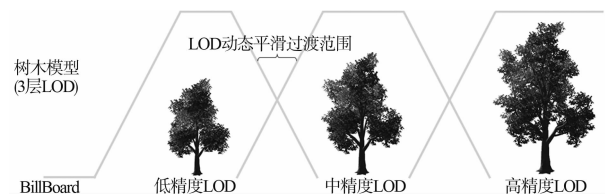


图7 LOD技术

虽然基于MultiGen-Paradigm Creator的LOD层次显隐可以通过设置过渡阈值(Transition)来减少模型突变^[7]，但是本质上依然是离散LOD，无法达到更好的连续变化效果，还会带来更多的绘制开销，本系统提出了精细剖分LOD动态过渡方法。

新算法不针对单个模型设计有限层次的精度模型，而是立足整个场景，精细剖分并建立多层次细节，最终得到一颗八叉树：构建场景包围盒，建立根节点，初始化为八叉树，读取场景中物体数据和内部细分数据，根据其包围盒将其插入到树中相应位置，得到整个八叉树结构。如图8所示。

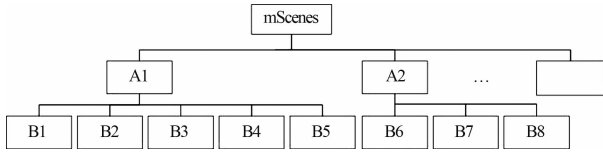


图 8 构建模型 LOD 八叉树结构

树中每个叶子节点都通过简化过程形成不同精度的几何信息, 如节点 B_i 拥有 4 个精度层次, $LOD_{B_i}^0$ 为原始模型层, μ^j 为受控简化过程, LOD 序列为:

$$LOD_{B_i}^0 \xrightarrow{\mu^0} LOD_{B_i}^1 \xrightarrow{\mu^1} LOD_{B_i}^2 \xrightarrow{\mu^2} LOD_{B_i}^3 \quad (3)$$

对树中每个叶子节点创建多层 LOD 序列, 从八叉树的叶子节点开始从下往上建立父节点 LOD 层次, 则父节点 A_1 的 LOD 序列为:

$$LOD_{A_1}^j = \cup_{1 \leq i \leq 5} LOD_{B_i}^j \quad (4)$$

其中 μ^{A_1} 是简化方法, 渲染阶段, 基于视点信息和误差对 LODs 进行裁剪和剔除并选择合适的 LOD, 在实时更新过程中根据视点变化对图形数据进行数据调度, 剔除不可见节点。图 9 为动态 LOD 系统流程。

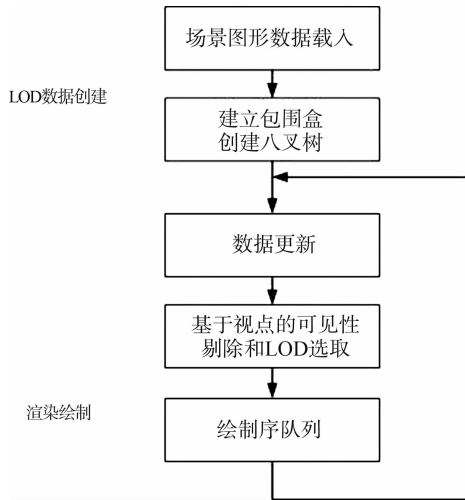


图 9 动态 LOD 系统流程

动态 LOD 算法根据绘制需要对场景灵活地精细剖分和渲染, 以较少的计算量达到了较好的细节体现, 在不影响实时效果的基础上大大提高了场景渲染的灵动感。

2.3 基于 Vega Prime 的实时软阴影优化

在高精度场景仿真系统中, 阴影绘制对于提

高用户感知度和沉浸感^[8]起着不可替代的作用。如果仅通过烘焙 (Render-to-textures) 技术^[9]制作静态阴影, 会影响用户的交互沉浸感; 而传统阴影映射技术^[10], 往往会产生阴影边缘锯齿化、自阴影 (self-shadowing) 走样等问题^[11]。

为了达到更好的效果, 系统使用阴影映射算法绘制阴影, 并做了反走样优化:

以光源为中心“视点”, 通过变换得到光源裁剪空间, 把三维空间图像深度值储存到阴影二维投影中, 得到阴影映射纹理。变换过程见图 10。

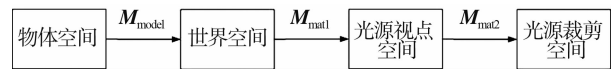


图 10 阴影映射变换

其中 M_{model} 是模型矩阵, M_{mat1} 是光源中心视图矩阵, M_{mat2} 是光源中心投影矩阵。定义 M_r 将变换取值限定于纹理坐标需要的取值范围 $[0, 1]$, 通过变换得到阴影映射纹理变换矩阵 M_{tra} 。变换过程为:

$$M_{tra} = M_r \times M_{mat2} \times M_{mat1} \times M_{model} \quad (5)$$

在实时阴影生成过程中, 从真实视点绘制场景, 对每个采样点, 计算其在光源裁剪空间中的深度值 D_{pixel} 和在阴影映射纹理中储存的深度值 D_{tex} 的大小关系, 如果 $D_{pixel} > D_{tex}$, 则采样与光源之间有遮蔽, 处于阴影中, 否则处于阴影外; 此时得到的场景阴影是离散的, 所以在比对深度值的过程中, 取采样点周边 3×3 区域中的邻接的 8 个关联采样点的深度值的平均值进行比较, 对进行深度比对后的值进行双线性差值滤波; 最后, 为防止阴影映射纹理分辨率较低产生阴影马赛克, 对阴影边缘做高斯模糊 (Gaussian Filter) 得到较好效果的软影区。

下面通过试验比较测试阴影优化算法和传统阴影算法的系统实时效率。分别以某港区企业建筑物内景和某港口码头区域作为试验图形数据库, 定义为区域 F_1 和区域 F_2 。运行环境: Windows 7 64 位, Intel Core i5, 4G 内存, ATI Radeon HD 4770 显卡。表 2 显示了采用两种不同算法的系统测试数据。

表2 不同算法的系统效率对比

区域	面片数	算法	帧数率/ Hz	时间/ms	
				裁剪 Cull	绘制 Draw
F ₁	5 326	传统算法	60	0.02	2.50
		优化算法	55	0.04	2.80
F ₂	84 432	传统算法	38	0.05	3.60
		优化算法	35	0.06	3.80

从测试结果可以看出，优化算法相比传统阴影算法帧数率和渲染处理速度有极小幅下降，对运算效率和系统流畅性并未造成太多影响，满足实时性的要求。

3 结语

1) “数字港口”三维可视化系统的研发，搭建了一个全面的、高3I（交互性 Interaction、沉浸性 Immersion、想象性 Imagination）属性的虚拟显示平台，高精度的场景呈现和便捷的用户交互为信息化条件下的港口建设规划、区域统筹管理和突发情况模拟训练提供了实时、高效的信息化方法。

2) 结合系统，提出了基于 Vega Prime 的系统总体构架设计和图形数据库优化管理，开发数据接口和各功能模块为港口运行管理提供了系统化、信息化平台，为后续道路交通、信息管控等管理需求提供了高效的分析决策手段。

3) 针对“数字港口”复杂场景高仿真呈现和系统实时效率之间的矛盾关系，提出并应用了多种策略。试验结果表明，该方法能够在满足仿真需求的基础上大大提高实时系统效率。

参考文献:

[1] Bansal V K. Use of GIS and topology in the identification and resolution of space conflicts[J]. American Society of Civil Engineers, 2011, 25(2): 159-171.
 [2] Washburn D, Sindhu U. Helping CIOs understand smart city initiatives[R]. USA: Forrester Research, 2010.

[3] Dan-Unterseh G, Courtaud J, McNeill A, et al. GIS Development for Geophysical and Geotechnical Data Integration: Application to West Africa Geohazard Assessment [C]//Houston, Texas, USA: Proceedings of Offshore Technology Conference, 2011.
 [4] MultiGen-Paradigm, Inc. OpenFlight API User's Guide (Version 3.2) [M]. Dallas, Texas, USA: MultiGen-Paradigm Inc, 2006.
 [5] Fernanda Leite, Asli Akcamete, Burcu Akinci, et al. Analysis of modeling effort and impact of different levels of detail in building information models[J]. Automation in Construction, 2011, 20(5): 601-609.
 [6] Koojoo Kwon, Eun-Seok Lee, Byeong-Seok Shin. GPU-accelerated 3D mipmap for real-time visualization of ultrasound volume data [J]. Computers in Biology and Medicine, 2013, 43(10): 1382-1389.
 [7] Ihab Hijazi, Manfred Ehle, Sisi Zlatanova, et al. Initial investigations for modeling interior utilities within 3D geo context: Transforming IFC-interior utility to CityGML/UtilityNetworkADE[J]. Advances in 3D Geo-Information Sciences, 2011(3): 1-22.
 [8] Alan B Craig, William R Sherman, Jeffrey D Will. Developing Virtual Reality Applications: Foundations of Effective Design [M]. Massachusetts USA: Morgan Kaufmann, 2009.
 [9] Diego Rossinelli, Michael Bergdorf, Georges-Henri Cottet, et al. GPU accelerated simulations of bluff body flows using vortex particle methods [J]. Journal of Computational Physics, 2010, 229(9): 3316-3333.
 [10] Cody Starr, Anthony Hinton, Neil McLaughlin, et al. Shadow Mapping in High Performance Visual Simulations [C]//Dallas, Texas, USA: Proceedings of IMAGE 2010 Conference, 2010.
 [11] Wolfgang Engel. Cascaded Shadow Maps. ShaderX5-Advanced Rendering Techniques[M]. USA: Charles River Media, 2006: 197-206.

(本文编辑 武亚庆)