

三维战场态势显示标绘技术

蒲玮, 李雄, 吴成海

装甲兵工程学院, 北京 100072

摘要 为提高三维战场态势显示标绘的图形效率和人机交互性,在总结分析战场态势信息基本理论概念的基础上,通过对显示与标绘两方面的军事需求和OSG/Qt结合使用的软件架构模式进行分析,提出了程序框架的线程模式优化方法和基于帧缓存对象的离屏纹理烘焙技术的军标符号显示生成算法。设计和实现了三维战场态势显示标绘软件模块,并对模块进行了多种实现方法的比较测试,检查验证了相关技术方案和关键技术对于提高系统绘制效率和人机交互的有效性。

关键词 三维战场态势;显示标绘;线程模式优化;帧缓存对象;OSG/Qt

中图分类号 TP391.41

文献标志码 A

doi 10.3981/j.issn.1000-7857.2014.34.009

Three-dimensional Battlefield Situation Display and Plotting Technology

PU Wei, LI Xiong, WU Chenghai

Academy of Armored Forces Engineering, Beijing 100072, China

Abstract This paper proposes an optimization method of thread mode for OSG/Qt framework and a generating algorithm for military standard symbol display using off-screen render to texture technology based on frame buffer object, to improve the display efficiency and human-computer interaction of three-dimensional battlefield situation display and plotting. The basic concept of battlefield situation information was summarized, and the military demand of display and plotting and OSG/Qt architecture were analyzed. On this basis, the software module of three-dimensional battlefield situation display and plotting was designed and tested to verify the plotting efficiency and the human-computer interactiveness of the solution and key technology.

Keywords three-dimensional battlefield situation; display and plotting; optimization of thread mode; frame buffer object; OSG/Qt

计算机作战仿真作为一种现代化的军事训练手段,在指挥训练、战术训练等军事训练领域得到了广泛运用。仿真训练中,战场态势信息是指作战双方在战斗力量对比、战斗部署和战斗行动等方面形成的状态和形式信息,是导调控制人员把握训练进程、参训人员正确决策的基础,战场态势显示标绘是作战仿真系统的核心功能之一。传统的战场态势标绘显示系统在二维军事地形图的基础上进行态势符号信息的标绘,随着计算机图形图像技术的发展,三维战场环境可视化仿真技术,以更加形象直观的优势成为仿真训练系统中战场环境呈现的基本形式。三维战场态势显示标绘是指在三维战场环境可视化仿真的基础上,运用二维的军队标号系统,对战场态势进行显示和标绘。由于战场环境可视化显示

方式从二维军事地图转变为三维战场环境,因此,出现了二维军标在三维环境中如何进行标绘和显示的技术问题。

针对这一问题,文献[1]将军标符号作为三维实体,采用动态网格平面的方法进行渲染,文献[2]在优化网格化分的基础上,提出了军标符号的3点地形匹配算法,文献[3]设计和实现了一种基于DEM网格划分的军标符号网格优化算法,提高了军标符号的地形匹配程度,文献[4]提出了一种采用公告板技术实现静态点状军标显示的问题,但以上方法存在以下不足:使用网格构建大面积复杂军标需增加三角面数目,增加系统负担;军标地形匹配采用3点或多特征点匹配算法,无法解决军标的部分面积被地形遮挡影响显示效果的问题;而采用全地形网格匹配,会加大系统负担,在进行大规模作战仿

收稿日期:2014-09-25;修回日期:2014-10-30

基金项目:国家自然科学基金项目(61473311);北京市自然科学基金项目(9142017)

作者简介:蒲玮,博士研究生,研究方向为装备信息化及装备作战仿真,电子信箱:puwei623@163.com

引用格式:蒲玮,李雄,吴成海.三维战场态势显示标绘技术[J].科技导报,2014,32(34):62-68.

真中,效率较低;采用公告板技术只能解决静态点状军标的显示,无法实现动态面状军标的显示。本文基于 OpenSceneGraph (OSG) 图形库的帧缓存对象 (frame buffer object, FBO) 技术,通过离屏渲染,动态生成标绘有军标符号的地形纹理,将生成的动态纹理映射在三维地形网络表面,实现三维战场态势的显示标绘,解决军标符号地形遮挡的问题,增强可视化显示效果,有效提高图形程序运行效率,解决以上方法在图形渲染方面存在的不足。此外,将 OSG 图形库与 Qt 开发框架相结合,优化军标标绘的人机交互界面,对三维战场态势显示标绘软件模块的开发进行程序框架方面的探索。

1 三维战场态势显示标绘的军事需求与工具分析

1.1 态势显示内容与形式的需求分析

战场态势显示作为战场态势信息的可视化表达,其显示内容由完整准确表达战场态势信息的需要决定。根据作战指挥学的基本理论,态势信息包括单位、人员、装备、设施、部署和行动6个主要因素,因此,战场态势显示的内容与之对应,也包括了上述6个主要内容。战场态势显示的形式,由导调和参训人员直观、高效、无歧义的获取战场态势信息的需要决定。一方面,军队标号规定系统完整,具有权威性和规范性,能够被导调和参训人员熟练识记,应作为态势显示的标号系统使用;另一方面,由于军队标号规定针对二维地形图进行设计,因此,需要将二维军标进行三维战场环境映射,进行地形匹配,随地形起伏进行变换,以满足三维战场环境漫游观察的需要。

1.2 态势标绘人机交互的需求分析

态势标绘交互是指使用计算机输入、输出设备,进行军标类型选择、军标特征点位置设置等人机交互操作。一方面,由于军标类型众多,需要使用复杂的人机交互界面采用树状的结构对军标库进行管理和现实,因此,态势标绘中的军标类型选择、参数设置等工作,需要使用二维窗口可视化界面的方式实现,保证操作的高效准确;另一方面,由于使用三维战场环境进行标绘,因此需要进行三维空间的位置漫游和拾取操作,保证操作的空间准确性。根据以上两个方面的需求分析,态势标绘需要使用二维软件界面嵌入三维场景渲染的软件框架进行实现。

1.3 OSG 图形开发库与 Qt 开发框架的特性分析

OSG 图形开发库与 Qt 开发框架的结合使用,可较好地满足复杂窗口操作界面和三维场景渲染相互嵌入的软件框架需要,实现跨平台、高性能等特性。

OSG 是一个开源的高性能 3D 图形开发库,主要为图形图像应用程序的开发提供场景管理和图形渲染优化功能,它使用可移植的 ANSI C++ 编写^[9],并使用已成为工业标准的 OpenGL 底层渲染 API。具有跨平台的特性,可以运行在 Windows, Max OSX 和大多数类型的 UNIX 和 Linux 操作系统

上^[6]。由于其开源的特性,便于根据需要进行二次开发,具有较大的灵活性,特别适合于实现三维战场态势标绘显示等非典型性的复杂三维图形渲染。

Qt 是一个开源的 C++ 图形用户界面应用程序开发框架,主要用于图形用户界面 (GUI) 程序的开发。Qt 具有优良的跨平台特性,支持几乎所有应用较为广泛的操作系统,Qt 的良好封装机制使得模块化程度非常高,重用性较好,底层对 OpenGL 的支持较好,特别是使用了一种称之为信号/槽 (signals/slots) 的机制替代回调 (callback),使得不同的软件开发库之间的协同工作更加安全便捷。由于其所具有的优良特性,特别适合与 OSG 等图形开发库相结合,实现三维战场态势标绘软件的 GUI 程序开发,提高工作效率。

2 三维战场态势显示标绘的关键技术

2.1 OSG/Qt 程序框架的线程模式优化

OSG 图形程序采用实时图形帧循环渲染模式,Qt 窗口程序采用消息激发重绘界面渲染模式,其典型应用的程序流程如图 1 所示。OSG 图形开发库与 Qt 开发框架的结合使用,建立集成程序开发框架的核心就是将 OSG 与 Qt 两个相对独立的程序流程和渲染模式相结合,建立和优化线程模式,确保协同工作,提高图形渲染效率。OSG/Qt 程序框架一般主要包括两种实现方法,一种为将 Qt 窗口消息处理函数和 OSG 的帧循环函数作为函数体,在主线程中建立循环函数,另一种为通过 QTimer 类建立计数器,根据固定的时间对 OSG 进行帧的循环更新^[7]。这两种实现方式本质相同,均属于单线程模式,其程序流程如图 2(a) 所示。

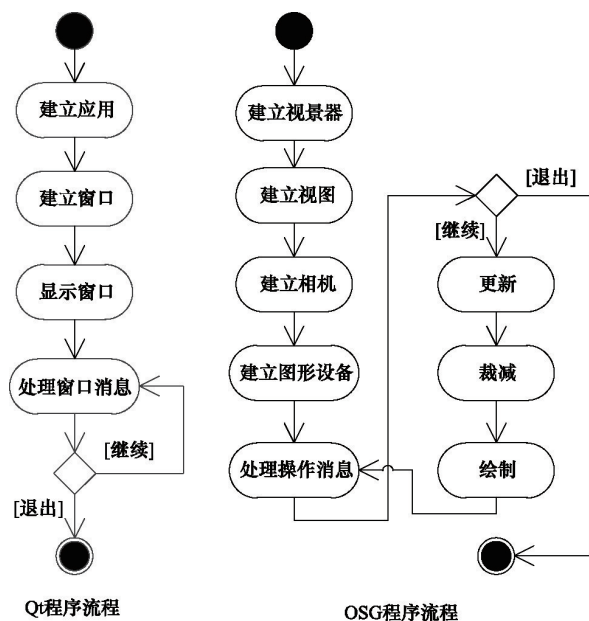


图 1 OSG 与 Qt 典型应用的程序流程
Fig. 1 Program flow of typical OSG and Qt application

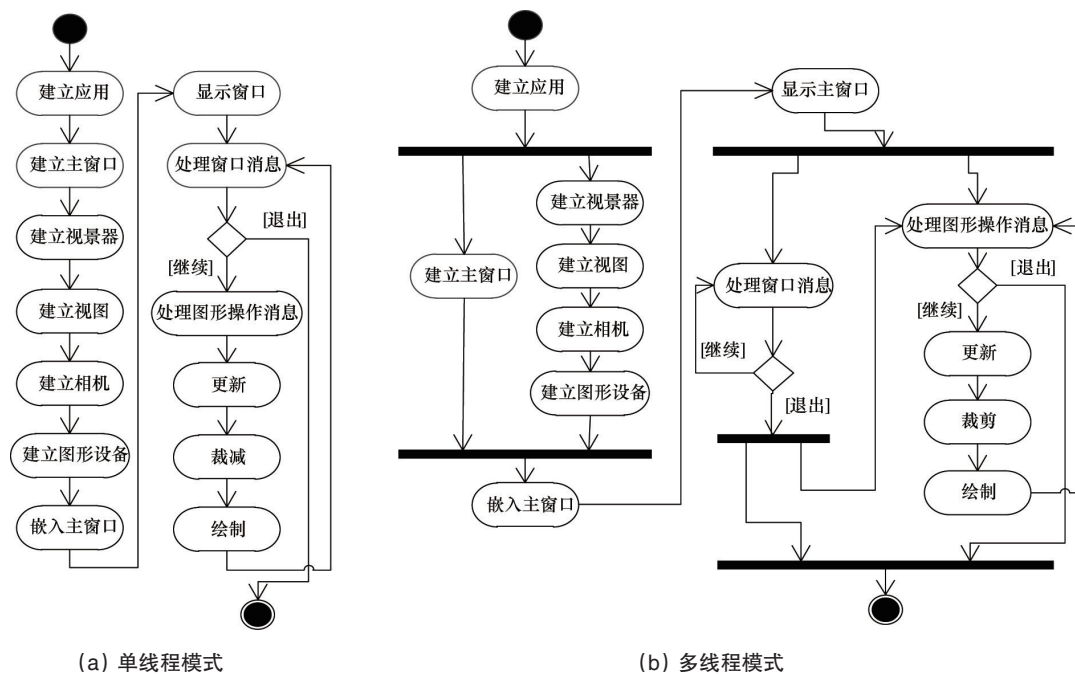


图2 OSG/Qt程序框架线程模式

Fig. 2 Thread mode of OSG/Qt application framework

单线程模式的程序框架存在两方面不足:一方面,窗口主循环与图形渲染在同一个线程内,当进行窗口GUI操作时,图形渲染线程被阻塞,引起图形渲染的停顿,影响战场态势的显示效果;另一方面,无法充分利用多CPU的硬件优势,提升图形渲染效率,制约了战场态势规模的扩大和显示效果的提高。为克服单线程模式的不足,提出和设计了一种多线程模式的程序框架,将GUI图形窗口的消息处理循环和图形渲染相分离,充分利用了硬件资源,克服了单线程模式的不足,满足了三维战场态势显示标绘对于程序框架的性能需求,程序总体流程如图2(b)所示。

2.2 基于离屏纹理烘焙技术的军标符号三维映射显示生成算法

2.2.1 基于FBO的离屏纹理烘焙技术的基本原理

离屏渲染(off screen rendering)是指将1至多个场景的渲染结果不直接输出至显示屏幕的渲染过程。纹理烘焙,又称

为渲染到纹理(render to texture, RTT),指将场景渲染的结果输出到纹理的过程,有两个主要作用:实现场景离屏渲染之后的“后置处理”(post-processing);实现多种不同场景的融合显示。本文主要使用第2种作用。一般纹理烘焙机制主要包括3种:直接复制帧缓存(frame buffer)中的像素、使用像素缓存设备^[8]和使用帧缓存对象(frame buffer object, FBO)。第1种机制需要额外的将数据复制到纹理的过程,效率较低;第2种机制由于平台相关,且需要建立自己的渲染环境(RC),切换管理十分困难;第3种机制实现了FBO与纹理的直接绑定,平台无关,切换管理十分迅速,执行效率较高^[9]。

离屏纹理烘焙是指将离屏渲染与纹理烘焙相结合,通过对屏幕渲染场景外的其他场景进行离屏渲染,将渲染结果作为动态纹理,在屏幕渲染场景中进行使用,实现多种不同场景的融合显示,使用该技术实现军标符号三维映射的原理如图3所示。

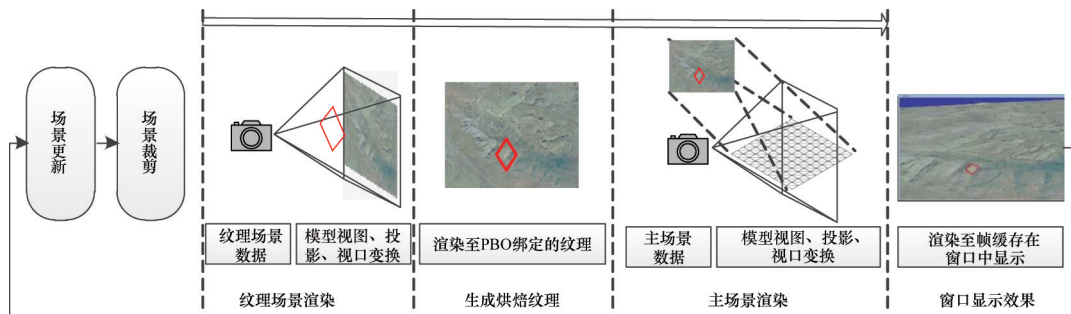


图3 基于离屏纹理烘焙技术的军标符号三维映射显示原理

Fig. 3 3D mapping display principle of military standard symbol based on off-screen render to texture technology

2.2.2 军标符号三维映射显示的算法流程

军标符号三维映射显示的算法流程基于离屏纹理烘焙技术的基本原理进行设计,主要包括了初始化和帧循环两大阶段,具体算法流程如图4所示。

初始化阶段主要完成两个场景的渲染环境、模型视图矩阵、投影矩阵、视口和三维实体的建立和参数设置;帧循环阶

段又分为3个子阶段:

1) 处理交互更新场景数据子阶段,主要完成军标位置和场景位置的更新;2) 纹理烘焙渲染子阶段,主要完成军标和地形纹理的渲染,生成烘焙纹理;3) 主场景渲染子阶段,主要完成使用生成的烘焙纹理,生成三维战场环境,在窗口内进行显示。

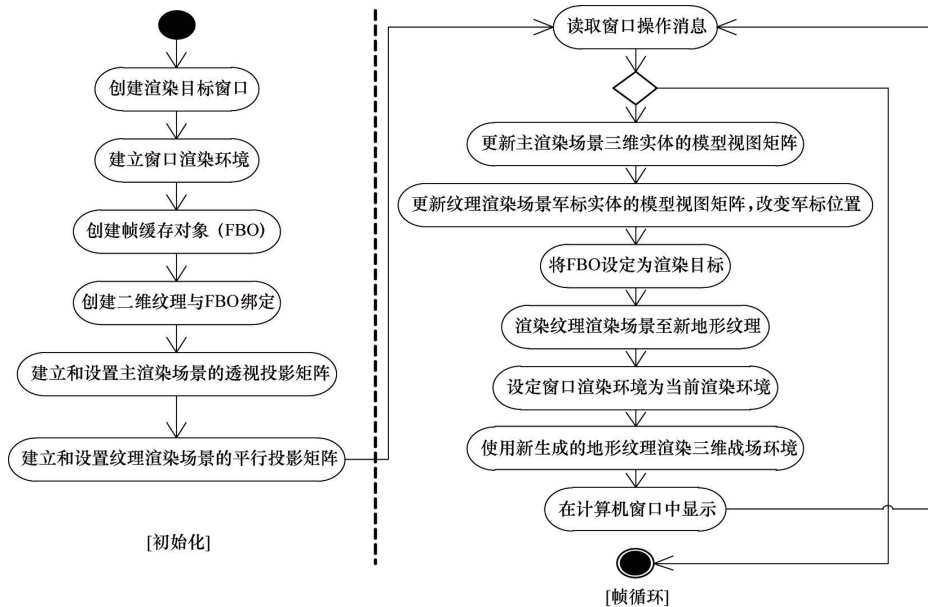


图4 军标符号三维映射显示的算法流程

Fig. 4 3D mapping display algorithm flow of military standard symbol

2.2.3 坐标映射计算

由于态势的标绘显示涉及两个渲染场景,即纹理烘焙场景和三维战场环境场景,在进行标图时,由于纹理烘焙采用离屏渲染的方式,需要在三维战场环境中进行军标特征点的选择拾取操作,确定位置点后,需要在纹理烘焙场景坐标系中更新军标位置,因此涉及三维坐标空间转换映射的计算,如图5所示,需要实现 P_{pick} 点(三维战场环境鼠标拾取)到 P_{pick_0} 点(纹理烘焙场景对应点坐标)的映射。

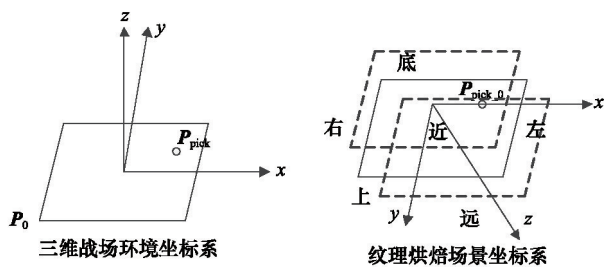


图5 坐标映射计算

Fig. 5 Calculation of coordinate mapping

1) 拾取点战场环境场景世界坐标系坐标映射为地形本地坐标系坐标。

设 $P_{pick} = (x_p, y_p, z_p)$, $P_0 = (x_0, y_0, z_0)$ 为地形实体原点在世界坐标系下的坐标,则

$$P_{pick_terrain} = P'_{pick} \cdot M_{terrain} = (x_p, y_p, z_p, 1) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix} \quad (1)$$

其中, $P_{pick_terrain}$ 为拾取点在地形实体本地坐标系中的坐标位置,地形实体本地坐标系以地形实体左下角为原点,坐标向量与世界坐标系相同。 P'_{pick} 为拾取点战场环境场景世界坐标系坐标向量。 $M_{terrain}$ 为地形实体在世界坐标系下的位置变换矩阵。

2) 拾取点地形本地坐标系坐标归一化。

设地形的 x 轴方向宽度为 w , y 轴方向长度为 h , 最高高程为 z_{max} , 则

$$P'_{pick_terrain} = P_{pick_terrain} \cdot M_{nor} = P_{pick_terrain} \cdot \begin{bmatrix} \frac{1}{w} & 0 & 0 & 0 \\ 0 & \frac{1}{h} & 0 & 0 \\ 0 & 0 & \frac{1}{z_{max}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

其中, M_{nor} 为拾取点地形本地坐标系归一化矩阵。

3) 拾取点本地坐标系坐标映射为纹理场景世界坐标系坐标。

设纹理场景平行投影参数左、右、上、下、远、近裁剪面的值为: $x_{left}, x_{right}, y_{top}, y_{bottom}, z_{near}, z_{far}$, 原始地形纹理图片实体的 z 坐标值为 $z_{texture}$, 则

$$P'_{pick_0} = P'_{pick_terrain} \cdot M_{orth} \cdot M_{tran} = \begin{bmatrix} x_{right} - x_{left} & 0 & 0 & 0 \\ 0 & y_{top} - y_{bottom} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & z_{tran} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_{right} & -y_{top} & 0 & 1 \end{bmatrix} = (x'_{pick_0}, y'_{pick_0}, z_{tran}, 1) \quad (3)$$

其中, M_{orth} 为归一化的拾取点地形坐标向纹理正视投影系统坐标转换的中间变换矩阵, 变换结果为拾取点距离正视投影坐标左界和上界的距离, M_{tran} 为拾取点距离正视投影坐标左界和上界的距离到正视投影坐标的转换矩阵。且 $z_{near} < z_{tran} < z_{texture}$ 。

因此, $P'_{pick_0} = (x'_{pick_0}, y'_{pick_0}, z_{tran})$ 为拾取点在纹理渲染场景中的空间坐标。

3 三维战场态势显示标绘软件模块的设计与实现

根据三维战场态势显示标绘的内容形式和人机交互方式的军事需求分析, 运用 OSG/Qt 程序框架和军标符号三维映射显示生成算法等关键技术的研究, 设计和实现了三维战场态势显示标绘软件模块。

3.1 软件模块的总体设计

软件模块的总体结构如图 6 所示, 主要包括用户界面程序, 军标数据库, 战场环境数据库和态势文件数据库。用户界面程序可读取各数据库的数据, 实现三维战场态势的渲染、键盘鼠标的操作等功能; 军标数据库以文件数据库的形式对军标图形的矢量数据进行存储; 战场环境数据库以文件数据库的形式进行存储, 主要包括地形高程数据、地表纹理

数据, 地形特征数据等; 态势文件数据库主要对已经标示完毕的态势进行保存。

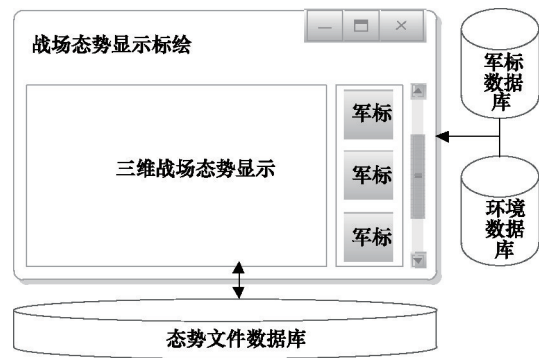


图6 软件模块结构

Fig. 6 Structure of Software Modules

3.2 软件模块的类结构设计

软件模块的功能通过 7 个主要的功能类进行封装实现, 如图 7 所示。其中, `MainWindow` 继承自 `QMainWindow`, 是程序模块的窗口框架类, 主要实现 GUI 界面的功能; `OsgWidget` 继承自 `osgViewer::Viewer`, 内嵌入 `MainWindow`, 实现 OSG 图形渲染窗口的功能, 管理渲染工作; `FrameThread` 类继承自 `OpenThreads::Thread` 类, 调用 `OsgWidget` 的 `Frame()` 函数, 实现 OSG 的渲染线程; `BattleRender` 类继承自 `osg::Geode` 类, 是软件模块渲染的核心。建立一个纹理渲染 `osg::Camera` 对象, 对纹理渲染场景进行管理, 通过读取战场环境数据库的基础数据, 实现对战场环境渲染相关的 OSG 场景图节点进行构建。该类所建立的对象, 作为子节点, 加入 `OsgWidget` 类对象的数据根节点下; `JBDB`, `BattleDB` 和 `TaishiDB` 类分别对军标数据、战场环境数据和态势数据进行管理。

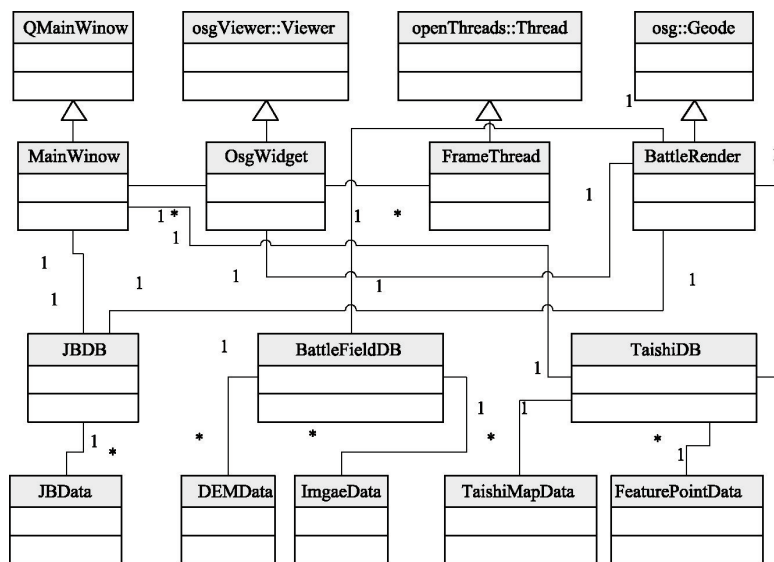


图7 软件模块类结构

Fig. 7 Class structure of software modules

3.3 软件模块的逻辑流程

软件模块的各功能类之间的协作关系、交互信息如图8所示。其中 InitRenderScene 消息的处理函数调用 BattleRender 类对象的一系列操作函数,实现 OSG 场景图的构建和渲染参数的设置,其伪代码为:

```

void InitRenderScene()
{
    BattleRender->新建渲染到纹理 Camera 节点;
    BattleRender->新建地形纹理,作为渲染到纹理 Camera 的渲染目标;
    BattleRender->设置纹理相机的渲染顺序为预先渲染,投影类型为平行投影;
}

```

```

BattleRender->读取地形 DEM 和纹理数据;
BattleRender->建立映射地形纹理的四边形平面和军标实体并作为渲染到纹理 Camera 的子节点;
BattleRender->建立战场环境的三维模型节点,将生成的地形纹理作为模型纹理;
OsgWidget->将渲染到纹理 Camera 节点和战场环境的三维模型节点作为主渲染场景 RootData 的子节点;
}

```

3.4 软件模块的实现效果

基于 Qt4.8.4 和 OpenSceneGraph-3.0.1 开发框架,使用 Microsoft Visual Studio 2010 对软件模块进行了开发实现,实现效果如图9所示。

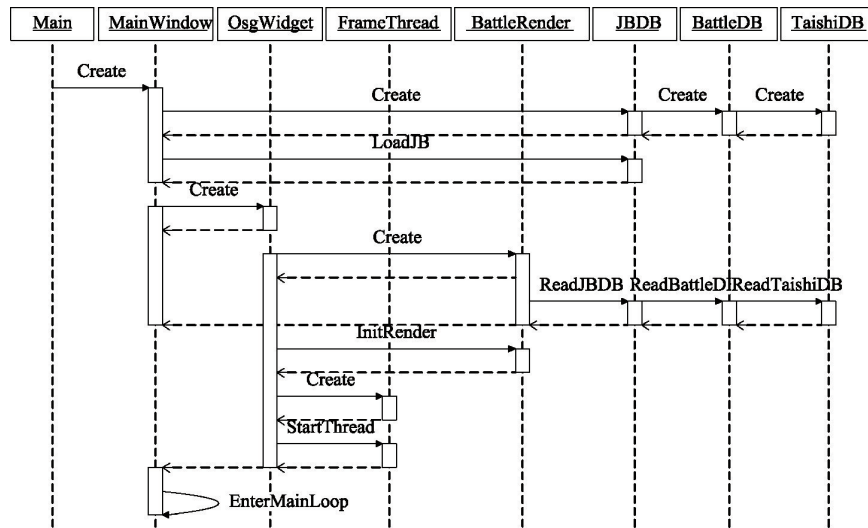


图8 软件模块的逻辑流程

Fig. 8 Logical flow of software modules

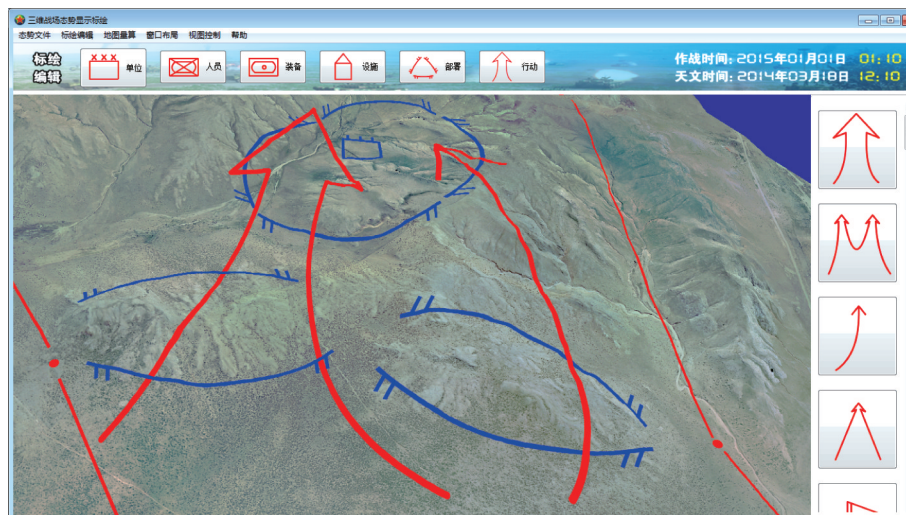


图9 软件模块实现效果

Fig. 9 Realization of software modules

4 实验结果

在实现的三维战场态势显示标绘软件模块的基础上,分别对网格化三维实体军标的三点地形匹配算法和全地形网格地形匹配算法进行了实现,与本文方法进行比较测试,实验环境为:CPU Intel i7 3770,内存 16G,显卡 NVIDIA Quadro5000,操作系统 Window 7 64 位旗舰版,使用中国北部边境某地区的 16 km×16 km 范围 1 m 分辨率的 DEM 数据和 1

像素/m 的卫星图片地形纹理的战场环境数据,标绘 50 和 200 个军标符号进行测试,实验结果如表 1 所示。

通过实验结果可见,基于 OSG/Qt 的三维战场态势显示标绘技术方案可行,通过线程优化和基于离屏纹理烘焙技术的军标符号三维映射显示生成算法提高了图形效率,满足了军事需求。该模块已经在某大型作战仿真系统的综合态势显示子系统中得到了应用。

表 1 性能测试结果

Table 1 Result of performance test

序号	军标算法		图形渲染帧率/(帧·s ⁻¹)		操作 Qt 界面三维场景是否停顿	是否适用于全类型军标	是否出现地形遮蔽问题
			50 个军标	200 个军标			
1	网格军标三点地形匹配	单线程	35	28	是	是	是
		线程优化	38	30	否		
2	网格军标全地形网格匹配	单线程	48	20	是	是	否
		线程优化	50	23	否		
3	广告牌军标	单线程	59	49	是	否	是
		线程优化	63	52	否		
4	离线纹理渲染军标	单线程	60	52	是	是	否
		线程优化	65	57	否		

5 结论

三维战场态势显示标绘在大规模作战仿真系统中有着重要的应用。本文基于 OSG/Qt 的软件架构,实现了在三维战场环境中进行军队标号的动态标绘和显示的功能。实验结果表明,该方法相对传统的网格实体军标建模与渲染方法而言,效率更高,更能满足军事应用需求,因此在大规模作战仿真系统和指挥信息系统的开发中有较好的应用前景。

参考文献(References)

[1] 张嵘,蔡洪斌,白忠建. 三维态势图中标绘箭头的实时约束变形[J]. 成都信息工程学院学报, 2008, 23(4): 380-383.
Zhang Rong, Cai Hongbin, Bai Zhongjian. Real-time constrained deformation of plotting arrow in 3D common operational picture[J]. Journal of Chengdu University of Information Technology, 2008, 23(4): 380-383.

[2] 马洪文,熊军,杨晓松,等. 基于 OpenGL 中 Bezier 曲面的三维箭标绘制研究[J]. 军事交通学院学报, 2009, 11(5): 86-90.
Ma Hongwen, Xiong Jun, Yang Xiaosong, et al. Study of the plotting of 3D bilinear arrowhead based on OpenGL Bezier curve[J]. Journal of Academy of Military Transportation, 2009, 11(5): 86-90.

[3] 汤晓安,陈敏. 一种基于网格控制的 3 维军标符号动态建模与优化方法[J]. 中国图象图形学报, 2009, 14(9): 1919-1924.
Tang Xiaohan, Chen Min. An approach of 3D military symbols dynamic modeling and optimizing based on control grid[J]. Journal of Image and Graphics, 2009, 14(9): 1919-1924.

[4] 杨强,陈敏,汤晓安,等. 三维静态军标的实时生成与标绘[J]. 计算机工程与设计, 2007, 28(14): 3419-3421.
Yang Qiang, Chen Min, Tang Xiaohan, et al. Real-time creating and protracting of 3D static unit symbols[J]. Computer Engineering and Design, 2007, 28(14): 3419-3421.

[5] Li J, Li G J, Hou W, et al. The research on virtual assembly technology and its application based on OSG[J]. Lecture Notes in Computer Science, 2013, 7719(1): 282-296.

[6] 肖鹏,刘更代,徐明亮. OpenSceneGraph 三维渲染引擎编成指南[M]. 北京:清华大学出版社, 2010: 2.
Xiao Peng, Liu Gengdai, Xu Mingliang. A programming guide to 3D Render Engine OpenSceneGraph[M]. Beijing: Tsinghua University Press, 2010: 2.

[7] 唐云,邓飞,黄地龙. 基于 Qt 和 OpenGL 的三维地质建模[J]. 计算机科学, 2011, 38(2): 281-283.
Tang Yun, Deng Fei, Huang Dilong. Three dimensional geological modeling based on Qt and OpenGL[J]. Computer Science, 2011, 38(2): 281-283.

[8] Lipowski J K. D-buffer: Irregular image data storage made practical[J]. Opto-Electronics Review, 21(1): 103-125.

[9] 王锐,钱学雷. OpenSceneGraph 三维渲染引擎设计与实践[M]. 北京:清华大学出版社, 2009: 181-182.
Wang Rui, Qian Xuelei. Design and practice of 3D render engine OpenSceneGraph[M]. Beijing: Tsinghua University Press, 2009: 181-182.

(责任编辑 刘志远)