OSG 中三维场景构建的关键技术

项　杰[1]，靳媛媛[2]，李爱光[1]
(1. 解放军信息工程大学 测绘学院，河南 郑州 450052; 2. 长安大学，陕西 西安 710054)

摘　要：随着基于 OpenGL 的各种三维渲染引擎的日趋成熟，如何更好地利用这些引擎已成为人们首先要解决的问题，而 OpenSceneGraph 作为一款高性能的、开源的渲染引擎也越来越受到人们的关注。系统地对 OSG 的结构和特性进行了分析，并对 OSG 中三维场景的构建过程以及其中的一些关键技术如三维场景的数据组织、场景的裁剪技术、渲染的流程等进行了研究。

关键词：OSG ；三维场景；渲染引擎；数据组织；裁剪；构建技术；地形生成
中图分类号：P208　　**文献标志码：**B　　**文章编号：**1672-4623（2012）01-0043-03

随着信息技术的发展，地理信息表达出现多元化的趋势。虚拟现实技术为地理信息的可视化表达提供了一种直观有效地表达手段。近年来，人们对地理信息的三维可视化表达进行了广泛的研究。而三维场景的构建技术是地理信息三维可视化表达的基础，本文就对在 OSG 开源库的基础上构建三维场景的技术进行了系统地研究。OpenSceneGraph（OSG）是一个开源的跨平台图形管理开发库，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，被广泛应用于虚拟现实、虚拟仿真、科学和工程可视化等领域。随着 OSG 模块和第三方附加库的不断完善，OSG 已具备对高性能渲染、海量地形数据库、地理信息及多通道的支持。

1　OSG 核心库简介

1.1　OSG 核心库

OSG 核心库的主要功能就是对实现最核心的场景数据库的组织和管理、对场景图形的操作以及为外部数据库的导入提供接口等。

osg 库：基本数据类，负责提供基本场景图类，构建场景图形结点，如节点类、状态类、绘制类、向量和数学矩阵运算以及一般的数据类。同时，它包含一些程序所需要的特定功能类，如命令行解析和错误调试信息等。

osgUtil 库：工具类库，提供通用的公用类，用于操作场景图形及内容，如更新、裁剪、遍历、数据统计及场景优化。

osgDB 库：数据的读写库，负责提供场景中数据的读写工作，提供了一个文件工具类。OSG 中场景图管理是通过遍历场景图层次结构来完成大部分的数据处理工作的。

osgViewer 库：视窗管理库，可以集成各种窗体管理系统，提供 OSG 与各种 GUI 的结合。因此，它是跨平台的 3D 管理窗口库。

1.2　OSG 特性

1）质量高。由于 OSG 的开源性，它的代码经历了许多优秀的开发成员的测试和修改，在程序架构和执行效率方面都有着较高的品质。

2）开源性。OSG 不同于目前商业化的一些三维引擎，它的所有源代码都是遵循 OSGPL 开源协议发布的。使用和发布基于 OSG 的程序和软件不需要额外的费用。

3）跨平台。OSG 支持的操作系统平台有 Windows、UNIX、Linux、Mac OS X、IRIX、Solaris、HP-UX、AIX 和 FreeBSD。

4）可扩展性。由于最初程序的架构设计，OSG 提高了强大的扩展能力，支持对节点、渲染属性、回调以及交互事件的扩展，并且支持第三方插件扩展，目前引入的插件已有几十种。

2　关键技术

2.1　三维场景数据的组织

2.1.1　节点

OSG 中三维场景的基本单位是节点，包括组节点和叶节点。叶节点管理一个或多个可绘制体的信息，并可以通过它的接口函数对可绘制体的信息进行查询。

2.1.2　场景树的构建

OSG 渲染引擎采用一种自顶向下的、分层的树状结构（场景树）来实现空间数据的组织，这样大大提高了场景的渲染效率。例如，地理要素就可以采用图

收稿日期：2011-04-26

1 方式来组织场景数据。

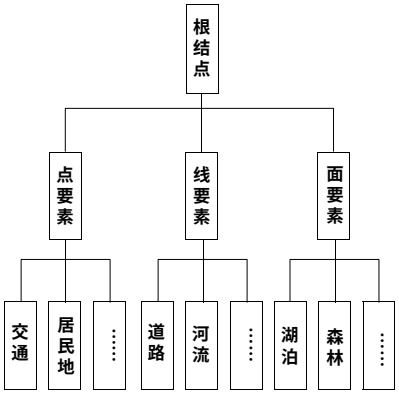

图 1 地理要素场景树的构建

2.1.3 访问机制

节点的访问就是获取相应节点的属性，以及对它进行相应的操作。而在实际应用的过程中，往往需要对一个场景子树的所有节点进行访问，由于不同节点数据结构及操作的异构性，很难编写通用的代码。因此，OSG 中通过 osg∶∶NodeVisitor 类（节点访问器）来解决这个问题。节点访问器是一种典型的对象行为模式。它支持对不同节点元素采用不同的用户自定义操作。访问者包括访问者角色及其实现过程、以及节点角色及其实现过程。采用“双重分派”的技术（如图 2 所示）来实现访问器与节点之间的交互。

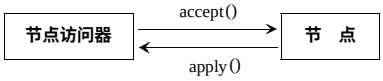

图 2 “双重分派”技术

2.2 场景的裁剪

2.2.1 裁剪技术

三维场景的裁剪是三维场景可视化的核心技术，对提高显示效率起到决定性作用。特别是在海量数据的情况下更显得尤为重要。裁剪的目的是提取，这样有助于显示的节点的数据，过滤掉冗余数据，从而达到提升显示效率的效果。OSG 中支持以下几种裁剪技术：

1）背面裁剪。当物体为不透明时，观察者只能观察到大约一半的物体，因此，过滤掉这些看不见的数据就可以减少 50%的绘制工作。

2）视锥体裁剪。将人的视线范围看成一个近似的锥体，在这个锥体之外的物体将被裁剪掉（见图 3），包括近平面裁剪、远平面裁剪以及视锥体裁剪 3 个部分。超出这 3 个部分之外的物体将剔出渲染队列之外。

3）细节裁剪。当场景中的物体的大小小到不影响观察者的视觉感受的时候，将这些数据裁剪掉，通常是由一个像素阈值来判定。因此，它是一种牺牲质量来换取速度的方法。

4）遮挡裁剪。判断场景物体的相互遮挡关系，裁剪掉被完全遮挡的物体。由于遮挡的算法具有较高的复杂度，往往计算的开销会大于绘制的开销，因此，这种方式并不实用。OSG 中对这种方法进行了改进，定义了一种“遮挡板”对象（Occluder），这样在场景中设置少量的遮挡板就能够大大加快裁剪的效率。

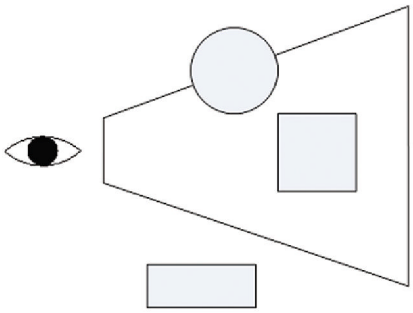

图 3 与视锥体没有交集的物体被裁剪

5）聚集裁剪。聚集裁剪是对背面裁剪的扩展，它将多个物体作为一个整体来进行裁剪，以提高背面裁剪的效率。其基本原理为：用截断圆锥体包含一组几何体的所有法向方向和所有点。如图 4 所示，如果对于视点 eye，满足如下条件:

$$n \cdot \frac{eye-cp}{|eye-cp|} < sina$$

则认为视点位于该截断面的背面，应当统一剔除它所包含的多个几何体。

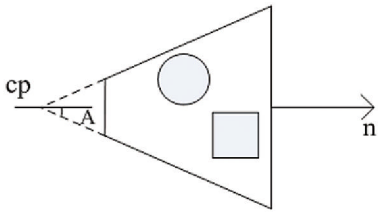

图 4 聚集裁剪基本原理

OSG 中通过 osg∶∶CullingSet 类（裁剪器）来提供对以上几种裁剪方式的支持，并且包含了视锥体、遮挡板等信息。

2.2.2 裁剪访问器

OSG 中通过裁器访问器（osgUtil:: CullVistor 类）来实现整个场景的裁剪功能，它的作用是遍历数据节点树，过滤不用显示的节点和分枝，并将节点树转换成按照 OpenGL 渲染状态排序的树状结构，即状态树。它是 OSG 的核心实现者之一。

2.3 渲染的基本流程

2.3.1 OSG 状态机

OpenGL 是一种状态机，OSG 继承了这一特性，通过 osg:: State 这个类来实现对 OpenGL 状态量、属性参数的封装，以及顶点数组的设置。它的主要功能有：①提供了对 OpenGL 状态堆栈的处理机制；②对将要进入渲染管线的数据进行排序和优化；③提供查询当前

OpenGL 状态值的接口函数。因此，可以说这个类是 OSG 场景绘制的核心类。

2.3.2　场景的渲染流程

图 5 为场景的渲染流程图，OSG 通过裁剪访问器对场景树进行遍历，同时将场景的渲染状态以及应用这些状态的数据依次传递到 OpenGL 渲染线程，并且生成状态树，然后状态机通过遍历状态树对叶节点进行绘制。

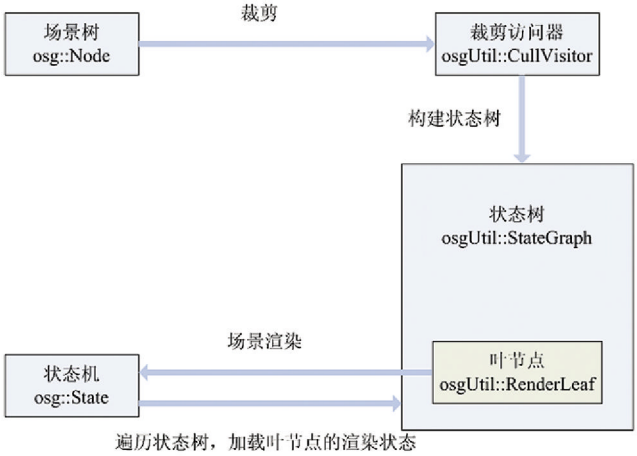

图 5　场景的基本渲染流程

3　OSG 中地形数据生成演示

首先，我们用基于 OSG 的地形生成工具 VPB 将 367M 的纹理数据（见图 7）和 187M 的高程数据（见图 6）生成 OSG 的地形文件 ive。

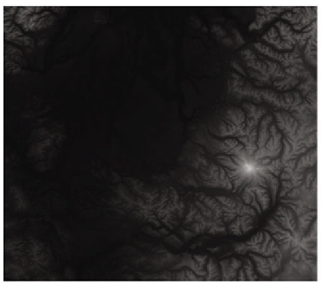

图 6　高程数据图

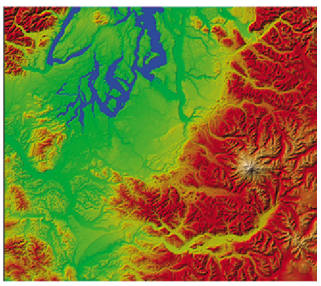

图 7　纹理数据

在 Win32 平台下的 C++编译器中，显示地形的基本步骤如下：

1）新建一个 Viewer 对象。

```
osg::ref_ptr<osgViewer> = new osgViewer::Viewer();
```

2）加载已生成的地形文件。

```
osg:: ref_ptr<osg:: Node> model = osgDB:: readNodeFiles (_T ("D: // demo.ive"));
```

3）对加载的地形数据进行优化，删除冗余的节点和状态等。

```
osgUtil:: Optimizer optimizer;
optimizer.optimize (loadedModel.get ());
```

4）设置渲染场景的数据。

```
viewer.setSceneData ( model.get () );
```

5）最终将数据在图形显示视备上可视化。

```
viewer.realize ();
viewer.run ();
```

笔者也对显示帧速率进行了测试，结果如图 8 所示，实验证明了该地形显示的平均速率在 60 帧以上。这说明 OSG 在对 OpenGL 的封装过程中不仅没有降低其显示效率，而且极大地提高了开发效率，简单的几句话就能够完成地形的导入与显示。

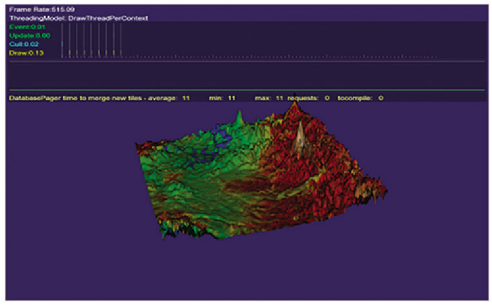

图 8　测试结果

注：实验的硬件环境是：Dual-Core 2.5GHz 双核 CPU，2G 内存，集成显卡。

3　结　语

OSG 自诞生以来，就得到了许多开源爱好者的青睐，使得 OSG 不断地完善和扩充。现在 OSG 以其高性能、可扩展、可移植、开源的特点吸引了越来越多的软件开发人员，并在各个相关领域也得到了广泛的应用，这其中也包括地理信息系统（GIS）。基于 OSG 产生了一系列的开源项目，包括海量地形生成工具 VPB、实时地形加载和渲染工具 osgEarth、兼容 GIS 数据的 osgGIS 等。随着研究的不断深入，OSG 将会在地理信息的三维可视化中发挥越来越重要的作用。

参考文献

[1] 游雄，万刚.战场可视化与数字地图[J].地理信息世界，2004，2(3)：19-24
[2] 高俊，游雄.虚拟现实及其在军事测绘与作训模拟中的应用[J].解放军测绘学院学报，1996，13(2)：133-137
[3] 淮永建，于鹏，黄友良，等.基于OSG数字林分景观可视化技术研究[J].北京林业大学学报，2009(6)：67-71
[4] OpenSceneGraph 官方网站. www. openscenegr- aph.org
[5] OpenSceneGraph 中国官方网站. www.osgch- ina.org
[6] 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
[7] 阮鲲，黄元生，顾大权.基于 OpenGL 的三维地形可视化[J].气象水文装备，2006，17(1)：11-13

第一作者简介：项杰，硕士，主要研究方向为地图制图和 GIS。