

基于 OSGEarth 的地图数据服务器构建方法

罗丽虹¹ 孙卡¹ 赵杰¹ 周燕¹
LUO Lihong SUN Ka ZHAO Jie ZHOU Yan

摘要

针对不同项目地图需求不同、数据占用内存大等问题,文章提出一种构建地图数据服务器的解决方法。主要对百度、谷歌等不同平台的瓦片数据等级和存储格式原理进行研究,然后根据需求实现一种可拓展的转换算法,并将下载的原始地图瓦片数据和数字高程数据(DEM)转换成相应数据。同时基于 Apache 服务器完成了地图数据服务器搭建,将地图数据通过磁盘阵列部署到服务器中。通过 OSGEarth 将地图数据可视化应用,搭建的地图数据服务器很好地满足了需求,能够作为一种可行方法进行应用。

关键词

OSGEarth; 地图数据; 数据下载; 瓦片转换; 服务器搭建

doi: 10.3969/j.issn.1672-9528.2024.12.006

0 引言

随着地图技术的快速发展,地图瓦片和数字高程数据精度越来越高,内存占用空间越来越大,每台主机存储大量的地图数据不符合实际需求。而对于局域网内多客户端共享地图数据的需求,构建一个地图数据服务器可更好的满足实际应用。并且由于不同项目对于地图精度、大小及保密性的要求不同,项目应用的实际需求超出了当前离线地图服务的能力范围。本文提出的可拓展转换算法,不仅能实现地图瓦片数据等级及存储格式的转换,同时还具有较高的转换效率。因此,提出的构建地图数据服务器的方法,对于各种专题地图需求迅速增加的问题具有参考意义。

对于构建地图服务器,国内许多学者也进行了相关的研究。陈红华等人^[1]基于 XML 信息交换技术构建的地图服务器可支持灵活定制地图服务,用于解决一些行业应用中对地图服务器有特殊要求的问题;李原野等人^[2]提出了一种基于 Web 的、以开源软件为开发基础的地图服务器的设计方法,可以实现融合多格式、多来源的空间数据,为普通的 Internet 用户提供地图服务;马娟等人^[3]提出一种新的基于集群技术的高性能混合动态地图服务方法,并在服务器集群技术的基础上,设计了一种支持高并发可伸缩的动态地图服务模式,采用两级中调度服务和相应的负载均衡技术解决了高并发请求时地图服务效率和正确性问题;陆丽华^[4]设计实现了基于 Web 的矢量化地图服务器。综上所述,国内对地图服务器的研究应用十分深入,但基础的地图数据服务器构建研究存在不足,因此搭建良好的地图数据服务器满足项目需求具有一定的应用价值。考虑到 Apache 服务器应用广泛,稳定性

等优越性,本文采用 Apache 服务器。

1 总体设计

基于 OSGEarth 构建地图数据服务器主要用于局域网内对地图的需求。首先,下载原始的地图瓦片数据和数字高程数据(DEM 瓦片),数据的文件目录等级不能直接用于加载生成地图,存储格式根据需要可进行选择转换,对此本文设计实现转换方法使其达到使用要求;然后,通过磁盘矩阵将数据和 XML 文件(地图数据解析文件)部署到服务器中,供 OSGEarth 加载生成三维地形地图。文中对各部分进行详述。总体设计步骤流程如图 1 所示。



图 1 设计流程图

2 地图数据下载

2.1 瓦片数据下载

目前,市面上有很多地图瓦片数据下载软件,可提供相应服务,其中包括提供下载影像数据(地图瓦片)和高程数据(DEM 瓦片)。例如,BIGEMAP 地图下载器、水经注地图下载器、LocalSpace Viewer、Mobile Atlas Creator 等,但其中 BIGEMAP 和水经注为收费软件,用户可根据需要选择合适的下载器。本文利用开源软件下载中国全部级别的谷歌影像数据,谷歌影像当前瓦片等级为 1~22,国内一般到 20 级,国外可达 22 级,百度影像为 1~19 级。由于全国地图瓦片数据量十分庞大,以国为单位下载瓦片耗时长、存储空间不足,本文选择以省为单位建立下载边界,再将省内细分为市下载等级更低的瓦片。软件中选择对应省市建立下载任务,自定

1. 南昌航空大学信息工程学院 江西南昌 330063

义选择存储目录和建立任务名称,选择下载等级,如图2所示。



图2 自定义下载界面

下载全国地图所有等级的瓦片数据需要硬盘内存达到百TB以上,将瓦片数据分省市、等级进行下载存储,为下文第三部分转换算法提供原始瓦片。

2.2 高程数据下载

数字高程模型(DEM)是基于对地形采样使用数字化方法表示地貌形态的空间分布,它描述了地形表面空间位置及相关属性。DEM的结构模型主要有规则格网模型、不规则三角形模型(TIN)和等高线模型。当前国内外主流的两类免费DEM数据是:ASTER GDEM(先进星载热发射和反射辐射仪全球数字高程模型)和SRTM DEM(对地球表面遥感测量的数据进行处理制成的数字地形高程模型)^[5],谷歌基于SRTM重新组织数据结构实现瓦片式的DEM。谷歌地球从第3级起开始提供DEM高程数据,13级以下基本实现全球覆盖(高纬度地区除外),某些重点区域如大城市最高19级可提供高程数据。此时DEM高程数据的空间分辨率可达10m,远高于中国区域内公开的SRTM3 90m的空间分辨率。因此谷歌地图的DEM高程数据几乎是中国境内公开渠道能获取的空间分辨率最高、精度最好的DEM高程数据。本文下载的高程数据同上文地图瓦片下载方法相同,同时下文中文瓦片等级转换算法与高程数据转换思路一致。

3 地图瓦片转换算法

地图瓦片转换算法主要包含瓦片等级转换、瓦片存储格式转换、DEM数据等级转换,本文以谷歌瓦片转换为例实现算法的验证,DEM数据的转换原理同瓦片数据方法思路相同。

3.1 瓦片地图原理

地球是一个椭球,而地图是显示在平面上的,故需通过投影将球面坐标转为平面坐标,最常用的投影是墨卡托投影。

在使用地图时,需要微观的地图信息,一幅精确到街道级别的地图图片像素量可达上百万至上千万,巨大的数据量无法一次性加载并且会导致内存崩溃。为此,将地图进行等级切分,最高级时(一般最高级为0)需要的信息最少,一般为1张256px×256px的图片,在下一级时信息量增多,用1张512px×512px的图片表示。将上一级图片分割成4个相等的图片,即瓦片,且每张瓦片均为256px×256px。以此类推,级别越低的图片像素越高,瓦片数据量越多,下一级的像素是当前级的4倍,瓦片数量以指数倍数增加。这样的等级切片从高到低构成了一个金字塔瓦片模型,如图3所示。



图3 金字塔瓦片模型示意图

3.2 算法分析

3.2.1 瓦片等级转换分析

百度地图、谷歌地图等网页上的在线地图均为金字塔瓦片数据,因此本文离线地图数据是“金字塔瓦片”数据(影像数据)。谷歌在2005年推出瓦片(Tile)式互联网地图Google Maps,其基本原理是将地图预先按照比例缩放至不同的级别大小,然后将不同分辨率的整幅影像切割成固定大小的栅格图片,并按照一定的命名规则存放地图文件到服务器之中^[6]。目前主流的互联网地图服务商谷歌、百度等都有各自的瓦片命名规则,根据命名规则将生成后的瓦片进行编号,命名规则主要分为四种:谷歌XYZ、TMS、QuadTree、百度XYZ,X为瓦片横向索引,Y为纵向索引,Z为瓦片的级别。现将国内外最常用的谷歌和百度命名规则作详细分析。

(1) 谷歌地图瓦片

谷歌采用WGS84坐标系,是从左上角经纬度(-180,90)开始计算瓦片,在最高级1时,世界地图为1块,在第二级时地图分为4块,瓦片编号是向右和向下,没有负数。由经纬度转换的瓦片公式为:

$$X = \left\lceil \frac{\lambda + 180}{360} 2^Z \right\rceil \quad (1)$$

$$Y = \left\lceil \left(\frac{\ln \left(\tan \left(\varphi \frac{\pi}{180} \right) + \frac{1}{\cos \left(\varphi \frac{\pi}{180} \right)} \right)}{\pi} \right) 2^{Z-1} \right\rceil \quad (2)$$

式中: X 代表行号, Y 代表列号, l 为经度, j 为纬度, Z 为缩放等级, 因此得出瓦片名 X_Y_Z 。

(2) 百度地图瓦片

百度采用 BD09 坐标系, 该坐标系基于 GCJ02 (中国国家测绘局制定的火星坐标系, 由 WGS84 坐标系加密制成) 加密。百度地图瓦片坐标以墨卡托投影地图中赤道与 0° 经线相交位置为原点, 沿着赤道往左方向为 X 轴, 沿着 0° 经线向上方向为 Y 轴。百度地图还定义了百度平面坐标系, 其原点与瓦片坐标原点相同, 以 18 级瓦片为基准, 此时平面坐标的一个单位等于一个屏幕像素。平面坐标与地图级别无关, 即同一经纬度坐标在不同级别下平面坐标是一致的。经纬度坐标以平面坐标为中间量, 与瓦片坐标、像素坐标进行相互转换。在百度坐标系下由经纬度转换的瓦片名公式为:

$$X = \left\lfloor 2^{Z-26} \frac{\pi \lambda R}{180} \right\rfloor \quad (3)$$

$$Y = \left\lfloor 2^{Z-26} R \cdot \ln \left(\tan \left(\frac{\pi \varphi}{180} + \sec \left(\frac{\pi \varphi}{180} \right) \right) \right) \right\rfloor \quad (4)$$

公式符号代表含义与公式 (1) (2) 相同。在下文中, 基于上述计算公式推导出新的瓦片名、瓦片等级、存储格式公式, 并生成要求的文件目录结构。

瓦片地图数据在服务器上的存储形式是直接以文件目录存放, 另外可以使用空间数据库对其进行管理和存储, 由于本文数据量较大, 地图数据存储在与连接服务器的磁盘冗余矩阵中。瓦片影像数据直接以文件目录的形式存储在服务器上, 海量的金字塔瓦片数据不可能存放在一个文件夹下, 不利于文件的维护管理, 所以在每层文件夹下, 还需建立子目录。将所有层的图片存放在按照层数命名的文件夹里, 如第一层的文件夹命名为 0, 第二层为 1, 以此类推。每个缩放级别是一个目录, 每一列是一个子目录, 该列中的每个瓦片是一个文件。本文在第二章已下载默认 JPG 格式的原始瓦片, 图片所示为谷歌影像北京 18 级瓦片, 示例如图 4。

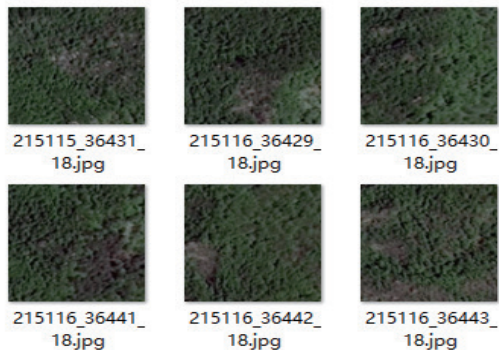


图 4 北京 18 级瓦片示例图

由于下载的图片名为默认格式, 而瓦片等级文件夹命名为自定义, 不能直接调用瓦片数据, 需要将瓦片数据存储重新转换成上述要求的文件存储结构。由图 4 可知, 18 级图片

名格式为 xxx_xxx_18 , 与本节计算公式所得瓦片名字格式一致, 在此代表瓦片的行号、列号和级别, 其中图片名带有瓦片所属等级 18, 因此可提取瓦片名动态创建对应级别的文件夹。同时, 若原始瓦片名不能适应不同客户端加载瓦片数据需求, 可以进行重新命名。

经分析, 转换公式为:

$$C = c - 2^{l-1} \quad (5)$$

$$R = 2^{l-1} - r - 1 \quad (6)$$

式中: C 为转换后瓦片的行号; c 为读取到的行号; R 为转换后瓦片的列号也为生成的列文件夹名; r 为读取到的列号; l 为读取的瓦片级别同时为生成的等级文件名。

首先遍历某省某级别的文件夹读取所有瓦片的名字, 分别对行列号和级别进行提取, 按照获得的级别号生成新的文件夹 (称为上一级目录), 根据计算公式获得新的列号和图片的命名, 按照新列号在上一级目录中生成新的子文件夹。并将瓦片进行重新命名存入该文件夹中, 生成符合加载条件的文件目录结构, 转换流程图如图 5 所示。

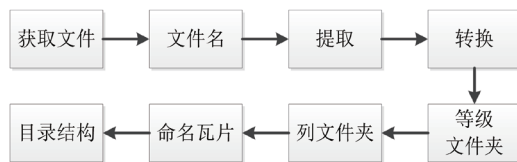


图 5 瓦片转换流程图

3.2.2 存储格式转换分析

瓦片数据的存储格式一般为 PNG、JPG、Mixed (混合模式) 格式。JPG 格式的瓦片为有损压缩且不支持透明背景, 所占存储空间相对较小; PNG 格式为无损压缩且可以设置背景透明, 所占存储空间相较 JPG 格式略大; 混合模式为两种格式同时存在, 具有存储空间小和支持背景透明双重优点。

针对瓦片存储格式要求不同的问题, 算法实现了对存储格式的转换。只需在遍历文件夹时过滤图像的格式, 选择目标转换格式, 最后结合 opencv 重新写出图像的新格式, 瓦片 4 名不变, 与上述瓦片等级转换流程基本相同。图像格式转换流程如图 6 所示。

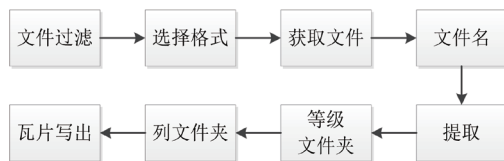


图 6 图像格式转换流程

3.3 算法实现结果

上述转换算法实现了对原始瓦片的等级划分、存储格式的转换, 利用 VS2010 结合 QT、opencv 进行算法的设计验证。

界面由 QT 实现，程序运行界面如图 7，点击菜单栏中不同的转换方式进行各格式的转换，通过点击各功能按钮进行界面操作，实现简单的交互。本文等级转换算法主要实现谷歌瓦片等级转换、高程数据等级转换、瓦片存储格式转换，其他转换实现同上述分析思路相同。



图 7 转换算法界面

4 Apache 服务器的搭建

4.1 服务器的搭建

Apache 是当前市场占有率最高且对外开放源码的网页服务器，支持 Linux、UNIX、Windows 等大多数计算机操作系统。它具有更广的兼容性和更高的安全性，因此被广泛使用，可以通过简单的接口进行扩展，同时具有开放源码、跨平台应用，支持多种系统、多种网页编辑语言，运行稳定和较高的安全性等一系列优点。

与传统使用共享目录或 U 盘相比较，搭建 Apache 服务器可以更好地适用于局域网内，随时随地调用地图数据，满足符合项目的开发要求。服务器搭建的步骤如下：

- (1) 根据电脑系统在官网下载个人需要的 Apache 服务器压缩包版本。
- (2) 待下载完成后解压，解压文件有两种选择：一是默认解压，二是自定义解压。本文选择第二种方式。
- (3) 修改配置文件后安装，参照解压文件中 ReadMe.txt 文档修改 httpd.conf 配置文件。
- (4) 运行服务器并测试服务器是否成功启动。
- (5) 默认端口为 80，可根据需要自行修改，本文服务器仅用于部署地图数据，因此不做修改。

4.2 磁盘阵列

在本文第二章，海量的瓦片数据和高程数据需要巨大的内存存储，服务器内存容量远远不够，因此采用磁盘阵列实现数据存储。磁盘阵列 (RAID) 实际是多个磁盘组合在一起形成的磁盘系统，常用的 RAID 等级方案有 RAID0、RAID1、RAID5、RAID6、RAID10。不同的等级提供了不同的性能、容错和成本优势，可根据实际需求进行选择。在本文的实际应用中，根据数据特点，考虑到性能与成本选择

RAID0 方式。

磁盘矩阵有 3 种常用的外挂存储方式，分别为：DAS、NAS、SAN。DAS (直连式存储) 通过电缆 (通常是 SCSI 接口电缆) 直接连接到服务器，能够大容量存储，具有较高的存取性能，操作简单，维护成本低，适用于小型网络环境；NAS (网络附属存储) 在网络中放置一个单独的存储服务器，此存储服务器开启网络共享，优点在于结构简单、易于实现、便于维护和管理，只需将 NAS 盘阵连接到 LAN，进行简单的配置即可实现数据共享，主要适用于多台服务器共享存储、数据量不大、数据存取时间比较分散且能够容忍一定访问时延的系统；SAN (存储区域网络) 需要连接单独的存储设备，例如使用交换机连接，且有管理方便、可扩展性好、容错能力强、可靠性高、传输距离远等优点。基于光纤通道的 SAN 可以根据要求进行配置，可以支持异构服务器，适用于环境中有多台服务器、多台盘阵、数据量大，且对容错、效率要求高的系统 [7]。结合实际考虑，本文数据采用 DAS 方式存储。

4.3 地图数据部署解析

地图数据存储在服务器上后，通过可视化工具将地图数据显示出来，验证数据是否部署成功且完整可行。为方便后续开发，本文基于 OSGEarth 框架完成地图数据可视化。OSGEarth 是基于三维引擎 (Open Scene Graph) 开发的三维数字地球引擎库，它提供了一种简单的方法来将大量的地理数据集成到 3D 场景中，并且支持各种数据源和插件。只需创建一个简单的 XML 文件，指向影响、高程和矢量模型信息，导入 OpenSceneGraph 中，即可实现渲染。

地图瓦片数据和 DEM 通过 XML 文件描述并导入 OSGEarth，使其按照数据所采用的瓦片等级和划分方法实现读取。信息包括：坐标系编号、文件类型设置、瓦片等级文件夹地址、瓦片等级编号、瓦片的像素和顶层瓦片所覆盖的经纬度范围等。XML 定义的实现代码如图 8 所示。

```
tms.xml - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<?xml version="1.0" ?>
<tilemap tilemapservice="http://imagery.osgeo.org/1.0.0" version="1.0.0">
  <abstract/>
  <abstract/>
  <srsEPSG:4326/>
  <vsrs/>
  <boundingbox maxx="-180.000000000000000000000000000000" maxy="90.000000000000000000000000000000" minx="-180.000000000000000000000000000000" miny="-90.000000000000000000000000000000" />
  <origin x="-180.000000000000000000000000000000" y="-90.000000000000000000000000000000" />
  <tileformat extension="tif" height="17" mime-type="image/tiff" width="17" />
  <tilesets profile="label-geoid">
    <tileset href="http://192.168.1.103:8080/elevation/0" order="0" units-per-pixel="10.703125000000000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/1" order="1" units-per-pixel="5.295625000000000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/2" order="2" units-per-pixel="2.647175781250000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/3" order="3" units-per-pixel="1.328789062500000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/4" order="4" units-per-pixel="0.664394531250000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/5" order="5" units-per-pixel="0.332197265625000000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/6" order="6" units-per-pixel="0.161098632812500000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/7" order="7" units-per-pixel="0.080549316406250000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/8" order="8" units-per-pixel="0.042746582031250000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/9" order="9" units-per-pixel="0.021373291015625000000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/10" order="10" units-per-pixel="0.010686645507812500000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/11" order="11" units-per-pixel="0.005343322753906250000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/12" order="12" units-per-pixel="0.002571661376953125000000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/13" order="13" units-per-pixel="0.001285830688476562500000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/14" order="14" units-per-pixel="0.000642915344238281250000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/15" order="15" units-per-pixel="0.000321457617191406250000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/16" order="16" units-per-pixel="0.0001985306884765625000000000" />
    <tileset href="http://192.168.1.103:8080/elevation/17" order="17" units-per-pixel="0.0000829153442382812500000000" />
    <tileset href="http://192.168.1.103:8080/elevation/18" order="18" units-per-pixel="0.00003145761719140625000000" />
  </tilesets>
</tilemap>
```

图 8 XML 实现代码

最后，将地图数据和 tms.xml 文件部署到服务器，服务器搭建完成，如图 9 所示。

0	2019/11/25 10:19	文件夹
1	2019/11/25 10:19	文件夹
2	2019/11/25 10:19	文件夹
3	2019/11/25 10:19	文件夹
4	2020/8/13 17:27	文件夹
5	2019/11/25 10:19	文件夹
6	2019/11/25 10:19	文件夹
7	2019/11/30 12:29	文件夹
8	2019/11/30 12:29	文件夹
9	2020/1/6 15:21	文件夹
10	2020/1/6 11:56	文件夹
11	2020/1/6 12:40	文件夹
12	2020/1/6 15:20	文件夹
13	2020/1/7 0:13	文件夹
15	2020/1/18 6:08	文件夹
16	2019/10/29 21:02	文件夹
17	2020/1/19 2:49	文件夹
tms.xml	2019/10/23 10:25	XML 文档

图 9 瓦片数据和 XML 文件存储

5 结语

基于 OSGEarth 对服务器中数据的访问，本文实现了瓦片和 DEM 的加载显示。图 10 为项目开发的系统界面，通过 XML 文件解析将地图数据加载到数字地球中，图 11 为加载显示的三维地图。

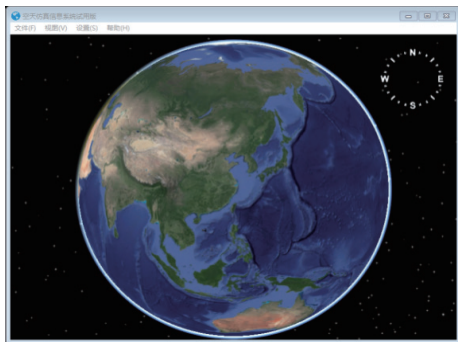


图 10 数字地球

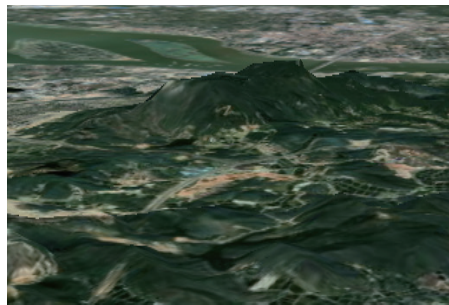


图 11 三维地图

本文提出的地图数据服务器构建方法，可以很好地满足不同项目的使用需求。在进行数据下载时自行选择瓦片等级和存储格式，根据下载的数据重新转换成适用于项目的文件结构，并实现了 PNG、JPG 等四种不同图片格式转换功能，最后搭建 Apache 服务器支撑局域网内每台主机对数据的访问加载。项目使用结果表明，该方法达到预期要求。

参考文献：

[1] 陈红华,李满春,李飞雪.基于 XML 信息交换的地图服务器构建[J].南京林业大学学报(自然科学版),2012,36(3):153-156.

[2] 李原野,张金芳,范植华,等.基于 Web 的地图服务器的设计与实现[J].计算机工程与设计,2006,27(22):4185-4188+4192.

[3] 马娟,邓富亮,石爱军,等.基于集群技术的高性能混合动态地图服务应用研究[J].水文地质工程地质,2015,42(2):145-150.

[4] 陆丽华.基于 Web 的矢量化地图服务器系统的设计与实现[D].西安:西安电子科技大学,2012.

[5] 刘耀龙,张蒙,李育珍.国外两种免费 DEM 数据对比分析[J].科技创新与生产力,2014(6):73-75.

[6] 陈磊.基于瓦片金字塔模型的矿山重大危险源监测系统设计与实现[D].武汉:武汉工程大学,2014.

[7] 卢中延.存储区域网管理系统中数据备份与复制技术的研究与实现[D].西安:西北工业大学,2003.

[8] 陆志烽.Linux 下 Apache 服务器的搭建[J].数字技术与应用,2016(5):100.

[9] 霍亮,杨耀东,刘小勇,等.瓦片金字塔模型技术的研究与实践[J].测绘科学,2012,37(6):144-146.

[10] 陈茂霖,万幼川,杨秋兰,等.面向多尺度影像的瓦片金字塔技术研究[J].地理空间信息,2013,11(4):72-75+187.

[11] 邹进贵,翟若明.OSGEarth 的 3D 场景可视化关键技术实现[J].测绘地理信息,2020,45(2):51-55.

[12] 李正学,许捍卫.异步非阻塞瓦片地图服务器的实现[J].测绘科学,2015,40(10):128-132.

【作者简介】

罗丽虹(2000—),女,江西赣州人,硕士,研究方向:计算机仿真。

孙卡(1982—),男,河南南阳人,博士,副教授,研究方向:计算机仿真。

(收稿日期:2024-08-22)