

基于 Web 的地形匹配系统设计与开发

高 香¹, 宋敦江², 梅 新¹

(1. 湖北大学 资源环境学院, 武汉 430061;

2. 中国科学院科技战略咨询研究院, 北京 100190)

摘要: 文章分析了当前图像匹配的应用领域以及地形匹配处理相关应用, 提出了构建基于 Web 的地形匹配系统体系; 地形匹配系统整个体系架构统由 web 网站、消息调度服务器、文档服务器、匹配处理服务四部分组成, 其中网站系统、消息调度服务器、文档服务器托管在阿里云服务器中, 匹配处理服务部署在另一台 GPU 服务器上面, web 网站通过消息传递机制与匹配处理服务进行通信, 实现 GPU 地形匹配计算与地图网站的功能分离; 文章对系统实现所依赖的软件环境进行了介绍, 实现了对匹配算法的封装, 并描述了地形匹配处理流程, 并对系统页面进行了设计, 并结合实例对系统地形匹配应用结果进行了分析。

关键词: 地形匹配; Web; 消息调度服务器

Design and Development of Web-based Terrain Matching System

Gao Xiang¹, Song Dunjiang², Mei Xin¹

(1. College of Resources and Environment, Hubei University, Wuhan 430061, China;

2. Institute of Science and Technology Strategy, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: This paper analyzes the current application field of image matching and the status quo of terrain matching processing application, and proposes to build a Web-based terrain matching system. The whole architecture of the terrain matching system consists of web site, message scheduling server, document server and matching processing service. The website system, message scheduling server and document server are hosted in Alibaba Cloud server, and the matching processing service is deployed in another system. On the GPU server, the web site communicates with the matching processing service through a message passing mechanism to realize the separation of the GPU terrain matching calculation from the function of the map website. This paper introduces the software environment on which the system implementation depends, implements the encapsulation of the matching algorithm, describes the terrain matching processing flow, and designs the system page, and analyzes the application results of the system terrain matching with examples.

Keywords: terrain matching; Web; message dispatch server

0 引言

地理信息系统 (GIS, geographic information system), 遥感 (RS, remote sensing) 以地理信息资源和测绘数据等为基础数据^[1], 包括大量的航空航天图片, 遥感影像, 地形图以及各尺度的 DEM (digital elevation model) 数据。这些地理信息基础数据具有与其它数据不同的特性, 拓扑关系强, 具有自相关性, 而且数据量庞大, 计算复杂, 处理困难。国内外学者将机器学习 (Machine Learning) 和深度学习算法应用于各领域大数据处理、分析, 获取有用信息^[2]。例如运用图像模式识别来进行大数据挖掘, 构建大数据智能地质学^[3]。

图像匹配技术是图像处理、分析的重要内容, 其结果的精度与可靠性直接决定了后续工作能否正常进行。图像匹配它是根据目标的特殊特征, 按一定相似性准则, 建立

未知图像与已知图像之间的匹配^[4]。随着社会信息化程度逐渐提高, 计算机存储和计算能力大幅度提升, 深度学习技术和方法在飞速发展, 图像匹配技术也趋于成熟, 图像匹配是几乎所有图像分析过程中的关键组成部分。图像匹配成功应用于例如导航、指导, 自动监视, 计算机视觉、绘图科学、无人驾驶、医学诊断交通监控等领域^[5]。

地形匹配是图像匹配的一种, 在远程巡航导弹导航、潜水器导航, 飞机航行跟随、地形回避, 模式识别, 地理定位, 目标跟踪等领域具有十分重要的作用^[6-7]。地形匹配制导自主、可靠、不受干扰、导航精度与航程无关, 无需在导航区域建立信号中继站等基础设施, 它不受气象条件和其他电子设备干扰等的影响, 是 GPS 导航的一种有效辅助手段^[8]。

地形匹配是利用已知地形数据, 从基准图中提取具有不变特征或明显特征的子区, 或者用已知地面控制点作为模板, 在所匹配的图中搜索与模板相似的区域。当两个地图的匹配相似性测度达到最大, 且超过预先规定的阈值时, 判定为找到了正确的匹配位置^[9]。在地形匹配过程中, 基准地形 (假定称为 A) 是全国或全球地形, 待匹配地形 (假定称为 T) 是任意一块具有明显地形特征的地形, 地形

收稿日期: 2019-02-01; 修回日期: 2019-03-06。

作者简介: 高 香 (1997-), 女, 河南商丘人, 主要从事地理信息科学 (GIS) 相关方向的研究。

通讯作者: 梅 新 (1973-), 男, 湖北武汉人, 副教授, 主要从事地理信息大数据、生态遥感等方向的研究。

匹配就是要在 A 中进行搜索或进行特征匹配, 确定 T 在 A 中的位置, 包括中心点坐标和边角拐点坐标。与地形匹配相关研究一般运用三维表面匹配算法、计算机视觉技术^[10]来确定基准地形 A 内恢复待匹配图形 T 的位置和方向。但到目前为止却很少有基于深度学习来进行地形匹配的相关研究。

尽管图像匹配技术发展快速, 地形匹配相关研究也很多, 但运用深度学习进行地形匹配相关研究却很少。将地形匹配数据, 相关匹配算法进行系统集成可以使地形匹配流程系统化, 能够对地形图进行快速匹配、更新。但由于地形数据数据量大, 计算复杂, 不能仅依赖于 CPU 的计算, CPU+GPU 的硬件框架更能满足实际需求, 由于 GPU 服务器租用昂贵, 从经济因素与算法安全性因素出发, 本文尝试 GPU 服务器和 WEB 服务器分离相互独立的做法。本文研究基于 Java 开发的地形匹配 Web 系统, 运用当今互联网、云计算等技术优势搭建的分布式在线处理系统, 解决了本地系统或局域网环境系统下的专业模型或处理算法在互联网上的应用, 提供了更广泛的共享^[11]。同时, 该架构采用了消息调度概念及应用模式, 对地形匹配算法进行了封装, 复用了已有的地形匹配算法, 开发效率大大提高。该系统托管在阿里云服务器中, 它通过消息传递机制与另外一台 GPU 服务器进行通信, 实现 GPU 地形匹配计算与地图网站的功能分离, 增加了系统的可扩展性, 方便进一步的升级。

1 地形匹配服务的架构设计

已有全国地形数据 A (30 米分辨率的 DEM), 对现有区域的地形等高线 (数据 A) (contourA.shp), 任意切割一个方形区域得到子集等高线数据 B, 对数据 B 进行平移和旋转得到数据 C, 对数据 C 随机加入噪音得到数据 T (contourT.shp)。地形匹配服务目标实现上传数据 T, 恢复或匹配、查找后确定数据 T 在数据 A 原始位置, 并反馈匹配结果。考虑算法的复用性, 以及进一步的改进与升级, 地形匹配服务在架构设计时, 采用基于“消息/订阅”模式的消息调度模式^[12], 对地形算法库进行了封装, 将网站平台模块, 消息队列传递模块等与地形匹配算法进行分离, 相互独立运行工作, 通过消息队列服务器建立网站与算法之间的通信, 构建了地形匹配服务体系。如图 1 所示。

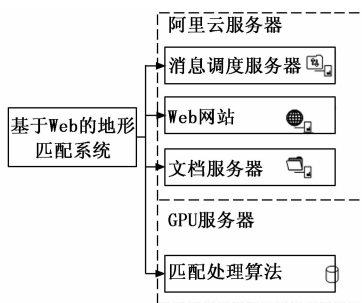


图 1 地形匹配服务架构体系

如图 1 所示, 地形匹配系统整个服务架构体系分为 Web 网站、消息调度服务器、文档服务器、匹配处理服务四部分, 其中 Web 网站、消息调度服务器、文档服务器部署于阿里云服务器中, 匹配处理算法部署于局域网环境。其中 Web 网站负责整个系统的前端的显示以及交互操作, 消息调度服务器、文档服务器负责门户网站与地形匹配算法服务的沟通, 匹配算法部分负责地形匹配运算。整个系统采取松耦合的模式, 彼此之间保持相对独立性。

Web 网站模块主要是网页部分, 负责为用户提供系统应用界面, 用户通过操作 Web 网站输入待匹配地形图片, 执行匹配命令, 网站获取地形匹配信息, 发起执行操作, 最后接收匹配处理后结果, 并储存到 Web 网站数据库中。

消息调度服务和文档服务器模块负责门户网站与地形匹配算法服务的沟通。消息队列调度服务^[13]可以简单的理解为系统各部分之间建立联系的“桥梁”, 它通过队列的方式实现各模块之间的消息传递, 属于系统的中间组件, 它将数据与计算进行分离, 确保了数据的独立性, 提高了算法的复用性, 便于后期算法的完善。文档服务器提供了消息监听机制, 后台匹配处理接收消息调度服务器发送的处理请求, 进行数据从阿里云文档服务器上下载。

匹配处理服务模块主要负责地形匹配算法的实现, 匹配处理后台需要集成消息队列接收处理消息, 包括文件名和数据库此数据的 id 号。通过 FTP 服务器 (File Transfer Protocol Server) 下载对应 save_name 的数据, 执行匹配处理, 然后通过 Web 网站 API 将数据结果及状态写入到数据库表中。

2 地形匹配系统的实现

2.1 软件环境搭建

网站部分通过 Java 体系开发, 采用 Play 框架, 消息队列采用 Rabbitmq 服务器, 后台空间匹配采用 Python 开发语言。考虑系统的安全性以及算法的重要性, 消息调度服务器, Web 网站和 FTP 服务器部署在阿里云上, 后台匹配算法部署在局域网内。系统实现流程图如图 2 所示。

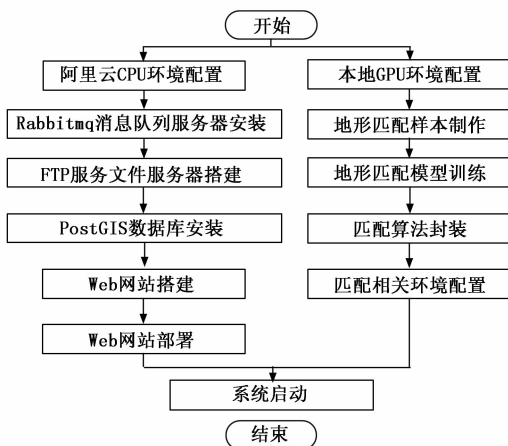


图 2 系统实现流程图

如图 2 所示, 在阿里云 CPU 环境中安装系统依赖软件, 并搭建 Web 网站, 配置相关环境, 在本地 GPU 环境中进行地形匹配算法的训练, 以及地形匹配算法封装及应用, 并配置相关环境。系统相关依赖软件如表 1 所示。

表 1 系统开发平台工具

依赖软件	开发工具
网站开发框架	Play
数据库系统	PostGIS
内存数据库	Pika
消息队列服务器	Rabbitmq
文件服务器	FTP 服务器
匹配后台算法	Python3 + pika + ftplib + 地形匹配算法等
网站部署环境	阿里云

如表 1 所示, 系统开发网站选择了 play 开发框架、Postgis 数据库、Pika 内存数据库、Rabbitmq 消息队列服务器、FTP 服务器、后台算法地形匹配算法使用 python3 语言开发、网站部署在阿里云环境中。

Play 框架 (<http://www.playframework.com/>), 它是开源 Web 框架, 相比于其他框架, 灵活性高, 具有热重载 (hot reload)^[14] 特性, 修改代码, 可直接刷新页面显示效果, 无需重新构建, 使用 Play 框架开发, 可以快速搭建网站, 大大提高了系统的开发效率。

PostGIS 数据库是对象关系型数据库系统 PostgreSQL (<https://www.postgresql.org/>) 的一个扩展, 能够有效的对空间数据进行管理与处理, 地形数据属于空间数据的一种, 为了便于存储空间数据的空间位置、空间关系, 选用了 PostGIS 数据库。

Rabbitmq (<http://www.rabbitmq.com/>) 是开源消息传递中间组件, 可在多平台, 多操作系统中运行, 具有可靠性, 使消息和消息队列具有可恢复性, 消息队列调度服务机制进行数据的传递, 将数据与后台算法进行分离, 确保了数据的独立性, 提高了算法的重用性^[15]。

Ftp 服务是通过 Internet 利用 FTP 服务器和 FTP 客户端实现的是一种高效可靠的文件传输服务, 客户端进行 FTP 会话, 服务器与客户端之间连接建立双向的传输文件的连接, 实现计算机与计算机之间实现相互通讯^[16]。此系统在阿里云上配置了 FTP 服务器, 建立了后台匹配处理与前端网站间的数据传输机制, 后台匹配平台接收消息调度服务器发送的处理请求, 连接到在阿里云主机上的 FTP 服务器程序进行数据的下载。匹配处理后将处理后的结果返回给前端网站。

Pika 是类似于 Redis 的一款开源的存储系统, 它的优势在于 Pika 是多线程的结构, 因此在线程数比较多的情况下, 某些数据结构的性能可以优于 Redis, 由于系统需要储存的为全国范围内的地形图, 数据量较大, 若选择 Redis 会因内存过大恢复时间长, 而 Pika 正好解决了用户使用 Redis 内存过大恢复时间长的问题。

匹配后台算法采用 Python 开发语言, 利用深度学习算法对全国地形数据进行训练, 之后模糊匹配, 再利用机器视觉方法进行精确定位。pika, ftplib 是 python 的内置的标准模块, Pika 包进行消息接收, ftplib 包提供了强大的对 FTP 服务器的操作, 通过它连接并操作 FTP 服务端使用。

地形匹配系统中消息调度服务器, Web 网站和 FTP 服务器部署在阿里云 (https://www.aliyun.com/?utm_content=se_1000301881) 上, 后台匹配算法部署在局域网内, 具体配置如图 3 所示。

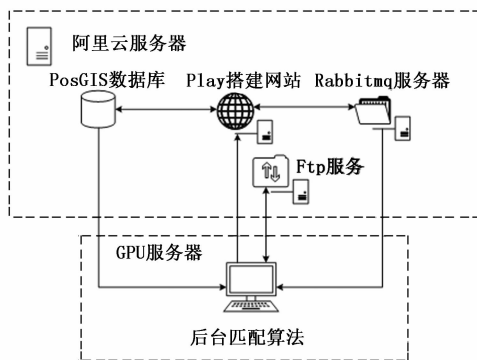


图 3 系统部署图

如图 3 所示, 地形匹配系统 Web 网站部署在阿里云上, 远程登录阿里云服务器, 进行 PostGIS 数据库、消息队列服务器 Rabbitmq、文件服务器 FTP 服务等依赖软件的安装。并运用 Play 框架搭建网站, 修改配置文件 application.conf 进行相关服务配置。修改配置信息如下所示:

```
file_path=["C:\zkydata\img\"]
save_file_path=["C:\zkydata\img\"]
mq = ["47.105.32.162"]
```

file_path 为输入文件路径, save_file_path 为保存文件路径 mq 为阿里云公网 IP 地址。然后构建消息队列服务, 新建队列 (queue), 命名为 web_pro1, 全国地形数据 A 存放于 PostGIS 数据库中。

接着进行匹配算法运行环境的搭建, 在后台匹配算法中进行 Rabbitmq 环境配置, 在匹配算法中输入消息队列登录名称, 以及登陆密码和 Rabbitmq 服务器配置网站 url, 并进行 FTP 环境的配置, 使后台匹配算法能够获取消息队列 message 信息, 需包括网站 url, 获取时间以及文件名称。

2.2 地形匹配算法封装

系统地形匹配的目标是实现输入任意变换后的地形数据 T 执行匹配命令, 匹配后得到 T 在全国地形数据 A 的具体位置。

地形匹配过程分成两个子过程:

1) 第一个子过程是先对总图 A 用深度学习的残差网络模型 (ResNet) 进行训练学习, 再进行模糊匹配, 找出子图 T 在 A 中的大概位置 T';

2) 第二个子过程是利用计算机视觉 OpenCV 里的尺度不变特征变换 (Scale-invariant feature transform, SIFT)

方法对 T 进行精确定位, 精确确定 T 与 T' 的相对位置。匹配结果示意图如图 4 所示。

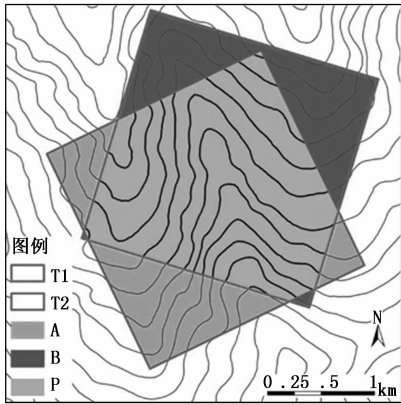


图 4 匹配结果示意图

如图 4 中所示 T1、T2 待匹配地形图为数据总图变换后的子图。P 为 T1、T2 重叠部分, A、P 为未重叠部分。

为了便于算法后期的完善和升级, 提高算法的复用性, 我们对地形匹配算法进行了封装。匹配算法通过集成消息队列接收消息体名称和队列名称, 获取处理请求, 得到文件名和数据库此数据的 id 号。通过 FTP 服务器下载对应 save_name 的数据, 执行匹配处理, 形成标准算法库。并将处理后的结果通过 Web 网站 API 将数据结果及状态写入到数据库表中。

算法库的封装, 实现了后台算法与前端网站的分离, 后台匹配算法只负责算法方面的开发, 前端网站只需通过消息调度机制与后台算法建立联系, 提高了匹配系统的效率, 以及系统的复用性。

2.3 地形匹配系统页面设计

地形匹配系统页面设计遵循简洁, 醒目的原则, 主要包括导航栏模块, 地图显示模块, 地形图片输入模块, 以及执行处理模块。如图 5 所示。



图 5 页面设计图

如图 5, 页面右侧模块设计用户输入待匹配地形图, 左侧设计执行匹配命令按钮, 并显示匹配信息, 中间模块显示地图, 并显示匹配成功结果。

2.4 地形匹配结果输出

地形数据经过地形匹配算法匹配后, 数据的输出形式为匹配地形图在参考全国地形数据 A 上的坐标信息, 即四

个顶点的经纬度信息, 并将坐标信息通过 Web 数据接口传递给前端网页, 并显示在前端网页地图中。

2.5 地形匹配处理流程

整个体系系统统分 Web 网站建设、消息调度服务、后台匹配处理服务三部分。消息调度服务器, Web 网站和 FTP 服务器在阿里云上部署。空间算法部署在局域网内, 用户通过操作 Web 网站浏览器页面输入待匹配的地形图, 执行匹配命令, 消息调度机制通过接收 Web 网站信息将处理请求发送给后台匹配处理, 匹配处理后台集成消息队列包括文件名和数据库此数据的 id 号接收处理消息, 然后通过 FTP 服务器下载阿里云环境中 PostGIS 数据库中对 save_name 的数据, 执行匹配处理, 处理完成后, 然后通过 Web 网站 API 将数据结果及状态写入到数据库表中并在 Web 浏览器页面中展示。具体过程如图 6 所示。

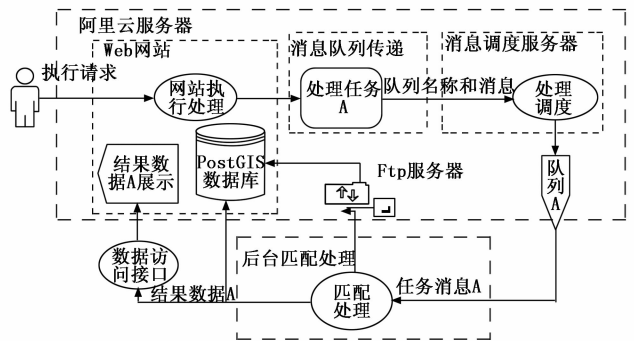


图 6 系统流程图

3 应用实例分析

3.1 实验步骤和方法

在实际应用中, 构建了基于 Web 的地形匹配系统, 浏览器页面由导航栏, 执行处理命令部分, 地图部分, 上传待匹配地形图四部分组成, 用户通过操作浏览器在页面右侧上传待匹配地形数据, 上传成功后, 点击页面左侧执行命令, 通过消息调度服务, 后台匹配算法进行处理, 将执行结果在浏览器地图上显示, 如图 7 所示。



图 7 匹配成功结果截图

图 7 中绿色虚线位置即为待匹配地形图 T 所在位置。

在地形匹配系统主要实现过程分为: 1) 样本数据制作; 2) 地形匹配算法模型训练; 3) 地形匹配算法模型应用。本文重点在于训练好的地形匹配模型的应用。

在全国地形数据 A 中随机提取一方形区域 T (300 行 * 300 列)。

300 列), 并对数据进行旋转得到 37 个样本数据, 其中包括 18 个副本数据, 随机提取 160 000 批数据, 得到实验数据 T 约为 600 万个样本数据。

以 600 万个样本数据作为训练样本, 利用深度学习的残差网络模型 (ResNet50) 进行训练, 进行模糊匹配, 随机选取 2 000 个样本数据作为测试样本测试精度, 每个训练样本返回 10 张地形图, 利用计算机视觉 OpenCV 里的尺度不变特征变换方法对 T 进行精确定位, 利用 ResNet50 训练模型, 匹配结果精度达到 97%。

将训练好的模型进行封装, 应用到地形匹配系统中。在阿里云 CPU 环境中, 搭建 Web 网站环境, 配置消息队列服务环境、FTP 服务器, 以及安装数据库系统等软件。在本地 GPU 环境部署匹配算法运行环境。首先远程登录阿里云服务器, 进行 PostGIS 数据库、消息队列服务器 Rabbitmq、文件服务器 FTP 服务等依赖软件的安装。并通过 Java 体系进行 Web 网站的搭建, 然后构建消息队列服务, 新建队列 (queue), 命名为 web_pro1, 然后将全国地形数据 A 存放于 PostGIS 数据库中, 之后将封装好的匹配算法匹配算法部署在本地 GPU 环境部署中, 在匹配算法中进行 Rabbitmq 环境配置, 在匹配算法中输入消息队列登录名称, 以及登陆密码和 Rabbitmq 服务器配置网站 url, 并进行 FTP 环境的配置, 使后台匹配算法能够获取消息队列 message 信息, 需包括网站 url, 获取时间以及文件名称。

在 Web 网站中点击上传图片或直接拖拽, 将实验数据 T 导入到网页中, 点击网页左侧执行匹配按钮, 通过消息队列将待匹配信息传送到后台, 匹配处理后需要集成消息队列接收处理消息, 包括文件名和数据库此数据的 id 号。通过 FTP 服务器下载对应 save_name 的数据, 执行匹配处理, 计算待匹配地形图在参考全国地形数据 A 上的坐标信息, 即四个顶点的经纬度信息, 然后通过 Web 网站 API 将数据结果及状态写入到数据库表中, 并将坐标信息通过 Web 数据接口传送给前端网页, 并显示在前端网页地图中。

基于 Web 的地形匹配系统采用了消息调度概念及应用模式, 对地形匹配算法进行了封装, 复用了已有的地形匹配算法, 实现 GPU 地形匹配计算与地图网站的功能分离, 增加了系统的可扩展性, 方便进一步的升级。

但系统同时存在不足, 例如地形匹配过程中上传地形数据必须和数据库中全国地形数据 A 同一比例尺才能进行匹配。

3.2 实验数据基础

系统匹配数据为全国地形数据 A (30 米分辨率的 DEM), 共计约 1000 块, 每块空间范围为 108 km * 108 km (3 600 行 * 3 600 列), 存放在 postgis 数据库中。

待匹配实验数据 T 由 Web 网站输入, 是从全国地形数据 A 随机提取一方形区域 T (300 行 * 300 列) 得到数据 B, 数据 B 经过旋转、平移等得到实验数据 T。

4 总结及展望

文章研究构建了基于 Web 的地形匹配系统, 对匹配算

法进行了封装, Web 网站通过消息调度服务器建立于后台匹配之间的信息传递与沟通, 三部分分工合作, 极大的提高了系统开发效率, 匹配算法基于深度学习的地形匹配 (Deep Learning-based Terrain Matching, DLTM), 匹配样本来自于数据集任意一块旋转, 变换, 加噪后的子集。相比于其他地形匹配研究, 文章有如下优点:

1) 文章将机器学习运用于地形匹配算法之中, 并对算法进行了封装, 可直接调用。

2) 文章将地形匹配数据, 相关匹配算法进行系统集成, 使地形匹配流程系统化, 能够对地形图进行快速匹配、更新。

3) 文章将 GPU 地形匹配计算与地图网站的功能分离, 将消息调度服务器, Web 网站和 Ftp 服务器部署在阿里云服务器, 匹配处理算法部署于局域网环境内本地 GPU 服务器中, 极大的降低了租用 GPU 服务器的成本, 同时确保了数据的独立性, 提高了算法的重用性, 便于后期算法的完善。

但文章中地形匹配算法无法应用于不同尺度地形数据的相互匹配和多分辨率的地形匹配。今后将对不同尺度地形数据的相互匹配、解决多分辨率的地形匹配、基于切割等高线方法的地形匹配、基于地学信息图谱的地形分类、辅助地形匹配、全球影像的自动定位等问题进行研究。

参考文献:

- [1] 何 沐. 地理信息行业计算机技术的发展现状及前景 [J]. 中国新技术新产品, 2018 (2): 26-27.
- [2] WANG L. Machine learning in big data [J]. International Journal of Advances in Applied Sciences, 2016, 4 (4): 117-123.
- [3] 周永章, 陈 铄, 张 旗, 等. 大数据与数学地球科学研究进展—大数据与数学地球科学专题代序 [J]. 岩石学报, 2018, 34 (2): 255-263.
- [4] 王红梅, 张 科, 李言俊. 图像匹配研究进展 [J]. 计算机工程与应用, 2004, 40 (19): 42-45.
- [5] Gruen A. Adaptive least squares correlation: a powerful image matching technique [J]. South African Journal of Photogrammetry, Remote Sensing and Cartography, 1985, 14 (3): 175-187.
- [6] 王广君, 房建成. 一种基于多特征融合的地形匹配 [J]. 航空学报, 2005, 26 (3): 340-343.
- [7] 张靖男, 赵兴锋, 郑志强, 等. 一种巡航导弹地形匹配系统研究 [J]. 战术导弹控制技术, 2006 (3): 56-59.
- [8] 张 涛, 徐晓苏, 李佩娟. 混沌优化水下地形匹配算法研究 [J]. 中国惯性技术学报, 2009, 17 (2): 156-158.
- [9] 曹跃云, 王广君, 柳 健. 基于神经网络的地形匹配方法研究 [J]. 海军工程大学学报, 2006, 18 (1): 51-56.
- [10] Rodriguez J J, Aggarwal J K. Navigation using image sequence analysis and 3-D terrain matching [A]. Proceedings. Workshop on Interpretation of 3D Scenes [C]. IEEE, 1989: 200-207.
- [11] 杨存祥, 张晓华, 黄布毅. Asynchronous operations and web services, Part 2. 电测与仪表, 2004, 41 (10): 51-53.

(下转第 235 页)