目 录

[前言 1](#_Toc39780594)

[第一章 绪论 2](#_Toc39780595)

[1.1研究背景和意义 2](#_Toc39780596)

[1.2 GIS的概述 2](#_Toc39780597)

[1.3 本文主要工作及切入点 2](#_Toc39780598)

[1.4本文的组织结构 3](#_Toc39780599)

[第二章 地理位置信息概述 4](#_Toc39780600)

[2.1地理位置数据 4](#_Toc39780601)

[2.2 OSM数据结构 6](#_Toc39780602)

[2.3 POSTGRESQL地理数据库 7](#_Toc39780603)

[2.4 POSTGIS中的空间索引 8](#_Toc39780604)

[2.5 最短路径算法 9](#_Toc39780605)

[第三章 需求分析 11](#_Toc39780606)

[3.1系统背景 11](#_Toc39780607)

[3.2系统目标 11](#_Toc39780608)

[3.2需求实现分析 11](#_Toc39780609)

[第四章 系统设计 13](#_Toc39780610)

[4.1开发环境准备 13](#_Toc39780611)

[4.2 系统流程图 13](#_Toc39780612)

[4.3辅助软件介绍 17](#_Toc39780613)

[第五章 系统实现 19](#_Toc39780614)

[5.1实现细节及步骤 19](#_Toc39780615)

[5.2测试结果 29](#_Toc39780616)

[第六章 总结 32](#_Toc39780617)

[6.1本文总结 32](#_Toc39780618)

[6.2未来展望 32](#_Toc39780619)

[参考文献 34](#_Toc39780620)

[致 谢 36](#_Toc39780621)

**摘 要**

近年来，随着云计算和移动互联网技术的飞速发展。新的用户场景和对于出行便利的要求使得大众对于地图或导航类应用的依赖越来越强烈。地图软件，导航软件，LBS社交，打车、拼车出行等这些应用程序为人们提供了极大的便利的同事，而且正在改变人们的生活。

自发地理信息（Volunteered Geographic Information，简称VGI）是指每个用户都可以通过在线合作互助的方式，以GPS手持终端设备、开放行的获取高分辨率遥感图像影像以及以个人空间的认知的地理出呗知识为基础参考，创建、编辑、管理、维护的地理信息。其中OpenStreetMap（OSM）是VGI的最著名来源之一。本文通过利用OpenStreetMap所提供的共享地理位置数据可以简易轻松的搭建出一个可靠的地理位置信息系统，并支持类似导航软件所提供的路网行车线路查询功能。在大型第三方地图服务商垄断的市场局面下，一定程度上提供了方便灵活的功能服务给到有此类需求的用户。

**关键词：**地理位置系统；OpenStreetMap；PostGIS；最短路径算法

# 前言

20 世纪以来，人们对地形图和各种专题地图的需求量迅速增加。立体航空摄影测 量和遥感成像技术的发展，使摄影测量工作者能以很高的精度，快速地进行大面积测 图，同时也为地球资源科学家们，如地质学家、土壤学家、生态学家等提供了极为优越 的条件来进行资源勘探和中等详细程度的制图工作，产生的专题地图已是资源调查和管 理最有用的信息源泉。

路径规划亦是近年来移动互联网流行的背景下的热门话题。在常见的地图，导航，自然灾害预计，无人机器的行进等多个商业领域都占有举足轻重的地位。例如共享经济下非常热门的拼车业务。如何在获取到繁杂的用车需求去正确且高效的匹配出最顺路的用户可以同乘一辆车。其中技术点包括：软件体系结构和数据通信技术，自动导航技术，智能计划和决策技术。其中，路径规划是此类研究的重要领域，属于基础建设。再例如大型电商平台京东近几年来推出的无人快递等业务，亦需要快递机器人在静态和动态障碍物并存的环境中，高效的找到从开始到给定终点的路径，以满足特定的评估标准，而整个进行过程中可以安全可靠地避开所有障碍物。本文亦探讨在现如今各大第三方地图服务商垄断的情况下，是否有可能靠比较小的代价和成本独自开发GIS系统。答案是肯定的。依靠OpenStreetMap所提供的开源的OSM地理位置数据，并结合PostgreSQL所支持的GIS插件功能。任何人都可以非常轻松的做出自己想要做到的GIS系统。诸如地址查询、线路规划，以及衍生出的拼车行程匹配等等，都为小团队或个人开发提供了一种可能性。

# 第一章 绪论

本章首先介绍了GIS系统的研究背景和意义，并在此基础上介绍VGI社区和OSM组织。其次介绍了最常见的两种最短路径法、数据源、数据结构等各个系统组成部分及其作用，并实践搭建出了一个简易的GIS系统，并概述了所做的主要工作和贡献以及创新点，在本章的最后介绍了论文的组织结构。

## 1.1研究背景和意义

自人类社会形成以来，人们在生产活动和社会活动中总在进行着信息的获取、交换和使用。从古代文明到现代社会，地理工作者、测绘工作者、航海家都致力于空间数据的收集整理，制图工作者则以地图形式表示这些数据。地图作为空间数据的载体长期为航海、军事以及现代经济建设服务。

## 1.2 GIS的概述

地理信息系统（Geographic Information System或 Geo－Information system，GIS）有时又称为“地学信息系统”。它是一种特定的十分重要的空间信息系统。在计算机硬、软件系统支持下，对整个或部分地球表层（包括大气层）空间中的有关地理分布数据进行采集、储存、管理、运算、分析、显示和描述的技术系统。

位置与地理信息既是LBS的核心，也是LBS的基础。一个单纯的经纬度坐标只有置于特定的地理信息中，代表为某个地点、标志、方位后，才会被用户认识和理解。用户在通过相关技术获取到位置信息之后，还需要了解所处的地理环境，查询和分析环境信息，从而为用户活动提供信息支持与服务。

由于地球是人类赖以生存的基础，地理信息系统作为一门与人类生存、发展和进步密切相关的信息技术，越来越受到人们的重视。地理信息系统按其范围可分为全球、区域和地方三种类型。一般来说，GIS主要研究地球表面几种元素的空间分布。它属于一个2-2.5维的GIS，在整个三维空间建立的GIS是一个真正的三维GIS。通常将数字位置模型(2D)和数字高程模型(1D)的组合称为2 + 1或3D，将具有时间坐标的GIS称为4D GIS或时态GIS。

**1.3 本文主要工作及切入点**

本文主要以OSM数据为切入点，分析了作为一个GIS系统所需要的各种知识储备和各种可行性分析。正是有此类高质量高活跃的公益组织的贡献，使得个人或者一个较小团队也能做出媲美GoogleMap、高德地图之类高质量的GIS系统。并在第五章进行实战的演练，初步实现了一个带有线路规划的简易版GIS系统。本文重点如下：

（1）认识GIS系统，了解到作为一个最基本的GIS系统应该具备哪些功能，如何实现。并在现阶段的移动互联网环境中起到的作用等。

（2）学习并了解地理数据库的特点，与传统文件式的相对比优点和缺点。以及了解到现阶段最新的地理数据库的发展现状，最常用的解决方案和使用成本。

（3）学习并了解现阶段VGI社区和OSM项目的发展现状。详细了解到此类的自发性社区是如何运作并且如何给到开发者便利去使用的。

（4）通过详细描述一个真实的GIS系统从数据准备，数据库设计，第三方支持的导入等维度，展示如何一步一步的搭建出一个真实的GIS系统。并在最后进行测试与验证。

## 1.4本文的组织结构

本文共分为五章，各章内容安排如下：

第一章：绪论。本章介绍了本研究背景及意义、GIS系统的如今社会行为中的重要意义和组成、本文的主要切入点等，最后介绍了本文的组织结构。

第二章：地理位置信息概述。本章主要指出了如何通过获取现有的地理位置信息取到并终点介绍OpenStreetMap的数据结构。并介绍了常见最短路径算法。分别解释了最流行的A\*算法和Dijkstra算法的过程及各自的优点和缺点。

第三章：需求分析，主要从用户及各开发解决视角去分析该系统应该如何定义功能及流程，详解需求步骤。

第四章：从系统架构层面设计数据库，前端系统。如何利用GIS插件所提供的函数直接实现既定的GIS系统功能。

第五章：通过实际项目案例操作，如何利用现有的VGI共享数据来搭建本地的地理位置数据库。并且通过PostGIS强大的支持能力。快速并可靠的搭建出一个媲美高德地图的简易GIS系统。实现基本的行车线路规划功能。

第六章：总结全文，分析此次课题中现有的不足之处，并提出未来工作的设想与展望。

**第二章 地理位置信息概述**

## 2.1地理位置数据

在公开开源的VGI贡献范畴中。最常见的收集地理位置信息的方式为移动设备的GPS芯片接受GPS信号。当移动设备的GPS芯片不能接收到GPS信号时，移动设备就需要与它所连接的手机信号塔通讯和估算它与信号塔之间的距离以不断报告它的地理位置。

美国科技博客下属研究机构BI Intelligence发表了一份报告，分析了地理位置数据是如何被收集的以及地理位置数据将如何改变整个移动行业。

现在，美国KPCB风险投资公司公布的最新手机调查报告，截止到2019智能手机数量已经超过24亿部，地理位置数据已经开始渗透到整个移动领域。除了象FourSquare和Shopkick那样的消费者应用之外，基于地理位置的服务在移动领域还有很多的应用机会。它可以支持广告和许多其他的服务，比如天气、旅游类应用。报告特别分析了基于地理位置数据的移动广告是如何激发消费者的兴趣，基于地理位置数据的功能是如何提高应用使用率，并且揭开了基础技术和隐私问题的神秘面纱。

纯粹的GPS解决方案以及它所生成的经纬度标签是地理位置数据的公认标准。但是至少还有4种方法可以获得地理位置数据：

（1）手机信号塔数据：当移动设备的GPS芯片不能接收到GPS信号时，移动设备就需要与它所连接的手机信号塔通讯和估算它与信号塔之间的距离以不断报告它的地理位置。通过这种方法获得的地理位置数据不如纯粹的GPS数据精确。

（2）WiFi连接：这是一种能够精确获得地理位置数据的方法，但是需要用到有效的WiFi热点。WiFi的地址与GPS坐标是一一对应的。它可以准确地标示出用户所处的位置，因此很多零售商才愿意提供免费的公共WiFi服务，那样它们就能够发布店内移动广告了。

（3）IP地址：地理位置也可以通过与数据连接有关的IP地址来获得。这种方法的准确性会因运营商而异，可靠性也比上述方法要低一些。

（4）用户报告：当用户注册电子邮件或移动应用和服务时，他们通常都会输入自己的地理位置和邮政编码。这些数据可以被转换为GPS坐标，从而建立起用户群的地理位置档案。

OpenStreetMap是一个开放式的在线地图协作平台，成立于2004年，从最初的小规模发展到2009年底的200,000多个贡献者。是一个存储海量XML 数据的数据库，只要注册账号，任何人均可以对其后台数据库进行编辑，从而被称为世界的维基地图。OSM的大多数数据是由“非专家”和“业余地理学家”收集的。尽管众源地图的编辑过程难以监控管理，但通过近年来的完善和修订，OSM的质量已经得到众多学者的认可。所有人都可以随意访问该组织所架设的服务网址（http://download.geofabrik.de/）下载各国的OSM地图，也可根据各自的需要下载。

VGI这类地理空间数据都是与志愿者贡献的“空间”类型的用户生成内容（UGC）相关的术语，一个世纪以来一直被大部分官方机构完全认可的功能。尽管公民的参与并不是什么新鲜事，但近年来它已成倍增长，这主要是由于一些重要技术的发展，例如Web 2.0，Google Maps，宽带通信，集成在照相机和智能手机中的更便宜的定位设备等。 出于许多原因而安排私人公民做出贡献的原因。 70％的计划已计入Elwood等人已经提到的清单中。尽管这类数据经常会遭很多的法规和精度问题，还有缺少正式的结构和质量控制程序，缺少元数据等。但主要的优势在于它们的数量，时间覆盖范围和其本地知识有关。贡献者比任何局外人都更了解周围的环境。所以截止目前为止仍然是绝大多数小规模GIS系统的数据来源的首选。

美国地质调查局（USGS）地震灾害计划于1999年左右启动的一项地震地图计划，旨在收集并提供有关人们对地震活动的感受和经验的信息，包括：受灾者的位置，救援区域等。根据成功的Wikipedia项目改编的Wikimapia是基于Google Maps的，是使用此类VGI平台的首批举措之一，通过该平台，具有Internet连接的人们可以通过绘制边界来选择世界地图上的任何位置，并提供描述和其他相关信息。

OSM是最著名和研究最多的VGI计划之一。这是由OpenStreetMap Foundation开发的项目，旨在向人免费提供地理数据，例如街道地图。用户可通过两种方式做出贡献：

（1）通过访问OSM网站并直接或使用桌面应用程序添加，编辑，删除或更新地理位置信息数据；

（2）使用具有GPS功能的设备直接从现场收集数据，然后再直接从应用程序或通过OSM网站将其上传到OSM服务器。两种方式都可以将元数据（如描述，名称等）与地理特征一起保存，并且可以随时编辑，删除或更新所有内容。数据的访问和使用可以通过以下多种方式进行：

* 通过访问具有自己呈现的OSM网站；
* 通过其他嵌入OSM地图的网站；
* 通过使用其应用程序编程接口（API）将OSM映射嵌入其自己的网站中；
* 通过下载原始数据或抓取完整信息包来实现自有GIS系统的搭建。

特别需要提到的是，OpenStreetMap此前一直使用属于创作共用（Creative Commons）协议的CC-BY-SA协议对OSM地图数据的采集和发布进行授权。但是被个人博客站点广泛采用的创作共用协议并非是数据共享的本意，特别是地图数据共享而设计。创作共用协议的设计初衷是用于创作内容的版权保护，主要包含文档、照片和绘图等互联网创作形式的版权保护，而协议本身对于互联网数据的采集和共享没有清晰权责界定。并且创作共用协议的创建者也不推荐将此协议用于信息数据库共享，包含教育或科研等用途的数据库共享。而后OpenStreetMap于2012年宣布，项目将正式采取Open Database Licence(ODbL)许可向地图数据使用者授权网站免费发布的OSM数据，包括点、线、面的坐标数据，地理位置间的相互拓扑关系，以及世界各地志愿者贡献的GPX轨迹数据。

还有一个不能忽视的问题就是在中国的法律上的问题。这是显然的，一定程度上也是OSM中国区不能很好发展的潜在因素。根据《地图审核管理规定》《地图审核程序规定》和《关于进一步加强互联网地图服务资质管理工作的通知》。

在中华人民共和国境内公开出版地图、引进地图、展示、登载地图以及在生产加工的产品上附加的地图图形的审核，应当遵守本规定。下列情况下，单位和个人（以下统称申请人），应当按照本规定向地图审核部门提出地图审核申请：（一）在地图出版、展示、登载、引进、生产、加工前；（二）使用国务院测绘行政主管部门或者省级测绘行政主管部门提供的标准画法地图，并对地图内容进行编辑改动的。受理地图审核申请时，受理窗口除要求申请人按照《地图审核管理规定》提交有关材料外，还应当提交下列材料：（四）引进的境外地图，需提交引进地图的相关证明材料；互联网地图服务资质单位应当使用经省级以上测绘地理信息行政主管部门审核批准的地图提供服务，并将地图审图号、测绘资质证书编号、互联网信息服务增值电信业务经营许可证编号共同标注在网站显著位置。

显然在国内使用OpenStreetMap数据并不能满足上述法律要求。这代表所有在中国境内想要从事这方面工作的从业人员来讲，获得合法许可是首要条件。

## 2.2 OSM数据结构

OpenStreetMap使用WGS84标准将坐标存储为代表每个点的纬度和经度的双精度浮点类型。 OSM数据的基础是球体，空间特征以“地理”坐标（有时称为“大地坐标”或“纬度/经度”）表示，以角度单位（度）表示。

OpenStreetMap包括空间数据以及属性数据。其中空间数据主要包括三种：点（Nodes）、路（Ways）和关系（Relations），这三种原始构成了整个地图画面。其中，Nodes定义了空间中点的位置；Ways定义了线或区域；Relations（可选的）定义了元素间的关系。属性数据Tags用于描述上述矢量数据基元。

**·**Node

node通过经纬度定义了一个地理坐标点。同时，还可以height=\*标示物体所海拔；通过layer=\* 和 level=\*，可以标示物体所在的地图层面与所在建筑物内的层数；通过place=\* and name=\*来表示对象的名称。同时，way也是通过多个点（node）连接成线（面）来构成的。

**·**Way

通过2-2000个点（nodes）构成了way。way可表示如下3种图形事物（非闭合线（**Open polyline**）、闭合线（**Closed polyline**）、区域（**Area**）。对于超过2000 nodes的way，可以通过分割来处理。

（1）Open polyline

非闭合线：收尾不闭合的线段。通常可用于表示现实中的道路、河流、铁路等。

（2）Closed polyline

闭合线：收尾相连的线。例如可以表示现实中的环线地铁。

**（3）**Area

区域：闭合区域。通常使用landuse=\* 来标示区域等。

**·**Relation

一个Relation是用来描述两个或多个基元的相互关系（nodes, ways 或者其他的relations），相互的关系通过role来定义，包括：

（1）route ：定义公路、自行车道、铁路等

（2）多个多边形：定义area例如建筑、河堤等

（3）边界：装门用来定义行政边界

（4）限制：用于描述限制比如“非左转”

**·**Tag

标签不是地图基本元素，但是各元素都通过tag来记录数据信息。通过'key' and a 'value'来对数据进行记录（了解xml或者数据库的应该都比较清楚了吧？）。例如，可以通过highway=residential来定义居住区道路；同时，可以使用附加的命名空间来添加附加信息，例如：maxspeed:summer=\*就表示夏天的最高限速。

## 2.3 POSTGRESQL地理数据库

地理数据库(也称为地理空间数据库)是空间数据基础设施的核心要素。与基于文件的空间数据库数据存储(通过地理信息系统访问)相比，其主要优势在于其结构上利用了关系数据库管理系统的现有能力，包括对SQL语句的支持和生成复杂地理空间查询的能力。此外空间查询是一种特殊的数据库查询类型，它由地理数据库结构支持。这些查询与SQL查询在几个重要方面有所不同。其中最重要的是，它们允许使用几何数据类型，如点、线和多边形，并且这些查询考虑空间的关系介于这些几何学支持SQL的典型数据库系统的索引支持快速数据检索，但这种方法对于空间查询并不理想。相反，空间数据库使用空间索引来加速数据库操作。空间索引使查询成为可行的大型数据成套设备在空间数据库中。如果没有索引，任何功能的搜索都需要对数据库中的每一条记录进行“顺序扫描”，效率非常低下。索引通过将数据组织到搜索树中来加快搜索速度，搜索树可以快速遍历以查找特定的记录。

PostgreSQL是一种特性非常齐全的自由软件的对象-关系型数据库管理系统（ORDBMS），是以加州大学计算机系开发的POSTGRES，4.2版本为基础的对象关系型数据库管理系统。POSTGRES的许多领先概念只是在比较迟的时候才出现在商业网站数据库中。PostgreSQL支持大部分的SQL标准并且提供了很多其他现代特性，如复杂查询、外键、触发器、视图、事务完整性、多版本并发控制等。同样，PostgreSQL也可以用许多方法扩展，例如通过增加新的数据类型、函数、操作符、聚集函数、索引方法、过程语言等。另外，因为许可证的灵活，任何人都可以以任何目的免费使用、修改和分发PostgreSQL。

PostGIS是PostgreSQL对象关系数据库系统的扩展，它允许将GIS对象存储在数据库中。PostGIS 包含对GIS对象进行基本分析的功能，更重要的是，它还支持空间索引方案。索引对于大型空间表非常重要，因为它们允许在查询期间快速检索记录。如果空间索引的检查是一项特别重要的任务，那么在分析大型数据集时经常使用PostGIS。同时空间索引也是作为地理数据库的一个重要组成部分，对于数据库的查询性能有着至关重要的影响。实验表明，随着表大小的增加，空间索引构建时间呈近似线性增加，但是，随着数据库大小的增加，如果不对空间查询进行索引，处理时间就会大大增加。

## 2.4 POSTGIS中的空间索引

索引是增强数据库性能的常用方法。索引使得数据库在查找和检索数据库的特定行的时候比没有索引快的多。但索引也增加了整个数据库系统的开销，所以应该合理使用。尤其是当作为地理数据库时，索引的选择和使用也至关重要。所以新诞生出一个概念名为：空间索引。空间索引是空间数据库的关键所在，空间索引强，空间数据库支持大规模的数据才有意义，可以说空间索引的好坏决定了数据库的强弱。

PostgreSQL数据库管理系统一般支持三种索引:

（1）B-树索引

（2）R-树索引

（3）GiST索引

B-树索引可以处理对可以排序成某些顺序的数据的等式和范围查询，例如常见的的 < > = 等运算符的比较时，应优先考虑使用B-树索引。但是它没有办法支持空间搜索中提到的几何对象的搜索需求（相交，包含，距离等），所以B-树索引一般在地理数据库中是不推荐使用的。

R-树索引曾经被用于一些特殊的GIS系统中，它可以将将空间数据按长方形组织，大的长方形套小的长方形。顶层是最大的长方形，接下来是很多小的长方形，再接下来更小，不停的切成多个层次。类似B-Tree的根、分支、叶子结构。

GiST其实是一种通用的索引接口。在最新版的PostgreSQL中的R-树索引已经被证实并不如GiST索引来的强健。除了GIS索引外，GiST还经常被用来加速搜索所有不适合普通B-树索引的不规则数据结构。并能够支持在一个索引结构中，存储平面、三维、多维的数据。

在PostgreSQL中，与R-树索引相比，GiST索引是空安全的。这意味着它们可以索引包含空值的列，PostgreSQL中的原始R-树索引并不支持这一点，所以在包含空几何图形的几何列上构建索引将失败。

## 2.5 最短路径算法

在图论中，最短路径问题是在图中的两个顶点（或节点）之间找到路径的问题，以使其构成边的权重之和最小。可以将在道路地图上找到两个交叉点之间的最短路径的问题建模为图形中最短路径问题的特例，其中顶点对应于交叉点，边对应于路段，每个边均由路段的长度加权分割。Dijkstra算法在不考虑目标信息的情况下搜索全局空间。导致求解时间长，难以满足快速路径规划的需要。

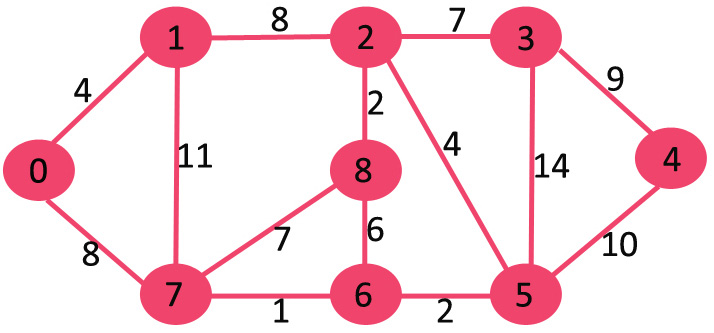
（1）Dijkstra算法

Dijkstra的算法（或Dijkstra的最短路径优先算法，SPF算法）是一种用于在图形中查找节点之间最短路径的算法，该路径可能代表例如道路网络。它是由计算机科学家Edsger W. Dijkstra于1956年构思的，并在三年后出版。

该算法的大致思想就是：根据初始点，挨个的把离初始点最近的点一个一个找到并加入集合，集合中所有的点的d[i]都是该点到初始点最短路径长度，由于后加入的点是根据集合S中的点为基础拓展的，所以也能找到最短路径。

对于在地图中给定了源节点后，该算法查找该节点与其他节点之间的最短路径，或者从单个节点到单个目标节点的最短路径。一旦确定了到达目标节点的最短路径，就可以使用该算法。

另外还有一个比较显著的特质，Dijkstra算法使用的标签需要是正整数或实数，它们是完全有序的。如果后续标签（遍历边缘时生成后续标签）单调非递减，则可以通用使用部分排序的标签，给定一个图和图中的一个源顶点，找到从源到给定图中所有顶点的最短路径。这种概括称为通用Dijkstra最短路径算法。

图2.1 Dijkstra寻路过程

在某些领域，尤其是人工智能，Dijkstra的算法或它的一种变体被称为统一成本搜索，并被表述为最佳优先搜索的更普遍概念的实例。

（2）A\*算法

A\*星形算法是一种路径规划方法，适用于已知全球环境信息的情况。它也适用于路径的二次编程。许多研究人员研究了A算法在路径规划中的用法。为了解决环境信息部分变化的情况下的路径规划问题，K.I。Trovato提出了差分A星算法。该算法的处理流程很复杂，需要大量的数学计算。公式表示为：

f(n)=g(n)+h(n)

f(n) 是从初始状态经由状态n到目标所处状态的成本估计，

g(n) 是在状态空间中从初始的状态到状态n的实际成本，

h(n) 是从n状态到目标所处状态的最佳路径的估计成本。

该算法的优势在于可以采用，修改或添加其他距离作为标准。这对该基本原理进行了广泛的修改。Raphael于1968年首次发布了该算法。作为Shakey项目的一部分，创建了A\*项目，该项目的目的是构建一个可以计划自己的动作的移动机器人。许多研究表明，在绝大多数情况下，A \*算法是最佳解决方案。

**第三章 需求分析**

本章将从一个实际的GIS系统开始分析，将从现有掌握的技术特点和实际相结合，着重分析出一个基本的GIS系统应该具有的哪些最基本特点和需要用到的技术点。

## 3.1系统背景

事实上现在市面上所有的大型第三方地图服务提供方，例如百度地图，高德地图都存在一个同样的痛点，那就是所有用户都必须先接入服务方提供的庞大的SDK，并且不能自由的选择自己所需要的最小单位的功能。例如寻路功能，高德地图只提供了极为庞大的导航API。通过本文实现GIS系统，用户可以快速规划多个线路的路径长度，借此可以实现一些常见的互联网场景，例如拼车寻顺路客户等。

## 3.2系统目标

通过利用现有OSM数据来源进行整合，实现一个可查询，可规划线路的GIS系统。并验证普通城区寻路、障碍物寻路、省级寻路三种情况生成的路径是否正确合理。

## 3.2需求实现分析

本系统基于OSM及PostGIS的组合下，可最低成本的实现目标，其中包括：

（1）通过OpenStreetMap组织下载中国地区地理位置信息。

（2）本机通过Postgresql创建数据库，并设计出一个地理空间数据库所需的表结构。

（3）将OSM数据导入到地理空间数据库，并生成路网拓扑。

（4）创建Node项目，前端页面使用mapbox呈现地图，在地图上随机选两点生成最短行车路径。

（5）由DBA进行地理空间数据库的架设，包括维护GIS插件、OSM数据的导入、定期更新最新OSM数据到数据库中。

（6）创建后端工程项目，建立数据库连接。开发API接口提供出地理位置数据库所包含的标准数据格式。

（7）创建前端工程项目，实现mapBox前端框架。该框架可以展示一个基本的地图。并支持在地图上展示指定经纬度，展示线路。

（8）最后可根据具体业务进行数据分析。例如根据经纬度展示地址、根据起点终点规划行车线路、根据不同的行程规划出行驶里程，相互匹配后可以得知哪些行程顺路与否。

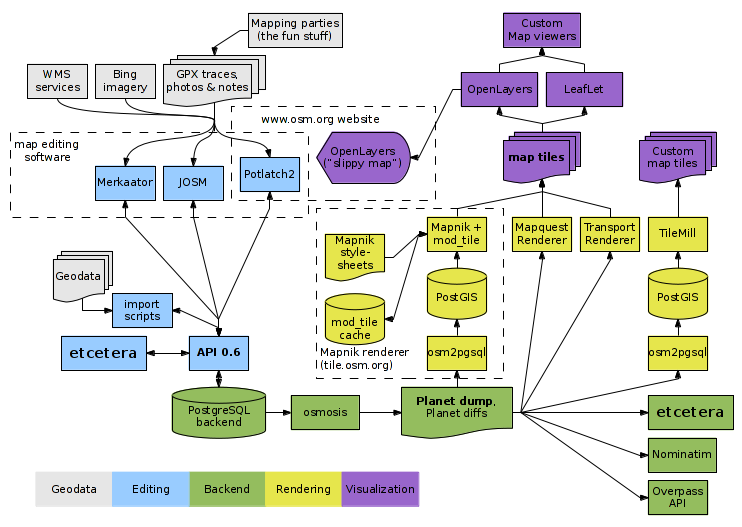
如图3.1所示为本系统用例图。

图地图上有字

描述已自动生成3.1 本系统用例图

**第四章 系统设计**

OSM wiki提供了一张架构图如图4.1所示，非常有价值，我们可以按此架构图一步步进行探索。

图4.1 OSM wiki推荐的标准GIS系统架构

Postgresql数据库配合上PostGIS插件非常适合用于存储地理空间数据，上面架构图的底层数据库也是此搭配，因此本文我们也使用此搭配。

## 4.1开发环境准备

本地环境为 macOS Catalina version 10.15.4。其余软件安装来源均由mac系统下的第三方软件管理工具HomeBrew提供。

（1）brew install postgresql # 版本 12.2

（2）brew install postgis # 版本2.5.0

（3）brew install osm2pgrouting # 版本2.3.6

（4）brew install pgrouting # 版本3.0.0

**4.2 系统流程图**

**地图上有字

描述已自动生成**图4.2 系统总流程图

图片包含 黑色, 电脑, 大, 白色

描述已自动生成依靠PostgreSQL插件可以自动生成一个完成的GIS数据库所需要的表结构，并不需要自行设计每一张数据表、结构等细节。如图4.3所示为完成表结构列表。

图4.3 GIS数据库表结构列表

本系统包含7张表用于存储地理位置元数据，32张表用于保存生成的GIS拓扑信息。下面对于7张主表结构进行介绍：

* planet\_osm\_line：包含所有导入的线路。
* planet\_osm\_point：包含所有带有标签的导入节点。
* planet\_osm\_polygon：包含所有导入的多边形路径。
* planet\_osm\_roads：包含“ planet\_osm\_line”的子集，适用于低缩放级别渲染。
* planet\_osm\_nodes：包含原始OSM节点数据（本研究中未使用）。
* planet\_osm\_rels：包含原始OSM关系数据（本研究中未使用）。
* planet\_osm\_way：包含原始OSM航路数据。下表的基准部分未使用此表这项研究，但是它被创建网络拓扑的功能所使用。

一并执行的Topology扩展则会额外添加两张表。

Network：包含网络线串拓扑数据。

vertices\_tmp：包含网络节点数据。该表是用于创建网络表的临时表。

其中最为关键的是表planet\_osm\_point、planet\_osm\_line、planet\_osm\_polygon。

如表4.1所示，planet\_osm\_point表包含带有已导入标签的所有节点。不导入没有标签的节点（如仅用于定义道路位置的节点）。

表4.1 planet\_osm\_point表结构

|  |  |  |
| --- | --- | --- |
| 列 | 类型 | 描述 |
| osm\_id\* | Integer | OSM文件中自带的唯一标识 |
| tag | text | 样式文件中指定的每个标签的一列包含标签值 |
| z\_order | integer | Z顺序（样式文件中指定），自动生成 |
| way\*\* | geometry | 几何节点 |

备注：其中的way实为WKB（well-know binary）形式。是一种用连续的二进制流来表达几何体的数据结构。在一般的现代GIS系统软件底层都有对于WKB的解析功能。

如表4.2所示，planet\_osm\_line表包含所有导入的非闭合的线路。

表4.2 planet\_osm\_line表结构

|  |  |  |
| --- | --- | --- |
| 列 | 类型 | 描述 |
| osm\_id\* | integer | OSM文件中自带的唯一标识 |
| tag | text | 样式文件中指定的每个标签的一列包含标签值 |
| z\_order | integer | Z顺序（样式文件中指定），自动生成 |
| way\_area | real | 区域（样式文件中指定），自动生成 |
| way\*\* | geometry | 几何节点 |

如表4.3所示，planet\_osm\_polygon表包含所有导入的闭合多边形区域。

表4.3 planet\_osm\_polygon表结构

|  |  |  |
| --- | --- | --- |
| 列 | 类型 | 描述 |
| osm\_id\* | integer | OSM文件中自带的唯一标识 |
| tag | text | 样式文件中指定的每个标签的一列包含标签值 |
| z\_order | integer | Z顺序（样式文件中指定），自动生成 |
| way\_area | real | 区域（样式文件中指定），自动生成 |
| way\*\* | geometry | 几何节点 |

如表4.4所示，network表以pgRouting期望的表格式包含路由网络拓扑。 PgRouting内置的多个重要函数均需要此表中的源和目标节点。

表4.4 network 表结构

|  |  |  |
| --- | --- | --- |
| 列 | 类型 | 描述 |
| gid | serial | 唯一标识 |
| osm\_id | integer | OSM文件中自带的唯一标识 |
| name | character | OSM文件中标识的路名 |
| source | integer | 由assign\_vertex\_id（）函数构建的起始节点标识符 |
| target | integer | 由assign\_vertex\_id（）函数构建的终节点标识符 |
| length | double | 线串几何的长度。 |
| way\*\* | geometry | 几何节点 |

电脑屏幕截图

描述已自动生成注：由于pgRouting函数仅依赖于表中的数据，因此不会为该表创建索引。

图4.4 GIS系统总ER图

**4.3辅助软件介绍**

本文中所有使用到的软件工具：

（1）PostgreSQL:最核心的数据库工具，由于其非常强大的数据库扩展能力故此选择作为本次数据软件的选择对象。

（2）PostGIS：是一个开源软件程序，该程序将对地理对象的支持添加到PostgreSQL对象关系数据库中。 PostGIS遵循开放地理空间联盟（OGC）的SQL简单功能规范。

从技术上讲，PostGIS是作为PostgreSQL外部扩展实现的。它的实现基于“轻量级”的几何形状和优化索引以减少磁盘和内存占用的空间。使用轻巧的几何形状有助于服务器增加从物理磁盘存储迁移到RAM的数据量，从而大幅提高查询性能。

（3）Osm2postgresql: 是一个Linux脚本，用于将OpenStreetMap数据加载到PostgreSQL数据库中，并在需要时设置服务器和数据库（PostGIS + hstore）。执行高级处理，简化渲染过程。

（4）Osm2pgrouting: 命令行工具，可利用已有的OSM数据生成路网拓扑。

Pgrouting:是基于开源空间数据库PostGIS用于网络分析的扩展模块，最初它只是利用Dijkstra算法实现最短路径搜索，之后慢慢添加了其他的路径分析算法，如A算法，双向A算法，Dijkstra算法，双向Dijkstra算法，tsp货郎担算法等，然后被更名为（5）pgRouting。该扩展库依托PostGIS自身的gis索引，丰富的坐标系与图形类型，强大的几何处理能力，如空间查询，空间处理，线性参考等优势，能保障在较大数据级别下的网络分析效果更快更好。

**第五章 系统实现**

本章内容为：利用OpenStreetMap所提供的地理位置信息数据，结合PostgreSQL强大的GIS扩展能力，最终实现一个任意两经纬度之间的路网最短路径生成功能的简易GIS系统。

## 5.1实现细节及步骤

本文选择使用命令行操作数据库。打开Terminal.app

1）创建名为ChinaOSM的数据库，注意一定要带上superuser,否则默认下没有修改权限。

createdb ChinaOSM -O superuser

2）对该数据库添加与GIS相关的扩展，

# 最核心的gis扩展

CREATE EXTENSION postgis;

# 增加 Topology扩展

CREATE EXTENSION postgis\_topology;

#增加Tiger用以支持模糊匹配

CREATE EXTENSION fuzzystrmatch;

#增加地理编码扩展

CREATE EXTENSION postgis\_tiger\_geocoder;

#用于存储属性tags，key-value

CREATE EXTENSION hstore;

# 增加pgrouting线路规划扩展

CREATE EXTENSION pgrouting;

运行后结果如图5.1所展示。若无任何错误提示代表所有插件的添加已经成功执行。

手机屏幕的截图

描述已自动生成图5.1 插件安装成功

当所有插件添加执行完成后，本地数据库会生成一个GIS系统中会用到的所有表结构。如图5.2所示为标记为‘public’的主表结构，用于存储OSM元数据。

手机屏幕截图

描述已自动生成图5.2 标记为public的表

访问 <https://www.openstreetmap.org/>下载GIS数据源。有几种方法可以获取我们想要的确切数据（整个地球Planet数据未压缩时超过617GB，显然我们不需要下载整个星球的数据）。。最便捷的方式是通过网站整理好的各主要国家或城市的聚合数据。

获取城市级别数据的更好方法是使用Metro Extract。 Mapzen提供的这项服务可为我们每周提供各种格式的所有主要大都市地区的摘要，以便快速，轻松地下载。对于像中国这样的大区域，最好下载Metro级别的文件。如果使用的是Mac平台，由于在撰写压缩本文时，.pdf格式的支持尚未编译到工具中。需要下载.osm或可用的.osm.bz2才可以在MAC平台上正确编译。

下载完成后的文件本质上是一个XML描述文件。其中主要以点、线、面的方式来描述地址坐标、公路、封闭山体湖泊等地理位置信息。并可以通过QGIS一类的软件加载并预览这类信息。（由于整个中国地区的信息过于庞大，本文不做展示）。

2）使用osm2pgsql工具将OSM资源文件内的地理位置数据导入到设置到的数据库中。

osm2pgsql -s -U wangyiji -d ChinaOSM /Users/mac020/desktop/path/osm\_my\_git/data/china.osm.pbf -H localhost -W;

图5.3为该命令执行的具体过程。

手机屏幕的截图

描述已自动生成图5.3 导入数据

3）所有数据导入成功后就可以使用osm2pgrouting对现有的地理位置信息生成路网拓扑。

//数据导入。其中第一段文件路径为OSM文件的具体路径，第二段的路径所指向的xml文件为所应用的pgrouting解码映射规则。其中包含了最原始的资源文件和数据库表及字段的对应关系。

osm2pgrouting --f /Users/alex.wang/Documents/GitHub/GIS/data/armenia-latest.osm --conf /usr/local/Cellar/osm2pgrouting/2.3.6\_1/share/osm2pgrouting/mapconfig.xml --dbname ChinaOSM --username wangyiji --clean

如果出现如图5.4界面，代表资源文件和数据库设置均正常。此时osm2pgrouting正在对文件内容进行解析。此过程需要等待较多时间。

手机屏幕的截图

描述已自动生成图5.4 生成路网拓扑

注：osm2pgrouting为一个命令行工具，用于将OpenStreetMap数据导入pgRouting数据库。它会自动构建路由网络拓扑，并为要素类型和道路类别创建表。osm2pgrouting主要是由DanielWendt编写。为pgRouting下属的子项目。

当执行到出现以图5.5所展示界面时，代表数据导入和网路拓扑已经全部完成。

电脑屏幕截图

描述已自动生成图5.5 拓扑生成完毕

最后就会用到pgrouting所提供的数据库内置函数pgr\_dijkstra来实现查询两点间最短路径的功能。

4）首先在地图上随机取两点。获取到起始点和终点的经纬度。如图5.6

地图的截图

描述已自动生成图5.6 提取测试坐标点

由于在地图上随机取到的两个经纬度不一定会正好匹配上在GIS数据库中所存储的node节点。所以需要先跟进经纬度查找距离最近的node节点坐标。执行以下SQL语句可得。

SELECT id FROM ways\_vertices\_pgr ORDER BY st\_distance(the\_geom, st\_setsrid(st\_makepoint(120.71597099304198,31.327612901966642), 4326)) LIMIT 1

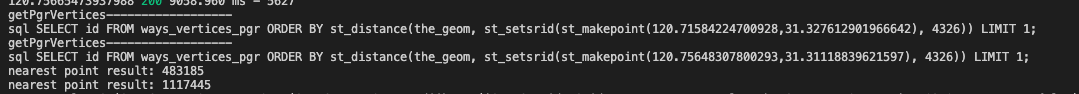
所查询到的id就是在本数据库中路网系统中最接近我们随机点击经纬度的坐标点。

图5.7 匹配node节点

由图5.7日志可以看出经过两次查询后，分别查找出了起始点和终点的nodeId。

执行pgrouting所提供的pgr\_dijkstra内置函数填入之前查到的起点和终点，即可查询出一条最短路径出来。图5.8所展示的为图5.6所提供的起点和终点之间规划出的最短路径点结果集。

社交网站的手机截图

描述已自动生成图5.8 途径经纬度列表

为了更直观的展示规划出的最短路径是正确并合理的。本章接下来内容为利用nodejs开发一个简易的展示系统。

配置项目内的数据库地址和数据库连接。

//数据库连接地址配置。

var path = require('path'),

var rootPath = path.normalize(\_\_dirname + '/../..');

server: {

port: process.env.OPENJS\_PORT || 8080,

hostname: process.env.OPENSHIFT\_NODEJS\_IP || '127.0.0.1'

},

database: {

url: process.env.OPENSHIFT\_POSTGRESQL\_DB\_URL || 'postgresql://127.0.0.1:5432',

dbname: process.env.OPENSHIFT\_APP\_NAME || 'routingonosm'

},

root: rootPath

//数据库连接方法。

function connect\_db() {

client.connect(function(err) {

if(err) {

return console.error('✗ Postgresql Connection Error. Please make sure Postgresql is running. -> ', err);

}

client.query('SELECT NOW() AS "theTime"', function(err, result) {

if(err) {

return console.error('✗ Postgresql Running Query Error', err);

}

console.log(result.rows[0].theTime);

});

});

在前端页面中导入mapbox框架作为地图显示定义地图点击方法，用以获取屏幕上所点击的经纬度。当获取到两点后即可开始route的寻路功能。

//地图点击事件，用以获取经纬度。

click: function(e) {

if(!ROSM.G.markers.hasSource()) {

var index = ROSM.G.markers.setSource(e.latlng);

ROSM.Geocoder.updateAddress(ROSM.C.SOURCE\_LABEL);

ROSM.G.markers.route[index].show();

ROSM.Routing.getRoute();

} else if(!ROSM.G.markers.hasTarget()) {

var index = ROSM.G.markers.setTarget(e.latlng);

ROSM.Geocoder.updateAddress(ROSM.C.TARGET\_LABEL);

ROSM.G.markers.route[index].show();

ROSM.Routing.getRoute();

}

}

规划的流程分为三步：

（1）获取前端页面map组件传来的经纬度，转换成数据库中最相近的node节点。此流程需执行两次，分别为起始点和终点。

// 根据传入的经纬度获取最近的node节点，并回调出去。

function getPgrVertices(reqPoints, callback) {

var getPoint = function getPoint(point, callback) {

var qStr = "SELECT id FROM ways\_vertices\_pgr ORDER BY " +

"st\_distance(the\_geom, st\_setsrid(st\_makepoint(" +

point.lng + "," + point.lat + "), 4326)) LIMIT 1;";

client.query(qStr, function(err,result) {

if(err) {

callback(err, null);

} else {

console.log('nearest point result: '+result.rows[0].id);

callback(null, result.rows[0].id);

}

});

};

（2）获取到标准node节点后，执行pgrouting所提供的pgr\_dijkstra函数来获取最短路径的所有途经点。

function getDijkstra(routesBeginEnd, callback) {

var routes = [];

// 执行pgrouting提供的'dijkstra'方法

var dijkstra = function(route, callback) {

var qStr = "WITH result AS (SELECT \* FROM ways JOIN (SELECT \* "+

"FROM pgr\_dijkstra('SELECT gid AS id, source, target, length AS cost "+

"FROM ways', "+route.begin+", "+route.end+", false)) "+

"AS route ON ways.gid = route.edge)"+

"SELECT ST\_AsEWKT(result.the\_geom), name from result;";

client.query(qStr, function(err, result) {

if(err) {

console.log(qStr);

return console.error('✗ Postgresql Running Query Error', err);

}

var r = parsingData(result.rows);

var queryStreets = collectLines(r);

callback(null, queryStreets);

});

};

async.map(routesBeginEnd, dijkstra, function(err, result) {

if(err) {

callback(err, null);

} else {

callback(null, result);

}

});

}

// 解析查询出的结果

function parsingData(data)

{

var x = 0;

var result = [];

for(var i = 0; i != data.length; i++)

{

var points = [];

var y = 0;

var tmp = data[i].st\_asewkt.split('(');

var tmp2 = tmp[1].split(')');

var tmp3 = tmp2[0].split(',');

for(var j = 0; j != tmp3.length; j++) {

var tmp4 = tmp3[j].split(' ');

var point = [tmp4[0], tmp4[1]];

tmp4[1];

points[y] = point;

y++;

}

var geoObj = toGeoJson(data[i].name, "LineString", points);

result[x++] = geoObj;

}

return result;

}

（3）前端map组件根据获取到查询完毕的线路数据。展示具体行程路径。

//展示路径

showRoute: function(response)

{

if(!response)

{

return;

}

if(!ROSM.G.markers.hasSource() || !ROSM.G.markers.hasTarget())

{

return;

}

if(ROSM.G.route.hasRoute())

{

ROSM.G.route.reset();

};

ROSM.G.response = response;

ROSM.RoutingGeometry.show(ROSM.G.response);

ROSM.RoutingDescription.show(ROSM.G.response);

ROSM.G.route.centerView();

},

## 5.2测试结果

图5.9中可以清晰的看到，从起始点到终点已经正确规划处了一条符合正常行车线路的最短路径。

人的地图

描述已自动生成

图5.9 常规规划

我们还需要本GIS系统支持多种道路类型和路况的规划功能。图5.10为测试跨越铁路线并上高架桥进行的线路规划。

地图的截图

描述已自动生成图5.10高架桥规划

至此，一个简易的GIS系统开发已经完成。通过对于不同的场景测试。OSM+PostGIS+pgrouting的搭配很好的做到了一个导航地图应有的基本功能。诸如此类的高德地图及百度地图在向第三方提供服务时，有时并不能很好满足所有客户的需求。此时如能快速可靠的搭建私有的GIS系统将会是一些中小型团队非常好的选择。

# 第六章 总结

## 6.1本文总结

本文主要先大概介绍了GIS系统的特性和基本概念及重要的组成部分。再详细解释并分析和A\*算法和dijkstra算法的实现方式和各自优缺点。并从OpenStreetMap此类的VGI社区便捷的获取到获取地理位置信息后实验性的编写了一个GIS系统并实现了点到点之前的最短路径查询。

OpenStreetMap项目是一个非常有趣并且非常有意义的开源项目。不但很好的发动了全球各国的VGI爱好者来补全全球的地理位置信息，并且配合PostGIS地理位置数据库能提供速开发和部署。OSM确实提供了一种非常便利的方式给到开发者。如果目前有任何的项目需要用到此类地图信息功能，国内绝大多数人第一时间会想到整合百度地图或高德地图，利用这些强大的第三方功能来实现自己的所有需求。但这其实也是一个问题所在。笔者在多年前参与的一个项目中也是遇到了此类的需求，第一时间想到的也是整合大的第三方地图服务公司，但之所以会研究本文所描述的这种方式也是出于这些第三方公司的打包服务太多，调研后发现几乎没有提供此类最纯粹的线路点及长度的功能。但采用本文所描述的方式，可以非常便捷的执行pgrouting所提供的st\_distance函数和pgr\_dijkstra函数来获取线路的长度和线路坐标队列。在此基础上亦可以完成很多其他的商业构想。例如热门的拼车软件最典型的一个场景：不同的用户会发布不同的拼车行程线路，如何在最短时间内找到最合适的乘客们合乘同一辆车，不至于乘坐一辆车的两位乘客的行程是天南地北完全不顺路的。解决此类场景无疑使用本文所描述的解决方案是最为合适的。

## 6.2未来展望

本文所描述的解决方案亦存在以下不足之处。

（1）OSM的数据的准确性需要提升。正是由于来源是由许多志愿者提供的，并且许多都是由志愿者的手持设备所采集的，所以在精度和准确度上有比较大的进步空间。有许多重要的结果表明。城镇很容易收集数据，而对农村地区或山崎的地形进行测绘则需要对土地覆被分类和严格的采样有所了解。OSM专家Haklay提出的一个专有名词称为“excitement of engagement”（参与兴奋）来绘制某些区域的背景下进行的，在这些区域中OSM志愿者急于绘制特征的情况可能会无意中代表了自然特征。

（2）本文并未涉及到数据更新方面的问题。由于目前中国的基础建设仍处于高速发展的阶段，一个地区的地理位置信息可能在短短几个月内就发生翻天覆地的变化。所以在数据的完整和即时性上，还是比不上国内一流大厂高德百度来的准确。这可能不仅仅是技术层面的问题，更多的是国家层面的授权以及信息安全层面需要考虑的问题。

（3）诸如OSM此类的VGI社区在中国严格来讲是不被允许的。本文也只是进行了学术层面的研究和讨论可行性方案，并不能也没有应用于商业用途。在国内进行任何的未被授权或私自进行的任何测绘活动都有可能被怀疑是“非法测绘”。从国家层面来讲地理信息涉及到国家安全，其中有敏感信息，比如军事设施，人防设施等。容易引起安全隐患。如能解决此问题，亦随着国家越来越开明的制度，相信该领域的从业者和创新点会越来越多，越来越好。

**参考文献**

[1] Ibrar Ahmed.Gregory Smith. PostgreSQL 9.6 High Performance[M], 2017

[2] 高慧君. 智慧的空间位置[M]. 测绘出版社, 2014

[3] 李浩. PostgreSQL查询引擎源码技术探析[M]. 测绘出版社, 2016

[4] Nicholas Terpstra. Mapping Space, Sense, and Movement in Florence. Routledge, 2016

[5] Project OSRM. http://project-osrm.org.

[6] PostgreSQL. https://www.postgresql.org/

[7] Pgrouting. http://pgrouting.org.

[8] PostGIS. https://www.postgis.net/

[9] C. Jung, D. Karch, S. Knopp, D. Luxen, and P. Sanders. Engineering Efficient Error Correcting Geocoding. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM Press, 2016.

[10] T. Kieritz, D. Luxen, P. Sanders, and C. Vetter. Distributed Time-Dependent Contraction Hierarchies. In P. Festa, editor, Proceedings of the 9th International Symposium on Experimental Algorithms (SEA’10), volume 6049 of Lecture Notes in Computer Science. Springer, May 2018.

[11] 刘光. 企业GIS案例分析与应用[M], 2017

[12] Michael F. Worboys,Matt Duckham. GIS : A Computing Perspective, Second Edition CRC Press[J], 2016

[13] Blazej Cipeluch, Ricky Jacob, and Adam Winstanley

Geotechnology Research Group. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Mapsm, 2017

[14] PhD John Krygier and Denis Wood. Making Maps[M]. The Guilford Press, 2015

[15] Haklay, M. How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets[J]. Environ. Plan. B, 2018

[16] Pascal Neis and Alexander Zipf. Analyzing the Contributor Activity of a Volunteered Geographic Information Project[J] The Case of OpenStreetMap, 2012

[17] 李妍妍. Dijkstra最短路径分析算法的优化实现[J].测绘与空间地理信息, 2014

[18] 张晓楠，任志国，曹一冰等.交通运输最短路径分析系统的设计与实现[J].测绘工程，2014

[19] Sui, D.Z. The wikification of GIS and its consequences: Or Angelina Jolie's new tattoo and the future of GIS. Comput. Environ. Urban Syst, 2018

[20] Mooney, P.; Corcoran, P. Using OSM for LBS—An Analysis of Changes to Attributes of Spatial Objects. In Advances in Location-Based Services; Gartner, G., Ortag, F., Eds.; Springer:Berlin/Heidelberg, Germany, 2010; Volume 34, Issue 6, pp. 496–507.

[21] 蒋波涛. 城市空间分析GIS应用指南[M].电子工业出版社, 2015

[22] 宋彦.城市空间分析GIS应用指南[M].中国建筑工业出版社, 2015

[23] 卜坤. Python 与开源GIS ——数据处理、空间分析与地图制图[M].清华大学出版社, 2019

[24] Stephen Wise, 朱定局译.GIS数据结构与算法基础[M].科学出版社, 2012

**致 谢**

在校的这段时间里很感谢老师们对我的谆谆教诲。指引着我们沿着正确方向前进。在点滴汇聚中是我逐渐的成长，并扎实的掌握了计算机方面的各项技能。特别要感谢我的指导老师程宝雷老师给予我很大的帮助。

在本文的撰写过程中，自始至终都得到了程宝雷老师的悉心指导，从论文选题、研究阶段的指定、技术线路的选择到系统的开发研制，各个方面都离不开老师热情耐心的帮助和教导。老师认真的工作态度，诚信宽厚的为人处世态度，都给我留下了难以磨灭的影响，也为我今后在校外的工作树立了优秀的榜样。

在此也要感谢我的家人，我的支持者们，正是在你们的支持下我才能一步一步的达到自己的目标，自己的期望。总是在我低落的时候为我打气，在我成功的时候与我一同庆祝。让我所有的一切都可以在你们这里得到理解与支持，得到谅解和分担。你们的支持和鼓励是我前进的动力。

最后想所有的亲人、师长、朋友们表示深深的谢意。