目录

[第一章 绪论 2](#_Toc38276793)

[1.1研究背景和意义 2](#_Toc38276794)

[1.2 GIS的概述 2](#_Toc38276795)

[1.3 本文的主要工作及切入点 2](#_Toc38276796)

[1.4本文的组织结构 3](#_Toc38276797)

[第二章 地理位置信息 3](#_Toc38276798)

[2.1地理位置数据 3](#_Toc38276799)

[2.2 Openstreemap简介 4](#_Toc38276800)

[第三章 最短路径算法 4](#_Toc38276801)

[3.1概念 4](#_Toc38276802)

[3.2 Dijkstra算法 4](#_Toc38276803)

[3.3 A\*算法 5](#_Toc38276804)

[第四章Postgresql数据库技术 6](#_Toc38276805)

[4.1 数据库简介 6](#_Toc38276806)

[4.2数据库特点 6](#_Toc38276807)

[4.3对于gis方面的可扩展性 6](#_Toc38276808)

[第五章 实践 7](#_Toc38276809)

[5.1工具介绍与环境准备 7](#_Toc38276810)

[5.2具体步骤 8](#_Toc38276811)

[5.3搭建本地数据库 10](#_Toc38276812)

[5.4设计数据结构导入数据 10](#_Toc38276813)

[5.5呈现在地图上 10](#_Toc38276814)

[第六章 总结 10](#_Toc38276815)

[6.1本文总结 11](#_Toc38276816)

[6.2未来展望 11](#_Toc38276817)

# 第一章 绪论

本章首先介绍了GIS系统的研究背景和意义，其次简单介绍了从算法、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。

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。

## 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。它是在计算机硬、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 本文的主要工作及切入点

数据是最主要的切入点，随着GIS数据及其各种应用（通常与原始数据集的目的不同）的可用性和可访问性不断扩展，GIS数据集的表征和质量评估变得越来越重要。生成，传输和利用的地理参考数据的数量不断增加，并且在空间数据库中嵌入的不确定性数量已成为具有至关重要的理论重要性和实际考虑因素的主要问题。同时，自愿地理信息Voluntary geographic information（VGI）成为地理信息的重要来源。 OpenStreetMap（OSM）是VGI的最著名来源之一。 OSM的大多数数据是由“非专家”和“业余地理学家”收集的，这引起了专业GIS社区对OSM质量的严重关注。OSM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。到2010年3月，英国OSM在英国的覆盖率已从去年的51.2％增长到69.8％。当导入活动扩展到导入政府或制图机构的空间数据时，只要可以在OSM许可证下使用，这两个数据上传到OSM的过程通常会迅速加快。本文的切入点为从openstreetmap直接导出整理好的数据文件，并通过Postgresql的GIS插件功能，可以高效快速的搭建出一个简易的GIS系统。

## 1.4本文的组织结构

# 第二章 地理位置信息

## 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。

## 

## 2.2 Openstreemap简介

osm是一个开放式在线地图平台，Open Street Map（简称 OSM）是一个存储海量XML 数据的数据库（在本文中简称原始 OSM 数据），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。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。

我们可以在网址（http://download.geofabrik.de/）下载各国的OSM地图，可根据各自的需要下载，下载完成之后将文件改为osm格式。

## 2.3 OSM数据结构

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。

### 2.1. 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。

### 2.2. 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。

  a）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。

  b）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。

c）Area

        区域：闭合区域。通常使用landuse=\* 来标示区域等。

### 2.3. 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：

       a）route ：定义公路、自行车道、铁路等

       b）多个多边形：定义area例如建筑、河堤等

       c）边界：装门用来定义行政边界

       d）限制：用于描述限制比如“非左转”

### 2.4. 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:winter=\*就表示冬天的最高限速。

# 第三章 最短路径算法

## 3.1概念

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。也可以为图定义最短路径问题，无论是无向图，有向图或混合图。在此为无向图定义。对于有向图，路径的定义要求通过适当的有向边连接连续的顶点。其中以Dijkstra算法和A\*算法最为常见。

## 3.2 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。

该算法存在多种变体。 Dijkstra的原始算法找到了两个给定节点之间的最短路径，但更常见的变体将单个节点固定为“源”节点，并找到从源到图中所有其他节点的最短路径，从而产生最短路径树。

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。例如，如果图形的节点表示城市，并且边路径成本表示通过直路连接的成对城市之间的行驶距离（为简单起见，请忽略红灯，停车标志，收费公路和其他障碍物），则可以使用Dijkstra算法寻找一个城市和所有其他城市之间的最短路线。最短路径算法的一种广泛应用是网络路由协议，最著名的是IS-IS（中间系统到中间系统）和开放式最短路径优先（OSPF）。它也被用作Johnson算法等其他算法的子例程。

Dijkstra算法使用的标签是正整数或实数，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。这种概括称为通用Dijkstra最短路径算法。

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。

Dijkstra的算法与Prim的最小生成树算法非常相似。像Prim的MST一样，我们以给定源为根生成SPT（最短路径树）。我们维护两组，一组包含最短路径树中包含的顶点，另一组包含尚未包含在最短路径树中的顶点。在算法的每个步骤中，我们都找到一个顶点，该顶点在另一个集合中（尚未包括在内），并且与源的距离最小。如图 3.2.1



图3.2.1

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。

## 3.3 A\*算法

A\*是一种图形遍历和路径搜索算法，由于其完整性，最优性和最优效率而经常在计算机科学中使用。[1]一个主要的实际缺陷是{{displaystyle O（b ^ {d}）} O（b ^ d）的空间复杂度，因为它将所有生成的节点存储在内存中。因此，在实际的旅行路线系统中，它通常比可以对图形进行预处理以获得更好的性能的算法[2]以及具有内存限制的方法要好。但是，在许多情况下，A \*仍然是最佳解决方案。[3]

斯坦福研究所（现为SRI International）的Peter Hart，Nils Nilsson和Bertram Raphael于1968年首次发布了该算法。[4]可以将其视为Edsger Dijkstra 1959年算法的扩展。通过使用启发式方法来指导搜索，A \*可获得更好的性能。

作为Shakey项目的一部分，创建了A\*，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。尼尔斯·尼尔森（NilsNilsson）最初提出使用图Traverser算法[5]进行Shakey的路径规划。[6] Graph Traverser由启发式函数{\ displaystyle h（n），} {\ displaystyle h（n），}指导，从节点{\ displaystyle n} n到目标节点的估计距离：它完全忽略{\ displaystyle g { n），} {\ displaystyle g（n），}从起始节点到{\ displaystyle n。} n的距离。 Bertram Raphael建议使用{\ displaystyle g（n）+ h（n）} {\ displaystyle g（n）+ h（n）}之和。[6]彼得·哈特（Peter Hart）发明了我们现在称为启发式函数的可容许性和一致性的概念。 A \*最初设计为在路径成本为边缘成本之和时查找成本最低的路径，但事实证明，A \*可用于为满足成本代数条件的任何问题找到最佳路径。[7]

1968年的原始A \*论文[4]包含一个定理，如果启发式函数是一致的并且适当选择了A \*的平局决胜规则，那么没有A \*类算法[8]可以扩展少于A \*的节点。几年后发表了“修正” [9]，声称不需要一致性，但是在Dechter和Pearl对A \*的最优性（现在称为最优效率）的权威研究中，这证明是错误的。的A \*的启发式算法是可允许的，但与可替代的类似A \*的算法相比，不能任意扩展更多的节点。

# 第四章Postgresql数据库技术

## 4.1 Postgresql数据库简介

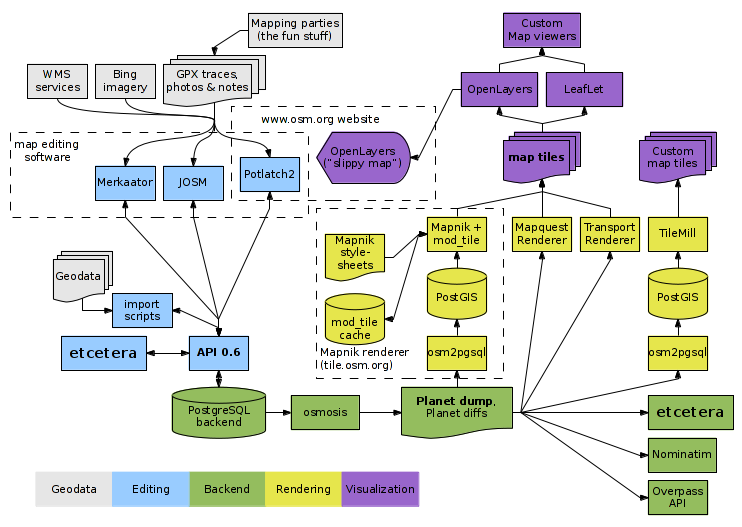
PostgreSQL是一种特性非常齐全的自由软件的对象-关系型数据库管理系统（ORDBMS），是以加州大学计算机系开发的POSTGRES，4.2版本为基础的对象关系型数据库管理系统。POSTGRES的许多领先概念只是在比较迟的时候才出现在商业网站数据库中。PostgreSQL支持大部分的SQL标准并且提供了很多其他现代特性，如复杂查询、外键、触发器、视图、事务完整性、多版本并发控制等。同样，PostgreSQL也可以用许多方法扩展，例如通过增加新的数据类型、函数、操作符、聚集函数、索引方法、过程语言等。另外，因为许可证的灵活，任何人都可以以任何目的免费使用、修改和分发PostgreSQL。

## 4.2数据库特点

## 4.3对于gis方面的可扩展性

# 第五章 实践

OSM wiki提供了张架构图，很有价值，我们可以按此架构图一步步进行探索。



Postgresql+插件PostGIS非常适合存储地理空间数据，上面架构图的底层数据库也是此搭配，因此我们也使用此搭配。

## 5.1工具介绍与环境准备

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的功能。

本文中所有会使用到的工具：

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。

PostGIS: 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。

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。

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

环境准备：

本地环境为 macOS Catalina version 10.15.4

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

//从brew安装postgresql

brew install postgresql

版本 12.2

//从brew安装postGIS

brew install postgis

版本2.5.0

//从brew 安装osm2pgrouting

brew install osm2pgrouting

版本2.3.6

//从brew安装pgrouting

brew install pgrouting

版本3.0.0

## 5.2具体步骤

1. 准备数据源，访问 <https://www.openstreetmap.org/>下载GIS数据源。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。注：如果使用的是Mac平台，在撰写压缩本文时，.pdf（压缩）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。

A screenshot of a cell phone

Description automatically generated

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。（由于整个中国地图的信息过于庞大，本文不做展示）。

1. 创建数据库并添加GIS插件

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;

-- Tiger用以支持模糊匹配

CREATE EXTENSION fuzzystrmatch;

-- 地理编码

CREATE EXTENSION postgis\_tiger\_geocoder;

-- 用于存储属性tags，key-value

CREATE EXTENSION hstore;

-- 线路规划

CREATE EXTENSION pgrouting;

运行后结果：

手机屏幕的截图

描述已自动生成

当所有插件添加执行完成后，本地数据库会生成所有GIS系统中会用到的所有表结构。

图片包含 黑色, 电脑, 大, 白色

描述已自动生成

3）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;

手机屏幕的截图

描述已自动生成

4） 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

如果出现如下界面代表资源文件和数据库设置均正确，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

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Daniel Wendt编写，现在托管在pgRouting项目站点上。为pgRouting下属的子项目。最新版本的pgRouting包含了以下最常用的最短路径的算法。

* All Pairs Shortest Path, Johnson’s Algorithm
* All Pairs Shortest Path, Floyd-Warshall Algorithm
* Shortest Path A\*
* Bi-directional Dijkstra Shortest Path
* Bi-directional A\* Shortest Path
* Shortest Path Dijkstra
* Driving Distance
* K-Shortest Path, Multiple Alternative Paths
* K-Dijkstra, One to Many Shortest Path
* Traveling Sales Person
* Turn Restriction Shortest Path (TRSP)

当执行到出现以下界面时，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5.2.2为最终的数据库ER图。

电脑屏幕截图

描述已自动生成

图5.2.2

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。

首先在地图上随机取两点。获取到经纬度。

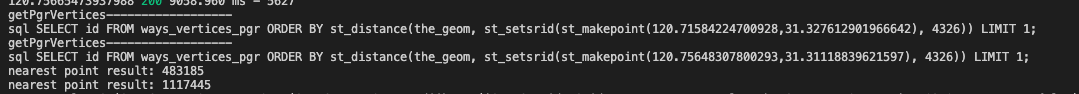
地图的截图

描述已自动生成

由于取到的是两个随机经纬度，所以不一定会正好匹配上我们GIS数据库内生成好的拓扑路网，所以需要先找到该经纬度附近最近的node点坐标。执行以下sql语句。

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的经纬度的坐标点。



由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。

最后，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。

社交网站的手机截图

描述已自动生成

将这些坐标点通过前端框架mapbox呈现出来就得到了以下效果。

地图的截图

描述已自动生成

# 第六章 总结

## 6.1本文总结

## 6.2未来展望