

中文摘要

摘要：随着社会经济的不断进步，科学水平的飞速发展，世界上大多数国家都已经建立的四通八达的交通网。与日俱增的公路交通系统的复杂性和拥挤程度，使得路径规划系统和地理信息系统显得格外的重。欧美的一些发达国家在路径规划技术和“数字城市”的研究领域一直处于领先水平。虽然我国在这个领域的起步比较晚，但是在该领域中的发展却很迅速，取得成绩也很突出，在一些大城市已经具备了欧美国家相应的技术水平。

本文首先介绍了电子地图在地理信息系统中起到的重要作用，电子地图的主要特点和两种典型的拓扑结构模型；其次对制作电子地图的工具软件 MapInfo Professional 的特点进行了介绍并且对 MapInfo 格式的电子地图的数据组织结构以及绘制电子地图中所涉及的必要的理论知识进行了阐述，而且展示了用 MapInfo Professional 制作电子地图的全部重要步骤；然后对 MapX 这款 GIS 控件的主要优点和突出的功能进行了讲解和分析，对 MapX 控件和 MapInfo Professional 软件之间的区别和联系进行了详细的说明；之后又描述了电子地图中道路的模式：节点—路段模式并且根据图论中的相关知识解决了路段的方向性问题；随后介绍了两种静态路径规划算法：Dijkstra 算法、A*算法，并且根据实验过程中遇到的实际情况，提出了改进的方法；最后通过软件演示的形式展示了路径规划算法和 MapInfo 格式的电子地图在导航软件中的应用，依据道路的不同情况，对每种情况下的相同两个节点之间的路径规划结果予以比较，同时对整个软件的电子地图的浏览、编辑、图层控制等功能也有详细的讲解。

本文的主要目的是根据地理信息系统和路径规划算法的理论知识，选择用 MapX 控件和 Dijkstra 算法，在 VB 的开发环境下实现一款导游软件。通过不断的努力，解决了在软件开发工程中遇到一系列的问题，从而使理论的知识和实际的应用有效的结合起来。使得实验中的导航软件在复杂的情况下能够有着良好的表现。本文实现的是静态的导航功能，今后在本文的实现结果上进一步进行无线功能的开发可以实时地解决动态导航的功能。

关键词：路径规划；MapX；电子地图；Dijkstra 算法

分类号：U491.13; TP391

ABSTRACT

ABSTRACT: With the continuing development of the social economy and science, most countries in the world have built intense transportation nets. Since the increasing complexity and congestion of the transportation system, Route Planning System and Geography Information System have become so important and necessary. Some developed countries in Europe have taken advantage in the field of Route Planning and Digital City. Although our country got involved in this area during recent decades, we have gained huge successes and experience. Some megalopolises have possessed the advanced technologies, which developed countries have.

Firstly, this paper introduces the significant role of Digital Map in GIS, the features and two classic kinds of the topology model about Digital Map. Secondly, introduce the MapInfo Professional, which used to make the digital map, it expatiates the data structure of the map and the knowledge, which seemed necessary in making a digital map, and display the process how to use the MapInfo Professional to make a digital map. Thirdly, analyses the merits and functions of the MapX that used in this paper, show the relationship between MapX and MapInfo Professional. Later, describe the model of road in digital map, then figure out the problem of directions of roads, explain the static Route Planning Algorithms: Dijkstra, A*. According to problems encountered in the actual situation, improve the methods proposed. Lastly, use the navigation software to demonstrate Dijkstra algorithm and the application of the digital map, compare the different result of route planning for some situations of roads, the software accomplish these map function: browse, edit, control layer of the map.

The purpose of the paper is using MapX and VB language to develop a navigation software based on theoretical knowledge of GIS and Route Planning Algorithms. Resolving these problems encountered in developing procedure, So that theoretical knowledge and practical application have effective integrated. The paper implements the static navigation, in the future the realization of further development of wireless capability can solve the real-time dynamic navigation function.

KEYWORDS: Route Planning; MapX; Digital Map; Dijkstra Algorithm


CLASSNO: U491.13; TP391

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年 月 日

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期：

年 月 日

致谢

本论文的工作是在我的导师李兴华副教授的悉心指导下完成的，李兴华老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢两年来李兴华老师对我的关心和指导。

荆涛副教授悉心指导我完成了实验室的科研工作，对于我的科研工作和论文都提出了许多的宝贵意见，在此向荆涛老师表示衷心的感谢。

在实验室工作及撰写论文期间，杨柳、戴庆国、刘万志、吕思凡、农光壹、张晓星等同学对我论文中的研究工作给予了热情帮助，让我度过了一个有意义的生活，在此向他们表达我的感激之情。

另外也感谢我的父母，他们的理解和支持使我能够在学校专心完成我的学业。

1 引言

1.1 智能交通系统概述

智能交通系统^[1](Intelligent Transportation Systems)简称ITS。ITS是将先进的信息技术(包括数据通信、计算机等)、传感器技术、自动控制理论、运筹学、人工智能等有效地综合运用于交通的运输、服务控制和车辆制造等方面,加强车辆、道路、管理者三者之间的联系,从而形成的一种实时、准确、高效的综合运输系统,最终使交通运输服务和管理智能化,使路网上的交通流运行处于最佳状态,改善交通拥挤和阻塞,最大限度地提高路网的通行能力,提高整个公路运输的机动性、安全性和生产效率。

随着汽车发展的社会化、工业化、信息技术智能化的实现,随着人类环境的可持续化发展的提出,随着道路堵塞和交通事故的急速增长,ITS是现代交通发展到一定阶段必然出现的产物。从20世纪60年代末开始,世界各国的交通工程师逐渐利用飞速发展的电子、信息、系统工程等科技手段来改善交通状况,将信息技术和交通系统结合起来研究交通四要素的时空关系。大约经历了20余年的时间,相距在世界范围内建立了新型交通系统,即智能交通系统,并且已经在许多国家和地区取得了良好的经济和社会效益。

由于各国具体情况不同,发展交通的重点也不一致,对于ITS研究的内容也不相同。在美国,按照服务功能和用户需求将ITS的研究内容分成七大类,相应地,美国的ITS开发项目以及实际应用系统也分为七个部分^[2]:

- 1) 先进的交通信息服务系统(ATIS)
- 2) 先进的交通管理系统(ATMS)
- 3) 先进的公共交通系统(APTS)
- 4) 先进的车辆控制系统(AVCS)
- 5) 货运管理系统(CVO)
- 6) 电子收费系统(ETC)
- 7) 紧急救援系统(EMS)

ITS作为新型的交通系统,ITS除了具有传统交通系统的特点外,还具有智能化、信息化和一体化的特点。其智能化的特点体现在以下几个方面:交通基础设施智能化、交通工具智能化、交通系统智能化。

ITS信息化的特点主要是有关交通要素的所有信息可以为交通系统的提供者、

维护者以及使用者共享。ITS 一体化的特点是指道路、车辆驾驶、乘客服务和系统管理的一体化。

我国ITS的发展起步较晚,20世纪70年代以来,从国外引进了一些项目,并进行了一些ITS或类似ITS基础项目的研究和应用。20世纪70年代中至80年代初,主要是进行城市交通信号控制试验研究,20世纪80年代中至90年代初,在一些大城市引进城市交通信号控制系统,实现了一些公路监控系统、高等级公路电子收费系统和路边信息服务系统,如广佛高速公路监控系统、首都机场高速公路电子收费系统。20世纪90年代中以来,开始研究部门ITS发展战略和地理信息系统(GIS)、全球定位系统(GPS)在交通中的应用,重视交通信息网络的建设,如交通部的公路智能运输系统发展战略研究、铁道部TMIS、DMIS等信息系统开发。目前,国内的研究与开发还都是某一地区或城市进行的,全国范围的ITS研究计划正在准备进行中。

总的来说,我国政府部门非常重视ITS的研究与开发,科技部门已经正式将ITS列入了中国高新技术开发和产业化计划,而且协同交通部、公安部、建设部、铁道部、信息产业部和许多高校及研究机构共同开发与研究。

1.2 车辆定位导航系统综述

车辆定位导航系统利用计算机和通讯技术,向行驶在道路上的车辆提供信息,引导车辆避开拥挤路段,沿最佳的线路到达目的地,它是智能化交通系统(ITS)中效益显著、见效快的项目,是欧、美、日等国争相研究与开发的重点^[3]。车辆导航系统同交通管理与控制系统融合,可以在大范围内进行交通流诱导,从而缓解道路交通拥挤状况;还可以向火警车辆、救护车辆与紧急救援车辆提供最短路径引导服务;此外,导航系统的应用将大幅提高道路通行能力,减少交通事故,节省能源与时间,对解决当前各国面临的严重的交通问题有着重要的意义。

1.2.1 车辆导航系统的组成模块

目前的智能交通系统是以道路和车辆作为主要的研究对象,以提高道路的通行能力、利用效率与安全行为为主要研究目标的新一代交通运输系统,重点是公路交通问题。智能交通系统基本是由四部分组成:管理和控制中心模块、车辆模块、道路模块和通信模块^[4]。如图1-1所示:

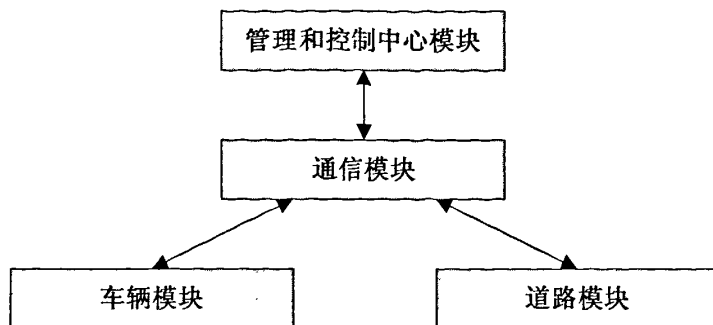


图1-1 智能交通系统的基本组成

Fig 1-1 Intelligent Transportation Systems Components

- 1) 管理和控制中心模块根据系统采集到的车辆和道路模块信息，进行交通管理和规划，实现对车辆的跟踪、调度和管理，完成收费管理和服务信息的发布，应急措施的安排和管理等工作。
- 2) 车辆模块用于根据各种传感器的信号，计算车辆的定位导航信息，以保证车辆按照调度命令或路径规划命令行使。
- 3) 道路模块用于提供实时的道路交通信息，完成道路交通状况的检测，行使电子收费、车辆管理和检测等功能。
- 4) 通信模块用于实现以上模块之间的数据、语音、服务信息和调度命令的传递和交互。

车辆模块作为智能交通系统的重要组成部分，其导航技术一直是许多国家研究的重点。车辆定位导航系统的功能模块如图1-2所示：

- 1) 数字地图数据库包含预先定义好存储格式的数字地图信息，其存储格式有助于计算机处理与地图有关的信息，如辨识场所、公路等级、交通规则。
- 2) 定位模块根据传感器的输出，自动地确定车辆的位置。典型的独立定位技术也是航迹推算定位，而典型的无线电信号定位技术是使用全球定位系统(GPS)定位。
- 3) 地图匹配是把测量到的或从定位模块获取的位置(轨迹)与地图数据库所提供的地图的位置(路径)进行匹配来确定车辆在地图上位置的一种方法。如果数据库精确，这种技术能改进定位模块的精度。对于市区来说数字地图的定位应该保证在15m以内。
- 4) 路径规划是帮助司机在行使前或运行中规划路线的过程，通常采用的技术是找到最小代价旅行路线。旅行代价可以是时间、距离等等。
- 5) 路径诱导是指司机沿着路径规划模块计算出的路线行驶的过程。为了确定车辆当前的位置和产生实时指令，需要借助地图数据库和准确的定位。
- 6) 人机接口允许用户与定位和导航计算机进行人机交互，将用户的要求通过

人机接口输送到计算机中，然后再通过人机接口将结果反馈给用户。

- 7) 无线通信模块是通过通信网络，使得车辆和它的使用者或者交通管理系统能够接受实时交通信息和报告，从而促使车载系统或整个公路网络工作的更加安全和有效。

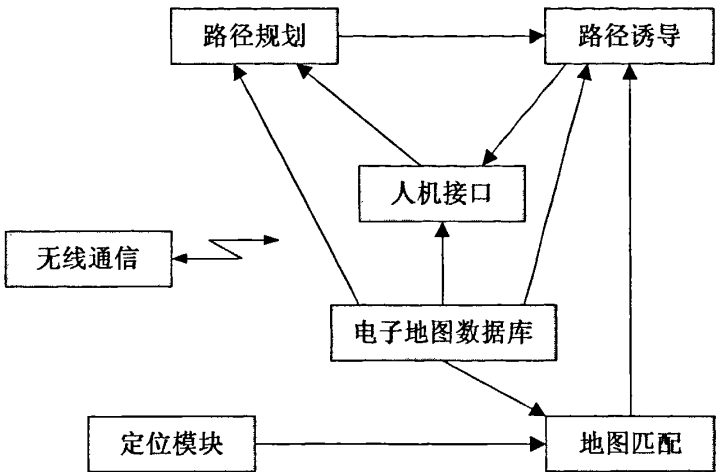


图1-2 车辆定位导航系统的基本模块

Fig 1-2 Navigation System Models

1.2.2 导航软件的基本功能

一般的导航软件的功能基本是：

- 1. 功能完善的电子地图显示浏览功能，完全可以代替传统地图。
- 2. 强大的查询功能，使旅游者能查询到景点、路线、宾馆、学校等所有地理位置信息。
- 3. 最优路径规划功能，可以进行道路距离的计算，规划出最优路径，使旅游者实现最短路径和最短时间的旅游。

在导航软件中，电子地图是一个不可缺少的组成部分，电子地图是随着电子制图系统的出现而发展的一种新的地图表现形式。它强调的是数据载体、符号化与显示^[5]。与传统地图集相比，电子地图具有许多新的特征：

- 1. 强调数据和图形的多媒体集成，强调将图形的直观性和声音的引导性相结合。
- 2. 具有查询检索和分析决策的功能，能够从地图图形到属性数据和从属性数据到地图图形的双向检索。
- 3. 图形动态变化功能，支持图形缩放，浏览阅读等基本功能到地图动画功能

多维动画图形模拟。

4. 易于用户理解, 操作。

路径规划算法是影响导航软件性能的另一个重要因素, 它要解决的主要问题是在给定的数字道路地图中寻找从出发地到目的地的最优路径, 如最短行车距离、最少旅行时间、最低交通费用等。而距离、时间、收费等信息都可以存在数字道路地图的路段属性中^[6]。根据路网数据模型, 可以将数字道路地图转化为带权有向图, 因此无论采用何种标准, 计算道路网络中两点之间最优路径问题都可以归纳为求解为带权有向图的最短路径问题。在图论中有许多比较成熟的最短路径算法可供选择, 如经典最短路径算法Dijkstra、A*算法等。算法解得的最优路径解可能并非理论上的最优, 而只是比较满意的次优或者较优路径。但如果两者之间的目标值相差不大, 而又能在运算速度和存储开销方面获得较大的改进, 那么该算法无疑也是非常合适车辆导航要求的。

目前国内外已经存在的路径规划系统基本上可以分为两大类:

1) 静态路径规划系统

以几何距离、道路质量为路阻计算的最短路径和静态路径都属于静态型最优路径, 目前车辆导航系统动态交通分配研究中多采用该类路径, 计算结果与真实最短路径存在较大差异。静态路径规划系统用于路径规划的路况信息数据库固定不变。主要产品有: 北京灵图科技的“天行者之纵横四海”, 它具有地图显示(缩放, 漫游)、GPS信息采集与滤波、车辆定位及路段匹配、地名查询(名称查询, 分类查询, 坐标查询)、最优路线设计、路径引导(文字引导及语音引导)、航迹记录与管理等功能, 还可以利用MID提供的无线通讯功能, 向服务中心申请兴趣点查询、路况信息服务、最佳路径导引等服务; 苏州宇达电通的“易上路”等。

2) 动态路径规划系统

实时动态最优路径是真实的最短路径, 并且若想提高导航信息的准确度, 最优路径的计算都应动态行程时间为依据, 可见, 车辆导航系统应以动态行程时间作为计算最优路径的基础, 并提供实时动态最优路径作为必备服务。动态路径规划系统用于导航的路况信息数据库按照交通流实时刷新, 这需要交通信息中心将交通信息, 如堵车情况, 交通拥挤情况通过无线通讯链路传到系统, 系统收到该信息后, 对原始信息数据库进行刷新, 从而可以选取最优路径, 并能动态调整选取新的路径。此类产品一般均处于研究阶段, 市场上还没有正式发售的。

1.3 电子地图介绍

随着计算机技术的发展, 为了方便地图的查询、制作、更新、复制和缩放的

需求,由计算机支持的电子地图便应运而生。电子地图是以数字形式表示的,具有在屏幕上动态显示、编辑、查询、检索、分析和决策等功能新型地图产品^[7]。由于科技含量高以及使用便捷,它的使用范围及发挥的作用已经触及到国计民生的许多方面,前景十分广阔。以电子地图为基础的智能交通系统,借助实时交通信息、通讯网络、定位系统和智能化分析与选线系统,可以缓解道路阻塞和减少交通事故,提高驾车者的方便性和舒适性。

在ITS中有很大一部分是和车辆定位及导航系统相关的,并且这部分内容是ITS最重要的一部分。只有通过显示屏幕中加载电子地图,用户才能和定位及导航系统进行交互,才能从定位及导航系统中获得自己所需要的信息;与此同时,用户只有通过对电子地图的操作,才能对ITS系统表达自己需要达到的目的,才能向ITS系统突出服务请求。一般把用于ITS中的电子地图成为导航电子地图。在车辆通过GPS定位的过程中,电子地图是各个系统的工作平台,同时也是工作对象。正是通过电子地图中量化的各种数据与GPS数据间的联系(地图匹配),这样才能确定车辆在实际空间中的位置;在车辆的导航系统中,电子地图也是进行最优路径规划的基础^[8]。

导航电子地图主要用来对车辆进行导航,其主要功能是:

- 1) 系统工作在实时环境下,具有实时性。
- 2) 软件运行速度快,空间处理。和分析操作时间短
- 3) 能够显示当前车辆的位置。
- 4) 可以按照用户输入的起点、终点位置迅速计算最优路径。
- 5) 具有适用于导航系统的多种功能,如:漫游、缩放等。
- 6) 具有凉或方便的查询功能

只有建立合理的电子地图数据引擎,在数据引擎的基础上来发出完备的导航电子地图,并且对导航电子地图进行合理的运用,ITS才能在实际应用中起到更大的作用。

1.4 本文工作的主要内容

本文主要完成的任务是在VB的开发环境下,基于MapX控件编写一款导航软件。该软件对电子地图的操作包括:放大、缩小、漫游、精确查询、地图图层控制、鹰眼、路径导航等功能。路径导航的算法采用的是经典的Dijkstra算法,同时对其它的典型算法进行了介绍和分析。利用MapInfo Professional V8.5手动制作一张电子地图,该地图包括:地图事物节点、道路网、铁路、河流等图层,并且地图对象具有各自的属性数据。

本文工作的主要目的是实现图1-2中人机交互、路径规划、电子地图数据库这三个模块的功能。也就是说，在PC机上实现静态路径优化导航的功能，并且对电子地图的制作进行研究，这样在以后通过移植到嵌入式的车载设备中，与GPS、GPRS等通信模块相结合，利用地图匹配和动态最优路径规划算法等技术，可以完成实时的车辆导航的功能。

电子地图导航系统的总体设计思想为：根据上面描述的系统应具备的主要功能，将电子地图导航系统划分为地图功能操作模块、地图信息查询模块、最优路径规划模块三个相互关联的模块，每个模块完成相应的功能。如图1-3所示：

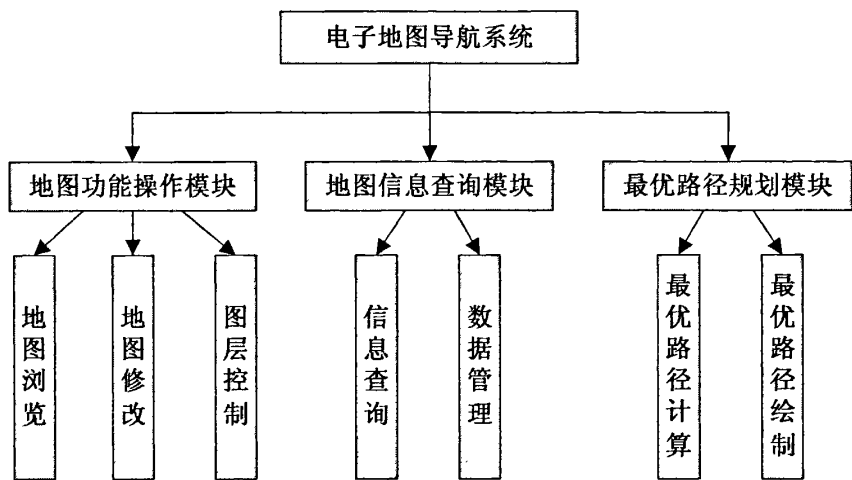


图1-3 电子地图导航系统总体结构

Fig 1-3 Digital Map Navigation System Framework

1.5 本文的组织结构和安排

本文第二章主要是介绍了电子地图的特点，讲述了MapInfo格式电子地图的数据组织结构和电子地图的绘制过程，同时也介绍了MapX控件的功能以及它的使用方法。

本文第三章针对最短路径规划算法进行了研究和分析，对电子地图中的节点一路段数据模型进行了相关的介绍。

本文第四章主要是对软件的实现过程进行了详细的讲解，包括软件的地图浏览功能、电子地图的图层控制功能、图元事务的查询功能、路径规划功能。

本文第五章是本课题所做工作的总结以及将来可能进行的工作的描述。

2 电子地图的设计与制作

电子地图是目前较为普遍的称呼，它又称为“屏幕地图”、“无纸地图”等。它是一种新型的地形信息产品。本文认为它可以呈现为二维矢量图形，也可以是栅格图形，或者用先进的计算机图形技术生成的多维地图。

2.1 地理信息系统概述

现实生活的应用中，与电子地图联系最密切的就是地理信息系统(GIS)。地理信息系统就是一种利用计算机对有关地理、空间位置的数据信息进行存储、处理、查询和显示的计算机支持系统^[9]。GIS的发展始于60年代，是计算机同步发展的结果。今天的地理信息系统集成了计算机数据库技术和计算机图形辅助设计软件。在图象处理上比上述两类软件更加全面，即地理信息系统所处理的事物对象具有空间地理特征，也具有统计信息特征。可以说，地理信息系统将空间信息数字化，并使这些信息可视化，通过功能强大的软件，使空间分析直观简明，数据管理便捷高效。

地理信息系统的核心是空间数据管理子系统，它由空间数据处理和空间数据分析构成。空间数据的主要来源有专题地图——等水位线、地形地质图等、遥感图像数据、统计数据及实测数据等。

地理信息系统具有以下七大功能：

- 1) 数据的提取、转换和编辑
- 2) 数据的集成
- 3) 数据重构和数据转换
- 4) 空间数据的查询和检索
- 5) 空间操作和分析
- 6) 空间显示和成果输出以及空间数据的更新
- 7) 空间数据的更新

GIS采用的基本技术可以归纳为地图分层、矢量抽象、空间数据与属性数据的划分等三个方面，其中，空间数据与属性数据划分技术使软件开发的重点放在对空间信息的管理上，借用已很成熟的数据库技术来管理传统的属性数据，实现多种信息的综合^[10]。

近年来地理信息系统技术发展迅速，主要动力是来自日益广泛的应用领域的对地理信息系统不断提高的要求。另一方面，计算机科学的飞速发展对地理信息

系统提供了先进的工具和手段。许多计算机领域的新技术，如面向对象技术、三维技术、图像处理技术和人工智能技术都可以直接应用到地理信息系统中。总而言之，由于GIS在人民生活和国防军事中的特殊作用，它将保持高速的发展势头，成为高科技领域的核心技术。

对于电子地图与GIS的关系，本文认为，电子地图是GIS的一种类型，它面向普通用户的日常应用需求。GIS与电子地图的首要区别是，GIS比电子地图更强调地图空间数据的完整性，强调其独立的地理意义，强调其空间与其属性的联系，地图表达是电子地图应用的目的。正因为电子地图与地理信息系统有着如此密不可分的关系，本文后面电子地图的制作所采用的辅助软件是MapInfo Professional Version 8.5，制作的电子地图具有利息信息系统的相关功能。

2.2 电子地图的分类和特点

地图数据库所存储和处理的信息分成两大类，电子地图的空间数据库的存储有两种方式：一种是栅格空间数据模型，另一种是矢量空间数据模型^[11]。

不同的存储结构决定了电子地图的生成方式和读取方式，按照空间数据的存储结构分，电子地图分为两种：一种是位图形式，另一种是矢量形式。

以位图文件的方式存储的电子地图称为位图电子地图，它是通过对地图册进行扫描后直接生成的。不仅与地图册的图形有一一对应的关系，而且可以保留原地图册上的丰富色彩。这种电子地图读取、显示程序简单，缩放自如，但是存储空间需求很大，另外读取显示时间比较长。这种电子地图的缺点在于缺乏良好的地图修正和删补功能，它只能保证与地图册的对应精度，却不具有与地图上的标准经纬度点的对应关系。这样会造成通过局部位图文件替换需要修正的部位来完成。但这样容易造成电子地图拼接上的误差，而且工作量也随之增大。

矢量电子地图的生成一般是通过数字化仪器将地图册上有用的信息以点和线的方式输入到计算机，同时加入经纬度点的标记。目前有很多矢量电子地图生成工具包来更好的完成矢量电子地图的生成工作。矢量电子地图的数据主要是相对于某点的偏移量数据。因此，他的数据占用空间小，并且与地图册上的点有经纬度的对应关系、易于修正和删补，同时能在地图上准确的显示图标。

2.2.1 电子地图的两种典型拓扑模型

拓扑数据结构是GIS网络分析所必需的，它描述了空间目标点、线、面之间的关联、邻接和包含关系。当前主流的GIS平台有美国ESRI公司的ArcInfo和MIS公司

的MapInfo。ArcInfo是大型GIS平台，以网络分析能力等见长。MapInfo是桌面型GIS平台，它的优点是图形化能力强，专题地图制作多样化，空间查询方便等。ArcInfo和MapInfo分别通过POLYVRT模型和“空间实体+空间索引”模型来定义空间数据的拓扑关系^[12]。

1. POLYVRT模型

POLYVRT模型由美国计算机图形与空间分析实验室(Laboratory for Computer Graphics and Spatial Analysis)研制。实现这类空间数据模型的系统主要是以ArcInfo为代表的一些大型GIS平台。

POLYVRT是一种以弧段(ARC)为基础的拓扑数据结构。弧段由任意多个点组成，在两端有节点，并伴有共享该段的左右两多边形的编码。多边形是由环绕其边界的弧段记录组成的。这种结构不仅存储了空间对象的集合信息，而且还存储了空间对象之间的拓扑关系^{[13][14]}。

这种结构的特点是，除节点外，每个空间对象都是由更基本的对象组成。只有节点的坐标是被实际存储的，其它复杂对象的坐标信息实际上是逻辑构成的，任一复杂对象能分解为一组节点及其拓扑关系的定义。拓扑关系显式地存储在特征表中^[15]。

2. “空间实体+空间索引”模型

20世纪80年代中后期出现的商用地理信息系统，尤其是桌面型GIS平台大多采用这种方式，其中以MapInfo为代表^[16]。

“空间实体+空间索引”模型的基础是空间实体。空间实体是地理实体的抽象，主要包括点、线、面三种基本类型。每个空间实体对象都是自包含的，也就是说，每个对象都维护着自己的所有属性。任一个空间实体，都是一个或多个部分组成的。部分是由“点集”组成的，“点集”是若干节点的集合。这样，在一个实体对象内部，记录了其全部空间信息。一个图层由多个空间实体组成，而每个空间实体都是自包含的，因此没有必要像POLYVRT模型那样通过特征表来建立对象到节点的引用关系，而是采用了空间索引^[17]。在MapInfo中空间索引的建立采用了R-Tree技术，将各个空间实体的最小外接矩形(MBR)存储在索引中，并按从小到大的范围进行索引搜索^[18]。

空间索引实际上实现了一种动态的拓扑关系。只有在需要时，系统才根据空间索引建立并使用实体间的拓扑关系，可以说是一种隐式的拓扑关系。

2.2.2 两种拓扑模型比较

POLYVRT模型中，复杂空间对象是由节点的引用构成的。其优点是多个不同

的对象可以共用相同的节点，从而节省了存储空间。由此而产生的问题是其结构更加复杂，数据的编辑和维护比较困难。

在空间实体模型中，复杂空间对象的节点是存储在实体对象内部的，因此会造成公共节点的重复存储。但是，更加结构化的实体模型使得对某个对象的更改不会影响到其它对象的定义，从而大大增强了空间数据的可维护性。

2.3 MapInfo 系列的电子地图

2.3.1 MapInfo Professional 软件的介绍

MapInfo公司于1986年成立于美国的特洛伊市，该公司一直致力于提供先进的数据可视化、信息地图化技术，其软件代表市桌面地图信息系统软件MapInfo^[19]。

MapInfo Professional软件是MapInfo公司主要的软件产品。软件如图2-1所示：

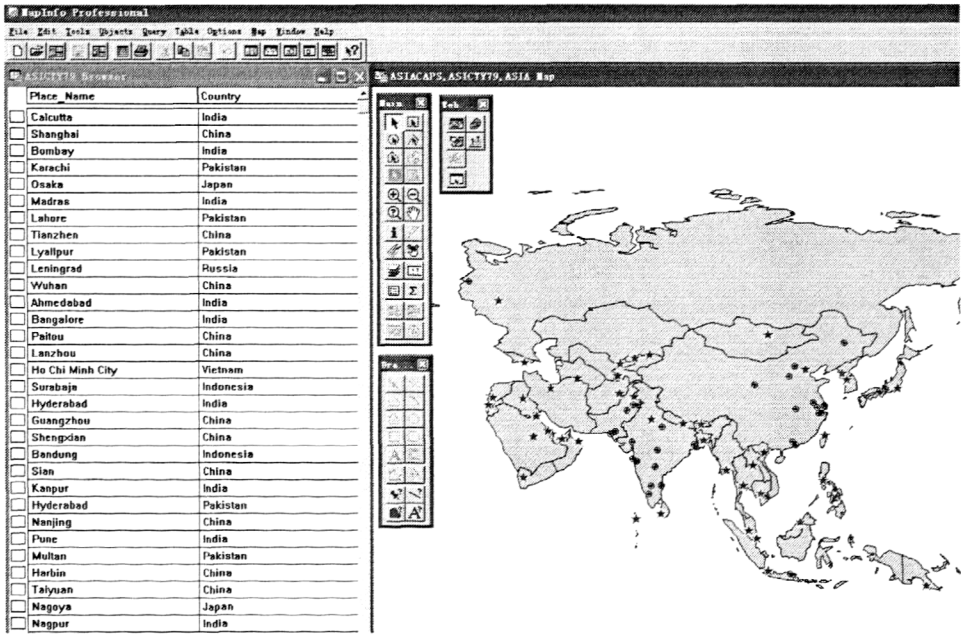


图2-1 MapInfo Professional软件界面

Fig 2-1 MapInfo Professional Software

MapInfo Professional软件不仅实现了电子地图的显示、管理、建立和修改功能，更重要的是实现了在电子地图上的地图对象与关系数据库中的记录的自动联接。通过MapInfo。可实现地图与数据库的双向查询，并能使地图上的对象(如建筑物、道路等)与数据库中的相关数据联接，以数据库中的数据来动态改变地图对象的可视属性，自动生成生动易用的专题地图(线路流量负载图、各分支机构营业收入图、机线设备利用率统计图等)，以供管理、分析、决策。对数据库的查询结果，可以

直接反映在地图上,也可直接在地图上选择对象,以查询相应数据库信息,可为数据库查询结果自动地建立一张结果地图或为地图上的选择结果自动建立数据表,这就为系统的管理、使用提供了极大的方便,尤其适合管理人员进行宏观查询,综合分析。

2.3.2 MapInfo 电子地图的数据组织结构

GIS中的矢量电子地图是按图层组织的,即将一幅地图分成多个层层叠加的透明层,这些透明层就称为图层。每个图层存放一类专题或一类信息,它由点、线、面等空间对象的集合组成。

使用MapX需要包含来自MapInfo的记录和地图的文件。MapX将其所有基础信息以MapInfo表的形式组织起来;每一表都是一组MapInfo文件,用来在地图中建立一个图层。

所有 MapInfo 表都将具有以下文件:

1. **tab文件:** 该文件描述MapInfo表的结构。它是描述包含数据的文件的格式的小文本文件。属性数据表结构文件定义了地图属性数据的表结构,包括字段(Fields)数、字段名称、字段类型和字段宽度、索引字段和相应图层的一些关键空间信息描述。tab文件实际上是一个ASCII文件。
2. **dat (.mdb、.aid 或 .dbf)文件:** 这些文件包含表格数据。属性数据文件中存放完整的地图属性数据。在文件头之后,为表结构描述,其后首尾相接地紧跟着各条具体的属性数据记录。
3. **map文件:** 该文件描述图形对象。空间数据文件具体包含了各地图对象的空间数据。空间数据包括空间对象的几何类型、坐标信息和颜色信息等。另外还描述了与该空间对象对应的属性数据记录在属性数据文件(.dat)中的记录号,这样当用户从地图上查询某一地图对象时,就能够方便地查到之相关的属性信息。
4. **id文件:** 该文件是将数据与对象相链接的交叉引用文件。交叉索引文件记录了地图中每一个空间对象在空间数据文件(.MAP)中的位置指针。每四个字节构成一个指针。指针排列的顺序与属性数据文件(.dat)中属性数据记录存放的顺序一致。交叉索引文件实际上是一个空间对象的定位表。
5. **ind文件:** 它是索引文件。通过该索引文件,您可以使用Find对象搜索地图对象。索引文件并不是必须的,只有当用户规定了数据库的索引字段后,MapInfo系统才会自动产生索引文件。索引文件中对应于每个索引字段都有一个索引表。在每个索引表中,先给出总的数据库记录数目,然后按照

索引顺序给出每条属性数据记录在对应的索引字段处的具体属性数据和该记录在属性数据文件(.dat)及交叉索引文件(.id)中的记录号。

2.3.3 MapInfo 电子地图的绘制

电子地图的传统制作方法是使用专业电子地图制作软件，本文中使用MapInfo软件，以“表”、“记录”的形式保存地图图元对象的属性数据，然后在应用程序中调用数据库的地图图形数据记录，动态绘制生成放大、缩小的图形和相关信息数据。实现了电子地图的基本功能，如漫游、放大以及缩小等功能。充分利用GSI工具软件对空间数据库的管理、分析功能，大大提高应用系统的开发效率，而且可靠性较高。

电子地图不同于普通的纸质地图的一个主要方面是：纸质地图是将地理的所有要素都绘制在一张纸上，而地理信息系统按照图层来组织地图。这就是说，将一幅计算机图层加工成多个层叠加的透明层。每个图层包含了地图中的不同类型的要素^[20]。如图2-2所示：

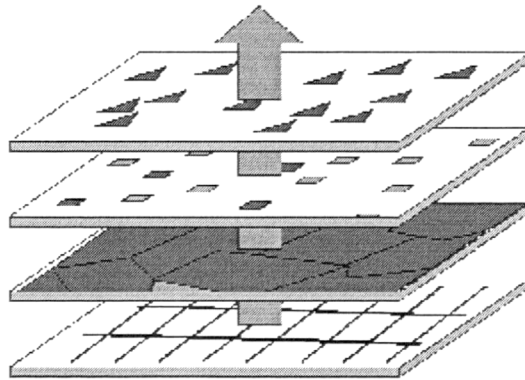


图2-2 图层示意图

Fig 2-2 Layers of Digital Map

我们可以将这些图层视为透明的部分，每个图层包含地图的不同组成部分。这些图层依次堆叠在一起，用户即可看到地图的全貌。

地图对象是图层的基本元素，大致可以分为以下四种^[21]：

- 1) 点对象，点是由平面位置决定的一个空间地理实体，由一对平面坐标表示。例如车站、超市、医院等，也可以用来表示地理线的起点、终点和交点。点对象的建立可由符号工具在相应坐标点标出，也可以由可地图化的MapInfo表中的地理坐标用“表>创建点”自动绘出。
- 2) 线对象，线是具有相同属性点的轨迹，在图层中由坐标集合的数据链表示，

具有一定的走向和长度，表示线状地物或点之间的地理联系。建立线对象的方法是：以配准过的栅格图像为基图，利用数字化仪和人工手动鼠标跟踪，选择相应的线型和颜色，便可描绘。

- 3) 区域对象，区域对象是具有相同属性点的轨迹，具有确定的范围和状态，表示空间连续分布的地理景观或作用范围。创建区域对象用区域对象工具，由数字化仪和人工手动鼠标跟踪的方法绘制。
- 4) 文本对象，文本对象用来描述地图的有关信息，或者以文本格式描述其他对象的属性。

图层的基本元素的分类如图2-3所示：

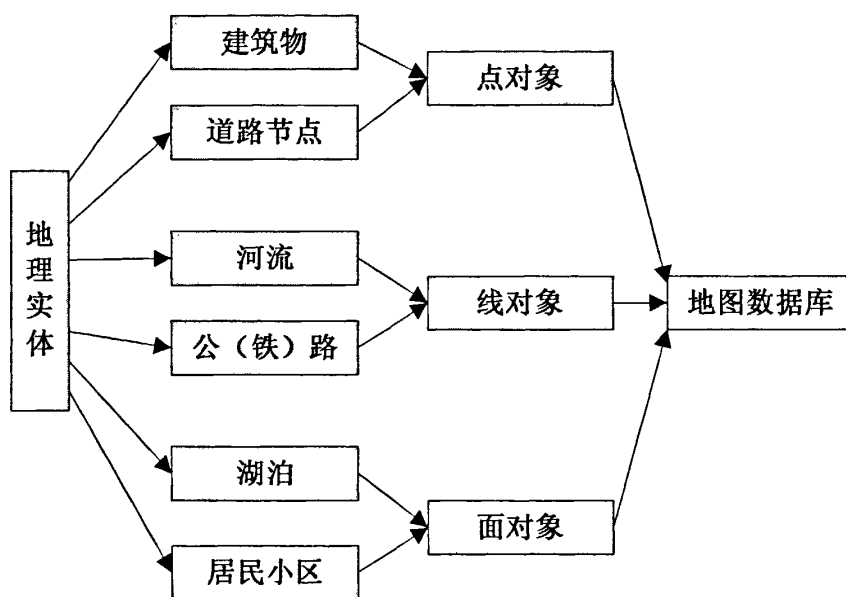


图2-3 地图对象的分类

Fig 2-3 Map Feature Sort

一般MapInfo格式电子地图的绘制分为二个关键步骤：

1. 从数据表或者拓扑关系图创建MapInfo .tab文件。
2. 进行地理编码或为相应数据创建点。

下面将会对这两个步骤进行详细的研究，本文所使用的软件是MapInfo Professional Version 8.5 版本。

1. 创建tab文件

在MapInfo格式的电子地图中的任何一个图层中的每一个元素都可能和其他的元素产生关系，而且它本身也根据地图的要求具有一些属性和数据。这些属性和数据都存储在tab文件中。因此在MapInfo格式的电子地图中使用数据的第一步是

从数据表创建tab文件。这一过程取用你所创建的数据(电子表格或数据库)并将信息提取到一个MapInfo软件可以使用的tab文件。在上述处理过程中，数据仍然保持不变。MapInfo Professional使用的处理过程取决于数据格式类型，你可以使用下面这些格式的数据文件通过MapInfo Professional软件来转换成需要的tab文件。

1. Microsoft Excel (.XLS)
2. Microsoft Access (.MDB)
3. d-Base (.DBF)
4. ASCII (.TXT)
5. RDBMS (Oracle、Informix、SQL Server)
6. XML (Web 地图服务)

本文选择的是将Microsoft Excel格式的数据文件转化为tab格式的数据存储文件。如图2-4：

RoadData Browser					
	RoadNum	RoadName	FNode	TNode	Length
<input type="checkbox"/>	19		6	25	65
<input type="checkbox"/>	20		25	6	65
<input type="checkbox"/>	21		24	25	30
<input type="checkbox"/>	22		25	24	30
<input type="checkbox"/>	23		4	16	170
<input type="checkbox"/>	24		16	4	170
<input type="checkbox"/>	25		16	18	150
<input type="checkbox"/>	26		18	16	150

图2-4 道路的数据文件

Fig 2-4 File of Road Data

这是一张电子地图道路信息的关系数据图，其中包含了：道路编号、道路名称、起始节点、目的节点和道路长度这些简单的道路信息。最重要的是这些道路的tab文件中包含了抽象的路由表，也就是每条路径的和其他路径的关系，这是后面最优路径搜索的必要条件。

除了道路元素的tab文件，节点的tab文件也是必不可少的，每一个节点代表着地图上的一个地图元素，后面简称图元。

图2-5只是一个简单的节点tab文件，只是确定了节点所代表的图元名称，这样是地图具有了查询图元的功能。而且根据节点所代表的图元类型不同，可以分成若干个图层。



	NodeNum	NodeName
<input type="checkbox"/>	1	华联商场
<input type="checkbox"/>	2	中国移动
<input type="checkbox"/>	3	德国大众
<input type="checkbox"/>	4	摩托罗拉
<input type="checkbox"/>	5	中国联通
<input type="checkbox"/>	6	中国石油
<input type="checkbox"/>	7	人民医院
<input type="checkbox"/>	8	妇幼医院
<input type="checkbox"/>	9	希尔顿饭店
<input type="checkbox"/>	10	外贸宾馆
<input type="checkbox"/>	11	广州酒店
<input type="checkbox"/>	12	浦江大饭店
<input type="checkbox"/>	13	友谊宾馆
<input type="checkbox"/>	14	国贸饭店

图2-5 节点的数据文件

Fig 2-5 File of Vertex Data

2. 进行地理编码

要在地图上显示数据，必须先为每个记录指定 X 和 Y 坐标。MapInfo指定这些坐标的一种方式：将数据库表中的地理信息和另一个已经具有X 和 Y 坐标的表(称为搜索表)中的地理信息匹配。

本文的电子地图采用手动的地理编码方式，结果如图2-6：

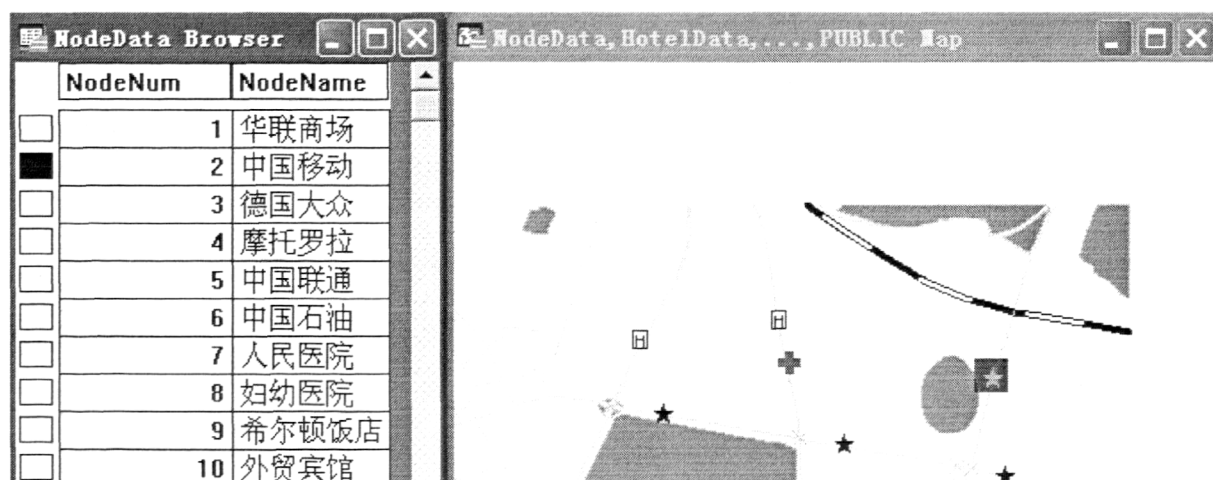


图2-6 地理编码示意图

Fig 2-6 Geography Code

MapInfo Professional软件的地理编码有两种方式：自动和手动。其中自动地理编码方式适合于地图信息比较庞大的工作，先利用自动地理编码方式完成初步的地理编码，然后再通过手动地理编码方式进行修改，最终完成整个电子地图的地理编码。手动地理编码方式对于具有“清洁”数据的数据库均可以正常工作。但有时，可能知道点在地图上的位置，但是该位置数据无法进行匹配。手动地理编码是通过单击地图位置，将数据记录置于地图之上的过程。这一方法尤其适用于表较小并且对其中的数据比较熟悉的情况。

图2-6种左侧的节点编号为2的图元代表的是“中国移动”,它在电子地图上对应于右侧被选中的图标。这样将电子地图上的所有图元都与显示中的地理信息相结合。

2.3.4 电子地图的实现

这里简单扼要的讲述一下，矢量地图的分层原则：基础图层是指组成地图的基本要素，这些要素是构成电子地图的基本内容，如道路、河流、街区等。对其分层应该根据需要进行，如道路部门的 GIS 电子地图必须将所有与道路有关的进行矢量化，以满足分析、查询、统计的需要^[22]。

本课题制作的电子地图的道路图层和节点图层，如 2-7、2-8 所示：

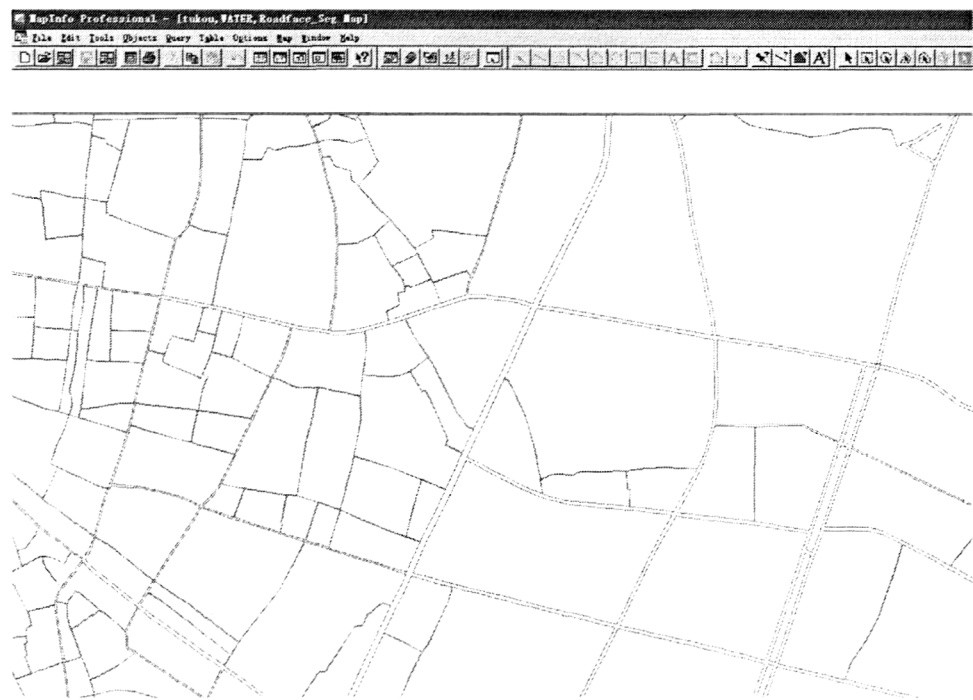


图 2-7 道路图层

Fig 2-7 Layer of Road

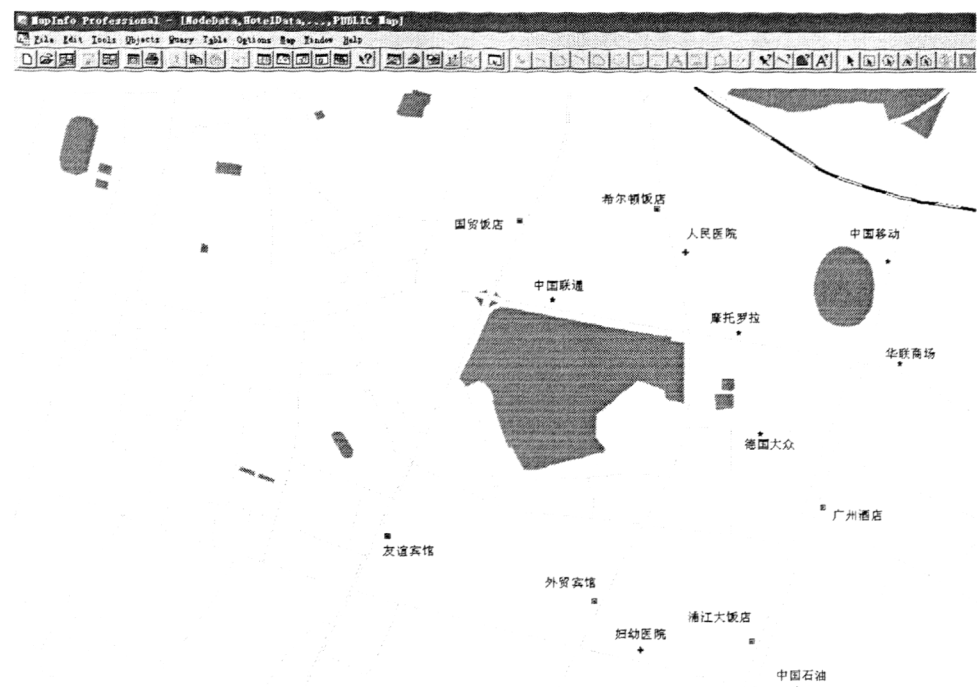


图 2-8 节点图层

Fig 2-8 Layer of Vertex

将节点图层、道路图层以及一些其他辅助图层按照一定的图层顺序排列就会得到图2-9的电子地图。

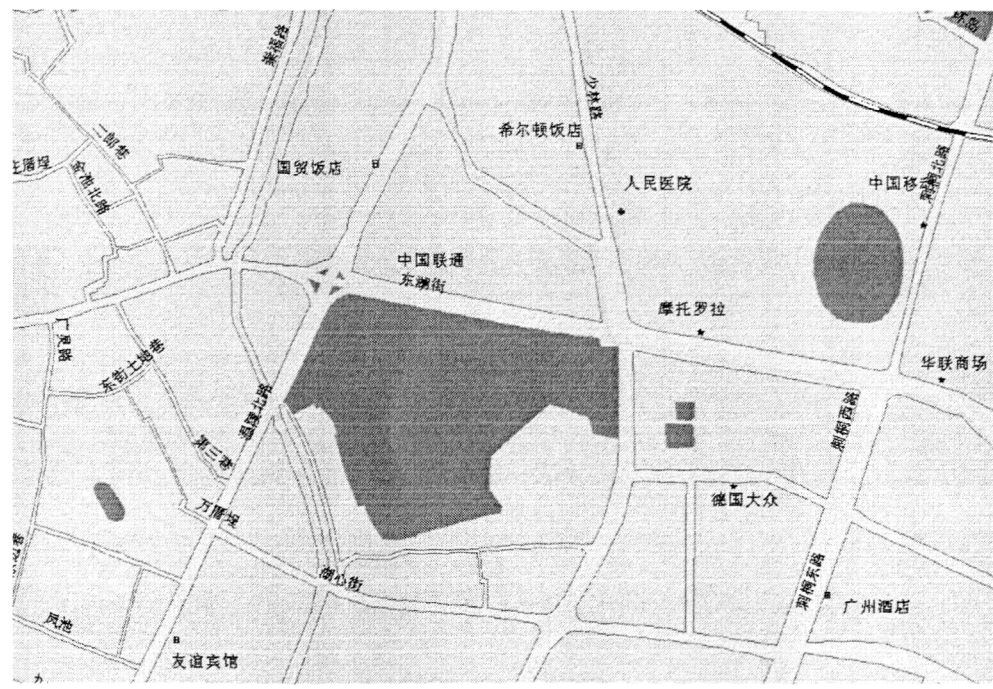


图 2-9 电子地图

Fig 2-9 Digital Map

2.4 MapX 控件介绍及其开发功能

MapX是MapInfo公司向用户提供的具有强大地图分析功能的OCX控件产品。利用MapX,能够简单快速地在企业应用中嵌入地图化功能,增强企业应用的空间分析能力,实现企业应用的增值。MapX采用基于MapInfo Professional的相同的地图化技术,可以实现MapInfo Professional具有的绝大部分地图编辑和空间分析功能,而且MapX提供了各种工具、属性和方法,这正是本文采用MapX控件作为开发工具的原因。

2.4.1 MapX 控件介绍

MapX是MapInfo提供的OCX控件,是MapInfo二次开发的一个强大的工具。在使用面向对象语言开发的应用项目中嵌入MapX可以使应用项目具有强大的地图功能。MapX基于与其它MapInfo产品所使用的相同的地图绘制技术。

MapX的优点主要有三点:

- 1) 增强信息可视化能力以及数据的深层表现力。
- 2) MapX强大的内嵌功能,由于采用的是控件技术,因此MapX可以无缝的嵌入到各个新的或者旧的应用系统中去。
- 3) 易于掌握,开发周期短,成本低,使用者只需一门通用计算机语言如VB、VC、Delphi、PowerBuilder就可以使用MapX进行二次开发,从而缩短应用开发周期,降低开发成本。

本课题中主要介绍的工具是MapInfo Professional软件和MapX控件,他们都能对MapInfo电子地图进行编辑。两者之间既有区别又有联系,它们的区别是:使用MapX开发,编辑人员不用掌握其他专业编程语言,就可以将地图应用功能嵌入到应用中,并且可以脱离MapInfo的软件平台独立运行。相反如果选用MapInfo Professional作为地图化应用的平台,则需要掌握很多技巧和知识来完成。例如MapBasic是一种类Basic语言的地图应用开发语言,它有自己的语法规则,同时它编译生成的执行程序离不开MapInfo Professional平台。而有编程经验的人则喜欢用MapX,因为他们可以使用自己熟悉的开发语言来进行工作,充分发挥他们自身的优势。它们的联系是:首先,MapX没有创建MapInfo地图的功能,所以除非用户已经拥有MapInfo电子地图,否则,使用者必须利用MapInfo Professional来制作电子地图。用户在MapInfo Professional中创建的电子地图,在MapX中都可以直接调用。其次,MapX可以对调入的MapInfo电子地图进行重新的编辑(增删点、线、面对象),修改对象的属性信息。

作为一个控件，先介绍它的模型结构：MapX组件的基本组成单元是Object（单个对象）和Collection（集合）。其中集合包括对象，是多个对象的组合。每种对象和集合负责处理地图某一方面的功能。

由图2-10可以知道，位于顶层的是Map对象本身，其它均由Map对象继承。Layers, DataSets, Annotations是Map对象下面的三个重要的分支。其中Layer主要用于操作地图的图层，DataSet用于访问空间数据表，Annotation用于在地图上增加文本或者符号。

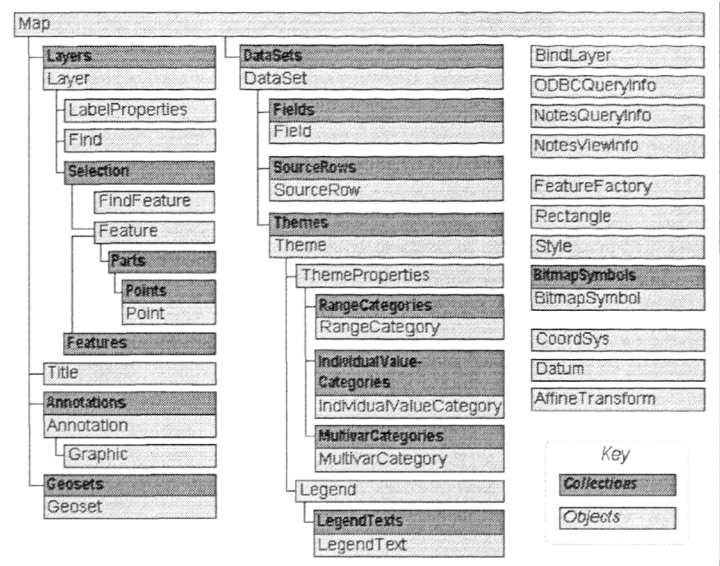


图2-10 MapX控件中的模型结构图

Fig 2-10 MapX Model Graph

下面本文将会对课题涉及的对象进行详细的介绍。

2.4.2 MapX 控件中的图层(Layer)

前文已经提到电子地图是由若干个图层叠加来实现的，因此图层对象，即Layer。Layer集合是地图控件的属性之一，包含Layer对象。这些Layer对象构建自MapInfo表，一起构成了地图。每个图层均包含不同的地图图元，例如区域、点、线或文本。Layer集合具有用于执行操作的属性和方法，例如从集合添加或移除Layer对象等操作。与Layer对象有关的属性和方法，可以参考MapX开发手册。

Layer对象代表采用地图图元形式的矢量制图数据，具有主要的图元类型，例如区域、直线或符号。通常Layer对象相应于源自MapInfo表的地理对象。Layer集合中的每个Layer对象的行为相互独立。可以在程序之内随时控制图层对象。也可以通过适当的方法从图层中获取图元。

Layer集合中的地图图层按照增序索引显示(例如Layers(1)是顶层，Layers(2)是

Layer(1)之下的图层), 底层最先绘制, 顶层最后绘制。正确设定图层的顺序非常重要。

图层的顺序在使用选择工具时也很重要。选择工具从最顶部的可选择图层中选择对象。如果在相同的位置上有若干个对象, 则难于选择所需的确切对象。还可以重新给图层排序, 以便要选择的图层是新置于最顶层的图层。

2.4.3 MapX 控件中的数据绑定对象(DataSet)和域对象(Field)

Dataset对象是将数据源中的数据绑定到MapX上的结果。数据绑定是将数据源中的数据导入MapX的过程。当今的商业中有许多不同类型的数据库; 因此, MapX使您可以绑定到多种不同类型的数据源。在MapX中, 数据显示为Dataset对象。

将数据绑定到地图上主要有以下两个优点:

1. 可以将数据作为地图上的图元查看。
2. 可以将属性数据绑定到地图上, 然后使用该绑定创建基于数据的主题阴影地图。

使用Datasets.Add方法, 可以将数据源中的数据绑定到MapX上。这样可以将外部数据源中的数据绑定到地图上。

数据源可能包含许多列数据。MapX对绑定的每个数据列均需开销, 所以, 应只绑定地图中需要的数据。使用 Datasets.Add 的 Fields 参数建立要绑定到地图上的字段(列)的Fields集合。

使用Fields.Add方法可以将数据源中的数据列作为字段加入 Fields集合。

DataSet对象和Field对象主要的区别是: DataSet对象一般与一个数据文件对应, 而Field对象则是和数据文件中一列的某个项目对应。

如图2-4中, RoadData数据文件对应一个DataSet对象, 而RoadData数据文件中的FNode域对应于一个Field对象, 这个Field对象包含所有道路的起始节点信息。

2.4.4 MapX 控件中图元(Feature)

地图图元是地图上的点、线条或区域之类的地理对象。在MapX中, 组成地图的不同图层通常在每个图层中拥有相同的图元类型。地图图层中的所有图元或地图图层中的部分图元表示为一个Features集合。许多Layer对象方法返回图层中的Features集合。Features集合中的图元在地图中不会突出显示。要突出显示, 应使用Selection集合。Selection集合也是Feature对象的集合。不过, Selection集合代表当前选择的Feature对象。MapX的基本功能是选择地图上的图元, 可以对图元执行其

它任务。

MapX可以创建、修改或删除组成地图图层的图元(点、线条、区域等)。可以编辑任何基于普通MapInfo表(.tab)文件的图层。Feature对象方法可以创建和处理独立图元对象。在创建独立图元对象时,必须将该图元对象连接到地图上,然后才能引用其方法或属性。将图元连接到地图上就是将地图的坐标系与图元关联。

创建新地图图元有两种方法。可以通过分配新Feature对象来创建图元,也可以通过使用Map对象FeatureFactory属性的方法对现有图元执行操作来创建图元。这些类型的图元称为独立图元。独立图元受到一些限制:不能加入任何集合,只允许使用用于定义图元的方法和属性。例如,不能对独立图元使用Area属性,独立图元没有Feature.Layer属性的值。图层中的图元不是独立图元,所有属性和方法均可以使用。

2.4.5 MapX 控件中查图元对象(Find)

Find对象可以确定图元在地图中的位置。可以查找线条、符号或区域图元。要使用Find方法,要求正在搜索的图层必须具有索引字段。

Find.Search方法返回它通过FindFeature对象找到的图元。

FindFeature对象将找到的Feature对象的属性作为它自己的属性存储。此外,FindFeature具有存储Find操作的结果码的FindRC属性。FindFeature对象是Feature的超类,增加了要返回的最匹配的字符串。FindRC属性是指示找到或未找到图元的原因的数值。

Find对象在地图查询功能中有着十分重要的作用。

2.5 本章小结

本章的主要工作是对导航软件配套的MapInfo格式电子地图的设计和制作过程进行了详细的介绍。首先,按照空间数据存储方式划分电子地图分为两种:一种是栅格空间数据模型,另一种是矢量空间数据模型。由于电子地图和地理信息系统密不可分的关系,使得电子地图的拓扑模型显得格外的重。文中列举了两种最常见的拓扑模型结构,并且进行了比较分析,因此决定选择MapInfo格式的电子地图。其次,介绍了软件MapInfo Professional,对MapInfo格式电子地图的数据组织结构进行了描述,对电子地图绘制的概念和实现方法也有详细的讲解。最后,对MapX这款控件进行了剖析,对它的几个重要的操作对象也有相应的介绍,对它在后面实现的电子地图操作功能的原理作好了铺垫准备。

3 路径规划算法的研究

路径规划是帮助司机在行驶前或运行中规划路线的过程,一般情况采用的技术是找到最小行驶开销的路线,这些开销可以是时间、距离或经济费用等因素。对于路径算法来说,推测未来行驶开销的准确性和可靠性就非常关键了。就目前来说,在目前的道路网络中搜索代价最小的目标路径问题有很多种算法,其中最短路径具有很大的代表性。因此最短路径算法一直是运筹学、计算机科学、地理信息科学研究的一个热点。按照规划目的的不同,路径规划可分为多车辆路径规划和单车辆路径规划,前者多用于车队调度和交通管制,后者则广泛应用于各种导航系统。车辆定位导航系统的路径规划属于单车辆规划的范畴,它要解决的主要问题是给定道路网中寻找从出发点到目的地之间的最优路径。经典图论和不断发展完善的计算机数据结构及算法的结合使得新的算法不断的涌现,它们在各自相应的领域都有着出色的表现。

3.1 节点—路段数据模型

在 GIS 中,经常将空间实物抽象成点、线、面等几何要素。点、线建立拓扑关系,可以组成网络。网络图形在几何上是由边连成的,边的端点、交点是网络的节点。网络在数学和计算机领域中是被抽象为图这个概念,其基础是图的存储表示。下面有必要对图论及其相关的知识予以简单的介绍。

3.1.1 图论的基本概念

图论是研究与图相关的理论和算法的一门学科,它的应用非常广泛,已渗入到诸如运筹学、逻辑学、物理学、化学、计算科学、系统科学、语言学等领域^[23]。随着计算机科学技术的不断发展,经典图论这门科学在数据结构、网络优化等方面起到的作用正越来越明显。

图论中所研究的图并不是普通意义上的图形,它表示的是定义在顶点集上的二元关系,其本质是抽象的概念。由若干个不同顶点与连接其中某些顶点的边所组成的图形就称为图。通常用一个大写字母 G 来表示图,用 V 来表示所有顶点的集合。 E 表示所有边的集合, $G = \{V(G), E(G), \Psi_G\}$ 是图的组成集合,其中 $V(G) \neq \emptyset$ 是顶点集合, $V(G)$ 的元素是图 G 的顶点, $V(G)$ 中所含元素的个数即顶点数,称为图的阶,记为 $|V(G)|$ 或 n ; $E(G)$ 是边集合, $E(G)$ 的元素为边, $E(G)$ 中所含的元素

的个数称为边数或弧度，记为 m ； Ψ_G 是关联函数，是 $E(G)$ 到 $V(G)$ 集合的映射。一般将图简写为： $G = (V, E)$ 。如图3-1所示：

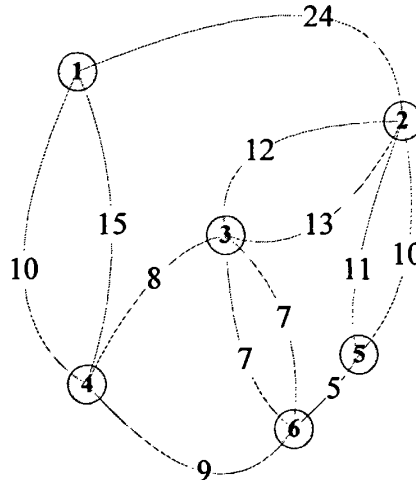


图3-1 抽象图

Fig 3-1 Abstract Graph

对于任意 $(u, v) \in E$ ，如果 (u, v) 是有序的，那么 G 是有向图， $\langle u, v \rangle$ 表示从顶点 u 到 v 的一条弧线，并且称 u 为弧尾或者起点， v 为弧头或者终点。如果 (u, v) 是无序的，即如果有 $\langle u, v \rangle \in E$ ，一定有 $\langle v, u \rangle \in E$ ，则用 (u, v) 无序对代替有序对，称为从顶点 u 到 v 的一条边，则 G 是无向图。

对于无向图 $G = (V, E)$ ，若有边 $(v', v) \in E$ ，那么称 v 和 v' 互为邻接点或 v 和 v' 相邻接，且边 (v, v') 依附于顶点 v 和 v' 或称 (v, v') 和顶点 v 和 v' 相关联。对于有向图 $G = (V, A)$ ，若有弧 $\langle v', v \rangle \in A$ ，则称顶点 v 邻接到顶点 v' ，顶点 v' 邻接自顶点 v ，并且弧 $\langle v, v' \rangle$ 与顶点 v 和 v' 相关联，它从顶点 v 出发，也称为顶点 v 的前向关联边，顶点 v' 的后向关联边。

在无向图中，一个顶点 v 的度是与它相关联的边的条数，记为 $TD(v)$ 。在有向图中，顶点的度是指与它先关联的弧的数目，等于该顶点的入度与出度之和，其中，顶点 v 的入度是以 v 为终点的弧的数目，记为 $ID(v)$ ；顶点 v 的出度是以 v 为起点的弧的数目，记为 $OD(v)$ ；顶点 v 的度 $TD(v) = ID(v) + OD(v)$ 。一般地，对于有 n 个顶点、 m 条边的图，有 $m = \frac{1}{2} \left\{ \sum_{i=1}^n TD(v_i) \right\}$ 。

图的边或弧可具有与之相关的量化信息，表示从一个顶点到另一个点点的距离、费用等开销。这种与图的边或弧相关的量化信息叫做边或弧的权值，这种边或者弧带有权值的图叫做带权图，通常也称为网络，图 3-1 就是一个带权图。其中的顶点称为节点(Node)。此外，图的节点也可能带有某种反映其特性的量化信息，

称为节点权值,这种带有节点权值的图称为点带权图。一般地,将弧 (u,v) 上带的权记为 $w(u,v)$,它也成为这条弧的长度。

在无向图 $G=(V,E)$ 中,路径是一个非空的顶点序列,从顶点 v 到顶点 v' 的一条路径是 $P=(v_{i0}=v, v_{i1}, v_{i2}, \dots, v_{ik}=v')$,其中 $(v_{ij-1}, v_{ij}) \in E, 1 \leq j \leq n$ 。如果是有向图,则路径也是有向的,定点序列应满足 $(v_{ij-1}, v_{ij}) \in E, 1 \leq j \leq n$ 。路径长度是指路径上边或者弧的数目,假如上述路径 P 的长度为 k 。序列中第一个顶点和最后一个顶点相同的路径称为回路或者环。序列中顶点不重复出现的路径成为简单路径。对于带权有向图,路径 P 的长度就是其所经过的弧的长度之和,即

$$L_P = \sum_{(u,v) \in P} w(u,v)。$$

在无向图中,如果从顶点 v 到顶点 v' 有路径存在,则 v 和 v' 是连通的。图中任意两个顶点都连通的无向图称为连通图。在有向图中,如果从顶点 v 到顶点 v' 有路径存在,则称 v' 相对于 v 是可达的;如果从 v 到 v' 和从 v' 到 v 都存在路径,则 v 和 v' 是连通的。图中每一对顶点都是连通的有向图称为强连通图。

设有向图 $G=(V,E)$ 中有一条路径 $P=(v_1, v_2, v_3 \dots, v_k)$,当 $1 \leq j < k$ 时,节点 v_{j+1} 称为 v_j 的后继,路径 P 中除最后一个节点 v_k 以外的每一个节点都有一个后继;当 $1 \leq j < k$ 时,节点 v_{j-1} 称为 v_j 的前驱,路径 P 中除第一个节点以外的每一个节点都有一个前驱。在图 G 中,节点 v_j 的所有后继节点组成的集合记为 $\Gamma(j)$,所有前驱节点组成的集合记为 $\Gamma^{-1}(j)$ 。

3.1.2 节点—路段数据模型的形式化定义

从路网的直观结构考虑,数字道路地图可以用图的形式来表示,把道路中的起点、终点和交叉路口等表示为节点,把道路表示为连接节点的弧,把道路的距离或者车辆通过该道路的平均行驶时间等量化信息表示为道路的权值,那么道路网就被抽象为带权有向图,而与之相关的问题就可以用图论的方法分析和解决。

在采用矢量编码方式的路网数字地图数据库中,道路以线或者折线对象的方式存储,表示为一系列坐标点的有序集^[24]。为了适应车辆导航系统需要,必须将包含交叉点的道路拆分成最基本的路段,使其只在端点处与其它路段相交。拆分后的基本路段信息对应于路网中的弧,其端点就是图中的顶点。可双向行驶的路段要用两条端点相同、方向相反的弧表示。目前的车辆导航系统应用中,很多都采用基于节点—路段的数字道路地图数据模型,用图的形式可以表示为:

$N_R = (V, E)$ 、 $V = \{v | v \in N_S\}$ 、 $E = \{R_S\}$ 、 $R_S = \{(u, v, Q^w) | u, v \in V\}$, 式中, N_R 代表道路网络; N_S 代表道路的节点集; R_S 为道路的有向路段集,对应图的弧集,可

以用道路网络中两个节点的拓扑关系表示； w 和 v 分别为起点和终点； Q^w 是路段的属性集，可表示长度、平均速度、通过时间、是否收费等。

1. 路段 路段一般作为道路的最小空间描述单元。一个路段指的是两个节点之间的道路，在路段的两端各有一个节点，路段之间是相互独立的，一个路段的道路情况发生变化不会影响其他的路段信息。事实上，对于每一条道路，它有自己的道路属性信息，像道路的交通限制、道路名称、道路宽度等信息，所以需要根据属性信息的变化把道路拆分成单元更小的路段。所以，一条道路由若干条路段组成，即路段是具有同一属性、在路网中相互独立且最小的道路图形特征。
2. 节点 节点是一个虚拟意义上的点，它是一个抽象的概念。路网中节点并不仅仅指交叉口或者道路的终点。在车辆导航系统中，节点还包括道路特性发生改变的点，如果道路的属性信息发生变化，比如通过该路段的开销、路段宽度等。只要道路的属性信息的变化会影响到路径规划的结果，就要通过节点把道路分割成不同的路段。

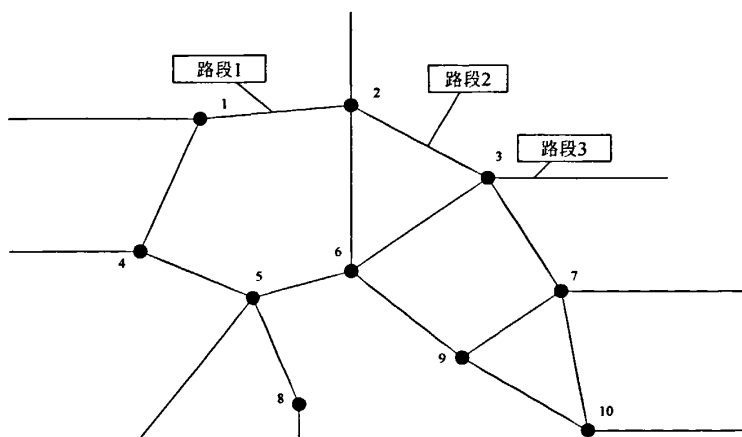


图 3-2 数字地图中的节点和道路表达

Fig 3-2 Vertex and Route in Digital Map

3.1.3 路段的方向性问题

在电子地图上，每条道路通常由若干条线段表示，道路的方向，道路的节点坐标以及道路的其他属性构成了道路的主要内容。对偶图法是运筹学中利用构造转换网络处理节点权重的一种方法，它的基本思想是将基于节点的图转化为基于

边的图。对偶网络的形成规则为：对偶网络的节点对应原网络中的边，并保持其特性；对偶网络的链代表转向行为；起点和终点在原网络和对偶网络中都是节点。对偶链的属性可能包含附加信息，如转向限制和交叉口延误。对偶网络中的节点和链与原网络中的弧和转向行为（直行、左转、右转、掉头）之间的对应关系，如图 3-3 所示：

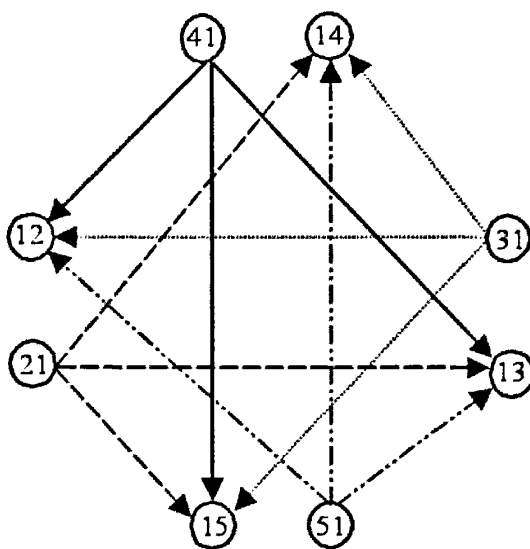


图 3-3 路网的对偶表示法示意图

Fig 3-3 A Figure of Antithetic Expression

对偶图法处理交叉口转向限制时，只需将禁止转向对应的链去掉即可。在对偶图中，对偶链的权重等于对偶链起点所代表的路段权重加上该对偶链所代表的转向行为的权重。这样可以将原网络中的节点权重转移到对偶链上，从而得到不带有节点权重的普通网络，便于一般的路线优化算法求解。然而，在实际的路网中，采用对偶图法构造拓扑结构同样具有明显的缺陷。

首先，理论上可以证明，对于方格形路网，其弧段数目和节点数目的比例上限为 2:1，实际的路网中，这一比例通常可达到 1.5:1。因此，对于大部分为双向行驶的路段来说，用对偶图法所产生的节点数目为原路网节点数目的 3 倍左右。其次，在一个交叉口处，原路网的弧段数通常是 4 条，而对偶图法最多所需构造的对偶链的数目为 12 条。可以估算，采用对偶图法能使拓扑数据中的弧段数最多增长 3 倍。这一点在大型城市网络中对路线优化算法的计算时间的影响是致命的。

再者，由于增设的大量的节点和对偶链和原有网络中的弧及节点混杂在一起，该算法同样要求能够判别原节点和对偶网络中派生的节点之间的关系，由此也增

加了算法的计算复杂度和编程的代价。

本文实验中的电子地图的路段属性信息可以使用对偶表示法，对程序的运行不会造成明显的影响。

3.2 最短路径算法分析

最短路径问题一直是计算机科学、运筹学、地理信息科学的一个研究热点，从图论方面看，最短路径问题就是指在带权有向图中，寻找从指定起点到终点的一条具有最小权值总合的路径问题。如果把权值看成是弧的长度属性，那么目标路径就是从起点到终点的最短路径，这也是其名称的由来。最短路径问题大致分为：单源最短路径问题和多源最短路径问题。从汽车实时导航的方面来看又分为：静态路径规划算法和动态路径规划算法^[25]。解决最短路径问题已经有了很多成熟的算法，下面会简单地介绍两个典型的算法，以及在下一章节中详细的分析本课题涉及的经典的 Dijkstra 算法。

3.2.1 迪杰斯特拉(Dijkstra)算法

迪杰斯特拉算法是一位荷兰数学家提出的一种适用于非负权值网络的单元最短路径算法，是目前求解最短路径问题的理论上最完备、应用最广泛的经典算法，它求出的是某一指定节点到图中其它所有有效节点的最短路径，这就是说，它属于单源最短路径问题。

在正式介绍 Dijkstra 算法之前，先介绍一下单源最短路径问题以及它的变形问题：

单目标最短路径问题：找出从每一节点 v 到某指定节点 u 的每条最短路径。

单对节点间最短路径问题：对于某给定节点 u 和 v ，找出从 u 到 v 的一条最短路径。如果解决了源节点为 u 的单源问题，则这一问题也就获得了解决。据市在最坏的情况下，从渐近意义上来分析，目前还没有比单源路径算法更好的算法来解决这一问题。

每对节点间最短路径问题：对于每对节点 u 和 v ，找出从 u 到 v 的最短路径。可以用单源算法对每个节点作为源点运行一次就可以解决这一问题。

Dijkstra 算法是一种贪心策略的最短路径算法^[26]。贪心算法的特点是所作的选择都是目前最佳的，它期望通过所作的局部最优选择产生出一个全局最优解。适用于贪心策略的问题一般具有两个特点：贪心选择性质和最优子结构。贪心选择性质是指：一个全局最优解可通过做局部最优选择来达到。在一个贪心算法中，

我们所作的总是当前最佳的选择,然后再解决作了该选择之后出现的子问题。贪心算法所作的当前选择可能要依赖于已经作出的所有选择,但不依赖于有待于作出的选择或子问题的解。一个问题呈现出最优子结构,如果它的一个最优解包含了其子问题的最优解。这个性质是用来对某问题中动态程序设计以及贪心算法的可应用性进行评价的关键一点。

Dijkstra 算法的主要思想是按照路径长度逐点增长的方法构造一棵路径树,从而得到从该树的根节点(即指定起点)到其它所有节点的最短路径^[27]。设集合 S 存放已经求出的最短路径的终点,初始状态时,集合 S 中只有一个源点 v_0 。以后每求得一条最短路径 (v_0, v_1, \dots, v_k) , 就将 v_k 加入到集合 S 中,直到全部顶点都加入到集合 S 中为止。

引入一个辅助变量 d , 它的每个分量 d_i 表示当前所找到的从源点 v_0 到其它顶点 v_i 的最短路径。它的初始状态为: 如果从 v_0 到 v_i 有弧线, 则 d_i 为弧上的权值, 否则令 d_i 为 ∞ 。设第一条最短路径为 (v_0, v_k) , 其中 k 满足

$$d_k = \min_i \{d_i \mid v_i \in V - v_0\} \quad (3-1)$$

那么下一条最短路径(终点为 v_j), 或者是 (v_0, v_j) , 或者是 (v_0, v_k, v_j) 。一般情况下, 假设 S 是存放已经求出的最短路径的终点集合, 那么下一条最短路径的中间节点一定是 S 中的节点, 其长度为:

$$d_k = \min_i \{d_i \mid v_i \in V - S\} \quad (3-2)$$

在每次求得一条最短路径之后, 其终点 v_k 加入集合 S , 然后对所有的其它顶点 $v_i \in V - S$, 修改其 d_i

$$d_i = \min\{d_i, d_k + c(v_k, v_i)\} \quad (3-3)$$

式中, $c(v_k, v_i)$ 是弧 (v_k, v_i) 权值。以上算法将产生从源点出发到达其他各项顶点的最短路径。对于单车辆路径规划, 只需要计算从源点到终点的一条最短路径, 即当发现最短路径的终点为目标路径的终点时, 算法终止。

根据上述基本原理, 设有带权有向图 $G = (V, E)$, 其中 V 包含 n 个顶点的顶点集合, E 是包含 m 条弧的弧集合, (v, w) 是 E 中从 v 到 w 的弧, $c(v, w)$ 是 (v, w) 的非负权值, 设 s 为 V 中的顶点, t 为 V 中可由 s 到达的顶点, 则求解从 s 到 t 的具有最小弧长权值和的最短路径搜索过程是这样的。

1) 对每一个顶点 v 分配三个信息, $k(v)$, $d(v)$, $p(v)$ 其中 $k(v)$ 是一个布尔型变量, 表明顶点 v 的最短路径是否已经求出; $d(v)$ 是从 s 到 v 的当前已知的最短路径长度的上界; $P(v)$ 是 v 的后项顶点指针, 并分别初始化为:

$$d(v) = \begin{cases} 0 & v = s \\ \infty & v \neq s \end{cases}; \quad k(v) = false;$$

2) 扫描 $k(v) = false$ 的顶点, 从中选择一个具有最小路径长度的顶点 v , 并令 $k(v) = true$; $d(v) = \min\{d(v_i) | k(v_i) = false, v_i \in V\}$

3) 检测每一个 $k(v) = false$ 且邻接与顶点 v 的顶点 w , 如果满足条件 $d(w) > d(v) + c(v, w)$, 则令 $d(w) = d(v) + c(v, w)$; $p(w) = v$

4) 重复执行第二步骤和第三步骤, 直到 $k(v) = true$ 。

5) 由 t 开始遍历后项顶点指针 p 直至源点 s , 即得到最短路径的解

$$P_{st} = \{v_0 = s, v_1, \dots, v_k = t\}, \text{ 其中 } v_i = p(v_{i+1}) \quad i = 0, 1, 2, \dots, k-1$$

对 Dijkstra 算法的时间复杂度进行分析的结果是 $O(n^2)$ 。分析一个算法的好坏主要看时间复杂度和空间复杂度, 它们代表着执行这个算法效果的好坏。一个算法的时间复杂度, 顾名思义, 就是执行该算法的时间代价。而算法运行的时间是以编写算法的实际代码在计算机中运行的时间来计算的, 由于影响算法运行的时间的因素很多, 诸如机器指令的读取、编译器的性能、处理器的性能等等很多客观因素, 因此, 一般采用大 O 表示法。基于各种运行数据结构的 Dijkstra 算法是解决路径规划问题的首选, 假设采用邻接矩阵作为图的存储结构, 该算法的位代码用 C 语言表示为^[28]:

```

1.  d[v] = 0; S[v] = 1;           //初始化
2.  for (int i = 0; i < n; i++) {   //主循环, 进行一条最短路径的计算
3.      double min = MAXNUM; int u = v;
4.      for (int j = 0; j < n; j++) //选择当前在 V - S 集合中的节点 u
5.          if (!S[j] && d[j] < min) { u = j; min = d[j] }
6.          S[u] = 1;
7.      for (int w = 0; w < n; w++) //更新当前最短路径长度
8.          if (!S[w] && d[u] + c[u][w] < d[w])
9.              d[w] = d[u] + w[u][w]
10. }

```

邻接矩阵是指用矩阵的形式来表示图中的各节点的路由拓扑关系, 假如图 3-4 是某一区域的抽象示意图。

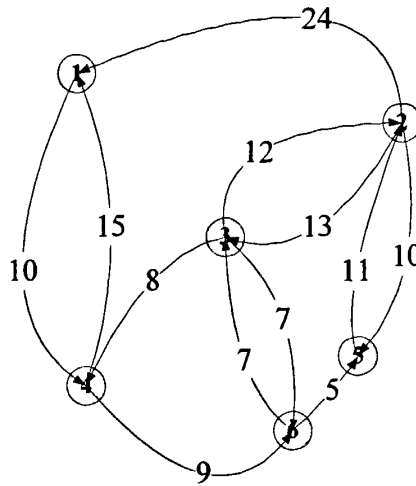


图 3-4 带权有向图

Fig 3-4 Graph with Weight

图 3-4 的邻接矩阵表示为：

$$A = \begin{bmatrix} \infty & \infty & \infty & 10 & \infty & \infty \\ 24 & \infty & 13 & \infty & 10 & \infty \\ \infty & 12 & \infty & 8 & \infty & 7 \\ 15 & \infty & \infty & \infty & \infty & 9 \\ \infty & 11 & \infty & \infty & \infty & \infty \\ \infty & \infty & 7 & \infty & 5 & \infty \end{bmatrix}$$

矩阵中的每一个元素都表示每两个节点之间的路径权值，通过这个矩阵就可以得到图 3-4 的所有拓扑关系。

从代码中知道，基于邻接矩阵的 Dijkstra 算法包括了一个双重嵌套的 for 循环语句，因此时间复杂度是 $O(n^2)$ 。

对于单车导航，一般的需求是计算道路网中从源节点到目标节点的最少开销，即用于实时定位导航情况。对 Dijkstra 算法进行改进，选取一定的搜索条件，就可以得到车辆从源节点到目标节点的最少开销路径^[29]。大致思想如下：

1. 对所有 $i \in V$ ， $p(i) = 0$ ；
2. 对所有 $i \in V$ 且 $i \neq r$ ， $l(i) \leftarrow \infty$ ；（初始费用设为无穷大）
3. $l(r) = 0$ ； $k = 0$ ； $G1 = \{r\}$ ； $G2 = \Phi$ ；
4. 如果 $G1 = \Phi$ ，则报错；
5. 否则{从 $G1$ 中选取一点记为 u ，满足 $l(u) = \min\{l(i)\}$ ，其中任意 $i \in G1$ ；

3.2.2 迪杰斯特拉(Dijkstra)算法的优化

通常，图有两种基本存储结构，我们用邻接矩阵和邻接表来存储路网信息。

使用邻接矩阵存储路网的最大优点在于它容易确定某一给定节点射入和发出的弧的集合。如图3-4的邻接矩阵所示，由于邻接矩阵有 n^2 个元素，故采用邻接矩阵存储路网的空间代价是 $O(n^2)$ ，而与弧的实际数量无关。这样，对于大型稀疏网络而言，利用邻接矩阵存储而与弧的，其数据冗余度过大，因而不适合。

邻接表是另一种常用的存储结构，常用于稀疏图。它为稀疏图G的每一个节点分配了一个链表，共计m个链表，节点 $v_i \in V$ 对应的链表包含的元素是邻接于节点 v_i 的节点集。链式存储结构相对于数组、线性表，在缩小冗余、减少内存空间占用、加快数据存取效率上有着绝对的优势。在某些应用情况下，链式存储结构是存储数据方法的唯一选择，例如在对于包含10000个顶点的网络图的数据存储中，用“邻接矩阵”法仅邻接矩阵就至少需要100M的存储空间，这显然是不现实的，所以必须采用基于链式存储结构的“邻接表”存储方法。邻接表是表示图内顶点之间相邻关系的N（N是图中顶点个数）个单向链表，N个单向链表形成顶点表。

一个实际的网络图及其邻接表的示意图如图3-6所示：

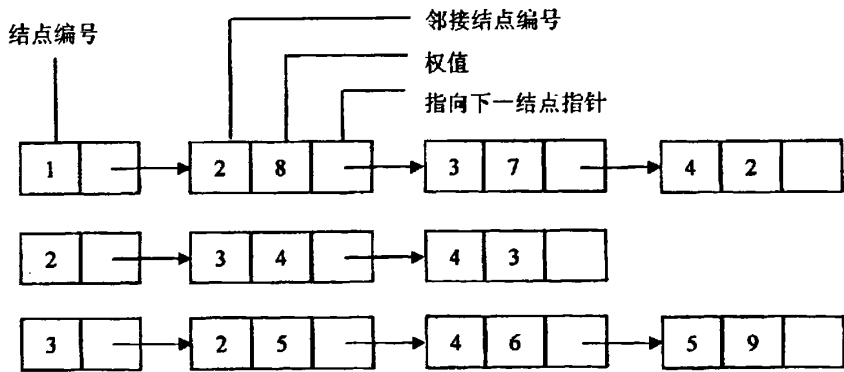


图3-6 邻接链表示意图

Fig 3-6 Linking Table Figure

根据定义，稀疏图有 $m = O(n)$ 条边。因此，用一个邻接表描述稀疏图需要的存储空间是 $O(n)$ ，显然优于用邻接矩阵描述稀疏图需要的存储空间 $O(n^2)$ ，但是由于通过链表来组织邻接于同一个节点的弧信息，通过指针来查询，对路径搜索的速度会有影响。

通过采用双向链表存贮路网上的节点，可以减少Dijkstra算法的比较次数，降低了空间复杂度，在寻找一条起点到终点的最优路径时，速度明显快于传统的

Dijkstra算法。节点的数据结构包括节点的坐标值，用于计算的其他属性值，一个前向指针指向上一个临时的最优节点，一个后向指针指向下一个预搜索节点。所有的节点都属于OPEN或者CLOSED节点列表。OPEN列表是已经生成但未扩展的节点集合，CLOSED列表是已经扩展的节点集合。扩展是指对某一节点生成了所有的子节点。这种方法其实是宽度优先搜索的一个特例，当所有的边具有相同的权重时，这种算法就变成了宽度优先搜索。

3.2.3 A*算法

目前在路径优化领域最流行的启发式搜索算法，它对实现道路网络的最佳优先搜索极为有效。该算法在选择下一个被检查的节点时，对当前节点距离终点的长度进行估计，评价其处于最优路径上的可能性量度，优先搜索可能性大的节点，从而提高了搜索效率。启发式搜索和 Dijkstra 算法不同，Dijkstra 算法是按照路径长度递增的顺序产生的最短路径，凡是到源点的最短路径长度少于目标点的路径长度的节点都会被搜索^[30]。启发式搜索是基于知识的搜索策略，通过选定一种估价函数，在搜索过程中的第一步，寻找估价函数数值最高的节点作为下一个搜索节点^[31]。利用启发式信息的有效方法是计算启发式函数，该函数估价每一生成节点处于最佳路径解上的可能性，从而优先搜索可能性大的节点，达到提高搜索效率的目的。

A*算法中引入当前定点 v 的启发式估计函数 $f'(v)$ ，定义为：

$$f'(v) = g(v) + h'(v) \quad (3-4)$$

式中， $g(v)$ 是从起点到当前顶点 v 的实际费用量度； $h'(v)$ 是从当前顶点 v 到终点的最小费用的估计，如果 $h'(v) = 0$ ，即没有利用任何启发式信息，这时的 A*算法就成为了 Dijkstra 算法。 $h'(v)$ 具体形式的选择取决于路径优化标准，在选择 $h'(v)$ 时，要满足一个要求，就是不能过高估计当前顶点 v 的最小费用，这被称为可纳性条件，只要启发式函数满足可纳性条件，且原问题存在最优解，则 A*算法一定能够计算出最优路径。如果选择最短距离为优化标准，那么以顶点累计权值 $d(v)$ 作为当前顶点的实际费用，以当前顶点到终点的欧式距离 $d'(v)$ 为最小估计函数，那么顶点 v 的启发式估计函数为：

$$f'(v) = d(v) + d'(v) \quad (3-5)$$

如果选择最少出行时间为优化标准，则可以定义顶点的启发式估计函数为：

$$f'(v) = t'(v) = g(v) + h'(v) = \sum_{i \in P(v)} \frac{d_i(v)}{V_i(v)} + \frac{d'(v)}{V'(v)} \quad (3-6)$$

式中, $t'(v)$ 为出行时间; $P(v)$ 为从起点到顶点 v 的当前最短路径; $d_i(v)$ 为路段 i 的长度; $V_i(v)$ 为路段 i 的最大行使速度。将 Dijkstra 算法中顶点累计权值 $d(v)$ 用 $f'(v)$ 代替, 即实现了 A* 算法。如果将 A* 算法中的最小费用估计函数进行下列修改, 就可以得到改进的 A* 算法, 即最佳邻点启发式搜索(Best Neighbor Heuristic Search)算法, 简称 B* 算法。该算法利用与当前顶点 v 相邻的所有顶点 w 定义新的最小费用估计函数 $h''(v)$

$$h''(v) = \begin{cases} 0 & w = t \\ \min\{c(v, w) + e(w, t) \mid (v, w) \in A\} & w \neq t \end{cases} \quad (3-7)$$

式中, A 表示路网的集合; 顶点 w 是 v 的后继顶点; $c(v, w)$ 是 (v, w) 的权值; $e(w, t)$ 是 w 到终点 t 的费用估计。由于 B* 算法扩展的所有顶点都可以由 A* 算法扩展, 反之不然, 因此, 在理论上具有更高的效率, 但是同时也增大了运算代价。

A* 算法中的估值函数起着非常重要的作用, 保证一定能够找到最短路径的充要条件是估价函数算出的两点间的距离必须小于等于世纪的距离。这个可以从数学上严格证明。如果你的估价函数不满足这点, 就只能叫做 A 算法, 并不能保证最后的结果是最优的, 但它可能速度非常的快。

A* 算法也是一个比较有效的搜索最短路径的算法, 但是不好确定估价函数。从理论上说, 一个完全正确的估价函数是可以非常迅速地得到问题的正确解答, 但一般完全正确的估价函数是得不到的, 因而 A* 算法不能保证它每次都得到正确解答, 一个不理想的估价函数可能会使它工作得很慢, 甚至会给出错误的解答。除此之外, A* 算法只适合处理静态路径求解, 对于交通容易堵塞的大城市来说很难实现。

A* 算法的主要程序设计思想是^[32]:

1. 对所有 $i \in V$, $p(i) = 0$;
2. 对所有 $i \in V$ 且 $i \neq r$, $l(i) \leftarrow \infty$;
3. $l(r) = 0$; $f'(r) = 0$; $k = 0$; $G1 = \{r\}$; $G2 = \Phi$;
4. 如果 $G1 = \Phi$, 则报错;
5. 否则{从 $G1$ 中选取一点记为 u , 使其具有最小费用, 满足

$$f'(u) = \min\{f'(i)\}, \text{ 其中任意 } i \in G1;$$
6. 从 $G1$ 中删去 u ;
7. $G2 = G2 \cup \{u\}$;

8. 如果 u 是目标节点, 那么跳到第 M 行};
9. 记 $Au = \{v | (u, v) \in A\}$, 对所有 $v \in Au$ 执行:
10. 如果 $v \in G1$
11. 那么 {如果 $l(u) + a(u, v) < l(v)$, 那么 $\{l(v) = l(u) + a(u, v); p(v) = u\}$ };
12. 如果 $v \in G2$
13. 那么 { 如果 $l(u) + a(u, v) < l(v)$, 那么 $\{l(v) = l(u) + a(u, v); p(v) = u;$
 $G1 = G1 \cup \{v\}$; 从 $G2$ 中删去 $v\}$ };
14. 如果 $v \notin G1$ 且 $v \notin G2$
15. 那么 $\{l(v) = l(u) + a(u, v); p(v) = u; f'(v) = l(v) + h'(v); G1 = G1 \cup \{v\}\}$;
16. 跳转到第 4 行
17. 输出路径结果

将节点的平均出度记为 b , 将从起始节点到目标节点的最短路径的搜索深度记为 d , 所谓搜索深度, 是搜索算法为了寻找最优解而必须遍历的树的层数。在这一概念下, 由于改进的最短路算法在完成搜索的过程中, 处理的树节点总数约为 b^d , 故其时间复杂度为 $O(b^d)$ 。

A*算法的时间复杂度也是 $O(b^d)$ 。由于 A*算法引入了已知估价函数作为判断下一个节点的依据, 如果选取适当的估价函数, 可以通过比较少的搜索次数而得到最优路径, 因而其总体的运算效率往往高于改进的 Dijkstra 算法。

3.3 本章小结

本章的主要工作是对实现路径规划功能的算法进行了分析和介绍。首先, 阐述了图论中一些重要的概念, 介绍了电子地图中的节点—路段数据模型, 对路段的方向性问题进行了讨论, 根据道路等级的高低分层存储路段信息。其次, 对两种典型的静态最短路径算法: Dijkstra、A*算法进行了分析, 对这两种算法进行最短路径的选择依据进行了阐述。

4 软件的设计和具体实现过程

导航电子地图的出现是当今城市化进程迅速发展的产物，是智能运输系统中的重要内容，是电子地图系统的功能扩展。本课题设计的导航软件是在 VB 环境下编写的，利用 MapX 控件进行二次开发，它具有电子地图的缩放、漫游功能，图层编辑功能、测距和测量范围的功能、鹰眼的功能以及路径规划功能。这样可以良好的与 MapInfo 格式的电子地图相结合，同时利用 MapInfo Professional Version 8.5 软件完成电子地图的制作工作。

导航软件采用的是模块化设计，下面介绍几个主要的功能模块，如图 4-1 所示：

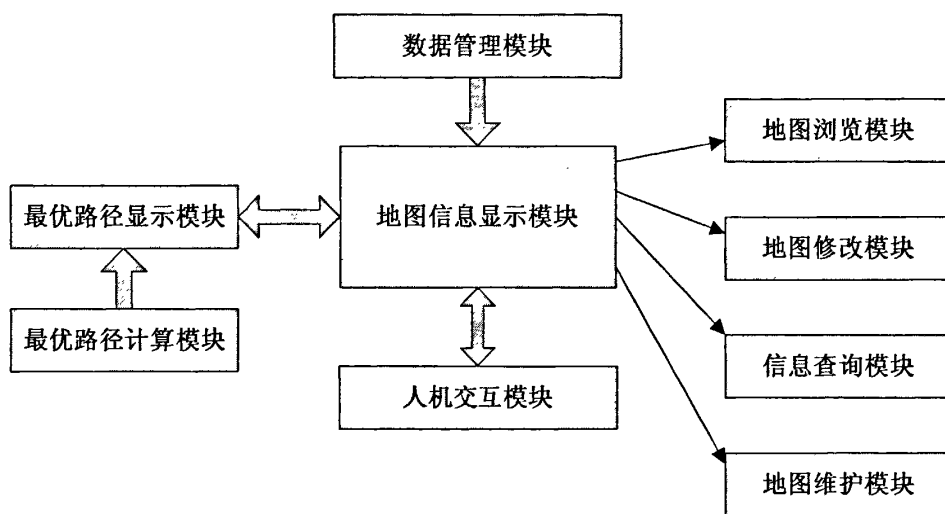


图 4-1 导航软件功能模块

Fig 4-1 Navigation Software Function Model

- 1) 地图浏览模块：主要实现电子地图系统的对地图的漫游、放大缩小、全图以及鹰眼浏览的操作。
- 2) 地图修改模块：主要实现对地图图元选择以及对图层和绘图属性的编辑和修改等图操作。
- 3) 信息查询模块：主要实现对地图现关信息的查询操作，然后在地图上定位，高亮显示查询结果，另外还可以完成距离测量和范围测量操作。
- 4) 地图属性维护模块：主要进行对地图的打开、保存、修改以及截图等对地图属性变更、图层设定的操作。
- 5) 数据库模块：主要实现链接数据库以及与 MapX 相关的地图属性数据相互绑定，从而在电子地图对应区域显示相应数据，以及对数据库的各项数据

的维护操作。

6) 最短路径计算模块：主要实现对最优路径的计算和选择操作。

4.1 电子地图浏览功能

电子地图一般包含大量的地理信息，在屏幕的范围内很难进行有效的操作。为了方便查询所需要的信息和良好的人眼视觉效果，电子地图的浏览功能是不可缺少的，它一般包括：放大、缩小、漫游、鹰眼等基本功能。这样，在需要整体显示地图的时候，可以只显示重要的标志性地理事物，而在局部显示地图时候，可以详细的显示地图的细节。

下面介绍这个模块中各个功能的具体实现：

1. “工具”菜单中的主要功能选项包括：

- 1) “放大”功能：把 CurrentTool 变量赋值为 miZoomInTool。
- 2) “缩小”功能：把 CurrentTool 变量赋值为 miZoomOutTool。
- 3) “漫游”功能：把 CurrentTool 变量赋值为 miPanTool。
- 4) “全部不选”功能：把每一个图层中的 Selection 集合都清空。

其中，CurrentTool 表示当前地图工具的变量，miZoomInTool、miZoomOutTool、miPanTool 分别是定义好的地图操作工具的常量名称^[33]。地图的放大和缩小功能指的是通过调整视图屏幕，是当前视图屏幕中的一部分区域放大显示到整个视图屏幕中，或进行相反的操作。矢量图形系统应具有无级放缩的功能。所谓“无级”，指的就是图形能够以任何的比例进行显示。实现图形放缩的方法很多，例如可以进行固定比例的放缩，即运行放缩功能，用鼠标点击地图中的某一点，系统会以该点为中心，按照一定的放缩比例进行地图显示，并且按照放缩的比例不同，选择性的显示地图中的地理事物，从而，避免在屏幕中显示大量的地理事物信息，影响视觉效果。

地图的漫游功能也可以认为是图形的移动，因为它通过调整视图的屏幕来显示图形的其他部分。漫游功能的实现方法和放缩功能的实现方法类似，不同的是比例保持不变，地图只是相对于屏幕坐标进行移动。

2. 电子地图的鹰眼功能

鹰眼功能是电子地图相关软件中一个基本的功能，鹰眼图又名缩略图，顾名思义，在鹰眼图上可以象从空中俯视一样查看地图框中所显示的地图在整个图中的位置。为了可以使你方便地观察一个目标（或区域）在整个城市中的地理位置，在鹰眼图中有一个矩形表示你要观察目标（或区域）在城市中的概略地理位置，这相当一个“鹰眼”，始终聚焦在某一个矩形区域。在电子地图中，用于显示当前窗

口在全图中的位置，当前窗口换图时，鹰眼自动进行相应变化。鹰眼进行漫游。通过改变鹰眼中窗口位置可改变相应的主窗口地图显示区域。

VB 下鹰眼图实现的思路是这样的：在某一 Form 上放两个 MapX 控件：Map1（主图），Map2（鹰眼图）；然后在鹰眼图上创建一个图层，在该图层上添加一个矩形 Feature，该矩形的大小随着主图边界而变化^{[34][35][36]}。具体的实现代码如下：

```

Dim m_Layer As Layer          '鹰眼图上临时图层
Dim m_Fea As MapXLib.Feature  '鹰眼图上反映主地图窗口位置的 Feature
Private Sub Form_Load()
Set m_Layer = Map2.Layers.CreateLayer("Rectlayer") '在 Map2 创建图层
End Sub
'根据 map1 的 Bounds 在 Map2 上绘制矩形 Feature
Private Sub Map1_MapViewChanged()
Dim tempFea As MapXLib.Feature '声明 Feature 变量
Dim tempPnts As MapXLib.Points '声明 Points 变量
Dim tempStyle As MapXLib.Style '声明 Style 变量
'矩形边框还没有创建时
If m_Layer.AllFeatures.Count = 0 Then
'设置矩形边框样式
Set tempStyle = New MapXLib.Style '创建 Style 对象
tempStyle.RegionPattern = miPatternNoFill '设置 Style 的矩形内部填充样式
tempStyle.RegionBorderColor = 255 '设置 Style 的矩形边框颜色
tempStyle.RegionBorderWidth = 2 '设置 Style 的矩形边框宽度
'在图层创建大小为 Map1 的边界的 Rectangle 对象
Set tempFea = Map2.FeatureFactory.CreateRegion(Map1.Bounds, tempStyle)
Set m_Fea = m_Layer.AddFeature(tempFea) '添加矩形边框
Else '否则，根据 Map1 的视野变化改变矩形边框的大小和位置
With m_Fea.Parts.Item(1)
.RemoveAll '除去已有的矩形边框的顶点
'添加大小和位置已变化的矩形边框的四个顶点
.AddXY Map1.Bounds.XMin, Map1.Bounds.YMin
.AddXY Map1.Bounds.XMax, Map1.Bounds.YMin
.AddXY Map1.Bounds.XMax, Map1.Bounds.YMax
.AddXY Map1.Bounds.XMin, Map1.Bounds.YMax
End With

```



```
m_Fea.Update      '更新显示
End If
End Sub
```

鹰眼功能的实现结果如图 4-2 所示，这是本导航软件加载的中国电子地图，右上角的矩形区域就是鹰眼功能的显示区域。

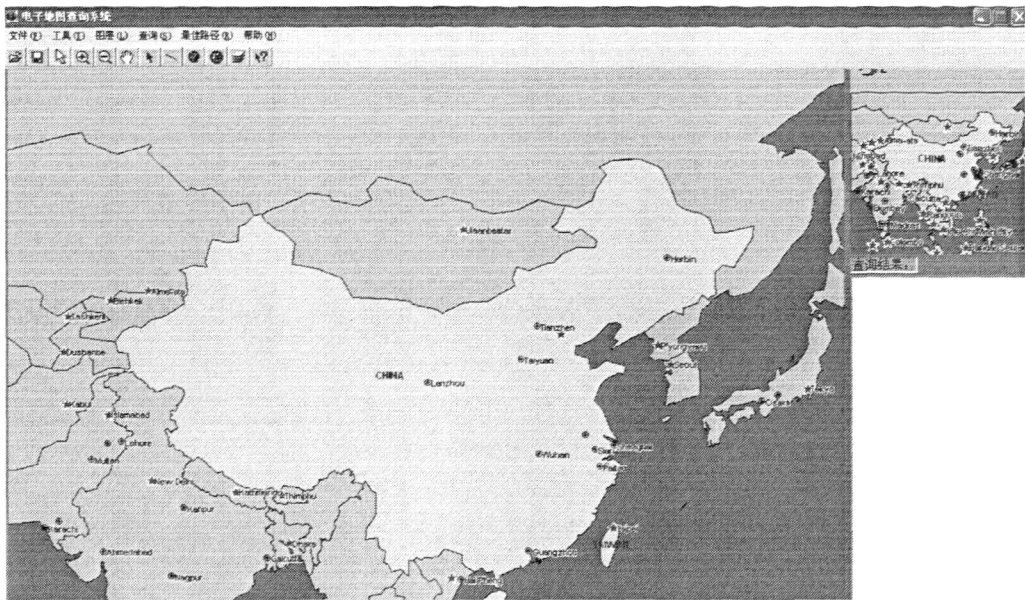


图 4-2 鹰眼功能示意图
Fig 4-2 Eagle Eyes Function

4.2 图层控制功能

图层控制功能模块主要是对电子地图中图层的数据组织进行管理，用户可以通过地图修改功能操作实现对地图图元选择以及对图层和绘图属性的编辑和修改等图操作。从而实现对地图的绘制，如点绘制、直线绘制、区域绘制、标签、文本标签的功能。在矢量图形系统中，图层用来对矢量图形元素进行分类管理。在已经建立了管理图层数据结构的基础上，可以很容易的实现与图层有关的操作。例如，要在系统中增加隐藏图层的功能，即当一个图层中处于隐藏状态时，这个图层中的所有图元对象不能进行绘制、选中等操作。对图层的控制管理，会影响到电子地图的显示结果^[38]。

“图层”菜单中的主要功能选项包括：

- 1) “加载图层”功能：首先要得到加载图层的所在路径，利用 Layers.Add 函数，把图层加载到软件中。

- 2) “加载图层集”功能：这是加载由 GeoSet Manager 工具生成的.gst 格式的文件，具体过程和“加载图层”相似。
- 3) “删除图层”功能：如图 4-3 对话框，利用 Layers.Remove 函数删除掉选中的图层。



图 4-3 “删除图层”对话框

Fig 4-3 “Delete Layer” Dialog Box

- 4) “图层可见”、“图层标注”、“移动图层”三项功能，分别利用 Layers.Visible、Layers.AutoLabel、Layers.Move 函数来完成，“图层可见”控制某一图层是否显示在屏幕上，“图层标注”控制某一图层上图元对象的标注是否显示，“移动图层”控制图层之间的上下的顺序关系。本文前面提到过，图层顺序正确与否直接影响电子地图的显示结果。具体如图 4-4、4-5、4-6 所示：



图 4-4 “图层可见”对话框

Fig 4-4 “Layer Visible” Dialog Box

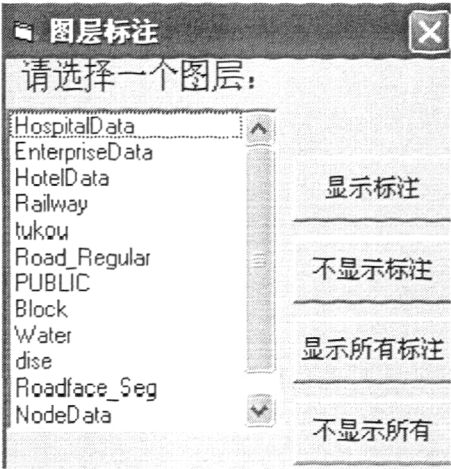


图 4-5 “图层标注” 对话框

Fig 4-5 “Layer Label” Dialog Box



图 4-6 “移动图层” 对话框

Fig 4-6 “Move Layer” Dialog Box

- 5) “图层控制”功能：该功能选项是一个比较全面的控制选项，因为它是直接调用 MapX 控件自带的对话框，如图 4-8 所示，它不仅具有上移、下移图层，添加、删除图层的功能，还具有使图层是否可见，是否可以编辑以及是否自动标注的功能。其中：显示对话框中可以控制该图层中的图元对象在一定的放缩比例下是否显示，标签对话框中也有类似的功能控制该图层对应的标签，另外还可以控制标签的位置、方向。这样可以避免在很小的范围内显示大量的地理事物信息，影响地图操作效果。

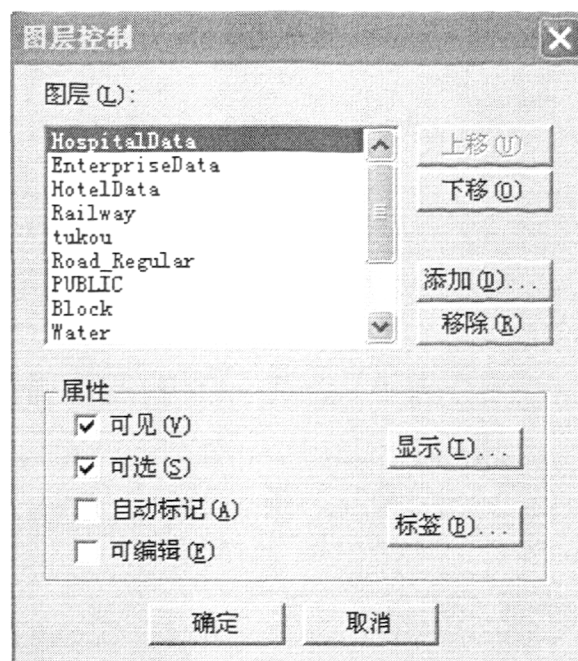


图 4-7 “图层控制”对话框

Fig 4-7 “Control Layer” Dialog Box

4.3 地理事物查询功能

在使用导航软件的过程中，会需要显示选定地理事物的各种属性信息，这些信息包括空间属性信息和非空间属性信息。通过这些有效信息，可以让用户充分了解到周围的地理信息，比如路长、路名、建筑物的名称等等。这就使得导航软件必须具有查询功能。通过图形连接单元和数据连结单元之间建立连接，可以将存储在数据表中的性质数据信息赋予个矢量图形元素。通过连接可以在图形连接单元和数据连接单元之间实现双向信息查询，即图形连接单元和数据连接单元彼此之间相互查询^[39]。

“查询”菜单中的主要功能选项包括：

- 1) “距离查询”、“范围查询”的功能类似，都是在地图上通过鼠标点击来确定要查询的距离或者范围。首先利用 CreateCustomTool 函数创造这两种工具，然后再调用这两个功能的函数时，把 CurrentTool 赋值为创建的上述两种工具。“距离查询”允许折线形式的线段，这样可以查询带有固定途径节点的起始和终点间的距离，查询结果如图 4-8；“范围查询”允许多边形范围的面积查询，这样可以更精确的测量待查询的区域，查询结果如图 4-9。



图 4-8 “距离查询” 结果示意图

Fig 4-8 “Distance Measure” Graph

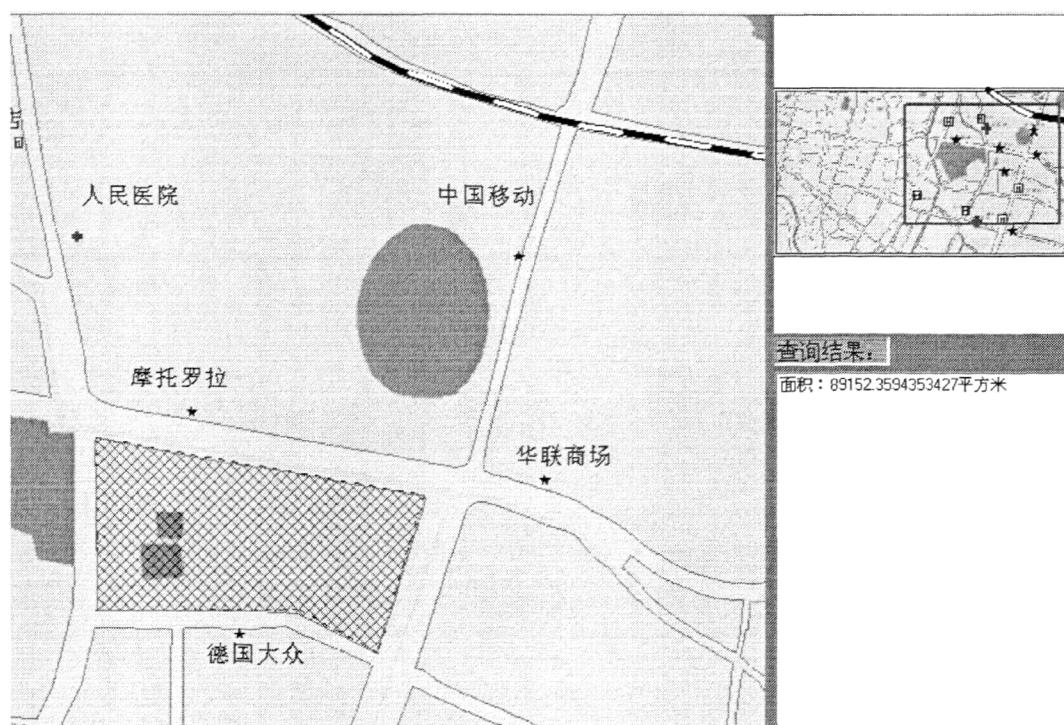


图 4-9 “范围查询” 结果示意图

Fig 4-9 “Area Measure” Graph

- 2) “地物查询”功能：该功能包括：“单点选择”、“矩形选择”、“圆形选择”、“多边形选择”四项子功能，分别对应不同的选择方式。“矩形选择”、“圆形选择”、“多边形选择”三种区域查询方式中可以根据三种

情况来判断该图元对象是否属于所查询范围，一是图元对象的质心点在查询区域内；二是图元对象全部在查询区域内；三是图元对象有部分区域在查寻范围内。

- 3) “精确查询”功能：它是根据图元对象所属的图层以及查询内容所属的域选项来精确的查询目标图元对象，返回的结果只有一个。如图 4-10 所示：

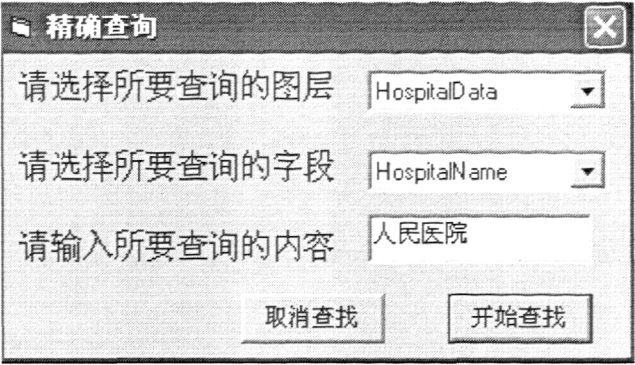


图 4-10 “精确查询”对话框

Fig 4-10 “Accurate Search” Dialog Box

下面把关键的部分程序讲述一下：

首先把电子地图中的所有图层对象都加入到 DataSets 中，其中：Formmain 是主界面名称；Map1 是 MapX 控件名称；DataSets 是第二章中提到过的数据绑定对象集合；lyr 是一个图层对象(Layer)。

Formmain.Map1.DataSets.Add miDataSetLayer, lyr, lyr.Name & "dataset"

其次是创建该图层的 Find 对象和创建该图层的 Find 对象的 DataSet，FindObj 定义为 Find 对象类型。

Set FindObj = Formmain.Map1.Layers(Combo1.Text).Find

Set FindObj.FindDataset = Formmain.Map1.DataSets(Combo1.Text & "dataset")

接下来创建查找的域(Field)，该域中包含图元对象的名称。

Set FindObj.FindField = FindObj.FindDataset.Fields(Combo2.Text)

然后是在该域中查找到对应图元对象名称的图元对象(Feature)，Fdat 定义为 FindFeature，Search 方法返回的查询对象都是 FindFeature 类型。

Set Fdat = FindObj.Search(Text1.Text)

最后是让查询到的图元对象处于选中状态。

Formmain.Map1.Layers(Combo1.Text).Selection.Replace Fdat

“精确查询”功能可以让你立刻定位到你所感兴趣的图元对象，从而对电子地图进行下一步的操作。查询结果如图 4-11，“人民医院”的图元对象被高亮显示，

右侧的“查询结果”栏中显示查询对象的名称。另外，也可以通过在电子地图上选择图元对象的图标，从而得到该图元对象的空间数据和属性数据。



图 4-11 “精确查询”结果

Fig 4-11 “Accurate Search” Graph

上面介绍了，软件与电子地图相关的功能，以及这些功能的实现的大致过程。通过以上这些功能可以很好的对电子地图进行浏览和编辑等操作。也可以把电子地图按照某种格式，如 BMP、JPG、GIF、TIF 等格式输出电子地图。

4.4 路径规划功能

路径规划功能是导航软件和地图软件的重要功能，它主要的功能是根据每条路径不同的开销权值，为使用者寻找一条合适的路线。它性能的好坏甚至直接决定了该软件的好坏。它的实现原理是指在指定的道路拓扑网络中，依靠节点—弧线的拓扑关系，按照一定的准则经典选择出路径权值累加和值最小的最佳路径，其中路径权值可以代表很多不同的实际意义，例如道路的长度、花费的时间、道路的收费等因素^[40]。最短路径算法 Dijkstra 算法是很有代表性的范例。它以良好的适应性得到了广泛的使用。

“路径规划”功能：首先在“路径规划”对话框中输入起始地址的名称和目的地址的名称，如图 4-12 所示：

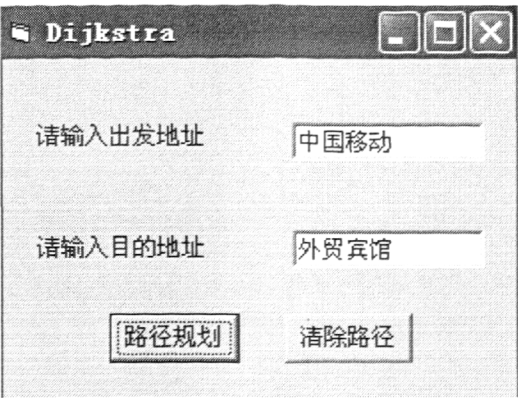


图 4-12 “路径规划”对话框

Fig 4-12 “Route Layout” Dialog Box

只有在输入了起始地址和目的地址的名称的时候，“路径规划”功能的按钮才被激活，而且当前地图有路径规划的模拟路径的时候，要先清除掉已有的规划路径，然后才能进行下一次路径规划。图 4-14 的路径规划结果，起始地点是“中国移动”，目的地点是“外贸宾馆”，在右侧的“查询结果”中按照途径顺序列出了最优路径经过的节点的名称，这样可以更好的理解路径规划算法的工作原理。如图 4-13 所示，经过长时间的测试以及加载大数据量信息地图都能够很好的完成路径规划的功能，而且运算时间也适合一般工作的需要。



图 4-13 “路径规划”结果示意图

Fig 4-13 “Route Layout” Graph

“路径规划”功能的整体流程图如下：

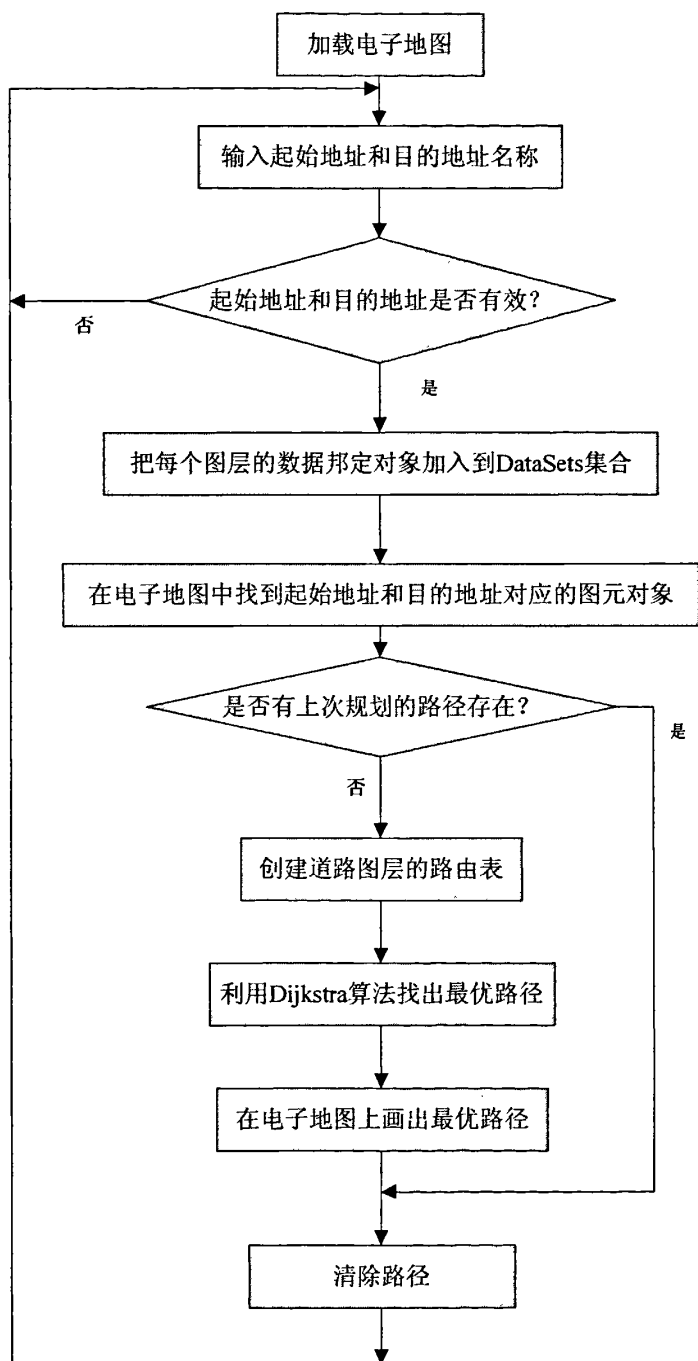


图 4-14 “路径规划”功能整体流程图

Fig 4-14 Flow Chart of Route Layout

在图 4-14 的整体流程图中,最重要的是利用 Dijkstra 算法计算出最优路径这个模块,因为它直接决定着整个功能的实现是否正确。如图 4-15 所示:

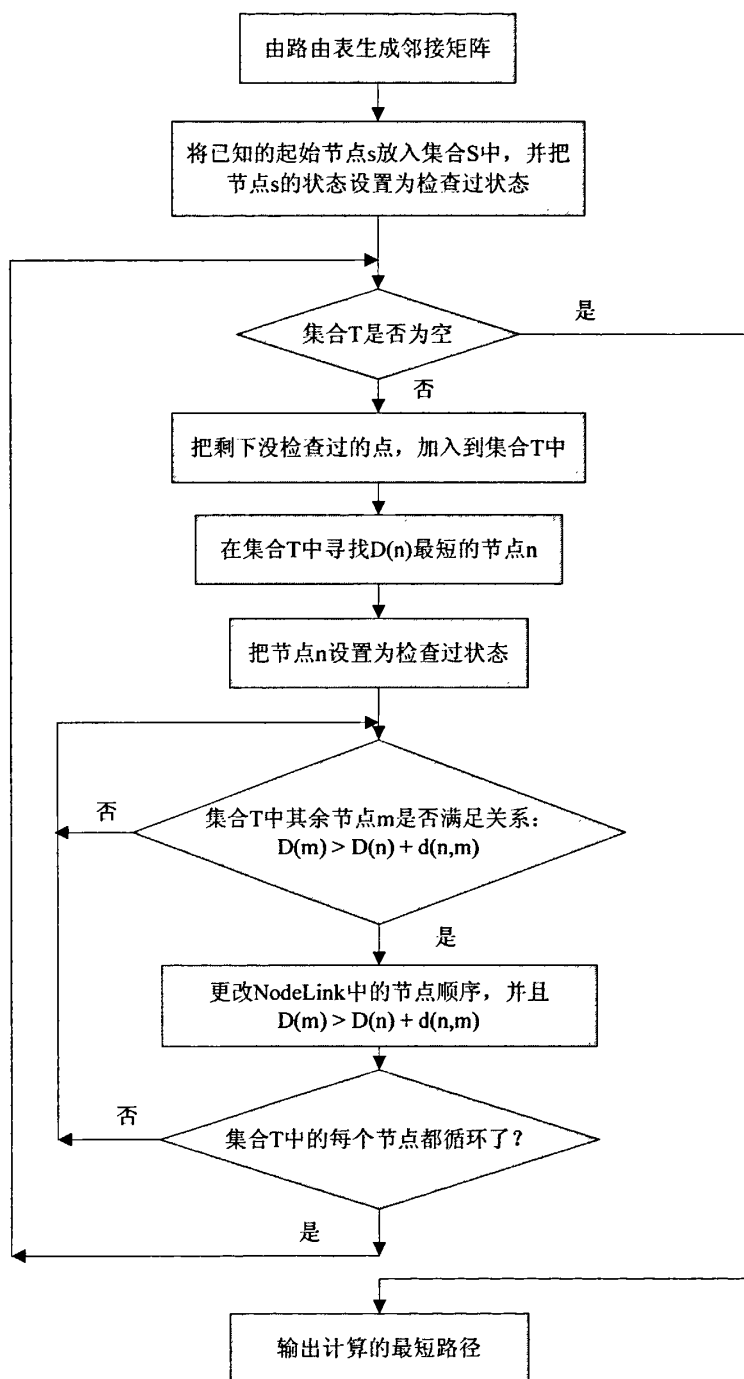


图 4-15 Dijkstra 算法的流程图

Fig 4-15 Flow chart of Dijkstra Arithmetic

根据图 4-14、4-15，对于“路径规划”功能的整体流程已经有了清晰的了解，接下来，对流程图中关键的模块的具体代码进行必要的分析。

1. 把每个图层中的数据绑定对象加入到 DataSets

因为数据绑定是将数据源中的数据导入 MapX 的过程，如果程序的操作涉及到图元对象的属性数据，那么一定要首先将电子地图图层的 DataSet，即数据绑定对象，加入到 DataSets 中，这样才可以通过 Find、Search 等方法查找到指定的图元对象。

```
For Each lyr In Formmain.Map1.Layers
    Formmain.Map1.DataSets.Add miDataSetLayer, lyr, lyr.Name & "dataset"
Next
```

其中，Formmain.Map1.Layers 是 MapX 对象所有图层的集合，lyr 是图层对象 Layer，miDataSetLayer 是用于创建数据集，引用源自 MapInfo 表的的字段，lyr.Name 是图层的名称，Formmain.Map1.DataSets.Add 是添加 DataSet 的方法，如本电子地图中主要的图层有：NodeData、RoadData、EnterpriseData、HotelData、HospitalData 等，它们各自的 DataSet 加入到 DataSets 的名称为：NodeData dataset、RoadData dataset、EnterpriseData dataset、HotelData dataset、HospitalData dataset。

2. 在电子地图上查找到起始节点和目的节点对应的图元对象

只有首先查找到起始节点和目的节点的图元对象，才能根据创建的邻接矩阵，即 NodeTable，初始化起始节点到其它节点的距离 d。后面路径的绘制也需要知道起始节点和目的节点的图元对象。

```
Set lyrds = Formmain.Map1.DataSets("NodeData" & "dataset")
Set FindObj = Formmain.Map1.Layers("NodeData").Find
Set FindObj.FindDataset = lyrds
Set FindObj.FindField = FindObj.FindDataset.Fields("NodeName")
Set FindFtrSt = FindObj.Search(StartName)
StartNode = lyrds.Value(FindFtrSt, "NodeNum")
Set FindFtrEd = FindObj.Search(EndName)
EndNode = lyrds.Value(FindFtrEd, "NodeNum")
```

其中，lyrds 是 Dataset 类型，FindObj 是 Find 对象类型，FindFtrSt、FindFtrEd 分别是查找到的起始节点和目的节点对象，StartNode、EndNode 分别是起始节点和目的节点的编号，FindObj.FindField 是查找对象依据的查询域，这里是“节点的名称”，因为输入的内容就是节点的名称。因此通过上面的代码，可以完成通过输入起始节点和目的节点的名称查找到起始节点和目的节点对应的图元对象和节点编号，图元对象是为了绘制规划的路径，节点编号是为了查询与节点相连的路径。

3. 创建路由关系矩阵 NodeTable

路由关系矩阵 NodeTable, 也就是邻接矩阵, 它是以矩阵形式存储电子地图中节点之间的路由关系, 包括是否连接、路径的权值。后面的最短路径算法也是根据这个邻接矩阵来选取路径。

```
Set lyrds = Formmain.Map1.DataSets("RoadData" & "dataset")
Set Ftrs = lyrds.Layer.AllFeatures
For Each Ftr In Ftrs
    fnode = lyrds.Value(Ftr, "FNode")
    tonode = lyrds.Value(Ftr, "TNode")
    lenode = lyrds.Value(Ftr, "Length")
    NodeTable(fnode, tonode) = lenode
Next
```

其中, Ftrs 是图元对象集合 Features, fnode、tonode 分别是某一条路径的头节点编号和尾节点编号, lenode 是该路径的权值。首先通过“道路”图层的数据绑定对象, 得到所有的“道路”图层中的图元对象, 同时得到每个道路对象的头节点编号和尾节点编号以及该道路的权值, 把权值赋到矩阵中对应的位置。

4. 利用 Dijkstra 算法计算出最优路径

正如图 4-15 中所描述的, 首先把所有图层的节点分为两个集合, $\{S\}$ 中的元素是已经检查过的点, 最开始的时候 $\{S\}$ 中只有起始节点 s , $\{T\}$ 中的元素是没有每检查的节点, 最开始的时候 $\{T\}$ 中包括除起始节点 s 之外的所有节点, 最后当 $\{T\}$ 中的元素个数为空的时候, 停止计算, 得到从起始节点 s 到其它节点的最优路径。

```
For i = 1 To NUM
    Distance(i) = NodeTable(StartNode, i)
Next i
```

其中, Distance() 是一个一维数组存放的是起始节点到其它节点的最短路径, 如果两个节点不相通, 赋值 ∞ 。

```
For j = 1 To NUM
    If Flags(j) = False Then
        TempNode(Sum_Temp) = j
        Sum_Temp = Sum_Temp + 1
    End If
Next j
```

其中, Flags() 是标志位数组, 数组元素是 Boolean 类型, 用来标志节点是否已经被检查过。TempNode() 是用来存放没有被检查过的节点的编号的数组,

Sum_Temp 是整形变量记录着没有被检查过的节点的个数。

```

For i = 1 To Sum_Temp
    If Distance(TempNode(i)) < Distance(Min_Temp) Then
        Min_Temp = TempNode(i)
    End If
Next i
Flags(Min_Temp) = True

```

其中, Min_Temp 是没有检查过的节点中距离起始节点 s 最近的节点的编号, 并且改变该节点对应的标志位, 使其成为检查过状态。

```

For k = 1 To Sum_Temp
    Num_Temp = TempNode(k)
    If Distance(Num_Temp) > Distance(Min_Temp) + NodeTable(Min_Temp,
Num_Temp) Then
        Distance(Num_Temp) = Distance(Min_Temp) + NodeTable(Min_Temp,
Num_Temp)
        i = 1
        j = 1
        While NodeLink(Min_Temp, i) <> -1
            NodeLink(Num_Temp, j) = NodeLink(Min_Temp, i)
            i = i + 1
            j = j + 1
        Wend
        NodeLink(Num_Temp, j) = Num_Temp
        NodeLink(Num_Temp, j + 1) = -1
    End If
Next k

```

其中, If 语句的作用是查看节点是否满足 $d(w) > d(v) + c(v, w)$ 的要求, 如果满足, 则 $d(w) = d(v) + c(v, w)$, 并且更改起始节点 s 到该节点的节点连接顺序。当把 TempNode() 中所有的节点循环完之后, 重新开始建立集合 $\{T\}$ 中的元素。这样集合 $\{T\}$ 中的元素的个数会越来越少, 直到为空集合, 此时路径规划计算完毕。

5. 绘制计算出的路径

前面已经计算出了最优路径的节点顺序, 但是还要将它们电子地图中显示出来, 同时还要在“查询结果”中显示出途径的一些节点。

```
Set lyrds = Formmain.Map1.DataSets("NodeData" & "dataset")
```

```
Set FindObj = Formmain.Map1.Layers("NodeData").Find
Set FindObj.FindDataset = lyrds
Set FindObj.FindField = FindObj.FindDataset.Fields("NodeNum")
Set FindFtr = FindObj.Search(StartNode)
Set RouteNode = Formmain.TreeView1.Nodes.Add(, "Route", "途径地点")
RouteNode.Expanded = True
Formmain.TreeView1.Nodes.Add RouteNode, tvwChild, , FindFtr.Name
Set Pnt = New MapXLib.Point
Set Pnts = New MapXLib.Points
Pnt.Set FindFtr.CenterX, FindFtr.CenterY
Pnts.Add Pnt
i = 1
While NodeLink(EndNode, i) <> -1
Set FindFtr = FindObj.Search(NodeLink(EndNode, i))
Formmain.TreeView1.Nodes.Add RouteNode, tvwChild, , FindFtr.Name
Pnt.Set FindFtr.CenterX, FindFtr.CenterY
Pnts.Add Pnt
i = i + 1
Wend
```

上面代码的主要作用是根据 NodeLink() 数组中的节点的顺序, 依次来画出最优路径的曲线, 大致的思想是: 首先, 通过“节点”图层的数据绑定对象, 依照节点编号, 依次找到其在图层中对应的图元对象, 以及坐标。然后创建临时图层, 并在临时图层上绘制路径曲线, 这样, 当进行下一次路径规划时, 可以方便的清除已有的路径。

Dijkstra 算法和 A* 算法以及双向搜索算法相比较, 它的运算时间最长, 主要是因为 A* 算法和双向搜索算法都有各自的优化和改进。其实主要影响 Dijkstra 算法的因素是路网的规模, 它在很大程度上影响了路径规划的效率。采用多层地图的分级搜索技术可以实现对搜索空间的控制, 减少运算时间。

在实际的路径规划中会出现各种复杂的情况, 如: 时间花费最小的最优路径、距离最短的最优路径、单向路段和转向受限的路段导航。这些导航条件时刻影响着导航结果的正确性, 不仅影响到使用者的驾驶情况, 更有甚者会造成安全危险。下面逐一列举出在各种情况下的路径规划结果如下图所示:

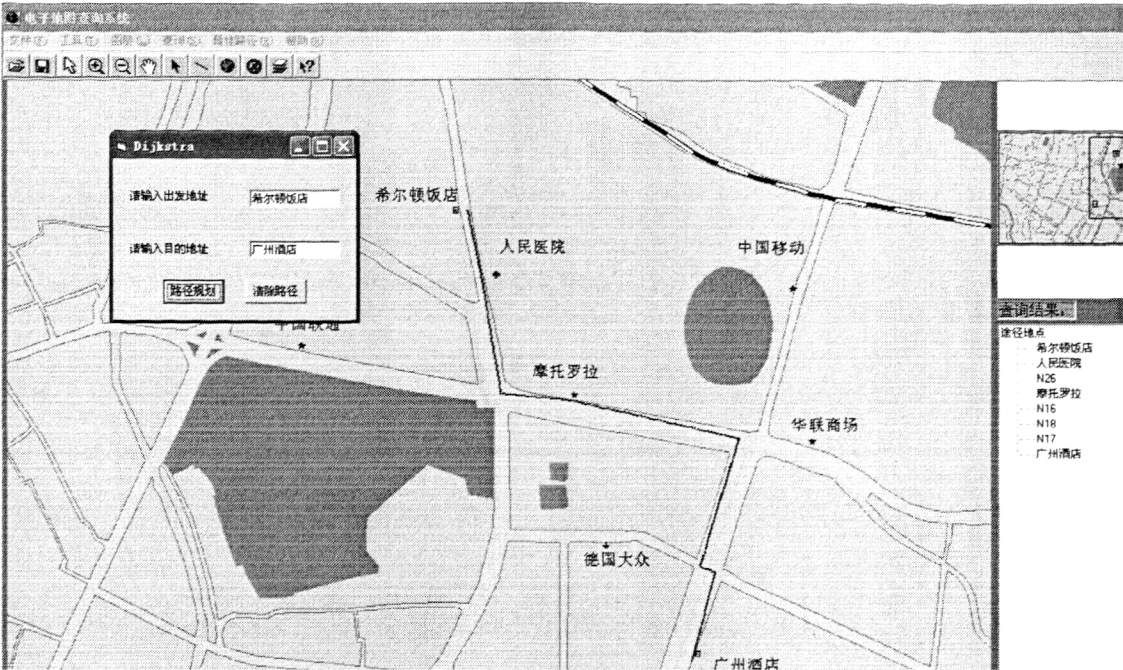


图 4-16 距离最短路径规划图

Fig 4-16 Shortest Distance Route Planning

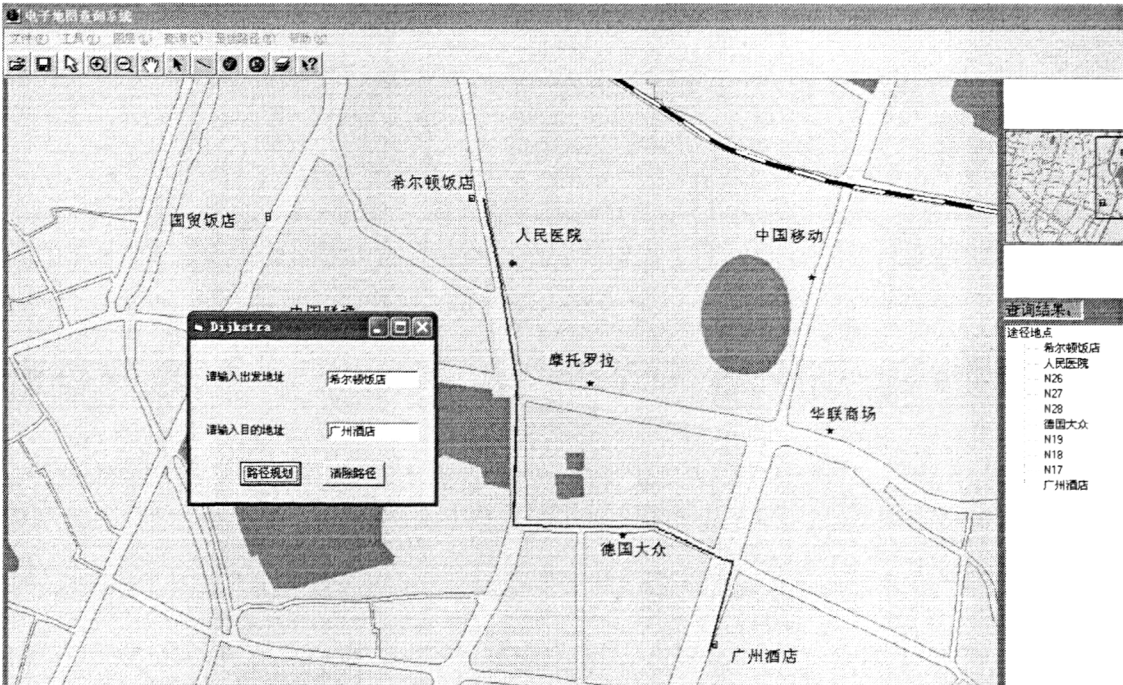


图 4-17 时间最短路径规划

Fig 4-17 Least Time Costing Route Planning



图 4-18 道路受阻情况下的路径规划

Fig 4-18 Route Planning Of Road Disable



图 4-19 单向道路情况的路径规划

Fig 4-19 Route Planning Of Single Direction

图 4-16 的出发地点是“希尔顿饭店”，目的地点是“广州酒店”。以最短距离为依据进行路径规划，途中经过了“人民医院”、“摩托罗拉”和四个道路节点。图 4-17 的出发地点和目的地点是一样的，按照最少花费时间来进行路径规划，得到的是经过辅路的时间最短路径。图 4-18 是在三条道路禁止行驶的情况下，寻找的最优路径，可以看到相同的两节点之间的最优路径与前两幅图差距很大。图 4-19 是当两条较近的路线由于单向行驶的限制时，往返的最优路径规划的出的是与图 4-16 和图 4-17 不同的路线。由上面四幅图片可以看出路径规划算法可以很好的适应道路的各种复杂情况，这也是 Dijkstra 算法被广泛应用的一个重要原因。

4.5 本章小结

本章的工作重点是根据导航软件的几项重要的功能，依次介绍了地图浏览功能、图层控制功能、地理事物查询功能、路径规划功能的开发过程。对开发过程中涉及的一些关键问题都给出了可行的解决办法，并且在介绍完每项功能的实现方法之后，都给出了相应功能的操作结果。在路径规划功能中，根据路段的不同情况，相同两个节点之间的最优路径结果会发生变化，Dijkstra 算法能够很好的根据当时的路况信息计算出合理的路径。

5 总结和展望

5.1 总结

近些年来,智能交通控制(ITS)学科领域的知识在实际生活中的应用是越来越广泛。最优路径规划是车辆导航中必不可少的一环,而路径规划的算法有很多种,因为所应用的领域不同,它们的性能也有所不同。所有的路径算法都可以抽象为图论中的最短路径的数学模型问题。

作为在校期间学习和实践的总结,本课题主要完成了下面一些工作:

1. 介绍了两种典型的拓扑模型结构,着重介绍了节点—道路数据模型下的 MapInfo 格式的电子地图,利用 MapInfo Professional 软件制作了一款区域电子地图,该地图的地理事物和道路具有各自的空间数据和属性数据信息,对 MapX 控件的对象模型和功能进行了必要的介绍。
2. 介绍、分析了两种静态路径规划算法: Dijkstra 算法、A*算法,对单源路径规划问题作了分析,并结合实验提出了相应的改进方法。
3. 对软件的设计和实现有了一套可行的方案,并对开发软件过程中的关键技术都进行了探讨,最后软件可以对电子地图进行浏览、操作、实现路径规划功能。验证了 Dijkstra 算法在一般的电子地图上都有着良好的表现。

5.2 展望

本课题的工作还有很多不足之处,而且工作远远没有结束,本课题只涉及到静态路径规划算法和单源最短路径问题的研究,接下来将会和 GPS 模块紧密联合起来,对于动态路径规划算法和多源最短路径问题进行深入的研究。这样会涉及到 GPS 接收、地图路况实时更新以及 GPRS 无线通信等一些新的问题,这些问题需要进一步的分析和研究。

随着计算机技术和通信技术的飞速发展,路径规划和地理信息已经成为人们生活中不可缺少的一部分,而且随着和其他通信设备、手持终端设备相结合,它会更大程度上的融入生活,服务于不同群体的对象。

参考文献

- [1] 吴晓强, 李鹏, 曲卫民编译. 智能交通系统研究回顾与展望. 国外公路. 2000, 20(4). 36-40
- [2] 梁晓艳译. 美国智能交通系统的发展. 山东交通科技. 1999, 3. 71-75.
- [3] 陈明杰. 车载定位导航系统的设计与实现. 哈尔滨工程大学. 2002, 12. 21-25
- [4] Kanwaljit, David Turner, Michael Shaw. Modernization of the Global Position System. ON GPS 2000, 19-22 Sep.2000, Salt Lake City, UT, 2175-2183. 16-19
- [5] 张其善, 吴今培, 杨东凯. 智能车辆定位导航系统及其应用. 科学出版社. 2002.7 52-60
- [6] 赵亦林著. 谭国真译. 车辆定位与导航系统. 电子工业出版社. 1999. 35-41
- [7] 刘光运. 电子地图技术与应用. 测绘出版社. 1996. 26-33
- [8] 杨天军, 杨晓光. GPS/GIS 车辆实时监控调度系统研究. 城市交通. 2004 第一期
- [9] 李德仁. 地理信息系统导论. 测绘出版社. 1993. 32-38. 7-15
- [10] Albert J. Geospatial information standards, a comparative study of approaches in the standardization of geospatial information. Computer & Geoscience. 1999. (25). 9-25
- [11] 王恒. 电子地图的建立及其运用. 电子科技大学. 2004. 6. 15-18
- [12] Peter A. Burrough, Racheal A. McDonnell. Principles and Geographical Information Systems. Oxford University Press. 1998. 26-34
- [13] Hinrich Claussen. Status and Directions of Digital Map Databases in Europe. In IEEE-IEE Vehicle Navigation & Information Systems Conference. Ottawa-VNIS.1993. 56-62
- [14] Susan Scott. Status and direction of map database standards efforts in North America. In: IEEE-IEE Vehicle Navigation & Information Systems Conference. Ottawa-VNIS. 1993. 42-49
- [15] Masao Shibata. Current status and future plans for digital map databases in Europe. In: IEEE-IEE Vehicle Navigation & Information Systems Conference. Ottawa-VNIS. 1993. 38-45
- [16] Paul A. Longley, Micheal F. Goodchild, etc. Geographic Information System(Volume 1)-Principles and Technical Issues. New York: John Wiley & Sons, Inc. 1999. 88-92
- [17] 李湘吉. GIS 数据空间理论与空间分析算法的研究和应用. 东北大学. 2002, 2. 46-57
- [18] 刘春. 导航电子地图中道路数据的空间索引和组织. 工程勘察. 2003. 1. 38-42.
- [19] 许五弟编著. 地理信息系统构建与应用. 中国建材工业出版社. 2005.8. 26-32
- [20] 詹舒波. 导航电子地图技术及其应用研究. 北京. 北京航空航天大学博士学位论文. 1996. 1.
- [21] 罗云启, 曾琨, 罗毅. 数字化地理信息系统建设与 MapInfo 高级应用. 北京. 清华大学出版社. 2003. 4. 116-135
- [22] 刘光等. 地理信息二次开发实例教程. 北京. 清华大学出版社. 2004. 7. 103-124
- [23] 殷剑宏, 吴开亚. 图论及其算法. 合肥. 中国科学技术大学出版社. 2003. 7. 22-39
- [24] 丁胜昔. 基于数字道路地图的车辆导航系统研究. 北京. 北京航空航天大学博士论文. 2004. 37-48
- [25] 刘奇志. 几种高效率的最短路径算法. 应用数学学报. 61-69.

- [26] 李元臣, 刘维群. 基于 Dijkstra 算法的网络最短路径分析. 微计算机应用. 2004. 25(3). 295-298
- [27] 乐阳, 龚健雅. Dijkstra 最短路径算法的一种高效率实现. 武汉测绘科技大学学报. 1999
- [28] Mark A. Allen Weiss. Data Structures and Algorithm Analysis in C. Pearson. 1996
- [29] Zhan F.B. Three Fastest Shortest Path Algorithm on Real Road Networks. Journal of Geographic Information and Decision Analysis I 1997 (1). 69-82
- [30] CALDWELL T. On Finding Minimum Routes in a Network with Turn Penalties. Communication of the ACM. 1961 4(2). 107-108
- [31] Stefan Edelkamp, Stefan schrodl. Route Planning and map Inference with Global Positioning Traces. Computer Science in Perspective: Essays Dedicated to Thomas Ottmann. 2003. 128-151
- [32] Adler J L. A Best Neighbor Heuristic Search for Finding Minimum Paths in Transportation Networks. Transportation Research Board 77th 1 Annual Meeting. 1998. 143-152
- [33] MapInfo Professional Guide. New York. 2003
- [34] MapX Developer's Guide. New York. 2002
- [35] 洪国胜编著. Visual Basic 5.0/6.0 范例教程. 北京. 中国铁道出版社. 1998. 137-169
- [36] 何斌, 王运坚, 冯峰, 刘醒. Visual Basic 6.0 应用指南. 北京. 人民邮电出版社. 148-192
- [37] 潘金贵, 顾铁成, 曾俭, 滕远方等编译. 现代计算机常用数据结构和算法. 南京. 南京大学出版社. 1994. 362-366
- [38] 陈建春. Visual C++ 开发 GIS 系统——开发实例剖析. 北京. 电子工业出版社. 2000. 162-163
- [39] 赵玲. 基于 MapInfo 的城市公交信息查询系统的研究与实现. 中南大学. 2003
- [40] 张歆奕, 吴今培, 张其善. 车载导航仪中路径规划算法及其实现. 计算机自动测量与控制. 2001. 9. 15-17

作者简历

高乾. 男. 生于 1982 年 7 月 31 日. 汉族.

教育经历:

2006 年-9 月—2007 年 6 月 北京交通大学 通信与信息系统专业 硕士学位

2001 年 9 月—2005 年 7 月 燕山大学 通信工程专业 学士学位