

Implementation of Intersection Support Subsystem in Vehicle Navigation System

Abstract

With the abundant development of electronic technology, communication technology and computer technology, the scientists begin to apply the electronic information technology to the traffic system in order to improve the efficiency and security of the transportation more and more, and a special field has been formed finally --- "Intelligent Transport System". To intellectualize the traffic system, we need to bring transformation to the whole traffic system. Vehicle --- as the main user of road network, the general way is to install an additional navigation system at present.

In current vehicle navigation system, equipments such as GPS, TV, CD, AM/FM, Vedio, Communication teminations have been combined into the base functions. Besides the traditional functions, driver can use navigation system to control them, such TV, phone, air-condition, even to access Internet. Vehicle navigations gradually become a informations platform.

For navigation system manufacturers, combination with more new technology becomes the best method to keep compatibility of product gradually. In this paper, intersection support is regarded as a research point, from this start point we find a series of ideaes and try to implement them. Finally, we find some good results.

Currently, Driver support functions mainly include warning about the status around the vehicles and communication between vehicles. According to the high frequency of accident happened on the car, this article regards the intersection surport as the entry point, makes a series of feasible resolution and implement it. At last, we find a good result.

At the beginning of the paper, the background information of the WinCE embedded platform and related software and hardware system are introduced firstly. After that, we describe the architecture of the driver support system and the design and implement of some main functions in it. Finally, some result is given by image.

Key words: Embedded System, Vehicle Navigation System, Intersection Support

独创声明

本人声明所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。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诚挚的谢意。

学位论文作者签名: 李海

签字日期: 2006.4.4.

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北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、交流。

（如作者和导师同意网上交流，请在下方签名：否则视为不同意）

学位论文作者签名: 导师签名:

签字日期: 签字日期:

第一章 绪 论

本系统是基于嵌入式系统 WinCE 平台下，实现车载导航系统中的交叉点支援子系统。为了便于对以下章节的理解，首先介绍一下车载导航系统的发展状况和嵌入式系统的一些背景知识。

1.1 嵌入式操作系统简介

随着数字信息技术和网络技术的高速发展，我们已经进入了数码时代。建立在其基础上的嵌入式系统已经广泛地渗入到科学研究、工程设计、军事技术、航天技术、商业文化艺术、娱乐业以及我们日常生活等各个方面。国内外一些嵌入式产品（如机顶盒、PDA、车载电脑等）的开发设计和推广，使得人们的生活与嵌入式技术更加紧密相关了。

1.1.1 嵌入式操作系统的发展

嵌入式这个概念在 20 世纪 60 年代的时候就已经存在了。当时在通信方面用于对电子机械电话交换的控制，被称为“存储式程序控制系统”（Stored Program Control）^[1]。而真正的发展时期是从 1971 年 11 月 Intel 公司推出的第一款处理器 Intel4004 开始的，之后嵌入式系统经历了几个发展阶段。尤其是进入 90 年代后以计算机和软件为核心的数字化技术取得了迅猛发展，不仅广泛渗透到社会经济、军事、交通、通信等相关行业，而且深入到家电、娱乐、艺术、社会文化等各个领域，掀起了一场数字化技术革命。多媒体技术与 Internet 的应用迅速普及，消费电子、计算机、通信（3C）一体化趋势日趋明显，嵌入式技术再度成为一个研究热点。综观嵌入式技术的发展，大致经历了表 1.1 中描述的 4 个阶段^[2]。

1.1.2 WinCE 系统平台介绍

Windows CE 作业系统是 Windows 家族中最新的成员，专门设计给掌上型电脑（HPCs）所使用的电脑环境。这样的作业系统可使完整的可携式技术与现有的 Windows 桌面技术整合工作^[3]。

Windows CE 的使用从可以随时随地使用的电脑到智慧型家电用品及丰富的多媒体家庭剧院，Microsoft Windows CE 为工作、家庭及其间的任何一点开启了动态的、崭新的开发远景。这个模组化、可自订的作业系统将 Windows 平台延伸到桌面之外，到达更小、更机动性、更特别的装置之上，然而它的 Windows 血统则保证了它的相容性，并且

支援了更广泛的开发基础。Microsoft Windows CE 揭露了革命性的系统架构，可以为开发者或科技领导者扩展消费者及工业电子上的新市场。

表 1.1 嵌入式技术发展阶段
Table 1.1 The Phases of Development of Embedded Technology

发展阶段	系统表现形式	应用领域及特点
第一阶段	以单芯片为核心的可编程控制器形式的系统，同时具有与监测、指示设备相配合的功能。	主要应用于工业控制领域，一般没有操作系统的支持，通过汇编语言编程对系统进行直接控制。结构和功能都相对单一，处理效率较低，存储容量较小，几乎没有用户接口
第二阶段	以嵌入式 CPU 为基础、以简单操作系统为核心的嵌入式系统	系统主要用来控制系统负载等。通用性比较弱；系统开销小，效率有所提高；操作系统具有一定的兼容性和扩展性；应用软件较专业，用户界面不够友好；
第三阶段	嵌入式操作系统为标志的嵌入式系统	嵌入式应用软件丰富。嵌入式操作系统跨平台性、兼容性、扩展性好；具备文件和目录管理等功能；应用程序接口丰富，开发应用程序简单；
第四阶段	以基于 Internet 为标志的嵌入式系统	网络家电，PDA 等。是 Internet 技术与各行业技术发展的结合，代表着嵌入式技术的真正未来。

Windows CE 允许嵌入式应用软件在公开接口上产生产品数据的功能将激起一股控制和 HMI 产品涌入市场的潮流，提供您需要的操作系统和开发工具以创建高级嵌入式应用程序。

Windows CE Platform Builder 改变了嵌入式软件工具获得和使用的方式，大幅降低了嵌入式产品开发的成本。Windows CE 和 Platform Builder 不仅仅是改变了嵌入式系统开发的经济模型，它们还改变了评估模型。Windows Platform Builder 可通过零售渠道获得，在价格上与其他同类型嵌入式开发工具比起来也具有相当优势。微软的开发工具在结合高性能和突出的价值方面一向做得非常好，Platform Builder 也不例外。

当今的嵌入式设计队伍处于高度分化状态中。有着多得让人眼花缭乱的实现目标和工具，包括微控制器、微处理器、定制和专用的操作系统、实时操作系统附件和内核，以及非标准化的嵌入式开发系统和重要的工具。目标和工具上的多样化也困扰着嵌入式系统的主要购买者，使得支持所有平台的环境的维护变得几乎不可能。这一障碍导致制造商和主要销售团体坚持嵌入式系统开发者必须遵守标准平台和开发工具的原则。作为嵌入式平台的 Windows CE 满足甚至超过了这些需求。Windows CE 和 Platform Builder 的设计实现满足了嵌入式系统设计者的需求，是过去那些昂贵的专用嵌入式开发工具的亟需的替代方案^[4]。Windows CE 操作系统及相关开发工具之所以具有吸引力，其大部

分原因在于他们提供了广为人知并广为使用的开发环境，并且和具有嵌入式应用所需的高性能、高效率和便携式操作系统结合在一起。Windows CE 的高级特性，诸如网络、通信以及图形功能，和模块化设计结合在一起，为中级开发人员创建高复杂度嵌入式系统提供了理想的环境。

现有技术并不能开发附带新型 I/O 系统的扩展嵌入式系统。Window CE 提供象 USB 这样的外围接口，允许高级 I/O 系统的即时支持。Windows CE 驱动程序模型在驱动程序体系结构方面有着很大的灵活性，允许现有驱动程序技术的快速引入。对嵌入式市场的这一新的开放性也是 Windows CE 最具希望的所在。

现今的嵌入式系统并不接受开放数据接口。与现在的嵌入式系统关联的进程数据依靠一个捆绑于系统上的单独的监控计算机收集获得。Windows CE 的网络和通信功能使嵌入式系统的开发者和用户能够使这些开放接口对嵌入式环境本地化。这一“基础架构开放性”也是 Windows CE 在嵌入式市场获得空前欢迎的主要原因之一。

Windows CE 的基本设计目标之一是简化嵌入式开发过程，这一目标的实现借助于当今嵌入式系统设计者在编程时最为广泛使用的 Win32 API 的引入，并和 Microsoft Visual C++ 等开发系统结合在一起。同时，Windows CE 提供一个稳定、灵活且被广泛支持的操作系统来处理建立在嵌入式系统上的多种多样的硬件平台和软件应用程序。对嵌入式系统的开发者和用户来说，Windows CE 带来的实实在在的好处是无可否认的。

1.2 车载导航系统简介

车载导航系统是一种车载电子设备，它除了能够显示电子地图和确定自身位置外，还能够进行信息查询和规划到达目的地的最优路径，并能引导车辆驾驶者到达目的地，从而提高道路通行效率。

1.2.1 车载导航系统的发展历程

车载导航系统的研究是从本世纪 70 年代末开始研究的。80 年代初诞生了 VNS (Vehicle Navigation System)，这类系统一经问世，就以其极大的应用价值，广阔的市场前景引起了各国政府、研究机关、企业、院校的广泛注意，成为交通、车辆、电子等相关专业的热门研究领域。目前，日本、欧美等许多发达国家都纷纷推出了自己的发展规划，越来越多的厂商加入到导航系统的研究中。与之相比，我国起步较晚^[3]。

最早的包含有电子元件的车载导航仪器是美国研制的车辆记程表。这种仪器用磁罗盘测量车辆的航向，用轮速测量车辆行驶过的距离，再用机械计算机解算出车辆的位置。

分原因在于他们提供了广为人知并广为使用的开发环境，并且和具有嵌入式应用所需的高性能、高效率和便携式操作系统结合在一起。Windows CE 的高级特性，诸如网络、通信以及图形功能，和模块化设计结合在一起，为中级开发人员创建高复杂度嵌入式系统提供了理想的环境。

现有技术并不能开发附带新型 I/O 系统的扩展嵌入式系统。Window CE 提供象 USB 这样的外围接口，允许高级 I/O 系统的即时支持。Windows CE 驱动程序模型在驱动程序体系结构方面有着很大的灵活性，允许现有驱动程序技术的快速引入。对嵌入式市场的这一新的开放性也是 Windows CE 最具希望的所在。

现今的嵌入式系统并不接受开放数据接口。与现在的嵌入式系统关联的进程数据依靠一个捆绑于系统上的单独的监控计算机收集获得。Windows CE 的网络和通信功能使嵌入式系统的开发者和用户能够使这些开放接口对嵌入式环境本地化。这一“基础架构开放性”也是 Windows CE 在嵌入式市场获得空前欢迎的主要原因之一。

Windows CE 的基本设计目标之一是简化嵌入式开发过程，这一目标的实现借助于当今嵌入式系统设计者在编程时最为广泛使用的 Win32 API 的引入，并和 Microsoft Visual C++ 等开发系统结合在一起。同时，Windows CE 提供一个稳定、灵活且被广泛支持的操作系统来处理建立在嵌入式系统上的多种多样的硬件平台和软件应用程序。对嵌入式系统的开发者和用户来说，Windows CE 带来的实实在在的好处是无可否认的。

1.2 车载导航系统简介

车载导航系统是一种车载电子设备，它除了能够显示电子地图和确定自身位置外，还能够进行信息查询和规划到达目的地的最优路径，并能引导车辆驾驶者到达目的地，从而提高道路通行效率。

1.2.1 车载导航系统的发展历程

车载导航系统的研究是从本世纪 70 年代末开始研究的。80 年代初诞生了 VNS (Vehicle Navigation System)，这类系统一经问世，就以其极大的应用价值，广阔的市场前景引起了各国政府、研究机关、企业、院校的广泛注意，成为交通、车辆、电子等相关专业的热门研究领域。目前，日本、欧美等许多发达国家都纷纷推出了自己的发展规划，越来越多的厂商加入到导航系统的研究中。与之相比，我国起步较晚^[5]。

最早的包含有电子元件的车载导航仪器是美国研制的车辆记程表。这种仪器用磁罗盘测量车辆的航向，用轮速测量车辆行驶过的距离，再用机械计算机解算出车辆的位置。盘测量车辆的航向，用轮速测量车辆行驶过的距离，再用机械计算机解算出车辆的位置。

分原因在于他们提供了广为人知并广为使用的开发环境，并且和具有嵌入式应用所需的高性能、高效率和便携式操作系统结合在一起。Windows CE 的高级特性，诸如网络、通信以及图形功能，和模块化设计结合在一起，为中级开发人员创建高复杂度嵌入式系统提供了理想的环境。

现有技术并不能开发附带新型 I/O 系统的扩展嵌入式系统。Window CE 提供象 USB 这样的外围接口，允许高级 I/O 系统的即时支持。Windows CE 驱动程序模型在驱动程序体系结构方面有着很大的灵活性，允许现有驱动程序技术的快速引入。对嵌入式市场的这一新的开放性也是 Windows CE 最具希望的所在。

现今的嵌入式系统并不接受开放数据接口。与现在的嵌入式系统关联的进程数据依靠一个捆绑于系统上的单独的监控计算机收集获得。Windows CE 的网络和通信功能使嵌入式系统的开发者和用户能够使这些开放接口对嵌入式环境本地化。这一“基础架构开放性”也是 Windows CE 在嵌入式市场获得空前欢迎的主要原因之一。

Windows CE 的基本设计目标之一是简化嵌入式开发过程，这一目标的实现借助于当今嵌入式系统设计者在编程时最为广泛使用的 Win32 API 的引入，并和 Microsoft Visual C++ 等开发系统结合在一起。同时，Windows CE 提供一个稳定、灵活且被广泛支持的操作系统来处理建立在嵌入式系统上的多种多样的硬件平台和软件应用程序。对嵌入式系统的开发者和用户来说，Windows CE 带来的实实在在的好处是无可否认的。

1.2 车载导航系统简介

车载导航系统是一种车载电子设备，它除了能够显示电子地图和确定自身位置外，还能够进行信息查询和规划到达目的地的最优路径，并能引导车辆驾驶者到达目的地，从而提高道路通行效率。

1.2.1 车载导航系统的发展历程

车载导航系统的研究是从本世纪 70 年代末开始研究的。80 年代初诞生了 VNS (Vehicle Navigation System)，这类系统一经问世，就以其极大的应用价值，广阔的市场前景引起了各国政府、研究机关、企业、院校的广泛注意，成为交通、车辆、电子等相关专业的热门研究领域。目前，日本、欧美等许多发达国家都纷纷推出了自己的发展规划，越来越多的厂商加入到导航系统的研究中。与之相比，我国起步较晚^[3]。

最早的包含有电子元件的车载导航仪器是美国研制的车辆记程表。这种仪器用磁罗盘测量车辆的航向，用轮速测量车辆行驶过的距离，再用机械计算机解算出车辆的位置。

六十年代末期，美国公路局研制的电子线路引导系统（ERGS）则采用了另外一种技术体制，它在关键地点的路面下埋有环状感性天线，车辆经过时即可知道自己的瞬时位置。1993 年在佛罗里达的奥兰多进行的 ITS 试验，所用的车载导航设备是 TRAVETEK，TRAVETEK 用了推算导航、地图匹配和 GPS。

日本于 1973 年在政府资助下开始了为期 6 年的 ITS 路线引导计划，所采用的技术方案是埋在地下的电感环行天线，只是试验的规模比当年的电子路线引导系统大的多。1981 年有许多公司在国内市场上推出了第一代车载导航系统，这些系统放弃了依赖埋地天线的方法而采用了车辆独立定位的方案。它们都以推算导航作为导航器，能给出用户的大概位置和路线。

在欧洲，主要是德国，车载导航的起步比日本晚些。七十年代末由 BOSCH 公司和大众汽车公司联合研制出 ALI 路线引导系统，它采用的技术与 ERGS 类似，即在车辆走进时检测车载设备并与之进行通信。BOSCH 公司于 1983 年演示了车辆独立导航与引导系统 EVA，该系统使用推算航位/地图匹配技术，1989 年与美国 ETAK 公司合作研制成了 Travel pilot，于 1991 年推入市场。

特别是 1994 年，美国 GPS 系统投入全面运行后，世界上使用 GPS 的车载导航系统逐渐成为市场主流。

1.2.2 车载导航系统的扩展

车载导航系统已经在发达国家得到较广泛的应用。日本最早从 1992 年开始大规模应用 GPS 导航系统，从 1996 年开始车载导航系统进入快速发展期，截止到 2002 年 6 月已累计销售达到 962 万台。目前年销售量维持在几百万套，超过 60%的新车出厂时就已安装了车载导航系统。在欧洲，2002 年汽车智能导航产品的销售量为 200 万台；在美国，预计 2005 年汽车智能导航产品的年销售量将超过 150 万台，到 2010 年将达到 1100 万辆。在我国，车载导航系统的产品也已经开始上市^[3]。

当今先进的车辆导航系统将 GPS、惯性导航元件、TV、CD、AM/FM、视频影碟、通讯终端等装置和功能集成在一起，除了具备传统导航功能以外，还可以通过导航设备控制空调、电话、TV 等设备，甚至可以进行 Internet 信息的访问，导航设备已经逐渐的开始发挥车载信息平台的作用。

原有导航系统中导航技术主要集中于向驾驶者提供可供选择的经路并提示前方可能出现的道路信息，包括交叉路口、高速设施、铁路道口等，信息的主要来源是对既有

电子地图数据的加工以及使用 GPS、陀螺仪感度等信息进行的车辆位置定位。

在此基础上, 得到进一步发展的引导功能可以根据 FM、光波等媒介以固定的格式向车辆发送当前的路况信息, 包括阻塞度、突发事件等。

在日本, VICS (Vehicle Information and Communication System) 已经作为全国性质的路况信息机构得到充分的利用。所谓 VICS 是指以提高道路交通的安全性为目的构筑的道路交通信息提供 System。

被提供的道路交通信息有以下几项:

- (1) 道路上的阻塞状况、通过某一条或某一段道路所需要的时间;
- (2) 速度限制及禁止通行等的规制信息;
- (3) 从车辆当前位置到行进方向上的城市所要的时间等。

这些信息由道路管理者、或各级交通警察收集然后集中于 VICS Center。在这里各类信息被转化成统一的格式再通过各种媒介提供给驾驶者。能够用来进行 VICS 信息传输的媒介包括 FM、电波、和光波 3 种。

VICS 提供的信息又分为文字表示型、简易图形表示型和地图表示型 3 个类型。

1.2.3 驾驶员支援辅助功能的现状

无论是传统的导航功能(地图显示、经路计算、经路引导等)还是后期发展的扩充功能(多媒体、车载设备管理、VICS 信息使用等), 这些功能都是在已有信息数据的进行处理和优化, 向驾驶者提供辅助信息。

但由于这些信息的制作是建立在固定的电子地图数据的基础上, 因此从实时的角度上考虑, 远远不能满足实际驾驶过程中千变万化的各种随机事件的发生, 而这些随机事件却是驾驶者在驾驶过程中需要密切注意的。

为了能够在行驶过程向驾驶者提供更加实时的信息, 各种驾驶者支援辅助功能应运而生, 并成为各大汽车导航厂商投入大量资源进行研究开发的热点技术。

交叉点支援技术作为驾驶者支援辅助技术的核心内容, 已经被业内人士看成是支持车的安全的主要技术之一。

目前驾驶者支援辅助功能主要包括车辆周边状况的提示以及和其他车辆的信息交互两个大的方面^[6]。

车辆周边状况提示功能中目前投入研究较多的是交叉点通过支援和车线变更支援。交叉点通过支援是针对车辆在经过交叉点附近时车辆速度变更比较大, 行驶方向通常也

会发生较大变化而设计的。而车线变更支援是在驾驶者变更车线之前提示相邻车线上的车辆行驶状况。

车间通信功能^[7]是通过雷达、电话以及无线网络作为载体，在车与车之间彼此交互当前的位置状况等信息，可以在导航地图上同时显示两辆或多辆车的位置。在生活中经常会出现这样的情况，朋友之间事先约定到同一个地方集合，但经常因为无法清楚告知对方自己当前的位置而烦恼，而车间通信功能很好地解决了这样的问题，使得不同车辆之间的彼此了解变得非常方便。

1.3 论文课题的提出

每天有数以百万计的车辆在道路上行驶，各种交通事故频频发生，其中相当一部分事故在车与车的相互作用下发生，而下述两种情况能够占到此类事故的半数以上，他们分别是追尾事故和撞头（迎头相撞）事故。

在前面提到的驾驶员支援辅助功能中，车辆周边状况提示功能就是针对这些状况提出的。目前，已经产生了用以解决追尾事故的 CMS 技术（Collision Mitigation Brake System），但对于撞头事故，还没有成熟的技术能够很好地解决此类问题。

因此，我选择了这一技术的研究作为我的硕士毕业论文设计。

1.4 论文的组织

本文通过对目前车载导航系统的分析，提出了一个通过分析路线前方的道路信息，结合车辆运行状态相驾驶者提供警告信息以减少撞头事故发生的技术方案，并提出了系统设计，接着详细论述讨论了其实现。全文共分八章：

第一章综述了嵌入式操作系统 WinCE 平台的背景，以及车载导航系统的简介及其发展。并概述了车载导航系统中的交叉点支援子系统的课题的提出。

第二章主要介绍了开发该系统所必须使用的相应软硬件系统的关键技术情况，主要说明了图像识别技术，基于 CANBus 总线的数据传输技术，电子地图数据的存储和应用技术以及当前导航系统中的基本功能和实现。

第三章对该系统的进行了分析和设计，包括可行性分析，主要是结构和技术方案的选择，最后提出了系统结构。

第四章详细论述了子系统中数据加工及对象交叉点抽出的算法设计与实现。

第五章详细论述了子系统中 CANBus 总线数据传输协议的定制与实现。

第六章详细论述了警告语音与警告图像的设计与实现。

第七章展示了针对该研究课题制作的一个调试用的可视化工具。

最后是论文中涉及的关键技术文献的一个资源索引。

结尾表达了对我研究生毕业设计和生活中给予我帮助和支持的人的感谢。

第二章 与实现交叉点支援系统相关的若干关键技术

交叉点支援系统作为一项研究课题，有一些功能还没有能够加载到整体导航环境中，因此需要导航系统和 PC 之间协同完成课题的研究。导航设备上完成和电子地图数据相关的功能以及危险警告信息的输出、PC 机上进行图像识别相关功能的开发，导航设备和 PC 之间使用 CANBus 进行连接和数据通信。

在本章中针对本系统将会用到或者需要考虑的关键技术及其背景知识进行介绍，为下一章的方案设计与分析打下基础。

2.1 图像识别技术

图像识别的过程就是图形刺激作用于感觉器官，人们辨认出它是经验过的某一图形的过程，也称图像再认。在图像识别中，既要有当时进入感官的信息，也要有记忆中存储的信息。只有通过存储的信息与当前的信息进行比较的加工过程，才能实现对图像的再认^[8]。

图像识别是人工智能的一个重要领域。为了编制模拟人类图像识别活动的计算机程序，人们提出了不同的图像识别模型。模板匹配模型认为，识别某个图像，必须在过去的经验中有这个图像的记忆模式，又叫模板^[9]。当前的刺激如果能与大脑中的模板相匹配，这个图像也就被识别了。原型匹配模型则认为，在长时记忆中存储的并不是所要识别的无数个模板，而是图像的某些“相似性”。从图像中抽象出来的“相似性”就可作为原型，拿它来检验所要识别的图像。如果能找到一个相似的原型，这个图像也就被识别了。泛模识别模型把图像识别过程分为不同的层次，每一层次都有承担不同职责的特征分析机制，它们依次进行工作，最终完成对图像的识别。

图像识别系统包括图像输入、图像的预处理、特征的提取和图像的解释（识别）等技术环节和设备^[10]。其关键技术或难点在于实现系统能在一种略加控制的环境下，针对移动目标实时地运行，这些目标通过静止摄像机可能会产生大小不同、角度不同及光照效果不同的图像。

这就要求系统采用适当的图像输入方法和预处理技术，以保证图像特征有效地提取或模板的生成。图像识别（分类、解释）的方法基本上分为统计方法和结构分析两类，前者是以数学决策理论为基础，建立统计学的识别模型，指纹、掌形的识别多采用这种

方法，其特点是稳定、但很少利用图像本身的结构关系。后者则主要是分析图像的结构，它充分地发挥了图像的特点，但容易受图像生成过程中噪声干扰的影响。

图像识别技术的应用可分为验证和识别两种方式。验证的目的是把当事人的身份与正在发生的行为联系在一起，确认行为的合法性，通常是验证“你是他？”的一对一（或较少的量）的比对系统。识别则是对系统的输入图像（可能是摄像机拍摄的活体图像）与存储在数据库中的大量的参考图像进行比对，来确定输入图像（目标）的身份，所以也称为是识别“你是谁？”的一对多的比对系统。验证系统因可对图像的输入加以更多的控制，系统的可靠性和稳定性好，也相对成熟，已广泛地应用于出入管理系统中。识别系统则因环境条件的限制，还没有成熟的产品。但其应用的效果及在安全防范中的作用已被人们认识和肯定，是研究的重点课题，其主要方向是：

高质量图像输入子系统，保证在各种环境条件下能采集到足够分辨率、适当方位和灰度变化的图像，它涉及图像传感器（如摄像机）的选择、安装方式、与入侵探测的关联及相关的数字化操作^[11]。

基于不同平台（PC、服务器或多处理器、网络的）的处理硬件，能提供系统所需的处理能力、运算速度、灵活性和信息存储能力。

图像处理和解释与分析软件的开发，综合自动模式识别和计算机视觉技术，开始开发各种实用的软件。如输入图像的分割、定位、轮廓提取和面像图像的加光技术；基于人面重心模板的实时面像检测；器官的位置信息和面像特征的提取以及各种比对算法等。

2.2 CANBus 数据总线技术

2.2.1 CAN 总线简介

随着车用电气设备越来越多，从发动机控制到传动系统控制，从行驶、制动、转向系统控制到安全保证系统及仪表报警系统，从电源管理到为提高舒适性而作的各种努力，使汽车电气系统形成一个复杂的大系统，而且这一系统都集中在驾驶室控制。另外，随着近年来 ITS 的发展，以 3G（GPS、GIS 和 GSM）为代表的新型电子通信产品的出现，对汽车的综合布线和信息的共享交互提出了更高的要求。

从布线角度分析，传统的电气系统大多采用点对点的单一通信方式，相互之间少有联系，这样必然需要庞大的布线系统。从而加剧了粗大的线束与汽车有限的可用空间之间的矛盾。无论从材料成本还是工作效率看，传统布线方法都将不能适应汽车的发展。

从信息共享角度分析,为了满足各子系统的实时性要求,有必要对汽车公共数据实行共享,如发动机转速、车轮转速、油门踏板位置等,但每个控制单元对实时性的要求是因数据的更新速率和控制周期不同而不同的。这就要求其数据交换网是基于优先权竞争的模式,且本身具有较高的通信速率。CAN 总线正是为满足这些要求而设计的。

控制器局域网 CAN(Controller Area Network)^[12]是由 Bosch 公司在 20 世纪 80 年代初开发的一种串行多主站总线通讯协议。它具有高传输速率,高抗电磁干扰性,并且能够检测出发生的几乎任何错误。由于其卓越性能,CAN 已广泛应用于交通工具、工业自动化、航天、医疗仪器以及建筑、环境控制等众多领域。CAN 技术在我国也正迅速普及推广。

使用 CAN 通信技术组成汽车内部网络,可满足汽车各 ECU(Electronic Control Unit,电子控制单元)对控制和数据通信的需要^[13]。CAN 总线是一种现场总线,它是为解决现代汽车中 ECU 之间的数据交换而开发的,可用来连接汽车和工业应用中的各种电子控制模块。任何一个需要稳定、可靠的低成本网络系统或设备,都有可能成为 CAN 节点。

2.2.2 CAN 总线的技术特点

CAN 总线可有效支持分布式控制或实时控制,通信介质可以是双绞线,同轴电缆或光纤,其主要特点是^[14]:

- (1) CAN 总线为多主站总线,各节点可在任意时刻向网络上的其他节点发送信息,不分主从,通信灵活;
- (2) CAN 总线采用独特的非破坏性总线仲裁技术,优先级高的节点优先传送数据,可满足实时性要求;
- (3) CAN 总线具有点对点、一点对多点及全局广播传送数据的功能;
- (4) CAN 总线采用短帧结构,每帧有效字节数最多为 8 个,数据传输时间短,并有 CRC 及其他校验措施,数据出错率极低;
- (5) CAN 总线上某一节点出现严重错误时,可自动脱离总线,而总线上的其他操作不受影响;
- (6) CAN 总线系统扩充时,可直接将新节点挂在总线上,因而走线少,系统扩充容易,改型灵活;
- (7) CAN 总线最大传输速率可达 1Mb/s (此时通信距离最长为 40m),直接通信距

离最远可达 10km (速率 5kbps 以下);

(8) CAN 总线上的节点数主要取决于总线驱动电路。在标准帧 (11 位报文标识符) 可达 110 个, 而在扩展帧 (29 位报文标识符) 其个数几乎不受限制。

2.2.3 CAN 总线技术在汽车中的应用

CAN 总线技术在汽车中的应用具有以下优势^[15]:

(1) 信息共享

采用 CAN 总线技术可以实现各 ECU 之间的信息共享, 减少不必要的线束和传感器。例如具有 CAN 总线接口的电喷发动机, 其它电器可共享其提供的转速、水温、机油压力、机油温度、油量瞬时流速等, 这样一方面可省去额外的水温、油压、油温传感器, 另一方面可以将这些数据显示在仪表上, 便于司机检查发动机运行工况, 从而便于发动机的保养维护。

(2) 减少线束

新型电子通讯产品的出现对汽车的综合布线和信息的共享交互提出了更高的要求, 传统的电气系统大多采用点对点的单一通信方式, 相互之间少有联系, 这样必然造成庞大的布线系统。CAN 总线可有效减少线束, 节省空间。例如车门、后视镜、摇窗机、门锁控制等的传统布线需要 20-30 根, 应用总线 CAN 则只需要 2 根。

(3) 关联控制

在一定事故下, 需要对各 ECU 进行关联控制, 而这是传统汽车控制方法难以完成的。CAN 总线技术可以实现多 ECU 的实时关联控制。在发生碰撞事故时, 汽车上的多个气囊可通过 CAN 协调工作, 它们通过传感器感受碰撞信号, 通过 CAN 总线将传感器信号传送到一个中央处理器内, 控制各安全气囊的启动弹出动作。

利用 CAN 总线构建一个车内网络, 需要解决的关键技术问题有^[16]:

- (1) 总线传输信息的速率、容量、优先等级、节点容量等技术问题;
- (2) 高电磁干扰环境下的可靠数据传输;
- (3) 确定最大传输时的延时大小;
- (4) 网络的容错技术;
- (5) 网络的监控和故障诊断功能。

通过 CAN 技术的使用, 可使汽车更加安全、舒适、可靠, 同时可以更简单、迅速地实现车辆的在线编程实现诊断功能。CAN 总线技术在我国汽车中将很快得到普及,

它是汽车发展的一个趋势^[17]。

2.3 电子地图数据存储技术

汽车导航系统是使用于汽车这一特定移动环境下的一种移动计算机产品，它在车辆的行驶中为驾驶者提供各种驾驶信息和帮助。这套系统的基石之一就是支撑着该系统的地理信息系统——电子地图。

从 80 年代中期第一部电子地图问世以来，电子地图的开发研制与应用都取得了长足的进展。

2.3.1 电子地图信息系统的特点

今天的电子地图已经是集存储、阅览、分析基于决策支持为一体的综合电子地图信息系统^[18]。这种信息系统主要有以下特点：

- (1) 图形、文字、声音和数字多媒体集成。把图形的直观性、数字的准确性、声音的引导性和亲切感相结合，充分调动读者的感观。
- (2) 查询检索。电子地图首先是一个图形信息库，能够支持从地图图形到属性数据库和从属性数据到定位图形双向检索的功能。
- (3) 决策支持。电子地图为某种决策提供数据组织上的支持。
- (4) 动态变化支持。电子地图要支持地图的动态显示，直至多维图形模拟，呈现千变万化的感受效果。
- (5) 多级比例变化。支持多级比例尺的地图信息存储和显示。
- (6) 客观自然状况的模拟支持。为使用户更好地融入系统，模拟客观自然状况会产生良好效果。

2.3.2 信息的存储方式

由于地域的广阔、地理信息量的巨大，将所有信息放在一个存储逻辑单元里面是不明智的做法。信息分“块”存储将带来管理、查询上的方便和快速。在每一个“块”里，所有应该为它拥有的客观地理信息都应存在并有序存放。

适合于人们全局和局部的思考方式和考虑查询上的方便快速，“级”的概念也就应运而生。“级”是比例尺的概念，但“级”的概念包涵了各比例尺地理信息间的逻辑关系。

由“块”和“级”而形成了如图 2.1 所示的地理信息存储方式。

在图 2.1 中，每一个方格就是存储的一个单位，我们称为“图叶”。最高层的图叶是

最主要的地理信息，数据量最小，但它有众多的子图叶和孙子图叶，为它补充记述所有应该有的信息。

需要注意的一点是，在高层的图叶中记述的所有地理信息在其所有子图叶和孙子图叶中都必须重新记述，或等价表示。即地域关系上，父图叶包含子图叶；地理信息上，子图叶包含父图叶。

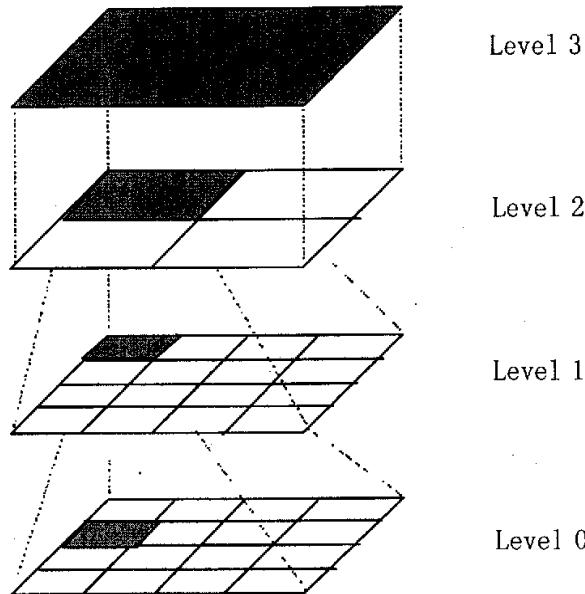


图 2.1 地理信息存储方式
Fig. 2.1 Storage Method of Geographical Information

2.3.3 索引的建立

前面描述的树状存储结构虽然有利于信息的查询，但随着级别较低的图叶数量的急剧膨胀，级别较低图叶的查询变得十分费时。因此，建立一个具有较高查询效率的索引结构非常必要。

地理信息采用了分级、分块的存储方式，相应的就应该提供一种可以对某级某块地理住处进行直接、间接查找的途径。这里采用了一种非文件系统的多级索引表的方式，建立一个紧凑的、利用相对偏移量表示某图叶位置的多级索引，其方式为：

- (1) 以某一“级”作为基准，向上、向下分别索引；
- (2) 对于子图叶或孙子图叶中，图叶数目较多的“级”采取默认图叶存在的手段用图叶的大小作为索引到图叶真实数据的途径。

利用这种多级索引建立起来的索引结构将有效地控制索引的大小，从而保证了较高的查询效率。

2.3.4 地理信息的分类存储

即使对于某一特定作用, 地理信息也是多种多样的。为了在一次存取中获得尽可能集中的一类信息, 对地理信息的分类存储是必要的, 对于地理信息的分类存储是以图叶为单位进行的。

也就是说, 对一个图叶的信息是分 Layer 存储的, 各 layer 中数据记录的数据种类是不同的。

layer 可以看作是一张透明的纸, 在上面写入以用途和属性等共同项括出的物体的数据。对于一个图叶, 一般有如下 8 个 layer 中的一个或几个一起来构成这个图叶所包含的地理信息。

1、文字记号 layer

文字记号 layer 是由若干个文字记号数据记录构成的。文字记号数据记录又包括如下信息: 文字列的属性、文字列的大小、文字列的方向、文字列指示的对象的地址、文字描画的基准点、文字列内容等。

2、背景 layer

背景 layer 是为了表示道路、铁路、河流、主要目标物等而设置的 layer, 由若干个背景数据记录构成的。背景数据记录包括如下信息: 背景种类、颜色种类、图形种类、填充模式、要素点数、要素指定点 list 等。

3、道路 layer

道路 layer 是为了 map matching 而提供道路网信息的 layer。道路 layer 的构成包括 node table、交叉点构成 node list、道路 list、相邻 node list 等。

4、服务 layer

服务层是为了表示在预想的显示装置中不可能同时表现出来的标记的种类, 还有与地图相连的各种各样的信息而设置的。为了把这些信息以最小的表示面积在背景上显示出来, 使用了服务符号。

5、扩展道路 layer

向硬件提供导航所必须的道路信息的层, 在这个层中记载有交叉点名称 list, 连接属性 list, 交叉点间连接距离 list, 父交叉点对应 list, 子交叉点对应 list, 连接方位 list 及连接距离 list。

6、信息 layer

存放必须层(文字、背景层)中的记号、建筑物及其它设施的详细信息的可选择层。

7、附属文字记号 layer、附属背景 layer

附属文字记号层、附属背景层是用来补充文字层、背景层的可选择层。它的内容包括了那为了避免画面数据密度太高而没有放入文字层、背景层的文字和背景信息，这些信息于优先度比较低而没有必要总显示在屏幕上。

2.4 导航系统功能及内部结构

本节着重对导航系统所实现的主要功能和内部结构进行介绍。

2.4.1 导航系统的主要功能

车辆导航系统，作为一种车载电子设备，它除了能够显示电子地图和确定自身位置外，还能够进行信息查询和规划达到目的地的最优路径，并能引导车辆驾驶者到达目的地，从而提高道路通行效率。这里，对导航系统的主要功能进行比较详细的介绍。

1、车辆定位功能

车辆定位主要是指能够在电子地图上标出车辆所在的位置以及行驶方向。

算出车辆的现在位置的定位方式有自立航法和电波航法。自立航法是通过安装在车辆上的方位传感器和距离传感器求得行驶轨迹，求得车辆的相对位置。电波航法里，GPS 系统被使用，通过接受由人工卫星发出的电波，求得车辆的绝对位置^[19]。还有，在各自的航法中通过比较地图的道路形状数据和车辆行驶轨迹的形状，提高现在地的推定位置的方法就是地图匹配。

2、地图显示功能

地图显示主要是指能够将实现存储在媒质中的电子地图显示到显示屏上。一般可以进行地图比例尺切换、地图滚动、地图视角切换等操作。

地图显示内容包括背景、道路、文字、地图记号、天空、建筑物、一方通行 Icon、路上停车场、走行轨迹等。

为了便于用户察看不同缩尺下的详细的地图数据，导航系统提供了根据用户的操作进行地图的比例尺变更。同时，为了便于用户对指定地点附近的地图数据进行查询，导航系统提供了 8 方向的地图滚动功能。

为了适应不同用户使用导航系统的方便和习惯，导航系统提供了北向上、车头向上、和 3D 视角显示地图的功能。其中，北向上的地图模式与我们通常见到的地图类似，而车头向上和 3D 视角则给以驾驶者更加真切的视角。

根据导航不同的画面布局，又可以分为：单画面、双画面、MiniMap。在单画面模

式下，给以驾驶者最大的视觉空间；双画面使驾驶者能够在相互参照中进行地图操作；而 MiniMap 的模式则多用于后面提到的检索功能，向用户提供指定地点的地图信息。

3、路径探索功能

路径探索分为初期路径探索，路径再探索，迂回经路探索三种。

当用户指定目的地后，可以为用户计算出最优路径，称之为初期路径探索。当车辆行驶的过程中偏离计算出来的经路的时候，导航机会自动计算一条新的经路供用户参考，称之为路径再探索。如果由于实际路况的原因或驾驶者的个人意愿，想要绕过前方一定距离内的经路，则可以选择迂回经路探索，迂回经路探索会根据原有经路计算出一条新的经路，新的经路将尽最大可能回避前方指定距离的原有经路。

目前，根据对不同道路赋予不同的 Cost 值，可以向用户提供四种不同的经路，分别是高速道路优先、通常道路优先、实际距离优先和道路宽度优先。这样可以给予用户更丰富的选择。经路计算结束后，驾驶者还可以根据自己的需要在不同的计算条件之间进行切换。

此外，驾驶者还可以在经路上设定多个经由地，这样可以在开始的时候将路途中想要去的地方一次全部设定，导航机会计算出按顺序依次到达各经由地的路径，省去了驾驶者需要多次进行目的地设定的麻烦。

4、信息检索功能

为了找到想要的目的地、经由地，导航系统提供了信息检索功能。考虑到用户对想要查询的地点了解的内容不全面，因此导航系统提供了多种方式的检索功能，包括：住所检索、设施检索、电话号码检索、五十音检索等。

导航系统提供了通过省==>市==>区==>道路==>门牌号码，逐级递进的住所检索方式。这个功能主要是为了方便了解具体地点地址的用户检索使用。

导航系统提供了针对不同设施的检索方式，同时在设施检索中可以进行地区的缩进，缩小检索范围，使得用户可以迅速的定位自己想要查询的地点。设施种类进行了详细划分，包括公共设施、交通设施、文化设施等大类，每个大类下再细分为很多个子类别。例如，公共设施包括医院、政府、警察局等；交通设施包括火车站、高速设施等。

电话号码检索就是通过用户输入的电话号码检索出电话号码所在的地点。

五十音是日本文字的注音，类似于中国的拼音的概念。导航系统提供了通过注音进行检索住所和设施的功能，用户只需要输入地名的读音，导航系统就能够找出和读音相符的地点信息。

对于所有的检索结果，导航系统都可以通过 MiniMap 的形式将地点周边的状况显示出来，便于用户确认检索的结果。

5、实时引导功能

在沿着最优路径走行时，会实时的给用户以指引。引导驾驶者安全、快速且放心的到达设定目的地。引导的方式包括语音和图形两个大的类别。

语音提示可以用语音的方式向驾驶者提示行进的方向，使驾驶者不需要看导航画面也能得到想要的信息，减少了驾驶的同时关注导航画面所带来的潜在危险。

图形提示是指在某些特别的地点以扩大图的形式详细显示道路现状。随着数据库存储量的不断增加，很多扩大图是以 Bitmap 的形式进行存储的，与实际的道路状况非常相似，给人以身临其境的感觉。

随着导航功能的不断扩充，向驾驶者提供引导的内容也越来越多。不仅包括简单的提示目的地到达和交叉点的方向，也有对各种道路设施物的引导，包括火车道口、收费站、道路合流。在加入 VICS 实时交通控制系统后，更是可以对实时的路况信息进行引导，包括道路阻塞程度、交通事故等。

2.4.2 导航系统内部结构

导航系统是建立在一组专用硬件平台基础上的嵌入式应用程序，为了实现各种复杂的导航功能，在导航系统内部建立了一套层次分明的总体框架，每一层中又根据其功能特点的不同划分成不同的模块。各模块之间互相联系，各司其职，使得整个导航系统有条不紊的运作。

导航系统中按照彼此的依赖关系自下而上分为 Platform 层，DataServer 层，机能层和 UserInterface 层。各层次在导航系统中所处的位置可以参照图 2.2。

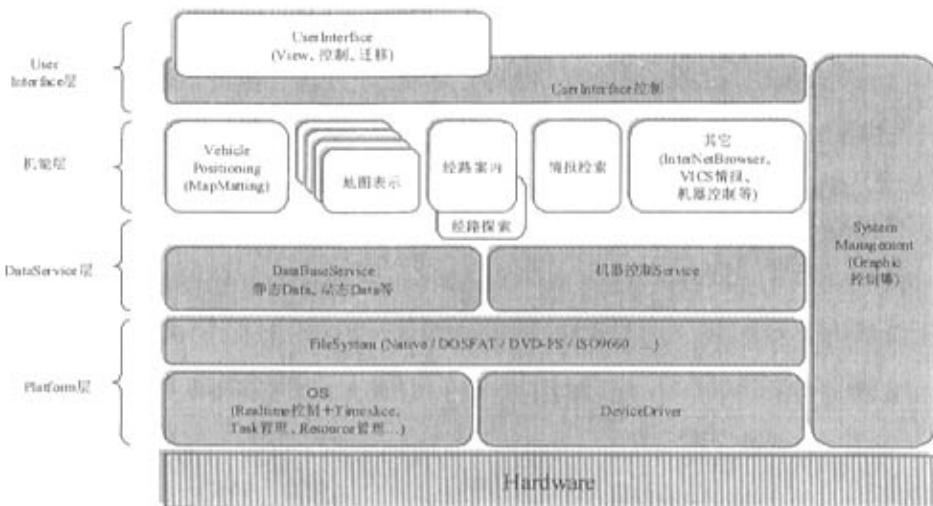


图 2.2 导航系统内部结构
Fig. 2.2 Navigation System Internal Architecture

1、Platform 层

Platform 层是和导航硬件平台直接构成联系的部分。当前的导航系统是采用 WinCE 作为操作系统并在此基础上构建的。Platform 中包含的内容很多，包括

- (1) 任务管理、内存管理等基本的操作系统功能
- (2) 各种外围设备驱动的管理，总线、显示设备、语音设备、GPS 接收设备等。
- (3) 负责从硬件接受各种信号并将其转发给应用程序。
- (4) 负责文件系统的管理，包括磁盘数据的读取和保存、DVD 设备的访问等。

2、机能层

机能层是实现各项导航机能的应用程序层。根据所负责功能的不同，又分为以下几个功能模块。

(1) 车辆定位模块

负责实现车辆定位功能。该模块的输入信息包括从硬件采集到的车速脉冲、GPS 信息、车辆转弯的感度信息和从电子地图数据中得到的道路信息。通过使用滤波算法进行计算并和道路形状数据进行匹配，确认车辆当前的位置。这些匹配的结果将会作为其他各模块功能实现的一个基准条件。

(2) 地图表示模块

该模块主要负责地图的描画。该模块依赖于电子地图数据中存储的道路、背景、文字等信息，同时参照车辆定位模块提供的车辆位置，通过 Platform 将描画出来的地图画

面显示在屏幕上。除了上述电子地图数据存储的信息外，该模块也负责将其他一些内容在地图上标示出来，既包括用户设定的经由地、目的地、标记点、迂回地点以及计算出来的经路，也包括从外界取得的实时路况甚至天气情况等。

(3) 经路引导模块

经路引导模块是导航系统中功能最繁多、结构最复杂，也是最为重要的一个模块。该模块的主要功能是提供探索出来的经路并根据经路和当前车辆的行驶状况给用户提供及时地引导。

经路探索过程使用 A 星算法作为基本算法，并参照电子地图中经路探索相关的数据设定了一整套适合于日本实际道路状况的计算参数，以求在最短的时间内计算出最合理的经路。

引导过程根据所要实现引导功能的不同会使用到不同的数据，例如计算高速道路的收费情况需要参照高速道路数据和收费数据、对道路上各种构造物的警告需要参照引导数据等。

(4) 检索模块

检索模块除了实现各项检索功能外，还负责管理用户设定的各种地点信息，包括目的地设定的履历信息、用户当前设定的经由地信息以及标记点信息等，同时管理和这些地点相关的图像信息。目前的导航系统中，由于采用了硬盘作为存储设备，存储容量大大增加，保存的地点个数可以多达数百个。

(5) 其它模块

除了前面几个导航系统中负责主要功能的模块之外，随着导航功能的扩展，出现了其他一些模块，分别负责不同的功能。例如负责解析 Vics 信息并向其他模块提供路况信息的 Vics 模块、负责通过车载电话获取网络信息的 InterNavi 模块等。还有一些模块，虽然并不独立实现某个导航功能，但能够为其他模块提供各方面的支持，包括负责管理 Icon 和字库信息的 FCS 模块、负责提供坐标变换库函数的 Dalcom 模块和负责管理调色板信息的 DPS 模块等。

3、DataService 层

DataService 层介于 Platform 层和机能层之间，功能包括地图数据的读取、用户数据管理以及对机能层生成的一些数据进行管理。

地图数据读取的模块称为 Loader，负责接受上层各模块发出的各种请求，计算所请求的数据在地图数据中的位置，从文件中读取相应的数据进行操作。此外，Loader 还负

负责 media 状态变更管理。

用户数据管理模块称为 UDS，负责将用户设定的信息保存起来，以保证在导航及重新启动的时候这些信息能够得到恢复。这些信息根据更新频度、数据量大小的不同使用不同的存储介质进行保存，例如标记点信息由于数据量比较大需要保存在硬盘上，而当前的引导状态等数据量很小但需要快速保存的信息则保存在 SRAM 中然后再定期保存到 FlashROM 中。

此外，由于使用经路数据的模块很多，专门设立了用于管理经路数据的 RouteService 模块。该模块建立了一整套负责数据更新和数据保护的机制，保证在多个使用者同时对经路数据进行操作的情况下不会出现问题。

4、UserInterface 层

UserInterface 层就是 HMI，它是导航功能的表示层，其作用就是显示供用户进行操作的可视化界面，并把机能层提供的各种信息反映给用户。

在 HMI 内部，实际上又可以分为 3 个大的部分，一是 AUI，这一层以 Form 形式出现，这部分的主要作用就是实现表示部分的静态描画；二是 AppliControl，主要作用是实现 Form 描画逻辑和画页迁移逻辑的控制部分，在这部分中，可以通过属性的变更控制 AUI 的描画，通过 AUI 提供的接口控制 Form 的加载和显示；三是 HMIManager，为 AppliControl 提供了各种基本的操作，可以实现对事件（EVENT）的路由控制、画页的迁移，并控制各个 AppliControl 之间的调度等。

第三章 系统总体方案设计

交叉点支援系统作为一项新技术，其实现过程需要涉及到多方面知识和技术，内容较为繁杂，而且对于每一部分的具体实现方法都有很多的选择，因而需要仔细地进行方案的选择及方案的论证。本章将会对系统的设计目标、可行性分析、以及系统方案的选择等进行详细地介绍。

3.1 设计目标

3.1.1 总体目标

一个好的系统应该要能满足目标的所有需求，要做到这点，首先需要对目标进行全面而准确的需求分析。一般来讲，需求可以分为两大类：功能性需求和非功能性需求，我们在设计系统时要从这两方面进行考虑。本系统是要应用于车辆导航系统，因而在设计时要考虑其特殊需求。下面将分别进行分析。

3.1.2 功能目标

本系统的功能需求有两方面，一是导航系统功能的需求，二是最终客户需求。经过功能分解，为了完成总体目标，需要将大的目标细化为以下功能目标：

1、导航系统功能的需求

（1）重要道路交叉点的检出

通过对 DB 中的数据进行处理，按照一定规则筛选出需要特别关注的道路交叉点。

（2）道路交叉点信息数据的设计实现

为了能够对道路交叉点的实际情况能够进行合理、有效的分析，需要考虑该交叉点多方面的数据，包括车辆距离道路交叉点的距离、车辆与交叉点的高度差异、道路的曲率等信息。

（3）实现通过 CANBus 总线的数据传输

改造现有系统中的 CANBus 总线 Driver，使之满足道路交叉点信息在导航系统和 PC 之间传输的要求，包括数据包格式的定义和数据传输模式的定制。

（4）输出警告语音并从导航画面上显示警告信息

结合现有导航系统的语音输出功能和 HMI 人机交互界面完成警告语音的输出和警告信息的实时显示。

2、可视化调试工具的实现

(1) 交叉点信息交互的演示

通过对话框等形式实时显示做成的交叉点信息信息和数据传输状况。

(2) 车辆走行位置与交叉点位置在电子地图上的显示

为了能够更方便的了解车辆与被警告交叉点的位置关系，要在电子地图工具上实时显示车辆位置和对象道路交叉点的位置。

3.1.3 非功能性目标

任何一个系统都不但要能够实现既定的功能性目标而且要有较好的非功能性指标，尤其是在嵌入式平台下的系统，更要注意非功能性指标的达成。

由于交叉点支援子系统需要非常及时地给予驾驶者必要的警告信息，因此对性能的要求非常高。主要包括两个方面：

(1) 对象道路交叉点检出和交叉点信息制作的速度

根据功能实现的必要要求，需要在一秒钟之内能够完成对象道路交叉点的检出和信息制作工作。

(2) 保证警告语音和警告信息的及时输出

一旦车辆前方某个交叉点被检测出存在危险，需要在最短的时间内将警告信息通知给驾驶者，否则就失去了功能本身的现实意义。因此，对于本系统来说，要求在检测出危险之后 100ms 之内能够发出警告语音并输出警告信息。

此外，作为整个导航系统的一部分，也需要综合考虑系统其他各项资源的使用状况，包括内存、CPU 占有率等。

3.2 可行性分析

下面从市场的角度、技术的成熟度以及系统运行环境等方面进行可行性分析。

3.2.1 市场可行性

车载导航的应用极大地改变人们的出行和生活方式，使行驶中的汽车不再是道路上的孤立物体，因而很快就受到广大用户的重视和欢迎，在全球发达国家的应用普及极快。1993 年，全世界车载导航装置销售量只有 13 万套，1997 年则上升到 200 万套，2000 年超出 400 万套。车载导航装置不仅仅是高档豪华轿车的象征和专用品，而且已扩展到大客车、出租车甚至载货汽车，就连经济型车辆也用上车载导航装置。车载导航系统由于能与无线手机等联用，将成为汽车上不可缺少的重要信息设备。据最新的日本市场调

查显示,日本市场对车载导航装置的需求2002年达到了266万套,2007年将达到378万套。估计到2005年,欧洲总销售量可达250万套,美国总销售量将超过千万套的水平。

由上述分析可知,目前车载导航系统能够得广泛应用,具有广阔的市场前景,很值得大力发展。

存在广阔的市场的同时,我们也必须看到目前各国(尤其是日本)的导航系统厂商都在不断努力丰富导航功能并开发新技术以占领更多的市场,因此,本子系统的开发和实现必将使采用该技术的导航系统具有更大的吸引力,并使之成为消费者竞相使用的汽车电子产品之一。

3.2.2 技术可行性

关于软件部分,经调查结论如下:

(1) 目前日本及很多欧美国家的电子地图数据,无论从存储格式还是数据本身在经过二十几年的不断完善和发展后,已经比较成熟,可以满足对象交叉点制作的需要。

(2) 图像识别技术虽然仍处于发展过程当中,但目前已经出现了一些可行的算法实现可供选择。

(3) 关于开发环境,准备采用面向过程的设计及编程方法,可以使整个系统具备良好的层次性和结构。由于面向WindowsCE,选择了编程工具EVC。

由上述分析可知,本系统在软件和硬件部分都具有很好的可实施性,具备产品开发的基础。

3.2.3 系统运行环境

(1) 硬件平台

PC,导航机(CPU400MHz,内存64M,ROM32M)。

(2) 操作系统

Windows2000,WindowsCE。

(3) 其它支撑软件

VC++6.0, DirectX 8.1, Platform Builder4.2。

3.3 系统总体结构

交叉点支援功能是在原有汽车导航基础上进行开发和研究的,因此,该子系统必须能够和原有系统结构及功能有机的结合起来,在不影响原有系统功能的基础上对应新的

需求。

3.3.1 交叉点支援子系统的构成

作为一项新技术的研究，还没有在实际的系统中采用，其中涉及到的一些子功能在导航设备中还没有加入，例如图像识别相关的内容。因此，本次研究作为课题研究的一种手段，采用的是 PC 和导航设备共同工作，通过 CANBus 总线完成数据传输的方式来¹进行的。

交叉点支援系统的系统构成如图 3.1，整个机能的实现需要包括几个部分：

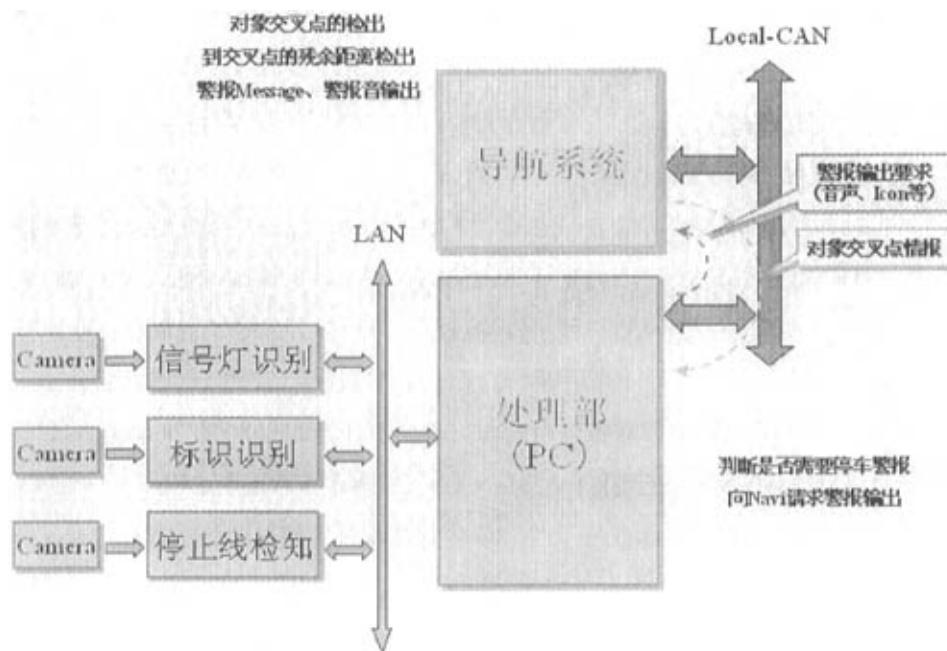


图 3.1 交叉点支援子系统的构成
Fig. 3.1 Structure of Driver Support Subsystem

首先，导航系统通过对 DB 数据的解析，检出认为可能出现危险状况的对象交叉点，并制作该交叉点的相关信息，通过连接导航机和 PC 的 CANBus 总线将这些信息按照一定的格式和时间间隔传输到 PC 机上。

PC 处理部根据分布安装在车辆各个部分的摄像设备取得的实际图像，利用已有的识别软件，收集有助于识别危险状况的各项实际状况，包括前方交叉点信号灯的状态、前方交叉点是否有停止线标志以及其他需要提醒驾驶者注意的其他标志物。

根据收集到的实际路况，结合导航机提供的交叉点信息，以及当前的车辆运行状况，按照一定的算法，确定当前危险存在的可能性。

一旦判断为存在相当的危险性，就通过 Bus 总线向导航设备发送警报通知，导航设备接到警报通知后在显示设备上显示警告信息并以语音的形式提醒驾驶者。

在本课题研究过程中，PC 处理中关于影像识别的部分以及危险度判断的部分还没有开发完成，不作为本论文阐述的重点。相关的功能可以使用调试工具加以模拟，不影响导航设备部分的功能实现。因此，后面各章节会着重阐述导航系统中实现的功能以及导航设备和 PC 间的数据通信，待到 PC 处理中的相关功能开发完成之后，交叉点支援子系统的就可以全部实现了。

3.3.2 交叉点支援子系统在导航系统中的功能实现

导航系统在整体功能的实现过程中，需要完成的功能包括对象交叉点的检出、交叉点信息制作、CANBus 数据传输格式定制、警告信息输出等功能。为了实现这些功能，需要对原有的导航系统进行部分的改造，使之符合功能实现的要求。

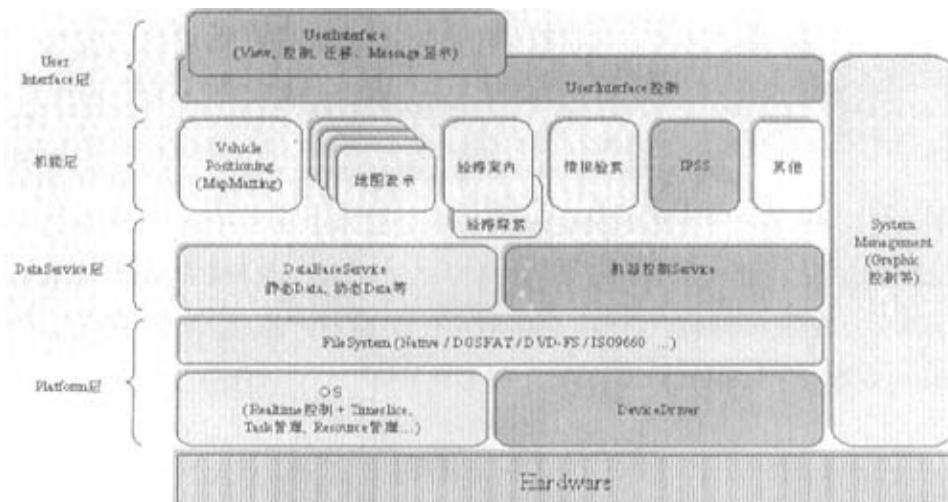


图 3.2 交叉点支援子系统在导航系统中的位置
Fig. 3.2 Position of Driver Support Subsystem in Navigation System

图 3.2 中白色和浅色的部分是导航系统的原有功能模块，深色的部分在本功能的实现过程中进行了部分或全部的改造。

IPSS 模块是完全新增加的模块，主要完成交叉点检出和交叉点信息制作部分的功能。

UserInterface 完成警告图像和警告 Message 的显示。

机器控制 Service 部分中利用 Platform 中提供的语音接口，建立了一个语音请求的管理机制。

DeviceDriver 中对 CANBus 的访问进行了重新封装, 使之能够更加适应当前所要求的传输频率和传输方式。

3.3.3 交叉点支援子系统的模块构成

在导航系统内部, 所有和交叉点支援子系统相关的各个模块联合在一起, 互相作用, 形成一个统一的整体, 共同完成相关功能, 如图 3.3 所示。

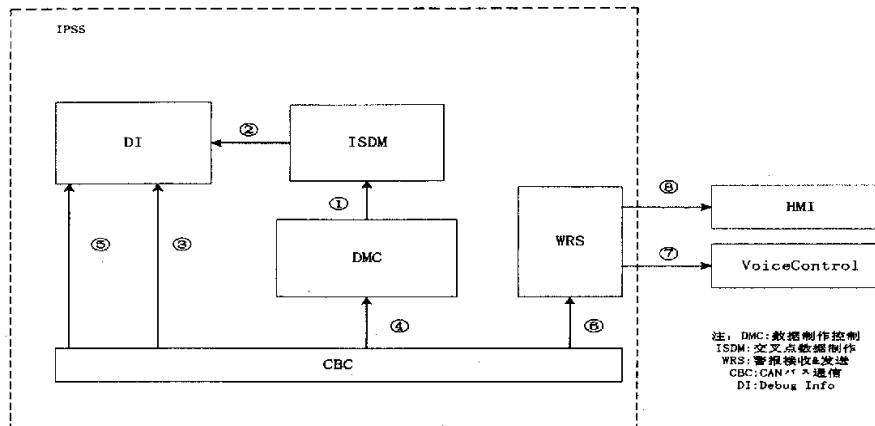


图 3.3 导航系统内部模块关联
Fig. 3.3 Module Relation in Navigation System

这里, 将语音控制模块和 CANBus 通信模块从机器控制 Service 和 DeviceDriver 中单独提取出来, 就能够更清晰的看到各个模块之间关系以及它们之间是如何工作的。

其中的 IPSS 模块功能较复杂, 在这里被分解成多个子模块, 其中各子模块的功能分工可以参照表 3.1 中的内容。

表 3.1 IPSS 内部子模块分工
Table 3.1 Function of Submodule in IPSS

模块名称	实现功能
DMC (DataMakeControl)	控制 ISDM 制作数据
ISDM (IntersectionDataMake)	预测经路、检出经路上的对象交叉点、做出对象交叉点的信息信息和部分 Debug 信息
WRS (WarningRequestService)	接受 CBC 的 Command, 控制 Voice Control 发警报音、控制 HMI 显示警报 Icon
CBC (CANBusControl)	将 ISDM 做成的对象交叉点信息和 Debug 信息发送给处理部 (PC), 接收处理部 (PC) 发送过来的警报 Command 传递给 WRS
DI (DebugInformation)	提供 Debug 信息
Voice	发警报音
UserInterface	显示警报 Icon

对于各部分子功能的具体实现, 在后面几章中会详细加以说明。

第四章 经路预测与对象交叉点制作

为了更好地实现功能，给驾驶者提供真正真实有效的信息，需要对车辆的运行状况进行很好的预测并对道路状况进行有效地分析。

在实际的世界里，道路情况千变万化，各不相同，并不是所有的交叉点都是我们需要关注的内容，那么到底我们应该关注什么样的交叉点成为我们需要优先解决的一个课题。

交叉点支援这一功能的主旨在于预防撞头事故的发生，因此分析此类事故在什么样的状况下发生的概率更高是制定有效算法的重要依据。

4.1 经路预测

所谓经路预测是指根据当前车辆所在的位置以及车辆前方的道路状况，预测出前方可能存在危险的道路。

4.1.1 经路预测方案分析

1、经路预测对象的选择

当驾驶者驾驶车辆驶向一个交叉点的时候，并不是所有的路线都存在危险，或者说对于大多数可选择路线，存在危险的可能性会小得多。

在一个通常的十字路口，对于绝大多数驾驶者来说，在转弯之前都会明显的降低车辆行驶的速度，左转或右转的的行车路线被认为是相对安全的。真正存在危险的行车路线是按照直行的方向行驶，因为通常情况下，直行的车辆不会明显的降低车速，一旦某些外来因素影响了驾驶者的注意力，使得驾驶者在通过路口的时候没有注意到路口发生的意外事件，例如信号灯的变化、突然出现的行人等，就有可能引发不可预期的事故。

由此，我们将直行的方向作为危险经路的选择对象，同时将该方向上存在的交叉路口作为分析危险路口的研究对象。

2、经路预测长度的选择

作为紧急预警功能，对实时性的要求很高，这样就要求在保证数据充足的前提下，需要处理的数据量尽可能少，数据处理的周期尽可能短。

考虑到驾驶者驾驶车辆的随意性，不适宜处理太多的道路数据，这样可以避免由于实际行驶经路脱离预测经路后造成的数据浪费。另外，交叉点的数量应该控制在比较少的数量下，这样处理的数据量也会比较少，可以一定程度上保证性能的要求。

从实际情况上来讲，在车辆和交叉点之间的距离大于一定长度的时候，驾驶者通常还有比较充分的时间来对交叉点的状况作出正确的反应。

基于以上考虑，初步将预测的经路长度定为 100m，基本上可以满足实际情况的要求。

3、经路预测起始条件和终止条件

那么什么时候开始进行经路预测以及什么时候终止经路预测呢？从客观上很容易给出一个感性的认识，但对于一个算法来讲，需要能够给出一个明确的定义。

在本次研究过程中，对经路预测的起始条件和终止条件进行了如下定义。

经路预测的起始条件包括：

- (1) 车辆初始启动时；
- (2) 导航车辆定位模块判断车辆偏离之前预测的经路。

经路预测的终止条件包括：

- (1) 预测的经路长度达到 100m；
- (2) 根据现有的信息无法找出更合理的选择。

4.1.2 经路预测算法设计

首先引入电子地图数据中关于 Link 和 Link 列的概念。

Link 是 DB 中存储道路信息的基本单位，每一个 Link 所表示的道路具有相同的道路属性，包括路宽、名称、番号等，且 Link 中不存在与其他 Link 的交点。

每一个 Link 根据道路通行方向的不同可以分为单方向 Link 和双方向 Link，为了表示双方向 Link 的走行方向，将 Link 的两个方向分别表示为顺方向和逆方向，而单方向 Link 就是指实际道路中的单行线，只有顺方向一个方向。

Link 列是一系列具有相同属性的 Link 所构成的 List。同一 Link 列中的道路 Link 是同一条道路被一个或多个交叉路口分割而成的，因此通常也具有相同的属性。

实际上，一些较为重要的道路通常两个走行方向是隔离开的，因此这些道路往往被分开存储为两个单方向 Link 列，这种 Link 列也被称为上下线分离 Link。通常情况下，同一个 Link 列中两个接续 Link 的联结方式都是直行的。

这里，我们将存在 Link L_i 的 Link 列记为 $L_1:L_N$ 。这里 L_1 是 $L_1:L_N$ 的始端 Link， L_N 是终端 Link。 $(1 \leq i \leq N, N$ 是 Link 列 $L_1:L_N$ 的 Link 数)。

令车辆现在位置对应的道路 Link 为 L_C 。通过以下 4 个 Step 得到我们想要的预测经

路：

1、Step 0——预测经路的初始化和再初始化

- (1) 如果预测经路 PR 为空 (empty), $Li = LC$, 执行 Step 1。
- (2) 如果预测经路 PR 不为空 (empty),
 - a) 从 PR 中将已经通过的道路 Link 删除。 $Li = PR$ 中最后一条 Link, 执行 Step1。
 - b) 当车辆所在 Link 在 PR 中不存在时, 将 PR 清空, $Li = LC$, 执行 Step 1。

2、Step 1——直行道路的选择 1

- (1) 如果车辆的行进方向是 Link Li 的顺方向,
 - a) 将 Link 列中从 Li 到 LN 追加到预测经路 PR 中,
 - b) $Ne=LN$ 的终端 Node, $Le=LN$ 。
- (2) 车辆的行进方向是 Link Li 的逆方向,
 - a) Link 列中从 Li 到 $L0$ 追加到预测经路 PR 中,
 - b) $Ne=L0$ 的始端 Node, $Le=L0$ 。
- (3) 如果预测经路 PR 的长度大于或等于 $100m+LC$ 的长度, 处理终了。
- (4) 如果预测经路 PR 的长度小于 $100m+LC$ 的长度, 执行 Step2。

3、Step 2——直行道路的选择 2

- (1) 将与 Node Ne 接续的所有道路 Link 的集合记为 $S0$ 。
- (2) 从集合 $S0$ 中将以下几种 Link 删除, 得到集合 $S1$.
 - a) Link 方向朝向 Ne 的上下线分离 Link
 - b) Link 方向朝向 Ne 的单方向通行 Link
 - c) Le
- (3) 集合 $S1$ 中的元素个数为 0, 处理终了 (没有可以预测的经路)。
- (4) 如果集合 $S1$ 中的元素个数为 1, 将 $S1$ 中的唯一元素作为 Li , 执行 Step 1。
- (5) 如果集合 $S1$ 中存在 2 个或 2 个以上元素, 执行 Step 3。

4、Step 3——分歧道路的选择 1

- (1) 将集合 $S1$ 中的元素, 按照下面的优先顺序进行排序。
道路种别 > Link 种别 > 车线数 > 道路宽度
- (2) 构成集合 $S1$ 的元素中, 优先度最高的元素构成集合 $S2$, 集合 $S2$ 中元素的个数可能是多个。
- (3) 如果集合 $S2$ 中的元素个数为 1, 将 $S2$ 中的唯一元素作为 Li , 执行 Step 1。

(4) 如果集合 S2 中存在 2 个或 2 个以上元素, 执行 Step 4。

5、Step 4——分歧道路的选择 2

(1) 计算集合 S2 中各元素和道路 Link Le 的方位差的绝对值。

(2) 将方位差的绝对值最小的元素和第二小的元素的方位差进行比较。

(3) 方位差的绝对值的差小于 10 度的情况, 处理终了(没有可以预测的经路)。

(4) 方位差的绝对值的差大于或等于 10 度的情况, 将方位差的绝对值最小的元素作为 Li, 执行 Step 1。

循环执行上面的四个步骤, 直至满足计算停止的条件, 即可得到一条预测经路。我们认为, 车辆沿着这条预测经路行驶的可能性较大, 而且在沿着该经路行驶时存在出现危险的可能。

4.2 对象交叉点抽出

在上一节中, 我们计算出了一条预测经路, 接着, 我们需要从这条经路上抽出需要重点关注的交叉点, 我们称之为对象交叉点。同样的, 我们选择的对象交叉点也应该是那些出现危险的可能性较高的路口。为此, 我们设定了一系列的判断条件作为对象交叉点的衡量标准, 如图 4.1 所示。

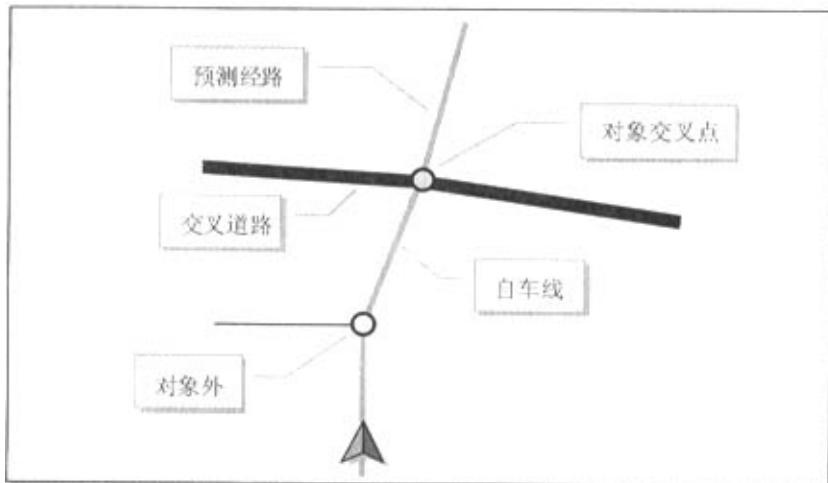


图 4.1 预测经路和对象交叉点
Fig. 4.1 Forecast Route and Object Intersection

- (1) 存在信号灯的交叉点。
- (2) 与该交叉点相连的道路中存在比当前道路优先度更高的道路。
- (3) 与该交叉点相连的道路中存在与当前道路具有同等优先度的道路。

这里提到的优先度，和经路预测过程中使用的道路优先度的概念相同。

但是，除了距离车辆最近的交叉点，第二个交叉点之后的对象交叉点需要附加下面的条件：从车辆正面看，相对方位在 $\pm 20^\circ$ 以内的交叉点。

这样做的原因是考虑到在进行整体的危险判定的过程中，需要同时参考从 DB 中抽出的交叉点信息和对摄像机采集到的图像的识别结果。由于通常情况下，位于车辆前部的摄像机是面向正前方的，而摄像机具有一定的摄像角度，拍摄不到相对方位 $\pm 20^\circ$ 以外的物体，因此对于该范围之外的交叉点无法和图像识别的结果进行综合分析，所以暂时不把这部分交叉点作为对象交叉点，如图 4.2 所示。

随着车辆的不断向前行驶，车头的方向也会不断的发生变更，一些 $\pm 20^\circ$ 范围以内的交叉点会变成对象外交叉点，一些 $\pm 20^\circ$ 范围以外的交叉点也可能变成对象交叉点。因此，和预测经路不同，在每次对对象交叉点进行处理的时候，无法保存上一次做成的结果，必须重新做成。

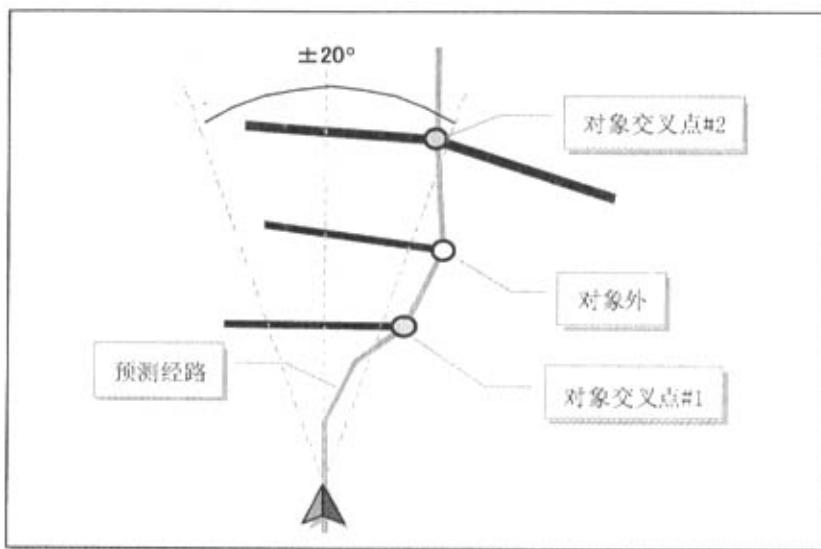


图 4.2 对象交叉点#2~#N 是 $\pm 20^\circ$ 范围以内
Fig. 4.2 Object Intersection #2~#N Must in the range of $\pm 20^\circ$

4.3 交叉点信息制作

接下来，需要制作各种交叉点信息，这些交叉点信息将在最终的危险判定过程中作为依据来使用。这些信息包括：

- 1、经路上对象交叉点的总数以及按照到车辆的距离的序号#1、#2、…、#N。
- 2、各交叉点被判断为对象交叉点的理由和根据。

- (1) 是否存在信号灯。(true or false)
- (2) 根据道路种别判断, 交叉道路比当前道路优先。(true or false)
- (3) 根据车线数判断, 交叉道路比当前道路优先。(true or false)
- (4) 根据道路宽度判断, 交叉道路比当前道路优先。(true or false)

除了信号灯之外的三个判断基准, 当存在两个以上条件符合或存在冲突的情况下, 按照下面的优先度只记录其中的一个。优先顺序是道路种别 > 车线数 > 道路宽度。具体的判断方式如下:

```
if(根据道路种别判断, 交叉道路比当前道路优先) {  
    当前道路的信息 = 当前道路的道路种别;  
    交叉道路的信息 = 交叉道路的道路种别;  
} else if(根据车线数判断, 交叉道路比当前道路优先) {  
    当前道路的信息 = 当前道路的车线数;  
    交叉道路的信息 = 交叉道路的车线数;  
} else if(根据道路宽度判断, 交叉道路比当前道路优先) {  
    当前道路的信息 = 当前道路的道路宽度;  
    交叉道路的信息 = 交叉道路的道路宽度;  
} else {  
    当前道路的信息 = 不定;  
    交叉道路的信息 = 不定;  
}
```

信号灯的有无, 根据地图数据中对应该交叉点的 Node 的属性来判断。

道路种别的优先度, 根据比较连接在对应该交叉点的 Node 上的各条道路 Link 的种别 Code 来得到。例如, 如果当前道路是县道而交叉道路是国道, 就可以判定交叉道路的优先度比当前道路高。

关于车线数和道路宽度, 地图数据存储的道路 Link 的属性中记录了该条 Link 所表示的实际道路的车线数的道路宽度, 可以根据这一属性判断出当前道路和交叉道路中哪条道路的车线数更多, 哪条道路更宽。

如果有两条交叉道路和这个交叉点相连, 无论道路种别、车线数、还是道路宽度, 都使用其中优先度最高的交叉道路与当前道路进行比较。表 4.1、表 4.2 和表 4.3 分别列出了电子地图数据中关于车线数、道路种别和道路宽度的定义以及优先度比较。

表 4.1 地图数据中的车线信息定义
Table 4.1 Lane Information Definition in DB Data

优先度	bit2	bit1	bit0	含义
低	0	0	0	双方向 1 车线 (窄车线)
	0	0	1	单方向 1 车线 (窄车线)
	0	1	0	双方向 1 车线 (宽车线)
	0	1	1	单方向 1 车线 (宽车线)
	1	0	0	单方向 2 车线
	1	0	1	单方向 3 车线
	1	1	0	单方向 4 车线
	1	1	1	单方向 5 车线以上
高				

表 4.2 地图数据中的道路种别定义
Table 4.2 Road Kind Definition in DB Data

道路种别 Code	道路种别 (日本)	道路种别 (欧美)
0 (优先度 高)	高速	Freeway class road (1)
1	都市高速	Freeway class road (2)
2	国道	Highway class road (> 91KPH)
3	主要地方道	Throughway class road (51-90KPH)
4	县道	Local road class road (31-50KPH)
5	一般道路 1 (干线)	Frontage road
6	一般道路 2 (其他可引导道路)	Very low speed road (< 30KPH)
7	导入路	Private road
8	细街路 1	Walkway
9	细街路 2 (走行中表示)	Non-navigable road
10	轮渡航线	Ferry route
11		Car Train
12		Public vehicle only road
13		Carpool lane
14		(RESERVED)
15 (优先度 低)		(RESERVED)

表 4.3 地图数据中的道路宽度定义
Table 4.3 Road Width Definition in DB Data

优先度	Bit2	bit1	bit0	含义
低	0	0	0	路宽 3.0m 以下
	0	0	1	路宽 3.0m 以上 5.5m 以下
	0	1	0	路宽 5.5m 以上 13.0m 以下
	0	1	1	路宽 13.0m 以上
	1	0	0	(RESERVED)
	1	0	1	(RESERVED)
	1	1	0	(RESERVED)
	1	1	1	(RESERVED)
高				

3、从车辆当前位置到各对象交叉点的沿路距离

为了计算图像识别 Senser (停止线检出 Senser、信号灯/标志检出 Senser) 的启动时机, 需要计算沿着道路的 Link 形状, 从车辆当前位置到各交叉点的折线长度, 如图 4.3 所示。

在这里，简单介绍一下 DB 格式中形状补间点的概念。前面曾经提到过 Link 是 DB 格式中存储道路的最小单位，每一个 Link 当中不存在与其他道路的连接点。但是，道路的形状是不规则的，即使在一个 Link 上也不能保证是一条直线，DB 格式中使用折线来近似表示道路的形状，而这些折线中间的顶点就叫做形状补间点，每个 Link 上可以包括很多形状补间点，也可以没有任何形状补间点。

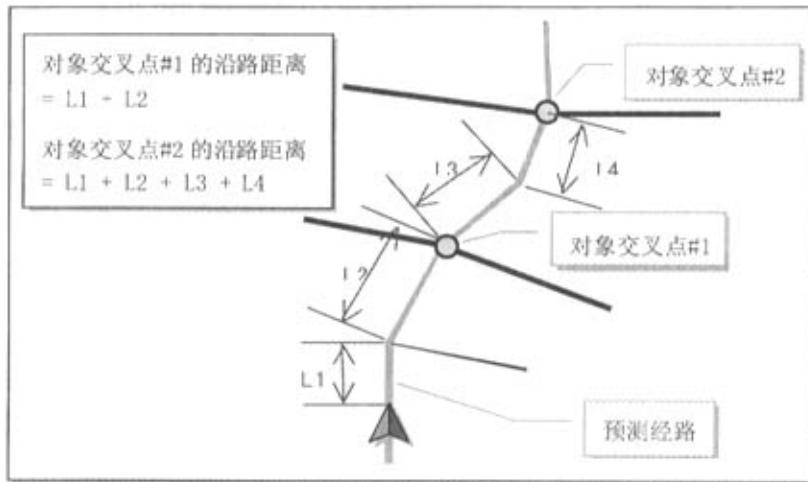


图 4.3 从车辆当前位置到各对象交叉点的沿路距离
 Fig. 4.3 Curve Distance from Vehicle to Intersections

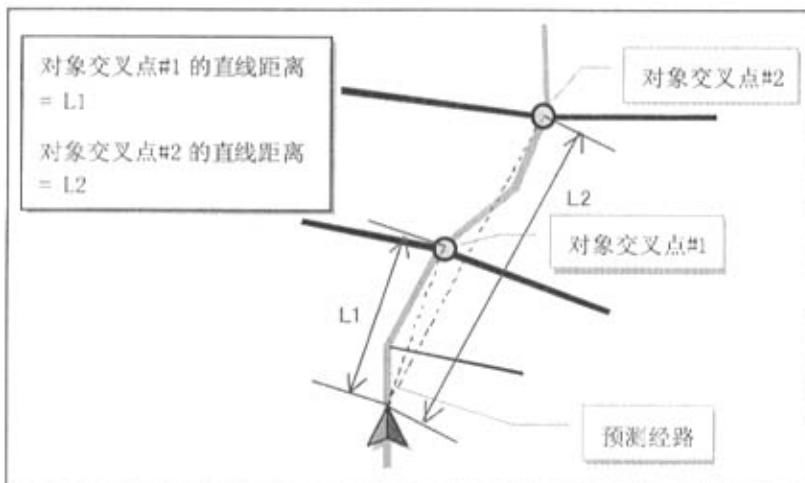


图 4.4 从车辆当前位置到各对象交叉点的直线距离
 Fig. 4.4 Direct Distance from Vehicle to Intersections

4、从车辆当前位置到各对象交叉点的直线距离

计算从车辆当前位置到各对象交叉点的直线距离是为了协助锁定前方交叉点中

的信号灯或其他标识，如图 4.4 所示。

5、从车辆正面到各对象交叉点的相对方位

同样是为了更好的锁定交叉点周边的标识，需要计算从车辆正面到各对象交叉点的相对方位，如图 4.5 所示。

对于摄像机来说，无论近处的物体还是远处的物体都会被拍摄下来，这样对判断哪一个信号灯或标识是来自于最近的交叉点存在一定的难度，但如果结合前方各交叉点的相对方位，判断的难度会降低很多。

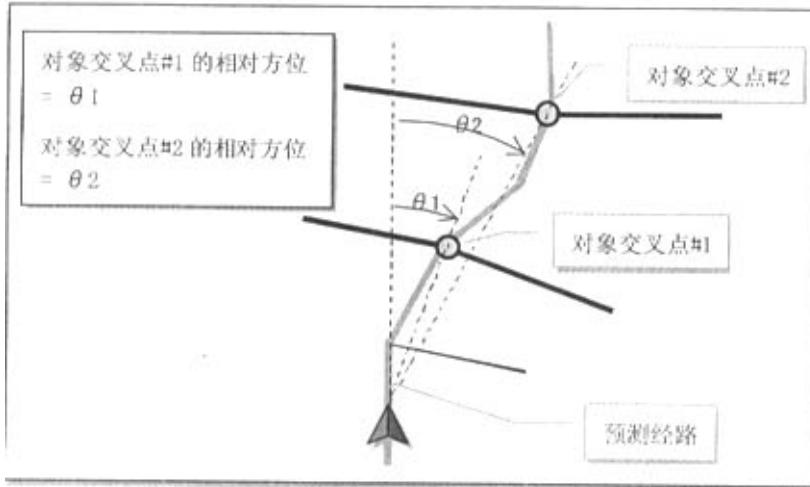


图 4.5 从车辆当前位置到各对象交叉点的相对方位
Fig. 4.5 Relative Direction from Vehicle to Intersections

6、从各对象交叉点面向当前车辆方向 50m 区间内的道路曲率

计算道路的曲率，如图 4.6 所示，主要是在下面这些情况下使用。

- (1) 检出前方道路上的停止线
- (2) 调整车辆前部的摄像机朝向车辆前方 30~50m 距离的道路

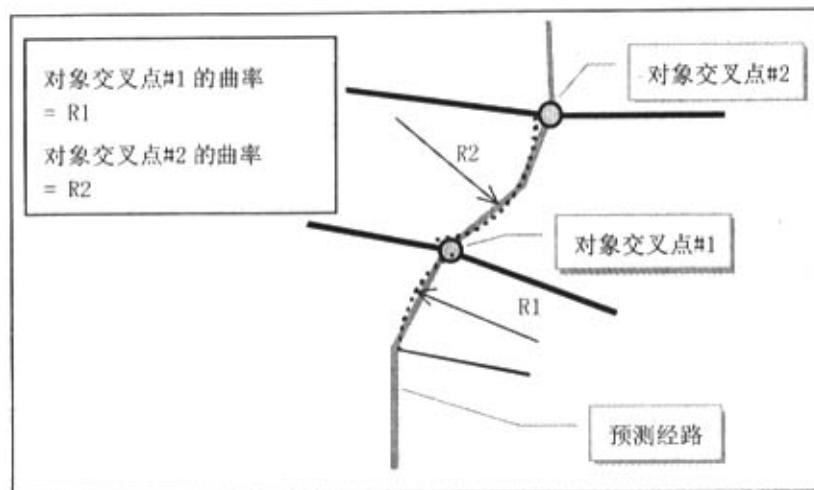


图 4.6 各对象交叉点前方的道路曲率
Fig. 4.6 Road Curvature Before Intersections

在计算曲率的过程当中，基本上采用的是三角形的正弦定理。参考下面的公式即可计算出三角形顶点处的近似曲率，如图 4.7 所示。

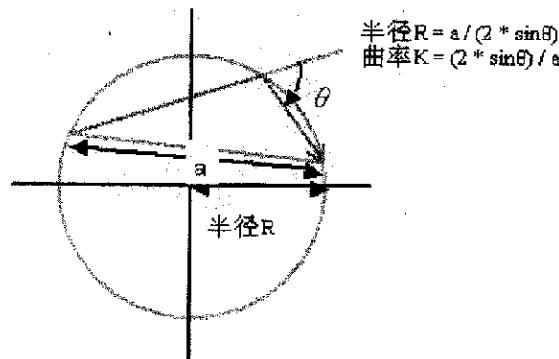


图 4.7 计算近似曲率的数学方法
Fig. 4.7 Math Method for Calculating Approximate Curvature

7、各对象交叉点的标高差

所谓标高差，是指车辆前方某一点对于车辆当前位置的相对高度。标高差的算出同样有利于锁定前方交叉点的相对位置，如图 4.8 所示。

标高差按照以下方式算出：

标高差#1 = 对象交叉点#1 的标高 - 车辆现在地的标高

标高差#2 = 对象交叉点#2 的标高 - 对象交叉点#1 的标高

.....

标高差#N = 对象交叉点#N 的标高 - 对象交叉点#N-1 的标高

车辆当前位置的标高为 $H(0)$ 、对象交叉点#i 的标高为 $H(i)$ ，则

$$\text{标高差}#I = H(i) - H(i-1)$$

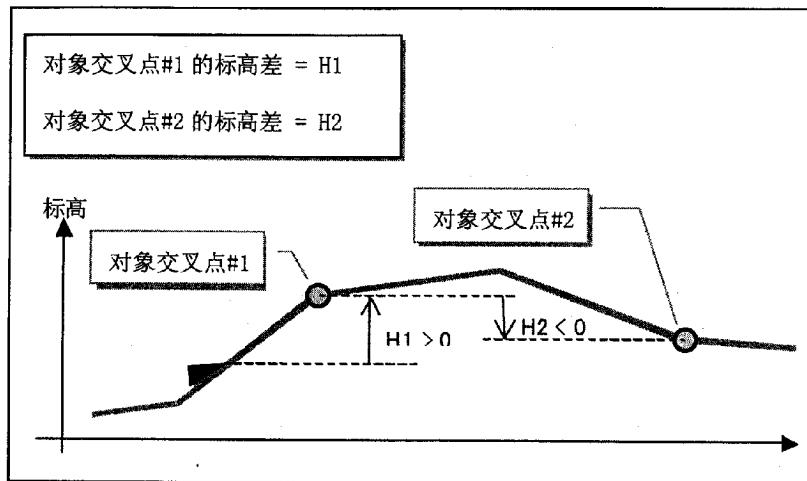


图 4.8 对象交叉点的相对高度
Fig. 4.8 Relative Altitude of Intersections

4.4 实现逻辑

经路预测和对象交叉点信息制作的工作是由 IPSS 内部的 DMC 模块和 ISDM 模块共同实现的^[20]。其中，DMC 模块负责数据制作的控制，包括定时启动数据制作工作以及在车辆位置发生更新时向 ISDM 发送数据更新通知；ISDM 模块负责实际的数据制作工作，根据地图数据以及当前的车辆位置情况，计算出相应的预测经路和对象交叉点信息，并将计算结果发送给 DI 和 CBC 等模块。

DMC 和 ISDM 与其他模块之间的关系可以参考图 4.9，图中表示的各个步骤的工作内容分别为：

- (1) 线程启动，同时启动时间 timer。之后 DMC 会根据 timer 每隔 1 秒钟触发一次数据的更新。
- (2) ISDM 获得数据制作事件后开始数据制作。
- (3) 调用 SSGIP 函数取得交叉点信息数据，这里的 SSGIP 可以看成是一系列从地图数据中解析交叉点信息的函数群。

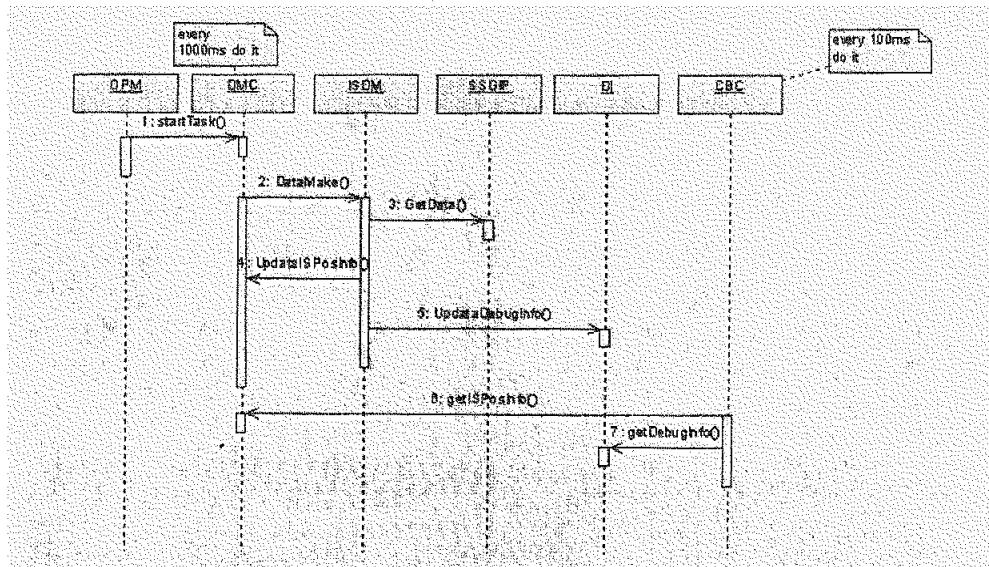


图 4.9 DMC、ISDM 与其他模块之间的关系
Fig. 4.9 Relation Between DMC, ISDM and Other Modules

- (4) 将 ISDM 制作的提示点信息保存在 DMC 中。
- (5) 制作 Debug 信息 登陆到 DI。
- (6) CBC 根据时间 timer 从 DMC 中取得交叉点信息。
- (7) 从 DI 中取得 debug 信息。

需要注意的是 CBC 从 DMC 取得交叉点信息的 Timer 是 CANBus 总线数据更新的频率，这个频率比 DMC 触发数据更新的频率要高（前者为每秒 10 次，后者是每秒 1 次）。这样处理主要是考虑到从程序上只能控制开始进行数据更新的时间，但无法控制数据更新结束的时间，因此为了保证每次数据更新的结果都能够反映到 CANBus 总线上，因此加大了向 CANBus 总线上输出数据的频率。当接受端接收到相同的数据时可以忽略不进行处理^[21]。

为了尽可能的提高性能，避免重复进行交叉点信息的制作，实际实现的过程中多处采用了数据再利用的机制，每次更新交叉点信息时都是在原有信息的情况下进行数据的更新，只有在数据完全不可用的情况下才会重新制作数据^[22]。

第五章 CANBus 总线数据传输

本章主要介绍如何通过 CANBus 完成导航设备和 PC 之间的数据传输。包括对 Driver 的改造和数据格式定义以及导航系统内部的实现流程。

5.1 协议定制及 Driver 改造

5.1.1 CANBus 报文格式

CANBus 总线上的信息以报文的形式进行传输，报文传输分为四种不同类型的帧：数据帧、远程帧、错误帧和过载帧^[16]。

数据帧从一个发送器承载数据到一个接收器。根据 CAN 规范，有两种数据帧格式：CAN 标准帧（也称为 CAN2.0A，支持 11 位长度的标识符）和 CAN 扩展帧（也称为 CAN2.0B，支持 29 位长度的标识符）。

远程帧是由一个接收 CAN 节点发送，用来请求带有远程帧中规定的标识符的数据帧。

错误帧将任何总线错误通知其它单元，在接收到这个帧时发送器会自动进行消息重发。

过载帧由一个忙的 CAN 节点送出，以请求在前后数据帧之间增加一个额外的延迟。

SOF	Identifier	RTR	Control	Data	CRC	ACK	EOF
-----	------------	-----	---------	------	-----	-----	-----

图 5.1 标准格式的数据帧
Fig. 5.1 Standard Format of Data Frame

图 5.1 是标准格式的数据帧，开始是帧起始 SOF (start-of-frame)；其后是 11 位标识符和远程发送请求位 RTR (remote transmission request)，这两部分构成了仲裁域；控制域由 6 位组成，它表示了后面数据域中的字节数目；数据域由数据帧里发送的数据组成可为 0—8 个字节；数据域后面是循环冗余码 CRC (cyclic redundancy checksum) 域，它用于接收器检验所接收到的位序列；两位的应答域 ACK (acknowledgment) 用于发送器接收任意接收器所发出的应答；最后是帧结尾 EOF (end-of-frame) 它包括 7 个位。

这种按数据块编码的方式，还可使不同的节点同时接收到相同的数据，这一点在分布式控制系统中非常有用。数据段长度最多为 8 个字节，可满足通常工业领域中控制命令、工作状态及测试数据的一般要求。

5.1.2 CANBus 接口的重新定义

在现有的导航系统平台中已经加载了 CANBus 的相关驱动, 用户可以根据需要按照设备文件读取的形式对 CANBus 进行读写操作。为了更好地实现使用 CANBus 进行数据的传输, 重新定制了一批用于进行 CANBus 操作的接口函数。

(1) vddCANCreateFile

该函数用来打开 CANBus 总线, 在对 CANBus 总线进行操作之前, 必须使用该函数取得总线的使用权。接口函数按照下面的方式进行定义:

```
HANDLE vddCANCreateFile(
    LPCTSTR     lpFileName,
    DWORD       dwDesiredAccess,
    DWORD       dwShareMode,
    LPSECURITY_ATTRIBUTES lpSecurityAttribute,
    DWORD       dwCreationDisposition,
    DWORD       dwFlagsAndAttributes,
    HANDLE      hTemplateFile
);
```

各参数所表示的内容如表 5.1 所示:

表 5.1 vddCANCreateFile 函数参数说明
Table 5.1 Declaration of Parameter in vddCANCreateFile

名称	内容	I/O	类型
返回值 Result	设备Open结果	O	HANDLE (成功: AccessHandle; 失败: NULL)
参数1 lpFileName	设备名	I	LPCTSTR ("VddCAN")
参数2 dwDesiredAccess	访问模式	I	DWORD (GENERIC_READ GENERIC_WRITE)
参数3 dwShareMode	设备共享模式	I	DWORD (FILE_SHARE_READ GENERIC_WRITE)
参数4 lpSecurityAttribute	—	I	LPSECURITY_ATTRIBUTES
参数5 dwCreationDisposition	机能模式	I	DWORD (OPEN_EXISTING)
参数6 dwFlagsAndAttributes	设备属性	I	DWORD (FILE_ATTRIBUTE_NORMAL)
参数7 hTemplateFile	—	I	HANDLE

(2) vddCANClose

该函数用来关闭 CANBus 总线, 在一个用户完成对 CANBus 总线的读写操作之后, 需要使用该函数放弃总线的使用权。接口函数按照下面的方式进行定义:

```
BOOL vddCANClose(HANDLE hFile);
```

各参数所表示的内容如表 5.2 所示:

表 5.2 vddCANClose 函数参数说明
Table 5.2 Declaration of Parameter in vddCANClose

名称	内容	I/O	类型
返回值 Result	设备Close结果	O	BOOL (成功: TRUE; 失败: FALSE)
参数1 hFile	对象Handle	I	HANDLE

(3) vddCANIoControl

该函数用来对 CANBus 总线进行 I/O 控制，包括从总线读出以及向总线写入数据和其他控制信息。接口函数按照下面的方式进行定义：

```
BOOL vddCANIoControl(
    HANDLE      hFile,
    DWORD       dwIoControlCode,
    LPVOID      lpInBuffer,
    DWORD       nInBufferSize,
    LPVOID      lpOutBuffer,
    DWORD       nOutBufferSize,
    LPDWORD     lpBytesReturned,
    LPOVERLAPPED lpOverLapped
)
```

各参数所表示的内容如表 5.3 所示：

表 5.3 vddCANIoControl 函数参数说明
Table 5.3 Declaration of Parameter in vddCANIoControl

名称	内容	I/O	类型
返回值 Result	I/O 处理结果	O	BOOL (成功: TRUE; 失败: FALSE)
参数1 hFile	文件Handle	I	HANDLE
参数2 dwIoControlCode	控制机能编号	I	DWORD
参数3 lpInBuffer	输入数据Buffer	I	LPVOID
参数4 nInBufferSize	输入数据Size	I	DWORD
参数5 lpOutBuffer	输出数据Buffer	O	LPVOID
参数6 nOutBufferSize	输出数据Size	I	DWORD
参数7 lpBytesReturned	实际输出的数据Size	O	LPWORD
参数8 lpOverLapped	—	I	LPOVERLAPPED

其中的 dwIoControlCode 参数指定了对 CANBus 进行了哪种操作，具体的定义方式参见表 5.4：

表 5.4 vddCANIoControl 控制 Code 说明
Table 5.4 Declaration of Control Code in vddCANIoControl

机能控制标识符	I/O	分类	机能
VDD_CAN_SET_CAN_DATA	I	控制	向CAN写入情报
VDD_CAN_GET_CAR_INFO	O	变化时取得	车辆情报变化时的取得要求
VDD_CAN_GET_MAINTENANCE_INFO	O	读取	Maintenance情报的即时取得要求
VDD_CAN_GET_DIAG_INFO	O	读取	Diag用情报的即时取得要求
VDD_CAN_GET_CAN_BUS_INFO	O	变化时取得	CanBUS状态变化时的取得要求
VDD_CAN_GET_NOW_CAR_INFO	O	读取	车辆情报的即时取得要求
VDD_CAN_GET_NOW_MAINTENANCE_INFO	O	读取	Maintenance情报的即时取得要求
VDD_CAN_GET_NOW_DIAG_INFO	O	读取	Diag用情报的即时取得要求
VDD_CAN_GET_NOW_TURN_SIGNAL_STATUS	O	读取	信号灯情报的即时取得要求
VDD_CAN_GET_NOW_WIPER_STATUS	O	读取	雨刷情报的即时取得要求
VDD_CAN_GET_NOW_LIGHT_STATUS	O	读取	车灯情报的即时取得要求
VDD_CAN_GET_NOW_SW_MODE	O	读取	雪路情报的即时取得要求
VDD_CAN_GET_NOW_BRAKE_STATUS	O	读取	制动器情报的即时取得要求
VDD_CAN_GET_NOW_CAN_BUS_INFO	O	读取	CanBUS状态的即时取得要求

5.2 数据通信格式定义

由于每一个 CANID 每次只能传输 8 个字节, 需要定义一套数据格式以满足交叉点信息传输的需要。

结合前面章节中对交叉点信息的定制, 定义了如图 5.2 的数据结构:

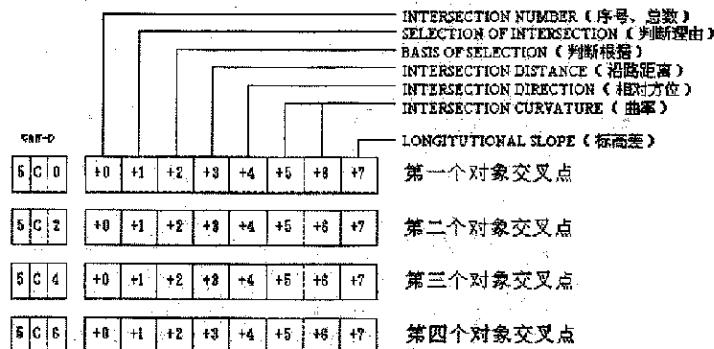


图 5.2 输出数据格式定义
Fig. 5.2 Format Definition of Output Data

INTERSECTION POSITION BODY {	8byte	交叉点信息 (× 4 个)
INTERSECTION NUMBER FIELD	1byte	对象交叉点信息序号
SELECTION OF INTERSECTION FIELD	1byte	判断理由的 BitPattern
BASIS OF SELECTION FIELD	1byte	判断根据
INTERSECTION DISTANCE FIELD	1byte	沿路距离
INTERSECTION DIRECTION FIELD	1byte	相对方位
INTERSECTION CURVATURE FIELD	2byte	道路曲率
LONGITUDINAL SLOPE FIELD	1byte	标高差
}		

在这个数据结构中, 使用一到两个字节来表示距离、方位、曲率等数值, 使用一个字节的不同位来表示顺序号、匹配状态以及判断理由等信息, 做到了数值空间利用的最大化。

对于发送的交叉点信息, 做以下要求:

- (1) 检出的对象交叉点的个数最大为 4。
- (2) 按照从车辆现在位置的沿路距离较近的对象交叉点顺序发送。
- (3) 按照每个对象交叉点使用一个 CAN-ID 计算, 一共需要使用四个 CAN-ID。
- (4) 每次向 CAN 总线发送 4 个交叉点信息。检出的对象交叉点的个数不足 4 个的

情况下，也发送 4 个交叉点信息。

例如，检出的对象交叉点的个数是 3 个的情况下，开头的 3 个交叉点信息设定为有效值，剩余的一个设定为无效值。交叉点信息是否有效按照对象交叉点信息中的序号来识别。

对于交叉点信息中的每一项，也分别进行了如下的详细定义：

1、对象交叉点信息顺序号

```
INTERSECTION NUMBER FIELD {      1byte  对象交叉点信息顺序号
    map matching status      2bit 检出时的匹配状态
    update counter          2bit 数据更新计数
    route guide demo flag  1bit Demo 走行模式 flag
    intersection total      3bit 对象交叉点的总数
}
```

在对象交叉点信息番号中设定的内容包括检出的对象交叉点的总数、检出时的匹配状态、Demo 走行模式 flag 和数据更新计数。

(1) 检出时的匹配状态 (map matching status):

交叉点检出时的匹配状态按照下面的定义来设定，

00b: 能够确定车辆走行在道路 Link 上的状态。

01b: 能够确定车辆走行在道路 Link 上的状态，但可信度比较低。

10b: (RESERVED)。

11b: 无法确定车辆走行在道路 Link 上的状态。

(2) 数据更新计数 (update counter):

数据更新计数设定为 0~3 的值，初次的设定值是 0，对象交叉点信息每更新一次加 1。在上次的设定值基础上加 1 后的结果溢出的情况下设定为 0。

(3) Demo 走行模式 flag (route guide demo flag):

Demo 走行模式 flag 设定为现在的动作模式，其意义分别是，

0: 通常模式 (也就是车辆实际走行的模式)

1: Demo 走行模式 (利用 Demo 走行机能模拟车辆走行的模式)

(4) 对象交叉点的总数 (intersection total):

检出的交叉点的总数设定为 (0~4) 之间的数值，

0: 检出 0 个对象交叉点。

- 1: 检出 1 个对象交叉点。
- 2: 检出 2 个对象交叉点。
- 3: 检出 3 个对象交叉点。
- 4: 检出 4 个对象交叉点。
- 5: (RESERVED)。
- 6: (RESERVED)。
- 7: 无法检出对象交叉点。

这里, 数据更新计数是为了在接受端能够区分更新前和更新后的数据。

表 5.5 中详细的表示了对象交叉点信息番号中各项值的含义。其中对象交叉点信息番号中对应数据更新计数的两个 bit 表示为 “*”。

表 5.5 输出用对象交叉点信息详细
Table 5.5 Detail Information of Intersection for Output

对象交叉点 情报序号	交叉点检出结果	走行匹配状态	动作模式
00**0000b	检出0个对象交叉点。	能够确定车辆走行在道路Link上。 通常模式	
00**0001b	检出1个对象交叉点。		
00**0010b	检出2个对象交叉点。		
00**0011b	检出3个对象交叉点。		
00**0100b	检出4个对象交叉点。		
01**0000b	检出0个对象交叉点。	能够确定车辆走行在道路Link上但 可信度比较低。	
01**0001b	检出1个对象交叉点。		
01**0010b	检出2个对象交叉点。		
01**0011b	检出3个对象交叉点。		
01**0100b	检出4个对象交叉点。		
11**0111b	无法检出对象交叉点。	无法确定车辆走行在道路Link上。	
00**1000b	检出0个对象交叉点。	能够确定车辆走行在道路Link上。 案内Demo走行模式	
00**1001b	检出1个对象交叉点。		
00**1010b	检出2个对象交叉点。		
00**1011b	检出3个对象交叉点。		
00**1100b	检出4个对象交叉点。		
01**1000b	检出0个对象交叉点。	能够确定车辆走行在道路Link上但 可信度比较低。	
01**1001b	检出1个对象交叉点。		
01**1010b	检出2个对象交叉点。		
01**1011b	检出3个对象交叉点。		
01**1100b	检出4个对象交叉点。		
11**1111b	无法检出对象交叉点。	无法确定车辆走行在道路Link上。	
其他	(RESERVED)	N/A	N/A

2、判断理由

SELECTION OF INTERECTION FIELD { 1byte 判断理由

```

    signal      1bit 有信号灯
    kinds of road  1bit 根据道路种别判断交叉道路比当前道路优先
    number of lane  1bit 根据车线数判断交叉道路比当前道路优先
    width of road   1bit 根据道路宽度判断交叉道路比当前道路优先
    Filler        4bit (全为 0)
}

```

这里的各个 bit 按照下面的规则分别设定为 0 或 1。

Signal: 有信号灯的情况下为 1, 否则为 0

Kinds of road: 交叉道路的道路种别比当前道路优先的情况下为 1, 否则为 0

Number of lane: 交叉道路的车线数比当前道路多的情况下为 1, 否则为 0

Width of road: 交叉道路比当前道路宽的情况下为 1, 否则为 0

当没有检出或无法检出对象交叉点的时候, 设定为任意值。

3、判断根据

```

BASIS OF SELECTION FIELD { 1byte 判断根据
    information of cruising lane  4bit 当前道路信息
    information of crossing lane  4bit 交叉道路信息
}

```

这里的当前道路信息和交叉道路信息根据判断理由的不同设定为当前道路和交叉道路的道路种别、车线或路宽。

当前道路信息和交叉道路信息的值根据地图 DB 中的定义来设定。

4、沿路距离

```

INTERSECTION DISTANCE FIELD { 1byte 沿路距离
    intersection distance  8bit 到对象交叉点的沿路距离
}

```

该距离设定为从车辆当前位置在预测经路商的垂直投影点到对应交叉点的沿路距离, 并按照下面的方式记录:

值的范围: 0~255

数据长度: 8bit

单位: [m] (小数点以下省略)

当不存在对象交叉点或无法检出对象交叉点的情况下, 可以设定为任意值。

5、相对方位

INTERSECTION DIRECTION FIELD { 1byte 相对方位
 intersection direction 8bit 从车辆正面到对象交叉点相对方位
 }

车辆正面作为±0 度, 对象交叉点在车辆右侧设定为正数, 右侧设定为负数

值的范围: -180~179

数据长度: 8bit

单位: [度] ([degrees])

单位数据表示的含义: 360/256 度 (小数点以下省略)

例如、

相对方位 = -30° 时, 值设定为 -21。($-30 / (360 / 256) = -21.3333\dots$)。

相对方位 = 118° 时, 值设定为 84。($118 / (360 / 256) = 83.9111\dots$)。

进行这样的处理, 是为了满足 CAN 总线数据传输的需要, 使得能够使用 8bit 能够表示 360 度, 虽然这种表示方式会造成一部分的数值丢失, 但考虑到对相对方位值的精度要求并不高, 所以认为可以接受。

当不存在对象交叉点或无法检出对象交叉点的情况下, 可以设定为任意值。

6、道路曲率

INTERSECTION CURVATURE FIELD { 2byte 道路曲率
 intersection curvature 16bit 对象交叉点前方 50m 区间的道路曲率
 }

车辆行进方向为准, 向右弯曲设定为正数, 向左弯曲定为负数

值的范围: -0.32768~+0.32767

数据长度: 16bit

单位: [m^{-1}]

单位数据表示的含义: 10^{-5} (小数点以下省略)

例如, 曲率 = -0.01 (曲率半径 = -100m) 的情况下、值设定为 -1000 ($-0.01 / 10^{-5} = -1000$); 曲率 = 0.02 (曲率半径 = 50m) 的情况下、值设定为 2000 ($0.02 / 10^{-5} = 2000$)。

此外, 当曲率小于 -0.32768 的情况下 (under flow)、值设定为 -32768; 曲率大于 0.32767 的情况下 (over flow)、值设定为 +32767。

当不存在对象交叉点或无法检出对象交叉点的情况下, 可以设定为任意值。

7、标高差

LONGITUDINAL SLOPE FIELD { 1byte 标高差

longitudinal slope 8bit 车辆当前位置和对象交叉点或对象交叉点之间的
标高差
}

值的范围: -128~+127

数据长度: 8bit

单位: [m] (小数点以下省略)

车辆现在位置在预测经路上的垂直投影点的标高 $H(0)$, 对象交叉点#I 的标高记为 $H(i)$, 对象交叉点#I 的标高差依次定义为:

对象交叉点#I 的标高差 = $H(i) - H(i-1)$

当不存在对象交叉点或无法检出对象交叉点的情况下, 可以设定为任意值。

5.3 CANBus 总线数据传输的实现逻辑

在交叉点支援子系统中, 专门设定了用于与 CANBus 进行数据交互的 CBC 模块。该模块的作用就是分别从 DMC 取得制作完成的交叉点信息数据和从 DI 模块取得 Debug 信息并定时将这些信息输出到 CANBus 总线上。具体的工作流程可以参考图 5.3。

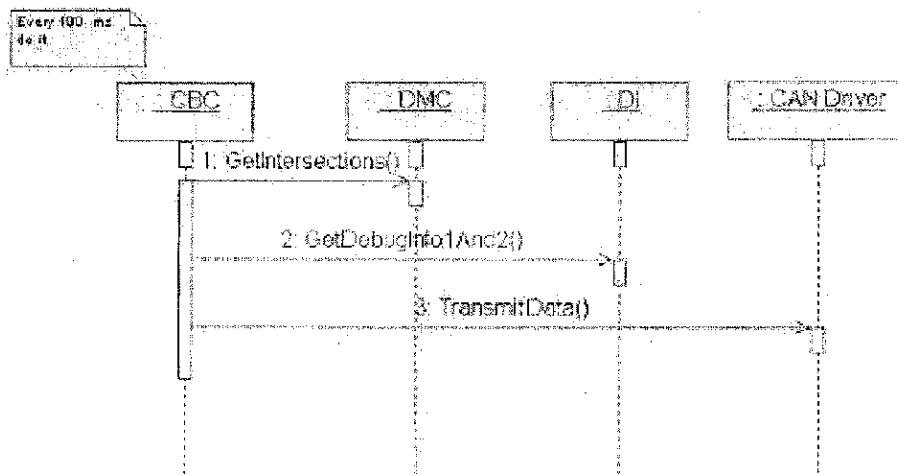


图 5.3 交叉点信息输出流程

Fig. 5.3 Sequence for Output Intersection Information

第六章 警告语音与警告图像

本系统中，当前方的交叉点存在潜在危险的时候，会通过警告语音和警告图像两种方式提醒驾驶者注意可能发生的危险。在本章将分别说明警告语音和警告图像的实现方法。

6.1 警告语音与警告图像的控制

交叉点支援子系统中的 WRS 模块负责警告信息的传递。当 PC 处理部根据导航设备输出的交叉点信息以及车辆信息判断出危险存在的时候，会通过 CANBus 总线将相关信息以 CommandCode 的形式传送到导航设备。

负责接收 CANBus 输入的是 CBC 模块，再由 CBC 模块将事件转发到 WRS 模块。WRS 模块的功能就是解析 CommandCode，再分别通知 VoiceControl 和 HMI 发出警告语音以及显示警告图像。根据警告级别的不同，警告语音和图像的内容也会有所不同^[23]。

警告语音与警告图像的控制流程可以参考图 6.1。

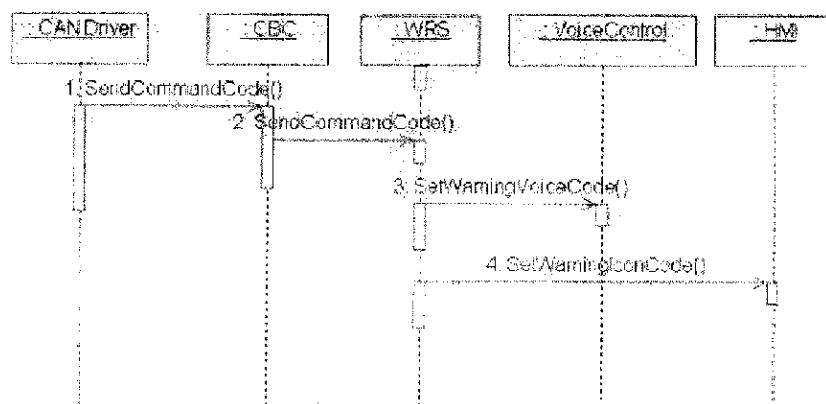


图 6.1 警告语音和警告图像控制流程
Fig. 6.1 Sequence for Control Warning Voice and Image

6.2 警告语音发声原理与实现

当应用程序判断出前方的交叉点存在潜在危险的时候，会用语音的方式提醒驾驶者注意。这里的语音并不是简单的硬件音，虽然硬件音更直接，但硬件发音过于简单，无法表达各种不同的状况。因此，导航系统中普遍采用实际语音作为发声手段^[24]。

现有导航系统的 Platform 中提供了支持 ADPCM 格式的语音接口，但为了能够满足

多个线程或进程间共同使用语音设备并且在不同的语音间不发生干扰，需要单独设立一个模块进行语音的管理和控制。

基于以上状况，在导航的 Service 层设立了一个 VoiceControl 模块进行语音的控制。

6.2.1 VoiceControl 的基本功能和内部构造

为了实现对语音的合理控制和管理，VoiceControl 模块需要具备以下一些基本功能：

- (1) 能够接受语音请求并对所有的请求进行管理；
- (2) 能够给不同的语音请求设定不同的属性，包括音量、优先级等；
- (3) 能够解析语音字符串形成语音数据并对语音数据进行管理；
- (4) 使用 Platform 提供的语音接口进行发声。

根据以上要求，将 VoiceControl 模块按照图 6.2 的方式进行了设计。

这里，VoiceControl 模块中包含 ControlManager 和 DataManager 两个部分。

ControlManager 负责语音控制包括外部请求的管理，语音设备的操作等，DataManager 负责将外部提供的语音字符串解析成 ADPCM 格式的语音数据提供给 ControlManager 发声时使用^[25]。

由于数据的读取以及语音数据格式的转换需要花费一定的时间，因此，为了满足语音的实时性，ControlManager 和 DataManager 被设计成两个不同的线程，这样在进行发声的同时可以进行其它语音请求的数据准备工作，可以有效地避免语音数据制作所带来的发声延迟。

ControlManager 内部又被划分为 MessageLibrary、SpeakController、BuffManager 三个子模块。MessageLibrary 实现 ControlManager 的消息管理，提供发送消息，接收消息等功能；SpeakController 管理发声请求，实现发声请求的注册，删除，查找等功能并实际通过 Platform 发声；BuffManager 管理缓存数据块，从 MessageLibrary 得到需要发声的语音字符串将其转发给 DataManager 制作语音数据并在发声时提供给 SpeakController 使用。

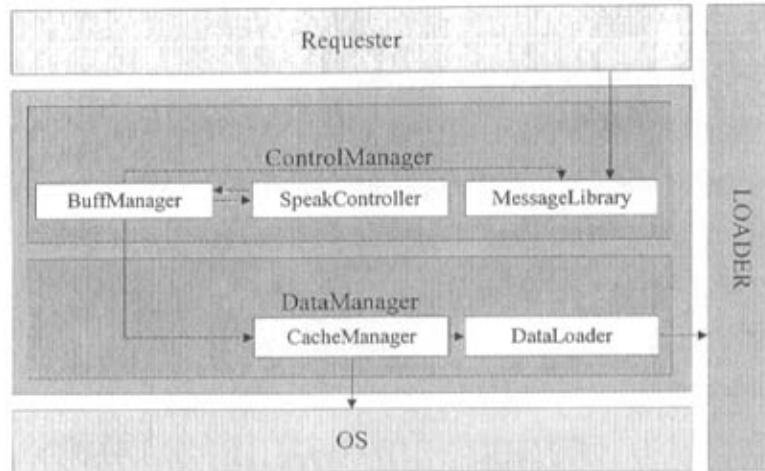


图 6.2 VoiceControl 模块内部结构与外部关联
Fig. 6.2 Internal Structure in VoiceControl Module and Extern Relation

DataManager 也根据职责的不同被划分为 DataLoader 和 CacheManager 两个部分。

DataLoader 负责通过 Loader 从地图数据中读取制作语音数据所必需的部分，将其进行必要的数据转换后提供给 CacheManager 进行存储^[29]。CacheManager 则负责语音数据的存储管理，把 DataLoader 从地图数据中读取的语音数据保存在从 OS 申请的内存中，并将这些数据在必要的时候提供给 ControlManager，同时负责在数据使用完毕后从 Cache 中删除。

对于 VoiceControl 模块的具体实现流程，请参考后面各节的详细说明。

6.2.2 VoiceControl 的实现方法

在 ControlManager 接收到外部发送的发声请求后，需要依次完成一系列的动作才能实现完整的发声过程。首先，从预留内存中取得本次发声所必需的数据缓冲 Buffer，然后请求数据缓冲，从地图数据中读取语音数据发到数据缓冲区中，然后才能正式的发出语音，发声之后，还要释放申请到的数据缓冲 Buffer。在整个过程当中，还要允许外部能够取消已经发出的发声请求，终止正在进行中的动作。

除了实际发声过程之外，还有一些附加的动作需要被关注，包括模块的初始化、音量控制、优先级设定等。

1、VoiceControl 的初始化

在系统启动的过程中，VoiceControl 模块也将被启动，届时 VoiceControl 的初始化将被实行，初始化的主要工作是 ControlManager 和 DataManager 两个任务的创建，并给这两个任务分配必要的系统资源（如 Event 和信号量等）。这一过程中 MessageLibrary、

SpeakController、BuffManager 以及 DataLoader 和 CacheManager 等子模块也将被初始化。同时，还要为数据缓存申请必要的预留内存。

2、数据缓冲区的取得

在发送发声请求之前，外部模块需要根据自身需求在 VoiceControl 内部申请一块数据缓冲区，之后在处理该模块所发出的发声请求时，就会使用这块数据缓冲区进行数据缓冲。数据缓冲区的取得可以在系统启动时一次性申请，也可以在使用之前进行申请，在申请缓冲内存量不大的情况下，使用前者比较方便。

缓冲区申请的主体流程包括：

(1) 外部模块设定好自己所需缓冲区的个数、大小、类型等属性，以 Message 的形式通知给 VoiceControl，同时外部模块进入等待状态，直到 VoiceControl 完成缓冲 Buffer 的分配。

(2) VoiceControl 接到请求后，首先检查当前预留内存中是否存在足够的缓冲 Buffer。

(3) 如果当前存在缓冲 Buffer 满足需要，将所找到的 Buffer 状态设定为等待状态，并向外部模块发送完成通知，通知的内容中包括所分配到的 Buffer 的块数以及表示该 Buffer 的 ID。

(4) 如果当前没有足够的缓冲 Buffer，VoiceControl 会在一定时间内定时地检查预留内存的状态，直到找到足够的缓冲 Buffer，如果经过长时间等待仍然无法满足需要，则返回失败。

通常情况下，如果发现存在缓冲 Buffer 大小不足的情况发生，可以通过调整 VoiceControl 预留内存的形式进行调整。而且作为预防措施，预留内存可以申请的略大一点以避免这种情况的出现。

3、语音数据的制作

外部模块取得数据缓冲区的使用权之后，就可以根据不同的时机发送发声请求。由于导航系统的发声时机通常是可控的，为了使语音能够及时地发出，VoiceControl 允许在实际发声之前提前一段时间进行语音数据的制作。

语音数据制作的主体流程包括：

(1) 外部模块首先可以根据语音字符串的长度取得发声时间长度的估计值，并利用该时间长度确定发送语音数据制作请求的时机。

(2) 外部模块向 VoiceControl 发出语音数据制作的请求。外部模块发出请求后不

需要等待数据制作完成，可以继续进行其他工作。

(3) ControlManager 接收到数据制作请求后，根据外部模块提供的 BufferID 找到给该模块使用的数据缓冲区，并在该缓冲区中找到处于等待状态的缓冲 Buffer。

(4) ControlManager 将缓冲 Buffer 传递给 DataManager 并请求 DataManager 制作相应的语音数据。同时，将缓冲 Buffer 的状态设定为缓冲进行中。

(5) DataManager 根据实际语音信息，通过解析电子地图语音数据中的管理信息，找到相应的语音数据并将其读入到缓冲 Buffer 中。在这个过程中，DataManager 需要通过 Loader 进行地图数据的交互。

(6) DataManager 完成数据读取后通知 ControlManager 数据制作完成，VoiceControl 在接受到结束通知后，将对应的缓冲 Buffer 的状态设定为缓冲结束。

经过以上步骤后，语音数据已经存在于缓冲 Buffer 中，只要接受到外部模块发出的发声请求，就可以进行发声操作了。

4、实际发声请求

当到达实际需要发声的时机，外部模块会发送发声请求，VoiceControl 会在接受到请求后进行实际的发声动作。考虑到可能会同时接收到来自不同外部模块的发声请求。因此，VoiceControl 中将所有的发声请求放到一个队列中同一管理，如果发现存在发声请求，会对队列中的请求按顺序依次进行处理。

处理发声请求的主体流程包括：

(1) 首先判断当前 VoiceControl 的状态是否适合进行发声，主要是为了避免硬件设备没有准备好或者 VoiceControl 的初始化没有完成的情况下接收到发声请求可能会导致的异常处理。

(2) ControlManager 从消息队列中取出一个请求，从数据缓冲中查找和该请求相对应的数据缓冲 Buffer，如果该 Buffer 的状态时未使用状态或进行中的状态，需要向 DataManager 发送数据制作的请求或等待数据制作完成。

(3) 根据请求设定硬件设备的发声音量，并调用 Platform 提供的接口进行实际的发声操作。

(4) 发声结束后需要释放和该发声请求对应的各种资源，包括从信息队列中删除处理完的请求并释放缓冲 Buffer。

5、取消发声请求

外部模块可以根据需要取消已经发出的发声请求，VoiceControl 除了需要中止当前

正在进行的语音外，对一些正在进行数据准备的请求也需要进行处理。因此，取消发声请求的动作不仅与 ControlMananger 相关，也可能和 DataManager 发生联系。

在此过程中需要关心的环节包括：

- (1) 如果当前的消息还没有被处理，需要从 ControlManager 的消息队列中删除相应消息。
- (2) 如果发声已经开始进行，需要根据要求来确定是否中断当前的语音。
- (3) 需要检查是否存在缓冲结束的缓冲 Buffer，如果存在，需要将这部分缓冲数据从缓冲区中删除。
- (4) 如果正在进行相关语音数据的缓冲，需要通知 DataManager 停止当前的数据制作。

6、数据缓冲区的释放

在发声结束之后，外部模块可以释放申请到的数据缓冲区。在申请量不大得情况下可以放到系统结束时进行释放。

释放的过程中需要确认当前数据缓冲区中的 Buffer 是否都处于未使用状态，如果仍然存在正在使用中的 Buffer，需要提示外部模块进行错误处理。

7、提供给外部模块使用的其他功能

(1) 语音音量的设定。VoiceControl 为整个导航系统设定了一个默认的音量，但同时也允许外部模块修改默认音量的值，也可以设定为不使用默认的音量而是用外部模块提供的音量值。

(2) 语音优先级的设定。VoiceControl 给语音设定了不同的优先级，在 VoiceControl 的信息队列中，各种发声请求是按照优先级的高低来排序的。优先级较高的优先级可以选择等待当前语音完成或者中断当前的语音从而先被响应。

6.2.3 关于警告语音字符串的定制

针对交叉点支援子系统的实际情况，需要以最快的方式向驾驶者提供警告语音。因此在定制警告语音字符串的时候将发声的优先级设定为最高，警告语音的数据准备和发声都优先于其他类型的语音，即使同时存在其他进行的语音，也允许警告语音能够中断当前语音。

6.3 警告图像的显示实现

一旦前方的交叉点存在潜在危险，在通过语音提醒驾驶者注意的时候，还要求通过

图像给予驾驶者一定的提示作用, 警告图像包括警告图形和警告信息(文本)两个部分。

这样做的原因是在很多情况下, 单纯的语音可能会被其他的背景噪音干扰, 可能会被驾驶者忽视。警告图像的显示是由 UserInterface 层(HMI)来实现的, 与 UserInterface 的其他显示内容相比, 被赋予了更高的优先级。

6.3.1 HMI 的基本内部结构

HMI 是导航功能的表示层, 在整个系统中, 它处于最上面的位置, 图 6.3 中显示了导航系统构架中 HMI 所处的位置。在交叉点支援子系统中, 警告图像最终将被显示在 Form 上。

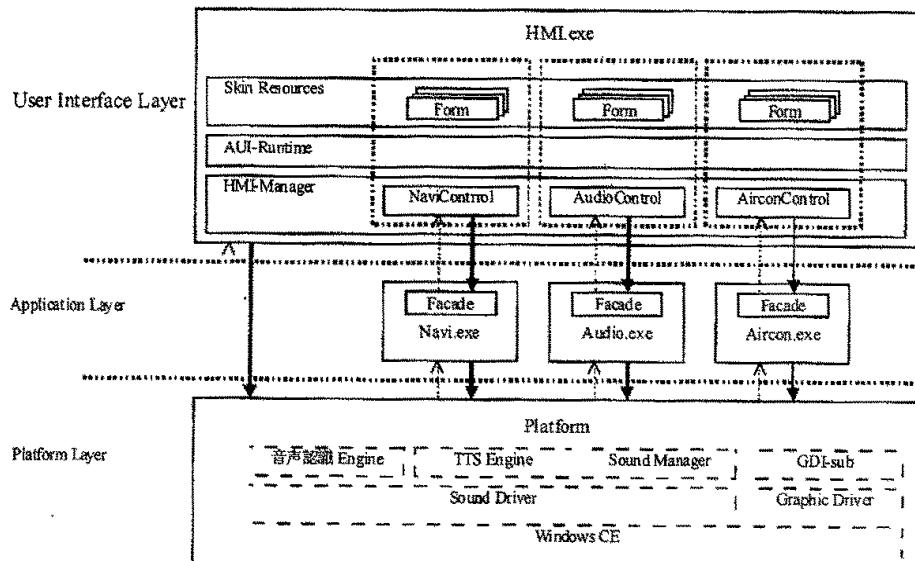


图 6.3 HMI 在车载导航系统的位置
Fig. 6.3 Position of HMI in Vehicle Navigation System

6.3.2 Form 的组成

Form 中显示的内容统称为部件, 其中主要的部件包括: Button, PictureBox, Label, Control, Form 等^[33]。

Button: 即按钮, 类似于对话框中的 Button 控件。用户可以通过点击触摸屏上显示的按钮来对导航机进行各种操作。Button 的形状、大小、位置以及 Button 上描画的图形和文字都可以根据产品的 Design 和用户的需求来定制。

PictureBox: 就是用来显示各种图像的区域, 类似于图形框。PictureBox 根据用途的不同可以被定制成不同的类型。既可以显示既定的图片、照片等, 也可以是一块透明的区域, 使得地图等描画的内容能够透过表层界面展现在用户面前。

Label: 实际上就是文本框。可以设定为不同的字体，并指定文本框中能够显示的字符长度。

Control: Control 就是控件，以上提到的 Button、PictureBox、Label 都是控件，这些信息被保存在 XML 文件中，应用程序通过解释 XML 文件中的信息将各种控件显示到屏幕的适当位置。

Form: Form 是各种控件的集合，不同的 Form 组合在一起就形成了用户看到的各种各样的画页，共同构成导航系统的人机交互界面。Form 通过接受用户的各种动作，可以将指令传达给应用层，同时也可将应用层制作的各种数据和计算结果显示在各种控件上传达给用户。

警告图像中图像的部分属于控件中的 PictureBox，而警告 Message 属于控件中的 Label。

6.3.3 Form 的实现

Button、PictureBox 和 Label 都是基本的部件，实现上简单易懂。Form 上只描述静态的内容，例如部件的状态，部件在 Form 中的位置，是否需要图片修饰，是否可见等等。而动态的变化是通过 AppliControl 传递的属性来实现的。下面是以 XML 形式定义各种控件的基本格式^[35]:

```
<Button Id="BC_Navi003_ReadingBtn" Base="BC_Button_FUNC5">
    <Type>Push</Type>    //Button 的类型是否存在 Push 状态
    <Height>36</Height>  //Button 所在 Form 中的位置，高度
    <Top>195</Top>       // Button 所在 Form 中的位置，左上的位置
    <Image>BMP_SaikaiSw</Image>           //图片修饰
    <Visible>AX_NAVICTL.ScrollMovement</Visible> //Button 是否可见
    <ActionEvent="DisableButtonDown">ACT_SetEvent(BC_Navi003_ReadingBtn,BEEP_ERRO
R,TSW_TELM_BRWS_SPEECH,0)</Action> //Button 状态为不可点击的时候触发的事件
    <ActionEvent="Click">ACT_SetEvent_P1(BC_Navi003_ReadingBtn,BEEP_ENTER,TSW_TE
LM_BRWS_SPEECH,0)</Action> //Button 状态为可点击的时候触发的事件
</Button>

<PictureBox Id="IC_Navi003_Cross_9" Base="IC_Img">
    <Left>229</Left>
    <Top>107</Top>
    <Width>21</Width>
```

```

<Height>20</Height>
<Image>BMP_icon_mapscroll_9</Image>
<Visible>AX_NAVICTL.ScrollMovement==1</Visible>
</PictureBox>

<Label Id="TC_Navi003_NowPosition" Base="TC_Lbl">
  <Left>341</Left>
  <Top>178</Top>
  <Width>123</Width>
  <Height>12</Height>
  <Font>FONT_MSGOTHIC12_W</Font> //Label 上显示文字的字形定义
  <Caption> AX_NAVICTL.FileNum</Caption> //Label 上显示的文字的内容
  <Visible>1 </Visible>
</Label>

```

Control 增加了 Form 上实现部件的多样化和灵活性。Control 的实现除了上述基本的内容之外，还有不同于其他部件的特殊性——需要 CustomDraw 的支撑^[37]。

(1) Form 上的实现

```

<Control Id="AX_Navi064_Mark11">
  <ProgID>DrawIcon.DrawIconControl.1</ProgID>
  <Left>74</Left>
  <Top>72</Top>
  <Width>16</Width>
  <Height>16</Height>
  <BitmapInfo>AX_NAVICTL.IconInfoSource(0)</BitmapInfo>
  <pIconData>AX_NAVICTL.IconData(0)</pIconData>
</Control>

```

(2) CustomDraw 的支撑

CustomDraw 的各个 Control 包含在组件服务器 CustomDraw.dll 中，通过 CustomDraw.tlb 提供接口。

CustomDraw 中的属性多数都是通过 xml 取得的，xml 中实例化组件的时候会 put 相应的属性，下面以 DrawIconControl 为例说明属性的取得过程^[41]。DrawIconControl 中有两个属性：BitmapInfo 和 pIconData，这两个属性是由 Form 来赋值的，如上述代码中：

```

<BitmapInfo>AX_NAVICTL.IconInfoSource(0)</BitmapInfo>
<pIconData>AX_NAVICTL.IconData(0)</pIconData>

```

在这个组件被实例化的时候会调用 DrawIconControl 的 put_BitmapInfo 和

put_pIconData 给属性赋值，对应的代码如下：

```

for (LONG lCnt = 0; lCnt < 30; lCnt++) {
    m_pIconInfoSource[lCnt] = NULL;
    m_pIconData[lCnt] = NULL;
    lRet = NAVILIB_GUIDE()->GetIconMemAddress(
        USERMARK_TYPE_1,
        m_pIconId[lCnt],
        &m_pIconInfoSource[lCnt],
        &m_pIconData[lCnt]);
    if (S_OK != lRet) {
        return E_FAIL;
    }
    if( m_pIconId[lCnt] == lIconID ){
        NAVICTL()->put_MarkKind(lCnt);
    }
    NAVICTL()->put_IconInfoSource(lCnt, (long)m_pIconInfoSource[lCnt]);
    NAVICTL()->put_IconData(lCnt, (long)m_pIconData[lCnt]);
}
NAVICTL()->FireAUIconData();

```

DrawIconControl 的属性：BitmapInfo 和 pIconData 的值就是这样赋给 Form 的。

根据数据取得的方式不同，CustomDraw 中常用的描画方法分为以下两种：

- (1) 利用资源 bmp 图来描画；
- (2) 直接利用底层提供的数据来描画；

无论采取哪种方法，最终都要调用 API 函数 DDBBitBlt 或 DDTransparentImage 把保存在 Cache 中的 pixels 传输到显示设备上，在这点上他们是相同的，其中 DDTransparentImage 函数会把某种指定的颜色滤过。DDBBitBlt 和 DDTransparentImage 的使用方法可以查阅 MSDN^[45]。

这两种方法的不同点在于取得 HBITMAP 句柄的方式不同：利用资源 bmp 图来描画的，是通过 API 函数 LoadBitmap 取得 HBITMAP 句柄，而直接利用底层提供的数据来描画的，是通过 API 函数 CreateDIBSection 来取得 HBITMAP 句柄。关于这两个函数的使用以及函数中参数的说明请查阅 MSDN^[46]。

6.3.4 警告图像的显示效果展示

通过前面的处理，警告的图像可以显示成如下图所示的情况。根据式样要求的不同，其中的 Message 的大小、位置、内容可以随意进行变更；图像既可以是小的 Icon，也可

以作成大的图片。

图 6.4 就是导航画面上显示警告 Message 时的效果图。



图 6.4 警告 Message 的显示
Fig. 6.4 Display of Warning Message

第七章 调试工具的展示

由于交叉点支援子系统的研究与开发还处于试验阶段，为了更方便的进行调试，使操作者能够更容易更清晰的看到当前系统内部的工作状况并判断当前处理的结果是否符合实际情况，在子系统开发的同时，使用 MFC 开发了一套专门用于调试的工具。

由于在该工具中也通过加载导航模块的方式实现了基本的地图描画等功能，因此可以清晰的看到当前车辆的位置，并以图形和文本的方式表示当前制作出来的交叉点信息。

通过文本方式表示出来的信息大体上与导航设备通过 CANBus 总线输出的内容一致，同时用图形的方式对标高、方位、曲率等内容进行表示，使调试者对现状的认识更直观。

(1) 调试工具的调试面板

通过调试面板可以指定调试工具中的哪些视图被显示出来，如图 7.1 所示。



图 7.1 调试工具的调试面板
Fig. 7.1 Control Panel for Debug Tool

(2) 可视化地图 View

利用现有的地图数据在 PC 机上进行了地图的模拟描画，这样可以清晰的看到当前车辆的位置，走行方向以及在此过程中导航系统发现的对象交叉点，如图 7.2 所示。



图 7.2 可视化地图 View
Fig. 7.2 Visible Map View

(3) 详细的交叉点信息

除此之外，也输出了包括车辆位置、匹配状态等需要关心的信息，如图 7.3 所示。

图 7.3 详细的交叉点信息
Fig. 7.3 Detail Intersection Information

(4) 标高信息展示

采取图形的方式将前方多个交叉点的标高信息显示出来，如图 7.4 所示。

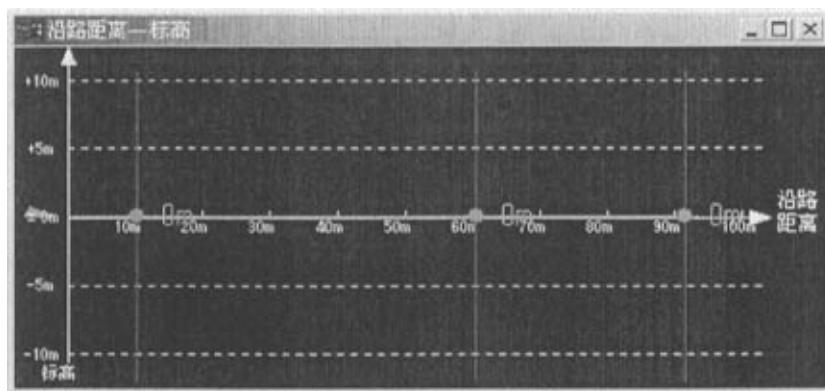


图 7.4 标高信息展示
Fig. 7.4 Show the Relative Altitude

(5) 道路曲率展示

采取图形的方式将每个对象交叉点前方的道路曲率显示出来，如图 7.5 所示。

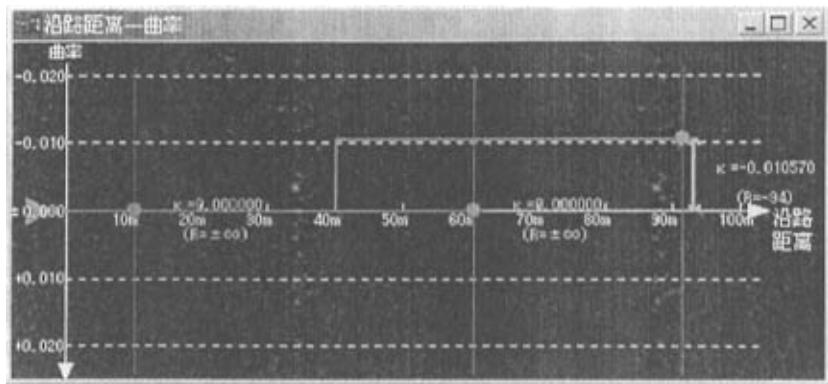


图 7.5 道路曲率展示
Fig. 7.5 Show the Road Curvature

(6) 交叉点相对方位展示

采取图形的方式将每个对象交叉点相对于当前车辆行驶方向的方位显示出来, 如图 7.6 所示。

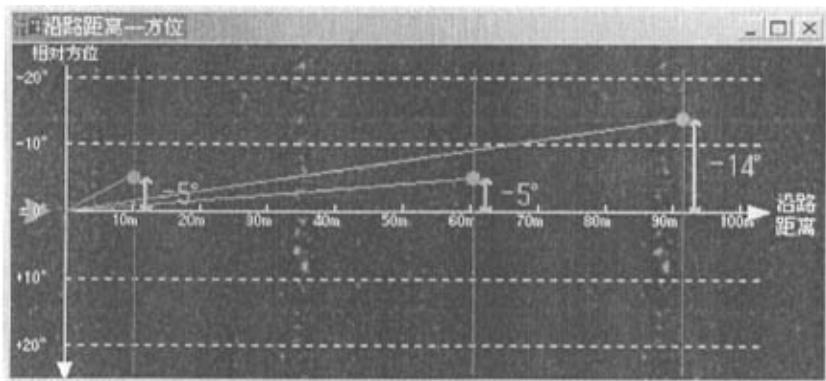


图 7.6 交叉点相对方位展示
Fig. 7.6 Show the Relative Direction

(7) 警报信号模拟 Panel

使用该 Panel 可以模拟发现危险后向导航系统发送的警告信号, 导航系统接收到警告信号后会显示相应的警告图像并发出警告语音, 如图 7.7 所示。



图 7.7 警报信号模拟 Panel
Fig. 7.7 Warning Signal Simulation Panel

第八章 结束语

车载导航系统已成为目前国际交通领域的主要研究方向，在我国，近两年车载导航系统产品业不断涌现，市场的竞争业逐渐激烈。其中，车载导航系统中除了导航基本功能之外的功能扩展越来越成为研究的重点。

本文结合了当前交叉点辅助功能中一个比较有现实意义的方向展开研究并提出一系列可行的方案并加以实现，取得了比较好的实际效果。

在该系统开发中，本人主要完成了以下任务：

- (1) 完成了经路预测和交叉点信息制作的设计与实现。
- (2) 深入的调查了当前导航系统中的语音发声机制并通过改善实现了警告音声相关的功能。
- (3) 深入的调查了当前导航系统中的人机交互的结构以及界面控制和显示的实现方法并在此基础上实现了警告图像和警告 Message 相关的功能。
- (4) 了解基于 CANBus 的数据传输机制并实现了导航设备和 PC 之间的数据通信。

通过参与交叉点支援系统的开发，使我对导航功能及其内部结构有了更加深入地了解，尤其是在基于 Windows CE 的嵌入式导航系统方面能够深入和系统地把握整个开发环节。同时，又开阔了视野，接触到了更多汽车导航相关的扩展领域。

目前，我所参与研究的交叉点支援系统已经通过了汽车导航生产厂家的项目验收，虽然在目前的研究成果中还有很多内容需要完善和扩充，但可以期望在不远的将来，必将会成为汽车导航系统中一个重要的角色。

参考文献

1. 沈连丰, 宋铁成, 叶芝慧. 嵌入式系统及其开发应用[M], 北京:电子工业出版社, 2005.9, 1-29
2. 陈文智. 嵌入式系统开发原理与实践[M], 北京:清华大学出版社, 2005.5, 1-114
3. 贾志平, 张瑞华. 嵌入式系统原理与接口技术[M], 北京:清华大学出版社, 2005.6, 22-58
4. Michael Barr 著, 于志宏译. C/C++嵌入式系统编程[M], 北京:中国电力出版社, 2001.3, 34-95
5. 车载导航信息产业研究分析报告[J], 博通智信咨询公司, 2005.3, 1-25
6. 张可. 车辆导航系统关键技术研究[D], 北京:北京工业大学, 2001, 17-23
7. 2004-2005 年中国汽车通讯与导航产业研究报告[J], 水清木华研究中心, 2005.9-57
8. 苏彦华. Visual C++数字图像识别技术典型案例[M], 北京:电子邮电出版社, 2004.8, 27-35
9. 杨淑莹, 李兰友. 图像模式识别——VC++技术实现[M], 北京:清华大学出版社, 2005, 53-63
10. 王耀南, 李树涛, 毛建旭. 计算机图像处理与识别技术[M], 北京:高等教育出版社, 2005.2, 3-10
11. 沈庭芝, 方子文. 数字图像处理及模式识别[M], 北京:北京理工大学出版社, 1998.6, 34-57
12. 饶云涛, 邹继军. 现场总线 CAN 原理与应用[M], 北京:北京航空航天大学出版社, 2003.6, 8-24
13. 王箴. CAN 总线在汽车中应用[J], 中国汽车报, 2004.9.20:8
14. 李映. 汽车业迎接 CAN 总线应用时代[J], 中国电子报, 2004.9:24.
15. 龚进峰. 浅析汽车总线技术[J], 智能交通研究, 2003.7:1-4.
16. 张平. 中国汽车电子产业初探[J], 信息技术快报, 2004.10:4.
17. 邬宽明. CAN 总线原理和应用系统设计[M], 北京:北京航空航天大学出版社, 2001.3, 10-30
18. 郑佳春. 车载电子地图系统的研究[D], 大连:大连海事大学, 2000, 23-34
19. 张其善, 吴今培, 杨东凯. 智能车辆定位导航系统及应用[M], 北京:科学出版社,

- 2002, 19-32, 163-169
20. Bruce Eckel. C++编程思想 第二版[M], 北京:机械工业出版社, 2002, 3-288
21. Cameron Hughes. C++面向对象多线程编程[M], 北京:人民邮电出版社, 2003.5, 35-88
22. Nell Dale, Chip Weems, Mark Headington. Programming in C++ (Second Edition)[M], 北京:高等教育出版社, 2001, 45-78
23. Grady Booch. Object-Oriented Analysis and Design with Applications(Second Edition)[M], 北京:中国电力出版社, 2003, 28-37
24. Elliotte Rusty Harold. XML 实用大全[M], 北京:中国水利水电出版社, 2000, 15-67
25. Bruce Eckel. C++编程思想 第二版[M], 北京:机械工业出版社, 2002, 3-288
26. Dino Esposito. Applied XML Programming for Microsoft .Net[M], 北京:机械工业出版社, 2003, 134-158
27. Wendy Boggs, Michael Boggs. UML with Rational Rose 从入门到精通[M], 北京:电子工业出版社, 2000, 4-120
28. Ben Shneiderman. Designing the User Interface Strategies for Effective Human-Computer Interaction[M], 北京:电子工业出版社, 2003, 1-60
29. Ronald.J.Norman. Object-oriented Systems Analysis and Design[M], 北京:清华大学出版社, 2002, 66-82
30. 张海藩, 牟永敏. 面向对象程序设计实用教程[M], 北京:清华大学出版社, 2001, 40-56
31. Robert V.Sumpf, Lavette C.Teague. Object Oriented Systems Analysis and Design With UML (Hardcover)[M], 北京:清华大学出版社, 2005, 1-47
32. Don Box, John Lam, Aaron Skonnard. Essential XML[M], 北京:科学出版社, 2003, 10-30
33. Christine Hofmeister, Robert Nord, Dilip Soni. 实用软件体系结构(影印版)[M], 北京:电子工业出版社, 2003, 1-60
34. 石伟伟, 王康宏. 以 GIS 为核心的数字化成图系统的设计与实现[J], 2001 中国 GIS 年会论文集, 2001, 3(1): 2-24
35. Felix Redmill, Chris Dale. 计算机系统的可信性实践[M], 北京:清华大学出版社, 2003, 15-20
36. Ben Shneiderman. Designing the User Interface Strategies for Effective Human-Computer Interaction Third Edition[M], 北京:电子工业出版社, 2004, 58-114

37. Sandy Tyndale-Biscoe, Oliver Sims. Business Modeling for Component Systems with UML[J]. Proceedings of the 6th International Enterprise Distributed Object Computing Conference, 2002:120-131
38. 杨斌, 田永青, 朱仲英. 基于 UML 的面向对象的软件系统分析、设计与开发技术[M], 北京:电子工业出版社, 2002, 5-8
39. 胡海静. XML 技术精髓[M], 北京:机械工业出版社, 2002, 1-70
40. 微软公司. XML 技术内幕[M]. 北京:清华大学出版社, 2001, 1-70
41. Yali Zhu, Gang Huang, Hong Mei. Modeling Architecture Based Development in UML[J], 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05), <http://books.google.com/>, 2005:56-65
42. Yinghua Ma, Richard Chbeir. Content and Structure Based Approach For XML Similarity[J], The Fifth International Conference on Computer and Information Technology (CIT'05), <http://books.google.com/>, 2005:136-140
43. Sunghwan Roh, Kyungrae Kim, Taewoong Jeon. Architecture Modeling Language based on UML2.0[J], 11th Asia-Pacific Software Engineering Conference (APSEC'04), <http://books.google.com/>, 2004:663-669
44. Yuichi Ohta, Pattern Recognition and Understanding for Visual Information Media[J], 16th International Conference on Pattern Recognition (ICPR'02) - Volume 1, 2002.8
45. Atsushi Maruyama, Naoki Shibata, Yoshihiro Murata, Keiichi Yasumoto, Minoru Ito, A Personal Tourism Navigation System to Support Traveling Multiple Destinations with Time Restrictions[J], 18th International Conference on Advanced Information Networking and Applications (AINA'04) Volume 2, 2004.3:4-39
46. Hassan Reza, Emanuel Grant. Model Oriented Software Architecture[J], 28th Annual International Computer Software and Applications Conference - Workshops and Fast Abstracts - (COMPSAC'04), <http://books.google.com/>, 2004:4-5
47. Dennis de Champeaux. The OOA process[J], Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum) (OOPSLA '92), 1992.10:99-107

致 谢

首先感谢孙萍老师和张石老师，尽管两年多的研究生生活和老师的接触不是很多，但是每次的交流都能感受到老师的真诚、率直，尤其是他们的严谨和勤奋都深深的影响了我，也非常感谢他们对我的无私帮助。

同时，要感谢部门的孙克文部长，他为我的创造了一个良好的工作环境和氛围，并为我的研究提供了很多宝贵的意见和提供了实习的机会。

感谢朱育松副局长，是他给了我能够研究现在的领域的机会。还要特别感谢部门项目总监陈洪涛和赵永茂，他们在我的研究中曾经多次和我进行讨论，并对我的一些研究提出了很多宝贵的意见。

感谢我们部门的所有同事在日常工作生活中对我的帮助关心与照顾，是他们在技术上给予我很大的支持，才能使我在技术上取得显著的提高。感谢他们在生活中给我的帮助，在学习中给我的指导和建议。

感谢一直支持和爱护我的父母，无论在什麼时候，他们总在我的背后默默的支持我，给我足够的勇气和信心让我去面对人生每一次挑战。

感谢在我成长过程中给予我关爱和帮助的所有的人！