
摘要

随着交通拥挤和堵塞等各种问题的日益突出，以及计算机技术的不断发展，智能交通系统(Intelligent Traffic System, ITS)得到了越来越广泛的应用，其中车型分类技术是重要的一个分支。本论文对基于图像识别的车型识别系统进行了研究，通过对已有的车型识别技术经验的研究和分析，实现了一个基于图像的车型识别系统。

本文首先将拍摄到的车辆图像进行预处理，把车辆图像分割出来；然后提取出完整的车辆外轮廓曲线，最后对轮廓曲线进行分析，通过提取出的汽车的特征信息与设定好的汽车的特征信息进行比较，利用最近邻法进行识别，得出识别结果。实验结果表明，该系统具有较好的识别率和鲁棒性。

关键词：智能交通；车型识别；车型分类；图像处理

Abstract

With the traffic congestion and blockage issues become increasingly prominent, and continuous development of computer technology, intelligent transport system (Intelligent Transportation system, ITS) has been more widely applied. The vehicle classification is an important branch of ITS. In this paper, a vehicle recognition system based on image is studied. Though analysis and research the technical experience of the previous research, we developed a vehicle recognition system based on image.

In our system, the vehicle image is preprocessed in first, and the vehicle is segmentation from the image which been preprocessed. Secondly, the contour of vehicle is extracted and analyzed. Finally, the feature of vehicle model is extracted, and the nearest neighbor classier method applied to classify. The effort of experiment show that the system have good classify effort and robustious characteristic.

Keywords: Intelligent traffic system; Vehicle classification; Vehicle Category; Image processing

目录

1 绪论.....	1
1.1 课题研究背景和意义	1
1.2 基于图像识别的车型分类系统研究现状	2
1.3 本文的主要研究内容和结构安排	4
2 车型识别系统的总体设计.....	5
2.1 基于 OPENCV 的车型识别系统设计	5
2.2 数字图像处理技术	6
2.3 本系统用到的图像处理技术	8
2.4 OpenCV	10
3 基于 OPENCV 的车型系统的实现.....	12
3.1 OpenCV 环境配置	12
3.2 图像的读取	14
3.3 图像的边缘检测	15
3.4 区域分割	17
3.5 图像的轮廓提取	18
3.5.1 轮廓方向序列的计算(T 算法).....	18
3.5.2 轮廓起始点的寻找(IP 算法).....	19
3.6 车辆的特征提取	20
3.7 分类识别	22
4 实验结果与处理.....	24
4.1 小轿车的车型识别	24
4.1.1 图像的轮廓提取.....	25
4.1.2 图像的腐蚀.....	25
4.1.3 车型的识别.....	26
4.2 小货车的车型识别	26
结 论.....	29
社会经济效益分析.....	30
参考文献.....	31
致 谢.....	32
附录 程序表.....	33

1 绪论

1.1 课题研究背景和意义

近年来,随着社会经济的发展,公路网通过能力已经不能满足交通流量日益增长的需要,经过长期和广泛的研究,世界各发达国家逐渐开始采用高新技术来改造现有的公路交通运输系统及其管理体系,从而大幅提高了公路网的通行能力和服务质量。随着人工智能、自动控制和模式识别等领域的发展,智能交通系统(Intelligent Traffic system, 简称 ITS)应运而生,并获得了极大的发展。ITS 包括了智能和交通两个方面。它将先进的人工智能、信息技术和自动控制技术等综合应用于整个交通运输管理体系,通过对交通信息的实时采集、传输和处理,对各种交通情况进行协调,建立起一种实时、准确、高效的综合运输管理体系,从而使交通设施得以充分利用,实现交通运输的集约式发展。ITS 系统是 21 世纪世界公路交通的发展趋势,90 年代中期我国学者也开始在相关领域开展大量的研究。

计算机视觉和模式识别技术的不断发展,为以图像理解为基础 ITS 系统进入实际应用提供了契机。计算机视觉就是用计算机来模拟人的视觉功能,从客观事物的图象中提取信息,进行处理并加以理解,最终用于实际检测、测量和控制等。计算机视觉不仅能使机器感知环境中的几何信息,包括位置、大小、形状、运动等,还能对它们进行描述、解释和理解。计算机视觉为交通系统提供了直观方便的分析手段,交通环境中的大量信息,如车辆、交通标志和路面标识等都来源于视觉。

基于图像识别的车辆分类系统正是在这种背景下提出来的。车辆检测和车辆分类在 ITS 应用领域中重要的分支,也是目前相对薄弱的技术环节。基于图像识别的车辆分类是一门集计算机视觉、图像处理、模式识别、工业测控技术、电子技术和系统工程技术与一体的综合技术,引入了计算机信息管理的思想,它能够自动、实时地对车辆进行检测和分类,作为交通管理、收费、调度和统计的依据。

目前,车型自动分类系统的主要应用场合为停车场、路桥收费管理和公安交通管理部门。其中停车场和路桥收费管理部门可利用该系统实现车型的自动识别及收费工作。车型自动分类系统识别车辆的车型,按照事先确定的收费标准确定车辆应收费额,自动保存收费记录,自动费用结算。这种自动收费方式可在不停车的情况

下完成收费工作，达到提高车速和道路通行能力的目的，而且可以杜绝工作人员询私、贪污等行为。交通管理部门则可以利用该系统实现无人值守警察功能，实时抓拍违规车辆，结合相关数据库还可实现未年检车辆稽查、违章未交罚款、特定车辆查询等功能。

1.2 基于图像识别的车型分类系统研究现状

基于图像识别的车型分类过程主要包括两个阶段：1、从图像中分割出目标车辆。2、提取车辆特征并进行分类。针对第一个阶段的算法主要利用图像序列之间存在的大量相关信息进行车辆区域定位和分割。第二阶段的算法比较多，主要可分为三类：1、简单参数分类法；2、模板匹配法；3、神经网络分类法。车型分类第二阶段的工作主要涉及车辆特征的提取和分类器的设计，而车辆特征的选取会影响到分类的精确程度，分类器的设计是否合理则直接决定了最终的分类结果。因此，一些学者针对第二阶段展开研究并取得了一定的成果。

下面介绍目前存在的部分研究成果：

有的学者采用一种基于三维结构模型的车辆特征提取方法，为车辆建立一个参数模型，然后以改进的 BP 神经网络作为车型分类器，采用梯度下降法训练网络，网络输入为 30 个特征参数。该文提出的参数模型能够细致地描述车辆的形状结构特征，但只考虑了固定简单背景下的情况，而且没有研究如何从图像中分割出完整的目标车辆区域。有的研究成果以顶长比(顶篷长度与车辆长度之比)、顶高比(顶篷长度与车辆高度之比)和前后比(以顶篷中垂线为界，前后两部分之比)三个车辆几何尺寸作为特征参数，将车辆划分为卡车、客车和货车三种类型。并设计了一种改进的 BP 算法(尺度共扼梯度法)对神经网络分类器进行训练，解决了一般算法收敛慢的问题。该研究工作主要解决的是上述第二阶段的问题，没有对目标车辆区域的分割作出研究。

有的研究利用背景消减法进行运动车辆区域的分割，并提出了一种基于 Gabor 滤波器和边缘特征的车型识别方法。与传统方法相比，提出一种新的基于车辆边缘的非均匀采样策略。首先将图像中长度大于某一确定值的边缘定为关键边缘，然后将取样窗口根据包含关键边缘像素的多少进行降序排列，把一定比例的取样窗口定位 KSW (Key Sampling windows)和 ASW (Assistant sampling Windows)，在 KSW 和 ASW 中采用不同的采样间隔进行样点的抽取，最后计算低维的特征矢量，并将得到的 Gabor 特征集与模板特征集进行匹配，从而实现车型分类。该方法只适用于几何

特征明显的物体识别，不能进行车型的细分。而且采用模板匹配法进行分类需要较多的计算时间。基于图像识别的车型自动分类系统

还有学者以车辆前上方的图像作为处理对象，利用车牌位置信息确定车头位置，并对车头部分的保险杠所在位置进行测量，以保险杠的长度作为车辆的宽度，同时找出车顶位置，确定车辆高度；然后找到车尾位置，结合车头的位置和摄像头拍摄的角度计算车辆的长度；最后采用 SVM (Support Vector Machine) 分类器对三个参数进行分类。该文把车辆的绝对长度、宽度和高度作为分类的依据，但对像轿车和吉普车这类车，只用三个参数并不能完全描述其形状，只能将车辆大致划分为大型、中型和小型三种类型，而且从车辆前上方确定其长度和高度存在很大的困难，对图像质量要求较高。

也有研究是利用一种模糊模式识别方法对车型进行分类。首先从车辆上方、和侧面两个角度采集车辆图像，通过分割和轮廓跟踪得到车辆轮廓图，然后采用模糊识别技术对车型进行分阶段识别。第一阶段以车辆的顶长比和前后比作为特征参数，利用模糊支持向量机实现客车和货车的分类；第二阶段利用车辆的长度、高度和宽度作为特征参数，利用模糊神经网络建立客车的识别模型，用遗传算法来同时优化隶属函数和推理规则，得到了稳定精简的识别模型，从而把客车划分为大型、中型和小型三类。该图像采集，而且采用车辆的绝对几何尺寸作为特征参数，这就要求摄像头必须安装在固定的位置，限制了其应用场合。

从已有的研究成果可以归纳出以下一些基于图像识别的车型分类系统的核心问题和关键技术：

1、目标车辆区域的分割和提取：如何在复杂的背景下分割出完整的目标车辆区域是整个系统的前提和基础，目标车辆区域分割的质量直接影响到最后的车型分类结果。

2、车辆特征的选取：在车辆图像经过预处理后，如何选取具有代表性的特征，并将其量化为描述车辆特征的特征参数至关重要。选取的特征应既能充分体现所属车型的特点，又具有简洁性，以减少计算复杂度。

3、车型分类：在得到车辆的特征参数后，需要设计合适的分类器以对各种车型进行分类。

针对第一项，目前采用的主要方法是基于图像序列的分析，然而目前的研究均没有考虑复杂背景下的情况，没有对车辆区域分割这一问题在实际应用中遇到于图

像识别的车型自动分类系统的困难作深入讨论。

第二项车辆特征的选取主要集中在车辆几何尺寸的度量,包括绝对几何尺寸和相对几何尺寸。绝对几何尺寸即根据摄像头与车辆之间的距离和图像拍摄的角度计算出车辆的实际大小,这种方法存在的缺陷是,摄像头与车辆之间的距离必须保持不变,而这在实际应用中几乎是无法实现的,而且车辆的大小有时并不能代表其所属车型;相对几何尺寸不要求计算车辆的实际尺寸,而是计算车辆各个几何参数之间的比例,这在一定程度上能反映车辆的整体形状结构,而且对车辆图像的大小不作限制,即对摄像头的安装要求比较灵活,从而扩大了其适用范围。

第三项对车型作最终分类,最简单的方法就是对得到的车辆特征参数与标准特征参数进行比较得出分类结果,这种方法运算量小,但只能适用于参数个数较少的情况,而参数个数太少就无法对车型进行有效分类;模板匹配法把分割得到的车辆图像直接与标准模板进行比较,这需要耗费较多的计算时间另外存在一个更实际的问题是,目前的研究都仅限于将车辆划分为轿车、客车和货车三类,或按车辆大小分成大型、中型和小型,而在实际应用场合中都要求将车辆按座位数或吨位划分,因此目前研究的车辆类型划分都很难应用于如公路收费等场合。

1.3 本文的主要研究内容和结构安排

智能交通系统(ITS)是现代交通发展的必然趋势,基于视频技术的车辆类型自动识别是智能交通系统的一个重要研究方向,在城市道路交通监控和高速公路收费等系统中有着广泛的应用前景。本设计要求对于拍摄的图像,在 VC 和 OPENCV 环境下,进行车型识别系统的设计。该系统的特征为:

(1) 具有完整的图像处理系统。

(2) 具有较高的识别率。

本文的研究内容主要有以下几部分:

1、分析车型分类系统的研究现状,确定了以图像识别为基础的车型分类方法,设计了系统的总体框架。

2、基于图像处理和结构模式识别理论,探索出针对车型分类系统的图像处理方法,包括静态车辆区域的分割提取、图像边缘检测、基于车辆特征的提取。

3、在分类方面,针对车型分类问题的独特性,选择最近邻分类法。

2 车型识别系统的总体设计

2.1 基于 OPENCV 的车型识别系统设计

基于图像识别的车型分类系统是集计算机视觉、图像处理、模式识别、工业测控技术、电子技术和系统工程技术于一体的综合综合识别系统，如图 2.1 所示，它一般包括车型图像获取，图像预处理、车型特征提取和分类识别 4 个大的步骤。

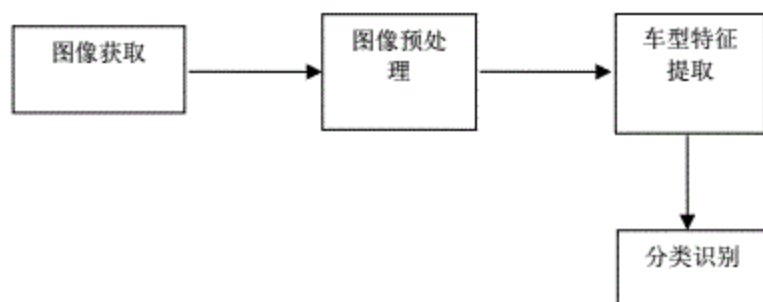


图 2.1 图像处理一般框图

图像获取主要通过摄像机获取动态的视频图像，根据一定的规则，采集包含完整汽车侧面的单帧图像，为后续处理和识别做准备。

图像预处理是采用图像处理技术对采集的单帧图像进行处理，包括滤波去噪、图像转换、二值化、边缘检测、形态学运算等步骤，为下一步的车型特征提取做准备。它是车型识别的重要的一个步骤。由于采集图像环境的复杂性，图像预处理效果的好坏，直接影响后期车型特征的提取，进而影响识别分类效果。

车型特征的提取是车型识别系统的关键步骤。根据先验知识，提取区分不同车型的本质特征。提取的特征一般应满足平移、旋转和缩放不变性。

分类识别是选择合适的分类器。对车型特征进行分类。一般采用传统的模式识别方法，如最近邻分类器等和神经网络分类方法，如 BP 神经网络方法等。

本论文的车型识别系统是对采集的汽车侧面静态图像进行识别。它的步骤如图 2.2 所示：

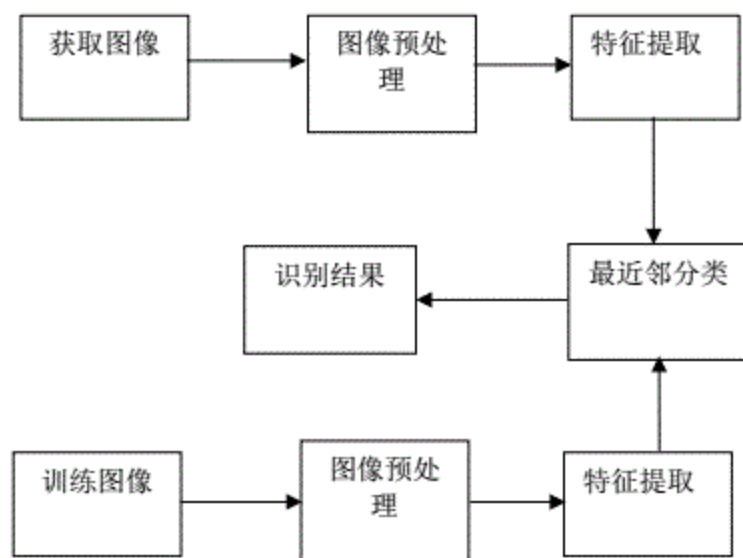


图 2.2 本设计的系统框图

对静态图像先进行图像预处理，然后提取出车型特征，在采用传统的模式识别方法最近邻分类器，通过与事先提取的车型特征进行对比，输出识别结果。

系统在 Windows 系统中，基于 VC++6.0 和 OPENCV 环境下编程实现。

2.2 数字图像处理技术

数字图像处理(Digital Image Processing)是通过计算机对图像进行去除噪声、增强、复原、分割、提取特征等处理的方法和技术。数字图像处理的产生和迅速发展主要受三个因素的影响:一是计算机的发展;二是数学的发展(特别是离散数学理论的创立和完善);三是广泛的农牧业、林业、环境、军事、工业和医学等方面的应用需求的增长。

一般来讲,对图像进行处理(或加工、分析)的主要目的有三个方面:

(1) 提高图像的视感质量,如进行图像的亮度、彩色变换、增强、抑制某些成分,对图像进行几何变换等,以改善图像的质量。

(2) 提取图像中所包含的某些特征或特殊信息,这些被提取的特征或信息往往为计算机分析图像提供便利。提取特征或信息的过程是模式识别或计算机视觉的预处理。提取的特征可以包括很多方面,如频域特征、灰度或颜色特征、边界特征、区域特征、纹理特征、形状特征、拓扑特征和关系结构等。

(3) 图像数据的变换、编码和压缩,以便于图像的存储和传输。

不管是何种目的的图像处理,都需要由计算机和图像专用设备组成的图像处理系统对图像数据进行输入、加工和输出。

数字图像处理主要研究的内容有以下几个方面:

(1) 图像变换由于图像阵列很大,直接在空间域中进行处理,涉及计算量很大。因此,往往采用各种图像变换的方法,如傅立叶变换、沃尔什变换、离散余弦变换等间接处理技术,将空间域的处理转换为变换域处理,不仅可减少计算量,而且可获得更有效的处理(如傅立叶变换可在频域中进行数字滤波处理)。目前新兴研究的小波变换在时域和频域中都具有良好的局部化特性,它在图像处理中也有着广泛而有效的应用。

(2) 图像编码压缩图像编码压缩技术可减少描述图像的数据量(即比特数),以便节省图像传输、处理时间和减少所占用的存储器容量。压缩可以在不失真的前提下获得,也可以在允许的失真条件下进行。编码是压缩技术中最重要的方法,它在图像处理技术中是发展最早且比较成熟的技术。

(3) 图像增强和复原图像增强和复原的目的是为了提高图像的质量,如去除噪声,提高图像的清晰度等。图像增强不考虑图像降质的原因,突出图像中所感兴趣的部分。如强化图像高频分量,可使图像中物体轮廓清晰,细节明显;如强化低频分量可减少图像中噪声影响。图像复原要求对图像降质的原因有一定的了解,一般讲应根据降质过程建立“降质模型”,再采用某种滤波方法,恢复或重建原来的图像。

(4) 图像分割图像分割是数字图像处理中的关键技术之一。图像分割是将图像中有意义的特征部分提取出来,其有意义的特征有图像中的边缘、区域等,这是进一步进行图像识别、分析和理解的基础。虽然目前已研究出不少边缘提取、区域分割的方法,但还没有一种普遍适用于各种图像的有效方法。因此,对图像分割的研究还在不断深入之中,是目前图像处理中研究的热点之一。

(5) 图像描述图像描述是图像识别和理解的必要前提。作为最简单的二值图像可采用其几何特性描述物体的特性,一般图像的描述方法采用二维形状描述,它有边界描述和区域描述两类方法。对于特殊的纹理图像可采用二维纹理特征描述。随着图像处理研究的深入发展,已经开始进行三维物体描述的研究,提出了体积描述、表面描述、广义圆柱体描述等方法。

(6) 图像分类(识别) 图像分类(识别)属于模式识别的范畴,其主要内容是图像经过某些预处理(增强、复原、压缩)后,进行图像分割和特征提取,从而进行判决分类。图像分类常采用经典的模式识别方法,有统计模式分类和句法(结构)模式分类,近年来新发展起来的模糊模式识别和人工神经网络模式分类在图像识别中也越来越受到重视。

数字图像处理的工具可分为三大类:第一类包括各种正交变换和图像滤波等方法,其共同点是图像变换到其它域(如频域)中进行处理(如滤波)后,再变换到原来的空间(域)中;第二类方法是直接在空间域中处理图像,它包括各种统计方法、微分方法及其它数学方法;第三类是数学形态学运算,它不同于常用的频域和空域的方法,是建立在积分几何和随机集合论的基础上的运算。

2.3 本系统用到的图像处理技术

该过程主要是把模拟图像信号转化为计算机所能接受的数字形式,以及把数字图像的显示和表现出来。这一过程包括摄取图像,光电转换及数字化等几个步骤。在车型识别中图像的获取分为静态的和动态的,动态的是通过利用摄像头获取图像,并对其进行车型识别。

图像的预处理包含:图像的灰度化,图像的二值话,图像复原,图像增强,图像分割,边缘检测。

1. 图像的灰度化

一幅完整的图像,是由红色绿色蓝色三个通道组成的。红色、绿色、蓝色三个通道的缩览图都是以灰度显示的。用不同的灰度色阶来表示“红,绿,兰”在图像中的比重。通道中的纯白,代表了该色光在此处为最高亮度,亮度级别是255。

在计算机领域中,灰度数字图像是每个像素只有一个采样颜色的图像。这类图像通常显示为从最暗黑色到最亮的白色的灰度,尽管理论上这个采样可以任何颜色的不同深浅,甚至可以是不同亮度上的不同颜色。灰度图像与黑白图像不同,在计算机图像领域中黑白图像只有黑色与白色两种颜色;灰度图像在黑色与白色之间还有许多级的颜色深度。但是,在数字图像领域之外,“黑白图像”也表示“灰度图像”,例如灰度的照片通常叫做“黑白照片”。在一些关于数字图像的文章中单色图像等同于灰度图像,在另外一些文章中又等同于黑白图像。

按一定的规律修改像素的灰度值,是图像的亮度或对比度放生改变,使之更容

易分辨，货或达到某种预定的视觉效果。

2. 图像的二值化

图像的二值化处理就是将图像上的像素点的灰度值设置为 0 或 255，也就是讲整个图像呈现出明显的黑白效果。

将 256 个亮度等级的灰度图像通过适当的阈值选取而获得仍然可以反映图像整体和局部特征的二值化图像。在数字图像处理中，二值图像占有非常重要的地位，首先，图像的二值化有利于图像的进一步处理，使图像变得简单，而且数据量减小，能凸显出感兴趣的目标的轮廓。其次，要进行二值图像的处理与分析，首先要把灰度图像二值化，得到二值化图像。

所有灰度大于或等于阈值的像素被判定为属于特定物体，其灰度值为 255 表示，否则这些像素点被排除在物体区域以外，灰度值为 0，表示背景或者例外的物体区域。

为了分析图像的特征，常常需要从图像分离出对象物，从而把图像看做仅由对象区域与背景区域所构成。

3. 图像复原

当造成图像退化的原因已知时，复原技术可以对图像进行修复。图像复原的关键的是对每种退化的偶需要有一个合理的模型。例如，掌握了聚焦不良成像系统的物理特性，便可恢复原模型，而且对获取图像的特定光学系统的直接测量也是可能的。退化模型和特点数据一起描述了图像的退化的影响，从而产生一个等价与理想成像系统所获得的图像。因为本文采用的图像为静态图像，从而选取的图像都为识别度很高的图像，则而不需要图像复原。

4. 图像增强

图像增强是对图像质量在一般意义上的改善。当无法知道图像有关的定量信息时，可以使用图像技术较为主观地改善图像的质量。所以，图像增强技术是用于改善图像质量所采取的一种方法，因为增强技术并非是针对某种退化所采取的方法。所以很难预测哪一种特定技术是最好的，只能通过实验和分析误差来选择一种的合适的方法。有时可能需要彻底改变图像的视觉效果，以便突出重要特征的可观察性，使人或计算机更以观察或检测。在这种情况下，可以把增强理解为增强感兴趣的特征的可观察性，而非改善视感质量。电视节目片头或片尾处的颜色，轮廓的变换，

其目的是得到一种特殊的艺术增强动感和力度。

5. 图像分割

把图像分成区域的过程就是图像分割。图像自动分割是图像处理中最困难的问题之一。人类视觉系统的优越性，使得人类能够将所观察的复杂环境观察出来，可计算机却是个难题，由于解决和分割有关的基本问题是特定领域中图像分析实用化的关键一步，因此，将各种方法融合在一起并使用技术来提高处理的可靠性和有效性是图像分割的研究热点。

6. 边缘提取

图像中的边缘是视觉感知的重要线索，这些边缘对应图像中特性“有意变化的地方。例如，对于车辆图像来说，边缘特征常常对应车辆的轮廓线或线。图像的边缘检测能够使图像的轮廓更加突出，是一种重要的区域处理技在对图像进行特征提取之前，一般都需要进行边缘检测，然后进行二值化处边缘检测将突出图像的边缘，边缘以外的图像区域通常被削弱。处理后边界度与原图中边缘周围的亮度变化率成正比边缘发生在图像灰度值不连续的地方，可以对图像进行差分运算得到。的边缘检测算子包括 sobel 算子、Prewitt 算子、Roberts 算子、earmy 算子和而本文采用了 Sobel 算子和 LaPlace 算子 Sobel 来进行边缘检测。

2.4 OpenCV

OpenCV 是英特尔公司开发的开源图像处理和计算机视觉函数库。它由一系列 C 函数和少量 C++ 类构成，实现了图像处理和计算机视觉方面的很多通用算法。OpenCV 是一个跨平台的中、高层 API 构成，目前包括 300 多个 C 函数。它不依赖与其它的外部库，尽管也可以使用某些外部库。OpenCV 对非商业用途和商业用途都是免费的。

这是目前的现状。而标准的 API 将简化计算机视觉程序和解决方案的开发。OpenCV 致力于成为这样的标准 API。

它有以下特点：

- (1) 开放的 C/C++ 源码
- (2) 基于 Intel 处理器指令集开发的优化代码
- (3) 统一的结构和功能定义

- (4) 强大的图像和矩阵运算能力
- (5) 方便灵活的用户接口
- (6) 同时支持 MS-WINDOWS、LINUX 平台

3 基于 OPENCV 的车型系统的实现

本章详细介绍基于 OPENCV 的车型系统的实现。

3.1 OpenCV 环境配置

首先按要求安装 OPENCV 1.0 软件，安装完成后要对其进配置。OPENCV 环境配置包括两个方面，一个是系统环境配置；一个是 VC++6.0 软件环境配置。

1. Windows 环境变量的配置

首先检查 C:\Program Files\Opencv\bin 是否已经被加入到环境变量 PATH，如果没有，请加入。先打开“我的电脑”属性，选择“高级”选项，点击“环境变量”，如下图所示：

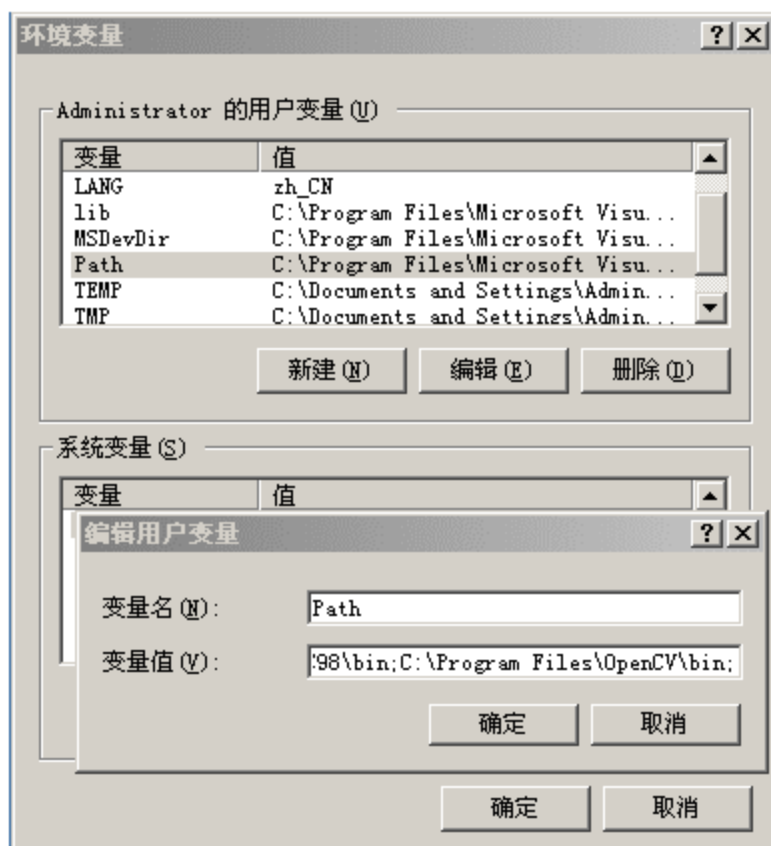


图 3.1 windows 环境变量框图

2. 配置 VC++6.0

菜单 Tools->Options->Directories; 先配置 lib 路径, 选择 Library files, 在下方填入路径:

C:\Program Files\OpenCV\lib

然后选择 include files, 在下方填入路径:

C:\Program Files\OpenCV\cxcore\include

C:\Program Files\OpenCV\cv\include

C:\Program Files\OpenCV\cvaux\include

C:\Program Files\OpenCV\ml\include

C:\Program Files\OpenCV\otherlibs\highgui

C:\Program Files\OpenCV\otherlibs\cvcam\include

如下图所示:

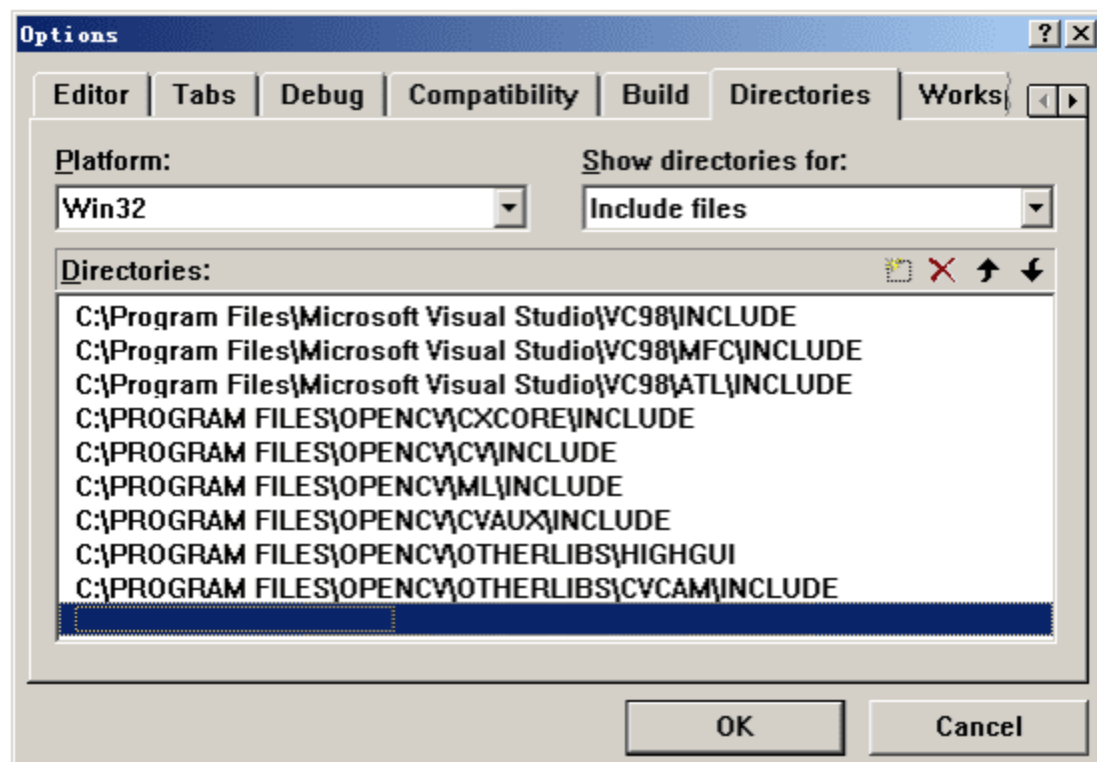


图 3.2 options 设置

3. 项目配置

每创建一个将要使用 OpenCV 的 VC Project, 都需要给它指定需要的 lib。菜单

Project->Settings, 然后将 Setting for 选为 AllConfigurations, 然后选择右边的 link 标签, 在 Object/library modules 附加上:

```
Cxcore.Lib cv.lib ml.lib cvaux.lib highgui.lib cvcam.lib
```

3.2 图像的读取

在 MFC 环境下, 要使用 OPENCV 的函数和功能, 主要通过 CImage 类及其成员函数实现。IplImage* 结构指针是 OPENCV 最基本的结构, OPENCV 的基本函数都是以此结构为基础实现的。如何实现 IplImage* 的对象与 CImage 类对象的数据传输, 从而实现图像数据的读取、显示和处理是关键。具体的实现过程是通过 CImage 的 CopyOf() 函数和 GetImage() 函数实现。

```
IplImage* Image;          //生成 OPENCV 处理的对象
```

```
CImage m_image;          //生成 CImage 对象 1
```

```
CImage MyImage;          //生成 CImage 对象 2
```

在打开文件的 OnOpenDocument(LPCTSTR lpszPathName) 函数中, 添加下列语句

```
m_image.Load(lpszPathName,-1); //以图片原本的通道方式打开图片
```

```
MyImage.CopyOf(m_image);      //复制图片, 接受以后的处理
```

```
Image=MyImage.GetImage();      //得到 IplImage* 指针, 方便运用 opencv 进行处理
```

这样就将图像读进了 m_image 中, 并复制给了 opencv 的对象 Image。

图像的显示在 OnDraw(CDC* pDC) 函数中实现:

```
CRect rect;
```

```
GetClientRect(rect);
```

```
//把显示区设为图像的真实宽度和高度
```

```
rect.bottom=pDoc->MyImage.Height();
```

```
rect.right=pDoc->MyImage.Width();
```

```
//定义一个 CSize 对象, 用于保存滚动条的尺寸, 设为图像的真实尺寸
```

```
CSize sizeTotal(rect.right, rect.bottom);
```

SetScrollSizes(MM_TEXT, sizeTotal); //设置滚动条的位置

//调用 CImage 类的成员函数 DrawToHDC 在视图窗口显示图像

//函数形参为当前的设备环境和显示区大小

pDoc->MyImage.DrawToHDC(pDC->m_hDC, rect);

OPENCV 函数处理完成后的图像，通过 CopyOf() 函数传回 MyImage，显示。

3.3 图像的边缘检测

图像中的边缘是视觉感知的重要线索，这些边缘对应图像中特性“有意变化的地方。例如，对于车辆图像来说，边缘特征常常对应车辆的轮廓线或线。图像的边缘检测能够使图像的轮廓更加突出，是一种重要的区域处理技在对图像进行特征提取之前，一般都需要进行边缘检测，然后进行二值化处边缘检测将突出图像的边缘，边缘以外的图像区域通常被削弱。处理后边界度与原图中边缘周围的亮度变化率成正比边缘发生在图像灰度值不连续的地方，可以对图像进行差分运算得到。

Sobel 是边缘检测的算法之一，使用扩展 Sobel 算子计算一阶、二阶、三阶或混合图像差分。

在 OpenCV 中运用 cvSobel 函数如下为运用 cvSobel 的函数算法函数程序：

void cvSobel (const CvArr* src(输入图像).CvArr*dst (输出图像). Int xorder (X 方向上的差分阶数). Int aperture - size=3);

函数 cvSobel 通过对图像用相应的内核进行卷积操作来计算图像的差分 (3.1)

$$\det(x, y) = \frac{d^{xorder+yorder} src}{dx^{xorder} dy^{yorder}}(x, y)$$

(3.1)

由于 Sobel 算子结合了 Gaussian 平滑和微分，所以，其结果或多或少的对噪声有一定的鲁棒性通常情况，函数调用用如下参数

(xorder=1, yorder=0, aperture_size=3) 或 (xorder=0, yorder=1, aperture_size=3) 来计算一阶图像差分。

Sobel 算子采有如表的 3.1 的模板。

表 3.1 Sobel 算子水平模板

-1	-2	-1
0	0	0
1	2	1

表 3.2 Sobel 算子垂直模板

-1	0	1
-2	0	2
-1	0	1

上面图表为对图像像素处理的模板矩阵，每个图像像素点都与这个两个模板作卷积，第一个模板用于检测水平边缘，第二个模板用于检测竖直边缘。两个卷积结果的最大值即为该点的输出值。Sobel 边缘检测算子对灰度渐变和噪声较大的图像处理得比较好。

LOG 边缘检测

(1) LOG 边缘检测的一般算法：

LOG 算子提取车辆图像边缘的方法是一种比较简便的处理方法，与其它方法比较，优于 Roberts 算子和 Sobel 算子的方法，边缘比较完整，位置比较准确；与小波分析的方法比较，虽然获得的图像质量略有不足。但其算法简单处理速度快，在流量较大路口的车辆检测系统中有较高实用价值

LOG 算法是一种 2 阶边缘检测方法。它通过寻找图像灰度值的中二阶微分中的过零点(Zero Crossing)来检测边缘点。因为，灰度缓变形成的边缘经过微分算子峰值两侧符号相反，而原先的极值点对应二阶微分中的零点为了消除噪声影响。必须用高斯函数对图像进行滤波在对滤波后的图像求二阶导数，即按照下式计算

$$v^2[G(x,y)*f(x,y)] \quad (3.2)$$

其中以 $f(x,y)$ 为图像， $G(X,Y)$ 为高斯函数，上述两个处理步骤可以合成一个算

子，经运算可得

$$v^2[G(x, y) * f(x, y)] = v^2 G(x, y) * v^2 f(x, y) \quad (3.3)$$

公式中 $V^2 G(x, y)$ 称为拉普拉斯高斯算子，经运算可得：

$$V^2 G(x, y) = \frac{1}{2\pi\sigma^4} \left(\frac{x^2 + y^2}{\sigma^2} \right) \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \quad (3.4)$$

$V^2 G(x, y)$ 是关于原点对称的函数，其主瓣宽度为： $W = 2\sqrt{2}\sigma$

$G(x, y)$ 比较符合人的视觉特性，LoG 算法能较好地反映了视觉模型。

(2) LoG 算法过零检测

对 $V^2 G(x, y)$ 简化得到下公式

$$V^2 G(x, y) = k \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \quad (3.5)$$

一般， σ 取值较大时，趋于平滑图像； σ 较小时，则趋于锐化图像。

事实上，检测边界就是寻找 $V^2 G(x, y)$ 的过零点，可用以下几种参数表示过零点处灰度变化的速率：

(1) 过零点处的斜率。

(2) 二次微分峰——峰差值。

(3) 二次微分峰——峰间曲线下面积绝对值之和。

边界点方向信息可由梯度算子给出。为减小计算量，在实用中可用高斯差分算子(DOG)

$$DOG(\sigma_1, \sigma_2) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2}\right) - \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{x^2 + y^2}{2\sigma_2^2}\right) \quad (3.6)$$

3.4 区域分割

图像中的区域是指一个相互连通的、具有一致的“有意义”属性的像元素。区域分割就是从全图出发，按“有意义”属性一致的准则，决定每个像素的归属区域。区域分割是图像处理中的一项基本内容，也是图像理解中关键的一步。区域分割将原始图像转化为更抽象更紧凑的形式，使得更高层的图像分析和理解成为可能。

区域分割算法大致可分为两类：一是基于图像的灰度统计特征，通过某一约束函数，选取合理的阈值进行区域分割。这类算法典型的有：直方图分析法、最佳熵自动门限法、最小误差法和类间方差法。二是基于图像的局部灰度梯度变化特征，如边缘检测，即先找出目标的边界，通过某一判别标准，识别出目标边界，用边界来表示目标，然后进行聚类，将目标从图像中识别出来。这类方法包括：自适应阈值法，模糊 C 均值法。本文根据车辆图像的特点和实际需要，采用上述第一类基于图像的灰度统计特征的阈值分割法对图像进行区域分割。下面简单介绍本文运用的直方图分析法。

3.5 图像的轮廓提取

识别车型，需要提取汽车外部轮廓特征，对二值图像进行轮廓跟踪，得到轮廓曲线。所以，首先要对上述二值化图像进行轮廓跟踪。轮廓跟踪的基本方法是：先根据某些严格的“探测准则”找出目标物体轮廓上的像素，再根据这些像素的某些特征用一定的“跟踪准则”找出目标物体上的其它像素点。传统的串行跟踪算法是：从上到下，从左到右扫描全图，找到所有以 0(黑像素)为邻点的 255(白像素)，按定义，这些值为 255 的点一定是边界点。这种算法要对每个边界像素周围的八个邻点进行判断，计算量较大。另一种称为虫随法的搜索轮廓的思想是这样的：从白边缘向黑色像素区走，当穿过黑像素时左转，并走向下一个像素，同时始终遵循遇黑左转，遇白右转的原则，一直到回到原始出发点为止。这样就可以得到目标物体的轮廓。但这种方法所得到的轮廓与初始点有关。

下面介绍的一种轮廓算法由两部分组成，一个是计算轮廓方向序列的方法称为 T 算法；另一是计算轮廓起始点方法称为 IP(Initial. Point)算法。轮廓算法的步骤是由这两个算法依次交叉进行。即第一个轮廓起始点找到后，进行第一个轮廓方向序列的计算，算完后再寻找第二个轮廓起始点，接着计算第二个轮廓方向序列；如此依次交叉的进行，计算完图中所有轮廓为止。

下面分别介绍一下轮廓方向序列及轮廓起始点算法。

3.5.1 轮廓方向序列的计算(T 算法)

轮廓方向序列是由轮廓起始点开始到轮廓上的第二点。第三点，直到最后一点再返回起始点为止，由方向所组成的方向序列。确定轮廓上一点走向下一点的方向是用“最先左看规则”，即从进入轮廓点(如 A 点)的方向看去(如图 3.3)，最先向左方向寻找，若遇到灰度级和 A 点相同的邻点，则轮廓由 A 点走向这一点，若左看没有灰度级和 A 点相同的邻点，再按向上看、向右看、向下看的顺序寻找，直到找到有灰度级相同的点为止。将轮廓由 A 点移向该点，若四个方向都没有表示这个轮廓只是由一个像素(孤立点)构成的。

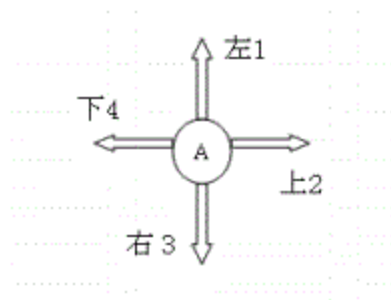


图 3.3 向右看图

3.5.2 轮廓起始点的寻找(IP 算法)

为了寻找轮廓起始点 IP，我们使用顺序扫描搜索方法，也就是从数字图像的左上角(第一行，第一列)的像素开始，按行从左到右，按列从上到下逐点顺序扫描，知道右小角最后一个像素为止，对扫描遇到的每一个像素，进行判别是否为轮廓起始点 IP。如果判别每一像素是否为 IP 呢?先来介绍一下“扫描搜索比较表”和“起始点判别准则”。

①扫描搜索比较表构成(CPL)

(1)仅每扫一行，制一个表，当然开始扫描前表是空的。

(2)对每一行扫描从左到右逐点进行判别。若遇到点标志为 A，将该点的灰度级值从表的右端依次填入表中。若遇到点标志是 D，将表中最靠近标志为 A 的灰度级划去。若遇到点标志是 D 或 I，表不变化，即不填也不划去什么。

(3)每一行扫描完毕，表也一定是空的，因为轮廓总是封闭的，轮廓通过某一行

有向下的点(D)，有向上的点(A)，即 A 点数一定等于 D 点数。

②轮廓起始点判断准则

在扫描搜索过程中，凡符合下列两个条件的就判为轮廓起始点 IP。

(1)标志是 I(即不是已确认过的轮廓上的点)。

(2)它的灰度级值不等于扫描搜索表中最靠近的标记为 A 的点的灰度级值。按照这两种准则就可以判断轮廓起始点，这里注意的就是一副数字图像的左上角的点(第一行，第一列)总认为是第一轮廓起始点，这是不难理解的。二值化图像轮廓提取还可以采用数学形态学的方法：设图像为 A，结构元素为 B，图像边缘为

$$\beta(A) = A - (A \otimes B)$$

即图像 A 与 A 被 B 腐蚀的结果的差为图像的边缘。本系统中采用如下跟踪准则：对差值按照从左到右，从下到上的顺序搜索，找到的第一个白色点一定是最左下方的边界点，以这个边界点起始，假设已经沿顺时针方向环绕整个图像一圈找到了所有的边界点。由于边界是连续的，所以，每个边界点都可以用这个边界点对前个边界点所张的角度来表示，所以，从第一个边界点开始，定义初始的搜索方向为沿左上方；如果左上方的点是白色的，则为边界点，否则搜索方向顺时针旋转 45 度。这样一直到找到另一个白色点为止。然后把这个白色点作为新的边界点，在当前搜索方向的基础上逆时针旋转 90 度，继续用同样的方法搜索下一个白点，直到返回最初的边界点为止。

3.6 车辆的特征提取

提取特征时我们必须考虑两个基本问题：一是特征能够较容易地从图像中提取；其二是所选取的特征必须有利于分类，也即对分类有较大的贡献。从外观上看，最能够区分不同车型的是车辆的形状，因此形状是区分车型的重要依据但在公路收费等场合中以车辆的座位数作为分类的依据，因此需要将车辆的形状特征与其座位数对应起来。

表 3.1 为车辆形状与类型的对应关系：我们首先对汽车的形状进行分类，即我们将这些汽车图像的形状作为汽车判断的标准，由这个汽车形状特征与读入的汽车图像形状特征进行比对，只要符合其中某一个汽车图像标准，就会将其识别为该汽车。

表 3.3 车辆形状与类型





形状	形状类别	车辆种类	车型分类
	形状 1	轿车	1 类
	形状 2	越野车	2 类
	形状 3	小型和大型面包车	3 类
	形状 4	客车	4 类

表 3.4 中 4 类的临近参数

车辆类型	车长与车高比
1 类	2.6768

2 类	1. 8872
3 类	2. 9045
4 类	2. 8773

上表格为识别汽车图像特征标准，前几章节主要介绍了图片显示与特征提取，通过对汽车进行图像预处理，从而进一步提取汽车图像的特征点，按要求转换为识别汽车图像所要求的图像的特征形式，利用最近邻方法，将提取出的特征信号设定的标准特征信号相比较，取与汽车标准特征信号特征距离最小的那个标准特征信号，可以得出汽车车型。总之，汽车车型的识别过程就是一个特征提取与特征比较的过程。

3.7 分类识别

对于任何一个模式分类器，除了希望它有尽可能高的识别率外，还希望能比较准确地估计它的分类结果的准确性，即置信度，置信度的主要用途有：(1) 为拒识提供依据。在很多应用场合下，我们不仅希望识别率尽可能高，还希望误识率低，在一定的分类器下，这只能通过拒识部分样本实现。这样就需要决定拒识哪一部分样本，如果拒识的样本总是识别置信度最低的样本，系统的误识率就会在拒识率一定的情况下获得最大幅度的下降。(2) 为多方案集成提供根据。现在，多方案集成是模式识别的一个重要趋势。人们已不再满足于简单的表决，而希望能充分利用单个分类器的各种信息。

其中，每个分类器的置信度是一种重要信息。其意义是明显的：当多个分类器的结果不同时，应当选择置信度比较大的分类器的结果，至少应给予较大的权重。鉴于置信度的重要地位，基本上各种应用系统都以某种方式给出置信度的估计。对于基于前向神经网络的分类器，文献中证明了在使用均方误差或交叉熵做代价函数时，神经网络输出的期望值是各个类别的后验概率，这就为用神经网的输出估计置信度提供了强有力的理论根据。而对于早在 60 年代就已提出、并在几十年的模式识别实践中得到广泛应用的最近邻分类器，尽管人们在实践中采用了各种置信度估计的经验公式，未见有对这些估计的理论。

在聚类分析中，由于没有样本类别的先验知识，因而按最近距离原则的基本思想进行分类。最近邻法在具有训练样本条件下，但不进行训练，而是按最近距离原

则对待识模式进行分类，这种分类技术思想直观，方法简单，效果较好，其中 k-NN 法可以看做是一种从样本中获取最大后验概率的方法，最近邻法的某些技术在理论上可以达到先验知识完备的贝叶斯决策的分类效果，能适应类域分布较复杂的情况，这类技术是最重要的模式识别技术之一。实际上，这类分类技术书贝叶斯决策思想的一种体现，因某类样本距待识模式最近，或某一小区域某类样本较多，意味着此处这类概率较大。

对于 c 类问题，设类 $\omega_i (i=1,2,\dots,c)$ 有 N 个训练样本 $x_j^{(i)} (j=1,2,\dots,N_i), N = \sum_{i=1}^c N_i$ 。分类的思想是对于一个待识模式 x ，分别计算它与各已知类别的训练样本 $x_j^{(i)}$ 的距离，将它判为距离最近的那个样本所需的类。在这样大的分类思想下， w_i 类的判断函数为

$$d_i(x) = \min_{j=1,2,\dots,N_i} \|x - x_j^{(i)}\| \quad (i=1,2, \dots, c)$$

(3.7)

判决规则为

如果

$$d_m(x) = \min_{i=1,2,\dots,c} d_i(x)$$

(3.8)

则判 $x \in w_m$

由于上述方法只根据离待识模式最近的一个样本的类别而决定其类别，通常称其为 1-NN 方法。为了克服单个样本类别的偶然性以增加样本哪一类最多，就将 x 判属哪一类。设 $k^1, k^2 \dots k^c$ 分别为待识模式 x 的 k 个最近邻样本属于 w_1, w_2, \dots, w_c 类样本数，定义 w_i 类的判别函数为

$$d_i(x) = k_i (i=1,2,\dots,c)$$

(3.9)

判决规则为

如果 $d_m(x) = \max_{i=1,2,\dots,c} [d_i(x)]$ 则判 $x \in w_m$

这种方法通常称为 k -NN 法。

4 实验结果与处理

本论文的车型识别主要识别主要为 4 类：小轿车，货车，吉普车，和大型客车。对于这 4 类的识别步骤都是相同其主要为：图像的预处理，图像的特征提取和车型的分类。以小轿车为例。

4.1 小轿车的车型识别

首先对小轿车进行图像的获取如下图 4.1 所示



图 4.1 轿车原图

对轿车图像进行图像的预处理得到的为下图所示；



图 4.2 边缘检测后的图像

原图与边缘检测后的图像可以看出边缘检测后的图像影子与车身是混在一起，造成这种现象的原因是因为原图是彩色图像，没有对原图进行灰度化，而程序识别的像位是 255 位，所以造成以上的原因。

一般对图像进行预处理时 一般会遇到的以下几个问题：

(1) 图像无法打开。

本设计所采用的图片格式为 BMP 格式，则所存文件夹的格式也应为 BMP 格式的，如图片的格式不为 BMP,软件是无法打开的，应将图片的格式转化为 BMP 格式。

(2) 图像为 BMP 格式，软件也无法打开。

在程序设计中设定的图片位数为 256 位，高于或低于 256 位的只能打开图像，程序是无法运行的，应将图像进行转化，转为 256 位的。

4.1.1 图像的轮廓提取

本论文的特正提取主要是对轮廓特征的提取，图 4.3 为轮廓特取。



图 4.3 轮廓提取图

而对于背景与车色相近的图像，本设计对其进行轮廓的提取，会生成背景与车混浊在一起的图像。

4.1.2 图像的腐蚀



图 4.4 图像的腐蚀

图像的腐蚀为图像形态学的一部分，主要作用是消除车边界点，使目标缩小，消除小于结构元素的噪声点，为下一步运行做出准备。

4.1.3 车型的识别

下图为轿车通过来识别系统，依次按照不同的处理算法，最后得出识别结果，识别效果为下图：



图 4.5 车型的识别

4.2 小货车的车型识别

下图为小货车的原图图像：



图 4.6 小货车

下一步将要对小货车进行图像预处理，其中包含中值滤波、阈值分割、二值化、图像形态学处理等。

首先对其进行中值滤波处理，去除其图像中的高频噪声，为以后的图像处理做好铺垫。

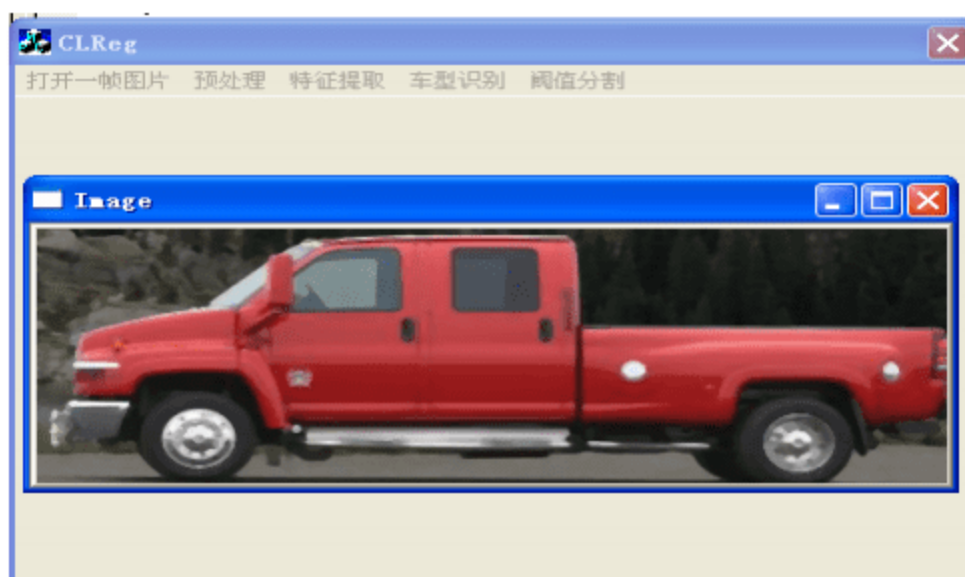


图 4.7 中值滤波处理图

再对小车做图像二值化处理，如下图：

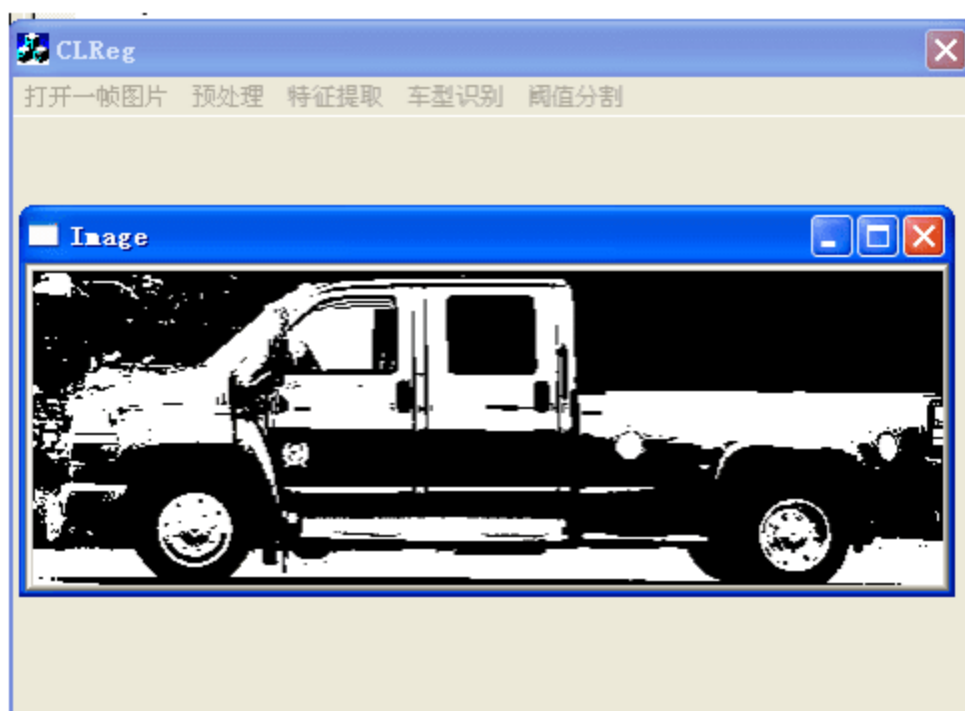


图 4.8 二值化处理图

最后再对小车进行识别，即可得出识别结果：



图 4.11 识别结果

通过对以上两种车型的反复试验，识别比较准确，可以满足识别的基本要求。

本论文的主要研究的是车型识别，在对客车进行车型识别时，有时会出现将其识别为货车，造成这一错误主要原因是因为特征提取时，将参数选择为车底和车高的比，而有些客车的车底和车高之比与货车的车底和车高的比相近。

结 论

近年来, 车型识别已成为许多研究领域的关键技术, 成为计算机视觉和图像处理的一个热点研究方向。但是, 用于车型识别的方法仍很有限, 本文针对这一课题展开了初步研究与设计。

本文重点讨论了车型识别中的三个关键性问题: 车辆图像的预处理、特征提取以及识别分类。最后根据本文提出的整合的特征提取策略和主成分分析方法做实验证明此方法有较好的稳定性。

车型识别涉及到很多理论和技术问题, 本文只是进行了初步的探索和尝试, 提出了一些新的设想与方法, 由于受时间、资金及认识等条件的限制, 本文仅完成了部分工作, 所取得的成果只是阶段性的, 离实际应用还有较大的距离。在这个基础上可进行的后续研究可以从以下几个方面考虑。

1. 本文主要是针对停车场的车辆将识别类别设定为摩托车和汽车两类, 但在很多实际情况中, 这种粗略的分类是不够全面的, 需要将汽车类别进行细化。车型类别的增多势必会给样本图像的选取、特征提取和识别分类等步骤带来困难, 增加算法的复杂度。因此, 如何选取样本图像, 提取特征乃至进一步优化降低复杂度需要进一步的研究。

2. 本文侧重于提取车辆图像的代数特征, 虽然代数特征具有比其他特征更为有效的特征, 但是若能结合其他方法提取一些其他特征, 或许能更好地描述和表示车辆图像, 从而进一步提高系统的识别率。

社会经济效益分析

随着我国经济的迅猛发展，我国的交通运输业，尤其是高速公路得到了迅猛的发展，随之而来的是高速路口收费站的日益繁忙，影响了工作人员的工作效率及工作热情，在这种背景下，车型识别系统成为研究的热门方向。车型分类系统的主要应用场合为停车场、路桥收费管理和公安交通管理部门。交通管理部门则可以利用该系统实现无人值守警察功能，实时抓拍违规车辆，结合相关数据库还可实现未年检车辆稽查、违章未交罚款、特定车辆查询等功能。

本论文的车型识别系统设计就是在这个应用背景下进行的一个有意义的尝试。本论文所设计的车型识别系统具有良好的识别效果，实时测量显示直观迅速、运行操作简单，基本达到本科毕业设计要求。软件方面，主要采用了 intel 公司推出的 opencv 计算机开源视觉库，它是由一系列的 C 函数和少量的 C++ 类构成，实现了图像处理和计算机视觉方面的很多通用算法。本系统的开发是基于 VC++6.0 以 opencv 为图像处理工具实现的。通过调试说明：利用本车型识别系统基本能够达到识别车型的要求，比较准确的实现了车型识别。本设计具有很好的适应性和较高的经济性，可以和好的再交通运输业推广。

参考文献

- [1] 孙即祥. 现代模式识别. 湖南省: 国防科技大学出版社, 2002.
- [2] 高守传, 姚邻田. 数字图像处理. 北京: 中国铁道出版社, 2005
- [3] 娄莉, 党瑞荣. 基于智能图像处理技术的车型识别, 微电子学与计算机, 2006: 7(4): 71-73
- [4] 章毓晋. 数字图象处理和分析. 北京: 清华大学出版社, 1993
- [5] 娄莉, 党瑞荣. 基于智能图像处理技术的车型识别. 微电子学与计算机, 2006 李在铭. 数字图像处理、压缩与识别技术, 西安: 电子科技大学出版社, 2000
- [6] 陈爱斌. 基于特征车的汽车车型识别, 信息技术, 2004
- [7] 沈勇武. 基于图像识别的车型自动分类系统, 浙江大学硕士学位论文, 2008.
- [8] 刘瑞祯, 于仕琪. OpenCV 教程基础篇. 北京: 北京航空航天大学出版社, 2006.
- [9] 章毓晋. 数字图象处理和分析. 北京: 清华大学出版社. 1993.
- [10] Guyon I. Application of neural networks to character recognition. International Journal of Pattern Recognition and Artificial Intelligence. 1993, 7(4): 721. 755.
- [11] Setchell J. Applications of Computer Vision to Road—Traffic Monitoring. PH. D Thesis. University of Bristol, England. 1997
- [12] Guyon I. Application of neural networks to character recognition. International Journal of Pattern Recognition and Artificial Intelligence. 1993, 7(4): 721-755

致 谢

本文是在张海军老师的指导下完成的，在整个论文的选题、研究与撰写期间，始终得到了张老师精心的指导、热情的鼓励和支持。在四年的年的本课学习时间里，张老师给予我学业上悉心的指引和教诲，使我终生难忘；张老师渊博的知识、严谨的治学态度和高瞻远瞩的学术思想使我受益匪浅，在此，谨向老师致以最崇高的敬意和衷心的感谢！

由衷地感谢帮助过我的所有学长，他们给予了我许多的指导和关怀，使我能够顺利地完学业和论文工作，谢谢他们的耐心指导和无私帮助。

特别要感谢的是我的家人，正是他们在物质和精神上的支持与鼓励，使我得以潜心于学业，顺利完成本论文。

最后，对所有曾经给予我关心和帮助的朋友们都表示诚挚的感谢。

附录 程序表

1. 下面为 opencv 图像显示部分程序:

```
#include "cv.h"

#include "highgui.h"

int main( int argc, char** argv )
{
    IplImage* pImg; //声明 IplImage 指针
    //载入图像
    if( argc == 2 &&
        (pImg = cvLoadImage( argv[1], 1)) != 0 )
    {
        cvNamedWindow( "Image", 1 );//创建窗口
        cvShowImage( "Image", pImg );//显示图像
        cvWaitKey(0); //等待按键
        cvDestroyWindow( "Image" );//销毁窗口
        cvReleaseImage( &pImg ); //释放图像
        return 0;
    }
    return -1;
}
```

- 2 图像创建, 保存和复制

```
include "cv.h"

#include "highgui.h"
```

```

IplImage* pImg2 = cvCreateImage(cvGetSize(pImg),
    pImg->depth,
    pImg->nChannels);
cvCopy(pImg, pImg2, NULL);
cvSaveImage(argv[2], pImg2); //把图像写入文件
cvNamedWindow( "Image", 1 ); //创建窗口
cvShowImage( "Image", pImg ); //显示图像
cvWaitKey(0); //等待按键
cvDestroyWindow( "Image" ); //销毁窗口
cvReleaseImage( &pImg ); //释放图像
cvReleaseImage( &pImg2 ); //释放图像
return 0;

```

3. 图像的预处理，中值处理运算程序

```

if(image==NULL)return;
//判断图像是否加载成功
cvSmooth(image,image,CV_MEDIAN);
//利用 opencv 子带的库函数，对图像进行中值滤波
cvShowImage("Image", image);
// 对图像进行显示
Invalidate();

```

4. 下面为图像预处理的二值化部分程序：

```

if(img1==NULL)return;
cvThreshold(img1, img1,100,255,CV_THRESH_BINARY);
//cvAdaptiveThreshold(img1,img1,255,CV_ADAPTIVE_THRESH_MEAN_C,

```

```
CV_THRESH_BINARY, 3, 5 );

cvShowImage("Image", img1); // display it

Invalidate();
```

5. 下面为图像预处理的形态学处理部分程序：

```
if(img1==NULL)return;
cvNamedWindow("Morph", 1);
IplConvKernel*kernal=cvCreateStructuringElementEx(1,2,0,0,CV_SHAPE_RECT);
IplConvKernel*kernal2=cvCreateStructuringElementEx(2,1,0,0,CV_SHAPE_RECT);
cvDilate(img1,img1,kernal);
IplConvKernel*kernal1=cvCreateStructuringElementEx(2,2,0,0,CV_SHAPE_RECT);
cvErode(img1,img1,kernal2);
cvDilate(img1,img1,kernal);
cvErode(img1,img1,kernal2);
cvReleaseStructuringElement(&kernal);
cvReleaseStructuringElement(&kernal1);
cvReleaseStructuringElement(&kernal2);
cvShowImage("Morph", img1); // display it
Invalidate();
```

6. 下面为图像预处理的形态学处理部分程序：

```
IplImage * pImage8uGray=NULL;
IplImage * pImage8uSmooth=NULL;
IplImage * pImage16uGraySobel=NULL;
IplImage * pImage8uGraySobelShow=NULL;
pImage8uGray=cvCreateImage(cvGetSize(image),IPL_DEPTH_8U,1);
pImage8uSmooth=cvCreateImage(cvGetSize(image),IPL_DEPTH_8U,1);
pImage8uGraySobelShow=cvCreateImage(cvGetSize(image),IPL_DEPTH_8U,1);
//转灰度
cvCvtColor(image,pImage8uGray,CV_BGR2GRAY);    //高斯滤波
cvSmooth(pImage8uGray,pImage8uSmooth,CV_GAUSSIAN,1,0,0); //cvSobel 要求目标图像必须是 IPL_DEPTH_16S
pImage16uGraySobel=cvCreateImage(cvGetSize(image),IPL_DEPTH_16S,1); //
```



```

    计算一阶 x 方向的差分,也可以计算一阶 y 方向
    cvSobel(pImage8uSmooth,pImage16uGraySobel,0,1,3); //cvSobel 要求目标图像必
    须是 IPL_DEPTH_16S
    cvConvertScaleAbs(pImage16uGraySobel,pImage8uGraySobelShow,1,0);//再把格式
    转回来,用于显示
    cvNamedWindow("灰度图像 Sobel 变换",CV_WINDOW_AUTOSIZE);
    cvShowImage("灰度图像 Sobel 变换",pImage8uGraySobelShow);
    cvWaitKey(0);
    cvDestroyWindow("灰度图像 Sobel 变换");
    cvReleaseImage(&image);
    cvReleaseImage(&pImage8uGray);
    cvReleaseImage(&pImage8uSmooth);
    cvReleaseImage(&pImage16uGraySobel);
    cvReleaseImage(&pImage8uGraySobelShow);
    Invalidate();
    IplImage * pImagePlanes[3]={NULL,NULL,NULL};
    IplImage * pImage16uColorSobel=NULL;
    IplImage * pImage8uColorSobelShow=NULL;
    int i;
    for (i=0;i<3;i++)
    {
        pImagePlanes[i]=cvCreateImage(cvGetSize(image),IPL_DEPTH_8U,1);
    }
    pImage16uColorSobel=cvCreateImage(cvGetSize(image),IPL_DEPTH_16S,1);
    pImage8uColorSobelShow=cvCreateImage(cvGetSize(image),IPL_DEPTH_8U,3);
    //分成 3 个单通道
    cvCvtPixToPlane(image,pImagePlanes[0],pImagePlanes[1],pImagePlanes[2],NULL);
    for (i=0;i<3;i++)
    {
        cvSobel(pImagePlanes[i],pImage16uColorSobel,0,1,3);
        cvConvertScaleAbs(pImage16uColorSobel,pImagePlanes[i],1,0);
    }

```

```

}
cvCvtPlaneToPix(pImagePlanes[0],pImagePlanes[1],pImagePlanes[2],NULL,pImage
8uColorSobelShow);
cvNamedWindow("彩色图像 Sobel 变换",CV_WINDOW_AUTOSIZE);
cvShowImage("彩色图像 Sobel 变换",pImage8uColorSobelShow);

```

7. 下面为图像预处理的拉普拉斯部分程序：

```

IplImage *pyr=0;
IplImage *laplace=0,*laplace1=0,*dst=0, *thr=0;
pyr=cvCreateImage(cvSize(image->width/2,image->height/2),8,3);
laplace=cvCreateImage(cvSize(image->width,image->height),8,3);
laplace1=cvCreateImage(cvSize(image->width,image->height),IPL_DEPTH_8U,1);
dst=cvCreateImage(cvSize(image->width,image->height),IPL_DEPTH_16S,1);
cvLaplace(laplace1,dst,3);
thr=cvCreateImage(cvSize(image->width,image->height),IPL_DEPTH_8U,1);
cvNamedWindow("LOG-二值化",1);
cvShowImage("LOG-二值化",dst);
cvReleaseImage(&thr);//wty
cvReleaseImage(&pyr);
cvReleaseImage(&laplace1);
cvReleaseImage(&dst);
cvReleaseImage(&laplace);

```

8. 下面为图像预处理的拉普拉斯部分程序：

```

cvThreshold(img1,img1,100,255,CV_THRESH_BINARY);
cvNamedWindow("cann",1);
cvShowImage( "cann",img1);
cvWaitKey(0);
cvReleaseImage(&img1);

```

9. 下面为图像预处理的汽车识别部分程序：

```

double m_ratio;
m_ratio=(double)m_Wcar/m_Hcar;
double m_flag,a[3];

```

```

int b;
a[0]=m_ratio-m_CarST[0];
a[1]=m_ratio-m_CarST[1];
a[2]=m_ratio-m_CarST[2];
a[3]=m_ratio-m_CarST[3];
if (a[0]>=0)
{
    a[0]=m_ratio-m_CarST[0];
}
else
{
    a[0]=(m_ratio-m_CarST[0])*(-1);
}
if (a[1]>=0)
{
    a[1]=m_ratio-m_CarST[1];
}
else
{
    a[1]=(m_ratio-m_CarST[1])*(-1);
}
if (a[2]>=0)
{
    a[2]=m_ratio-m_CarST[2];
}
else
{
    a[2]=(m_ratio-m_CarST[2])*(-1);
}
if (a[3]>=0)
{
    a[3]=m_ratio-m_CarST[3];
}

```

```
}  
else  
{  
  a[3]=(m_ratio-m_CarST[3])*(-1);  
}  
m_flag=a[0];  
b=0;  
if (m_flag>a[1])  
{  
  m_flag=a[1];  
  b=1;  
}  
if (m_flag>a[2])  
{  
  m_flag=a[2];  
  b=2;  
}  
if (m_flag>a[3])  
{  
  m_flag=a[3];  
  b=3;  
}
```