

基于 OpenCV 的计算机 视觉技术实现

陈胜勇 刘 盛等 编著

科学出版社

北 京

内 容 简 介

OpenCV 是用来实现计算机视觉相关技术的开放源码工作库,是计算机视觉、图像处理、模式识别、计算机图形学、信号处理、视频监控、科学可视化等相关从业人员的好工具。本书介绍了大约 200 多个典型的技术问题,覆盖了基于 OpenCV 基础编程的主要内容,利用大量生动有趣的编程案例和编程技巧,从解决问题和答疑解惑入手,以因特网上最新资料为蓝本,深入浅出地说明了 OpenCV 中最典型和用途最广的程序设计方法。全书结构清晰、合理,范例实用、丰富,理论结合实践,即使读者只是略懂计算机视觉原理,也能入手对相关理论方法直接进行编码实现。

本书可供广大科研人员、工程技术人员、高校相关专业师生及计算机视觉和图像编程爱好者阅读参考。

图书在版编目(CIP)数据

基于 OpenCV 的计算机视觉技术实现/陈胜勇,刘盛等编著.—北京:科学出版社,2008

ISBN 978-7-03-021210-8

I.基… II.①陈…②刘… III.①图像处理-C语言-程序设计②计算机视觉 IV.TP312 TP391.41 TP302.7

中国版本图书馆 CIP 数据核字(2008)第 035227 号

责任编辑:王志欣 任 静 / 责任校对:朱光光
责任印制:刘士平 / 封面设计:耕者工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

中国科学院印刷厂印刷

科学出版社发行 各地新华书店经销

*

2008 年 5 月第 一 版 开本:B5(720×1000)

2008 年 5 月第一次印刷 印张:30 1/2 插页 00

印数:1—4 000 字数:598 000

定价:60.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

《基于 OpenCV 的计算机视觉技术实现》

编写人员名单

编著者 陈胜勇 刘 盛 姚春燕 毛国红
王中杰 林 陶 杜云鹤 唐 诗
范 芸 杨庆丰 方银锋 管 秋

主 审 王万良

前 言

计算机视觉和图像处理是一门新兴的学科,在经过大约 30 年的发展后,目前正向两个相反方向生根长叶:一方面在研究理论上往更深、更高层次上发展(即视觉感知理论);另一方面,很多已经趋于成熟的技术则往工程实用化方向发展(即实现细节)。然而无论哪个方面,它们有一个至关重要的共同环节,就是程序实现!如果没有对基础理论进行编程实现,以方便地调用现有的基本处理过程,理论研究便是空中楼阁,没有测试基础;而如果没有对视觉和图像技术进行编程实现,工程实用化更是无从说起。本书结合理论说明和 OpenCV 编程实现,在计算机视觉领域起到承上启下的作用。

OpenCV 是用来实现计算机视觉相关技术的开放源码工作库,是图像处理、计算机视觉、模式识别、计算机图形学、信号处理、视频监控、科学可视化等相关从业人员的好工具,已经被数以万计的科研人员、技术工程师、软件开发人员、研究生所使用。目前,我国有大量计算机视觉领域的从业人员,虽然每年有数以千计的计算机视觉图书和学位论文,然而很少有专门用于计算机视觉编程开发的参考书。所幸的是,最早落户于美国英特尔服务器的全球计算机视觉开放源码 OpenCV (open source computer vision library) 为广大从业人员提供了极大的方便。OpenCV 对非商业用途和商业用途都免费,由一系列 C 和 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。本书由浅入深,理论结合实践,详细说明基于 OpenCV 的 C++ 编程方法,即使读者只是计算机视觉的初学者,也能轻易上手对相关理论方法直接进行编码实现。

本书内容共分 18 章,涉及 200 多个技术问题,覆盖了基于 OpenCV 基础编程的大部分内容。利用大量生动有趣的编程案例、编程技巧,从答疑解惑和解决问题入手,以因特网上最新资料为蓝本,以简洁明快的语言、清晰直观的条理,全面地对 OpenCV 编程过程中常见问题及故障给予了具体解决办法和答案。深入浅出地说明了 OpenCV 中最典型和用途最广的程序设计方法。并且结合作者所在计算机视觉研究团队多年来的研究实践经验对各个专题作较全面的解述。

全书结构清晰、合理,范例实用、丰富。对广大从事图像处理、计算机视觉、模式识别、计算机图形学、信号处理、视频监控、科学可视化等相关领域的科研技术人员、研究生以及高年级本科生等,不失为一本重要的自学、教学参考书。本书将帮助您实现计算机视觉相关领域中一些高深的、奇妙的、急需的或者是有趣的想法!

本书获浙江工业大学专著与研究生教材出版基金资助(20070205)。本书涉及作者的科研成果是在国家自然科学基金(60405009, 60605013, 60573123), 浙江省重点科技计划项目(2006C21002), 浙江省自然科学基金项目(Y105101), 浙江省教育厅科研计划项目(20051450, 20060864)等的资助下取得的。国内外每年发表的关于计算机视觉方面的论文达数千篇, 在本书的撰写过程中参考了其中大量的学术论文, 直接引用的有数十篇, 一些程序代码和翻译内容还来自众多国内作者发表在网络上的共享论文和资源, 在此向相关作者表示感谢。

由于作者水平有限, 书中不足在所难免, 恳请广大读者和同行批评指正。

编 者

2008 年于杭州

目 录

前言	
第一章 使用 OpenCV 实现计算机视觉技术	1
1.1 计算机视觉技术	1
1.2 什么是 OpenCV	8
1.3 基于 OpenCV 库的编程方法	11
本章小结	16
第二章 OpenCV 的编程环境	17
2.1 OpenCV 环境介绍	17
2.2 OpenCV 的体系结构	25
2.3 OpenCV 实例演示	33
本章小结	39
第三章 OpenCV 编程风格	40
3.1 命名约定	40
3.2 结构	41
3.3 函数接口设计	43
3.4 函数实现	44
3.5 代码布局	45
3.6 移植性	46
3.7 文件操作	46
3.8 文档编写	52
本章小结	54
第四章 数据结构	55
4.1 基本数据结构	55
4.2 数组有关的操作	59
4.3 动态结构	85
本章小结	95
第五章 数据交互	96
5.1 绘图函数	96
5.2 文件存储	106
5.3 运行时类型信息和通用函数	118

5.4	错误处理函数	120
5.5	系统函数	122
	本章小结	124
第六章	图像处理	125
6.1	边缘检测	125
6.2	直方图	137
6.3	Hough 变换	142
6.4	几何变换	150
6.5	形态学	155
	本章小结	161
第七章	结构与识别	162
7.1	轮廓处理函数	162
7.2	计算几何	166
7.3	平面划分	176
7.4	目标检测函数	183
7.5	生成与控制贝塞尔曲线	190
7.6	用 OpenCV 进行人脸检测	197
	本章小结	203
第八章	图形界面 (HighGUI)	204
8.1	读取和保存图像	204
8.2	OpenCV 中的实用系统函数	216
	本章小结	224
第九章	视频处理 (CvCAM)	225
9.1	使用 HighGUI 对视频进行读写处理	225
9.2	CvCam 对摄像头和视频流的使用	234
	本章小结	242
第十章	OpenCV 附加库第一部分	243
10.1	附加库介绍	243
10.2	形态学 (morphing functions)	249
	本章小结	282
第十一章	OpenCV 附加库第二部分——隐马尔可夫模型	283
11.1	隐马尔可夫模型概述	283
11.2	隐马尔可夫模型中的基本结构与函数介绍	288
11.3	隐马尔可夫模型中的函数介绍	289
11.4	人脸识别工具	294

本章小结	312
第十二章 核心库综合例程	313
12.1 检测黑白格标定板内指定矩形区域内的角点	313
12.2 解线性标定方程组程序	323
本章小结	331
第十三章 运动与跟踪	332
13.1 图像统计的累积函数	332
13.2 运动模板函数	334
13.3 对象跟踪	336
13.4 光流	338
13.5 预估器	341
13.6 Kalman 滤波器跟踪示例	345
13.7 用 Snake 方法检测可变形体的轮廓	349
13.8 运动目标跟踪与检测	354
本章小结	363
第十四章 立体视觉第一部分——照相机定标	364
14.1 坐标系介绍	364
14.2 透视投影矩阵的获得	366
14.3 摄像机参数的获取	368
14.4 径向畸变的校正	370
14.5 使用 OpenCV 及 CVUT 进行摄像机定标	371
14.6 OpenCV 中的定标函数	377
14.7 CVUT 介绍	383
本章小结	389
第十五章 立体视觉第二部分——三维重建	390
15.1 极线几何	390
15.2 特征点匹配	394
15.3 三维重建	400
15.4 OpenCV 中相关函数介绍	401
本章小结	403
第十六章 立体视觉第三部分——三维重建算法	404
16.1 图像校正	404
16.2 已校正图像的快速三维重建	410
16.3 Birchfield 算法	411
16.4 OpenCV 中相关函数介绍	418

本章小结	420
第十七章 立体视觉第四部分——立体视觉实例	421
17.1 图像校正实例代码	421
17.2 基于窗口的稀疏点匹配及三维重建之一	422
17.3 基于窗口的稀疏点匹配及三维重建之二	437
17.4 Birchfield 算法的 OpenCV 实现	455
本章小结	456
第十八章 常见问题解疑	457
18.1 安装与编译出错解决方法	457
18.2 OpenCV 库基本技术问题	459
18.3 OpenCV 在 Linux 下的相关问题	464
18.4 OpenCV 库中的陷阱和 bug	467
本章小结	475
参考文献	476

第一章 使用 OpenCV 实现计算机视觉技术

本章首先介绍了计算机视觉的概念及其现状和应用,然后对 OpenCV 作了简要的介绍,并指导读者如何养成良好的编程习惯以达到高效率的编程,最后介绍基于开源代码 OpenCV 进行程序开发的基本方法。此外还给出了用 OpenCV 编程时常用的资料信息及网络资源链接。

1.1 计算机视觉技术

人眼与大脑的协作使得人们可以获得、处理及理解视觉信息。人类利用视觉感知外界环境信息的效率很高,事实上人类获取的环境信息中 80%左右是通过视觉得到的。视觉信息来自于环境中的可见光,环境中的物体在可见光照射下,在人眼的视网膜上形成图像,通过感光细胞转换为神经脉冲信号并通过神经纤维传入大脑皮层进行处理与理解。所以,视觉并不仅指对光信号的感受,它包括对视觉信息的获取、传输、处理、储存以及理解的全过程。那么,究竟什么是计算机视觉技术呢?

1.1.1 计算机视觉技术

近年来,随着计算机技术和数字信号处理技术的迅猛发展,人们用摄像机获取环境图像并将其转换成数字信号,且利用计算机实现对视觉信息处理的全过程,这就是计算机视觉技术的起源。

计算机视觉成为一门独立的学科,至少可以从美国麻省理工学院(MIT)Marr教授这一代人所做的奠基工作开始追溯,因此这门学科至少已有二三十年的历史。20世纪50年代归入模式识别——主要集中在二维图像分析和识别上,例如,光学字符识别、工件表面、显微图片和航空图片的分析和解释等。60年代MIT的Roberts通过计算机程序从数字图像中提取出诸如立方体、楔形体、棱柱体等多面体的三维结构,并对物体形状及物体的空间关系进行描述。Roberts的研究工作开创了以理解三维场景为目的的三维计算机视觉的研究,Roberts对积木世界的创造性研究给人们以极大的启发。许多人相信,一旦由白色积木玩具组成的三维世界可以被理解,则可以推广到理解更复杂的三维场景。在70年代,已经出现了一些视觉应用系统。70年代中期麻省理工学院人工智能(AI)实验室正式开设“机器视觉”(machine vision)课程,由Horn教授讲授。1977年,Marr教授提出了不同

于“积木世界”分析方法的计算视觉理论——Marr 视觉理论,该理论在 80 年代成为计算机视觉研究领域中的一个十分重要的理论框架。到 80 年代中期,计算机视觉获得了迅速发展,主动视觉理论框架、基于感知特征群的物体识别理论框架等新概念、新方法、新理论不断涌现。而到 90 年代,计算机视觉在工业环境中得到广泛应用,同时基于多视几何的视觉理论得到迅速发展。纵观计算机视觉发展的历史,由于计算机视觉具有的潜在应用十分广泛,所涉及的学科知识极其繁多,研究的问题又极富挑战性,因此它一直是计算机学科中的一个热门领域,并受到了许多从事心理学、神经科学、生理学、生物物理学、数学与计算机科学等学科的研究人员的关注,从而把图像处理、模式识别、人工智能、数学、认知科学、机器学习、计算机图形学等方面的研究成果融汇进来。

计算机视觉是指用计算机实现人的视觉功能——对客观世界的三维场景的感知、识别和理解,这意味着计算机视觉技术的研究目标是使计算机具有通过二维图像认知三维环境信息的能力。因此不仅需要使机器能感知三维环境中物体的几何信息(形状、位置、姿态、运动等),而且能对它们进行描述、存储、识别与理解。可以认为,计算机视觉与研究人类或动物的视觉是不同的:它借助于几何、物理和学习技术来构筑模型,从而用统计的方法来处理数据。这里主要有两类方法:一类是仿生学的方法,即参照人类视觉系统的结构原理,建立相应的处理模块完成类似的功能和工作;另一类是工程的方法,即从分析人类视觉过程的功能着手,并不去刻意模拟人类视觉系统内部结构,而仅考虑系统的输入和输出,并采用任何现有的可行的手段实现系统功能。其中的第二类方法是目前计算机视觉技术研究的趋向,目前已经发展起一套独立的计算理论与算法。

因此,在透彻理解摄像机性能与物理成像过程的基础上,计算机视觉技术需要实现的是,对每个像素进行简单的推理,将在多幅图像中可能得到的信息合成和谐的整体,确定像素集之间的联系以便将它们彼此分割开或推断一些形状信息,使用几何信息或概率统计技术来识别物体。

常引起争议的一个问题是计算机视觉技术和机器视觉技术的定义如何界定。本书给出当前一种较为合理的分析:计算机视觉技术的研究目标是使计算机具有通过一幅或多幅图像认知周围环境的能力(包括对客观世界三维环境的感知、识别与理解)。这意味着计算机不仅要模拟人眼的功能,而且更重要的是使计算机完成人眼所不能胜任的工作。而机器视觉则是建立在计算机视觉理论基础之上,偏重于计算机视觉技术的工程化,能够自动获取和分析特定的图像,以控制相应的行为。与计算机视觉所研究的视觉模式识别、视觉理解等内容不同,机器视觉技术重点在于感知环境中物体的形状、位置、姿态、运动等几何信息。在计算机视觉研究中,经常使用几何模型、复杂的知识表达,采用基于模型的匹配和搜索技术,搜索的策略常使用自底而上、自顶而下、分层和启发式控制策略。机器视觉是一门新兴的

发展迅速的学科,20 世纪 80 年代以来,机器视觉的研究经历了从实验室走向实际应用的发展阶段。从简单的二值图像处理到高分辨率多灰度的图像处理,从一般的二维信息处理到三维视觉机理以及模型和算法的研究都取得了很大的进展。而计算机工业水平的飞速提高以及人工智能、并行处理和神经元网络等学科的发展,更促进了机器视觉系统的实用化和涉足许多复杂视觉过程的研究。目前,机器视觉系统正在广泛地应用于视觉检测、机器人的视觉引导和自动化装配领域中。机器视觉是个相对较新的技术,它为制造业在提高产品质量、提高生产效率和操作安全性上提供了许多技术支持。机器视觉越来越受欢迎有这样一个主要原因:光学传感具有与生俱来的清洁性、安全性和通用性。用视觉有可能去做其他人不知道用传感方法可以做到的事情,比如识别污点、铁锈或者表面腐蚀等。具体说来计算机视觉为机器视觉提供图像和景物分析的理论及算法基础,机器视觉则为计算机视觉的实现提供传感器模型、系统构造和实现手段。

通过上面的定义,可以看出,其实机器视觉和计算机视觉并没有很清晰的界限,而是紧密的联系在一起,它们有着相同的理论,只是在应用中根据具体实际应用目标的不同而不同。计算机视觉与机器视觉都是要从图像或图像序列中获取对世界的描述,因此,对基本层的图像获取、图像处理,中层的图像分割、图像分析和高层的图像理解这些理论知识的掌握对两者来说都是“万变不离其宗”。正因为两者关系密切且具有共同的计算理论框架、模型和算法,所以实际中并不加以严格划分,对于工业应用常使用“机器视觉”,而一般情况下则常用“计算机视觉”。

1.1.2 Marr 的视觉理论框架

MIT AI 实验室吸引了国际上许多知名学者参与计算机视觉的理论、算法、系统设计的研究,Marr 教授就是其中的一位。他于 1973 年应邀在 MIT AI 实验室领导一个以博士生为主体的研究小组,1977 年提出了不同于“积木世界”分析方法的计算视觉理论(computational vision),该理论在 20 世纪 80 年代成为计算机视觉研究领域中的一个十分重要的理论框架。框架首次从信息处理的角度综合了图像处理、心理物理学、神经生物学及临床神经学等方面已取得的重要研究成果,是第一个较为完善的视觉系统框架,并且使计算机视觉研究有了一个比较明确的体系。此框架尽管存在很多缺陷,但过去 20 多年一直处于主导地位,以下简称该理论框架。

Marr 的视觉理论将视觉系统研究划分为三个层次,即:

- (1) 计算理论层次,给出系统各部分的计算目的和策略。
- (2) 表达与算法层次,给出各部分的信息表达和实现各部分目标的具体算法。
- (3) 硬件层次,回答如何用硬件实现以上算法。

按照 Marr 的理论,计算视觉理论要回答视觉系统的计算目的和策略是什么,

或视觉系统的输入和输出是什么,如何由系统的输入求出系统的输出。在这个层次上,信息系统的特征是将一种信息(输入)映射为另一种信息(输出)。例如,系统输入是二维灰度图像,输出则是三维物体的形状、位置和姿态,视觉系统的任务就是如何建立输入输出之间的关系和约束,如何由二维灰度图像恢复物体的三维信息。表示与算法层次是要进一步回答如何表示输入和输出信息,如何实现计算理论所对应的功能的算法,以及如何由一种表示变换成另一种表示,比如创建数据结构和符号。一般来说,不同的输入、输出和计算理论,对应不同的表示,而同一种输入、输出或计算理论也可能对应若干种表示。在解决了理论问题和表示问题后,最后一个层次是解决用硬件实现上述表示和算法的问题,例如,计算机体系结构及具体的计算装置及其细节。从信息处理的观点来看,至关重要的乃是最高层次,即计算理论层次。这是因为构成知觉的计算本质,取决于解决计算问题本身,而不取决于用来解决计算问题的特殊硬件。换句话说,通过正确理解待解决问题的本质,将有助于理解并创造算法。如果考虑解决问题的机制和物理实现,则对理解算法往往无济于事。

上述三个层次之间存在着逻辑的因果关系,但它们之间的联系不是十分紧密,因此,某些现象只能在其中一个或两个层次上进行解释。比如神经解剖学原则上与第三层次,即物理实现联系在一起。突触机制、动作电位、抑制性相互作用都在第三个层次上。心理物理学与第二层次(即表示与算法)有着更直接的联系。更一般地说,不同的现象必须在不同的层次上进行解释,这会有助于人们把握正确的研究方向。例如,人们常说,人脑完全不同于计算机,因为前者是并行加工的,后者是串行的。对于这个问题,应该这样回答:并行加工和串行加工是在算法这个层次上的区别,而不是根本性的区别,因为任何一个并行的计算程序都可以写成串行的程序。因此,这种并行与串行的区别并不支持这种观点,即人脑的运行与计算机的运算是不同的,因而人脑所完成的任务是不可能通过编制程序用计算机来完成。

Marr 将视觉过程划分为三个阶段,如图 1-1 所示。

第一阶段(也称为早期阶段)是将输入的原始图像进行处理,抽取图像中诸如角点、边缘、纹理、线条、边界等基本特征,这些特征的集合称为要素图(primitive sketch);第二阶段(中期阶段)是指在以观测者为中心的坐标系中,由输入图像和基元图恢复场景可见部分的深度、法线方向、轮廓等,这些信息中包含了深度信息,但不是真正的物体三维表示,因此,称为二维半图(2.5 dimensional sketch);在以物体为中心的坐标系中,由输入图像、要素图、二维半图来分层次组成若干三维模型,每个三维模型都是在几个轴线空间的基础上构成的,所有体积基元或面积形状基元都附着在轴线上,这一恢复、表示和识别三维物体的过程称为视觉的第三阶段(后期阶段)。

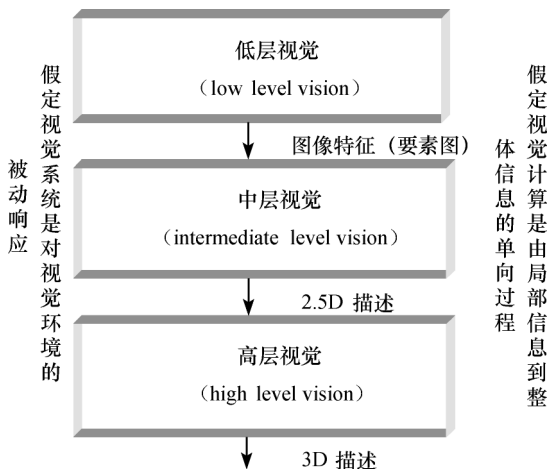


图 1-1 Marr 视觉计算理论框架

Marr 理论也有其不足之处,其中有四个有关整体框架的问题:

- (1) 框架中输入是被动的。
- (2) 框架中加工的目的不变,总是恢复场景中物体的位置和形状等。
- (3) 框架缺乏或者说未足够重视高层知识的指导作用。
- (4) 整个框架中信息加工过程基本自下而上,单向流动,没有反馈。

另外, Marr 没有研究如何用数学方法严格描述视觉信息的问题,虽然较充分地研究了早期视觉,但基本没有论及对视觉知识的表达、使用和基于视觉知识的识别等。虽然存在这些不足,但 Marr 的理论使人们对视觉信息的研究有了明确的内容和较完整的基本体系,仍被看做是研究的主流;新提出的理论框架多数被看做是它的补充和发展,均包含 Marr 视觉理论的基本成分。

1.1.3 计算机视觉技术的研究内容

计算机视觉研究大致可划分为以下五大研究内容。

1. 输入设备(input device)

输入设备包括成像设备和数字化设备。成像设备是指通过光学摄像机或红外、激光、超声、X 射线对周围场景或物体进行探测成像,得到关于场景或物体的二维或三维数字化图像的仪器。

2. 低层视觉(low level vision)

低层视觉主要是对输入的原始图像进行处理。这一过程借用了大量的图像处理技术和算法,如图像滤波、图像增强、边缘检测等,以便从图像中抽取诸如角点、

边缘、线条、边界以及色彩等关于场景的基本特征；这一过程还包含了各种图像变换(如校正)、图像纹理检测、图像运动检测等。

3. 中层视觉(intermediate level vision)

中层视觉主要任务是恢复场景的深度、表面法线方向、轮廓等有关场景的 2.5 维信息,实现的途径有立体视觉(stereo vision)、测距成像(range finder)、运动估计(motion estimation)、明暗特征、纹理特征等所谓的从 X 恢复形状的估计方法。系统标定、系统成像模型等研究内容一般也是在这个层次上进行的。

4. 高层视觉(high level vision)

高层视觉主要任务是在以物体为中心的坐标系中,在原始输入图像、图像基本特征、2.5 维图的基础上,恢复物体的完整三维图,建立物体三维描述,识别三维物体并确定物体的位置和方向。

5. 体系结构(system architecture)

体系结构这一术语最通常的含义是指在高度抽象的层次上,根据系统模型而不是根据实现设计的具体例子来研究系统的结构。为了说明这一点,可以考虑建筑设计中某一时期的建筑风格(如清朝时期)和根据这一风格设计出来的具体建筑之间的区别。体系结构研究涉及一系列相关的课题:并行结构、分层结构、信息流结构、拓扑结构以及从设计到实现的途径。

1.1.4 计算机视觉与其他学科研究领域之间的关系

计算机视觉技术是一种典型的交叉学科研究领域,因此存在与其他许多学科或研究方向之间相互渗透、相互支撑的关系,以下作简要介绍。

1. 计算机视觉与图像处理(image processing)

图像处理所实现的功能简单来说是从输入图像到另一种图像的转换。在图像处理中,人是最终的解释者,而在计算机视觉技术中,计算机是图像的解释者。因此,机器视觉的重点是在人的最小干预下,由计算机自动恢复场景信息。图像处理算法在机器视觉系统的早期阶段起着很大的作用,它们通常被用来增强特定信息并抑制噪声。计算机视觉系统必须有图像处理模块存在。

2. 计算机视觉与计算机图形学(computer graphics)

计算机图形学实现的是从三维描述到二维图像显示,而计算机视觉则是从二维图像数据到三维描述。因此,计算机图形学属于图像综合,计算机视觉属于图像

分析,在一定意义上讲,计算机视觉是计算机图形学的逆问题。二者间的关系如图 1-2 所示。

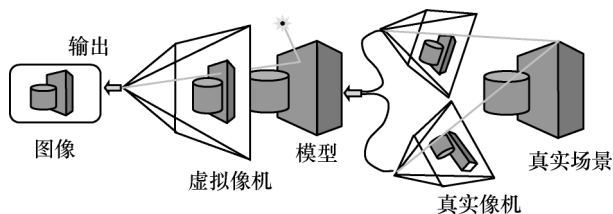


图 1-2 计算机视觉与计算机图形学的融合

这两个领域在其发展的早期阶段是没有什么联系的,但是经过近十几年来的发展变得越来越相近了。计算机视觉使用了计算机图形学中的曲线和曲面表示方法以及其他的一些技术,而计算机图形学也使用计算机视觉技术,以便在计算机中建立逼真的图像模型。可视化和虚拟现实把这两个领域紧密地联系在一起。两者从最初相互独立的平行发展到现在的融合是一大趋势。

3. 计算机视觉与模式识别(pattern recognition)

模式识别技术研究各类模式的分类,模式可以有不同的物理意义和表现形式,其中,图像模式的分类是计算机视觉中的一个重要问题。模式识别主要用于识别各种符号、图画等平面图形。模式一般指一类事物区别于其他事物所具有的共同特征。模式识别方法主要有统计方法和句法方法两种,统计方法是指从模式抽取一组特征值,并以划分特征空间的方法来识别每一个模式;句法方法是指利用一组简单的子模式(模式基元)通过文法规则来描述复杂的模式。模式识别中的许多方法可以应用于计算机视觉中。

4. 计算机视觉与人工智能(artificial intelligence, AI)

人工智能技术主要研究智能系统的设计和有关智能的计算理论与方法。人工智能可被分为三个阶段:感知、认知和动作执行。计算机视觉常被视为 AI 的一个分支。

5. 神经网络(artificial neural network, ANN)

神经网络是一种信息处理系统,它是由大量简单的处理单元(称为神经元)通过具有强度的连接(connection)相互联系起来,实现并行分布式处理(parallel distribution processing, PDP)。神经网络的最大特点是可以通过改变连接强度来调整系统,使之适应复杂的环境,实现类似人的学习、归纳和分类等功能。

神经网络已经在许多工程技术领域得到了广泛的应用。神经网络作为一种方法和机制将用于解决计算机视觉中的许多问题。

6. 神经生物物理学(neurobiophysics)与认知科学(cognitive science)

神经生物物理学与认知科学长期将人类视觉作为主要的研究对象。机器视觉中已有的许多方法与人类视觉极为相似。目前,许多机器视觉研究者对研究人类视觉计算模型比研究机器视觉系统更感兴趣,希望机器视觉更加自然化,更加接近生物视觉。

7. 计算机视觉与心理物理学(psychophysics)

心理物理学主要研究人类自身视觉系统的有关机理与现象,而心理物理学研究结果往往被用来作为建立计算机视觉模型的理论依据。

1.2 什么是 OpenCV

OpenCV 是一个很优秀的开源图像处理算法库,它包含了一系列 C 函数和少量 C++ 类,实现了图像处理和计算机视觉方面的很多通用算法。OpenCV 拥有包括 300 多个 C/C++ 函数的跨平台的中、高层 API。它不依赖于其他的外部库(尽管也可以使用某些外部库)。通过调用 OpenCV 算法库,研究者们可以在前人已完成的成熟算法基础上迅速地开展自己的工作。作为一个基本的计算机视觉、图像处理和模式识别的开源项目,OpenCV 可以直接应用于很多领域,作为二次开发的理想工具。

另外,OpenCV 也为 Intel 公司的 integrated performance primitives (IPP) 提供了透明接口。这意味着如果有为特定处理器(当然是 Intel 的处理器了)优化的 IPP 库,OpenCV 将在运行时自动加载这些库。

更多关于 IPP 的信息请参考:<http://www.intel.com/software/products/ipp/index.htm>。

1. 谁创建了 OpenCV

作者列表可以在文件 AUTHORS 中找到。此外,还有很多人给出了建议、补丁、BUG 报告等。一个不太完整的列表在文件 THANKS 中。

2. 谁可以用 OpenCV

OpenCV 是开放源码。在网页:<http://creativecommons.org/licenses/by/2.5/cn/deed.zh> 上有“知识共享署名 2.5 中国大陆许可协议”。其“知识共享”声

明如图 1-3 所示(针对中国大陆地区)。

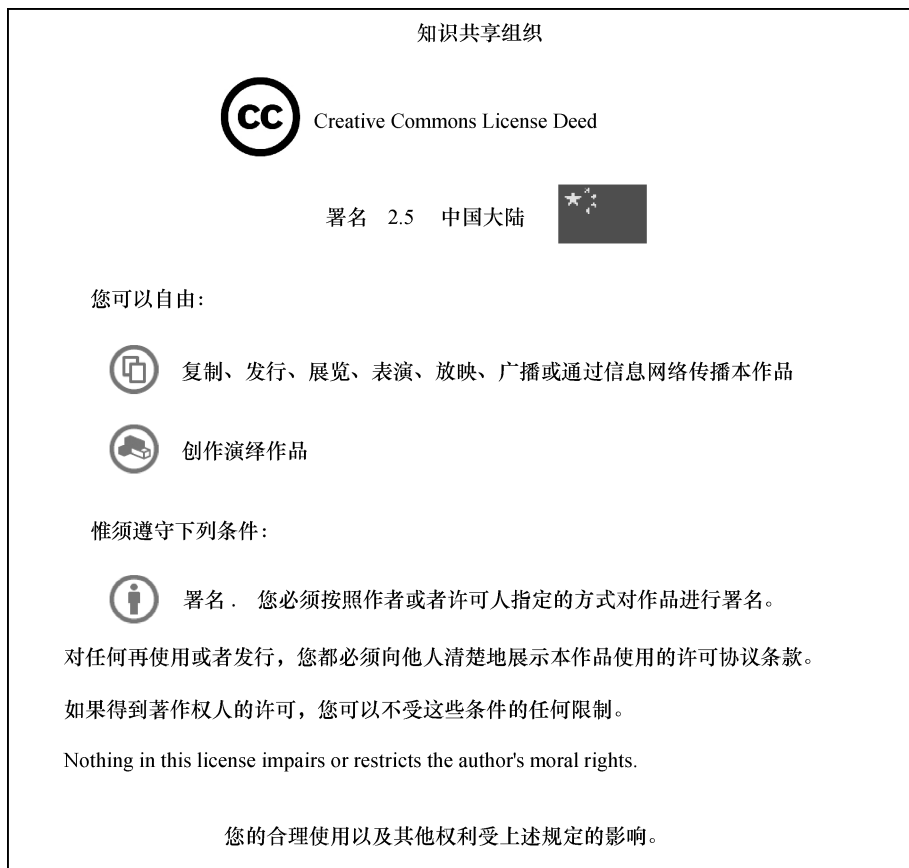


图 1-3 知识共享声明

3. 从哪里下载 OpenCV

访问 <http://www.sourceforge.net/projects/opencvlibrary>。如果有问题,请在 Google(<http://www.google.com>)中输入“OpenCV”进行搜索。

4. OpenCV 库的特点及组成

OpenCV 库拥有以下显著优点:①跨平台——Windows, Linux。独立于操作系统、硬件和图形管理器。②免费(开放源代码)——无论非商业或商业应用,OpenCV 都是免费的(细节参考发布版本的 license)。这一最大优点使之成为当前计算机视觉研究人员的最爱。③速度快、使用方便。④可扩展性好,包括底层和高层的应用开发包。⑤通用的图像/视频载入、保存和获取模块。

OpenCV 库目前包括如图 1-4 所示的模块。① CxCore: 包括一些基本结构和算法函数, 如数据结构与线性代数支持, 主要提供对各种数据类型的基本运算功能。② Cv: 图像处理和计算机视觉功能, 是主要的 OpenCV 函数, 包括图像处理、结构分析、运动分析、物体跟踪、模式识别及摄像机标定等功能。③ CvAux: OpenCV 附加库函数, 包含一些实验性的函数, 如 View Morphing, 三维跟踪, PCA, HMM 等函数。④ HighGUI: 用户交互部分 (GUI, 图像视频 I/O, 系统调用函数)。⑤ ml: 新加入的机器学习模块, 目前的主要内容为分类器。

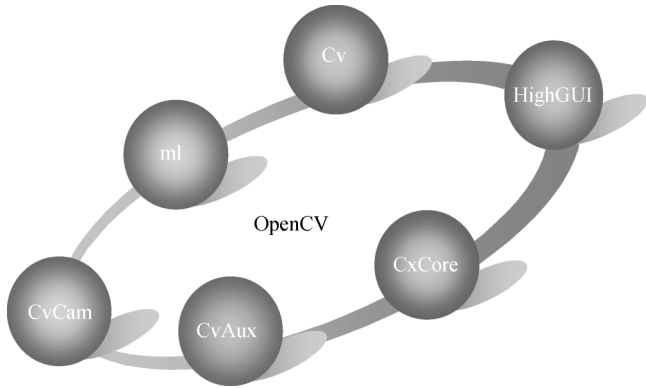


图 1-4 OpenCV 的组成结构

另外还有一个模块 CvCam, 不过在 Linux 版本中已经抛弃。而 Windows 版本中将 DirectX 支持加入 HighGUI 后, CvCam 将被彻底去掉。

OpenCV 库是一个扩充性很好的算法库, 其模块可以自由添加, 从而不断丰富其功能, 因此在进行 OpenCV 编程时, 要能不断接受新思想、新方法 (事实上这正是 OpenCV 开源库的目的和意义)。常常访问 OpenCV 相关网站和论坛也不失为一种很好地及时把握 OpenCV 库最新内容的方法。

目前而言, 以上所述的各模块具体包括以下功能: ① 图像数据的操作 (分配、释放、复制、设置和转换)。② 图像和视频的输入输出 I/O (文件与摄像头的输入、图像和视频文件输出)。③ 矩阵和向量的操作以及线性代数的算法程序 (矩阵积、解方程、特征值以及奇异值等)。④ 各种动态数据结构 (列表、队列、集合、树、图等)。⑤ 基本的数字图像处理 (滤波、边缘检测、角点检测、采样与差值、色彩转换、形态操作、直方图、图像金字塔等)。⑥ 结构分析 (连接部件、轮廓处理、距离变换、各自距计算、模板匹配、Hough 变换、多边形逼近、直线拟合、椭圆拟合、Delaunay 三角划分等)。⑦ 摄像头定标 (发现与跟踪定标模式、定标、基本矩阵估计、齐次矩阵估计、立体对应)。⑧ 运动分析 (光流、运动分割、跟踪)。⑨ 目标识别 (特征法、隐马尔可夫模型; HMM)。⑩ 基本的 GUI (图像与视频显示、键盘和鼠标事件处理、滚动条)

和图像标注(线、二次曲线、多边形、画文字)。

在后面的章节中,将按模块分别详细讨论其具体内容,这里不再赘述。

从计算机视觉技术的研究现状中可以看到,OpenCV 拥有大量的用户群,并不断得到扩展。然而,想在 Visual C++ 平台中灵活自如地实现 OpenCV 编程,编程人员必须具备良好的编程习惯和高效的编程方法,而且必须结合 OpenCV 库的具体特点来进行。接下来的内容将讨论这一问题。

1.3 基于 OpenCV 库的编程方法

作为当前计算机科技中的一大研究热点,计算机视觉正吸引着越来越多研究者的目光。开发出一个实用的计算机视觉应用系统,一直是很多研究人员的希望。但是,实现计算机视觉中众多算法、完成系统编程工作却是一大重复性的难题。拥有良好的编程习惯和编程风格将使计算机视觉系统的开发工作更为高效。

1.3.1 事半功倍——良好的编程习惯和高效编程技巧

大量关于 VC++ 编程技巧的书籍都给程序设计人员提供了养成良好编程习惯和技巧的建议。在此,为读者列出如下几点常规建议。

(1) 全面系统地分析和规划。编程之前必须已经非常清楚自己下面所做的内容以及各个模块之间的关系,换句话说,系统分析要尽量完善。

(2) 养成良好的文档习惯。良好的文档是正规研发流程中非常重要的环节。一个好的程序是先写好设计文档再进行编程,在设计文档的指导下,才能写出安全的代码。只要能将设计思想及实现算法或步骤描述清楚就是好的文档,写文档时要善于用图表来说明设计者的思想。

(3) 充分利用开放源代码。开源软件的优点包括:①可以灵活地选择最能满足用户需要的解决方案。②创新速度快,版本更新周期短。③从发现软件缺陷到修复缺陷的时间缩短。④没有直接的许可证费用。⑤总拥有成本较低,因为无需支付升级费用或购买软件。

(4) 代码风格要规范严谨,格式清晰,避免错乱不堪。变量函数命名规范化。

(5) 代码效率要高,减少代码重用率,尽量将功能封装成函数或者过程。

(6) 充分利用 MSDN,它远胜任何一本编程参考书。MSDN 是 Microsoft 当前提供的有关编程信息的最全面的资源,它包含微软最新的技术数据库、易学易用的全文检索功能,可以让使用者迅速找到任何需要的技术参考数据,随时拥有与全世界精英同步的技术,掌握最丰富的程序开发资源。事实上,在编程中遇到的很多问题,都可以在 MSDN 中找到答案!

(7) 掌握好跟踪调试技巧。跟踪调试程序是一件烦琐而又复杂的事情,掌握

必要的调试策略及技巧却可以使这些工作变得轻松起来。

(8) 多写注释。这一点非常重要！设想一个程序员两个月以后再去看自己没有注释的上万行代码……结果让人可想而知。

(9) 系统要具有一定可扩展性,各个模块之间追求低耦合度。这样做便于模块的重复利用和后续的修改与维护;不能奢望在一个版本中完成所有的功能,应该尽力完善几个关键的部分,但是可以留下接口。

(10) 养成自我测试的习惯。测试工作应由测试工程师来做,但在写完一个模块或一个软件时,还是要亲自先测试一下,保证不要出现一些低级的错误。

(11) 阶段性地做专题总结。知识要温故而知新,在此建议读者要养成阶段性地做专题总结的习惯,比如,本月学习或在做与多线程有关的模块或项目,那么当项目完成后,就可以好好地总结一下所有与多线程相关的技术,包括理论知识,实践方法以及各种技巧及优秀文章等,这对各种能力的提高将有很大的帮助。

(12) 善于通过各种渠道寻求帮助。善于交流和沟通,特别是经常与一些编程高手交流一下学习的心得体会,另外,从网络寻求问题的解答也是一大渠道(比如编程技术论坛等信息量大的网络资源)。

(13) 最后,当然也是最重要的一点:要有持之以恒的精神。当然,要利用 OpenCV 库完成计算机视觉软件开发工作,必须结合 OpenCV 库的特点来开展,因此讨论 OpenCV 编程方法十分必要。下一部分简单介绍基于 OpenCV 库的编程方法。

1.3.2 基于 OpenCV 的基本编程方法

基于 OpenCV 的程序开发过程大致可分为以下四个步骤。

1. 首先获取最新版本的 OpenCV

从 <http://www.sourceforge.net/projects/opencvlibrary> 获取官方版本(目前 OpenCV 最新的 Windows 版本是 Version 1.0)。

从“cvs-z9-d; pserver: anonymous@cvs.sourceforge.net:/cvsroot/opencvlibrary”获取最新的 CVS(源代码版本控制软件)版本。

2. 安装和配置

在安装好 Visual C++ 的 Windows 操作系统下运行 OpenCV 安装文件 OpenCV 1.0.exe,完成安装过程。打开 opencv.dsw 选择“全部编译”,把 opencv/bin 目录加入系统路径。安装好 OpenCV 库后,只需简单配置即可在自己的程序中使用它,具体的安装和配置步骤将在第二章中详细讨论,此处不作详细说明。

3. 编程

生成一个自己的项目(参考 opencv/docs/faq.htm),然后在 VC 项目属性中设好 OpenCV 头文件以及库文件的路径(详见第二章),包括进相应的头文件并根据不同需要调用 OpenCV 中的函数,完成程序编制。

4. 编译并运行程序

OpenCV 如何安装? 怎样基于 OpenCV 开发自己的视觉系统? 当使用 OpenCV 编程时,如何找到帮助和调试? 这些问题是 OpenCV 初学者常遇到的,如果在安装/运行/使用 OpenCV 中遇到类似的问题,请采用以下方法解决:

(1) 阅读中文 FAQ,请参阅安装目录中的 `<opencv-root>\docs\index.htm,faq.htm`(`<opencv_root>`表示安装 OpenCV 后的系统目录,下同)。

(2) 参考 OpenCV 的例子代码,阅读参考手册。这是学习 OpenCV 十分有效的方法,从安装目录中的 `<opencv-root>\example\c` 子目录下可以轻松找到并学习这些非常有用的学习代码,从而快速入门。

(3) 在 OpenCV 邮件列表(<http://tech.groups.yahoo.com/group/OpenCV/>)中搜索。

(4) 加入到“yahoo group”上的 OpenCV 邮件列表中(如何加入请参考 FAQs),并发送问题到邮件列表中(这个邮件列表可能会迁移到 OpenCV's SourceForge site,请访问:<http://www.sourceforge.net/projects/opencvlibrary>)。

充分利用这些资源能很好地解决在使用 OpenCV 中常遇到的问题。下面结合程序 1-1 来说明基于 OpenCV 的基本编程方法。

程序 1-1 使用 OpenCV 的代码示范

```
1. #include <cxcore.h>
2. #include <highgui.h>
3. #include <math.h>
4. int main(int argc,char ** argv) {
5.     CvPoint center;
6.     double scale=-3;
7.     IplImage * image = argc==2 ? cvLoadImage(argv[1]) : 0;
8.     if( ! image) return -1;
9.     center = cvPoint(image->width/2,image->height/2);
```

```
10. for(int i=0;i<image->height;i++)
11.     for(int j=0;j<image->width;j++) {
12.         double dx=(double)(j-center.x)/center.x;
13.         double dy=(double)(i-center.y)/center.y;
14.         double weight=exp((dx * dx+dy * dy) * scale);
15.         uchar * ptr = &CV_IMAGE_ELEM(image,uchar,i,j * 3);
16.         ptr[0] = cvRound(ptr[0] * weight);
17.         ptr[1] = cvRound(ptr[1] * weight);
18.         ptr[2] = cvRound(ptr[2] * weight); }
19. cvSaveImage("copy.png",image);
20. cvNamedWindow("test",1);
21. cvShowImage("test",image);
22. cvWaitKey();
23. return 0; }
```

这段代码实现的是图像的径向梯度变换。若该文件命名为 `test.c`，编译后在命令行运行 `test lena.jpg` (或进入文件所在目录，然后运行 `./test lena.jpg`)，可以得到如图 1-5 的输出效果。



图 1-5 径向梯度变换

从代码中不难看出，OpenCV 程序的编制十分简单，设置好环境和项目以后，

只需要在程序中使用预编译命令包含对应的头文件,即可调用 OpenCV 提供的函数实现相应的功能。

最后,将一些使用 OpenCV 编程时常用的资料信息及网络资源链接汇总如下。

(1) OpenCV 中文站:<http://www.opencv.org.cn/>。

这是国内的 OpenCV 中文官方网站,由中科院自动化所维护。很多有用的中文文档资料及例程都可以在该网站找到,是国内 OpenCV 开发人员必去访问的网络资源。

(2) OpenCV 中文论坛:<http://www.opencv.org.cn/forum>。

(3) OpenCV 英文 wiki:<http://opencvlibrary.sourceforge.net/>。

(4) 项目主页、源代码及文档下载:<http://sourceforge.net/projects/opencv-library/>。

(5) Intel 的 OpenCV 主页:<http://www.intel.com/technology/computing/opencv/index.htm>。

(6) YAHOO OpenCV 的邮件列表及论坛(英文,信息量大):<http://tech.groups.yahoo.com/group/opencv/>。

(7) Hunish 的 OpenCV 专栏(有许多代码实例):<http://blog.csdn.net/hunish>。

(8) CMU(卡耐基-梅隆大学)的计算机视觉主页(计算机视觉方面的资料等):<http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html>。

(9) 书籍:*Open Source Computer Vision Library*。作者是 Gary R. Bradski, Vadim Pisarevsky 和 Jean-Yves Bouguet, Springer 2006 年 6 月出版。

(10) 参考手册:[<opencv-root>\docs\index.htm](http://opencv-root/docs/index.htm)。

(11) 自带例程:

① 视频处理例程及图像处理例程(位于 [<opencv-root>\samples\c\](http://opencv-root/samples/c/) 目录中);

② 色彩跟踪——`camshiftdemo`;

③ 点跟踪——`lkdemo`;

④ 动作分割——`motempl`;

⑤ 视频流边缘检测——`laplace`;

⑥ 边缘检测——`edge`;

⑦ 分割——`pyramid_segmentation`;

⑧ 形态——`morphology`;

⑨ 直方图——`demhist`;

⑩ 距离转换——`disttrans`;

① 椭圆拟合——fitellipse。

本章小结

本章首先对计算机视觉的基本概念、视觉理论框架做了简单介绍,分类讨论计算机视觉技术的研究内容,分析了计算机视觉技术和其他学科的关系。第二部分则简单介绍了 OpenCV 计算机视觉开源算法库。第三部分首先分析如何养成良好的编程习惯以及达到高效率的编程方法,然后结合一个简单的例子分析了基于 OpenCV 库的编程方法。此外还给出了用 OpenCV 编程时常用的资料信息及网络资源链接。

第二章 OpenCV 的编程环境

在第一章中已经认识了 OpenCV 的开源计算机图像库,并且知道它包含了很多 C 和 C++ 的类库,那么接下来就要在 Microsoft Visual C++ 环境下编译运行含有 OpenCV 类库的程序以达到实现计算机视觉图像处理的目的。

本章将介绍 OpenCV 环境及帮助工具的使用,说明 Visual C++ 下的安装与配置,介绍基于 OpenCV 编程时常用的基本类和基本函数,让读者初步了解 OpenCV 应用程序的开发过程,最后通过实例来说明如何在 Visual C++ 集成环境下快速建立应用程序框架。

2.1 OpenCV 环境介绍

Visual C++ 是 Microsoft 公司开发的基于 Windows 操作系统的编程工具。像其他的可视化开发环境(如 Visual Basic, Delphi, C++ Builder)一样,VC++ 集程序的代码编辑、编译、链接和调试于一体,给编程人员提供了一个完整而又方便的开发环境,和许多有效的辅助开发工具。同时,因为 VC++ 是微软的产品,用它来编写 Windows 程序有强大的程序接口和丰富的开发资源的支持,加之 VC++ 严谨的内存管理,在堆栈上良好的分配处理,生成代码的体积小,稳定性高的优点,Visual C++ 就成为目前主流开发工具。由于 OpenCV 的源代码是用 C 和 C++ 所编写而且完全开放,因此可以在 Microsoft Visual Studio 环境下很方便地使用 OpenCV 所提供的库来进行图像处理。在介绍 OpenCV 编程环境之前,首先要来了解如何安装 OpenCV 和基本的编程软件,下面的说明假设已经安装完成 Microsoft Visual C++ 6.0。

2.1.1 OpenCV 的安装

首先从官方网站 <http://www.opencv.org.cn> 下载 OpenCV 安装程序。默认的安装路径为 C:\Program Files\OpenCV(图 2-1~图 2-5 为 OpenCV 1.0 的安装界面,其他版本安装界面与此界面大致相同)。在安装时选择“将 \OpenCV\bin 加入系统变量”(勾选 Add\OpenCV\bin to the system PATH 前面的对号)。

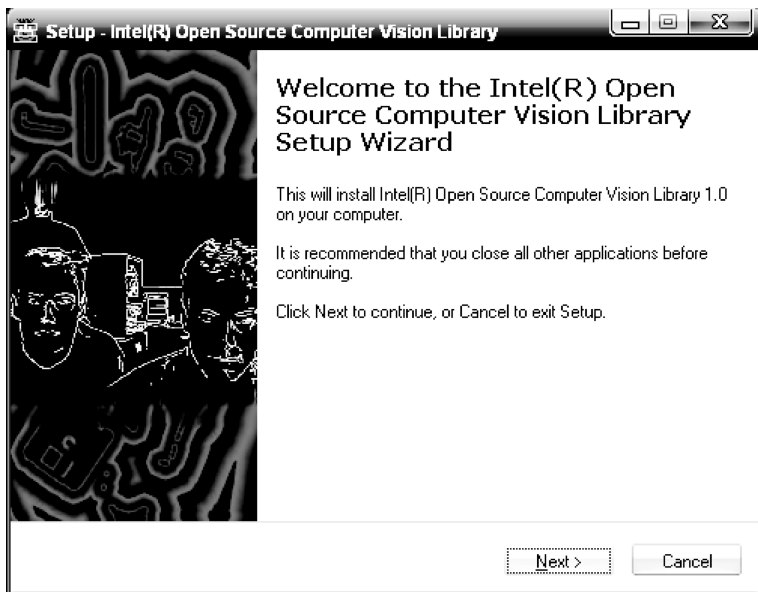


图 2-1 OpenCV 的安装界面(1)

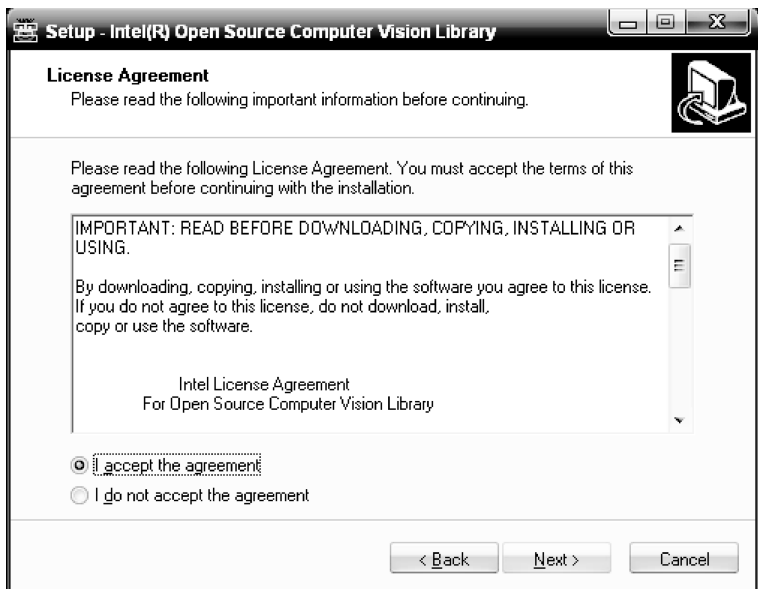


图 2-2 OpenCV 的安装界面(2)

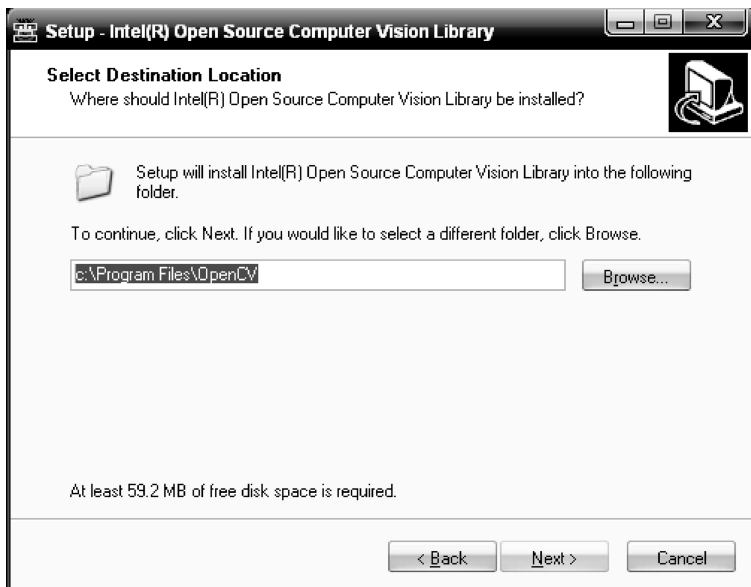


图 2-3 OpenCV 的安装界面(3)

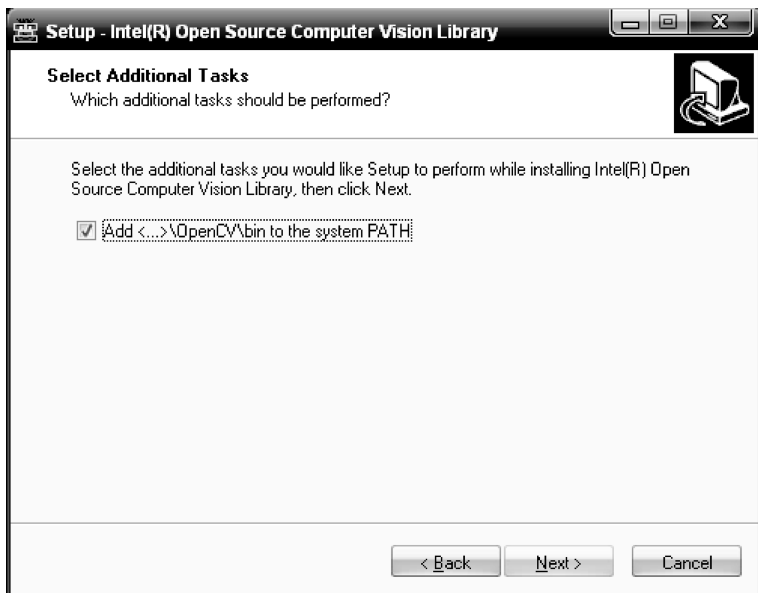


图 2-4 OpenCV 的安装界面(4)

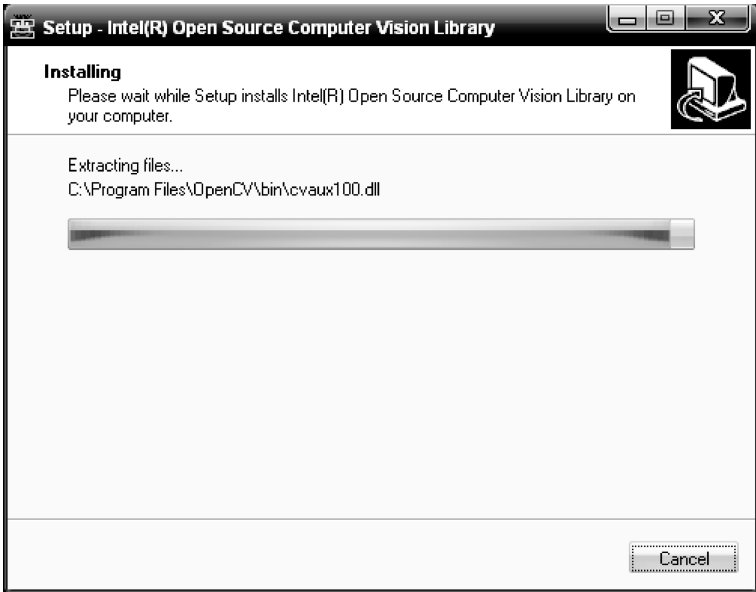


图 2-5 OpenCV 的安装界面(5)

2.1.2 配置 Windows 环境变量

检查 C:\Program Files\OpenCV\bin 是否已经被加入到环境变量 PATH 中,如果没有就添加该路径,如图 2-6、图 2-7 所示。

2.1.3 如何配置 VC++

1. 全局配置

在当前版本下,要先设置系统变量,以后就不需要在每次使用时手工复制 dll 或设置 dll 的路径。设置系统变量的方法如下:

我的电脑->属性->高级->环境变量->path->C:\Program Files\OpenCV\bin(根据当前计算机上 OpenCV 的安装路径进行设置,若设置多个路径,则用分号隔开)。

为了防止每次都从安装目录下复制相关函数的 cpp 文件到建立的工程下面,需要设置全局变量来默认函数的存放位置。设置全局变量的方法如下:

菜单 Tools->Options->Directories 先设置 lib 路径,选择 Library files,填入路径:

C:\Program Files\OpenCV\lib

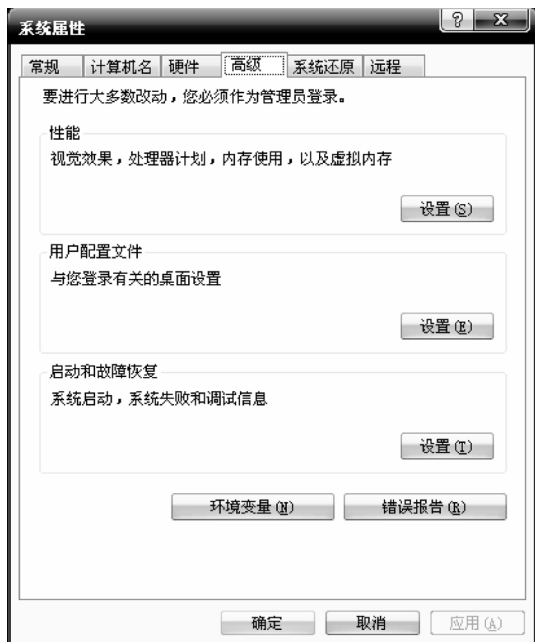


图 2-6 Windows 环境变量的配置(1)



图 2-7 Windows 环境变量的配置(2)