

计算机 视觉原理与应用

The principle of computer vision
and Applications

孟 璟 编著



東北大學出版社
Northeastern University Press

计算机视觉原理与应用

孟 琢 编著

东北大学出版社

· 沈 阳 ·

©孟 璟 2012

图书在版编目 (CIP) 数据

计算机视觉原理与应用/孟璟编著. —沈阳: 东北大学出版社,
2012. 11

ISBN 978-7-5517-0237-9

I. ①计… II. ①孟… III. ①计算机视觉
IV. ①TP302. 7

中国版本图书馆 CIP 数据核字 (2012) 第 256069 号

出 版 者: 东北大学出版社

地址: 沈阳市和平区文化路 3 号巷 11 号

邮编: 110004

电话: 024—83680267 (社务室) 83687331 (市场部)

传真: 024—83680265 (办公室) 83680178 (出版部)

网址: <http://www.neupress.com>

E-mail: neuph@neupress.com

印 刷 者: 沈阳市池陆广告印刷有限公司

发 行 者: 东北大学出版社

幅面尺寸: 170mm × 235mm

印 张: 9.75

字 数: 197 千字

出版时间: 2012 年 11 月第 1 版

印刷时间: 2012 年 11 月第 1 次印刷

策划编辑: 石玉玲

责任编辑: 王延霞

责任校对: 北 辰

封面设计: 刘江旻

责任出版: 唐敏志

ISBN 978-7-5517-0237-9

定 价: 25.00 元

前言

随着计算机技术与数字图像技术的不断发展，并在现实需求的推动之下，计算机视觉逐渐成为一个新兴的交叉学科，并受到了国内外诸多学者、厂商越来越热切的关注。

本书力求理论与实践相结合。一方面，对于本书涉及的各种理论、模型等，都尽量给出了必要的数学推导；另一方面，为了加深读者的理解，本书对大部分涉及的算法给出了 Matlab 相关的实现代码，其中，较简短的直接在书中给出，相对复杂的则通过网盘表示。

本书第1章对计算机视觉、数字图像等相关概念进行定义，并综述了其历史发展；第2章对计算机视觉的基本方法进行介绍，由浅入深，力争使读者对其相关的最基本理论、模型、算法有所了解；第3至6章，重点介绍了当前计算机视觉领域较热门和常见的研究领域，在对国内外学者提出的知名方法进行介绍的同时，加入了作者的若干理解。

为便于读者理解书中计算机视觉相关原理，作者将各章节参考源码上传到网盘，感兴趣的读者可以自行从网络下载，网址为 www.diskes.com，提取码为 MjAxMjA5MTIxMDE3MjUzNDIyMTc3。

本书得到了国家自然科学基金项目的资助（No. 61101057）。本书由东北大学出版社进行审核、校对，在此，对为本书出版付出辛苦劳动的各位编辑，致以诚挚的谢意。

由于作者水平和学识有限，加之编写时间仓促，书中错误和不妥之处在所难免，望读者批评指正。

编 者
2012 年 6 月

目 录

第 1 章 绪 论	1
1.1 计算机视觉	1
1.2 数字图像	4
1.3 国内外研究现状	6
第 2 章 数字图像处理的基本方法	9
2.1 数字图像在计算机中的表达	9
2.1.1 位图的类型	9
2.1.2 对应的数据结构	12
2.2 傅里叶变换	16
2.3 滤波与算子	17
2.3.1 中值、均值滤波	17
2.3.2 低通、高通滤波	19
2.3.3 算子	23
2.4 浅析图像的多尺度	31
2.4.1 图像金字塔	32
2.4.2 高斯尺度空间	33
2.5 小波变换	37
2.5.1 连续小波变换	37
2.5.2 连续小波变换的离散化	38
第 3 章 图像复原	40
3.1 图像去噪	40
3.1.1 小波去噪	40
3.1.2 偏微分方程去噪	44
3.2 图像去模糊	51
3.3 图像去雾	54
3.3.1 雾天图像降质模型	54
3.3.2 暗通道去雾	55

第 4 章 图像分割	59
4.1 数据驱动的分割方法	59
4.1.1 阈值分割	59
4.1.2 区域增长	62
4.1.3 分水岭	64
4.2 模型驱动的分割方法	80
4.2.1 Snake 模型	80
4.2.2 Level-Set 模型	85
4.3 显著性区域检测	92
4.3.1 总体模型框架	93
4.3.2 视觉特征提取	94
4.3.3 特征图计算	95
4.3.4 显著图合并	96
4.3.5 注意焦点转移	97
第 5 章 特征提取	99
5.1 一般特征	99
5.1.1 颜色特征	99
5.1.2 纹理特征	101
5.1.3 形状特征	106
5.2 角点特征 (Harris)	112
5.3 具备不变性的特征 (SIFT)	115
第 6 章 目标识别与跟踪	120
6.1 目标识别	120
6.1.1 帧间差分法	120
6.1.2 混合高斯背景模型	122
6.1.3 光流法	128
6.2 目标跟踪	131
6.2.1 Mean Shift	132
6.2.2 Kalman 滤波器	139
参考文献	145

第1章 绪论

1.1 计算机视觉

计算机视觉是一门研究如何使计算机学会像人一样“观察”的科学，更进一步地说，就是指用摄影机和电脑代替人眼对目标进行识别、跟踪和测量等机器视觉，并进一步作图像处理，用电脑处理成为更适合人眼观察或传送给仪器检测的图像。

作为一个科学学科，计算机视觉研究相关的理论和技术，试图建立能够从图像或者多维数据中获取“信息”的人工智能系统。因为感知可以看作从感官信号中提取信息，所以计算机视觉也可以看作研究如何使人工系统从图像或多维数据中“感知”的科学。

作为一个工程学科，计算机视觉寻求基于相关理论与模型来建立分析系统。这类系统的组成部分包括：

- ① 程序控制（例如工业机器人和无人驾驶汽车）；
- ② 事件监测（例如图像监测）；
- ③ 信息组织（例如图像数据库和图像序列的索引建立）；
- ④ 物体与环境建模（例如工业检查、医学图像分析和拓扑建模）；
- ⑤ 交互互动（例如人机互动的输入设备）。

计算机视觉同样可以被看作生物视觉的一个补充。在生物视觉领域中，人类和各种动物的视觉都得到了研究，从而建立了这些视觉系统感知过程中所使用的物理模型。另外，在计算机视觉中，靠软件和硬件实现的人工智能系统得到了研究与描述。生物视觉与计算机视觉进行的学科间交流为彼此都带来了巨大价值。

计算机视觉包含如下一些分支：图像去噪（Image Denoise）、图像去模糊（Image Deblur）、图像分割（Image Segmentation）、显著性检测（Saliency Detection）、特征提取（Feature Extraction）、配准（Registration）、目标跟踪（Object Tracking）、目标识别（Object Recognition）、画面拼接（Image Mosaic）、三维重建（3D Rebuild）、机器学习（Machine Learning），等等。部分分支如图 1.1 ~ 图

1.7 所示。



图 1.1 图像去噪

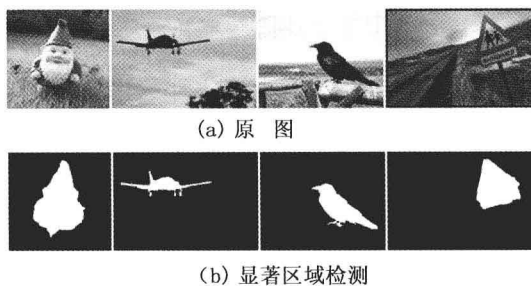


图 1.2 显著性检测

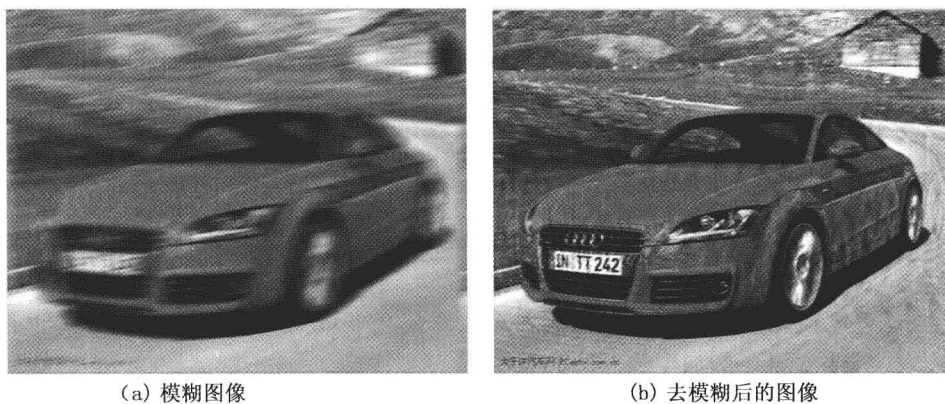


图 1.3 图像去模糊



图 1.4 目标跟踪

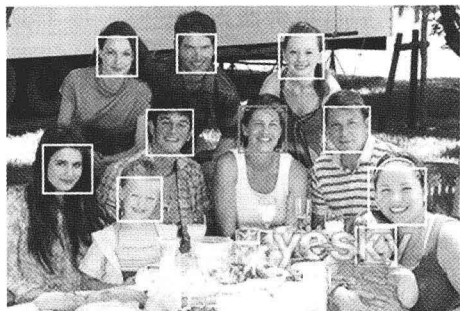


图 1.5 人脸识别

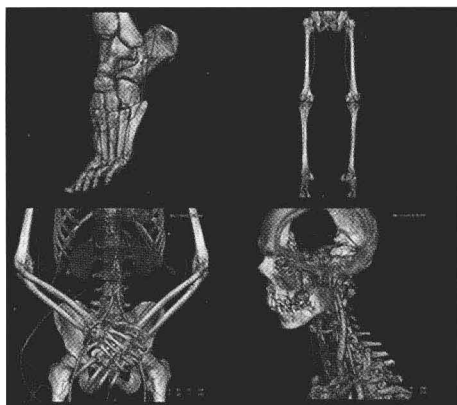
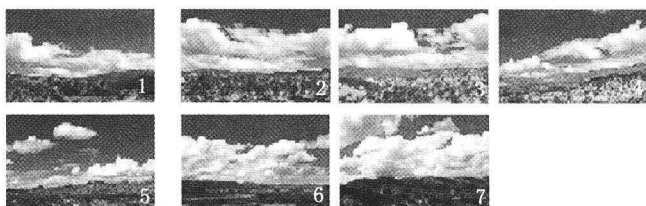


图 1.6 医学图像三维重建



(a) 待拼接的七幅图像



(b) 拼接后的图像

图 1.7 图像拼接

计算机视觉、图像处理、图像分析、机器人视觉和机器视觉是彼此紧密关联的学科。如果翻开带有上面这些名字的教材，会发现在技术和应用领域中它们都有着大部分的重叠。这表明这些学科的基础理论大致是相同的，甚至让人怀疑它们是同一学科被冠以不同的名称。

然而，各研究机构、学术期刊、会议及公司往往把自己特别地归为其中某一个领域，于是各种各样的用来区分这些学科的特征便被提了出来。下面将给出一种区分方法。

计算机视觉的研究对象主要是在映射到单幅或多幅图像上的三维场景，例如三维场景的重建。计算机视觉的研究在很大程度上是针对图像的内容。

图像处理与图像分析的研究对象主要是二维图像，实现图像的转化，尤其针对像素级的操作，例如提高图像对比度、边缘提取、去噪声和几何变换（如图像



旋转)。这一特征表明,无论是图像处理还是图像分析,其研究内容都与图像的具体内容无关。

机器视觉主要是指工业领域的视觉研究,例如自主机器人的视觉,用于检测和测量的视觉。这表明在这一领域通过软件硬件,图像感知与控制理论往往通过与图像处理的紧密结合来实现高效的机器人控制或各种实时操作。

1.2 数字图像

数字图像最早起源于20世纪20年代,其最早应用于报纸印刷业。当时,第一次通过海底电缆,运用Bartlane电缆图片传输系统,将图像从伦敦传往纽约,把横跨大西洋传送一幅图片所需的时间从一个多星期减少到3个小时。为了用电缆传输图片,首先要进行编码,然后在接收端用特殊的打印设备重构该图片。图1.8所示就是用这种方法传送并利用电报打印机通过字符模拟中间色调还原出来的图像。

这些早期数字图像视觉质量的改进工作,涉及打印过程的选择和亮度等级的分布等问题。用于得到图1.8的打印方法到1921年底就被彻底淘汰了,转而支持一种基于光学还原的技术,该技术在电报接收端用穿孔纸带打出图片。图1.9就是用这种方法得到的图像,对比图1.8,它在色调质量和分辨率方面的改进都很明显。

早期的Bartlane系统可以用5个灰度等级对图像编码,到1929年已增加到15个等级。图1.10所示的这种典型类型的图像就是用15级色调设备得到的。在这一时期,由于引入了一种用编码图像纸带去调制光束而使底片感光的系统,明显地改善了复原过程。

至此的数字图像,还维持在“编码—传输—还原”层次上的数字化。20世纪40年代,约翰·冯·诺依曼提出了计算机的两个重要概念:①保存程序和数据的存储器;②条件分支。这两个概念是中央处理单元(CPU)的基础,从而引发了一系列重要技术进步,使得计算机技术进入了一个飞速发展的阶段。到了20世纪60—70年代,随着大规模集成电路的发展,在计算机上进行数字图像的存储、显示、处理等操作成为了可能。在经过了抽样、量化、编码等步骤,将图像中的灰度、颜色进行数字化处理,从而可以将图像转换成计算机能够存储、显示的形式,这也就是现在通常所指的“数字图像”。

最早利用计算机技术进行数字图像处理的工作,始于1964年美国加利福尼亚的喷气推进实验室。当时由“旅行者7号”卫星传送的月球图像由一台计算机进行了处理,以校正航天器上电视摄像机中各种类型的图像畸变。图1.11显示了由“旅行者7号”于1964年7月31日上午9点9分(美国东部时间)在光线

影响月球表面约 17 分钟时摄取的第一张月球图像。



图 1.8 1921 年的数字图像，运用 Bart-lane 传输系统

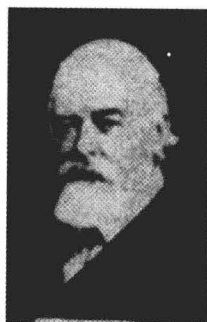


图 1.9 1922 年的数字图像，基于光学还原技术



图 1.10 1929 年的数字图像，15 个灰度等级

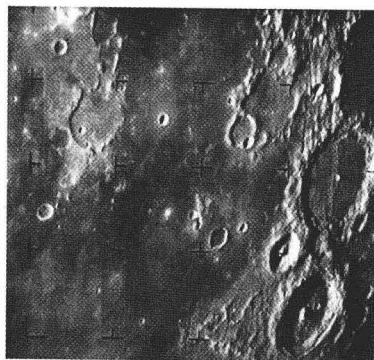


图 1.11 1964 年的数字图像，利用了计算机技术

数字图像处理经过多年的发展，已成为一门新兴的交叉性学科。1972 年，英国 EMI 公司工程师 Housfield 发明了用于头颅诊断的 X 射线计算机断层摄影装置，也就是通常所说的 CT (Computer Tomograph)。CT 的基本方法是根据人的头部截面的投影，经计算机处理来重建截面图像，称为图像重建。1975 年，EMI 公司又成功研制出全身用的 CT 装置，获得了人体各个部位鲜明清晰的断层图像。1979 年，这项无损伤诊断技术被授予诺贝尔奖，以表彰它对人类作出的划时代贡献。

从 20 世纪 70 年代中期开始，随着计算机技术、人工智能和思维科学等研究领域的迅猛进步及相互交融，数字图像处理技术也向着更高、更深的层次发展。人们已经开始研究如何用计算机系统解释图像，类似人类视觉系统理解外部世界，这被称为图像理解或计算机视觉。此时，数字图像处理的基本技术已经比较完善，人们的研究更加着重于智能化。很多国家，特别是发达国家投入较多的人力、物力进行这项研究，取得了不少重要的研究成果。其中代表性的成果是 70



年代末 MIT 的 Marr 提出的视觉计算理论, 这个理论成为计算机视觉领域其后十多年的主导思想。而这一时期, 数字图像处理应用的深度和广度也在飞快地加深和扩大。

20 世纪 80 年代末期, 人们开始将其应用于地理信息系统, 研究海图的自动读入、自动生成方法。而此后数字图像处理技术的应用领域不断拓展, 多种学科都可以看得到它的身影。

但是, 数字图像处理技术的大发展是从 20 世纪 90 年代初才开始的。这是由于, 自 1986 年以来, 小波理论与变换方法迅速发展, 它克服了傅里叶分析不能用于局部分析等方面的不足之处, 被认为是调和半个世纪以来工作的结晶。Mallat 在 1988 年有效地将小波分析应用于图像分解和重构。小波分析被认为是信号与图像分析在数学方法上的重大突破。随后数字图像处理技术迅猛发展, 到目前为止, 图像处理在图像通讯、办公自动化系统、地理信息系统、医疗设备、卫星照片传输及分析和工业自动化领域的应用越来越多。

计算机视觉是一门研究如何使计算机学会像人一样“观察”的科学, 而数字图像又是计算机“看”到的内容在计算机内的表达形式。因此可以说, 数字图像是计算机视觉的主要分析对象, 数字图像处理是计算机视觉的基础。

1.3 国内外研究现状

早期计算机视觉的研究主要集中在二维图像分析上, 如光学字符识别、工件表面质量检测、显微图和航空图片分析等。早期计算机视觉与图像处理、模式识别并没有明确的划分, 泛指一切使用计算机处理图像信息的技术。随着研究的深入, 图像处理侧重二维图像的基础处理; 模式识别偏重于图像特征的描述、辨认、分类和解释; 而计算机视觉则注重于图像中三维场景信息的提取、描述与分析。虽然这三类学科互有差别, 但是它们的研究领域也有一定程度上的交叉重叠^[1]。

计算机视觉系统的典型结构一般由视觉传感器、图像采集设备、计算机和图像处理软件组成, 如图 1.12 所示。系统典型的工作流程为: 由视觉传感器获取图像, 经图像采集设备传送到计算机, 最后通过图像处理软件对图像进行一系列分析、计算, 得到任务所需求的具体信息。

随着计算机视觉理论研究的深入, 国内外学者相继提出很多视觉理论, 比较著名的有通用视觉理论^[2], 目的视觉理论^[3], 主动视觉理论^[4-5], 活跃视觉理论^[6], 定性视觉理论^[7], 等等。文献 [8] 总结的以下两种计算机视觉理论框架具有重要意义。

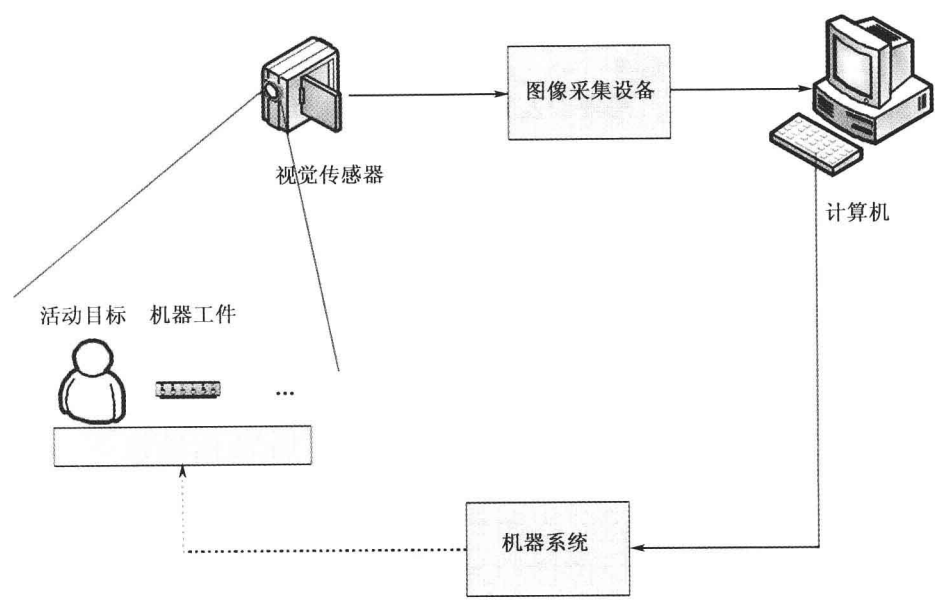


图 1.12 计算机视觉系统结构

(1) Marr 的计算机视觉理论框架

在神经生理学、心理物理学和临床神经病理学的视觉研究基础上，Marr 提出了人类视觉的数学诠释和计算理论，并以此为根据提出了 Marr 计算视觉理论^[9]。该理论把视觉过程分为三个层次：计算理论、算法与数据结构以及硬件实现。这一理论着重强调计算理论层次，并认为视觉的目的是从图像中建立物体形状和位置的描述，即三维重建。在计算理论层次方面，Marr 将整个视觉过程分为早期视觉、中期视觉和后期视觉三个阶段，如图 1.13 所示。图中早期视觉处理阶段主要对输入的原始图像进行处理，使用图像滤波、图像增强、边缘检测等技术提取物体表面的物理特性，如纹理、角点、边缘、线条以及色彩等基本特征；中期视觉处理阶段的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、运动估计、纹理提取等视觉技术；后期视觉处理阶段的主要任务是在前期处理的基础上，建立物体的三维描述，识别三维物体，是视觉信息处理的最后阶段。

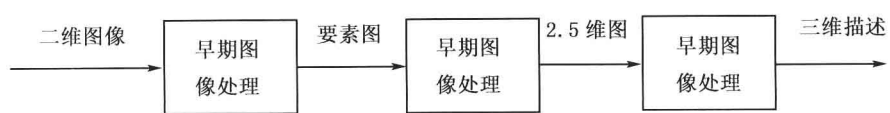


图 1.13 视觉理论框架

Marr 视觉计算理论首次为计算机视觉研究建立了理论框架，使该领域的研究有了比较明确的体系，极大地推动了计算机视觉研究的发展，对神经科学的发展



和人工智能的研究也产生了深远的影响。

(2) 主动视觉理论框架

主动视觉理论框架^[5]是根据人类视觉具有“主动”的特点提出来的。人类视觉的“主动”体现在两点。① 视角的多变。人通常会通过改变视线方向、距离、观察范围来全方位地观察物体，这在人类认知周围世界的过程中起到至关重要的作用。② 选择性。人类根据需要有选择地对物体某一部分加以特别的注意，不需要的部分即使处于视线范围内也会视而不见。主动视觉框架如图 1.14 所示。

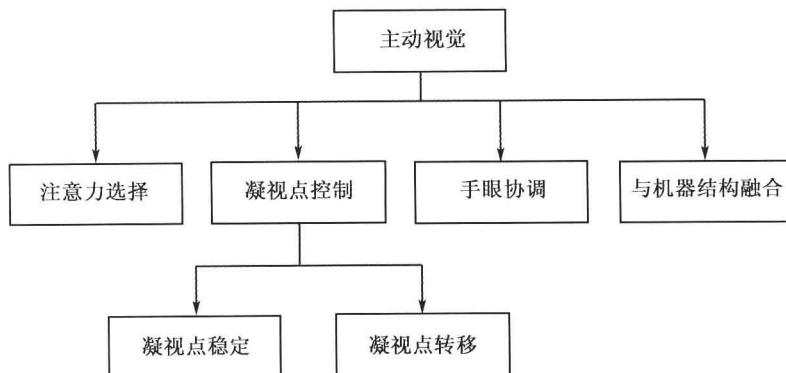


图 1.14 主动视觉理论框架

在主动视觉框架中，视觉系统所支持的动作性能与其视觉行为充分结合，根据已有的分析结果和视觉的当前要求，通过主动控制摄像机参数的机制来改变摄像机的位置、焦距和光圈，模拟人类视觉观察的过程，协调摄像机与周围世界的关系。另外，结合先验知识与设定的目的信息，主动视觉还可以通过改变摄像机参数或选择不同的图像特征，实现对研究对象有选择的感知。

第2章 数字图像处理的基本方法

2.1 数字图像在计算机中的表达

随着计算机技术的不断进步，数字图像技术也在不断发展。进入20世纪90年代，以Windows为代表的视窗操作系统的出现，大大增强了计算机的功能，操作便利性、用户友好性以及计算机显示功能都产生了质的飞跃。在这种趋势下，在Windows操作系统下，对数字图像的表达、存储、传输也发生了改变，多种图像格式应运而生，其中具有代表性的包括：BMP格式、JPG格式、TIF格式、PNG格式、GIF格式，等等。图像格式众多，然而万变不离其宗，本节以BMP图像格式为例，详细解读数字图像在计算机中的表达方式。

BMP是一种与硬件设备无关的图像文件格式，使用非常广。它采用位映射存储格式，除了图像深度可选以外，不采用其他任何压缩，因此，BMP文件所占用的空间很大。BMP文件的图像深度可选1 bit，4 bit，8 bit及24 bit。BMP文件存储数据时，图像的扫描方式是按从左到右、从下到上的顺序。

典型的BMP图像文件由四部分组成^[10-11]：

- ① 位图文件头数据结构，包含BMP图像文件的类型、显示内容等信息；
- ② 位图信息数据结构，包含BMP图像的宽、高、压缩方法，以及定义颜色等信息；
- ③ 调色板，这个部分是可选的，有些位图需要调色板，有些位图，比如真彩色图（24位的BMP），就不需要调色板；
- ④ 位图数据，这部分的内容根据BMP位图使用位数的不同而不同，在24位图中直接使用RGB，而其他小于24位的使用调色板中颜色索引值。

2.1.1 位图的类型

位图有两种类型，即设备相关位图（Device Dependent Bitmap，DDB）和设备无关位图（Device Independent Bitmap，DIB）。DDB位图在早期的Windows系统（Windows 3.0以前）中是很普遍的，事实上它也是唯一的。然而，随着显示



器制造技术的进步,以及显示设备的多样化,DDB 位图的一些固有的问题开始浮现出来了。比如,它不能够存储(或者说获取)创建这张图片的原始设备的分辨率,这样,应用程序就不能快速地判断客户机的显示设备是否适合显示这张图片。为了解决这一难题,微软创建了 DIB 位图格式。

(1) DIB 位图

DIB 位图包含下列颜色和尺寸信息。

- ① 原始设备(即创建图片的设备)的颜色格式。
- ② 原始设备的分辨率。
- ③ 原始设备的调色板。
- ④ 一个位数组,由红、绿、蓝(RGB)三个值代表一个像素。
- ⑤ 一个数组压缩标志,用于表明数据的压缩方案(如果需要的话)。

以上这些信息保存在 BITMAPINFO 结构中,该结构由 BITMAPINFOHEADER 结构和两个或更多个 RGBQUAD 结构所组成。BITMAPINFOHEADER 结构所包含的成员表明图像的尺寸、原始设备的颜色格式以及数据压缩方案等信息。RGBQUAD 结构标志了像素所用到的颜色数据。

DIB 位图也有两种形式,即底到上型 DIB 和顶到下型 DIB。底到上型 DIB 的原点在图像的左下角,而顶到下型 DIB 的原点在图像的左上角。如果 DIB 的高度值(由 BITMAPINFOHEADER 结构中的 biHeight 成员标志)是一个正值,那么就表明这个 DIB 是一个底到上型 DIB,如果高度值是一个负值,那么它就是一个顶到下型 DIB。注意:顶到下型的 DIB 位图是不能被压缩的。

位图的颜色格式是通过颜色面板值(planes)和颜色位值(bitcount)计算得来的,颜色面板值永远是 1,而颜色位值则可以是 1, 4, 8, 16, 24, 32 其中的一个。如果它是 1,则表示位图是一张单色位图(通常是黑白位图,只有黑和白两种颜色,当然它也可以是任意两种指定的颜色);如果它是 4,则表示这是一张 VGA 位图;如果它是 8, 16, 24 或是 32,则表示该位图是其他设备所产生的。如果应用程序想获取当前显示设备(或打印机)的颜色位值(或称位深度),可调用 API 函数 GetDeviceCaps(),并将第二个参数设为 BITSPIXEL 即可。

显示设备的分辨率是以每米多少个像素来表明的,应用程序可以通过以下三个步骤来获取显示设备或打印机的水平分辨率。

第一,调用 GetDeviceCaps()函数,指定第二个参数为 HORZRES。

第二,再次调用 GetDeviceCaps()函数,指定第二个参数为 HORZSIZE。

第三,用第一个返回值除以第二个返回值,即 $\text{DetDeviceCaps}(hDC, \text{HORZRES}) / \text{GetDeviceCaps}(hDC, \text{HORZSIZE})$ 。

应用程序也可以使用相同的三个步骤来获取设备的垂直分辨率,不同之处只是要将 HORZRES 替换为 VERTRES,把 HORZSIZE 替换为 VERTSIZE 即可。调色

板被保存在一个 RGBQUAD 结构的数组中, 该结构指出了每一种颜色的红、绿、蓝的分量值。位数组中的每一个索引都对应于一个调色板项 (即一个 RGBQUAD 结构), 应用程序将根据这种对应关系, 将像素索引值转换为像素 RGB 值 (真实的像素颜色)。应用程序也可以通过调用 `GetDeviceCaps()` 函数来获取当前显示设备的调色板尺寸 (将该函数的第二个参数设为 `NUMCOLORS` 即可)。

Win32 API 支持位数据的压缩 (只对 8 位和 4 位的底到上型 DIB 位图)。压缩方法是采用运行长度编码方案 (RLE), RLE 使用两个字节来描述一个句法, 第一个字节表示重复像素的个数, 第二个字节表示重复像素的索引值。有关压缩位图的详细信息请参见对 `BITMAPINFOHEADER` 结构的解释。

应用程序可以从一个 DDB 位图创建出一个 DIB 位图, 步骤是, 先初始化一些必要的结构, 然后调用 `GetDIBits()` 函数。不过, 有些显示设备有可能不支持这个函数, 则可以通过调用 `GetDeviceCaps()` 函数来确定 (`GetDeviceCaps()` 函数在调用时指定 `RC_ DI_ BITMAP` 作为 `RASTERCAPS` 的标志)。

应用程序可以用 DIB 去设置显示设备上的像素 (也就是显示 DIB), 方法是调用 `SetDIBitsToDevice()` 函数或调用 `StretchDIBits()` 函数。同样, 有些显示设备也有可能不支持以上这两个函数, 这时可以指定 `RC_ DIBTODEV` 作为 `RASTERCAPS` 的标志, 然后调用 `GetDeviceCaps()` 函数来判断该设备是否支持 `SetDIBitsToDevice()` 函数。也可以指定 `RC_ STRETCHDIB` 作为 `RASTERCAPS` 标志来调用 `GetDeviceCaps()` 函数, 来判断该设备是否支持 `StretchDIBits()` 函数。

如果应用程序只是要简单地显示一个已经存在的 DIB 位图, 那么只要调用 `SetDIBitsToDevice()` 函数就可以。比如一个电子表格软件, 可以打开一个图表文件, 在窗口中简单地调用 `SetDIBitsToDevice()` 函数, 将图形显示在窗口中。但如果应用程序要重复地绘制位图的话, 则应该使用 `BitBlt()` 函数, 因为 `BitBlt()` 函数的执行速度要比 `SetDIBitsToDevice()` 函数快很多。

(2) DDB 位图

DDB 位图之所以现在还被系统支持, 只是为了兼容旧的 Windows 3.0 软件, 如果程序员现在要开发一个与位图有关的程序, 则应该尽量使用或生成 DIB 格式的位图。

DDB 位图是被一个单个结构 `BITMAP` 所描述, 这个结构的成员标明了该位图的宽度、高度、设备的颜色格式等信息。

DDB 位图也有两种类型, 即可废弃的 DDB 和不可废弃的 DDB。可废弃的 DDB 位图就是一种当系统内存缺乏, 并且该位图也没有被选入设备描述表 (DC) 的时候, 系统就会把该 DDB 位图从内存中清除 (即废弃)。不可废弃的 DDB 则是无论系统内存多少都不会被系统清除的 DDB。API 函数 `CreateDiscardableBitmap()` 函数可用于创建可废弃位图。而函数 `CreateBitmap()`, `CreateCompatibleBit-`