



普通高等教育“十三五”计算机类规划教材

计算机图形学

◎ 郭晓新 徐长青 杨瀛涛 编著

Computer Graphics

第3版



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十三五”计算机类规划教材

计算机图形学

第3版

郭晓新 徐长青 杨瀛涛 编著



机械工业出版社

本书系统地介绍了计算机图形学的有关原理、算法及实现。全书共分八章,主要包括:基本图形的生成、图形几何变换、曲线和曲面造型、基本的图形运算、几何造型、消隐算法、真实感图形绘制等内容。在内容编排上通俗易懂、深入浅出,使学生能够充分掌握计算机图形学的基础理论和绘制技术。

本书既可作为高等院校计算机类和电子信息类相关专业本科生、研究生学习计算机图形学的教材,也可作为从事计算机图形学、计算机辅助设计、科学计算可视化、计算机动画等领域的广大科技人员的参考书。

图书在版编目(CIP)数据

计算机图形学/郭晓新,徐长青,杨瀛涛编著.—3版.—北京:机械工业出版社,2017.12

普通高等教育“十三五”计算机类规划教材

ISBN 978-7-111-58868-9

I. ①计… II. ①郭… ②徐… ③杨… III. ①计算机图形学-高等学校-教材 IV. ①TP391.41

中国版本图书馆CIP数据核字(2017)第330890号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:路乙达 责任编辑:路乙达 王 康

责任校对:郑 婕 封面设计:张 静

责任印制:张 博

三河市宏达印刷有限公司印刷

2018年2月第3版第1次印刷

184mm×260mm·15.5印张·376千字

标准书号:ISBN 978-7-111-58868-9

定价:38.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

服务咨询热线:010-88379833

读者购书热线:010-88379649

网络服务

机工官网:www.cmpbook.com

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

金书网:www.golden-book.com

封面无防伪标均为盗版

前 言

计算机图形学是研究通过计算机将数据转换为图形,并在显示设备上显示的原理、方法和技术的学科。计算机图形学研究内容非常广泛,涉及图形交互技术、光栅图形生成算法、图元绘制、曲线曲面造型、实体造型、真实感图形计算与显示、非真实感绘制,以及科学计算可视化、计算机动画、游戏程序设计、虚拟现实、人机交互技术、计算机视觉等领域,并为这些领域提供理论与技术支撑,在计算机类专业的知识结构中起着关键作用。

2004年和2010年,我们分别编撰出版了本书的第1版和第2版,经过这些年的课程教学实践,并同时参考国内外一些专著,我们对第2版又进行了必要的改动和补充,形成了目前的新版本,力图全面、准确地介绍计算机图形学的原理、算法及实现。本书在内容上,介绍了计算机图形学的核心概念、理论与方法;在组织结构上,采用了自底向上和自顶向下相结合和系统化的方法阐述计算机图形学的理论。本书系统地讲解了基本图形的扫描转换、二维变换和裁剪、三维变换和投影、自由曲线和曲面、分形几何、消隐和真实感图形绘制,使读者在较短的时间内了解计算机图形学,学习和掌握基础理论,培养工程实践的应用能力。

郭晓新、徐长青、杨瀛涛共同完成了本书的编撰工作。李群、孙超、李玮等研究生参与了本书的部分资料整理工作,在此对他们的辛苦付出表示衷心感谢。

受水平和能力所限,书中不足之处在所难免,衷心希望读者、专家不吝赐教,给予批评与指正。

编 者
于吉林大学

目 录

前 言	
第一章 计算机图形学简介	1
第一节 计算机图形学	1
第二节 计算机图形学的起源与发展	2
第三节 计算机图形学的研究内容及应用	3
第四节 图形系统的硬件	4
一、显示处理器	5
二、图形显示器	5
第五节 计算机图形编程接口	6
一、OpenGL	6
二、DirectX	7
三、Open Inventor	7
习题	8
第二章 图形基元的显示	9
第一节 直线扫描转换算法	9
一、DDA 直线扫描转换算法	9
二、中点画线法	11
三、Bresenham 画线算法	13
第二节 圆的扫描转换算法	15
一、中点画圆法	16
二、Bresenham 画圆算法	19
第三节 椭圆扫描转换算法	21
第四节 区域填充	25
一、种子填充算法	25
二、扫描线填充算法	28
三、多边形的扫描转换算法	31
四、边填充算法	35
五、图案填充	38
习题	39
第三章 图形变换	41
第一节 齐次坐标	42
第二节 二维图形变换	42
第三节 二维视见变换	47
第四节 三维图形变换	49
一、平移变换	49
二、比例变换	50
三、旋转变换	51
第五节 投影变换	54
一、平行投影	55
二、透视投影	57
第六节 裁剪	62
一、直线段裁剪算法	62
二、其他图形的裁剪	68
三、三维图形的裁剪	78
习题	82
第四章 曲线和曲面	85
第一节 曲线和曲面表示的基础知识	85
一、曲线和曲面参数表示	85
二、基本概念	86
第二节 Hermite 多项式	88
一、Lagrange 插值	89
二、三次 Hermite 插值	90
三、规范化三次 Hermite 插值	91
四、分段三次 Hermite 插值	92
第三节 Bézier 曲线	95
一、Bézier 曲线的定义	95
二、Bézier 曲线的性质	97
三、Bézier 曲线的拼接	98
四、Bézier 曲线的绘制	99

五、Bézier 曲线的升阶	106	四、八叉树	173
六、有理 Bézier 曲线	107	第三节 分形	177
第四节 Bézier 曲面	108	一、分形的概念	177
一、Bézier 曲面的定义	108	二、分形一般算法	180
二、Bézier 曲面的性质	109	三、Von Koch 算法	181
三、Bézier 曲面示例	109	四、Julia 和 Mandelbort 集	184
四、Bézier 曲面的拼接	112	第四节 粒子系统	186
第五节 B 样条曲线	113	习题	187
一、B 样条曲线的定义	114	第七章 消隐算法	190
二、B 样条曲线的性质	117	第一节 线面比较法消除隐藏线	190
三、均匀 B 样条曲线	118	一、凸多面体的可见性	190
四、准均匀 B 样条曲线	123	二、非凸多面和多个形体的 可见性	191
五、B 样条曲线的绘制	127	第二节 浮动水平线算法	194
六、非均匀有理 B 样条曲线	130	第三节 深度排序算法	197
第六节 B 样条曲面	134	第四节 z-缓冲算法	199
习题	137	第五节 扫描线算法	200
第五章 图形运算	140	一、扫描线算法的数据结构	200
第一节 线段的交点计算	140	二、扫描线算法的实施步骤	202
一、两条线段求交	140	第六节 区域分割算法	204
二、多条线段求交	141	一、多边形和区域的关系	204
第二节 多边形表面的交线计算	144	二、区域分割方法	206
第三节 平面中的凸壳算法	147	第七节 BSP 树算法	207
第四节 包含与重叠	149	一、构造 BSP 树	207
一、简单多边形的包含算法	149	二、BSP 树遍历	207
二、凸多边形的包含算法	150	第八节 光线投射算法	208
三、凸多边形重叠计算	152	习题	209
第五节 简单多边形的三角剖分	155	第八章 真实感图形的绘制	211
习题	160	第一节 漫反射及具体光源的 照明	211
第六章 形体的表示及其数据结构	162	一、环境光	211
第一节 二维形体的表示	162	二、漫反射	212
一、二维图形的边界表示	162	三、镜面反射与 Phong 模型	212
二、平面图形的四叉树表示 方法	165	四、光的衰减	213
第二节 三维几何模型	168	第二节 多边形网的明暗处理	214
一、几何元素	168	一、常数明暗法	215
二、线框、表面及实体表示	169	二、Gouraud 方法	215
三、三维形体表示方法	171		

三、Phong 方法	216	三、光线跟踪算法	226
第三节 阴影	218	四、加速光线跟踪算法	228
一、影域多边形方法	218	第七节 颜色模型	230
二、曲面细节多边形方法	219	一、色度	230
第四节 透明	219	二、三色学说	231
一、非折射透明	219	三、原色混合系统与颜色匹配实验	232
二、考虑折射的透明	220	四、CIE 色度图	232
第五节 纹理	221	五、几种颜色系统	235
一、颜色纹理	221	习题	240
二、几何纹理	223	参考文献	241
第六节 整体光照模型	225		
一、整体光照模型的概念	225		
二、Whitted 整体光照模型	226		

第一章 计算机图形学简介

客观世界中的事物是多姿多彩的，而呈现在人们眼前的往往是它们的外观，通过事物的外观可以进一步地认识及研究它们。以图画为表现形式的图形信息在人类的社会生活中起着非常重要的作用。与其他信息表现形式相比，图形信息具有容易理解、容易记忆、直观等特点。随着现代科学技术的发展，用计算机来处理图形信息，完成图形的构造、显示与分析很自然地成为人们研究与探索的领域。

第一节 计算机图形学

伴随着计算机技术的快速发展，涉及图形方面的应用也越来越深入，比如零件的构造与显示、卫星照片的处理及手写文字的识别等。经过多年的研究与发展，逐渐地形成了多个与图形相关的分支，计算机图形学（Computer Graphics）、图像处理（Image Processing）和模式识别（Pattern Recognition）就是其中的典型代表。

简单来说，计算机图形学是指用计算机产生对象图形的输出的技术。更确切地说，计算机图形学是研究通过计算机将数据转换为图形，并在专门显示设备上显示的原理、方法和技术的学科。它综合了应用数学、计算机科学等多方面的知识。

图形是对象的一种外在表现形式，它是对象有关信息的具体体现。所谓对象，可以是各种具体的、实在的物体，如家具、机械零件、房屋建筑等，也可以是抽象的、假想的事物，如天气形势、人口分布、经济增长趋势等。能够正确地表达出一个对象性质、结构和行为的描述信息，称为这个对象的模型。计算机图形学中产生图形的方法是建立对象的模型，即对该对象做出正确的信息描述，然后利用计算机对该模型进行各种必要的处理，从无到有地产生能正确反映对象某种性质的图形输出。可以说，计算机产生图形的过程就是将数据（对象的模型表示）转化为图形的过程。

图像处理是指用计算机来改善图像质量的数字技术。可见或不可见的图像经过量化后输入到计算机中（扫描仪扫描输入、数码相机拍照），由计算机按应用的需要对已有的图像进行增强、复原、分割、重建、编码、存储、传输等种种不同的处理，再把加工后的图像进行输出。在太空探索中分析宇宙飞船发回的各种照片，在生物医学工程中发展起来的计算机 X 射线断层摄影技术（Computer Tomography, CT）是计算机图像处理技术的典型例子。

模式识别是指用计算机对输入图形进行识别的技术。图形信息输入计算机后，先进行特征抽取等预处理，然后用统计判定方法或语法分析方法对图形做出识别，最后按照使用的要求给出图形的分类或描述。各种中西文字符及工程图纸的自动阅读装置，是模式识别技术的应用实例。

与计算机图形处理有关的上述三门学科是独立发展起来的。当前由于光栅扫描显示器的广泛使用及解决复杂实际问题的需要，它们已经相互渗透，也使人们对这三门学科的相互关系和共同技术产生越来越大的兴趣。

图 1-1 是计算机图形学、图像处理和模式识别的相互关系。

此外,与计算机图形学关系密切的学科还有计算几何学。计算几何学是研究几何模型和数据处理的学科。几何模型指描述物体形状的数据集合。显然,寻找对复杂形体的描述方法和在计算机中存放适当的数据结构并不容易。通常认为,二维、三维物体及曲线、曲面的描述,以及几何问题算法的设计和分析,都是计算几何学研究的内容。

交互式计算机图形学是指用计算机交互式地产生图形的技术。交互式绘图允许操作人员以对话方式控制和操纵图形的生成。图形可以边生成,边显示,边修改,直到产生符合使用要求的图形为止。交互式绘图可以使人的逻辑思维能

力、分析能力和计算机准确快速的计算能力结合起来,从而发挥更大的威力,使人们运用起来更加方便。其中,交互设备是实现交互技术,完成交互任务的基础。一般来说,交互设备有定位、键盘、选择、取值和拾取。交互式技术是用户用交互设备把信息输入计算机的不同方式,交互任务是用户输入到计算机的一个单元信息,基本任务有四种:定位、字串、选择、取数,如用鼠标选择菜单项或定位坐标点作图等。开发软件系统的人员需要了解相应的概念与原理,以便选择适宜的交互设备实现计算机系统和用户的沟通,而用户只需了解交互设备的作用和种类即可。交互技术是完成交互任务的手段,它的实现有赖于交互设备及其支撑环境。

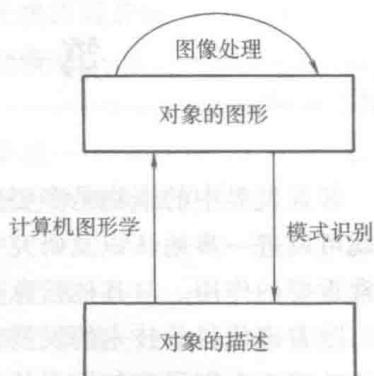


图 1-1 图像处理、模式识别和计算机图形学的相互关系

第二节 计算机图形学的起源与发展

1946年2月14日,世界上第一台电子计算机 ENIAC 在美国宾夕法尼亚大学问世。1950年,第一台图形显示器诞生于麻省理工学院,从此计算机具有了图像显示功能,也搭建了图形显示与计算机技术联系的桥梁。1959年,麻省理工学院林肯实验室第一次使用了具有指挥和控制功能的阴极射线管显示器,让单纯显示的“被动式”图形学开始迈向交互式计算机图形学。同一时期,许多商业公司也陆续在工业设计和工业生产中运用计算机图形技术,比如美国 Calcomp 公司的滚筒式绘图仪和美国 GerBer 公司的平板式绘图仪。这些学术研究和商业应用初步奠定了计算机图形学作为一个学科研究领域的基础。

“计算机图形学”这一术语最早出现在伊凡·苏泽兰 (Ivan Sutherland) 于 1962 年在麻省理工学院发表的博士生论文《Sketchpad: 一个人机交互通信的图形系统》中。伊凡·苏泽兰在这篇论文中阐述并展望了交互计算机图形学这一具有前瞻性的研究领域,并提出了一些至今仍被广泛使用的技术与基本概念,比如交互技术和分层存储符号的数据结构等。该博士论文的发表被视为是计算机图形学作为一个正式独立学科分支的开始。伊凡·苏泽兰也因为在交互式图形学方向的杰出贡献获得 1988 年的图灵奖。

在 20 世纪 60~70 年代,随着光栅显示器的诞生,对光栅图形学算法的研究迅速地发展壮大起来,大量基本概念以及相应的算法应运而生,计算机图形学进入了第一个兴盛时期。光栅图形学将图像转化为点阵表示,通过区域填充、裁剪、反走样等技术将图像在显示器上尽可能完美地显示出来。光栅图形的出现弥补了矢量图形数据结构复杂、难以进行位置搜索

和难以计算多边形形状和面积的缺点。同时,随着实用图形软件的发展,人们提出了大量图形软件的基本标准,比如 Core Graphics System (ACM SIGGRAPH, 1977)、Computer Graphics Metafile (ANSI, ISO/IEC, W3C, 1986) 等。这些标准为计算机图形学的推广、应用以及资源的共享发挥了巨大作用。

20 世纪 70 年代以来,真实感图形学和实体造型技术开始获得广泛的关注和研究,产生了大量开创性的工作。1970 年,美国计算机专家 Bouknight 提出了第一个光反射模型;1971 年,法国计算机专家 Gourand 提出“漫反射模型+插值”的思想;1975 年,美国计算机专家 Phong 提出 Phong 模型;这些光照模型都使得计算机图像显示更加接近人们日常生活中的真实场景。此外,在 20 世纪 70 年代初期,英国剑桥大学的 BUILD-1 系统,德国柏林工业大学的 COMPAC 系统,日本北海道大学的 TIPS-1 系统和美国罗切斯特大学的 PADL-1、PADL-2 系统等实体造型系统相继出现,这些系统都使用了多面体表示形体的方式,为计算机辅助设计等领域的发展做出了重要贡献。20 世纪 80 年代中期之后,随着计算机硬件的高速发展,特别是 20 世纪 90 年代出现的图形处理器 (GPU),计算机图形学开始具有强大的硬件计算基础。在此基础上发展起来的全局光照模型推动了真实感图形学的进一步发展,并大量运用于 CAD、科学计算可视化、动画、影视娱乐等各个领域。

计算机图形学是人们和计算机通信最通用和最有力的手段。计算机图形显示对不同年龄的用户都具有强大的吸引力,图形直观清晰的特性也大大推广了它的应用范围。大规模集成电路技术以及新型廉价硬件的诞生使计算机图形学得到飞速的发展。由于各种图形设备性能不断提高、价格不断下降,计算机图形学的应用领域正在逐渐增加。毋庸置疑,这一技术将继续普及和发展。

第三节 计算机图形学的研究内容及应用

计算机图形学在诸如工业、商业、政府部门、教育、科研、医学、娱乐和广告等领域,都有着广泛的应用。

在科学技术事业中,可以使用计算机来绘制表示数值计算或数据处理结果的图形。例如各种函数的图形、统计用的直方图、百分比图等。

在制图学方面,可以利用计算机来绘制精确的地形图、天气图、海洋图、石油开发图、人口密度图等。

计算机辅助设计 (Computer Aided Design, CAD) 和计算机辅助制造 (Computer Aided Manufacturing, CAM) 是计算机图形学的重要应用领域。CAD 和 CAM 技术已经相当广泛地应用到电子、机械、建筑、汽车、飞机、船舶等方面的设计和制造工作中。

在计算机仿真与动画方面,可以用计算机制作动画电影来表现真实物体或模拟物体的运动或变化,例如训练飞行员可以使用一种采用计算机控制的能够产生所需动态效果的飞行模拟器。

在过程控制中,可用以显示被控制对象有关环节在操作过程中的状态,使操作人员可以进行必要的调节和处理。

在办公室自动化方面,图形显示技术有助于数据及其相互关系的有效表达,以帮助人们

进行正确的决策。

计算机艺术的出现,使计算机的应用进入到艺术领域。利用计算机可以创造出具有一定水平和独特风格的艺术作品,例如传统的中国绘画、书法、油画、美术图案以及动画电影和广告等。

计算机图形学的一个新的应用是虚拟现实(Virtual Reality, VR),由计算机生成一个虚拟的环境,用户在其中可与三维场景中的对象进行交互。

图形用户界面(Graphical User Interface, GUI)为应用软件方便快捷的使用奠定了坚实的基础。其中窗口、图标和菜单是图形操作界面的典型元素。

随着计算机的推广和普及,计算机图形学的应用领域还将继续深入和扩大。

计算机图形学的研究内容是十分丰富的。虽然有些部分的研究工作已经进行了多年,取得了不少成果,但随着图形显示技术应用领域的扩大和深入,不断有新的研究课题涌现出来。从计算机图形学发展和应用的现状看,这门学科的主要研究内容可以概括为以下几个方面:

1) 图形的生成和表示技术。例如线段、圆弧、曲线和曲面的生成算法,区域填充算法,基本几何体的截交、相贯及展开算法以及投影、隐藏线和隐藏面消除,浓淡处理,灰度与色彩等各种表示技术。

2) 图形的操作与处理方法。例如图形的剪裁、开窗、平移、旋转、放大、缩小等各种操作的方法及软件或硬件的实现技术。

3) 图形输出设备与输出技术的研究。

4) 图形输入设备、交互技术及用户接口技术的研究。

5) 图形信息的数据结构及存储、检索方法。例如图形信息的各种机内表示方法,组织形式,存取技术,图形数据库的管理,图形信息的通信等。

6) 几何模型构造技术。刻画被处理对象几何性质的描述信息就构成它的几何模型。图形生成和操作的基础就是对象的几何模型,所以要研究几何模型的构造方法及性能分析等。

7) 动画技术。研究实现各种高速动画生成的各种软硬件方法、开发工具、动画语言等。

8) 图形软硬件的系列化、模块化和标准化研究。

9) 科学计算的可视化。科学计算可视化是指应用计算机图形生成技术将科学及工程计算的结果和测量数据等以图像的形式在计算机屏幕上显示出来,使人们能观察到用常规手段难以观察到的自然现象和规律,实现计算环境和工具的进一步现代化。

第四节 图形系统的硬件

如同其他任何一个计算机研究的分支,计算机图形学也有其必不可少的硬件基础。在计算机的发展史上,各个部分的硬件同图形学本身共同进步,相互促进。硬件水平的提高为更高级的图形学算法和技术提供了必需的物质准备,而图形学的进步也不断地为硬件的发展提出了要求和方向。

一般情况下,计算机图形学的硬件系统分为计算机、显示处理器、图形显示器、输入设备和硬拷贝设备五个部分。其中某些部分从图形学的专业角度来看,尤其值得关注。

一、显示处理器

显示处理器 (Graphic Processing Unit, GPU) 是专门用于图形显示过程中涉及运算的处理器, 是一种具有专门用途的 CPU。1999 年, NVIDIA 公司发布 GeForce 256 图形处理芯片时率先提出了 GPU 这一概念。由于在现代的计算机系统中, 图形的处理变得越来越重要, 除了计算机本身的 CPU 以外, 还需要一个专门处理图形计算的处理器, 这便是 GPU。GPU 的加入实现了将三维图像和特效处理功能集中在显示芯片内, 即所谓的“硬件加速”功能, 使图形的显示效率大大提高。

图形处理过程中涉及多种运算, 如多边形转换与光源处理 (Transform and Lighting, T&L) 运算。它是 3D 渲染中的一个重要部分, 用于计算多边形的 3D 位置并处理动态的光线效果。良好的 T&L 单元可以细致入微地展现 3D 物体和高级的光线特效。但是, 在过去的大多数 PC 中, T&L 运算同其他计算机内的运算一样交由 CPU 处理, 这也被称为软件 T&L。

作为整个计算机系统的中央处理单元, CPU 通常任务繁忙, 即使不考虑 T&L, 也要为内存管理、输入输出响应等基本工作提供运算服务。在这种情况下, 还要让 CPU 去承担运算负荷极大的图形运算, 势必会导致运算效率低下, 同时也影响了 CPU 原本的正常工作。所以显卡等待 CPU 数据运算完成的情况非常常见, 这样的运算速度是完全不能满足当下复杂的三维图形处理 (最具代表性的就是大型三维游戏) 要求的。

此时, GPU 便应运而生, 它能够在硬件水平基础上完成图形处理过程中的多种运算。GPU 独立地完成了这些专门的运算, 使得 CPU 从繁重的工作中解脱了出来。另外, 由于 GPU 的专用性, 在它的设计过程中可以充分考虑图形运算的特性, 而放弃其他运算所需要的元素, 使其图形运算范围内的工作效率更优于通用 CPU。

今天, GPU 通用计算技术的不断发展受到多方的广泛关注, GPU 的工作能力已不再仅仅局限于 3D 图形处理。各种各样的事实也证明在浮点运算、并行计算等方面, GPU 可以提供数十倍乃至上百倍于 CPU 的性能。在微软新发布的操作系统中, 一个显著的特性就是能够协同地发挥 GPU 和 CPU 各自的优势, 充分挖掘 GPU 的硬件价值, CPU 运算复杂的序列代码, 而 GPU 则处理大规模的并行计算。

二、图形显示器

图形显示器是将最终的显示效果呈现出来的部件。随着计算机图形学的蓬勃发展, 图形显示器也经历了从存储管式显示器、随机扫描显示器、光栅扫描显示器、彩色 CRT 光栅扫描显示器到平板显示器等发展阶段。而目前, 液晶显示器已经成为显示器领域的主流, 除了通用计算机外, 还广泛应用于手持式计算机、个人数字助理 (PDA) 等设备的显示屏幕。

阴极射线管 (Cathode Ray Tube, CRT) 曾是最为广泛使用的显示器。CRT 纯平显示器可视角度大、无坏点、色彩还原度高、响应时间短。在 CRT 作为主要显示器的时代, 光栅扫描方式应用较广。在光栅扫描工作方式的显示器中, 电子束从显示器后部的电子枪中发出, 逐行击打在扫描显示平面上的所有点上。当电子束从屏幕的左上角扫描到达右下角时, 即完成了整个屏幕的扫描。根据每一个点待显示的色彩不同, 电子束击打在各个位置时的能量也不同, 这样便形成了一幅具有丰富色彩的画面。每一次这样的扫描过程所产生的图像称为一帧, 然后电子束回到屏幕的左上角的起始位置, 开始下一个重复的扫描过程。

而作为图形显示领域的后起之秀, 现在的液晶显示器 (Liquid Crystal Display, LCD) 已取代了 CRT 显示器。19 世纪末, 奥地利的植物学家发现了液晶这种同时具备液体的流动性

和类似晶体排列特性的物质，并在今天大大地改变了人类将纯粹的数据生动地呈现在眼前的方法。LCD 在制造时把液晶灌入两个有细槽的平面之间，两个平面上的槽互相垂直，使得一个平面上的分子和另一个平面上的分子相互垂直地排列，而两个平面之间的分子却呈现出 90° 扭转。因为光线的传播方向顺着分子的排列方向，所以经过液晶的时候光线的传播也会 90° 扭转。但当给液晶加上一个电压时，分子便会重新垂直排列，使光线能直射出去，而不再扭转。用于通用计算机的 LCD，需要采用更加复杂的结构来实现彩色显示，所以还需要具备彩色显示处理能力的色彩过滤层。

与传统的 CRT 显示器相比，LCD 体积小、厚度薄、重量轻、能耗低、无辐射、无闪烁。在人们更加看重便捷性和健康的今天，LCD 的这些优势远远超越了前者。

在光栅扫描图形显示器和液晶显示器中，屏幕上可以点亮或熄灭的最小单位称为像素。像素的总数称为分辨率，通常情况下描述为每行的像素数与行数的乘积，例如，分辨率为 1024×768 。分辨率是表现图形显示器性能最重要的参数，显而易见，显示器的分辨率越大，就越能够细致地描绘显示画面，显示的质量就越好。

图形显示器性能的另一个重要参数是亮度等级。亮度等级（或称灰度等级）指的是单色显示器一个像素的亮度从完全黑色到完全白色之间可以有多少种不同的变化。彩色显示器稍微复杂一些，涉及两个参数，一个是显示器可以显示的颜色总数，另一个是在一帧画面上可以出现多少种颜色。

图形显示器中有一块存储空间，称为帧存储器，用来存储各个像素的值。可以将帧存储器视为一个二维矩阵，其中的值保存的是显示器上对应位置的颜色或亮度。在颜色数为 256 的情况下，帧存储器的每一单元至少占据 8 个 bit，则分辨率为 1024×768 的显示器中帧存储器的大小约为 1MB。彩色显示器的颜色系统有多种构造形式，其中 RGB（红、绿、蓝）形式最为常见。

第五节 计算机图形编程接口

近些年来，包括 DirectX 和 OpenGL 等计算机图形编程接口的广泛发展和诸如 Open Inventor 的图形库在业界的大规模使用，更是极大地促进了高质量图形界面的开发和推广。

一、OpenGL

OpenGL 被定义为“图形硬件的一种软件接口”。从本质上说，它是一个 3D 图形和模型库，具有高度的可移植性，并且具有非常快的速度。OpenGL 严格按照计算机图形学原理设计而成，符合光学和视觉原理，可以创建极其逼真的 3D 图形。

OpenGL 的前身是 SGI 的 IRIS GL。它最初是个 2D 图形函数库，后来逐渐演化为由这家公司的高端 IRIS 图形工作站所使用的 3D 编程 API。OpenGL 就是 SGI 对 IRIS GL 的移植性进行改进和提高的结果。这个新的图形 API 不仅具有 GL 的功能，而且是一个“开放”的标准。它的输入来自于其他图形硬件厂商，并且更容易应用到其他硬件平台和操作系统。从根本上说，OpenGL 就是为了 3D 几何图形处理而量身定做的。

作为 GL 与硬件无关的版本，OpenGL 在 20 世纪 90 年代早期就制定出来。这一图形软件包现在由代表许多图形公司和组织的 OpenGL 结构评议委员会（OpenGL Architecture Review Board）进行维护和更新。OpenGL 函数库专为高效处理三维应用而设计，但它也能

按 z 坐标为零的三维特例来处理二维场景描述。

图形函数定义为独立于任何程序设计语言的一组规范。语言绑定则是为特定的高级程序语言而定义的。它给出该语言访问各种图形函数的语法。每一个语言绑定以最佳地使用有关的能力及处理好数据类型、参数传递和出错等各种语法问题为目标来定义。图形软件包在特定语言中的实现描述由国际标准化组织来制定。OpenGL 的 C 和 C++ 语言绑定也一样如此。

在 OpenGL 中允许视景对象用图形方式表达,例如,由物体表面顶点坐标集合构成的几何模型。这些图形数据含有丰富的几何信息,得到的仿真图像能充分表达出其形体特征;而且在 OpenGL 中有针对三维坐标表示的顶点的几何变换,通过该变换可使顶点在三维空间内进行平移和旋转,对于由顶点的集合表达的物体则可以实现其在空间的各种运动。

OpenGL 通过光照处理能表达出物体的三维特征,其光照模型是整体光照模型,它把顶点到光源的距离、顶点到光源的方向向量以及顶点到视点方向向量等参数代入该模型,计算顶点颜色。因此,可视化仿真图形的颜色体现着物体与视点以及光源之间的空间位置关系,具有很强的三维效果。

为弥补图形方法难于生成复杂自然背景的不足,OpenGL 提供了对图像数据的使用方法,即直接对图像数据读、写和拷贝,或者把图像数据定义为纹理,与图形方法结合在一起生成视景图像以增强效果。为增强计算机系统三维图形的运算能力,有关厂家已经研制出了专门对 OpenGL 进行加速的三维图形加速卡,其效果可与图形工作站相媲美。

二、DirectX

同 OpenGL 一样,DirectX 中的 Direct3D 也是一套底层三维图形 API,是微软公司所制定的 API,与 Windows 操作系统兼容性好,可绕过图形显示接口 (GDI) 直接进行支持该 API 的各种硬件的底层操作,大大提高了图形渲染速度。

DirectX 是基于组件对象模型 (Component Object Model, COM) 的,它不是一个单一的实体,而是多个互相作用互相依赖的组件的集合。它包含的组件主要有: DirectGraphics (Direct3D 和 DirectDraw)、DirectInput、DirectPlay、DirectSound,为开发应用程序提供了一整套的多媒体接口方案。DirectX 开发之初是为了弥补 Windows 3.1 系统对图形、多媒体处理能力的不足,而今已发展成对整个多媒体系统的各个方面都有决定性影响的接口。Direct3D 为应用程序提供了 3D 图形处理的能力,其在 3D 图形方面的优秀表现,让 DirectX 的其他方面显得黯淡无光。DirectDraw 为应用程序提供了 2D 处理能力。目前 DirectDraw 和 Direct3D 合称为 DirectGraphics,为开发图形应用程序提供了强有力的底层支撑。DirectInput 为外围设备 (比如游戏手柄) 提供了相应的支持。DirectPlay 利用网络来为多人游戏提供支撑。DirectSound 把声音整合进了应用程序中。

DirectX 从 5.0 版本开始逐渐成熟起来,目前已发展到 13.0 版本,现在大部分常见的 3D 游戏都支持 DirectX。微软也正以两个月就更新一次 DirectX 版本的速度快速发展。从 9.0 版本开始,DirectX 对硬件新功能的支持已超过 OpenGL。从 DirectX 的发展史中可以看到,微软的 3D API 和硬件一同发展,新硬件带来新的 DirectX 特性,新的 DirectX 特性加速硬件的发展。

三、Open Inventor

Open Inventor (以下简称 OIV) 是 SGI 公司开发的基于 OpenGL 的面向对象三维图形软

件开发包,使用 OIV 开发包,程序员可以快速、方便地开发出各种类型的交互式三维图形软件,且可以任意修改。OIV 具有平台无关性,可以在 Microsoft Windows, Unix, Linux 等多种操作系统中使用。OIV 允许使用 C、C++、Java、DotNet 等多种编程语言进行程序开发。经过十多年的发展,Open Inventor 已经基本上成为面向对象的 3D 图形开发的工业标准,广泛地被应用在机械工程设计与仿真、医学和科学图像、石油钻探、虚拟现实、科学数据可视化等科技领域。OIV 是由很多系列的对象模块组成的,通过利用这些对象模块,工程人员有可能以最少的编程成本,开发出能充分利用其强大的图形硬件特性的程序。OIV 是一个建立在 OpenGL 基础上的对象库,开发人员可以任意使用、修改和扩展对象库。OIV 对象包括:形体、属性、组、数据库图元和引擎等对象;还有例如像轨迹球和手柄盒等操作器、材质编辑器、方向灯编辑器、examiner 观察器等组件。OIV 提供了一个完整且经济高效的面向对象系统。

OIV 是面向对象的,因为它本身就是使用 C++编写的,允许用户从已存在的类中派生出自己的类,通过派生的方式可以很容易地扩展 OIV 库。OIV 支持场景、观察器和动作等高级功能,用户可以把 3D 物体保存在场景中,通过观察器来显示 3D 物体。利用动作可以对 3D 物体进行特殊的操作(例如拾取操作、选中操作等)。正是因为有了这些高级功能,才使得普通程序员也能编写出功能强大的三维交互式应用软件。

Open Inventor 应用领域包括商业图形、机械 CAE 和 CAD、绘画、建筑设计、医学和科学图像、化学工程设计、地学、虚拟现实、科学数据可视化和仿真动画。

习 题

1. 试比较计算机图形学与图像处理、模式识别的共同点和不同点。
2. 试举出几个计算机图形学的应用实例。

第二章 图形基元的显示

光栅扫描图形显示系统的出现极大地推动了计算机图形学的发展。在光栅扫描显示器中,帧缓冲存储器中每个地址都必须有正确的图形亮度值,图形描述模型中的点、线和面的表示都必须转换成存储图形的像素矩阵表示。将图形描述转换成用像素矩阵表示的过程称为扫描转换。每次图形改变都要进行扫描转换,因此高效快速的扫描转换算法是非常重要的。

直接采用像素矩阵描述图形显然是不方便的。实际图形往往由一些基本图形组合产生。图形基元,或称输出图形元素,是图形系统能产生的最基本图形。图形基元的选择可以不同,但通常都把线段、圆、椭圆、多边形等图形选为图形基元。本章将讨论线段、圆、椭圆、多边形等图形基元的扫描转换算法。

第一节 直线扫描转换算法

一、DDA 直线扫描转换算法

在高级图形系统中,画一条线段通常要求给出线段的两个端点,线段扫描转换算法的任务是根据端点位置求出构成该线段的所有像素位置的坐标。如图 2-1 所示,在光栅扫描显示方式中像素坐标是行和列的位置值,只能取整数,是理想线段上点坐标的近似值。显然,当光栅扫描图形显示器的显示分辨率较低时,画出的线段会呈现阶梯状。好的线段扫描转换算法画出的线段应尽可能逼近原直线,线的亮度应均匀且与线的方向无关,线段两端截断要精确,要能够尽可能快速地产生整条线段。

设待画线段两端点的坐标值是 (x_1, y_1) 和 (x_2, y_2) ,不妨假定 $x_1 < x_2$,待画线段所在直线方程是

$$y = mx + b \quad (2-1)$$

则有

$$m = \frac{y_2 - y_1}{x_2 - x_1}, b = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} \quad (2-2)$$

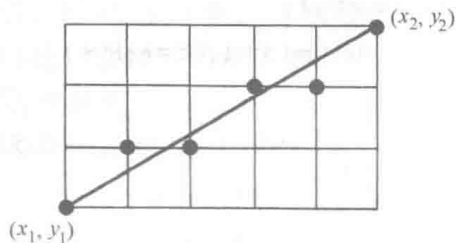


图 2-1 光栅扫描方式显示的直线

于是得到线段扫描转换最直接的算法:给出 (x_1, y_1) 和 (x_2, y_2) ,利用式 (2-2) 求出 m 和 b 。当 $|m| \leq 1$ 时,对 x 每增加 1 取允许的各整数值,用式 (2-1) 进行乘法和加法运算求 y 后再取整;当 $|m| > 1$ 时,应先对 y 取各整数值再计算 x ,如图 2-2 所示。这个算法计算量大,画线慢,不是好的扫描转换算法。

注意到对直线,其数值微分是

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (2-3)$$

由式 (2-3) 得, $y_{i+1} = y_i + m(x_{i+1} - x_i)$,可知,当 x_i 增加 1,即 $x_{i+1} = x_i + 1$ 时, $y_{i+1} = y_i + m$ 为画线精确,应使画出的相邻点的坐标值最大相差 1,这样便得到绘制线段的数值微分分析

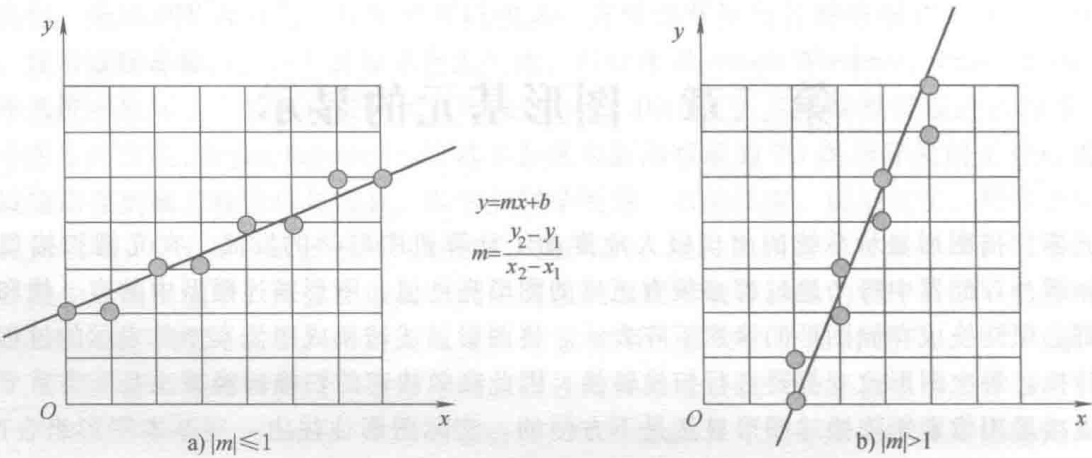


图 2-2 $|m| \leq 1$ 和 $|m| > 1$ 时的直线扫描转换算法

器 (Digital Differential Analyzer, DDA) 算法, 如下:

算法 2-1 数值微分分析器算法

```
void DDALine(int x1,int y1,int x2,int y2)
```

```
{
    double dx,dy,e,x,y;
    dx=x2-x1;
    dy=y2-y1;
    e=( fabs(dx)>fabs(dy) )? fabs(dx):fabs(dy);
    dx/=e;
    dy/=e;
    x=x1;
    y=y1;
    for(int i=1;i<=e;i++)
    {
        SetPixel( (int)(x+0.5), (int)(y+0.5));
        x+=dx;
        y+=dy;
    }
}
```

算法中 $(int)(x+0.5)$, $(int)(y+0.5)$ 是四舍五入后取得的整数。注意到算法在逐点计算的循环中已经没有乘除运算, 但计算下一点位置的加法是实数运算。

下面给出采用 DDA 算法绘制从 $P_0(0, 0)$ 到 $P_1(5, 2)$ 直线段的例子。各步数据见表 2-1, 绘制结果如图 2-3 所示。

表 2-1 采用 DDA 算法绘制直线段的各步数据

x	0	1	2	3	4	5
int(y+0.5)	0	0	1	1	2	2
y+0.5	0	0.4+0.5	0.8+0.5	1.2+0.5	1.6+0.5	2.0+0.5