

计算机图形学 习题与解答

COMPUTER GRAPHICS

最佳的复习资料，实用的辅助教材

与国外高校计算机水平保持同步

为考研和出国深造奠定坚实基础

Zhigang Xiang Roy Plastock 著

陈泽琳 陈展文 陈屹 等译

Mc
Graw
Hill



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

全球销售超过
3000万册

**全美经典
学习指导系列**

计算机图形学 习题与解答

COMPUTER GRAPHICS

Zhigang Xiang Roy Plastock 著

陈泽琳 陈展文 陈屹 等译



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书主要讲述了计算机图形学中的关键内容，清晰地介绍了计算机图形学的基本概念和它们之间的关系，也描述了计算机图形学的整体框架，并着重于数学结果和算法实现。书中所有的示例都可以很容易地转换成一个特定计算机的工作版本，使读者能够更好地了解图形学的技术用途，并可以自己动手实现图形显示。

Zhigang Xiang, Roy Plastock: Computer Graphics .

Copyright © 2000 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press & CITIC Publishing House.

本书中文简体字翻译版由机械工业出版社、中信出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书版权登记号：图字：01-2002-0354

图书在版编目（CIP）数据

计算机图形学习题与解答/（美）向志刚（Xiang, Z.）等著；陈泽琳等译。—北京：机械工业出版社，2002.8

（全美经典学习指导系列）

书名原文：Computer Graphics

ISBN 7-111-10822-1

I. 计… II. ①向… ②陈… III. 计算机图形学—解题 IV. TP391.41-44

中国版本图书馆CIP数据核字（2002）第062692号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码100037）

责任编辑：华章

北京忠信诚印刷厂印刷·新华书店北京发行所发行

2002年8月第1版第1次印刷

787mm×1092mm 1/16·19印张

印数：0 001-5 000册

定价：29.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

作 者 序

我们生活在一个科技飞速发展的时代。近年来,人们已越来越关注发展十分迅速的、称之为计算机图形学的学科。每一台计算机几乎都装有图形用户界面(GUI),用户和计算机系统可以通过文字或图形的形式进行人机交互;无论是年轻还是年长,人们都对用最新科技产生的流行的电影和视频游戏感兴趣;只要你坐到电视机前,很快就会发现计算机图形学在广告中的神奇表现。

本书既是学习计算机图形学基本原理的一本独立的教材,又是一本有价值的辅助参考书。本书采用面向学习目标的编写方法,讨论计算机图像生成过程中的重要概念、基本数学知识和算法的实现。书中包含了数百道题解,有助于读者加强对图形学领域概念的理解,同时用实例说明高效的解题技巧。

虽然本书的主要读者对象是计算机科学或计算机工程专业学习计算机图形学课程的学生,但是任何受过教育并希望对计算机图形学有所了解的人士都应该能从中学到入门的知识。学习本书推荐的预备知识包括:简单的计算机系统的工作原理,相当于一到两个学期的程序设计,对数据结构和算法有初步了解,以及线性代数和解析几何的基础知识。

计算机图形学领域有两个特点,其一是最新应用的技术发展迅速,其二是图形系统层出不穷。图形硬件的生命周期似乎变得越来越短,一种计算机图形学的工业标准往往在其完成前就可能已经过时了,而一种设计图形应用程序的编程语言,可能在学生上学期间还是流行的工具,到毕业时已经处于被淘汰的边缘。

本书试图覆盖计算机图形学中相对比较持久的关键内容。我们力争清晰地解释基本的概念和它们之间的相互关系,而不去描述那些与高级设备相关或与计算环境相关的信息。我们讨论计算机图形学的整体框架,并着重于数学结果和算法实现。书中的算法用伪代码表示,而不用某种特定的编程语言。所有例子均有详细说明,可以非常容易地转换成一个特定计算机系统的工作版本。

我们相信,这种编写原则可以使不同的读者都从中获益。首先,对那些只想了解图形学技术而不做具体实现的人来说,这是一本计算机图形学的入门书;其次,对于教师和学生来说,本书可以作为任何综合基础教材的补充;第三,对于从事图形学工作的人来说,本书可以作为实践者的阶梯,因为这本书比他们要看的图形系统程序员手册更容易理解。

本书的第一版很好地为读者服务了十多年,在此笔者对做了大量基础工作的同事们表示敬意和感谢。现行版本对原版本进行了重大改进,有些章节完全被新的内容替换,全书其余的部分也都有所更新。笔者希望,此书能在未来的若干年中再给读者提供帮助。

感谢你选择我们的书,愿此书能激励你并使你获得回报。

目 录

第 1 章 绪论	(1)
1.1 概述	(1)
1.2 进一步学习	(4)
第 2 章 图像表示	(5)
2.1 RGB 颜色模型	(5)
2.2 直接编码	(6)
2.3 查找表图像表示法	(7)
2.4 显示器	(8)
2.5 打印机	(10)
2.6 图像文件	(13)
2.7 设定像素的颜色属性	(13)
2.8 例子:可视化 Mandelbrot 集	(14)
习题与解答	(17)
补充题	(21)
第 3 章 扫描转换	(22)
3.1 扫描转换一个点	(22)
3.2 扫描转换一条直线	(23)
3.3 扫描转换一个圆	(26)
3.4 扫描转换一个椭圆	(30)
3.5 弧和扇形弧的扫描转换	(35)
3.6 矩形的扫描转换	(36)
3.7 区域填充	(36)
3.8 扫描转换一个字符	(40)
3.9 反走样	(41)
3.10 例子:递归定义的图形	(44)
习题与解答	(46)
补充题	(58)
第 4 章 二维变换	(59)
4.1 几何变换	(59)
4.2 坐标变换	(61)

4.3 组合变换	(62)
4.4 图例变换	(65)
习题与解答	(66)
补充题	(75)
第5章 二维观察和裁剪	(76)
5.1 窗口到视区映射	(77)
5.2 点裁剪	(78)
5.3 线段裁剪	(78)
5.4 多边形裁剪	(82)
5.5 例子:二维图形处理流程	(84)
习题与解答	(86)
补充题	(97)
第6章 三维变换	(99)
6.1 几何变换	(99)
6.2 坐标变换	(101)
6.3 组合变换	(102)
6.4 图例变换	(102)
习题与解答	(103)
补充题	(109)
第7章 数学投影	(111)
7.1 投影分类	(111)
7.2 透视投影	(111)
7.3 平行投影	(114)
习题与解答	(115)
补充题	(129)
第8章 三维观察和裁剪	(130)
8.1 三维观察	(130)
8.2 裁剪	(133)
8.3 观察变换	(135)
8.4 例子:三维图形的处理流程	(136)
习题与解答	(137)
补充题	(149)
第9章 几何表示	(150)
9.1 简单几何形体	(150)
9.2 线框模型	(151)
9.3 曲面表示	(152)

9.4 曲线绘制	(152)
9.5 多项式基函数	(153)
9.6 插值问题	(155)
9.7 逼近问题	(156)
9.8 曲面绘制	(159)
9.9 曲线和曲面变换	(160)
9.10 二次曲面	(161)
9.11 例子:地形图生成	(163)
习题与解答	(165)
补充题	(169)
第 10 章 隐藏面消隐	(170)
10.1 深度比较	(170)
10.2 Z-缓冲器算法	(172)
10.3 后向面消隐	(173)
10.4 画家算法	(173)
10.5 扫描转换算法	(175)
10.6 区域再分算法	(178)
10.7 消除隐藏线	(179)
10.8 数学曲面的绘制	(180)
习题与解答	(182)
补充题	(196)
第 11 章 颜色和明暗处理模型	(197)
11.1 光和颜色	(197)
11.2 PHONG 模型	(201)
11.3 插值的明暗处理方法	(203)
11.4 纹理	(205)
习题与解答	(208)
补充题	(214)
第 12 章 光线跟踪	(215)
12.1 单孔摄影	(215)
12.2 递归的光线跟踪算法	(216)
12.3 光线的参数向量表示	(218)
12.4 光线与物体表面求交	(218)
12.5 执行效率	(220)
12.6 反走样	(222)
12.7 其他视觉效果	(223)

习题与解答	(224)
补充题	(231)
附录 1 二维计算机图形的数学基础	(233)
A1.1 二维笛卡儿坐标系	(233)
A1.2 极坐标系	(236)
A1.3 向量	(237)
A1.4 矩阵	(239)
A1.5 函数和变换	(241)
习题与解答	(242)
附录 2 三维计算机图形的数学基础	(255)
A2.1 三维笛卡儿坐标	(255)
A2.2 三维曲线和三维曲面	(256)
A2.3 三维向量	(258)
A2.4 齐次坐标系	(261)
习题与解答	(263)
补充题答案	(272)

第1章 绪 论

计算机图形学是计算机科学的一个分支,它涉及数字化图像合成的理论和技术。计算机生成的图像既可以是简单的单色背景上的一个三角形图形,也可以是复杂的热带森林中的巨型恐龙。它们如何成为计算机的图形?什么使得计算机图形有别于铅笔画的草图或相机拍的照片?本章我们将引入一些基本概念,同时简述这些概念之间的关系。简述计算机图形学的目的是为了弄清上述问题,本书的其余部分将在问题的局部和整体上进行详细的讲解。

1.1 概述

首先考虑画一个三角形(见图 1-1)。现实生活中我们先在心里确定这样一个几何形状,如三角形的类型和大小,然后再将它画在纸上。若将这个过程用计算机图形学的术语表示,则我们心里想像的工作叫物体定义,即在选择的一个抽象空间中定义了这个三角形,该空间是连续的,我们称之为物空间。把想像的物体映射成纸上的三角形,这个动作构成了另一个空间中的一个连续平面,这个空间称之为像空间。同时,这个映射过程还受到所选择的三角形位置和方向的影响。换言之,我们既可以把这个三角形放在纸的中间,也可以放在纸的左上角;既可以把三角形的尖角指向右边,也可以指向左边。

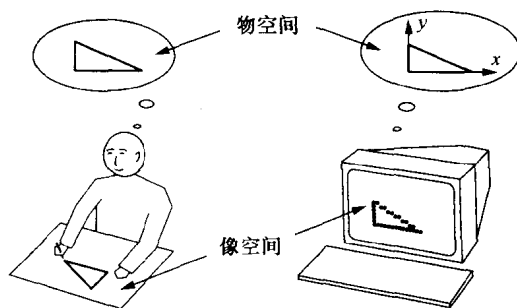


图 1-1 画一个三角形

计算机在产生这个图形时也有类似的过程,这个过程的主要步骤构成了计算机图形学的几个重要领域。以快速和有效的方式定义物体(例如三角形)的领域称为几何表示。在上例中可以把一个二维笛卡儿坐标系放在物空间,三角形就能用三个顶点坐标 x 和 y 表示出来了,计算机系统将把顶点 1 和顶点 2 用线段连起来,顶点 2 和顶点 3 用另一条线段连起来,同样顶点 3 和顶点 1 也用一条线段连起来。

处理三角形位置的下一个计算机图形学领域称之为变换,利用矩阵运算可以实现该三角形从物空间到像空间的映射。可以建立变换矩阵来控制显示三角形的位置和方向,甚至可以放大或缩小三角形的尺寸。而且,对同一个物空间的模型,通过使用多个矩阵变换,可以使计算机产生多个位置不同、大小不同或方向不同的三角形。

至此,许多读者可能感到奇怪,画在纸上的三角形和显示在屏幕上的三角形为什么有如此大的差别(如屏幕上夸张的图形),前者是用光滑的边连起来的,而后者实际上就不是一条线。其根本原因在于,通常计算机图形学中的像空间是不连续的,它由一组按行和列排列的离散点(即像素)组成,因此,一条水平线或垂直线就分别是连续的一行或一列像素,而一条斜线段就是由像楼梯似的一组像素组成。将连续图形(如一条线段)转化为一组离散的像素点,这是计算机图形学的另一个领域,称为扫描转换。

连续空间到离散空间的转换所引起的失真称之为走样。当一个像素的尺寸足够的小,这种失真就不那么明显了,但这样做需要使用许多计算机的资源。举例来说,如果把每个像素在水平方向和垂直方向都切成一半的话,那么要保持图形相同的物理尺寸就需要原来 4 倍数量的像素,因而存储这个图像需要 4 倍的内存。用其他计算机算法解决走样问题是计算机图形学另外一个重要的研究领域,称之为反走样。

把上述讲过的内容放在一起形成了一个简化的计算机图形实现流程(见图 1-2),也勾画出了一个典型的图形系统的结构。最初我们有与应用相关的数据结构表示原始的物体,例如,可以将三角形的顶点坐标: (x_1, y_1) 、 (x_2, y_2) 、 (x_3, y_3) 简单地存放到一个 3×2 的数组中,图形系统首先根据用户定义的参数进行原始数据的变换,然后进行扫描转换把图形显示在屏幕上,该扫描转换也可以进行反走样。图 1-2 中间方框中的坐标系是世界坐标系,它是连接左侧物坐标系和右侧象坐标系(或设备坐标系)之间的纽带,变换后的物体表示在世界坐标系下,并用于构成最后的图形。例子中的中间方框里有两个三角形:右边的是对原始三角形进行成比例缩放后又向上和向右移动,左边的是经过成比例缩放后逆时针旋转 90° 再用同样方法向上和向右移动。

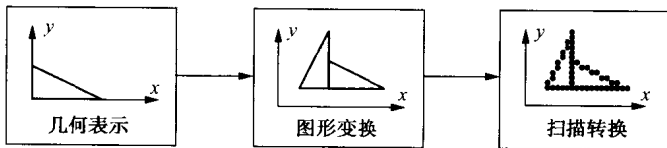


图 1-2 一个简单的计算机图形实现流程

计算机图形的典型实现过程如下:用宿主语言编写应用程序,调用子程序库执行图形操作。一些子程序通过参数实现变换,还有一些子程序实现画图,即从实现流程的一头输入原始数据,系统自动地把这些数据变成最终结果从实现流程的另一头输出,这就是屏幕上的图形。

二维图形的主要内容介绍到这里,我们再来看一下三维图形的处理过程。增加了第三维,则物体和图像之间的处理过程变得更加复杂。图 1-3 表示了几种可能的立方体的画法,但没有一种能够准确地表示物体本身,这些图形只是表示了该三维物体在二维平面上的投影,这也

意味着除了三维表示和变换以外,计算机图形学还有另外一个研究领域——投影方法。

你是否注意到图 1-3 中的任何一个图形都只是显示立方体的三个面,而实际上三维立方体有六个面。但是,画出的图形和现实生活中所看到的是一样的。我们只画出了那些可见的面,不可见的面没有画出来。处理这个

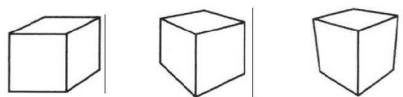


图 1-3 显示立方体的几种方法

计算任务在计算机图形学领域中称之为隐藏面消除。在前面所述的简单的计算机图形实现过程中,若在图形变换和扫描转换之间增加投影和隐藏面消除,则就构成了三维计算机图形的实现流程。

现在我们来考虑这样一个问题:要产生一个和现实生活中的物体一样的图形,这给计算机图形学带来了巨大挑战,因为有一个非常有效的办法来形成这样的图形,即摄影。为了使计算机生成真实感效果的图形,即看起来和照片一样,则需要研究照相机的成像原理。

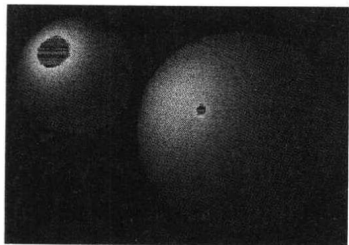


图 1-4 两个有明暗的球体

当我们用相机给由光源照着的实际物体拍照时,光源产生的光能从物体表面通过相机镜头反射到底片上,形成这个物体的影像。一般而言,物体离光源近的部分在照片上则显得亮一些,离光源远的部分则显得暗一些,同时物体背向光源的部分则相对更暗一些。图 1-4 显示了有一个光源照着的两个球体的计算机图像,大约 10 点~11 点的太阳光源照射在球体和“相机”之间。尽管两个球体都有渐进的明暗,但是大球的亮点看起来好像是光源的反射,从而引起了两个球反射性质的不同

(大球要比小球亮)。由于直接从光源照射到物体表面的能量不是到达该表面能量的全部,因此模拟这类光学现象的数学公式称之为局部光照模型。光能还可以从一个物体的表面反射到另一个物体的表面,也能穿过透明或半透明的物体到达其他位置,用于提供比局部光照模型更精确的光传递量的计算机方法称之为全局光照模型。

现在可以来仔细看一下图 1-4,这两个物体似乎具有超光滑的表面。它们是用什么造的?它们怎么能如此完美呢?在你周围是否看见过许多具有这样表面特征的实际物体?此外,小球好像放在光源和大球之间,难道我们不应该看见它在大球中的影子吗?在计算机图形学中,区别木质表面、大理石表面或者其他类型表面的明暗变化称之为表面纹理,有许多种方法可以把表面纹理附加到物体上,使物体看上去更真实。另外,在图形中包括阴影计算的工作称之为阴影生成。

在详细研究上述专题领域之前,首先来简单讨论一下与计算机图形学相关的其他计算机科学领域。

图像处理

图像处理(或数字图像处理)与计算机图形学的根本区别是:图像处理的工作是从存在于图像空间中的图像开始,通过完成一系列基于像素的操作,产生能够展示某些期望特征的新图像。例如,可以重新设置图 1-1 屏幕上图像的每一个像素的颜色并为其补色(例如黑变白或白

变黑),把白色背景上的黑三角形变成黑色背景上的白三角形,或者相反。图像处理与计算机图形学两个领域有各自不同的侧重点,互相之间也是交叉和互补的。事实上,惊人的视觉效果经常是通过计算机图形学和图像处理技术两者的结合得到的。

人机交互

虽然计算机图形学的主要课题是研究图像的产生,但人机交互领域促进了人与机器的有效交流,两者的结合促使了图形用户界面的诞生。从键盘到鼠标,有很多物理设备可以附加到计算机上提供人与机器的交互,每个物理设备都可以通过编程来完成各种逻辑设备的功能(例如位、选择——见下文)。例如,鼠标可以用来定义图像空间的位置(作用像一个定位设备),在此情形下,经常显示一个光标作为反馈让用户看到他定义的位置。也可以用鼠标在下拉菜单或弹出菜单中选项(作用象一个选择设备),此时它是选中菜单项的指示,通常该菜单项整体高亮显示(基本与光标的绝对位置无关)。由此我们看到,一个物理设备可以有不同的用途,信息也可以用不同的图形形式传递给用户。关键的问题是设计一种设备和图形能有效交流的交互协议,即用户界面友好——易用、直观、高效等。

1.2 进一步学习

我们刚刚对计算机图形学这个美丽的王国进行了短暂的游览,希望那些重要领域能够给你留下较深的印象,同时激起你继续探索的渴望。后面的章节将进一步展开计算机图形学各种课题领域的研究,每一章开始都有必须的背景信息(例如来龙去脉和术语)及以后各节要讨论的内容概述。

我们在书中力图提供清晰的解释并保持各章节之间的连贯性,恰当地运用各种例子来说明所讨论的抽象概念。尽管本书的主要任务是对图形学的基本理论和技术作集中式的入门介绍,但为了使具备合理的、广泛的相关知识,我们还对基本定义和实现约定等的一些重要变化作了介绍。此外,也尽可能早地引进一些应用实例来突出图形学技术的用途,当然也以此激励那些渴望参与实践的学生。

本书的算法和程序例子是用类 C 语言的伪代码表示的,其语法和基本结构与广泛应用的 C++ 和 JAVA 类似。希望籍此减少语法困难以便把你的精力放到代码表示的技术实质上来。

每章末尾附有大量已经解决的习题帮助你强化理论探讨,其中有些习题介绍了正文中省略的计算步骤,它对于寻找问题进一步的细节和解释非常有用,其他习题则为正文讨论的内容补充了新的信息。

第2章 图像表示

数字图像(或简称图像)是由离散的像素点(pixel)构成。这些像素点通过按行列排列的方式构成一个矩形图片区域,有时也称为光栅。显然,一幅图像的像素点数目由图像的大小和水平、垂直方向上单位长度(如英寸)的像素点数目决定。这里的单位长度像素点数目是指图像的分辨率。因此一个3×2英寸大小的图像在300像素/英寸分辨率下共有540 000个像素。

通常,图像的大小(即图像总的像素个数)由水平方向像素个数乘以垂直方向像素个数表示(例如,512×512、640×480或1024×768)。虽然这种表示法能够简明地表示一幅图像的总像素数目,但却无法表示前面提到的图像大小和分辨率。例如,在一幅640×480大小的图像中,分辨率为96像素/英寸的图像大小是 $6\frac{2}{3} \times 5$ 英寸(例如,显示或打印),而在400像素/英寸的分辨率下表示为1.6×1.2英寸。

无论是用单位长度度量还是用像素数目度量,一幅图像的宽和高的比率都可以称为纵横比。2×2英寸大小的图像与512×512大小的图像的纵横比都是1/1,而 $6 \times 4\frac{1}{2}$ 英寸大小的图像与1024×768大小的图像的纵横比都是4/3。

图像的像素点可以通过坐标表示。通常图像左下角的像素点表示为像素坐标系的原点(0,0)。因此,640×480大小的图像右下角的像素点坐标为(639,0),右上角的像素点坐标为(639,479)。

计算机上创造的一幅图像实质上是设置像素点的值。不同颜色属性的像素组合在一起即构成了我们所看到的图。在本章中,首先介绍了在计算机图形学里最常用的颜色定义方法(见2.1节)。接着介绍直接像素颜色编码的图像表示方法(见2.2节)和颜色查找表的图像表示方法(见2.3节)。然后介绍两种典型的图像输出设备(显示器见2.4节,打印机见2.5节)的主要工作原理。在本章的2.6节将分析几种主要用于存储和传输图像的文件格式。在本章的2.7节介绍一些最基本的图形操作,主要有设置像素的颜色属性。最后,为了说明如何在离散的图像空间中直接创建美丽的图像,介绍了实现可视化Mandelbrot集合所要用到的数学背景和算法细节。

2.1 RGB 颜色模型

颜色是一种复杂的、从物理学到心理学的跨学科的概念。在这一节中只是介绍了计算机图形学中最基本的、应用最广的颜色表示方法,在以后的章节中将会进一步讨论有关颜色的概念。

图 2-1 表示了一个具有三基色:R(红)、G(绿)和 B(蓝)的坐标系。每一种基色的亮度可以从 0 到 1,即从最暗到最亮(或从关到开)。通过混合不同亮度的三种基色可以表示多种颜色。通过红绿蓝三基色的线性组合所得到的所有颜色形成了一个立方体形状的 RGB 颜色空间。RGB 颜色立方体中的坐标原点的顶角表示黑色,而与坐标原点对角线相对的顶角表示白色。介于黑色和白色之间的对角线上的点表示两种颜色之间的灰度值。这条线也称作灰度轴。

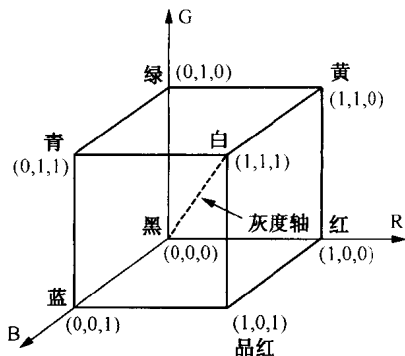


图 2-1 RGB 颜色空间

在 RGB 颜色模型中,用颜色坐标 (r, g, b) 可以表示颜色立方体中的任意一个颜色。例如, $(0, 0, 0)$ 表示黑色, $(1, 1, 1)$ 表示白色, $(1, 1, 0)$ 表示黄色等。在 $(0.7, 0.7, 0.7)$ 处的颜色灰度值是介于 $(0.9, 0.9, 0.9)$ 和 $(0.5, 0.5, 0.5)$ 之间的灰度值。

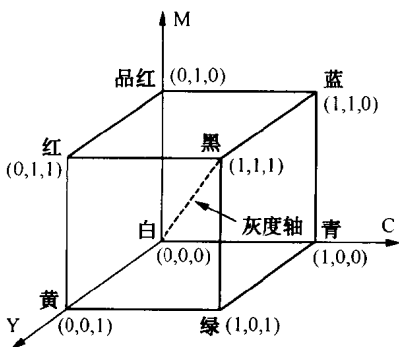


图 2-2 CMY 颜色空间

使用 RGB 模型定义颜色是一种加色的过程。由黑色开始,接着加入合适的基色元素得到希望的颜色。这种方法和显示器的工作原理很相似(见 2.4 节)。另一方面,还有一个补色模型——CMY 颜色模型,它定义颜色的过程是一种减色过程,这种方法和打印机的工作原理很相似(见 2.5 节)。

在 CMY 颜色模型中,由白色开始,接着减去合适的基色元素得到希望的颜色。例如,如果从白色中减去红色,剩下的颜色为绿色和蓝色,则组成了青色。从另一个角度看,可以通过红色的补色青色的量度来控制红色的量度,等价于 1 减去青色的数量。图 2-2 说明了使用三基色的补色:C(青),M(品红)和 Y(黄)表示的坐标系。CMY 颜色立方体中的坐标原点 $(0, 0, 0)$ 的顶角表示白色,坐标为 $(1, 1, 1)$ 的顶点表示黑色(不是红或绿或蓝)。下面的公式概括了两模型之间的转换:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix} \quad \begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

2.2 直接编码

图像表示本质上说是像素颜色的表示。使用直接编码时,可为每个像素分配一定的存储空间以用于颜色编码。例如,可为每个像素分配 3 比特空间,可每一比特表示一种基色(见

图 2-3)。这种 3 比特表示法使得每种基色各自用两个亮度值表示:0(关)或者 1(开)。因此每一个像素可以表示 8 种在 RGB 颜色立方体顶角所代表的颜色。

比特 1:r	比特 2:g	比特 3:b	颜色名
0	0	0	黑
0	0	1	蓝
0	1	0	绿
0	1	1	青
1	0	0	红
1	0	1	品红
1	1	0	黄
1	1	1	白

图 2-3 用 3 比特的直接编码

一个公认的标准是每个像素使用 3 个字节,即 24 比特,其中每种基色都用一个字节表示。通过这种方法,每种基色有 256 个亮度等级,从二进制的 00000000 到 11111111。因此一个像素可以表示 $256 \times 256 \times 256$ 或 1.67 千万种可能的颜色,这种 24 比特格式通常被称为真彩色表示。一般情况下,因为人眼不易察觉由于某个或多个基色亮度不同而造成的两种颜色的差别,因此使用更多的比特来更精确地表示颜色是没有什么意义的。

直接编码的一个特例是黑白(二值表示)和灰度映像表示。在这两种图像中,三种基色的值都是一样的,因此不需要用三个分量编码。每一个黑白图像的像素只需要 1 个比特表示,0 代表黑色,1 代表白色。每一个灰度映像的像素可以使用 8 个比特表示共 256 种亮度或者灰度级别。

尽管这种直接编码方法比较简单,而且支持很多应用程序,但是这种方法在使用 24 比特标准时对存储空间有很高的要求。例如,一幅 1000×1000 的真彩色图像约占用 3 兆字节的存储空间,或者说,如果图像里每一个像素的颜色都彼此不同,图像里就有一百万种不同的颜色。在许多应用程序中,一幅特定的图像所包含的颜色数量实际上比一百万种颜色数要少得多。因此,在一幅图像中用 24 比特表示 1.67 千万种不同颜色的表示法就好像是大材小用。

2.3 查找表图像表示法

使用查找表表示图像时,既可以减少存储空间,同时又能够表示足够多的颜色。在这个表示方法中,不是直接对像素值进行编码,取而代之的是用寻址或索引的方法表示颜色。一个像素的颜色是由其查找表对应元素的颜色值确定的。

图 2-4 是一个有 256 个表元素的查找表。表元素的值从 0 到 255。每一个表元素包含一个 24 比特 RGB 颜色值。每个像素由 1 个字节或 8 比特表示。像素值为 i 的颜色是由地址为 i 的表元素决定,其中 $0 \leq i \leq 255$ 。这种 24 比特的 256 个表元素的查找表通常被称为 8 比特格式。用这种方法表示 1000×1000 的图像时,存储容量为 1 兆字节加上 768 个字节。这种方法可以同时表示的颜色数目是 1.67 千万种颜色中的 256 种颜色。

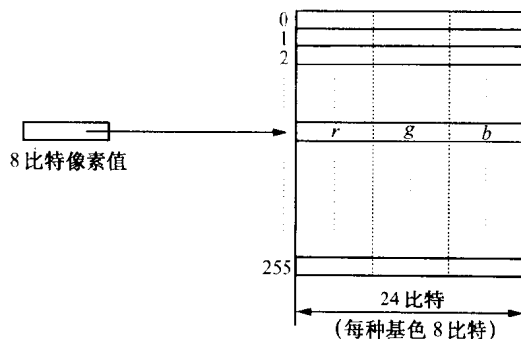


图 2-4 24 比特 256 个表元素查找表

值得注意的是,使用查找表方法时,一幅图像不仅由它的像素值决定,而且还由对应的查找表颜色值决定,这些颜色值构成了这幅图像的一个颜色映射图。

2.4 显示器

有多种把数字化的图像转化为可视化图片的图像显示设备,显示器或视频监视器是其中的一种。

首先介绍单色显示器的工作原理。它主要由一个阴极射线管(CRT)和相应的控制电路组成。CRT 是一个真空玻璃管,它的一边是荧光屏,另一边是连接在控制电路上的插脚(见图 2-5)。有一种特殊材料涂在荧光屏内表面,这种物质就是荧光粉,在受到电子束冲击后这种材料会发出一段时间的光点,光的颜色和持续时间因荧光粉而异。这种由于电子束冲击荧光粉发出的光就是所谓的荧光,当电子束消失后依然存在光的现象就是荧光现象,荧光的持续现象就是余辉现象。

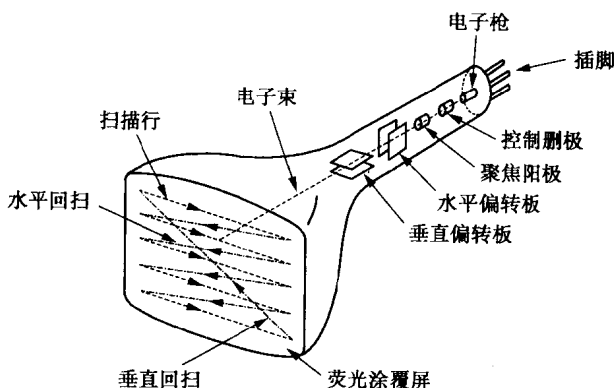


图 2-5 CRT 单色显示器示意图

当荧光涂覆屏对面的电子枪加热时会发出电子。电子由控制栅极产生,通过聚焦阳极将

电子束变为一个狭窄束投射到荧光屏产生小光点。当电子束通过水平偏转板和垂直偏转板时,电子束会因电磁场的作用而产生偏转和弯曲。水平偏转板控制电子束从左到右扫描并从右到左回扫。垂直偏转板控制电子束从上到下扫描并从下到上回扫。这些动作是由控制电路同步产生的,因此电子束可以投射到每一行的每一像素点位置。静电偏转阴极射线管可以由环绕 CRT 外部的磁偏转阴极射线管产生的磁场来替代。

荧光屏的亮度是由电子束强度决定的。在水平和垂直回扫的过程中,控制电路会关闭电子束的发射。电子束在某一像素点位置的亮度是由对应的图像中需要显示的像素点的亮度值所决定的。

要显示的图像存储在一个专用的系统内存空间里,通常称为帧缓冲器或刷新缓冲器。控制电路根据帧缓冲器产生正确的视频信号传送给显示器。帧缓冲器的内容送到显示器的频率称为刷新频率,典型的刷新频率是每秒 60 帧或者更高。显示器的一个主要考虑因素是避免产生屏幕闪烁,而闪烁通常是由于低的刷新率使得荧光屏产生的光脉冲不能在我们的视觉系统中形成稳定的图像所造成的。换句话说,为了保持显示器荧光持续发光,不但要使一帧屏幕持续可见的时间足够长,还要在下一帧屏幕切换显示时荧光消失的时间足够的短。

有一些显示器采用隔行扫描的方式“双倍”提高它们的刷新率。在这种情况下,每次只有帧中的一半扫描行被刷新,首先是奇数行,接着是偶数行。因此,从屏幕的顶部到底部扫描一次的时间只是逐行扫描方式的一半。虽然这种方法并不是真正的提高刷新频率,但是这种方法有效地减少了屏幕的闪烁现象。

彩色显示

在彩色显示器 CRT 中,使用的不再是一支电子枪,而是三支电子枪(见图 2-6),每支电子枪发射一种基色。显示屏内部的荧光涂覆层是由三种不同的荧光粉的点模式构成。这些荧光物质可以分别发出红绿蓝三种光。相邻点模式之间的中心点距离称为彩色 CRT 的间距。它确定了显示屏幕可编址的上限。在荧光屏和电子枪之间有一个很薄的金属屏幕,称为荫罩。荫罩上的小孔限制了电子束只能发射到荧光屏上对应的点。从一定距离看过去,由三支电子枪发射的三种不同荧光聚合在一起则产生了多种颜色。

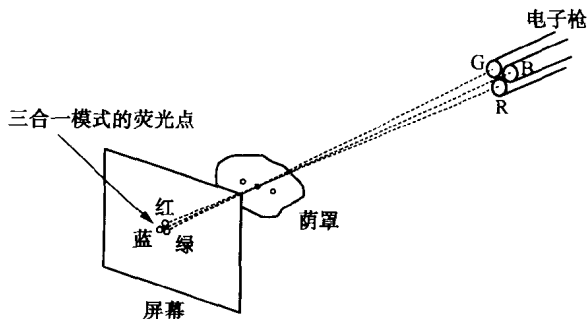


图 2-6 使用荫罩的彩色 CRT