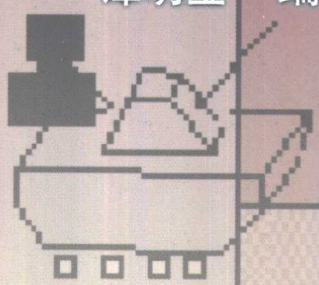
- 体现数字图形图像处理内涵
- 讲解动画显示效果实用技能
- 示例手写字体描述和显示
- 分析像素放大、数字地图等高级应用

Visual C++

图形编程技巧与实例

谭明金　编著



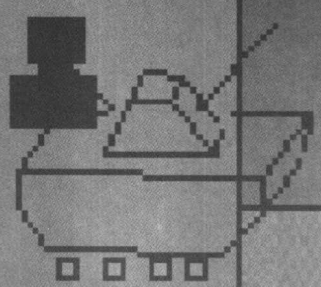
附光盘

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

Visual C++

图形编程技巧与实例

谭明金　编著



人民邮电出版社

图书在版编目（CIP）数据

Visual C++图形编程技巧与实例/谭明金编著.—北京：人民邮电出版社，2002.9
ISBN 7-115-10574-X

Ⅰ．V…　Ⅱ．谭…　ⅢC 语言—程序设计　Ⅳ．TP312

中国版本图书馆 CIP 数据核字（2002）第 063306 号

内 容 提 要

本书重点介绍了在 Windows 95/98/NT 环境下，编写图形应用程序所需要的典型技术和算法知识，并给出了实用性和技巧性很强的典型实例。

全书分为 8 章。每章以一个可独立运行的应用程序来综合体现某一方面的图形应用知识和编程技术。书中的每个程序都是经过精心挑选的实例程序，是实践经验的总结，在功能和实现方面充分体现了典型性、综合性和实用性。全书各章一致地按照“先原理后实现、精原理重实现”的指导思想和结构形式来组织内容的叙述，即在简要介绍一般知识和算法原理的同时，着重叙述程序的实现细节，并给出了注释详细的完整代码（全部代码用 Visual C++编写）与应用结果。此外，本书对 Windows 编程中的一些具有综合性和先进性的编程知识也适当地进行了介绍。

对于图形图像编程的初学者和从事图形图像应用开发的工程人员，本书具有较高的学习和参考价值。

Visual C++图形编程技巧与实例

◆ 编　　著　谭明金
责任编辑　王文娟

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
读者热线　010-67180876
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：29
字数：711 千字　　2002 年 9 月第 1 版
印数：1-5 000 册　　2002 年 9 月北京第 1 次印刷

ISBN7-115-10574-X/TP・3048

定价：48.00 元（附光盘）

前　言

计算机多媒体技术的发展，使图形图像这类具有独特优势的信息表达和传播手段，在社会生活的各个领域中都得到充分应用，并发挥着越来越重要的作用。但由于图形图像应用的技术含量和艺术含量都很高，实现起来难度较大，工作量也较大，从而对应用人员的相关知识和技能要求较高，这使图形应用的普及受到很大的限制。

有关计算机图形图像方面的书籍很多，这些书籍大致可以分为三种，一种是介绍图形图像处理的一般知识的（如基本原理、算法或相关硬件知识等），一种是介绍编程语言和开发系统的使用知识的（如 Windows 图形编程知识或 OpenGL 编程技术等），一种是介绍如何使用一些图形工具软件来进行图像处理的（如 Adobe Photoshop 或者 3DS Max 的使用知识等）。

本书的重点不在于介绍这些方面的内容，而是力图通过一些具有典型应用功能的实用程序，来体现图形编程的实用技术和技能。本书是在作者实际工作经验的基础上写成的。本书将通过一些具有趣味性的实例程序介绍，使读者能够在轻松愉快之中，对计算机图形编程应用的基本知识有一个完整的了解。更重要的是，使读者能够掌握一些具有很强的技巧性、同时又有很强的代表性和针对性的实用编程知识。以实用性为主兼顾先进性是本书的主要特点。具体的体现是，这些实例程序都是精心挑选的，具有相当的趣味性（很多例子实际上就是一个小游戏程序）、技巧性（有些算法实现起来需要有一定的技巧，尤其对于初学者来说更是如此）、先进性（算法和应用领域体现很强的专业性，比如区域填充、边沿跟踪或地图着色、象素生成算法、矢量字体的艺术变形、模式识别、分数维模型等）、实用性（跟通常应用紧密结合，并考虑应用效果和应用性能，每个实例程序实际上都是一个典型的应用程序）和代表性（不面面俱到，但充分体现完整性）。

全书由 8 章组成。每章以一个可独立运行的应用程序来综合体现图形应用某些方面的知识和编程技术。全书各章一致地按照“先原理后实现、精原理重实现”的结构形式来组织内容的叙述，即在先精炼介绍一般知识和算法原理的同时，着重叙述程序的实现细节，并给出注释详细的完整代码与应用结果。

各章的主要内容如下：

第 1 章系统地介绍进行图形编程所需要具备的颜色知识以及 Windows 图形编程方面的基本知识。在此基础上，实现了一个可以对显示位图进行任意着色或者颜色分析的演示功能程序。

第 2 章介绍实现彩色图形动态显示的基本编程技术。其中，特别介绍了一种可以对任意封闭区域进行象素遍历与填充的简便算法。然后，实现了一个模拟 54 张扑克的发牌、理牌与收牌动画过程的实例程序。

第 3 章以 Brehensam 直线象素生成算法为中心，介绍了象素生成方面的知识。作为象素生成知识的应用，本章实现了一个可以按书写过程动态显示手写字体的应用程序。

第 4 章对计算机图形变换及其应用方面的基本知识进行了全面介绍。在 Windows 环境下应用 OpenGL 图形功能，实现了一个可以创建诸如立体字、斜体字、圆形字、纹理字、光照字等几种典型形式的艺术字及字幕效果的应用程序。

第 5 章着重介绍图像象素放大方面的知识。利用这些知识，实现了一个可以对显示位图的任何形状区域进行任意倍数放大的放大镜工具程序。此外，实例程序还实现了图像漫游显示的功能。

第 6 章介绍图像变换与处理方面的基本知识。以此为基础，实现了一个能够对显示位图进行辉度或者对比度调整、单色化、二值化、柔化、马赛克晶格化、纹波扰动以及找边等 8 种图像处理功能的应用程序。

第 7 章介绍数字地图方面的应用知识。其中着重介绍了地图着色、目标跟踪和模式识别等方面的实用技能。然后，利用这些知识，实现了一个模拟排暴车在某辖区内执行巡逻任务、跟踪与发现目标以及指挥调度排暴作业等功能的实例程序。

第 8 章介绍了生成自然景物的分数维模型方面的知识。作为对这些知识的应用，本章实现了一个可以生成山峦、焰火、树与草等自然景物的应用程序。

由于时间仓促，加之作者的知识和技能有限，本书也会存在一些不足，恳请读者不吝赐教，批评指正。本书作者的电子邮件为 tanmj2_72_10@sina.com，本书责任编辑的电子邮件为 wangwenjuan@ptpress.com.cn。

作者

2002 年 4 月

目　录

第 1 章　颜色与绘图模式 1
1.1　颜色的基本知识 1
1.1.1　颜色的概念 2
1.1.2　颜色管理 5
1.1.3　调色板 5
1.2　绘图编程应用 6
1.2.1　设备上下文 6
1.2.2　绘图模式 6
1.2.3　坐标模式 6
1.2.4　Windows 绘图的基本过程 7
1.3　实例程序的功能与逻辑 7
1.3.1　程序功能 7
1.3.2　绘图模式的颜色算法 8
1.3.3　位图的读取与显示 8
1.3.4　程序逻辑结构及主要函数 11
1.4　程序结果与代码 11
1.4.1　运行环境说明 12
1.4.2　程序操作与显示 12
1.4.3　程序代码清单 13

第 2 章　画面动态显示编程 45
2.1　画面的动态显示 45
2.1.1　画面动态显示的基本原理 46
2.1.2　区域像素遍历与填充 46
2.2　实例程序的分析与设计 47
2.2.1　程序的图形显示 47
2.2.2　背景擦除与重现 49
2.3　实例程序的数据结构、逻辑与函数 49
2.3.1　自定义数据类型 50
2.3.2　函数功能逻辑与实现 52
2.4　程序运行与结果 55
2.4.1　总体结构 55
2.4.2　运行与结果 56
2.4.3　程序清单 58

第 3 章　手写字体的书写显示 115
3.1　Bresenham 直线像素生成算法 115
3.1.1　图形的像素表示与生成 115
3.1.2　Bresenham 直线生成算法 116
3.2　手写字体的笔划描述 117
3.2.1　用直线段表示手写字体的笔划 117
3.2.2　手写字体的书写特性及定义 118
3.2.3　针对整个手写字符的笔划表示 118
3.3　动态展示手写字体书写过程的综合应用程序 119
3.3.1　笔划录制 119
3.3.2　手写字体逐笔划显示 120
3.3.3　录制数据的显示 121
3.3.4　手写字符的提供 121
3.4　程序的运行与代码 121
3.4.1　程序运行与操作 121
3.4.2　程序代码清单 123

第 4 章　艺术字体及字幕实现 153
4.1　图形变换 154
4.1.1　计算机图形变换的一般过程 154
4.1.2　齐次坐标与变换矩阵 155
4.1.3　模型变换 156
4.1.4　视图变换 157
4.1.5　投影变换 158
4.1.6　局部坐标系与全局坐标系 160
4.1.7　视区变换 160
4.2　图形动画显示技术 161
4.2.1　图形动画显示原理与性能要求 161
4.2.2　提高动画显示性能的技术 162
4.3　在 Windows 编程中使用 OpenGL 164
4.3.1　OpenGL 与 Windows 设备上下文 164
4.3.2　在 Windows 编程中使用 OpenGL 的一般过程 165
4.4　实现艺术字与字幕生成功能的实例程序 166
4.4.1　艺术字体与字幕的实现思路 166
4.4.2　主要逻辑及其函数 166
4.5　程序的运行与代码 169
4.5.1　运行与操作 170
4.5.2　程序代码清单 173

第 5 章　图像像素放大 215
5.1　像素放大原理 215
5.1.1　像素放大概念 216
5.1.2　像素放大的基本实现 216
5.1.3　通过逆向映射实现像素放大 217
5.2　封闭区域的绘制与遍历 218
5.2.1　圆的 Brehensam 像素生成算法 218
5.2.2　指定像素放大的区域 219
5.3　放大镜与漫游工具实例程序 221
5.3.1　实例程序的总体框架结构 221
5.3.2　数据结构与变量 222
5.3.3　主要函数及其逻辑 223
5.4　程序的运行与代码 225
5.4.1　运行与操作 225
5.4.2　程序代码清单 227

第 6 章　图像处理 295
6.1　图像处理 295
6.1.1　图像像素表示 296
6.1.2　图像变换 297
6.2　具有典型图像处理功能的综合实例程序 300
6.2.1　程序的总体逻辑 300
6.2.2　几种图像处理效果的实现逻辑及函数 300
6.3　实例程序的运行与代码 302
6.3.1　实例程序的运行与操作 303
6.3.2　程序代码清单 306

第 7 章　数字地图 345
7.1　数字地图 345
7.1.1　数字地图信息的描述 346
7.1.2　数字地图着色 348
7.2　某辖区内执行巡逻与排暴作业的模拟程序 349
7.2.1　实例程序总体结构 349
7.2.2　主要逻辑及其相关函数与数据变量 349
7.3　程序的运行与代码 352
7.3.1　运行与操作 352
7.3.2　程序代码清单 353

第 8 章　自然景物生成 405
8.1　分数维 405

8.1.1 分数维概念 405
8.1.2 分数维造型 407
8.2 生成山峦、焰火、树与草等自然景物的实例程序 409
8.2.1 程序的总体逻辑 409
8.2.2 实例程序的逻辑、数据结构及函数 410
8.3 实例程序的运行与代码 412
8.3.1 程序运行与操作 412
8.3.2 实例程序代码清单 414

第 1 章　颜色与绘图模式

本章重点：

- 颜色的基本知识
- Windows 图形编程知识
- 各种绘图模式及其效果
- 位图的读取、加载与显示
- 颜色综合应用实例（颜色变换）

主要内容：

本章介绍有关颜色的概念及其编程应用方面的知识。

颜色的一般性知识包括颜色概念、三原色原理、颜色空间、颜色转换、颜色匹配、颜色管理和调色板等内容。这些知识对于在编程中有效地应用颜色是必不可少的，但同时，这部分内容往往又容易被忽视。通过本章的学习，读者将能够比较全面地了解颜色的基本知识，为后续颜色的使用打下基础。

为了满足应用程序的不同着色需要（比如动态改变颜色和混合颜色），Windows 编程系统一般都提供不同的绘图模式（着色模式）。程序员可以通过简单的函数调用设置并选择需要的模式，在对应模式下使用颜色。

尽管绘图模式从概念上讲是明确的，但理解起来并不十分直观。要在编程应用中使用颜色，选用合理的颜色模式是非常重要的，它是有效使用颜色的前提。其次，还应该熟悉颜色的种类及其分解与合成。

本章的实例程序为附盘中 Chapter1 文件夹下的 color.exe 文件，该实例从多方面综合模拟了 Windows 全部绘图模式的内容及效果，提供了一个高效的学习 Windows 颜色应用的环境。同时，为初学者理解、感知和体会颜色提供了一个理想的工具。

此外，程序还介绍了各种来源的位图的存取与显示，以及其他关于 Windows 编程方面的知识和技能。

1.1　颜色的基本知识

即使不是作为从事计算机成像或者印刷工作的专业人员，而只是作为普通的应用程序开发人员，颜色方面的知识也是必不可少的。在应用中使用颜色，已经成为现代程序开发的基本要素和必备技能。比如，几乎所有 Windows 应用程序都少不了提供工具栏按钮。尽管这些工具栏按钮图标看上去比较简单，但要设计得很友好也并非易事。

当然，对普通应用程序开发人员来说，使用颜色并不要求具备物理学家那样多的光学知识，也不需要具有画家处理颜色的技巧，而只要具有相关基本知识就能指导实际应用。

下面介绍这些基本知识。

1.1.1 颜色的概念

所谓颜色，实际上是可见光范围内的一种或几种波长的光波被视觉系统感知而在头脑中形成的一种概念。

1. 三原色 RGB

人类可以清楚分辨的颜色多达百万种。乍看起来，要对这么多的颜色逐个进行描述似乎是困难的。然而，研究表明，各种颜色都可以用红（R）、绿（G）、蓝（B）这三种最基本的颜色进行定量描述。由此形成的一套理论称为三原色原理，并相应地将 RGB 三种颜色称为三原色，R、G、B 分别表示构成颜色的一个分量。

这种认识来源于对人类视觉形成过程的研究。人们发现，颜色主要是由视网膜上称为锥状体的三种不同的感光细胞感知不同波长的光而形成的。这三种锥状体分别感知红色、绿色或者蓝色（的光）。所有其他颜色则是这三种颜色的不同混合效果。比如，三种锥状体接收到等量的红色、绿色或蓝色的光，则形成白色的视觉效果；三种锥状体没有接收到任何红色、绿色或蓝色的光，则产生黑色（即无光）的视觉。

三原色原理的发现，至少从理论上简化了颜色的描述和生成，因为它将各种各样的颜色都归结为三种基本颜色的数量关系。

在实际应用中，也可以根据需要，通过 RGB 的变换形式而不是直接的 RGB 形式来描述颜色。比如，针对不同的起点，颜色描述可以分为叠加原色法和相减原色法两种基本方法。对颜色描述的具体解释涉及到颜色空间的知识。

2. 颜色空间

颜色空间也叫颜色模型，它指的是通过基本颜色分量来定义其他各种颜色的模型结构。之所以将颜色模型称为颜色空间，是因为人们发现用笛卡尔空间坐标的形式来映射颜色模型往往显得更加直观、有效，颜色空间由此得名。

常用的颜色空间有 RGB 颜色空间（如图 1.1 所示）和 CMY 颜色空间（如图 1.2 所示）等类型。前者多用于显示器，后者多用于打印机。

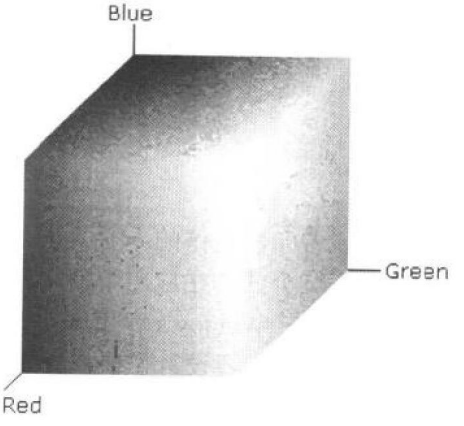


图 1.1 RGB 颜色空间

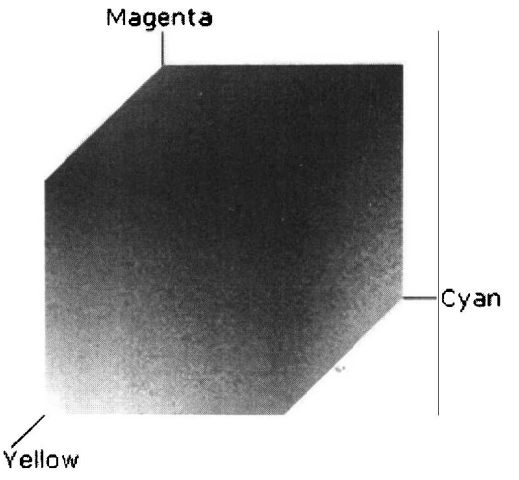


图 1.2　CMY 颜色空间

从图中可以看出，这两种颜色空间将所有颜色值映射到一个单位立方体中，即将颜色值进行归一化处理。这样，任意颜色值都可以由 3 个取值在 0 到 1 之间的颜色分量来表示。取值为 0 表示最小的颜色分量，而 1 则表示最大的颜色分量。比如，在 RGB 空间中（0，0，0）表示黑色，而（1，1，1）表示白色。当然，实际表示颜色时，并不一定要用小数来描述颜色，比如在 Windows 的编程中，通常就用一个整数来表示颜色分量。对应于一个像素，用来表示颜色分量的二进制位数决定了某个颜色的灰度等级。比如，用 8 位表示一个颜色分量，可以将该分量分成 256 个灰度等级，0 对应于颜色空间的 0，而 255 则对应于颜色空间的 1。

尽管 RGB 颜色空间和 CMY 颜色空间看起来有些差异，实际上它们之间并没有根本的不同，只是从不同的假定来看待颜色的起点而已。

在 RGB 颜色空间中，三个颜色分量都为 0 的颜色值即黑色（无光）在坐标原点，以（0，0，0）为起点，其他颜色都可以看作是在原点即黑色的基础上叠加三种颜色分量形成的。这种形成颜色的方法称为叠加原色法，相应 RGB 三原色称为叠加三原色。

在 CMY 颜色空间中，坐标原点（0，0，0）表示的颜色是白色——这个各颜色分量都取最大值的颜色。其他颜色仍然是在原点叠加颜色分量得到的，不过，经过叠加得到的颜色值的 RGB 分量都较白色要小，当达到坐标（1，1，1）点即各分量最大时，代表的颜色值——黑色实际上是无光的。这是以白色为起点生成颜色的方法，在 RGB 颜色分量逐渐减少的意义上将其称为相减原色法，同时将该颜色空间中的颜色分量 CMY 称为相减三原色。在这里，所有颜色都被看成是在白色基础上减去颜色分量而得到的。

除了 RGB 和 CMY 颜色空间以外，出于不同的应用需要，人们又发明了 HSV、HLS 等颜色空间形式。在 HSV 颜色空间和 HLS 颜色空间中，各种颜色一起映射为一种锥体的形式，这两种空间主要在艺术领域中使用。

3. 设备相关颜色空间

任何设备在实际生成颜色时，都是以白色（对应颜色空间中所能生成的最白的颜色）为基准的，通常将白色在颜色空间中的坐标点称为白点。不同颜色空间的白点是不同的；

即使采用相同的颜色空间，不同设备在具体实现颜色时，白点也会存在差异。也就是说，不同设备所实现的颜色空间是不同的，具体情况跟特定设备密切相关。据此，将跟特定设备密切相关的颜色空间称为设备相关颜色空间。用设备相关颜色空间表示的图像，很难找到一种将它从一种设备移植到另一种设备上的标准的方法。

4. 设备无关颜色空间

为了满足颜色在不同设备之间移植时标准化处理的应用需要，CIE 国际组织虚构了一种在实际生成颜色时使用的颜色空间 CIE XYZ。在这种颜色空间中描述的颜色与具体设备是不相关的，从而将这种颜色空间称为设备无关颜色空间。

不过，由于 CIE XYZ 数据量大，在 Internet 上传输图像要占用大量的带宽，这使其应用范围受到限制。目前，由 Microsoft 公司和 Hewlett-Packard 公司联合推荐的 sRGB 颜色空间，可望成为新的有效的设备无关颜色空间。

前面介绍的 RGB、CMY、HSV 和 HLB 等颜色空间，都可以是设备相关的，也可以是设备无关的。

5. 设备色阶

在计算机颜色应用中，人们常常为这样的情形所困扰：在一台计算机上实现的图像放到另一台计算机上显示时，显示效果会发生变化；特别是，在计算机屏幕上满足要求的显示效果，打印出来以后，效果明显受损。这个问题涉及到设备色阶的概念。

不同的设备所采用的颜色空间可能是不同的，并且在理论值和实际值之间存在差异，需要进行适当的补偿。比如，打印机实现的并不是 CMY 空间中的三个颜色分量，而是 CMYK 四个分量，其中 K 代表黑色，用于在实际应用时填补理论和实际之间的空隙。

即使选用的颜色空间相同，设备实现的白点也会存在差异，因此不同的设备实际显示的颜色值及其范围都是不一样的。

此外，完全一样的颜色在不同的设备上也可能使人产生不同的色觉。

这些与设备相关的因素所产生的综合效果，形成了特定设备所能应用的实际颜色值及其范围，即所谓的设备的色阶或者色域。

6. 颜色转换

对于同种真实的颜色，映射在不同颜色空间所得到的颜色值的描述形式是不同的。为了在某个特定设备（目的设备）中使用在另一个设备（源设备）中描述的图像，就需要将颜色值从源设备的颜色空间映射到目的设备的颜色空间中来，这个处理过程称为颜色转换。颜色转换是在不同颜色空间之间完成的。

颜色转换主要完成两方面的任务，一是将一个颜色空间中的白点映射到另一个颜色空间的白点；二是在映射白点的基础上，确保各种颜色与白点的相对位置关系符合新的颜色空间的结构。

7. 颜色匹配

所谓颜色匹配，指的是经过颜色转换而映射到新的颜色空间后，在目的设备中用与图像中各个颜色值最接近的颜色匹配源图像颜色的过程。

颜色匹配包括颜色的矫正和补偿。颜色匹配是在同种颜色空间中不同设备的色阶之间进行的。

1.1.2　颜色管理

在实际应用当中，颜色转换和颜色匹配总是需要的。从前面的介绍可以知道，进行颜色转换和颜色匹配并不是一件容易的事情。将这样的处理任务让普通开发人员来完成，显然是不合理的，这样做不仅对应用人员提出了更高的要求，而且执行效率也不高。

基于应用的实际需要，一些开发商提供了对颜色进行管理的工具软件，用于完成颜色应用所需要的转换和匹配任务。像微软公司提供的 ICM（Imaging Color Management）1.0 系统，就是一种很有效的颜色管理系统工具。ICM 的一些功能已经集成到许多 Windows 编程开发环境中了。

1.1.3　调色板

调色板是一个表示能够在输出设备上显示或者绘制的颜色的 RGB 值矩阵。调色板被那些能够生成许多颜色，但在给定时刻仅能显示或者绘制其中的部分颜色的设备所使用。调色板在系统和应用程序之间提供了一个映射颜色的接口，使应用程序和系统之间通过接口而不是直接和系统进行通信。

调色板分为系统调色板和逻辑调色板。系统调色板是硬件系统使用的调色板，其中包括了在某个给定时刻，系统所能显示和绘制的颜色值。而逻辑调色板指的是，某个应用程序用来存放自己所需要的颜色信息的矩阵。

有了调色板，应用程序就可以按需要使用“自己的”颜色，同时又不与使用不同颜色的其他应用程序相互干扰。某个应用程序在需要显示某些颜色时，只要将颜色填充到逻辑调色板中，然后，将逻辑调色板中的颜色信息映射到系统调色板，就可以得到所需要的颜色了。即使此时其他应用程序使用不同的颜色，但由于一个应用程序并不改变另一个应用程序的逻辑调色板中的颜色，所以，在需要的时候另一个应用程序仍然可以通过将逻辑调色板中的信息映射到系统调色板中得到，并不会因别的应用程序占用系统调色板而破坏应用程序的调色板信息。

系统能够提供多个逻辑调色板，但在任何时刻只能提供一个系统调色板。Windows 系统中，调色板的典型容量为 256，即调色板中能够包含 256 个不同的颜色信息。系统调色板中，有些颜色信息是可以改变的，有些是不可以改变的。不可以改变的颜色有 20 个，称为静止颜色，它们是系统保留使用的。可以改变的颜色约有 236 个，作为应用程序映射逻辑调色板之用。在这样的系统中，应用程序一次只能改变 236 种颜色。

当然，调色板的容量也可以不只是 256 而是其他的值，这由硬件系统决定。对于计算机来说，这里的硬件系统指的是显示适配器，即通常所说的显卡，显示适配器决定了设备的实际颜色数量，即色阶。

此外，对许多新的显卡如真彩色显卡而言，由于可以一次显示硬件所能生成的全部颜色，在这样的系统中，应用程序不用逻辑调色板就可以得到所需要的颜色信息。不过，为了保持向后兼容，以及为了满足应用程序通过调色板而不是直接存放颜色信息的需要，系统仍然提供调色板接口机制。

1.2 绘图编程应用

前面介绍的颜色的基本知识，可以很好地指导实际的编程工作。当然，要在实际编程中使用颜色，还需要了解编程开发环境和运行环境对颜色的支持情况，只有这样才能真正实现颜色的应用。这里介绍典型的 Windows 操作系统所支持的颜色应用开发与运行环境的情况。

1.2.1 设备上下文

在 Windows 环境中，系统提供了一种设备上下文机制，使应用人员可以通过设备上下文实现与绘图相关的编程需要。

所谓设备上下文，实质上是指一个存放诸如显示器或者打印机这样的绘图设备信息的 Windows 数据结构。所有的绘图功能调用都必须通过一个封装了画线、形体和字符等 Windows API 函数的设备上下文对象来进行。设备上下文可以将内容绘制到显示器、打印机或者元文件中。

设备上下文对象要占用较多的内存等系统资源，Windows 系统一次能够提供的设备上下文对象的数目是有限的，因此，需要考虑在编程中合理使用设备上下文。应用程序在进行绘制操作之前，需要首先获得设备上下文对象，并在不再需要时立即进行释放。

1.2.2 绘图模式

绘图模式指的是系统利用当前画笔的颜色和屏幕显示的颜色进行混合以得到新的显示颜色的方式。绘图模式决定了颜色混合的算法。比如，异或画笔绘图模式是将画笔的颜色值按位与屏幕上像素的当前颜色进行异或计算，得到新的像素显示颜色。而复制画笔绘图模式，则是用画笔的颜色覆盖屏幕像素颜色，即直接用画笔颜色作为像素新的显示颜色。

实际上，系统通过绘图模式机制提供了一种动态显示彩色信息的手段。比如，对于异或模式，当在同一个位置多次显示一个图形时，显示奇数次时图形出现，否则图形消失，这样就得到了显示和擦除图像的动态效果。当然，由于彩色包含比较丰富的颜色信息，绘图模式的选用需要具有一定的技巧。

1.2.3 坐标模式

应用中的所有绘制内容，都需要坐标值来表达位置信息。为了达到合适地体现所绘制内容的比例和形状的目的，需要在实际绘制之前，利用系统提供的函数合理设置或者映射当前程序所处的坐标系统模式。坐标模式决定了坐标的方向、比例和单位。

一般的 Windows 应用开发环境都提供多种坐标模式及其操作函数。设置坐标模式的工作非常容易完成，这里就不再赘述。

1.2.4　Windows 绘图的基本过程

根据以上的叙述，可以得到如下 Windows 程序绘图的基本步骤：

- 设置坐标模式（可以使用缺省模式）。
- 获取设备上下文。
- 建立并实现逻辑调色板（如不需要应用程序专用的调色板则可以省去这一环节）。
- 设置绘图模式（可以使用缺省模式）。
- 选用画笔（或者背景等）颜色。
- 调用设备上下文的绘图函数按坐标信息绘制具体内容。

1.3　实例程序的功能与逻辑

本章实现的实例程序请运行所附光盘中 chapter1 文件夹下的 color.exe 文件，这是一个综合性的颜色应用程序，包含的功能比较多，但主要体现在代码编写细节，从逻辑上讲并不太复杂，许多功能本身就很清楚地体现了实现逻辑。在这里，对实例程序中的功能和实现逻辑进行必要说明。

1.3.1　程序功能

实例程序通过对一定的图像像素施加不同的运算并进行显示，来体现绘图模式与颜色使用方面的知识和技能。具体功能如下。

位图显示。实例程序可以使用默认位图、打开任意位图文件或者让程序自动生成色彩连续变化的位图（将 1670 万种颜色分成 256 块进行显示），作为着色操作的背景素材，以得到整体的和可比较的处理效果。

运算结果的文字显示。除了位图以外，实例程序还根据当前选择的绘图模式，将对应的算法用文字的形式进行描述，以对照了解绘图模式的具体含义。

绘图模式的色彩示例显示。实例程序将当前给定的屏幕颜色（某一像素）、画笔颜色以及通过运算将得到的颜色，以彩色块的形式单独显示，得到具体的可比较的颜色。

色彩信息的数字显示。实例程序将当前给定的屏幕颜色（某一像素）、画笔颜色以及通过运算得到的颜色，分解成 RGB 三个分量并对应显示，以了解对应模式下的实际运算过程。

绘制模式选择。绘图模式可以通过一个下拉组合框来设置。

颜色的多种指定方式。可以通过数字或颜色对话框的方式为画笔指定颜色。同时，可以通过鼠标从当前显示的位图上拾取颜色，或者通过数字或者颜色对话框来指定参与运算的屏幕示例颜色。

整个程序设计成一个主窗口和一个非模态对话框的界面布局。采用这种界面布局的目的是，使设置的多项内容可以直接在多个方面体现出来。要使设置项在位图上体现出来，需要进行更新显示，其操作界面如图 1.3 所示。

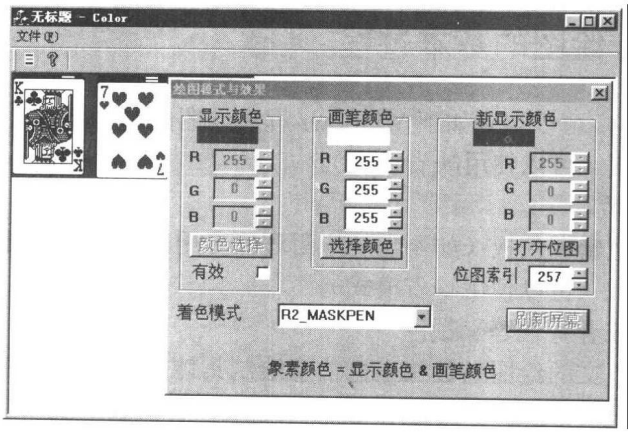


图 1.3 操作界面

1.3.2 绘图模式的颜色算法

本实例程序的主要功能就是模拟各种绘图模式的处理效果。Windows 95/NT 4.0 及其以后版本提供了 16 种绘图模式。Visual C++ 6.0 封装的 CClientDC 设备上下文类的 SetROP2() 成员函数，可以通过不同的参数来设置应用程序的各种绘图模式。设置了绘图模式之后，后续的绘制功能调用都将在该模式下起作用。具体地说，后续绘制操作将涉及到的像素逐个根据绘图模式实现的算法，用像素当前显示颜色和当前画笔颜色值算出该像素新的显示颜色值并显示之，直到重新设置新绘图模式为止。绘图模式对后续操作没有涉及到的像素没有影响。

在实际应用中，像素颜色值的计算与显示都是由系统根据当前模式自动完成的。实例程序在实现各种绘制模式的逻辑时，并不实际通过设置绘制模式来体现，而是根据绘图模式的颜色计算算法体现，逐个算出像素的新颜色值并将这种颜色直接显示出来（这可以通过将当前模式设置为画笔拷贝绘图模式来实现）。

关于 16 种绘图模式的像素颜色计算方法的具体内容，读者可以查看相关书籍，Visual C++ 6.0 的 MSDN 在线帮助系统，对这方面的内容有详细说明。这些内容是非常简单的，这里就不再罗列。实例程序在 ModeArith()函数中通过一条 switch 语句，详细实现了各种绘图模式的像素颜色计算逻辑，并相应进行显示，以得到实际的图像处理综合效果。在实例程序的操作中，读者可以通过一个下拉组合框来选定各种绘图模式对应的逻辑。

1.3.3 位图的读取与显示

位图是 Windows 系统支持的标准图像文件，本实例程序也采用位图作为着色的素材。实例程序可以显示三种不同来源的位图，从而为着色提供各种素材。一种位图是随程序打包的位图资源，可以通过位图加载而作为默认位图进行显示。另一种位图是位图文件，这可以通过文件读取并显示之。还有一种位图，是由程序按照一定的算法即时生成的。下面分别介绍这三种位图在实例程序中的使用情况。