

计算机三维图形学 与三维视觉

李伟生 胡晓玉 著

原理与应用

三维物体与场景表示
三维变换
单眼与双眼视系统
单图像立体视觉原理
BMP图像操作
内存管理

完整魔画生成源程序

武汉测绘科技大学出版社

计算机 三维图形学与三维视觉

□ 李伟生 胡晓玉 著

武汉测绘科技大学出版社

(鄂)新登字 14 号

图书在版编目(CIP)数据

计算机三维图形学与三维视觉/李伟生,胡晓玉著.

武汉:武汉测绘科技大学出版社,1997.7

ISBN 7-81030-549-2

I. 计…

II. 李…

III. 计算机-制图

IV. TP302.4

责任编辑:郭毅 封面设计:曾兵

武汉测绘科技大学出版社出版发行
武汉测绘科技大学出版社印刷厂印刷

※

787×1092 1/16 11 印张 4 彩页 282 千字
1997 年 7 月第 1 版 1997 年 7 月第 1 次印刷

印 数:1~2000 册 定 价:18.00 元

目 录

| | |
|--------------------------|-------|
| 引 言 | (1) |
| 第一章 预备工作 | (3) |
| 1.1 图形显示模式设置 | (3) |
| 1.2 画点与读点 | (3) |
| 1.3 设置调色板 | (4) |
| 1.4 图形基本设置的程序 | (4) |
| 1.5 数学基础程序 | (16) |
| 第二章 图形学几何基础 | (30) |
| 2.1 坐标系 | (30) |
| 2.2 仿射变换 | (31) |
| 2.3 实现仿射变换 | (35) |
| 2.4 点集变换 | (41) |
| 2.5 视系统——单视点系统 | (41) |
| 2.6 计算机图形学三维处理基本程序 | (46) |
| 第三章 物体表示与场景表示 | (54) |
| 3.1 物体表示——外存结构 | (54) |
| 3.2 物体表示——内存表示 | (55) |
| 3.3 构造物体 | (59) |
| 3.4 构造物体的例子 | (62) |
| 3.5 场景表示 | (89) |
| 3.6 场景的处理过程 | (91) |
| 3.7 显示场景的框架图 | (97) |
| 第四章 双视点视系统 | (100) |
| 4.1 立体视觉 | (100) |
| 4.2 立体视觉系统 | (104) |
| 4.3 匹配原理 | (105) |
| 4.4 隐藏点 | (106) |
| 第五章 立体图制作 | (108) |
| 5.1 准备大缓冲区——利用 EMS 和 XMS | (108) |
| 5.2 读写 BMP 文件 | (123) |
| 5.3 空间多边形填充 | (130) |
| 5.4 Z 缓冲区 | (136) |
| 5.5 实现匹配 | (138) |

| | | |
|------|-------------|-------|
| 5.6 | 底图初始化 | (143) |
| 5.7 | 主程序 | (146) |
| 5.8 | 编译说明 | (157) |
| 附录 A | 源程序说明 | (159) |
| 附录 B | 场景文件例 | (162) |
| 附录 C | 例图 | (172) |

引 言

蓝色的大海、绿色的山峦、白色的云纱……如此视野，令人愉悦。绘画、雕塑、摄影……人类从没有放弃用视觉去表达心灵。当一篇美妙的散文慢慢流入脑海时，我们的思绪顺着那语言链起伏；当绝妙的视觉艺术映入眼帘时，那全方位的心灵震撼常常令人无以言状。一维的语言难以表达三维的世界，离散的语言无法表达连续的世界。无法用语言替代的视觉表达，真奇，真妙。

计算机一冒出这个世界，就一直有人试着用它满足一下视觉的需求。似乎所有学习计算机编程语言的人，都会挂念这样的问题：怎样用它画出图来。世界就是这样的缠绵，你越想得到的，它就越不让你轻易的到手。我们没法像敲出一个“a”那样，马上在屏幕上敲出一个点。

事实是这样的，在不同的计算机平台上，不同的操作系统上，即使使用同一种计算机语言，也得用不同的方式才能在屏幕上画出一个点。

即使对同一计算机平台，同一操作系统，不同开发商的同一语言，在处理图形操作上也常常不一样。

千差万别，问题的实质却没有差别。这个“实质”，就是“相应”的数学。

等等，我们先来圈定一个我们感兴趣的问题。否则，无法谈论“相应”。

计算机图形世界里，有许多令人振奋的话题。三维与动画则是这一两年的热门。与此有关的书籍比比皆是。

有的书籍讨论基于某一三维动画软件，如何制作成功的作品。这当然不是这本书的话题，也不是编程好手们的兴趣所在。

有的书籍详细讨论计算机图形学中的几何问题。虽然，把握住了问题的“实质”，但缺乏完整的范例，让初学者难以涉足。

有的书籍，详细地展示了某一应用方面的编程实例，但对“实质”问题却描写的含含糊糊。使得读者难于自由发挥其想象的天赋。

本书意在综合以上后两种书籍的优点，既全面的展示一个有趣的应用，又完整地介绍计算机图形学中的几何学基础。作者希望读了本书的读者，既能把握三维计算机图形学实质，又是处理实际问题的高手。

本书将全面展示的应用是：单图像立体图制作。单图像立体图仅用一副图展示三维世界的空间关系，不必佩戴任何眼镜，只要稍稍用点技巧就可让人发现一个令人吃惊的三维世界。本书选择这样一个专题，除了它可全面展示三维图形几何学外，还因为它本身所具有的无穷魅力。

本书介绍的计算机图形学的三维几何学包括物体表示、场景表示、坐标平移变换、旋转变换、缩放变换、投影变换、单视点视系统、双视点视系统、匹配表示等。这些知识几乎是所有与计算机图形操作有关的的应用的基础。

要构造一个实用的系统，除了理论基础外还需要一点技巧。本书介绍的与图形学有关的技巧有：VESA 图形卡使用；Bitmap 图像读、写、显示、写屏；扩展内存与扩充内存使用。

本书的实例建立在 IBM386 以上兼容机平台、DOS 操作系统之上。所有程序在 Borland C++3.1 之上编译成功。

本书各章是如下安排的：

第一章 预备工作 这一部分介绍初始化图形设备，在屏幕上描绘点、读取点的操作及后续工作所需的基本数学函数。

第二章 计算机图形学中的三维几何学 这一部分讨论坐标系、仿射变换、单视点视系统等计算机图形学的数学基础。

第三章 物体表示和场景表示 这一章讨论表示物体和场景的数学方法。这包括讨论如何表示单个物体、存取单个物体，如何将多个物体组合到一个场景中去，如何将一个场景显示到屏幕上。

第四章 双视点视系统 这一章讨论立体视觉理论中的问题，是本书的精华。讨论的问题包括立体视觉系统描述、匹配关系的表示、隐藏点的性质。

第五章 立体图像生成系统的实现 这一章讨论如何构造一个实际的立体图像制作系统。所涉及的技术包括空间多边形填充技术、Z-Buffer 技术、匹配描绘技术、内存使用技术、图像操作技术。

在本书的附录里，还有一些相关信息。并有一些立体图样本。

本书中部分程序（基本数学函数、某些物体构造过程）取自 Larry Sharp 的著作 Programming in 3D（当然，其中有些部分作了相应调整）。本书中关于扩展内存与扩充内存的程序是从 Internet 中获得的（当然，也作了一些改动）。

第一章 预备工作

本书全部实例都是基于 IBM PC 兼容机的。只要你使用的是 386 以上机型,且具有 4M 以上的内存,就能运行这些程序。本书的程序在 Borland C++3.1 环境下编译通过。

图形操作常受显示卡的限制。本书采用 1024 宽×768 高×256 色的模式显示图像。对于这个模式,不同的显示卡使用不同的设置方式。本书中,除了 Graphini.c 中的几个过程与设备有关外,其它的程序一概与显示设备无关。如果你使用的是一般的 SVGA,TVGA 卡,很容易找到一本介绍它们的书,参考它们编出替代 Graphini.c 的程序。如果你只想用通常的 VGA 显示特性,只须用 C 语言的库函数替代 Graphini.c 中的函数,并将程序中屏幕尺寸及图形尺寸的宏定义改一下就可以了。这里使用的是 CL GD542X VESA 显示卡。

处理图形免不了要使用数学计算。预备一些常见的数学函数对以后的工作很有好处。虽然 C 语言为我们准备了一些数学函数,似乎还满足不了我们的需要。

这一章就是为了上述两项基础而设的。

1.1 图形显示模式设置

屏幕可用多种显示模式工作。具体如何工作取决于模式设置。模式设置通常由中断函数调用来实现。下面的函数为 VESA 显示卡设置了 1024×768×256 的显示模式。

```
struct REGPACK rp;  
Void SetVesa256(Void)  
{  
rp.r_bx=0x105;  
rp.r_ax=0x4f02;  
intr(16,8rp);  
}
```

参数 0x4f02 看起来似乎有点奇怪。

过程 SetText()用以结束图形工作方式。程序可在 1.4 节找到。

1.2 画点与读点

在设置了图形模式后,就可以在屏幕上指定的位置上画上一个色彩点了。由于我们定义的模式有 256 个颜色,所以我们可以调用色彩序号 0~255 来表示色彩。

屏幕的定位坐标系是这样的:以屏幕左上角为原点,水平向右为 x 轴正向,垂直向下为 y 轴正向。横向坐标从 0 到 1023,纵向坐标从 0 到 767。

在屏幕上的某个位置画点,或在那个位置读点,实际上就是在显示卡内存的某个地址上写入色彩标号或读出色彩值。

这样,所需的工作就是定位、赋值、取值。

定位由 Locate()函数完成。这一过程的任务是对指定的坐标取得操作地址。事实上,在屏

幕上,每隔 32 行,地址值重复出现。所以,仅仅对外给一个操作地址是不够的。这个过程还调用 SetBank()过程,根据当前所处的第几个 32 行称作带,设置内存的页。SetBank()通过调用中断函数实现。它每次都要检查,当前所处“带”是否与要设置的“带”一样,如一样,则不做任何事情。

涂色用 Plot()函数完成,它只是在 Locate()调用所获得的地址上赋一个值。

读屏幕上的一个点的值由 Get-Pixel()函数完成。它只要简单地返回当前 Locate()所定位的值。

1.3 设置调色板

每一个色彩由红、绿、蓝三种色彩合成。这三种色彩的不同强度组成了各种不同的色彩组合。

为了表示色彩,定义数据结构 RGB。

```
typedef struct {
    Byte Red; //红色分量
    Byte Grn; //绿色分量
    Byte Blu; //蓝色分量
}RGB;
```

在我们设置的模式中,最多可以定义 256 种色彩组合。注意,每一色彩分量的取值从 0 到 256。将定义的色彩以一定顺序排在一起,依顺序编成 0~255 号。在给显示内存赋值时,就是赋的这个序号。

色彩的排序称作调色板。其数据结构定义为长度为 256 的 RGB 类型数组。

我们提供了两种调色板。Init_Palette()通过简单的循环赋值为调色板赋值,而 Init_Palette_2()则是对调色板的每一个色彩的每一个分量逐一给值。第二种调色板产生的方式虽然繁,但效果似乎更好。

将调色板设置到硬件环境中去,则是 Set_Palette()过程的工作。这个工作是通过调用中断函数完成的。

1.4 图形基本设置的程序

以上的各功能函数可以在 graphini.c 中找到。定义函数原形的文件是 graphini.h。头文件 gv.h 定义了几个几乎全书都要用到的全局变量。头文件 Pre.h 则定义了一些数据类型。

```
文件 gv.h
//File GV.h
/*
Here are some global variables for the system.
*/
#ifdef _GV_H
#define _GV_H

//The global variables

int XRes, YRes;
```

```

        Word      X_Off,Y_Off;
        float     Mx,My,Mz,D;
        int       Angl,Tilt;
        Byte far  * Address;

#endif

文件 pre.h
//File PRE.H
/*
This is for defining some basic data types,
and declare some extern variabes.
*/

#ifndef _PRE_H
#define _PRE_H

#define Ln10      2.30258509299405E+000
#define OneOverLn10 0.43429448190325E+000
#define Pi       3.1415927
#define PiOver180 1.74532925199433E-002
#define PiUnder180 5.72957795130823E+001

typedef unsigned char      Byte;
typedef unsigned int      Word;
typedef unsigned long     DWord;
typedef enum {false,true} Boolean;

typedef struct {
        Byte Red;
        Byte Grn;
        Byte Blu;
        }RGB; //color value

typedef RGB Palette_Register[256]; //color palette
typedef float VECT[3]; //3d vector with float value
typedef int VECTI[3]; //3d vector with int value
typedef float FDA[4]; //4d vector
typedef float Matrx4x4[4][4]; //4x4 matrix

extern int      XRes, YRes;
extern Word     X_Off,Y_Off;
extern float    Mx,My,Mz,D;
extern int      Angl,Tilt;
extern Byte far * Address;

#endif

```

文件 graphini. h

```
//File GRAPHINI. H
/*
This program is for the basic operators for driving the graphical device.
It is for VESA board. If you has a different graphical board on your maching
What you needed to is just rewrite graphini. h and graphini. c.
*/
#ifndef _GRAPHINI_H
#define _GRAPHINI_H

/* * * * * *
*       work swith vesa board       *
* * * * *
SetVesa256()    —init graphics as 1024x768x256
SetText()      —init text diaply mode
*/
extern void     SetVesa256(void);
extern void     SetText(void);

/* * * * * *
*       work with the palette       *
* * * * *
Init_Palette()    —init global color palette variable Color
Init_Palette_2() —the second way to init global color palette
                  variable Color
Set_Palette()     —the actual action to set color palette into
                  the service
Cycle_Palette()  —round shift the palette forward one index
*/

extern void     Set_Palette(Palette_Register Hue);
extern void     Init_Palette(Palette_Register Color);
extern void     Init_Palette_2(Palette_Register Color);
extern void     Cycle_Palette(Palette_Register Hue);

/* * * * * *
*       drawing and picking the color   *
* * * * *
Locate()         —locate the actual address for giving coordinates
                  in the displaying page
Set_Bank()       —the inner process for locating the address
                  in the displaying page
Plot()           —to paint the giving color in the giving coordinates
Get_Pixel()      —to get the color value in the giving pixel
*/
extern void     Locate(Word x,Word y);
```

```

extern void SetBank(Word);
extern void Plot(Word x, Word y, Byte color);
extern Byte Get_Pixel(Word x, Word y);

```

```
#endif
```

文件 graphini.c

```
//File GRAPHINI.C
```

```
/*
```

This program is for initialisation of the graphical device

This is the only part that is related with the device.

If you has other type of the graphical board, just change this part.

```
*/
```

```
#include <dos.h>
```

```
#include "pre.h"
```

```
#include "graphini.h"
```

```
union REGS reg;
```

```
struct SREGS inreg;
```

```
struct REGPACK rp;
```

```
Word oldbank;
```

```
void SetBank(Word bank)
```

```
    //bank --the using memory bank
```

```
{
```

```
    if(bank !=oldbank)
```

```
    {
```

```
        rp.r_bx=0;
```

```
        rp.r_dx=bank * 8;
```

```
        rp.r_ax=0x4f05;
```

```
        intr(16,&rp);
```

```
        oldbank=bank;
```

```
    }
```

```
}
```

```
void SetVesa256(void)
```

```
{
```

```
    rp.r_bx=0x105;
```

```
    rp.r_ax=0x4f02;
```

```
    intr(16,&rp);
```

```
}
```

```
void SetText(void)
```

```
{
```

```
    reg.h.ah=0;
```

```
    reg.h.al=3;
```

```
    int86(0x10,&reg,&reg);
```

```

}

void Locate(Word x, Word y)
    //(x,y)    -the coordinates to locate
    {
    long int l;
    l=(y%32) * 1024+x;
    SetBank(y>>5);
    Address=(char far *) (0xA0000000L +l); //0xA0000000L is the start
                                           //address of the drawing page
    }

```

```

void Plot(Word x, Word y, Byte color)
    //(x,y)    -the coordinates to draw
    //color    -the color to paint
    {
    y=y;
    x=x;
    Locate(x,y);
    * Address=color;
    }

```

```

static Palette _Register Color;

```

```

void Set _Palette(Palette _Register Hue)
    //Hue -the aolor palette to set
    {
    reg. x. ax=0x1012;
    reg. x. bx=0;
    reg. x. cx=256;
    reg. x. dx=FP _OFF(Hue);
    inreg. es=FP _SEG(Hue);
    int86x(0x10,&reg,&reg,&inreg);
    }

```

```

void Init _Palette(Palette _Register Color)
    //Color    -the palette to set
    {
    Word i;
    for(i=0;i<64;i++)
        {
        Color[i].Red=i;
        Color[i].Grn=i;
        Color[i].Blu=i;
        }
    for (i=64;i<128;i++)
        {
        Color[i].Red=i-64;

```

```

        Color[i].Grn=0;
        Color[i].Blu=0;
    }
    for(i=128;i<192;i++)
    {
        Color[i].Red=0;
        Color[i].Grn=i-128;
        Color[i].Blu=0;
    }
    for(i=192;i<=255;i++)
    {
        Color[i].Red=0;
        Color[i].Grn=0;
        Color[i].Blu=i-192;
    }
}

```

void Init_Palette_2(Palette_Register Color)

```

{
    Color[0].Red=0; Color[0].Grn=0; Color[0].Blu=0;
    Color[1].Red=63; Color[1].Grn=0; Color[1].Blu=0;
    Color[2].Red=0; Color[2].Grn=63; Color[2].Blu=0;
    Color[3].Red=63; Color[3].Grn=63; Color[3].Blu=0;
    Color[4].Red=0; Color[4].Grn=0; Color[4].Blu=63;
    Color[5].Red=63; Color[5].Grn=0; Color[5].Blu=63;
    Color[6].Red=0; Color[6].Grn=63; Color[6].Blu=63;
    Color[7].Red=63; Color[7].Grn=63; Color[7].Blu=63;
    Color[8].Red=58; Color[8].Grn=58; Color[8].Blu=58;
    Color[9].Red=51; Color[9].Grn=51; Color[9].Blu=51;
    Color[10].Red=44; Color[10].Grn=44; Color[10].Blu=44;
    Color[11].Red=38; Color[11].Grn=38; Color[11].Blu=38;
    Color[12].Red=31; Color[12].Grn=31; Color[12].Blu=31;
    Color[13].Red=25; Color[13].Grn=25; Color[13].Blu=25;
    Color[14].Red=19; Color[14].Grn=19; Color[14].Blu=19;
    Color[15].Red=6; Color[15].Grn=6; Color[15].Blu=6;
    Color[16].Red=61; Color[16].Grn=0; Color[16].Blu=0;
    Color[17].Red=63; Color[17].Grn=28; Color[17].Blu=0;
    Color[18].Red=63; Color[18].Grn=42; Color[18].Blu=0;
    Color[19].Red=63; Color[19].Grn=53; Color[19].Blu=0;
    Color[20].Red=61; Color[20].Grn=61; Color[20].Blu=0;
    Color[21].Red=53; Color[21].Grn=63; Color[21].Blu=0;
    Color[22].Red=42; Color[22].Grn=63; Color[22].Blu=0;
    Color[23].Red=28; Color[23].Grn=63; Color[23].Blu=0;
    Color[24].Red=0; Color[24].Grn=61; Color[24].Blu=0;
    Color[25].Red=0; Color[25].Grn=63; Color[25].Blu=28;
    Color[26].Red=0; Color[26].Grn=63; Color[26].Blu=42;
    Color[27].Red=0; Color[27].Grn=63; Color[27].Blu=53;
    Color[28].Red=0; Color[28].Grn=61; Color[28].Blu=61;
}

```

Color[29].Red=0; Color[29].Grn=53; Color[29].Blu=63;
Color[30].Red=0; Color[30].Grn=42; Color[30].Blu=63;
Color[31].Red=0; Color[31].Grn=28; Color[31].Blu=63;
Color[32].Red=0; Color[32].Grn=0; Color[32].Blu=61;
Color[33].Red=28; Color[33].Grn=0; Color[33].Blu=63;
Color[34].Red=42; Color[34].Grn=0; Color[34].Blu=63;
Color[35].Red=53; Color[35].Grn=0; Color[35].Blu=63;
Color[36].Red=61; Color[36].Grn=0; Color[36].Blu=61;
Color[37].Red=63; Color[37].Grn=0; Color[37].Blu=53;
Color[38].Red=63; Color[38].Grn=0; Color[38].Blu=42;
Color[39].Red=63; Color[39].Grn=0; Color[39].Blu=28;
Color[40].Red=57; Color[40].Grn=0; Color[40].Blu=0;
Color[41].Red=57; Color[41].Grn=26; Color[41].Blu=0;
Color[42].Red=57; Color[42].Grn=38; Color[42].Blu=0;
Color[43].Red=57; Color[43].Grn=48; Color[43].Blu=0;
Color[44].Red=57; Color[44].Grn=57; Color[44].Blu=0;
Color[45].Red=48; Color[45].Grn=57; Color[45].Blu=0;
Color[46].Red=38; Color[46].Grn=57; Color[46].Blu=0;
Color[47].Red=26; Color[47].Grn=57; Color[47].Blu=0;
Color[48].Red=0; Color[48].Grn=57; Color[48].Blu=0;
Color[49].Red=0; Color[49].Grn=57; Color[49].Blu=26;
Color[50].Red=0; Color[50].Grn=57; Color[50].Blu=38;
Color[51].Red=0; Color[51].Grn=57; Color[51].Blu=48;
Color[52].Red=0; Color[52].Grn=57; Color[52].Blu=57;
Color[53].Red=0; Color[53].Grn=48; Color[53].Blu=57;
Color[54].Red=0; Color[54].Grn=38; Color[54].Blu=57;
Color[55].Red=0; Color[55].Grn=26; Color[55].Blu=57;
Color[56].Red=0; Color[56].Grn=0; Color[56].Blu=57;
Color[57].Red=26; Color[57].Grn=0; Color[57].Blu=57;
Color[58].Red=38; Color[58].Grn=0; Color[58].Blu=57;
Color[59].Red=48; Color[59].Grn=0; Color[59].Blu=57;
Color[60].Red=57; Color[60].Grn=0; Color[60].Blu=57;
Color[61].Red=57; Color[61].Grn=0; Color[61].Blu=48;
Color[62].Red=57; Color[62].Grn=0; Color[62].Blu=38;
Color[63].Red=57; Color[63].Grn=0; Color[63].Blu=26;
Color[64].Red=51; Color[64].Grn=0; Color[64].Blu=0;
Color[65].Red=51; Color[65].Grn=24; Color[65].Blu=0;
Color[66].Red=51; Color[66].Grn=33; Color[66].Blu=0;
Color[67].Red=51; Color[67].Grn=43; Color[67].Blu=0;
Color[68].Red=51; Color[68].Grn=51; Color[68].Blu=0;
Color[69].Red=43; Color[69].Grn=51; Color[69].Blu=0;
Color[70].Red=34; Color[70].Grn=51; Color[70].Blu=0;
Color[71].Red=24; Color[71].Grn=51; Color[71].Blu=0;
Color[72].Red=0; Color[72].Grn=51; Color[72].Blu=0;
Color[73].Red=0; Color[73].Grn=51; Color[73].Blu=24;
Color[74].Red=0; Color[74].Grn=51; Color[74].Blu=34;
Color[75].Red=0; Color[75].Grn=51; Color[75].Blu=43;
Color[76].Red=0; Color[76].Grn=51; Color[76].Blu=51;

Color[77]. Red=0; Color[77]. Grn=43; Color[77]. Blu=51;
Color[78]. Red=0; Color[78]. Grn=34; Color[78]. Blu=51;
Color[79]. Red=0; Color[79]. Grn=24; Color[79]. Blu=51;
Color[80]. Red=0; Color[80]. Grn=0; Color[80]. Blu=51;
Color[81]. Red=24; Color[81]. Grn=0; Color[81]. Blu=51;
Color[82]. Red=34; Color[82]. Grn=0; Color[82]. Blu=51;
Color[83]. Red=43; Color[83]. Grn=0; Color[83]. Blu=51;
Color[84]. Red=51; Color[84]. Grn=0; Color[84]. Blu=51;
Color[85]. Red=51; Color[85]. Grn=0; Color[85]. Blu=43;
Color[86]. Red=51; Color[86]. Grn=0; Color[86]. Blu=34;
Color[87]. Red=51; Color[87]. Grn=0; Color[87]. Blu=24;
Color[88]. Red=42; Color[88]. Grn=0; Color[88]. Blu=0;
Color[89]. Red=42; Color[89]. Grn=20; Color[89]. Blu=0;
Color[90]. Red=42; Color[90]. Grn=28; Color[90]. Blu=0;
Color[91]. Red=42; Color[91]. Grn=35; Color[91]. Blu=0;
Color[92]. Red=42; Color[92]. Grn=42; Color[92]. Blu=0;
Color[93]. Red=35; Color[93]. Grn=42; Color[93]. Blu=0;
Color[94]. Red=28; Color[94]. Grn=42; Color[94]. Blu=0;
Color[95]. Red=20; Color[95]. Grn=42; Color[95]. Blu=0;
Color[96]. Red=0; Color[96]. Grn=42; Color[96]. Blu=0;
Color[97]. Red=0; Color[97]. Grn=42; Color[97]. Blu=20;
Color[98]. Red=0; Color[98]. Grn=42; Color[98]. Blu=28;
Color[99]. Red=0; Color[99]. Grn=42; Color[99]. Blu=35;
Color[100]. Red=0; Color[100]. Grn=42; Color[100]. Blu=42;
Color[101]. Red=0; Color[101]. Grn=35; Color[101]. Blu=42;
Color[102]. Red=0; Color[102]. Grn=28; Color[102]. Blu=42;
Color[103]. Red=0; Color[103]. Grn=20; Color[103]. Blu=42;
Color[104]. Red=0; Color[104]. Grn=0; Color[104]. Blu=42;
Color[105]. Red=20; Color[105]. Grn=0; Color[105]. Blu=42;
Color[106]. Red=28; Color[106]. Grn=0; Color[106]. Blu=42;
Color[107]. Red=35; Color[107]. Grn=0; Color[107]. Blu=42;
Color[108]. Red=42; Color[108]. Grn=0; Color[108]. Blu=42;
Color[109]. Red=42; Color[109]. Grn=0; Color[109]. Blu=35;
Color[110]. Red=42; Color[110]. Grn=0; Color[110]. Blu=28;
Color[111]. Red=42; Color[111]. Grn=0; Color[111]. Blu=20;
Color[112]. Red=31; Color[112]. Grn=0; Color[112]. Blu=0;
Color[113]. Red=31; Color[113]. Grn=16; Color[113]. Blu=0;
Color[114]. Red=31; Color[114]. Grn=22; Color[114]. Blu=0;
Color[115]. Red=31; Color[115]. Grn=27; Color[115]. Blu=0;
Color[116]. Red=31; Color[116]. Grn=31; Color[116]. Blu=0;
Color[117]. Red=27; Color[117]. Grn=31; Color[117]. Blu=0;
Color[118]. Red=22; Color[118]. Grn=31; Color[118]. Blu=0;
Color[119]. Red=16; Color[119]. Grn=31; Color[119]. Blu=0;
Color[120]. Red=0; Color[120]. Grn=31; Color[120]. Blu=0;
Color[121]. Red=0; Color[121]. Grn=31; Color[121]. Blu=16;
Color[122]. Red=0; Color[122]. Grn=31; Color[122]. Blu=22;
Color[123]. Red=0; Color[123]. Grn=31; Color[123]. Blu=27;
Color[124]. Red=0; Color[124]. Grn=31; Color[124]. Blu=31;

Color[125].Red=0; Color[125].Grn=27; Color[125].Blu=31;
Color[126].Red=0; Color[126].Grn=22; Color[126].Blu=31;
Color[127].Red=0; Color[127].Grn=16; Color[127].Blu=31;
Color[128].Red=0; Color[128].Grn=0; Color[128].Blu=31;
Color[129].Red=16; Color[129].Grn=0; Color[129].Blu=31;
Color[130].Red=22; Color[130].Grn=0; Color[130].Blu=31;
Color[131].Red=27; Color[131].Grn=0; Color[131].Blu=31;
Color[132].Red=31; Color[132].Grn=0; Color[132].Blu=31;
Color[133].Red=31; Color[133].Grn=0; Color[133].Blu=27;
Color[134].Red=31; Color[134].Grn=0; Color[134].Blu=22;
Color[135].Red=31; Color[135].Grn=0; Color[135].Blu=16;
Color[136].Red=63; Color[136].Grn=42; Color[136].Blu=42;
Color[137].Red=63; Color[137].Grn=48; Color[137].Blu=42;
Color[138].Red=63; Color[138].Grn=53; Color[138].Blu=42;
Color[139].Red=63; Color[139].Grn=58; Color[139].Blu=42;
Color[140].Red=63; Color[140].Grn=63; Color[140].Blu=42;
Color[141].Red=58; Color[141].Grn=63; Color[141].Blu=42;
Color[142].Red=53; Color[142].Grn=63; Color[142].Blu=42;
Color[143].Red=48; Color[143].Grn=63; Color[143].Blu=42;
Color[144].Red=42; Color[144].Grn=63; Color[144].Blu=42;
Color[145].Red=42; Color[145].Grn=63; Color[145].Blu=48;
Color[146].Red=42; Color[146].Grn=63; Color[146].Blu=53;
Color[147].Red=42; Color[147].Grn=63; Color[147].Blu=58;
Color[148].Red=42; Color[148].Grn=63; Color[148].Blu=63;
Color[149].Red=42; Color[149].Grn=58; Color[149].Blu=63;
Color[150].Red=42; Color[150].Grn=53; Color[150].Blu=63;
Color[151].Red=42; Color[151].Grn=48; Color[151].Blu=63;
Color[152].Red=42; Color[152].Grn=42; Color[152].Blu=63;
Color[153].Red=48; Color[153].Grn=42; Color[153].Blu=63;
Color[154].Red=53; Color[154].Grn=42; Color[154].Blu=63;
Color[155].Red=58; Color[155].Grn=42; Color[155].Blu=63;
Color[156].Red=63; Color[156].Grn=42; Color[156].Blu=63;
Color[157].Red=38; Color[157].Grn=42; Color[157].Blu=58;
Color[158].Red=63; Color[158].Grn=42; Color[158].Blu=53;
Color[159].Red=63; Color[159].Grn=42; Color[159].Blu=48;
Color[160].Red=57; Color[160].Grn=38; Color[160].Blu=38;
Color[161].Red=57; Color[161].Grn=43; Color[161].Blu=38;
Color[162].Red=57; Color[162].Grn=48; Color[162].Blu=38;
Color[163].Red=57; Color[163].Grn=53; Color[163].Blu=38;
Color[164].Red=57; Color[164].Grn=57; Color[164].Blu=38;
Color[165].Red=53; Color[165].Grn=57; Color[165].Blu=38;
Color[166].Red=48; Color[166].Grn=57; Color[166].Blu=38;
Color[167].Red=43; Color[167].Grn=57; Color[167].Blu=38;
Color[168].Red=38; Color[168].Grn=57; Color[168].Blu=38;
Color[169].Red=38; Color[169].Grn=57; Color[169].Blu=43;
Color[170].Red=38; Color[170].Grn=57; Color[170].Blu=48;
Color[171].Red=38; Color[171].Grn=57; Color[171].Blu=53;
Color[172].Red=38; Color[172].Grn=57; Color[172].Blu=57;