

第二章 计算机图形硬件及绘图系统

- 计算机图形系统
- 图形输入设备
- 图形显示设备
- 显示子系统
- 图形硬拷贝设备
- 图形标准及软件包

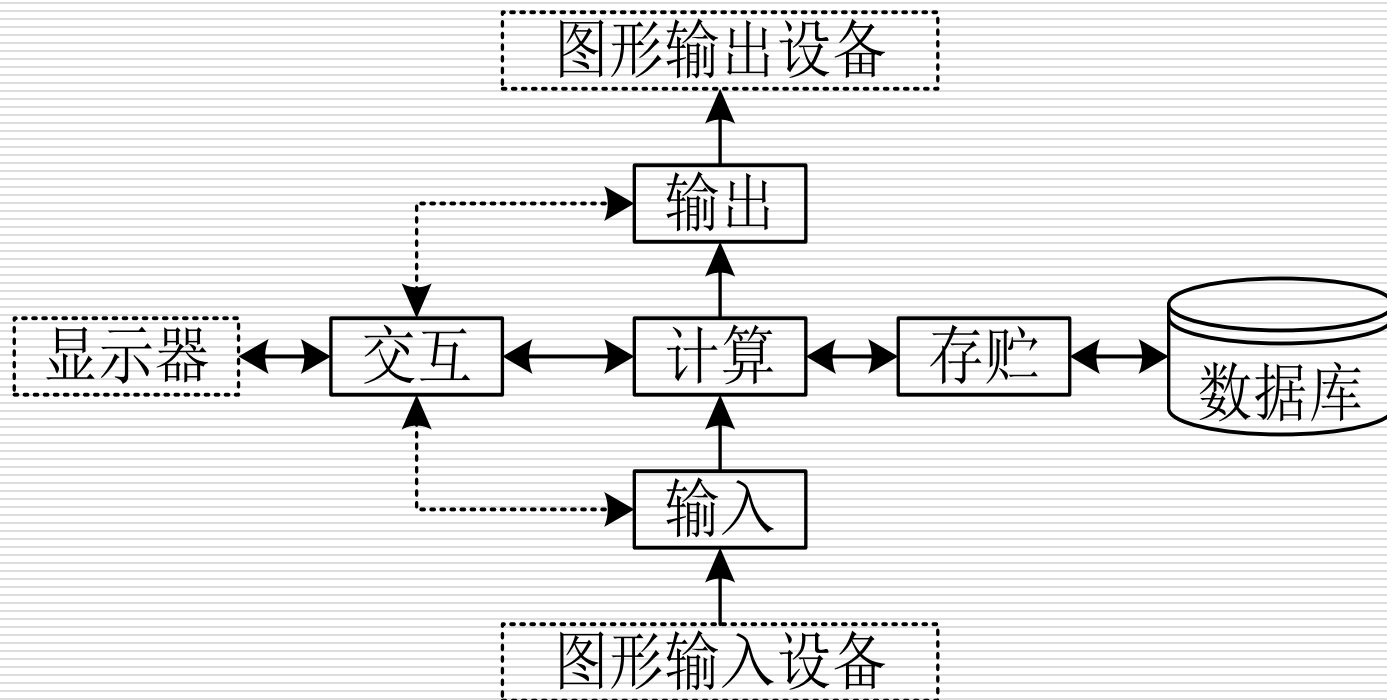


2.1 计算机图形系统

- 计算机图形系统的功能
- 计算机图形系统的结构



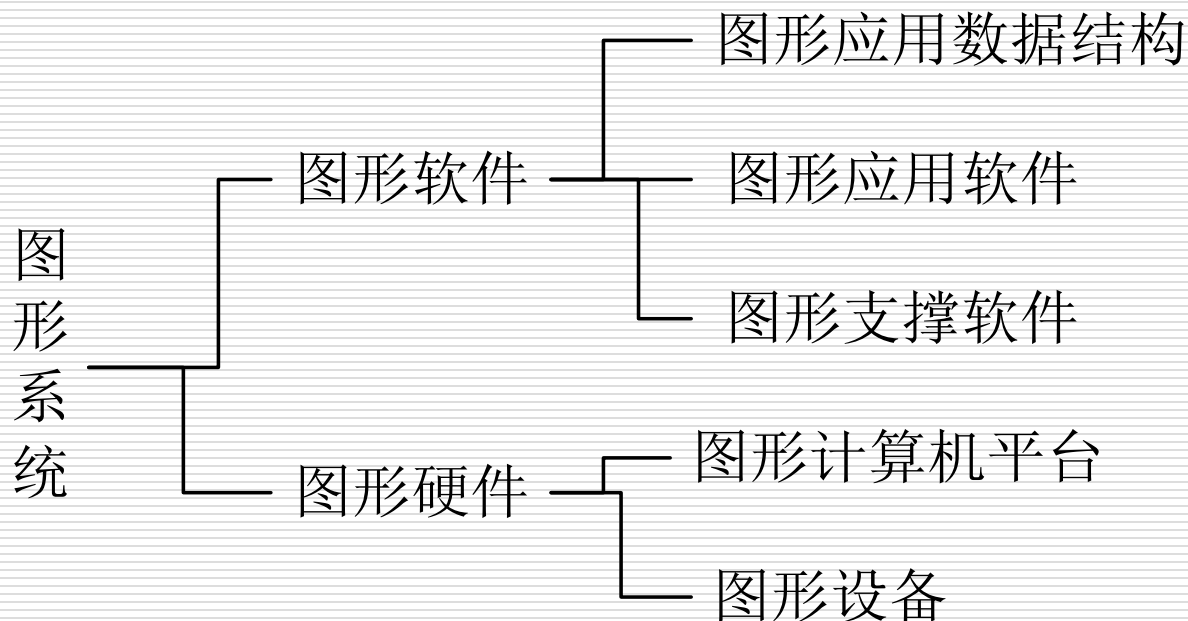
计算机图形系统的功能



计算机图形系统的功能



计算机图形系统的结构



计算机图形系统的结构



2.2 图形输入设备

□ 键盘 (keyboard)


键盘能用来进行屏幕坐标的输入、菜单选择、图形功能选择，控制图形的移动、旋转、图形拉近、推远等。



图形输入设备

□ 鼠标器 (mouse)

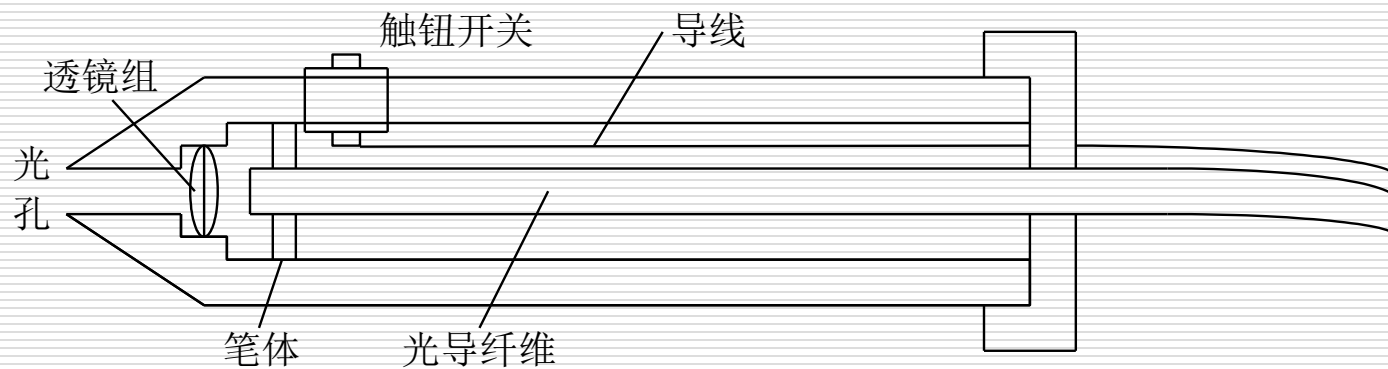
当拖动鼠标器在显示屏上移动时，鼠标将记录移动的方向和距离，这个方向和距离被传送给计算机，转换成对应的光标的位移。

另外，鼠标可作为图形交互时，用户手势(gesture)输入的设备，如：手势  可用与选择某个区域的图形。



图形输入设备

□ 光笔(light pen)



光笔的结构



图形输入设备

□ 触摸屏(touch screen)

当用手指或者小杆触摸屏幕时，触点位置便以光学的（红外线式触摸屏）、电子的（电阻式触摸屏和电容式触摸屏）或声音的（声音探测式）方式记录下来。



图形输入设备

□ 操纵杆(joystick)

操纵杆是由一根小的垂直杠杆组成的可摇动装置，该杠杆装配在一个其四周可移动的底座上用来控制屏幕光标。



图形输入设备

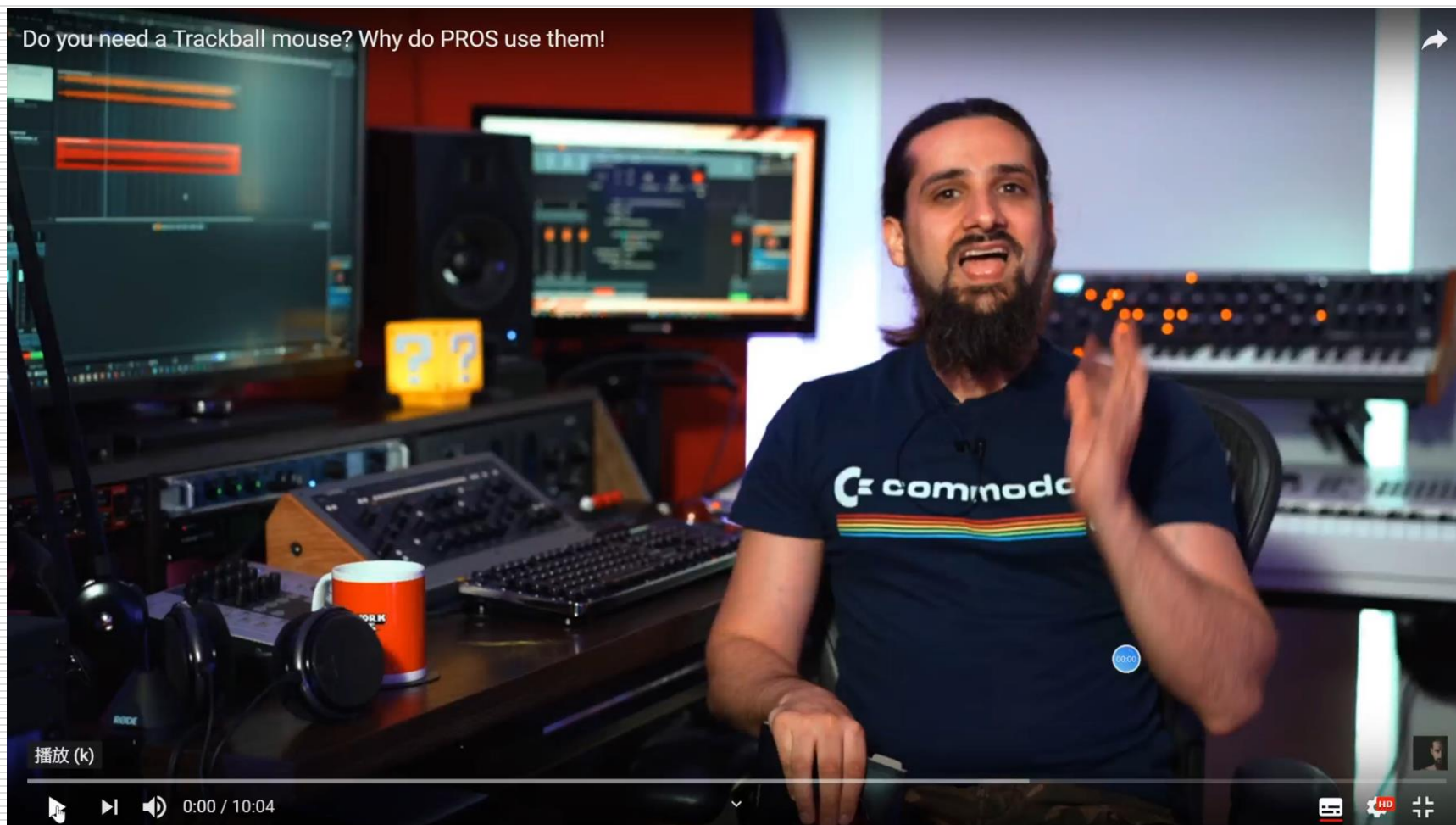


跟踪球

Logitech M570 Wireless Trackball



Do you need a Trackball mouse? Why do PROS use them!



图形输入设备

□ 数据手套(Data Glove)

是一种戴在手上的传感器，可以用来抓住“虚拟对象”，它由一系列检测手和手指运动的传感器构成，用来提供关于手的位置和方向的信息。



BeBop Data Glove Hands-On at CES 2019: Virtual and real worlds merge

BeBop Sensors



播放 (k)

0:00 / 3:30

HD



图形输入设备

□ 数字化仪(Digitizer)

用于在二维或三维的图形对象上扫描，以输入一系列二维或三维的坐标值。这些坐标值代表的坐标点，在系统中将以直线段或曲线段连接，以逼近图形对象的描绘曲线或表面形状。



Digitizer Tablet



播放 (k)



0:00 / 2:13



15



图形输入设备

□ 图像扫描仪(Scanner)

图像扫描仪可直接把纸质文稿、图纸、图表、照片等输入到计算机中，当它们通过一个光学扫描机构时，灰度或彩色等级被记录下来，并按图像方式进行存储。



图形输入设备

☐ 音频输入系统

☐ 视频输入系统



2.3 图形显示设备

- 阴极射线管
- 彩色阴极射线管
- CRT图形显示器

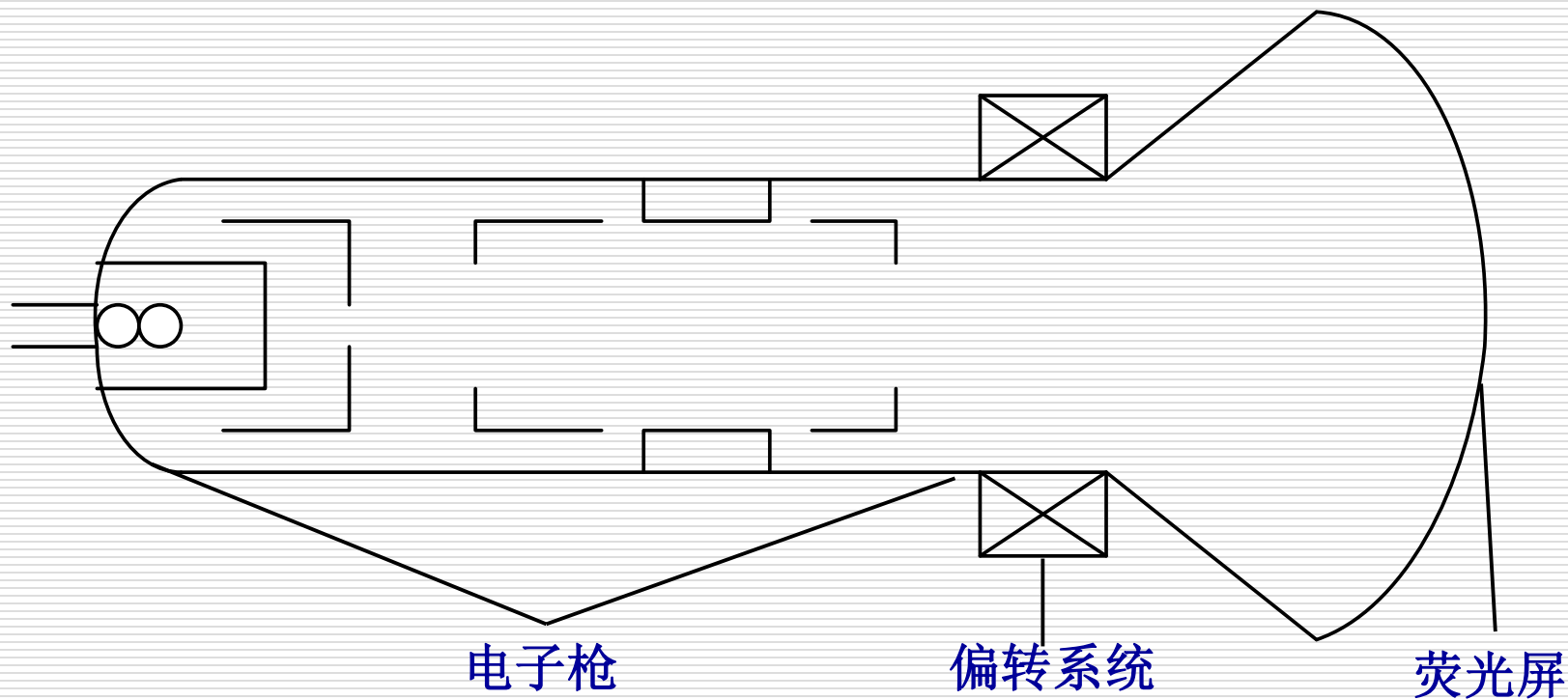


阴极射线管 (CRT)

- ❑ **CRT (Cathode Ray Tube)** 是一种真空器件，它利用电磁场产生高速的、经过聚焦的电子束，偏转到屏幕的不同位置轰击屏幕表面的荧光材料而产生可见图形。
- ❑ **CRT**从结构上分为：电子枪、偏转系统、荧光屏。



阴极射线管 (CRT)



CRT的结构

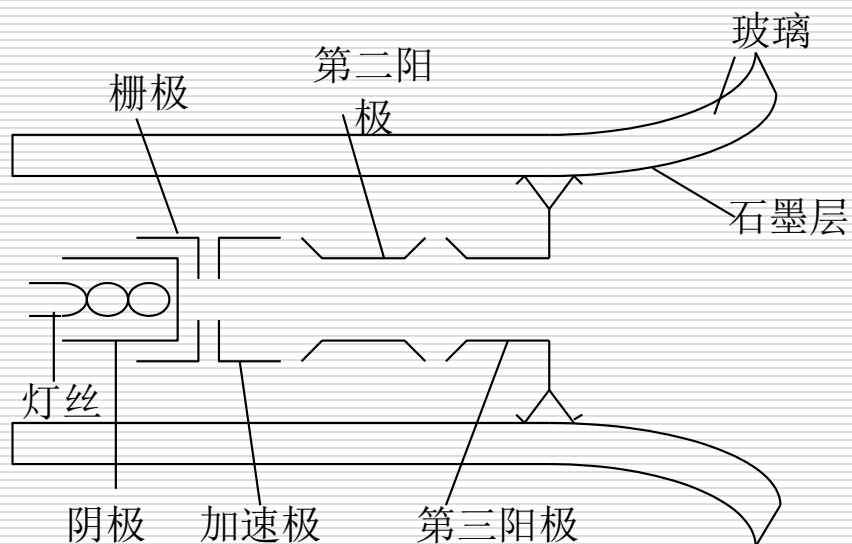


阴极射线管 (CRT)

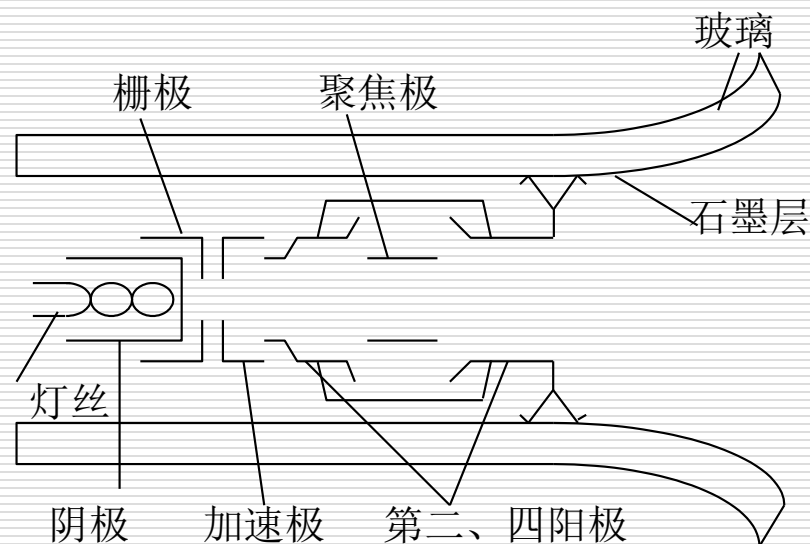
- 电子枪：产生一个沿管轴（**Z**轴）方向前进的高速的细电子束轰击荧光屏。
 - 具有足够的电流强度。
 - 电流的大小和有无必须是可控的。
 - 具有很高的速度。
 - 在荧光屏上应能聚焦很小的光亮，以保证显示器有足够的分辨率。



阴极射线管 (CRT)



(a) 三极电子枪



(b) 四极电子枪

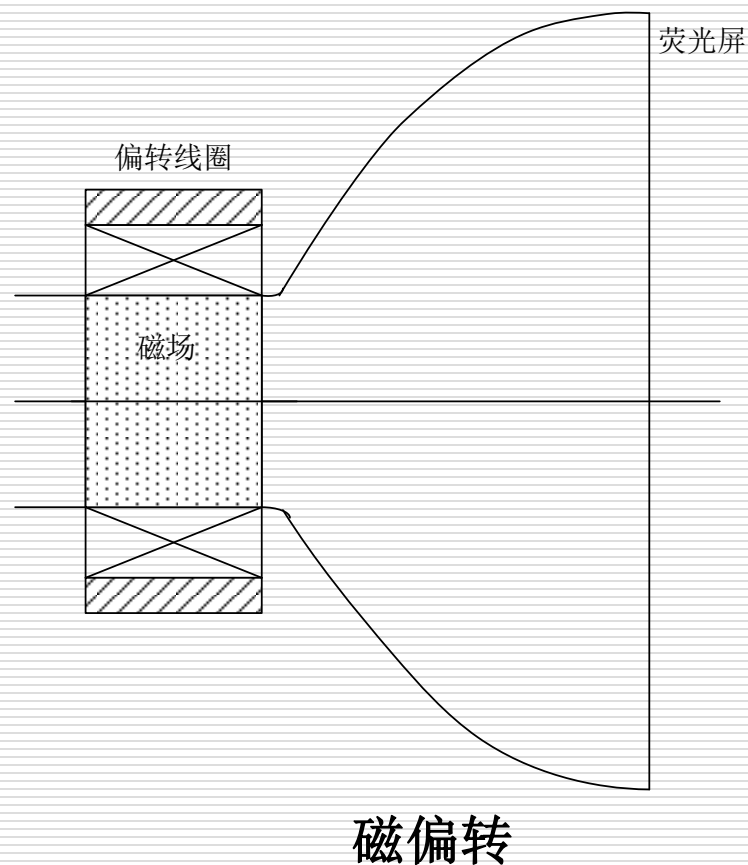
CRT电子枪的结构



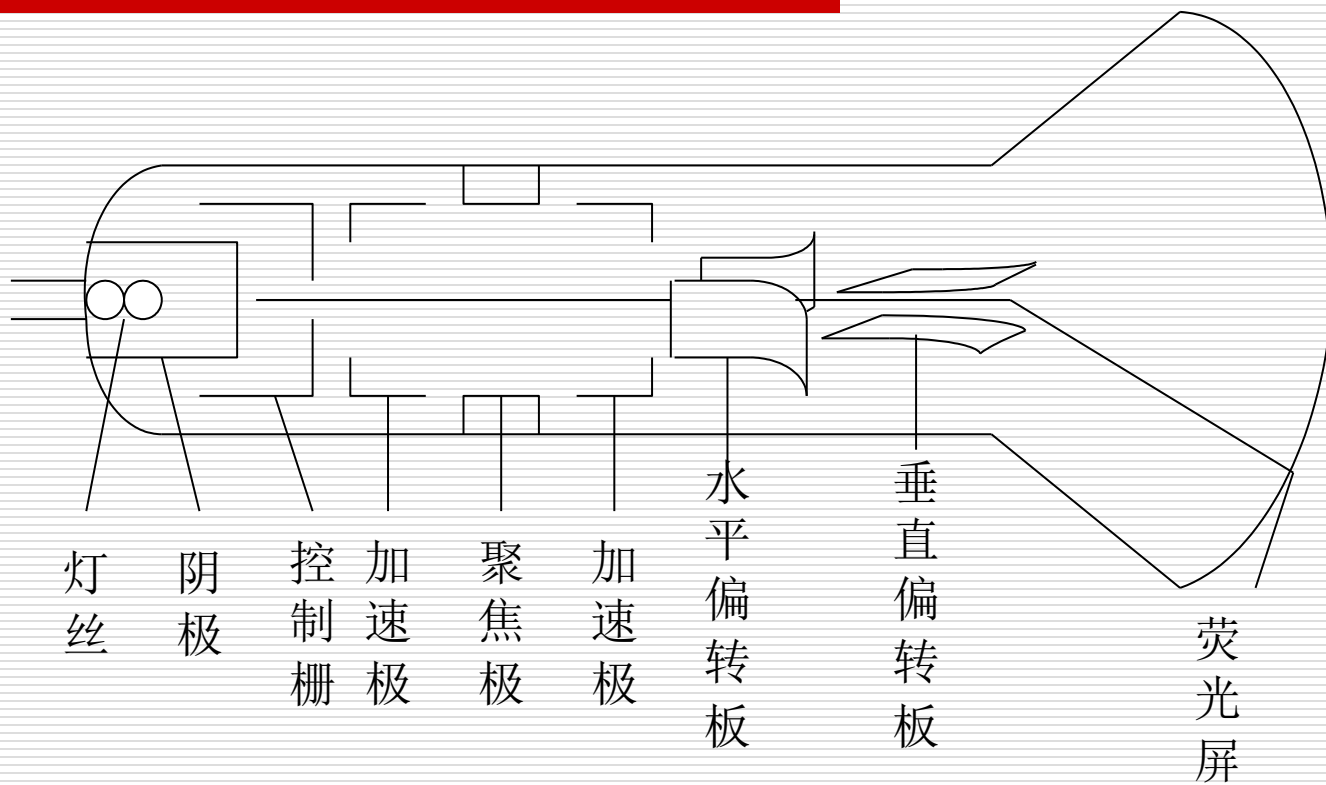
阴极射线管 (CRT)

□ 偏转系统

为了在荧光屏上显示图形及字符，必须用电子束偏转扫描来实现。



阴极射线管 (CRT)



电偏转



阴极射线管 (CRT)

□ 荧光屏 (Phosphor Screen)

- 荧光屏是用荧光粉涂敷在玻璃底壁上制成的，常用沉积法涂敷荧光粉。玻璃底壁要求无气泡，表面光学抛光。
- 荧光粉的性能要求是：发光颜色满足标准白色、发光效率高、余辉时间合适以及寿命长等。



彩色阴极射线管——荫罩式

- 三基色原则
- 主要结构：三色荧光屏、三支电子枪、荫罩板

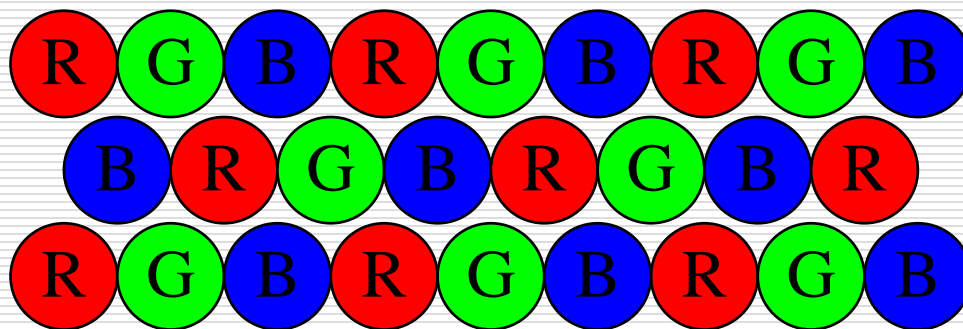
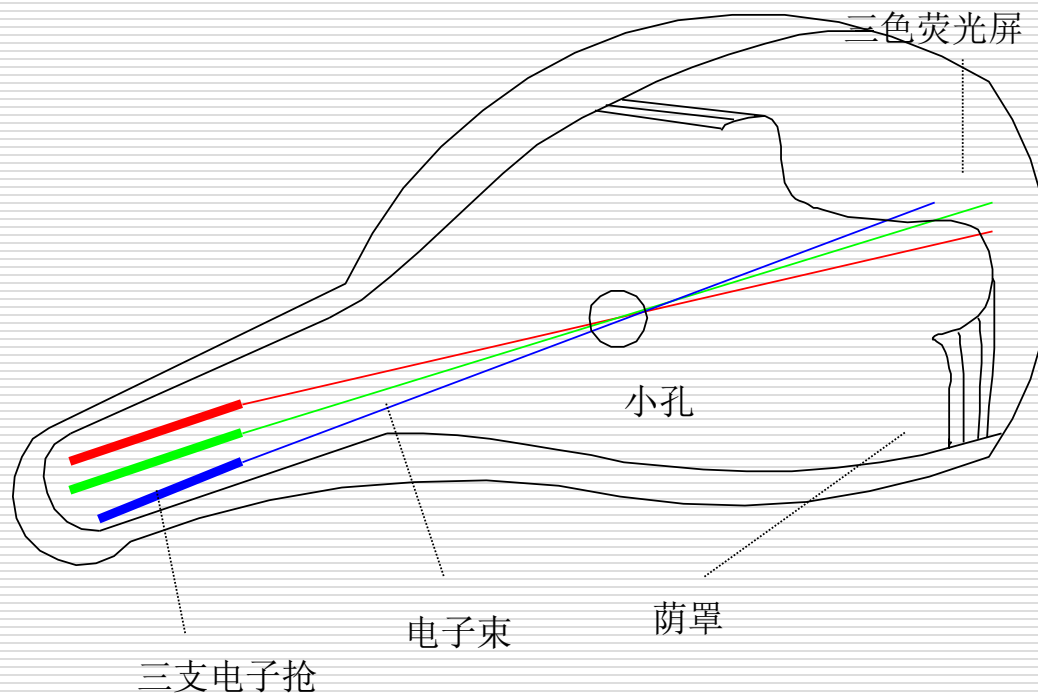


图2.11 三色荧光屏



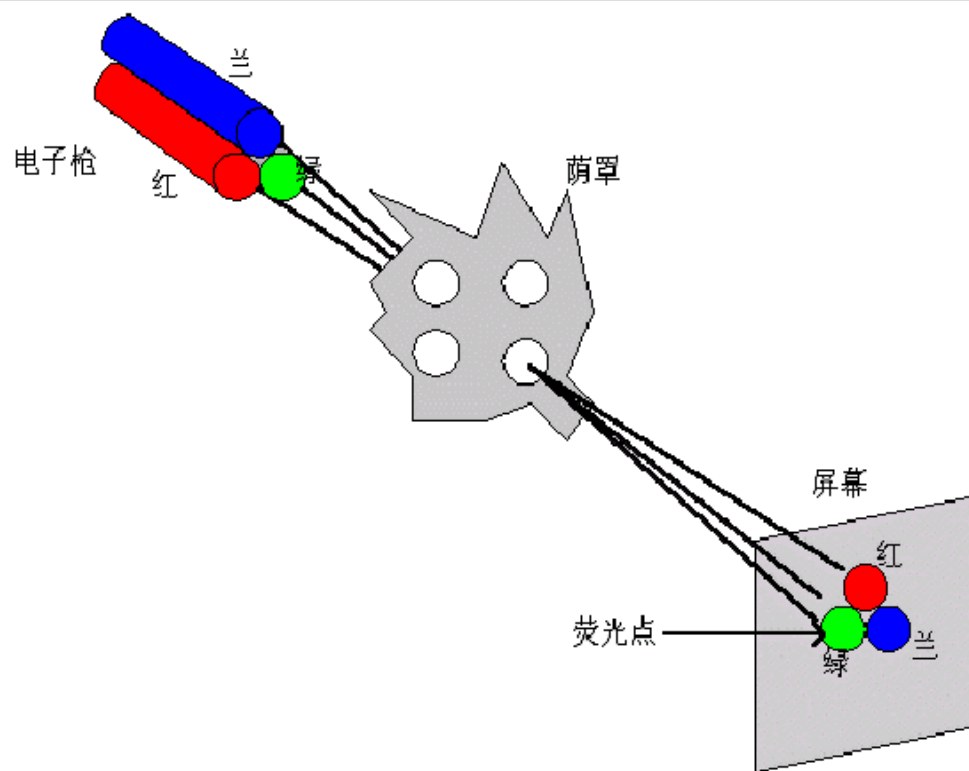
彩色阴极射线管——荫罩式



三支电子枪

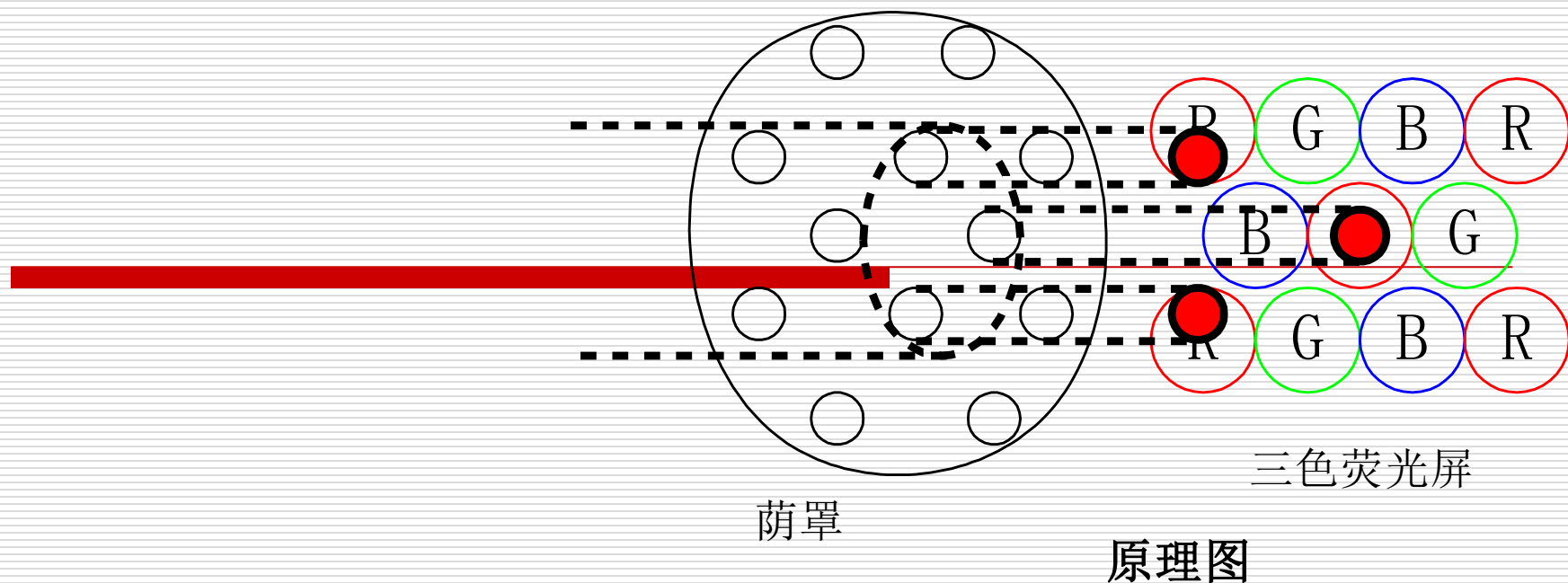


彩色阴极射线管——荫罩式



荫罩

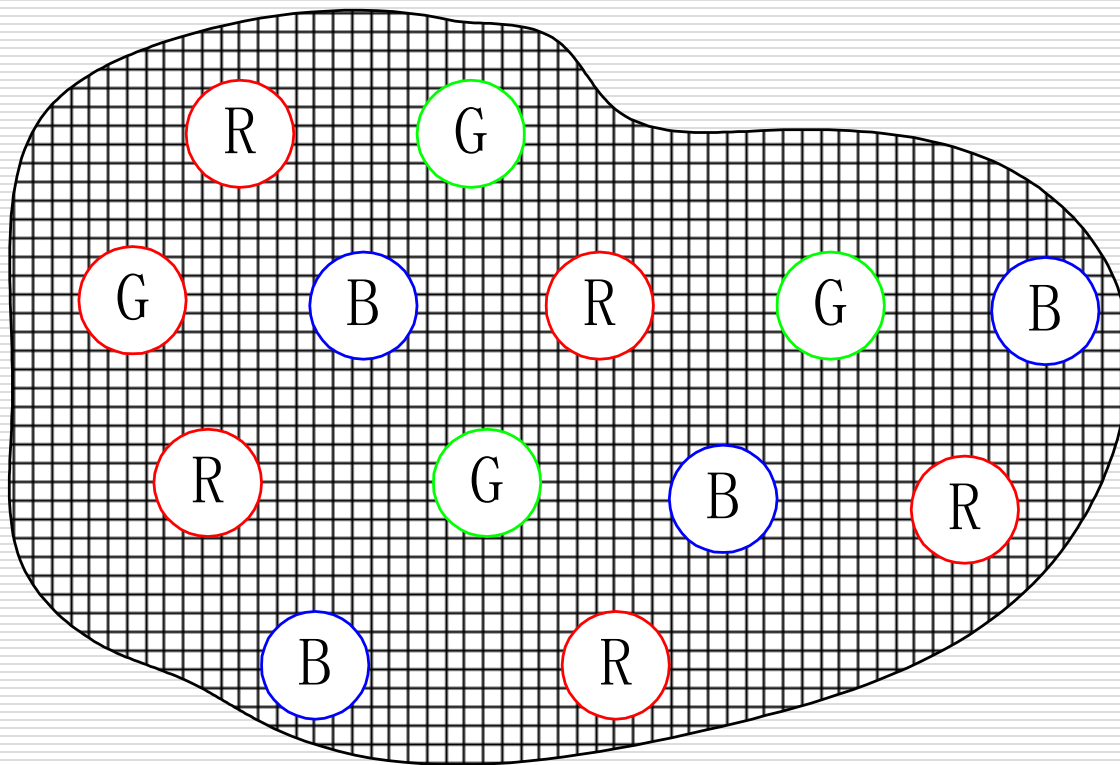




- ❑ 为了避免混色，荧光点的面积比电子束截面要大些。
- ❑ 只有**15%**左右的电子到达屏幕。
- ❑ 荫罩吸收大量电子，容易发热变形。
- ❑ 管子的亮度低，要提高管子的亮度，则需提高第二阳极的高压。



彩色阴极射线管——荫罩式



黑底荫罩管



CRT图形显示器

- 随机扫描的图形显示器
- 光栅扫描的图形显示器

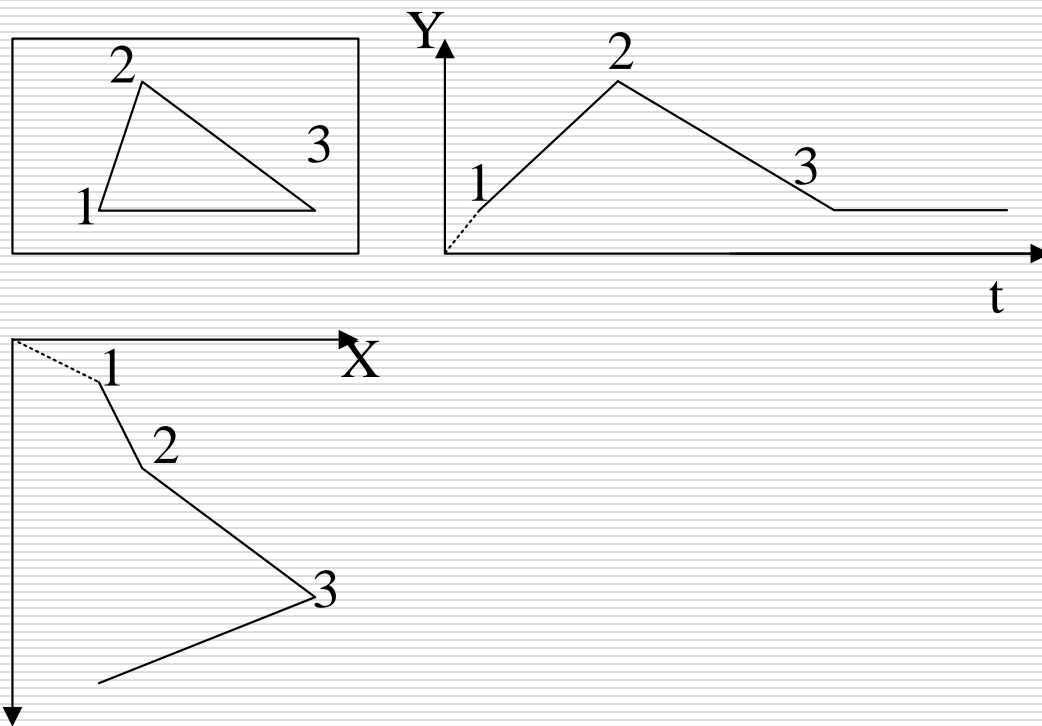


随机扫描的图形显示器

- 向量（**vector**）显示器、笔划（**Stroke writing**）显示器。
- 随机扫描（**random-scan**）的图形显示器中电子束的定位和偏转具有随机性，即电子束的扫描轨迹随显示内容而变化，只在需要的地方扫描，而不必全屏扫描。



随机扫描的图形显示器



随机扫描图形显示器的工作原理

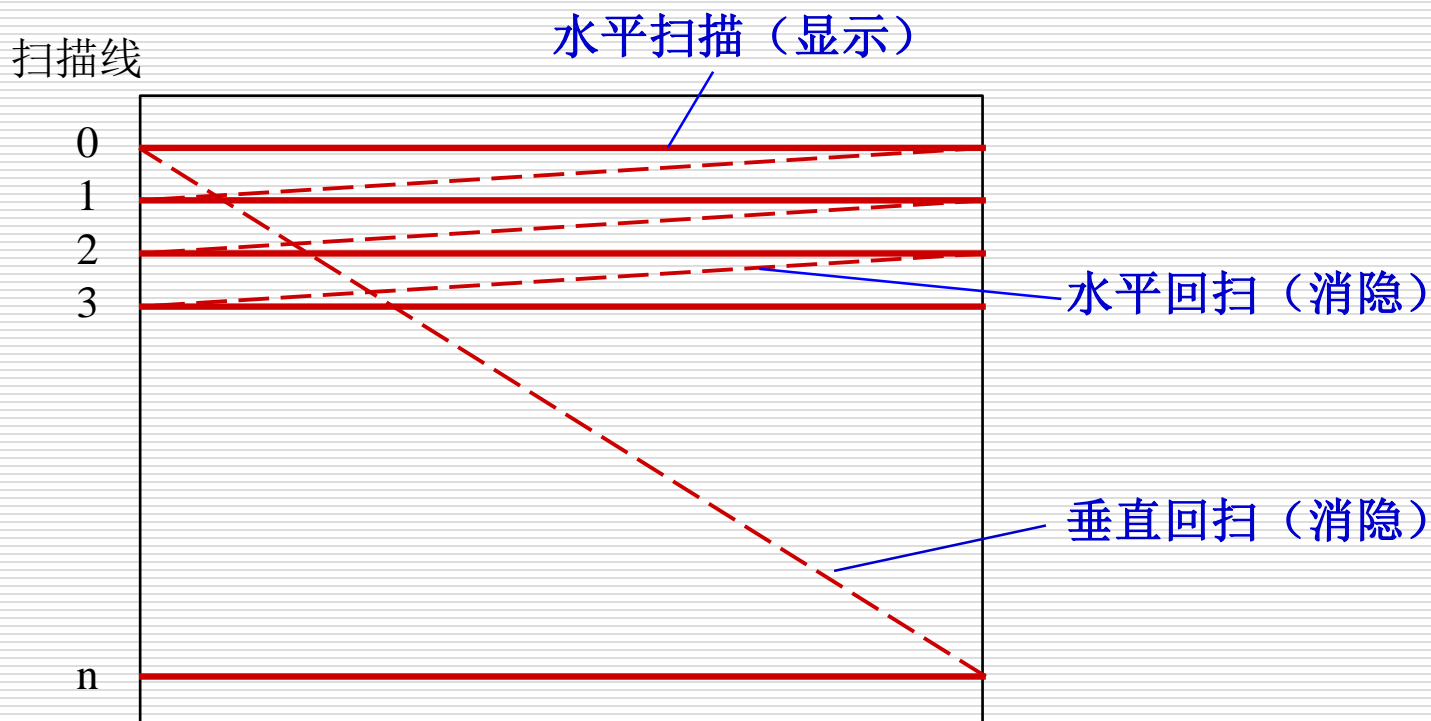


随机扫描的图形显示器

- 无冗余扫描、速度快、图像清晰、比光栅系统更高的分辨率、生成光滑线条；
- 不能显示逼真场景、和电视标准不一致、驱动系统也较复杂。



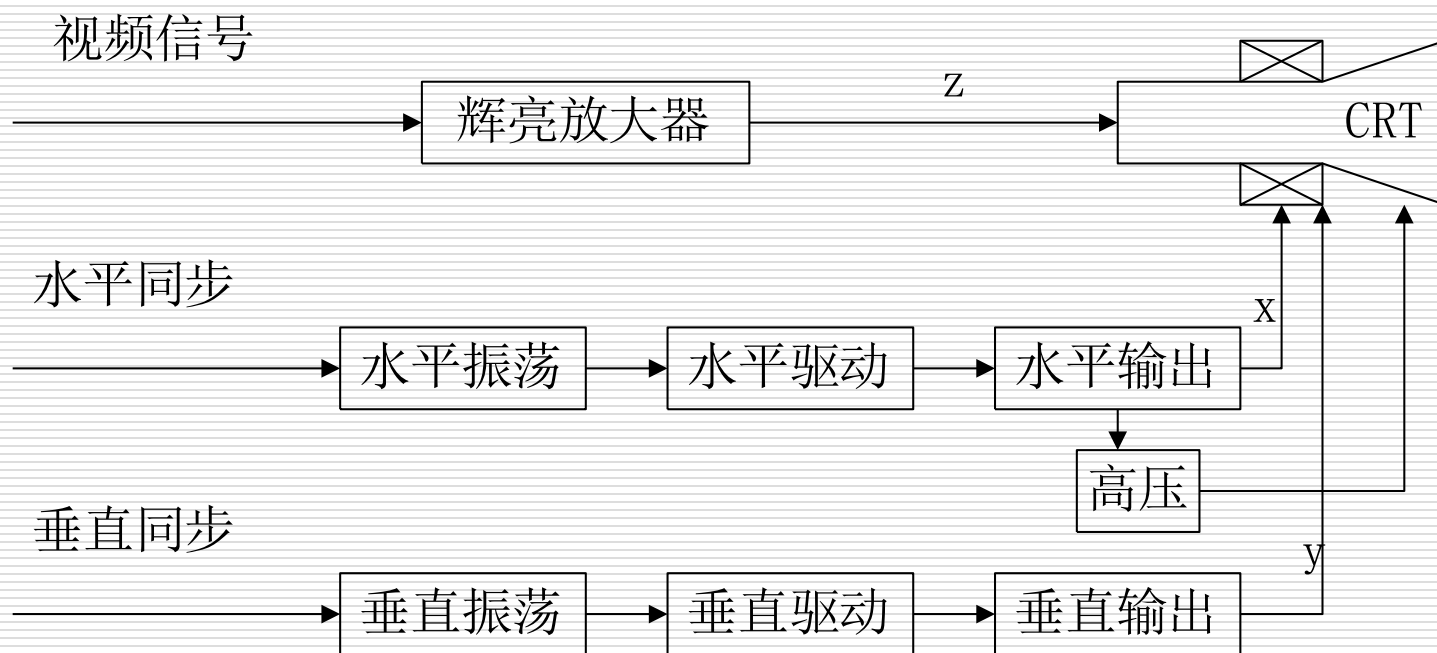
光栅扫描图形显示器



光栅扫描原理



工作原理：光栅扫描是控制电子束按某种光栅形状进行的顺序扫描，而字符、图像是靠**Z**轴信号控制辉亮来形成的。



光栅扫描图形显示器的工作原理



光栅扫描图形显示器

- **刷新缓冲存储器**（refresh buffer）或称帧缓冲存储器（frame buffer）。
- **像素**（pixel或pel，是picture element的简写）。
- 像素信息从应用程序转换并放入帧缓冲区的过程称之为**扫描转换过程**。

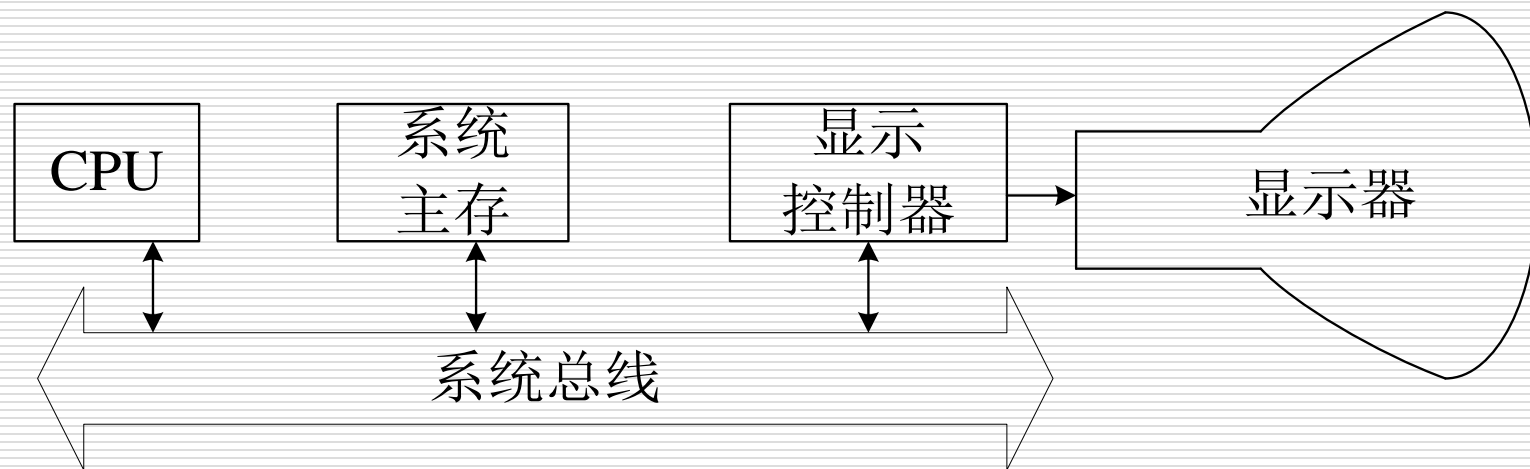


2.4 显示子系统

- 光栅扫描图形显示子系统的结构
- 绘制流水线
- 相关概念



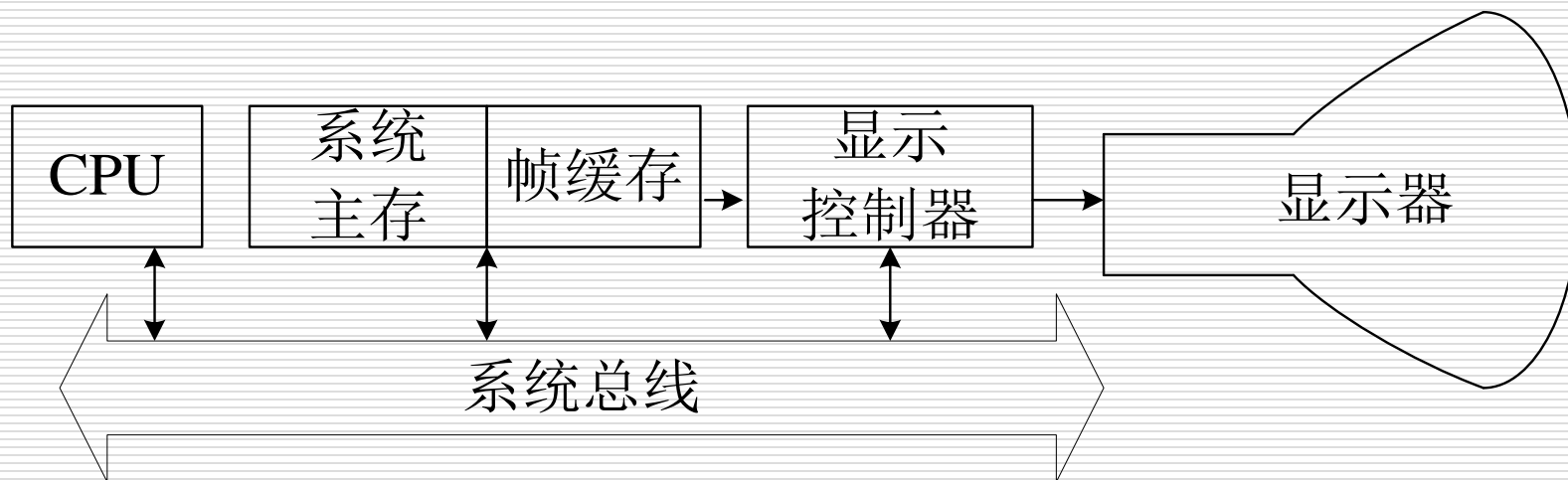
光栅扫描图形显示子系统的结构



简单的光栅图形显示子系统



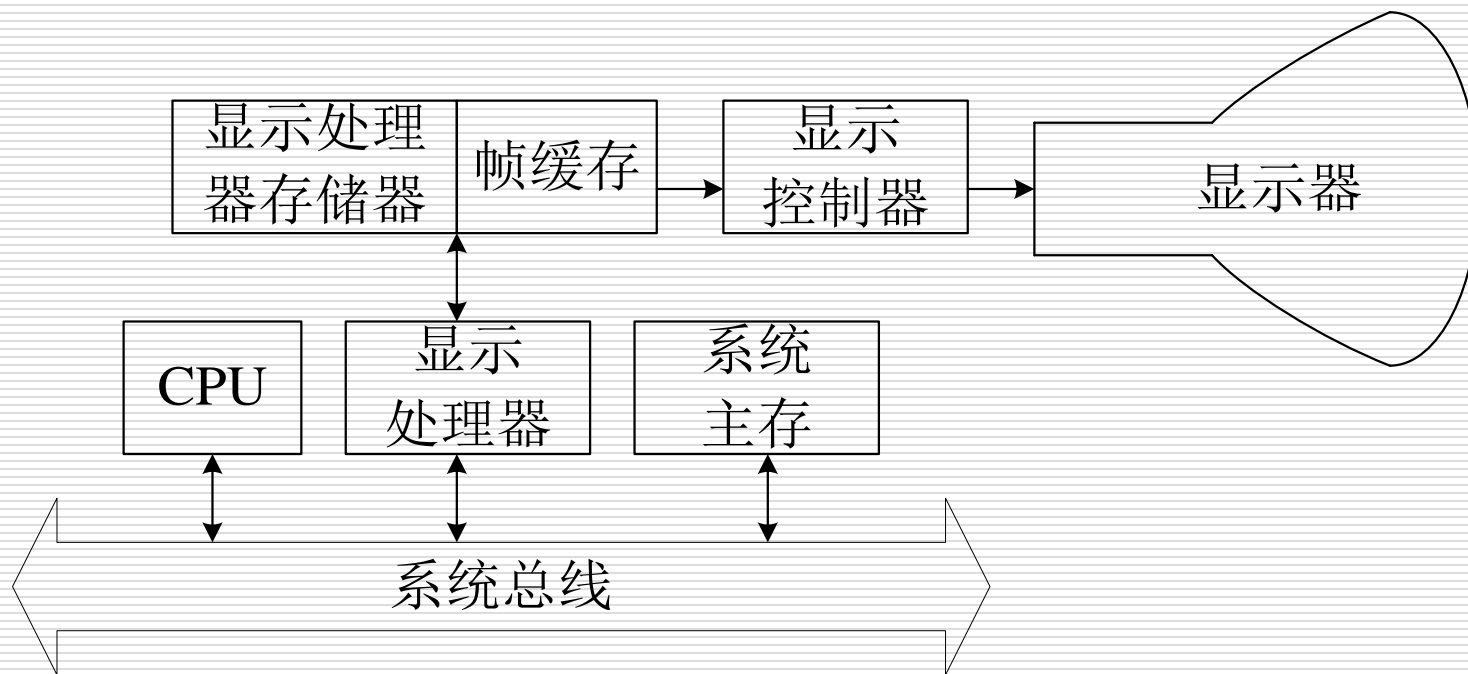
光栅扫描图形显示子系统的结构



早期的光栅图形显示子系统



光栅扫描图形显示子系统的结构



现代的光栅图形显示子系统



光栅扫描图形显示子系统的结构

- **帧缓冲存储器**：用来存储像素颜色（灰度）值的存储器。
- **双缓存**：一个缓存用来刷新的同时，另一个写入数据信息，尔后这两个缓存可互换角色。这种方式称为双缓存，它可以使得显示的动画流畅而没有滑动感。



光栅扫描图形显示子系统的结构

- **显示控制器（Display Controller）**，又称视频控制器（**Video Controller**），主要功能是依据设定的显示工作方式，自主地、反复不断地读取帧缓存中的图像点阵（包括图形、字符文本）数据，将它们转换成**R、G、B**三色信号并配以同步信号送至显示器，即可刷新屏幕。



光栅扫描图形显示子系统的结构

□ 显示处理器（**Display Processor**），又称图形控制器（**Graphics Controller**）或显示协处理器（**Display Coprocessor**），它把**CPU**从图形显示处理的事务中解脱出来，其主要任务是扫描转换待显示的图形以及某些附加的操作等。

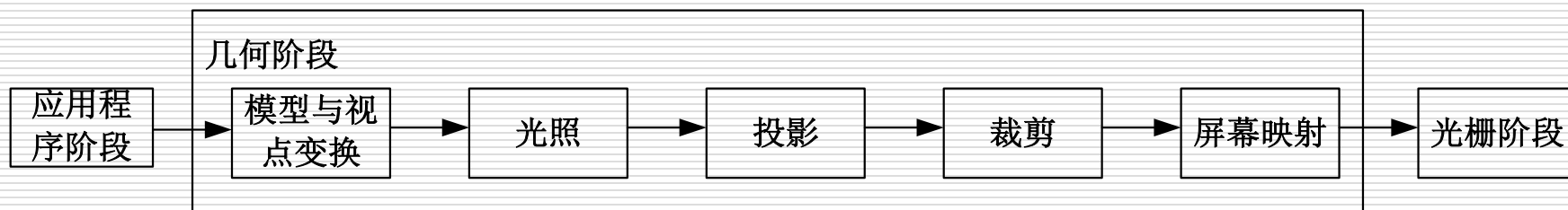


绘制流水线

- 图形子系统的图形绘制功能常常采用流水线（**Pipeline**）结构绘制，或者称为管线绘制。
- 绘制流水线的基本结构从概念上包括三个阶段
 - 应用程序阶段
 - 几何阶段
 - 光栅阶段



绘制流水线



绘制流水线的结构



相关概念

- 分辨率
- 像素与帧缓存
- 颜色查找表
- 显示长宽比
- 屏幕坐标系



相关概念——分辨率

- **光点**一般是指电子束打在显示器的荧光屏上，显示器能够显示的最小的发光点。
- **像素点**是指图形显示在屏幕上时候，按当前的图形显示分辨率所能提供的最小元素点。



相关概念——分辨率

- **屏幕分辨率**，也称为光栅分辨率，它决定了显示系统最大可能的分辨率，任何显示控制器所提供的分辨率也不能超过这个物理分辨率。
- 通常用水平方向上的光点数与垂直方向上的光点数的乘积来表示。



相关概念——分辨率

- **显示分辨率**是计算机显示控制器所能够提供的显示模式分辨率，实际应用中简称为**显示模式**。
- 对于文本显示方式，显示分辨率用水平和垂直方向上所能显示的字符总数的乘积表示。
- 对于图形显示方式，则用水平和垂直方向上所能显示的像素点总数的乘积表示。
- **显示分辨率不同，它所对应的像素点大小也不同。**



相关概念——分辨率

- 图形的存储分辨率是指帧缓冲区的大小，一般用缓冲区的字节数表示。
- 存储分辨率不仅与显示分辨率有关，还与像素点的色彩有关。
- 帧缓存大小的计算：

x 方向的像素点数 \times y 方向的像素点数 $\times \log_2 n / 8$ (BYTE)

其中： n 为颜色数或灰度等级数



相关概念——分辨率

- ❑ 屏幕分辨率决定了所能显示的最高分辨率。
- ❑ 显示分辨率和存储分辨率对所能显示的图形分辨率有控制作用。
- ❑ 显示器的带宽

带宽 = $A \times \text{水平像素点数} \times \text{垂直像素点数} \times \text{刷新频率}$
 A 常取1.344



相关概念——像素与帧缓存

屏幕上一个象素点就对应帧缓存中的一组信息。

- ❑ 组合像素法 (Packed Pixel Method)
- ❑ 颜色位面法 (Color Plane Method)



相关概念——像素与帧缓存

- 在组合像素法中，一个图形像素点的全部信息被编码成一个数据字节，按照一定方式存储到帧缓存中，编码字节的长度与点的属性（如颜色、灰度等）有关。

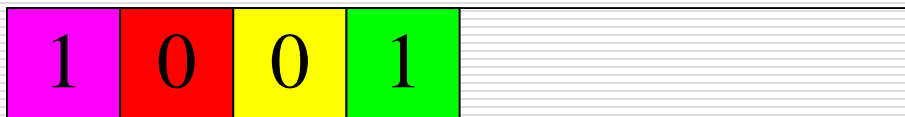


图2.29 组合像素法



相关概念——像素与帧缓存

- 在颜色位面法中，帧缓存被分成若干独立的存储区域，每一个区域称为一个位面（Bit Plane），每一个图形像素点在每个位面中占一位，通过几个位面中的同一位组合成一个像素颜色。

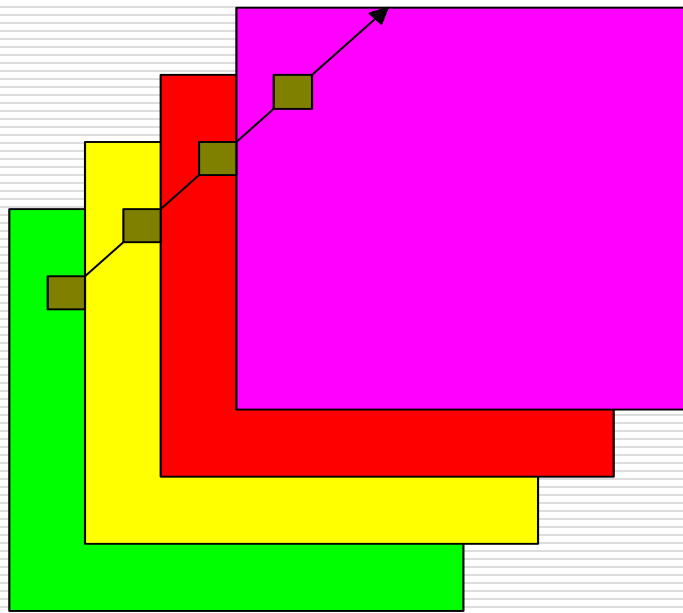


图2.30 颜色位面法



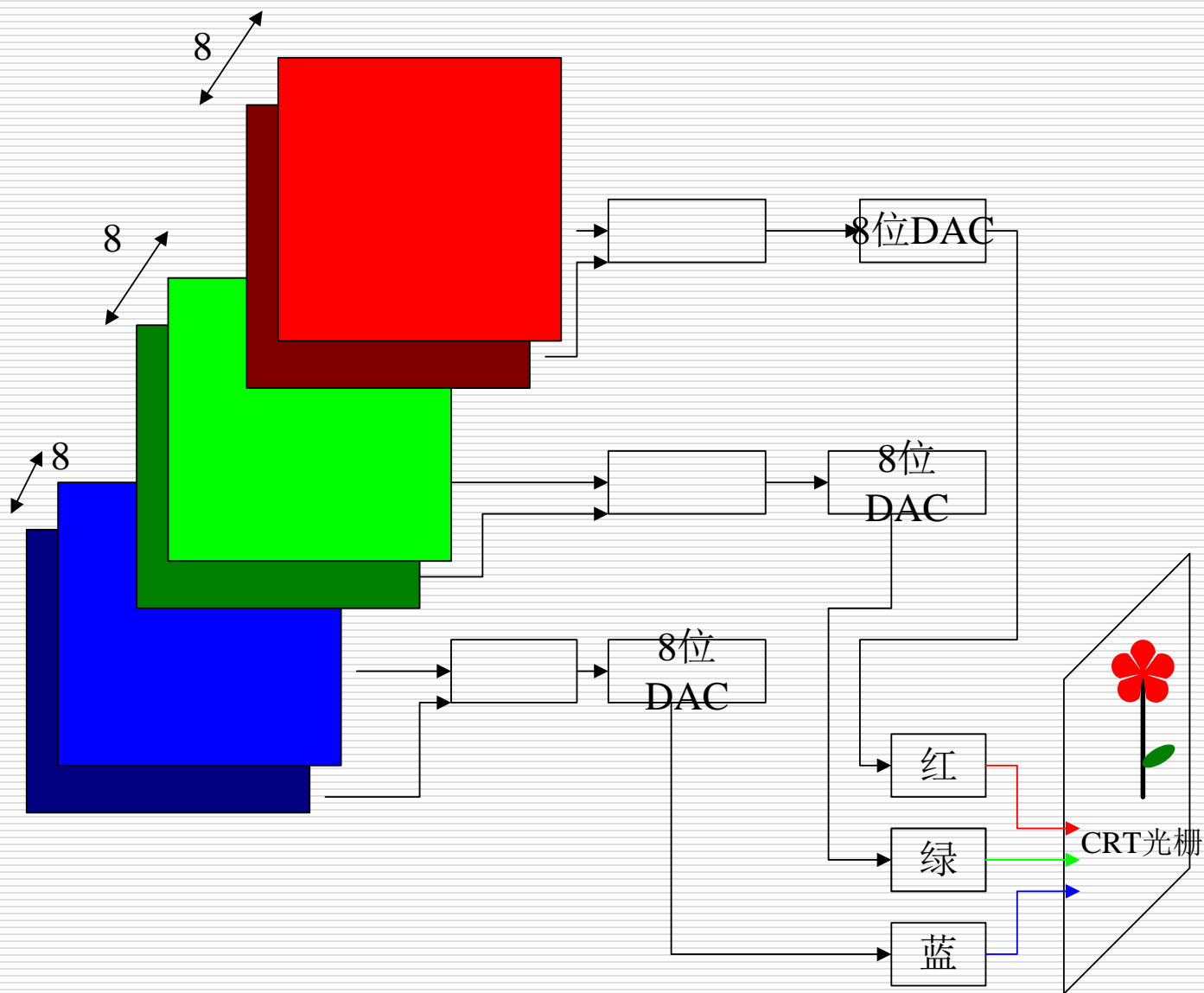


图2.31 具有24位面彩色帧缓存的显示器



相关概念——颜色查找表

□ 颜色查找表也称调色板，是由高速的随机存储器组成，用来储存表达像素色彩的代码。此时帧缓冲存储器中每一像素对应单元的代码不再代表该像素的色彩值，而是作为查色表的地址索引。



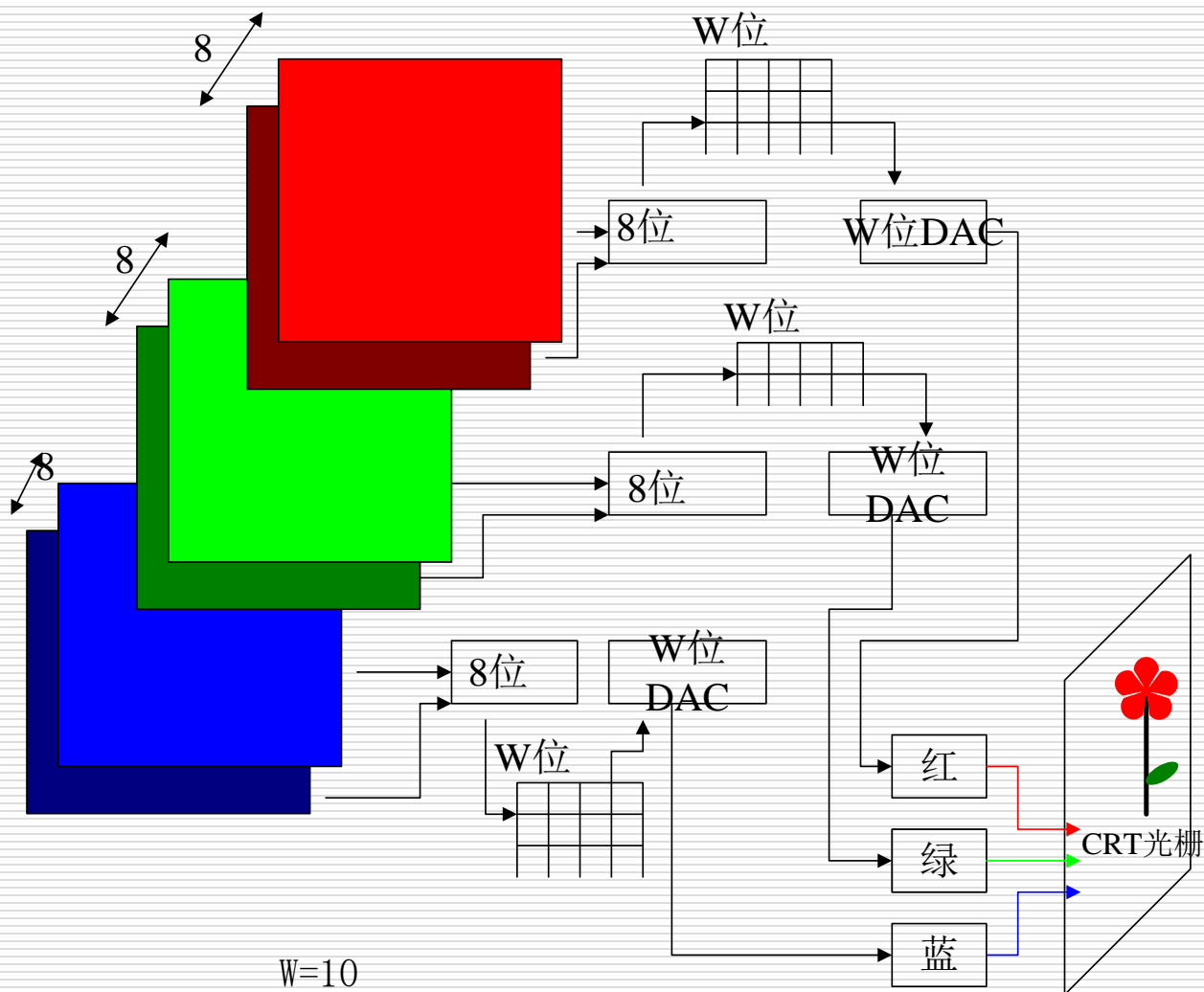


图2.32 具有24位面彩色帧缓存和10位颜色查找表的显示器



相关概念——显示长宽比

□ 显示长宽比，即水平点数与垂直点数之比。

要求在屏幕两个方向上相同像素点数产生同样长度的线段，以使图形不至发生畸变。



相关概念——屏幕坐标系

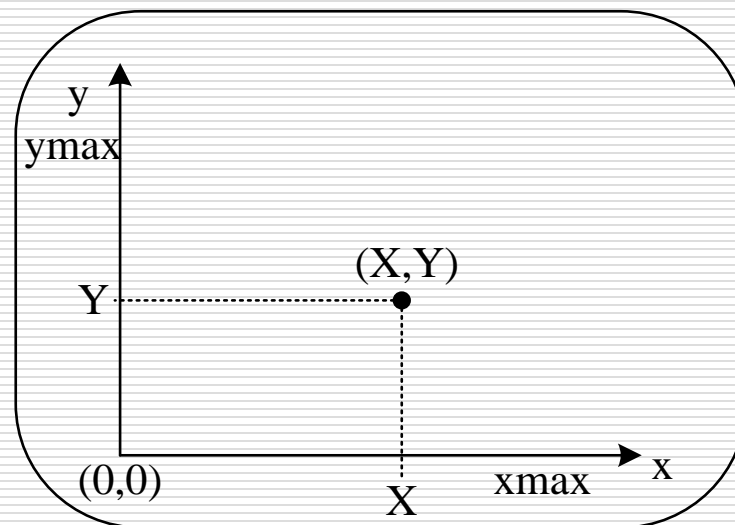
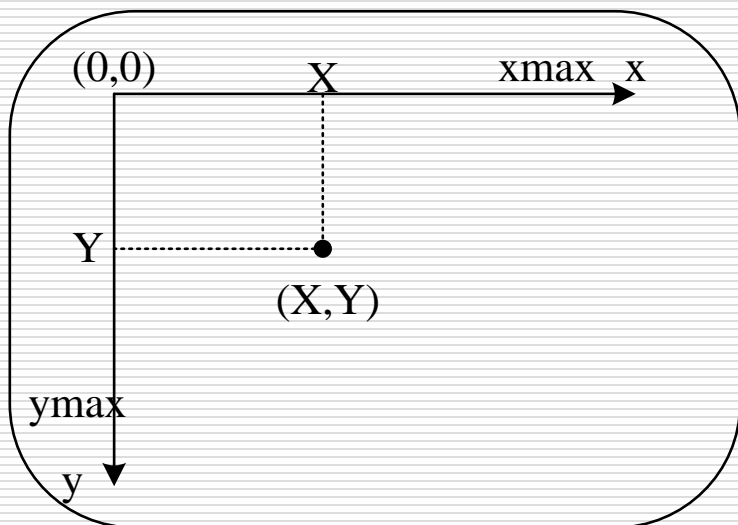


图2.33 不同显示器的坐标



2.5 图形硬拷贝设备

□ 打印机

□ 绘图仪



打印机

- ❑ 打印机是画点设备。
- ❑ 击式打印机主要指针式打印机，它隔着色带将某种点阵图案压在纸上。
- ❑ 非撞击式打印机包括激光打印机、喷墨打印机、热转印打印机、染料热升华打印机和静电打印机等。



绘图仪

- 笔式绘图仪（**Pen Plotter**）是画线设备。
- 笔式绘图仪可分为平板式绘图仪（**Flatbed Plotter**）和滚筒式绘图仪（**Drum Plotter**）



2.6 图形标准及软件包

- DirectX
- OpenGL



DirectX

DirectX是由微软公司创建的一系列专为多媒体以及游戏开发的应用程序接口。旗下包含Direct3D、Direct2D、DirectCompute等等多个不同用途的子部分，因为这一系列API皆以Direct字样开头，所以DirectX就成为这一巨大的API系列的统称。



DirectX

What is DirectX and How Does it Work? (DX11 vs. DX12)



▶ ▶| 🔊 0:00 / 7:44



OpenGL图形软件包

- ❑ OpenGL是SGI（Silicon Graphics Inc.）公司对IRIS GL进行改进，扩展可移植性，形成的一个跨平台开放式图形编程接口。
- ❑ 目前，OpenGL标准由1992年成立的独立财团OpenGL Architecture Review Board（ARB）以投票方式产生，并制成规范文档公布。



OpenGL图形软件包

OpenGL是用于渲染2D、3D矢量图形的跨语言、跨平台的应用程序编程接口。这个接口由近350个不同的函数调用组成，用来从简单的图形比特绘制复杂的三维景象。OpenGL常用于CAD、虚拟现实、科学可视化程序和电子游戏开发。



OpenGL图形软件包

Welcome to OpenGL



播放 (k)



0:00 / 16:03



69



OpenGL图形软件包

- OpenGL的主要功能
- OpenGL的绘制流程
- OpenGL的基本语法
- OpenGL的绘制流程详解



OpenGL的主要功能

- 模型绘制
- 模型观察
- 颜色模式
- 光照应用
- 图像效果增强
- 位图和图像处理



OpenGL的主要功能

- 纹理映射
- 实时动画
- 交互技术



OpenGL的绘制流程——工作方式

- 一个完整的窗口系统的
OpenGL图形处理系统的
结构为：最底层为图形硬
件，第二层为操作系统，
第三层为窗口系统，第四
层为OpenGL，最上面的
层为应用软件。

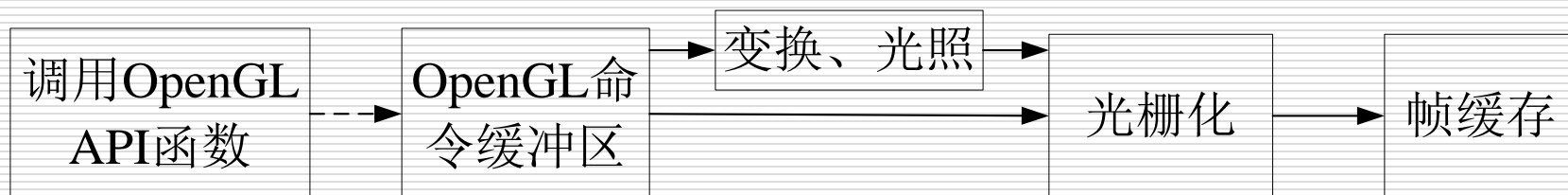


**OpenGL图形处理系统在计
算机系统中的层次结构**



OpenGL的绘制流程——流水线

- OpenGL命令将被放在一个命令缓冲区中，这样命令缓冲区中包含了大量的命令、顶点数据和纹理数据。当缓冲区被清空时，缓冲区中的命令和数据都将传递给流水线的下一个阶段。



一条简化的OpenGL绘制流水线



OpenGL的基本语法——相关库

- ❑ OpenGL核心库: **gl**
- ❑ OpenGL实用程序库: **glu**
- ❑ OpenGL编程辅助库: **aux**
- ❑ OpenGL实用程序工具包（OpenGL utility toolkit, GLUT）: **glut**
- ❑ Windows专用库: **wgl**



OpenGL的基本语法——命名规则

- OpenGL函数都遵循一个命名约定，即采用以下格式：

<库前缀><根命令><可选的参数个数><可选的参数类型>

例如函数glColor3f(...), gl表示这个函数来自库gl.h, color表示该函数用于颜色设定, 3f表示这个函数采用了三个浮点数参数。



OpenGL的基本语法——数据类型

表2.1 OpenGL的数据结构

OpenGL数据类型	内部表示法	定义为C类型	C字面值后缀
GLbyte	8位整数	signed char	B
GLshort	16位整数	short	S
GLint, GLsizei	32位整数	long	L
GLfloat, GLclampf	32位浮点数	float	F
GLdouble, GLclampd	64位浮点数	double	D
GLubyte, GLboolean	8位无符号整数	unsigned char	Ub
GLushort	16位无符号整数	unsigned short	Us
GLuint, GLenum, GLbitfield	32位无符号整数	unsigned long	Ui



OpenGL程序实例——头文件包含

- 利用OpenGL实现图形绘制，首先要引入OpenGL核心库以及其他需要使用的库的头文件。
- GLUT保证了gl.h和glu.h被正确包含。

```
#include <gl/glut.h>
```

```
#include <windows.h>
```



OpenGL程序实例——窗口管理

- ❑ 初始化（ `glutInit` ）
- ❑ 创建窗口（ `glutCreateWindow` ）
- ❑ 设定窗口的显示模式（ `glutInitDisplayMode` ）
- ❑ 指定窗口的位置和大小（
 `glutInitWindowPosition`和`glutInitWindowSize`）
- ❑ 指定窗口的显示内容函数（ `glutDisplayFunc` ）
- ❑ 运行框架（ `glutMainLoop` ）



OpenGL程序实例——绘制图形

- 指定窗口背景色（ `glClearColor` ）
- 颜色管理，使用**RGB**颜色模型。一种颜色用红、绿、蓝三种颜色成分混合而成，每种颜色成分使用**0.0**到**1.0**之间的任意有效浮点数来表示颜色值。



OpenGL程序实例——绘制图形

表2.2 OpenGL的一些常用混合色

混合色	红色成分 (R)	绿色成分 (G)	蓝色成分 (B)
黑	0.0	0.0	0.0
红	1.0	0.0	0.0
绿	0.0	1.0	0.0
黄	1.0	1.0	0.0
蓝	0.0	0.0	1.0
紫	1.0	0.0	1.0
青	0.0	1.0	1.0
深灰	0.25	0.25	0.25
浅灰	0.75	0.75	0.75
棕	0.60	0.40	0.12
南瓜橙	0.98	0.625	0.12
粉红	0.98	0.04	0.70
紫红	0.60	0.40	0.70
白	1.0	1.0	1.0



OpenGL程序实例——绘制图形

□ 刷新窗口的缓冲区（ `glClear` ）

□ 设定投影参数

```
glMatrixMode(GL_PROJECTION);
```

```
gluOrtho2D(0.0,200.0,0.0,150.0);
```

□ 绘制图形

```
glRectf(50.0f, 100.0f, 150.0f, 50.0f);
```



OpenGL程序实例——绘图命令



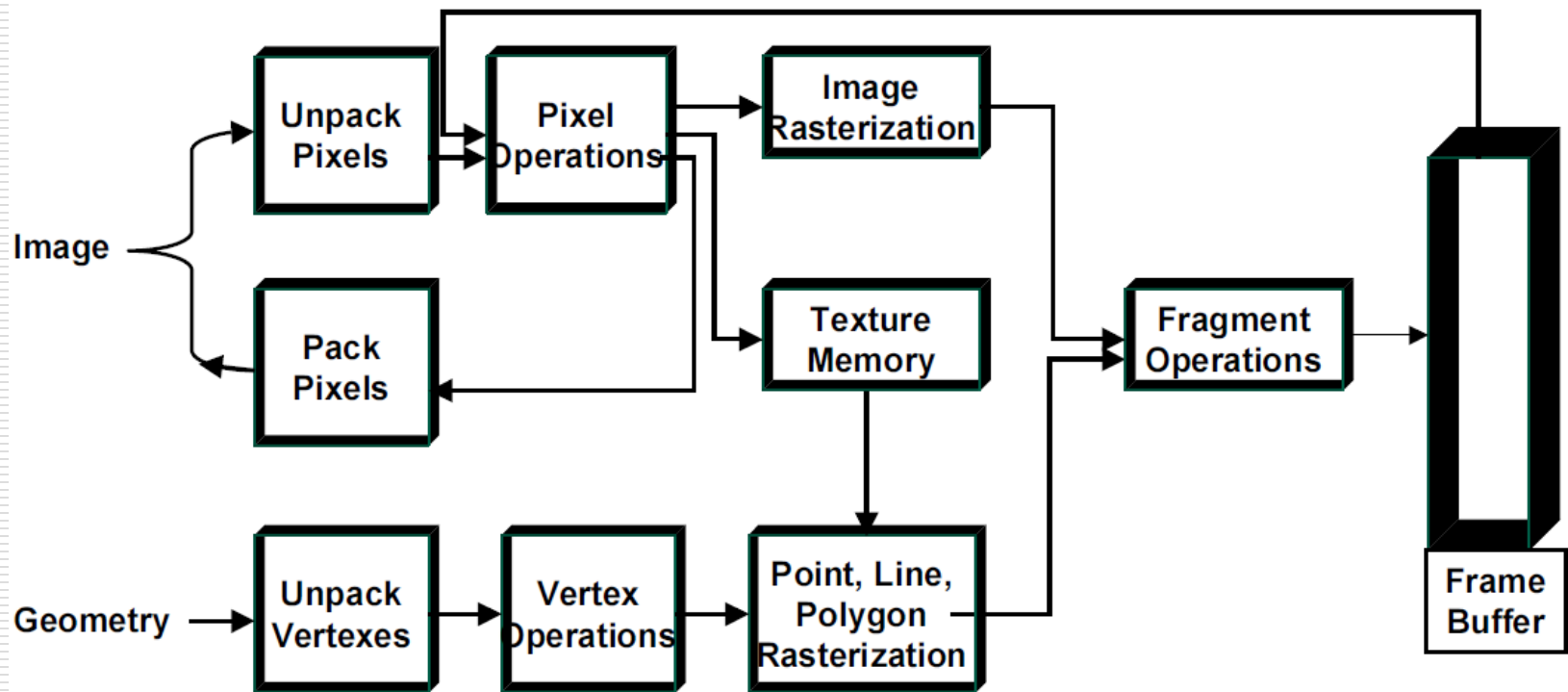
```
glBegin(GL_POLYGON) ;  
    glColor (RED) ;  
    glVertex3i (0,0,0) ;  
    glVertex3i (1,0,0) ;  
    glVertex3i (0,1,0) ;  
glEnd()
```



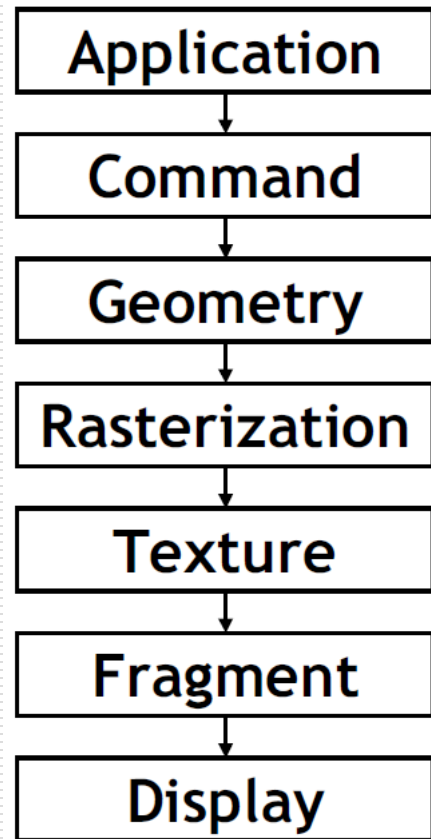
```
glBegin(GL_POLYGON) ;  
    glColor (RED) ;  
    glVertex3i (0,0,0) ;  
    glColor (BLUE) ;  
    glVertex3i (1,0,0) ;  
    glColor (BLUE) ;  
    glVertex3i (0,1,0) ;  
glEnd()
```



OpenGL 结构



OpenGL绘制管道流水



OpenGL程序——命令阶段

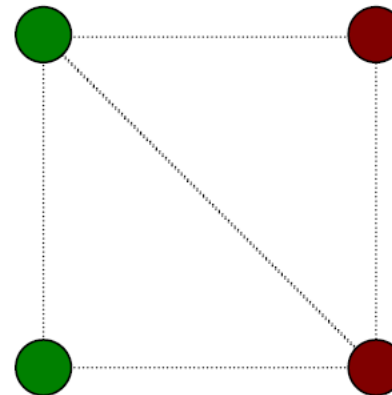
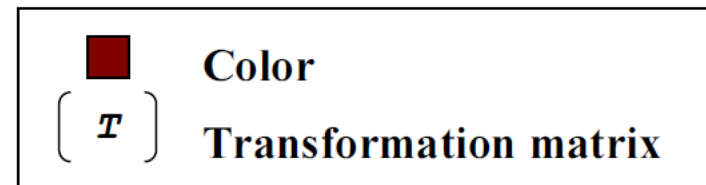
Command buffering

Command interpretation

Unpack and perform format conversion

Maintain graphics state

```
glLoadIdentity( );
glMultMatrix( T );
glBegin( GL_TRIANGLE_STRIP );
glColor3f ( 0.0, 0.5, 0.0 );
glVertex3f( 0.0, 0.0, 0.0 );
glColor3f ( 0.5, 0.0, 0.0 );
glVertex3f( 1.0, 0.0, 0.0 );
glColor3f ( 0.0, 0.5, 0.0 );
glVertex3f( 0.0, 1.0, 0.0 );
glColor3f ( 0.5, 0.0, 0.0 );
glVertex3f( 1.0, 1.0, 0.0 );
...
glEnd( );
```

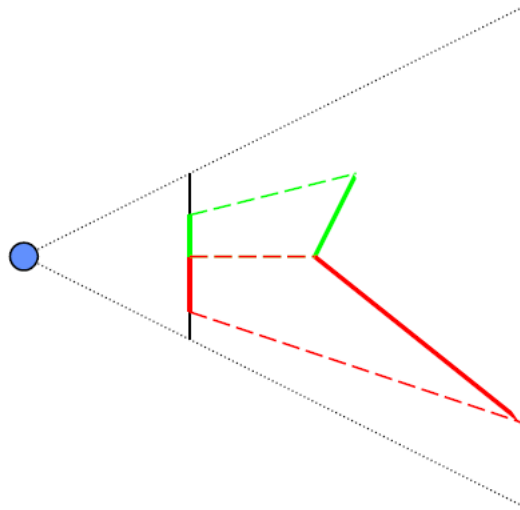


OpenGL程序——几何阶段

Evaluation of polynomials for curved surfaces

Transform and projection

Clipping, culling and primitive assembly



OpenGL程序——几何阶段

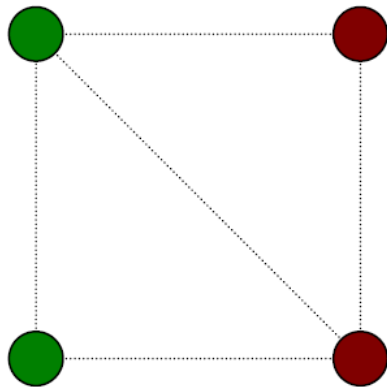
Evaluation of polynomials for curved surfaces

Transform and projection (object -> image space)

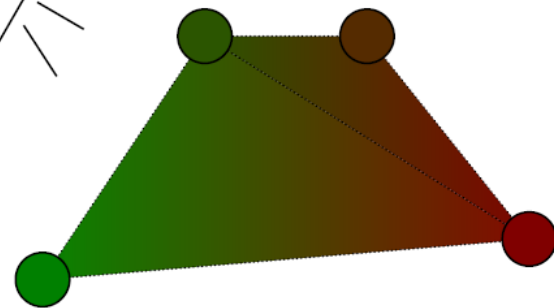
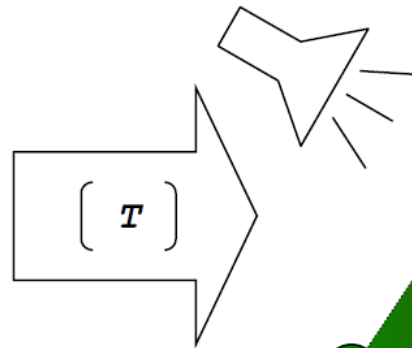
Clipping, culling and primitive assembly

Lighting (light sources and surface reflection)

Texture coordinate generation



Object-space triangles



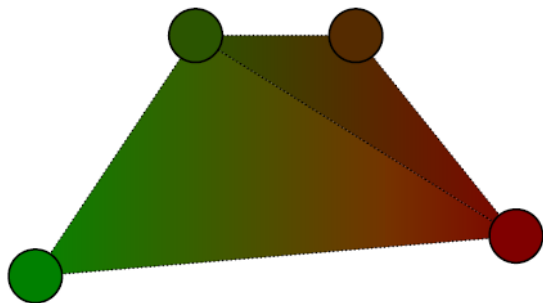
Screen-space lit triangles

OpenGL程序——光栅化阶段

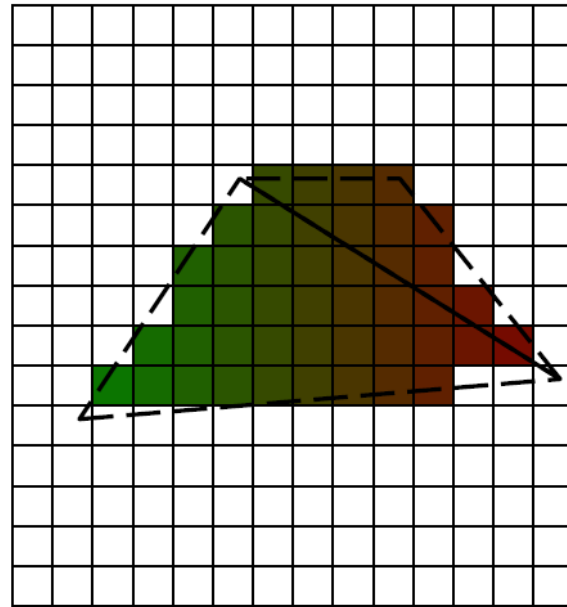
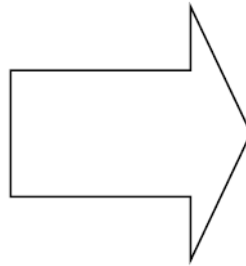
Setup (per-triangle)

Sampling (triangle = {fragments})

Interpolation (interpolate colors and coordinates)



Screen-space triangles



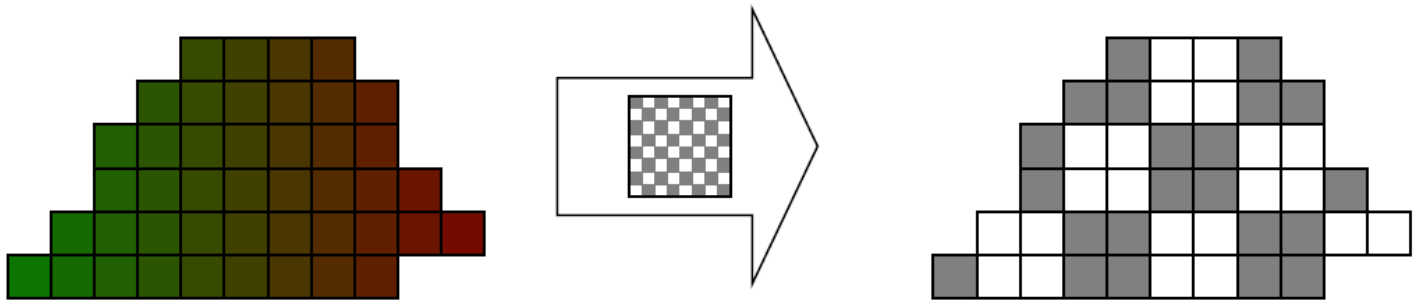
Fragments

OpenGL程序——纹理阶段

Texture transformation and projection

Texture address calculation

Texture filtering

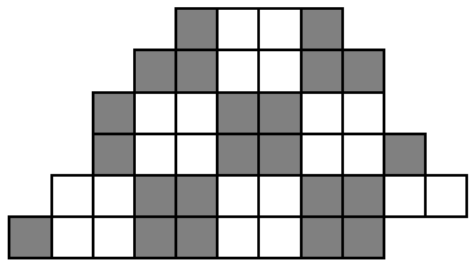


Fragments

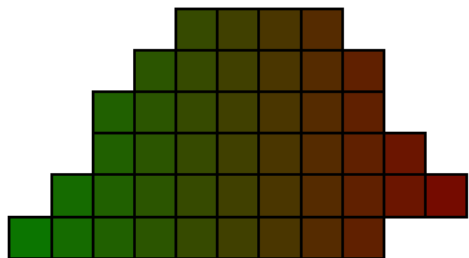
Texture Fragments

OpenGL程序--片元阶段 (像素着色)

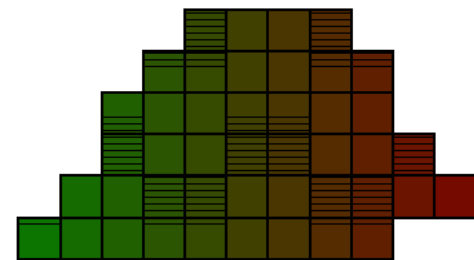
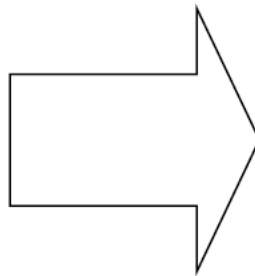
Texture combiners



Texture Fragments



Fragments



Textured Fragments

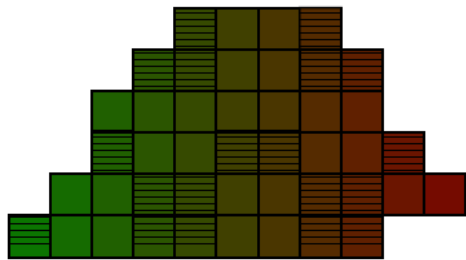
OpenGL程序--片元阶段 (像素着色)

Texture combiners and fog

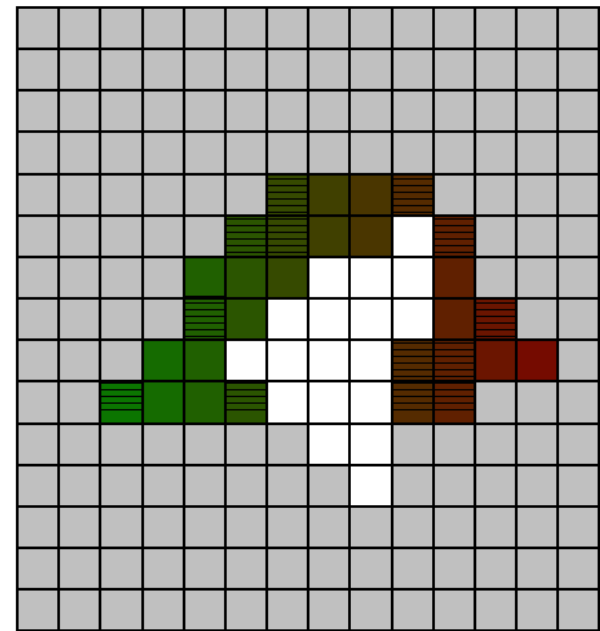
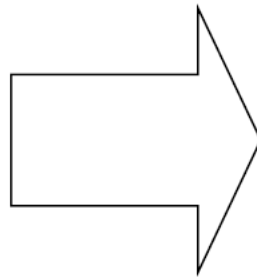
Owner, scissor, depth, alpha and stencil tests

Blending or compositing

Dithering and logical operations

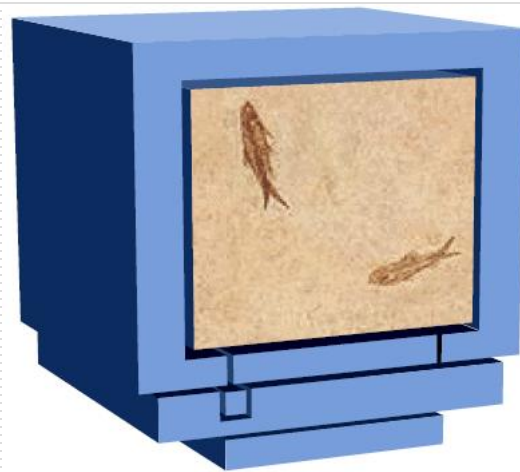
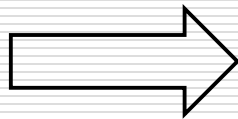
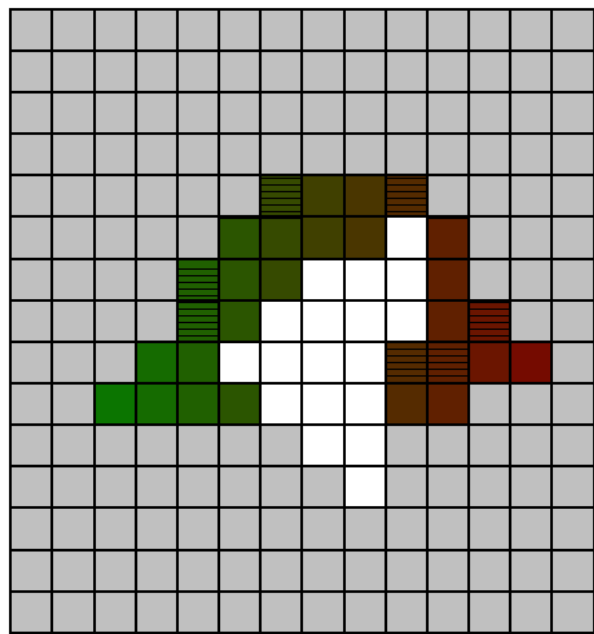


Textured Fragments



Framebuffer Pixels

OpenGL程序——显示阶段



帧缓冲**Framebuffer**中的像素