

GIS专业主干课 : 21905001

计算机图形学

Computer Graphics

林伟华

中国地质大学（武汉）信息工程学院

lwhecug@163.com

目录

- 计算机图形系统
- 计算机显示设备
- 计算机输入设备
- 图形软件标准
- 坐标系统

研究对象

图形与图像：从客观世界中抽象出来的带有**颜色及形状**信息。

—点阵表示（**图像**）

- 枚举出图形中所有的点
- 简称为图像（数字图像）
- 或叫点阵图或位图图像

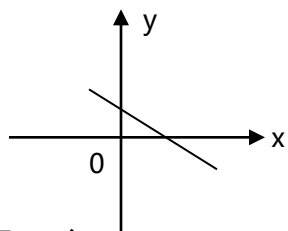
a11	a12	a13	a14	a15
a21	a22	a23	a24	a25
a31	a32	a33	a34	a35
a41	a42	a43	a44	a45
a51	a52	a53	a54	a55

灰度或颜色信息

5×5图像

—参数表示（**图形**）

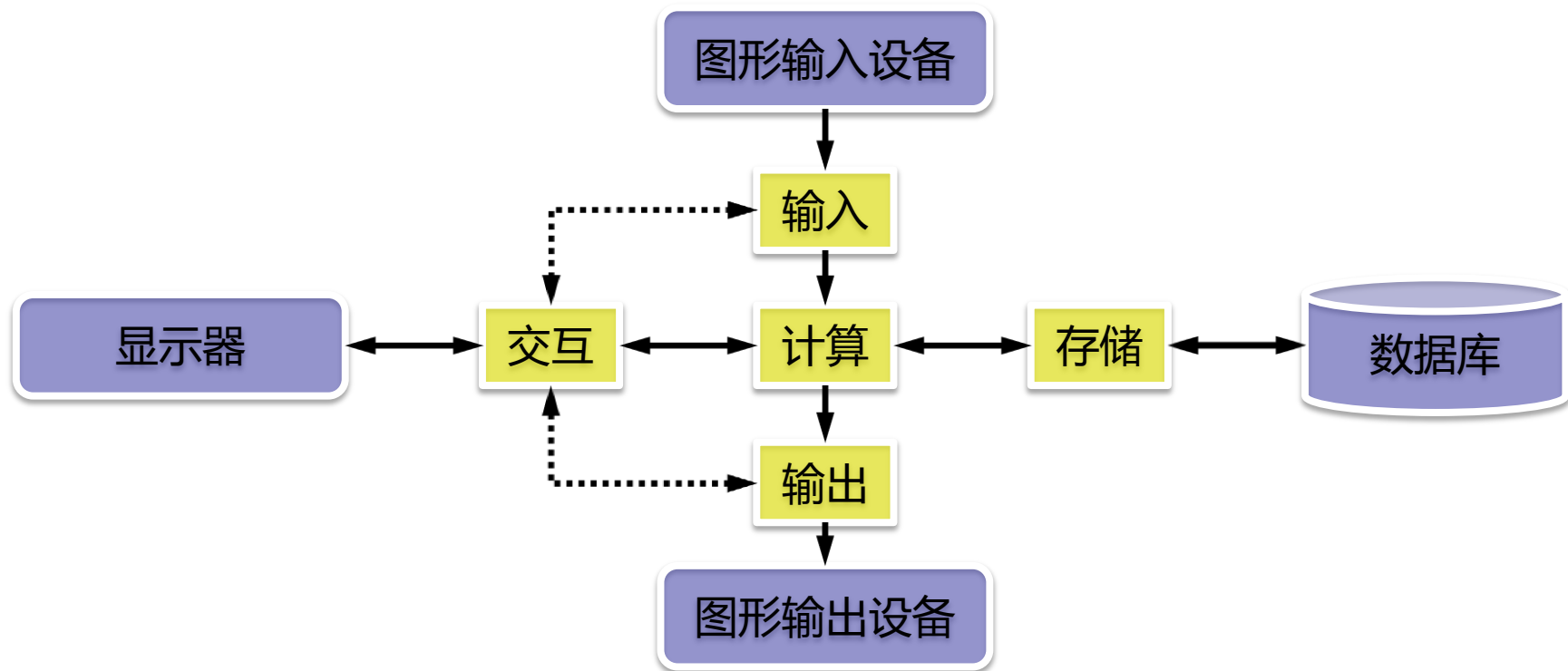
- 形状参数+属性参数
- 简称为图形
- 或叫矢量图形或参数图形



直线的参数方程：
 $ax+by+c=0$

图形系统工作过程

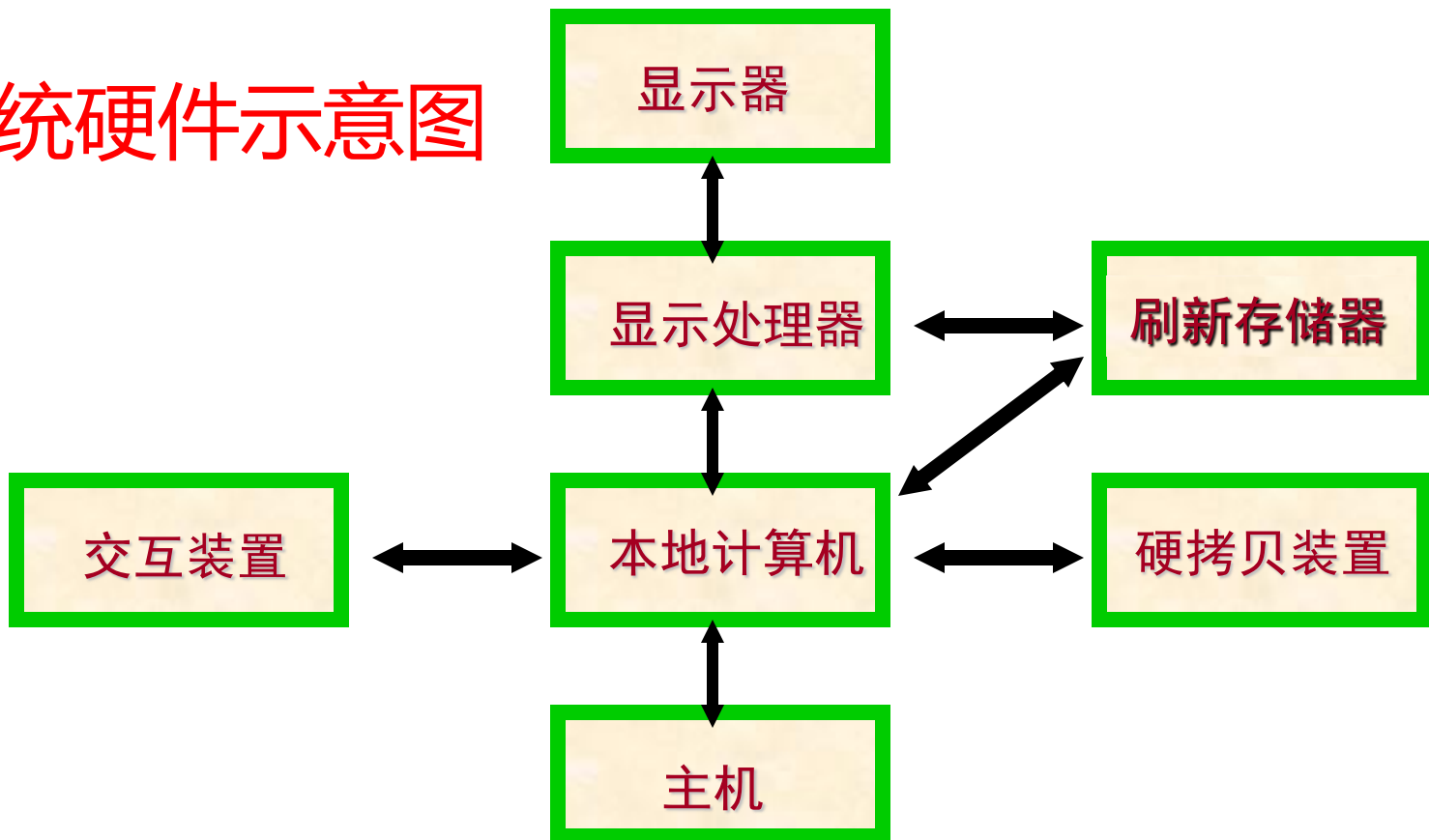
计算机图形系统是一个由软、硬相互结合的有机整体



计算机图形系统：计算机硬件+图形输入输出设备
+计算机系统软件+图形软件

图形系统组成

- 系统硬件示意图

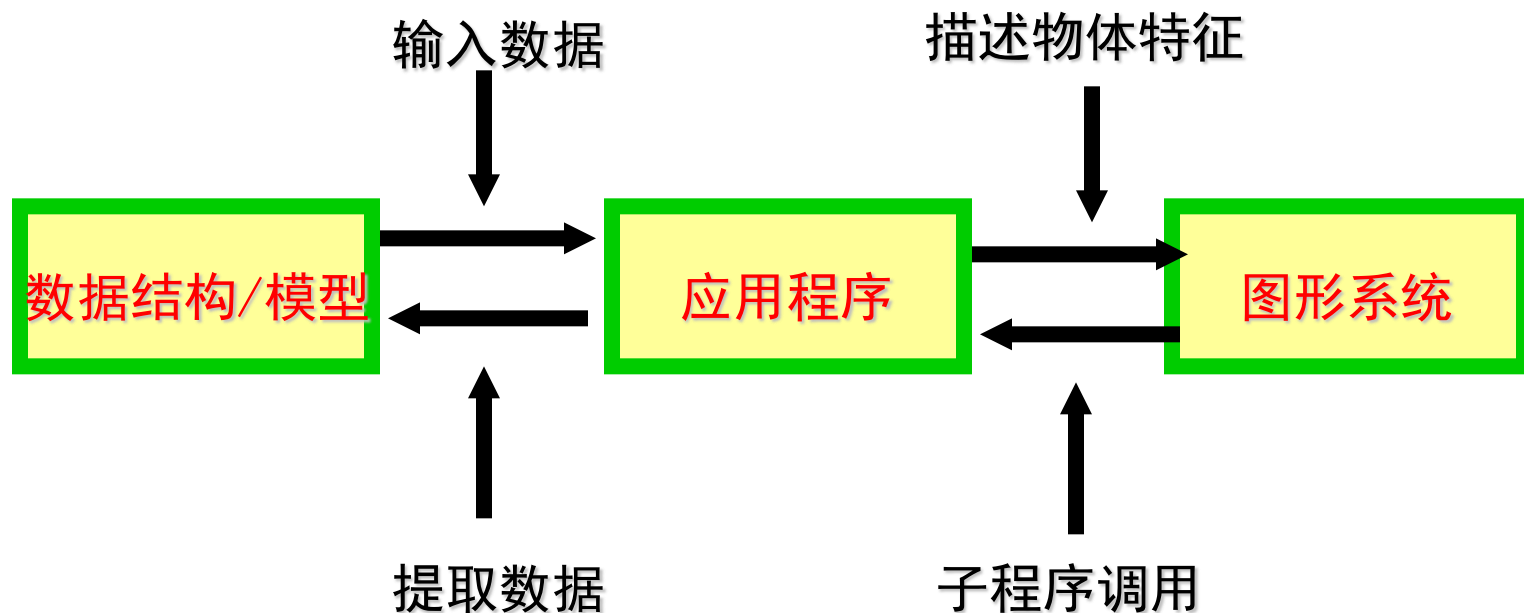


本地计算机： 生成显示文件；图形互换。

主机： 语言的解释或编译，数据库管理等。

图形系统组成

• 系统软件示意图



应用程序：需要执行的指令序列

数据结构/模型：全面描述对象的特点

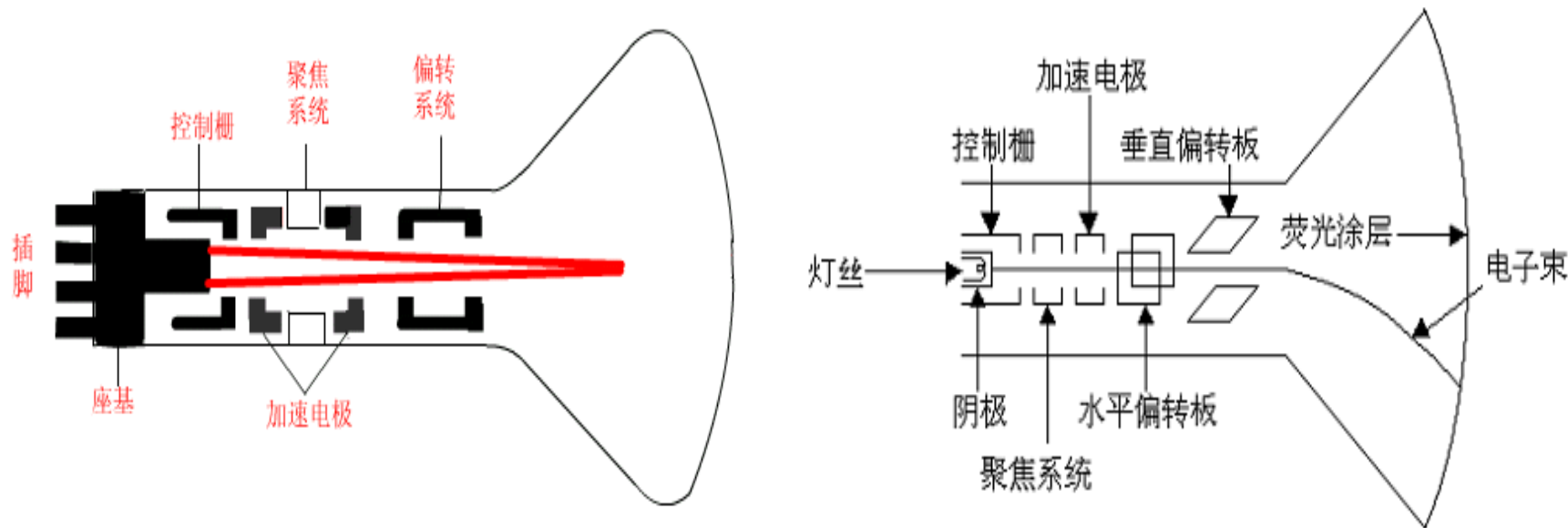
图形系统：完成实际功能，是与硬件的接口

图形输设备

- **图形显示**指的是在屏幕上输出图形，如各类显示器。
- **图形绘制**通常指把图形画在纸上，也称**硬拷贝**，打印机和绘图仪是两种最常用的硬拷贝设备。

阴极射线管

阴极射线管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。



阴极射线管

- **余辉时间**：持续发光时间，电子束离开某点后，该点的亮度值衰减到初始值。
- **刷新(Refresh)**：为了让荧光物质保持一个稳定的亮度值
- **刷新频率**：每秒钟重绘屏幕的次数

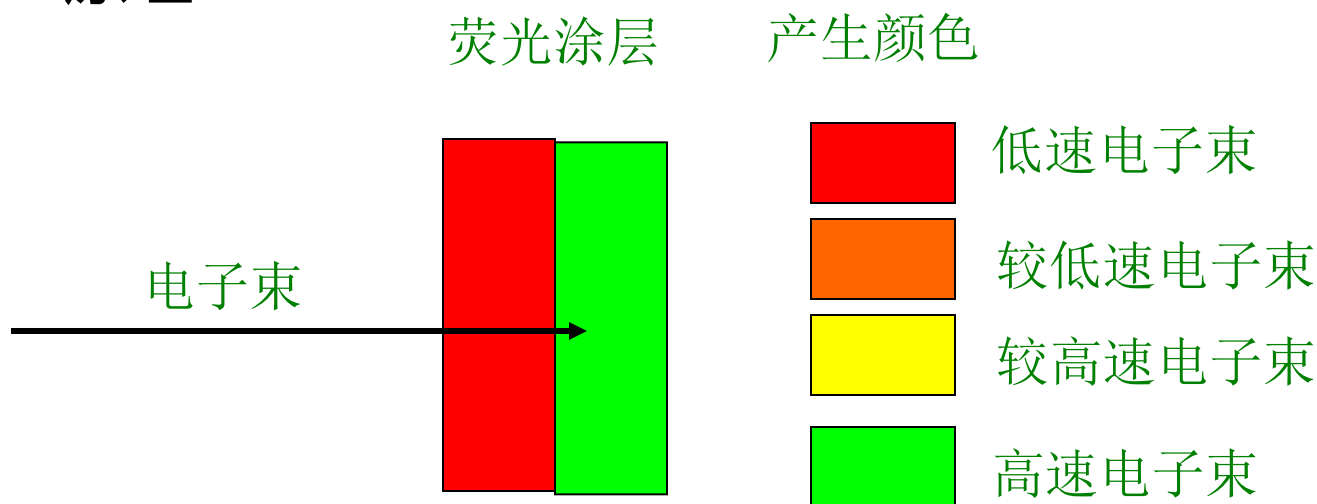
某种CRT产生稳定图像所需的最小刷新频率
= 1秒 / 荧光物质的持续发光时间
(例如) = $1000 / 40 = 25\text{Hz}$

- **像素(Pixel)**：构成屏幕（图像）的最小元素
- **分辨率(Resolution)**：CRT在水平或竖直方向单位长度上能识别的最大像素个数；单位通常为dpi (dots per inch)
如 640×480 , 800×600 , 1024×768 , 1280×1024 等

彩色阴极射线管

— 射线穿透法

- 原理

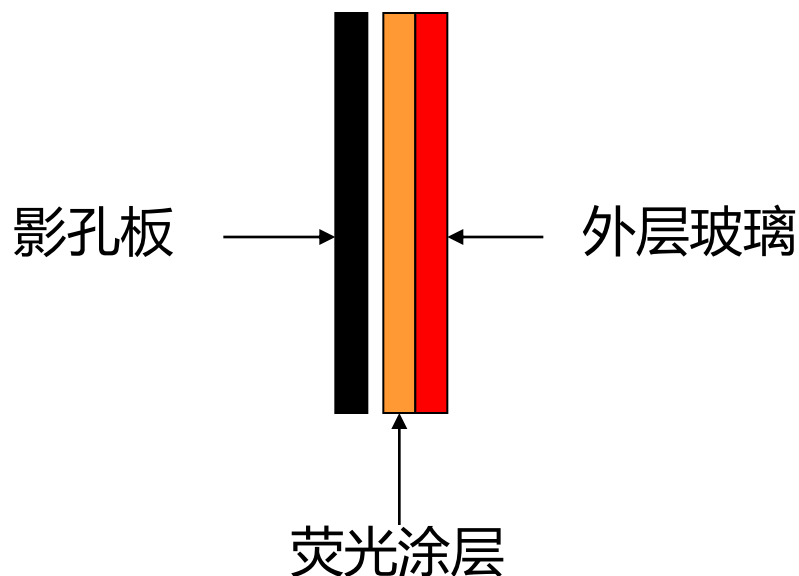


- 应用：画线显示器
- 优点：成本低
- 缺点：只能产生有限几种颜色

彩色阴极射线管

— 影孔板法

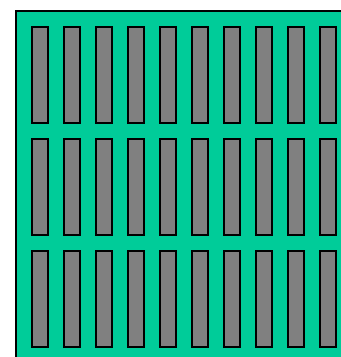
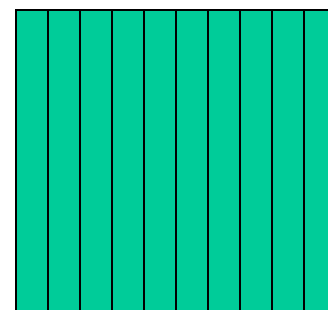
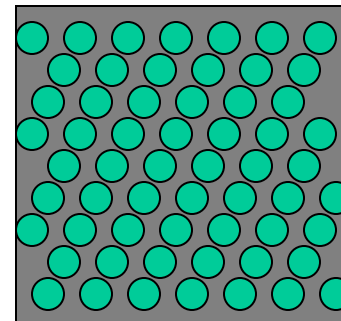
影孔板被安装在荧光屏的内表面，用于精确定位像素的位置



彩色阴极射线管

— 影孔板的类型

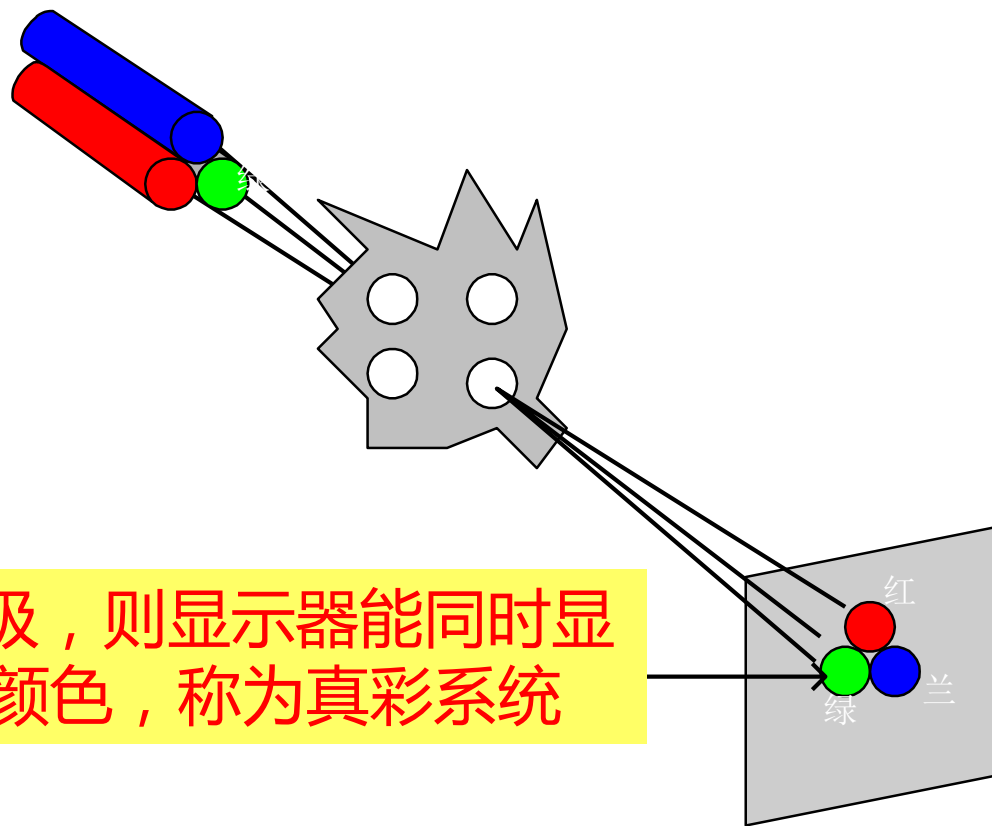
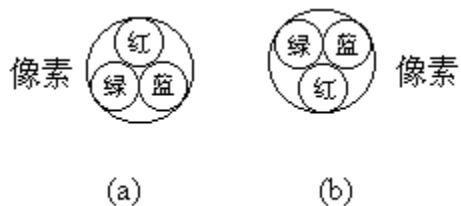
- 点状影孔板（荫罩式）
代表：球面显像管
- 栅格式影孔板（荫栅式）
代表：柱面显像管
日本索尼公司的特丽珑管（Trinitron）
三菱公司的钻石珑管（Diamondtron）
- 沟槽式影孔板
代表：LG的Flatron显像管



彩色阴极射线管

一点状影孔板（荫罩法）工作原理

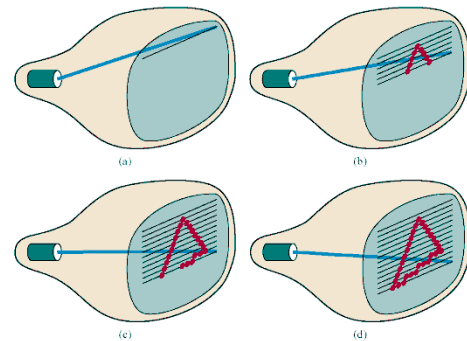
- 红、绿、兰三基色
- 三色荧光点
- 三个电子枪



如果每个电子枪有256个等级，则显示器能同时显示 $256 \times 256 \times 256 = 16\text{M}$ 种颜色，称为真彩系统

随机扫描显示器

向量 (vector) 显示器
笔划 (Stroke writing) 显示器



特点：

- **数据表示**：矢量表示，只有端点信息，无线段中间点。
- **扫描方式**：电子束像一支快速移动的画笔，可随意移动，只扫描荧屏上要显示的部分，与示波器工作原理类似。
- **显示图形**：几何属性(geometric attribute)为主，线架图
- **优点**：扫描速度快，分辨率高，线条质量好，易修改，交互性好，动态性能好。
- **缺点**：价格贵，只能显示线画图形，应用于军事、CAD领域；不能显示逼真场景、和电视标准不一致、驱动系统也较复杂。

光栅扫描显示器

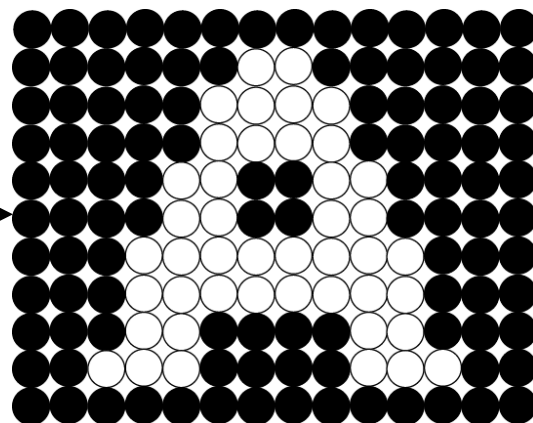
- 将屏幕分成由**像素**构成的光栅网格，其中像素具有灰度和颜色；
- 电子束按照固定的扫描线和扫描顺序**从左到右、自上而下**进行扫描；
- 若光栅扫描显示器有 N 条扫描线，每条扫描线有 M 个像素，通常 $N*M$ 称为该显示器的**分辨率**；
- **图像是由像素点阵组成**，显示一幅图像所需要的时间等于显示整个光栅所需的时间，而与图像的复杂程度无关。

光栅扫描显示器

- **帧缓冲存储器（帧缓存）**

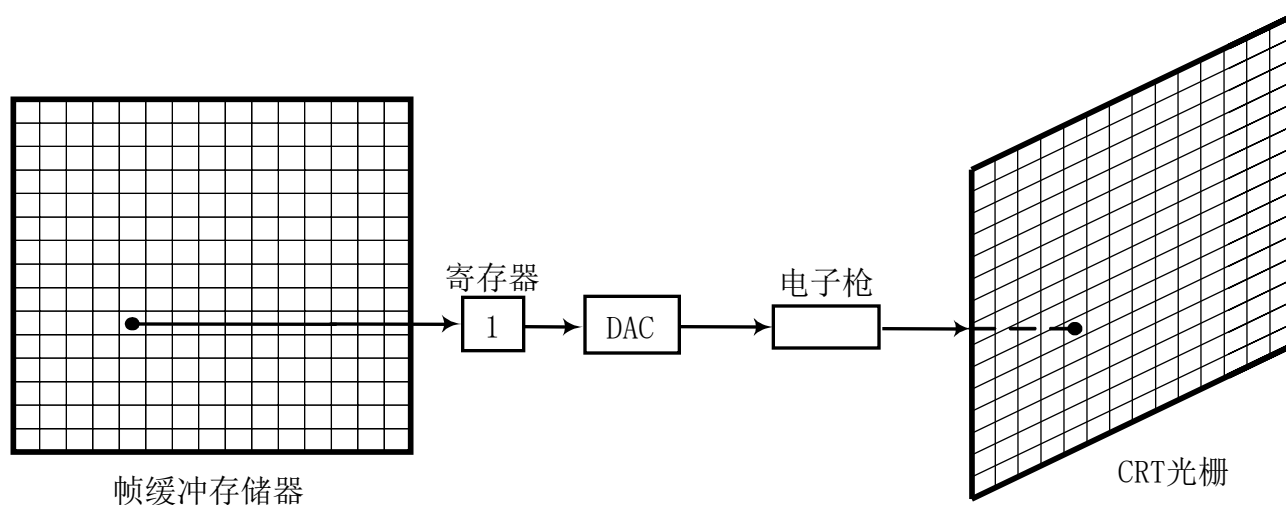
- 实现光栅CRT图形显示器的最常见方法是使用**帧缓存**；
- 帧缓冲存储器是一块连续的存储空间，光栅中的每个像素在帧缓存中至少要有1位（bit），每个像素1位的存储容量称为**位面**（bit plane），画面就是由帧缓存中的这些位信息组成的；
- 帧缓存是**数字设备**，光栅显示器是**模拟设备**，要把帧缓存中的信息在光栅显示器屏幕上输出必须经过**数字/模拟转换**，这个工作由视频控制器完成。

0000000000000000
00000011000000
00000111100000
00000111100000
00001100110000
00001100110000
00011111111000
00011111111000
00011000011000
00011000011000
00111000011100
00000000000000



光栅扫描显示器

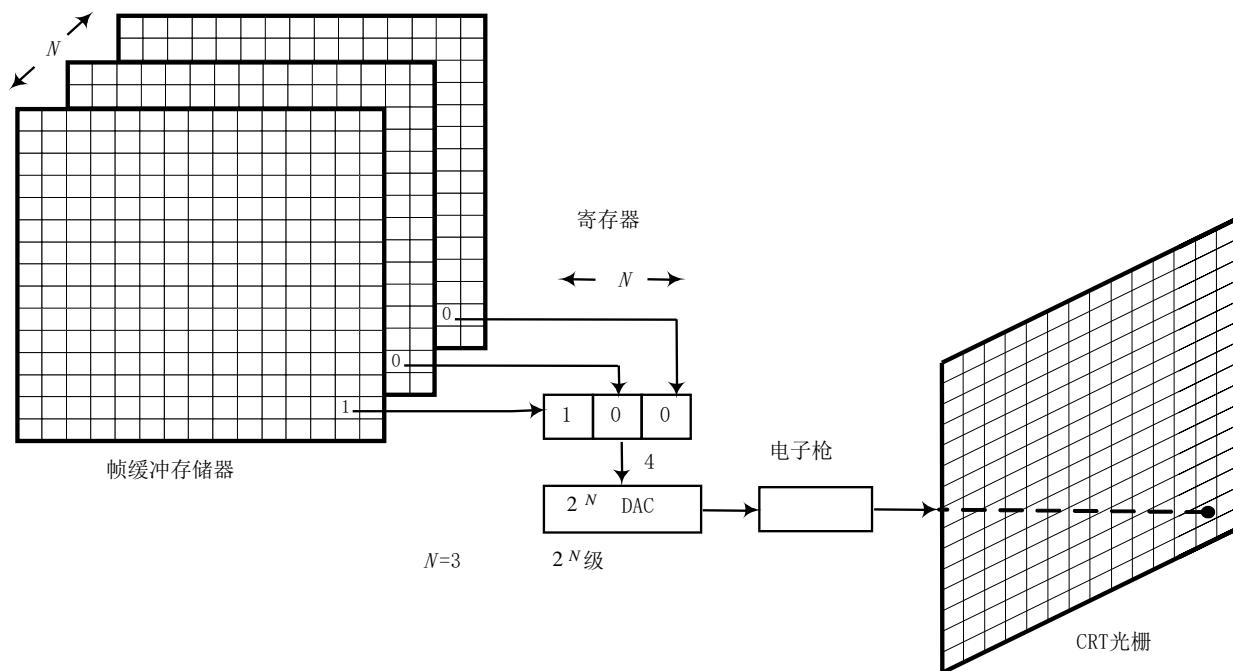
- 图形在计算机中是一位一位产生的，每一个存储位只有0或1两个状态，因此一个位面的帧缓存只能产生黑白图形；



具有1位帧缓存的黑白光栅显示器结构图

光栅扫描显示器

- 光栅图形显示器需要有足够的位面才能反映图形的颜色和灰度等级。

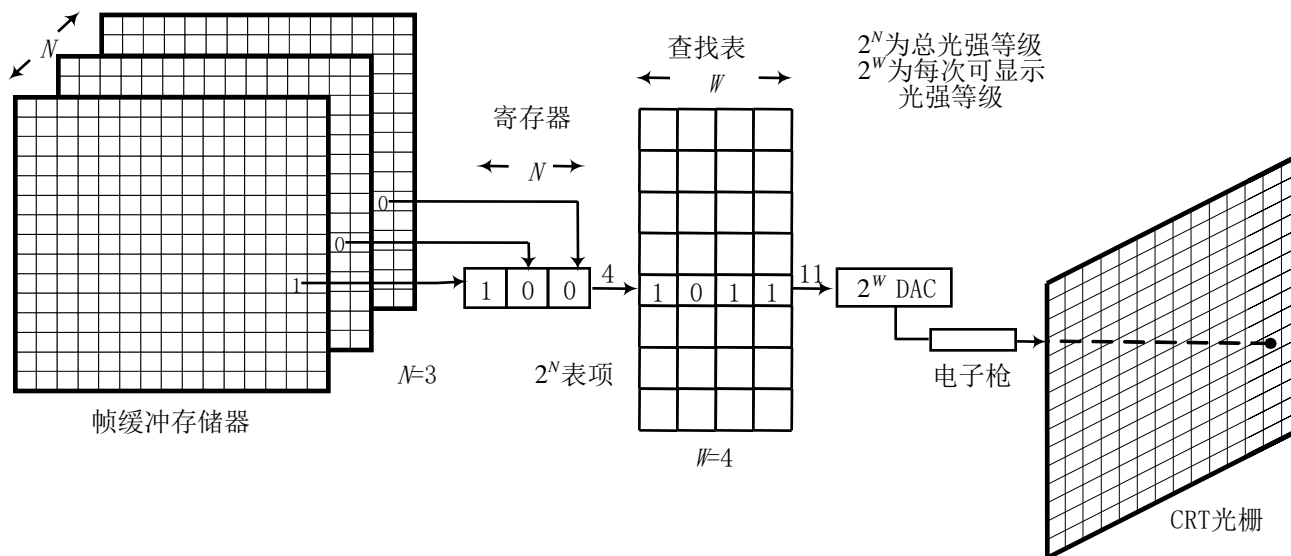


具有N位帧缓存的黑白灰度光栅显示器结构图

光栅扫描显示器

• 颜色查找表

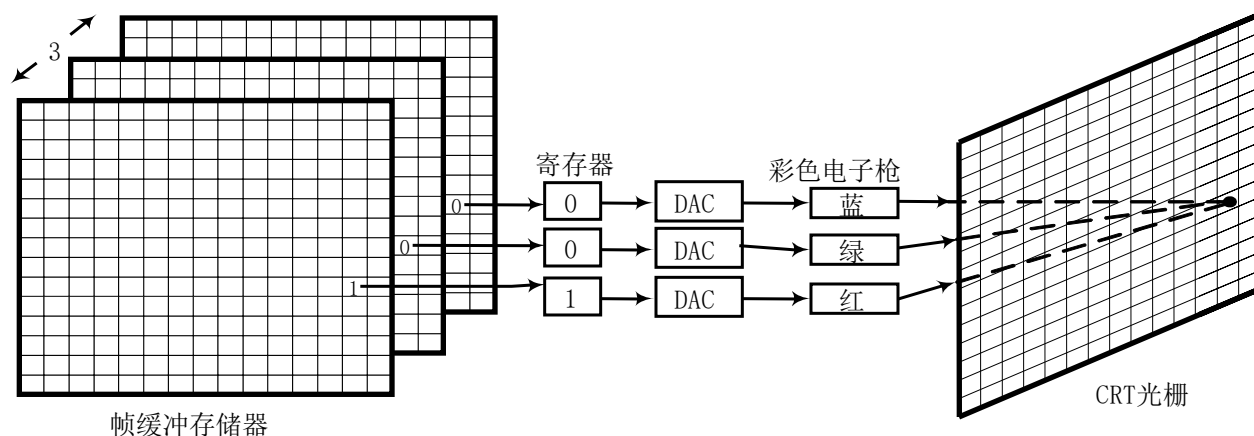
- 采用颜色查找表可以提高灰度级别，如下图所示，帧缓存中的位面号为颜色查找的索引，颜色查找表必须有 2^N 项，每一项具有 W 位字宽。



具有 N 位帧缓存和 W 位颜色查找表的光栅显示器结构图

光栅扫描显示器

- 三个位面实现一个简单的彩色帧缓冲存储器。RGB帧缓存、数模转换器(DAC)、三个电子枪和CRT光栅组成。



一个简单的彩色帧缓冲存储器

光栅扫描显示器

- **主要性能参数**

- **分辨率**

- 指显示器在屏幕水平(垂直)方向可显示多少像素
 - 分辨率越高，显示的字符或图像越清晰

- **亮度等级数目和色彩**

- 指单种颜色亮度可变化的数目
 - 亮度等级范围的提升可使图像看上去更柔和自然

- **点距**

- 指相邻像素点之间的距离（即像素的直径）

- **显示速度**

- 指显示字符、图形，特别是动态图像的速度

液晶显示器

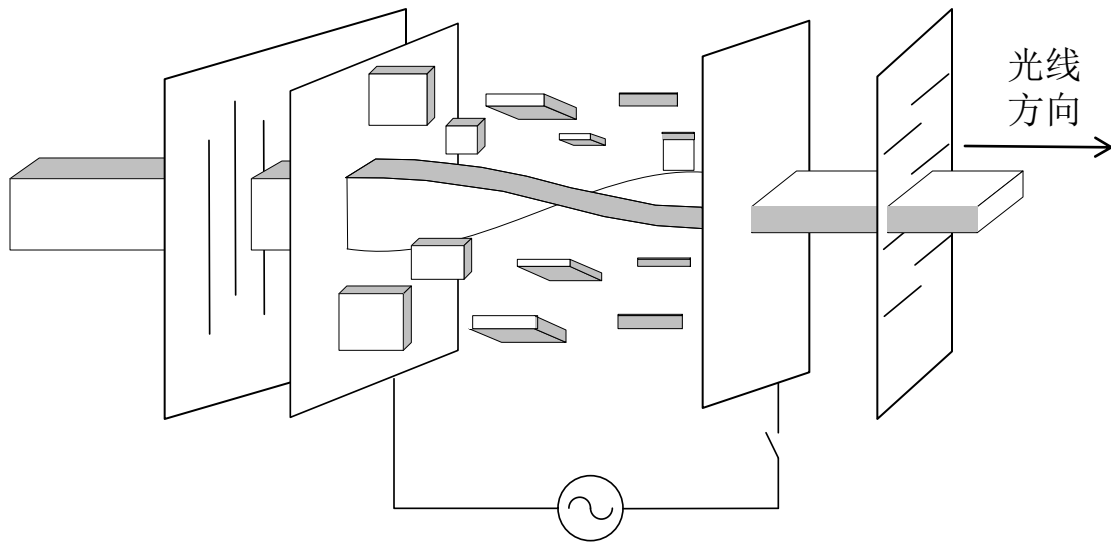
- **液晶的物理特性**

- 液晶即液态晶体，具有线状结晶结构的分子，可象液体那样流动；
- 液晶分子的排列柔软易变形，受电场、磁场、温度、应力等外部条件作用时会重新排列；
- 当通电和不通电时液晶分子处于两种不同的排列，一种排列光线容易通过，而另一种排列阻止光线通过。

液晶显示器

• 单色液晶显示器的原理

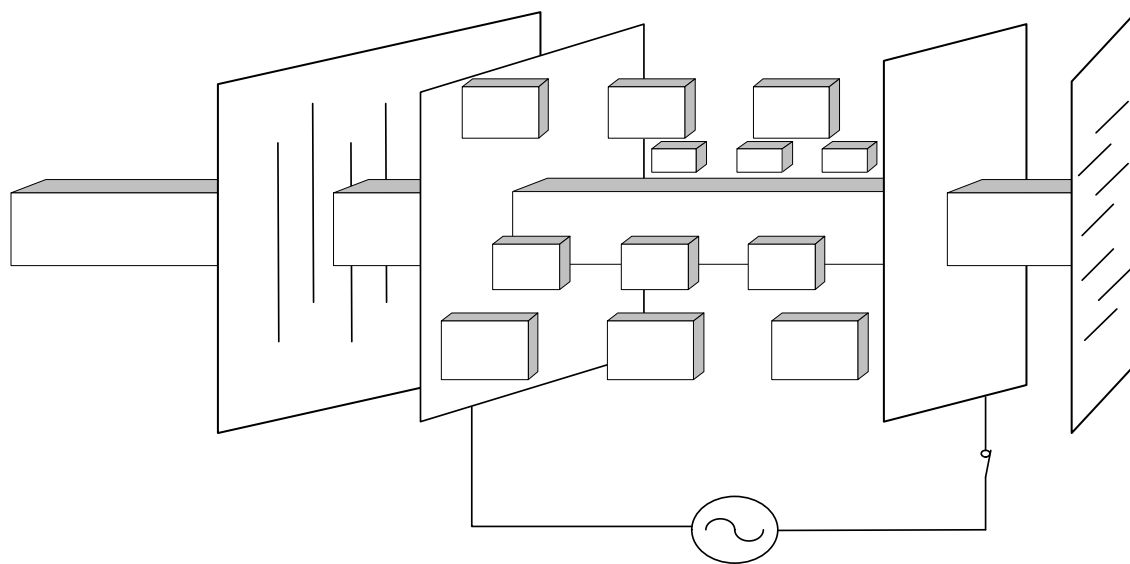
- 如果两个极化滤光器互相垂直，穿过第一个极化滤光器的光线，经过扭曲的液晶后，光线被扭曲 90° ，正好可以穿过第二个极化滤光器。



光线穿透示意图

液晶显示器

- 但若为液晶加一个电压，液晶分子又会重新排列并完全平行，使光线不再扭转，将被第二个滤光器挡住。总之，加电光线将被阻断，不加电光线将射出。



光线阻断示意图

液晶显示器

- **液晶显示器的特点**

- **优点：**

- 工作电压低、功耗小；
 - 无辐射，对人体健康无损害；
 - 完全平面，无闪烁、失真，可视面积大，又薄又轻，能大量节省空间；
 - 抗干扰能力比CRT显示器强得多。

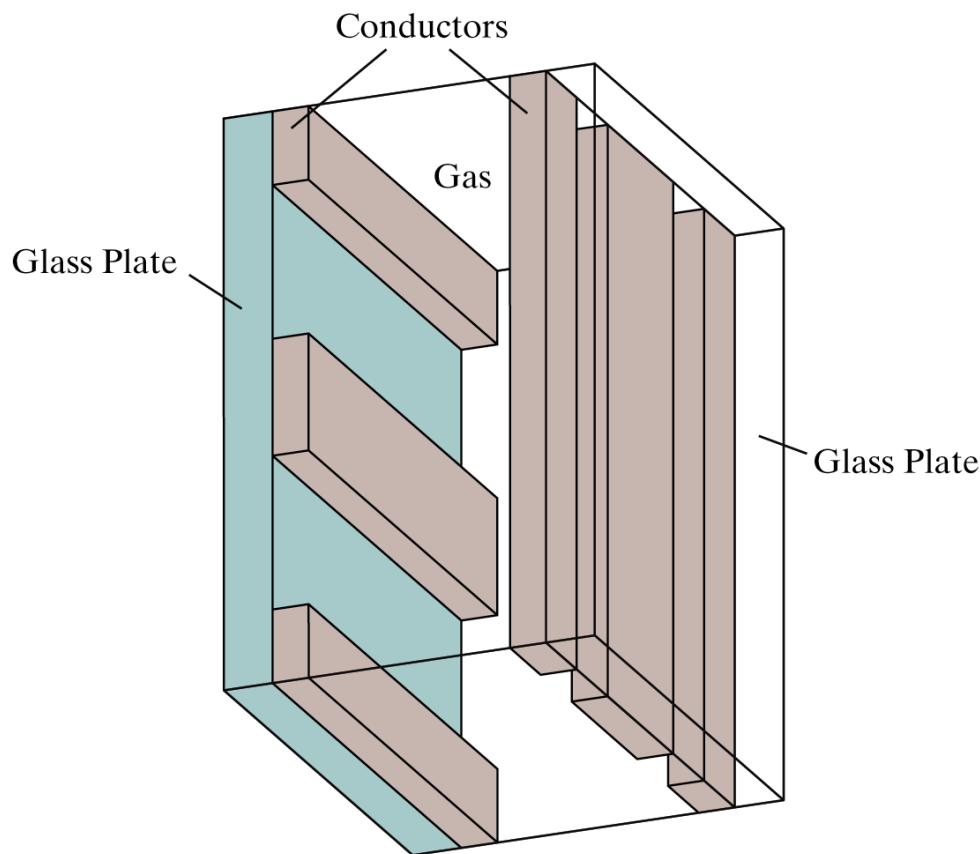
- **缺点：**

- 视角太小、亮度和对比度不够大。

等离子显示器

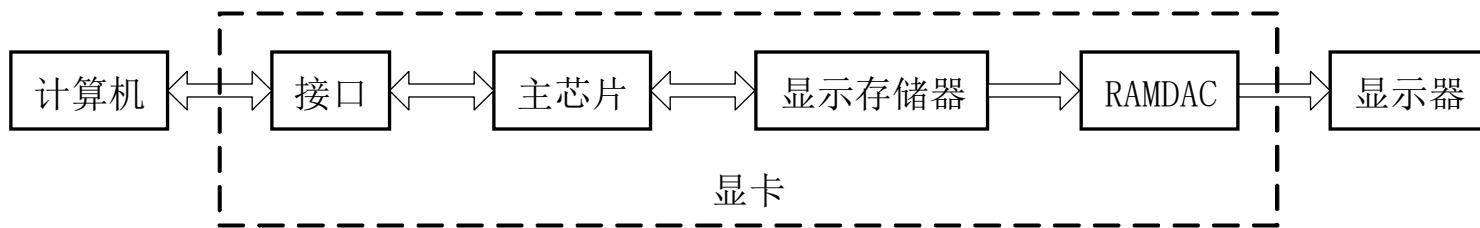
• 气体放电显示器

- 气体（氖气）混合物填充在两块玻璃板；
- 水平和垂直导电带施加点火电压，使其放电发出紫外光；
- 紫外光撞击荧光体发出可见光成像。



显卡

- 显卡根据CPU提供的指令和有关数据将程序运行过程和结果进行相应的处理、并转换成显示器能够接受的文字和图形显示信号，通过屏幕显示出来。
- 显示器必须依靠显卡提供的显示信号才能显示出各种字符和图像。
- 常见显卡的结构中包括：**显卡BIOS芯片、图形处理芯片、显存、数模转换器(RAMDAC)芯片、接口等。**



显卡工作原理图

图形输入设备

- 1、键盘 (Keyboard)
- 2、鼠标 (Mouse)
- 3、光笔 (Light Pen)
- 4、触摸屏(Touch Screen)
- 5、操纵杆(Joystick)
- 6、跟踪球(Trackball)和空间球(Spaceball)
- 7、数据手套(Data Glove)
- 8、数字化仪(Digitizer)
- 9、图像扫描仪(Scanner)
- 10、音频输入系统
- 11、视频输入系统

光笔 (Light Pen)

- 一种检测装置，靠检测荧光屏上的发光点来选择屏幕的位置坐标。
- 功能：定位、拾取、笔划跟踪。

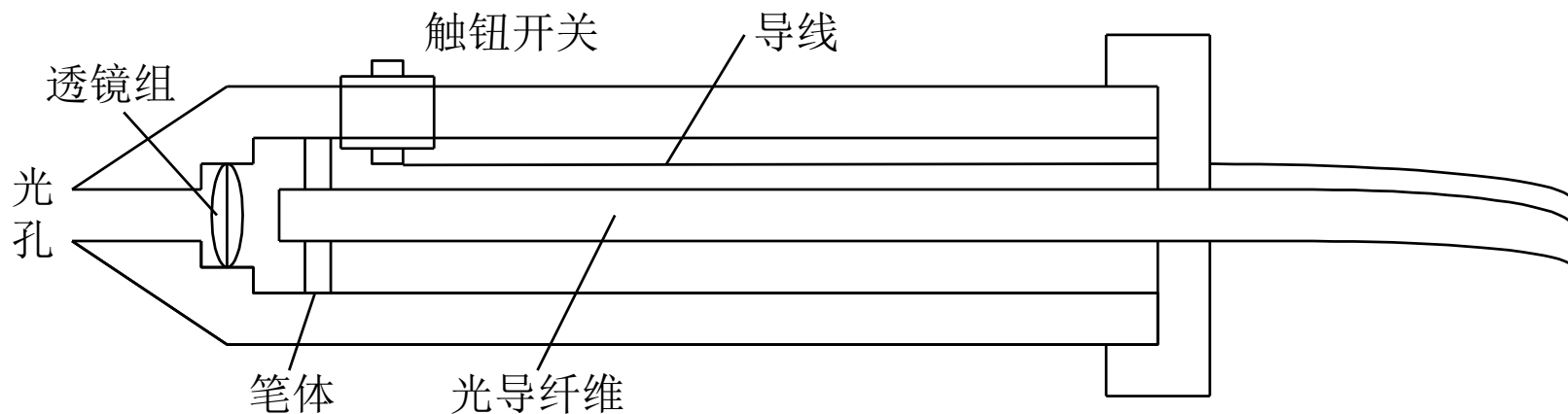


图2-2 光笔的结构

触摸屏（ Touch Screen ）

- 以手指触摸的方式选择屏幕位置。



红外线式触摸屏

光学

电阻式触摸屏

电子

电容式触摸屏

电子

声波（声音探测式）触摸屏

声音

操纵杆 (Joystick)

- 由一根小的垂直杠杆组成的可摇动装置，装配在四周可移动的底座上用来控制屏幕光标。



另：压力检测操纵杆
手柄不可移动，
压力 -> 位移

跟踪球&空间球

- 跟踪球：二维



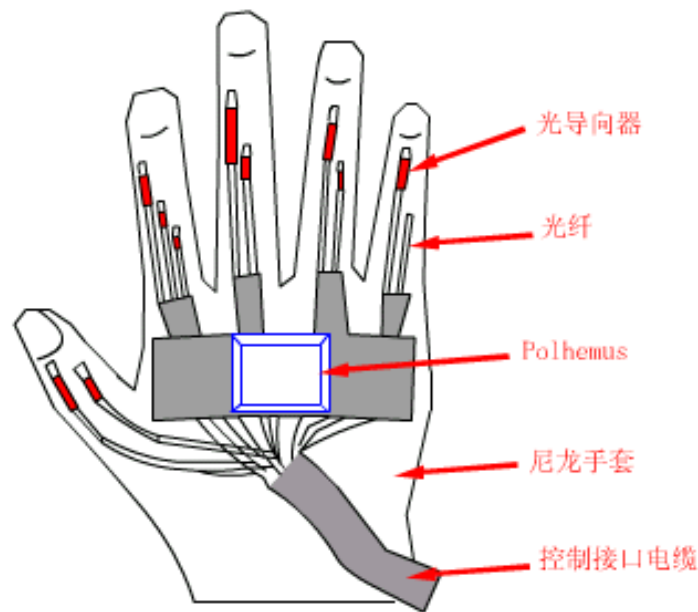
- 空间球：三维



数据手套 (Data Glove)

戴在手上的传感器，可以用来抓住“虚拟对象”

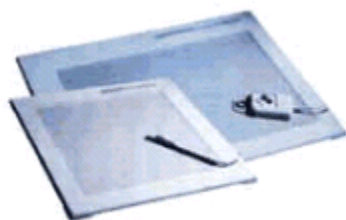
- 可测量出手的位置和形状，从而实现环境中的虚拟手及其对虚拟物体的操纵。
- 数据手套通过手指上的弯曲、扭曲传感器和手掌上的弯度、弧度传感器，确定手及关节的位置和方向。



数字化仪（Digitizer）

把图形变成一种计算机能接收的数字形式的专用设备。

- **工作原理**：电磁感应技术；
- **性能指标**：最大有效幅面；数字化速率；最高分辨率。
- **工作方式**：点方式、连续方式（流方式）、相对坐标方式



图象扫描仪 (Scanner)

用来在所作的画或对象上扫描，并输入一组坐标位置，相互以直线段连接，以逼近曲线或表面形状。

分类

支持的颜色： 单色 彩色

固态器件： 电荷耦合 MOS电路 紧贴型

扫描宽度和操作方式： 大型 台式 手动

评价指标

幅面： A0 A1 A4

分辨率： 300dpi~1000dpi (Dot per Inch)

支持颜色 (灰度等级)：4、8、24位面颜色



大型 (滚筒式)



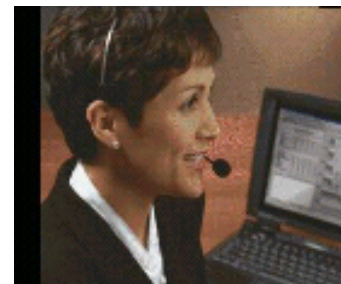
台式



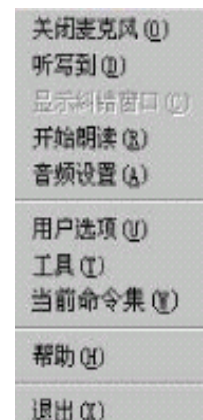
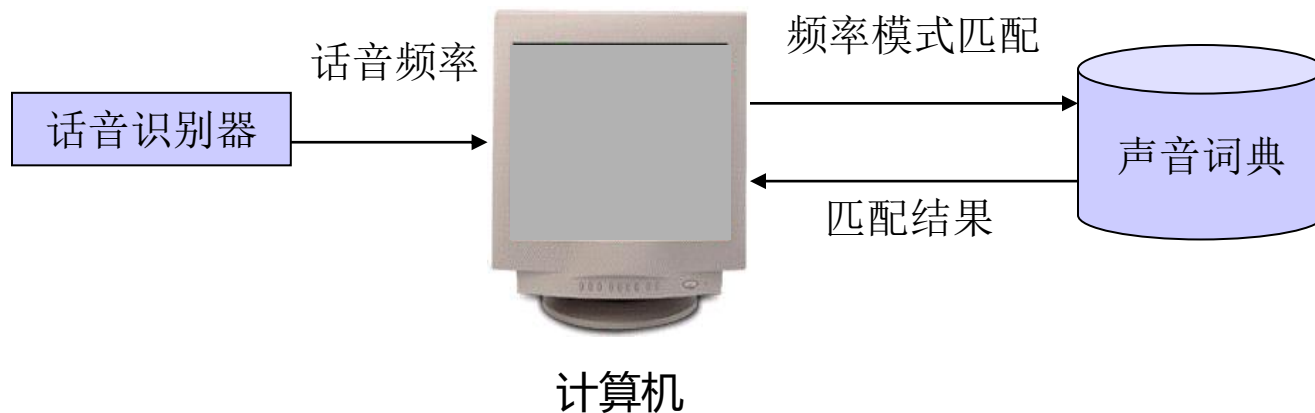
手动

音频输入系统

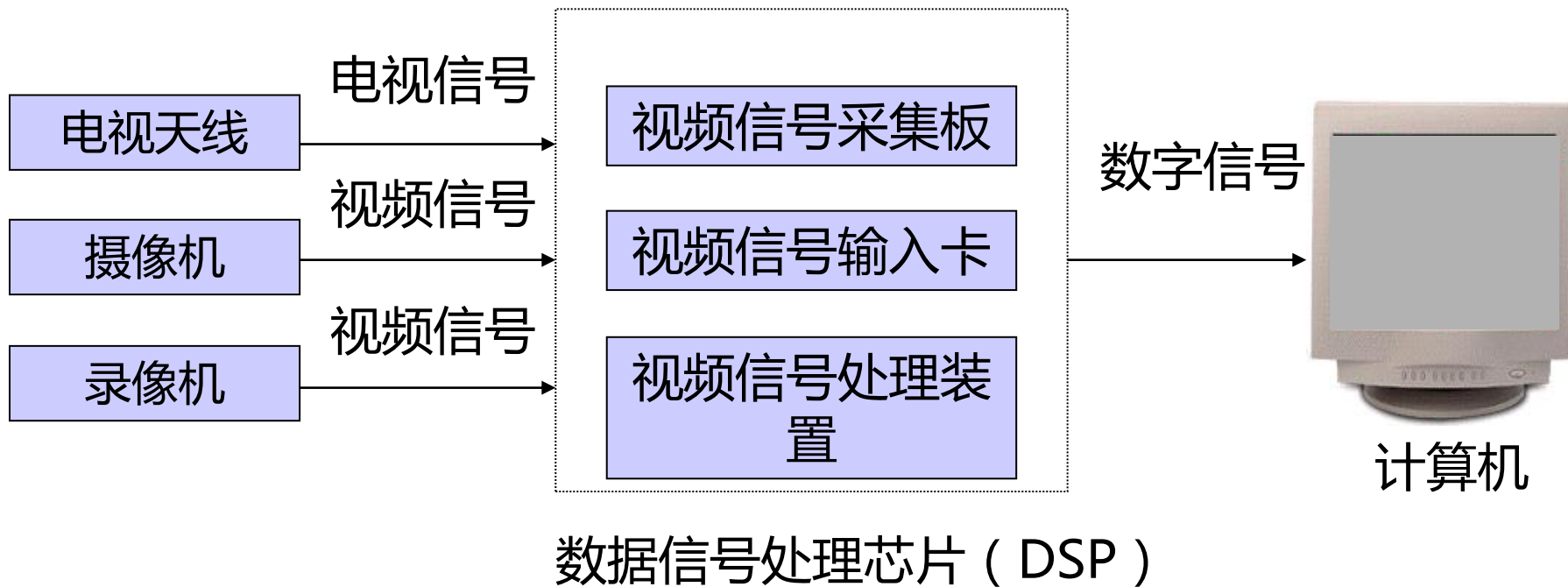
采用话音识别器作为输入设备，以接收操作者的命令。



例：IBM公司的Viavoice语音识别系统



视频输入系统



其他图形输入设备

■ 用于虚拟现实环境的显示器类型

- 头盔式显示器
- 空间沉浸式显示器

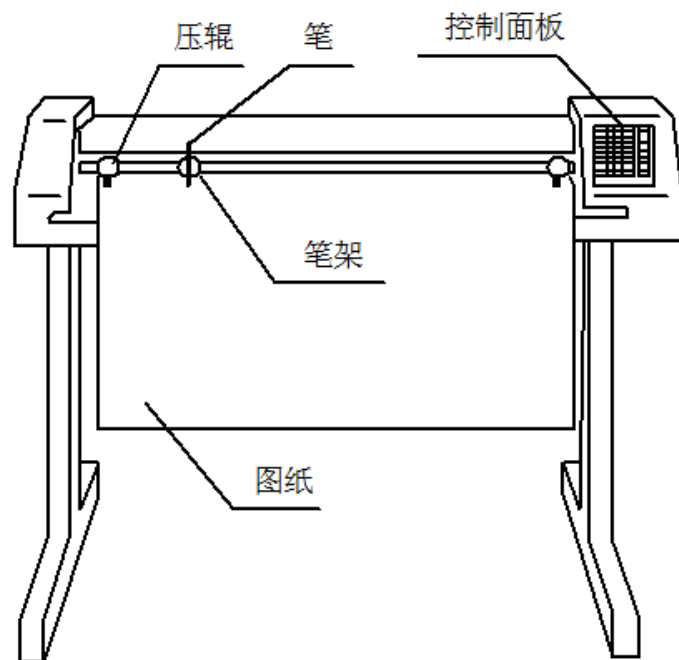
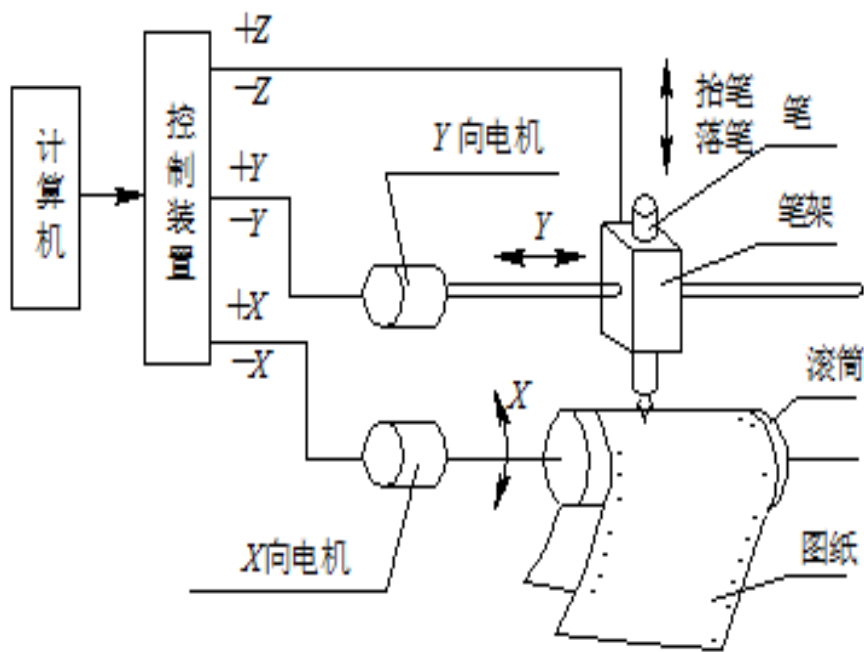


■ 头盔式显示器

- 将观察者的**头部位置及运动方向告诉计算机**，计算机就可以调整观察者所看到的图景，使得呈现图像更趋于真实感绝大多数头盔式显示器使用两个显示器；
- 利用特殊光学设备对图像进行处理，使**图像看上去立体感更强**；
- 把用户的**视觉、听觉和其他感觉封装**起来，产生一种身在虚拟环境中的错觉。

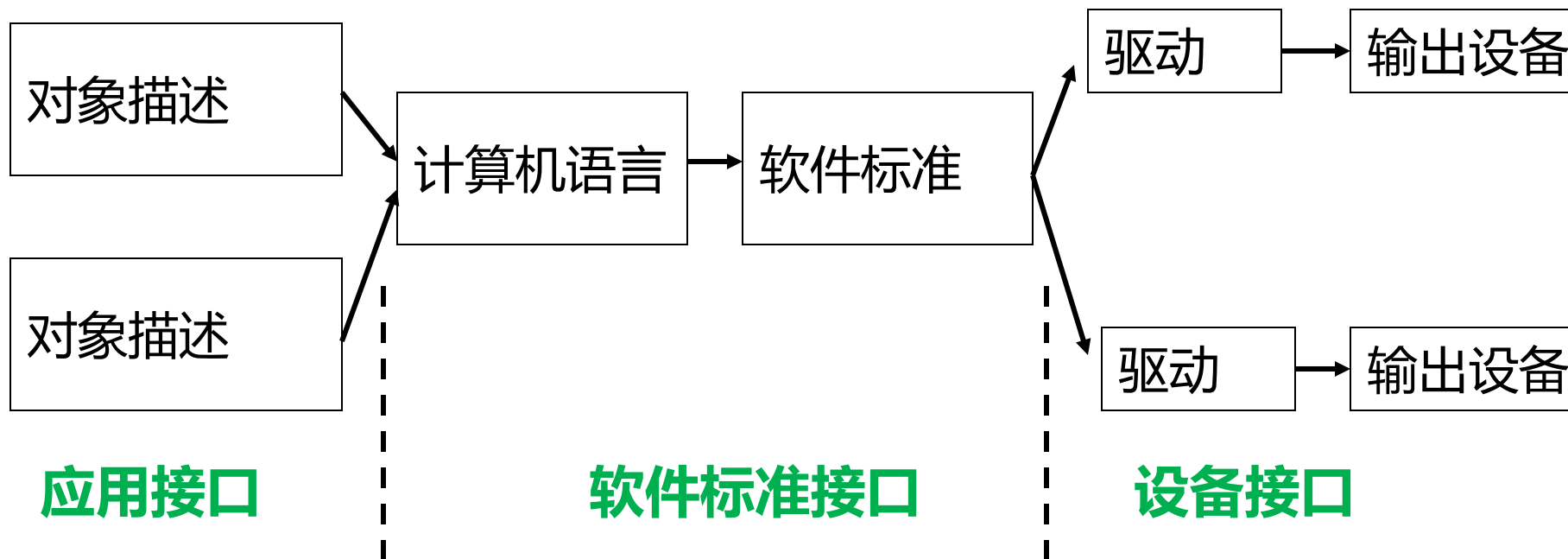
硬拷贝设备

- **打印机**：点阵针式打印机、喷墨打印机、激光打印机、热敏打印机、热转印打印机；
- **绘图仪**



图形软件标准

- GKS(Graphical Kernel System) : ISO
- PHIGS(Programmer' s Hierarchical Interactive Graphics Standard): GKS扩充
- OpenGL



图形软件类型

■ 通用软件包

用现有某种计算机语言写成的子程序包。如：GKS、OpenGL.....

■ 基于通用语言的扩展图形软件

扩充某种计算机语言使其具有图形生成和处理功能，例如：Fortran、Pascal、Basic (Visual Basic)、C、C++ (Visual C++)、AutoLisp.....

■ 专用图形软件包

针对某一种设备或应用，设计/配置专用的图形生成语言或函数集，例如：场景描述：Open Inventor；建立虚拟世界的三维模型：VRML；生成三维Web显示：Java3D；创建Java applet中的二维场景：Java 2D；光照模型下的场景：Renderman Interface (Pixar)

.....

OpenGL简介

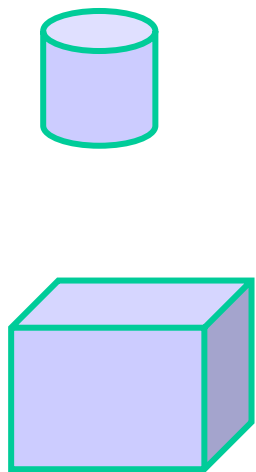
OpenGL是一个开放的三维图形软件包，它独立于窗口系统和操作系统

- 1、建模：**提供基本的点、线、多边形以及复杂的三维物体（球、锥、多面体、茶壶等）以及复杂曲线和曲面绘制函数。
- 2、变换：**基本变换和投影变换。
- 3、颜色模式设置：**RGBA模式和颜色索引。
- 4、光照和材质设置：**辐射光、环境光、漫反射光和镜面光。
- 5、纹理映射**
- 6、位图显示和图象增强图象：**图像拷贝和像素读写，以及融合、反走样和雾的特殊图象效果处理。
- 7、双缓存动画：**后台缓存计算场景、生成画面，前台缓存显示后台缓存已画好的画面。

坐标系统

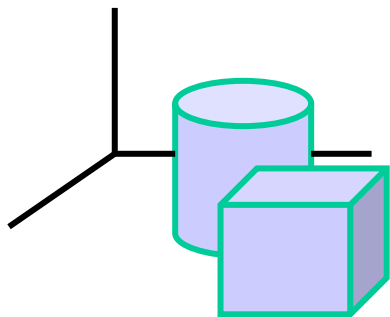
- 为描述对象、构造场景或完成图形变换，需要不同的坐标系
 - 模型坐标系：定义对象
 - 世界坐标系：定义对象与外界环境的关系
 - 设备坐标系：定义图形显示的位置、大小
 - 规范化坐标：为保证互换性（与设备无关）而定义的辅助坐标

建模坐标



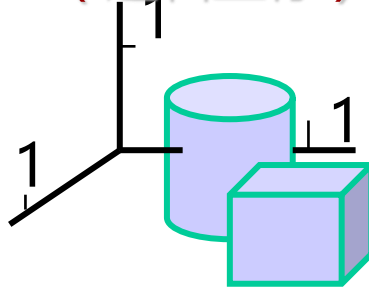
Modeling Coordinate
Local Coordinate
Master Coordinate

世界坐标



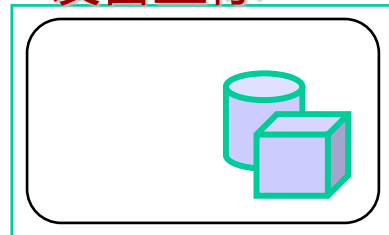
World Coordinate

规范化坐标
(逻辑坐标)



Normalized
Coordinate

设备坐标



绘图仪

其它输出设备

Device Coordinate
Screen Coordinate

VC++坐标系统

- **窗口**：规范化坐标（逻辑坐标系）上用户设置的一个区域。
- **视口**：设备坐标系（实际输出设备）上用户设定的一个区域。
- **相关函数**

SetWindowOrg(HDC hdc,int x,int y) //设置窗口原点
SetViewportOrg(HDC hdc,int x,int y) //设置视口原点

SetWindowExt(HDC hdc,int nHeight,int nWidth)
//设置窗口区域，用来设置比例
SetViewportExt(HDC hdc,int nHeight,int nWidth)
//设置视口区域，用来设置比例

VC++坐标系统

- **映射模式**：定义了将逻辑坐标转为设备坐标的度量单位以及设备的x、y方向。

- **相关函数**：

SetMapMode(hdc,nMapMode); //设置映像模式

nMapMode=GetMapMode(hdc); //得到映像模式

- **映射参数**：

映像模式 将一个逻辑坐标映像为 **坐标系设定**

- **MM_ANISOTROPIC** **XY比例可异** x向右y向下
- MM_ISOTROPIC XY比例相同 x向右y向下
- MM_HIENGLISH 0.001英寸 x向右y向上
- MM_HIMETRIC 0.01毫米 x向右y向上
- MM_LOENGLISH 0.01英寸 x向右y向上
- MM_LOMETRIC 0.1毫米 x向右y向上
- MM_TEXT 1个像素 x向右y向下
- MM_TWIPS 1/1440英寸 x向右y向上

VC++坐标系统

- 绘图方法

- 1、构建画笔或画刷

```
CPen( int nPenStyle, int nWidth, COLORREF crColor );  
CBrush( COLORREF crColor );
```

- 2、选入画笔：将该画笔选入到设备环境中

```
CPen * SelectObject(CPen * pPen)  
CBrush * SelectObject(CBrush * pBrush)
```

- 3、获取设备环境

- 使用GetDC()函数

```
CDC *CWnd GetDC() ; int ReleaseDC(CDC * pDC )
```

- 直接构造CDC对象 CClientDC dc(CWnd *)

- 4、设置映射模式

```
CDC::SetMapMode ( MM_TEXT)
```

- 5、调用绘图函数

```
CDC::Ellipse(100,100,500,300)
```

VC++绘图演示

- **MM_TEXT模式绘图**
- **MM_ANISOTROPIC模式绘图**
 - pDC->SetMapMode(MM_ANISOTROPIC);
 - pDC->SetWindowOrg(100,100);等价pDC->SetViewportOrg(-100,-100); ?
 - pDC->Rectangle(100,100,500,500);
 - pDC->SetWindowExt(1,1); pDC->SetViewportExt(2,4);
 - pDC->SetWindowExt(100,100); pDC->SetViewportExt(200,400);
 - **两组效果相同 ?**
- **MM_ISOTROPIC模式绘图**
 - pDC->SetWindowExt(1,1); pDC->SetViewportExt(3,2);
 - pDC->SetWindowExt(1,1); pDC->SetViewportExt(2,3);
 - **XY方向比例不同 ? 两组效果相同 ?**

VC与GIS中绘图函数

- **1、获取设备环境函数**

- **VC:**

- `CDC *CWnd GetDC() ; int ReleaseDC(CDC * pDC)`

- **MapGIS :**

- `MyDC _GetMyDC(); _DeleteMyDC(MyDC *pMyDC)`

- **2、设置映射模式**

- **VC:**

- `void DPtoLP(LPPOINT lpPoints) const;`

- `void LPtoDP(LPPOINT lpPoints) const;`

- **MapGIS :**

- `void _LpToDp(float x, float y, int *pfx, int *pfy);`

- `void _DpToLp(int x, int y, double *pfx, double *pfy);`

- **3、调用绘图函数**

- **VC:**

- `Rectangle(int x1, int y1, int x2, int y2);`

- **MapGIS :**

- `_Rect(MyDC mdc, double x1, double y1, double x2, double y2);`

VC++ 绘图抽象方法

- 1、声明基类

```
class CBaseTool
{
public:
    CBaseTool(CView * ownerObj);
    virtual ~CBaseTool();
public:
    //鼠标消息，处理多种鼠标消息
    virtual short ProcMouse(UINT msg,CPoint * xy,UINT keyFlags){return 1;};
    //键盘消息:ProcKey在KeyDown中处理
    virtual short ProcKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags){return 1;};
    //在KeyUp中处理
    virtual short ProcKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags){return 1;};
    //处理重画消息
    virtual void ProcPaint(CDC *pDC){};
    virtual short Running(){return 1;};
};
```

VC++ 绘图抽象方法

- 2、声明派生类

```
class CRectTool: public CBaseTool
{public:
    CMyTool(CView * ownerObj);
    virtual ~CMyTool();
public:
    virtual short ProcMouse(UINT msg,CPoint * xy,UINT keyFlags);
    //处理重画消息
    virtual void ProcPaint(CDC *pDC);
    virtual short Running();
private:
    CView * m_pView;
};
short CRectTool ::ProcMouse(UINT msg,CPoint * xy,UINT keyFlags)
{
    switch(msg)
    {
        case WM_LBUTTONDOWN: ..... break;
        case WM_RBUTTONDOWN: ..... break;
        default: return 1;
    }
}
```

VC++ 绘图抽象方法

- 3、修改CView类

```
class CDemoToolView : public CView
{public:
    void SetTool(CBaseTool *pTool);
    CBaseTool * GetCurTool(){ return m_pTool;}
protected:
    CBaseTool *m_pTool;
...
}
void CDemoToolView::SetTool(CBaseTool *pTool)
{    if(m_pTool)
        delete m_pTool;
    m_pTool = pTool;
}
```

在消息响应函数中调用 CBaseTool:: ProcMouse(),把消息传递到 CBaseTool对象中

VC++ 绘图抽象方法

- 4、绘制图形工具调用

//响应菜单命令消息

```
void CMainFrame::OnMenuItemRectangle()
{
    CDemoToolView* pView = (CDemoToolView *)GetCurActiveView();
    if(!pView)
        return;

    CMyTool* pTool = new CMyTool(pView);

    if(!pTool)
        return;
    pView->SetTool(pTool);

    if(pView->GetCurTool()->Running() < 1)
        return;
}
```

思考

1、试解释 VC 中函数 SetMapMode , SetWindowExtEx SetViewportExtEx , SetWindowOrgEx 的功能，这些功能是为了达到什么目的？

2、在 MapGIS 的二次开发函数中有输入点的函数 _Append(..., double x, double y...) , VC 的 MFC 中有输入点函数 SetPixel(..., int x, int y,...) , 为何 _Append 不直接由 SetPixel 代替？

谢 谢 !