图像处理编程中常遇的一些问题及编程技巧

1. 在一个按钮上绘制一个带有填充色的矩形框

```
RECT rect;
CButton* pButton=(CButton*)GetDlgItem(ID_YOURBUTTON);
pButton->GetClientRect(&rect);//获得按钮的矩形区域
CDC* pDC=pButton->GetDC();//使用按钮的设备上下文
CBrush* pRedBrush=new CBrush;
pRedBrush->CreateSolidBrush(RGB(255,0,0));
CGdiObject* pOldBrush=pDC->SelectObject(pRedBrush);
pDC->FillRectangle(&rect);
pDC->SelectObject(pOldBrush);
delete pRedBrush;
```

2. 用 C++实现把一个十进制数转换为一个十六进制数

```
//将十进制整数转换成十六进制整数
int nDecNum = 100;//需要转换的数值
int nHexNum;
nHexNum = nDecNum/10*16 + nDecNum%10;
//将十进制整数转换成十六进制字符
int nDecNum = 100;
CString strHex;
strHex.Format("%x",nDecNum); //转换后的字符串为 strHex.
```

3. 通过对话框加载一幅上面有不同标志的图片，通过鼠标用直线连接两个任意不同的标志并实时的清除

在 OnPaint 里画图，在 mouse 消息处理函数里画线，要清除的地方 call 一下 invalidate。

4. 修改文档界面的默认背景色

Windows 向窗口发送一个 WM_ERASEBKGND 消息通知该窗口擦除背景，可以使用 ClassWizard 重载该消息的缺省处理程序来擦除背景（实际是画），并返回 TRUE 以防止 Windows 擦除窗口。具体程序如下：

```
BOOL CSampleView : : OnEraseBkgnd (CDC* pDC)
{
// Create a pruple brush.
CBrush Brush (RGB (128, 0 , 128) )
// Select the brush into the device context .
CBrush* pOldBrush = pDC—>SelcetObject (&brush)
// Get the area that needs to be erased .
CRect reClip
pDC—>GetCilpBox (&rcClip)
//Paint the area.
pDC—> PatBlt (rcClip.left , rcClip.top , rcClip.Width ( ) , rcClip.Height( ) , PATCOPY )
//Unselect brush out of device context .
```

```
pDC—>SelectObject (pOldBrush )
// Return nonzero to half fruther processing .
return TRUE
}
```

5. 将子窗口置于父窗口的中心位置

首先，调用父窗口的 GetWindowRect 函数，得到它在屏幕上的位置及大小；然后调用子窗口的 GetWindowRect 函数，得到它的大小；最后计算出子窗口位置，调用子窗口的 MoveWindow 函数。

6. 在已知二维数组的情况下，将点连成光滑的曲线

采用 CDC 中的贝塞尔曲线函数，BOOL PolyBezier(const POINT* lpPoints, int nCount);

7. 在处理基于视频的时间序列图像时，除了采用建立线程的方法外，还可使用定时器的方法

设置定时器：

首先告诉 WINDOWS 一个时间间隔，然后 WINDOWS 以此时间间隔周期性触发程序。通常有两种方法来实现：发送 WM_TIMER 消息和调用应用程序定义的回调函数。

1) 用 WM_TIMER 来设置定时器

```
UINT_PTR SetTimer(
HWND hWnd, //  窗口句柄
UINT_PTR nIDEvent, //  定时器 ID，多个定时器时，可以通过该 ID 判断是哪个定时器
UINT uElapse,    //  时间间隔,单位为毫秒
TIMERPROC lpTimerFunc    //  回调函数
);
```

例如: SetTimer(m_hWnd,1,1000,NULL); //一个 1 秒触发一次的定时器。

在 MFC 程序中 SetTimer 被封装在 CWnd 类中，调用就不用指定窗口句柄了，例如:

```
UINT SetTimer(1,100,NULL); //
```

函数返回值就是第一个参数值 1，表示此定时器的 ID 号; 第二个参数表示要等待 100 毫秒时间再重新处理一次; 第三个参数在这种方法中一般用 NULL。

2) 调用回调函数

首先写一个如下格式的回调函数

```
void CALLBACK TimerProc(HWND hWnd,UINT nMsg,UINT nTimerid,DWORD dwTime);
```

然后再用 SetTimer(1,100,TimerProc)函数来建一个定时器，第三个参数就是回调函数地址。

取消定时器：

调用 KillTimer 来取消定时，KillTimer 的原型

```
BOOL KillTimer(
 HWND hWnd, //  窗口句柄
 UINT_PTR uIDEvent    // ID
);
```

在 MFC 程序中可以直接调用 KillTimer(int nIDEvent)来取消定时器。

8. 内存的快速初始化和拷贝

用 memset（）函数可以对一块连续内存进行初始化，memcpy（）函数可以将一块连续内存的数据拷贝到另一块连续内存中。

例如 memset(m_pImgDataOut, 0, sizeof（int）*m_imgWidthOut*m_imgHeightOut)语句将 m_pImgDataOut 指针指向的一块连续 int 类型的内存初始化为 0。而 memcpy(m_pImgDataOut, buf, Imgsize)语句可以将 buf 指针指向的一块连续内存的数据快速拷贝到 m_pImgDataOut 指针指向的一块连续内存中。

9. avi 文件中的数据格式

AVI 中每一帧采用了 Bitmap 头格式，但是数据都是编过码的，与 bmp 文件数据是不一样的，因为 bmp 数据没有压缩；AVI 格式没有规定具体的编码（codec）和解码（decode），当用户使用的播放器不知道用什么 codec 的时候会提醒下载；在每一帧 decode 后，即显示之前时才能算是一个 bmp 文件。

10. 图像高斯平滑后可以较好的保持图像边缘

高斯平滑就是高斯函数和图像函数的卷积运算，时域卷积就是频域乘法，数字图像经过频域变换后得到的图像，实际是图像信号变换到频域后的函数以图像形式表示出来，其低频部分极接近原点部分对应于图像平滑变化不大的地方，如图像的背景之类，而高频部分对应于图像变化剧烈的地方，如图像的边缘。

11. 在分析基于视频的图像序列时，由于图像受亮度影响较大，去除背景时若直接把背景图和带有目标的图片做差再根据结果设置阈值效果不好，此时可考虑采用下述方法

若只考虑图像的亮度影响，可对背景图和目标图都减去各自的均值，在得到的两个图像上再进行求差，差图像中绝对值比较大的就应该是目标所在位置；还可以假设图像符合正态分布，把目标图和背景图调节成同均值同方差的，也可以消除亮度影响。

12. 基于视频流的运动目标提取时，阈值化后得到差分的二值图像里面有很多噪声，而且目标物体并不是一个大的连通域，除了腐蚀膨胀的方法去除噪声，还可考虑其它算法

例如用分水岭算法将目标分割成许多闭合的区域，然后根据其他信息进行融合，一般可以得到完整的目标。

13. 在进行图像处理时，有时仅需要对图像中的某一个区域进行处理，若用常规的方法响应鼠标左键按下、放开和鼠标移动的消息，编程相对比较复杂。此时可以用 MFC 类库中的 CRectTracker 类来获得操作区域

具体用法如下：

首先，在 View 类头文件中声明一个对象 CRectTracker RectTracker；

然后，在 OnDraw（）函数中加入下面的语句

```
RectTracker.Draw（pDC）;
```

最后，在响应鼠标左键按下消息处理函数 OnLButtonDown（）中加入语句

```
if （RectTracker.TrackRubberBand（this，point，TRUE））
  {
    Invalidate（）;
  }
```

由此，便可用 RectTracker 的成员变量 m_rect 获得所选的区域。

VC 调试中一些常见的错误信息及解决方法：

1、调试时出现 LIBCD.lib(crt0.obj) : error LNK2001: unresolved external symbol _main **错误**
原因：需要 MFC 支持的程序需要用 win32 Application 来生成，如果用 win32 的控制台程序就会出现上面的错误信息。

2、error C2018: unknown character '0xa1'
原因：输入了不可见的非法字符，把那一行前后的代码删掉重写一遍就 OK

3、error C2018: unknown character '0xa3'**与** error C2018: unknown character '0xbb'**常常同时出现**
原因：报错的行有中文字符或标点

4、fatal error C1010: unexpected end of file while looking for precompiled header directive
原因：寻找预编译头文件路径时遇到了不该遇到的文件尾。（一般是没有#include "stdafx.h"）

5、error C2011: 'C……': 'class' type redefinition
原因：类“C……”重定义。

6、error C2018: unknown character '0xa3'
原因：不认识的字符'0xa3'。（一般是汉字或中文标点符号）

7、LINK : fatal error LNK1168: cannot open Debug/P1.exe for writing
连接错误：不能打开 P1.exe 文件，以改写内容。（一般是 P1.Exe 还在运行，未关闭）

8、error LNK2001: unresolved external symbol "public: virtual _ _thiscall C……::~C……(void)"
连接时发现没有实现的外部符号（变量、函数等）。

9、fatal error C1010　unexpected end of file while looking for precompiled header
没加入预编译头 #include <stdafx.h>