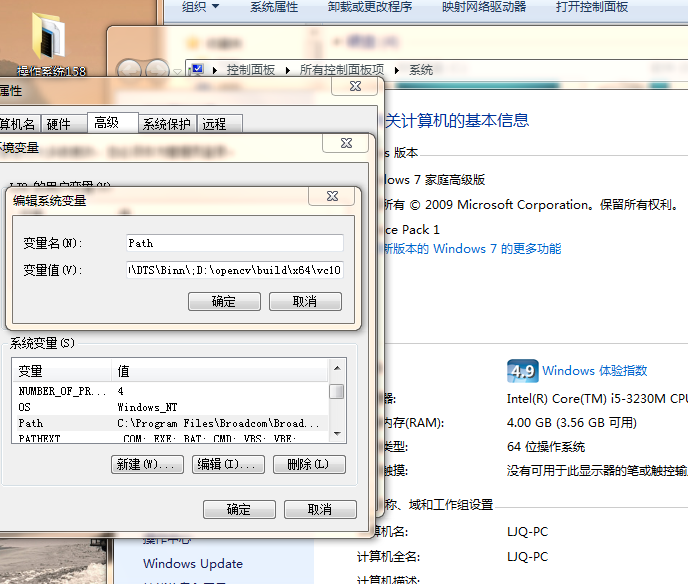
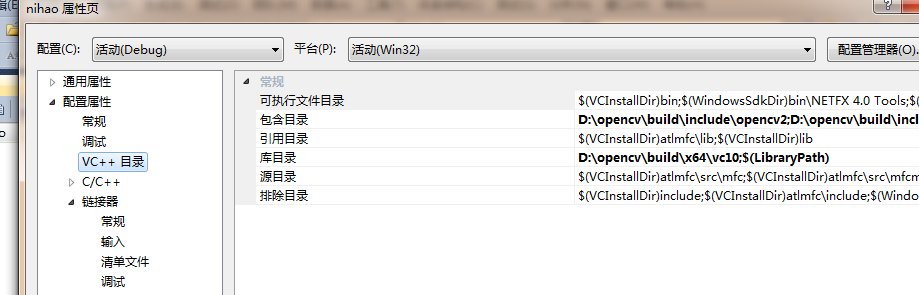
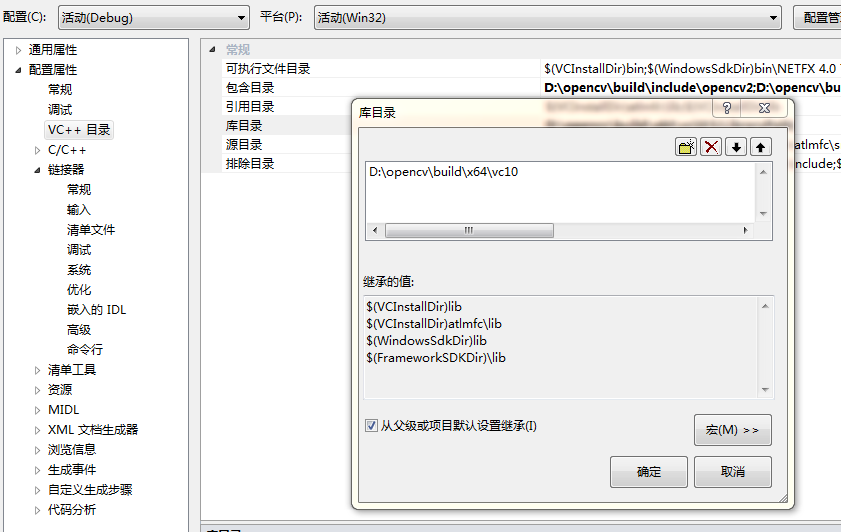
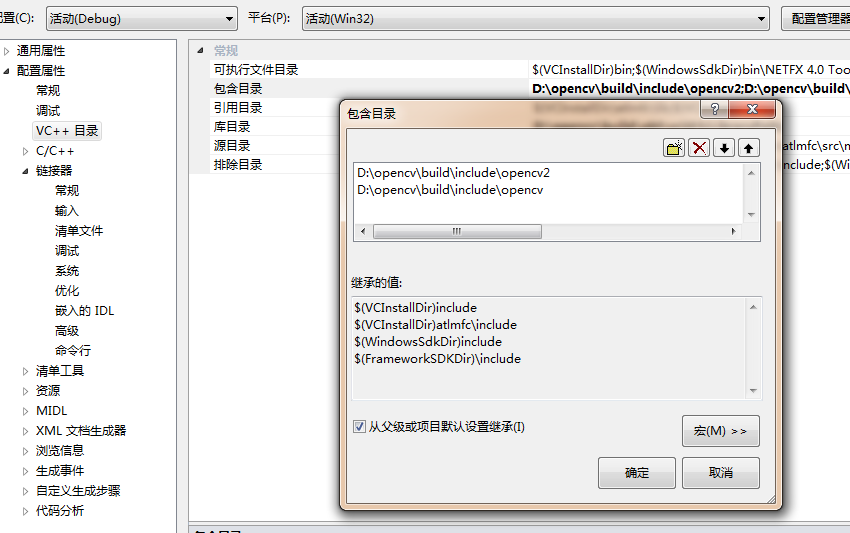
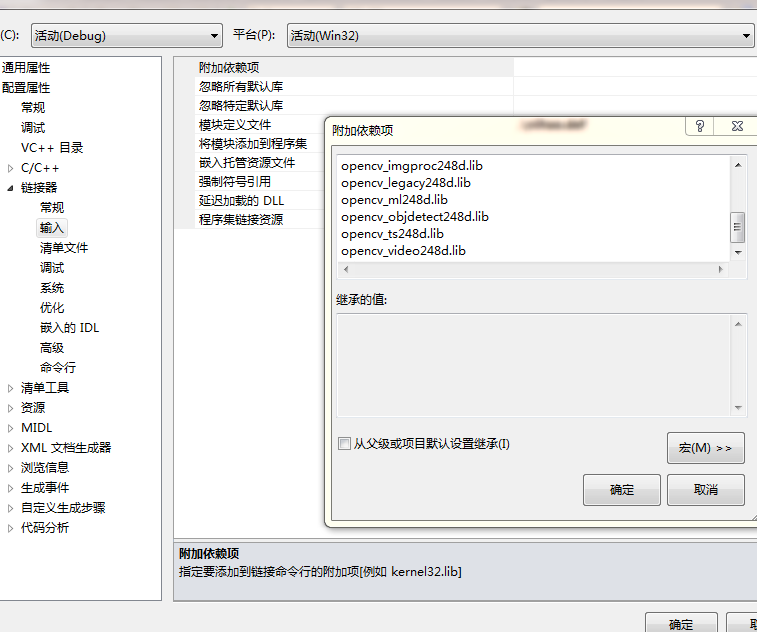
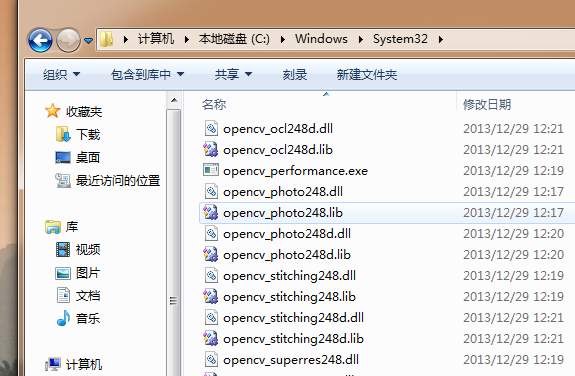
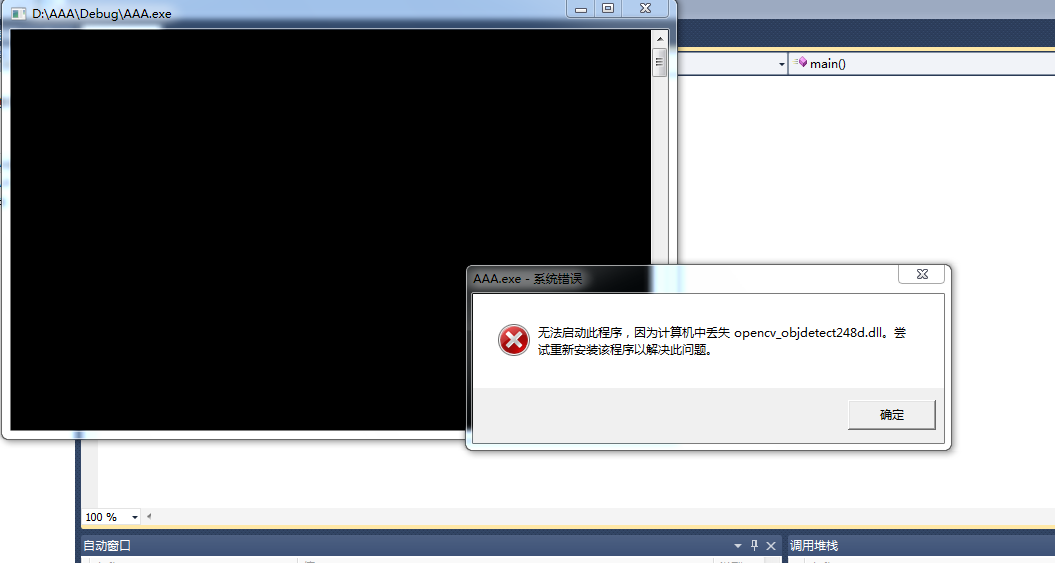
# 配置opencv的实验报告及程序

1. 下载opencv2.4.8,机器vs2010,window7 64操作系统。
2. 配置opencv的环境变量，在计算机->属性->高级系统设置->高级->环境变量->path（用户，系统里面的path任选其一）->在变量值里面添加路径
3. 接着在2010的属性中配置，在vc++的目录中的包含目录中添加opencv的路径，D:\opencv\build\include\opencv2和D:\opencv\build\include\opencv，接下来在库目录中添加D:\opencv\build\x64\vc10。在包含目录和库目录中有个从父级或项目默认设置继承，前面有个选项，这个钩要有。
4. 在属性中还有连接器-输入-附加依赖项添加opencv的lib路径，从父级或项目默认设置继承，前面有个选项，这个钩不要。这个配好就当前项目可以用，要是想一次配好就在Debug中配。
5. 在第一次运行是可以出现结果的，而在新建一个项目的时候就出现这样的问题

在C:\Windows\System32中添加opencv的lib和bin就可以这个路径问题解决了。

添加64位的操作系统不能用，需要改成32的，

如果是系统冲突的话那就是库文件中的路径的问题.

6.新建一个项目win32控制应用程序，在源文件中添加c++文件，粘贴如下代码

#include "opencv2/imgproc/imgproc.hpp"

#include "opencv2/highgui/highgui.hpp"

#include "highgui.h"

#include <stdlib.h>

#include <stdio.h>

using namespace cv;

/// 全局变量

Mat src, erosion\_dst, dilation\_dst;

int erosion\_elem = 0;

int erosion\_size = 0;

int dilation\_elem = 0;

int dilation\_size = 0;

int const max\_elem = 2;

int const max\_kernel\_size = 21;

/\*\* Function Headers \*/

void Erosion( int, void\* );

void Dilation( int, void\* );

/\*\* @function main \*/

int main( int argc, char\*\* argv )

{

/// Load 图像

src = imread( argv[1] );

if( !src.data )

{ return -1; }

/// 创建显示窗口

namedWindow( "Erosion Demo", CV\_WINDOW\_AUTOSIZE );

namedWindow( "Dilation Demo", CV\_WINDOW\_AUTOSIZE );

cvMoveWindow( "Dilation Demo", src.cols, 0 );

/// 创建腐蚀 Trackbar

createTrackbar( "Element:\n 0: Rect \n 1: Cross \n 2: Ellipse", "Erosion Demo",

&erosion\_elem, max\_elem,

Erosion );

createTrackbar( "Kernel size:\n 2n +1", "Erosion Demo",

&erosion\_size, max\_kernel\_size,

Erosion );

/// 创建膨胀 Trackbar

createTrackbar( "Element:\n 0: Rect \n 1: Cross \n 2: Ellipse", "Dilation Demo",

&dilation\_elem, max\_elem,

Dilation );

createTrackbar( "Kernel size:\n 2n +1", "Dilation Demo",

&dilation\_size, max\_kernel\_size,

Dilation );

/// Default start

Erosion( 0, 0 );

Dilation( 0, 0 );

waitKey(0);

return 0;

}

/\*\* @function Erosion \*/

void Erosion( int, void\* )

{

int erosion\_type;

if( erosion\_elem == 0 ){ erosion\_type = MORPH\_RECT; }

else if( erosion\_elem == 1 ){ erosion\_type = MORPH\_CROSS; }

else if( erosion\_elem == 2) { erosion\_type = MORPH\_ELLIPSE; }

Mat element = getStructuringElement( erosion\_type,

Size( 2\*erosion\_size + 1, 2\*erosion\_size+1 ),

Point( erosion\_size, erosion\_size ) );

/// 腐蚀操作

erode( src, erosion\_dst, element );

imshow( "Erosion Demo", erosion\_dst );

}

/\*\* @function Dilation \*/

void Dilation( int, void\* )

{

int dilation\_type;

if( dilation\_elem == 0 ){ dilation\_type = MORPH\_RECT; }

else if( dilation\_elem == 1 ){ dilation\_type = MORPH\_CROSS; }

else if( dilation\_elem == 2) { dilation\_type = MORPH\_ELLIPSE; }

Mat element = getStructuringElement( dilation\_type,

Size( 2\*dilation\_size + 1, 2\*dilation\_size+1 ),

Point( dilation\_size, dilation\_size ) );

///膨胀操作

dilate( src, dilation\_dst, element );

imshow( "Dilation Demo", dilation\_dst );

}