案例五：图像文件读入和显示

1.案例目的

•熟悉Visual Studio 2010开发环境。

•了解OpenCV开发库，学会在VS2010环境下安装和配置OpenCV。

•掌握开源库的通用安装和配置方法。

•通过运行一个简单的程序，初步了解OpenCV程序的特点。

2.案例内容

•编写程序掌握OpenCV加载及显示图像的方法。

•查看、理解并掌握图像的属性（包括类型、大小、分辨率等）。

3.案例知识点

Opencv的使用

4.案例时长

共1学时，具体安排如下：

•安装OpenCV（0.5学时）

•编写代码配置程序（0.5学时）

5.案例实验环境

**•软件环境：**

1）OpenCV 2.4.11

**•开发环境与工具：**

1）VS2010

6.案例分析

本案例主要分为以下2步骤：

1. 检查所用的计算机系统是否已安装了VS2010（中英文版本均可）。
2. 安装OpenCV 2.4.11，若已安装，则查看所在目录，这里设opencv的安装目录为：D:\opencv。
3. 编写代码并配置opencv程序，完成程序的编译、链接和运行。步骤如下：**（如想要每一步的详细图片展示，请参考本章第8小节“OpenCV配置文档”）：**

7.案例实验过程

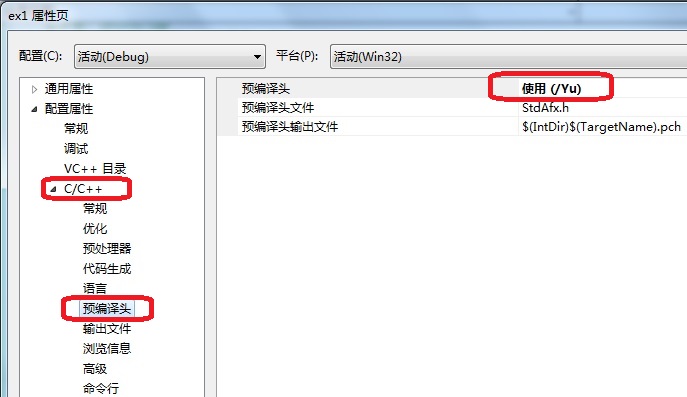
7.1程序中常见问题及OpenCV函数解释

【关于预编译头】

预编译头就是将程序的头文件部分编译成一个二进制中间文件，提高整个工程的编译效率，一般以.pch格式存储。

如果使用了预编译头，则#include "stdAfx.h "  必须放到所有#include的前面，也就是程序的最前面，否则它前面的内容会被忽略。

若不想使用预编译头，需将项目属性中的 “预编译头”改为不使用，如下：



【关于\_tmain】

\_tmain()是unicode版本的的main()函数，是为了保证移植unicode（统一码，万国码）的正确性而使用的main的别名，编译后仍为main。

 \_tmain的定义在<tchar.h>可以找到，即 #define \_tmain main

imread从文件读取图像

语法：Mat imread(const string& filename, intflags=1 );

参数：

filename：待载入的图像文件名，包括扩展名。可以载入BMP，DIB，JPEG，PNG，PBM，PGM，PPM，SR，RAS和TIFF等文件格式。

flag表示是否是彩色图像（多通道），默认为1，表示读入的图像被强制为3通道的彩色图像

flag=0，表示载入图像强制为单通道灰度图像

flag>1，表示载入图像由文件中的图像通道数决定。

返回值为Mat结构指针。

7.2 OpenCV配置文档

选择开发工具：Visul Studio 2010（编译环境）+OpenCV 2.4.11（开源库）

本配置文档是在32位Win7操作系统下测试的，版本如有不同，配置略有不同。

首先：安装OpenCV 2.4

将下载（给定）的OpenCV-2.4.11压缩包直接解压到你要安装的目录中，比如：D:\opencv

（一）添加环境变量

计算机——（右键）属性——高级系统设置——环境变量——系统变量，在此添加OpenCV的路径：

双击“系统变量”下的“Path”，在变量值一栏添加：

D:\opencv\build\x86\vc10\bin

如图1所示：

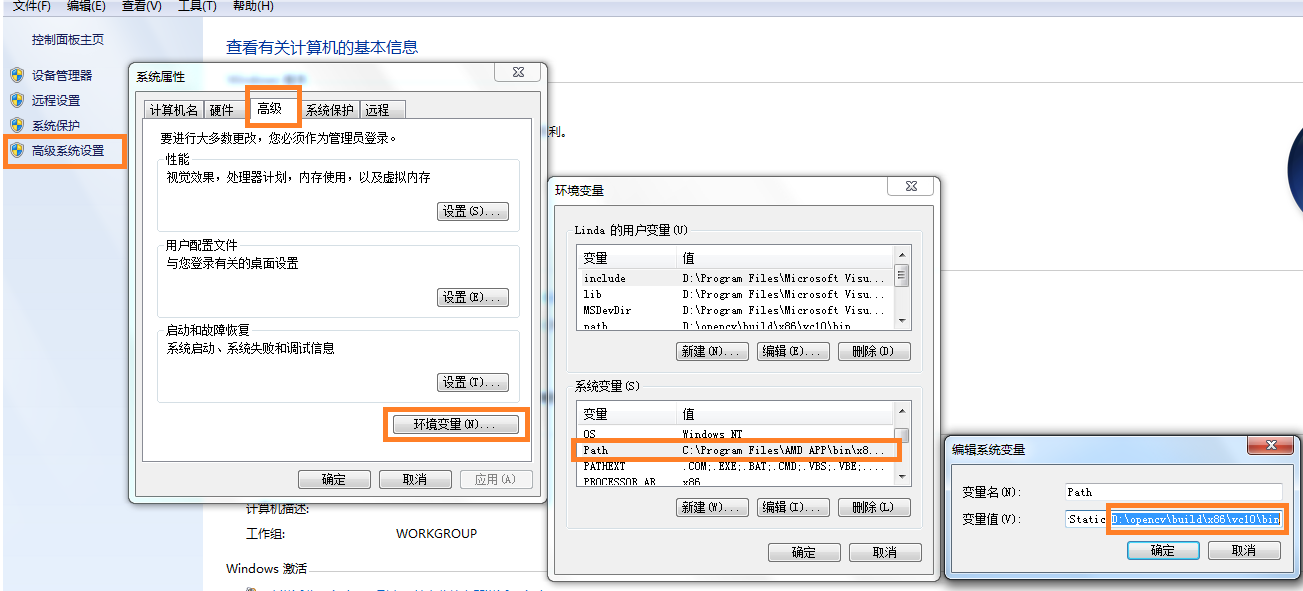


图1 添加opencv环境变量

（二）新建文件或工程，书写代码。

打开VS2010，新建项目，选择“Win32控制台应用程序”，在下面的“名称”栏输入项目名称，在“位置”栏输入要存放的位置，单击“确定”，进入代码编写页面。如图2所示：

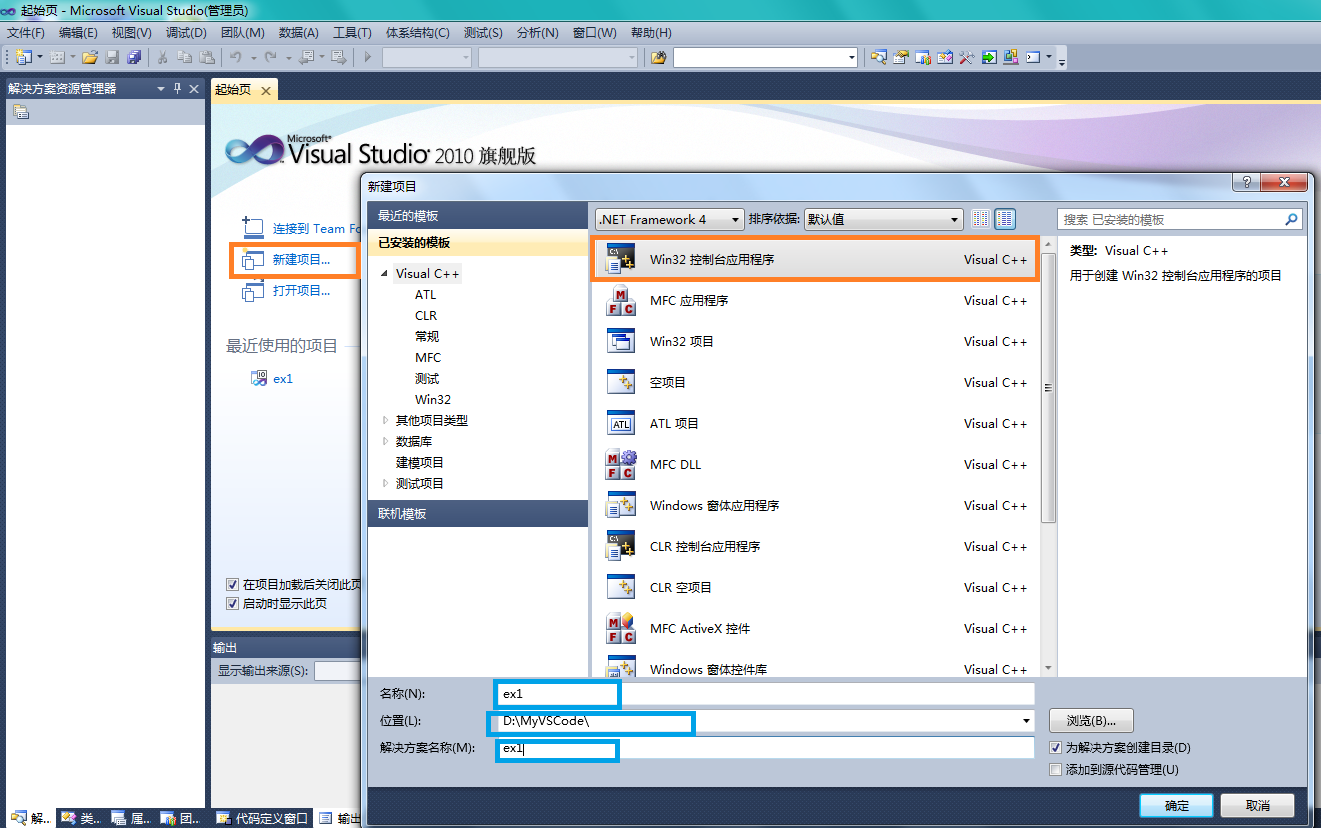


图2 在VS2010中新建项目

（三）配置工程的OpenCV依赖

此步对于每一个新建工程都要重新配置。分为如下三个步骤：

1.添加头文件和库文件目录

在VS2010环境下，针对当前工程：

项目——属性——VC++目录，在此处添加“包含目录”和“库目录”，如图3、图4所示：

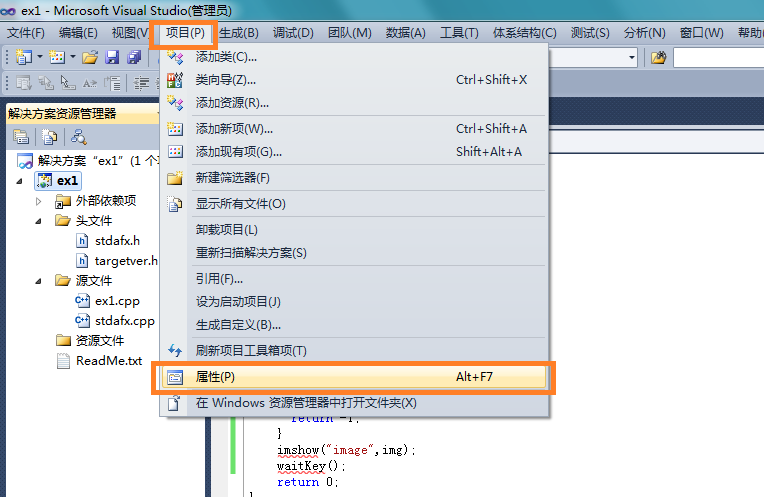


图3 项目属性

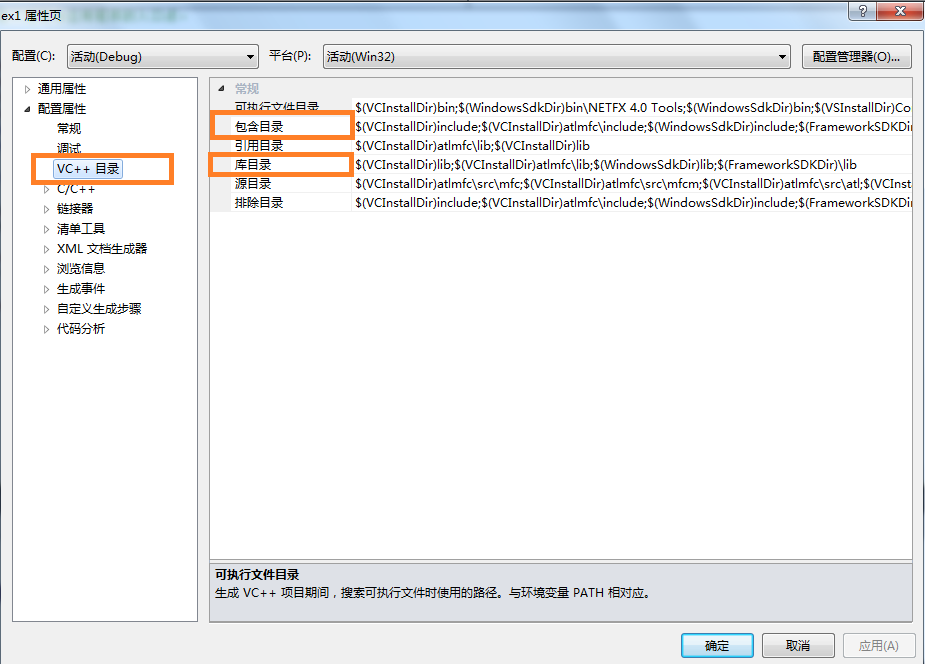


图4 VC++目录

（1）“包含目录”：点击“编辑”，在弹出的窗口空白处，输入opencv包含文件所在的目录：（或浏览到此目录自动添加）

D:\opencv\build\include

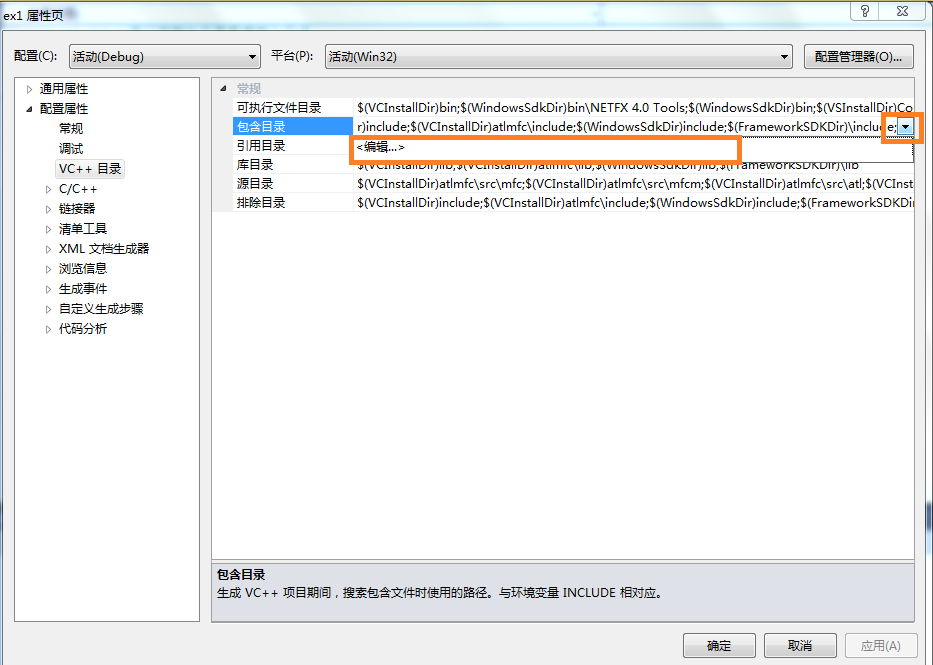


图5 包含目录，点击“编辑”

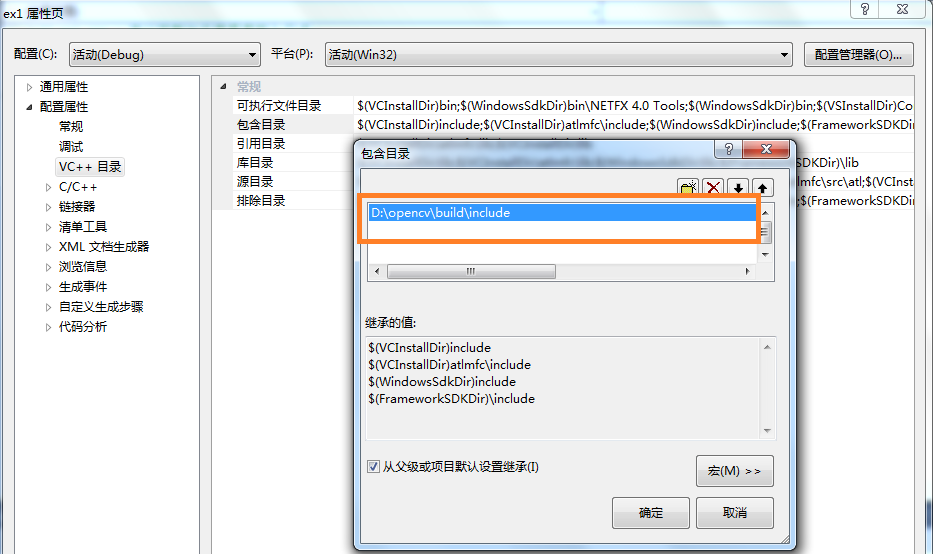


图6 输入opencv包含文件目录

（2）“库目录”：点击“编辑”，添加（或浏览到此目录自动添加）：

D:\opencv\build\x86\vc10\lib

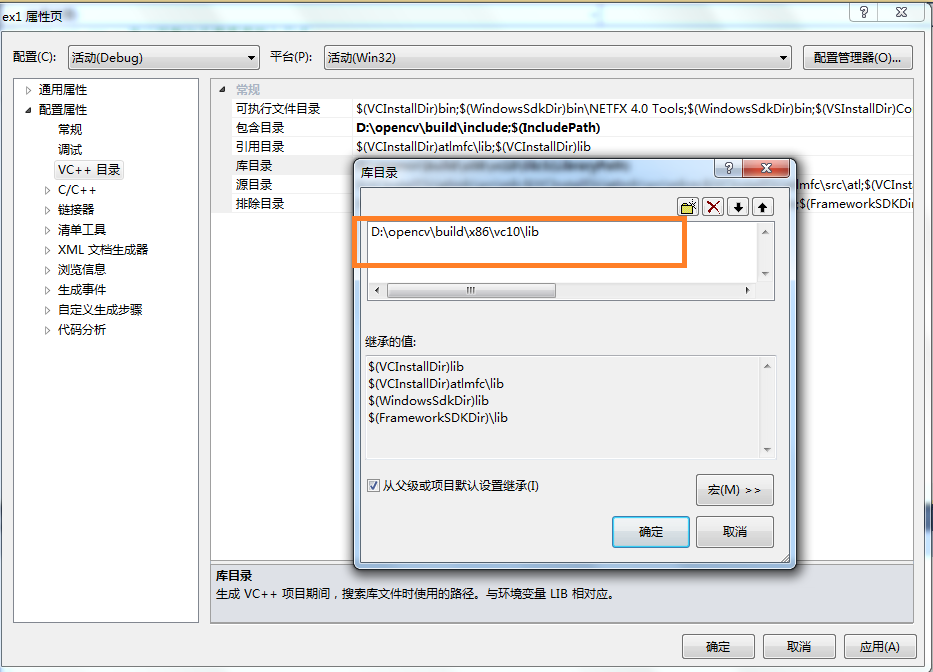


图7 输入opencv库目录

所有目录添加完毕，应该如图8所示：

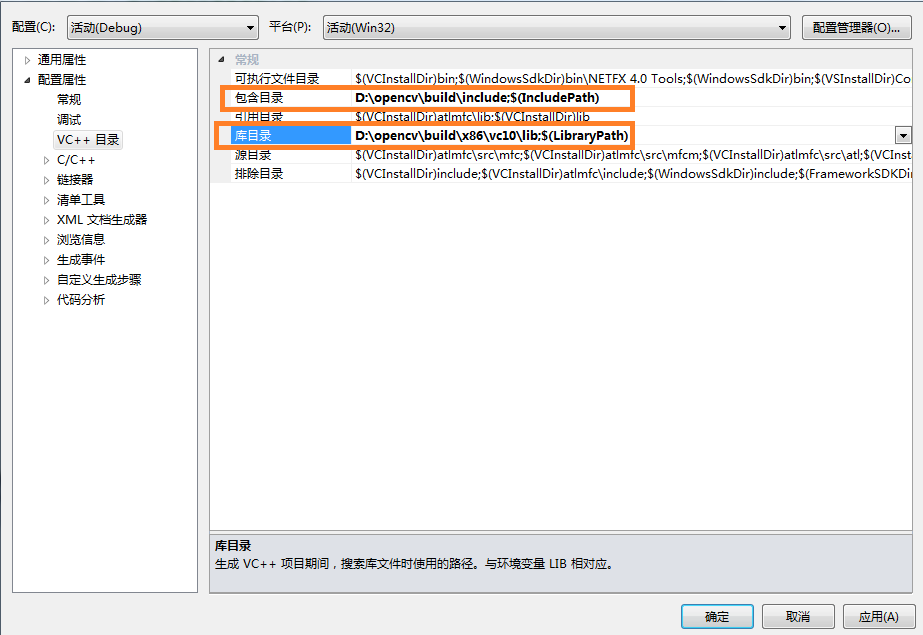


图8 目录添加完毕

2.添加库文件链接

项目——属性——链接器——输入——（右侧）附加依赖项——编辑：

针对Debug模式添加以下13个静态链接库文件：

opencv\_calib3d2411d.lib

opencv\_contrib2411d.lib

opencv\_core2411d.lib

opencv\_features2d2411d.lib

opencv\_flann2411d.lib

opencv\_gpu2411d.lib

opencv\_highgui2411d.lib

opencv\_imgproc2411d.lib

opencv\_legacy2411d.lib

opencv\_ml2411d.lib

opencv\_objdetect2411d.lib

opencv\_ts2411d.lib

opencv\_video2411d.lib

如果程序使用release模式，则将上面的13个库文件“.lib”前面的d字母都去掉，添加到release下的附加依赖项即可。如图9、10所示：

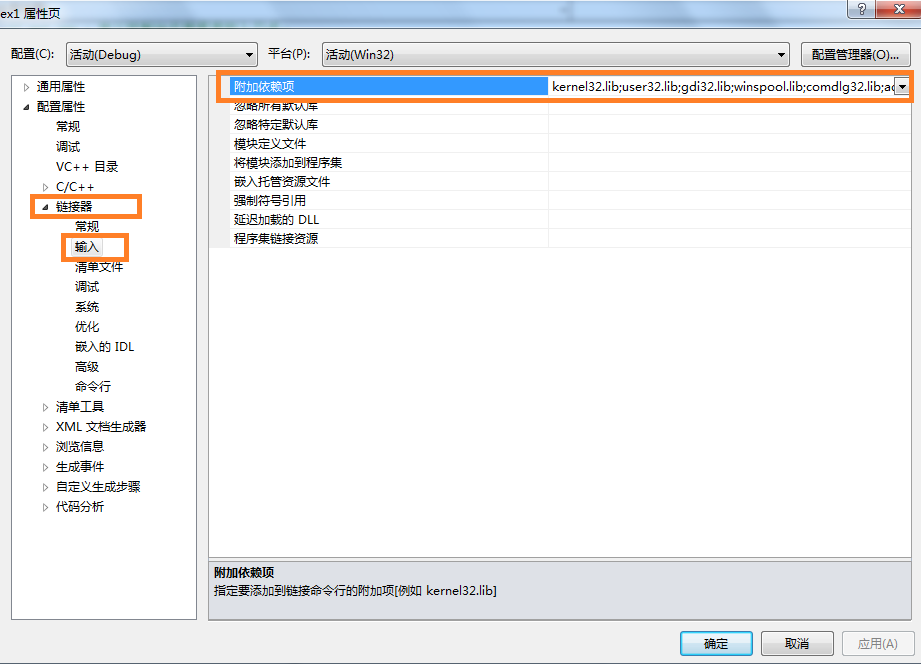


图9 打开附加依赖项

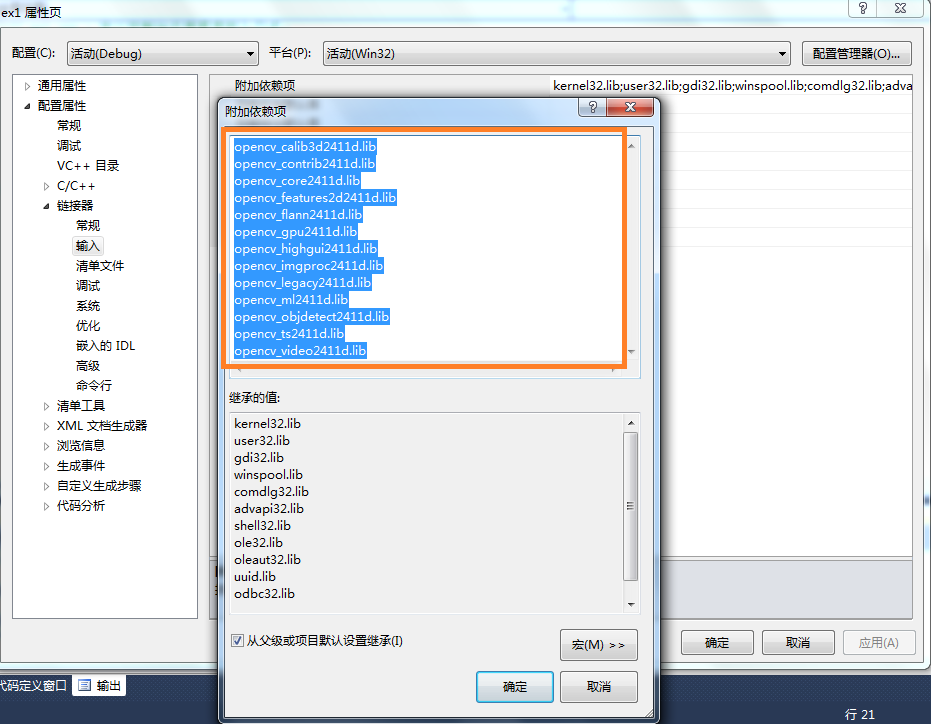


图10 添加附加依赖项（13个opencv库文件）

3.添加dll支持

将OpenCV的所有动态链接库文件dll（OpenCV->bin目录下的8个dll文件）直接拷贝到系统目录windows\system32下。

配置此步骤的原因：程序在链接、运行时，需要OpenCV的动态链接库的支持，如没有，程序会提示找不到dll文件。

7.3配置环境

（一）添加环境变量

计算机——（右键）属性——高级系统设置——环境变量——系统变量，在此添加OpenCV的路径：

双击“系统变量”下的“Path”，在变量值一栏添加：

D:\opencv\build\x86\vc10\bin

（二）新建文件或工程，书写代码。

打开VS2010，新建项目，**选择“Win32控制台应用程序”**，在下面的“名称”栏输入项目名称，在“位置”栏输入要存放的位置，单击“确定”，进入代码编写页面。

将本课最后的***“4.参考程序代码 Hello1”***拷贝到代码页面。

（三）配置工程的OpenCV依赖

**此步对于每一个新建工程都要重新配置。**分为如下三个步骤：

1.添加头文件和库文件目录

在VS2010环境下，针对当前工程：

项目——属性——VC++目录，在此处添加“包含目录”和“库目录”

（1）“包含目录”：点击“编辑”，在弹出的窗口空白处，输入opencv包含文件所在的目录：（或浏览到此目录自动添加）

D:\opencv\build\include

（2）“库目录”：点击“编辑”，添加（或浏览到此目录自动添加）：

D:\opencv\build\x86\vc10\lib

2.添加库文件链接

项目——属性——链接器——输入——（右侧）附加依赖项——编辑：

针对Debug模式添加以下13个静态链接库文件：

opencv\_calib3d2411d.lib

opencv\_contrib2411d.lib

opencv\_core2411d.lib

opencv\_features2d2411d.lib

opencv\_flann2411d.lib

opencv\_gpu2411d.lib

opencv\_highgui2411d.lib

opencv\_imgproc2411d.lib

opencv\_legacy2411d.lib

opencv\_ml2411d.lib

opencv\_objdetect2411d.lib

opencv\_ts2411d.lib

opencv\_video2411d.lib

如果程序使用release模式，则将上面的13个库文件“.lib”前面的d字母都去掉，添加到release下的附加依赖项即可。

3.添加dll支持

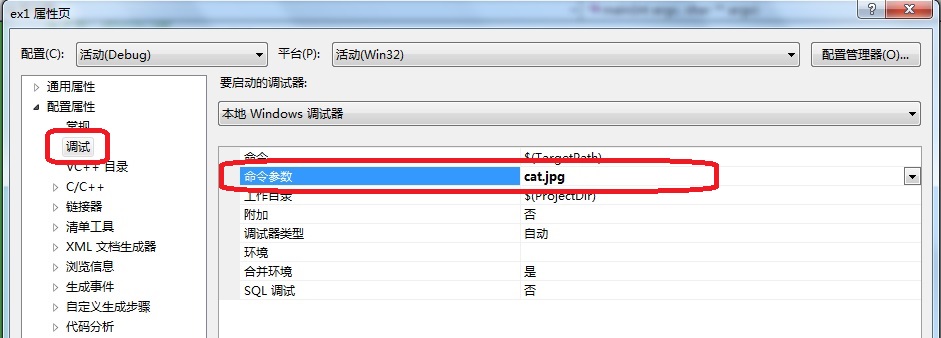
将OpenCV的所有动态链接库文件dll（OpenCV->bin目录下的8个dll文件）直接拷贝到系统目录windows\system32下。

配置此步骤的原因：程序在链接、运行时，需要OpenCV的动态链接库的支持，如没有，程序会提示找不到dll文件。

7.4运行程序

（1）将代码改为“***7.参考程序代码 Hello2***”，重新运行。

该例程演示了如何从外部输入main函数的参数（命令行参数），输入界面如下：



带参数的main函数：int main(int argc, char\*\* argv)

参数argc——表示参数个数（本程序的exe +　外部输入的参数个数）

参数argv——存储本程序的exe 以及 外部输入的参数 的二维字符数组

如本例中，argc = 2, argv[0]=”hello2.exe”, argv[1]=”cat.jpg”

8.案例代码

【案例代码】hello.cpp

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：hello.cpp

功 能：读入指定文件，并在屏幕上显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace cv;

using namespace std;

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

Mat image; //声明图像变量

image = imread("cat.jpg"); //读取图像

if (image.data != NULL)

{

cv::namedWindow("Original Image"); //声明窗口

imshow("Original Image",image); //显示图像

waitKey(0);

}else{

cout<<"error:图片不存在!"<<endl;

}

return 0;

}

【案例代码】hello2.cpp

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：hello2.cpp

功 能：读入指定文件，并在屏幕上显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace cv;

using namespace std;

int main1(int argc, char\*\* argv){

Mat image; //声明图像变量

if (argc == 2 &&(image = imread(argv[1])) .data != NULL )

{

cv::namedWindow("Original Image"); //声明窗口

imshow("Original Image",image); //显示图像

waitKey(0);

}else{

cout<<"error:图片不存在!"<<endl;

}

return 0;

}