**OpenCV2.4.9+VS2012安装手记**

Step 0. 如已获得bin、lib、和include 文件夹下编译好动态库和静态库，直接跳到Step 9进行配置。

Step 1.确认Visual Studio VC++软件已经安装（VS2012版本）

Step 2.登陆<https://opencv.org/releases.html> 下载OpenCV 2.4.9(2014-4-11版本) （用迅雷）；

登陆<https://cmake.org/download/> 下载Cmake（ Windows win32-x86）软件（直接下载）；

并把两个文件存储在C:\Users\lenovo\Documents\AIRecSysDesign\Software目录下。

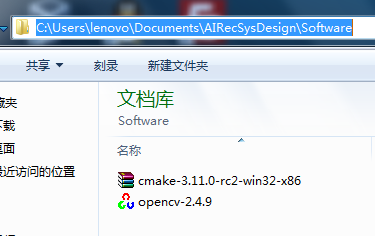
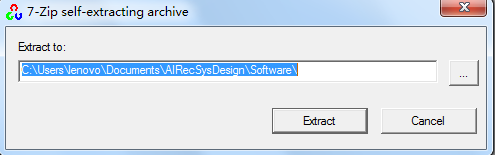
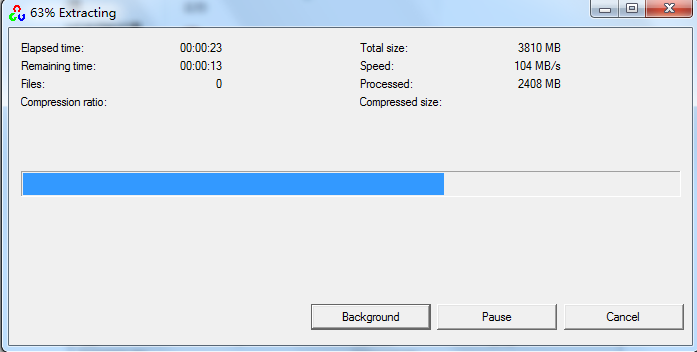


图1 已经下载的文件

Step 3.点击opencv-2.4.9.exe和cmake-3.11.0-rc2-win32-x86.zip解压相关文件



（a）解压路径（默认）



（b）解压中

图2 解压opencv-2.4.9.exe

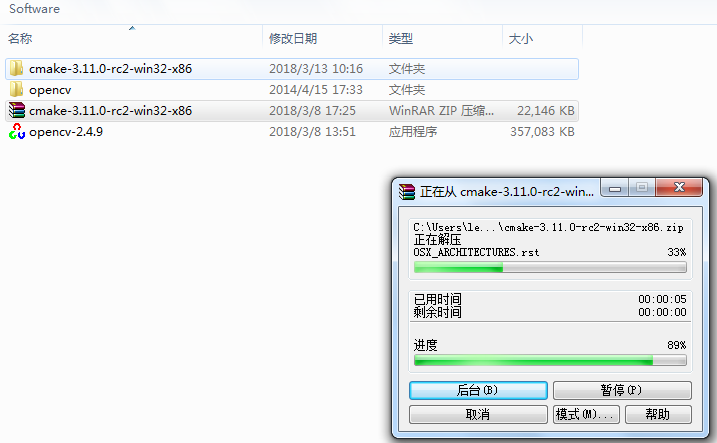


图3 解压cmake-3.11.0-rc2-win32-x86.zip

Step 4.运行cmake-gui，并进行路径设置。例如设置源代码路径为：C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources（假设OpenCV文件存放在C:\Users\lenovo\Documents\AIRecSysDesign\Software目录）。同时再设置编译结果的目标位置，需要新建文件夹C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012，用于存放编译结果。

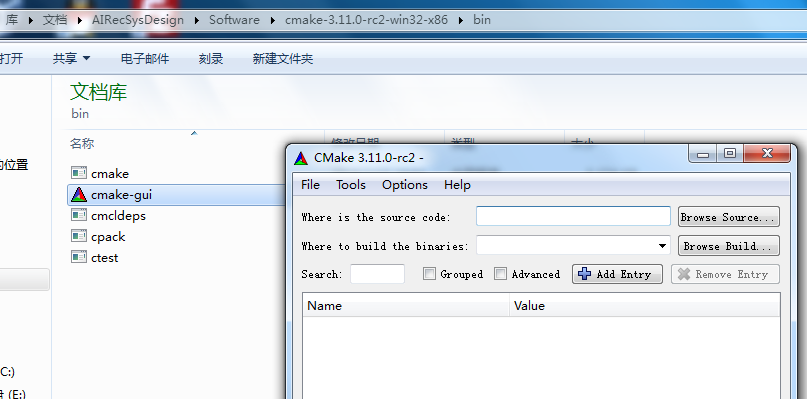


图4. cmake-gui.exe的运行界面

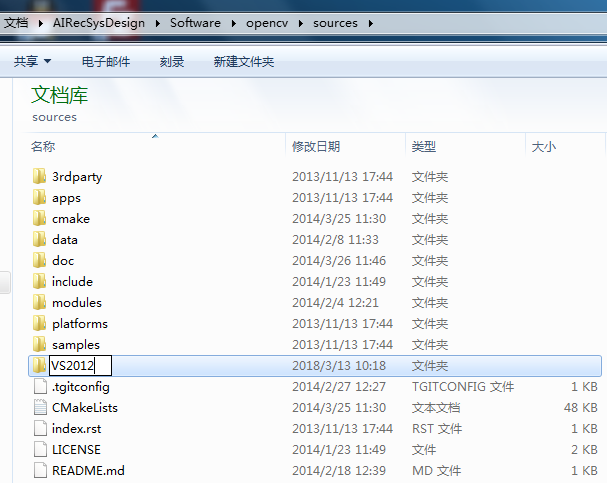


图5. 在sources下面新建VS2012文件夹

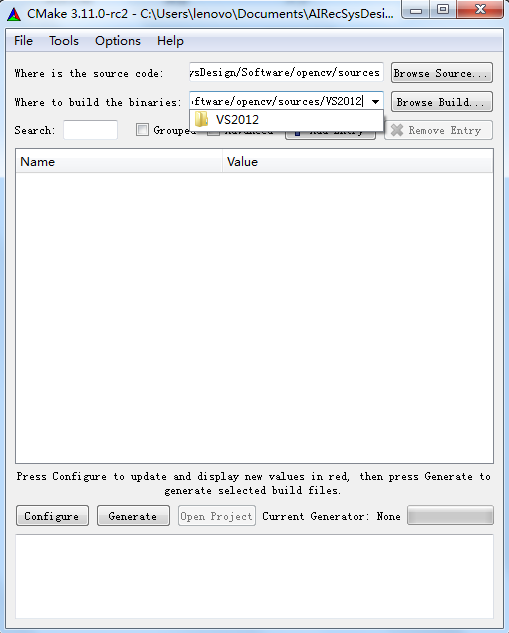
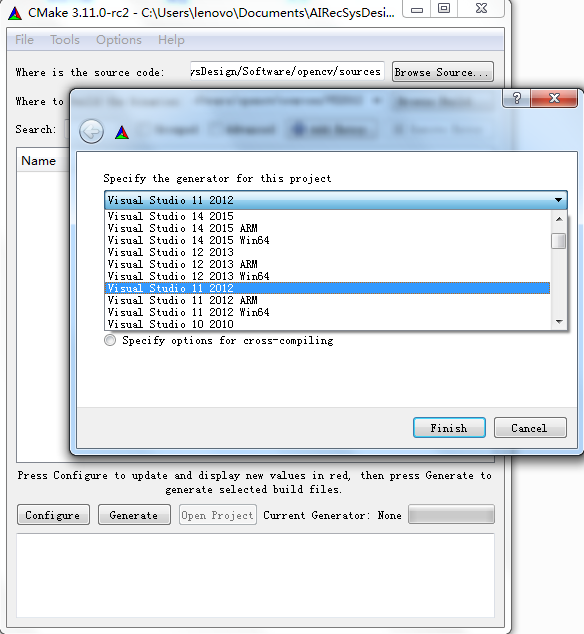


图6. cmake-gui中的设置

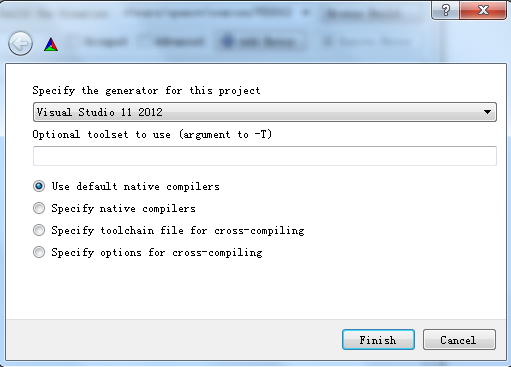
源文件位置：C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources

结果存放：C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012

Step 5.然后点 configure，在弹出的对话框内选择 Visual Studio 11 2012。

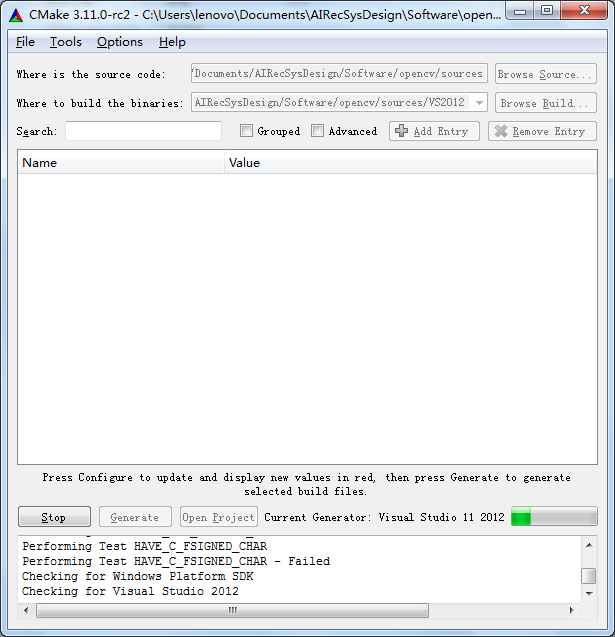


（a）选择编译器

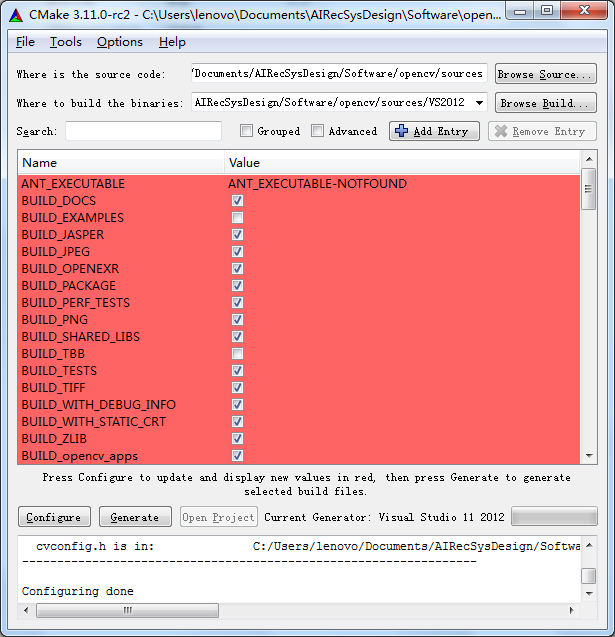


（b）选完后的界面

图7. 在configure中选择编译器



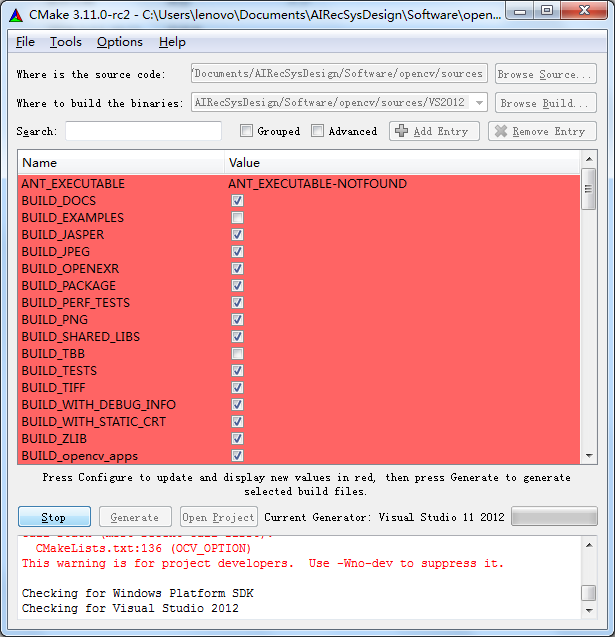
（a）第一次configure编译中



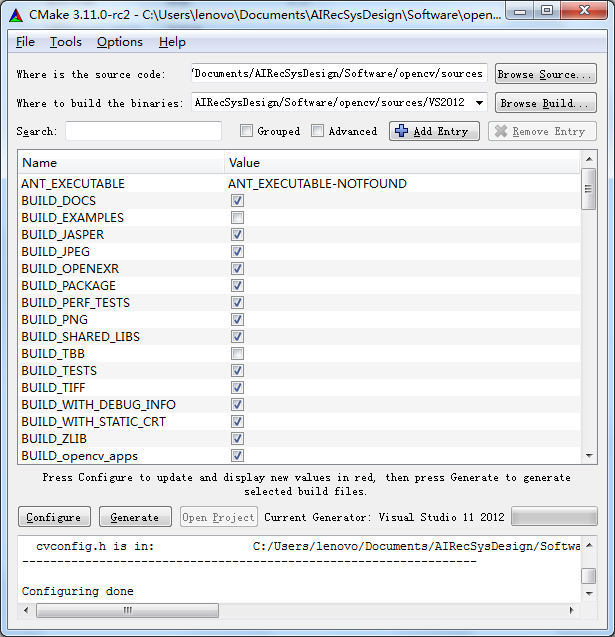
（b）第一次configure 完成

图8. 第一次configure的过程

Step 6.以上步骤运行完后，再次点击configure



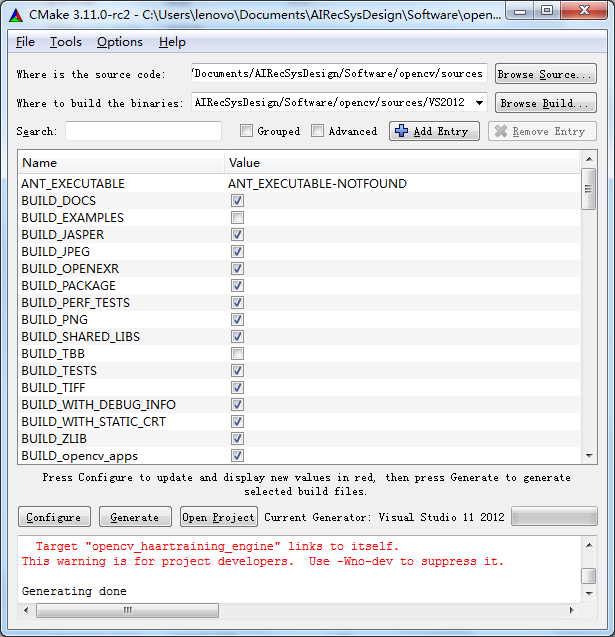
（a）第二次configure编译中



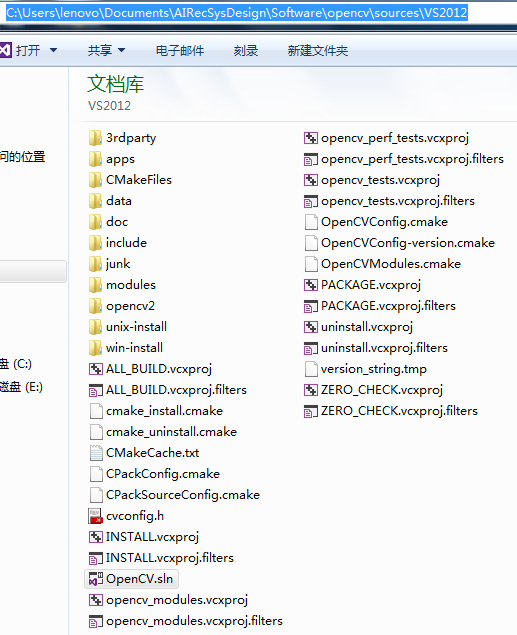
（b）第二次configure编译完成

图9. 第二次configure编译过程

Step 7.点击generate， C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012目录下会生成OpenCV.sln等相关工程文件，用于后面编译之用。



（a）Generate编译完



（b）Generate编译完后新建立的VS2012文件夹下面产生OpenCV.sln相关的工程文件

图10. Generate编译完成及其生成结果

Step 8.编译 OpenCV Debug和Release版本库

完成上一步骤后，将在C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012目录下生成OpenCV.sln的VC Solution File，请用VC++ 2012 打开OpenCV.sln，然后执行如下操作：

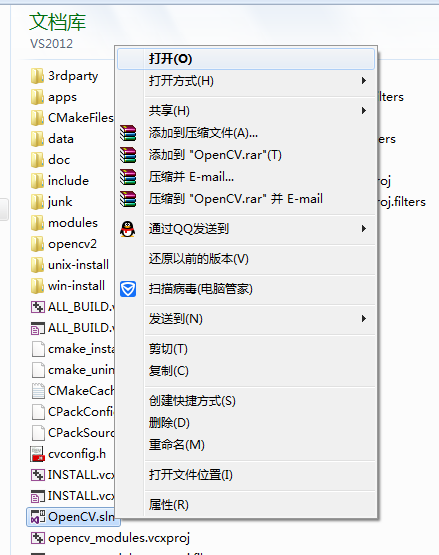
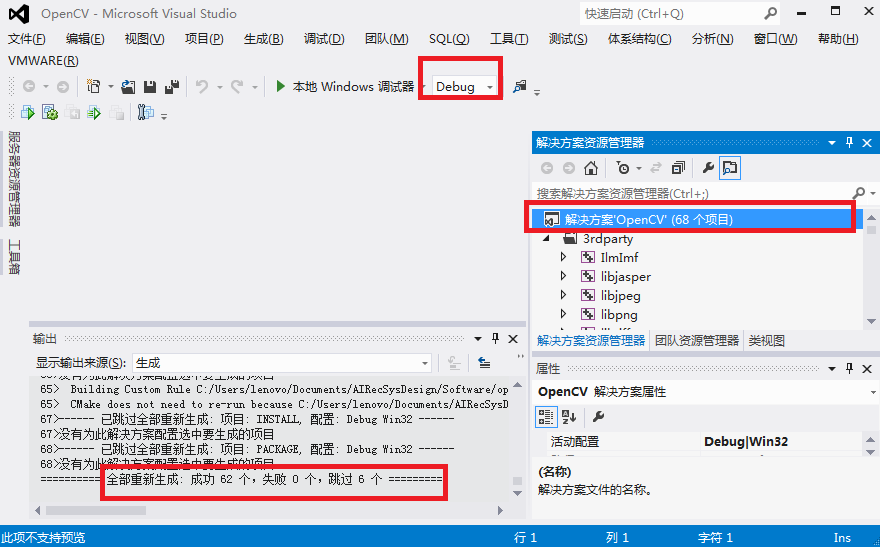


图11.用VC++2012打开OpenCV.sln

8.1)在Debug下，选择Solution Explorer里的 Solution OpenCV，点右键，运行"Rebuild Solution"；如编译无错误，再选择INSTALL项目，运行"Build"。

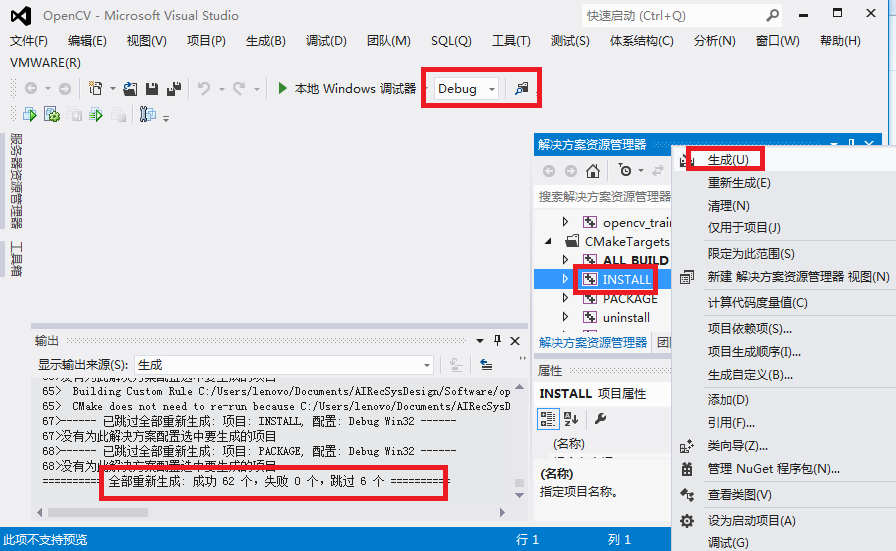


（a）Debug下在解决方案处右键属性栏中选择“重新生成解决方案”

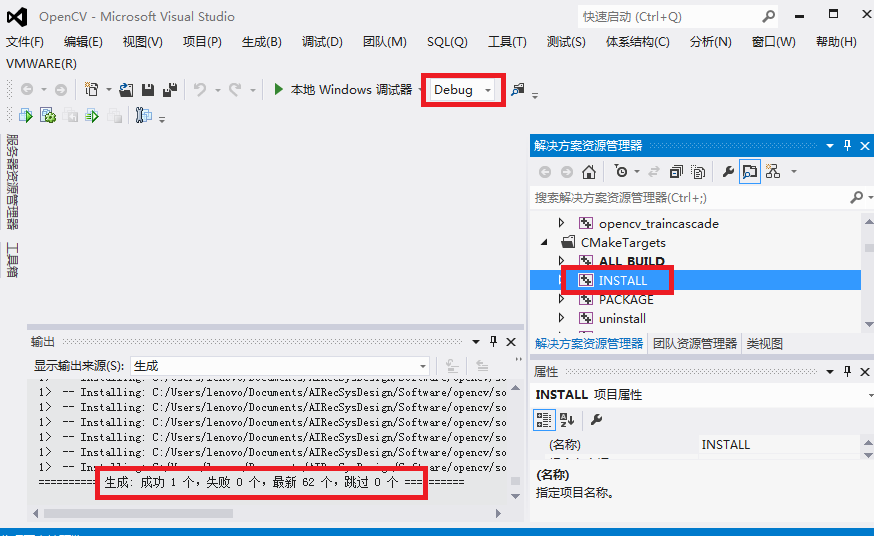


（b）Debug下Rebuild Solution生成成功后的状态

图12. 在Debug下运行Rebuild Solution



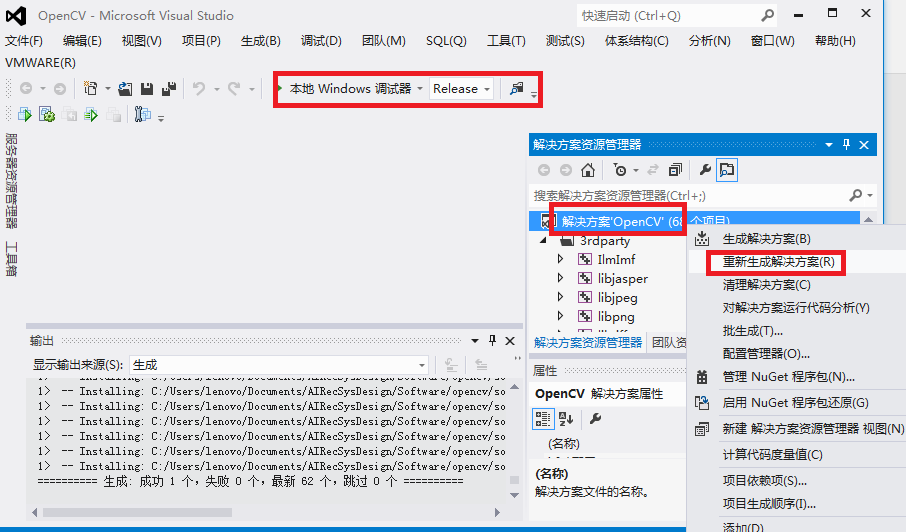
（a）在Debug下，选择INSTALL项目，运行右键属性中的"生成"



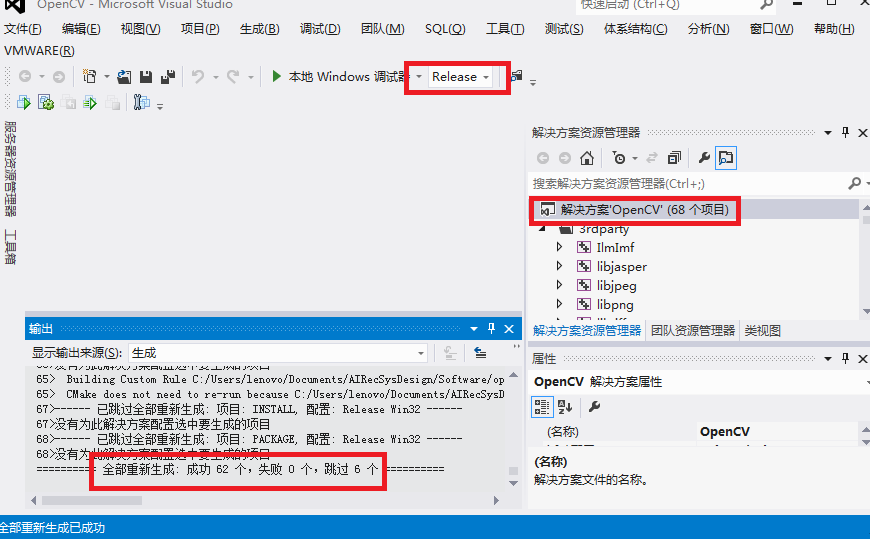
（b）Debug下对INSTALL进行生成，成功后的状态

图13.在Debug下INSTALL生成成功

8.2) 把本地Windows调试器转换到Release版本，重复以上操作。在Release下，选择Solution Explorer里的 Solution OpenCV，点右键，运行"Rebuild Solution"；如编译无错误，再选择INSTALL项目，运行"Build"。

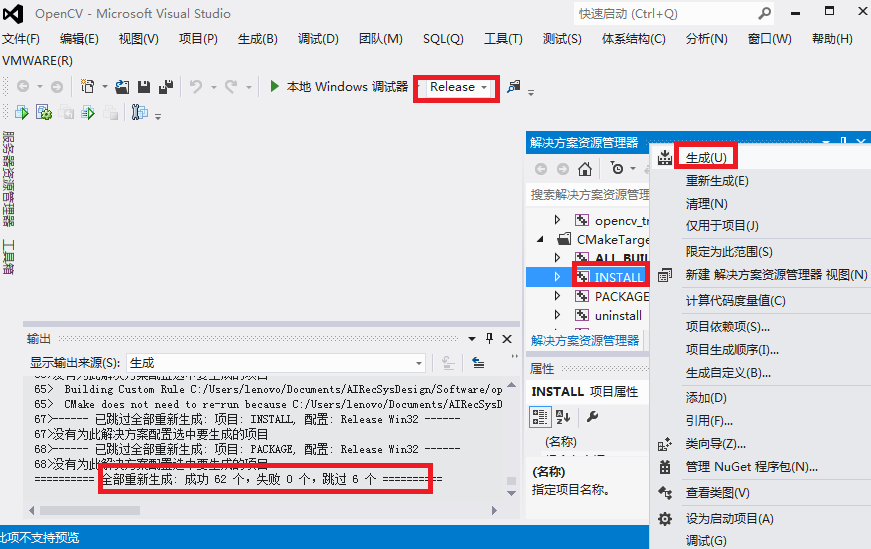


（a）Release下在解决方案处右键属性栏中选择“重新生成解决方案”

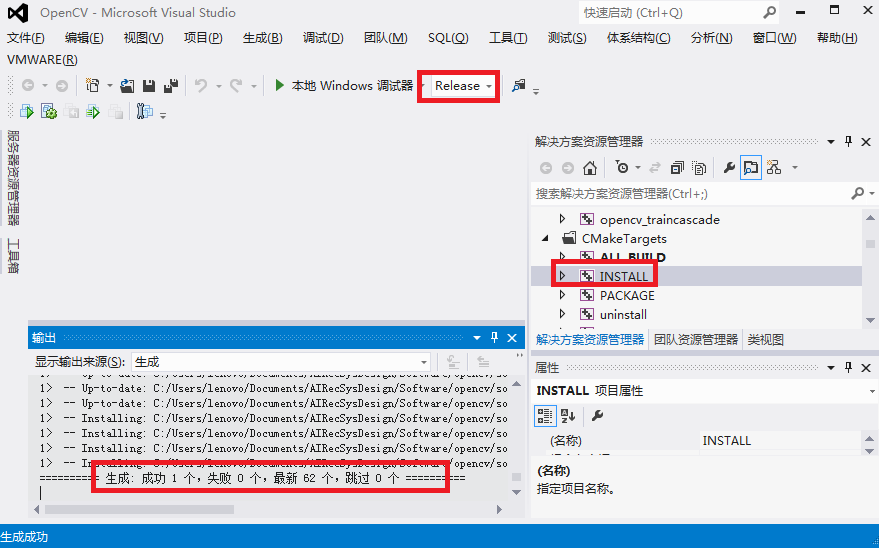


（b）Release下Rebuild Solution生成成功后的状态

图14. 在Release下运行Rebuild Solution



（a）在Release下，选择INSTALL项目，运行右键属性中的"生成"



（b）Release下对INSTALL进行生成，成功后的状态

图15. 在Release下INSTALL生成成功

8.3)此时，会生成以下文件：

（1）OpenCV的\*d.dll文件（debug版本）和\*.dll文件（release版本）将出现在C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012\bin目录中；

（2）OpenCV的\*d.lib文件（debug版本）和\*.lib文件（release版本）将出现在C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012\lib目录；

（3）头文件\*.h出现在

C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012\install\include中。

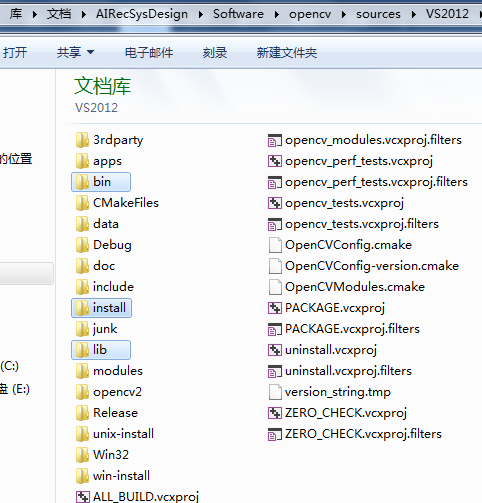


图16. VS2012下生成的bin、lib和install文件夹及相关文件

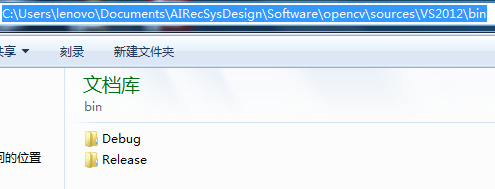


图17. Bin目录下的文件夹

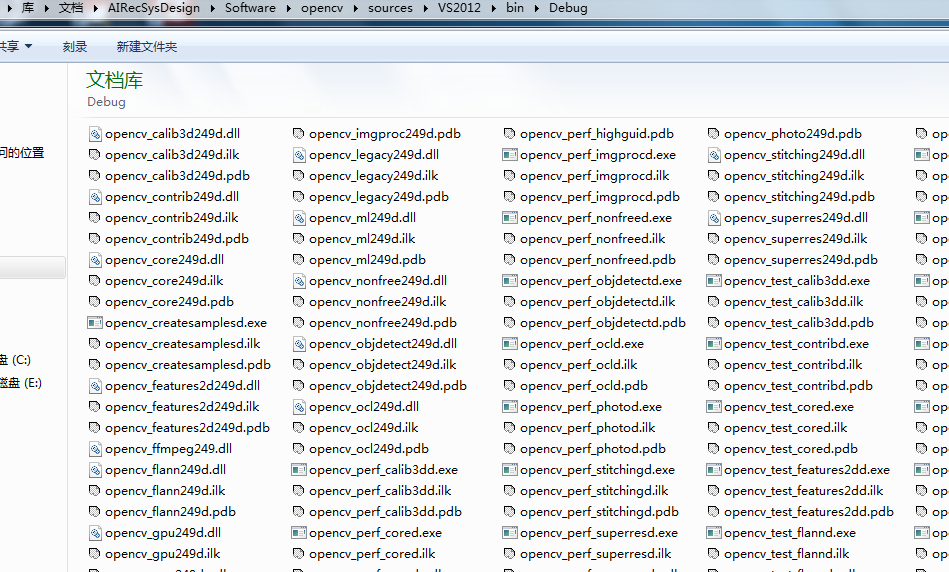


图18. Bin目录下Debug文件夹中的文件

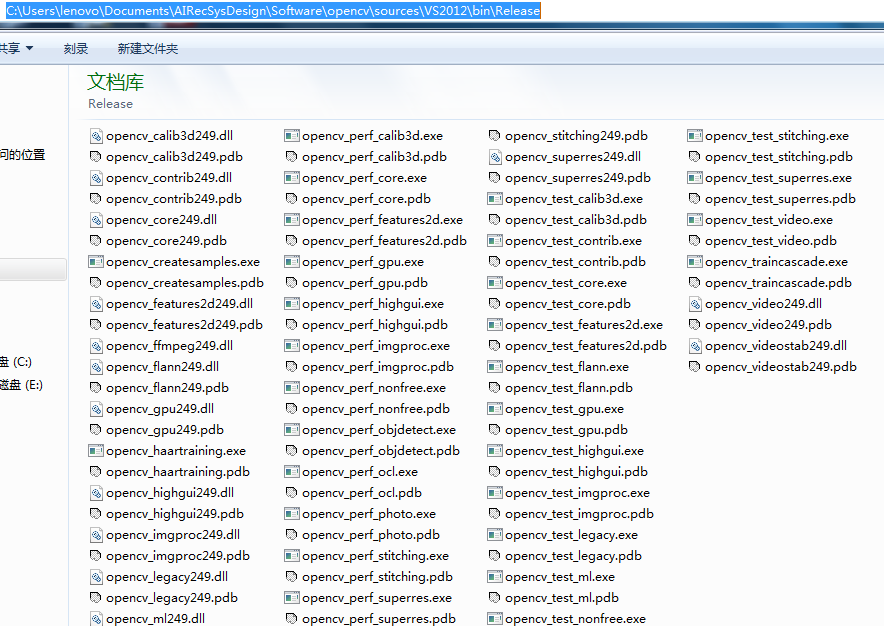


图19. Bin目录下Release文件夹中的文件

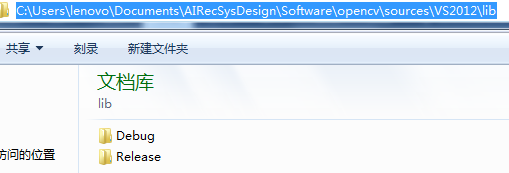


图20. Lib目录下的文件夹

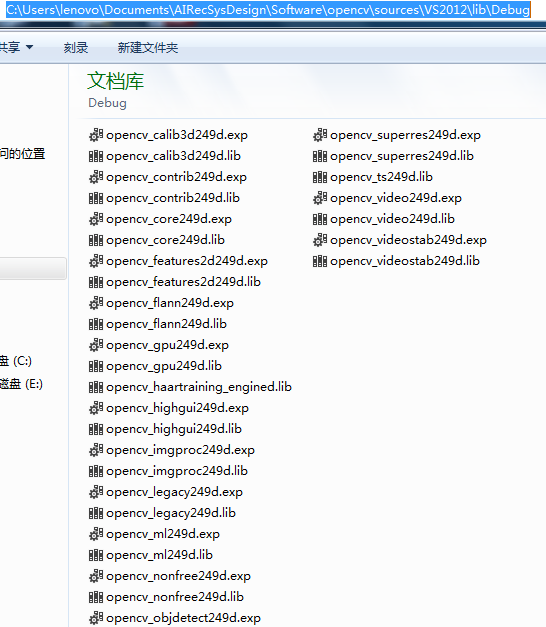


图21. Lib目录下Debug文件夹中的文件

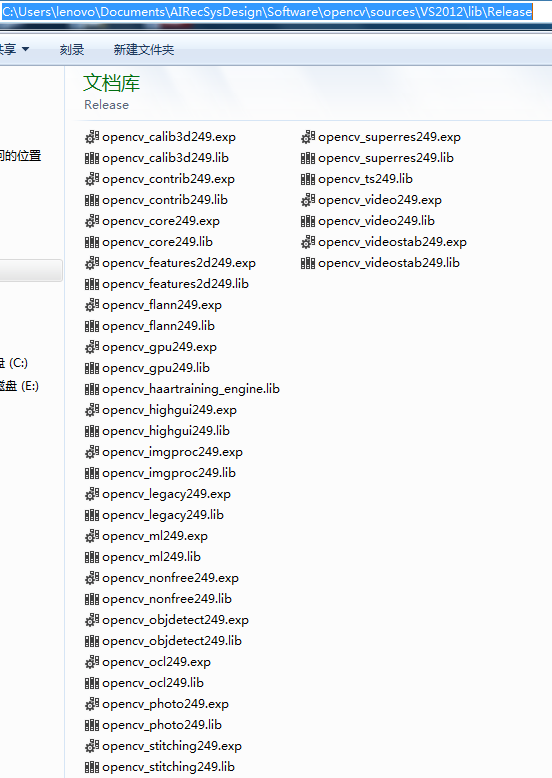


图22. Lib目录下Release文件夹中的文件

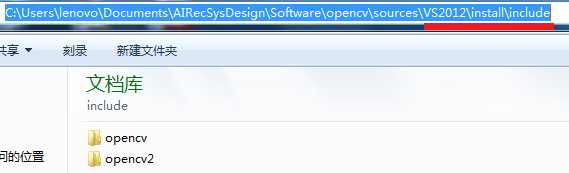


图22. Install\include目录下的文件夹

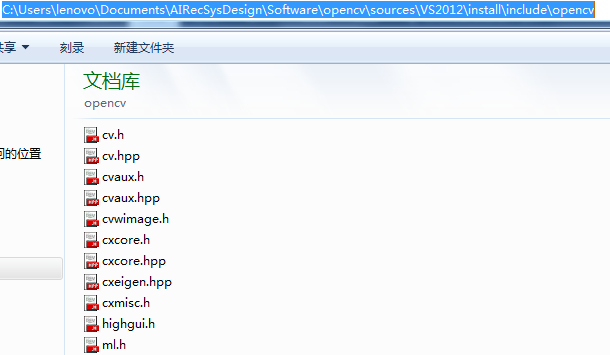


图23. Install\include\opencv目录下的文件夹

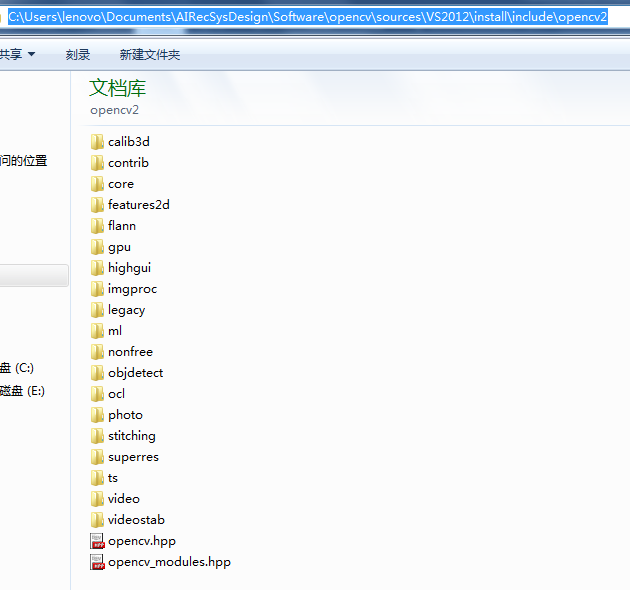


图24. Install\include\opencv2目录下的文件夹

Step 9. 整理关键文件

在上级目录C:\Users\lenovo\Documents\AIRecSysDesign\Software中新建文件夹OpenCV2.4.9，把新生成的lib，bin，include文件统一放在一起。即把

C:\Users\lenovo\Documents\AIRecSysDesign\Software\opencv\sources\VS2012目录下的lib，bin和install目录下的include拷贝到新建的OpenCV2.4.9文件夹下，用于系统配置之用。

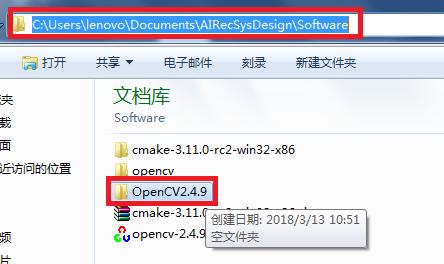


图25. 新建文件夹存放新生成的关键文件

Step 10.把OpenCV2.4.9\bin目录下的Debug和Release目录下的文件都拷贝至bin目录下，然后保留.dll文件，删除其余文件

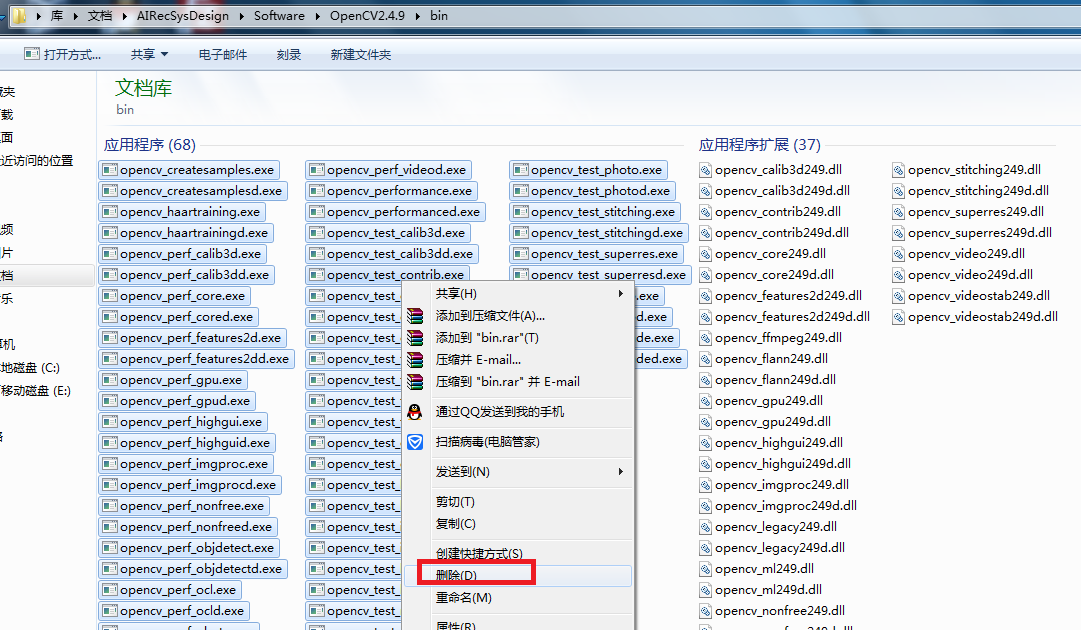


图26. bin目录下，保留dll文件，删除其余文件

Step 11.把OpenCV2.4.9\lib目录下的Debug和Release目录下的文件都拷贝至lib目录下，然后保留.lib文件，删除其余文件

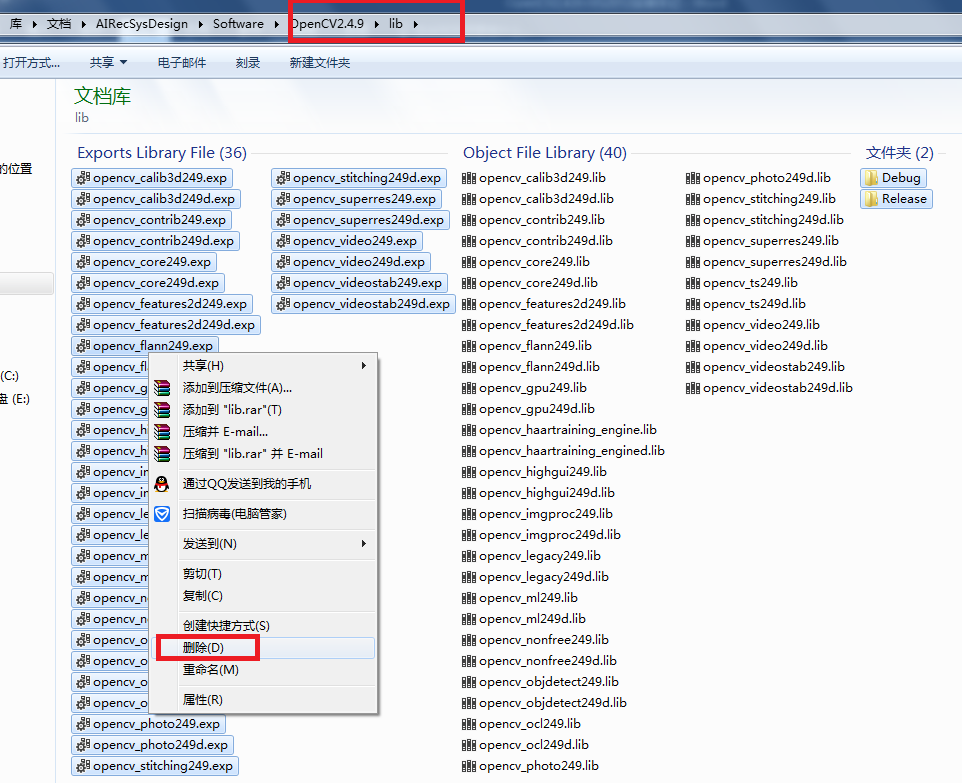


图27. lib目录下，保留lib文件，删除其余文件

Step 12. 至此，我们获得OpenCV2.4.9目录下的三个配置文件夹，三个文件夹的路径分别为

C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\bin

C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\include

C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\lib

其中获得：（在配置时候，不同版本要使用不同版本的库文件）

Degub版本的库文件

opencv\_calib3d249d.lib;opencv\_contrib249d.lib;opencv\_core249d.lib;opencv\_features2d249d.lib;opencv\_flann249d.lib;opencv\_gpu249d.lib;opencv\_haartraining\_engined.lib;opencv\_highgui249d.lib;opencv\_imgproc249d.lib;opencv\_legacy249d.lib;opencv\_ml249d.lib;opencv\_nonfree249d.lib;opencv\_objdetect249d.lib;opencv\_ocl249d.lib;opencv\_photo249d.lib;opencv\_stitching249d.lib;opencv\_superres249d.lib;opencv\_ts249d.lib;opencv\_video249d.lib;opencv\_videostab249d.lib

Release版本的库文件

opencv\_calib3d249.lib;opencv\_contrib249.lib;opencv\_core249.lib;opencv\_features2d249.lib;opencv\_flann249.lib;opencv\_gpu249.lib;opencv\_haartraining\_engine.lib;opencv\_highgui249.lib;opencv\_imgproc249.lib;opencv\_legacy249.lib;opencv\_ml249.lib;opencv\_nonfree249.lib;opencv\_objdetect249.lib;opencv\_ocl249.lib;opencv\_photo249.lib;opencv\_stitching249.lib;opencv\_superres249.lib;opencv\_ts249.lib;opencv\_video249.lib;opencv\_videostab249.lib

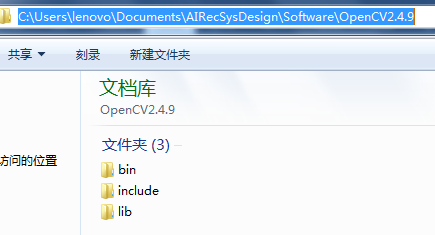


图28. 在OpenCV2.4.9目录下，已获得bin、lib和include三个文件夹及其相关文件

Step 13. 把bin文件夹加入系统路径，按下图操作，即把C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\bin加入系统变量中的Path中。

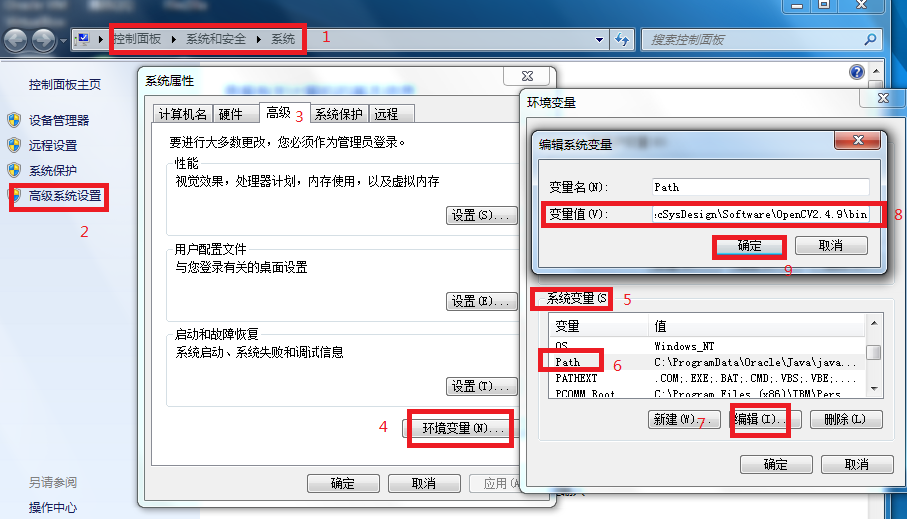


图29. 设置系统路径

Step 14. 新建一个VC++控制台程序，在配置中设置lib和include

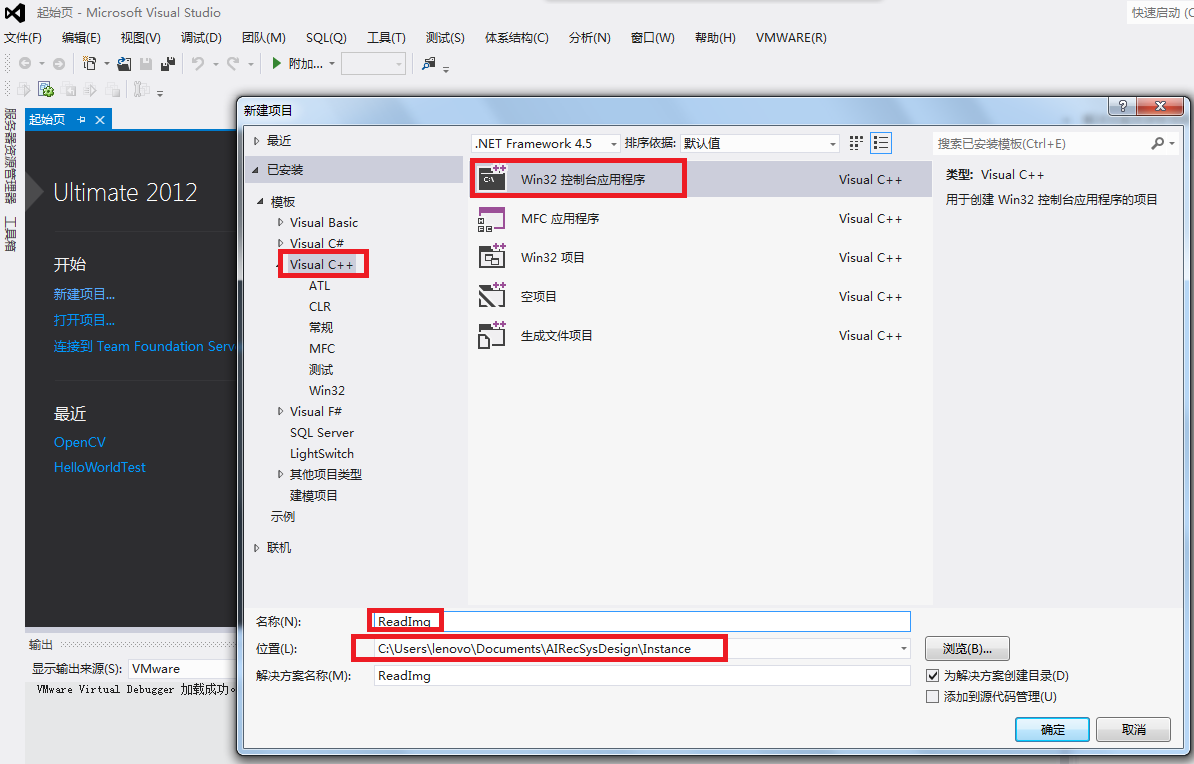


图30. 新建VC++控制台程序

Step 15. 在工程项目解决方案工程名字处右键选择属性

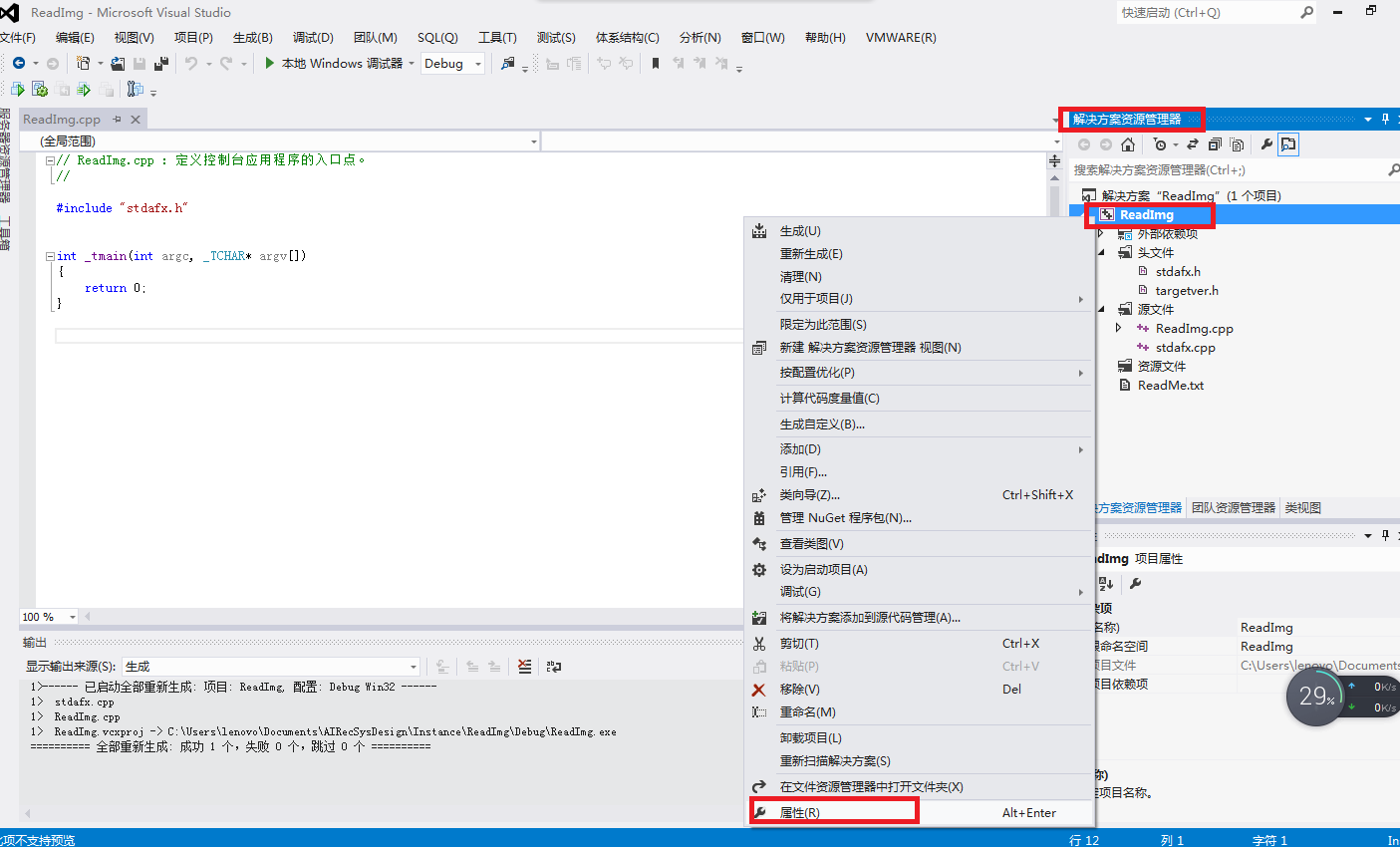


图31. 打开属性页面

Step 16. 在“活动Debug”下，把下面两个路径加入到配置属性中VC++目录的“包含目录”和库目录（追加到最后，注意用；分开）

C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\include

C:\Users\lenovo\Documents\AIRecSysDesign\Software\OpenCV2.4.9\lib

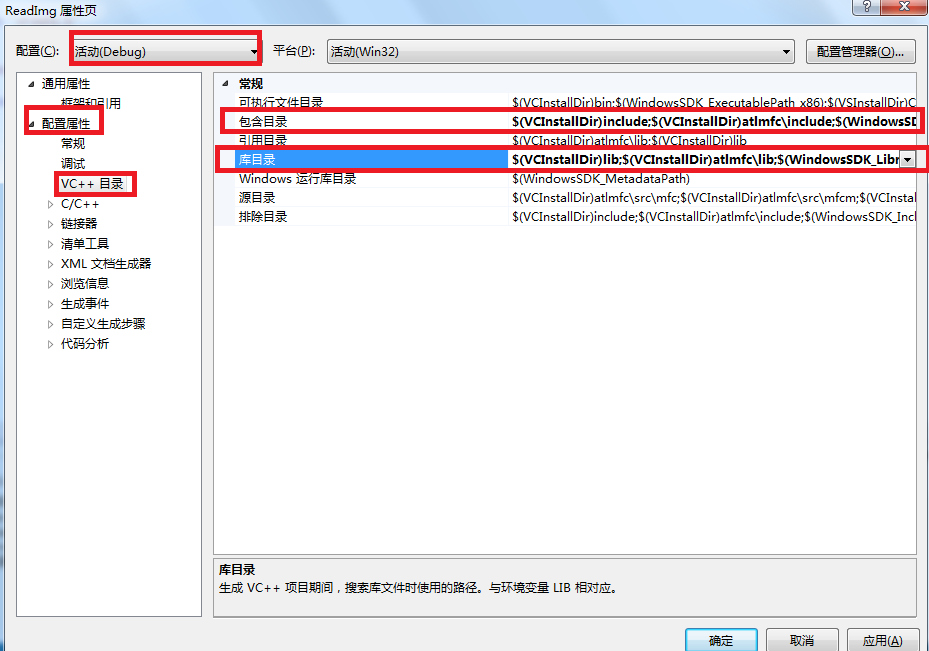


图32. 属性中的VC++目录配置

Step 17. 把Debug版本的静态链接库加入到“属性”中“连接器”的“输入”的“附加依赖项”，即把以下内容拷贝进去。

opencv\_calib3d249d.lib;opencv\_contrib249d.lib;opencv\_core249d.lib;opencv\_features2d249d.lib;opencv\_flann249d.lib;opencv\_gpu249d.lib;opencv\_haartraining\_engined.lib;opencv\_highgui249d.lib;opencv\_imgproc249d.lib;opencv\_legacy249d.lib;opencv\_ml249d.lib;opencv\_nonfree249d.lib;opencv\_objdetect249d.lib;opencv\_ocl249d.lib;opencv\_photo249d.lib;opencv\_stitching249d.lib;opencv\_superres249d.lib;opencv\_ts249d.lib;opencv\_video249d.lib;opencv\_videostab249d.lib;

而附加依赖项原先的内容如下（新追加的内容在最前面加，注意用；隔开）：

kernel32.lib;user32.lib;gdi32.lib;winspool.lib;comdlg32.lib;advapi32.lib;shell32.lib;ole32.lib;oleaut32.lib;uuid.lib;odbc32.lib;odbccp32.lib;%(AdditionalDependencies)

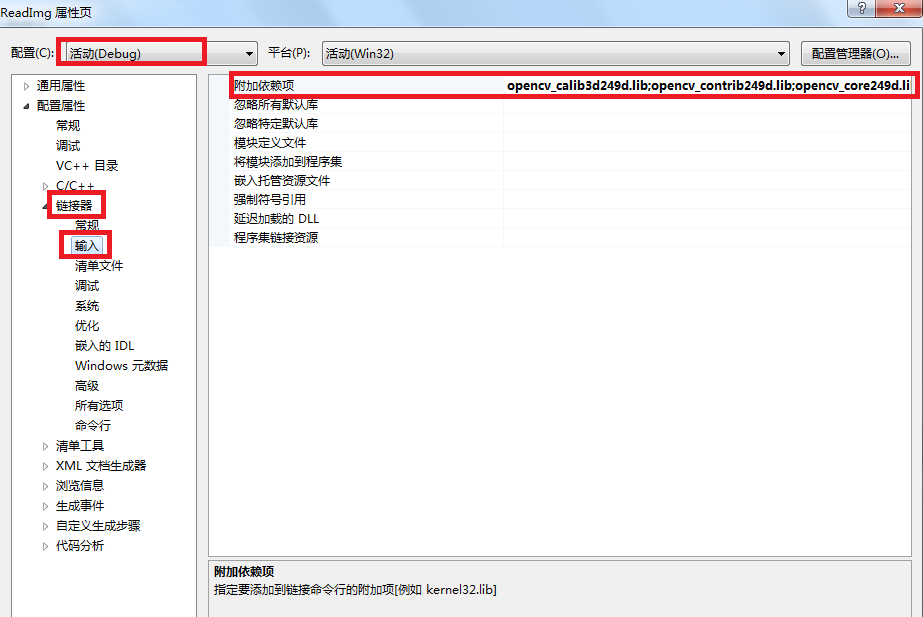


图33. 属性中的链接库设置

Step 18. 按下图加入头文件信息和读取图片的代码，并在项目文件夹下存入图片，即完成OpenCV的配置与图片读取功能。



图34. 图像读取代码

具体代码如下：

#include "stdafx.h"

#include <opencv2/opencv.hpp>

using namespace std;

using namespace cv;

void Read\_Show();

int \_tmain(int argc, \_TCHAR\* argv[])

{

Read\_Show();

return 0;

}

void Read\_Show()

{

const char\* imagename = "boldt.jpg";

//从文件中读入图像

Mat img =imread(imagename);

//如果读入图像失败

if(img.empty())

{

fprintf(stderr, "Can not load image %s\n", imagename);

exit(0);

}

//显示图像

namedWindow("image",1);

cout<<"函数功能：读入并显示和保存一张图像"<<endl;

imwrite("save.jpg",img);

imshow("image", img);

//此函数等待按键，按键盘任意键就返回

waitKey(0);

}