VS2022 下的 OpenCV 的配置（C++）

摘 要

OpenCV 介绍：OpenCV 全称是“Open Source Computer Vision Library”，即开源计算机视觉库。是一个主要针对实时计算机视觉的编程功能库。该库拥有超过 2500 种优化算法，其中包括全面的经典和最先进的计算机视觉和机器学习算法。是做图像处理及计算机视觉必学的库。

VS2022 介绍：Visual Studio 是微软的集成开发环境（IDE），以 Windows 为主要平台开发的一套功能全面而强大的 IDE，支持 Python、F#、VB、C/C++、HTML 等 36 种语言的开发。

本文档介绍了如何在 VS2022 下配置 OpenCV 的方法，使用的 OpenCV 库的版本是 4.6.0。

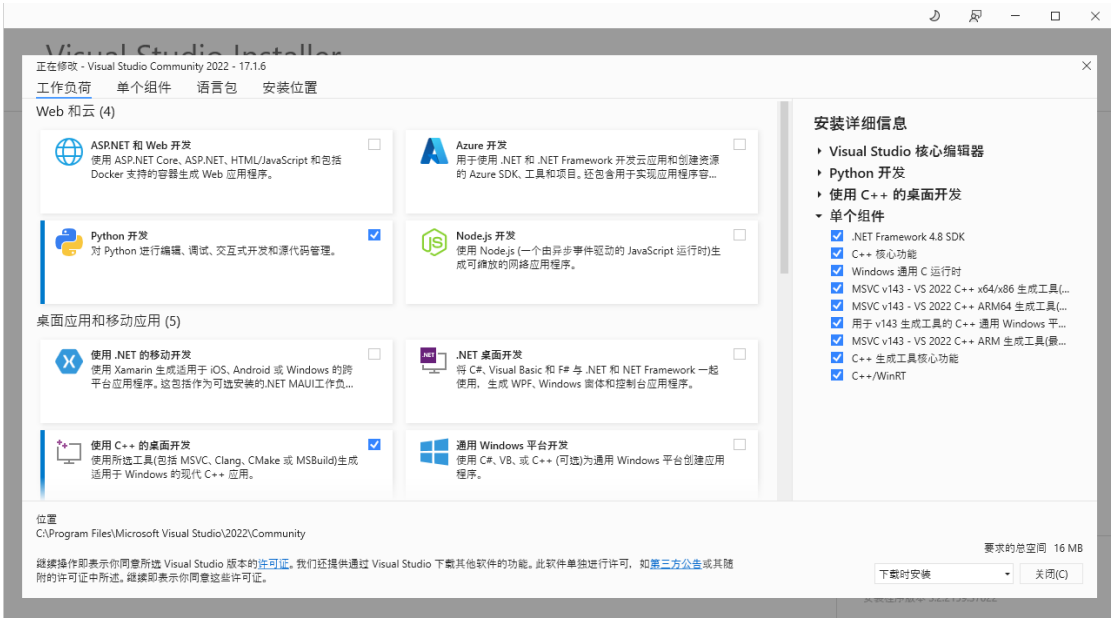
目录

1 下载及安装

1.1 安装 Visual Studio 2022

Visual Studio 2022 下载地址：
https://visualstudio.microsoft.com/zh-hans/downloads/
点击上述地址进入下载页面，选择社区版进行下载

下载完成并安装后进入如下图所示界面

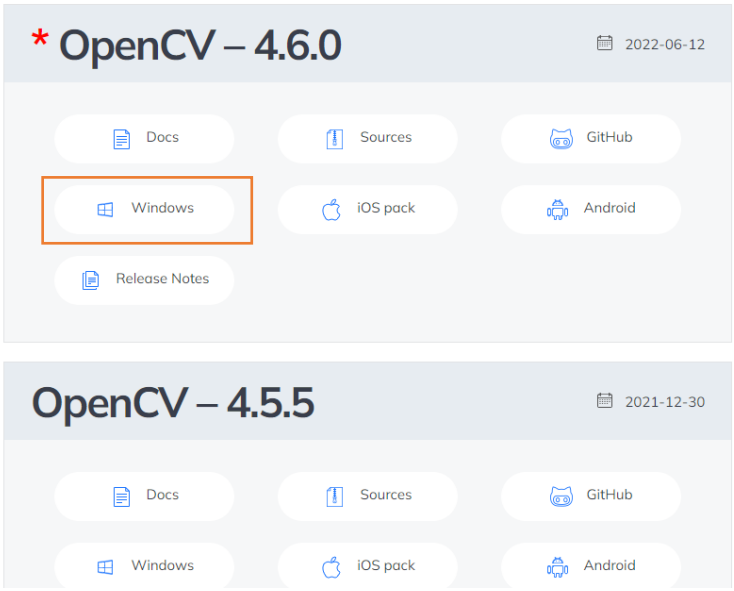


可根据自己的需要选择所需安装的组件，这里我们选择 C++桌面开发即可

之后点击右下角的安装，等待安装完成即可。

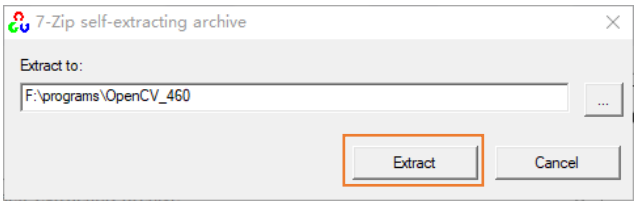
1.2 下载 OpenCV

OpenCV 官网：https://opencv.org/
下载页面：https://opencv.org/releases/



选择需要的版本进行下载，我们这里使用的是 4.6.0 的 windows 版，在此页面还可以下载 OpenCV 的 Sources 源码，和查看说明文档（Docs）。

下载完成后双击下载的文件，设置合适的路径后点击 Extract

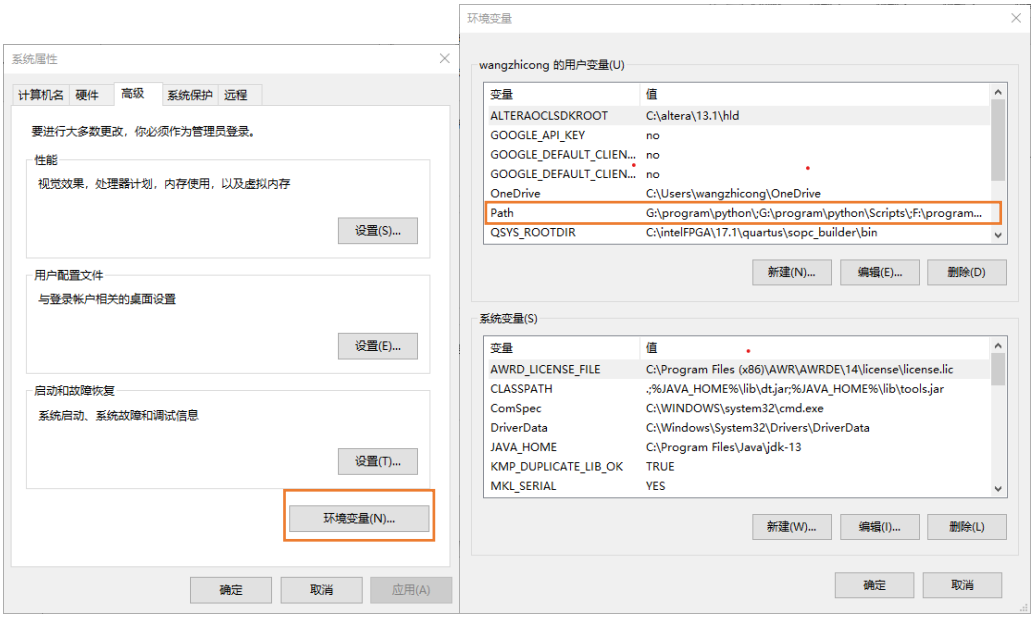


这里我们放在“F:\programs\OpenCV_460”。

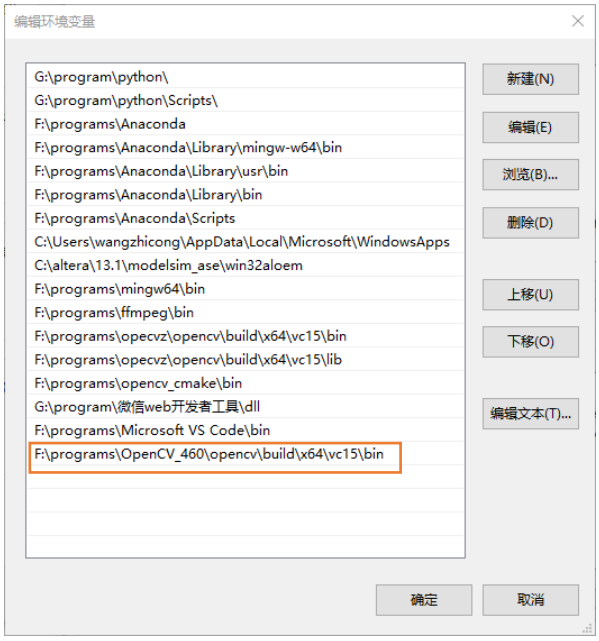
1.3 添加系统变量

使用 OpenCV 需要调用 OpenCV 的 dll 静态库文件，在 windows 下使用 dll 静态库文件会按照以下顺序寻找（程序当前目录 > System32 目录 > 环境变量 Path 所设置路径）。因此，我们把 OpenCV 的库路径添加到系统变量 Path 中。

“此电脑”→右键“属性”→“高级系统设置”→“高级”→“环境变量”→“Path”。如下图：



双击“Path”把“F:\programs\OpenCV_460\opencv\build\x64\vc15\bin”（此处添加自己设置的路径）添加入“Path”，如果是 VS2022，则对应的 VC 版本为 VC15。如下图所示：

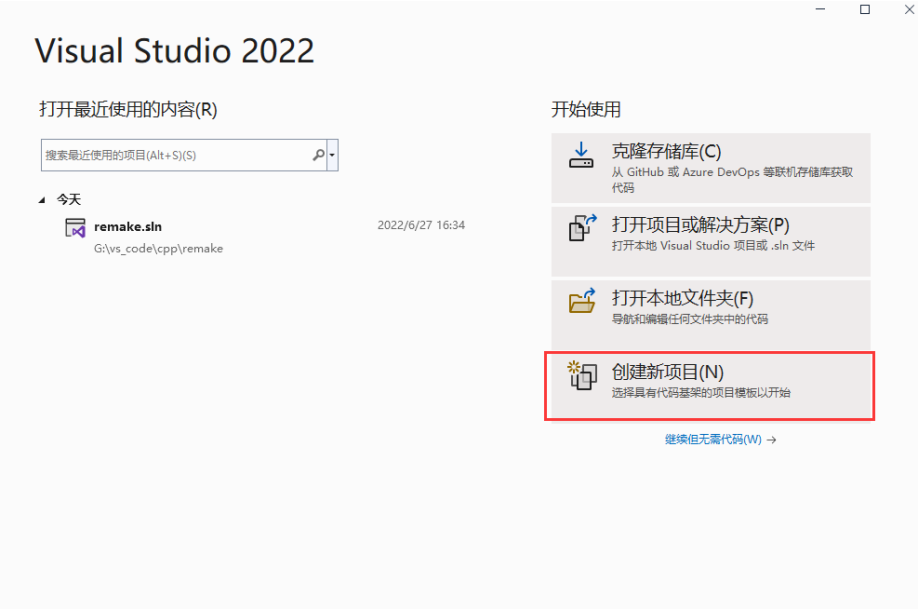


之后点击确定即可完成环境变量的配置。

2 建立属性表

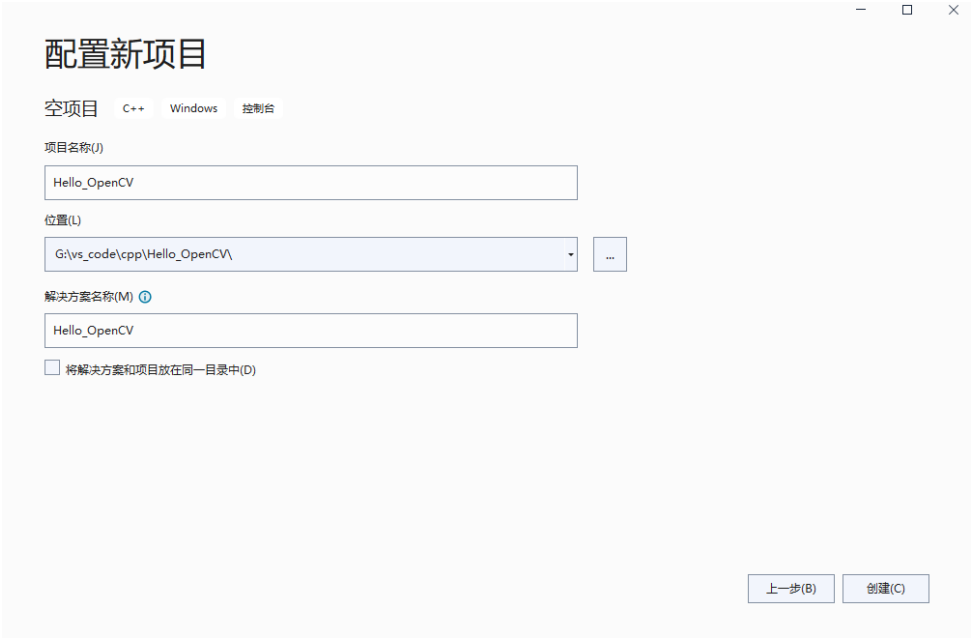
2.1 新建项目

打开 VS2022，点击创建新项目

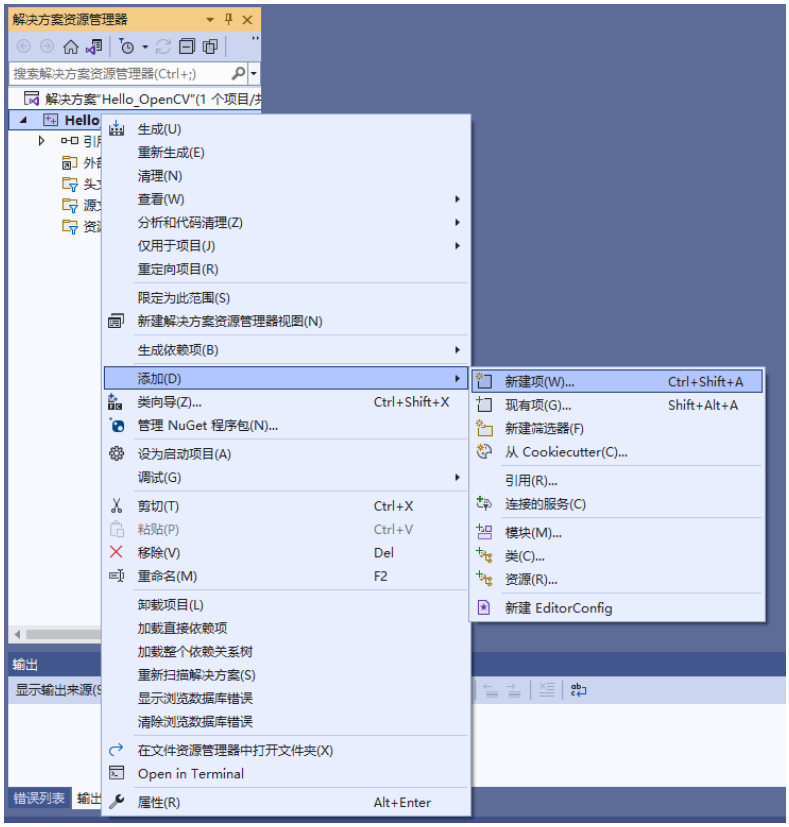


选择空项目，点击下一步

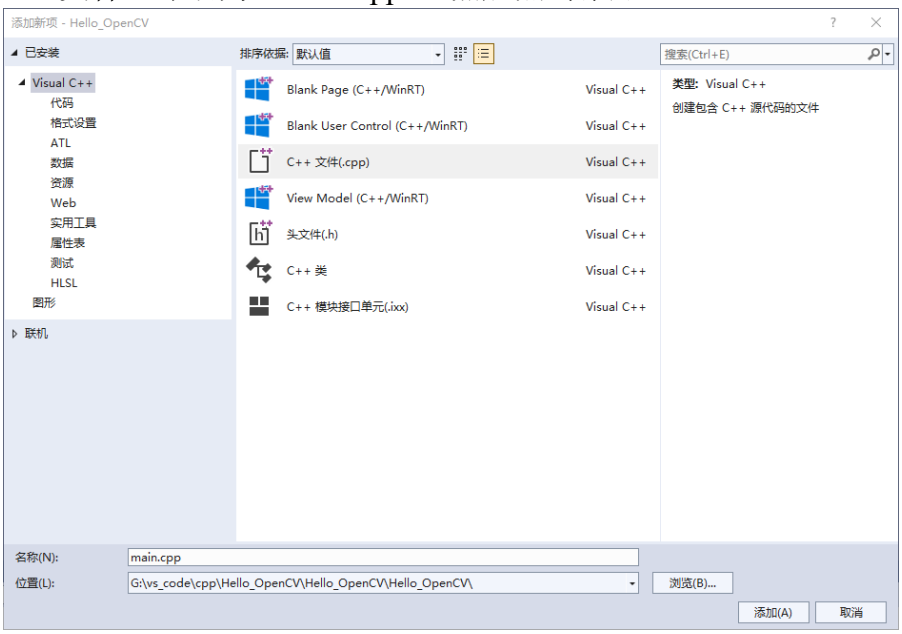
为项目命名并选择合适路径后点击创建



这里我们命名为 Hello_OpenCV，创建完成之后右键“Hello_OpenCV”→“添加”→“新建项”。

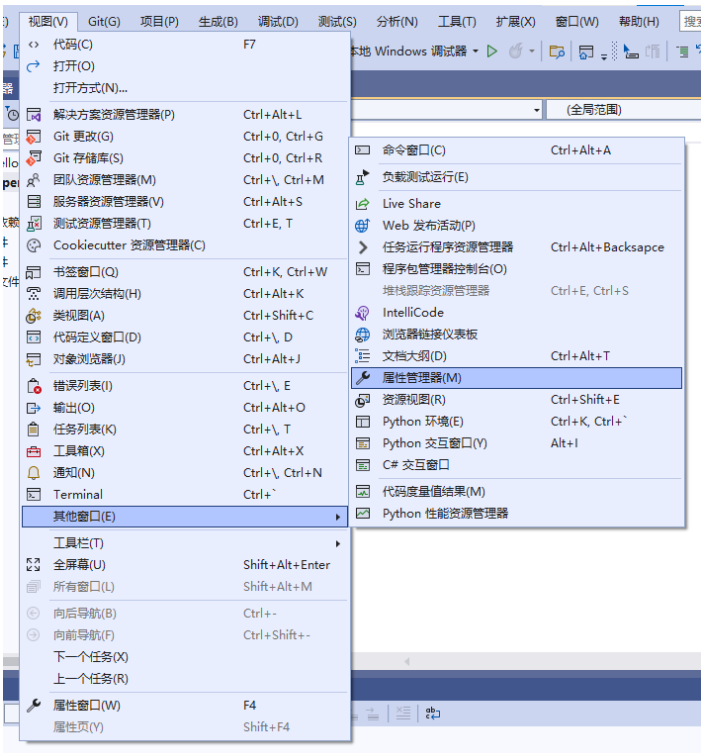


选择 C++文件，命名为“main.cpp”，然后点击添加

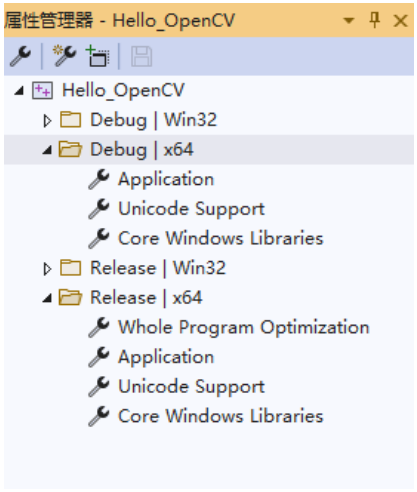


2.2 Debug 模式的属性表

首先打开属性管理器，“视窗”→“其他窗口”→“属性管理器”。



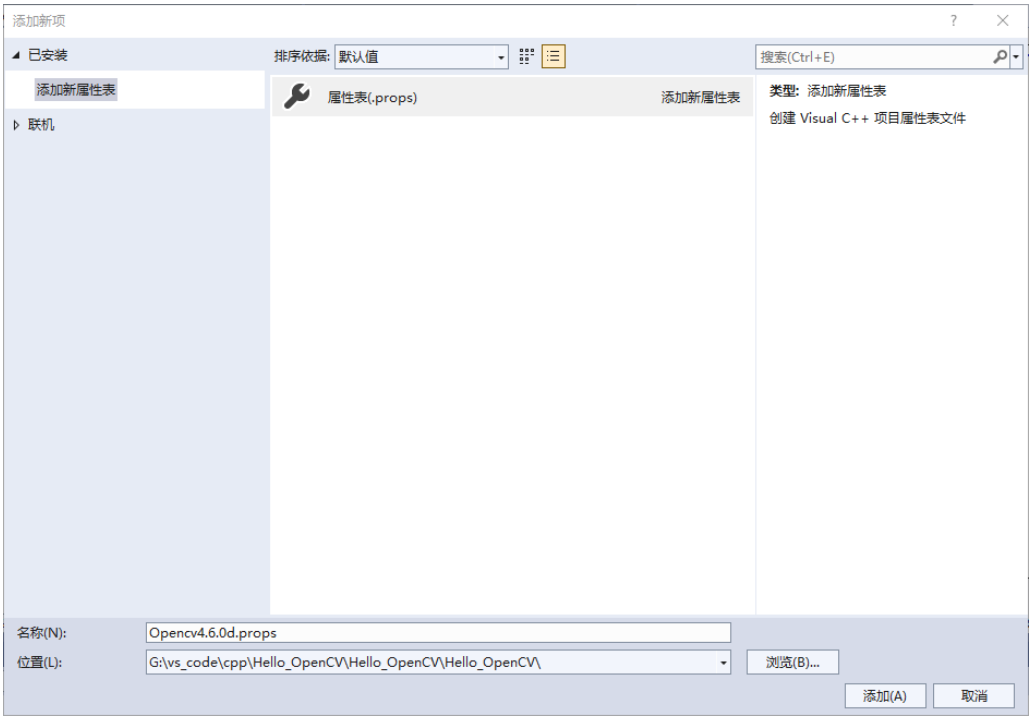
之后出现属性管理器窗口，如下图：



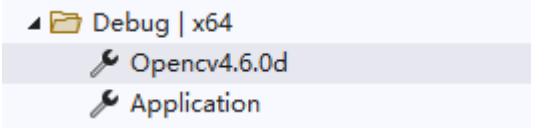
我们使用的 x64 平台，所以这里要对“Debug | x64”和“Release | x64”进行配置。

右键“Debug | x64”→“添加新的属性表”。

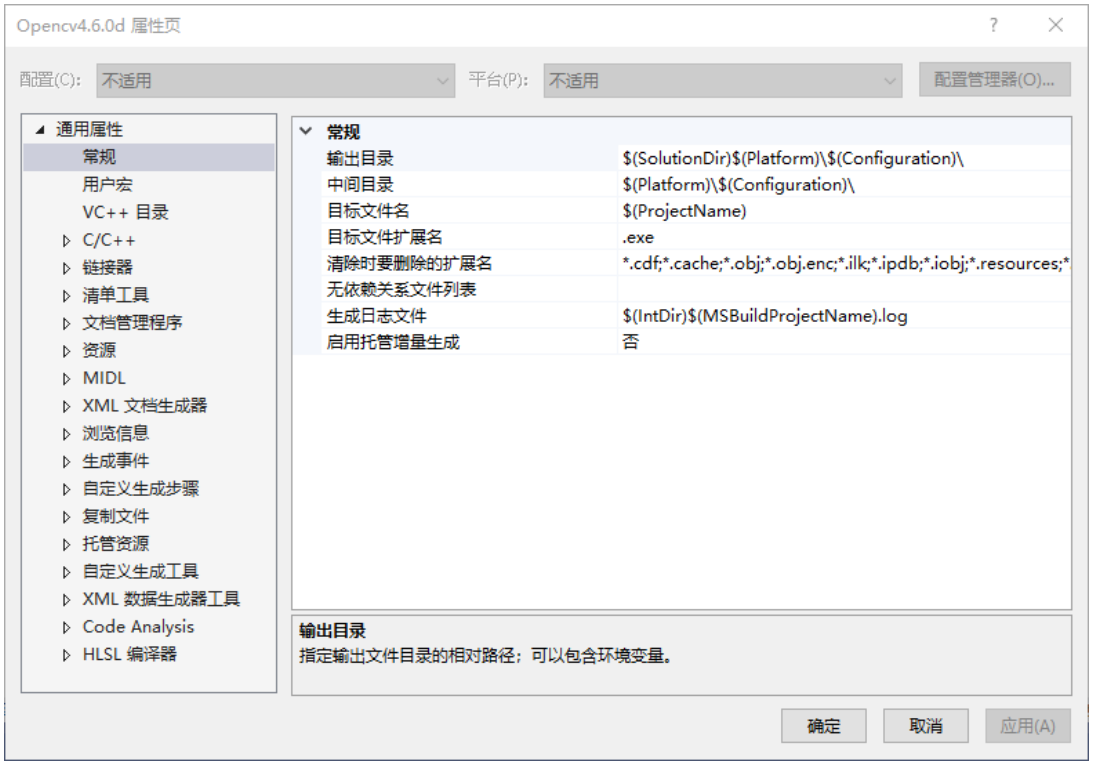
名称改为“Opencv4.6.0d.props”，因为可能会建立不同的 OpenCV 版本的属性表，所以取名时带上版本号方便区分，后缀 d 表示 debug 模式的版本号。如下图：



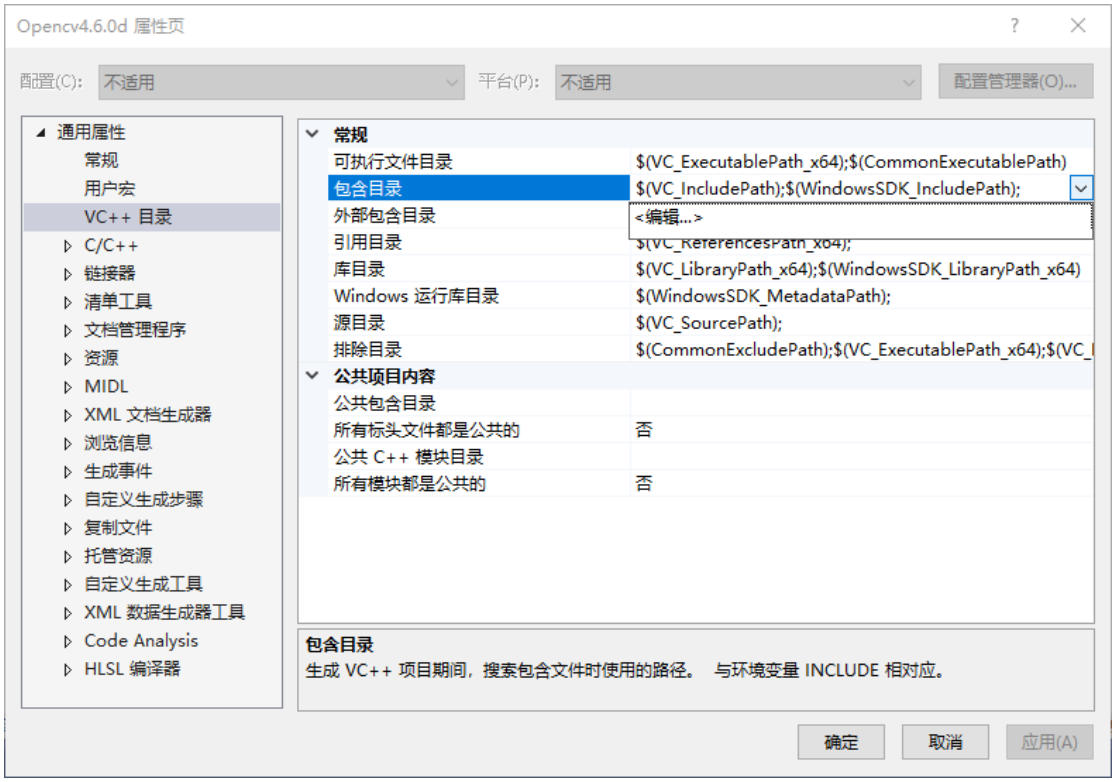
点击“添加”，我们将会看到“Debug | x64”下出现了我们所创建的属性表：



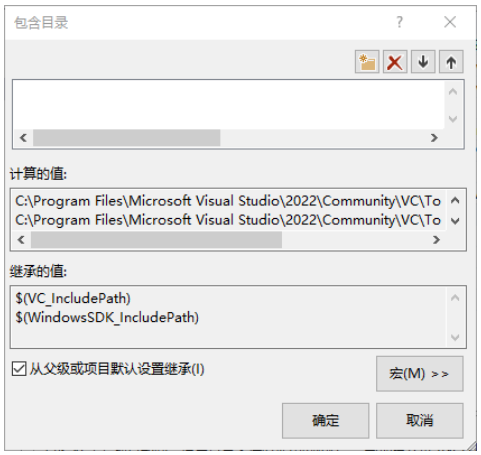
双击“Debug | x64”下的“Opencv4.6.0d”，出现如下图所示的属性页面：



首先添加包含目录，“VC++目录”→“包含目录”→点击右侧小三角→“编辑”



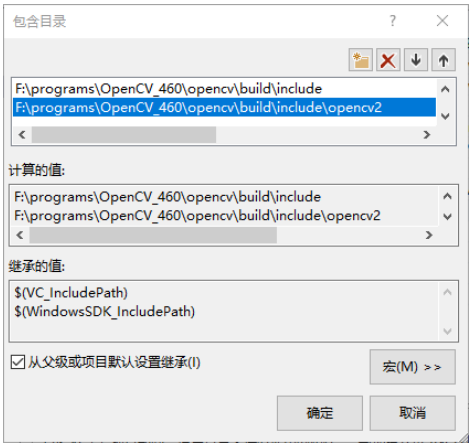
点击右边文件夹符号



添加两个路径

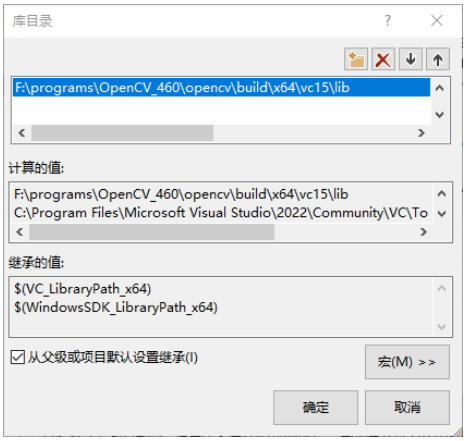
F:\programs\OpenCV_460\opencv\build\include

F:\programs\OpenCV_460\opencv\build\include\opencv2

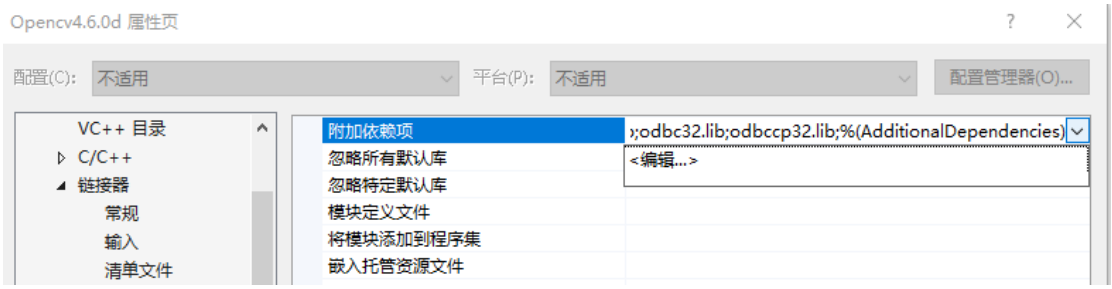


然后添加库目录，“VC++目录”→“库目录”→点击右侧小三角→“编辑”。

添加“F:\programs\OpenCV_460\opencv\build\x64\vc15\lib”路径。

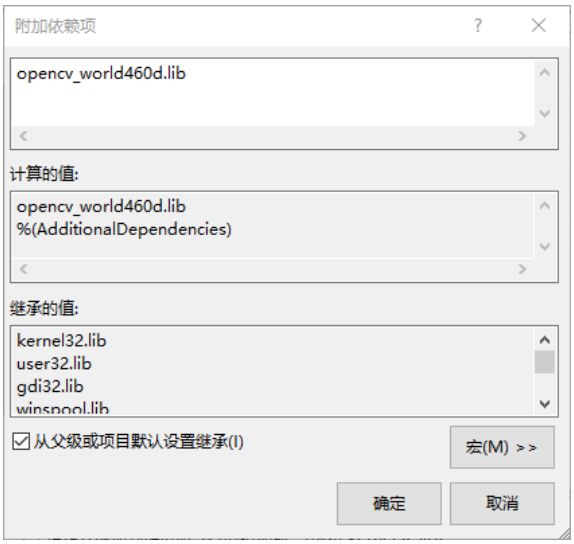


最后添加附加依赖项，“链接器”→“输入”→“附加依赖项”→点击右侧小三角→“编辑”。



输入“opencv_world460d.lib”

此处为 Debug 模式配置方法，若是在 Release 模式下，其他步骤完全一样，只是在这一步的时候输入“opencv_world460.lib”，后面没有 d 后缀。这里如果输入错误，程序会出现 bug。（d 指 Debug，没有 d 则指 Release）



依赖项文件名对应 OpenCV 版本号，会有不同，比如我们配置 Opencv4.6.0 的版本，这里就需要输入“opencv_world460d.lib”， 完成后，在属性页面，点击“应用”再点击“确定”。

之后在“属性管理器”中右键“Opencv4.6.0d”→“保存 Opencv4.6.0d”这样，我们做的设置全部保存到属性表中。以后新建项目时，不需要重新配置，只需要加载属性表文件即可。

2.3 Release 模式的属性表

重复 2.2 节内容，生成一个 Release 模式下用的属性表，命名为“Opencv4.6.0.props”没有后缀 d。其他步骤相同，只是在增加附加依赖项这一步中输入“opencv_world460.lib”，后面没有 d 后缀。

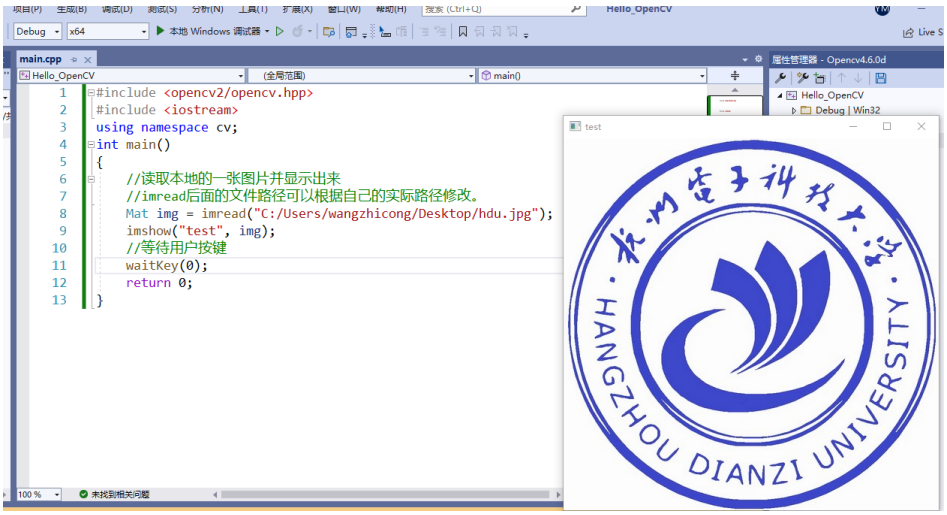
3 测试程序

3.1 生成测试程序

将如下代码拷贝至刚才生成的“main.cpp”文件中，imread 后面的文件路径可以根据自己的实际路径修改。

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace cv;
int main()
{
    //读取本地的一张图片并显示出来
    //imread 后面的文件路径可以根据自己的实际路径修改。
    Mat img = imread("C:/Users/wangzhicong/Desktop/hdu.jpg");
    imshow("test", img);
    //等待用户按键
    waitKey(0);
    return 0;
}
```

编译运行程序，如果可以成功显示图片，说明配置成功。

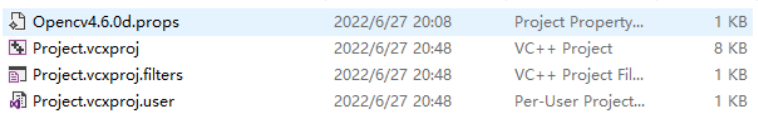


3.2 加载属性表

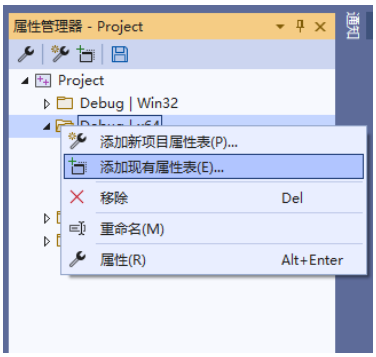
每新建一个项目之后，都需要将之前保存的属性表“Opencv4.6.0d”加载一遍。

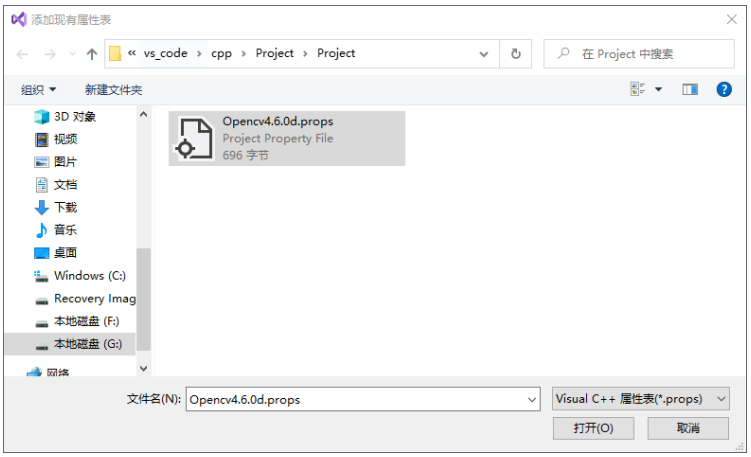
属性表保存在工程所在路径中，如下图：

首先，建一个项目“Project”，然后将之前保存好的属性表复制到工程目录中，如图：

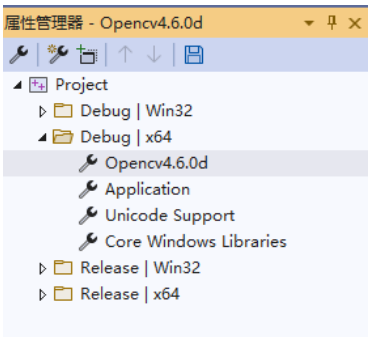


然后，打开属性管理器。右键“Debug | x64”→“添加现有属性表”找到属性表“Opencv4.6.0d”→“打开”





这时候就会发现这个属性表已经被添加到 Debug | x64 中了，如下图：



如果是 Release 版的话，和 debug 版一样，只需把之前配置好的属性表“Opencv4.6.0”添加到 Release | x64 即可。

附　录

包含目录：搜索在源代码中引用的包含文件的目录，即寻找#include<xxxx.h>中的 xxxx.h 的搜索目录。Opencv 中包含很多这样的头文件。

引用目录：搜索通过#using 指令在源代码中引用的程序集和模块（元数据）文件的目录。对应于 LIBPATH 环境变量。

库目录：搜索所包含静态链接库（lib 文件）的目录。与环境变量 LIB 相对应。

版本

时间	版本	作者	备注
2022/6/27	2.0	李竹	