最近整理了一下这两年一直在用的摄像机定标与立体匹配测试程序，将代码进行了重构，界面也做了调整，分享出来方便有需要的朋友使用。当然我的编程能力有限，程序可能还有各种bug，请大家多多包涵。相关问题欢迎留言或email联系讨论，谢谢！

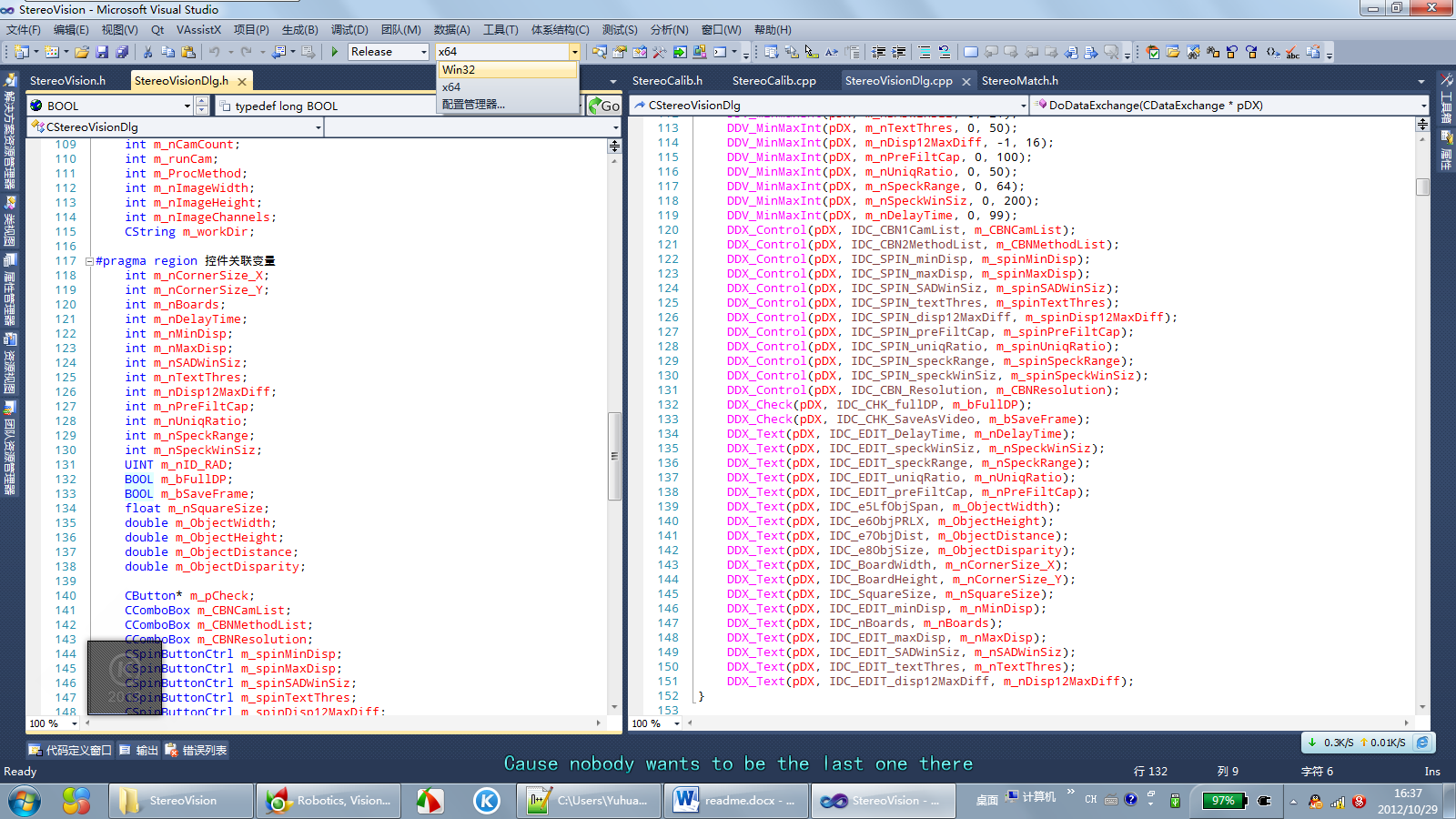
源码下载与Bug汇集请见：

<http://yuhuazou.sinaapp.com/2012/10/30/a_simple_camera_calibration_stereo_match_program_mfc_opencv/>

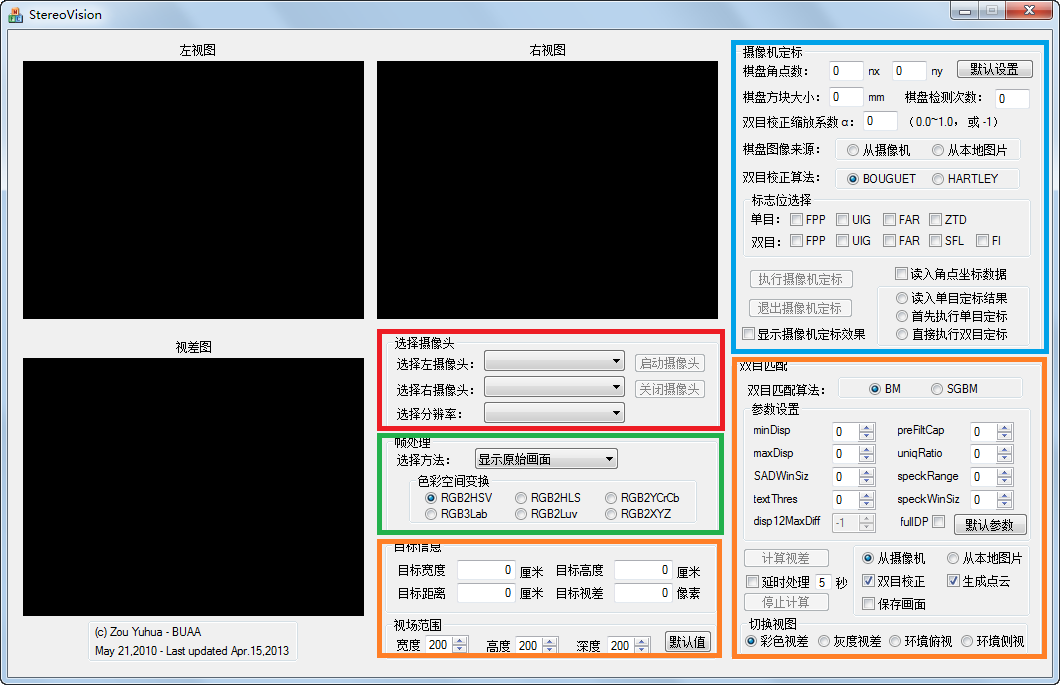
1. 原始编译环境

* Windows 7 64位旗舰版
* OpenCV SVN （v2.4.9）
* x64 应用

1. 编译修改说明
2. 双击 .sln 文件打开解决方案
3. 在项目属性的 “VC++ 目录”中分别修改OpenCV相关的 include 路径和 lib 路径；directshow的路径使用了相对路径，不需要修改
4. 在项目属性的“连接器--输入—附加依赖项”修改OpenCV各项lib的版本号
5. 在“解决方案平台”中选择“win32”或“x64”，分别对应32位系统和64位系统



1. 使用说明
2. 界面



红色：摄像头初始化区域； 绿色：图像基本处理区域；

蓝色：摄像头定标区域； 橙色：双目匹配计算区域；

1. 操作步骤
2. 摄像头初始化
3. 选择拟打开的摄像头：

* 当电脑接入多个摄像头时：A.从“选择左/右摄像头”下拉菜单中选择该视角应该对应的摄像头设备名称（序号）；B. 注意不同视角的摄像头设备序号不能相同，若相同会弹出警告窗口。
* 当电脑只有一个摄像头接入时：直接从“选择左摄像头”下拉菜单中选择摄像头设备名称（序号）。

1. 选择摄像头的分辨率：

* 从“选择分辨率”下拉菜单中选择，目前有三种分辨率：640\*480、352\*288、320\*240.

1. 启动摄像头：

* 点击“启动摄像头”按钮，界面上各功能按钮生效，即可进行摄像头定标、双目匹配等操作。
* 点击“关闭摄像头”按钮，可以重新调整摄像头的配置。

1. 摄像机定标
2. 点击“默认设置”，加载默认参数和选项；
3. 按照实际需要调整相关参数和选项；
4. 棋盘角点数：nx和ny分别是棋盘上横向和纵向的角点个数，注意必须按照实际棋盘横向和纵向的最大角点数输入，否则会造成角点检测失败；
5. 棋盘方块大小：棋盘方块的实际大小，毫米为单位；
6. 棋盘检测次数：以10-20次为宜；
7. 双目校正缩放系数：取值范围为0至1，或者 -1。具体含义参见：<http://blog.csdn.net/chenyusiyuan/article/details/8131496>
8. 棋盘图像来源：默认为“从摄像机”，也可选择“从本地图片”；若选择“从本地图片”读入，在执行定标时，将弹出对话框分别选择左视图图像和右视图图像；注意左、右视图的图像数均需要至少4张；当左、右视图图像数与棋盘检测次数不一致时，取三者最小值为准；
9. 双目校正算法：默认为“BOUGUET”，也可选择“HARTLEY”；不过“HARTLEY”方法的双目校正尚未进行测试，可能存在bug；
10. 标志位选择：各标志位的设置请参见OpenCV相关文档；

FPP: CV\_CALIB\_FIX\_PRINCIPAL\_POINT

UIG: CV\_CALIB\_USE\_INTRINSIC\_GUESS

FAR: CV\_CALIB\_FIX\_ASPECT\_RATIO

ZTD: CV\_CALIB\_ZERO\_TANGENT\_DIST

SFL: CV\_CALIB\_SAME\_FOCAL\_LENGTH

FI: CV\_CALIB\_FIX\_INTRINSIC

1. 读入角点坐标数据：如果曾进行过摄像机定标操作，程序文件夹内将保存有上一次操作的棋盘角点数据的文件（文件名为CornerData.yml）。点击该选项，将自动加载上一次操作的棋盘角点数据，跳过“棋盘角点检测”这一步，直接进行摄像机定标。如果文件夹内没有CornerData.yml文件或者文件读取失败，将弹出警告窗口，应检查该文件是否存在、或者不选择此项，重新进行棋盘角点检测；
2. 读入单目定标结果：将从本地文件（cameraParams\_left.yml和cameraParams\_right.yml）中加载摄像机参数，然后进行摄像机定标；如果加载失败，将弹出警告窗口；
3. 首先执行单目定标：在获取棋盘角点数据后，先调用cv::calibrate- Camera函数对左、右摄像机分别进行定标，获取摄像机内部参数，并将参数保存到 cameraParams\_left/right.yml文件中。然后再执行双目定标（当电脑接入的摄像头数大于1个时）；
4. 直接执行双目定标：在获取棋盘角点数据后，直接调用stereoCalibrate函数进行双目定标（当电脑接入的摄像头数大于1个时）；定标结果保存到calib\_paras.xml文件中。

注意：H、I、J项是三选一的选项。

1. 显示摄像机定标效果：如果仅接入一个摄像头，则显示单目定标后消除畸变的效果；否则显示双目定标后左右视图畸变消除和行对准的效果。
2. 点击“执行摄像机定标”，开始定标操作，程序将依次执行以下工作：
3. 棋盘角点检测
4. 单目定标
5. 双目定标
6. 双目校正
7. 显示定标效果
8. 点击“退出摄像机定标”，完成或中止定标操作，恢复摄像机正常显示。该按钮仅在“棋盘角点检测”和“显示定标效果”阶段有效。
9. 双目匹配
10. 选择“双目匹配算法”：“BM”或“SGBM”，点击后将清零相关的参数；
11. 点击“默认配置”：载入所选算法的默认参数；
12. 视需要调整参数，可以在计算视差期间实时调整各项参数；
13. “参数设置”中各项参数的意义和取值请参见OpenCV相关文档；
14. 选择图像来源：“从摄像机”或“从本地图片”；
15. “双目校正”：点选后，将需要手动选择摄像机定标文件，然后程序将根据定标参数对图像进行校正，再进行双目匹配和视差计算。若不点选，则程序直接对原始的左右视图进行视差计算，这种情况一般用于从本地读取下载好的公共测试图像，这些公共测试图像一般都已经过标定和行对齐，可直接用于视差计算；
16. “三维点云”：点选后，将根据视差图生成三维点云，并检测距离最近的物体；但若读入的摄像机定标文件中记录的校正方法不是 BOUGUET 方法，则无法生成点云；
17. “保存画面”：点选后，将自动保存每帧左、右视图和视差图到本地；
18. “延时处理”：点选后，将在计算完每帧视图的视差图和三维点云后，延时一定时间，再进入下一帧的处理；延时时间默认为 5 秒；可自行调整；
19. “切换视图”：在“视差图”窗口中显示 彩色视差、灰度视差、环境俯视、环境侧视 图像；
20. “视场范围”：设置视场的宽度、高度和深度范围。
21. 点击“计算视差”，将执行如下操作：
22. 若选择“从本地图片”载入左右视图，则将弹出对话框分别选择左右视图图像，可单选或多选；
23. 若点选了“双目校正”，将弹出对话框，由用户选择双目摄像机标定参数文件（文件名一般为calib\_paras.xml），程序将分析相关参数；
24. 程序进入循环，自动从本地或摄像机加载图像，进行双目匹配和视差计算等。
25. 如果点选了“生成点云”，程序将根据视差图生成三维点云，并检测出距离最近的物体，在界面中显示物体的相关信息。
26. 点击“停止计算”，退出双目匹配和视差计算操作，恢复摄像机正常显示。
27. 帧处理测试

此处可进行各种简单的图像处理操作，目前有canny边缘检测、直方图平衡、颜色空间转换等。

1. 问题

编译时遇到了missing ';' before identifier 'PVOID64 的错误，根据网上的资料来看，大致有两种方法：  
１.　可以“通过调整IDE 工具-> 选项-> 项目vc++目录;包含文件include 的顺序来解决这类问题   
使dxsdk需要的winnt.h所在文件夹在最上面。” from http://hagejid.blog.51cto.com/141754/294797/  
但是这个方法在我这边不管用。。  
  
２.　第二种直接修改winnt.h  
“在winnt.h文件中修改：  
添加：#define POINTER\_64 \_\_ptr64  
在typedef void \*PVOID; typedef void \* POINTER\_64 PVOID64;这两句之前  
原因opencv的定义与微软的有差异，而引起的。”　（from http://blog.csdn.net/yihaizhiyan/article/details/5776000）