

工业相机采集保存图片使用说明

【摘要】

本文档介绍了基于工业相机SDK开发的单相机采集并保存图像Demo的使用。工业相机分为GigE相机和U3V相机。本文主要描述如何加载相机的库文件以及对工业相机SDK基本接口的使用。

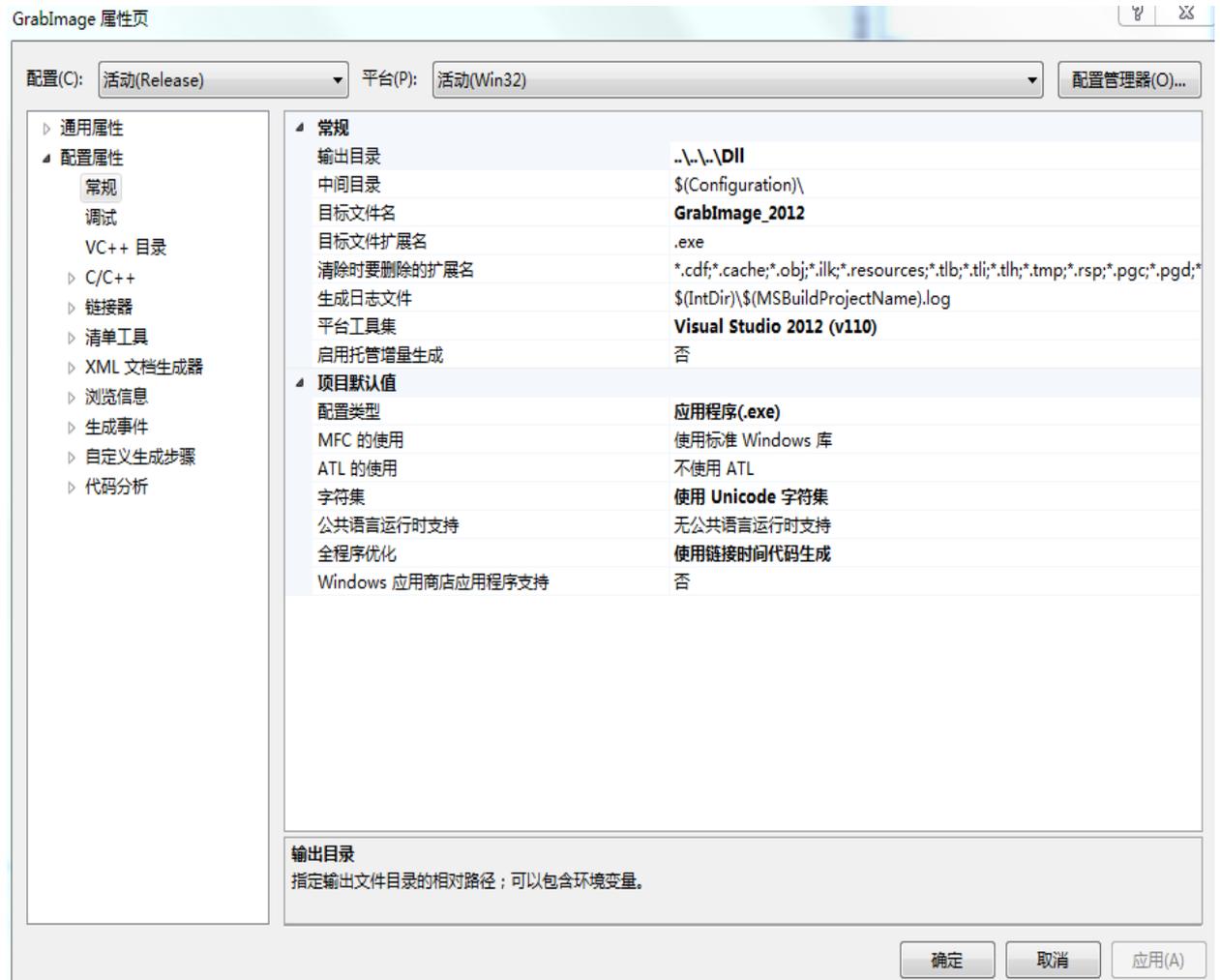
本文介绍的采集Demo 为VS2012 下开发的：

一. 如何加载相机采集所需库文件

1. 需要包含的头文件：

MVS\Includes 下所有头文件，加载方法为：

在 vs2012 下， 属性\配置属性\C/C++\附加包含目录 中加入路径 ..\..\..\Includes (Release 和 debug 模式下都需要添加)



2. 需要加载的 lib 文件，路径为 MVS\Libraries

vs2012 下加载：MvCameraControl.lib

加载方法为：

属性\配置属性\链接器\常规\附加库目录 中加入 ..\..\..\Libraries

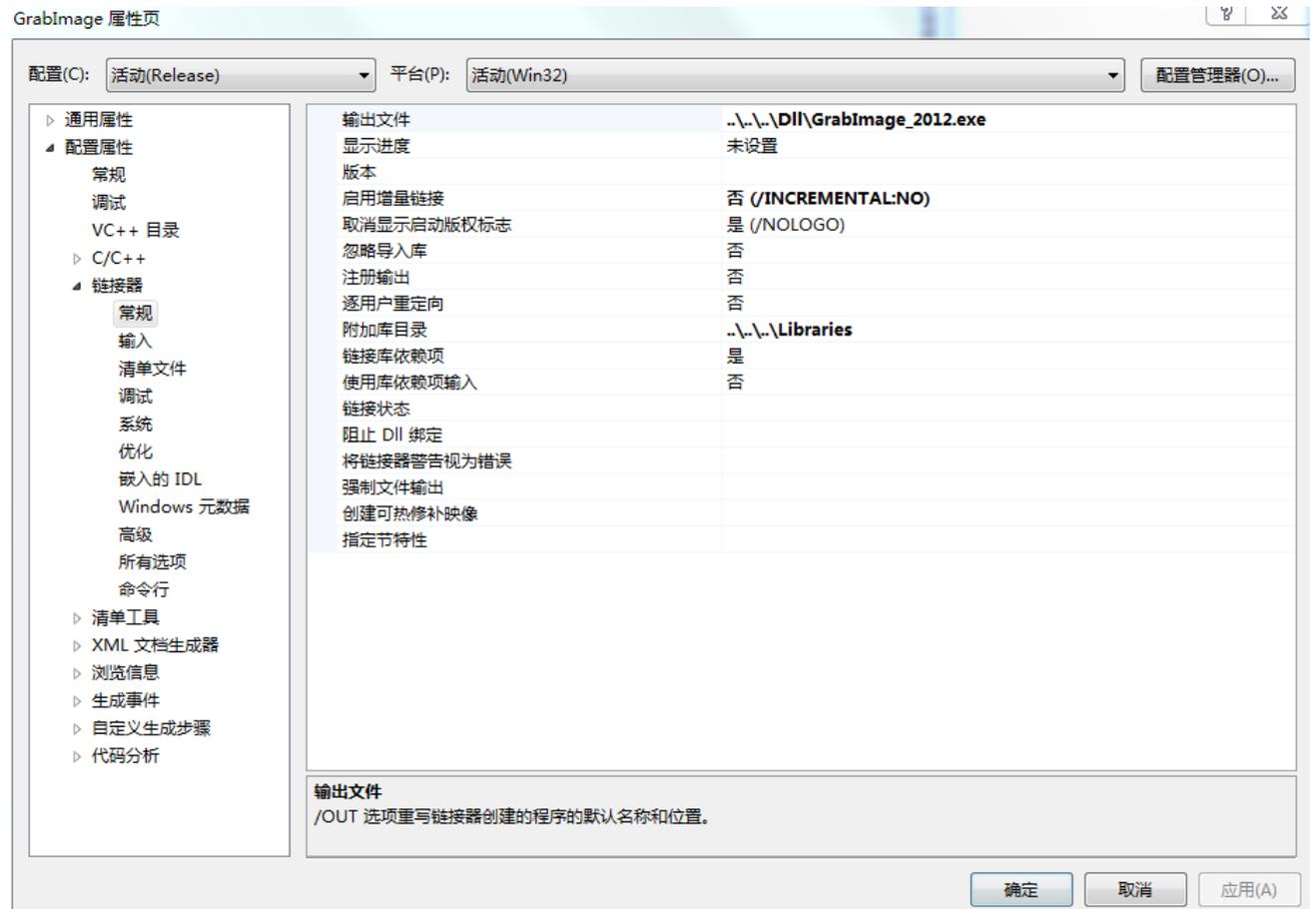
属性\配置属性\链接器\输入\附加依赖项 中加入 MvCameraControl.lib(Release 和 debug 模式下都需要添加)

3. 需要加载输出文件路径，路径为 MVS\D11

加载方法为：

属性\配置属性\链接器\常规\输出文件 中加入 ..\..\..\D11\GrabImage_2012.exe

属性\配置属性\常规\输出目录 中加入 ..\..\..\..\D11(Release 和 debug 模式下都需要添加)



GrabImage 属性页

配置(C): 活动(Release) 平台(P): 活动(Win32) 配置管理器(O)...

- ▷ 通用属性
- ▲ 配置属性
 - 常规
 - 调试
 - VC++ 目录
 - ▷ C/C++
 - ▲ 链接器
 - 常规
 - 输入**
 - 清单文件
 - 调试
 - 系统
 - 优化
 - 嵌入的 IDL
 - Windows 元数据
 - 高级
 - 所有选项
 - 命令行
 - ▷ 清单工具
 - ▷ XML 文档生成器
 - ▷ 浏览信息
 - ▷ 生成事件
 - ▷ 自定义生成步骤
 - ▷ 代码分析

附加依赖项	MvCameraControl.lib
忽略所有默认库	
忽略特定默认库	
模块定义文件	
将模块添加到程序集	
嵌入托管资源文件	
强制符号引用	
延迟加载的 DLL	
程序集链接资源	

附加依赖项
指定要添加到链接命令行的附加项(例如 kernel32.lib)

确定 取消 应用(A)

GrabImage 属性页

配置(C): 活动(Release) 平台(P): 活动(Win32) 配置管理器(O)...

- ▷ 通用属性
- ▲ 配置属性
 - 常规**
 - 调试
 - VC++ 目录
 - ▷ C/C++
 - ▲ 链接器
 - 常规
 - 输入
 - 清单文件
 - 调试
 - 系统
 - 优化
 - 嵌入的 IDL
 - Windows 元数据
 - 高级
 - 所有选项
 - 命令行
 - ▷ 清单工具
 - ▷ XML 文档生成器
 - ▷ 浏览信息
 - ▷ 生成事件
 - ▷ 自定义生成步骤
 - ▷ 代码分析

▲ 常规	
输出目录	..\..\..\DII
中间目录	\$(Configuration)\
目标文件名	GrabImage_2012
目标文件扩展名	.exe
清除时要删除的扩展名	*.cdf;*.cache;*.obj;*.ilk;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;*
生成日志文件	\$(IntDir)\\$(MSBuildProjectName).log
平台工具集	Visual Studio 2012 (v110)
启用托管增量生成	否
▲ 项目默认值	
配置类型	应用程序(.exe)
MFC 的使用	使用标准 Windows 库
ATL 的使用	不使用 ATL
字符集	使用 Unicode 字符集
公共语言运行时支持	无公共语言运行时支持
全程序优化	使用链接时间代码生成
Windows 应用商店应用程序支持	否

输出目录
指定输出文件目录的相对路径；可以包含环境变量。

确定 取消 应用(A)

二. 工业相机 SDK 基本接口的使用

1. 枚举设备

可通过函数 `MV_CC_EnumDevices(IN unsigned int nLayerType, IN OUT MV_CC_DEVICE_INFO_LIST* pstDevList)` 来枚举相机。

`nLayerType`: 用户输入的传输层类型 (也就是相机类型), 一般有 `MV_GIGE_DEVICE`, `MV_USB_DEVICE` 分别对应 GigE 和 U3V 相机。 `pstDevList`: 枚举到的设备都存储到这个结构体中了, 供之后使用。

2. 创建句柄

可通过函数 `MV_CC_CreateHandle(OUT void ** handle, IN const MV_CC_DEVICE_INFO* pstDevInfo)` 创建句柄。

`handle`: 创建句柄成功后, 该句柄返回到 `handle`。 `pstDevInfo`: 用户输入的设备信息, 枚举设备时所获取, 这样的话该设备就和该句柄绑定在一起了, 以后只用句柄就完成所有任务。

3. 打开设备

可通过函数 `MV_CC_OpenDevice(IN void* handle, IN unsigned int nAccessMode = MV_ACCESS_Exclusive, IN unsigned short nSwitchoverKey = 0)` 打开设备。

这个函数只需要输入一个参数即可, 就是上面创建成功的句柄 `handle`, 后两个参数一般使用默认参数, 返回成功后表示打开了对应相机。

4. 开启抓图

可通过函数 `MV_CC_StartGrabbing(IN void* handle)` 开始抓图。

此操作依然只输入一个 `handle` 即可, 但开启抓图并没有图像, 只是有流数据传输而已。若需要取图有两种方式, 一种注册回调, 另一种主动调用 `MV_CC_GetOneFrame` 来取图。

5. 获取一帧并保存成图像。

可通过函数 `MV_CC_GetOneFrame(IN void* handle, IN OUT unsigned char * pData, IN unsigned int nDataSize, IN OUT MV_FRAME_OUT_INFO* pFrameInfo)` 获取一帧。所获取的帧属于裸数据, 数据保存在 `pData`, 并无图像格式。 `pFrameInfo` 表示输出帧的信息。

若需要保存图像, 需调用 `MV_CC_SaveImage(IN OUT MV_SAVE_IMAGE_PARAM* pSaveParam)`, 调用前需填写 `MV_SAVE_IMAGE_PARAM` 结构体。

6. 停止抓图

可通过函数 `MV_CC_StopGrabbing(IN void* handle)` 来停止抓图。

只输入一个 `handle` 即可成功停止抓图, 便没有数据流动了。

7. 关闭设备

可通过函数 `MV_CC_CloseDevice(IN void* handle)` 来关闭设备。

只输入一个 handle 即可成功关闭设备。

8. 销毁句柄

可通过函数 `MV_CC_DestroyHandle(IN void * handle)` 来销毁句柄。

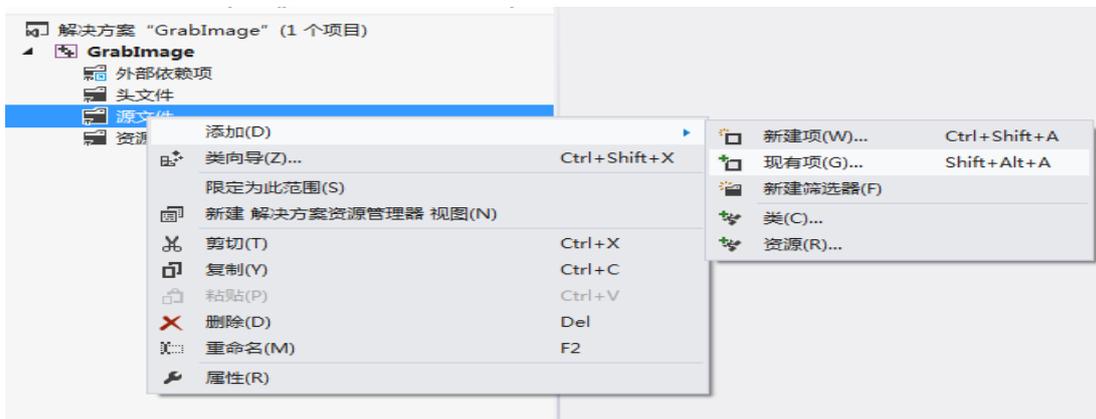
只输入一个 handle 即可销毁句柄。

三. 工业相机 GrabImage C 接口的加载和使用

1. 如何加载 GrabImage 的 C 接口。

首先，将相关的 .cpp 拷贝到相应工程目录下。

然后，将 GrabImage.cpp 加载到工程程序中：



2. 如何使用 GrabImage 的 C 接口。

在需要调用的文件中包含头文件即可

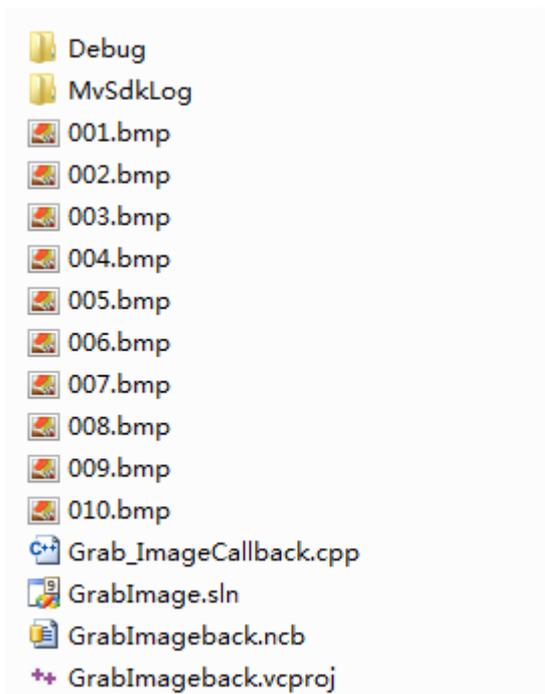
```
#include "MvCameraControl.h"
```

四. 采集保存图像 Demo 的使用

本 Demo 实现单相机的采集并保存图像功能。主要采用连续采集，并用主动方式去获取并保存图像。运行程序是操作界面如下：

当选取好相机后开始采集并保存图像。

```
nCurrentIp: a41824d
nCurrentSubNetMask: 4294967040
nDefaultGateWay: 172052990
chManufacturerName: Hikvision
chModelName: MV-CA013-30GC
chDeviceVersion: V1.8.0 160711
chManufacturerSpecificInfo: Hikvision
chSerialNumber: 00543500828
chUserDefinedName: test1
nNetExport: a418205
*****
2
mkdir MvSdkLog fail, this folder may exist!
hupftest: Width[1280],Height[960],FrameNum[1]
hupftest: Width[1280],Height[960],FrameNum[2]
hupftest: Width[1280],Height[960],FrameNum[3]
hupftest: Width[1280],Height[960],FrameNum[4]
hupftest: Width[1280],Height[960],FrameNum[5]
hupftest: Width[1280],Height[960],FrameNum[6]
hupftest: Width[1280],Height[960],FrameNum[7]
hupftest: Width[1280],Height[960],FrameNum[8]
hupftest: Width[1280],Height[960],FrameNum[9]
hupftest: Width[1280],Height[960],FrameNum[10]
请按任意键继续. . .
```



取到 10 张图后程序结束，对应目录会有所保存的图像