

工业相机 Windows SDK

HALCON 插件使用及示例说明

在法律允许的最大范围内，本手册以及所描述的产品（包含其硬件、软件、固件等）均“按照现状”提供，可能存在瑕疵或错误。本公司不提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的等保证；亦不对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害进行赔偿，包括但不限于商业利润损失、系统故障、数据或文档丢失产生的损失。

目 录

第 1 章 HALCON 插件使用	1
1.1 环境配置	1
1.2 相机配置	1
1.2.1 驱动选择配置	2
1.2.2 设备选择配置	2
1.2.3 设备操作	3
1.2.4 参数配置	3
第 2 章 HALCON 插件接口	4
2.1 运行环境	4
2.2 功能与限制	4
2.3 接口介绍	4
2.3.1 info_framegrabber 参数	4
2.3.2 open_framegrabber 参数	6
2.3.3 set_framegrabber_param 参数	8
2.3.4 get_framegrabber_param 参数	8
2.3.5 set_framegrabber_lut	10
2.3.6 get_framegrabber_lut	10
2.3.7 set_framegrabber_callback	10
2.3.8 get_framegrabber_callback	11
2.3.9 grab_image_start	11
2.3.10 grab_image	11
2.3.11 grab_image_async	11
2.3.12 grab_data	11
2.3.13 grab_data_async	11
2.3.14 close_framegrabber	11

第3章 HALCON 示例介绍	12
3.2 使用步骤	12
3.2.1 软件	12
3.2.2 控制台	13
3.3 开发步骤	14
3.3.1 HalconGrabImage	14
3.3.2 Raw2Himage_C 和 Raw2Himage_Csharp	16
3.3.3 Raw_2_3DFile_C 和 Raw_2_3DFile_Csharp	20

第1章 HALCON 插件使用

1.1 环境配置

在您使用 HALCON 插件前，您需配置环境。

操作步骤

- 根据当前的 HALCON 版本，安装完客户端后，在其安装目录下的 Development\ThirdPartyPlatformAdapter 路径中找到 hAcqMVision.dll 文件，将其拷贝到 HALCON 安装目录下的 bin 文件夹中，如下图所示。

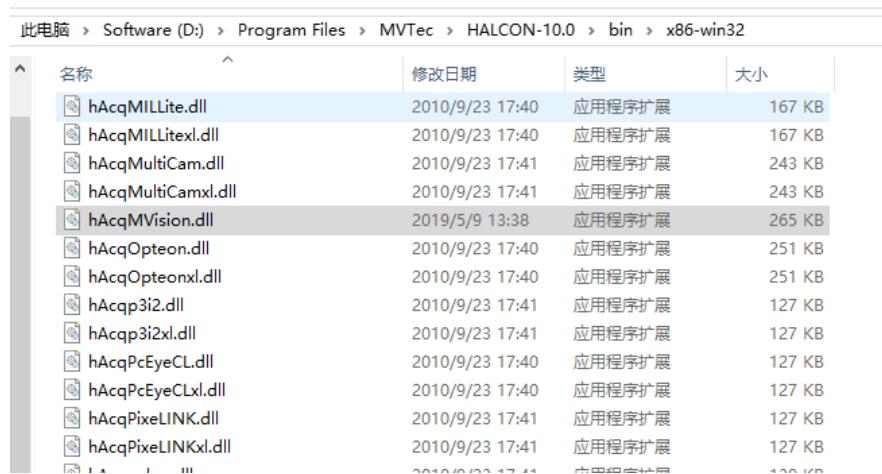


图1-1 拷贝动态库

① 说明

- 若您使用的是 64 位版本程序，您需将 dll 文件复制到 64 位对应目录下。
 - 若您使用的是 HALCON XL 程序，您需将 hAcqMVisionxl.dll 文件复制到 32 或 64 位的对应目录下。
 - 必要时，您可重命名 dll 文件，但一定要保留“hAcq”的前缀，也就是说，重命名后的文件名可为“hAcq + XXX.dll”的形式。例如重命名为 hAcqImageAcquisition.dll 时，在“图像获取接口”的下拉框中选择名字为 ImageAcquisition 的驱动。
- 打开客户端，若 PC 与相机不在同一网段，则配置 IP，配置方式请参考客户端用户手册。
 - 设置相机参数，确保相机能在客户端软件上正常取流。

1.2 相机配置

通过 HALCON 插件，您可简单对相机进行配置。

1.2.1 驱动选择配置

操作步骤

1. 打开 HALCON 菜单栏的助手，并打开新的 Image Acquisition。
2. 在弹出的 Image Acquisition 对话框中，选中“图像获取接口”，在下拉框中选择“MVision”，如下图所示。



图1-2 驱动选择配置

1.2.2 设备选择配置

操作步骤

1. 在 Image Acquisition 对话框中切换到“连接”的界面。
2. 在设备列表中选择想要连接的设备，点击“连接”即可，如下图所示。

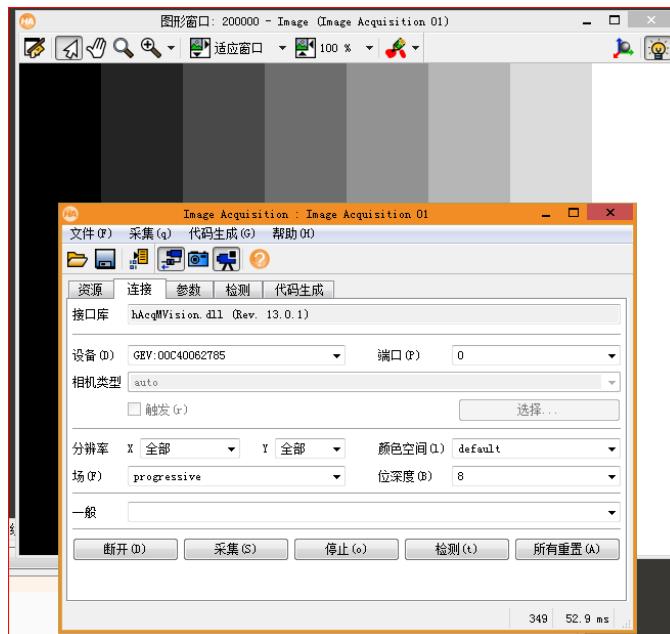


图1-3 设备选择配置

1.2.3 设备操作

连接设备后，您可让相机取流并查看实时画面。

单击 Image Acquisition 对话框的“实时”按钮，您可播放实时画面。此时，您可从“图形窗口”看到画面；单击“停止”可停止实时播放。

如果没有弹出“图形窗口”的对话框，您可在菜单栏的“可视化>打开图形窗口”中打开图像显示对话框。

1.2.4 参数配置

在 Image Acquisition 对话框中可以切换到“参数”界面，在此界面可以对相机的常用属性参数进行设置，如下图所示。

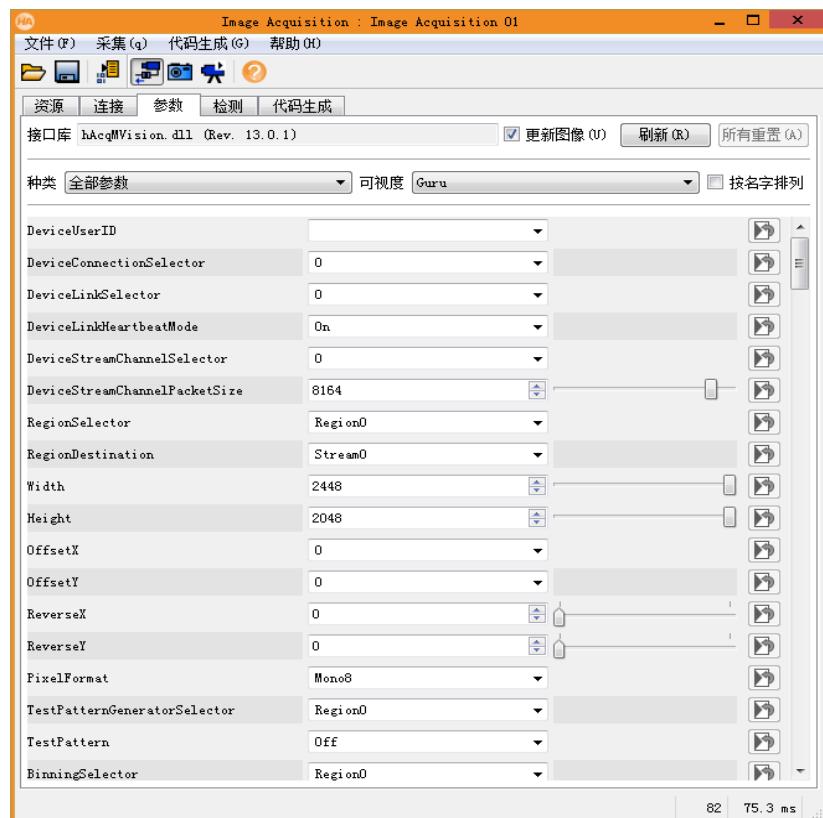


图1-4 参数配置

说明

若该列表中缺少部分参数设置，您可检查用户等级是否为大师等级。在调整为大师等级后，刷新参数列表。

第2章 HALCON 插件接口

2.1 运行环境

在运行 HALCON 插件接口前，您需确保满足以下要求：

1. 当前操作系统为 Windows 7 (32/64)、Windows 10 (32/64)、Windows 11、Windows Server 2008 R2 及以上。
2. 已安装工业相机 SDK 版本 3.2.0 及以上版本。
3. 已安装相应设备的驱动，保证设备能正常取流。
4. 已设置防火墙允许连接相机，否则可能导致相机无法获取到图像。

2.2 功能与限制

HALCON 插件接口可实现如下功能：

- 多相机取流
- 相机属性设置
- 同步和异步取流
- 用户外部注册事件回调和异常回调（仅 SDK4.5.0 及以上版本支持）

说明

HALCON 插件接口不支持 Camera Link 协议、CoaXPress 协议，XoFLink 协议。

2.3 接口介绍

2.3.1 info_framegrabber 参数

表2-1 info_framegrabber 参数说明

PARAMETER	VALUE LIST	TYPE	KIND	DESCRIPTION
'bits_per_channel'	8	integer	pre-defined	Values for bits per channel.
'camera_type'				Unused

PARAMETER	VALUE LIST	TYPE	KIND	DESCRIPTION
'color_space'	['gray', 'rgb']	string	pre-defined	Values for color space.
"defaults"	[1, 1, 0, 0, 0, 0, 'progressive', 8, 'default', -1, 'false', 'auto', '0', 0, 0]	mixed	pre-defined	Default values for open_framegrabber.
'device'	[<unique_device_id>]	string	dynamic	Device names of the available devices.
'external_trigger'	['false', 'true']	string	pre-defined	Unsupported query.
'field'				Unused
'general'		string		Information about the this interface.
'generic'				Unsupported query.
'horizontal_resolution'	1	integer	pre-defined	Value list for horizontal resolution.
'image_height'				Unsupported query.
'image_width'				Unsupported query.
'info_boards'				Unused
'parameters'	[<parameters>]	string	pre-defined	Pre-defined parameters of the HALCON interface.
'parameters_READONLY'			pre-defined	Pre-defined read-only parameters of the HALCON interface.

PARAMETER	VALUE LIST	TYPE	KIND	DESCRIPTION
'parameters_writeonly'			pre-defined	Pre-defined write-only parameters of the HALCON interface.
'port'				Unused
'revision'	'<revision>'	string	pre-defined	Revision number of the this interface.
'start_column'				Unsupported query.
'start_row'				Unsupported query.
'vertical_resolution'	1	integer	pre-defined	Value list for horizontal resolution.

2.3.2 open_framegrabber 参数

表2-2 open_framegrabber 参数说明

PARAMETER	VALUES	DEFAULT	TYPE	DESCRIPTION
Name	'MVision'		string	Name of the HALCON interface.
HorizontalResolution	1	1	integer	The desired horizontal image resolution.
VerticalResolution	1	1	integer	The desired vertical image resolution.
ImageWidth	0	0	integer	'0' stands for the complete image
ImageHeight	0	0	integer	'0' stands for the complete image

PARAMETER	VALUES	DEFAULT	TYPE	DESCRIPTION
StartRow	0	0	integer	Row coordinate of the upper left pixel within the desired image part
StartColumn	0	0	integer	Column coordinate of the upper left pixel within the desired image part.
Field				Ignored
BitsPerChannel	8	8	integer	Number of bits per channel of the resulting HALCON image.
ColorSpace	'default', 'gray', 'rgb'	'default"	string	Desired color space and thus the number of image channels of the resulting HALCON image.
Generic				Ignored
ExternalTrigger				Ignored
CameraType				Ignored
Device	'GEV: <serial_number><device_us er_id>' or 'U3V: <serial_number><device_us er_id>'	0	string	To open a specific camera the device name as shown in info_framegrabber(.., 'device', ..) has to be set.
Port				Ignored
Lineln				Ignored

2.3.3 set_framegrabber_param 参数

您可调用此接口获取相机属性节点的值。调用时，您需要输入属性节点名称，例如 set_framegrabber_param(.., 'Gain', ..)。此外，您还可以设置以下参数：

说明

您可在客户端中查询属性节点的名称。

表2-3 set_framegrabber_param 参数说明

PARAMETER	VALUES	DEFAULT	TYPE	DESCRIPTION
'do_abort_grab'				Cancel current grab
'do_flush_buffers'				Clean image buffer
'grab_timeout'	<milliseconds >	1000	integer	Specify the desired timeout (milliseconds) for aborting a pending grab
'num_buffers'		2	integer	number of buffers to be used
'rotate'	['disable', '90', '180', '270']	'disable'	string	Rotate image
'start_async_after_grab_async'	['enable', 'disable']	'enable'	string	flag for grab_image_async

2.3.4 get_framegrabber_param 参数

包含以下后缀的只读参数：

- '_access': 以字符串形式返回参数的可访问权限，可返回 'ro' (read-only), 'wo' (write-only)，和 'rw' (read/write)。
- '_category': 以字符串形式返回相应参数的类别。
- '_description': 以字符串形式返回相应参数的描述。
- '_range': 以整型或者浮点型返回数组，包含最大值，最小值，步进值，默认值，例如 get_framegrabber_param(.., 'Gain_range', ..) 返回[min, max, step, default]。
- '_values': 返回一组有效值数据，例如，get_framegrabber_param(.., 'volatile_values', ..) 会返回['enable', 'disable']。
- '_visibility': 以字符串形式返回相应参数的访问等级，返回的值有'beginner', 'expert'，和

'guru'。

说明

当您调用 `info_framegrabber (.., 'parameters', ..)` 时，上述所有这些后缀参数名称都不会返回，它们用于通过图形用户界面（HDevelop Image Acquisition Assistant）实现轻松的参数化。

除了上述参数以外，您还可通过该接口获取如下参数：

表2-4 `get_framegrabber_param` 参数说明

PARAMETER	VALUES	DEFAULT	TYPE	KIND	DESCRIPTION
'available_callback_types'	[<callback_type>]		string	dynamic	Query all callback types which are supported by this interface. Refer to Chapter2.3.7
'available_parameter_names'	[<parameters>]		string	dynamic	The list contains the names of all available parameters.
'grab_timeout'	<millisecond>	1000	integer	pre-defined	Current grab timeout in milliseconds.
'num_buffers'	<number>	2	integer	pre-defined	Number of buffers used for the image acquisition.
'revision'	'<revision>'		string	pre-defined	Revision number of the Hikrobot interface.
'start_async_after_grab_async'	'disable', 'enable'	'enable'	string	pre-defined	Status of 'start_async_after_grab_async'.
'rotate'	['disable', '90', '180', '270']	'disable'	string	dynamic	Rotate of the HALCON image

2.3.5 set_framegrabber_lut

不支持此接口。

2.3.6 get_framegrabber_lut

不支持此接口。

2.3.7 set_framegrabber_callback

该接口支持设置相机特定的事件，通常和 get_framegrabber_callback 接口配套使用，用于获取和设置相机特定的事件。

您可在调用 get_framegrabber_param 接口时传参'available_callback_types'，查询到当前支持的事件列表。若您注册了回调，且设备触发了相关事件，则程序将调用指定的回调函数。如果回调函数设置为 NULL，则相应的回调将被注销。



注意

如需开启或者停用特定的回调，您也必须开启或关闭相应的设备事件。可通过设置参数“EventSelector”为特定事件，将“EventNotification”参数设置为 0 或者 1 来实现。

回调函数定义：

```
Herror (__stdcall *HAcqCallback)(void *AcqHandle, void *Context, void *UserContext)
```

- AcqHandle：图像采集实例的句柄。
- Context：这个参数未使用，插件内部置为 NULL。
- UserContext：用户自定义内容。



注意

回调函数中不适合处理耗时较长的功能，否则会导致程序阻塞。

该接口支持注册的事件包含：

表2-5 set_framegrabber_callback 支持的事件说明

TYPE	DESCRIPTION
'Exception'	设备异常掉线
'ExposureEnd'	曝光结束
'ExposureStart'	曝光开始

TYPE	DESCRIPTION
'FrameEnd'	图像传输结束

2.3.8 get_framegrabber_callback

您可通过调用此接口获取注册的回调函数的指针和用户自定义内容，详细请参考上述 set_framegrabber_callback 接口。

2.3.9 grab_image_start

该接口用于开始异步抓取，详细使用说明，请参见 HALCON Operator Reference 中采集控制和抓取操作的部分。

2.3.10 grab_image

此接口用于抓取一帧图像，该接口将获取的图像转换为由参数“bits_per_channel”和“color_space”指定的所需图像格式。

2.3.11 grab_image_async

异步抓取一帧图像，即接口返回一帧图像后，开始下一帧图像的抓取，该接口将获取的图像转换为由参数“bits_per_channel”和“color_space”指定的所需图像格式。

2.3.12 grab_data

不支持此接口。

2.3.13 grab_data_async

不支持此接口。

2.3.14 close_framegrabber

关闭设备。

第3章 HALCON 示例介绍

本节主要展示使用 HALCON 接口进行开发程序的方法及过程。在 SDK 开发包目录下，提供 5 个示例程序，请见下表。这些示例程序分别从不同角度展示利用 halcondontnet 和 MvCameraControl.Net 进行开发的方法。

表3-1 Halcon 示例程序

示例程序	开发语言	程序界面	功能说明
HalconGrabImage	C++	MFC 界面	基本示例程序，包含 HALCON 常用接口调用
Raw2Himage_C	C++	MFC 界面	使用 HALCON 接口进行图像像素转换以及显示示例程序
Raw2Himage_Csharp	C#	MFC 界面	
Raw_2_3DFile_C	C++	控制台界面	使用 HALCON 接口进行 3D 图像转换示例程序
Raw_2_3DFile_CSharp	C#	控制台界面	

i 说明

- Raw2Himage_C 和 Raw2Himage_CSharp、Raw_2_3DFile_C 和 Raw_2_3DFile_CSharp 功能完全相同，只是开发的语言不同。
- C++与 C#版示例程序均兼容中英文，对关键的程序均有中英文的注释，界面控件均有中英文区分，可通过切换属性的 language 实现。
- 对于 C++示例程序，因 HALCON 版本差异，单个 Demo 无法适配所有的 HALCON 版本，故在工程名中备注适用的 HALCON 版本，如 HalconGrabImage_10 表示该示例用于 HALCON10，HalconGrabImage_11-13 表示该示例用于 HALCON11-13 版本。

3.2 使用步骤

3.2.1 软件

软件界面如下图所示，不同编译环境和示例程序下的软件界面略有差异，但功能基本一致，各区域功能介绍请见下表。

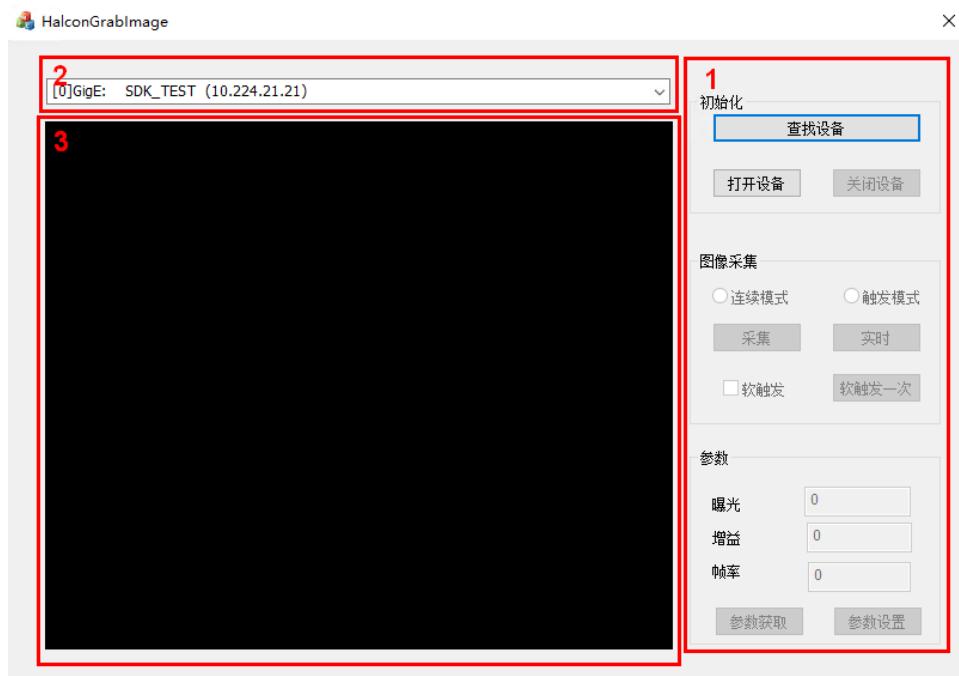


图3-1 示例软件

表3-2 示例软件功能说明

编号	名称	功能说明
1	控制模块	可进行初始化、图像采集、参数设置等操作。
2	下拉设备列表	下拉可显示设备进行连接。
3	图像显示区域	可显示已采集到的图像。

① 说明

- 您可参照 VC 示例 BasicDemo，操作此 HALCON 示例程序。
- 在使用 HALCON 软件前，您需复制工业相机 SDK HALCON 插件至 HALCON 软件对应的路径下，详细步骤请参见第 1 章 1.1 环境配置。

3.2.2 控制台

控制台界面如下图所示，具有查找设备、打开设备、关闭设备、开始采集、停止采集、存取 Halcon3D 图像等功能。

```
d:\SVN\Release\MVS\Samples\Halcon\VC\Raw_2_3DFile_C\x64\Debug\Raw_2_3DFile_C_11-13.exe
UserDefinedName: !@#0$#
[device 20]:
CurrentIp: 10.64.52.4
UserDefinedName: 600

[device 21]:
CurrentIp: 10.64.52.165
UserDefinedName: DSB123

[device 22]:
CurrentIp: 10.64.52.34
UserDefinedName: SmartCamera

[device 23]:
CurrentIp: 10.64.52.142
UserDefinedName: SmartCamera

[device 24]:
CurrentIp: 10.64.52.218
UserDefinedName: 2100Sub

Please Input camera index:10
Get One Frame: Width[1536], Height[1], FrameNum[1]
*****
* .ply; 1.obj; *
*****
Select FileType: 0
Press a key to exit.
```

图3-2 控制台示例

操作步骤

1. 打开控制台界面后，HALCON 自动枚举在线设备；
2. 枚举完成后，输入对应枚举 3D 相机设备的下标，调用 SDK 中的取流函数，获取图像数据；
3. 根据提示，可选择将采集到的图像保存为 ply 或 obj 格式。

3.3 开发步骤

3.3.1 HalconGrabImage

HalconGrabImage 是一个基本示例程序，包含 HALCON 常用的接口调用，初次使用 HALCON 接口进行二次开发推荐首先参考 HalconGrabImage。

前提条件

- 已安装工业相机客户端软件。
- 已在设备环境变量中自动写入 32 位和 64 位的 dll 文件。
- 已配置工业相机 SDK HALCON 插件。

操作步骤

1. 创建 C++工程并添加引用，如下图所示。

工业相机 Windows SDK • HALCON 插件使用及示例说明

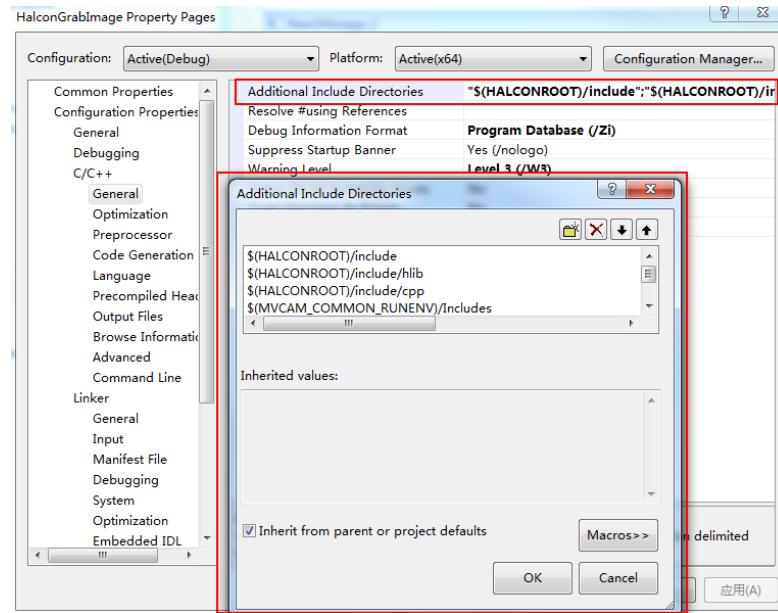


图3-3 创建工程

2. 将 HALCON 和 C++ 的 SDK 头文件和 lib 文件加入到工程中 , 如图 3-5 和图 3-4 所示。

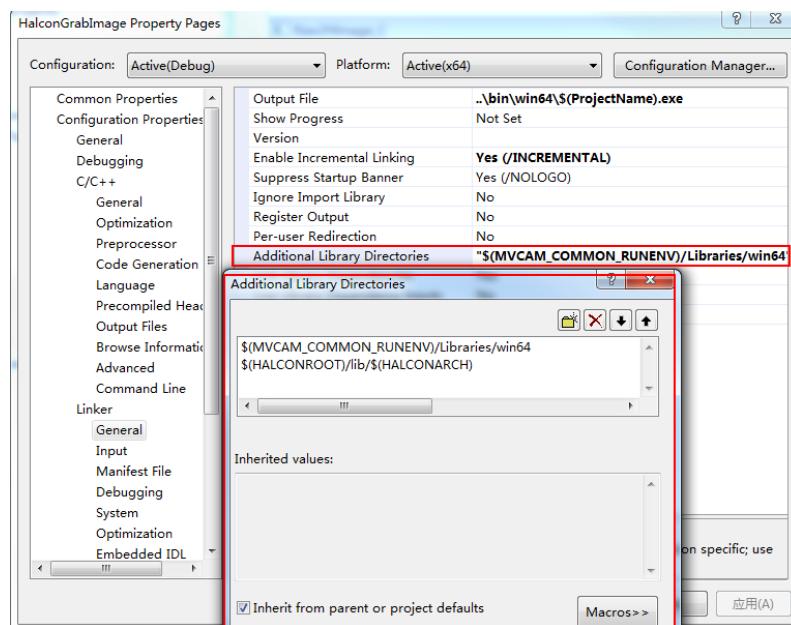


图3-4 加入 lib 库

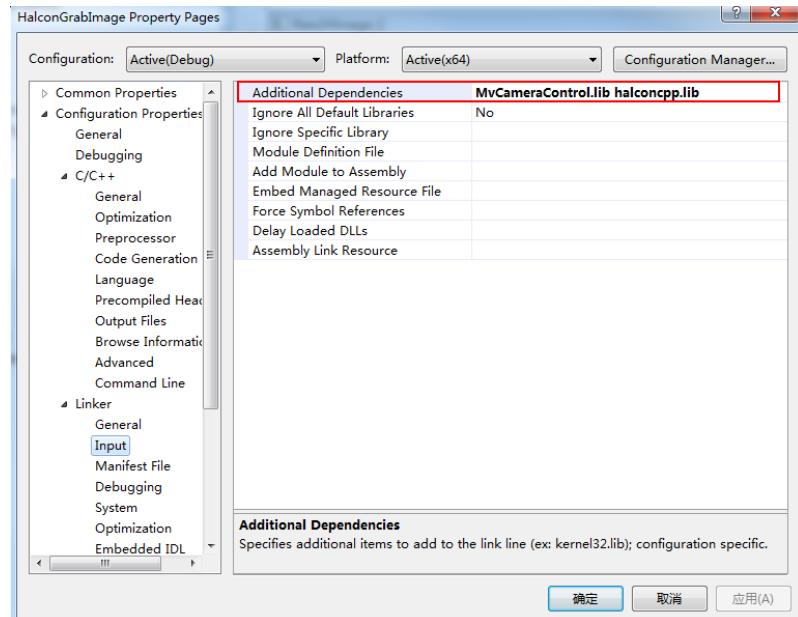


图3-5 加入 lib 文件

3. 添加引用后在工程中引用 MvCameraControl.h 和 HalconCpp.h，调用 HALCON 和 SDK 中相机操作的函数，如下图所示。

```
1 //·HalconGrabImageDlg.h··头文件
2 //·
3 //
4 //
5 #pragma once
6 #include "afxwin.h"
7 #include "MvCameraControl.h"
8 #include "HalconCpp.h"
9
10 using namespace Halcon;
11
12 /*函数返回码定义*/
13 typedef int Status;
14 #define STATUS_OK.....0
15 #define STATUS_ERROR.....-1
16
17 //·HalconGrabImageDlg·对话框
18 class CHalconGrabImageDlg : public CDialog
19 {
20     //·构造
21     public:
22     ~CHalconGrabImageDlg (CWnd* pParent = NULL); //·标准构造函数
23
24     //·对话框数据
25     enum { IDD = IDD_HALCONGRABIMAGE_DIALOG };
26
27     protected:
28     virtual void DoDataExchange (CDataExchange* pDX); //·DDX/DDV·支持
29
30     //·实现
31     protected:
32     HICON m_hIcon;
33
34     //·生成的消息映射函数
35     virtual BOOL OnInitDialog ();
36     afx_msg void OnSysCommand (UINT nID, LPARAM lParam);
37 }
```

图3-6 引用头文件

3.3.2 Raw2Himage_C 和 Raw2Himage_Csharp

Raw2Himage_C 和 Raw2Himage_Csharp 展示如何使用 HALCON 接口进行图像像素转换以及显示。

前提条件

- 已安装工业相机客户端软件。
- 已在设备环境变量中自动写入 32 位和 64 位的 dll 文件。
- 已配置工业相机 SDK HALCON 插件。

操作步骤

1. 创建 C++工程并添加引用，如下图所示。

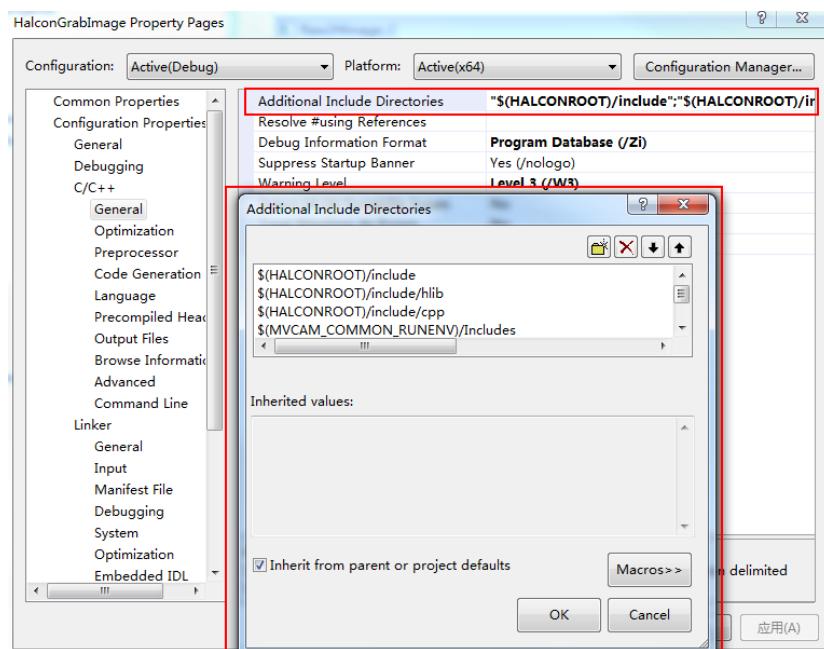


图3-7 创建工程

2. 将 HALCON 和 C++ 的 SDK 头文件和 lib 文件加入到工程中，如图 3-8 和图 3-9 所示。

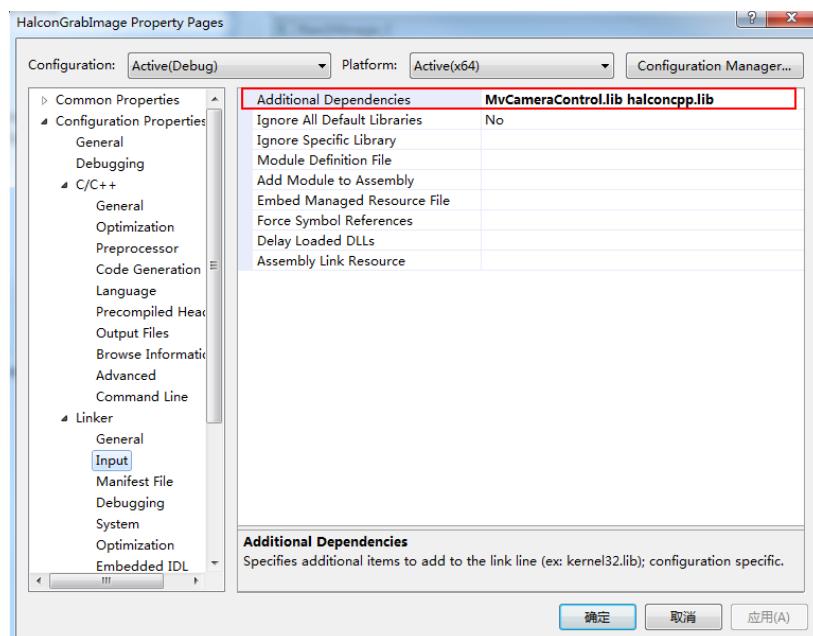


图3-8 加入 lib 文件

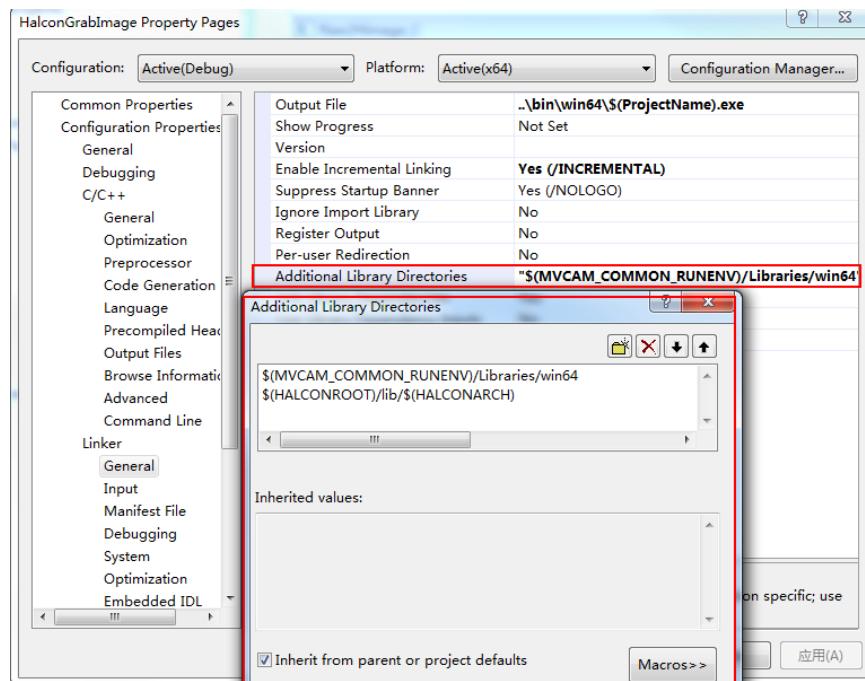


图3-9 加入头文件

① 说明

Raw2Himage_Csharp 中需创建 CS 工程并添加引用，将 halcondotnet.dll 和 MvCameraControl.Net.dll 文件加入到工程中，如下图所示。

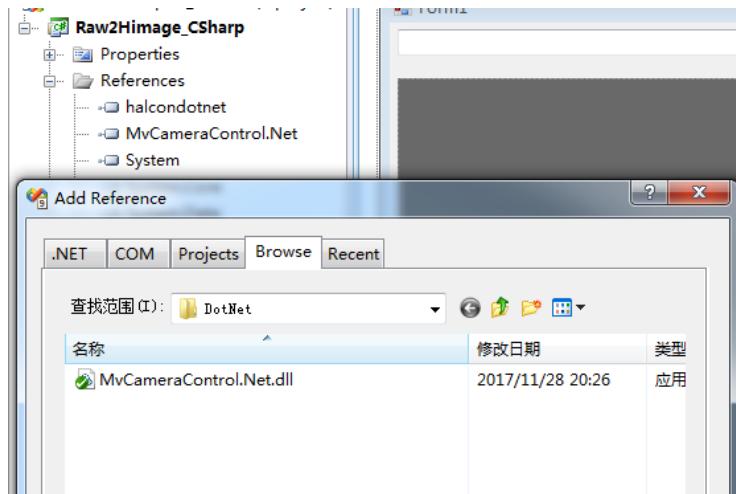


图3-10 Raw2Himage_Csharp 中加入 dll 文件

3. 添加引用后在工程中引用 MvCameraControl.h 和 HalconCpp.h，调用 HALCON 和 SDK 中相机操作的函数，如下图所示。

```
1 //·HalconGrabImageDlg.h··头文件
2 ///////////////////////////////////////////////////////////////////
3 ///////////////////////////////////////////////////////////////////
4 ///////////////////////////////////////////////////////////////////
5 #pragma once
6 #include "afxwin.h"
7 #include "MvCameraControl.h"
8 #include "HalconCpp.h".
9
10 using namespace Halcon;
11
12 /*函数返回码定义*/
13 typedef int Status;
14 #define STATUS_OK.....0
15 #define STATUS_ERROR.....-1
16
17 //·HalconGrabImageDlg·对话框
18 class CHalconGrabImageDlg : public CDialog
19 {
20 //·构造
21 public:
22     CHalconGrabImageDlg(CWnd* pParent = NULL); //·标准构造函数
23
24 //·对话框数据
25 enum { IDD = IDD_HALCONGRABIMAGE_DIALOG };
26
27 protected:
28     virtual void DoDataExchange(CDataExchange* pDX); //·DDX/DDV·支持
29
30
31 //·实现
32 protected:
33     HICON m_hIcon;
34
35     //·生成的消息映射函数
36     virtual BOOL OnInitDialog();
37     afx_msg void OnSvsCommand(UINT nID, LPARAM lParam);
```

图3-11 引用头文件

说明

Raw2Himage_Csharp 中需引用命名空间 using MvCamCtrl.NET 和 using HalconDotNet，如图 3-12 和图 3-13 所示。

```
public static object ByteToStruct(byte[] bytes, Type type);
public IntPtr GetCameraHandle();
public int MV_CC_CloseDevice_NET();
public int MV_CC_ConvertPixelType_NET(ref MyCamera.MV_PIXEL_CONVERT_PARAM pstCvtParam);
public int MV_CC_CreateDevice_NET(ref MyCamera.MV_CC_DEVICE_INFO stDeviceInfo);
public int MV_CC_CreateDeviceWithoutLog_NET(ref MyCamera.MV_CC_DEVICE_INFO stDeviceInfo);
public int MV_CC_DestroyDevice_NET();
public int MV_CC_Display_NET(IntPtr hWnd);
public static int MV_CC_EnumDevices_NET(uint nTLayerType, ref MyCamera.MV_CC_DEVICE_INFO_LIST stDevList);
public static int MV_CC_EnumDevicesEx_NET(uint nTLayerType, ref MyCamera.MV_CC_DEVICE_INFO_LIST stDevList, string pM);
public static int MV_CC_EnumerateDevices_NET();
public int MV_CC_FeatureLoad_NET(string pfileName);
public int MV_CC_FeatureSave_NET(string pfileName);
public int MV_CC_FileAccessRead_NET(ref MyCamera.MV_CC_FILE_ACCESS pstFileAccess);
public int MV_CC_FileAccessWrite_NET(ref MyCamera.MV_CC_FILE_ACCESS pstFileAccess);
public int MV_CC_GetAcquisitionLineRate_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetAcquisitionMode_NET(ref MyCamera.MVCC_ENUMVALUE pstValue);
public int MV_CC_GetAllMatchInfo_NET(ref MyCamera.MV_ALL_MATCH_INFO pstInfo);
public int MV_CC_GetAOInfoSetX_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetAOInfoSetY_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetAutoExposureTimeLower_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetAutoExposureTimeUpper_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetBalanceRatioBlue_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetBalanceRatioGreen_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetBalanceRatioRed_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetBalanceWhiteAuto_NET(ref MyCamera.MVCC_ENUMVALUE pstValue);
public int MV_CC_GetBoolValue_NET(string strKey, ref bool pbValue);
public int MV_CC_GetBrightness_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetBurstFrameCount_NET(ref MyCamera.MVCC_INTVALUE pstValue);
public int MV_CC_GetDeviceInfo_NET(ref MyCamera.MV_CC_DEVICE_INFO stDeviceInfo);
public int MV_CC_GetDeviceUserID_NET(ref MyCamera.MVCC_STRINGVALUE pstValue);
public int MV_CC_GetEnumValue_NET(string strKey, ref MyCamera.MVCC_ENUMVALUE pstValue);
public int MV_CC_GetExposureAutoMode_NET(ref MyCamera.MVCC_ENUMVALUE pstValue);
public int MV_CC_GetExposureTime_NET(ref MyCamera.MVCC_FLOATVALUE pstValue);
public int MV_CC_GetFloatValue_NET(string strKey, ref MyCamera.MVCC_FLOATVALUE pstValue);
public int MV_CC_GetFrameRate_NET(ref MyCamera.MVCC_FLOATVALUE pstValue);
```

图3-12 引用 using MvCamCtrl.NET 后调用接口

```

namespace HalconDotNet
{
    public class HOperatorSet
    {
        public HOperatorSet();
        ...
        public static void AbsDiffImage(HObject image1, HObject image2, out HObject imageAbsDiff, HTuple mult);
        public static void AbsFunctId(HTuple function, out HTuple functionAbsolute);
        public static void AbsImage(HObject image, out HObject imageAbs);
        public static void AbsInvarFourierCoeff(HTuple realInvar, HTuple imaginaryInvar, HTuple coefP, HTuple coefQ, HTuple matrixID);
        public static void AbsMatrix(HTuple matrixID, out HTuple matrixAbsID);
        public static void AbsMatrixMod(HTuple matrixID);
        public static void AccessChannel(HObject multiChannelImage, out HObject image, HTuple channel);
        public static void ActivateComputeDevice(HTuple deviceHandle);
        public static void AdaptTemplate(HObject image, HTuple templateID);
        public static void AddChannels(HObject regions, HObject image, out HObject grayRegions);
        public static void AddImage(HObject image1, HObject image2, out HObject imageResult, HTuple mult, HTuple add);
        public static void AddMatrix(HTuple matrixAID, HTuple matrixBID, out HTuple matrixSumID);
        public static void AddMatrixMod(HTuple matrixID);
        public static void AddNoiseDistribution(HObject image, out HObject imageNoise, HTuple distribution);
        public static void AddNoiseWhite(HObject image, out HObject imageNoise, HTuple amp);
        public static void AddNoiseWhiteContourId(HObject contours, out HObject noisyContours, HTuple numRegrPoints, HTuple classID, HTuple randomize);
        public static void AddSampleClassGmm(HTuple GMHandle, HTuple features, HTuple classID, HTuple randomize);
        public static void AddSampleClassMp(HTuple MLPHandle, HTuple features, HTuple target);
        public static void AddSampleClassSvm(HTuple SVMHandle, HTuple features, HTuple classVal);
        public static void AddSampleImageClassGmm(HObject image, HObject classRegions, HTuple GMHandle, HTuple randomize);
        public static void AddSampleImageClassMp(HObject image, HObject classRegions, HTuple MLPHandle);
        public static void AddSampleImageClassSvm(HObject image, HObject classRegions, HTuple SVMHandle);
        public static void AdjustMosaicImages(HObject images, out HObject correctedImages, HTuple from, HTuple to, HTuple homMat2D);
        public static void AffineTransContourId(HObject contours, out HObject contoursAffinTrans, HTuple homMat2D);
        public static void AffineTransImage(HObject image, out HObject imageAffinTrans, HTuple homMat2D, HTuple interpOrder);
        public static void AffineTransImageSize(HObject image, out HObject imageAffinTrans, HTuple homMat2D, HTuple interpOrder);
        public static void AffineTransObjectModel3d(HTuple objectModel3dID, HTuple homMat3D, out HTuple objectModel3dIDaf);
        public static void AffineTransPixel(HTuple homMat2D, HTuple row, HTuple col, out HTuple rowTrans, out HTuple colT);
        public static void AffineTransPoint2d(HTuple homMat2D, HTuple px, HTuple py, out HTuple qx, out HTuple qy);
    }
}

```

图3-13 引用 using HalconDotNet 后调用接口

4. 调用 SDK 中的开始取流接口 StartGrabbing，并创建一个线程 WorkThread，在该线程中循环调用 SDK 的获取一帧图像的接口 GetOneFrameTimeout。

i 说明

Raw2Himage_Csharp 中需调用开始取流接口 MV_CC_StartGrabbing_NET，并创建 ReceiveImageWorkThread 线程，在该线程中循环获取一帧图像接口 MV_CC_GetOneFrameTimeout_NET。

5. 获取图像后进行图像格式转换，保存成 Himage 图片格式，并用 HALCON 自带的显示接口 HalconDisplay 进行显示。

3.3.3 Raw_2_3DFile_C 和 Raw_2_3DFile_Csharp

Raw_2_3DFile_C 和 Raw_2_3DFile_CSharp 重点展示如何使用 HALCON 接口进行 3D 图像的转换。

相机基本操作流程与 HalconGrabImage 相似。本节重点介绍如何将相机的 3D 数据转换为 HALCON 中 3D 的数据格式，并存取为文件，存取的文件类型有 ply 和 obj 两种。

操作步骤

1. 调用 SDK 中的开始取流接口 StartGrabbing；
2. 调用取流接口 MV_CC_GetOneFrameTimeout 获取一帧图像；
3. 获取图像后进行图像格式转换，保存为 ply 和 obj 的 3D 数据格式。

