

HALCON插件使用说明

Version 2.0.0

2018.09.17

目录

1 使用 HDevelop 平台连接相机	3
1.1 使用步骤	3
1.1.1 安装相机 SDK	3
1.1.2 拷贝 SDK 的 Halcon 插件库	4
1.1.3 设置相机 IP (如果使用 GigE 相机)	4
1.1.4 使用 HDevelop 操作相机	5
2 使用插件二次开发	7
2.1 连续采集	8
2.2 软触发采集	9
2.3 硬触发采集	11

1 使用 HDevelop 平台连接相机

1.1 使用步骤

1.1.1 安装相机 SDK

文件说明		文件位置
Halcon10 插件	32 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon10\win32
	64 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon10\x64
Halcon11 插件	32 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon11\win32
	64 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon11\x64
Halcon12 插件	32 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon12\win32
	64 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon12\x64
Halcon13 插件	32 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon13\win32
	64 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon13\x64

Halcon17 插件	32 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon17\win32
	64 位	【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon17\x64
插件使用说明文档		【SDK 安装目录】\Development\3rdPartyPlatformAdapter\HalconHDevelopAdapter\halcon 适配层使用手册.pdf

1.1.2 拷贝 SDK 的 Halcon 插件库

以 32 位 Halcon10 为例，说明拷贝方法。



图 1 拷贝 SDK 的 Halcon 插件库

1.1.3 设置相机 IP（如果使用 GigE 相机）

使用相机 SDK 客户端软件，将相机的 IP 地址设置到 PC 网卡所在的网段。

1.1.4 使用 HDevelop 操作相机

- ① 打开 HDevelop，在主菜单中选择“助手” -> “打开新的 Image Acquisition”。

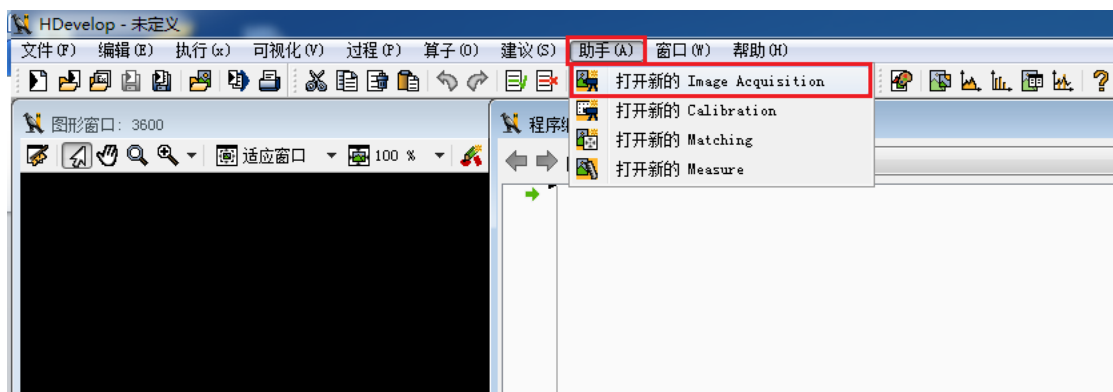


图 2 打开新的 Image Acquisition

- ② 在弹出的“Image Acquisition”对话框中，选中“图像获取接口”，然后在下拉框中选择“HVM3rdParty”。

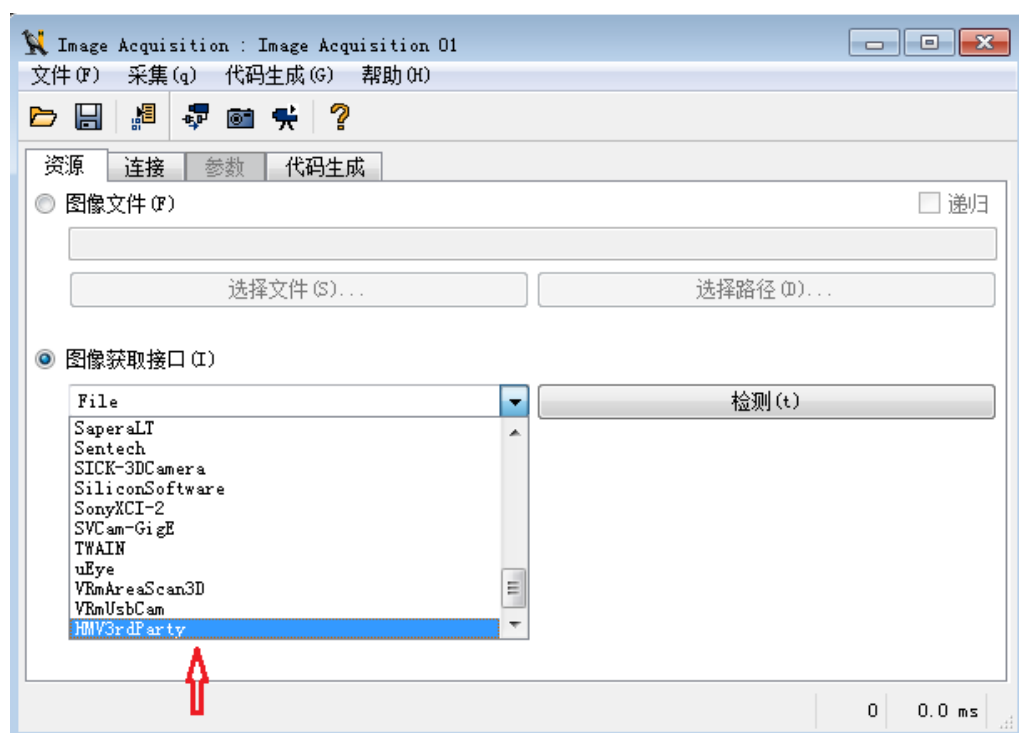


图 3 选择 HVM3rdParty

③ 在 “Image Acquisition” 对话框中切换到 “连接” 标签页，然后在设备列表中选择想要连接的设备，点击 “连接”，就可以将相机连接到软件了。

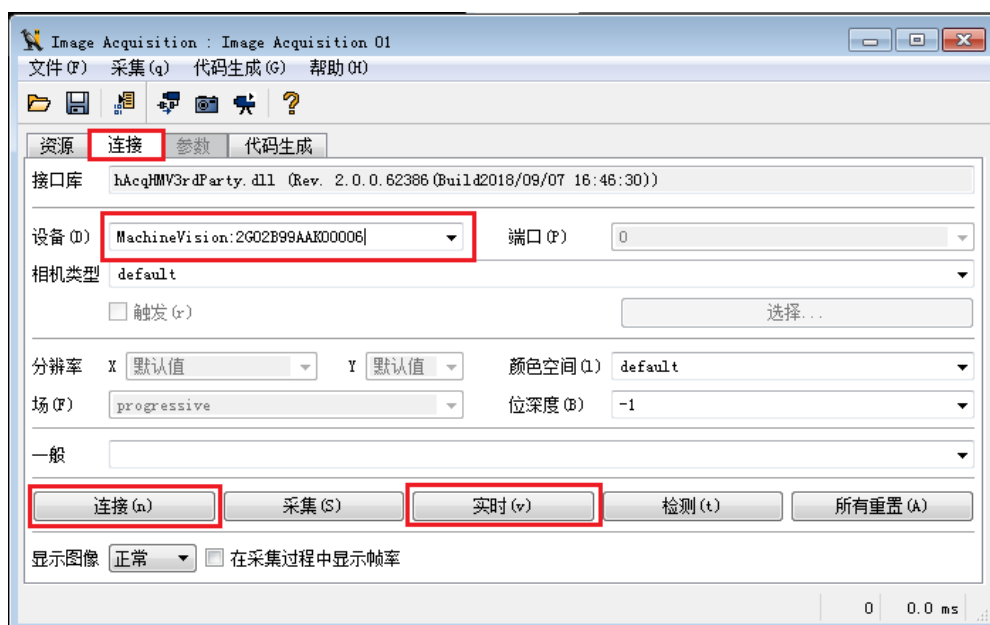


图 4 连接相机和开始拉流

④ 点击 “实时”, 进行实时播放, 可以在 “图形窗口” 对话框中看到实时播放的画面。
点击 “停止” 可以停止实时播放。

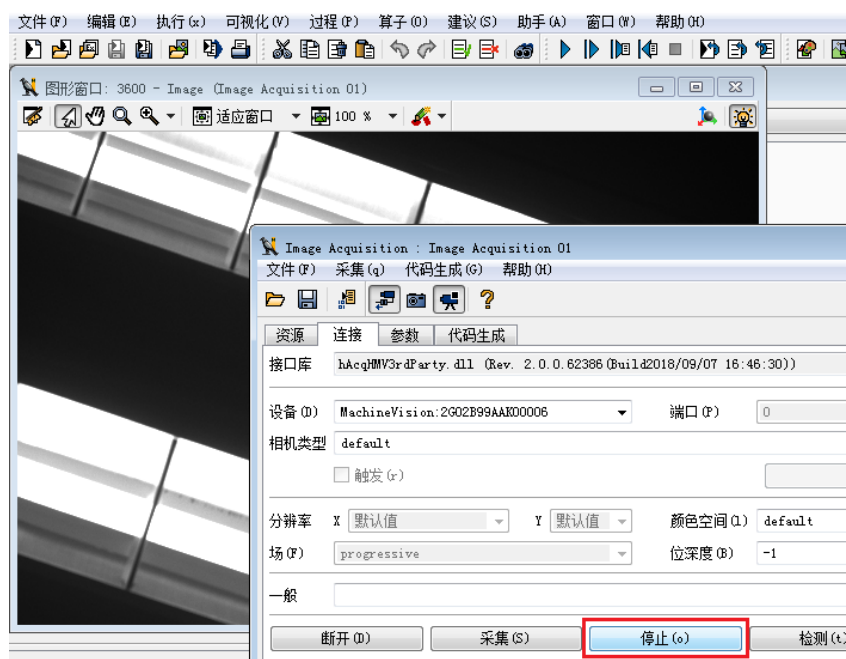


图 5 停止拉流

⑤ 在“Image Acquisition”对话框中切换到“参数”标签页，可以对相机的属性进行设置。

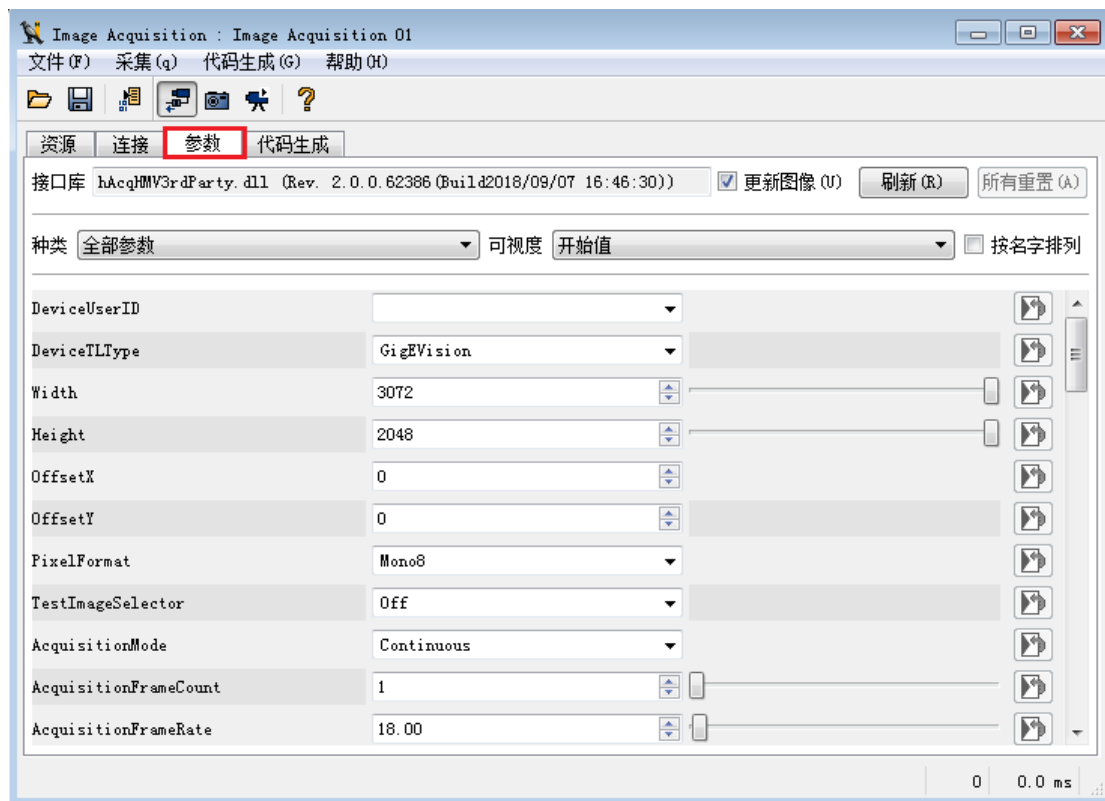


图6 相机属性列表

2 使用插件二次开发

SDK的Halcon插件也支持导出代码程序的用法。

以下，以C++为例，展示如何在Halcon导出代码中实现连续采集、软触发采集、硬触发采集。

2.1 连续采集

```
#include "HalconCpp.h"
using namespace Halcon;
#ifdef NO_EXPORT_MAIN
// Main procedure
void action()
{
    // 获取相机列表

    HTuple          acqname("HVM3rdParty");
    HTuple info;
    HTuple cameras;
    info_framegrabber(acqname, "device", &info, &cameras);

    // 打印相机列表

    int nCnt = 0;
    while (nCnt < cameras.Num())
    {
        printf("nCnt:%d  %s\n", nCnt, cameras[nCnt].S());
        nCnt++;
    }

    // 打开相机列表中的第一台设备

    HTuple  AcqHandle;
    open_framegrabber(acqname, 0, 0, 0, 0, 0, 0, "progressive", -1, "default", -1, "false",
"default", cameras[0], 0, -1, &AcqHandle);

    // 关闭触发

    set_framegrabber_param(AcqHandle, "TriggerMode", "Off");

    // 开始异步取流

    grab_image_start(AcqHandle, -1);
}
```



```

// 异步取流

Hobject Image;
while (0 != 1)
{
    grab_image_async(&Image, AcqHandle, -1);
    // Do something
}

// 关闭设备

close_framegrabber(AcqHandle);
}

#ifdef NO_EXPORT_APP_MAIN
int main(int argc, char *argv[])
{
    using namespace Halcon;
    // Default settings used in HDevelop (can be omitted)
    set_system("do_low_error", "false");
    action();
    return 0;
}
#endif
#endif

```

2.2 软触发采集

```

#include "HalconCpp.h"
using namespace Halcon;
#ifdef NO_EXPORT_MAIN
// Main procedure
void action()
{
    // 获取相机列表

    HTuple          acqname("HMV3rdParty");
    HTuple info;
    HTuple cameras;
    info_framegrabber(acqname, "device", &info, &cameras);
}

```

```

// 打印相机列表

int nCnt = 0;
while (nCnt < cameras.Num())
{
    printf("nCnt:%d  %s\n", nCnt, cameras[nCnt].S());
    nCnt++;
}

// 打开相机列表中的第一台设备

HTuple  AcqHandle;
open_framegrabber(acqname, 0, 0, 0, 0, 0, 0, "progressive", -1, "default", -1, "false",
"default", cameras[0], 0, -1, &AcqHandle);

// 打开触发

set_framegrabber_param(AcqHandle, "TriggerSelector", "FrameStart");
set_framegrabber_param(AcqHandle, "TriggerMode", "On");

// 选择触发源为Software

set_framegrabber_param(AcqHandle, "TriggerSource", "Software");

// 开始异步取流
grab_image_start(AcqHandle, -1);

// 异步取流
Hobject  Image;
while (0 != 1)
{
    // 执行软触发
    set_framegrabber_param(AcqHandle, "TriggerSoftware", "");

    // 取图
    grab_image_async(&Image, AcqHandle, -1);
    // Do something
}

// 关闭设备
close_framegrabber(AcqHandle);
}

```

```

#ifndef NO_EXPORT_APP_MAIN
int main(int argc, char *argv[])
{
    using namespace Halcon;
    // Default settings used in HDevelop (can be omitted)
    set_system("do_low_error", "false");
    action();
    return 0;
}
#endif
#endif

```

2.3 硬触发采集

```

#include "HalconCpp.h"
using namespace Halcon;
#ifndef NO_EXPORT_MAIN
// Main procedure
void action()
{
    // 获取相机列表

    HTuple          acqname("HMV3rdParty");
    HTuple info;
    HTuple cameras;
    info_framegrabber(acqname, "device", &info, &cameras);

    // 打印相机列表

    int nCnt = 0;
    while (nCnt < cameras.Num())
    {
        printf("nCnt:%d  %s\n", nCnt, cameras[nCnt].S());
        nCnt++;
    }

    // 打开相机列表中的第一台设备

    HTuple  AcqHandle;
    open_framegrabber(acqname, 0, 0, 0, 0, 0, 0, "progressive", -1, "default", -1, "false",
"default", cameras[0], 0, -1, &AcqHandle);

```

```

// 打开触发
set_framegrabber_param(AcqHandle, "TriggerSelector", "FrameStart");
set_framegrabber_param(AcqHandle, "TriggerMode", "On");

// 选择触发源为Line1
set_framegrabber_param(AcqHandle, "TriggerSource", "Line1");

// 设置取图超时时间为-1(永久等待)
set_framegrabber_param(AcqHandle, "grab_timeout", -1);

// 开始异步取流
grab_image_start(AcqHandle, -1);

// 异步取流
Hobject Image;
while (0 != 1)
{
    grab_image_async(&Image, AcqHandle, -1);
    //Do something
}

// 关闭设备
close_framegrabber(AcqHandle);
}

#ifdef NO_EXPORT_APP_MAIN
int main(int argc, char *argv[])
{
    using namespace Halcon;
    // Default settings used in HDevelop (can be omitted)
    set_system("do_low_error", "false");
    action();
    return 0;
}
#endif
#endif

```