

OpenCV + DirectShow Device via AVerMedia SDK

Request:

Client wish to acquire video stream and process video via OpenCV library.

The challenge:

OpenCV focuses on video processing but acquiring a Directshow device to work in cooperation will be the challenge!

KEYWORDS: UNABLE TO SELECT USB DEVICE/ MULTIPLE DEVICES/ CAPTURE CARD IN OPENCV

Solution:

AVerMedia SDK (DirectShow) to “Open & Control” devices, OpenCV to “Process” video.

Application example: *AVerMedia SDK Pro(1.0.1.27) OpenCV(opencv-3.2.0-vc14)*

To download project sample code, please click on [OneDrive](#).

```
// ConsoleApplication3.cpp : Defines the entry point for the console application.
// Create by AVerMedia Jay/Bruce
// Demo how to use AVerMedia SDK + OpenCV
// Contact jay.liu@avermedia.com or TechSupport@avermedia.com for further support
// or check OpenCV website http://opencv.org/
// Environment : AVerMedia SDK Pro(1.0.1.27) OpenCV(opencv-3.2.0-vc14)
// Virtual Studio 2015

#include"stdafx.h"
#include"atlstr.h"
#include<iostream>
#include"AVerCapAPI_Pro.h"
#include"opencv2/core/core.hpp"
#include"opencv2/highgui/highgui.hpp"
#include<windows.h>
usingnamespace std;
usingnamespace cv;

voidErrorMsg(DWORDErrorCode)
```

```
{
    printf("ErrorCode = %d\n", ErrorCode);
    if (ErrorCode == CAP_EC_SUCCESS)
    {
        printf("CAP_EC_SUCCESS\n");
    }
    if (ErrorCode == CAP_EC_INIT_DEVICE_FAILED)
    {
        printf("CAP_EC_INIT_DEVICE_FAILED\n");
    }
    if (ErrorCode == CAP_EC_DEVICE_IN_USE)
    {
        printf("CAP_EC_DEVICE_IN_USE\n");
    }
    if (ErrorCode == CAP_EC_NOT_SUPPORTED)
    {
        printf("CAP_EC_NOT_SUPPORTED\n");
    }
    if (ErrorCode == CAP_EC_INVALID_PARAM)
    {
        printf("CAP_EC_INVALID_PARAM\n");
    }
    if (ErrorCode == CAP_EC_TIMEOUT)
    {
        printf("CAP_EC_TIMEOUT\n");
    }
    if (ErrorCode == CAP_EC_NOT_ENOUGH_MEMORY)
    {
        printf("CAP_EC_NOT_ENOUGH_MEMORY\n");
    }
    if (ErrorCode == CAP_EC_UNKNOWN_ERROR)
    {
        printf("CAP_EC_UNKNOWN_ERROR\n");
    }
    if (ErrorCode == CAP_EC_ERROR_STATE)
    {
        printf("CAP_EC_ERROR_STATE\n");
    }
    if (ErrorCode == CAP_EC_HDCP_PROTECTED_CONTENT)
    {
```

```
        printf("CAP_EC_HDCP_PROTECTED_CONTENT\n");
    }
}

BOOLWINAPI CaptureVideo(VIDEO_SAMPLE_INFOVideoInfo, BYTE *pbData, LONGlLength,
__int64tRefTime, LONGlUserData);
BOOLbGetData = FALSE;
Mat ans2;
int main(intargc, char** argv)
{
    LONGlRetVal;
    DWORDdwDeviceNum;
    DWORDdwDeviceIndex = 0;
    HANDLEhAverCapturedevice[10];
    //Device Control
    //1. Get Device Number
    lRetVal = AVerGetDeviceNum(&dwDeviceNum);

    if (lRetVal != CAP_EC_SUCCESS) {
        printf("\nAVerGetDeviceNum Fail");
        ErrorMsg(lRetVal);
        system("pause");
    }
    if (dwDeviceNum == 0) {
        printf("NO device found\n");
        system("pause");
    }
    else {
        printf("Device Number = %d\n", dwDeviceNum);
    }

    //2. Create device representative object handle
    for (DWORDdwDeviceIndex = 0; dwDeviceIndex<dwDeviceNum; dwDeviceIndex++) {
        lRetVal = AVerCreateCaptureObjectEx(dwDeviceIndex, DEVICETYPE_ALL, NULL,
&hAverCapturedevice[dwDeviceIndex]);
        if (lRetVal != CAP_EC_SUCCESS) {
            printf("\nAVerCreateCaptureObjectEx Fail\n");
            ErrorMsg(lRetVal);
            system("pause");
        }
    }
}
```

```
        else
            printf("\nAVerCreateCaptureObjectEx Success\n");
    }
    //3. Start Streaming//

    //3.1 set video source
    //lRetVal = AVerSetVideoSource(hAverCaptureddevice[0], 3);
    lRetVal = AVerSetVideoSource(hAverCaptureddevice[0], 3); //HDMI 3, Composite
0

    //3.2 set Video Resolution &FrameRate
    VIDEO_RESOLUTION VideoResolution = { 0 };
    INPUT_VIDEO_INFO InputVideoInfo;
    ZeroMemory(&InputVideoInfo, sizeof(InputVideoInfo));
    InputVideoInfo.dwVersion = 2;
    Sleep(500);
    lRetVal = AVerGetVideoInfo(hAverCaptureddevice[0], &InputVideoInfo);
    VideoResolution.dwVersion = 1;
    VideoResolution.dwVideoResolution = VIDEORESOLUTION_1280X720;
    //VideoResolution.dwVideoResolution = VIDEORESOLUTION_1920X1080;

    lRetVal = AVerSetVideoResolutionEx(hAverCaptureddevice[0], &VideoResolution);
    lRetVal = AVerSetVideoInputFrameRate(hAverCaptureddevice[0], 6000);

    //3.3 Start Streaming
    lRetVal = AVerStartStreaming(hAverCaptureddevice[0]);
    if (lRetVal != CAP_EC_SUCCESS) {
        printf("\nAVerStartStreaming Fail\n");
        ErrorMsg(lRetVal);
        //system("pause");
    }
    else
    {
        printf("\nAVerStartStreaming Success\n");
        //system("pause");
    }

    //4. Capture Single Image
#endif 0
```

```
CAPTURE_IMAGE_INFO m_CaptureImageInfo = { 0 };

char text[] = "E:\\avermedia.bmp";
wchar_t wtext[20];

#define _CRT_SECURE_NO_WARNINGS
#pragma warning( disable : 4996 )
    mbstowcs(wtext, text, strlen(text) + 1); // Plus null
    LPWSTR m_strSavePath = wtext;
    CAPTURE_SINGLE_IMAGE_INFO pCaptureSingleImageInfo = { 0 };
    pCaptureSingleImageInfo.dwVersion = 1;
    pCaptureSingleImageInfo.dwImageType = 2;
    pCaptureSingleImageInfo.bOverlayMix = FALSE;
    pCaptureSingleImageInfo.lpFileName = m_strSavePath;
    // pCaptureSingleImageInfo.rcCapRect = 0;
    lRetVal = AVerCaptureSingleImage(hAverCapturedevice[0],
    &pCaptureSingleImageInfo);
    printf("\\AverCaptureSingleImage\\n");
    ErrorMessage(lRetVal);
#endif
#if 1
    // video capture
    VIDEO_CAPTURE_INFO VideoCaptureInfo;
    ZeroMemory(&VideoCaptureInfo, sizeof(VIDEO_CAPTURE_INFO));
    VideoCaptureInfo.bOverlayMix = FALSE;
    VideoCaptureInfo.dwCaptureType = CT_SEQUENCE_FRAME;
    VideoCaptureInfo.dwSaveType = ST_CALLBACK_RGB24;
    VideoCaptureInfo.lpCallback = CaptureVideo;
    VideoCaptureInfo.lCallbackUserData = NULL;
    lRetVal = AVerCaptureVideoSequenceStart(hAverCapturedevice[0],
    VideoCaptureInfo);
    if (FAILED(lRetVal))
    {
        return lRetVal;
    }
    //system("pause"); // hange up

#endif
inti;
scanf_s("%d", &i, 4); // must input any number in console !!
```

//5. Stop Streaming

```
lRetVal = AVerCaptureVideoSequenceStop(hAverCapturedevice[0]);
lRetVal = AVerStopStreaming(hAverCapturedevice[0]);
//printf("\nAVerStopStreaming Success\n");
ErrorMsg(lRetVal);

return 0;
}

BOOLWINAPI CaptureVideo(VIDEO_SAMPLE_INFOVideoInfo, BYTE *pbData, LONGlLength,
__int64tRefTime, LONGlUserData)
{
    if (!bGetData)
    {
        //ans2 = Mat(VideoInfo.dwHeight, VideoInfo.dwWidth, CV_8UC3,
(uchar*)pbData).clone();//single capture image
        ans2 =Mat(VideoInfo.dwHeight, VideoInfo.dwWidth, CV_8UC3,
(uchar*)pbData); //sequence capture image
        bGetData = TRUE;
    }
    imshow("ans2", ans2);

    waitKey(1);
    returnTRUE;
}
```