OpenCV in MFC

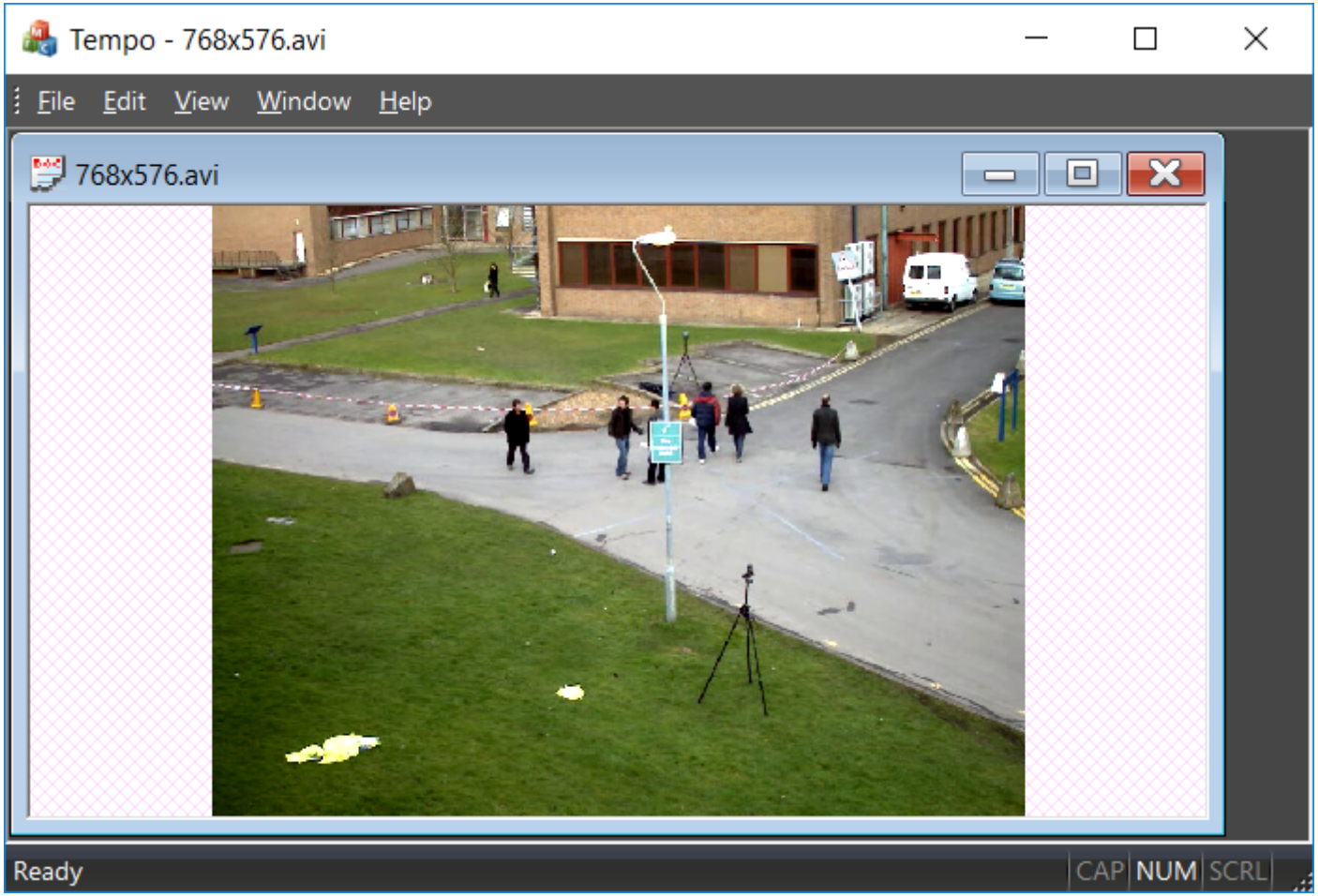
_Flaviu

11 Sep 2018 CPOL 2 min read

Rate me: 4.75/5 (10 votes)

A way to use OpenCV in MFC project

Download source code - 1 MB



Project (OneDrive): https://onedrive.live.com/embed?cid=DEDCB6EF190B8FD4&resid=DEDCB6EF190B8FD4%21411&authkey=ANtICXVGKPsQ030

Introduction

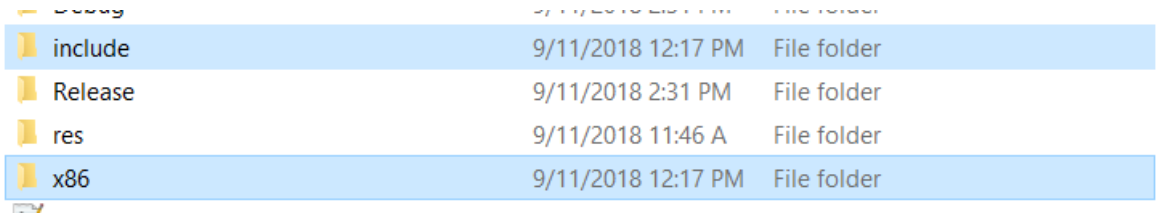
Several times, I have read questions regarding how to use OpenCV in MFC projects, so I decided to made a simple application demo, a VS2010 project, that reveals a way to use computer vision library inside of VC/MFC project, developed in an MDI application. Through my findings, as far as I remember, I saw somewhere a `CDialog` based app that embeds OpenCV ... here is another approach: multi-document interface, where you can load multiple media files.

Background

This article does not treat the OpenCV compilation, it does contain the compiled OpenCV library, ready to go, inside of two folders: *include*, and *res86*, all of them are included inside the project. On the internet, you can easily find the way in which to compile OpenCV, this is not the purpose of this article.

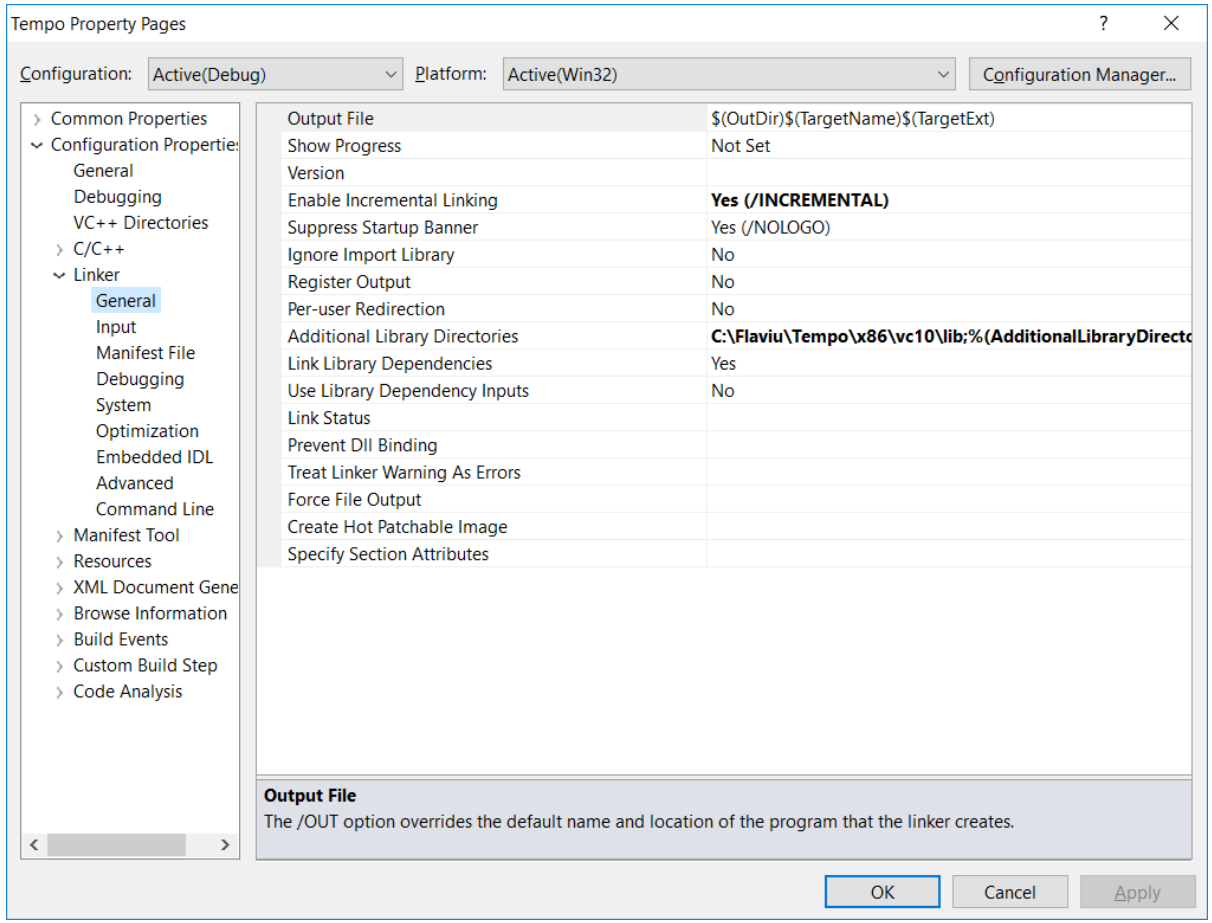
Step to Setup the Project

1. Put "*input*" and "*x86*" folders inside your project:

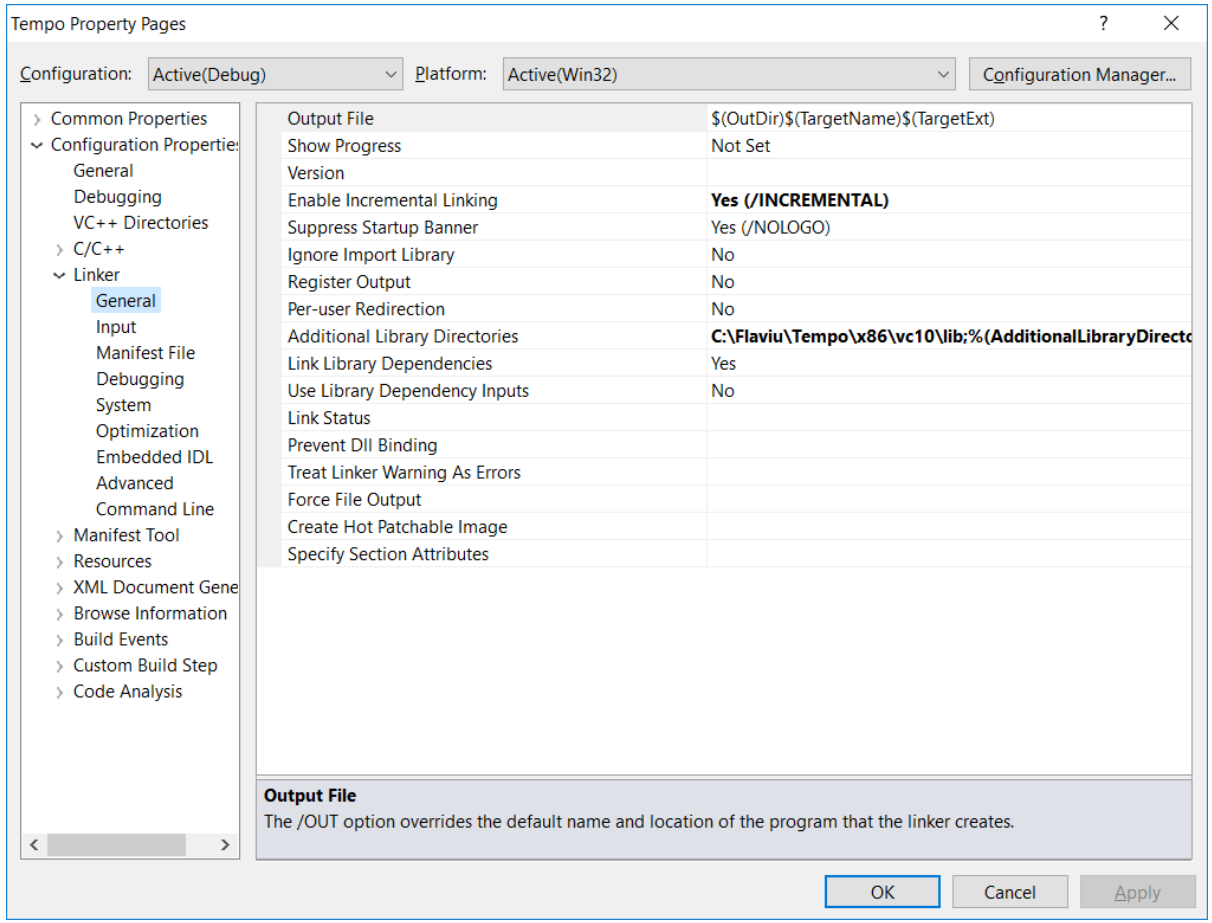


2. Setup project settings as follows:

Additional include directories: C:\Flaviu\Tempo\include;%(AdditionalIncludeDirectories) (replace "C:\Flaviu\" with your path).



Additional library directories: *C:\Flaviu\Tempo\x86\vc10\lib;%(AdditionalLibraryDirectories)* (replace "*C:\Flaviu*" with your path).



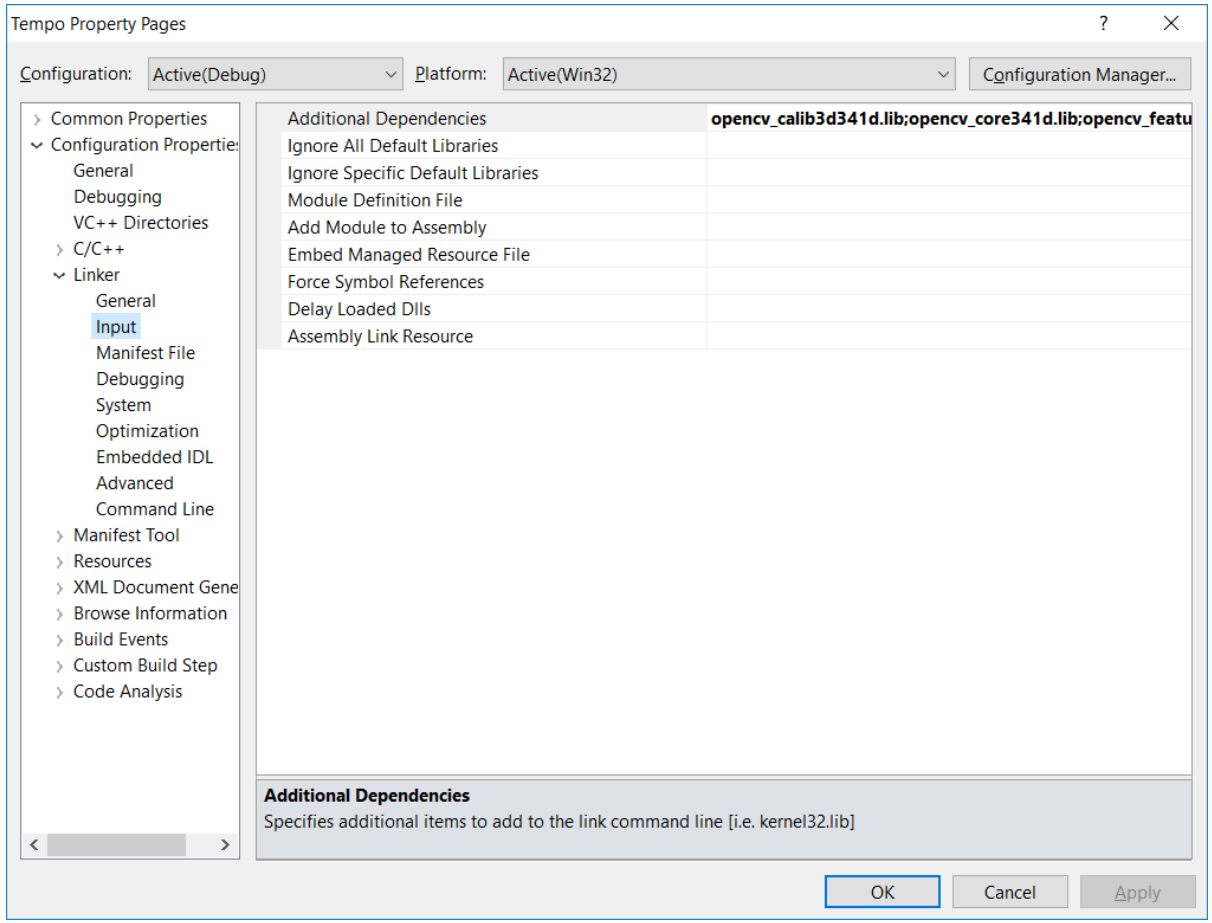
Additional dependencies:

Copy Code

```
opencv_calib3d341d.lib
opencv_core341d.lib
opencv_features2d341d.lib
opencv_flann341d.lib
opencv_highgui341d.lib
opencv_imgcodecs341d.lib
opencv_imgproc341d.lib
opencv_ml341d.lib
opencv_objdetect341d.lib
opencv_photo341d.lib
opencv_shape341d.lib
opencv_stitching341d.lib
```

```
opencv_superres341d.lib
opencv_video341d.lib
opencv_videoio341d.lib
opencv_videostab341d.lib
```

in Debug case, and the same libs for Release but without "d" from file names. Example: *opencv_calib3d341.lib* as *opencv_calib3d341d.lib*.



Using the Code

The structure of the project is pretty simple: the document class contains a `cv::VideoCapture` object, used to read video files, a `cv::Mat` object that is needed to convert `cv::VideoCapture` into `cv::Mat` object, and a `BITMAPINFO` structure used to convert `cv::Mat` object into `bitmap` object in order to be used for `CView` rendering. This conversion could be found on `CTempoDoc::SetupBitmapInfo` method.

C++ Copy Code

```
void CTempoDoc::SetupBitmapInfo(cv::Mat& mat)
{
    if(NULL != m_pBmi)
    {
        delete m_pBmi;
        m_pBmi = NULL;
    }
    m_pBmi = new BITMAPINFO;
    BITMAPINFOHEADER* pHeader = &m_pBmi->bmiHeader;
    pHeader->biSize             = sizeof(BITMAPINFOHEADER);
    pHeader->biPlanes           = 1;
    pHeader->biCompression      = BI_RGB;
    pHeader->biXPelsPerMeter    = 100;
    pHeader->biYPelsPerMeter    = 100;
    pHeader->biClrUsed          = 0;
    pHeader->biClrImportant     = 0;
    pHeader->biWidth            = m_Mat.cols;
    pHeader->biHeight           = -m_Mat.rows;
    pHeader->biBitCount         = 24;
    m_pBmi->bmiHeader.biSizeImage = 0;
}
```

I should say that `CTempoDoc` has a tricky method: `ResizeMat(cv::Mat& mat)` ... this method corrects the `cv::Mat` that does not have the right format (actually, add a new column to matrix `OpenCV` object retrieved from `cv:imread`, when it is needed).

The rendering is handled inside `CView` class. Here, you can spot the well known `OnDraw` method, which renders the content of `CTempoDoc::BITMAPINFO` content, double-buffered. And a timer is implemented for the case when the loaded file is a video stream, and this timer will know when to load and render the next frame from video. Also, this class contains another handler that takes care of how the content is erased and drawn on `CView`, when the size of the window has been changed or when the next frame of video is ready to render, and some methods that scale the images inside window. All of them can be explored inside the sample project.

Here, you have an application that can load static images or all kinds of videos into an MDI app. Enjoy it!

Points of Interest

The computer vision library is used more and more, and I offer a way here to include your OpenCV algorithms inside a real life desktop application.

History

- 11th September, 2018 - Published the article

License

Share

About the Author

_Flaviu
Romania

No Biography provided

Comments and Discussions

You must Sign In to use this message board.

Search Comments

Spacing Relaxed Layout Normal Per page 25 Update

First Prev Next

I can't download from onedriver	**yousse**	**9-Oct-19 18:26**
Re: I can't download from onedriver	_Flaviu	1-Jan-20 8:34
Cannot implement grayed image with openCV	**ptpham**	**15-May-19 22:30**
compile failed	**wallollaw**	**12-Sep-18 14:46**
Re: compile failed	_Flaviu	12-Sep-18 21:21
have you consider to post this as a tip?	**Nelek**	**11-Sep-18 9:11**

Last Visit: 31-Dec-99 18:00 Last Update: 13-Oct-21 19:50 Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.