Subject: OpenCV
Posted by GaroRobe on Sat, 28 May 2011 06:49:53 GMT

When trying to build program with subj I get numerous "ambiguous symbol" errors in opencv2/core/types_c.h and uppsrc\Core\Defs.h files.

How do I fix it? Plus, maybe there are some practices using algorithms from OpenCV in U++?

Subject: Re: OpenCV
Posted by koldo on Mon, 30 May 2011 06:31:14 GMT

Hello Artem

Kasome did a post with a nice sample here.

If you need more help we can upload a simpler demo.

Are you using linux or windows?

Subject: Re: OpenCV
Posted by GaroRobe on Tue, 31 May 2011 06:10:08 GMT

Well, as far as I can see in given example only opencv v1 was used. I use 2.2 (opencv2 API).

No fuctional yet, just exploring:

```
#include <opencv2/opencv.hpp>
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

void proto1::FileOpenClick()
{
 selectDlg.ExecuteOpen();
 IplImage* img = cvLoadImage(~selectDlg);
}
```

Gives erros:

Quote:c:\upp\uppsrc\core\config.h(97) : warning C4005: '_CRT_SECURE_NO_DEPRECATE' : macro redefinition
        C:\OpenCV2.2\include\opencv2/core/types_c.h(47) : see previous definition of '_CRT_SECURE_NO_DEPRECATE'

C:\MyApps\proto1\main.cpp(20) : error C2040: 'img' : 'IplImage *' differs in levels of indirection from 'Upp::Image'

Subject: Re: OpenCV
Posted by koldo on Tue, 31 May 2011 06:37:35 GMT
View Forum Message <> Reply to Message

Hello Artem

Today I will upload you an Open CV demo.

Subject: Re: OpenCV
Posted by koldo on Thu, 02 Jun 2011 06:45:41 GMT
View Forum Message <> Reply to Message

Hello Artemm

The package is uploaded in Bazaar. See here.

The sample detects lines in the picture using Hough transform. This is the result:

Please do not forget to set the folders in Build methods menu:

- INCLUDE directories
\plugin\OpenCV\include
C:\Desarrollo\Upp\plugin_BSD\OpenCV\include\opencv

- LIB directories
\plugin\OpenCV\lib

- PATH - executable directories
\plugin\OpenCV\bin

File Attachments
1) dib.PNG, downloaded 1358 times

Subject: Re: OpenCV
Posted by tojocky on Thu, 02 Jun 2011 08:17:46 GMT
View Forum Message <> Reply to Message

Very interesting,

But I am a little confused. What library is better to use for 2d and 3d acceleration.

From practice can somebody tell what to recommend?

Subject: Re: OpenCV
Posted by koldo on Thu, 02 Jun 2011 09:29:51 GMT
View Forum Message <> Reply to Message

tojocky wrote on Thu, 02 June 2011 10:17Very interesting,

But I am a little confused. What library is better to use for 2d and 3d acceleration.

From practice can somebody tell what to recommend?
Hello Ion

I do not understand you. OpenCV is used for image processing.

You open images or real time video and process it, extracting the info you want.

It also accepts stereo images and lets you calibrate the camera and spatially position picture elements as in photogrametry.

Subject: Re: OpenCV
Posted by tojocky on Thu, 02 Jun 2011 09:57:43 GMT
View Forum Message <> Reply to Message

koldo wrote on Thu, 02 June 2011 12:29
Hello Ion

I do not understand you. OpenCV is used for image processing.

You open images or real time video and process it, extracting the info you want.

It also accepts stereo images and lets you calibrate the camera and spatially position picture elements as in photogrametry.
From my question "2d and 3d acceleration" is a little confuse.

I mean that exist other libraries for image processing which uses gpu acceleration.

It is better to uses some optimized library than reinventing the wheel. In my example about reinventing the wheel was how to autolevel/autocorect an image.

And question:
From your experience what library is the recommended for use in image processing?

Subject: Re: OpenCV
Posted by koldo on Thu, 02 Jun 2011 11:06:18 GMT

Hello Ion

OpenCV is recognized as the fastest. See for example this comparison taken from the "Learning OpenCV" book. The number of users is also very high.

Added:
However fastest is not always the best. In my case OpenCV camera calibration functions do not calibrate properly the cameras I use...

Subject: Re: OpenCV
Posted by GaroRobe on Thu, 02 Jun 2011 23:07:41 GMT

There is "gpu" subfolder in lib, but I didn't have the time to look into it yet. Some structures and routines are definetly already there. Can't say what they do though

BTW, have another noob question, a bit offtopic though. When I include <opencv22/opencv2/opencv.hpp>, everything else is included recursively and builds fine, but not everything is found by Ctrl+Space, e.g. if I try to explore "cv" namespace, I try typing "cv::" and press Assist shortcut, but get nothing. By just pressing shortcut, I don't even see "cv" namespace in the list. Once again, why?

Subject: Re: OpenCV
Posted by koldo on Fri, 03 Jun 2011 05:18:18 GMT

Uuff. That is a weak point for me.

Honza, (or other) could you answer him?

Subject: Re: OpenCV
Posted by tojocky on Fri, 03 Jun 2011 07:06:51 GMT

koldo wrote on Thu, 02 June 2011 14:06Hello Ion

OpenCV is recognized as the fastest. See for example this comparison taken from the "Learning OpenCV" book. The number of users is also very high.

Added:

However fastest is not always the best. In my case OpenCV camera calibration functions do not calibrate properly the cameras I use...
Thank you Koldo!

Subject: Re: OpenCV
Posted by dolik.rce on Fri, 03 Jun 2011 07:34:13 GMT
View Forum Message <> Reply to Message

GaroRobe wrote on Fri, 03 June 2011 01:07BTW, have another noob question, a bit offtopic though. When I include <opencv22/opencv2/opencv.hpp>, everything else is included recursively and builds fine, but not everything is found by Ctrl+Space, e.g. if I try to explore "cv" namespace, I try typing "cv::" and press Assist shortcut, but get nothing. By just pressing shortcut, I don't even see "cv" namespace in the list. Once again, why?
It is quite simple The parser in theide ignores namespaces completely. It is a tradeof to get the parsing faster. You should be able too see most of the methods (classes, variables, macros, etc...) by just skipping the namespace part. Unless the code is too cryptic, uses macros extensively, or something like that, that the parser chokes on, but that doesn't happen too often.

Honza

Subject: Re: OpenCV
Posted by GaroRobe on Sat, 04 Jun 2011 05:12:27 GMT
View Forum Message <> Reply to Message

Don't quite understand. What do you mean "skipping the namespaces"?
Well, it seems that OpenCV C++ API is right that "too cryptic" case. Take, for example, cv::calibrateCamera method. I might use cvCalibrateCamera (OpenCV C API), but I, peronally, find such decisions inconsistent - I use C++ because I want to use it, not because I can use C syntax in it.

Subject: Re: OpenCV
Posted by dolik.rce on Sat, 04 Jun 2011 06:31:13 GMT
View Forum Message <> Reply to Message

GaroRobe wrote on Sat, 04 June 2011 07:12Don't quite understand. What do you mean "skipping the namespaces"?
Well, it seems that OpenCV C++ API is right that "too cryptic" case. Take, for example, cv::calibrateCamera method. I might use cvCalibrateCamera (OpenCV C API), but I, peronally, find such decisions inconsistent - I use C++ because I want to use it, not because I can use C syntax in it.
By "skipping namespaces" I meant just not typing them E.g. type just calibrat and Assist should show calibrateCamera. It works as if there was always appropriate using namespace ... even when there isn't

---

Honza

Subject: Re: OpenCV
Posted by GaroRobe on Sat, 04 Jun 2011 08:57:35 GMT
View Forum Message <> Reply to Message

My point is it doesn't. See for it yourself

Subject: Re: OpenCV
Posted by dolik.rce on Sat, 04 Jun 2011 15:14:43 GMT
View Forum Message <> Reply to Message

Ok, my bad... Now I actually looked at the package and it is all clear Sorry for misleading you before.

The way the package is constructed, it uses libraries and include files from outside of U++. And Assist++ parser only reads the files that are present in the packages. That is another trade-off for speed, parsing all the system headers would be very slow.

Also, while I was looking at the pacakge, I fixed it to compile and link on Linux, but I get a lot of memory leaks. Looking at the valgrind output, it seems that most of them is associated with gtk functions. My guess is that the opencv windowing functions (like cvNamedWindow) are to be blamed. It would be probably better to find a way to export the images to Upp::Image and handle all the GUI using U++. I haven't time to do that so far though...

Best regards,
Honza

Subject: Re: OpenCV
Posted by GaroRobe on Sun, 05 Jun 2011 00:26:38 GMT
View Forum Message <> Reply to Message

I'll look into ways to import/export later, for now I compile mainly in win32 with msvc10 - it's the main target platform FOR NOW anyway. But lacking assist is really annoying.

Well what do we have:
1. Assist++ works good dynamically in most cases.
2. Assist++ fails in some obscure cases which are much-much-more less frequent, but sometimes annoying as hell.

My point is...:
3. Why not have an additional functionality for packages in form of Assist++ hinting dictionaries? Make it so that package maintainers might make such dictionaries (say, "func

prototype"/"header"[-"string#"][/"src"[-"string#"]][/ "descr"]) and assist would attach 'em to its list once when package is attached and the rest of A++ would work dynamically like it does now.

I really don't know A++ internals and don't really understand all the problems following names parsing, but such function would be, to my mind, a decent tradeoff.

Subject: Re: OpenCV
Posted by koldo on Sun, 05 Jun 2011 05:02:06 GMT
View Forum Message <> Reply to Message

Hello Honza

It would be probably better to find a way to export the images to Upp::Image and handle all the GUI using U++. I haven't time to do that so far though...
No problem. As I did not like OpenCV windowing system (HighGui) I included in OpenCV a U++ image window class and functions to convert images from/to U++ and OpenCV .

Go to:
IplImage *ImageToCv(Image &img, bool iscolor = false);
Image CvToImage(IplImage *imgCV);
class ImageWindow;

Subject: Re: OpenCV
Posted by GaroRobe on Sun, 05 Jun 2011 06:22:22 GMT
View Forum Message <> Reply to Message

Sweet!

Subject: Re: OpenCV
Posted by dolik.rce on Sun, 05 Jun 2011 20:36:07 GMT
View Forum Message <> Reply to Message

GaroRobe wrote on Sun, 05 June 2011 02:263. Why not have an additional functionality for packages in form of Assist++ hinting dictionaries?
Make it so that package maintainers might make such dictionaries (say, "func prototype"/"header"[-"string#"][/"src"[-"string#"]][/ "descr"]) and assist would attach 'em to its list once when package is attached and the rest of A++ would work dynamically like it does now.
That is actually quite good idea It should be even relatively easy to implement. There is already one "backdoor" to feed information to Assist++ or to hide it, the parser directives //$, //$- and //$+. If we can get the description of the entire OpenCV API in C++ syntax, dumped in one file and prefixed with //$, it should make the Assist work. Of your course better (faster) solution would be to store the pre-parsed data, but the parser is quite fast, maybe it wouldn't even be necessary...

Honza

Subject: Re: OpenCV
Posted by GaroRobe on Sun, 05 Jun 2011 23:41:47 GMT

I'll look into it immidiately.

Okay, what I managed to dig so far is that A++ fails to parse prototypes with macroes in function arguments, e.g.:
//$ void foo(MY_DEFINE int bar)
fails, while
//$ void foo(int bar)
works.
OpenCV is tangled with CV_[SOMETHING] inside out, so it's quite a headache and calls for some intensive sed and awk usage .

Moreover, that's not the only one case:
//$ CV_EXPORTS void RQDecomp3x3 (const Mat& M, Mat& R, Mat& Q);
Works.
//$ CV_EXPORTS_W Vec3d RQDecomp3x3 (const Mat& M, Mat& R, Mat& Q, Mat& Qx, Mat& Qy, Mat& Qz);
Fails.

Don't get the reason.

Subject: Re: OpenCV
Posted by dolik.rce on Mon, 06 Jun 2011 07:06:52 GMT

Well, the parser is not perfect... This might be a good reason to make it work better

Extensive playing with sed/awk might not be necessary. There is plenty of tools that work with C++ code/binaries, we can reuse them. One possible way that comes to my mind: Compiling a binary using OpenCV in debug mode and listing the files in gdb, using its 'info functions', 'info variables' and 'info types' commands. It lists the declarations sorted by file, so filtering out the OpenCV part would be easy. One thing I'm not sure how to get is the information about classes...

Honza

EDIT: Found how to get the classes/struct/union info There is a 'ptype' command for this in gdb So it is in theory possible to get all the information with relatively simple script...

Subject: Re: OpenCV
Posted by GaroRobe on Mon, 06 Jun 2011 11:38:52 GMT

To my shame never used gdb beyond step-to-step debugging in eclipse. So much to do, so much

to learn...

Subject: Re: OpenCV
Posted by GaroRobe on Mon, 04 Jul 2011 00:12:58 GMT
View Forum Message <> Reply to Message

Okay, enough of being full of myself - I will find neither time nor bravery to start solving this problem so I do what might solve it - just ask for your help. Anyone who can comprehend solution completely, please, post here everything related - this feature would be really useful when start working with completely foreign API.