

TUTORIAL



Installation and Configuration OpenCV 1.1 on Borland C++ Builder 6.0

Sunu Wibirama

Hamamoto Laboratory
Department of Information Media Technology
School of Information Science and Technology
Tokai University, JAPAN

WARNING

This tutorial is free and can be redistribute without prior permission from author. However, author does not provide ANY WARRANTY TO USE THIS TUTORIAL FOR DEVELOPING COMMERCIAL SOFTWARE. Please use this tutorial in a kind manner for supporting your study and research. No further support is provided. Please use Google to overcome your problem when you are experiencing any trouble.

Copyright (c) 2011 Sunu Wibirama (<http://wibirama.com>). All right reserved.

A. Introduction

OpenCV 1.1 library is initially developed under Microsoft Visual Studio platform (running under Microsoft Windows). Therefore, all *.lib files were generated under Visual Studio compiler. Borland C++ Builder uses different method to import libraries. More information can be referred in [1].

In this tutorial, we will explain several steps to install and configure OpenCV 1.1 on Borland C++ Builder. This tutorial is mainly based and inspired by [2] and [3]. We will also explain some tricks to overcome conflict between highgui.dll and several ActiveX libraries (such as wxWidgets or VideoOCX), when both of them are used simultaneously to conduct real-time video processing.

B. Installation

Firstly, please download OpenCV 1.1 from Sourceforge, as shown in Fig.1 below. Install OpenCV 1.1 in directory : C:\Program Files\OpenCV1.1

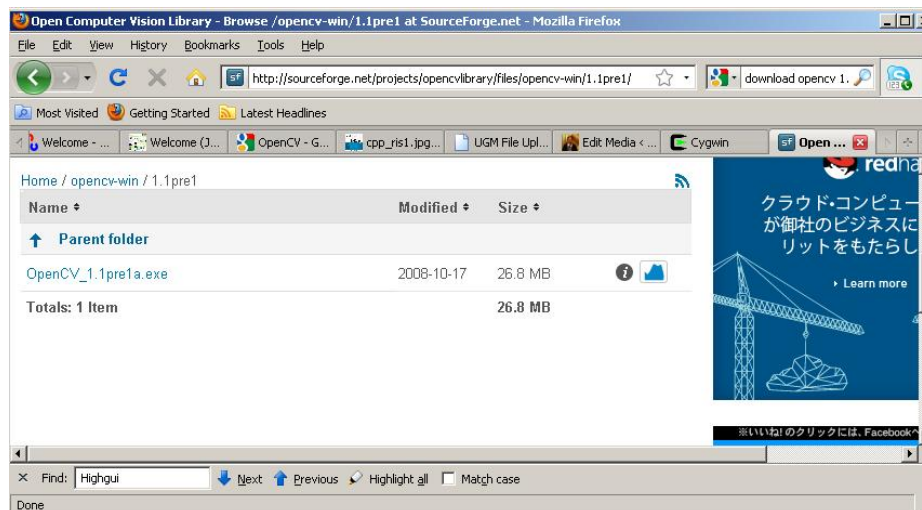


Fig.1. Download OpenCV_1.1pre1a.exe from Sourceforge

To use OpenCV 1.1 with Borland C++ Builder (BCB), all *.lib files should be converted in BCB format. This can be done by either using *coff2omf* utility from BCB or by compiling directly the source code using *makefile*. In the end of this second section, we will provide Complete explanation can be found in [1] :

1. Converting library using *coff2omf*

BC++ comes with the command line utility *coff2omf*. Here is a (Cygwin bash) shell command to achieve the conversion of all .libs (it is equally possible to do this with some Windows script). Run it in the lib subdirectory of the OpenCV installation directory:

```
mkdir -p borland
for i in *.lib; do coff2omf $i borland/$i; done
```

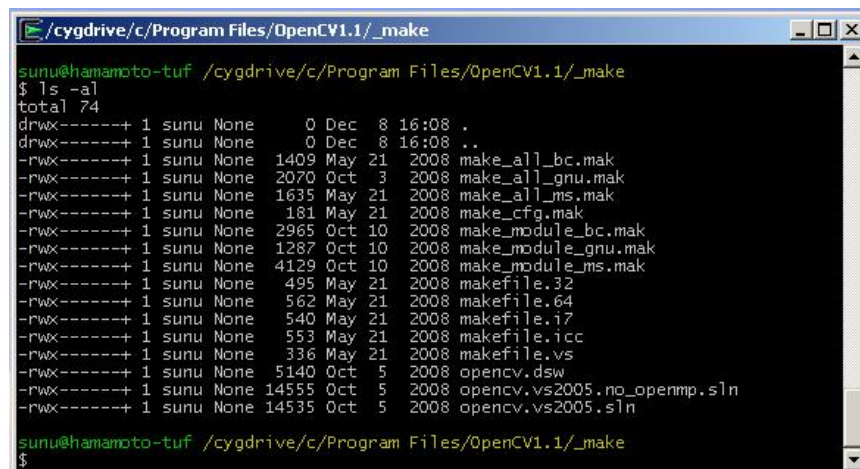
You can use the script below for MS Windows:

```
mkdir borland
for %%F in (*.lib) do coff2omf %%F borland\%%F
```

Now the import libraries in `lib\borland` can be used by BC++ - just add them to your project.

`bcc32.exe` has a switch (among many others): `-VM` for Microsoft Visual C++ compatibility (see the output of `bcc32 -h -V`). Its corresponding IDE option (at least in Borland Developer Studio 2006) can be found under source under C++ options of the project options.

2. Compiling from source code



```
sunu@hamamoto-tuf /cygdrive/c/Program Files/OpenCV1.1/_make
$ ls -al
total 74
drwx-----+ 1 sunu None    0 Dec  8 16:08 .
drwx-----+ 1 sunu None    0 Dec  8 16:08 ..
-rwx-----+ 1 sunu None  1409 May 21 2008 make_all_bc.mak
-rwx-----+ 1 sunu None  2070 Oct  3 2008 make_all_gnu.mak
-rwx-----+ 1 sunu None  1635 May 21 2008 make_all_ms.mak
-rwx-----+ 1 sunu None   181 May 21 2008 make_cfg.mak
-rwx-----+ 1 sunu None  2965 Oct 10 2008 make_module_bc.mak
-rwx-----+ 1 sunu None  1287 Oct 10 2008 make_module_gnu.mak
-rwx-----+ 1 sunu None  4129 Oct 10 2008 make_module_ms.mak
-rwx-----+ 1 sunu None   495 May 21 2008 makefile_32
-rwx-----+ 1 sunu None   562 May 21 2008 makefile_64
-rwx-----+ 1 sunu None   540 May 21 2008 makefile_i7
-rwx-----+ 1 sunu None   553 May 21 2008 makefile_icc
-rwx-----+ 1 sunu None   336 May 21 2008 makefile_vs
-rwx-----+ 1 sunu None   5140 Oct  5 2008 opencv.dsw
-rwx-----+ 1 sunu None 14555 Oct  5 2008 opencv.vs2005.no_openmp.sln
-rwx-----+ 1 sunu None 14535 Oct  5 2008 opencv.vs2005.sln
sunu@hamamoto-tuf /cygdrive/c/Program Files/OpenCV1.1/_make
$
```

Fig.2. Cygwin interface running on Microsoft Windows XP

To compile OpenCV, you can use file `_make\make_all_bc.mak` which is executed using Cygwin bash, as shown in Fig.2. To install Cygwin, please refer to [4].

3. Using ready to use *.lib files

If you are experiencing error when converting *.lib files or installation using source code, please download ready to use *.lib files from links below. List of libraries is shown in Fig.3.

Mirror 1 : <http://upload.ugm.ac.id/434bcb.zip>

Mirror2: <http://wibirama.com/dip/wp-content/uploads/2011/12/bcb.zip>

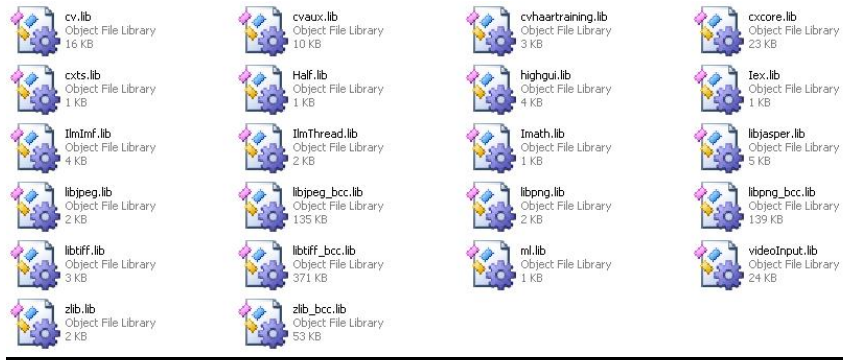


Fig.3. Ready to use *.lib files under BCB folder

C. Configuration

After completing conversion of all *.lib files, copy and paste all BCB version *.lib files under directory C:\Program Files\OpenCV1.1\lib\bcb (see Fig.4 below). All *.lib files must be copied, as shown in Fig.3.

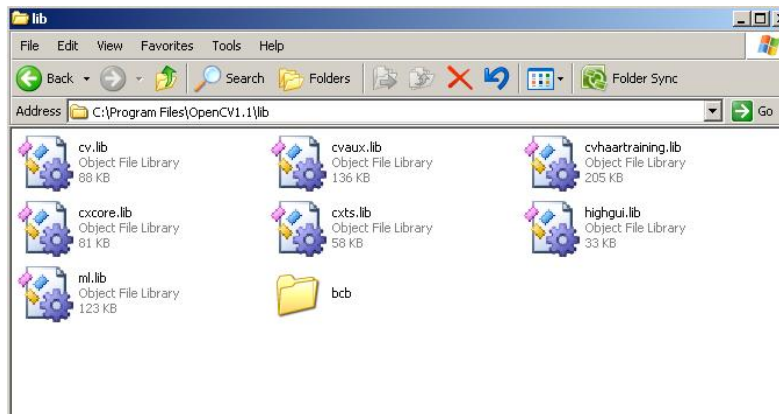


Fig.4. Create folder “bcb” under “lib” folder

Include all OpenCV 1.1 *.lib files in your BCB project. To do so, click View → Project Manager and *right-click* on your Project file as shown in Fig.5 below.

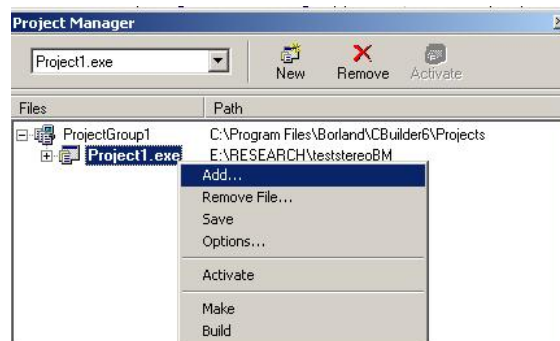


Fig.5. Adding library to BCB project

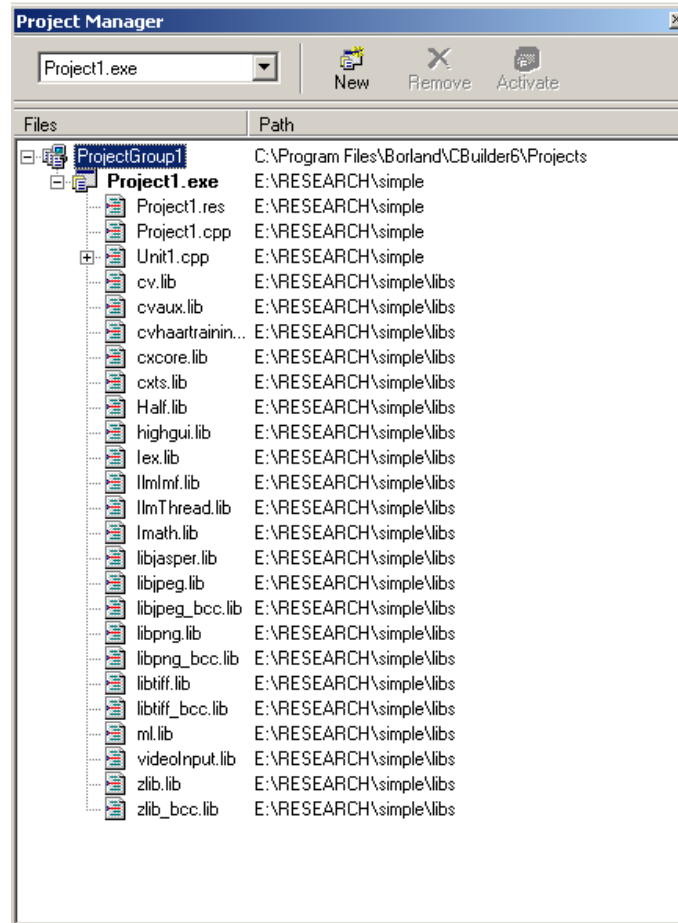


Fig.6. Included library in BCB project

All included libraries are shown in Fig. 6 above. Next, we should define the path to several header file related with these included libraries. To do so, open “Include Path” window by clicking:

Project → Options → Directories/Conditionals→Include Path.

Enter several path below in Include Path window (see Fig. 7):

```

C:\Program Files\OpenCV1.1\cv\include
C:\Program Files\OpenCV1.1\cvaux\include
C:\Program Files\OpenCV1.1\cxcore\include
C:\Program Files\OpenCV1.1\ml\include
C:\Program Files\OpenCV1.1\otherlibs\_graphics\include
C:\Program Files\OpenCV1.1\otherlibs\ffopencv
C:\Program Files\OpenCV1.1\otherlibs\highgui

```

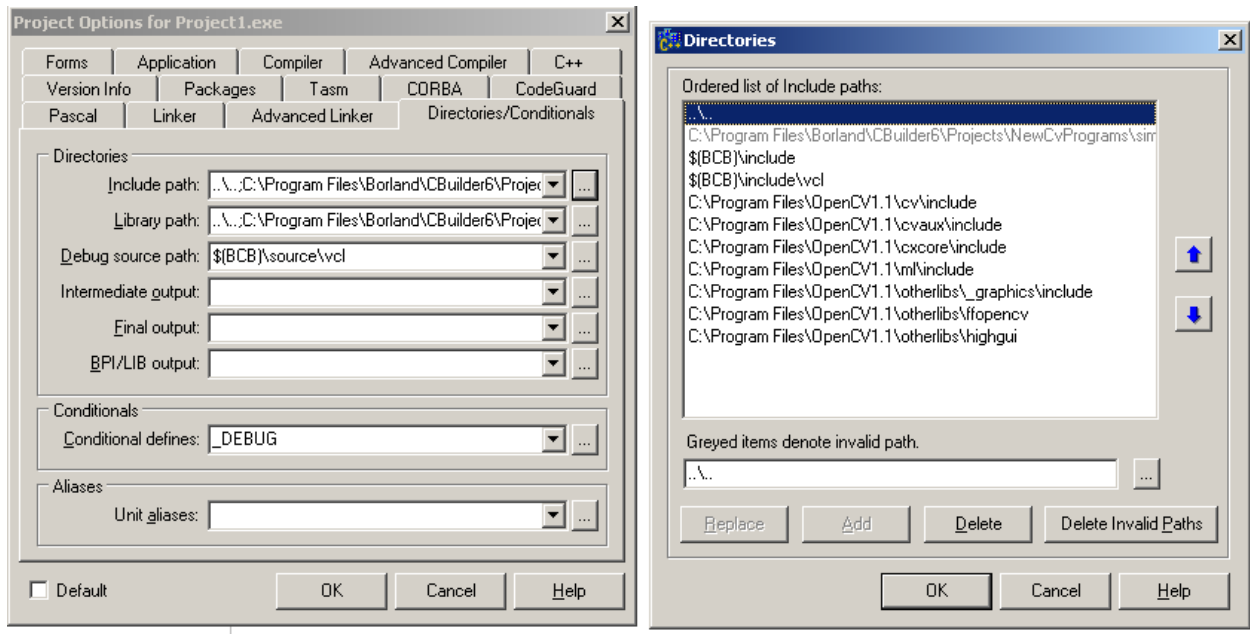


Fig.7. Path configuration in BCB project

Next, we should activate MFC compatibility to handle several OpenCV function, such as cvGetSize() and so on. Click “Advanced Compiler” and check “MFC Compatibility” (see Fig, 8).

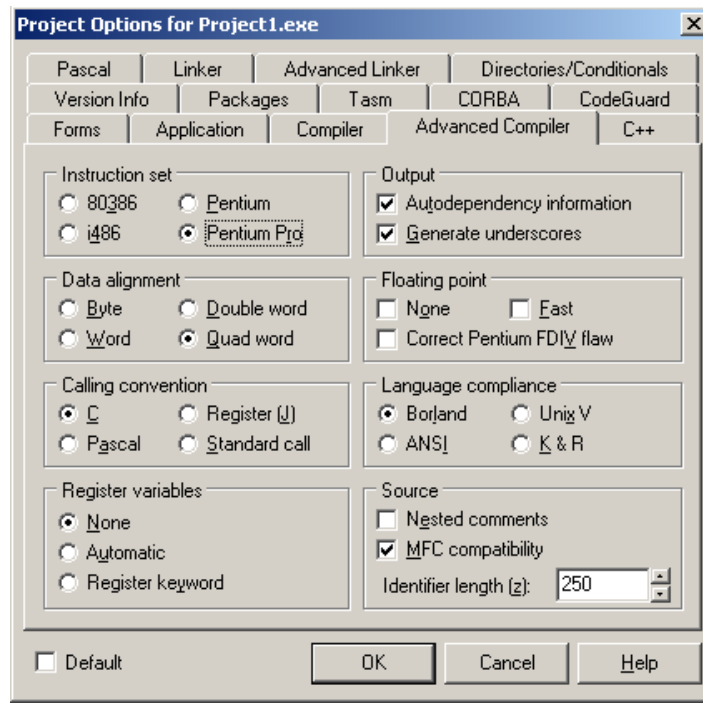


Fig.8. Activating MFC compatibility

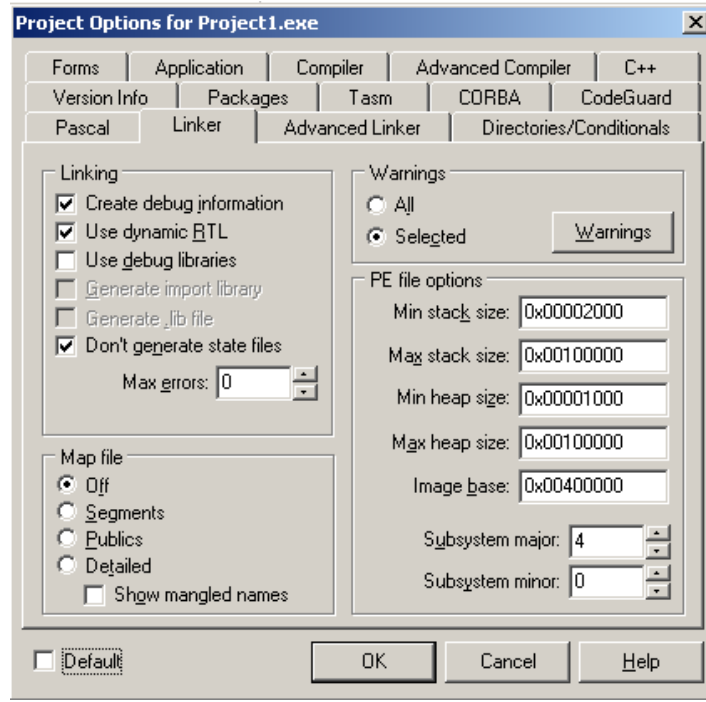


Fig.9. Use Dynamic RTL

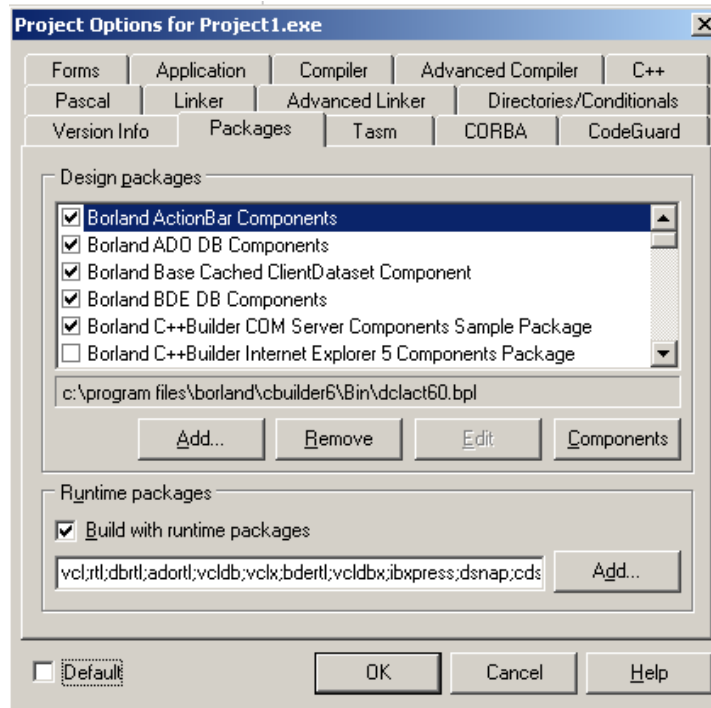
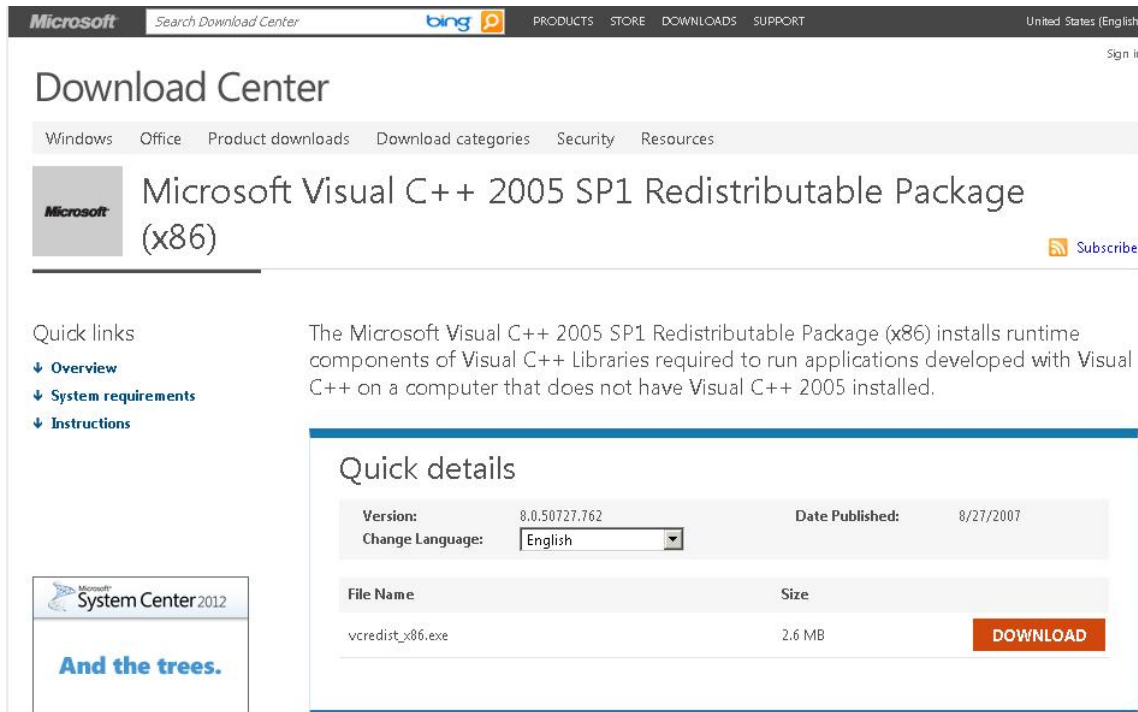


Fig.10. Build with runtime packages

Next, we should activate option “Use dynamic RTL” and “Build with runtime packages” to be able working with function related to haartraining.dll file. See Fig. 9 and Fig.10 to configure these options. Finally, we download and install “Microsoft Visual C++ 2005 SP1 redistributable Package (x86)” from [5] to provide additional support to several OpenCV files related to MSVCRT 8.0 (Visual Studio run-time).



The screenshot shows the Microsoft Download Center page for the Microsoft Visual C++ 2005 SP1 Redistributable Package (x86). The page includes a search bar, navigation links (Windows, Office, Product downloads, Download categories, Security, Resources), and a 'Sign in' link. The main content area features the product title, a description, and a 'Quick details' section with a table of file information.

File Name	Size	
vcredist_x86.exe	2.6 MB	DOWNLOAD

Fig.11. Microsoft Visual C++ 2005 Redistributable Package Download Page

To test your installation and configuration, we provide a free software called “Stereo BM Test – OpenCV1.1”. This software was developed under Borland C++ Builder 6.0 using OpenCV 1.1pre1a. This software was based on Stereo Calibration source code of “Learning OpenCV” book.

You can download “Stereo BM Test – OpenCV1.1” from these links:

Mirror 1: <http://wibirama.com/dip/wp-content/uploads/2011/12/teststereoBM.zip>

Mirror 2: <http://upload.ugm.ac.id/433teststereoBM.zip>

D. Using OpenCV 1.1 with ActiveX Library

When you are using ActiveX library to capture and process image from your camera (such as wxWidgets [6] or VideoOCX [7]), you should configure your OpenCV dll files since highgui.dll load OLE libraries before your ActiveX does. This normally resulting an error message in your application.

OLE is a technology that enables an application to create compound documents that contain information from a number of different sources. For example, a document in an OLE-enabled word processor can accept an embedded spreadsheet object. Unlike traditional "cut and paste" methods where the receiving application changes the format of the pasted information, embedded documents retain all their original properties. If the user decides to edit the embedded data, Windows activates the originating application and loads the embedded document [8].

To configure your application, click

Project → Options → Directories/Conditionals→Include Path

And open "Advanced Linker" tab as shown in Fig. 12 and proceed to "Delay load" section. Add cv110.dll, cxcore110.dll, and highgui110.dll as "Dlls to delay-load". Now, try to re-compile your application. ActiveX and OpenCV1.1 should work perfectly like a charm.

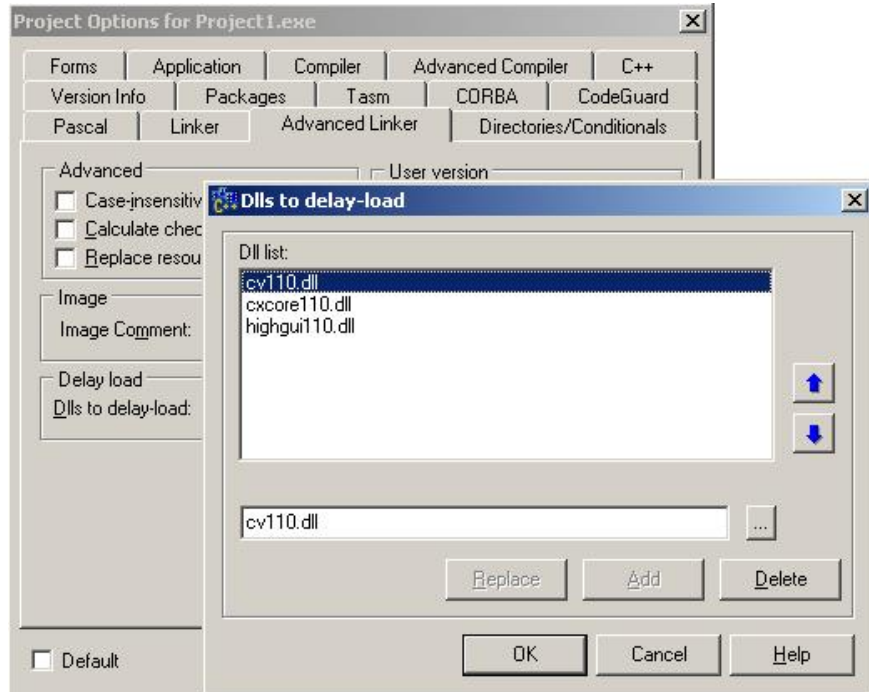


Fig.12. Delaying some dll files

References:

- [1] OpenCV with C++ Builder. Available: <http://opencv.willowgarage.com/wiki/C%2B%2BBuilder>
- [2] OpenCV 1.1, creating new project in Borland C++ Builder.
Available : http://www.compvision.ru/wiki/%D0%94%D0%BB%D1%8F_OpenCv_1.1pre1
(Russian Language)
- [3] OpenCV Installation. Available: <http://www.compvision.ru/forum/index.php?showtopic=2>
(Russian Language)
- [4] Official website of Cygwin. Available : <http://www.cygwin.com/>
- [5] Microsoft Visual C++ 2005 SP1 redistributable Package (x86).
Available: <http://www.microsoft.com/download/en/details.aspx?id=5638>
- [6] wxWidgets Official Website. Available : <http://www.wxwidgets.org/>
- [7] VideoOCX. Available: <http://www.rsoftr.com/other/videoocx-22877.html>
- [8] OLE Concepts and Requirements Overview. Available: <http://support.microsoft.com/kb/86008>