

ABOUT THIS DOCUMENT

This document was created as part of a final project for a BSc degree at the Academic College of Tel-Aviv Yaffo, by Rachely Esman and Yoad Snapir, under the supervision of [Tal Hassner](#).

As part of the project, we used Intel's [OpenCV](#) library, calling its functions from [MATLAB](#). We found the subject tricky and thus decided to share the experience of our work with others describing the common pitfalls.

You can use this document freely according to the copyrights noted below but under your sole responsibility.

Rachely and Yoad.

OPENCV AND MATLAB MEX FILES

This guide will provide a quick walk through on compiling C++ OpenCV modules as MEX runtime files of MATLAB under:

- MS Visual Studio 2005
- MATLAB 7.5.0
- Windows 32bit

And will probably be clear enough for other platforms / versions.

RECOMMENDED KNOWLEDGE

- Good understanding of C++ Compilation and linkage concepts.
- DLL general usage
- Visual Studio 2005 environment
- Basic MATLAB and MEX files knowledge

GENERAL CONCEPTS

MEX FILES

MEX files are actually simple .DLL files compiled with extension .MEXW32 using ordinary compilation tools provided with VS2005. Those .DLL files have a fixed entry point "mexFunction" with a fixed signature. (List of IN/OUT parameters)

Follow this link to get an overview of MEX files and information on using them:

<http://www.mathworks.com/support/tech-notes/1600/1605.html#intro>

MEX files are compiled (and linked) from within the MATLAB environment using the following syntax:

```
mex 'srcfile1.cpp' 'srcfile2.cpp' 'objfile1.obj' ...
```

Before compiling, you need to setup the compilation options using the command:

```
mex -setup
```

This actually creates an options file named "mexopts.bat" that will be used for further compilation and linkage. We will look more into that options file later.

OPENCV LIBRARY

OpenCV is a free computer vision development tools collection. The OpenCV source files are available for the developer and so are the libraries for linkage and dll's for runtime.

Since OpenCV development environment has its own complex setup procedure we will assume familiarity and a working OpenCV project in MS VS2005.

Also, the method described in this paper will try to avoid specific technical details regarding OpenCV.

THE PROCESS

MATLAB documentation does gives fairly good examples for compiling MEX file from standard C/C++ function. The problem with OpenCV is that it uses another large set of include files, libraries, and runtime dlls which makes MATLAB documentation practically useless.

When you setup the mex compilation options, the "mexopt.bat" file contains:

- The compiler name from VS2005 and additional include files location in the environment variable INCLUDE for MATLAB C++ functions.
- The linker name from VS2005 and additional MATLAB libraries to link with.

When you setup the OpenCV development environment in VS2005 you also need to specify include files for the compiler and libraries for the linker.

So basically two things need to be done:

1. Allowing VS2005 to compile and link MATLAB functions.
2. Allowing MATLAB to compile and link OpenCV functions.

No. 2 is kinda complex and tedious (although possible) so we will demonstrate a simpler approach.

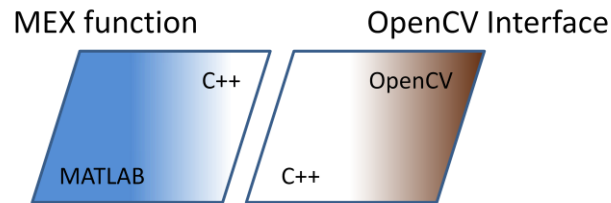
In order to compile all the files in VS2005 we will perform No. 1, But MATLAB will only compile source files with MATLAB and standard C++ function (no OpenCV). MATLAB **will** though, link with OpenCV libraries and MATLAB libraries.

You will achieve that by decoupling the OpenCV C++ function from the MATLAB C++ function. The MEX function will "speak" MATLAB to get and return parameters from MATLAB environment but will use **pure** c++ to call what we name the "OpenCV interface function". This interface function will use **pure** c++ to get and return parameters from the MEX function but will "speak" OpenCV to call the OpenCV implementation.

The MEX function will be contained solely in its CPP source file. (important also since the function name in MATLAB comes from the CPP filename)

The OpenCV interface function can be included in the other OpenCV source files.

Since the signatures of the OpenCV interface is only written in c++ one H include file would contain their declarations. The MEX function CPP file will #include this file for proper compilation.



SETTING VS2005 TO COMPILE & LINK MATLAB MEX FILES

Edit the project properties:

- Add under C/C++ - General section these additional include directories:
 - <MATLAB_DIR> \extern\include
 - <MATLAB_DIR>\simulink\include
- Add under Linker – General section these additional libraries directory:
 - <MATLAB_DIR>\extern\lib\win32\microsoft
- Add under Linker – Input section these additional dependencies:
 - libmx.lib
 - libmex.lib
 - libmat.lib

You can now #include the "mex.h" file and compile mx* functions.

SETTING MATLAB TO COMPILE C++ AND MATLAB MEX FILES AND LINK THEM WITH OPENCV OBJ FILES

First, use "mex –setup" command to setup mex for VS2005 as described in ordinary documentation.

Edit the mexopts.bat and add the following line:

```
set OPENCVDIR=C:\Program Files\OpenCV
```

Edit the "set" command for LIB variable to also include:

- %OPENCVDIR%\lib
- %OPENCVDIR%\bin

Edit the "set" command for INCLUDE variable to also include:

- %OPENCVDIR%\cxcore\include

- %OPENCVDIR%\cv\include
- %OPENCVDIR%\cvaux\include
- %OPENCVDIR%\otherlibs\highgui

Edit the "set" command for the variable LINKFLAGS to include also the following LIB files:

- cv.lib
- highgui.lib
- cvaux.lib
- cxcore.lib

Just add the names after the other LIB files written on that line.

We assume here you use the ordinary OpenCV libraries. You can add more if you use others.

Note: On my platform the mexopts.bat file is in

<doc&settings>\<username>\Application Data\MathWorks\MATLAB\R2007b\

COMPILING & RUNNING

You would need the OpenCV DLLs registered in order to run MEX files that reference OpenCV compiled OBJ files.

In VS2005 you can only include one MEX CPP file at a time in one project since the "mexFunction" can be linked once in a linkage process. Two of those CPP files would cause a linker error.

After you compile successfully your MEX CPP (Let's call it mexf.cpp) file and the other OpenCV CPP file (Let's call it opencv.cpp) you will get an OBJ file for the OpenCV CPP - "opencv.obj". You can ignore mexf.obj.

From MATLAB use the command: (you better use full paths here...)

```
mex "mexf.cpp" "opencv.obj"
```

That will compile mexf.cpp using MATLAB includes and link it with opencv.obj. The result would be a mexf.mexw32 file you can call from MATLAB as the function mexf().

Good Luck, We hope this small guide helped you setting up your project! Share it...

Yoad Snapir (yoadsn@gmail.com)

Rachely Esman (rachelyesman@gmail.com)