

OpenARK CMAKE build instructions for Windows:

Prerequisites:

We recommend using Visual Studio 2015 and 64-bit Windows for OpenARK. When given the option please select Visual Studio 2015 (also known as vc14) and x64 for all prerequisite software.

1. Install Visual Studio 2015. Select custom installation and make sure to install the C++ build tools under Programming Languages.
2. Install cmake: <https://cmake.org/download/>
Make sure to select "add cmake to system PATH" either for current user or all users.
3. Install PCL1.8, easiest way is to use the all-in-one installer made by a kind PCL user:
<http://unanancyowen.com/en/pcl18/#Download>
Make sure to select "add PCL to system PATH" either for current user or all users.
OpenNI2 will be installed as part of this installer. You will need to add the OpenNI2 dlls to the system path manually. The default location these are installed to is C:\Program Files\OpenNI2\Tools. See Step 4 for more details on modifying system variables.
4. Install OpenCV3: http://docs.opencv.org/3.0-beta/doc/tutorials/introduction/windows_install/windows_install.html
Follow the "Installation by Using the Pre-built Libraries" instructions EXCEPT for step 5.

For step 5: "Set the OpenCV environment variable and add it to the systems path". Instead of using the command prompt you can edit the environment variables using an interface by going to "Control Panel->System and Security->System->Advanced system settings->Environment Variables".

We will need to modify two system variables in order to use OpenCV. First we want to add the system variable OpenCV_DIR and set it to "<extraction_directory>\opencv\Build" where <extraction_directory> is the directory you extracted OpenCV to.

You will also need to add the bin directory to the PATH variable. Select the variable "Path" and add a line with the value "%OpenCV_DIR%\x64\vc14\bin". This allows your computer to find the proper library files when running programs using OpenCV.

Hit "OK" to close the Environment Variables dialog box. The variables will not be set until you do this. You will also have to reopen any open command prompt windows for the variables to be updated for that window.

SLAM Specific Dependencies:

General CMAKE Build Instructions:

Most of the following files will follow the general CMAKE build steps shown below:

1. make a build directory to store build files:
`mkdir build`
2. Generate a Visual Studio solution:
`cd build`
`cmake -G"Visual Studio 14 2015 Win64" ..`
3. Build and install:
`cmake --build . --config Release --target install`

Install Eigen 3:

1. Download source from <http://eigen.tuxfamily.org/>. Extract zip folder and cd to the extracted directory.
2. Follow CMAKE steps above.

Install Glog:

1. Download source from <https://github.com/google/glog>. Extract zip folder and cd to the extracted directory.
2. Follow CMAKE steps above.

Install SuiteSparse:

1. Download source from <https://github.com/joemenke/suitesparse-metis-for-windows>. Extract zip folder and cd to the extracted directory. (Note this is our forked version that builds with VS2015).
2. Follow CMAKE steps above.
3. You will need to add the "SuiteSparse_DIR" environment variable and set it to "<build dir from step 2>\install"
4. You will also need to add the lapack install location to your Path. Open the Path environment variable and add a new row and set it to "<build dir from step 2>\install\lib64\lapack_blas_windows"

Install Ceres Solver:

1. Download source from <https://github.com/joemenke/ceres-solver>. Extract zip folder and cd to the extracted directory. (Note: this is simply a fork from the original git repository to make install on windows easier)
2. Follow CMAKE steps above.

Install Boost:

1. Download source from (boost location). Extract zip folder and cd to the extracted directory.
2. Set up:
`bootstrap`

3. Install:

```
.\b2 --toolset=msvc-14.0 --build-type=complete --prefix=C:\Boost architecture=x86 address-model=64 install
```

Note: You may get an error about the msvc version not matching. If so you need to modify project-config.jam with “using msvc : 14.0 ;”

4. Boost doesn't officially support Visual Studio 2015 (though it works fine), you need to modify \$(boost install location)/include/boost-1_XX/boost/config/auto_link.hpp and change:

```
#elif defined(BOOST_MSVC)
#define BOOST_LIB_TOOLSET “vc120”
to
#elif defined(BOOST_MSVC)
#define BOOST_LIB_TOOLSET “vc140”
```

Install OpenGV:

1. Download source from <https://github.com/joemenke/opencv>. Extract zip folder and cd to the extracted directory. (Note: this is simply a fork from the original git repository to make install on windows easier)
2. Follow CMAKE steps above.
3. You will need to add the environment variable “OpenGV_DIR” and set it to the install location of OpenGV (probably C:\Program Files\opencv)

Install Brisk:

1. Download source from <https://github.com/joemenke/brisk>. Extract zip folder and cd to the extracted directory. (Note: this is simply the brisk originally included with okvis, modified to make install on windows easier).
2. Follow CMAKE steps above.

Install DLib:

1. Download source from <https://github.com/joemenke/DLib>. Extract zip folder and cd to the extracted directory. (Note: this is simply a fork from the original git repository to make install on windows easier).
2. Follow CMAKE steps above.

Install DBoW2:

1. Download source from <https://github.com/joemenke/DBoW2>. Extract zip folder and cd to the extracted directory. (Note: this is simply a fork from the original git repository to make install on windows easier).
2. Follow CMAKE steps above.
3. You will need to add the environment variable “DBoW2_ROOT_DIR” and set it to the install location of DBoW2 (probably C:\Program Files\DBoW2)

Install Okvis+:

1. Download source from <https://github.com/joemenke/okvis>. Extract zip folder and cd to the extracted directory.

2. Follow CMAKE steps above.

(Optional) Setting up the Intel Realsense SR300:

1. Plug in SR300 Camera and wait for drivers to finish installing.
2. Open Device Manager. Under “imaging devices” you should see three Realsense Camera Devices SR300 Depth, SR300 RGB, and SR300 Virtual Driver. If the virtual driver is not present download and install the SR300 Camera Driver from the intel website: <https://software.intel.com/en-us/intel-realsense-sdk/download>
3. (Important: do NOT do this step until you have completed step 1 and 2) Install the Intel Realsense SDK Essentials from the Intel website: <https://software.intel.com/en-us/intel-realsense-sdk/download>
This can affect driver installation, so it is important that the drivers are installed before installing the SDK.
4. Set the Environment Variable RSSDK_DIR to the Realsense SDK install location. Typically:
C:\Program Files (x86)\Intel\RSSDK

(Optional) Setting up the PMD Pico:

1. You will need to obtain the PMD SDK using your customer account from when you purchased your PMD Device.
2. Set the Environment Variable PMDDIR to location you installed the PMD SDK
3. Add %PMDDIR%/bin to the Path Environment Variable

Building OpenARK:

1. Open a VS2015 x64 Native Tools Command Prompt
2. cd to the directory to which you have downloaded the OpenARK source code
3. make a build directory to store build files:
`mkdir build`
4. Generate a Visual Studio solution:
`cd build`
`cmake -G"Visual Studio 14 2015 Win64" ..`

5. You can now either open the Visual Studio Solution generated in the build directory labeled “OpenARK.sln” or continue to build using the command prompt.

Building via Visual Studio:

Right click on the OpenARK project in the solution explorer and select “Set as Startup Project”. Build and run as usual.

Building via Command Prompt:

```
cmake --build . --config Release
```

To run:

```
cd Release
```

```
OpenARK.exe
```