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 (VC version 14) 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. IMPORTANT: Visual Studio 2017 will not work because it is not currently supported by PCL.

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. The 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 the 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 environment variables.

4. Install OpenCV (v3.2+ required):
   https://docs.opencv.org/3.2.0/d3/d52/tutorial_windows_install.html
   Follow the “Installation by Using the Pre-built Libraries” instructions EXCEPT for step 5.

   For step 5: “Set the OpenCV enviroment 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.

(Optional) Selecting a Depth Camera:

We require that at least one of the following: Intel's new cross-platform RealSense SDK (RSSDK 2), the old RealSense Windows SDK, or the PMD SDK to be installed to compile OpenARK. By default, CMake looks for the SDKs in the above order, and proceeds to build with the first one it finds. It is possible, however, to enable multiple SDKS in the same build. If you wish to a specific set of SDKs, you should set the USE_RSSDK2, USE_RSSDK, USE_PMDSDK flags in CMake.

For instance, to force CMake look all three SDK's, replace:
`cmake -G"Visual Studio 14 2015 Win64" ..`
with:
`cmake -G"Visual Studio 14 2015 Win64" -DUSE_RSSDK2:=ON -DUSE_RSSDK:=ON -DUSE_PMDSDK:=ON`

Specific instructions for setting up each camera may be found below. Build instructions continue on the next page.

(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 already present, download and install the SR300 Camera Driver from the Intel website: https://downloadmirror.intel.com/25044/eng/intel_rs_dcm_sr300_3.3.27.5718.exe

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://registrationcenter.intel.com/en/forms/?productid=2797
You may need to register to 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

Building OpenARK-test:

OpenARK-test is a module for testing the OpenARK finger tracking system. It contains a benchmark dataset and code for evaluating the quality of the OpenARK finger tracking.

To include OpenARK-test in our build we simply need to set the BUILD_TESTS flag in CMake.

To do this, replace:
cmake -G"Visual Studio 14 2015 Win64" ..
with:
cmake -G"Visual Studio 14 2015 Win64" -DBUILD_TESTS:=ON ..

This will automatically grab the test code and database from *Github* and include it as a separate project in our generated Visual Studio solution.