

Orbbec SDK Development Menu

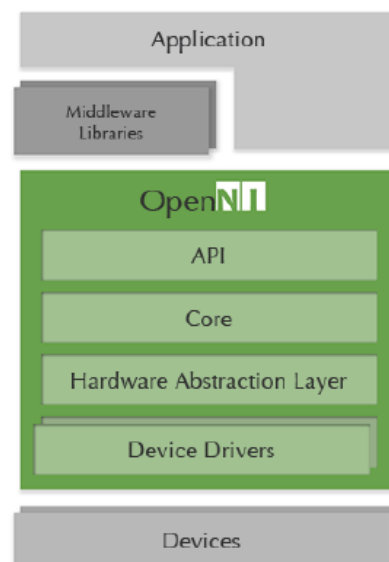
Table of Contents

1. Overview	2
2. Scope	2
3. Windows	2
3.1 Environment Setup	2
3.2 Development Specifications	3
4. Linux.....	4
4.1 Environment Setup	4
4.2 Development Specifications	4
5. OpenNI2 Program Structure	6
6. Open OpenNI2 Debug Log.....	6
7. Windows/Linux Sample Code Instruction.....	7
8. Android.....	8
8.1 Android Studio Environment Set Up.....	8
8.2 Development Instruction.....	12

1. Overview

Orbbec SDK is developed based on OpenNI. OpenNI or Open Natural Interaction is a multi-language and cross-platform software framework. It defines the interface between application, middleware, and 3D sensing device. Orbbec's Astra Series Camera fully support OpenNI protocol. All application that developed by OpenNI can work seamlessly with Astra Devices. OpenNI2 is the second and newest version of OpenNI. When using Orbbec SDK, it is assumed that the reader has basic knowledge about OpenNI2. Please referring to the document "OpenNI2 Coding Instruction" or official OpenNI documents in order to understand more about OpenNI development.

OpenNI SDK 2 Architecture



Primary advantages of OpenNI SDK 2:

- Supports latest generations of 3D Sensors (Short range,...)
- Easier for development: Clean and improved design and API
- Simpler distribution: private copy of OpenNI and NiTE for each application
- More flexible: depth units control, exposure control, etc.
- More open: Middleware API are separate for each library
- Better and simplified multi-sensor support
- Allows event-driven programming
- Better backwards compatibility
- Large offer of third party Middleware Libraries: Body Tracking, 3D Reconstruction, Object Recognition, Analytics, and many more.

2. Scope

Device: Astra Series Camera, including Astra, Astra Pro, Astra Mini, etc.

Platform: Windows 7 and above; Ubuntu 14.04 and above.

3. Windows

3.1 Environment Setup

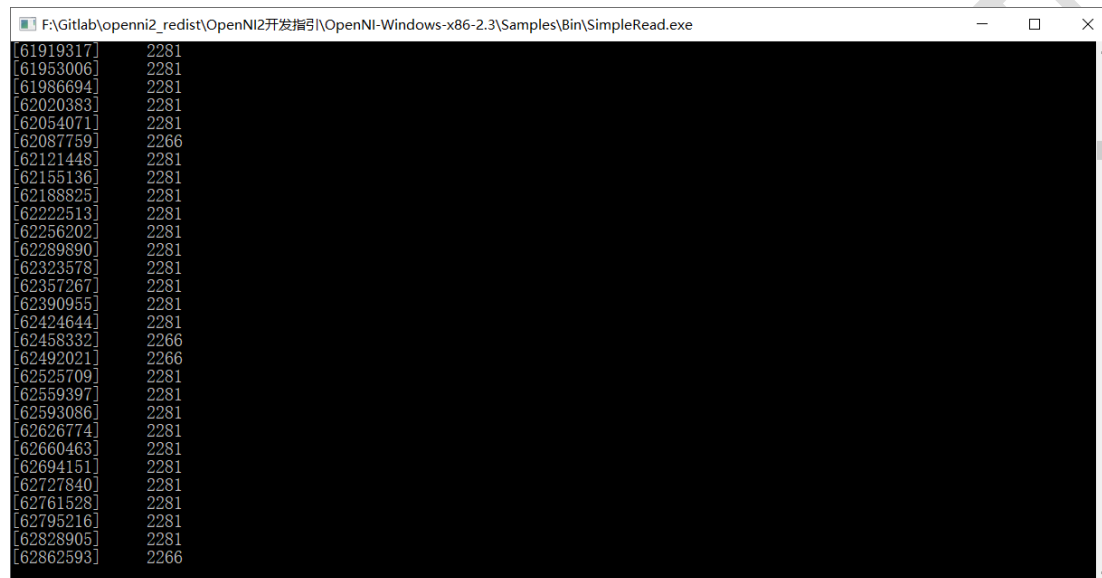
Before developing using OpenNI2 on Windows, please make sure that Astra Driver has be installed, Orbbec's device is showed on Device Manager, and normal Depth map can be seen by using NiViewer tool from OpenNI2. Please refer to document "Astra Driver Installation and Device diagnostics" for more information.

3.2 Development Specifications

What included in Orbbec SDK:

- Documentataion: API document and development manual of OpenNI2;
- Include: Header files needed to develop OpenNI2 programs;
- Lib: Static libraries that need to be linked during developing OpenNI2 program;
- Redist: Dynamic libraries that needed while running OpenNI2 program;
- Samples: OpenNI2 related development routines;

SimpleRead and SimpleViewer are important routines of SDK. SimpleRead is a routine reading depth value of middle point of the image. After connecting camera, running Samples\Bin\SimpleRead.exe. The result will be shown as follows:

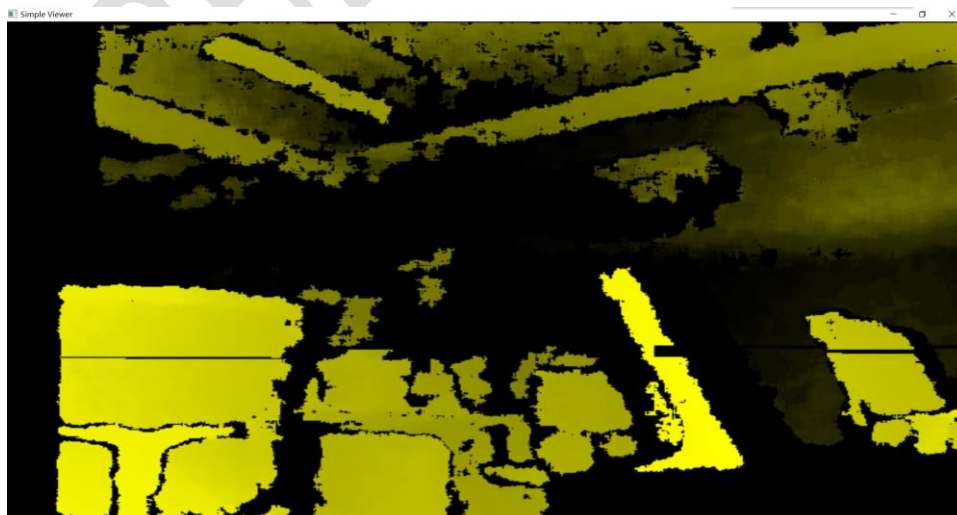


```

F:\Gitlab\openni2_redist\OpenNI2开发指引\OpenNI-Windows-x86-2.3\Samples\Bin\SimpleRead.exe
[61919317] 2281
[61953006] 2281
[61986694] 2281
[62020383] 2281
[62054071] 2281
[62087759] 2266
[62121448] 2281
[62155136] 2281
[62188825] 2281
[62222513] 2281
[62256202] 2281
[62289890] 2281
[62323578] 2281
[62357267] 2281
[62390955] 2281
[62424644] 2281
[62458332] 2266
[62492021] 2266
[62525709] 2281
[62559397] 2281
[62593086] 2281
[62626774] 2281
[62660463] 2281
[62694151] 2281
[62727840] 2281
[62761528] 2281
[62795216] 2281
[62828905] 2281
[62862593] 2266
  
```

Figure. For the specific implementation code, please referring to the project under “Samples\SimpleRead\” directory. Readers can try it by compiling through VS.

SimpleViewer is a routine implementing color and depth map. It reads the data and draw the data through OpenGL. After connecting camera, running Samples\Bin\SimpleViewer.exe. The result will be shown as follows:



Keyboard shortcuts:

- 1 - Display Depth and color overlapped map;
- 2 - Display depth map;
- 3 - Display color map;
- m - Switch mirror display;

For the specific implementation code, please refer to the project under “Samples\SimpleViewer\” directory. Readers can try it by compiling through VS.

4. Linux

4.1 Environment Setup

- 1) Before developing using OpenNI2 SDK on Linux, please execute the following command to install and configure the environment. Example on x64 platform: In order to run properly, install the following program:

```
$ sudo apt-get install build-essential freeglut3 freeglut3-dev
```

- 2) Check the version of udev. Orbbec’s driver depends on “libudev.so.1”, which can be found by Execute the following command. Please manually link libudev.so.x.x to libudev.so.1 if failed to find.

```
$ldconfig -p | grep libudev.so.1
$cd /lib/x86_64-linux-gnu
$sudo ln -s libudev.so.x.x.x libudev.so.1
```

- 3) Copy Linux OpenNI2 SDK Package to a path, decompress, and run the install script in the unzipped directory to generate OpenNI2 related environment variables.

```
$ tar zxvf OpenNI-Linux-x64-2.2-0118.tgz
$ cd OpenNI-Linux-x64-2.2
$ sudo ./install.sh
$ source OpenNIDevEnvironment
```

- 4) Unplug the device, then plug again to work.

4.2 Development Specifications

SDK for Linux platform: OpenNI-Linux-Arm-2.3, OpenNI-Linux-Arm64-2.3, OpenNI-Linux-x86-2.3 and OpenNI-Linux-x64-2. The contents that mainly included:

- Documentataion: API document of OpenNI2;
- Include: Header files for OpenNI2 programs;
- Redist: Library files for compiling OpenNI2 program;

- Samples: OpenNI2 related development routines;
- Tools: NiViewer that provided by OpenNI2.

SimpleRead and SimpleViewer are important routines of SDK. The examples below will using Ubuntu 14.04, x64 platform.

SimpleRead is a routine reading depth value of middle point of the image. After connecting camera, running Samples\Bin\SimpleRead in Shell. The result will be shown as follows:

```
[44502404] 2535
[44536092] 2535
[44569781] 2554
[44603469] 2535
[44637157] 2554
[44670846] 2554
[44704534] 2554
[44738223] 2535
[44771911] 2535
[44805600] 2535
[44839288] 2535
[44872976] 2535
[44906665] 2535
[44940353] 2516
[44974042] 2535
[45007730] 2535
[45041418] 2535
[45075107] 2535
[45108795] 2535
[45142484] 2535
[45176172] 2516
[45209861] 2535
[45243549] 2535
[45277237] 2554
[45310926] 2535
[45344614] 2516
[45378303] 2516
[45411991] 2535
[45445680] 2516
[45479368] 2535
[45513056] 2535
[45546745] 2535
[45580433] 2535
[45614122] 2535
[45647810] 2535
[45681498] 2516
[45715187] 2535
[45748875] 2535
[45782564] 2535
[45816252] 2516
[45849941] 2516
```

Figure. For the specific implementation code, please refer to the source code under “Samples\SimpleRead\” directory. Readers can try it by compiling using make command.

SimpleViewer is a routine implementing color and depth map. It reads the data and draw the data through OpenGL. After connecting camera, running Samples\Bin\SimpleViewer in Shell. The result will be shown as follows:



Keyboard shortcuts:

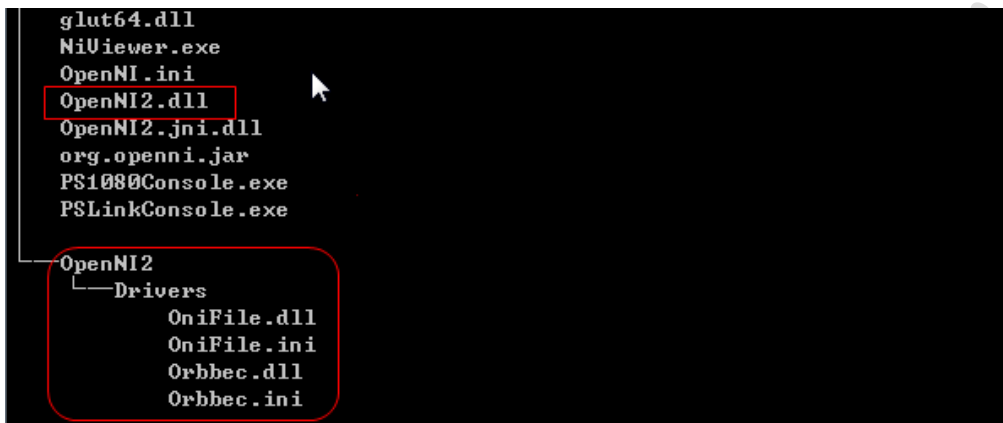
- 1 - Display Depth and color overlapped map;
- 2 - Display depth map;

- 3 - Display color map;
- m - Switching mirror display;

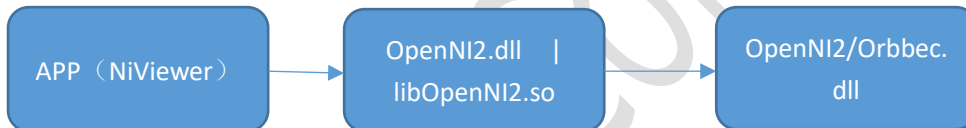
For the specific implementation code, please refer to the project under “Samples\SimpleViewer\” directory. Readers can try it by compiling using make command.

5. OpenNI2 Program Structure

A standard OpenNI program always follows the file directory structure below. It can be changed by modifying OpenNI.ini.



The figure below shows the hierarchical order of calls:



Orbbec.ini is saved as initial configuration of Astra. The configuration includes resolution, output format, mirror, etc. It can be changes by coding or modifying the parameter of the file.

6. Open OpenNI2 Debug Log

OpenNI2 offers many Debug Log. Open Log in corresponding level to view information including depth map frame, color map frame, etc. This function is mainly implemented by modifying OpenNI.ini file.

```

[Log]
; 0 - Verbose; 1 - Info; 2 - Warning; 3 - Error. Default - None
; Verbosity=0 // Remove ; , change Verbosity=0, for detailed log information
; LogToConsole=1 // Remove ; ,change LogToConsole=1, for logging the info to console
; LogToFile=1 // Remove ; ,change LogToFile=1, for logging to file
; LogToAndroidLog=1 // Remove ; ,change LogToAndroidLog=1, output to logcat

[Device]
; Override=
; RecordTo=
  
```

[Drivers]

; Location of the drivers, relative to OpenNI shared library location. When not provided, "OpenNI2/Drivers" will be used.

;Repository=OpenNI2/Drivers

; List of drivers to load, separated by commas. When not provided, OpenNI will try to load each shared library in Repository.

;List=

7. Windows/Linux Sample Code Instruction.

Orbbec SDK provides a series of sample code based on OpenNI2 in order to help developers to better understand OpenNI related development including turn on/off device, obtain depth map data, obtain color map data, etc.

Two method can be used to obtain frame data from video stream: callback and acquirement. Using acquirement method needs to manually control the data in the loop. Callback method is to let OpenNI register relevant event, and the new coming data will trigger the event. The callback method is recommended. The code below uses acquirement method unless otherwise specified.

SimpleRead – A basic code in OpenNI2, please view the code before using. This program is a console program. The main function is to open the device, create a depth stream, get the depth frame data, and print the depth value of each frame center.

MultipleStreamRead – Console program. Similar function as SimpleRead. However, it adds the function of creating color stream and obtaining both color and depth stream data.

MultipleStreamRead-Callback – Console program. Similar function as MultipleStreamRead. However, the method of obtaining color/depth frame data is callback method. It also shows the method of frame synchronization and adding device to connect listener.

SimpleViewer – It demonstrates the method to compile an OpenGL graphic program to open device, create depth/color stream, and show both color and depth map. It also displays the histogram of the depth data and displays it graphically.

MultiDepthViewer – Shows the method to compile a program to display multiple RGBD data under one system, and switch device by number keypad to view the depth map of corresponding device.

EventBasedRead – Similar function as SimpleRead. However, it the method to obtain frame data is callback.

MWClosestPoint – Shows the method to compile a middleware library file based on OpenNI. The middleware can be used to analyze frame data and find the closest point to the camera.

MWClosestPointApp – Shows the method to compile a console program to print the closest point by using library generated by MWClosestPoint.

ClosestPointViewer – Shows the method to compile a GL figure program to show the closest point by using library generated by MWClosestPoint.

Simpleviewer.Java – Shows the method to use Java to compile a figure program, and display both color and depth stream.

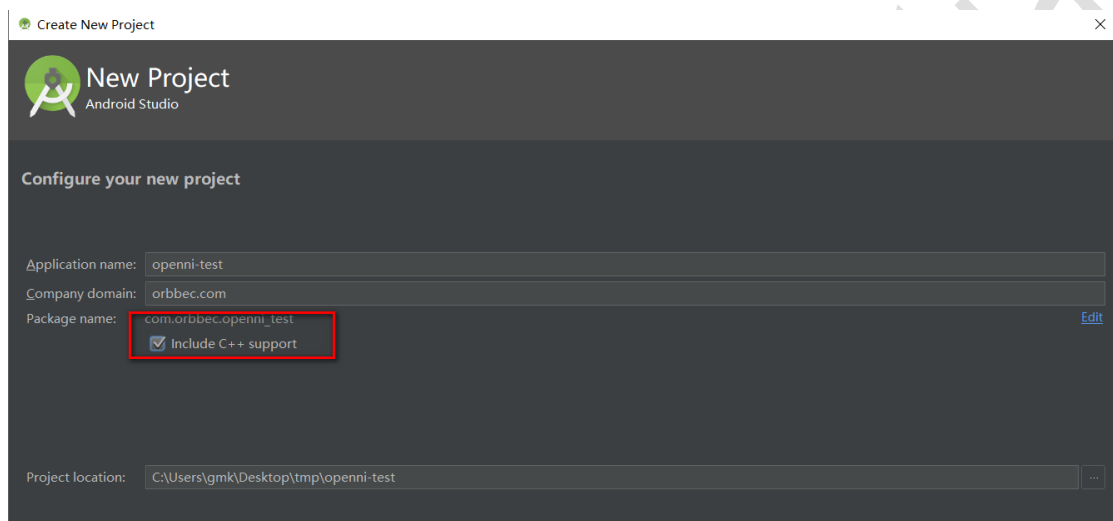
ExtendedAPI – Extend function of **orbbecc sdk**. Shows the method to acquire device serial number, acquire device type, read save calibration parameter (Astra S only), save parameter to camera, get IR gain, set IR gain, get/set IR exposure, turn on/off LDP, and turn on/off laser.

8. Android

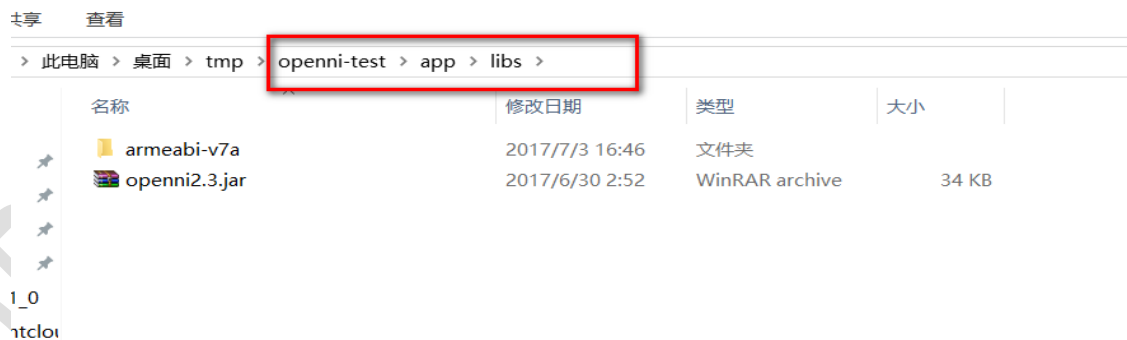
8.1 Android Studio Environment Set Up

Please check the following procedure in order to integrate OpenNI2 SDK into an Android project.

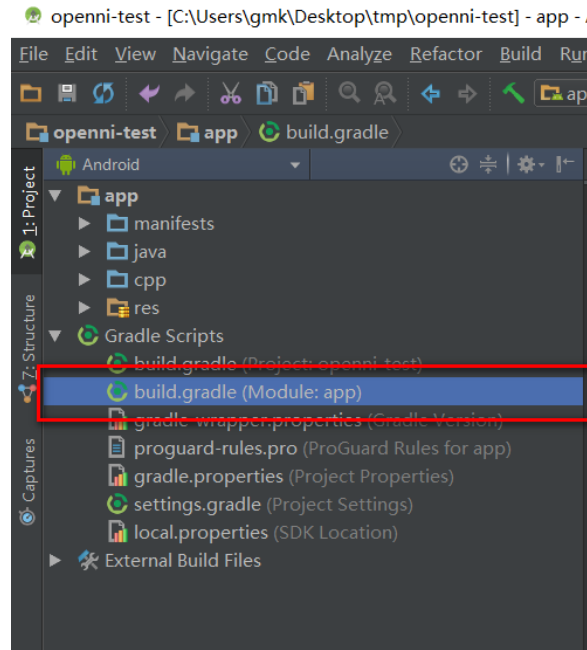
- In Android Studio. create a new project **openni-test**, which supports **c++** project.



- Copy the contents in **OpenNI_Android_32bit_2.3** -> **libs** to the directory **openni-test/app/libs**



- Project configuration in Gradle file of **openni-test** project and Jar configuration.



- Add following content(figure) to Gradle file, and add abiFilters of ndk to defaultConfig:

```

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.3"
    defaultConfig {
        applicationId "com.orbec.openni_test"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
        externalNativeBuild {
            cmake {
                cppFlags "-frtti -fexceptions"
            }
        }
        ndk {
            abiFilters 'armeabi-v7a'
        }
    }
}

```

- After buildTypes, add sourceSets to configure library file directory:

```
externalNativeBuild {
    cmake {
        cppFlags "-fexceptions"
    }
}

ndk {
    abiFilters 'armeabi-v7a'
}

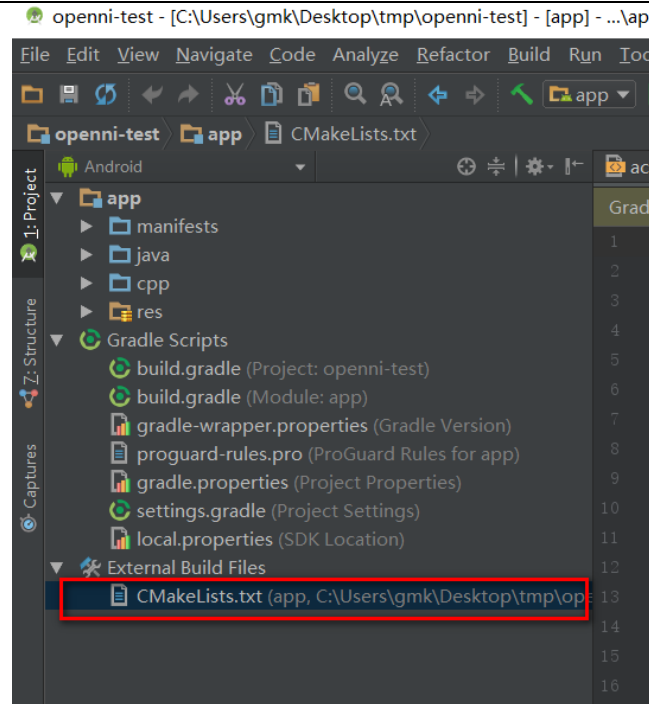
buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}

sourceSets {
    main {
        jniLibs.srcDirs = ['libs']
    }
}
```

- Add relevant openni package directory to dependencies in order to help with compiling.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile files('libs/openni2.3.jar')
}
```

- Do cpp configuration in cmake file of openi-test:



- For self-created app file, please refer to depthforopenni2 project configuration. depthforopenni2 project will create DepthUtils.cpp file in cpp directory, and the corresponding cmake is set to:

```
add_library( # Sets the name of the library.
    DepthUtils

    # Sets the library as a shared library.
    SHARED

    # Provides a relative path to your source file(s).
    # Associated headers in the same location as their source
    # file are automatically included.
    src/main/cpp/DepthUtils.cpp )
```

```
# Specifies libraries CMake should link to your target library. You
# can link multiple libraries, such as libraries you define in the
# build script, prebuilt third-party libraries, or system libraries.

target_link_libraries( # Specifies the target library.
    DepthUtils

    # Links the target library to the log library
    # included in the NDK.
    ${log-lib} )
```

- Copy asserts in OpenNI_Android_32bit_2.3 package to openni-test/app/src/main:

电脑 > 桌面 > tmp > openni-test > app > src > main				
名称	修改日期	类型	大小	
assets	2017/7/3 17:11	文件夹		
cpp	2017/7/3 16:45	文件夹		
java	2017/7/3 16:44	文件夹		
res	2017/7/3 16:44	文件夹		
AndroidManifest.xml	2017/7/3 16:45	XML 文档	1 KB	

After completing the above steps, developers can read and display the color/depth /IR map in the new project following the code in Demos directory of the OpenNI_Android_32bit_2.3 package.

8.2 Development Instruction

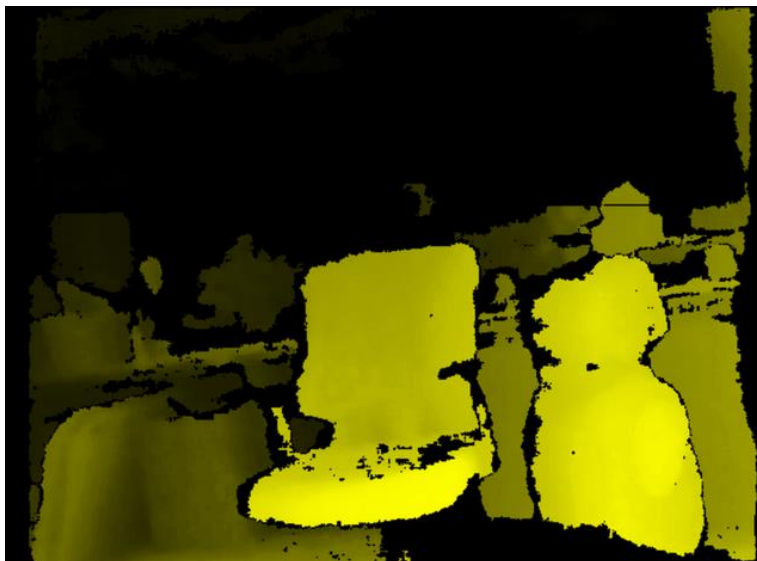
Two packages are provided for SDK in Android platform: OpenNI_Android_32bit_2.3 and OpenNI_Android_Native_2.3. OpenNI_Android_32bit_2.3 is Java level SDK, including:

- Demos: Routine source code for viewing color/depth/IR map;
- Apks: Compiled APK file to view color, depth and IR map;
- Asserts: OpenNI2 related configuration files;
- Libs: jar package and so library for Android OpenNI2;

OpenNI_Android_Native_2.3 is Native level SDK, including:

- Demos : Routine source code for viewing color/depth/IR map;
- Apks: Compiled APK file to view color, depth and IR map;
- Asserts: OpenNI2 related configuration files;
- Libs: So library for Android OpenNI2;

depthforopenni2 is an important routine in Java SDK. It is used to read and show depth map. After connecting camera, open program, the result is in the figure below. Please remember that the Android system has root access and SeLinux have been set to Permissive mode in order to get system USB permission, and obtain color/depth/IR data.



Please refer to project depthforopenni2 in Demos for source code, which can be used by compiled in Android Studio. When running the Android related program, plug in the camera and open the APK program, a USB authorization popup window will appear. If the popup window does not appear and the USB permission is not obtained, the related data cannot be read. In this case, USB authorization related content needs to be checked.