



Users'

Guide

Revision Sheet

Release No.	Date	Revision Description	Author
Rev 1.0	12/03/2018	User's Guide	FAE Team

User's Manual

Authorization Memorandum

I have carefully assessed the User's Guide for Orbbec's 3D senspr. This document has been completed in accordance with the requirements of System Development Methodology.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

_____ The document is accepted.

_____ The document is accepted pending the changes noted.

_____ The document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

NAME

Project Leader

DATE

NAME

Operations Division Director

DATE

NAME

Program Area/Sponsor Representative

DATE

NAME

Program Area/Sponsor Director

DATE

目录

1. 公司与产品简介	1-1
1.1 奥比中光简介	1-1
1.1.1 3D Sensor	1-1
1.1.2 3D与2D视觉对比	1-2
1.1.3 不同3D Sensing技术方案优劣	1-3
1.2 Orbbec产品线与开发资源	1-4
1.2.1 Orbbec产品线	1-4
1.2.2 系统要求与开发资源	1-5
1.3 Orbbec产品概览	1-1
1.3.1 产品命名规则	1-1
1.3.2 产品PID分配表	1-1
1.3.3 Orbbec产品参数	1-2
1.4 Orbbec产品Sensor类型对比	1-3
1.5 3D Sensor常见名词解释	1-1
1.5.1 深度Depth	1-1
1.5.2 深度图	1-1
1.5.3 常用术语	1-2
2. 产品集成设计	2-1
2.1 硬件集成设计	2-1
2.1.1 产品物理接口	2-1
2.2 结构集成设计	2-2
2.2.1 结构密封设计参考	2-2
2.2.2 绝对距离参考点	2-2
2.2.1 散热设计参考	2-3
2.3 驱动安装与系统适配	2-3
2.3.1 Windows系统驱动安装与诊断	2-3
1) 系统要求	2-3
2) 驱动导入	2-3
2.3.2 Android系统适配	2-4
1) 系统要求	2-4
2) 测试方法	2-4
3) 异常处理	2-4
3. SDK介绍与使用	3-1
3.1 SDK简介	3-1
3.2 OpenNI2.3.0.50 SDK目录结构	3-2
3.2 SDK 工程环境及配置	3-2

3.2.1	Windows工程环境配置	3-2
(1)	适用范围	3-2
(2)	开发平台搭建.....	3-2
(3)	SimpleCode说明	3-4
3.2.2	Linux环境配置	3-5
(1)	适用范围	3-5
(2)	Linux环境配置说明	3-5
(3)	udev 说明	3-6
(4)	SimpleCode说明	3-6
3.2.3	Android工程环境配置	3-6
(1)	适用范围	3-6
(2)	Android环境配置说明	3-6
(3)	SimpleCode说明	3-8
3.2.4	.ini配置文件使用说明	3-11
(1)	OpenNI.ini配置说明	3-11
(2)	Orbbec.ini配置说明	3-11
4.	常用API介绍	4-1
4.1	标准API介绍	4-1
4.1.1	OpenNI类.....	4-2
1)	简介	4-2
2)	设备的基础访问.....	4-2
3)	视频流（Video Streams）的基础访问	4-2
4)	设备的事件驱动访问	4-2
5)	错误信息	4-3
6)	版本信息	4-3
4.1.2	Device类	4-3
1)	简介	4-3
2)	设备连接条件	4-3
3)	基本操作	4-3
4.1.3	VideoStream类.....	4-5
1)	简介	4-5
2)	视频流的基础功能.....	4-5
3)	获取关于视频流的信息	4-5
4)	配置视频流	4-6
4.1.4	VideoFrameRef类	4-7
1)	简介	4-7
2)	访问帧数据	4-7

3) 数据裁剪 (Cropping data)	4-7
4) 时间戳(Timestamp).....	4-7
5) 帧索引 (FrameIndex)	4-7
5) 视频模式 (Video Mode)	4-7
6) 数组跨度 (Array Stride)	4-7
4.1.5 其他支撑类.....	4-8
1) 简介.....	4-8
2) 传感器配置类	4-8
3) 传感器信息 (SensorInfo)	4-8
4) 视频模式 (VideoMode)	4-8
5) 摄像头设置 (CameraSetting)	4-8
4.2 扩展API介绍	4-8
4.2.1 获取设备型号.....	4-8
4.2.2 获取设备序列号	4-8
4.2.3 设置LDP开关	4-9
4.2.4 设置LDM开关.....	4-9
4.2.5 设置和获取IR曝光值	4-9
4.2.6 设置及获取IR增益	4-9
4.2.7 设置保存和获取相机标定参数	4-9

1.0 公司与产品简介

1. 公司与产品简介

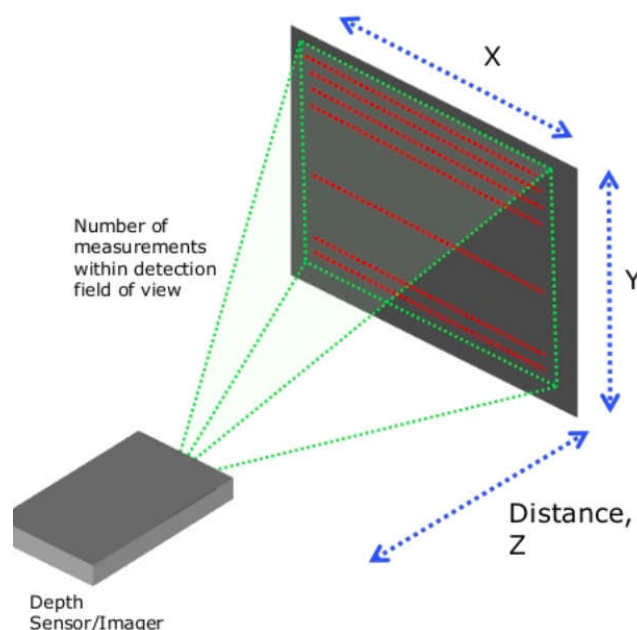
1.1 奥比中光简介

奥比中光是一家集研发、生产、销售为一体的3D视觉传感技术方案提供商，是全球第四家能够量产3D Sensor的公司。主要面向人脸识别、手势识别、骨架识别、三维测量、环境感知和三维地图重建等技术领域提供硬件层（物理层）的支持，对于应用层一整套解决方案，公司整合了合作伙伴的资源，为客户提供支持。

ORBEC®Astra系列、P系列、Deeeye系列以及苍龙深度相机利用结构光3D成像技术获取物体的深度图像，同时利用彩色相机采集物体的彩色图像。其中Astra系列、P系列、Deeeye系列基于单目结构光3D成像技术，苍龙系列基于双目结构光3D成像技术。支持跨平台的开发工具包SDK。

1.1.1 3D Sensor

3D摄像头特点在于除了能够获取平面图像以外，还可以获得拍摄对象的深度信息（Z轴），即位置和距离信息。3D摄像头实时获取环境物体深度信息、三维尺寸以及空间信息，为动作捕捉、三维建模、VR/AR、室内导航与定位等“痛点型”应用场景提供了基础的技术支持。



3D Sensor输出的并不是3D模型，而是一张RGB图，加上一张深度图像。

1.1.2 3D与2D视觉对比

➤ 更多的特征信息和更高的识别准确率

3D摄像头技术进行拍摄时采集得到人脸图像深度信息，能够获取更多的特征信息在传统人脸识别技术基础上大幅提升识别准确率。与2D人脸识别系统相比，3D人脸识别能够采集眼角距、鼻尖点、鼻翼点、两个太阳穴之间的距离、从耳到眼的距离等深度特征信息，并且这些参数一般不会随着一个人整容、换发型而发生较大变化，因而3D人脸识别能够在用户特征发生变化时继续保持极高的识别准确率。

	3D	2D
FRA (错误接收率: 系统接受错误对象概率)	0.047%	0.120%
FRR (错误拒绝率: 系统拒绝真实对象概率)	0.103%	9.790%
姿态变化后识别率	100.00%	23.00%
头发遮挡后识别率	87.00%	50.00%
头部遮挡 (帽子、头盔) 后之识别率	95.00%	<5.00%
弱光环境识别率	100%	0.00%

资料来源: 新加坡出入境管理局、安信证券研究中心

➤ 更加准确的物体分割与更高的安全性

2D图像在复杂场景下由于缺少深度的信息无法正确分割物体;

2D图像也可以做人脸识别，但是尚未有用2D摄像头做刷脸支付的，因为它的安全性不够，其实也就是说平面的信息采集会有比较大的安全风险，可能会被照片攻破。

➤ 手势识别: 全新的交互方式

手势识别的关键便在于3D感知技术，3D摄像头获取影像深度信息，通过算法处理对用户手势进行识别，从而实现用户隔空操控智能终端，将用户从触碰屏幕解放出来，主动捕捉用户手势动作并进行识别处理将成为下一个交互痛点！

➤ 三维数据的模型重建

利用3D摄像头可以建立周围环境的“点云”，点云数据结合环境图像的RGB信息便可以进行场景还原，此后可以在此基础上衍生出多重应用如测距、虚拟购物、装修等等，比如虚拟的家具摆放，由于还原的场景具有深度信息，因此模拟出来的家具在碰到障碍物时便不能继续推动，具有超强真实感。

1.1.3 不同3D Sensing技术方案优劣

➤ 3D 结构光方案

结构光顾名思义就是有特殊结构的光，比如离散光斑、条纹光、编码结构光等。将这样的一维或二维的图像投影至被测物上，根据图像的大小畸变，就能判断被测物的表面形状即深度信息。举个例子，拿一个手电照射墙壁，站近或站远，墙上的光斑是不同大小的，从不同角度照射墙，光斑也会呈现不同的椭圆，这就是结构光的基础。

因为衍射光斑在一定距离外能量密度会降低，所以不适用于远距离的深度信息采集。

➤ TOF方案

TOF发射的不是散斑，而是面光源，所以在一定距离内不会大量衰减，随距离增加深度精度不会显著下降。但相应功耗更高，且目前深度图像分辨率低。

➤ 双目立体视觉

双目立体视觉使用两个或者两个以上的摄像头同时采集图像，通过比对这些不同摄像头在同一时刻获得的图像的差别，使用算法来计算深度信息，从而多角三维成像。

方案	双目	结构光	TOF
基础原理	双目匹配 视差算法	激光散斑编码	飞行时间（相位差）
激光光源	无（被动式）	散斑激光	均匀面激光
功耗	低	中	高
图像分辨率	中高	中	低
精度	低	中	中高
帧率	低	中	高
低光照表现	弱	良好	良好
强光照表现	良好	弱	中等
优点	技术成熟，平面信息分辨率高， 功耗较低	抗干扰好，识别距离 远	强光抗干扰好，功耗 高
突出缺点	算法复杂度高，实时性差； 昏暗环境中不适用； 目标特征纹理变化不明显场景无法使用。	强光条件下会受光照 影响	深度图像分辨率低；

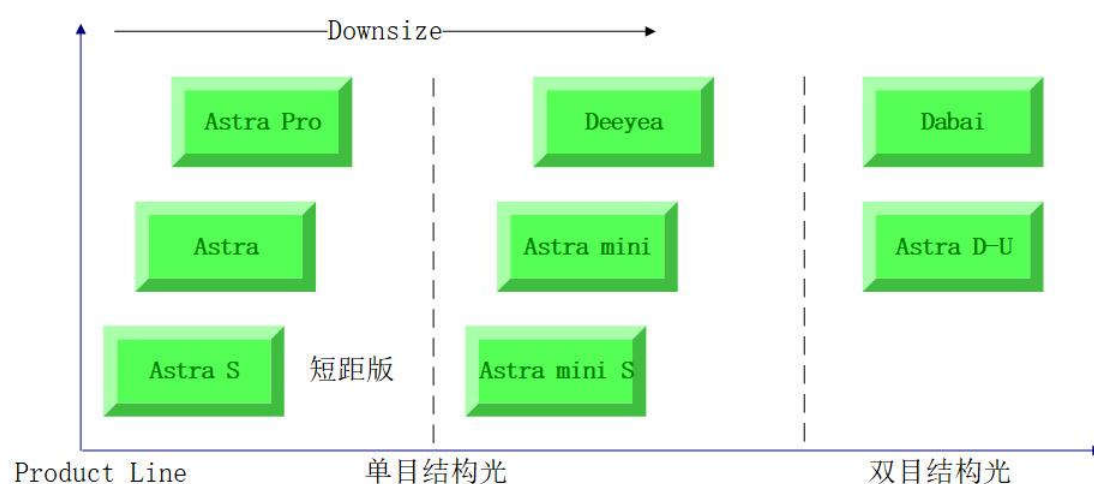
简单点来说，双目具有良好的户外表现，但不适用纹理变化不明显的场合，如白墙；

TOF在远距离的精度更佳，但目前受限于图像分辨率低、功耗大；

结构光综合表现目前来看，在三种3D传感技术中最优，但是户外强光干扰明显和远距离处精度差。所以，奥比中光推出了940nm波段的二代产品P1、Deeyea和双目结构光结合的产品苍龙一号，尽可能改善结构光的固有缺陷。

1.2 Orbbec产品线与开发资源

1.2.1 Orbbec产品线



- ✧ 通用标品中，以Astra为标准产品，后缀mini表示尺寸小型化；后缀S为short简称，表示深度检测范围更短；后缀pro，表示RGB升级版，支持更高分辨率的彩色图像，且彩色数据流通过UVC协议传输，与其他产品不同。
- ✧ Deeyea为我司最新一代单目标品
- ✧ Astra D-U和Dabai为最新的双目结构光产品，拥有最好的户外表现性能

1.2.2 系统要求与开发资源

系统要求 Android

- Android OS 4.4/ 5.1/6.0/7.1
- USB 2.0 或 3.0 (支持 host 接口)
- 建议 RK3288 (四核 Cortex-A17 , 主频 1.8GHz) 或以上
- 建议 2GB RAM 或以上
- 支持 LibUSB+LibUVC
- 支持 UVC 设备
- 支持 SELinux 权限访问 UVC 设备

系统要求 Windows

- Windows 7,10, 32-bit and 64-bit
- USB 2.0 或 3.0
- 双核 , 主频 2.2+GHz 或以上
- 建议 4GB RAM 或以上

奥比中光为客户提供Windows/Android/Linux/Unity/ROS系统下的开发代码包，支持C++/Java语言开发。SDK包的获取需要联系工作人员。

Note:

奥比中光中英文官网上提供的Astra SDK2.0，通过该渠道获取的SDK仅供手势和骨架开发学习使用；

1.3 Orbbec产品概览

1.3.1 产品命名规则

对于Astra、Astra S、Astra mini、Astra mini S和Astra Pro五款产品，以字母+序列号的方式命名。不同字母与产品的对照如下。

产品类型	标识字母
Astra	A
Astra S	B
Astra Pro	C
Astra mini	D
Astra mini S	E

通过查看设备SN号，可以直观判断产品类型、芯片代次和生产日期，具体如下。

E1804300018					
E	18	4	30	3	0018
mini s	年	月	日	二代芯片	序列号
备注：2代表一代芯片					

1.3.2 产品PID分配表

VID: 0x2BC5				
序号	型号	PID	UVC	备注
1	Astra	0x0401		单目结构光
2	Astra S	0x0402		单目结构光
3	Astra Pro	0x0403	0x0501	单目结构光
4	Astra mini	0x0404		单目结构光
5	Astra mini S	0x0407		单目结构光
6	Astra D-U	0x0608	0x0508	双目结构光
7	Deeyea	0x060b	0x050b	单目结构光
8	Astra Pro Plus	0x060d	0x050d	单目结构光
9	Dabai	0x060e	0x050e	双目结构光

Linux下可以通过lsusb查看是否有对应设备连接。

1.3.3 Orbbec产品参数

Orbbec产品概览								
项目	Astra	Astra S	Astra Pro	Astra mini	Astra mini S	Deeyae	Dabai	Astra D-U
Orbbec ASIC	✓	✓	✓	✓	✓	✓	✓	✓
独立ISP	×	×	✓	×	×	✓	✓	✓
接近传感器	HW	HW	HW	×	×	SW	×	×
外壳	✓	✓	✓	×	×	✓	×	×
麦克风	✓	✓	✓	×	×	×	×	×
深度范围（米）	0.6-8	0.4-2	0.6-8	0.6-8	0.4-2	0.25-1.5	0.3-3	0.3-3
Baseline	75mm	75mm	75mm	55mm	55mm	40mm	40mm	40mm
平均功耗（W）	<2.4	<2.4	<2.5	<2.4	<2.4	<2.5	1.6W	2.36W
彩色支持UVC	×	×	✓	×	×	✓	✓	✓
精度	±3mm@1m	±3mm@1m	±3mm@1m	±3mm@1m	±3mm@1m	±3mm@1m	TBD	±5mm@1m
深度分辨率	1280x1024@7fps 640x480@30fps 320x240@30fps 160x120@30fps	1280x1024@7fps 640x480@30fps 320x240@30fps 160x120@30fps	1280x1024@7fps 640x480@30fps 320x240@30fps 160x120@30fps	1280x1024@7fps 640x480@30fps 320x240@30fps 160x120@30fps	1280x1024@7fps 640x480@30fps 320x240@30fps 160x120@30fps	1280x800@30FPS 640x400@30FPS	640*400@30FPS 320*200@30FPS	640*400@30FPS 320*200@30FPS
深度FOV	H 58.4° V 45.5°	H 58.4° V 45.5°	H 58.4° V 45.5°	H 58.4° V 45.5°	H 58.4° V 45.5°	H 67.9° V 45.3°	H 67.9° V 45.3°	H 67.9° V 45.3°
彩色分辨率	1280x960@7fps 640x480@30fps	1280x960@7fps 640x480@30fps	1280x720@30fps 640x480@30fps 320x240@30fps	1280x960@7fps 640x480@30fps 320x240@30fps	1280x960@7fps 640x480@30fps 320x240@30fps	1920x1080@30FPS 1280x720@30FPS 640x480@30FPS	1920*1080@30FPS 1280*720@30FPS 640*480@30FPS	1920*1080@30FPS 1280*720@30FPS 640*480@30FPS
彩色FOV	H 63.1° V 49.4°	H 63.1° V 49.4°	H 66.1° V 40.2°	H 63.1° V 49.4°	H 63.1° V 49.4°	H 71.5° V 56.7°	H 71° V 43.7°	H 59.9° V 46.5°
接口方式	标准USB2.0	标准USB2.0	标准USB2.0	标准USB2.0	标准USB2.0	USB3.0 Type-C	USB 物理接口非标	USB 物理接口非标
结构尺寸（mm）	164.85*30*48.25	164.85*30*48.25	164.85*30*48.25	80*20*19.72	80*20*19.72	68.6*22.3*14.8	59.6*17.4*11.1	59.6*17.4*10.4
适用场景	室内	室内	室内	室内	室内	室内/室外 (50000Lux@1m)	室内/室外 (50000Lux@1m)	室内/室外 (50000Lux@1m)
备注	单目结构光	单目结构光	单目结构光	单目结构光	单目结构光	单目结构光	双目结构光	双目结构光

1.4 Orbbec产品Sensor类型对比

应用产品	Astra/Astra S/Astra mini/Astra mini S																																																																																											
Sensor	MT9M001		MT9M114																																																																																									
Spec	<table><tr><th colspan="2">Parameter</th><th>Typical Value</th></tr><tr><td colspan="2">Optical format</td><td>1/2-inch (5:4)</td></tr><tr><td colspan="2">Active imager size</td><td>6.66mm(H) x 5.32mm(V)</td></tr><tr><td colspan="2">Active pixels</td><td>1,280H x 1,024V</td></tr><tr><td colspan="2">Pixel size</td><td>5.2μm x 5.2μm</td></tr><tr><td colspan="2">Shutter type</td><td>Electronic rolling shutter (ERS)</td></tr><tr><td colspan="2">Maximum data rate/ master clock</td><td>48 MP/48 MHz</td></tr><tr><td>Frame rate</td><td>SXGA (1280 x 1024)</td><td>30 fps progressive scan; programmable</td></tr><tr><td colspan="2">ADC resolution</td><td>10-bit, on-chip</td></tr><tr><td colspan="2">Responsivity</td><td>2.1 V/lux-sec</td></tr><tr><td colspan="2">Dynamic range</td><td>68.2dB</td></tr><tr><td colspan="2">SNR_{MAX}</td><td>45dB</td></tr><tr><td colspan="2">Supply voltage</td><td>3.0V–3.6V, 3.3V nominal</td></tr><tr><td colspan="2">Power consumption</td><td>325mW at 3.3V; Standby 275μW</td></tr></table>		Parameter		Typical Value	Optical format		1/2-inch (5:4)	Active imager size		6.66mm(H) x 5.32mm(V)	Active pixels		1,280H x 1,024V	Pixel size		5.2μm x 5.2μm	Shutter type		Electronic rolling shutter (ERS)	Maximum data rate/ master clock		48 MP/48 MHz	Frame rate	SXGA (1280 x 1024)	30 fps progressive scan; programmable	ADC resolution		10-bit, on-chip	Responsivity		2.1 V/lux-sec	Dynamic range		68.2dB	SNR _{MAX}		45dB	Supply voltage		3.0V–3.6V, 3.3V nominal	Power consumption		325mW at 3.3V; Standby 275μW	<table><tr><th>Parameter</th><th>Typical Value</th></tr><tr><td>Optical Format</td><td>1/6-inch</td></tr><tr><td>Active Pixels</td><td>1296 (H) × 976 (V) = 1.26 Mp</td></tr><tr><td>Pixel Size</td><td>1.9 μm × 1.9 μm</td></tr><tr><td>Color Filter Array</td><td>RGB Bayer</td></tr><tr><td>Shutter</td><td>Electronic Rolling Shutter (ERS)</td></tr><tr><td>Input Clock Range</td><td>6–54 MHz</td></tr><tr><td>Output MIPI Data Rate Maximum</td><td>768 Mb/s</td></tr><tr><td>Max. Frame Rate</td><td>30 fps Full Res 36.7 fps 720p 75 fps VGA 120 fps QVGA (Note 2)</td></tr><tr><td>Responsivity</td><td>2.24 V/Lux–sec (550 nm)</td></tr><tr><td>SNR_{MAX}</td><td>37 dB</td></tr><tr><td>Dynamic Range</td><td>70.8 dB</td></tr><tr><td>Supply Voltage</td><td></td></tr><tr><td> Digital</td><td>1.7–1.95 V</td></tr><tr><td> Analog</td><td>2.5–3.1 V</td></tr><tr><td> I/O</td><td>1.7–1.95 V or 2.5–3.1 V</td></tr><tr><td> PLL</td><td>2.5–3.1 V</td></tr><tr><td> PHY</td><td>1.7–1.95 V</td></tr><tr><td>Power Consumption</td><td>135 mW (Note 1)</td></tr><tr><td>Operating Temperature Range (Ambient) – T_A</td><td>–30°C to 70°C</td></tr><tr><td>Chief Ray Angle</td><td>27.7°</td></tr><tr><td>Active Imager Size</td><td>2.46 mm (H) × 1.85 mm (V), 3.08 mm Diagonal</td></tr><tr><td>Package Options</td><td>Bare Die, CSP</td></tr></table>		Parameter	Typical Value	Optical Format	1/6-inch	Active Pixels	1296 (H) × 976 (V) = 1.26 Mp	Pixel Size	1.9 μm × 1.9 μm	Color Filter Array	RGB Bayer	Shutter	Electronic Rolling Shutter (ERS)	Input Clock Range	6–54 MHz	Output MIPI Data Rate Maximum	768 Mb/s	Max. Frame Rate	30 fps Full Res 36.7 fps 720p 75 fps VGA 120 fps QVGA (Note 2)	Responsivity	2.24 V/Lux–sec (550 nm)	SNR _{MAX}	37 dB	Dynamic Range	70.8 dB	Supply Voltage		Digital	1.7–1.95 V	Analog	2.5–3.1 V	I/O	1.7–1.95 V or 2.5–3.1 V	PLL	2.5–3.1 V	PHY	1.7–1.95 V	Power Consumption	135 mW (Note 1)	Operating Temperature Range (Ambient) – T _A	–30°C to 70°C	Chief Ray Angle	27.7°	Active Imager Size	2.46 mm (H) × 1.85 mm (V), 3.08 mm Diagonal	Package Options	Bare Die, CSP
	Parameter		Typical Value																																																																																									
	Optical format		1/2-inch (5:4)																																																																																									
	Active imager size		6.66mm(H) x 5.32mm(V)																																																																																									
	Active pixels		1,280H x 1,024V																																																																																									
	Pixel size		5.2μm x 5.2μm																																																																																									
	Shutter type		Electronic rolling shutter (ERS)																																																																																									
	Maximum data rate/ master clock		48 MP/48 MHz																																																																																									
	Frame rate	SXGA (1280 x 1024)	30 fps progressive scan; programmable																																																																																									
	ADC resolution		10-bit, on-chip																																																																																									
	Responsivity		2.1 V/lux-sec																																																																																									
	Dynamic range		68.2dB																																																																																									
	SNR _{MAX}		45dB																																																																																									
	Supply voltage		3.0V–3.6V, 3.3V nominal																																																																																									
	Power consumption		325mW at 3.3V; Standby 275μW																																																																																									
Parameter	Typical Value																																																																																											
Optical Format	1/6-inch																																																																																											
Active Pixels	1296 (H) × 976 (V) = 1.26 Mp																																																																																											
Pixel Size	1.9 μm × 1.9 μm																																																																																											
Color Filter Array	RGB Bayer																																																																																											
Shutter	Electronic Rolling Shutter (ERS)																																																																																											
Input Clock Range	6–54 MHz																																																																																											
Output MIPI Data Rate Maximum	768 Mb/s																																																																																											
Max. Frame Rate	30 fps Full Res 36.7 fps 720p 75 fps VGA 120 fps QVGA (Note 2)																																																																																											
Responsivity	2.24 V/Lux–sec (550 nm)																																																																																											
SNR _{MAX}	37 dB																																																																																											
Dynamic Range	70.8 dB																																																																																											
Supply Voltage																																																																																												
Digital	1.7–1.95 V																																																																																											
Analog	2.5–3.1 V																																																																																											
I/O	1.7–1.95 V or 2.5–3.1 V																																																																																											
PLL	2.5–3.1 V																																																																																											
PHY	1.7–1.95 V																																																																																											
Power Consumption	135 mW (Note 1)																																																																																											
Operating Temperature Range (Ambient) – T _A	–30°C to 70°C																																																																																											
Chief Ray Angle	27.7°																																																																																											
Active Imager Size	2.46 mm (H) × 1.85 mm (V), 3.08 mm Diagonal																																																																																											
Package Options	Bare Die, CSP																																																																																											
制造商	ON semiconductor		ON semiconductor																																																																																									
Shutter	Rolling shutter		Rolling shutter																																																																																									
Pixels	1-Megapixel		1.26-Megapixel																																																																																									

应用产品	Astra Pro																																																	
Sensor	MT9M001		OV9712																																															
Spec	<table><tr><th colspan="2">Parameter</th><th>Typical Value</th></tr><tr><td colspan="2">Optical format</td><td>1/2-inch (5:4)</td></tr><tr><td colspan="2">Active imager size</td><td>6.66mm(H) x 5.32mm(V)</td></tr><tr><td colspan="2">Active pixels</td><td>1,280H x 1,024V</td></tr><tr><td colspan="2">Pixel size</td><td>5.2μm x 5.2μm</td></tr><tr><td colspan="2">Shutter type</td><td>Electronic rolling shutter (ERS)</td></tr><tr><td colspan="2">Maximum data rate/ master clock</td><td>48 MPS/48 MHz</td></tr><tr><td>Frame rate</td><td>SXGA (1280 x 1024)</td><td>30 fps progressive scan; programmable</td></tr><tr><td colspan="2">ADC resolution</td><td>10-bit, on-chip</td></tr><tr><td colspan="2">Responsivity</td><td>2.1 V/lux-sec</td></tr><tr><td colspan="2">Dynamic range</td><td>68.2dB</td></tr><tr><td colspan="2">SNR_{MAX}</td><td>45dB</td></tr><tr><td colspan="2">Supply voltage</td><td>3.0V–3.6V, 3.3V nominal</td></tr><tr><td colspan="2">Power consumption</td><td>325mW at 3.3V; Standby 275μW</td></tr><tr><td colspan="2">Operating temperature</td><td>0°C to +70°C</td></tr><tr><td colspan="2">Packaging</td><td>48-pin CLCC</td></tr></table>	Parameter		Typical Value	Optical format		1/2-inch (5:4)	Active imager size		6.66mm(H) x 5.32mm(V)	Active pixels		1,280H x 1,024V	Pixel size		5.2μm x 5.2μm	Shutter type		Electronic rolling shutter (ERS)	Maximum data rate/ master clock		48 MPS/48 MHz	Frame rate	SXGA (1280 x 1024)	30 fps progressive scan; programmable	ADC resolution		10-bit, on-chip	Responsivity		2.1 V/lux-sec	Dynamic range		68.2dB	SNR _{MAX}		45dB	Supply voltage		3.0V–3.6V, 3.3V nominal	Power consumption		325mW at 3.3V; Standby 275μW	Operating temperature		0°C to +70°C	Packaging		48-pin CLCC	<ul style="list-style-type: none">■ active array size: 1280 x 800■ power supply:<ul style="list-style-type: none">- core: 1.5 VDC ±5%- analog: 3.0 - 3.6 V- I/O: 1.7 - 3.6 V■ power requirements:<ul style="list-style-type: none">- active: 110 mW- standby: 50 μA■ temperature range:<ul style="list-style-type: none">- operating: -30°C to 70°C- stable image: 0°C to 50°C■ output formats: 10-bit RAW RGB data■ lens size: 1/4"■ lens chief ray angle: 25° non-linear■ input clock frequency: 6 - 27 MHz■ dynamic range: 69 dB■ maximum image transfer rate:<ul style="list-style-type: none">- WXGA (1280x800): 30 fps- HD 720p (1280x720): 30 fps- VGA (640x480): 60 fps■ sensitivity: 3300 mV/(lux-sec)■ S/N ratio: 39 dB■ maximum exposure interval: 826 x t_{row}■ pixel size: 3 μm x 3 μm■ well capacity: 13 Ke⁻■ dark current: 20 mV/s @ 60°C■ image area: 3888 μm x 2430 μm■ package/die dimensions:<ul style="list-style-type: none">- CSP2: 5415 μm x 4415 μm- COB: 5430 μm x 4430 μm
Parameter		Typical Value																																																
Optical format		1/2-inch (5:4)																																																
Active imager size		6.66mm(H) x 5.32mm(V)																																																
Active pixels		1,280H x 1,024V																																																
Pixel size		5.2μm x 5.2μm																																																
Shutter type		Electronic rolling shutter (ERS)																																																
Maximum data rate/ master clock		48 MPS/48 MHz																																																
Frame rate	SXGA (1280 x 1024)	30 fps progressive scan; programmable																																																
ADC resolution		10-bit, on-chip																																																
Responsivity		2.1 V/lux-sec																																																
Dynamic range		68.2dB																																																
SNR _{MAX}		45dB																																																
Supply voltage		3.0V–3.6V, 3.3V nominal																																																
Power consumption		325mW at 3.3V; Standby 275μW																																																
Operating temperature		0°C to +70°C																																																
Packaging		48-pin CLCC																																																
制造商	ON semiconductor		OmniVision																																															
Shutter	Rolling shutter		Rolling shutter																																															
Spec	1-MegaPixel		1-Megapixel																																															

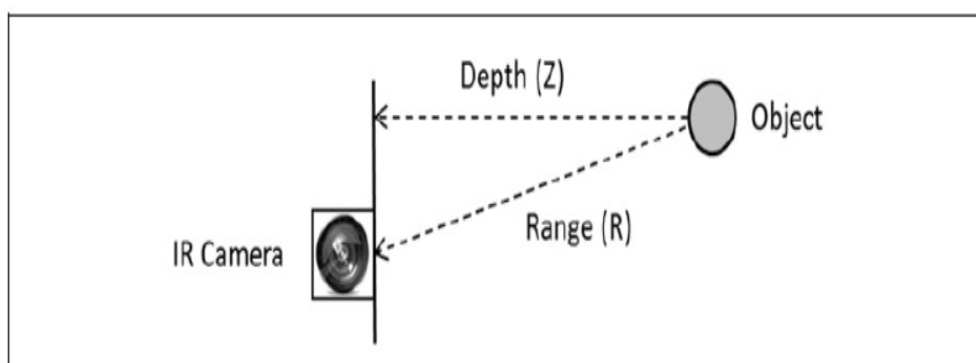
应用产品	Deeyea	
Sensor	OV9282	OV5695
Spec	<ul style="list-style-type: none"> active array size: 1296 x 816 max 5/N ratio: 38 dB power supply: <ul style="list-style-type: none"> core: 1.2V (nominal) analog: 2.8V (nominal) I/O: 1.8V (nominal) dynamic range: 68 dB power requirements: <ul style="list-style-type: none"> active: 156 mW standby: 150 μA XSHUTDOWN: 150 μA maximum image transfer rate: <ul style="list-style-type: none"> - 1280 x 800: 120 fps sensitivity: <ul style="list-style-type: none"> 13000 mV/μW.cm⁻².sec) @ 850 nm 6500 mV/μW.cm⁻².sec) @ 940 nm scan mode: progressive temperature range: <ul style="list-style-type: none"> - operating: -30°C to +85°C junction temperature - stable image: 0°C to +50°C junction temperature minimum exposure time: 1 row period output interfaces: <ul style="list-style-type: none"> 2-lane MIPI serial output and DVP parallel output maximum exposure time: <ul style="list-style-type: none"> frame length - 12 row periods, where frame length is set by registers [0x380E, 0x380F] output formats: 8/10-bit RAW pixel size: 3 μm x 3 μm lens size: 1/4" dark current: 80e⁻/sec @ 50°C junction temperature input clock frequency: 6 - 27 MHz image area: 3896 μm x 2453 μm lens chief ray angle: <ul style="list-style-type: none"> - OV9281: 9° linear - OV9282: 26.78° non-linear package dimensions: <ul style="list-style-type: none"> - OV9281 CSP: 5237 μm x 4463 μm - OV9282 RW: 5252 μm x 4478 μm 	<ul style="list-style-type: none"> active array size: 2592 x 1944 lens size: 1/4" power supply: <ul style="list-style-type: none"> core: 1.14 - 1.26V (1.2V nominal) analog: 2.7 - 3.0V (2.8V nominal) I/O: 1.7 - 1.9V (1.8V nominal) lens chief ray angle: 31.08° non-linear power requirements: <ul style="list-style-type: none"> active: 155 mW standby: 61 μW XSHUTDOWN: 36 μW input clock frequency: 6 - 27 MHz maximum image transfer rate: <ul style="list-style-type: none"> - 5MP (2592x1944): 30 fps - quad HD (2560x1440): 30 fps - 1080p (1920x1080): 60 fps - 720p (1280x720): 60 fps - VGA (640x480): 120 fps temperature range: <ul style="list-style-type: none"> - operating: -30°C to +70°C junction temperature - stable image: -20°C to +60°C junction temperature pixel size: 1.4 μm x 1.4 μm output interface: <ul style="list-style-type: none"> 2-lane MIPI serial output dark current: 15 e⁻/sec @ 60°C junction temperature output formats: 10-bit RGB RAW image area: 3684 μm x 2763 μm dimensions: <ul style="list-style-type: none"> - COB: 5022 μm x 3933 μm - RW: 5072 μm x 3983 μm
制造商	OmniVision	Omni Vision
Shutter	Global shutter	Rolling shutter
Pixel	1-Megapixel	5-Megapixel

应用产品	Astra D-U																									
Sensor	OV9282	AR0330																								
Spec	<div><ul style="list-style-type: none">■ active array size: 1296 x 816■ max 5/N ratio: 38 dB■ power supply:<ul style="list-style-type: none">- core: 1.2V (nominal)- analog: 2.8V (nominal)- I/O: 1.8V (nominal)■ dynamic range: 68 dB■ power requirements:<ul style="list-style-type: none">- active: 156 mW- standby: 150 μA- XSHUTDOWN: 150 μA■ maximum image transfer rate:<ul style="list-style-type: none">- 1280 x 800: 120 fps■ sensitivity:<ul style="list-style-type: none">13000 mV/μW.cm⁻².sec) @ 850 nm6500 mV/μW.cm⁻².sec) @ 940 nm■ scan mode: progressive■ temperature range:<ul style="list-style-type: none">- operating: -30°C to +85°C junction temperature- stable image: 0°C to +50°C junction temperature■ minimum exposure time: 1 row period■ output interfaces:<ul style="list-style-type: none">2-lane MIPI serial output and DVP parallel output■ maximum exposure time:<ul style="list-style-type: none">frame length - 12 row periods, where frame length is set by registers [0x380E, 0x380F]■ output formats: 8/10-bit RAW■ pixel size: 3 μm x 3 μm■ lens size: 1/4"■ dark current: 80e⁻/sec @ 50°C junction temperature■ input clock frequency: 6 - 27 MHz■ image area: 3896 μm x 2453 μm■ lens chief ray angle:<ul style="list-style-type: none">- OV9281: 9° linear- OV9282: 26.78° non-linear■ package dimensions:<ul style="list-style-type: none">- OV9281 CSP: 5237 μm x 4463 μm- OV9282 RW: 5252 μm x 4478 μm</div>	<table><tr><th>Parameter</th><th>Typical Value</th></tr><tr><td>Optical Format</td><td>1/3-inch (6.0 mm) Entire Array: 6.09 mm Still Image: 5.63 mm (4:3) HD Image: 5.82 mm (16:9)</td></tr><tr><td>Active Pixels</td><td>2304 (H) x 1536 (V): (Entire Array) 5.07mm (H) x 3.38mm (V) 2048 (H) x 1536 (V) (4:3, Still Mode) 2304 (H) x 1296 (V) (16:9, HD Mode)</td></tr><tr><td>Pixel Size</td><td>2.2 x 2.2 μm</td></tr><tr><td>Color Filter Array</td><td>RGB Bayer</td></tr><tr><td>Shutter Type</td><td>ERS and GRR</td></tr><tr><td>Input Clock Range</td><td>6-27 MHz</td></tr><tr><td>Output Clock Maximum</td><td>196 Mp/s (4-lane HiSPI or MIPI)</td></tr><tr><td>Output Video - 4-lane HiSPI</td><td>2304 x 1296 at 60 fps < 450 mW (V_{CM} 0.2 V, 198 MP/s) 2304 x 1296 at 30 fps < 300 mW (V_{CM} 0.2 V, 98 MP/s)</td></tr><tr><td>Responsivity</td><td>2.0 V/lux-sec</td></tr><tr><td>SNR_{MAX}</td><td>39 dB</td></tr><tr><td>Dynamic Range</td><td>69.5 dB</td></tr></table>	Parameter	Typical Value	Optical Format	1/3-inch (6.0 mm) Entire Array: 6.09 mm Still Image: 5.63 mm (4:3) HD Image: 5.82 mm (16:9)	Active Pixels	2304 (H) x 1536 (V): (Entire Array) 5.07mm (H) x 3.38mm (V) 2048 (H) x 1536 (V) (4:3, Still Mode) 2304 (H) x 1296 (V) (16:9, HD Mode)	Pixel Size	2.2 x 2.2 μ m	Color Filter Array	RGB Bayer	Shutter Type	ERS and GRR	Input Clock Range	6-27 MHz	Output Clock Maximum	196 Mp/s (4-lane HiSPI or MIPI)	Output Video - 4-lane HiSPI	2304 x 1296 at 60 fps < 450 mW (V _{CM} 0.2 V, 198 MP/s) 2304 x 1296 at 30 fps < 300 mW (V _{CM} 0.2 V, 98 MP/s)	Responsivity	2.0 V/lux-sec	SNR _{MAX}	39 dB	Dynamic Range	69.5 dB
Parameter	Typical Value																									
Optical Format	1/3-inch (6.0 mm) Entire Array: 6.09 mm Still Image: 5.63 mm (4:3) HD Image: 5.82 mm (16:9)																									
Active Pixels	2304 (H) x 1536 (V): (Entire Array) 5.07mm (H) x 3.38mm (V) 2048 (H) x 1536 (V) (4:3, Still Mode) 2304 (H) x 1296 (V) (16:9, HD Mode)																									
Pixel Size	2.2 x 2.2 μ m																									
Color Filter Array	RGB Bayer																									
Shutter Type	ERS and GRR																									
Input Clock Range	6-27 MHz																									
Output Clock Maximum	196 Mp/s (4-lane HiSPI or MIPI)																									
Output Video - 4-lane HiSPI	2304 x 1296 at 60 fps < 450 mW (V _{CM} 0.2 V, 198 MP/s) 2304 x 1296 at 30 fps < 300 mW (V _{CM} 0.2 V, 98 MP/s)																									
Responsivity	2.0 V/lux-sec																									
SNR _{MAX}	39 dB																									
Dynamic Range	69.5 dB																									
制造商	OmniVision	ON semiconductor																								
Shutter	Global shutter	ERS(electronic rolling shutter) GRR(global reset release)																								
Pixel	1-Megapixel	3.15-Megapixel																								

1.5 3D Sensor常见名词解释

1.5.1 深度Depth

3D相机相较于传统2D相机，增加了深度信息。所谓深度值Depth，指的是被测物体到相机平面的垂直距离，即下图中Z值，而非R。



1.5.2 深度图

不同于RGB图像，深度图中每个像素点保存的是视场范围内物体距离相机平面的深度值数据。深度原始数据通常为16-bit unsigned int类型，单位可通过SDK指定，通常为1mm，即深度图中每个像素点保存着16-bit无符号整型数据，单位为1mm。

为了将深度数据可视化的显示出来，通常将其转换为灰度图显示，如下图所示，不同灰度级表示不同的深度值大小。



1.5.3 常用术语

术语	描述
Baseline	1) 单目结构光: 红外相机成像中心与红外投影仪投影中心之间的距离 2) 双目结构光: 左、右红外相机成像中心之间的距离
Depth	深度视频流就像彩色视频流一样, 除了每个像素都有一个值代表距离摄像机的距离 而不是颜色信息
FOV	视场角, 用于描述相机成像给定场景的角度范围, 主要有水平视场角(H FOV)、垂 直视场角(V FOV)和对角线视场角(D FOV)三种
Depth processor	深度计算处理器, 用于实现深度计算算法并输出深度图像的专用ASIC 芯片, 如 MX400、MX6000
IR camera	红外相机, 或红外摄像头
RGB camera	彩色相机, 或彩色摄像头
LDMP/LDM	红外投影仪(IR projector), 也称红外激光投影仪、结构光投影仪等, 用于发射结构 光图案
Depth camera	深度相机, 包含深度成像模组以及彩色成像模组, 其中深度成像模组一般由红外投 影仪、红外相机以及深度云计算处理器组成, 彩色成像模组一般指彩色相机

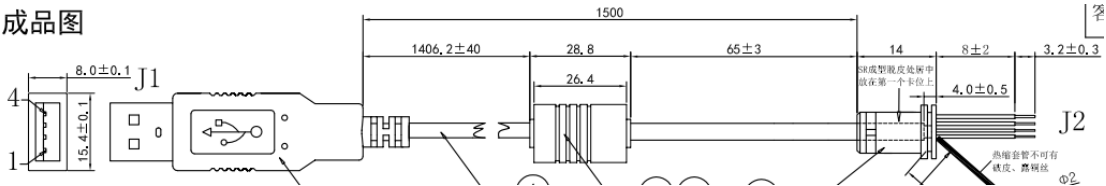
2.0 产品集成设计

2. 产品集成设计

2.1 硬件集成设计

2.1.1 产品物理接口

- 奥比中光所有产品均为USB接口，大部分产品为标准Type-A接口。



- Deeyea为Type-C接口，详细接口定义可查看规格书。
- P1为micro-USB接口，20PIN的附加接口为定制，可对普通用户开放定义，不提供线材。

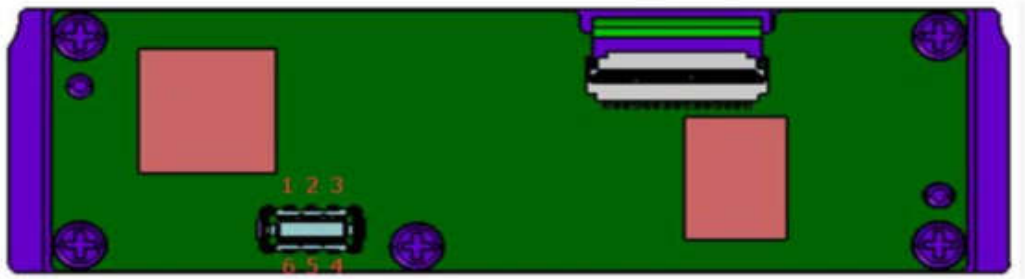
Pin	名称	线的颜色	描述
1	VBUS	Red (红)	电源正 5 V
2	D-	White (白)	数据线负
3	D+	Green (绿)	数据线正
4	ID	none (无)	分为 A 和 B 两种接口 A: 与地线相连 B: 不与地线相连
5	GND	Black (黑)	电源负

表 1 Micro USB 对应接口的功能

Pin	名称
1	VBUS
2	VBUS
3	VBUS
4	GND
5	GND
6	D+
7	D-
8	GND
9	GND
10	GND

表 2 20PIN 连接器对应接口的功能

- Astra D-U物理接口为6PIN的B2B连接器，型号为CPB9506-0113E，通过USB协议传输数据



POSITION	NAME
1	VBUS
2	D+
3	D-
4	GND
5	ID
6	GND

2.2 结构集成设计

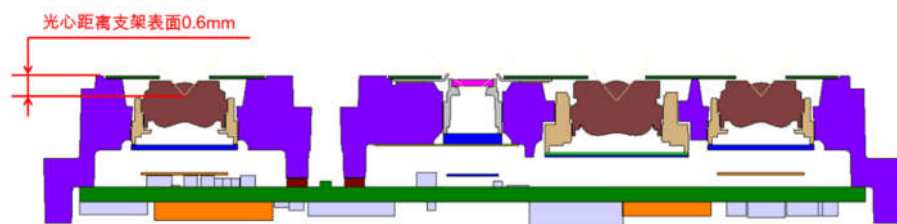
2.2.1 结构密封设计参考

详见产品Datasheet中结构密封设计部分描述。

2.2.2 绝对距离参考点

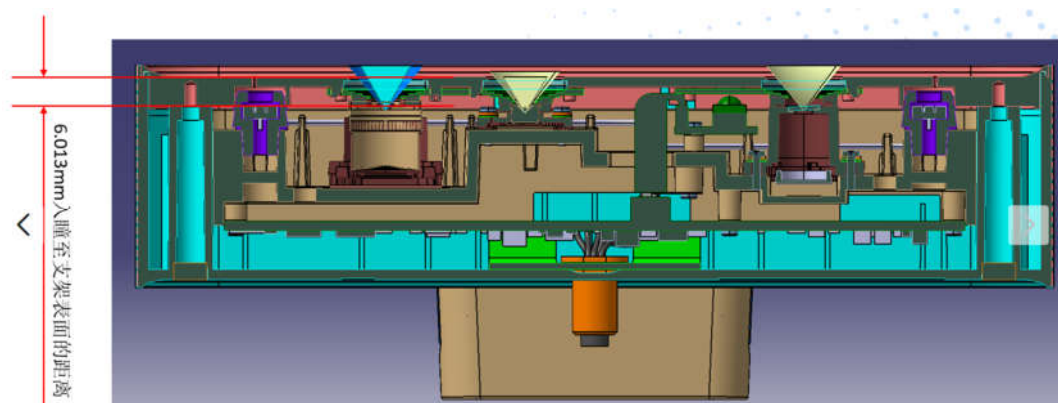
1) Astra D-U

Astra D-U的入瞳到支架表面的距离为0.6mm，测试绝对距离时，常以支架外表面为参考。



2) Astra S

Astra S的入瞳至外壳内表面的距离为6.013mm，测试绝对距离时，常以外壳表面为参考。



2.2.1 散热设计参考

公司小型化产品（Astra mini、Deeyea、Astra D-U等）嵌入设计时需要考虑散热，详见散热设计参考文件。

2.3 驱动安装与系统适配

2.3.1 Windows系统驱动安装与诊断

1) 系统要求

Windows 7, 8 and 10 (x86, x86-64); x86-based processor @ 1.8+ GHz

2) 驱动导入

Windows系统测试使用前，需要安装驱动，驱动文件在SDK/drivers目录下，正确安装驱动后，可以通过设备管理器查看设备。如果遇到异常，请查看SDK中《Astra驱动安装及设备诊断指南》进行驱动异常处理。

2.3.2 Android系统适配

1) 系统要求

Android系统无须额外安装产品驱动。

- Android OS 4.4/ 5.1/6.0/7.1
- USB 2.0 或 3.0 (支持 host 接口)
- 建议 RK3288 (四核 Cortex-A17 , 主频 1.8GHz) 或以上
- 建议 2GB RAM 或以上
- 支持 LibUSB+LibUVC
- 支持 UVC 设备
- 支持 SELinux 权限访问 UVC 设备

2) 测试方法

安装SDK中的NiViewer.apk应用到对应的设备中，就可以查看设备的深度图和彩色图。

有以下要点需要注意：

- ✧ Orbbec设备分为UVC和非UVC设备，对于UVC设备，如Deeyea、苍龙等，是无法通过NiViewer_for_android.apk查看彩色图像的；UVC为通用的USB视频捕获设备的协议标准，可以使用Android系统相机直接打开查看彩色图像，开发时可以使用libuvc或者Android系统camera。
- ✧ Orbbec不同设备的分辨率不同，使用NiViewer_for_android.apk注意选择对应的分辨率，如Deeyea、P1等需要选择640*400分辨率，而Orbbec Pro A需要选择640*480，否则深度图预览不正常。

3) 异常处理

✧ Android设备无法识别Camera

Android设备常见设备无法识别的问题，可通过如下步骤排查解决：

- 将设备插入Windows端，安装驱动SensorDriver，查看设备管理器中是否正常加载设备。
如果无法正常加载，安装《驱动安装与诊断指南》排除驱动问题；
- 使用Windows端NiViewer.exe测试设备能否正常出图，排除设备问题
- 如果正常，重新接入Android设备，连接adb，使用dumpsys USB，查看能否正常加载Orbbec设备节点。Orbbec设备的Vendor ID是2bc5。
- 如果无法正常加载，通常判断是设备端USB接口硬件电路或者供电问题，Orbbec设备一般需要5V 500mA供电，启动瞬间电流峰值可至700mA。

- e) 如果找到的设备节点，基本可判断是系统兼容性问题，确认系统是否支持USB host。对于Android6.0以上系统，需要确认关闭Selinux权限，使用`getenforce`查看状态是否为Permissive；如果否，使用`setenforce 0`在管理员权限下将其关闭。
- f) 如果仍然无法正常使用设备，可打开NiViewer应用，并抓取logfile具体分析。

3.0 SDK介绍与使用

3. SDK介绍与使用

3.1 SDK简介

奥比中光提供的OpenNI2.3系列SDK基于OpenNI2开发，OpenNI2（开放自然交互）是一个多语言、跨平台的框架，它定义了应用程序、中间件和3D传感设备之间的接口。

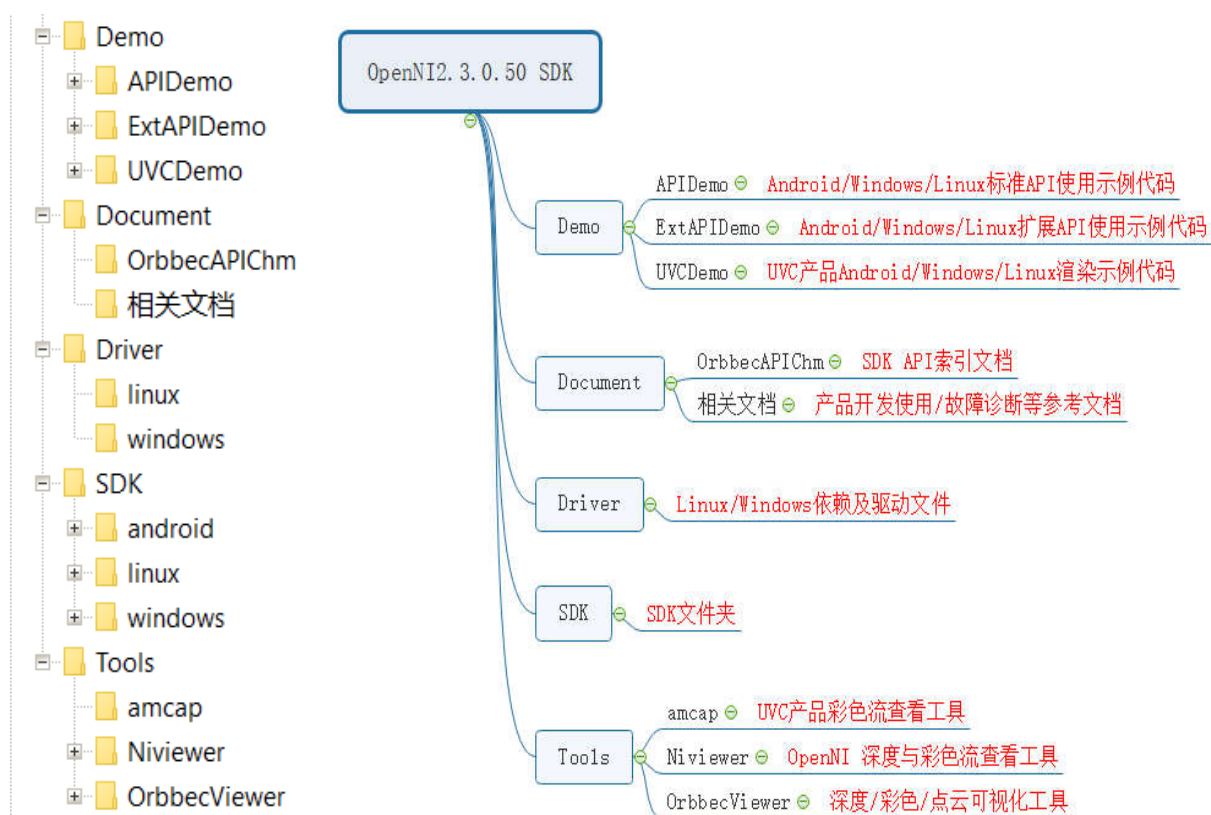
奥比中光的全系列产品全面支持OpenNI协议，这意味着任何使用OpenNI开发的应用，都可以无缝对接到公司产品。

在使用OpenNI2.3.0.50SDK时，假定使用者对OpenNI有一定了解，如果之前未接触过OpenNI相关开发，可以参考OpenNI2官方帮助文档。OpenNI2的整体框架如下图所示：



- 最上层为关于OpenNI2的应用如NITE手势识别，身体运动检测等
- 接下来是OpenNI2的对外提供的统一接口，这些接口对应的头文件为OpenNI.h
- OpenNI Core为OpenNI2的核心部分，OpenNI.h中的结构实现都在这部分；对Driver层提供统一的API，用于Driver的开发扩展，这部分API对应的头文件为OniDriverAPI.h
- 最底层为硬件驱动、或者第三方库

3.2 OpenNI2.3.0.50 SDK目录结构



3.2 SDK 工程环境及配置

3.2.1 Windows工程环境配置

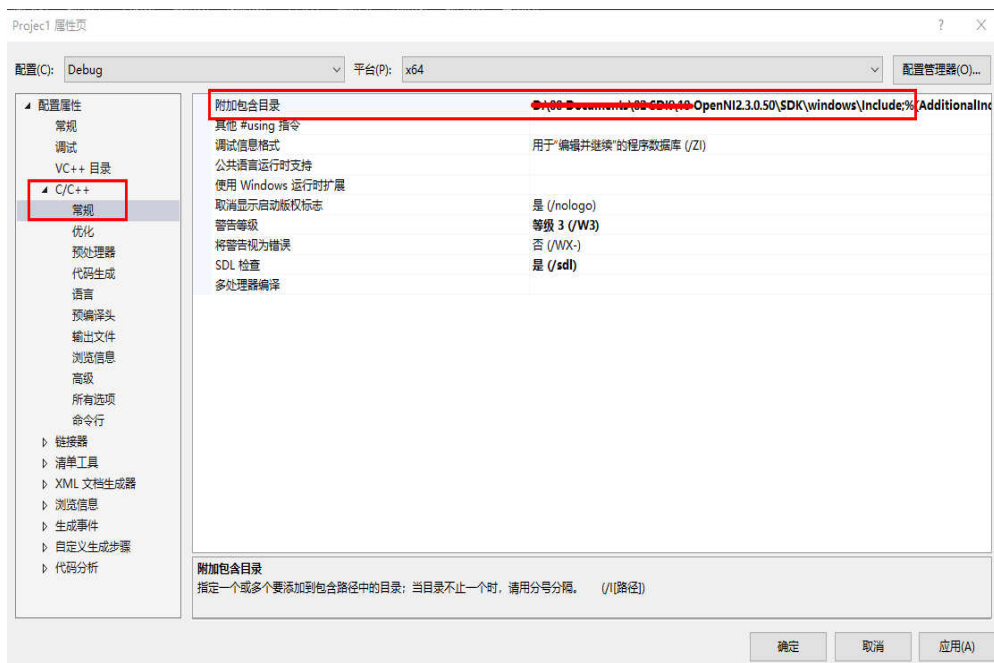
(1) 适用范围

本公司SDK适用X86/X64 Windows7及以上平台

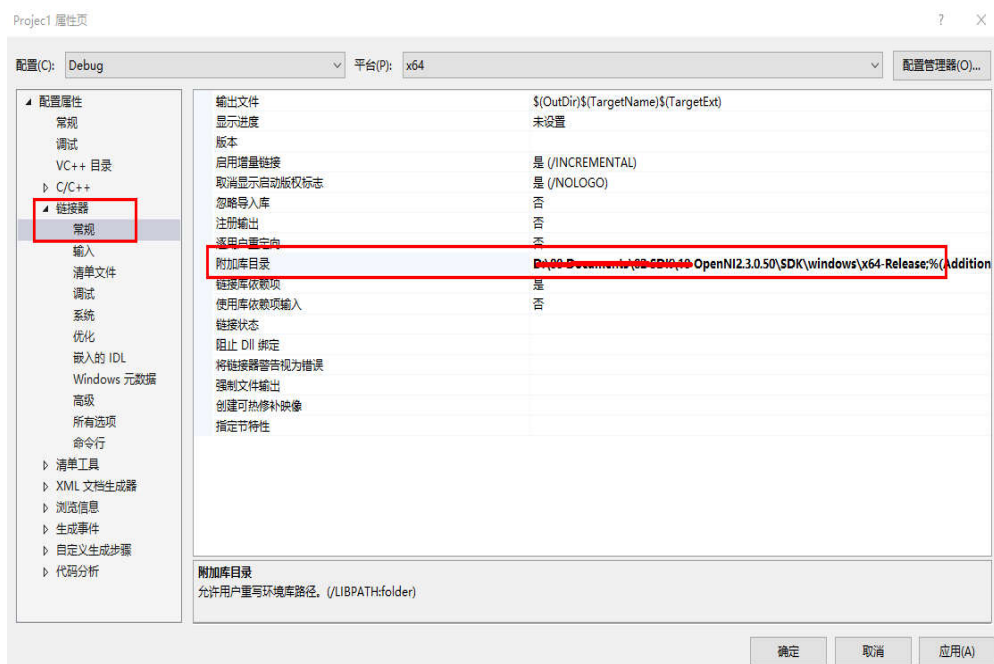
(2) 开发平台搭建

推荐的Windows开发平台为Visual Studio 2013或以上版本。对于Visual Studio平台，可通过以下步骤搭建开发环境：

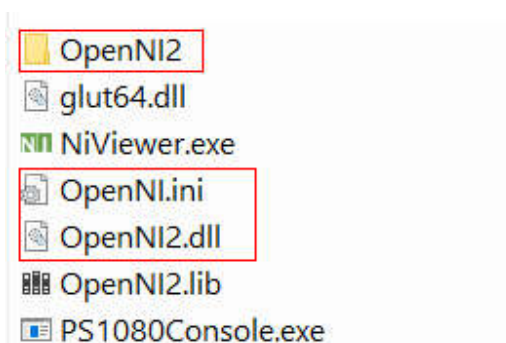
- 新建或打开C++工程
- 选择“项目——属性”，将\SDK\windows\Include目录添加到头文件依赖路径下，注意属性配置（x86/x64）与运行时一致



- 选择“链接器——常规”项目，添加x86或者x64对应的OpenNI.lib文件所在目录，



- 在“链接器——输入”目录下新增OpenNI.lib项，完成环境配置，可以开始编辑自己的代码了。
- 注意编译运行工程时，把x64-release或者x86-release文件夹里面OpenNI文件夹以及ini和dll文件拷贝到exe文件所在目录下，否则会闪退或报错。



(3) SimpleCode 说明

Windows系统下，针对SDK中标准API、扩展API等应用提供了不同sample code，具体操作和使用说明，请查看对应文件夹下的readme文件。

获取深度流步骤：

- (1) OpenNI 的初始化，该方法必须调用，在此之后才能使用Videostream、 Device等

```
Status rc = OpenNI::initialize();
```

- (2) 打开一个设备，这里的ANY_DEVICE为打开系统中连接的第一个设备

```
Device device;
```

```
rc = device.open(ANY_DEVICE);
```

- (3) 打开设备后，获取传感器信息，然后根据传感器信息创建视频流

```
if (device.getSensorInfo(SENSOR_DEPTH) != NULL)
```

- (4) 创建一个深度视频流

```
rc = depth.create(device, SENSOR_DEPTH);
```

- (5) 调用start方法开始采集深度信息

```
rc = depth.start();
```

- (6) 采用轮询方式获取码流信息

```
VideoFrameRef frame;
```

```
VideoStream* pStream = &depth;
```

```
rc = OpenNI::waitForAnyStream(&pStream, 1, &changedStreamDummy,
```

```
SAMPLE_READ_WAIT_TIMEOUT);
```

- (7) 读取一帧数据

```
rc = depth.readFrame(&frame);
```

- (8) 获取深度图像中心点的深度值

```
DepthPixel* pDepth = (DepthPixel*)frame.getData();
```

```
float x,y,z;
CoordinateConverter coorvert;
int middleIndex = (frame.getHeight()+1)*frame.getWidth()/2;
coorvert.convertDepthToWorld(depth, frame.getWidth()/4, frame.getHeight()/4,
pDepth[middleIndex], &x, &y, &z);
```

(9) 停止视频采集后关闭相关设备

```
depth.stop();
depth.destroy();
device.close();
OpenNI::shutdown();
```

3.2.2 Linux环境配置

(1) 适用范围

Linux平台下，本公司提供针对Intel x86/x64和ARM32/64的SDK包。使用时，请注意平台一致性。对于ARM平台，根据处理能力不同，提供了正常版本和without-filter（无软件滤波）版本两种SDK包，如运行正常版本时卡顿，可采用无软件滤波版本的SDK。

(2) Linux环境配置说明

在Linux平台上使用OpenNI2 SDK进行相关开发前，请执行如下命令安装配置相关环境。以x64平台为例，为了运行相关程序首先需要安装如下程序：

```
$sudo apt-get install cmake
$sudo apt-get install libusb-1.0-0-dev
$sudo apt-get install libudev-dev
$sudo apt-get install libglfw3-dev
$sudo apt-get install build-essential freeglut3 freeglut3-dev
```

检查udev版本，Orbbec的驱动需要依赖libudev.so.1，可以执行如下命令查找，如果没有找到需要手动链接libudev.so.x.x到libudev.so.1

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

将SDK包拷贝到某个路径，解压缩，进行例程编译。


```

$cd Demo/APIDemo/example/build      -----常用代码例程
$make ..
$make
$cd Demo/ExtAPIDemo/linux-x64/ExtendedAPI      ---扩展Api代码例程
$make
$cd Demo/UVCdemo/Linux/OrbbecStreamSample-Linux      ---uvc代码例程

```

插上摄像头运行例程程序。

```

$lsusb      ---确认设备是否识别
$./DepthRead      ---运行对应例程

```

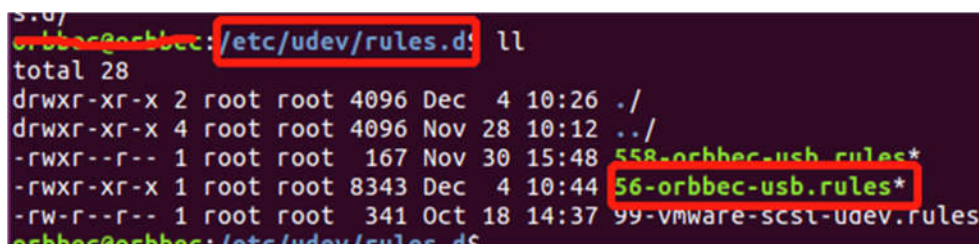
(3) udev 说明

Linux系统下，为了保证摄像头设备能被正常识别打开，应加入对应udev，具体操作如下：

```

$cd Driver/linux
$ ./ create_udev_rules

```



```

$ cd /etc/udev/rules.d; ll
total 28
drwxr-xr-x 2 root root 4096 Dec  4 10:26 ./
drwxr-xr-x 4 root root 4096 Nov 28 10:12 ../
-rwxr--r-- 1 root root 167 Nov 30 15:48 558-orbbec-usb.rules*
-rwxr-xr-x 1 root root 8343 Dec  4 10:44 56-orbbec-usb.rules*
-rw-r--r-- 1 root root 341 Oct 18 14:37 99-vmware-scsi-udev.rules
orbbec@orbbec: /etc/udev/rules.d$

```

(4) SimpleCode说明

Linux系统下，针对SDK中标准API、扩展API等应用提供了不同sample code，具体操作和使用说明，请参考Linux环境配置中例程编译步骤，获取深度流步骤请参考Windows。

3.2.3 Android工程环境配置

(1) 适用范围

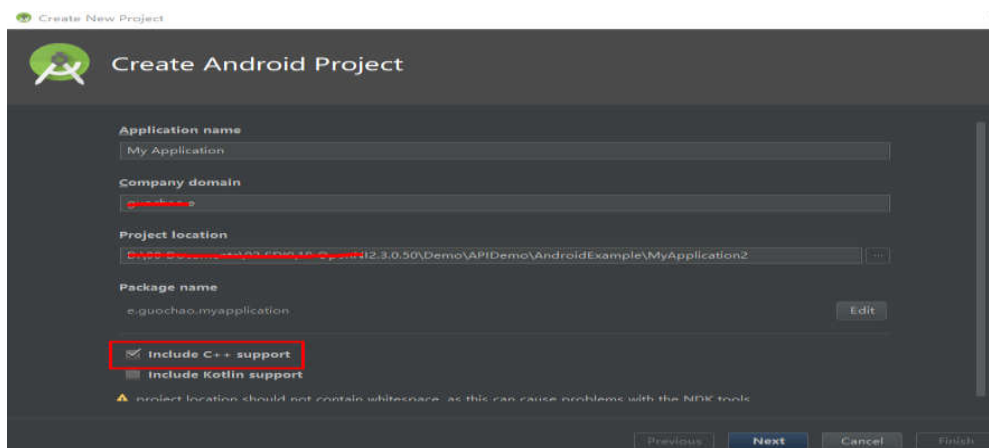
Android平台下，本公司SDK适用于Android 4.4.2及以上平台

(2) Android环境配置说明

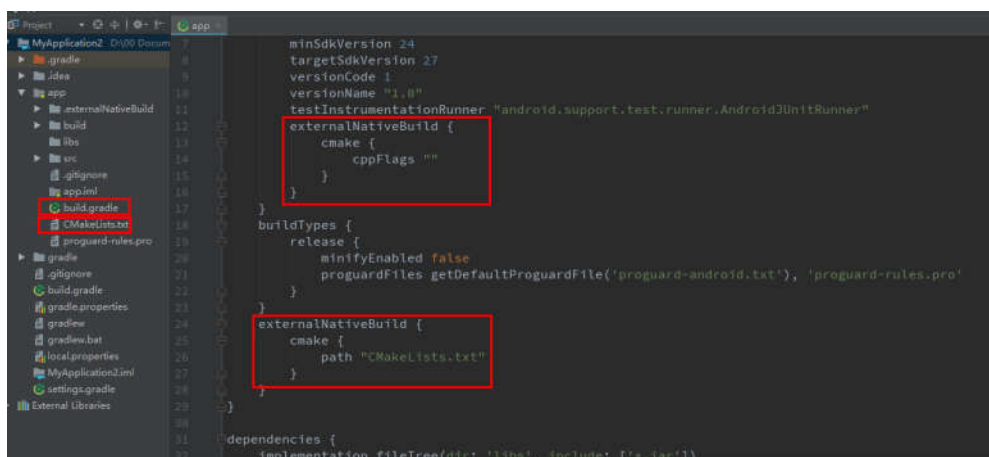
进行Android SDK开发之请搭建好Android Studio开发环境。可以搭建纯Java环境进行开发，也可以搭建支持C++的Java工程进行开发，配置NDK，通过JNI调用C++ API。Android2.2之前通过

NDK+Android.mk+Application.mk实现JNI，之后的版本则采用ndk+Cmake方式实现。下面介绍下Android Studio NDK环境的配置过程。

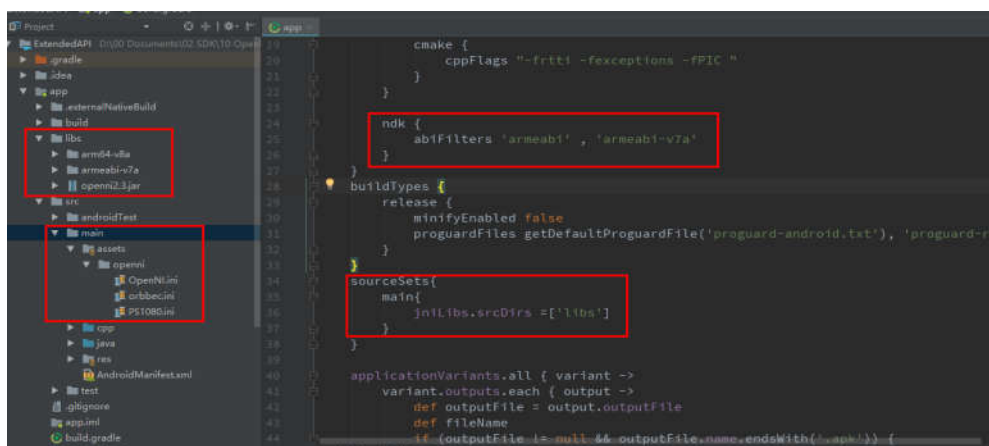
- 新建支持C++的Android Studio，勾选Include C++ support，点击Next直至创建完成。



- 通过目录结构可以看出，相比普通Android工程有几项不同。



- 添加SDK中库和资源文件，配置gradle文件



- 修改cmake文件，配置创建的cpp文件，就可以在Java工程中调用C++ API，详细工程配置及API注册调用可以参考ExtAPIDemo工程文件。

```

10 # You can define multiple libraries, and CMake builds it for you.
11 # Gradle automatically packages shared libraries with your APK.
12
13 add_library(OpenNI2
14             SHARED
15             IMPORTED
16 )
17
18 set_target_properties(OpenNI2
19                       PROPERTIES IMPORTED_LOCATION
20                                ../../../../libs/armeabi-v7a/libOpenNI2.so)
21
22 add_library( # Sets the name of the library.
23             IRUtils
24
25             # Sets the library as a shared library.
26             SHARED
27
28             # Provides a relative path to your source file(s).
29             # Associated headers in the same location as their source
30             # file are automatically included.
31             src/main/cpp/IRUtils.cpp )
32
33
34
35
36
37
38
39
40 find_library( # Sets the name of the path variable.
41             log-lib
42
43             # Specifies the name of the NDK library that
44             # you want CMake to locate.
45             log )
46
47 # Specifies libraries CMake should link to your target library. You
48 # can link multiple libraries, such as libraries you define in the
49 # build script, prebuilt third-party libraries, or system libraries.
50
51 target_link_libraries( # Specifies the target library.
52                       IRUtils
53                       OpenNI2
54                       # Links the target library to the log library
55                       # included in the NDK.
56                       ${log-lib} )

```

(3) SimpleCode说明

获取深度流步骤

1、创建OpenNIHelper类

```
private OpenNIHelper mOpenNIHelper;
mOpenNIHelper = new OpenNIHelper(this);
```

2、注册设备打开监听器

```
mOpenNIHelper.requestDeviceOpen(this);
```

3、实现DeviceOpenListener接口抽象函数onDeviceOpened、onDeviceOpenFailed

```
@Override
```

```

public void onDeviceOpened(UsbDevice device) {
    init(device);
    mStream = VideoStream.create(mDevice, SensorType.IR);
    mVideoModes = mStream.getSensorInfo().getSupportedVideoModes();
    for (VideoMode mode : mVideoModes) {
        int X = mode.getResolutionX();
        int Y = mode.getResolutionY();
        int fps = mode.getFps();
        Log.d(TAG, " support resolution: " + X + " x " + Y + " fps: " + fps + ", (" +
mode.getPixelFormat() + ")");
        if (X == mWidth && Y == mHeight && mode.getPixelFormat() == PixelFormat.RGB888 &&
fps == 30) {
            mStream.setVideoMode(mode);
            Log.v(TAG, " setmode");
        }
    }
    startThread();
}

@Override
public void onDeviceOpenFailed(String msg) {
    showAlertAndExit("Open Device failed: " + msg);
}

```

4、init(device)中完成设备初始化，并打开OpenNI类设备

```

private void init(UsbDevice device) {
    OpenNI.setLogAndroidOutput(true);
    OpenNI.setLogMinSeverity(0);
    OpenNI.initialize();
    List<DeviceInfo> opennolist = OpenNI.enumerateDevices();
    if (opennolist.size() <= 0) {
        Toast.makeText(this, " openni enumerateDevices 0 devices", Toast.LENGTH_LONG).show();
        return;
    }
    mDevice = null;
}

```

```

//Find device ID
for (int i = 0; i < opennilist.size(); i++) {
    if (opennilist.get(i).getUsbProductId() == device.getProductId()) {
        mDevice = Device.open();
        break;
    }
}

if (mDevice == null) {
    Toast.makeText(this, " openni open devices failed: " + device.getDeviceName(),
Toast.LENGTH_LONG).show();
    return;
}
}

```

5、startThread（）开启线程读取视频流

```

void startThread() {
    mInit_Ok = true;
    m_thread = new Thread() {
        @Override
        public void run() {
            List<VideoStream> streams = new ArrayList<VideoStream>();
            streams.add(mStream);
            mStream.start();

            while (!mExit) {
                try {
                    OpenNI.waitForAnyStream(streams, 2000);
                } catch (TimeoutException e) {
                    e.printStackTrace();
                    continue;
                }
                synchronized (m_sync) {
                    mGLView.update(mStream);
                }
            }
        }
    };
}

```

```

    }
}
};
m_thread.start();
}

```

3.2.4 .ini配置文件使用说明

(1) OpenNI.ini配置说明

OpenNI2.3.0.50提供了丰富的调试log，开启相应层级的log，就能在屏幕或者文件中查看当前深度、彩色流的信息。该功能可以通过修改OpenNI.ini文件来实现。

```

[Log]
; 0 - Verbose; 1 - Info; 2 - Warning; 3 - Error. Default - None
; Verbosity=0 // 去掉注释符 ;, 修改为 Verbosity=0 查看最详细调试信息
; LogToConsole=1 // 去掉注释符 ;, 修改为 LogToConsole=1 将调试信息输出到控制台
; LogToFile=1 // 去掉注释符 ;, 修改为 LogToFile=1 可以保持 log 到文件
; LogToAndroidLog=1 // 去掉注释符 ;, 修改 LogToAndroidLog=1 输出 Log 到 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=

```

(2) Orbbec.ini配置说明

OpenNI2.3.0.50提供了灵活的外部配置功能，通过修改orbbec.ini文件，软件库通过解析配置文件设置摄像头工作状态，以下为配置文件关键设置项。

```

;----- Sensor Default Configuration -----
[Device]
; Mirroring. 0 - Off (default), 1 - On
;Mirror=1

; FrameSync. 0 - Off (default), 1 - On
;FrameSync=1

; Stream Data Timestamps. 0 - milliseconds, 1 - microseconds (default)
;HighResTimestamps=1

; Stream Data Timestamps Source. 0 - Firmware (default), 1 - Host
;HostTimestamps=0

; A filter for the firmware log. Default is determined by firmware.
;FirmwareLogFilter=0

; Automatic firmware log retrieval. 0 - Off (default), or the number of milliseconds between log retrievals operations.
;FirmwareLogInterval=1000

; Print firmware log to console when automatic firmware log retrieval is on. 0 - Off (default), 1 - On
;FirmwareLogPrint=1

; Is APC enabled. 0 - Off, 1 - On (default)
;APCEnabled=1

; USB interface to be used. 0 - FW Default, 1 - ISO endpoints (default on Windows), 2 - BULK endpoints (default on Linux/Mac/Android machines)
;UsbInterface=2

```

符号[]内项目表示当前配置设备，可对整机Device、深度相机Depth、彩色相机Image和红外相机IR项进行配置，同时保留了深度图和彩色图裁剪的配置项Depth.Cropping和IR.Cropping。以上图[Device]中配置项为例，想要修改具体项目，需要将注释用的分号去掉，再根据说明选择对应配置值。下面对常用的配置项目进行说明。

➤ [Device]:

Mirror——设备镜像模式选择；

FrameSync——深度与彩色帧同步设置，应注意，仅非UVC设备支持该配置项。

➤ [Depth]:

outputFormat——选择输出的深度原始数据格式

Mirror——设置深度图镜像模式

Resolution——深度图像分辨率设置

FPS——深度帧率选择

Registration——设置RGBD硬件对齐

HoleFilter——设置滤波模式，对于深度和彩色分辨率不一致的机型，通过设置HoleFilter=2，可以去除RGBD对齐之后深度图上的黑洞。

➤ [Image]:

outputFormat——选择输出的彩色原始数据格式

Mirror——设置彩色图像镜像模式

其余配置项与Depth类似，但须注意，Image配置项下的设置值仅对非UVC设置的彩色相机生效，UVC相机，如Deeyea、Astra D-U等，无法通过该项配置彩色相机。

➤ [IR]:

outputFormat——选择输出的近红外图像数据格式

Mirror——设置IR图像镜像

Resolution——选择IR图像分辨率

应注意的是，由于Depth图像是基于IR图像计算得到的，因此对于Depth图像的分辨率、帧率等配置修改，需要同步修改IR配置项中对应选项。

4.0 常用API介绍

4. 常用API介绍

在SDK包中Document文件夹下，提供了OrbbecSDK_v2.3.chm文件，汇总了SDK中所有支持的API，并提供接口介绍和使用说明。也可根据开发语言的不同，选择c++.chm或者openni2java.chm查看C++或Java API说明，文档目录结构如下：



4.1 标准API介绍

标准API是OpenNI开源软件框架提供的原生API，包含设备类Device、视频流类VideoStream、视频模式类VideoMode等常用类。

获取深度视频流需要使用4个主要的类（class）。

1. openni::OpenNI提供了一个静态的API进入点。它提供访问设备，设备相关事件，版本和错误信息。当然，首先得确保你连接了设备。
 2. openni::Device提供了一个传感器设备连接系统的接口（个人理解就是通过Device类来访问控制传感器）。在它创建之前需要对OpenNI类进行初始化。Device可以访问流（Streams）。
 3. openni::VideoStream从一个设备（Device）里提取一个视频流，需要获取视频帧引用（VideoFrameRefs）。
 4. openni::VideoFrameRef从相关的源数据里提取一个视频帧。这是从一个特定的流里面获取的。
- 除了这些主要的类以外，还有许多类和结构体用来保持一些特殊类型的数据。Recorder类就是用来存储OpenNI视频流到文件的。还有Listener类用来监听OpenNI和Stream类产生的事件。视频流可以通过两种方式来获取数据：轮询和事件。下面具体介绍各个常用类：

4.1.1 OpenNI类

1) 简介

OpenNI类提供了库函数的静态入口，每个OpenNI2.0应用均需要使用OpenNI类初始化SDK和设备驱动。也可以生成许多设备连接和断开事件，和提供以轮询方式访问数据流的功能一样。

2) 设备的基础访问

OpenNI类提供了OpenNI::initialize()方法初始化库函数并且扫描系统中所有可用的传感器设备。所有使用OpenNI的应用程序在使用其他API之前都应该调用此方法。

一旦初始化方法完成，将可能会创建设备(Device)对象，并使用这些对象来和真实的传感器硬件进行交互。OpenNI::enumerateDevices()方法会返回一个已经连接上系统的可用的传感器设备列表。

当应用程序准备退出时，必须调用OpenNI::shutdown()方法来关闭所有驱动并且正确退出。

3) 视频流 (Video Streams) 的基础访问

流的轮询访问接口是OpenNI::waitForStream()方法实现的。此方法的参数之一就是流的列表。调用时，会锁定直到列表中的流有新的可用数据或超时。然后返回一个状态码(status code)并指向具体数据流。

4) 设备的事件驱动访问

OpenNI类提供了一个在事件驱动方式 (event driven manner) 中访问设备的框架。OpenNI定义了3个事件：设备连接事件(onDeviceConnected)，设备断开事件(onDeviceDisconnected)，设备状态改变事件(onDeviceStateChanged)。设备连接事件是在一个新的设备连接并通过OpenNI可用时产生的，设备断开事件是在一个设备从系统中移除时产生的。设备状态改变事件是在设备的设置被改变时产生的。

可以用下列方法从事件处理列表中增添或者移除监听器类(Listener classes):

OpenNI::addDeviceConnectedListener()//添加设备连接事件监听器

OpenNI::addDeviceDisconnectedListener()//添加设备断开事件监听器

OpenNI::addDeviceStateChangedListener()//添加设备状态改变事件监听器

OpenNI::removeDeviceConnectedListener()//移除设备连接事件监听器

OpenNI::removeDeviceDisconnectedListener()//移除设备断开事件监听器

OpenNI::removeDeviceStateChangedListener()//移除设备状态改变事件监听器

3个事件都提供了一个指针指向OpenNI::DeviceInfo对象。这个对象用来获取被事件提交的设备的细节和标识。此外，设备状态改变事件还提供了一个指针指向DeviceState对象，这个对象被用来查看设备新的状态信息。事件驱动通过视频流类(VideoStream class)来访问真实的视频流

5) 错误信息

在SDK中有许多方法都会返回一个类型为“Status”的值。当错误发生，Status就会包含有一个记录或者显示给用户的代码。OpenNI::getExtendedError()方法会返回更多的关于错误的可读信息。

6) 版本信息

API的版本信息由OpenNI::getVersion()方法来获取。这个方法返回应用程序目前使用的API的版本信息

4.1.2 Device类

1) 简介

openni::Device类提供了一个物理硬件设备的抽象接口，也可以通过一个从物理设备得来的ONI记录文件提供了一个模拟硬件设备的接口。

Device对象被用来连接和配置底层文件或者硬件设备，并从设备中创建流。Device对象提供了设备配置查询和修改的接口，可用于使能RGBD对齐或帧同步。

在创建和初始化视频流之前，必须构造Device对象，并指向物理设备。

2) 设备连接条件

在设备类能连接到物理硬件设备前，设备必须在物理上正确地连接到主机，并且驱动必须安装完毕。如果连接的是ONI文件，那要求在系统运行应用程序时ONI记录文件必须可用，而且应用程序有足够的权限去访问。

当然，也需要在联系设备前openni::OpenNI::initialize()方法被调用。这将会初始化驱动，使API知道设备连接了。

3) 基本操作

➤ 构造函数

设备类（Device class）的构造函数没有参数，也不会联系到物理硬件设备。只是简单地创建对象。

➤ Device::open()

此方法用来连接到物理硬件设备。open()方法有一个参数，设备的URI（统一资源标识符），方法返回一个状态码指示是否成功。

最简单的用法是用常量openni::ANY_DEVICE作为设备的URI，用这个常量会使系统连接所有的硬件设备，但仅适合只有一个设备连接的情况。

如果多个传感器连接了，那就得先调用`OpenNI::enumerateDevices()`来获取可用的设备列表。然后找到你要找的设备，通过调用`DeviceInfo::getUri()`来获取URI，用此方法的输出作为`Device::open()`的参数，然后就能打开对应的设备了。

如果打开文件，那参数就是ONI文件的路径。

➤ `Device::close()`

`close()`方法用来关闭硬件设备。按照惯例，所有打开的设备必须要关闭。这会分离硬件设备和驱动，这样后面的应用程序才能再次打开设备。

➤ `Device::isValid()`

`isValid()`方法用来确定设备是否正确地和设备对象连接

➤ `Device::setImageRegistrationMode()`

Orbbec产品会同时产生深度流和彩图流，分别由不同的物理摄像头（RGB camera和IR camera）生成。物理摄像头的实际位置不同，导致深度和彩色画面的视场差异，这就使得从同一设备对象不同的流得到的图像有所不同。

两个摄像头之间的几何关系和距离对于设备对象来说都是已知的，就可以通过数学上的变换来使得两幅图像能够一致，相互叠加，比如深度图的每个像素叠加到彩图上。这个过程就是对齐（每一个像素叠加到另一张图上）。

Orbbec设备支持硬件运算，可以在模组ISP内校准数据。同时，那硬件上有个标示flag来进行开或关。

设备对象提供了`isImageRegistrationSupported()`方法来测试已连接的设备是否支持对齐功能。

如果支持，那`getImageRegistrationMode()`能用来查询这个功能的状态，

`setImageRegistrationMode()`就可以设置它。`openni::ImageRegistrationMode`枚举提供了以下值用来set或get:

`IMAGE_REGISTRATION_OFF` - Hardware registration features are disabled 硬件对齐功能被禁用

`IMAGE_REGISTRATION_DEPTH_TO_IMAGE` - The depth image is transformed to have the same apparent vantage point as the RGB image 深度图像被变换叠加至彩图上

需要注意的是两个传感器的可视范围（FOV）有部分不重叠。这就导致部分深度图不会在显示在结果中。

➤ `Device::setDepthColorSyncEnabled()`

当深度和彩图流都可用，轻微的帧率不同或者帧到达时间不同，可能会导致帧同步问题。

Orbbec为非UVC设备提供帧同步的功能，为了在确定的最大时间范围内分别从获取到两个帧，通常这个最大值都小于两帧间隔。

启用或禁用此功能用`setDepthColorSyncEnable()`。

但应当注意的是，所有UVC设备都无法通过该API进行帧同步设置。

4.1.3 VideoStream类

1) 简介

由设备类创建的视频流类封装了所有的数据流。用户可以对数据流进行开始，停止和配置，也可以进行流一级（和设备一级相对）的参数配置。

2) 视频流的基础功能

➤ 创建和初始化视频流

调用视频流默认的构造函数会创建一个空的未初始化的视频流对象。在使用前，这个对象必须调用`VideoStream::create()`进行初始化。而`create()`方法要求一个已经初始化的设备对象。一旦创建，你就可以调用`VideoStream::start()`方法来产生数据流。`VideoStream::stop()`方法则会停止产生数据流。

➤ 基于轮询的数据读取

一旦视频流创建完毕，就可以通过`VideoStream::readFrame()`方法进行读取数据。如果有新的可用数据了，这个方法就会返回一个可以访问由视频流生成的最新的视频帧引用（`VideoFrameRef`）。如果没有新的帧可用，那就会锁定直到有新的帧可用。

需要注意的是，如果非循环地从记录中读取，那么在最后一帧读取完毕后程序将永远卡死在此方法。

➤ 基于事件的数据读取

可以通过事件驱动方式（`event driven manner`）从视频流中读取数据。

首先，需要创建一个`VideoStream::Listener`的继承类，此类应该实现方法`onNewFrame()`。一旦你创建了这个类，实例化了它，就可以通过`VideoStream::addListener()`方法来添加监听器。当有新的帧到达，自定义的监听器类的`onNewFrame()`方法就被调用。然后你就可以调用`readFrame()`读取了。

3) 获取关于视频流的信息

➤ 传感器信息（`SensorInfo`）和视频模式（`VideoMode`）

传感器信息和视频模式类可以一直追踪视频流的信息。视频模式封装了视频流的帧率（`frame rate`），分辨率（`resolution`）和像素格式（`pixel format`）。传感器信息包含了产生视频流的传感器

的类型和每个流的视频模式对象列表。通过遍历这个列表，那就能确定传感器生成的流的所有可能的模式。

使用VideoStream::getSensorInfo能够得到当前流的传感器信息对象

➤ 视野（Field of View）

此功能为确定的传感器的视野范围。使用getHorizonFieldOfView()和getVerticalFieldOfView()方法来确定视野，其返回的值是弧度。

➤ 像素最大最小值（Min and Max PixelValues）

在深度流中，通常知道一个像素可能出现的最大值和最小值是很有用的。用getMinPixelValue()和getMaxPixelValue()方法就能获取这些信息。

4) 配置视频流

➤ 视频模式（Video Mode）

可以设置给定流的帧率（frame rate），分辨率（resolution）和像素格式（pixel type）。设置这些就要用到setVideoMode()方法。在此之前，你首先需要获取已配置视频流的传感器信息

（SensorInfo），然后你才能选择一个可用的视频模式。

➤ 裁剪（Cropping）

如果传感器支持裁剪，视频流会提供方法来控制它。使用VideoStream::isCroppingSupported()方法来确定是否支持。

如果支持，使用setCropping()来使能裁剪和设置裁剪的具体配置。ResetCropping()方法被用来再次关闭裁剪。getCropping()方法用来获取当前的裁剪设置。

➤ 镜像（Mirroring）

镜像，顾名思义，就是使视频流所展现的看起来就像在镜子里一样。启用或禁用镜像，使用VideoStream::setMirroringEnable()方法。设置true为启用，设置false为禁用。可用getMirroringEnable()来获取当前设置。

➤ 通用属性（General Properties）

在固件层，大多数的传感器设置都存储为地址/值的对（address/value pairs，就是一种键值对）。所以可以通过setProperty和getProperty方法直接操作。这些方法被SDK内部用来实现裁剪，镜像，等等。而它们通常不会被应用程序频繁地使用，因为大多数有用的属性都被更加友好的方法封装了。

4.1.4 VideoFrameRef类

1) 简介

视频帧引用类封装了从视频流读取的单个帧的所有数据。它提供了访问包含了帧数据（元数据，工作所需的帧）基础数组的方法。

视频帧引用对象是从VideoStream::readFrame()方法获取的。

视频帧引用数据可以从红外摄像头，RGB摄像头或者深度摄像头获取。getSensorType()方法用来确定产生此帧的传感器类型。它会返回传感器类型。

2) 访问帧数据

VideoFrameRef::getDate()方法返回一个直接指向帧数据的指针。类型为void，这样每个像素的数据类型才能正确地索引。

3) 数据裁剪（Cropping data）

数据帧引用知道视频流的裁剪设置，因此可以用来确定裁剪框的原点，裁剪框的大小和帧是否启用裁剪功能。实现方法如下：getCropOriginX(), getCropOriginY(), getCroppingEnable().若启用裁剪功能，则裁剪框大小等于帧大小。

4) 时间戳(Timestamp)

每帧数据都有个时间戳。这个值是基于任意0值开始的微秒数。不同于两帧之间时间差，同一设备的所有流用的都是同一0值，所以时间戳的差值可以用来比较不同流的帧。

OpenNI 2.0中，时间戳的0值是第一帧数据的到达时间。然而这无法保证每次都一样，所以程序代码应该使用时间戳增量。时间戳的值本身不应该用作一种绝对的时间指向。

5) 帧索引（FrameIndex）

除了时间戳，帧还提供了连续的帧索引号。这在确定已知的两帧之间有多少帧很有用。如果使能了帧同步功能，那相应的帧的帧号应该就是一致的。

如果没有同步，那帧号将不一定匹配。这种情况下，用时间戳来确定相关帧的位置更有效。

5) 视频模式（Video Mode）

VideoFrameRef::getVideoMode()用来确定生成当前帧的传感器的视频模式。信息包括像素格式，分辨率，帧率。

6) 数组跨度（Array Stride）

包含帧的数组跨度可以用getStrideBytes()来获取。它将返回数组每行数据的大小，单位字节byte。主要用于索引二维图像数据。

4.1.5 其他支撑类

1) 简介

OpenNI除了上述主要类外，还有一系列的支撑类。这些类主要服务于封装数据，在其他主要类的章节都有所提及。

2) 传感器配置类

设备信息 (DeviceInfo)

此类记录了设备的所有配置，包括设备名，URI，USB VID/PID描述符和供应商。

3) 传感器信息 (SensorInfo)

此类存储了传感器的所有配置，这里的传感器仅是三大传感器之一的一个。一个设备有多个传感器。

4) 视频模式 (VideoMode)

此类存储了分辨率，帧率和像素格式。用于视频流的设置和查看设置，由视频帧引用查看这些设置，由传感器信息提供一个视频模式的列表。

5) 摄像头设置 (CameraSetting)

存储了非UVC设备的RGB摄像头的设置，可以启用或禁用自动白平衡和自动曝光。

4.2 扩展API介绍

Orbbec在通用的OpenNI2功能的基础上进行了扩展，是API接口可以实现读取设备序列号、获取设备类型、使用flash保存数据、读取相机标定参数、IR摄像头参数调节、LDP控制等功能。

扩展API在Device类中定义，用户如果需要使用这些扩展功能，在调用扩展API之前，需要先确保OpenNI2 initialize成功，设备打开成功。具体API使用说明如下：

4.2.1 获取设备型号

```
char devType[32];
int dataSize = sizeof(devType);
memset(devType, 0, dataSize);
g_Device.getProperty(openni::OBEXTENSION_ID_DEVICETYPE, (uint8_t*)&devType, &dataSize);
```

4.2.2 获取设备序列号

```
char serNumber[12];
int dataSize = sizeof(serNumber);
memset(serNumber, 0, dataSize);
g_Device.getProperty(openni::OBEXTENSION_ID_SERIALNUMBER, (uint8_t*)&serNumber,
```

```
&dataSize);
```

4.2.3 设置LDP开关

```
int dataSize = 4;
int ldp_en = enable;
g_Device.setProperty(openni::OBEXTENSION_ID_LDP_EN, (uint8_t *)&ldp_en, dataSize);
```

4.2.4 设置LDM开关

```
int dataSize = 4;
int laser_en = enable;
g_Device.setProperty(openni::OBEXTENSION_ID_LASER_EN, (uint8_t *)&laser_en, dataSize);
```

4.2.5 设置和获取IR曝光值

```
int exposure = 0;
int dataSize = 4;
g_device.getProperty(openni::OBEXTENSION_ID_IR_EXP, (uint8_t *)&exposure, &dataSize);
printf("ir exposure value : 0x%x\n", exposure);
exposure += 256;
g_device.setProperty(openni::OBEXTENSION_ID_IR_EXP, (uint8_t *)&exposure, dataSize);
```

4.2.6 设置及获取IR增益

```
int gain = 0;
int dataSize = 4;
g_device.getProperty(openni::OBEXTENSION_ID_IR_GAIN, (uint8_t *)&gain, &dataSize);
printf("ir gain value : 0x%x\n", gain);
gain++;
g_device.setProperty(openni::OBEXTENSION_ID_IR_GAIN, (uint8_t *)&gain, dataSize);
```

4.2.7 设置保存和获取相机标定参数

```
typedef struct OBCameraParams
{
float l_intr_p[4];          //[fx,fy,cx,cy]
float r_intr_p[4];          //[fx,fy,cx,cy]
```

```

float r2l_r[9];           //[r00,r01,r02;r10,r11,r12;r20,r21,r22]
float r2l_t[3];           //[t1,t2,t3]
float k[5];               //[k1,k2,p1,p2,k3]
int is_mirror;
} OBCameraParams;

OBCameraParams m_CamParams = { 0 };
int dataSize = sizeof(OBCameraParams);
m_CamParams.l_intr_p[0] = 577.318970;
m_CamParams.l_intr_p[1] = 577.318970;
m_CamParams.l_intr_p[2] = 308.729004;
m_CamParams.l_intr_p[3] = 269.143005;
m_CamParams.r_intr_p[0] = 517.447998;
m_CamParams.r_intr_p[1] = 517.447998;
m_CamParams.r_intr_p[2] = 305.432007;
m_CamParams.r_intr_p[3] = 250.410995;
m_CamParams.r2l_r[0] = 0.999972;
m_CamParams.r2l_r[1] = -0.005735;
m_CamParams.r2l_r[2] = 0.004735;
m_CamParams.r2l_r[3] = 0.005736;
m_CamParams.r2l_r[4] = 0.999983;
m_CamParams.r2l_r[5] = -0.000298;
m_CamParams.r2l_r[6] = -0.004733;
m_CamParams.r2l_r[7] = 0.000325;
m_CamParams.r2l_r[8] = 0.999989;
m_CamParams.r2l_t[0] = -25.147900;
m_CamParams.r2l_t[1] = 0.015202;
m_CamParams.r2l_t[2] = -0.648167;
m_CamParams.k[0] = -0.077348;
m_CamParams.k[1] = 0.208761;
m_CamParams.k[2] = -0.196780;
m_CamParams.k[3] = 0.000617;
m_CamParams.k[4] = 0.001059;
m_CamParams.is_mirror = 0;

```

```
g_Device.setProperty(openni::OBEXTENSION_ID_CAM_PARAMS, (uint8_t *)&m_CamParams,
dataSize);
g_Device.getProperty(openni::OBEXTENSION_ID_CAM_PARAMS, (uint8_t *)&m_CamParams,
&dataSize);
```