

DM_TCM_320240E_SDK 用户手册

Version: 1.1

2017 Feb



Copyright © 2018 Digital Miracle

目录

DM_TCM_320240E_SDK 用户手册	0
Version: 1.1.....	0
1 SDK 概述	3
1.1 USB 驱动安装	3
1.1.1 Windows	3
1.1.2 Linux.....	3
2 DMCAM C/C++库	3
2.1 DMCAM C 样例说明	3
2.2 Msys2 编译环境说明	4
2.2.1 环境安装	4
2.2.2 编译运行	4
3 DMCAM Python 扩展	6
3.1 Windows 下 python 扩展使用	6
3.2 Linux 下 python 扩展使用	7
4 DMCAM ROS 扩展	9
4.1 概述.....	9
4.2 API 使用说明	10
4.2.1 TofHandle	10
4.2.2 Published Topics	10
4.2.3 API 接口说明.....	10
4.2.4 Service	11
4.3 环境安装及测试.....	11
4.3.1 安装步骤	11
4.3.2 设备权限修改	12
4.3.3 Smart_tof 使用	12
4.3.4 tof_sample 使用	16
5 Java 扩展	16
6 命令行工具	17
6.1 dmcam-cli.....	17
6.1.1 硬件设备信息获取	20

6.1.2	硬件参数设置	20
6.1.3	数据采集	21
6.1.4	固件更新	22
7	GUI 工具	23
7.1	SmartTOFViewer	23
7.1.1	Windows 平台	23
7.1.2	Linux 平台	24
8	修订历史	25

1 SDK 概述

本文档是主要介绍模组的 SDK 安装测试，及基于 SDK 应用开发方法，同时包含模组固件更新，采集数据以及一些注意事项。详细使用说明可参见：

<https://github.com/smarttofsdk/SDK/wiki>。

1.1 USB 驱动安装

1.1.1 Windows

双击 SDK 中 drivers\usb_driver 目录下面的 install_driver.bat 文件，进行 USB 驱动安装。




图 1-1 Windows 下 USB 驱动安装

1.1.2 Linux

安装 libusb-1.0.17 或高于此版本的都可以。Ubuntu 14.04, 16.04 版本默认已安装 libusb，可不用安装。

2 DMCAM C/C++库

DMCAM 接口及使用方法可以参见 dmcam.h 中函数接口说明及样例。

2.1 DMCAM C 样例说明

DMCAM C 库使用样例位于 SDK example/c/test 目录下，主要用来帮助用户了解 dmcam 库的使用方法。

文件名	描述
sample_capture_frames.c	测试 dmcam 层
sample_set_param .c	测试通用参数设置和获取
sample_filter.c	测试功能接口配置

表 2-1 使用样例例列表

2.2 Msys2 编译环境说明

2.2.1 环境安装

A. 在 <http://www.msys2.org/> 下载 `msys2-i686-20161025.exe` :

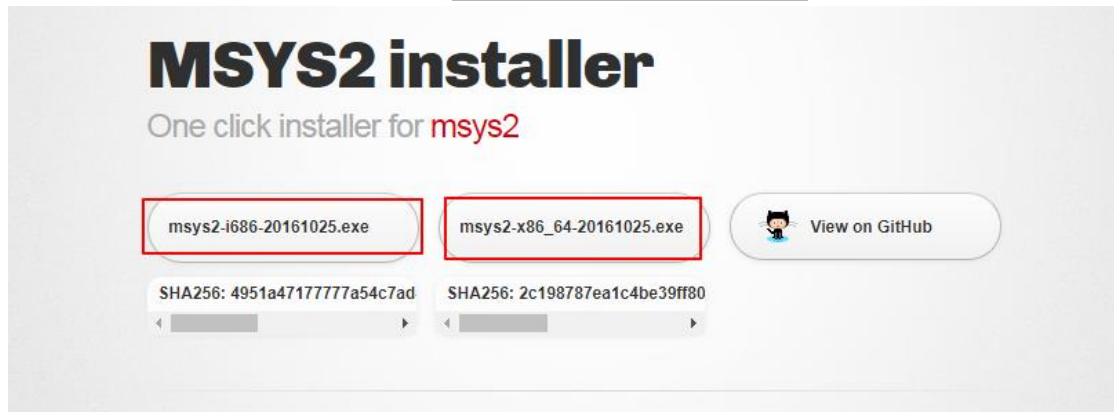


图 2-1 下载 msys2

B. 安装 msys2，根据提示安装硬盘，启动脚本更新及安装对应的包。

2.2.2 编译运行

- 1) 进入 SDK 的 windows/dmcam/samples/c 文件夹下的 CMakeList.txt 所在目录建立 build 文件
- 2) 打开 msys2 进入到对应工程，相应的命令如下

```
1$ cd build
2$ cmake .. -G "MSYS Makefiles" //第一次 cmake 失败可以再尝试一次
3$ make -j //or mingw32-make
```

- 3) 在 build 目录下有 sample 样例生成，运行 ./sample_capture_frames.exe，运行的参考结果如图 2-2。

```
Administrator@DEEP-1704270257 MINGW32 /e/WORK/SVN_ALL/packedSDK/SmartToF_github/SDK-SmartTof-v1.11/SDK-SmartTof-v1.11/windows/dmcam/samples/c/build
$ ./sample_capture_frames.exe
1 dmcam device found
Could not open ./grayscale660_gain_offset_w184_C094.txt file.
gOffsetDRNU_Temp0:9930
gTempCal0:462
gStepCalMM0:281.3000
Can not open epc660_drnu_w184_C094_20.csv file
ambientLightFactor = 250.0000
print = 0
diffTemp = 7
drnuDelay = 200000
numAveragedFrames = 25
epc635_TemperatureCoeff = 1.561310
epc660_TemperatureCoeff = 1.700000
speed of light div 2 = 149896229.00
epc660_images_drnu_w184_C094_10.bin
*** device type and IDs:
chip_type: 27
icType: 7
partVersion: 7
waferID: 184
chipID: 94
sequencerVersion: 9
devAddress: 0x20
currentMode: 0
currentFreq: 1
modulationFrequency: 12000
modulationFrequencies[0]: 24000
modulationFrequencies[1]: 12000
modulationFrequencies[2]: 6000
modulationFrequencies[3]: 3000
modulationFrequencies[4]: 1500
modulationFrequencies[5]: 750
*** runtime settings:
integrationTime2D: 700
integrationTime3D: 700
integrationTime3DHDR: 700
nDCSTOF: 4
absEnabled: 1
isFLIM: 0
*** calibration/correction settings:
correctionBGMode: 0
enabledDRNU: 1
enabledDRNU2: 1
enabledGrayscale: 0
enabledTemperatureCorrection: 0
enabledAmbientLightCorrection: 0
*** Print level:
gPrint: 0
*** trimming values for temperature readout level:
gX: 0x10
gY: 0x39
gM: 52
gC: 141
gL: 0
gZ: -268
width = 320, height = 120
2018/02/26 14:39:16.300[5904][INF] [dmcam_dev_open] Load calibration data
Find epc660_images_drnu_w184_C094_10.bin
Could not open ./grayscale660_gain_offset_w184_C094.txt file.
gOffsetDRNU_Temp0:9930
gTempCal0:462
```

图 2-2 win32 使用样例执行

3 DMCAM Python 扩展

3.1 Windows 下 python 扩展使用

DMCAM python windows 扩展支持 python 2.7/3.4/3.5/3.6 的 32bit&64 位版本。运行 python 使用样例的步骤如下：

- 1) 安装 python: 从 <https://www.python.org/> 下载安装 python 32bit (2.7/3.4/3.5/3.6)。
- 2) 安装 dmcam python 扩展:

- 进入 SDK 的 windows/dmcam/python 目录
- 执行如下命令(XX 为 python 对应版本):

```
pip install dmcam_0.3.0-cpXX-cpXXm-win32.whl
```

- 以 python3.4 32 位版本为例，执行后结果如图 3-1。

```
(pyenv34_win32) E:\WORK\SVN_ALL\packedSDK\SmartToF_gitrelease\SDK-SmartToF-v1.11\SDK-SmartToF-v1.11\windows\dmcam\python>pip install -U dmcam-0.3.0-cp34-cp34m-win32.whl
Processing e:\work\svn_all\packedsdk\smartrtoF_gitrelease\sdk-smartrtoF-v1.11\sdk-smartrtoF-v1.11\windows\dmcam\python\dmcam-0.3.0-cp34-cp34m-win32.whl
Installing collected packages: dmcam
Found existing installation: dmcam 0.3.0
Uninstalling dmcam-0.3.0:
Successfully uninstalled dmcam-0.3.0
Successfully installed dmcam-0.3.0
```

图 3-1 安装运行库

- 3) 安装 Python Sample 依赖库包:
- 进入 SDK 的 windows/dmcam/python 目录
 - 执行如下命令安装依赖库:

```
pip install -r requirement.txt
```

- 4) 运行 Python Sample: test_param.py 运行结果如图 3-2 所示。

```
(pyenv34_win32) E:\WORK\SVN_ALL\packedSDK\SmartToF_gitrelease\SDK-SmartToF-v1.11\SDK-SmartToF-v1.11\windows\dmcam\samples\python>python test_param.py
Open dmcam device .
Could not open ./grayscale660_gain_offset_W184_C094.txt file.
Can not open epc660_drnu_W184_C094_20.csv file
Could not open config.txt file.
ambientLightFactor= 250.000000
printVal= 0
diffTemp= 7
drnuDelay= 200000
numDRNUAveragedFrames= 25
epc635_TemperatureCoeff= 1.5613
epc660_TemperatureCoeff= 1.7000
speed of light div 2= 149896229.00
Can not open epc660_images_drnu_W184_C094_10.bin file
*** device type and IDs:
chip_type: 27
icType: 7
partVersion: 7
waferID: 184
chipID: 94
sequencerVersion: 9
devAddress: 0x20
currentMode: 0
currentFreq: 1
modulationFrequency: 12000
modulationFrequencies[0]: 24000
modulationFrequencies[1]: 12000
modulationFrequencies[2]: 6000
modulationFrequencies[3]: 3000
modulationFrequencies[4]: 1500
modulationFrequencies[5]: 750
*** runtime settings:
integrationTime2D: 700
integrationTime3D: 700
integrationTime3DHDR: 700
mDCSTOF: 4
absEnabled: 1
isFLIM: 0
*** calibration/correction settings:
correctionBGMode: 0
enabledDRNU: 0
enabledDRNU2: 0
enabledGrayscale: 0
enabledTemperatureCorrection: 0
enabledAmbientLightCorrection: 0
*** Print level:
gPrint: 0
*** trimming values for temperature readout level:
gX: 0x10
gY: 0x39
gM: 52
gC: 141
gL: 0
gZ: -268
width = 320, height = 120
```

图 3-2 执行使用样例

3.2 Linux 下 python 扩展使用

DMCAM python Linux 扩展支持 python 2.7/3.4/3.5 的 64bit 版本。运行 python 使用样例的步骤如下：

1) 安装 dmcam python 扩展：

- 进入 SDK 的 linux/dmcam/python 目录
- 执行如下命令(XX 为 python 对应版本)

```
pip install dmcam_0.3.0-cpXX-cpXXm-linux_x86_64.whl
```

2) 安装 Python Sample 依赖库包：

- 进入 SDK 的 linux/dmcam/samples 目录
- 执行如下命令安装依赖库

```
pip install -r requirement.txt
```

3) 运行 PythonSample：

- test_gui_pygame.py 运行结果如图 3-3 所示。

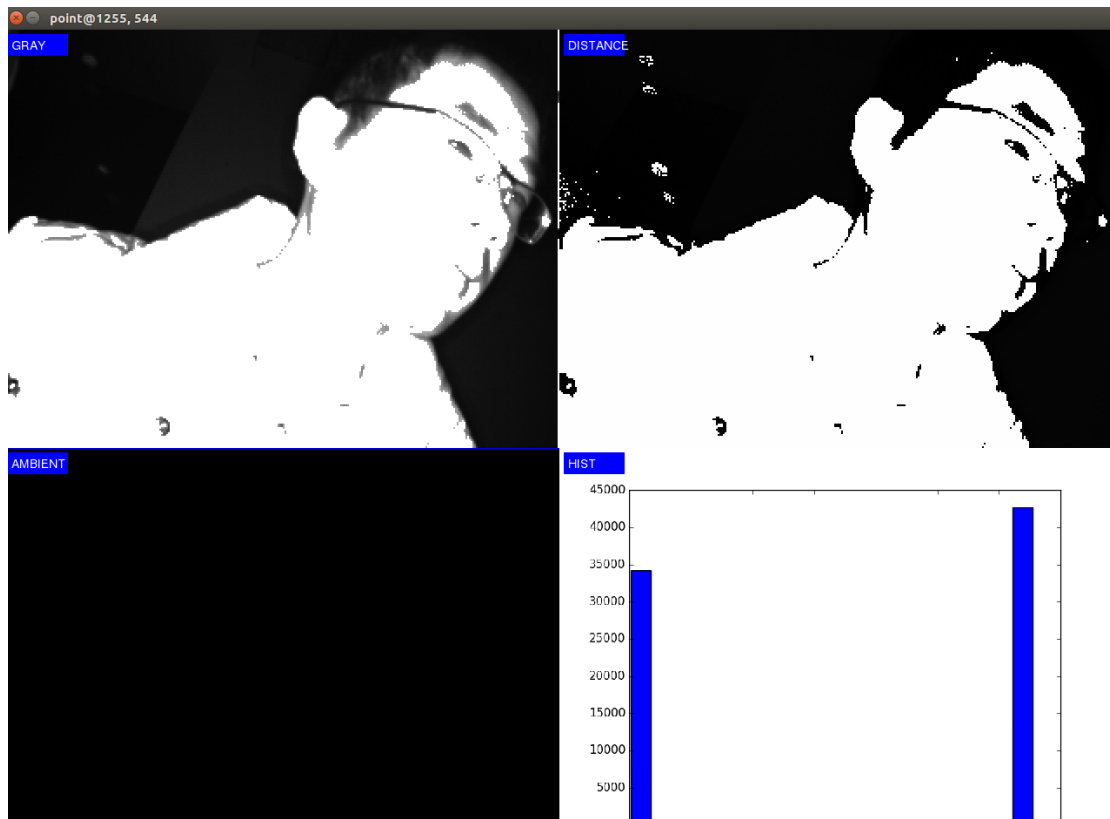


图 3-3 test_gui_pygame 运行截图

- test_gui_pyQtGraph.py 运行结果如图 3-4 所示。

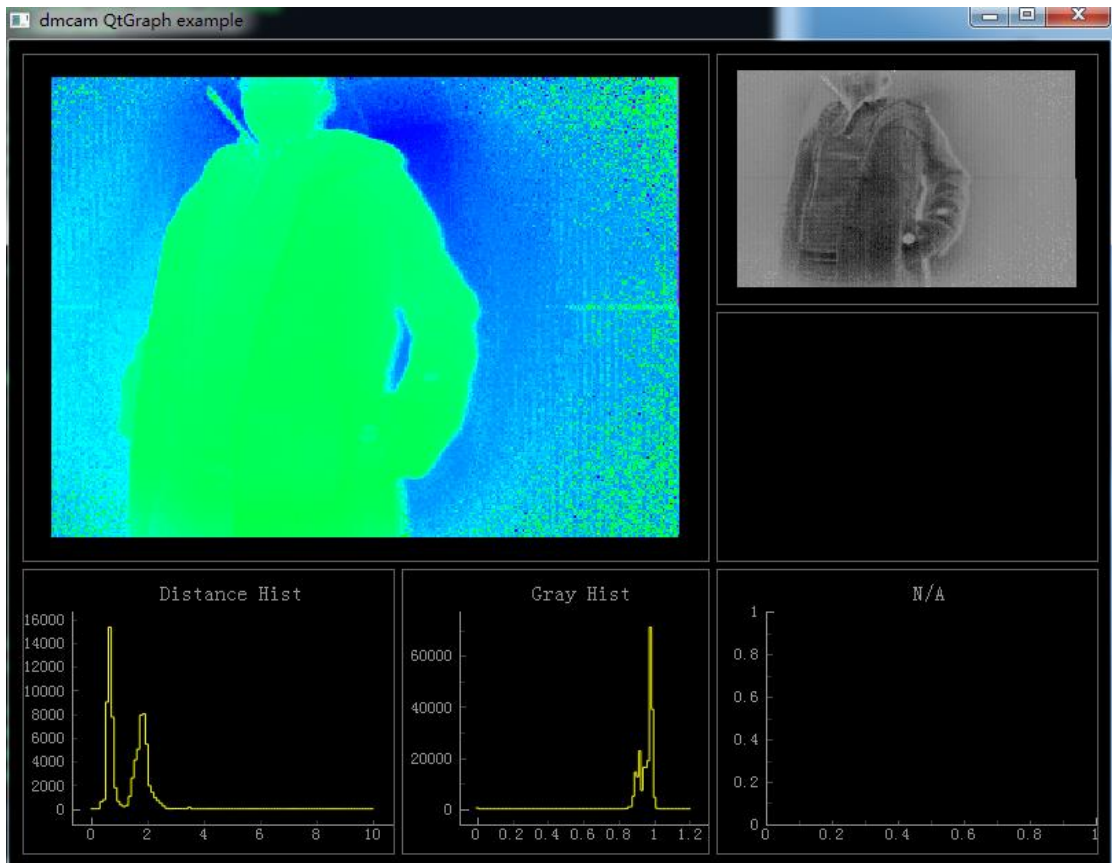


图 3-4 test_gui_ pyQtGraph 运行截图

注：如是 ubuntu14.04 的系统 python2.7 安装 pygame 时可能会出现 `sdl-config not found` 的问题，可使用 `apt-get install python-pygame` 来安装。如提示 `matplotlib not found`，可使用 `apt-get install python-matplotlib` 来安装。

4 DMCAM ROS 扩展

4.1 概述

linux\dmcam\samples\ros 目录下是 ROS 对于模组的 package 封装，package 内有一个 node:TofHandle，三个 topic,如表 4-1:

表 4-1 Topic 说明

Topic 名称	描述
/smarttof/camera_info	相机参数

/smarttof/image_gray	灰度图
/smarttof/image_dist	深度

4.2 API 使用说明

4.2.1 TofHandle

此 node 即为 Package 的主线程

4.2.2 Published Topics

对于 TOF 深度相机，总共有四个 topic，取用及显示他们的方法如下：

- topic1./smarttof/cameraInfo, 命令如下

```
rostopic echo /smarttof/camera_info
```

- topic2./smarttof/image_gray, 命令如下

```
roslaunch image_view image_view image:=/smarttof/image_gray
```

- topic3./smarttof/image_dis, 命令如下

```
roslaunch image_view image_view image:=/smarttof/image_dist
```

4.2.3 API 接口说明

表 4-2 Public 函数列表

函数名	功能
TofHandle(ros::NodeHandle *_n);	TofHandle 构造函数
pub_all()	发布所有的 topic
get_tof_state()	标志 TOF 模组是否初始化好

表 4-3 Private 函数列表

函数名	功能
get_one_frame()	从 TOF 模组中取一帧图片
tof_init()	初始化 TOF 模组
param_init(dmcam_dev_t* dev_0)	参数初始化
topic_init()	Topic 初始化
sudodev(dmcam_dev_t *_dev)	获得 TOF 模组设备的 sudo 权限

表 4-4 变量名列表

变量名	功能
-----	----

NEEDSUDO	该 package 是否取得设备的 sudo 权限
ros::NodeHandle *nh	主线程的 handle
dmcam_dev_t *dev	所打开的 TOF 模组设备
bool tof_state	TOF 设备是否初始化好
uint32_t dev_mode	设置相机模态
uint32_t mod_freq	模态频率
dmcam_roi_t roi	ROI 结构体
uint32_t format	帧格式
uint8_t power_percent	发光功率百分比
uint32_t fps	帧率
uint16_t intg_us	积分时间

4.2.4 Service

利用 rosservice 的形式，实现命令行修改参数。

4.3 环境安装及测试

4.3.1 安装步骤

- 安装 ubuntu16.04 系统
- 安装 ROS，打开命令行终端，进入 package 所在文件夹，运行一下命令：

```
sudo chmod 755 install_ros.sh
```

```
./install_ros.sh
```

- 编译包，执行如下命令，编译 package

```
make
```

- 进入 devel 目录，执行如下命令

```
source setup.sh
```

此 ROS 扩展是在 ubuntu16.04 系统下安装使用，支持 kinetic&indigo，推荐安装 kinetic

4.3.2 设备权限修改

在运行 launch 文件时候，会出现打开设备失败的情形，是因为 udev 的权限导致，可以通过下面方法解决。

- 1) 创建一个新的 udev 规则:

```
sudo vi /etc/udev/rules.d/99-persistent-usb.rules
SUBSYSTEMS=="usb",ATTRS{idVendor}=="111b",ATTRS{idProduct}=="1238",OWNER="username",GROUP="users", MODE="0666"
```

其中"idVendor"和"idProduct"是根据 TOF 设备实际"idVendor"和"idProduct"填写，MODE="0666"表示 USB 设备的权限；99 可以根据电脑上其他文件一致即可。

- 2) 重启电脑并重新加载 udev 规则:

```
sudo udevadm control --reload
```

- 3) 将模组和电脑连接，查看设备，执行如下命令，结果如图 4-1:

```
lsusb
```

```
i$ lsusb
Bus 001 Device 004: ID 111b:1238
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

图 4-1 查看 USB 设备

4.3.3 Smart_tof 使用

- 1) 开启 ROS 环境，命令如下;

```
roscore
```

- 2) 修改相机参数，通过 parameters/init_param.yaml 文件修改初始参数。
- 3) 运行 launch 文件

```
roslaunch smart_tof start.launch
```

- 4) 显示深度图像结果如图 4-2，命令如下。

```
roslaunch image_view image_view image:=/smarttof/image_dist
```

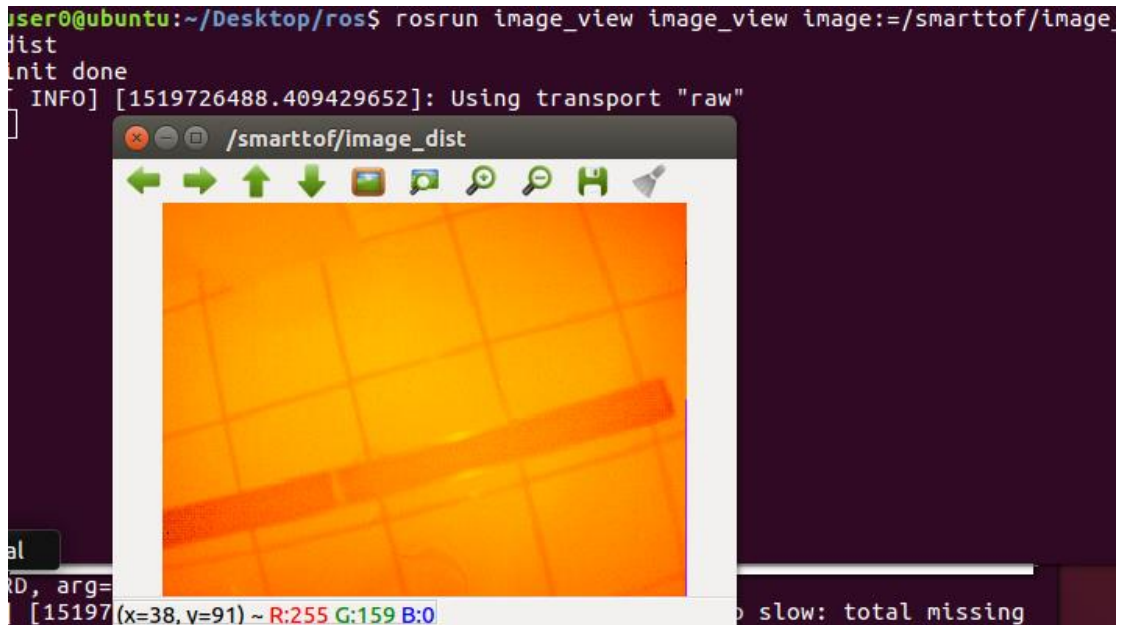


图 4-2 显示深度图

5) 显示点云，步骤如下：

- 在命令行中输入下面命令，显示结果如图 4-3：

```
rosrun rviz rviz
```

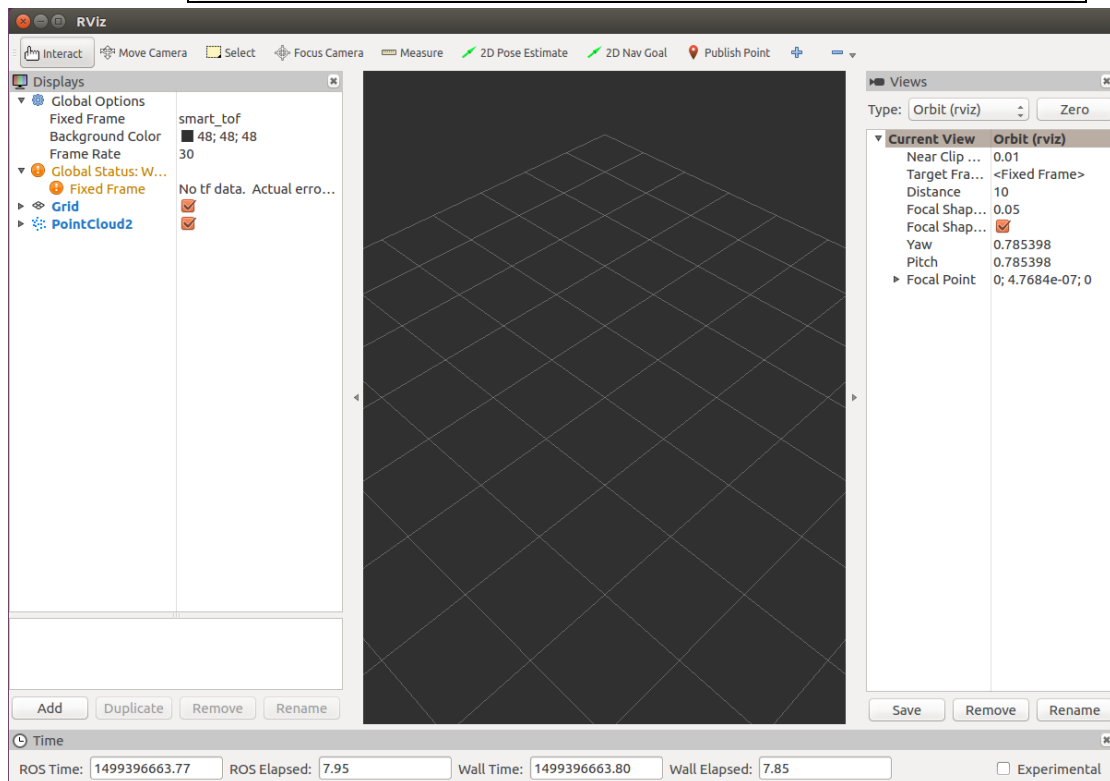


图 4-3 启动 RVIZ

- 点击 Rviz 界面左下角的 Add 按钮，如图 4-4：

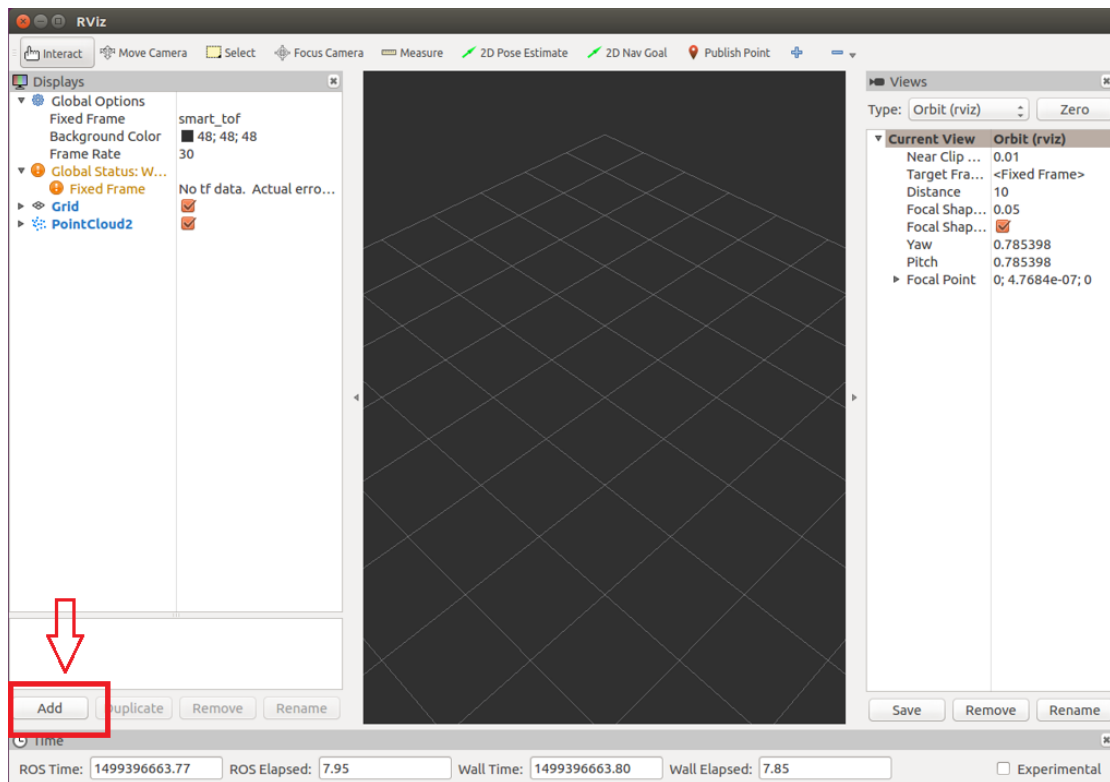
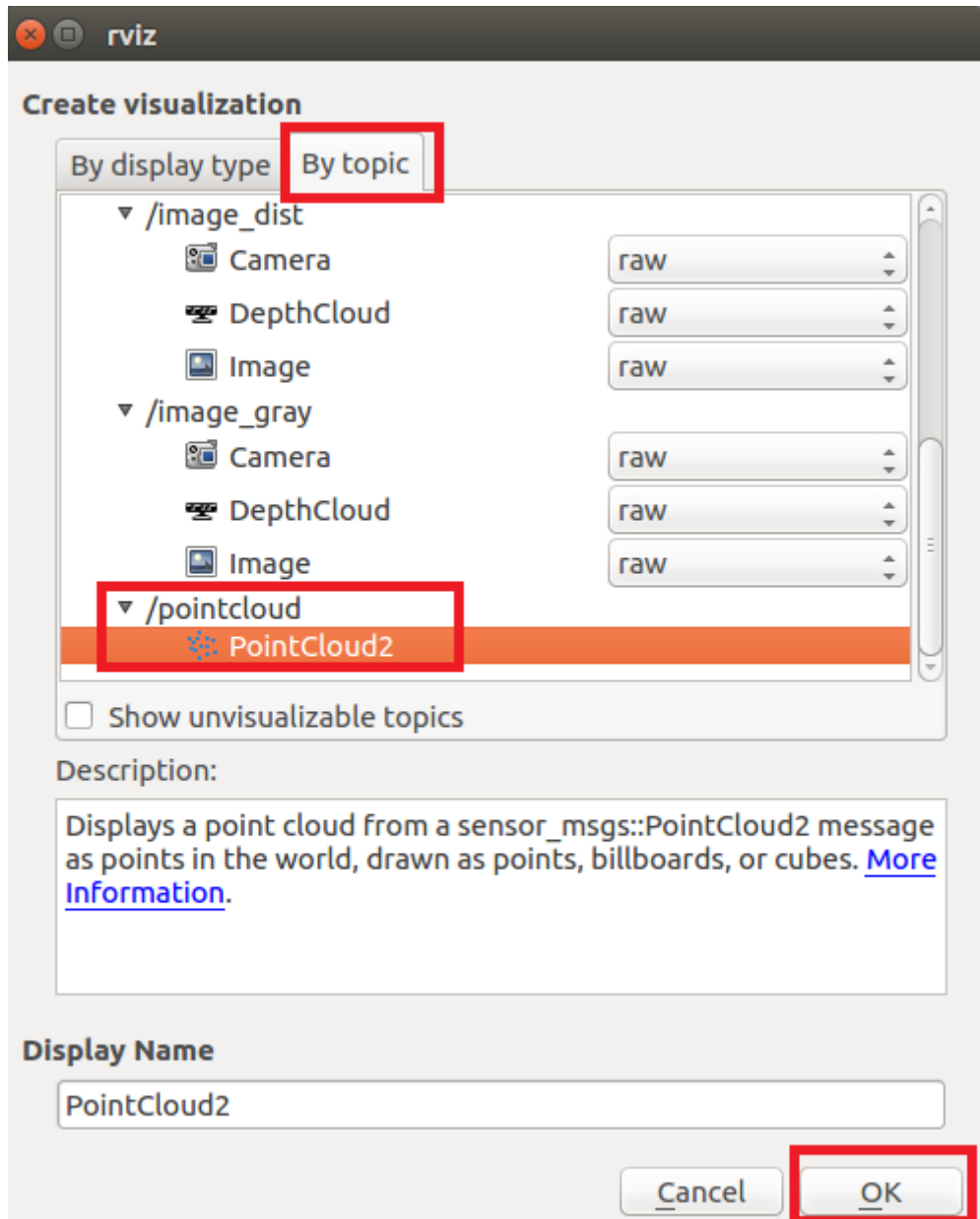


图 4-4 添加 topic

- 出现以下对话框, 选择 By topic 按钮, 将滚动条拉到底部, 选择 PointCloud2, 点击 ok:



- 在 Rviz 界面的左上角，选择 Global Option/fixed frame，在对应的框中输入 smart_tof，如图 4-5 下所示：

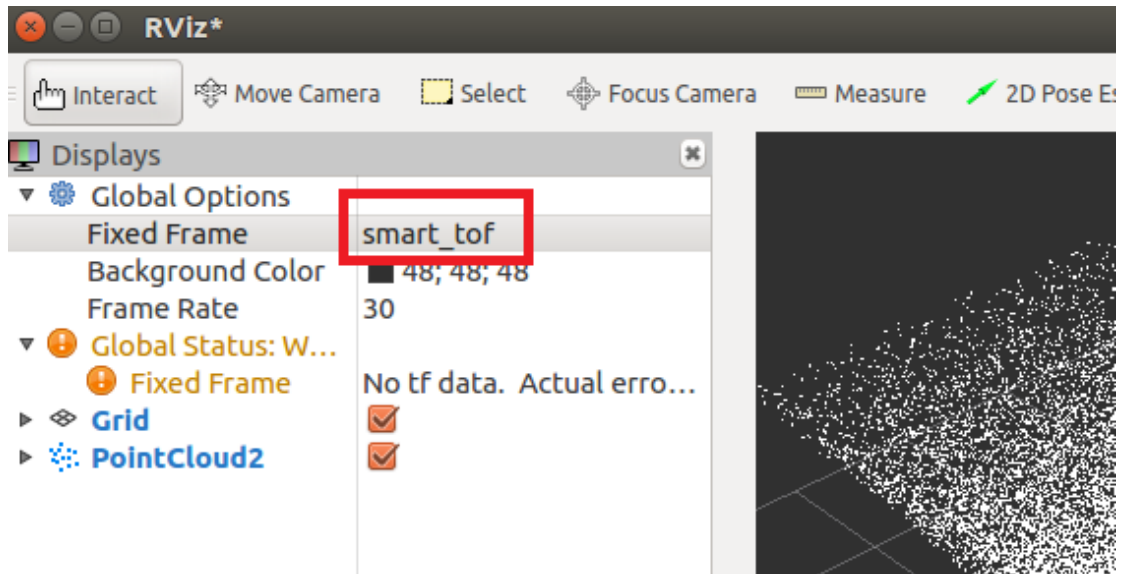


图 4-5 添加 smart_tof

6) 动态配置参数:

```
rosservice call /smartof/change_power []
rosservice call /smartof/change_power []
```

4.3.4 tof_sample 使用

1) 开启 ros 环境:

```
roscore
```

2) 运行 launch 文件:

```
roslaunch tof_sample start.launch
```

5 Java 扩展

在 windows/samples/java 目录下面, 有 java 测试的例子, 可参见 readme.txt。



图 6 JAVA 测试例子

6 命令行工具

命令行工具相关文件说明:

- dmcam 动态链接库。
- dmcam-cli: 命令行工具 (cli: Command Line Interface)

若 linux 下运行 dmcam-cli 提示找不到 dmcam.so, 可将 dmcam.so 拷贝至系统目录/usr/lib/下或者通过 LD_LIBRARY_PATH 指定 so 的路径

6.1 dmcam-cli

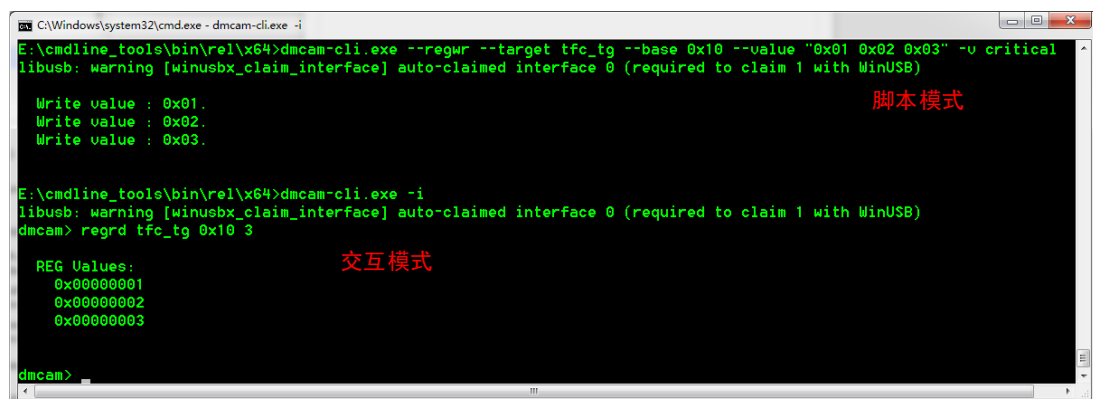
dmcam-cli 可以通过命令行与硬件设备进行交互, 主要功能包括:

- 硬件设备信息获取
- 硬件参数设置
- 数据采集
- 固件更新

dmcam-cli 可以通过以下三种方式使用:

- 命令行参数方式: 具体参数定义参见表 6-1
- 脚本文件方式: 参见表 6-1 中 '`-s, --script <file>`' 选项。
- 交互模式: 参见表 6-1 中 '`-i, --interactive`' 选项。

图 6-1 是脚本命令参数和交互模式的参考使用截图。



```
C:\Windows\system32\cmd.exe - dmcam-cli.exe -i
E:\cmdline_tools\bin\rel\x64>dmcam-cli.exe --regwr --target tfc_tg --base 0x10 --value "0x01 0x02 0x03" -v critical
libusb: warning [winusb_claim_interface] auto-claimed interface 0 (required to claim 1 with WinUSB)

Write value : 0x01.
Write value : 0x02.
Write value : 0x03.

E:\cmdline_tools\bin\rel\x64>dmcam-cli.exe -i
libusb: warning [winusb_claim_interface] auto-claimed interface 0 (required to claim 1 with WinUSB)
dmcam> regwr tfc_tg 0x10 3

REG Values:
0x00000001
0x00000002
0x00000003

dmcam>
```

图 6-1 脚本模式&交互模式参考使用截图

表 5-1 展示了各命令参数及使用样例。更详细的说明可输入 **--help** 和 **--help-interactive** 参数进行查看。

表 6-1 命令列表

Script mode functions	options	appended options	example
指定设备	-d, --device <device>		dmcam-cli.exe --print info --device 0
set verbosity	-v, --verbosity <level>		dmcam-cli.exe --print info --device 0 -v critical
枚举设备	-l, --list		dmcam-cli.exe -l
复位设备	-r, --reset		dmcam-cli.exe -r
执行 interactive mode command, 可多次使用	-e, --exec <command>		dmcam-cli.exe -e "print info" -e "print frame_format"
Run provided script	-s, --script <file>		dmcam-cli.exe -s script.txt
进入 interactive mode	-i, --interactive		dmcam-cli.exe -i
print cli version	--version		dmcam-cli.exe --version
print cli help info	-h, --help		dmcam-cli.exe --help
show interactive mode help info	--help-interactive		dmcam-cli.exe --help-interactive
固件升级	-f, --flash-MCU-firmware <file>	--version=<ver>	dmcam-cli.exe --flash-MCU fw_stm32.bin --version=111
	-T, --flash-TFC-firmware <file>	--version=<ver>	dmcam-cli.exe --flash-TFC opt_fw_tie_v2.0.bin --version=112
写寄存器	--regwr	--target <target>	dmcam-cli.exe --regwr --target tfc_tg --base 0x100 --value 0xaa
		--base <base>	dmcam-cli.exe --regwr --target tfc_tg --base 0x100 --value "0xaa 0xbb 0xcc"
		--value <val>	dmcam-cli.exe --regwr --target tfc_tg --base 0x100 --value test.bin
读寄存器	--regrd	--target <target>	dmcam-cli.exe --regrd --target tfc_tg --base 0x100
		--base <base>	dmcam-cli.exe --regrd --target tfc_tg --base 0x100

Script mode functions	options	appended options	example
			--cnt 5
			dmcam-cli.exe --regrd --target tfc_tg --base 0x100 --cnt 5 --value test1.bin
写参数	--set <param>	[--param <param>] --value <val>	dmcam-cli.exe --set mode --value 1
读参数	--print <param>	[--param <param>]	dmcam-cli.exe --print mode
交互模式			
固件升级	flash <target> <version>		flash mcu fw_stm32.bin 109
			flash tfc opt_fw_tie_v2.0.bin 111
写寄存器	regwr <target> <base> [<&file> <P0> <P1>... <P4n>]		regwr tfc_de 0x10 0x11 0x12 13
			regwr tfc_de 0x10 test.bin
			regwr mcu 0x11 test.bin
			regwr ir_1 0x12 test.bin
读寄存器	regrd <parameter> <base> [cnt] [&file]		regrd tfc_de 0x10 3
			regrd tfc_de 0x10 test2.bin
			regrd ir_1 0x01 5
			regrd ir_2 0x01 5
			regrd mcu 0x01 5
			regrd tfc_tg 1 50
写参数	set <parameter> <arguments>		set mode 1
			set frequency 1
			set roi 1 2 3 4 5 6
			set frame format 1
			set illum_power 50
			set frame rate 30
			set intg_time 30
读参数	print/p [parameter]		print
			print info
			print mode
			print frequency
			print format
			print frame rate
			print roi

Script mode functions	options	appended options	example
采集固定数量的 frame 到文件	rx <file name> <frame buffer size in MB> [<frame count>]		rx frame.bin 2
			rx frame.bin 2 5
采集固定数量数据到 buffer	read <frame count>		read 5
同 print info	info		info
显示所有 version 信息	version		version
others			cls
			quit
			help
			h rx
			echo who am i

6.1.1 硬件设备信息获取

当设备连接后，可以通过 dmcam-cli 交互模式的 print 命令进行硬件设备信息获取，命令格式如下（参考使用截图参见图 6-2）。

p [parameter]

```

dmcam> p info

Vendor   : Digital Miracle
Product  : TM-E1-2.0

Capability:
max frame width      : 320
max frame height     : 240
max frame depth      : 2
max fps              : 60
max integrate time(us) : 333

Serial info: 0x17013028, 0xAA6D94A9, 0x582CB544

DFU Information:
MCU Firmware Version : 108
TFC Firmware Version  : 0
  
```

图 6-2 交互模式设备信息获取参考使用截图

6.1.2 硬件参数设置

当设备连接后，可以通过 dmcam-cli 交互模式的 set 命令设置硬件参数，命令格式如下（参考使用截图参见图 6-3）。

set <parameter> <arguments>

```

dmcam> h s

Usage: set <parameter> <arguments>

The set command takes a parameter and an arbitrary number of arguments
for that particular parameter. The parameter is one of:

-----
Parameter Description
-----
mode Device Mode settings

frequency Frequency settings

roi ROI of the device.
    In format: "srow scol erow ecol cur_fsize max_fsize".

frame_format Frame format of the device.

illum_power Illuminate power of the device

frame_rate Frame Rate n
    intg_time integrate time.
-----

dmcam> set intg_time 300

Set integrate time :300 us

dmcam>

```

图 6-3 交互模式设置设备参数参考使用截图

6.1.3 数据采集

当设备连接后，可以通过 dmcam-cli 交互模式的 rx 命令进行数据采集，并将数据存入指定文件，命令格式如下（参考使用截图参见图 6-4）。

```
rx <&file name> <frame buffer size in MB> [<frame count>]
```

```

C:\Windows\system32\cmd.exe - dmcam-cli.exe -i
E:\cmdline_tools\bin\rel\x64>dmcam-cli.exe -i
libusb: warning [winusb_claim_interface] auto-claimed interface 0 (required to claim 1 with WinUSB)
dmcam> h rx

Usage: rx <&file> <buffer size> <frame count>

Receive frames and write them to the specified file.
Below is a list of available parameters.

-----
Parameter Description
-----

file Filename to write received samples to

buffers size Max size of sample buffers to be use in the receive
              stream. The unit is MB.

frame count Number of samples to receive.
-----

dmcam> rx frame.bin 2 5
.....get 5 frames: 1st frame @ [1, 307200, 0]

dmcam>

```

图 6-4 交互模式数据采集参考使用截图

6.1.4 固件更新

注：固件更新方法请与销售代表联系。

7 GUI 工具

7.1 SmartTOFViewer

7.1.1 Windows 平台

- 1) 设备连接：确保设备通过 USB 与 PC 连接。
- 2) 执行 GUI：在 SDK 的 windows\tools 目录下面，双击“smartTOFViewer.exe”，（GUI 打开后的参考截图如图 7-1）。1
- 3) 点击开始按钮进行数据采集（参考截图如图 7-2 所示）。
- 4) 可通过选择“视图类型”，“最近距离”，“最远距离”“最小幅值”切换深度或灰度图的可视效果
- 5) 可通过“积分时间”滑动条调整采集参数



图 7-1 smartTOFViewer 打开效果(Windows)

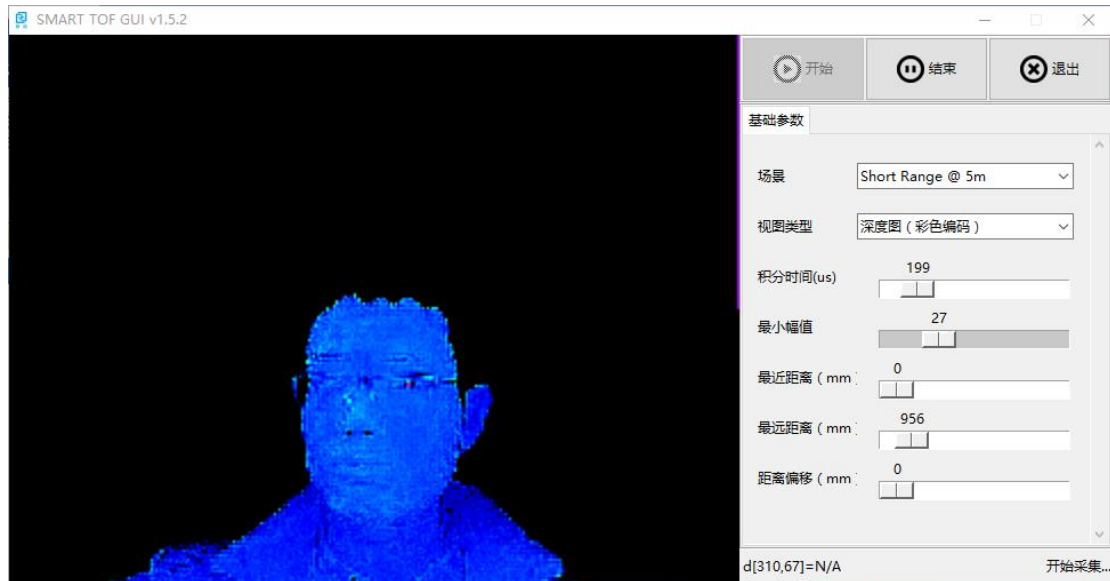


图 7-2 smartTOFViewer 数据采集效果(Windows)

7.1.2 Linux 平台

- 1) 设备连接：确保设备通过 USB 与 PC 连接。
- 2) 执行 GUI：在 SDK 的 bin\gui 目录下面，执行“./smartTOFViewer”，（GUI 打开后的参考截图如图 7-3）。
- 3) 点击开始按钮进行数据采集（参考截图如图 7-4 所示）。
- 4) 可通过选择“视图类型”，“最近距离”，“最远距离”切换深度或灰度图的可视效果
- 5) 可通过“积分时间”及“发光功率”滑动条调整采集参数

若 smartTOFViewer 文件不可执行，可使用命令更改权限:

`chmod a+x smartToFViewers`



图 7-3 smartTOFViewer 打开效果(Linux)

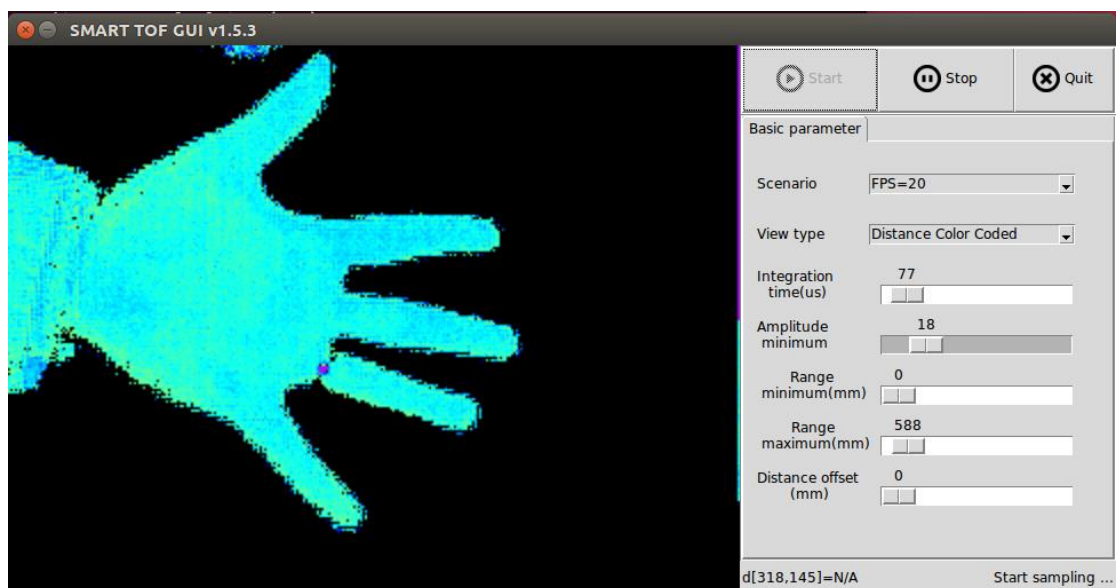


图 7-4 smartTOFViewer 数据采集效果(Linux)

8 修订历史

版本	日期	描述
1.0	12/22/2017	发布 1.0 版本
1.1	2/27/2018	发布 1.1 版本，更新 GUI 图片，