

# SmartTof SDK ROS 用户手册

Version: 1.0

**2018 APRIL**



---

Copyright © 2018 Digital Miracle

## 目录

SmartTof SDK ROS 用户手册 .....	0
Version: 1.0.....	0
1 DMCAM ROS 简介 .....	3
1.1 概述.....	3
2 DMCAM ROS 主要使用流程 .....	4
2.1 DMCAM ROS 使用前准备.....	4
2.2 测试深度数据和灰度数据.....	4
2.3 动态修改参数.....	5
3 cloud_viewer 样例说明.....	5
3.1 cloud_viewer 简介 .....	5
3.2 显示点云数据.....	5
4 API 接口说明 .....	7
4.1 dmcam_ros 发布的话题.....	7
4.1.1 /smarttof/image_dist .....	7
4.1.2 /smarttof/image_gray .....	7
4.1.3 /smarttof/camera_info.....	7
4.1.4 /smarttof/pointcloud.....	7
4.2 dmcam_ros 发布的服务.....	7
4.2.1 /smarttof/change_power .....	7
4.2.2 /smarttof/change_intg .....	7
4.2.3 /smarttof/change_mod_freq .....	8
4.2.4 /smarttof/change_frame_rate .....	8
4.2.5 /smarttof/change_sync_delay.....	8
4.2.6 /smarttof/change_filter .....	8
4.2.7 /smarttof/disable_filter .....	9
5 SDK ROS 系统安装及环境配置.....	9
5.1 ROS 系统安装准备工作 .....	9
5.1.1 ROS 支持的平台.....	9
5.1.2 安装前准备工作 .....	10
5.2 Ubuntu 系统下快速安装（推荐使用） .....	10
5.3 Ubuntu 系统下命令行安装.....	11

---

5.4	ROS 环境配置及 DMCAM ROS 编译.....	12
5.4.1	ROS 环境配置.....	12
5.4.2	DMCAM ROS 系统编译.....	12
6	RVIZ 显示图像、点云数据 .....	12
6.1	rviz 简介 .....	12
6.2	rviz 使用前准备 .....	13
6.3	rviz 显示深度图像 .....	13
6.4	Rviz 显示点云图像 .....	14
附录	.....	16
设备权限修改 .....		16
创建一个新的 udev 规则:.....		16

# 1 DMCAM ROS 简介

## 1.1 概述

DMCAM ROS 是基于 ROS 系统对 DMCAM API 的封装，具有如下所列功能：

- 深度数据和灰度数据采集与显示
- 动态修改参数
- 点云数据采集与显示

DMCAM ROS 基于 ROS 系统，创建 `dmcam_ros` 和 `cloud_viewer` 两个包，`dmcam_ros` 包用来采集与显示深度、灰度数据和动态修改参数，`cloud_viewer` 包是一个示例，用来显示点云数据，组织架构图如下图 1-1 所示：

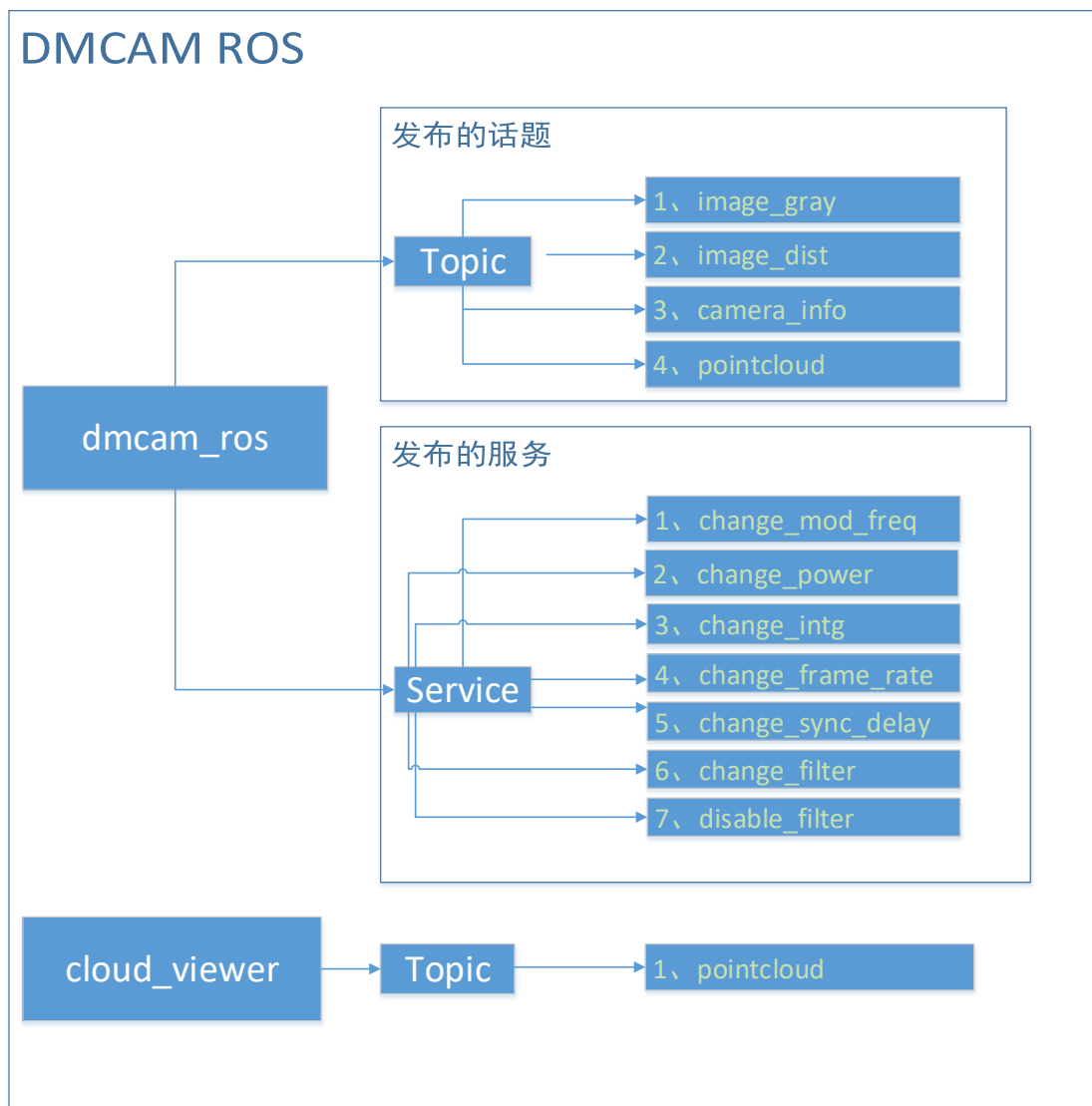


图 1-1 DMCAM ROS 架构

## 2 DMCAM ROS 主要使用流程

### 2.1 DMCAM ROS 使用前准备

DMCAM ROS 在使用前需要先安装 Ubuntu 系统和 ROS 系统，详细安装参阅第 5 章

### 2.2 测试深度数据和灰度数据

- 1) 开启 ROS 环境:

```
roscore&
```

- 2) 进入 ros 所在文件夹初始化环境变量

```
source ./devel/setup.bash
```

- 3) 运行 launch 文件

```
roslaunch dmcam_ros start.launch
```

- 4) 显示深度图像结果如图 2-1，命令如下:

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

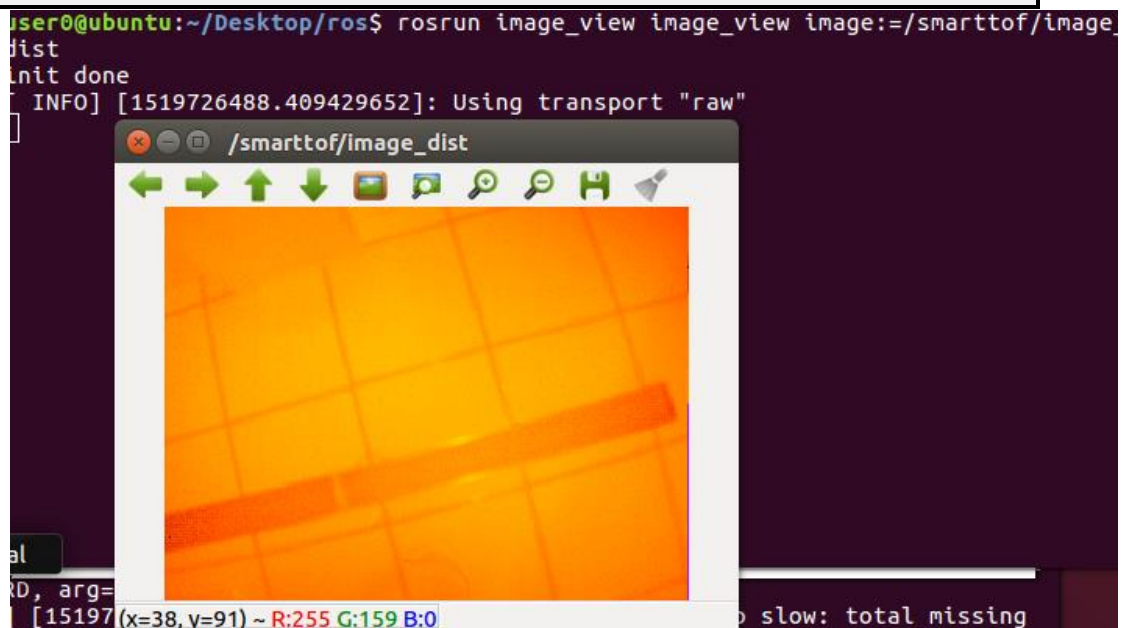


图 2-1 显示深度图

- 5) 显示灰度图像结果如图 2-2，命令如下。

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

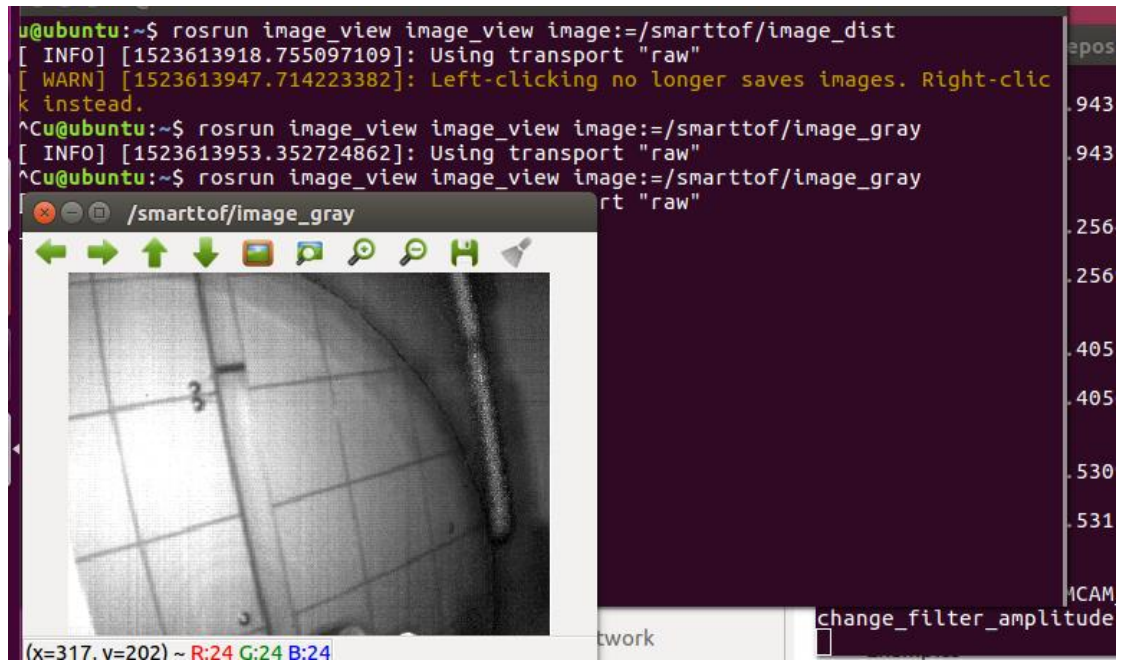


图 2-2 显示灰度图

## 2.3 动态修改参数

开启一个滤波功能，如 DMCAM\_FILTER\_ID\_AUTO\_INTG

```
rosservice call /smarttof/change_filter "filter_id:
'DMCAM_FILTER_ID_AUTO_INTG'
filter_value: 0"
```

关闭一个滤波功能，如 DMCAM\_FILTER\_ID\_AUTO\_INTG

```
rosservice call /smarttof/disable_filter "filter_id:
'DMCAM_FILTER_ID_AUTO_INTG'"
```

## 3 cloud\_viewer 样例说明

### 3.1 cloud\_viewer 简介

cloud\_viewer 是一个简单的使用 dmcam\_ros 来显示点云数据的样例，这个样例简单的实现了怎么从 dmcam\_ros 发布的话题 pointcloud 中获取点云数据并显示出来。

### 3.2 显示点云数据

- 1) 开启 ROS 环境：

```
roscore&
```

- 1) 进入 ros 所在文件夹初始化环境变量

```
source ./devel/setup.bash
```

2) 运行 launch 文件

```
roslaunch cloud_viewer start.launch
```

3) 显示点云图像如图 3-1

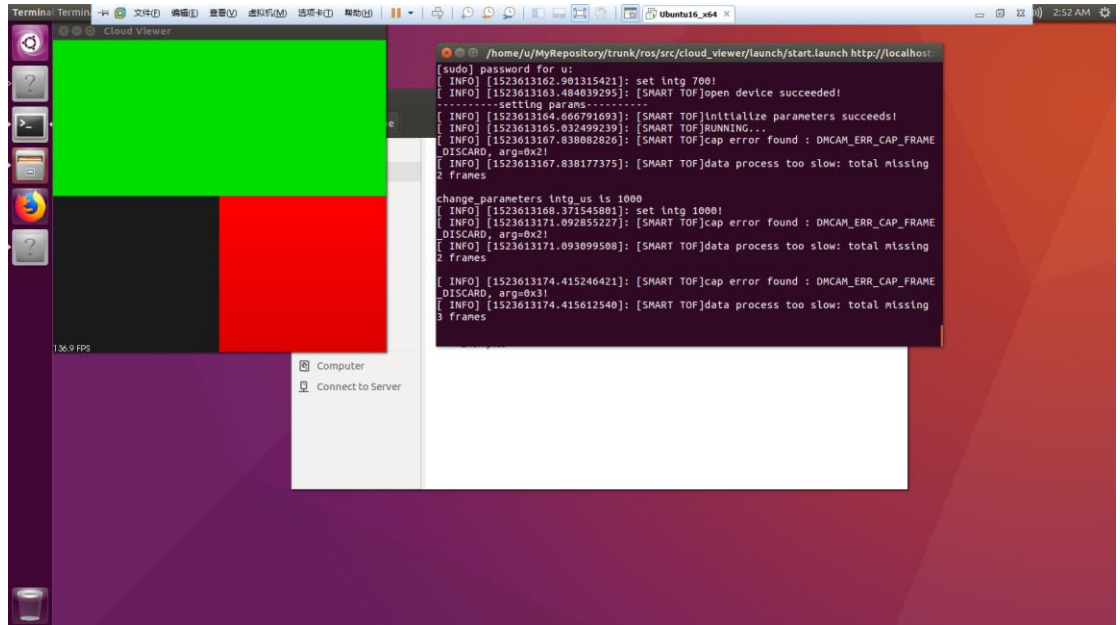


图 3-1 点云显示

4) 通过鼠标中间的滑轮和鼠标左键调整点云显示图像，最终效果如图 3-2

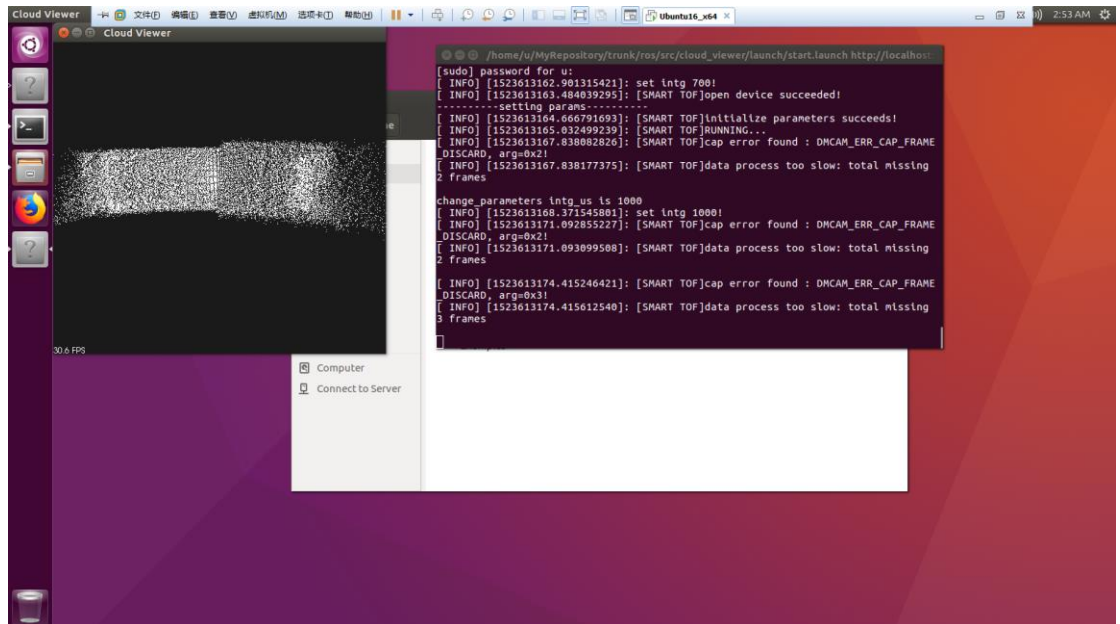


图 3-2 点云显示效果

## 4 API 接口说明

### 4.1 dmcam\_ros 发布的话题

#### 4.1.1 /smarttof/image\_dist

使用命令	<code>roslaunch image_view image_view image:=/smarttof/image_dist</code>
功能描述	从 image_dist 发布的话题中获取深度数据

#### 4.1.2 /smarttof/image\_gray

使用命令	<code>roslaunch image_view image_view image:=/smarttof/image_gray</code>
功能描述	从 image_gray 发布的话题中获取灰度数据

#### 4.1.3 /smarttof/camera\_info

使用命令	<code>rostopic echo /smarttof/camera_info</code>
功能描述	从 camera_info 发布的话题中打印摄像头的信息

#### 4.1.4 /smarttof/pointcloud

使用命令	<code>rviz</code>
功能描述	从 rviz 中显示通过 pointcloud 发布的话题中的点云数据

### 4.2 dmcam\_ros 发布的服务

#### 4.2.1 /smarttof/change\_power

使用命令	<code>rosservice call /smarttof/change_power "power_value: 0"</code>
功能描述	动态修改 PARAM_ILLUM_POWER 的值，
函数参数	保留

#### 4.2.2 /smarttof/change\_intg

使用命令	<code>rosservice call /smarttof/change_intg "intg_value: 0"</code>
功能描述	动态修改 PARAM_INTG_TIME 的值，PARAM_INTG_TIME 为积分时间



函数参数	"intg_value: 0"中积分时间的范围为 0-1500
------	---------------------------------

#### 4.2.3 /smarttof/change\_mod\_freq

使用命令	rosservice call /smarttof/ change_mod_freq "mod_freq_value: 0"
功能描述	动态修改 PARAM_MOD_FREQ 的值，PARAM_MOD_FREQ 为时钟频率
函数参数	"mod_freq_value:0"中目前固定为 12MHz

#### 4.2.4 /smartof/change\_frame\_rate

使用命令	rosservice call /smarttof/ change_frame_rate "frame_rate_value: 0"
功能描述	动态修改 PARAM_FRAME_RATE 的值，PARAM_FRAME_RATE 为帧率
函数参数	"frame_rate_value:0"中的范围为 10-30

#### 4.2.5 /smartof/change\_sync\_delay

使用命令	rosservice call /smarttof/ change_sync_delay "sync_delay_value: 0"
功能描述	动态修改 PARAM_SYNC_DELAY 的值，PARAM_SYNC_DELAY 为同步延时时间
函数参数	"sync_delay_value:0"，0 为自动，1-10 为指定范围

#### 4.2.6 /smartof/change\_filter

使用命令	rosservice call /smarttof/change_filter "filter_id: " filter_value: 0"
功能描述	打开 filter_id 中指定 id 值的滤波功能
函数参数	<p>"filter_id: "中的 id 值可以设置为</p> <p>DMCAM_FILTER_ID_LEN_CALIB, //镜头校准</p> <p>DMCAM_FILTER_ID_PIXEL_CALIB //像素校准</p> <p>DMCAM_FILTER_ID_KALMAN //卡尔曼滤波器</p> <p>DMCAM_FILTER_ID_GAUSS //高斯滤波器</p> <p>DMCAM_FILTER_ID_AMP //幅值滤波器</p> <p>DMCAM_FILTER_ID_AUTO_INTG //积分滤波器</p> <p>DMCAM_FILTER_ID_SYNC_DELAY //消除多模组使用干扰</p> <p>"filter_value:0"中的 value 值默认为 0</p> <p>目前仅 DMCAM_FILTER_ID_AMP 需要设置 filter_value 值，范围为 0-100，</p>

	其它的 filter_id 该值默认为 0 即可。
--	---------------------------

#### 4.2.7 /smartof/disable\_filter

使用命令	rosservice call /smartof/disable_filter "filter_id: ""
功能描述	关闭 filter_id 中指定 id 值的滤波功能
函数参数	<p>"filter_id: ""中的 id 值可以设置为</p> <p>DMCAM_FILTER_ID_LEN_CALIB, //镜头校准</p> <p>DMCAM_FILTER_ID_PIXEL_CALIB //像素校准</p> <p>DMCAM_FILTER_ID_KALMAN //卡尔曼滤波器</p> <p>DMCAM_FILTER_ID_GAUSS //高斯滤波器</p> <p>DMCAM_FILTER_ID_AMP //幅值滤波器</p> <p>DMCAM_FILTER_ID_AUTO_INTG //积分滤波器</p> <p>DMCAM_FILTER_ID_SYNC_DELAY //消除多模组使用干扰</p>

## 5 SDK ROS 系统安装及环境配置

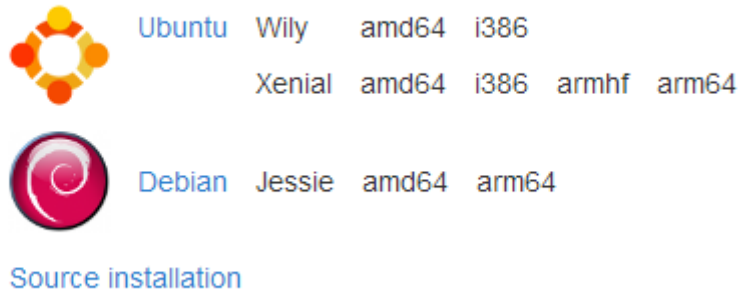
本文档主要介绍 ROS 系统安装、环境配置、测试以及基于 ROS 系统中模组的使用方法。

### 5.1 ROS 系统安装准备工作

#### 5.1.1 ROS 支持的平台

如下图 4-1 所示，推荐使用 Ubuntu14.04 (Trusty) 或者 Ubuntu16.04 (Xenial)。

### 稳定版:



### 测试版:



图 4-1 支持平台

#### 5.1.2 安装前准备工作

安装过程中大概会下载 500MB 左右的软件包，为了避免下载软件包速度过慢，推荐使用网易源或者其他国内源进行下载安装（替换为网易源请参考 <http://mirrors.163.com/help/ubuntu.html>）。

以下安装方法以安装 ROS Kinetic 版本 为例，其他版本安装方法类似。

## 5.2 Ubuntu 系统下快速安装（推荐使用）

- 1) 打开命令行终端，进入 ros 所在文件夹，运行命令:

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

- 2) 出现如下图 4-2 所示选择安装版本，手动输入版本名称后按回车开始安装，Ubuntu14.04 推荐使用 indigo，Ubuntu16.04 推荐使用 kinetic。

```
u@ubuntu:~/Documents/ros$ ./install_ros.sh
u Os Version is 16.04
Your os version is 16.04,please input kinetic or indigo u want to install:kinetic
Your choice is kinetic
```

图 4-2

### 5.3 Ubuntu 系统下命令行安装

- 1) 添加 sources.list, 设置你的电脑可以从 packages.ros.org 接收软件.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release
-sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

- 2) 添加 keys

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80
--recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
sudo apt-get update
```

- 3) 在 ROS 中, 有很多不同的库和工具。我们提供了四种默认的配置来帮助你开始。你也可以单独安装 ROS 包。

- 桌面完整版: (推荐): 包含 ROS、rqt、rviz、机器人通用库、2D/3D 模拟器、导航以及 2D/3D 感知。

```
sudo apt-get install ros-kinetic-desktop-full
```

- 桌面版安装: 包含 ROS、rqt、rviz 以及通用机器人函数库。

```
sudo apt-get install ros-kinetic-desktop
```

- 基础版安装: (简版) 包含 ROS 核心软件包、构建工具以及通信相关的程序库, 无 GUI 工具。

```
sudo apt-get install ros-kinetic-ros-base
```

- 单个软件包安装: 你也可以安装某个指定的 ROS 软件包 (使用软件包名称替换掉下面的 PACKAGE):

```
sudo apt-get install ros-kinetic-PACKAGE
```

例如:

```
sudo apt-get install ros-kinetic-slam-gmapping
```

- 4) 初始化 rosdep

在开始使用 ROS 之前你还需要初始化 rosdep。rosdep 可以方便在你需要编译某些源码的时候为其安装一些系统依赖, 同时也是某些 ROS 核心功能组件所必需用到的工具。

```
sudo rosdep init
rosdep update
```

## 5.4 ROS 环境配置及 DMCAM ROS 编译

### 5.4.1 ROS 环境配置

每次使用 ROS 系统前需要初始化安装版本的环境变量，以 Kinetic 为例，Kinetic 默认安装在/opt/ros/kinetic/目录下，该环境变量配置文件位置/opt/ros/kinetic/setup.bash, 每次使用前需要初始化 ros 环境，命令如下：

```
source /opt/ros/kinetic/setup.bash
```

为了简化配置环境变量的过程，可以选择把环境变量的配置放在 ~/.bashrc 文件中，这样每次打开一个新终端的时候，ROS 的环境变量会自动配置好。

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

### 5.4.2 DMCAM ROS 系统编译

1) 进入 ros 所在目录，通过 ls 命令查看目录中文件如下：

```
install_ros.sh  Makefile  src
```

2) 使用 catkin\_make（Makefile 中实现了 catkin\_make 使用的步骤，也可以使用 make 命令来编译）。

```
source /opt/ros/kinetic/setup.bash(未在 bashrc 中设置初始化环境需要这一步)  
catkin_make
```

3) 编译完成后会生成 devel 和 build 目录，通过 ls 命令查看编译生成的文件

```
build  devel  install_ros.sh  Makefile  src
```

4) 初始化 devel 中的环境变量

```
source devel/setup.bash
```

## 6 RVIZ 显示图像、点云数据

### 6.1 rviz 简介

rviz 是 ros 自带的一个图形化工具，可以方便的对 ros 的程序进行图形化操作。其使用也是比较简单。整体界面如下图 6-1 所示：

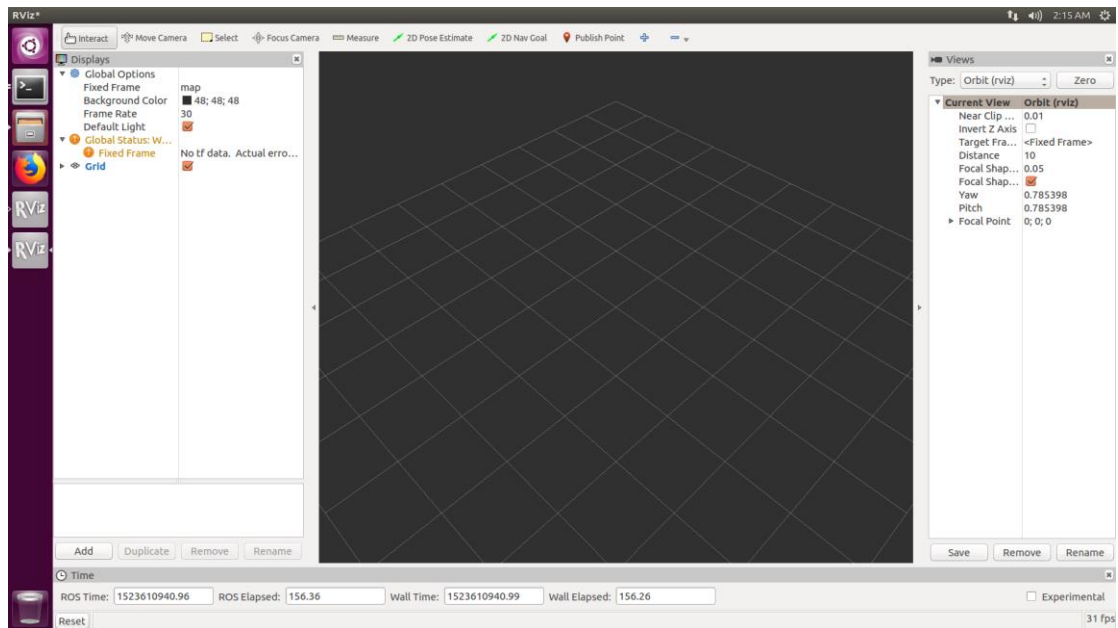


图 6-1 rviz 界面

界面主要分为左侧的显示设置区域，中间的大的显示区域和右侧的视角设置区域。最上面是和导航相关的几个工具。最下面是 ros 状态相关的一些数据的显示。

## 6.2 rviz 使用前准备

- 1) 开启 ROS 环境：

```
Roscore&
```

- 2) 进入 ros 所在文件夹初始化环境变量

```
source ./devel/setup.bash
```

- 3) 运行 launch 文件

```
roslaunch dmcam_ros start.launch
```

## 6.3 rviz 显示深度图像

- 1) 打开一个终端，运行 rviz

```
rviz
```

- 2) 选中 add, By topic 中选中 image\_dist 下的 Image, 最后确认添加, 如下图 6-2 所示:

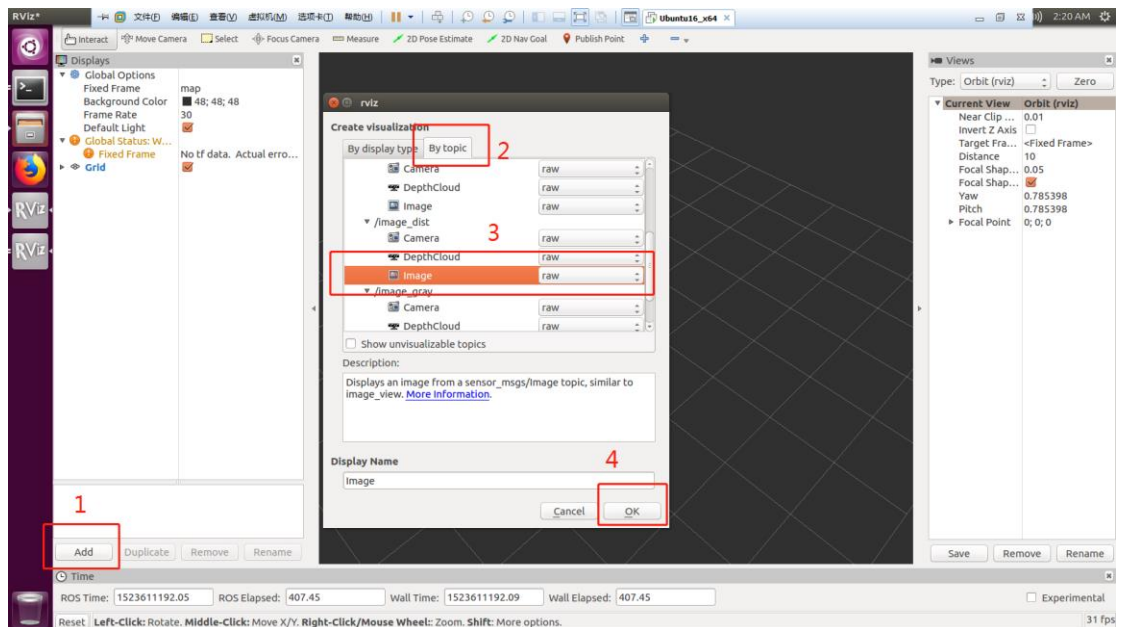


图 6-2 添加 image\_view

- 3) 显示效果如下图 6-3 所示:

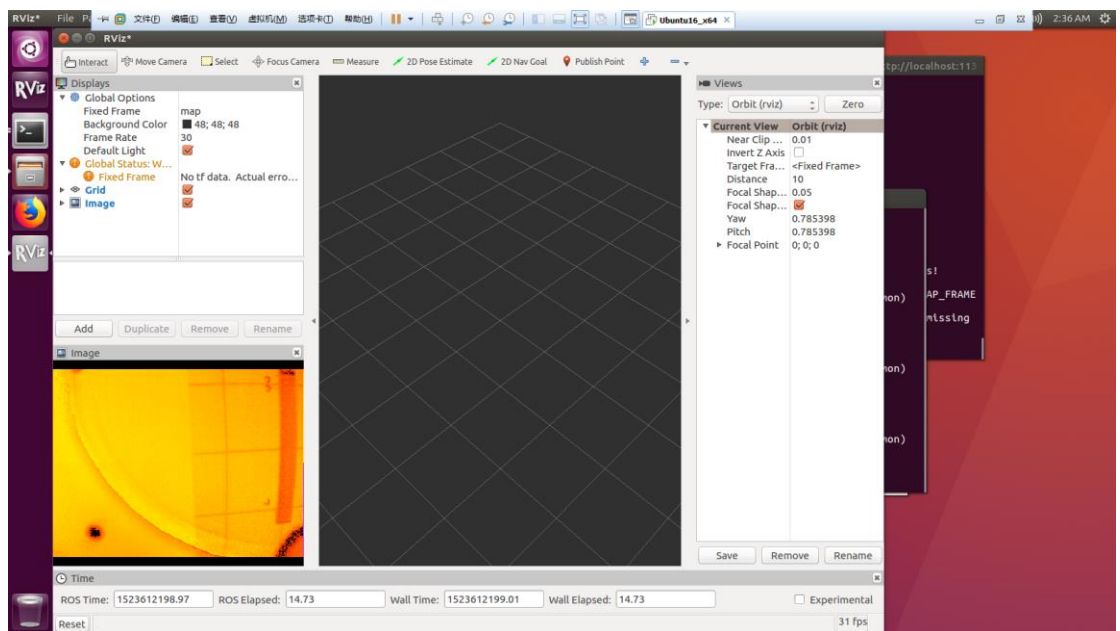


图 6-3 深度图像显示效果

## 6.4 Rviz 显示点云图像

- 1) 打开一个终端, 运行 rviz

```
rviz
```

- 2) 选中 add, byTopic 中选中 pointcloud 下的 PointCloud2, 最后确认添加。



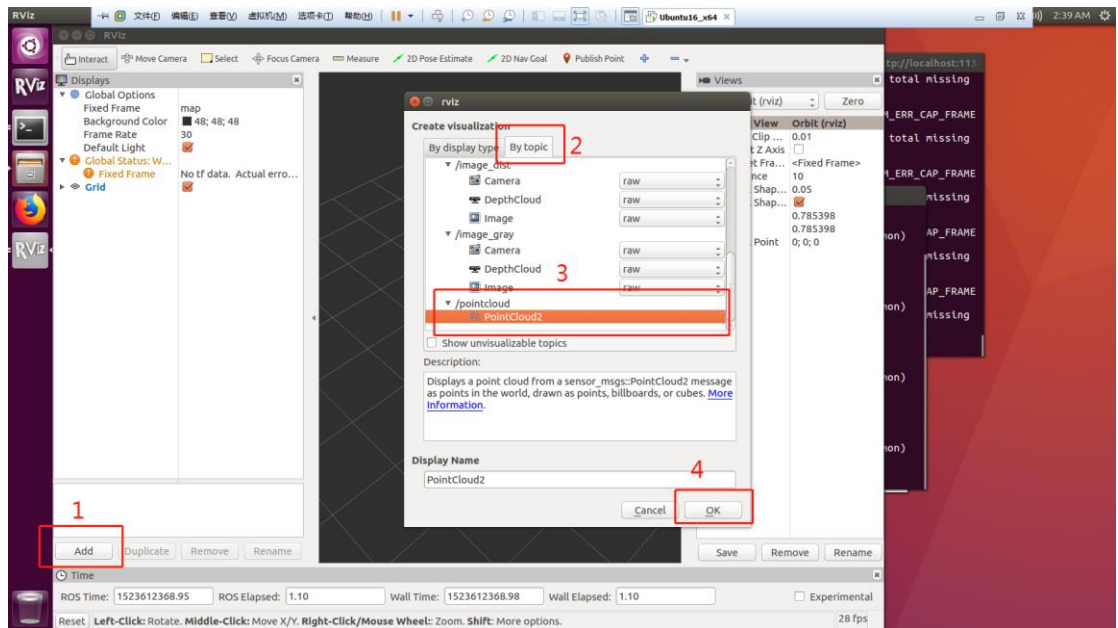


图 6-4 添加点云显示

- 3) 在 rviz 左上角的 displays 区域，修改 GlobalOptions 下的变量 FixedFrame 值为 dmcam\_ros，点云显示如下图

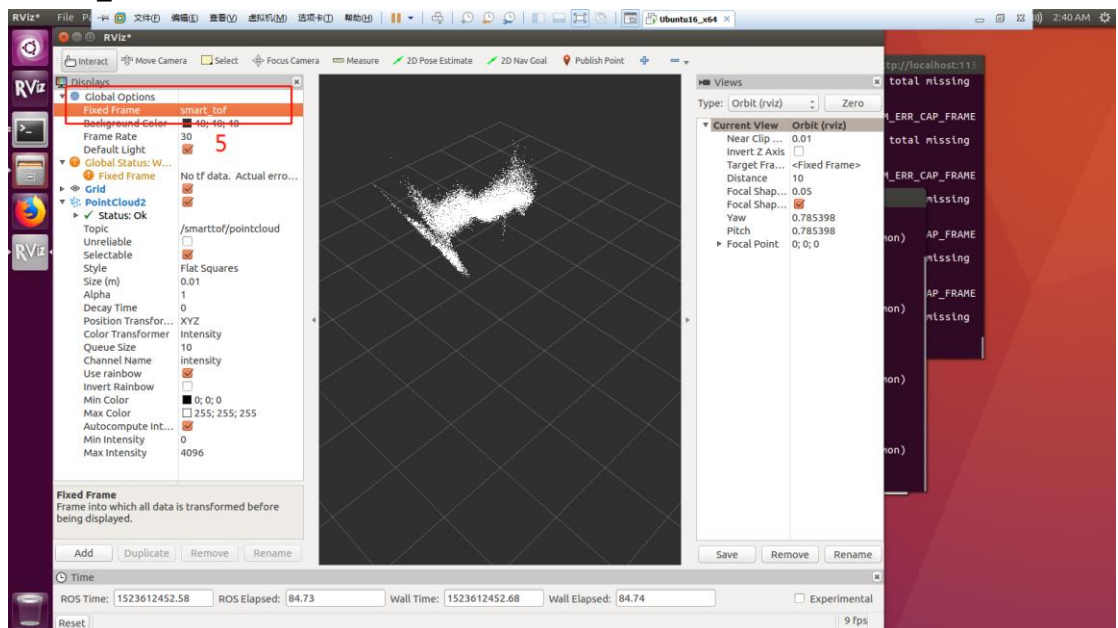


图 6-5 点云显示效果



## 附录

### 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。

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="u  
sername",GROUP="users", MODE="0666"
```

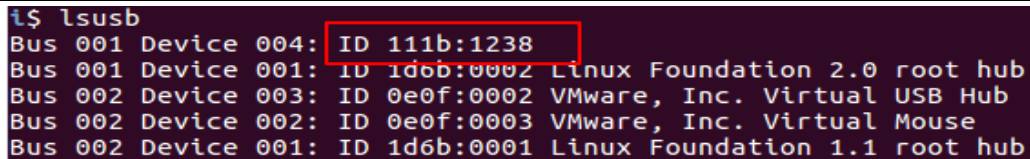
其中“idVendor”和“idProduct”是根据 TOF 设备实际“idVendor”和“idProduct”填写，MODE="0666"表示 USB 设备的权限；99 可以根据电脑上其他文件一致即可。

1) 重启电脑并重新加载 udev 规则：

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

2) 将模组和电脑连接，查看设备，执行如下命令，结果如下图：

```
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
```