Camera Calibration portion of Sensor Fusion Calibration

Description: This tutorial will give step-by-step instructions on how the camera of the Drive PX2 is being calibrated through ROS.

What is needed:

-stereo camera

-GMSL Driver: https://github.com/DavidTorresOcana/ros_gmsl_driver

-ROS Kinetic

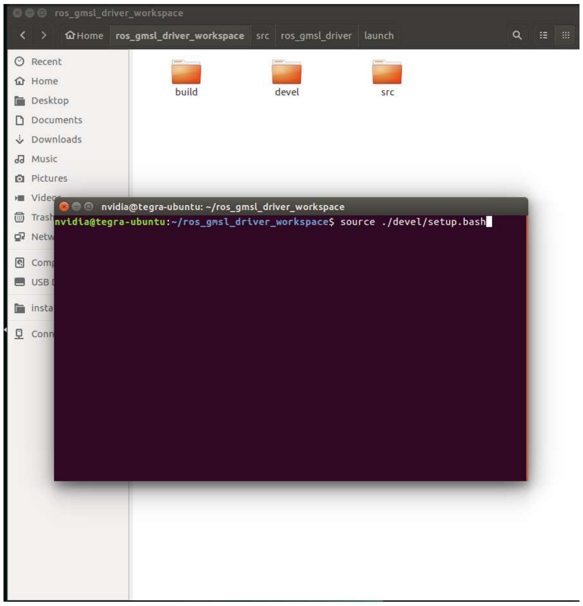
-Checkboard cardboard cutout

Steps

1. Open a terminal at the directory of the top level of your workspace where the camera driver package is located

2. source the workspace in order to recognize the packages. Run the below command on the terminal

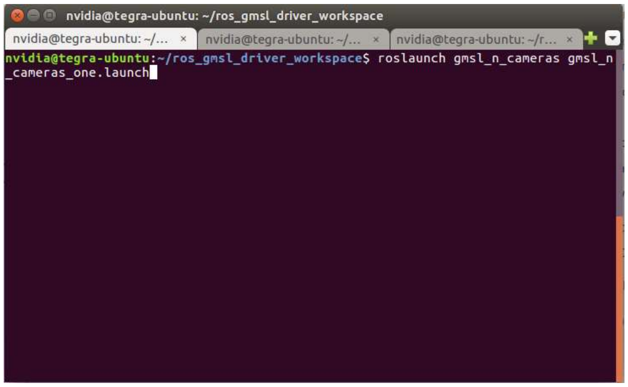
>> source ./devel/setup.bash



```
nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ source ./devel/setup.bash
```

3. Next we will launch the camera driver launch file. Run the below command on the terminal

>> roslaunch gmsl_n_cameras gmsl_n_cameras_one.launch

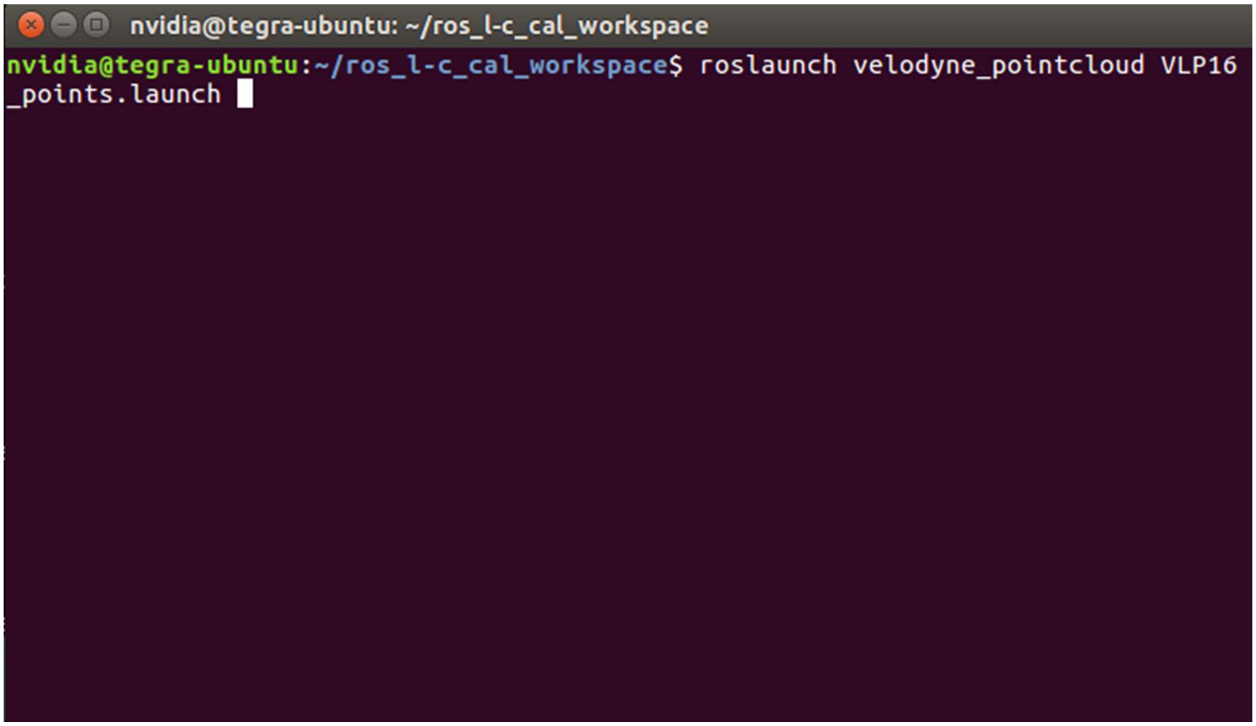


nvidia@tegra-ubuntu: ~/ros_gmsl_driver_workspace
nvidia@tegra-ubuntu: ~/...
nvidia@tegra-ubuntu: ~/...
nvidia@tegra-ubuntu: ~/r...

```
nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ roslaunch gmsl_n_cameras gmsl_n
_cameras_one.launch
```

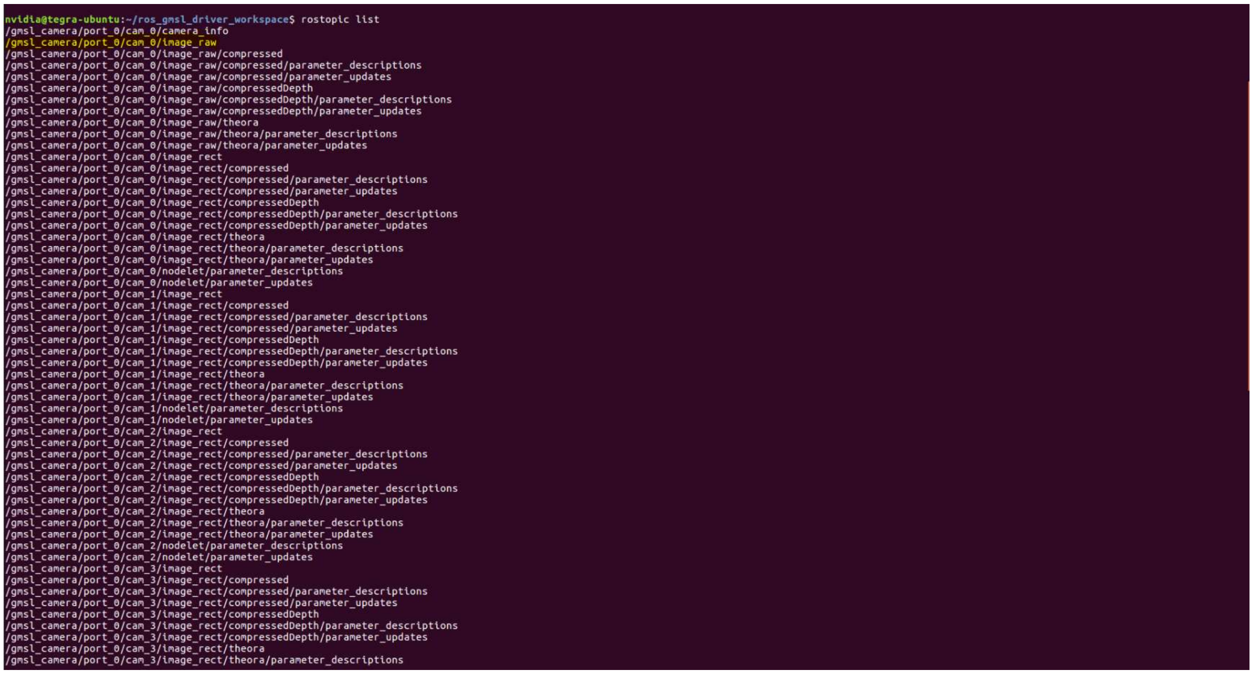
4. Next we will launch the lidar launch file for it to run. Run the below command on the terminal

>> roslaunch velodyne_pointcloud VLP16_points.launch



```
nvidia@tegra-ubuntu:~/ros_l-c_cal_workspace$ roslaunch velodyne_pointcloud VLP16
_points.launch
```

5. Next run the rostopic list command and find the topic for image_raw.



```
nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ rostopic list
/gmsl_camera/port_0/cam_0/camera_info
/gmsl_camera/port_0/cam_0/image_raw
/gmsl_camera/port_0/cam_0/image_raw/compressed
/gmsl_camera/port_0/cam_0/image_raw/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/compressed/parameter_updates
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_0/image_raw/theora
/gmsl_camera/port_0/cam_0/image_raw/theora/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/theora/parameter_updates
/gmsl_camera/port_0/cam_0/image_rect
/gmsl_camera/port_0/cam_0/image_rect/compressed
/gmsl_camera/port_0/cam_0/image_rect/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_rect/compressed/parameter_updates
/gmsl_camera/port_0/cam_0/image_rect/compressedDepth
/gmsl_camera/port_0/cam_0/image_rect/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_rect/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_0/image_rect/theora
/gmsl_camera/port_0/cam_0/image_rect/theora/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_rect/theora/parameter_updates
/gmsl_camera/port_0/cam_0/nodelet/parameter_descriptions
/gmsl_camera/port_0/cam_0/nodelet/parameter_updates
/gmsl_camera/port_0/cam_1/image_rect
/gmsl_camera/port_0/cam_1/image_rect/compressed
/gmsl_camera/port_0/cam_1/image_rect/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_1/image_rect/compressed/parameter_updates
/gmsl_camera/port_0/cam_1/image_rect/compressedDepth
/gmsl_camera/port_0/cam_1/image_rect/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_1/image_rect/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_1/image_rect/theora
/gmsl_camera/port_0/cam_1/image_rect/theora/parameter_descriptions
/gmsl_camera/port_0/cam_1/image_rect/theora/parameter_updates
/gmsl_camera/port_0/cam_1/nodelet/parameter_descriptions
/gmsl_camera/port_0/cam_1/nodelet/parameter_updates
/gmsl_camera/port_0/cam_2/image_rect
/gmsl_camera/port_0/cam_2/image_rect/compressed
/gmsl_camera/port_0/cam_2/image_rect/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_2/image_rect/compressed/parameter_updates
/gmsl_camera/port_0/cam_2/image_rect/compressedDepth
/gmsl_camera/port_0/cam_2/image_rect/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_2/image_rect/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_2/image_rect/theora
/gmsl_camera/port_0/cam_2/image_rect/theora/parameter_descriptions
/gmsl_camera/port_0/cam_2/image_rect/theora/parameter_updates
/gmsl_camera/port_0/cam_2/nodelet/parameter_descriptions
/gmsl_camera/port_0/cam_2/nodelet/parameter_updates
/gmsl_camera/port_0/cam_3/image_rect
/gmsl_camera/port_0/cam_3/image_rect/compressed
/gmsl_camera/port_0/cam_3/image_rect/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_3/image_rect/compressed/parameter_updates
/gmsl_camera/port_0/cam_3/image_rect/compressedDepth
/gmsl_camera/port_0/cam_3/image_rect/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_3/image_rect/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_3/image_rect/theora
/gmsl_camera/port_0/cam_3/image_rect/theora/parameter_descriptions
```

6. Once the topic is located, you finally run the camera calibration python script. The example command and parameter description can be seen below.

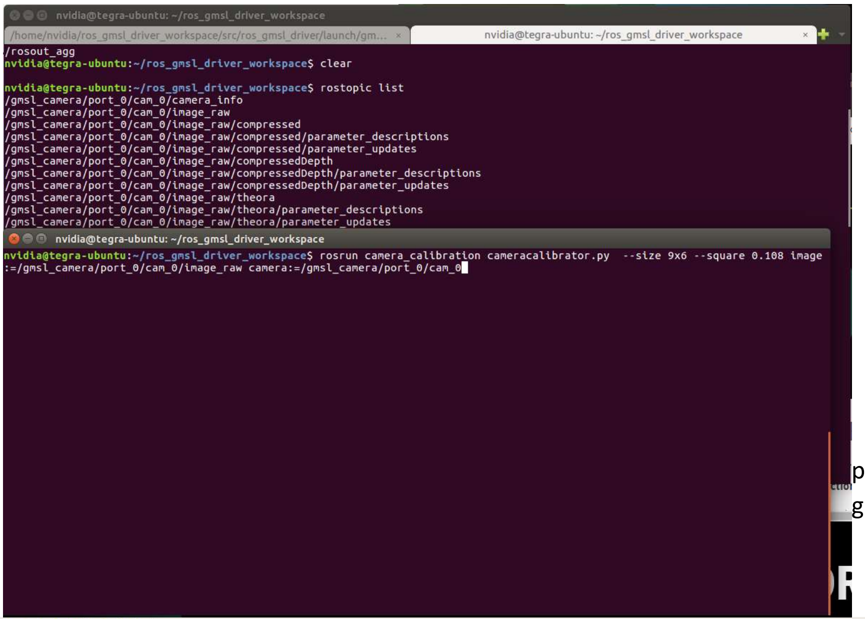
>> rosrun camera_calibration cameracalibrator.py --size 9x6 --square 0.108 image:=/gmsl_camera/port_0/cam_0/image_raw camera:=/gmsl_camera/port_0/cam_0

--size: Indicates the size of the checkboard that will be using. It is described as the cross-sections only and not the actual checker squares. An example, if you have a 10x7 checkboard, you will input a 9x6 into the size parameters since the program does not check the edges but the cross-sections between the squares.

--square: This indicates the size of the individual square used on the check board

--image: This label is used in the calibration script. You will have to map the raw_image to this label

--camera: This label is used in the calibration script. You will have to map the camera topic that is used without the raw_image.

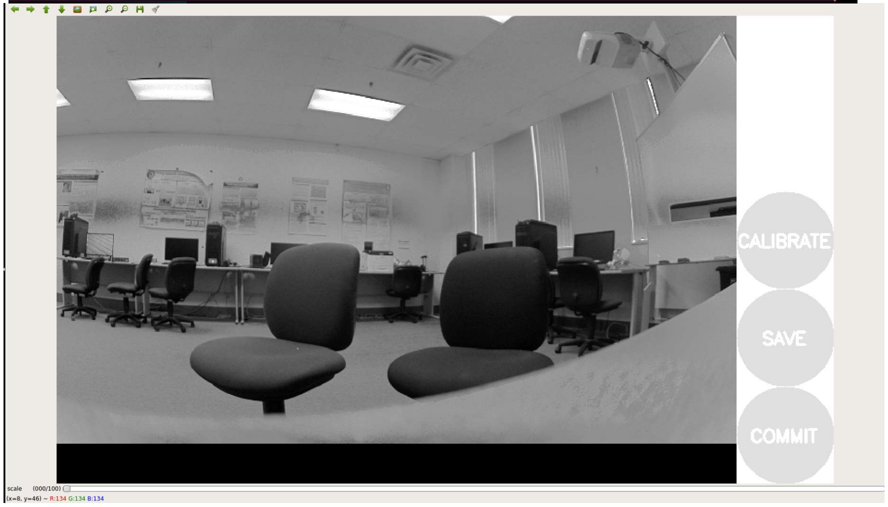


```
nvidia@tegra-ubuntu: ~/ros_gmsl_driver_workspace
/home/nvidia/ros_gmsl_driver_workspace/src/ros_gmsl_driver/launch/gm...    nvidia@tegra-ubuntu: ~/ros_gmsl_driver_workspace
/rosout_agg
nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ clear

nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ rostopic list
/gmsl_camera/port_0/cam_0/camera_info
/gmsl_camera/port_0/cam_0/image_raw
/gmsl_camera/port_0/cam_0/image_raw/compressed
/gmsl_camera/port_0/cam_0/image_raw/compressed/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/compressed/parameter_updates
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/compressedDepth/parameter_updates
/gmsl_camera/port_0/cam_0/image_raw/theora
/gmsl_camera/port_0/cam_0/image_raw/theora/parameter_descriptions
/gmsl_camera/port_0/cam_0/image_raw/theora/parameter_updates
```

```
nvidia@tegra-ubuntu: ~/ros_gmsl_driver_workspace
nvidia@tegra-ubuntu:~/ros_gmsl_driver_workspace$ rosrun camera_calibration cameracalibrator.py  --size 9x6 --square 0.108 image
:=/gmsl_camera/port_0/cam_0/image_raw camera:=/gmsl_camera/port_0/cam_0
```

7. Once the program is running, you will get a screen like the one below.



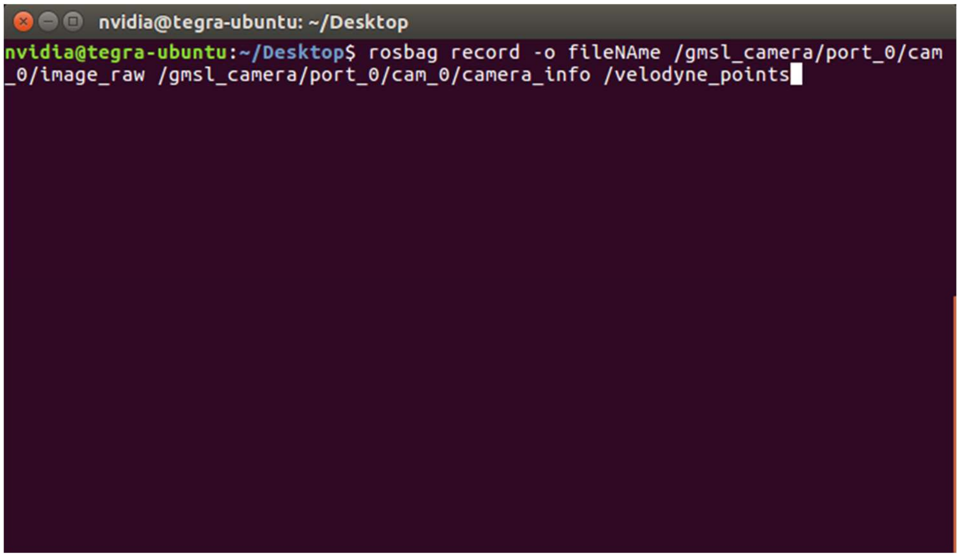
8. Now for our sensor fusion calibration, we will need a rosbag of all the camera and lidar data that can be calibrated to be able to go through the procedure. You will need to record your camera topics, camera_info, and point cloud of your lidar to get through the next steps. See below example

rosbag record -o [fileName to name the bag file] [camera topic to record] [camera info topic to record] [lidar point cloud topic to record]

Example:

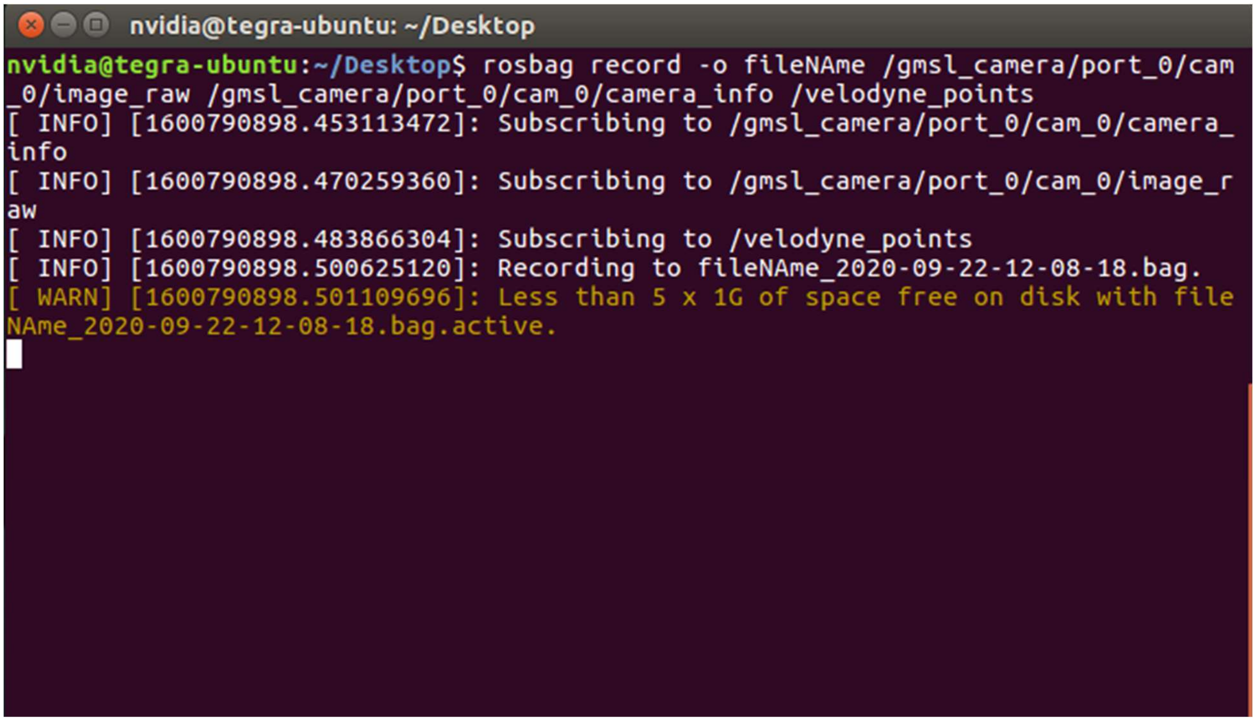
>> rosbag record -o fileName /gmsl_camera/port_0/cam_0/image_raw /gms_camera/port_0/cam_0/camera_info /velodyne_points

Before



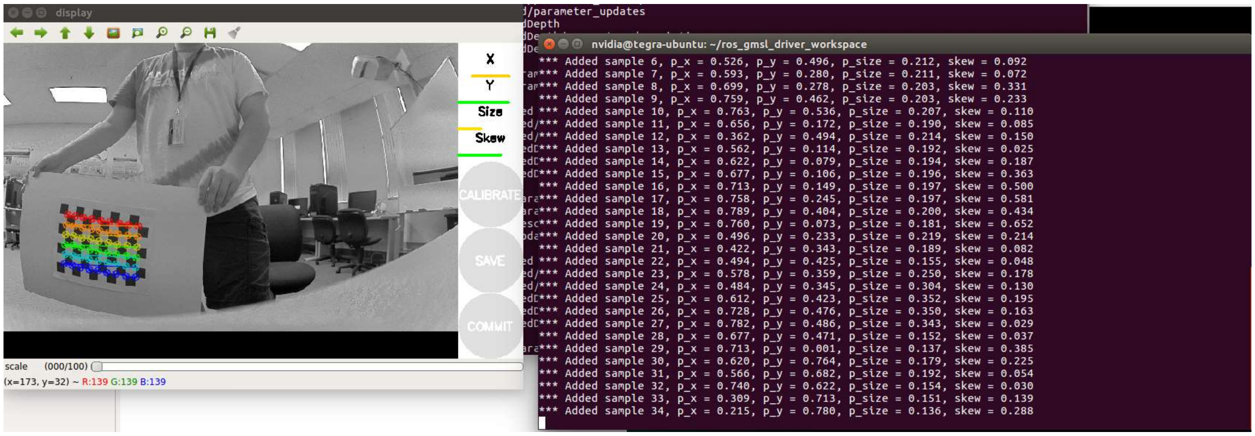
```
nvidia@tegra-ubuntu:~/Desktop$ rosbag record -o fileNAme /gmsl_camera/port_0/cam
_0/image_raw /gmsl_camera/port_0/cam_0/camera_info /velodyne_points
```

After



```
nvidia@tegra-ubuntu:~/Desktop$ rosbag record -o fileNAme /gmsl_camera/port_0/cam
_0/image_raw /gmsl_camera/port_0/cam_0/camera_info /velodyne_points
[ INFO] [1600790898.453113472]: Subscribing to /gmsl_camera/port_0/cam_0/camera_
info
[ INFO] [1600790898.470259360]: Subscribing to /gmsl_camera/port_0/cam_0/image_r
aw
[ INFO] [1600790898.483866304]: Subscribing to /velodyne_points
[ INFO] [1600790898.500625120]: Recording to fileNAme_2020-09-22-12-08-18.bag.
[ WARN] [1600790898.501109696]: Less than 5 x 1G of space free on disk with file
NAme_2020-09-22-12-08-18.bag.active.
```

9. Now the program has to get data in order to calibrate the camera. Move the checkerboard image infront of the camera. The program will get data like the images below.

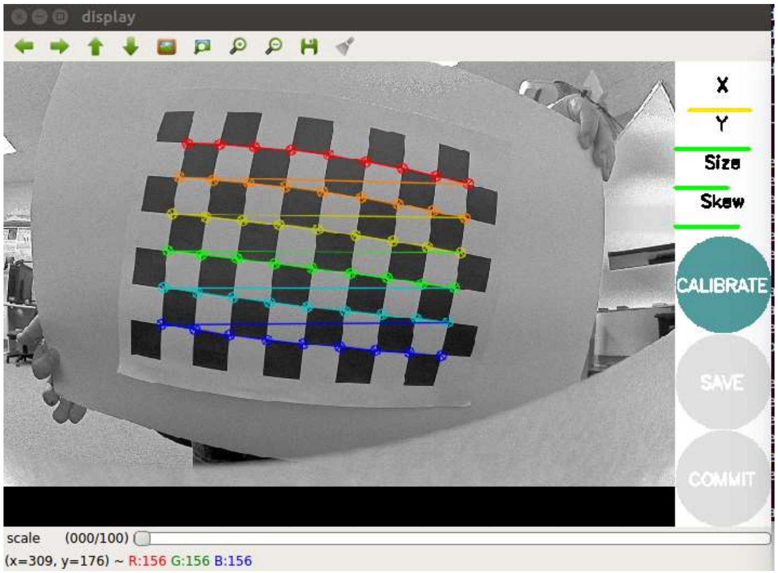


```
/parameter_updates
Depth
```

nvidia@tegra-ubuntu: ~/ros_gmsl_driver_workspace

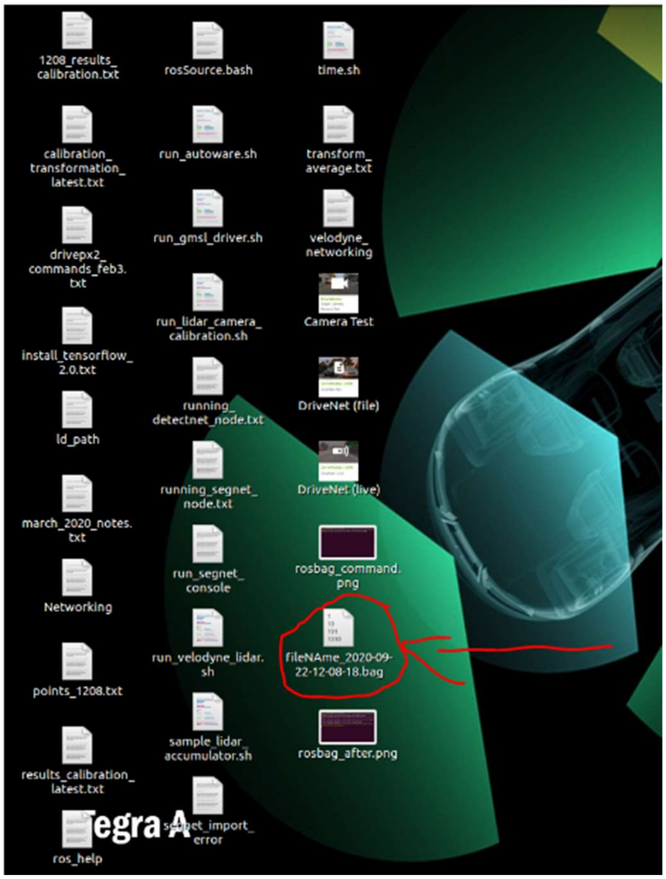
```
*** Added sample 6, p_x = 0.526, p_y = 0.496, p_size = 0.212, skew = 0.092
*** Added sample 7, p_x = 0.593, p_y = 0.280, p_size = 0.211, skew = 0.072
*** Added sample 8, p_x = 0.699, p_y = 0.278, p_size = 0.203, skew = 0.331
*** Added sample 9, p_x = 0.759, p_y = 0.462, p_size = 0.203, skew = 0.233
*** Added sample 10, p_x = 0.763, p_y = 0.536, p_size = 0.207, skew = 0.110
*** Added sample 11, p_x = 0.656, p_y = 0.172, p_size = 0.190, skew = 0.085
*** Added sample 12, p_x = 0.362, p_y = 0.494, p_size = 0.214, skew = 0.150
*** Added sample 13, p_x = 0.562, p_y = 0.114, p_size = 0.192, skew = 0.025
*** Added sample 14, p_x = 0.622, p_y = 0.079, p_size = 0.194, skew = 0.187
*** Added sample 15, p_x = 0.677, p_y = 0.106, p_size = 0.196, skew = 0.363
*** Added sample 16, p_x = 0.713, p_y = 0.149, p_size = 0.197, skew = 0.500
*** Added sample 17, p_x = 0.758, p_y = 0.245, p_size = 0.197, skew = 0.581
*** Added sample 18, p_x = 0.789, p_y = 0.404, p_size = 0.200, skew = 0.434
*** Added sample 19, p_x = 0.760, p_y = 0.073, p_size = 0.181, skew = 0.652
*** Added sample 20, p_x = 0.496, p_y = 0.233, p_size = 0.219, skew = 0.214
*** Added sample 21, p_x = 0.422, p_y = 0.343, p_size = 0.189, skew = 0.082
*** Added sample 22, p_x = 0.494, p_y = 0.425, p_size = 0.155, skew = 0.048
*** Added sample 23, p_x = 0.578, p_y = 0.359, p_size = 0.250, skew = 0.178
*** Added sample 24, p_x = 0.484, p_y = 0.345, p_size = 0.304, skew = 0.130
*** Added sample 25, p_x = 0.612, p_y = 0.423, p_size = 0.352, skew = 0.195
*** Added sample 26, p_x = 0.728, p_y = 0.476, p_size = 0.350, skew = 0.163
*** Added sample 27, p_x = 0.782, p_y = 0.486, p_size = 0.343, skew = 0.029
*** Added sample 28, p_x = 0.677, p_y = 0.471, p_size = 0.152, skew = 0.037
*** Added sample 29, p_x = 0.713, p_y = 0.001, p_size = 0.137, skew = 0.385
*** Added sample 30, p_x = 0.620, p_y = 0.764, p_size = 0.179, skew = 0.225
*** Added sample 31, p_x = 0.566, p_y = 0.682, p_size = 0.192, skew = 0.054
*** Added sample 32, p_x = 0.740, p_y = 0.622, p_size = 0.154, skew = 0.030
*** Added sample 33, p_x = 0.309, p_y = 0.713, p_size = 0.151, skew = 0.139
*** Added sample 34, p_x = 0.215, p_y = 0.780, p_size = 0.136, skew = 0.288
```

10. Once the program thinks it ready to calibrate, the calibrate button will turn blue indicating calibration is ready. **Please keep calibrating until all the X,Y, Size, and Skew bars below are green colors.** For good lidar data, make sure you get multiple instances of the checkerboard box further away from the camera.



11. Once the parameters are green, press control c and end the rosbag record. Once the rosbag is finished writing, then make sure it is saved to the specific directory your terminal is set to. It will be called a .bag file with the file name you specified in the rosbag record command.

References



- GMSL Camera Driver
  https://github.com/DavidTorresOcana/ros_gmsl_driver
- Calibrating a stereo camera through ROS
  http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration