

Discussions for the week

1. Where is odometry/filtered coming from?
2. We need better bags, but moreso I need static obstacles which are 3D planar rich
3. I've found few target based methods we could try (mentioned below)
4. I need camera intrinsics (distortion coefficients and intrinsics) even for the
extended_lidar_camera_calib
(right now it's empty in camera_info topic)

Target-Less Methods

All the ones below need basic intrinsics to be known.

1. [External Lidar-Camera Calib \(Shrijit's suggestion\)](#)
Essentially a combination of FLOAM and Livox
2. [Open Calib's lidar2camera](#)
However, this needs an extrinsics estimate

Target Based Methods

1. [Joint Camera Intrinsic and Lidar2Cam calib](#)
Seems feasible in ros1, but poor documentation and board seems bit hard to make although CSV seems flexible
2. [velo2cam calibration](#)
The board for this seems easier (just need large carboard and 4 A4 sheets for aruco tags)
3. [Lidar Reflectance based method, might not work in field, also untested on VLP16](#)
4. [Manul calib](#)
Good documentation and seems to work on noetic. We can try

velo2cam

1. All objects must be static for at least 30 frames
2. The calibration board as well **as the surface behind it** should be visible

3. The frame_id of the sensor_msgs/Image topic and the camera_calibration topic must have the same frame ID
4. The camera_calibration topic must have the distortion parameters
5. Pattern should be kept in FOV of both Lidar and Mono Camera

Pipeline build

This was simple and the only packages required in noetic was:

```
sudo apt install ros-noetic-opencv-apps
sudo apt install ros-noetic-tf-conversions
sudo apt install ros-noetic-image-geometry
```

Issues with current rosbag

oak/rgb/camera_info and oak/rgb/image_raw

The topics for this camera seem perfect with distortion parameters filled as well as the frame_id for both the topics matching (which is required)

```
root@sush-pc:/home/sush/input_bags/calibration-20230815T123740Z-001/calibration# rostopic echo -n 1 --noarr /oak/rgb/image_raw
header:
  seq: 1664
  stamp:
    secs: 1690912215
    nsecs: 724100650
  frame_id: "oak_rgb_camera_optical_frame"
height: 720
width: 1280
encoding: "bgr8"
is_bigendian: 0
step: 3840
data: "<array type: uint8, length: 2764800>"
---
```

```
root@sush-pc:/home/sush/input_bags/calibration-20230815T123740Z-001/calibration# rostopic echo /oak/rgb/camera_info [430/457]
header:
  seq: 8935
  stamp:
    secs: 1690912205
    nsecs: 426179451
  frame_id: "oak_rgb_camera_optical_frame"
height: 720
width: 1280
distortion_model: "rational_polynomial"
D: [-5.864197731018066, 56.85762405395508, 0.0004187868908047676, 0.000990380416624248, 9.401820182800293, -5.694280624389648, 55.49209213256836, 22.847972869873047]
K: [759.4764404296875, 0.0, 635.1179809570312, 0.0, 759.4764404296875, 361.7841796875, 0.0, 0.0, 1.0]
R: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
P: [759.4764404296875, 0.0, 635.1179809570312, 0.0, 0.0, 759.4764404296875, 361.7841796875, 0.0, 0.0, 0.0, 1.0, 0.0]
binning_x: 0
binning_y: 0
roi:
  x_offset: 0
  y_offset: 0
  height: 0
  width: 0
  do_rectify: False
---
```

However, it looks like this was not the camera intended for calibration (target not in view)



Bayer Camera

This topic did not have any distortion parameters in the camera info (will need to be filled if it is to be used). The frame ID will also need to be manually set to satisfy the velo2cam requirements

```
root@sush-pc:/home/sush/input_bags/calibration-20230815T123740Z-001/calibration# rostopic echo -n 1 --noarr /camera/image_raw
header:
  seq: 1157
  stamp:
    secs: 1690912214
    nsecs: 627232479
  frame_id: "camera"
height: 1032
width: 1384
encoding: "bayer_rggb8"
is_bigendian: 0
step: 1384
data: "<array type: uint8, length: 1428288>"
```



```

root@sush-pc: /home/sush/input_bags/calibration-20230815T123740Z-001/calibration# rostopic echo /camera/camera_info
header:
  seq: 2224
  stamp:
    secs: 1690912207
    nsecs: 627306323
  frame_id: "camera"
height: 0
width: 0
distortion_model: ''
D: []
K: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
R: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
P: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
binning_x: 1
binning_y: 1
roi:
  x_offset: 0
  y_offset: 0
  height: 0
  width: 0
  do_rectify: True
---
```

The target in the rosbag also has to be stationary, in the rosbag it was handheld, will need to capture another rosbag in FRC.

