Perception & Sensing WG Meeting 19/04/2023 #3442

drwnz started this conversation in **Working group meetings**

drwnz on Apr 19, 2023 Collaborator edited

Administrative

- [Previous meeting slides]
- Proposed regular meeting time: Every second Wednesday, 10am UTC (alternating with Mapping WG)
- [WG Google Drive]

Attendees

▼ Chaired by David Wong (TIER IV)

- Bonolo Mathibela
- Ryohsuke Mitsudome
- Fatih Cırıt
- Zeynep Akbalık
- Kaan Colak
- Alexey Panferov
- Kotaro Uetake
- Shunsuke Miura
- Yoshi Ri
- Ba Dai Nguyen
- Akihito Ohsato

Agenda

- Introduction of new attendees
- WG best practices (Bonolo)
- Introductory slides recap
- Progress report on Completion of Sensing & Perception Documentation
- Progress report on Add perception benchmark to CI/CD process
- Progress report on Improve point cloud synchronization
- Progress report on Increase LiDAR Detection Accuracy
- Progress report on Camera-only Detection Pipeline
- Progress report on ML-based Occupancy Grid Map Generation
- Progress report on Improving Camera Exposure
- Progress report on Update Radar Fused Detection Pipeline
- Progress report on Camera-LiDAR DNN Fusion Detection

Category

Working group meetings

Labels

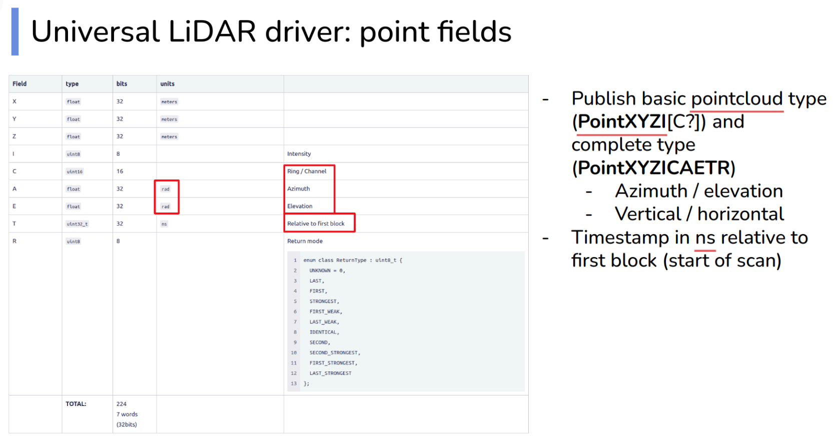
meeting:sensing-p...

3 participants

- Progress report on Providing ML Training Platform
- Progress report on Universal LiDAR Driver
- Progress report on Calibration Tools

Discussion topics

- WG best practices (Bonolo)
  - Ref: WG Best Practices
- Introductory slides recap
  - Proposed task for this WG (Bonolo): Update documentation so it is easy to understand which sensors are supported. Ref: issue, previous hardware working group, supported hardware list
- Progress report on Completion of Sensing & Perception Documentation
  - Github documentation
  - Draft perception document to be started by Miura-san (TIER IV) and Zeynep Akbalık (LeoDrive)
  - Fatih: best to first explain current design and then add future plans in each document
- Progress report on Universal LiDAR Driver
  - Update on Pointcloud field types by David Wong
    - Proposed point field types:



Universal LiDAR driver: point fields

Field	type	bits	units	
X	float	32	meters	
Y	float	32	meters	
Z	float	32	meters	
I	uint8	8		Intensity
C	uint16	16		Ring / Channel
A	float	32	rad	Azimuth
E	float	32	rad	Elevation
T	uint32_t	32	ns	Relative to first block

R	uint8	8		Return mode
				enum class ReturnType : uint8_t {
				UNKNOWN = 0,
				LAST,
				FIRST,
				STRONGEST,
				FIRST_WEAK,
				LAST_WEAK,
				IDENTICAL,
				SECOND,
				SECOND_STRONGEST,
				FIRST_STRONGEST,
				LAST_STRONGEST
				};

	TOTAL:	224 7 words (32bits)		

- Publish basic pointcloud type (PointXYZI[C?]) and complete type (PointXYZICAETR)
  - Azimuth / elevation
  - Vertical / horizontal
- Timestamp in ns relative to first block (start of scan)

    - Radians for angular values is positive for all
    - Still split between A, E and V, H for polar coords - both parties happy with either though

Action items

- □ Announce formation of working group on ROS Discourse

↑ 1

2 comments · 6 replies

Oldest Newest Top

xmfcx on Apr 24, 2023 Maintainer

Open issues:

- Map Based Prediction Incorrect Predictions autoware.universe#494

↑ 1 0 replies

VRichardJP on May 24, 2023 Collaborator

Hi, sorry to hijack old report. I have a few remarks/questions regarding the Universal LIDAR driver point field:

- Is there any reason to keep intensity field in basic point type?
- Why is there no "distance" field? X/Y/Z and Azimuth/Elevation/Distance describe a point in 2 different coordinate systems, if there is Azimuth and Elevation already, why not distance aswell?
- From data alignment perspective, it would be more efficient to pack the return value along the intensity field. Is there a reason for it to be at the end?

↑ 1 6 replies

Show 1 previous reply

VRichardJP on May 24, 2023 Collaborator edited ▾

I agree XYZIRC looks better than just XYZI, because if the data is packed then reading/writing unaligned data point might be slow, and if the data is aligned, then we have 24 unused bits. Now maybe channel and return_type data are not that useful. I don't know what return_type can be used for, as for channel I think that currently it is only used in ring_outlier_filter (which needs azimuth as well). What about RGB color? I have seen some lidar sensors can output color, which could fit in 24 bits.

xmfcx on May 24, 2023 Maintainer edited ▾

The lidars that I know which supply the RGB color internally mix camera and lidar data and output the colored point cloud. These lidars are generally short range like:

- https://www.intelrealsense.com/lidar-camera-l515/ (9m)
- https://www.stereolabs.com/zed-2/ (stereo camera(no lidar), 20m max, accuracy low and gets lower as distance increases)

I think it's too early to think about incorporating them just yet.

drwnz on May 24, 2023 Collaborator Author

@VRichardJP @xmfcx thank you for your comments.
Indeed I missed out Distance in that image.
The actual data structure we use is ordered like this:

```
struct PointXYZICAETR
{
  float x;
  float y;
  float z;
  float azimuth;
  float elevation;
  std::uint8_t return_type;
  std::uint8_t intensity;
  std::uint16_t channel;
  std::uint32_t time_stamp;
};
```

The naming order is different in order more for clarity I believe: @amc-nu is that right?

xmfcx on May 24, 2023 Maintainer edited

@drwnz If we are planning to use `XYZIRC` for the default point type, it'd make more sense to restructure it as the following (also added the missing distance field):

```
struct PointXYZIRCAEDT
{
  float x;
  float y;
  float z;
  std::uint8_t intensity;
  std::uint8_t return_type;
  std::uint16_t channel;
  float azimuth;
  float elevation;
  float distance;
  std::uint32_t time_stamp;
};
```

Is this ok?

👍 1

drwnz on May 29, 2023 Collaborator Author

@xmfcx - yes this looks good, we will modify accordingly.
Thank you for investigating this, as always your suggestions have been very helpful!

❤️ 1