

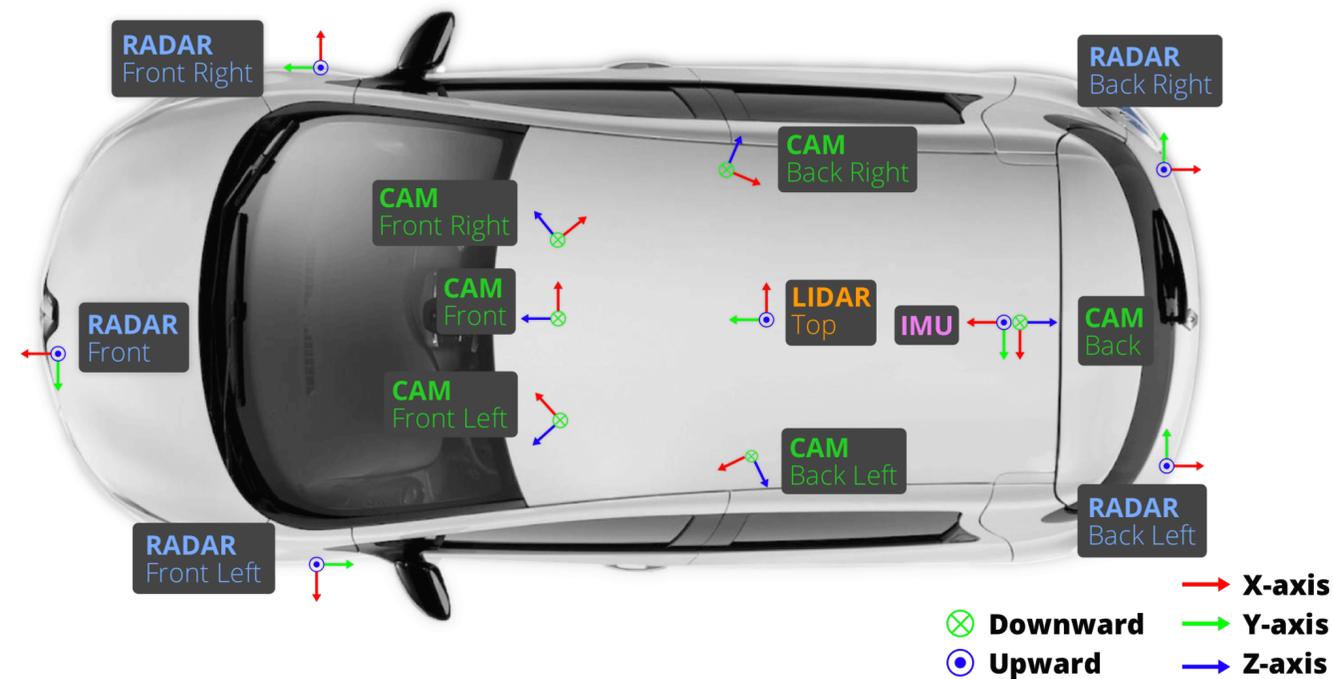
Dataset Analysis: nuLiDARtext

倪培洋, 2024 年 10 月 25 日

nuLiDARtext

Overview

- 在 nuScenes 的基础上，针对单帧 LiDAR 数据的配对文本描述进行了优化
- 由 nuScenes 提供的分布在 850 个场景（scenes）的 3,4149 个关键帧（sample frames）以及对应的优化文本描述对组成



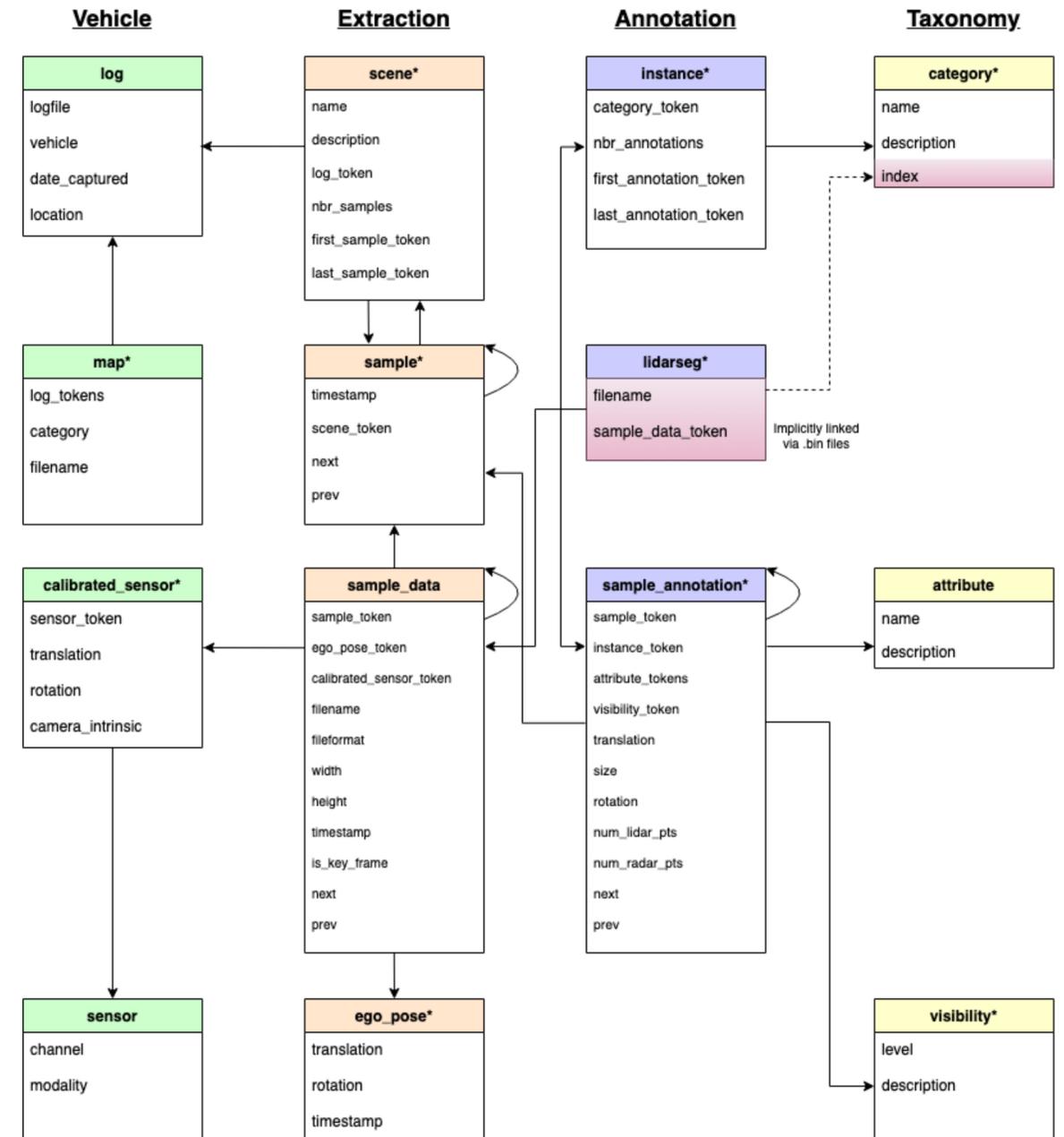
nuScenes

Overview

- 在 4 张地图上（波士顿 Seaport、Queenstown 以及新加坡 One North、Holland Village）
- 使用 2 台载具，每台载具 12 个采集设备（6 个 Camera、5 个 Radar 以及 1 个 LiDAR）

nuScenes schema

Asterisks (*) indicate modifications compared to the nulmages schema.
Tables and fields added in nuScenes-lidarseg have a purple background color.



nuScenes

map.json

- 存储四张地图的信息以及指向每张地图对应的日志数据。

```
1 {
2   "category": "semantic_prior",
3   "token": "37819e65e09e5547b8a3ceaefba56bb2",
4   "filename": "maps/37819e65e09e5547b8a3ceaefba56bb2.png",
5   "log_tokens": [
6     "853a9f9fe7e84bb8b24bff8ebf23f287",
7     "e55205b1f2894b49957905d7ddfdb96d",
8     "8fefc430cbfa4c2191978c0df302eb98",
9     "f93e8d66ce4b4fba7062d19b1fe29fb",
10    "89a56a5dc3aa4e56a2e57b52de738da5"
11  ]
12 }
```

nuScenes

log.json

- 共有两辆数据采集车，编号分别为`n008`和`n015`，在log.json中记录了每辆车进行数据采集的日期和地点，共68个log records。

```
1 {
2   "token": "6b6513e6c8384cec88775cae30b78c0e",
3   "logfile": "n015-2018-07-18-11-07-57+0800",
4   "vehicle": "n015",
5   "date_captured": "2018-07-18",
6   "location": "singapore-onenorth"
7 },
```

nuScenes

scene.json

- 共有 850 个场景 (scene)，每个场景是一个 20 秒长的帧序列。
- 在 scene.json 中记录了每个场景对应的描述、采集的样本数以及对应的 log 记录，并且以外键的形式额外提供了每个场景的第一个样本的 token 以及最后一个样本的 token。
- 一个 scene 对应一个 log，一个 log 可以对应多个 scene。

```
1 {
2   "token": "55638ae3a8b34572aef756ee7fbce0df",
3   "log_token": "6577357788b24c35a3b0419c138f50db",
4   "nbr_samples": 40,
5   "first_sample_token": "4777f381d14a40b79e014b5038712d58",
6   "last_sample_token": "ab7d1d045d4b4a228a86ea12bf5bc43a",
7   "name": "scene-0517",
8   "description": " cars, pedestrians"
9 },
```

nuScenes

sample.json

- 记录所有场景对应的关键 sample 帧 (LiDAR 数据)。
- 包含当前关键帧的 token、前一关键帧的 token 以及后一关键帧的 token，形成链表结构表示时间序列。
- 共计 3,4149 个 LiDAR sample 关键帧。

```
1 {
2   "token": "b8002d4effb540c0b1bb70addf423f32",
3   "timestamp": 1531883533950168,
4   "prev": "140c678882194ae69e47136c81168b6d",
5   "next": "1e19d0a5189b46f4b62aa47508f2983e",
6   "scene_token": "73030fb67d3c46cfb5e590168088ae39"
7 },
```

nuScenes

sample_data.json

- 记录采集的所有数据对应的真实数据文件名称，并以链表的形式形成时间序列。
- 包括 Camera data、Radar data 以及 LiDAR data 的所有 sample 关键帧以及 sweep 非关键帧。
- 其中，LiDAR 数据的关键帧就是 `sample.json` 文件中的 record（时间戳与 `sample.json` 中的时间戳基本相同），非关键帧的所有数据以及关键帧的 LiDAR 之外的其他数据围绕关键帧的 LiDAR 数据展开存储（时间戳是在 LiDAR 关键帧的前后）。
- 共计 2,631,083 个 data。

```
1 {
2   "token": "d97532bddc024fa09673a021a4295943",
3   "sample_token": "140c678882194ae69e47136c81168b6d",
4   "ego_pose_token": "d97532bddc024fa09673a021a4295943",
5   "calibrated_sensor_token": "7781065816974801afc4dcdaf6acf92c",
6   "timestamp": 1531883533145867,
7   "fileformat": "pcd",
8   "is_key_frame": false,
9   "height": 0,
10  "width": 0,
11  "filename": "sweeps/RADAR_FRONT/n015-2018-07-18-11-07-57+0800__RADAR_FRONT__1531883533145867.pcd",
12  "prev": "b6e0db5decd74ea4b89effc60f932aeb",
13  "next": "36724a4f3c2f4a35b836d8a0e4758dea"
14 },
```

nuScenes

attribute.json

- 8 个属性，定义了物体的状态，
attribute 在 category 不变的情况下改变，如载具是运动状态还是静止状态。

```
1 {  
2   "token": "cb5118da1ab342aa947717dc53544259",  
3   "name": "vehicle.moving",  
4   "description": "Vehicle is moving."  
5 },
```

nuScenes

instance.json

- 存储某一 scene 中标注的实例信息，如载具、行人等，每个 instance 可能在同一 scene 的不同 sample 中都有标注。

```
1 {  
2   "token": "eb26858eca654e749a1494d0c77f418e",  
3   "category_token": "1fa93b757fc74fb197cdd60001ad8abf",  
4   "nbr_annotations": 9,  
5   "first_annotation_token": "b53021db47ca44a8bb8e2d1f9bc86b41",  
6   "last_annotation_token": "9de935c1ff134ecd8c2317cdcd87a13f"  
7 },
```



nuScenes

sample_annotation.json

- 记录从 sample 关键帧中标注的 bounding box 信息,
- 以及对应的实例 (instance)、属性 (attribute) 以及可见度 (visibility)。一个 sample 关键帧可以拥有多个 annotation。

```
1 {
2   "token": "70aecbe9b64f4722ab3c230391a3beb8",
3   "sample_token": "cd21dbfc3bd749c7b10a5c42562e0c42",
4   "instance_token": "6dd2cbf4c24b4caeb625035869bca7b5",
5   "visibility_token": "4",
6   "attribute_tokens": [
7     "4d8821270b4a47e3a8a300cbec48188e"
8   ],
9   "translation": [
10    373.214,
11    1130.48,
12    1.25
13  ],
14  "size": [
15    0.621,
16    0.669,
17    1.642
18  ],
19  "rotation": [
20    0.9831098797903927,
21    0.0,
22    0.0,
23    -0.18301629506281616
24  ],
25  "prev": "a1721876c0944cdd92ebc3c75d55d693",
26  "next": "1e8e35d365a441a18dd5503a0ee1c208",
27  "num_lidar_pts": 5,
28  "num_radar_pts": 0
29  },
```

nuLiDARtext VS nuScenes

Comparison

- nuLiDARtext 主要对 LiDAR 数据的关键帧 (sample) 的配对文本进行了优化, 如更改有歧义的缩写、修改连续性动作。Description 保存在 scene.json 中

```
1 {
2   "token": "c5224b9b454b4ded9b5d2d2634bbda8a",
3   "log_token": "8ff48ad1df8e4966a2151730c92b7f3c",
4   "nbr_samples": 40,
5   "first_sample_token": "c1676a2feac74eee8aa38ca3901787d6",
6   "last_sample_token": "63c24b51feb94f14bec29022dae4975d",
7   "name": "scene-0796",
8   "description": "Scooter, peds on sidewalk, bus, cars, truck, fake construction worker, bicycle, cross intersection, car overtaking us"
8   "description": "Scooter, pedestrians, bus, truck, bicycle, cross intersection, car"
9 },
```



nuLiDARtext

Range View 映射

1. 从笛卡尔坐标转换为球坐标:

- 原始的LiDAR点云数据是以三维笛卡尔坐标表示的, 即每个点用 $r = (x, y, z) \in \mathbb{R}^3$ 表示。
- 通过公式将其转换为球坐标 $s = (\theta, \phi, d)$:
 - 深度 d 是每个点与LiDAR传感器之间的距离, 公式为 $d = \sqrt{x^2 + y^2 + z^2}$ 。
 - 倾角 θ 是指LiDAR光束的仰角, 公式为 $\theta = \arccos(z/\sqrt{x^2 + y^2 + z^2})$ 。
 - 方位角 ϕ 是与x轴的夹角, 公式为 $\phi = \text{atan2}(y, x)$ 。

2. 深度转换为对数形式:

- 深度 d 在某些情况下被进一步转换为对数表示 $d = \log(d + 1) / \log(d_{\max} + 1)$, 将其映射到 $[0, 1]^{H \times W}$ 的图像空间中。
- 这种转换方式可以更好地处理深度值范围的差异, 特别是当深度值跨度较大时。

3. Intensity 通道的处理:

- 除了深度通道外, 还提到了对LiDAR点云数据的强度 (intensity) 通道进行类似的处理。这样就形成了一个“双通道等距矩形图像”, 其中一个通道代表深度, 另一个通道代表强度。

nuLiDARtext

Range View 映射

对于 32 beam 的 LiDAR 设备，每一帧 range view 的 shape 为 (32, 1024, 2).

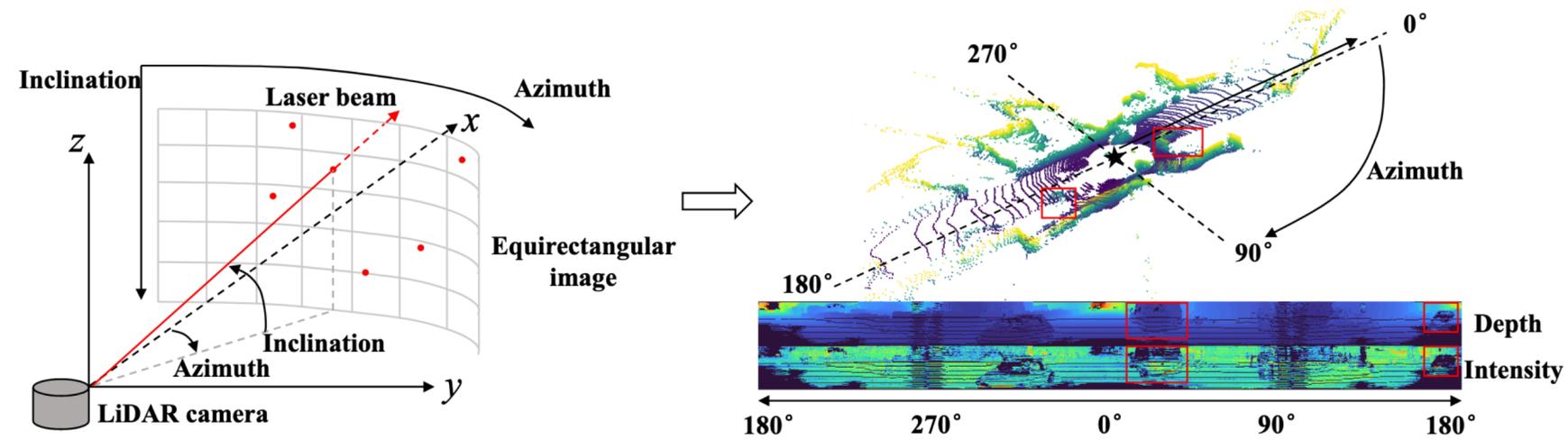


Figure 4: Schematic diagram of the data representation process.

nuLiDARtext

傅立叶映射

$$p = c * [2^{\text{arange}(\log_2 64)}, 2^{\text{arange}(\log_2 1024)}],$$

Fourier positional features f :

$$f = [\sin(p), \cos(p), \sin(2p), \cos(2p), \dots, \sin(2^{m-1}p), \cos(2^{m-1}p)]^T, \quad (2)$$

where m can be used to adjust the range of frequency components and is set to 1 in our approach. By incorporating Fourier coordinate features into the equirectangular image, we can leverage the object contour prior knowledge brought by depth information more effectively. At the same time, it enlarges the gap between adjacent pixels, which is more conducive for learning in our MLP-based transformer denoising network. Table 1 demonstrates the advantages of our method when using Fourier positional features. From the table, it can be observed that employing the Fourier positional features designed in this paper effectively enhances the model performance, with improvements of 3.35, 167.56, 0.55×10^{-4} , and 1.02×10^{-2} for the four metrics, respectively.

- c 代表映射到 range view 之后的每个像素的二维坐标
- $*$ 代表卷积过程
- arange 代表根据参数从 0 开始步长为 1 进行取整