Astyx Dataset HiRes2019 specification

Table of contents

1 Intention

This document specifies the dataset format and sensor setup of the “Astyx Dataset HiRes2019”.

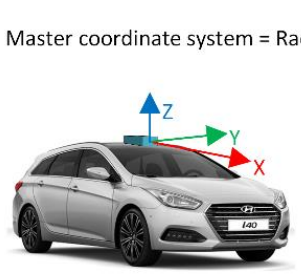
- Website for download: www.astyx.de
- Contact: info@astyx.de

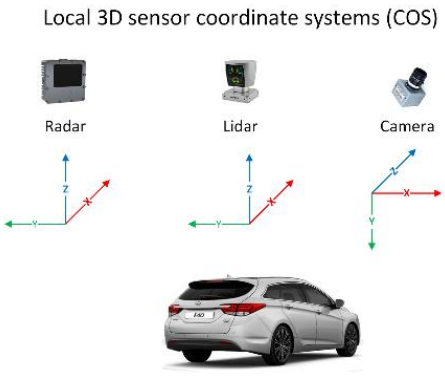
2 Sensor setup

- Camera
  - Blackfly 3.2 MP Color GigE PoE
  - Fujinon HF12XA-5M Objektiv
  - HFOV = 33 deg, VFOV = 25 deg
- Radar
  - HiRes1 (= HiRes 6455)
  - HFOV = 110 deg, VFOV = 10 deg
- Lidar
  - Velodyne VLP-16
  - HFOV = 360 deg , VFOV = 30.0 deg

2.1 Coordinate systems

- Automatic Radar ↔ Lidar ↔ Camera calibration is done using Astyx internal tooling





3 Disk storage interface

3.1 Dataset structure

- All data belonging to one recording is stored in the same parent directory, with each "sensor" residing in a separate folder.
- The number of sensors and their folder names are variable and optional (specification in file "dataset_info.json").
- Labeled groundtruth (e.g. 3D bounding box, but could be also 2D pixelwise segmentation, or 3D pointwise segmentation) is stored in a separate (and optional) folder as well and also counts as a "sensor".
- Data is assumed to be synchronized (either by hardware trigger or via timestamps on software side) up to a feasible accuracy.
- File names are typically integer based, starting from 0 with format %06d, depicting the frame index of the dataset recording.
- Each "sensor" has a 4x4 transformation called "T_to_ref_COS" defining the 6DoF transformation of all sensor data into a common coordinate system.

Example:

```
astyx_dataset_demo/
  dataset_info.json

  calibration/
    000000.json
    000001.json
    ...
  lidar_os1/
    000000.txt
    000001.txt
    ...
  camera_front/
    000000.png
    000001.png
    ...
  radar_hires1/
    000000.txt
    000001.txt
    ...
  radar_shortrange/
    000000.txt
    ...
  radar_polarimetric/
    000000.txt
    ....
  groundtruth_obj3d/
    000000.txt
    000001.txt
    ...
```

3.2 Dataset info file

- Stores the mapping of "sensor UIDs" to the corresponding data folder on a per-frame basis.

Example:

```
{
  "dataset_type": "Astyx_HiRes",        # Defines the dataset type
  "sensors": [                          # Overall information about the sensors
    {
      "data_reader": "DataReader_Radar_HiRes3D",  # Which datareader to use (therefore which
                                                  # format of the sensor data)
      "sensor_flags": "",                         # For separating multiple sensors of the same
                                                  # type (e.g. 3D object groundtruth and
                                                  # detections, or two radar sensors)
      "sensor_type": "radar",                     # Specifying the type of the sensor
      "sensor_uid": "radar_hires1"                # Each sensor is defined via a unique ID (which
                                                  # is used in the per-frame data)
    },
    {
      "data_reader": "DataReader_Lidar",
      "sensor_flags": "",
      "sensor_type": "lidar",
      "sensor_uid": "lidar_os1"
    },
    {
      "data_reader": "DataReader_Camera",
      "sensor_flags": "",
      "sensor_type": "camera",
      "sensor_uid": "camera_bfly"
    },
    {
      "data_reader": "DataReader_Calibration_Astyx",
      "sensor_flags": "",
      "sensor_type": "calibration",
      "sensor_uid": "calibration"
    },
    {
      "data_reader": "DataReader_Obj3D_Astyx",
      "sensor_flags": "groundtruth",
      "sensor_type": "labels_object3d",
      "sensor_uid": "groundtruth_obj3d"
    }
  ]
  "data": {
                # Sensor data (per frame) is stored in this dict with frame_index as key
                # Specified paths are relative to the corresponding json file
                # Entries with frame index of the specified data in this dataset.
                # Should be 0-based and unique.
                # Frame with index 0 contains data for these sensors, pointing to specific
                # files (=dict with sensor UID as key)
                # Sensor UIDs need to match the ones defined before in field "sensors"
    "0": {
        "calibration": "calibration/000000.json",
        "camera_bfly": "camera_bfly/000000.jpg",
        "groundtruth_obj3d": "groundtruth_obj3d/000000.json",
        "lidar_os1": "lidar_os1/000000.txt",
        "radar_hires1": "radar_hires1/000000.txt"
      }
    },
    "1": {
      "frame_index": 1,
      "sensors": {
        "calibration": "calibration/000283.json",
        "camera_bfly": "camera_blfy/000283.jpg",
        "groundtruth_obj3d": "groundtruth_obj3d/000283.json",
        "lidar_os1": "lidar_os1/000283.txt",
        "radar_hires1": "radar_hires1/000283.txt"
      }
    }
  },
}
```

3.3 File format calibration data

- Each "sensor" has a 4x4 transformation called "T_to_ref_COS" defining the 6DoF transformation of all sensor data into a common coordinate system.
- One calibration file per frame, because the dataset could originate from different merged recordings.

Example:

```
{
    "sensors": [   # dict for sensor calibration data with sensor UID as key
        {
            "sensor_uid": "radar_hires1"
            "calib_data": {
                "T_to_ref_COS": [     # 4x4 transformation as relative 6DoF transformation
                                      # to the common reference coordinate system
                    [
                        1.0,
                        0.0,
                        0.0,
                        0.0
                    ],
                    [
                        0.0,
                        1.0,
                        0.0,
                        0.0
                    ],
                    [
                        0.0,
                        0.0,
                        1.0,
                        0.0
                    ],
                    [
                        0.0,
                        0.0,
                        0.0,
                        1.0
                    ]
                ]
            },
        },
        {
            "sensor_uid": "lidar_os1"
            "calib_data": {
                "T_to_ref_COS": [
                    [
                        0.9982015,
                        0.04843654,
                        -0.03532153,
                        -0.1326520881017172
                    ],
                    [
                        -0.04609754,
                        0.99686721,
                        0.06427134,
                        0.0157771060919497
                    ],
                    [
                        0.03832396,
                        -0.06252752,
                        0.99730719,
                        0.11511798526730879
                    ],
                    [
                        0.0,
                        0.0,
                        0.0,
                        1.0
                    ]
                ]
```

```
            },
        },
        {
            "sensor_uid": "camera_bfly"
            "calib_data": {
                "K": [                # Additional, sensor-specific calibration data
                                      # (here 3x3 camera intrinsics)
                    [
                        1817.98103,
                        0.0,
                        1040.27484
                    ],
                    [
                        0.0,
                        1816.83987,
                        319.497539
                    ],
                    [
                        0.0,
                        0.0,
                        1.0
                    ]
                ],
                "T_to_ref_COS": [
                    [
                        0.015721251542359097,
                        0.0388038506130491,
                        0.9991231397751291,
                        -0.04286578070231163
                    ],
                    [
                        -0.9983693562813132,
                        -0.054233366081888695,
                        0.017815695191944 3,
                        -0.0019011493702534147
                    ],
                    [
                        0.054877145205922295,
                        -0.9977740503554758,
                        0.0378879601312635,
                        0.011313703630010682
                    ],
                    [
                        0.0,
                        0.0,
                        0.0,
                        1.0
                    ]
                ]
            },
        }
    ]
}
```

3.4 File format - camera data

- jpg images
- 2048 x 618 x RGB

3.5 File format - radar data

- txt/csv format
- 1 row = 1 point (5D): [X, Y, Z, V_r (relative radial velocity), magnitude]

3.6 File format - lidar data

3.6.1 bin format

- 4D pointcloud stored in binary format (32 bit float): [X, Y, Z, Intensity]
- Points successively stored in array: 4*32Bit * num_points

3.6.2 txt/csv format

- 1 row = 1 point (4D): [X, Y, Z, Intensity]
- or
- 1 row = 1 point (6D): [X, Y, Z, Reflectivity, LaserID, Timestamp]

3.7 File format - groundtruth data - 3D objects

- All labeled or detected 3D objects in one frame (plus meta information) are stored in a json file

Identifier	Type	Description
Per file		
frame_index	int	Index of the file w.r.t. the dataset (0-based)
frame_flag	string	Containing optional information of the current frame (weather, who should label that frame, ...)
header	string	Containing information about the dataset version
Per object		
classname	string	Describes the type of object. E.g. 'car', 'pedestrian', etc
center3d	3x float	3D object position x,y,z (center of 3D bounding box) in master sensor coordinate system (in meters)
dimension3d	3x float	3D object dimensions: width, length, height (in meters). Distinction of width and length is given by the parameter orientation / the looking vector.
orientation_quat	4x float	3DoF rotation as quaternion in master sensor coordinate system. Forward facing cars (same direction as ego vehicle) have orientation_quat = (1,0,0,0).
score	float	Confidence of detection [0..1], needed for Precision/Recall curves. The higher the better. Only for detections, can be missing in groundtruth.
object_id	int	Object wise ID (for tracking/detecting one specific object in a sequence). Must be unique per frame. 0-based. If not available / labeled: -1.

occlusion	int	occlusion state (0=OCCLUSION_NONE, 1=OCCLUSION_PARTLY, 2=OCCLUSION_LARGELY, 3=OCCLUSION_FULLY)
label_certainty	int	Certainty of box dimension (0=DIM_CERTAINTY_SURE, 1=DIM_CERTAINTY_SEMISURE, 2=DIM_CERTAINTY_UNSURE)
measured_by	dict	Dictionary with sensor having measured (or not measured) the objects. Example: { "camera": 1, "lidar": 1, "radar": 0 }
created_by	string	String representation of where this label comes from / was generated from

3.7.1 Example (with 2 objects)


```
{
    "frame_index": 0,
    "frame_flag": 0,
    "header": "Astyx Version_01_00",
    "objects": [
        {
            "center3d": [
                11.683995388198515,
                -0.9560522831965559,
                0.7298276901344742
            ],
            "classname": "Car",
            "created_by": "bob",
            "dimension3d": [
                4,
                1.8,
                1.5
            ],
            "label_certainty": 0,
            "measured_by": {
                "camera": 1,
                "lidar": 1,
                "radar": 0
            },
            "object_id": -1,
            "occlusion": 0,
            "orientation_quat": [
                [
                    0.995150944420393,
                    -0.09367697718285199,
                    -0.016925052374905138,
                    -0.02475407778912337
                ]
            ],
            "score": -1
        },
        {
            "center3d": [
                27.436975052376937,
                2.9250727400954366,
                0.7298276901344565
            ],
            "classname": "Car",
```

```
            "created_by": "alice",
            "dimension3d": [
                4,
                1.8,
                1.5
            ],
            "label_certainty": 0,
            "measured_by": {
                "camera": 1,
                "lidar": 1,
                "radar": 0
            },            "object_id": -1,
            "occlusion": 0,
            "orientation_quat": [
                [
                    0.9990465311984655,
                    -0.031730218014223065,
                    -0.01842883468904623,
                    -0.023655862213985912
                ]
            ],
            "score": -1
        }
    ]
}
```

4 Versioning

Version: 1.0

Author(s): Georg Kuschk

Number of pages: 10

State of document: **Final**

Document history

Date	Version	Name	Short Description / changed chapters
25.09.2019	1.0	Georg Kuschk	Initial Version