

# Astyx Dataset specification

## Table of contents

1	Intention .....	2
2	Sensor setup.....	2
2.1	Coordinate systems.....	2
3	Disk storage interface .....	3
3.1	Dataset structure .....	3
3.2	Dataset info file .....	4
3.3	File format – Calibration data .....	5
3.4	File format – Camera data .....	7
3.5	File format – Lidar data.....	7
3.5.1	bin format.....	7
3.5.2	txt/csv format .....	7
3.6	File format – Radar data.....	7
3.7	File format – Meta data.....	7
3.8	File format - CAN data .....	8
3.9	File format - GPS data.....	8
3.10	File format - Groundtruth data - 3D objects.....	9
3.10.1	Example (with 2 objects).....	10
4	Versioning .....	11

# 1 Intention

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

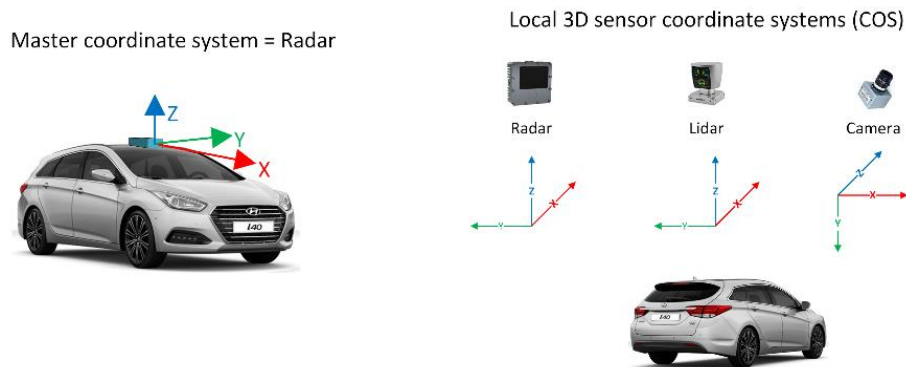
- Website for download: [www.astyx.de](http://www.astyx.de)
- Contact: [info@astyx.de](mailto:info@astyx.de)

# 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
- GPS/IMU
  - WORK IN PROGRESS / IN CLARIFICATION
  - Properties identical or similar to: <https://www.advancednavigation.com/de/product/spatial-dual>

## 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  
  
  metadata/  
    000000.json  
    000001.json  
    ...  
  calibration/  
    000000.json  
    000001.json  
    ...  
  canbus/  
    000000.json  
    000001.json  
    ...  
  camera_front/  
    000000.png  
    000001.png  
    ...  
  lidar_os1/  
    000000.txt  
    000001.txt  
    ...  
  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\_info.json:

```
{
  "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_bfly/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 calibration\_data.json:

```
{
  "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,  
      0.0178156951919443,  
      -0.0019011493702534147  
    ],  
    [  
      0.054877145205922295,  
      -0.9977740503554758,  
      0.0378879601312635,  
      0.011313703630010682  
    ],  
    [  
      0.0,  
      0.0,  
      0.0,  
      1.0  
    ]  
  ]  
},  
]
```

### 3.4 File format – Camera data

- png images (lossless compression)
- 2048 x 618 x RGB
- Images corrected for lens distortion (undistorted)

### 3.5 File format – Lidar data

#### 3.5.1 bin format

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

#### 3.5.2 txt/csv format

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

### 3.6 File format – Radar data

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

### 3.7 File format – Meta data

- Framewise meta data is stored as a .json for every frame in the extensible format shown below with the current properties:
- **Weather:** sunny, cloudy, raining, snowing, OTHER
- **Location:** downtown, residential area, highway, overland, parking lot, tunnel, OTHER

Example meta\_data.json:

```
{
  "generic_string_properties":
  [
    {"name": "scene", "value": "highway"},
    {"name": "weather", "value": "sunny"}
  ]
}
```

### 3.8 File format - CAN data

- CAN data is stored as a .json for every timestamp in following format. The datatypes of the entries depend on the signals. For a mapping see CAN data.
- The timestamp is the frame timestamp, all other values are the nearest neighbors (temporally) of the radar timestamp.

Example can\_data.json:

```
{  
  "GPS_A_Lat": 1.46900000000000012,  
  "GPS_A_Lon": 0.23799999999999955,  
  "GPS_Altitude": 563.15,  
  "GPS_Banking": -3.5999999999994543,  
  "GPS_CourAccu": 4.03,  
  "GPS_Course": 285.07,  
  "GPS_GDOP": 1.85,  
  "GPS_HDOP": 0.86,  
  "GPS_HorAccu": 0.41200000000000003,  
  "GPS_Lat_dez": 48.0538009,  
  "GPS_Latitude": 48.0322805,  
  "GPS_Lon_dez": 11.650628399999999,  
  "GPS_Longitud": 11.390376999999999,  
  "GPS_PDOP": 1.6500000000000001,  
  "GPS_SpAccu": 0.5399999999999999,  
  "GPS_Speed_3D": 10.979999999999999,  
  "GPS_VDOP": 1.4000000000000001,  
  "GPS_V_Sat": 10.51,  
  "GPS_VerAccu": 0.613,  
  "GPS_Yawrate": 28.870000000000005,  
  "LAT_ACCEL": -0.6600000000000001,  
  "LONG_ACCEL": 0.8900000000000006,  
  "Timestamp": 1561533360.526821,  
  "YAW_RATE": -16.680000000000003,  
  "vehicle_speed": 10.0  
}
```

### 3.9 File format - GPS data

- Work in progress, currently in clarification



### 3.10 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
<b>Per file</b>		
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
<b>Per object</b>		
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: length, width, 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

### 3.10.1 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
      ]
    },
    {
      "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
      ]
    }
  ]
}
```

## 4 Versioning

Version: 1.2  
Author(s): Georg Kuschk  
Number of pages: 11

State of document: **Draft**

### Document history

Date	Version	Name	Short Description / changed chapters
25.09.2019	1.0	Georg Kuschk	Initial Version
08.11.2019	1.1	Georg Kuschk	Added GPS and meta data specs
31.01.2020	1.2	Georg Kuschk	Fixed order of dimension3d