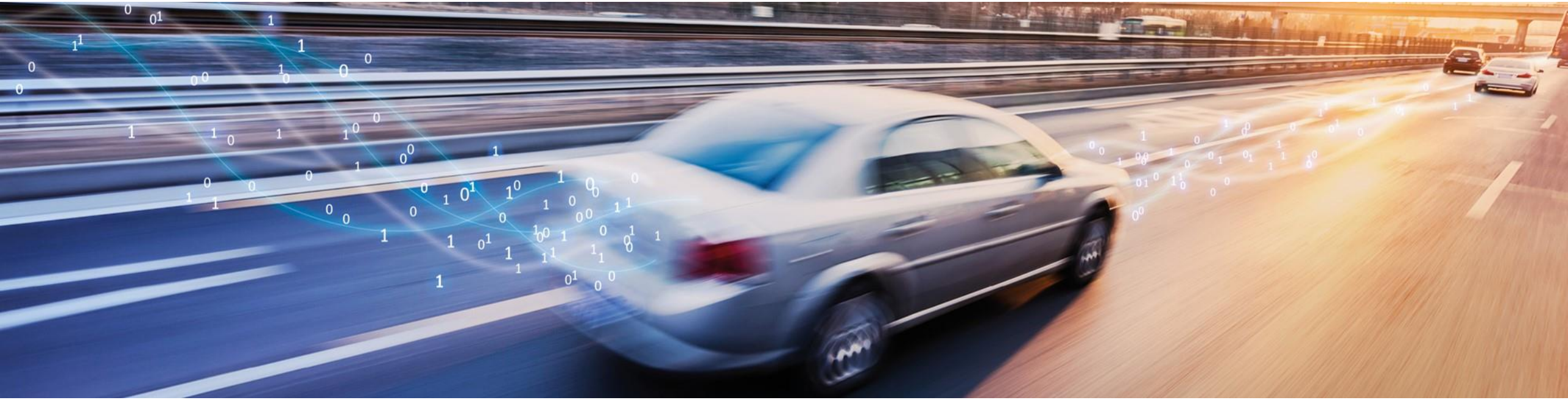
ASAM OpenLABEL V1.0.0 Webinar

Multi-Sensor Data Labeling and Scenario Tagging

Marcos Nieto
Principal Researcher, Vicomtech
OpenLABEL V1.0.0 Data format

17. November 2021
München



ASAM Association for Standardization of Automation and Measuring Systems

Agenda

OpenLABEL V1.0.0 Release Webinar

1	**Introduction to Data Format**
2	**OpenLABEL JSON Schema**
3	**Overview OpenLABEL JSON Structure**
4	**Data Types**

Introduction to Data Format

OpenLABEL V1.0.0 Release Webinar

ASAM

Introduction to Data Format

https://openlabel.asam.net/V1-0-0/schema/openlabel_json_schema.json

Annotation format

- Defines the structure of annotations, data types and conventions needed to unambiguously interpret the annotations

Annotation file format

- Specifies how the annotation data is encoded for storage into computer files

JSON schema

- Defines the structure, data types and restrictions of annotations as JSON (JavaScript Object Notation) data for validation

OpenLABEL JSON Schema

OpenLABEL V1.0.0 Release Webinar

ASAM

OpenLABEL JSON Schema

https://openlabel.asam.net/V1-0-0/schema/openlabel_json_schema.json

The Schema defines:

- The terms of the labels (keys)
- The types of the values
- The structure/hierarchy
- Restrictions (required/optional) and conditions of keys and values

The Schema is used to:

- Validate JSON files
- Provide formal documentation
- Be parsed by programming languages to build object data (e.g. classes in OOP)

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "additionalProperties": false,
    "definitions": {
        "action": {
            "additionalProperties": false,
            "properties": {
                "frame_intervals": {
                    "item": {
                        "$ref": "#/definitions/frame_interval"
                    },
                    "type": "array"
                },
                "name": {
                    "type": "string"
                },
                "ontology_uid": {
                    "type": "integer"
                },
                "stream": {
                    "type": "string"
                },
                "type": {
                    "type": "string"
                }
            },
            "required": [
                "name",
                "type"
            ],
            "type": "object"
        },
        "area_reference": {
            "properties": {
                "additionalProperties": false,
                "attributes": {
                    "$ref": "#/definitions/attributes"

...
```

Overview OpenLABEL JSON Structure

OpenLABEL V1.0.0 Release Webinar

ASAM

Overview OpenLABEL JSON Structure

Basic JSON file

OpenLabel JSON

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        }
    }
}
```

“openlabel” root key

Overview OpenLABEL JSON Structure

Basic JSON file – root level

OpenLabel root level

Keys defined in **JSON schema**

Values subject to schema types and properties

Overview OpenLABEL JSON Structure

Basic JSON file – object

OpenLabel JSON simple **object**

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "objects": {
            "0": {
                "name": "car1",
                "type": "Car",
            }
        }
    }
}
```

```
{
   "openlabel": {
       "metadata": {
           "schema_version": "1.0.0"
       },
       "objects": {
          "c44c1fc2-ee48-4b17-a20e-829de9be1141": {
              "name": "van1",
              "type": "Van"
          },
       }
   }
}
```

uid unique identifier as numerical string

name of the instance
type of the class

uuid Universal Unique Identifier

Overview OpenLABEL JSON Structure

Basic JSON file – object class from ontology

OpenLabel JSON simple object

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "ontologies": {
            "0": "https://www.somedomain.org/ontology",
            "1": "https://openlabel.asam.net/V1-0-0/ontologies/openlabel_ontology_scenario_tags.ttl"
        },
        "objects": {
            "0": {
                "name": "car1",
                "type": "VehicleCar",
                "ontology_uid": "1",
            }
        }
    }
}
```

`ontologies` can contain as many ontologies as needed

`ontology_uid` specifies the ontology that contains the definition of the `type` of objects and other elements

Overview OpenLABEL JSON Structure

Basic JSON file – tags

OpenLabel JSON simple tag

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0",
            "tagged_file": "../resources/scenarios/scenario_file",
        },
        "ontologies": {
            "0": {
                "uri": "https://code.asam.net/simulation/standard/openxontology/ontologies/openlabel",
                "boundary_list": ["motorway", "road"],
                "boundary_mode": "include"

            },
            "1": "http://mycompany/ontologies/v1",
        },
        "tags": {
            "0": {
                "type": "DoubleRoundabout",
                "ontology_uid": "1",
                "tag_data": {
                    "num": [{"type": "value", "val": 2}]
                }
            }
        }
    }
}
```

`tagged_file` specifies which (e.g. scenario) file is being tagged

`ontologies` can be specified and sub-defined via `boundary_list` and `boundary_mode`

Overview OpenLABEL JSON Structure

Basic JSON file – object with object_data

OpenLabel JSON simple **object** with **object_data**


```
{
   "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "objects": {
           "0": {
               "name": "pedestrian1",
               "type": "Pedestrian",
               "object_data": {
                   "bbox" : [{
                             "name" : "body",
                             "val" : [303.73, 935.58, 135.62, 330.88]
                         }, {
                             "name" : "head",
                             "val" : [289.93, 814.08, 38.20, 39.96]
                         }
                   ]
               }
           }
        }
   }
}
```

bbox array, to allow multiple entries per object

object_data contains data about object

name of bbox
val (x, y, w, h)

Overview OpenLABEL JSON Structure

Basic JSON file – object with object_data with attributes

OpenLabel JSON simple **object** with **object_data** with **attributes**

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "objects": {
            "0": {
                "name": "car1",
                "type": "Car",
                "object_data": {
                    "bbox" : [{
                            "name" : "shape",
                            "val" : [100, 100, 500, 300],
                            "attributes": {
                                "boolean": [{
                                        "name": "visible",
                                        "val": true
                                    },
                                    {
                                        "name": "interpolated",
                                        "val": false
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

object_data contains data about object

attributes of bbox

Overview OpenLABEL JSON Structure

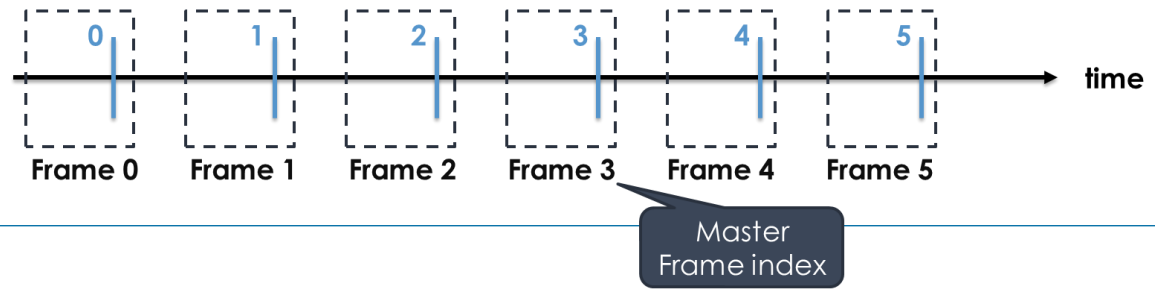
Basic JSON file – frames

OpenLabel JSON simple **frames**

```
{
   "openlabel": {
       "metadata": {
           "schema_version": "1.0.0"
       },
       "frame_intervals": [{
               "frame_start": 0, "frame_end": 1
           }, {
               "frame_start": 5, "frame_end": 7
           }
       ],
       "frames": {
           "0": { ... },
           "1": { ... },
           "5": { ... },
           "6": { ... },
           "7": { ... }
       }
   }
}
```

Frame_intervals ranges of frame numbers with data

frames contains dictionary of frames



Overview OpenLABEL JSON Structure

Basic JSON file – frames with object_data

OpenLabel JSON object_data at frame

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "frames": {
            "0": {
                "objects": {
                    "1": {
                        "object_data": {
                            "bbox": [{
                                    "name": "shape",
                                    "val": [12, 867, 600, 460]
                                }
                            ]
                        }
                    }
                }
            },
            "1": { ... }
        },
        "objects": {
            "1": {
                "name": "van1",
                "type": "Van",
                "frame_intervals": [{"frame_start": 0, "frame_end": 1}]
            },
        }
    }
}
```

bbox specified for frame "0" for object "1"

Overview OpenLABEL JSON Structure

Basic JSON file – streams

OpenLabel JSON **streams**

```
{
   "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "streams": {
            "Camera1": {
                "type": "camera",
                "uri": "./some_path/some_video.mp4",
                "description": "Frontal camera",
                "stream_properties": {
                    "intrinsics_pinhole": {
                        "camera_matrix_3x4": [ 1000.0,    0.0, 500.0, 0.0,
                                               0.0, 1000.0, 500.0, 0.0,
                                               0.0,    0.0,   0.0, 1.0],
                        "distortion_coeffs_1xN": [],
                        "height_px": 480,
                        "width_px": 640
                    },
                }
            }
        }
    }
}
```

streams contain dictionary of sequences of data subject of labeling

stream_properties define properties of stream **"Camera1"**

intrinsics_pinole
intrinsics_fisheye
intrinsics_custom

ASAM

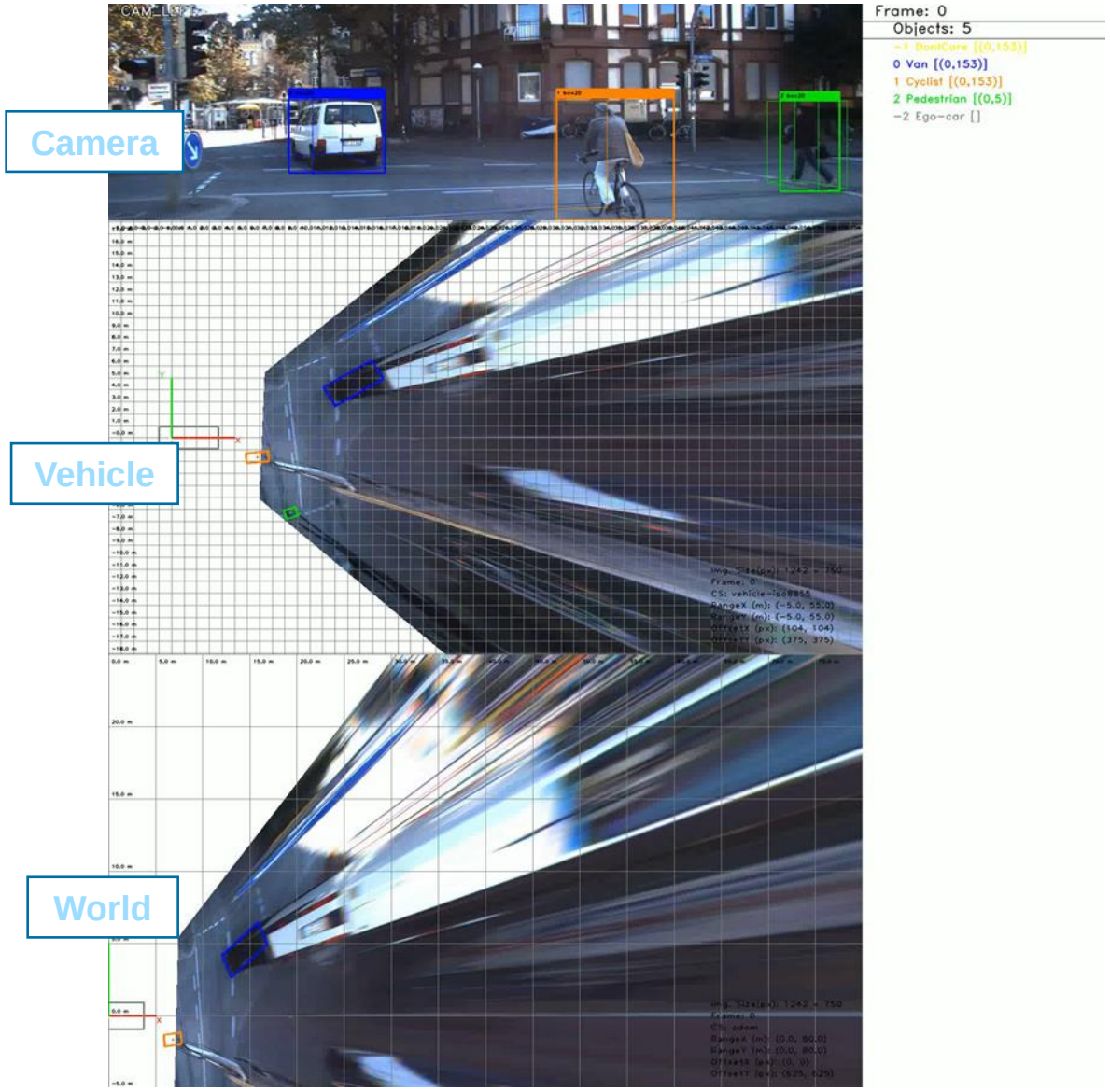
Overview OpenLABEL JSON Structure

Basic JSON file – coordinate_systems and transforms

```
{
    "openlabel": {
        "metadata": {
            "schema_version": "1.0.0"
        },
        "coordinate_systems": {
            "odom": {
                "type": "scene_cs",
                "parent": "",
                "children": [
                    "vehicle-iso8855"
                ]
            },
            "vehicle-iso8855": {
                "type": "local_cs",
                "parent": "odom",
                "children": [
                    "CAM_1",
                    "CAM_2"
                ]
            },
            "CAM_1" : {
                "type" : "sensor_cs",
                "parent" : "base",
                "children" : [],
                "pose_wrt_parent" : {
                    "matrix4x4" : [0.984807753012208, 0.0, 0.17364817766693033, 2.3, 0.0, 1.0, 0.0, 0.0, -
0.17364817766693033, 0.0, 0.984807753012208, 1.3, 0.0, 0.0, 0.0, 1.0]
                }
            },
            "CAM_2" : {
                "type" : "sensor_cs",
                "parent" : "base",
                "children" : [],
                "pose_wrt_parent" : {
                    "euler_angles" : [0.0, 0.17453292519943295, 0.0],
                    "translation" : [2.3, 0.0, 1.3],
                    "sequence" : "ZYX"
                }
            }
        },
        ...
    }
}
```

coordinate_systems contain dictionary of defined coordinate systems

Hierarchy of **"parent"** and **"children"** poses



ASAM

OpenLABEL Data Types

OpenLABEL V1.0.0 Release Webinar

ASAM

OpenLABEL Data Types

Generic data types

OpenLabel JSON generic data types: object_data

```
"num": [{
    "name": "height_m",
    "val": 1.98
}]
```

```
"text": [{
    "name": "license plate",
    "val": "8440CMN"
}]
```

```
"vec": [{
    "name": "scores",
    "val": [0.98, 0.76, 0.98]
}]
```

```
"vec": [{
    "name": "locations",
    "val": ["Madrid", "Paris", "Rome"]
}]
```

```
"boolean": [{
    "name": "visible",
    "val": true
}]
```

OpenLabel JSON data type extension

```
"num": [{
    "name": "height_m",
    "val": 1.98
}]
```

```
"num": [{
    "name": "height_m",
    "val": 1.98,
    "coordinate_system": "WORLD",
    "attributes": {
        "num": [{
            "name": "confidence",
            "val": 0.98
        }]
    },
    "custom_prop1": "SomeValue",
    "custom_prop2": 0.99
}]
```

Any **object_data** *can* have other generic **object_data** as attributes

All **object_data** *can* be specified to correspond to specific **coordinate_systems**

All **object_data** *can* have custom additional properties

OpenLABEL Data Types

Generic geometric data types

OpenLabel JSON geometric data types: object_data

```
"bbox": [{
    "name": "head",
    "val": [400, 200, 100, 120]
}]
```

```
"rbbox": [{
    "name": "outline",
    "val": [400, 200, 100, 120, 0.785]
}]
```

```
"cuboid": [{
    "name": "shape",
    "val": [12.0, 20.0, 0.0, 1.0, 1.0, 1.0, 1.0,
 4.0, 2.0, 1.5]
}]
```

```
“image": [{
    "name": "color",
    "val": "iVBORw0KGgoAA..."
    “mime_type": "image/png",
    “encoding": "base64"
}]
```

```
“mat": [{
    "name": “points3d_4xN",
    "val": [...].
    “width": 325,
    “height": 4,
    “channels": 1,
    “data_type": “uint32",
}]
```

```
“image": [{
    "name": "color",
    "val": "iVBORw0KGgoAA..."
    “mime_type": "image/png",
    “encoding": "base64"
}]
```

```
"poly2d": [
    {
        "name": "poly1",
        "val": ["5","5","1","mBIIOIII"],
        "mode": "MODE_POLY2D_SRF6DCC",
        "closed": false
    }, {
        "name": "poly2",
        "val": [5,5,10,5,11,6,11,8,9,10,5,10,3,8,3,6,4,5],
        "mode": "MODE_POLY2D_ABSOLUTE",
        "closed": false
    }
]
```

```
"poly3D" : [{
    "closed" : false,
    "coordinate_system" : "vehicle_iso8855",
    "name" : "lane_marking",
    "val" : [557.02, 29.69, -1.63, 562.51, 29.97, -
1.59, 568.00, 30.36, -1.58, 571.98, 30.76, -1.57]
}]
```

```
“point2d": [{
    "name": “A2",
    "val": [400, 200],
    “id": 1
}]
```

```
“point3d": [{
    "name": “A3",
    "val": [400, 200, 0],
    “id": 1
}]
```

```
"mesh" : [{
    "name" : "parkslot1",
    "point3d" : {
        "0" : {
            "name" : "Vertex0",
            "val" : [25, 25, 0],
        },
        …
    },
    "line_reference" : {
        "0" : {
            "name" : "Edge",
            "reference_type" : "point3d",
            "val" : [0, 1],
        },
        "1" : {
            "name" : "Edge",
            "reference_type" : "point3d",
            "val" : [1, 2],
        },
        …
    },
    "area_reference" : {
        "0" : {
            "name" : "Slot",
            "reference_type" : "line_reference",
            "val" : [0, 1, 2, 3],
        },
        "1" : {
            "name" : "Slot",
            "reference_type" : "line_reference",
            "val" : [4, 5, 6, 1],
        }
    }
}]
```

Thank you for your attention!

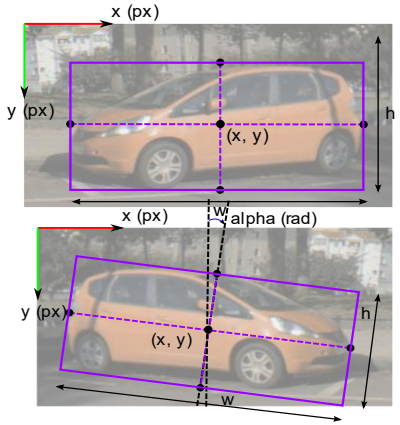
Marcos Nieto
Principal Researcher - Vicomtech
ASAM OpenLABEL V1.0.0 Data Format

Email: mnieto@vicomtech.org

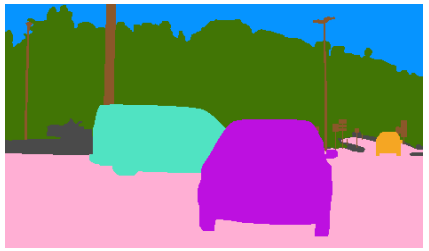
OpenLABEL Data Types

Use cases

bbox
rbbox



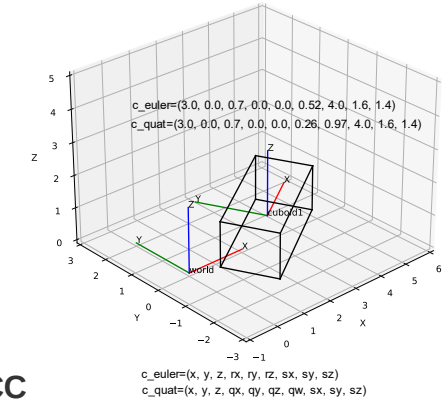
poly2d



Lossless SRF6DCC algorithm

The **ASAM OpenLABEL SRF6DCC** approach produces a **63 kB JSON file** for a 2048x1536 image

cuboid

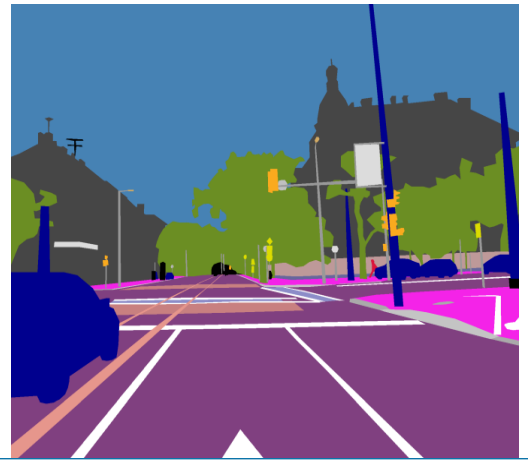


binary

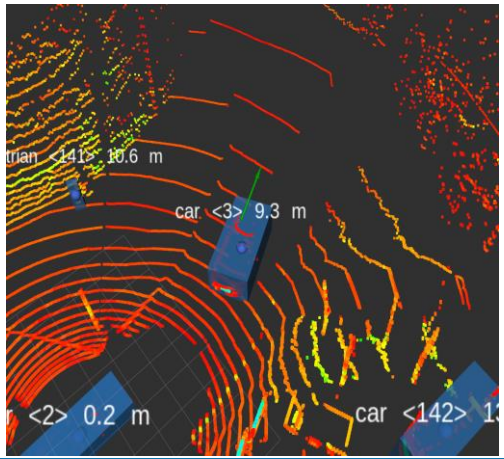
Lossless RLE-based binary data

The **ASAM OpenLABEL RLE-based** approach produces a **88 kB JSON file** from ~89 MB CSV files

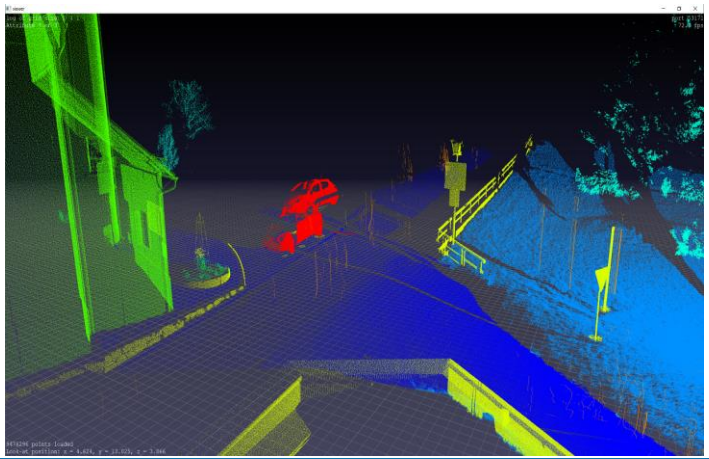
2D Objects - Bounding Box



2D Objects - Segmentation



3D Objects - Cuboids



3D Objects - Segmentation

ASAM