

3DHD CityScenes: Documentation

Version: v1.0

In this document, we provide a detailed description of the data format regarding the dataset files (defining the training, validation, and test set), map items contained in the HD map, and included map meta data.

Dataset files

In the “Dataset” subdirectory, we provide train.json, val.json, and test.json, which define the samples that we use to generate point cloud crops (in the ego vehicle’s reference frame) and map items contained in this crop.

Table 1: Number of samples contained in the training, validation, and test set

Dataset partition	Number of samples
train.json	54,140
val.json	7,528
test.json	11,881

Subsequently, we describe the data format of samples contained in a respective JSON file.

Table 2: Data format description for samples

Data field	Description
ID	Sample ID, only unique within one run (e.g., OP_1)
run_name	Run name, from which the sample originates
loc_data	Dictionary of localization data
Latitude_deg	Latitude in WGS84 (degree)
Longitude_deg	Longitude in WGS84 (degree)
Altitude_m	Altitude as geodetic height, WGS84 reference ellipsoid (meter)
Yaw_deg	Orientation of the ego vehicle (in degree), counterclockwise from the x-axis
X_m	x-position in UTM (meter)
Y_m	y-position in UTM (meter)

HD Map

The map items are provided in the “HD_Map” subdirectory. In the following, we describe the data formats for all map item types. First, we provide a description for data fields that are shared by different map item types. Subsequently, we describe map-item-specific data fields.

Table 3: Data format description for shared data fields

Data field	Description
type	Type of the map item. Possible types: [TrafficSign, TrafficLight, Pole, CS_Obstacle_Point, CS_Obstacle_Line, Curb, Marking_Line, Marking_Polygon_Ordinary,

	Marking_Polygon_Negation, Marking_Polygon_Arrow, Marking_Polygon_Text, Marking_Polygon_Symbol, Concstruction_Site, Lane_Ordinary, Lane_Temporary, Relation]
id	ID of the map item. Unique also across item types.
deviation_extension	temporary: Map item is intended to be removed experimental: Map item is temporary, but without construction site inconsistent: Map item is semantically inconsistent with the scene negated: Map item is not valid
lat	Latitude in WGS84 (degree)
lon	Longitude in WGS84 (degree)
x_utm	x-coordinate in UTM32N (meter)
y_utm	y-coordiante in UTM32N (meter)
z_utm	z-coordiante in UTM32N (meter) (geodetic height using the WGS84 reference ellipsoid)
points_utm	List of tuples (x_utm, y_utm, z_utm) defining the item's position for line-shaped items (e.g., curbs, lanes, etc.)
color	Color of a marking. Possible colors: [white, yellow]

For traffic signs, x_utm, y_utm, and z_utm refer to the center point of the bounding rectangle.

Table 4: Data format description for traffic signs

Data field	Description
yaw_utm	Sign orientation in the UTM reference frame, clockwise defined. Assuming a bounding rectangle along the x-axis, rotate respective edge points by -yaw_utm to obtain the sign's orientation.
cls	Class number of the traffic sign, according to the German traffic sign catalogue: https://www.bast.de/EN/Traffic_Engineering/Subjects/verkehrszeichen/vz-start.html .
subcls	Further classification number of the traffic sign, according to the catalogue
shape	Shape of the traffic sign. Possible shapes: [circle, rectangle, triangle_up, triangle_down, diamond, arrow, octagon]
height	Height of the bounding rectangle (meter)
width	Width of the bounding rectangle (meter)
two_sided	Flag. True if the sign has two separate sides that are further apart, modeled as two separate signs with individual bounding rectangles
fused	Flag. Two sides of the same sign modeled as two separate bounding rectangles (two_sided flag set) are merged into a single sign, for which the fused flag is set. For our publications, only fused signs are considered.

For traffic lights, x_utm, y_utm, and z_utm refer to the center point of the light's face plate.

Table 5: Data format description for traffic lights

Data field	Description
yaw_utm	Orientation of the traffic light's face plate in the UTM reference frame, clockwise defined. Assuming a bounding rectangle along the x-axis, rotate respective edge points by $-\text{yaw_utm} + 90^\circ$ to obtain the light's orientation
cls	Classification of the traffic light. Possible classes: [vehicle, people]
orientation_cls	Orientation of the light's face plate. Possible orientations: [vertical, horizontal]
signal_cls	Signal class of the traffic light. Possible signals: [bike/pedestrian, round, pedestrian, bike, right_arrow, right_diagonal_arrow, straight_arrow, left_diagonal_arrow, left_arrow, straight_and_left_arrow, straight_and_right_arrow]
height	Height of the face plate (meter)
width	Width of the face plate (meter)
has_red	True if traffic light has a red light
has_green	True if traffic light has a green light
has_yellow	True if traffic light has a yellow light

For poles, x_utm, y_utm, and z_utm refer to the center of the pole's base point.

Table 6: Data format description for poles

Data field	Description
cls	Classification of the pole. Possible classes: [tree, traffic_sign_pole, bollard, lamppost, guardrail_pole, general_pole, traffic_light_pole, advertising_pillar, flag_pole, pillar, telephone_pole, billboard_pole, gantry_pole]
diameter	Diameter of the pole (meter)

For construction site obstacles (point), x_utm, y_utm, and z_utm refer to the center of the item's base point.

Table 7: Data format description for construction site obstacles (point)

Data field	Description
cls	Classification of the obstacle. Possible classes: [cone, delineator, reflector]
diameter	Diameter of the obstacle (meter)

Table 8: Data format description for construction site obstacles (line)

Data field	Description
cls	Classification of the obstacle. Possible classes: [fence, guide_board, reflectors, barrier]
width	Width of the obstacle (default value depending on class)
height	Height of the obstacle (default value depending on class)

Table 9: Data format description for construction sites

Data field	Description
points_utm	List of tuples (x_utm, y_utm, z_utm) defining a polygon surrounding the construction site

Table 10: Data format description for curbs

Data field	Description
cls	Classification of the curb. Possible classes: [high_curb, low_curb]

Table 11: Data format description for lanes (ordinary and temporary)

Data field	Description
connection_type	Connection type of the lane. Possible types: [continuation, split, merge]
width	Width of the lane (meter)
successors	List of succeeding lanes with their IDs

Table 12: Data format description for markings (line)

Data field	Description
cls	Classification of the marking. Possible classes: [pedestrian_crossing, solid_line, dashed_line, bicycle_crossing]

Table 13: Data format description for polygon-shaped markings (arrow)

Data field	Description
cls	Classification of the arrow. Possible classes: [arrow_left, arrow_straight_right, arrow_straight, arrow_right, arrow_straight_left, arrow_deflection_left, arrow_deflection_right, arrow_left_right]

Table 14: Data format description for polygon-shaped markings (ordinary)

Data field	Description
cls	Classification of the marking. Possible classes: [stop_line, shaded_area_marking, zebra_marking]

Table 15: Data format description for polygon-shaped markings (symbol)

Data field	Description
symbol	Possible symbols: [bicycle, wheelchair, traffic_sign_136-10, car, pedestrian, electric_car]

Table 16: Data format description for polygon-shaped markings (text)

Data field	Description
text	Possible texts: [BUS, TAXI, B4, 30, BAB, ZOB, B5, ALTONA B4 B4, LADEZONE, TUNNEL, ALTONA, LADEPLATZ, HBF]

Table 17: Data format description for relations

Data field	Description
points_utm	List of two tuples (x_utm, y_utm, z_utm). First point: coordinate of the main object Second point: coordinate of the secondary object
main_object	ID of the main object (e.g., sign, light, or negation)
sec_object	ID of the secondary object (e.g., lane)
cls	Classification of the relation. Possible classes: [is_traffic_light_for, is_traffic_sign_for, is_negation_for]

Map Meta Data

Table 18: Data format description for HDPC tile definition (JSON)

Data field	Description
name	Name of the point cloud tile being defined
points_utm	List of tuples (x_utm, y_utm, z_utm) defining the bounding polygon of a tile