IEEE dataport

Intelligent Ride

Exported on 12/18/2023

Table of Contents

1 OSHA dataset

Abstract:

Volkswagen Group of America Innovation and Engineering Center California (VW IECC) is a research facility in Belmont, California working on the future of the mobility. In the recent years exciting developments have happened for the autonomous vehicles. In general, lack of data is the main problem to tackle to solve the task of autonomous driving. One of the important tasks in this topic is the overtaking and lane changes, especially in the highway scenarios. Due to the cost and challenges of real-world data collection, specifically in a highway environment, a simulation environment named **SimPilot** was developed by Innovation Center Europe (ICE) for data collection in highway scenarios. To solve the task of overtaking and automatic lane changing, we release a large-scale (more than 8M frames) dataset called **O**vertaking in **S**imulated **H**ighw**A**y (**OSHA**) of Ego vehicles driving in a highway environment with traffic, consisting of lane ID sensor maps and data frame recorded at each frame. The scene is a 3.2 km long loop consisting of 3 lanes with different traffic densities at different episodes. Other movable objects present in the scene are limited to vehicles (no busses, bikes, trucks, or pedestrians) and are controlled by SUMO.

The Ego was controlled by an in-house developed rule-based algorithm to ensure the maximum safety and correctness of every lane change, as well as following the traffic rules and speed limits. The 8M frames were recorded at each step, where steps are 50ms apart from each other (about 50 hrs of Ego driving). Moreover, traffic densities are set by number of vehicles per km which varies from 5 to 35. On top of that, the speed limit is randomly changed at each episode and different sections of the road have different speed limits.

Dataset description:

We are releasing two datasets, both raw and pre-processed dataset that was used in our paper : **link to the paper**. Difference between these two datasets can be seen in the table below:

Dataset	**length**
raw	8,970,692
pre-processed	8,206,442

raw columns:

initials, time, milestone, task, eps, step, pythonTime, ego_speed, position_x, poisition_y, timestamp, heading_x, heading_y, acceleration_x, acceleration_y, orientation, continuous_lane_id, lane_relative_t, angle_to_lane, controller_state, vehicle_switching_lane, traj_pose_x, traj_pose_y, traj_pose_v, control_points, static_lanes, image_name, speed_limit, expert_type, collision_type, left_lane_available, right_lane_available, allowed_speed, movable_obj, speed_action, lane_change_command, travel_assist_lane_change_state

pre-processed columns:

index, initials, time, milestone, task, eps, step, pythonTime, ego_speed, position_x, poisition_y, timestamp, heading_x, heading_y, acceleration_x, acceleration_y, orientation, continuous_lane_id, lane_relative_t, angle_to_lane, controller_state, vehicle_switching_lane, static_lanes, image_name, speed_limit, expert_type, collision_type, left_lane_available, right_lane_available, allowed_speed, movable_obj, speed_action, lane_change_command, travel_assist_lane_change_state

future_x_local_array, future_y_local_array, future_v_global_array, future_points, future_ta_lane_change_array, car_matrix, lane_change_command_modified, sorted_movable_obj, movable_obj_EucDist

2 Ego Observation

Index	Feature name	Type	Unit	Range	Description
0	initials	string			Initials of the person collecting data
1	time	float	s	ℝ+	Game time since application start.
2	milestone	string			Data collection milestone.
3	task	string			Data collection task.
4	eps	int		ℝ+	Episode number.
5	step	int		ℝ+	Step of the episode.
6	pythonTime	float	s	ℝ+	Time since the python connection is established.
7	ego_speed	float	m/s	ℝ	Velocity of ego.
8	position_x	float	m	ℝ	Global x-coordinate of ego position.
9	position_y	float	m	ℝ	Global y-coordinate of ego position.
10	heading.x	float	Unit circle	[-1, +1]	This value is the same as cos(Orientation)
11	heading.y	float	Unit circle	[-1, +1]	This value is the same as sin(Orientation)
12	acceleration.x (Longitudinal)	float	m/s^2	ℝ	Longitudinal acceleration of ego (forward or backward).

13	acceleration.y (Lateral)	float	m/ s²	ℝ	Longitudinal acceleration of ego (forward or backward).
14	orientation	float	rad	[-2π, +2π]	This value corresponds to the yaw values of the trajectory
15	continuousLaneId	int		{3, 4, 5}	Lane ID of ego.
16	laneRelativeT	float	m	ℝ	Distance between ego and the center of the current lane.
17	angleToLane	float	rad	[- π, + π]	Signed-angle between ego heading and lane heading Negative values if the ego-vehicle's heading vector is located on the right side of the heading, and positive if on the left side.
18	ControllerState (Depreicated)	int		{0, 1, 2, 3, 4, 5, 6}	The state of the external controller 0: Off, 1: Passive, 2: Active, 3: Rejection, 4:Error, 5: Init, 6: Deinit
19	vehicleSwitchingLane	int		{0, 1, 2}	Whether ego is currently switching lanes or not. 0 observation off; 1 vehicle currently switching lanes; 2 vehicle has both front wheels inside one lane
20	traj_pose_x (Depreicated)	array(10)	m	ℝ	x-coordinate of 23 future posses (old controller).
21	traj_pose_y (Depreicated)	array(10)	m	ℝ	y-coordinate of 123future posses (old controller).
22	traj_pose_v (Depreicated)	array(10)	m/s	ℝ+	velocity of 23 future poses (old controller).

23	control_points (Depreicated)	array(230)			Control points for the 23 pose controller (old controller). Each point has 10 features which are all available in this observation as a flatten 23*10 array/\.
24	static_lanes	array(60)		ℝ+	a 20*3 flatten array where each triple (lane ID, speed limit, lane type) is available for a maximum of 20 lanes. lane ID and speed limit are the same as other available observations in this list. Lane type is available in the next section).
25	image_name	string			Name of the laneID sensor frame corresponding to this step.
26	speed_limit	float		ℝ+	Speed limit in the current lane.
27	expert_type	string			Sumo (in-house rule-based) or Human expert.
28	collision_type	int (bitflag)			The bit values correspond to the elements of the following enum, starting from the least significant bit: CollisionType { Curb = 1, Sidewalk = 2, Pole = 4, Animal = 8, Human = 16, Bicycle = 32, Motorcycle = 64, Mini-wheeler = 128, PassengerCar = 256, Bus = 512, Truck = 1024, OtherVehicle = 2048, Train = 4096, Building = 8198, NatureObject = 16384, TrafficSign = 32768, Trafficlight = 65536, ObstacleOnRoad = 131072, Other = 262144 }
29	left_lane_available	bool			Left lane availability for ego
30	right_lane_available	bool			Right lane availability for ego
31	allowed_speed	float	m/s	ℝ+	Speed limit of the ego's current lane

32	movable_obj	object list (20 * 12)			Information related to 20 movable objects within the range of 100m from ego.
33	speed_action	float	m/s	ℝ	Speed command sent to the controller at this step.
34	lane_change_command	int		{0, 1, 2}	0: Keep the current lane, 1: Left lane change, 2: Right lane change
35	travel_assist_lane_change_state	int		{0, 1, 2, 3, 4, 5, 6}	Current lane change state of the TravelAssist controller* 0: None, 1: Initiated, 2: ReadyToStartLateralMovement, 3: StartedLateralMovement, 4: Interrupted, 5: Success, 6: Failed

TravelAssistLaneChangeState:

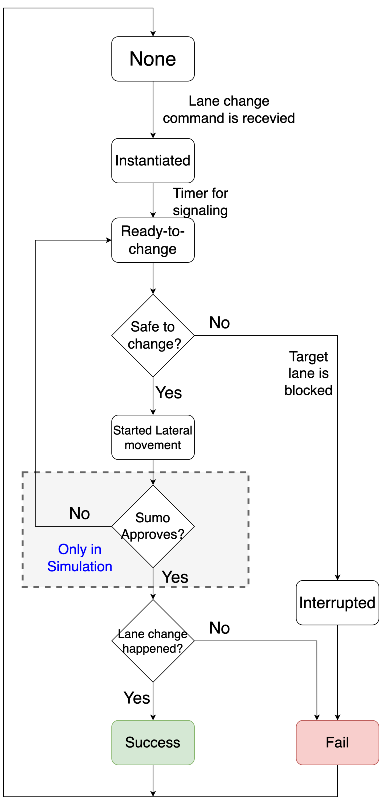
Please note:

- The observation can **skip** internal states depending on timings and the decision period.
- It is **guaranteed** that the final states **Success and Failed** are always returned for **one step**.

Travel assist states and flowchart:

Index	Feature name	Description
0	None	Currently no queued up lane change command.
1	Initiated	The vehicle is signaling (blinking) its intention to perform a lane change. The duration is coupled to the values *SignalDurationMin* and *SignalDurationMax* in the advanced_settings.json
2	ReadyToStartLateralMovement	The vehicle would perform the lane change if it wasn't blocked by other vehicles. The maximal duration for this state, before discarding the initial lane change command, can be set via the *LaneChangeRetryDuration* value in the advanced_setting.json

Index	Feature name	Description
3	StartedLateralMovement	The vehicle is in the process of lane changing.
4	Interrupted	The vehicle was already moving laterally, but stopped the movement (most probably due to being blocked). The lane change may or may not be continued.
5	Success	
6	Failed	Possible reasons: 1. Target lane blocked 2. Timeout (Should never happen)



2.1 LaneType

Summary: Type of a lane.
Type: int

Mapping:

0	None
1	Driving
2	Stop
3	Shoulder
4	Biking
5	Sidewalk
6	Boarder
7	Restricted
8	Parking
9	Bidirectional
10	Median
11	Special1
12	Special2
13	Special3
14	RoadWorks
15	Tram
16	Rail
17	Entry

18	Exit
19	OffRamp
20	OnRamp

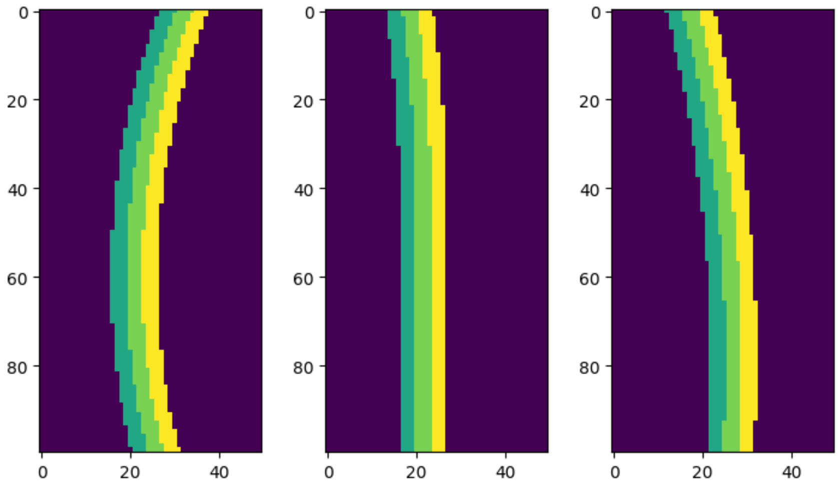
3 Movable Objects

Array of 20 movable objects in sight (20x12). Each object is defined as below. None existing objects are filled with zeros.

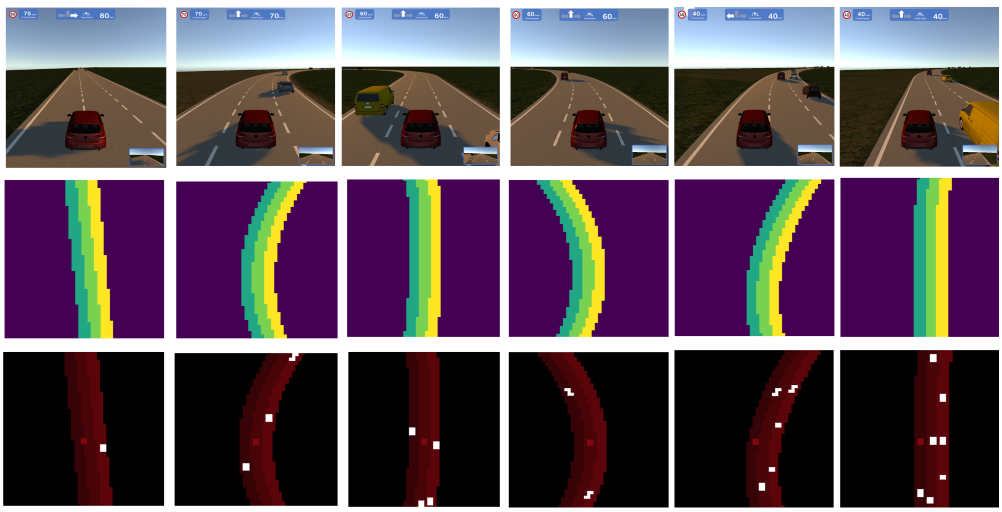
0	ID	int		description
1	Position.x	float	m	Longitudinal distance from the center of the rear axle of ego.
2	Position.y	float	m	Lateral distance from the center of the rear axle of ego.
3	Velocity.x (Longitudinal)	float	m/s	Absolute longitudinal velocity.
4	Velocity.y (Lateral)	float	m/s	Absolute lateral velocity.
5	Orientation	float	rad	
6	ContinuousLaneId	int		Movable object's lane ID.
7	Boundingbox.length	float	m	
8	Boundingbox.width	float	m	
9	ObjectType	int		
10	LaneRelativeT	float	m	Movable object's distance from the center of its current lane.
11	SignalState	int/bitflag		• Bit 1 (least significant bit): Left turn signal is on • Bit 2: Right turn signal is on • Bit 3: Brake light is on • Bit 4: Emergency blue light is on

Example lane id's and the related lane id view

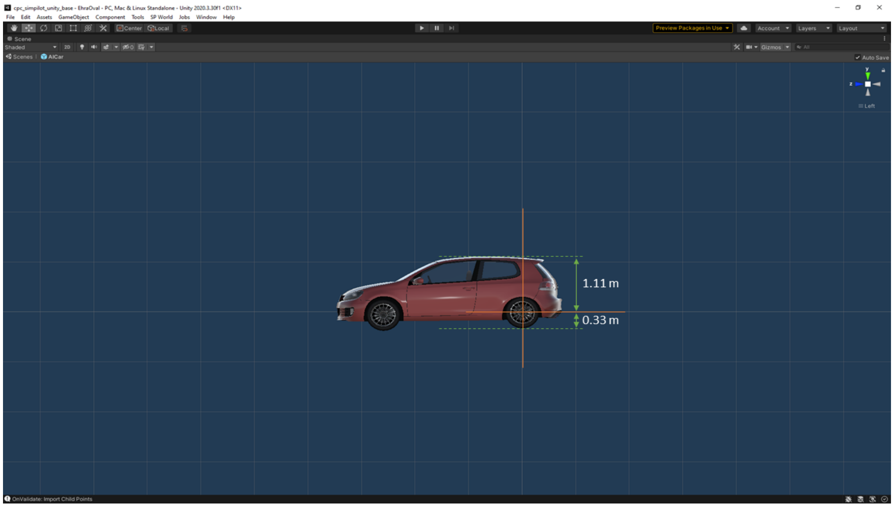
A 100 * 50 image where each pixel's value represents the lane ID it belongs to. Pixels valued 0 are off the road. 3, 4, and 5 stand for the left, center, and right lane on the road. The road always has 3 lanes (similar to a German autobahn).

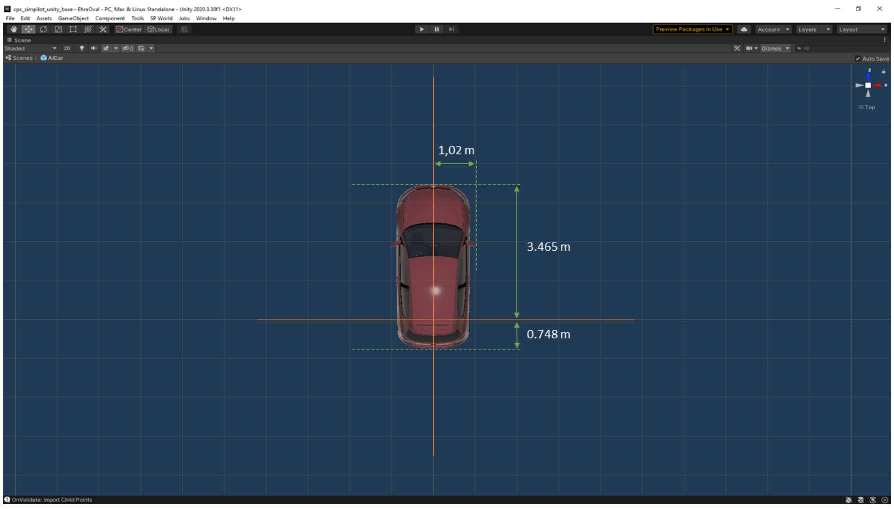


where the ego is center of the image.



4 Ego dimensions



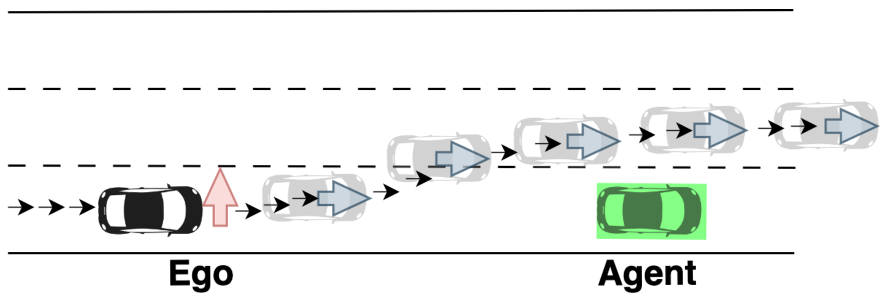


5 Processed features

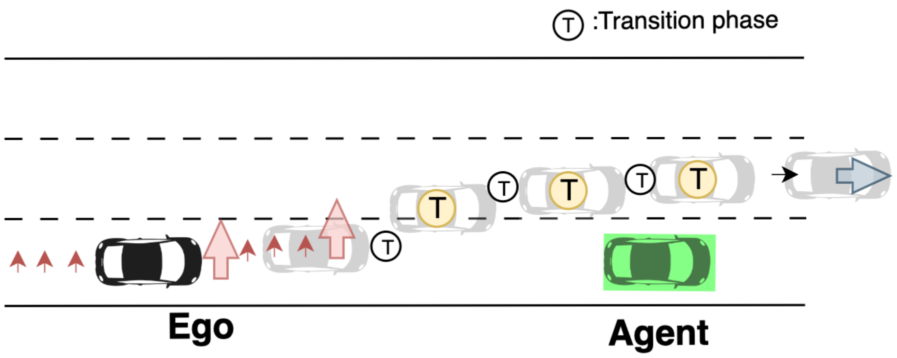
Feature name	Shape	Unit	Range	Description
future_x_local_array	array(5)	m	ℝ	x-coordinate of 5 future points with respect to current time stamp.
future_y_local_array	array(5)	m	ℝ	y-coordinate of 5 future points with respect to current time stamp.
future_v_global_array	array(5)	m/s	ℝ+	Global speed values of 5 future points.
future_points	matrix(5, 2)	m	ℝ	(x, y) coordinates of the 5 future points with respect to current time stamp.
future_ta_lane_change_array	array(5)		{0, 1, 2}	5 future travel assist commands. 0: keep the current lane, 1: left lane change, 2: right lane change
car_matrix	matrix(21, 21)	m	ℝ+	car_matrix[i][j]: Euclidean distance between i-th movable object and j-th movable objects where ego is 0-th object.
lane_change_command_modified	array(5)		{0, 1, 2, 3}	5 future travel assist modified commands. 0: keep the current lane, 1: left lane change, 2: right lane change, 3: transition. Plus, there will be more lane change commands based on the pre-processing.
sorted_movable_obj	matrix(20, 12)		ℝ+	Movable object list sorted based on Euclidean distance with ego in an ascending order.
movable_obj_EucDist	array(20)	m	ℝ	Euclidean distance of movable object list.

The 5 future points are 500ms apart (2.5s in the future) which is 25 simulation steps (each simulation step is 20ms). For the new modified lane change commands, all the steps where controller is either instantiated or read to change lane are marked with the lane change command. Plus, up to 5 steps before the actual lane change command (100ms) are changed to the same lane change command (left or right lane change). The transition states are added to the steps where controller has already started changing lane (i.e. started change).

Raw collected data:



Pre-processed data:



6 Serialized dataset

We have released two processed versions of our dataset, as well as the raw dataset. One serialized and one non-serialized. Both these datasets have been processed from the same raw data. However, in the serialized data each processed step is available as a .pkl file which can be loaded as a pandas data frame, with the corresponding image. On the other hand, the non-serialized version is one single files containing all the steps from different episodes and the images are provided separately to be loaded from disk at the training time. As the single file is quite large (~70GB) and training such file might not be feasible for everyone, we decided to also make this dataset available in the serialized form. Needless to mention, the downside of using the serialized dataset would be the time spent on reading files for every batch of training data, whereas in the single file the data is already loaded on the RAM and only images need to be loaded from disk.