A Survey of State-Action Representations for Autonomous Driving
Edouard Leurent

To cite this version:
Edouard Leurent. A Survey of State-Action Representations for Autonomous Driving. 2018. hal-01908175

HAL Id: hal-01908175
https://hal.archives-ouvertes.fr/hal-01908175
Submitted on 29 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Survey of State-Action Representations for Autonomous Driving

Edouard Leurent\textsuperscript{1,2,3}, Yann Blanco\textsuperscript{3}, Denis Efimov\textsuperscript{2}, and Odalric-Ambrym Maillard\textsuperscript{1}

\textsuperscript{1}Inria Lille Nord Europe – SequeL Team\textsuperscript{2}Inria Lille Nord Europe – Non-A Team\textsuperscript{3}Renault Group

Contents

1 Representing the vehicles 2
  1.1 Encodings .............................................. 2
    1.1.1 Continuous coordinates ......................... 2
    1.1.2 Discrete coordinates ........................... 3
    1.1.3 Spatial grid ...................................... 4
  1.2 Coordinates systems .................................... 6
    1.2.1 Cartesian coordinates .......................... 6
    1.2.2 Polar coordinates .............................. 6
    1.2.3 Lane-centric coordinates ...................... 7
  1.3 Camera images ........................................ 8
  1.4 Other features ......................................... 9
    1.4.1 Intentions ........................................ 9
    1.4.2 Lateral features ............................... 9
    1.4.3 Longitudinal features ......................... 9
  1.5 Full trajectory ....................................... 10

2 Representing the environment 10
  2.1 Cartesian coordinates ............................... 10
  2.2 Cartesian occupancy grid ............................ 10
  2.3 Polar occupancy grid .................................. 11
  2.4 Camera images ....................................... 11
  2.5 Road structure ....................................... 11
  2.6 Traffic laws .......................................... 12

3 The action space 12
  3.1 Continuous actions .................................. 12
  3.2 Discrete actions ..................................... 12
  3.3 Temporal abstraction ................................ 13

4 Conclusion 14
Abstract

In this survey, we list different representations for the state and action spaces of a driving policy. We focus on the literature dedicated to decision making rather than perception, and on the context of autonomous driving only.

To formulate the autonomous driving problem as a Markovian Decision Process, one must first describe the state of the system and its encoding in a space $S$. We list the main state representations found in the literature in sections 1 and 2. Then, one must define a set of actions $A$ that gives the agent control over the state dynamics, which we explore in section 3.

1 Representing the vehicles

1.1 Encodings

1.1.1 Continuous coordinates

A vehicle driving on a road can be described in the most general way by its continuous position, heading and velocity.

$$s = \begin{bmatrix} x & y & \psi & v \end{bmatrix}^T$$

(1)

The composite state (or joint-state) of a road traffic with one ego-vehicle (denoted $X_0$) and $N$ other vehicles can then be described by the set of the states of all vehicles.

$$s = \{s_k\}_{k \in [0,N]}$$

(2)

The reference frame can be absolute, but as the behaviour followed by the ego-vehicle should be the same at any given location and only depend on the relative position of entities around it, it is common to use an ego-centric reference frame. It allows to concentrate the distribution of visited states around the origin in both position, heading and velocity space, as other vehicles are often close to the ego-vehicle and with similar speed and heading. This reduces the region of state-space in which the policy must perform.
The size of this state-space is $\mathbb{R}^{4(N+1)}$.

This representation is used in (Forbes et al., 1995; Wheeler et al., 2015; Bai et al., 2015; Gindele et al., 2015; Song et al., 2016; Sunberg et al., 2017; Paxton et al., 2017; Lee and Seo, 2017; Shalev-Shwartz et al., 2017; Galceran et al., 2017; Chen et al., 2017; Paxton et al., 2017).

In some cases, this representation is used only for memory storage though the surrounding vehicles are not actually part of the optimized state, but rather used to define constraints or penalties regarding the ego-vehicle state. See, for instance, Levine and Koltun (2012); Ziegler et al. (2014); Qian et al. (2016); Sadigh et al. (2016).

1.1.2 Discrete coordinates

To reduce the size of the state-space, any continuous variable $z \in \mathbb{R}$ of the state can be quantized to its closest value within a discrete and often finite set $Z = \{z_i\}$.

This shrinks the state-space from $\mathbb{R}^{(N+1)}$ to a discrete set of size $|I|^{4(N+1)}$, where $I$ is the union quantization of the state-space of a single vehicle.
Most of the time, the quantization is chosen uniform for simplicity, as in (Gómez Plaza et al., 2009; Brechtel et al., 2011; Du et al., 2010; Bandyopadhyay et al., 2013; Loiacono et al., 2010; Osipych et al., 2015; Rehder et al., 2017a).

However, it is often the case with uniform quantization that the state-space either remains too large or becomes too coarse. To address this issue, Tehrani et al. (2015) adapt the size of the grid cells with the relative speed of each other vehicle, while Brechtel et al. (2014) suggest to automatically learn a sufficient and efficient discrete partition of the continuous space for a given task.

1.1.3 Spatial grid

The two encodings described so far are efficient in the sense that they use the smallest quantity of information necessary to represent the scene. However, they lack two important properties:

1. Permutation invariance

We expect a driving policy not to be dependent on the order in which all vehicles in the traffic are listed. Ideally, this property should derive naturally from an architectural design and not rely only on data augmentation to cover the $N!$ possible permutations of any given traffic state. That is, if we denote the traffic state representation as $s = (X_1, ..., X_N)$ and the policy as $\pi(\cdot|s)$, we require that

$$\pi(\cdot|(s_1, \ldots, s_N)) = \pi(\cdot|(s_{\sigma(1)}, \ldots, s_{\sigma(N)})) \quad \forall \sigma \in S_N$$

This desired property can be implemented within the policy architecture, as done in (Chen et al., 2017) or (Qi et al., 2016), but also directly in the state representation.
2. Dependency on the number of vehicles

In these formalizations, the size of the state depends on the number $N$ of vehicles considered. For the sake of function approximation which often expects constant-sized inputs, and in order to avoid having a growing computational complexity when more vehicles are on the road, we may wish to get rid of this dependency.

These limitations are addressed by the occupancy grid representation, that uses a different approach for representing a quantity $z$ localized in a space $X$. Instead of explicitly representing spatial dimensions as variables $x$ within a state $\{s_k = (x_k, z_k)\}_{k \in [0, N]}$ indexed on the vehicles, they are represented implicitly through the layout of several variables $z_i$ organized in a grid-like structure indexed on a quantization of the space $X$.

$$X = \oplus_{i \in I} X_i$$

$$s_i = \begin{cases} z_k & \text{if } \exists k \in [1, N] \text{ s.t. } x_k \in X_i \\ 0 & \text{else} \end{cases}$$

The $z$ variable often corresponds to mere presence information (0-1) but can also include additional channels such as heading and velocity.

The size of this state space is then $|Z|^{|I|}$.

![Figure 3: The occupancy grid representation](image)

This representation is used in (Mukadam et al., 2018; Isele et al., 2017; Fridman et al., 2018).
1.2 Coordinates systems

1.2.1 Cartesian coordinates

In most cases, the vehicles locations are expressed in a Cartesian coordinates system, as seen in 1.1.

1.2.2 Polar coordinates

By using a coordinate system polarized at the ego-vehicle, the scene becomes explicitly described from its point of view. This scheme is consistent with the data format of many sensors in the autonomous driving industry, such as LIDARs and radars.

Figure 4: The polar coordinates representation

**Angular sector indexing**  Following the grid representation introduced in 1.1.3, instead of directly indexing on the different vehicles, one can bin them according to the (discretized) angular sector they belong in. By keeping a constant number of vehicle described in each angular sector (usually only one: the closest), the state representation size becomes independent on the number of vehicles, at the expense of removing some of the vehicles from the state, usually the further ones.
Figure 5: The polar grid representation

Other features can be added as additional channels, such as the velocity, type of vehicle, etc.

This representation is used in (Hadsell et al., 2009; Cardamone et al., 2009; Sharifzadeh et al., 2016; Kuefler et al., 2017; Pfeiffer et al., 2017a; Plessen, 2017; Bhattacharyya et al., 2018)

1.2.3 Lane-centric coordinates

But one can observe that the tactical-level decision making policy should also be the same on a straight or curved road, and that only low-level motion planning needs to take the shape of the road into account. Hence, instead of the Euclidean coordinate system, a lane-centric coordinate system can be used where each vehicle is described in terms of its current lane and Frenet coordinate within this lane.
This representation is used in (Coulom, 2002) for car racing, in (Paxton et al., 2017; Wang and Chan, 2017) for highway driving scenarios, and in (Shalev-Shwartz et al., 2016a) for a roundabout insertion scenario. In (Riedmiller et al., 2007), the track-coordinate system is used to represent not only the vehicle’s position but also heading and yaw rate, which are relative the the lane curve heading and yaw rate at the projected position.

**Lane indexing**  Again, a grid encoding can be chosen so as to index directly the lanes on the road and describe where the vehicles are located in each lane. The state size can be made independent of the number of vehicles in the scene by keeping a constant finite number of vehicles in each lane (e.g. front, middle, rear).

This representation is used in (Abbeel and Ng, 2004; Wei et al., 2010; Levine et al., 2011; Ulbrich and Maurer, 2013; Wang and Chan, 2017; Alché and Fortelle, 2017; Bhattacharyya et al., 2018), and is studied in (Chen et al., 2015) where the authors refer to it as a direct perception approach that leverages affordance indicators of the road situation.

Li et al. (2017) uses a quantization of positions in the lanes as close/nominal/far and of velocities as approaching/stable/moving away. Likewise, Wray et al. (2017) use a semantic quantization tailored for handling intersections: approaching/at/edged/inside/empty.

### 1.3 Camera images

The information of presence and location of other vehicles can also be encoded directly through a raw camera image.

An image can be taken from different viewpoints, such as a top-view camera like in (Liu et al., 2018), or a front-view camera like in (Mnih et al., 2015) (see Enduro), and also represented in different spaces (RGB, HSV, etc.).

In this setting, almost no preprocessing of the raw data is needed to obtain the state representation. However, the price to pay for this lack of abstraction is high-dimensionality.
A transition model is also rarely available for planning as it requires solving the difficult task of video prediction. (Finn and Levine, 2017) is one of the few attempts to learn a predictive model of future video frames and use it for motion planning.

1.4 Other features

1.4.1 Intentions

The future trajectory of a vehicle often depends on the internal intentions of its driver. In order to model these dynamics in the Markov Decision Process framework, the intentions must be made part of the vehicle state, even though they are rarely directly observable.

Bai et al. (2015) and Bandyopadhyay et al. (2013) both add to the state of each agent in the scene an unobserved discrete intention, representing their desired goal location coordinates. In (Song et al., 2016), semantic goal locations are used instead of coordinates to represent agents intentions at intersections, among {straight, left, right, stop}.

Instead of the mere destination, other properties of the agents decision process can also be represented, such as their politeness and aggressiveness in (Sunberg et al., 2017).

Even entire behaviours executed by the agents can be listed and represented in the state. This is what Driggs-Campbell and Bajcsy (2013) call intent modes and implement with the examples of lane keeping, preparing to lane change, and lane changing. Forbes et al. (1995) suggest to use Bayesian inference to predict these modes of behaviours based on the observations (e.g. predict a lane change when observing a blinking turn signal), just like Galceran et al. (2017) who focus on detecting changepoints between several policies.

Finally, the other agents intentions can be modelled with a high degree of expressive power as an objective function, under the hypothesis that the agents are rational and execute near-optimal policies. Sadigh et al. (2016) and Huang et al. (2017) show that this approach can be used in pair with a dynamics model to predict future human actions, which yields a closed loop predictive model.

1.4.2 Lateral features

Ulbrich and Maurer (2013) append to the state additional rule-based features, such as whether performing a lane change now is possible, and whether it is beneficial. However, it is more common to see this information stored in the set of actions available in the state and their action-values, like in (Mukadam et al., 2018) or (Liu et al., 2018) where a heuristic rule-based action masker filters out apparently unsafe actions.

1.4.3 Longitudinal features

Driving at high speed into a distant obstacle is quite similar to driving at low speed into a close obstacle. In both situation, what matters for decision making is the time-to-collision (TTC): the decision of braking or steering out of the way has to be taken before the TTC is lower than the duration of these actions.

The longitudinal time-to-collision is defined for all $i \in [1, N]$ as

$$
\tau_i = \frac{x_i - x_0}{v_i - v_0}
$$

(6)

Another related indicator is the time gap, defined as the time needed for the ego-vehicle to reach a vehicle’s current position.
These features are used in (Driggs-Campbell and Bajcsy, 2015; Ulbrich and Maurer, 2013; Isele et al., 2017; Alché and Fortelle, 2017; Liu et al., 2018; Bhattacharyya et al., 2018).

1.5 Full trajectory

Instead of only considering the current timestep in the state representation, one can decide to work directly at the scale of entire trajectories. Thus, Kretzschmar et al. (2014, 2016); Pfeiffer et al. (2016) use cubic splines to represent the future trajectory of each moving entity. However, this representation is not suited for sequential decision making but rather for direct optimization of the objective over the trajectory representation. The main advantage of this approach is the ability to work in continuous time instead of discrete time, which can be very useful to access infinitesimal variations of the kinematics, such as the instantaneous acceleration or jerk, which are relevant in the assessment of comfort. This is the case of Kuderer et al. (2015) who use this framework to infer comfortable driving styles from human demonstrations.

2 Representing the environment

The policy needs to take into account its static environment: the location of the obstacles, of the drivable space, and information regarding traffic laws.

2.1 Cartesian coordinates

To represent the location of obstacles on the road around the ego-vehicle, a first approach is simply to list each obstacle and store their location and geometry (spherical, parallelepiped, etc.) in a tuple.

This representation is compact in the sense that free space is not stored in memory. However, it has varying size and lacks spatial structure in the data structure.

It is used by Abbas et al. (2014).

2.2 Cartesian occupancy grid

To introduce spatial structure and fixed size in the representation of obstacles, the same trick as in 1.1.3 can be used: convert the tuple of coordinates into a spatial grid.

This representation is used by (Ziebart et al., 2009; Richter et al., 2014; Chen et al., 2016; Tamar et al., 2016; Shankar et al., 2016; Wulfmeier et al., 2016; Sallab et al., 2017; Rehder et al., 2017a; Hoermann et al., 2017; Williams et al., 2017; Mukadam et al., 2018; Rhinehart et al., 2018).

The representation choice can be different for dynamic and static obstacles, as shown by Pfeiffer et al. (2017a) who use a Cartesian grid to encode the static environment and a polar grid for the location of pedestrians.
2.3 Polar occupancy grid

Just like in 1.2.2, the polar coordinate system can also be used to describe the environment. Thus, Pfeiffer et al. (2017b); Koutník et al. (2013); Manuelli and Florence (2015); Plessen (2017) and Trehard et al. (2015) describe the shape of the road as ranges to obstacles in a set of angular sectors.

This representation is very close to the natural data structure produced by LIDAR sensors or depth cameras, that measure the distance along rays in every direction.

2.4 Camera images

As mentioned in 1.3, when using raw camera images we benefit from a very rich information feed. It contains information about neighbour vehicles, but also concerning the shape of the road, the location of the lanes and static obstacles. Hence, it can be used as a simple and rich representation of the environment.

The most commonly-occurring setting is the front-view camera, used in (Pomerleau, 1989; Cardamone et al., 2009; Ross et al., 2011; Koutník et al., 2013; Bojarski et al., 2016; Yu et al., 2016; Xu et al., 2016; Sallab et al., 2016; Eraqi et al., 2017; Koppula, 2017; Sallab et al., 2017; Codevilla et al., 2017; Rehder et al., 2017c; Rezagholizadeh and Haidar, 2018).

But the images can also come from a top-view camera, as seen in (Bagnell et al., 2010; Rehder et al., 2017b,c).

Instead of using the raw images, a preprocessing step can be considered, usually for the sake of semantic segmentation to identify the drivable area of the image, like in (Hadsell et al., 2009; Barnes et al., 2017). Another objective of preprocessing can also be the transfer of skills learnt in simulation to real use cases, as Pan et al. (2017) who use style transfer to convert the simulator images distribution to a realistic images distribution that is fed to the driving policy during its training, so as to improve robustness in real usage. Sometimes, this preprocessing is not used as a transformation of the observation but rather as an auxiliary task. Thus, Eraqi et al. (2017) performs semantic segmentation of camera images and shows that it improves performance on the main task of steering angle regression.

Finally, camera images being high-dimensional observations, compressed representations can be learnt explicitly through the use of unsupervised learning and auto-encoders, such as in (Ha and Schmidhuber, 2018; Kendall et al., 2018).

2.5 Road structure

The knowledge of the road network provides meaningful information. For instance, Van Den Heuvel et al. (2013) describes the track curve through its orientation, width and lateral offset with respect to a finite set of positions along a planned route. Seff and Xiao (2016) extract from camera images features describing the road structure, such as the drivable directions and distance to intersections.

However, it is rarely included directly in the state-space.

Liniger et al. (2014); Song et al. (2016); Wray et al. (2017) use it as static reference information that defines constraints on the state-space, such as forbidden error states. In (Shalev-Shwartz et al., 2017), the road geometry is used to assess responsibility in a potential collision, and to forbid only collision states where the blame is on the ego-vehicle.

More commonly, it is used to define the reward function. For instance, more reward can be assigned to driving in some parts of the road, typically along a known navigation route and in the center of the lane, as in (Levine and Koltun, 2012; Liu et al., 2018). Likewise
Abbeel et al. (2008); Ziebart et al. (2008); Stiller and Ziegler (2012); Gindele et al. (2015) represent the road network as a graph and use it to evaluate the cost of the ego-vehicle’s planned trajectory. Road boundaries can also be described by curve equations, such as in (Williams et al., 2018).

The downside of excluding road information from the state is that at test time the policy cannot rely on this information, so the optimal policy is tailored for a specific road configuration and has to be computed again whenever it changes. As a consequence, the trajectory optimization often has to be performed online in this setting.

2.6 Traffic laws

In addition to the road structure, traffic laws can also be encoded in the state representation. For instance, Paxton et al. (2017) adds in the state-space some features describing the current lane’s speed limit, whether vehicle has entered a stop region or has the right of way. These properties can be extracted from camera images in a perception module, like in (Seff and Xiao, 2016) that learns to differentiate between one-way vs two-way roads and whether the vehicle is driving in the wrong way.

Again, these traffic laws can be used as reward rather than directly in the state. Liu et al. (2018) uses them to penalize entering intersections when the traffic lights are red, or driving in opposite or biking lanes. As the policy has no access to this information at inference time, it has to be optimized in an online manner to stay consistent with the evolution of these data.

3 The action space

We now study the different action spaces used in the driving policy literature.

3.1 Continuous actions

When driving a car, there are only a few actuators to consider: the steering wheel angle, the acceleration and brake pedals, and the gearbox. The different pedals and gears are often merged into a single acceleration command for simplicity.

Hence the canonc continuous action-space is composed of the longitudinal acceleration and steering angle, and often used for low-level control tasks, such as (Sadigh et al., 2016; Cardamone et al., 2009; Ross et al., 2011; Levine and Koltun, 2012; Garcia and Fernandez, 2012; Koutník et al., 2013; Van Den Heuvel et al., 2013).

Higher-level representations can also be used if we ignore some parts of the dynamics, as showed by Chen et al. (2017) who act directly with longitudinal and lateral velocities.

Finally, only one of these two actions can be considered if we assume that the other one is chosen according to a separate policy. For instance, Hester and Stone (2006); Shalev-Shwartz et al. (2016a) only consider the choice of continuous acceleration while assuming that a lane keeping lateral controller is available and independently sets the steering angle.

3.2 Discrete actions

To reduce complexity and accelerate the policy optimization, it is common to prefer a discrete action-space to a continuous one. From there, it is straight-forward to discretize the original continuous action-space by binning ranges of actions together.
The binning is often chosen uniform for simplicity, like in (Isele et al., 2017; Pyeatt and Howe, 1998; Gómez Plaza et al., 2009).

It is often the case that only a small number of possible actions are chosen. For instance, Bandyopadhyay et al. (2013); Bai et al. (2015); Song et al. (2016) use only three possible acceleration values: \{decelerate \(a = -\alpha\), maintain velocity \(a = 0\), accelerate \(a = +\alpha\)\}.

However, a uniform binning often suffers from being either too coarse or too high-dimensional. As the distribution of steering angle is heavily concentrated around the center, Xu et al. (2016) suggest to perform the binning in a log-space or even according to the data distribution, in order to have a fine sampling only around frequent actions.

Again, only one dimension can be considered. Brechtel et al. (2014); Bandyopadhyay et al. (2013); Bai et al. (2015); Song et al. (2016) focus on choosing the longitudinal acceleration while assuming that a lateral control tracks a pre-planned trajectory, like simple lane keeping or following the route at intersections.

As noted by Sallab et al. (2016) who compares the continuous and discrete settings, the discretization implies the introduction of discontinuities in the commands, which can lead to instability and jerky trajectories. One way to address this is to assume that discrete actions are all abstract and imply an underlying smooth continuous control policy. Alternatively we can use commands corresponding to higher order derivatives of the dynamics, so as to benefit from the smoothing properties of the integration in the controlled system. This is the case of (Riedmiller et al., 2007; Gindele et al., 2015; Manuelli and Florence, 2015) and (Huang et al., 2017) who use the steering rate or heading rate instead of the steering angle to generate smoother trajectories.

### 3.3 Temporal abstraction

The actions presented so far are commands affecting the car dynamics, which means they must be updated at a high frequency, in the order of 10Hz. In (Shalev-Shwartz and Shashua, 2016; Shalev-Shwartz et al., 2016b) and (Shalev-Shwartz et al., 2017), the authors argue that due to this dense time resolution of decision making, the estimation of the value function faces a very small signal-to-noise ratio as its variance grows linearly with the time horizon in terms of actions count, which makes training difficult.

To address this issue, some approaches such as (Mnih et al., 2015) or (Brechtel et al., 2011) suggest repeating primitive actions for several steps to reduce increase their duration and reduce the time resolution, hence decreasing the variance during training.

A more principled approach is proposed by Sutton et al. (1999): the options framework. In this setting, a set of sub-policies over primitive actions called options are used as high-level decisions by a policy over options. The options are often provided by the system designer as a way to introduce prior domain knowledge. Though they reduce the expressive power of the policy compared to primitive actions, they still allow to define complex behaviours while being sample-efficient through temporally extended actions.

In the context of autonomous driving, there are some options choices that are widely shared among the community.

Indeed, the lateral behaviour of the ego-vehicle is often handled by three lane change options: \{change to left lane, change to right lane, stay on current lane\}. This is the case in (Abbeel and Ng, 2004; Ulbrich and Maurer, 2013; Osipychev et al., 2015; Sharifzadeh et al., 2016; Li et al., 2017; Mukadam et al., 2018). Additionally, Ulbrich and Maurer (2015) also consider the behaviours of preparing a lane change by edging on the side of the lane, and indicating a lane change with the blinker. Galceran et al. (2017) also introduces
separate options for handling turns at intersections: \{turn right, turn left\}. Specific manoeuvres can also be used in the presence of obstacles, such as a pass behavior learnt in (Pyeatt and Howe, 1998) and an overtake behavior in (Loiacono et al., 2010).

The longitudinal behavior can also be dealt with by using options. They can be used to ensure safety, like the brake option of (Sunberg et al., 2017) which applies a maximum safe acceleration with respect to the front vehicle. Isele et al. (2017) and Wray et al. (2017) define three options for coming to a stop at an intersection, waiting or edging slightly, and proceeding through the intersection. They recognize that these options are less expressive than a set of discrete acceleration, but the resulting policy is easier to learn. It is also possible to use open-loop policies, like Wei et al. (2010) who define a set of acceleration profiles, such as: "keep constant acceleration for \( t_1 \) seconds, then keep constant velocity for another \( t_2 \) seconds".

Finally, options combining both lateral and longitudinal goals can be used, like in (Shalev-Shwartz et al., 2017) where \( 10^4 \) semantic actions are generated by an option graph specifying both the lateral goal on lanes and longitudinal goal of relative positioning with respect to other vehicles and speed profile. In (Paxton et al., 2017) options generated by reinforcement learning as optimal policies with respect to specific manually-defined rewards to favour multiple behaviour such as following the front vehicle, changing lane only when it is safe, passing a vehicle, or stopping at an intersection. In (Codevilla et al., 2017), a low-level control policy acting over acceleration and steering angle is modulated by a high-level conditioning specifying the desired route at an intersection, among \{left, right, straight\}.

4 Conclusion

In this survey, we listed the most common representations for state and actions used in the autonomous driving literature. There is a wide variety of formulations with different properties in terms of size (large or small, fixed or variable), continuousness or discreteness, invariance to permutations, characteristic time-scale, ease to model the dynamics, smoothness of the state-action mapping, etc. There is no best representation, and one must be chosen by considering many aspects of the intended use-case.

References


Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Sun Lee. Intention-Aware Online POMDP Planning for Autonomous Driving in a Crowd. 2015.

Tirthankar Bandyopadhyay, Kok Sung Won, Emilio Frazzoli, David Hsu, Wee Sun Lee, and Daniela Rus. Intention-Aware Motion Planning. 2013.


Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. 2016.


Régmi Coulom. *Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur*. PhD thesis, 2002.


Jan Koutník, Giuseppe Cuccu, Jürgen Schmidhuber, and Faustino Gomez. Evolving Large-Scale Neural Networks for Vision-Based TORCS. 2013.


Denis Osipychev, Duy Tran, Weihua Sheng, Girish Chowdhary, and Ruili Zeng. Proactive MDP-based collision avoidance algorithm for autonomous cars. In 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, 2015. ISBN 9781479987290. doi: 10.1109/CYBER.2015.7288078.


Shai Shalev-Shwartz, Nir Ben-Zrihem, Aviad Cohen, and Amnon Shashua. Long-term Planning by Short-term Prediction. pages 1–9, 2016a. URL http://arxiv.org/abs/1602.01580.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. 2016b. URL http://arxiv.org/abs/1610.03295.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a Formal Model of Safe and Scalable Self-driving Cars. 2017.


Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for Bertha - A local, continuous method. In IEEE Intelligent Vehicles Symposium, Proceedings, 2014. ISBN 9781479936380. doi: 10.1109/IVS.2014.6856581.