Functional Discretization of Space Using Gaussian Processes for Road Intersection

Mathieu Barbier, Christian Laugier, Olivier Simonin, Javier Ibanez-Guzman

Functional Discretization of Space Using Gaussian Processes for Road Intersection Crossing

Mathieu Barbier[1,2], Christian Laugier[2], Olivier Simonin[3] and Javier Ibañez-Guzmán[1]

***Abstract*— A framework for the discretization of navigable space within and around a cross intersection is proposed in this paper. The purpose of our approach is to capture the manner in which drivers manoeuvre in an intersection in order to facilitate and understand the decision-making tasks. Gaussian processes are used to learn and predict the most likely trajectories taken by multiple drivers in different situations. The merging and crossing areas are found by searching for the overlap between two predicted trajectories, whereas the area approaching the intersection is discretized by using the most probable occupancy. The generated areas are stored in the vehicle navigation map. The correlation between the proposed discretization and the driver's behaviour is demonstrated. The proposed framework enables also the discretization of the vehicle velocity profile, information that can be used to govern the decision-making function.**

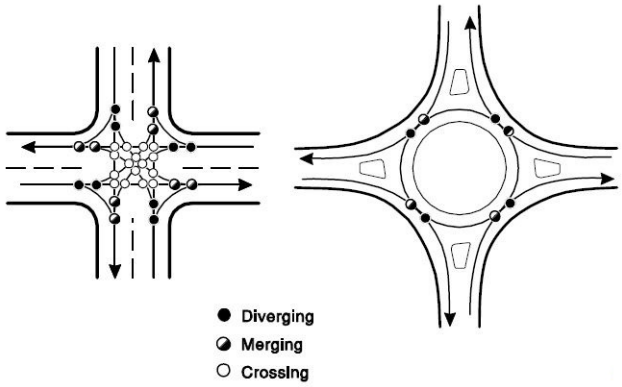


Fig. 1. Vehicle conflict point comparison from [3]

I. Introduction

A. Motivation

Progress in perception, wireless connectivity, computer processing have triggered strong interest within industry on autonomous driving. Different demonstrators exist having different capability levels. Whilst road mortality has seen a reduction due to improvements in vehicle design and legislation, this remains high, in France alone [1] there were 3384 deaths in 2014 at a cost of 37.3 b€. Driver error is identified as the leading cause. Accidentology studies have identified that of them are due to loss of situation awareness leading to the ability to perceive hazards.

Advanced driver assistance systems (ADAS) and autonomous driving systems use data from perception and communication to obtain an instantaneous descriptions of road elements in their surroundings. Indeed the road is composed of many persistent elements such as traffic lanes and traffic signs. Instead of using perception to detect them while driving, systems can use maps constructed with data collected from a company (e.g. HERE) or other drivers (for e.g. collaborative maps). Such maps can store intersection data, position of the white lanes or a pre-computed path for the car to follow. Specific map formats can be created for autonomous driving such as the one put forth in [2]. However such descriptions might not be compatible with geographic information system (GIS) standards and other tools created by the cartographer community.

Accidents at intersections are responsible for 30% of corporeal accidents and 13 % of fatalities[1]. With a layout stored in a map, possible crossing and merging paths where such accidents occur can be found. Crossing points are located at the intersection of paths of two vehicles going in different directions. Merging point is the junction of the path of two vehicles going in the same direction. For example a four way intersection will have 16 collision points and 4 merging compared to roundabout which only has 4 merging points as shown in figure 1. Conflicts in interaction happen around these areas . Collision points are especially dangerous due to its implied side collision. For each of these points, a relation of priority between vehicles can be found. This relation can be set by traffic signs or traffic lights. Otherwise, by default the right priority is applied. Not respecting these rules lead to collisions. Approaching an intersection, drivers will adapt their behavior (switch lanes, stop or pass). This can be observed as a change in their velocity.

B. Contribution and paper outline

The purpose of this paper is to provide a framework for modeling intersection using functional areas for safe intersection crossing manoeuvres.

The complexity and diversity of an intersection motivated our work to propose a functional discretization of space taking into account trajectories taken by drivers. We propose to use a dataset of trajectories to train Gaussian processes. This dataset contains different approaches towards an intersection with various type of cars. Using this representation, relevant

[1]Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. name.surmane@renault.com
[2]Inria Grenoble Rhône-Alpes , Chroma team,655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France name.surmane@inria.fr
[3]INSA Lyon, CITI Lab., Inria Chroma team,655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France name.surmane@inria.fr
This work is supported by a CIFRE fellowship from Renault S.A.S

crossing and merging areas can be ascertained by searching for overlaps in two trajectories. Furthermore, we observe a driver's approach to an intersection as well as his velocity. This can be later used in situation understanding or decisions making .

Resulting areas are represented by polygons in a standard GIS format that can easily be embedded in a map. One contribution of our work is to be able to subdivide the space taking into account how drivers move inside the intersection. We also show the correlation between the functional discretization with simulated data and real data.

The paper is organized as follows. Section II presents related work on space discretization and on the use of Gaussian processes. Principle of Gaussian processes and our framework is explained in Section III. Section IV describes the data set that has been created from the simulator while Section V presents the result and discussions.

II. Related Work

Discretization is the process used for transferring a continuous space model into a discrete counterpart. In this paper the continuous model are possible positions for a vehicle in an intersection, and the discrete counter part are the proposed functional areas.

In perception, such a discretization aims to capture the smallest object or the smallest movement to be perceived. Smaller movements are noise or not relevant for further analysis . For example in [4], the space is discretized in a 2D square grid with 0.1mX0.1m cells. Within this, a probability of occupation is computed. The main advantage is that the occupancy of each cells can be computed separately and enables the possibility of parallel processing. Squared cells can also be found in [5], where large squares are used to represent the position of a vehicle in space. The advantage is that the state of the vehicle is updated each time its crosses a new square, thus avoiding unnecessary computation cause by small movement. Such a representation fits when cars are driving in straight lanes with orthogonal intersections but will be complex to adapt with other layouts. Furthermore, they do not take in account how drivers move in the space and some cells might be too big to capture subtle changes in driver's behaviours.

Driver behaviour is deduce from the context and observing his trajectory [6]. In [7] a comparison between a reference velocity profile and data from communication systems is used within a Bayesian network. Then by comparing what the expected behaviour and observed behaviour, the risk of witnessing a non-compliant drivers can be estimated. However that reference trajectory doesn't capture the variation caused by the driving style. [8] used Gaussian processes to model these different styles. They showed the advantage of using Gaussian processes to represent trajectories of a driver's approach to an intersection. However their method is not scalable to the rest of the intersection and for different manoeuvres.

Gaussian processes (GP) are known for their capability of modeling process involving time and uncertainty. They have been successfully used to model trajectory in a two dimensional space in [9] where trajectories were classified based on their learned hyperparameters and used to find smooth trajectories. In [10], GP have been used to grow an RRT in order to navigate through a dynamic environment.

Gaussian processes have been used to classify driver behaviour at an intersection in [11]. Using trajectories of vehicles extracted from a video, the system was able to identify behaviours of drivers in the field of view of the camera. Due to the sensing setup, only the visible part of the trajectories are tracked and the approach, not recorded.

A representation of merging and crossing points using trajectories learnt in a Gaussian process is proposed in [12]. Their areas are found with simple rules applied to a graph node created with a motion pattern learnt with a Gaussian process.

III. Proposed framework

A. Gaussian processes principles

Gaussian processes (GP) has been applied for classification and regression problems in many domains. More information about Gaussian process and its applications can be found in [13]. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution (definition from [13]). A GP aims to retrieve the functional dependency $f(x_t) = y_t + \epsilon_t$ from a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$. For simplification this data set is represented with two matrices X, a $D \times n$ matrix that contains all training input $\{\mathbf{x}_i\}_{i=1}^n$with D the dimension of x_i and $\mathbf{y}$ a vector with size n that contains observed values$\{y_i\}_{i=1}^n$. The regression problem uses this data set to find the distribution of $p(y_*|x_*, D)$ with x_* an input and y_* corresponding output. The GP is defined with its covariance $k(,)$ (or kernel) and its mean function $\mu()$ written $GP(k(,), \mu())$ with:

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{1}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] \tag{2}$$

A squared exponential covariance function is chosen (3) as in [8], [9] :

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2{}_n \exp(-\frac{1}{2l^2}(\mathbf{x} - \mathbf{x}')^2) \tag{3}$$

σ_n is the signal variance and l the length scale, they form a set of values $\theta = \{\sigma_n, l\}$ called hyperparameters. The learning process of a GP tunes values of θ to maximize the posteriori θ when $p(\mathbf{y}|X, \theta)$ is the higher. We use a minimization algorithm to minimize the log marginal likelihood:

$$\log p(\mathbf{y}|X, \theta) = -\frac{1}{2} y^T K_y^{-1} \mathbf{y} - \frac{1}{2} \log |K_y| - \frac{n}{2} \log(2\pi) \tag{4}$$

where $K_y = k(\mathbf{x}, \mathbf{x}) + \sigma_n I$ and $|.|$ is the matrix determinant.

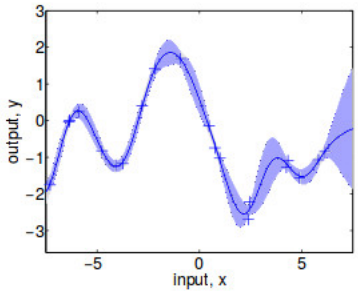


Fig. 2. Gaussian process example. + corresponds to prediction made by the Gaussian process, the light blue area corresponds to 95% area of confidence of the prediction.

After this learning process, the distribution of a testing input $\mathcal{N}(\mu_*, \Sigma_*)$ can be found with

$$\mu^* = k(\mathbf{x}, \mathbf{x}_*)\left(k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I\right)^{-1} Y \tag{5}$$

$$\Sigma^* = k(\mathbf{x}_*\mathbf{x}_*) - k(\mathbf{x}, \mathbf{x}_*)(k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} k(\mathbf{x}, \mathbf{x}_*)^T \tag{6}$$

A typical output of a GP is shown in figure 2. In this example, the learning and optimization manage to find a good approximation of the desired function since the covariance (light blue area) stays small for all the tested outputs

B. Learning phase

Considering variation in the duration of trajectories due to different speed and/or action (slow down, stop,etc) normalized time has been selected as input vector. In the literature, the use of distance to intersection such as in [8] is often chosen. However, this distance is not relevant once the car has entered the intersection or it has turned left or right. Figures 3 shows the difference in the use of the real time vector and the normalized one. Bounds of the input vector are the same regarding the variation in speed.

The normalized time vector is computed as:

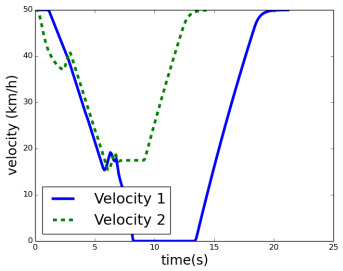
$$T_i = \frac{t_i}{L} \tag{7}$$

with t_i time when the sample has been taken, L duration of the trajectory, T_i the normalized time. Each Gaussian process is trained with one of the following features as output:

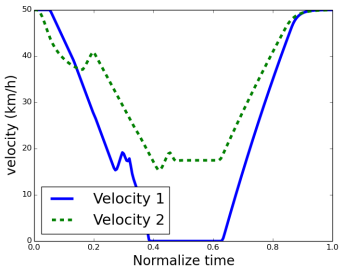
$$Position : x_T, y_T \in \mathbb{R}^2 \tag{8}$$

$$yaw : \theta_T \in \{0, 2\pi\} \tag{9}$$

A set of GP is required for each possible direction (turn left/right, straight) and for each entrance. Thus a cluster method, using the first and the last position of a trajectories, regroups similar manoeuvres in a single cluster.



(a)



(b)

Fig. 3. (a) Shows two velocity profiles plotted against time with a different length in the input vector,(b) shows the same velocity profile with the normalized time vector

Then each GP is trained with a cluster of trajectories and optimized with the minimization method. The set of GP can now be used to represent the trajectory inside the intersection with its variation. Formally:

$$f(T) = GP(T) = [\mu(T), \sigma(T)] \tag{10}$$

$$traj_{pred,i}(T) = \{f_x(T), f_y(T), f_{yaw}(T)\} \tag{11}$$

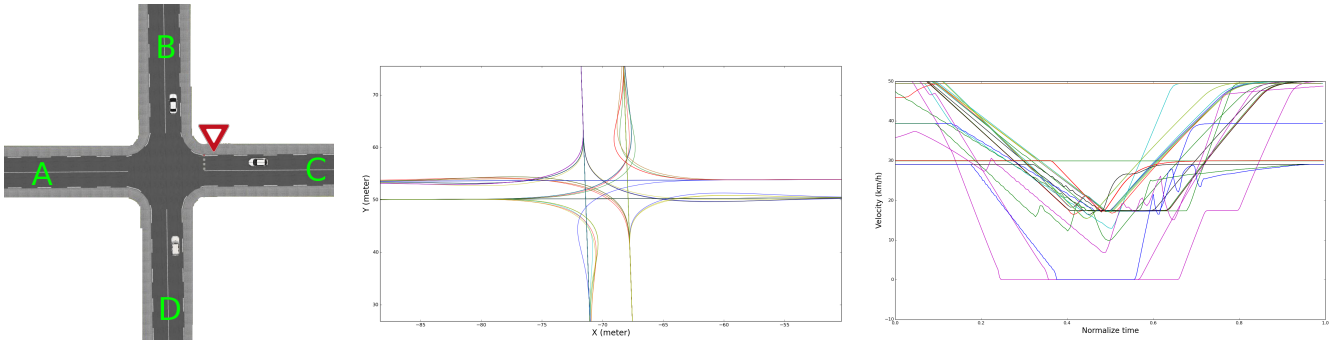
with $T \in \mathbb{R}$ and $0 < T < 1$ and $i \in \mathbb{N}$ and $0 < i < N$, where N is the number of paths possible to enter an exit the intersection.

C. Discretization of crossing and merging areas

Algorithm 1 Compute map with maximum probability of occupation

```
function MAX_MAP(traj_pred())
    map ← zeroes(center_x ± 60, center_y ± 60)
    for T in Normalized time do
        x, y, σ_x, σ_y, θ ← traj_pred(T)
        distribution ← Rotate(N([x, y], [σ_x, σ_y]), θ)
        map ← Maximun(map, distribution)
    end for
    return map
end function
```

In this part, polygons corresponding to merging and crossing areas are found using GP learned previously. For each point of the predicted trajectory, a 2-dimensional Gaussian



(a) (b) (c)

Fig. 4. Sample of the data set,(a) shows the simulated intersection used to generate the data set, (b) samples of path in the data set with variation due to the model of the car, (c) velocity profiles of some cars driving in the intersection with variation due to the situation. In (b) and (c) it can be observed that different behaviour are generated by the simulator.

Algorithm 2 Find crossing and merging areas

for all learnt trajectories **do**
  $Map_{ref} \leftarrow Max\_map(trajectory_{pred,ref}())$
  for all other trajectories **do**
    $Map_{other} \leftarrow Max\_map(trajectory_{pred,other})$
    $Area_{other,ref}$
    $\leftarrow Where(Map_{ref} * Map_{other} > \alpha)$
  end for
end for

distribution is added to a map; if two distributions overlap only the highest probability is kept. Thus, for each prediction, a map where the car has most likely been is computed. This method is described in algorithm 1. Then, two of the maps are multiplied in order to obtain a map representing the probability of having overlapping trajectories. We define a merging or crossing area as following :

$$\forall x, y \subset space\, x, y \in Area\, if\, P(overlap|x, y) > \alpha \quad (12)$$

with $space$ an area of 60 meters around the intersection, $overlap$ the event of two vehicles to have their trajectories overlapping and α, a threshold value. The process to generate every areas in the intersection is given in algorithm 2

In order to take into account the size of a vehicle the resulting area is geometrically grown by 0.75 meter. This value corresponds to half the width of a city car.

D. Discretization of an approaching trajectory

The process flow of [14], shows that drivers have to get through different states while approaching an intersection. This part of the framework aims to capture that change observing the probable occupancy found using previously trained GPs. The same concept of maps from algorithm 1 is used. However, the mean of the probability of occupation over time is computed. This allows to capture where the car has stay the longer over the duration of the manoeuvre. Then, the obtained maps is segmented using threshold values $t_{stop}, t_{slow1}, t_{slow2}$. First the area where the vehicle will most likely stop, is found with t_{stop} and $prediction$ is the number of predictions made:

$$\forall x, y \subset space\, x, y \in Area\, if \quad \frac{\sum_{T=0}^{1} P(overlap|x, y, T)}{prediction} > t_{stop} \quad (13)$$

Such areas show a high mean value, because over the entire prediction time, they have been occupied longer. Areas found are grown to take into account the lateral uncertainty (as with crossing areas) and then removed from the map. Then the second threshold is applied and so on. This process is shown in algorithm 4.

Algorithm 3 Compute map with mean probabilities

function COMPUTEMAP($traj_{pred}()$)
  $map \leftarrow zeroes(center_x \pm 60, center_y \pm 60)$
  for T in Normalized time **do**
    $x, y, \sigma_x, \sigma_y, \theta \leftarrow traj_{pred}(T)$
    $distribution \leftarrow Rotate(\mathcal{N}([x, y], [\sigma_x, \sigma_y]), \theta)$
    $map \leftarrow Map + distribution$
  end for
  map $\leftarrow Map/numberofprediction$
  return map
end function

Algorithm 4 Find approaching areas

for all learnt trajectories **do**
  $Map_{ref} \leftarrow Computemap(trajectory_{pred,ref}())$
  $Area_{stop} \leftarrow Where(Map_{ref}) > t_{stop})$
  $Area_{slow1} \leftarrow Where(Map_{ref}) > t_{slow1})$
  $Area_{slow2} \leftarrow Where(Map_{ref}) > t_{slow1})$
end for

IV. DATA ACQUISITION

In order to train the GP, a large data set of trajectories were created. The use of a simulator allowed us to gather data

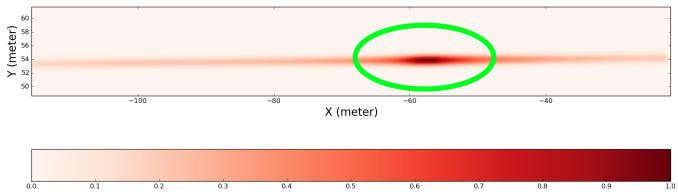


Fig. 5. Map created with prediction from set of GPs, the highlighted area have a high mean probability

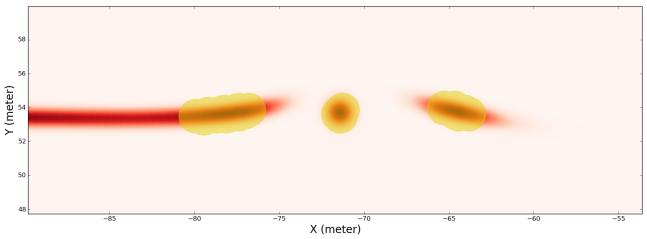


Fig. 6. Segmentation of crossing and merging areas, in red the probability of two cars being in the same position and in yellow where this probability is higher than the threshold

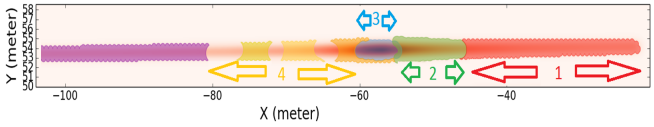


Fig. 7. Discretization of approaching area for cars coming C to A

from a vehicle controlled by the software. SCANER™ is a simulator used in the automotive industry to test most of the aspects of the vehicles (dynamics, driver monitoring, HMI, etc.). Dynamic models used in this simulator take into account the size and weight of the car. Car behaviour also takes into account the situation and the traffic. Figure 4a shows a simulated four-way intersection - a yield sign is present in order to impose a yield manoeuver to the car coming from right. 2.5 hours of simulated driving were recorded with 680 passages in the intersection. Sampling capacity of the simulator is 100hz but has been down sampled to 10hz to match known perception systems. Only a part of the trajectory, within 60 meters of the center of the intersection, is used. This value is chosen to correspond to the distance where a warning sign to a driver should be displayed at an intersection. Figure 4 shows a part of the data set. Three cars were driven autonomously in the scenario to generate random situation at the intersection. Thus, it can be observed that the paths taken by the cars are different (4b), as the velocities (4c).

V. Results and Discussion

A. Discretization results

In this section, the space discretization for a car coming from C to A in figure 4a is shown (an other example is shown in the appendix). Trajectories starting from B are used to generate crossing and merging areas. Figure 5 shows the GP predictions after the learning process. It can be observed that the highlighted area has a high probability of occupancy. It is the consequence of the yield sign that forces drivers to slow down or stop in case of other drivers approaching.

Applying the algorithm 2 creates three areas (in figure 6) that correspond to the most probable merging and crossing areas that a car coming from B will cause. α is set to 0.3.

The size and shape of the shown areas depend on the uncertainty of the predicted trajectories. Engaging the intersection while any of these areas are occupied may lead to a collision. Next, algorithm 4 is applied to the occupancy map. Figure 7 shows the found areas. Here, the space to the right of the intersection is segmented into four areas. These areas correspond to the different states of a driver's behaviour in approaching an intersection. In this example, the threshold values has been set as follows $t_{stop} = 0.8, t_{slow1} = 0.4, t_{slow2} = 0.1$. These values are set for a scenario with a cars that might stop at the entrance of the intersection. This values are sensible to speed limitations and the density of vehicles.

Combining results from algorithms 2 and 4 gives the result shown in figure 8. In this example, the intersection space has been segmented into 10 areas which can then be stored in map layer and loaded before approaching the intersection.

B. Correlation with simulated velocity profile

To highlight the capability to segment the space regarding the change in the velocity, The velocity profile of a car approaching an intersection is shown in figure 9. Vertical lines show when the car entered an area. It shows that approaching areas capture deceleration or changes in velocity. Before the red area (1), the car begins to slow down from the maximum allowed velocity. In area 1, the driver adapts his speed, a local maximum can be observed, caused by an interaction with another car. In area 2, the car continues to slow down

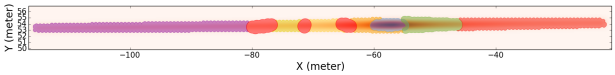


Fig. 8. Final discretization for the proposed scenario

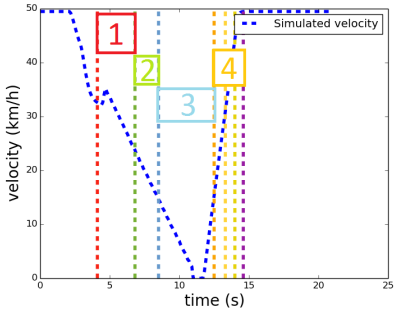


Fig. 9. Simulated velocity profile with vertical line signal areas entrance, numbers and colors match the figure 7

to stop in zone 3. If the situation allows it (the car can safely enter and exit the intersection), the driver accelerates to enter the intersection (area 4), and leaves it. This shows the coherence of the proposed framework to discretize the space of the intersection while taken into account all possible families of trajectories in the intersection and its vicinity.

C. Correlation with real velocity profile

In order to have a preliminary result with real data, an experimental platform (figure 10a) was used to record trajectories executed by human drivers (figure 10b). This intersection showed similar layout as the one used by the simulator with a stop sign. Velocity and position of the car were recorded with an IMU (Xsens). The drivers move in an open environment and had to stop before the intersection. Figure 11a shows the discretization on the velocity profile. It can be observed that results are similar to the one shown in figure 9. It shows that even with a learning phase based on simulated trajectories, the result can be applied on real life intersection.

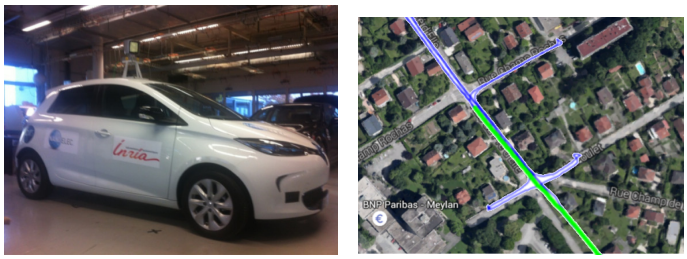


Fig. 10. Experimental setup,(a) show the Zoe experimental platform,(b) in blue the path driven on the road, the green part correspond to the extract used for figure 11

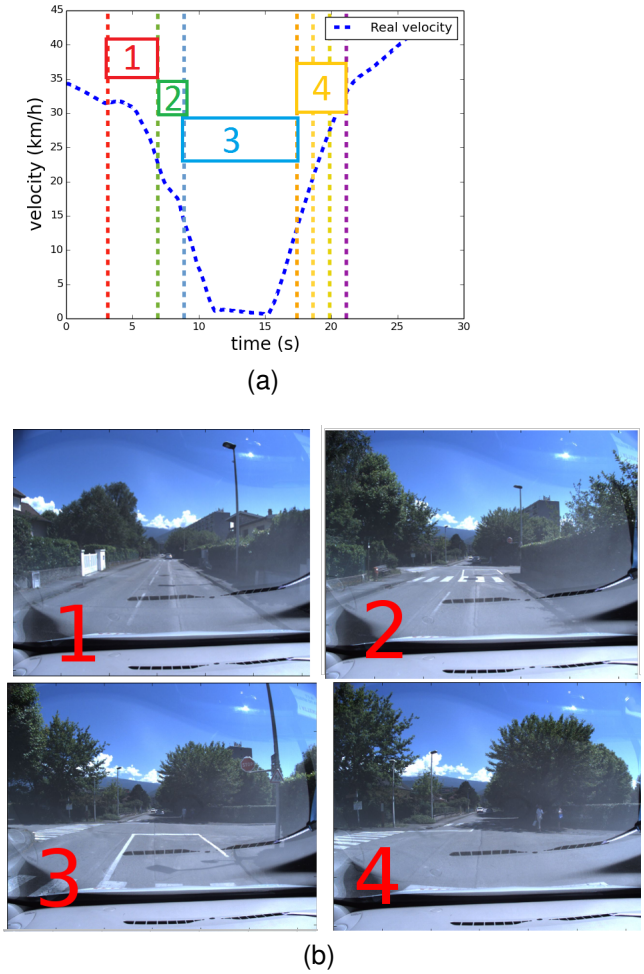


Fig. 11. Discretization of the space applied on real velocity profile,(a)Real velocity profile with vertical line signal an areas entrance, numbers and colors match the figure 7,(b)Images taken at the entrance of each zone, it shows what drivers see entering one of them and elements that can be observed

D. Discussion

1) Limitations: The current computation time required by the prediction step within Gaussian process prohibit real time usage. This will be problematic in situations when factors such as the traffic density or temporary road modifications occur. In these cases, the relevance of the areas might be reduced. However, different set of areas could be generated with other simulation parameters and accessed only for a specific situation. If long-term modifications of the road occur, back end computers should recompute the areas and update the map.

Values used for different thresholds should be changed to adapt each scenarios, even if their values should show small changes between intersections. Future work will investigate adaptive threshold and clustering to improve the reliability of our framework.

Some features are missing in the simulated scenario. The intersection is isolated from the road network. In city environment, more than one intersection can be present and they could influence driver's behaviour. Areas that could represent this conjoined influences could be created. But this cascade of situations might require the simulation of the entire road network.

Also, pedestrians and cycles are not used in our scenario since their simulated behaviours was not be realistic enough to match real life data. Including them would also increase the number of possible situations, thus increasing the size of the data set.

Within the simulation, cars always behave with respect to the highway code, therefore current areas cannot capture common mistakes done by drivers in real life situations. Especially in intersections and roundabouts where experimented drivers have changed their driving style. The problem of having non conforming trajectories will be addressed in the future and how they could or should be included in the learning process.

2) Future works : Application of the proposed discretization will help in action and manoeuvre analysis within decisions-making systems. In order to understand the behaviour of a driver, trajectories need to be analyzed in context. Within one of our area any, new elements should modify

the vehicle context, thus its behaviour. This hypothesis is reinforced by our preliminary results in figure 11a. The next step is to observe the features of trajectories within each area to extract patterns. If a vehicle is not following one of the previously observed patterns a signal can be sent to a decision-making system to react safely. Also the decision-making system could plan trajectories using these areas knowing that an element of the scene will most likely be relevant when being in one area.

As seen in the previous part, the use of a simulator is good enough for simple cases, but it reach its limits to match real life situations. Instead of relying only on the simulator to generate the data set, we plan to include real life trajectories in the data set. The proposed framework allows the usage of such a data set for the learning phase. The required trade-off between real life and simulated data is important for the domain. Collecting data is a time and cost expansive task, where as simulator is cost effective but not always representative. Such a data set could include non conforming trajectories, thus having the car to react to pedestrians, cycles and a larger range of cars.

VI. Conclusion

In this paper, we proposed a functional discretization of the space within and around a cross intersection. The framework uses Gaussian processes to learn most likely trajectories that are used to find merging and crossing areas, as well as subdividing the approaching space. The main advantage of this discretization is to take into account how drivers behave approaching an intersection. By looking at velocity profile divisions, we showed the coherence between our framework and driver's behaviours. We also showed that even with a learning phase using simulated data, the resulting discretization can be applied to real life trajectories. Currently only simulated data are used for the learning phase. We want to explore the usage of real life and simulated trajectories to train Gaussian processes and study required trade-offs between them.

Appendix

The figure 12 present the result obtains for a trajectory from B to C. In this example, we used $t_{stop} = 0.8, t_{slow1} = 0.55, t_{slow2} = 0.1$. Only t_{slow1}had to be changed to take into account speed variation due to the curvature of the road.

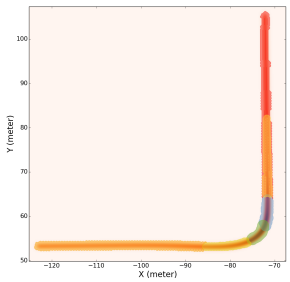


Fig. 12. Segmentation of the approaching area for a cars coming from A to B

Acknowledgment

The authors would like to thank Pavan Vasishta for his help correcting the papers and Sarouthan Sriranjan for his help generating the dataset.

References

[1] O. national interministériel de la sécurité routiere, "Bilan de l'accidentalité de l'année 2014," tech. rep., ONISR, 2015.

[2] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 420–425, June 2014.

[3] L. A. Rodegerdts, B. Nevers, B. Robinson, J. Ringert, P. Koonce, J. Bansen, T. Nguyen, J. McGill, D. Stewart, J. Suggett, *et al.*, "Signalized intersections: informational guide," tech. rep., Federal Highway Administration Research and Technology, 2004.

[4] A. Negre, L. Rummelhard, and C. Laugier, "Hybrid sampling bayesian occupancy filter," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 1307–1312, June 2014.

[5] V. Sezer, T. Bandyopadhyay, D. Rus, E. Frazzoli, and D. Hsu, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 3578–3585, Sept 2015.

[6] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1625–1631, Sept 2010.

[7] S. Lefevre, *Risk estimation at road intersections for connected vehicle safety applications*. Theses, Université de Grenoble, Oct. 2012.

[8] A. Armand, D. Filliat, and J. Ibanez-Guzman, "Modelling stop intersection approaches using gaussian processes," in *ITSC*, (Netherlands), p. xx, Oct. 2013.

[9] C. Tay and C. Laugier, "Modelling smooth paths using gaussian processes," in *Proc. of the Int. Conf. on Field and Service Robotics*, (Chamonix, France), 2007. voir basilic : http://emotion.inrialpes.fr/bibemotion/2007/TL07/.

[10] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[11] G. Aoude, V. Desaraju, L. Stephens, and J. How, "Driver behavior classification at intersections and validation on large naturalistic data set," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 724–736, June 2012.

[12] W. Liu, S.-W. Kim, and M. H. Ang, "Probabilistic road context inference for autonomous vehicles," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1640–1647, May 2015.

[13] C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.

[14] Y. Liu and U. Ozguner, "Human driver model and driver decision making for intersection driving," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 642–647, June 2007.