

Interactive Ramp Merging Planning in Autonomous Driving: Multi-Merging Leading PGM (MML-PGM)

Chiyu Dong*, John M. Dolan†, and Bakhtiar Litkouhi‡

Abstract—Cooperative driving behavior is essential for driving in traffic, especially for ramp merging, lane changing or navigating intersections. Autonomous vehicles should also manage these situations by behaving cooperatively and naturally. The challenge of cooperative driving is estimating other vehicles’ intentions. In this paper, we present a novel method to estimate other human-driven vehicles’ intentions with the aim of achieving a natural and amenable cooperative driving behavior, without using wireless communication. The new approach allows the autonomous vehicle to cooperate with multiple observable merging vehicles on the ramp with a leading vehicle ahead of the autonomous vehicle in the same lane. To avoid calculating trajectories, simplify computation, and take advantage of mature Level-3 components, the new method reacts to merging cars by determining a following target for an off-the-shelf distance keeping module (ACC) which governs speed control of the autonomous vehicle. We train and evaluate the proposed model using real traffic data. Results show that the new approach has a lower collision rate than previous methods and generates more human driver-like behaviors in terms of trajectory similarity and time-to-collision to leading vehicles.

I. INTRODUCTION

Since the 2007 DARPA Urban Challenge, autonomous driving-related technology has developed rapidly. Some of the results are currently being used in commercial vehicles.

Though some advanced driving assistant systems enable vehicles to drive hands-free under certain conditions or below certain speeds, they do not guarantee proper “social driving” interactions with other vehicles. Even if autonomous vehicles become affordable to consumers and successfully commercialized in the future, there will be a long period of time before human-driven vehicles disappear. It is therefore important for autonomous vehicles to exhibit social behaviors to properly interact with human-driven vehicles or other autonomous vehicles.

Especially in a relatively crowded and congested merging scenario, interacting with one merging vehicle may not be sufficient, since multiple merging vehicles could be on a ramp. An autonomous vehicle must therefore coordinate with all relevant vehicles which are detected on the ramp. The autonomous vehicle should try to estimate intentions for each merging vehicle, and then choose a proper strategy resulting in efficient interactions. In a congestion scenario, there could be other constraints that affect behaviors of merging vehicles

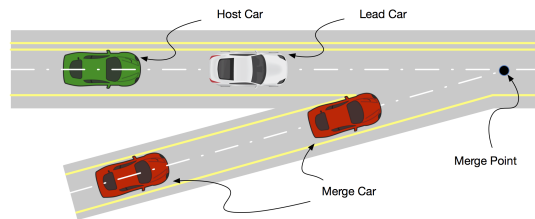


Fig. 1: Merge scenario. The host vehicle (green) is an autonomous vehicle, running on the main road; the leading vehicle (white) is a human-driven vehicle, running on the main road, ahead of the host vehicle; the merge vehicles (red) are human-driven vehicles, running on the ramp.

and limit the reaction of the autonomous vehicle, such as a leading vehicle in front of the autonomous vehicle. The leading vehicle and merging vehicles will mutually affect each other, resulting in different behavior of the autonomous vehicle. Hence the autonomous vehicle should try to identify intentions of the leading vehicle with respect to merging vehicles. Ignoring the leading vehicle’s effect may result in the autonomous vehicle not taking the best possible action.

In this paper, we focus on ramp merge control with a leading vehicle running ahead of an autonomous vehicle, as shown in Figure 1. The goal of our method is to estimate whether or not the merging vehicles intend to yield to the host vehicle, and then react to them. The method takes full advantage of the current Adaptive-Cruise-Control (ACC) model on commercial cars to avoid calculating trajectories, which serves as a intermediate module in our behavioral framework [1]. We start from a model to handle 1-on-1 intention estimation [2] which only considers one host vehicle and one merging vehicle. The 1-on-1 model is used as a building block for the intention estimation of multiple merging vehicles from both the host vehicle and leading vehicle standpoints. A deterministic rule then ensures that the host properly reacts to the merging vehicles and leading vehicle. Experimental results show that our proposed method controls the host vehicles in the face of merging traffic with a lower collision rate than previous approaches and outperforms them in terms of similarity to human-driven vehicles, with smaller differences between the autonomous vehicles’ and human-driven vehicles’ trajectories and similar distribution on time-of-collision to the leading vehicles.

II. RELATED WORK

There are several references that address the merging problem. Urmson et al. [3], Hidas [4] and Marinescu et al. [5] all use the same idea of a slot-based approach for cooperative

This work was supported by GM funding

*Chiyu Dong is with Department of Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, PA 15213 USA

†John M. Dolan is with the The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA

‡Bakhtiar Litkouhi is with GM R&D Center, Warren, MI 48090 USA

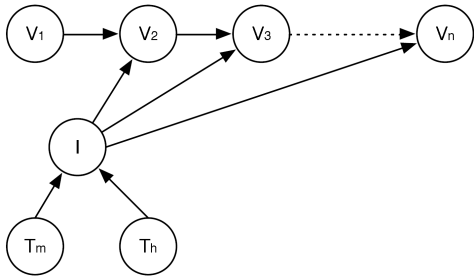


Fig. 2: 1-on-1 PGM, V_1, \dots, V_n are n speed nodes; I is intention node; T_m is the time-to-arrival from the merging car to the merging point; T_h is the time-to-arrival from the host vehicle to the merging point.

merging control. They first check merging availability for each slot in the target-lane (a slot is the free area between two cars). Then they check feasibility of actions to find the best feasible slot for acceptable merging acceleration. Their decision is based on current states and no historical data are considered, which can lead to failures in some cases.

J. Wei et al. [6] proposed an intention-integrated framework to enable an autonomous car to perform cooperative social behavior. Accelerations of cars merging from a ramp are considered, which is hard to obtain by onboard sensors. The estimation again only considers the merging vehicle’s current state, ignoring its historical state. The lack of historical data leads to instability in estimated intention, which results in oscillation or delayed reaction to the autonomous vehicle. To react to surrounding vehicles and reduce computational time, Wei et al. [7] proposed a QMDP single-lane behavior framework which takes uncertainties into account. They also applied a cost function to evaluate and select the proper strategy. The Markov Decision Process (MDP) implicitly estimates intention based only on current state, again without considering historical data.

Schlechtriemen et al. [8] calculate lane-changing probabilities by vehicles’ lateral speeds using Random-Decision-Forest and Gaussian Mixture Regression. Whereas we predict merging intention by a vehicle’s longitudinal speeds over time using a graphical model. Lenz et al. [9] generate cooperative planning for autonomous highway driving using Monte-Carlo Tree Search (MCTS). Their method can handle multiple vehicle interactions in a merging scenario in the simulator. However, all vehicles in the simulator depend on the designed cost function and model. The method has not been tested extensively on real world data.

Nilsson et al. [10], [11] formulate cooperative planning as an optimization problem under a Model Predictive Control (MPC) framework. The weighted effects from acceleration and braking are optimized subject to the trajectory’s shape and feasibility. The author transforms the problem into a QP optimization. However, it requires prior knowledge of other vehicles’ trajectories and the manual tuning of weights is difficult.

The work described above has focused on current state and neglected historical data. One possible reason is that involving more data dramatically increases the dimension

of the parameters, which makes the computation intractable. Alexander et al. and Galcera et al. [12] extended the reaction ability of autonomous cars from a single lane to multiple lanes, including lane changes and intersection navigation. They formulated the problem as a Multi-policy Decision Making (MDPM), and used a finite set of *a priori* known policies and sampling to make the computation tractable. Their method requires forward simulation and a hand-tuned reward function. As mentioned in the literature, the forward simulation is time-consuming and requires a simple motion model.

Prior methods either ignore past data or need complex forward simulations to obtain a discretized policy. However, past data are helpful in recognizing human intentions and reduce the effect of sensing failures. Therefore, in our model, past data play an important role in intention estimation. Our method does not require forward simulation or a manually designed reward function and discount factor. The transition model is directly trained from real driving data, and the only free parameter we need to determine is the number of speed nodes. The proposed method is computationally more efficient than the multi-policy decision making framework and more robust than other methods which only consider current states. In addition, compared with previous methods (MPDM in [12] and GeoACC, which is used as a baseline method in [5] and iPCB [5]), experimental results show that the proposed method is safer and more efficient.

III. METHOD

A. 1-on-1 PGM and its evaluation

Human drivers estimate others’ intentions by observing their kinematic information and environment, e.g., speed changing, locations and road geometry. The 1-on-1 PGM[2], emulates this observation-estimation process to obtain human-like social behavior. Hence the essential part of the method is to establish the relationship among observable variables (e.g., speeds and locations) of merging vehicles and their intentions. As shown in Fig. 2, the PGM takes speeds and time-to-arrival. Speed nodes V_1, V_2, \dots, V_n keep track of speed changes of a merging vehicle for a short period, at relatively high frequency. Here we assume that the current speed is only affected by previous speed and the intention (which will decide either accelerating, decelerating or keeping the previous speed), and the intention does not change as fast as the update cycle rate. That is, the intention stays unchanged during each estimation. Time-to-arrival nodes T_m, T_h are for the merging vehicle and the host vehicle, respectively. The topology describes this dependency: times-to-arrival affect intentions, therefore speed changes. However, the intention is a latent variable which cannot be directly detected. The estimation of the intention depends on observations of historical speeds, times-to-arrival and the structure of the PGM. Thus, the estimation of the intention is to evaluate the conditional probability of the intention given observed speeds and times-to-arrival, in Equation 1.

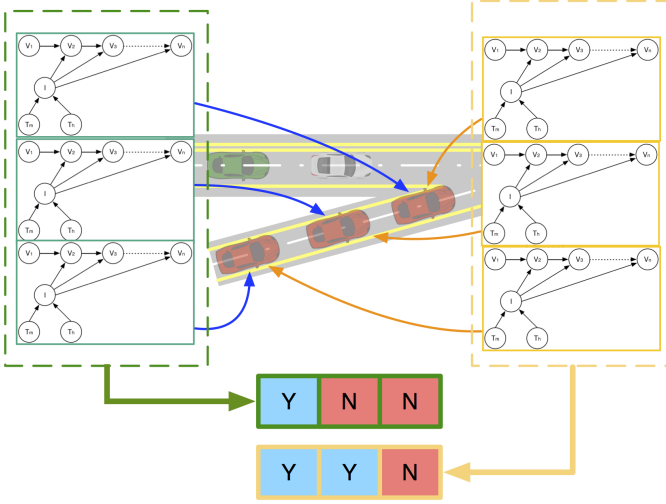


Fig. 3: Multi-Merging PGM with a leading car.

$$\begin{aligned}
 \log P(I|\mathbf{V}, T_m, T_h) &= \log P(\mathbf{V}, T_m, T_h|I)P(I) \\
 &= \log P(\mathbf{V}|I)P(T_m, T_h|I)P(I) \\
 &= \underbrace{\alpha \sum_{i=2}^n \log P(V_i|V_{i-1}, I)}_{\text{Speed Term}} + \\
 &\quad \underbrace{\log P(T_m, T_h|I)}_{\text{Time Term}} + \\
 &\quad \underbrace{\log P(I)}_{\text{Prior Term}}
 \end{aligned} \tag{1}$$

In Equation 1, I is intention, V is a vector of recent speeds: $\mathbf{V} = [V_0, V_1, \dots, V_n]$, and T_m, T_h are respectively the times-to-arrival of the merging and host vehicles. Both speeds and distances to the merging points affect the intentions. Equation 1 factorizes the complex conditional distribution of the intention into parts: Speed Term, Time Term and Prior Term. The factorization clearly separates the effects from speeds and distances. Real data are applied to learn each term. The final estimated intention is the one that maximizes the combination of three terms. The estimated intention is:

$$I^* = \arg \max_I \log P(I|\mathbf{V}, T_m, T_h) \tag{2}$$

B. Multi-Merging Leading PGM

The 1-on-1 model handles one merging vehicle w.r.t. a host vehicle. In reality (also in our dataset), there is often more than one merging vehicle on the ramp, so the 1-on-1 model is insufficient. Moreover, in the main lane, there is often a leading vehicle running closely in front of the host vehicle. Obviously, while reacting to the merging vehicle, we need to be cognizant of the leading vehicle by keeping a safe distance. The following sections build on the 1-on-1 model to create a Multi-Merging Leading PGM to handle multiple merging vehicle and leading vehicle scenarios. In addition, a rule is introduced to react based on estimated intentions w.r.t. the host and the leading vehicle.

The Multi-Merging PGM method has two significant modifications compared with the single PGM model:

- 1 The single PGM model is duplicated and applied to each merging vehicle to generate an instant intention array.
- 2 The process is also applied to the leading vehicle, to obtain the leading vehicle's estimate of the merging vehicles' intentions.

Fig. 3 shows a merging example with three merging vehicles and one leading vehicle. The three PGMs on the left estimate the intentions of merging vehicles w.r.t. the host vehicle; the three PGMs on the right estimate the intentions w.r.t. the leading vehicle. At the bottom of the figure, there are two merging vehicle intention arrays: one for the autonomous vehicle (with green-outlined boxes) and one for the leading vehicle (with yellow-outlined boxes). Each element of an array contains the estimated intention of one of the merging vehicles.

Ideally, each of the arrays contains at most one pivot. The pivot corresponds to the vehicle that divides the array of merging vehicles into two groups: a not yielding and a yielding group. Merging vehicles running ahead of the pivot (including the pivot itself) will not yield to the vehicle on the main road; those running behind the pivot will yield. According to the definition, if no vehicle yields, the pivot vehicle will be the last one in the merging group; if all vehicles yield, the pivot is "NULL". Since the pivot is the last vehicle that does not yield, the autonomous / leading vehicle should follow the pivot by using an aggressive distance keeping model, without considering other merging vehicles. (Because the merging vehicles which run behind the pivot are identified to yield to the vehicle on the main road, the merging vehicle behind the pivot will naturally keep a reasonable distance between itself and the main-road vehicle.) After identifying pivots for both the leading vehicle and the autonomous vehicle, a deterministic rule-based planner is proposed to activate different behaviors for the autonomous vehicle. The rule-based planner is described in the next section.

C. Pivot rules

Note that the autonomous vehicle is always behind the leading vehicle, and the merging vehicles behind the pivot must yield to their corresponding main-road vehicle. We use P_h, P_l to denote the pivot merging car for the autonomous car and the leading car, respectively. Then $P_h = P_l$ if the autonomous car and the leading car have the same pivot; $P_h < P_l$ if the pivot for the autonomous car runs behind the leading car's; $P_h > P_l$ if the pivot for the autonomous car runs ahead of the leading car's. Deterministic cases and corresponding rules are shown in Fig. 4. In each subfigure, the upper array (green boxes) is the merging intentions w.r.t. the host vehicle; and the lower array (yellow boxes) is the merging intentions w.r.t. the leading vehicle. In each array, each cell corresponds to a merging car. The number of merging cars and cells may change. A blue cell means *Yield*, and a red cell means *Not Yield*. The leftmost red cell corresponds to the pivot merging car.

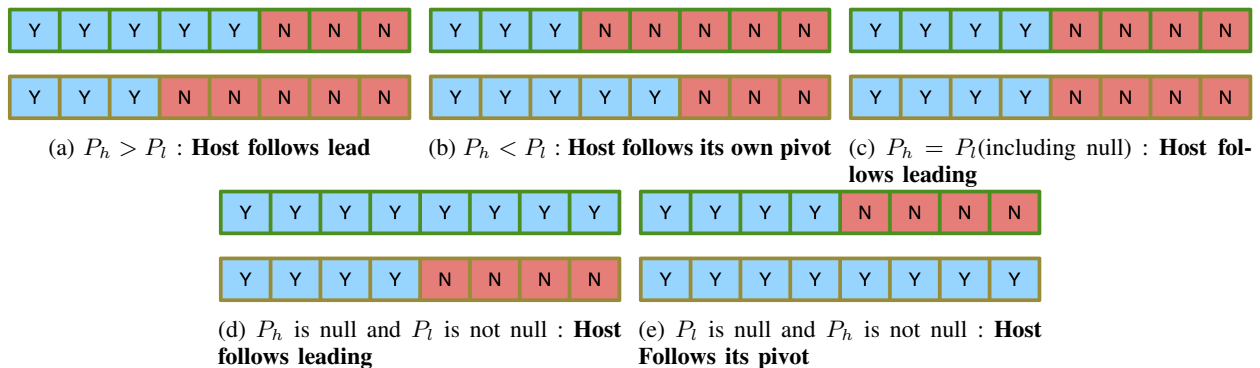


Fig. 4: Illustration of the pivot rules, with examples of pivot cases and corresponding rules. Numbers of cars and positions of pivot dynamically change according to real situations.

D. Distance Keeping Model

The essential parts of the proposed method are the evaluation of the conditional probability and the pivot rule. The ramp merging control yields to a distance keeping problem. Unlike the standard car-following model, it is not necessary to follow the car which runs ahead of the host. Instead, the host vehicle may follow and keep the longitudinal distance with one car on the merging ramp. Sections III-B and III-C introduced the estimation approach to decided which car to follow. The method can easily be plugged into a behavioral planning framework such as that proposed by Wei [1], by using off-the-shelf distance keeping model, as described in [13], [14].

IV. EXPERIMENTAL RESULTS

The performance of the proposed method was tested in simulation. The simulation runs on a standard laptop equipped with Intel CORE i7-level processors. The intention estimation methods are single-thread, non-parallel. In the tests, 70% of the real merging ramp data from the US-101 and I-80 highways in the NGSIM dataset [15] are used to exemplify merging vehicle behavior and train the model; the remaining 30% are replayed to generate simulated scenarios for testing. The datasets are taken from merging ramp regions which are about 600 meters long. Each vehicle traveling in the rightmost lane of the main road is considered a potential host vehicle, as long as it does not change lanes. From the time a host enters the monitored region until it leaves it, the leading vehicle and all vehicles on the ramp are grouped with the host vehicle. There are 354 merging-host-leading groups in the US-101 dataset, and 452 such groups in the I-80 dataset, thus 806 groups in total. Each group contains a host vehicle, a leading vehicle and multiple merging vehicles. Trajectories of these groups are used to train and test the model.

Details of the traffic condition and geometry information are shown in Table I. The low SMS [16] (space mean speed value) indicates congestion conditions. It can be seen in Table I that there is congestion and I-80 and US-101 have similar traffic conditions.

TABLE I: Features of the US-101 and I-80 datasets

Dataset	L_{merge} m	SMS m/s (mph)	Num. of Groupsp.
US-101	90.4	12.4 (27.7)	354
I-80	110.8	14.2 (31.3)	452

The proposed algorithm is compared with three previous merging methods: GeoACC, iPCB [6] and MPDM [12]. GeoACC is ACC with an expanded field-of-view to be able to track both merging vehicles and the leading vehicle. It sets the closest vehicle ahead of the host (whether in the same lane or on the merging ramp) as the following target. Real merging-vehicle data are replayed and the host vehicle reacts to the real merging vehicles' intentions and behaviors. The start point of the host vehicle is set to be the same as that in the real data, but it is subsequently controlled by a combination of either GeoACC, iPCB, MPDM or MML-PGM for target selection with ACC for distance-keeping. All vehicles except the host vehicle are replayed from the real data. We use the following criteria to examine different aspects of the performance of the algorithms:

- A Collision rate and time-to-collision
- B Similarity to human drivers
- C Computational efficiency

A. Collision rate and time-to-collision

We use collision rate and time-to-collision (TTC) to the vehicle which runs in front of the host as safety criteria. Collision rates for the four algorithms are shown in Table II. In the dataset, human-driven vehicles have no collisions. GeoACC, iPCB and MPDM have high collision rates relative to MML-PGM. GeoACC only follows the closest merging vehicle without considering its intention, whereas iPCB considers the closest vehicle's intention. MPDM estimates the most likely behavior based on a simple forward simulation model and a manually designed reward function. None of them is easy to design to reflect real maneuver actions in forward simulation and a rewards mechanism for selecting a proper trajectory. Note that even the proposed MML-PGM has a non-zero collision rate, which means the algorithm is still imperfect. The model is not yet sophisticated enough,

TABLE II: Statistical results for different Merging Control Algorithms and human-driven vehicles.

Algorithms	GeoACC	iPCB	MPDM	MML-PGM	Human
Collision Rate	20.0%	18.9%	15.9%	7.2%	0.0%
K-L divergence	0.24	0.11	0.61	0.02	—
Rate/car (ms)	0.05	0.20	100.55	0.08	—

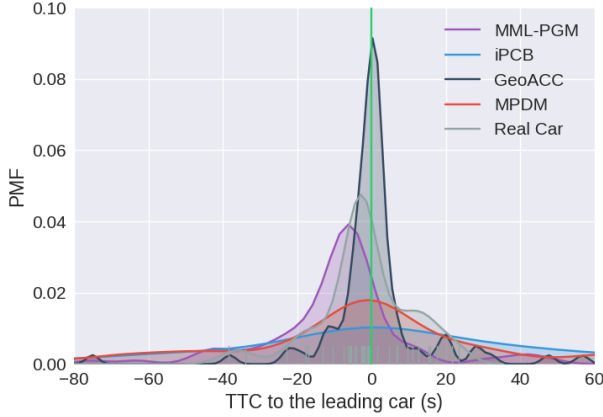


Fig. 5: TTC to the leading vehicle when the host vehicle reaches the merging point.

since it: 1) does not predict trajectories of merging vehicles, the host vehicle and the leading vehicle; 2) only has a binary output to react to the merging vehicles and the leading vehicle; 3) cannot handle uncertainty, e.g., noise.

The second safety criterion considered is the Time-To-Collision (TTC) between the host vehicle (the ego vehicle) and the leading vehicle (which runs ahead of the host vehicle) when the host vehicle reaches the merge point:

$$TTC = (car_{lead} - car_{host}) / (v_{host} - v_{lead}) \quad (3)$$

Note that the leading vehicle always runs ahead of the host vehicle, so the numerator is always non-negative. If the speed of the host vehicle is lower than that of its leading vehicle (either the original leading vehicle or a merged vehicle), the denominator is negative, so the TTC is negative and no collision can occur, since the vehicle ahead is moving faster than the vehicle behind. There will be a collision if the TTC is positive, which means that the host vehicle is moving faster than the leading vehicle. We examine the distribution of the host vehicles' TTC for the four algorithms with the real data. The results are shown in Fig. 5.

Fig. 5 shows the Probability Mass Functions (PMF on the y-axis) over the TTCs that are generated from the four different algorithms and human-driven vehicles. The peak and majority (about 80%) of the MML-PGM TTCs are negative, whereas the other algorithms' TTCs have a peak at zero on the x-axis (indicating collision) and about 50% of the testing cases have positive TTC. The TTC results correspond to what is shown in Table II: in most of the test cases, MML-PGM has negative TTC, which ensures that the gap between the host and its leading vehicle keeps growing, and results in a lower collision rate than that of the other algorithms.

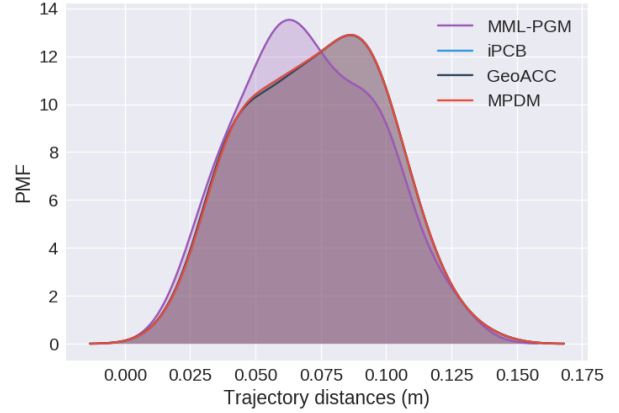


Fig. 6: Distributions of differences between human-driven trajectories and four algorithms.

Note that the majority of the human-driven vehicles (grey curve) also have negative TTC. But the peak is closer to zero, which means that human drivers are more aggressive than MML-PGM but still retain a zero collision rate.

B. Similarity to human drivers

We also examined the similarity of the trajectories generated by each of the four algorithms to those of human drivers using two criteria:

- The Kullback-Leibler divergence (as used in [17]) between the host-vehicle TTC distribution of each of the four algorithms and that of the real data. The results are given in the second row of Table II.
- The distribution of the differences between human-driven vehicles' trajectories and the trajectories generated by the four algorithms, which is shown in Fig. 6;

The second row of Table II shows the K-L divergence of the TTC distributions (the curves in Fig. 5) between human-driven vehicles and each of the four algorithms. K-L divergence indicates the differences between two distributions, and it provides an overall description of similarity between the generated behaviors and the human-driven vehicles in terms of the time-to-collision to the leading vehicle at the moment when the host vehicle reaches the merging point. In the table, GeoACC, iPCB and MDPM have high K-L divergences relative to MML-PGM. The result indicates that MML-PGM behaves more similarly to human-driven vehicles at the merging point.

While the second row of Table II compares the host-lead vehicle TTC only at the time the host vehicle reaches the merge point, Fig. 6 evaluates the similarity of the full trajectories generated by the four algorithms to those of the human-driven vehicles. The figure shows the distribution of the average L2-norm (D) between two trajectories:

$$D = \frac{1}{T} \int_0^T (f_{alg}(t) - h(t))^2 dt \quad (4)$$

where $h(t)$ is the human-driven trajectory, $f_{alg}(t)$ is the trajectory resulting from applying one of the four algorithms,

and T is the time for each merging episode, which is used to normalize the result due to differing trajectory lengths. Fig. 6 shows that the distribution of D using MML-PGM tends to be closer to zero than using the other algorithms. The result indicates that MML-PGM generates behaviors more similar to those of human drivers than does GeoACC, iPCB or MDPM.

C. Computational Efficiency

The third row of Table II shows the time required for estimating intentions for each merging vehicle. MPDM needs the longest time for estimating one merging vehicle, and the forward simulation takes the majority of the computation time. Approximately 100ms is acceptable for one merging car, which results in a 10Hz update rate. However, in a congestion scenario, a 100-meter merging ramp can have about 10-20 vehicles. Thus the update rate will drop to 0.5-1Hz for estimating all merging vehicles, which is not sufficient for driving behavioral planning in this scenario. In the original paper, the author stated that their C implementation runs at 1Hz updating rate. Our proposed method takes only 0.08ms for estimating one vehicle. Even though MML-PGM is not the most computationally efficient algorithm, its update rate is high enough and close to GeoACC. In a congestion scenario, MML-PGM is still able to run at a 625Hz (1.6ms) update rate to estimate intentions of all merging cars. However, since human decisions do not change at such a high rate, we limit the cycle rate to 10Hz in tests.

V. CONCLUSION

In this paper, we introduced the Multi-Merging Leading PGM (MML-PGM) algorithm, which enables an autonomous vehicle to generate cooperative behaviors with multiple human-driven vehicles in highway merging scenarios. The method senses speed and distance changes of other vehicles in order to estimate their intentions. A probabilistic graphical model is applied to organize the relationships between sensed data and intentions. Furthermore, using the PGM, higher-dimensional conditional probability is separated into several simpler parts, which makes the evaluation more feasible and efficient.

The method was tested in simulation with real highway ramp merging data. The method integrates effects of surrounding vehicles, including the leading vehicle running ahead of the host vehicle on the main road and merging vehicles running on the ramp. The results exhibit a significantly lower collision rate and more negative TTCs to leading vehicles compared with previous methods, which indicates the advantages of MML-PGM over these methods. The lower divergence between human-driven vehicles and MML-PGM TTC distributions, and the lower average distance to human-driven trajectories indicate that MML-PGM generates more human-like behaviors than either GeoACC, iPCB or MPDM.

In the future, we will refine the model to handle uncertainties, e.g., noisy observations of vehicle location and speed, and extend the model to estimate longer-term intentions. Based on the long-term estimation of possible trajectories

of merging vehicles, the interaction or negotiation between merging vehicles and host vehicle will be improved. We will also refine the model to fit more types of cooperative situations, such as navigating through intersections and making active lane changes.

ACKNOWLEDGMENT

The authors would like to thank Junqing Wei and Wenda Xu, who contributed to previous ideas of ramp merging and the simulation environment. Thanks also to Tianyu Gu for fruitful discussions about the topic.

REFERENCES

- [1] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 458–464.
- [2] C. Dong, J. M. Dolan, and B. Litkouhi, "Intention estimation for ramp merging control in autonomous driving," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2017 (Accepted, to appear).
- [3] C. Urmsion, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [4] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 37 – 62, 2005.
- [5] D. Marinescu, J. urn, M. Bourouche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 900–906.
- [6] J. Wei, J. M. Dolan, and B. Litkouhi, "Autonomous vehicle social behavior for highway entrance ramp management," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 201–207.
- [7] J. Wei, J. M. Dolan, J. M. Snider, and B. Litkouhi, "A point-based mdp for robust single-lane autonomous driving behavior under uncertainties," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2586–2592.
- [8] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K. D. Kuhnert, "When will it change the lane? a probabilistic regression approach for rarely occurring events," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1373–1379.
- [9] D. Lenz, T. Kessler, and A. Knoll, "Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 447–453.
- [10] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1253–1258.
- [11] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh, "Longitudinal and Lateral Control for Automated Yielding Maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1404–1414, may 2016.
- [12] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multi-policy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Autonomous Robots*, 2017, in Press.
- [13] Z. Yihuan, L. Qin, W. Jun, and V. Sicco, "Learning car-following behaviors using timed automata," in *Proceedings of the 20th World Congress of the International Federation of Automatic Control, 2017*. (Accepted, to appear).
- [14] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-09-08, February 2009.
- [15] "NGSIM homepage. FHWA." 2005-2006. [Online]. Available: <http://ngsim.fhwa.dot.gov>.
- [16] N. J. Garber and L. A. Hoel, *Traffic and highway engineering*. Cengage Learning, 2014.
- [17] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," *arXiv preprint arXiv:1701.06699*, 2017.