Human Driver Behavior Prediction based on UrbanFlow*

Zhiqian Qiao[1], Jing Zhao[2], Jin Zhu[3], Zachariah Tyree[4], Priyantha Mudalige[4], Jeff Schneider[3] and John M. Dolan[3]

***Abstract*— How autonomous vehicles and human drivers share public transportation systems is an important problem, as fully automatic transportation environments are still a long way off. Understanding human drivers' behavior can be beneficial for autonomous vehicle decision making and planning, especially when the autonomous vehicle is surrounded by human drivers who have various driving behaviors and patterns of interaction with other vehicles. In this paper, we propose an LSTM-based trajectory prediction method for human drivers which can help the autonomous vehicle make better decisions, especially in urban intersection scenarios. Meanwhile, in order to collect human drivers' driving behavior data in the urban scenario, we describe a system called UrbanFlow which includes the whole procedure from raw bird's-eye view data collection via drone to the final processed trajectories. The system is mainly intended for urban scenarios but can be extended to be used for any traffic scenarios.**

I. INTRODUCTION

A major challenge in recent work on autonomous vehicles is making proper decisions about how to deal with interactions with human-driven vehicles [1]. However, interactions among human drivers are hard to model via equations directly [2]. To address this problem, learning-based methods [3] for characterizing human-driver behavior become good choices and make it easier to simulate a human driver's behavior in simulations such as CARLA [4], VTD [5], etc. However, such methods require a large amount of driving data in order to learn human drivers' diverse behavior. For a long time, NGSIM [6] was the only public trajectory-based dataset from which human driver behavior could be extracted. In 2018, highD [7] became available, but it only includes highway scenarios. Moreover, how to extract and classify the human driver behavior without manually labeling a large amount of data for ground-truth is another time-consuming challenge when dealing with raw human driver data.

The current state of the art in acquiring and using such data faces several problems. First, some published work relies on privately collected datasets, the inaccessibility of which makes them impossible to use as benchmarks for comparisons between various algorithms. Second, some datasets are collected by autonomous vehicles from the perspective of the ego vehicle. Although this perspective is ultimately the one available to an autonomous vehicle, it is difficult for it to provide full sequences showing the social behavior of surrounding vehicles. To derive models for such behavior, bird's-eye view datasets are useful. In response to these problems, this paper constructs a method for benchmarking human driver behavior based on a bird's-eye-view data collection system via drone. Figure 1 shows the pipeline of the data processing procedure.

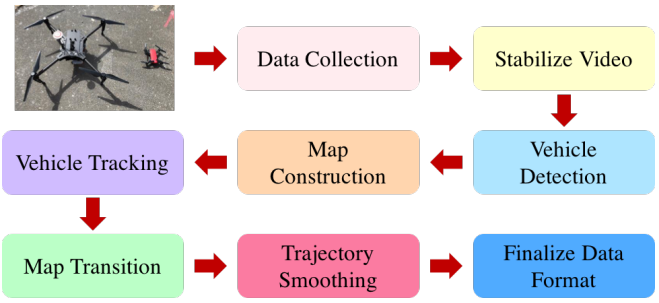

Fig. 1: The UrbanFlow dataset processing pipeline. The pipeline includes the drone data collection and process flow from raw video data to the final trajectory data.

Based on such datasets, predicting other vehicles' intentions or trajectories is an essential component of autonomous vehicle behavior planning during decision making or trajectory planning. Most traffic lights control *Going Straight* (*GS*), *Turning Left* (*TL*) and *Turning Right* (*TR*) with one light with the result that at urban intersections, many interactions occur between vehicles approaching from opposite directions with intention pair of *GS* and *TL* or *TL* and *TR*. In these situations, 'who will go first' between the two interacting vehicles is a key problem even for human drivers.

The main contributions of this work are:

- A drone-based data collection and processing system to analyze bird's-eye view trajectory data of human drivers.
- An algorithm which can predict the interacting human drivers' intentions as well as trajectories based on the historical trajectories occupying a given period of time when approaching an urban intersection.

II. RELATED WORK

This section introduces previous work related to this paper, which can be categorized as follows: 1) papers related to traffic data collection; 2) papers that propose intention and trajectory prediction of human drivers.

*This work is supported by General Motors

[1]Zhiqian Qiao is a Ph.D. student of Electrical and Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, USA zhiqianq@andrew.cmu.edu

[2] Mechanical Engineering, Carnegie Mellon University

[3] The Robotics Institute, Carnegie Mellon University

[4] Research & Development, General Motors

A. Data Extraction

With the current popularity of autonomous driving, various datasets are available for researchers to develop and test their algorithms. These datasets can be categorized into two classes. The first one is traffic-flow-based datasets, which focus on a particular scene and simultaneously capture all the vehicles within it. This type of dataset uses a bird's-eye view to observe vehicle trajectories within the scene. The NGSIM dataset [6] is the best-known such dataset and includes highway and urban scenarios. Last year, RWTH Aachen University released the highD dataset [7], which used advanced computer vision technology to improve the data collection mechanism based on the NGSIM dataset. Another kind of dataset is based on the sensors mounted on the ego-vehicle and data collected while driving the ego car over a given route. Most such datasets create various vision-based benchmarks for further study. The KITTI dataset [8] offers a vision benchmark for different autonomous vehicle-related tasks. The Oxford RobotCar dataset [9] collected 20 million images from 6 cameras mounted on the vehicle, along with LIDAR, GPS, and INS ground truth. Recently, UC Berkeley released the BDD100k [10], which includes diverse driving videos collected from a camera mounted on the vehicle with scalable annotation tooling.

In the current work, in order to gain a comprehensive view of the traffic situation, we use a bird's-eye-view method to collect traffic-flow-based datasets via drone. The portable end-to-end system allows researchers to collect their own data from any site of interest, unlike the NGSIM system, which depended on the installation of a fixed camera. While our data collection method is similar to that used for the highD dataset, our method focuses primarily on urban intersections, which are more challenging than the highway scenarios that the highD dataset focuses on.

B. Prediction

Liebner [11] proposed an explicit model to extract characteristic desired velocity profiles from real-world data that allow the Intelligent Drive Model (IDM) to account for turn-related deceleration to represent both car-following and turning behavior. Derek et al. [12] used LSTMs to classify vehicle maneuvers at intersections. They predicted whether a driver will turn left, turn right, or continue straight up to 150m with consistent accuracy before reaching the intersection using LSTM, with the mean cross-validated prediction accuracy averaging over 85% for both three- and four-way intersections. There are other works on predicting complete trajectories using Hidden Markov Models [13], Gaussian Processes [14], Dynamic Bayesian Networks [15], Support Vector Machines [16], and inverse reinforcement learning [17].

Compared with [11], besides velocity profile, multiple factors are added in our models, such as yaw variation, target motion features, etc., which contain information on environmental changes for the ego vehicle. The work concentrates on the interaction of ego and target car pairs. We also introduce the idea of direction intention prediction and use the result to determine a more detailed trajectory prediction. The main challenge is that the human driver's intentions and vehicle trajectories are highly variable when approaching an urban intersection with heavy traffic flow compared to highway situations. The main contribution of the paper is a prediction algorithm which combines direction and yield intentions and accordingly derives trajectory predictions.

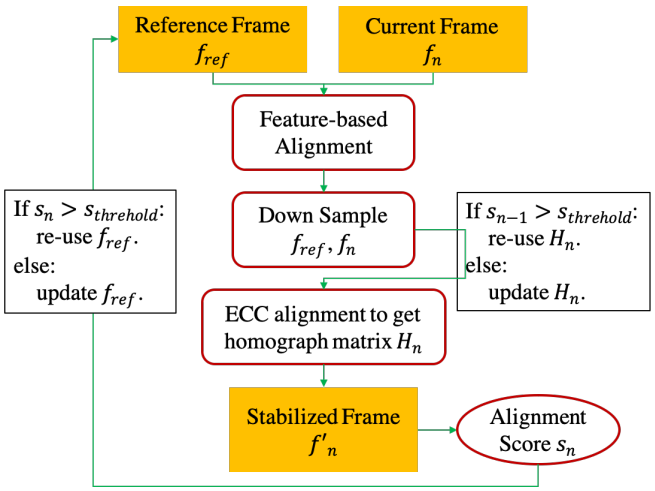

Fig. 2: Optimized stabilization method flow.

III. Methodology

In this section we propose UrbanFlow as a procedure to deal with the collected bird's-eye view data. Then, based on UrbanFlow, we propose a method for predicting the human driver's intention as well as trajectories.

A. UrbanFlow

1) Video Stabilization: The two main challenges for video stabilization are the robustness and the speed of the alignment [18][19]. In this paper, we propose several steps for the video stabilization in order to deal with the displacement of the drone during the data collection process. Figure 2 visualizes the flow of the stabilization method. For each frame f_n at time step n, the algorithm chooses a reference frame f_{ref} according to the alignment evaluation score gotten from the result of the last time step and corresponding homography matrix in order to get the stabilized frame. Firstly, a re-alignment is performed when the result of the ECC alignment score is lower than a threshold. ECC takes a lot of time to converge and is not adaptive to align the current frame with a reference frame when their similarity is lower than a threshold. Secondly, since the alignment is time-consuming, it is only performed when a reference frame needs to be re-chosen. The homography matrix is re-used for the following frames until a new reference frame is chosen when the evaluation score drops to the threshold. Then, the homography matrix calculated from ECC alignment during the previous step is used for initializing the guess for ECC in the next step to speed up the convergence. Lastly, images are down-sampled [20] so that ECC uses fewer pixels during the calculation.

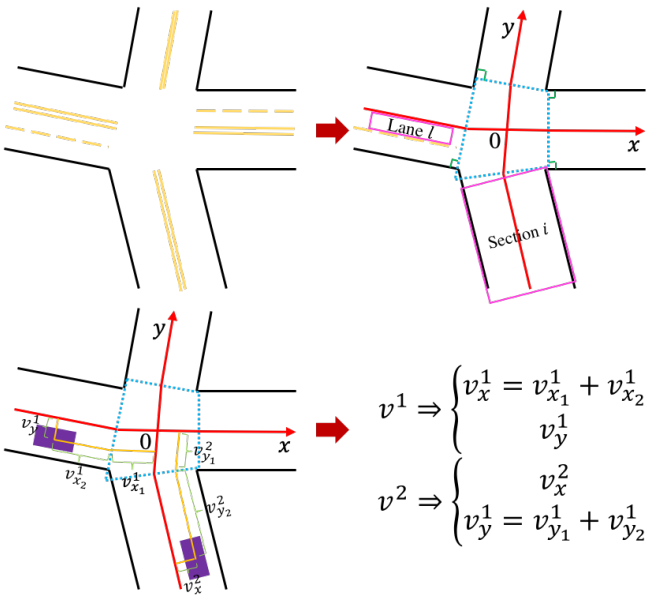

Fig. 3: Transition from original image-based coordinate to road-based Coordinate

2) Object Detection: In the proposed pipeline, RetinaNet [21] is used for detecting vehicles in the images. The training dataset contains all the bounding boxes and their corresponding labels for each image. The input images are re-sized to ensure that the size of detection objects is greater than 32-by-32 pixels as well as not too large for the GPU computational capability. Images are masked to crop out the roads in order to make detection easier. RetinaNet was fine-tuned using pre-trained weights from the COCO dataset [22].

3) Map Construction and Coordinate Transition: The first step in the creation of the map is to crop the area of interest, which in this case is the roads. To attack this problem, we took advantage of the image segmentation network "U-net", described in Ronneberger et al. [23], with just a few adjustments based on the work of Iglovikov et al. [24]. We preserve the decoder section of the network because by adding a large number of feature channels, it allows the network to propagate context information to the higher-resolution layers. The important change was in the encoder section, where it was replaced by the down-sampling elements of the VGG16 architecture in order to take advantage of the pre-trained weights in ImageNet [25], due to the limitation of the quantity of the collected data.

After detecting the road and applying a color filter to detect the lane markings on the road, the work transforms all the detected vehicle positions from the original image-based coordinates to the road-based coordinates. Figure 3 shows the method to generate the road-based coordinate based on a random road geometry which may occur in the real world. The method firstly chooses an origin and then proceeds to obtain the x axis and y axis along the lane markings which separate the opposite directions of moving vehicles. For the given vehicles v^1 and v^2, the figure shows two examples of how to extract the road-based positions. Finally, it is able to represent the vehicles' information, which contains the following items:

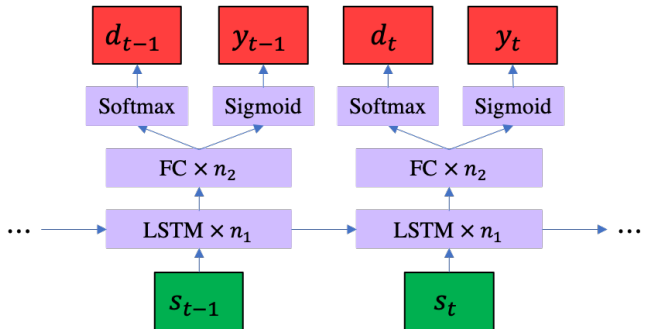

Fig. 4: Intention Prediction Network Structure

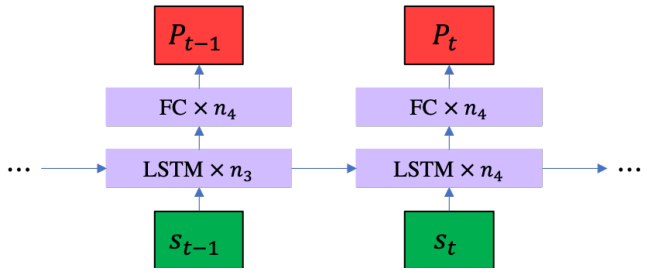

Fig. 5: Trajectories Prediction Network Structure

- Local x and y based on the road-based coordinates
- Vehicle length and width
- Section ID i
- Lane ID l

4) Vehicle Tracking and Trajectory Smoothing: A Kalman filter [26] is used for tracking and trajectory smoothing. Based on the car's dynamic model, characteristics of the system noise and measurement noise, the measurement variables are used as the input signal, and the estimation variables that we need to know are the output of the filter. After the positions of vehicles have been transformed into the local (road) coordinates, we apply the tracking algorithm to track each car. Meanwhile, we smooth each vehicle's trajectory. In the system, we use vehicle position as the state variable. F is the state transition matrix and H is the measurement matrix. $V_q(n)$ and $V_p(n)$ represent the system noise and measurement noise, respectively.

B. Driving Behavior Prediction

1) Intention Prediction: For the driving behavior task, we construct the network with an LSTM layer which gets the Direction Intention d and Yield Intention y (see Figure 4). The direction intentions include *Going Straight* (*GS*), *Turing Left* (*TL*) and *Turning Right* (*TR*). The yield intention indicates the prediction of which car will go through the potential crash point first. For the interacting driver pairs with intentions of *GS* and *TL* or *TL* and *TR*, the input states include the positions, velocities, heading angles and relative distance to the intersection center of both cars of each pair. During the interaction procedure, the yield motion also changes based on the counterpart's behavior. This will contribute as a key factor to the next-step motion planning module and help to generate a safer and feasible trajectory.

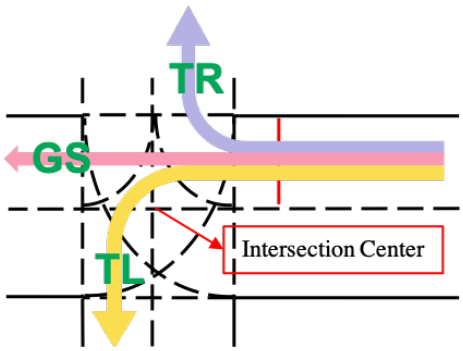

Fig. 6: Reference trajectories according to the direction intention. Follow the center of the lanes.

TABLE I: Comparison between different stabilization methods. DS means down sampled.

Method	Processing Time ($s/frame$)	SSIM
ORB + ECC w/o DS	1.7609	0.8032
ORB + ECC, $\frac{1}{2}$ DS	0.6724	0.7759
ORB + ECC, $\frac{1}{4}$ DS	0.4779	0.7324
ORB + ECC, $\frac{1}{8}$ DS	0.4071	0.7160
SURF + ECC w/o DS	0.6599	0.81896
SURF + ECC, $\frac{1}{2}$ DS	0.6627	0.7758
SURF + ECC, $\frac{1}{4}$ DS	0.4960	0.7336
SURF + ECC, $\frac{1}{8}$ DS	0.3750	0.7166
ECC, $\frac{1}{2}$ DS	0.9345	0.9404
ORB	3.4665	0.8278
SIFT	13.575	0.8370
SURF	2.1060	0.8390

2) Trajectory Prediction: Based on the results of direction and yield predictions, a more detailed trajectory prediction procedure includes more information on the future trajectories. In Figure 5, P_t includes information on velocities and positions of the target car. A reference trajectory is first selected according to the intention prediction results. According to the reference trajectories (see Figure 6) with intersection geometry information, the velocities, heading angles and relative distance to the intersection center of both cars, the network can predict the future trajectories.

IV. Experiments

In this section, we show the results for methods corresponding to different data processing procedures.

A. UrbanFlow

1) Video Stabilization: In the previous section, we have introduced the combination of the feature-matching-based and homography-based alignment methods. Here we compare different combinations of feature-matching-based and homography-based video stabilization algorithms with various down-sampling ratios. Table I shows the results of different choices of algorithms and the corresponding structural similarity (SSIM) score which is used to calculate the similarity between any two images. The higher SSIM score indicates a better stabilization result.

We finally chose the Speeded Up Robust Features (SURF) detector combined with ECC and $\frac{1}{8}$ down-sampling to get a relatively good tradeoff between stabilization and computational efficiency. We visualize images with and without stabilization in Figure 7 with four sub-figures. The Reference Frame shows the anchor frame for the stabilization. Ideally, the roads can be perfectly aligned in the Reference Frame and Target Frame. Before stabilization, the Reference Frame and the Target Frame are blended, which is shown as the Target Blended Frame. It is obvious that the two frames have a big misalignment. After the stabilization of the frame, the result is shown as the Stabilized Frame and then the new blended result is shown as the Stabilized Blended Frame. The final trajectory dataset is provided for later use [1].

Fig. 7: Two blended frames before stabilization and after stabilization

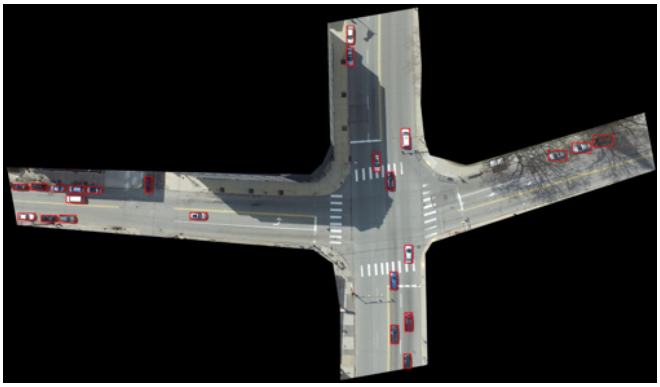

Fig. 8: Vehicle detection results of Retinanet algorithm for a selected frame.

B. Prediction

1) Vehicle Detection: By using Retinanet to do vehicle detection, we trained a good model to detect vehicles from a bird's-eye view. For testing, only 97 out of 2322 vehicles are not detected, giving an accuracy of 96%, and the average intersection over union is 92%. This accuracy is high since the vehicles in the test cases are similar to the ones during training. False positives were removed using non-maximum suppression and thresholding the confidence score for a prediction. If vehicles such as a bus appear in testing but had never appeared in training, these vehicles will not be

[1] https://drive.google.com/drive/folders/1SwPnVHcQLJ_VwFUZ1R3DQsv0ftJxRaXk?usp=sharing

TABLE II: Detection Result

Prediction / True	Car (Train / Test)	No Car (Train / Test)
Car	387 / 2369	1 / 0
No Car	3 / 72	0 / 0
$\frac{GT \cap PR}{GT \cup PR}$	0.95 / 0.994	
$\frac{GT \cap PR}{GT}$	0.97 / 0.995	
$\frac{GT \cap PR}{PR}$	0.97 / 0.996	

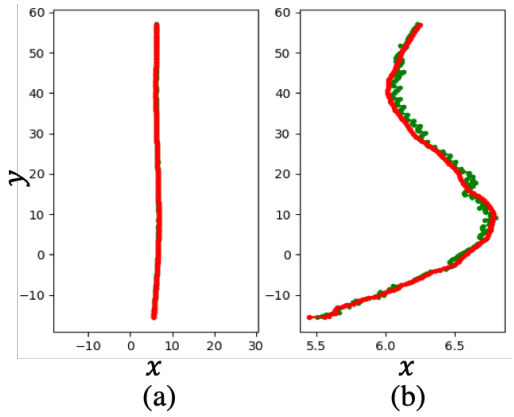

Fig. 9: Comparison between vehicle trajectories with and without smoothing.

detected, since they are too different from what the network has learned both in size and color. The images with incorrect detection were relabeled to fine-tune the network.

For each frame, RetinaNet is applied to detect vehicles. Figure 8 visualizes one of the testing images after applying the vehicle detection method. The original image-based positions of all the red bounding boxes detected as vehicles are saved. Table II shows the quantitative results of training and testing. *GT* is the abbreviation for the area of Ground Truth and *PR* is the abbreviation for the area of Predicted Results.

2) Trajectory Smoothing: Most existing vehicle trajectory datasets, such as NGSIM [6], only provide raw trajectories, which are noisy and therefore hard to use directly due to the jerky trajectories. Figure 9 visualizes the results of the trajectories for one of the vehicles with and without smoothing. The Figure 9(a) shows the result with equal scaling of the x and y axes. It is hard to find the difference between the trajectories with (RED) and without (GREEN) smoothing. However, when the x axis is enlarged in figure 9(b), the trajectory without smoothing (GREEN) is much jerkier than the one with smoothing (RED).

Finally, the video[2] includes all the dynamic results proposed in the pipeline.

3) Scenario: We tested the algorithm based on the UrbanFlow dataset. We selected pairs of interacting vehicles with the driving directions of *GS* and *TL* or *TL* and *TR* from the dataset. Figure 10 shows a pair of two interacting vehicles. The blue rectangle with *E* is the ego car and the green rectangle with *T* is the target vehicle. The input state

[2]https://www.youtube.com/watch?v=cnfH1pi5pJQ

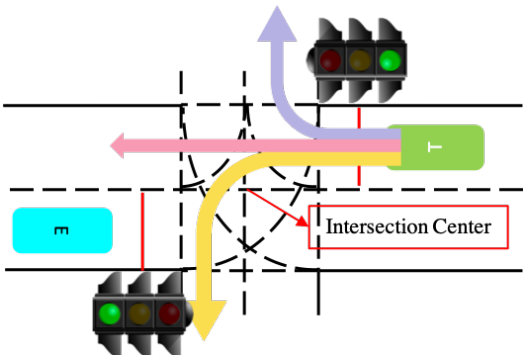

Fig. 10: Scenario of interacting vehicles pair.

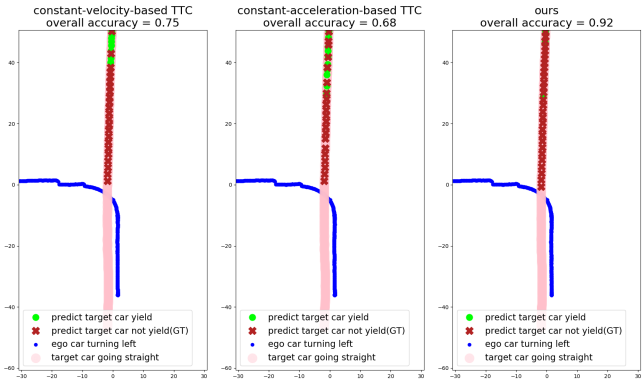

Fig. 11: Comparison of the intention prediction results between TTC and our algorithm.

includes velocities, heading angles and relative distances to the intersection center of both ego and target cars.

4) Intention Prediction: According to the described state, the intention network predicts the direction intention as well as the yield intention. Figure 12 visualizes the results of direction and yield intentions. All the ego cars approach the intersection (intersection center is coordinate $(0,0)$) from the bottom. Different colors with marker $\bullet$ show the results of direction predictions when the target vehicle reaches that position and the other colors with marker **X** present the yield intention prediction results. A quantitized comparison between the TTC (time-to-collision) approach and our algorithm is shown in Table III. In TTC model, we made assumptions of constant velocity and constant acceleration set-up for yield estimation. The result shows that our algorithm is better for estimation at the proposed scenario. Both of these three methods have similar results on threshold time of 0.5s, 1.0s, 1.5s. Figure 11 shows one of comparisons among constant-velocity-based TTC, constant-acceleration-based TTC and our proposed method. It shows that the TTC-based methods are more unstable for prediction results especially when the vehicles are far away from the

TABLE III: Comparison with TTC.

Method	Average accuracy
TTC w/ constant velocity	0.74
TTC w/ constant acceleration	0.62
Ours	0.92

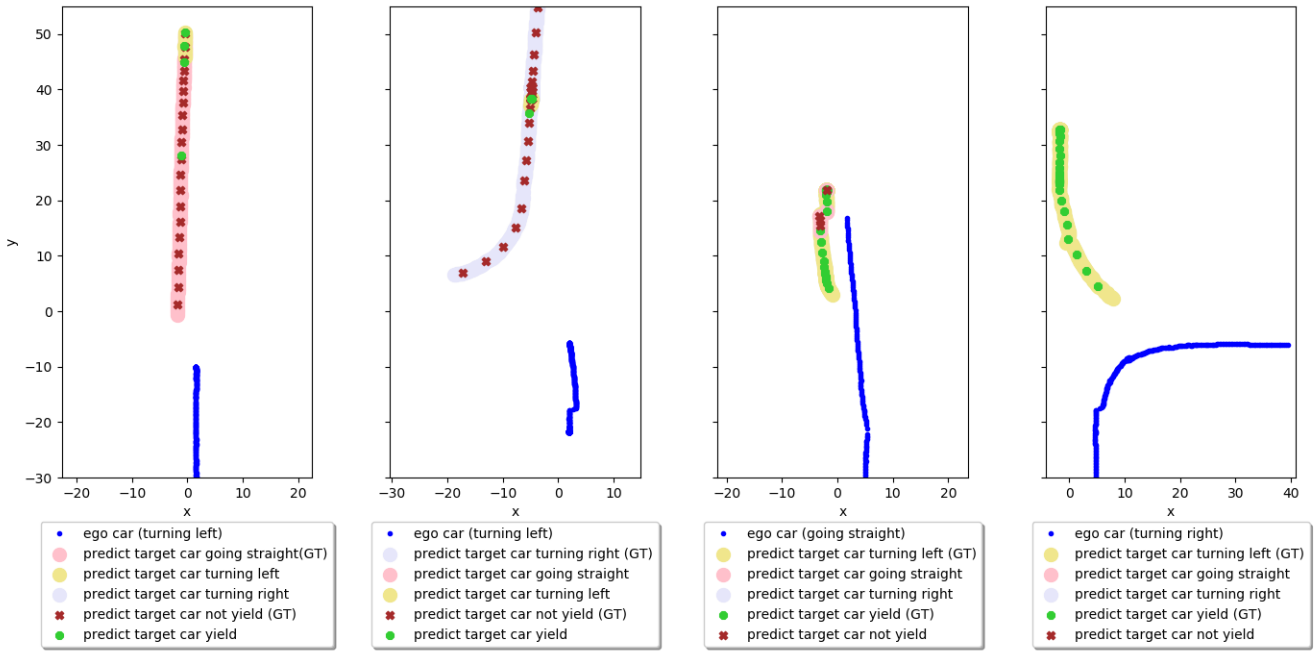

Fig. 12: The direction and yield prediction result of selected interacting pairs. GT means that the corresponding intention is the ground truth of the selected pair. Direction intention of ego cars is noted in parentheses in the ego car legend.

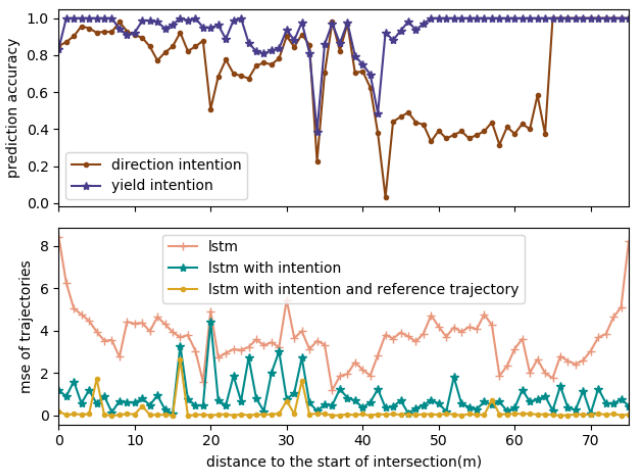

Fig. 13: Direction and yield intention as well as MSE of trajectories prediction for the target vehicle.

intersection.

Figure 13 shows the prediction accuracy with respect to the distance to the start of the intersection for the target vehicle. The prediction accuracy drops at the 35m to 40m range, which is due to some trajectories started at this range and lacked of history frame information, which will lead to an incorrect prediction. The number of samples above 50 m for yield and 65 m for direction is too small for statistical significance, so those results can be ignored.

5) Trajectory Prediction: We compared the mean squared error (MSE) results between the trajectory prediction with and without intention results and reference trajectories. Table IV presents the average MSE for different methods and

TABLE IV: MSE of trajectory predictions.

Method	Average MSE (m)
LSTM	3.71
LSTM w/ intention	0.89
LSTM w/ intention and reference trajectory	0.18

Figure 13 shows the MSE with respect to the distance to the start of the intersection for the target vehicle. Once past the start position of the intersection, the trajectories become diverse due to various direction intentions, and as a result, the MSE of the trajectory prediction increases. In Figure 13, for the trajectory prediction part, when the vehicles are observed within the scene initially, the trajectory history information is lacking for accurate prediction without reference trajectories. Meanwhile, when vehicles enter the intersection and begin to do turning behavior, the trajectory prediction becomes harder without reference trajectories.

V. CONCLUSIONS

In this paper, we propose a pipeline called UrbanFlow which is used to deal with traffic data collected by drones in urban environments. The raw data are processed through video stabilization, vehicle detection, map construction and coordinate transformation, vehicle tracking, and trajectory smoothing. Moreover, the paper proposes a method for driving behavior predictions and tests it on the UrbanFlow dataset. The following work for improving the dataset will focus on increasing the quantity of the dataset. More types of urban scenarios like T-intersections, stop-sign intersections and yield intersections will be included in the dataset.

REFERENCES

[1] Z. Qiao, K. Muelling, J. Dolan, P. Palanisamy, and P. Mudalige, "Pomdp and hierarchical options mdp with continuous actions for autonomous driving at intersections," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2377–2382.

[2] C. Dong, Y. Chen, and J. M. Dolan, "Interactive trajectory prediction for autonomous driving via recurrent meta induction neural network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1212–1217.

[3] Z. Qiao, K. Muelling, J. M. Dolan, P. Palanisamy, and P. Mudalige, "Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1233–1238.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[5] "VTD homepage." 2019. [Online]. Available: https://vires.com/vtd-vires-virtual-test-drive

[6] "NGSIM homepage. FHWA." 2005-2006. [Online]. Available: http://ngsim.fhwa.dot.gov.

[7] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[9] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[10] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[11] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 1162–1167.

[12] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1665–1670.

[13] B. Tang, S. Khokhar, and R. Gupta, "Turn prediction at generalized intersections," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1399–1404.

[14] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011.

[15] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1764–1771.

[16] G. S. Aoude, B. D. Luders, K. K. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1855–1862.

[17] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[18] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in computer vision*. Elsevier, 1987, pp. 726–740.

[20] S. Overflow, "cv2 motion euclidean for the warpmode in ecc image alignment method."

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[22] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," *arXiv preprint arXiv:1801.05746*, 2018.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] Y. Chan, A. Hu, and J. Plant, "A kalman filter based tracking scheme with input estimation," *IEEE transactions on Aerospace and Electronic Systems*, no. 2, pp. 237–244, 1979.