2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

A Field-Based Representation of Surrounding Vehicle Motion from a Monocular Camera

Lifeng Zhu[1,3], Xuanpeng Li[1], Wenjie Lu[2] and Yongjie Jessica Zhang[3]

Abstract— **Sensing and presenting on-road information of moving vehicles is essential for fully and semi-automated driving. It is challenging to track vehicles from affordable onboard cameras in crowded scenes. The mismatch or missing data are unavoidable and it is ineffective to directly present uncertain cues to support the decision-making. In this paper, we propose a physical model based on incompressible fluid dynamics to represent the vehicle's motion, which provides hints of possible collision as a continuous scalar riskmap. We estimate the position and velocity of other vehicles from a monocular on-board camera located in front of the ego-vehicle. The noisy trajectories are then modeled as the boundary conditions in the simulation of advection and diffusion process. We then interactively display the animating distribution of substances, and show that the continuous scalar riskmap well matches the perception of vehicles even in presence of the tracking failures. We test our method on real-world scenes and discuss about its application for driving assistance and autonomous vehicle in the future.**

I. INTRODUCTION

With the development of computer vision techniques, onboard cameras are regularly integrated into the advanced driver assistance system (ADAS) and autonomous driving system. Visual data from cameras could generate multiple modality features, such as color, shape and learned features via deep convolution network, which are beneficial to scene understanding in traffic context.

As the automobiles moving forward, the extracted motion of on-road vehicles from the front view provides critical information to make decisions. Understanding the motion of surrounding vehicles involves detection, multi-target tracking, trajectory analysis, and risk estimation in the context of active safety. It is allowed to perceive and predict potential collisions and take actions to avoid most traffic accidents if useful information in the front view is efficiently presented. However, because of the built-in drawbacks of cameras, there are several common problems in the sensed motions of surrounding vehicles. Due to occlusion, disruptive illumination or other unfavorable imaging conditions, the sensed data may suffer from missing detections or mismatching vehicles. Even in combination with other on-board sensors such as Radar or Lidar, these problems are still unavoidable due to the complexity of real environments and various driving situations. How to efficiently present the imperfect data into ADAS and autonomous vehicle is still a challenge.

Occupancy grid is a typical solution to present the raw sensor data [21] [6]. The spatial information of vehicles, pedestrians or other obstacles is quantified into a distribution of occupancy by using probabilities or belief mass, and it is then shown as a grid map to notify the drivers whether there are risks on road. Because of the occlusion, only the occupancy in the visible region is displayed. If a vehicle is failed to be tracked due to occlusion, it has high probability to reappear. Therefore, in order to improve driving safety, we aim to develop a better representation of surrounding vehicles' movement that reveals spatiotemporal information even if tracking is lost. Besides, standard occupancy grid only displays the occupancy status at the current frame. We expect to improve the efficiency of the representation by bringing short-term prediction in the map, in order to reduce the burden of decision-making in a limited time frame.

In this paper, we introduce a physically-based model to efficiently encode the spatiotemporal information of on-road vehicles in a riskmap and propose to use it to assist decision making. We simulate the evolving distribution of substance around surrounding vehicles and use it to guide vehicle's behavior. The advection and diffusion process in the simulation help reveal hidden spatial information in presence of missing data. In order to incorporate the short-term prediction, we modify the standard diffusion process and put emphasis on the diffusion under tracked velocity. Based on the proposed model, we also conservatively emit warning or recommend actions to prevent collisions.

In the following sections, after introducing related works in Sec.II, we talk about the system and computation of the proposed field-based riskmap in Sec.III. We test our method with real-world data captured from a front camera in Sec.IV and provide discussions and conclude remarks in Sec.V.

II. RELATED WORK

In this section we review recent studies that have contributed to the tasks of object detection, multi-object tracking and motion representation on road.

Benefited from the fast development of deep convolution networks [10], [9], [25], [23], [20], object detection has achieved a great advance. Recent object detection work can be separated into two categories: two-stage detectors and one-stage detectors. The former is the dominant paradigm in modern object detection which sacrifices speed for accuracy,

[1] Lifeng Zhu and Xuanpeng Li are with the State Key Laboratory of Bioelectronics, Jiangsu Key Lab of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, 210096, Nanjing, China `lfzhulf@gmail.com`, `li_xuanpeng@seu.edu.cn`

[2] Wenjie Lu is with the Traffic Management Research Institute of the Ministry of Public Security, 214151, Wuxi, China `luwenjie0122@msn.cn`

[3] Lifeng Zhu and Yongjie Jessica Zhang are with Carnegie Mellon University, Pittsburgh, PA 15213, USA `lifengz2@andrew.cmu.edu`, `jessicaz@andrew.cmu.edu`

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

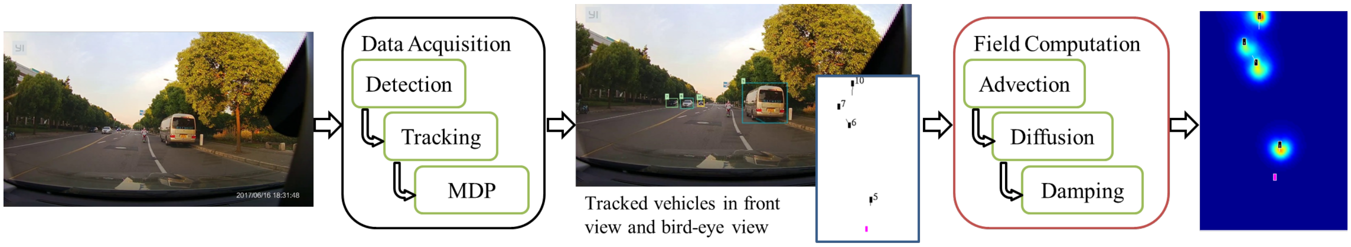


Fig. 1. The pipeline of our field-based riskmap system.

while the latter focuses on the moderate accuracy with higher speed. Among the two-stage detectors, R-CNN [10] is the first to utilize deep convolution network features yielding large gains in accuracy. Following R-CNN, Fast R-CNN [9] and Faster R-CNN [25] were proposed with end-to-end training and Region Proposal Network (RPN) to improve R-CNN serials detectors. Numerous extensions to this framework, such as R-FCN [17], deformable convolution network [7], and Mask R-CNN [11], have been proposed. On the other hand, YOLO [23] and SSD [20] have recently renewed research in one-stage methods, which have been tuned for speed/accuracy trade-off. RetinaNet handles the foreground-background class imbalance problem to improve the performance of one-stage detector about 10% AP on COCO test-dev [19]. A more recent light-head R-CNN [18] achieves a better speed/accuracy trade-off by means of reducing the feature dimension of two-stage network.

Recent research in multi-object tracking (MOT) has focused on the tracking-by-detection and learning-to-track principals. Both batch methods [15], [4] and online methods [12], [1] have explored how to learn a similarity function for data association. More recent studies on MOT have integrated the hierarchical features from deep convolution networks [29], [16] and correlation filter [22]. In addition, reinforcement learning algorithm has been proposed to link data in online MOT, and Markov decision processes (MDP) have been proved to be suitable for dynamic environment [31]. Multi-object can be modeled as multi-agent which has its own lifetime to perform certain tasks and maintain certain states.

Description of surrounding vehicles' motion in traffic scenes is an important issue. Trajectory analysis [8], [26] and occupancy grid mapping [5], [27], [6] are two major approaches to compactly represent other vehicles' motion. Trajectory analysis mainly focuses on the temporal characteristic of vehicle's motion. In [26], both a monocular camera and a stereo camera are used to obtain vehicles' trajectories in front of the ego-vehicle. The behaviors of the observed trajectories are then learned by unsupervised learning method. Panoramic camera arrays are used to capture a full surround view, and trajectories of surrounding vehicles are then extracted and classified by the hidden Markov model (HMM) into certain state like overtaking and lane change [8]. On the other side, occupancy grid mapping essentially relies on probabilistic spatial information, e.g., using recursive Bayesian filtering [6]. Dynamic probabilistic drivability maps [27] are constructed to represent the on-road environment with a variety of sensors including camera, radar and lidar. In [5], a probabilistic grid-based approach is used to exploit surrounding vehicles' movements and infer the presence and location of occluded road surface as a complementary method to already existing road detection systems.

In addition, the field-based approaches have been proposed for automatic vehicle guidance for a long history. Electric field model is used to interpret the vehicle's motion as an electron within an electric field, and the system is transformed into a riskmap reflecting the risk at a certain position in the dynamic environment [24]. In [30], a vehicle collision avoidance system in a full two-dimensional field is considered with lane, road, car, and velocity potential function components. More recent work [14] assesses collision risks by risk potential modeling of predicted motion of the surrounding vehicles in various driving conditions.

III. SYSTEM

As shown in Fig. 1, we take the video streams from the front monocular camera as the input of our system. To acquire the field of various moving vehicles, we need to track vehicles between continuous frames, which is a problem of data association. After detecting and tracking vehicles in the front view, we convert their positions into a bird eye view with estimated speed and direction. Taking the moving points in the bird eye view as the boundary conditions, we simulate the animation of the substances around each vehicle by using advection and diffusion, and finally display the distribution to guide the ego-vehicle. In the following, we will discuss about the details of each step.

A. Data Acquisition

Tracking of surrounding vehicles captured by front facing cameras is still challenging in crowded scenes, and the major problem is the association of noisy object detections between various frames. In our framework, the tracking-by-detection principle is coupled with Markov decision process (MDP), which models the lifetime of a target [31]. A MDP maintains multiple states, such as tracked and occluded states for each object, according to calculated overlay region based on detected results.

Vehicle detection. In our work, we adopt the Xception-like light-head R-CNN architecture [18] to detect the vehicles in video streams. Light-head R-CNN changes the heavy head of two-stage object detectors, which has negative influence on the computational speed, into thin feature maps.

It could not only improve accuracy but also save memory and computation time during training and inference. To improve efficiency of the vehicle detection algorithm without losing too much accuracy, we choose a small backbone of Xception-like network instead of the resnet-101 network.

In order to generate thin feature maps, a large separable convolution kernel pair has been deployed, and we reduce the number of kernels to 64. Besides, PSRoI pooling is another important component to reduce R-CNN overhead and improve the performance simultaneously. Meanwhile, we use Region Proposal Network (RPN) with non-maximum suppression (NMS) to automatically remove heavily overlapping proposals. We use our own dataset of road scenes in China to finetune the network with ImageNet pretrained models. More details can be referred to [18].

Multi-vehicle tracking. Multi-vehicle tracking involves the single object tracking and multi-object association between frames. We use the online tracking method [31] to update multiple MDP agents to track objects in a reinforcement learning way. Tracking-Learning-Detection (TLD) tracker is originally deployed for long-term pedestrian tracking [13]. It is needed to modify aspect ratio of the template, while the aspect ratio of vehicles varies with the orientation where they are observed. We adopt the aspect ratio of 1.5 as mentioned in [8], and we use four states including active, tracked, lost, and inactive states, and the transition relationship among various states is used in the same manner as [31].

Since each object has its own lifetime, this framework can naturally handle the birth/death and appearance/disappearance of various vehicles by transiting their state in the MDP. It means that even though some vehicles are lost in the front view, this system could retain its last state and then update it. In our work, the lost state is the most important stage since in environment perception one should have cognition on the "blind" targets. We could predict their speeds and orientations for the sake of active safety. In the MDP framework, a lost state could be transited to a tracked state, or marked as inactive in light of a pre-trained SVM classification. In our experiments, we use the annotated KITTI dataset to train our MDP parameters.

B. Field-Based Representation

With the obtained relative trajectory of tracked vehicles, we are able to compute the riskmap based on a field-based representation. Let $\mathbf{p}_i^{(j)} = (x_i^{(j)}, y_i^{(j)})$ denote the relative position of the ith vehicle at the jth frame, where $x_i^{(j)}$ and $y_i^{(j)}$ represent the lateral and longitudinal distance between the ith vehicle and the ego-vehicle at the jth time frame. We compute the relative velocity of the ith vehicle at time j as $\mathbf{v}_i^{(j)} = \mathbf{p}_i^{(j)} - \mathbf{p}_i^{(j-1)}$. Note that we do not require the extracted trajectory to be perfect. In case the position of the ith vehicle at the last frame is failed to be tracked, we keep its velocity unchanged. At the beginning, we simply set the relative velocity of all surrounding vehicles to be zero.

In our model, we compute the distribution of the substance around each vehicle as a density map. We use incompressive flow to model the behavior of substance under the moving vehicles. Instead of directly solving the Navier-Stokes equation on a grid, similar to the fluid simulation pipeline in computer graphics [3], we decompose the computation of the density flow into an advection, a projection and a diffusion stage. Suppose at the jth frame, the detected vehicles form a set C_j. We take the status $\{(\mathbf{p}_i^{(t)}, \mathbf{v}_i^{(t)})\}, i \in C_j$ at the jth frame as an example to describe the involved computation of the corresponding field as a density map $D(x,y)^{(j)}$. After $D(x,y)^{(j)}$ is obtained, we advance to the computation at the $(j+1)$th frame. Because we update the field frame by frame, we will drop the superscript for simplicity in the following. The input and output are illustrated in Fig. 2, where we also show the position of the ego-vehicle in magenta and other detected vehicles in black.

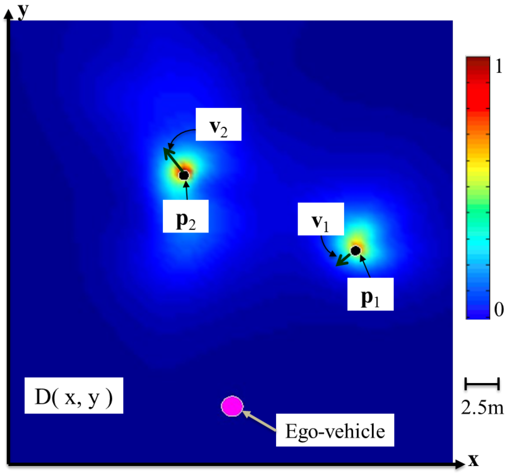


Fig. 2. At each time frame, we compute the field $D(x,y)$ from the position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$ of each tracked vehicle.

As the vehicles move, we model the velocity as the boundary condition in the advection process. Specifically, we first solve the differential equation

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u}$$

for the velocity field $\mathbf{u}$, with the boundary condition such that the velocity at the grid where the ith vehicle occupies equals to the sensed velocity $\mathbf{u}(x_i, y_i) = \mathbf{v}_i, i \in C_j$. After the velocity field is obtained, we then project the velocity field to a divergence-free field $\mathbf{w}(x,y)$ by using Helmholtz-Hodge Decomposition [2]. In this case, the substance will be incompressible under the advection guided by $\mathbf{w}(x,y)$. Therefore we compute the advection of the substance $D_1(x,y)$ by solving

$$\frac{\partial D_1}{\partial t} = -\mathbf{w} \cdot \nabla D_1$$

after $\mathbf{w}$ is obtained.

Diffusion is the other effect of the moving vehicles on substances. In our model, each vehicle is supposed to persistently emit substances in the map. In this case, the existence of surrounding vehicles can be persistently observed even with a long-term advection of the density field. Formally,

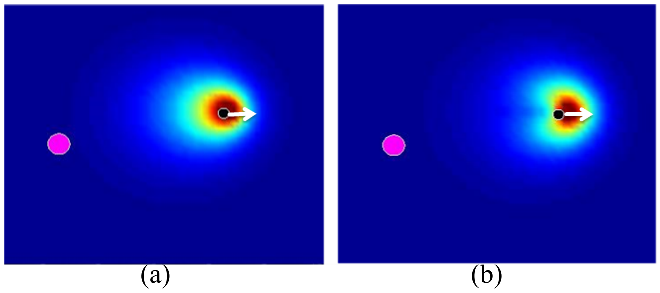


Fig. 3. The field encodes short-term prediction of the surrounding vehicles when using our anisotropic diffusion model.(a)Density field from isotropic diffusion;and (b)density field from anisotropic diffusion.

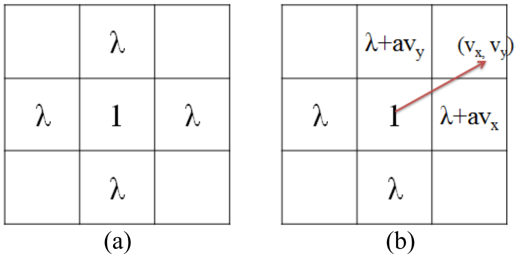


Fig. 4. The isotropic (a) and anisotropic (b) stencils used in the diffusion stage.

we solve the diffusion equation

$$\frac{\partial D_2}{\partial t} = -\lambda \Delta D_2 + f,$$

where we set the external source of diffusion at the position of each vehicle $f(x_i, y_i) = s$. The parameter λ controls the rate of diffusion and the parameter s measures the strength of the source. In our implementation we set $\lambda = 1$ and $s = 1$. We take the density field from advection D_1 as the initial condition to start the diffusion and obtain the density field D_2 after the diffusion. We also introduce a damping stage to prevent the space being fully filled with substances if the sources keep emitting substances. At the end of the frame, we damp the density field and display the field as $D = \omega D_2$, where the damping coefficient ω is set to 0.98 in our implementation.

By setting the boundary conditions as above, we adopt the solver proposed in [28] to solve the advection and diffusion equations, as well as the projection for incompressibility.

C. Prediction for Planning Assistance

We further extend our model to support a short-term prediction. In Fig. 3(a) we show an obtained density field by using the method described in the previous subsection. From the figure, we see that our model estimates the risk in the given domain including invisible regions by using the diffusion and advection. However, the distribution of the density around each vehicle is isotropic. In order to better assist driving, we hope to predict the change of the field in a short time and display the information in the density map. Our solution is to increase the diffusion along the moving direction of each vehicle. In this case, the higher density of the substances help warn the drivers the potential position of surrounding vehicles in a short time.

In our implementation, we introduce anisotropy along the velocity of the tracked vehicles in the diffusion step to create the desired risk distribution. As the diffusion equation is solved by iteratively applying isotropic Gaussian filter in the solver provided in [28], we directly update the stencil in the filter to model the anisotropy in the diffusion. As shown in Fig. 4(a), the diffusion stencil in isotropic diffusion is uniform around each grid, i.e., $w(i \pm 1, j) = \lambda$. We modify the stencil by taking the velocity field in the grid into consideration, as illustrated in Fig. 4(b). If the velocity at the grid (i, j) is (v_x, v_y), we update the stencil at neighboring grids as $w(i + sgn(v_x), j) = \lambda + a \cdot v_x$ and $w(i, j + sgn(v_y)) = \lambda + a \cdot v_y$, where a is the parameter controlling the anisotropy and the sign function $sgn(x) = \frac{d}{dx}|x|$ if $x \neq 0$, $sgn(x) = 0$ when $x = 0$. Note that it is possible to compute anisotropic Laplace operator on the grid and solve the diffusion equation implicitly. We opt to use this simple updating scheme because in our application we only aim for a simple yet efficient method to help make decision rather than a high precision in serious physical simulation. See the obtained density field from anisotropic diffusion in Fig 3(b), which encodes short-term prediction of surrounding vehicles.

IV. EXPERIMENTS

In this section, we experiment our method with real-world data collected from one single camera on Chinese urban road with free-flow traffic. We collect 1,000 images of a resolution 1,920x1,080 with well annotated vehicle data, which are split into 800 train set and 200 validation set. Our detector is end-to-end finetuned with ImageNet pretrained model of a Xception-like architecture on 4 TITAN X GPUs using synchronized SGD. We use a setting with a weight decay of 0.0001, a momentum of 0.9, and a learning rate of 0.001 for 50 epochs, and each mini-batch has 2 images per GPU. After finetuning, the detector achieves mAP 92% of PASCAL metric with a speed of 90 fps. Then, we use our trained model to detect vehicles on the raw video clips. Following the detection process, our MDP trakcer has been trained on a sequence of our collected data, using ground-truth annotations. All tracked data are then transferred into real-world coordinate as a bird eye view by inverse perspective mapping. We show the corresponding field-based representation in Fig. 5.

The riskmap of field-based representation created from our method is relatively stable with respect to noises in the vehicle tracking. As shown in Fig. 6(a), the 40th vehicle is mismatched as the 10th vehicle when it blocks the 10th vehicle. If we display the trajectory of the surrounding vehicles, an abnormal trajectory would be observed, as shown in Fig. 6(c). With our method, the detected velocity is noisy or even erroneous due to the incorrect data association, but the generated field still looks smooth with the help of the physically-based model, as shown in Fig. 6(b).

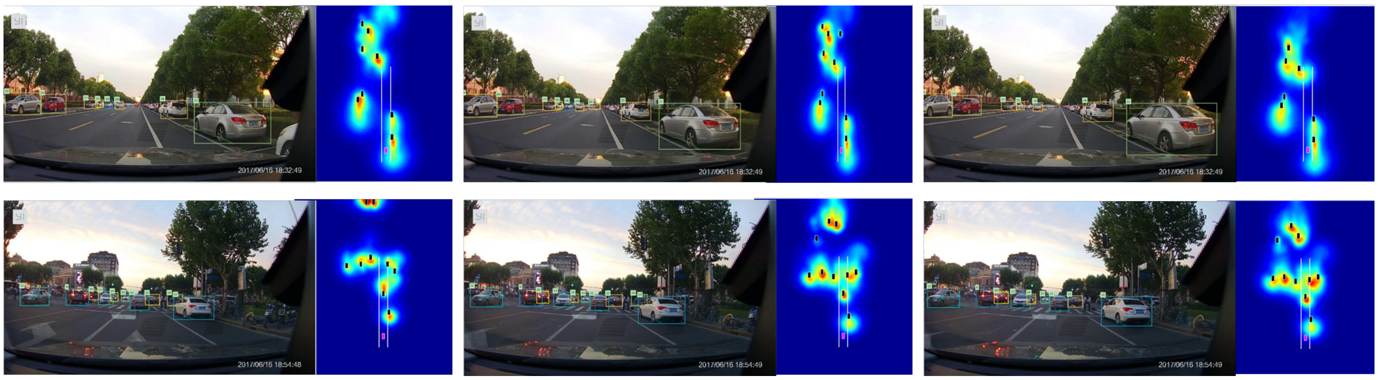
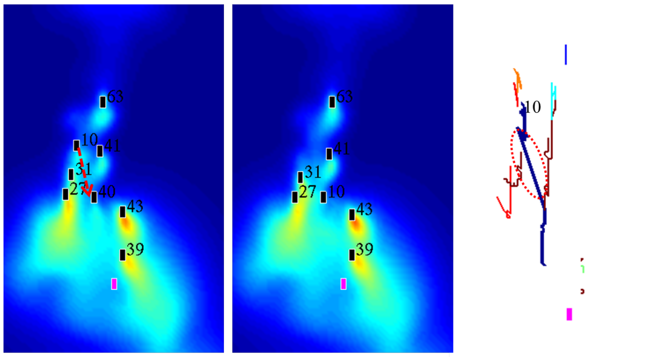


Fig. 5. The field-based representation computed from real-world captures. Note that we do not filter the trajectories of surrounding vehicles and directly use the tracked positions to create the map.



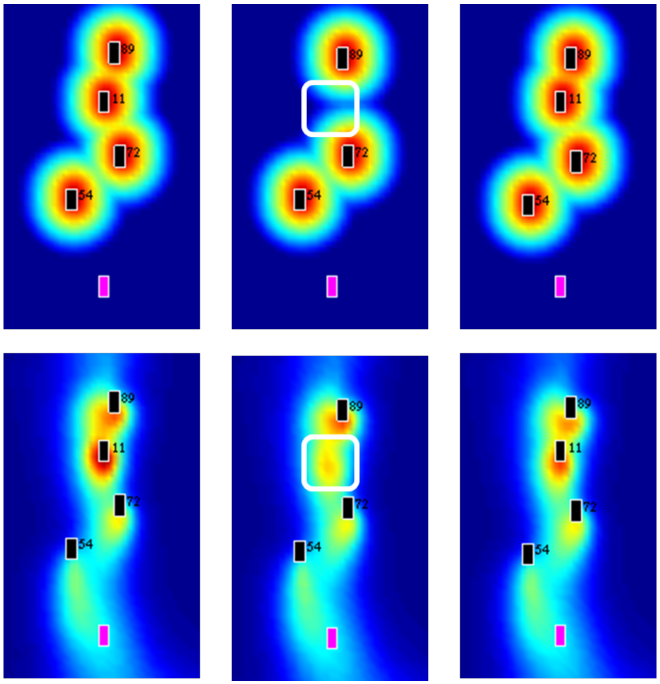
(a) (b) (c)

Fig. 6. The generated field-based representation (b) under the cases when surrounding vehicles are not correctly matched (a), comparing with the trajectories displayed in (c).

We highlight the behavior of our method using a case with tracking failure in Fig. 7, where the 11th vehicle is lost tracking in (b) compared with (a). Because it takes time to advect and diffuse the substances, even if a vehicle disappears due to tracking failure, the evolving substances from it in the previous frames gradually damp and still leave useful hints to the driving system. Due to the time-coherence, the missing vehicle is of high probability to reappear and it makes sense to keep the information from it. Comparing with the distance field shown in the upper row of Fig. 7, when the vehicles disappear and reappear, the distance field drastically changes and it may misguide the decision-making.

We test the computation of the proposed riskmap on a single CPU thread with 2.5GHz, for a field with resolution 80x128, it takes about 25 milliseconds to update one frame of the field from the tracking data.

As an application of the proposed field-based representation of riskmap, we can use it to provide warning, because our vehicle can intuitively observe whether it is close to other vehicles in the front view based on the proposed representation. As the information is continuously quantified, we can also provide recommendation for driving actions by analyzing the field around the ego-vehicle. If the ego-vehicle is at a grid with higher density, we regard it with higher risk to collide with surrounding vehicles. Our system then recommends to drive the ego-vehicle toward the negative gradient direction, which is the fast descending direction to decrease the density. The level of the recommendation is set to be proportional to the density at the grid.



(a) (b) (c)

Fig. 7. A case study (a) with the 11th vehicle disappears (b) and reappears (c). Top row: the distance field; Bottom row: the field-based representation from our method.

V. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we propose a physically-based model to present noisy sensed data from an on-board front camera into a riskmap. By introducing the advection and diffusion process from sensed vehicles, we simulate the substances around them and then display the evolving distribution as a density field. We also modify the diffusion stage to enable a short-term prediction of the distribution along the moving direction of each vehicle and propose to give driving recommendations according to the proposed field-based representation. By experimenting our method with real-world data, we show the efficiency of the proposed model.

In this work, we only use the sensed data from a single on-board camera and shows the efficiency of our method to display the data. Although the advantage of our model is to deal with imperfect data, in case when we have trajectory data with high quality, the proposed model is still beneficial in the sense that it provides a short-term prediction to help decision-making. Note that our model takes trajectories of moving objects as the input, therefore the sensed data can be from more than one single on-board camera. As long as the motion of any obstacle around the ego-vehicle can be sensed, our method can be used to model a riskmap. Therefore, we can use stereo-cameras or even combine multiple Lidars or Radars and extract on-road information, such as pedestrians, motorcycles, and other obstacles, to make the map more effective for decision-making.

Driving assistance and automated driving are high-level tasks which require the understanding of the driving semantics or intention, while our method so far only takes the estimated trajectories and creates information at the geometric level. Besides the information collected from the on-board sensors, in the future we will also explore how to combine the proposed riskmap with detected or pre-stored road map, and make the driving recommendation more intelligent.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Jiangsu Province under Grants No. BK20160700 and BK20150634, the National Science Foundation of China (NSFC) under Grants No. 61502096, 61673114 and 61773219, and the National Key Technologies R&D Program under Grants No.2016YFB1001300.

REFERENCES

[1] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1218–1225, 2014.

[2] H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer. The Helmholtz-Hodge decomposition - a survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1386–1404, 2013.

[3] R. Bridson. *Fluid simulation for computer graphics*. CRC Press, 2015.

[4] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1846–1853, 2013.

[5] E. Casapietra, T. H. Weisswange, C. Goerick, F. Kummert, and J. Fritsch. Building a probabilistic grid-based road representation from direct and indirect visual cues. In *IEEE Intelligent Vehicles Symposium (IV)*, pp. 273–279, 2015.

[6] C. Cou, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessire. Bayesian occupancy filtering for multitarget tracking: an automotive application. *The International Journal of Robotics Research*, 25(1):19–30, 2006.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017.

[8] J. V. Dueholm, M. S. Kristoffersen, R. K. Satzoda, T. B. Moeslund, and M. M. Trivedi. Trajectories and maneuvers of surrounding vehicles with panoramic camera arrays. *IEEE Transactions on Intelligent Vehicles*, 1(2):203–214, 2016.

[9] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[11] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask R-CNN. *arXiv preprint arXiv:1703.06870*, 2017.

[12] J. Hong Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1392–1400, 2016.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[14] K. Kim, B. Kim, K. Lee, B. Ko, and K. Yi. Design of integrated risk management-based dynamic driving control of automated vehicles. *IEEE Intelligent Transportation Systems Magazine*, 9(1):57–73, 2017.

[15] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 685–692, 2010.

[16] L. Leal-Taix, C. Canton-Ferrer, and K. Schindler. Learning by tracking: siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33–40, 2016.

[17] Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 379–387, 2016.

[18] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-Head R-CNN: in Defense of Two-Stage Object Detector. *arxiv preprint 1711.07264*, 2017.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: single shot multibox detector. In *European Conference on Computer Vision*, pp. 21–37. Springer, 2016.

[21] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M.-M. Meinecke. Stereo-camera-based urban environment perception using occupancy grid and object tracking. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):154–165, 2012.

[22] S.-H. Park, K. Lee, and K.-J. Yoon. Robust online multiple object tracking based on the confidence-based relative motion network and correlation filter. In *IEEE International Conference on Image Processing (ICIP)*, pp. 3484–3488, 2016.

[23] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[24] D. Reichardt and J. Shick. Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway. In *Proceedings of the Intelligent Vehicles*, pp. 74–78, 1994.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[26] S. Sivaraman, B. Morris, and M. Trivedi. Learning multi-lane trajectories using vehicle-based vision. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2070–2076, 2011.

[27] S. Sivaraman and M. M. Trivedi. Dynamic probabilistic drivability maps for lane change and merge driver assistance. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2063–2073, 2014.

[28] J. Stam. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.

[29] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pp. 100–111. Springer, 2016.

[30] M. T. Wolf and J. W. Burdick. Artificial potential functions for highway driving with collision avoidance. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3731–3736, 2008.

[31] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: online multi-object tracking by decision making. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4705–4713, 2015.