Context-Aware Tracking of Moving Objects for Distance Keeping

Wenda Xu, Jarrod Snider, Junqing Wei, and John M. Dolan

***Abstract*— We propose a robust object tracking algorithm for distance keeping. Taking advantage of a context-based region of interest, we are able to maximize the performance of each sensor, and reduce the computation time since we only focus on the targets inside the region. Tracking targets in road coordinates enables finding the distance-keeping target on any curved road, while a commercial Adaptive Cruise Control (ACC) system works best on straight roads. We demonstrate that the overall performance of the proposed algorithm is better than that of a commercial ACC system. The distance-keeping target can either be used for lane following for a standalone ACC system or an autonomous vehicle. Our object tracking algorithm can also be extended to find the target of interest for lane changing or ramp merging for an autonomous vehicle.**

I. Introduction

Over one million deaths and fifty million injuries are due to traffic accidents every year worldwide, most of which are caused by human mistakes or distractions. To improve driving safety, autonomous vehicle technology has received intensive attention over the past few decades [1], [2], [3]. Before full autonomy arrives, semi-autonomy may be widely introduced in the near future thanks to developments in sensor engineering and computer technology. In particular, Advanced Driver Assist Systems (ADAS) have been widely used in high-end cars, of which Adaptive Cruise Control (ACC) [4] is the one most deployed. Basically, an ACC system can automatically adjust the vehicle speed to keep a reasonable distance from a leading vehicle. For an ACC system, the quality of the distance-keeping target (also called ACC target) is a key factor, including the reliability of finding the leading target and the smoothness of the range and range rate of the target. Our goal in this paper is to provide a robust and accurate distance-keeping target.

Distance keeping is important not only for ACC systems but also for an autonomous vehicle's lane following on highways or urban roads. In unstructured environments, the autonomous vehicle is trying to reach goals rather than just following a specific lane. Therefore, as for a common robot, motion planning plays a key role for an autonomous vehicle to avoid static and dynamic obstacles and reach its goal. However, when the autonomous vehicle is driving down road lanes, the movement of the vehicle is far more constrained. Unlike a common robot, the autonomous vehicle has to follow traffic rules and cooperate with human-driven vehicles. Most of the time, the autonomous vehicle is following a specific lane at a safe distance from the vehicle ahead. Therefore, a high-quality distance-keeping target is essential for the safety and comfort of an autonomous vehicle.

Fig. 1: The CMU autonomous vehicle research platform

This paper describes the context-aware tracking of moving objects (CATMO) for distance keeping used by the new CMU autonomous vehicle research platform (as shown in Fig. 1). In order to provide a robust result, data from multiple sensors are fused, including LIght Detection And Ranging (LIDAR), RAdio Detection And Ranging (RADAR), and cameras. Given the behavioral context, a region of interest is generated in the road network. We then find the candidate targets that are inside the region of interest and project them into road coordinates. The distance-keeping target is obtained by associating all candidate targets from different sensors. The proposed algorithm has been tested on the CMU autonomous vehicle research platform for one year over more than 5000 miles.

The rest of the paper is organized as follows. An overview of related object tracking research is described in Section II. The system architecture used in our vehicle is described in Section III. Our tracking system is presented in Section IV. The performance of the system and examples of comparison to a commercial ACC target are shown in Section V. Conclusions are given in Section VI.

II. Related Work

Commercial Adaptive Cruise Control systems have been widely used in high-end cars [4]. However, all of them assume the road is straight or almost straight, which means they only work well on highways. In this work, we are able to track targets for any road shape by considering the behavioral context [5], [6].

This work was supported by NSF Grant CNS1035813

Wenda Xu, Jarrod Snider, and Junqing Wei are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA {wendax,junqingw,jmsnider}@andrew.cmu.edu

John M. Dolan is with the Department of Electrical and Computer Engineering and Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA jmd@cs.cmu.edu

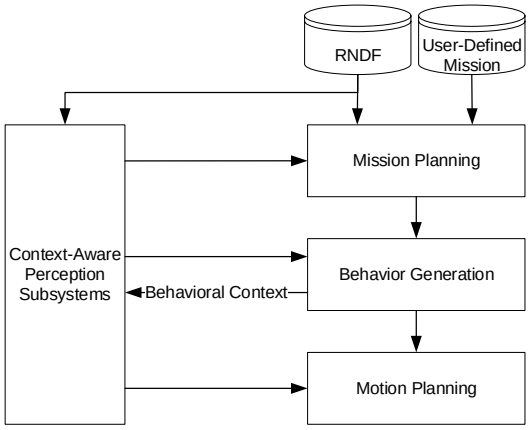


Fig. 2: System architecture

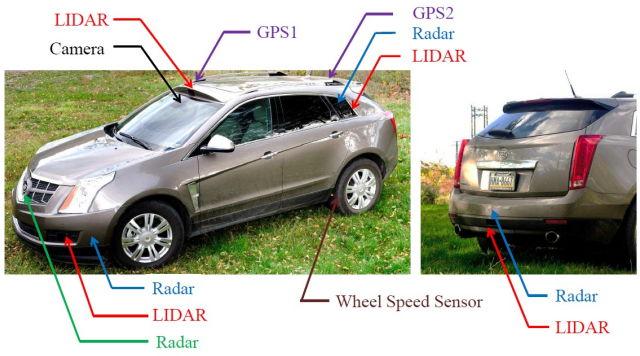
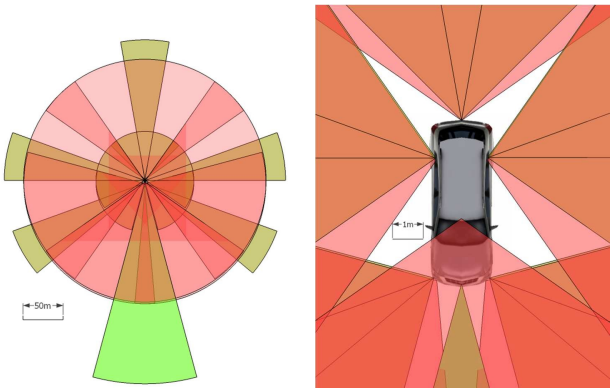


Fig. 3: Sensor installation [2]



(a) Sensor field of view (b) Blind spots

Fig. 4: Perception system coverage [2]

During the DARPA Urban Challenge held in 2007, the autonomous vehicle "Boss" built by CMU's Tartan Racing Team won the competition [7], which took place in an urban environment with low-density and low-speed traffic. The high-definition 3D LIDAR sensor (i.e., Velodyne HDL-64) on top of the vehicle provided 3D point clouds dense enough for estimating the shape of cars, curbs and undrivable areas. With such a high-definition 3D LIDAR sensor, "Boss" was able to track all moving objects around the vehicle and generate a static map.

In CMU's autonomous vehicle research platform [2], only production-grade LIDAR and RADAR sensors are available, and they are all concealed in the car. This not only reduces the cost but also enables a neat appearance. However, without a high-definition 3D LIDAR, the perception tasks become more challenging, especially for moving object tracking. In addition, each type of sensor has different range and noise characteristics. To address these problems, we break the moving object tracking system into several context-based subsystems. This method enables us to maximize the performance of each type of sensor for different driving contexts. In this paper, we focus on perception for on-road driving.

Another low-cost, unobtrusive sensing method has been reported in [8], in which a vision-based traffic scene understanding algorithm is presented. By reasoning about visual cues rather than road markings, this approach jointly obtains the geometry model of intersections and tracks the objects in the scene. This method has great potential to work without map. However, the performance of the method, such as topology accuracy and location error, is not good enough for autonomous driving yet.

In summary, the **contributions of this paper** are: 1) Using behavioral context, our algorithm is able to exploit strengths of different sensors as well as reduce computational time; 2) The use of road coordinates enables the ability to track objects on any type of road, including straight and curved roads; 3) The idea of context-based object perception subsystems can be extended to different driving scenarios.

III. System Architecture

A. Behavioral Context

Fig. 2 shows the hierarchical system architecture for the vehicle. The Mission Planning module takes the human-desired destination and the Road Network Definition File (RNDF) as input, and generates the fastest route within the road network. The RNDF defines the connectivity of the road network, such as the width of each lane and the connectivity with other lanes.

Following the policy generated by the mission planner, the Behavior module identifies driving contexts. Driving contexts help the autonomous vehicle to focus on the area of interest. Typical behavioral contexts include lane driving, intersection handling, and achieving a zone pose (e.g. parking lots). Based on the driving contexts, the Behavior module also outputs a sequence of incremental motion goals. A motion goal could be a location within a road lane when performing on-road driving, or a location within a zone when traversing through a zone. In this paper, we are most interested in on-road driving. For on-road driving, the motion goals decide the desired lane and turn at each intersection. In our method, the region of interest is generated based on the motion goals and RNDF.

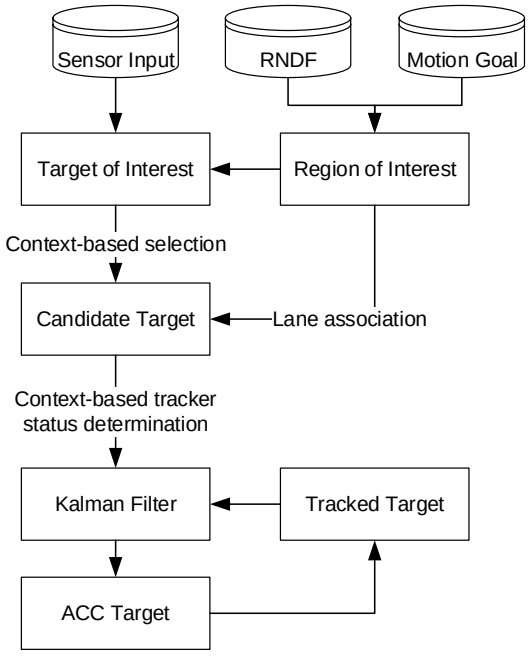


Fig. 5: Distance-keeping object tracking framework

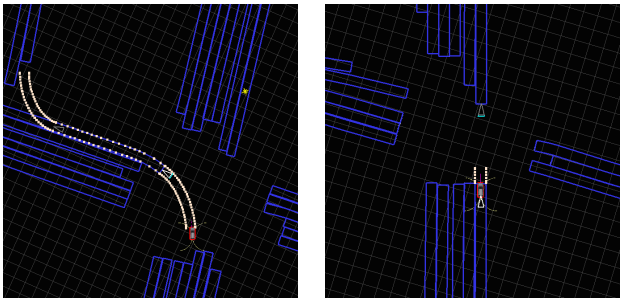
B. Sensor Installation

Fig. 3 shows the sensor installation. There are six RADARs, six LIDARs, and three cameras on the vehicle. One of the cameras is made by MobilEye [9]. All sensors are well concealed in the vehicle. Although perception is more challenging as a result, this sensor configuration option is safer and more practical for mass production compared to other autonomous vehicles which have non-automotive-grade sensors on top of the car.

As shown in Fig. 4, the installed sensors give 360-degree coverage around the car. Objects within 200 meters will be detected by at least one type of sensor, and objects within 60 meters will be detected by at least two types of sensors. A leading vehicle in the same lane as the autonomous vehicle will be covered by all three types of sensors most of the time. In addition, the blind spots are small enough that no vehicle will be neglected.

IV. Scene-Aware Object Tracking

In this paper, we focus on object tracking for distance keeping. Fig. 5 shows the framework for the object tracking system for a distance-keeping target, which is one of the Context-Aware Perception Subsystems. By breaking a universal tracking system into a set of context-aware subsystems, we are able to optimize the performance of each type of sensor. In the first step, we generate the region of interest (ROI) using the RNDF and motion goals. We then gather all measurements from LIDAR, RADAR and camera, excluding measurements outside the ROI. The remaining measurements are marked as targets of interest. We project targets of interest into road coordinates, and get longitudinal distances and lateral distances. The context allows us to assume that if there is a distance-keeping target, it must be the one longitudinally closest to the autonomous vehicle among all targets. For each type of sensor, we apply this context to all targets of interest to select a candidate target. Based on the status of the tracked target and the candidate targets, the tracking status, such as prediction or update, is determined. At the final step, a Kalman Filter [10] is applied to estimate the final state of the distance-keeping target.



(a) Cross an intersection (b) Stop at a stop line

Fig. 6: Region of interest. It is given by the dotted white lines.

A. Region of Interest

In our system, the Behavior module identifies driving contexts and generates a set of corresponding Motion Goals. The region of interest (ROI) is generated using motion goals and RNDF. Fig. 6 shows two examples for the region of interest. Fig. 6a shows the region of interest when the vehicle is crossing an intersection. Our algorithm is able to generate the region of interest along the lane for any type of road. Fig. 6b shows the region of interest when the vehicle is waiting at a stop line for the traffic light. It can intelligently choose the length of the region of interest to avoid paying attention to cross-traffic when waiting at the stop line.

B. Sensor Input

Three types of sensors are used for the moving object tracking system: LIDAR, RADAR, and camera. LIDAR can directly detect ranges to objects. It also has the potential to estimate shapes of objects. However, LIDAR has worse robustness to weather conditions than RADAR. RADAR can directly detect ranges and range rates of objects. However, RADAR outputs are not as accurate as LIDAR for distance information, and they are usually not informative enough to estimate shapes of objects. Among these three types of sensors, cameras are the one most informative but most sensitive to weather conditions. In this paper, we directly use the vehicle detection results from MobilEye, which include position and speed information. From our experience, the MobilEye's vehicle detection results have false negatives, but rarely have false positives.

As discussed above, LIDAR data may be sensitive to rain and dust, which means there are unpredictable false positives from LIDAR in rainy conditions. Moreover, the distance-keeping target is directly sent to the distance keeping controller, so it is very sensitive to false positives. For instance,

if an autonomous vehicle is driving at a high velocity such as 65 miles per hour, and a static object, which is a false positive caused by LIDAR noise, shows up right in front of the vehicle, the vehicle would brake suddenly, which is inappropriate both for safety and comfort reasons. Therefore, in the distance-keeping target tracking subsystem, we do not directly use LIDAR data to update the tracker, but only use them to verify measurements from RADARs and camera.

C. Road Coordinate

Unlike other methods [7], [11], which use global coordinates for tracking, we use road coordinates for tracking. In order to enable tracking in road coordinates, we project all measurements from global to road coordinates, as shown in Fig. 7. The measurement from RADAR and camera has the state vector $[x, y, \theta, v]$, where x and y are the global location of the target, θ is the heading (direction of velocity vector) of the target, and v is the longitudinal velocity of the target. The state vector in road coordinates can be represented as $[s, d, \alpha, v_r]$, respectively the longitudinal distance (also called range), lateral distance, direction, and the longitudinal velocity along the road. For distance keeping, we only care about the range and range rate of the target to the autonomous vehicle, thus we can further simplify the model to two dimensions: $[s, \dot{s}]$, where s is the range and $\dot{s}$ is the range rate. This model allows us to easily tune the covariance matrix for the Kalman Filter.

D. Candidate Target

For distance keeping, we only have to track one target, which is the leading target with the smallest range. Therefore, we can use this context to select a candidate target from each type of sensor (RADAR and camera). It also helps us to reduce the computation time since only one target is tracked. To get the candidate target, we first sort all targets of interest into ascending order by range. In order to remove false positives, we use LIDAR points to verify if a target is valid. Basically, the verification process is operated by checking the number of LIDAR points around a target within a predefined radius. A target is considered valid only if its velocity is high enough or the number of LIDAR points is no fewer than a predefined number. As the range becomes larger, the predefined radius gets bigger, and the desired LIDAR point number reduces. The sorted targets are checked one by one until we find the first valid target.

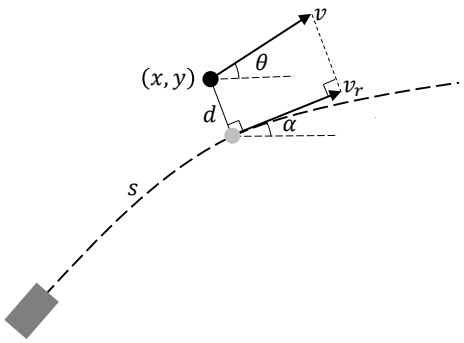


Fig. 7: Project target into road coordinates

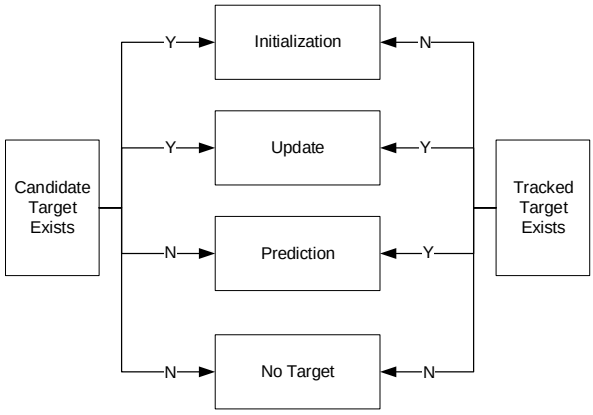


Fig. 8: Tracking status determination

E. Tracking Status

Since only one target is tracked, the tracking status can be determined directly from the status of the last tracked target and the candidate targets. Four tracking statuses are defined: initialization, update, prediction, and no target. If there is no tracked target and there is any candidate target, the tracker will be initialized. If both tracked target and candidate target exist, the tracker will be updated. If there is a tracked target but no candidate target, the prediction will be performed. This logic is summarized in Fig. 8.

F. Kalman Filter

The motion model and measurement model for the Kalman Filter are defined as follows.

$$\mathbf{x}_k = A_k \mathbf{x}_{k-1} + \mathbf{w}_k, \qquad \mathbf{w}_k \sim N(0, W_k) \tag{1}$$

$$\mathbf{z}_k = H_k \mathbf{x}_{k-1} + \mathbf{v}_k, \qquad \mathbf{v}_k \sim N(0, V_k) \tag{2}$$

$$\mathbf{x}_k = \begin{bmatrix} s \\ \dot{s} \end{bmatrix}, \tag{3}$$

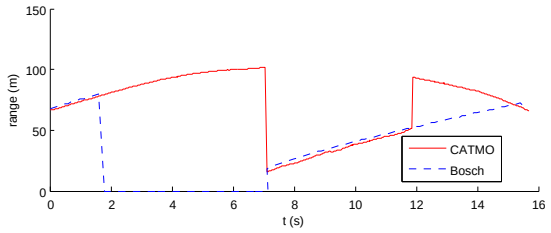
$$A_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{4}$$

$$H_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

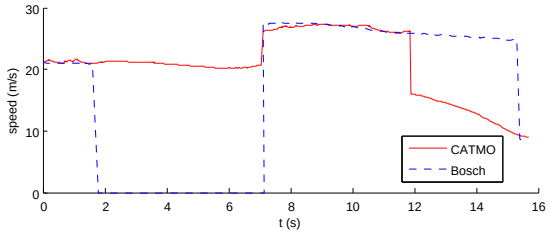
In the motion model, $\mathbf{x}_k$ is the state vector, A_k is the transition matrix, and $\mathbf{w}_k$ is the process noise with zero-mean Gaussian distribution. In the measurement model, $\mathbf{z}_k$ is the measurement vector, H_k is the observation matrix, and $\mathbf{v}_k$ is the measurement noise with zero-mean Gaussian distribution. Δt is the discretization time step. $\mathbf{x}_k$ and $\mathbf{z}_k$ are both composed of range and range rate.

V. Results

We tested the proposed algorithm on CMU's SRX driving autonomously. To verify the performance of the algorithm, three scenarios were used: curved road, stop and go, and cut in and out. For each scenario, we also compared the performance of our algorithm to that used in the Bosch



(a) Range



(b) Range rate

Fig. 9: Case 1: Cut in and out

LRR3 RADAR [12]. The results are composed of the range and range rate during the test, and some snapshots including bird's-eye view tracking results and front-view images. If the algorithm dose not detect any target, the range and range rate are set to 0.

In the snapshots of the tracking results, the white polygons are the region of interest, the yellow arrows indicate the distance-keeping target, the blue number close to the distance-keeping target is the speed (m/s) of the target, the white number near the autonomous vehicle is its speed (m/s), and the green number between the two speeds is the range (m) from the front bumper of the autonomous vehicle to the rear bumper of the distance-keeping target.

A. Cut In and Out

The cut in and out scenario tests the tracking algorithm's responsivity when a vehicle cuts in and out in front of us. Good performance will initialize the tracker for cutting in and clear it for cutting out. The results are shown in Fig. 9 and Fig. 10. As shown in Fig. 9, the Bosch ACC algorithm lost track of the target when the range was greater than $80m$. A truck cut in at around $t = 7s$, and left the lane at around $t = 12s$. After the truck left the lane, the Bosch ACC algorithm still tracked the truck for more than $2s$, while our algorithm switched to track the correct distance-keeping target as soon as the truck changed to another lane. Table I shows the statistical results for the latency of the Bosch ACC in recognizing a cut-in or cut-out relative to CATMO.

TABLE I: Latency of Bosch ACC relative to CATMO

	Average (s)	STD (s)	Cases
Cut in	0.58	0.33	12
Cut out	0.74	0.42	9

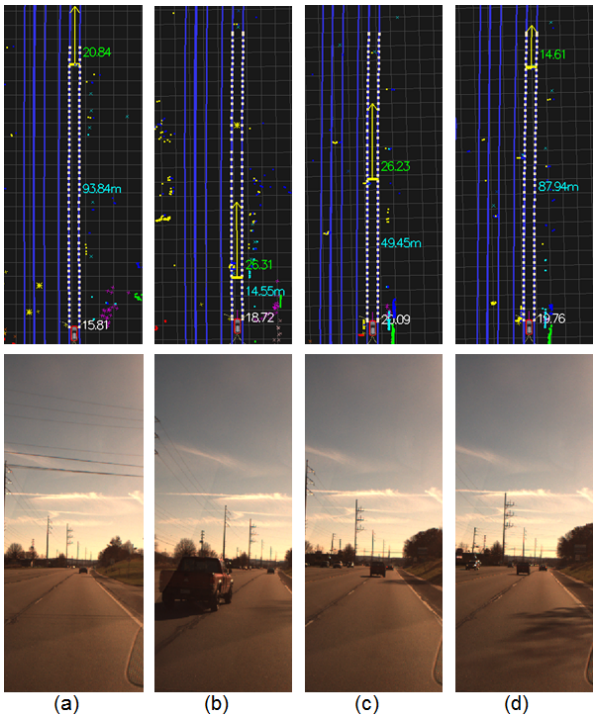


Fig. 10: Case 1: Cut in and out snapshots. (a) Following the leading vehicle which was far away; (b) A truck cutting in; (c) Following the truck; (d) The truck changing to neighboring lane

B. Stop and Go

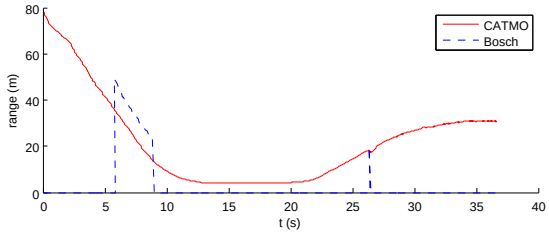
The stop-and-go scenario was designed to test the algorithm's ability to detect static objects, and responsivity when approaching a stopped car and when a stopped car starts moving. The results are shown in Fig. 11 and Fig. 12. Fig. 11 shows that the Bosch ACC algorithm was unable to track a stopped car and unable to follow the car when it started moving, while our algorithm worked well. The Bosch ACC algorithm is a black box to us, so we cannot explain why it was unable to detect the ACC target in the stop-and-go scenario. One possible reason could be that the Bosch ACC algorithm is optimized for targets with high speeds. In our testing, Bosch LRR3 failed 3 times out of 28 cases to find the ACC target when approaching a stopped car and CATMO succeeded every time.

C. Curved Road

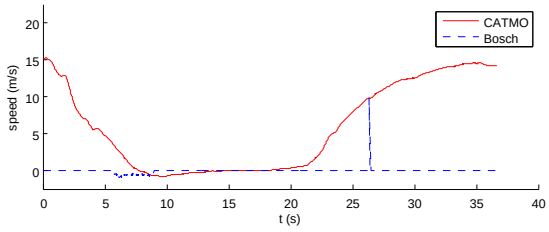
The curved road test was performed at an intersection. This test was designed to verify the tracking algorithm's ability to deal with any shape of road. The results are shown in Fig. 13 and Fig. 14. Fig. 13 shows that our algorithm is able to track the front vehicle in a turn, while the Bosch ACC algorithm lost the tracking of the target after $1.5s$. In our testing, the Bosch algorithm was unable to identify the ACC target on a curved road in 16 cases and CATMO succeeded every time.

VI. Conclusions

We have presented a context-aware object tracking algorithm. In this paper, we focus on selecting the distance-



(a) Range

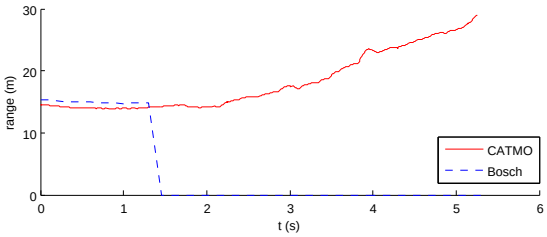


(b) Range rate

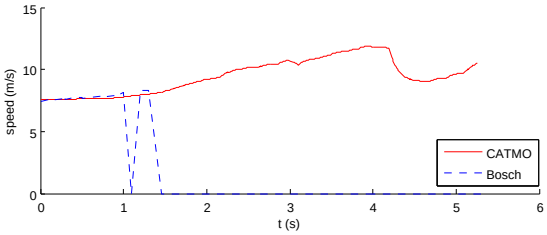
Fig. 11: Case 2: Stop and go

(a) (b) (c)

Fig. 12: Case 2: Stop and go snapshots. (a) Approaching an intersection; (b) Stopped behind a vehicle for red light; (c) Starting moving when the traffic light turned green.

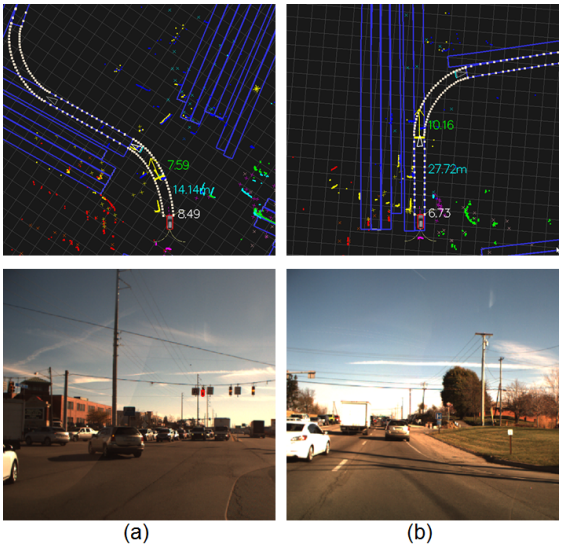


(a) Range



(b) Range rate

Fig. 13: Case 3: Curved road



(a) (b)

Fig. 14: Case 3: Curved road snapshots. Following a leading vehicle at an intersection.

keeping target. The proposed algorithm takes advantage of behavior contexts, which enables maximizing the potential of different sensors. It also helps to reduce the computation time and track targets on any shape of road. We also tested the algorithm in three cases which are difficult for distance keeping. We compared our algorithm with the algorithm used in the Bosch LRR3 RADAR in these three cases. The results showed that our algorithm outperformed Bosch's algorithm in finding the distance-keeping target without losing performance on smoothness. This context-aware algorithm can be extended for object tracking in other scenarios, such as cross-traffic handling and ramp merging.

REFERENCES

[1] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, C. R. Baker, R. E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J. M. Dolan, D. Duggins, D. Ferguson , *et al.*, "Autonomous driving in

urban environments: Boss and the urban challenge," *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, vol. 25, pp. 425–466, June 2008.

[2] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2013.

[3] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, *et al.*, "Making bertha drive? an autonomous journey on a historic route," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 2, pp. 8–20, 2014.

[4] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 143–153, 2003.

[5] C. R. Baker and J. M. Dolan, "Traffic interaction in the urban challenge: Putting boss on its best behavior," in *International Conference on Intelligent Robots and Systems (IROS 2008)*, pp. 1752–1758, 2008.

[6] J. Wei, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2014.

[7] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 475–485, 2009.

[8] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[9] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," in *SAE World Congress & Exhibition, Detroit, USA*, 2005.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[11] C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppé, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe, *et al.*, "Moving object detection with laser scanners," *Journal of Field Robotics*, vol. 30, no. 1, pp. 17–43, 2013.

[12] D. Freundt and B. Lucas, "LRR3 by Bosch–long range radar sensor for high-volume driver assistance systems market," in *SAE World Congress & Exhibition, Detroit, USA*, 2008.