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

Multi-Target Track-to-Track Fusion Based on Permutation Matrix Track Association

Kuan-Hui Lee[1] Yusuke Kanzawa[1], Matthew Derry[1], Michael R. James[†]

***Abstract*— This paper proposes the Permutation Matrix Track Association (PMTA) algorithm to support track-to-track, multi-sensor data fusion for multiple targets in an autonomous driving system. In this system, measurement data from different sensor modalities (LIDAR, radar, and vision) is processed by object trackers operating on each sensor modality independently to create the tracks of the objects. The proposed approach fuses the object track lists from each tracker, first by associating the tracks within each track list, followed by a state estimation (filtering) step. The eventual output is the unified tracks of the objects provided for further autonomous driving processing, such as path and motion planning. The permutation matrix track association (PMTA) algorithm considers both spatial and temporal information to associate object tracks from different sensor modalities. Experimental results show that the proposed approach improves not only the performance of the multiple-target track-to-track fusion, but also stability and robustness in the resulting speed control and decision making in the autonomous driving system.**

I. INTRODUCTION

Recently, autonomous driving has attracted significant academic, commercial, and popular interest. Since the 2005 DARPA Grand Challenge and the 2007 Urban Challenge, researchers have developed several successful technologies for autonomous driving applications. An often stated goal for developing the autonomous driving technologies is to improve driving safety, both to people inside and outside of the autonomous vehicle. To achieve this goal, the perception subsystem becomes a critical, first-line module in any highly autonomous driving architecture.

The perception subsystem plays an important role in an autonomous driving system, not only detecting objects/obstacles to avoid collisions and accidents, but also tracking object motion to make predictions about the intentions and trajectories of other agents within proximity of the vehicle. Typically, the perception subsystem includes various types of sensors, and different sensors have different detection/tracking approaches according to the characteristics of the sensors. The advantage of using multiple sensor modalities is to utilize the strengths of each sensing modality to compensate for the deficiencies in the other sensor modalities. This provides a robust perception system with a wider and longer perceptual field. Therefore, object detection/tracking based on multi-sensor fusion is indispensable for any robust autonomous driving system. Sensor fusion can take many different forms, depending on where in the perception pipeline that data fusion occurs [1]:

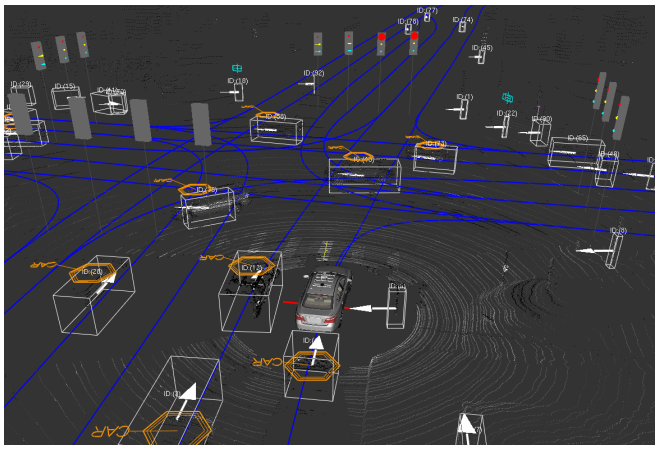

Fig. 1. An example of the view seen by the robot, through the perception subsystem. Each white bounding box is a tracked target, which are unified by several tracked objects detected by multiple sensors.

1) Raw/Low-level fusion: The fusion task registers raw data directly obtained from the sensors in to a common reference frame, and the fused raw data is then given as input to a detection and a tracking algorithm. The advantage of the low-level fusion is to possibly classify data at a early stage of the system from different sensors, thereby providing a more complete picture of the object state to a given detection algorithm. This approach can be particularly sensitive to correct sensor calibration and timing.

2) Feature/Middle-level fusion: Instead of directly using raw data from different sensors, middle-level fusion extracts certain features, such as bounding box from a vision-based object detector, from raw data through a pre-processing step. Then, the tracking procedure is processed by fusing the extracted features, where the tracking approaches are traditional filtering or occupancy grid based approaches.

3) Track/High-level fusion: Each object is detected and tracked by a specific (type of) tracker, and each tracker provides a track of the object. Then, a track-to-track fusion, usually along with filtering, is applied to unify the the associated objects' tracks.

In this paper, we focus on multi-target track-to-track fusion with LIDAR, radar, and vision. Figure 1 shows an example of what our robot sees through the perception subsystem, where the white bounding boxes are the tracked targets detected by multiple sensors. We assume the objects are detected and tracked by a multi-target tracker for each sensor modality, and treat the problem as a track association task for multiple

[1] Kuan-Hui Lee, Yusuke Kanzawa, and Matthew Derry are with Toyota Research Institute, USA, kuan.lee@tri.global

[†] Michael R. James is with Waymo; this work had been done when Michael R. James was in Toyota Research Institute.

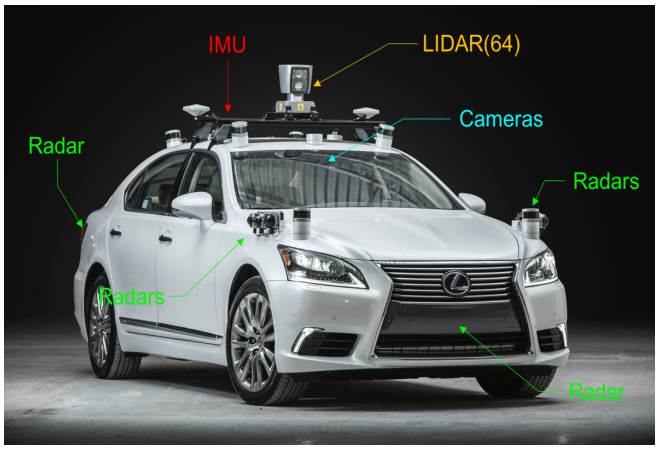

Fig. 2. Hardware and sensor configuration of the test vehicle.

targets. There are two key contributions from this work:

a) A novel track association algorithm, Permutation Matrix Track Association, in which we apply the idea of a permutation matrix to the multi-target, multi-sensor fusion scenario, considering both spatial and temporal information. The advantage of PMTA is to constrain the problem to one-by-one association between different tracks, and thus reduce ambiguity of the track association between the multiple sensors.

b) Experimental results both comparing the proposed track association algorithm with other well known data association algorithms and demonstrating the overall effect on autonomous driving performance.

The rest of the paper is organized as follows. Section II gives a brief survey on the related work. In Section III, the overview of the multi-sensor perception in our autonomous driving system is briefly described. Section IV depicts the proposed permutation matrix track association algorithm for multi-target track-to-track fusion. The experimental results are shown in Section V, followed by the conclusion in Section VI.

II. RELATED WORK

Thanks to the growing interest in autonomous driving, a considerable amount of related work has been published [2] [3] [4] [5]. As one of the core processes, the perception subsystem is requested to robustly detect and track surrounding objects (e.g., car, pedestrian, bicyclist, obstacle, etc) [6] [7] [8] [9]. Multi-sensor fusion is one important way to improve detection/tracking performance. Here we only summarize the earlier works that are relevant to multi-sensor fusion for object detection and tracking.

Typical tracking approaches utilizing multi-sensor fusion have been studied extensively [10] [11], and the fusion with LIDARs and radars are widely exploited in the 2007 DARPA Urban Challenge for practical applications [3] [4] [7]. Stiller et. al [12] started to use LIDAR, radar, and stereo vision for obstacle detection and tracking. This work inspired other researchers to develop the approach of fusing LIDAR sensors with vision sensors for pedestrian and vehicle tracking. Mählisch et al. [13] proposed a cross-calibration approach between a 2-D LIDAR and a monocular camera to track vehicles. Premebida et al. [14] used a 3-D LIDAR and a monocular camera for pedestrian detection. Vu et al. [15] proposed a fusion task with a 2-D LIDAR, a radar, and a monocular camera for detecting, tracking, and classifying object simultaneously. Cho et al. [16] fused several LIDARs, several radars, and a camera, to improve the performance of movement classification and tracking model selection. These raw-level and middle-level fusion approaches have been successfully used in many autonomous applications.

Recently, track-level fusion between multiple sensors is receiving increased attention [1] [17] [18]. The main advantage is to not only have flexibility and scalability but also abstract the implementations and the details of the sensor-specific processing pipelines. Aeberhard et la. [17] proposed a track-to-track fusion method by using the Information Matrix Filter (IMF) with asynchronous data. Li et al. [18] developed a mixture of the IMF and the Split Covariance Intersection Filter (SCIF), where IMF is used for track temporal correlation and the SCIF is used for spatial correlation. Their approaches mainly focused on fusing multi-sensor data for single target tracking, where the sensor data belonging to the target is pre-associated. As for multi-targets, most approaches apply well-known data association methods (such as Nearest Neighbor (NN), Joint Probabilistic Data Association (JPDA), Multiple Hypothesis Tracking (MHT), and Markov Chain Monte Carlo Data Association (MCMCDA), etc [19] [10]), followed by the state/noise estimation with a filter (such as a variant of Kalman filter, particle filer, etc). Möbus et al. [20] proposed a multi-target, multi-sensor fusion of radar and infrared by using Probabilistic Data Association (PDA). Houenou et al. [21] adopted NN along with a naïve independent Kalman filter to fuse multiple sensors for tracking multiple targets. Such a track-to-track fusion problem is highly similar to the problem of feature points matching between images [22] [23], which adopted a binary permutation matrix to represent the correspondence between the feature points in different images. Inspired by their work, we apply the concept to the scenario of track-to-track, multi-sensor data association. Such a framework restricts one-by-one association between different tracks, and thus reduce ambiguity of the track association between the multiple sensors.

III. MULTI-SENSOR PERCEPTION

The robot is a test vehicle including software modules for perception, localization, path planning, and motion control, as shown in Figure 2. The test vehicle is equipped with: an Applanix POS LV 220 (fused GPS + IMU), 360 degree, 64-beam LIDAR (Velodyne HDL-64e), multiple millimeter-wave radars, and multiple cameras. The test vehicle utilizes a detailed digital map containing a high-quality 3D road network with positions for each lane (center). When driving autonomously, the test vehicle estimates its precise 6-DOF pose on the map by correcting output from Applanix-based localization system. The perception subsystem (LIDAR, radars, and cameras) detects and tracks surrounding objects,

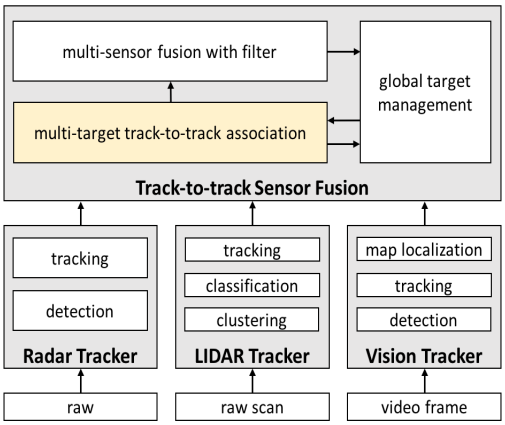

Fig. 3. The overview of the proposed perception subsystem.

and provides the tracks of the objects for use in the decision making process.

Figure 3 shows the overview of the proposed multi-sensor perception subsystem in the experimental autonomous driving system. Each type (LIDAR, radars, and vision) of raw sensor data is pre-processed by a sensor-specific, multi-object tracker. The LIDAR tracker utilizes clustering and classification algorithms to discriminate surrounding objects, and then adopts a 3-D point tracking algorithm with a Bayesian filter to track the objects. The radar tracker detects and tracks the objects, providing 2-D position and range rate of each object. The vision tracker detects the objects in the video frame and then adopts the tracking-by-detection algorithm [24] to track the objects in image space. The positions in image space are further map-localized to 3-D space by referencing the digital map. By taking the multiple tracks as input, a track-to-track sensor fusion mechanism is applied to unify the tracks from different trackers. The track-to-track sensor fusion includes two stages: multi-target track-to-track association and state estimation via Bayesian filtering. The first stage is to associate the tracks of the multiple objects provided from different trackers, by the proposed permutation matrix track association. Then, the second stage is to fuse the associated objects' data within a Bayesian filtering framework, for each target. Finally a global target management stage is used to manage the identification of the objects and the associations of their tracks. The output of the perception subsystem is the fused tracks of the objects, which are used for further planning and control.

IV. PERMUTATION MATRIX TRACK ASSOCIATION

A. Two Sensors Scenario

Assume we have two sets of observations, $\mathbf{X}_1$ and $\mathbf{X}_2$, representing two observations from sensor s_1 and s_2, respectively: $\mathbf{X}_1 = [\mathbf{x}_{11} \ ... \ \mathbf{x}_{1M}]$, $\mathbf{X}_2 = [\mathbf{x}_{21} \ ... \ \mathbf{x}_{2N}]$, where $\mathbf{x}_{1i}$ is the i^{th} observation from the sensor s_1, $\mathbf{x}_{2j}$ is the j^{th} observation from the sensor s_2, and M and N are the numbers of the tracks in s_1 and s_2 at a certain timestamp. The observation contains the general information of the track, for example, the position and the velocity of the object. Hence, to determinate the correspondence between s_1 and s_2, the goal is to find an $(M+1) \times (N+1)$ binary permutation matrix $\hat{\mathbf{P}}$, such that the entry $\mathbf{P}_{ij}$ in $\mathbf{P}$ is 1 if $\mathbf{x}_{1i}$ corresponds to $\mathbf{x}_{2j}$; otherwise, it is 0. In the matrix $\mathbf{P}$, each row and column indicates an observation from s_1 and s_2 respectively, and the $(M+1)^{th}$ row and the $(N+1)^{th}$ column represent the mismatched correspondence. Therefore, the problem can be formulated as a constrained minimization integer programming problem:

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} J(\mathbf{P}, \mathbf{X}_1, \mathbf{X}_2), \tag{1}$$

$$\text{s.t.} \quad \mathbf{P}_{ij} \in \{0,1\} \quad \forall i \leq M+1, \forall j \leq N+1, \tag{2}$$

$$\sum_{i=1}^{M+1} \mathbf{P}_{ij} = 1 \ \forall j \leq N, \quad \sum_{j=1}^{N+1} \mathbf{P}_{ij} = 1 \ \forall i \leq M, \tag{3}$$

where J is the objective cost function to be minimized. The constraint equations (2) and (3) restrict the one-to-one association between the observations. By incorporating soft-assignment [22] instead of a hard decision, the problem can be relaxed by replacing constraint (2) with:

$$\mathbf{P}_{ij} \geq 0 \quad \forall i \leq M+1, \forall j \leq N+1. \tag{4}$$

Thus, the entry $\mathbf{P}_{ij}$ can be treated as a likelihood of the i^{th} observation from s_1 matching the j^{th} observation from s_2. By applying deterministic annealing [22], Such a relaxation not only avoids being trapped in a local minima during optimization, but also improves the convergence to the original integer problem.

1) Spatial Cost: The spatial cost considers general spatial information of an object, i.e., position, velocity, and heading, which are included in both $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$. Assume d_p, d_v, and d_ϕ are the measurement distances between $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$, with respect to the position, velocity and heading, the association likelihood is defined by:

$$\ell_s(\mathbf{x}_{1i}, \mathbf{x}_{2j}) = K(d_p, \sigma_p^2) \cdot K(d_v, \sigma_v^2) \cdot K(d_\phi, \sigma_\phi^2), \tag{5}$$

where $K(\bullet)$ is a radial basis function (RBF) kernel; σ_p^2, σ_v^2, and σ_ϕ^2 are the predefined variance of each RBF kernel respectively. Based on the likelihood, we can further apply maximum likelihood estimation to recover the confidence of the association:

$$J_{spatial} = -\sum_{i=1}^{M}\sum_{j=1}^{N} \mathbf{P}_{ij} log(\ell_s(\mathbf{x}_{1i}, \mathbf{x}_{2j})). \tag{6}$$

2) Temporal Cost: In the track-to-track scenario, temporal information is usually known because of the tracks of the objects. Therefore, we assume the association follows the Markov property, which implies that the current association is only affected by the previous association. The association likelihood of the $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$ at the previous timestamp, $\mathbf{P}'_{ij}$, is considered in the temporal cost:

$$J_{temporal} = -\sum_{i=1}^{M}\sum_{j=1}^{N} \mathbf{P}_{ij} log(\mathbf{P}'_{ij}). \tag{7}$$

Note that $\mathbf{P}_{ij}$ in Eq.(7) is the track association at the current timestamp for consistency.

3) Mismatched Cost: If $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$ are not associated, an additional term is considered to penalize the mismatched association. Instead of increasing the cost in the $(M+1)^{th}$ row and $(N+1)^{th}$ column of $\mathbf{P}$, i.e., mismatched entries, we reduce the cost of $\mathbf{P}_{ij}$ for $1 \leq i \leq M$ and $1 \leq j \leq N$, i.e., matched entries:

$$J_{mismatched} = -\beta \sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{P}_{ij}. \tag{8}$$

where β is a predefined factor for controlling the penalty cost when $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$ are mismatched.

4) Entropy Cost: To optimize the problem using deterministic annealing [22], an iterative scheme is widely used to stress "uncertainty", as well as the entropy of the entries in $\mathbf{P}$:

$$J_{entropy} = \gamma \sum_{i=1}^{M+1} \sum_{j=1}^{N+1} \mathbf{P}_{ij} log(\mathbf{P}_{ij}). \tag{9}$$

In the early iterations, the factor γ is initialized with a higher value, in order to highly emphasize the impact of this cost function with respect to the objective function $J(\mathbf{P}, \mathbf{X}_1, \mathbf{X}_2)$. The purpose is to flexibilize the value $\mathbf{P}_{ij}$ in the space for searching the optimum. γ is then gradually decreased over the iterations, in order to restrict the value of $\mathbf{P}_{ij}$ for improved convergence. By incorporating the maximum entropy principle, the entropy cost can also be regarded as a barrier function for the inequality constraints in Eq.(4)

5) Objective Cost Function: The objective cost function $J(\mathbf{P}, \mathbf{X}_1, \mathbf{X}_2)$ is then the sum of the above cost functions:

$$\begin{aligned} J(\mathbf{P}, \mathbf{X}_1, \mathbf{X}_2) = J_{spatial} + J_{temporal} + \\ J_{mismatched} + J_{entropy}. \end{aligned} \tag{10}$$

This problem can be solved by deterministic annealing, and the constraints (3) can be satisfied by the alternative row-column normalization based on the Sinkhorn's theorem [22]. Finally, $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$ are matched as long as $\mathbf{P}_{ij}$ is larger than a threshold α.

B. Multiple Sensors Scenario

As shown in Figure 3, the proposed perception subsystem has three different types of the sensors. To associate three sensors s_1, s_2 and s_3, we separately apply the PMTA to each pair of the sensors, i.e., s_1 and s_2, s_2 and s_3, s_1 and s_3. However, due to sensor noise and measurement error, there can still exist ambiguities in the association after performing PMTA. For example, $\mathbf{x}_{11}$ is associated with $\mathbf{x}_{21}$, $\mathbf{x}_{11}$ is associated with $\mathbf{x}_{31}$, but $\mathbf{x}_{21}$ is not associated with $\mathbf{x}_{31}$; in this case, we select the association with larger $\mathbf{P}_{ij}$. Another example is that $\mathbf{x}_{11}$ is associated with $\mathbf{x}_{21}$ at timestamp t but associated with $\mathbf{x}_{22}$ at timestamp $t+1$ due to misleading tracking results. Such spatial and temporal ambiguities are heuristically handled by the global target management unit.

V. MULTI-SENSOR FUSION WITH FILTER

After PMTA, the multi-sensor fusion with filter is applied to each target corresponding to multiple associated objects. Assume ς_k is the covariance matrix of the s_k, $k \in \{1, 2, ...\}$, the final state of the target $\mathbf{x}_f$ is summarized by the states of the objects $\mathbf{x}_k$. Finally, an Unscented Kalman Filter (UKF) is applied to $\mathbf{x}_f$ for state/noise estimation of the target. For convenience, we describe several multi-sensor fusion methods in terms of two sensors in this section.

1) Naïve Information Matrix Filter: The basic idea of NIMF is to simply fuse the states according to their corresponding covariance matrices ς_*, without considering the correlation between the sensors:

$$\begin{aligned} \varsigma_f^{-1} &= \varsigma_1^{-1} + \varsigma_2^{-1}, \\ \varsigma_f^{-1}\mathbf{x}_f &= \varsigma_1^{-1}\mathbf{x}_1 + \varsigma_2^{-1}\mathbf{x}_2. \end{aligned} \tag{11}$$

2) Generalized Information Matrix Filter: The GIMF is a generalization of the information matrix filter for asynchronous tracks [25]. Assume s_1 is collocated with fusion center, and s_2 has certain delay relative to s_1. The fused track at t_f is given by:

$$\begin{aligned} \varsigma_f^{-1} &= \varsigma_1^{-1} + [\varsigma_2^{-1} - \varsigma_2'^{-1}], \\ \varsigma_f^{-1}\mathbf{x}_f &= \varsigma_1^{-1}\mathbf{x}_1 + [\varsigma_2^{-1}\mathbf{x}_2 - \varsigma_2'^{-1}\mathbf{x}_2'], \end{aligned} \tag{12}$$

where $\{\mathbf{x}_2', \varsigma_2'\}$ is the fused state of the last fused state and $\mathbf{x}_2$ when being observed. The strategy is to pre-fuse $\mathbf{x}_2$ and the fused state, and then fuse all the states until next $\mathbf{x}_1$ is observed.

3) Covariance Intersection: The Covariance Intersection (CI) [26] involves weights to deal with the fusion problem of two Gaussian estimates. The fusion task can be given by:

$$\begin{aligned} \varsigma_f^{-1} &= \omega\varsigma_1^{-1} + (1-\omega)\varsigma_2^{-1}, \\ \varsigma_f^{-1}\mathbf{x}_f &= \omega\varsigma_1^{-1}\mathbf{x}_1 + (1-\omega)\varsigma_2^{-1}\mathbf{x}_2, \\ \omega &= \arg\min_{\omega} \det(\varsigma_f). \end{aligned} \tag{13}$$

By minimizing the determinant of ς_f with respect to ω, the solution is able to obtain a consistent estimate of the covariance matrix.

4) Sequential-Sensor: The sequential-sensor (SS) method [11] [27] is to consider each observation as an independent, sequential update to the state/noise estimate. Different from the group-sensor approaches which fuse several observations at the same time, the sequential-sensor approach fuses the observations one by one, by sequentially feeding to a filter's estimation process (UKF in the proposed system).

VI. EXPERIMENTS

The perception subsystem runs on a computing cluster with Titan X GPU, including a LIDAR-based tracker operating at 10Hz, a radar-based tracker operating at 100 Hz and a vision-based tracker operating at 20Hz. To evaluate the performance of the proposed approach, we drove the test vehicle and collected data over about 50-minutes of driving. The routes contain highways and surface roads with many intersections, including roundabouts, in Ann Arbor, Michigan.

TABLE I

QUANTITATIVE EVALUATION OF THE TRACK-TO-TRACK ASSOCIATION

	NN (baseline)	JPDA	MCMCDA	**PMTA** (proposed)
precision	95.48%	96.86% (+1.38%)	96.52% (+1.04%)	97.80% (+2.32%)
recall	85.60%	85.21% (-0.39%)	84.92% (-0.68%)	91.11% (+5.51%)
accuracy	98.06%	98.15% (+0.09%)	98.09% (+0.03%)	98.84% (+0.78%)

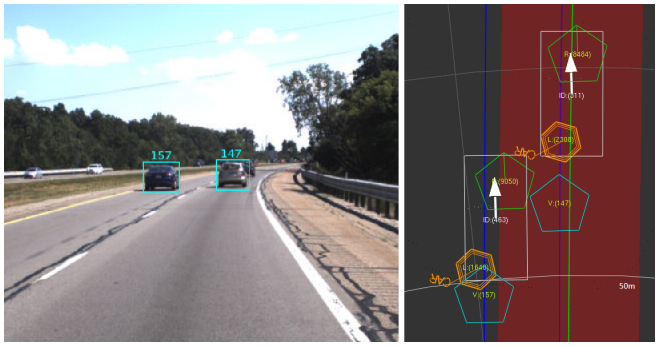

Fig. 4. An example of spatial ambiguity. The orange hexagons indicate the objects from LIDAR, denoted as L:(id); the green pentagons indicate the objects from radars, denoted as R:(id); the blue pentagons indicate the objects from vision, denoted as V:(id). The white rectangles indicate the unified targets, where ID:(311) contains L:(2308), R:(8484) and V:(147), and ID:(463) contains L:(1640), R:(9050) and V:(157).

Our benchmark has 70+ objects, including dynamic objects (vehicles, pedestrians, and bicyclist) and static objects (vegetation, fence, obstacles, etc), with 27421 association timestamps. Moreover, to demonstrate the effect of the approach on autonomous driving, we tested the approach with the robot in autonomous driving mode. Figure 6 shows some tracking results during operating in autonomous mode.

A. Track-to-Track Association

The performance of the track-to-track association is evaluated in terms of precision, recall and accuracy, where the 25000+ sets of association ground truth were manually labeled from the collected data. We compared the proposed PMTA with other data association methods: NN, JPDA, and MCMCDA. The distance measurement for all the approaches (i.e., d_p, d_v, and d_ϕ) are the Euclidean distance. The NN is treated as baseline, by selecting the nearest neighbor within a certain range, i.e., $d_p \leq 5$, $d_v \leq 6$, and $d_\phi \leq 0.05$. Both JPDA and MCMCDA have detection probability as 0.95 and false alarm probability as 0.1. Moreover, the MCMCDA has 10000 samples for the Monte Carlo method. As for the PMTA, we chose $\alpha = 0.7$, $\beta = 1.4$, $\sigma_p^2 = 5$, $\sigma_v^2 = 6$, and $\sigma_\phi^2 = 0.05$; γ is initialized by 0.001 and multiplied by 1.2 for every iterations.

The quantitative results of the track-to-track association are shown in Table I. The NN approach achieved 95% precision and 98% accuracy, which implies that NN performs well in general cases. The JPDA and the MCMCDA can improve precision and accuracy slightly, but perform worse in recall; this means that the number of the false negative increases due to ambiguous associations. As for the PMTA, the performance of the recall increases about 5%, which shows the proposed approach efficiently reduces the spatial and temporal ambiguities. Most cases can be handled well by the algorithms, but the PMTA can further deal with certain cases of association ambiguity. Figure 4 shows an example of the spatial ambiguity, where the R(9050) and the V(147) are much closer to the L(2308) rather than L(1640); but both R(9050) and V(147) are actually associated with the L(1640). Both the JPDA and the MCMCDA partially fail in this case, but the PMTA can associate tracks successfully.

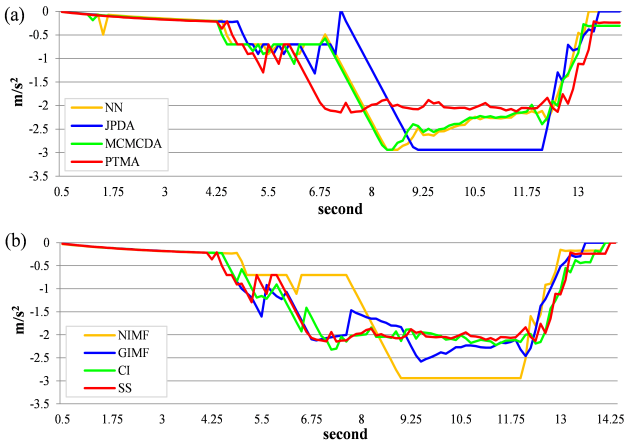

Fig. 5. The performance impact of speed controlling with different track association approaches and filtering methods. (a) Comparison of NN, JPDA, MCMCDA and PMTA with SS filtering. (b) Comparison of NIMF, GIMF, CI and SS with PMTA.

B. Speed Controlling

To evaluate the performance impact of the track association approaches on speed controlling, we compared the track association and filtering methods under a test scenario: our robot is driving toward a stopped vehicle (100m far away) with gradually decreased speed until stopping behind the vehicle. Figure 5 shows the acceleration comparison of different track association methods as well as comparing the different filtering methods. As shown in the figure, all compared approaches have a hard brake around 8 seconds, which implies that the brake is caused by ID switches due to ambiguous track associations. On the contrary, the proposed approach provides relative smooth braking. Figure 5 (b) shows the comparison of different filtering along with PMTA. The NIMF has worst performance, while CI and SS have comparable results of performing smooth deceleration. This implies asynchronous filtering can achieve better performance than synchronous ones.

VII. CONCLUSIONS

In this paper we propose an efficient track-to-track association approach for multi-target, multi-sensor track fusion. The proposed PMTA improves not only the performance of the multiple targets' track-to-track fusion but also stability and robustness in subsequent speed controlling in the autonomous driving system. In the future, such a permutation matrix framework can be extended to multi-dimension

Fig. 6. Some tracking results during autonomous driving; the top row shows the views of the robot through the perception system, the bottom row shows the corresponding camera views. (a) Tracking of vehicles and a bicyclist. (b) Tracking results while staying in an intersection. (c) Tracking of a vehicle in long distance. (d) Tracking results while driving up to a roundabout.

applications, so as to represent variant scenarios of data association.

REFERENCES

[1] M. Aeberhard and N. Kaempchen. High-level sensor data fusion architecture for vehicle surround environment perception. In *Proceedings of 8th International Workshop on Intelligent Transportation*, 2011.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M.N. Clark, J. Dolan, D. Duggins, T. Galatali, and C. Geyer. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[3] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, and S. Karaman. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008.

[4] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, and B. Huhnke. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[5] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, and N. Zych. Team cornell's skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, 25(8):493–527, 2008.

[6] M. Darms, P. Rybski, C.R. Baker, and C. Urmson. Obstacle detection and tracking for the urban challenge. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):475–485, September 2009.

[7] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, et al. Towards fully autonomous driving: systems and algorithms. In *IEEE Intelligent Vehicles Symposium*, pages 163–168, June 2011.

[8] M. Delp, N. Nagasaka, N. Kamata, and M. R. James. Classifying and passing 3d obstacles for autonomous driving. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1240–1247, November 2015.

[9] D. Held, D. Guillory, B. Rebsamen, S. Thrun, and S. Savarese. A probabilistic framework for real-time 3d segmentation using spatial, temporal, and semantic cues. In *Robotics: Science and Systems*, June 2016.

[10] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Journal of Information Fusion*, 14(1):28–44, 2013.

[11] H. Durrant-Whyte and T. C. Henderson. Multisensor data fusion. In *Springer Handbook of Robotics*, pages 867–896. 2016.

[12] C. Stiller, J. Hipp, C. Rössig, and A. Ewald. Multisensor obstacle detection and tracking. *Journal of Image and Vision Computing*, 18(5):389–396, 2000.

[13] M. Mählisch, R. Schweiger, W. Ritter, and K. Dietmayer. Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In *IEEE Intelligent Vehicles Symposium*, pages 424–429, June 2006.

[14] C. Premebida, O. Ludwig, and U. Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9):696–711, 2009.

[15] T.-D. Vu, O. Aycard, and F. Tango. Object perception for intelligent vehicle applications: A multi-sensor fusion approach. In *IEEE Intelligent Vehicles Symposium*, pages 774–780, June 2014.

[16] H. Cho, Y.-W. Seo, B. V. K. Vijaya Kumar, and R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *IEEE International Conference on Robotics and Automation*, May 2016.

[17] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram. Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception. *IEEE Transaction on Intelligent Transportation Systems*, 13(4):1717–1726, 2012.

[18] H. Li, F. Nashashibi, B. Lefaudeux, and E. Pollard. Track-to-track fusion using split covariance intersection filter-information matrix filter (scif-imf) for vehicle surrounding environment perception. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1430–1435. IEEE, 2013.

[19] J. Liu, M. Chu, and J. E. Reich. Multitarget tracking in distributed sensor networks. *IEEE Signal Processing Magazine*, 24(3):36–46, 2007.

[20] R. Möbus and U. Kolbe. Multi-target multi-object tracking, sensor fusion of radar and infrared. In *IEEE Intell. Veh. Symp.*, pages 732–737, 2004.

[21] A. Houenou, P. Bonnifait, V. Cherfaoui, and J.-F. Boissou. A track-to-track association method for automotive perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 704–710, 2012.

[22] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Journal of Computer Vision and Image Understanding*, 89(2):114–141, 2003.

[23] C.-T. Chu, J.-N. Hwang, J.-Y. Yu, and K.-Z. Lee. Tracking across nonoverlapping cameras based on the unsupervised learning of camera link models. In *IEEE/ACM International Conference on Distributed Smart Cameras*, 2012.

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

[25] X. Tian and Y. Bar-Shalom. On algorithms for asynchronous track-to-track fusion. In *IEEE International Conference on Information Fusion*, 2010.

[26] S. Julier and J. K. Uhlmann. General decentralized data fusion with covariance intersection (ci). In *Multisensor Data Fusion*. CRC Press, 2001.

[27] X. Yang, W.-A. Zhang, M. Z. Q. Chen, and L. Yu. Hybrid sequential fusion estimation for asynchronous sensor network-based target tracking. *IEEE Transaction on Control System Technology*, 25(2):669–676, 2017.