MOT subsystem in Auto Drive

Anonymous CVPR submission

Paper ID 11910501 11910934

Abstract

This survey investigates the MOT, which is moving object tracking, subsystem in the architecture of auto driving system. We first present the overview of an auto driving system, including perception system and decision system. Then, we focus on the MOT subsystem, introducing some common implementation techniques. We end this introduction with some problems we found in these techniques, and introduce collaborative perception to solve them and improve the MOT subsystem.

1. Introduction

Since the mid-1980s, many universities, research centers, automotive companies, and companies in other industries around the world have been researching and developing self-driving cars (also known as autonomous and driverless cars) [1] [2] [3]. Important examples of self-driving vehicle research platforms in the last two decades are Navlab's mobile platform, the University of Pavia and the University of Parma's car, ARGO, and UBM's vehicles, VaMoRs and VaMP. To stimulate the development of self-driving car technology, the Defense Advanced Research Projects Agency (DARPA) has organized three competitions in the past decade [6] [4] [16] [15].

In the first competition, self-driving cars are required to drive a 142-mile route across a desert trail in a 10-hour period. All participating cars failed within the first few miles. In the second competition, self-driving cars were required to drive a 132-mile route through flatlands, dry lake beds and mountain passes, including three narrow tunnels and more than 100 or so sharp turns. The competition had 23 finalists, and four cars completed the route in the allotted time [14] [13]. In the third competition, self-driving cars were required to drive a 60-mile route in a simulated urban environment with other self-driving and human-driven cars over a six-hour period. The cars had to obey California traffic rules. There were 11 finalists in this competition and 6 cars completed the race in the allotted time [12].

To measure the level of autonomy of self-driving cars, the Society of Automotive Engineers International (SAE) has published a classification system based on the level of human driver intervention and attention they require, where the level of autonomy of self-driving cars can range from Level 0 (where the car's autonomous system issues a warning and may intervene momentarily, but there is no continuous control of the car) to Level 5 (where no human intervention is required in any situation) [5] [7].

Through the history of this famous self-driving car competition, we can see that self-driving techniques have evolved quickly in the past several decades. The typical architecture of self-driving cars were also decided and stabilized during the research [8]. The architecture of autonomous systems for self-driving vehicles is usually divided into two main parts: the perception system and the decision system. The perception system is typically divided into many subsystems responsible for tasks such as autonomous vehicle localization, static obstacle mapping, road mapping, moving obstacle detection and tracking, and traffic signalization detection and recognition. The decision system is also typically divided into many subsystems responsible for tasks such as route planning, path planning, behavior selection, motion planning, obstacle avoidance, and control [9].

The rest of this paper is structured as follows. In Section 2, we present the typical structure of an autonomous system for self-driving cars, consisting of the decision system and their subsystems. In Section 3, we introduce common techniques for implementing MOT subsystems, presenting the problems of the current mainstream implementation approaches. In Section 4, we introduce the newly emerged cooperative perception technology, aiming to use this new technology to solve the problems we identified.

2. Typical architecture of self-driving cars

In this section, we present a typical architecture of an automated system for self-driving cars, consisting of the decision system and their subsystems.

Fig. 1 shows a block diagram of a typical architecture of an automated system for a self-driving car, where the perception and decision system is shown as a collection of

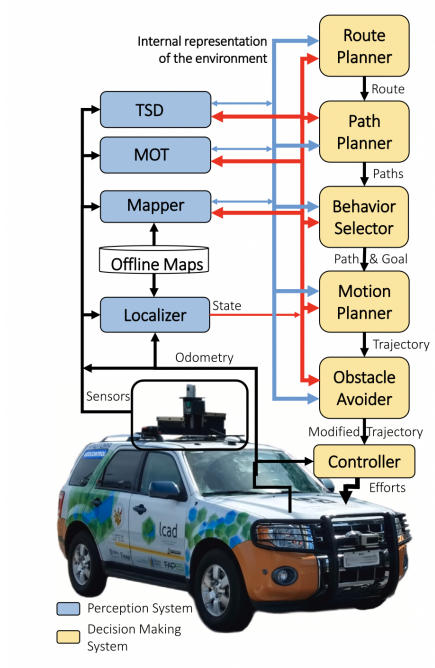

Figure 1. Overview of the typical architecture of the automation system of self-driving cars. TSD denotes Traffic Signalization Detection and MOT denotes Moving Objects Tracking

subsystems in different colors. The perception system is responsible for the estimation of the state of the car and the creation of an internal (to the autonomous driving system) representation of the environment using data captured by on-board sensors such as Light Detection and Ranging (LIDAR), Radio Detection and Ranging (RADAR), cameras, Global Positioning System (GPS), Inertial Measurement Unit (IMU), odometer, etc., as well as models about the sensors, road network, the traffic rules, vehicle dynamics, and other a priori information. The decision system is in charge of navigating the car from its initial position to a user-defined final destination, taking into account the current state of the car and the internal representation of the environment, as well as the traffic rules and the passenger's safety and comfort.

To navigate the car across the environment, the decision system needs to know where the self-driving car is located within it. The localizer subsystem is in charge of the estimation of the car's state (pose, linear velocity, angular velocity, etc.) from static maps of the environment. Such static maps, or offline maps, are computed autonomously prior to the self-driving operation, usually by using the self-driving car's own sensors, although manual annotation or editing is often required. Self-driving cars can use one or more offline maps for localization, such as occupancy grid maps, mitigation maps, or landmark maps.

Information about the rules and regulations (traffic direction, maximum speed, lane delineation, etc.) governing how self-driving cars move on roads and highways is also crucial for decision-making systems. This information is usually embedded in road maps, where geometric and topological features are used to represent it in such maps.

The localizer subsystem receives offline maps, sensor data, and the odometer of the self-driving car as inputs and calculates the state of the car as an output (Figure 1). It is worth noting that although GPS can help the positioning process, GPS positioning alone is not sufficient in urban environments due to interference caused by tall trees, buildings, tunnels, etc., which makes GPS positioning unreliable.

The mapper subsystem receives an offline map and the state of the self-driving car as input and generates an online map as output. This online map is usually a merging of the information in the offline map and an occupancy grid map calculated online using sensor data and the current state of the car. It is preferable that the online map contains only a static representation of the environment, as this may assist in the operation of certain subsystems of the decision system.

Information about the attitude and speed of moving obstacles is also vital to the decision-making system. This information enables the system to make decisions to avoid collisions with moving obstacles. It also allows moving obstacles to be removed from the online map. The moving object tracking subsystem, the MOT, receives the offline map and the state of the self-driving car and detects and tracks, i.e. calculates, the pose and speed of the nearest moving obstacle (e.g. other vehicles and pedestrians). We survey the literature on moving object detection and tracking methods for self-driving cars in Sec. 3.

Horizontal (i.e. lane markings) and vertical (i.e. speed limits, traffic lights, etc.) traffic signalisation must be recognised and obeyed by autonomous vehicles. The traffic signalisation detector subsystem, the TSD, is responsible for detecting and recognising traffic signalisation. It receives data from sensors and the state of the car, detects the position of the traffic signals and identifies their level or status.

That's all about the subsystems of perception system, the decision making systems is out of concern of this survey. It will not be explained in this paper.

3. MOT subsystem

The Moving Object Tracker (MOT) subsystem is responsible for detecting and tracking the posture of moving obstacles in the environment around the self-driving car. This subsystem is critical in enabling self-driving cars to make decisions on how to act to avoid colliding with potential moving objects. Over time, the position of moving obstacles is usually estimated by range sensors such as LIDAR and radar or data captured by stereo and monocular cameras. Images from monocular cameras help to provide a

wealth of information on appearance that can be used to improve assumptions about moving obstacles.

3.1. Traditional based MOT

The traditional MOT approach follows three main steps: data segmentation, data association and filtering. In the data segmentation step, the sensor data is segmented using clustering or pattern recognition techniques. In the data association step, segments of the data are linked to targets using data association techniques. In the filtering phase, for each target, the position is estimated by taking the geometric mean of the data assigned to that target. The position estimate is usually updated by a Kalman or particle filter. One traditional approach is to use 3D LiDAR sensors to detect and track moving vehicles. The 3D LiDAR point cloud is segmented into clusters of points using Euclidean distances. Obstacles observed in the current scan of the sensor are correlated with obstacles observed in previous scans using a nearest neighbour algorithm. The state of the obstacle is estimated using a particle filtering algorithm. Obstacles with velocities above a given threshold are considered as moving vehicles. Data correlation is solved by an optimisation algorithm. A multi-hypothesis tracking algorithm is used to reduce association errors. Once filtered, object tracking was performed based on a segmental matching technique using features extracted from the images and 3D points.

3.2. Model based MOT

The model-based approach uses a physical model of the sensor and a geometric model of the object with a non-parametric filter to infer directly from the sensor data. Data segmentation and association steps are not required as the geometric object model associates the data with the target. The hypothesis of using differences in LiDAR data between successive scans to detect moving vehicles. Instead of separating the data segmentation and association steps, new sensor data is incorporated by updating the state of each vehicle target, which includes the vehicle's pose and geometry [10].This is achieved by combining a hybrid formulation of Kalman filtering and Rao-Blackwellized particle filtering. The geometry becomes a tracked variable, meaning that its previous state is also used to predict the current state. A trajectory is a sequence of object shapes generated by an object over time which satisfies the constraints of the measurement model and the motion model from frame to frame. Due to the high computational complexity of this solution, the Data Driven Markov Chain Monte Carlo (DD-MCMC) technique is used, which efficiently traverses the solution space to find the best solution.DD-MCMC aims to sample the probability distribution of a set of trajectories, given the set of observations over a time interval. In each iteration, DD-MCMC draws a new state from the current state according to the proposed distribution. The new candidate state is accepted with a certain probability. To provide DD-MCMC with an initial proposal, dynamic segments that fall into the free or unexplored areas occupying the grid map are detected from laser measurements and a moving obstacle hypothesis is generated by fitting a predefined object model to the dynamic segments. A Bayesian filter is responsible for jointly estimating the pose of the sensor, the geometry of the static local background, and the dynamics and geometry of the object. The geometric information consists of boundary points obtained with a 2D LiDAR. Basically, the system operates by iteratively updating the tracking state and associating the new measurements with the current target. The hierarchical data association works on two levels. At the first level, the new observations are matched to the current dynamic or static target. At the second level, the boundary points of the obstacle are updated.

3.3. Stereo vision based MOT

The stereo vision-based approach relies on the colour and depth information provided by stereo image pairs to detect and track moving obstacles in the environment. The method is used for obstacle detection and recognition and uses only synchronised video from a front-view stereo camera. The focus of their work is on obstacle tracking based on the output of each frame from pedestrian and car detectors. For obstacle detection, a support vector machine (SVM) classifier is typically used. For obstacle tracking, a "hypothesis-verification" strategy is typically used to match a set of trajectories to possible detected obstacles. The set of candidate trajectories is generated by Extended Kalman Filters (EKFs) and initialised with obstacle detection results. Finally, using model selection techniques, only the smallest, conflict-free set of trajectories that can explain past and present observations is retained [11].

3.4. Grid map based MOT

The grid map-based approach begins with constructing a occupied grid map of the dynamic environment (Petrovskaya). The map building step is followed by data segmentation, data association, and filtering steps to provide an object-level representation of the scene. Nguyen proposed a grid-based method for detecting and tracking moving targets using stereo cameras. Their work focuses on pedestrian detection and tracking. Reconstruct three-dimensional points from stereoscopic image pairs. On this basis, the inverse sensor model is used to estimate the occupancy probability of each cell in the grid map. The hierarchical segmentation method is used to cluster the grid cells according to the regional distance between them. Finally, the Interactive Multiple Model (IMM) method is used to track moving obstacles [12]. Azim and Aycard use an eight-axis tree-based 3D local occupancy grid map to divide the environment into occupied, free, and unknown voxels. Once a local grid map

is built, motion barriers can be detected based on inconsistencies between the observed free space and occupied space in the local grid map. Dynamic voxels are clustered into moving objects, which in turn are further divided into layers. Using geometric features extracted from each layer, moving objects are classified into known categories (pedestrians, bicycles, cars, or buses). Ge utilized 2.5D occupancy grid maps to simulate static backgrounds and detect moving obstacles. Mesh cells store the average height of three-dimensional points where a two-dimensional projection falls into the mesh domain. Detect motion assumptions from differences between the current mesh and background models.

3.5. Sensor fusion based MOT

Sensor fusion-based approaches fuse data from various sensors, such as lidar, radar, and cameras, to explore their respective characteristics and improve situational awareness. Darms propose a sensor-fusion-based approach to detecting and tracking moving vehicles employed by autonomous vehicle Boss (Urmson) (Carnegie Mellon University Carnegie Mellon University's car, which won first place in the 2007 DARPA Urban Challenge). The MOT subsystem is divided into two layers. The sensor layer extracts features from the sensor data that can be used to describe the movement barrier hypothesis based on a point model or box model. The sensor layer also attempts to link features with current prediction assumptions from the fusion layer. Features that cannot be associated with existing assumptions are used to generate new proposals. Generates an observation for each feature associated with a given hypothesis and encapsulates all the information needed to update the hypothetical state estimate. Based on the recommendations and observations made by the sensor layer, the fusion layer selects the best tracking model for each hypothesis and uses Kalman filtering to estimate (or update the estimate) the hypothesis state. Cho describe a new MOT subsystem used in a new experimental autonomous vehicle at Carnegie Mellon University. The previous MOT subsystem proposed by Darms was extended to utilize camera data to identify classes of moving objects (e.g., cars, pedestrians, and bicycles) and augment measurements from automotive-grade active sensors (e.g., lidar and radar). The scan lines used by Mertz can be obtained directly from 2D lidar, projection of 3D lidar on a 2D plane, or from the fusion of multiple sensors (lidar, radar, and camera). Scan lines are converted to world coordinates and split. Extract line and corner features for each segment. Segments are updated with existing barriers and targets using the Kalman filter. Na merged the trajectories of moving obstacles generated by multiple sensors such as radar, 2D lidar, and 3D lidar. The 2D lidar data is projected onto a 2D plane and moving obstacles are tracked using a joint probabilistic data correlation filter (JPDAF). The 3D lidar data is projected onto the image and segmented into moving obstacles using an area growth algorithm. Finally, iterative closest point (ICP) matching or image-based data correlation is used to estimate or update trajectory poses. Xu describe the use of situational awareness to track moving obstacles to maintain distance using a new experimental autonomous vehicle from Carnegie Mellon University. In the context of a given behavior, the road network generates ROI. Candidate targets are found within the ROI and projected into road coordinates. The distance keeping goal is achieved by associating all candidate targets from different sensors (lidar, radar, and camera). Xue fused lidar and camera data to improve the accuracy of pedestrian detection. They use a high degree of prior knowledge of pedestrians to reduce false detection. They estimated the height of pedestrians based on the pinhole camera equation (a combination of camera and lidar measurements).

3.6. Deep learning based MOT

The deep learning-based approach uses deep neural networks to detect the location and geometry of moving obstacles and track their future state based on current camera data. Huval et al. propose a neural network-based approach to motion vehicle detection that uses the Overfeat convolutional neural network (CNN) (Sermanet) and monocular input images with a focus on real-time. Overfeat CNN's goal is to predict the position and distance (depth) of a car in the same driving direction just by looking at the car's rearview mirror. Mutz et al. solved the problem of motion obstacle tracking for a closely related application "Follow the Leader" that is primarily suitable for fleets of autonomous vehicles. This tracking method is based on generic target tracking (GOTURN) using regression networks (Held et al.). GOTURN IS A PRE-TRAINED DEEP NEURAL NETWORK THAT TRACKS GENERAL OBJECTS WITHOUT FURTHER TRAINING OR OBJECT-SPECIFIC FINE-TUNING. Initially, GOTURN receives an image and a manually demarcated leading vehicle bounding box as input. Suppose the object of interest is in the center of the bounding box. Subsequently, for each new image, GOTURN gives an estimate of the position and geometry (height and width) of the bounding box as output. The location of the lead vehicle is estimated with lidar points that fall within the bounding box and are considered vehicles.

4. Collaborative Perception

Perception is one of the key modules of autonomous driving systems, but the limited capabilities of bicycles create a bottleneck for improving perception performance. In order to break through the limitations of individual perception, collaborative perception is proposed, enabling vehicles to share information and perceive the environment beyond the line of sight and beyond the field of vision. This paper

reviews promising work on collaborative sensing technologies, including basic concepts, collaborative models, and key elements and applications. Finally, the challenges and problems of openness in this research area are discussed and further directions are given. Two important problems with single perception are long-range occlusion and sparse data. The solution to these problems is that vehicles in the same area share a common perception message (CPM) with each other, and the co-perception environment is called co-perception or cooperative perception Thanks to the construction of communication infrastructure and the development of communication technologies such as V2X, vehicles can exchange information in a reliable way and thus enable collaboration. Recent work has shown that cooperative sensing between vehicles can improve the accuracy of context perception as well as the robustness and safety of transportation systems.

In addition, autonomous vehicles are often equipped with high-fidelity sensors for reliable perception, resulting in high costs. Collaborative sensing can alleviate the demanding requirements of individual vehicles for sensing devices.

4.1. Synergistic Classification

Collaborative sensing shares information with nearby vehicles and infrastructure, enabling autonomous vehicles to overcome certain perception limitations, such as occlusion and short field of view. However, achieving real-time and robust co-sensing requires addressing some of the challenges posed by communication capacity and noise. Recently, there has been some work on the strategy of collaborative perception, including what collaboration is, when to collaborate, how to collaborate, alignment of shared information, etc.

Similar to fusion, there are 4 categories of synergistic classification:

1. Early collaboration

   Early collaboration takes place in the input space, sharing raw sensory data between the vehicle and infrastructure. It aggregates raw measurements from all vehicles and infrastructure to get a holistic view. Therefore, each vehicle can perform the following processing and complete the perception based on the overall perspective, which can fundamentally solve the occlusion and long-distance problems that occur in single perception.

   However, sharing raw sensory data requires a lot of communication and tends to congest communication networks due to excessive data load, which in most cases hinders its practical application.

2. Post-synergy

   Later collaboration is carried out in the output space, which promotes the fusion of the perceptual results of each intelligent output to achieve refinement.

   Although late synergy has bandwidth economy, it is very sensitive to the positioning error of the intelligent body, and suffers from high estimation error and noise due to incomplete local observation.

3. Intermediate synergy

   Intermediate collaboration takes place in intermediate feature space. It is able to transfer intermediate features generated by individual Homo sapiens predictive models. After fusing these features, each sapient body decodes the fused features and produces a perceptual result. Conceptually, representative information can be compressed into these features, which can save communication bandwidth compared to early collaboration, and improve perception compared to later collaboration.

   In practice, the design of this collaborative strategy is algorithmically challenging in two aspects: i) how to select the most efficient and compact features from the original measurements for transmission; and ii) how to maximize the integration of the characteristics of other intellects to enhance the perception of each body.

4. Hybrid synergy

   As mentioned above, each synergy mode has its advantages and disadvantages. As a result, some efforts employ hybrid collaboration, combining two or more collaboration models to optimize collaboration strategies.

4.2. Factors

The main factors of co-sensing include:

1. Synergy diagram

   Graphs are a powerful tool for collaborative perception modeling because they have good interpret ability for modeling non-Euclidean data structures. In some work, the vehicles participating in collaborative sensing form a complete collaboration graph, where each vehicle is a node, and the synergy relationship between the two vehicles is the edge between these two nodes.

2. Pose alignment

   Since collaborative sensing requires the fusion of data from vehicles and infrastructure at different locations and at different times, achieving precise data alignment is critical to successful collaboration.

3. Information fusion Info fusion is a core component of a multi-body system, and its goal is to fuse the most

informative parts of other bodies in an efficient manner.

4. Resource allocation based on reinforcement learning

   Limited communication bandwidth in real-world environments requires the best use of available communication resources, which makes resource allocation and spectrum sharing important. In the vehicle communication environment, rapidly changing channel conditions and increasing service requirements make the optimization of distribution problems very complex and difficult to solve using traditional optimization methods. Some work utilizes multi-intelligence reinforcement learning (MARL) to solve optimization problems.

4.3. Challenging Questions

4.3.1 Communication robustness

Effective co-ordination relies on reliable communication between intellects and bodies. However, communication is not perfect in practice: i) as the number of vehicles in the network increases, the available communication bandwidth per vehicle is limited; ii) difficulty for vehicles to receive real-time information from other vehicles due to unavoidable communication delays; iii) communications may sometimes be interrupted, resulting in a disruption of communications; iv) V2X communications are under attack and cannot always provide reliable services. Despite the continuous development of communication technology and the continuous improvement of the quality of communication services, the above problems will continue to exist for a long time. However, most existing work assumes that information can be shared in a real-time and lossless manner, so considering these communication constraints and designing robust collaborative sensing systems is important for further work.

4.3.2 Heterogeneity and trans modality

Most co-system perception efforts focus on lidar point cloud-based perception. However, there are more types of data available for perception, such as images and millimeter-wave radar point clouds. This is a potential way to leverage multi modal sensor data for more effective collaboration. In addition, in some scenarios, there are different levels of autonomous vehicles that provide different quality information. Therefore, how to perform collaboration in heterogeneous vehicle networks is a problem for further practical applications of collaborative sensing. Unfortunately, little work has focused on heterogeneous and cross-modal collaborative awareness, which has also become an open challenge.

4.3.3 Large-scale datasets

Large-scale datasets and the development of deep learning methods have improved perceptual performance. However, existing datasets in the field of co-sensing research are either small in size or not publicly available.

The lack of public large-scale datasets hinders the further development of collaborative sensing. In addition, most datasets are based on simulations. While simulation is an economical and safe way to validate algorithms, real-world datasets are needed to put collaborative sensing into practice.

References

[1] Shunsuke Aoki, Takamasa Higuchi, and Onur Altintas. Cooperative perception with deep reinforcement learning for connected vehicles. *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 328–334, 2020. 1

[2] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2022. 1

[3] Carlos Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, Wei-Lun Chao, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21351–21360, 2022. 1

[4] Chuqing Hu, Sinclair Hudson, Martin Ethier, Mohammad K. Al-Sharman, Derek Rayside, and William W. Melek. Sim-to-real domain adaptation for lane detection and classification in autonomous driving. *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 457–463, 2022. 1

[5] Shengchao Hu, Li Chen, Peng Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. *ArXiv*, abs/2207.07601, 2022. 1

[6] Yeping Hu, Xiaogang Jia, Masayoshi Tomizuka, and Wei Zhan. Causal-based time series domain generalization for vehicle intention prediction. *2022 International Conference on Robotics and Automation (ICRA)*, pages 7806–7813, 2021. 1

[7] Yan Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yulin Jiang. Polarformer:

Multi-camera 3d object detection with polar transformers. *ArXiv*, abs/2206.15398, 2022. 1

[8] Peizhao Li, Puzuo Wang, Karl Berntorp, and Hongfu Liu. Exploiting temporal relations on radar perception for autonomous driving. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17050–17059, 2022. 1

[9] Rafid Mahmood, James Lucas, David Acuna, Daiqing Li, Jonah Philion, J. M. López Álvarez, Zhiding Yu, Sanja Fidler, and Marc Teva Law. How much more data do i need? estimating requirements for downstream tasks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 275–284, 2022. 1

[10] Gregory P. Meyer, Jake Charland, Shreyash Pandey, Ankita Gajanan Laddha, Shivam Gautam, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Laserflow: Efficient and probabilistic object detection and motion forecasting. *IEEE Robotics and Automation Letters*, 6:526–533, 2020. 3

[11] Shunli Ren, Siheng Chen, and Wenjun Zhang. Collaborative perception for autonomous driving: Current status and future trend. *ArXiv*, abs/2208.10371, 2022. 3

[12] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9881–9891, 2022. 1, 3

[13] Hao-Chiang Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Tang Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. *ArXiv*, abs/2207.14024, 2022. 1

[14] Guohang Yan, Liu Zhuochun, Chengjie Wang, Chunlei Shi, Pengjin Wei, Xinyu Cai, Tengyu Ma, Zhizheng Liu, Zebin Zhong, Yuqian Liu, Ming Zhao, Zheng Ma, and Yikang Li. Opencalib: A multi-sensor calibration toolbox for autonomous driving. *ArXiv*, abs/2205.14087, 2022. 1

[15] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *ArXiv*, abs/2110.11334, 2021. 1

[16] Lingyao Zhang, PoHao Su, Jerrick Hoang, G. Clark Haynes, and Micol Marchetti-Bowick. Map-adaptive goal-based trajectory prediction. In *Conference on Robot Learning*, 2020. 1