Vision-RADAR fusion for Robotics BEV Detections: A Survey

Apoorv Singh
Perception Team
Motional
Pittsburgh, USA
apoorv.singh@motional.com

2023 IEEE Intelligent Vehicles Symposium (IV) | 979-8-3503-4691-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IV55152.2023.10186647

Abstract*—Due to the trending need of building autonomous robotic perception system, sensor fusion has attracted a lot of attention amongst researchers and engineers to make best use of cross-modality information. However, in order to build a robotic platform at scale we need to emphasize on autonomous robot platform bring-up cost as well. Cameras and radars, which inherently includes complementary perception information, has potential for developing autonomous robotic platform at scale. However, there is a limited work around radar fused with Vision, compared to LiDAR fused with vision work. In this paper, we tackle this gap with a survey on Vision-Radar fusion approaches for a BEV object detection system. First we go through the background information viz., object detection tasks, choice of sensors, sensor setup, benchmark datasets and evaluation metrics for a robotic perception system. Later, we cover per-modality (Camera and RADAR) data representation, then we go into detail about sensor fusion techniques based on sub-groups viz., early-fusion, deep-fusion, and late-fusion to easily understand the pros and cons of each method. Finally, we propose possible future trends for vision-radar fusion to enlighten future research. Regularly updated summary can be found at: *https://github.com/ApoorvRoboticist/Vision-RADAR-Fusion-BEV-Survey

***Index Terms*—computer vision; radar; sensor fusion; camera radar fusion; object detection; BEV Perception; robotics; autonomous driving; review; survey**

I. Introduction

SAE (Society of Automotive Engineers) has divided the roles of a human driver and driving automation capabilities by the levels of automation viz., Level 0: No driving automation; Level 1: Driver assistance; Level 2: Partial driving automation; Level 3: Conditional driving automation; Level 4: High driving automation; Level 5: Full driving automation. 3D object detection is an essential task for autonomous driving for Level 2 on-wards. However, in order to make these robotic platform at large-scale we need to emphasize on affordable active-saftey hard-wares. Camera and radar perception sensors setup is a low-cost, high-reliability, and low-maintenance one. It can provide rich semantic information with cameras; and long-range detections that are robust to lighting/ weather conditions with radars. LiDAR is a popular choice of sensor for Level 4+ cars however, cameras and radars have dominated L2-L3 levels cars that have been in production over a decade. Recently a lot of interesting work has been explored to utilize camera-radar combination in higher level of automation like [1], [2], and [3].

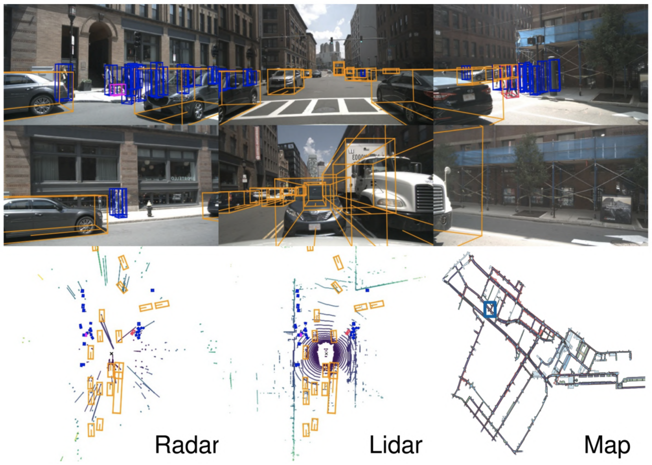

Fig. 1. BEV Perception with camera, radar, HD-map and lidar data. Snapshot taken from Multi-modal dataset, nuScenes [4].

Reference [5] has shown camera and radar's characteristics and how they complement each other. Cameras are not very good in generalizing in BEV predictions, as the input they receive is constrained by 2D pixels. However, they include very rich semantic and boundary information. Radar's data already includes 3D as well as velocity data in the input point-cloud. However, it lacks dense semantic information. For these reasons, camera-radar sensor combination can work very well together, however data received by these sensors need to be mapped to a single coordinate frame. Input data received by them can be visualized in as Fig. 1

Previous work [6] has only considered vision and lidar aspects. [7] and [8] have covered vision and radar, but they don't dive deep enough on modern deep learning based techniques which are trending in literature these days. With this paper, we plan to target this gap by covering the basics behind BEV detection and senor modalities and then diving deeper into modern vision-radar fusion techniques giving more focus on the trending transformer based approaches.

As shown in Fig. 2, rest of the paper is organized as follows: We first look at the background information required

This review was done while working at Perception team at Motional.

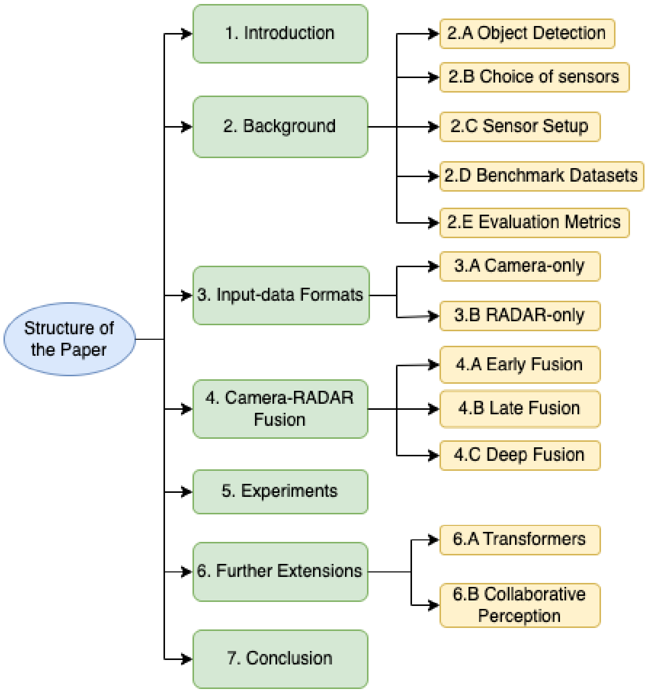

Fig. 2. Structure of this Survey Paper.

to understand robotic BEV perception viz., information about Object detection tasks, Choice of sensors, Benchmark datasets, Evaluation metrics etc in section II. Then, we introduce input-data formats for cameras and radars in section III. In section IV, we will go through detailed analysis of techniques involved with camera-radar fusion methods. We will also sub-group them so that readers can follow through easily. Later, in section V we will show how discussed methods evaluate on the camera-radar benchmark, nuScenes [4]. Then in section VI we will go through possible extensions looking at the current research trend that may enlighten future research. Finally in section VII we will conclude our findings.

II. Background

A. Object Detection Task

3D Object Detection is an essential task for robotic/ autonomous driving platform. Object detection is a combination of two fundamental computer-vision problems viz., classification and localization. The goal of object detection is to detect all instances of the predefined classes and provide its localization in the image/ BEV space with axis-aligned boxes. It is generally seen as a supervised learning problem which leverages huge amount of labelled images. Few of the key challenges in the object detection task include:

- *Box BEV representation*: Camera images are in perspective-view, however down-the-stream autonomy tasks operate in the Bird's Eye View (BEV). Hence we need a way to transform perspective information to the orthogonal space, BEV. This comes with inherent problem of depth ambiguity, as we are adding a new dimension of depth to this problem.
- *Rich Semantic Information*: Sometimes we need to distinguish between very similar looking objects, for example multiple similar looking objects in close vicinity or maybe a pedestrian operating on a skate-board. In later example, pedestrian on skate-board should follow a cyclist's motion-model, but it is a very hard to detect this attribute of a pedestrian. To identify for these fine-grained information we need to embed deep semantics in our model.
- *Efficiency*: As we are building bigger and deeper networks, we need expensive computation resources to make an deploy-time inference. Edge devices being the common place for deployment platform, it can easily become a bottleneck.
- *Out of domain objects*: There is a limit to the classes we can train the network with. There will always be some class of object which we may encounter at test time that we haven't seen during training time. We always have lack of some generalization capabilities with detectors.

B. Choice of Sensors

Cameras and sensors have complementary features, which sets them for robust perception sensor combination. Camera's contribution for detection comes from: Rich semantic information and accurate boundaries. Camera is not very good in fusing temporal data or predicting boxes with accurate depth specially in bad weather conditions. However, radar picks up where camera lags behind. Radars can predict depth and velocity of objects very accurately leveraging doppler effect in their point-cloud. Radar data is very sparse, so it doesn't take too much compute load as well. Radars longer wavelength compared to other laser sensors enables them to be the only perception sensor, for which performance doesn't degrade with adverse weather conditions viz., rain/ snow/ dust etc. These characteristics is very well summarized by [5] in Fig. 3.

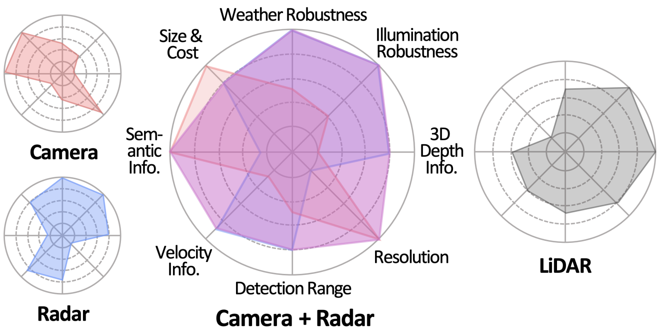

Fig. 3. Sensor characteristics of camera, radar, and LiDAR. Camera-radar fusion has high potential considering spatially and contextually complementary properties.

One of the other less talked issue with radars is its inability to detect velocity components of agents along the radial direction as shown in Fig. 5. Another place where radar, and in-fact any laser-based sensor falls behind is detecting black objects/ cars that absorb most of the lasers that falls on them. Camera is the fall-back sensor to rely in these special cases.

C. Sensor Setup

There's a setup of suite of sensors in autonomous vehicles (AV), which may vary depending on different autonomous car companies. Typically there are $6-12$ cameras and $3-6$ radars per vehicle. These many sensors are needed to cover the entire surrounding 3D scene. We are limited to use cameras with normal FOV (Field of view) otherwise we may get image distortions that are beyond recovery, like with the Fish-eye cameras (Wide FOV), which are only good for up to few tens of meters. A perception sensor setup in one of the most cited benchmark-dataset, nuScenes [4] in the AV space can be seen in the Fig. 4. For the affordability reasons, AV/ mobile-robots industry have always been more invested in radars and cameras in production cars compared to lidars. In this example we see that there are qty. 5 radars, qty. 6 cameras and only qty. 1 lidar. These numbers are representative of other L3+ car companies as well.

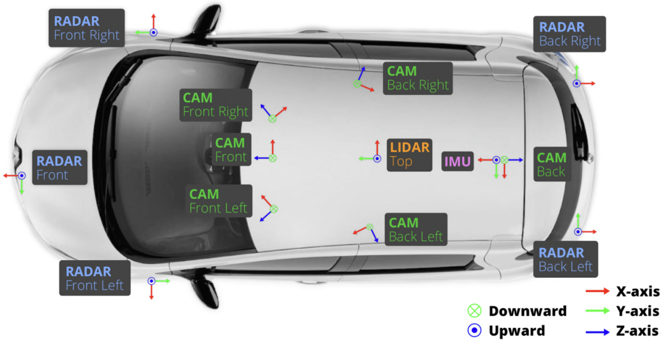

Fig. 4. nuScenes [4] sensor setup.

D. Benchmark Datasets

nuScenes [4], KITTI [9] and Waymo Open Dataset (WOD) [10] are the three most commonly used 3D BEV object detection task. Apart from them H3D [11], Lyft L5 [12], BDD [13], STF [14] and Argoverse [15] can also be used for BEV perception tasks. Detailed information on these datasets can be reviewed in Table I

E. Evaluation Metrics

3D object detectors use multiple criteria to measure performance of the detectors viz., precision and recall. However, mean Average Precision (mAP) is the most common evaluation metric. Intersection over Union (IoU) is the ratio of the area of overlap and area of the union between the predicted box and ground-truth box. An IoU threshold value (generally 0.5) is used to judge if a prediction box matches with any particular ground-truth box. If IoU is greater than the threshold, then that prediction is treated as a True Positive (TP) else it is a False Positive (FP). A ground-truth object which fails to detect with any prediction box, is treated as a False Negative (FN). Precision is the fraction of relevant instances among the retrieved instances; while recall is the fraction of relevant instances that were retrieved.

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

Based on the above equations, average precision is computed separately for each class. To compare performance between different detectors (mAP) is used. It is a weighted mean based on the number of ground-truths per class.

In addition, there are a few dataset specific metrics viz., KITTI introduces Average Orientation Similarity (AOS), which evaluates the quality of orientation estimation of boxes on the ground plane. mAP metric only considers 3D position of the objects, however, ignores the effects of both dimension and orientation. In relation to that, nuScenes introduces TP metrics viz., Average Translation Error (ATE), Average Scale Error (ASE) and Average Orientation Error (AOE). WOD introduces Average Precision weighted by heading (APH) as its main metric. It takes heading/ orientation information into the account as well. Also, given depth confusion for 2D-sensors like camera, WOD introduces Longitudinal Error Tolerant 3D Average Precision(LET-3D-AP), which emphasizes more on lateral errors than longitudinal errors in predictions.

III. Input-Data Formats

In this section we will cover raw data format returned by camera and radars, and meta data used to get them into the unified coordinate system i.e. egocentric Cartesian coordinate system.

A. Cameras

Surround-view camera images can be represented by $\mathbf{I} \in \mathbb{R}^{\mathrm{N{\times}V{\times}H{\times}W{\times}3}}$. Here, N,V, H and W are the number of temporal-frames, number views, height and width respectively. Given V camera images $\mathbf{X_k} \in \mathbb{R}^{\mathrm{3xHxW}}{}_V$, each with an extrinsics matrix $\mathbf{E_k} \in \mathbb{R}^{\mathrm{3x4}}$ and an intrinsics matrix $\mathbf{I_k} \in \mathbb{R}^{\mathrm{3x3}}$, we can find a rasterized BEV map of the feature in BEV coordinate frame as $\mathbf{y} \in \mathbb{R}^{\mathrm{CxXxY}}$, where C, X, and Y are channel depth, and height and width of BEV map. The extrinsic and intrinsic matrices together define the mapping from reference coordinates (x, y, z) to local pixel coordinates (h, w, d) for each of the V camera views. Refer to Fig. 1 for surround image view on an autonomous car.

B. RADARs

Radars are another set of active sensors used in robotics which transmit radio waves to sense the environment and measure the reflected waves to determine the location and velocity of objects. Raw output from the sensor is in polar coordinates, which can be easily converted to BEV space with

TABLE I
INFORMATION ON BENCHMARK DATASET COMMONLY USED FOR 3D BEV OBJECT DETECTION IN AUTONOMOUS DRIVING.

Dataset Sensor Setup	Kitti [9]	BDD [13]	WOD [10]	nuScenes [4]	STF [14]
RGB Cameras	2	1	5	6	2
RGB Resolution	$1242 * 372$	$1280 * 720$	$1920 * 1080$	$1600 * 900$	$1920 * 1024$
Lidar Sensors	1	×	5	1	2
Lidar Resolution	64	0	64	32	64
Radar Sensor	×	×	×	4	1
Frame Rate	10 Hz	30 Hz	10 Hz	10 Hz	10 Hz

sensor calibration matrices. However, noisy radar points have to go through filtering which would utilize some form of clustering and temporal tracking. This temporal tracking can be achieved by Kalman filters [16]. Kalman filters is a recursive algorithm, which can estimate the current state of the target by obtaining the previously observed target state estimation and the measured value of the current state. After running internal filtering they return 2D points in BEV (without height dimension), providing azimuth angle and radial distance to the object. It also produces radial velocity vector component per 2D point, as shown by [3] in Fig. 5. Here points can be treated as detected objects. In modern BEV sensor fusion research work, radar detections are represented as a 3D point in the egocentric coordinate system. This 3D point in the radar point-cloud is parameterized as $P = (x, y, z, v_x, v_y)$ where (x, y, z) is the position and (v_x, v_y) is the radial velocity of the object in the x and y direction. This radial velocity is a relative velocity, hence it needs to be compensated with ego vehicle's motion. Due to the high sparsity of this radar point-cloud, we generally aggregate 3-5 temporal sweeps. It adds a temporal dimension to the point cloud representation. Since in lot of approaches detection head runs on the 360° surround-scene, we merge 3D points from all the radars around the vehicle into a single merged point-cloud. The nuScenes [4] dataset provides the calibration parameters needed for mapping the radar point clouds from the radar coordinate system to the egocentric coordinate frame. Refer to Fig. 1 for radar point-cloud from an autonomous car.

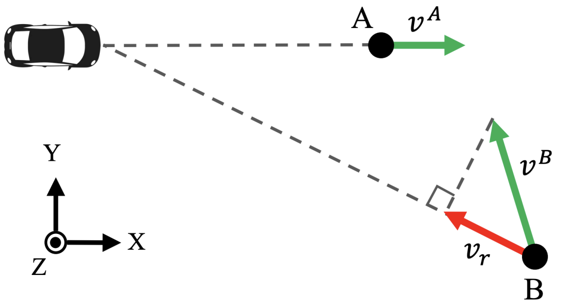

Fig. 5. [3] shows difference between actual and radial velocity. For target A, velocity in the vehicle coordinate system and the radial velocity are the same (v_A). For target B on the other hand, radial velocity (v_r) as reported by the radar is different from the actual velocity of the object (v_B) in the vehicle coordinate system

IV. CAMERA-RADAR FUSION

Based on at what stage we fuse information of two sensors, these methods can be categorized into three classes viz., early, late and deep fusion. The early and late fusions both have only one interactive operation of different features, which is processed either at the beginning or at the end of the module. However, the deep fusion has more interactive operations of different features. These three approaches can be easily summarized in Fig. 6.

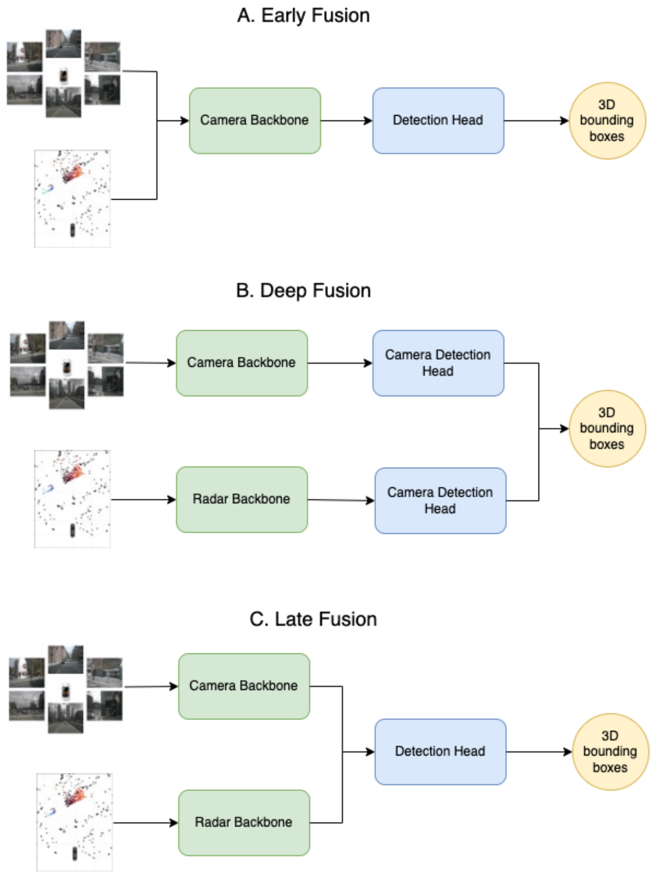

Fig. 6. Modality Fusion Methods viz., A. Early Fusion; B. Late Fusion; C. Deep Fusion.

A. Early Fusion

Early fusion is also referred as data-level fusion. It is the least explored option out of the three. In this approach information from both the sensors is fused together at very early stage i.e. before computation of any features. One of the key challenges in this approach is the synchronization of data. We have cameras and radars data which comes in different coordinate space and moreover nature of the data is also quite orthogonal, where former is a densely packed 2D pixels and later is sparsely packed BEV point cloud. This approach has minimal data-loss issues however, there is no effective way to handle the complexity of aggregating raw data from camera and radar. The common line of work in this fusion category is generally done sequentially. Here we first extract region of interests (ROI) based on radar point; then project them on camera; and use some heuristics to gather camera features in the region as done in [17], [18]. This is not very reliable approach as there is a high probability that critical objects might get filtered out in radar point-cloud beforehand and due to the nature of the design we won't even look for those objects in the images. However, added benefit with this approach is that we will only run convolution operations on the part of the image that lies inside ROI, hence saving us some compute budget.

B. Late Fusion

Late fusion stream of work is the easiest of the three, which makes it the most common approach since past decade of work on camera-radar fusion based object detection. It is pretty certain from our previous reasoning that some of the objects and attributes are better handled by cameras and others by radars. This method lets respective sensors detect objects, and fuse 2 sets of detections together as 1 set of detections using trivial data association techniques [19]. However, this approach is not able to leverage the fact that features in one sensor-detector won't be able to be augmented by features of the other detector. For example cameras in general can detect boundaries very well and radars can detect velocity with good confidence. Work in this stream can be further classified in two section:

1) Probabilistic Reasoning Based: In this approach Bayesian tracking method tracks multi-agent targets with probability-density for multi-modes. It approximates each mode with component probability density. Bayesian algorithm and Particle filter (PF) handles the non-linearities and non-gaussian estimations. It is an iterative algorithm which recursively estimates the state of multiple targets and determine the current target number using maximum-likelihood. Refer [20] and [21] for representative work.

2) Kalman Filter Based: In this approach we estimate the current state of the target by obtaining the previously observed target state estimation and the measured value of the current state as in [16]. Simple kalman filter can not incorporate nonlinear systems accurately. However, EKF (Extended Kalman Filter) and UKF (Unscented Kalman Filter) are more sophisticated systems that can incorporate non-linearities in the system. EKF linearizes the nonlinear problem, whereas UKF adopts statistical linearization technique to linearize nonlinear function of random variables by sampled points.
SORT [22] and Deep-SORT [23] are seminal papers in this category. SORT explores multi-object tracking task with hungarian matching for data association and constant velocity motion model with kalman fitler estimation. Deep SORT is further extension to this work where authors also add in appearance information in the form of image features in the algorithm. Both these algorithms are very cheap and can be easily handled by the edge-device. MHT [24] is another tracking-by-detection approach which maintains small list of potential hypotheses, which can be facilitated with the accurate object detectors that are currently available.

Late fusion methods can have the benefit of exploiting the off-the-shelf detection algorithms that are independently developed as modular components. However, late fusion strategies that rely on heuristics and post-processing techniques suffer from performance-reliability trade-offs, especially when these two sensors disagrees.

C. Deep Fusion

Deep Fusion (a.k.a mid-fusion) [25] is also referred as feature-level fusion. In this approach we fuse information of the two sensors in the form of features, so take it as an intermediate of the previously discussed methods. This approach seems most future promising based on the current research work. This is a learning-based approach, where features from cameras and radars can be computed in-parallel and then soft-associated with each other. This approach can be further classified in three sections:

1) Radar Image Generation based: In order to bring radar information into the image domain, radar's features are extracted and transformed into image-like matrix information. This is referred as *radar imagery*. Channels of this radar imagery represents information from the point representation of the radar i.e. physical quantity like distance, speed and so on. [26], [27], [28], [29] follow this line of work. This approach hasn't been very successful because of inherent sparsity in radar point-cloud which makes them incompatible to formulate into a good image-like matrix.

2) CNN based: This line of work focuses on convolution neural networks (CNNs) for doing feature fusion from two different modalities. CNN based detectors used to be SOTA until 2 years ago, until transformer started making contribution on spatial context. In CNN's segment, one of the representative work [30] uses a neural network that builds on RentinaNet [31] with VGG backbone [32]. It uses radar channels to augment the image. This model makes the problem simpler by estimating 2D boxes. As authors of [30] claim that the amount of information encoded in one radar point is different from that of one pixel, we can not just simply early fuse this distinct information. A more optimal solution would be to do it in CNN's deeper layer where information is more compressed

TABLE II
Benchmark result for 3D object detection using Camera-radar fusion methods. Abbreviations defined in section V. *No publication yet, results are taken from nuScenes leaderboad.

Method	Year	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓	NDS ↑
CenterFusion [3]	2020	0.326	0.631	0.261	0.516	0.614	0.115	0.449
CRAFT [5]	2022	0.411	0.467	0.268	0.456	0.519	0.114	0.523
DETR3d RV*	2022	0.439	0.537	0.264	0.408	0.477	0.119	0.539
RCFormer*	2022	0.485	0.549	0.248	0.360	0.320	0.116	0.583
KRRDepth*	2022	0.519	0.0.416	0.257	0.422	0.377	0.135	0.608
KRRDepth*	2022	0.519	0.0.416	0.257	0.422	0.377	0.135	0.608

and contains more relevant information in the latent space. Since it is hard to abstract that what depth is the right depth for fusion, authors designed a network in a way that it learns itself this fusion strategy. These authors also introduced a technique called *BlackIn* [33] where they use dropout strategy but at sensor level instead of neuron level. This helps in leveraging sparse radar points information more which could have been easily shadowed by the dense camera pixels.

CenterFusion [3] is another modern work which builds on center-point detection framework [34] to detect objects. They solve key data association problem using a novel frustum-based method to associate the radar detections to their corresponding object center. Associated radar detections are used to generate radar-based feature maps to complement the image features and regress to object properties such as depth, rotation and velocity. They claim that just adding the radar input can significantly improve velocity estimation without the need of complex temporal information. Major issue with this work is that it treats primary sensor as camera and will be straight-away discarding detections which are only sensed by the radars. Another problem that we see with this approach is it samples radar points based on BEV center in image. However, there is no guarantee that image network would be able to predict good BEV center due to it's 2D perspective view input-data.

3) Transformer based: This line of work typically utilizes transformers module viz., cross-attention to cross-attend features from different modalities and form a finer feature representation. A representative work in CRAFT [5] associates image proposals with radar point in the polar coordinate system to efficiently handle the discrepancy between the coordinate system and spatial properties. Then in second stage, they use consecutive cross-attention based feature fusion layers to share spatio-contextual information between camera and radar. This paper is one of the SOTA methods on the leaderboard [4] as of date. MT-DETR is another approach which utilizes similar cross-attention structure to fuse cross-modality features.

V. Experiments

nuScenes [4] is the widely used datasets in the literature for which sensor setup in Fig. 4 includes 6 calibrated cameras and 5 radars covering the entire 360° scene. Results on discussed pioneer works are shown on the test set of nuScenes in Table II. This is under the filter *camera-radar track detections*. The key for the metric abbreviations is as follows: mAP: mean Average Precision; mATE: mean Average Translation Error; mASE: mean Average Scale Error; mAOE: mean Average Orientation Error; mAVE: mean Average Velocity Error; mAAE: mean Average Attribute Error; NDS: nuScenes detection score.

VI. Further Extensions

Based on the most-recent developments around the production multi-domain BEV perception detections, we will highlight possible directions for the future research.

A. Transformer Extensions

Looking at the trend in the benchmark datasets, it is pretty apparent that transformer based networks are able to establish right modelling between vision and radar data for getting good fused-feature representation. Even in vision-only based approaches transformers are ahead of thier convolutional counterparts. As highlighted in Table II DETR3D [35] and BEVFormer [36] can be easily extended to initiate queries from radar point-cloud as well. Instead of cross-attending to just vision features, a new cross-attention layer can be added for radar imagery.

B. Collaborative Perception

A relatively new field of area is how to make use of multi-agents, multi-modal transformers to enable collaborative perception. This setup requires a minimal infrastructure setup to enable smooth communications between different autonomous vehicle on the road. CoBEVT [37] shows initial proof of how Vehicle-to-Vehicle communication may lead to superior perception performance. They test their performance on OPV2V [38] benchmark dataset for V2V perception.

VII. Conclusion

For the autonomous vehicle's perception reliability, 3D object detection is one of the key challenge which we need to solve. In-fact, to make this problem even harder, we need to do this with sensors which are affordable enough to extend this technology to masses, thereby proving that the life-time cost of an AV is less than that of a driver-operated cab/ vehicle. Camera and RADAR are one of the key sensors that we can leverage to achieve this target. In this paper, we first covered background information to understand why it makes good technical as well as business sense to use cameras and radars for BEV object detection. Then we went into more details on how camera and radar input data is represented. Then we cover state-of-the-art techniques used in the literature and industry for camera-radar fusion by sub-grouping them so that readers can easily follow through. We hope our work will inspire future research on camera-radar fusion for 3D object detection.

References

[1] T.-Y. Lim, "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," in *arXiv*, 2019.

[2] Y. Kim, J. W. Choi, and D. Kum, "Grif net: Gated region of interest fusion network for robust 3d object detection from radar point cloud and monocular image," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10857–10864, 2020.

[3] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1527–1536, January 2021.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *CoRR*, vol. abs/1903.11027, 2019.

[5] Y. Kim, S. Kim, J. W. Choi, and D. Kum, "Craft: Camera-radar 3d object detection with spatio-contextual fusion transformer," *arXiv preprint arXiv:2209.06535*, 2022.

[6] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multimodal sensor fusion for auto driving perception: A survey," *CoRR*, vol. abs/2202.02703, 2022.

[7] Z. Wei, F. Zhang, S. Chang, Y. Liu, H. Wu, and Z. Feng, "Mmwave radar and vision fusion for object detection for autonomous driving: A review," *CoRR*, vol. abs/2108.03004, 2021.

[8] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[10] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," *CoRR*, vol. abs/1912.04838, 2019.

[11] A. Patil, S. Malla, H. Gang, and Y. Chen, "The H3D dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," *CoRR*, vol. abs/1903.01568, 2019.

[12] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," 2020.

[13] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *CoRR*, vol. abs/1805.04687, 2018.

[14] M. Bijelic, F. Mannan, T. Gruber, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep sensor fusion in the absence of labeled training data," *CoRR*, vol. abs/1902.08913, 2019.

[15] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] B. Lagos-Álvarez, L. Padilla, J. Mateu, and G. Ferreira, "A kalman filter method for estimation and prediction of space–time data with an autoregressive structure," *Journal of Statistical Planning and Inference*, vol. 203, pp. 117–130, 2019.

[17] S. Milch and M. Behrens, "Pedestrian detection with radar and computer vision," in *arXiv*, 2001.

[18] F. Garcia, P. Cerri, A. Broggi, A. de la Escalera, and J. M. Armingol, "Data fusion for overtaking vehicle detection based on radar and optical flow," in *2012 IEEE intelligent vehicles symposium*, pp. 494–499, IEEE, 2012.

[19] S. Gautam, G. P. Meyer, C. Vallespi-Gonzalez, and B. C. Becker, "Sdv-tracker: Real-time multi-sensor association and tracking for self-driving vehicles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3012–3021, 2021.

[20] M. L. Krieg, "Joint multi-sensor kinematic and attribute tracking using bayesian belief networks," *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, vol. 1, pp. 17–24, 2003.

[21] K. Kim, C. Lee, D. Pae, and M. Lim, "Sensor fusion for vehicle tracking with camera and radar sensor," in *ICCAS 2017 - 2017 17th International Conference on Control, Automation and Systems - Proceedings*, vol. 2017-October, pp. 1075–1077, IEEE Computer Society, Dec. 2017. 17th International Conference on Control, Automation and Systems, ICCAS 2017 ; Conference date: 18-10-2017 Through 21-10-2017.

[22] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *CoRR*, vol. abs/1602.00763, 2016.

[23] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017.

[24] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4696–4704, 2015.

[25] S.-Y. Chu and M.-S. Lee, "Mt-detr: Robust end-to-end multimodal detection with confidence fusion," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5241–5250, 2023.

[26] H. Park, Y. J. Yoo, and N. Kwak, "MC-GAN: multi-conditional generative adversarial network for image synthesis," *CoRR*, vol. abs/1805.01123, 2018.

[27] V. John and S. Mita, "Rvnet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments," in *Pacific-Rim Symposium on Image and Video Technology*, 2019.

[28] S. Chang, Y. Zhang, F. Zhang, X. Zhao, S. Huang, Z. Feng, and Z. Wei, "Spatial attention fusion for obstacle detection using mmwave radar and vision sensor," *Sensors*, vol. 20, no. 4, 2020.

[29] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," *CoRR*, vol. abs/2005.07431, 2020.

[30] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," *CoRR*, vol. abs/2005.07431, 2020.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[33] S. Ji, S. V. N. Vishwanathan, N. Satish, M. J. Anderson, and P. Dubey, "Blackout: Speeding up recurrent neural network language models with very large vocabularies," 2015.

[34] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019.

[35] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3d object detection from multi-view images via 3d-to-2d queries," *CoRR*, vol. abs/2110.06922, 2021.

[36] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," 2022.

[37] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," 2022.

[38] R. Xu, H. Xiang, X. Xia, X. Han, J. Liu, and J. Ma, "OPV2V: an open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," *CoRR*, vol. abs/2109.07644, 2021.