Full-Velocity Radar Returns by Radar-Camera Fusion

Yunfei Long[1], Daniel Morris[1], Xiaoming Liu[1],
Marcos Castro[2], Punarjay Chakravarty[2], and Praveen Narayanan[2]
[1]Michigan State University, [2]Ford Motor Company
{longyunf,dmorris,liuxm}@msu.edu, {mgerard8,pchakra5,pnaray11}@ford.com

Abstract

A distinctive feature of Doppler radar is the measurement of velocity in the radial direction for radar points. However, the missing tangential velocity component hampers object velocity estimation as well as temporal integration of radar sweeps in dynamic scenes. Recognizing that fusing camera with radar provides complementary information to radar, in this paper we present a closed-form solution for the point-wise, full-velocity estimate of Doppler returns using the corresponding optical flow from camera images. Additionally, we address the association problem between radar returns and camera images with a neural network that is trained to estimate radar-camera correspondences. Experimental results on the nuScenes dataset verify the validity of the method and show significant improvements over the state-of-the-art in velocity estimation and accumulation of radar points.

1. Introduction

Radar is a mainstream automotive 3D sensor, and along with LiDAR and camera, is used in perception systems for driving assistance and autonomous driving [34, 18, 2]. Unlike LiDAR, radar has been widely installed on existing vehicles due to its relatively low cost and small sensor size, which makes it an easy fit into various vehicles without changing their appearance. Thus, advances in radar vision systems have potential to make immediate impact on vehicle safety. Recently, with the release of a couple of autonomous driving datasets with radar data included, *e.g.*, Oxford Radar RobotCar [1] and nuScenes [5], there is great interest in the community to explore how to leverage radar data in various vision tasks such as object detection [24, 37].

In addition to measuring 3D positions, radar has the special capability of obtaining radial velocity of returned points based on the Doppler effect. This extra capability is a significant advantage over other 3D sensors like LiDAR, enabling, for instance, instantaneous moving object detection. However, due to the inherently ambiguous mapping from radial velocity to full velocity, using radial velocity directly to account for the real movement of radar points is inadequate and sometimes misleading. Here, the full velocity denotes the actual velocity of radar points in 2D or 3D space. While radial velocity can well approximate full velocity when a point is moving away from or towards the radar, these two can be very different when the point is moving in the non-radial directions. An extreme case occurs for objects moving tangentially as these will have zero radial velocity regardless of target speed. Therefore, acquiring point-wise full velocity instead of radial velocity is crucial to reliably sense the motion of surrounding objects.

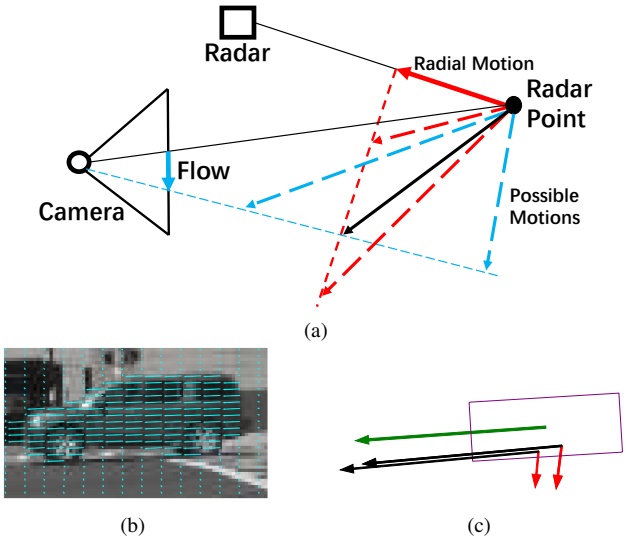


Figure 1: (a) Full motion cannot be determined with a single sensor: all motions ending on the blue dashed line (*i.e.*, blue dashed arrows) map to the same optical flow and all motions terminated on the red dashed line (*i.e.*, red dashed arrows) fit the same radial motion. However, with a radar-camera pair, the full motion can be uniquely decided: only the motion drawn in black satisfies both optical flow and radial motion. (b) Optical flow in the camera-image and (c) a bird's-eye view of the observed vehicle. This shows measured radar points with radial velocity (red), our predicted point-wise, full velocity (black), and ground truth full velocity of the vehicle (green).

Apart from measuring the velocity of objects, another important application of point-wise velocity is the accumulation of radar points. Radar returns from a single frame are much sparser than LiDAR in both azimuth and elevation, *e.g.*, typically LiDAR has an azimuth resolution $10\times$ higher than radar [37]. Thus, it is often essential to accumulate multiple prior radar frames to acquire sufficiently dense point clouds for downstream tasks, *e.g.*, object detection [26, 6, 7]. To align radar frames, in addition to compensating egomotion, we shall consider the motion of moving points in consecutive frames, which can be estimated by point-wise velocity and time of movement. As the radial velocity does not reflect the true motion, it is desirable to have point-wise full velocity for point accumulation.

To solve the aforementioned dilemma of radial velocity, we propose to estimate point-wise full velocity of radar returns by fusing radar with a RGB camera. Specifically, we derive a closed-form solution to infer point-wise full velocity from radial velocity as well as associated projected image motion obtained from optical flow. As shown in Fig. 1, constraints imposed by optical flow resolve the ambiguities of radial-full velocity mapping and lead to a unique and closed-form solution for full velocity. Our method can be considered as a way to enhance raw radar measurement by upgrading point-wise radial velocity to full velocity, laying the groundwork for improving radar-related tasks, *e.g.*, velocity estimation, point accumulation and object detection.

Moreover, a prerequisite for our closed-form solution is the association between moving radar points and image pixels. To enable a reliable association, we train a neural network to predict radar-camera correspondences as well as discerning occluded radar points. Experimental results demonstrate that the proposed method improves point-wise velocity estimates and their use for object velocity estimation and radar point accumulation.

In summary, the main contributions of this work are:

- We define a novel research task for radar-camera perception systems, *i.e.*, estimating point-wise full velocity of radar returns by fusing radar and camera.
- We propose a novel closed-form solution to infer full radar-return velocity by leveraging the radial velocity of radar points, optical flow of images, and the learned association between radar points and image pixels.
- We demonstrate state-of-the-art (SoTA) performance in object velocity estimation, radar point accumulation, and 3D object localization.

2. Related Works

Application of Radar in Vision Radar data differs from LiDAR data in various aspects [4]. In addition to the popular point representation (also named radar target [27]), an analogy to LiDAR points, there are other radar data representations containing more raw measurements, *e.g.*, range-azimuth image and spectrograms, which have been applied in tasks such as activity classification [33], detection [20], and pose estimation [30]. Our method is based on radar points, with the format available in the nuScenes dataset [5].

The characteristics of radar have been explored to complement other sensors. The Doppler velocity of radar points is used to distinguish moving targets. For example, RSS-Net [15] uses radial velocity as a motion cue for image semantic segmentation. Chadwick *et al.* [6] use radial velocity to detect distant moving vehicles—difficult to detect with only images. Fritsche *et al.* [10] combine radar with LiDAR for measurement under poor visibility. With a longer detection range than LiDAR, radar is also deployed with LiDAR to better detect far objects [37].

The *sparsity* of radar makes it difficult to directly apply well-developed techniques for LiDAR on radar [20, 24]. For example, Danzer *et al.* [8] adopt PointNets [28] on radar points for 2D car detection, while sparsity limits it to large objects like cars. Similar to LiDAR-camera depth completion [12, 13], Long *et al.* [22] develop radar-camera depth completion by learning a probabilistic mapping from radar returns to images. To obtain denser radar points, Lombacher *et al.* [21] use occupancy grid [9] to accumulate radar frames. Yet, the method assumes a static scene and cannot cope with moving objects. Radar points are projected on images and represented as regions near projected points, such as vertical bars [26] and circles [6, 7], to account for uncertainty of projection due to measurement error. While accumulating radar frames is desirable, without reliably compensating object motion, these methods need to carefully decide the number of frames to trade off between the gain in accumulation and loss in accuracy due to delay [26]. Our estimated point-wise velocity can compensate object motion and realize more accurate accumulation.

Velocity Estimation in Perception Systems Researchers have used monocular videos [3] or radial velocity of radar points to estimate *object-wise* velocity. With only radar data of a single frame, Kellner *et al.* [16, 17] compute full velocity of moving vehicles from radial velocities and azimuth angles of at least two radar hits. However, for a robust solution, the method requires that 1) radar captures more radar hits on each object, 2) radar points have significantly different azimuth angles and 3) object points are clustered before velocity estimation [16, 32, 31]. Obviously due to sparsity of radar in a single frame, it is difficult to obtain at least two radar hits on distant vehicles, let alone objects of smaller sizes. Also, it is common that radar points on the same object, *e.g.*, a distant or small object, have similar azimuth.

Recognizing the density and accuracy limitation of radar, researchers fuse radar with other sensors, *e.g.*, LiDAR and camera, for object-wise velocity estimation. Specifically,

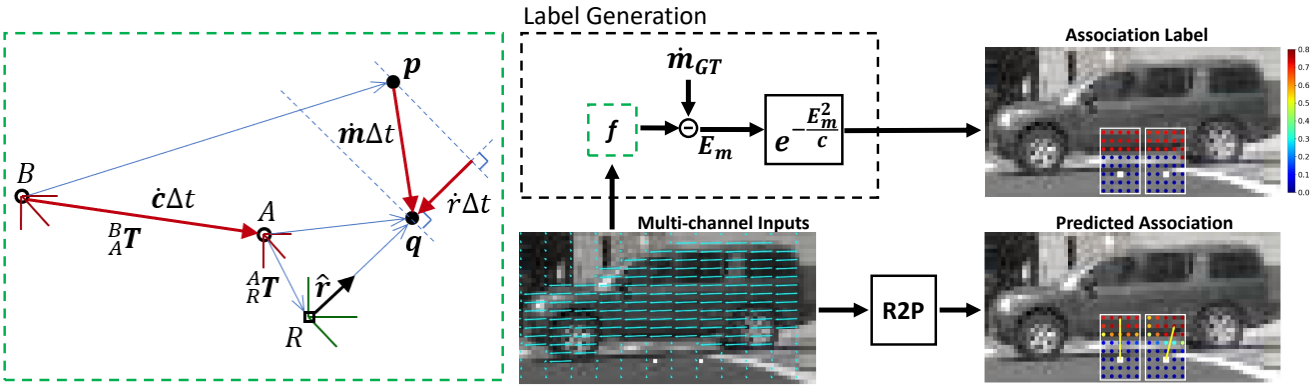


Figure 2: **Full velocity estimation and learning to associate radar points to camera pixels**. (a) A 3D point, $\boldsymbol{p}$, is observed by a camera at B. A short interval, Δt, later, the point has moved by $\dot{\boldsymbol{m}}\Delta t$ to $\boldsymbol{q}$ while the camera has moved by $\dot{\boldsymbol{c}}\Delta t$ to A. At the same time, the radar measures both the position of $\boldsymbol{q}$ and the radial speed $\dot{r}$, which is the radial component of $\dot{\boldsymbol{m}}$. Using radial speed $\dot{r}$ and the associated optical flow of $\boldsymbol{q}$ in images, we derive a closed-form equation (denoted as $\boldsymbol{f}()$) to estimate $\boldsymbol{q}$'s full velocity $\dot{\boldsymbol{m}}$. (b) As the closed-form solution requires point-wise association of two sensors, we train a Radar-2-Pixel (R2P) network to take a multi-channel input and predict the association probabilities for pixels within a neighborhood of the raw projection (white dot) obtained via known pose ${}^{A}_{R}\boldsymbol{T}$. A pixel with the highest probability (yellow arrow) is deemed as the associated pixel of a radar point. To obtain labels for training R2P, our label generation module uses $\boldsymbol{f}()$ to compute velocities of all neighboring pixels, then calculates velocity error E_m by using the ground truth velocity $\dot{\boldsymbol{m}}_{GT}$, and finally obtains association probabilities of these neighbors based on E_m.

existing techniques [38, 36, 19] for images or LiDAR are employed to obtain preliminary detections. Radar data, including radial velocity, once associated with the initial detections, are used as additional cues to predict full velocities of objects. For instance, in RadarNet [37] temporal point clouds of radar and LiDAR, modeled as voxels, are used to acquire initial detections and their motions. Object motion direction is used to resolve the ambiguities in radar-point association by back-projecting their radial velocities on the motion direction. Yet, a sequence of LiDAR frames is required to obtain the initial detection and motion estimation.

CenterFusion [24] integrates radar with camera for object-wise velocity estimation. Well-developed image-based detector is applied to extract preliminary boxes. After associating radar points with detections, the method combines radar data, radial velocity and depth, with image features within detected regions to regress a full velocity per detection. However, without a closed-form solution, the mapping from radial to full velocity needs to be learned from a great number of labeled data. In contrast, we present a point-wise *closed-form solution* for full-velocity estimation of radar points, without performing object detection. To our knowledge, there is no prior method able to perform point-wise full-velocity estimation for radar returns.

3. Proposed Method

We consider the case of a camera and radar rigidly attached to a moving platform, *e.g.*, a vehicle, observing moving objects in the environment. In this section we develop equations relating optical flow measurements in the camera to position and velocity measurements made by the radar.

3.1. Physical Configuration and Notation

The physical configuration of our camera and radar measurements is illustrated in Fig. 2(a). Three coordinate systems are shown: A and B specifying camera poses and R specifying a radar pose. The camera at B observes a 3D point $\boldsymbol{p}$. A short interval later, Δt, the point has moved to $\boldsymbol{q}$, the camera to A and the radar to R, and both the camera and radar observe the target point $\boldsymbol{q}$. These 3D points are specified by 4-dim homogeneous vectors, and when needed, a left-superscript specifies the coordinate system in which it is specified, *e.g.*, ${}^{A}\boldsymbol{q}$ indicates a point relative to a coordinate system A. The target velocity, $\dot{\boldsymbol{m}}$, and camera velocity $\dot{\boldsymbol{c}}$ are specified by 3-dim vectors, again optionally with a left superscript to specify a coordinate system.

Coordinate transformations, containing both a rotation and translation, are specified by 4×4 matrices, such as ${}^{B}_{A}\boldsymbol{T}$, which transforms points from the left-subscript coordinate system to the left-superscript coordinate system. In this case we transform a point from A to B with:

$$ {}^{B}\boldsymbol{q} = {}^{B}_{A}\boldsymbol{T}\ {}^{A}\boldsymbol{q}. \tag{1} $$

Only the rotational component of these transformations is needed to transform velocities. For example, ${}^{A}\dot{\boldsymbol{m}}$ is transformed to ${}^{B}\dot{\boldsymbol{m}}$ by the 3×3 rotation matrix ${}^{B}_{A}\boldsymbol{R}$:

$$ {}^{B}\dot{\boldsymbol{m}} = {}^{B}_{A}\boldsymbol{R}\ {}^{A}\dot{\boldsymbol{m}}. \tag{2} $$

A vector with a right subscript, *e.g.*, $\boldsymbol{p}_i$, indicates the i'th element of $\boldsymbol{p}$, while a right subscript of "1:3" puts the first 3

elements in a 3-dim vector. For a matrix, the right subscript indicates the row. Thus ${}^{B}_{A}\boldsymbol{R}_i$ is a 1×3 row vector containing its i-th row. A right superscript "^{T}" is a matrix transpose.

The projections of points $\boldsymbol{p}$ and $\boldsymbol{q}$ are specified in either undistorted raw pixel coordinates, *e.g.*, (x_q, y_q) or their normalized image coordinates (u_q, v_q) given by:

$$u_q = (x_q - c_x)/f_x, \qquad v_q = (y_q - c_y)/f_y. \tag{3}$$

Here c_x, c_y, f_x, f_y are intrinsic camera parameters, while the right subscript of the pixel refers to the point being projected. Vectors for 3D points can be expressed in terms of the normalized image coordinates:

$${}^{A}\boldsymbol{q} = \begin{pmatrix} u_q d_q \\ v_q d_q \\ d_q \\ 1 \end{pmatrix} \quad \text{and} \quad {}^{B}\boldsymbol{p} = \begin{pmatrix} u_p d_p \\ v_p d_p \\ d_p \\ 1 \end{pmatrix}. \tag{4}$$

Here d_q and d_p are depths of points ${}^{A}\boldsymbol{q}$ and ${}^{B}\boldsymbol{p}$ respectively.

We assume dense optical flow is available that maps target pixel coordinates observed in A to B as follows:

$$\text{Flow}\left((u_q, v_q)\right) \rightarrow (u_p, v_p). \tag{5}$$

Further, we assume the following are known: camera motion, ${}^{B}_{A}\boldsymbol{T}$, relative radar pose, ${}^{A}_{R}\boldsymbol{T}$, and intrinsic parameters.

3.2. Full-Velocity Radar Returns

The Doppler velocity measured by a radar is just one component of the three-component, full-velocity vector of an object point. Here our goal is to leverage optical flow from a synchronized camera to augment radar and estimate this full-velocity vector for each radar return.

3.2.1 Relationship of Full Velocity to Radial Velocity

The target motion from $\boldsymbol{p}$ to $\boldsymbol{q}$ is modeled as constant velocity, $\dot{\boldsymbol{m}}$, over time Δt, such that

$$\dot{\boldsymbol{m}} = \frac{\boldsymbol{q}_{1:3} - \boldsymbol{p}_{1:3}}{\Delta t}. \tag{6}$$

Our goal is to estimate the full target velocity, $\dot{\boldsymbol{m}}$. Radar provides an estimate of the target position, $\boldsymbol{q}$, but not the previous target location $\boldsymbol{p}$. Radar also provides the signed radial speed, $\dot{r}$, which is one component of $\dot{\boldsymbol{m}}$. In the nuScenes dataset $\dot{r}$ is given by:

$$\dot{r} = \hat{\boldsymbol{r}}^{\mathsf{T}}\dot{\boldsymbol{m}}. \tag{7}$$

Here $\hat{\boldsymbol{r}}$ is the unit-norm vector along the direction to the target ${}^{R}\boldsymbol{q}$. Note that this equation is coordinate-invariant, and could be equally written in A using ${}^{A}\hat{\boldsymbol{r}}$ and ${}^{A}\dot{\boldsymbol{m}}$. Now Eq. (7) is actually the egomotion-corrected Doppler speed. The raw Doppler speed, $\dot{r}_{raw}$, is the radial component of the *relative* velocity between target and sensor, $\dot{\boldsymbol{m}} - \dot{\boldsymbol{c}}$, and this constraint is given by:

$$\dot{r}_{raw} = \hat{\boldsymbol{r}}^{\mathsf{T}}(\dot{\boldsymbol{m}} - \dot{\boldsymbol{c}}), \tag{8}$$

where $\dot{\boldsymbol{c}}$ is the known ego-velocity. Either Eq. (7) or (8) can be used in our formulation, depending on whether $\dot{r}$ or $\dot{r}_{raw}$ is available from the radar.

3.2.2 Relationship of Full Velocity to Optical Flow

In solving the velocity constraints, we first identify the known variables. The radar measures ${}^{R}\boldsymbol{q}$, and transforming this we obtain ${}^{A}\boldsymbol{q} = {}^{A}_{R}\boldsymbol{T}\ {}^{R}\boldsymbol{q}$ which contains d_q as the third component. Image coordinates (u_q, v_q) are obtained by projection, and using optical flow in Eq. (5), we can also obtain the (u_p, v_p) components of ${}^{B}\boldsymbol{p}$. The key parameter we do not know from this is the depth, d_p, in B.

Next we eliminate this unknown depth from our constraints. Eq. (6) can be rearranged and each component expressed in frame B:

$${}^{B}\boldsymbol{p}_{1:3} = {}^{B}\boldsymbol{q}_{1:3} - {}^{B}_{A}\boldsymbol{R}\ {}^{A}\dot{\boldsymbol{m}}\Delta t, \tag{9}$$

where the second term on the right is the transformation of the target motion into B coordinates. The third row of this equation is an expression for d_p:

$$d_p = {}^{B}\boldsymbol{q}_3 - {}^{B}_{A}\boldsymbol{R}_3\ {}^{A}\dot{\boldsymbol{m}}\Delta t. \tag{10}$$

Substituting this for d_p, and the components of ${}^{B}\boldsymbol{p}$ from Eq. (4), into the first two rows of Eq. (9), we obtain

$$\begin{bmatrix} u_p({}^{B}\boldsymbol{q}_3 - {}^{B}_{A}\boldsymbol{R}_3\ {}^{A}\dot{\boldsymbol{m}}\Delta t) \\ v_p({}^{B}\boldsymbol{q}_3 - {}^{B}_{A}\boldsymbol{R}_3\ {}^{A}\dot{\boldsymbol{m}}\Delta t) \end{bmatrix} = \begin{bmatrix} {}^{B}\boldsymbol{q}_1 - {}^{B}_{A}\boldsymbol{R}_1\ {}^{A}\dot{\boldsymbol{m}}\Delta t \\ {}^{B}\boldsymbol{q}_2 - {}^{B}_{A}\boldsymbol{R}_2\ {}^{A}\dot{\boldsymbol{m}}\Delta t \end{bmatrix}, \tag{11}$$

and rearrange to give two constraints on the full velocity:

$$\begin{bmatrix} {}^{B}_{A}\boldsymbol{R}_1 - u_p\ {}^{B}_{A}\boldsymbol{R}_3 \\ {}^{B}_{A}\boldsymbol{R}_2 - v_p\ {}^{B}_{A}\boldsymbol{R}_3 \end{bmatrix} {}^{A}\dot{\boldsymbol{m}} = \begin{bmatrix} \left({}^{B}\boldsymbol{q}_1 - u_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \\ \left({}^{B}\boldsymbol{q}_2 - v_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \end{bmatrix}. \tag{12}$$

3.2.3 Full-Velocity Solution

We obtain three constraints on the full velocity, ${}^{A}\dot{\boldsymbol{m}}$, from Eq. (12) and by converting Eq. (7) to A coordinates. Combining these we obtain:

$$\begin{bmatrix} {}^{B}_{A}\boldsymbol{R}_1 - u_p\ {}^{B}_{A}\boldsymbol{R}_3 \\ {}^{B}_{A}\boldsymbol{R}_2 - v_p\ {}^{B}_{A}\boldsymbol{R}_3 \\ {}^{A}\hat{\boldsymbol{r}}^{\mathsf{T}} \end{bmatrix} {}^{A}\dot{\boldsymbol{m}} = \begin{bmatrix} \left({}^{B}\boldsymbol{q}_1 - u_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \\ \left({}^{B}\boldsymbol{q}_2 - v_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \\ \dot{r} \end{bmatrix}. \tag{13}$$

Then inverting the 3×3 coefficient of ${}^{A}\dot{\boldsymbol{m}}$ gives a closed form solution for the full velocity:

$${}^{A}\dot{\boldsymbol{m}} = \begin{bmatrix} {}^{B}_{A}\boldsymbol{R}_1 - u_p\ {}^{B}_{A}\boldsymbol{R}_3 \\ {}^{B}_{A}\boldsymbol{R}_2 - v_p\ {}^{B}_{A}\boldsymbol{R}_3 \\ {}^{A}\hat{\boldsymbol{r}}^{\mathsf{T}} \end{bmatrix}^{-1} \begin{bmatrix} \left({}^{B}\boldsymbol{q}_1 - u_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \\ \left({}^{B}\boldsymbol{q}_2 - v_p\ {}^{B}\boldsymbol{q}_3\right)/\Delta t \\ \dot{r} \end{bmatrix}. \tag{14}$$

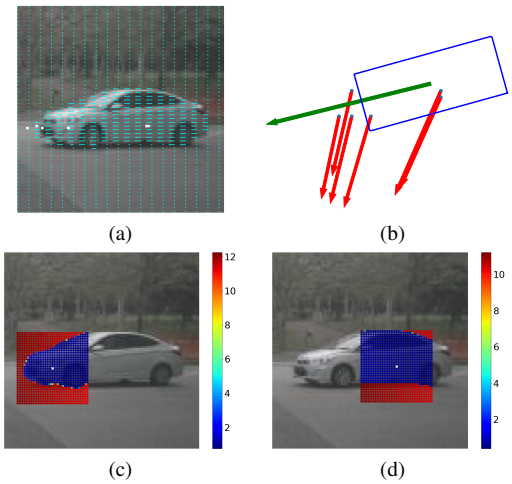


Figure 3: (a) Optical flow; (b) Bird's-eye view of GT bounding box, radial velocity (red) and GT velocity (green); (c) and (d) show E_m, computed by using Eq. (16), for two radar projections (white square) over 41×41 pixel regions, respectively. For radar hits reflected from the vehicle, E_m is small for neighboring pixels on the car and large on the background.

Recall in Fig. 1(a) the red/blue dashed lines show the velocity constraints from radar/flow. The solution of Eq. (14) is the full velocity that is consistent with both constraints. We note that this can handle moving sensors, although Fig. 1(a) shows the case of a stationary camera for simplicity. Further, if we set $\Delta t < 0$, Eq. (14) also applies to the case that the point shifts from $\boldsymbol{q}$ to $\boldsymbol{p}$ as the camera moves from A to B. And one limitation is that Eq. (14) cannot estimate full velocity for radar points occluded in the camera view, although we can typically identify those occlusions.

3.3. Image Pixels and Radar Points Association

Our solution for point-wise velocity in Eq. (14) assumes that we know the pixel coordinates (u_q, v_q) of the radar-detected point, ${}^R\boldsymbol{q}$. It appears straightforward to obtain this pixel correspondence by projecting a radar point onto the image using the known radar-image coordinate transformation, ${}^A_R\boldsymbol{T}$. We refer to this corresponding pixel as "raw projection". However, there are a number of reasons why raw projection of radar points into an image is inaccurate. Radar beam-width typically subtends a few degrees and is large relative to a pixel, resulting in low resolution target location in both azimuth and elevation. Also, a radar displaced from a camera can often see behind an object, as viewed by the camera, and when these returns are projected onto an image they incorrectly appear to correspond to the foreground occluding object. Using flow from an occluder or an incorrectly associated object pixel may result in incorrect full-velocity estimation. To address these issues with raw projection, we train a neural network model, termed Radar-2-Pixel (R2P) network, to estimate associated radar pixels in the neighborhood of raw projection and identify occluded radar points. Similar models have been applied to image segmentation [14] and radar depth enhancement [22].

3.3.1 Model Structure

Our method estimates association probabilities (ranging from 0 to 1) between a moving radar point and a set of pixels in the neighborhood of its raw projection. The R2P network is an encoder-decoder structure with inputs and outputs of image resolution. Stored in 8 channels, the input data include image, radar depth map (with depth on raw projections) and optical flow. The output has N channels, representing predicted association probability for N pixel neighbors. The association between the radar point, ${}^A\boldsymbol{q}$, and the k-th neighbor of raw projection (x, y) is stored in $A(x, y, k)$, where $k = 1, 2, ..., N$.

3.3.2 Ground Truth Velocity of Moving Radar Points

The nuScenes [5] provides the GT (ground truth) velocity of object bounding boxes. We associate radar hits on an object to its labeled bounding box, and assign the velocity of the box to its associated radar points. The association is determined based on two criteria: 1) in radar coordinates, the distance between radar points and associated box is smaller than a threshold T_d; and 2) the percentage error between the radial velocity of a radar point and the radial component of the velocity of associated box is smaller than a threshold T_p.

3.3.3 Generating Association Labels

We can project a radar point expressed in corresponding camera coordinates, ${}^A\boldsymbol{q}$, to pixel coordinates (u_q, v_q), but as mentioned before, often this image pixel does not correspond to the radar return. Our proposed solution is to search in a neighboring region around (u_q, v_q) for a pixel whose motion is consistent with the radar return. This neighborhood search is shown in Fig. 2. If a pixel is found, then we correct the 3D radar location ${}^A\boldsymbol{q}$ to be consistent with this pixel, otherwise we mark this radar return as occluded.

We learn this radar-to-pixel association and correction by training the R2P network. We generate true association score between a radar point and a pixel according to the compatibility between the true velocity and the optical flow at that pixel: high compatibility indicates high association. To quantify the compatibility, assuming a pixel is associated with a radar point, we compute a hypothetical full velocity for the radar point by using the optical flow of that pixel according to Eq. (14). The flow is considered compatible if the hypothetical velocity is close to the GT velocity. Specifically, the hypothetical velocity can be computed as

$$ {}^A\dot{\boldsymbol{m}}_{est}(x, y, k) = \boldsymbol{f}\left(\breve{u}_q, \breve{v}_q, \breve{u}_p, \breve{v}_p, d_q, \dot{r}, {}^B_A\boldsymbol{T}, {}^A_R\boldsymbol{T}\right), \quad (15) $$

where $k = 1, \cdots, N$, $\boldsymbol{f}(\cdot)$ is the function to solve full velocity via Eq. (14), and (x, y) is the raw projection of the radar point. Note that $\breve{u}_q = u_q\,[x + \Delta x(k), y + \Delta y(k)]$, $\breve{v}_q$ is defined similarly, and $[\Delta x(k), \Delta y(k)]$ is the coordinate offset from raw projection to the k-th neighbor. Using flow, Eq. (5), we obtain $(\breve{u}_p, \breve{v}_p)$ from $(\breve{u}_q, \breve{v}_q)$.

Second, we calculate the L_2 norm of errors between ${}^A\dot{\boldsymbol{m}}_{est}(x, y, k)$ and ground truth velocity ${}^A\dot{\boldsymbol{m}}_{GT}(x, y)$ by

$$E_m(x, y, k) = \|{}^A\dot{\boldsymbol{m}}_{est}(x, y, k) - {}^A\dot{\boldsymbol{m}}_{GT}(x, y)\|_2. \quad (16)$$

Fig. 3 shows examples of E_m for two radar hits on a car.

Finally, we transform E_m to an association score with

$$L(x, y, k) = e^{-\frac{E_m^2(x,y,k)}{c}}, \quad (17)$$

where L is used as a label for association probability between a radar and its k-th neighbor. Note that L increases with decreasing E_v, and c is a parameter adjusting the tolerance of velocity errors when converting errors to association. We use the cross entropy loss to train the model.

3.3.4 Estimate Association and Identify Occlusion

With a trained model, we can estimate association probability between radar points and N pixels around their raw projections (x, y), *i.e.*, $A(x, y, k)$. Among the N neighbors, the radar return velocity may be compatible with a number of pixels, and we select the pixel with the maximum association, A_{max}, as the neighbor ID k_{max}:

$$k_{max} = \arg\max_k [A(x, y, k)]. \quad (18)$$

If A_{max} is equal or larger than a threshold T_a, we estimate the associated pixel as $[x + \Delta x(k_{max}), y + \Delta y(k_{max})]$. Otherwise there is no associated pixels in the neighborhood, and an occlusion is identified.

4. Experimental Results

4.1. Comparison of Point-wise Full Velocity

To the best of our knowledge, there is no existing method estimating *point-wise* full velocity for radar returns. Thus, we use point-wise radial velocity from raw radar returns as the baseline to compare with our estimation. We extract data from the nuScenes Object Detection Dataset [5], with 6432, 632, and 2041 samples in training, validation and testing set, respectively. Each sample consists of a radar scan and two images for optical flow computation, *i.e.*, one image synchronizing with the radar and the other is a neighboring image frame. The optical flow is computed by the RAFT model [35] pre-trained on KITTI [11]. The R2P network is an U-Net [29, 23] with five levels of resolutions and 64 channels for intermediate filters. The neighborhood skips every other pixel, and its size (in pixels) is (left: 4, right: 4, top: 10, bottom: 4) and an example of the neighborhood is illustrated in Fig. 2(b). The threshold of association scores T_a is 0.3. Parameters associating radar points with GT bounding box are set as $T_d = 0.5$m and $T_p = 20\%$. Parameter c in Eq. (17) is 0.36. To obtain GT point-wise velocity, based on the criteria in Sec. 3.3.2, we first associate moving radar points to GT detection boxes, whose GT velocity is assigned to associated points as their GT velocity. The GT velocity of bounding boxes is estimated from GT center positions in neighboring frames with timestamps.

Mean Error (STD) (m/s)	Ours (R2P Network)	Ours (Raw Projection)	Baseline
Full Velocity	**0.433** (**0.608**)	0.577 (1.010)	1.599 (2.054)
Tangential Comp.	**0.322** (**0.610**)	0.472 (1.024)	1.536 (2.083)
Radial Comp.	0.205 (0.196)	0.205 (0.196)	0.205 (0.196)

Table 1: Comparison of point-wise velocity error of our methods and the baseline (raw radial velocity).

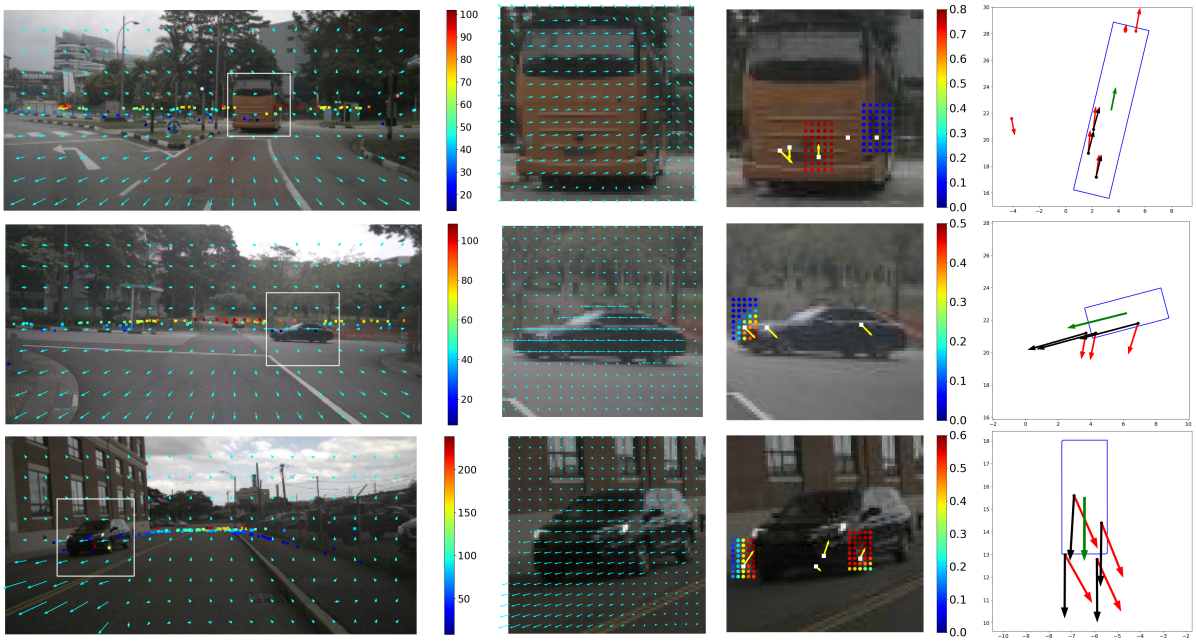
Methods	Error (m/s)
Ours	**0.451**
CenterFusion [24]	0.826

Table 2: Comparison of object-wise velocity errors. For a fair comparison we inherit the same set of detected objects from [24].

Tab. 1 shows the average velocity error for moving points. The proposed method achieves substantially more accurate velocity estimation than the baseline. For instance, the error of our tangential component is only 21% of that of the baseline. We also have much smaller standard deviation, indicating more stable estimates. In addition, we list in Tab. 1 velocity error of our method using raw radar projection for radar-camera association. Results show that, compared with using raw projection, using R2P network achieves higher estimation accuracy. Fig. 4 illustrates qualitative results of our point-wise velocity estimation.

4.2. Comparison of Object-wise Velocity

Although there are no existing methods for point-wise velocity estimation for radar, a related work, CenterFusion [24], estimates *object-wise* full velocity via object detection with image and radar inputs. To fairly compare with CenterFusion, we convert our point-wise velocity to object-wise velocity. Specifically, we use the average velocity of radar points associated with the same detected box as our estimate of object velocity. Points are associated with detected boxes according to distance. Note the point-wise velocity to object-wise velocity conversion is straightforward for comparison purposes, and there would be more advanced approaches to integrate point-wise full velocities in a detection network, which is beyond the scope of this work. Tab. 2 shows that with our estimated full velocity, the velocity estimation for objects is significantly improved.



(a) (b) (c) (d)

Figure 4: Visualization of point-wise velocity estimation: (a) depth of all measured radar returns as well as flow, (b) optical flow in the white box region, (c) association scores around the selected radar projections as well as predicted mapping from raw radar projections to image pixels (yellow arrow) and (d) radial velocity (red), estimated full velocity (black) and GT velocity (green) in bird's-eye view.

4.3. Radar Point Accumulation

Accumulating radar points over time can overcome the sparsity of radar hits acquired in a single sweep, achieving dense point cloud for objects and thus allowing techniques designed for processing LiDAR points to be applicable for radar. The point-wise velocity estimate makes it possible to compensate the motion of dynamic objects appearing in a temporal sequence of measurements for accumulation. Specifically, for a moving radar point (with estimated velocity $\dot{\boldsymbol{m}}$) in a previous frame i captured at time t_i, its motion from t_i to the time at the current frame, t_0, can be compensated by,

$$\boldsymbol{p_0} = \boldsymbol{p_i} + \dot{\boldsymbol{m}}(t_0 - t_i), \tag{19}$$

where $\boldsymbol{p_i}$ and $\boldsymbol{p_0}$ are the radar point coordinates at t_i and t_0 in radar coordinates of t_i. Then $\boldsymbol{p_0}$ is transformed to current radar coordinates by known egomotion from t_i to t_0.

Qualitative results Fig. 6 shows accumulated points of moving vehicles in radar coordinates. For comparison, we show accumulated radar points compensated by our estimated full velocity, compensated with radial velocity (baseline) and without motion compensation. Compared with the baseline and no motion compensation, our accumulated points are more consistent with the GT bounding boxes.

Quantitative results To quantitatively evaluate the accuracy of radar point accumulation, we use the mean distance

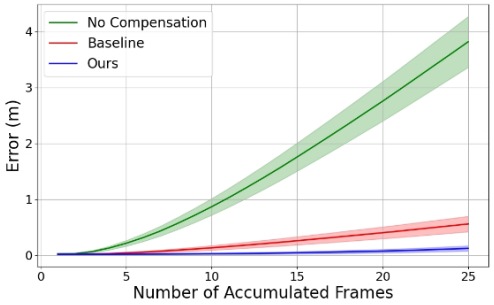
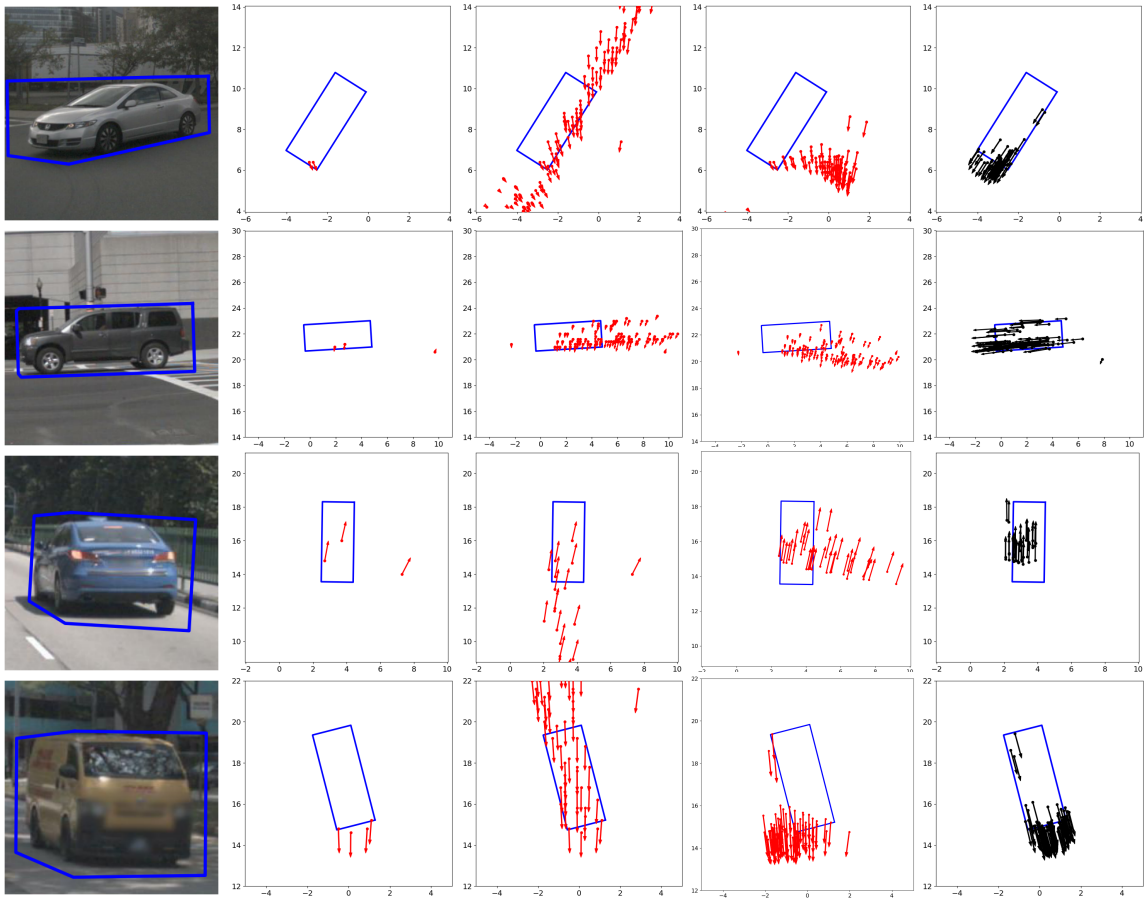


Figure 5: Error comparison when accumulating radar points from increasing number of frames. The lines represent mean error and shaded area $\pm 0.1\times$ STD. Our full velocity based accumulation outperforms the ones with radial velocity, or no compensation.

from accumulated points (of up to 25 frames) to their corresponding GT boxes as the accumulation error. This distance for points inside the box is zero, and outside it is the distance from the radar point to the closest point on the box's boundary. In Fig. 5, we compare the accumulation for our method, the baseline and accumulation without motion compensation. While error increases with the number of frames for all methods, our method has the lowest rate of error escalation.

Application of pose estimation To demonstrate the utility of accumulated radar points for downstream applications, we apply a pose estimation method, *i.e.*, BoxNet [25], on the accumulated 2D radar points via our full velocity and



(a) (b) (c) (d) (e)

Figure 6: Moving radar points are plotted with point-wise radial (red) and full (black) velocity, including image with bounding box (a), single-frame radar points in bird's-eye view (b), accumulated radar points from 20 frames without motion compensation (c), with radial velocity based compensation (d), and with our full-velocity based compensation (e). Our accumulated points are tightly surrounding the bounding box, which will benefit downstream tasks such as pose estimation and object detection.

Metric	Ours	Baseline
Center Error (m) ↓	**0.834**	0.997
Orientation Error (degree) ↓	**6.873**	7.517
IoU ↑	**0.546**	0.462

Table 3: Comparison of pose estimation performance: average error in center and orientation as well as Intersection over Union (IoU), by using BoxNet [25] on radar points accumulated using our velocity and the radial velocity as a baseline.

radial velocity (baseline), respectively. BoxNet takes pre-segmented 2D point clouds of an object as input and predicts a 2D bounding box with parameters as center position, length, width and orientation. We use accumulated radar points of 5702, 559 and 2001 moving vehicles with corresponding GT bounding boxes as training, validation and testing data, respectively. Tab. 3 shows our accumulated radar achieves higher accuracy than the baseline.

5. Conclusion

A drawback of Doppler radar has been that it provides only the radial component of velocity, which limits its utility in object velocity estimation, motion prediction and radar return accumulation. This paper addresses this drawback by presenting a closed-form solution to the full velocity of radar returns. It leverages optical flow constraints to upgrade radial velocity into full velocity. As part of this work, we use GT bounding-box velocities to supervise a network that predicts association corrections for the raw radar projections. We experimentally verify the effectiveness of our method and demonstrate its application on motion compensation for integrating radar sweeps over time.

This method developed here may apply to additional modalities such as full-velocity estimation from Doppler LiDAR and cameras.

Acknowledgement This work was supported by the Ford-MSU Alliance.

References

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar Dataset: A radar extension to the Oxford RobotCar Dataset. In *IEEE International Conference on Robotics and Automation*, pages 6433–6438, 2020. 1

[2] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *Proceeding of International Conference on Computer Vision*, Seoul, South Korea, October 2019. 1

[3] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3D object detection in monocular video. In *Proceeding of European Conference on Computer Vision*, Virtual, August 2020. 2

[4] Daniel Brodeski, Igal Bilik, and Raja Giryes. Deep radar detector. In *IEEE Radar Conference*, pages 1–6, 2019. 2

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 2, 5, 6

[6] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. In *IEEE International Conference on Robotics and Automation*, pages 8311–8317, 2019. 2

[7] Shuo Chang, Yifan Zhang, Fan Zhang, Xiaotong Zhao, Sai Huang, Zhiyong Feng, and Zhiqing Wei. Spatial attention fusion for obstacle detection using mmWave radar and vision sensor. *Sensors*, 20(4):956, 2020. 2

[8] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2D car detection in radar data with PointNets. In *IEEE Intelligent Transportation Systems Conference*, pages 61–66, 2019. 2

[9] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. 2

[10] Paul Fritsche, Björn Zeise, Patrick Hemme, and Bernardo Wagner. Fusion of radar, LiDAR and thermal information for hazard detection in low visibility environments. In *IEEE International Symposium on Safety, Security and Rescue Robotics*, pages 96–101, 2017. 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6

[12] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin-surface extrapolation at occlusion boundaries. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, Nashville, TN, June 2021. 2

[13] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, Long Beach, CA, January 2019. 2

[14] Michael Kampffmeyer, Nanqing Dong, Xiaodan Liang, Yujia Zhang, and Eric P Xing. ConnNet: A long-range relation-aware pixel-connectivity network for salient segmentation. *IEEE Transactions on Image Processing*, 28(5):2518–2529, 2018. 5

[15] Prannay Kaul, Daniele De Martini, Matthew Gadd, and Paul Newman. RSS-Net: Weakly-supervised multi-class semantic segmentation with FMCW radar. In *IEEE Intelligent Vehicles Symposium*, pages 431–436, 2020. 2

[16] Dominik Kellner, Michael Barjenbruch, Klaus Dietmayer, Jens Klappstein, and Jürgen Dickmann. Instantaneous lateral velocity estimation of a vehicle using Doppler radar. In *International Conference on Information Fusion*, pages 877–884, 2013. 2

[17] Dominik Kellner, Michael Barjenbruch, Jens Klappstein, Jürgen Dickmann, and Klaus Dietmayer. Instantaneous full-motion estimation of arbitrary objects using dual Doppler radar. In *IEEE Intelligent Vehicles Symposium*, pages 324–329, 2014. 2

[18] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020. 1

[19] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for LiDAR point clouds in autonomous driving: a review. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 3

[20] Teck-Yian Lim, Amin Ansari, Bence Major, Daniel Fontijne, Michael Hamilton, Radhika Gowaikar, and Sundar Subramanian. Radar and camera early fusion for vehicle detection in advanced driver assistance systems. In *Conference on Neural Information Processing Systems Workshops*, 2019. 2

[21] Jakob Lombacher, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Potential of radar for static object classification using deep learning methods. In *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4, 2016. 2

[22] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12507–12516, 2021. 2, 5

[23] Daniel Morris. A pyramid CNN for dense-leaves segmentation. In *Conference on Computer and Robot Vision*, pages 238–245. IEEE, 2018. 6

[24] Ramin Nabati and Hairong Qi. CenterFusion: Center-based radar and camera fusion for 3D object detection. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1527–1536, 2021. 1, 2, 3, 6

[25] Ehsan Nezhadarya, Yang Liu, and Bingbing Liu. BoxNet: A deep learning method for 2D bounding box estimation from bird's-eye view point cloud. In *IEEE Intelligent Vehicles Symposium*, pages 1557–1564, 2019. 7, 8

[26] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *Sensor Data Fusion: Trends, Solutions, Applications*, pages 1–7. IEEE, 2019. 2

[27] Andras Palffy, Jiaao Dong, Julian FP Kooij, and Dariu M Gavrila. CNN based road user detection using the 3D radar

cube. *IEEE Robotics and Automation Letters*, 5(2):1263–1270, 2020. 2

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241, 2015. 6

[30] Fabian Roos, Dominik Kellner, Jürgen Dickmann, and Christian Waldschmidt. Reliable orientation estimation of vehicles in high-resolution radar images. *IEEE Transactions on Microwave Theory and Techniques*, 64(9):2986–2993, 2016. 2

[31] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. A multi-stage clustering framework for automotive radar data. In *IEEE Intelligent Transportation Systems Conference*, pages 2060–2067, 2019. 2

[32] Johannes Schlichenmaier, Fabian Roos, Philipp Hügler, and Christian Waldschmidt. Clustering of closely adjacent extended objects in radar images using velocity profile analysis. In *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4, 2019. 2

[33] Mehmet Saygın Seyfioğlu, Ahmet Murat Özbayoğlu, and Sevgi Zubeyde Gürbüz. Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. *IEEE Transactions on Aerospace and Electronic Systems*, 54(4):1709–1723, 2018. 2

[34] Leo Stanislas and Thierry Peynot. Characterisation of the Delphi electronically scanning radar for robotics applications. In *Australasian Conference on Robotics and Automation*, pages 1–10, 2015. 1

[35] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419, 2020. 6

[36] Yutian Wu, Yueyu Wang, Shuwei Zhang, and Harutoshi Ogai. Deep 3D object detection networks using LiDAR data: A review. *IEEE Sensors Journal*, 21(2):1152–1171, 2020. 3

[37] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. RadarNet: Exploiting radar for robust perception of dynamic objects. In *European Conference on Computer Vision*, pages 496–512, 2020. 1, 2, 3

[38] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. 3