RC-AutoCalib: An End-to-End Radar-Camera Automatic Calibration Network

Van-Tin Luu[1], Yon-Lin Cai[1], Vu-Hoang Tran[2], Wei-Chen Chiu[1], Yi-Ting Chen[1], Ching-Chun Huang[1*]
[1]National Yang Ming Chiao Tung University, Taiwan
[2]Ho Chi Minh City University of Technology and Education, Vietnam
{tinery.ee12, yukitaka.10, walon, ychen, chingchun}@nycu.edu.tw
hoangtv@hcmute.edu.vn

Abstract

This paper presents a groundbreaking approach - the first online automatic geometric calibration method for radar and camera systems. Given the significant data sparsity and measurement uncertainty in radar height data, achieving automatic calibration during system operation has long been a challenge. To address the sparsity issue, we propose a Dual-Perspective representation that gathers features from both frontal and bird's-eye views. The frontal view contains rich but sensitive height information, whereas the bird's-eye view provides robust features against height uncertainty. We thereby propose a novel Selective Fusion Mechanism to identify and fuse reliable features from both perspectives, reducing the effect of height uncertainty. Moreover, for each view, we incorporate a Multi-Modal Cross-Attention Mechanism to explicitly find location correspondences through cross-modal matching. During the training phase, we also design a Noise-Resistant Matcher to provide better supervision and enhance the robustness of the matching mechanism against sparsity and height uncertainty. Our experimental results, tested on the nuScenes dataset, demonstrate that our method significantly outperforms previous radar-camera auto-calibration methods, as well as existing state-of-the-art LiDAR-camera calibration techniques, establishing a new benchmark for future research. The code is available at https://github.com/nycu-acm/RC-AutoCalib

1. Introduction

Radars and cameras are increasingly favored in advanced driver-assistance systems (ADAS) due to their cost-effectiveness and robust performance in diverse weather conditions. A critical research area in these systems involves data fusion and multi-modal calibration to ensure reliable functioning in real-world settings [16, 19, 37, 39, 44]. Conventional calibration techniques for 3D radar and cameras primarily focus on offline methods, which often rely on specialized calibration targets like checkerboards or corner reflectors, and are generally limited to the radar measurement plane [3, 4, 10–12, 29, 32]. These methods, while effective, require substantial time and manual effort, and they do not account for sensor displacements that can occur under normal operating conditions. This limitation underscores the necessity for online auto-calibration, which can dynamically adjust to changes over time.

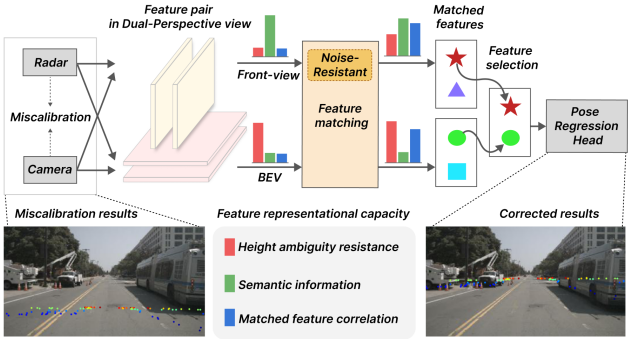
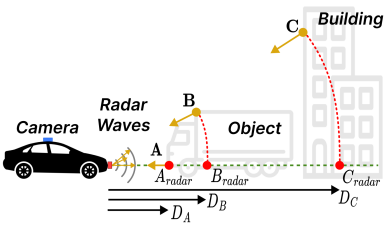


Figure 1. An overview of the proposed RC-AutoCalib method. The approach takes input from radar-camera miscalibration, representing it as feature pairs in Dual-perspective view. These feature representations are then enhanced through feature matching block, from which reliable features are selected to predict the rotation vector and translation.

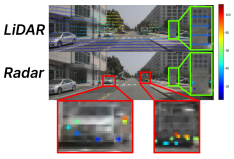
Online auto-calibration methods eliminate the need for calibration targets, focusing on matching natural features collected by radar and camera sensors. While this approach offers greater flexibility in real-world scenarios, exploration in this area remains limited, with no established benchmarks using publicly available datasets to date. Only the approach by Schöller et al. [26] utilizes deep learning to address the problem of online auto-calibration for radar and camera. However, their focus remains exclusively on rotational calibration, without addressing translational calibration.

In contrast, online auto-calibration methods for LiDAR and camera [8, 9, 18, 20, 21, 25, 27, 31, 35, 36, 41–43] have

*Corresponding author



(a) Elevation Ambiguity in Radar Point Clouds [19].



(b) Noisy depth measurements and lack of object structures.

Figure 2. Challenges of 3D Millimeter-Wave Radar. (a) The green dashed line represents the height plane the radar focuses on. Points $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ denote actual reflection positions, whereas $\mathbf{A}_{\text{radar}}$, $\mathbf{B}_{\text{radar}}$, and $\mathbf{C}_{\text{radar}}$ are the positions recorded by the radar. $\mathbf{D}_{\text{A}}$, $\mathbf{D}_{\text{B}}$, and $\mathbf{D}_{\text{C}}$ represent the recorded and noisy radar depths. (b) The top image shows a "LiDAR" depth map projected onto the camera plane, while the bottom image displays a "radar" depth map projected similarly. The red box highlights the issue where depths of points on the same object should be similar, yet significant variations are evident, indicating the presence of noise. Moreover, the green box shows a structural comparison: the "LiDAR" point cloud distinctly outlines the object's contour, while the "radar" point cloud fails to convey structural information.

been extensively explored by researchers and have demonstrated powerful capabilities. Most of these methods share a common concept of using RGB images and mis-calibrated LiDAR data as input, with the overall process divided into feature extraction, feature matching, and parameter regression. Instead of directly extracting features from RGB images [20, 27, 31, 42], some methods have modified the representation of RGB images. For instance, [18, 41] implicitly extract features in the form of semantics and edges, while [35, 43] transform RGB images into depth maps to achieve a unified representation consistent with 3D data. Overall, these methods project 3D data onto the frontal view to fuse with the camera information for further processing. The above-mentioned LiDAR-camera methods provide a reference and comparison for developing radar-camera auto-calibration. However, we find that relying solely on a single viewpoint for radar-camera auto-calibration makes achieving high accuracy challenging. As depicted in Fig. 2, the frontal depth map often contains noisy point cloud data due to uncertainty from the lack of radar height information. Additionally, radar data is inherently sparse and lacks object structures. When projecting radar point clouds onto the frontal view to form the depth map, the projected points tend to overlap and become even sparser. To mitigate these challenges, we have introduced a Dual-Perspective representation that leverages attention-based selection to extract more reliable features.

Furthermore, feature matching between radar and camera is a critical component for auto-calibration. Some methods [27, 35, 43] rely on concatenation followed by several convolutional layers to facilitate feature matching, while others [20] use cost volume to represent the correlation between the two sensors. However, traditional approaches depend solely on implicit supervision from the final calibration loss to guide the matching process. This lack of explicit identification of matched local pairs between sensors renders the calibration process indistinct. To address this, we have developed a Noise-Resistant Matcher that provides direct supervision for feature matching and correspondence finding.

Accordingly, as illustrated in Fig. 1, we propose RC-AutoCalib, an end-to-end network for automatic 3D radar and camera calibration, addressing the challenges of sparse and noisy radar data. To counteract these issues, we enhance a Dual-Perspective representation that integrates features from both the frontal view and the bird's eye view (BEV). The frontal view is prone to noise due to missing height information in radar point clouds, while the BEV provides more stable features, unaffected by this limitation. Our model includes a Selective Fusion Mechanism to discern and utilize beneficial features from each perspective. Additionally, we incorporate a Multi-Modal Cross-Attention Mechanism to focus on relevant areas in sparse radar point clouds. To improve calibration accuracy, we introduce Explicit Feature Matching Supervision with a Noise-Resistant Matcher, which helps the model identify and learn from correspondence points between radar and camera, filtering out noise in the process. Our results on the nuScenes dataset demonstrate significant improvements over existing LiDAR-camera calibration methods, as well as previous radar-camera auto-calibration approaches, setting a new benchmark for future research. In summary, our contributions are:

- We introduce RC-AutoCalib, an end-to-end network for calibrating 3D radar and cameras, featuring a Dual-Perspective representation that counters the height information limitations of 3D radar data. This network includes a novel Selective Fusion Mechanism to optimally integrate features from both the frontal view and BEV perspectives.
- We develop a feature-matching module incorporating a Multi-Modal Cross-Attention Mechanism to enhance the utilization of radar point clouds. This module integrates a Noise-Resistant Matcher to provide Explicit Feature Matching Supervision. Thereby, RC-AutoCalib can effective filter out noise caused by height inaccuracies and enable robust learning of radar-image correspondences for calibration.
- Our approach demonstrates superior experimental results on the nuScenes dataset compared to existing LiDAR-camera calibration methods, establishing a new benchmark for future research.

2. Related Works

2.1. Offline Calibration

Offline calibration methods primarily depend on specific calibration targets and cannot address real-time errors. These methods are tailored for fixed environments and necessitate substantial manual effort to achieve precision, rendering them unsuitable for dynamic conditions and generally reserved for controlled settings. Early radar-camera calibration techniques focused on merging radar signals with camera data through homography projection that maps points from the radar's horizontal plane to the camera image plane. Due to inherent noise in radar sensors, these early methods often required specialized trihedral reflectors to establish accurate correspondences [10, 11, 29, 32]. However, the radar's limitation in accurately measuring the elevation of distant targets indicated that reflectors had to be positioned precisely on the radar's horizontal plane [29]. More recent radar calibration algorithms aim to minimize "reprojection error" to better synchronize object detection across both sensor fields of view, using techniques like estimating radar-to-camera transformations via reprojection error [12], or intersecting back-projected camera rays with 3D "arcs" that conform to radar measurements to determine necessary transformations [4]. Despite improvements, these methods still rely on specific targets and manual input efforts.

2.2. Online Calibration

Online methods primarily extract features from natural scenes for calibration, offering greater flexibility and adaptability to various scenarios. The rapid development of deep learning has demonstrated neural networks' powerful feature extraction capabilities. However, due to the aforementioned challenges associated with radar, online calibration methods for radar and cameras are less prevalent. In this paper, we focus on developing an end-to-end architecture for the online auto-calibration of radar and cameras, leveraging robust benchmarks established by LiDAR and camera calibration methods.

LiDAR and Camera. Li et al. [15] categorized targetless calibration methods into information theory-based, feature-based, ego-motion-based, and learning-based approaches. Pandey et al. [22] used mutual information between point cloud intensities and image grayscale values. Taylor and Nieto [30] utilized sensor ego-motion on moving vehicles to estimate extrinsic parameters. Levinson and Thrun [13] as well as Yuan et al. [41] optimized depth-discontinuous and depth-continuous edge features, respectively. Regnet [25] and CalibNet [8] employed deep learning to match features and regress calibration parameters. CalibRCNN [27] combined CNN with LSTM [24] and added pose constraints for accuracy. LCCNet [20] used cost volume for feature correlation. Despite achieving positive results, these methods do not explicitly learn the correspondence between point clouds and images. In contrast, in this paper, we introduce Explicit Feature Matching Supervision to guide the model in learning the correspondence relationship between point clouds and images more effectively.

Radar and Camera. Perši'c et al. [23] proposed an online calibration method based on detecting and tracking moving objects, focusing on rotational calibration. Schöller et al. [26] used deep learning to learn rotational calibration matrices but did not address translational calibration. Additionally, their methods utilize stationary traffic radars fixed on highway positions, differing from ours that employ vehicle-mounted 3D radars moving with the car. Wisec et al. [33] developed a targetless calibration method for 3D radar and cameras, using radar velocity information and motion-based camera pose measurements, solved with nonlinear optimization. Later, the same research team extended their work [33] to include radar ego-velocity estimates and unscaled camera pose measurements in [34] for a more complete spatiotemporal calibration. However, these methods overly rely on radar speed measurements, making them less robust to noise. Additionally, they do not leverage the power of deep learning and fail to explicitly establish the correspondence between radar and images.

3. Methods

The overall pipeline of the RC-AutoCalib method, depicted in Fig. 3, begins with RGB images and radar point clouds as inputs. These inputs pass through the Data Transform module, yielding the frontal view estimated depth map, frontal view miscalibrated radar depth map, pseudo-BEV image, and miscalibrated radar BEV. Subsequent processing occurs in the Feature Extraction module, where features are extracted. These features are then analyzed in the Feature Matching module to enhance understanding of the correlation between feature pairs. This module incorporates a Multi-Modal Cross-Attention Mechanism, Explicit Feature Matching Supervision, and a Noise-Resistant Matcher. Following this, the Selective Fusion Mechanism aggregates the features, and the system performs parameter regression to predict rotation and translation vectors necessary for auto-calibration. Detailed descriptions are provided below.

3.1. Data Transform module

To address issues of uncertainty caused by elevation ambiguity and the sparsity of radar data, we propose a Dual-Perspective feature representation. This approach projects two types of 3D data (i.e., the image plus its depth map and the radar point clouds) onto two different perspectives: bird's-eye view (BEV) and frontal view (FV). The BEV provides a domain where radar data is less impacted by uncertain height and offers more information about the geometry of the scene. Simultaneously, the FV retains rich

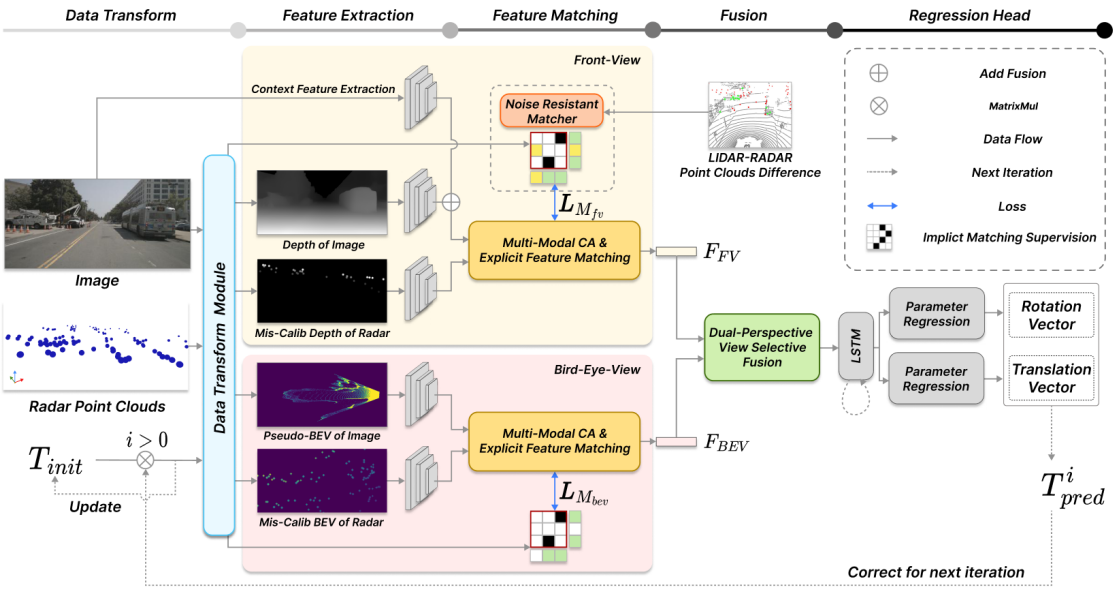


Figure 3. Our system flow for iterative online auto-calibration starts with the input image, point cloud, and initial calibration parameters T_{init}, which first pass through the Data Transform module (Sec. 3.1). Here, we obtain the estimated image depth map and miscalibrated radar depth map from the frontal view (FV) perspective, along with the pseudo-BEV image and miscalibrated BEV radar projection. These outputs are then processed in the Feature Extraction module (Sec. 3.2), where features from both FV and BEV perspectives undergo Feature Matching (Sec. 3.3) between the image and radar data. Subsequently, after Feature Matching and Fusion (Sec. 3.4), the Regression Head (Sec. 3.5) generates the rotation and translation vectors that form the transformation matrix, $\hat{T}^i_{pred}$, to refine calibration. Finally, $\hat{T}^i_{pred}$ is fed back to T_{init} to update the calibration parameters for the next i-th iteration.

semantic information, preserving important contextual details.

Radar data. Given a random initialized or roughly-estimated extrinsic transformation T_{init} [8, 20, 31, 42], consisting of a rotation matrix R_{init} and a translation vector t_{init}, we transform a 3D radar point $P_r = (X_r, Y_r, Z_r)$ from the radar coordinate to $P^c_r = (X^c_r, Y^c_r, Z^c_r)$ in the camera coordinate using Eq. (1). The projection formula in Eq. (2) is then used to generate mis-calibrated FV and BEV information maps. For the FV map, the recorded pixel value is computed as $I^{FV}_R(u_f, v_f) = Z^c_r$; as for the BEV map, the recorded value is determined as $I^{BEV}_R(u_b, v_b) = y^c_r$, with y^c_r being Y^c_r plus an offset of the camera height (i.e., the distance above the ground) to eliminate negative values. In Eq. (2), (u_f, v_f) and (u_b, v_b) are the coordinates of a radar point P_r projected onto FV and BEV planes using projection matrices K and K' correspondingly. Here, K is the original camera intrinsic matrix. The intrinsic parameters are manually pre-defined for K' based on the map resolution and map center.

$$P^c_r = T_{init} \begin{bmatrix} P_r \\ 1 \end{bmatrix} = \begin{bmatrix} R_{init} & t_{init} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_r \\ 1 \end{bmatrix}, \tag{1}$$

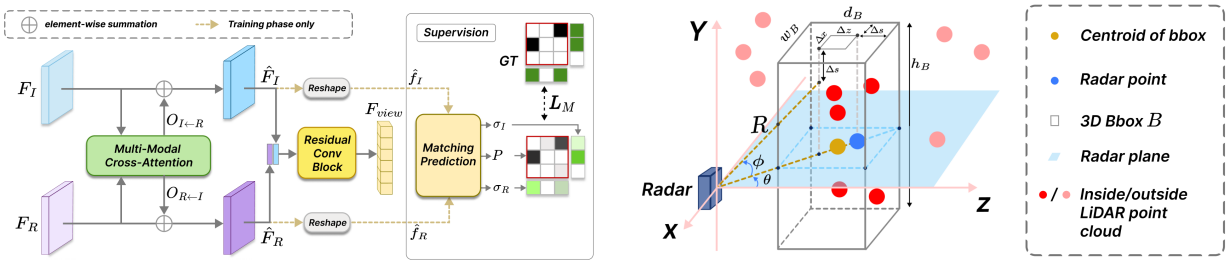
$$\begin{bmatrix} u_f \\ v_f \\ 1 \end{bmatrix} = K \begin{bmatrix} X^c_r/Z^c_r \\ Y^c_r/Z^c_r \\ 1 \end{bmatrix}, \; and \; \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix} = K' \begin{bmatrix} X^c_r \\ Z^c_r \\ 1 \end{bmatrix}. \tag{2}$$

Camera data. For the camera data, the FV information map $I^{FV}_I(u_d, v_d)$ is derived using the Metric Depth Estimation module, which predicts the depth image from the input image. This module employs two sequential methods: DepthAnything[40] for relative depth prediction and ZoeDepth[1] for refining it into metric depth, resulting in $I^{FV}_I(u_d, v_d)$. From the depth image, we convert each pixel into a pseudo point cloud $P_p = (X_p, Y_p, Z_p)$ based on Eq. (3), which are then transformed/projected using matrix K' to form the pseudo-BEV image I^{BEV}_I similar to the radar case I^{BEV}_R.

$$\begin{bmatrix} X_p & Y_p & Z_p \end{bmatrix}^\top = K^{-1} \cdot I^{FV}_I(u_d, v_d) \cdot \begin{bmatrix} u_d & v_d & 1 \end{bmatrix}^\top. \tag{3}$$

3.2. Feature Extraction

After transforming point clouds and images into two unified representations—FV depth maps (I^{FV}_R, $I^{FV}_I \in \mathbb{R}^{H \times W}$) and BEV maps ($I^{BEV}_I$, $I^{BEV}_R \in \mathbb{R}^{H' \times W'}$)—we use ResNet [5] and convolutional layers to extract features from these maps. Additionally, to enhance the FV semantic content, context features are extracted from the original image using ResNet18 and are integrated with the features from I^{FV}_I. The resulting feature sets, representing different perspectives for radar and camera, are denoted as F^{FV}_R, $F^{FV}_I \in \mathbb{R}^{H/8 \times W/8 \times C}$ and F^{BEV}_R, $F^{BEV}_I \in$



(a) Cross attention and Explicit Feature Matching Supervision

(b) 3D bbox B illustration

Figure 4. Illustration of the proposed Feature Matching module.

$\mathbb{R}^{H'/8\times W'/8\times C}$. We detail the network architecture in the supplementary.

3.3. Feature Matching

To estimate the 6-DoF (degrees of freedom) extrinsic transformation between radar and camera sensors, our model first focuses on identifying corresponding feature pairs from each sensor's perspective. As illustrated in Fig. 3, we conduct feature matching across two different perspectives. For each perspective, we implement similar matching modules that include a Multi-Modal Cross-Attention Mechanism and an Explicit Feature Matching Supervision. Additionally, in the FV case, we incorporate a Noise-Resistant Matcher.

Multi-Modal Cross-Attention Mechanism. Considering both perspectives, the projected point cloud data from radar is sparse with mostly zero values, while camera data is dense with rich depth information and geometric features. We propose a Multi-Modal Cross-Attention (MCA) Mechanism that enables the model to focus on non-zero feature regions and identify correlations between camera and radar features. As described in Equation Eq. (4), the inputs to the MCA are F_I and F_R, and the outputs are $O_{I\leftarrow R}$ and $O_{R\leftarrow I}$. These outputs are used to compute the updated features $\hat{F}_I$ and $\hat{F}_R$ as shown in Eq. (5).

$$\begin{aligned}(O_{I\leftarrow R}, O_{R\leftarrow I}) &= (\Theta(F_I, m_{I\leftarrow R}), \Theta(F_R, m_{R\leftarrow I})) \\ &= MCA(F_I, F_R)\end{aligned} \tag{4}$$

$$\hat{F}_I = F_I + O_{I\leftarrow R}, \quad \hat{F}_R = F_R + O_{R\leftarrow I}, \tag{5}$$

To obtain $O_{I\leftarrow R}$ and $O_{R\leftarrow I}$ in Eq. (4), we first compute the attended features $m_{I\leftarrow R}$ and $m_{R\leftarrow I}$ between the radar and image inside MCA. Inspired by [6, 17], we use Eq. (6) to compute attended features $m_{I\leftarrow R}$ and $m_{R\leftarrow I}$, which rely on the cross-attention score a_{IR} computed by Eq. (7).

$$m_{R\leftarrow I} = \text{Softmax}(a_{IR}^{\top})\, V_I, m_{I\leftarrow R} = \text{Softmax}(a_{IR})\, V_R, \tag{6}$$

$$a_{IR} = K_I^{\top} K_R, \tag{7}$$

where V_* and K_* are the value and key, respectively, extracted from the feature F_* through linear projections. Note that, we reshape F_* to $(m \times c)$ dimension before projection and $* \in \{I, R\}$. After obtaining these attention maps $m_{I\leftarrow R}$ and $m_{R\leftarrow I}$, we apply the cross-modal feature refinement function Θ to calculate the $O_{I\leftarrow R}$ and $O_{R\leftarrow I}$ by Eq. (4) given F_I and F_R. The details of Θ can be found in the supplementary material.

After obtaining $\hat{F}_I$ and $\hat{F}_R$, as shown in Fig. 4a, a Residual Conv Block is used to aggregate them into corresponding F_{view} for each perspective branch using Eq. (8), with $view \in \{BEV, FV\}$.

$$\begin{aligned}F_{view} = \Phi(&\text{conv}(\text{concat}(\hat{F}_I, \hat{F}_R)) \\ &+ \text{conv}(\text{conv}(\text{concat}(\hat{F}_I, \hat{F}_R)))),\end{aligned} \tag{8}$$

where the first term of Φ consists of the concatenated features of $\hat{F}_I$ and $\hat{F}_R$ after passing through one convolutional layer, while the second term involves passing through two convolutional layers. These terms are then added together to form our Residual Conv Block. Here, "conv" denotes a block that includes convolutional layers followed by the leakyReLU [38] activation function. The block Φ sequentially applies the leakyReLU activation function, flattens the features, and utilizes a multi-layer perceptron (MLP).

Explicit Feature Matching Supervision. Previously, feature matching was implicitly supervised solely by the final calibration loss to understand overall errors without specifically identifying matched pairs. We find this implicit supervision insufficient and propose directly supervising feature matching using true matching pairs generated from the correct calibration matrix. In other words, we add matching prediction and auxiliary loss during training to enhance the understanding of local feature matching pairs between $\hat{F}_I$ and $\hat{F}_R$.

Motivated by [17], an extra branch is designed to perform the task of Local Feature Matching during the training phase as shown in Fig. 4a. Assuming each reshaped feature $\hat{f}_* \in \mathbb{R}^{m\times c}$ from $\hat{F}_*$ includes m (i.e., $H/8 \times W/8$) keypoints, with each keypoint having a feature descriptor of dimension c. The assignment matrix $P \in [0,1]^{m\times m}$ is estimated based on Eq. (9).

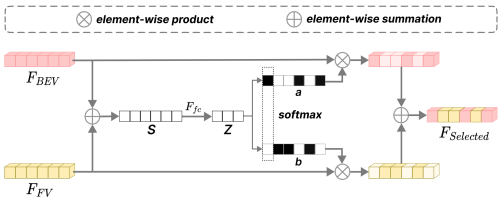


Figure 5. Details of the Selective Fusion Mechanism

$$P = \sigma_I^\top \, \sigma_R \, \text{Softmax}(S^\top)^\top \, \text{Softmax}(S), \tag{9}$$

where $S \in \mathbb{R}^{m \times m}$, calculated by equation Eq. (10), is the similarity score matrix between the features extracted from two sensors after the Multi-Modal Cross-Attention step. Meanwhile, $\sigma_* \in [0,1]^{1 \times m}$ is the matchable score of feature points, estimated by equation Eq. (11), where $* \in \{I, R\}$. A point with a high value of σ means it is more likely to have a corresponding point on another map.

$$S = \text{Linear}(\hat{f}_I)^\top \text{Linear}(\hat{f}_R), \tag{10}$$

$$\sigma_* = \text{Sigmoid}(\text{Linear}(\hat{f}_*)). \tag{11}$$

In the training phase, we use the estimated assignment matrix P and the matching loss defined in Eq. (16) to directly supervise the cross-sensor matching.

Noise-Resistant Matcher. For the FV case, when preparing the ground truth matches matrix $\mathcal{M}$ to supervise the assignment matrix P, we recognize that the radar training data contains many unreliable data points due to elevation ambiguity. Additionally, radar signals reflected by objects far from the radar plane can lead to unreliable data points. Therefore, we propose using LiDAR data to identify and remove these unreliable data points from the list of true feature matching pairs $\mathcal{M}$. First, we transform the LiDAR and radar point clouds into a unified camera coordinate system. For each radar 3D point $P_r^c = (X_r^c, Y_r^c, Z_r^c) \in \mathbb{R}^3$, a neighbor region is created using a 3D bounding box B. If the number of LiDAR point clouds $P_l^c = (X_l^c, Y_l^c, Z_l^c) \in \mathbb{R}^3$ within this box B exceeds a threshold τ, the radar point cloud P_r^c is considered reliable.

The 3D bounding box B is adaptively designed for each radar point. Denote ϕ, θ, and R respectively represent the elevation angle, azimuth angle, and point length relative to a 3D radar point. (Δx, Δz) are the noises caused by elevation ambiguity as defined in the method [28] and Δs is the error between the two sensors. As shown in Fig. 4b, the box height h_B is calculated by Eq. (12) given Δs, ϕ, R, and a predefined parameter δ, which represents the allowable height error from a 3D point to the radar plane.

$$h_B = 2(\Delta y + \Delta s) = 2(\delta + \Delta s), \; and \cos\phi = \sqrt{1 - (\delta/R)^2}. \tag{12}$$

The width w_B and depth d_B of B, are then calculated by Eqs. (13) and (14). The center of a 3D bounding box B of a radar point $P_r^c(X_r^c, Y_r^c, Z_r^c)$ is determined as: $(X_r^c - \Delta x/2,\ Y_r^c,\ Z_r^c - \Delta z/2)$.

$$w_B = \Delta x + 2\Delta s = R \sin\theta(1 - \cos\phi) + 2\Delta s, \tag{13}$$

$$d_B = \Delta z + 2\Delta s = R \cos\theta(1 - \cos\phi) + 2\Delta s. \tag{14}$$

Due to limited space, we provide additional information and details in the supplementary material.

3.4. Selective Fusion Mechanism

After the Feature Matching step, we can extract the features F_{BEV} and F_{FV} in the corresponding perspective branch. To enhance calibration accuracy by leveraging the distinctive contributions of each perspective, we propose a Dual-Perspective View Selective Fusion Mechanism that combines these features influenced by SKNet [14]. As illustrated in Fig. 5, we first calculate the compact feature z from the sum of the two feature vectors using Eq. (15). Subsequently, a channel-wise attention mechanism adaptively selects diverse elements in the corresponding input features, guided by the compact feature descriptor z. The obtained results are then added together to create the final feature F_{select}.

$$z = F_{fc}(F_{BEV} + F_{FV}), \tag{15}$$

where F_{fc} denotes the use of a fully connected layer, BatchNorm [7], and the leakyReLU activation function.

3.5. Regression Head

To estimate the rotation and translation parameters and form the updated transformation matrix T_{pred}, we leverage the sequence generative decoder from CalibDepth [43]. Specifically, this method employs LSTM [24], where the output is defined as a sequence of actions of length N in an autoregressive manner to address the inherent inaccuracies in the one-shot regression approach. After determining T_{pred} at the current iteration, we update $T_{init} = T_{init} \cdot T_{pred}^i$, as illustrated in Fig. 3.

3.6. Loss Function

Our model operates under two supervised tasks. The primary task focuses on auto-calibration, while an auxiliary task centers on local feature matching, which clarifies the correspondence between feature pairs. Consequently, we employ two corresponding loss functions: the matching loss and calibration loss.

Explicit Correspondences Matching. With ground truth match matrix $\mathcal{M} \in \{0,1\}^{N \times m \times m}$ and the predicted matching $P \in [0,1]^{N \times m \times m}$ and $\sigma_I, \sigma_R \in [0,1]^{N \times m}$, the matching loss function is designed to minimize the log-likelihood

of the predicted matches as in Eq. (16). Since our calibration includes N iterations, the matching loss term aggregates the loss from all iterations.

$$L_M = -\sum_{n}^{N}\left(\frac{\lambda}{\Sigma\mathcal{M}}L_{pos}^{n} + \frac{1-\lambda}{\Sigma\mathcal{N}_{\mathcal{I}} + \Sigma\mathcal{N}_{\mathcal{R}}}L_{neg}^{n}\right), \quad (16)$$

$$L_{pos} = \sum_{i,j}\log(P^{ij})\mathcal{M}^{i,j}, \quad (17)$$

$$L_{neg} = \sum_{i}\log(1-\sigma_I^i)\mathcal{N}_I^i + \sum_{j}\log(1-\sigma_R^j)\mathcal{N}_R^j, \quad (18)$$

where λ is the balancing coefficient between positive and negative instances. $\mathcal{N}_R$ and $\mathcal{N}_I$ are ground truth for "No Matchable" scores for pixels in the radar and camera maps respectively. The true matches matrix $\mathcal{M}$ between camera and radar maps is dynamically computed based on the true translation T_{gt} between the two sensors and the current, yet imperfect, updated translation T_{init}. $\mathcal{N}_R$ and $\mathcal{N}_I$ are derived from $\mathcal{M}$ by Eq. (19).

$$\mathcal{N}_I^i = 1 - \sum_{j}\mathcal{M}_{ij}, \quad \mathcal{N}_R^j = 1 - \sum_{i}\mathcal{M}_{ij}. \quad (19)$$

The final matching loss function is then computed as the sum of losses from two perspectives defined as $L_{matching} = L_{M_{bev}} + L_{M_{fv}}$.

Calibration Loss. To both ensure that the calibration results at each iteration step are asymptotic to the ground truth and avoid divergence between different iteration steps, we use the calibration loss L_{calib} proposed in [43]. By controlling the weight parameter β, the final total loss consists of the two losses mentioned above, defined as $L_{\text{total}} = L_{\text{calib}} + \beta L_{\text{matching}}$.

4. Experimental Results

4.1. Dataset and Evaluation Metrics

Dataset Preparation. We utilize a subset of images from the nuScenes dataset [2] for our training and testing processes. This subset comprises 12,610 samples for training, 1,628 samples for validation, and 1,623 samples for testing. The training and testing depth range spans from 0 to 200 meters, with input and output resolutions set at 400 × 192 pixels.

Evaluation Metrics. To facilitate comparison with previous work, we convert the output rotation vector to Euler angles. Then, we calculate the absolute error between the predicted values and the ground truth in all dimensions of angles and translation vectors. For all tables, the best and the second-best results are highlighted in bold and underlined, respectively.

Range	Methods	Rotation(°)				Translation(cm)			
		Mean	Roll	Pitch	Yaw	Mean	X	Y	Z
R1	LCCNet-1	1.603	**0.123**	3.130	1.556	16.531	22.992	17.648	**8.954**
	NetCalib2	1.205	0.387	2.289	**0.941**	12.297	**12.532**	12.076	12.284
	CalibDepth	0.807	0.390	0.345	1.686	12.608	12.860	12.250	12.715
	Coarse [26]	2.035	0.581	1.519	4.004	-	-	-	-
	Fine [26]	1.692	0.442	0.939	3.695	-	-	-	-
	Ours	**0.427**	0.130	**0.198**	0.953	**9.498**	12.563	**3.295**	12.635
R2	LCCNet-3	2.156	1.526	2.364	2.579	89.672	**71.660**	89.605	107.751
	LCCNet-5	1.898	0.919	2.314	2.461	88.302	74.216	85.239	105.450
	NetCalib2	2.778	1.465	4.688	2.180	71.037	76.001	57.204	79.906
	CalibDepth	1.686	1.149	0.808	3.102	55.380	77.146	12.918	76.078
	Coarse [26]	4.388	1.866	3.251	8.048	-	-	-	-
	Fine [26]	3.334	1.368	1.937	6.696	-	-	-	-
	Ours	**0.852**	**0.3597**	**0.4423**	**1.7544**	**47.537**	74.777	**5.415**	**62.420**

Table 1. Comparison with LiDAR-Camera-Based and Radar-Camera-Based Auto-Calibration Methods on the nuScenes dataset. The methods are compared with two mis-calibration ranges, R1 ($\pm 10, \pm 0.25m$) and R2 ($\pm 20, \pm 1.5m$).

FV	BEV	SF	MCA	EMS	NR	Rotation(°)				Translation(cm)			
						Mean	Roll	Pitch	Yaw	Mean	X	Y	Z
✓						0.657	0.235	0.301	1.436	12.602	12.858	12.247	12.700
	✓					0.689	0.295	0.381	1.392	12.605	12.870	12.285	12.660
✓	✓					0.575	0.209	0.284	1.232	12.315	12.863	11.416	12.667
✓	✓	✓				0.529	0.175	0.237	1.176	11.842	12.882	9.976	12.670
✓	✓	✓	✓			0.502	0.175	0.235	1.097	12.574	12.883	12.150	12.688
✓	✓	✓	✓	✓		0.463	0.140	0.206	1.042	9.627	12.563	3.682	12.636
✓	✓	✓	✓	✓	✓	**0.427**	**0.130**	**0.198**	**0.953**	**9.498**	**12.563**	**3.295**	**12.635**

Table 2. Module Impact Ablation (FV: Front View, BEV: Bird's Eye View, SF: Selective Fusion, MCA: Multi-Modal Cross-Attention Mechanism, EMS: Explicit Feature Matching Supervision, NR: Noise-Resistant Matcher)

4.2. Main results

We compared it against the method by Schöller et al. [26]. Although this method dates back to 2019 and may not incorporate recent advancements, we have also compared it with state-of-the-art LiDAR-camera-based auto-calibration methods, including LCCNet [20], CalibDepth [43], and NetCalib2 [36]. To ensure a fair comparison, we trained and tested all these methods using the same set of parameters and **radar-image dataset**. As shown in Sec. 4.2, our focus lies on testing two mis-calibration ranges: [10°, 0.25m] and [20°, 1.5m], representing small and large error ranges. For "LCCNet-number", the "number" corresponds to the number of iteration steps. The achieved results demonstrate that our method exhibits significantly superior average errors in both rotation and translation compared to other methods across both mis-calibration ranges.

4.3. Ablation Studies

Impact of each Module. Sec. 4.3 shows the impact of each module on experimental outcomes. Using both FV and BEV perspectives together reduced the mean absolute error in rotation by 12.5%. The Selective Fusion mechanism further reduced this error by 16.5% by allowing the model to choose suitable features from each perspective. The multi-

Fusion Method	Rotation(°)				Translation(cm)			
	Mean	Roll	Pitch	Yaw	Mean	X	Y	Z
Add Fusion	0.642	0.217	0.371	1.338	12.547	**12.832**	12.096	12.714
Concat Fusion	0.575	0.209	0.284	1.232	12.315	12.863	11.416	**12.667**
Selective Fusion	**0.529**	**0.175**	**0.237**	**1.176**	**11.842**	12.882	**9.976**	12.669

Table 3. Comparison of Dual-Perspective Fusion Methods with Basic Methods

β	Rotation(°)				Translation(cm)			
	Mean	Roll	Pitch	Yaw	Mean	X	Y	Z
0.1	0.470	**0.150**	0.215	1.044	**10.304**	12.587	**5.684**	12.640
0.3	0.471	0.160	0.224	1.029	10.377	12.555	5.940	**12.637**
0.5	**0.446**	0.164	**0.214**	**0.960**	12.112	**12.549**	11.151	**12.637**

Table 4. Ablation Study on Weight Impacts in Explicit Feature Matching Loss

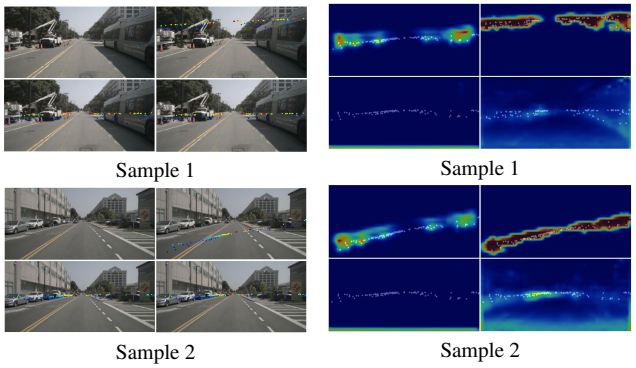
modal cross-attention mechanism also effectively reduced rotation error by 5%, demonstrating its capability to address the challenges of sparse radar depth maps. However, the mean absolute error in translation did not improve clearly based on the above modules due to the sparse and noisy radar point clouds. Next, introducing Explicit Feature Matching Supervision decreased the translation error by 23.4%, highlighting the benefit of explicit matching labels in learning correspondences. Finally, the incorporation of the Noise-Resistant Matcher further filtered out highly inaccurate noise points in the FV perspective and aided in translational calibration. As a result, the mean absolute error in rotation decreased by 7.8%

Dual-Perspective Fusion Methods. Additionally, we compare our Selective Fusion Mechanism with commonly used methods such as add fusion and concatenation fusion. In Sec. 4.3, the results demonstrate a significant improvement of the proposed method over the conventional fusion methods.

Weight of Explicit Feature Matching Loss. In Sec. 4.3, we test three β values: 0.1, 0.3, and 0.5, excluding the noise-resistant matcher module in our method. $\beta = 0.5$ achieved the lowest rotation error but the highest translation error. $\beta = 0.1$ provided the best translation accuracy with negligible rotation error. Thus, $\beta = 0.1$ was selected as the optimal balanced coefficient.

4.4. Qualitative Results

Our method provides accurate calibration results across various initial mis-calibration conditions and scenes. Fig. 6a visualizes these results, showing precise calibration even with significant initial errors and sparse radar points. To highlight the effectiveness of our explicit matching supervision, we compute the cross-attention maps $I_{I\leftarrow R}$, $I_{R\leftarrow I}$ between radar and image features, as visualized in Fig. 6b. The detailed formulas for computing these attention maps are provided in the supplementary material. We represent the attention maps using heatmaps and denote the projected



(a) Examples of calibration results by projecting radar points onto the frontal-view image. Top-left: input RGB image. Top-right: initial mis-calibrated radar point cloud projection. Bottom-left: network-predicted projection. Bottom-right: ground truth projection.

(b) Examples of cross-attention maps highlighting the important regions the model focuses on. Top: image-attentive radar importance regions. Bottom: radar-attentive image importance regions. Left/Right: Results with and without explicit matching supervision. The projected 3D points are used to highlight the critical regions. With explicit matching supervision, our model can better identify these critical regions.

Figure 6. Results visualization in the front view.

radar points with white color to indicate critical regions. Without explicit matching supervision, we observe that due to many zero-value relationships, the image-attentive radar exhibits weaker and less focused attention, while the radar-attentive image focuses on non-critical region, such as the ground. After incorporating explicit matching supervision, the image-attentive radar's attention becomes more concentrated on the non-zero regions, specifically the areas of radar point cloud projection. Meanwhile, the radar-attentive image effectively focuses on crucial regions, such as radar projection locations and vehicle contours, rather than being restricted to non-critical region.

5. Conclusion

In this work, we present RC-AutoCalib, an end-to-end network for 3D radar and camera calibration. By incorporating dual perspectives, we address the elevation ambiguity in 3D radar. Our Selective Fusion Mechanism integrates useful features from both FV and BEV perspectives. We also developed a Feature Matching module with a Multi-Modal Cross-Attention Mechanism to enhance radar point cloud utilization and a Noise-Resistant Matcher to filter out height-inaccurate noise. Our method achieves a calibration error of 0.427° in rotation and 9.498 cm in translation on the nuScenes dataset, demonstrating competitive performance with these SOTA auto-calibration methods using dense point clouds and establishing a benchmark for future research in 3D radar and camera calibration.

Acknowledgments This work was financially supported in part (project number: 112UA10019) by the Co-creation Platform of the Industry Academia Innovation School, NYCU, under the framework of the National Key Fields Industry-University Cooperation and Skilled Personnel Training Act, from the Ministry of Education (MOE) and industry partners in Taiwan. It also supported in part by the National Science and Technology Council, Taiwan, under Grant NSTC-112-2221-E-A49-089-MY3, Grant NSTC-110-2221-E-A49-066-MY3, Grant NSTC-111- 2634-F-A49-010, Grant NSTC-112-2425-H-A49-001, and in part by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and the Ministry of Education (MOE), Taiwan. We also would like to express our gratitude for the support from MediaTek Inc, Hon Hai Research Institute (HHRI), E.SUN Financial Holding Co Ltd, Advantech Co Ltd, Industrial Technology Research Institute (ITRI)

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 4

[2] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 7

[3] Joris Domhof, Julian FP Kooij, and Dariu M Gavrila. An extrinsic calibration tool for radar, camera and lidar. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8107–8113. IEEE, 2019. 1

[4] Ghina El Natour, Omar Ait Aider, Raphael Rouveure, François Berry, and Patrice Faure. Radar and vision sensors calibration for outdoor 3d reconstruction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2084–2089. IEEE, 2015. 1, 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[6] Markus Hiller, Krista A. Ehinger, and Tom Drummond. Perceiving longer sequences with bi-directional cross-attention transformers. *ArXiv*, abs/2402.12138, 2024. 5

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 6

[8] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1110–1117. IEEE, 2018. 1, 3, 4

[9] Xin Jing, Xiaqing Ding, Rong Xiong, Huanjun Deng, and Yue Wang. Dxq-net: differentiable lidar-camera extrinsic calibration using quality-aware flow. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6235–6241. IEEE, 2022. 1

[10] Du Yong Kim and Moongu Jeon. Data fusion of radar and image measurements for multi-object tracking via kalman filtering. *Information Sciences*, 278:641–652, 2014. 1, 3

[11] Jihun Kim, Dong Seog Han, and Benaoumeur Senouci. Radar and vision sensor fusion for object detection in autonomous vehicle surroundings. In *2018 tenth international conference on ubiquitous and future networks (ICUFN)*, pages 76–78. IEEE, 2018. 3

[12] Taehwan Kim, Sungho Kim, Eunryung Lee, and Miryong Park. Comparative analysis of radar-ir sensor fusion methods for object detection. In *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pages 1576–1580. IEEE, 2017. 1, 3

[13] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: science and systems*. Citeseer, 2013. 3

[14] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 510–519, 2019. 6

[15] Xingchen Li, Yuxuan Xiao, Beibei Wang, Haojie Ren, Yanyong Zhang, and Jianmin Ji. Automatic targetless lidar–camera calibration: a survey. *Artificial Intelligence Review*, 56(9):9949–9987, 2023. 3

[16] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection, 2022. 1

[17] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 5

[18] Zhijian Liu, Haotian Tang, Sibo Zhu, and Song Han. Semalign: Annotation-free camera-lidar calibration with semantic alignment loss. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8845–8851, 2021. 1, 2

[19] Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12507–12516, 2021. 1, 2

[20] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2894–2901, 2021. 1, 2, 3, 4, 7

[21] Xudong Lv, Shuo Wang, and Dong Ye. Cfnet: Lidar-camera registration using calibration flow network. *Sensors*, 21(23): 8112, 2021. 1

[22] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2053–2059, 2012. 3

[23] Juraj Peršić, Luka Petrović, Ivan Marković, and Ivan Petrović. Online multi-sensor calibration based on moving object tracking. *Advanced Robotics*, 35(3-4):130–140, 2021. 3

[24] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014. 3, 6

[25] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE, 2017. 1, 3

[26] Christoph Schöller, Maximilian Schnettler, Annkathrin Krämmer, Gereon Hinz, Maida Bakovic, Müge Güzet, and Alois Knoll. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3934–3941. IEEE, 2019. 1, 3, 7

[27] Jieying Shi, Ziheng Zhu, Jianhua Zhang, Ruyu Liu, Zhenhua Wang, Shengyong Chen, and Honghai Liu. Calibrcnn: Calibrating camera and lidar by recurrent convolutional neural network and geometric constraints. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10197–10202. IEEE, 2020. 1, 2, 3

[28] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9285, 2023. 6

[29] Shigeki Sugimoto, Hayato Tateda, Hidekazu Takahashi, and Masatoshi Okutomi. Obstacle detection using millimeter-wave radar and its visualization on image sequence. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 342–345. IEEE, 2004. 1, 3

[30] Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor arrays. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4843–4850. IEEE, 2015. 3

[31] Guangming Wang, Jiahao Qiu, Yanfeng Guo, and Hesheng Wang. Fusionnet: Coarse-to-fine extrinsic calibration network of lidar and camera with hierarchical point-pixel fusion. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8964–8970. IEEE, 2022. 1, 2, 4

[32] Tao Wang, Nanning Zheng, Jingmin Xin, and Zheng Ma. Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. *Sensors*, 11(9):8992–9008, 2011. 1, 3

[33] Emmett Wise, Juraj Peršić, Christopher Grebe, Ivan Petrović, and Jonathan Kelly. A continuous-time approach for 3d radar-to-camera extrinsic calibration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13164–13170. IEEE, 2021. 3

[34] Emmett Wise, Qilong Cheng, and Jonathan Kelly. Spatiotemporal calibration of 3-d millimetre-wavelength radar-camera pairs. *IEEE Transactions on Robotics*, 2023. 3

[35] Shan Wu, Amnir Hadachi, Damien Vivet, and Yadu Prabhakar. Netcalib: A novel approach for lidar-camera auto-calibration based on deep learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6648–6655. IEEE, 2021. 1, 2

[36] Shan Wu, Amnir Hadachi, Damien Vivet, and Yadu Prabhakar. This is the way: Sensors auto-calibration approach based on deep learning for self-driving cars. *IEEE Sensors Journal*, 21(24):27779–27788, 2021. 1, 7

[37] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5418–5427, 2022. 1

[38] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 5

[39] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 496–512. Springer, 2020. 1

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 4

[41] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters*, 6(4):7517–7524, 2021. 1, 2, 3

[42] Ganning Zhao, Jiesi Hu, Suya You, and C-C Jay Kuo. Calibdnn: multimodal sensor calibration for perception using deep neural networks. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXX*, pages 324–335. SPIE, 2021. 2, 4

[43] Jiangtong Zhu, Jianru Xue, and Pu Zhang. Calibdepth: Unifying depth map representation for iterative lidar-camera online calibration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733. IEEE, 2023. 1, 2, 6, 7

[44] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16280–16290, 2021. 1