CalibBEV: LiDAR-Camera Calibration via BEV Alignment

Filippo D'Addeo [*] [†]
filippo.daddeo2@unibo.it

Lorenzo Cipelli [*] [‡]
lorenzo.cipelli@unipr.it

Adriano Cardace [§]
cardace@stanford.edu

Emanuele Ghelfi [¶]
eghelfi@ambarella.com

Andrea Zinelli [¶]
azinelli@ambarella.com

Massimo Bertozzi [‡]
massimo.bertozzi@unipr.it

Abstract

*We present **CalibBEV**, a novel Bird's Eye View (BEV) alignment approach for LiDAR-camera calibration. Our method unifies LiDAR and camera data into a shared 3D spatial representation, enabling accurate and robust cross-modal calibration. CalibBEV extracts sensor-wise BEV features from each modality using domain-specific architectures and estimates the calibration matrix through a two-step alignment process. First, we perform an implicit alignment by regressing a coarse calibration matrix directly from the BEV features. To ease this alignment, we enforce semantic consistency between BEV representations across modalities using a contrastive loss inspired by CLIP, guiding both networks toward a unified feature space. In the second step, we leverage our BEV formulation to explicitly align the features of one modality with the other, refining the initial coarse estimate into a final, more accurate calibration matrix. CalibBEV significantly outperforms prior point-to-pixel matching methods, achieving state-of-the-art calibration accuracy. On the KITTI and nuScenes benchmarks, our method reduces the Relative Rotation Error (RRE) by 51% and 68%, and the Relative Translation Error (RTE) by 80% and 91%, respectively, compared to previous methods.*

1. Introduction

Image-to-Point Cloud registration is crucial for applications in autonomous driving, 3D computer vision, augmented/virtual reality, and various robotics domains. The goal is to estimate the rigid transformation, including translation and rotation, that accurately aligns the projection of a LiDAR point cloud with a reference image.

Early works [18, 31] focus on extracting 2D and 3D features from RGB images and point clouds to establish 2D-3D correspondences. However, these methods often struggle with matching image and point cloud features, as different modalities and architectures do not inherently produce a shared feature space for seamless feature matching. To address this challenge, several attempts have been made: [42] introduces a new voxel-based branch to learn more similar representations, [2] uses virtual point cloud to match corresponding representations of points, and [20] introduces correspondence queries to extract keypoints.

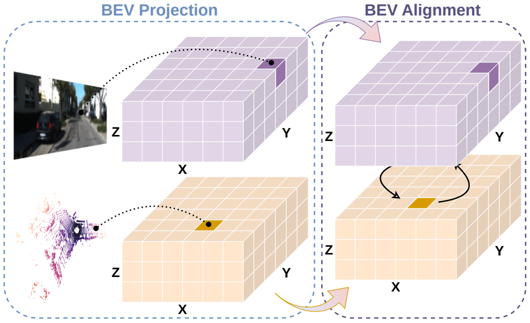

Figure 1. **BEV alignment.** Domain-specific features are computed independently and projected into 3D BEV representations. Then, our BEV alignment module estimates the calibration matrix between the two sensors.

In this work, we address the problem of Image-to-Point Cloud registration from a novel perspective, formulating it as a *Bird's Eye View (BEV) alignment* task inspired by object detection literature [7, 15, 21, 27, 38, 39]. As illustrated in Fig. 1, we aim to generate two BEV feature representations of the same scene, one from a sparse LiDAR point cloud and the other from an RGB image, while leveraging modality-specific architectures without imposing any architectural constraints. Although the two BEV representations are derived from different modalities, they capture semantically similar scene structures, and we can frame Image-to-Point Cloud registration as the problem of aligning these BEV features. The BEV alignment task is achieved with a two-step algorithm, referred to as implicit and explicit

[*] Equal contribution.
[†] University of Bologna, Department of Industrial Engineering, Italy.
[‡] University of Parma, Department of Engineering and Architecture, Italy.
[§] Stanford University, USA.
[¶] VisLab srl, an Ambarella Inc. company, Italy.

alignment. First, in the implicit alignment, we consider the height dimension of the two BEVs as the channels of two top views, and feed them to a CNN-based decoder which is trained together with a rotation and a translation heads to predict a coarse rigid transformation. Noticeably, we do not explicitly align the two BEVs. Instead, we let the decoder implicitly learn the registration parameters from the misaligned BEV feature maps by supervising the decoder and the prediction heads with the ground-truth rigid transformations. We argue that this is feasible due to the large receptive field of CNNs, which enables a global understanding of the BEV feature maps and allows the network to reason about the geometry of the two views for accurate calibration.

Although this strategy seems to be already effective as proven by our experiments reported in Sec. 4.4, a key improvement can be done by reasoning on the modality gap: the two BEV representations are generated by different networks and thus reside in distinct feature spaces. Simply feeding them into the same decoder results in suboptimal performance, as the decoder must simultaneously handle both alignment and feature discrepancies. To address this, inspired by recent multi-modal alignment techniques [30], we adopt a CLIP loss that guides point features to be similar to their corresponding re-projected pixel features and vice-versa, leveraging known point-to-pixel correspondences during training. This encourages modality-specific networks to learn a unified feature space representation. As a result, the BEV feature maps become more compatible, simplifying the Implicit Alignment task for the decoder and improving overall registration performance.

In the second step, we leverage our BEV formulation to perform an explicit alignment. Specifically, we use the coarse transformation matrix from the previous step to warp the LiDAR BEV features, reducing the misalignment with respect to the RGB BEV features. The aligned feature maps are then fed into a final decoder, which refines the calibration estimate to produce a final registration matrix.

Finally, unlike previous methods [2, 20, 23, 42], we take one final step further by easily extending our model to leverage a multi-camera setup, in order to have a BEV representation with richer and denser RGB information.

To summarize, our contributions are:

- We propose `CalibBEV`, a framework that for the first time frames the problem of Image-to-Point Cloud registration as a BEV alignment problem.
- Leveraging our unique BEV formulation, we propose a two-step alignment algorithm, where the first step performs an implicit alignment to estimate a coarse calibration, which is exploited in the explicit alignment step to obtain the final and refined calibration.
- We demonstrate that `CalibBEV` is easily extendable to support multi-camera setups, leading to better registration performances.
- `CalibBEV` achieves state-of-the-art-performance in both KITTI and nuScenes, surpassing previous methods by a significant margin while being more robust.

2. Related Work

2.1. Image-to-Point Cloud Registration

The LiDAR-Camera extrinsic calibration problem has been extensively studied, with early solutions relying on classical target-based and target-less methods. Target-based approaches [5, 33, 35] use predefined objects and optimization techniques to align sensor perceptions. In contrast, target-less methods [17, 26] exploit environmental features or statistical models, resulting in more versatile approaches. Moreover, differently from the target-based, the target-less approaches can also be employed in online applications. Our approach falls into the target-less category. Latest state-of-the-art correspondence-based, target-less methods [10, 14, 40] achieve remarkable registration performance. However, they typically involve heavy pre-processing, numerous frames accumulation, and accurate parameter fine tuning for each scene, resulting unsuitable for driving scenario, where scenes might drastically change in few seconds and fast inference is required. In contrast, `CalibBEV` doesn't require any particular pre-processing or accumulation technique.

2.2. Projection-based Methods

Projection-based methods estimate sensor calibration by projecting the LiDAR point cloud onto the image plane, generating a sparse depth map, and training a neural network to regress calibration parameters. RegNet [32] employs two separate backbones to extract features from the RGB image and sparse depth map, which are then fused for calibration estimation. Similarly, CalibNet [11] enforces geometric consistency by ensuring that the projected sparse depth map, transformed with predicted extrinsics, aligns with the ground-truth sparse depth. Building on these foundations, later methods introduced more advanced architectures: LCCNet [24] incorporates a feature matching layer to correlate RGB and depth features, while CalibFormer [37] leverages attention mechanisms for feature fusion. Despite their effectiveness, projection-based methods have a key limitation: when there are no correspondences between the sparse depth projection and RGB, registration becomes impossible. In contrast, our approach directly processes RGB images and point clouds, accommodating uncalibrated sensors with rotations up to ±360°.

2.3. Point-based Methods

Unlike projection-based methods, point-based algorithms operate directly in the 3D domain, processing point clouds instead of 2D sparse depth maps. DeepI2P [18] is one of

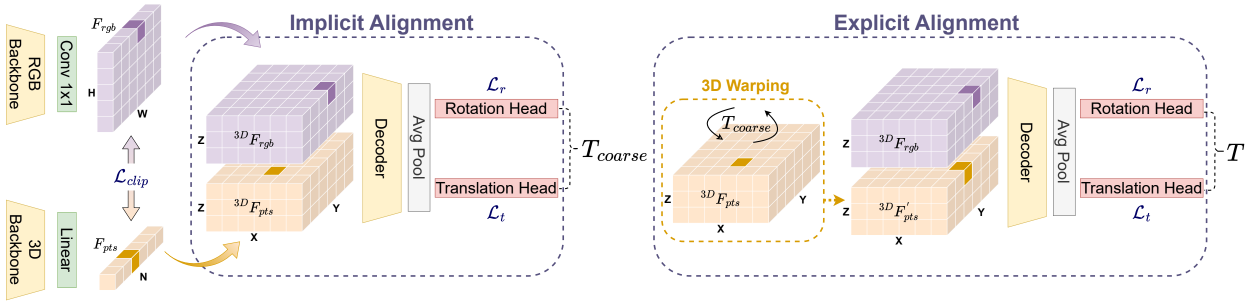

Figure 2. **Overview of our framework**. We process RGB images and point clouds independently, and adopt a CLIP loss between point and pixels features to encourage the networks to learn similar representations for corresponding pixel-point pairs. The extracted features are then lifted into two BEV representations, from which a decoder and prediction heads directly regress an initial coarse calibration matrix T_{coarse} (**Implicit Alignment**). In the **Explicit Alignment** step, all previous network components are frozen, and the coarse calibration matrix is applied to warp the LiDAR BEV features toward the RGB BEV reference using a feature warping operator. Both the decoder and prediction heads are initialized from the previous step, and fine-tuned on the aligned features to produce the final calibration matrix T.

the first methods to leverage 3D architectures [19, 29] to directly extract features from the point cloud. In its first stage, DeepI2P classifies whether each point lies within the camera frustum, or not. The second stage addresses the inverse camera projection problem by coarsely classifying points into corresponding image patches. Finally, a RANSAC-based PnP algorithm is applied to the grid classification output. Similarly, CorrI2P [31] proposes an early classification stage to detect the overlapping region between the RGB and the point cloud while introducing cross-attention into the framework. In the second stage, the camera pose can be obtained by applying EPnP [16] within RANSAC on the dense correspondences between 2D and 3D features. These methods often struggle with matching image and point cloud features, as different modalities and architectures (CNNs for images and MLPs for point clouds) do not inherently produce a shared feature space for seamless feature matching. To address this challenge, VP2P-Match [42] introduces voxel-based representations into the framework exploiting an additional backbone. The voxel branch captures spatially local patterns, akin to the image branch, ensuring that pixels and voxels share similar feature structures. Finally, a differentiable probabilistic PnP solver [4] estimates the camera pose, making the network fully end-to-end differentiable. Later, CurrI2P [23] improves both VP2P-Match and CorrI2P through the curriculum learning technique. However, these methods tend to be slow due to iterative nature of the outlier filtering methods such as RANSAC and the use of differentiable solvers, while also heavily relying on the classification of which points fall into the camera frustum. For this reason, GraphI2P [2] focuses on the feature consistency by estimating the camera pose given the LiDAR point cloud and a virtual cloud, obtained via depth estimation [1], and by leveraging a point sampling on an unified spherical representation to build consistent features patterns, facilitating both the feature fusion and the graph-based correspondence selection phase. Finally, ICLM [20] introduces a set of learnable correspondence queries specialized as 2D and 3D keypoints detectors for estimating the final pose. Similarly to previous approaches, we operate in a setting where features from the 2D and 3D domains are extracted from two independent networks and we adopt the same benchmark of point-based methods, consisting of a random vertical axis rotation in the range ±360° and a random forward-backward and left-right axis translation in the range ±10m.

3. Method

3.1. Background

As illustrated in Fig. 2 (left), our network takes as input both an image $I \in \mathbb{R}^{H \times W \times 3}$ and a point cloud $P \in \mathbb{R}^{N \times 3}$ collected by an RGB camera and a LiDAR, respectively, where H and W are the height and the width of the images and N is the number of points in the cloud. Following previous works [7, 20, 42], we adopt two state-of-the-art 2D and 3D branches to independently encode the input modalities, opting for this approach over more complex multi-modal fusion architectures to achieve a lighter and faster network. Building on [7], we first encode images with a ResNet-50 [8] and then we bilinearly upsample the features C4 of dimensions $H/16 \times W/16 \times C_4$, to $H/8 \times W/8 \times C_4$. These upsampled features are concatenated with the ResNet features C3 and finally encoded with a two-layer convolutional neural network (CNN), with instance normalization and ReLU activations, building the final 2D feature F_{rgb} of size $H/8 \times W/8 \times C_{rgb}$. In parallel, as in [42], we encode the point cloud using a Point Transformer [41], which consists of a downsampling and an upsampling module. Specifically, the downsampling applies a sequence of set abstraction layers [28] with a self-attention mechanism in order to obtain a global feature with dimensions $1 \times C'_{pts}$, while the upsampling employs a set of feature propagation modules [28] computing the 3D point-wise features $F_{pts} \in \mathbb{R}^{N \times C_{pts}}$. We refer to the supplementary material for a detailed graphical representation of both the backbones. Subsequently, to ensure the same features dimensionality C for both the RGB and point features, we encode the F_{rgb} and F_{pts} with a 1×1 convolutional layer and a single linear layer, respectively. Then, we lift and project the 2D RGB and the 3D point cloud features into two 3D Bird's Eye View (BEV) spaces, and an Implicit Alignment module is introduced to estimate a coarse calibration matrix T_{coarse} (see Sec. 3.2) directly from the mis-registered BEVs features. Finally, the coarse transformation is employed to explicitly align the BEV features of the 3D branch with the one from the RGB, and predict the final registration matrix (see Sec. 3.3).

3.2. Implicit Alignment

The majority of the existing approaches handle the Image-to-Point Cloud registration as a classification task followed by a correspondence-matching phase, which enables estimating the mis-registration pose between the camera and the 3D sensor. In order to have a flexible architecture, we choose to formulate the Image-to-Point Cloud registration as a BEV alignment task. Indeed, we aim to obtain two BEV representations of the same scene: one from the RGB image and the other from the sparse LiDAR point cloud. For this reason, inspired by object detection literature [7], we first define a volume of 3D coordinates $G \in \mathbb{R}^{X \times Y \times Z \times 3}$, and a zero-initialized 3D space ${}^{3D}F_{rgb} \in \mathbb{R}^{X \times Y \times Z \times C}$, representing the 3D coordinates and 3D feature space obtained from images surrounding the vehicle. We employ a East-Down-North (EDN) coordinate system, centering the volume with the camera frame origin. Then, in order to sample and lift the 2D features from F_{rgb} to ${}^{3D}F_{rgb}$, the 3D coordinates G are projected from the camera EDN frame $\{c\}$ onto the image plane with the intrinsics matrix K through Eq. (1) to get the corresponding 2D coordinates:

$$\begin{bmatrix} u & v & d \end{bmatrix}^T = K \cdot g, \;\; \forall g = \begin{bmatrix} {}^cx & {}^cy & {}^cz \end{bmatrix}^T \in G, \quad (1)$$

and we sample features given the 2D coordinates to fill the corresponding cell in ${}^{3D}F_{rgb}$.

Similarly, we define a zero-initialized 3D space ${}^{3D}F_{pts} \in \mathbb{R}^{X \times Y \times Z \times C}$. The mis-aligned point cloud is then projected onto this three-dimensional space and the features F_{pts} are scattered into ${}^{3D}F_{pts}$ accordingly to the corresponding projection cells. Afterward, we flatten the Y and C axes both in ${}^{3D}F_{rgb}$ and ${}^{3D}F_{pts}$, obtaining BEV feature maps with dimensions $X \times Z \times (Y \cdot C)$. We then concatenate the two BEVs along the channel dimension, and feed them to a single 3×3 convolutional layer $\varphi(\cdot)$ followed by instance normalization and a non-linearity for fusing the two modalities:

$${}^{3D}F_{fused} = \varphi \left({}^{3D}F_{rgb} \oplus {}^{3D} F_{pts} \right), \quad (2)$$

with $\oplus$ being the concatenation. Finally, the fused features are processed by a convolutional decoder which, thanks to its large receptive field, can reason globally about the geometry and misalignment between the two BEVs, producing high-level features that benefit the rotation and translation heads. It is important to note that although the two BEVs are not explicitly aligned, we empirically found that the decoder is sufficiently powerful to extract the necessary features to solve the calibration task, as demonstrated by our experiments. To this end, we adopt a ResNet-18 [8] architecture as the decoder, where the average-pooled C5 features serve as input to the translation and rotation prediction heads. Both heads are implemented as two-layer MLPs with ReLU non-linearity. We refer to the supplementary material for further implementation details. For the translation component, we directly predict the translation vector $t = \begin{bmatrix} x & y & z \end{bmatrix}^T$. Conversely, for the rotation component, we estimate the sine and cosine values of the rotation angles along each axis. From these predictions, we construct the corresponding rotation matrices $R_z(\psi)$, $R_y(\phi)$, and $R_x(\theta)$, which are then combined to decode the coarse calibration matrix as:

$$T_{coarse} = \begin{bmatrix} R_z(\psi) \cdot R_y(\phi) \cdot R_x(\theta) & t \\ 0 & 1 \end{bmatrix}. \quad (3)$$

As discussed in Sec. 1 and demonstrated by our experiments (see Sec. 4), the decoder effectively aligns the concatenated feature maps ${}^{3D}F_{fused}$, yielding accurate calibration matrix estimates. Nonetheless, we propose to further boost performance by adopting a CLIP-based [30] loss function that encourages ${}^{3D}F_{rgb}$ and ${}^{3D}F_{pts}$ to reside in a unified feature space. In particular, we aim at guiding the two backbones to learn similar features for corresponding pixel-point pairs. This alignment facilitates the decoder's task, allowing it to focus solely on spatial alignment rather than compensating for modality-specific feature discrepancies. During training, feature maps from the 2D and 3D branches are first normalized row-wise obtaining the normalized features $\tilde{F}$ as follows:

$$\tilde{F} = D^{-1} \cdot F, \;\; F \in \{F_{rgb}, F_{pts}\}, \quad (4)$$

with D being a square diagonal matrix with $D_{ii} = \|f_i\|_2$ and i indicating the i-th row. Then, the similarity matrix $S \in \mathbb{R}^{(H/8 \times W/8) \times N}$ between $\tilde{F}_{rgb}$ and $\tilde{F}_{pts}$ is computed as follows:

$$S = e^{\gamma} \left(\tilde{F}_{rgb} \cdot \tilde{F}^T_{pts} \right), \quad (5)$$

where γ is a learnable temperature parameter.

In order to build the target one-hot 2D-3D correspondences matrix $\hat{S} \in \mathbb{R}^{(H/8 \times W/8) \times N}$, we rely on the known ground truth registration matrix $\begin{bmatrix}\hat{R} & \hat{t}\end{bmatrix}$ at training time to correctly transform the point cloud into camera coordinates and project it into the image plane to find the corresponding pixel-point pairs:

$$\begin{bmatrix}u & v & d\end{bmatrix}^T = K\left(\hat{R}\,p + \hat{t}\right) \quad \forall p \in P. \tag{6}$$

Following CLIP [30], we compute the average symmetric cross-entropy loss in both directions:

$$\mathcal{L}_{clip} = \frac{\mathcal{L}_{\texttt{CE}(\sigma(S),\hat{S})} + \mathcal{L}_{\texttt{CE}(\sigma(S^T),\hat{S}^T)}}{2}, \tag{7}$$

with σ being the Softmax operator.

At the same time, we supervise the rotation and translation heads. To this end, we employ an L1 loss $\mathcal{L}_r$ for supervising the sine and the cosine rotation values along each axis, while we minimize the L2 norm difference $\mathcal{L}_t$ between the predicted and the ground truth translation vectors to compute the translation error. The overall optimization function is:

$$\mathcal{L} = \alpha\,\mathcal{L}_{clip} + \mathcal{L}_r + \mathcal{L}_t, \tag{8}$$

where α is the CLIP loss weighting factor.

3.3. Explicit Alignment

Peculiar to our BEV formulation, we can exploit the coarse transformation matrix from the Implicit Alignment step to explicitly align ${}^{3D}F_{pts}$ with ${}^{3D}F_{rgb}$, as depicted in Fig. 2 (right), and provide partially aligned feature maps to the decoder and the prediction heads, leading to a better estimate of the calibration matrix. Specifically, in the second step, we freeze the Implicit Alignment module and we employ T_{coarse}, to pre-align the 3D points features, ${}^{3D}F_{pts}$, with the RGB features ${}^{3D}F_{rgb}$ obtaining ${}^{3D}F^{'}_{pts}$. We do this by applying a differentiable sampling mechanism [12] in which each feature vector at location g in ${}^{3D}F^{'}_{pts}$ is sampled from a position $\bar{g} = T_{coarse}(g)$ in ${}^{3D}F_{pts}$ in the following way:

$${}^{3D}F^{'}_{pts}(g) = \sum_{g_i \in \mathcal{N}(\bar{g})} w_{g_i}\,{}^{3D}F_{pts}(g_i), \tag{9}$$

where w_{g_i} and $\mathcal{N}$ are the bi-linear kernel weights obtained from T_{coarse} and the set of neighboring pixels, respectively. Hence, Eq. (9) can be considered as a backward warping, where the new features in ${}^{3D}F^{'}_{pts}$ are obtained by warping ${}^{3D}F_{pts}$ according to T_{coarse}. Finally, ${}^{3D}F_{rgb}$ and ${}^{3D}F^{'}_{pts}$ are fused following Eq. (2), and a new decoder with the corresponding prediction heads is employed to predict the fine calibration matrix T_{fine}.

The final transformation matrix T is obtained by combining the coarse and fine calibration matrix:

$$T = T_{fine} \cdot T_{coarse}. \tag{10}$$

Notably, both the decoder and prediction heads of the Explicit Alignment module adopt the same architecture as in the previous stage. We initialize their weights from the previous step for faster convergence, employing the same loss functions to supervise the final transformation matrix T as before.

4. Experiments

4.1. Datasets and Metrics

We use the public KITTI [6] and nuScenes [3] datasets for training and evaluating our network. Regarding the KITTI dataset, we use the odometry benchmark which consists of 22 stereo sequences containing the images collected by two RGB stereo cameras facing forward the vehicle and the point cloud acquired by a LiDAR mounted on top of the vehicle. Following previous works, we use the sequences from 00 to 08 for training and the sequences 09 and 10 for testing. In addition, we employ the images acquired by both the RGB stereo cameras both for training and testing. The nuScenes dataset consists of 1000 different scenes containing the images collected by 6 different cameras returning a 360° Field Of View (FOV) of the scene around the vehicle and the point clouds acquired by 5 radars and a LiDAR. Following previous works, we use the official 700 sequences in the train split and the 150 scenes in the evaluation split for training, while we use the remaining 150 scenes of the test split for testing. We use only the images acquired by the front camera and the point cloud acquired by the LiDAR for both training and testing.

As evaluation metrics, we follow previous works [2, 20, 42], and we evaluate the registration performance with the average Relative Translation Error (RTE) and the average Relative Rotation Error (RRE) [25]. Moreover, for a complete comparison with other works, we also report the registration accuracy (Acc.) [42] which corresponds to the proportions of registrations achieving RTE $<$ 2m and RRE $<$ $5°$, even if it does not properly asses the fine-grained quality of predicted registrations. Finally, we report the RTE and RRE standard deviations, computed by aggregating all the RTE and RRE values for each test sample.

4.2. Implementation Details

As done in [2, 20, 23], for the KITTI dataset, we set the input RGB images dimensions to be 160×512. These are obtained by removing the top-50 rows, applying a 0.5 downsample factor, and randomly cropping the resulting images to match the desired dimensions. Afterward, we downsample the point cloud size to $N = 40960$. Similarly, for nuScenes dataset, we set images to be of size 160×320. These are obtained by removing the top-100 rows, applying

Method	KITTI			nuScenes		
	RTE(m)↓	RRE(°)↓	Acc.↑	RTE(m)↓	RRE(°)↓	Acc.↑
Grid Cls. + PnP [18]	3.64 ± 3.46	19.19 ± 28.96	11.22	3.02 ± 2.40	12.66 ± 21.01	2.45
DeepI2P (3D) [18]	4.06 ± 3.54	24.73 ± 31.69	3.77	2.88 ± 2.12	20.65 ± 12.24	2.26
DeepI2P(2D) [18]	3.59 ± 3.21	11.66 ± 18.16	25.95	2.78 ± 1.99	4.80 ± 6.21	38.10
CorrI2P [31]	3.78 ± 65.16	5.89 ± 20.34	72.42	3.04 ± 60.76	3.73 ± 9.03	49.00
CurrI2P (CorrI2P) [23]	1.55 ± 7.99	3.61 ± 10.88	-	2.16 ± 4.20	2.98 ± 5.35	-
VP2P-Match [42]	0.75 ± 1.13	3.29 ± 7.99	83.04	0.89 ± 1.44	2.15 ± 7.03	88.33
CurrI2P (VP2P-Match) [23]	0.53 ± 1.00	2.11 ± 9.48	-	1.04 ± 1.64	2.67 ± 8.61	-
RelaI2P [9]	0.72 ± 1.45	2.92 ± 6.67	85.60	-	-	-
ICLM [20]	$\underline{0.20} \pm 0.21$	$\underline{1.24} \pm 2.34$	97.49	0.63 ± 0.44	2.13 ± 3.75	90.94
GraphI2P [2]	0.32 ± 0.81	1.65 ± 1.32	$\underline{99.61}$	$\underline{0.49} \pm 1.22$	$\underline{1.73} \pm 1.63$	$\underline{99.48}$
CalibBEV (ours)	**0.04 ± 0.10**	**0.61 ± 0.52**	**99.96**	**0.04 ± 0.08**	**0.54 ± 0.45**	**99.98**

Table 1. Registration results on the KITTI odometry and nuScenes datasets. Lower is better for both RTE and RRE, while higher is better for accuracy. Values are from original papers and '-' due to missing results and training code.

a 0.2 downsample factor, and randomly cropping the resulting images to the desired dimensions. In regard to the 3D branch, we set the point cloud dimension $N = 40960$ by accumulating from the past frames, since a single LiDAR sweep in nuScenes contains roughly 36k points. However, this is not a required step for CalibBEV. Hence, we refer to the supplementary material for a further analysis. In the alignment modules, we set X, Y, and Z to 200, 8 and 200, respectively, spanning a 3D metric space of ±25m in both the forward-backward and left-right directions and of ±5m in up-bottom directions, with features dimension C to 128.

We set the batch size to 8 in all our experiments and train for 260k iterations using Adam [13] as optimizer. We use 1×10^{-4} as initial learning rate, which is decayed by a 0.5 factor every 51k iterations. Moreover, following [30], we initialize the learnable temperature parameter γ defined in Sec. 3.2 to $\log(1/0.07)$. Finally, we set the $\mathcal{L}_{clip}$ weighting factor α to 0.5.

As regards the Explicit Alignment module, after completely training the Implicit Alignment, both the new decoder and the prediction heads are initialized with the weights of their corresponding components from the previous step and both the backbones are frozen, as well as the Implicit Alignment decoder and prediction heads, leaving only the Explicit Alignment module parameters to be learnable. Hence, we train the Explicit Alignment module for an additional 120k iterations, setting the learning rate to 1×10^{-5} for all the learnable parameters and decayed, once again, by a 0.5 factor every 51k iterations. For all remaining hyper-parameters, we keep them as stated above.

4.3. Comparison with State-of-the-Art

In Tab. 1, we compare the performance of CalibBEV with the latest state-of-the-art methods according to the standard benchmark, which consists in sampling a random rotation along the vertical axis in the range ±360° and a random translation along the forward-backward and left-right directions in the range ±10m. We also follow [2, 20, 42] and do not remove the samples with large errors before computing the metrics, which would result in an unsuitable way to report the real performance. We refer to the supplementary material for this specific evaluation. On the KITTI dataset, CalibBEV outperforms ICLM, which is to the best of our knowledge the current state-of-the-art model on the KITTI dataset, by 0.16m on the average RTE, by 0.63° on the average RRE and by 2.47% on the registration accuracy. We also highlight CalibBEV is more robust compared to all previous works. Indeed, our proposal results in a reduction in the standard deviation of 0.11m and 1.82° on the RTE and RRE, respectively. Similarly, on the nuScenes dataset, CalibBEV outperforms the current state-of-the-art model GraphI2P by 0.45m, 1.19°, and 0.5% on the RTE, RRE, and registration accuracy, respectively. Once again, CalibBEV demonstrates superior robustness compared to all prior works, as it is able to reduce the standard deviation by 0.45m and by 1.18° on the RTE and RRE, respectively.

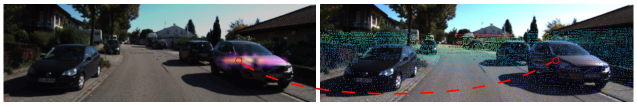

Figure 3. **2D-3D feature similarity heatmap.** For a given 3D point projected into the image plane (right), we highlight the most similar pixels in features space (left).

4.4. Ablation Studies

In this section, we conduct ablation studies to highlight how different parameters and architectural choices reflect on the registration task. We report the effectiveness of each contribution of our work in Tab. 2. Inspired by [18], we test a baseline architecture just composed by the modality-specific backbones, recovering one single global feature from the RGB and one from the point cloud, directly regressing the calibration from their concatenation. This ar-

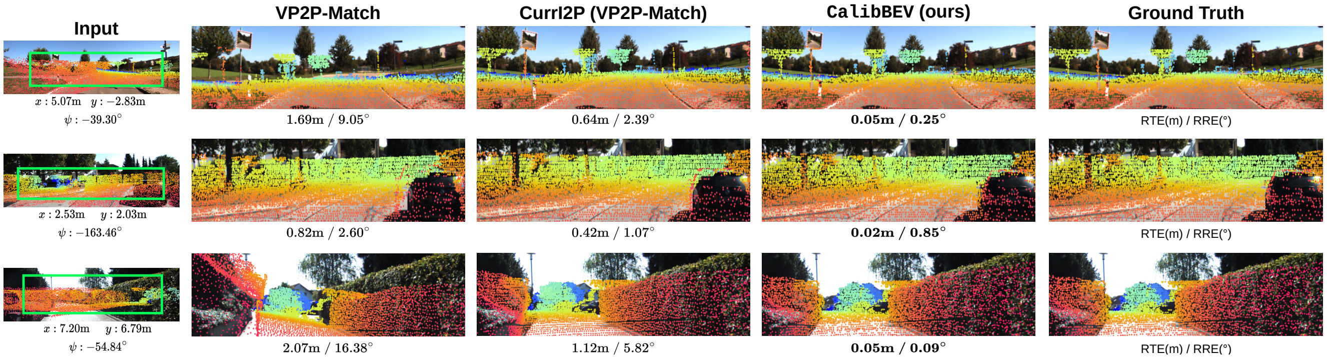

Figure 4. Qualitative comparison of Image-to-Point Cloud registration results on the KITTI dataset. From left to right: mis-aligned point cloud and image inputs, VP2P-Match calibration result, our model calibration result, and the ground truth camera-LiDAR alignment.

			KITTI		nuScenes	
Implicit	$\mathcal{L}_{clip}$	Explicit	RTE(m)↓	RRE(°)↓	RTE(m)↓	RRE(°)↓
			1.07	13.73	0.71	10.83
✓			0.04	1.61	0.07	1.48
✓	✓		0.05	1.17	0.04	1.16
✓	✓	✓	**0.04**	**0.61**	**0.04**	**0.54**

Table 2. Ablation studies of each component. Implicit: Implicit Alignment module, $\mathcal{L}_{clip}$: effect of CLIP loss between corresponding point-pixels pairs, Explicit: Explicit Alignment module.

chitecture suffers from a large error in both rotation and translation, caused by the lack of spatial information, serving only as a reference with respect to further methods. Then, we ablate the effect of the Implicit Alignment module (second row of Tab. 2) to gain spatial information by leveraging two BEV representations from both the RGB image and the LiDAR point cloud. We highlight that our first contribution already achieves comparable results to the state-of-the-art model on KITTI, while outperforming other methods on nuScenes. This result endorses our choice to leverage the geometry of the BEV representation as core for the calibration estimation.

Next, by adding a CLIP-like loss to encourage the similarity between RGB features and corresponding point features and vice versa, we are able to reduce the RRE by 0.44° and 0.32° on KITTI and nuScenes respectively (third row of Tab. 2). We argue that this result shows the effectiveness of a well-established loss function in enhancing the similarity between cross-domain features, leading to a positive impact on the `CalibBEV` performance by simplifying the decoder task. We further illustrate the effect of $\mathcal{L}_{clip}$ in Fig. 3, where a selected 3D point (circled in red, bottom figure) is projected onto the image plane for visualization. The most similar pixels in feature space are highlighted on the image (top figure). As shown, for a 3D point belonging to the car on the right side, the highest-scoring pixels consistently lie within the car region, demonstrating that both modality-specific networks learn semantically consistent features for corresponding objects across modalities.

		KITTI		
Train	Test	RTE(m)↓	RRE(°)↓	Acc.↑
L+R	R	0.04 ± 0.10	0.61 ± 0.52	99.96
L	R	0.03 ± 0.02	0.72 ± 0.61	99.89

Table 3. Analysis on `CalibBEV` generalization ability when trained on left camera images and tested on right ones.

		nuScenes		
Method	Num. Cams.	RTE(m)↓	RRE(°)↓	Acc.↑
`CalibBEV` (ours)	1	0.04 ± 0.08	0.54 ± 0.45	99.98
`CalibBEV` (ours)	6	**0.04** ± **0.03**	**0.28** ± **0.22**	**100.0**

Table 4. Multi-camera contribution analysis.

Lastly, we test our Explicit Alignment module, where we explicitly align the BEV geometries from the RGB and the point cloud as explained in Sec. 3.3. This contribution leads to an RRE reduction of 0.56° and 0.62° on KITTI and nuScenes, respectively (last row of Tab. 2), supporting our choice to exploit the realignment of features at spatial level through the BEV representation.

Generalization Analysis. In zero-shot generalization, different scenes or sensors inter-calibrations may weaken the registration performance if the model training leaned towards overfitting a particular sensor suite. On account of this problem, we decided to investigate the `CalibBEV` generalization ability when trained solely on the left camera images and evaluated on the right camera ones on the KITTI dataset. In Tab. 3, we show how `CalibBEV` is comfortably able to generalize, even when the camera position changes between training and testing. Indeed, the registration performance doesn't get undermined when the model is trained solely on the left camera images and evaluated on the right camera ones, compared to the `CalibBEV` trained on both the cameras images and evaluated on the right ones.

Method	KITTI		
	RTE(m)↓	RRE(°)↓	Acc.↑
CorrI2P [31]	0.96 ± 3.10	2.87 ± 4.58	85.76
Calibnet [11]	5.88 ± 2.80	10.92 ± 6.09	3.03
LCCNet [24]	0.40 ± 0.29	4.27 ± 3.70	75.75
CalibBEV (ours)	**0.10 ± 0.06**	**1.13 ± 0.62**	**99.96**

Table 5. 6DoF registration analysis. For each axis sample a rotation between ±20° and a translation between ±1.5m.

BEV	Range(m)	Cell(m)	KITTI		
			RTE(m)↓	RRE(°)↓	Acc.↑
200 × 200	±100	1.0	0.11	2.08	93.05
200 × 200	±75	0.75	0.10	1.82	96.61
200 × 200	±50	0.50	0.07	1.18	99.55
200 × 200	±25	0.25	**0.04**	**0.61**	**99.96**
400 × 400	±50	0.25	0.06	0.78	99.76

Table 6. BEV hyperparameters analysis.

Multi-camera Analysis. In Tab. 4, we explore the contribution for multi-camera configurations. Indeed, real-world applications and modern datasets such as [3, 22, 34, 36] are typically characterized by multi-camera setups, returning a 360° FOV of the scene surrounding the vehicle. Differently from previous works [2, 20, 42], CalibBEV is easily extendable to multi-camera configurations by simply adding few lines of modifications in the RGB to BEV projection. Specifically, the RGB backbone encodes each input image obtaining 2D features F_{rgb} of size $NC \times H/8 \times W/8 \times C_{rgb}$, with NC being the number of cameras. Subsequently, for each camera, these features are sampled and lifted from F_{rgb} to ${}^{3D}F_{rgb}$ as described in Sec. 3.2. Finally, following [7] we apply a weighted-sum for fusing the different volumes computed for each camera, taking into account for intersection between adjacent camera frustums by averaging the features projected in these regions. In this way, when dealing with multiple cameras, CalibBEV demonstrates better calibration performances, outperforming by 0.48% in RRE the single camera setup, while being more robust in all the metrics, highlighting a 100% registration accuracy.

6DoF Registration Analysis. Although this is not the standard benchmark for point-based methods, in Tab. 5 we show CalibBEV registration performances in a Six Degrees of Freedom (6DoF) scenario, where variations for the three spatial coordinates and rotation angles are involved. Following [24], for each axis we sample both a random rotation and translation in range ±20° and ±1.5m, respectively. CalibBEV outperforms the latest state-of-the-art point-based approach with publicly available training code CorrI2P [31] by 0.86m , 1.74°, and 14.2% on the RTE, RRE, registration accuracy, respectively, also demonstrating the CalibBEV ability to estimate the height by encoding it into the channels dimension. For a fair and complete analysis, we also compare CalibBEV against Calibnet [11] and LCCNet [24], two state-of-the-art projection-based approaches. We outperform Calibnet by 5.78m, 9.79°, and 96.93% on the RTE, RRE, registration accuracy, respectively, while we surpass LCCNet by 0.30m, 3.14°, and 24.21% on the RTE, RRE, registration accuracy, respectively. Once again, CalibBEV shows to be more robust compared to all previous works that have been tested.

BEV Resolution Analysis. In Tab. 6 we compare different 3D grid resolutions. We found an optimal setup by setting the cell dimension to 0.25m along the forward-backward and left-right directions. Specifically, we limit the BEV dimensions to 400 × 400, since higher resolutions would sacrifice the inference time.

Qualitative Analysis. A qualitative analysis on the KITTI dataset is shown in Fig. 4. For visualization, we first apply the predicted alignment transformation matrix to the point cloud, which is then projected onto the image plane through the known camera intrinsics parameters. We compare CalibBEV (fourth column from left) against the state-of-the-art models CurrI2P and VP2P-Match, and the ground truth (last column from left), reporting the RTE and RRE for each prediction. Instead, in the first column, we show the mis-registered image and point cloud inputs. Notably, we highlight the CalibBEV ability to achieve better alignment performance compared to CurrI2P in complex scenarios, in which the network might struggle to extract distinctive visual features. Indeed, by leveraging BEV features, our model demonstrates to have a better understanding of the geometrical relationships in the surrounding environment. Finally, we refer to the supplementary material for a further analysis on the nuScenes dataset and the failure cases.

5. Conclusion

In this paper we introduced CalibBEV, the first framework to cast the LiDAR-camera calibration problem as a BEV alignment task. Our key insight is to extract 3D BEV representations from both modalities and directly estimate the calibration matrix by aligning these representations through an Implicit Alignment module. Leveraging our unique BEV formulation, we further proposed an Explicit Alignment step, where the initial coarse estimate is used to warp the BEV features and refine the calibration matrix. Extensive experiments demonstrate that our approach achieves state-of-the-art performance, surpassing previous methods by a significant margin in terms of accuracy and robustness. Future works may include developing an end-to-end training pipeline to simultaneously optimize both the Implicit and Explicit Alignments, avoiding a two-step strategy.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. 3

[2] Lin Bie, Shouan Pan, Siqi Li, Yining Zhao, and Yue Gao. Graphi2p: Image-to-point cloud registration with exploring pattern of correspondence via graph learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22161–22171, 2025. 1, 2, 3, 5, 6, 8

[3] 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. 5, 8

[4] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation, 2022. 3

[5] Lei Cheng, Arindam Sengupta, and Siyang Cao. 3d radar and camera co-calibration: A flexible and accurate method for target-based extrinsic calibration, 2023. 2

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 5

[7] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What really matters for multi-sensor bev perception? In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1, 3, 4, 8

[8] 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. 3, 4

[9] Minghui Hou, Gang Wang, Zhiyang Wang, and Baorui Ma. Relai2p: Relational learning for image-to-point cloud registration. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 6

[10] Zhiwei Huang, Yikang Zhang, Qijun Chen, and Rui Fan. Online, target-free lidar-camera extrinsic calibration via cross-modal mask matching. *IEEE Transactions on Intelligent Vehicles*, 2024. 2

[11] 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)*. IEEE, 2018. 2, 8

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 5

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[14] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11301–11307. IEEE, 2023. 2

[15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1

[16] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem, 2009. 3

[17] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, 2013. 2

[18] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 15955–15964. IEEE, 2021. 1, 2, 6

[19] Jiaxin Li, Ben M. Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis, 2018. 3

[20] Xinjun Li, Wenfei Yang, Jiacheng Deng, Zhixin Cheng, Xu Zhou, and Tianzhu Zhang. Implicit correspondence learning for image-to-point cloud registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16922–16931, 2025. 1, 2, 3, 5, 6, 8

[21] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[22] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 8

[23] Liwei Lin, Chunyu Lin, Lang Nie, Shujuan Huang, and Yao Zhao. Curri2p: inter-and intra-modality similarity curriculum learning for image-to-point cloud registration. *The Visual Computer*, pages 1–14, 2025. 2, 3, 5, 6

[24] Xudong Lv, Boya Wang, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network, 2021. 2, 8

[25] Yanxin Ma, Yulan Guo, Jian Zhao, Min Lu, Jun Zhang, and Jianwei Wan. Fast and accurate registration of structured point clouds with small overlaps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016. 5

[26] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 26, 2012. 2

[27] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 1

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on

point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3, 4

[29] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 3

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 4, 5, 6

[31] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. Corri2p: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1198–1208, 2023. 1, 3, 6, 8

[32] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks, 2017. 2

[33] S. Sugimoto, H. Tateda, H. Takahashi, and M. 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 Vol.3, 2004. 2

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 8

[35] 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. 2

[36] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 8

[37] Yuxuan Xiao, Yao Li, Chengzhen Meng, Xingchen Li, Jianmin Ji, and Yanyong Zhang. Calibformer: A transformer-based automatic lidar-camera calibration network, 2024. 2

[38] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023. 1

[39] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 1

[40] 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. 2

[41] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 3

[42] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 3, 5, 6, 8