# BEVLOC: END-TO-END *6-DOF* LOCALIZATION VIA CROSS-MODALITY CORRELATION UNDER BIRD'S EYE VIEW

*Nanjie Chen[1], Jinping Wang[1], Hao Chen[2], Ying Shen[1], Shuai Wang[1], Xiaojun Tan[1,*]*

[1]School of Intelligent Systems Engineering, Sun Yat-sen University, China
[2]School of Computer Science, University of Birmingham, UK

## ABSTRACT

Accurate ego-centric localization assumes a paramount significance in the domain of autonomous driving. However, traditional methods for camera-LiDAR map localization rely on perspective projection to create a unified representation, which often falls short due to challenges such as occlusion and the sparse nature of point cloud data. Despite the recent surge in popularity of the Bird's-Eye-View (BEV) paradigm within autonomous driving, its potential applications in localization tasks have remained relatively underexplored. In response to this concern, this paper presents a pioneering end-to-end approach called the *BEV Localization Network via LiDAR Map* (BEVLoc). By fusing the image and LiDAR map in the BEV space via the concept of optical flow-based correlation, the BEVLoc framework can leverage the synergistic power of cross-modalities in localizing the vehicle. Experimental results conducted on the KITTI dataset highlight the efficacy and performance of BEVLoc in the realm of autonomous vehicle localization.

***Index Terms***— Visual localization, LiDAR map, Bird's Eye View, Optical flow, Pose estimation

## 1. INTRODUCTION

*6-DoF* localization serves as a cornerstone in the field of autonomous driving. One of the primary challenges in achieving this is the limitations of Global Navigation Satellite Systems (GNSSs), which are widely used but may not always meet the high precision requirements of navigation. This is particularly true in situations where signal drifts and blockages occur. To address these challenges, ego-centric localization methods have been proposed as a means of improving both the precision and robustness of localization by leveraging the sensors embedded within the ego-centric system.

Ego-perception systems, such as cameras and Light Detection And Ranging (LiDAR), are capable of accurately obtaining 3D position information and measuring the profiles of real-world surroundings. LiDAR-based methods [1, 2] are known for their ability to acquire rich landmark information. Although this geometric information can be effectively applied to navigation for path planning, LiDAR-based methods are not widely adopted due to their high cost and inability to withstand adverse weather conditions, e.g., rain and snow. In contrast, camera-based methods [3, 4] are more cost-effective and therefore more likely to be adopted on a broader scale. Historically, camera-based approaches have lagged behind LiDAR-based methods in terms of performance. However, recent advancements in camera-based methods have significantly narrowed this performance gap between the two.

Conventional customary procedure of map matching, involving the projection of LiDAR points onto the image plane, is not devoid of its limitations, as it may result in the omission of vital characteristics. The inherent dissimilarity in density between camera and LiDAR features becomes apparent, with less than 5% of camera features aligning with LiDAR points when employing a 32-channel LiDAR scanner (as indicated by [5]).

Recently, Bird's Eye View (BEV) based methods have experienced a surge in popularity, encompassing various tasks such as 3D object detection, semantic segmentation, and trajectory planning, thereby attracting significant attention. Notable advancements, such as LSS (as referenced in [6]) and BEVFormer (as cited in [7]), serve as exemplary techniques that have emerged in this domain. The application of BEV presents a compelling opportunity to extract insights from a bird's-eye view perspective.

By adopting this viewpoint, a more comprehensive understanding of the interplay between the ego-centric vehicle and its surrounding environment can be attained. However, the misalignment nature between sensor data and map information, which stem from accumulated errors resulting from untracked vehicle motion is critical for *6-DoF* localzation task.In the context of BEVLoc, our research endeavors to address this challenge by integrating optical flow [8, 9] with cross-modality BEV feature matching correlation, incorporating both LiDAR and camera data. The principal advancements stemming from our research can be encapsulated as follows:

1) Our research introduces the novel BEVLoc framework, meticulously designed to tackle the complex task of

*Corresponding author: Xiaojun Tan.

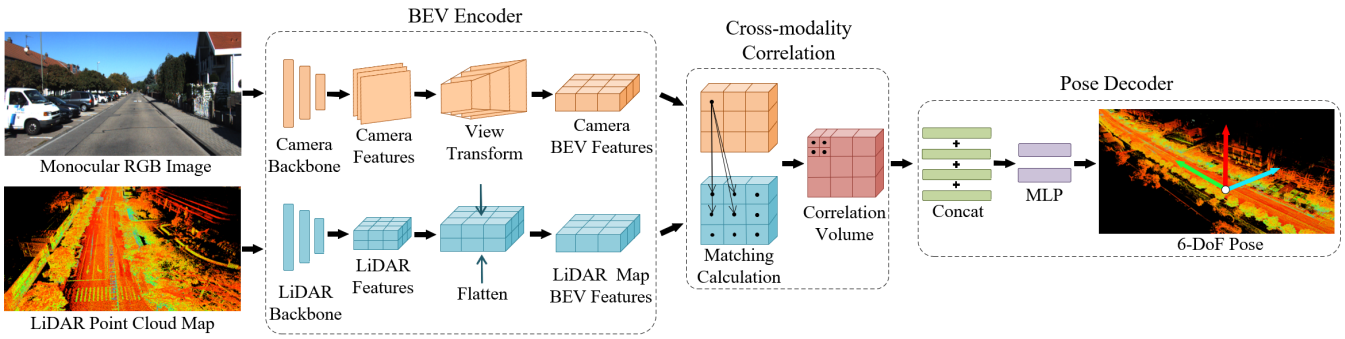



**Fig. 1**: The network architecture of BEVLoc, consisting of BEV Encoder, Cross-modality Correlation and Pose Decoder. First, we extract the monocular image feature and LiDAR point cloud map feature and transform them into BEV features. Then, the cross-modality correlation calculation of camera BEV feature and LiDAR map BEV feature is performed to obtain the correlation volume. At last, the pose regressor predicts the 6-DoF pose of the ego vehicle using the correlation volume.

pinpointing an ego-vehicle's position using a monocular camera.

2) We harnesses the power of optical flow techniques to bridge the gap between LiDAR point cloud data and camera images. This ingenious approach conquers the longstanding challenge of aligning these modalities by seamlessly unifying them in Bird's-Eye View (BEV) space.

3) Rigorous experimental evaluations serve as concrete proof of our model's effectiveness, demonstrating its superior performance when compared to the current state-of-the-art methods. Our research sets a new benchmark in the field of ego-vehicle localization, promising a future of safer and more accurate navigation.

## 2. METHODOLOGY

### 2.1. Problem Formulation

In this paper, we propose an end-to-end visual BEV localization network via LiDAR map named BEVLoc. Our goal is to estimate an optimal pose given an pre-built LiDAR map, an online monocular image and an initial pose. Let $\mathcal{M}$ represent the pre-built LiDAR map, $\mathcal{I}$ represent the monocular image and $\mathcal{P}_{init}$ represent the initial 6-DoF pose. And for a camera onboard, its intrinsics $\mathcal{K}$ and extrinsics $\mathcal{E}$ relative to the LiDAR are fixed. Our network $\mathcal{N}$ can estimate a pose offset $\Delta\psi$ through these few inputs, so that the initial pose $\mathcal{P}_{init}$ can be aligned with the ground truth $\mathcal{P}_{gt}$. In other words, we can define the network output as follows:

$$\begin{aligned} \Delta\psi &= \mathcal{N}(\mathcal{I}, \mathcal{M}, \mathcal{P}_{init}), \\ \mathcal{P}_{gt} &= \mathcal{P}_{init} \oplus \Delta\psi, \end{aligned} \tag{1}$$

where $\oplus$ is the pose composition operator.

### 2.2. BEV Encoder

BEV images include camera BEV image and LiDAR BEV image, which are generated in two pipelines respectively.

#### 2.2.1. *Camera BEV Image Generation*

Camera BEV image generation includes image feature extraction and view transform. First, the monocular image $\mathcal{I}$ is used as the input of the image feature extractor $\phi_{\mathcal{I}}$ to generate a feature map $\mathcal{F}_{\mathcal{I}} \in \mathbb{R}^{C\times H\times W}$:

$$\mathcal{F}_{\mathcal{I}} = \phi_{\mathcal{I}}(\mathcal{I}). \tag{2}$$

Then we transform the extracted range-view image feature map $\mathcal{F}_{\mathcal{I}}$ to BEV space through a view transformer $\mathcal{V}_{\mathcal{I}}$, which contains depth prediction module $\mathcal{V}_{\mathcal{IDP}}$ and feature pooling module $\mathcal{V}_{\mathcal{IFP}}$. $\mathcal{V}_{\mathcal{IDP}}$ predicts the depth distribution of features, then with the given camera intrinsics $\mathcal{K}$ and extrinsics $\mathcal{E}$, a 3D frustum point cloud in the ego coordinate system is generated. $\mathcal{F}_{\mathcal{I}}$ is scaled by outer product according to depth probability to generate 3D feature point cloud.

$$\mathcal{F}_{\mathcal{I}}^{\mathcal{V}_{\mathcal{IDP}}} = \mathcal{V}_{\mathcal{IDP}}(\mathcal{F}_{\mathcal{I}}, \mathcal{K}, \mathcal{E}). \tag{3}$$

At last, $\mathcal{V}_{\mathcal{IFP}}$ applies a pooling operation to flatten the frustum feature point cloud along the vertical direction. The 3D features are merged into the BEV feature $\mathcal{I}_{\mathcal{CB}} \in \mathbb{R}^{C_{BEV}\times H_{BEV}\times W_{BEV}}$:

$$\mathcal{I}_{\mathcal{CB}} = \mathcal{V}_{\mathcal{IFP}}(\mathcal{F}_{\mathcal{I}}^{\mathcal{V}_{\mathcal{IDP}}}). \tag{4}$$

#### 2.2.2. *LiDAR BEV Image Generation*

LiDAR-to-BEV projection obtains BEV images of sparse point clouds by flattening LiDAR features $\mathcal{F}_{\mathcal{L}}$ along the height direction. First, the LiDAR map points $\{p_i | i = 1, 2, ..., n\}$ are fed into a voxel extractor $\phi_{\mathcal{L}}$. Then, a pillar

feature 3D backbone $\mathcal{V}_{\mathcal{L}}$ is used to generate BEV pseudo-images $\mathcal{I}_{\mathcal{LB}}$. In this way, we get a unified representation of the camera and LiDAR map in the BEV space:

$$\begin{aligned} \mathcal{F}_{\mathcal{L}} &= \phi_{\mathcal{L}}(p_i), \\ \mathcal{I}_{\mathcal{LB}} &= \mathcal{V}_{\mathcal{L}}(\mathcal{F}_{\mathcal{L}}). \end{aligned} \tag{5}$$

### 2.3. Cross-Modality Correlation

Our cross-model correlation module exploits the concept of optical flow to compute the similarity of two BEV images, associating the image and map to predict ego-pose. For the two input BEV images $\mathcal{I}_{\mathcal{CB}}$ and $\mathcal{I}_{\mathcal{LB}}$ from formula (4) and (5), we first use a CNN pyramid $\phi_{\mathcal{BEV}}$ with three convolutional layers to downsample and aggregate information, obtaining $f_I$ and $f_L$ representing image and LiDAR features respectively. Next, the cross-model correlation is defined as a cost volume that stores the matching costs for associating the corresponding features as follows:

$$\begin{aligned} f_I = \phi_{\mathcal{BEV}}(\mathcal{I}_{\mathcal{CB}}), f_L = \phi_{\mathcal{BEV}}(\mathcal{I}_{\mathcal{LB}}), \\ cv({f_I}^i, {f_L}^j) = \frac{1}{\mathcal{N}}(c({f_I}^i))^{\intercal} c({f_L}^j), \end{aligned} \tag{6}$$

where $cv({f_I}^i, {f_L}^j)$ is the cost volume and $\mathcal{N}$ is the length of the column vector $c({f_I}^i)$, $c$ is columnize operation, ${f_I}^i$ is one feature pixel in $f_I$. For each feature pixel, the cost volume only needs to calculate its correlation with feature pixels within a certain range $d$ around the same position in $f_L$. Because after down-sampling, a one pixel offset at the top level corresponds to $2^{L-1}$ pixels at the full resolution BEV image. Thus the range $d$ can be set to be small. The cost volume shape is $d^2 \times H_{3th} \times W_{3th}$, where $H_{3th}$ and $W_{3th}$ are the height and width of the 3th layer features, respectively.

### 2.4. Pose Decoder

The correlation filter establishes correspondence cost between two BEV images, $\mathcal{I}_{\mathcal{CB}}$ and $\mathcal{I}_{\mathcal{LB}}$. We apply a series of convolution layers with stride 1 and concat layers on channel to obtain cost volume features that fuse multi-level information. Subsequently, several fully connected layers are utilized to predict the translation $\mathcal{T}$ along the xyz directions and the quaternion $\mathcal{Q}$ for rotation, effectively representing the 6-DoF camera pose. To ensure the predicted translation and rotation fall within appropriate ranges, we include an additional tanh layer as an output layer. Finally, the network outputs the deviation between the current estimated pose and the initial pose:

$$[\mathcal{T}, \mathcal{Q}] = \mathcal{PD}(cv(f_I, f_L)). \tag{7}$$

## 3. EXPERIMENTS

In this section, we conduct experiments on the KITTI Odometry dataset [10] and choose CMRNet [4] and HyperMap [11] as our baseline. CMRNet, proposed in 2019, is the first CNN-based approach that registers monocular images to 3D LiDAR map. Then in 2021, HyperMap was proposed, which is the latest and more accurate LiDAR map localization algorithm. And because map metric localization methods involve different sensors and map, we also compare the overall accuracy of different methods [12, 13, 14, 15].

### 3.1. Experimental Details

#### *3.1.1. Dataset Preprocessing*

The point cloud map of KITTI Odometry dataset, the ground truth poses and the initial poses of the validation set are provided by CMRNet. The training set is the sequences of 03, 05, 06, 07, 08, 09 in KITTI Odometry dataset. In order to compare with other algorithms on the same benchmark, we selected the representative 00 sequence (4541 frames) with a large scene range as the validation set. Due to the insufficient accuracy of RTK(Real-Time Kinematic) ground truth provided by the KITTI dataset, it can result in map discontinuity in loop closures. So we used the ground truth poses of CMRNet optimized by LiDAR SLAM.

#### *3.1.2. Input and output*

The camera-LiDAR map localization task uses a camera image and a LiDAR point cloud map as input and an estimated global pose of the vehicle as output. The localization output is generated in the ego vehicle system, measuring the 6DoF pose. We use Smooth L1 loss for translation and quaternion angular distance loss for rotation as CMRNet.

#### *3.1.3. Training Settings*

We implement our proposed framework using PyTorch on a NVIDIA 3080 GPU. All the models are trained using learning rate $5e-5$ and batch size 4 with Adam optimizer. Because the image sizes in the KITTI dataset are inconsistent, all images are padded to $1280 \times 384$.

For the camera encoder, we use ResNet34 [16] pretrained on ImageNet [17] as the visual backbone to extract image features. Then we adopt the visual transformation paradigm of LSS [6] to establish the correspondence between the image and horizontal BEV plane. The BEV features $I_{CB} \in \mathbb{R}^{64\times128\times256}$ represents a spatial range of $[32m \times 16m]$ around the vehicle and the BEV gird size is $0.125m$. For the LiDAR encoder, we use FC-64-64 with BatchNorm as a lightweight PointNet [18]. And a modified PointPillars [19] builds the pillars sparsely. The spatial range of point cloud BEV feature is consistent with the setting of camera BEV feature, so that correlation calculation can be performed through the optical flow net built on the basic structure of PWC-Net [20] . And the correlation range $d$ is set to 4.

During training, we need to simulate random initial poses bias online. Specifically, we sample the random translation deviation within $[-2m, -2m]$ in $xyz$ directions, and rotation deviation within $[-10°, 10°]$ about $xyz$ axes for the baseline experiments. While for the testing time, the initial poses are

fixed. The LiDAR point cloud map is first transformed to the vehicle coordinate system and then translated by these initial deviations. The task of the network is to predict accurate deviations from a biased map using a image. Therefore, the imposed initial pose bias can serve as supervision, enabling the network to be trained end-to-end.

### 3.2. Results

#### *3.2.1. Accuracy performance*

We verified that BEVLoc has better accuracy than the baseline, both in terms of rotation and translation. Our method, as shown in Table 1, outperforms the baseline with the same initial pose errors, achieving low translation and rotation errors. Table 2 shows the comparison of the proposed BEVLoc against other existing map metric localization algorithms. Our method achieves substantially lower errors on translation and comparable results on rotation. It can be seen that BEVLoc has the best localization accuracy on the KITTI dataset, indicating that BEV space can provide more information constraints than matching methods using perspective projection. In addition, our method also demonstrates the feasibility of the monocular visual localization problem as a subtask of existing BEV-based autonomous driving large models.

**Table 1**: Comparison with baseline. Our result is the same as the baseline to take the median localization error for comparison. The initial translation and rotation deviation are $[-2m, 2m]$, $[-10°, 10°]$, respectively.

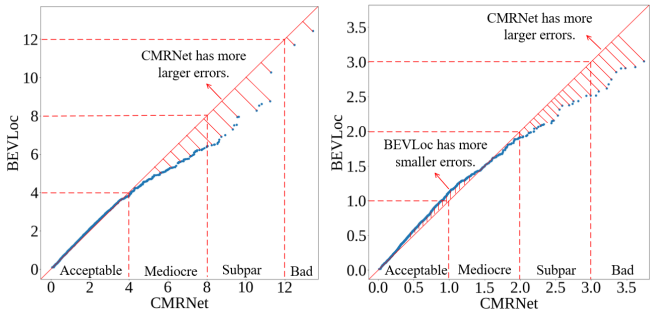
| Method | Translation ($m$) | Rotation (°) |
|---|---|---|
| CMRNet [4] | 0.51 | 1.39 |
| HyperMap [11] | 0.48 | 1.42 |
| BEVLoc (Ours) | **0.39** | **1.28** |

**Table 2**: Average localization error comparison with existing map metric localization methods.

| Method | Map | Sensors | Translation ($m$) | Rotation (°) |
|---|---|---|---|---|
| Brubaker et al. 2015 [12] | Open Street | Monocular | 16.0 | 2.00 (Yaw) |
| Brubaker et al. 2015 [12] | Open Street | Binocular | 2.10 | 1.20 (Yaw) |
| Elhousni et al. 2022 [13] | Open Street | LiDAR | 1.37 | 1.15 |
| Miller et al. 2021 [14] | Satellite | LiDAR | 2.00 | N/A |
| Zuo et al. 2020 [15] | Point Cloud | Binocular | 0.47 | **0.87** |
| BEVLoc (Ours) | Point Cloud | Monocular | **0.44** | 1.68 |

Figure 2 illustrates the quantile-quantile plot for the rotation and translation components. For rotation errors, the data distribution is below the reference line and has a tendency to curve upward, which means that the quantiles of BEVLoc are smaller and more densely clustered at smaller values. It indicates that the distribution of BEVLoc is more concentrated with smaller variance compared to CMRNet. And for translation errors, the data distribution has a downward curving trend, meaning that the distribution of BEVLoc has shorter tails, i.e. fewer extreme values.

BEVLoc greatly reduces the deviation of initial rotation and translation, validating that the model can handle large initial errors and predict accurate pose offsets.




(a) Rotation Errors (b) Translation Errors

**Fig. 2**: Quantile-Quantile Plot of BEVLoc errors distribution relative to CMRNet.

#### *3.2.2. Effect of BEV grid size*

BEV grid size determines the resolution of the BEV feature image, so we further explore the effect of different BEV grid sizes on the model results in Table 3. We can observe that a smaller grid size improves the prediction accuracy, which can be interpreted as higher resolution provides more information on location constraints. But on the other hand, higher resolution will increase the model parameters and increase the computational burden.

**Table 3**: Effect of different BEV grid sizes on BEVLoc.

| BEV grid size ($m$) | Translation ($m$) | Rotation (°) |
|---|---|---|
| 0.5 | 0.53 | 1.50 |
| 0.2 | 0.41 | 1.37 |
| 0.125 | 0.39 | 1.28 |

#### *3.2.3. Evaluation of runtime*

In order to verify the real-time performance of the algorithm, we calculate the average runtime of the model processing one frame. The rumtime performance of BEVLoc is evaluated based on the model processing the entire KITTI 00 sequence. The average runtime of one frame is 33.7 $ms$, achieving about 30 fps. This shows that the model can support the real-time localization requirements of the real scene.

## 4. CONCLUSION

In this paper, we present BEVLoc, an end-to-end 6-DoF localization network via cross-modality correlation under BEV, integrating localization task with the popular BEV paradigm in autonomous driving. We first describe the construction of two modal BEV features, representing real-time images and pre-built LiDAR map, respectively. Then, inspired by the idea of optical flow, we calculate the matching cost volume of the two modalities. Based on the matching cost, the network predicts the global ego pose in the map. Furthermore, experiments on the KITTI dataset demonstrate the accuracy of BEVLoc. In the comparison of state-of-art localization algorithms, BEVLoc achieves better performance.

## 5. REFERENCES


[1] Letian Zhang, Jinping Wang, Lu Jie, Nanjie Chen, Xiaojun Tan, and Zhifei Duan, "Lmbao: A landmark map for bundle adjustment odometry in lidar slam," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[2] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song, "L3-net: Towards learning based lidar localization for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6389–6398.

[3] Mengdan Feng, Sixing Hu, Marcelo H Ang, and Gim Hee Lee, "2d3d-matchnet: Learning to match keypoints across 2d image and 3d point cloud," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4790–4796.

[4] Daniele Cattaneo, Matteo Vaghi, Augusto Luis Ballardini, Simone Fontana, Domenico G Sorrenti, and Wolfram Burgard, "Cmrnet: Camera to lidar-map registration," in *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 1283–1289.

[5] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[6] 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*. Springer, 2020, pp. 194–210.

[7] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*. Springer, 2022, pp. 1–18.

[8] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao, "Gmflow: Learning optical flow via global matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8121–8130.

[9] Zachary Teed and Jia Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.

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

[11] Ming-Fang Chang, Joshua Mangelson, Michael Kaess, and Simon Lucey, "Hypermap: Compressed 3d map for monocular camera registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11739–11745.

[12] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun, "Map-based probabilistic visual self-localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 652–665, 2015.

[13] Mahdi Elhousni, Ziming Zhang, and Xinming Huang, "Lidar-osm-based vehicle localization in gps-denied environments by using constrained particle filter," *Sensors*, vol. 22, no. 14, pp. 5206, 2022.

[14] Ian D Miller, Anthony Cowley, Ravi Konkimalla, Shreyas S Shivakumar, Ty Nguyen, Trey Smith, Camillo Jose Taylor, and Vijay Kumar, "Any way you look at it: Semantic crossview localization and mapping with lidar," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2397–2404, 2021.

[15] Xingxing Zuo, Wenlong Ye, Yulin Yang, Renjie Zheng, Teresa Vidal-Calleja, Guoquan Huang, and Yong Liu, "Multimodal localization: Stereo over lidar map," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1003–1026, 2020.

[16] 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*, 2016, pp. 770–778.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[19] 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*, 2019, pp. 12697–12705.

[20] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.