Progressive Camera-LiDAR Adaptation for Scene Flow Estimation

Ting Han[1], Yang Luo[2], Siyu Chen[2], Xiangyi Xie[1], Chaolei Wang[1], Hongchao Fan[3], Yiping Chen[1,*]

[1] Sch. of Geospatial Eng. & Sci., SYSU, Zhuhai, China - (hant23, xiexy99, wangchlei5, chenyp79)@mail.sysu.edu.cn
[2] Sch. of Comp. Eng., JMU, Xiamen, China - (yangl, chensy)@jmu.edu.cn
[3] Dept. of Civ. & Environ. Eng., NTNU, Trondheim, Norway – hongchao.fan@ntnu.no

Keywords: LiDAR Point Cloud, Camera-LiDAR Fusion, Scene Flow, Autonomous Driving, Remote Sensing

Abstract

3D scene flow aims to recover the dense geometry and 3D motion of dynamic scenes. This paper explores the transformation and adaptation of the 2D-3D feature space in the joint estimation of optical flow and scene flow. Our key insight is to fully leverage the unique characteristics of each modality and maximize their inter-modality complementarity. To achieve this, we propose a novel architecture, named PAFlow, which consists of Camera-LiDAR Adaptation and Spatial Characteristics Adaptation. PAFlow achieves an error of 4.23% on real-world KITTI Scene Flow benchmark, with significantly fewer parameters compared to previous methods. This study will support dynamic scene understanding for the geospatial community.

1. Introduction

Scene flow describes the 3D motion field of the dynamic scene, and optical flow is the pixel-level motion across camera frames (Zhai et al., 2021). Both are critical for high-level scene understanding tasks in remote sensing and geospatial computer vision (Vedula et al., 2005, Menze et al., 2015, Menze et al., 2018), as they enable a comprehensive understanding of the dynamics of the scene.

With the advancement of deep learning, earlier approaches (Behl et al., 2017, Ma et al., 2019, Yang and Ramanan, 2021) have employed convolutional neural networks for optical and scene flow estimation. However, these methods typically process camera frames and point clouds independently, failing to exploit the complementary advantages of both modalities. Moreover, the modular nature of these architectures means that any limitation in an individual module can negatively impact overall performance.

Some recent studies (Rishav et al., 2020, Teed and Deng, 2021, Han et al., 2024a, Chen et al., 2025) adopt a feature-level fusion strategy that combines camera images with dense depth maps to predict 3D motion. However, this fusion pipeline struggles to effectively utilize the full extent of 3D structural information. LiDAR (Light Detection and Ranging) is a technique that acquires data by emitting pulsed laser beams toward a target and measuring the reflected signals to determine the distance. Many studies (Luo et al., 2025) have demonstrated that LiDAR is highly robust against various visual noise and serves as a valuable complement to monocular camera frames by providing precise 3D geometric information. Although DeepLiDARFlow (Rishav et al., 2020) integrates images and LiDAR point clouds by projecting them onto a 2D plane for multi-source fusion, this process leads to information loss and accumulates errors in subsequent stages. Consequently, this approach continues to face challenges in enabling effective interaction between the two modalities.

CamLiFlow (Liu et al., 2022) inspired a point-based branch to process point clouds, enabling the extraction of fine-grained 3D geometric information without voxelization or projection. Subsequent methods (Peng et al., 2023, Liu et al., 2024) have adopted a dual-branch structure to handle both LiDAR point clouds and camera frames, incorporating a learnable module to connect the two branches. However, due to the sparsity and varying density of point clouds, there remains a significant discrepancy between the distribution of points and the corresponding image pixels. Although some approaches attempt to remove outliers, they still fail to provide an effective fusion mechanism for integrating LiDAR point clouds and images.

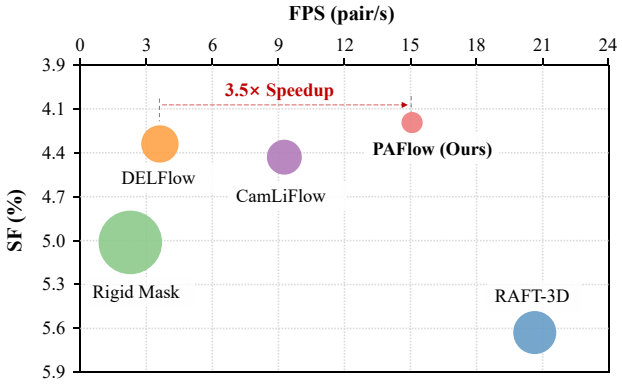

Figure 1. Performance vs. Speed. PAFlow achieves the best performance while maintaining competitive efficiency.

We provide insights into two key challenges that hinder the effective integration of LiDAR and visual information. First, raw LiDAR point clouds and camera frames exist in different spatial spaces, making it challenging to define a suitable space for seamless data fusion. Second, integrating features extracted from LiDAR and visual images is also complex, as they are represented in fundamentally different forms, which can lead to inconsistencies in feature alignment and fusion.

Given the synergy between optical flow estimation and scene flow estimation, we aim to fully integrate and complement the 2D-3D data from camera and LiDAR frames. To this end, we propose **PAFlow**, which consists of two components, **Camera-LiDAR Adaptation (CLA)** and **Spatial Characteristics Adaptation (SCA)**. These components explore the fusion of the

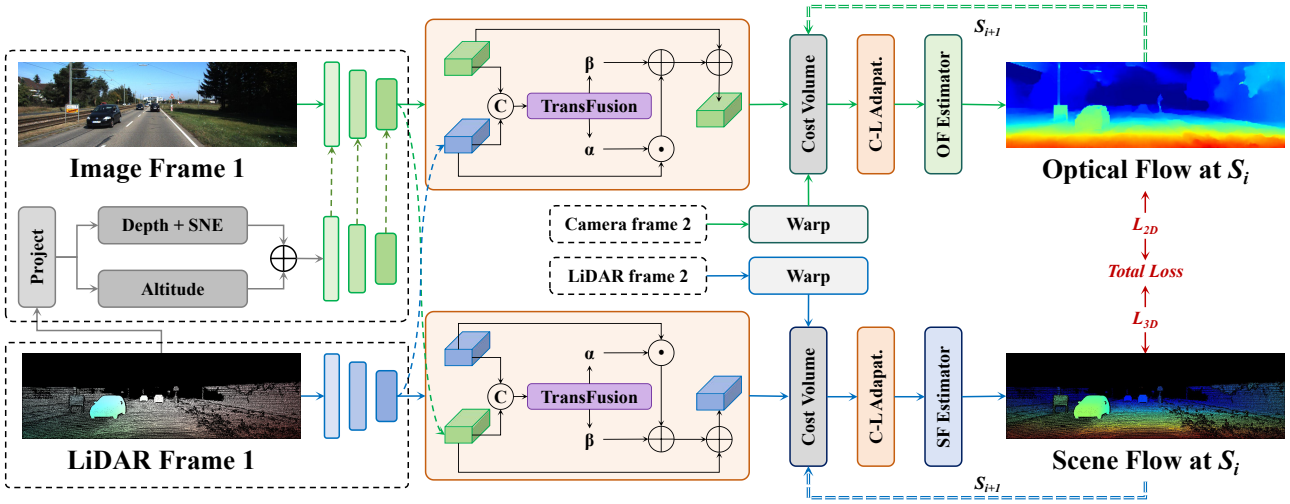

Figure 2. The architecture of PAFlow, which takes camera and LiDAR frames as input to jointly estimate optical flow and scene flow.

feature spaces of point clouds and images, as well as the adaptation of spatial properties in the camera frames. The effectiveness of our design is verified by experiments on widely used FlyingThings3D and KITTI Scene Flow benchmark. Besides, the individual components of our design are also verified by extensive experiments. As shown in Fig. 1, experiments demonstrate that our approach achieves better performance with much fewer parameters on the FlyingThings3D and KITTI Scene Flow benchmark. The main contributions of this work are as follows:

- We introduce a novel camera-LiDAR adaptation framework, named PAFlow, designed for progressive optical flow and scene flow estimation. Our approach is highly flexible and can be seamlessly integrated into various network architectures.
- We propose Camera-LiDAR Adaptation (CLA) and Spatial Characteristics Adaptation (SCA) to align data and features between camera frames and LiDAR point clouds, ensuring better complementarity and mutual enhancement between the two modalities.
- Extensive experiments on the FlyingThings3D and KITTI datasets demonstrate that PAFlow achieves strong and robust performance, outperforming existing methods in both accuracy and efficiency.

2. Related Work

With the advancement of deep learning in 3D point cloud processing (Qi et al., 2017a, Qi et al., 2017b, Han et al., 2024b), FlowNet3D (Liu et al., 2019) is one of the pioneering approaches to directly process point clouds for 3D scene flow estimation in an end-to-end manner. PointPWC-Net (Wu et al., 2019) further extends this idea by a patch-to-patch matching method, which considers multiple points from the first frame during correlation and incorporates a coarse-to-fine structure inspired by optical flow estimation (Sun et al., 2018) into scene flow estimation. HPLFlowNet (Gu et al., 2019) introduces a series of operations to restore rich geometric information from point clouds.

However, interpolation from point clouds to the permutohedral lattice inevitably causes information loss. To better assign different weights to the correlated points within a patch, HALFLOW (Wang et al., 2021) proposes a hierarchical neural network with a dual-attentive embedding layer. Then, FLOT (Puy et al., 2020) achieves competitive performance with significantly fewer parameters using optimal transport techniques.

Recently, the rigidity assumption has also been widely adopted in several works (Ma et al., 2019, Menze et al., 2015). HCRFFlow (Li et al., 2021b) formulates the rigidity constraint as a high-order term to improve scene flow estimation. Inspired by the architecture of RAFT (Teed and Deng, 2020), RAFT-3D (Teed and Deng, 2021) uses rigid motion embeddings to cluster neighboring points into rigid objects and refined the 2D flow field using a recurrent structure. Instead, we focus on leveraging the 3D information from LiDAR point clouds to optimize optical flow estimation and generate robust 3D scene flow predictions, while also exploring a more effective fusion strategy between camera frames and LiDAR data.

3. Method

Given a pair of synchronized camera image frames $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ and corresponding LiDAR point clouds $\mathcal{P} \in \mathbb{R}^{N\times 3}$, our PAFlow jointly estimates dense optical flow and sparse scene flow through a progressive interactive dual-branch architecture, as shown in Fig. 2.

3.1 Overview Architecture

Our PAFlow consists of pyramid stages including Feature Extraction, Warping, Cost Volume, Camera-LiDAR Adaptation, and Flow Estimation. Within each stage, the image frame and LiDAR features are extracted in separate branches and are fused in an adaptation module to pass complementary information. To be specific, the image frames and LiDAR point cloud are downsampled using ResNet and PointConv, respectively, to extract visual textural information and spatial geometric information. At each stage, those features are warped towards the reference frames. Next, we employ cost volume to store the matching costs using 4-neighbor around each pixel and learnable layer for image and point cloud, respectively. Then, we design the Camera-LiDAR Adaptation module to fuse the two cost volumes. Finally, following CamLiFlow, we build optical

Method	Input	2D Metrics		3D Metrics		
		EPE_{2D} ↓	ACC_{1px} (%) ↑	EPE_{3D} (10^2) ↓	$ACC_{.05}$ (%) ↑	$ACC_{.10}$ (%) ↑
FlowNet2.0 (Ilg et al., 2017)	RGB	5.05	72.8	-	-	-
PWC-Net (Sun et al., 2018)	RGB	6.55	64.3	-	-	-
RAFT (Teed and Deng, 2020)	RGB	3.12	81.1	-	-	-
FlowNet3D (Liu et al., 2019)	LiDAR	-	-	16.9	25.4	57.9
PointPWC (Wu et al., 2019)	LiDAR	-	-	13.2	44.3	67.4
OGSF-Net (Ouyang and Raviv, 2021)	LiDAR	-	-	16.3	-	-
RAFT-3D (Teed and Deng, 2021)	RGB + Depth	2.37	87.1	9.4	80.6	-
CamLiFlow (Liu et al., 2022)	RGB + LiDAR	2.18	87.3	6.1	85.6	91.9
DELFlow (Peng et al., 2023)	RGB + LiDAR	**2.02**	**85.9**	5.8	86.7	93.2
PAFlow (Ours)	RGB + LiDAR	2.07	86.1	**5.6**	**87.0**	**93.5**

Table 1. Quantitative results compared with recent methods on the FlyingThings3D dataset. The performance are evaluated on all point. The best results are in **bold**.

Method	Input	D1 (%) ↓	D2 (%) ↓	Fl (%) ↓	SF (%) ↓	Param. (M)	FPS (pair/s) ↑
PRSM (Vogel et al., 2015)	Stereo	4.27	6.79	6.68	8.97	-	
SSF (Ren et al., 2017)	Stereo	4.42	7.02	7.14	10.07	-	-
Sense (Jiang et al., 2019)	Stereo	2.22	5.89	7.64	9.55	13.4	16.6
DRISF (Ma et al., 2019)	Stereo	2.55	4.04	4.73	6.31	58.9	-
ACOSF (Li et al., 2021a)	Stereo	3.58	5.31	5.79	7.90	-	-
M-FUSE (Mehl et al., 2023)	Stereo	**1.65**	3.13	3.46	4.83	1.8	6.2
RigidMask (Yang and Ramanan, 2021))	Stereo + LiDAR	1.89	3.23	3.50	4.89	145.3	2.4
Scale-flow (Ling et al., 2022)	Mono	1.81	3.51	5.32	6.94	41.4	**40**
RAFT-3D (Teed and Deng, 2021)	Mono + Depth	1.81	3.67	4.29	5.77	51.3	20
OpticalExp (Yang and Ramanan, 2020)	Mono + LiDAR	1.81	4.25	6.30	8.12	-	-
DELFlow (Peng et al., 2023)	Mono + LiDAR	**1.65**	2.84	3.07	4.34	20.1	3.9
CamLiFlow (Liu et al., 2024)	Mono + LiDAR	1.81	2.95	3.10	4.43	19.7	9.5
PAFlow (Ours)	Mono + LiDAR	1.81	**2.81**	**2.98**	**4.23**	8.5	15.4

Table 2. Performance comparison in the KITTI Scene Flow benchmark, where the best results are in **bold**.

flow estimator and scene flow estimator for each modality using DenseNet and PointConv. We supervise optical flow estimation and scene flow estimation respectively and utilize multi-task loss for joint optimization.

3.2 Camera-LiDAR Adaptation

Both LiDAR and visual frames exist in different spaces, making it extremely challenging to directly and smoothly integrate visual frames with point clouds. The Camera-LiDAR Adaptation module takes image features $F^I \in \mathbb{R}^{h \times w \times C^I}$ and point cloud features $F^P \in \mathbb{R}^{N \times C^P}$ as input, transforming the feature space to enable better complementarity in a learning-based manner. Specially, we assume that the linear transformation δ is able to properly define a feature space adaptation. We construct the transformation function $\delta : \mathcal{P} \to \mathcal{I}$ as $\delta(F^P) = \alpha F^I + \beta$, where α and β denote the scalar vector and offset vector. To estimate α and β, we introduce a fully convolutional operation to learn the weight parameters from the fused features $(F^I \biguplus F^P)$. Similarly, we apply the same approach $\varphi : \mathcal{I} \to \mathcal{P}$ to transform the visual feature space into the point cloud feature space. In this way, the layer-wise visual features and LiDAR features are adapted to each other.

3.3 Spatial Characteristics Adaptation

To align the relationship between image pixels and the real-world space, we project the point clouds into altitude difference $\mathcal{A}$ and depth images $\mathcal{D}$ using calibration parameters to generate auxiliary frame. Moreover, we apply surface normal estimation $\eta : \mathcal{D} \to \mathcal{D}_n$ to calculate the normal information of the pixels in the depth images, where $\eta : n_i = [f_x g_x, f_y g_y, -\frac{f_x \Delta X_i g_x + f_y \Delta Y_i g_y}{\Delta Z_i}]$, f_x, f_y are from calibration parameters, $[X, Y, Z]$ represents the 3D point coordinate, and g_x, g_y are the horizontal and vertical image gradient. These auxiliary frame features are then fused with image features to represent the geometric-based visual frames.

3.4 Warping and Cost Volume

At each pyramid layer, both image features and point clouds are warped toward the reference frame using the upsampled flow from the lower layer. Since the warping process does not introduce any learnable parameters, we do not apply feature fusion immediately after this stage. The cost volume stores the matching costs between the reference frame and the warped target frame. For the image branch, we construct a partial cost volume by restricting the search range to four pixels around each pixel. In contrast, for the point branch, we design a learnable cost volume layer. While the pixel-based 2D cost volume maintains a fixed neighborhood search range, the point-based 3D cost volume adapts dynamically to varying point distributions. To effectively leverage the complementary nature of both modalities, we integrate these two cost volumes using our Camera-LiDAR Adaptation (CLA) module.

3.5 Training Objective

During optimization, we supervise optical flow estimation and scene flow estimation, respectively, and utilize multi-task loss for joint optimization, following (Liu et al., 2024): $\mathcal{L}_{2D} = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} \|f_{2D}^l(\mathbf{x}) - \hat{f}_{2D}^l(\mathbf{x})\|_2$ and $\mathcal{L}_{3D} = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{p}} \|\mathbf{f}_{3D}^l(\mathbf{p}) - \hat{\mathbf{f}}_{3D}^l(\mathbf{p})\|_2$ where $\hat{f}_{2D}^l$ and $\hat{f}_{3D}^l$ are the ground truth optical flow and scene flow in the l-th layer, respectively. $\|\cdot\|_2$ computes the L_2 norm. For fine-tuning, we

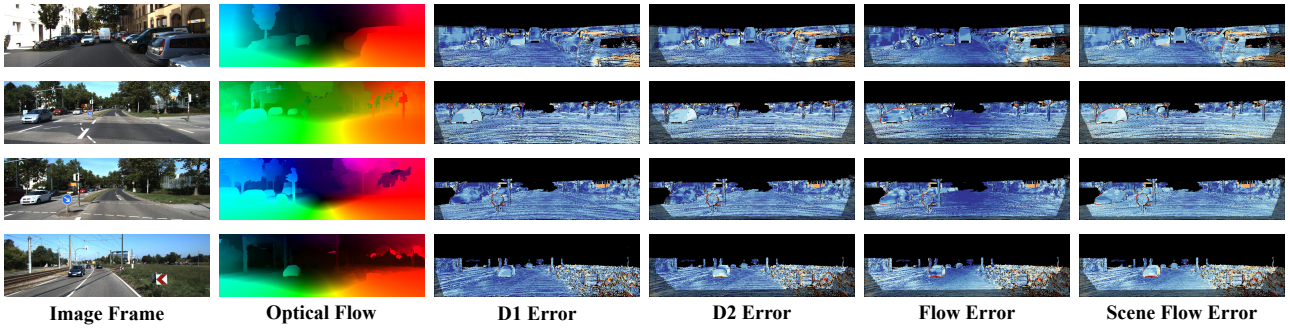

Figure 3. Qualitative results on the KITTI Scene Flow test set, where blue indicates a low error and red indicates a higher error.

use the training loss as:

$$\mathcal{L}_{2D} = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} (|f_{2D}^l(\mathbf{x}) - \hat{f}_{2D}^l(\mathbf{x})| + \epsilon)^q, \quad (1)$$

$$\mathcal{L}_{3D} = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} (|f_{3D}^l(\mathbf{x}) - \hat{f}_{3D}^l(\mathbf{x})| + \epsilon)^q, \quad (2)$$

where $|\cdot|$ computes the L_1 norm, $q = 0.4$ gives less penalty to outliers and ϵ is set to 0.01. The final loss is a weighted sum of the losses defined above:

$$\mathcal{L} = \mathcal{L}_{2D} + \lambda \mathcal{L}_{3D}, \quad (3)$$

where λ is set to 1.0 for all our experiments.

4. Experiments

4.1 Datasets

We evaluate our method on the synthetic dataset FlyingThings3D (Mayer et al., 2016) and the real-world dataset KITTI (Menze et al., 2015). FlyingThings3D consists of stereo and RGB-D images rendered with multiple randomly moving objects. The training and validation set contains 19,640 and 3,824 pairs of camera-LiDAR frames, respectively. KITTI is a real-world and challenging benchmark for autonomous driving, which consists of 200 training scenes and 200 test scenes. We divide the 200 training images into train and validation splits based on the $4:1$ ratio. The ground truth disparity maps are lifted into point clouds using the calibration parameters.

4.2 Metrics

For FlyingThings3D, we evaluate the performance using 2D and 3D end-point error (EPE), as well as threshold metrics ACC_{1px}, $\text{ACC}_{.05}$, and $\text{ACC}_{.10}$, which measure the portion of error within a threshold.

4.3 Configurations and Settings

All experiments are conducted on a machine with two NVIDIA RTX 3090 GPUs. We train our model on 800 epochs with a batch size of 4 and AdamW optimizer. The initial learning rate of optical flow and scene flow are set to $1e-4$ and $1e-3$, respectively, and decay with a decaying rate of 0.9. We apply various data augmentation methods, such as color jitter, random horizontal flipping, random scaling, and random cropping.

Configuration		3D Metrics		
CLA	SCA	EPE_{3D} $(10^2)\downarrow$	$\text{ACC}_{.05}$ (%) $\uparrow$	$\text{ACC}_{.10}$ (%) $\uparrow$
-	-	6.1	85.6	91.9
✓	-	5.8	86.3	92.8
-	✓	5.8	86.1	92.3
✓	✓	5.6	87.0	93.5

Table 3. The ablation study results of Camera-LiDAR Adaptation and Spatial Characteristics Adaptation.

4.4 Quantitative Results

In Tab. 1, we compare to several state-of-the-art methods that utilize different input modalities on FlyingThings3D. The results show that our method demonstrates comparable performance both in 2D metrics and 3D metrics. Although the 2D Metrics is a little behind the latest DELFlow, the 3D metrics $\text{ACC}_{.05}$ and $\text{ACC}_{.10}$ achieve an improvement of approximately 1%. Compared to the baseline method, our approach shows an improvement of around 2%. The quantitative results on the KITTI Scene Flow dataset are shown in Tab. 2, which show that our method outperforms prior-arts both in D2 (2.81 vs. 2.84), Fl (2.98 vs. 3.07), and SF (4.23 vs. 4.34) metrics.

4.5 Qualitative Results

To validate the effectiveness of our method in 2D optical flow and 3D scene flow prediction, we present representative visualization results on the KITTI Scene Flow dataset, as shown in Fig. 3. Our method demonstrates the lowest error rates, highlighting its superior accuracy and robustness.

4.6 Model Size and Memory

Our PAFlow is small in size. It has 8.5 M parameters for the 3D scene flow predictions. Compared to previous methods, our approach maintains a competitive parameter count and inference speed. Specifically, compared to the latest methods (DELFlow and CamLiFlow), our model is approximately $2\times$ more lightweight in terms of parameters and achieves 2-5$\times$ faster inference speeds, respectively.

4.7 Ablation Study

We conduct a series of ablation studies to verify the effectiveness of each component of our design in the validation set of the FlyingThings3D dataset.

The effectiveness of components. presents the results of different configurations of our proposed design. We observe that both SCA (Spatial Characteristics Adaptation) and CLA (Camera-LiDAR Adaptation) make significant contributions to

Configuration		3D Metrics	
Projection	SNE	EPE_{3D} $(10^2)\downarrow$	$ACC_{.05}$ (%) $\uparrow$
-	-	6.1	85.6
Depth	-	6.0	85.6
	✓	5.9	85.9
Altitude	-	5.9	85.8
	✓	6.1	85.4
Depth + Altitude	✓	5.8	86.1

Table 4. The different configurations comparisons in Spatial Characteristics Adaptation.

improving 3D scene flow estimation. Notably, their synergistic effect plays a key role in enhancing overall 3D motion prediction. Experimental results further demonstrate that with the integration of SCA, the spatial features of LiDAR point clouds and camera frames become more closely aligned, leading to more effective feature fusion and alignment.

3D spatial characteristics on the image plane. We selected two different projection methods: depth projection and height difference projection. We find that both projection methods could be effectively integrated with RGB images. Furthermore, to better capture the spatial characteristics of pixels, we follow the SNE-RoadSeg (Fan et al., 2020) for surface normal estimation. Experiments showed that surface normal estimation has a positive impact on depth maps but a negative impact on elevation maps. Finally, we adopt a strategy that combines depth maps with surface normals and elevation maps to construct our Spatial Characteristics Adaptation.

5. Conclusion

In this paper, we introduce PAFlow, a novel framework for joint optical flow and scene flow estimation. It consists of camera-LiDAR adaptation and spatial characteristics adaptation modules, which work together to maximize the inter-modality complementarity. Experiments show that PAFlow outperforms the previous methods.

Limitations: Our method relies on strict correspondence and calibration parameters between the camera and LiDAR. In the future, we will explore the mutual optimization of optical flow and scene flow without using calibration parameters to support a wider range of remote sensing tasks.

Acknowledgement

The authors would like to thank the National Natural Science Foundation of China under Project 42371343, and the Basic and Applied Basic Research Foundation of Guangdong Province with grant No.2024A1515010986.

References

Behl, A., Hosseini Jafari, O., Karthik Mustikovela, S., Abu Alhaija, H., Rother, C., Geiger, A., 2017. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? *Proceedings of the IEEE International Conference on Computer Vision*, 2574–2583.

Chen, S., Han, T., Zhang, C., Su, J., Wang, R., Chen, Y., Wang, Z., Cai, G., 2025. HSPFormer: Hierarchical Spatial Perception Transformer for Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*.

Fan, R., Wang, H., Cai, P., Liu, M., 2020. Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. *European Conference on Computer Vision*, Springer, 340–356.

Gu, X., Wang, Y., Wu, C., Lee, Y. J., Wang, P., 2019. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3254–3263.

Han, T., Chen, S., Li, C., Wang, Z., Su, J., Huang, M., Cai, G., 2024a. Epurate-Net: Efficient Progressive Uncertainty Refinement Analysis for Traffic Environment Urban Road Detection. *IEEE Transactions on Intelligent Transportation Systems*, 25(7), 6617-6632.

Han, T., Chen, Y., Ma, J., Liu, X., Zhang, W., Zhang, X., Wang, H., 2024b. Point cloud semantic segmentation with adaptive spatial structure graph transformer. *International Journal of Applied Earth Observation and Geoinformation*, 133, 104105.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T., 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2462–2470.

Jiang, H., Sun, D., Jampani, V., Lv, Z., Learned-Miller, E., Kautz, J., 2019. Sense: A shared encoder network for scene-flow estimation. *Proceedings of the IEEE/CVF international conference on computer vision*, 3195–3204.

Li, C., Ma, H., Liao, Q., 2021a. Two-stage adaptive object scene flow using hybrid cnn-crf model. *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 3876–3883.

Li, R., Lin, G., He, T., Liu, F., Shen, C., 2021b. Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 364–373.

Ling, H., Sun, Q., Ren, Z., Liu, Y., Wang, H., Wang, Z., 2022. Scale-flow: Estimating 3d motion from video. *Proceedings of the 30th ACM International Conference on Multimedia*, 6530–6538.

Liu, H., Lu, T., Xu, Y., Liu, J., Li, W., Chen, L., 2022. Camliflow: bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5791–5801.

Liu, H., Lu, T., Xu, Y., Liu, J., Wang, L., 2024. Learning Optical Flow and Scene Flow With Bidirectional Camera-LiDAR Fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4), 2378-2395.

Liu, X., Qi, C. R., Guibas, L. J., 2019. Flownet3d: Learning scene flow in 3d point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 529–537.

Luo, Y., Han, T., Liu, Y., Su, J., Chen, Y., Li, J., Wu, Y., Cai, G., 2025. CSFNet: Cross-modal Semantic Focus Network for Sematic Segmentation of Large-Scale Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*.

Ma, W.-C., Wang, S., Hu, R., Xiong, Y., Urtasun, R., 2019. Deep rigid instance scene flow. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3614–3622.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T., 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

Mehl, L., Jahedi, A., Schmalfuss, J., Bruhn, A., 2023. M-fuse: Multi-frame fusion for scene flow estimation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020–2029.

Menze, M., Heipke, C., Geiger, A., 2015. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2, 427–434.

Menze, M., Heipke, C., Geiger, A., 2018. Object Scene Flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140, 60-76.

Ouyang, B., Raviv, D., 2021. Occlusion guided scene flow estimation on 3d point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2805–2814.

Peng, C., Wang, G., Lo, X. W., Wu, X., Xu, C., Tomizuka, M., Zhan, W., Wang, H., 2023. Delflow: Dense efficient learning of scene flow for large-scale point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16901–16910.

Puy, G., Boulch, A., Marlet, R., 2020. Flot: Scene flow on point clouds guided by optimal transport. *European conference on computer vision*, Springer, 527–544.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Ren, Z., Sun, D., Kautz, J., Sudderth, E., 2017. Cascaded scene flow prediction using semantic segmentation. *2017 International Conference on 3D Vision (3DV)*, IEEE, 225–233.

Rishav, R., Battrawy, R., Schuster, R., Wasenmüller, O., Stricker, D., 2020. Deeplidarflow: A deep learning architecture for scene flow estimation using monocular camera and sparse lidar. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 10460–10467.

Sun, D., Yang, X., Liu, M.-Y., Kautz, J., 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8934–8943.

Teed, Z., Deng, J., 2020. Raft: Recurrent all-pairs field transforms for optical flow. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 402–419.

Teed, Z., Deng, J., 2021. Raft-3d: Scene flow using rigid-motion embeddings. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8375–8384.

Vedula, S., Rander, P., Collins, R., Kanade, T., 2005. Three-dimensional scene flow. *IEEE transactions on pattern analysis and machine intelligence*, 27(3), 475–480.

Vogel, C., Schindler, K., Roth, S., 2015. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115, 1–28.

Wang, G., Wu, X., Liu, Z., Wang, H., 2021. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30, 5168–5181.

Wu, W., Wang, Z., Li, Z., Liu, W., Fuxin, L., 2019. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*.

Yang, G., Ramanan, D., 2020. Upgrading optical flow to 3d scene flow through optical expansion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1334–1343.

Yang, G., Ramanan, D., 2021. Learning to segment rigid motions from two frames. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1266–1275.

Zhai, M., Xiang, X., Lv, N., Kong, X., 2021. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114, 107861.