

FlowMamba: Learning Point Cloud Scene Flow with Global Motion Propagation

Min Lin¹, Gangwei Xu², Yun Wang², Xianqi Wang¹, Xin Yang^{2,3*}

¹School of Artificial Intelligence and Automation, Huazhong University of Science & Technology

²School of EIC, Huazhong University of Science & Technology

³Hubei Key Laboratory of Smart Internet Technology, Huazhong University of Science & Technology

{minlin, gwxu, wangyun, xianqiw, xinyang2014}@hust.edu.cn

Abstract

Scene flow methods based on deep learning have achieved impressive performance. However, current top-performing methods still struggle with ill-posed regions, such as extensive flat regions or occlusions, due to insufficient local evidence. In this paper, we propose a novel global-aware scene flow estimation network with global motion propagation, named FlowMamba. The core idea of FlowMamba is a novel Iterative Unit based on the State Space Model (ISU), which first propagates global motion patterns and then adaptively integrates the global motion information with previously hidden states. As the irregular nature of point clouds limits the performance of ISU in global motion propagation, we propose a feature-induced ordering strategy (FIO). The FIO leverages semantic-related and motion-related features to order points into a sequence characterized by spatial continuity. Extensive experiments demonstrate the effectiveness of FlowMamba, with 21.9% and 20.5% EPE3D reduction from the best published results on FlyingThings3D and KITTI datasets. Specifically, our FlowMamba is the first method to achieve millimeter-level prediction accuracy in FlyingThings3D and KITTI. Furthermore, the proposed ISU can be seamlessly embedded into existing iterative networks as a plug-and-play module, improving their estimation accuracy significantly.

Introduction

Scene flow estimation is the task of calculating three-dimensional motion fields from consecutive frames with various downstream applications, e.g., autonomous driving (Teng et al. 2023; Xu et al. 2022, 2023b), robotic manipulation (Seita et al. 2023), augmented reality (Wang et al. 2023b), etc. Nowadays, scene flow estimation still faces many challenges such as in flat regions, slender structures, or occlusions. These areas can be broadly defined as ill-posed areas: *regions with insufficient or even missing geometric features*. Such regions pose apparent challenges in predicting scene flow, as they introduce considerable local ambiguity and unreliable point correspondences between frames.

Earlier approaches address the challenge areas by employing neighboring points to recover motion. This can be achieved by using CNNs to learn the relationships between neighboring points (Wu et al. 2020; Wang et al. 2022) or by

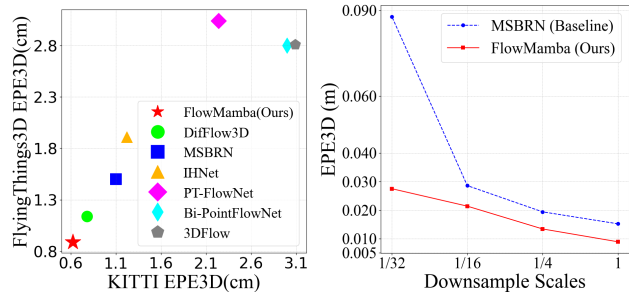


Figure 1: **Left:** Comparison with state-of-the-art scene flow methods (Liu et al. 2024a; Cheng and Ko 2023; Wang et al. 2023a; Fu et al. 2023; Cheng and Ko 2022; Wang et al. 2022) on FlyingThings3D and KITTI. Notably, we achieved millimeter-level precision on both datasets for the first time. **Right:** Comparison with the accuracy of each layer output on FlyingThings3D. Our FlowMamba can achieve superior results from the coarsest level. In practical applications, adjusting the number of levels and iterations allows for a trade-off between efficiency and accuracy.

constraining the scene flow to exhibit rigid motion within a local region (Li et al. 2021; Wang et al. 2023a). However, both methods are limited by the local information available within a small operational window, focusing primarily on local evidence. While local evidence becomes inadequate for recovering hidden motion, it severely reduces the performance of current state-of-the-art methods. Recent studies, such as the work by Lu et al. (Lu and Cheng 2023), have investigated the use of non-local approaches to model long-range dependencies between local descriptors, aiming to address the issue of insufficient local evidence. While these approaches help to some extent, they still tend to fail because a severe lack of geometric structure significantly diminishes the representational power of local descriptors.

A potential solution to tackle local ambiguity is to utilize global interpretations, such as learning global relationships through Transformers (Zhang et al. 2024b). However, the quadratic complexity of the attention mechanism imposes significant computational demands and hinders inference speed. Motivated by recent studies (Liu et al. 2024b; Zhang et al. 2024a; Liang et al. 2024) that apply structured state space

*Corresponding author

models (SSM) and Mamba to point cloud processing, which presents an effective global receptive field with linear complexity, we introduce a novel scene flow estimation method named FlowMamba. FlowMamba benefits from a novel iterative SSM-based update module (ISU) and a feature-induced ordering (FIO) strategy to efficiently capture long-range motion and model complex patterns, effectively addressing the issues in local ambiguous regions, as illustrated in Figure 4.

Specifically, we claim that the ISU module plays the key role of propagating global information, which motivates two critical designs. Firstly, we advocate for bidirectional sequence modeling to ensure comprehensive information aggregation from all other points. Secondly, the ISU integrates global hidden information in an iterative update process, enhancing the model’s capability to capture complex motion patterns. However, the inherent irregularity of point clouds can restrict the propagation of global motion when directly applying the ISU module without ordering, as this irregularity may introduce erroneous spatial relationship during the sequence modeling process. To overcome this limitation, we propose the feature-induced ordering (FIO) strategy. The core concept is to enable the network to construct causal dependencies among points at a higher level, implicitly preserving spatial consistency. Specifically, we harness semantic-related and motion-related features—contextual features, motion information, and the updated hidden information—as crucial cues to generate a score for each point and determine the sequence order.

We demonstrate that long-range connections facilitated by the ISU in FlowMamba significantly enhance scene flow estimation, especially for addressing the motion of ill-posed regions where local information is inadequate. On the FlyingThings3D dataset (Mayer et al. 2016), our FlowMamba reaches the state-of-the-art EPE3D with 21.9% and 20.9% lower errors under the non-occluded and occluded scenarios, respectively. And on the real-world KITTI dataset (Geiger et al. 2013), FlowMamba improves the generality by decreasing the errors 20.5% and 9.6% on non-occluded and occluded scenarios, respectively.

Overall, the contributions of our paper are as follows:

1. We propose a novel scene flow estimation architecture, named FlowMamba, which incorporates global motion propagation to significantly enhance the robustness of motion estimation.
2. We introduce a novel iterative SSM-based update module (ISU), enabling the efficient integration of global motion information in the point cloud.
3. We introduce a feature-induced ordering strategy (FIO) to alleviate the impact of point cloud irregularity on global motion propagation.
4. Our method outperforms existing published methods on FlyingThings3D and KITTI datasets. Especially, our FlowMamba achieves millimeter-level precision on both datasets for the first time. We also verify the universality of our ISU on several scene flow methods.

Related Work

Point Cloud-Based Scene Flow Estimation

Currently, the emergence of deep learning on point cloud (Qi et al. 2017a,b; Wu, Qi, and Fuxin 2019; Wu, Fuxin, and Shan 2023) has sparked interest in obtaining scene flow directly from point cloud data. FlowNet3D (Liu, Qi, and Guibas 2019) was the pioneering method to address scene flow estimation in point clouds using learning-based techniques. PointPWC-Net (Wu et al. 2020) employs a coarse-to-fine approach for estimating scene flow in point clouds. Bi-PointFlowNet (Cheng and Ko 2022) employs bidirectional flow (Xu et al. 2024b) embedding layers to improve scene flow estimation. RMS-FlowNet (Battrawy et al. 2022) introduces the Patch to Dilated Patch flow embedding design, allowing for more robust scene flow prediction in conjunction with Random-Sampling. 3DFlow (Wang et al. 2022) introduces an all-to-all flow embedding layer with backward reliability verification based on the coarse-to-fine construct. IHNet (Wang et al. 2023a) proposes a resampling scheme to alleviate the problem of poor correspondence. Nevertheless, these methods face limitations, such as error accumulation in early steps and a tendency to miss fast-moving objects.

In recent years, many iterative methods (Kittenplon, Eldar, and Raviv 2021; Wei et al. 2021; Gu et al. 2022; Fu et al. 2023; Xu et al. 2023a, 2024a; Wang et al. 2024) have gradually become the mainstream of research. PV-RAFT (Wei et al. 2021) proposes point-voxel correlation fields to handle large and small displacements. RCP (Gu et al. 2022) employs a two-stage recurrent network, where 3D flows are optimized at a point-wise level in the first stage and then globally regularized in a recurrent network in the second stage. PT-FlowNet (Fu et al. 2023) applies transformer (Zhao et al. 2021; Zhang et al. 2020) in all functional stages of the task and achieves outstanding results. MSBRN (Cheng and Ko 2023) proposes an effective and efficient architecture by iterative optimizing coarse-to-fine scene flow. DiffFlow3D (Liu et al. 2024a) introduces an uncertainty-aware scene flow estimation network with the diffusion probabilistic model to enhance resilience to challenging cases. *However, how to efficiently enhance matching capabilities in complex regions through global motion propagation remains a challenging issue.*

State Space Model in Point Cloud

Inspired by continuous state space models in control systems, recently, There has been a significantly increasing focus on the state space models (SSMs) (Gu, Goel, and Ré 2021; Gu and Dao 2023), which have been proven to have competitive long-range dependency modeling ability. In particular, S4 (Gu, Goel, and Ré 2021) introduces a normalization technique for the parameters to achieve stable diagonalization, significantly reducing computational overhead and memory consumption. Mamba (Gu and Dao 2023) presents a selection mechanism and hardware-aware algorithms, resulting in superior outcomes compared to transformers. Currently, many methods apply mamba to 3D point cloud classification, segmentation tasks, static point cloud generation, and completion (Liu et al. 2024b; Zhang et al. 2024a; Liang et al.

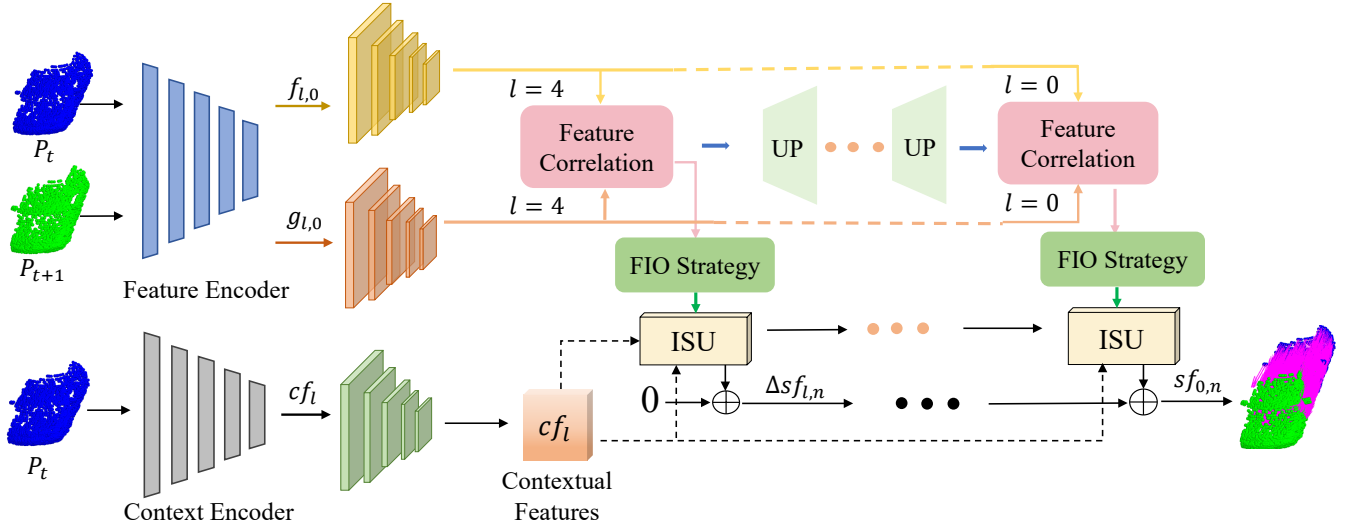


Figure 2: **Overview of our proposed FlowMamba.** Feature encoders abstract the point clouds to obtain the multi-scale point feature and context feature. The correlation features can be obtained by using local cost volumes retrieved from the feature pyramid. The iterative SSM-based update module (ISU) is designed to update the hidden information and scene flow by capturing long-range dependencies and comprehensive motion patterns with global motion propagation. The feature-induced ordering (FIO) strategy is designed to construct reasonable causal dependencies in point cloud.

2024; Li, Yang, and Fei 2024; Mo 2024). PointMamba (Liang et al. 2024) is the first to apply mamba to point cloud learning. 3DMambaComplete (Li, Yang, and Fei 2024) proposes to use mamba’s global modeling capabilities to enhance the understanding of point cloud reconstruction. DiM-3D (Mo 2024) uses mamba to improve the scalability of models and the performance of outputting high-resolution voxels. *However, it remains unclear whether the state space model can model dynamic point clouds, i.e., scene flow.*

Methods

We based our network design on the successful MSBRN (Cheng and Ko 2023) architecture. The overview of our proposed method is shown in Figure 2. For the given two consecutive point clouds $P_t = \{p_i \in \mathbb{R}^3\}_{i=1}^{N_1}$, $P_{t+1} = \{q_j \in \mathbb{R}^3\}_{j=1}^{N_2}$, we estimate the 3D motion vector for each point as $SF = \{sf \in \mathbb{R}^3\}_{i=1}^{N_1}$.

Preliminaries

The SSM-based models, such as structured state space sequence models (S4) (Gu, Goel, and Ré 2021) and Mamba (Gu and Dao 2023), are motivated by a continuous-time learnable framework for mapping between continuous-time scalar inputs $x(t) \in \mathbb{R}$ and outputs $y(t) \in \mathbb{R}$ through an implicit latent state $h(t) \in \mathbb{R}^N$.

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t) \end{aligned} \quad (1)$$

Where the \mathbf{A} is the evolution parameter and \mathbf{B} , \mathbf{C} is the projection parameters.

The S4 (Gu, Goel, and Ré 2021) and Mamba (Gu and Dao 2023) are the discrete versions of the continuous system.

Zero-order hold (ZOH) discretizes the continuous-time SSM to a discrete-time SSM by using a timescale parameter Δ .

$$\begin{aligned} \mathbf{A} &= \exp(\Delta \mathbf{A}), \\ \mathbf{B} &= (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B} \end{aligned} \quad (2)$$

In this case the formula (1) can be rewritten as:

$$\mathbf{h}_t = \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}\mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t \quad (3)$$

Finally, the models compute output through a global convolution.

$$\mathbf{K} = (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{N-1}\bar{\mathbf{B}}), \quad y = x * \mathbf{K} \quad (4)$$

Feature Extraction

Following the design introduced in (Qi et al. 2017b; Wu, Qi, and Fuxin 2019; Cheng and Ko 2023), we utilize multi-scale point feature extractions as the backbone. Feature encoder and context encoder produces L -level pyramid of point features $f_{l,0} \in \mathbb{R}^{N_1 \times C}$, $g_{l,0} \in \mathbb{R}^{N_2 \times C}$, and $cf \in \mathbb{R}^{N_1 \times C}$, starting with the input point clouds at the top. At each level l , the farthest point sampling (FPS) subsamples dense points and features to create a sparse set. For each sparse point, k -nearest neighbors (KNN) form a local region. A Pointconv layer (Wu, Qi, and Fuxin 2019) then dynamically weights and aggregates features from these neighbors to obtain a local feature for each sparse point.

Iterative SSM-Based Update Module

The local receptive field in the classical GRU structure hinders the propagation of matching information and the modeling of long-range dependency motions. To tackle this limita-

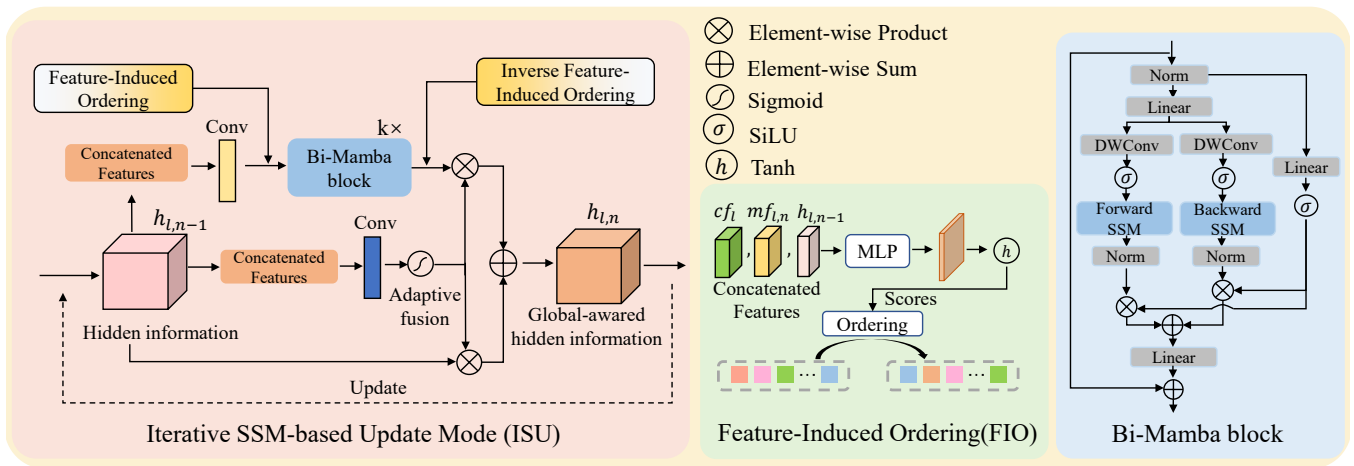


Figure 3: **The architecture of proposed module.** Left: Iterative SSM-based Update (ISU) Module. Middle: Feature-Induced Ordering(FIO) strategy. Right: Bi-directional Mamba block (Bi-Mamba).

tion, we propose an iterative update module based on SSM with global motion propagation (ISU).

As illustrated in Figure 3, for a certain layer l , our iterative module takes the coordinates $p_i \in \mathbb{R}^3$ and features $f_{i,0} \in \mathbb{R}^{N_1 \times C_1}$ of the source point cloud, the coordinates $q_j \in \mathbb{R}^3$ and features $g_{j,0} \in \mathbb{R}^{N_2 \times C_1}$ of the target point cloud, context information $cf_l \in \mathbb{R}^{N_1 \times C_1}$, motion features $mf_{l,0} \in \mathbb{R}^{N_1 \times C_2}$, flow $sf_{l,0} \in \mathbb{R}^{N_1 \times 3}$, and the hidden information from last layer $h_{l+1,0}$ as input. After N iterations, the updated point cloud features $f_{l,n}$, $g_{l,n}$, flow $sf_{l,n}$, and hidden state $h_{l,n}$ are used as inputs for subsequent scales.

Our ISU module mainly consists of two parts: global hidden information optimization and adaptive fusion update.

Global hidden information optimization. We stack several layers of bidirectional Mamba (Bi-Mamba) blocks to achieve global information aggregation. Specifically, we first sort the point cloud in each iteration, as detailed in the following section. The sorted hidden information is then fed into bidirectional Mamba blocks for global optimization. Each bidirectional Mamba block utilizes layer normalization (LN), bidirectional Selective SSM, depth-wise convolution (DW) (Chollet 2017), and residual connections. The bidirectional Mamba layer is shown on the right of Figure 3, and the output can be present as follows:

$$\begin{aligned}
 h'_{l,n-1} &= \text{LN}(\text{Conv}_{1d}([cf_l, mf_{l,n}, h_{l,n-1}])), \\
 h'_{l,n} &= \sigma(\text{DW}(\text{Linear}(h'_{l,n-1}))) \\
 h''_{l,n} &= \sigma(\text{Linear}(h'_{l,n-1})) \\
 h_{l,n} &= \text{Linear}(\text{Bi-SSM}(h'_{l,n}) \odot h''_{l,n}) + h_{l,n-1}
 \end{aligned}$$

Where $h_{l,n} \in \mathbb{R}^{N_1 \times C}$ is the output of the l -th block in n -th iterations, and σ indicates *SiLU* activation (Hendrycks and Gimpel 2016). The bidirectional SSM (Zhu et al. 2024; Liu et al. 2024b) is the key to the Bi-Mamba block.

Adaptive fusion update. The globally optimized hidden information is then fused with the historical information from

the previous iteration. Specifically, we use a gated mechanism to adaptively fuse them, resulting in the final global-aware hidden information. The adaptive fusion method is defined as follows:

$$\begin{aligned}
 w_{l,n} &= \text{sigmoid}(\text{Conv}_{1d}([cf_l, mf_{l,n}, h_{l,n-1}])) \\
 \hat{h}_{l,n} &= (1 - w_{l,n}) \odot h_{l,n-1} + w_{l,n} \odot h_{l,n}
 \end{aligned} \quad (5)$$

Where $w_{l,n}$ denotes the attentive weight, \odot denotes the Hadamard product, $h_{l,n-1}$ denotes the hidden information from the previous iteration.

Feature-Induced Ordering Strategy

The ordering strategy is critical for the effective utilization of SSM. To mitigate the impact of point cloud irregularity on the propagation of global information, We propose a novel feature-induced ordering strategy and sort all points in each iteration. In detail, we generate a score for each point by using three features related to motion estimation: contextual information, correlation feature, and the updated hidden information. We then sort these scores to determine the neighborhoods of the points. These features represent the spatial and motion states of the point cloud in the current iteration, revealing the intrinsic relationships between the points. Thus, the FIO strategy ensures the construction of causal dependencies and spatial continuity in the point cloud, thereby facilitating the propagation of features.

$$\text{Score} = \text{tanh}(\text{MLP}[cf_l, mf_{l,n}, h_{l,n-1}]) \quad (6)$$

Where cf_l denotes the context feature, $mf_{l,n}$ denotes the motion features in n -th iterations, and $h_{l,n-1}$ denotes the updated hidden information from the $n - 1$ iterations.

The Architecture of FlowMamba

We adopt a coarse-to-fine structure to estimate the residual flow at different scales. For each layer, the sparse scene flow $sf_{l+1,n}$, hidden state $h_{l+1,n}$, and cf_{l+1} in the coarse layer are upsampled to dense. Following the approach in (Cheng and

Method	FT3D _s						KITTI _s					
	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓	EPE2D↓	Acc2D↑	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓	EPE2D↓	Acc2D↑
FlowNet3D	0.1136	0.4125	0.7706	0.6016	5.9740	0.5692	0.1767	0.3738	0.6677	0.5271	7.2141	0.5093
HPLFlowNet	0.0804	0.6144	0.8555	0.4287	4.6723	0.6764	0.1169	0.4783	0.7776	0.4103	4.8055	0.5938
PointPWC-Net	0.0588	0.7379	0.9276	0.3424	3.2390	0.7994	0.0694	0.7281	0.8884	0.2648	3.0062	0.7673
HALFlow	0.0492	0.7850	0.9468	0.3083	2.7555	0.8111	0.0622	0.7649	0.9026	0.2492	2.5140	0.8128
FLOT	0.0520	0.7320	0.9270	0.3570	—	—	0.0560	0.7550	0.9080	0.2420	—	—
PV-RAFT	0.0461	0.8169	0.9574	0.2924	—	—	0.0560	0.8226	0.9372	0.2163	—	—
RCP	0.0403	0.8567	0.9635	0.1976	—	—	0.0481	0.8491	0.9448	0.1228	—	—
3DFlow	0.0281	0.9290	0.9817	0.1458	1.5229	0.9279	0.0309	0.9047	0.9580	0.1612	1.1285	0.9451
Bi-PointFlowNet	0.0280	0.9180	0.9780	0.1430	1.5820	0.9290	0.0300	0.9200	0.9600	0.1410	1.0560	0.9490
PT-FlowNet	0.0304	0.9142	0.9814	0.1735	1.6150	0.9312	0.0224	0.9551	0.9838	0.1186	0.9893	0.9667
IHNet	0.0191	0.9601	0.9865	0.0715	1.0918	0.9563	0.0122	0.9779	0.9892	0.0913	0.4993	0.9862
MSBRN	0.0150	0.9730	0.9920	0.0560	0.8330	0.9700	0.0110	0.9710	0.9890	0.0850	0.4430	0.9850
DiffFlow3D	0.0114	0.9836	0.9949	0.0350	0.6220	0.9824	0.0078	0.9817	0.9924	0.0795	0.2987	0.9932
Ours	0.0089	0.9861	0.9955	0.0243	0.4946	0.9848	0.0062	0.9876	0.9938	0.0741	0.2604	0.9919

Table 1: Comparison results on FlyingThings3D (Mayer et al. 2016) and KITTI (Geiger et al. 2013) datasets without occlusion (Gu et al. 2019). Our method has 21.9% and 20.5% EPE3D reduction respectively compared with previous works. It is worth noting that, for the first time, we achieved millimeter-level precision on both datasets (0.0089 on FlyingThings3D and 0.0062 on KITTI). The best results are in bold.

Ko 2023; Liu et al. 2024a), we then employ a warp operation to iteratively align the source point cloud with the target point cloud using the updated flow field. Subsequently, we implement a bidirectional feature enhancement strategy based on the coordinates and features of the two point clouds. Using SetConv (which includes shared multilayer perceptrons and max-pooling layers), we effectively aggregate features from the other frame to enhance the feature representation of the current point cloud. Finally, the enhanced features are cached to provide a foundation for subsequent iteration processes or scaling.

Loss Function

Flow Supervision. We follow most of the supervised scene flow approaches to design the loss function (Wang et al. 2022; Cheng and Ko 2023; Gu et al. 2019, 2022; Fu et al. 2023; Wang et al. 2021). Specifically, we use the L2 norm between the ground truth flow and the estimated flow in each iteration.

$$L = \sum_{l=1}^L \alpha^{(l)} \sum_{n=1}^N \left\| s f_{l,n}^{gt} - s f_{l,n} \right\|_2 \quad (7)$$

Where $s f_{l,n}$ is the scene flow of the n^{th} iterative estimation. N is the total number of iterations, and $\alpha^{(l)}$ is the weight of the l^{th} layer. We adopt the hyper-parameter setting in (Cheng and Ko 2023; Liu et al. 2023) with $\alpha^{(0)} = 0.16$, $\alpha^{(1)} = 0.08$, $\alpha^{(2)} = 0.04$, $\alpha^{(3)} = 0.02$.

Experiments

Datasets and Implementation Details

Datasets. For a fair comparison, we trained FlowMamba following the previous methods (Wei et al. 2021; Fu et al. 2023; Cheng and Ko 2023) on FlyingThings3D (Mayer et al. 2016) and tested on both FlyingThings3D and KITTI (Geiger et al. 2013). FlyingThings3D is a large synthetic dataset, including 19,640 pairs of labeled training samples and 3,824 samples in the test set. We directly evaluated the model on

Dataset	Method	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓
FT3D _o	FlowNet3D	0.169	0.254	0.579	0.789
	FLOT	0.156	0.343	0.643	0.700
	FESTA	0.111	0.431	0.744	—
	3DFlow	0.063	0.791	0.909	0.279
	Bi-PointFlowNet	0.073	0.791	0.896	0.274
	MSBRN	0.053	0.836	0.926	0.231
	DiffFlow3D	0.043	0.891	0.944	0.133
	Ours	0.034	0.9165	0.9531	0.0904
KITTI _o	FlowNet3D	0.173	0.276	0.609	0.649
	FLOT	0.110	0.419	0.721	0.486
	FESTA	0.097	0.449	0.833	—
	3DFlow	0.073	0.819	0.890	0.261
	Bi-PointFlowNet	0.065	0.769	0.906	0.264
	MSBRN	0.044	0.873	0.950	0.208
	DiffFlow3D	0.031	0.955	0.966	0.108
	Ours	0.028	0.959	0.971	0.096

Table 2: Comparison results on FlyingThings3D and KITTI datasets with occlusion (Liu, Qi, and Guibas 2019). Our method outperforms previous works by 20.9% and 9.6% in terms of EPE3D. The best results are in bold.

KITTI (Geiger et al. 2013) without any fine-tuning to validate the generalization ability on the real-world KITTI dataset, which contains 200 pairs of test data.

Implementation Details. All experiments were conducted using PyTorch (Paszke et al. 2019). For a fair comparison, we randomly sampled 8192 points as the network input. Consistent with prior work (Cheng and Ko 2023), the point numbers at each layer are $N^1 = 2048$, $N^2 = 512$, $N^3 = 256$, and $N^4 = 64$. During training, we performed 4 update iterations per layer, which were also maintained during evaluation. We used the AdamW (Kingma and Ba 2014; Loshchilov and Hutter 2017) optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate was adjusted using the CosineLR strategy, starting at $1e-3$. The model was trained for a total of 300 epochs, and the same evaluation metrics as in recent studies (Wu et al. 2020; Cheng and Ko 2023; Liu et al. 2024a) were employed.

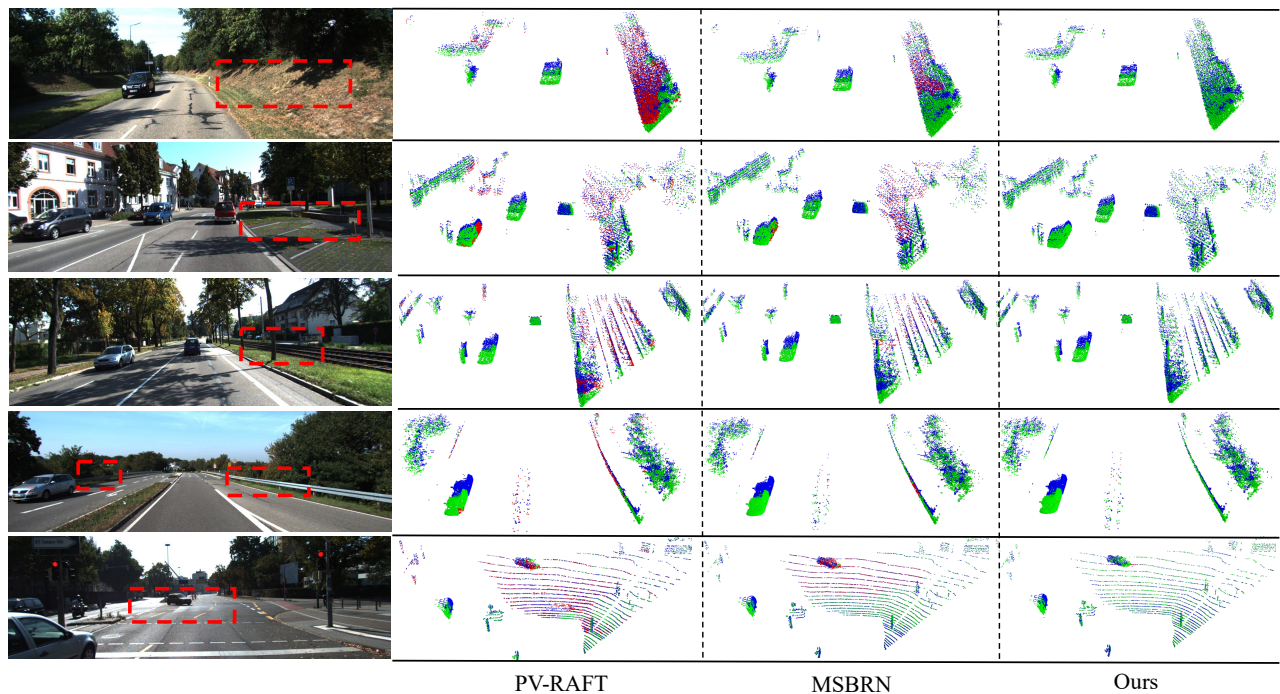


Figure 4: **Qualitative results on the test set of KITTI.** It shows that providing global motion propagation improves performance in areas with ambiguous geometric characteristics, such as embankments, roadside grassy areas, and some slender structures (curbs or tracks). Blue, green and red points respectively indicate the first frame P_t , accurately estimated in P_t and inaccurately estimated in P_t (measured by Acc3DS).

Comparisons with State-of-the-Art

We validate the superiority of FlowMamba on datasets with two distinct pre-processing conditions, one including occlusion (Liu, Qi, and Guibas 2019) and the other excluding it (Gu et al. 2019), respectively.

Comparison on point clouds without occlusion. We compare our FlowMamba with a series of State-of-the-art (SOTA) methods on the FlyingThings3D and KITTI without occlusion. Table 1 showcases the superiority of our FlowMamba approach, surpassing all previous methods across both 3D and 2D evaluation metrics. In comparison to the state-of-the-art DifFlow3D method (Liu et al. 2024a), our model exhibits a notable reduction in EPE3D by 21.9% on the FlyingThings3D and 20.5% on the KITTI, respectively. FlowMamba also has excellent generalization capabilities that achieve a millimeter-level EPE3D of 0.0062m on the KITTI dataset, just training on the synthetic FlyingThings3D dataset. Notably, we have achieved millimeter-level precision on both datasets for the first time.

Comparison on point clouds with occlusion. We also evaluate FlowMamba on the datasets with occlusion and compare it with the SOTA results. Experiments in Table 2 demonstrate that FlowMamba outperforms all previous methods on the FlyingThings3D and KITTI. In comparison to the state-of-the-art DifFlow3D method (Liu et al. 2024a), our model achieves a remarkable reduction in End-Point Error (EPE3D) by 20.9% and 9.6% on the FlyingThings3D and KITTI datasets, respectively.

Universality of Proposed Modules

Our proposed ISU can be used as a plug-and-play module to improve the accuracy of several iterative-based methods (Wei et al. 2021; Fu et al. 2023; Cheng and Ko 2023). As shown in Table 3, we replace their GRU with our ISU, resulting in significant improvements in baseline accuracy on both the FlyingThings3D and KITTI datasets. After incorporating ISU, the EPE3D of PV-RAFT decreased by 28.2% on FlyingThings3D and 8.2% on KITTI, respectively. The EPE3D of PT-FlowNet was reduced by 20.4% and 6.7%, respectively. Although MSBRN has achieved impressive results, our ISU has demonstrated the potential to further boost its performance significantly (28.6% and 5.4% respectively).

Ablation Study

In this section, we evaluate our model in different settings to verify our proposed modules in several aspects. All results are obtained using the same number of iterations as reporting in (Cheng and Ko 2023). Figure 4 illustrates the superiority of our proposed method. It demonstrates that our method improves performance in areas with ambiguous geometric characteristics, such as flat regions and slender structures (curbs or tracks).

Effectiveness of proposed modules. We compare different global information propagation settings to demonstrate the effectiveness of our design, as shown in Table 4. We first try using a standard unidirectional Mamba instead of the Conv GRU and find it brings a decline in performance, as

Method	FT3D _s				KITTI _s			
	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓
PV-RAFT	0.0461	0.8169	0.9574	0.2924	0.0560	0.8226	0.9372	0.2163
PV-RAFT(w/ISU)	0.0331 (↓28.2%)	0.9198	0.9783	0.1802	0.0514 (↓8.2%)	0.8952	0.9576	0.1672
PT-FlowNet	0.0304	0.9142	0.9814	0.1735	0.0224	0.9551	0.9838	0.1186
PT-FlowNet(w/ISU)	0.0242 (↓20.4%)	0.9494	0.9860	0.1166	0.0209 (↓6.7%)	0.9553	0.9846	0.1137
MSBRN	0.0150	0.9730	0.9920	0.0243	0.0110	0.9710	0.9890	0.0850
MSBRN(w/ISU)	0.0107 (↓28.6%)	0.9741	0.9927	0.0892	0.0104 (↓5.4%)	0.9718	0.9895	0.0829

Table 3: **The plug-and-play capability of our methods.** Our Iterative SSM-based Update Module (ISU) can effectively improve the accuracy introduced into recent methods on both FlyingThings3D and KITTI datasets. The best results are in **bold**.

Method	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓
Conv GRU	0.0134	0.9763	0.9931	0.0466
Mamba	0.0141	0.9751	0.9929	0.0482
Bi-Mamba	0.0127	0.9799	0.9935	0.0353
ISU(Ours)	0.0101	0.9835	0.9946	0.0286
ISU + FIO (Ours)	0.0089	0.9861	0.9955	0.0243

Table 4: Ablation studies the effectiveness of proposed modules on FlyingThings3D.

Component	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓
w/o context	0.0096	0.9849	0.9951	0.0265
w/o correlation	0.0095	0.9851	0.9953	0.0264
w/o hidden state	0.0093	0.9853	0.9953	0.0259
Ours (full)	0.0089	0.9861	0.9955	0.0243

Table 5: Ablation experiment of FIO strategy on FlyingThings3D.

the limitations of unidirectional sequence modeling in effectively capturing global dependencies. Subsequently, we replaced it with bidirectional Mamba, utilizing the bidirectional mechanism to ensure comprehensive aggregation of information from other points. However, due to the impact of global noise propagation, the performance improvement remains marginal. We then replaced it with the ISU, which can adaptively perform global information fusion and updating, significantly improving estimation accuracy. Additionally, we incorporated the FIO strategy, which constructs coherent spatial dependencies, further enhancing the global information propagation capabilities of the ISU.

Feature-induced ordering. To verify the impact of ordering settings with different features on final performance, we compared the estimation results by removing various features. As shown in Table 5, the feature-induced ordering strategy enhances motion estimation accuracy by sorting the point cloud. Additionally, using all three features for estimation leads to more convincing results.

Number of iterations. Our FlowMamba can achieve better performance with a smaller number of iterations. As shown in Table 6, our network gets the same performance in just 2 iterations compared to MSBRN with 4 iterations. This indicates that our module can filter and reduce noise in hidden states through global motion propagation.

Dataset	Method	Number of Iterations			
		1	2	3	4
FT3D	MSBRN	0.0403	0.0241	0.0177	0.0152
	Ours	0.0176	0.0118	0.0097	0.0089
KITTI	MSBRN	0.0452	0.0239	0.0143	0.0110
	Ours	0.0204	0.0110	0.0070	0.0062

Table 6: Ablation study of the number of iterations.

Method	Parameters	Runtime
Baseline	6.1M	283ms
Ours	6.0M	304ms

Table 7: **Runtime comparison.** The runtime comparison between the baseline and our FlowMamba with 4 iterations setting. The baseline, based on MSBRN, includes an additional encoder for extracting contextual features.

Parameters and Runtime

We compare FlowMamba with baseline methods to show the efficiency of ISU. All evaluations were conducted on a single RTX 3090 GPU. As shown in Table 7, FlowMamba demonstrates highly competitive efficiency. Our baseline, based on MSBRN (Cheng and Ko 2023), incorporates an additional encoder to obtain contextual features. Compared to the baseline, FlowMamba has fewer parameters, with only a slight increase in computational time, while achieving significant performance improvements as reported in Table 1.

Conclusion

We proposed a novel scene flow estimation method, called FlowMamba. Its core component, the ISU module, is designed to efficiently propagate matching information globally and model long-range motion dependencies. To alleviate the impact of point cloud irregularity and enhance the global propagation capabilities of the ISU, we introduced an FIO strategy to order points into a sequence with spatial continuity at a high level. Extensive experiments demonstrate the superiority of our FlowMamba and its generalization ability on FlyingThings3D and KITTI. It shows the critical importance of global information propagation for point cloud motion estimation. The proposed method also showcases a strong universality as a plug-and-play module for various approaches.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62122029, 62472184), the Fundamental Research Funds for the Central Universities, and the National Natural Science Foundation of China (623B2036).

References

- Battrawy, R.; Schuster, R.; Mahani, M.-A. N.; and Stricker, D. 2022. Rms-flownet: Efficient and robust multi-scale scene flow estimation for large-scale point clouds. In *2022 International Conference on Robotics and Automation (ICRA)*, 883–889. IEEE.
- Cheng, W.; and Ko, J. H. 2022. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision*, 108–124. Springer.
- Cheng, W.; and Ko, J. H. 2023. Multi-scale bidirectional recurrent network with hybrid correlation for point cloud based scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10041–10050.
- Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.
- Fu, J.; Xiang, Z.; Qiao, C.; and Bai, T. 2023. PT-FlowNet: Scene Flow Estimation on Point Clouds With Point Transformer. *IEEE Robotics and Automation Letters*, 8(5): 2566–2573.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.
- Gu, X.; Tang, C.; Yuan, W.; Dai, Z.; Zhu, S.; and Tan, P. 2022. Rcp: Recurrent closest point for point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8216–8226.
- Gu, X.; Wang, Y.; Wu, C.; Lee, Y. J.; and Wang, P. 2019. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3254–3263.
- Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kittenplon, Y.; Eldar, Y. C.; and Raviv, D. 2021. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4114–4123.
- Li, R.; Lin, G.; He, T.; Liu, F.; and Shen, C. 2021. Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 364–373.
- Li, Y.; Yang, W.; and Fei, B. 2024. 3dmambacomplete: Exploring structured state space model for point cloud completion. *arXiv preprint arXiv:2404.07106*.
- Liang, D.; Zhou, X.; Wang, X.; Zhu, X.; Xu, W.; Zou, Z.; Ye, X.; and Bai, X. 2024. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*.
- Liu, J.; Wang, G.; Ye, W.; Jiang, C.; Han, J.; Liu, Z.; Zhang, G.; Du, D.; and Wang, H. 2023. DiffFlow3D: Toward Robust Uncertainty-Aware Scene Flow Estimation with Diffusion Model. *arXiv preprint arXiv:2311.17456*.
- Liu, J.; Wang, G.; Ye, W.; Jiang, C.; Han, J.; Liu, Z.; Zhang, G.; Du, D.; and Wang, H. 2024a. DiffFlow3D: Toward Robust Uncertainty-Aware Scene Flow Estimation with Iterative Diffusion-Based Refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15109–15119.
- Liu, J.; Yu, R.; Wang, Y.; Zheng, Y.; Deng, T.; Ye, W.; and Wang, H. 2024b. Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy. *arXiv preprint arXiv:2403.06467*.
- Liu, X.; Qi, C. R.; and Guibas, L. J. 2019. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 529–537.
- Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Lu, Z.; and Cheng, M. 2023. GMA3D: Local-global attention learning to estimate occluded motions of scene flow. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 16–27. Springer.
- Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.
- Mo, S. 2024. Efficient 3D Shape Generation via Diffusion Mamba with Bidirectional SSMs. *arXiv preprint arXiv:2406.05038*.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.
- Seita, D.; Wang, Y.; Shetty, S. J.; Li, E. Y.; Erickson, Z.; and Held, D. 2023. Toolflownet: Robotic manipulation with tools

- via predicting tool flow from point clouds. In *Conference on Robot Learning*, 1038–1049. PMLR.
- Teng, S.; Hu, X.; Deng, P.; Li, B.; Li, Y.; Ai, Y.; Yang, D.; Li, L.; Xuanyuan, Z.; Zhu, F.; et al. 2023. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*.
- Wang, G.; Hu, Y.; Liu, Z.; Zhou, Y.; Tomizuka, M.; Zhan, W.; and Wang, H. 2022. What matters for 3d scene flow network. In *European Conference on Computer Vision*, 38–55. Springer.
- Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30: 5168–5181.
- Wang, X.; Xu, G.; Jia, H.; and Yang, X. 2024. Selective-stereo: Adaptive frequency information selection for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19701–19710.
- Wang, Y.; Chi, C.; Lin, M.; and Yang, X. 2023a. Ihnet: Iterative hierarchical network guided by high-resolution estimated information for scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10073–10082.
- Wang, Z.; Nguyen, C.; Asente, P.; and Dorsey, J. 2023b. PointShopAR: Supporting Environmental Design Prototyping Using Point Cloud in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–15.
- Wei, Y.; Wang, Z.; Rao, Y.; Lu, J.; and Zhou, J. 2021. Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6954–6963.
- Wu, W.; Fuxin, L.; and Shan, Q. 2023. Pointconvformer: Revenge of the point-based convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21802–21813.
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 9621–9630.
- Wu, W.; Wang, Z. Y.; Li, Z.; Liu, W.; and Fuxin, L. 2020. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, 88–107. Springer.
- Xu, G.; Cheng, J.; Guo, P.; and Yang, X. 2022. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12981–12990.
- Xu, G.; Wang, X.; Ding, X.; and Yang, X. 2023a. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21919–21928.
- Xu, G.; Wang, X.; Zhang, Z.; Cheng, J.; Liao, C.; and Yang, X. 2024a. Igev++: Iterative multi-range geometry encoding volumes for stereo matching. *arXiv preprint arXiv:2409.00638*.
- Xu, G.; Wang, Y.; Cheng, J.; Tang, J.; and Yang, X. 2023b. Accurate and efficient stereo matching via attention concatenation volume. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Xu, G.; Wang, Y.; Gu, J.; Xue, T.; and Yang, X. 2024b. HDR-Flow: Real-Time HDR Video Reconstruction with Large Motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24851–24860.
- Zhang, D.; Zhang, H.; Tang, J.; Wang, M.; Hua, X.; and Sun, Q. 2020. Feature pyramid transformer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, 323–339. Springer.
- Zhang, T.; Li, X.; Yuan, H.; Ji, S.; and Yan, S. 2024a. Point could mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*.
- Zhang, Y.; Edstedt, J.; Wandt, B.; Forssén, P.-E.; Magnusson, M.; and Felsberg, M. 2024b. Gmsf: Global matching scene flow. *Advances in Neural Information Processing Systems*, 36.
- Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.
- Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.