3DSF-MixNet: Mixer-Based Symmetric Scene Flow Estimation From 3D Point Clouds

Shuaijun Wang, *Member, IEEE*, Rui Gao, Ruihua Han, *Graduate Student Member, IEEE*, and Qi Hao, *Member, IEEE*

***Abstract*—The scene flow estimation aims at accurately achieving the motion of 3D points, imposing challenges like mis-registration, object occlusions, and non-uniform upsampling. This paper introduces a scene flow estimation framework featuring a unified scene flow estimator, a symmetric cost volume approach, and a geometric/semantic feature based upsampling strategy. The novelty of this work is threefold: (1) developing a novel progressive framework which integrates the cost volume module and scene flow estimator, enhancing scene flow estimation; (2) developing a symmetric inter-frame correlation feature extraction method through cost volume estimation using MLP-Mixer operations; (3) developing an upsampling strategy based on both the semantic and geometric feature similarities between sparse and dense samples. Experimental results show that our method outperforms state-of-the-art baseline methods, especially in scenarios involving challenging conditions, the improvements of our method achieving at most 0.109 m/0.089 m/0.091 m in EPE3D, $54.23\%/53.67\%/74.1\%$ in AS, $32.75\%/21.87\%/40.25\%$ in AR, and $70.98\%/58.06\%/43.56\%$ in outliers, when tested on FlyingThings3D ($\mathbf{FT3D_S}$, $\mathbf{FT3D_H}$) and $\mathbf{KITTI_H}$ datasets, respectively.**

***Index Terms*—Point clouds, scene flow.**

I. Introduction

THE scene flow refers to the motion fields of one frame of 3D spatial points and the point-wise correspondence relationship between two consecutive frames, which can be used for many tasks of autonomous driving (AD), including object tracking, SLAM, navigation [1], [2], etc. Usually, the main components of scene flow estimation include (1) point feature extraction, (2) inter-frame correlation feature extraction, (3) correspondence map construction, and (4) scene flow estimation. Many scene flow estimation methods have been developed based on cost volume (CV) estimation and various feature extraction schemes [3], [4], [5], [6], [7], [8], achieving much better performance than traditional brute force matching-based methods [9]. The CV is a tensor that contains matching cost information between neighboring 3D point pairs from the input two frames. Most CV methods aggregate point-to-patch correlation features along feature channels within each patch. However, developing a robust scene flow estimation method in dynamic environments has to address the following technical challenges.

1) *Missing Correspondences in Scene Flow Estimation:* Real-world scenarios usually contain a large number of moving objects. The 3D point clouds of these objects might be partially occluded by the objects themselves or other objects [10], and lose the local geometric structure information, as shown in Fig. 1(a)-1, leading to missing correspondences [11], [12], [13]. Besides, downsampling is used to improve the computational efficiency and enlarge the receptive fields. However, it cannot remove the corresponding points from both source and target frames simultaneously, as shown in Fig. 1(a)-2, leading to mis-registrations. The resultant mis-registrations will degrade the performance of scene flow feature extraction. Developing a correlation feature extraction module against mis-registrations and improving the quality of scene flow feature extraction is still an important issue.
2) *Errors from Scene Flow Upsampling:* Most scene flow upsampling methods only rely on geometric distances to search the candidate points through k-NN methods [4], [5], [6], [8], without using any contexture information. The k-NN method may select the candidate points from another nearby object, which will lead to false upsampled scene flow, as shown in Fig. 1(b). Therefore, it is necessary to develop an accurate scene flow upsampling module for fine-grained information propagation.

A number of methods have been proposed to utilize point-to-patch correlation features to predict occlusion maps [11], [12], [13], which are further used to remove those occluded points. However, there are not methods yet to alleviate the missing points and errors from downsampling and upsampling in scene flow estimation. Modern scene flow estimation frameworks utilize encoder-decoder schemes to address these issues through point feature extraction and pyramid reconstruction processes,

Manuscript received 17 September 2023; accepted 7 November 2023. Date of publication 22 November 2023; date of current version 30 November 2023. This letter was recommended for publication by Associate Editor X. Huang and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the Southern University of Science and Technology Research Institute of Trustworthy Autonomous Systems, in part by the Shenzhen Key Laboratory of Robotics and Computer Vision, in part by Shenzhen Fundamental Research Program under Grant JCYJ20220818103006012, and in part the National Natural Science Foundation of China under Grant 62261160654. *(Corresponding author: Qi Hao.)*

Shuaijun Wang is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China, and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: jeanswsj@gmail.com).

Rui Gao is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: 12032493@mail.sustech.edu.cn).

Ruihua Han is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Computer Science, The University of Hong Kong, Pok Fu Lam 999077, Hong Kong (e-mail: hanrh@connect.hku.hk).

Qi Hao is with the Department of Computer Science and Engineering, Shenzhen Key Laboratory of Robotics and Computer Vision, and the Sifakis Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hao.q@sustech.edu.cn).

Digital Object Identifier 10.1109/LRA.2023.3335776

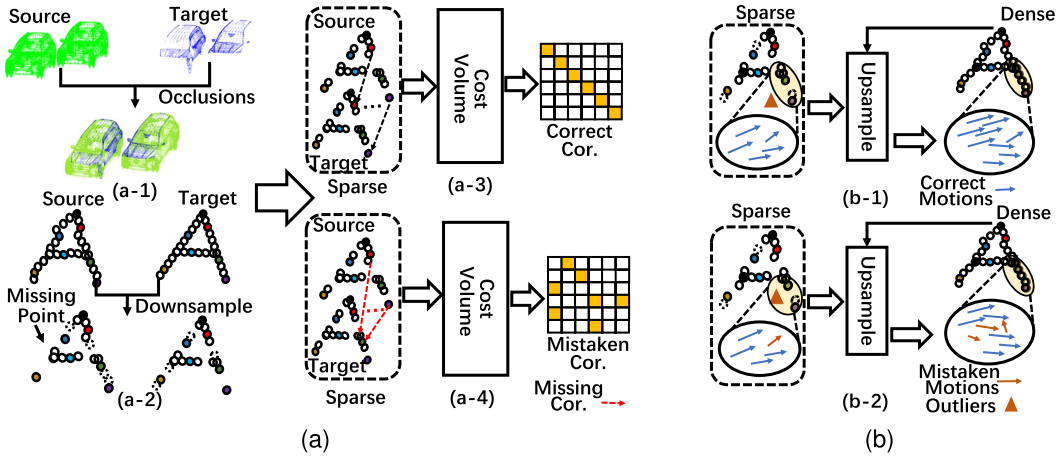

Fig. 1. Illustration of two major challenges for developing a robust scene flow estimation method in dynamic environments. (a) Missing points come from object occlusions and downsampling, respectively, leading to missing correspondences and poor correspondence maps, where Cor. is the abbreviation of correspondence. (b) Upsampling without using semantic information leads to mistaken interpolations of motions.

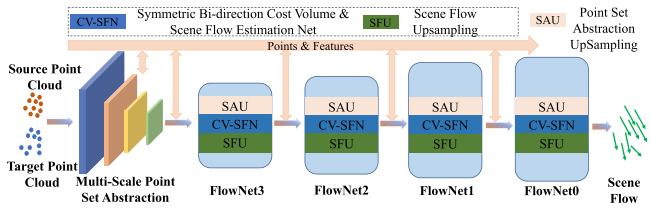

Fig. 2. System diagram of the proposed scene flow estimation framework. The two major modules are (1) the symmetric bi-direction CV based scene flow estimator (CV-SFN), and (2) the scene flow upsampling (SFU), respectively. The point cloud feature extraction module uses the point set abstraction (SA) operation of PointNet++[14].

as shown in Fig. 2. CV modules play a central role for many advanced scene flow estimation methods [11], [12], [13], [15], which primarily focus on feature correlations within local 3D point neighborhoods, overlooking interactions between different sets of points. An MLP-Mixer method, namely Point-Mixer [16], has been developed for point feature extraction through feature aggregation along position channels and feature channels, which can solve the limitations of the CV. However, such an MLP-Mixer concept [17] has not been used for scene flow correlation feature extraction yet.

Therefore, this paper proposes a novel framework[1] for scene flow estimation that integrates a symmetric MLP-Mixer-based CV approach for scene flow estimation, along with a geometric and semantic feature-based upsampling strategy. The main contributions of this paper include

- developing a symmetric bi-direction CV module for inter-frame correlation feature extraction, which is based on MLP-Mixer operations and can be used against the disturbance of point with missing correspondence;
- developing a universal scene flow estimation method, which alternately integrates the inter-set and intra-set operators of the CV module and scene flow estimator to improve scene flow estimation performance and simplify the network structure;
- developing a reliable scene flow upsampling module for fine-grained information propagation, which are based on intra- and inter-frame patch-to-patch geometric and semantic features to better aggregate the candidate scene flow fields.

The rest of this paper is organized as follows. Section II reviews the related work on 3D point correspondence map construction and scene flow estimation. Section III introduces the MLP-mixer operations. Section IV describes the proposed method. Section V provides the experiment and ablation study results. Section VI concludes this paper and outlines future work.

II. Related Work

Dataset: The FlyingThings3D dataset [18] is a large-scale synthetic dataset comprising stereo and RGB-D images. These images are generated from scenes featuring multiple randomly moving objects sampled from ShapeNet [19]. The FT3D scene flow dataset is reconstructed using optical flow maps and ground truth disparities. $\mathrm{FT3D_S}$ was introduced in HPLFlowNet [6] and designed with the removal of occluded points in mind. Its training dataset contains 19,632 frames, while the test dataset comprises 3,816 frames. Conversely, $\mathrm{FT3D_H}$, was introduced in FlowNet3D [3], including the occluded points. The training dataset for $\mathrm{FT3D_H}$ comprises 20,006 frames, with the test dataset consisting of 2,007 frames. The distance range of 3D points in both $\mathrm{FT3D_S}$ and $\mathrm{FT3D_H}$ is limited to less than 35 meters.

On the other hand, the KITTI scene flow dataset [20], [21] serves as an evaluation benchmark for RGB-D stereo-based methods. This dataset provides ground truth disparity maps and optical flow maps. However, the test set lacks disparity information. Therefore, only 150 scenes from the training set were utilized to reconstruct the KITTI scene flow 3D dataset. The KITTI scene flow 3D dataset, referred to as $\mathrm{KITTI_H}$ in FlowNet3D [3], also includes occluded points, where the height of 3D points less than -1.4 meters is removed as the ground points.

Missing Points in Scene Flow Estimation: Many occlusion-aware scene flow estimation methods have been developed [11], [12], [13], [22]. These methods can be divided into two groups:

[1]https://github.com/SJWang2015/MixSF-3DNet.git

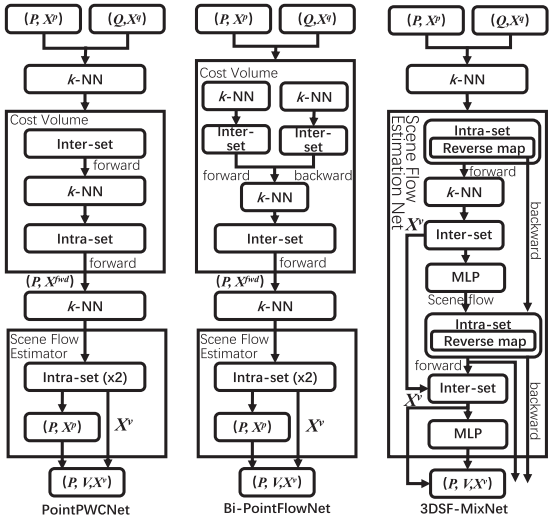

Fig. 3. Illustration of the block design comparison. The block design comparison in 3DSF-MixNet neural network includes an inter-set block for correlation operation between the source and target frames (P and Q), and an intra-set block for scene flow feature extraction within the source frame (P). The backward operation is obtained using the reverse index map of the forward operation. The scene flow prediction is represented by V, while X^p, X^q, and X^v refer to the source point feature, target point feature, and scene flow features, respectively. The forward and backward correlation features are denoted as X^{fwd} and X^{bwd}, respectively. The operator is repeated twice (indicated by $\times 2$).

(1) removing the occluded points and (2) occlusion-weighted scene flow features [11], [22]. The former removes those points in the current frame corresponding to missing points based on the predicted binary occlusion map, which will inevitably lose some geometric information of objects. The latter, such as self-supervised 3D-OGFlow [22], aggregates the features of occluded and non-occluded sets by using two separate CV, which needs a pre-trained scene flow model, Chamfer distance (CD), and Earth Mover's Distance (EMD) as the auxiliary information for measuring the similarity between a pair of point clouds [8], [23]. However, CD is usually insensitive to the mis-registrations between point clouds and prefers to choose the boundary parts of the point clouds to compute the distance, and EMD is generally dominated by the global distributions of point clouds while overlooking the fidelity of detailed structures [23]. The Bi-PointFlowNet [24] uses the asymmetric bidirectional manner to extract the construction map, as shown in Fig. 3. Due to the inconsistent grouped bidirectional features, Bi-PointFlowNet [24] is still under the missing point affection. Therefore, our work develops a symmetric bi-direction CV module for adaptively against the disturbance of point with missing correspondence, which can be used to improve the quality of scene flow features.

Correspondence Feature Construction and Scene Flow Estiamtion: Many correspondence estimation and scene flow generation strategies have been developed for scene flow estimation [3], [4], [5], [6], [7], [8], [25]. Some methods use the stacked convolution operations with pointwise Conv1D/Conv2D operations, similar as the attention mechanism, to build the scene flow features [3], [4], [5], [6], [7], [8], [25]. Due to the non-uniform density and unorderness of point clouds, the stacked convolution operations cannot fully aggregates these candidate points features along the channel dimension in the given spatial size. HPLFlowNet [6], a Bilateral Convolutional Layers (BCL) based method, uses inter-frame correlation features to construct the correspondence map. Usually, BCL based methods [6], [26] can preserve local spatial structure information and achieve effective feature extraction. Still, their performances degrade for sparse or noisy points whose features are less discriminative and more unstable.

Another branch of correspondence estimation strategies, such as FLOT [5], uses Optimal Transport (OT) [27] to iteratively build reliable correspondences, however, whose scalability and numerical stability are rather limited. By contrast, CV based methods [4], [6], [7], [8] use self-attention mechanisms and token-wise mixing to achieve feature aggregation and estimate correspondence. Point-Mixer [16], an MLP-Mixer strategy, has demonstrated that channel-wise feature aggregations of point clouds are better than self-attention mechanisms [15] and dense token-wise interactions [17] in terms of parameter efficiency and correspondence accuracy. However, most scene flow estimation methods design the CV module and the scene flow estimation module independently, which neglects the mutual improvement relationship between the CV module and the scene flow estimator. Therefore, this work develops an MLP-Mixer based CV module to achieve correspondence estimation through channel-mixer operations and permutation invariant operations.

Scene Flow Upsampling: Various upsampling (UP) strategies have been used to construct scene flow fields from sparse levels to dense levels with proper weights based on the point features; those weights can be calculated by two methods: (1) trilinear interpolation based [3], [4], [7], [8]; (2) intra-frame patch features based [6], [28]. The former uses interpolation functions to represent the distances between each anchor point and its neighboring points; the weights of neighboring points are then determined by the distances to the anchor point. The latter uses the similarity of the patch features of each anchor point and its neighboring points; the weights of the neighboring points are then determined by the degree of similarity within each frame. Due to the sparsity of the point clouds, the points of the source and target frames might not consistent, as shown in Fig. 1(b). Upsampling biases primarily occur at the boundaries of different motions and in sparse point cloud areas. The trilinear interpolation-based upsampling method suffers from mutual cancellation between distance weights and motion vectors when a small weight times a large motion vector, resulting in scene flow bias. Challenges arise with intra-set feature similarity weighting due to limited candidate point samples and high feature similarity within the candidate points set with different motions. These challenges lead to weighted interpolation approximating the average candidate point sampling, ultimately affecting subsequent scene flow estimation. This work develops a reliable scene flow upsampling module for fine-grained information propagation, which contains channel-mixer operations based on the product of inter-frame patch features and intra-frame patch-to-patch similarities to better aggregate the candidate local scene flow fields.

III. Preliminary

A. MLP-Mixer Operation

The Point-Mixer method [16], an MLP-Mixer based method for point cloud processing, includes two steps for extracting features, $\mathcal{Y} = \{y_i\}_{i=1}^{N}$, from a point cloud, $\mathcal{P} = \{p_i\}_{i=1}^{N}$, with an initial feature, $\mathcal{X} = \{x_i\}_{i=1}^{N}$. The first step is the channel mixing operation on the local set of points, $\mathcal{S}_i$, searched by k-NN method, which is given by

$$m_j = \varphi_2\{[\varphi_1(x_j); \delta(p_j - p_i)]\}, \; where \; \forall j \in \mathcal{S}_i, \quad (1)$$

where $[\cdot]$ is the feature concatenation operation; $\varphi(\cdot)$ is the channel-mixing MLPs operation, $\delta(\cdot)$ is the positional embedding MLPs operation; m_j is the output feature; x_j and p_j are the j^{th} input point feature and 3D point in the patch S_i centered at the 3D point p_i, respectively.

The second step uses the $\mathrm{softmax}(\cdot)$ and $\mathrm{sum}(\cdot)$ functions to replace the token-mixing for aggregating the patch features, which is given by

$$y_i = \sum_{\forall j \in \mathcal{S}_i} \mathrm{softmax}(m_j) \odot [\varphi_3(x_j) + \delta(p_j - p_i)], \quad (2)$$

where $\mathrm{softmax}(\cdot)$ is used to output the weight of the spatial dimension and makes the MLP-mixer operation permutation-invariant; $\odot$ is the element-wise product; and y_j is the final output point feature by using the Point-Mixer method which is permutation-invariant in processing the unordered points [16].

IV. Method

A. Symmetric CV Based Scene Flow Estimator

1) Symmetric CV: Fig. 2 shows the proposed scene flow estimation neural network in this paper. The CV [8] module aims at building the correspondence map of correlated point features, which can measure the feature similarity between the source and target frames. To obtain a more accurate correspondence map, the proposed CV module has two steps: (1) aggregating the point features within each patch of source and target frames, respectively, as the patch features, and (2) computing the matching cost of the patch features between source and target frames. Therefore, this work first obtains the forward index map from the source frame to the target one (forward direction) based on the k-NN method, as follows:

$$S_j^{p\to q} = k\mathrm{NN}(p_j + v_j, Q), where \forall p_j \in \mathcal{P}, \quad (3)$$

where $S_j^{p\to q}$ is the neighbouring points set in the target frame of the j^{th} 3D point in the source frame, and $v_j \in \mathcal{V}$ is the scene flow element of the j^{th} 3D point in the source frame.

Given a pair of point clouds with initial features in the source and target frames respectively, the forward patch features $x_j^{p,fwd}$ of the source frame is given by

$$\begin{cases} m_i^{fwd} = \varphi_2\{[\varphi_1([q_i - p_j, x_i^q, x_j^p]); \delta(q_i - p_j)]\}, \\ y_i^{fwd} = \varphi_3([q_i - p_j, x_i^q, x_j^p]) + \delta(q_i - p_j), \\ x_j^{p,fwd} = \sum_{\forall i \in \mathcal{S}_j^{p\to q}} \mathrm{softmax}(m_i^{fwd}) \odot y_i^{fwd}, \end{cases}$$
$$where \forall i \in \mathcal{S}_j^{p\to q}, \quad (4)$$

where $\varphi_1(\cdot)$, $\varphi_2(\cdot)$, $\varphi_3(\cdot)$ and $\delta(\cdot)$ are the sub-module of MLP-Mixer operation, respectively, as described in Section III-A, to obtain the patch features, $[\cdot]$ is a concatenation operation along the feature channel dimension, and $x_j^{p,fwd}$ is the forward patch features of the j^{th} 3D point in the source frame, aggregated from both the source and target frames, namely inter-frame forward patch feature.

To save the computation consuming and accurately measure the inter-frame features similarity, the backward correlation feature is obtained by reversely aggregate the weighted point-to-point correlation features, y_i, within the patch of each 3D point in the target frame, as shown in Fig. 4. Using the forward index map obtained by k-NN method, we can symmetrically obtain the backward patch features (from target to source frames). The reverse point-wise weight can be obtained via scatter softmax ($\mathrm{softmax}^s(\cdot)$) based on the forward index map, which is given by

$$w_i^{bwd} = \mathrm{softmax}^s\left(m_i^{fwd}\right), where \forall i \in \mathcal{S}_j^{q\to p}, \quad (5)$$

where $\mathcal{S}_j^{q\to p}$ is the reverse index map based on the forward index map. Using the inter-frame point-wise correlation feature and weight, the backward feature matching cost between the patch features of source and target frames is given by

$$x_j^{q,bwd} = \sum_{\forall i \in \mathcal{S}_j^{q\to p}}{}^s w_i^{bwd} \odot y_i^{fwd}, \quad (6)$$

where $\odot$ is the element-wise product, and $\sum^s(\cdot)$ is the scatter summation based on the reverse index map.

2) Scene Flow Estimator: The final scene flow estimation module include one MLP-based intra-set operation block to output the scene flow feature, and one scene flow regressor to predict the scene flow fields, which are given by

$$x_j^v = \mathrm{MaxPooling}\left(\mathrm{MLP}^x\left([q_i - p_j, x_j^{p,fwd}, x_i^{q,bwd}, f_j^{v,us}]\right)\right),$$
$$where \forall i \in \mathcal{S}_j^q, \quad (7a)$$
$$v_j = \mathrm{MLP}\left(x_j^v\right), where \forall j \in \mathcal{P}, \quad (7b)$$

where $\mathrm{MLP}^x(\cdot)$ is used to obtain the intra-frame scene flow feature, which is the two-layer Conv2D operation, and $\mathrm{MLP}(\cdot)$ is used to predict the scene flow fields. The functionalities of each module will be verified with ablation studies, as described in Section V-C.

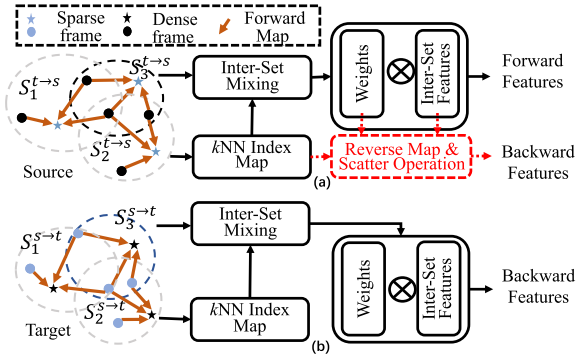

Fig. 4. Illustration of the symmetric inter-frame correlation feature extraction method. (a) Forward inter-frame correlation features are acquired through a k-NN index map using the MLP-Mixer operation, where each 3D point in the source frame seeks neighboring points in the target frame. The symmetric backward (from the target frame to the source one) feature is obtained by aggregating weighted point-to-point correlation features based on the reverse k-NN index map adopted in the forward process. Notably, this process this process in a lookup-table- manner consumes less time than the forward inter-frame correlation feature extraction process. (b) Asymmetric backward inter-frame correlation features are extracted using the MLP-Mixer operation and k-NN methods, where the process replaces the one within the red box in the subfigure (a).

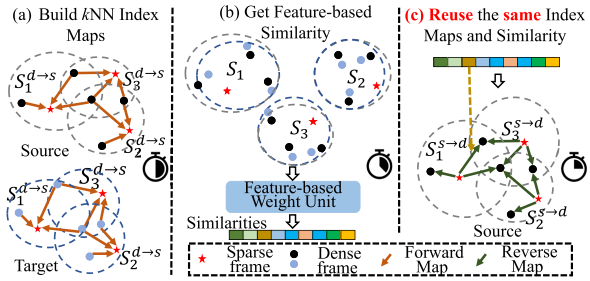

Fig. 5. Illustration of the scene flow upsampling method. The left shows that the each point before upsampling search the neighbouring points in both source and target frames via k-NN. The middle shows the process of obtaining the point-wise feature similarity in groups. The right shows the reverse index map from a sparse frame to a dense one to obtain the corresponding similarity values.

B. Scene Flow Upsampling

This module aims to upsample the scene flow fields from a sparse layer to a dense layer; the output of this module will be used as the input of the scene flow estimation module. The sparse scene flow, $\mathcal{V}_l$, is upsampled to the $\mathcal{V}_{l-1}$ of the $(l-1)^{th}$ layer according to the similarities of patch-to-patch features. The upsampling module includes three steps, as shown in Fig. 5: (1) obtaining the k-NN based index map between the point clouds of sparse and dense frames, (2) aggregating the anchor point feature with each of its neighboring point features in the dense frame and computing their similarities, and (3) interpolating the dense scene flow fields with the reverse index map and corresponding similarity values from multiple local sparse scene flow fields. The sparse point clouds, downsampled by the farthest-point sampling (FPS) method layer-by-layer in neural network, can represent a group of sets to cover the whole dense point clouds. Due to the non-uniform density of point clouds, the number of 3D points of a dense frame covered by each set (centered at one 3D point of sparse point clouds) is different, as shown in Fig. 5. So the distance based interpolation which upsamples sparse scene flow fields to denser ones cannot correctly aggregate neighboring information and are easily affected by the outliers or noises [3], [4], [7], [8], [11].

Therefore, this work first obtains the index map from the dense frame to the sparse one based on the k-NN method, which is the index map for the Softmax and Sum operations in the *scatter* operation to reversely aggregate the weighted sparse scene flow fields within the patch of each 3D point in the dense frame, as follows:

$$\begin{cases} S_j^{p,l-1\to l} = k\mathrm{NN}\left(P^{l-1}, p_j^l\right), \\ S_j^{q,l-1\to l} = k\mathrm{NN}\left(p_j^l + v_j^l, Q^{l-1}\right), \end{cases} \quad where\, \forall p_j^l \in \mathcal{P}_l, \quad (8)$$

where $S_j^{l-1\to l}$ is the neighbouring points in the dense frame (in the $(l-1)^{th}$ scale) of the j^{th} 3D point in the sparse frame (in the l^{th} scale), and $v_j^l \in \mathcal{V}_l$ is the scene flow element of the j^{th} 3D point, p_j, in the sparse frame. Using the k-NN method, we can obtain the patch features based on the $\mathrm{MLPMixer}(\cdot)$ operation (Section III-A) in each patch pair of the source and target frames, respectively. The patch features of the source and target frames are denoted by $\hat{X}_{l-1}^p$ and $\hat{X}_{l-1}^q$, respectively. Then the two patch features are used to learn the weight values of the neighboring sparse scene flow fields of each 3D point in the dense frame, which is given by

$$\begin{cases} w_{i,l-1} = \mathrm{softmax}^s\left(\mathrm{MLP}([\hat{x}_{i,l-1}^q - \hat{x}_{i,l-1}^p])\right), \\ \bar{x}_{j,l-1}^v = \sum_{\forall i \in \mathcal{S}_{j,l}}{}^s w_{i,l-1} \odot x_{i,l}^v, \\ \bar{v}_{j,l-1} = \sum_{\forall i \in \mathcal{S}_{j,l}}{}^s w_{i,l-1} \odot v_{i,l}, \end{cases}$$
$$where\, \forall i \in \mathcal{S}_{j,l-1}^p, \quad (9)$$

where $\mathrm{MLP}(\cdot)$ is the FC operation for enhancing the overlapped similarity features between the sparse (l^{th}) and dense ($(l-1)^{th}$) layers, $\odot$ is the element-wise product, $w_{i,l-1}$ is the weight value of the i^{th} neighbour sparse scene flow vector (in the l^{th} scale) of each 3D point in the $(l-1)^{th}$ scale, $\mathcal{S}_{j,l-1}^p$ is the candidates set in sparse frame through the reverse index map of the $S_j^{p,l-1\to l}$, $\bar{x}_{j,l-1}^v$ is the upsampled scene flow features, and $\bar{v}_{i,l-1} \in \bar{\mathcal{V}}_{l-1}$ is the upsampled scene flow element from the l^{th} scale to the $(l-1)^{th}$ scale.

C. Loss Functions

In order to demonstrate the efficiency of the proposed method, we train our neural network model in a fully supervised manner. Thus, our training loss $\mathcal{L}$ is formulated as follows:

$$\mathcal{L}(\theta) = \sum_{l=1}^{L} \alpha_l \left\| V_l - V_l^{gt} \right\|_2, \quad (10)$$

where $\alpha_i\,(i = 1, 2, 3, 4)$, set to 0.02, 0.04, 0.08, 0.16, is the weight for each layer of scene flow estimation; V_l^{gt} and V_l are the ground truth and estimated scene flow fields at the l^{th} scale, respectively; and θ is the set of learnable parameters of the proposed neural network.

V. Experiments

A. Dataset

With the same configurations as those baseline methods [3], [6], [7], [8], [24], [30], the proposed scene flow inference is performed on both FlyingThings3D ($\mathrm{FT3D_S}$ [6] and $\mathrm{FT3D_H}$ [3]) and KITTI ($\mathrm{KITTI_H}$ [3]). The proposed scene flow neural network is trained on two types of FT3D datasets with only geometric attributes. The learning rate is set to 0.001 with the Adam optimizer [31]. The input point clouds with size 8192 and 2048 are fed into the neural network for training the $\mathrm{FT3D_S}$ and $\mathrm{FT3D_H}$ datasets, respectively.

B. Results

Benchmarks and Evaluation Metrics: We chose the 10 most popular works as the baseline methods, as shown in Table I. There are 4 metrics that are used as the quantitative evaluation standard, which are depicted as follows.

- *EPE3D (m) (End-Point-Error)* indicates the error between the predicted value and the ground-truth value at each corresponding point;
- *AS (%)* (Accuracy Strict) indicates the error value of $EPE3D \leq 0.05$m, and the relative error ratio $\leq 5\%$;
- *AR (%)* (Accuracy Relaxation) indicates the error value of $EPE3D \leq 0.10$m, and the relative error ratio $\leq 10\%$;

TABLE I
COMPARISON OF SCENE FLOW ESTIMATION PERFORMANCES BETWEEN THE BASELINE METHODS AND OUR METHOD

Method	$FT3D_H$				$FT3D_S$				$KITTI_H$ (geo.-only)			
	EPE(m)↓	AS(%)↑	AR(%)↑	Outliers(%)↓	EPE(m)↓	AS(%)↑	AR(%)↑	Outliers(%)↓	EPE(m)↓	AS(%)↑	AR(%)↑	Outliers(%)↓
FlowNet3D [3]	0.1694	25.37	57.85	82.531	0.114	41.2	77.1	60.2	0.1220	18.53	57.03	-
FLOT [5]	0.156	34.3	64.3	70.0	0.052	73.2	92.7	35.7	0.106	45.3	73.7	46.7
HPLFlowNet [6]	0.1318	32.78	63.22	65.46	0.0804	61.44	85.55	42.87	0.1190	30.83	64.76	-
PointPWC-Net [8]	0.1310	34.22	65.78	-	0.0588	73.9	92.76	34.24	0.1094	35.98	73.84	-
FESTA [7]	0.1253	39.52	71.24	-	-	-	-	-	0.0936	44.85	83.35	-
3D-OGFlow [22]	0.1217	55.18	77.67	51.80	-	-	-	-	0.075	70.60	86.93	32.77
FlowStep3D [4]	-	-	-	-	0.0455	81.62	96.14	21.65	-	-	-	-
SCTN [29]	-	-	-	-	0.038	84.70	96.8	26.8	-	-	-	-
Bi-PointFlowNet [24]	0.073	79.1	89.6	27.4	0.028	91.8	97.8	14.3	0.037	87.36	95.46	18.12
3DFlowNet [30]	0.063	79.1	**90.9**	27.9	0.0281	92.90	98.17	14.58	0.056	69.37	91.39	31.55
3DSF-MixNet (Ours)	**0.060**	**79.6**	90.6	**11.55**	**0.025**	**94.87**	**98.97**	**2.14**	**0.031**	**92.63**	**97.28**	**3.14**

Symbol −: This functionality is not supported. Geo.-only means only use the geometric coordinates as the input of the neural network.

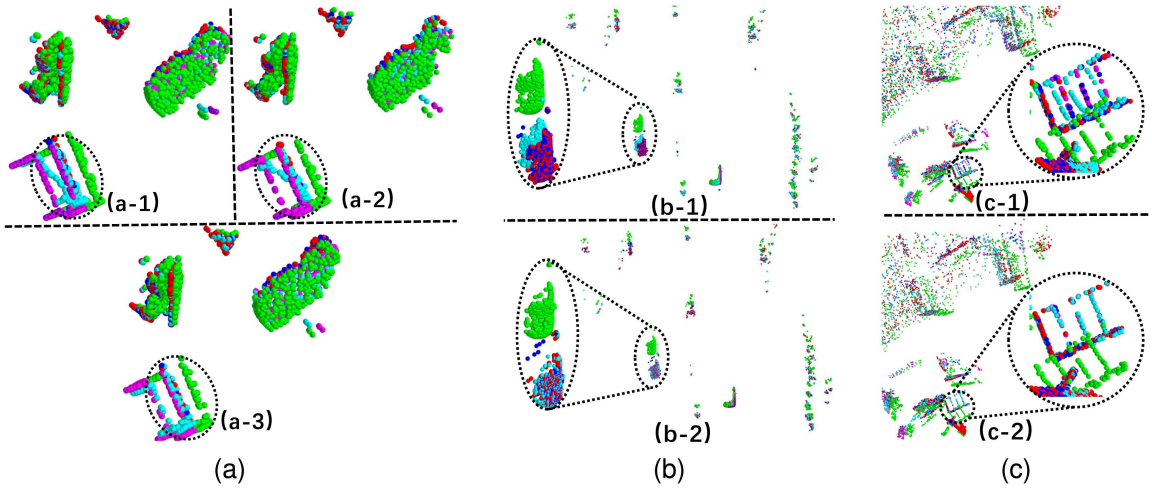

Fig. 6. Illustration of scene flow estimation results by using the proposed 3DSF-MixNet with different module configurations on the $FT3D_H$, and $KITTI_H$ datasets, respectively. Both the 1^{st} and 2^{nd} frames of original point clouds are colored in green and red, respectively. The predicted frames (in cyan) by using 3DSF-MixNet are consistent with the ground truth (in blue). The occluded ground truth points are colored in pink. (a) Scene flow estimation results on $FT3D_H$ using (a-1) PointPWC-Net estimation framework, (a-2) Bi-PointFlowNet estimation framework, and (a-3) our estimation framework, respectively. (b) Scene flow estimation results on $KITTI_H$ using (b-1) asymmetric MLP operations and (b-2) our proposed symmetric MLP-Mixer operations of the CV module in our framework. (c) Scene flow estimation results on $FT3D_H$ with (c-1) trilinear-based upsampling strategy and (c-2) our proposed patch feature-based upsampling strategy in our framework.

- *Outliers (%)* indicates the error value of $EPE3D > 0.10m$, and the relative error ratio $> 30\%$;

Quantitative Evaluation: Table I shows a comparison between our method and the baseline. As shown in Table I, it can be seen that the missing points ($FT3D_H$ and $KITTI_H$) degrade the scene flow estimation performance. Our method outperforms the baseline methods on all evaluation metrics. For both the $FT3D_H$ and $KITTI_H$, 3DSF-MixNet is tested with point clouds in the size of 2048; for $FT3D_S$, in the size of 8192, which are the same for baseline methods.

As shown in the bottom row of Fig. 6, our method can achieve accurate scene flow predictions, although the dataset has occluded points (Fig. 6(a)-2, and (c)-2) and significant motion (Fig. 6(b)-2). For both $FT3D_H$ and $KITTI_H$, 3DSF-MixNet can achieve the best scene flow estimation accuracy with improvements in $EPE3D$ within the range: [0.003, 0.109]/[0.025, 0.091] (unit: meter), in AS within the range of [0.5%, 54.23%]/[23.26%, 74.1%], and in AR within the range of [-0.3%, 32.75%]/[5.89%, 40.25%], respectively. For the dataset of $FT3D_S$, our method also outperforms the baseline methods, and improves the $EPE3D$ by at most 0.089 meters, AS by at most 53.67%, AR by at most 21.87%, $Outliers$ by at least 12.16%, respectively.

C. Ablation Study

In this section, we study the unified scene flow estimation module, CV operation, and the upsampling module in scene flow estimation, respectively.

Unified Scene Flow Estimator: By using different scene flow estimation modules, the method has three frameworks, as shown in Fig. 3, (1) the PointPWC-Net [8] framework includes a single direction CV module and a scene flow estimation module, (2) the Bi-PointFlowNet [24] framework includes an asymmetric bi-direction CV module and a scene flow estimation module, and (3) our method with an integration symmetric bi-direction framework, respectively. Table II compares the scene flow estimation performance of our method with three types of scene flow estimation frameworks using MLP-Mixer operations tested on three different datasets. It can be seen that the proposed scene flow estimation neural network can outperform the framework

TABLE II
COMPARISON OF SCENE ESTIMATION PERFORMANCES USING DIFFERENT FRAMEWORKS OF 3DSF-MIXNET

Dataset	Framework	Metrics			
		EPE(m) ↓	AS(%) ↑	AR(%) ↑	Outliers(%) ↓
$FT3D_H$ (n=2048)	PointPWC-Net [8]	0.109	57.99	79.13	24.52
	Bi-PointFlowNet [24]	0.0731	74.59	88.23	14.35
	Ours	**0.060**	**79.60**	**90.64**	**11.55**
$FT3D_S$ (n=8192)	PointPWC-Net [8]	0.041	87.87	96.94	4.51
	Bi-PointFlowNet [24]	0.034	91.30	97.65	3.09
	Ours	**0.025**	**94.87**	**98.87**	**2.14**
KITTI (n=2048)	PointPWC-Net [8]	0.056	76.50	93.59	7.59
	Bi-PointFlowNet [24]	0.041	83.86	95.84	5.54
	Ours	**0.031**	**92.63**	**97.28**	**3.14**

TABLE III
COMPARISON OF SCENE ESTIMATION PERFORMANCES USING DIFFERENT CV CONFIGURATIONS OF 3DSF-MIXNET

Dataset	CV Module	CV Structure	Metrics				
			EPE(m) ↓	AS(%) ↑	AR(%) ↑	Outliers(%) ↓	Time(s) ↓
$FT3D_H$ (n=2048)	MLP	Asym.	0.106	56.49	78.72	26.00	**0.078**
	MLP	Sym.	0.104	57.08	79.18	25.54	0.075
	MLP-Mixer	Asym.	0.064	77.96	90.24	12.13	0.108
	MLP-Mixer	Sym.	**0.060**	**79.60**	**90.64**	**11.55**	0.099
$FT3D_S$ (n=8192)	MLP	Asym.	0.042	85.94	96.41	5.87	**0.080**
	MLP	Sym.	0.040	87.22	96.64	5.66	0.077
	MLP-Mixer	Asym.	0.026	**95.03**	98.82	**1.69**	0.125
	MLP-Mixer	Sym.	**0.025**	94.87	**98.97**	2.14	0.114
$KITTI_H$ (n=2048)	MLP	Asym.	0.060	69.78	91.31	11.18	**0.078**
	MLP	Sym.	0.056	74.07	92.41	9.42	0.075
	MLP-Mixer	Asym.	0.036	89.46	97.04	3.97	0.108
	MLP-Mixer	Sym.	**0.031**	**92.63**	**97.28**	**3.14**	0.099

Asym. means asymmetric operation adopted in the CV structure. Sym. means symmetric operation adopted in the CV structure. The CV-based symmetric MLP-Mixer operation is the proposed module in this paper.

based on PointPWC-Net [8] and Bi-PointFlowNet [24]. In contrast to other baseline methods which design the CV module and scene flow estimation module independently, the proposed scene flow estimation framework can alternately obtain accurate inter-frame correlation features and the scene flow prediction, as shown in Fig. 6(a).

Symmetric CV based MLP-Mixer operation: To evaluate the proposed symmetric CV-based MLP-Mixer operation, we conducted experiments from two aspects: association operation (pure MLP operation and MLP-Mixer operation) and CV structure (asymmetric and symmetric). The CV module employing the MLP operation is a variant version of our proposed CV module, which the MLP-Mixer operators have replaced with MLP operators accordingly. As shown in Table III, the scene flow estimation method using the symmetric CV structure outperforms those using the asymmetric CV structure scene flow estimation methods in terms of estimation performance and time consumption. The pure MLP operation-based CV only involves calculating feature correlation information exclusively between inter-set 3D points of two frames. In contrast, the MLP-Mixer operation-based CV computes the feature correlation information between inter-set and intra-set 3D points of two frames separately. Consequently, the MLP-Mixer operation-based CV (Fig. 6(b)-2) achieves better scene flow estimation than the pure MLP operation-based CV (Fig. 6(b)-1) at a higher computational cost. Our proposed method, applied to the $FT3D_S$ dataset without occluded points, can achieve more accurate forward scene flow estimation. The method uses an asymmetric CV module to enhance backward correlation features based on the instantly estimated forward scene flow, thus outperforming symmetric operator-based method under the *AS* and the *Outliers* metrics.

TABLE IV
COMPARISON OF SCENE ESTIMATION PERFORMANCES USING DIFFERENT UPSAMPLING MODULE CONFIGURATIONS OF 3DSF-MIXNET

Dataset	Upsample Module	Upsample Structure	Metrics				
			EPE(m) ↓	AS(%) ↑	AR(%) ↑	Outliers(%) ↓	Time(s) ↓
$FT3D_H$ (n=2048)	Trilinear-based	-	0.088	57.12	86.44	16.75	**0.091**
	PFS-based	Unilateral	0.071	75.95	89.06	13.49	0.101
	PFS-based	Bilateral	**0.060**	**79.60**	**90.64**	**11.55**	0.099
$FT3D_S$ (n=8192)	Trilinear-based	-	0.043	80.79	97.37	4.76	**0.110**
	PFS-based	Unilateral	0.028	94.15	98.40	2.19	0.126
	PFS-based	Bilateral	**0.025**	**94.87**	**98.87**	**2.14**	0.114
$KITTI_H$ (n=2048)	Trilinear-based	-	0.039	89.75	96.37	4.09	**0.091**
	PFS-based	Unilateral	0.035	90.46	96.22	4.12	0.101
	PFS-based	Bilateral	**0.031**	**92.63**	**97.28**	**3.14**	0.099

Unilateral means the upsampling weight obtained by directly combining the similarities between a sparse frame and the two dense (source and target) frames. Bilateral means the upsampling weight obtained by using the reverse index map from a sparse frame to a dense one (source or target), which is the proposed method in this paper.

Scene Flow Upsampling: In this section, we conducted experiments by replacing the patch features similarity based (PFS-based) upsampling module with the trilinear interpolation based upsampling module. The experimental results show that two types of PFS-based upsampling methods outperform the trilinear interpolation based upsampling method in scene flow estimation, as shown in Table IV. The proposed upsampling method (Fig. 6(c)-2) can outperform the trilinear interpolation based upsampling method (Fig. 6(c)-1) when facing the occluded points. The proposed bilateral PFS-based upsampling method yields better scene flow estimation than the unilateral PFS-based upsampling method at a lower computational cost, as shown in Table IV.

Efficiency Evaluation: Table V shows a comparison of the time efficiency and pre-trained model size between the proposed method and SOTA methods. Table V shows the run time of 3DSF-MixNet and baseline methods. Although our method does not has the highest computational efficiency, it outperforms the baseline methods in terms of scene flow estimation accuracy.

Discussions: As shown in Table II and IV, our method benefits from unified scene flow estimation module, CV operation, and the upsampling module in scene flow estimation. The proposed symmetric bi-direction CV module is achieved through the MLP-Mixer operations and allows for the extraction of reliable correlation features from the point clouds of the source and target frames. This module progressively improves the quality of inter-frame correlation features through the symmetric bi-directional operations and scene flow predictions. This helps to decrease the impact of missing points on measuring the similarity of point-to-point features. Overall, these modules improve the quality of scene flow estimation.

As shown in Fig. 5, any 3D point of the dense frame can be covered by the neighborhoods of multiple points from a sparse frame. The proposed upsampling scene flow module can provide reliable candidate sparse scene flow fields within these neighborhoods around the given 3D points of the dense frame via the reverse neighbor index map from the sparse frame to the dense one; then those point-to-patch correlation features are used to determine whether these points belong to the same object, which helps the upsampled scene flow to be more accurate and more robust against the bias. The similarity of patch features between the source and target frames can reinforce the weight of the points belonging to the same objects through sparse-to-dense-frame and dense-to-dense-frame correlation aggregation, which can give relevant candidates sparse scene flow fields higher weights.

TABLE V
RUNTIME MEASURED ON SINGLE TESLA-V100 AND MODEL SIZE

Dataset	Metric	Method									
		Ours	FlowNet3D	FLOT	PointPWC-Net	FESTA	3D-OGFlow	FlowStep3D	SCTN	Bi-PointFlowNet	3DFlowNet
$FT3D_S$ (n=8192)	Size (MB) ↓	12.5	5.0	**0.445**	29.7	-	-	8.5	93.6	32.0	19.5
	Time (s) ↓	0.114	0.0511	0.0665	**0.0488**	-	-	1.307	0.1297	0.053	0.059
$KITTI_H$&$FT3D_H$ (n=2048)	Size (MB) ↓	10.7	4.76	**0.445**	29.7	16.1	29.6	-	-	30.5	18.6
	Time (s) ↓	0.099	0.0265	**0.015**	0.0461	0.0865	0.1370	-	-	0.032	0.057

Symbol −: This functionality is not supported.

VI. CONCLUSION

This letter presents a robust scene flow estimation approach based on MLP-Mixer architecture, consisting of a MLP-Mixer based CV module, a symmetric bi-direction CV based scene flow estimator, and a geometric/semantic feature based scene flow upsampling module. By leveraging the temporal and spatial correlations of contexture features of 3D points, the proposed symmetric bi-direction CV can adaptively aggregate the point features with high similarity scores and generate forward and backward correlation features accordingly. The integration of CV and scene flow estimation modules into a single block allows for mutual improvement of each other's performance through alternative usage. The proposed geometric/semantic patch feature similarity based scene flow upsampling module accurately propagates the scene flow fields from sparse to dense scales, thus improving the quality of scene flow estimation. Experimental results show that our method outperforms SOTA baseline methods when tested on $FT3D_S$, $FT3D_H$ and $KITTI_H$ datasets, respectively. However, occlusions heavily degrade the scene flow estimation performance, and we plan to investigate further active sensor navigation schemes to reduce occlusions for scene flow estimation.

REFERENCES

[1] S. Wang, R. Gao, R. Han, S. Chen, C. Li, and Q. Hao, "Adaptive environment modeling based reinforcement learning for collision avoidance in complex scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 9011–9018.

[2] R. Han et al., "RDA: An accelerated collision free motion planner for autonomous navigation in cluttered environments," *IEEE Robot. Automat. Lett.*, vol. 8, no. 3, pp. 1715–1722, Mar. 2023.

[3] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 529–537.

[4] Y. Kittenplon, Y. C. Eldar, and D. Raviv, "FlowStep3D: Model unrolling for self-supervised scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4114–4123.

[5] G. Puy, A. Boulch, and R. Marlet, "FLOT: Scene flow on point clouds guided by optimal transport," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 527–544.

[6] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang, "HPLFlowNet: Hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3254–3263.

[7] H. Wang, J. Pang, M. A. Lodhi, Y. Tian, and D. Tian, "FESTA: Flow estimation via spatial-temporal attention for scene point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14173–14162.

[8] W. Wu, Z. Y. Wang, Z. Li, W. Liu, and L. Fuxin, "PointPWC-Net: Cost volume on point clouds for (self-) supervised scene flow estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 88–107.

[9] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, 1992.

[10] Q. Xu, Y. Zhong, and U. Neumann, "Behind the curtain: Learning occluded shapes for 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2893–2901.

[11] B. Ouyang and D. Raviv, "Occlusion guided scene flow estimation on 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2805–2814.

[12] R. Saxena, R. Schuster, O. Wasenmuller, and D. Stricker, "PWOC-3D: Deep occlusion-aware end-to-end scene flow estimation," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 324–331.

[13] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 614–630.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–10.

[15] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, and C. Zheng, "Synthesizer: Rethinking self-attention for transformer models," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 10183–10192.

[16] J. Choe, C. Park, F. Rameau, J. Park, and I. S. Kweon, "PointMixer: MLP-mixer for point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 620–640.

[17] I. O. Tolstikhinet al., "MLP-Mixer: An all-MLP architecture for vision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 24261–24272.

[18] N. Mayeret al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4040–4048.

[19] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[20] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 427–434, 2015.

[21] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *ISPRS J. Photogrammetry Remote Sens.*, vol. 140, pp. 60–76, 2018.

[22] B. Ouyang and D. Raviv, "Occlusion guided self-supervised scene flow estimation on 3D point clouds," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 782–791.

[23] J. Z. T. W. Z.L.D. L. Tong Wu and L. Pan, "Density-aware Chamfer distance as a comprehensive metric for point cloud completion," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 1–13.

[24] W. Cheng and J. H. Ko, "Bi-PointFlowNet: Bidirectional learning for point cloud based scene flow estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 108–124.

[25] Y. Wei, Z. Wang, Y. Rao, J. Lu, and J. Zhou, "PV-RAFT: Point-voxel correlation fields for scene flow estimation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6954–6963.

[26] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning sparse high dimensional filters: Image filtering, dense CRFS and bilateral neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4452–4461.

[27] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4452–4461.

[28] G. Wang, X. Wu, Z. Liu, and H. Wang, "Hierarchical attention learning of scene flow in 3D point clouds," *IEEE Trans. Image Process.*, vol. 30, pp. 5168–5181, 2021.

[29] B. Li, C. Zheng, S. Giancola, and B. Ghanem, "SCTN: Sparse convolution-transformer network for scene flow estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1254–1262.

[30] G. Wanget al., "What matters for 3D scene flow network," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 38–55.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–13.