Multi-Scale Bidirectional Recurrent Network with Hybrid Correlation for Point Cloud Based Scene Flow Estimation

Wencan Cheng
Department of Artificial Intelligence
Sungkyunkwan University
cwc1260@skku.edu

Jong Hwan Ko
College of Information and Communication Engineering
Sungkyunkwan University
jhko@skku.edu

Abstract

Scene flow estimation provides the fundamental motion perception of a dynamic scene, which is of practical importance in many computer vision applications. In this paper, we propose a novel multi-scale bidirectional recurrent architecture that iteratively optimizes the coarse-to-fine scene flow estimation. In each resolution scale of estimation, a novel bidirectional gated recurrent unit is proposed to bidirectionally and iteratively augment point features and produce progressively optimized scene flow. The optimization of each iteration is integrated with the hybrid correlation that captures not only local correlation but also semantic correlation for more accurate estimation. Experimental results indicate that our proposed architecture significantly outperforms the existing state-of-the-art approaches on both FlyingThings3D and KITTI benchmarks while maintaining superior time efficiency. Codes and pre-trained models are publicly available at https://github.com/cwc1260/MSBRN.

1. Introduction

Scene flow estimation is a fundamental task that estimates the dense 3D motion field of points from two consecutive frames [32, 19]. As it provides the basic motion understanding of a dynamic environment, it is meaningful in a variety of high-level applications such as autonomous driving, augmented reality, and robotics [15].

Early scene flow estimation approaches rely on 2D representations such as RGB images [14, 11, 13, 28, 12]. They basically estimate optical flow and disparity map separately in the 2D space instead of directly estimating scene flow vectors in the 3D space. Recently, with the advances in LiDAR sensors and point cloud-based learning technologies, learning scene flow directly from point clouds has been extensively studied. The pioneering work known as FlowNet3D [19] is the first to introduce hierarchical PointNet++ [26] to directly predict the 3D scene flow based on the point cloud. Following this work, a diverse variety of architectures [33, 36, 9, 32, 15] have been proposed and significantly enhanced performance.

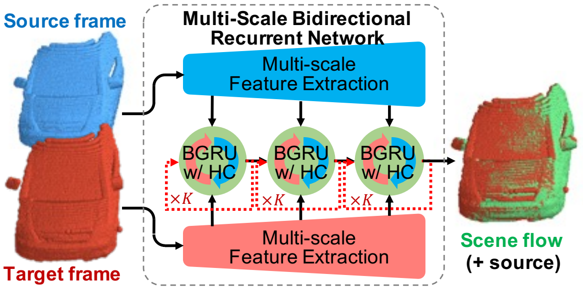
Source frame
Multi-Scale Bidirectional Recurrent Network
Multi-scale Feature Extraction
BGRU w/ HC
BGRU w/ HC
BGRU w/ HC
×K
×K
×K
Multi-scale Feature Extraction
Target frame
Scene flow (+ source)

Figure 1. Illustration of the multi-scale bidirectional recurrent network for scene flow estimation. The multi-scale features extracted from each input frame are fed into novel bidirectional gated recurrent units (BGRUs) that respectively iterates K times for optimizing scene flows on each scale. Introducing hybrid correlation (HC) further improves the performance by searching correspondence in both the Euclidean and latent feature space. The estimated scene flows are warped with the source frame for a clear comparison with the target frame.

Recently, there is a rising trend of iteratively optimizing estimated scene flow by utilizing the recurrent scheme [15, 9, 31] to progressively improve the estimation accuracy. However, they only focus on optimizing single resolution scale which causes large computation latency. On the other hand, another series of works [28, 4] presented their superior efficiency by estimating multi-scale scene flow in a coarse-to-fine manner. However, their single-shot estimation on each scale restricts their performance. To achieve high estimation accuracy while holding high efficiency, we propose an effective and efficient architecture, ***M****ulti*-***S****cale* ***B****idirectional* ***R****ecurrent* ***N****etwork* (MSBRN), that iteratively optimizes coarse-to-fine scene flow. Moreover, in our view, the optimization phases of scene flow can be regarded as temporal sequences, which can benefit from the bidirec-

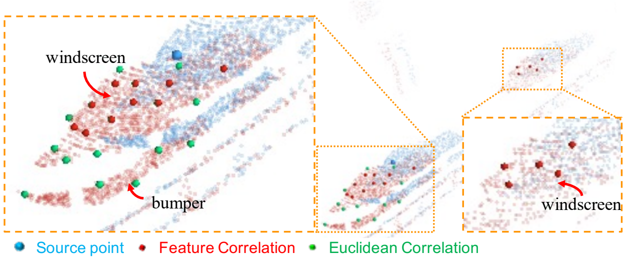

Figure 2. Examples of two variants of correlation extraction. The bold red and green points are the grouped correlated points of the bold blue source point through the feature-induced correlation and Euclidean distance-based correlation, respectively.

tional recurrent architecture [27, 8, 5, 16]. In each specific resolution scale, thus, we apply a novel bidirectional gated recurrent unit (BGRU) to bidirectionally and iteratively augment point features and produce progressively optimized scene flow.

Moreover, we also noticed a common limitation of current point cloud-based flow estimation approaches. They aggregate the grouped neighboring points in the Euclidean space when generating the correspondence features between two point frames. The correspondence grouping used in these methods generally exploits *k-nearest-neighbor* (kNN) that only focuses on a specific local region. Thus, these methods may be harmed by nearby non-correlated points in Euclidean space, as shown in Figure 2. To resolve these issues, we first proposed a feature-induced correlation that collects the nearest neighbor points in the latent feature space. Thus, the grouped points from the feature space are semantically related regardless of their distances in Euclidean space. Afterwards, we combined the proposed correlation with the conventional Euclidean distance-based correlation forming the hybrid correlation. Therefore, the network is able to capture distance-insensitive correspondences as well as local correlations.

We evaluated MSBRN on two challenging benchmarks, the synthetic dataset FlyingThings3D [22] and the real-world LiDAR scan dataset KITTI [23], under both occluded and non-occluded settings. Following the evaluation settings of previous studies, the proposed model is trained only on the FlyingThings3D dataset and evaluated on both datasets to confirm the accuracy and generalization performance. The experimental results show that MSBRN outperforms all other approaches on the FlyingThings3D dataset with 46% and 27% lower errors under the occluded and non-occluded conditions, respectively. Moreover, MSBRN achieves improved generality on the real-world KITTI dataset with 66% and 32% lower errors under the occluded and non-occluded conditions, respectively. Our MSBRN also shows better time efficiency while maintaining higher accuracy compared to other iterative state-of-the-arts.

The key contributions of this paper are summarized as follows:

- We propose a novel multi-scale bidirectional recurrent architecture used for a 3D scene flow estimation task based on point cloud. The model can iteratively and bidirectionally enhance features and scene flow estimations in a coarse-to-fine manner in order to significantly improve the performance while maintaining high efficiency.
- We propose a hybrid correspondence grouping that collects corresponding points from the other point frame in both the latent feature space and Euclidean space.
- The proposed model achieves state-of-the-art performance and generality on the synthetic FlyingThings3D and real-world KITTI benchmarks under both occluded and non-occluded conditions.

2. Related Work

2.1. Scene Flow Estimation on Point Clouds

The first study in solving flow estimation on raw point clouds with learning-based methods was FlowNet3D [19]. FlowNet3D exploited hierarchical PointNet++ [26] and a local correlation learning layer to estimate the scene flow from two raw point cloud frames. Following this scheme, FlowNet3D++ [33] and HCRF-Flow [18] were proposed to further refine the final estimation by introducing auxiliary geometric constraints and high-order CRFs, respectively. However, the performance of these FlowNet3D-based models was restricted by the single-scale flow correlation. To address this flaw, HPLFlownet [10] was proposed in order to capture multi-scale correlations. PointPWC-Net [36] further suggested a coarse-to-fine architecture to regress multi-scale scene flow by applying local correlation extraction hierarchically. Based on the coarse-to-fine design, OGSFNet [24] additionally estimated occlusion masks with an occlusion-aware correlation layer. Furthermore, Bi-PointFlowNet [4] introduced a bidirectional mechanism to bidirectionally propagate point features for informative correlation extraction.

Nevertheless, the above-mentioned approaches all rely on the local correlation, which has a limit on estimation accuracy when corresponding points are out of the local region. However, the hybrid correlation proposed in this work can alleviate this limitation by considering both local correlation and semantic correlation.

2.2. Recurrent Models for Flow Estimation

Many recent studies have demonstrated a reasonable performance by employing a recurrent architecture to itera-

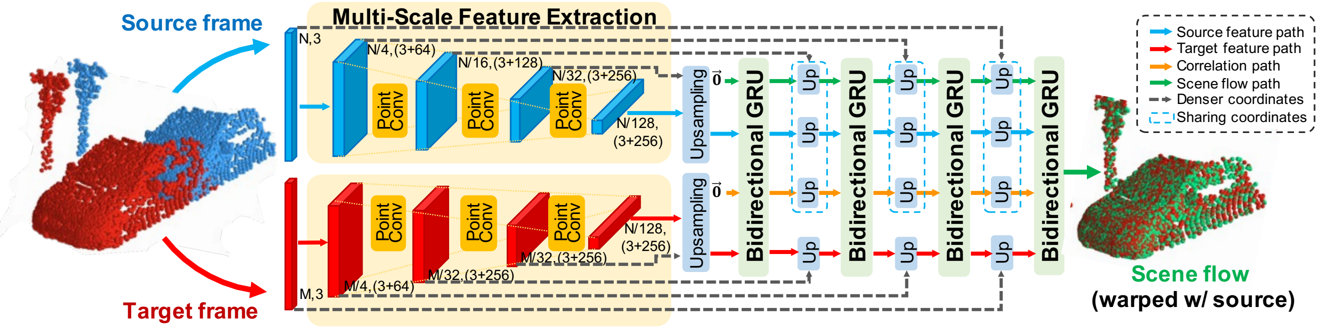

Figure 3. Architecture of MSBRN for scene flow estimation. First, the two consecutive input point frames are fed into the shared hierarchical feature extraction module for multi-level features. At each level, a novel BGRU bidirectionally augments point features and iteratively optimizes correlations and flow predictions. The upsampling layers are adopted between adjacent levels for propagating features and flows from high levels to low levels. The figure is best viewed in color.

tively refine the estimations [13, 21, 30, 31, 34, 15]. Inspired by the previous iterative refinement models [14, 28, 11, 9, 6], IRR [13] reused its whole architecture as a recurrent unit to iteratively optimize the optical flow. Lv et. al. proposed a approach that iterates a learning-based Inverse Compositional algorithm to optimize dense 3D rigid motion. To further improve the efficiency and convergence, RAFT [30] developed a 4D all-to-all correlation volume with the use of a gated recurrent unit (GRU). Based on RAFT, RAFT-3D [31] promoted the recurrent scheme into scene flow estimation by iteratively estimating a dense field of per-pixel SE(3) motion. Afterward, PV-RAFT [34] was proposed to iteratively capture the point-voxel correlation for scene flow estimation from the point cloud inputs. Different from the RAFT-series methods, FlowStep3D [15] and RCP [9] iteratively optimized the correlation that gradually aligns point clouds based on the iterative closest point (ICP) algorithm [3, 1]. FlowStep3D [15] deployed a gated recurrent unit and the source frame warping for iterative updates correlation at a single dense resolution. Similarly, RCP [9] performed iterative point-wise optimization and subsequently introduced the GRU for regularization. Dong *et al.* [6] introduced direct multi-body rigidity constraints to a GRU-based recurrent neural network for robust iterative optimization of scene flow estimation. There are several other approaches [25, 17] that treated scene flow estimation as solving an optimal transport problem. They typically required point-wise features to iteratively optimize the all-to-all correlation.

Nevertheless, the existing iterative methods proceed with the optimization only at a single resolution. To achieve acceptable accuracy, they typically optimize dense resolution, which severely restricts the estimation efficiency. Therefore, we introduce a multi-scale iterative optimization to achieve high accuracy while maintaining high efficiency.

3. MSBRN Architecture

The goal of the scene flow estimation task is to estimate 3D point-wise motion vectors representing the non-rigid transformation from two consecutive point frames sampled from a dynamic scene. To solve this task, we propose MSBRN, a coarse-to-fine bidirectional recurrent architecture with the feature-induced correlation, as shown in Figure 3. MSBRN accepts as input the consecutive source and target frames that are represented by only 3D coordinates, $S = \{s_i \in \mathbb{R}^3\}_{i=1}^N$ and $T = \{t_j \in \mathbb{R}^3\}_{j=1}^M$, where N and M denote the number of points in the source and target frame, respectively. Note that, N and M are not necessarily to be equal because of the sparsity and occlusion in a point cloud. The expected output of MSBRN are 3D scene flow vectors $V = \{v_i \in \mathbb{R}^3\}_{i=1}^N$ that describe the 3D displacement for every point in the source frame aligning with the target frame.

Similar to the previous studies [36, 4], MSBRN is implemented as a coarse-to-fine architecture (Sec. 3.3) with the assistance of two existing modules: a multi-scale feature extraction module that extracts multi-scale point features and an upsampling layer that propagates features from higher scales to lower scales. At each upsampled scale, we deploy a novel BGRU (Sec. 3.1) to iteratively augment correlations and optimize scene flows. BGRU also applies a novel hybrid correlation (Sec. 3.2) in order to enhance the quality of the captured correlations.

3.1. Bidirectional Gated Recurrent Unit

We propose a novel BGRU that not only *optimizes correlations* but also *augments point features* of both frames bidirectionally and iteratively in *various scales*, as shown in Figure 4. In contrast, the previous methods [15, 9] that only apply conventional recurrent units to iteratively optimize only correlations for flow estimation in only one single scale. For each specific scale l, the BGRU accepts source

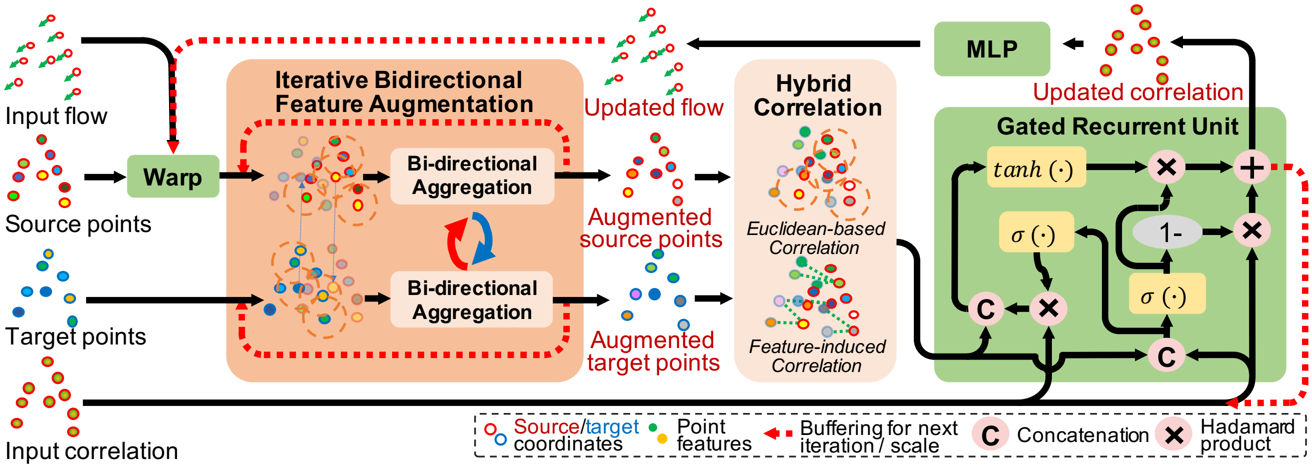

Figure 4. Structure of the bidirectional gated recurrent unit (BGRU). The proposed BGRU accepts source/target point features, flow estimations and correlations from the previous scale as input and hidden states. The source/target point features are first bidirectionally updated and fed to the proposed hybrid correlation for the candidate correlation generation. Subsequent GRU utilizes the candidate correlations for iteratively updating the correlation states.

point coordinates and features $S^{l,0} = \{(s^l, f^{(l,0)})\}$, target point coordinates and features $T^{l,0} = \{(t^l, g^{(l,0)})\}$, correlations $E^{l,0}$ and flow $V^{l,0}$ as input as well as hidden state variables that will be iteratively updated. After K iterations, updated $P^{l,K}$, $Q^{l,K}$, $E^{l,K}$ and $V^{l,K}$ are sent to the subsequent scale. In this section, we focus on one specific scale and omit the notation l for clarity.

Warping: Warping enables the source points to progressively approach the target frame. The estimated scene flows of upsampling layers (for the first iteration only) or previous iterations are immediately accumulated to the source frame, i.e. $s = v^{k-1} + s$. The warped source points are further fed to the subsequent feature augmentation for more accurate correlation extraction.

Iterative Bidirectional Feature Augmentation (IBFA): IBFA accepts as input two frames of point coordinates and corresponding features from the previous $(k-1)$-th iteration, $S^{k-1} = \{(s, f^{k-1})\}$ and $T^{k-1} = \{(t, g^{k-1})\}$. For each point in both frames, the features are augmented bidirectionally for the current k-th iteration by aggregating features from the other frame as follows:

$$f^k = \underset{(t_i, g_i^{k-1}) \in \mathcal{N}_T^{k-1}\{(s, f^{k-1})\}}{SetConv}([t_i - s, g_i^{k-1}, f^{k-1}]), \quad (1)$$

$$g^k = \underset{(s_j, f_j^{k-1}) \in \mathcal{N}_S^{k-1}\{(t, g^{k-1})\}}{SetConv}([s_j - t, f_j^{k-1}, g^{k-1}]), \quad (2)$$

where *SetConv* [19] is composed of a shared multi-layer perceptron and a maxpooling layer, $\mathcal{N}_T\{(s, \cdot)\}$ denotes the nearest neighbor points of s in T, subscript i, j indicate the indices of points in the neighbor group $\mathcal{N}$, and '$[\cdot, \cdot]$' denotes the channel concatenation operator. Note that, the *SetConv* in Equation (1) and (2) share the equivalent parameters. The output are buffered as hidden states for the subsequent iteration or scale.

Iterative Correlation Update: The bidirectionally-augmented features are fed to the hybrid correlation extraction module (Sec. 3.2) to generate candidate correlation features. The candidate correlations serve as the input to a GRU in order to update the state correlation features from the previous iteration. Formally, the correlation features updated by the current iteration is denoted as:

$$z = \sigma(\underset{(s_i, \tilde{e}_i^k) \in \mathcal{N}_{\tilde{S}}^{k-1}\{(s, e^{k-1})\}}{SetConv_z}([s_i - s, e^{k-1}, \tilde{e}_i^k])), \quad (3)$$

$$r_i = \sigma(MLP_r([s_i - s, e^{k-1}, \tilde{e}_i^k])), \quad (4)$$

$$\hat{e}^k = tanh(\underset{(s_i, \tilde{e}_i^k) \in \mathcal{N}_{\tilde{S}}^{k-1}\{(s, e^{k-1})\}}{SetConv_h}([s_i - s, e^{k-1} \odot r_i, \tilde{e}_i^k])), \quad (5)$$

$$e^k = (1 - z) \odot e^{k-1} + z \odot \hat{e}^k, \quad (6)$$

where $\odot$ denotes the Hadamard product.

Scene Flow Estimation: Given the updated correlation feature of each point, an MLP is followed to transform the correlation into a flow vector. Additionally, we introduce a residual mechanism by estimating the flow vector refinement ΔV_k. The final flow estimation is then accumulated as $V^k = V^{k-1} + \Delta V^k$.

3.2. Hybrid Correlation

Feature-induced Correlation: Instead of capturing neighbor points in the conventional Euclidean space, we fetch neighbor points from the latent feature space in order to acquire semantically correlative points. For instance, Figure 2 shows that the proposed correlation can capture semantically similar points (front windscreen) even if they are far away. Thus, the proposed correlation is different from the Euclidean-based correlation that may capture semantically unrelated points (bumper). Formally, given two point sets consisted of 3-dimensional coordinates and C-dimensional features $S = \{(s, f)|s \in \mathbb{R}^3, f \in \mathbb{R}^C\}$ and $T = \{(t, g)|t \in \mathbb{R}^3, g \in \mathbb{R}^C\}$, for each point $(s, f) \in S$, its collected neighbor points $\mathcal{N}_T\{(s, f)\}$ from T satisfy $\delta(f, g_i) < \delta(f, g_j)$, when $\forall(t_i, g_i) \subset \mathcal{N}_T\{(s, f)\}$ and $(t_j, g_j) \subset Q\backslash\mathcal{N}_T\{(p, f)\}$. Note that, $|\mathcal{N}_T\{(s, f)\}| = K$ is the hyperparameter that defines the amount of the top-K neighbor points, and $\delta(\cdot, \cdot)$ is the cosine similarity measure.

After the corresponding points are collected, the features of the collected points are further aggregated by a learnable symmetric function. Hence, the correlation feature for each point $(p, f) \in P$ is formally denoted as:

$$\widetilde{e} = \underset{(t_i,g_i)\in\mathcal{N}_T\{(s,f)\}}{SetConv}([t_i - s, g_i, f]). \tag{7}$$

Hybrid Correlation: Considering that the conventional grouping can capture fine-grained local motion and the proposed feature-induced grouping can capture correspondence regardless of distances, we integrate the two techniques to take advantage of both correlation extraction strategies. Thus, we fuse the collected points with the feature-induced grouping and conventional grouping as a hybrid neighbor set $\mathcal{N}_T\{(s, f)\}$ in Equation (7) for the subsequent correlation aggregation.

3.3. Coarse-to-Fine Architecture

In this section, we thus introduce two existing layers [36] to construct a coarse-to-fine network for multi-scale feature generation and propagation.

Multi-scale Feature Extraction: The multi-scale feature extraction follows the design that is commonly used in point cloud processing [26, 35]. The feature extraction generates L-level pyramid of point features, where the top level is the input point clouds. At each level l, dense input points and their corresponding features are first subsampled through the *furthest point sampling*, which forms a sparse point set. Then, for each subsampled sparse point, the *k-nearest neighbor* groups dense points locally forming a local region for the feature extraction. Afterwards, a *Pointconv* [35] layer is applied to aggregate the features from the grouped local points through dynamic weights learned on their local coordinates, and produced the local feature for each sparse point.

Upsampling Layer: Since each level produces scene flows and iteratively augmented bidirectional features and correlations, we introduced the upsampling layer to propagate these features to the following level for a denser optimization. Following the previous works [36, 4], the upsampling layer adopts the 3D interpolation that aggregates k nearest neighbors with their inverse distances as weights. Specifically, in the case of upsampling correlations from the l-th level to the $(l-1)$-th level, the input to the upsampling layer are l-th level's sparse coordinates $\{x_j^l\}_{j=1}^{N^l}$, l-th level's correlations $\{e_j^l\}_{j=1}^{N^l}$ and $(l-1)$-th level's dense coordinates $\{x_i^{l-1}\}_{i=1}^{N^{l-1}}$, where N^{l-1} and N^l are the number of points and $N^{l-1} > N^l$. The interpolated feature for each point x_i^l in the dense $(l-1)$-th level is defined as:

$$e_i^{l-1} = \frac{\sum_{j=1}^{k} w(x_j^l, x_i^{l-1}) e_j^l}{\sum_{j=1}^{k} w(x_j^l, x_i^{l-1})}, \tag{8}$$

where $w(x_j^l, x_i^{l-1}) = 1/||x_j^l - x_i^{l-1}||_2$, and $k = 3$ as suggested by the previous studies [4, 36].

3.4. Loss Function

Following the previous studies for optical flow estimation [7, 29] and scene flow estimation [32, 36, 4], we train the proposed model in the multi-scale supervision manner. Furthermore, we also supervise the estimations of all intermediate iterations at each scale. All the estimated flows are supervised by the ground truth with the L2 measure. Let $\{v_i^{l,k}\}_{i=1}^{N^l}$ denote the scene flow vectors estimated from the k-th iteration at the l-th level and $\{\hat{v}_i^l\}_{i=1}^{N^l}$ denote the ground truth scene flow vectors of the l-th scale. The training loss is defined as:

$$\mathcal{L} = \sum_{l=0}^{L-1} \alpha^l \sum_{k=1}^{K} \sum_{i=1}^{N^l} \|\hat{v}_i^l - v_i^{l,k}\|_2, \tag{9}$$

where α^l is the weight for scale l. The weights are $\alpha^0 = 0.16$, $\alpha^1 = 0.08$, $\alpha^2 = 0.04$, $\alpha^3 = 0.02$ by default.

4. Experiments

4.1. Experimental Settings

As shown in Fig. 3, we implemented a hierarchical model with $L = 4$ scales. We used $N = M = 8,192$ points as inputs. The point numbers of each scale are defined as $N^1 = 2,048$, $N^2 = 512$, $N^3 = 256$, and $N^4 = 64$. We adopted the synthetic FlyingThings3D [22] dataset and the real-world KITTI Scene Flow 2015 [23] dataset to evaluate our model. As in the previous methods, we trained networks only on the synthetic FlyingThings3D dataset and validated the performance on the FlyingThings3D dataset (Sec. 4.3). Finally, we directly evaluated the model trained on FlyingThings3D without any fine-tuning to validate the generalization ability on the real-world KITTI dataset (Sec. 4.4). The experiments are conducted on an NVIDIA TITAN RTX GPU with PyTorch.

Dataset	Method	EPE3D (m) ↓	ACC3DS ↑	ACC3DR ↑	Outliers3D ↓	EPE2D (px) ↓	ACC2D ↑
FT3D$_s$	FlowNet3D [19]	0.113	0.412	0.771	0.602	5.974	0.569
	HPLFlowNet [10]	0.080	0.614	0.855	0.429	4.672	0.676
	PointPWC [36]	0.059	0.738	0.928	0.342	3.239	0.799
	FLOT [25]	0.052	0.732	0.927	0.357	-	-
	HCRF-Flow [18]	0.048	0.835	0.950	0.261	2.565	0.870
	PV-RAFT [34]	0.046	0.816	0.957	0.292	-	-
	FlowStep3D [15]	0.045	0.816	0.961	0.216	-	-
	RCP [9]	0.040	0.856	0.963	0.197	-	-
	Bi-PointFlowNet [4]	0.028	0.918	0.978	0.143	1.582	0.929
	Ours	**0.015**	**0.973**	**0.992**	**0.056**	**0.833**	**0.970**
KITTI$_s$	FlowNet3D [19]	0.177	0.374	0.668	0.527	7.214	0.509
	HPLFlowNet [10]	0.117	0.478	0.778	0.410	4.805	0.593
	PointPWC [36]	0.069	0.728	0.888	0.265	1.902	0.866
	FLOT [25]	0.056	0.755	0.908	0.242	-	-
	HCRF-Flow [18]	0.053	0.863	0.944	0.179	2.070	0.865
	PV-RAFT [34]	0.056	0.822	0.937	0.216	-	-
	FlowStep3D [15]	0.054	0.805	0.925	0.149	-	-
	RCP [9]	0.048	0.849	0.944	0.122	-	-
	Bi-PointFlowNet [4]	0.030	0.920	0.960	0.141	1.056	0.949
	Ours	**0.011**	**0.971**	**0.989**	**0.085**	**0.443**	**0.985**

Table 1. Comparison of the proposed method with previous state-of-the-art methods on the non-occluded FT3D$_s$ and KITTI$_s$ datasets. All methods are trained only on the FT3D$_s$ dataset.

4.2. Evaluation Measures

For a fair comparison, we adopted the same evaluation measures that are used in the related works [10, 36, 25, 15, 18, 4].

EPE3D, EPE3D$_{full}$ (m): the main evaluation measure measuring 3D end-point-error $\|\hat{v}_i^l - v_i^l\|_2$ averaged over non-occluded points and all points, respectively.

ACC3DS: the percentage of points that satisfy EPE3D $<$ 0.05m or relative error $< 5\%$.

ACC3DR: the percentage of points that satisfy EPE3D $<$ 0.1m or relative error $< 10\%$.

Outliers3D: the percentage of points that satisfy EPE3D $>$ 0.3m or relative error $> 10\%$.

EPE2D (px): 2D end-point-error measured by projecting points back to the 2D image plane, which is a common measure for optical flow evaluation.

ACC2D: the percentage of points that satisfy EPE2D $<$ 3px or relative error $< 5\%$.

4.3. Training and Evaluation on FlyingThings3D

Due to the challenge with labeling for dense point cloud scenes, recent models [10, 36, 25, 15, 18, 4] only utilized the synthetic FlyingThing3D dataset for training. The FlyingThing3D [22] dataset provides 19,640 pairs of frames for training and 3,824 pairs of frames for testing. Each frame contains synthetic stereo and RGB-D images rendered from virtual scenes with multiple moving objects sampled from the ShapeNet [2] dataset. We follow the same preprocessing that generates two subsets: FT3D$_o$ that includes occluded points and FT3D$_s$ that excludes the occluded points, as suggested in [10, 36, 25, 4]. The consecutive input frames are formed by randomly sampling $N = 8,192$ of points from the dataset with non-correspondence.

For training, we used the AdamW optimizer [20] with beta1 = 0.9, beta2 = 0.999. The learning rate is initially set as α = 0.0001 and reduced by half every 80 epochs. We trained the model for a total of 560 epochs. The numbers of iterations for all BGRUs are set to $K_{tr} = 4$ during training and $K_{in} = 4$ during testing.

Results. We compared the performance of the proposed model with other state-of-the-art approaches based on point cloud [19, 10, 36, 25, 15, 24, 9, 4]. As presented in Table 1, the proposed method significantly outperforms all recent state-of-the-art methods with more than 46% reduction of estimation error under the non-occluded condition. When compared to the related iterative methods FlowStep3D [15] and RCP [9], our model achieves an error reduction of 62%.

4.4. Generalization on KITTI

Following the same evaluation strategy as in the recent studies [19, 10, 36, 25, 15, 24, 4], we evaluated the generalization ability of MSBRN on the real-world KITTI [23] dataset with the model that was only trained on the synthetic dataset. The KITTI dataset provides 200 scenes for training and 200 scenes for testing. However, the disparities were missing in the testing scene and parts of the training scenes. Thus, 142 non-occluded scenes (KITTI$_s$, re-

Dataset	Method	EPE3D$_{full}$ (m) ↓	EPE3D (m) ↓	ACC3DS ↑	ACC3DR ↑	Outliers3D ↓
FT3D$_o$	FlowNet3D [19]	0.211	0.157	0.228	0.582	0.804
	HPLFlowNet [10]	0.201	0.168	0.262	0.574	0.812
	FLOT [25]	0.250	0.153	0.396	0.660	0.662
	PointPWC [36]	0.195	0.155	0.416	0.699	0.638
	OGSFNet [24]	0.163	0.121	0.551	0.776	0.518
	Bi-PointFlowNet [4]	0.102	0.073	0.791	0.896	0.274
	Ours	**0.080**	**0.053**	**0.836**	**0.926**	**0.231**
KITTI$_o$	FlowNet3D [19]	0.183	-	0.098	0.394	0.799
	HPLFlowNet [10]	0.343	-	0.103	0.386	0.814
	FLOT [25]	0.130	-	0.278	0.667	0.529
	PointPWC [36]	0.118	-	0.403	0.757	0.496
	OGSFNet [24]	0.075	-	0.706	0.869	0.327
	Bi-PointFlowNet [4]	0.065	-	0.769	0.906	0.264
	Ours	**0.044**	-	**0.873**	**0.950**	**0.208**

Table 2. Comparison of the proposed method with previous state-of-the-art methods on the occluded FT3D$_o$ and KITTI$_o$ datasets. All methods are trained only on the FT3D$_o$ dataset.

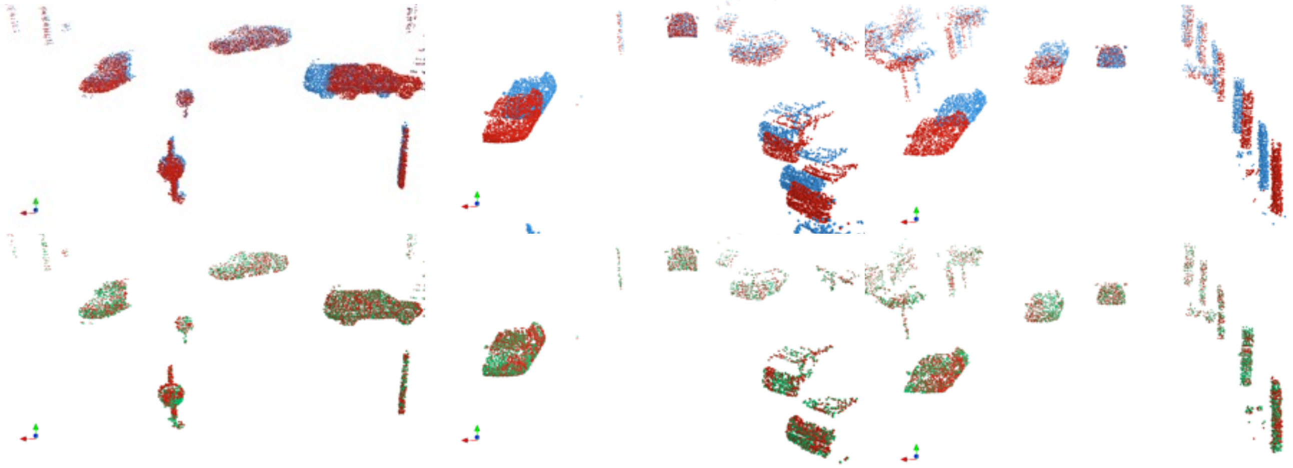

Figure 5. Qualitative results of MSBRN on the non-occluded KITTI dataset. Points are colored to indicate points as from source frame, target frame or as MSBRN estimated points (source frame + scene flow).

moving occluded points) and 150 occluded scenes (KITTI$_o$, remaining occluded points) were available for evaluation with raw point clouds. We removed ground points by height < 0.3 m as suggested by the previous approaches [19, 10, 36, 25, 15, 24] for a fair comparison.

Results. We compared the generalization ability of MSBRN with the existing state-of-the-art methods based on point cloud [19, 10, 36, 25, 15, 24, 9, 4]. As shown in Table 1 and 2, our proposed method represented the state-of-the-art generality when the model is trained on the synthetic datasets and tested on and real-world KITTI dataset. Our method significantly outperforms other methods by a large margin. Focusing on the main EPE3D measure, our method outperforms the recent state-of-the-art with 63% of error reduction on the non-occluded scenes. Compared to the similar iterative methods FlowStep3D [15] and RCP [9], our method shows over 77% of error reduction. Moreover, our model shows the superior generality on the occluded scenes, as shown in Table 2. Our model shows an over 32% reduction of EPE3D$_{full}$.

4.5. Ablation Study

Number of Iterations. The iteration number of the BGRU is a significant factor that affects the estimation accuracy. It is worth noting that the iteration configuration can be different between the training stage (K_{tr}) and the evaluation stage (K_{in}). Therefore, we trained and evaluated our method with the different iteration numbers. Note that, all levels proceed with the same iteration number as configured. Table 3 reports the performance comparison between models with the different numbers of BGRU iterations. The result shows that iteratively adopting the proposed BGRU effectively reduces the estimation error on both the synthetic dataset and the real-world dataset. The model delivers

K_{tr}		EPE3D (m) @ K_{in}					
		1	2	3	4	5	6
FT3D$_s$	1	0.026	0.071	0.101	0.149	0.204	0.281
	2	0.033	0.021	0.030	0.037	0.046	0.049
	3	0.038	0.020	0.016	0.016	0.016	0.018
	4	0.040	0.024	0.017	0.015	0.015	0.015
	5	0.049	0.027	0.019	0.015	0.014	0.014
KITTI$_s$	1	0.038	0.129	0.112	0.146	0.172	0.230
	2	0.039	0.022	0.040	0.069	0.057	0.076
	3	0.039	0.021	0.015	0.014	0.013	0.013
	4	0.040	0.022	0.015	0.011	0.010	0.009
	5	0.057	0.031	0.021	0.017	0.013	0.012

Table 3. Comparison on the EPE2D metric between different training/inference iteration configurations. The model is first trained under a specific K_{tr} on the FT3D$_s$ dataset and then evaluated with various K_{in}s on both FT3D$_s$ and KITTI$_s$ dataset.

Correlation type		Euclidean	Hybrid	Feature
# neighbor (dist : feat)		(32 : 0)	(16 : 16)	(0 : 32)
EPE3D (m)	FT3D$_s$	0.019	**0.016**	0.022
	KITTI$_s$	0.018	**0.015**	0.036

Table 4. Comparison between different numbers of the neighbor points for the correlation extraction. 'Dist' and 'feat' indicate the Euclidean distance-based neighbor points and the feature-induced neighbor points, respectively.

IBFA	HC	GRU	FT3D$_s$ EPE3D (m)	KITTI$_s$ EPE3D (m)
×	×	×	0.029	0.030
√	×	×	0.025	0.025
×	√	×	0.029	0.038
√	×	√	0.019	0.018
√	√	×	0.021	0.018
×	√	√	0.027	0.033
√	√	√	**0.016**	**0.015**

Table 5. Ablations of different components. IBFA, HC and GRU represent whether the iterative bidirectional feature augmentation, hybrid correlation and GRU are deployed in the model, respectively. The iterative models are trained and evaluated with K_{tr}=K_{in}=3.

the optimal estimation performance when K_{tr}=4. Moreover, the model also demonstrated that the performance is saturated at K_{tr}=5. On the other hand, we observe that the model keeps improving for a few more iterations even though the inference K_{in} is larger than the training when the training K_{tr} $\geq$3. However, the improvement is negligible thus we stop evaluating at K_{in}=K_{tr}.

Hybrid Correlation. As mentioned in Section 3.2, the hybrid correlation consists of the Euclidean distance-based correlation and the proposed feature-induced correlation, thus can benefit from both of their advantages. To validate the effectiveness of the hybrid correlation (16:16), we implemented two ablation experiments: a model with only the Euclidean distance-based correlation (32:0) and a model with only the proposed feature-induced correlation (0:32). Note that, the numbers in brackets indicate the number of Euclidean neighbor points and the number of feature-induced neighbor points, respectively. The total number of neighbor points is fixed at 32 for all experiments for a fair comparison. As demonstrated in Table 4, introducing our feature-induced correlation in the conventional Euclidean distance-based approaches can boost performance. However, the proposed feature-induced correlation must work with the Euclidean distance-based correlation. It is because the feature-induced correlation can capture semantically similar points from other instances located far away. Additionally applying the Euclidean distance-based correlation can introduce a proper constraint on the range.

Method	Runtime	Method	Runtime
PV-RAFT [34]	781.1ms	Ours (K_{in}=2)	209.6ms
FlowStep3D [15]	972.7ms	Ours (K_{in}=3)	287.6ms
RCP [9]	2854.6ms	Ours (K_{in}=4)	365.8ms

Table 6. Runtime comparison of iterative methods. The results are evaluated on a single TITAN RTX GPU. PV-RAFT, FlowStep3D and RCP are evaluated with their optimal iteration configurations which are 32, 4 and 14, respectively.

Analysis of different BGRU configurations. To verify the contributions of the proposed components in the BGRU, we incrementally adopted the IBFA, hybrid correlation, and GRU based on a baseline model. Note that, the baseline model replaces IBFA and the hybrid correlation with BFP [4] and the distance-based correlation, respectively, and removes GRU. As shown in Table 5, IBFA, hybrid correlation and GRU all contribute considerably to the effective estimation. In particular, IBFA improves the accuracy by 0.5 mm on KITTI. Furthermore, using the hybrid correlation and GRU mechanism further reduces the estimation error by 0.7 mm and 0.3 mm on KITTI, respectively. However, Table 5 also reveals that the hybrid correlation and GRU must cooperate with IBFA to get convincing improvement.

4.6. Runtime

We compare the running time of our proposed methods to other state-of-the-art iterative approaches [34, 15, 9] in Table 6. All methods are measured on a single NVIDIA TITAN RTX GPU. Table 6 and 1 show that our proposed method outperforms by a large margin in terms of running time while achieving superior accuracy and generality. Even under K_{in}=2, our method also presents outperformed accuracy with faster speed compared to the existing iterative approaches.

5. Conclusion

This paper presented MSBRN, a novel recurrent bidirectional architecture that is capable of iteratively estimating progressively accurate multi-scale 3D scene flows from two consecutive point cloud frames. Our proposed network also utilized the hybrid correlation for improved flow estimation performance. Experimental results showed that our network significantly outperforms previous state-of-the-art methods on two challenging benchmarks. The proposed method also showed superior efficiency compared to existing iterative state-of-the-art. Extensive experiments also demonstrated the excellent effectiveness of the novel components proposed in this work.

Acknowledgement

This work was supported by the Ministry of Science and ICT (MSIT) of Korea under the National Research Foundation (NRF) grant (2022R1A4A3032913) and the Institute of Information and Communication Technology Planning Evaluation (IITP) grants (IITP-2019-0-00421, IITP-2023-2020-0-01821, IITP-2021-0-02052, IITP-2021-0-02068), and by the Technology Innovation Program (RS-2023-00235718) funded by the Ministry of Trade, Industry & Energy (1415187474). Wencan Cheng was partly supported by the China Scholarship Council (CSC).

References

[1] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

[3] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 3

[4] Wencan Cheng and Jong Hwan Ko. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. *arXiv preprint arXiv:2207.07522*, 2022. 1, 2, 3, 5, 6, 7, 8

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[6] Guanting Dong, Yueyi Zhang, Hanlin Li, Xiaoyan Sun, and Zhiwei Xiong. Exploiting rigidity constraints for lidar scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12776–12785, 2022. 3

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 5

[8] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005. 2

[9] Xiaodong Gu, Chengzhou Tang, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Rcp: Recurrent closest point for scene flow estimation on 3d point clouds. *arXiv preprint arXiv:2205.11028*, 2022. 1, 3, 6, 7, 8

[10] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. 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*, pages 3254–3263, 2019. 2, 6, 7

[11] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 1, 3

[12] Junhwa Hur and Stefan Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 312–321, 2017. 1

[13] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2019. 1, 3

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1, 3

[15] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021. 1, 3, 6, 7, 8

[16] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. 2

[17] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1254–1262, 2022. 3

[18] Ruibo Li, Guosheng Lin, Tong He, Fayao Liu, and Chunhua Shen. 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*, pages 364–373, 2021. 2, 6

[19] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 529–537, 2019. 1, 2, 4, 6, 7

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[21] Zhaoyang Lv, Frank Dellaert, James M Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4581–4590, 2019. 3

[22] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 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*, pages 4040–4048, 2016. 2, 5, 6

[23] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 2, 5, 6

[24] Bojun Ouyang and Dan Raviv. Occlusion guided scene flow estimation on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2805–2814, 2021. 2, 6, 7

[25] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 527–544. Springer, 2020. 3, 6, 7

[26] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2, 5

[27] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997. 2

[28] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 3

[29] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012. 5

[30] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 3

[31] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2021. 1, 3

[32] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021. 1, 5

[33] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 91–98, 2020. 1, 2

[34] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. 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*, pages 6954–6963, 2021. 3, 6, 8

[35] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 5

[36] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision*, pages 88–107. Springer, 2020. 1, 2, 3, 5, 6, 7